

Feed Forward Neural Network Is All You Need

Digitale Bildverarbeitung und Mustererkennung

Studiengang Elektrotechnik

Studienrichtung Fahrzeugelektronik

Duale Hochschule Baden-Württemberg Ravensburg, Campus Friedrichshafen

von

Emre Ekici

Erstellungsdatum:	5. Januar 2024
Matrikelnummer:	7680655
Kurs:	TFE21-2
Gutachter der Dualen Hochschule:	Mark Schutera

1. Motivation

Der Einsatz von Machine Learning in der digitalen Bildverarbeitung umfasst eine Reihe von Methoden. Ein CNN, dessen Aufbau sich am visuellen Cortex orientiert, ist dabei häufig die erste Wahl. [Jia19, vgl. S.6 f.] Im Folgenden soll trotz dessen gezeigt werden, dass ein FFNN für die gegebene Aufgabenstellung eine sinnvolle Alternative darstellt. Daher trägt diese Arbeit den Namen "Feed Forward Neural Network Is All You Need" (angelehnt an [Vas+23]). Sie umfasst die methodische Vorgehensweise, zentrale Ergebnisse, Experimente und eine abschließende Diskussion. Dabei stehen stets Ansätze zur Verbesserung der DL-Pipeline im Mittelpunkt. Damit sollen Modellkomplexität und die Anzahl verwendeter Label reduziert werden, während die Test Accuracy gesteigert wird.

2. Zielsetzung

Um zunächst eine Baseline für die Modellleistung zu schaffen, werden fundamentale ML-Methoden auf den initialen Datensatz angewandt. Konkret sind das die logistische Regression, der k-Neares-Neighbors-Algorithmus und die Support Vector Machine. Eine andere Möglichkeit ist eine zufällige Klassifikation, welche eine deutlich niedrigere Genauigkeit erreicht. Die Ergebnisse in Tabelle 1 zeigen bereits, dass MNIST ein einfacher Datensatz ist und keine hohe Modellkomplexität erfordert.

Tabelle 1: Ergebnis fundamentaler ML-Methoden für den MNIST-Datensatz

	Logistic Regression	Support Vector Machine	K-Nearest Neighbor
Test Accuracy	92.55%	96.88%	97.92%

Dennoch muss ein Kompromiss für die drei zu verbessernden Größen gefunden werden, weshalb für die Test Accuracy lediglich die Mindestangabe von 80% erfüllt werden soll. Anhand von 10 gelabelten Bildern soll das Netz mit einem Bild pro Klasse trainiert werden. Eine weitere Verringerung würde zum Ausschluss einer Klasse im Training führen und erscheint daher nicht als sinnvoll. Es wird ein FFNN ohne verborgene Schicht angestrebt, um die Modellkomplexität minimal zu halten.

3. Vorgehen

Im ersten Schritt der Datenvorverarbeitung wird der Input in ein eindimensionales Array überführt und skaliert. Der Output wird mittels One-Hot-Encoding binär dargestellt, um das Lernen einer Rangordnung zwischen den Kategorien zu vermeiden. [HK20, vgl. S.14 f.] Im nächsten Schritt werden verschiedene Techniken der Dimensionsreduktion angewandt, um die Anzahl an Merkmalen auf zwei zu reduzieren. Dazu gehören die PCA, LDA und UMAP. [HB22] [Bur19, vgl. S.162 ff.] Durch die Darstellung der neuen Merkmalsräume wird gezeigt, dass die Klassen durch eine UMAP am deutlichsten unterschieden werden. Anhand diesem Vergleich werden die aus der UMAP resultierenden Merkmale gewählt.

Für die Auswahl der 10 Bilder für das Training kommt im nächsten Schritt der unveränderte Datensatz zum Einsatz, um keinen Informationsverlust mitzuführen. Da die Berechnung zum Vergleich der Bilder zu einem erheblichen Rechenaufwand führt, werden diese gespeichert und falls möglich lediglich geladen.

Daraufhin wird die Modellarchitektur definiert, welche aufgrund der Zielsetzung stark eingeschränkt ist. Regularisierungsmethoden wie der L1- und L2-Regularisierung, sowie Dropout kommen aufgrund der bereits sehr geringen Modellkomplexität nicht zum Einsatz. [Sri+14] Eine weitere Reduzierung der Modellparameter wäre durch die Abwandlung des Problems in eine Regression möglich. Vielversprechende Ergebnisse sind damit aufgrund der niedrigen Anzahl an Labeln kaum denkbar.

Das Training wird anhand vorheriger Erfahrungen und Empfehlungen aus der Literatur festgelegt. Als Optimierer wurde Adam gewählt, da dieser eine hohe Konvergenzgeschwindigkeit mit einer hohen Konvergenzqualität aufweist. [Aur20, vgl. S.359 ff.][KB17, vgl. S.6] Der Adam Optimierer nutzt für jedes Gewicht zwar eine individuelle Lernrate, wählt als Obergrenze jedoch den festgelegten Wert. Daher wird die Lernrate mit zu Beginn 0.005 eher hoch angesetzt, während dem Training jedoch bei Bedarf reduziert. [Smi18] Aufgrund der niedrigen Anzahl an Labeln wird mit 2000 eine besonders hohe Anzahl an Epochen gewählt. Alle Callbacks überwachen den Loss des Trainings, da keine Validierungsdaten verwendet werden. Das würde zwar Aufschluss über die Generalisierungsfähigkeit des Modells geben, jedoch eine Erhöhung der Anzahl an Labeln zur Folge haben. Außerdem erscheint die Betrachtung der Accuracy bei 10 Datenpunkten als ungeeignet.

Während dem Training werden ebenfalls keine Validierungsdaten verwendet, was grundsätz-

lich sinnvoll wäre. Obwohl in der ursprünglich gegebenen Pipeline die Testdaten zur Validierung verwendet werden, wird dieses Vorgehen grundsätzlich ausgeschlossen. Die Verfügbarkeit von Validierungsdaten würde außerdem eine Hyperparameteroptimierung ermöglichen, beispielsweise mit dem KerasTuner unter der Verwendung von Random Search. [Hut19, vgl. S.7] [Bur19, vgl. S.87] Um die Eignung der Pipeline für andere Datensätze zu evaluieren, wird dasselbe Vorgehen für Fashion-MNIST und CIFAR-10 angewandt. Dabei wird das bestehende Netz mit Daten aus Fashion-MNIST nachtrainiert, während für CIFAR-10 die gesamte Pipeline von Beginn an durchlaufen wird.

4. Ergebnisse

Die Ergebnisse in Tabelle 2 zeigen, dass die gewählte Vorgehensweise effektiv ist. Die Mindestanforderung für die Accuracy wird mit 10 Labeln erreicht. Eine gleichzeitige Verbesserung der Genauigkeit gegenüber den zu Beginn betrachteten Methoden wird nicht erzielt. Es zeigt sich, dass die Methodik für den ebenfalls einfachen Datensatz Fashion-MNIST erfolgversprechend ist. Für komplexere Bilder wie in CIFAR-10 bedarf es einer anderen Netzarchitektur, da die Modellkomplexität zu gering ist.

Tabelle 2: Ergebnis fundamentaler ML-Methoden für den MNIST-Datensatz

Datensatz	Anzahl Label	Anzahl Parameter	Test Accuracy
MNIST	10	30	88.34%
Fashion-MNIST	60 000	30	71.20%
CIFAR-10	50 000	30	22.12%

5. Fazit und Ausblick

Die Ergebnisse haben gezeigt, dass der Titel der Arbeit durchaus mit Humor zu nehmen ist. FFNN sind für die digitale Bildverarbeitung nicht alles was man braucht, vor allem bei komplexeren Datensätzen nicht. Sie reichen jedoch eben für die gegebene Aufgabenstellung in Kombination mit bestimmten Techniken zur Datenvorverarbeitung aus. Für komplexere Aufgaben im Gebiet *Computer Vision* wie beispielsweise der Objekterkennung sind CNNs nicht wegzudenken. Doch auch der gewählte Ansatz kann durch weitere Methoden verbessert werden. Mit Ensemble Learning, Monte Carlo Dropout und dem Einfügen einer Outlier-Klasse seien nur einige Möglichkeiten genannt.

Literatur

- [Aur20] Aurélien Géron. *Praxiseinstieg Machine Learning mit Scikit-Learn, Keras und TensorFlow*. dpunkt, 2020. ISBN: 978-3-96010-339-4.
- [Bur19] Andriy Burkov. *Machine Learning kompakt. Alles, was Sie wissen müssen*. 1st ed. Description based on publisher supplied metadata and other sources. Frechen: mitp, 2019. 1202 S. ISBN: 9783958459960.
- [GB10] Xavier Glorot und Yoshua Bengio. „Understanding the difficulty of training deep feedforward neural networks“. In: *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*. Hrsg. von Yee Whye Teh und Mike Titterton. Bd. 9. Proceedings of Machine Learning Research. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, S. 249–256. URL: <https://proceedings.mlr.press/v9/glorot10a.html>.
- [HB22] Chun Kit Jeffery Hou und Kamran Behdinan. „Dimensionality Reduction in Surrogate Modeling: A Review of Combined Methods“. en. In: *Data Science and Engineering* 7.4 (Dez. 2022), S. 402–427. ISSN: 2364-1541. DOI: 10.1007/s41019-022-00193-5. URL: <https://doi.org/10.1007/s41019-022-00193-5> (besucht am 04.01.2024).
- [HK20] John T. Hancock und Taghi M. Khoshgoftaar. „Survey on categorical data for neural networks“. In: *Journal of Big Data* 7.1 (Apr. 2020), S. 28. ISSN: 2196-1115. DOI: 10.1186/s40537-020-00305-w. URL: <https://doi.org/10.1186/s40537-020-00305-w> (besucht am 05.01.2024).
- [Hut19] Hutter. *Automated Machine Learning : Methods, Systems, Challenges*. Springer International Publishing, 2019. ISBN: 978-3-030-05318-5.

- [Jia19] Xiaoyue Jiang. *Deep Learning in Object Detection and Recognition*. Hrsg. von Abdenour Hadid u. a. Description based on publisher supplied metadata and other sources. Singapore: Springer Singapore Pte. Limited, 2019. 1237 S. ISBN: 9789811051524.
- [KB17] Diederik P. Kingma und Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].
- [Smi18] Leslie N. Smith. *A disciplined approach to neural network hyper-parameters: Part 1 – learning rate, batch size, momentum, and weight decay*. Techn. Ber. arXiv:1803.09820 [cs, stat] type: article. arXiv, Apr. 2018. DOI: 10.48550/arXiv.1803.09820. URL: <http://arxiv.org/abs/1803.09820> (besucht am 05.01.2024).
- [Sri+14] Nitish Srivastava u. a. „Dropout: A Simple Way to Prevent Neural Networks from Overfitting“. In: *Journal of Machine Learning Research* 15.56 (2014), S. 1929–1958. URL: <http://jmlr.org/papers/v15/srivastava14a.html>.
- [Vas+23] Ashish Vaswani u. a. *Attention Is All You Need*. Techn. Ber. arXiv:1706.03762 [cs] type: article. arXiv, Aug. 2023. DOI: 10.48550/arXiv.1706.03762. URL: <http://arxiv.org/abs/1706.03762> (besucht am 03.01.2024).