

# 1 Dokumentation

## Programmmentwurf Jakob Böger, Matrikelnummer: 1500285

### 1.1 Zieldefinition

Als Ziel für diesen Programmmentwurf wurde festgelegt, ein Netz mit möglichst hoher `val_accuracy` zu generieren. Dafür sollen verschiedene Methoden angewandt werden, um aus verschiedenen Herangehensweisen die besten Techniken in das zu bearbeitende Netz zu integrieren. Die anderen wichtigen Werte, Parameter und Label, wurden durch den besonderen Fokus auf die Accuracy als eher unwichtig betrachtet und sind deswegen sehr hoch.

### 1.2 Umsetzung

In diesem Kapitel sollen die getätigten Änderungen dokumentiert werden. Außerdem wird bei jeder Änderung festgehalten, welche wie sich die betrachteten Werte durch die Anpassung geändert haben.

Bei dem ursprünglichen Netz wurde eine Accuracy von 0.9787 bei 101770 Parametern und 60000 Labeln erreicht.

Durch experimentieren mit Batch Size und Zahl der Epochen konnte das Netz mit

einer Batch Size Änderung von 8 auf 128 und einer Epochen Änderung von 10 auf 20 auf eine Accuracy von 0.9819 bei gleichbleibenden Labeln und Parametern verbessert werden.

Um die Accuracy weiter zu verbessern, wurden im Anschluss weitere Layer hinzugefügt, die das Netz genauer werden lassen sollten. Zum einen wurde ein zweiter Dense-Layer mit 256 Neuronen und einer relu activation hinzugefügt und zum anderen ein Dense-Layer mit mit 10 Neuronen und einer softmax activation. Die softmax activation wurde gewählt, weil man mit dieser Aktivierungsfunktion mehrere verschiedene Klassifizierungen abdecken kann [Mar18]. Außerdem wurde der Dropout auf 0.5 erhöht um overfitting zu verhindern. Zusätzlich wurden die Werte der fit Funktion auf 30 Epochen und 256 Batch Size erhöht. So wurde eine Accuracy von 0.9856 bei 235146 Parametern und 60000 Labeln erreicht.

Um die Accuracy weiter zu verbessern wurden einige weitere Layer hinzugefügt, was die Parameterzahl und somit auch die Trainingsdauer deutlich erhöhte. Es wurden 7 2D Convolution Layer mit padding = 'same' hinzugefügt, die sich besonders gut für Bilderkennung eignen, was bei den gegebenen Daten zum Einsatz kommt [Ebe23]. Um das Training etwas zu beschleunigen und overfitting weiter zu minimieren, wurde zwischen den Layern Batch Normalization eingesetzt. Immer vor einer Aktivierungsfunktion wurde diese Batch Normalization durchgeführt [Ser15]. Eine weitere Funktion, die hinzugefügt wurde um overfitting weiter zu reduzieren, ist das early stopping. Außerdem wurde ein lr scheduler hinzugefügt. Dadurch wird die Learning Rate adaptiv an den Trainingsprozess angepasst. Mit diesen Änderungen wurde eine Accuracy von 0.9951 bei 6771562 Parametern und weiterhin 60000 Labeln.

Um die Accuracy noch etwas weiter zu verbessern wurde im Anschluss daran noch eine L2 Regularisierung zum ersten Dense Layer hinzugefügt. Diese Änderung hat, wie schon einige vorherige Änderungen, das Ziel, overfitting zu minimieren [Ros09]. Nach dieser Änderung konnte das Netz bei gleichbleibenden Parametern und Labeln auf eine Accuracy von 0.9956 verbessert werden.

Da die Anzahl der Parameter mit 6771562 sehr hoch ist und somit die Trainingsdauer

auch dementsprechend hoch ist, soll im nächsten Schritt versucht werden, die Parameter zu verringern. Jedoch soll die Zahl nur so verringert werden, dass die Accuracy nicht stark davon beeinflusst wird. Durch die Anpassung der unteren vier Convolution Layer von padding = 'same' zu padding = 'valid', konnte die Anzahl der Parameter auf 2944362 verringert werden. Bei padding = 'same' ist die Größe des Outputs genau so groß wie die Größe des Inputs. Dadurch erhält man sehr große Outputgrößen. Bei Verwenden von padding = 'valid' werden keine zusätzlichen Pixel um den Input herum hinzugefügt und dadurch wird die Output Größe kleiner als der Input. Um die Accuracy noch weiter zu verbessern wurde ein Image Data Generator zum Trainingsprozess hinzugefügt. Durch diesen werden die Trainingsbilder in Rotation, Verschiebung und Zoom leicht verändert, um ein breiteres Spektrum mit dem Trainingsprozess abzudecken. Die dadurch erreichten Ergebnisse werden im Kapitel 1.3 dargestellt.

## 1.3 Endergebnisse

Das Ziel des Projekts war es, die val\_accuracy auf einen möglichst hohen Wert anzuheben. Die Werte Label und Parameter werden dabei eher hinten angestellt. Durch Anpassen von Batch Size, Epochenzahl und Dropout sowie Einsatz von Convolutional Layern, verschiedenen Aktivierungsfunktionen, Batch Normalization, early stopping, lr scheduler, l2 Regularisierung und einem Image Data Generator konnte die val\_accuracy des Netzes auf 0.9965 erhöht werden. Das Netz hat dabei 2944362 Parameter und 60000 Label.

# Literaturverzeichnis

- [Ebe23] Dennis Gramlich Patricia Pauli Carsten W Scherer Frank Allgower Christian Ebenbauer. *Convolutional Neural Networks as 2-D systems*. 2023. URL: <https://doi.org/10.48550/arXiv.2303.03042> (besucht am 03.01.2024).
- [Mar18] Chigozie Enyinna Nwankpa Winifred Ijomah Anthony Gachagan Stephen Marshall. *Activation Functions: Comparison of Trends in Practice and Research for Deep Learning*. 2018. URL: <https://doi.org/10.48550/arXiv.1811.03378> (besucht am 03.01.2024).
- [Ros09] Corinna Cortes Mehryar Mohri Afshin Rostamizadeh. *L2 Regularization for Learning Kernels*. 2009. URL: <https://doi.org/10.48550/arXiv.1205.2653> (besucht am 03.01.2024).
- [Ser15] Christian Szegedy Sergey Ioffe. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. 2015. URL: <https://doi.org/10.48550/arXiv.1502.03167> (besucht am 03.01.2024).