

MicroProfile

Optimizing Enterprise Java for a microservices architecture

Björn Beha und Suhay Sevinc

Zusammenfassung—This document is a model and instructions for L^AT_EX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Formulas or Math in Paper Title or Abstract.

Der Hype um den Modularisierungsansatz Microservices dauert an.

II. TRAININGSVORGANG

A. Trainingsvorgang

Der Trainingsvorgang basiert auf ein vollständig verbundenes mehrschichtiges Deep Learning-Netzwerk mit einem Feed-Forward-Prinzip. Hier existieren die Schichten:

Input-Schicht

Hidden-Schicht

Output-Schicht

Die Input-Schicht stellt dabei die Eingangsdaten dar, welche als Trainingsmaterial für das Training gelten. Bei den Eingangsdaten handelt es sich um die vorverarbeiteten Sprachaufnahmen. In der Hidden-Layer-Schicht geschieht das eigentliche Training, welches normalerweise durch die Sigmund-Freud-Funktion aktiviert wird [?].

$$\text{sigm}(x) := \frac{1}{1 + e^{-x}}$$

Die Ableitung dieser Funktion gibt Hinweise, die wiederum Nachteile sich bringt.

$$\text{sigm}(x)' := \frac{e^x}{(e^x + 1)^2}$$

Beim Backpropagation wird die Ableitung benötigt. Das ist ein Verfahren, welches zur Fehlerrückführung dient. Es wird als Verallgemeinerung der Delta-Regeln auf mehrschichtige neuronale Netze angewandt. Hier entsteht ein Lernverlust bis zu 25 Prozent. Da bei der Sprachverarbeitung viel Daten anfallen, würde der Trainingsverlust zu groß sein. Somit wäre das Lernen nicht mehr effizient, welches in 1 dargestellt ist [1].

Anstelle der Sigmund-Freud-Funktion wird in den modernsten Deep-Learning-Netzen Rectified linear Units (ReLU) verwendet. Dabei gibt die ReLU-Funktion bei einem negativen Input-Wert, den Wert null zurück. Bei einem positiven Input-Wert wird dieser als Output-Wert übernommen. Diese Funktion ist dem menschlichen Neuronen ähnlicher und deutlich effizienter [2]. Hinzukommt, dass in die Formel noch die Kantengewichtungen der Neuronen hinzukommen müssen [3].

$$y_j = \text{ReLU}(x_j) = \max(0, x_j)$$

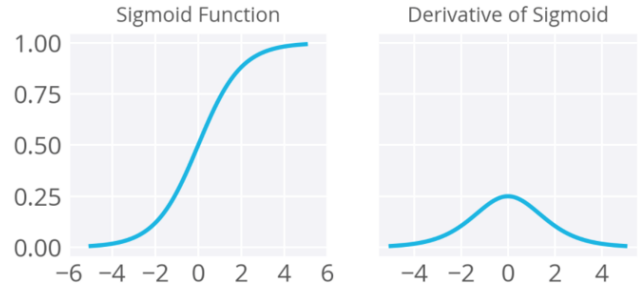


Abbildung 1. Darstellung der Sigmund-Freud-Funktion und dessen Ableitung [1]

$$x_j = b_j + \sum x_{ij} * y_j \quad (1)$$

Dabei werden die Berechnung der Kantengewichte x_j und Aktivierungsfunktion x_j über den Index j gemapped. Die Forschung zeigt, dass ReLUs zu einem viel schnelleren Trainingsvorgang für große Netzwerke führen. Die meisten Frameworks wie TensorFlow und TFLearn machen es einfach, ReLUs auf Hidden-Schicht anzuwenden, sodass diese Implementierungen bereits zur Verfügung gestellt werden. Als nächstes folgt die Output-Schicht, welches die Eingangsdaten dann zu den Klassen zuordnet. Diese Schicht ist als Softlayer konfiguriert, welches die Klassen in eine eindimensionale Matrix kategorisiert. Die Klasse steht für die Sprache, die gelernt werden. Da mehrere Sprachen trainiert werden, werden automatisch die gleiche Anzahl an Klassen benötigt. Die endgültige Sprachidentifikation geschieht über Wahrscheinlichkeitsverteilung, welches in Abbildung 2 dargestellt wird [1]. Dies geschieht



Abbildung 2. Klassenzuordnung über Wahrscheinlichkeiten in der Softmax-Konfiguration [1]

durch die Funktion $p(j)$. Dabei steht der Index l für die Klasse, also die Sprache, die gelernt wird.

$$p(j) := \frac{\exp(x_j)}{\sum_l \exp(x_l)}$$

Für die Backpropagation wird ebenfalls eine cost function benötigt, welche für Korrektur der Kantengewichte ist. Diese geschieht durch Cross-Entropy-Verlust-Funktion.

$$C := \sum_l t_j * \log(p_j)$$

Dabei misst die Funktion die Leistung des Klassifikationsmodells, dessen Ausgabe. Es handelt sich um ein Wahrscheinlichkeitswert zwischen 0 und 1. Der Cross-Entropy-Verlust nimmt zu, wenn die vorhergesagte Wahrscheinlichkeit von der tatsächlichen Beschriftung abweicht [4]. Bei t_j handelt es sich um die Zielklasse, für die der Verlust berechnet wird [3].

B. Netztopologie

Die Netztopologie beschreibt die Infrastruktur des Netzes. Die Auswahl der Topologie bestimmt die Qualität des Trainingsvorgangs. Aufgrund dessen fallen Topologien von Ansatz zu Ansatz unterschiedlich aus, welche auch unterschiedliche Spracherkennungsergebnisse liefern. Dafür wird die Topologievorschlag von Gonzales et al. betrachtet. Für die Eingangsdaten werden 40 Filterbanken verwendet. Um die Menge der Trainingsdaten zu erhöhen, Diese werden benötigt, um die Daten sampilen zu können. Dadurch entsteht eine Menge von Input-Schicht von 26 Knoten. Um unerwünschte Latenzzeiten der Frames zu vermeiden wird ein asymmetrischer Kontext verwendet. Die Hidden-Schichte beträgt hier vier Ebenen mit einer Gesamtzahl von 2560 Units. Die Output-Schicht enthält wie bereits erwähnt eine Softmax-Konfiguration, welche eine Dimension der Anzahl der Zielsprachen besitzt.

C. Verbesserung des Trainingsverfahrens durch Multitasking learning (MTL)

Bei maschinellem Lernen wird besonders wert gelegt, bestimmte Metriken, wie beispielsweise Klassifizierungs-Genauigkeit und Trainingsdauer, zu optimieren. Daraufhin wird das Modell soweit optimiert bis die Leistung des Modells nicht mehr gesteigert werden kann [5]. Das Lernen der einzelnen Sprachen läuft sequenziell ab. Hier setzt das multitasking learning ein. Hier werden mehrere Lernaufgaben gleichzeitig erledigt statt sequentiell, um das Trainingsverfahren effizienter zu gestalten. Das führt zu einer verbesserten Lerneffizienz und Vorhersagegenauigkeit. Im Klassifizierungskontext zielt MTL darauf ab, die Leistung mehrerer Klassifizierungsaufgaben zu verbessern, indem sie gemeinsam erlernt werden [6]. Ein Beispiel hierfür ist ein Spamfilter. Der Schlüssel zur erfolgreichen Anwendung von MTL besteht darin, dass die Aufgaben miteinander verknüpft werden. Dies bedeutet nicht, dass die Aufgaben ähnlich sein müssen. Stattdessen bedeutet es, dass Aufgaben auf verschiedene Ebenen abstrahiert und geteilt werden. Wenn die Lernaufgaben tatsächlich ähnlich sind, können sie gemeinsam zu gelernt werden. Dabei kann das Wissen zwischen Aufgaben auf andere Lernaufgaben übertragen werden, welches die Effizienz des Trainingsgeschwindigkeits deutlich erhöht. Multitask Learning ist vor allem dann nützlich, wenn die Größe des Trainingssatzes im Vergleich

zur Modellgröße klein ist. Dabei wird grundsätzlich zwei Arten von MTL unterschieden: Hard parameter sharing und soft parameter sharing.

Hard parameter sharing stellt das meist genutzte Art dar [5]. Es wird normalerweise auf die hidden-Schicht angewendet, indem die Aufgaben gemeinsam gelernt werden, während die spezifischen Aufgaben separat gelernt werden. Dieser Ansatz reduziert das Risiko von Overfitting erheblich. Je mehr Aufgaben gleichzeitig gelernt wird, desto mehr muss das Modell eine Repräsentation finden, die alle Aufgaben erfassen muss. Dadurch ist die Chance geringer, das Netz zu overfitten.

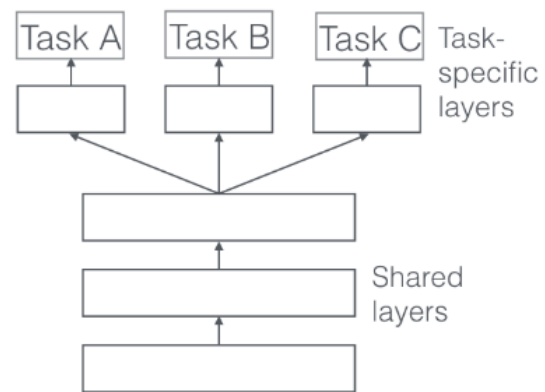


Abbildung 3. Darstellung der Sigmoid-Freud-Funktion und dessen Ableitung [1]

III. FAZIT

IV. AUSBLICK

LITERATUR

- [1] GitHub, "Kulbear/deep-learning-nano-foundation," 2017, (Accessed on 2018-04-15). [Online]. Available: <https://github.com/Kulbear/deep-learning-nano-foundation/wiki/ReLU-and-Softmax-Activation-Functions>
- [2] M. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, and G. Hinton, "On rectified linear units for speech processing," in *38th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vancouver, 2013.
- [3] J. Gonzalez-Dominguez, D. Eustis, I. Lopez-Moreno, A. Senior, F. Beaufays, and P. J. Moreno, "A real-time end-to-end multilingual speech recognition architecture," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 749–759, 2015.
- [4] M. Cheatsheet, "Loss functions — ml cheatsheet documentation," http://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html, 2017, (Accessed on 2018-06-03).
- [5] S. Ruder, "An overview of multi-task learning in deep neural networks," 2017. [Online]. Available: <http://ruder.io/multi-task/>
- [6] Y. Lu, F. Lu, S. Sehgal, S. Gupta, J. Du, C. H. Tham, P. Green, and V. Wan, "Multitask learning in connectionist speech recognition," 2015.