DHBW
Duale Hochschule
Baden-Württemberg
Ravensburg
Campus Friedrichshafen

Entwerfen und Verbessern eines neuronalen Netzes zur Erkennung von Daten aus dem MNIST-Datensatz

Verbesserung hinichtlich der Genauigkeit (Accuracy)

Projektarbeit Modul Digitale Bildverarbeitung und Musterkennung

Studiengang Elektrotechnik

Studienrichtung Fahrzeugelektronik

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Daniel Schmid

Abgabedatum:	05.01.2024
Bearbeitungszeitraum:	25.10.2023 - 05.01.2024
Matrikelnummer:	7749394
Kurs:	TFE21-2

Erklärung

gemäß Ziffer 1.1.14 der Anlage 1 zu §§ 3, 4 und 5 der Studien- und Prüfungsordnung für die Bachelorstudiengänge im Studienbereich Technik der Dualen Hochschule Baden-Württemberg vom 29.09.2017 in der Fassung vom 24.07.2023.

Ich versichere hiermit, dass ich meine Projektarbeit Modul Digitale Bildverarbeitung und Musterkennung mit dem Thema:

> *Entwerfen und Verbessern eines neuronalen Netzes zur Erkennung von Daten aus dem MNIST-Datensatz*

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Ich versichere zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Musterstadt, den 5. Januar 2024

_____

Daniel Schmid

Inhaltsverzeichnis

1 Einleitung und Ziel

Ein fundamentales Anwendungsgebiet der Künstlichen Intelligenz ist die Bilderkennung, die in der fortschreitenden Technik in vielen Bereichen eine zentrale Rolle spielt. Denkt man beispielsweise an das Fortbewegen mit autonom fahrenden Fahrzeugen wird klar, dass das Erkennen von Zahlen auf Verkehrsschildern von hoher Bedeutung ist. In dieser Hausarbeit steht die Konzeption und Umsetzung eines einfachen neuronalen Netzwerks im Fokus, das für die Klassifizierung von handgeschriebenen Ziffern des MNIST-Datensatzes entworfen wurde. Der MNIST-Datensatz enthält 60.000 Ziffern im Trainingsdatensatz und weitere 10.000 im Testdatensatz. Abbildung 1.1 zeigt einen Ausschnitt aus dem Trainingsdatensatz.[Wik22]

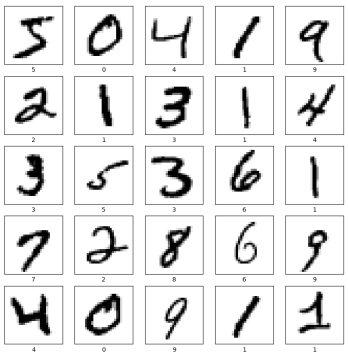

Abbildung 1.1: Ausschnitt der im MNIST-Datensatz vorhandenen Trainingsdaten

Das vorgestellte Netzwerk wird auf der Tensorflow-Plattform mithilfe der keras-Bibliothek implementiert. Die Struktur und Funktionsweise des neuronalen Netzes werden detailliert erläutert und anschließend durch verschiedene graphischen Darstellungen analysiert. Das Hauptaugenmerk liegt dabei auf der Accurancy und Loss-Funktion. Zunächst wird nun das Basisnetz aufgezeigt, welches es hinsichtlich der Accurancy zu verbessern galt.

2 Basisstand des neuronalen Netzes

Der Basisstand des neuronalen Netzes, wie es in der Datei 'TheseAreNotTheSolutionsYouAreLookingFor.ipynb' definiert wurde, weist einen Verlauf der Accuracy wie in Abbildung 2.1 gezeigt auf. Es ist zu erkennen, dass nach etwa 3 Epochen die Accuracy der Trainingsdaten die der Testdaten überschreitet. Dies lässt sich ebenso im Loss-Verlauf erkennen (hier unterschreitet die Trainingskennlinie die Testkennlinie).

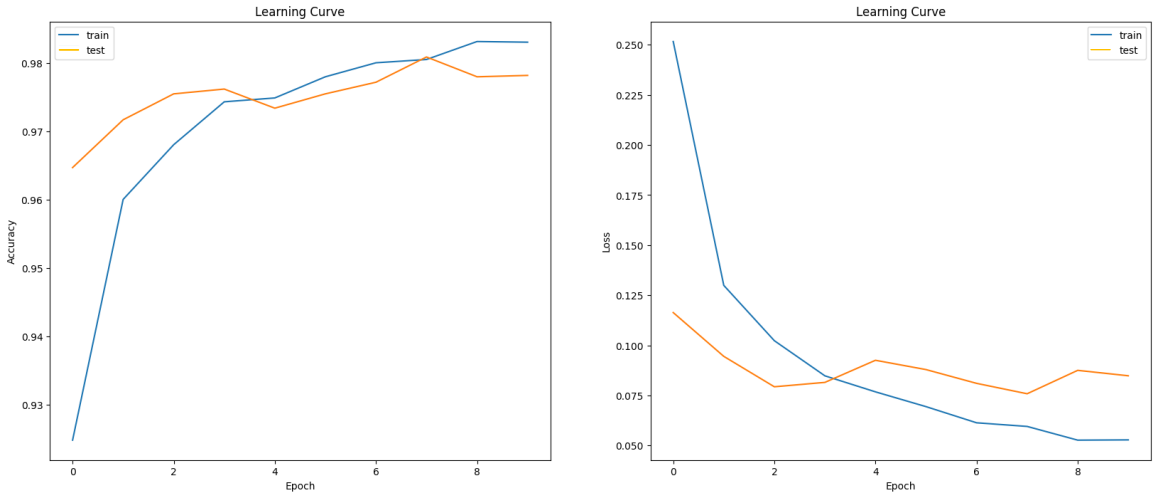

Abbildung 2.1: Accuracy und Loss-funktion des Basisnetzes über 10 Epochen

Das Netz wurde hier mit den Größen der Kennzahlen Epochen=10 und Batchsize=8 trainiert. Die Architektur des Modells bestand aus einer Flatten-Layer, zwei Dense-Layern und einer Dropout-Layer, die wie auf folgender Seite gezeigt definiert wurden.

```
marvin = tf.keras.models.Sequential([
tf.keras.layers.Flatten(input_shape=(28,28,1 )),
tf.keras.layers.Dense(128, activation='relu'),
tf.keras.layers.Dropout(0.2),
tf.keras.layers.Dense(10)
])
```

3 Grundgedanken zur Verbesserung des neuronalen Netzes

Wie in Abbildung 2.1 aus Kapitel 2 zu erkennen ist, übersteigt die Kennlinie der Accurancy des Trainings die des Tests. Dies deutet auf Overfitting hin. Overfitting bedeutet, dass das Modell eine sehr hohe Accuracy bei den Trainingsdaten erreicht, jedoch relativ schlecht abschneidet, sobald dem Modell neue/ bisher unbekannte Daten (in diesem Fall handgeschriebene Ziffern) zugeführt werden. Das neuronale Netz soll nach Abschluss des Trainings auf noch nicht gesehene Daten mindestens genau so gut reagieren wie auf die Trainingsdaten.[Opp23]
Um eine bessere Leistung bei neuen Daten erzielen zu können gibt es verschiedene Möglichkeiten, das Overfitting zu vermeiden:

- Regularisierungen (Lasso-Regularisierung und Ridge-Regularisierung)
- Dropout (Regularisierungstechnik, um zufällig bestimmte Zahl an Neuronen oder Einheiten zu deaktivieren)
- Early Stopping (Training beenden, sobald Leistung nicht mehr verbessert wird)
- Anpassung der Epochenzahl während des Trainings [23a]

Um die allgemeine Genauigkeit des Modells zu verbessern gibt es die Möglichkeit sogenannte Convolutional Layer einzufügen. Ein gewöhnliches neuronales Netz (Fully-Connected) verarbeitet eine Eingabe indem mehrere Hidden Layer durchlaufen werden. Jeder dieser Schichten besteht aus Neuronen, welche mit allen Neuronen der vorherigen Schicht vernetzt sind. Es ist jedoch nicht möglich mit anderen Bausteinen in der Schicht zu kommunizieren. Der Nachteil dieser Fully-Connected Layer ist, dass sie sich nicht gut skalieren lassen. Ein Convolutional Layer in einem neuronalen Netzwerk erkennt lokale Muster in Daten wie Bildern durch Anwendung von Filtern. Vorteile sind die effektive Mustererkennung, Parameter-Sharing für Effizienz, translatorische Invarianz und hierarchische Merkmalsextraktion, was zu verbesserten Leistungen bei Bildverarbeitungsaufgaben führt. Es gibt verschiedene Layer mit verschiedenen Funktionen, die nun kurz und knapp erklärt werden sollen:

- In einem **Convolutional Layer** wird ein Filter definiert, der bestimmt wie groß die Teilbilder sein sollen. Zudem entscheidet er, mit wie vielen Pixeln zwischen den Berechnungen weitergefahren wird.

- Im **Pooling Layer** erfolgt prinzipiell dieselbe Rechnung wie im Convolutional Layer, mit dem Unterschied, dass vom Ergebnis nur der Durchschnitts- oder Maximalwert übernommen wird.

- Nach den genannten Layern folgt noch ein **Fully-Connected Layer**, der die einzelnen entstandenen Teilbildern wieder miteinander verknüpft, um die Zusammenhänge zu erkennen und die Klassifizierung vorzunehmen.[23b]

Es gibt zudem noch weitere Layer, die die Genauigkeit des Modells verbessern, jedoch nicht zu den Convolutional Layern gehören. Mit der sogenannten Batch Normalization können Aktivierungen unabhängig von der Schicht normalisiert werden. Random Zoom wird verwendet, um zufällige Vergrößerungen und Verkleinerungen auf Bilder anzuwenden. Random Rotation rotiert die Bilder zufällig, um Robustheit des Modells zu verbessern.

4 Verbessertes Neuronales Netz

Durch die Anwendung der eben aufgezeigten Verbesserungsmöglichkeiten konnte das Modell hinsichtlich der Accuracy und Loss, sowie des Problems des Overfittings deutlich verbessert werden. Die Architektur des Modells wurde dafür wie folgt angepasst.

```
marvin = tf.keras.models.Sequential([
tf.keras.layers.InputLayer(input_shape=(28,28,1)),
tf.keras.layers.RandomRotation(0.06),
tf.keras.layers.RandomZoom(0.02),
tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
tf.keras.layers.BatchNormalization(),
tf.keras.layers.MaxPooling2D((2, 2)),
tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
tf.keras.layers.BatchNormalization(),
tf.keras.layers.MaxPooling2D((2, 2)),
tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
tf.keras.layers.BatchNormalization(),
tf.keras.layers.Flatten(),
tf.keras.layers.Dense(64, activation='relu'),
tf.keras.layers.BatchNormalization(),
tf.keras.layers.Dropout(0.7),
tf.keras.layers.Dense(128, activation='relu'),
tf.keras.layers.BatchNormalization(),
tf.keras.layers.Dropout(0.5),
tf.keras.layers.Dense(10, activation='softmax')
])
```

Durch mehrere gleiche Layer mit unterschiedlichen Parametern wird dafür gesorgt, dass das Modell eine größere Repräsentationskapazität hat. Die Fähigkeit des Modells kann erhöht werden, da jeder Layer unterschiedliche Merkmale oder Muster lernt. Abbildung 4.1 zeigt nun den aktualisierten Verlauf der Accuracy und Loss-Funktion.

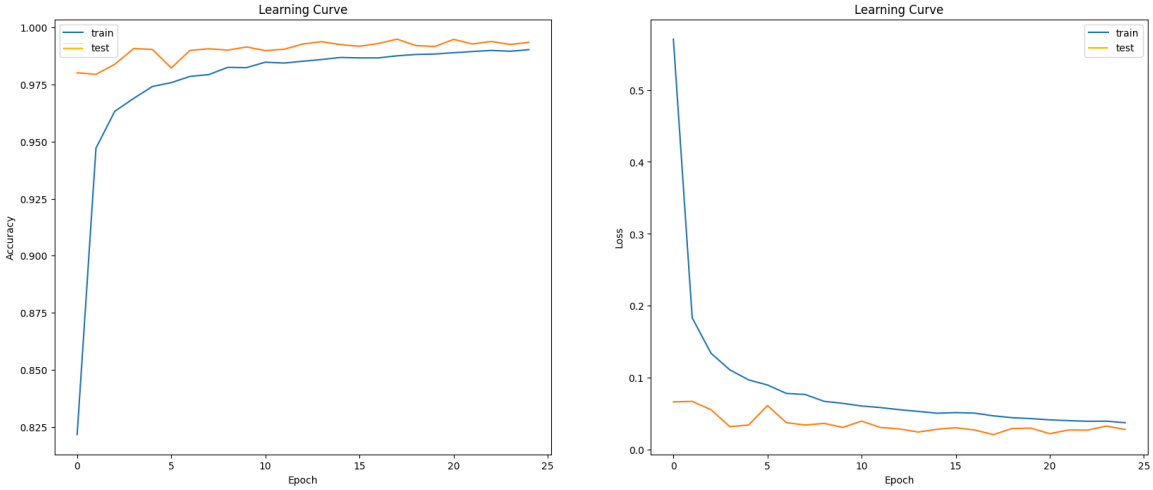

Abbildung 4.1: Accuracy und Loss-funktion des verbesserten Netzes über 25 Epochen

Deutlich zu sehen ist, dass die Trainingskurve der Accuracy-Funktion die der Testkurve nun nicht mehr übersteigt, sondern lediglich annähert. Dies deutet darauf hin, dass das Problem des Overfittings beseitigt wurde. Zudem wurde eine maximale Trainingsaccuracy von $99,02\%$, sowie eine maximale Testaccuracy von $99,34\%$ erreicht. Die Parameter Epochen und Batchsize wurden auf 25 bzw. 64 gesetzt. Da keine großen Schwankungen in der Testkurve vorliegen kann angenommen werden, dass die Parameter Epochen und Batchsize gut gewählt wurden.

5 Zusammenfassung

Ausgehend von einem grundlegenden neuronalen Netzwerk wurde eine Reihe von Verbesserungen vorgenommen, um die Modellleistung zu steigern. Fokus wurde dabei auf die Vermeidung von Overfitting, sowie der Erhöhung der Accuracy gelegt. Zudem wurde eine sorgfältige Auswahl von Data Augmentation-Methoden, wie Random Rotation und Random Zoom, angewendet, um die Robustheit des Modells zu erhöhen. Die Netzwerkarchitektur wurde durch die Einführung von Batch Normalization-Schichten optimiert, um das Training zu stabilisieren und die Konvergenz zu beschleunigen. Durch experimentelle Iteration und Hyperparameteroptimierung wurde die Anzahl und Konfiguration der Convolutional Layers, Dense Layers und weiterer Komponenten des Netzes angepasst. Die Ergebnisse zeigen eine signifikante Verbesserung sowohl in Bezug auf die Reduzierung von Overfitting als auch auf die Steigerung der Accuracy auf einem Validierungsdatensatz. Diese Optimierungen verdeutlichen die Wirksamkeit verschiedener Techniken zur Erzielung eines ausgewogenen Modells, das gut auf neue Daten generalisiert.

Literaturverzeichnis

[23a] März 2023. URL: https://arelium.de/glossar/overfitting/.

[23b] Apr. 2023. URL: https://databasecamp.de/ki/convolutional-neural-network.

[Opp23] Artem Oppermann. *Overfitting und underfitting in Neuronalen Netzen*. Dez. 2023. URL: https://artemoppermann.com/de/overfitting-und-underfitting-in-deep-learning/.

[Wik22] Wikipedia. *MNIST-Datenbank — Wikipedia, die freie Enzyklopädie*. 2022. URL: https://de.wikipedia.org/w/index.php?title=MNIST-Datenbank&oldid=226047788.