Schiller-Gymnasium Hameln

12g Seminarfacharbeit

Mathematik/Physik

Umsetzung einer Künstlichen Intelligenz zur Erkennung handgeschriebener Zahlen

Autor:
Bui Anh Minh Leon Phan

Tutoren:
Hr. Fistler, Hr. Dr. Kajari

27. Januar 2023

Inhaltsverzeichnis

Abbildungsverzeichnis

Abkürzungsverzeichnis

KI Künstliche Intelligenz

CNN Faltendes Neuronales Netzwerk

1 Einleitung

(Seite 5) In der heutigen Zeit spielt Künstliche Intelligenz (KI) eine sehr wichtige Rolle in unserer Gesellschaft. Es gibt viele Anwendungsbereiche für KI, sei es im Bereich Verkehr, Gesundheit, Werbung oder auch Cybersicherheit. [1] KI wird zum unverzichtbaren Mittel der Zukunft. So muss der Forschungsstand zum Bereich KI besonders weit sein, da die Fehlerberechnungen auf das Minimum beschränkt werden müssen. Nun ist es aber so, dass die jetzigen KIs nur auf einzelne Themen spezialisiert werden. Eine KI, die in mehreren Themengebieten professionell funktionieren kann, ist heutzutage noch nicht möglich. [2] Außerdem hinterfragt keiner, wie eine KI aufgebaut ist und wie dieser funktioniert. Zum Schluss schreiben

2 Künstliche Intelligenz

Der Begriff Künstliche Intelligenz (KI) ist sehr schwer definierbar, denn es mangelt an der Defintion des Begriffs Intelligenz. Allgemein kann die KI definiert werden als der Versuch, bestimmte Entscheidungsstrukturen von Menschen nachzubilden mithilfe von Algorithmen. Manche Experten behaupten aber auch, dass eine KI nicht gleich eine KI ist, sondern sie in zwei Bereiche unterteilt werden kann. Dabei wird die KI zwsichen einer starke KI und einer schwache KI unterschieden. Schwache KIs sind nur in bestimmten Teilbereichen in der Lage an die menschliche Intelligenz heranzukommen. Doch sie ist nicht anpassungsfähig. Das bedeutet, dass eine schwache KI, die zum Beispiel auf Schach spezialisiert wird, nur die Intelligenz im Bereich Schach besitzt. Selbstständiges fahren ist mit der KI nicht möglich. Dafür muss eine andere KI spezialisiert werden, damit sie selbstständig fahren kann. Dagegen kann eine starke KI sich an derzeitigen Situatinen anpassen. Diese nennen sich auch „Superintelligenz". Solche starke KIs sind heutzutage noch nicht umsetzbar und es wird immer noch daran geforscht. Die schwache KI, die in der Facharbeit speziell zur Erkennung von handgeschriebenen Zahlen programmiert wird, können in 2 Teilbereichen unterteilt werden. Eine Unterkategorie der KI wäre der

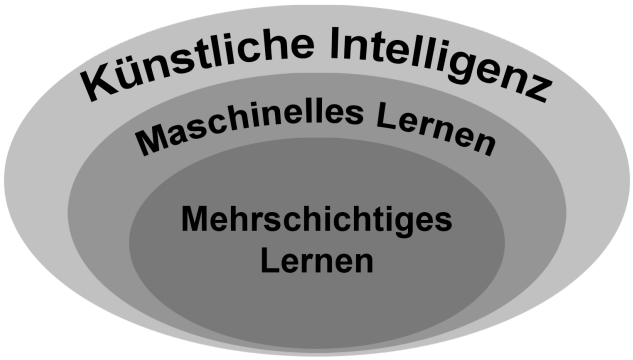

Abbildung 1: *Unterkategorien der Künstliche Intelligenz.*

Bereich Maschinelles Lernen. Maschinelles Lernen versucht mithilfe von Algorithmen mit

einer großen Anzahl von Daten ein Muster zu erkennen. Dazu werden die aus Daten gewonnen Erkenntnisse für Problemlösung verwendet oder der Algorithmus wird auf die Daten zugeschnitten und verallgemeinert. Der zweite Teilbereich wäre mehrschichtiges Lernen, das sehr an maschinelles Lernen ähnelt. Der Unterschied liegt aber an der Art und Weise, wie sie mit den Daten arbeitet. Mehrschichtiges Lernen versucht das menschliche Gehirn zu imitieren durch ein künstliches Neuronales Netzwerk. Durch das Imitieren sind die Strukturen vom mehrschichtigen Lernen deutlich komplexer als die vom maschinellen Lernen. Deswegen sind sehr große Datensätzen von Vorteil für mehrschichtiges Lernen, da deutlich mehr Merkmale der Daten erkannt werden. Darauf basiert sich die KI in der Facharbeit.

2.1 Überwachtes Lernen

Überwachtes Lernen ist eine Methode, wie die KI mit Daten lernen kann. Dazu hat die Datei 2 wichtige Merkmale, nämlich die Eingabe und die Ausgabe. Die Eingabe wird an die KI zum Lernen geschickt und dabei gibt die KI ein Ergebnis aus. Mit der Ausgabe der Datei wird nun überprüft, ob die Antwort, die die KI ausgegeben hat, übereinstimmt. Falls sie nicht übereinstimmt, lernt die KI aus den Fehlern und verbessert sich dadurch.

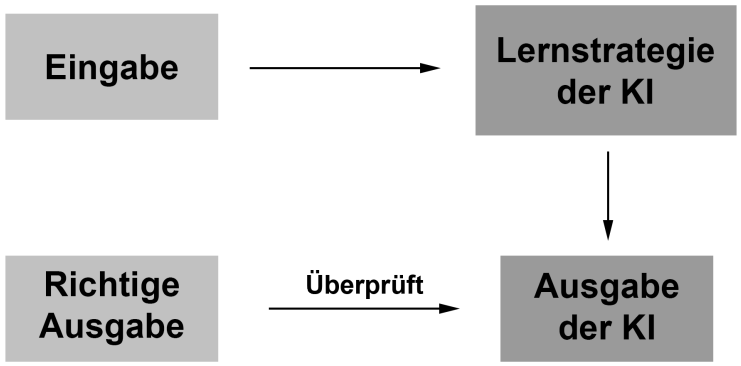

Abbildung 2: *Ablauf eines überwachtes Lernen.*

Überwachtes Lernen ist eine angemessene Methode für das Problem der Facharbeit, da die Antwort, die die KI ausgibt, übereinstimmen muss. Dadurch wird die Fehlerquote der KI möglichst kleingehalten, sodass eine genaue Erkennung der handgeschrieben Zahlen gelingen kann. Dabei ist anzumerken, dass die KI zwei Arten von Lösungen angeben kann. In dem Fall wird die sogenannte Klassifikation als Lösungsart aktzeptiert. Jede Zahl wird als eine eigene Klasse zugeordnet. Die KI gibt aus, um welche Klasse es sich handelt.

3 Voraussetzung für mehrschichtiges Lernen

Irgendetwas schreiben für die Voraussetzung

3.1 Datensatz

Datensätze sind für die KIs das wichtigste Werkzeug für das Lernen. Denn die Qualität des Datensatzen bestimmt die Lerneffizienz der KI. Je schlechter die Qualität ist, desto schlechter lernt auch die KI. Wichtige Merkmale für Qualität ist die Auflösung, die Anzahl der Daten und die Varibialität. Für mehrschichtiges Lernen wird ein großer Datensatz gebraucht. Dabei wird für die Problemlösung der MNIST Datensatz von Yann LeCun gewählt. Sie besitzt insgesamt 70000 Bilder mit handgeschrieben Zahlen, die die Werte zwischen 0 und 9 besitzt. Jedes Bild hat ein 28×28 Pixel Format und ist farblos. Sie wird nur mit der Hellgikeit dargstellt zwischen 0 und 255. Außerdem hat jedes Bild einen Wert, um welche Zahl es sich handelt, damit später geprüft werden kann, ob die KI die Zahl richtig erkannt hat oder ob sie falsch liegt. Der Datensatz ist gut zum Lernen, denn sie

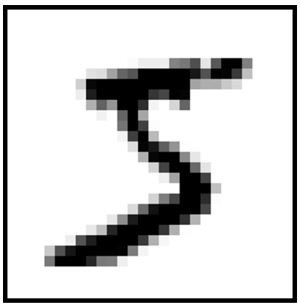

Abbildung 3: *Handgeschriebene Zahl als Abbildung in 28×28 Pixeln dargestellt, dabei ist die richtige Ausgabe die Zahl 5.*

besitzt keine Fehlwerte oder verzerrende Bilder, die die KI beim Lernen stören kann. Auch ist die Auflösung in Ordnung und 70000 Bilder sind für kleine Aufgaben angemessen.

3.2 Künstliches neuronales Netzwerk

Im menschlichen Gehirn befinden sich großen Mengen von Neuronen, die miteinander interagieren. Das Neuron besteht aus drei relevante Komponente. Der erste Komponent besteht aus Empfängern, die Signale von anderen verbundeten Neuronen empfängt. Sie werden auch Dendriten genannt. Ein weiterer Komponent wären die Synapsen. Die Aufgabe ist es sogenannte Neurotransmitter an anderen verbundenen Neuronen weiterzuleiten, falls das Signal ankommen sollte. Damit das Signal ankommt, braucht es den dritten Komponent, zwar den Axon. Sie wird als Leitungsbahn verwendet, um Signale transportieren zu können und besteht aus einer lange dünne Röhre. Diese Röhre kann das Signal zu den Synapsen weiterleiten, unter Voraussetzung, dass der Schwellenwert erreicht ist. Fall er nicht erreicht werden sollte, wird das Signal abgebrochen. Der Schwellenwert wird erreicht, wenn genug Neurotransmitter vorhanden sind WAS ABSOLUT EINFACH DARGSTELLT IST UND EIGENTLICH VERBESSERT WERDEN MUSS, ABER ICH FRAGE ZUR SICHERHEIT NOCHMAL NACH. Sobald das Signal ankommt,

werden die Neurotransmitter ausgeschüttet und die Dendriten der anderen verbunden Neuronen nehmen sie auf. Dadurch ändert sich die Struktur des Neurons und es kommt wieder zur Entscheidung, ob das Signal weitergeleitet werden soll, aber abgebrochen wird. Es ensteht eine Kettenreaktion. Da das menschliche Gehirn imitiert werden soll, werden künstliche Neuronen erschaffen, die einen Wert von den vorherigen künstlichen Neuronen verarbeitet wurden. Durch die ganzen Verbindungen zwischen den Neuronen wird das auch als Netzwerk bezeichnet. Viele Eigenschaften der Neuronen werden mitgenommen, wie zum Beispiel die Dendriten, die Synapsen, sowie der Schwellenwert, in der Informatik auch Bias genannt. Der Bias spielt in Kapitel FORWARD NETWORK eine Rolle. Unterschied ist aber, dass die künstlichen Neuronen deren Struktur nicht ändern können, dafür können denen aber Werte gegeben werden. Das bedeutet, dass das künstliche Neuron negative Werte annehmen kann, was bei den Neuronen nicht der Fall ist, da der Axon nur das Signal weiterleiten oder abbrechen kann. Sie können auch mit 0 und 1 bezeichnet werden. Das Netzwerk kann unterschiedliche Anordnungen von künstlichen

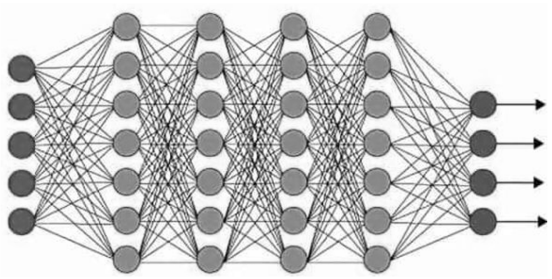

Abbildung 4: *Künstliches Neuronales Netz mit insgesamt 37 künstliche Neuronen in 6 Schichten verteilt.*

Neuronen besitzen. Die geeignetste Anordnung nennt sich das konvolutionales neuronales Netzwerk (CNN), denn sie wird häufig für Bildererkennung verwendet. Grund dafür ist die Extrahierung der Merkmale des Bildes. Denn bevor das Bild an die künstlichen Neuronen weitergeleitet werden, werden unnötige Merkmale mithilfe von Konvolution aus dem Bild entfernt, sodass der Lernprozess der KI auf das Wichtigste begrenzt wird.

4 Konvolutionales neuronales Netzwerk

Das CNN kann in zwei Bereichen gegliedert werden. Der erste Bereich nennt sich Merkmalsextraktion und ist dafür zuständig, die wichtige Merkmale des Bildes herauszufiltern. Im zweiten Bereich werden die wichtigen Merkmalen aus der Merkmalsextraktion in die sogenannte Klassifikationsebene weitergeleitet. Dort fängt der Lernprozess der KI an, die dann am Ende des Netzwerkes eine Klasse ausgibt.

4.1 Klassifikationsebene

In der Klassifikationsebene befindet sich das künstliches neuronales Netzwerk, die eine ähnliche Struktur wie bei Abbildung 4 hat. Diese Struktur nennt sich auch Fully Connected Neural Network und ist in mehreren Schichten aufgebaut. In Abbildung 4 wäre die Anzahl der Schichten $A = 6$. Die erste Schicht nennt sich Eingabeschicht, denn von dort kommen die Daten in die Neuronen rein, während bei der letzte Schicht die KI ein Ergebnis ausgibt. Diese Schicht heißt Ausgabeschicht. Die Schichten dazwischen werden ausgeblendete Schichten genannt. Die ausgeblendete Schichten sind wichtige Teile des Netzwerkes, denn durch diese werden Merkmale gesucht und analysiert. Das Netzwerk läuft also von links nach rechts durch. Das gibt nochmal eine deutlich stärkere Lerntiefe für die KI. Die einzelne Neuronen können einen Wert besitzen, sie wird als $x_i^{(\alpha)}$ definiert, wobei $\alpha \in \{1, \ldots, A\}$ gilt. Der Index i beschreibt um welches Neuron in der Schicht α es sich handelt. Dieser Wert lässt sich aus den ganzen Verbindungen der vorherigen Schicht berechnen, indem sie wie in Abbildung 5 addiert werden. Dabei werden die einzelne Werte $x_j^{(\alpha-1)}$ mit einem Gewicht $w_{i,j}^{(\alpha)}$ multipliziert, womit $j \in \{1, \ldots, N_{\alpha-1}\}$ gilt. N_α beschreibt die Anzahl der Neuronen auf der Schicht α. Allgemein gilt dann für die Berechnung

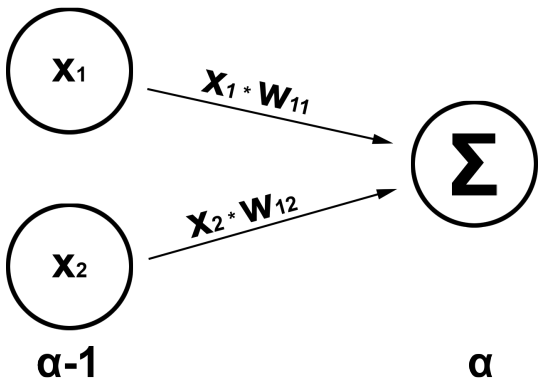

Abbildung 5: *Verbindungen zwischen den Neuronen.*

$$\bar{x}_i^{(\alpha)} := \left(\sum_{j=1}^{N_{\alpha-1}} w_{i,j}^{(\alpha)} x_j^{(\alpha-1)}\right) + b_i^{(\alpha)} \quad i \in \{1, \ldots, N_\alpha\}, j \in \{1, \ldots, N_{\alpha-1}\}. \tag{4.1}$$

Das $b_i^{(\alpha)}$ in der Gleichung steht für Bias und ist ein weiterer Parameter zum Einstellen wie das Gewicht. Beide spielen hier eine wichtige Rolle für das Weiterleiten der Neuronen. Das Gewicht entscheidet, wie wichtig der Wert vom Neuron ist. Je höher der Wert ist, desto entscheidener ist das Neuron. Der Bias stellt ein, wie leicht das Neuron sich aktiviert. Auch hier gilt, je größer der Wert, desto entscheidener ist das Neuron. Wichtig ist anzumerken, dass die Gewichte und die Biases verstellbare Parameter sind, die für das Lernen der KI eine wichtige Rolle spielt. Eine weitere Sache, die hinzugefügt werden muss, ist die Aktivierungsfunktion, denn sie gibt das Netzwerk eine Komplexität. Ohne diese Komplexität wäre das Netzwerk linear, was Nachteile mit sich bringt für die KI. In Abbildung 6 ist die Regression bei einer nicht lineare Struktur deutlich besser als dies

einer lineare Struktur. Welche Aktivierungsfunktionen verwendet werden kommt in Lektion

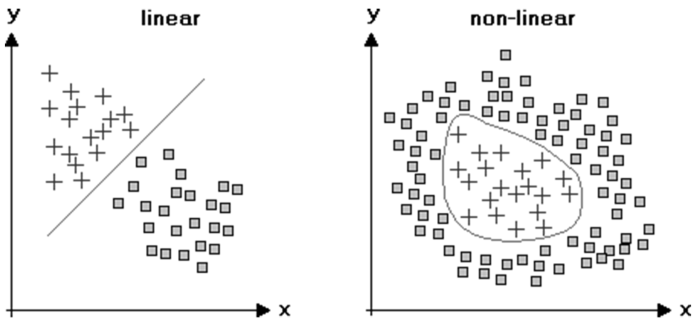

Abbildung 6: *In den ersten beiden Graphen sind die noch gut mit linearen Regressionen lösbar. Beim dritten Graph ist es unmöglich eine genaue Regression zu bekommen mit der lineare Regression. Dafür wird eine nicht linieare Regression verwendet.*

AKTIVIERUNGSFUNKTION dran. So ergibt sich

$$x_i^{(\alpha)} := f^{(\alpha)}(\bar{x}_i^{(\alpha)}) = f^{(\alpha)}((\sum_{j=1}^{N_\alpha} w_{i,j}^{(\alpha)} x_j^{(\alpha-1)}) + b_i^{(\alpha)}). \tag{4.2}$$

Nun kann die Schicht auch als einen Vektor angesehen werden, wobei die Neuronen die Elemente vom Vektor sind. Folglich kann das Netzwerk in Abbildung 4 in einem Matrixblock umgeschrieben werden. So definiert sich

$$\vec{x}^{(\alpha)} := \begin{bmatrix} x_1^{(\alpha)} \\ x_2^{(\alpha)} \\ \dots \\ x_{N_\alpha}^{(\alpha)} \end{bmatrix}. \tag{4.3}$$

Sobald die letzte Schicht erreicht ist, berechnet die Verlustfunktion Fehlermaße des Netzwerkes und gibt Aussage darüber, wie gut unsere KI ist. Mehr dazu im Kapitel VERLUST-FUNKTION.

4.1.1 Aktivierungsfunktion

Mithilfe der Aktivierungsfunktion kann die KI eine nicht lineare Regression durchführen, denn aus der Gleichung 4.2 wird deutlich, dass alle Operationen linear sind, wenn nur der Paramater der Funktion $f^{(\alpha)}(x)$ betrachtet wird. Für eine nicht lineare Regression sollte die Aktivierungsfunktion differenzierbar sein. Dafür gibt es verschiedene Formen von Aktivierungsfunktionen, die in der Praxis angewendet werden. Eine der meist verwendeten Aktivierungsfunktionen ist der Rectified linear unit (ReLU):

$$f^{(\alpha)}(\bar{x}_i^{(\alpha)}) = \begin{cases} \bar{x}_i^{(\alpha)} & \text{falls } \bar{x}_i^{(\alpha)} > 0 \\ 0 & \text{falls } \bar{x}_i^{(\alpha)} \leq 0 \end{cases} \tag{4.4}$$

Gründe für die Nutzung ist die Effizienz für linear gebaute Neuronale Netzwerke, sowie die Berechnung der Werte. Außerdem kann durch ReLU Neuronen dauerhaft deaktiviert werden, da alle Werte, die ≤ 0 sind, zu 0 gesetzt werden. Eine weitere Aktivierungsfunktion, die für die Facharbeit verwendet wird ist die Softmax Funktion:

$$f^{(\alpha)}(\bar{x}_i^{(\alpha)}) = \frac{e^{\bar{x}_i^{(\alpha)}}}{\sum_{j=1}^{N_\alpha} e^{\bar{x}_i^{(\alpha)}}} \tag{4.5}$$

Meistens wird die Aktivierungsfunktion an der letzte Schicht verwendet, denn der Zweck dieser Aktivierungsfunktion ist es mithilfe des Vektors $\vec{x}^{(A)}$ in einen neuen Vektor gleicher Dimension zu erstellen, deren Elemente zusammenaddiert 1 ergibt. Der Wertebereich der Funktion liegt bei $[0;1]$. Dies kann dann in einer Wahrscheinlichkeitsverteilung dargestellt werden. Dies erleichtert die Dokumentierung der Ergebnisse der KI und eine Rückschlussführung, warum die KI wahrscheinlich diese Zahl nimmt, ist möglich.

4.2 Feature Extraction??/Merkmalsextraktion (DeepL)

Die Merkmalsextraktion besteht aus mehreren Schichten von Operationen mit dem das eingebene Bild verarbeitet wird. Eine Hauptoperation ist der 2D Konvolution, denn diese Operation ist dafür zuständig, dass wichtige Merkmale gefiltert werden. Weitere wichtige Operationen ist der Max-Pooling und der Flattening. Aufgabe von Max-Pooling ist es unnötige Merkmale zu ignorieren, indem nur die hellsten Pixel rausgefiltert werden. Die Dimension des Bildes halbiert sich. Flattening ist für die Übertragung auf die Klassifikationseben zuständig, indem die Matrix des Bildes in einen Vektor geflacht wird, sodass die KI optimal lernen kann. ABBILDUNG (FEATURE EXTRACTION, HIER KÖNNEN WIR JA SCHONMAL ANFANGEN UNSERE KI ZU BAUEN) Das Bild kann in einer 28×28 Matrix dargestellt werden

$$X^{(1,1)} := (x_{ij}) \quad i \in \{1,\dots,28\}, j \in \{1,\dots,28\}. \tag{4.6}$$

Der erste Index beschreibt auf welcher Schicht die Variable sich befindet und der zweite Index beschreibt auf welches Bild sie drauf bezieht, denn in einer Schicht können mehrere Bilder vorkommen. Dies nennt sich Tiefe. Da am Anfang aber nur ein Bild eingegeben wird, gilt $B^{(1)} = 1$. Für die nächsten Schichten gilt im Allgemeinen

$$X^{(\alpha,\beta)} := (x_{ij}) \quad i \in \{1,\dots,m_x^{(\alpha,\beta)}\}, j \in \{1,\dots,n_x^{(\alpha,\beta)}\}. \tag{4.7}$$

Die Werte $m_x^{(\alpha,\beta)}$ und $n_x^{(\alpha,\beta)}$ werden basiert auf die Operation berechnet, wobei $\alpha \in \{1,\dots,A\}$ und $\beta \in \{1^{(\alpha)},\dots,B^{(\alpha)}\}$. Dabei ist die Anzahl der Tiefe von der jeweilige Schicht abhängig.

4.2.1 2D Konvolution

Mithilfe von 2D Konvolution können wichtige Merkmale herausgefiltert werden. Damit sie aber herausgefiltert werden kann, wird eine neue Matrix definiert, die auch als Kernel

genannt wird

$$K^{(\alpha,\beta)} := (k_{i,j})^{(\alpha,\beta)} \quad i \in \{1;2;\ldots;m_k^{(\alpha,\beta)}\}, j \in \{1;2;\ldots;n_k^{(\alpha,\beta)}\} \tag{4.8}$$

Wenn der Kernel nach bestimmten Mustern definiert wurde, entsteht durch die 2D Konvolution eine neue Matrix, in dem Fall ein neues Bild ABBILDUNG CONVOLUTION Diese neue Matrix wird definiert als

$$A := (a_{i,j}) \quad i \in \{1,\ldots,m_a\}, j \in \{1,\ldots,n_a\}, \tag{4.9}$$

wobei $m_a = m_x - m_k + 1$ und $n_a = n_x - n_k + 1$ Da sowohl die Eingabe des Bildes, sowie der Kernel beide Matritzen sind, kann mit der 2D Konvolution gerechnet werden. Sie ist folgendermaßen defininiert

$$a_{i,j} = (X * K)_{i,j} := \sum_{m'=1}^{m_k} \sum_{n'=1}^{n_k} x_{i-m'+1,j-n'+1} k_{m',n'}. \tag{4.10}$$

Eine weitere Operation, die sehr wichtig für die 2D Konvolution ist der Cross Correlation

$$a_{i,j} = (X \star K)_{i,j} := \sum_{m'=1}^{m_k} \sum_{n'=1}^{n_k} x_{i+m'-1,j+n'-1} k_{m',n'}. \tag{4.11}$$

Aus (4.10) wird deutlich, dass Cross Correlation nichts anderes ist als eine 2D Konvolution, wo nur der Kernel um 180° rotiert ist. Dies wird später für die Fehlerrückführung in Kapitel 6.4.1 wichtig sein. Wenn die Schichten und die Tiefen der Merkmalsextraktion berücksichtigt werden, so gilt für die nächste Schicht

$$x_{i,j}^{(\alpha,\beta)} = f^{(\alpha)}((\sum_{\beta'=1}^{B^{(\alpha-1)}} \sum_{m'=1}^{m_k} \sum_{n'=1}^{n_k} x_{i+m'-1,j+n'-1}^{(\alpha-1,\beta')} k_{m',n'}^{(\alpha,\beta')}) + b_{i,j}^{(\alpha,\beta)}) \tag{4.12}$$

4.2.2 Max-Pooling

4.2.3 Flattening??

Damit die Ki mit der Matrix lernen kann, muss diese Matrix in einen Vektor umgewandelt werden

$$\vec{x}^{(\alpha,\beta)} := vec_{n_x^{(\alpha,\beta)},m_x^{(\alpha,\beta)}}(X^{(\alpha-1,\beta)T}) \tag{4.13}$$

4.3 Verlustfunktion

4.4 Optimierung

- Seite 10 bis 12 - Für die Optimierung benutzen wir die ADAM Optimierer

4.4.1 Fehlerrückführung

- Eine Methode, um die Gradiente für die Optimierung zu berechnen

5 Umsetzung

5.1 Werkzeuge

- Seite 14 - Python wird hier verwendet, da gut für Data Science - Hierfür wird NumPy benutzt für Matrix - Auch wird der Matplotlib verwendet für Visualisierungen

5.2 Trainingsphase

- Daten müssen in 3 Sets aufgeteilt werden: Training, Test und Validation - Irgendwo muss noch Overfitting und Underfitting rein, das mit Validation vermieden wird

5.3 Auswertung

- Seite 14 - Verschiedene Grafiken werden ausgewertet - Z.B. Lernverlauf - Accuracy anschauen - Wert der Verlustfunktion im Verlauf anschauen

5.3.1 Trainingsdaten

5.3.2 Validationsdaten

5.3.3 Testdaten

6 Schlussfolgerung

- Seite 19 Paar Formeln, wobei gilt: Linare Funktion zwischen zwei Neuronenschichten:

$$\vec{x}^{(\alpha)} = f^{(\alpha)}(W^{(\alpha)}\vec{x}^{(\alpha-1)} + \vec{b}^{(\alpha)})$$

RelU:

$$x_i^{(\alpha)} := f^{(\alpha)}(\bar{x}_i^{(\alpha)})$$

$$x_i^{(\alpha)} = \begin{cases} \bar{x}_i^{(\alpha)} & \text{falls } \bar{x}_i^{(\alpha)} > 0 \\ 0 & \text{falls } \bar{x}_i^{(\alpha)} \leq 0 \end{cases}$$

Ableitung von RelU:

$$\frac{\partial x_i^{(\alpha)}}{\partial \bar{x}_i^{(\alpha)}} = \begin{cases} 1 & \text{falls } \bar{x}_i^{(\alpha)} > 0 \\ 0 & \text{falls } \bar{x}_i^{'(\alpha)} \leq 0 \end{cases}$$

Softmax:

$$x_i^{(\alpha)} = \frac{e^{\bar{x}_i^{(\alpha)}}}{\sum_{j=1}^{N_\alpha} e^{\bar{x}_i^{(\alpha)}}}$$

Ableitung von Softmax:

$$\frac{\partial x_i^{(\alpha)}}{\partial \bar{x}_k^{(\alpha)}} = \begin{cases} x_i^{(\alpha)}(1 - x_k^{(\alpha)}) & \text{falls } i = k \\ -x_i^{(\alpha)} x_k^{(\alpha)} & \text{falls } i \neq k \end{cases}$$

Cross Entropy:

$$C(\vec{y}, \vec{x}^{(A)}) = -\sum_{i=1}^{N_A} y_i * \ln(x_i^{(A)})$$

Ableitung von Cross Entropy:

$$\frac{\partial C}{\partial x_i^{(A)}} = -\frac{y_i}{x_i^{(A)}}$$

Backpropagation Bias FNN:

$$\frac{\partial C}{\partial b_i^{(\alpha)}} = \frac{\partial C}{\partial \bar{x}_i^{(\alpha)}}$$

Backpropagation Weight FNN:

$$\frac{\partial C}{\partial w_{i,j}^{(\alpha)}} = \frac{\partial C}{\partial \bar{x}_i^{(\alpha)}} x_j^{(\alpha-1)}$$

5 Backpropagation Error FNN:

$$\frac{\partial C}{\partial x_j^{(\alpha-1)}} = \sum_{i=1}^{N_\alpha} \frac{\partial C}{\partial \bar{x}_i^{(\alpha)}} w_{i,j}^{(\alpha)}$$

Literaturverzeichnis

[1] Europäisches Parlament: Was ist künstliche Intelligenz und wie wird sie genutzt?. 2021. Stand: 24.11.2022. https://www.europarl.europa.eu/news/de/headlines/society/20200827STO85804/was-ist-kunstliche-intelligenz-und-wie-wird-sie-genutzt [1]

[2] Uni Oldenburg: Schwache KI und Starke KI. 2008/2009. Stand: 24.11.2022. http://www.informatik.uni-oldenburg.de/~iug08/ki/Grundlagen_Starke_KI_vs._Schwache_KI.html [1]

[3] Yann LeCun, Corinna Cortes, Christopher J.C. Burges: THE MNIST DATABASE of handwritten digits. Stand: 24.11.2022. http://yann.lecun.com/exdb/mnist/

[4] Bundesministerium für Wirtschaft und Energie: Zur Diskussion der Effekte Künstlicher Intelligenz in der wirtschaftswissenschaftlichen Literatur. 2018. Stand: 24.11.2022. https://www.bmwk.de/Redaktion/DE/Downloads/Diskussionspapiere/20190205-diskussionspapier-effekte-kuenstlicher-intelligenz-in-der-wirtschaftswissenschaftlichen-literatur.pdf?__blob=publicationFile&v=6

[5] Michael Nielsen: Neural Network and Deep Learning. 2019. Stand: 24.11.2022. http://neuralnetworksanddeeplearning.com/

[6] My Great Learning: Types of Neural Networks and Definition of Neural Network. 2022. Stand: 24.11.2022. https://www.mygreatlearning.com/blog/types-of-neural-networks/

[7] Chi Nhan Nguyen, Oliver Zeigermann: Machine Learning kurz & gut O'Reillys Taschenbibliothek. 2. Auflage. 2021

[8] Musstafa: Optimizers in Deep Learning. 2021. Stand 24.11.2022. https://medium.com/mlearning-ai/optimizers-in-deep-learning-7bf81fed78a0

[9] Bui Anh Minh Leon Phan: handwrittenDigits. 2022. Stand: 24.11.2022. https://github.com/xXChezyXx/handwrittenDigits

[10] Bundeswettbewerb KI: Künstliche Intelligenz. 2022. Stand: 24.11.2022. https://www.bw-ki.de/

Schülererklärung

Hiermit erkläre ich, dass ich die vorliegende Seminarfacharbeit selbstständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und die Stellen der Seminarfacharbeit, die im Wortlaut oder im wesentlichen Inhalt aus anderen Werken entnommen wurden, mit genauer Quellenangabe kenntlich gemacht habe.

7 Zusatz: ChatGPT