

Proseminar

Künstliche neuronale Netze

Jens Ostertag

12. Mai 2022

Inhaltsverzeichnis

1	Computer lernen lassen	2
1.1	Aufbau neuronaler Netze [1]	3
1.2	Drei Arten des maschinellen Lernens [2]	5
1.2.1	Überwachtes Lernen	5
1.2.2	Unüberwachtes Lernen	6
1.2.3	Bestärktes Lernen	7
1.3	Umgang mit neuronalen Netzen [2]	7
1.3.1	Vorbereiten eines Trainingsdatensatzes	7
1.3.2	Trainieren	8
1.3.3	Auswertung und Anwendung	8
2	Deep Convolutional Neural Networks [3]	8
2.1	Convolutional Layer	9
2.1.1	Eindimensionale Cross Correlation	9
2.1.2	Zweidimensionale Cross Correlation	10
2.1.3	Übersicht über Vorteile	12
2.2	Subsampling Layer	12
3	Resultate	13

1 Computer lernen lassen

„Künstliche Intelligenz“, „Maschinelles Lernen“ und „Neuronale Netze“ - Alle diese Begriffe sind gewissermaßen miteinander verbunden, denn ihre Technik beruht darauf, dass Computer lernen können. Vor einem tieferen Einblick möchte ich diese Begriffe jedoch einordnen und differenzieren.

Bei einem künstlichen neuronalen Netz handelt es sich um die Simulation des menschlichen bzw. natürlichen neuronalen Netzes, einer Struktur, die im Gehirn vorhanden ist. Neuronale Netze sind die Grundlage für alles, das Lernen kann oder Entscheidungen treffen soll.

Maschinelles Lernen beschreibt das Lernen eines solchen neuronalen Netzes. Wie auch der Mensch ist ein künstliches neuronales Netz in der Lage, sich unterschiedliche Fähigkeiten anzueignen, unabhängig von der Komplexität der jeweiligen Aufgabe. Dadurch wird beispielsweise ermöglicht, dass ein Programm Bilder erkennt und klassifiziert oder die Position bestimmter Objekte darin bestimmen kann.

Die künstliche Intelligenz ist die undefinierteste Form neuronaler Netze, was auch damit zusammenhängt, dass es keine sehr genaue Definition von Intelligenz gibt. Ist man intelligent, wenn man schnell rechnen kann? Somit ist bereits ein simpler Taschenrechner von vor 40 Jahren äußerst intelligent. Oder bedeutet Intelligenz, dass man selbstständig denken und fühlen können muss, in dessen Zusammenhang der Begriff „Künstliche Intelligenz“ meistens verwendet wird? Aufgrund dieser Undefiniertheit wird im Folgenden, sofern möglich, auf diesen Begriff verzichtet.

Obwohl die Theorie schon etwas länger existiert, erfolgte ein großer Vorsprung in der Entwicklung des maschinellen Lernens erst in den letzten Jahren. Auch wenn es zuerst nicht so scheint, kommt heute jeder täglich damit in Berührung, beispielsweise

- im Verkehr, durch schlau gesteuerte Ampelphasen oder nahezu selbst-fahrende Autos sowie Assistenzsysteme wie Spurhalteassistenten, Einparkhilfen oder Ähnliches,
- im Internet, wo jedem im Sekundentakt neue Inhalte basierend auf persönlichen Präferenzen vorgeschlagen werden,
- im medizinischen Bereich, wo Bilderkennung bei der Auswertung bildgebender Verfahren hilft,

- in der Industrie, wo Produktionsabläufe durch maschinelles Lernen optimiert wurden.

Im Folgenden soll es darum gehen, wie ein neuronales Netz funktioniert und wie man es mit maschinellem Lernen verwenden kann.

1.1 Aufbau neuronaler Netze [1]

Künstliche neuronale Netze sind sehr stark an natürliche neuronale Netze angelehnt, weshalb es sinnvoll ist, zuerst diese zu betrachten.

Sogenannte „Neuronen“ (Nervenzellen) sind durch Synapsen miteinander verbunden, welche für den Elektronenfluss zwischen jeweils zwei Neuronen mithilfe eines „Synapsengewichts“ verantwortlich sind. Mit diesem lässt sich die Anfälligkeit des nachgehenden Neurons auf das Signal des vorhergehenden Neurons regulieren. [4]

Es ergibt sich somit das folgende Bild:

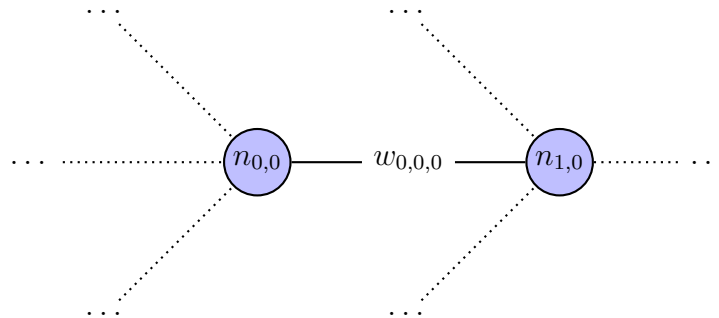


Abbildung 1: Synapse zwischen zwei Neuronen

Es sei jedoch angemerkt, dass ein Neuron beliebig viele vorausgehende oder nachgehende Neuronen haben kann, die in Abbildung 1 der Übersicht halber nicht enthalten sind. Als grobe Größenordnung sei die Anzahl der Neuronen eines ausgewachsenen, menschlichen Gehirns genannt, welche sich je nach Literatur auf 80 bis 100 Milliarden beläuft. In der Abbildung sind vernachlässigte Ein- und Ausgänge von Neuronen mit gepunkteten Linien symbolisiert.

Eine wichtige Grundlage, um aus dem natürlichen neuronalen Netz ein Künstliches abzuleiten, ist das Betrachten von Ausgaben von Neuronen und

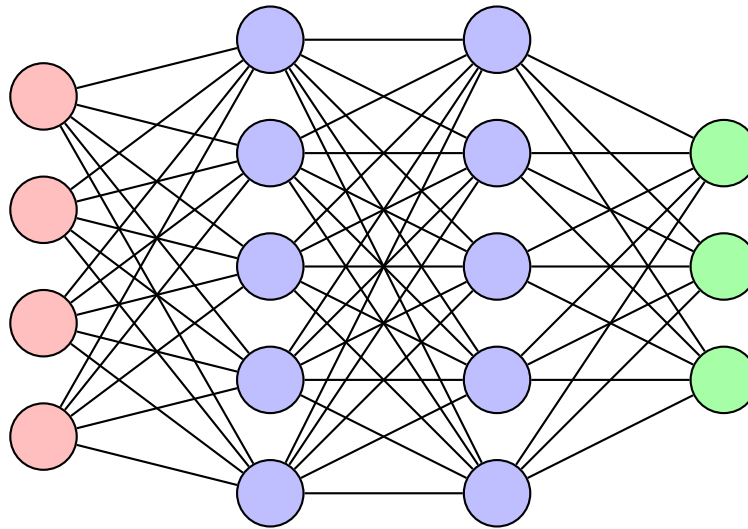


Abbildung 2: Aufbau eines einfachen neuronalen Netzes

von Synapsengewichten als Zahlen. Das ermöglicht die Anwendung von Formeln zur mathematischen Beschreibung.

Ebenso wichtig ist die Einteilung aller Neuronen des Netzes in unterschiedliche Schichten, die teilweise sogar unterschiedliche Aufgaben übernehmen müssen. Als Veranschaulichung dafür dient Abbildung 2, die den Aufbau eines einfachen neuronalen Netzes darstellt.

Die erste Schicht ist die sogenannte „Eingabeschicht“, dessen Neuronen (rot) vergleichbar sind mit Sinneszellen. Hier werden Daten gesammelt, mit denen eine Ausgabe generiert werden soll. Es folgen beliebig viele „versteckte Schichten“. Diese erfüllen den Zweck, dass das Netz auch kompliziertere Dinge lernen kann, da die Gesamtheit der Gewichte ohne versteckte Schicht nur eine linear separierbare Funktion darstellen könnte. Abschließend folgt eine „Ausgabeschicht“, welche beispielsweise Entscheidungen des Netzes ausgibt, welche dann in einem anderen Teil des Programms verarbeitet werden. In diesem Fall sind alle Neuronen einer Schicht mit allen Neuronen der nächsten Schicht verbunden. Wie jedoch in Kapitel 2 noch thematisiert wird, existieren auch weitere Strukturen für den Aufbau eines neuronalen Netzes.

Diese Grundlagen ermöglichen eine Berechnung von beispielsweise Aus-

geben mithilfe der Formel

$$o_{i,j} = \varphi \left(\sum_{k=0}^{|n_{i-1}|} o_{i-1,k} * w_{i-1,k,j} \right) \quad (1)$$

mit

- $n_{i,j}$: Neuron in der Schicht i an der Stelle j
- $o_{i,j}$: Ausgabe des Neurons $n_{i,j}$
- φ : Differenzierbare Aktivierungsfunktion
- $|n_i|$: Anzahl der Neuronen in der Schicht i
- $w_{i,k,j}$: Synapsengewicht zwischen den Neuronen $n_{i,k}$ und $n_{i+1,j}$

1.2 Drei Arten des maschinellen Lernens [2]

Damit ein neuronales Netz lernt, müssen seine Gewichte angepasst werden. Dieser Vorgang wird auch Training genannt. Das kann auf drei unterschiedliche Arten erfolgen.

1.2.1 Überwachtes Lernen

Das überwachte Lernen ist die wohl am häufigsten verwendete Methode, ein neuronales Netz zu trainieren. Es zielt darauf ab, auch zu bislang unbekannten Eingaben eine passende Ausgabe zu generieren, meistens geht es darum, die Eingaben in Gruppen bestimmter Eigenschaften einzuteilen.

Während des Trainingsvorgangs wird mithilfe eines Trainingsdatensatzes (einige Eingaben mit zugehörigen erwarteten Ausgaben) angelernet, welche Muster in den Eingabedaten zu bestimmten Ausgaben gehören. Erhält das Netz nach abgeschlossenem Training eine Eingabe, ist es in der Lage, ein Muster darin zu erkennen und anhand dessen eine passende Ausgabe zu generieren.

In Abbildung 3 entsprechen alle Punkte einer bestimmten Eingabe, gleichfarbige Punkte haben dieselbe erwartete Ausgabe. Während des Trainings lernt das Netz beispielsweise, dass es Eingaben mit hohem x - und niedrigem y -Wert die Ausgabe grüner Punkte zuteilen soll.

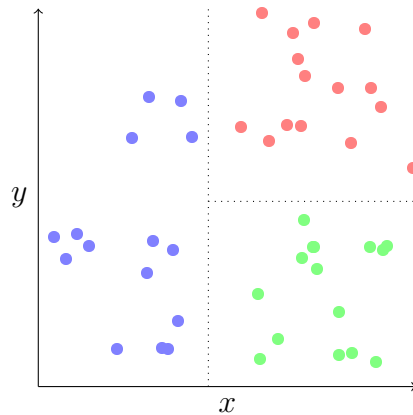


Abbildung 3: Klassifizierung von Daten

1.2.2 Unüberwachtes Lernen

Das Ziel des unüberwachten Lernens ist es, innerhalb einer Menge von Eingaben Gruppen anhand ähnlicher Strukturen zu finden. Dafür wird keine Bearbeitung oder Sortierung der Daten benötigt.

Daher ist diese Lernart besonders interessant für die Clusteranalyse, welche dasselbe Ziel anstrebt. Sie spielt besonders im Internet eine Rolle, wo Nutzer unterschiedlicher Zielgruppen unterschiedliche Inhalte, Produktvorschläge oder Werbungen angezeigt bekommen sollen.

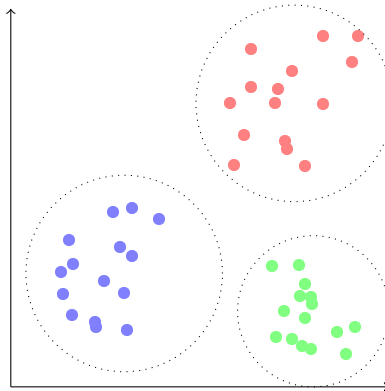


Abbildung 4: Gruppierung von Daten

In Abbildung 4 entsprechen alle Punkte unterschiedlichen Daten, gleichfarbige Punkte haben jedoch ähnliche Eigenschaften. Das neuronale Netz

soll diese Ähnlichkeiten erkennen und die Punkte gleicher Farbe jeweils in die gleiche Kategorie einteilen.

1.2.3 Bestärktes Lernen

Bestärktes Lernen dient dazu, den Entscheidungsprozess eines neuronalen Netzes bezüglich einer bestimmten Tätigkeit zu trainieren.

Dabei wird mit Belohnungen gearbeitet: Je besser eine Ausgabe war, umso höher ist die Belohnung für das Netz, was darin resultiert, dass die jeweilige Entscheidung verstärkt wird. Entschied sich das Netz falsch, ist die Belohnung gering und die Entscheidung wird geschwächt. Während des Lernvorgangs wird stets versucht, die Belohnung zu maximieren, um immer die bestmögliche Ausgabe zu erreichen.

Die Belohnungen sind hierbei eine Art Feedback für das Netz, weshalb man auch von einer Art des überwachten Lernens sprechen kann.

1.3 Umgang mit neuronalen Netzen [2]

Vor der Implementation maschinellen Lernens in einem Programm muss ein neuronales Netz entwickelt werden, das die gewünschte Aufgabe zuverlässig erledigen kann. Für die einzelnen Lernarten existieren standardisierte Abläufe, die von Netz zu Netz nahezu gleich sind. Im Folgenden wird der des überwachten Lernens vorgestellt.

1.3.1 Vorbereiten eines Trainingsdatensatzes

Zuerst müssen Daten gesammelt und zu einem Trainingsdatensatz zusammengefügt werden. Bei allen Daten handelt es sich um mögliche Eingaben für das Netz. Sollen also beispielsweise Bilder von unterschiedlichen Objekten unterschieden werden, sind die Daten ebenfalls Bilder, auf denen jeweils eines der zu klassifizierenden Objekte enthalten ist.

Neben dem einfachen Zusammenstellen des Datensatzes kann es jedoch auch vorkommen, dass Daten zuerst bearbeitet werden müssen, sodass einzelne Details auffälliger sind. Das kann geschehen, indem beispielsweise ein Bild auf die relevanten Pixel reduziert wird. Dadurch lernt das Netz nicht nur besser, sondern auch schneller und mit einer geringeren Datenmenge als würden Daten nicht optimiert werden.

Sämtliche Anpassungen der Daten müssen einheitlich und auch bei der Anwendung des Netzes mit unbekannten Daten erfolgen.

Außerdem ist es für die bevorstehende Auswertung wichtig, dass ebenfalls ein Testdatensatz erstellt wird. Sein Aufbau ist gleich wie der des Trainingsdatensatzes, es sind lediglich andere Daten enthalten.

1.3.2 Trainieren

Mit dem fertigen Trainingsdatensatz kann das neuronale Netz trainiert werden. Dafür existieren unterschiedliche Trainingsalgorithmen, die sich auch zwischen besonderen Aufbauten neuronaler Netze und deren spezifischen Aufgaben unterscheiden können.

Je nach Aufgabe oder spezifischen Struktur des Netzes können die Algorithmen unterschiedlich gut abschließen. Daher ist es lohnenswert, das Training nicht nur auf einen der Algorithmen zu beschränken, sondern mehrere mit Bezug auf die durchzuführende Aufgabe zu testen.

1.3.3 Auswertung und Anwendung

Nachdem das Training abgeschlossen wurde, ist es interessant zu wissen, mit welcher Genauigkeit oder Zuverlässigkeit eine Aufgabe ausgeführt wird. Dafür kommt der zuvor erstellte Testdatensatz in Einsatz, mit dem die Anzahl der falschen bzw. unerwünschten Ausgaben ermittelt wird und der somit eine Einschätzung des Netzes möglich macht.

Wenn die Aufgabe zufriedenstellend abgeschlossen wird, können die Gewichte und die Konfiguration des Netzes in Dateien abgespeichert und in einem Anwendungsprogramm importiert. Das ermöglicht eine Rekonstruktion des neuronalen Netzes ohne großen Zeitaufwand und ohne erneut ein Training durchführen zu müssen.

2 Deep Convolutional Neural Networks [3]

Deep Convolutional Neural Networks sind besondere Formen der neuronalen Netze, die besonders in der Bilderkennung und -Klassifikation sehr häufig verwendet werden. Grund dafür ist, dass sie viel schneller und besser trainiert werden können als herkömmliche neuronale Netze. Während diese eine Genauigkeit von über 80% nur selten erreichen, können Deep Convolutio-

nal Neural Networks Bilder meist problemlos über 95% der Bilder korrekt klassifizieren.

Das ist auf deren Aufbau und Lernweise zurückzuführen, welche der des Menschen noch mehr ähnelt als die herkömmlicher neuronaler Netze.

2.1 Convolutional Layer

Verwendet man ein übliches neuronales Netz für die Bildklassifikation, ist die Relevanz zweier nebeneinanderliegender Pixel genauso hoch wie die von weit entfernten Pixeln. Das folgt daraus, dass die Ausgabe jedes Neurons an jedes Neuron in der folgenden Schicht propagiert wird.

In einem Convolutional Layer ist das anders: Mithilfe eines Filters werden aus nebeneinanderliegenden Pixeln Eigenschaften erkannt. In den ersten Schichten des Netzes beschränken sich diese Eigenschaften auf Kanten und andere einfache Formen, je tiefer sich eine Schicht jedoch im Netz befindet, kann diese immer komplexere Objekte anhand von Konturen unterscheiden.

Um dies zu erreichen, wird die Convolution oder die Cross Correlation angewandt.

2.1.1 Eindimensionale Cross Correlation

Als „Cross Correlation“ wird im Folgenden eine Operation mit zwei Vektoren x und w betrachtet, aus dem ein weiterer Vektor y berechnet werden soll. Sie stellt eine wichtige Grundlage der Convolutional Neural Networks dar, die sogenannte „Convolutional Layer“ enthalten. Außerdem ist sie eine Vereinfachung der Convolution, welche selbst nur in wenigen Machine-Learning-Frameworks implementiert ist. Der Unterschied besteht zwischen der Handhabung der Vektoren, die bei der Convolution in alle Dimensionen gespiegelt sind. Im Anwendungsfall mit einem Algorithmus ist dieser Effekt allerdings vernachlässigbar.

Unter Definition der Variablen

- x : Eingabevektor
- w : Filter (vergleichbar mit Gewichtsvektor)
- y : Ausgabevektor

mit den Bedingungen

$$|x| \geq |w| \quad |x|, |w| > 0$$

kann die valide Cross Correlation im eindimensionalen Raum (folgend \odot) mathematisch durch die Formel

$$y = x \odot w \quad \rightarrow \quad y_i = \sum_{k=0}^{|w|-1} x_{i+k} * w_k \quad (2)$$

beschrieben werden. In Worten bedeutet das, dass der Filter w an den Eingaben x angelegt wird. Addiert man das Produkt der einzelnen Werte x_i und w_i , ermittelt man y_i . Um y_{i+1} zu berechnen, muss der Filtervektor um eine Einheit verschoben werden. Ein Sinnbild ist in Abbildung 5 dargestellt.

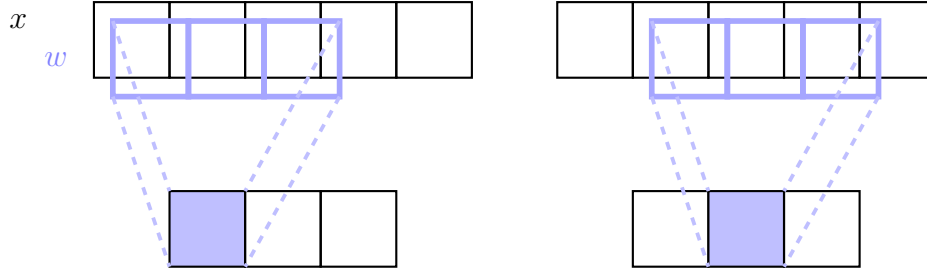


Abbildung 5: Cross Correlation im eindimensionalen Raum, ohne Padding

Dabei ist offensichtlich, dass die Länge der Ausgabe stets kleiner als die Länge der Eingabe ist. Aus diesem Grund wird ein „Padding“ eingeführt, welches den Eingabevektor x künstlich mit Nullen erweitern soll. Zur Ermittlung des Ausgabevektors y entsteht das in Abbildung 6 dargestellte Schema.

Somit wird ein Ausgabevektor y erreicht, der dieselbe Länge hat wie der Eingabevektor x .

2.1.2 Zweidimensionale Cross Correlation

Bilder sind jedoch nicht eindimensional. Daher wird eine zweidimensionale Cross Correlation benötigt. Zwar lässt sie sich in Formeln durch

$$Y = X \odot W \quad \rightarrow \quad Y_{i,j} = \sum_{k=0}^{|W|} \sum_{l=0}^{|W_l|} X_{i+k,j+l} * W_{k,l} \quad (3)$$

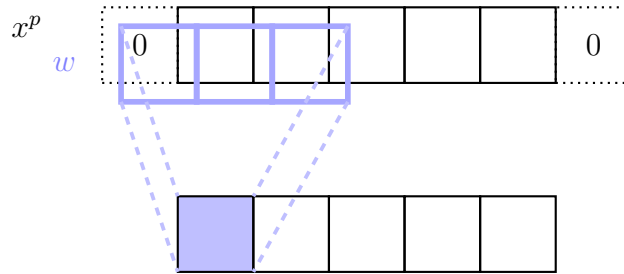


Abbildung 6: Cross Correlation im eindimensionalen Raum, mit Padding

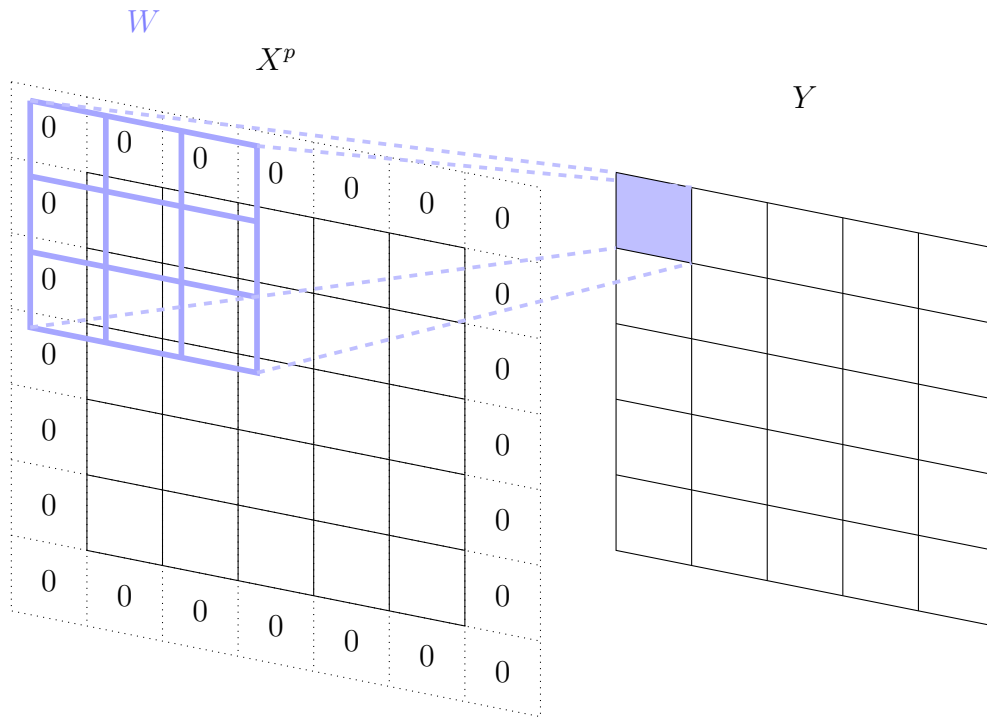


Abbildung 7: Cross Correlation im zweidimensionalen Raum

beschreiben, anhand eines Schemas ist sie allerdings einfacher zu verstehen.

Auch hier wird die Filtermatrix W an die Eingabematrix X angelegt und verschoben, um die Werte der Ausgabe Y zu berechnen. Es ist jedoch zu beachten, dass W nun nicht nur in eine, sondern in beide Dimensionen verschoben wird.

Damit keine Pixel verloren werden, wird die Eingabe X ebenfalls zu einem

X^p mit Padding erweitert.

2.1.3 Übersicht über Vorteile

Ein Convolutional Layer ist also eine Schicht in einem neuronalen Netz, die nur nebeneinanderliegende Pixel mit einem immer wiederkehrenden Filter (in der Regel eine 3×3 oder eine 5×5 -Matrix) betrachtet und dadurch zunächst einfache Formen, nach wenigen dieser Schichten allerdings schon komplexere Konturen erkennen kann. Dadurch wird ebenfalls die absolute Position eines zu klassifizierenden Objekts innerhalb eines Bildes egal, da lediglich die Relativität von Formen zueinander eine Rolle spielen.

Da die Schicht mit einem einheitlichen Filter mit etwa 9 bis 25 Gewichten auf Eigenschaften untersucht wird, gibt es auch wenige Variablen, die bei einem Trainingsvorgang verändert werden müssen, wodurch dieser wesentlich kürzer dauert als in einem vollständig verbundenen neuronalen Netz.

2.2 Subsampling Layer

Subsampling Layers kommen in neuronalen Netzen vor, um die Menge an Eigenschaften einer Eingabe mittels Pooling-Operationen zu verringern. Das resultiert in einem deutlich schnelleren Trainingsvorgang, da nach einem Subsampling Layer weniger Gewichte angepasst werden müssen.

Eine Eingabe X wird in mehrere kleine Pooling-Matrizen P aufgeteilt. Ab diesem Punkt können zwei unterschiedliche Arten angewandt werden: Bei „Max-Pooling“ wird jeweils der größte Wert der Pooling-Matrix in die Ausgabe weitergeleitet, bei „Mean-Pooling“ wird der Durchschnitt aller darin enthaltenen Werte ermittelt.

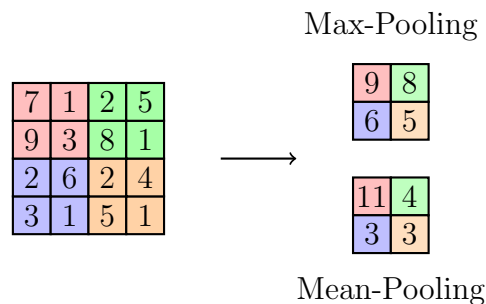


Abbildung 8: Min- und Mean-Pooling

Subsampling Layer selbst haben keine Gewichte sondern beruhen auf simplen mathematischen Berechnungen wie dem Durchschnitt oder einem Größenvergleich.

3 Resultate

Literatur

- [1] CALLAN, ROBERT: *Neuronale Netze im Klartext*. Pearson Studium, 1. Auflage, 2003.
- [2] RASCHKA, SEBASTIAN und VAHID MIRJALILI: *Python Machine Learning*. Packt, 3. Auflage, 2015. Kapitel 1: Giving Computers the Ability to Learn from Data.
- [3] RASCHKA, SEBASTIAN und VAHID MIRJALILI: *Python Machine Learning*. Packt, 3. Auflage, 2015. Kapitel 15: Classifying Images with Deep Convolutional Networks.
- [4] SPITZER, MANFRED: *The Mind Within the Net: Models of Learning, Thinking and Acting*, Seite 21. MIT Press, 2. Auflage, 1999.
<https://books.google.de/books?id=lniT11DbRX4C&pg=0>, 23.04.2022.