

Орлова А.Р.

*Научный руководитель: старший преподаватель М.В. Залугина  
Муромский институт (филиал) федерального государственного образовательного  
учреждения высшего образования «Владимирский государственный университет  
имени Александра Григорьевича и Николая Григорьевича Столетовых»  
602264, г. Муром, Владимирская обл., ул. Орловская, 23  
e-mail: anas.orlova2015@yandex.ru*

### **Кünstliche Neuronale Netzen**

Bei Künstlichen Neuronalen Netzen (KNNs) handelt sich um ein parallelisierbares Rechenmodell mit vielen sehr einfachen Recheneinheiten, die Neuronen genannt werden. Das Modell ist durch biologische Nervensysteme inspiriert

KNNs kann man sowohl in diskreten als auch kontinuierlichen Zeitschritten verwenden. Meist wird die diskrete Variante genutzt, worauf sich hier beschränkt wird. Weitere Unterscheidungen betreffen die Topologie der Verknüpfungen und orientieren sich an [1].

Vollverknüpfte Netze funktionieren sehr einfach hin zum ähnlichsten bekannten Muster. Der Rahmen dieses Papers reicht leider nicht aus, diesen Netztyp vorzustellen. Diesen Netztyp ist hier dennoch erwähnt, weil er einfach verständlich ist, und dennoch ein sehr wichtiges (und ggf. ungewohntes) Konzept illustriert: das assoziative, nicht-lokalisierte Gedächtnis.

Feed-Forward-Netze ist Netze, in deren Verknüpfungen Zyklen vorkommen, nennt man rekurrent. Mit ihnen lässt sich eine Art Arbeitsgedächtnis realisieren. Nicht-rekurrente Netze heißen hingegen Feed-Forward- Netze. Ein wichtiger Vertreter hierfür sind die sog. Perceptrons.

Convolutional Neural Networks (CNN) ist eine wichtige und sehr erfolgreiche Erweiterung der Perceptrons sind die sog. Convolutional Neural Networks (CNN) [1].

Beim Lernen eines KNN geht es in den allermeisten Fällen um die geeignete Anpassung der Verknüpfungsgewichte. Grundsätzlich sind drei verschiedene Lernformen zu unterscheiden: überwachtes, Unüberwachtes und Bestärkendes Lernen (engl. Supervised, Unsupervised bzw. Reinforcement Learning).

Back Propagation. Für die eingangs genannten Lern-Kategorien existieren verschiedene Trainingsverfahren. Vorgestellt werden soll hier die Methode, die sich v.a. beim überwachten Lernen als äußerst erfolgreich herausgestellt hat: Back Propagation mittels Gradient Descent:

$$D_k(y_1, y_2) = (y_1 - a_1)^2 + (y_2 - a_2)^2 \text{ [2].}$$

Anwendung im mobile systems engineering. In werden mit Hilfe eines Convolutional Neural Networks (CNN) Hindernisse für ein bewegtes Fahrzeug erkannt. Ferner wird auf ähnliche Weise auf denselben Bildern die befahrbare Straße segmentiert.

Der Lernvorgang von StixelNet ist stark von entlehnt. Als Lernmethode wird Stochastic Gradient Descent mit  $|P| = 128$  Bildern/ Epoche verwendet. Initiale Lernrate ist  $\eta_0 = 0.01$ . Alle 10000

Epochen wird die Lernrate halbiert, um das Minimum nicht zu verfehlen. Ebenfalls kommt ein sog. Momentum Term zum Einsatz: Trotz Null-Gradienten erleichtert der "Schwung" früherer Gradienten, Plateaus und schlechte lokale Minima zu überwinden [3].

Vor- und Nachteile der KNNs im Vergleich zu handgefertigten math. Modellen scheinen sich zurzeit in etwa die Waage zu halten. Ihre Verwendung scheint weniger eine Frage der objektiven Nützlichkeit als eine der persönlichen Präferenzen, Erfahrung sowie der Bereitschaft zu sein, einige Zeit in Wahl und Justierung der sog. Hyperparameter (z.B. Anzahl und Größe von Faltungsmasken je Ebene, Anzahl an Schichten und Neuronen pro Schicht, Lernrate etc.) zu investieren [4].

### **Литература**

1. David Kriesel. Ein kleiner Überblick über Neuronale Netze. 2007.
- 2.Y. Le Cun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, Nov 1998.

3. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012.

4. [https://www.researchgate.net/publication/312114357\\_Neuronale\\_Netze](https://www.researchgate.net/publication/312114357_Neuronale_Netze)