Konvergenzraten von Bildklassifikatoren basierend auf faltenden neuronalen Netzen

Rate of convergence of image classifiers based on convolutional neural networks

Vom Fachbereich Mathematik der Technischen Universität Darmstadt
Zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.)
Genehmigte Dissertation von Benjamin Walter aus Wiesbaden
Tag der Einreichung: 19. April 2023, Tag der Prüfung: 28. Juni 2023

1. Gutachten: Prof. Dr. Michael Kohler
2. Gutachten: Prof. Dr. Frank Aurzada
Darmstadt – D17

TECHNISCHE UNIVERSITÄT DARMSTADT

Fachbereich Mathematik
Arbeitsgruppe Stochastik

Konvergenzraten von Bildklassifikatoren basierend auf faltenden neuronalen Netzen
Rate of convergence of image classifiers based on convolutional neural networks

Genehmigte Dissertation von Benjamin Walter

Tag der Einreichung: 19. April 2023
Tag der Prüfung: 28. Juni 2023

Darmstadt – D17

Bitte zitieren Sie dieses Dokument als:
URN: urn:nbn:de:tuda-tuprints-243332
URL: http://tuprints.ulb.tu-darmstadt.de/24333

Dieses Dokument wird bereitgestellt von tuprints,
E-Publishing-Service der TU Darmstadt
http://tuprints.ulb.tu-darmstadt.de
tuprints@ulb.tu-darmstadt.de

Danksagung

Ich möchte mich bei all jenen bedanken, die mich während meiner Promotion unterstützt haben.

Mein besonderer Dank gilt Prof. Dr. Michael Kohler, der durch sein Fachwissen, seine Geduld sowie seine Empathie eine unschätzbare Hilfe für mich war und mich stets motiviert hat. Seine fachlichen Anleitungen und Inspirationen haben maßgeblich zur Qualität dieser Arbeit beigetragen. Außerdem möchte ich mich für die Möglichkeiten, die ich während meiner Promotion hatte, bedanken. Hierzu zählen beispielsweise die Einladung als Vortragender zum IMS Annual Meeting in London oder die Forschungsreise nach Montreal.

Ich danke der Deutschen Forschungsgemeinschaft für die finanzielle Förderung meiner Arbeit.

Darüber hinaus danke ich Prof. Dr. Frank Aurzada für das zweite Gutachten meiner Arbeit und den Prüfern Prof. Dr. Jens Lang und Prof. Dr. Yann Disser.

Weiterhin danke ich auch der gesamten Arbeitsgruppe Stochastik für die offene und förderliche Atmosphäre, die sie geboten hat. Ich habe mich dort sehr wohl gefühlt. Insbesondere möchte ich mich bei Sophie und Sebastian bedanken, die mir gerade zu Beginn meiner Promotion eine große Hilfe waren. Ein spezieller Dank geht auch an Selina, die mir viele hilfreiche Tipps für die Lehre gegeben hat.

Ein ganz besonderes Dankeschön geht an meine Eltern und meine Schwester, die mich in jeder Lage motiviert und unterstützt haben sowie diese Arbeit korrekturgelesen haben. Speziell meine Mutter hat dabei vielen mathematischen Monologen, ohne erkennbare Ermüdungserscheinungen, standgehalten. Außerdem bedanke ich mich auch bei meinen Freunden für die Unterstützung, insbesondere bei Lukas, der jederzeit ein offenes Ohr für mich hatte.

Zusammenfassung

In der vorliegenden Arbeit wird das Konvergenzverhalten von Bildklassifikatoren untersucht, die auf faltenden neuronalen Netzen basieren. Es wird gezeigt, dass die Klassifikatoren, welche durch Kleinste-Quadrate-Schätzer als Plug-In Klassifikatoren definiert werden, dimensionsfreie Konvergenzraten für die Differenz des Missklassifikationsrisikos der Schätzung und dem optimalen Missklassifikationsrisiko erzielen und somit den *Fluch der hohen Dimension* umgehen. Diese Analyse liefert eine theoretische Erklärung für die Nützlichkeit der Komponenten von faltenden neuronalen Netzen in der Bildklassifikation, gibt theoretische Anhaltspunkte für eine geeignete Wahl der Netzwerkparameter und liefert einen theoretischen Hinweis für den Vorteil dieser Architekturen gegenüber anderen Klassifizierungsmethoden.

In vorhergehenden Arbeiten konnte im Rahmen der Regressionsschätzung gezeigt werden, dass Neuronale-Netze-Schätzer unter kompositionellen Annahmen an die zu schätzende Regressionsfunktion eine dimensionsfreie Konvergenzrate erreichen. Die so erzielten Ergebnisse lieferten bisher allerdings keine theoretische Begründung für die Überlegenheit von faltenden neuronalen Netzen gegenüber anderen Netzwerkarchitekturen in Anwendungen der Bildklassifikation. Um dies zu ermöglichen, wird der obige Ansatz auf die Bildklassifikation übertragen, indem Struktur- und Glattheitsannahmen an die a-posteriori Wahrscheinlichkeit formuliert werden. Auf diese Weise werden drei statistische Modelle zur Bildklassifikation eingeführt, in denen das Konvergenzverhalten geeigneter Klassifikatoren untersucht wird.

Das erste Modell beinhaltet die folgenden grundlegenden Beobachtungen zur Bildklassifikation: Zum einen hängt die Klasse eines Bildes von der Existenz von bestimmten Objekten ab, die möglicherweise deutlich kleiner als der gesamte Bildbereich sind, und zum anderen lassen sich Teilbereiche eines Bildes hierarchisch aus benachbarten kleineren Bereichen zusammensetzen. Das zweite Modell wird um den Aspekt ergänzt, dass es nur auf den ungefähren relativen Abstand von Merkmalen der Objekte zueinander ankommt. Die für das zweite Modell eingeführten Netzwerkarchitekturen von faltenden neuronalen Netzen enthalten insbesondere lokale Pooling Schichten. Für das dritte Modell wird ein allgemeinerer Rahmen eingeführt, in dem Bilder als Zufallsvariablen mit Werten in einem Funktionenraum betrachtet werden, wobei die beobachtete Stichprobe durch Diskretisierung solcher Zufallsvariablen gebildet wird. Es wird dann ein Modell für die funktionale a-posteriori Wahrscheinlichkeit eingeführt, welches Klassifikationsprobleme beinhaltet, bei denen die Rotation von Objekten um beliebige Winkel irrelevant für eine korrekte Klassifizierung ist. Für dieses Modell wird eine dimensionsfreie Konvergenzrate erzielt, wenn ein von der Auflösung der diskretisierten Bilder abhängiger Fehlerterm vernachlässigt wird.

Für die Verifizierung der entsprechenden Resultate werden Approximationseigenschaften für faltende neuronale Netze hergeleitet und die Komplexität der Funktionsklassen dieser Netzwerkarchitekturen beschränkt.

Abschließend wird das Verhalten der eingeführten Bildklassifikatoren bei endlichem Stichprobenumfang analysiert. Hierfür werden die Klassifikatoren sowohl auf simulierte als auch auf reale Bilddatensätze angewendet und die Ergebnisse mit verschiedenen alternativen Klassifikationsmethoden verglichen.

Abstract

In this thesis, the rate of convergence of image classifiers based on convolutional neural networks is investigated. It is shown that classifiers defined by least squares estimators as plug-in classifiers achieve a rate of convergence for the difference of the misclassification risk of the estimate towards the optimal misclassification risk which does not depend on the input dimension and therefore circumvent the *curse of dimensionality*. This analysis provides a theoretical explanation for the usefulness of convolutional neural network components in image classification, provides theoretical guidance for an appropriate choice of network parameters, and provides theoretical indication for the advantage of these architectures over other classification methods.

In previous work, it has been shown in the context of regression estimation that neural network estimators achieve a rate of convergence which does not depend on the input dimension under compositional assumptions on the regression function. However, these results have not yet provided a theoretical justification for the superiority of convolutional neural networks compared to other network architectures in image classification applications. To enable this, the above approach is applied to image classification by formulating structural and smoothness assumptions on the a-posteriori probability. In this way, three statistical models for image classification are introduced, in which the convergence behavior of suitable classifiers is investigated.

The first model includes the following basic observations about image classification: First, the class of an image depends on the existence of specific objects that are possibly much smaller than the entire image area, and second, subparts of an image can be hierarchically composed of neighboring smaller subparts. The second model is extended by the aspect that only approximate relative distances between features of objects are important. The network architectures of convolutional neural networks introduced for the second model include, in particular, local pooling layers. For the third model, a more general framework is introduced in which images are considered as random variables with values in a functional space, where the observed sample consists of discretizations of such random variables. A model for the functional a-posteriori probability is introduced, which includes classification problems in which the rotation of objects through arbitrary angles is irrelevant concerning a correct classification. For this model, a convergence rate which is independent of the input dimension is achieved if a resolution-dependent error term is neglected.

To verify the corresponding results, approximation properties for convolutional neural networks are derived and the complexity of the classes of these network architectures is bounded.

Finally, the finite sample size behavior of the introduced image classifiers is analyzed. For this purpose, the classifiers are applied to both simulated and real images and the results are compared to alternative classification methods.

Inhaltsverzeichnis

Notation

Dieses Kapitel liefert eine Übersicht der Symbole und Bezeichnungen, die in dieser Arbeit verwendet werden.

Tabelle 1.: Notationsverzeichnis

$\mathbb{N}$	Menge der natürlichen Zahlen.
$\mathbb{N}_0$	Menge der natürlichen Zahlen inklusive 0.
$\mathbb{R}$	Menge der reellen Zahlen.
$\mathbb{R}^+$	Menge der positiven reellen Zahlen.
$\mathbb{R}_0^+$	Menge der positiven reellen Zahlen inklusive der Null.
$\mathbf{P}_X$	Mit $\mathbf{P}_X$ wird das mit der Zufallsvariable X assozierte Maß bezeichnet.
$\mathbf{E}$	Erwartungswert.
η	A-posteriori Wahrscheinlichkeit: $\eta(\mathbf{x}) = \mathbf{P}\{Y = 1 \vert \mathbf{X} = \mathbf{x}\}$.
η_Φ	Funktionale a-posteriori Wahrscheinlichkeit: $\eta_\Phi(\phi) = \mathbf{P}\{Y = 1 \vert \Phi = \phi\}$.
f^*	Bayes-Klassifikator: $f^*(\mathbf{x}) = \begin{cases} 1, & \text{falls } \eta(\mathbf{x}) > \frac{1}{2} \\ 0, & \text{sonst.} \end{cases}$
$\mathcal{D}_n$	Bezeichnet die Datenmenge $\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$.
$L(f_n)$	Missklassifikationsrisiko eins Klassifikators f_n: $L(f_n) = \mathbf{P}\{f_n(\mathbf{X}) \neq Y \vert \mathcal{D}_n\}$.
$\arg\min_{\mathbf{z} \in D} f(\mathbf{z})$	Ist für $D \subseteq \mathbb{R}^d$ und eine Funktion $f : D \to \mathbb{R}$ definiert als $\arg\min_{\mathbf{z} \in D} f(\mathbf{z}) = \{\mathbf{z} \in D : \forall \mathbf{y} \in D \; f(\mathbf{y}) \geq f(\mathbf{z})\}$.
$\Vert\mathbf{x}\Vert$	$\Vert\mathbf{x}\Vert$ bezeichnet die Euklidische Norm von $\mathbf{x} \in \mathbb{R}^d$.
$\Vert\mathbf{x}\Vert_\infty$	$\Vert\mathbf{x}\Vert_\infty$ bezeichnet die Supremumsnorm von $\mathbf{x} \in \mathbb{R}^d$.
$\Vert f\Vert_\infty$	Supremumsnorm einer Funktion $f : \mathbb{R}^d \to \mathbb{R}$: $\Vert f\Vert_\infty = \sup_{\mathbf{x} \in \mathbb{R}^d} \vert f(\mathbf{x})\vert$.
$\Vert f\Vert_{A,\infty}$	Supremumsnorm einer Funktion $f : A \to \mathbb{R}$ auf $A \subseteq \mathbb{R}^d$: $\Vert f\Vert_{A,\infty} = \sup_{\mathbf{x} \in A} \vert f(\mathbf{x})\vert$.
$f \circ g$	Bezeichnet die Komposition der Funktionen f und g.
$\log(\cdot)$	Natürlicher Logarithmus (mit Basis e).
$\lceil z \rceil$	Aufrundungsfunktion: $\lceil z \rceil = \min\{n \in \mathbb{Z} : n \geq z\}$.
$\lfloor z \rfloor$	Abrundungsfunktion: $\lfloor z \rfloor = \max\{n \in \mathbb{Z} : n \leq z\}$.
$I_A(\cdot)$	Indikatorfunktion einer Menge $A \subseteq \mathbb{R}^d$: $I_A(\mathbf{x}) = \begin{cases} 1, & \text{falls } \mathbf{x} \in A \\ 0, & \text{sonst.} \end{cases}$

$1\big\vert_A(\cdot)$	Konstante Funktion $1\big\vert_A : A \to \mathbb{R}$ für ein $A \subseteq \mathbb{R}^d$: $1\big\vert_A(\mathbf{x}) = 1 \quad (\mathbf{x} \in A)$.
$\text{sgn}(\cdot)$	Vorzeichenfunktion: $\text{sgn}(z) = \begin{cases} 1 & , \text{ falls } z > 0 \\ 0 & , \text{ falls } z = 0 \\ -1 & , \text{ falls } z < 0. \end{cases}$
$T_\beta z$	Stützungsfunktion für $z \in \mathbb{R}$ und $\beta > 0$: $T_\beta z = \begin{cases} z & , \text{ falls } -\beta \leq z \leq \beta \\ \beta \cdot \text{sgn}(z) & , \text{ falls } \vert z\vert > \beta. \end{cases}$
A^I	Für eine nichtleere endliche Indexmenge I und $A \subseteq \mathbb{R}$ definiert als $A^I = \{(a_i)_{i \in I} : a_i \in A \ (i \in I)\}$.
$\mathbf{x}_I$	Für eine nichtleere endliche Indexmenge $I \subseteq \{1, \ldots, d\}$ und $\mathbf{x} \in \mathbb{R}^d$ definiert durch $\mathbf{x}_I = (x_i)_{i \in I}$.
$\mathbf{x} + M$	Für $M \subseteq \mathbb{R}^d$ und $\mathbf{x} \in \mathbb{R}^d$ definiert als $\mathbf{x} + M = \{\mathbf{x} + \mathbf{z} : \mathbf{z} \in M\}$.
C_h	Quader der Breite $h > 0$: $C_h = [-h/2, h/2]^2 \subset \mathbb{R}^2$.
$[0,1]^{C_h}$	Raum der (kontinuierlichen) Bilder der Breite $h > 0$: $[0,1]^{C_h} \coloneqq \{f : C_h \to [0,1] \ : \ f \text{ ist eine Abbildung}\}$.
G_λ	Gitter auf dem Quader C_1 der Auflösung $\lambda \in \mathbb{N}$: $G_\lambda = \left\{ \left(\frac{i-\frac{1}{2}}{\lambda} - \frac{1}{2}, \frac{j-\frac{1}{2}}{\lambda} - \frac{1}{2} \right) \ : \ i, j \in \{1, \ldots, \lambda\} \right\}$.
$g_\lambda(\cdot)$	Bezeichnet die Funktion $g_\lambda : [0,1]^{C_1} \to [0,1]^{\{1,\ldots,\lambda\}^2}$ $g_\lambda(\phi) = \left(\phi \left(\left(\frac{i-\frac{1}{2}}{\lambda} - \frac{1}{2}, \frac{j-\frac{1}{2}}{\lambda} - \frac{1}{2} \right) \right) \right)_{(i,j) \in \{1,\ldots,\lambda\}^2}$.
$\tau_{\mathbf{v}}(\cdot)$	Translationsfunktion $\tau_{\mathbf{v}} : \mathbb{R}^d \to \mathbb{R}^d$ für $\mathbf{v} \in \mathbb{R}^d$: $\tau_{\mathbf{v}}(\mathbf{x}) = \mathbf{x} + \mathbf{v} \quad (\mathbf{x} \in \mathbb{R}^d)$.
$rot^{(\alpha)}(\cdot)$	Rotationsfunktion $rot^{(\alpha)} : \mathbb{R}^2 \to \mathbb{R}^2$ für $\alpha \in \mathbb{R}$: $rot^{(\alpha)}(\mathbf{x}) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \cdot \mathbf{x} \quad (\mathbf{x} \in \mathbb{R}^2)$.
c_i	Bezeichnet Konstante $c_i > 0$ mit $i \in \mathbb{N}$ zur Unterscheidung.

1. Einführung

1.1. Motivation

Das Ziel der Bildklassifikation ist es, einem vorliegenden Bild eine Klasse zuzuordnen. Die Klasse des Bildes wird dabei aus einer Liste von endlich vielen Klassen ausgewählt und beschreibt, welche Art von Objekten auf dem Bild dargestellt sind. Zum Beispiel kann die Klassifizierungsaufgabe darin bestehen, zu entscheiden, welche Tierart auf einem Bild dargestellt wird. Die entsprechenden Klassen wären dann beispielsweise „Löwe“, „Tiger“, „Amsel“, u.s.w.. Computer lösen solche Klassifizierungsaufgaben, indem Algorithmen mit bereits klassifizierten Bildern gefüttert werden und mithilfe dieser so genannten Trainingsdaten „lernen“, auch fremde Bilder richtig zu klassifizieren. Man spricht in diesem Zusammenhang dann von *maschinellem Lernen*. Gerade in Anwendungsbereichen der Bildklassifizierung lassen sich die erfolgreichsten Verfahren dem Bereich des *Deep Learning* zuweisen. Die hier verwendeten Methoden basieren auf sogenannten tiefen künstlichen neuronalen Netzen, d.h. Funktionen, die dem biologischen Vorbild der neuronalen Netze im Gehirn nachempfunden sind und aus vielen verdeckten Schichten von künstlichen Neuronen bestehen. In vielen Anwendungen ist die Netzwerkarchitektur der faltenden neuronalen Netze (englisch: *convolutional neural networks*) besonders erfolgreich (Rawat und Wang (2017)). Dies ist eine spezielle Netzwerkarchitektur, bei der das neuronale Netz gewisse Symmetriebedingungen erfüllt.

Bei der in den Jahren 2010 bis 2017 stattgefundenen ImageNet Large Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al. (2015)), einer der größten Wettbewerbe für Objekterkennung, gewannen zum Beispiel seit dem Jahre 2012 ausschließlich Methoden basierend auf tiefen faltenden neuronalen Netzen (siehe Krizhevsky et al. (2012), Zeiler und Fergus (2014), Simonyan und Zisserman (2015), Szegedy et al. (2015) und He et al. (2016)). Auch in den Folgejahren gehören faltende neuronale Netze zu den erfolgreichsten Netzwerkarchitekturen bei der Klassifizierung der in diesem Wettbewerb verwendeten Bilddatensätze (Dai et al. (2021)).

Praktische Anwendungen finden Bildklassifikatoren basierend auf faltenden neuronalen Netzen zum Beispiel beim autonomen Fahren (Grigorescu et al. (2019)), in medizinischen Diagnoseverfahren (Veeling et al. (2018)), bei der Gesichtserkennung (Lu et al. (2020)) oder bei der Erkennung von Bedrohungen durch Sicherheits-Röntgenscans (Morris et al. (2018)). Auch in Gebieten, die nicht der Bildklassifikation zuzuschreiben sind, finden faltende neuronale Netze eine Anwendung. So konnten faltende neuronale Netze bei Sprachübersetzung (Wu et al. (2016)) oder der Bewältigung von Spielen (Silver et al. (2017)) verwendet werden. Wie in Goodfellow et al. (2016) erwähnt, wird außerdem erwartet, dass Methoden des Deep Learnings in Zukunft in vielen weiteren wissenschaftlichen Feldern auftreten werden.

Trotz des großen Erfolgs in praktischen Anwendungen der immer komplexer werdenden Netzwerkarchitekturen, mangelt es an theoretischen Erklärungen für diesen Erfolg (Rawat und Wang (2017)). Da ein besseres theoretisches Verständnis dazu führen kann, die bestehenden Verfahren weiter zu verbessern und das Adaptieren auf neue Anwendungsgebiete zu erleichtern, besteht ein großes Interesse an theoretischen Erklärungen. Ein anderer Grund für dieses Interesse ist, dass es bei Anwendungen in der Medizin oder der Strafjustiz wichtig ist, das Zustandekommen der Vorhersagen zu erklären (Rudin und Ustun (2018)).

Die theoretische Untersuchung von Deep Learning Methoden lässt sich in die drei Bereiche *Approximationsfähigkeit*, *Verallgemeinerung* und *Optimierung* unterteilen (Kutyniok (2022)). Im Bereich der Optimierung wird

der Trainingsalgorithmus theoretisch untersucht, mit dessen Hilfe die neuronalen Netze an die Trainingsdaten angepasst werden. Da wir in dieser Arbeit keinen Bezug auf den Forschungsgegenstand der Optimierung nehmen, sei für einen Überblick über Resultate aus diesem Gebiet auf Fan et al. (2021) verwiesen. Bei der Approximationsfähigkeit geht es um die Approximationsgüte der verwendeten Netzwerkarchitekturen für bestimmte Funktionsklassen, wogegen im Bereich der Verallgemeinerung theoretisch analysiert wird, wie gut die Verfahren auf neuen unbekannten Daten abschneiden. Neben den eben beschriebenen drei Bereichen zur Untersuchung von Deep Learning Methoden gibt es auch noch den etwas neueren Forschungsbereich der *Erklärbarkeit*, in dem versucht wird, Entscheidungen von bereits trainierten neuronalen Netzen zu erklären (Kutyniok (2022)).

Im Rahmen dieser Dissertation leisten wir einen Beitrag zur Approximationsfähigkeit und der Verallgemeinerung von faltenden neuronalen Netzen im Kontext der Bildklassifizierung. Der Inhalt dieser Dissertation lässt sich dabei wie folgt gliedern: Im ersten Kapitel wird die Bildklassifikation im Rahmen der Mustererkennung eingeführt, wichtige Resultate aus der Literatur diskutiert und die neuen Resultate dieser Arbeit zusammenfassend vorgestellt. In Kapitel 2 werden dann drei statistische Modelle für die Bildklassifikation eingeführt. Im anschließenden Kapitel werden Bildklassifikatoren basierend auf faltenden neuronalen Netzen eingeführt und unter den Annahmen der Modelle aus Kapitel 2 drei Resultate zur Konvergenzgeschwindigkeit formuliert. Im weiteren Verlauf von Kapitel 3 werden diese Resultate bewiesen. In Kapitel 4 wird das Verhalten der eingeführten Bildklassifikatoren bei endlichem Stichprobenumfang untersucht, indem die Verfahren auf simulierte und reale Bilddatensätze angewendet werden.

1.2. Einführung der Bildklassifikation im Rahmen der Mustererkennung

In diesem Abschnitt führen wir die Bildklassifikation im Rahmen der Mustererkennung ein. Für eine ausführliche Einführung in die allgemeine Problemstellung der Mustererkennung siehe z.B. Devroye et al. (1996) oder Anthony und Bartlett (1999). Bei der Mustererkennung ist ein $\mathbb{R}^d \times \{1, 2, \dots, M\}$-wertiger Zufallsvektor $(\mathbf{X}, Y)$ für $d, M \in \mathbb{N}$ gegeben. Hierbei besitzt eine (zufällige) $\mathbb{R}^d$-wertige Beobachtung $\mathbf{X}$ die (zufällige) Klasse Y. Wir beschränken uns in dieser Arbeit auf die binäre Bildklassifikation, d.h. wir unterscheiden lediglich zwischen zwei Klassen von Bildern (z.B. „Bild enthält Gesicht“ und „Bild enthält kein Gesicht“). Es gibt verschiedene Methoden, um aus mehreren binären Klassifikatoren auf eine Klassifikation mit mehr als zwei Klassen überzugehen (siehe beispielsweise Géron (2017)). Außerdem beschränken wir uns einfachheitshalber auf Bilder in Graustufen. Um ein (zufälliges) Bild mit (zufälliger) Klasse beschreiben zu können, benötigen wir die Schreibweise $[0,1]^J = \{(a_j)_{j \in J} \,:\, a_j \in [0,1]\}$ für eine nichtleere endliche Indexmenge J. Wir bezeichnen mit $d_1, d_2 \in \mathbb{N}$ die Bilddimensionen und in unserem Fall sei $(\mathbf{X}, Y)$ ein Zufallsvektor mit Werten in

$$[0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \times \{0,1\}.$$

Dabei wird ein (zufälliges) Bild mit der (zufälligen) Klasse Y durch die Zufallsmatrix $\mathbf{X}$ beschrieben, die d_1 Spalten und d_2 Zeilen besitzt. Der Wert der Zufallsmatrix $\mathbf{X}$ an der Position $(i,j) \in \{1,\dots,d_1\} \times \{1,\dots,d_2\}$ entspricht dabei dem entsprechenden Graustufenwert. Das Ziel ist es nun einen Klassifikator

$$f : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \{0,1\}$$

zu finden, für den die Wahrscheinlichkeit einer falschen Vorhersage

$$L(f) \coloneqq \mathbf{P}\{f(\mathbf{X}) \neq Y\}$$

möglichst klein ist. Dieses Problem wird durch den sogenannten *Bayes-Klassifikator*

$$f^*(\mathbf{x}) = \begin{cases} 1, & \text{falls } \eta(\mathbf{x}) > \frac{1}{2} \\ 0, & \text{sonst,} \end{cases}$$

wobei

$$\eta(\mathbf{x}) = \mathbf{P}\{Y = 1|\mathbf{X} = \mathbf{x}\} \quad \left(\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}\right) \tag{1.1}$$

die *a-posteriori Wahrscheinlichkeit* der Klasse 1 bezeichnet, gelöst. Denn dieser minimiert die Wahrscheinlichkeit einer falschen Vorhersage, d.h. es gilt

$$\min_{f:[0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}\to\{0,1\}} \mathbf{P}\{f(\mathbf{X}) \neq Y\} = L(f^*)$$

(vgl. Theorem 2.1 in Devroye et al. (1996)). Da die Verteilung von $(\mathbf{X}, Y)$ im Allgemeinen unbekannt ist, kann der Bayes-Klassifikator f^* nicht berechnet werden. Das Vorgehen ist stattdessen, ausgehend von Beobachtungen

$$\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)\},$$

wobei $(\mathbf{X}, Y)$, $(\mathbf{X}_1, Y_1)$, ..., $(\mathbf{X}_n, Y_n)$ unabhängig identisch verteilte Zufallsvariablen sind, einen Klassifikator

$$f_n = f_n(\cdot, \mathcal{D}_n) : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \{0,1\}$$

zu konstruieren, für den das Missklassifikationsrisiko

$$L(f_n) \coloneqq \mathbf{P}\{f_n(\mathbf{X}) \neq Y|\mathcal{D}_n\}$$

möglichst klein ist. Wir bewerten die statistische Performanz des Klassifikators f_n, indem wir eine obere Schranke für die erwartete Differenz des Missklassifikationsrisikos unserer Schätzung und dem optimalen Missklassifikationsrisikos herleiten. Genauer gesagt konstruieren wir einen Klassifikator f_n, sodass

$$\mathbf{E}\{L(f_n) - L(f^*)\} = \mathbf{E}\{L(f_n)\} - L(f^*)$$

mit wachsendem Stichprobenumfang n möglichst schnell gegen 0 konvergiert.

Wir verwenden *Plug-In Klassifikatoren* der Form

$$f_n(\mathbf{x}) = \begin{cases} 1, & \text{falls } \eta_n(\mathbf{x}) \geq \frac{1}{2} \\ 0, & \text{sonst,} \end{cases} \tag{1.2}$$

wobei

$$\eta_n(\cdot) = \eta_n(\cdot, \mathcal{D}_n) : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R} \tag{1.3}$$

eine Schätzung der a-posteriori Wahrscheinlichkeit (1.1) darstellt. Neben dem Ansatz eines Plug-In Klassifikators, bei dem die a-posteriori Wahrscheinlichkeit geschätzt wird, existieren auch andere Methoden (siehe z.B. Cannings et al. (2020), Breiman et al. (1984), Steinwart und Scovel (2007) und Kim et al. (2021)).

Für die Schätzung (1.3) der a-posteriori Wahrscheinlichkeit gibt es eine ganze Reihe von Ansätzen. Eine Übersicht, in der die gängigsten Verfahren besprochen und theoretisch behandelt werden, findet sich in Györfi et al. (2002). Zu diesen gehören beispielsweise *lokale Durchschnittsschätzer*, wie z.B. *Kern-*, *Partitionen-* oder *Nächste-Nachbar-Schätzer*, *Kleinste-Quadrate-Schätzer* unter Verwendung von Splines, neuronalen Netzen oder radialen Basisfunktionen und *Kleinste-Quadrate-Schätzer mit Strafterm*. In dieser Arbeit werden wir kleinste-Quadrate-Schätzer basierend auf faltenden neuronalen Netzen verwenden. Kleinste-Quadrate-Schätzer minimieren für eine Klasse $\mathcal{F}$ von reellwertigen Funktionen auf $[0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$ das empirische L_2-Risko:

$$\eta_n = \arg\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} |f(\mathbf{X}_i) - Y_i|^2. \tag{1.4}$$

In unserem Fall wird die Klasse $\mathcal{F}$ aus faltenden neuronalen Netzen bestehen, welche wir im nächsten Abschnitt einführen werden. Wegen

$$L(f_n) - L(f^*) \leq 2 \cdot \int |\eta_n(\mathbf{x}) - \eta(\mathbf{x})| \, \mathbf{P}_{\mathbf{X}}(d\mathbf{x})$$

(siehe z.B. Korollar 6.1 in Devroye et al. (1996)), gilt unter Verwendung der Cauchy-Schwarz-Ungleichung

$$\mathbf{E}\{L(f_n)\} - L(f^*) \leq 2 \cdot \sqrt{\mathbf{E}\left\{\int |\eta_n(\mathbf{x}) - \eta(\mathbf{x})|^2 \, \mathbf{P}_{\mathbf{X}}(d\mathbf{x})\right\}}. \tag{1.5}$$

Wir erhalten daher eine obere Schranke für $\mathbf{E}\{L(f_n)\} - L(f^*)$, indem wir eine obere Schranke für den erwarteten L_2-Fehler

$$\mathbf{E}\left\{\int |\eta_n(\mathbf{x}) - \eta(\mathbf{x})|^2 \, \mathbf{P}_{\mathbf{X}}(d\mathbf{x})\right\} \tag{1.6}$$

der Schätzung η_n der a-posteriori Wahrscheinlichkeit herleiten. Einen Schätzer η_n der a-posteriori Wahrscheinlichkeit herzuleiten, sodass der L_2-Fehler klein ist, ist ein Spezialfall der sogenannten Regressionsschätzung, welche wir in Abschnitt 1.4 einführen werden. Verwenden wir Ungleichung (1.6), um eine obere Schranke für den Fehler $\mathbf{E}\{L(f_n)\} - L(f^*)$ herzuleiten, lösen wir das Klassifikationsproblem daher mittels Regressionsschätzung.

1.3. Bildklassifikatoren basierend auf faltenden neuronalen Netzen

Wir führen im Folgenden Bildklassifikatoren ein, die auf faltenden neuronalen Netzen basieren. Faltende neuronale Netze wurden von LeCun et al. (1989) erstmalig vorgestellt und durchlaufen seitdem eine stetige Entwicklung zu immer komplexeren Netzwerkarchitekturen. Die wichtigsten Komponenten von faltenden neuronalen Netzen sind *faltende Schichten*, *Pooling Schichten* und *vollverbundene neuronale Netze*.

Zunächst führen wir *vollverbundene (mehrschichtige vorwärtsgerichtete) neuronale Netze* ein, welche $L_{net} \in \mathbb{N}$ verdeckte Schichten und $k_r \in \mathbb{N}$ Neuronen in Schicht $r \in \{1, \ldots, L_{net}\}$ besitzen. Als Aktivierungsfunktion verwenden wir die sogenannte *ReLU-Aktivierungsfunktion* $\sigma : \mathbb{R} \to \mathbb{R}_0^+$, welche durch

$$\sigma(x) = \max\{x, 0\} \quad \big(x \in \mathbb{R}\big)$$

definiert ist. Dies ist die übliche Wahl der Aktivierungsfunktion bei Netzwerkarchitekturen, die in der Bildklassifikation eingesetzt werden (siehe beispielsweise Nwankpa et al. (2018), Ramachandran et al. (2018) und Sharma et al. (2020)). Für eine Übersicht aller gängigen Aktivierungsfunktionen siehe beispielsweise Sharma et al. (2020). Ein vollverbundenes neuronales Netz ist dann eine Funktion $g : \mathbb{R}^d \to \mathbb{R}$ der Form

$$g(\mathbf{x}) = \sum_{i=1}^{k_{L_{net}}} w_i^{(L_{net})} \cdot g_i^{(L_{net})}(\mathbf{x}) + w_0^{(L_{net})} \quad \big(\mathbf{x} \in \mathbb{R}^d\big), \tag{1.7}$$

wobei

$$w_0^{(L_{net})}, \ldots, w_{k_{L_{net}}}^{(L_{net})} \in \mathbb{R} \tag{1.8}$$

die äußeren Gewichte bezeichnen und die Funktionen $g_i^{(L_{net})}$ für $i \in \{1, \ldots, k_{L_{net}}\}$ rekursiv definiert sind durch

$$g_i^{(r)}(\mathbf{x}) = \sigma\left(\sum_{j=1}^{k_{r-1}} w_{i,j}^{(r-1)} \cdot g_j^{(r-1)}(\mathbf{x}) + w_{i,0}^{(r-1)}\right) \quad \big(\mathbf{x} \in \mathbb{R}^d\big)$$

für $i \in \{1, \dots, k_r\}$, $r \in \{1, \dots, L_{net}\}$, $k_0 = d$, die inneren Gewichte

$$w_{i,0}^{(r-1)}, \dots, w_{i,k_{r-1}}^{(r-1)} \in \mathbb{R} \tag{1.9}$$

und

$$g_i^{(0)}(\mathbf{x}) = x_i \qquad \left(\mathbf{x} = (x_1, \dots, x_d)^T \in \mathbb{R}^d\right)$$

für $i \in \{1, \dots, k_0\}$. In praktischen Anwendungen der Bildklassifikation bilden vollverbundene neuronale Netze meist den Abschluss einer Architektur von faltenden neuronalen Netzen und folgen einer globalen Pooling Schicht. Wir sprechen in dieser Arbeit von standardmäßigen vorwärtsgerichteten neuronalen Netzen, wenn die neuronalen Netze der obigen Form (1.7) sind und keine spezifischen Symmetrieeigenschaften aufweisen, wie z. B. die im Folgenden vorgestellten faltenden neuronalen Netze.

Als Nächstes werden wir faltende Schichten einführen, welche sich als Spezialfall von vollverbundenen Schichten auffassen lassen, bei denen die Gewichte gewissen Symmetriebedingungen unterliegen. Sie besitzen $k' \in \mathbb{N}$ Eingabekanäle und $k \in \mathbb{N}$ Ausgabekanäle, welche für $i_1, i_2 \in \mathbb{N}$ jeweils aus $i_1 \cdot i_2$ Neuronen bestehen. Da die Neuronen in den Kanälen in Ebenen angeordnet sind (siehe Abbildung 1.3), führen wir die Indexmenge $I = \{1, \dots, i_1\} \times \{1, \dots, i_2\}$ ein. Außerdem hängt eine faltende Schicht von einer Filtergröße $M \in \mathbb{N}$, einem *Zero-Padding* Parameter $P \in \{1, \dots, M\}$ und den folgenden trainierbaren Gewichten ab:

1. Einer *Gewichtsmatrix* (sogenannter Filter)

$$(w_{i,j,s_1,s_2})_{1 \le i,j \le M, s_1 \in \{1,\dots,k'\}, s_2 \in \{1,\dots,k\}} \tag{1.10}$$

2. und dem *Bias-Term*

$$(w_{s_2})_{s_2 \in \{1,\dots,k\}}. \tag{1.11}$$

Wir fassen die Gewichte einer faltenden Schicht durch den Gewichtsvektor

$$\mathbf{w} = \left((w_{i,j,s_1,s_2})_{1 \le i,j \le M, s_1 \in \{1,\dots,k'\}, s_2 \in \{1,\dots,k\}}, (w_{s_2})_{s_2 \in \{1,\dots,k\}}\right)$$

zusammen. Eine faltende Schicht ist dann eine Funktion

$$o_{(k',k),M,P,\mathbf{w}} : \mathbb{R}^{I \times \{1,\dots,k'\}} \to \mathbb{R}_0^{+\, I \times \{1,\dots,k\}}$$

der Form

$$\left(o_{(k',k),M,P,\mathbf{w}}(\mathbf{x})\right)_{(i,j),s_2} = \sigma\left(\sum_{s_1=1}^{k'} \sum_{\substack{t_1,t_2 \in \{1,\dots,M\} \\ (i+t_1-P, j+t_2-P) \in I}} w_{t_1,t_2,s_1,s_2} \cdot x_{(i+t_1-P, j+t_2-P), s_1} + w_{s_2}\right) \quad \left(\mathbf{x} \in \mathbb{R}^{I \times \{1,\dots,k'\}}\right) \tag{1.12}$$

für $(i,j) \in I$ und $s_2 \in \{1, \dots, k\}$ (für eine Darstellung der Berechnung siehe Abbildung 1.1). Befindet sich die Position $(i,j) \in I$ am Rand eines Eingabekanals $(x_{(i,j),s_1})_{(i,j) \in I}$, besteht die innere Summe in (1.12) wegen der Einschränkung $(i + t_1 - P, j + t_2 - P) \in I$ (möglicherweise) aus weniger als M^2 Summanden. Statt dieser Einschränkung hätte man auch zunächst den entsprechenden Eingabekanal an den Rändern mit Nullen auffüllen können (siehe Abbildung 1.2), weswegen diese Methode Zero-Padding genannt wird. Der Parameter P beschreibt dabei anschaulich, an welcher Seite eines Eingabekanals ein Filter $(w_{t_1,t_2,s_1,s_1})_{1 \le t_1 \le t_2 \le M}$ den Eingabekanal wie weit überlappt. Wählen wir $P = 1$, erhalten wir ein *einseitiges Zero-Padding* (für eine Darstellung siehe Abbildung 1.2a). In vielen Anwendungen werden ungerade Filtergrößen M verwendet, um

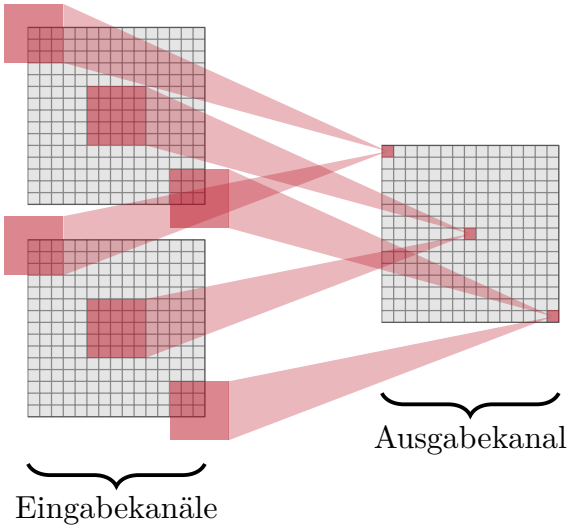


Abbildung 1.1.: Berechnung eines Ausgabekanals mit den Parametern $k' = 2$, $I = \{1, \dots, 15\}^2$, $M = 5$ und $P = 3$.

mit der Wahl $P = \lceil M/2 \rceil$ ein *symmetrisches Zero-Padding* zu erhalten (für eine Darstellung siehe Abbildung 1.2b). Das Zero-Padding hat zur Folge, dass ein Ausgabekanal einer faltenden Schicht nicht weniger Neuronen als die entsprechende Eingabeschicht besitzt, d.h. sowohl Eingabe- als auch Ausgabekanäle bestehen aus $i_1 \cdot i_2$ Neuronen.

Im Fall eines einzigen Eingabekanals ($k' = 1$) identifizieren wir $\mathbb{R}^{I \times \{1\}}$ mit $\mathbb{R}^I$ und definieren eine faltende Schicht durch eine Funktion

$$o_{(1,k),M,P,\mathbf{w}} : \mathbb{R}^I \to \mathbb{R}_0^{+\, I \times \{1,\dots,k\}},$$

welche durch

$$\left(o_{(1,k),M,P,\mathbf{w}}(\mathbf{x})\right)_{(i,j),s_2} = \sigma\left(\sum_{\substack{t_1,t_2 \in \{1,\dots,M\} \\ (i+t_1-P, j+t_2-P) \in I}} w_{t_1,t_2,1,s_2} \cdot x_{(i+t_1-P,j+t_2-P)} + w_{s_2}\right) \quad \left(\mathbf{x} \in \mathbb{R}^I\right)$$

für $(i,j) \in I$ und $s_2 \in \{1, \dots, k\}$ definiert ist.

Auf eine faltende Schicht folgt entweder die nächste faltende Schicht oder eine sogenannte Pooling Schicht. Diese verringert die Auflösung der Ausgabekanäle der vorangegangenen faltenden Schicht. Ein Beispiel ist eine sogenannte *lokale Max-Pooling Schicht*, welche eine von dem Parameter $s \in \mathbb{N}$ abhängige Funktion

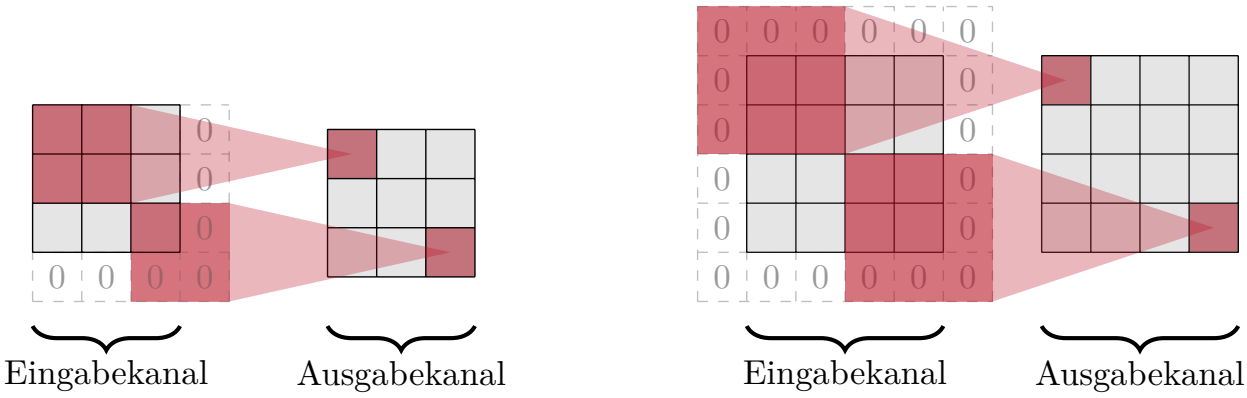
$$f_{max}^{(s)} : \mathbb{R}^{I \times \{1,\dots,k\}} \to \mathbb{R}^{\left\{1,\dots,\left\lceil \frac{i_1}{s} \right\rceil\right\} \times \left\{1,\dots,\left\lceil \frac{i_2}{s} \right\rceil\right\} \times \{1,\dots,k\}}$$

darstellt. Die Ausgabe der lokalen Max-Pooling Schicht berechnet sich durch Zusammenfassen lokaler Bereiche der Größe s via Maximumsberechnung durch

$$\left(f_{max}^{(s)}(\mathbf{x})\right)_{(i,j),s_2} = \max_{(j_1,j_2) \in \left(\{(i-1)\cdot s+1\dots,i\cdot s\} \times \{(j-1)\cdot s+1\dots,j\cdot s\}\right) \cap I} x_{(j_1,j_2),s_2} \quad \left(\mathbf{x} \in \mathbb{R}^{I \times \{1,\dots,k\}}\right) \tag{1.13}$$

für $(i,j) \in \{1, \dots, \lceil i_1/s \rceil\} \times \{1, \dots, \lceil i_2/s \rceil\}$ und $s_2 \in \{1, \dots, k\}$. Ein weiteres Beispiel einer Pooling Schicht ist eine *Subsampling Schicht*, welche ebenfalls eine von einem Parameter $s \in \mathbb{N}$ abhängige Funktion

$$f_{sub}^{(s)} : \mathbb{R}^{I \times \{1,\dots,k\}} \to \mathbb{R}^{\left\{1,\dots,\left\lceil \frac{i_1}{s} \right\rceil\right\} \times \left\{1,\dots,\left\lceil \frac{i_2}{s} \right\rceil\right\} \times \{1,\dots,k\}}$$



(a) Einseitiges Zero-Padding für $M = 2$, $P = 1$ und $I = \{1, 2, 3\}^2$.

(b) Symmetrisches Zero-Padding für $M = 3$, $P = \lceil M/2 \rceil = 2$ und $I = \{1, \ldots, 4\}^2$.

Abbildung 1.2.: Darstellung des Zero-Paddings.

darstellt. Hier werden lediglich die Werte in den oberen linken Ecken der lokalen Bereiche weitergegeben:

$$\left(f_{sub}^{(s)}(\mathbf{x})\right)_{(i,j),s_2} = x_{((i-1)\cdot s+1,(j-1)\cdot s+1),s_2} \quad \left(\mathbf{x} \in \mathbb{R}^{I\times\{1,\ldots,k\}}\right) \tag{1.14}$$

für $(i, j) \in \{1, \ldots, \lceil i_1/s \rceil\} \times \{1, \ldots, \lceil i_2/s \rceil\}$ und $s_2 \in \{1, \ldots, k\}$. In Anwendungen werden Subsampling Schichten meist durch sogenannte *convolutional strides* direkt in den vorangegangen faltenden Schichten realisiert (vgl. Goodfellow et al. (2016)). Sowohl lokale Max-Pooling Schichten als auch Subsampling Schichten besitzen keine trainierbaren Gewichte und sind lediglich an den Parameter s gekoppelt. Einen Überblick gängiger Pooling Methoden findet sich z.B. in Gholamalinezhad und Khosravi (2020).

Da die Ausgabe unserer faltenden neuronalen Netze reellwertig sein soll, benötigen wir eine *Ausgabeschicht* $f_{out} : \mathbb{R}^{I\times\{1,\ldots,k\}} \to \mathbb{R}$, welche aus den Ausgaben einer faltenden Schicht oder einer Pooling Schicht einen reellen Wert berechnet. In unserem Fall besteht die Ausgabeschicht aus einer globalen Max-Pooling Schicht und einer linearen Schicht. Die Ausgabeschicht verwendet die trainierbaren Gewichte

$$\mathbf{w}_{out} = (w_s)_{s\in\{1,\ldots,k\}}, \tag{1.15}$$

und hängt außerdem von Ausgabeschranken $\mathbf{A} = (A_1, A_1', A_2, A_2') \in \mathbb{N}^4$ ab, wobei $1 \leq A_j \leq A_j' \leq i_j$ für $j = 1, 2$ gilt. Die Ausgabe berechnet sich dann gemäß

$$f_{out}^{(\mathbf{A})}(\mathbf{x}) = \max_{\substack{i\in\{A_1,\ldots,A_1'\},\\ j\in\{A_2,\ldots,A_2'\}}} \sum_{s_2=1}^{k} w_{s_2} \cdot x_{(i,j),s_2} \quad \left(\mathbf{x} \in \mathbb{R}^{I\times\{1,\ldots,k\}}\right). \tag{1.16}$$

Die faltenden neuronalen Netze, die wir in dieser Arbeit behandeln werden, entsprechen dann einer Funktion $f : [0, 1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}} \to \mathbb{R}$ der Form

$$f(\mathbf{x}) = \left(f_{out}^{(\mathbf{A})} \circ f_L \circ \cdots \circ f_1\right)(\mathbf{x}) \quad \left(\mathbf{x} \in [0, 1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}}\right), \tag{1.17}$$

wobei die Funktionen $f_r : \mathbb{R}^{I_{r-1}\times k_{r-1}} \to \mathbb{R}^{I_r\times k_r}$ $(r = 1, \ldots, L)$ faltenden Schichten der Form (1.12) oder lokalen Max-Pooling bzw. Subsampling Schichten der Form (1.13) und (1.14) entsprechen (für eine Darstellung eines faltenden neuronalen Netzes der Form (1.17) siehe Abbildung 1.3). Die durch die endlichen Indexmengen

$$I_{r-1} \times \{1, \ldots, k_{r-1}\} \subset \mathbb{N}^3 \quad (r = 1, \ldots, L)$$

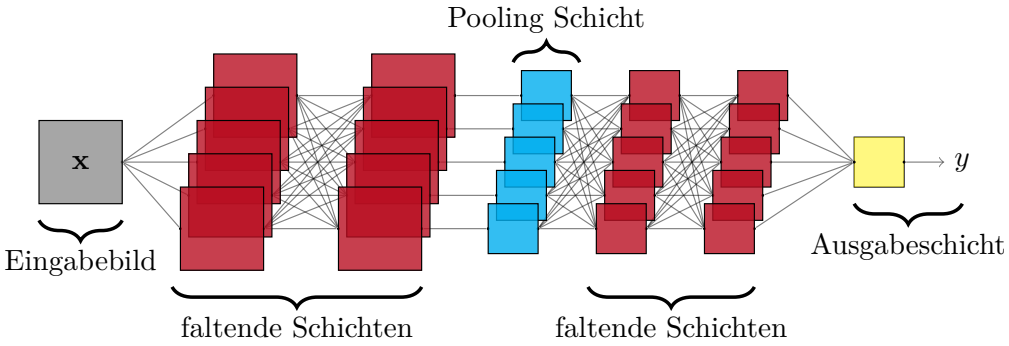


Abbildung 1.3.: Darstellung eines faltenden neuronalen Netzes.

gegebenen Definitionsbereiche der einzelnen Schichten ergeben sich dabei jeweils aus den Indexmengen der Wertebereiche der vorangegangen Schichten, wobei $I_0 = \{1, \dots, d_1\} \times \{1, \dots, d_2\}$ und $k_0 = 1$ gilt. In zwei der in dieser Arbeit verwendeten Netzwerkarchitekturen werden wir mehrere faltende neuronale Netze der Form (1.17) parallel berechnen und auf die Ausgaben ein vollverbundenes neuronales Netz der Form (1.7) anwenden.

Einen Bildklassifikator basierend auf faltenden neuronalen Netzen definieren wir dann als Plug-In Klassifikator (1.2) eines Kleinste-Quadrate-Schätzers der a-posteriori Wahrscheinlichkeit gemäß Gleichung (1.4) über einer Klasse $\mathcal{F}$ von faltenden neuronalen Netzen, wobei hier bezüglich der trainierbaren Gewichte (1.8), (1.9), (1.10), (1.11) oder (1.15) einer festen Netzwerkarchitektur minimiert wird. Wir nehmen dabei an, dass das Minimum in (1.4) existiert und von unserem Schätzer angenommen wird. Ist dies nicht der Fall, gelten die Resultate in dieser Arbeit auch für Schätzer der a-posteriori Wahrscheinlichkeit, deren empirisches L_2-Risiko nah genug am Infiumum dran ist. In Gleichung (1.4) minimieren wir bezüglich des *mittleren quadratischen* Fehlers. In praktischen Anwendungen der Bildklassifikation wird statt des mittleren quadratischen Fehlers meist die *Cross-Entropy-Verlustfunktion* oder die *Hinge-Verlustfunktion* verwendet. In der theoretischen Analyse solcher Klassifikatoren führt das allerdings dazu, dass weitere Annahmen an die Verteilungen von $(\mathbf{X}, Y)$ notwendig sind (siehe beispielsweise Kim et al. (2021), Kohler und Langer (2020) oder Liu et al. (2021)), weshalb wir hier dennoch den mittleren quadratischen Fehler verwenden. Resultate, die zeigen, dass bei einer Klasse von überparametrisierten neuronalen Netzen (d.h. die Anzahl der trainierbaren Gewichte übersteigt den Stichprobenumfang n) unter Verwendung von Gradientenabstiegsverfahren sogar das globale Minumum in (1.4) gefunden werden kann, finden sich beispielsweise in Zou et al. (2020), Allen-Zhu et al. (2019) und Du et al. (2019). Allerdings konnten Kohler und Krzyżak (2021) zeigen, dass überparametrisierte neuronale Netze, welche als Kleinste-Quadrate-Schätzer definiert sind, im Allgemeinen keine gute statistische Performanz aufweisen.

Ein Grund, weshalb neuronale Netze Schätzer in praktischen Anwendungen so erfolgreich sind, ist neben der Verfügbarkeit von effizienten Methoden zur näherungsweisen Lösung des Minimierungsproblems (1.4) auch die Approximationsfähigkeit von neuronalen Netzen gegenüber multivariaten Funktionen. Approximationsresultate für neuronale Netze mit mehreren verdeckten Schichten finden sich z.B. in Eldan und Shamir (2016), Mhaskar und Poggio (2016), Yarotsky und Zhevnerchuk (2020) und Kohler und Langer (2021). In den Arbeiten von Eldan und Shamir (2016) und Mhaskar und Poggio (2016) konnte speziell gezeigt werden, dass tiefe neuronale Netze bessere Approximationseigenschaften als neuronale Netze mit einer, bzw. zwei verdeckten Schichten, besitzen. Das Approximationsresultat von Kohler und Langer (2021), welches dem Resultat von Yarotsky und Zhevnerchuk (2020) ähnelt, werden wir in dieser Arbeit für die Approximation unserer Modelle zur Bildklassifikation durch faltende neuronale Netze verwenden. Approximationsresultate für faltende neuronale Netze erzielten beispielsweise Zhou (2020), Petersen und Voigtlaender (2020) und

Yarotsky (2022). Spezielle Netzwerkarchitekturen von faltenden neuronalen Netzen, die beispielsweise den Aspekt der Rotation von Objekten mit in die Architektur aufnehmen, finden sich in Delchevalerie et al. (2021), Dieleman et al. (2015) und Cohen und Welling (2016).

1.4. Bezug zur Regressionsschätzung

Im Folgenden geben wir eine kurze Einführung in die nichtparametrische Regression, da wir zum einen unser Klassifikationsproblem mithilfe eines Regressionsschätzers lösen werden und zum anderen viele wichtige Resultate, die eng mit unseren Resultaten verwandt sind, im Zusammenhang der Regressionsschätzung erzielt wurden. Für eine ausführliche Einführung in die Problemstellung der Regressionsschätzung siehe z.B. Györfi et al. (2002). Im Gegensatz zur Mustererkennung sind bei der nichtparametrischen Regression die unabhängigen identisch verteilten $\mathbb{R}^d \times \mathbb{R}$-wertigen Zufallsvariablen

$$(\mathbf{X}, Y), (\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)$$

gegeben, wobei $\mathbf{E}\{Y^2\} < \infty$ vorausgesetzt wird. Gesucht ist dann eine messbare Funktion $f : \mathbb{R}^d \to \mathbb{R}$, sodass $f(\mathbf{X})$ „möglichst nah“ am Wert von Y ist. Hierfür wird das L_2-Risiko

$$\mathbf{E}\{|f(\mathbf{X}) - Y|^2\}$$

betrachtet. Dieses wird durch die sogenannte *Regressionsfunktion*

$$m(\mathbf{x}) = \mathbf{E}\{Y|\mathbf{X} = \mathbf{x}\} \quad (\mathbf{x} \in \mathbb{R}^d)$$

minimiert, d.h. es gilt

$$\mathbf{E}\{|m(\mathbf{X}) - Y|^2\} = \min_{f:\mathbb{R}^d \to \mathbb{R}} \mathbf{E}\{|f(\mathbf{X}) - Y|^2\}$$

(vgl. Gleichung (1.1) in Györfi et al. (2002)). Da auch hier im Allgemeinen die Verteilung von $(\mathbf{X}, Y)$ unbekannt ist und damit auch die Regressionsfunktion, wird ausgehend von den Beobachtungen

$$\mathcal{D}_n = \{(\mathbf{X}_1, Y_1), \ldots, (\mathbf{X}_n, Y_n)\}$$

ein Schätzer der Regressionsfunktion

$$m_n(\cdot) = m_n(\cdot, \mathcal{D}_n) : \mathbb{R}^d \to \mathbb{R}$$

konstruiert, der ein möglichst kleines L_2-Risiko $\mathbf{E}\{|m_n(\mathbf{X}) - Y|^2|\mathcal{D}_n\}$ besitzen soll. Wegen

$$\mathbf{E}\{|m_n(\mathbf{X}) - Y|^2|\mathcal{D}_n\} = \int_{\mathbb{R}^d} |m_n(\mathbf{x}) - m(\mathbf{x})|^2 \mathbf{P}_\mathbf{X}(d\mathbf{x}) + \mathbf{E}\{|m(\mathbf{X}) - Y|^2\}$$

(siehe Gleichung (1.2) in Györfi et al. (2002)), reicht es, den L_2-Fehler

$$\int_{\mathbb{R}^d} |m_n(\mathbf{x}) - m(\mathbf{x})|^2 \mathbf{P}_\mathbf{X}(d\mathbf{x})$$

als Qualitätsmaß der Schätzung zu Rate zu ziehen. Stone (1977) konnte zeigen, dass unter der Verwendung eines Nächste-Nachbar Schätzers der Regressionsfunktion für beliebige Verteilungen von $(\mathbf{X}, Y)$ gilt, dass

$$\mathbf{E}\left\{\int |m_n(\mathbf{x}) - m(\mathbf{x})|^2 \, \mathbf{P}_\mathbf{X}(d\mathbf{x})\right\} \to 0 \quad (n \to \infty).$$

Da die a-posteriori Wahrscheinlichkeit η aus dem vorherigen Kapitel als Regressionsfunktion des Zufallsvektors $(\mathbf{X}, Y)$ aufgefasst werden kann, hat die Existenz von Schätzern mit der obigen Eigenschaft eine direkte Konsequenz für die Bildklassifikation: Wegen Ungleichung (1.5) wissen wir nun, dass eine Folge von Klassifikatoren $(f_n)_{n\in\mathbb{N}}$ existiert, sodass $\mathbf{E}\{L(f_n)\} - L(f^*)$ für beliebige Verteilungen von $(\mathbf{X}, Y)$ gegen 0 konvergiert. Eine Folge $(f_n)_{n\in\mathbb{N}}$ von Klassifikatoren, für die der Fehler $\mathbf{E}\{L(f_n)\} - L(f^*)$ gegen 0 konvergiert, wird *konsistent* genannt. Konvergiert $\mathbf{E}\{L(f_n)\} - L(f^*)$ für eine Folge $(f_n)_{n\in\mathbb{N}}$ von Klassifikatoren für alle Verteilungen $(\mathbf{X}, Y)$ gegen 0, so wird die Sequenz *universell konsistent* genannt (siehe z.B. Kapitel 1 in Devroye et al. (1996)). Dass die Existenz von universell konsistenten Klassifikatoren nicht zufriedenstellend ist, werden wir im nächsten Abschnitt besprechen. Wie Theorem 6.5 in Devroye et al. (1996) zeigt, ist die Mustererkennung „leichter“ als die Regressionschätzung. Genauer konnte hier gezeigt werden, dass für beliebige Verteilungen von $(\mathbf{X}, Y)$ und beliebige Folgen von Schätzern der a-posteriori Wahrscheinlichkeit $(\eta_n)_{n\in\mathbb{N}}$, für die der erwartete L_2-Fehler (1.6) gegen 0 konvergiert, gilt, dass

$$\lim_{n\to\infty} \frac{\mathbf{E}\{L(f_n)\} - L(f^*)}{\sqrt{\mathbf{E}\left\{\int |\eta_n(\mathbf{x}) - \eta(\mathbf{x})|^2 \, \mathbf{P}_{\mathbf{X}}(d\mathbf{x})\right\}}} = 0, \tag{1.18}$$

wobei $(f_n)_{n\in\mathbb{N}}$ die durch (1.2) definierte Folge von Plug-In Klassifikatoren bezeichnet. Wie wir im nächsten Abschnitt sehen werden, liefert uns Ungleichung (1.5) für einige Klassen von Verteilungen dennoch optimale Konvergenzraten für Plug-In Klassifikatoren, was deren Verwendung rechtfertigt.

1.5. Konvergenzgeschwindigkeit und Fluch der hohen Dimension

In diesem Abschnitt werden wir sehen, dass es nötig ist, die Klasse der Verteilungen von $(\mathbf{X}, Y)$ einzuschränken, um sinnvolle Aussagen über die statistische Performanz von Bildklassifikatoren treffen zu können. Wenn nicht explizit erwähnt, beziehen wir uns in diesem Abschnitt auf den allgemeinen Fall der Mustererkennung, bei dem $\mathbf{X}$ eine $\mathbb{R}^d$-wertige Zufallsvariable darstellt.

Cover (1968) konnte zeigen, dass für alle Folgen $(f_n)_{n\in\mathbb{N}}$ von Klassifikatoren, für beliebig kleine algebraische Raten (d.h. Raten der Form $a_n = n^{-\beta}$ für $0 < \beta < 1$), eine Verteilung von $(\mathbf{X}, Y)$ existiert, sodass

$$\mathbf{E}\{L(f_n)\} - L(f^*) \geq a_n$$

für unendlich viele n gilt. Devroye (1982) konnte dieses Resultat auf Raten $(a_n)_{n\in\mathbb{N}}$ verallgemeinern, die beliebig langsam gegen 0 konvergieren. Ein noch strengeres Resultat gelang Devroye et al. (1996):

Theorem 1.1 (Theorem 7.2 in Devroye et al. (1996))**.** *Es sei $(a_n)_{n\in\mathbb{N}}$ eine positive reelle Folge, die gegen Null konvergiert mit $1/16 \geq a_1 \geq a_2 \geq \ldots$. Dann existiert für jede Folge $(f_n)_{n\in\mathbb{N}}$ von Klassifikatoren eine Verteilung von $(\mathbf{X}, Y)$ mit $L(f^*) = 0$, sodass*

$$\mathbf{E}\{L(f_n)\} \geq a_n$$

für alle n.

In Devroye et al. (1996) wird erwähnt, dass Theorem 1.1 auch gültig ist, wenn $\mathbf{X}$ eine stetige Gleichverteilung auf $[0, 1]$ besitzt und die a-posteriori Wahrscheinlichkeit η unendlich oft stetig differenzierbar ist, und in Devroye (1982) wird erwähnt, dass deren Resultat auch gültig ist, wenn $\mathbf{X}$ eine Zufallsvariable mit fester beliebiger Dichtefunktion ist. Daher gelten diese Negativresultate zum einen auch für unseren Fall der Bildklassifikation, bei dem $\mathbf{X}$ Werte in $[0, 1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}}$ annimmt, und zum anderen bedeutet das, dass es nicht ausreicht, die Verteilungen von $\mathbf{X}$ einzuschränken, sondern wir entsprechend strenge Annahmen an die a-posteriori Wahrscheinlichkeit formulieren müssen, um nichttriviale Konvergenzraten für den Fehler $\mathbf{E}\{L(f_n)\} - L(f^*)$ erzielen zu können.

Eine Möglichkeit ist, die Bauart der a-posteriori Wahrscheinlichkeit soweit vorzugeben, sodass diese nur noch von endlich vielen Parametern abhängt. Man spricht in diesem Zusammenhang dann von *parametrischer Regressionsschätzung*. Es genügt dann, die entsprechenden Parameter zu schätzen. Der Nachteil dieser Methode ist, dass der Schätzer nur so gut sein kann, wie die beste Funktion dieser Bauart. Lässt sich der zu schätzende funktionale Zusammenhang nicht gut durch eine Funktion dieser Bauart approximieren, entsteht daher ein entsprechender Fehler bei unserer Schätzung (siehe Kapitel 1.5 in Györfi et al. (2002)). Aus diesem Grund untersuchen wir die Bildklassifikation im Kontext der *nichtparametrischen Regression*, d.h. wir versuchen möglichst allgemeine Annahmen an die a-posteriori Wahrscheinlichkeit zu treffen, die für entsprechende Situationen plausibel erscheinen. In der nichtparametrischen Regression werden hierfür Glattheitsannahmen sowie manchmal auch Strukturannahmen an die Regressionsfunktion gestellt.

Um Glattheitsannahmen an die a-posteriori Wahrscheinlichkeit formulieren zu können, führen wir den Begriff der (p, C)-Glattheit ein, welcher eine Verallgemeinerung der Hölderstetigkeit auf höhere Ableitungen darstellt.

Definition 1. Sei $p = k + \beta$ für ein $k \in \mathbb{N}_0$ und $0 < \beta \leq 1$ und sei $C > 0$. Eine Funktion $f : \mathbb{R}^d \to \mathbb{R}$ heißt *(p,C)-glatt*, falls für jedes $\alpha = (\alpha_1, ..., \alpha_d)$ mit $\alpha_i \in \mathbb{N}_0$ und $\sum_{j=1}^{d} \alpha_j = k$ die partielle Ableitung

$$\frac{\partial^k f}{\partial x_1^{\alpha_1} ... \partial x_d^{\alpha_d}}$$

existiert und für alle $x, z \in \mathbb{R}^d$ gilt

$$\left| \frac{\partial^k f}{\partial x_1^{\alpha_1} ... \partial x_d^{\alpha_d}}(x) - \frac{\partial^k f}{\partial x_1^{\alpha_1} ... \partial x_d^{\alpha_d}}(z) \right| \leq C \cdot \|x - z\|^{\beta},$$

für eine Vektornorm $\|\cdot\| : \mathbb{R}^d \to \mathbb{R}_0^+$.

Stone (1982) konnte zeigen, dass unter der Annahme von (p, C)-glatten Regressionsfunktionen,

$$n^{-\frac{2p}{2p+d}}$$

die optimale Konvergenzrate für Regressionsschätzer darstellt. Mit dem Beweis der Existenz von Schätzern, die diese Rate erzielen, konnte er wegen Ungleichung (1.5) damit insbesondere auch zeigen, dass unter der Annahme einer (p, C)-glatten a-posteriori Wahrscheinlichkeit Plug-In Klassifikatoren existieren, für die der Fehler $\mathbf{E}\{L(f_n)\} - L(f^*)$ mit der Rate

$$n^{-\frac{p}{2p+d}} \tag{1.19}$$

gegen 0 konvergiert. Yang (1999) konnte zeigen, dass eine bessere Konvergenzrate für $\mathbf{E}\{L(f_n)\} - L(f^*)$ unter dieser Annahme nicht möglich ist und die obige Rate damit eine optimale Konvergenzrate für die Mustererkennung darstellt. Ein alternativer Beweis dafür findet sich in Antos (1999). Hier wird ersichtlich, dass das Resultat auch unter der Annahme gültig ist, dass die a-posteriori Wahrscheinlichkeit η einen kompakten Support besitzt, wie es in der Bildklassifikation der Fall ist, da die Zufallsvariable $\mathbf{X}$ nur Werte im Kompaktum $[0, 1]^{\{1,...,d_1\} \times \{1,...,d_2\}}$ annimmt. Zwar zeigt uns das nun, dass es trotz Gleichung (1.18) Sinn macht, Plug-In Klassifikatoren zu verwenden und den Fehler $\mathbf{E}\{L(f_n)\} - L(f^*)$ mithilfe von Ungleichung (1.5) abzuschätzen, allerdings ergibt sich aus der obigen Rate (1.19) ein anderes Problem: Ist die Dimension d der Daten groß, konvergiert der Fehler $\mathbf{E}\{L(f_n)\} - L(f^*)$ möglicherweise nur sehr langsam gegen 0. Dieses Phänomen wird auch *Fluch der hohen Dimension* genannt. Da in der Bildklassifikation üblicherweise große Bilddimensionen d_1 und d_2 vorliegen, sind Klassifikatoren mit der obigen Rate für uns nicht zufriedenstellend. Unter strengeren Bedingungen an die a-posteriori Wahrscheinlichkeit, z.B. der sogenannten *margin condition*, können schnellere

Konvergenzraten erzielt werden (siehe z.B. Mammen und Tsybakov (1999), Tsybakov und van de Geer (2005) und Audibert und Tsybakov (2007)). Es ist jedoch unklar, inwieweit diese Annahmen für welche Art von Bildklassifikationsproblemen realistisch sind. In dieser Arbeit werden wir daher im nächsten Abschnitt verschiedene Modelle zur Bildklassifikation einführen, die möglichst allgemeine und für Bildklassifikationsprobleme plausible Annahmen bereitstellen, unter denen eine Dimensionsreduktion möglich sein wird. Bevor wir diese Annahmen formulieren, geben wir einen Überblick über verwandte Resultate aus der Literatur.

In einer Reihe von Arbeiten gelang eine Dimensionsreduktion durch Strukturannahmen an die Regressionsfunktion im Kontext der Regressionsschätzung. Stone (1985) untersuchte sogenannte *additive Modelle*. Hier wird angenommen, dass sich die Regressionsfunktion als Summe von univariaten (p, C)-glatten Funktionen $m_i : \mathbb{R} \to \mathbb{R}$ $(i = 1, \dots, d)$ auffassen lässt:

$$m(\mathbf{x}) = m_1(x^{(1)}) + \cdots + m_d(x^{(d)}) \quad (\mathbf{x} = (x^{(1)}, \dots, x^{(d)}) \in \mathbb{R}).$$

Unter dieser Annahme zeigte Stone eine optimale Konvergenzrate von

$$n^{-\frac{2p}{2p+1}},$$

womit der Fluch der hohen Dimension umgangen werden kann. Stone (1994) konnte dieses Modell zu einem sogenannten *Interaktionsmodell* verallgemeinern. Hier hat die Regressionsfunktion die Form

$$m(\mathbf{x}) = \sum_{I \subset \{1,\dots,d\}, |I| = d^*} m_I(\mathbf{x}_I) \quad (\mathbf{x} \in \mathbb{R}^d)$$

mit (p, C)-glatten Funktionen $m_I : \mathbb{R}^{|I|} \to \mathbb{R}$. Unter dieser Annahme konnte er eine optimale Rate von

$$n^{-\frac{2p}{2p+d^*}}$$

herleiten. Weitere Annahmen, bei denen sich die Regressionsfunktion aus Funktionen niedrigerer Dimension zusammensetzt und eine Dimensionsreduktion gelang, sind das *Single-Index-Modell*

$$m(\mathbf{x}) = g(\mathbf{a}^T \mathbf{x}) \quad (\mathbf{x} \in \mathbb{R}^d)$$

für ein $\mathbf{a} \in \mathbb{R}^d$ und eine (p, C)-glatte Funktion $g : \mathbb{R} \to \mathbb{R}$ (siehe Härdle und Stoker (1989), Härdle et al. (1993), Yu und Ruppert (2002), Kong und Xia (2007) und Lepski und Serdyukova (2014)) sowie das *Projection-Pursuit-Modell*

$$m(\mathbf{x}) = \sum_{l=1}^{r} g_l(\mathbf{a}_l^T \mathbf{x}) \quad (\mathbf{x} \in \mathbb{R}^d)$$

für ein $r \in \mathbb{N}$, $\mathbf{a}_l \in \mathbb{R}^d$ und (p, C)-glatte Funktionen $g_l : \mathbb{R} \to \mathbb{R}$ $(l = 1, \dots, r)$ (siehe Friedman und Stuetzle (1981) und Huber (1985)). Bis auf einen logarithmischen Term konnte in Theorem 22.2 in Györfi et al. (2002) eine optimale Rate von $n^{-2p/(2p+1)}$ für Kleinste-Quadrate-Schätzer, basierend auf stückweisen Polynomen, für diese beiden Modelle gezeigt werden. Eine Verallgemeinerung der Modelle lieferten Horowitz und Mammen (2007), bei der die Regressionsfunktion die Form

$$m(\mathbf{x}) = g\left(\sum_{l_1=1}^{L_1} g_{l_1}\left(\sum_{l_2=1}^{L_2} g_{l_1,l_2}\left(\cdots \sum_{l_r=1}^{L_r} g_{l_1,\dots,l_r}(x^{l_1,\dots,l_r})\right)\right)\right) \quad (\mathbf{x} \in \mathbb{R}^d)$$

hat, wobei $g, g_{l_1}, \dots, g_{l_1,\dots,l_r}$ univariate (p, C)-glatte Funktionen sind und $x^{l_1,\dots,l_r}$ einzelne Komponenten von $\mathbf{x} \in \mathbb{R}^d$ bezeichnen, die sich für unterschiedliche Indizes $l_1, \dots, l_r$ nicht zwingend unterscheiden müssen.

Sie konnten für Kleinste-Quadrate-Schätzer mit Strafterm basierend auf einem Smoothing-Splineschätzer ebenfalls eine Rate von $n^{-2p/(2p+1)}$ herleiten.

Für Kleinste-Quadrate-Schätzer basierend auf neuronalen Netzen mit einer verdeckten Schicht konnten im Kontext der Regressionsschätzung Konvergenzratenresultate von Barron (1991), Barron (1993), Barron (1994) und McCaffrey und Ronald Gallant (1994) nachgewiesen werden. Barron (1994) konnte bis auf einen logarithmischen Term eine von der Dimension unabhängige Rate von $n^{-\frac{1}{2}}$ zeigen, für den Fall, dass die Fourier-Transformierte der Regressionsfunktion ein endliches absolutes erstes Moment besitzt. McCaffrey und Ronald Gallant (1994) leiteten eine Rate von $n^{-\frac{2p}{2p+d+5}+\epsilon}$ für (p, C)-glatte Regressionsfunktionen unter der Verwendung einer speziellen Kosinusfunktion als Aktivierungsfunktion her.

Weitere Modelle für die Regressionsfunktion, welche das obige Modell von Horowitz und Mammen (2007) und die vorherigen Modelle erweitern, formulierten Kohler und Krzyżak (2017), Schmidt-Hieber (2020) und Kohler und Langer (2021). Wir führen hier das sogenannte *hierarchische Kompositionsmodell* von Kohler und Langer (2021) an, da es die jeweiligen Modelle in Kohler und Krzyżak (2017) und Schmidt-Hieber (2020) und alle bisher genannten Modelle, wie in Kohler und Langer (2021) erwähnt wird, verallgemeinert.

Definition 2 (Hierarchisches Kompositionsmodell)**.** Es sei $d \in \mathbb{N}$, $l \in \mathbb{N}_0$, $m : \mathbb{R}^d \to \mathbb{R}$ und $\mathcal{P}$ eine Teilmenge von $(0, \infty) \times \mathbb{N}$.

a) Die Funktion m genügt einem *hierarchischen Kompositionsmodell vom Level* 0 *mit Ordnungs- und Glattheitsbedingung* $\mathcal{P}$*,* falls ein $K \in \{1, ..., d\}$ existiert, sodass

$$m(\mathbf{x}) = x^{(K)} \quad \text{für alle } \mathbf{x} = (x^{(1)}, ..., x^{(d)})^T \in \mathbb{R}^d.$$

b) Die Funktion m genügt einem *hierarchischen Kompositionsmodell vom Level* $l + 1$ mit Ordnungs- und Glattheitsbedingung $\mathcal{P}$, falls $(p, K) \in \mathcal{P}$, $C > 0$, eine Funktion $g : \mathbb{R}^K \to \mathbb{R}$ und Funktionen $f_1, ..., f_K : \mathbb{R}^d \to \mathbb{R}$ existieren, sodass die Funktion g (p, C)-glatt ist und die Funktionen $f_1, ..., f_K$ einem *hierarchischen Kompositionsmodell vom Level* l mit Ordnungs- und Glattheitsbedingung $\mathcal{P}$ genügen und es gilt, dass

$$m(\mathbf{x}) = g\left(f_1(\mathbf{x}), ..., f_K(\mathbf{x})\right) \quad \text{für alle } \mathbf{x} \in \mathbb{R}^d.$$

Wie in Kohler und Krzyżak (2017) und Kohler und Langer (2020) erklärt wird, findet ein solches Modell beispielsweise Anwendung bei der Schätzung von Eingabe- und Ausgabebeziehungen von komplexen technischen Systemen. Das obige Modell ist nur eine leichte Verallgemeinerung des Modells in Schmidt-Hieber (2020), bei dem keine verschiedenen Glattheiten und Dimensionen innerhalb eines Levels erlaubt sind. Im sogenannten *verallgemeinerten hierarchischen Interaktionsmodell* von Kohler und Krzyżak (2017) geht der Komposition stets eine Summenbildung voraus. In allen drei Artikeln gelang eine Dimensionsreduktion für Kleinste-Quadrate-Schätzer basierend auf neuronalen Netzen unter der Annahme der entsprechenden Modelle für die Regressionsfunktion. Für das hierarchische Kompositionsmodell aus Definition 2 konnte bis auf einen logarithmischen Faktor die Rate

$$\max_{(p,K)\in\mathcal{P}} n^{-\frac{2p}{2p+K}}$$

für Glattheits- und Ordnungsbedingungen $\mathcal{P} \subseteq [1, \infty) \times \mathbb{N}$ nachgewiesen werden. In Kohler und Krzyżak (2017) wurden Hölder-stetige Funktionen ($p \in (0, 1]$) betrachtet und eine Dimensionsreduktion gelang auch für einen Kleinste-Quadrate-Schätzer basierend auf polynomiellen Splines. Bauer und Kohler (2019) konnten eine Dimensionsreduktion für das verallgemeinerte hierarchische Interaktionsmodell auch für den Fall $p > 1$ für geeignet gewählte differenzierbare Aktivierungsfunktionen herleiten. In Kohler und Langer (2021) und Schmidt-Hieber (2020) wurde die ReLU Aktivierungsfunktion verwendet, wobei die neuronalen Netze in

Kohler und Langer (2021) vollverbunden sind (d.h. sie haben die Form (1.7)) und die neuronalen Netze in Schmidt-Hieber (2020) unvollständig verbunden sind mit einer nicht genauer spezifizierten Architektur.

Im Fall einer Regressionsfunktion mit geringer lokaler Dimension, d.h. diese hängt auf lokalen Bereichen nur von $d^* \leq d$ Variablen ab, konnten Kohler et al. (2022) eine Dimensionsreduktion für unvollständig verbundene Neuronale-Netze-Schätzer zeigen. Die lokalen Bereiche wurden hier als d-dimensionale Polytope definiert, auf denen die Regressionsfunktion jeweils (p, C)-glatt ist und nur von $d^* \leq d$ Variablen abhängt. Für Regressionsfunktionen, die nur auf Partitionen glatt sind, erzielten Imaizumi und Fukamizu (2019) bis auf einen logarithmischen Faktor optimale Konvergenzraten für Neuronale-Netze-Schätzer. Suzuki und Nitanda (2021) konnten eine Dimensionsreduktion in verallgemeinerten Besov-Räumen mit Neuronale-Netze-Schätzern zeigen.

Optimale Konvergenzratenresultate für Regressionschätzer basierend auf faltenden neuronalen Netzen gelangen Oono und Suzuki (2019). Hier wurde eine spezielle Netzwerkarchitektur der sogenannten residualen faltenden neuronalen Netze verwendet, bei denen Verbindungen über Schichten hinweg bestehen. Mit ihrer Analyse konnten sie jedoch keine Situationen aufzeigen, in denen die residualen faltenden neuronalen Netze standardmäßigen vorwärtsgerichteten neuronalen Netzen überlegen sind.

Im Kontext der Klassifikation zeigten Kim et al. (2021), Bos und Schmidt-Hieber (2022) und Hu et al. (2020) Konvergenzratenresultate unter der Verwendung von standardmäßigen vorwärtsgerichteten neuronalen Netzen. Liu et al. (2021) verwendeten dagegen residuale faltende neuronale Netze und konnten eine Dimensionsreduktion für den Fall zeigen, dass die Daten eine niedrig dimensionale geometrische Struktur vorweisen. Lin und Zhang (2019) erzielten Schranken zur statistischen Performanz von faltenden neuronalen Netzen in einem Setting der Klassifikation. Speziell für die binäre Bildklassifikation erzielten Kohler und Langer (2020) und Langer und Schmidt-Hieber (2022) Konvergenzratenresultate, bei denen der Fluch der hohen Dimension unter Verwendung von faltenden neuronalen Netzen umgangen werden konnte. Im Paper von Kohler und Langer (2020) wird hierfür das im nächsten Kapitel eingeführte hierarchische Max-Pooling Modell für die a-posteriori Wahrscheinlichkeit verwendet (siehe Definition 3 in Abschnitt 2.1). Der Klassifikator wird hier an die Trainingsdaten angepasst, indem bezüglich der Cross-Entropy-Verlustfunktion minimiert wird, was in der theoretischen Analyse die zusätzliche Annahme erfordert, dass die a-posteriori Wahrscheinlichkeit mit hoher Wahrscheinlichkeit nah an der 0 oder 1 liegt. In Langer und Schmidt-Hieber (2022) wird der Ansatz verfolgt, dass sich Bilder der beiden Klassen jeweils durch Deformationen einer festen reellwertigen Funktion auf $\mathbb{R}^2$ erzeugen lassen. Der Nachteil dieser Methode ist, dass sich Objekte einer Klasse nur durch Helligkeit, Skalierung und Translation unterscheiden, was in den meisten pratkischen Anwendungen der Bildklassifikation nicht der Fall ist.

1.6. Konvergenzverhalten von Bildklassifikatoren basierend auf faltenden neuronalen Netzen

Im vorangegangenen Abschnitt haben wir gesehen, dass es im Kontext der Regressionsschätzung möglich ist, durch Struktur- und Glattheitsannahmen an die Regressionsfunktion, den Fluch der hohen Dimension zu umgehen. Dabei wurde beispielsweise mit dem hierarchischen Kompositionsmodell (siehe Definition 2 im vorangegangenen Abschnitt) ein für Anwendungen plausibles Modell eingeführt, das Situationen aufzeigt, in denen Neuronale-Netze-Schätzer ein aus theoretischer Sicht gutes Konvergenzverhalten besitzen. Die entsprechenden Resultate lassen dadurch Rückschlüsse auf sinnvolle Netzwerkarchitekturen zu, beziehungsweise erklären den Erfolg von in Anwendungen eingesetzten Verfahren. Bisher wurde dieser Ansatz nicht auf die Bildklassifikation übertragen, was bedeutet, dass noch keine entsprechenden Struktur- und Glattheitsannahmen an die a-posteriori Wahrscheinlichkeit formuliert wurden, die realistisch für Anwendungen der Bildklassifikation erscheinen, um anschließend eine Dimensionsreduktion für Klassifikatoren

basierend auf faltenden neuronalen Netzen herzuleiten. Da dieser Ansatz auch hier verspricht, den Erfolg, der in praktischen Anwendungen verwendeten Verfahren, theoretisch besser zu erklären und Hinweise auf sinnvolle Netzwerkarchitekturen zu geben, ergibt sich eine Forschungslücke. Das Ziel dieser Arbeit ist es daher, einerseits sinnvolle Struktur- und Glattheitsannahmen für die a-posteriori Wahrscheinlichkeit (1.1) zu formulieren und eine von der Bilddimension unabhängige Konvergenzrate für den Fehler $\mathbf{E}\{L(f_n) - L(f^*)\}$ herzuleiten. Dabei soll der Klassifikator f_n auf faltenden neuronalen Netzen basieren und die Annahmen für die a-posteriori Wahrscheinlichkeit sollen für Anwendungen der Bildklassifikation natürlich erscheinen. Der erste Beitrag dieser Arbeit besteht deshalb darin, geeignete Annahmen an die Verteilung von $(\mathbf{X}, Y)$ zu formulieren. In Kapitel 2 werden zu diesem Zweck drei Modelle für die a-posteriori Wahrscheinlichkeit eingeführt, welche unterschiedliche Aspekte von Problemen der Bildklassifikation berücksichtigen und sich aus Struktur- und Glattheitsannahmen zusammensetzen. Die drei Modelle sind dabei von dem Vorgehen eines Menschen beim Klassifizieren von Bildern abgeleitet. Der zweite Beitrag besteht darin, für jedes dieser drei Modelle Bildklassifikatoren einzuführen, die auf faltenden neuronalen Netzen basieren und eine von der Bilddimension unabhängige Konvergenzrate für den Fehler $\mathbf{E}\{L(f_n)\} - L(f^*)$ zu zeigen.

Zunächst wird das *verallgemeinerte hierarchische Max-Pooling Modell vom Level $l \in \mathbb{N}$ und der Ordnung $d^* \in \mathbb{N}$ mit Glattheitsbedingungen $p_1, p_2 \in [1, \infty)$* eingeführt. In diesem Modell wird angenommen, dass eine (p_2, C)-glatte Funktion $g : \mathbb{R}^{d^*} \to [0,1]$ für ein $d^* \in \mathbb{N}$ und Funktionen $m_1, \ldots, m_{d^*} : [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}} \to [0,1]$ existieren, sodass für die a-posteriori Wahrscheinlichkeit η (1.1) gilt:

$$\eta(\mathbf{x}) = g(m_1(\mathbf{x}), \ldots, m_{d^*}(\mathbf{x})) \quad \left(\mathbf{x} \in [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}}\right).$$

Die Interpretation ist hierbei, dass die Wahrscheinlichkeit, ob ein Bild zu der Klasse 1 gehört, durch Anwendung der Funktion g auf die Wahrscheinlichkeiten der Existenz von $d^* \in \mathbb{N}$ Objekten $(m_1, \ldots, m_{d^*})$ berechnet wird. Die Wahrscheinlichkeit der Existenz eines einzelnen Objekts (m_j für ein $j \in \{1, \ldots, d^*\}$) wird berechnet, indem für jeden Teilbereich des Bildes einer festen Größe die Wahrscheinlichkeit berechnet wird, dass dieser Teilbereich das Objekt enthält. Die Wahrscheinlichkeit der Existenz des Objekts im gesamten Bildbereich entspricht im Modell dann gerade dem Maximum all dieser Wahrscheinlichkeiten. Für eine mathematische Formulierung dieser Annahme benötigen wir die beiden folgenden Notationen: Für eine Teilmenge $M \subseteq \mathbb{R}^d$ und $\mathbf{x} \in \mathbb{R}^d$ definieren wir

$$\mathbf{x} + M = \{\mathbf{x} + \mathbf{z} \,:\, \mathbf{z} \in M\}$$

und für $I \subseteq \{1, \ldots, d_1\} \times \{1, \ldots, d_2\}$ und $\mathbf{x} = (x_i)_{i \in \{1,\ldots,d_1\}\times\{1,\ldots,d_2\}} \in [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}}$ verwenden wir die Schreibweise

$$\mathbf{x}_I = (x_i)_{i \in I}.$$

Es sei $I \subseteq \{0, \ldots, d_1 - 1\} \times \{0, \ldots, d_2 - 1\}$ eine Indexmenge, welche die betrachteten Teilbereiche des gesamten Bildbereichs aus der obigen Annahme festlegt. Wir nehmen dann an, dass für alle $j \in \{1, \ldots, d^*\}$ eine Funktion $f : [0,1]^{(1,1)+I} \to [0,1]$ existiert, sodass

$$m_j(\mathbf{x}) = \max_{(i_2,j_2)\in\mathbb{Z}^2 \,:\, (i_2,j_2)+I \subseteq \{1,\ldots,d_1\}\times\{1,\ldots,d_2\}} f\left(\mathbf{x}_{(i_2,j_2)+I}\right) \quad \left(\mathbf{x} \in [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}}\right)$$

(die Funktion f ist hierbei (möglicherweise) verschieden für verschiedene $j \in \{1, \ldots, d^*\}$). Außerdem wird angenommen, dass sich die Wahrscheinlichkeit $f(\mathbf{x}_{(i,j)+I})$, ob ein gegebener Teilbereich $\mathbf{x}_{(i,j)+I}$ ein bestimmtes Objekt enthält, hierarchisch berechnen lässt, indem die Entscheidungen von immer kleineren Teilbereichen zu Entscheidungen größerer Teilbereiche kombiniert werden. Hierbei werden jeweils vier Entscheidungen benachbarter Bereiche durch (p_1, C)-glatte reellwertige Funktionen

$$g_{k,s} : \mathbb{R}^4 \to [0,1] \quad (k = 1, \ldots, l, s = 1, \ldots, 4^{l-k})$$

zu Entscheidungen des zusammengesetzten Bereichs kombiniert. Für die mathematische Modellierung sei die obige Indexmenge gegeben durch $I = \{0, \dots, 2^l - 1\} \times \{0, \dots, 2^l - 1\}$ für ein $l \in \mathbb{N}$ mit $2^l \leq \min\{d_1, d_2\}$. Wir setzen $f = f_{l,1}$ und definieren die Funktion $f_{l,1}$ rekursiv durch die Funktionen $f_{k,s} : [0,1]^{\{1,\dots,2^k\}\times\{1,\dots,2^k\}} \to \mathbb{R}$ gemäß

$$\begin{aligned} f_{k,s}(\mathbf{x}) = g_{k,s}\big(&f_{k-1,4\cdot(s-1)+1}(\mathbf{x}_{\{1,\dots,2^{k-1}\}\times\{1,\dots,2^{k-1}\}}), f_{k-1,4\cdot(s-1)+2}(\mathbf{x}_{\{2^{k-1}+1,\dots,2^k\}\times\{1,\dots,2^{k-1}\}}), \\ &f_{k-1,4\cdot(s-1)+3}(\mathbf{x}_{\{1,\dots,2^{k-1}\}\times\{2^{k-1}+1,\dots,2^k\}}), f_{k-1,4\cdot s}(\mathbf{x}_{\{2^{k-1}+1,\dots,2^k\}\times\{2^{k-1}+1,\dots,2^k\}})\big) \end{aligned} \tag{1.20}$$

für $\mathbf{x} \in [0,1]^{\{1,\dots,2^k\}\times\{1,\dots,2^k\}}$, $k = 2, \dots, l$, $s = 1, \dots, 4^{l-k}$ und

$$f_{1,s}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}) = g_{1,s}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}) \quad (x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2} \in [0,1])$$

für $s = 1, \dots, 4^{l-1}$. Eine detaillierte Einführung und Motivation des verallgemeinerten hierarchischen Max-Pooling Modells befindet sich in Abschnitt 2.1.

Um die a-posteriori Wahrscheinlichkeit zu schätzen, werden dann tiefe faltende neuronale Netze der Form

$$f(\mathbf{x}) = g_{net}(f_1(\mathbf{x}), \dots, f_{d^*}(\mathbf{x})) \quad \big(\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}\big) \tag{1.21}$$

verwendet, wobei die Funktionen $f_1, \dots, f_{d^*} : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}$ jeweils faltenden neuronalen Netzen mit $L \in \mathbb{N}$ faltenden Schichten und k Kanälen in jeder Schicht der Form

$$f_{out}^{(\mathbf{A})} \circ o_{(k,k),M_L,1,\mathbf{w}_L} \circ \cdots \circ o_{(k,k),M_2,1,\mathbf{w}_2} \circ o_{(1,k),M_1,1,\mathbf{w}_1}(\mathbf{x}) \quad \big(\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}\big)$$

entsprechen. Die Funktion $f_{out}^{(\mathbf{A})}$ ist dabei eine gemäß (1.16) definierte Ausgabeschicht sowie $o_{(k_r',k),M_r,1,\mathbf{w}_r}$ gemäß (1.12) definierte faltende Schichten und die Funktion $g_{net} : \mathbb{R}^{d^*} \to \mathbb{R}$ entspricht einem vollverbundenen neuronalen Netz der Form (1.7), welches $L_{net} \in \mathbb{N}$ verdeckte Schichten und $r_{net} \in \mathbb{N}$ Neuronen in jeder Schicht besitzt. Der entsprechende Bildklassifikator $f_n(\cdot, \mathcal{D}_n) : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \{0,1\}$ ist dann als Plug-In Klassifikator (1.2) des Kleinste-Quadrate-Schätzers (1.4) über der durch die obige Form (1.21) gegebenen Netzwerkarchitektur definiert (der genaue Schätzer wird in Abschnitt 3.1 vollständig definiert). Das folgende Resultat beschreibt dann das Konvergenzverhalten des Bildklassifikators f_n im verallgemeinerten hierarchischen Max-Pooling Modell.

Resultat I. Es sei $n > 1$. Die a-posteriori Wahrscheinlichkeit $\eta(\mathbf{x}) = \mathbf{P}\{Y = 1 | \mathbf{X} = \mathbf{x}\}$ genüge einem verallgemeinerten hierarchischen Max-Pooling Modell der Ordnung d^* und Level l mit Glattheitsbedingungen $p_1, p_2 \in [1, \infty)$. Weiterhin sei

$$L_n = \max\left\{ n^{\frac{4}{2\cdot(2\cdot p_1+4)}}, n^{\frac{d^*}{2\cdot(2\cdot p_2+d^*)}} \right\}.$$

Die Netzwerkarchitektur des faltenden neuronalen Netzes $f : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}$ des Klassifikators $f_n(\cdot, \mathcal{D}_n)$ aus Gleichung (1.21) sei so gewählt, dass sie $L = \lceil c_1 \cdot L_n \rceil$ faltende und $L_{net} = \lceil c_2 \cdot L_n \rceil$ vollverbundene Schichten besitzt, wobei c_1 und c_2 sowie die sonstigen Netzwerkparameter genügend großen Konstanten entsprechen, die möglicherweise von den Modellparametern der a-posteriori Wahrscheinlichkeit abhängen. Dann gilt

$$\mathbf{E}\{L(f_n)\} - L(f^*) \leq c_3 \cdot \sqrt{\log(d_1 \cdot d_2)} \cdot (\log n)^2 \cdot \max\left\{ n^{-\frac{p_1}{2\cdot p_1+4}}, n^{-\frac{p_2}{2\cdot p_2+d^*}} \right\} \tag{1.22}$$

für eine Konstante $c_3 > 0$, die nicht von d_1, d_2 und n abhängt.

Für eine präzise Formulierung des obigen Resultats, bei der die Parameter der Netzwerkarchitektur sowie die Annahmen genauer spezifiziert werden, siehe Theorem 3.1. Da die Rate

$$\max\left\{n^{-\frac{p_1}{2\cdot p_1+4}}, n^{-\frac{p_2}{2\cdot p_2+d^*}}\right\}$$

in (1.22) nicht von den Bilddimensionen d_1 und d_2 abhängt, zeigt dies, dass es unter geeigneten Annahmen an die a-posteriori Wahrscheinlichkeit möglich ist, den Fluch der hohen Dimension durch faltende neuronale Netze in der Bildklassifikation zu umgehen. Außerdem liefert das Resultat theoretische Anhaltspunkte für die Wahl einer geeigneten Netzwerkarchitektur.

Die in der Praxis erfolgreichen Netzwerkarchitekturen besitzen neben faltenden Schichten meist auch lokale Pooling Schichten (vgl. beispielsweise Krizhevsky et al. (2012) oder Simonyan und Zisserman (2015)). Mit dem oben beschriebenen hierarchischen Max-Pooling Modell ist es nicht möglich, Situationen aufzuzeigen, in denen die Verwendung von lokalen Pooling Schichten aus theoretischer Sicht sinnvoll ist. Dies gelingt mit dem zweiten statistischen Modell zur Bildklassifikation, dem *hierarchischen Max-Pooling Modell vom Level $l \in \mathbb{N}$, lokalem Max-Pooling Parameter $(n_1, \dots, n_{l-1}) \in \mathbb{N}^{l-1}$ und Glattheitsbedingung $p \in [1, \infty)$*, welches das hierarchische Max-Pooling Modell dahingehend erweitert, dass dieses noch realistischer für einige Anwendungen der Bildklassifikation ist. Im Gegensatz zum verallgemeinerten hierarchischen Max-Pooling Modell ist in dem zweiten Modell nur die Existenz eines einzelnen Objekts entscheidend, weswegen hier nur (p, C)-glatte Funktionen $g_{k,s} : \mathbb{R}^4 \to [0, 1]$ benötigt werden, welche die Entscheidungen benachbarter Teilbereiche zu Entscheidungen des zusammengesetzten Teilbereichs kombinieren (vgl. Gleichung (1.20)). Die hierarchische Tiefe wird hier wie im vorherigen Modell durch den Wert von l festgelegt. In dem neuen Modell ist zusätzlich enthalten, dass die Abstände von Teilbereichen, welche im hierarchischen Modell zu größeren Teilbereichen zusammengefasst werden, einen variablen relativen Abstand zueinander haben und sich gegebenenfalls auch zu einem gewissen Teil überlappen können. Dies hat den Hintergrund, dass bei einigen Bildklassifikationsproblemen bestimmte Merkmale von Objekten nur einen ungefähren relativen Abstand zueinander haben müssen. Beispielsweise müssen bei der Gesichtserkennung die Merkmale „Nase“, „Augen“ und „Mund“ nur einen ungefähren relativen Abstand zueinander haben, damit ein Gesicht erkannt wird. Um dies in dem erweiterten Modell zu berücksichtigen, werden die Wahrscheinlichkeiten von einem Merkmal in benachbarten Teilbereichen zu einem einzigen Wert zusammengefasst. Dieser Wert ergibt sich durch den maximal auftretenden Wert in der lokalen Nachbarschaft und kann als Wahrscheinlichkeit interpretiert werden, dass das Merkmal in der gesamten lokalen Nachbarschaft enthalten ist. Mathematisch formuliert ist die Idee, in dem hierarchischen Modell aus dem verallgemeinerten hierarchischen Max-Pooling Modell Entscheidungen $f_{k,s} : [0,1]^{\{1,\dots,2^k\}\times\{1,\dots,2^k\}} \to [0,1]$ für benachbarte Teilbereiche eines Bildes $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$ der Größe $n \in \mathbb{N}$ wie folgt zu einzelnen Werten zusammenzufassen:

Wir berechnen das Maximum

$$\max_{(i_2,j_2)\in\{(i-1)\cdot n+1,\dots,i\cdot n\}\times\{(j-1)\cdot n+1,\dots,j\cdot n\}\cap\{1,\dots,d_1-2^k+1\}\times\{1,\dots,d_2-2^k+1\}} f_{k,s}\big(\mathbf{x}_{\{i_2,\dots,i_2+2^k-1\}\times\{j_2,\dots,j_2+2^k-1\}}\big)$$

für $(i, j) \in \{1, \dots, \lceil d_1/n \rceil\} \times \{1, \dots, \lceil d_2/n \rceil\}$. Auf diese Weise wird die Auflösung des hierarchischen Modells verringert und es werden anschließend Entscheidungen von Teilbereichen kombiniert, welche einen variablen relativen Abstand zueinander besitzen. Gemäß der obigen Idee definieren die Parameter $n_1, \dots, n_{l-1} \in \mathbb{N}$ die Nachbarschaftsgrößen in dem entsprechenden Modell. Da dieses neue Modell sehr technisch ist und eine ausführliche Einführung erfordert, wird es an dieser Stelle nicht aufgeführt. Eine detaillierte Einführung und Definition des Modells befindet sich in Abschnitt 2.2 (siehe Definition 4).

Um die a-posteriori Wahrscheinlichkeit unter der Annahme dieses komplexeren Modells zu schätzen, werden dann drei verschiedene Netzwerkarchitekturen eingeführt, welche neben faltenden Schichten und der Ausgabeschicht diesmal auch lokale Max-Pooling bzw. Subsampling Schichten enthalten. Die erste Netzwerkarchitektur enthält $L \cdot z^{(1)}$ faltende Schichten, $L - 1$ lokale Max-Pooling Schichten (für eine Definition siehe

Gleichung (1.13)) und ist der Form

$$\begin{aligned} f^{(1)}(\mathbf{x}) = & f_{out}^{(\mathbf{A})} \circ o_{(k,k),M_L,1,\mathbf{w}_{L\cdot z^{(1)}}} \circ \cdots \circ o_{(k,k),M_L,1,\mathbf{w}_{(L-1)\cdot z^{(1)}+1}} \\ & \circ f_{max}^{(s_{L-1})} \circ o_{(k,k),M_{L-1},1,\mathbf{w}_{(L-1)\cdot z^{(1)}}} \circ \cdots \circ o_{(k,k),M_{L-1},1,\mathbf{w}_{(L-2)\cdot z^{(1)}+1}} \circ \dots \\ & \circ f_{max}^{(s_1)} \circ o_{(k,k),M_1,1,\mathbf{w}_{z^{(1)}}} \circ \cdots \circ o_{(1,k),M_1,1,\mathbf{w}_1}(\mathbf{x}) \qquad \left(\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}\right). \end{aligned} \tag{1.23}$$

Die zweite Netzwerkarchitektur enthält $L \cdot z^{(2)}$ faltende Schichten, $L-1$ lokale Subsampling Schichten (für eine Definition siehe Gleichung (1.14)) und ist der Form

$$\begin{aligned} f^{(2)}(\mathbf{x}) = & f_{out}^{(\mathbf{A})} \circ o_{(k,k),M_L,1,\mathbf{w}_{L\cdot z^{(2)}}} \circ \cdots \circ o_{(k,k),M_L,1,\mathbf{w}_{(L-1)\cdot z^{(2)}+1}} \\ & \circ f_{sub}^{(s_{L-1})} \circ o_{(k,k),M_{L-1},1,\mathbf{w}_{(L-1)\cdot z^{(2)}}} \circ \cdots \circ o_{(k,k),M_{L-1},1,\mathbf{w}_{(L-2)\cdot z^{(2)}+1}} \circ \dots \\ & \circ f_{sub}^{(s_1)} \circ o_{(k,k),M_1,1,\mathbf{w}_{z^{(2)}}} \circ \cdots \circ o_{(1,k),M_1,1,\mathbf{w}_1}(\mathbf{x}) \qquad \left(\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}\right). \end{aligned} \tag{1.24}$$

Die dritte Netzwerkarchitektur enthält $z^{(3)}$ faltende Schichten, eine einzige lokale Subsampling Schicht und ist der Form

$$f^{(3)}(\mathbf{x}) = f_{out}^{(\mathbf{A})} \circ f_{sub}^{(s)} \circ o_{(k,k),M_{z^{(3)}},1,\mathbf{w}_{z^{(3)}}} \circ \cdots \circ o_{(1,k),M_1,1,\mathbf{w}_1}(\mathbf{x}) \quad \left(\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}\right). \tag{1.25}$$

Die entsprechenden Bildklassifikatoren $f_n^{(j)}(\cdot, \mathcal{D}_n) : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \{0,1\}$ $(j = 1,2,3)$ sind dann wieder als Plug-In Klassifikatoren (1.2) der Kleinste-Quadrate-Schätzer (1.4) durch die obigen Netzwerkarchitekturen (1.23), (1.24) und (1.25) definiert (die genauen Schätzer werden vollständig in Abschnitt 3.1 definiert). Das folgende Resultat beschreibt dann das Konvergenzverhalten der Bildklassifikatoren $f_n^{(j)}$ $(j = 1,2,3)$ im hierarchischen Max-Pooling Modell mit zusätzlichem lokalen Max-Pooling.

Resultat II. Es sei $n > 1$. Die a-posteriori Wahrscheinlichkeit $\eta(\mathbf{x}) = \mathbf{P}\{Y = 1 | \mathbf{X} = \mathbf{x}\}$ genüge einem hierarchischen Max-Pooling Modell vom Level l, lokalem Max-Pooling Parameter $(n_1, \dots, n_{l-1}) \in \mathbb{N}^{l-1}$ und Glattheitsbedingung $p \in [1, \infty)$ mit geeigneten Modellparametern. Weiterhin sei

$$L_n = n^{\frac{4}{2\cdot(2\cdot p+4)}}.$$

Die Netzwerkarchitekturen der Klassifikatoren $f_n^{(j)}(\cdot, \mathcal{D}_n)$ $(j = 1,2,3)$ aus den Gleichungen (1.23), (1.24) und (1.25) seien so gewählt, dass $z^{(1)} = \lceil c_4 \cdot L_n \rceil$, $z^{(2)} = \lceil c_5 \cdot L_n \rceil$ und $z^{(3)} = \lceil c_6 \cdot L_n \rceil$, wobei c_4, c_5 und c_6 sowie die sonstigen Netzwerkparameter genügend großen Konstanten entsprechen, die möglicherweise von den Modellparametern der a-posteriori Wahrscheinlichkeit abhängen. Dann gilt für alle $j \in \{1,2,3\}$

$$\mathbf{E}\{L(f_n^{(j)})\} - L(f^*) \leq c_7 \cdot \sqrt{\log(d_1 \cdot d_2)} \cdot (\log n)^2 \cdot n^{-\frac{p}{2\cdot p+4}}$$

für eine Konstante $c_7 > 0$, die nicht von d_1, d_2 und n abhängt.

Für eine präzise Formulierung des obigen Resultats, bei der die Parameter der Netzwerkarchitekturen und die Modellannahmen genauer spezifiziert werden, siehe Theorem 3.2. Da die obige Konvergenzrate

$$n^{-\frac{p}{2\cdot p+4}}$$

erneut nicht von der Bilddimension abhängt, wird auch unter der Annahme dieses Modells der Fluch der hohen Dimension umgangen. Damit konnte aus theoretischer Sicht gezeigt werden, dass allgemeinere Netzwerkarchitekturen, die lokale Pooling Schichten enthalten, in einigen Situationen der Bildklassifikation nützlich

sind. Das Resultat gibt außerdem theoretische Hinweise für die Wahl einer geeigneten Netzwerkarchitektur. Überraschenderweise ist es hierbei möglich, die gleiche Konvergenzrate auch für einen Klassifikator zu zeigen (den Klassifikator $f_n^{(3)}$), der auf einer Netzwerkarchitektur basiert, die neben faltenden Schichten und der Ausgabeschicht lediglich eine einzige Subsampling Schicht enthält.

Im dritten statistischen Modell zur Bildklassifikation, dem *rotationssymmetrischen hierarchischen Max-Pooling Modell vom Level l, der Breite h und Glattheitsbedingung $p \in [1, \infty)$*, werden Bildklassifikationsprobleme betrachtet, bei denen es für die richtige Klassifizierung egal ist, ob die relevanten Objekte um beliebige Winkel rotiert dargestellt sind. Werden Bilder wie bisher diskret als Pixelwerte auf dem endlichen Gitter $\{1, \dots, d_1\} \times \{1, \dots, d_2\}$ definiert, ist es nicht möglich, Teilbereiche von Bildern um beliebige Winkel zu rotieren, da die rotierten Teilbereiche nicht zwingend auf dem entsprechenden Gitter liegen (dies kann nur für den Fall von Vielfachen von $90°$ Rotationen garantiert werden). Aus diesem Grund wird die Annahme getroffen, dass die beobachteten (diskreten) Bilder aus idealisierten kontinuierlichen Bildern entstehen, indem diese auf einem Gitter ausgewertet werden. Ein kontinuierliches Bild wird als $[0, 1]$-wertige Funktion auf dem Quader

$$C_h = \left[-\frac{h}{2}, \frac{h}{2}\right] \times \left[-\frac{h}{2}, \frac{h}{2}\right] \subset \mathbb{R}^2$$

beschrieben, welcher die Bildfläche der Breite $h > 0$ definiert. Der Funktionswert an der Stelle $\mathbf{u} \in C_h$ wird dann als entsprechender Graustufenwert interpretiert. Den Raum aller (kontinuierlichen) Bilder der Breite h wird mit

$$[0, 1]^{C_h} := \{f : C_h \to [0, 1] \ : \ f \text{ ist eine Abbildung}\}$$

bezeichnet. Da der Raum aller auf C_h definierten $[0, 1]$-wertigen Funktionen, ausgestattet mit der durch die Supremumsnorm $\|\cdot\|_\infty$ induzierten Metrik, einen metrischen Raum definiert, erhalten wir einen messbaren Raum $([0, 1]^{C_h}, \mathcal{B}([0, 1]^{C_h}))$ mit der entsprechenden Borel-σ-Algebra auf dem durch die offenen Mengen gegebenen topologischen Raum. Für das statistische Setting zur Bildklassifizierung seien (Φ, Y), (Φ_1, Y_1), $\dots$, (Φ_n, Y_n) unabhängig identisch verteilte Zufallsvariablen mit Werten in $[0, 1]^{C_1} \times \{0, 1\}$. Wie oben erwähnt, wird angenommen, dass ein beobachtetes diskretes Bild aus einem kontinuierlichen Bild entsteht, indem dieses auf einem Gitter ausgewertet wird. Hierfür wird die Bildfläche C_1 in λ^2 gleichgroße Quadrate unterteilt und das kontinuierliche Bild an den Mittelpunkten aller Quadrate ausgewertet, wobei mit $\lambda \in \mathbb{N}$ die Auflösung des diskreten Bildes bezeichnet wird. Damit eine solche Diskretisierung mit der Notation aus Abschnitt 1.2 verträglich ist, werden die diskretisierten Bilder als Elemente von $[0, 1]^{\{1,\dots,\lambda\}^2}$ aufgefasst. Zu diesem Zweck wird die (stetige) Funktion $g_\lambda : [0, 1]^{C_1} \to [0, 1]^{\{1,\dots,\lambda\}^2}$ definiert, die ein gegebenes Bild $\phi \in [0, 1]^{C_1}$ auf dem Gitter

$$G_\lambda = \left\{ \left(\frac{i - \frac{1}{2}}{\lambda} - \frac{1}{2}, \frac{j - \frac{1}{2}}{\lambda} - \frac{1}{2} \right) \ : \ i, j \in \{1, \dots, \lambda\} \right\} \tag{1.26}$$

wie folgt auswertet:

$$g_\lambda(\phi) = \left(\phi \left(\left(\frac{i - \frac{1}{2}}{\lambda} - \frac{1}{2}, \frac{j - \frac{1}{2}}{\lambda} - \frac{1}{2} \right) \right) \right)_{(i,j) \in \{1,\dots,\lambda\}^2}.$$

Setzen wir $d_1 = d_2 = \lambda$ und

$$\mathbf{X} = g_\lambda(\Phi), \ \mathbf{X}_1 = g_\lambda(\Phi_1), \dots, \mathbf{X}_n = g_\lambda(\Phi_n) \tag{1.27}$$

befinden wir uns in der Situation wie in Abschnitt 1.2. Im Gegensatz zu den bisher beschriebenen Modellen zur Bildklassifikation werden im rotationssymmetrischen hierarchischen Max-Pooling Modell Annahmen an die funktionale a-posteriori Wahrscheinlichkeit

$$\eta_\Phi(\phi) = \mathbf{P}\{Y = 1 | \Phi = \phi\} \quad \left(\phi \in [0, 1]^{C_1}\right) \tag{1.28}$$

für (kontinuierliche) Bilder formuliert. Um Teilbereiche von Bildern zu rotieren, führen wir die Rotaionsfunktion $rot^{(\alpha)} : \mathbb{R}^2 \to \mathbb{R}^2$ ein, welche durch

$$rot^{(\alpha)}(\mathbf{x}) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \cdot \mathbf{x} \quad (\mathbf{x} \in \mathbb{R}^2)$$

gegeben ist. Diese rotiert einen zweidimensionalen Vektor $\mathbf{x} \in \mathbb{R}^2$ gegen den Uhrzeigersinn mit dem Winkel $\alpha \in [0, 2\pi]$ um den Ursprung $\mathbf{0} \in \mathbb{R}^2$. Außerdem definieren wir die Translation $\tau_{\mathbf{v}} : \mathbb{R}^2 \to \mathbb{R}^2$ um den Vektor $\mathbf{v} \in \mathbb{R}^2$ durch

$$\tau_{\mathbf{v}}(\mathbf{x}) = \mathbf{x} + \mathbf{v} \quad \big(\mathbf{x} \in \mathbb{R}^2\big).$$

Ein mit dem Uhrzeigersinn um den Winkel $\alpha \in \mathbb{R}$ rotierter Teilbereich eines Bildes $\phi \in [0,1]^{C_1}$ mit der Breite $0 < h \leq 1/\sqrt{2}$ an der Bildposition $\mathbf{v} \in [-1/2 + h/\sqrt{2}, 1/2 - h/\sqrt{2}]^2$ ist dann durch die Funktion

$$\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}\big|_{C_h} \in [0,1]^{C_h}$$

gegeben (für eine Darstellung siehe Abbildung 2.5). In unserem Modell für die funktionale a-posteriori Wahrscheinlichkeit nehmen wir nun an, dass eine Funktion $f : [0,1]^{C_h} \to [0,1]$ existiert, die für einen Teilbereich der Breite $0 < h \leq 1/\sqrt{2}$ eines kontinuierlichen Bildes die Wahrscheinlichkeit berechnet, dass ein spezifisches Objekt in diesem Teilbereich in einer nicht-rotierten Version dargestellt wird. Die a-posteriori Wahrscheinlichkeit berechnet sich dann, indem in die Funktion f alle möglichen unterschiedlich rotierten Teilbereiche eingesetzt werden und das Supremum über alle Positionen der Teilbereiche und alle möglichen Rotationswinkel berechnet wird:

$$\eta_\Phi(\phi) = \sup_{\mathbf{v} \in \left[-\frac{1}{2}+b, \frac{1}{2}-b\right]^2} \sup_{\alpha \in [0,2\pi]} f\left(\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}\big|_{C_h}\right) \quad \big(\phi \in [0,1]^{C_1}\big).$$

Der Parameter b mit $h/\sqrt{2} \leq b \leq 1/2$ gibt dabei an, wie weit die Teilbereiche an den Rand des Bildbereichs herankommen. Die Funktion f genügt ähnlich den beiden vorherigen Modellen auch einem hierarchischen Modell vom Level l, bei dem Entscheidungen von vier benachbarten kleineren Teilbereichen durch (p, C)-glatte Funktionen $g_{k,s} : \mathbb{R}^4 \to [0,1]$ $(k = 1, \ldots, l, s = 1, \ldots, 4^{l-k})$ zu Entscheidungen der zusammengesetzten Teilbereiche kombiniert werden (vgl. Gleichung (1.20)). Analog zum verallgemeinerten hierarchischen Max-Pooling Modell ergeben sich auf diese Weise die Funktionen $f_{k,s} : [0,1]^{C_{h/2^{l-k}}} \to [0,1]$ $(k = 0, \ldots, l, s = 1, \ldots, 4^{l-k})$, die Entscheidungen für die verschiedenen (kleineren) Teilbereiche des in f eingesetzten Teilbereichs treffen. Außerdem werden für dieses Modell zwei weitere Annahmen formuliert (siehe Annahme 1 und Annahme 2 in Abschnitt 2.3). Diese betreffen die Funkionen $f_{0,s}$ $(s = 1, \ldots, 4^l)$, welche die Entscheidungen für die kleinsten Teilbereiche der Breite $h/2^l$ berechnen. Die zweite Annahme hängt dabei von einem von der Auflösung λ abhängigen Fehlerterm $\epsilon_\lambda \in [0,1]$ ab. Wir nehmen dabei an, dass der Fehlerterm für große Auflösungen λ klein ist (für ein Beispiel, in dem dies der Fall ist siehe Beispiel 2.1). Eine detaillierte Einführung und Motivation des rotationssymmetrischen hierarchischen Max-Pooling Modells befindet sich in Abschnitt 2.3.

Zur Klassifikation innerhalb dieses Modells verwenden wir eine ähnliche Netzwerkarchitektur wie für das verallgemeinerte hierarchische Max-Pooling Modell von Resultat I. Die Architektur unterscheidet sich lediglich darin, dass statt des einseitigen Zero-Paddings ein symmetrisches Zero-Padding verwendet wird (siehe Abschnitt 1.3 für eine Erklärung des Zero-Paddings). Die verwendeten faltenden neuronalen Netze sind damit der Form

$$f(\mathbf{x}) = g_{net}(f_1(\mathbf{x}), \ldots, f_t(\mathbf{x})) \quad \big(\mathbf{x} \in [0,1]^{\{1,\ldots,d_1\} \times \{1,\ldots,d_2\}}\big) \tag{1.29}$$

für ein $t \in \mathbb{N}$, wobei die Funktionen $f_1, \ldots, f_t : [0,1]^{\{1,\ldots,d_1\} \times \{1,\ldots,d_2\}} \to \mathbb{R}$ faltenden neuronalen Netzen mit $L \in \mathbb{N}$ faltenden Schichten und k Kanälen pro Schicht der Form

$$f_{out}^{(\mathbf{A})} \circ o_{(k,k),M_L,\lceil M_L/2 \rceil,\mathbf{w}_L} \circ \cdots \circ o_{(k,k),M_2,\lceil M_2/2 \rceil,\mathbf{w}_2} \circ o_{(1,k),M_1,\lceil M_1/2 \rceil,\mathbf{w}_1}(\mathbf{x}) \quad \big(\mathbf{x} \in [0,1]^{\{1,\ldots,d_1\} \times \{1,\ldots,d_2\}}\big)$$

entsprechen und das vollverbundene neuronale Netz $g_{net} : \mathbb{R}^t \to \mathbb{R}$ $L_{net} \in \mathbb{N}$ verdeckte Schichten und $r_{net} \in \mathbb{N}$ Neuronen in jeder Schicht besitzt. Der entsprechende Bildklassifikator $f_n(\cdot, \mathcal{D}_n) : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \{0,1\}$ ist dann wieder als Plug-In Klassifikator (1.2) des Kleinste-Quadrate-Schätzers (1.4) über die durch die obige Form (1.29) gegebene Netzwerkarchitektur definiert (der genaue Schätzer wird in Abschnitt 3.1 vollständig definiert). Die Beobachtungen entsprechen hierbei

$$\mathcal{D}_n = \{(g_\lambda(\Phi_1), Y_1), \dots, (g_\lambda(\Phi_n), Y_n)\}.$$

Das folgende Resultat beschreibt dann das Konvergenzverhalten des Bildklassifikators f_n im rotationssymmetrischen hierarchischen Max-Pooling Modell.

Resultat III. Es sei $n > 1$. Die funktionale a-posteriori Wahrscheinlichkeit $\eta_\Phi(\phi) = \mathbf{P}\{Y = 1 | \Phi = \phi\}$ genüge einem rotationssymmetrischen hierarchischen Max-Pooling Modell vom Level l, der Breite h mit Glattheitsbedingung $p \in [1, \infty)$ für geeignete Modellparameter. Außerdem erfülle das Modell die zwei oben erwähnten weiteren Annahmen (siehe Annahme 1 und Annahme 2 in Abschnitt 2.3). Die zweite Annahme sei dabei für einen Fehlerterm $\epsilon_\lambda \in [0,1]$ erfüllt. Weiterhin sei

$$L_n = n^{\frac{4}{2\cdot(2\cdot p+4)}}.$$

Die Netzwerkarchitektur des faltenden neuronalen Netzes $f : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}$ des Klassifikators $f_n(\cdot, \mathcal{D}_n)$ aus Gleichung (1.29) sei so gewählt, dass sie $L = \lceil c_8 \cdot L_n \rceil$ faltende Schichten besitzt, wobei c_8 sowie die sonstigen Netzwerkparameter genügend großen Konstanten entsprechen, die möglicherweise von den Modellparametern der funktionalen a-posteriori Wahrscheinlichkeit abhängen. Dann gilt

$$\mathbf{E}\{L(f_n)\} - L(f^*) \leq c_9 \cdot \sqrt{\log(\lambda) \cdot (\log n)^4 \cdot n^{-\frac{2p}{2p+4}} + \epsilon_\lambda}$$

für eine Konstante $c_9 > 0$, die nicht von λ und n abhängt.

Für eine präzise Formulierung des obigen Resultats, bei der die Parameter der Netzwerkarchitekturen und die Modellannahmen genauer spezifiziert werden, siehe Theorem 3.3. Vernachlässigt man den Fehlerterm ϵ_λ in

$$\sqrt{\log(\lambda) \cdot (\log n)^4 \cdot n^{-\frac{2p}{2p+4}} + \epsilon_\lambda},$$

ist der Schätzer daher auch hier in der Lage, den Fluch der hohen Dimension zu umgehen. Das Resultat liefert wieder theoretische Hinweise für die Wahl einer geeigneten Netzwerkarchitektur. Die Rotationssymmetrie innerhalb des Modells für die a-posteriori Wahrscheinlichkeit wird dadurch erreicht, dass die Filter der parallel berechneten faltenden neuronalen Netze die gleichen Gewichte an unterschiedlich rotierten Postionen besitzen. Diese Erkenntnis gibt Hinweise auf neue Netzwerkarchitekturen, die stärker auf den Aspekt der Rotation ausgelegt sind.

2. Statistische Modelle zur Bildklassifikation

Wie wir in Abschnitt 1.5 gesehen haben, ist es nötig, die Klasse der Verteilungen von $(\mathbf{X}, Y)$ einzuschränken, um nichttriviale Aussagen über das Konvergenzverhalten von Bildklassifikatoren tätigen zu können. Wir werden daher in diesem Kapitel zwei Modelle für die a-posteriori Wahrscheinlichkeit (1.1) und ein Modell für die funktionale a-posteriori Wahrscheinlichkeit (1.28) einführen. Die vorgestellten Modellannahmen sollen jeweils Aspekte der Bildklassifikation berücksichtigen, die plausibel für entsprechende Anwendungen erscheinen und gleichzeitig möglichst allgemein sein, d.h. viele solcher Probleme enthalten.

2.1. Verallgemeinertes hierarchisches Max-Pooling Modell

Für das erste Modell für die a-posteriori Wahrscheinlichkeit, welches wir hier vorstellen werden, berücksichtigen wir die folgenden drei grundlegenden Beobachtungen zur Bildklassifikation:

(B1) *Ob ein Bild einer bestimmten Klasse angehört, hängt davon ab, ob das Bild spezielle Objekte enthält (in Abbildung 2.1a beispielsweise das einzelne Objekt „Hund“).*

(B2) *Ein für die korrekte Klassifizierung relevantes Objekt befindet sich in einem Teilbereich des Bildes, welcher möglicherweise deutlich kleiner ist als der gesamte Bildbereich (vgl. Abbildung 2.1a).*

(B3) *Ein Teilbereich eines Bildes lässt sich durch Unterteilung aus benachbarten kleineren Teilbereichen zusammensetzen (vgl. Abbildung 2.1b). Ob ein Objekt in einem Teilbereich enthalten ist, hängt dann davon ab, was jeweils auf den kleineren benachbarten Teilbereichen dargestellt ist.*

(a) Für Klassifizierung relevantes Objekt befindet sich in einem Teilbereich des Bildes.

(b) Ein Bild lässt sich aus kleineren benachbarten Teilbereichen zusammensetzen.

Abbildung 2.1.: Darstellung der grundlegenden Beobachtungen zur Bildklassifikation.

Zur Formulierung struktureller Annahmen für die a-posteriori Wahrscheinlichkeit überlegen wir uns nun, wie ein Mensch unter Beachtung der Beobachtungen *(B1)*, *(B2)* und *(B3)* vorgehen würde, um die Wahrscheinlichkeit zu bestimmen, dass ein Bild einer bestimmten Klasse angehört. Aus der Vorgehensweise leiten wir wie folgt strukturelle Annahmen für die a-posteriori Wahrscheinlichkeit her:

1. Wegen Beobachtung *(B1)* würde ein Mensch zunächst entscheiden, ob das Bild die entsprechenden Objekte enthält. Hierfür kann der Mensch die Wahrscheinlichkeiten der Existenz der einzelnen Objekte schätzen. Dies geschieht im *verallgemeinerten hierarchischen Max-Pooling Modell* unten durch die Funktionen $m_1, \dots, m_{d^*} : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to [0,1]$ in Teil d) von Definition 3 für $d^* \in \mathbb{N}$ Objekte. Um aus den Wahrscheinlichkeiten der Existenz der einzelnen Objekte die Wahrscheinlichkeit für die Klasse des Bildes abzuleiten, wird dann die Funktion $g : \mathbb{R}^{d^*} \to [0,1]$ auf die Ausgaben der Funktionen $m_1, \dots, m_{d^*}$ angewendet.

2. Um die Wahrscheinlichkeit für ein spezielles Objekt zu schätzen, könnte ein Mensch das gesamte Bild inspizieren und wegen Beobachtung *(B2)* für jeden Teilbereich eine Wahrscheinlichkeit schätzen, dass der entsprechende Teilbereich das gesuchte Objekt enthält. Wäre die Wahrscheinlichkeit in einem Teilbereich hinreichend groß, würde er daraus schließen, dass sich das entsprechende Objekt auf dem Bild befindet. Wir nehmen daher an, dass sich die Wahrscheinlichkeit für das gesamte Bild als Maximum der Wahrscheinlichkeiten aller Teilbereiche berechnet. Dies führt zu dem *Max-Pooling Modell* aus Definition 3 a). Die Wahrscheinlichkeit für einen Teilbereich wird hier durch die Funktion f geschätzt (siehe Definition 3 a)).

3. Eine weitere strukturelle Annahme wird aus Beobachtung *(B3)* abgeleitet: Die Entscheidung ob ein Teilbereich ein spezielles Objekt enthält kann ein Mensch schätzen, indem er den Teilbereich weiter in benachbarte kleinere Bereiche unterteilt und zunächst für die kleineren Bereiche Entscheidungen trifft. Diese Entscheidungen kann er dann wiederum zu einer Entscheidung für den größeren Teilbereich kombinieren. Wiederholt er dieses Vorgehen auch für die kleineren Bereiche, führt das zu dem *hierarchischen Modell vom Level l* in Definition 3 b). Hier wird ein quadratischer Teilbereich entsprechend in vier kleinere Quadrate unterteilt. Die Entscheidungen für einzelne Teilbereiche werden durch die Funktionen $f_{k,s} : [0,1]^{\{1,\dots,2^k\}\times\{1,\dots,2^k\}} \to \mathbb{R}$ getroffen. Der Wert von $k \in \{1,\dots,l\}$ gibt dabei die Größe bzw. das Level des betrachteten Teilbereichs an und der Wert von $s \in \{1,\dots,4^{l-k}\}$ gibt an, um welchen Teilbereich vom Level k es sich handelt. Die Funktionen $g_{k,s} : \mathbb{R}^4 \to [0,1]$ kombinieren dann die Entscheidungen von vier benachbarten Teilbereichen zu einer Entscheidung des zusammengesetzten größeren Teilbereichs.

Die folgende Definition liefert dann das entsprechende Modell für die a-posteriori Wahrscheinlichkeit.

Definition 3. Es seien $d_1, d_2 \in \mathbb{N}$ mit $d_1, d_2 > 1$ und $m : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}$.

a) Wir sagen, dass die Funktion m einem *Max-Pooling Modell mit Indexmenge*

$$I \subseteq \{0,\dots,d_1-1\} \times \{0,\dots,d_2-1\},$$

genügt, falls eine Funktion $f : [0,1]^{(1,1)+I} \to \mathbb{R}$ existiert, sodass

$$m(\mathbf{x}) = \max_{(i,j)\in\mathbb{Z}^2 \,:\, (i,j)+I \subseteq \{1,\dots,d_1\}\times\{1,\dots,d_2\}} f\left(\mathbf{x}_{(i,j)+I}\right) \quad (\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}).$$

b) Es sei $I = \{0,\dots,2^l-1\} \times \{0,\dots,2^l-1\}$ für ein $l \in \mathbb{N}$. Wir sagen, dass die Funktion

$$f : [0,1]^{\{1,\dots,2^l\}\times\{1,\dots,2^l\}} \to \mathbb{R}$$

einem *hierarchischen Modell vom Level l* genügt, falls Funktionen

$$g_{k,s} : \mathbb{R}^4 \to [0,1] \quad (k = 1,\dots,l, s = 1,\dots,4^{l-k})$$

existieren, sodass

$$f = f_{l,1},$$

wobei die Funktion $f_{l,1}$ durch die Funktionen $f_{k,s} : [0,1]^{\{1,\dots,2^k\}\times\{1,\dots,2^k\}} \to \mathbb{R}$ wie folgt rekursiv definiert ist:

$$\begin{aligned} f_{k,s}(\mathbf{x}) \;=\; & g_{k,s}\big(f_{k-1,4\cdot(s-1)+1}(\mathbf{x}_{\{1,\dots,2^{k-1}\}\times\{1,\dots,2^{k-1}\}}), \\ & \quad f_{k-1,4\cdot(s-1)+2}(\mathbf{x}_{\{2^{k-1}+1,\dots,2^k\}\times\{1,\dots,2^{k-1}\}}), \\ & \quad f_{k-1,4\cdot(s-1)+3}(\mathbf{x}_{\{1,\dots,2^{k-1}\}\times\{2^{k-1}+1,\dots,2^k\}}), \\ & \quad f_{k-1,4\cdot s}(\mathbf{x}_{\{2^{k-1}+1,\dots,2^k\}\times\{2^{k-1}+1,\dots,2^k\}})\big) \\ & \qquad\qquad \Big(\mathbf{x} \in [0,1]^{\{1,\dots,2^k\}\times\{1,\dots,2^k\}}\Big) \end{aligned}$$

für $k = 2,\dots,l$, $s = 1,\dots,4^{l-k}$, und

$$f_{1,s}(x_{1,1},x_{1,2},x_{2,1},x_{2,2}) = g_{1,s}(x_{1,1},x_{1,2},x_{2,1},x_{2,2}) \quad (x_{1,1},x_{1,2},x_{2,1},x_{2,2} \in [0,1])$$

für $s = 1,\dots,4^{l-1}$.

c) Wir sagen, dass $m : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}$ einem *hierarchischen Max-Pooling Modell vom Level l* genügt (wobei $2^l \leq \min\{d_1,d_2\}$), falls die Funktion m einem Max-Pooling Modell mit der Indexmenge

$$I = \Big\{0,\dots,2^l-1\Big\} \times \Big\{0,\dots,2^l-1\Big\}$$

genügt und die Funktion $f : [0,1]^{(1,1)+I} \to \mathbb{R}$ in der Definition des Max-Pooling Modells einem hierarchischen Modell vom Level l genügt.

d) Sei $d^* \in \mathbb{N}$. Wir sagen, dass $m : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}$ einem *verallgemeinerten hierarchischen Max-Pooling Modell der Ordnung d^* und Level l* genügt, falls Funktionen

$$m_1,\dots,m_{d^*} : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to [0,1]$$

existieren, welche einem hierarchischen Max-Pooling Modell vom Level l genügen, und falls eine Funktion $g : \mathbb{R}^{d^*} \to [0,1]$ existiert, sodass

$$m(\mathbf{x}) = g(m_1(\mathbf{x}),\dots,m_{d^*}(\mathbf{x})).$$

e) Es seien $p_1,p_2 \in (0,\infty)$. Wir sagen, dass ein *verallgemeinertes hierarchisches Max-Pooling Modell der Ordnung d^* und Level l die Glattheitsbedingungen p_1 und p_2* erfüllt, falls alle Funktionen $g_{k,s}$ in den Definitionen der Funktionen m_i für ein $C_1 > 0$ und für alle $i \in \{1,\dots,d^*\}$ (p_1,C_1)–glatt sind, und falls die Funktion g (p_2,C_2)–glatt ist für ein $C_2 > 0$.

Bemerkung 2.1. Die Unterteilung in Teilbereiche in Teil b) der Definition erfolgt wie in Abbildung 2.1b dargestellt.

Bemerkung 2.2. Das verallgemeinerte hierarchische Max-Pooling Modell erfüllt die strukturellen Annahmen eines in Definition 2 eingeführten hierarchischen Kompositionsmodells von Kohler und Langer (2021).

2.2. Hierarchisches Max-Pooling Modell mit zusätzlichem lokalen Max-Pooling

In diesem Abschnitt interpretieren wir den Wert von $f_{k,s}(\mathbf{x}_{\{i,\dots,i+2^k-1\}\times\{j,\dots,j+2^k-1\}})$ aus dem hierarchischen Modell von Definition 3 b) als Wahrscheinlichkeit, dass der Teilbereich $\mathbf{x}_{\{i,\dots,i+2^k-1\}\times\{j,\dots,j+2^k-1\}}$ des Bildes $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$ ein bestimmtes Merkmal besitzt. In Definition 3 Teil b) kombinieren wir dann vier Wahrscheinlichkeiten von Merkmalen aus vier Teilbereichen, die einen festen relativen Abstand zueinander haben (der Abstand der Teilbereiche beträgt 2^k bei Teilbereichen der Größe $2^k \times 2^k$). In einigen Bildklassifikationsproblemen ist die exakte Position von Merkmalen relativ zueinander aber nicht entscheidend. Betrachten wir zum Beispiel Abbildung 2.1b, so ist die exakte Positionierung des Hundekopfes, des Schwanzes und der Beine relativ zueinander nicht entscheidend, wir könnten die Position dieser Objekte leicht abändern und würden dennoch einen Hund erkennen. Dies führt uns zu unserer nächsten Beobachtung.

(B4) Bei einigen Bildklassifikationsproblemen kommt es nur auf den ungefähren relativen Abstand von Merkmalen zueinander an.

Das Ziel ist es nun, das hierarchische Max-Pooling Modell aus dem vorherigen Abschnitt um diesen weiteren Aspekt der Bildklassifikation zu erweitern und damit ein noch realistischeres Modell für einige Anwendungen zu erhalten. Zu diesem Zweck bringen wir das hierarchische Max-Pooling Modell zunächst in einen abstrakteren Rahmen. Die Funktionen $f_{k,s}$ in Definition 3 b) werden aufgrund der Definition des Max-Pooling Modells aus Teil a) auf viele Teilbereiche des Bildes der Größe $2^k \times 2^k$ angewendet. Betrachten wir die Funktion $f_{k,s}$ nun für alle möglichen Teilbereiche eines Input Bildes $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$ gleichzeitig, liefert das eine neue Darstellung des Input Bildes. Diese Darstellung können wir durch eine Funktion

$$y_{k,s} : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to [0,1]^{\{1,\dots,d_1(k)\}\times\{1,\dots,d_2(k)\}}$$

beschreiben, welche durch

$$y_{k,s}(\mathbf{x}) = \left(f_{k,s}(\mathbf{x}_{\{i,\dots,i+2^k-1\}\times\{j,\dots,j+2^k-1\}})\right)_{(i,j)\in\{1,\dots,d_1(k)\}\times\{1,\dots,d_2(k)\}} \quad \left(\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}\right) \tag{2.1}$$

mit Dimensionen

$$d_1(k) = d_1 - 2^k + 1 \text{ und } d_2(k) = d_2 - 2^k + 1 \tag{2.2}$$

definiert wird. Der Wert der Funktion $(y_{k,s}(\mathbf{x}))_{(i,j)}$ entspricht dann der Wahrscheinlichkeit eines Merkmals für den Teilbereich $\mathbf{x}_{\{i,\dots,i+2^k-1\}\times\{j,\dots,j+2^k-1\}}$. Die Wahl (2.2) der Dimensionen $d_1(k)$ und $d_2(k)$ stellt sicher, dass alle Teilbereiche betrachtet werden, welche sich vollständig im Bildbereich befinden. Außerdem nehmen wir an, dass es für Teilbereiche der Größe $2^k \times 2^k$ endlich viele mögliche relevante Merkmale $s \in \{1,\dots,b_k\}$ für ein $b_k \in \mathbb{N}$ gibt. Daher nennen wir die Darstellung (2.1) des Input Bildes *Feature Map vom Level* $k \in \{1,\dots,l\}$ *des Merkmals* $s \in \{1,\dots,b_k\}$. In Definition 3 b) bleibt es dagegen unklar, wie viele mögliche Merkmale es für Teilbereiche der Größe $2^k \times 2^k$ gibt, da für jeden Teilbereich $s \in \{1,\dots,4^{l-k}\}$ des ursprünglichen Teilbereichs der Größe $2^l \times 2^l$ eine extra Funktion $f_{k,s} : [0,1]^{\{1,\dots,2^k\}\times\{1,\dots,2^k\}} \to \mathbb{R}$ eingeführt wurde. Unsere neue Notation erweist sich insbesondere als nützlich, wenn die Werte von b_k deutlich kleiner als 4^{l-k} sind. Der Grund ist, dass die später eingeführten Netzwerkarchitekturen von faltenden neuronalen Netzen dann deutlich weniger Schichten und Kanäle pro Schicht benötigen (siehe Theorem 3.2). Da der Fall $b_{k-1} < 4 \cdot b_k$ für $k \in \{1,\dots,l\}$ erlaubt ist, kombinieren wir in unserem hierarchischen Modell aus Definition 3 b) zur Berechnung von $f_{k,s}(\mathbf{x}_{\{i,\dots,i+2^k-1\}\times\{j,\dots,j+2^k-1\}}) = (y_{k,s}(\mathbf{x}))_{(i,j)}$ $(s \in \{1,\dots,b_k\})$ statt der Wahrscheinlichkeiten der vier Merkmale $4\cdot(s-1)+1$, $4\cdot(s-1)+2$, $4\cdot(s-1)+3$ und $4\cdot s$ (die Merkmale befinden sich möglicherweise nicht in der Menge $\{1,\dots,b_{k-1}\}$) nun die Wahrscheinlichkeiten der vier Merkmale

$$r_1(k,s), r_2(k,s), r_3(k,s), r_4(k,s) \in \{1,\dots,b_{k-1}\}.$$

Das hierarchische Modell aus Definition 3 b) wird dann zu

$$\big(y_{k,s}(\mathbf{x})\big)_{(i,j)} = g_{k,s}\Big(\big(y_{k-1,r_1(k,s)}(\mathbf{x})\big)_{(i,j)}, \big(y_{k-1,r_2(k,s)}(\mathbf{x})\big)_{(i+2^{k-1},j)}, \\ \big(y_{k-1,r_3(k,s)}(\mathbf{x})\big)_{(i,j+2^{k-1})}, \big(y_{k-1,r_4(k,s)}(\mathbf{x})\big)_{(i+2^{k-1},j+2^{k-1})}\Big)$$

für $k = 1, \dots, l$, $s = 1, \dots, b_k$, $r_1(k,s), r_2(k,s), r_3(k,s), r_4(k,s) \in \{1, \dots, b_{k-1}\}$ und $(i,j) \in \{1, \dots, d_1(k)\} \times \{1, \dots, d_2(k)\}$. In unserem Beispiel aus Abbildung 2.1b könnten die Merkmale zur Berechnung von $y_{l,1}(\mathbf{x})$ (Feature Map des Merkmals „Hund“) beispielsweise wie folgt interpretiert werden:

$r_3(l,1)$ entspricht „Hundekopf“, $r_4(l,1)$ entspricht „Hundeschwanz“,
$r_1(l,1)$ entspricht „Hundebeine“, $r_2(l,1)$ entspricht „Hundebeine“,

wobei hier wegen $r_1(l,1) = r_2(l,1)$ dann $b_{l-1} = 3 < 4$ gilt. Motiviert durch die Beobachtung *(B4)* können wir nun wie folgt eine entsprechende Annahme herleiten:

Da die exakte Position von Merkmalen nicht entscheidend ist, partitionieren wir Feature Maps in lokale Nachbarschaften und nehmen an, dass wir diese lokalen Nachbarschaften jeweils zu der Wahrscheinlichkeit zusammenfassen können, sodass die gesamte lokale Nachbarschaft das entsprechende Merkmal vorweist. Analog zur Idee des Max-Pooling Modells aus Definition 3 a) inspizieren wir hierfür lokale Nachbarschaften des Bildes nach Merkmalen, indem wir für alle Teilbereiche einer lokalen Nachbarschaft die Wahrscheinlichkeit eines Merkmals berechnen und dann das Maximum der Wahrscheinlichkeiten für das Merkmal über der lokalen Nachbarschaft bilden. Zu diesem Zweck führen wir unten die Nachbarschaften $N^{(k)}_{(i,j)}$ ein. Es ergeben sich dann Feature Maps mit lokalem Max-Pooling $z_{k,s}$, die aus den oben eingeführten Feature Maps $y_{k,s}$ resultieren und eine geringere Auflösung als diese besitzen, da lokale Nachbarschaften zusammengefasst wurden (vgl. Abbildung 2.2).

Für eine gegebene Feature Map $y_{k,s}$ vom Level k mit den Dimensionen $d_1(k)$ und $d_2(k)$ definieren wir quadratische Nachbarschaften der Größe $n_k \in \mathbb{N}$ wie folgt:

Für $(i,j) \in \{1, \dots, \lceil d_1(k)/n_k \rceil\} \times \{1, \dots, \lceil d_2(k)/n_k \rceil\}$ definieren wir die Nachbarschaft

$$N^{(k)}_{(i,j)} = \Big(\{(i-1)\cdot n_k + 1, \dots, i \cdot n_k\} \times \{(j-1)\cdot n_k + 1, \dots, j \cdot n_k\}\Big) \cap \Big(\{1, \dots, d_1(k)\} \times \{1, \dots, d_2(k)\}\Big) \tag{2.3}$$

und führen für die gegebene Feature Map $y_{k,s}$ eine *Feature Map mit lokalen Max-Pooling vom Level* k durch die Funktion

$$z_{k,s} : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to [0,1]^{\left\{1,\dots,\left\lceil \frac{d_1(k)}{n_k} \right\rceil\right\} \times \left\{1,\dots,\left\lceil \frac{d_2(k)}{n_k} \right\rceil\right\}}$$

ein, welche durch

$$\big(z_{k,s}(\mathbf{x})\big)_{(i,j)} = \max_{(i_2,j_2) \in N^{(k)}_{(i,j)}} \big(y_{k,s}(\mathbf{x})\big)_{(i_2,j_2)}$$

für $(i,j) \in \{1, \dots, \lceil d_1(k)/n_k \rceil\} \times \{1, \dots, \lceil d_2(k)/n_k \rceil\}$ definiert ist. Da wir trotz des lokalen Max-Poolings noch Teilbereiche des Bildes hierarchisch kombinieren wollen, die ungefähr den Abstand der Größe der Teilbereiche selbst haben (vgl. Definition 3 b)), müssen wir diesen Abstand an die geringere Auflösung der Feature Maps mit lokalem Max-Pooling anpassen (vgl. Abbildung 2.2). Dies geschieht in der folgenden Definition durch den Parameter δ_k, welcher den angepassten Abstand der hierarchisch kombinierten Teilbereiche vom Level k beschreibt. Der Einfachheit halber betrachten wir Klassifikationsprobleme, bei denen lediglich ein einzelnes Objekt erkannt werden muss. Wir können das hierarchische Max-Pooling Modell aus Definition 3 dann durch die folgende Definition verallgemeinern.

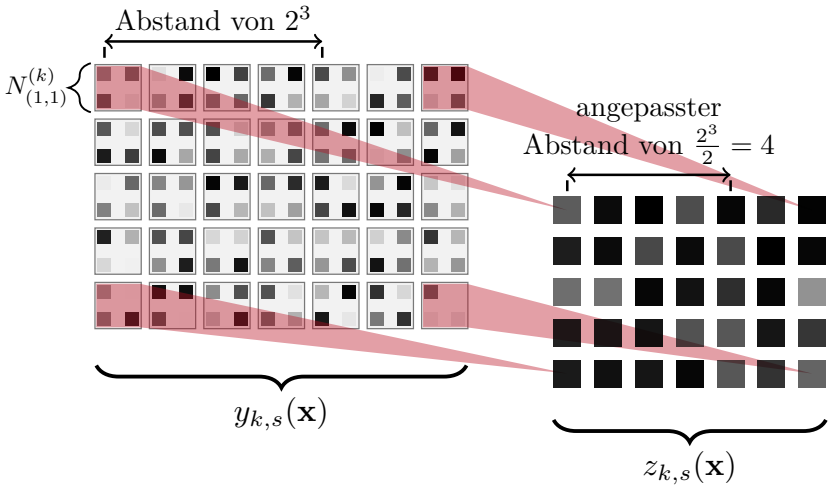


Abbildung 2.2.: Darstellung des lokalen Max-Poolings einer Feature Map mit der Nachbarschaftsgröße $n_k = 2$.

Definition 4. Es seien $d_1, d_2, l \in \mathbb{N}$, $n_0, n_1, \ldots, n_l \in \{2^0, 2^1, \ldots, 2^{l-1}\}$ mit $n_0 = n_l = 1$, sodass gilt

$$\prod_{i=1}^{k} n_i \leq 2^k \tag{2.4}$$

für $k \in \{1, \ldots, l-1\}$ und

$$\min\{d_1, d_2\} \geq 2^l + \prod_{k=1}^{l-1} n_k - 1. \tag{2.5}$$

Wir setzen $\mathbf{n} = (n_1, \ldots, n_{l-1})$ und $\mathbf{b} = (b_1, \ldots, b_{l-1})$ für $b_0, \ldots, b_l \in \mathbb{N}$ mit $b_0 = b_l = 1$.

a) Für $k = 1, \ldots, l$ setzen wir $\delta_{k-1} = 2^{k-1} / \prod_{i=0}^{k-1} n_i$ und definieren rekursiv die Dimensionen

$$d_1(k) = \left\lceil \frac{d_1(k-1)}{n_{k-1}} \right\rceil - \delta_{k-1} \quad \text{und} \quad d_2(k) = \left\lceil \frac{d_2(k-1)}{n_{k-1}} \right\rceil - \delta_{k-1}, \tag{2.6}$$

wobei wir $d_1(0) = d_1$ und $d_2(0) = d_2$ setzen. Wir sagen

$$z : [0,1]^{\{1,\ldots,d_1\} \times \{1,\ldots,d_2\}} \to [0,1]^{\{1,\ldots,d_1(l)\} \times \{1,\ldots,d_2(l)\}}$$

genügt einem *hierarchischen Modell vom Level l mit Feature Bedingung* $\mathbf{b}$ *und lokalem Max-Pooling Parameter* $\mathbf{n}$, falls Funktionen

$$g_{k,s} : \mathbb{R}^4 \to [0,1] \quad (k = 1, \ldots, l, s = 1, \ldots, b_k)$$

existieren, sodass

$$z = z_{l,1}$$

für Funktionen

$$z_{k,s} : [0,1]^{\{1,\ldots,d_1\} \times \{1,\ldots,d_2\}} \to [0,1]^{\left\{1,\ldots,\left\lceil \frac{d_1(k)}{n_k} \right\rceil\right\} \times \left\{1,\ldots,\left\lceil \frac{d_2(k)}{n_k} \right\rceil\right\}} \quad (k = 0, \ldots, l, s = 1, \ldots, b_k)$$

gilt, welche wie folgt rekursiv definiert sind:

1. Wir setzen
$$z_{0,1}(\mathbf{x}) = \mathbf{x}.$$
2. Wir verwenden folgendes hierarchisches Modell, um die Feature Maps
$$y_{k,s} : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to [0,1]^{\{1,\dots,d_1(k)\}\times\{1,\dots,d_2(k)\}}$$
zu definieren:
$$\begin{aligned}\big(y_{k,s}(\mathbf{x})\big)_{(i,j)} = g_{k,s}\Big(&\big(z_{k-1,r_1(k,s)}(\mathbf{x})\big)_{(i,j)}, \big(z_{k-1,r_2(k,s)}(\mathbf{x})\big)_{(i+\delta_{k-1},j)},\\ &\big(z_{k-1,r_3(k,s)}(\mathbf{x})\big)_{(i,j+\delta_{k-1})}, \big(z_{k-1,r_4(k,s)}(\mathbf{x})\big)_{(i+\delta_{k-1},j+\delta_{k-1})}\Big)\end{aligned}$$
für $k = 1,\dots,l$, $s = 1,\dots,b_k$, $(i,j) \in \{1,\dots,d_1(k)\}\times\{1,\dots,d_2(k)\}$ und $r_1(k,s)$, $r_2(k,s)$, $r_3(k,s)$, $r_4(k,s) \in \{1,\dots,b_{k-1}\}$ (für eine Illustration siehe Abbildung 2.3).
3. Als Nächstes definieren wir die Feature Maps mit lokalem Max-Pooling durch
$$\big(z_{k,s}(\mathbf{x})\big)_{(i,j)} = \max_{(i_2,j_2)\in N^{(k)}_{(i,j)}} \big(y_{k,s}(\mathbf{x})\big)_{(i_2,j_2)}$$
für $k = 1,\dots,l$, $s = 1,\dots,b_k$ und $(i,j) \in \{1,\dots,\lceil d_1(k)/n_k\rceil\}\times\{1,\dots,\lceil d_2(k)/n_k\rceil\}$, wobei die Nachbarschaften $N^{(k)}_{(i,j)}$ durch Gleichung (2.3) definiert sind.

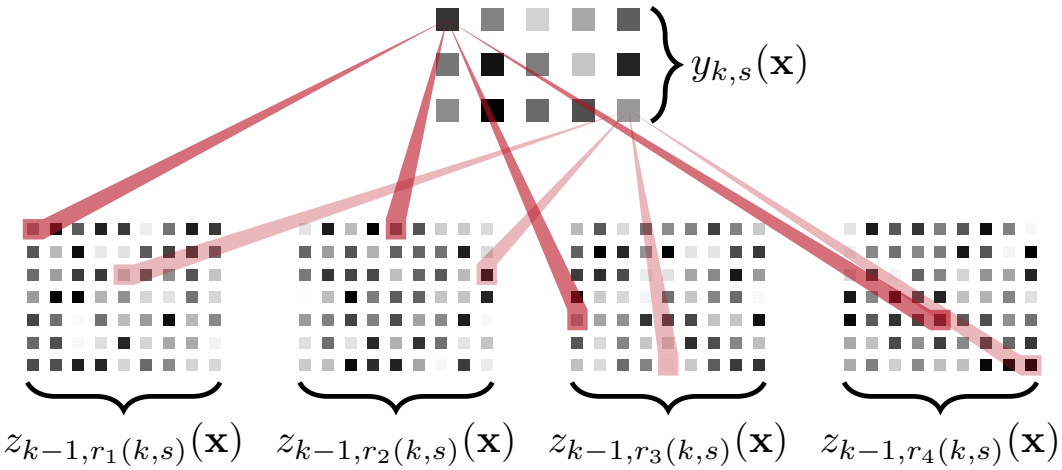


Abbildung 2.3.: Hierarchisches Modell der Feature Maps aus Definition 4 a) mit $\delta_{k-1} = 4$

b) Wir sagen $m : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to [0,1]$ genügt einem *hierarchischen Max-Pooling Modell vom Level l mit Feature Bedingung* $\mathbf{b}$ *und zusätzlichem lokalen Max-Pooling Parameter* $\mathbf{n}$, falls eine Funktion
$$z : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to [0,1]^{\{1,\dots,d_1(l)\}\times\{1,\dots,d_2(l)\}}$$
existiert, die einem hierarchischen Modell vom Level l mit Feature Bedingung $\mathbf{b}$ und lokalem Max-Pooling Parameter $\mathbf{n}$ genügen, sodass
$$m(\mathbf{x}) = \max_{(i,j)\in\{1,\dots,d_1(l)\}\times\{1,\dots,d_2(l)\}} \big(z(\mathbf{x})\big)_{(i,j)}.$$

c) Es sei $p = q+s$ für ein $q \in \mathbb{N}_0$, $s \in (0,1]$ und es sei $C > 0$. Wir sagen, dass ein hierarchisches Max-Pooling Modell mit zusätzlichem lokalen Max-Pooling die *Glattheitsbedingung* p hat, falls alle Funktionen $g_{k,s}$ in dessen hierarchischem Modell (p, C)–glatt sind.

Bemerkung 2.3. Definition (2.6) der Dimensionen $d_1(k)$ und $d_2(k)$ der Feature Maps $y_{k,s}$ lässt sich wie folgt begründen: Durch das lokale Max-Pooling wird zunächst die Auflösung der Feature Maps $y_{k-1,s}$ um den Faktor $1/n_{k-1}$ verringert und es ergeben sich die Feature Maps $z_{k-1,s}$ mit den Dimensionen $\lceil d_1(k-1)/n_{k-1} \rceil$ sowie $\lceil d_2(k-1)/n_{k-1} \rceil$ (siehe Abbildung 2.2). Da anschließend durch das hierarchische Modell Einträge der Feature Maps $z_{k-1,s}$ mit dem Abstand δ_{k-1} zu Merkmalen größerer Teilbereiche kombiniert werden, haben die Feature Maps $y_{k,s}$ um den Abstand δ_{k-1} weniger Einträge als die Feature Maps $z_{k-1,s}$ (siehe Abbildung 2.3). Somit ergeben sich die obigen Dimensionen $d_1(k)$ und $d_2(k)$ aus Definition (2.6). Außerdem können wir durch eine rekursive Definition der Dimensionen $\tilde{d}_1(k) = d_1 - 2^k + 1$ und $\tilde{d}_2(k) = d_2 - 2^k + 1$ aus Gleichung (2.2) einen Bezug zu der Definition der Dimensionen $d_1(k)$ und $d_2(k)$ herstellen. Für $k = 1, \dots, l$ gilt nämlich

$$\tilde{d}_1(k) = \tilde{d}_1(k-1) - 2^{k-1} \quad \text{sowie} \quad \tilde{d}_1(k) = \tilde{d}_1(k-1) - 2^{k-1},$$

mit $\tilde{d}_1(0) = d_1$ und $\tilde{d}_2(0) = d_2$. Verringern wir nun die Auflösung der Feature Maps des Levels $k-1$ um den Faktor $1/n_{k-1}$ und verwenden die angepassten Abstände δ_{k-1} statt der Werte von 2^{k-1} ergeben sich die angepassten Dimensionen $d_1(k)$ und $d_2(k)$.

Bemerkung 2.4. Das hierarchische Max-Pooling Modell vom Level l aus Definition 3 c) ist ein Spezialfall des obigen Modells wenn wir $n_1 = n_2 = \cdots = n_{l-1} = 1$, $r_i(k,s) = 4 \cdot (s-1) + i$ für $i = 1, \dots, 4$ wählen und $b_k = 4^{l-k}$ für $k \in \{1, \dots, l\}$ setzen.

Bemerkung 2.5. Die Bedingung an die Nachbarschaftsgrößen (2.4) stellt sicher, dass die Nachbarschaften im Verhältnis zu den zugrunde liegenden Teilbereichen des Bildes nicht zu groß werden. Die zweite Bedingung (2.5) stellt sicher, dass die rekursiv definierten Dimensionen $d_1(k)$ und $d_2(k)$ für alle $k \in \{1, \dots, l\}$ größer als 0 sind.

Bemerkung 2.6. Da wir $n_l = 1$ gesetzt haben, gilt $z_{l,1} = y_{l,1}$ in Teil a).

2.3. Rotationssymmetrisches hierarchisches Max-Pooling Modell

In diesem Abschnitt wollen wir ein Modell für die funktionale a-posteriori Wahrscheinlichkeit (1.28) einführen, das neben den bisherigen Überlegungen des hierarchischen Max-Pooling Modells aus Definition 3 a) und b) auch die folgende Beobachtung der Bildklassifikation mit einbezieht:

(B5) Bei einigen Bildklassifikationsproblemen ist die Rotation von Objekten um beliebige Winkel irrelevant für eine korrekte Klassifizierung.

In Abbildung 2.4 würden wir beispielsweise allen drei Bildern die Klasse „Hund“ zuordnen, obwohl das Objekt Hund jeweils um unterschiedliche Winkel rotiert dargestellt ist. Praktische Anwendungen, bei denen Objekte einer Klasse beliebig rotiert vorliegen, finden sich zum Beispiel bei medizinischen Diagnoseverfahren, bei denen Röntgen-Scans von Gewebe ausgewertet werden (siehe Veeling et al. (2018)), oder bei der Untersuchung von Bildern von Galaxien (siehe Willett et al. (2013)). Für weitere Anwendungen siehe Delchevalerie et al. (2021) und die darin zitierte Literatur.

Die Idee, wie wir diesen Aspekt der Bildklassifikation in ein Modell für die funktionale a-posteriori Wahrscheinlichkeit (1.28) aufnehmen können, lässt sich wie folgt beschreiben: Wir nehmen an, dass eine Funktion $f : [0,1]^{C_h} \to [0,1]$ existiert, die für einen Teilbereich der Breite $h > 0$ eines (kontinuierlichen) Bildes die

Abbildung 2.4.: Irrelevanz der Rotation von Objekten.

Wahrscheinlichkeit berechnet, dass ein bestimmtes Objekt in dem Teilbereich in einer nicht rotierten Version dargestellt wird. Die Idee, in einem Teilbereich auch rotierte Objekte zu erkennen, besteht nun darin, in die Funktion f den Teilbereich auch um alle möglichen Winkel rotiert einzusetzen und das Supremum der entsprechenden Funktionswerte über allen Winkeln aus dem Intervall $[0, 2\pi]$ zu berechnen (vgl. Definition 5 a)). Auch diese Annahme ist durch das Vorgehen eines Menschen motiviert. Denn dieser kann ein Bild so lange rotieren, bis er das entsprechende Objekt in die „bekannte“ nicht-rotierte Form gebracht hat und daraufhin erkennt.

Teilbereiche von Bildern rotieren wir durch die in Abschnitt 1.6 eingeführte Rotationsfunktion $rot^{(\alpha)} : \mathbb{R}^2 \to \mathbb{R}^2$, die einen zweidimensionalen Vektor gegen den Uhrzeigersinn mit dem Winkel $\alpha \in [0, 2\pi]$ um den Ursprung $\mathbf{0} \in \mathbb{R}^2$ rotiert. Um verschiedene Bildpositionen betrachten zu können, verwenden wir die ebenfalls in Abschnitt 1.6 eingeführte Translation $\tau_{\mathbf{v}}(\mathbf{x}) = \mathbf{x} + \mathbf{v}$ um den Vektor $\mathbf{v} \in \mathbb{R}^2$. Ein mit dem Uhrzeigersinn um den Winkel $\alpha \in \mathbb{R}$ rotierter Teilbereich eines Bildes $\phi \in [0,1]^{C_1}$ mit der Breite $0 < h \leq 1/\sqrt{2}$ an der Bildposition $\mathbf{v} \in [-1/2 + h/\sqrt{2}, 1/2 - h/\sqrt{2}]^2$ ergab sich dann durch die Funktion

$$\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}\big|_{C_h} \in [0,1]^{C_h}$$

(für eine Darstellung siehe Abbildung 2.5). Wir müssen hier $h \leq 1/\sqrt{2}$ und $\mathbf{v} \in [-1/2 + h/\sqrt{2}, 1/2 - h/\sqrt{2}]^2$

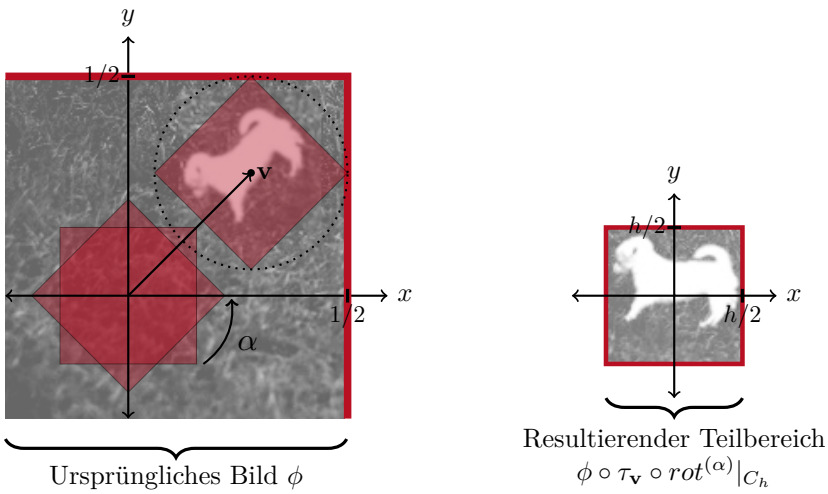


Abbildung 2.5.: Darstellung eines rotierten Teilbereichs.

fordern, um sicherzustellen, dass die Funktion $\tau_{\mathbf{v}} \circ rot^{(\alpha)}\big|_{C_h}$ für alle Winkel α in den Bildbereich C_1 abbildet. Ein nicht rotierter Teilbereich der Breite $0 < h' \leq h$ eines Bildes $\phi \in [0,1]^{C_h}$ ist dann durch $\phi \circ \tau_{\mathbf{v}}\big|_{C_{h'}} \in [0,1]^{C_{h'}}$ für ein $\mathbf{v} \in \mathbb{R}^2$ mit $\mathbf{v} + C_{h'} \subseteq C_h$ gegeben. Wir können ein Bild $\phi \in [0,1]^{C_h}$ der Breite h daher durch die Bilder

$$\phi \circ \tau_{(-h/4,-h/4)}\big|_{C_{h/2}}, \phi \circ \tau_{(h/4,-h/4)}\big|_{C_{h/2}}, \phi \circ \tau_{(-h/4,h/4)}\big|_{C_{h/2}}, \phi \circ \tau_{(h/4,h/4)}\big|_{C_{h/2}} \in [0,1]^{C_{h/2}}$$

der Breite $h/2$ in vier benachbarte kleinere Teilbereiche unterteilen (für eine Darstellung siehe Abbildung 2.6). Kombinieren wir nun die obige Idee mit den Überlegungen des hierarchischen Max-Pooling Modells aus

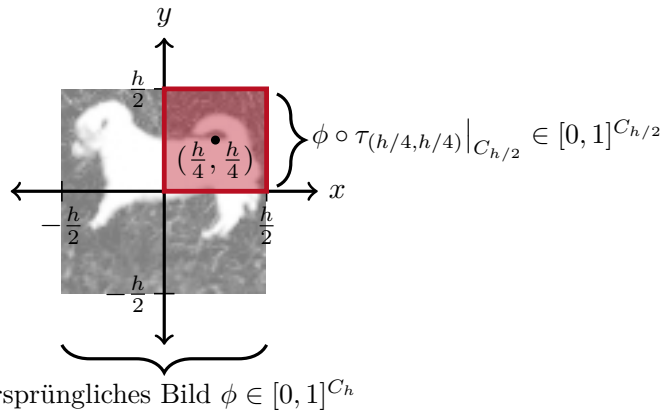


Abbildung 2.6.: Darstellung zur Unterteilung eines Teilbereichs in vier benachbarte kleinere Teilbereiche.

Definition 3 a) und b), ergibt sich das folgende Modell für die funktionale a-posteriori Wahrscheinlichkeit:

Definition 5. Es sei $m : [0,1]^{C_1} \to [0,1]$.

a) Es sei $0 < h \leq 1/\sqrt{2}$ und

$$h/\sqrt{2} \leq b \leq 1/2. \tag{2.7}$$

Wir sagen m genügt einem *rotationsymmetrischen Max-Pooling Modell der Breite h und dem Randabstand b*, falls eine Funktion $f : [0,1]^{C_h} \to [0,1]$ existiert, sodass

$$m(\phi) = \sup_{\mathbf{v} \in [-\frac{1}{2}+b, \frac{1}{2}-b]^2} \sup_{\alpha \in [0,2\pi]} f\left(\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}|_{C_h}\right) \quad \left(\phi \in [0,1]^{C_1}\right).$$

b) Es sei $l \in \mathbb{N}$, $h > 0$ und wir setzen $h_k = h/2^{l-k}$ für $k \in \{-1, 0, 1, \ldots, l\}$. Wir sagen $f : [0,1]^{C_h} \to [0,1]$ genügt einem *hierarchischen Modell vom Level l*, falls Funktionen

$$g_{k,s} : \mathbb{R}^4 \to [0,1] \quad (k = 1, \ldots, l, s = 1, \ldots, 4^{l-k})$$

und Funktionen

$$f_{0,s} : [0,1]^{C_{h_0}} \to [0,1] \quad (s = 1, \ldots, 4^l)$$

existieren, sodass

$$f = f_{l,1}$$

für Funktionen $f_{k,s} : [0,1]^{C_{h_k}} \to \mathbb{R}$, die wie folgt rekursiv definiert sind:

$$\begin{aligned} f_{k,s}(\phi) = g_{k,s}\Big(& f_{k-1,4\cdot(s-1)+1}\big(\phi \circ \tau_{(-h_{k-2},-h_{k-2})}|_{C_{h_{k-1}}}\big), \\ & f_{k-1,4\cdot(s-1)+2}\big(\phi \circ \tau_{(h_{k-2},-h_{k-2})}|_{C_{h_{k-1}}}\big), \\ & f_{k-1,4\cdot(s-1)+3}\big(\phi \circ \tau_{(-h_{k-2},h_{k-2})}|_{C_{h_{k-1}}}\big), \\ & f_{k-1,4\cdot s}\big(\phi \circ \tau_{(h_{k-2},h_{k-2})}|_{C_{h_{k-1}}}\big)\Big) \\ & \qquad \left(\phi \in [0,1]^{C_{h_k}}\right) \end{aligned}$$

für $k = 1, \ldots, l$ und $s = 1, \ldots, 4^{l-k}$.

c) Wir sagen m genügt einem *rotationssymmetrischen hierarchischen Max-Pooling Modell vom Level l, der Breite h und dem Randabstand b,* falls m einem rotationssymmetrischen Max-Pooling Modell der Breite h und dem Randabstand b genügt und die Funktion $f : [0,1]^{C_h} \to [0,1]$ in der Definition des rotationssymmetrischen Max-Pooling Modells einem hierarchischen Modell vom Level l genügt.

d) Es sei $p = q + s$ für ein $q \in \mathbb{N}_0$, $s \in (0,1]$ und es sei $C > 0$. Wir sagen, dass ein rotationssymmetrisches hierarchisches Max-Pooling Modell die *Glattheitsbedingung* p hat, falls alle Funktionen $g_{k,s}$ in dessen hierarchischem Modell (p, C)–glatt sind.

Bemerkung 2.7. Bedingung (2.7) an den Randabstand b stellt zum einen sicher, dass die in Teil a) betrachteten Teilbereiche innerhalb des Bildbereichs liegen und zum anderen, dass die Menge $[-1/2 + b, 1/2 - b]^2$ der Positionen der Teilbereiche nicht leer ist.

Da wir die Performanz von Bildklassifikatoren untersuchen, welche die Klasse von diskretisierten Bildern schätzen (siehe Gleichung (1.27)), werden wir im Beweis des Resultats zur Konvergenzgeschwindigkeit die funktionale a-posteriori Wahrscheinlichkeit durch ein faltendes neuronales Netz approximieren, in welches wir lediglich die Diskretisierung des kontinuierlichen Bildes einsetzen (für das entsprechende Approximationsresultat siehe Lemma 11). Für ein solches Approximationsresultat benötigen wir zwei weitere Annahmen an die Verteilung von (Φ, Y). Um diese Annahmen einzuführen, nehmen wir an, dass die funktionale a-posteriori Wahrscheinlichkeit $\eta_\Phi(\phi) = \mathbf{P}\{Y = 1 | \Phi = \phi\}$ einem rotationssymmetrischen hierarchischen Max-Pooling Modell vom Level l und Breite h mit Glattheitsbedingung p genügt. Es seien $f_{0,s} : [0,1]^{C_{h_0}} \to [0,1]$ $(s = 1, \dots, 4^l)$ die Funktionen des hierarchischen Modells von η_Φ, wobei $h_0 = h/2^l$. Die erste Annahme ist eine Glattheitsannahme an die Funktionen $f_{0,s}$, wenn wir diese auf konstante Bilder anwenden. Um ein konstantes Bild mathematisch zu beschreiben, sei $1\big|_A : A \to \mathbb{R}$ die konstante Funktion auf $A \subseteq \mathbb{R}^2$ mit Wert Eins.

Annahme 1. *Für alle $s \in \{1, \dots, 4^l\}$ existiert eine (p, C)-glatte Funktion $g_{0,s} : \mathbb{R} \to [0,1]$, sodass*

$$g_{0,s}(x) = f_{0,s}\left(x \cdot 1\big|_{C_{h_0}}\right)$$

für alle $x \in [0,1]$ gilt. Im Folgenden sei $\lambda \in \mathbb{N}$ die Auflösung der Beobachtungen (1.27). In der zweiten Annahme beschränken wir den Fehler der auftritt, wenn wir die Eingabe der Funktionen $f_{0,s}$, welche einem möglicherweise rotierten Teilbereich eines Bildes $\phi \in [0,1]^{C_1}$ entspricht, durch ein konstantes Bild ersetzen, dessen Graustufenwert aus einer lokalen Nachbarschaft des entsprechenden Teilbereichs gewählt wird. Die Größe des Teilbereichs und die Größe der Nachbarschaft des Teilbereichs hängen beide von der Auflösung λ ab, wie in Abbildung 2.7 dargestellt wird.

Annahme 2. *Es existiert ein messbares $A \subset [0,1]^{C_1}$ mit $P_\Phi(A) = 1$, $\epsilon_\lambda \in [0,1]$ und ein Skalierungsfaktor $c > 1$ mit $h_0 \leq \min\{(c \cdot \sqrt{2})/\lambda, 1/\sqrt{2}\}$, sodass für alle $\phi \in A$, $\mathbf{v} \in [h_0/\sqrt{2} - 1/2, 1/2 - h_0/\sqrt{2}]^2$, $\alpha \in [0, 2\pi]$ und $s \in \{1, \dots, 4^l\}$:*

$$\sup_{\mathbf{z} \in C_1 \,:\, \|\mathbf{v}-\mathbf{z}\|_\infty \leq \frac{c}{\lambda}} \left| f_{0,s}\Big(\underbrace{\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}\big|_{C_{h_0}}}_{\text{Teilbereich von } \phi \text{ mit Zentrum } \mathbf{v}} \Big) - f_{0,s}\Big(\phi(\mathbf{z}) \cdot 1\big|_{C_{h_0}}\Big) \right| \leq \epsilon_\lambda.$$

Bemerkung 2.8. Die Bedingung $h_0 \leq (c \cdot \sqrt{2})/\lambda$ stellt sicher, dass der Teilbereich mit Breite h_0 in der entsprechenden Nachbarschaft enthalten ist. Wie in Abbildung 2.7 dargestellt wird, werden für einen kleinen Skalierungsfaktor c Teilbereiche betrachtet, deren Größe ungefähr im Bereich der Inversen der Bildauflösung λ^{-1} liegt.

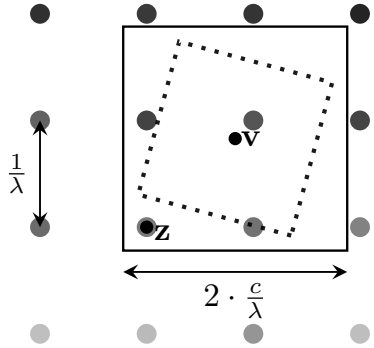


Abbildung 2.7.: Darstellung zu Annahme 2 mit $c = 1.05$ und $h_0 = (c \cdot \sqrt{2})/\lambda$.

In Abbildung 2.7 wird ein möglicher Teilbereich mit Zentrum $\mathbf{v}$ und ein möglicher Punkt $\mathbf{z}$ aus Annahme 2 dargestellt. Im Hintergrund sind mögliche Pixelwerte auf dem Gitter $G_\lambda \subset C_1$ eingezeichnet. Als Nächstes führen wir noch ein Beispiel an, um zu sehen, dass Annahme 2 realistisch für kleine $\epsilon_\lambda \in [0, 1]$ ist.

Beispiel 2.1. Wir nehmen an, dass eine Auflösung $\lambda_{max} \in \mathbb{N}$ mit $\lambda_{max} \geq 2$ existiert, sodass ein (kontinuierliches) Bild eindeutig durch Pixelwerte auf dem Gitter

$$H_{\lambda_{max}} = \left\{ \left(\frac{i}{\lambda_{max} - 1} - \frac{1}{2}, \frac{j}{\lambda_{max} - 1} - \frac{1}{2} \right) \; : \; i, j \in \{0, \dots, \lambda_{max} - 1\} \right\}.$$

definiert ist. Diese Annahme lässt sich dadurch motivieren, dass Menschen ein begrenztes Auflösungsvermögen besitzen (siehe z.B. Gimel'farb und Delmas (2018))) und ihnen Bilder damit immer nur mit einer maximalen Auflösung vorliegen, sie aber trotzdem gut darin sind, Bilder zu klassifizieren. Wir nehmen daher an, dass $P_\Phi(A) = 1$ für eine messbare Menge

$$A \subseteq \{\phi_{\mathbf{x}} \in [0, 1]^{C_1} : \mathbf{x} \in [0, 1]^{H_{\lambda_{max}}}\} \subset [0, 1]^{C_1}$$

gilt, wobei das Bild $\phi_{\mathbf{x}} : C_1 \to [0, 1]$ der bilinearen Interpolation entspricht, welche für $\mathbf{x} \in [0, 1]^{H_{\lambda_{max}}}$ wie folgt definiert ist: Für $\mathbf{v} = (v_1, v_2) \in C_1$ seien

$$(a_1^{(\mathbf{v})}, b_1^{(\mathbf{v})}), (a_2^{(\mathbf{v})}, b_1^{(\mathbf{v})}), (a_1^{(\mathbf{v})}, b_2^{(\mathbf{v})}), (a_2^{(\mathbf{v})}, b_2^{(\mathbf{v})}) \in H_{\lambda_{max}} \tag{2.8}$$

Gitter- bzw. die Eckpunkte eines Quaders der Breite $1/(\lambda_{max} - 1)$, sodass $\mathbf{v} \in [a_1^{(\mathbf{v})}, a_2^{(\mathbf{v})}] \times [b_1^{(\mathbf{v})}, b_2^{(\mathbf{v})}]$ und

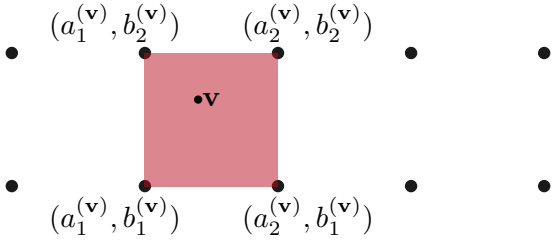


Abbildung 2.8.: Darstellung der Gitterpunkte zur Berechnung der Interpolation $\phi_{\mathbf{x}}(\mathbf{v})$.

$|a_1^{(\mathbf{v})} - a_2^{(\mathbf{v})}| = |b_1^{(\mathbf{v})} - b_2^{(\mathbf{v})}| = 1/(\lambda_{max} - 1)$ (siehe Abbildung 2.8 für eine Darstellung der Gitterpunkte (2.8)). Wir definieren die Koeffizienten

$$k_0^{(\mathbf{v})}, k_1^{(\mathbf{v})}, k_2^{(\mathbf{v})}, k_3^{(\mathbf{v})} \in \mathbb{R} \tag{2.9}$$

durch

$$\begin{pmatrix} k_0^{(\mathbf{v})} \\ k_1^{(\mathbf{v})} \\ k_2^{(\mathbf{v})} \\ k_3^{(\mathbf{v})} \end{pmatrix} = (\lambda_{max} - 1)^2 \cdot \begin{pmatrix} a_2^{(\mathbf{v})} \cdot b_2^{(\mathbf{v})} & -a_2^{(\mathbf{v})} \cdot b_1^{(\mathbf{v})} & -a_1^{(\mathbf{v})} \cdot b_2^{(\mathbf{v})} & a_1^{(\mathbf{v})} \cdot b_1^{(\mathbf{v})} \\ -b_2^{(\mathbf{v})} & b_1^{(\mathbf{v})} & b_2^{(\mathbf{v})} & -b_1^{(\mathbf{v})} \\ -a_2^{(\mathbf{v})} & a_2^{(\mathbf{v})} & a_1^{(\mathbf{v})} & -a_1^{(\mathbf{v})} \\ 1 & -1 & -1 & 1 \end{pmatrix} \begin{pmatrix} x_{(a_1^{(\mathbf{v})}, b_1^{(\mathbf{v})})} \\ x_{(a_1^{(\mathbf{v})}, b_2^{(\mathbf{v})})} \\ x_{(a_2^{(\mathbf{v})}, b_1^{(\mathbf{v})})} \\ x_{(a_2^{(\mathbf{v})}, b_2^{(\mathbf{v})})} \end{pmatrix}.$$

Die bilineare Interpolation von $\phi_{\mathbf{x}}$ ist dann definiert durch

$$\phi_{\mathbf{x}}(\mathbf{v}) = k_0^{(\mathbf{v})} + k_1^{(\mathbf{v})} \cdot v_1 + k_2^{(\mathbf{v})} \cdot v_2 + k_3^{(\mathbf{v})} \cdot v_1 \cdot v_2 \quad (\mathbf{v} = (v_1, v_2) \in C_1)$$

(siehe Abbildung 2.9 für die Darstellung einer bilinearen Interpolation und siehe beispielsweise Kirkland (2010) für die Herleitung der obigen Formel). Außerdem nehmen wir an, dass die Funktionen $f_{0,s} : [0,1]^{C_{h_0}} \to [0,1]$,

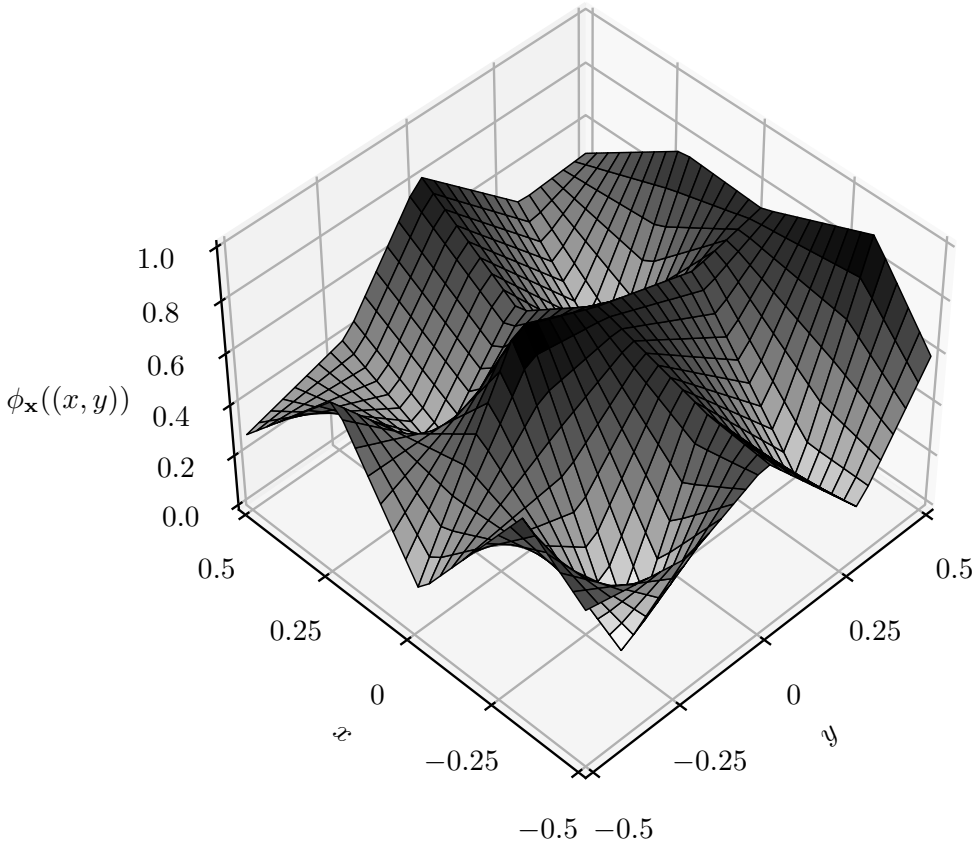


Abbildung 2.9.: Beispiel einer bilinearen Interpolation $\phi_{\mathbf{x}}$ für $\mathbf{x} \in [0,1]^{H_5}$.

welche die Entscheidungen für die kleinsten Teilbereiche berechnen, durch

$$f_{0,s}(\phi) = \sup_{\mathbf{x} \in C_{h_0}} \phi(\mathbf{x}) \quad \left(\phi \in [0,1]^{C_{h_0}}\right)$$

für alle $s \in \{1, \ldots, 4^l\}$ definiert sind (die Funktionen $f_{0,s}$ $(s = 1, \ldots, 4^l)$ hängen damit nicht von s ab). Es gilt dann insbesondere

$$f_{0,s}\left(x \cdot 1\big|_{C_{h_0}}\right) = x$$

für alle $x \in [0,1]$, womit Annahme 1 mit der Funktion $g_{0,s}(x) = x$ für beliebige Glattheiten p erfüllt ist. Unter den Bedingungen von Annahme 2 gilt dann für $\lambda \geq 32 \cdot c \cdot \lambda_{max}^2$

$$\sup_{\mathbf{z} \in C_1 \,:\, \|\mathbf{v}-\mathbf{z}\|_\infty \leq \frac{c}{\lambda}} \left| f_{0,s}\Big(\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}\big|_{C_{h_0}}\Big) - f_{0,s}\Big(\phi(\mathbf{z}) \cdot 1\big|_{C_{h_0}}\Big) \right| \leq \frac{32 \cdot c \cdot \lambda_{max}^2}{\lambda} =: \epsilon_\lambda. \tag{2.10}$$

Der Fehlerterm ϵ_λ ist unter diesen Annahmen daher für entsprechend große Auflösungen λ beliebig klein. Der Beweis von Ungleichung (2.10) befindet sich in Abschnitt A.2 des Anhangs.

3. Konvergenzverhalten von Bildklassifikatoren basierend auf faltenden neuronalen Netzen

In diesem Kapitel werden in Abschnitt 3.1 verschiedene Bildklassifikatoren, die auf faltenden neuronalen Netzen basieren, vorgestellt. Im anschließenden Abschnitt werden unter den Annahmen der Modelle aus Kapitel 2 drei Hauptresultate zur Konvergenzgeschwindigkeit für die eingeführten Klassifikatoren formuliert. Im weiteren Verlauf von Kapitel 3 werden die entsprechenden Resultate bewiesen.

3.1. Definition der Schätzer

Zunächst wiederholen wir die bereits in Abschnitt 1.3 eingeführten Komponenten von faltenden neuronalen Netzen (siehe daher Abschnitt 1.3 für eine detailliertere Einführung). Wir verwenden für alle Netzwerkarchitekturen die ReLU-Aktivierungsfunktion $\sigma : \mathbb{R} \to \mathbb{R}_0^+$, die durch

$$\sigma(x) = \max\{x, 0\} \quad \big(x \in \mathbb{R}\big)$$

definiert ist. Ein vollverbundenes neuronales Netz mit $L_{net} \in \mathbb{N}$ verdeckten Schichten und $k_r \in \mathbb{N}$ Neuronen in Schicht $r \in \{1, \dots, L_{net}\}$ entspricht einer Funktion $g : \mathbb{R}^d \to \mathbb{R}$ der Form

$$g(\mathbf{x}) = \sum_{i=1}^{k_{L_{net}}} w_i^{(L_{net})} \cdot g_i^{(L_{net})}(\mathbf{x}) + w_0^{(L_{net})} \quad \big(\mathbf{x} \in \mathbb{R}^d\big), \tag{3.1}$$

wobei $w_0^{(L_{net})}, \dots, w_{k_{L_{net}}}^{(L_{net})} \in \mathbb{R}$ die äußeren Gewichte bezeichnen und die Funktionen $g_i^{(L_{net})}$ $(i = 1, \dots, k_{L_{net}})$ rekursiv definiert sind durch

$$g_i^{(r)}(\mathbf{x}) = \sigma\left(\sum_{j=1}^{k_{r-1}} w_{i,j}^{(r-1)} \cdot g_j^{(r-1)}(\mathbf{x}) + w_{i,0}^{(r-1)}\right) \quad \big(\mathbf{x} \in \mathbb{R}^d\big)$$

für $i \in \{1, \dots, k_r\}$, $r \in \{1, \dots, L_{net}\}$, $k_0 = d$, die inneren Gewichte $w_{i,0}^{(r-1)}, \dots, w_{i,k_{r-1}}^{(r-1)} \in \mathbb{R}$ und

$$g_i^{(0)}(\mathbf{x}) = x_i \qquad \big(\mathbf{x} = (x_1, \dots, x_d)^T \in \mathbb{R}^d\big)$$

für $i \in \{1, \dots, k_0\}$. Für die Definition einer faltenden Schicht und den Definitionen weiterer Komponenten sei $k' \in \mathbb{N}$ die Anzahl der Eingabekanäle, $k \in \mathbb{N}$ die Anzahl an Ausgabekanälen, $I = \{1, \dots, i_1\} \times \{1, \dots, i_2\}$ die Indexmenge der Neuronen eines Kanals mit $i_1, i_2 \in \mathbb{N}$, $M \in \mathbb{N}$ die Filtergröße sowie $P \in \{1, \dots, M\}$ der Zero-Padding Parameter. Die trainierbaren Gewichte einer faltenden Schicht setzen sich aus den Filtern und Bias-Termen zusammen, die wir in dem Gewichtsvektor

$$\mathbf{w} = \Big(\big(w_{i,j,s_1,s_2}\big)_{1 \le i,j \le M, s_1 \in \{1,\dots,k'\}, s_2 \in \{1,\dots,k\}}, \big(w_{s_2}\big)_{s_2 \in \{1,\dots,k\}}\Big)$$

zusammenfassen. Eine faltende Schicht ist dann eine Funktion

$$o_{(k',k),M,P,\mathbf{w}} : \mathbb{R}^{I\times\{1,\dots,k'\}} \to \mathbb{R}_0^{+\,I\times\{1,\dots,k\}}$$

der Form

$$\left(o_{(k',k),M,P,\mathbf{w}}(\mathbf{x})\right)_{(i,j),s_2} = \sigma\left(\sum_{s_1=1}^{k'} \sum_{\substack{t_1,t_2\in\{1,\dots,M\}\\(i+t_1-P,j+t_2-P)\in I}} w_{t_1,t_2,s_1,s_2}\cdot x_{(i+t_1-P,j+t_2-P),s_1} + w_{s_2}\right) \quad \left(\mathbf{x}\in\mathbb{R}^{I\times\{1,\dots,k'\}}\right) \tag{3.2}$$

für $(i,j)\in I$ und $s_2\in\{1,\dots,k\}$. Im Fall eines einzigen Eingabekanals ($k'=1$) identifizieren wir $\mathbb{R}^{I\times\{1\}}$ mit $\mathbb{R}^I$ und definieren eine faltende Schicht durch eine Funktion

$$o_{(1,k),M,P,\mathbf{w}} : \mathbb{R}^I \to \mathbb{R}_0^{+\,I\times\{1,\dots,k\}},$$

welche durch

$$\left(o_{(1,k),M,P,\mathbf{w}}(\mathbf{x})\right)_{(i,j),s_2} = \sigma\left(\sum_{\substack{t_1,t_2\in\{1,\dots,M\}\\(i+t_1-P,j+t_2-P)\in I}} w_{t_1,t_2,1,s_2}\cdot x_{(i+t_1-P,j+t_2-P)} + w_{s_2}\right) \quad \left(\mathbf{x}\in\mathbb{R}^I\right)$$

für $(i,j)\in I$ und $s_2\in\{1,\dots,k\}$ definiert ist. Als nächsten definieren wir lokale Max-Pooling Schichten und Subsampling Schichten. Dies sind jeweils Funktionen

$$f : \mathbb{R}^{I\times\{1,\dots,k\}} \to \mathbb{R}^{\left\{1,\dots,\left\lceil\frac{i_1}{s}\right\rceil\right\}\times\left\{1,\dots,\left\lceil\frac{i_2}{s}\right\rceil\right\}\times\{1,\dots,k\}},$$

die von einem Parameter $s\in\mathbb{N}$ abhängen. Eine lokale Max-Pooling Schicht hat die Form

$$\left(f_{max}^{(s)}(\mathbf{x})\right)_{(i,j),s_2} = \max_{(j_1,j_2)\in\left(\{(i-1)\cdot s+1\dots,i\cdot s\}\times\{(j-1)\cdot s+1\dots,j\cdot s\}\right)\cap I} x_{(j_1,j_2),s_2} \quad \left(\mathbf{x}\in\mathbb{R}^{I\times\{1,\dots,k\}}\right) \tag{3.3}$$

für $(i,j)\in\{1,\dots,\lceil i_1/s\rceil\}\times\{1,\dots,\lceil i_2/s\rceil\}$ sowie $s_2\in\{1,\dots,k\}$ und eine Subsampling Schicht hat die Form

$$\left(f_{sub}^{(s)}(\mathbf{x})\right)_{(i,j),s_2} = x_{((i-1)\cdot s+1,(j-1)\cdot s+1),s_2} \quad \left(\mathbf{x}\in\mathbb{R}^{I\times\{1,\dots,k\}}\right) \tag{3.4}$$

für $(i,j)\in\{1,\dots,\lceil i_1/s\rceil\}\times\{1,\dots,\lceil i_2/s\rceil\}$ sowie $s_2\in\{1,\dots,k\}$. Eine Ausgabeschicht $f_{out}:\mathbb{R}^{I\times\{1,\dots,k\}}\to\mathbb{R}$ eines faltenden neuronalen Netzes hängt von den trainierbaren Gewichten $\mathbf{w}_{out}=(w_s)_{s\in\{1,\dots,k\}}$ sowie den Ausgabeschranken $\mathbf{A}=(A_1,A_1',A_2,A_2')\in\mathbb{N}^4$ ab, wobei $1\le A_j\le A_j'\le i_j$ für $j=1,2$ gilt, und ist durch

$$f_{out}^{(\mathbf{A})}(\mathbf{x}) = \max_{\substack{i\in\{A_1,\dots,A_1'\},\\ j\in\{A_2,\dots,A_2'\}}} \sum_{s_2=1}^{k} w_{s_2}\cdot x_{(i,j),s_2} \quad \left(\mathbf{x}\in\mathbb{R}^{I\times\{1,\dots,k\}}\right) \tag{3.5}$$

definiert.

Wir führen nun die verschiedenen Netzwerkarchitekturen faltender neuronaler Netze ein, die wir verwenden werden, um die in Kapitel 2 eingeführten Modelle zur Bildklassifikation zu schätzen. Anschließend definieren wir die entsprechenden Bildklassifikatoren als Plug-In Klassifikatoren. Wir führen zunächst eine einfache Architektur von faltenden neuronalen Netzen ein, die ohne lokale Pooling Schichten auskommen und in jeder

ihrer L Schichten $k \in \mathbb{N}$ Kanäle besitzen. Ein faltendes neuronales Netz der beschriebenen Bauart ist dann eine Funktion $f : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}$ der Form

$$f(\mathbf{x}) = f_{out}^{(\mathbf{A})} \circ o_{(k,k),M_L,P_L,\mathbf{w}_L} \circ \cdots \circ o_{(k,k),M_2,P_2,\mathbf{w}_2} \circ o_{(1,k),M_1,P_1,\mathbf{w}_1}(\mathbf{x}) \quad \big(\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}\big), \tag{3.6}$$

wobei die einzelnen faltenden Schichten $o_{(k'_r,k),M_r,P_r,\mathbf{w}_r}$ $(r = 1,\dots,L)$ und die Ausgabeschicht $f_{out}^{(\mathbf{A})}$ gemäß der Gleichungen (3.2) und (3.5) definiert sind. Für ein faltendes neuronales Netz der obigen Form setzen wir $\mathbf{M} = (M_1,\dots,M_L)$, $\mathbf{P} = (P_1,\dots,P_L)$ und bezeichnen mit

$$\mathcal{F}_{CNN}(\boldsymbol{\theta}) = \big\{f : f \text{ hat die Form (3.6) mit Parametern } \boldsymbol{\theta} = (L,k,\mathbf{M},\mathbf{P},\mathbf{A})\big\} \tag{3.7}$$

die Klasse all dieser faltenden neuronalen Netze mit dem Parametervektor $\boldsymbol{\theta}$. In dieser Arbeit verwenden wir zwei Netzwerkarchitekturen von faltenden neuronalen Netzen, bei denen $t \in \mathbb{N}$ faltende neuronale Netze der Form (3.6) parallel berechnet werden und auf deren t Ausgaben ein vollverbundenes neuronales Netz aus der Klasse

$$\mathcal{G}_t(L_{net}, r_{net}) = \big\{g_{net} \ : \ g_{net} : \mathbb{R}^t \to \mathbb{R} \text{ hat die Form (3.1) mit } k_1 = \cdots = k_{L_{net}} = r_{net}\big\} \tag{3.8}$$

angewendet wird:

$$\mathcal{F}_j(\boldsymbol{\theta}) = \big\{g_{net} \circ (f_1,\dots,f_t) : g \in \mathcal{G}_t(L_{net}, r_{net}), f_1,\dots,f_t \in \mathcal{F}_{CNN}((L,k,\mathbf{M},\mathbf{P}_j,\mathbf{A}))\big\} \tag{3.9}$$

für $j = 1,2$. Die beiden Netzwerkarchitekturen unterscheiden sich lediglich darin, dass in der ersten Architektur ein einseitiges Zero-Padding und in der zweiten Architektur ein symmetrisches Zero-Padding verwendet wird (siehe Abbildung 1.2 für eine Darstellung). Diese beiden unterschiedlichen Zero-Padding Methoden erleichtern das spätere Beweisvorgehen der jeweiligen Resultate. Die Zero-Padding Parameter $\mathbf{P}_1$ und $\mathbf{P}_2$ sind dementsprechend durch

$$\mathbf{P}_1 = (1,1,\dots,1) \quad \text{und} \quad \mathbf{P}_2 = (\lceil M_1/2\rceil, \lceil M_2/2\rceil, \dots, \lceil M_L/2\rceil) \tag{3.10}$$

definiert und die Funktionsklassen hängen dann beide von einem Parametervektor $\boldsymbol{\theta} = (t, L_{net}, r_{net}, L, k, \mathbf{M}, \mathbf{A})$ ab. Für eine Darstellung eines faltenden neuronalen Netzes aus den Klassen $\mathcal{F}_1(\boldsymbol{\theta})$ bzw. $\mathcal{F}_2(\boldsymbol{\theta})$ siehe Abbildung 3.1. Neben den in (3.9) eingeführten Netzwerkarchitekturen verwenden wir noch drei weitere Bauarten, welche zusätzliche Pooling Schichten enthalten. Die ersten beiden Architekturen unterscheiden sich von einem faltenden neuronalen Netz der Form (3.6) darin, dass sie nach jeweils $z \in \mathbb{N}$ faltenden Schichten eine lokale Max-Pooling bzw. Subsampling Schicht enthalten. Die so verwendeten $L-1$ lokalen Max-Pooling bzw. Subsampling Schichten hängen von einem Parametervektor $\mathbf{s} = (s_1,\dots,s_{L-1}) \in \mathbb{N}^{L-1}$ ab. Die beiden Architekturen haben die Formen

$$\begin{aligned} f(\mathbf{x}) = & f_{out}^{(\mathbf{A})} \circ o_{(k,k),M_L,1,\mathbf{w}_{L\cdot z}} \circ \cdots \circ o_{(k,k),M_L,1,\mathbf{w}_{(L-1)\cdot z+1}} \\ & \circ f_{max}^{(s_{L-1})} \circ o_{(k,k),M_{L-1},1,\mathbf{w}_{(L-1)\cdot z}} \circ \cdots \circ o_{(k,k),M_{L-1},1,\mathbf{w}_{(L-2)\cdot z+1}} \circ \dots \\ & \circ f_{max}^{(s_1)} \circ o_{(k,k),M_1,1,\mathbf{w}_z} \circ \cdots \circ o_{(1,k),M_1,1,\mathbf{w}_1}(\mathbf{x}) \qquad \big(\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}\big) \end{aligned} \tag{3.11}$$

und

$$\begin{aligned} f(\mathbf{x}) = & f_{out}^{(\mathbf{A})} \circ o_{(k,k),M_L,1,\mathbf{w}_{L\cdot z}} \circ \cdots \circ o_{(k,k),M_L,1,\mathbf{w}_{(L-1)\cdot z+1}} \\ & \circ f_{sub}^{(s_{L-1})} \circ o_{(k,k),M_{L-1},1,\mathbf{w}_{(L-1)\cdot z}} \circ \cdots \circ o_{(k,k),M_{L-1},1,\mathbf{w}_{(L-2)\cdot z+1}} \circ \dots \\ & \circ f_{sub}^{(s_1)} \circ o_{(k,k),M_1,1,\mathbf{w}_z} \circ \cdots \circ o_{(1,k),M_1,1,\mathbf{w}_1}(\mathbf{x}) \qquad \big(\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}\big), \end{aligned} \tag{3.12}$$

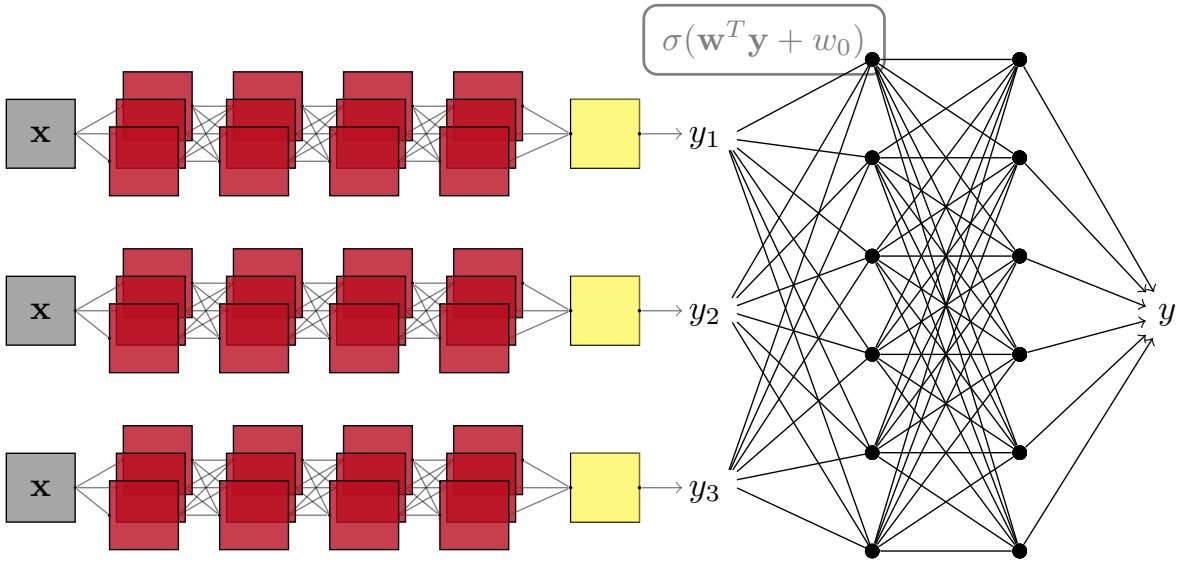


Abbildung 3.1.: Darstellung eines faltenden neuronalen Netzes der Klasse $\mathcal{F}_1(\boldsymbol{\theta})$ bzw. $\mathcal{F}_2(\boldsymbol{\theta})$ mit $t = 3$, $L = 4$, $k = 3$, $L_{net} = 2$ und $r_{net} = 6$.

wobei die Max-Pooling Schichten $f_{max}^{(s_r)}$ und die Subsampling Schichten $f_{sub}^{(s_r)}$ $(r = 1, \ldots, L-1)$ gemäß der Gleichungen (3.3) und (3.4) definiert sind. Da wir jeweils ein einseitiges Zero-Padding mit $P = 1$ verwendet haben, hängen die beiden Bauarten nur von den Parametern L, k, $\mathbf{M} = (M_1, \ldots, M_L)$, z, $\mathbf{s}$ und $\mathbf{A}$ ab. Wir bezeichnen alle Funktionen dieser Bauarten mit

$$\mathcal{F}_3(\boldsymbol{\theta}) = \big\{f : f \text{ hat die Form (3.11) mit Parametern } \boldsymbol{\theta} = (L, k, \mathbf{M}, z, \mathbf{s}, \mathbf{A})\big\}$$

beziehungsweise

$$\mathcal{F}_4(\boldsymbol{\theta}) = \big\{f : f \text{ hat die Form (3.12) mit Parametern } \boldsymbol{\theta} = (L, k, \mathbf{M}, z, \mathbf{s}, \mathbf{A})\big\}.$$

Für eine Illustration der beiden Netzwerkarchitekturen aus den Klassen $\mathcal{F}_3(\boldsymbol{\theta})$ und $\mathcal{F}_4(\boldsymbol{\theta})$ siehe Abbildung 3.2. Die letzte Netzwerkarchitektur, die wir hier einführen werden, besitzt lediglich eine Subsampling Schicht und

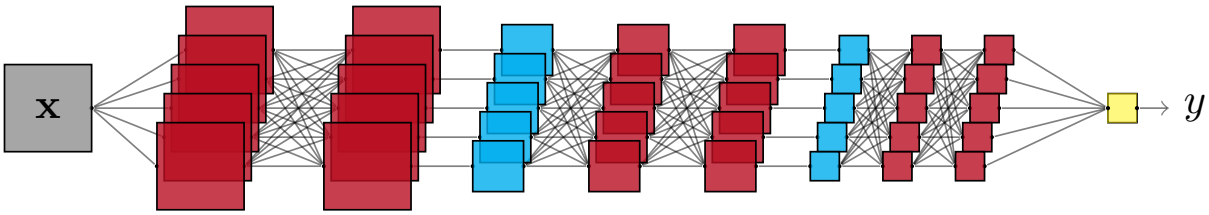


Abbildung 3.2.: Darstellung eines faltenden neuronalen Netzes der Klasse $\mathcal{F}_3(\boldsymbol{\theta})$ bzw. der Klasse $\mathcal{F}_4(\boldsymbol{\theta})$ mit $L = 3$, $k = 5$ und $z = 2$.

hat die Form

$$f(\mathbf{x}) = f_{out}^{(\mathbf{A})} \circ f_{sub}^{(s)} \circ o_{(k,k),M_L,1,\mathbf{w}_L} \circ \cdots \circ o_{(1,k),M_1,1,\mathbf{w}_1}(\mathbf{x}) \quad \left(\mathbf{x} \in [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}}\right) \tag{3.13}$$

und hängt von den Parametern L, k, $\mathbf{M} = (M_1, \ldots, M_L)$, s und $\mathbf{A}$ ab. Wir bezeichnen die entsprechende

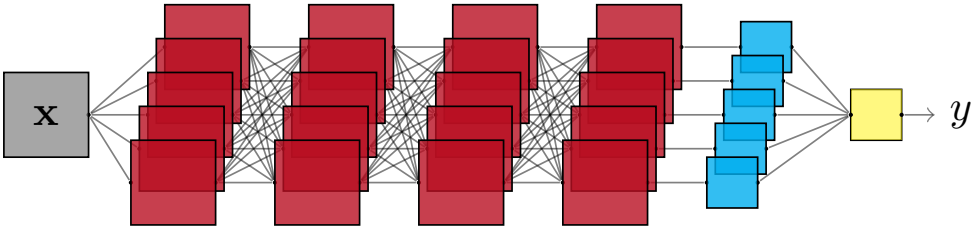


Abbildung 3.3.: Darstellung eines faltenden neuronalen Netzes der Klasse $\mathcal{F}_5(\boldsymbol{\theta})$ mit $L = 4$ und $k = 5$.

Funktionsklasse mit

$$\mathcal{F}_5(\boldsymbol{\theta}) = \big\{f : f \text{ hat die Form (3.13) mit Parametern } \boldsymbol{\theta} = (L, k, \mathbf{M}, s, \mathbf{A})\big\}.$$

Für eine Illustration der Netzwerkarchitektur der Klasse $\mathcal{F}_5(\boldsymbol{\theta})$ siehe Abbildung 3.3. Die entsprechenden Schätzer der a-posteriori Wahrscheinlichkeit definieren wir dann als Kleinste-Quadrate-Schätzer über den Funktionsklassen $\mathcal{F}_j(\boldsymbol{\theta}_j)$ $(j = 1, \ldots, 5)$ gemäß

$$\eta_n^{(j)} = \operatorname*{arg\,min}_{f \in \mathcal{F}_j(\boldsymbol{\theta}_j)} \frac{1}{n} \sum_{i=1}^{n} |Y_i - f(\mathbf{X}_i)|^2 \tag{3.14}$$

und die entsprechenden Bildklassifikatoren $f_n^{(j)}$ $(j = 1, \ldots, 5)$ durch

$$f_n^{(j)}(\mathbf{x}) = \begin{cases} 1, & \text{falls } \eta_n^{(j)}(\mathbf{x}) \geq \frac{1}{2} \\ 0, & \text{sonst.} \end{cases} \tag{3.15}$$

Wie bereits in Abschnitt 1.3 beschrieben, wird das Minimierungsproblem (3.14) bezüglich der reellwertigen trainierbaren Gewichte gelöst, während die Parameter $\boldsymbol{\theta}_j$ $(j = 1, \ldots, 5)$ fest sind. In Anwendungen können die Netzwerkparameter datenabhängig durch Unterteilung der Stichprobe gewählt werden (siehe beispielsweise Györfi et al. (2002)). Diese Methode wird in Kapitel 4 erklärt und für die dortigen Anwendungen verwendet. In den Resultaten zur Konvergenzgeschwindigkeit aus dem nächsten Abschnitt hängen manche Netzwerkparameter von Parametern der zugrunde liegenden Verteilung von $(\mathbf{X}, Y)$ ab. Hier können dann speziell die Parameter der Verteilung datenabhängig durch Unterteilung der Stichprobe gewählt werden, aus denen dann die Netzwerkparameter resultieren. Auch dieses Vorgehen wird in Abschnitt 4 Anwendung finden.

3.2. Hauptresultate zur Konvergenzgeschwindkeit

In den folgenden drei Abschnitten präsentieren wir die drei Hauptresultate zur Konvergenzgeschwindigkeit, welche jeweils eine obere Schranke der Differenz des erwarteten Missklassifikationsrisikos unserer Schätzung und dem optimalen Missklassifikationsrisiko darstellen.

3.2.1. Resultat im verallgemeinerten hierarchischen Max-Pooling Modell

Im ersten Hauptresultat wird eine Konvergenzrate des Fehlers $\mathbf{E}\{L(f_n^{(1)})\} - L(f^*)$ für den in Abschnitt 3.1 eingeführten Bildklassifikator $f_n^{(1)}$ basierend auf der Klasse $\mathcal{F}_1(\boldsymbol{\theta})$ hergeleitet. Hier wird angenommen, die a-posteriori Wahrscheinlichkeit genüge dem verallgemeinerten hierarchischen Max-Pooling Modell aus Definition 3 in Abschnitt 2.1.

Theorem 3.1. *Es seien $d_1, d_2 \in \mathbb{N}$ mit $d_1, d_2 > 1$ und $(\mathbf{X}, Y)$, $(\mathbf{X}_1, Y_1)$, …, $(\mathbf{X}_n, Y_n)$ seien unabhängig identisch verteilte $[0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \times \{0,1\}$-wertige Zufallsvariablen mit $n > 1$. Weiterhin genüge die a-posteriori Wahrscheinlichkeit $\eta(\mathbf{x}) = \mathbf{P}\{Y = 1 | \mathbf{X} = \mathbf{x}\}$ einem verallgemeinerten hierarchischen Max-Pooling Modell der Ordnung d^* und Level l mit Glattheitsbedingungen $p_1, p_2 \in [1, \infty)$. Für hinreichend große Konstanten $c_{10}, c_{11} > 0$ wähle*

$$L_n = \max\left\{\left\lceil c_{10} \cdot n^{\frac{4}{2\cdot(2\cdot p_1+4)}}\right\rceil, \left\lceil c_{10} \cdot n^{\frac{d^*}{2\cdot(2\cdot p_2+d^*)}}\right\rceil\right\}, \quad k = 2 \cdot 4^{l-1} + c_{11} \quad \textit{und} \quad r_{net} = c_{11}$$

sowie

$$L = l \cdot 4^{l-1} \cdot (L_n + 1), \quad L_{net} = L_n, \quad t = d^* \quad \textit{und} \quad \mathbf{A} = (1, d_1 - 2^l + 1, 1, d_2 - 2^l + 1).$$

Wähle außerdem

$$M_{(r-1)\cdot 4^{l-1}\cdot(L_n+1)+1} = 2^r, \dots, M_{r\cdot 4^{l-1}\cdot(L_n+1)} = 2^r \quad (r = 1, \dots, l)$$

und setze $\mathbf{M} = (M_1, \dots, M_L)$. Definiere $f_n^{(1)}$ als Plug-In Klassifikator gemäß der Gleichungen (3.14) *und* (3.15) *unter Verwendung des Parametervektors*

$$\boldsymbol{\theta} = (t, L_{net}, r_{net}, L, k, \mathbf{M}, \mathbf{A})$$

basierend auf der Funktionsklasse $\mathcal{F}_1(\boldsymbol{\theta})$. Es gilt dann

$$\mathbf{E}\{L(f_n^{(1)})\} - L(f^*) \leq c_{12} \cdot \sqrt{\log(d_1 \cdot d_2)} \cdot (\log n)^2 \cdot \max\left\{n^{-\frac{p_1}{2\cdot p_1+4}}, n^{-\frac{p_2}{2\cdot p_2+d^*}}\right\} \tag{3.16}$$

für eine Konstante $c_{12} > 0$, die nicht von d_1, d_2 und n abhängt.

Beweis. Der Beweis befindet sich in Abschnitt 3.6.1. Das allgemeine Beweisvorgehen wird in Abschnitt 3.3 beschrieben.

Bemerkung 3.1. Die Konvergenzrate $\max\left\{n^{-\frac{p_1}{2\cdot p_1+4}}, n^{-\frac{p_2}{2\cdot p_2+d^*}}\right\}$ (bis auf einen logarithmischen Faktor) in (3.16) hängt nicht von der Dimension $d_1 \cdot d_2$ von $\mathbf{X}$ ab. Daher ist unser Bildklassifikator in der Lage, den Fluch der hohen Dimension unter den obigen Struktur- und Glattheitsannahmen an die a-posteriori Wahrscheinlichkeit η zu umgehen. Wir haben uns hier auf die Konvergenzrate konzentriert und keinen Versuch unternommen, die Konstanten in den Definitionen von L, L_{net}, k und r_{net} so klein wie möglich zu wählen. Da die Konstante c_{12} in (3.16) polynomiell von 2^l abhängt, tritt die Dimension $d_1 \cdot d_2$ in (3.16) nur für den Fall logarithmisch auf, wenn 2^l sehr viel kleiner als die Dimensionen $d_1 \cdot d_2$ ist bzw. durch ein Polynom in $\log(d_1 \cdot d_2)$ beschränkt werden kann.

Bemerkung 3.2. Im Beweis von Theorem 3.1 zeigen wir, dass für eine gestutzte Version $\hat{\eta}_n$ des Kleinste-Quadrate-Schätzers $\eta_n^{(1)}$ (d.h. $\hat{\eta}_n(\mathbf{x}) = T_{\beta_n}\eta_n^{(1)}(\mathbf{x})$ für ein $\beta_n > 0$) gilt

$$\mathbf{E}\int |\hat{\eta}_n(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{\mathbf{X}}(d\mathbf{x}) \leq c_{13} \cdot \log(d_1 \cdot d_2) \cdot (\log n)^4 \cdot \max\left\{n^{-\frac{2\cdot p_1}{2\cdot p_1+4}}, n^{-\frac{2\cdot p_2}{2\cdot p_2+d^*}}\right\}$$

für eine Konstante $c_{13} > 0$, die nicht von d_1, d_2 und n abhängt. Da

$$\max\left\{n^{-\frac{2\cdot p_1}{2\cdot p_1+4}}, n^{-\frac{2\cdot p_2}{2\cdot p_2+d^*}}\right\}$$

die optimale Minimax-Konvergenzrate zur Schätzung einer Regressionsfunktion ist, die einem verallgemeinerten hierarchischen Max-Pooling Modell der Ordnung d^* mit Glattheitsbedingungen p_1 und p_2 genügt (siehe Abschnitt D des Anhangs von Kohler et al. (2022)), besitzt unser gestutzter Schätzer $\hat{\eta}_n$ bei der Schätzung einer solchen Regressionsfunktion eine (bis auf einen logarithmischen Faktor) optimale Konvergenzrate.

3.2.2. Resultat im hierarchischen Max-Pooling Modell mit zusätzlichem lokalen Max-Pooling

Im zweiten Hauptresultat wird eine Konvergenzrate des Fehlers $\mathbf{E}\{L(f_n^{(j)})\} - L(f^*)$ $(j = 3, 4, 5)$ für die in Abschnitt 3.1 eingeführten Bildklassifikatoren $f_n^{(3)}$, $f_n^{(4)}$ und $f_n^{(5)}$, basierend auf den Klassen $\mathcal{F}_3(\boldsymbol{\theta})$, $\mathcal{F}_4(\boldsymbol{\theta})$ und $\mathcal{F}_5(\boldsymbol{\theta})$, hergeleitet. Hier wird angenommen, die a-posteriori Wahrscheinlichkeit genüge dem hierarchischen Max-Pooling Modell mit zusätzlichem lokalen Max-Pooling aus Definition 4 in Abschnitt 2.2.

Theorem 3.2. *Es seien $d_1, d_2 \in \mathbb{N}$, es sei $n \in \mathbb{N}$ mit $n > 1$ und es seien $(\mathbf{X}, Y)$, $(\mathbf{X}_1, Y_1)$, …, $(\mathbf{X}_n, Y_n)$ unabhängig identisch verteilte $[0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \times \{0,1\}$-wertige Zufallsvariablen. Nehme an, dass die a-posteriori Wahrscheinlichkeit $\eta(\mathbf{x}) = \mathbf{P}\{Y = 1 | \mathbf{X} = \mathbf{x}\}$ einem hierarchischen Max-Pooling Modell vom Level l mit Feature Bedingung $\mathbf{b} = (b_1, \dots, b_{l-1})$, lokalem Max-Pooling Parameter $\mathbf{n} = (n_1, \dots, n_{l-1})$ und Glattheitsbedingung $p \in [1, \infty)$ genügt. Weiterhin nehme an, dass die Bilddimensionen*

$$d_1 = \prod_{i=1}^{l-1} n_i \cdot m_1 - 1, \quad d_2 = \prod_{i=1}^{l-1} n_i \cdot m_2 - 1 \quad \textit{und} \quad \min\{d_1, d_2\} \geq 2^l + \prod_{i=1}^{l-1} n_i - 1 \tag{3.17}$$

für $m_1, m_2 \in \mathbb{N}$ erfüllen. Setze

$$L_n = \left\lceil c_{14} \cdot n^{\frac{4}{2\cdot(2\cdot p+4)}} \right\rceil$$

für eine hinreichend große Konstante $c_{14} > 0$ und wähle die Parameter der faltenden neuronalen Netze aus den Funktionsklassen $\mathcal{F}_3(\boldsymbol{\theta}_3)$, $\mathcal{F}_4(\boldsymbol{\theta}_4)$ und $\mathcal{F}_5(\boldsymbol{\theta}_5)$ wie folgt:

Setze $L = l$, $\mathbf{s} = (n_1, \dots, n_{L-1})$, $s = n_1 \cdot \ldots \cdot n_{L-1}$, $b_{max} = \max\{b_1, \dots, b_{l-1}\}$, $z = b_{max} \cdot (L_n + 1)$,

$$\mathbf{A} = (A_1, A_1', A_2, A_2') = \left(1, \frac{d_1 - 2^l + 1}{\prod_{i=1}^{l-1} n_i}, 1, \frac{d_2 - 2^l + 1}{\prod_{i=1}^{l-1} n_i}\right),$$

$k = 2 \cdot b_{max} + c_{15}$ für eine hinreichend große Konstante $c_{15} > 0$, $\bar{k} = 2 \cdot k + 4$ und

$$\bar{z} = z + 3 \cdot k \cdot \log_2(\max\{n_1, \dots, n_{L-1}\}).$$

Weiterhin setze

$$M_r = \frac{2^{r-1}}{\prod_{i=0}^{r-1} n_i} + 1 \quad \textit{und} \quad \bar{M}_{(r-1)\cdot\bar{z}+1} = 2^{r-1} + 1, \dots, \bar{M}_{r\cdot\bar{z}} = 2^{r-1} + 1 \quad (r = 1, \dots, l)$$

sowie $\mathbf{M} = (M_1, \dots, M_L)$, $\bar{\mathbf{M}} = (\bar{M}_1, \dots, \bar{M}_{L\cdot\bar{z}})$, $\boldsymbol{\theta}_3 = (L, k, \mathbf{M}, z, \mathbf{s}, \mathbf{A})$, $\boldsymbol{\theta}_4 = (L, \bar{k}, \mathbf{M}, \bar{z}, \mathbf{s}, \mathbf{A})$ und $\boldsymbol{\theta}_5 = (L \cdot \bar{z}, \bar{k}, \bar{\mathbf{M}}, s, \mathbf{A})$. Definiere $f_n^{(j)}$ $(j = 3, 4, 5)$ als Plug-In Klassifikatoren durch Gleichung (3.15), basierend auf den Klassen $\mathcal{F}_3(\boldsymbol{\theta}_3)$, $\mathcal{F}_4(\boldsymbol{\theta}_4)$ und $\mathcal{F}_5(\boldsymbol{\theta}_5)$. Es gilt dann

$$\mathbf{E}\{L(f_n^{(j)})\} - L(f^*) \leq c_{16} \cdot \sqrt{\log(d_1 \cdot d_2)} \cdot (\log n)^2 \cdot n^{-\frac{p}{2\cdot p+4}}$$

für alle $j \in \{3, 4, 5\}$ und eine Konstante $c_{16} > 0$, die nicht von d_1, d_2 und n abhängt.

Beweis. Der Beweis befindet sich in Abschnitt 3.6.2. Das allgemeine Beweisvorgehen wird in Abschnitt 3.3 beschrieben.

Bemerkung 3.3. Auch in Theorem 3.2 hängt die Konvergenzrate nicht von den Bilddimensionen d_1 und d_2 ab, sodass unter den Annahmen an die a-posteriori Wahrscheinlichkeit aus Theorem 3.2 unsere Bildklassifikatoren basierend auf faltenden neuronalen Netzen in der Lage sind, den Fluch der Dimension zu umgehen. Wie in Theorem 3.1 hängt die Konstante c_{16} polynomiell von 2^l ab, weswegen die Dimension $d_1 \cdot d_2$ nur für den Fall logarithmisch auftritt, falls 2^l sehr viel kleiner als die Dimensionen $d_1 \cdot d_2$ ist bzw. durch ein Polynom in $\log(d_1 \cdot d_2)$ beschränkt werden kann.

Bemerkung 3.4. Die obige Bedingung (3.17) an die Bilddimensionen ist beispielsweise erfüllt, wenn

$$d_1 = 2^l \cdot m_1 - 1 \quad \text{und} \quad d_2 = 2^l \cdot m_2 - 1$$

für $m_1, m_2 \in \mathbb{N} \setminus \{1\}$. Da wir beliebige Bilddimensionen durch Hinzufügen eines Randes (beispielsweise durch ein entsprechendes Zero-Padding) in die obige Form bringen können, stellt die Annahme keine echte Einschränkung dar.

3.2.3. Resultat im rotationssymmetrischen hierarchischen Max-Pooling Modell

Im letzten Hauptresultat dieser Arbeit wird eine Konvergenzrate des Fehlers $\mathbf{E}\{L(f_n^{(2)})\} - L(f^*)$ für den in Abschnitt 3.1 eingeführten Bildklassifikator $f_n^{(2)}$ basierend auf der Klasse $\mathcal{F}_2(\boldsymbol{\theta})$ hergeleitet. Hier wird angenommen, die funktionale a-posteriori Wahrscheinlichkeit genüge dem rotationssymmetrischen hierarchischen Max-Pooling Modell aus Definition 5 in Abschnitt 2.3. Außerdem werden dabei die Annahmen 1 und 2 aus Abschnitt 2.3 vorausgesetzt.

Theorem 3.3. *Es sei $n \in \mathbb{N}$ mit $n > 1$ und $l \in \mathbb{N}$. Wähle $\lambda \in \mathbb{N}$ mit*

$$\lambda \geq 2^l + 2 \cdot l - 1, \quad (3.18) \qquad \text{sei } 0 < h \leq \frac{2^l}{\sqrt{2} \cdot \lambda}, \quad (3.19) \qquad \text{setze } b = \frac{2^l + 2 \cdot l - 1}{2 \cdot \lambda}, \quad (3.20)$$

und sei $p \in [1, \infty)$. Es seien (Φ, Y), (Φ_1, Y_1), ...,(Φ_n, Y_n) unabhängige identisch verteilte $[0,1]^{C_1} \times \{0,1\}$-wertige Zufallsvariablen. Nehme an, dass die funktionale a-posteriori Wahrscheinlichkeit $\eta_\Phi(\phi) = \mathbf{P}\{Y = 1 | \Phi = \phi\}$ einem rotationssymmetrischen hierarchischen Max-Pooling Modell vom Level l, der Breite h, mit Randabstand b und mit Glattheitsbedingung p genüge. Weiterhin nehme an, dass Annahme 1 für (p, C)-glatte Funktionen $\{g_{0,s}\}_{s=1,\dots,4^l}$ erfüllt sei und Annahme 2 für $\epsilon_\lambda \in [0,1]$, ein messbares $A \subset [0,1]^{C_1}$ und einen Skalierungsfaktor $c > 1$ erfüllt sei.

Wähle $L_n = \lceil c_{17} \cdot n^{2/(2p+4)} \rceil$ für eine hinreichend große Konstante $c_{17} > 0$, setze

$$L = \frac{4^{l+1} - 1}{3} \cdot (L_n + 1), \quad A_1 = A_2 = 2^{l-1} + l, \quad A_1' = A_2' = \lambda - 2^{l-1} - l + 1,$$

$$\mathbf{A} = (A_1, A_1', A_2, A_2'), \quad t = \left\lceil \frac{2^{l-1/2} \cdot \pi}{c - 1} \right\rceil, \quad L_{net} = \begin{cases} \lceil \log_2 t \rceil & \text{, falls } t > 1 \\ 1 & \text{, falls } t = 1, \end{cases}$$

$r_{net} = 3 \cdot t$ sowie $k = 5 \cdot 4^{l-1} + c_{18}$ für eine hinreichend große Konstante $c_{18} > 0$ und setze

$$M_s = I_{\{r>1\}} \cdot 2^{r-1} + 3 \quad \left(s = 1 + \sum_{i=0}^{r-1} 4^{l-i} \cdot (L_n + 1), \dots, \sum_{i=0}^{r} 4^{l-i} \cdot (L_n + 1) \right)$$

für $r = 0, \dots, l$, wobei die leere Summe als Null definiert wird. Weiterhin setze $\mathbf{M} = (M_1, \dots, M_L)$. Definiere $f_n^{(2)}$ als Plug-In Klassifikator durch (3.15) *unter Verwendung des Parametervektors $\boldsymbol{\theta} = (t, L_{net}, r_{net}, L, k, \mathbf{M}, \mathbf{A})$ basierend auf der Funktionsklasse $\mathcal{F}_2(\boldsymbol{\theta})$. Es gilt dann*

$$\mathbf{E}\{L(f_n^{(2)})\} - L(f^*) \leq c_{19} \cdot \sqrt{\log(\lambda) \cdot (\log n)^4 \cdot n^{-\frac{2 \cdot p}{2 \cdot p + 4}} + \epsilon_\lambda} \quad (3.21)$$

für eine Konstante $c_{19} > 0$, welche nicht von λ und n abhängt.

Beweis. Der Beweis befindet sich in Abschnitt 3.6.3. Das allgemeine Beweisvorgehen wird in Abschnitt 3.3 beschrieben.

Bemerkung 3.5. Die Konstante c_{19} in (3.21) hängt polynomiell von 2^l ab. Daher tritt die Auflösung λ in (3.21) nur logarithmisch für den Fall auf, in dem 2^l sehr viel kleiner als die Auflösung λ ist bzw. durch ein Polynom in $\log(\lambda)$ beschränkt werden kann, was zu kleinen Breiten h führt (vgl. Gleichung (3.19)). Da der Term

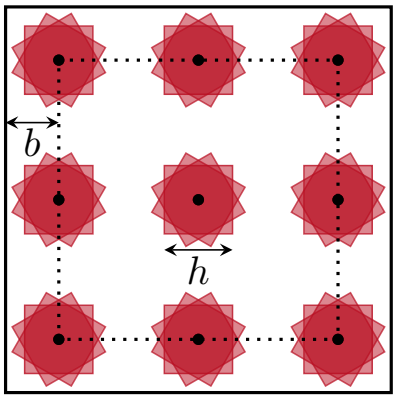
$$n^{-\frac{2\cdot p}{2\cdot p+4}}$$

in (3.21) nicht von der Auflösung λ abhängt, ist unser Bildklassifikator in der Lage, den Fluch der Dimension zu umgehen, wenn die a-posteriori Wahrscheinlichkeit den Annahmen in Theorem 3.3 genügt und wir den von der Auflösung abhängigen Fehlerterm ϵ_λ vernachlässigen.

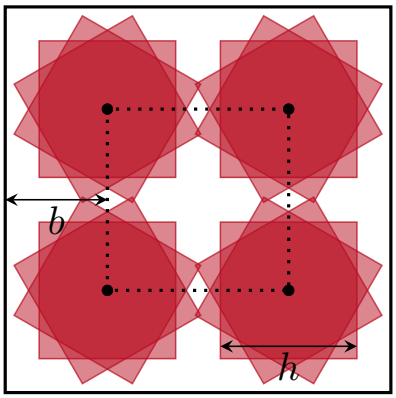
Bemerkung 3.6. Bedingung (3.18) stellt sicher, dass der Randabstand b, welcher durch Gleichung (3.20) definiert ist, kleiner oder gleich $1/2$ bleibt und dass wegen Gleichung (3.19) außerdem $h \leq 1/\sqrt{2}$ für die Breite h gilt. Weiterhin garantiert Bedingung (3.18), dass $h/\sqrt{2} \leq b$. Im Fall der maximalen Breite $h = 2^l/(\sqrt{2}\cdot\lambda)$ erhalten wir für großes l einen Randabstand, der nah an den minimalen Randabstand von $h/\sqrt{2}$ herankommt, da

$$b = \frac{2^l + 2\cdot l - 1}{2\cdot\lambda} = \frac{h}{\sqrt{2}} \cdot \underbrace{\frac{2^l + 2\cdot l - 1}{2^l}}_{\approx 1}.$$

Bedingung (3.18) und die Wahl des Randabstandes (3.20) sind daher keine tiefgreifenden Einschränkungen an unser Modell und wir erhalten, wie in Abbildung 3.4 dargestellt wird, sinnvolle Randabstände b und Breiten h der Teilbereiche. Weiterhin besteht die Möglichkeit, das Problem, dass relevante Objekte zu nah am Rand des Bildes liegen, zu vermeiden, indem den Bildern zuerst ein neutraler Rand hinzugefügt wird (beispielsweise durch ein entsprechendes Zero-Padding).



(a) Mögliche Teilbereiche der Breite $h = 2^l/(\sqrt{2}\cdot\lambda)$ im Fall $l = 7$ und $\lambda = 2^9$.



(b) Mögliche Teilbereiche der Breite $h = 2^l/(\sqrt{2}\cdot\lambda)$ im Fall $l = 8$ und $\lambda = 2^9$.

Abbildung 3.4.: Darstellung möglicher Teilbereiche des rotationssymmetrischen hierarchischen Max-Pooling Modells, welche durch die Bedingungen in Theorem 3.3 zugelassen sind.

Bemerkung 3.7. Im Beweis von Theorem 3.3 approximieren wir die funktionale a-posteriori Wahrscheinlichkeit durch ein faltendes neuronales Netz der obigen Klasse (für das Approximationsresultat siehe Lemma 11 in Abschnitt 3.4.3), bei dem die t faltenden neuronalen Netze, die parallel berechnet werden, die gleichen Gewichte haben. Genauer gesagt wird das faltende neuronale Netz dort so gewählt, dass jeder Filter einer beliebigen Schicht einem gedrehten Filter in derselben Schicht in einem parallel berechneten faltenden neuronalen Netz entspricht (die Gewichte haben nur unterschiedliche Positionen innerhalb der Filter). Mit

einer geeigneten Einschränkung unserer Funktionsklasse $\mathcal{F}_2(\boldsymbol{\theta})$, sodass die t faltenden neuronalen Netze geteilte Gewichte besitzen, könnte man daher die Konvergenzrate in Theorem 3.3 um einen konstanten Faktor verbessern. In einigen Anwendungen der Bildklassifikation, in denen rotierte Objekte einander entsprechen, erhöht eine solche Einschränkung die Performanz, siehe z.B. Marcos et al. (2016), Dieleman et al. (2015), Wu et al. (2015) und Cabrera-Vives et al. (2017). Unsere theoretische Analyse unterstützt daher die Verwendung einer solchen zusätzlichen Gewichtsteilung.

3.3. Allgemeines Beweisvorgehen und Hilfsresultate aus der Literatur

Die im letzten Abschnitt vorgestellten Resultate zur Konvergenzgeschwindigkeit unserer Bildklassifikatoren werden wir erst am Ende von Kapitel 3 beweisen. Hierfür werden jeweils eine Reihe von Hilfsresultaten benötigt, die in den folgenden Abschnitten behandelt werden. Um die entsprechenden Lemmata besser in den Gesamtzusammenhang einordnen zu können, beschreiben wir zunächst das allgemeine Beweisvorgehen der drei Resultate:

Zur Abschätzung des Fehlers $\mathbf{E}\{L(f_n^{(j)})\} - L(f^*)$ verwenden wir Ungleichung (1.5), womit es genügt,

$$\mathbf{E}\left\{\int |\hat{\eta}_n(\mathbf{x}) - \eta(\mathbf{x})|^2 \, \mathbf{P}_{\mathbf{X}}(d\mathbf{x})\right\}$$

abzuschätzen. In den Beweisen der Hauptresultate werden wir annehmen können, dass der Kleinste-Quadrate-Schätzer $\hat{\eta}_n$ der a-posteriori Wahrscheinlichkeit gestutzt wurde (siehe die Definition von $\hat{\eta}_n$ in Lemma 1 unten), weswegen die Schranke aus Lemma 1 unten eine Dekomposition des obigen erwarteten L_2-Fehlers in zwei Terme liefert. Der erste Term entspricht dabei der Komplexität der Funktionsklasse der verwendeten faltenden neuronalen Netze, für welche die folgende Definition der *L_p-ϵ-Überdeckungszahl* ein Maß liefert.

Definition 6. Es sei X die Klasse aller Funktionen $f : \mathbb{R}^d \to \mathbb{R}$ und $\mathcal{F} \subset X$. Außerdem seien $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$ mit $\mathbf{x}_1^n = (\mathbf{x}_1, ..., \mathbf{x}_n)$, $\epsilon > 0$ und $1 \leq p < \infty$.

a) Eine endliche Menge $\mathcal{G} \subset X$ heißt *L_p-ϵ-Überdeckung* von $\mathcal{F}$ auf $\mathbf{x}_1^n$, falls für alle $f \in \mathcal{F}$ gilt

$$\min_{g \in \mathcal{G}} \left(\frac{1}{n} \sum_{i=1}^{n} |f(\mathbf{x}_i) - g(\mathbf{x}_i)|^p \right)^{\frac{1}{p}} < \epsilon.$$

b) Wir nennen

$$\mathcal{N}_p(\epsilon, \mathcal{F}, \mathbf{x}_1^n) = \inf\left\{|\mathcal{G}| \ : \ \mathcal{G} \text{ ist } L_p\text{-}\epsilon\text{-Überdeckung von } \mathcal{F} \text{ auf } x_1^n\right\}$$

die *L_p-ϵ-Überdeckungszahl* von $\mathcal{F}$ auf x_1^n ($\inf \emptyset := \infty$).

Der zweite Term entspricht dem Approximationsfehler, der entsteht, wenn wir die a-posteriori Wahrscheinlichkeit durch ein faltendes neuronales Netz approximieren.

Lemma 1. *Es seien $(\mathbf{X}, Y)$, $(\mathbf{X}_1, Y_1)$, …, $(\mathbf{X}_n, Y_n)$ unabhängig identisch verteilte $\mathbb{R}^d \times \mathbb{R}$-wertige Zufallsvariablen. Nehme an, die Verteilung von $(\mathbf{X}, Y)$ genüge*

$$\mathbf{E}\{\exp(c_{20} \cdot Y^2)\} < \infty$$

für eine Konstante $c_{20} > 0$ und die Regressionsfunktion $\eta(\cdot) = \mathbf{E}\{Y|\mathbf{X} = \cdot\}$ sei beschränkt. Weiterhin sei η_n der Kleinste-Quadrate-Schätzer

$$\eta_n(\cdot) = \arg\min_{f \in \mathcal{F}_n} \frac{1}{n} \sum_{i=1}^{n} |Y_i - f(\mathbf{X}_i)|^2$$

basierend auf einer Funktionsklasse $\mathcal{F}_n$ bestehend aus Funktionen $f : \mathbb{R}^d \to \mathbb{R}$. Setze $\hat{\eta}_n = T_{c_{21} \cdot \log(n)} \eta_n$ für eine Konstante $c_{21} > 0$. Dann gilt

$$\begin{aligned}
&\mathbf{E}\left\{ \int |\hat{\eta}_n(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{\mathbf{X}}(d\mathbf{x}) \right\} \\
&\leq \frac{c_{22} \cdot (\log(n))^2 \cdot \sup_{\mathbf{x}_1^n \in (\mathbb{R}^d)^n} \left(\log\left(\mathcal{N}_1\left(\frac{1}{n \cdot c_{21} \log(n)}, T_{c_{21} \log(n)} \mathcal{F}_n, \mathbf{x}_1^n \right) \right) + 1 \right)}{n} \\
&\quad + 2 \cdot \inf_{f \in \mathcal{F}_n} \int |f(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{\mathbf{X}}(d\mathbf{x})
\end{aligned}$$

für $n > 1$ und eine Konstante $c_{22} > 0$, welche nicht von n oder den Parametern des Schätzers abhängt.

Beweis. Dieses Resultat folgt durch eine leichte Modifizierung aus dem Beweis von Theorem 1 in Bagirov et al. (2009). Ein vollständiger Beweis findet sich im Anhang von Bauer und Kohler (2019). □

Um den Approximationsfehler zu beschränken, werden wir in Abschnitt 3.4 die Approximationseigenschaften der Funktionsklassen von faltenden neuronalen Netzen untersuchen, die in Abschnitt 3.1 eingeführt wurden. Das Ziel ist es, dabei zu zeigen, dass die entsprechenden Architekturen die verschiedenen Modelle der a-posteriori Wahrscheinlichkeit gut approximieren können. Für die Beweise aller drei Hauptresultate wird hierfür eine Verbindung zwischen faltenden neuronalen Netzen und vollverbundenen neuronalen Netzen hergestellt (siehe Lemma 4 in Abschnitt 3.4.1 und Lemma 14 in Abschnitt 3.4.3). Die vollverbundenen neuronalen Netze sind aus den Klasse $\mathcal{G}_d(L_{net}, r_{net})$ (siehe Gleichung (3.8)) und sollen dabei jeweils die (p, C)-glatten Funktionen $g_{k,s} : \mathbb{R}^d \to [0, 1]$, die innerhalb der drei Modelle (siehe Definition 3, Definition 4 und Definition 5) auftreten, approximieren (es treten die Fälle $d = 1$, $d = 4$ und $d = d^*$ auf). Dies wird es ermöglichen, das folgende Approximationsresultat für (p, C)-glatte Funktionen durch sehr tiefe vollverbundene neuronale Netze von Kohler und Langer (2021) zu verwenden.

Lemma 2. *Es sei $d \in \mathbb{N}$ und es sei $f : \mathbb{R}^d \to \mathbb{R}$ eine (p, C)–glatte Funktion für ein $p = q + s$, $q \in \mathbb{N}_0$, $s \in (0, 1]$ und $C > 0$. Außerdem sei $M \in \mathbb{N}$ mit $M > 1$ hinreichend groß, wobei*

$$M^{2p} \geq c_{23} \cdot \left(\max\left\{ 2, \sup_{\substack{\mathbf{x} \in [-2,2]^d \\ (l_1,\ldots,l_d) \in \mathbb{N}^d \\ l_1 + \cdots + l_d \leq q}} \left| \frac{\partial^{l_1 + \cdots + l_d} f}{\partial^{l_1} x^{(1)} \ldots \partial^{l_d} x^{(d)}}(\mathbf{x}) \right| \right\} \right)^{4(q+1)}$$

für eine hinreichend große Konstante $c_{23} \geq 1$ gelte. Sei $\sigma : \mathbb{R} \to \mathbb{R}$ die ReLU Aktivierungsfunktion

$$\sigma(x) = \max\{x, 0\}$$

und seien $L, r \in \mathbb{N}$, sodass

(i)

$$L \geq 5M^d + \left\lceil \log_4\left(M^{2p+4 \cdot d \cdot (q+1)} \cdot e^{4(q+1) \cdot (M^d - 1)} \right) \right\rceil \cdot \lceil \log_2(\max\{d, q\} + 1) \rceil + \lceil \log_4(M^{2p}) \rceil$$

(ii)

$$r \geq 132 \cdot 2^d \cdot \lceil e^d \rceil \cdot \binom{d+q}{d} \cdot \max\{q + 1, d^2\}.$$

Dann existiert ein neuronales Netz

$$f_{net} \in \mathcal{G}_d(L, r),$$

sodass

$$\sup_{\mathbf{x}\in[-2,2]^d} |f(\mathbf{x}) - f_{net}(\mathbf{x})| \leq c_{24} \cdot \left(\max \left\{ 2, \sup_{\substack{\mathbf{x}\in[-2,2]^d \\ (l_1,\dots,l_d)\in\mathbb{N}^d \\ l_1+\cdots+l_d\leq q}} \left| \frac{\partial^{l_1+\cdots+l_d} f}{\partial^{l_1} x^{(1)} \dots \partial^{l_d} x^{(d)}}(\mathbf{x}) \right| \right\} \right)^{4(q+1)} \cdot M^{-2p}.$$

Beweis. Siehe Theorem 2 in Kohler und Langer (2021). Ein alternativer Beweis eines eng verwandten Resultats findet sich in Yarotsky und Zhevnerchuk (2020) (Theorem 4.1). □

Der Unterschied zu bestehenden Arbeiten wie Oono und Suzuki (2019) oder Petersen und Voigtlaender (2020), in denen ebenfalls die Approximationsfähigkeit vollverbundener neuronaler Netze genutzt wird, besteht darin, dass wir die Approximationsfähigkeit vollverbundener neuronaler Netze nicht nutzen, um den Gesamtapproximationsfehler unseres faltenden neuronalen Netzes abzuschätzen, sondern die faltenden neuronalen Netze an die jeweiligen Modelle anpassen und die Approximationsfähigkeit der vollverbundenen neuronalen Netze an vielen Stellen innerhalb der faltenden neuronalen Netze verwenden.

Eine Analyse der Komplexität der Funktionsklassen von faltenden neuronalen Netzen, d.h. die Herleitung einer Abschätzung der L_1-ϵ-Überdeckungszahl der verwendeten Netzwerkarchitekturen, erfolgt in Abschnitt 3.5. Hier wird eine neue Schranke der VC-Dimension für faltende neuronale Netze gezeigt (siehe Lemma 20 in Abschnitt 3.5 für das entsprechende Resultat und Definition 10 in Abschnitt 3.5 für die Definition der VC-Dimension). Das entsprechende Resultat stellt eine Modifikation von Theorem 6 in Bartlett et al. (2019) dar. Konkret haben wir den dortigen Beweis an Architekturen von faltenden neuronalen Netzen angepasst, d.h. insbesondere für den Fall, dass die Architektur eine globale Max-Pooling Schicht enthält.

3.4. Approximationseigenschaften von faltenden neuronalen Netzen

Wie im letzten Abschnitt bereits angemerkt, benötigen wir für die Beweise der drei Hauptresultate aus Abschnitt 3.2 einige Approximationseigenschaften von faltenden neuronalen Netzen, welche wir in in diesem Abschnitt formulieren und beweisen werden.

3.4.1. Approximation des hierarchischen Max-Pooling Modells mit zusätzlichem lokalen Pooling

In diesem Abschnitt stellen wir eine Verbindung zwischen vollverbundenen neuronalen Netzen und faltenden neuronalen Netzen her, die es uns ermöglichen wird, in den Beweisen von Theorem 3.1 und Theorem 3.2 Approximationsresultate für das verallgemeinerte hierarchische Max-Pooling Modell sowie das hierarchische Max-Pooling Modell mit zusätzlichem lokalen Max-Pooling mittels faltender neuronaler Netze zu erzielen. Da das hierarchische Max-Pooling Modell aus Definition 3 c), wie in Bemerkung 2.4 beschrieben, ein Spezialfall des hierarchischen Max-Pooling Modells mit zusätzlichem lokalen Max-Pooling darstellt, benötigen wir lediglich ein entsprechendes Resultat für das allgemeinere Modell mit zusätzlichem lokalen Max-Pooling. Dieses Approximationsresultat können wir dann sowohl für den Beweis von Theorem 3.1 als auch für den Beweis von Theorem 3.2 verwenden. Zur Approximation des Modells verwenden wir faltende neuronale Netze aus der Klasse $\mathcal{F}_3(\boldsymbol{\theta})$.

Im Folgenden bezeichnen $d_1, d_2 \in \mathbb{N} \setminus \{1\}$ die Bilddimensionen und wir nehmen an, die Funktion

$$m : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to [0,1]$$

genüge einem hierarchischen Max-Pooling Modell vom Level $l \in \mathbb{N}$ mit Feature Bedingung $\mathbf{b} = (b_1, \ldots, b_{l-1})$ und lokalem Max-Pooling Parameter $\mathbf{n} = (n_1, \ldots, n_{l-1})$ (siehe Definition 4 in Abschnitt 2.2). Die entsprechenden Feature Maps sind dann gegeben durch

$$z_{k,s} : [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}} \to [0,1]^{\{1,\ldots,\lceil d_1(k)/n_k\rceil\}\times\{1,\ldots,\lceil d_2(k)/n_k\rceil\}}$$

für $k = 0, \ldots, l$ und $s = 1, \ldots, b_k$ sowie

$$y_{k,s} : [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}} \to [0,1]^{\{1,\ldots,d_1(k)\}\times\{1,\ldots,d_2(k)\}}$$

und

$$g_{k,s} : \mathbb{R}^4 \to [0,1] \tag{3.22}$$

für $k = 1, \ldots, l$ und $s = 1, \ldots, b_k$. Die obigen Dimensionen $d_1(k)$ und $d_2(k)$ sind wie in Definition 3 definiert durch

$$d_1(k) = \left\lceil \frac{d_1(k-1)}{n_{k-1}} \right\rceil - \delta_{k-1} \text{ und } d_2(k) = \left\lceil \frac{d_2(k-1)}{n_{k-1}} \right\rceil - \delta_{k-1},$$

für $k = 1, \ldots, l$ mit $d_1(0) = d_1$, $d_2(0) = d_2$ und $\delta_{k-1} = 2^{k-1}/\prod_{i=0}^{k-1} n_i$. Wir zeigen in diesem Abschnitt, dass ein faltendes neuronales Netz der Klasse $\mathcal{F}_3(\boldsymbol{\theta})$ eine Funktion

$$\bar{m} : [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}} \to \mathbb{R}_0^+$$

der nachfolgenden Form imitieren kann, welche wir dann verwenden um die Funktion m zu approximieren.

Die Funktion $\bar{m}$ habe die Form

$$\bar{m}(x) = \max_{(i,j)\in\{1,\ldots,d_1(l)\}\times\{1,\ldots,d_2(l)\}} \big(\bar{z}_{l,1}(\mathbf{x})\big)_{(i,j)}, \tag{3.23}$$

wobei

$$\bar{z}_{k,s} : [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}} \to \mathbb{R}_0^{+\{1,\ldots,\lceil d_1(k)/n_k\rceil\}\times\{1,\ldots,\lceil d_2(k)/n_k\rceil\}}$$

für $k = 1, \ldots, l$ und $s = 1, \ldots, b_k$ wie folgt rekursiv definiert ist:

1. Es sei $\bar{g}_{k,s} : \mathbb{R}^4 \to \mathbb{R}$ eine Funktion für $k \in \{1, \ldots, l\}$ und $s \in \{1, \ldots, b_k\}$. Weiterhin setze

$$\bar{z}_{0,1}(\mathbf{x}) = \mathbf{x}.$$

2. Definiere rekursiv die Funktionen

$$\bar{y}_{k,s} : [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}} \to \mathbb{R}_0^{+\{1,\ldots,d_1(k)\}\times\{1,\ldots,d_2(k)\}}$$

durch

$$\begin{aligned}\big(\bar{y}_{k,s}(\mathbf{x})\big)_{(i,j)} = \sigma\Big(\bar{g}_{k,s}\Big(&\big(\bar{z}_{k-1,r_1(k,s)}(\mathbf{x})\big)_{(i,j)}, \big(\bar{z}_{k-1,r_2(k,s)}(\mathbf{x})\big)_{(i+\delta_{k-1},j)},\\ &\big(\bar{z}_{k-1,r_3(k,s)}(\mathbf{x})\big)_{(i,j+\delta_{k-1})}, \big(\bar{z}_{k-1,r_4(k,s)}(\mathbf{x})\big)_{(i+\delta_{k-1},j+\delta_{k-1})}\Big)\Big)\end{aligned} \tag{3.24}$$

für $k = 1, \ldots, l$, $s = 1, \ldots, b_k$, $(i,j) \in \{1, \ldots, d_1(k)\} \times \{1, \ldots, d_2(k)\}$ und

$$r_1(k,s), r_2(k,s), r_3(k,s), r_4(k,s) \in \{1, \ldots, b_{k-1}\}.$$

3. Als Nächstes definiere

$$\big(\bar{z}_{k,s}(\mathbf{x})\big)_{(i,j)} = \max_{(i_2,j_2)\in N^{(k)}_{(i,j)}} \big(\bar{y}_{k,s}(\mathbf{x})\big)_{(i_2,j_2)} \tag{3.25}$$

für $k = 1, \dots, l$, $s = 1, \dots, b_k$ und $(i,j) \in \{1, \dots, \lceil d_1(k)/n_k \rceil\} \times \{1, \dots, \lceil d_2(k)/n_k \rceil\}$,

wobei die Nachbarschaften $N^{(k)}_{(i,j)}$ durch Gleichung 2.3 definiert sind (mit $n_l = 1$). Die Funktion $\bar{m}$ entspricht im Allgemeinen keinem hierarchischen Max-Pooling Modell mit zusätzlichem lokalen Max-Pooling aus Definition 5 b), da die Verkettungen $\sigma \circ \bar{g}_{k,s}$ (siehe Gleichung (3.24)) keine $[0,1]$-wertigen Funktionen darstellen, sondern nach $\mathbb{R}_0^+$ abbilden. In Gleichung (3.24) wird die ReLU Funktion $\sigma(x) = \max\{x, 0\}$ auf die Ausgabe der Funktion $\bar{g}_{k,s}$ angewendet, da uns dies ermöglicht, die Funktion $\bar{m}$ als ein faltendes neuronales Netz darzustellen, für den Fall, dass die Funktionen $\bar{g}_{k,s}$ vollverbundenen neuronalen Netzen aus der Klasse $\mathcal{G}_4(L_{net}, r_{net})$ entsprechen (siehe Lemma 4). Wie wir im folgenden Lemma sehen werden, beeinflusst die Anwendung der ReLU Funktion nicht unser Approximationsresultat, da die Funktionen $g_{k,s}$ in der Definition von m $[0,1]$-wertig sind.

Lemma 3. *Nehme an, die Einschränkungen $g_{k,s}|_{[-2,2]^4} : [-2,2]^4 \to [0,1]$ der Funktionen (3.22) seien lipschitzstetig für alle $k \in \{1, \dots, l\}$ und $s \in \{1, \dots, b_k\}$ mit einer Lipschitzkonstanten $C > 0$ (in Bezug auf die euklidische Metrik). Weiterhin nehme an, für alle $k \in \{1, \dots, l\}$ und $s \in \{1, \dots, b_k\}$ gelte*

$$\big\|\bar{g}_{k,s}\big\|_{[-2,2]^4,\infty} \leq 2. \tag{3.26}$$

Dann gilt

$$|m(\mathbf{x}) - \bar{m}(\mathbf{x})| \leq (2C+1)^{l-1} \cdot \max_{k\in\{1,\dots,l\}, s\in\{1,\dots,b_k\}} \|g_{k,s} - \bar{g}_{k,s}\|_{[-2,2]^4,\infty}$$

für alle $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$.

Beweis. Wir verwenden im Beweis, dass

$$|\max_{i=1,\dots,n} a_i - \max_{i=1,\dots,n} b_i| \leq \max_{i=1,\dots,n} |a_i - b_i| \tag{3.27}$$

für a_1, b_1, …, a_n, $b_n \in \mathbb{R}$. Um dies einzusehen, sei o.B.d.A. $a_1 = \max_{i=1,\dots,n} a_i \geq \max_{i=1,\dots,n} b_i$. Dann gilt

$$|\max_{i=1,\dots,n} a_i - \max_{i=1,\dots,n} b_i| = a_1 - \max_{i=1,\dots,n} b_i \leq a_1 - b_1 \leq \max_{i=1,\dots,n} |a_i - b_i|.$$

Daher reicht es aus

$$\begin{aligned}&\max_{(i,j)\in\{1,\dots,d_1(l)\}\times\{1,\dots,d_2(l)\}} \Big|\big(z_{l,1}(\mathbf{x})\big)_{(i,j)} - \big(\bar{z}_{l,1}(\mathbf{x})\big)_{(i,j)}\Big| \\ &\leq (2C+1)^{l-1} \cdot \max_{k\in\{1,\dots,l\}, s\in\{1,\dots,b_k\}} \|g_{k,s} - \bar{g}_{k,s}\|_{[-2,2]^4,\infty}\end{aligned}$$

zu zeigen, was wiederum impliziert wird durch

$$|\big(z_{k,s}(\mathbf{x})\big)_{(i,j)} - \big(\bar{z}_{k,s}(\mathbf{x})\big)_{(i,j)}| \leq (2C+1)^{k-1} \cdot \max_{m\in\{1,\dots,k\}, s\in\{1,\dots,b_m\}} \|g_{m,s} - \bar{g}_{m,s}\|_{[-2,2]^4,\infty} \tag{3.28}$$

für alle $k \in \{1, \dots, l\}$, $s \in \{1, \dots, b_k\}$, $(i,j) \in \{1, \dots, d_1(k)\} \times \{1, \dots, d_2(k)\}$ und $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$. Die obige Ungleichung zeigen wir im Folgenden durch Induktion über k.

Wegen Ungleichung (3.27) erhalten wir für $(i,j) \in \{1, \dots, \lceil d_1(1)/n_1 \rceil\} \times \{1, \dots, \lceil d_2(1)/n_1 \rceil\}$ und $s \in \{1, \dots, b_1\}$, dass

$$\Big|\big(z_{1,s}(\mathbf{x})\big)_{(i,j)} - \big(\bar{z}_{1,s}(\mathbf{x})\big)_{(i,j)}\Big|$$

$$
\begin{aligned}
&= \Big| \max_{(i_2,j_2)\in N^{(1)}_{(i,j)}} g_{1,s}(x_{i_2,j_2}, x_{i_2+1,j_2}, x_{i_2,j_2+1}, x_{i_2+1,j_2+1}) \\
&\qquad - \max_{(i_2,j_2)\in N^{(1)}_{(i,j)}} \sigma\big(\bar{g}_{1,s}(x_{i_2,j_2}, x_{i_2+1,j_2}, x_{i_2,j_2+1}, x_{i_2+1,j_2+1})\big)\Big| \\
&\leq \max_{(i_2,j_2)\in N^{(1)}_{(i,j)}} \big|g_{1,s}(x_{i,j}, x_{i+1,j}, x_{i,j+1}, x_{i+1,j+1}) - \sigma\big(\bar{g}_{1,s}(x_{i_2,j_2}, x_{i_2+1,j_2}, x_{i_2,j_2+1}, x_{i_2+1,j_2+1})\big)\big| \\
&\leq \max_{(i_2,j_2)\in N^{(1)}_{(i,j)}} \big|g_{1,s}(x_{i,j}, x_{i+1,j}, x_{i,j+1}, x_{i+1,j+1}) - \bar{g}_{1,s}(x_{i_2,j_2}, x_{i_2+1,j_2}, x_{i_2,j_2+1}, x_{i_2+1,j_2+1})\big| \\
&\leq \big\|g_{1,s} - \bar{g}_{1,s}\big\|_{[0,1]^4,\infty},
\end{aligned}
$$

wobei wir bei der zweiten Abschätzung verwendet haben, dass die Funktionen $g_{1,s}$ $(s = 1, \dots, b_1)$ $[0,1]$-wertig sind. Wir nehmen nun an, es gelte (3.28) für ein $k \in \{1, \dots, l-1\}$. Die Definition von $\bar{z}_{k,s}$ und Ungleichung (3.26) implizieren

$$
\Big|\big(\bar{z}_{k,s}(\mathbf{x})\big)_{(i,j)}\Big| \leq 2
$$

für alle $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$, $s \in \{1, \dots, b_k\}$ und $(i,j) \in \{1, \dots, d_1(k)\} \times \{1, \dots, d_2(k)\}$. Es gilt dann wegen (3.27), der $[0,1]$-Wertigkeit der Funktionen $g_{k+1,s}$, der Dreiecksungleichung, der Lipschitzstetigkeit von $g_{k+1,s}$ und Ungleichung, dass

$$
\begin{aligned}
&\Big|\big(z_{k+1,s}(\mathbf{x})\big)_{(i,j)} - \big(\bar{z}_{k+1,s}(\mathbf{x})\big)_{(i,j)}\Big| \\
&= \left| \max_{(i_2,j_2)\in N^{(k+1)}_{(i,j)}} \big(y_{k+1,s}(\mathbf{x})\big)_{(i_2,j_2)} - \max_{(i_2,j_2)\in N^{(k+1)}_{(i,j)}} \big(\bar{y}_{k+1,s}(\mathbf{x})\big)_{(i_2,j_2)} \right| \\
&\overset{(3.27)}{\leq} \max_{(i_2,j_2)\in N^{(k+1)}_{(i,j)}} \Big|\big(y_{k+1,s}(\mathbf{x})\big)_{(i_2,j_2)} - \big(\bar{y}_{k+1,s}(\mathbf{x})\big)_{(i_2,j_2)}\Big| \\
&\leq \max_{(i_2,j_2)\in N^{(k+1)}_{(i,j)}} \Big| g_{k+1,s}\Big(\big(z_{k,r_1(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2)}, \big(z_{k,r_2(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2)}, \\
&\qquad\qquad \big(z_{k,r_3(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2+\delta_k)}, \big(z_{k,r_4(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2+\delta_k)}\Big) \\
&\qquad\qquad - \sigma\Big(\bar{g}_{k+1,s}\big(\big(\bar{z}_{k,r_1(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2)}, \big(\bar{z}_{k,r_2(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2)}, \\
&\qquad\qquad\qquad \big(\bar{z}_{k,r_3(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2+\delta_k)}, \big(\bar{z}_{k,r_4(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2+\delta_k)}\big)\Big)\Big| \\
&\leq \max_{(i_2,j_2)\in N^{(k+1)}_{(i,j)}} \Big| g_{k+1,s}\Big(\big(z_{k,r_1(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2)}, \big(z_{k,r_2(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2)}, \\
&\qquad\qquad \big(z_{k,r_3(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2+\delta_k)}, \big(z_{k,r_4(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2+\delta_k)}\Big) \\
&\qquad\qquad - \bar{g}_{k+1,s}\big(\big(\bar{z}_{k,r_1(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2)}, \big(\bar{z}_{k,r_2(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2)}, \\
&\qquad\qquad\qquad \big(\bar{z}_{k,r_3(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2+\delta_k)}, \big(\bar{z}_{k,r_4(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2+\delta_k)}\big)\Big| \\
&\leq \max_{(i_2,j_2)\in N^{(k+1)}_{(i,j)}} \Big| g_{k+1,s}\Big(\big(z_{k,r_1(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2)}, \big(z_{k,r_2(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2)}, \\
&\qquad\qquad \big(z_{k,r_3(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2+\delta_k)}, \big(z_{k,r_4(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2+\delta_k)}\Big) \\
&\qquad\qquad - g_{k+1,s}\Big(\big(\bar{z}_{k,r_1(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2)}, \big(\bar{z}_{k,r_2(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2)},
\end{aligned}
$$

$$
\begin{aligned}
&\qquad\qquad \big(\bar{z}_{k,r_3(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2+\delta_k)}, \big(\bar{z}_{k,r_4(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2+\delta_k)}\Big)\Big| \\
&\quad + \Big|g_{k+1,s}\Big(\big(\bar{z}_{k,r_1(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2)}, \big(\bar{z}_{k,r_2(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2)}, \\
&\qquad\qquad \big(\bar{z}_{k,r_3(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2+\delta_k)}, \big(\bar{z}_{k,r_4(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2+\delta_k)}\Big) \\
&\qquad - \bar{g}_{k+1,s}\Big(\big(\bar{z}_{k,r_1(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2)}, \big(\bar{z}_{k,r_2(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2)}, \\
&\qquad\qquad \big(\bar{z}_{k,r_3(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2+\delta_k)}, \big(\bar{z}_{k,r_4(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2+\delta_k)}\Big)\Big| \\
&\leq \max_{(i_2,j_2)\in N_{(i,j)}^{(k+1)}} C \cdot \Bigg(\Big|\big(z_{k,r_1(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2)} - \big(\bar{z}_{k,r_1(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2)}\Big|^2 \\
&\quad + \Big|\big(z_{k,r_2(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2)} - \big(\bar{z}_{k,r_2(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2)}\Big|^2 \\
&\quad + \Big|\big(z_{k,r_3(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2+\delta_k)} - \big(\bar{z}_{k,r_3(k+1,s)}(\mathbf{x})\big)_{(i_2,j_2+\delta_k)}\Big|^2 \\
&\quad + \Big|\big(z_{k,r_4(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2+\delta_k)} - \big(\bar{z}_{k,r_4(k+1,s)}(\mathbf{x})\big)_{(i_2+\delta_k,j_2+\delta_k)}\Big|^2\Bigg)^{1/2} \\
&\quad + \|g_{k+1,s} - \bar{g}_{k+1,s}\|_{[-2,2]^4,\infty} \\
&\overset{(3.28)}{\leq} (2\cdot C)\cdot(2C+1)^{k-1} \cdot \max_{m\in\{1,\dots,k\},s\in\{1,\dots,b_m\}} \|g_{m,s} - \bar{g}_{m,s}\|_{[-2,2]^4,\infty} \\
&\quad + \|g_{k+1,s} - \bar{g}_{k+1,s}\|_{[-2,2]^4,\infty} \\
&\leq (2C+1)^k \cdot \max_{m\in\{1,\dots,k+1\},s\in\{1,\dots,b_m\}} \|g_{m,s} - \bar{g}_{m,s}\|_{[-2,2]^4,\infty}
\end{aligned}
$$

für alle $s \in \{1,\dots,b_{k+1}\}$, $(i,j) \in \{1,\dots,d_1(k+1)\} \times \{1,\dots,d_2(k+1)\}$ und $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$. □

Das folgende Resultat ist das Hauptresultat dieses Abschnitts. Es ermöglicht uns, die Funktion $\bar{m}$ als ein faltendes neuronales Netz aus der Klasse $\mathcal{F}_3(\boldsymbol{\theta})$ darzustellen, für den Fall, dass die Funktionen $\bar{g}_{k,s}$ vollverbundenen neuronalen Netzen aus der Klasse $\mathcal{G}_4(L_{net}, r_{net})$ entsprechen.

Lemma 4. *Nehme an, es existieren $m_1, m_2 \in \mathbb{N}$, sodass für die Bilddimensionen d_1 und d_2 gilt:*

$$
d_1 = \prod_{i=1}^{l-1} n_i \cdot m_1 - 1, \quad d_2 = \prod_{i=1}^{l-1} n_i \cdot m_2 - 1 \quad \textit{und} \quad \min\{d_1, d_2\} \geq 2^l + \prod_{i=1}^{l-1} n_i - 1.
$$

Weiterhin nehme an, dass die Funktionen $\bar{g}_{r,s} : \mathbb{R}^4 \to \mathbb{R}$ in der Definition (3.23) *von $\bar{m}$ für alle $r \in \{1,\dots,l\}$ und $s \in \{1,\dots,b_r\}$ in $\mathcal{G}_4\{L_{net}, r_{net}\}$ enthalten sind, wobei $L_{net}, r_{net} \in \mathbb{N}$. Außerdem setze $b_{max} = \max\{b_1,\dots,b_l\}$, wähle die Parameter*

$$
L = l, \quad \mathbf{A} = (1, d_1(l), 1, d_2(l)) \quad \textit{und} \quad z = b_{max} \cdot (L_{net} + 1),
$$

und für $r \in \{1,\dots,L\}$ wähle

$$
k = 2 \cdot b_{max} + r_{net}, \quad M_r \geq \delta_{r-1} + 1, \quad s_r = n_r.
$$

Dann existiert ein $m_{net} \in \mathcal{F}_3\big((L, k, \mathbf{M}, z, \mathbf{s}, \mathbf{A})\big)$, sodass

$$
\bar{m}(\mathbf{x}) = m_{net}(\mathbf{x})
$$

für alle $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$.

Um Lemma 4 zu beweisen, benötigen wir das in Lemma 5 folgende Hilfsresultat. Im Folgenden verwenden wir die Bezeichnung eines faltenden Blocks. Ein faltender Block ist eine Verkettung von $z \in \mathbb{N}$ faltenden Schichten, welche jeweils gleich viele Ausgabekanäle $k \in \mathbb{N}$ und die gleiche Filtergröße $M \in \mathbb{N}$ besitzen. Für $i_1, i_2, k' \in \mathbb{N}$ und eine Indexmenge $I = \{1, \dots, i_1\} \times \{1, \dots, i_2\}$ ist ein faltender Block dann eine Funktion

$$o^{(z)}_{(k',k),M} : \mathbb{R}^{I \times \{1,\dots,k'\}} \to \mathbb{R}_0^{+\,I \times \{1,\dots,k\}},$$

welche definiert ist durch

$$o^{(z)}_{(k',k),M}(\mathbf{x}) = \big(o_{(k,k),M,1,\mathbf{w}_z} \circ \cdots \circ o_{(k',k),M,1,\mathbf{w}_1}\big)(\mathbf{x}) \quad \big(\mathbf{x} \in \mathbb{R}^{I \times \{1,\dots,k'\}}\big). \tag{3.29}$$

Lemma 5. *Es sei $g_{net} \in \mathcal{G}_4(L_{net}, r_{net})$ mit $L_{net}, r_{net} \in \mathbb{N}$ (siehe Abschnitt 3.1 für die Definition von $\mathcal{G}_4(L_{net}, r_{net})$). Sei*

$$f : \mathbb{R}^{\{1,\dots,d_1\} \times \{1,\dots,d_2\}} \to \mathbb{R}^{I \times \{1,\dots,k'\}}$$

eine Funktion, wobei $d_1, d_2, i_1, i_2, k' \in \mathbb{N}$ und $I = \{1, \dots, i_1\} \times \{1, \dots, i_2\}$. Außerdem seien $t \in \mathbb{N}$, $s_1, \dots, s_4 \in \{1, \dots, k'\}$, $s_5 \in \{1, \dots, t\}$ und $M, \delta \in \mathbb{N}$ mit $M \geq \delta + 1$. Dann existiert ein faltender Block

$$o^{(L_{net}+1)}_{(k',t+r_{net}),M} : \mathbb{R}^{I \times \{1,\dots,k'\}} \to \mathbb{R}^{I \times \{1,\dots,t+r_{net}\}},$$

welcher wie in Gleichung (3.29) *definiert ist und beliebige Gewichte in den Kanälen*

$$\{1, \dots, t\} \setminus \{s_5\}$$

besitzt, sodass

$$\begin{aligned}
&\Big(\big(o^{(L_{net}+1)}_{(t+r_{net},t+r_{net}),M} \circ f\big)(\mathbf{x})\Big)_{(i,j),s_5} \\
&= \sigma\Big(g_{net}\Big(\big(f(\mathbf{x})\big)_{(i,j),s_1}, \big(f(\mathbf{x})\big)_{(i+\delta,j),s_2}, \big(f(\mathbf{x})\big)_{(i,j+\delta),s_3}, \big(f(\mathbf{x})\big)_{(i+\delta,j+\delta),s_4}\Big)\Big)
\end{aligned}$$

für alle $(i,j) \in I$ und $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\} \times \{1,\dots,d_2\}}$, wobei wir die Schreibweise $\big(f(\mathbf{x})\big)_{(i,j),s} = 0$ für $(i,j) \notin I$ verwenden.

Beweis. Wir nehmen an, das vollverbundene neuronale Netz g_{net} sei gegeben durch

$$g_{net}(\mathbf{x}) = \sum_{i=1}^{r_{net}} w^{(L_{net})}_{1,i} g^{(L_{net})}_i(\mathbf{x}) + w^{(L_{net})}_{1,0},$$

wobei $g^{(L_{net})}_i$ rekursiv definiert ist durch

$$g^{(r)}_i(\mathbf{x}) = \sigma\left(\sum_{j=1}^{r_{net}} w^{(r-1)}_{i,j} g^{(r-1)}_j(\mathbf{x}) + w^{(r-1)}_{i,0}\right)$$

für $i \in \{1, \dots, r_{net}\}$, $r \in \{2, \dots, L_{net}\}$ und

$$g^{(1)}_i(\mathbf{x}) = \sigma\left(\sum_{j=1}^{4} w^{(0)}_{i,j} x^{(j)} + w^{(0)}_{i,0}\right) \quad (i \in \{1, \dots, r_{net}\}).$$

Wir wählen die Gewichte des faltenden Blocks

$$o^{(L_{net}+1)}_{(k',t+r_{net}),M} = \big(o_{(t+r_{net},t+r_{net}),M,1,\mathbf{w}_{L_{net}+1}} \circ \cdots \circ o_{(k',t+r_{net}),M,1,\mathbf{w}_1}\big),$$

indem wir die Gewichte von g_{net} verwenden. Die Gewichte von $o^{(L_{net}+1)}_{(k',t+r_{net}),M}$ bezeichnen wir mit

$$\mathbf{w}_1 = \left(\left(w^{(1)}_{i,j,s_{2,1},s_{2,2}}\right)_{1\le i,j\le M, s_{2,1}\in\{1,\dots,k'\}, s_{2,2}\in\{1,\dots,t+r_{net}\}}, \left(w^{(1)}_{s_{2,2}}\right)_{s_{2,2}\in\{1,\dots,t+r_{net}\}}\right)$$

und

$$\mathbf{w}_r = \left(\left(w^{(r)}_{i,j,s_{2,1},s_{2,2}}\right)_{1\le i,j\le M, s_{2,1},s_{2,2}\in\{1,\dots,t+r_{net}\}}, \left(w^{(r)}_{s_{2,2}}\right)_{s_{2,2}\in\{1,\dots,t+r_{net}\}}\right) \quad (r = 2,\dots,L_{net}+1)$$

Außerdem setzen wir im Folgenden

$$o^{(r)} = o_{(t+r_{net},t+r_{net}),M,1,\mathbf{w}_r} \circ \cdots \circ o_{(k',t+r_{net}),M,1,\mathbf{w}_1}$$

für $r = 1,\dots,L_{net}+1$. In der ersten Schicht setzen wir für $i \in \{1,\dots,r_{net}\}$ in Kanal $t+i$

$$w^{(1)}_{t_1,t_2,s,t+i} = 0,$$

falls $t_1, t_2 \notin \{1, \delta+1\}$ oder $s \notin \{s_1,\dots,s_4\}$ und wählen die einzigen Gewichte, die ungleich Null sind durch

$$\begin{aligned} w^{(1)}_{1,1,s_1,t+i} &= w^{(0)}_{i,1}, & w^{(1)}_{\delta+1,1,s_2,t+i} &= w^{(0)}_{i,2},\\ w^{(1)}_{1,\delta+1,s_3,t+i} &= w^{(0)}_{i,3}, & w^{(1)}_{\delta+1,\delta+1,s_4,t+i} &= w^{(0)}_{i,4}, \end{aligned}$$

und $w^{(1)}_{t+i} = w^{(0)}_{i,0}$. Dann gilt

$$\begin{aligned} &\big((o^{(1)} \circ f)(\mathbf{x})\big)_{(i_2,j_2),t+i}\\ &= \sigma\left(\sum_{s=1}^{k'} \sum_{\substack{t_1,t_2\in\{1,\dots,M\}\\(i_2+t_1-1,j_2+t_2-1)\in I}} w^{(1)}_{t_1,t_2,s,t+i}\cdot(f(\mathbf{x}))_{(i_2+t_1-1,j_2+t_2-1),s} + w^{(1)}_{t+i}\right)\\ &= \sigma\Big(w^{(0)}_{i,1}\cdot(f(\mathbf{x}))_{(i_2,j_2),s_1} + w^{(0)}_{i,2}\cdot(f(\mathbf{x}))_{(i_2+\delta,j_2),s_2} + w^{(0)}_{i,3}\cdot(f(\mathbf{x}))_{(i_2,j_2+\delta),s_3}\\ &\qquad + w^{(0)}_{i,4}\cdot(f(\mathbf{x}))_{(i_2+\delta,j_2+\delta),s_4} + w^{(0)}_{i,0}\Big)\\ &= g^{(1)}_i\Big((f(\mathbf{x}))_{(i_2,j_2),s_1}, (f(\mathbf{x}))_{(i_2+\delta,j_2),s_2}, (f(\mathbf{x}))_{(i_2,j_2+\delta),s_3}, (f(\mathbf{x}))_{(i_2+\delta,j_2+\delta),s_4}\Big) \end{aligned} \tag{3.30}$$

für alle $i \in \{1,\dots,r_{net}\}$, $(i_2,j_2) \in I$ und $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$. In den Schichten $r \in \{2,\dots,L_{net}\}$ in Kanal $t+i$ setzen wir

$$w^{(r)}_{t_1,t_2,s,t+i} = 0,$$

falls $(t_1,t_2) \neq (1,1)$ oder $s \in \{1,\dots,t\}$ und wählen

$$w^{(r)}_{1,1,t+j,t+i} = w^{(r-1)}_{i,j}, \quad w^{(r)}_{t+i} = w^{(r-1)}_{i,0} \quad (j \in \{1,\dots,r_{net}\})$$

für $i \in \{1, \dots, r_{net}\}$ als einzige Gewichte, welche ungleich Null sind. Damit erhalten wir

$$\left(o^{(r)} \circ f(\mathbf{x})\right)_{(i_2,j_2),t+i} = \sigma\left(\sum_{j=1}^{r_{net}} w_{i,j}^{(r-1)} \cdot \left(o^{(r-1)} \circ f(\mathbf{x})\right)_{(i_2,j_2),t+j} + w_{i,0}^{(r-1)}\right)$$

für alle $i \in \{1, \dots, r_{net}\}$, $r \in \{2, \dots, L_{net}\}$, $(i_2, j_2) \in I$ und $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$. Wegen Gleichung (3.30) und der Definition von $g_i^{(r)}$ gilt dann

$$\left(o^{(r)} \circ f(\mathbf{x})\right)_{(i_2,j_2),t+i} = g_i^{(r)}\Big((f(\mathbf{x}))_{(i_2,j_2),s_1}, (f(\mathbf{x}))_{(i_2+\delta,j_2),s_2}, (f(\mathbf{x}))_{(i_2,j_2+\delta),s_3}, (f(\mathbf{x}))_{(i_2+\delta,j_2+\delta),s_4}\Big)$$

für alle $i \in \{1, \dots, r_{net}\}$, $r \in \{2, \dots, L_{net}\}$, $(i_2, j_2) \in I$ und $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$. In Schicht $L_{net}+1$ in Kanal s_5 setzen wir nun

$$w_{t_1,t_2,s,s_5}^{(L_{net}+1)} = 0,$$

falls $(t_1, t_2) \neq (1,1)$ oder $s \in \{1, \dots, t\}$ und wählen

$$w_{1,1,t+i,s_5}^{(L_{net}+1)} = w_{1,i}^{(L_{net})} \quad \text{sowie} \quad w_{s_5}^{(L_{net}+1)} = w_{1,0}^{(L_{net})}$$

für $i \in \{1, \dots, r_{net}\}$ als einzige Gewichte, welche ungleich Null sind. Wir erhalten dann

$$\begin{aligned}
&\left(o_{(t+r_{net},t+r_{net}),M}^{(L_{net}+1)} \circ f(\mathbf{x})\right)_{(i_2,j_2),s_5} \\
&= \sigma\left(\sum_{i=1}^{r_{net}} w_{1,i}^{(L_{net})} \cdot \left(o^{(L_{net})} \circ f(\mathbf{x})\right)_{(i_2,j_2),t+i} + w_{1,0}^{(L_{net})}\right) \\
&= \sigma\left(g_{net}\Big((f(\mathbf{x}))_{(i_2,j_2),s_1}, (f(\mathbf{x}))_{(i_2+\delta,j_2),s_2}, (f(\mathbf{x}))_{(i_2,j_2+\delta),s_3}, (f(\mathbf{x}))_{(i_2+\delta,j_2+\delta),s_4}\Big)\right)
\end{aligned}$$

für alle $(i_2, j_2) \in I$ und $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$. □

Beweis von Lemma 4. Im Beweis verwenden wir, dass

$$\sigma(x) = \max\{x, 0\} = x$$

für $x \geq 0$ gilt. Dies ermöglicht es uns, nichtnegative Werte $x_{(i,j),s_1} \geq 0$, welche durch eine faltende Schicht in Kanal s_1 an der Position (i, j) berechnet wurden, an die nächste Schicht durch eine faltende Schicht $o_{(k',k),M,1,\mathbf{w}}$ mittels

$$\left(o_{(k',k),M,1,\mathbf{w}}(\mathbf{x})\right)_{(i,j),s_2} = \sigma\Big(x_{(i,j),s_1}\Big) = x_{(i,j),s_1} \tag{3.31}$$

weiterzugeben. Hierbei wird ein Gewichtsvektor $\mathbf{w}$ verwendet, dessen Gewichte entsprechend aus der Menge $\{0, 1\}$ in Kanal s_2 gewählt werden.

Die Klasse der faltenden neuronalen Netze $\mathcal{F}_3\left(\left(L, k, \mathbf{M}, z, \mathbf{s}, \mathbf{A}\right)\right)$ besteht aus Funktionen der Form

$$f_{out}^{(\mathbf{A})} \circ o_{(k,k),M_L}^{(z)} \circ f_{max}^{(n_{L-1})} \circ o_{(k,k),M_{L-1}}^{(z)} \circ \cdots \circ f_{max}^{(n_1)} \circ o_{(1,k),M_1}^{(z)}, \tag{3.32}$$

wobei die Funktionen $o_{(k_r',k),M_r}^{(z)} : \mathbb{R}^{I_r\times\{1,\dots,k_r'\}} \to \mathbb{R}^{I_r\times\{1,\dots,k\}}$ wie in (3.29) definierte Blöcke, bestehend aus z faltenden Schichten bezeichnen. Dabei ergeben sich die Indexmengen $I_1, \dots, I_L$ jeweils aus der Eingabe des entsprechenden faltenden Blocks und für die Anzahl der Eingangskanäle gilt

$$k_r' = \begin{cases} 1, & \text{für } r = 1, \\ k, & \text{sonst.} \end{cases}$$

Die Idee des Beweises ist es, nacheinander die Werte

$$\big((\bar{y}_{r,s}(\mathbf{x}))_{(i,j)}\big)_{(i,j)\in\{1,\dots,d_1(r)\}\times\{1,\dots,d_2(r)\}},$$

definiert durch Gleichung (3.24), in unterschiedlichen Kanälen des Blocks $o^{(z)}_{(k'_r,k),M_r}$ entsprechend des Index $s \in \{1,\dots,b_r\}$ durch Anwendung von Lemma 5 zu berechnen. Sind Werte einmal berechnet, können sie unter Verwendung der Gewichte aus Gleichung (3.31) an die nächste faltende Schicht weitergegeben werden. Die Werte

$$\big((\bar{z}_{r,s}(\mathbf{x}))_{(i,j)}\big)_{(i,j)\in\{1,\dots,\lceil d_1(r)/n_r\rceil\}\times\{1,\dots,\lceil d_2(r)/n_r\rceil)\}}$$

aus Gleichung (3.25) können dann durch die lokale Max-Pooling Schicht $f^{(n_r)}_{max}$ berechnet werden.

Da wir das Resultat durch Induktion über die l faltenden Blöcke zeigen werden, definieren wir eine rekursive Darstellung von (3.32). Wir definieren $f^{(0)} : \mathbb{R}^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}\times\{1\}}$ durch

$$\big(f^{(0)}(\mathbf{x})\big)_{(i,j),1} = x_{i,j}$$

für $(i,j) \in \{1,\dots,d_1\}\times\{1,\dots,d_2\}$ und definieren rekursiv die Funktionen

$$g^{(r)} = o^{(z)}_{(k'_r,k),M_r} \circ f^{(r-1)} \quad (r = 1,\dots,l)$$

und

$$f^{(r)} = f^{(n_r)}_{max} \circ g^{(r)} \quad (r = 1,\dots,l)$$

(mit $n_l = 1$). Wir zeigen durch Induktion über r, dass wir die Gewichte von $f^{(l)}$ so wählen können, sodass die folgende Bedingung für alle $r \in \{0,\dots,l\}$ erfüllt ist:

($*$) Für alle $s \in \{1,\dots,b_r\}$, $(i,j) \in \{1,\dots,\lceil d_1(r)/n_r\rceil\}\times\{1,\dots,\lceil d_2(r)/n_r\rceil\}$ und $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$ gilt, dass

$$\Big(f^{(r)}(\mathbf{x})\Big)_{(i,j),s} = \big(\bar{z}_{r,s}(\mathbf{x})\big)_{(i,j)}.$$

Wegen $b_0 = n_0 = 1$ ist die Bedingung ($*$) für $r = 0$ erfüllt. Nun nehme an, die Bedingung ($*$) gelte für $r \in \{0,\dots,l-1\}$. Wir zeigen in zwei Schritten, dass dann die Bedingung ($*$) für $r+1$ erfüllt ist.

Im ersten Schritt zeigen wir, dass ein faltender Block $o^{(z)}_{(k'_{r+1},k),M_{r+1}}$ existiert, sodass

$$\begin{aligned}\Big(g^{(r+1)}(\mathbf{x})\Big)_{(i,j),s} = \sigma\Big(\bar{g}_{r+1,s}\Big(&\big(\bar{z}_{r,r_1(r+1,s)}(\mathbf{x})\big)_{(i,j)}, \big(\bar{z}_{r,r_2(r+1,s)}(\mathbf{x})\big)_{(i+\delta_r,j)},\\ &\big(\bar{z}_{r,r_3(r+1,s)}(\mathbf{x})\big)_{(i,j+\delta_r)}, \big(\bar{z}_{r,r_4(r+1,s)}(\mathbf{x})\big)_{(i+\delta_r,j+\delta_r)}\Big)\Big)\end{aligned} \tag{3.33}$$

für alle $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$, $(i,j) \in \{1,\dots,\lceil d_1(r)/n_r\rceil - \delta_r\}\times\{1,\dots,\lceil d_2(r)/n_r\rceil - \delta_r\}$ und $s \in \{1,\dots,b_{r+1}\}$. Ein faltender Block $o^{(z)}_{(k'_{r+1},k),M_{r+1}}$ hat die Form

$$o^{(z)}_{(k'_{r+1},k),M_{r+1}} = o_{(k,k),M_{r+1},\mathbf{w}_z} \circ \dots \circ o_{(k'_{r+1},k),M_{r+1},\mathbf{w}_1}.$$

Da wir die Gewichtsvektoren $\mathbf{w}_1,\dots,\mathbf{w}_z$ Schicht für Schicht wählen werden, setzen wir

$$o^{(t)} = o_{(k,k),M_{r+1},\mathbf{w}_t} \circ \dots \circ o_{(k'_{r+1},k),M_{r+1},\mathbf{w}_1}$$

für $t = 1, \ldots, z$. Wir werden Lemma 5 für die Berechnung jedes Netzwerks

$$\sigma\Bigg(\bar{g}_{r+1,s}\Big(\big(\bar{z}_{r,r_1(r+1,s)}(\mathbf{x})\big)_{(i,j)}, \big(\bar{z}_{r,r_2(r+1,s)}(\mathbf{x})\big)_{(i+\delta_r,j)}, \\ \big(\bar{z}_{r,r_3(r+1,s)}(\mathbf{x})\big)_{(i,j+\delta_r)}, \big(\bar{z}_{r,r_4(r+1,s)}(\mathbf{x})\big)_{(i+\delta_r,j+\delta_r)}\Big)\Bigg) \tag{3.34}$$

für $s = 1, \ldots, b_{r+1}$ verwenden und berechnete Werte in den entsprechenden Kanälen

$$1, \ldots, b_{r+1}$$

speichern, indem wir Gleichung (3.31) anwenden. Mit der Idee, bereits berechnete Werte in die nächste Schicht mittels Gleichung (3.31) weiterzugeben, und wegen der Induktionshypothese $(*)$, können wir die Gewichte unseres faltenden Blocks in den Kanälen

$$b_{max} + 1, \ldots, b_{max} + b_r$$

so wählen, dass

$$\Big(\big(o^{(t)} \circ f^{(r)}\big)(\mathbf{x})\Big)_{(i,j),b_{max}+s} = \big(\bar{z}_{r,s}(\mathbf{x})\big)_{(i,j)}$$

für alle $\mathbf{x} \in [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}}$, $t \in \{1, \ldots, b_{max}\cdot(L_{net}+1)\}$, $s \in \{1, \ldots, b_r\}$ und $(i,j) \in \{1, \ldots, \lceil d_1(r)/n_r \rceil\} \times \{1, \ldots, \lceil d_2(r)/n_r \rceil\}$. Verwenden wir nun Lemma 5 mit den Parametern

$$s_m = I_{\mathbb{N}\setminus\{1\}}(s) \cdot b_{max} + r_m(r+1, s)$$

für $m \in \{1, \ldots, 4\}$ und $s_5 = s$, können wir die Werte (3.34) in den Schichten

$$(s-1)\cdot(L_{net}+1) + 1, \ldots, s\cdot(L_{net}+1)$$

berechnen, indem wir die Gewichte in den entsprechenden Kanälen

$$s, 2\cdot b_{max} + 1, \ldots, 2\cdot b_{max} + r_{net}$$

so wählen, dass

$$\Big(\big(o^{(s\cdot(L_{net}+1))} \circ f^{(r)}\big)(\mathbf{x})\Big)_{(i,j),s} \\ = \sigma\Bigg(\bar{g}_{r+1,s}\Big(\big(\bar{z}_{r,r_1(r+1,s)}(\mathbf{x})\big)_{(i,j)}, \big(\bar{z}_{r,r_2(r+1,s)}(\mathbf{x})\big)_{(i+\delta_r,j)}, \\ \big(\bar{z}_{r,r_3(r+1,s)}(\mathbf{x})\big)_{(i,j+\delta_r)}, \big(\bar{z}_{r,r_4(r+1,s)}(\mathbf{x})\big)_{(i+\delta_r,j+\delta_r)}\Big)\Bigg)$$

für alle $\mathbf{x} \in [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}}$, $(i,j) \in \{1, \ldots, \lceil d_1(r)/n_r \rceil - \delta_r\} \times \{1, \ldots, \lceil d_2(r)/n_r \rceil - \delta_r\}$ und $s \in \{1, \ldots, b_{r+1}\}$. Wurde ein Wert in Schicht $s \cdot (L_{net}+1)$ für $s \in \{1, \ldots, b_{r+1}\}$ berechnet, wird er unter Verwendung von Gleichung (3.31) in die nächste Schicht weitergegeben, sodass

$$\Big(\big(o^{(b_{max}\cdot(L_{net}+1))} \circ f^{(r)}\big)(\mathbf{x})\Big)_{(i,j),s}$$

$$= \sigma\Bigg(\bar{g}_{r+1,s}\Big(\big(\bar{z}_{r,r_1(r+1,s)}(\mathbf{x})\big)_{(i,j)}, \big(\bar{z}_{r,r_2(r+1,s)}(\mathbf{x})\big)_{(i+\delta_r,j)},$$
$$\big(\bar{z}_{r,r_3(r+1,s)}(\mathbf{x})\big)_{(i,j+\delta_r)}, \big(\bar{z}_{r,r_4(r+1,s)}(\mathbf{x})\big)_{(i+\delta_r,j+\delta_r)}\Big)\Bigg)$$

für alle $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$, $s \in \{1,\dots,b_{r+1}\}$ und $(i,j) \in \{1,\dots,\lceil d_1(r)/n_r\rceil - \delta_r\} \times \{1,\dots,\lceil d_2(r)/n_r\rceil - \delta_r\}$, was Gleichung (3.33) impliziert.

Im zweiten Schritt werden wir aus Gleichung (3.33) folgern, dass Bedingung $(*)$ für $r+1$ erfüllt ist. Zunächst bemerken wir, dass $d_1(r+1) = \lceil d_1(r)/n_r\rceil - \delta_r$ und $d_2(r+1) = \lceil d_2(r)/n_r\rceil - \delta_r$. Da sich $d_1(r+1)$ und $d_2(r+1)$ durch n_{r+1} teilen lassen (siehe Lemma 6 unten) ergibt sich

$$N_{(i,j)}^{(r+1)} = \{(i-1)\cdot n_{r+1}+1,\dots,i\cdot n_{r+1}\} \times \{(j-1)\cdot n_{r+1}+1,\dots,j\cdot n_{r+1}\}$$

für alle $(i,j) \in \{1,\dots,\lceil d_1(r+1)/n_{r+1}\rceil\} \times \{1,\dots,\lceil d_2(r+1)/n_{r+1}\rceil\}$ und wir erhalten deshalb zusammen mit Gleichung (3.33), dass

$$\begin{aligned}\Big(f^{(r+1)}(\mathbf{x})\Big)_{(i,j),s} &= \Big(\big(f_{max}^{(n_{r+1})} \circ g^{(r+1)}\big)(\mathbf{x})\Big)_{(i,j),s}\\ &= \max_{(i_2,j_2)\in N_{(i,j)}^{(r+1)}} \big(g^{(r+1)}(\mathbf{x})\big)_{(i_2,j_2),s}\\ &\overset{(3.33)}{=} \max_{(i_2,j_2)\in N_{(i,j)}^{(r+1)}} \big(\bar{y}_{r+1,s}(\mathbf{x})\big)_{(i_2,j_2)}\\ &= \big(\bar{z}_{r+1,s}(\mathbf{x})\big)_{(i,j)}\end{aligned}$$

für alle $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$, $s \in \{1,\dots,b_{r+1}\}$ und $(i,j) \in \{1,\dots,\lceil d_1(r+1)/n_{r+1}\rceil\} \times \{1,\dots,\lceil d_2(r+1)/n_{r+1}\rceil\}$, was Bedingung $(*)$ für alle $r \in \{0,\dots,l\}$ impliziert. Als Nächstes wählen wir die Gewichte

$$\mathbf{w}_{out} = (w_s)_{s\in\{1,\dots,k\}},$$

indem wir $w_1 = 1$ und $w_{s_2} = 0$ für $s_2 \in \{2,\dots,k\}$ setzen. Es folgt, dass sich die Ausgabe unseres Netzwerks berechnet durch

$$\begin{aligned}m_{net}(\mathbf{x}) &= \Big(f_{out}^{(\mathbf{A})} \circ f^{(l)}\Big)(\mathbf{x})\\ &= \max\Big\{\big(\bar{z}_{l,1}(\mathbf{x})\big)_{(i,j)} : (i,j) \in \{1,\dots,d_1(l)\}\times\{1,\dots,d_2(l)\}\Big\}\\ &= \bar{m}(\mathbf{x}).\end{aligned}$$

für alle $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$. □

Lemma 6. *Es sei $l \in \mathbb{N}$ und es seien $n_0, n_1, \dots, n_l \in \{2^0, 2^1, \dots, 2^{l-1}\}$ mit $n_0 = n_l = 1$, sodass*

$$\delta_k = 2^k / \prod_{i=0}^{k} n_i \geq 1 \quad (k = 0,\dots,l). \tag{3.35}$$

Weiterhin sei

$$d = \prod_{i=1}^{l-1} n_i \cdot m - 1$$

für ein $m \in \mathbb{N}$, sodass $d \geq 2^l + \prod_{i=1}^{l-1} n_i - 1$. Für $k = 1, \dots, l$ definieren wir rekursiv

$$d(k) = \left\lceil \frac{d(k-1)}{n_{k-1}} \right\rceil - \delta_{k-1},$$

wobei wir $d(0) = d$ setzen.

Es gilt dann für alle $k \in \{0, \dots, l\}$, dass

$$d(k) = n_k \cdot \left(\prod_{i=k+1}^{l} n_i \cdot m - \delta_k \right) = n_k \cdot \left(\frac{d - 2^k + 1}{\prod_{i=0}^{k} n_i} \right), \tag{3.36}$$

wobei wir das leere Produkt als 1 definieren. Insbesondere ist dann $d(k)$ durch n_k teilbar.

Beweis. Wir zeigen die Behauptung durch Induktion über k. Für $k = 0$ gilt die Behauptung nach Definition. Nehme nun an, die Behauptung gelte für $k \in \{0, \dots, l-1\}$. Dann folgt

$$\begin{aligned} d(k+1) &= \left\lceil \frac{d(k)}{n_k} \right\rceil - \delta_k \\ &= \prod_{i=k+1}^{l} n_i \cdot m - 2 \cdot \delta_k \\ &= n_{k+1} \cdot \left(\prod_{i=k+2}^{l} n_i \cdot m - \frac{2^{k+1}}{\prod_{i=0}^{k+1} n_i} \right) \\ &= n_{k+1} \cdot \left(\prod_{i=k+2}^{l} n_i \cdot m - \delta_{k+1} \right). \end{aligned}$$

Annahme (3.35) impliziert wegen der Wahl der $n_0, \dots, n_{k+1}$, dass $\delta_{k+1} \in \mathbb{N}$. Die obige Gleichheit impliziert dann, dass $d(k+1)$ durch n_{k+1} teilbar ist. Die zweite Gleichheit in Gleichung (3.36) folgt durch Erweitern mit $\prod_{i=0}^{k} n_i$ und der Definition von d sowie δ_k. Wegen der Voraussetzung $d \geq 2^l + \prod_{i=1}^{l-1} n_i - 1$ sind die $d(k)$ dann insbesondere positiv für alle $k \in \{0, \dots, l\}$. □

3.4.2. Eine Verbindung zwischen verschiedenen faltenden neuronalen Netzen

Das Ziel dieses Abschnitts ist es, die folgende Verbindung zwischen den in Abschnitt 3.1 eingeführten Netzwerkarchitekturen $\mathcal{F}_3(\boldsymbol{\theta}_3)$, $\mathcal{F}_4(\boldsymbol{\theta}_4)$ und $\mathcal{F}_5(\boldsymbol{\theta}_5)$ zu zeigen, welche wir im Beweis von Theorem 3.2 verwenden werden.

Lemma 7. *Es seien $d_1, d_2, L, z, k \in \mathbb{N}$ und $s_0, \dots, s_{L-1} \in \{2^0, \dots, 2^{L-1}\}$ mit*

$$\prod_{i=0}^{r} s_i \leq 2^r \quad (r = 1, \dots, L-1). \tag{3.37}$$

Setze $\mathbf{s} = (s_1, \dots, s_{L-1})$, $\bar{k} = 2 \cdot k + 4$, $\bar{z} = 3 \cdot k \cdot \log_2(\max\{s_1 \dots, s_{L-1}\}) + z$ und setze

$$s = \prod_{i=1}^{L-1} s_i, \qquad M_r = \frac{2^{r-1}}{\prod_{i=0}^{r-1} s_i} + 1 \quad \textit{und} \quad \bar{M}_{(r-1)\cdot\bar{z}+1}, \dots, \bar{M}_{r\cdot\bar{z}} = 2^{r-1} + 1 \quad (r = 1, \dots, L).$$

Weiterhin seien $\mathcal{F}_3(\boldsymbol{\theta}_3)$, $\mathcal{F}_4(\boldsymbol{\theta}_4)$ und $\mathcal{F}_5(\boldsymbol{\theta}_5)$ die Funktionsklassen aus Abschnitt 3.1 mit Parametern

$$\boldsymbol{\theta}_3 = (L, k, \mathbf{M}, z, \mathbf{s}, \mathbf{A}), \quad \boldsymbol{\theta}_4 = (L, \bar{k}, \mathbf{M}, \bar{z}, \mathbf{s}, \mathbf{A}), \quad \boldsymbol{\theta}_5 = (L \cdot \bar{z}, \bar{k}, \bar{\mathbf{M}}, s, \mathbf{A})$$

mit beliebigen Ausgabeschranken $\mathbf{A} = (A_1, A_1', A_2, A_2')$. Dann gilt

$$\mathcal{F}_3(\boldsymbol{\theta}_3) \subset \mathcal{F}_4(\boldsymbol{\theta}_4) \subset \mathcal{F}_5(\boldsymbol{\theta}_5).$$

Der Beweis von Lemma 7 erfolgt am Ende dieses Abschnitts. Um zu zeigen, dass wir ein faltendes neuronales Netz mit mehreren Max-Pooling Schichten (aus $\mathcal{F}_3(\boldsymbol{\theta}_3)$) als ein faltendes neuronales Netz mit mehreren Subsampling Schichten (aus $\mathcal{F}_4(\boldsymbol{\theta}_4)$) darstellen können, verwenden wir das folgende Lemma.

Lemma 8. *Es seien $k, i_1, i_2 \in \mathbb{N}$, $I = \{1, \dots, i_1\} \times \{1, \dots, i_2\}$ und es sei*

$$f : \mathbb{R}^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}_0^{+I\times\{1,\dots,2\cdot k+4\}}$$

eine Funktion. Weiterhin sei $n \in \mathbb{N}_0$ und $M \in \mathbb{N}$ mit $M \geq 2^{n-1} + 1$. Dann existiert ein faltender Block

$$o^{(z)}_{(2\cdot k+4, 2\cdot k+4), M} : \mathbb{R}^{I\times\{1,\dots,2\cdot k+4\}} \to \mathbb{R}^{I\times\{1,\dots,2\cdot k+4\}}$$

(definiert wie in Gleichung (3.29)) mit $z = 3 \cdot n \cdot k$ faltenden Schichten, sodass

$$\Big(\big(f^{(2^n)}_{sub} \circ o^{(z)}_{(2\cdot k+4,2\cdot k+4),M} \circ f\big)(\mathbf{x})\Big)_{(i,j),s} = \Big(\big(f^{(2^n)}_{max} \circ f\big)(\mathbf{x})\Big)_{(i,j),s}$$

für alle $s \in \{1, \dots, k\}$, $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$ und $(i,j) \in \{1, \dots, \lceil i_1/2^n \rceil\} \times \{1, \dots, \lceil i_2/2^n \rceil\}$ gilt.

Beweis. O.B.d.A. können wir annehmen, dass $n > 0$ ist (im Fall $n = 0$ ist die Behauptung trivial, da $f^{(1)}_{sub} = f^{(1)}_{max}$). Wir verwenden im Beweis die Funktion $g_{max} : \mathbb{R}^4 \to \mathbb{R}$, welche das Maximum ihrer Argumente berechnet, im Fall, dass diese nichtnegativ sind. Um die Funktion g_{max} als neuronales Netz zu darzustellen, berechnen wir für $a \geq 0$ und $b \in \mathbb{R}$

$$\max\{a, b\} = \max\{b - a, 0\} + \max\{a, 0\} = \sigma(b - a) + \sigma(a). \tag{3.38}$$

Die Funktion g_{max} kann dann als neuronales Netz mit zwei verdeckten Schichten und höchstens vier Neuronen pro Schicht wie folgt dargestellt werden:

$$g_{max}(\mathbf{x}) = \sigma\Big(\sigma(x_2 - x_1) + \sigma(x_1) - \big(\sigma(x_4 - x_3) + \sigma(x_3)\big)\Big) + \sigma\Big(\sigma(x_4 - x_3) + \sigma(x_3)\Big).$$

Für $\mathbf{x} \in \mathbb{R}_0^{+4}$ impliziert Gleichung (3.38)

$$g_{max}(\mathbf{x}) = \max\{\max\{x_1, x_2\}, \max\{x_3, x_4\}\} = \max\{x_1, x_2, x_3, x_4\}.$$

Die Idee ist es, Lemma 5 mehrmals mit dem neuronalen Netz $g_{net} = g_{max}$ und wachsenden Werten für δ zu verwenden, um die Maximumsberechnung der lokalen Max-Pooling Schicht zu imitieren. Da wir die Gewichtsvektoren $\mathbf{w}_1, \dots, \mathbf{w}_z$ unseres faltenden Blocks $o^{(z)}_{(2\cdot k+4,2\cdot k+4),M}$ Schicht für Schicht definieren werden, führen wir die Funktion

$$o^{(0)} : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}_0^{+I\times\{1,\dots,2\cdot k+4\}}$$

mit $o^{(0)} = f$ ein und definieren rekursiv die Funktionen $o^{(t)} : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}^{I\times\{1,\dots,2\cdot k+4\}}$ durch

$$o^{(t)} = o_{(2\cdot k+4,2\cdot k+4),M,1,\mathbf{w}_t} \circ o^{(t-1)}$$

für $t = 1, \dots, z$. Wir zeigen durch Induktion über r, dass wir die Gewichte $\mathbf{w}_1, \dots, \mathbf{w}_z$ von $o^{(z)}_{(2\cdot k+4,2\cdot k+4),M}$ so wählen können, dass die folgende Bedingung $(*)$ für alle $r \in \{0, \dots, n\}$ erfüllt ist:

($*$) Für alle $s \in \{1, \dots, k\}$, $(i,j) \in I$ und $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$ gilt

$$\left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i,j),s} = \max_{(i_2,j_2)\in\{i,\dots,i+2^r-1\}\times\{j,\dots,j+2^r-1\}\cap I} \left(f(\mathbf{x})\right)_{(i_2,j_2),s},$$

wobei wir $\left(f(\mathbf{x})\right)_{(i_2,j_2),s} = 0$ setzen, falls $(i_2, j_2) \notin I$.

Für den Fall $r = 0$ folgt die Behauptung direkt aus der Definition von $o^{(0)}$. Wir nehmen nun an, Bedingung ($*$) sei für ein $r \in \{0, \dots, n-1\}$ erfüllt. Die Idee ist es, nacheinander Lemma 5 für die Berechnung jedes Netzwerks

$$\sigma\Bigg(g_{max}\Big(\left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i,j),s}, \left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i+2^r,j),s}, \left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i,j+2^r),s}, \left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i+2^r,j+2^r),s}\Big)\Bigg) \tag{3.39}$$

für $s = 1, \dots, k$ zu verwenden und die berechneten Werte in den entsprechenden Kanälen

$$1, \dots, k$$

zu speichern, indem Werte an die nächste Schicht mittels Gleichung (3.31) weitergegeben werden. Wegen Gleichung (3.31) und der Induktionshypothese ($*$) können wir die Gewichte

$$\mathbf{W}_{r\cdot 3\cdot k+1}, \dots, \mathbf{W}_{(r+1)\cdot 3\cdot k}$$

unseres faltenden neuronalen Blocks in den Kanälen

$$k+1, \dots, 2\cdot k$$

so wählen, dass

$$\left(o^{(t)}(\mathbf{x})\right)_{(i,j),k+s} = \left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i,j),s}$$

für alle $t \in \{r\cdot 3\cdot k+1, \dots, (r+1)\cdot 3\cdot k\}$, $s \in \{1, \dots, k\}$, $(i,j) \in I$ und $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$. Verwenden wir nun Lemma 5 für alle $s \in \{1, \dots, k\}$ mit den Parametern

$$s_m = I_{\mathbb{N}\setminus\{1\}}(s)\cdot k + s$$

für $m \in \{1, \dots, 4\}$ und $s_5 = s$, so können wir die Werte (3.39) in den Schichten

$$3\cdot r\cdot k + (s-1)\cdot 3 + 1, \dots, 3\cdot r\cdot k + s\cdot 3$$

berechnen, indem wir die Gewichte in den Kanälen

$$s, 2\cdot k+1, \dots, 2\cdot k+4$$

entsprechend wählen, sodass

$$\left(\left(o^{(3\cdot r\cdot k+3\cdot s)}\right)(\mathbf{x})\right)_{(i,j),s} = \sigma\Bigg(g_{max}\Big(\left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i,j),s}, \left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i+2^r,j),s}, \left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i,j+2^r),s}, \left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i+2^r,j+2^r),s}\Big)\Bigg)$$

für alle $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$, $s \in \{1,\dots,k\}$ und $(i,j) \in I$, wobei wir

$$\left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i,j),s} = 0$$

für $(i,j) \notin I$ setzen. Wurde ein Wert in Schicht $3 \cdot r \cdot k + 3 \cdot s$ berechnet, wird er unter Verwendung der Gewichte aus Gleichung (3.31) an die nächste Schicht weitergegeben, sodass

$$\begin{aligned}\left(\left(o^{(3\cdot(r+1)\cdot k)}\right)(\mathbf{x})\right)_{(i,j),s} &= \sigma\Bigg(g_{max}\Big(\left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i,j),s}, \left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i+2^r,j),s}, \\ &\qquad \left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i,j+2^r),s}, \left(o^{(r\cdot 3\cdot k)}(\mathbf{x})\right)_{(i+2^r,j+2^r),s}\Big)\Bigg) \\ &= \max_{(i_2,j_2)\in\{i,\dots,i+2^{r+1}-1\}\times\{j,\dots,j+2^{r+1}-1\}\cap I} \left(f(\mathbf{x})\right)_{(i_2,j_2),s}\end{aligned}$$

für alle $(i,j) \in I$ und $s \in \{1,\dots,k\}$ gilt, wobei die letzte Zeile aus der Induktionshypothese folgt. Letztendlich folgt die Behauptung aus der Definition der Subsampling Schicht f_{sub} und der lokalen Max-Pooling Schicht f_{max}. □

Zur Darstellung eines faltenden neuronalen Netzes mit mehreren Subsampling Schichten durch ein faltendes neuronales Netz mit nur einer einzigen Subsampling Schicht (aus $\mathcal{F}_5(\boldsymbol{\theta}_5)$) wird das folgende Lemma verwendet.

Lemma 9. *Es seien $i_1, i_2, n, k \in \mathbb{N}$ und*

$$I = \{1,\dots,i_1\}\times\{1,\dots,i_2\} \text{ sowie } \tilde{I} = \{1,\dots,\lceil i_1/n\rceil\}\times\{1,\dots,\lceil i_2/n\rceil\}.$$

Außerdem sei

$$f : \mathbb{R}^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}_0^{+\,I\times\{1,\dots,k\}}$$

eine Funktion und es sei

$$o^{(z)}_{(k,k),M} : \mathbb{R}^{\tilde{I}\times\{1,\dots,k\}} \to \mathbb{R}^{\tilde{I}\times\{1,\dots,k\}}$$

ein faltender Block, definiert wie in (3.29) *mit $z \in \mathbb{N}$ Schichten und der Filtergröße $M \in \mathbb{N}$ mit festen Gewichtsvektoren $\mathbf{w}_1,\dots,\mathbf{w}_z$. Dann existiert ein faltender Block*

$$\tilde{o}^{(z)}_{(k,k),(M-1)\cdot n+1} : \mathbb{R}^{I\times\{1,\dots,k\}} \to \mathbb{R}^{I\times\{1,\dots,k\}},$$

sodass

$$f^{(n)}_{sub} \circ \tilde{o}^{(z)}_{(k,k),(M-1)\cdot n+1} \circ f = o^{(z)}_{(k,k),M} \circ f^{(n)}_{sub} \circ f.$$

Beweis. Die Idee besteht darin, in jeder faltenden Schicht von $\tilde{o}^{(z)}_{(k,k),(M-1)\cdot n+1}$ nur auf die Positionen zuzugreifen, die nach Anwendung der Subsampling Schicht $f^{(n)}_{sub}$ auf die vorherige Schicht übrig bleiben würden. Wir erreichen dies, indem wir die Gewichtsvektoren $\tilde{\mathbf{w}}_1,\dots,\tilde{\mathbf{w}}_z$ von $\tilde{o}^{(z)}_{(k,k),(M-1)\cdot n+1}$ so wählen, dass die entsprechenden Filter der Gewichte nur Werte ungleich Null auf der Indexmenge $J \subset \{1,\dots,(M-1)\cdot n+1\}^2$ haben, welche durch

$$J = \{((t_1-1)\cdot n+1, (t_2-1)\cdot n+1) : t_1, t_2 \in \{1,\dots,M\}\}.$$

definiert ist. Die Werte der Gewichtsvektoren $\tilde{\mathbf{w}}_1, \ldots, \tilde{\mathbf{w}}_z$, die ungleich Null sind, entsprechen den Gewichten der Gewichtsvektoren $\mathbf{w}_1, \ldots, \mathbf{w}_z$ des faltenden Blocks $o^{(z)}_{(k,k),M}$. Wir merken an, dass die Gewichtsvektoren $\mathbf{w}_1, \ldots, \mathbf{w}_z$ folgende Form haben:

$$\mathbf{w}_t = \left(\left(w^{(t)}_{i,j,s_1,s_2} \right)_{1 \leq i,j \leq M, s_1, s_2 \in \{1,\ldots,k\}}, \left(w^{(t)}_{s_2} \right)_{s_2 \in \{1,\ldots,k\}} \right) \quad (t = 1, \ldots, z).$$

Für $(i,j) \in \{1, \ldots, (M-1) \cdot n + 1\}^2$, $t \in \{1, \ldots, z\}$ und $s_1, s_2 \in \{1, \ldots, k\}$ setzen wir

$$\tilde{w}^{(t)}_{i,j,s_1,s_2} = \begin{cases} w^{(t)}_{\frac{i-1}{n}+1, \frac{j-1}{n}+1, s_1, s_2}, & \text{falls } (i,j) \in J \\ 0, & \text{sonst} \end{cases} \tag{3.40}$$

sowie

$$\tilde{w}^{(t)}_{s_2} = w^{(t)}_{s_2}.$$

Da wir die Behauptung durch Induktion über die Schichten des faltenden Blocks zeigen werden, definieren wir die Funktionen $\tilde{o}^{(0)} = f$ und $o^{(0)} = f^{(n)}_{sub} \circ f$ sowie die Funktionen

$$o^{(t)} = o_{(k,k),M,1,\mathbf{w}_t} \circ o^{(t-1)} \text{ und } \tilde{o}^{(t)} = \tilde{o}_{(k,k),(M-1)\cdot n+1,1,\tilde{\mathbf{w}}_t} \circ \tilde{o}^{(t-1)} \quad (t = 1, \ldots, z).$$

Wir zeigen dann durch Induktion über t, dass

$$\left(\tilde{o}^{(t)}(\mathbf{x}) \right)_{((i-1)\cdot n+1,(j-1)\cdot n+1),s} = \left(o^{(t)}(\mathbf{x}) \right)_{(i,j),s}$$

für alle $(i,j) \in \tilde{I}$, $\mathbf{x} \in \mathbb{R}^{\{1,\ldots,d_1\} \times \{1,\ldots,d_2\}}$, $s \in \{1, \ldots, k\}$ und $t \in \{0, \ldots, z\}$ gilt. Für $t = 0$ folgt die Behauptung direkt aus der Definition der Subsampling Schicht $f^{(n)}_{sub}$. Nun nehmen wir an, die Behauptung gelte für ein $t \in \{0, \ldots, z-1\}$. Für $(i,j) \in \tilde{I}$, $s \in \{1, \ldots, k\}$ und $\mathbf{x} \in \mathbb{R}^{\{1,\ldots,d_1\} \times \{1,\ldots,d_2\}}$ erhalten wir wegen der Induktionshypothese und der Definition der Gewichte (3.40)

$$\begin{aligned}
&\left(\tilde{o}^{(t+1)}(\mathbf{x}) \right)_{((i-1)\cdot n+1,(j-1)\cdot n+1),s} \\
&= \left(\left(\tilde{o}_{(k,k),(M-1)\cdot n+1,1,\mathbf{w}_{t+1}} \circ \tilde{o}^{(t)} \right)(\mathbf{x}) \right)_{((i-1)\cdot n+1,(j-1)\cdot n+1),s} \\
&= \sigma \left(\sum_{s_1=1}^{k} \sum_{\substack{t_1,t_2 \in J \\ ((i-1)\cdot n+t_1,(j-1)\cdot n+t_2) \in I}} \tilde{w}^{(t+1)}_{t_1,t_2,s_1,s} \cdot \left(\tilde{o}^{(t)}(\mathbf{x}) \right)_{((i-1)\cdot n+t_1,(j-1)\cdot n+t_2),s_1} + \tilde{w}^{(t+1)}_{s} \right) \\
&= \sigma \left(\sum_{s_1=1}^{k} \sum_{\substack{t_1,t_2 \in \{1,\ldots,M\} \\ ((i+t_1-2)\cdot n+1,(j+t_2-2)\cdot n+1) \in I}} w^{(t+1)}_{t_1,t_2,s_1,s} \cdot \left(\tilde{o}^{(t)}(\mathbf{x}) \right)_{((i+t_1-2)\cdot n+1,(j+t_2-2)\cdot n+1),s_1} + w^{(t+1)}_{s} \right) \\
&= \sigma \left(\sum_{s_1=1}^{k} \sum_{\substack{t_1,t_2 \in \{1,\ldots,M\} \\ (i+t_1-1,i+t_2-1) \in \tilde{I}}} w^{(t+1)}_{t_1,t_2,s_1,s} \cdot \left(o^{(t)}(\mathbf{x}) \right)_{(i+t_1-1,i+t_2-1),s_1} + w^{(t+1)}_{s} \right) \\
&= \left(o^{(t+1)}(\mathbf{x}) \right)_{(i,j),s}.
\end{aligned}$$

Die vierte Gleichheit folgt dabei aus der Induktionshypothese zusammen mit dem Fakt, dass

$$((i+t_1-2)\cdot n+1,(j+t_2-2)\cdot n+1)\in I \iff (i+t_1-1,i+t_2-1)\in \tilde{I},$$

was durch Lemma 10 a) unten impliziert wird. Die Behauptung folgt dann aufgrund der Definition der Subsampling Schicht $f_{sub}^{(n)}$. □

Lemma 10. *Es seien* $a,b,c\in\mathbb{N}$.

a) Es gilt

$$(a-1)\cdot b+1\le c \iff a\le\left\lceil\frac{c}{b}\right\rceil.$$

b) Es gilt

$$\left\lceil\frac{c}{a\cdot b}\right\rceil=\left\lceil\frac{\lceil c/a\rceil}{b}\right\rceil.$$

Beweis. a) Es gelten die Äquivalenzen

$$(a-1)\cdot b+1\le c \iff a\le\frac{c+(b-1)}{b} \overset{a\in\mathbb{N}}{\iff} a\le\left\lfloor\frac{c+(b-1)}{b}\right\rfloor$$

Es folgt

$$\begin{aligned}\left\lfloor\frac{c+(b-1)}{b}\right\rfloor &= \max\left\{k\in\mathbb{Z}: k\le\frac{c+(b-1)}{b}\right\}\\ &= \min\left\{k\in\mathbb{Z}: k+1>\frac{c+(b-1)}{b}\right\}\\ &= \min\left\{k\in\mathbb{Z}: k+1\ge\frac{c+(b-1)}{b}+\frac{1}{b}\right\}\\ &= \left\lceil\frac{c}{b}\right\rceil,\end{aligned}$$

wobei man sich die vorletzte Gleichheit durch die Fallunterscheidung „b teilt $c+(b-1)$“ und „b teilt $c+(b-1)$ nicht“ klar machen kann.

b) Wir setzen $x:=\lceil c/a\rceil-c/a$ und berechnen

$$\begin{aligned}\left\lceil\frac{c}{a\cdot b}\right\rceil &= \min\left\{k\in\mathbb{Z}: k\ge\frac{c}{a\cdot b}\right\}\\ &= \min\{k\in\mathbb{Z}: x\ge\lceil c/a\rceil-k\cdot b\}\\ &\overset{x\in[0,1)}{=} \min\{k\in\mathbb{Z}: 0\ge\lceil c/a\rceil-k\cdot b\}\\ &= \left\lceil\frac{\lceil c/a\rceil}{b}\right\rceil.\end{aligned}$$

□

Beweis von Lemma 7. Als erstes werden wir zeigen, dass $\mathcal{F}_3(\boldsymbol{\theta}_3)\subset\mathcal{F}_4(\boldsymbol{\theta}_4)$. Dafür sei $f\in\mathcal{F}_3(\boldsymbol{\theta}_3)$ definiert durch

$$f=f_{out}^{(\mathbf{A})}\circ o_{(k,k),M_L}^{(z)}\circ f_{max}^{(s_{L-1})}\circ o_{(k,k),M_{L-1}}^{(z)}\circ\cdots\circ f_{max}^{(s_1)}\circ o_{(1,k),M_1}^{(z)}$$

mit dem Gewichtsvektor $\mathbf{w} = \big((\mathbf{w}_{r,1}, \dots, \mathbf{w}_{r,z})_{r\in\{1,\dots,L\}}\big)$, wobei die Gewichtsvektoren $\mathbf{w}_{r,1}, \dots, \mathbf{w}_{r,z}$ die Gewichtsvektoren des r-ten faltenden Blocks bezeichnen. Die Idee ist es, das faltende neuronale Netz f so darzustellen, dass wir Lemma 8 am Ende jedes faltenden Blocks verwenden können, um jede lokale Max-Pooling Schicht durch einen faltenden Block und eine Subsampling Schicht ersetzen zu können. Um Lemma 8 verwenden zu können, fügen wir jeder faltenden Schicht $k+4$ Kanäle hinzu. Bereits existierende Gewichte bleiben dabei gleich und die Gewichte, die wir in den Kanälen $1, \dots, k$ hinzufügen, erhalten den Wert 0. D.h. für die Gewichtsvektoren

$$\mathbf{w'}_{r,t} = \left(\left(w'^{(t)}_{i,j,s_1,s_2} \right)_{1\le i,j\le M_r, s_1\{1,\dots,k'_{r,t}\}, s_2\in\{1,\dots,2\cdot k+4\}}, \left(w'^{(t)}_{s_2} \right)_{s_2\in\{1,\dots,2\cdot k+4\}} \right) \quad (r=1,\dots,L, t=1,\dots,z)$$

mit

$$k'_{r,t} = \begin{cases} 1, & \text{für } r=t=1, \\ 2\cdot k+4, & \text{sonst,} \end{cases}$$

der entsprechend erweiterten faltenden Blöcke

$$o'^{(z)}_{(1,2\cdot k+4),M_1}, o'^{(z)}_{(2\cdot k+4,2\cdot k+4),M_2}, \dots, o'^{(z)}_{(2\cdot k+4,2\cdot k+4),M_L}, \tag{3.41}$$

gilt

$$w'^{(t)}_{s_2} = w^{(t)}_{s_2} \qquad \text{und} \qquad w'^{(r,t)}_{t_1,t_2,s_1,s_2} = \begin{cases} 0, & \text{falls } s_1 > k, \\ w^{(r,t)}_{t_1,t_2,s_1,s_2}, & \text{sonst} \end{cases}$$

für $s_2 \in \{1,\dots,k\}$, $r \in \{1,\dots,L\}$, $t_1, t_2 \in \{1\dots,M_r\}$, $t \in \{1,\dots,z\}$ und $s_1 \in \{1,\dots,k'_{r,t}\}$. Die faltenden Blöcke (3.41) haben damit die folgende Eigenschaft:

$$\begin{aligned} &\big(o'^{(z)}_{(2\cdot k+4,2\cdot k+4),M_r} \circ \dots \circ f^{(s_2)}_{max} \circ o'^{(z)}_{(2\cdot k+4,2\cdot k+4),M_2} \circ f^{(s_1)}_{max} \circ o'^{(z)}_{(1,2\cdot k+4),M_1}(\mathbf{x})\big)_{(i,j),s} \\ &= \big(o^{(z)}_{(k,k),M_r} \circ \dots \circ f^{(s_2)}_{max} \circ o^{(z)}_{(k,k),M_2} \circ f^{(s_1)}_{max} \circ o^{(z)}_{(1,k),M_1}(\mathbf{x})\big)_{(i,j),s} \end{aligned} \tag{3.42}$$

für alle $s \in \{1,\dots,k\}$, $r \in \{1,\dots,L\}$, $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$ und $(i,j) \in I_r$, wobei die Indexmengen $I_r \subset \mathbb{N}^2$ $(r=1,\dots,L)$ durch die Indexmengen der Wertebereiche der faltenden Blöcke des Netzwerkes f definiert sind. Wegen Ungleichung (3.37) gilt

$$M_r = \frac{2^r}{\prod_{i=0}^{r} s_i} \cdot \frac{s_r}{2} + 1 \ge \frac{s_r}{2} + 1 \quad (r=1,\dots,L-1).$$

Deshalb können wir Lemma 8 anwenden, was faltende Blöcke

$$\tilde{o}^{(3\cdot k\cdot\log_2(s_1))}_{(2\cdot k+4,2\cdot k+4),M_1}, \dots, \tilde{o}^{(3\cdot k\cdot\log_2(s_{L-1}))}_{(2\cdot k+4,2\cdot k+4),M_{L-1}}$$

liefert, sodass zusammen mit Gleichung (3.42) folgt

$$\begin{aligned} &\big(o'^{(z)}_{(2\cdot k+4,2\cdot k+4),M_L} \circ f^{(s_{L-1})}_{sub} \circ \tilde{o}^{(3\cdot k\cdot\log_2(s_{L-1}))}_{(2\cdot k+4,2\cdot k+4),M_{L-1}} \circ o'^{(z)}_{(2\cdot k+4,2\cdot k+4),M_{L-1}} \circ \dots \\ &\qquad \circ f^{(s_1)}_{sub} \circ \tilde{o}^{(3\cdot k\cdot\log_2(s_1))}_{(2\cdot k+4,2\cdot k+4),M_1} \circ o'^{(z)}_{(1,2\cdot k+4),M_1}(\mathbf{x})\big)_{(i,j),s} \\ &= \big(o^{(z)}_{(k,k),M_L} \circ f^{(s_{L-1})}_{max} \circ o^{(z)}_{(k,k),M_{L-1}} \circ \dots \circ f^{(s_1)}_{max} \circ o^{(z)}_{(1,k),M_1}(\mathbf{x})\big)_{(i,j),s} \end{aligned}$$

für alle $s \in \{1,\dots,k\}$, $\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$ und $(i,j) \in I_L$. Mit der Idee, bereits berechnete Werte durch Gleichung 3.31 in die nächste Schicht weiterzugeben, können wir nun entsprechend viele faltende Schichten zu jedem Block hinzufügen, sodass faltende Blöcke

$$\bar{o}^{(\bar{z})}_{(1,\bar{k}),M_1}, \dots, \bar{o}^{(\bar{z})}_{(\bar{k},\bar{k}),M_L}$$

existieren, sodass

$$\begin{aligned}&\left(\bar{o}^{(\bar{z})}_{(\bar{k},\bar{k}),M_L} \circ f^{(s_{L-1})}_{sub} \circ \bar{o}^{(\bar{z})}_{(\bar{k},\bar{k}),M_{L-1}} \circ \cdots \circ f^{(s_1)}_{sub} \circ \bar{o}^{(\bar{z})}_{(1,\bar{k}),M_1}(\mathbf{x})\right)_{(i,j),s} \\ &= \left(o^{(z)}_{(k,k),M_L} \circ f^{(s_{L-1})}_{max} \circ o^{(z)}_{(k,k),M_{L-1}} \circ \cdots \circ f^{(s_1)}_{max} \circ o^{(z)}_{(1,k),M_1}(\mathbf{x})\right)_{(i,j),s}\end{aligned} \tag{3.43}$$

für alle $s \in \{1,\ldots,k\}$, $(i,j) \in I_L$ und $\mathbf{x} \in [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}}$. Letztlich definieren wir die Ausgabeschicht $\tilde{f}^{(\mathbf{A})}_{out} : \mathbb{R}^{I_L\times\{1,\ldots,2\cdot k+4\}} \to \mathbb{R}$, indem wir die Gewichte

$$\tilde{\mathbf{w}}_{out} = (\tilde{w}_s)_{s\in\{1,\ldots,2\cdot k+4\}}$$

von $\tilde{f}^{(\mathbf{A})}_{out}$ durch die Gewichte $\mathbf{w}_{out} = (w_s)_{s\in\{1,\ldots,k_L\}}$ von $f^{(\mathbf{A})}_{out}$ wie folgt definieren:

$$\tilde{w}_s = \begin{cases} w_s, & \text{falls } s \in \{1,\ldots,k\} \\ 0, & \text{sonst.}\end{cases}$$

Wir schließen aus Gleichung (3.43)

$$f = \tilde{f}^{(\mathbf{A})}_{out} \circ \bar{o}^{(\bar{z})}_{(\bar{k},\bar{k}),M_L} \circ f^{(s_{L-1})}_{sub} \circ \bar{o}^{(\bar{z})}_{(\bar{k},\bar{k}),M_{L-1}} \circ \cdots \circ f^{(s_1)}_{sub} \circ \bar{o}^{(\bar{z})}_{(1,\bar{k}),M_1} \in \mathcal{F}_4(\boldsymbol{\theta}_4).$$

Es bleibt zu zeigen, dass $\mathcal{F}_4(\boldsymbol{\theta}_4) \subset \mathcal{F}_5(\boldsymbol{\theta}_5)$. Sei daher $f \in \mathcal{F}_4(\boldsymbol{\theta}_4)$ gegeben durch

$$f = f^{(\mathbf{A})}_{out} \circ o^{(\bar{z})}_{(\bar{k},\bar{k}),M_L} \circ f^{(s_{L-1})}_{sub} \circ o^{(\bar{z})}_{(\bar{k},\bar{k}),M_{L-1}} \circ \cdots \circ o^{(\bar{z})}_{(\bar{k},\bar{k}),M_2} \circ f^{(s_1)}_{sub} \circ o^{(\bar{z})}_{(1,\bar{k}),M_1}.$$

Hier ist die Idee, solange ein Ausdruck der Form $o^{(\bar{z})}_{(\bar{k},\bar{k}),M} \circ f^{(s)}_{sub}$ in der Darstellung der Funktion f existiert, diesen durch einen Ausdruck der Form

$$f^{(s)}_{sub} \circ \tilde{o}^{(\bar{z})}_{(\bar{k},\bar{k}),(M-1)\cdot s+1}$$

zu ersetzen, indem Lemma 9 angewendet wird. Wenn wir mit dem Ausdruck $o^{(\bar{z})}_{(\bar{k},\bar{k}),M_L} \circ f^{(s_{L-1})}_{sub}$ starten und jede Subsampling Schicht auf diese Weise bis ans Ende aller faltenden Blöcke verschieben, in absteigender Reihenfolge $L-1, L-2, \ldots, 1$, müssen wir Lemma 9

$$\underbrace{1}_{\text{für } f^{(s_{L-1})}_{sub}} + \underbrace{2}_{\text{für } f^{(s_{L-2})}_{sub}} + \cdots + \underbrace{L-1}_{\text{für } f^{(s_1)}_{sub}} = \frac{L\cdot(L-1)}{2}$$

mal anwenden, um faltende Blöcke $\tilde{o}^{(\bar{z})}_{(\bar{k},\bar{k}),M_2^{(1)}}, \ldots, \tilde{o}^{(\bar{z})}_{(\bar{k},\bar{k}),M_L^{(L-1)}}$ zu erhalten, sodass

$$\begin{aligned}f &= f^{(\mathbf{A})}_{out} \circ o^{(\bar{z})}_{(\bar{k},\bar{k}),M_L} \circ f^{(s_{L-1})}_{sub} \circ o^{(\bar{z})}_{(\bar{k},\bar{k}),M_{L-1}} \circ \cdots \circ f^{(s_1)}_{sub} \circ o^{(\bar{z})}_{(1,\bar{k}),M_1} \\ &= f^{(\mathbf{A})}_{out} \circ f^{(s_{L-1})}_{sub} \circ \cdots \circ f^{(s_1)}_{sub} \circ \tilde{o}^{(\bar{z})}_{(\bar{k},\bar{k}),M_L^{(L-1)}} \circ \cdots \circ \tilde{o}^{(\bar{z})}_{(\bar{k},\bar{k}),M_2^{(1)}} \circ o^{(\bar{z})}_{(1,\bar{k}),M_1},\end{aligned}$$

wobei die Filtergrößen $M_r^{(r-1)}$ rekursiv definiert sind durch

$$M_r^{(t)} = \left(M_r^{(t-1)} - 1\right)\cdot s_{r-t} + 1$$

für $r \in \{2,\ldots,L\}$, $M_r^{(0)} = M_r$ und $t \in \{1,\ldots,r-1\}$. Durch Induktion über t ist es leicht zu sehen, dass

$$M_r^{(t)} = \frac{2^{r-1}}{\prod_{i=0}^{r-1-t} s_i} + 1$$

für $t \in \{0, \dots, r-1\}$ gilt, was $M_r^{(r-1)} = 2^{r-1} + 1 = \bar{M}_{(r-1)\cdot\bar{z}+1}, \dots, \bar{M}_{r\cdot\bar{z}}$ für $r \in \{2, \dots, L\}$ impliziert. Es bleibt zu zeigen, dass

$$f_{sub}^{(s_1 \cdot s_2 \cdot \dots \cdot s_{L-1})} = f_{sub}^{(s_{L-1})} \circ f_{sub}^{(s_{L-2})} \circ \dots \circ f_{sub}^{(s_1)},$$

was aus Lemma 10 b) folgt. □

3.4.3. Approximation des rotationssymmetrischen hierarchischen Max-Pooling Modells

In diesem Abschnitt beweisen wir das folgende Resultat zur Approximation des rotationssymmetrischen hierarchischen Max-Pooling Modells durch faltende neuronale Netze, der in Abschnitt 3.1 eingeführten Netzwerkarchitektur $\mathcal{F}_2(\boldsymbol{\theta})$.

Lemma 11. *Es seien $n, l, \lambda \in \mathbb{N}$ mit $(2^l + 2l - 1) \leq \lambda$ und es sei $0 < h \leq 2^l/(\sqrt{2} \cdot \lambda)$. Setze $b = (2^l + 2l - 1)/(2\lambda)$ und sei $p \in [1, \infty)$. Außerdem sei $\eta : [0,1]^{C_1} \to [0,1]$ eine Funktion, die einem rotationssymmetrischen hierarchischen Max-Pooling Modell vom Level l, der Breite h, mit Randabstand b und mit Glattheitsbedingung p genügt. Weiterhin nehme an, dass Annahme 1 (siehe Abschnitt 2.3) für (p, C)-glatte Funktionen $g_{0,s} : \mathbb{R} \to [0,1]$ $(s = 1, \dots, 4^l)$ und Annahme 2 (siehe Abschnitt 2.3) für ein $\epsilon_\lambda \in [0,1]$, ein messbares $A \subset [0,1]^{C_1}$ und einen Skalierungsfaktor $c > 1$ erfüllt ist. Wähle die Netzwerkparameter L_n und $\boldsymbol{\theta} = (t, L_{net}, r_{net}, L, k, \mathbf{M}, \mathbf{A})$ wie in Theorem 3.3. Dann existiert ein faltendes neuronales Netz $f_{CNN} \in \mathcal{F}_2(\boldsymbol{\theta})$, sodass*

$$|f_{CNN}(g_\lambda(\phi)) - \eta(\phi)|^2 \leq c_{25} \cdot \left(n^{-\frac{2 \cdot p}{2 \cdot p + 4}} + \epsilon_\lambda^2 \right)$$

für alle $\phi \in A$ und eine Konstante $c_{25} > 0$ gilt, die nicht von λ und n abhängt.

Bevor wir Lemma 11 beweisen, beschreiben wir das grundlegende Beweisvorgehen, welches sich in drei Schritte gliedern lässt. Noch vor dem ersten Schritt werden wir zunächst ein neues Modell einführen, nämlich das diskretisierte hierarchische Max-Pooling Modell (siehe Definition 7 unten).

1. In dem *ersten Schritt* (siehe Lemma 12) zeigen wir, dass wir das rotationssymmetrische hierarchische Max-Pooling Modell durch das (neue) diskretisierte hierarchische Max-Pooling Modell approximieren können, wenn die Funktionen $\bar{g}_{k,s}^{(i)}$ des diskretisierten Modells den Funktionen $g_{k,s}$ des rotationssymmetrischen hierarchischen Max-Pooling Modells entsprechen.

2. Im *zweiten Schritt* (siehe Lemma 13) wird gezeigt, wie der Fehler, der auftritt, wenn die Funktionen $g_{k,s}^{(i)}$ im diskretisierten hierarchischen Max-Pooling Modell durch Approximationen $\bar{g}_{k,s}^{(i)}$ ersetzt werden, beschränkt werden kann.

3. Im *dritten Schritt* (siehe Lemma 14) werden wir zeigen, dass ein diskretisiertes hierarchisches Max-Pooling Modell durch ein faltendes neuronales Netz aus der obigen Klasse $\mathcal{F}_2(\boldsymbol{\theta})$ dargestellt werden kann, wenn die Funktionen $\bar{g}_{k,s}^{(i)}$ vollverbundenen neuronalen Netzen entsprechen.

Da die Funktionen $g_{k,s}$ des rotationssymmetrischen hierarchischen Max-Pooling Modells (p, C)-glatt sind, können wir dann das Approximationsresultat aus Lemma 2 für vollverbundene neuronale Netze verwenden, um Lemma 11 durch eine Kombination der drei Schritte zu beweisen.

Das neue diskretisierte hierarchische Max-Pooling Modell ist ähnlich wie das hierarchische Max-Pooling Modell aus Definition 3, mit dem Hauptunterschied, dass die Positionen der Teilbereiche, die hierarchisch kombiniert werden, nicht vorgegeben sind. Wir verwenden in diesem Abschnitt die folgende Notation:

$$I^{(k)} = \left\{ -(2^{k-1} + k - 1), \dots, -1, 0, 1, \dots, 2^{k-1} + k - 1 \right\}^2 \subset \mathbb{Z}^2 \quad (k \in \mathbb{N})$$

und $I^{(0)} = \{0\} \times \{0\}$.

Definition 7. Es seien $\lambda, l, d \in \mathbb{N}$ mit $2^l + 2 \cdot l - 1 \leq \lambda$.

a) Wir sagen $\bar{\eta} : [0,1]^{\{1,\dots,\lambda\}^2} \to \mathbb{R}$ genügt einem *diskretisierten Max-Pooling Modell der Ordnung d*, falls Funktionen $\bar{f}^{(i)} : [0,1]^{I^{(l)}} \to \mathbb{R}$ $(i = 1, \dots, d)$ existieren, sodass

$$\bar{\eta}(\mathbf{x}) = \max_{\mathbf{u} \in \{1,\dots,\lambda\}^2 \,:\, \mathbf{u} + I^{(l)} \subseteq \{1,\dots,\lambda\}^2} \max_{i \in \{1,\dots,d\}} \bar{f}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}).$$

b) Wir sagen $\bar{f} : [0,1]^{I^{(l)}} \to \mathbb{R}$ genügt einem *diskretisierten hierarchischen Modell vom Level l mit Funktionen* $\{\bar{g}_{k,s}\}_{k \in \{0,\dots,l\}, s \in \{1,\dots,4^{l-k}\}}$, wobei

$$\bar{g}_{k,s} : \mathbb{R}^4 \to \mathbb{R}_0^+ \quad \left(k = 1, \dots, l, s = 1, \dots, 4^{l-k}\right)$$

und

$$\bar{g}_{0,s} : [0,1] \to \mathbb{R}_0^+ \quad \left(s = 1, \dots, 4^l\right),$$

falls Gitterpunkte

$$\mathbf{i}_{k,s} \in \left\{-\lfloor 2^{k-1} \rfloor - 1, \dots, 0, \dots, \lfloor 2^{k-1} \rfloor + 1\right\}^2 \quad \left(k = 0, \dots, l-1, s = 1, \dots, 4^{l-k}\right)$$

existieren, sodass

$$\bar{f} = \bar{f}_{l,1}$$

für Funktionen $\bar{f}_{k,s} : [0,1]^{I^{(k)}} \to \mathbb{R}$ $(k = 0, \dots, l, s = 1, \dots, 4^{l-k})$, die wie folgt rekursiv definiert sind:

$$\bar{f}_{k,s}(\mathbf{x}) = \bar{g}_{k,s}\Big(\bar{f}_{k-1,4\cdot(s-1)+1}(\mathbf{x}_{\mathbf{i}_{k-1,4\cdot(s-1)+1} + I^{(k-1)}}), \bar{f}_{k-1,4\cdot(s-1)+2}(\mathbf{x}_{\mathbf{i}_{k-1,4\cdot(s-1)+2} + I^{(k-1)}}),$$
$$\bar{f}_{k-1,4\cdot(s-1)+3}(\mathbf{x}_{\mathbf{i}_{k-1,4\cdot(s-1)+3} + I^{(k-1)}}), \bar{f}_{k-1,4\cdot s}(\mathbf{x}_{\mathbf{i}_{k-1,4\cdot s} + I^{(k-1)}})\Big)$$

für $k = 1, \dots, l$ und $s = 1, \dots, 4^{l-k}$ sowie

$$\bar{f}_{0,s}(x) = \bar{g}_{0,s}(x)$$

für $s = 1, \dots, 4^l$.

c) Wir sagen $\bar{\eta} : [0,1]^{\{1,\dots,\lambda\}^2} \to \mathbb{R}$ genügt einem *diskretisierten hierarchischen Max-Pooling Modell vom Level l der Ordnung d mit Funktionen* $\{\bar{g}_{k,s}^{(i)}\}_{i \in \{1,\dots,d\}, k \in \{0,\dots,l\}, s \in \{1,\dots,4^{l-k}\}}$, falls $\bar{\eta}$ einem diskretisierten Max-Pooling Modell der Ordnung d genügt und die Funktionen $\bar{f}^{(i)} : [0,1]^{I^{(l)}} \to \mathbb{R}$ in der Definition des diskretisierten Max-Pooling Modells einem diskretisierten hierarchischen Modell vom Level l mit den Funktionen $\{\bar{g}_{k,s}^{(i)}\}_{k \in \{0,\dots,l\}, s \in \{1,\dots,4^{l-k}\}}$ $(i = 1, \dots, d)$ genügen.

Bemerkung 3.8. Damit die Funktionen $\bar{f}_{k,s}$ in Definition 7 b) wohldefiniert sind, benötigen wir

$$\mathbf{i}_{k,s} + I^{(k)} \subseteq I^{(k+1)}$$

für alle $k \in \{0, \dots, l-1\}$, $s \in \{1, \dots, 4^{l-k}\}$ und $\mathbf{i}_{k,s} \in \left\{-\lfloor 2^{k-1} \rfloor - 1, \dots, 0, \dots, \lfloor 2^{k-1} \rfloor + 1\right\}^2$, was aus

$$\max_{\substack{\mathbf{u} \in \mathbf{i}_{k,s} + I^{(k)} \\ \mathbf{i}_{k,s} \in \left\{-\lfloor 2^{k-1} \rfloor - 1, \dots, 0, \dots, \lfloor 2^{k-1} \rfloor + 1\right\}^2}} \|\mathbf{u}\|_\infty \leq \max_{\mathbf{i}_{k,s} \in \left\{-\lfloor 2^{k-1} \rfloor - 1, \dots, 0, \dots, \lfloor 2^{k-1} \rfloor + 1\right\}^2} \|\mathbf{i}_{k,s}\|_\infty + \max_{\mathbf{u} \in I^{(k)}} \|\mathbf{u}\|_\infty$$

$$
\begin{aligned}
&= \left(\lfloor 2^{k-1} \rfloor + 1\right) + \left(\lceil 2^{k-1} \rceil + k - 1\right) \\
&= 2^{(k+1)-1} + (k+1) - 1 \\
&= \max_{\mathbf{u} \in I^{(k+1)}} \|\mathbf{u}\|_\infty
\end{aligned}
$$

folgt.

Wie oben beschrieben, zeigen wir im folgenden Lemma zunächst, dass wir das rotationssymmetrische hierarchische Max-Pooling Modell durch das diskretisierte hierarchische Max-Pooling Modell approximieren können.

Lemma 12. *Es seien $\lambda, l \in \mathbb{N}$ mit $2^l + 2l - 1 \leq \lambda$. Setze $b = (2^l + 2l - 1)/(2\lambda)$. Außerdem sei $0 < h \leq 2^l/(\sqrt{2} \cdot \lambda)$ und $h_k = h/2^{l-k}$ für $k \in \mathbb{Z}$. Weiterhin sei $\eta : [0,1]^{C_1} \to \mathbb{R}$ eine Funktion, die einem rotationssymmetrischen hierarchischen Max-Pooling Modell vom Level l, der Breite h und dem Randabstand b genügt, welches durch die Funktionen*

$$g_{k,s} : \mathbb{R}^4 \to [0,1] \quad (k = 1, \dots, l, s = 1, \dots, 4^{l-k})$$

und

$$f_{0,s} : [0,1]^{C_{h_0}} \to [0,1] \quad (s = 1, \dots, 4^l)$$

gegeben ist. Die Funktionen $f_{k,s} : [0,1]^{C_{h_k}} \to [0,1]$ $(k = 1, \dots, l, s = 1, \dots, 4^{l-k})$ seien wie in Definition 5 b) definiert. Außerdem nehmen wir an, dass die Einschränkungen

$$g_{k,s}\big|_{[0,1]^4} : [0,1]^4 \to [0,1] \quad (k = 1, \dots, l, s = 1, \dots, 4^{l-k})$$

lipschitzstetig mit Lipschitzkonstante $L > 0$ sind (bezüglich der durch die Maximumsnorm induzierte Metrik) und dass Annahme 2 (siehe Abschnitt 2.3) für ein $\epsilon_\lambda \in [0,1]$, ein messbares $A \subset [0,1]^{C_1}$ und ein Skalierungsfaktor $c > 1$ erfüllt ist. Dann existiert ein diskretisiertes hierarchisches Max-Pooling Modell $\bar{\eta} : [0,1]^{\{1,\dots,\lambda\}^2} \to \mathbb{R}$ vom Level l, der Ordnung

$$d = \left\lceil \frac{2^{l-1/2} \cdot \pi}{c - 1} \right\rceil \tag{3.44}$$

mit Funktionen $\{\bar{g}_{k,s}^{(i)}\}$, welche durch

$$\bar{g}_{k,s}^{(i)} = g_{k,s} \quad \big(i = 1, \dots, d, k = 0, \dots, l, s = 1, \dots, 4^{l-k}\big)$$

mit $g_{0,s}(x) = f_{0,s}(x \cdot 1\big|_{C_{h_0}})$ $(x \in [0,1])$ für $s = 1, \dots, 4^l$ gegeben sind, sodass

$$|\bar{\eta}(g_\lambda(\phi)) - \eta(\phi)| \leq L^l \cdot \epsilon_\lambda \quad \big(\phi \in A\big).$$

Beweis. Im Beweis verwenden wir, wie im Beweis von Lemma 3 (siehe Gleichung (3.26)), dass

$$\left| \max_{i=1,\dots,n} a_i - \max_{i=1,\dots,n} b_i \right| \leq \max_{i=1,\dots,n} |a_i - b_i| \tag{3.45}$$

für beliebige $n \in \mathbb{N}$, $a_1, \dots, a_n, b_1, \dots, b_n \in \mathbb{R}$ gilt. Außerdem benötigen wir im Beweis die Bijektion

$$\varphi : \{1, \dots, \lambda\}^2 \to G_\lambda,$$

die definiert ist durch

$$\varphi((i,j)) = \left(\frac{i - \frac{1}{2}}{\lambda} - \frac{1}{2}, \frac{j - \frac{1}{2}}{\lambda} - \frac{1}{2} \right) \quad \big((i,j) \in \{1, \dots, \lambda\}^2\big) \tag{3.46}$$

und die Umkehrabbildung $\varphi^{-1} : G_\lambda \to \{1,\dots,\lambda\}^2$ besitzt, die durch

$$\varphi^{-1}((i,j)) = \left(\lambda \cdot \left(i + \frac{1}{2}\right) + \frac{1}{2}, \lambda \cdot \left(j + \frac{1}{2}\right) + \frac{1}{2}\right) \quad \big((i,j) \in G_\lambda\big) \tag{3.47}$$

definiert ist, wobei das Gitter G_λ das in Gleichung (1.26) definierte Gitter bezeichnet. Bevor wir das vollständige diskretisierte hierarchische Max-Pooling Modell $\bar{\eta}$ definieren, d.h., bevor wir die Gitterpunkte $\mathbf{i}_{k,s}^{(i)}$ $(i = 1,\dots,d)$ des Modells definieren, werden wir $|\bar{\eta}(g_\lambda(\phi)) - \eta(\phi)|$ von oben abschätzen, indem wir Gleichung (3.45) verwenden. Hierfür definieren wir das Gitter

$$\begin{aligned} G &= \{\mathbf{u} \in \{1,\dots,\lambda\}^2 \,:\, \mathbf{u} + I^{(l)} \subseteq \{1,\dots,\lambda\}^2\} \\ &= \{2^{l-1} + l, \dots, \lambda - 2^{l-1} - l + 1\}^2 \end{aligned} \tag{3.48}$$

sowie für $(i,j) \in G$ den Quader

$$\begin{aligned} P_{(i,j)} &= \left(\varphi((i,j)) + \left[-\frac{1}{2\lambda}, \frac{1}{2\lambda}\right]^2\right) \cap \left[-\frac{1}{2} + b, \frac{1}{2} - b\right]^2 \\ &= \left(\left(\frac{i - \frac{1}{2}}{\lambda} - \frac{1}{2}, \frac{j - \frac{1}{2}}{\lambda} - \frac{1}{2}\right) + \left[-\frac{1}{2\lambda}, \frac{1}{2\lambda}\right]^2\right) \cap \left[-\frac{1}{2} + b, \frac{1}{2} - b\right]^2 \end{aligned}$$

und erhalten wegen der Definition von $I^{(l)}$ und b, dass

$$\begin{aligned} \bigcup_{\mathbf{u} \in G} P_{\mathbf{u}} &= \bigcup_{(i,j) \in G} \left(\left(\frac{i - \frac{1}{2}}{\lambda} - \frac{1}{2}, \frac{j - \frac{1}{2}}{\lambda} - \frac{1}{2}\right) + \left[-\frac{1}{2\lambda}, \frac{1}{2\lambda}\right]^2\right) \cap \left[-\frac{1}{2} + b, \frac{1}{2} - b\right]^2 \\ &= \bigcup \Bigg\{ \left[-\frac{1}{2} - \frac{i-1}{\lambda}, -\frac{1}{2} + \frac{i}{\lambda}\right] \times \left[-\frac{1}{2} - \frac{j-1}{\lambda}, -\frac{1}{2} + \frac{j}{\lambda}\right] \\ &\qquad\qquad : (i,j) \in \{2^{l-1} + l, \dots, \lambda - 2^{l-1} - l + 1\}^2 \Bigg\} \\ &\quad \cap \left[-\frac{1}{2} + \frac{2^{l-1} + l - \frac{1}{2}}{\lambda}, \frac{1}{2} - \frac{2^{l-1} + l - \frac{1}{2}}{\lambda}\right]^2 \\ &= \left[-\frac{1}{2} + \frac{2^{l-1} + l - \frac{1}{2}}{\lambda}, \frac{1}{2} - \frac{2^{l-1} + l - \frac{1}{2}}{\lambda}\right]^2 \\ &= \left[-\frac{1}{2} + b, \frac{1}{2} - b\right]^2. \end{aligned} \tag{3.49}$$

Außerdem ermöglicht uns Definition (3.44), das Intervall $[0, 2\pi]$ durch Intervalle $\{\Theta_i\}_{i=1,\dots,d}$ mit der Seitenlänge $(c-1)/(2^{l-3/2})$ und Mittelpunkten $\{\alpha_i\}_{i=1,\dots,d}$ zu überdecken. Im Folgenden bezeichnen die Funktionen $\bar{f}_{k,s} : [0,1]^{I^{(k)}} \to \mathbb{R}$ $(k = 0,\dots,l, s = 1,\dots,4^{l-k})$, die durch das diskretisierte hierarchische Max-Pooling-Modell von $\bar{\eta}$ gemäß Definition 7 gegebenen Funktionen. Für $\phi \in A$ und $\mathbf{x} := g_\lambda(\phi)$ implizieren Ungleichung (3.27) und Gleichung (3.49)

$$\begin{aligned} &|\bar{\eta}(\mathbf{x}) - \eta(\phi)| \\ &= \left| \max_{\mathbf{u} \in \{1,\dots,\lambda\}^2 \,:\, \mathbf{u} + I^{(l)} \subseteq \{1,\dots,\lambda\}^2} \max_{i \in \{1,\dots,d\}} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) - \sup_{\mathbf{v} \in \left[-\frac{1}{2}+b, \frac{1}{2}-b\right]^2} \sup_{\alpha \in [0,2\pi]} f_{l,1}(\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}|_{C_h}) \right| \end{aligned}$$

$$
\begin{aligned}
&= \left| \max_{\mathbf{u}\in G} \max_{i\in\{1,\dots,d\}} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) - \max_{\mathbf{u}\in G} \sup_{\mathbf{v}\in P_{\mathbf{u}}} \max_{i\in\{1,\dots,d\}} \sup_{\alpha\in\Theta_i} f_{l,1}(\phi\circ\tau_{\mathbf{v}}\circ rot^{(\alpha)}\big|_{C_h}) \right| \\
&\leq \max_{\mathbf{u}\in G} \left| \max_{i\in\{1,\dots,d\}} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) - \sup_{\mathbf{v}\in P_{\mathbf{u}}} \max_{i\in\{1,\dots,d\}} \sup_{\alpha\in\Theta_i} f_{l,1}(\phi\circ\tau_{\mathbf{v}}\circ rot^{(\alpha)}\big|_{C_h}) \right| \\
&\leq \max_{\mathbf{u}\in G} \sup_{\mathbf{v}\in P_{\mathbf{u}}} \max_{i\in\{1,\dots,d\}} \sup_{\alpha\in\Theta_i} \left| \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) - f_{l,1}(\phi\circ\tau_{\mathbf{v}}\circ rot^{(\alpha)}\big|_{C_h}) \right|.
\end{aligned}
$$

Es genügt nun, für alle $i \in \{1,\dots,d\}$ zu zeigen, dass Gitterpunkte $\mathbf{i}_{k,s}^{(i)}$ $(k = 0,\dots,l-1,\ s = 1,\dots,4^{l-k})$ des diskretisierten hierarchischen Modells $\bar{f}_{l,1}^{(i)}$ existieren, sodass

$$
\left| \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) - f_{l,1}(\phi\circ\tau_{\mathbf{v}}\circ rot^{(\alpha)}\big|_{C_h}) \right| \leq L^l \cdot \epsilon_\lambda \tag{3.50}
$$

für alle $\mathbf{u} \in G$, $\mathbf{v} \in P_{\mathbf{u}}$, $i \in \{1,\dots,d\}$ und $\alpha \in \Theta_i$.

Um dies zu zeigen, seien im Weiteren $\mathbf{u} \in G$, $\mathbf{v} \in P_{\mathbf{u}}$, $i \in \{1,\dots,d\}$ und $\alpha \in \Theta_i$ fest. Die Idee ist es, Gitterpunkte $\mathbf{i}_{k,s}^{(i)}$ zu konstruieren, welche nicht von $\mathbf{u}$, $\mathbf{v}$ und α abhängen, sodass wir in der Lage sind, Gleichung (3.50) zu zeigen, indem wir durch Induktion über k die folgende Ungleichung verifizieren:

$$
\left| \bar{f}_{k,s}^{(i)}(\mathbf{x}_{\varphi^{-1}(\mathbf{u}_{k,s})+I^{(k)}}) - f_{k,s}(\phi\circ\tau_{\mathbf{v}_{k,s}}\circ rot^{(\alpha)}\big|_{C_{h_k}}) \right| \leq L^k \cdot \epsilon_\lambda \tag{3.51}
$$

für alle $k = 0,\dots,l$ und $s = 1,\dots,4^{l-k}$, wobei $\mathbf{u}_{l,1} = \varphi(\mathbf{u})$ und $\mathbf{v}_{l,1} = \mathbf{v}$ sowie

$$
\mathbf{u}_{k-1,4\cdot(s-1)+j} = \mathbf{u}_{k,s} + \frac{1}{\lambda}\cdot \mathbf{i}_{k-1,4\cdot(s-1)+j}^{(i)} \text{ und } \mathbf{v}_{k-1,4\cdot(s-1)+j} = \mathbf{v}_{k,s} + rot^{(\alpha)}\left(\mathbf{h}_{k-2}^{(j)}\right) \tag{3.52}
$$

für $k = 1,\dots,l$, $s = 1,\dots,4^{l-k}$ und $j = 1,\dots,4$ mit

$$
\begin{aligned}
\mathbf{h}_{k-2}^{(1)} &= (-h_{k-2}, -h_{k-2}), \quad \mathbf{h}_{k-2}^{(2)} = (h_{k-2}, -h_{k-2}), \\
\mathbf{h}_{k-2}^{(3)} &= (-h_{k-2}, h_{k-2}), \quad \mathbf{h}_{k-2}^{(4)} = (h_{k-2}, h_{k-2}).
\end{aligned}
$$

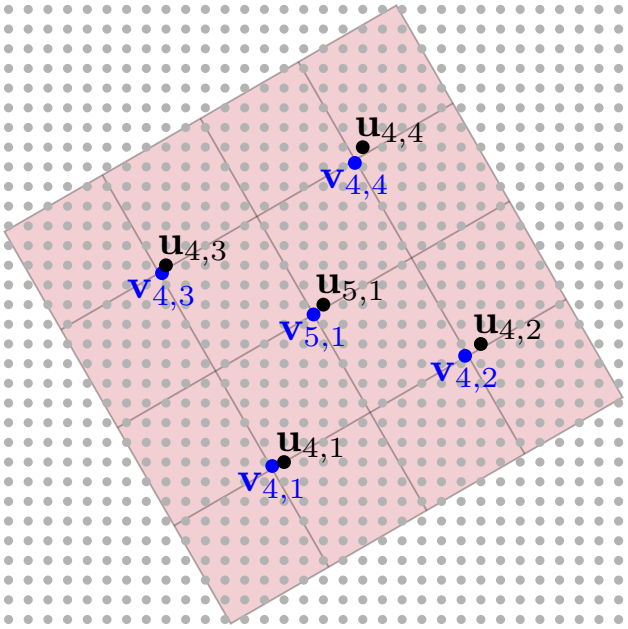
Der restliche Beweis kann in vier Schritte gegliedert werden. *Im ersten Schritt* definieren wir die Gitterpunkte $\mathbf{i}_{k,s}^{(i)}$ und zeigen, dass diese gemäß Defintion 7 b) wohldefiniert sind. *Im zweiten Schritt* zeigen wir, dass $\mathbf{u}_{k,s}$ „nah“ an $\mathbf{v}_{k,s}$ liegt (siehe Abbildung 3.5a für ein Beispiel). *Im dritten Schritt* verwenden wir Annahme 2 und zeigen, dass Ungleichung (3.51) für $k = 0$ gilt, um *im vierten Schritt* den entsprechenden Induktionsschritt für den Beweis von Gleichung (3.51) auszuführen (siehe Abbildung 3.5b für eine Darstellung der in die Funktionen $\bar{f}_{k,s}^{(i)}$ und $f_{k,s}$ gemäß Ungleichung (3.51) eingesetzten diskreten bzw. kontinuierlichen Teilbereiche).

Schritt 1: Als erstes betrachteten wir einen um den Ursprung mit dem Winkel α_i rotierten Teilbereich der Breite h, wobei α_i den Mittelpunkt des Intervalls Θ_i bezeichnet. Analog zu der Definition von $\mathbf{v}_{k,s}$ unterteilen wir den Teilbereich in immer kleinere Teilbereiche und wählen die Punkte $\mathbf{z}_{z,k}^{(i)}$ als Mittelpunkte dieser Teilbereiche. Die Idee ist, dass $\mathbf{z}_{k,s}^{(i)}$ dann „nah“ an $\mathbf{v}_{k,s} - \mathbf{v}$ liegt, wie wir im *zweiten Schritt* sehen werden. Wir setzen $\mathbf{z}_{l,1}^{(i)} = (0,0)$ und definieren rekursiv

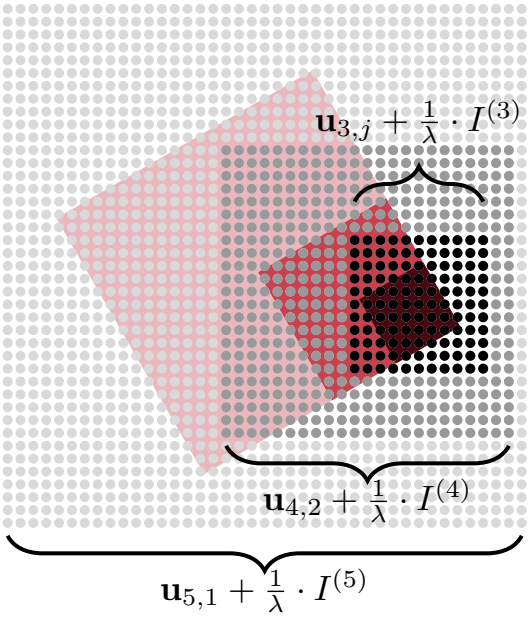
$$
\mathbf{z}_{k-1,4\cdot(s-1)+j}^{(i)} = \mathbf{z}_{k,s}^{(i)} + rot^{(\alpha_i)}\left(\mathbf{h}_{k-2}^{(j)}\right)
$$

für $k = 1,\dots,l$, $s = 1,\dots,4^{l-k}$ und $j = 1,\dots,4$. Da $\mathbf{i}_{k,s}^{(i)}$ ganzzahlige Gitterpunkte sein sollen, wählen wir

$$
\bar{\mathbf{z}}_{k,s}^{(i)} \in \underset{\mathbf{z}\in\left\{-\frac{2^{l-1}+l-1}{\lambda},\dots,\frac{-1}{\lambda},0,\frac{1}{\lambda},\dots,\frac{2^{l-1}+l-1}{\lambda}\right\}^2}{\arg\min} \|\mathbf{z} - \mathbf{z}_{k,s}^{(i)}\|_\infty, \quad \left(k = 0,\dots,l, s = 1,\dots,4^{l-k}\right) \tag{3.53}
$$



(a) Darstellung der Punkte $\mathbf{v}_{k,s}$ und $\mathbf{u}_{k,s}$ wie sie im Beweis von Lemma 12 verwendet werden.



(b) Darstellung der in die Funktionen $\bar{f}_{k,s}^{(i)}$ und $f_{k,s}$ gemäß Ungleichung (3.51) eingesetzten diskreten bzw. kontinuierlichen Teilbereiche mit $j = 4 \cdot (2-1) + 2 = 6$.

Abbildung 3.5.: Skizzen zum Beweis von Lemma 12 mit $\alpha = \alpha_i = \pi/6$, $\lambda = 100$ und $h = 2^5/(\sqrt{2} \cdot \lambda)$.

und definieren

$$\mathbf{i}_{k-1,4\cdot(s-1)+j}^{(i)} = \lambda \cdot \left(\bar{\mathbf{z}}_{k-1,4\cdot(s-1)+j}^{(i)} - \bar{\mathbf{z}}_{k,s}^{(i)} \right) \quad \left(k = 1, \ldots, l, s = 1, \ldots, 4^{l-k}, j = 1, \ldots, 4 \right).$$

Um zu zeigen, dass die Gitterpunkte $\mathbf{i}_{k,s}^{(i)}$ gemäß Definition 7 b) wohldefiniert sind, verwenden wir $h \leq 2^l/(\sqrt{2}\cdot\lambda)$ und berechnen

$$\left\| rot^{(\beta)} \left(\mathbf{h}_{k-2}^{(j)} \right) \right\|_\infty \leq \sqrt{2} \cdot h_{k-2} = \frac{\sqrt{2} \cdot h}{2^{l-(k-2)}} \leq \frac{2^{k-2}}{\lambda} \tag{3.54}$$

für $k = 1, \ldots, l$, $j = 1, \ldots, 4$ sowie einen beliebigen Winkel $\beta \in [0, 2\pi]$ und erhalten deshalb

$$\|\mathbf{z}_{k-1,4\cdot(s-1)+j}^{(i)}\|_\infty \leq \|\mathbf{z}_{k,s}^{(i)}\|_\infty + \|rot^{(\alpha_i)} \left(\mathbf{h}_{k-2}^{(j)} \right)\|_\infty \leq \|\mathbf{z}_{k,s}^{(i)}\|_\infty + \frac{2^{k-2}}{\lambda}$$

für $k = 1, \ldots, l$, $s = 1, \ldots, 4^{l-k}$ und $j = 1, \ldots, 4$. Wegen $\mathbf{z}_{l,1} = (0,0)$ gilt dann

$$\|\mathbf{z}_{k,s}^{(i)}\|_\infty \leq \sum_{j=k+1}^{l} \frac{2^{j-2}}{\lambda} = \frac{1}{2\cdot\lambda} \left(\sum_{j=0}^{l-1} 2^j - \sum_{j=0}^{k-1} 2^j \right) = \frac{2^l - 2^k}{2\cdot\lambda}$$

und wegen (3.53) auch

$$\|\mathbf{z}_{k,s}^{(i)} - \bar{\mathbf{z}}_{k,s}^{(i)}\|_\infty \leq \frac{1}{2\cdot\lambda} \tag{3.55}$$

für $k = 0, \ldots, l$, $s = 1, \ldots, 4^{l-k}$. Unter Verwendung der Dreiecksungleichung, Ungleichung (3.55) und (3.54) ergibt sich

$$\|\mathbf{i}_{k-1,4\cdot(s-1)+j}^{(i)}\|_\infty$$

$$
\begin{aligned}
&= \lambda \cdot \|\bar{\mathbf{z}}_{k,s}^{(i)} - \bar{\mathbf{z}}_{k-1,4\cdot(s-1)+j}^{(i)}\|_\infty \\
&\leq \lambda \cdot \left(\|\bar{\mathbf{z}}_{k,s}^{(i)} - \mathbf{z}_{k,s}^{(i)}\|_\infty + \|\mathbf{z}_{k,s}^{(i)} - \mathbf{z}_{k-1,4\cdot(s-1)+j}^{(i)}\|_\infty + \|\mathbf{z}_{k-1,4\cdot(s-1)+j}^{(i)} - \bar{\mathbf{z}}_{k-1,4\cdot(s-1)+j}^{(i)}\|_\infty \right) \\
&\leq \lambda \cdot \left(\frac{1}{2\cdot\lambda} + \|rot^{(\alpha_i)}(\mathbf{h}_{k-2}^{(j)})\|_\infty + \frac{1}{2\cdot\lambda} \right) \\
&\leq 2^{k-2} + 1
\end{aligned}
$$

für $k = 1, \ldots, l$, $s = 1, \ldots, 4^{l-k}$ und $j = 1, \ldots, 4$, was zusammen mit dem Fakt, dass $\mathbf{i}_{k,s}^{(i)}$ ein Vektor von ganzen Zahlen ist, impliziert, dass

$$
\mathbf{i}_{k,s}^{(i)} \in \left\{ -(\lfloor 2^{k-1} \rfloor + 1), \ldots, 0, \ldots, \lfloor 2^{k-1} \rfloor + 1 \right\}^2 \quad \left(k = 0, \ldots, l-1, s = 1, \ldots, 4^{l-k}\right).
$$

Schritt 2: Für $k = 1, \ldots, l$, $s = 1, \ldots, 4^{l-k}$ und $j = 1, \ldots, 4$ gilt

$$
\begin{aligned}
&\|\mathbf{z}_{k-1,4\cdot(s-1)+j}^{(i)} - (\mathbf{v}_{k-1,4\cdot(s-1)+j} - \mathbf{v})\|_\infty \\
&\leq \|\mathbf{z}_{k,s}^{(i)} - (\mathbf{v}_{k,s} - \mathbf{v})\|_\infty + \left\| rot^{(\alpha_i)}\left(\mathbf{h}_{k-2}^{(j)}\right) - rot^{(\alpha)}\left(\mathbf{h}_{k-2}^{(j)}\right) \right\|_\infty \\
&= \|\mathbf{z}_{k,s}^{(i)} - (\mathbf{v}_{k,s} - \mathbf{v})\|_\infty + \left\| \begin{pmatrix} \cos(\alpha_i) - \cos(\alpha) & \sin(\alpha) - \sin(\alpha_i) \\ \sin(\alpha) - \sin(\alpha_i) & \cos(\alpha) - \cos(\alpha_i) \end{pmatrix} \mathbf{h}_{k-2}^{(j)} \right\|_\infty \\
&\leq \|\mathbf{z}_{k,s}^{(i)} - (\mathbf{v}_{k,s} - \mathbf{v})\|_\infty + 2 \cdot h_{k-2} \cdot \max\{|\sin(\alpha) - \sin(\alpha_i)|, |\cos(\alpha) - \cos(\alpha_i)|\} \\
&\leq \|\mathbf{z}_{k,s}^{(i)} - (\mathbf{v}_{k,s} - \mathbf{v})\|_\infty + h_{k-1} \cdot |\alpha - \alpha_i| \\
&\leq \|\mathbf{z}_{k,s}^{(i)} - (\mathbf{v}_{k,s} - \mathbf{v})\|_\infty + \frac{2^{k-1}}{\sqrt{2}\cdot\lambda} \cdot \frac{\sqrt{2}\cdot(c-1)}{2^l},
\end{aligned}
$$

was zusammen mit $\mathbf{z}_{l,1}^{(i)} = \mathbf{v}_{l,1} - \mathbf{v} = (0,0)$ impliziert, dass

$$
\|\mathbf{z}_{k,s}^{(i)} - (\mathbf{v}_{k,s} - \mathbf{v})\|_\infty \leq \frac{c-1}{2^l \cdot \lambda} \cdot \sum_{i=k}^{l-1} 2^i = \frac{(c-1)\cdot(2^l - 2^k)}{\lambda \cdot 2^l} < \frac{c-1}{\lambda} \tag{3.56}
$$

für $k = 0, \ldots, l$ und $s = 1, \ldots, 4^{l-k}$. Weiterhin gilt

$$
\mathbf{u}_{k,s} = \varphi(\mathbf{u}) + \bar{\mathbf{z}}_{k,s}^{(i)} \tag{3.57}
$$

für $k = 0, \ldots, l$ wegen $\mathbf{u}_{l,1} = \varphi(\mathbf{u})$, $\bar{\mathbf{z}}_{l,1}^{(i)} = (0,0)$ und

$$
\mathbf{u}_{k-1,4\cdot(s-1)+j} = \mathbf{u}_{k,s} + \frac{1}{\lambda} \cdot \mathbf{i}_{k-1,4\cdot(s-1)+j}^{(i)} = \mathbf{u}_{k,s} + \bar{\mathbf{z}}_{k-1,4\cdot(s-1)+j}^{(i)} - \bar{\mathbf{z}}_{k,s}^{(i)}
$$

für $k = 1, \ldots, l$, $s = 1, \ldots, 4^{l-k}$ und $j = 1, \ldots, 4$. Wegen den Ungleichungen (3.55), (3.56) und (3.57) ergibt sich

$$
\begin{aligned}
\|\mathbf{u}_{k,s} - \mathbf{v}_{k,s}\|_\infty &= \|\varphi(\mathbf{u}) - \mathbf{v} + \bar{\mathbf{z}}_{k,s}^{(i)} - \mathbf{z}_{k,s}^{(i)} + \mathbf{z}_{k,s}^{(i)} - \mathbf{v}_{k,s} + \mathbf{v}\|_\infty \\
&\leq \|\varphi(\mathbf{u}) - \mathbf{v}\|_\infty + \|\bar{\mathbf{z}}_{k,s}^{(i)} - \mathbf{z}_{k,s}^{(i)}\|_\infty + \left\| \mathbf{z}_{k,s}^{(i)} - (\mathbf{v}_{k,s} - \mathbf{v}) \right\|_\infty \\
&\leq \frac{1}{2\cdot\lambda} + \frac{1}{2\cdot\lambda} + \frac{c-1}{\lambda} \\
&= \frac{c}{\lambda}
\end{aligned} \tag{3.58}
$$

für alle $k = 0, \ldots, l$ und $s = 1, \ldots, 4^{l-k}$.

Schritt 3: Um Annahme 2 zu verwenden, zeigen wir zunächst, dass $\mathbf{v}_{0,s} \in [h_0/\sqrt{2} - 1/2, 1/2 - h_0/\sqrt{2}]^2$ für alle $s = 1, \ldots, 4^l$. Durch Anwendung von Ungleichung (3.54) ergibt sich

$$\|\mathbf{v}_{k-1,4\cdot(s-1)+j} - \mathbf{v}\|_\infty \leq \|\mathbf{v}_{k,s} - \mathbf{v}\|_\infty + \|rot^{(\alpha)}\left(\mathbf{h}_{k-2}^{(j)}\right)\|_\infty \leq \|\mathbf{v}_{k,s} - \mathbf{v}\|_\infty + \frac{2^{k-2}}{\lambda}$$

für $k = 1, \ldots, l$, $s = 1, \ldots, 4^{l-k}$ und $j = 1, \ldots, 4$, was zusammen mit $\mathbf{v}_{l,1} = \mathbf{v}$ impliziert, dass

$$\|\mathbf{v}_{k,s} - \mathbf{v}\|_\infty \leq \sum_{j=k+1}^{l} \frac{2^{j-2}}{\lambda} = \frac{1}{2\cdot\lambda}\left(\sum_{j=0}^{l-1} 2^j - \sum_{j=0}^{k-1} 2^j\right) = \frac{2^l - 2^k}{2\cdot\lambda} \tag{3.59}$$

für $k = 0, \ldots, l$ und $s = 1, \ldots, 4^{l-k}$. Unter Verwendung von Ungleichung (3.59), $\mathbf{v} \in [-1/2 + b, 1/2 - b]^2$ und $h_0 \leq 1/(\sqrt{2}\cdot\lambda)$ erhalten wir

$$\begin{aligned}
\|\mathbf{v}_{0,s}\|_\infty &\leq \|\mathbf{v}\|_\infty + \|\mathbf{v}_{0,s} - \mathbf{v}\|_\infty \\
&\leq \frac{1}{2} - b + \frac{2^l - 1}{2\cdot\lambda} \\
&\leq \frac{1}{2} - \frac{2^l + 2\cdot l - 1}{2\cdot\lambda} + \frac{2^l - 1}{2\cdot\lambda} \\
&= \frac{1}{2} - \frac{l}{\lambda} \\
&\leq \frac{1}{2} - \frac{1/(\sqrt{2}\cdot\lambda)}{\sqrt{2}} \\
&\leq \frac{1}{2} - \frac{h_0}{\sqrt{2}}
\end{aligned}$$

für $s = 1, \ldots, 4^l$. Da außerdem Ungleichung (3.58) gilt, können wir Annahme 2 verwenden und erhalten

$$\begin{aligned}
&\left|\bar{f}_{0,s}^{(i)}(\mathbf{x}_{\varphi^{-1}(\mathbf{u}_{0,s})+I^{(0)}}) - f_{0,s}(\phi \circ \tau_{\mathbf{v}_{0,s}} \circ rot^{(\alpha)}|_{C_{h_0}})\right| \\
&= \left|g_{0,s}(x_{\varphi^{-1}(\mathbf{u}_{0,s})}) - f_{0,s}(\phi \circ \tau_{\mathbf{v}_{0,s}} \circ rot^{(\alpha)}|_{C_{h_0}})\right| \\
&= \left|f_{0,s}(\phi(\mathbf{u}_{0,s})\cdot 1_{C_{h_0}}) - f_{0,s}(\phi \circ \tau_{\mathbf{v}_{0,s}} \circ rot^{(\alpha)}|_{C_{h_0}})\right| \\
&\leq \epsilon_\lambda
\end{aligned}$$

für $s = 1, \ldots, 4^l$.

Schritt 4: Nun nehmen wir an, Ungleichung (3.51) gelte für ein $k \in \{0, \ldots, l-1\}$ und alle $s \in \{1, \ldots, 4^{l-k}\}$. Wegen Definition (3.52) und der Definition der Abbildung φ^{-1} (siehe Gleichung (3.47)) gilt

$$\begin{aligned}
\varphi^{-1}(\mathbf{u}_{k,s}) + \mathbf{i}_{k-1,4\cdot(s-1)+j}^{(i)} &= \lambda\cdot\left(\mathbf{u}_{k,s} + \frac{1}{2}\right) + \frac{1}{2} + \mathbf{i}_{k-1,4\cdot(s-1)+j}^{(i)} \\
&= \lambda\cdot\left(\mathbf{u}_{k,s} + \frac{1}{\lambda}\cdot\mathbf{i}_{k-1,4\cdot(s-1)+j}^{(i)} + \frac{1}{2}\right) + \frac{1}{2} \\
&= \varphi^{-1}(\mathbf{u}_{k-1,4\cdot(s-1)+j})
\end{aligned}$$

für $k = 1, \ldots, l$, $s = 1, \ldots, 4^{l-k}$ und $j = 1, \ldots, 4$, was mit der Lipschitzbedingung an die Funktionen $g_{k+1,s}$, der Linearität der Funktion $rot^{(\alpha)}$ und der Induktionshypothese (3.51) impliziert, dass

$$\left|\bar{f}_{k+1,s}^{(i)}(\mathbf{x}_{\varphi^{-1}(\mathbf{u}_{k+1,s})+I^{(k+1)}}) - f_{k+1,s}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ rot^{(\alpha)}|_{C_{h_{k+1}}})\right|$$

$$
\begin{aligned}
&= \Big| g_{k+1,s}\Big(\bar{f}^{(i)}_{k,4\cdot(s-1)+1}(\mathbf{x}_{\varphi^{-1}(\mathbf{u}_{k+1,s})+\mathbf{i}^{(i)}_{k,4\cdot(s-1)+1}+I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot(s-1)+2}(\mathbf{x}_{\varphi^{-1}(\mathbf{u}_{k+1,s})+\mathbf{i}^{(i)}_{k,4\cdot(s-1)+2}+I^{(k)}}), \\
&\qquad \bar{f}^{(i)}_{k,4\cdot(s-1)+3}(\mathbf{x}_{\varphi^{-1}(\mathbf{u}_{k+1,s})+\mathbf{i}^{(i)}_{k,4\cdot(s-1)+3}+I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot s}(\mathbf{x}_{\varphi^{-1}(\mathbf{u}_{k+1,s})+\mathbf{i}^{(i)}_{k,4\cdot s}+I^{(k)}}) \Big) \\
&\quad - g_{k+1,s}\Big(f_{k,4\cdot(s-1)+1}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ rot^{(\alpha)} \circ \tau_{(-h_{k-1},-h_{k-1})}\big|_{C_{h_k}}), \\
&\qquad f_{k,4\cdot(s-1)+2}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ rot^{(\alpha)} \circ \tau_{(h_{k-1},-h_{k-1})}\big|_{C_{h_k}}), \\
&\qquad f_{k,4\cdot(s-1)+3}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ rot^{(\alpha)} \circ \tau_{(-h_{k-1},h_{k-1})}\big|_{C_{h_k}}), \\
&\qquad f_{k,4\cdot s}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ rot^{(\alpha)} \circ \tau_{(h_{k-1},h_{k-1})}\big|_{C_{h_k}}) \Big) \Big| \\
&= \Big| g_{k+1,s}\Big(\bar{f}^{(i)}_{k,4\cdot(s-1)+1}(\mathbf{x}_{\varphi^{-1}(\mathbf{u}_{k,4\cdot(s-1)+1})+I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot(s-1)+2}(\mathbf{x}_{\varphi^{-1}(\mathbf{u}_{k,4\cdot(s-1)+2})+I^{(k)}}), \\
&\qquad \bar{f}^{(i)}_{k,4\cdot(s-1)+3}(\mathbf{x}_{\varphi^{-1}(\mathbf{u}_{k,4\cdot(s-1)+3})+I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot s}(\mathbf{x}_{\varphi^{-1}(\mathbf{u}_{k,4\cdot s})+I^{(k)}}) \Big) \\
&\quad - g_{k+1,s}\Big(f_{k,4\cdot(s-1)+1}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ \tau_{rot^{(\alpha)}(\mathbf{h}^{(1)}_{k-1})} \circ rot^{(\alpha)}\big|_{C_{h_k}}), \\
&\qquad f_{k,4\cdot(s-1)+2}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ \tau_{rot^{(\alpha)}(\mathbf{h}^{(2)}_{k-1})} \circ rot^{(\alpha)}\big|_{C_{h_k}}), \\
&\qquad f_{k,4\cdot(s-1)+3}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ \tau_{rot^{(\alpha)}(\mathbf{h}^{(3)}_{k-1})} \circ rot^{(\alpha)}\big|_{C_{h_k}}), \\
&\qquad f_{k,4\cdot s}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ \tau_{rot^{(\alpha)}(\mathbf{h}^{(4)}_{k-1})} \circ rot^{(\alpha)}\big|_{C_{h_k}}) \Big) \Big| \\
&\leq L \cdot \max_{j\in\{1,\dots,4\}} \Big| \bar{f}^{(i)}_{k,4\cdot(s-1)+j}(\mathbf{x}_{\varphi^{-1}(\mathbf{u}_{k,4\cdot(s-1)+j})+I^{(k)}}) \\
&\qquad\qquad - f_{k,4\cdot(s-1)+j}(\phi \circ \tau_{\mathbf{v}_{k,4\cdot(s-1)+j}} \circ rot^{(\alpha)}\big|_{C_{h_k}}) \Big| \\
&\leq L^{k+1} \cdot \epsilon_\lambda
\end{aligned}
$$

für alle $s \in \{1, \dots, 4^{l-(k+1)}\}$. □

Wir zeigen nun, wie wir den Fehler, der auftritt, wenn die Funktionen $g^{(i)}_{k,s}$ in einem diskretisierten hierarchischen Max-Pooling Modell durch Approximationen $\bar{g}^{(i)}_{k,s}$ ersetzt werden, abschätzen können.

Lemma 13. *Es seien $\lambda, l, t \in \mathbb{N}$ mit $2^l + 2 \cdot l - 1 \leq \lambda$ und es seien*

$$
g^{(i)}_{k,s} : \mathbb{R}^4 \to [0,1],\ \bar{g}^{(i)}_{k,s} : \mathbb{R}^4 \to \mathbb{R}^+_0 \quad \big(i = 1, \dots, t, k = 1, \dots, l, s = 1, \dots, 4^{l-k}\big)
$$

sowie

$$
g^{(i)}_{0,s} : [0,1] \to [0,1],\ \bar{g}^{(i)}_{0,s} : [0,1] \to [0,2] \quad \big(i = 1, \dots, t, s = 1, \dots, 4^l\big)
$$

Funktionen, sodass die Einschränkungen $\{g^{(i)}_{k,s}\big|_{[0,2]^4}\}_{i=1,\dots,t,k=1,\dots,l,s=1,\dots,4^{l-k}}$ lipschitzstetig mit Lipschitzkonstante $C > 0$ sind (bezüglich der durch die Maximumsnorm induzierten Metrik) und

$$
\left\| \bar{g}^{(i)}_{k,s} \right\|_{[0,2]^4,\infty} \leq 2 \quad \big(i = 1, \dots, t, k = 1, \dots, l, s = 1, \dots, 4^{l-k}\big).
$$

Es sei $\eta : [0,1]^{\{1,\dots,\lambda\}^2} \to \mathbb{R}$ eine Funktion, die einem diskretisierten hierarchischen Max-Pooling Modell vom Level l der Ordnung d mit Funktionen $g^{(i)}_{k,s}$ genügt und es sei $\bar{\eta} : [0,1]^{\{1,\dots,\lambda\}^2} \to \mathbb{R}$ eine Funktion, die einem diskretisierten hierarchischen Max-Pooling Modell vom Level l, der Ordnung t mit Funktionen $\bar{g}^{(i)}_{k,s}$ genügt. Außerdem nehme an,

dass die beiden diskretisierten hierarchischen Max-Pooling Modelle die gleichen Gitterpunkte $\{\mathbf{i}_{k,s}^{(i)}\}$ besitzen. Dann gilt für alle $\mathbf{x} \in [0,1]^{\{1,\dots,\lambda\}^2}$, dass

$$|\eta(\mathbf{x}) - \bar{\eta}(\mathbf{x})| \leq (C+1)^l \cdot \max_{\substack{i\in\{1,\dots,t\},j\in\{1,\dots,4^l\},\\ k\in\{1,\dots,l\},s\in\{1,\dots,4^{l-k}\}}} \left\{ \|g_{0,j}^{(i)} - \bar{g}_{0,j}^{(i)}\|_{[0,1],\infty}, \|g_{k,s}^{(i)} - \bar{g}_{k,s}^{(i)}\|_{[0,2]^4,\infty} \right\}.$$

Beweis. Der Beweis ist dem Beweis von Lemma 3 sehr ähnlich und verläuft weitestgehend analog. Daher befindet sich der Beweis im Anhang dieser Arbeit. □

Als Nächstes zeigen wir, dass wir ein diskretisiertes hierarchisches Max-Pooling Modell durch ein faltendes neuronales Netz darstellen können, wenn die Funktionen $\bar{g}_{k,s}^{(i)}$ des diskretisierten hierarchischen Modells vollverbundenen neuronalen Netzen entsprechen.

Lemma 14. *Es seien $\lambda, l, t \in \mathbb{N}$ mit $2^l + 2 \cdot l - 1 \leq \lambda$. Für $L_{net}, r_{net} \in \mathbb{N}$ seien*

$$g_{net,m,s}^{(i)} \in \mathcal{G}_4(L_{net}, r_{net}) \quad \left(i = 1,\dots,t, m = 1,\dots,l, s = 1,\dots,4^{l-m}\right)$$

und

$$g_{net,0,s}^{(i)} \in \mathcal{G}_1(L_{net}, r_{net}) \quad \left(i = 1,\dots,t, s = 1,\dots,4^{l}\right).$$

Es sei $\bar{\eta} : [0,1]^{\{1,\dots,\lambda\}^2} \to \mathbb{R}$ eine Funktion, die einem diskretisierten hierarchischen Max-Pooling Modell vom Level l, der Ordnung t mit Funktionen $\{\bar{g}_{m,s}^{(i)}\}$ genügt, wobei

$$\bar{g}_{m,s}^{(i)} = \sigma \circ g_{net,m,s}^{(i)} \quad \left(i = 1,\dots,t, m = 0,\dots,l, s = 1,\dots,4^{l-m}\right).$$

Setze $r_t = 3 \cdot t$, $k = 5 \cdot 4^{l-1} + r_{net}$, $A_1 = A_2 = 2^{l-1} + l$, $A_1' = A_2' = \lambda - 2^{l-1} - l + 1$,

$$L = \frac{4^{l+1} - 1}{3} \cdot (L_{net} + 1), \quad L_t = \begin{cases} \lceil \log_2 t \rceil & \text{, falls } t > 1 \\ 1 & \text{, falls } t = 1, \end{cases}$$

$\mathbf{A} = (A_1, A_1', A_2, A_2')$ und für $r = 0,\dots,l$ setze

$$M_s = I_{\{r>1\}} \cdot 2^{r-1} + 3 \quad \left(s = \sum_{i=0}^{r-1} 4^{l-i} \cdot (L_{net}+1) + 1, \dots, \sum_{i=0}^{r} 4^{l-i} \cdot (L_{net}+1)\right),$$

wobei wir die leere Summe als 0 definieren. Dann existiert ein $f_{CNN} \in \mathcal{F}_2(\boldsymbol{\theta})$ mit $\boldsymbol{\theta} = (t, L_t, r_t, L, k, \mathbf{M}, \mathbf{A})$, sodass

$$\bar{\eta}(\mathbf{x}) = f_{CNN}(\mathbf{x})$$

für alle $\mathbf{x} \in [0,1]^{\{1,\dots,\lambda\}^2}$.

Beweis. Der Beweis ist dem Beweis von Lemma 4 sehr ähnlich und verläuft weitestgehend analog. Daher befindet sich auch dieser Beweis im Anhang der Arbeit. □

Beweis von Lemma 11. Sei $\bar{\eta}$ das diskretisierte hierarchische Max-Pooling Modell vom Level l und der Ordnung t, welches durch die Funktionen $\{\bar{g}_{m,s}^{(i)}\}$ und die Gitterpunkte $\{\mathbf{i}_{m,s}^{(i)}\}$ aus Lemma 12 gegeben ist, sodass

$$|\eta(\phi) - \bar{\eta}(g_\lambda(\phi))| \leq c_{26} \cdot \epsilon_\lambda \tag{3.60}$$

für alle $\phi \in A$ und eine Konstante $c_{26} > 0$ gilt (für $p \in [1,\infty)$ folgt die Lipschitzstetigkeit der Einschränkungen $g_{m,s}\big|_{[0,1]^4}$ aus der (p,C)-Glattheit der Funktionen $g_{m,s}$). Außerdem seien $g_{net,0,s}^{(i)} \in \mathcal{G}_1(L_n, r_{net})$ und

$g_{net,m,s}^{(i)} \in \mathcal{G}_4(L_n, r_{net})$ $(m > 0)$ die vollverbundenen neuronalen Netze von Kohler und Langer (2021) aus Lemma 2, welche die Ungleichung

$$\left\|\bar{g}_{m,s}^{(i)} - \sigma \circ g_{net,m,s}^{(i)}\right\|_{[0,2]^4,\infty} \leq \left\|\bar{g}_{m,s}^{(i)} - g_{net,m,s}^{(i)}\right\|_{[0,2]^4,\infty} \leq c_{27} \cdot L_n^{-\frac{2\cdot p}{4}} \leq c_{28} \cdot n^{-\frac{p}{2\cdot p+4}}$$

für $i = 1, \dots, t$, $m = 1, \dots, l$, $s = 1, \dots, 4^{l-m}$ und Konstanten $c_{27}, c_{28} > 0$ sowie die Ungleichung

$$\left\|\bar{g}_{0,s}^{(i)} - \sigma \circ g_{net,0,s}^{(i)}\right\|_{[0,1],\infty} \leq \left\|\bar{g}_{0,s}^{(i)} - g_{net,0,s}^{(i)}\right\|_{[0,1],\infty} \leq c_{29} \cdot L_n^{-2\cdot p} \leq c_{30} \cdot n^{-\frac{p}{2\cdot p+1}},$$

für $i = 1, \dots, t$, $s = 1, \dots, 4^l$ und Konstanten $c_{29}, c_{30} > 0$ erfüllen (wegen der Wahl aus Lemma 12 und Annahme 1 haben die Funktionen $\{\bar{g}_{0,s}^{(i)}\}$ (p, C)-glatte Erweiterungen auf $\mathbb{R}$). Wir können die Konstante c_{17} in der Definition von L_n (siehe Theorem 3.3) hinreichend groß wählen, sodass wir wegen der Dreiecksungleichung und dem Fakt, dass die Funktionen $\bar{g}_{m,s}^{(i)}$ nur Werte im Intervall $[0, 1]$ annehmen, folgern können, dass

$$\left\|\sigma \circ g_{net,m,s}^{(i)}\right\|_{[0,2]^4,\infty} \leq \|\bar{g}_{m,s}^{(i)}\|_{[0,2]^4,\infty} + \left\|\bar{g}_{m,s}^{(i)} - \sigma \circ g_{net,m,s}^{(i)}\right\|_{[0,2]^4,\infty} \leq 1 + c_{27} \cdot L_n^{-\frac{2\cdot p}{4}} \leq 2$$

für alle $m = 1, \dots, l$ und $s = 1, \dots, 4^{l-m}$ sowie

$$\left\|\sigma \circ g_{net,0,s}^{(i)}\right\|_{[0,1],\infty} \leq \left\|\bar{g}_{0,s}^{(i)}\right\|_{[0,1],\infty} + \left\|\bar{g}_{0,s}^{(i)} - \sigma \circ g_{net,0,s}^{(i)}\right\|_{[0,1],\infty} \leq 1 + c_{29} \cdot L_n^{-2\cdot p} \leq 2$$

für alle $s = 1, \dots, 4^l$ gilt. Als Nächstes verwenden wir das faltende neuronale Netz $f_{CNN} \in \mathcal{F}_2(\boldsymbol{\theta})$ aus Lemma 14, sodass f_{CNN} einem diskretisierten hierarchischen Max-Pooling Modell genügt, welches durch die Funktionen $\{\sigma \circ g_{net,m,s}^{(i)}\}$ und die Gitterpunkte $\{\mathbf{i}_{m,s}^{(i)}\}$ gegeben ist. Unter der Verwendung von $(a+b)^2 \leq 2a^2+2b^2$, Ungleichung (3.60) und Lemma 13 erhalten wir

$$\begin{aligned}
|f_{CNN}(g_\lambda(\phi)) - \eta(\phi)|^2 &\leq 2 \cdot |f_{CNN}(g_\lambda(\phi)) - \bar{\eta}(g_\lambda(\phi))|^2 + 2 \cdot |\bar{\eta}(g_\lambda(\phi)) - \eta(\phi)|^2 \\
&\leq c_{31} \cdot \Big(\max_{m\in\{1,\dots,l\}, s\in\{1,\dots,4^{l-m}\}, j\in\{1,\dots,4^l\}, i\in\{1,\dots,t\}} \Big\{ \|\sigma \circ g_{net,0,j}^{(i)} - \bar{g}_{0,j}^{(i)}\|_{[0,1],\infty}, \\
&\qquad \|\sigma \circ g_{net,m,s}^{(i)} - \bar{g}_{m,s}^{(i)}\|_{[0,2]^4,\infty} \Big\} \Big)^2 + 2 \cdot c_{26}^2 \cdot \epsilon_\lambda^2 \\
&\leq c_{32} \cdot \left(n^{-\frac{2\cdot p}{2\cdot p+4}} + \epsilon_\lambda^2 \right)
\end{aligned}$$

für Konstanten $c_{31}, c_{32} > 0$, welche nicht von λ und n abhängen. □

3.5. Abschätzung der Überdeckungszahl von faltenden neuronalen Netzen

In diesem Kapitel wollen wir die in Definition 6 eingeführte L_1-ϵ-Überdeckungszahl der in dieser Arbeit verwendeten Funktionsklassen von faltenden neuronalen Netzen beschränken. Da die Überdeckungszahl monoton ist, d.h., es gilt

$$\mathcal{N}_p(\epsilon, \mathcal{F}, \mathbf{x}_1^n) \leq \mathcal{N}_p(\epsilon, \tilde{\mathcal{F}}, \mathbf{x}_1^n) \tag{3.61}$$

für $\mathcal{F} \subseteq \tilde{\mathcal{F}}$, genügt es, ein entsprechendes Resultat für eine Funktionsklasse von faltenden neuronalen Netzen zu zeigen, die alle in dieser Arbeit verwendeten Netzwerkarchitekturen enthält. Zu diesem Zweck führen wir eine Funktionsklasse von faltenden neuronalen Netzen ein, bei der, wie bei den in Abschnitt 3.1 eingeführten Funktionsklassen $\mathcal{F}_1(\boldsymbol{\theta})$ und $\mathcal{F}_2(\boldsymbol{\theta})$, $t \in \mathbb{N}$ faltende neuronale Netze parallel berechnet werden und auf die t Ausgaben ein vollverbundenes neuronales Netz aus $\mathcal{G}_t(L_{net}, r_{net})$ angewendet wird. Die Architektur der parallel

berechneten faltenden neuronalen Netze entspricht hierbei der in Abschnitt 3.1 eingeführten Bauart der Klasse $\mathcal{F}_5(\boldsymbol{\theta})$ mit dem Unterschied, dass wir ein beliebiges Zero-Padding verwenden. Das heißt wir verwenden Funktionen $f : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}$ der Form

$$f(\mathbf{x}) = f_{out}^{(\mathbf{A})} \circ f_{sub}^{(s)} \circ o_{(k,k),M_L,P_L,\mathbf{w}_L} \circ \cdots \circ o_{(1,k),M_1,P_1,\mathbf{w}_1}(\mathbf{x}) \quad \left(\mathbf{x} \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}\right) \tag{3.62}$$

und verwenden in diesem Abschnitt die durch

$$\mathcal{F}(\boldsymbol{\theta}) = \big\{f : f \text{ hat die Form (3.62) mit Parametern } \boldsymbol{\theta} = (L,k,\mathbf{M},s,\mathbf{P},\mathbf{A})\big\}, \tag{3.63}$$

definierte Klasse, wobei wir $\mathbf{P} = (P_1,\dots,P_L)$ gesetzt haben. Die beschriebene Funktionsklasse ist dann definiert durch

$$\mathcal{F}_6(\boldsymbol{\theta}) = \big\{g_{net} \circ (f_1,\dots,f_t) : g_{net} \in \mathcal{G}_t(L_{net}, r_{net}), f_1,\dots,f_t \in \mathcal{F}((L,k,\mathbf{M},s,\mathbf{P},\mathbf{A}))\big\} \tag{3.64}$$

und hängt von den Parametern $\boldsymbol{\theta} = (t, L_{net}, r_{net}, L, k, \mathbf{M}, s, \mathbf{P}, \mathbf{A})$ ab.

Bemerkung 3.9. Dass die in Abschnitt 3.1 definierten Netzwerkarchitekturen $\mathcal{F}_j(\boldsymbol{\theta}_j)$ $(j = 1,\dots,5)$ für entsprechende Parametervektoren in der Funktionsklasse $\mathcal{F}_6(\boldsymbol{\theta})$ enthalten sind, können wir folgendermaßen einsehen: Sind die Zero-Padding Parameter $\mathbf{P}_1$ und $\mathbf{P}_2$ definiert wie in Gleichung (3.10) ergibt sich

$$\mathcal{F}_1\big((t, L_{net}, r_{net}, L, k, \mathbf{M}, \mathbf{A})\big) = \mathcal{F}_6\big((t, L_{net}, r_{net}, L, k, \mathbf{M}, 1, \mathbf{P}_1, \mathbf{A})\big)$$

und

$$\mathcal{F}_2\big((t, L_{net}, r_{net}, L, k, \mathbf{M}, \mathbf{A})\big) = \mathcal{F}_6\big((t, L_{net}, r_{net}, L, k, \mathbf{M}, 1, \mathbf{P}_2, \mathbf{A})\big),$$

da die Subsampling Schicht $f_{sub}^{(1)}$ der Identität entspricht. Definieren wir die Parametervektoren

$$\boldsymbol{\theta}_3 = (L, k, \mathbf{M}, z, \mathbf{s}, \mathbf{A}), \quad \boldsymbol{\theta}_4 = (L, \bar{k}, \mathbf{M}, \bar{z}, \mathbf{s}, \mathbf{A}), \quad \boldsymbol{\theta}_5 = (L \cdot \bar{z}, \bar{k}, \bar{\mathbf{M}}, s, \mathbf{A})$$

gemäß Lemma 7 gilt außerdem

$$\mathcal{F}_3(\boldsymbol{\theta}_3) \subset \mathcal{F}_4(\boldsymbol{\theta}_4) \subset \mathcal{F}_5(\boldsymbol{\theta}_5) \subset \mathcal{F}_6\big((1,1,2, L \cdot \bar{z}, \bar{k}, \bar{\mathbf{M}}, s, \mathbf{P}_1, \mathbf{A})\big),$$

da wir $g_{net} \in \mathcal{G}_1(1,2)$ durch

$$g_{net}(x) = \sigma(x) - \sigma(-x) = \max\{x,0\} - \max\{-x,0\} = x \quad (x \in \mathbb{R})$$

wählen können.

Das Hauptresultat dieses Abschnitts ist dann die folgende Abschätzung der L_1-ϵ-Überdeckungszahl der Funktionsklasse $\mathcal{F}_6(\boldsymbol{\theta})$.

Lemma 15. *Es sei $\mathcal{F}_6(\boldsymbol{\theta})$ wie oben definiert mit dem Parametervektor $\boldsymbol{\theta} = (t, L_{net}, r_{net}, L, k, \mathbf{M}, s, \mathbf{P}, \mathbf{A})$. Setze*

$$k_{max} = \max\{k, t, r_{net}\}, \quad M_{max} = \max\{M_1,\dots,M_L\} \quad \textit{und} \quad L_{max} = \max\{L_{net}, L\}.$$

Nehme an, es gelte $d_1 \cdot d_2 > 1$ und $c_{33} \cdot \log n \geq 2$ für eine Konstante $c_{33} > 0$. Dann gilt für alle $\epsilon \in (0,1)$:

$$\sup_{\mathbf{x}_1^n \in (\mathbb{R}^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}})^n} \log\left(\mathcal{N}_1\left(\epsilon, T_{c_{33}\cdot\log n}\mathcal{F}_6(\boldsymbol{\theta}), \mathbf{x}_1^n\right)\right)$$
$$\leq c_{34} \cdot L_{max}^2 \cdot \log(L_{max} \cdot d_1 \cdot d_2) \cdot \log\left(\frac{c_{33} \cdot \log n}{\epsilon}\right)$$

für eine Konstante $c_{34} > 0$, welche nur von k_{max} und M_{max} abhängt.

Im Folgenden geben wir Hilfsresultate an, mit deren Hilfe wir am Ende dieses Abschnitts Lemma 15 beweisen werden. Um die Überdeckungszahl der Funktionsklasse $T_{c_{33}\cdot\log n}\mathcal{F}_6(\boldsymbol{\theta})$ von oben zu beschränken, führen wir in der folgenden Definition den Begriff der L_p-ϵ-*Packzahl* ein.

Definition 8. Es sei $\mathcal{F}$ eine Menge von Funktionen $f : \mathbb{R}^d \to \mathbb{R}$. Außerdem seien $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$ mit $\mathbf{x}_1^n = (\mathbf{x}_1, ..., \mathbf{x}_n)$, $\epsilon > 0$ und $1 \leq p < \infty$.

a) Ein Tupel $(g_1, ..., g_N)$ mit $g_i \in \mathcal{F}$ für alle $i \in \{1, ..., N\}$ für ein $N \in \mathbb{N}$ heißt L_p-ϵ-*Packung* von $\mathcal{F}$ auf $\mathbf{x}_1^n$, falls für alle $1 \leq j < k \leq N$ gilt

$$\left(\frac{1}{n}\sum_{i=1}^{n} |g_j(\mathbf{x}_i) - g_k(\mathbf{x}_i)|^p\right)^{\frac{1}{p}} \geq \epsilon.$$

b) Wir nennen

$$\mathcal{M}_p\left(\epsilon, \mathcal{F}, \mathbf{x}_1^n\right) = \sup\left\{N \in \mathbb{N} \ : \ (g_1, ..., g_N) \text{ ist } L_p\text{-}\epsilon\text{-Packung von } \mathcal{F} \text{ auf } \mathbf{x}_1^n\right\}$$

die L_p-ϵ-*Packzahl* von $\mathcal{F}$ auf $\mathbf{x}_1^n$ ($\sup \mathbb{N} = \infty$).

Mithilfe des folgendes Lemmas gelingt es, die L_1-ϵ-Überdeckungszahl unter Verwendung der L_1-ϵ-Packzahl von oben abzuschätzen.

Lemma 16. *Es sei $\mathcal{G}$ eine Klasse reellwertiger Funktionen auf $\mathbb{R}^d$ sowie $\epsilon > 0$. Dann gilt*

$$\mathcal{N}_1(\epsilon, \mathcal{G}, \mathbf{x}_1^n) \leq \mathcal{M}_1(\epsilon, \mathcal{G}, \mathbf{x}_1^n)$$

für alle $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$ mit $\mathbf{x}_1^n = (\mathbf{x}_1, ..., \mathbf{x}_n)$.

Beweis. Siehe Lemma 9.2 in Györfi et al. (2002). □

Das Ziel ist es nun, die L_1-ϵ-Packzahl von oben abzuschätzen. Zu diesem Zweck führen wir für eine Klasse von Teilmengen von $\mathbb{R}^d$ den Begriff der VC-Dimension ein.

Definition 9. Es sei $\emptyset \neq \mathcal{A} \subset \mathcal{P}(\mathbb{R}^d)$ (wobei $\mathcal{P}(\mathbb{R}^d)$ die Potenzmenge von $\mathbb{R}^d$ bezeichnet) und $n \in \mathbb{N}$.

a) Für $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$ definieren wir

$$s(\mathcal{A}, \{\mathbf{x}_1, ..., \mathbf{x}_n\}) = |\left\{A \cap \{\mathbf{x}_1, ..., \mathbf{x}_n\} \ : \ A \in \mathcal{A}\right\}|.$$

b) Wir nennen

$$S(\mathcal{A}, n) = \max_{\{\mathbf{x}_1, ..., \mathbf{x}_n\} \subset \mathbb{R}^d} s(\mathcal{A}, \{\mathbf{x}_1, ..., \mathbf{x}_n\})$$

den n-*ten Zerlegungskoeffizient* von $\mathcal{A}$.

c) Die *VC-Dimension* (Vapnik-Chervonenkis-Dimension) von $\mathcal{A}$ ist definiert als

$$\mathcal{V}_{\mathcal{A}} = \sup\{n \in \mathbb{N} \ : \ S(\mathcal{A}, n) = 2^n\}.$$

Die L_1-ϵ-Packzahl lässt sich nun wie folgt mithilfe der soeben eingeführten VC-Dimension abschätzen.

Lemma 17. *Es sei $\mathcal{G}$ eine Klasse von Funktionen $g : \mathbb{R}^d \to [-B, B]$ mit $\mathcal{V}_{G^+} \geq 2$ und $0 < \epsilon < \frac{B}{2}$, wobei*

$$\mathcal{G}^+ := \{\{(\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R} : g(\mathbf{x}) \geq y\} : g \in \mathcal{G}\}. \tag{3.65}$$

Dann gilt

$$\mathcal{M}_1(\epsilon, \mathcal{G}, \mathbf{x}_1^n) \leq 3 \cdot \left(\frac{4eB}{\epsilon} \log \frac{6eB}{\epsilon}\right)^{\mathcal{V}_{\mathcal{G}^+}}$$

für alle $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$ mit $\mathbf{x}_1^n = (\mathbf{x}_1, ..., \mathbf{x}_n)$.

Beweis. Siehe Theorem 9.4 in Györfi et al. (2002). □

Um wiederum die VC-Dimension einer Klasse von Teilmengen von $\mathbb{R}^d$ abzuschätzen, erweitern wir den Begriff der VC-Dimension auch auf Klassen von reellwertigen Funktionen:

Definition 10. Es bezeichne $\mathcal{H}$ eine Funktionsklasse von $\{0, 1\}$-wertigen Funktionen auf $\mathbb{R}^d$ und es sei $\mathcal{F}$ eine Klasse von reellwertigen Funktionen auf $\mathbb{R}^d$.

a) Für $m \in \mathbb{N}$ definieren wir die *Wachstumsfunktion* von $\mathcal{H}$ durch

$$\Pi_{\mathcal{H}}(m) := \max_{\mathbf{x}_1, ..., \mathbf{x}_m \in \mathbb{R}^d} |\{(h(\mathbf{x}_1), \ldots, h(\mathbf{x}_m)) : h \in \mathcal{H}\}|.$$

b) Die *VC-Dimension* (Vapnik-Chervonenkis-Dimension) von $\mathcal{H}$ definieren wir als

$$\text{VCdim}(\mathcal{H}) := \sup\{m \in \mathbb{N} : \Pi_{\mathcal{H}}(m) = 2^m\}.$$

c) Für $f \in \mathcal{F}$ verwenden wir die Notationen $\text{sgn}(f) := I_{\{f \geq 0\}}$ und $\text{sgn}(\mathcal{F}) := \{\text{sgn}(f) : f \in \mathcal{F}\}$. Die *VC-Dimension* von $\mathcal{F}$ ist dann definiert als

$$\text{VCdim}(\mathcal{F}) := \text{VCdim}(\text{sgn}(\mathcal{F})).$$

Einen entsprechenden Zusammenhang der VC-Dimension für Klassen von Teilmengen von $\mathbb{R}^d$ und der VC-Dimension für Klassen von reellwertigen Funktionen liefert der folgende in der Literatur bekannte Zusammenhang, dessen Beweis wir hier der Vollständigkeit halber anführen.

Lemma 18. *Es sei $\mathcal{F}$ eine Klasse von reellwertigen Funktionen auf $\mathbb{R}^d$. Außerdem sei die Menge $\mathcal{F}^+$ gemäß Gleichung* (3.65) *definiert. Weiterhin definiere die Klasse $\mathcal{H}$ von reellwertigen Funktionen auf $\mathbb{R}^d \times \mathbb{R}$ durch*

$$\mathcal{H} := \{h((\mathbf{x}, y)) = f(\mathbf{x}) - y : f \in \mathcal{F}\}.$$

Dann gilt

$$\mathcal{V}_{\mathcal{F}^+} = \text{VCdim}(\mathcal{H}).$$

Beweis. Für alle paarweise verschiedenen $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m) \in \mathbb{R}^d \times \mathbb{R}$ mit $m \in \mathbb{N}$ gilt

$$\begin{aligned}
s(\mathcal{F}^+, \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\}) &= \left|\left\{A \cap \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\} \ : \ A \in \mathcal{F}^+\right\}\right| \\
&= \left|\left\{\{(\mathbf{x}, y) \in \mathbb{R}^d \times \mathbb{R} : f(\mathbf{x}) \geq y\} \cap \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\} \ : \ f \in \mathcal{F}\right\}\right| \\
&= |\{\{(\mathbf{x}, y) \in \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)\} : f(\mathbf{x}) \geq y\} \ : \ f \in \mathcal{F}\}| \\
&= |\{\{i \in \{1, \ldots, m\} : f(\mathbf{x}_i) \geq y_i\} \ : \ f \in \mathcal{F}\}| \\
&= |\{(\text{sgn}(f(\mathbf{x}_1) - y_1), \ldots, \text{sgn}(f(\mathbf{x}_m) - y_m)) : f \in \mathcal{F}\}|
\end{aligned}$$

$$= |\{(\text{sgn}(h(\mathbf{x}_1, y_1)), \dots, \text{sgn}(h(\mathbf{x}_m, y_m))) : h \in \mathcal{H}\}|.$$

Es folgt, dass

$$S(\mathcal{F}^+, m) = \Pi_{\mathcal{H}}(m)$$

für alle $m \in \mathbb{N}$ gilt, was

$$\mathcal{V}_{\mathcal{F}^+} = \text{VCdim}(\mathcal{H})$$

impliziert. □

Wir können die VC-Dimension $\mathcal{V}_{\mathcal{F}^+}$ unserer Funktionsklasse also beschränken, indem wir die VC-Dimension $VCdim(\mathcal{H})$ der in Lemma 18 definierten Funktionsklasse $\mathcal{H}$ beschränken. Hierfür benötigen wir das folgende Hilfsresultat über die Anzahl der möglichen Vorzeichenvektoren, die durch Polynome begrenzten Grades erreicht werden. Dabei bezeichnet ein Polynom vom Grad höchstens $D \in \mathbb{N}$ in $W \in \mathbb{N}$ Variablen eine Funktion $f : \mathbb{R}^W \to \mathbb{R}$ der Form

$$f(w_1, \dots, w_W) = \sum_{\substack{k_1,\dots,k_W \in \mathbb{N}_0 \\ k_1+\dots+k_W \leq D}} c_{k_1,\dots,k_W} \cdot w_1^{k_1} \cdot w_2^{k_2} \cdot \ldots \cdot w_W^{k_W}$$

für $c_{k_1,\dots,k_W} \in \mathbb{R}$ (siehe Forster (2017), S.27).

Lemma 19. *Es seien $W, m \in \mathbb{N}$ mit $W \leq m$ und es seien $f_1, ..., f_m : \mathbb{R}^W \to \mathbb{R}$ Polynome von Grad höchstens D in W Variablen. Definiere*

$$K \coloneqq |\{(\text{sgn}(f_1(\mathbf{w})), \dots, \text{sgn}(f_m(\mathbf{w}))) : \mathbf{w} \in \mathbb{R}^W\}|. \tag{3.66}$$

Dann gilt

$$K \leq 2 \cdot \left(\frac{2 \cdot e \cdot m \cdot D}{W}\right)^W.$$

Beweis. Siehe Theorem 8.3 in Anthony und Bartlett (1999). □

Das nächste Lemma stellt eine Modifikation von Theorem 6 in Bartlett et al. (2019) bereit, mit dessen Hilfe wir die VC-Dimension unserer Funktionsklasse abschätzen können.

Lemma 20. *Es sei $\mathcal{F}_6(\boldsymbol{\theta})$ die durch Gleichung (3.64) definierte Klasse von faltenden neuronalen Netzen mit dem Parametervektor $\boldsymbol{\theta} = (t, L_{net}, r_{net}, L, k, \mathbf{M}, s, \mathbf{P}, \mathbf{A})$. Setze*

$$k_{max} = \max\{k, t, r_{net}\}, \quad M_{max} = \max\{M_1, \dots, M_L\} \quad L_{max} = \max\{L_{net}, L\}.$$

Nehme weiterhin an, dass $d_1 \cdot d_2 > 1$. Dann gilt

$$\mathcal{V}_{\mathcal{F}_6(\boldsymbol{\theta})^+} \leq c_{35} \cdot L_{max}^2 \cdot \log_2(L_{max} \cdot d_1 \cdot d_2)$$

für eine Konstante $c_{35} > 0$, welche nur von k_{max} und M_{max} abhängt.

Beweis. Im Beweis benutzen wir die Notation

$$\mathbf{k} = (k_0, \dots, k_{L+L_{net}+2}) = (1, \underbrace{k, \dots, k}_{L \text{ mal}}, t, \underbrace{r_{net}, \dots, r_{net}}_{L_{net} \text{ mal}}, 1).$$

Wir verwenden Lemma 18, um $\mathcal{V}_{\mathcal{F}_6(\boldsymbol{\theta})^+}$ durch $\mathrm{VCdim}(\mathcal{H})$ zu beschränken, wobei $\mathcal{H}$ die Klasse der reellwertigen Funktionen auf $[0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \times \mathbb{R}$, definiert durch

$$\mathcal{H} \coloneqq \{h((\mathbf{x},y)) = f(\mathbf{x}) - y : f \in \mathcal{F}_6(\boldsymbol{\theta})\},$$

bezeichnet. Es sei $h \in \mathcal{H}$. Dann hängt h von t faltenden neuronalen Netzen

$$f_1, \dots, f_t \in \mathcal{F}((L,k,\mathbf{M},s,\mathbf{P},\mathbf{A}))$$

(siehe Gleichung (3.63) für die Definition der Klasse $\mathcal{F}(\boldsymbol{\theta})$) und einem vollverbundenen neuronalen Netz $g_{net} \in \mathcal{G}_t(L_{net}, r_{net})$ ab, sodass

$$h((\mathbf{x},y)) = g_{net} \circ (f_1, \dots, f_t)(\mathbf{x}) - y.$$

Die t faltenden neuronalen Netze hängen dabei von den Gewichtsvektoren

$$\mathbf{w}^{(b)} = \left(w^{(b,r)}_{i,j,s_1,s_2}\right)_{1\le i,j\le M_r, s_1\in\{1,\dots,k_{r-1}\}, s_2\in\{1,\dots,k_r\}, r\in\{1,\dots,L\}} \quad (b=1,\dots,t),$$

den Bias-Termen

$$\mathbf{w}^{(b)}_{bias} = \left(w^{(b,r)}_{s_2}\right)_{s_2\in\{1,\dots,k_r\}, r\in\{1,\dots,L\}} \quad (b=1,\dots,t)$$

und den Gewichten der Ausgabeschichten

$$\mathbf{w}^{(b)}_{out} = (w^{(b)}_{s_2})_{s_2\in\{1,\dots,k_L\}} \quad (b=1,\dots,t)$$

ab. Das vollverbundene neuronale Netz $g_{net} \in \mathcal{G}_t(L_{net}, r_{net})$ hängt von den inneren Gewichten

$$w^{(r-1)}_{i,j}$$

für $r \in \{2,\dots,L_{net}\}$, $j \in \{0,\dots,r_{net}\}$ und $i \in \{1,\dots,r_{net}\}$ sowie

$$w^{(0)}_{i,j}$$

für $j \in \{0,\dots,t\}$, $i \in \{1,\dots,r_{net}\}$ und den äußeren Gewichten

$$w^{(L_{net})}_{i}$$

für $i \in \{0,\dots,r_{net}\}$ ab. Wir zählen als Nächstes die Anzahl der reellwertigen Gewichte, welche bis einschließlich zur r-ten Schicht im faltenden Teil unserer Architektur verwendet werden: Wir setzen $W_0 \coloneqq 0$ und

$$W_r \coloneqq t \cdot \left(\sum_{s_2=1}^{r} M_{s_2}^2 \cdot k_{s_2} \cdot k_{s_2-1} + \sum_{s_2=1}^{r} k_{s_2}\right) \quad (r=1,\dots,L)$$

sowie

$$W_{L+1} \coloneqq W_L + t \cdot k_L.$$

Wir fahren im vollverbunden Teil der Architektur fort und zählen die Gewichte, die bis einschließlich zur r-ten vollverbundenen Schicht verwendet werden:

$$W_{L+1+r} = W_{L+r} + (k_{L+r}+1) \cdot k_{L+r+1} \quad (r=1,\dots,L_{net}).$$

Die Gesamtzahl der verwendeten Gewichte bezeichnen wir mit

$$\begin{aligned} W &= W_{L+L_{net}+2} \\ &= W_{L+L_{net}+1} + k_{L+L_{net}+1} + 1 \\ &\leq L \cdot t \cdot \left(M_{max}^2 \cdot k_{max}^2 + k_{max}\right) + t \cdot k_{max} \\ &\quad + L_{net} \cdot ((k_{max}+1) \cdot k_{max}) + k_{max} + 1 \\ &\leq L \cdot t \cdot \left(M_{max}^2 \cdot (k_{max}+1) \cdot k_{max}\right) \\ &\quad + L_{net} \cdot ((k_{max}+1) \cdot k_{max}) \\ &\quad + 2 \cdot t \cdot (k_{max}+1) \\ &\leq (L + L_{net} + 2) \cdot t \cdot M_{max}^2 \cdot (k_{max}+1) \cdot k_{max} \\ &\leq 2 \cdot (L + L_{net} + 2) \cdot t \cdot M_{max}^2 \cdot k_{max}^2. \end{aligned} \tag{3.67}$$

Als Nächstes definieren wir einen Vektor, der alle Gewichte enthält, welche bis einschließlich zur r-ten Schicht verwendet werden: Wir bezeichnen mit $\mathbf{w}_{\{1,\dots,W_0\}}$ den leeren Vektor und setzen

$$\begin{aligned} \mathbf{w}_{\{1,\dots,W_r\}} \coloneqq \Big(&\mathbf{w}_{\{1,\dots,W_{r-1}\}}, w_{1,1,1,1}^{(1,r)}, \dots, w_{M_r,M_r,k_{r-1},k_r}^{(1,r)}, w_1^{(1,r)}, \dots, w_{k_r}^{(1,r)}, \\ &\dots, w_{1,1,1,1}^{(t,r)}, \dots, w_{M_r,M_r,k_{r-1},k_r}^{(t,r)}, w_1^{(t,r)}, \dots, w_{k_r}^{(t,r)}\Big) \in \mathbb{R}^{W_r} \quad (r = 1, \dots, L), \end{aligned}$$

$$\mathbf{w}_{\{1,\dots,W_{L+1}\}} \coloneqq (\mathbf{w}_{\{1,\dots,W_L\}}, w_1^{(1)}, \dots, w_{k_L}^{(1)}, \dots, w_1^{(t)}, \dots, w_{k_L}^{(t)}) \in \mathbb{R}^{W_{L+1}}$$

sowie

$$\mathbf{w}_{\{1,\dots,W_{r+L+1}\}} \coloneqq \left(\mathbf{w}_{\{1,\dots,W_{r+L}\}}, w_{1,0}^{(r-1)}, \dots, w_{k_{r+L+1},k_{r+L}}^{(r-1)}\right) \in \mathbb{R}^{W_{r+L+1}} \quad (r = 1, \dots, L_{net})$$

und

$$\mathbf{w} \coloneqq \left(\mathbf{w}_{\{1,\dots,W_{L+L_{net}+1}\}}, w_0^{(L_{net})}, \dots, w_{L_{net}}^{(L_{net})}\right) \in \mathbb{R}^W.$$

Mit der obigen Notation können wir unsere Funktionsklasse umschreiben zu

$$\mathcal{H} = \{(\mathbf{x}, y) \mapsto h((\mathbf{x}, y), \mathbf{w}) : \mathbf{w} \in \mathbb{R}^W\}$$

und

$$\mathcal{F}(L, k, \mathbf{M}, s, \mathbf{P}, \mathbf{A}) = \{\mathbf{x} \mapsto f_b(\mathbf{x}, \mathbf{w}) : \mathbf{w} \in \mathbb{R}^W\} \quad (b = 1, \dots, t),$$

wobei die faltenden neuronalen Netze $f_1(\cdot, \mathbf{w}), \dots, f_t(\cdot, \mathbf{w}) \in \mathcal{F}(L, k, \mathbf{M}, s, \mathbf{P}, \mathbf{A})$ jeweils nur von W_{L+1}/t Variablen von $\mathbf{w}$ abhängen.

Wir können im Folgenden o.B.d.A annehmen, dass

$$\text{VCdim}(\mathcal{H}) \geq \sum_{r=1}^{L+L_{net}+2} W_r \tag{3.68}$$

gilt, da im Fall $\text{VCdim}(\mathcal{H}) < \sum_{r=1}^{L+L_{net}+2} W_r$

$$\begin{aligned} \text{VCdim}(\mathcal{H}) &< (L + L_{net} + 2) \cdot W \\ &\overset{(3.67)}{\leq} 2 \cdot (L + L_{net} + 2)^2 \cdot t \cdot M_{max}^2 \cdot k_{max}^2 \\ &\leq c_{36} \cdot L_{max}^2 \end{aligned}$$

für eine Konstante $c_{36} > 0$ gilt, die nur von M_{max} und k_{max} abhängt, und wir die Behauptung mit Lemma 18 erhalten. Um eine obere Schranke für die VC-Dimension $\text{VCdim}(\mathcal{H})$ herzuleiten, werden wir die Wachstumsfunktion $\Pi_{\text{sgn}(\mathcal{H})}(m)$ beschränken. Hierbei genügt es wegen (3.68) im Folgenden den Fall

$$m \geq W \tag{3.69}$$

zu betrachten, was uns ermöglichen wird, Lemma 19 mehrere Male anzuwenden. Um die Wachstumsfunktion $\Pi_{\text{sgn}(\mathcal{H})}(m)$ von oben zu beschränken, fixieren wir die Eingabewerte

$$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m) \in [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}} \times \mathbb{R}$$

und betrachten $h \in \mathcal{H}$ als Funktion ihres Gewichtsvektors $\mathbf{w} \in \mathbb{R}^W$:

$$h_l(\mathbf{w}) = g_{net} \circ (f_1, \ldots, f_t)(\mathbf{x}_l, \mathbf{w}) - y_l \quad (l = 1, \ldots, m).$$

Eine obere Schranke für

$$K := |\{(\text{sgn}(h_1(\mathbf{w})), \ldots, \text{sgn}(h_m(\mathbf{w}))) : \mathbf{w} \in \mathbb{R}^W\}|$$

impliziert dann auch eine obere Schranke für $\Pi_{\text{sgn}(\mathcal{H})}(m)$. Für jede Partition

$$\mathcal{S} = \{S_1, \ldots, S_M\}$$

von $\mathbb{R}^W$ gilt

$$K \leq \sum_{i=1}^{M} |\{(\text{sgn}(h_1(\mathbf{w})), \ldots, \text{sgn}(h_m(\mathbf{w})) : \mathbf{w} \in S_i\}|. \tag{3.70}$$

Wir werden eine Partition $\mathcal{S}$ von $\mathbb{R}^W$ konstruieren, sodass innerhalb jeder Teilmenge $S \in \mathcal{S}$ die Funktionen $h_l(\cdot)$ $(l = 1, \ldots, m)$ als feste Polynome von beschränktem Grad dargestellt werden können. Auf diese Weise können wir jeden Summanden von Gleichung (3.70) durch die Anwendung von Lemma 19 beschränken. Dies gelingt uns in zwei Schritten.

Im ersten Schritt konstruieren wir eine Partition $\mathcal{S}^{(1)}$ von $\mathbb{R}^W$, sodass innerhalb jeder Teilmenge $S \in \mathcal{S}^{(1)}$ die t faltenden neuronalen Netze $f_{1,l}(\mathbf{w}), \ldots, f_{t,l}(\mathbf{w})$ für alle $l \in \{1, \ldots, m\}$ als Polynome in $\mathbf{w}$ mit einem Grad von höchstens $L+1$ dargestellt werden können, wobei wir die Bezeichnung

$$f_{b,l}(\mathbf{w}) = f_b(\mathbf{x}_l, \mathbf{w}) \quad (b = 1, \ldots, t, l = 1, \ldots, m)$$

verwenden. Wegen der Definition der Subsampling Schicht $f_{sub}^{(s)}$ gilt

$$f_{b,l}(\mathbf{w}) = \max_{\substack{i\in\{A_1,\ldots,A_1'\},\\ j\in\{A_2,\ldots,A_2'\}}} \sum_{s_2=1}^{k_L} w_{s_2}^{(b)} \cdot o^{(L)}_{((i-1)\cdot s+1,(j-1)\cdot s+1),b,s_2,\mathbf{x}_l}(\mathbf{w}_{\{1,\ldots,W_L\}}) \quad (b = 1, \ldots, t, l = 1, \ldots, m),$$

für $1 \leq A_1 \leq A_1' \leq \lceil d_1/s \rceil$ und $1 \leq A_2 \leq A_2' \leq \lceil d_2/s \rceil$, wobei $o^{(L)}_{(i,j),b,s_2,\mathbf{x}}(\mathbf{w}_{\{1,\ldots,W_L\}})$ für $\mathbf{x} \in [0,1]^{\{1,\ldots,d_1\}\times\{1,\ldots,d_2\}}$ wie folgt rekursiv definiert ist:

$$\begin{aligned} &o^{(r)}_{(i,j),b,s_2,\mathbf{x}}(\mathbf{w}_{\{1,\ldots,W_r\}}) \\ &= \sigma\left(\sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2\in\{1,\ldots,M_r\}\\ i+t_1-P_r\in\{1,\ldots,d_1\}\\ j+t_2-P_r\in\{1,\ldots,d_2\}}} w^{(b,r)}_{t_1,t_2,s_1,s_2} \cdot o^{(r-1)}_{(i+t_1-P_r,j+t_2-P_r),b,s_1,\mathbf{x}}(\mathbf{w}_{\{1,\ldots,W_{r-1}\}}) + w^{(b,r)}_{s_2}\right) \end{aligned}$$

für $(i,j) \in \{1,\dots,d_1\} \times \{1,\dots,d_2\}$ und $r \in \{1,\dots,L\}$ sowie

$$o^{(0)}_{(i,j),b,1,\mathbf{x}}(\mathbf{w}_\emptyset) = x_{(i,j)}$$

für $(i,j) \in \{1,\dots,d_1\} \times \{1,\dots,d_2\}$. Zunächst konstruieren wir eine Partition $\mathcal{S}_L = \{S_1,\dots,S_M\}$ von $\mathbb{R}^W$, sodass sich innerhalb jeder Teilmenge $S \in \mathcal{S}_L$ die Funktionen

$$o^{(L)}_{(i,j),b,s_2,\mathbf{x}_l}(\mathbf{w}_{\{1,\dots,W_L\}}) \quad (l = 1,\dots,m, s_2 = 1,\dots,k_L, b = 1,\dots,t, (i,j) \in \{1,\dots,d_1\} \times \{1,\dots,d_2\})$$

als feste Polynome von Grad höchstens L in den W_L Variablen $\mathbf{w}_{\{1,\dots,W_L\}}$ von $\mathbf{w} \in S$ darstellen lassen. Wir konstruieren die Partition $\mathcal{S}_L$ Schicht für Schicht, indem wir eine Sequenz $\mathcal{S}_0,\dots,\mathcal{S}_L$ erzeugen, sodass jedes $\mathcal{S}_r$ $(r = 0,\dots,L)$ eine Partition von $\mathbb{R}^W$ mit den folgenden beiden Eigenschaften ist:

1. Es gilt $|\mathcal{S}_0| = 1$ und für alle $r \in \{1,\dots,L\}$ gilt

$$\frac{|\mathcal{S}_r|}{|\mathcal{S}_{r-1}|} \leq 2\left(\frac{2 \cdot e \cdot t \cdot k_r \cdot d_1 \cdot d_2 \cdot m \cdot r}{W_r}\right)^{W_r}. \tag{3.71}$$

2. Für alle $r \in \{0,\dots,L\}$, $S \in \mathcal{S}_r$, $(i,j) \in \{1,\dots,d_1\} \times \{1,\dots,d_2\}$, $s_2 \in \{1,\dots,k_r\}$, $l \in \{1,\dots,m\}$ und $b \in \{1,\dots,t\}$ lässt sich

   $$o^{(r)}_{(i,j),b,s_2,\mathbf{x}_l}(\mathbf{w}_{\{1,\dots,W_r\}})$$

   als ein festes Polynom von einem Grad kleiner gleich r in den W_r Variablen $\mathbf{w}_{\{1,\dots,W_r\}}$ von $\mathbf{w}$ darstellen, sofern $\mathbf{w}$ in S variiert.

Wir zeigen die Existenz der Sequenz $\mathcal{S}_0,\dots,\mathcal{S}_L$ mittels Induktion über $r \in \{0,\dots,L\}$: Wir setzen $\mathcal{S}_0 \coloneqq \{\mathbb{R}^W\}$. Da

$$o^{(0)}_{(i,j),b,1,\mathbf{x}_l}(\mathbf{w}_\emptyset) = (\mathbf{x}_l)_{(i,j)}$$

ein konstantes Polynom ist, sind die obigen Eigenschaften für $r = 0$ erfüllt. Nun nehmen wir an, es seien $\mathcal{S}_0,\dots,\mathcal{S}_{r-1}$ wie oben definiert (d.h. es gelten die beiden Eigenschaften). Ausgehend von diesen Partitionen konstruieren wir die Partition $\mathcal{S}_r$. Für $S \in \mathcal{S}_{r-1}$ sei

$$p_{(i,j),b,s_1,\mathbf{x}_l,S}(\mathbf{w}_{\{1,\dots,W_{r-1}\}})$$

die Polynomdarstellung von $o^{(r-1)}_{(i,j),b,s_1,\mathbf{x}_l}(\mathbf{w}_{\{1,\dots,W_{r-1}\}})$, wenn $\mathbf{w} \in S$. Wegen der Induktionshypothese ist

$$p_{(i,j),b,s_1,\mathbf{x}_l,S}(\mathbf{w}_{\{1,\dots,W_{r-1}\}})$$

ein Polynom mit Grad kleiner gleich $r-1$ in den W_{r-1} Variablen $\mathbf{w}_{\{1,\dots,W_{r-1}\}}$ von $\mathbf{w}$ für alle $b \in \{1,\dots,t\}$, $l \in \{1,\dots,m\}$, $(i,j) \in \{1,\dots,d_1\} \times \{1,\dots,d_2\}$ und $s_1 \in \{1,\dots,k_{r-1}\}$. Daher ist für alle $b \in \{1,\dots,t\}$, $l \in \{1,\dots,m\}$, $(i,j) \in \{1,\dots,d_1\} \times \{1,\dots,d_2\}$ und $s_2 \in \{1,\dots,k_r\}$

$$\sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2 \in \{1,\dots,M_r\} \\ i+t_1-P_r \in \{1,\dots,d_1\} \\ j+t_2-P_r \in \{1,\dots,d_2\}}} w^{(b,r)}_{t_1,t_2,s_1,s_2} \cdot p_{(i+t_1-P_r,j+t_2-P_r),b,s_1,\mathbf{x}_l,S}(\mathbf{w}_{\{1,\dots,W_{r-1}\}}) + w^{(b,r)}_{s_2}$$

ein Polynom mit Grad kleiner gleich r in den W_r Variablen $\mathbf{w}_{\{1,\dots,W_r\}}$ von $\mathbf{w}$. Wegen Bedingung (3.69) gilt $t \cdot k_r \cdot m \cdot d_1 \cdot d_2 \geq W_r$. Somit können wir Lemma 19 anwenden, womit die Menge der Polynome

$$\left\{ \sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2 \in \{1,\dots,M_r\} \\ i+t_1-P_r \in \{1,\dots,d_1\} \\ j+t_2-P_r \in \{1,\dots,d_2\}}} w_{t_1,t_2,s_1,s_2}^{(b,r)} \cdot p_{(i+t_1-P_r,j+t_2-P_r),b,s_1,\mathbf{x}_l,S}(\mathbf{w}_{\{1,\dots,W_{r-1}\}}) + w_{s_2}^{(b,r)} : \right. \tag{3.72}$$
$$\left. b \in \{1,\dots,t\}, l \in \{1,\dots,m\}, (i,j) \in \{1,\dots,d_1\} \times \{1,\dots,d_2\}, s_2 \in \{1,\dots,k_r\} \right\}$$

höchstens

$$\Pi \coloneqq 2\left(\frac{2 \cdot e \cdot t \cdot k_r \cdot m \cdot d_1 \cdot d_2 \cdot r}{W_r}\right)^{W_r}$$

verschiedene Vorzeichenmuster für $\mathbf{w} \in S$ annimmt (die Anzahl der Vorzeichenmuster einer endlichen Menge von Polynomen ist gemäß Gleichung (3.66) definiert). Deshalb können wir $S \subset \mathbb{R}^W$ in Π disjunkte Teilmengen partitionieren, sodass alle Polynome innerhalb einer dieser Teilmengen nicht deren Vorzeichen wechseln. Verfahren wir so für alle $S \in \mathcal{S}_{r-1}$, erhalten wir die geforderte Partition $\mathcal{S}_r$, indem wir alle so entstandenen disjunkten Teilmengen zu einer Partition von $\mathbb{R}^W$ zusammenfassen. Wegen der Definition von Π ist dann die erste von uns geforderte Bedingung erfüllt. Wir zeigen nun, dass auch die zweite Bedingung erfüllt ist. Fixiere ein $S' \in \mathcal{S}_r$. Aufgrund der Definition der Partition $\mathcal{S}_r$ wechseln die Polynome in (3.72) innerhalb von S' nicht deren Vorzeichen. Daher ist

$$\begin{aligned}
&o_{(i,j),b,s_2,\mathbf{x}_l}^{(r)}(\mathbf{w}_{\{1,\dots,W_r\}}) \\
&= \sigma\left(\sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2 \in \{1,\dots,M_r\} \\ i+t_1-P_r \in \{1,\dots,d_1\} \\ j+t_2-P_r \in \{1,\dots,d_2\}}} w_{t_1,t_2,s_1,s_2}^{(b,r)} \cdot o_{(i+t_1-P_r,j+t_2-P_r),b,s_1,\mathbf{x}}^{(r-1)}(\mathbf{w}_{\{1,\dots,W_{r-1}\}}) + w_{s_2}^{(b,r)} \right) \\
&= \max\left\{ \sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2 \in \{1,\dots,M_r\} \\ i+t_1-P_r \in \{1,\dots,d_1\} \\ j+t_2-P_r \in \{1,\dots,d_2\}}} w_{t_1,t_2,s_1,s_2}^{(b,r)} \cdot o_{(i+t_1-P_r,j+t_2-P_r),b,s_1,\mathbf{x}}^{(r-1)}(\mathbf{w}_{\{1,\dots,W_{r-1}\}}) + w_{s_2}^{(b,r)}, 0 \right\}
\end{aligned}$$

innerhalb von S' entweder ein Polynom mit einem Grad kleiner gleich r in den W_r Variablen $\mathbf{w}_{\{1,\dots,W_r\}}$ von $\mathbf{w}$ oder das konstante Polynom mit Wert 0 für alle $(i,j) \in \{1,\dots,d_1\} \times \{1,\dots,d_2\}$, $b \in \{1,\dots,t\}$, $s_2 \in \{1,\dots,k_r\}$ und $l \in \{1,\dots,m\}$. Daher ist auch die 2. Eigenschaft erfüllt und wir haben die gewünschte Partition $\mathcal{S}_L$ konstruiert. Wegen Ungleichung (3.71) der 1. Eigenschaft, gilt dann

$$|\mathcal{S}_L| \leq \prod_{r=1}^{L} 2\left(\frac{2 \cdot e \cdot t \cdot k_r \cdot d_1 \cdot d_2 \cdot m \cdot r}{W_r}\right)^{W_r}.$$

Für alle $(i,j) \in \{A_1,\dots,A_1'\} \times \{A_2,\dots,A_2'\}$, $b \in \{1,\dots,t\}$ und $l \in \{1,\dots,m\}$ definieren wir

$$f_{(i,j),b,\mathbf{x}_l}(\mathbf{w}_{\{1,\dots,W_{L+1}\}}) \coloneqq \sum_{s_2=1}^{k_L} w_{s_2}^{(b)} \cdot o_{((i-1)\cdot s+1,(j-1)\cdot s+1),b,s_2,\mathbf{x}_l}^{(L)}(\mathbf{w}_{\{1,\dots,W_L\}}).$$

Außerdem bezeichnen wir für alle $S \in \mathcal{S}_L$ mit $p_{(i,j),b,S,\mathbf{x}_l}(\mathbf{w}_{\{1,\dots,W_{L+1}\}})$ die Polynomdarstellung von $f_{(i,j),b,\mathbf{x}_l}(\mathbf{w}_{\{1,\dots,W_{L+1}\}})$, wenn $\mathbf{w} \in S$ gilt. Wegen der Konstruktion von $\mathcal{S}_L$ ist dies ein Polynom mit Grad kleiner gleich $L+1$ in den W_{L+1} Variablen $\mathbf{w}_{\{1,\dots,W_{L+1}\}}$ von $\mathbf{w}$. Aufgrund von Bedingung (3.69) können wir Lemma 19 anwenden, womit die Menge der Polynome

$$\begin{aligned}&\Big\{p_{(i_1,j_1),b,S,\mathbf{x}_l}(\mathbf{w}_{I^{(L+1)}}) - p_{(i_2,j_2),b,S,\mathbf{x}_l}(\mathbf{w}_{I^{(L+1)}}) : \\ &\quad (i_1,j_1),(i_2,j_2) \in \{A_1,\dots,A_1'\}\times\{A_2,\dots,A_2'\}, (i_1,j_1)\neq(i_2,j_2), b\in\{1,\dots,t\}, l\in\{1,\dots,m\}\Big\}\end{aligned} \tag{3.73}$$

höchstens

$$\Delta \coloneqq 2\left(\frac{2\cdot e\cdot t\cdot d_1^2\cdot d_2^2\cdot m\cdot(L+1)}{W_{L+1}}\right)^{W_{L+1}}$$

verschiedene Vorzeichenmuster für $\mathbf{w}\in S$ annimmt. Deswegen können wir $S\subset\mathbb{R}^W$ in Δ disjunkte Teilmengen partitionieren, sodass alle Polynome innerhalb einer dieser Teilmengen nicht ihre Vorzeichen wechseln. Verfahren wir so für alle $S\in\mathcal{S}_L$, erhalten wir die geforderte Partition $\mathcal{S}^{(1)}$, indem wir alle so entstandenen disjunkten Teilmengen zu einer Partition zusammenfassen. Für die Größe der Partition $\mathcal{S}^{(1)}$ gilt dann

$$|\mathcal{S}^{(1)}| \leq \prod_{r=1}^{L} 2\cdot\left(\frac{2\cdot t\cdot e\cdot k_r\cdot d_1\cdot d_2\cdot m\cdot r}{W_r}\right)^{W_r}\cdot 2\cdot\left(\frac{2\cdot e\cdot t\cdot d_1^2\cdot d_2^2\cdot m\cdot(L+1)}{W_{L+1}}\right)^{W_{L+1}}.$$

Um die geforderte Eigenschaft der Partition $\mathcal{S}^{(1)}$ nachzuweisen, fixieren wir ein $S'\in\mathcal{S}^{(1)}$. Wegen der Definition der Partition $\mathcal{S}^{(1)}$ wechseln die Polynome (3.73) innerhalb von S' nicht deren Vorzeichen. Daher existiert für alle $b\in\{1,\dots,t\}$ und $l\in\{1,\dots,m\}$ eine Permutation $\pi^{(b,l)}$ der Menge

$$\{A_1,\dots,A_1'\}\times\{A_2,\dots,A_2'\},$$

sodass

$$f_{\pi^{(b,l)}((A_1,A_2)),b,\mathbf{x}_l}(\mathbf{w}_{\{1,\dots,W_{L+1}\}}) \geq \dots \geq f_{\pi^{(b,l)}((A_1',A_2')),b,\mathbf{x}_l}(\mathbf{w}_{\{1,\dots,W_{L+1}\}})$$

für alle $\mathbf{w}\in S'$, $l\in\{1,\dots,m\}$ und $b\in\{1,\dots,t\}$. Es gilt dann

$$\begin{aligned}f_{b,l}(\mathbf{w}) &= \max\left\{f_{(A_1,A_2),b,\mathbf{x}_l}\left(\mathbf{w}_{\{1,\dots,W_{L+1}\}}\right),\dots,f_{(A_1',A_2'),b,\mathbf{x}_l}\left(\mathbf{w}_{\{1,\dots,W_{L+1}\}}\right)\right\}\\ &= f_{\pi^{(b,l)}((A_1,A_2)),b,\mathbf{x}_l}(\mathbf{w}_{\{1,\dots,W_{L+1}\}})\end{aligned}$$

für alle $\mathbf{w}\in S'$. Da $f_{\pi^{(b,l)}((A_1,A_2)),b,\mathbf{x}_l}(\mathbf{w}_{\{1,\dots,W_{L+1}\}})$ innerhalb von S' einem Polynom entspricht, ist auch $f_{b,l}(\mathbf{w})$ ein Polynom innerhalb von S' mit einem Grad kleiner gleich $L+1$ in den W_{L+1} Variablen $\mathbf{w}_{\{1,\dots,W_{L+1}\}}$ von $\mathbf{w}\in\mathbb{R}^W$.

Im zweiten Schritt konstruieren wir ausgehend von der Partition $\mathcal{S}^{(1)}$ eine Partition $\mathcal{S}$, sodass sich die Funktionen $h_l(\cdot)$ $(l=1,\dots,m)$ innerhalb jeder Teilmenge als Polynome von Grad kleiner gleich $L+L_{net}+2$ darstellen lassen. Es gilt

$$h_l(\mathbf{w}) = \sum_{i=1}^{k_{L+L_{net}+1}} w_i^{(L_{net})}\cdot g_{i,l}^{(L_{net})}\left(\mathbf{w}_{\{1,\dots,W_{L+L_{net}+1}\}}\right) + w_0^{(L_{net})} - y_l,$$

wobei die Funktionen $g_{i,l}^{(L_{net})}$ rekursiv definiert sind durch

$$g_{i,l}^{(r)}\left(\mathbf{w}_{\{1,\dots,W_{L+r+1}\}}\right) = \sigma\left(\sum_{j=1}^{k_{L+r}} w_{i,j}^{(r-1)} g_{j,l}^{(r-1)}(\mathbf{w}_{\{1,\dots,W_{L+r}\}}) + w_{i,0}^{(r-1)}\right)$$

für $i \in \{1, \dots, k_{L+r+1}\}$ und $r \in \{1, \dots, L_{net}\}$ und

$$g_{i,l}^{(0)}(\mathbf{w}_{\{1,\dots,W_{L+1}\}}) = f_{i,l}(\mathbf{w})$$

für $i \in \{1, \dots, k_{L+1}\}$ ($k_{L+1} = t$). Wie oben konstruieren wir die Partition $\mathcal{S}$ Schicht für Schicht, indem wir eine Sequenz von Partitionen $\mathcal{S}_0, \dots, \mathcal{S}_{L_{net}}$ von $\mathbb{R}^W$ konstruieren, welche die folgenden beiden Eigenschaften besitzt:

1. Wir setzen $\mathcal{S}_0 = \mathcal{S}^{(1)}$ und für $r \in \{1, \dots, L_{net}\}$ gilt

$$\frac{|\mathcal{S}_r|}{|\mathcal{S}_{r-1}|} \leq 2 \left(\frac{2 \cdot e \cdot k_{L+r+1} \cdot m \cdot (L+r+1)}{W_{L+r+1}} \right)^{W_{L+r+1}}.$$

2. Für alle $r \in \{0, \dots, L_{net}\}$, $S \in \mathcal{S}_r$, $i \in \{1, \dots, k_{L+r+1}\}$ und $l \in \{1, \dots, m\}$ lässt sich

$$g_{i,l}^{(r)}(\mathbf{w}_{\{1,\dots,W_{L+r+1}\}})$$

als ein festes Polynom von einem Grad kleiner gleich $L+r+1$ in den W_{L+r+1} Variablen $\mathbf{w}_{\{1,\dots,W_{L+r+1}\}}$ von $\mathbf{w}$ darstellen, sofern $\mathbf{w}$ in S variiert.

Analog zu Schritt 1 zeigen wir die Existenz der Sequenz $\mathcal{S}_0, \dots, \mathcal{S}_{L_{net}}$ mittels Induktion über $r \in \{0, \dots, L_{net}\}$: Wie wir bereits in Schritt 1 gezeigt haben, ist die 2. obige Eigenschaft für $r = 0$ erfüllt. Wir nehmen nun an, die Sequenz $\mathcal{S}_0, \dots, \mathcal{S}_{r-1}$ sei definiert und erfülle die obigen Eigenschaften. Ausgehend von diesen Partitionen konstruieren wir im Folgenden die Partition $\mathcal{S}_r$. Für $S \in \mathcal{S}_{r-1}$, $l \in \{1, \dots, m\}$ und $j \in \{1, \dots, k_{L+r}\}$ sei $p_{j,l,S}(\mathbf{w}_{\{1,\dots,W_{L+r}\}})$ die Polynomdarstellung von $g_{j,l}^{(r-1)}(\mathbf{w}_{\{1,\dots,W_{L+r}\}})$ für $\mathbf{w} \in S$. Aufgrund der Induktionshypothese ist $p_{j,l,S}(\mathbf{w}_{\{1,\dots,W_{L+r}\}})$ ein Polynom mit Grad kleiner gleich $L+r$ in den W_{L+r} Variablen $\mathbf{w}_{\{1,\dots,W_{L+r}\}}$ von $\mathbf{w}$. Daher ist für alle $l \in \{1, \dots, m\}$ und $i \in \{1, \dots, k_{L+r+1}\}$ auch

$$\sum_{j=1}^{k_{L+r}} w_{(i,j)}^{(r-1)} \cdot p_{j,l,S}(\mathbf{w}_{\{1,\dots,W_{L+r}\}}) + w_{i,0}^{(r-1)}$$

ein Polynom mit Grad kleiner gleich $L+r+1$ in den W_{L+r+1} Variablen $\mathbf{w}_{\{1,\dots,W_{L+r+1}\}}$ von $\mathbf{w}$. Wegen Bedingung (3.69) gilt $k_{L+r+1} \cdot m \geq W_{L+r+1}$. Somit können wir Lemma 19 anwenden, womit die Menge der Polynome

$$\left\{ \sum_{j=1}^{k_{L+r}} w_{(i,j)}^{(r-1)} \cdot p_{j,l,S}(\mathbf{w}_{\{1,\dots,W_{L+r}\}}) + w_{i,0}^{(r-1)} : l \in \{1, \dots, m\}, i \in \{1, \dots, k_{L+r+1}\} \right\} \tag{3.74}$$

höchstens

$$\Pi \coloneqq 2 \left(\frac{2 \cdot e \cdot k_{L+r+1} \cdot m \cdot (L+r+1)}{W_{L+r+1}} \right)^{W_{L+r+1}}$$

verschiedene Vorzeichenmuster für $\mathbf{w} \in S$ annimmt. Deshalb können wir $S \subset \mathbb{R}^W$ in Π disjunkte Teilmengen partitionieren, sodass alle Polynome innerhalb einer dieser Teilmengen nicht ihre Vorzeichen wechseln. Verfahren wir so für alle $S \in \mathcal{S}_{r-1}$, erhalten wir die geforderte Partition $\mathcal{S}_r$, indem wir alle so entstandenen disjunkten Teilmengen zu einer Partition zusammenfassen. Wegen der Definition von Π ist dann die erste von uns geforderte Bedingung erfüllt. Auch die zweite Bedingung lässt sich analog zu Schritt 1 verifizieren: Fixiere ein $S' \in \mathcal{S}_r$. Aufgrund der Definition der Partition $\mathcal{S}_r$ wechseln die Polynome in (3.74) innerhalb von S' nicht deren Vorzeichen. Daher ist

$$g_{i,l}^{(r)}\left(\mathbf{w}_{\{1,\dots,W_{L+r+1}\}}\right) = \sigma\left(\sum_{j=1}^{k_{L+r}} w_{i,j}^{(r-1)} g_{j,l}^{(r-1)}(\mathbf{w}_{\{1,\dots,W_{L+r}\}}) + w_{i,0}^{(r-1)} \right)$$

$$= \max\left\{\sum_{j=1}^{k_{L+r}} w_{i,j}^{(r-1)} g_{j,l}^{(r-1)}(\mathbf{w}_{\{1,\dots,W_{L+r}\}}) + w_{i,0}^{(r-1)}, 0\right\}$$

innerhalb von S' entweder ein Polynom mit einem Grad kleiner gleich $L+r+1$ in den W_{L+r+1} Variablen $\mathbf{w}_{\{1,\dots,W_{L+r+1}\}}$ von $\mathbf{w}$ oder das konstante Polynom mit Wert 0 für alle $i \in \{1,\dots,k_{L+r+1}\}$ und $l \in \{1,\dots,m\}$. Daher ist auch die 2. Eigenschaft erfüllt und wir haben die gewünschte Partition $\mathcal{S}_{L_{net}}$ konstruiert. Daher ist die Funktion

$$h_l(\mathbf{w}) = \sum_{i=1}^{k_{L+L_{net}+1}} w_i^{(L)} \cdot g_{i,k}^{(L_{net})}\left(\mathbf{w}_{\{1,\dots,W_{L+L_{net}+1}\}}\right) + w_0^{(L)} - y_l$$

innerhalb von $S \in \mathcal{S} \coloneqq \mathcal{S}_{L_{net}}$ ein Polynom mit einem Grad kleiner gleich $L+L_{net}+2$ in den W Variablen von $\mathbf{w} \in \mathbb{R}^W$ für alle $l \in \{1,\dots,m\}$. Für die Größe der Partition $\mathcal{S}$ gilt

$$\begin{aligned}
|\mathcal{S}| \leq & \prod_{r=1}^{L} 2 \cdot \left(\frac{2 \cdot e \cdot t \cdot k_r \cdot d_1 \cdot d_2 \cdot m \cdot r}{W_r}\right)^{W_r} \cdot 2 \cdot \left(\frac{2 \cdot e \cdot d_1^2 \cdot d_2^2 \cdot m \cdot (L+1)}{W_{L+1}}\right)^{W_{L+1}} \\
& \cdot \prod_{r=1}^{L_{net}} 2 \cdot \left(\frac{2 \cdot e \cdot k_{L+r+1} \cdot m \cdot (L+r+1)}{W_{L+r+1}}\right)^{W_{L+r+1}} \\
\leq & \prod_{r=1}^{L+L_{net}+1} 2 \cdot \left(\frac{2 \cdot e \cdot t \cdot k_r \cdot d_1^2 \cdot d_2^2 \cdot m \cdot r}{W_r}\right)^{W_r}.
\end{aligned}$$

Wegen Bedingung (3.69) und einer weiteren Anwendung von Lemma 19 gilt

$$|\{(\operatorname{sgn}(h_1(\mathbf{w})),\dots,\operatorname{sgn}(h_m(\mathbf{w}))) : \mathbf{w} \in S'\}| \leq 2 \cdot \left(\frac{2 \cdot e \cdot m \cdot (L+L_{net}+2)}{W}\right)^W$$

für alle $S' \in \mathcal{S}$. Nun können wir K unter Verwendung von Gleichung (3.70) abschätzen und erhalten die folgende Abschätzung für die Wachstumsfunktion:

$$\begin{aligned}
\Pi_{\operatorname{sgn}(\mathcal{H})}(m) \leq & \prod_{r=1}^{L+L_{net}+2} 2 \cdot \left(\frac{2 \cdot e \cdot t \cdot k_r \cdot d_1^2 \cdot d_2^2 \cdot r \cdot m}{W_r}\right)^{W_r} \\
\leq & 2^{L+L_{net}+2} \cdot \left(\frac{\sum_{r=1}^{L+L_{net}+2} 2 \cdot e \cdot t \cdot k_r \cdot d_1^2 \cdot d_2^2 \cdot r \cdot m}{\sum_{r=1}^{L+L_{net}+2} W_r}\right)^{\sum_{r=1}^{L+L_{net}+2} W_r} \\
= & 2^{L+L_{net}+2} \cdot \left(\frac{R \cdot m}{\sum_{r=1}^{L+L_{net}+2} W_r}\right)^{\sum_{r=1}^{L+L_{net}+2} W_r}, \qquad (3.75)
\end{aligned}$$

wobei $R \coloneqq 2 \cdot e \cdot t \cdot d_1^2 \cdot d_2^2 \cdot \sum_{r=1}^{L+L_{net}+2} k_r \cdot r$. In der zweiten Zeile haben wir die gewichtete AM-GM Ungleichung verwendet (siehe Ungleichung (A.15) und Lemma 24 aus Abschnitt 3 des Anhangs). O.B.d.A können wir annehmen, dass $\operatorname{VCdim}(\mathcal{H}) \geq \sum_{r=1}^{L+L_{net}+2} W_r$ gilt, da im Fall $\operatorname{VCdim}(\mathcal{H}) < \sum_{r=1}^{L+L_{net}+2} W_r$

$$\begin{aligned}
\operatorname{VCdim}(\mathcal{H}) & < (L+L_{net}+2) \cdot W \\
& \overset{(3.67)}{\leq} 2 \cdot (L+L_{net}+2)^2 \cdot t \cdot M_{max}^2 \cdot k_{max}^2 \\
& \leq c_{37} \cdot L_{max}^2
\end{aligned}$$

für eine Konstante $c_{37} > 0$, welche nur von M_{max} und k_{max} abhängt, und erhalten die Behauptung mit Lemma 18. Daher erhalten wir mit der Definition der VC–Dimension und Ungleichung (3.75) (welche nur für $m \geq W$ gilt)

$$2^{\text{VCdim}(\mathcal{H})} = \Pi_{\text{sgn}(\mathcal{H})}(\text{VCdim}(\mathcal{H})) \leq 2^{L+L_{net}+2} \cdot \left(\frac{R \cdot \text{VCdim}(\mathcal{H})}{\sum_{r=1}^{L+L_{net}+2} W_r} \right)^{\sum_{r=1}^{L+L_{net}+2} W_r}.$$

Da

$$R \geq 2 \cdot e \cdot t \cdot d_1^2 \cdot d_2^2 \cdot \sum_{r=1}^{1+1+2} r \geq 2 \cdot e \cdot t \cdot d_1^2 \cdot d_2^2 \cdot 10 \geq 16,$$

impliziert Lemma 21 unten (mit den Parametern R, $m = \text{VCdim}(\mathcal{H})$, $w = \sum_{r=1}^{L+L_{net}+2} W_r$ und $L' = L+L_{net}+2$)

$$\begin{aligned}
\text{VCdim}(\mathcal{H}) &\leq (L + L_{net} + 2) + \left(\sum_{r=1}^{L+L_{net}+2} W_r \right) \cdot \log_2(2 \cdot R \cdot \log_2(R)) \\
&\leq (L + L_{net} + 2) + (L + L_{net} + 2) \cdot W \\
&\qquad \cdot \log_2(2 \cdot (2 \cdot e \cdot t \cdot d_1^2 \cdot d_2^2 \cdot (L + L_{net} + 2) \cdot k_{max})^2) \\
&\leq 2 \cdot (L + L_{net} + 2) \cdot W \cdot \log_2 \left((2 \cdot e \cdot t \cdot (L + L_{net} + 2) \cdot k_{max} \cdot d_1 \cdot d_2)^4 \right) \\
&\overset{(3.67)}{\leq} 16 \cdot t \cdot (L + L_{net} + 2)^2 \cdot k_{\text{max}}^2 \cdot M_{\text{max}}^2 \\
&\qquad \cdot \log_2 \left(2 \cdot e \cdot t \cdot (L + L_{net} + 2) \cdot k_{max} \cdot d_1 \cdot d_2 \right) \\
&\leq c_{38} \cdot L_{max}^2 \cdot \log_2(L_{max} \cdot d_1 \cdot d_2),
\end{aligned}$$

für eine Konstante $c_{38} > 0$, welche nur von k_{max} und M_{max} abhängt. In der dritten Zeile haben wir Gleichung (3.67) für die Gesamtanzahl der Gewichte W verwendet. Mit Lemma 18 folgt

$$V_{\mathcal{F}^+} \leq c_{38} \cdot L_{max}^2 \cdot \log_2(L_{max} \cdot d_1 \cdot d_2).$$

□

Lemma 21. *Es gelte $2^m \leq 2^{L'} \cdot (m \cdot R/w)^w$ für $R \geq 16$ und $m \geq w \geq L' \geq 0$. Dann gilt*

$$m \leq L' + w \cdot \log_2(2 \cdot R \cdot \log_2(R)).$$

Beweis. Siehe Lemma 16 in Bartlett et al. (2019) □

Beweis von Lemma 15. Unter Verwendung von Lemma 20 und

$$\mathcal{V}_{T_{c_{34} \cdot \log n} \mathcal{F}_6(\boldsymbol{\theta})^+} \leq \mathcal{V}_{\mathcal{F}_6(\boldsymbol{\theta})^+}$$

können wir zusammen mit Lemma 16 und Lemma 17 folgern, dass

$$\begin{aligned}
&\mathcal{N}_1 \left(\epsilon, T_{c_{34} \cdot \log n} \mathcal{F}_6(\boldsymbol{\theta}), \mathbf{x}_1^n \right) \\
&\leq 3 \cdot \left(\frac{4e \cdot c_{34} \cdot \log n}{\epsilon} \cdot \log \frac{6e \cdot c_{34} \cdot \log n}{\epsilon} \right)^{\mathcal{V}_{T_{c_{34} \cdot \log n} \mathcal{F}_6(\boldsymbol{\theta})^+}} \\
&\leq 3 \cdot \left(\frac{6e \cdot c_{34} \cdot \log n}{\epsilon} \right)^{2 \cdot c_{36} \cdot L_{max}^2 \cdot \log(L_{max} \cdot d_1 \cdot d_2)}.
\end{aligned}$$

□

3.6. Beweise der Hauptresultate

Dieser Abschnitt enthält die Beweise der drei Hauptresultate Theorem 3.1, Theorem 3.2 und Theorem 3.3.

3.6.1. Beweis von Theorem 3.1

Es sei $c_{39} > 0$ hinreichend groß, sodass $c_{39} \cdot \log n \geq 2$ (dies benötigen wir insbesondere für die Anwendung von Lemma 15, um $\mathcal{N}_1(\epsilon, T_{c_{39}\cdot\log n}(\mathcal{F}_1(\boldsymbol{\theta})), \mathbf{x}_1^n)$ für ein $\epsilon \in (0,1)$ abzuschätzen). Dann gilt $z \geq 1/2$ genau dann, wenn $T_{c_{39}\cdot\log n} z \geq 1/2$ gilt, woraus

$$f_n(\mathbf{x}) = \begin{cases} 1, & \text{falls } T_{c_{39}\cdot\log n}\eta_n(\mathbf{x}) \geq \frac{1}{2} \\ 0, & \text{sonst.} \end{cases}$$

folgt. Daher impliziert Ungleichung (1.5), dass es genügt,

$$\mathbf{E}\int |T_{c_{39}\cdot\log n}\eta_n(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_\mathbf{X}(d\mathbf{x}) \leq c_{40}\cdot\log(d_1\cdot d_2)\cdot(\log n)^4 \cdot \max\left\{n^{-\frac{2\cdot p_1}{2\cdot p_1+4}}, n^{-\frac{2\cdot p_2}{2\cdot p_2+d^*}}\right\}$$

zu zeigen:

Unter Verwendung von Lemma 1 folgt

$$\begin{aligned} &\mathbf{E}\int |T_{c_{39}\cdot\log n}\eta_n(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_\mathbf{X}(d\mathbf{x}) \\ &\leq \frac{c_{41}\cdot(\log n)^2 \cdot \sup_{\mathbf{x}_1^n}\left(\log\left(\mathcal{N}_1\left(\frac{1}{n\cdot c_{39}\cdot\log n}, T_{c_{39}\cdot\log n}\mathcal{F}, \mathbf{x}_1^n\right)\right)+1\right)}{n} \\ &\quad +2\cdot\inf_{f\in\mathcal{F}}\int |f(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_\mathbf{X}(d\mathbf{x}), \end{aligned}$$

wobei $\mathcal{F} = \mathcal{F}_1(\boldsymbol{\theta})$. Die Schranke an die Überdeckungszahl aus Lemma 15 liefert zusammen mit Bemerkung 3.9

$$\begin{aligned} &\frac{c_{41}\cdot(\log n)^2 \cdot \sup_{\mathbf{x}_1^n}\left(\log\left(\mathcal{N}_1\left(\frac{1}{n\cdot c_{39}\cdot\log n}, T_{c_{39}\cdot\log n}\mathcal{F}, \mathbf{x}_1^n\right)\right)+1\right)}{n} \\ &\leq c_{42}\cdot\frac{\log(d_1\cdot d_2)\cdot(\log n)^3\cdot L_{max}^2\cdot\log L_{max}}{n} \\ &\leq c_{43}\cdot\log(d_1\cdot d_2)\cdot(\log n)^4\cdot\max\left\{n^{-\frac{2\cdot p_1}{2\cdot p_1+4}}, n^{-\frac{2\cdot p_2}{2\cdot p_2+d^*}}\right\}, \end{aligned}$$

wobei $L_{max} = \max\{L, L_{net}\}$. Als Nächstes schätzen wir den Approximationsfehler

$$\inf_{f\in\mathcal{F}}\int |f(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_\mathbf{X}(d\mathbf{x})$$

ab. Wegen den Annahmen an η gilt

$$\eta(\mathbf{x}) = g(m_1(\mathbf{x}), \ldots, m_{d^*}(\mathbf{x})),$$

sodass die Funktionen m_a $(a = 1, \ldots, d^*)$ einem Max-Pooling Modell mit Indexmenge

$$I = \{0, \ldots, 2^l - 1\} \times \{0, \ldots, 2^l - 1\}$$

und einem hierarchischen Modell vom Level l mit Funktionen

$$g_{m,s}^{(a)} : \mathbb{R}^4 \to [0,1] \quad (m = 1, \dots, l, s = 1, \dots, 4^{l-m})$$

genügen. Für $a \in \{1, \dots, d^*\}$, $m \in \{1, \dots, l\}$ und $s \in \{1, \dots, 4^{l-m}\}$ seien $\bar{g}_{m,s}^{(a)} \in \mathcal{G}_4(L_n, r_{net})$ und $\bar{g} \in \mathcal{G}_{d^*}(L_n, r_{net})$ die neuronalen Netze aus Lemma 2, für die gilt

$$\|g_{m,s}^{(a)} - \bar{g}_{m,s}^{(a)}\|_{[-2,2]^4,\infty} \leq c_{44} \cdot L_n^{-\frac{2 \cdot p_1}{4}} \leq c_{45} \cdot n^{-\frac{p_1}{2 \cdot p_1 + 4}} \tag{3.76}$$

und

$$\|g - \bar{g}_{net}\|_{[-2,2]^{d^*},\infty} \leq c_{44} \cdot L_n^{-\frac{2 \cdot p_2}{d^*}} \leq c_{45} \cdot n^{-\frac{p_2}{2 \cdot p_2 + d^*}}. \tag{3.77}$$

Wegen Bemerkung 2.4 liefert uns Lemma 4 (mit $n_1 = \cdots = n_{l-1} = 1$) faltende neuronale Netze

$$\bar{m}_1, \dots, \bar{m}_{d^*} \in \mathcal{F}_3((l, k, \bar{\mathbf{M}}, z, \mathbf{s}, \mathbf{A}))$$

mit

$$\bar{\mathbf{M}} = (2^1, 2^2, \dots, 2^l), \quad z = 4^{l-1}(L_n + 1) \quad \text{und} \quad \mathbf{s} = (\underbrace{1, 1, \dots, 1}_{l-1 \text{ mal}}),$$

sodass

$$\bar{m}_a(\mathbf{x}) = \max_{(i,j) \in \mathbb{Z}^2 : (i,j)+I \subset \{1,\dots,d_1\} \times \{1,\dots,d_2\}} \bar{f}^{(a)}(x_{(i,j)+I}) \quad (a = 1, \dots, d^*),$$

wobei

$$\bar{f}^{(a)} = \bar{f}_{l,1}^{(a)} \quad (a = 1, \dots, d^*)$$

für Funktionen $\bar{f}_{m,s}^{(a)} : [0,1]^{\{1,\dots,2^m\} \times \{1,\dots,2^m\}} \to \mathbb{R}$, welche rekursiv definiert sind durch

$$\begin{aligned} \bar{f}_{m,s}^{(a)}(\mathbf{x}) \quad = \quad & \bar{g}_{m,s}^{(a)}\big(\bar{f}_{m-1,4\cdot(s-1)+1}^{(a)}(\mathbf{x}_{\{1,\dots,2^{m-1}\} \times \{1,\dots,2^{m-1}\}}), \\ & \bar{f}_{m-1,4\cdot(s-1)+2}^{(a)}(\mathbf{x}_{\{2^{m-1}+1,\dots,2^m\} \times \{1,\dots,2^{m-1}\}}), \\ & \bar{f}_{m-1,4\cdot(s-1)+3}^{(a)}(\mathbf{x}_{\{1,\dots,2^{m-1}\} \times \{2^{m-1}+1,\dots,2^m\}}), \\ & \bar{f}_{m-1,4\cdot s}^{(a)}(\mathbf{x}_{\{2^{m-1}+1,\dots,2^m\} \times \{2^{m-1}+1,\dots,2^m\}})\big) \end{aligned}$$

für $a = 1, \dots, d^*$, $m = 2, \dots, l$ und $s = 1, \dots, 4^{l-m}$ sowie

$$\bar{f}_{1,s}^{(a)}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}) = \bar{g}_{1,s}^{(a)}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2})$$

für $a = 1, \dots, d^*$ und $s = 1, \dots, 4^{l-1}$. Wegen der Definition der Netzwerkparameter gilt

$$\bar{g} \circ (\bar{m}_1, \dots, \bar{m}_{d^*}) \in \mathcal{F}. \tag{3.78}$$

Da die Funktionen $g_{m,s}^{(a)}$ $[0,1]$–wertig sind, ist es wegen den Ungleichungen (3.76) und (3.77) möglich, die Konstante c_{10} in der Defintion von L_n hinreichend groß zu wählen, sodass mit der Dreiecksungleichung gilt

$$\|\bar{g}_{m,s}^{(a)}\|_{[-2,2]^4,\infty} \leq 2 \quad (a = 1, \dots, d^*, m = 1, \dots, l, s = 1, \dots, 4^{l-m}).$$

Zusammen mit der Definition der Funktionen $\bar{f}_{m,s}^{(a)}$, impliziert dies

$$|\bar{m}_a(\mathbf{x})| \leq 2 \quad \big(\mathbf{x} \in [0,1]^{\{1,\dots,d_1\} \times \{1,\dots,d_2\}}, a = 1, \dots, d^*\big).$$

Außerdem ist wegen der (p_2, C_2)-Glattheit der Funktion g mit $p_2 \in [1,\infty)$ die Einschränkung $g\big|_{[-2,2]^{d^*}}$ lipschitzstetig für eine Lipschitzkonstante $C > 0$. Die Dreiecksungleichung, die Lipschitzstetigkeit von g und Anwendung von Lemma 3 liefern dann

$$
\begin{aligned}
&\inf_{f\in\mathcal{F}} \int |f(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{\mathbf{X}}(d\mathbf{x}) \\
&\overset{(3.78)}{\leq} \int |\bar{g}(\bar{m}_1(\mathbf{x}),\dots,\bar{m}_{d^*}(\mathbf{x})) - g(m_1(\mathbf{x}),\dots,m_{d^*}(\mathbf{x}))|^2 \mathbf{P}_{\mathbf{X}}(d\mathbf{x}) \\
&= \int \big(|g(m_1(\mathbf{x}),\dots,m_{d^*}(\mathbf{x})) - g(\bar{m}_1(\mathbf{x}),\dots,\bar{m}_{d^*}(\mathbf{x}))| \\
&\quad + |g(\bar{m}_1(\mathbf{x}),\dots,\bar{m}_{d^*}(\mathbf{x})) - \bar{g}(\bar{m}_1(\mathbf{x}),\dots,\bar{m}_{d^*}(\mathbf{x}))|\big)^2 \mathbf{P}_{\mathbf{X}}(d\mathbf{x}) \\
&\leq \int \left(C \cdot \left(|m_1(\mathbf{x}) - \bar{m}_1(\mathbf{x})|^2 + \dots + |m_{d^*}(\mathbf{x}) - \bar{m}_{d^*}(\mathbf{x})|^2\right)^{1/2} + \|g - \bar{g}\|_{[-2,2]^{d^*},\infty}\right)^2 \mathbf{P}_{\mathbf{X}}(d\mathbf{x}) \\
&\leq \left(\sqrt{d^*} \cdot C \cdot (2C+1)^{l-1} \cdot \max_{a\in\{1,\dots,d^*\}, j\in\{1,\dots,l\}, s\in\{1,\dots,4^{l-j}\}} \|g_{j,s}^{(a)} - \bar{g}_{j,s}^{(a)}\|_{[-2,2]^4,\infty} + \|g - \bar{g}\|_{[-2,2]^{d^*},\infty}\right)^2 \\
&\leq \left(\sqrt{d^*} \cdot (2C+1)^{l} \cdot \max_{\substack{a\in\{1,\dots,d^*\},\\ j\in\{1,\dots,l\}, s\in\{1,\dots,4^{l-j}\}}} \left\{\|g_{j,s}^{(a)} - \bar{g}_{j,s}^{(a)}\|_{[-2,2]^4,\infty}, \|g - \bar{g}\|_{[-2,2]^{d^*},\infty}\right\}\right)^2 \\
&\overset{(3.76),(3.77)}{\leq} c_{46} \cdot \max\left\{n^{-\frac{2\cdot p_1}{2\cdot p_1+4}}, n^{-\frac{2\cdot p_2}{2\cdot p_2+d^*}}\right\}.
\end{aligned}
$$

Durch Zusammenfassen der obigen Resultate ergibt sich die Behauptung. □

3.6.2. Beweis von Theorem 3.2

Wie im Beweis von Theorem 3.1 sei $c_{47} > 0$ eine Konstante, sodass $c_{47} \cdot \log n \geq 2$. Dann gilt $z \geq 1/2$ genau dann, wenn $T_{c_{47}\cdot\log n} z \geq 1/2$ gilt, woraus

$$
f_n^{(j)}(\mathbf{x}) = \begin{cases} 1, & \text{falls } T_{c_{47}\cdot\log n}\eta_n^{(j)}(\mathbf{x}) \geq \frac{1}{2} \\ 0, & \text{sonst} \end{cases} \qquad (j = 3,4,5)
$$

folgt. Daher impliziert Ungleichung (1.5), dass es genügt,

$$
\mathbf{E}\int |T_{c_{47}\cdot\log n}\eta_n^{(j)}(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_X(d\mathbf{x}) \leq c_{48} \cdot \log(d_1 \cdot d_2) \cdot (\log n)^4 \cdot n^{-\frac{2p}{2p+4}} \qquad (j = 3,4,5)
$$

für eine Konstante $c_{48} > 0$ zu zeigen:

Mit Lemma 7 folgt

$$
\mathcal{F}_3(\boldsymbol{\theta}_3) \subset \mathcal{F}_4(\boldsymbol{\theta}_4) \subset \mathcal{F}_5(\boldsymbol{\theta}_5)
$$

und da die Überdeckungszahl (siehe Gleichung (3.61)) und das Infimum monoton sind, erhalten wir zusammen mit Lemma 1

$$
\begin{aligned}
&\mathbf{E}\int |T_{c_{47}\cdot\log n}\eta_n^{(j)}(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{\mathbf{X}}(d\mathbf{x}) \\
&\leq \frac{c_{49} \cdot (\log(n))^2 \cdot \sup_{\mathbf{x}_1^n}\left(\log\left(\mathcal{N}_1\left(\frac{1}{n\cdot c_{47}\log(n)}, T_{c_{47}\log(n)}\mathcal{F}_5(\boldsymbol{\theta}_5), \mathbf{x}_1^n\right)\right) + 1\right)}{n} \\
&\quad + 2 \cdot \inf_{f\in\mathcal{F}_3(\boldsymbol{\theta}_3)} \int |f(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{\mathbf{X}}(d\mathbf{x})
\end{aligned}
$$

für $j \in \{3,4,5\}$. Die Schranke an die Überdeckungszahl aus Lemma 15 liefert zusammen mit Bemerkung 3.9

$$\begin{aligned}
&\frac{c_{50} \cdot (\log(n))^2 \cdot \sup_{\mathbf{x}_1^n} \left(\log\left(\mathcal{N}_1\left(\frac{1}{n \cdot c_{47} \log(n)}, T_{c_{47}\log(n)} \mathcal{F}_5(\boldsymbol{\theta}_5), \mathbf{x}_1^n \right) \right) + 1 \right)}{n} \\
&\leq c_{51} \cdot \frac{\log(d_1 \cdot d_2) \cdot (\log n)^3 \cdot z^2 \cdot \log z}{n} \\
&\leq c_{52} \cdot \log(d_1 \cdot d_2) \cdot (\log n)^4 \cdot n^{-\frac{2 \cdot p}{2 \cdot p + 4}}.
\end{aligned}$$

Als Nächstes beschränken wir den Approximationsfehler

$$\inf_{f \in \mathcal{F}_3(\boldsymbol{\theta}_3)} \int |f(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{\mathbf{X}}(d\mathbf{x}).$$

Für $m \in \{1, \ldots, l\}$ und $s \in \{1, \ldots, b_m\}$ seien $g_{net,m,s} \in \mathcal{G}_4(L_n, c_{15})$ die neuronalen Netze aus Lemma 2 für die gilt

$$\|g_{m,s} - g_{net,m,s}\|_{[-2,2]^4,\infty} \leq c_{53} \cdot L_n^{-\frac{2p}{4}}.$$

Da die Funktionen $g_{m,s}$ $[0,1]$–wertig sind, können wir c_{14} in der Definition L_n hinreichend groß wählen, sodass

$$\|g_{net,m,s}\|_{[-2,2]^4,\infty} \leq 1 + c_{53} \cdot L_n^{-\frac{2p}{4}} \leq 2$$

für alle $m \in \{1, \ldots, l\}$ und $s \in \{1, \ldots, b_m\}$. Wir definieren $\bar{m} \in \mathcal{F}_3(\boldsymbol{\theta}_3)$ wie in Lemma 4. Dann implizieren Lemma 3 und Lemma 4 (für $p \in [1, \infty)$ folgt die Lipschitzstetigkeit der Einschränkungen $g_{m,s}|_{[0,1]^4}$ aus der (p, C)-Glattheit der Funktionen $g_{m,s}$)

$$\begin{aligned}
\inf_{f \in \mathcal{F}_3(\boldsymbol{\theta}_3)} \int |f(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{\mathbf{X}}(d\mathbf{x}) &\leq \int |\bar{m}(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{\mathbf{X}}(d\mathbf{x}) \\
&\leq c_{54} \cdot \max_{m \in \{1,\ldots,l\}, s \in \{1,\ldots,b_m\}} \|g_{m,s} - g_{net,m,s}\|^2_{[-2,2]^4,\infty} \\
&\leq c_{55} \cdot L_n^{-p}.
\end{aligned}$$

Setzen wir den Wert von L_n ein und fassen die obigen Resultate zusammen, ergibt sich die Behauptung. □

3.6.3. Beweis von Theorem 3.3

Wir setzen $\mathcal{F} := \mathcal{F}_2(\boldsymbol{\theta})$ und wählen wie in den Beweisen von Theorem 3.1 und Theorem 3.2 $c_{56} > 0$ hinreichend groß, sodass $c_{56} \cdot \log n \geq 2$. Dann gilt $z \geq 1/2$ genau dann, wenn $T_{c_{56} \cdot \log n} z \geq 1/2$ und somit

$$f_n(\mathbf{x}) = \begin{cases} 1, & \text{falls } T_{c_{56} \cdot \log n} \eta_n(\mathbf{x}) \geq \frac{1}{2} \\ 0, & \text{sonst.} \end{cases}$$

Daher impliziert Ungleichung (1.5), dass es genügt,

$$\mathbf{E}\left\{ \int |T_{c_{56} \cdot \log n} \eta_n(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{g_\lambda(\Phi)}(d\mathbf{x}) \right\} \leq c_{57} \cdot \left(\log(\lambda) \cdot (\log n)^4 \cdot n^{-\frac{2 \cdot p}{2 \cdot p + 4}} + \epsilon_\lambda^2 \right)$$

für eine Konstante $c_{57} > 0$ zu zeigen:
Aufgrund von Lemma 1 gilt

$$\mathbf{E}\left\{ \int |T_{c_{56} \cdot \log n} \eta_n(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{g_\lambda(\Phi)}(d\mathbf{x}) \right\}$$

$$\leq \frac{c_{58} \cdot (\log n)^2 \cdot \sup_{\mathbf{x}_1^n} \left(\log \left(\mathcal{N}_1 \left(\frac{1}{n \cdot c_{56} \cdot \log(n)}, T_{c_{56} \cdot \log(n)} \mathcal{F}, \mathbf{x}_1^n \right) \right) + 1 \right)}{n}$$
$$+ 2 \cdot \inf_{f \in \mathcal{F}} \int |f(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{g_\lambda(\Phi)}(d\mathbf{x})$$

für eine Konstante $c_{58} > 0$. Für den ersten Term liefert Lemma 15 zusammen mit Bemerkung 3.9

$$\frac{c_{58} \cdot (\log n)^2 \cdot \sup_{\mathbf{x}_1^n} \left(\log \left(\mathcal{N}_1 \left(\frac{1}{n \cdot c_{56} \cdot \log(n)}, T_{c_{56} \cdot \log(n)} \mathcal{F}, \mathbf{x}_1^n \right) \right) + 1 \right)}{n}$$
$$\leq \frac{c_{59} \cdot L^2 \cdot \log(L) \cdot \log(\lambda) \cdot (\log n)^3}{n}$$
$$\leq c_{60} \cdot \log(\lambda) \cdot (\log n)^4 \cdot n^{-\frac{2 \cdot p}{2 \cdot p + 4}}$$

für Konstanten $c_{59}, c_{60} > 0$. Als Nächstes ermitteln wir eine Schranke für den Approximationsfehler

$$\inf_{f \in \mathcal{F}} \int |f(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{g_\lambda(\Phi)}(d\mathbf{x}).$$

Da die funktionale a-posteriori Wahrscheinlichkeit η_Φ das L_2-Risko (in Bezug auf den Zufallsvektor (Φ, Y)) minimiert, gilt wegen $\mathbf{P}_\Phi(A) = 1$ und Lemma 11

$$\begin{aligned}
\inf_{f \in \mathcal{F}} \int |f(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{g_\lambda(\Phi)}(d\mathbf{x}) &\leq \int |\bar{f}(\mathbf{x}) - \eta(\mathbf{x})|^2 \mathbf{P}_{g_\lambda(\Phi)}(d\mathbf{x}) \\
&= \mathbf{E}\left\{ |\bar{f}(g_\lambda(\Phi)) - Y|^2 \right\} - \mathbf{E}\left\{ |\eta(g_\lambda(\Phi)) - Y|^2 \right\} \\
&\leq \mathbf{E}\left\{ |\bar{f}(g_\lambda(\Phi)) - Y|^2 \right\} - \mathbf{E}\left\{ |\eta_\Phi(\Phi) - Y|^2 \right\} \\
&= \int_A |\bar{f}(g_\lambda(\phi)) - \eta_\Phi(\phi)|^2 \mathbf{P}_\Phi(d\phi) \\
&\leq c_{61} \cdot \left(n^{-\frac{2 \cdot p}{2 \cdot p + 4}} + \epsilon_\lambda \right)
\end{aligned}$$

für ein durch Lemma 11 gewähltes $\bar{f} \in \mathcal{F}$ und eine Konstante $c_{61} > 0$. Durch Zusammenfassung der obigen Resultate ergibt sich die Behauptung. □

4. Anwendung auf synthetische und reale Bilddatensätze

Die im letzten Kapitel gewonnenen Erkenntnisse zur Konvergenzgeschwindigkeit garantieren eine gute statistische Performanz der Bildklassifikatoren für einen hinreichend großen Stichprobenumfang n. Nun wollen wir das Verhalten der in Abschnitt 3.1 eingeführten Bildklassifikatoren bei endlichem Stichprobenumfang analysieren. Hierfür wenden wir die Klassifikatoren sowohl auf simulierte als auch auf reale Bilddatensätze an und vergleichen die Ergebnisse mit verschiedenen alternativen Klassifikationsmethoden. Wir wollen damit die praktische Relevanz der speziellen Netzwerkarchitekturen aufzeigen, da sie einige Unterschiede zu den in der Praxis bereits verwendeten Netzwerkarchitekturen haben (z.B. durch die spezielle Wahl der Netzwerkparameter).

Wir nehmen an, ein Bilddatensatz sei durch endlich viele Realisierungen $(\mathbf{x}_1, y_1)$, $(\mathbf{x}_2, y_2)$,... einer Zufallsvariable

$$(\mathbf{X}, Y) \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \times \{0,1\}$$

gegeben und bezeichnen einen Bilddatensatz bestehend aus $n \in \mathbb{N}$ Realisierungen mit

$$\mathcal{D}_n = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}.$$

Da alle Klassifikatoren, d.h. sowohl die in Abschnitt 3.1 eingeführten Klassifikatoren als auch die alternativen Methoden, von weiteren Parametern abhängen werden, die nicht während des Trainingsprozesses gewählt werden (sogenannte Hyperparameter), wählen wir diese Parameter datenabhängig durch Unterteilung der Stichprobe. Zu diesem Zweck unterteilen wir den Datensatz $\mathcal{D}_n$ in einen Lerndatensatz $\mathcal{D}_{n_l}$ der Größe $n_l = \lfloor \frac{4}{5} \cdot n \rfloor$ und einen Testdatensatz $\mathcal{D}_n \setminus \mathcal{D}_{n_l}$ der Größe $n_t = n - n_l$. Wir trainieren einen Klassifikator dann für die verschiedene Parameterkombinationen auf dem Lerndatensatz und wählen einen Klassifikator $f_{n_l}(\cdot, \mathcal{D}_{n_l})$ aus, der das empirische Missklassifikationsrisiko

$$\epsilon_{n_t}(f_{n_l}) = \frac{1}{n_t} \sum_{k=1}^{n_t} I_{\{f_{n_l}(\mathbf{x}_{n_l+k}) \neq y_{n_l+k}\}}$$

auf den Testdaten $(\mathbf{x}_{n_l+1}, y_{n_l+1}), \dots, (\mathbf{x}_n, y_n)$ minimiert. Die Qualität des so resultierenden Klassifikators $f_n(\cdot, \mathcal{D}_n)$ wird durch das empirische Missklassifikationsrisiko

$$\epsilon_N(f_n) = \frac{1}{N} \sum_{k=1}^{N} I_{\{f_n(\mathbf{x}_{n+k}) \neq y_{n+k}\}} \tag{4.1}$$

beurteilt, wobei $(\mathbf{x}_{n+1}, y_{n+1}), \dots, (\mathbf{x}_{n+N}, y_{n+N})$ weitere Realisierungen von $(\mathbf{X}, Y)$ bezeichnen. Die gesamte Implementierung erfolgt mittels der Programmiersprache *Python*. Speziell für die Implementierung von Klassifikatoren, die auf neuronalen Netzen basieren, verwenden wir die *Keras* Bibliothek (siehe Chollet et al. (2015)). Hierbei lösen wir das Kleinste-Quadrate Minimierungsproblem (3.14) approximativ mit der stochastischen Gradientenabstiegsmethode *Adam* (siehe Kingma und Ba (2014)), welche in die Keras Bibliothek integriert wurde. Zur Erzeugung aller synthetischen Bilddatensätze verwenden wir die Python Bibliothek *Pillow* (siehe Clark (2015)) und die Python Bibliothek *shapely* (siehe Gillies et al. (2007)).

4.1. Anwendung I: Bildklassifikatoren basierend auf faltenden neuronalen Netzen

Wir analysieren zunächst das Verhalten bei endlichem Stichprobenumfang von dem in Theorem 3.1 verwendeten Klassifikator $f_n^{(1)}$, der auf der Netzwerkarchitektur der Klasse $\mathcal{F}_1(\boldsymbol{\theta})$ basiert. Wir vergleichen die Ergebnisse unseres Klassifikators mit anderen herkömmlichen Klassifizierungsmethoden. Zunächst betrachten wir einen Klassifikator $g_{n,net}$ basierend auf einem vollständig verbundenen neuronalen Netz aus der Klasse $\mathcal{G}_{d_1 \cdot d_2}(L, r)$. Auch hier verwenden wir die Kleinste-Quadrate-Schätzung wie in (1.4) und definieren den Klassifikator als Plug-In Klassifikator gemäß (1.2). Als zweiten alternativen Ansatz betrachten wir eine *Support-Vector-Machine* mit einem *Radiale-Basisfunktion-Kernel* (abgekürzt: *svm-rbf*) und einem polynomialen Kernel (abgekürzt: *svm-p*). Beide Support-Vector-Machine-Ansätze haben einen Parameter C, der die Bedeutung des Regularisierungsterms steuert, und einen Parameter γ, der den Kernelkoeffizienten darstellt. Der polynomiale Kernel hat einen Grad von d (für eine genaue Erklärung der Ansätze siehe beispielsweise Géron (2017)). Für die Implementierung verwenden wir die Funktion *SVC*, die in der Python Bibliothek *scikit-learn* integriert ist (siehe Pedregosa et al. (2011)). Wir vergleichen unseren Schätzer auch mit einem k_n-Nächste-Nachbarn-Schätzer (abgekürzt: *neighbor*) und einem *Random Forest* Klassifikator (abgekürzt *rand-f*), wobei wir die Funktion *RandomForestClassifier* aus der Bibliothek scikit-learn verwenden. Der k_n-Nächste-Nachbar Klassifikator hat nur den Parameter k_n. Für unseren Random Forest Klassifikator (abgekürzt *rand-f*) wählen wir N_k als die maximale Anzahl an Knoten und N_b als die Anzahl der Bäume. Die adaptive Wahl all der genannten Hyperparameter erfolgt mit der im vorherigen Abschnitt beschriebenen Methode der Unterteilung der Stichprobe. Die entsprechenden Parameterkombinationen sind in Tabelle 4.1 zusammengefasst. Dabei ist zu beachten, dass die Hyperparameter unserer Netzwerkarchitektur $\mathcal{F}_1(\boldsymbol{\theta})$ aus Parametern des verallgemeinerten hierarchischen Max-Pooling Modells resultieren. Wir haben diese dabei nicht genau wie in Theorem 3.1 gewählt, sondern die Werte etwas vereinfacht. Um eine Überparametrisierung zu vermeiden, verwenden wir für unseren Klassifikator, welcher auf der Klasse $\mathcal{F}_1(\boldsymbol{\theta})$ basiert, nur solche Parameterkombinationen, bei denen die Gesamtzahl der trainierbaren Gewichte den Stichprobenumfang n nicht überschreitet. Im Folgenden testen wir die obigen

Ansatz	*Adaptiv gewählte Parameter*	*Resultierende Parameter*
$\eta_n^{(1)} \in \mathcal{F}_1(\boldsymbol{\theta})$	$l \in \{2,3,4\}$, $d^* \in \{1,2\}$, $L_n \in \{1,2,3\}$ $k \in \{2,4,8\}$, $r_{net} \in \{5,10\}$	$L = l \cdot L_n$, $L_{net} = L_n$, $A = 32 - 2^l + 1$ $M_{(r-1)\cdot L_n+1}, \ldots, M_{r\cdot L_n} = 2^r$ für $r = 1, \ldots, l$, $t = d^*$, $\mathbf{A} = (1, A, 1, A)$, $\boldsymbol{\theta} = (t, L_{net}, r_{net}, L, k, \mathbf{M}, \mathbf{A})$
$\eta_{n,net} \in \mathcal{G}_{32^2}(L, r)$	$L \in \{1, 2, \ldots, 8\}$, $r \in \{10, 20, 50, 100, 200\}$	
neighbor	$k_n \in \{1,2,3\} \cup \{4, 8, 12, 16, \ldots, 4 \cdot \lfloor \frac{n_l}{4} \rfloor\}$	
rand-f	$N_k \in \{8, 16, 32\}$, $N_b \in \{50, 100, 200\}$	
svm-p	$d \in \{1,2,3,4\}$, $C \in \{10^{-2}, 10^{-1}, 1, 10\}$ $\gamma \in \{10^{-2}, 10^{-1}, 1, 10\}$	
svm-rbf	$C \in \{10^{-2}, 10^{-1}, 1, 10\}$, $\gamma \in \{10^{-2}, 10^{-1}, 1, 10\}$	

Tabelle 4.1.: Anwendung I: Wahl der Hyperparameter, wobei die Funktionen $\eta_n^{(1)}$ und $\eta_{n,net}$ die Kleinste-Quadrate-Schätzer der Plug-In Klassifikatoren $f_n^{(1)}$ und $g_{n,net}$ bezeichnen.

Methoden anhand von drei Klassifikationsproblemen. Die ersten beiden Probleme verwenden synthetisch erzeugte Bilddatensätze, bei denen zwischen zwei Klassen von geometrischen Objekten unterschieden werden soll. Die Datensätze wurden von Glorot und Bengio (2010) inspiriert. Im letzten Problem testen wir die

Verfahren an einem realen Bilddatensatz. In allen drei Bilddatensätzen gilt $d_1 = d_2 = 32$.

Klassifikationsproblem 1: Enthält das Bild einen Kreis?
Die erste Klassifizierungsaufgabe besteht darin, zu erkennen, ob ein Bild einen Kreis enthält. Daher besteht unser synthetischer Bilddatensatz aus Bildern, die keinen Kreis enthalten, und Bildern, die mindestens einen Kreis enthalten. Im Folgenden beschreiben wir, wie ein solches $[0,1]^{\{1,\dots,32\}^2}$-wertiges (zufälliges) Bild $\mathbf{X}$ erzeugt wird. Jedes Bild besteht aus drei geometrischen Objekten, wobei die Objekte zunächst theoretisch auf $[0,32]^2$ definiert werden und dann mit Hilfe der *Pillow* Bibliothek entsprechend auf dem diskreten Gitter $\{1,\dots,32\} \times \{1,\dots,32\}$ ausgewertet werden. Um die geometrischen Objekte zu definieren, verwenden wir die Python Bibliothek *shapely*. Für jedes Objekt wählen wir zufällig und unabhängig voneinander zwischen einem Quadrat, einem gleichseitigen Dreieck und einem Kreis mit jeweils festen Wahrscheinlichkeiten. Der Kreis wird mit der Wahrscheinlichkeit $p = 1 - 0.5^{\frac{1}{3}}$ und das Quadrat und das gleichseitige Dreieck mit der Wahrscheinlichkeit $q = \frac{1}{2} \cdot 0.5^{\frac{1}{3}}$ gewählt. Nachdem ein Objekt als Quadrat, Dreieck oder Kreis definiert wurde, werden seine Fläche, Drehung und Graustufenwerte zufällig ausgewählt. Für jedes Objekt werden Rotation und Fläche unabhängig voneinander gewählt und sind dabei auf den Intervallen $[0, 2\pi]$ und $[60, 80]$ gleichverteilt. Wir bestimmen die Graustufenwerte der drei Objekte durch zufällige Permutation der Liste $(\frac{1}{3}, \frac{2}{3}, 1)$ der drei Graustufenwerte. Die Positionen der Objekte werden nacheinander bestimmt. Für das erste Objekt bestimmen wir dessen Position durch eine Gleichverteilung auf dem begrenzten Bildbereich, sodass das Objekt vollständig im Bild enthalten ist. Die Position des zweiten Objekts wird auf die gleiche Weise gewählt, mit der zusätzlichen Einschränkung, dass das zweite Objekt nur maximal ein Prozent der Fläche des ersten Objekts abdeckt. Hierbei wird die Positionswahl so lange wiederholt, bis diese Bedingung erfüllt ist. Für die Platzierung des dritten Objekts gilt die entsprechende Einschränkung, dass das dritte Bild nur maximal ein Prozent der Fläche des ersten und zweiten Objekts abdeckt. Mit dem obigen Verfahren ist die Klasse Y auf der Menge $\{0,1\}$ (diskret) gleichverteilt, da die Wahrscheinlichkeit, dass das Bild keinen Kreis enthält, $(2 \cdot q)^3 = 0,5$ beträgt. In Abbildung 4.1 sind Realisierungen von $\mathbf{X}$ dargestellt.

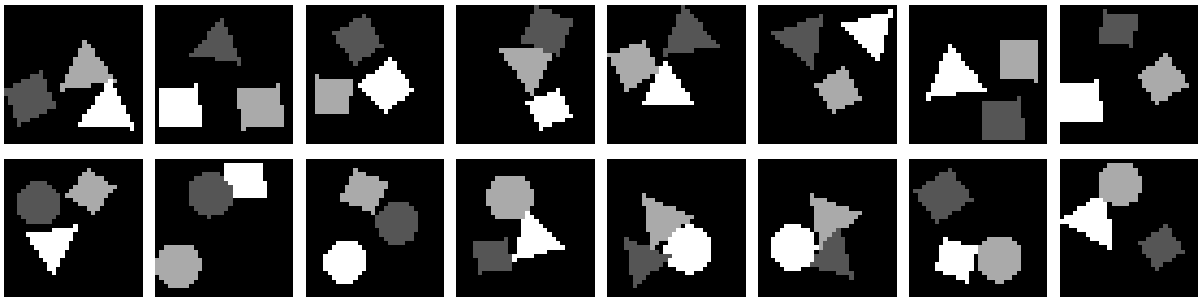


Abbildung 4.1.: Klassifikationsproblem 1: Einige Realisierungen der Zufallsvariable $\mathbf{X}$, wobei die erste Zeile Bilder der Klasse 0 und die untere Zeile Bilder der Klasse 1 zeigt.

Klassifikationsproblem 2: Enthält das Bild zwei gleiche Objekte?
In unserer zweiten Klassifizierungsaufgabe sollen wir entscheiden, ob ein Bild zwei gleiche geometrische Objekte enthält. Der erste Unterschied zum obigen Problem besteht darin, dass nur die beiden geometrischen Objekte Kreis und Dreieck vorhanden sind und jedes Bild nur zwei geometrische Objekte enthält. Ansonsten werden die Bilder auf dieselbe Weise wie oben erzeugt, mit dem Unterschied, dass die beiden Objekte mit einer Wahrscheinlichkeit von $p = 0.5$ ausgewählt werden und die Liste der Graustufenwerte nur aus den Werten $\frac{1}{2}$ und 1 besteht. Auch hier ist die Klasse Y diskret und gleichmäßig auf $\{0,1\}$ verteilt, da die Wahrscheinlichkeit, dass das Bild zwei identische Objekte enthält, durch $2 \cdot p^2 = 0.5$ gegeben ist. In Abbildung 4.2 sind Realisierungen von $\mathbf{X}$ dargestellt.

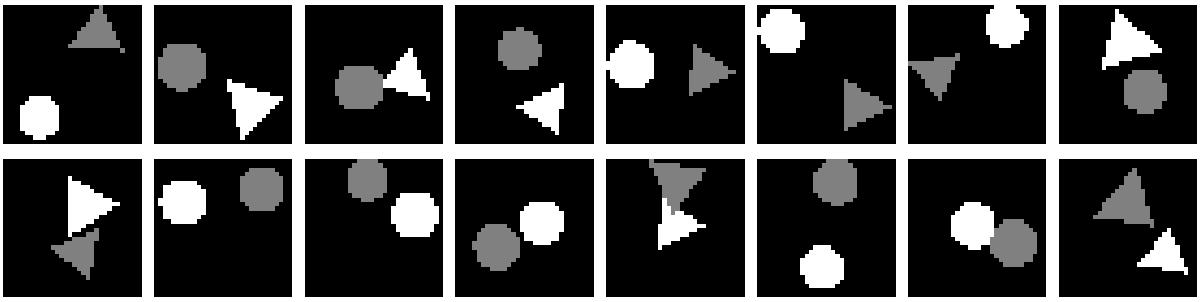


Abbildung 4.2.: Klassifikationsproblem 2: Einige Realisierungen der Zufallsvariable $\mathbf{X}$, wobei die erste Zeile Bilder der Klasse 0 und die untere Zeile Bilder der Klasse 1 zeigt.

Wir vermuten, dass die a-posteriori Wahrscheinlichkeit des ersten Klassifikationsproblems unser verallgemeinertes hierarchisches Max-Pooling Modell mit der Ordnung 1 erfüllt, da nur ein Objekt erkannt werden muss. Um das zweite Klassifikationsproblem zu lösen, wenden wir eine Funktion auf die Information über die Existenz der beiden Objekte an. Daher vermuten wir, dass für das zweite Klassifikationsproblem die a-posteriori Wahrscheinlichkeit einem verallgemeinerten hierarchischen Max-Pooling Modell der Ordnung 2 genügt. Für beide Klassifikationsprobleme betrachten wir den Fall $n = 1000$ sowie den Fall $n = 2000$ und beurteilen die Klassifikatoren anhand eines Datensatzes des Umfangs $N = 10^5$ gemäß (4.1). Da unsere Ergebnisse von zufälligen Daten abhängen, berechnen wir die Schätzer und deren Fehler (4.1) auf der Grundlage von 25 unabhängig generierten Datensätzen $\{(\mathbf{x}_1, y_1), \dots (\mathbf{x}_{n+N}, y_{n+N})\}$. Tabelle 4.2 enthält den Median und den Interquartilabstand (IQR) aller Durchläufe. Es ergibt sich, dass der Klassifikator, der auf der Klasse $\mathcal{F}_1(\boldsymbol{\theta})$

	Klassifikationsproblem 1		*Klassifikationsproblem 2*	
Stichprobenumfang	$n = 1000$	$n = 2000$	$n = 1000$	$n = 2000$
Ansatz	Median (IQR)	Median (IQR)	Median (IQR)	Median (IQR)
$f_n^{(1)}$	**0.0313 (0.0133)**	**0.0139 (0.0121)**	**0.0351 (0.0175)**	**0.0145 (0.0118)**
$g_{n,net}$	0.4737 (0.0068)	0.4591 (0.0120)	0.5003 (0.0041)	0.4980 (0.0089)
neighbor	0.4858 (0.0138)	0.4754 (0.0134)	0.4997 (0.0071)	0.4984 (0.0049)
rand-f	0.4768 (0.0096)	0.4650 (0.0117)	0.4983 (0.0061)	0.4985 (0.0050)
svm-p	0.4377 (0.0129)	0.4168 (0.0105)	0.4988 (0.0074)	0.4992 (0.0071)
svm-rbf	0.4977 (0.0073)	0.4944 (0.0185)	0.4981 (0.0038)	0.4981 (0.0081)

Tabelle 4.2.: Klassifikationsproblem 1 und 2: Median und Interquartilsabstand des empirischen Missklassifikationsrisikos bei 25 Durchläufen.

von faltenden neuronalen Netzen basiert, die anderen Ansätze in beiden Klassifikationsproblemen übertrifft. Die Fehler unseres Klassifikators sind deutlich kleiner als die Fehler der anderen Ansätze. Auch die relative Verbesserung unseres Klassifikators mit zunehmendem Stichprobenumfang ist deutlich größer als die relativen Verbesserungen der anderen Ansätze. Dies könnte darauf hinweisen, dass unser Klassifikator auch eine bessere Konvergenzrate vorweist. Beim zweiten Klassifikationsproblem sind alle Ansätze, bis auf der Klassifikator $f_n^{(1)}$, nicht in der Lage, zufriedenstellende Ergebnisse zu erzielen, da die Fehler der alternativen Ansätze dem erwarteten Fehler eines Klassifikators entsprechen, der immer die gleiche Klasse schätzt.

Klassifikationsproblem 3: Enthält das Bild ein Schiff oder ein Auto?
Wir betrachten den in Krizhevsky (2009) beschriebenen CIFAR-10 Datensatz. Dieser enthält 60.000 Bilder, die aus zehn verschiedenen Klassen bestehen. Wir beschränken uns auf nur zwei dieser Klassen (12.000 Bilder).

Eine Klasse enthält Bilder von Autos und die andere Klasse enthält Bilder von Schiffen. Die Größe jedes Bildes beträgt wie oben 32×32 Pixel. Da die Bilder in Farbe sind, haben wir sie in Graustufen konvertiert. In Abbildung 4.3 sind einige der verwendeten Bilder dargestellt.

Abbildung 4.3.: Klassifikationsproblem 3: Die erste Zeile zeigt einige Bilder der Schiffe und die untere Zeile Bilder der Autos aus dem in Graustufen konvertierten CIFAR-10 Datensatz.

Aus 10.000 der 12.000 Bilder der beiden Klassen des CIFAR-10 Datensatzes wählen wir zufällig $n/2$ Trainingsbilder pro Klasse aus aus und beurteilen die Klassifikatoren anhand der verbleibenden $N = 2.000$ Testbilder. Diesen Vorgang des zufälligen Auswählens der insgesamt n Trainingsbilder und anschließendem Beurteilen der Klassifikatoren wiederholen wir mehrere Male. Tabelle 4.3 zeigt dann den Median und Interquartilabstand von 10 Durchgängen. Auch hier zeigt sich, dass der in dieser Arbeit eingeführte Klassifikator

Stichprobenumfang	$n = 1000$	$n = 2000$
Ansatz	Median (IQR)	Median (IQR)
$f_n^{(1)}$	**0.1982 (0.0201)**	**0.1693 (0.0245)**
$g_{n,net}$	0.2380 (0.0206)	0.2197 (0.0276)
neighbor	0.3643 (0.0140)	0.3738 (0.0086)
rand-f	0.2278 (0.0168)	0.2153 (0.0066)
svm-p	0.3008 (0.0155)	0.2758 (0.0139)
svm-rbf	0.3320 (0.0060)	0.3195 (0.0031)

Tabelle 4.3.: Klassifikationsproblem 3: Median und Interquartilsabstand des empirischen Missklassifikationsrisikos bei zehn Durchläufen.

besser abschneidet als die alternativen Ansätze. Anders als im Fall der synthetischen Bilddatensätze fällt der Unterschied zu den alternativen Ansätzen wesentlich geringer aus. Im Fall der realen Bilder ist es jedoch schwieriger, Vermutungen über die Modellparameter der a-posteriori Wahrscheinlichkeit des verallgemeinerten hierarchischen Max-Pooling Modells anzustellen. Insbesondere haben wir nur die Werte $t \in \{1, 2\}$ getestet, die den Ordnungen $d^* \in \{1, 2\}$ unseres verallgemeinerten hierarchischen Max-Pooling Modells entsprechen.

4.2. Anwendung II: Bildklassifikatoren basierend auf faltenden neuronalen Netzen mit lokalen Pooling Schichten

Als Nächstes wollen wir auch das Verhalten der in Theorem 3.2 verwendeten Klassifikatoren $f_n^{(3)}$, $f_n^{(4)}$ und $f_n^{(5)}$ bei endlichem Stichprobenumfang analysieren. Wir vergleichen die drei Klassifikatoren, die alle drei mindestens eine lokale Pooling Schicht enthalten, mit einem Klassifikator $f_n^{(6)}$, der keine lokale Pooling Schicht enthält und der Architektur aus Theorem 3.1 nachempfunden ist. Da die drei in Theorem 3.2

vorgestellten Netzwerkarchitekturen allerdings nur aus einem faltenden neuronalen Netz bestehen und nicht aus t parallel berechneten Netzwerken, auf deren Ausgaben ein vollverbundenes neuronales Netz angewendet wird, verwenden wir für den Schätzer ohne lokale Pooling Schicht ein faltendes neuronales Netz aus der Klasse $\mathcal{F}_5(\boldsymbol{\theta})$ (siehe Abschnitt 3.1), wobei der Pooling Parameter $s = 1$ gewählt wird (damit entspricht die Pooling Schicht der Identität). Die Parametermengen für die adaptive Wahl der Hyperparameter ist wie oben in Tabelle 4.4 zusammengefasst. Auch bei den Anwendungen in diesem Abschnitt wollen wir eine Überparametrisierung vermeiden und berücksichtigen nur die Parameterkombinationen, bei denen die Gesamtanzahl der trainierbaren Gewichte nicht den Stichprobenumfang n überschreitet. Wir haben diesmal $d_1 = d_2 = 31$ gewählt, damit die Bilddimensionen Bedingung (3.17) aus Theorem 3.2 erfüllen (siehe auch Bemerkung 3.4). Im Folgenden testen wir die vier Klassifikatoren anhand drei weiterer Klassifikationsprobleme. Das erste Klassifikationsproblem beinhaltet synthetisch erzeugte Bilder, bei denen zwischen zwei Klassen von geometrischen Objekten unterschieden werden soll. Danach wird die Performanz noch unter der Verwendung zweier realer Bilddatensätze beurteilt.

Ansatz	*Adaptiv gewählte Parameter*	*Resultierende Parameter*
$\eta_n^{(3)} \in \mathcal{F}_3(\boldsymbol{\theta})$	$l \in \{3,4\}$, $z \in \{1,2,3\}$, $k \in \{2,4,8\}$ $n_r \in \{2^0,\dots,2^r\}$ mit $\prod_{i=0}^{r} n_i \leq 2^r$ für $r = 0,\dots,l-1$	$L = l$, $A = \frac{31-2^l+1}{\prod_{i=1}^{l-1} n_i}$, $\mathbf{s} = (n_1,\dots,n_{L-1})$ $M_r = \frac{2^{r-1}}{\prod_{i=0}^{r-1} n_i} + 1$ für $r = 1,\dots,L$ $\mathbf{A} = (1,A,1,A)$, $\boldsymbol{\theta} = (L,k,\mathbf{M},\mathbf{s},\mathbf{A})$
$\eta_n^{(4)} \in \mathcal{F}_4(\boldsymbol{\theta})$	$l \in \{3,4\}$, $z \in \{1,2,3\}$, $k \in \{2,4,8\}$ $n_r \in \{2^0,\dots,2^r\}$ mit $\prod_{i=0}^{r} n_i \leq 2^r$ für $r = 0,\dots,l-1$	$L = l$, $A = \frac{31-2^l+1}{\prod_{i=1}^{l-1} n_i}$, $\mathbf{s} = (n_1,\dots,n_{L-1})$ $M_r = \frac{2^{r-1}}{\prod_{i=0}^{r-1} n_i} + 1$ für $r = 1,\dots,L$ $\mathbf{A} = (1,A,1,A)$, $\boldsymbol{\theta} = (L,k,\mathbf{M},\mathbf{s},\mathbf{A})$
$\eta_n^{(5)} \in \mathcal{F}_5(\boldsymbol{\theta})$	$l \in \{3,4\}$, $z \in \{1,2,3\}$, $k \in \{2,4,8\}$ $n_r \in \{2^0,\dots,2^r\}$ mit $\prod_{i=0}^{r} n_i \leq 2^r$ für $r = 0,\dots,l-1$	$L = l$, $A = \frac{31-2^l+1}{\prod_{i=1}^{l-1} n_i}$, $s = n_1 \cdot \dots \cdot n_{L-1}$ $M_r = 2^{r-1} + 1$ für $r = 1,\dots,l$ $\mathbf{A} = (1,A,1,A)$, $\boldsymbol{\theta} = (L,k,\mathbf{M},s,\mathbf{A})$
$\eta_n^{(6)} \in \mathcal{F}_5(\boldsymbol{\theta})$	$l \in \{3,4\}$, $z \in \{1,2,3\}$, $k \in \{2,4,8\}$	$L = l \cdot z$, $A = 31 - 2^l + 1$, $s = 1$ $M_{(r-1)\cdot z+1},\dots,M_{r\cdot z} = 2^{r-1}+1$ für $r = 1,\dots,l$, $\mathbf{A} = (1,A,1,A)$, $\boldsymbol{\theta} = (L,k,\mathbf{M},s,\mathbf{A})$

Tabelle 4.4.: Anwendung II: Wahl der Hyperparameter, wobei die Funktion $\eta_n^{(j)}$ den Kleinste-Quadrate-Schätzer des Plug-In Klassifikators $f_n^{(j)}$ $(j = 3,\dots,6)$ bezeichnet.

Klassifikationsproblem 4: Sind die geometrischen Objekte relativ zueinander richtig angeordnet?
Beide Klassen unserer synthetisch erzeugten Bilder bestehen aus denselben geometrischen Objekten, nämlich einem Kreis, einem gleichseitigen Dreieck und einem Quadrat. Der einzige Unterschied zwischen den beiden Klassen sind die relativen Positionen der geometrischen Objekte zueinander. Zunächst bestimmen wir die Klasse Y durch eine (diskrete) Gleichverteilung auf $\{0,1\}$ und erzeugen anschließend ein (zufälliges) Bild der entsprechenden Klasse. Wir beschreiben im Folgenden, wie die Bilder der beiden Klassen erzeugt werden. Die Objekte werden zunächst theoretisch auf dem Quader $[0,31]^2$ definiert und erst danach mit Hilfe der Python Bibliothek *Pillow* auf das Gitter $\{1,\dots,31\} \times \{1,\dots,31\}$ heruntergerechnet. Wir beginnen mit der zufälligen Wahl der Flächen und der Graustufenwerte der drei Objekte. Für jedes Objekt wird die Fläche unabhängig und gleichverteilt auf dem Intervall $[20,40]$ gewählt. Wir bestimmen die Graustufenwerte der drei Objekte durch eine zufällige Permutation der Liste $(0, \frac{1}{3}, \frac{2}{3})$. Als Nächstes beschreiben wir, wie die Positionen der Objekte im Verhältnis zueinander bestimmt werden und wie dies zu den spezifischen Positionen innerhalb des Bildbereichs führt. Für beide Klassen werden in einem ersten Schritt die Objekte ungefähr als Ecken eines

großen Quadrats angeordnet, dessen Fläche durch eine Gleichverteilung auf dem Intervall $[80, 160]$ gewählt wird. Für beide Klassen definiert die untere rechte Ecke die Position des Quadrats. Die Position des Kreises wird für die Klasse 0 durch die obere linke Ecke und für die Klasse 1 durch die untere linke Ecke bestimmt. Die Position des Dreiecks wird für die Klasse 0 durch die untere linke Ecke und für die Klasse 1 durch die obere rechte Ecke bestimmt. Da die Objekte nur annähernd diese Positionen zueinander haben sollen, werden in einem zweiten Schritt die Positionen durch eine Gleichverteilung auf einer Kreisfläche gewählt, die sich auf den entsprechenden Ecken des großen Quadrats befindet, wobei der Radius der Kreisflächen ein Drittel einer Seitenlänge des großen Quadrats beträgt. Sobald die Positionen der Objekte relativ zueinander auf diese Weise bestimmt wurden, werden die so angeordneten Objekte gemeinsam durch eine Gleichverteilung auf der Bildfläche verschoben, sodass alle Objekte vollständig im Bild enthalten sind. Dies führt uns zu den gewünschten Positionen. In einem letzten Schritt addieren wir zu jedem Pixelwert ein $\mathcal{N}(0, 0.01)$-verteiltes „Rauschen“ und stutzen die resultierenden Werte auf das Intervall $[0, 1]$. In Abbildung 4.4 und Abbildung 4.5 sind einige Beispielbilder dargestellt.

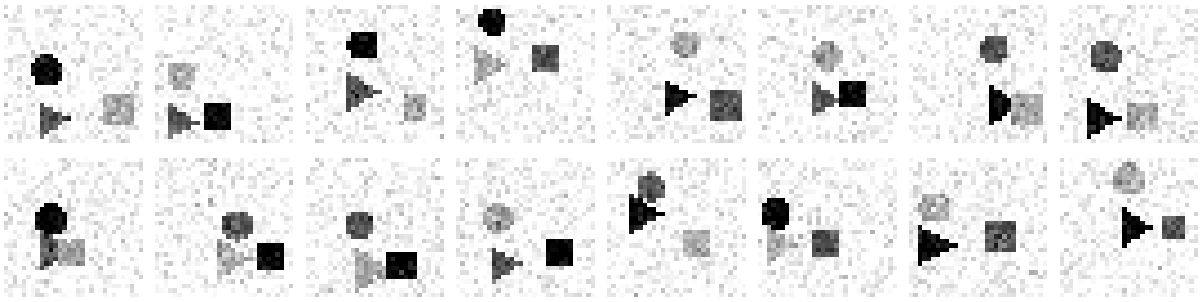


Abbildung 4.4.: Klassifikationsproblem 4: Einige Beispielbilder der Klasse 0.

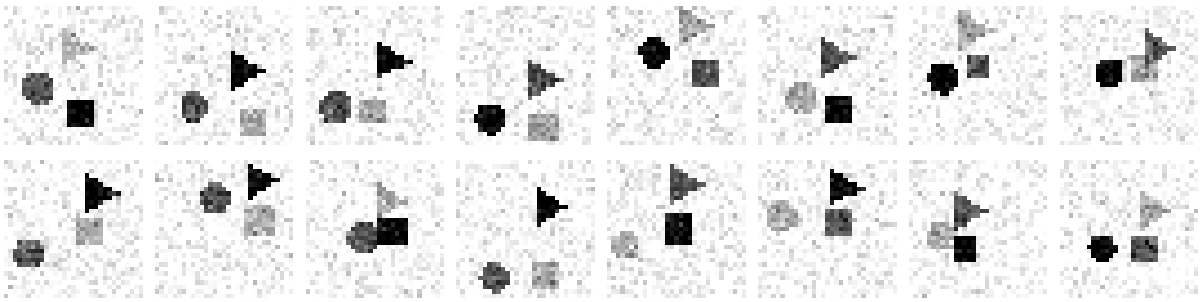


Abbildung 4.5.: Klassifikationsproblem 4: Einige Beispielbilder der Klasse 1.

Wir betrachten den Fall $n = 200$ und den Fall $n = 400$ und beurteilen die Klassifikatoren anhand eines Datensatzes der Größe $N = 10^5$ gemäß (4.1). Wie in Abschnitt 4.1 berechnen wir den Median und Interquartilabstand (IQR) des empirischen Missklassifikationsrisikos von 25 Durchläufen. Die Ergebnisse sind in Tabelle 4.5 zusammengefasst. Der Klassifikator $f_n^{(3)}$ mit mehreren lokalen Max-Pooling Schichten, dessen Netzwerkarchitektur sich am meisten auf das Modell für die a-posteriori Wahrscheinlichkeit stützt (siehe Lemma 4), schneidet am besten ab. Dies könnte ein Indiz dafür sein, dass unsere Annahme des hierarchischen Max-Pooling Modells mit zusätzlichem lokalen Pooling plausibel ist. Der Klassifikator $f_n^{(4)}$ schneidet am zweitbesten ab. Die Klassifikatoren $f_n^{(5)}$ und $f_n^{(6)}$ schneiden deutlich schlechter ab. Unsere Analyse demonstriert somit experimentell die Nützlichkeit lokaler Pooling Schichten in unseren speziellen Netzwerkarchitekturen.

Stichprobenumfang	$n = 200$	$n = 400$
Ansatz	Median (IQR)	Median (IQR)
$f_n^{(3)}$	**0.1142 (0.0409)**	**0.0412 (0.0196)**
$f_n^{(4)}$	0.1600 (0.1208)	0.0505 (0.0158)
$f_n^{(5)}$	0.4649 (0.0773)	0.3488 (0.1033)
$f_n^{(6)}$	0.4558 (0.0446)	0.3215 (0.1288)

Tabelle 4.5.: Klassifikationsproblem 4: Median und Interquartilabstand des empirischen Missklassifikationsrisikos bei 25 Durchläufen.

Klassifikationsproblem 5: Enthält das Bild einen Hund oder eine Katze?
Für das nächste Klassifikationsproblem betrachten wir wieder den in Krizhevsky (2009) beschriebenen CIFAR-10 Datensatz. Im Gegensatz zu Klassifikationsproblem 3 verwenden wir diesmal die beiden Klassen „Hund“ und „Katze“, da wir hier vermuten, dass Merkmale der beiden Objekte variable relative Abstände zueinander besitzen. Damit auch hier die Bedingung 3.17 an die Bilddimensionen erfüllt ist, haben wir die Pixel der untersten Zeile und der rechten Spalte aller Bilder entfernt, womit $d_1 = d_2 = 31$ gilt. Außerdem konvertieren wir die Bilder wieder in Graustufen. In Abbildung 4.6 sind einige Beispielbilder dargestellt. Wie für das dritte Klassifikationsproblem wählen wir aus 10.000 der 12.000 Bilder der beiden Klassen des CIFAR-10 Datensatzes zufällig $n/2$ Trainingsbilder pro Klasse aus und bewerten die Klassifikatoren anhand der verbleibenden $N = 2.000$ Testbilder. Tabelle 4.6 zeigt dann den Median und Interquartilabstand (IQR) des empirischen Missklassifikationsrisikos von 25 Durchgängen.

Abbildung 4.6.: Klassifikationsproblem 5: Die erste Zeile zeigt einige Bilder der Hunde und die untere Zeile Bilder der Katzen aus dem in Graustufen konvertierten CIFAR-10 Datensatz.

Klassifikationsproblem 6: Enthält das Bild die Hausnummer Vier oder die Hausnummer Neun?
Als Nächstes betrachten wir den SVHN Datensatz (siehe Netzer et al. (2011)), der aus Bildern von Hausnummern besteht. Wie oben reduzieren wir die Auflösung von ursprünglich 32×32 Pixeln auf 31×31 Pixel und konvertieren die Bilder in Graustufen. Weiterhin verwenden wir nur die beiden Klassen „Vier“ und „Neun“. In Abbildung 4.7 haben wir einige Beispielbilder dargestellt. Der Originaldatensatz reduziert sich dann auf 12.117 Trainingsbilder und $N = 4.118$ Testbilder. Wir verfahren analog zu oben und wählen zufällig und mehrere Male $n/2$ Trainingsbilder pro Klasse aus den 12.117 Bildern aus und beurteilen die Klassifikatoren anhand der N Testbilder. Die Ergebnisse von 25 Durchläufen sind in Tabelle 4.6 zusammengefasst.

Für den CIFAR-10 Datensatz schneidet der Klassifikator $f_n^{(6)}$ besser ab als die anderen Klassifikatoren, auch wenn wir für alle vier Ansätze ähnliche Fehler erhalten. Die Klassifikationsaufgabe scheint für die betrachteten Klassifikatoren und die Werte für den Stichprobenumfang n zu schwer zu sein, da die Fehler nah bei dem erwarteten Fehler eines Klassifikators liegen, der immer die gleiche Klasse schätzt. Beim SVHN

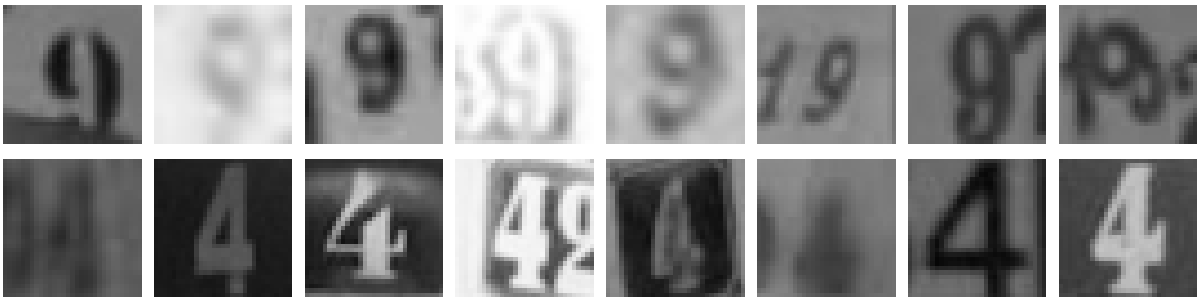


Abbildung 4.7.: Klassifikationsproblem 6: Die erste Zeile zeigt einige Bilder der Neuner und die untere Zeile Bilder der Vierer aus dem in Graustufen konvertierten SVHN Datensatz.

	Klassifikationsproblem 5		Klassifikationsproblem 6	
Stichprobenumfang	$n = 200$	$n = 400$	$n = 200$	$n = 400$
Ansatz	Median (IQR)	Median (IQR)	Median (IQR)	Median(IQR)
$f_n^{(3)}$	0.4760 (0.0275)	0.4555 (0.0295)	**0.2938 (0.0765)**	**0.2305 (0.0709)**
$f_n^{(4)}$	0.4760 (0.0335)	0.4700 (0.0295)	0.3334 (0.0505)	0.2579 (0.0367)
$f_n^{(5)}$	0.4710 (0.0285)	0.4575 (0.0170)	0.4696 (0.0668)	0.3834 (0.1370)
$f_n^{(6)}$	**0.4634 (0.0245)**	**0.4440 (0.0210)**	0.4388 (0.0682)	0.3822 (0.1146)

Tabelle 4.6.: Klassifikationsproblem 5 und 6: Median und Interquartilabstand des empirischen Missklassifikationsrisikos bei 25 Durchläufen.

Datensatz zeigt sich, dass der Klassifikator $f_n^{(3)}$ besser abschneidet als die anderen Ansätze, wobei sich die Fehler aller Klassifikatoren im Verhältnis zueinander in etwa so verhalten wie bei den synthetischen Bildern aus dem vierten Klassifikationsproblem (die Klassifikatoren $f_n^{(3)}$ und $f_n^{(4)}$ schneiden deutlich besser ab als die Klassifikatoren $f_n^{(5)}$ und $f_n^{(6)}$). Dabei können die Klassifikatoren $f_n^{(3)}$ und $f_n^{(4)}$ auch eine größere relative Verbesserung mit zunehmendem Stichprobenumfang vorweisen als die beiden anderen Klassifikatoren. Dies könnte darauf hindeuten, dass diese Klassifikatoren auch eine bessere Konvergenzrate besitzen. Einerseits zeigt dies erneut die Nützlichkeit lokaler Pooling Schichten unserer speziellen Netzwerkarchitekturen, andererseits gehen wir wie oben davon aus, dass dies zumindest für den SVHN Datensatz ein Indiz für die Plausibilität unserer Annahme des zusätzlichen lokalen Max-Poolings an die a-posteriori Wahrscheinlichkeit liefert.

4.3. Anwendung III: Klassifizierung rotierter Objekte mit faltenden neuronalen Netzen

In der letzten Simulationsstudie analysieren wir das Verhalten des in Theorem 3.3 eingeführten Klassifikators bei endlichem Stichprobenumfang sowohl auf einem synthetischen als auch auf einem realen Bilddatensatz. Außerdem führen wir drei weitere Netzwerkarchitekturen ein, die wir durch unsere theoretische Analyse aus Kapitel 3 motivieren können. Wir vergleichen die Leistung aller vier Bildklassifikatoren, die auf faltenden neuronalen Netzen basieren, mit den zwei alternativen Ansätzen $g_{n,net}$ und *neighbor* aus Abschnitt 4.1, die nicht speziell auf den Aspekt der Rotation ausgerichtet sind. In der ersten alternativen Netzwerkarchitektur ersetzen wir das vollverbundene neuronale Netz in der Definition der Klasse $\mathcal{F}_2(\boldsymbol{\theta})$ durch eine Maximumsberechnung über die t Ausgaben der t faltenden neuronalen Netze:

$$\mathcal{F}_7(\boldsymbol{\theta}) = \{f(\mathbf{x}) = \max\{f_1(\mathbf{x}), \ldots, f_t(\mathbf{x})\} : f_1, \ldots, f_t \in \mathcal{F}_{CNN}(L, k, \mathbf{M}, \mathbf{P}_2, \mathbf{A})\},$$

wobei $\mathcal{F}_{CNN}(\boldsymbol{\theta})$ die in Gleichung (3.7) eingeführte Netzwerkarchitektur bezeichnet und der Vektor $\mathbf{P}_2$ für das symmetrische Zero-Padding in Gleichung (3.10) definiert wurde. Die Netzwerkarchitektur $\mathcal{F}_7(\boldsymbol{\theta})$ hängt dann von einem Parametervektor $\boldsymbol{\theta} = (t, L, k, \mathbf{M}, \mathbf{A})$ ab. Folgen wir dem Beweis von Theorem 3.3, ist es leicht zu sehen, dass der entsprechende Kleinste-Quadrate Plug-In Klassifikator über dieser Funktionsklasse die gleiche Konvergenzrate wie in Theorem 3.3 erreicht. Die zweite alternative Architektur ist inspiriert durch die Beobachtung von Bemerkung 3.7. Wir folgen hier beispielsweise Dieleman et al. (2015) oder Cabrera-Vives et al. (2017), indem wir das gleiche faltende neuronale Netz auf mehrere rotierte Versionen des Eingabebildes anwenden. Die Gesamtausgabe der Netzwerkarchitektur berechnen wir durch das Maximum der einzelnen Ausgaben. Wir rotieren das Eingabebild durch $90°$, $180°$ und $270°$ Rotationen, da Vielfache von $90°$ Rotationen das Gitter $\{1, \dots, \lambda\}^2$ auf sich selbst abbilden. An dieser Stelle macht es keinen Unterschied, ob wir die Eingabekanäle einer faltenden Schicht rotieren und anschließend die Ausgabekanäle zurück rotieren, oder ob wir die in der Schicht verwendeten Filter rotieren (für eine Illustration und eine ausführlichere Erklärung siehe Dieleman et al. (2016)). Daher entspricht diese Architektur in unserem Fall einer Architektur, die geteilte rotierte Filter besitzt. Die Rotationsfunktion $rot_{90°} : [0,1]^{\{1,\dots,\lambda\}^2} \to [0,1]^{\{1,\dots,\lambda\}^2}$, die ein Bild mit der Auflösung $\lambda \in \mathbb{N}$ um $90°$ rotiert, ist gegeben durch

$$\big(rot_{90°}(\mathbf{x})\big)_{(i,j)} = x_{\lambda-j+1,i} \quad \Big(\mathbf{x} \in [0,1]^{\{1,\dots,\lambda\}^2}\Big)$$

für alle $i, j \in \{1, \dots, \lambda\}$. Wir definieren dann die von einem Parametervektor $\boldsymbol{\theta} = (t, L, k, \mathbf{M}, \mathbf{A})$ abhängige Funktionsklasse

$$\mathcal{F}_8(\boldsymbol{\theta}) = \Big\{ f(\mathbf{x}) = \max\{g(\mathbf{x}), g(rot_{90°}(\mathbf{x})), g(rot_{90°} \circ rot_{90°}(\mathbf{x})), g(rot_{90°} \circ rot_{90°} \circ rot_{90°}(\mathbf{x}))\} : g \in \mathcal{F}_7(\boldsymbol{\theta}) \Big\}.$$

Für die dritte alternative Netzwerkarchitektur erweitern wir den obigen Ansatz der Funktionsklasse $\mathcal{F}_8(\boldsymbol{\theta})$, indem wir ein Eingabebild zunächst um alle Winkel einer Diskretisierung

$$\{\alpha_1, \dots, \alpha_t\} = \left\{ \frac{2\pi}{t} \cdot 0, \frac{2\pi}{t} \cdot 1, \dots, \frac{2\pi}{t} \cdot (t-1) \right\}$$

von $[0, 2\pi)$ für ein $t \in \mathbb{N}$ rotieren. Zu diesem Zweck definieren wir eine Rotationsfunktion, die ein diskretes Bild auch für beliebige Winkel rotiert. Wir verwenden hierbei aus zwei Gründen eine Nächste-Nachbar-Interpolation: Erstens kann eine Nächste-Nachbar-Interpolation einfach mit der *Keras* Bibliothek als Schicht eines faltenden neuronalen Netzes implementiert werden, sodass der entsprechende Klassifikator mit der *Adam* Methode trainiert werden kann. Zweitens könnte unsere Theorie leicht auf einen solchen Schätzer erweitert werden, da die Nächste-Nachbar-Interpolation auf eine Abbildung zurückgeführt werden kann, die das Gitter $\{1, \dots, \lambda\}^2$ auf sich selbst abbildet (vgl. Gleichung (4.2) unten) und demnach Pixelwerte durch an anderen Positionen vorhandene Pixelwerte ersetzt. Um zu vermeiden, dass Teile des Bildes aus dem Bildbereich heraus rotiert werden, definieren wir zunächst eine Funktion $f_z : [0,1]^{\{1,\dots,\lambda\}^2} \to [0,1]^{\{1,\dots,\lambda+2\cdot z\}^2}$, die jeweils $z \in \mathbb{N}_0$ Zeilen bzw. Spalten an Nullen an den vier Bildseiten hinzufügt. Die Ausgabe der Funktion f_z ist gegeben durch

$$\big(f_z(\mathbf{x})\big)_{(i,j)} = \begin{cases} x_{i,j}, & \text{falls } z+1 \leq i,j \leq z+\lambda \\ 0, & \text{sonst} \end{cases}$$

für $i, j \in \{1, \dots, \lambda + 2 \cdot z\}$. Wir wählen

$$z_\lambda = \left\lceil \frac{\sqrt{2} \cdot \lambda - \lambda}{2} \right\rceil,$$

um sicherzustellen, dass eine rotierte Version eines Bildes das vollständige ursprüngliche Bild enthält. Als Nächstes definieren wir die Funktion $g^{(\alpha)} : \{1, \dots, \lambda'\}^2 \to \{1, \dots, \lambda'\}^2$, welche die Bildpositionen eines Bildes

mit Auflösung $\lambda' \in \mathbb{N}$ um den Winkel $\alpha \in [0, 2\pi)$ rotiert. Die Ausgabe dieser Funktion ist gegeben durch

$$g^{(\alpha)}(\mathbf{v}) = \varphi^{-1}\left(\underset{\mathbf{u}\in G_{\lambda'}}{\arg\min} \|\mathbf{u} - rot^{(\alpha)}(\varphi(\mathbf{v}))\|_2\right) \quad \big(\mathbf{v} \in \{1, \dots, \lambda'\}^2\big), \tag{4.2}$$

wobei wir im Falle eines nicht eindeutigen Minimums den kleinsten Index wählen (betrachte dazu eine Bijektion von $\{1, \dots, \lambda'\}^2$ auf $\{1, \dots, \lambda'^2\}$). Außerdem ist die Bijektion $\varphi : \{1, \dots, \lambda'\}^2 \to G_{\lambda'}$ wie im Beweis von Lemma 12 durch Gleichung (3.46) definiert, und die Funktion $rot^{(\alpha)} : \mathbb{R}^2 \to \mathbb{R}^2$ bezeichnet die Rotationsfunktion aus Abschnitt 2.3. Die Rotationsfunktion $f_{rot}^{(\alpha)} : [0,1]^{\{1,\dots,\lambda\}^2} \to [0,1]^{\{1,\dots,\lambda+2\cdot z_\lambda\}^2}$, die ein diskretes Bild um den Winkel $\alpha \in [0, 2\pi)$ rotiert, ist dann definiert durch

$$\big(f_{rot}^{(\alpha)}(\mathbf{x})\big)_{(i,j)} = \big(f_{z_\lambda}(\mathbf{x})\big)_{g^{(\alpha)}((i,j))} \quad \big(\mathbf{x} \in [0,1]^{\{1,\dots,\lambda\}^2}\big)$$

für $(i,j) \in \{1, \dots, \lambda + 2 \cdot z_\lambda\}^2$. Somit ergibt sich die Funktionsklasse

$$\mathcal{F}_9(\boldsymbol{\theta}) = \Big\{f(\mathbf{x}) = \max\{g(f_{rot}^{(\alpha_1)}(\mathbf{x})), g(f_{rot}^{(\alpha_2)}(\mathbf{x})), \dots, g(f_{rot}^{(\alpha_t)}(\mathbf{x}))\} : g \in \mathcal{F}_{CNN}(L, k, \mathbf{M}, \mathbf{P}_2, \mathbf{A})\Big\},$$

die von einem Parametervektor $\boldsymbol{\theta} = (t, L, k, \mathbf{M}, \mathbf{P}_2, \mathbf{A})$ abhängt. Die Parametermengen für die adaptive Wahl der Hyperparameter sind in Tabelle 4.7 zusammengefasst. Im Unterschied zu den beiden vorherigen Anwendungen aus Abschnitten 4.1 und 4.2 lassen wir diesmal auch Parameterkombinationen zu, bei denen die Gesamtzahl der trainierbaren Gewichte den Stichprobenumfang n überschreitet. Der Grund ist, dass wir den Vergleich der vier Netzwerkarchitekturen fair gestalten wollen, sodass jeder der vier Schätzer die Möglichkeit hat, gleich viele Rotationswinkel zu lernen. Das erklärt auch die unterschiedliche Wahl des Parameters t, der für die Klasse $\mathcal{F}_8(\boldsymbol{\theta})$ aus der Menge $\{1, 2\}$ und für die Klasse $\mathcal{F}_9(\boldsymbol{\theta})$ $t = 8$ gewählt wird, wodurch die beiden Klassen genauso viele Rotationswinkel wie die Klassen $\mathcal{F}_2(\boldsymbol{\theta})$ und $\mathcal{F}_7(\boldsymbol{\theta})$ lernen können. Außerdem stellt die adaptive Wahl der Schichten und Filtergrößen der Netzwerkarchitektur der Klasse $\mathcal{F}_2(\boldsymbol{\theta})$ (siehe Tabelle 4.7) eine leichte Vereinfachung der in Theorem 3.3 verwendeten Netzwerkarchitektur dar. Wie in Abschnitt 2.3 bezeichnet $\lambda \in \mathbb{N}$ die Auflösung der Bilder (d.h. es gilt $d_1 = d_2 = \lambda$). In unserem ersten Beispiel verwenden wir die Werte $\lambda = 32$ und $\lambda = 64$.

Klassifikationsproblem 7: Enthält das Bild ausschließlich vollständige Quadrate?
Die Bilder beider Klassen enthalten jeweils drei zufällig gedrehte geometrische Objekte, wobei die Bilder der Klasse 0 drei Quadrate enthalten. Die Bilder der Klasse 1 enthalten ebenfalls drei Quadrate, wobei mindestens einem der Quadrate genau ein Viertel fehlt (siehe Abbildung 4.8). Um ein Zufallsbild mit einer entsprechenden Klasse zu erzeugen, wird zunächst der Graustufenwert des Hintergrunds des Bildbereichs $[0, \lambda]^2$ auf 0 gesetzt und für jedes der drei Quadrate wird zufällig (unabhängig) bestimmt, ob ein Viertel entfernt wird oder nicht. Die Wahrscheinlichkeit, dass ein Viertel aus einem Quadrat entfernt wird, beträgt dabei $p = 1 - 0.5^{1/3}$, was bedeutet, dass die Klasse Y eines Bildes diskret auf $\{0, 1\}$ gleichverteilt ist. Als Nächstes werden die Fläche, die Rotation und der Graustufenwert jedes geometrischen Objekts bestimmt. Die Fläche wird für jedes Objekt (unabhängig) durch eine Gleichverteilung auf dem Intervall $[0.02, 0.08]$ für vollständige Quadrate und auf dem Intervall $[0.02, 0.06]$ für Quadrate, denen ein Viertel fehlt, bestimmt (das zweite Intervall ist kleiner, um zu große Seitenlängen dieser Objekte zu vermeiden). Der Winkel, um den ein Objekt rotiert wird, wird (unabhängig) durch eine Gleichverteilung auf dem Intervall $[0, 2\pi]$ bestimmt. Die Graustufenwerte der drei Objekte werden durch eine zufällige Permutation der Liste $(1/3, 2/3, 1)$ von drei Graustufenwerten bestimmt. Schließlich werden die Positionen der Objekte nacheinander wie folgt bestimmt: Wir wählen die Position des ersten Objekts durch Gleichverteilung auf dem eingeschränkten Bildbereich, sodass das Objekt vollständig innerhalb des Bildbereichs liegt. Die Positionierung des zweiten Objekts wiederholen wir auf die gleiche Weise solange, bis das zweite Objekt nur maximal fünf Prozent der Fläche des ersten Objekts überdeckt. Für die

Ansatz	*Adaptiv gewählte Parameter*	*Resultierende Parameter*
$\eta_n^{(2)} \in \mathcal{F}_2(\boldsymbol{\theta})$	$l \in \{2,3\}$, $k \in \{2,4\}$, $L_n \in \{1,2\}$, $t \in \{4,8\}$	$L = L_n \cdot l$, $A = 1 + 2^{l-1} - l + 1$, $A' = \lambda - 2^{l-1} + l - 1$, $\mathbf{A} = (A, A', A, A')$ $M_{(r-1)\cdot L_n+1}, \ldots, M_{r\cdot L_n} = I_{\{r>2\}} \cdot 2^{r-2} + 3$ für $r = 1, \ldots, l$, $L_{net} = \lceil \log_2(t) \rceil$, $r_{net} = 3 \cdot t$, $\boldsymbol{\theta} = (t, L_{net}, r_{net}, L, k, \mathbf{M}, \mathbf{A})$
$\eta_n^{(7)} \in \mathcal{F}_7(\boldsymbol{\theta})$	$l \in \{2,3\}$, $k \in \{2,4\}$, $L_n \in \{1,2\}$, $t \in \{4,8\}$	$L = L_n \cdot l$, $A = 1 + 2^{l-1} - l + 1$, $A' = \lambda - 2^{l-1} + l - 1$, $\mathbf{A} = (A, A', A, A')$ $M_{(r-1)\cdot L_n+1}, \ldots, M_{r\cdot L_n} = 2^{r-1} + 1$ für $r = 1, \ldots, l$, $\boldsymbol{\theta} = (t, L, k, \mathbf{M}, \mathbf{A})$
$\eta_n^{(8)} \in \mathcal{F}_8(\boldsymbol{\theta})$	$l \in \{2,3\}$, $k \in \{1,2\}$, $L_n \in \{1,2\}$, $t \in \{1,2\}$	$L = L_n \cdot l$, $A = 1 + 2^{l-1} - l + 1$, $A' = \lambda - 2^{l-1} + l - 1$, $\mathbf{A} = (A, A', A, A')$ $M_{(r-1)\cdot L_n+1}, \ldots, M_{r\cdot L_n} = 2^{r-1} + 1$ für $r = 1, \ldots, l$, $\boldsymbol{\theta} = (t, L, k, \mathbf{M}, \mathbf{A})$
$\eta_n^{(9)} \in \mathcal{F}_9(\boldsymbol{\theta})$	$l \in \{2,3\}$, $k \in \{1,2\}$, $L_n \in \{1,2\}$	$L = L_n \cdot l$, $t = 8$, $A = 1 + 2^{l-1} - l + 1$, $A' = \lambda - 2^{l-1} + l - 1$, $\mathbf{A} = (A, A', A, A')$ $M_{(r-1)\cdot L_n+1}, \ldots, M_{r\cdot L_n} = 2^{r-1} + 1$ für $r = 1, \ldots, l$, $\boldsymbol{\theta} = (t, L, k, \mathbf{M}, \mathbf{A})$
$\eta_{n,net} \in \mathcal{G}_{\lambda^2}(L, r)$	$L \in \{1, 2, 3, \ldots, 8\}$, $r \in \{10, 20, 50, 100, 200\}$	
neighbor	$k_n \in \{1,2,3\} \cup \{4, 8, 12, \ldots, 4 \cdot \lfloor \frac{n_l}{4} \rfloor\}$	

Tabelle 4.7.: Anwendung III: Wahl der Hyperparameter, wobei die Funktion $\eta_n^{(j)}$ den Kleinste-Quadrate-Schätzer des Plug-In Klassifikators $f_n^{(j)}$ $(j = 2, 7, 8, 9)$ bezeichnet.

Platzierung des dritten Objekts verfahren wir analog, bis das dritte Objekt nur noch maximal fünf Prozent der Fläche des ersten bzw. zweiten Objekts überdeckt. Anschließend verwenden wir wieder das Python-Paket *Pillow*, um das kontinuierliche Bild auf dem Gitter $\{1, \ldots, \lambda\}^2$ zu diskretisieren.

Wir betrachten wieder den Fall $n = 200$ und den Fall $n = 400$ und beurteilen die Klassifikatoren anhand eines Datensatzes der Größe $N = 10^5$ gemäß (4.1). Wie in Abschnitt 4.1 berechnen wir den Median und Interquartilabstand (IQR) des empirischen Missklassifikationsrisikos von 25 Durchläufen. Die Ergebnisse sind in Tabelle 4.8 zusammengefasst. Es ergibt sich, dass die beiden Klassifikatoren $f_n^{(8)}$ und $f_n^{(9)}$ besser abschneiden als die beiden auf faltenden neuronalen Netzen basierenden Klassifikatoren, die keine zusätzlichen geteilten Gewichte enthalten, was Bemerkung 3.7 bekräftigt. In drei von vier Fällen schneidet der Klassifikator $f_n^{(9)}$ am besten ab. Die Klassifikatoren $g_{n,net}$ und *neighbor* sind nicht in der Lage, zufriedenstellende Ergebnisse zu erzielen, da die Fehler dieser Schätzungen in etwa dem erwarteten Fehler eines Klassifikators entsprechen, der immer die gleiche Klasse schätzt. Wir beobachten auch, dass eine größere Auflösung zu einer besseren Performanz führt, was darauf hindeutet, dass der Fehlerterm ϵ_λ aus Annahme 2 für große Auflösungen λ klein ist.

Klassifikationsproblem 8: Enthält das Bild eine Vier oder eine Neun?

In unserer zweiten Anwendung testen wir unsere Bildklassifikatoren an realen Bildern. Hier verwenden wir die Klassen „Vier“ und „Neun“ des MNIST-rot Datensatzes (Larochelle et al. (2007)), der Bilder von handgeschriebenen Ziffern enthält. Die Ziffern wurden zufällig um Winkel des Intervalls $[0, 2\pi)$ rotiert (siehe Abbildung 4.9).

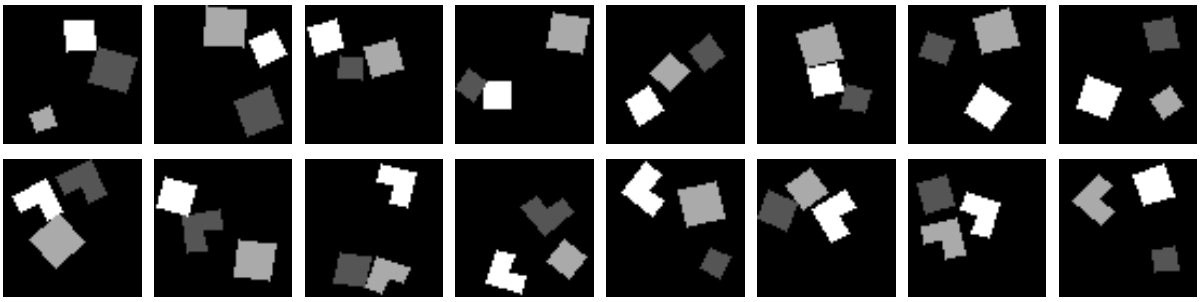


Abbildung 4.8.: Klassifikationsproblem 7: Einige Realisierungen der Zufallsvariable $\mathbf{X}$, wobei die erste Zeile Bilder der Klasse 0 und die untere Zeile Bilder der Klasse 1 zeigt.

Auflösung	$\lambda = 32$		$\lambda = 64$	
Stichprobenumfang	$n = 200$	$n = 400$	$n = 200$	$n = 400$
Ansatz	Median (IQR)	Median (IQR)	Median (IQR)	Median (IQR)
$f_n^{(2)}$	0.4200 (0.0131)	0.2582 (0.0898)	0.3993 (0.0874)	0.2765 (0.1557)
$f_n^{(7)}$	0.3939 (0.0767)	0.1851 (0.0602)	0.1985 (0.1731)	0.0715 (0.0378)
$f_n^{(8)}$	0.1267 (0.0576)	0.0597 (0.0250)	**0.0428** (0.0186)	0.0194 (0.0141)
$f_n^{(9)}$	**0.0884** (0.0545)	**0.0467** (0.0460)	0.0545 (0.0279)	**0.0130** (0.0225)
$g_{n,net}$	0.4867 (0.0071)	0.4869 (0.0086)	0.4869 (0.0115)	0.4888 (0.0094)
neighbor	0.4913 (0.0090)	0.4928 (0.0077)	0.4907 (0.0111)	0.4900 (0.0115)

Tabelle 4.8.: Klassifikationsproblem 7: Median und Interquartilabstand des empirischen Missklassifikationsrisikos bei 25 Durchläufen.

Der Datensatz von den Bildern der beiden Klassen besteht aus 2.400 Trainingsbildern und $N = 10.000$ Testbildern der Auflösung $\lambda = 28$. Aus den 2.400 Trainingsbildern wählen wir zufällig $n/2$ Trainingsbilder pro Klasse aus und beurteilen unsere Klassifikatoren anhand der N Testbilder. Tabelle 4.9 zeigt den Median und Interquartilabstand (IQR) des empirischen Missklassifikationsrisikos von 25 Durchläufen. Auch hier ergibt sich, dass der Klassifikator $f_n^{(9)}$ die anderen Klassifikatoren übertrifft und der Klassifikator $f_n^{(8)}$ am zweitbesten abschneidet. Im Gegensatz zur Anwendung mit synthetischen Bildern aus dem siebten Klassifikationsproblem, schneidet der Klassifikator $f_n^{(2)}$ relativ zu den anderen Klassifikatoren diesmal deutlich besser ab. Der Klassifikator $f_n^{(7)}$ schneidet ungefähr so gut wie die beiden alternativen Ansätze $g_{n,net}$ und *neighbor* ab.

4.4. Einordnung der Anwendungsergebnisse

Das vorliegende Kapitel hatte zum Ziel, die praktische Relevanz unserer Bildklassifikatoren mit den speziellen Netzwerkarchitekturen aufzuzeigen, indem das Verhalten der Bildklassifikatoren bei endlichem Stichprobenumfang untersucht wurde. In Abschnitt 4.1 haben wir gesehen, dass unser Bildklassifikator, der weder lokale Pooling Schichten enthält noch auf den Aspekt der Rotation von Objekten ausgerichtet ist, den alternativen Klassifizierungsmethoden bei den betrachteten Klassifikationsprobemen überlegen ist. Insbesondere bei den Klassifikationsproblemen 1 und 2, bei denen simulierte Bilder verwendet wurden, waren die alternativen Ansätze im Gegensatz zu dem in dieser Arbeit vorgestellten Klassifikator nicht in der Lage, zufriedenstellende Ergebnisse zu erzielen.

In Abschnitt 4.2 haben wir die experimentelle Nützlichkeit lokaler Pooling Schichten in unseren speziellen Netzwerkarchitekturen in den Klassifikationsproblemen 4 und 6 nachgewiesen. Klassifikationsproblem 5

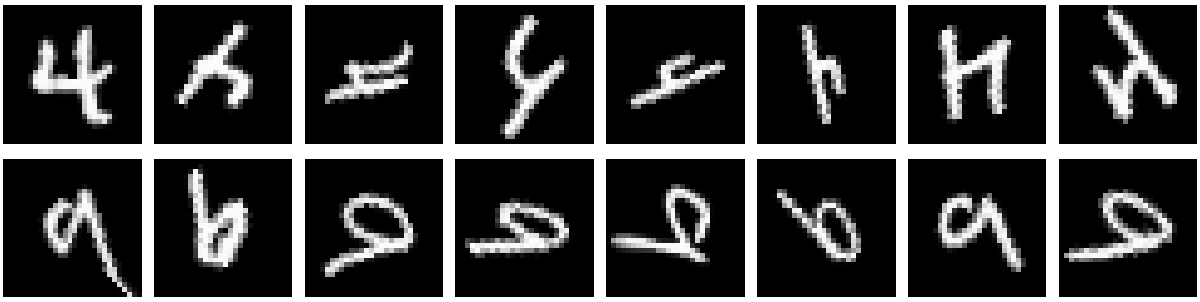


Abbildung 4.9.: Klassifikationsproblem 8: Die erste Zeile zeigt einige Bilder der Vierer und die untere Zeile Bilder der Neuner aus dem MNIST-rot Datensatz.

Auflösung	$\lambda = 28$	
Stichprobenumfang	$n = 200$	$n = 400$
Ansatz	Median (IQR)	Median (IQR)
$f_n^{(2)}$	0.2018 (0.0729)	0.1325 (0.0403)
$f_n^{(7)}$	0.3190 (0.0380)	0.2144 (0.0841)
$f_n^{(8)}$	0.1646 (0.0532)	0.1249 (0.0310)
$f_n^{(9)}$	**0.1296** (0.1149)	**0.0850** (0.0257)
$g_{n,net}$	0.2979 (0.0276)	0.2162 (0.0221)
neighbor	0.2772 (0.0291)	0.2136 (0.0125)

Tabelle 4.9.: Klassifikationsproblem 8: Median und Interquartilabstand des empirischen Missklassifikationsrisikos bei 25 Durchläufen.

erschien für den verwendeten Stichprobenumfang und für die verwendeten Klassifikatoren zu schwierig zu sein. Hier konnte die Nützlichkeit der lokalen Pooling Schichten in unseren Architekturen nicht experimentell nachgewiesen werden.

Darüber hinaus konnten wir durch unsere theoretische Analyse speziell bei der Klassifizierung rotierter Objekte in Abschnitt 4.3 Hinweise auf sinnvolle komplexere Netzwerkarchitekturen geben, die wir implementiert haben und die bei den betrachteten Klassifikationsproblemen 7 und 8 zu besseren Ergebnissen geführt haben. Dies deutet darauf hin, dass die theoretische Analyse auch zu Verbesserung der in der Praxis verwendeten Verfahren führen kann.

Insgesamt haben wir also die praktische Relevanz der in dieser Arbeit eingeführten Bildklassifikatoren gezeigt, was unsere theoretischen Ergebnisse zusätzlich stützt. Es ist zu beachten, dass unklar bleibt, inwieweit die verwendeten simulierten und realen Bilddatensätze tatsächlich unseren statistischen Modellen für die Bildklassifikation genügen.

A. Anhang: Ausgegliederte Beweise und die gewichtete AM-GM Ungleichung

Der Anhang enthält die Beweise von Lemma 13 und Lemma 14 sowie den Beweis von Ungleichung (2.10) aus Beispiel 2.1. Außerdem ist am Ende des Anhangs die gewichtete AM-GM Ungleichung angeführt, in der Form, in der sie im Beweis von Lemma 20 verwendet wurde.

A.1. Beweis von Lemma 13 und Lemma 14

Beweis von Lemma 13. Wegen Ungleichung (3.27) genügt es zu zeigen, dass

$$\begin{aligned}&\max_{i\in\{1,\dots,t\}}\max_{\mathbf{u}\in\{1,\dots,\lambda\}^2\,:\,\mathbf{u}+I^{(l)}\subseteq\{1,\dots,\lambda\}^2}\left|f_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}})-\bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}})\right|\\&\leq(C+1)^l\cdot\max_{\substack{i\in\{1,\dots,t\},j\in\{1,\dots,4^l\},\\k\in\{1,\dots,l\},s\in\{1,\dots,4^{l-k}\}}}\left\{\|g_{0,j}^{(i)}-\bar{g}_{0,j}^{(i)}\|_{[0,1],\infty},\|g_{k,s}^{(i)}-\bar{g}_{k,s}^{(i)}\|_{[0,2]^4,\infty}\right\}.\end{aligned}$$

Dies folgt aus

$$\begin{aligned}&\left|f_{k,s}^{(i)}(\mathbf{x})-\bar{f}_{k,s}^{(i)}(\mathbf{x})\right|\\&\leq(C+1)^k\cdot\max_{m\in\{1,\dots,k\},n\in\{1,\dots,4^{l-m}\},j\in\{1,\dots,4^l\}}\left\{\|g_{0,j}^{(i)}-\bar{g}_{0,j}^{(i)}\|_{[0,1],\infty},\|g_{m,n}^{(i)}-\bar{g}_{m,n}^{(i)}\|_{[0,2]^4,\infty}\right\}\end{aligned}\tag{A.1}$$

für alle $\mathbf{x}\in[0,1]^{I^{(k)}}$, $i\in\{1,\dots,t\}$,$k\in\{0,\dots,l\}$ und $s\in\{1,\dots,4^{l-k}\}$, was wir durch Induktion über k zeigen.

Für $k=0$, $s\in\{1,\dots,4^l\}$ und $i\in\{1,\dots,t\}$ gilt

$$\left|f_{0,s}^{(i)}(x)-\bar{f}_{0,s}^{(i)}(x)\right|=\left|g_{0,s}^{(i)}(x)-\bar{g}_{0,s}^{(i)}(x)\right|\leq\left\|g_{0,s}^{(i)}-\bar{g}_{0,s}^{(i)}\right\|_{[0,1],\infty}$$

für alle $x\in[0,1]$. Wir nehmen nun an, Gleichung (A.1) gelte für ein $k\in\{0,\dots,l-1\}$. Aufgrund der Definition von $\bar{f}_{k,s}^{(i)}$ ist

$$0\leq\bar{f}_{k,s}^{(i)}(\mathbf{x})\leq 2$$

für alle $\mathbf{x}\in[0,1]^{I^{(k)}}$, $i\in\{1,\dots,t\}$, $k\in\{0,\dots,l-1\}$ und $s\in\{1,\dots,4^{l-k}\}$. Die Dreiecksungleichung und die Lipschitzbedingung an die Funktion $g_{k+1,s}^{(i)}\big|_{[0,2]^4}$ implizieren

$$\begin{aligned}&\left|f_{k+1,s}^{(i)}(\mathbf{x})-\bar{f}_{k+1,s}^{(i)}(\mathbf{x})\right|\\&\leq\Big|g_{k+1,s}^{(i)}\Big(f_{k,4\cdot(s-1)+1}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+1}^{(i)}+I^{(k)}}),f_{k,4\cdot(s-1)+2}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+2}^{(i)}+I^{(k)}}),\\&\qquad f_{k,4\cdot(s-1)+3}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot(s-1)+3}^{(i)}+I^{(k)}}),f_{k,4\cdot s}^{(i)}(\mathbf{x}_{\mathbf{i}_{k,4\cdot s}^{(i)}+I^{(k)}})\Big)\end{aligned}$$

$$
\begin{aligned}
&- g^{(i)}_{k+1,s}\Big(\bar{f}^{(i)}_{k,4\cdot(s-1)+1}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot(s-1)+1}+I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot(s-1)+2}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot(s-1)+2}+I^{(k)}}),\\
&\qquad \bar{f}^{(i)}_{k,4\cdot(s-1)+3}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot(s-1)+3}+I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot s}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot s}+I^{(k)}})\Big)\Big|\\
&+ \Big|g^{(i)}_{k+1,s}\Big(\bar{f}^{(i)}_{k,4\cdot(s-1)+1}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot(s-1)+1}+I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot(s-1)+2}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot(s-1)+2}+I^{(k)}}),\\
&\qquad \bar{f}^{(i)}_{k,4\cdot(s-1)+3}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot(s-1)+3}+I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot s}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot s}+I^{(k)}})\Big)\\
&- \bar{g}^{(i)}_{k+1,s}\Big(\bar{f}^{(i)}_{k,4\cdot(s-1)+1}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot(s-1)+1}+I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot(s-1)+2}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot(s-1)+2}+I^{(k)}}),\\
&\qquad \bar{f}^{(i)}_{k,4\cdot(s-1)+3}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot(s-1)+3}+I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot s}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot s}+I^{(k)}})\Big)\Big|\\
&\leq C \cdot \max_{j\in\{1,\dots,4\}} \left|f^{(i)}_{k,4\cdot(s-1)+j}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot(s-1)+j}+I^{(k)}}) - \bar{f}^{(i)}_{k,4\cdot(s-1)+j}(\mathbf{x}_{\mathbf{i}^{(i)}_{k,4\cdot(s-1)+j}+I^{(k)}})\right|\\
&\quad + \|g^{(i)}_{k+1,s} - \bar{g}^{(i)}_{k+1,s}\|_{[0,2]^4,\infty}\\
&\leq C\cdot(C+1)^k \cdot \max_{m\in\{1,\dots,k\},n\in\{1,\dots,4^{l-m}\},j\in\{1,\dots,4^l\}} \left\{\|g^{(i)}_{0,j} - \bar{g}^{(i)}_{0,j}\|_{[0,1],\infty}, \|g^{(i)}_{m,n} - \bar{g}^{(i)}_{m,n}\|_{[0,2]^4,\infty}\right\}\\
&\quad + \|g^{(i)}_{k+1,n} - \bar{g}^{(i)}_{k+1,n}\|_{[0,2]^4,\infty}\\
&\leq (C+1)^{k+1} \cdot \max_{m\in\{1,\dots,k+1\},n\in\{1,\dots,4^{l-m}\},j\in\{1,\dots,4^l\}} \left\{\|g^{(i)}_{0,j} - \bar{g}^{(i)}_{0,j}\|_{[0,1],\infty}, \|g^{(i)}_{m,n} - \bar{g}^{(i)}_{m,n}\|_{[0,2]^4,\infty}\right\}
\end{aligned}
$$

für alle $\mathbf{x} \in [0,1]^{I^{(k+1)}}$, $i \in \{1,\dots,t\}$ und $s \in \{1,\dots,4^{l-(k+1)}\}$. □

Um Lemma 14 zu beweisen, verwenden wir die folgenden zwei Hilfsergebnisse. Das erste Hilfsresultat lässt uns das Maximum von endlich vielen reellen Zahlen durch ein vollverbundenes neuronales Netz berechnen. Ein ähnliches Resultat mit einer größeren Anzahl an Neuronen pro Schicht findet sich beispielsweise in Arora et al. (2018).

Lemma 22. *Es sei $t \in \mathbb{N}$, $r_{net} = 3\cdot t$,*

$$
L_{net} = \begin{cases} \lceil \log_2 t \rceil & \text{, falls } t > 1 \\ 1 & \text{, falls } t = 1 \end{cases}
$$

und es sei $\mathcal{G}_t(L_{net}, r_{net})$ definiert wie in Gleichung (3.8)*. Dann existiert ein $g_{net} \in \mathcal{G}_t(L_{net}, r_{net})$, sodass*

$$
g_{net}(\mathbf{x}) = \max\{x_1,\dots,x_t\}
$$

für alle $\mathbf{x} = (x_1,\dots,x_t) \in \mathbb{R}^t$.

Beweis. O.B.d.A können wir annehmen, dass $t > 1$, da wir für $t = 1$ das neuronale Netz $g_{net} \in \mathcal{G}_1(1,3)$ durch

$$
g_{net}(x) = \sigma(x) - \sigma(-x) = \max\{x,0\} - \max\{-x,0\} = x
$$

definieren können. Im Beweis verwenden wir das Netzwerk $g_{max} : \mathbb{R}^2 \to \mathbb{R}$, welches durch

$$
g_{max}(x_1,x_2) = \sigma(x_2 - x_1) + \sigma(x_1) - \sigma(-x_1) \quad (x_1,x_2 \in \mathbb{R})
$$

definiert ist und

$$
g_{max}(x_1,x_2) = \max\{x_2 - x_1, 0\} + \underbrace{\max\{x_1,0\} - \max\{-x_1,0\}}_{=x_1} = \max\{x_1,x_2\}
$$

für alle $x_1, x_2 \in \mathbb{R}$ erfüllt. Für $t \in \mathbb{N} \setminus \{1\}$ setzen wir

$$r(t) = 3 \cdot 2^{\lceil \log_2(t) \rceil - 1} \quad \text{sowie} \quad L(t) = \lceil \log_2 t \rceil$$

und zeigen die Behauptung, indem wir folgende stärkere Aussage verifizieren: Für alle $t \in \mathbb{N} \setminus \{1\}$ existiert ein

$$g_{net} \in \mathcal{G}_t(L_{net}, r(t)) \overset{r(t)<r_{net}}{\subset} \mathcal{G}_t(L_{net}, r_{net}),$$

sodass

$$g_{net}(\mathbf{x}) = \max\{x_1, \ldots, x_t\}$$

für alle $\mathbf{x} \in \mathbb{R}^t$. Wir weisen dies durch Induktion über t nach.

Für $t = 2$ folgt die Behauptung, indem wir das Netzwerk g_{max} verwenden. Nun nehmen wir an, es sei $t > 2$ und die Behauptung gelte für alle natürlichen Zahlen kleiner als t und größer als 1. Dann existiert ein $g \in \mathcal{G}_{\lceil t/2 \rceil}(L(\lceil t/2 \rceil), r(\lceil t/2 \rceil))$, sodass

$$g(\mathbf{x}) = \max\{x_1, \ldots, x_{\lceil t/2 \rceil}\}$$

für alle $\mathbf{x} \in \mathbb{R}^{\lceil t/2 \rceil}$. Wir definieren $g_{net} \in \mathcal{G}_t(L(\lceil t/2 \rceil) + 1, 2 \cdot r(\lceil t/2 \rceil))$ durch

$$g_{net}(\mathbf{x}) = g_{max}(g(x_1, \ldots, x_{\lceil t/2 \rceil}), g(x_{\lfloor t/2 \rfloor + 1}, \ldots, x_t)) = \max\{x_1, \ldots, x_t\}.$$

Es genügt nun zu zeigen, dass

$$L(t) = L(\lceil t/2 \rceil) + 1 \quad \text{und} \quad r(t) = 2 \cdot r(\lceil t/2 \rceil)).$$

Da $2^k < t \leq 2^{k+1}$ für ein $k \in \mathbb{N}$ gilt

$$\lceil \log_2(2 \cdot \lceil t/2 \rceil) \rceil \geq \lceil \log_2(t) \rceil = k + 1 = \left\lceil \log_2\left(2 \cdot 2^k\right) \right\rceil \geq \lceil \log_2(2 \cdot \lceil t/2 \rceil) \rceil,$$

was

$$\lceil \log_2(2 \cdot \lceil t/2 \rceil) \rceil = \lceil \log_2(t) \rceil \tag{A.2}$$

impliziert. Unter Verwendung von Gleichung (A.2) resultiert

$$L(\lceil t/2 \rceil) + 1 = \lceil \log_2 \lceil t/2 \rceil \rceil + 1 = \lceil \log_2(2 \cdot \lceil t/2 \rceil) \rceil = \lceil \log_2(t) \rceil = L(t)$$

und

$$\begin{aligned} 2 \cdot r(\lceil t/2 \rceil)) &= 2 \cdot 3 \cdot 2^{\lceil \log_2(\lceil t/2 \rceil) \rceil - 1} \\ &= 3 \cdot 2^{\lceil \log_2(\lceil t/2 \rceil) \rceil + 1 - 1} \\ &= 3 \cdot 2^{\lceil \log_2(2 \cdot \lceil t/2 \rceil) \rceil - 1} \\ &= 3 \cdot 2^{\lceil \log_2(t) \rceil - 1} \\ &= r(t). \end{aligned}$$

□

Das zweite Hilfsresultat ermöglicht es uns, die vollverbundenen neuronalen Netze $\sigma \circ g_{net,m,s}^{(i)}$ aus Lemma 14 in einem faltenden neuronalen Netz zu berechnen. Da die Eingangsdimension der neuronalen Netze $d = 1$ für $m = 0$ und $d = 4$ für $m \in \{1, \dots, l\}$ ist, betrachten wir den allgemeinen Fall $d \in \mathbb{N}$. Lemma 9 stellt nur eine leichte Modifikation von Lemma 5 dar, weswegen es, wie auch der Beweis von Lemma 14, in den Anhang ausgegliedert wurde. Da wir in der Netzwerkarchitektur in Theorem 3.3 ein symmetrisches Zero-Padding verwenden, verallgemeinern wir den Begriff eines faltenden Blocks aus Gleichung (3.29) auf ein beliebiges Zero-Padding: Für $i_1, i_2, k' \in \mathbb{N}$ und eine Indexmenge $I = \{1, \dots, i_1\} \times \{1, \dots, i_2\}$ definieren wir die Funktion

$$o_{(k',k),M,P}^{(z)} : \mathbb{R}^{I \times \{1,\dots,k'\}} \to \mathbb{R}_0^{+I \times \{1,\dots,k\}},$$

durch

$$o_{(k',k),M,P}^{(z)}(\mathbf{x}) = \big(o_{(k,k),M,P,\mathbf{w}_z} \circ \cdots \circ o_{(k',k),M,P,\mathbf{w}_1}\big)(\mathbf{x}) \quad \big(\mathbf{x} \in \mathbb{R}^{I \times \{1,\dots,k'\}}\big).$$

Lemma 23. *Es sei $d \in \mathbb{N}$ und $g_{net} \in \mathcal{G}_d(L_{net}, r_{net})$ für $L_{net}, r_{net} \in \mathbb{N}$ (siehe Abschnitt 3.1 für die Definition von $\mathcal{G}_d(L_{net}, r_{net})$). Sei*

$$f : \mathbb{R}^{\{1,\dots,\lambda\}^2} \to \mathbb{R}^{\{1,\dots,\lambda\}^2 \times \{1,\dots,k'\}}$$

eine Funktion, wobei $\lambda, k' \in \mathbb{N}$. Außerdem sei $t \in \mathbb{N}$, $m \in \mathbb{N}_0$, $M = I_{\{m>0\}} \cdot 2^m + 3$ und seien

$$(i_1, j_1), \dots, (i_d, j_d) \in \{-\lfloor 2^{m-1} + 1 \rfloor, \dots, 0, \dots, \lfloor 2^{m-1} + 1 \rfloor\}^2$$

sowie $s_0 \in \{1, \dots, t\}$ und $s_1, \dots, s_d \in \{1, \dots, k'\}$. Dann existiert ein faltender Block

$$o_{(k',t+r_{net}),M,\lceil M/2 \rceil}^{(L_{net}+1)} : \mathbb{R}^{I \times \{1,\dots,k'\}} \to \mathbb{R}^{I \times \{1,\dots,t+r_{net}\}},$$

welcher wie in Gleichung (3.29) definiert ist und beliebige Gewichte in den Kanälen $\{1, \dots, t\} \setminus \{s_0\}$ besitzt, sodass

$$\begin{aligned}
&\Big(\big(o_{(k',t+r_{net}),M,\lceil M/2 \rceil}^{(L_{net}+1)} \circ f\big)(\mathbf{x})\Big)_{(i',j'),s_0} \\
&= \sigma\Big(g_{net}\Big(\big(f(\mathbf{x})\big)_{(i'+i_1,j'+j_1),s_1}, \big(f(\mathbf{x})\big)_{(i'+i_2,j'+j_2),s_2}, \dots, \big(f(\mathbf{x})\big)_{(i'+i_d,j'+j_d),s_d}\Big)\Big)
\end{aligned}$$

für alle $(i', j') \in \{1, \dots, \lambda\}^2$, wobei wir die Schreibweise $\big(f(\mathbf{x})\big)_{(i',j'),s} = 0$ für $(i', j') \notin \{1, \dots, \lambda\}^2$ verwenden.

Beweis. Wir nehmen an, das vollverbundene neuronale Netz g_{net} sei gegeben durch

$$g_{net}(\mathbf{x}) = \sum_{i=1}^{r_{net}} w_{1,i}^{(L_{net})} g_i^{(L_{net})}(\mathbf{x}) + w_{1,0}^{(L_{net})},$$

wobei $g_i^{(L_{net})}$ rekursiv definiert ist durch

$$g_i^{(r)}(\mathbf{x}) = \sigma\left(\sum_{j=1}^{r_{net}} w_{i,j}^{(r-1)} g_j^{(r-1)}(\mathbf{x}) + w_{i,0}^{(r-1)}\right)$$

für $i \in \{1, \dots, r_{net}\}$, $r \in \{2, \dots, L_{net}\}$, und

$$g_i^{(1)}(\mathbf{x}) = \sigma\left(\sum_{j=1}^{d} w_{i,j}^{(0)} x^{(j)} + w_{i,0}^{(0)}\right) \quad (i \in \{1, \dots, r_{net}\}).$$

O.B.d.A können wir annehmen, dass $(s_{n_1}, i_{n_1}, j_{n_1}) \neq (s_{n_2}, i_{n_2}, j_{n_2})$ für alle $n_1, n_2 \in \{1, \dots, d\}$ mit $n_1 \neq n_2$ gilt (ansonsten können wir die Behauptung für ein entsprechend definiertes $g'_{net} \in \mathcal{G}_{d'}(L_{net}, r_{net})$ mit $d' < d$ zeigen). Da $M = 2 \cdot \lfloor 2^{m-1} \rfloor + 3$ und $\lceil M/2 \rceil = \lfloor 2^{m-1} \rfloor + 2$ gilt

$$\begin{aligned}
&\Big(\big(o_{(k',t+r_{net}),M,\lceil M/2\rceil,\mathbf{w}_1} \circ f\big)(\mathbf{x})\Big)_{(i',j'),t+i} \\
&= \sigma\left(\sum_{s=1}^{k'} \sum_{\substack{t_1,t_2 \in \{1,\dots,M\} \\ i'+t_1-\lceil M/2\rceil \in \{1,\dots,\lambda\} \\ j'+t_2-\lceil M/2\rceil \in \{1,\dots,\lambda\}}} w^{(1)}_{t_1,t_2,s,t+i} \cdot (f(\mathbf{x}))_{(i'+t_1-\lceil M/2\rceil, j'+t_2-\lceil M/2\rceil),s} + w^{(1)}_{t+i}\right) \\
&= \sigma\left(\sum_{s=1}^{k'} \sum_{\substack{t_1,t_2 \in \{-\lfloor 2^{m-1}+1\rfloor,\dots,\lfloor 2^{m-1}+1\rfloor\} \\ (i'+t_1,j'+t_2) \in \{1,\dots,\lambda\}^2}} w^{(1)}_{\lfloor 2^{m-1}\rfloor+2+t_1,\lfloor 2^{m-1}\rfloor+2+t_2,s,t+i} \cdot (f(\mathbf{x}))_{(i'+t_1,j'+t_2),s} + w^{(1)}_{t+i}\right)
\end{aligned} \tag{A.3}$$

für alle $i \in \{1, \dots, r_{net}\}$ und $(i', j') \in \{1, \dots, \lambda\}^2$. Das Ziel ist es, die Gewichte in (A.3) so zu wählen, dass

$$\begin{aligned}
&\Big(\big(o_{(k',t+r_{net}),M,\lceil M/2\rceil,\mathbf{w}_1} \circ f\big)(\mathbf{x})\Big)_{(i',j'),t+i} \\
&= \sigma\left(\sum_{n=1}^{d} w^{(0)}_{i,n} \cdot (f(\mathbf{x}))_{(i'+i_n,j'+j_n),s_n} + w^{(0)}_{i,0}\right) \\
&= g^{(1)}_i\Big(\,(f(\mathbf{x}))_{(i'+i_1,j'+j_1),s_1}, (f(\mathbf{x}))_{(i'+i_2,j'+j_2),s_2}, \dots, (f(\mathbf{x}))_{(i'+i_d,j'+j_d),s_d}\,\Big)
\end{aligned}$$

für alle $i \in \{1, \dots, r_{net}\}$ und $(i', j') \in \{1, \dots, \lambda\}^2$. Deshalb wählen wir in den Kanälen $t+1, \dots, t+r_{net}$ die einzigen Gewichte, welche ungleich Null sind durch

$$w^{(1)}_{\lfloor 2^{m-1}\rfloor+2+i_n,\lfloor 2^{m-1}\rfloor+2+j_n,s_n,t+i} = w^{(0)}_{i,n} \quad \text{und} \quad w^{(1)}_{t+i} = w^{(0)}_{i,0}$$

für $n \in \{1, \dots, d\}$ sowie $i \in \{1, \dots, r_{net}\}$ und erhalten

$$\begin{aligned}
&\Big(\big(o_{(k',t+r_{net}),M,\lceil M/2\rceil,\mathbf{w}_1} \circ f\big)(\mathbf{x})\Big)_{(i',j'),t+i} \\
&= \sigma\left(\sum_{n=1}^{d} w^{(0)}_{i,n} \cdot (f(\mathbf{x}))_{(i'+i_n,j'+j_n),s_n} + w^{(0)}_{i,0}\right) \\
&= g^{(1)}_i\Big(\,(f(\mathbf{x}))_{(i'+i_1,j'+j_1),s_1}, (f(\mathbf{x}))_{(i'+i_2,j'+j_2),s_2}, \dots, (f(\mathbf{x}))_{(i'+i_d,j'+j_d),s_d}\,\Big)
\end{aligned} \tag{A.4}$$

für alle $i \in \{1, \dots, r_{net}\}$ und $(i', j') \in \{1, \dots, \lambda\}^2$. Im Folgenden verwenden wir die Schreibweise

$$o^{(r)} = o_{(t+r_{net},t+r_{net}),M,\lceil M/2\rceil,\mathbf{w}_r} \circ \cdots \circ o_{(k',t+r_{net}),M,\lceil M/2\rceil,\mathbf{w}_1} \quad (r = 1, \dots, L_{net}+1).$$

In den Schichten $r = 2, \dots, L_{net}$ ergibt sich

$$\begin{aligned}\Big(\big(o^{(r)} \circ f\big)(\mathbf{x})\Big)_{(i',j'),t+i} &= \sigma\Bigg(\sum_{s=1}^{t+r_{net}} \sum_{\substack{t_1,t_2\in\{1,\dots,M\} \\ i'+t_1-\lceil M/2\rceil\in\{1,\dots,\lambda\} \\ j'+t_2-\lceil M/2\rceil\in\{1,\dots,\lambda\}}} \\ &\qquad w^{(r)}_{t_1,t_2,s,t+i} \cdot \Big(\big(o^{(r-1)} \circ f\big)(\mathbf{x})\Big)_{(i'+t_1-\lceil M/2\rceil,j'+t_2-\lceil M/2\rceil),s} + w^{(r)}_{t+i} \Bigg) \\ &= \sigma\Bigg(\sum_{s=1}^{t+r_{net}} \sum_{\substack{t_1,t_2\in\{1-\lceil M/2\rceil,\dots,M-\lceil M/2\rceil\} \\ (i'+t_1,j'+t_2)\in\{1,\dots,\lambda\}^2}} \\ &\qquad w^{(r)}_{\lceil M/2\rceil+t_1,\lceil M/2\rceil+t_2,s,t+i} \cdot \Big(\big(o^{(r-1)} \circ f\big)(\mathbf{x})\Big)_{(i'+t_1,j'+t_2),s} + w^{(r)}_{t+i} \Bigg)\end{aligned}$$

für $r \in \{2, \dots, L_{net}\}$, $i \in \{1, \dots, r_{net}\}$ und $(i', j') \in \{1, \dots, \lambda\}^2$. Hier ist das Ziel die Gewichte so zu wählen, dass

$$\Big(\big(o^{(r)} \circ f\big)(\mathbf{x})\Big)_{(i',j'),t+i} = \sigma\Bigg(\sum_{j=1}^{r_{net}} w^{(r-1)}_{i,j} \cdot \Big(\big(o^{(r-1)} \circ f\big)(\mathbf{x})\Big)_{(i',j'),t+j} + w^{(r-1)}_{i,0} \Bigg) \tag{A.5}$$

für alle $r \in \{2, \dots, L_{net}\}$, $i \in \{1, \dots, r_{net}\}$ und $(i', j') \in \{1, \dots, \lambda\}^2$. Zu diesem Zweck wählen wir in den Kanälen $t + 1, \dots, t + r_{net}$ die einzigen Gewichte ungleich Null durch

$$w^{(r)}_{\lceil M/2\rceil,\lceil M/2\rceil,t+j,t+i} = w^{(r-1)}_{i,j} \quad \text{und} \quad w^{(r)}_{t+i} = w^{(r-1)}_{i,0}$$

für $r \in \{2, \dots, L_{net}\}$, $i \in \{1, \dots, r_{net}\}$ und $j \in \{1, \dots, r_{net}\}$, was Gleichung (A.5) impliziert. In Schicht $L_{net} + 1$ berechnen wir

$$\begin{aligned}\Big(\big(o^{(L_{net}+1)} \circ f\big)(\mathbf{x})\Big)_{(i',j'),s_0} &= \sigma\Bigg(\sum_{s=1}^{t+r_{net}} \sum_{\substack{t_1,t_2\in\{1-\lceil M/2\rceil,\dots,M-\lceil M/2\rceil\} \\ (i'+t_1,j'+t_2)\in\{1,\dots,\lambda\}^2}} \\ &\qquad w^{(L_{net}+1)}_{\lceil M/2\rceil+t_1,\lceil M/2\rceil+t_2,s,s_0} \cdot \Big(\big(o^{(L_{net})} \circ f\big)(\mathbf{x})\Big)_{(i'+t_1,j'+t_2),s} + w^{(L_{net}+1)}_{s_0} \Bigg)\end{aligned}$$

für $(i', j') \in \{1, \dots, \lambda\}^2$ und wollen die Gewichte so wählen, dass

$$\Big(\big(o^{(L_{net}+1)} \circ f\big)(\mathbf{x})\Big)_{(i',j'),s_0} = \sigma\Bigg(\sum_{i=1}^{r_{net}} w^{(L_{net})}_{1,i} \cdot \Big(\big(o^{(L_{net})} \circ f\big)(\mathbf{x})\Big)_{(i',j'),t+i} + w^{(L_{net})}_{1,0} \Bigg) \tag{A.6}$$

für alle $(i', j') \in \{1, \dots, \lambda\}^2$. Hierfür wählen wir im Kanal s_0 die einzigen Gewichte ungleich Null durch

$$w^{(L_{net}+1)}_{\lceil M/2\rceil,\lceil M/2\rceil,t+i,s_0} = w^{(L_{net})}_{1,i} \quad \text{und} \quad w^{(L_{net}+1)}_{s_0} = w^{(L_{net})}_{1,0}$$

für $i \in \{1, \dots, r_{net}\}$, was Gleichung (A.6) impliziert. Kombinieren wir die Gleichungen (A.4), (A.5) und (A.6), liefert das die Behauptung. □

Beweis von Lemma 14. Wie im Beweis von Lemma 4 verwenden wir, dass wir durch eine faltende Schicht der Form (3.31) nichtnegative Werte an die nächste Schicht weitergeben können (da wir in Lemma 14 ein symmetrisches Zero-Padding verwenden, müssen die Gewichte hierfür entsprechend angepasst werden).

Zunächst sei $g_{max} \in \mathcal{G}_t(L_t, r_t)$ das neuronale Netz aus Lemma 22, sodass

$$\begin{aligned}\bar{\eta}(\mathbf{x}) &= \max_{\mathbf{u}\in\{1,\dots,\lambda\}^2 \,:\, \mathbf{u}+I^{(l)}\subseteq\{1,\dots,\lambda\}^2} \max_{i\in\{1,\dots,t\}} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) \\ &= \max_{i\in\{1,\dots,t\}} \max_{\mathbf{u}\in\{1,\dots,\lambda\}^2 \,:\, \mathbf{u}+I^{(l)}\subseteq\{1,\dots,\lambda\}^2} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) \\ &= g_{\max}\left(\max_{\mathbf{u}\in\{1,\dots,\lambda\}^2 \,:\, \mathbf{u}+I^{(l)}\subseteq\{1,\dots,\lambda\}^2} \bar{f}_{l,1}^{(1)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}), \dots, \max_{\mathbf{u}\in\{1,\dots,\lambda\}^2 \,:\, \mathbf{u}+I^{(l)}\subseteq\{1,\dots,\lambda\}^2} \bar{f}_{l,1}^{(t)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) \right)\end{aligned}$$

für alle $\mathbf{x} \in [0,1]^{\{1,\dots,\lambda\}^2}$. Wegen der Definition der Funktionsklasse $\mathcal{F}_2(\boldsymbol{\theta})$ genügt es zu zeigen, dass für alle $i \in \{1,\dots,t\}$ ein $f_i \in \mathcal{F}_{CNN}(L, k, \mathbf{M}, \mathbf{P}_2, \mathbf{A})$ (siehe Abschnitt 3.1 für die Definition der Funktionsklasse) existiert mit

$$f_i(\mathbf{x}) = \max_{\mathbf{u}\in\{1,\dots,\lambda\}^2 \,:\, \mathbf{u}+I^{(l)}\subseteq\{1,\dots,\lambda\}^2} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) \tag{A.7}$$

für alle $\mathbf{x} \in [0,1]^{\{1,\dots,\lambda\}^2}$. Deshalb sei im restlichen Beweis $i \in \{1,\dots,t\}$ fest. Die Idee ist es, nacheinander die Ausgaben der Funktionen

$$\bar{f}_{0,1}^{(i)}, \dots, \bar{f}_{0,4^l}^{(i)}, \dots, \bar{f}_{m,1}^{(i)}, \dots, \bar{f}_{m,4^{l-m}}^{(i)}, \dots, \bar{f}_{l-1,1}^{(i)}, \dots, \bar{f}_{l-1,4}^{(i)}, \bar{f}_{l,1}^{(i)}$$

aus dem diskretisierten hierarchischen Modell $\bar{f}_{l,1}^{(i)}$ zu berechnen, indem wir die Funktionen $\{\bar{g}_{m,s}^{(i)}\}$ durch wiederholtes Anwenden von Lemma 23 berechnen. Für $m = 0$ verwenden wir Lemma 23 mit $d = 1$ und für $m = 1,\dots,l$ mit $d = 4$. Wir speichern die Ausgaben der Funktionen $\bar{f}_{m,s}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(m)}})$ mithilfe von Gleichung (3.31) in den entsprechenden Kanälen, sodass wir die Ausgaben mehrmals verwenden können. Für die Berechnung des Maximums in Gleichung (A.7) werden wir letztlich die globale Max-Pooling Schicht unserer Netzwerkarchitektur verwenden.

Ein faltendes neuronales Netz $f_i \in \mathcal{F}_{CNN}(L, k, \mathbf{M}, \mathbf{P}_2, \mathbf{A})$ hat die Form

$$f_i(\mathbf{x}) = f_{out}^{(\mathbf{A})} \circ o_{(k,k),M_L,\lceil M_L/2\rceil,\mathbf{w}_L} \circ \cdots \circ o_{(k,k),M_2,\lceil M_2/2\rceil,\mathbf{w}_2} \circ o_{(1,k),M_1,\lceil M_1/2\rceil,\mathbf{w}_1}(\mathbf{x}) \quad \left(\mathbf{x} \in [0,1]^{\{1,\dots,\lambda\}^2}\right)$$

mit den Gewichten

$$\mathbf{w}_1 = \left(\left(w_{i,j,s_1,s_2}^{(1)} \right)_{1\le i,j\le M, s_1\in\{1\}, s_2\in\{1,\dots,k\}}, \left(w_{s_2}^{(1)} \right)_{s_2\in\{1,\dots,k\}} \right),$$

$$\mathbf{w}_r = \left(\left(w_{i,j,s_1,s_2}^{(r)} \right)_{1\le i,j\le M, s_1,s_2\in\{1,\dots,k\}}, \left(w_{s_2}^{(r)} \right)_{s_2\in\{1,\dots,k\}} \right) \quad (r = 2,\dots,L)$$

und

$$\mathbf{w}_{out} = \left(w_s \right)_{s\in\{1,\dots,k\}}.$$

Wir definieren außerdem die Funktion $o^{(r)} : [0,1]^{\{1,\dots,\lambda\}^2} \to \mathbb{R}_0^{+\,\{1,\dots,\lambda\}^2\times\{1,\dots,k\}}$ durch

$$o^{(r)} = o_{(k,k),M_r,\lceil M_r/2\rceil,\mathbf{w}_r} \circ \cdots \circ o_{(k,k),M_2,\lceil M_2/2\rceil,\mathbf{w}_2} \circ o_{(1,k),M_1,\lceil M_1/2\rceil,\mathbf{w}_1}. \quad (r = 1,\dots,L)$$

Im ersten Schritt zeigen wir, wie wir die Gewichtsvektoren $\mathbf{w}_1,\dots,\mathbf{w}_L$ wählen, sodass

$$\left(o^{(L)}(\mathbf{x}) \right)_{(i',j'),1} = \bar{f}_{l,1}^{(i)}(\mathbf{x}_{(i',j')+I^{(l)}}) \tag{A.8}$$

für alle $(i', j') \in \{2^{l-1} + l, \ldots, \lambda - 2^{l-1} - (l-1)\}^2$. Für $m = 0, \ldots, l$ setzen wir

$$r(m) = \sum_{i=0}^{m} 4^{l-i} \cdot (L_{net} + 1)$$

und zeigen Gleichung (A.8), indem wir durch Induktion über m zeigen, dass

$$\left(o^{(r(m))}(\mathbf{x})\right)_{(i',j'),s} = \bar{f}^{(i)}_{m,s}\left(\mathbf{x}_{(i',j')+I^{(m)}}\right) \tag{A.9}$$

für alle $(i', j') \in \{\lceil 2^{m-1} \rceil + m, \ldots, \lambda - \lceil 2^{m-1} \rceil - (m-1)\}^2$, $m \in \{0, \ldots, l\}$ und $s \in \{1, \ldots, 4^{l-m}\}$.

Wir starten mit $m = 0$ und zeigen, dass

$$\left(o^{(r(0))}(\mathbf{x})\right)_{(i',j'),s} = \sigma\left(g^{(i)}_{net,0,s}\left(x_{i',j'}\right)\right)$$

für alle $(i', j') \in \{1, \ldots, \lambda\}^2$ und $s \in \{1, \ldots, 4^l\}$. Die Idee ist, nacheinander Lemma 23 für die Berechnung jedes Netzwerks

$$\left\{\sigma\left(g^{(i)}_{net,0,s}\left(x_{i',j'}\right)\right)\right\}_{(i',j') \in \{1,\ldots,\lambda\}^2} \tag{A.10}$$

für $s \in \{1, \ldots, 4^l\}$ anzuwenden und die berechneten Werte in den Kanälen

$$1, \ldots, 4^l$$

unter Verwendung von Gleichung (3.31) zu speichern (angepasst an das symmetrische Zero-Padding aus Lemma 14). Bevor wir Lemma 23 anwenden, wählen wir die Gewichte im Kanal

$$4^l + 1$$

wie in Gleichung (3.31), sodass

$$\left(o^{(r)}(\mathbf{x})\right)_{(i',j'),4^l+1} = x_{i',j'}$$

für alle $r \in \{1, \ldots, r(0)\}$ und $(i', j') \in \{1, \ldots, \lambda\}^2$. Als Nächstes spezifizieren wir, wie wir Lemma 23 verwenden. Zunächst merken wir an, dass

$$M_1, \ldots, M_{r(0)} = 3.$$

Verwenden wir nun Lemma 23 mit den Parametern $d = 1$,

$$s_1 = \begin{cases} 1, & \text{falls } s = 1 \\ 4^l + 1, & \text{sonst} \end{cases}$$

$s_0 = s$, und $f = o^{((s-1)\cdot(L_{net}+1))}$ (wobei $o^{(0)}$ der Identität entspricht), können wir die Werte (A.10) in den Schichten

$$(s-1) \cdot (L_{net} + 1) + 1, \ldots, (s-1) \cdot (L_{net} + 1) + L_{net} + 1$$

berechnen, indem wir entsprechende Gewichte in den Kanälen

$$s, 5 \cdot 4^{l-1} + 1, \ldots, 5 \cdot 4^{l-1} + r_{net}$$

wählen, sodass

$$\left(o^{(s\cdot(L_{net}+1))}(\mathbf{x})\right)_{(i',j'),s} = \sigma\left(g^{(i)}_{net,0,s}\left(x_{i',j'}\right)\right)$$

für alle $(i',j') \in \{1,\dots,\lambda\}^2$ und $s \in \{1,\dots,4^l\}$. Wurde ein Wert einmal in Schicht $s \cdot (L_{net}+1)$ für $s \in \{1,\dots,4^l\}$ berechnet, wird er an die nächste Schicht weitergegeben, indem die Gewichte aus Gleichung (3.31) verwendet werden, sodass

$$\left(o^{(r(0))}(\mathbf{x})\right)_{(i',j'),s} = \sigma\left(g^{(i)}_{net,0,s}\left(x_{i',j'}\right)\right)$$

für alle $(i',j') \in \{1,\dots,\lambda\}^2$ und $s \in \{1,\dots,4^l\}$, was Gleichung (A.9) für $m=0$ impliziert.

Jetzt nehmen wir an, Gleichung (A.9) gelte für ein $m \in \{0,\dots,l-1\}$ und zeigen, dass Gleichung (A.9) dann auch für $m+1$ gilt, indem wir geeignete Gewichte in den Schichten

$$r(m)+1,\dots,r(m+1)$$

wählen, sodass

$$\begin{aligned}&\left(o^{(r(m+1))}(\mathbf{x})\right)_{(i',j'),s}\\ &= \sigma\Big(g^{(i)}_{net,m+1,s}\Big(\bar{f}^{(i)}_{m,4\cdot(s-1)+1}\Big(\mathbf{x}_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+1}+I^{(m)}}\Big), \bar{f}^{(i)}_{m,4\cdot(s-1)+2}\Big(\mathbf{x}_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+2}+I^{(m)}}\Big),\\ &\qquad \bar{f}^{(i)}_{m,4\cdot(s-1)+3}\Big(\mathbf{x}_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+3}+I^{(m)}}\Big), \bar{f}^{(i)}_{m,4\cdot s}\Big(\mathbf{x}_{(i',j')+\mathbf{i}_{m,4\cdot s}+I^{(m)}}\Big)\Big)\Big)\end{aligned} \tag{A.11}$$

für alle $\mathbf{x} \in [0,1]^{\{1,\dots,\lambda\}^2}$, $(i',j') \in \{2^m+m+2,\dots,\lambda-2^m-m-1\}^2$ und $s \in \{1,\dots,4^{l-(m+1)}\}$, wobei $\mathbf{i}_{m,4\cdot(s-1)+j}$ $(j=1,\dots,4)$ die Gitterpunkte des hierarchischen Modells von $\bar{f}^{(i)}_{l,1}$ bezeichnen. Da

$$\mathbf{i}_{m,s} \in \left\{-(\lfloor 2^{m-1}\rfloor+1),\dots,0,\dots,\lfloor 2^{m-1}\rfloor+1\right\}^2$$

für alle $s \in \{1,\dots,4^{l-m}\}$ gilt, haben wir

$$(i',j')+\mathbf{i}_{m,s} \in \{\lceil 2^{m-1}\rceil+m,\dots,\lambda-\lceil 2^{m-1}\rceil-(m-1)\}^2$$

für alle $(i',j') \in \{2^m+m+1,\dots,\lambda-2^m-m\}^2$ und $s \in \{1,\dots,4^{l-m}\}$. Wegen der Induktionshypothese ist Gleichung (A.11) damit äquivalent zu

$$\begin{aligned}&\left(o^{(r(m+1))}(\mathbf{x})\right)_{(i',j'),s}\\ &= \sigma\Big(g^{(i)}_{net,m+1,s}\Big(\left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+1},4\cdot(s-1)+1}, \left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+2},4\cdot(s-1)+2},\\ &\qquad \left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+3},4\cdot(s-1)+3}, \left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot s},4\cdot s}\Big)\Big).\end{aligned}$$

Analog zum Induktionsanfang ist die Idee, nacheinander Lemma 23 für die Berechnung jedes Netzwerks

$$\begin{aligned}\sigma\Big(g^{(i)}_{net,m+1,s}\Big(&\left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+1},4\cdot(s-1)+1}, \left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+2},4\cdot(s-1)+2},\\ &\left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+3},4\cdot(s-1)+3}, \left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot s},4\cdot s}\Big)\Big)\end{aligned} \tag{A.12}$$

für $s \in \{1,\dots,4^{l-(m+1)}\}$ zu verwenden und berechnete Werte mittels Gleichung (3.31) in den entsprechenden Kanälen

$$1,\dots,4^{l-(m+1)}$$

zu speichern. Bevor wir Lemma 23 anwenden, wählen wir die Gewichte in den Kanälen

$$4^{l-(m+1)}+1,\dots,4^{l-(m+1)}+4^{l-m}$$

so, dass

$$\left(o^{(r)}(\mathbf{x})\right)_{(i',j'),4^{l-(m+1)}+s}=\left(o^{(r(m))}(\mathbf{x})\right)_{(i',j'),s}$$

für alle $r\in\{r(m)+1,\dots,r(m+1)\}$, $(i',j')\in\{1,\dots,\lambda\}^2$ und $s=1,\dots,4^{l-m}$ durch eine weitere Anwendung von Gleichung (3.31). Als Nächstes spezifizieren wir, wie wir Lemma 23 verwenden. Zunächst merken wir an, dass

$$M_{r(m)+1},\dots,M_{r(m+1)}=2\cdot\lfloor 2^{m-1}\rfloor+3.$$

Verwenden wir nun Lemma 23 für $s\in\{1,\dots,4^{l-(m+1)}\}$ mit Parametern $d=4$,

$$s_j=\begin{cases}j, & \text{falls } s=1\\ 4^{l-(m+1)}+4\cdot(s-1)+j, & \text{sonst}\end{cases}$$

für $j=1,\dots,4$, $s_0=s$ und der Funktion $f=o^{(r(m)+(s-1)\cdot(L_{net}+1))}$, können wir die Werte (A.12) in den Schichten

$$r(m)+(s-1)\cdot(L_{net}+1)+1,\dots,r(m)+(s-1)\cdot(L_{net}+1)+L_{net}+1$$

berechnen, indem wir entsprechende Gewichte in den Kanälen

$$s,5\cdot 4^{l-1}+1,\dots,5\cdot 4^{l-1}+r_{net}$$

wählen, sodass

$$\begin{aligned}&\left(o^{(r(m)+s\cdot(L_{net}+1))}(\mathbf{x})\right)_{(i',j'),s}\\&=\sigma\Big(g^{(i)}_{net,m+1,s}\Big(\left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+1},4\cdot(s-1)+1},\left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+2},4\cdot(s-1)+2},\\&\qquad\left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+3},4\cdot(s-1)+3},\left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot s},4\cdot s}\Big)\Big)\end{aligned}$$

für alle $(i',j')\in\{2^m+m+2,\dots,\lambda-2^m-m-1\}^2$ und $s\in\{1,\dots,4^{l-(m+1)}\}$. Wurde ein Wert einmal in Schicht $r(m)+s\cdot(L_{net}+1)$ für $s\in\{1,\dots,4^{l-(m+1)}\}$ berechnet, wird er an die nächste Schicht weitergegeben, indem die Gewichte aus Gleichung (3.31) verwendet werden, sodass

$$\begin{aligned}&\left(o^{(r(m+1))}(\mathbf{x})\right)_{(i',j'),s}\\&=\sigma\Big(g^{(i)}_{net,m+1,s}\Big(\left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+1},4\cdot(s-1)+1},\left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+2},4\cdot(s-1)+2},\\&\qquad\left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot(s-1)+3},4\cdot(s-1)+3},\left(o^{(r(m))}(\mathbf{x})\right)_{(i',j')+\mathbf{i}_{m,4\cdot s},4\cdot s}\Big)\Big)\end{aligned}$$

für alle $(i',j')\in\{2^m+m+2,\dots,\lambda-2^m-m-1\}^2$ und $s\in\{1,\dots,4^{l-(m+1)}\}$, woraus der erste Schritt folgt.

Im zweiten Schritt wählen wir die Gewichte $\mathbf{w}_{out}$ der Ausgabeschicht, sodass Gleichung (A.7) gilt. Hier können wir $w_1=1$ und $w_s=0$ für $s\in\{2,\dots,k\}$ wählen und erhalten zusammen mit Gleichung (A.8)

$$f_i(\mathbf{x})=\max\left\{\sum_{s''=1}^{k}w_{s''}\cdot\left(o^{(L)}(\mathbf{x})\right)_{(i',j'),s''}\;:\;(i',j')\in\{2^{l-1}+l,\dots,\lambda-2^{l-1}-(l-1)\}^2\right\}$$

$$
\begin{aligned}
&= \max\left\{ \left(o^{(L)}(\mathbf{x})\right)_{(i',j'),1} \,:\, (i',j') \in \{2^{l-1}+l,\dots,\lambda-2^{l-1}-(l-1)\}^2 \right\} \\
&= \max\left\{ \bar{f}_{l,1}^{(i)}(\mathbf{x}_{(i',j')+I^{(l)}}) \,:\, (i',j') \in \{2^{l-1}+l,\dots,\lambda-2^{l-1}-(l-1)\}^2 \right\} \\
&= \max_{\mathbf{u}\in\{1,\dots,\lambda\}^2 \,:\, \mathbf{u}+I^{(l)}\subseteq\{1,\dots,\lambda\}^2} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}),
\end{aligned}
$$

wobei wir in der letzten Zeile Gleichung (3.48) verwendet haben. □

A.2. Beweis zu Beispiel 2.1

In diesem Abschnitt beweisen wir Ungleichung (2.10) aus Beispiel 2.1. Hierfür sei $h_0 \leq \min\{(c\cdot\sqrt{2})/\lambda, 1/\sqrt{2}\}$, $\mathbf{v} \in [h_0/\sqrt{2}-1/2, 1/2-h_0/\sqrt{2}]^2$ und $\phi \in A$ wie in Annahme 2 (es gelten außerdem alle weiteren Annahmen aus Annahme 2 und Beispiel 2.1). Für $\mathbf{z} \in C_1$ mit $\|\mathbf{z}-\mathbf{v}\|_\infty \leq \frac{c}{\lambda}$ und $\mathbf{y} \in C_{h_0}$ gilt

$$
\|\tau_{\mathbf{v}} \circ rot^{(\alpha)}(\mathbf{y}) - \mathbf{z}\|_\infty = \|\mathbf{v} + rot^{(\alpha)}(\mathbf{y}) - \mathbf{z}\|_\infty \leq \|\mathbf{v}-\mathbf{z}\|_\infty + \|rot^{(\alpha)}(\mathbf{y})\|_\infty \leq 2\cdot\frac{c}{\lambda},
$$

womit wir

$$
\begin{aligned}
&\sup_{\mathbf{z}\in C_1 \,:\, \|\mathbf{v}-\mathbf{z}\|_\infty \leq \frac{c}{\lambda}} \left| f_{0,s}\Big(\phi\circ\tau_{\mathbf{v}}\circ rot^{(\alpha)}\big|_{C_{h_0}}\Big) - f_{0,s}\Big(\phi(\mathbf{z})\cdot 1\big|_{C_{h_0}}\Big)\right| \\
&= \sup_{\mathbf{z}\in C_1 \,:\, \|\mathbf{v}-\mathbf{z}\|_\infty \leq \frac{c}{\lambda}} \left| \max_{\mathbf{y}\in C_{h_0}} \phi\circ\tau_{\mathbf{v}}\circ rot^{(\alpha)}\big|_{C_{h_0}}(\mathbf{y}) - \phi(\mathbf{z})\right| \\
&\leq \sup_{\mathbf{z}\in C_1 \,:\, \|\mathbf{v}-\mathbf{z}\|_\infty \leq \frac{c}{\lambda}} \max_{\mathbf{y}\in C_{h_0}} \left|\phi\circ\tau_{\mathbf{v}}\circ rot^{(\alpha)}(\mathbf{y}) - \phi(\mathbf{z})\right| \\
&\leq \sup_{\mathbf{y},\mathbf{z}\in C_1 \,:\, \|\mathbf{y}-\mathbf{z}\|_\infty \leq 2\cdot\frac{c}{\lambda}} |\phi(\mathbf{y}) - \phi(\mathbf{z})|
\end{aligned}
$$

folgern können. Wir zeigen die Behauptung, indem wir im Folgenden zeigen, dass

$$
\sup_{\mathbf{y},\mathbf{z}\in C_1 \,:\, \|\mathbf{y}-\mathbf{z}\|_\infty \leq \delta} |\phi_{\mathbf{x}}(\mathbf{y}) - \phi_{\mathbf{x}}(\mathbf{z})| \leq 16\cdot\lambda_{max}^2\cdot\delta \tag{A.13}
$$

für beliebiges $0 \leq \delta \leq \frac{1}{\lambda_{\max}-1}$ und $\mathbf{x} \in [0,1]^{H_{max}}$ gilt. Die Behauptung folgt aus Ungleichung (A.13) dann mit der Wahl $\delta = 2\cdot\frac{c}{\lambda}$. Zunächst merken wir an, dass für die Koeffizienten (2.9) gilt

$$
\max\left\{|k_1^{(\mathbf{y})}|, |k_2^{(\mathbf{y})}|, |k_3^{(\mathbf{y})}|\right\} \leq 2\cdot(\lambda_{max}-1)^2. \tag{A.14}
$$

Für $\mathbf{y},\mathbf{z} \in C_1$ mit $\|\mathbf{y}-\mathbf{z}\|_\infty \leq \delta$ können wir die Gitterpunkte (2.8) für die Berechnung der bilinearen Interpolationen $\phi_{\mathbf{x}}(\mathbf{y})$ und $\phi_{\mathbf{x}}(\mathbf{z})$ wegen $\delta \leq 1/(\lambda_{max}-1)$ so wählen, dass ein $\mathbf{u} \in C_1$ existiert mit

$$
\mathbf{u} \in ([a_1^{(\mathbf{y})}, a_2^{(\mathbf{y})}] \times [b_1^{(\mathbf{y})}, b_2^{(\mathbf{y})}]) \cap ([a_1^{(\mathbf{z})}, a_2^{(\mathbf{z})}] \times [b_1^{(\mathbf{z})}, b_2^{(\mathbf{z})}])
$$

(siehe Abbildung A.1 für eine Darstellung der entsprechenden Gitterpunkte und möglichen Punkten $\mathbf{u}$) und

$$
\max\{\|\mathbf{y}-\mathbf{u}\|_\infty, \|\mathbf{z}-\mathbf{u}\|_\infty\} \leq \delta.
$$

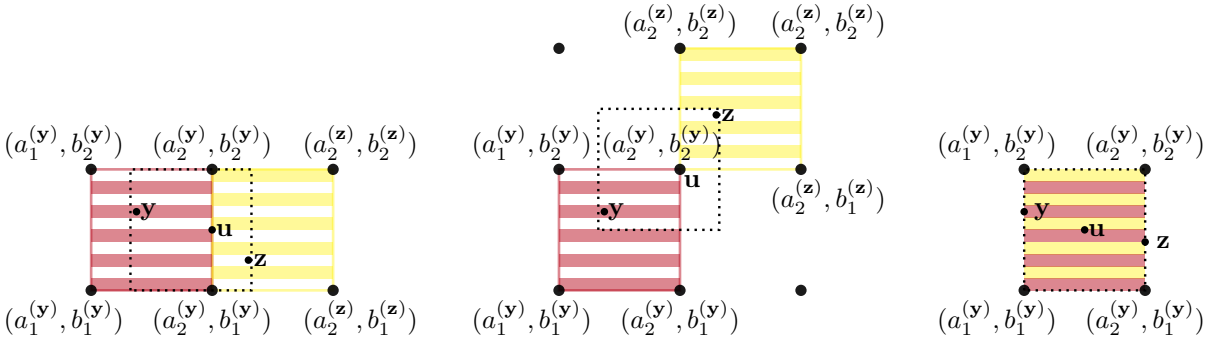


Abbildung A.1.: Wahl der Gitterpunkte zur Berechnung der Interpolationen $\phi_{\mathbf{x}}(\mathbf{y})$ und $\phi_{\mathbf{x}}(\mathbf{z})$.

Um $\phi_{\mathbf{x}}(\mathbf{u})$ zu berechnen, können wir daher die gleichen Gitterpunkte (2.8) wie für die Berechnung von $\phi_{\mathbf{x}}(\mathbf{y})$ verwenden, sodass zusammen mit Ungleichung (A.14) folgt

$$\begin{aligned}
|\phi_{\mathbf{x}}(\mathbf{y}) - \phi_{\mathbf{x}}(\mathbf{u})| &= |k_1^{(\mathbf{y})} \cdot (v_1 - u_1) + k_2^{(\mathbf{y})} \cdot (v_2 - u_2) + k_3^{(\mathbf{y})} \cdot (v_1 \cdot v_2 - u_1 \cdot u_2)| \\
&\leq |k_1^{(\mathbf{y})}| \cdot |v_1 - u_1| + |k_2^{(\mathbf{y})}| \cdot |v_2 - u_2| + |k_3^{(\mathbf{y})}| \cdot |v_1 \cdot v_2 - u_1 \cdot u_2| \\
&\leq 4 \cdot (\lambda_{max} - 1)^2 \cdot \delta + |k_3^{(\mathbf{y})}| \cdot (|v_1| \cdot |v_2 - u_2| + |u_2| \cdot |v_1 - u_1|) \\
&\leq 8 \cdot (\lambda_{max} - 1)^2 \cdot \delta
\end{aligned}$$

und analog auch

$$|\phi_{\mathbf{x}}(\mathbf{u}) - \phi_{\mathbf{x}}(\mathbf{z})| \leq 8 \cdot (\lambda_{max} - 1)^2 \cdot \delta,$$

was

$$|\phi_{\mathbf{x}}(\mathbf{y}) - \phi_{\mathbf{x}}(\mathbf{z})| \leq |\phi_{\mathbf{x}}(\mathbf{y}) - \phi_{\mathbf{x}}(\mathbf{u})| + |\phi_{\mathbf{x}}(\mathbf{u}) - \phi_{\mathbf{x}}(\mathbf{z})| \leq 16 \cdot \lambda_{max}^2 \cdot \delta$$

und damit auch Ungleichung (A.13) impliziert. □

A.3. Gewichtete AM-GM Ungleichung

Im Beweis von Lemma 20 verwenden wir, dass

$$\prod_{r=1}^{L} \left(\frac{b_r}{W_r} \right)^{W_r} \leq \left(\frac{\sum_{r=1}^{L} b_r}{\sum_{r=1}^{L} W_r} \right)^{\sum_{r=1}^{L} W_r} \tag{A.15}$$

für beliebige $L \in \mathbb{N}$, $b_1, \ldots, b_L \in \mathbb{R}^+$ sowie $W_1, \ldots, W_L \in \mathbb{R}^+$. Dies folgt aus einer Anwendung der hier angeführten gewichteten AM-GM Ungleichung (Ungleichung vom gewichteten arithmetischen und geometrischen Mittel), welche aus der Literatur bekannt ist.

Lemma 24 (Gewichtete AM-GM Ungleichung)**.** *Es sei $L \in \mathbb{N}$, $a_1, \ldots, a_L \in \mathbb{R}^+$ und $\alpha_1, \ldots, \alpha_L \in [0, 1]$, $i = 1, 2, \ldots, n$, sodass $\alpha_1 + \alpha_2 + \cdots + \alpha_n = 1$. Dann gilt*

$$a_1^{\alpha_1} \cdot a_2^{\alpha_2} \cdot \cdots \cdot a_n^{\alpha_n} \leq a_1 \cdot \alpha_1 + a_2 \cdot \alpha_2 + \cdots + a_n \cdot \alpha_n.$$

Beweis. Siehe Beweis von Theorem 7.6 in Cvetkovski (2012). □

Gleichung (A.15) ergibt sich nun, indem wir Lemma 24 mit $a_r = \frac{b_r}{W_r}$ $(r = 1, \ldots, L)$ und $\alpha_r = \frac{W_r}{\sum_{j=1}^{L} W_j}$ $(r = 1, \ldots, L)$ anwenden. Wir erhalten damit

$$\prod_{r=1}^{L} \left(\frac{b_r}{W_r}\right)^{W_r} = \left(\prod_{r=1}^{L} \left(\frac{b_r}{W_r}\right)^{\frac{W_r}{\sum_{r=1}^{L} W_r}}\right)^{\sum_{r=1}^{L} W_r} \leq \left(\sum_{r=1}^{L} \frac{b_r}{W_r} \cdot \frac{W_r}{\sum_{j=1}^{L} W_j}\right)^{\sum_{r=1}^{L} W_r} = \left(\frac{\sum_{r=1}^{L} b_r}{\sum_{r=1}^{L} W_r}\right)^{\sum_{r=1}^{L} W_r}.$$

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] Allen-Zhu, Z., Li, Y., und Song, Z. (2019). A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning, PMLR*, Band 97, Seiten 242–252.

[2] Anthony, M. und Bartlett, P. L. (1999). *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, Cambridge.

[3] Antos, A. (1999). Lower bounds on the rate of convergence ofnonparametric pattern recognition. In *Computational Learning Theory*, Seiten 241–252. Springer Berlin Heidelberg.

[4] Arora, R., Basu, A., Mianjy, P., und Mukherjee, A. (2018). Understanding deep neural networks with rectified linear units. In *6th International Conference on Learning Representations, ICLR 2018*.

[5] Audibert, J.-Y. und Tsybakov, A. B. (2007). Fast learning rates for plug-in classifiers. *The Annals of Statistics*, 35(2):608–633.

[6] Bagirov, A. M., Clausen, C., und Kohler, M. (2009). Estimation of a regression function by maxima of minima of linear functions. *IEEE Transactions on Information Theory*, 55:833–845.

[7] Barron, A. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945.

[8] Barron, A. (1994). Approximation and estimation bounds for artificial neural networks. *Machine Learning*, 14:115–133.

[9] Barron, A. R. (1991). *Complexity Regularization with Application to Artificial Neural Networks*, Seiten 561–576. Springer Netherlands, Dordrecht.

[10] Bartlett, P. L., Harvey, N., Liaw, C., und Mehrabian, A. (2019). Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20:1–17.

[11] Bauer, B. und Kohler, M. (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *Annals of Statistics*, 47(4):2261–2285.

[12] Bos, T. und Schmidt-Hieber, J. (2022). Convergence rates of deep ReLU networks for multiclass classification. *Electronic Journal of Statistics*, 16(1):2724 – 2773.

[13] Breiman, L., Friedman, J. H., Olshen, R. A., und Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

[14] Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., und Maureira, J. C. (2017). Deep-hits: Rotation invariant convolutional neural network for transient detection. *The Astrophysical Journal*, 836(1):97.

[15] Cannings, T., Berrett, T., und Samworth, R. (2020). Local nearest neighbour classification with applications to semi-supervised learning. *Annals of Statistics*, 48(3):1789–1814.

[16] Chollet, F. et al. (2015). Keras. https://keras.io.

[17] Clark, A. (2015). Pillow (pil fork) documentation. https://buildmedia.readthedocs.org/media/pdf/irskep-pillow/latest/irskep-pillow.pdf.

[18] Cohen, T. S. und Welling, M. (2016). Group equivariant convolutional networks. *International Conference on Machine Learning (ICML)*, 48:2990–2999.

[19] Cover, T. M. (1968). Rates of convergence of nearest neighbor procedures. In *Proceedings of the Hawaii International Conference on Systems Siences*, Seiten 413–415. Honolulu, HI.

[20] Cvetkovski, Z. (2012). *Inequalities: Theorems, Techniques and Selected Problems*. Springer, Berlin, Heidelberg.

[21] Dai, Z., Liu, H., Le, Q. V., und Tan, M. (2021). Coatnet: Marrying convolution and attention for all data sizes. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, und J. W. Vaughan, Herausgeber, *Advances in Neural Information Processing Systems*, Band 34, Seiten 3965–3977.

[22] Delchevalerie, V., Bibal, A., Frénay, B., und Mayer, A. (2021). Achieving rotational invariance with bessel-convolutional neural networks. In *Advances in Neural Information Processing Systems*, Band 34, Seiten 28772–28783.

[23] Devroye, L. (1982). Necessary and sufficient conditions for the pointwise convergence of nearest neighbor regression function estimates. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, 61:467–481.

[24] Devroye, L., Györfi, L., und Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer, New York.

[25] Dieleman, S., De Fauw, J., und Kavukcuoglu, K. (2016). Exploiting cyclic symmetry in convolutional neural networks. *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, 48:1889–1898.

[26] Dieleman, S., Willett, K. W., und Dambre, J. (2015). Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, 450(2):1441–1459.

[27] Du, S., Lee, J., Li, H., Wang, L., und Zhai, X. (2019). Gradient descent finds global minima of deep neural networks. In *36th International Conference on Machine Learning, ICML 2019*, Seiten 3003–3048. International Machine Learning Society (IMLS).

[28] Eldan, R. und Shamir, O. (2016). The power of depth for feedforward neural networks. In *Conference on Learning Theory*, Band 49, Seiten 907–940.

[29] Fan, J., Ma, C., und Zhong, Y. (2021). A Selective Overview of Deep Learning. *Statistical Science*, 36(2):264 – 290.

[30] Forster, O. (2017). *Analysis 2: Differentialrechnung im IRn, gewöhnliche Differentialgleichungen*. Vieweg+Teubner Verlag, 11. Auflage.

[31] Friedman, J. H. und Stuetzle, W. (1981). Projection pursuit regression. *Journal of the American Statistical Association*, 76(376):817–823.

[32] Géron, A. (2017). *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Incorporated, Sebastopol, CA.

[33] Gholamalinezhad, H. und Khosravi, H. (2020). Pooling methods in deep neural networks, a review. arXiv: 2009.07485.

[34] Gillies, C., S. and Van der Wel, Van den Bossche, J., Taves, M. W., Arnott, J., und Ward, B. C. (2007). Shapely: manipulation and analysis of geometric objects. `https://github.com/Toblerity/Shapely`.

[35] Gimel'farb, G. und Delmas, P. (2018). *Image Processing And Analysis: A Primer*. World Scientific.

[36] Glorot, X. und Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research - Proceedings Track*, 9:249–256.

[37] Goodfellow, I., Bengio, Y., und Courville, A. (2016). *Deep Learning*. MIT Press, London.

[38] Grigorescu, S., Trasnea, B., Cocias, T., und Macesanu, G. (2019). A survey of deep learning techniques for autonomous driving. *Journal of Field Robotics*, 37:362–386.

[39] Györfi, L., Kohler, M., Krzyzak, A., und Walk, H. (2002). *A Distribution-Free Theory of Nonparametric Regression*. Springer, New York.

[40] He, K., Zhang, X., Ren, S., und Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, Seiten 770–778.

[41] Horowitz, J. L. und Mammen, E. (2007). Rate-optimal estimation for a general class of nonparametric regression models with unknown link functions. *The Annals of Statistics*, 35(6):2589–2619.

[42] Hu, T., Shang, Z., und Cheng, G. (2020). Sharp rate of convergence for deep neural network classifiers under the teacher-student setting. arXiv: 2001.06892.

[43] Huber, P. J. (1985). Projection Pursuit. *The Annals of Statistics*, 13(2):435–475.

[44] Härdle, W., Hall, P., und Ichimura, H. (1993). Optimal Smoothing in Single-Index Models. *The Annals of Statistics*, 21(1):157–178.

[45] Härdle, W. und Stoker, T. M. (1989). Investigating smooth multiple regression by the method of average derivatives. *Journal of the American Statistical Association*, 84(408):986–995.

[46] Imaizumi, M. und Fukamizu, K. (2019). Deep neural networks learn non-smooth functions effectively. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*. Naha, Okinawa, Japan.

[47] Kim, Y., Ohn, I., und Kim, D. (2021). Fast convergence rates of deep neural networks for classification. *Neural Networks*, 138:179–197.

[48] Kingma, D. und Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

[49] Kirkland, E. J. (2010). *Advanced Computing in Electron Microscopy*. Springer New York, NY, 2. Auflage.

[50] Kohler, M. und Krzyżak, A. (2017). Nonparametric regression based on hierarchical interaction models. *IEEE Transactions on Information Theory*, 63(3):1620–1630.

[51] Kohler, M. und Krzyżak, A. (2021). Over-parametrized deep neural networks minimizing the empirical risk do not generalize well. *Bernoulli*, 27:2564–2597.

[52] Kohler, M., Krzyżak, A., und Walter, B. (2022). On the rate of convergence of image classifiers based on convolutional neural networks. *Annals of the Institute of Statistical Mathematics*, 74(6):1085–1108.

[53] Kohler, M., Krzyżak, A., und Langer, S. (2022). Estimation of a function of low local dimensionality by deep neural networks. *IEEE Transactions on Information Theory*, 68(6):4032–4042.

[54] Kohler, M. und Langer, S. (2020). Discussion of: "Nonparametric regression using deep neural networks with ReLU activation function". *The Annals of Statistics*, 48(4):1906–1910.

[55] Kohler, M. und Langer, S. (2020). Statistical theory for image classification using deep convolutional neural networks with cross-entropy loss. arXiv: 2011.13602.

[56] Kohler, M. und Langer, S. (2021). On the rate of convergence of fully connected very deep neural network regression estimates. *Annals of Statistics*, 49:2231–2249.

[57] Kohler, M. und Walter, B. (2023). Analysis of convolutional neural network image classifiers in a rotationally symmetric model. *Erscheint in IEEE Transactions on Information Theory 2023*.

[58] Kong, E. und Xia, Y. (2007). Variable selection for the single-index model. *Biometrika*, 94(1):217–229.

[59] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Technischer Bericht, Department of Computer Science, University of Toronto.

[60] Krizhevsky, A., Sutskever, I., und Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira et al. (Eds.), *Advances In Neural Information Processing Systems*, 25:1097–1105. Red Hook, NY: Curran.

[61] Kutyniok, G. (2022). The mathematics of artificial intelligence. arXiv: 2203.08890.

[62] Langer, S. und Schmidt-Hieber, J. (2022). A statistical analysis of an image classification problem. arXiv: 2206.02151.

[63] Larochelle, H., Erhan, D., Courville, A., Bergstra, J., und Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*.

[64] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., und Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551.

[65] Lepski, O. und Serdyukova, N. (2014). Adaptive estimation under single-index constraint in a regression model. *The Annals of Statistics*, 42(1):1–28.

[66] Lin, S. und Zhang, J. (2019). Generalization bounds for convolutional neural networks. arXiv: 1910.01487.

[67] Liu, H., Chen, M., Zhao, T., und Liao, W. (2021). Besov function approximation and binary classification on low-dimensional manifolds using convolutional residual networks. *Proceedings of the 38th International Conference on Machine Learning (PMLR)*, 139:6770–6780.

[68] Lu, P., Song, B., und Xu, L. (2020). Human face recognition based on convolutional neural network and augmented dataset. *Systems Science & Control Engineering*, 9:1–9.

[69] Mammen, E. und Tsybakov, A. B. (1999). Smooth discrimination analysis. *The Annals of Statistics*, 27(6):1808– 829.

[70] Marcos, D., Volpi, M., und Tuia, D. (2016). Learning rotation invariant convolutional filters for texture classification. *International Conference on Pattern Recognition (ICPR)*, Seiten 2012–2017.

[71] McCaffrey, D. F. und Ronald Gallant, A. (1994). Convergence rates for single hidden layer feedforward networks. *Neural Networks*, 7(1):147–158.

[72] Mhaskar, H. und Poggio, T. (2016). Deep vs. shallow networks : An approximation theory perspective. *Analysis and Applications*, 14(6):829–848.

[73] Morris, T., Chien, T., und Goodman, E. (2018). Convolutional neural networks for automatic threat detection in security x-ray images. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, Seiten 285–292.

[74] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., und Ng, A. Y. (2011). Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*.

[75] Nwankpa, C., Ijomah, W. L., Gachagan, A., und Marshall, S. (2018). Activation functions: Comparison of trends in practice and research for deep learning. arXiv: 1811.03378.

[76] Oono, K. und Suzuki, T. (2019). Approximation and non-parametric estimation of resnet-type convolutional neural networks. *In International Conference on Machine Learning*, Seiten 4922–4931.

[77] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., und Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

[78] Petersen, P. und Voigtlaender, F. (2020). Equivalence of approximation by convolutional neural networks and fully-connected networks. *Proceedings of the American Mathematical Society*, 148:1567–1581.

[79] Ramachandran, P., Zoph, B., und Le, Q. V. (2018). Searching for activation functions. arXiv: 1710.05941.

[80] Rawat, W. und Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29:2352–2449.

[81] Rudin, C. und Ustun, B. (2018). Optimized scoring systems: Toward trust in machine learning for healthcare and criminal justice. *Interfaces*, 48.

[82] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., und Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252.

[83] Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with relu activation function. *Annals of Statistics*, 48(4):1875–1897.

[84] Sharma, S., Sharma, S., und Athaiya, A. (2020). Activation functions in neural networks. *International Journal of Engineering Applied Sciences and Technology*, 4(12):310–316.

[85] Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., und Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature*, 550:354–359.

[86] Simonyan, K. und Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations, ICLR 2015*.

[87] Steinwart, I. und Scovel, C. (2007). Fast rates for support vector machines using Gaussian kernels. *The Annals of Statistics*, 35(2):575–607.

[88] Stone, C. J. (1977). Consistent nonparametric regression. *The Annals of Statistics*, 5(4):595–620.

[89] Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. *Annals of Statistics*, 10:1040–1053.

[90] Stone, C. J. (1985). Additive Regression and Other Nonparametric Models. *The Annals of Statistics*, 13(2):689–705.

[91] Stone, C. J. (1994). The Use of Polynomial Splines and Their Tensor Products in Multivariate Function Estimation. *The Annals of Statistics*, 22(1):118–171.

[92] Suzuki, T. und Nitanda, A. (2021). Deep learning is adaptive to intrinsic dimensionality of model smoothness in anisotropic besov space. In *Advances in Neural Information Processing Systems*, Band 34, Seiten 3609–3621. Curran Associates, Inc.

[93] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., und Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seiten 1–9.

[94] Tsybakov, A. und van de Geer, S. (2005). Square Root Penalty: Adaptation to the Margin in Classification and in Edge Estimation. *The Annals of Statistics*, 33:1203–1224.

[95] Veeling, B. S., Linmans, J., Winkens, J., Cohen, T., und Welling, M. (2018). Rotation equivariant cnns for digital pathology. In A. F. Frangi, J. A. Schnabel, C. Davatzikos, C. Alberola-López, und G. Fichtinger, Herausgeber, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, Seiten 210–218. Springer International Publishing, Cham, Switzerland.

[96] Walter, B. (2023). Analysis of convolutional neural network image classifiers in a hierarchical max-pooling model with additional local pooling. *Journal of Statistical Planning and Inference*, 224:109–126.

[97] Willett, K. W., Lintott, C. J., Bamford, S., Masters, K. L., Simmons, B. D., Casteels, K., Edmondson, E. M., Fortson, L., Kaviraj, S., Keel, W. C., Melvin, T. R. O., Nichol, R., Raddick, M. J., Schawinski, K., Simpson, R. J., Skibba, R. A., Smith, A. M., of Minnesota, D. T. U., of Oxford, U., Planetarium, A., of Nottingham, U., of Portsmouth, U., SepNet, de Barcelona, U. A., of Hertfordshire, U., of South Alabama, U., University, J. H., zurich, E., und of California at San Diego, U. (2013). Galaxy zoo 2: detailed morphological classifications for 304,122 galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society*, 435:2835–2860.

[98] Wu, F., Hu, P., und Kong, D. (2015). Flip-rotate-pooling convolution and split dropout on convolution neural networks for image classification. arXiv: 1507.08754.

[99] Wu, Y., Schuster, M., Chen, Z., Le, Q., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, u., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., und Dean, J. (2016). Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv: 1609.08144.

[100] Yang, Y. (1999). Minimax nonparametric classification – part I: Rates of convergence. *IEEE Transactions on Information Theory*, 45(7):2271–2284.

[101] Yarotsky, D. (2022). Universal approximations of invariant maps by neural networks. *Constructive Approximation*, 55:1–68.

[102] Yarotsky, D. und Zhevnerchuk, A. (2020). The phase diagram of approximation rates for deep neural networks. In *Advances in Neural Information Processing Systems*, Band 33, Seiten 13005–13015.

[103] Yu, Y. und Ruppert, D. (2002). Penalized spline estimation for partially linear single-index models. *Journal of the American Statistical Association*, 97:1042–1054.

[104] Zeiler, M. D. und Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Computer Vision – ECCV 2014*, Seiten 818–833. Springer International Publishing.

[105] Zhou, D.-X. (2020). Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, 48(2):787–794.

[106] Zou, D., Cao, Y., Zhou, D., und Gu, Q. (2020). Stochastic gradient descent optimizes over-parameterized deep relu networks. *Machine Learning*, 109:467–492.

Wissenschaftlicher Werdegang

04/2013 – 07/2016	**Johannes Gutenberg-Universität Mainz** *B.Sc. Mathematik mit Nebenfach Experimentalphysik* Thesis: „Der numerische Wertebereich und der Satz von Toeplitz und Hausdorff“
10/2016 – 12/2019	**Technische Universität Darmstadt** *M.Sc. Mathematik mit Nebenfach Informatik* Thesis: „Schätzung einer Regressionsfunktion durch einen linearen Kleinste-Quadrate-Schätzer basierend auf neuronalen Netzen“
03/2020 – 06/2023	**Technische Universität Darmstadt** *Promotion Mathematik* Wissenschaftlicher Mitarbeiter in der Arbeitsgruppe Stochastik