Dokumentation

Abgabe digitale Bildverarbeitung

Studiengang Elektrotechnik

Studienrichtung Fahrzeugelektronik

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Robert Köber

Abgabedatum:	05.01.2024
Bearbeitungszeitraum:	12.11.2023-05.01.2024
Matrikelnummer:	3866986
Kurs:	TFE21-2

1 Problemstellung und Zielsetzung

Die MNIST-Datenbank enthält handgeschriebene Ziffern von 0 bis 9, die als Bildpixel repräsentiert sind. In dem Datensatz sind 60000 Trainingsdaten mit 10000 Testdaten enthalten. Das neuronale Netz hat das Ziel, diese Bilder zu erkennen und den entsprechenden Ziffern zuzuordnen. Das Ziel ist es unter 5000 Parametern zu haben. Dabei werden 10000 Trainingsdaten verwendet und eine Genauigkeit von über 95% angestrebte.

2 Datenvorbereitung

In der Datenpräparation für das neuronale Netzwerk wurden mehrere Schritte durchgeführt, um die Eingabedaten optimal für das Training vorzubereiten. Zunächst wurden die Pixelwerte der Trainings- und Testbilder normalisiert, indem sie durch 255 geteilt wurden. Dieser Schritt bringt die Werte auf einen Bereich zwischen 0 und 1.
Anschließend erfolgte die Binarisierung der Bilder durch Anwendung einer Schwelle von 0,5. Dies bedeutet, dass alle Pixelwerte über dieser Schwelle auf 1 gesetzt wurden, während diejenigen darunter auf 0 gesetzt wurden. Diese binäre Repräsentation erleichtert die Verarbeitung und Interpretation der Daten.
Um die Trainingsdaten weiter zu optimieren, wurde ein ausgeglichener Teildatensatz ausgewählt. Dieser Schritt verringert die Anzahl der Labels auf 10000. Hierbei wurden zufällig 1000 Beispiele pro Klasse ausgewählt, um die Datenmenge zu reduzieren und gleichzeitig die Klassenverteilung beizubehalten.

3 Architektur des Netzes

Die Architektur des neuronalen Netzwerks wurde als Sequential-Modell in TensorFlow erstellt, um eine Verarbeitung von Bildern aus der MNIST-Datenbank zu ermöglichen. Die einzelnen Schichten des Modells sind im Detail beschrieben.

Das Modell fängt mit einer Conv2D-Schicht an. Das ist wie ein kleines Fenster, das über das Bild geht und nach bestimmten Mustern sucht. In diesem Fall versucht es, 8 verschiedene Muster in den Bildern der Größe 28x28 Pixel zu finden. Das Fenster, das es dabei verwendet, ist 3x3 Pixel groß.
Nachdem das Fenster verschoben wurde, wird der Wert beibehalten, wenn er positiv ist, andernfalls auf null gesetzt. Dies unterstützt das Modell dabei, Muster effektiver zu erfassen.
Nach der Conv2D-Schicht kommt eine MaxPooling2D-Schicht mit einer Pooling-Größe von (2, 2). Diese Schicht reduziert die räumliche Dimension der Daten, indem sie die Daten in kleinen 2x2-Bereichen zusammenfasst. Dabei werden die dominierenden Merkmale beibehalten, was bedeutet, dass nur die wichtigsten Informationen übernommen werden, um die Datenmenge zu verringern und die Berechnungen effizienter zu gestalten.
Die Prozedur wird erneut durchgeführt, indem eine weitere Conv2D-Schicht mit 16 Filtern hinzugefügt wird, gefolgt von einer weiteren MaxPooling2D-Schicht. Dieser Schritt trägt zur strukturierten Schichtung der extrahierten Merkmale bei, indem er die Anzahl der Filter erhöht und erneut eine räumliche Reduktion durchführt.
Die Flatten-Schicht wandelt die Daten in einen flachen Vektor um, um sie für die nachfolgende Dense-Schicht vorzubereiten. Um Überanpassung vorzubeugen, wurde eine Dropout-Schicht mit einer Auslassrate von 0.1 eingefügt.
Die abschließende Dense-Schicht mit 10 Neuronen verwendet die Softmax-Aktivierung, um Wahrscheinlichkeiten für die Ziffern 0 bis 9 zu generieren. Diese Schicht stellt die Ausgabe des Modells dar.
Die Gesamtanzahl der Parameter beträgt 5258, von denen alle trainierbar sind.

4 Training des Netzes

Das Modell wurde über einen Trainingszeitraum von 15 Epochen trainiert, wobei ein verkleinerter Satz von Trainingsdaten verwendet wurde. Die Methode "fit"wurde aufgerufen, um das Training durchzuführen, wobei die Trainingsdaten, die Anzahl der Epochen und die Batch-Größe (32) spezifiziert

wurden. Darüber hinaus wurden Validierungsdaten verwendet, um die Leistung des Modells während des Trainings zu überwachen.
Während des Trainings wurden die Verlustfunktion und die Genauigkeit aufgezeichnet, wobei jeder Durchlauf detailliert protokolliert wurde. Die Ergebnisse für jede Epoche zeigen eine allmähliche Verbesserung der Genauigkeit und eine Verringerung des Verlusts sowohl im Trainings- als auch im Validierungsdatensatz.
Die abschließende Genauigkeit auf dem Validierungsdatensatz beträgt beeindruckende 97,75%. Dies deutet darauf hin, dass das trainierte Modell in der Lage ist, handgeschriebene Ziffern mit hoher Präzision zu klassifizieren. Der Validation loss ist dennoch etwas hoch was darauf hinweisen kann dass die Lernrate zu hoch ist.

5 Leistungsbewertung

Die Loss-Funktion spielt eine entscheidende Rolle im Training eines neuronalen Netzes, indem sie den Unterschied zwischen den vorhergesagten und den tatsächlichen Werten quantifiziert. In diesem Fall wurde die Sparse Categorical Crossentropy Loss-Funktion verwendet, die speziell für Klassifizierungsprobleme mit mehreren Klassen geeignet ist. Die letzte Schicht des Modells verwendet eine Softmax-Aktivierung, um Wahrscheinlichkeiten für die verschiedenen Klassen zu generieren.
Während des Trainings wird die Loss-Funktion dazu verwendet, die Parameter des Modells zu optimieren, indem sie den Fehler zwischen den vorhergesagten und den tatsächlichen Klassen minimiert.
Die Kompilierung des Modells erfolgte mit dem Adam-Optimizer, der eine adaptive Lernrate verwendet, und der Loss-Funktion 'sparse_categorical_crossentropy'. Als Evaluationsmetrik wurde die Genauigkeit ('accuracy') gewählt. Diese Metrik misst den Anteil der korrekt klassifizierten Beispiele und dient als Indikator für die Leistung des Modells während des Trainings.
Die verwendeten Optimierungsalgorithmen und Loss-Funktionen tragen dazu bei, dass das Modell während des Trainings konvergiert und genaue Vorhersagen auf neuen Daten macht.