h_da
HOCHSCHULE DARMSTADT
UNIVERSITY OF APPLIED SCIENCES
fbi
FACHBEREICH INFORMATIK

Neurale Netzwerke zur Bildklassifikation in Tensorflow

Luise Wiesalla

Computer Vision Sommersemester 2020

1 Einleitung

Seit einigen Jahren profitieren viele Computer Vision Anwendungen von den breit einsetzbaren *Convolutional Neural Networks* (CNN). Die CNNs sind eine Weiterentwicklung herkömmlicher *Dense Neural Networks* (DNN) mit zusätzlichen Faltungsschichten. Durch diesen Zusatz sind die CNNs besonders für die Bilderkennung geeignet.

Für einen einfachen und abstrahierten Zugang wird in dieser Arbeit das Framework Tensorflow für die Erstellung von neuronalen Netzen benutzt.

Ziel dieser Arbeit ist die Realisierung von Neuronalen Netzen (DNN und CNN) zur Klassifikation des MNIST Datensatzes, sowie die Anwendung weiterführender Funktion im Bereich Data Augmentation und der Visualisierung der Zwischenschritte im Netz.

2 Vorüberlegungen und Konzept

Für die Umsetzung wird Tensorflow 2.1.0 mit der integrierten Keras-API verwendet. Der ausgewählte Datensatz MNIST (siehe hierzu [LeCB12]) ist frei zugänglich. Die Bilddaten enthalten 70.000 handschriftliche Ziffern als 28 x 28 Pixel Grauwertbild und sind in ein Trainings- und Validierungsset gegliedert.

Zunächst wird ein DNN erstellt, welches die MNIST-Daten pixelweise analysiert. Hierbei wird ein Bild in einen mehrdimensionalen Featurevektor (1x784) überführt.

Im weiteren Verlauf wird ein CNN für die Klassifikation erstellt. Die Pixel als Features behalten hier ihre räumliche Anordnung und gewinnen durch die Faltungs- und Max-Pooling-Schicht eine relative Skalierungs- und Translationsinvarianz. Diese theoretischen Überlegungen sind in [L+15] ausgeführt und sollen in der praktischen Umsetzung nachgewiesen werden. Hierbei werden die Bildzentren von 10 Bilder zufällig in x und y um einen zufälligen Betrag aus dem Intervall [-8, 8] verschoben. Anschließend werden die Daten mit dem CNN und zum Vergleich auch mit dem DNN klassifiziert.

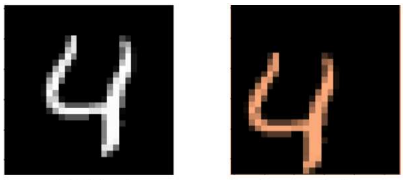

Abbildung 1 Beispiel verschobenes Bildzentrum um jeweils 4 Pixel nach unten und links

Schließlich werden die Zwischenschritte eines CNN mit 3 Farbkanälen visualisiert, in dem die Ausgaben nach der ersten Convolutional Layer nach der Methodik von [Brow19] selektiert und visualisiert werden.

3 Realisierung in Tensorflow

Für die praktische Umsetzung wird ein interaktives Jupyter Notebook angelegt. Der Aufbau des einfachen Neuronalen Netzwerk ohne Convolutional Layer ist wie folgt:

Schicht	Output	Gewichte
Flatten Layer	784 x 1	0
Dense Layer	128 x 1	100480
Dense Layer	10 x 1	1290

Abbildung 2 Aufbau DNN mit Angabe der Output-Dimensionen und trainierbaren Gewichte (mit Bias)

Als Eingabewerte dienen die Grauwerte der 784 Pixel. Das Training mit jeweils 60.000 Bilder in 5 Epochen kann in 23s abgeschlossen werden. Die Korrektklassifizierungsrate auf dem Validierungsset beträgt 85,86%. Mit einer weiteren Schicht kann die Genauigkeit auf 96,14% - mit einer Trainingszeit von 64s - gesteigert werden.

Bei den verschobenen Ziffern werden 5 von 10 erkannt. Die nicht erkannten Ziffern und deren falsche Prognose sind in Abb. 3 dargestellt.

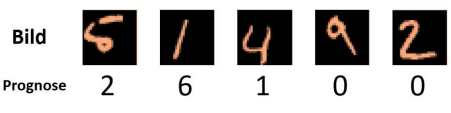

Abbildung 3 Falsch klassifizierte augmentierte Bilder

Luise Wiesalla | 738810 | Luise.Wiesalla@stud.h-da.de

Für die Klassifikation des MNIST-Datensatzes mit einem CNN wird folgender Aufbau verwendet:

Schicht	Output	Gewichte
Convolutional Layer 5x5x32	28 x 28 x 32	832
Max-Pooling 6x6	14 x 14 x 32	0
Convolutional Layer 5x5x64	14 x 14 x 64	51264
Max-Pooling 4x4	7 x 7 x 64	0
Flatten Layer	3136 x 1	0
Dense Layer	720 x 1	2258640
Dense Layer	10 x 1	7210

Abbildung 4 Aufbau CNN mit Parameteranzahl mit einem Padding bei der Faltung und bei Max-Pooling

Hierbei werden in 2 Epochen mit einer Trainingszeit von 596s eine Korrekt-klassifizierung des Validierungsset von 96,84% erzielt.
Bei der Klassifikation, der in x- und y-Richtung verschobenen Daten werden 8 von 10 Bildern richtig klassifiziert. Ähnlich wie das DNN werden die Ziffern 4 und 9 falsch klassifiziert.

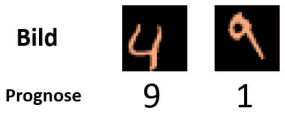

Abbildung 5 falsch klassifizierte augmentierte Bilder CNN

Neben dem MNIST Datensatz wird noch ein weiteres CNN erstellt, welches Farbbilder des CIFAR Datensatzes klassifiziert. Für die Visualisierung der *Feature Map* wird das Modell vor der ersten Max-Pooling-Layer abgeschnitten. Das Ergebnis der ersten Convolutional Layer wird zu der neuen Modellausgabe und kann visualisiert werden.

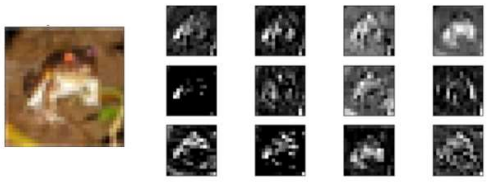

Abbildung 6 Visualisierung der Feature Maps

4 Diskussion der Ergebnisse

Bei der Klassifikation des MNIST Datensatzes schneidet das CNN nur um 0,7% besser als das DNN mit zwei Schichten ab. Das DNN benötigt für die ähnliche Genauigkeit jedoch nur $\frac{1}{9}$ der Trainingszeit. Auf der Website des MNIST-Datensatzes [LeCB12] werden die Test-Fehlerrate vieler Modelle verglichen. Das aktuell beste Modell basiert auch hier auf Faltungen.

Bei der Prognose von augmentierten Daten punkten schneidet das CNN ebenfalls besser ab. Das DNN besitzt vom Aufbau her keine besondere Translationsinvarianz. Trotzdem werden 5 von 10 Bildern richtig klassifiziert. Dies könnte an der (mehrere Pixel) breite Linie der Ziffern und der durch den Zufall teilweise auch nur geringfügige Verschiebung liegen.

Das CNN kann in vieler Hinsichten weiter optimiert werden. Die Wahl der Filtergröße und die Evaluierung verschiedener Aktivierungsfunktionen sind nur einige Beispiele für die Optimierung der Hyperparameter. In einer Praxisanwendung sollten Schritte zur Vermeidung einer Überanpassung (z.B. Drop-Out-Schicht) unternommen werden.

Bei Betrachtung der Featuremap des CIFAR-Netzes werden herausgearbeitete Strukturen sichtbar. Eine weitere Verfolgung durch das Netz ist durch den Aufbau sehr erschwert. Mögliche Ansätze zur Lösung finden sich unter dem Schlagwort „ExplainableAI“ wieder.

Insgesamt konnte ein gründlicher Einblick in den Aufbau, das Training und die Data Augmentation mit Tensorflow gewonnen werden.

5 Quellen

[Brow19] Brownlee J.: How to Visualize Filters and Feature Maps in Convolutional Neural Networks 2019, https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/, Abruf am 03.08.2020

[L+15] Quoc V Le et al.: A tutorial on deep learning part 1 - Nonlinear classifiers and the backpropagation algorithm, 2015.

[LeCB12] LeCun Y.; Cortes C.; Burges C.: The MNIST Database of handwritten digits. 2012, http://yann.lecun.com/exdb/mnist/ , Abruf am 03.08.2020