Technology
Arts Sciences
TH Köln

Technische Hochschule Köln
Fakultät für Informatik und Ingenieurwissenschaften

P R A K T I K U M

Neuronale Netze

Gruppe M.16

Vorgelegt an der TH Köln
Campus Gummersbach
Mathematik 2

ausgearbeitet von:
Maximilian Luca Ramacher
Marius Kühnast Muratcan Garanli
Kai Murza Nick Struckmeyer

Erster Betreuer: Marc Oedingen
Zweiter Betreuer: Peter Wagner

Gummersbach, im <Monat der Abgabe>

Zusammenfassung

Platz für das deutsche Abstract...

Abstract

Platz für das englische Abstract...

Inhaltsverzeichnis

Abbildungsverzeichnis

1 Einleitung Neuronale Netze

NN (wofür, Einsatzgebiete, Grobes Prinzip). EINFÜGEN

Inhalt

Die Einleitung umfasst folgende Elemente[a]:

- Wofür benötigt man Neuronale Netze
- Einsatzgebiete
- Grobes Prinzip

> Eine Einleitung muss auch durch die Arbeit führen. Sie muss dem Leser helfen, sich in der Arbeit und ihrer Struktur zu Recht zu finden. Für jedes Kapitel sollte eine ganz kurze Inhaltsangabe gemacht werden und ggf. motiviert werden, warum es geschrieben wurde. Oft denkt sich ein Autor etwas bei der Struktur seiner Arbeit, auch solche Beweggründe sind dem Leser zu erklären[b]:.

[a] Vgl. u.a. [BBoJ], S. 5-6
[b] [BBoJ], S. 6

1.1 Wofür benötigt man Neuronale Netze

1.2 Einsatzgebiete

1.3 Grobes Prinzip

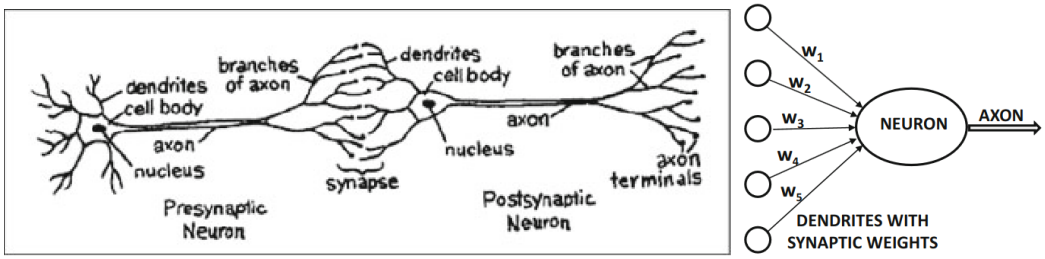
2 Neuronen

Inhalt

- Was sind Neuronen
- Arten von Neuronen
- Funktionsweise
- Aktivierungsfunktion
- Schichtenmodell

2.1 Was sind Neuronen?

Das menschliche Nervensystem besteht aus Neuronen, welche mit Axonen oder Dendriten verknüpft sind. Diese Verbindungen werden auch Synapsen genannt. Die variable Stärke der Synapsen ermöglichen das Lernen. Dieser biologische Mechanismus wird durch neurale Netze simuliert.



(a) Synapse (b) Neurales Netzwerk

Abbildung 1: Bild aus dem Buch 'Neural Networks and Deep Learning' von Charu C. Aggarwal

Ein neuronales Netz besteht aus mindestens einem Neuron. Neuronen sind essentielle Bestandteile von neuralen Netzen. Sie nehmen Eingabedaten entgegen und wandeln diese in Ausgabedaten um. Neben den Eingabedaten werden auch Weight-Parameter übergeben, welche die zu berechnenden Werte beinflussen. Das eigenliche „Lernen“ erfolgt durch diesen Einfluss.

2.2 Arten von Neuronen

2.2.1 Input Neuronen

todo

2.2.1.1 Bias Neuronen

2.2.2 Output Neuronen

todo

2.2.3 Versteckte Neuronen

todo

2.3 Funktionsweise

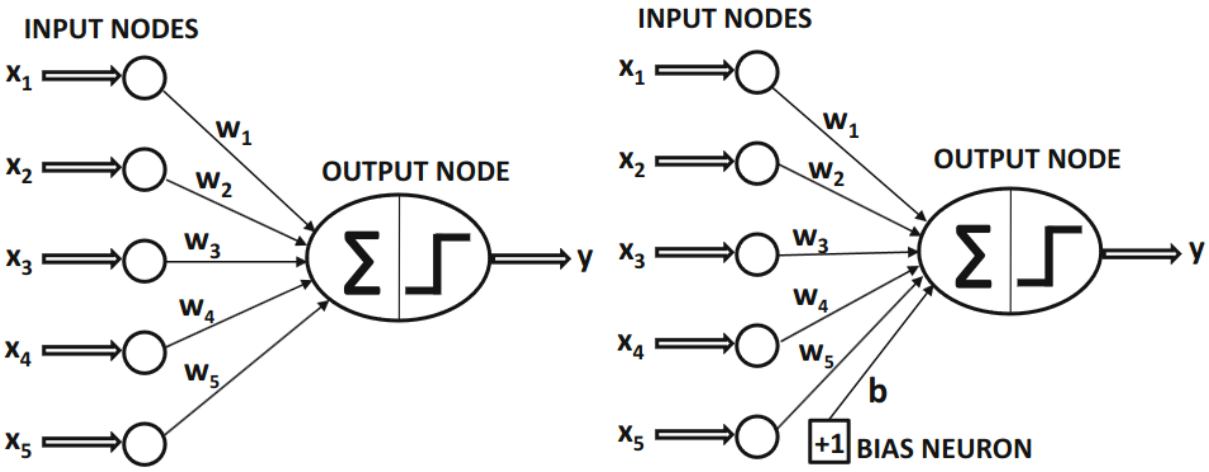
2.3.1 Perceptrons

Die simpelste Form eines Neuralen Netzwerks ist ein Perceptron. Es kann nur binäre Entscheidungen {-1,+1} treffen. Ein Perceptron besteht aus einer Input Layer und einem Output Node. Die Input Layer beinhaltet d nodes welches d Merkmale $\overline{X} = [x_1...x_d]$ mit Weights $\overline{W} = [w_1...w_d]$ übermittelt. Die lineare Aktivierungsfunktion sign berechnet dann die Vorhersage $\hat{y}$.

$$\hat{y} = sign\{\overline{W} \cdot \overline{X}\} = sign\{\sum_{j=1}^{d} w_j x_j\}$$

In vielen Fällen wird ein nicht variables Element b mit in der Rechnung berücksichtigt. Dies verursacht das der Mittelwert der Vorhersage nicht 0 ist.

$$\hat{y} = sign\{\overline{W} \cdot \overline{X} + b\} = sign\{\sum_{j=1}^{d} w_j x_j + b\}$$



(a) Perceptron ohne Bias (b) Perceptron mit Bias

Abbildung 2: Aufbau von Perceptronen, Bild aus dem Buch 'Neural Networks and Deep Learning' von Charu C. Aggarwal

Durch der so genannten Minimierung lässt sich der Fehler der Vorhersage verringern. Hierzu werden neben den Features x, auch Labels y in einem Feature-Label Paar eingeführt.

$$\text{Minimize}_{\overline{W}} L = \sum_{(\overline{X},y)\in\mathcal{D}} (y - \hat{y})^2 = \sum_{(\overline{X},y)\in\mathcal{D}} (y - sign\{\overline{W} \cdot \overline{X}\})^2$$

Diese Art der Minimierung wird auch als Loss-Funktion bezeichnet. Die obige Funktion führt zu einer treppenstufigen Loss-Ebene, welche für gradient-descent ungeeignet ist. Um diesem Problem entgegen zu wirken, wird eine Smooth-Funktion angewendet.

$$\Delta L_{\text{smooth}} = \sum_{(\overline{X},y)\in\mathcal{D}} (y - \hat{y})\overline{X}$$

Beim Trainieren eines Neuralen Netzwerks werden Eingabedaten $\overline{X}$ einzelt oder in kleinen Batches eingespeist um die Vorhersage $\hat{y}$ zu generieren. Die Weights werden dann durch den Fehlerwert $E(\overline{X}) = (y - \hat{y})$ aktualisiert.

$$\overline{W} \Rightarrow \overline{W} + \alpha(y - \hat{y})\overline{X}$$

Der Parameter α reguliert die Lernrate des Neuralen Netzwerks. Der Perceptron-Algorithmus durchläuft die Trainingsdaten mehrmals bis die Weight-Werte konvergieren. Ein solcher Durchlauf wird als Epoche bezeichnet.
Der Perceptron-Algorithmus kann auch als stochastische Gradientenabstiegsmethode betrachtet werden. TODO Erklärung hier?

2.3.2 Aktivierungsfunktion

Aktivierungsfunktionen ermöglichen den Neuronen nicht-lineare Outputs zu produzieren. Außerdem wird durch diese Funktionen entschieden, welche Neuronen aktiviert werden und wie die Inputs gewichtet werden. Zur Notation von Aktivierungsfunktion nutzen wir Φ.

$$\hat{y} = \Phi(\overline{W} \cdot \overline{X})$$

2.3.2.1 Lineare Aktivierung

Die simpelste Aktivierungsfunktion $\Phi(\cdot)$ ist die lineare Aktivierung. Sie bietet keine nicht linearität. Sie wird oft in Output Nodes verwendet, wenn das Ziel ein reeler Wert ist.

$$\Phi(v) = v$$

2.3.2.2 Nicht lineare Aktivierung

In den frühen Tagen der Entwicklung von neuralen Netzen wurden sign, sigmoid und hyperbolic tangent Funktionen genutzt.

2.3.2.2.1 Sign Aktivierung

Die sign Funktion generiert nur binäre {-1,+1} Ausgaben. Aufgrund der Nichtstätigkeit der Funktion, können beim Trainieren keine Loss-Funktionen verwendet werden.

$$\Phi(v) = \text{sign}(v)$$

2.3.2.2.2 Sigmoid Aktivierung

Die Sigmoid Funktion generiert Werte zwischen 0 und 1. Sie eignet sich deshalb für Rechnungen die als Wahrscheinlichkeiten interpetiert werden sollen.

$$\Phi(v) = \frac{1}{1 + e^{-v}}$$

2.3.2.2.3 Tanh Aktivierung

Der Graph der Tanh Funktion hat eine ähneliche Form wie die der Sigmoid Funktion. Sie unterscheidet sich jedoch in der Skalierung, denn ihre Wertebereich liegt zwischen -1 und 1.

$$\Phi(v) = \frac{e^{2v} - 1}{e^{2v} + 1}$$

Die Tanh Funktion lässt sich auch durch die Sigmoid Funktion darstellen.

$$\tanh(v) = 2 \cdot \text{sigmoid}(2v) - 1$$

2.3.2.3 Piecewise lineare Aktivierung

2.3.2.3.1 ReLU

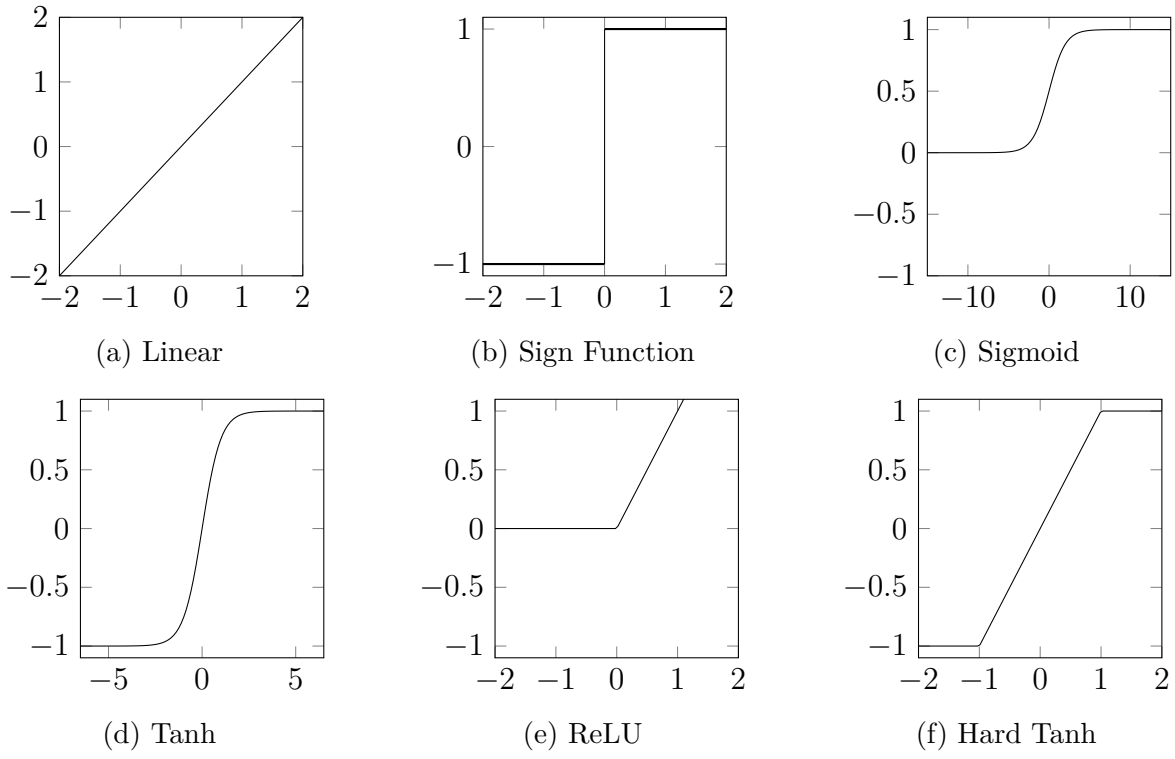
TODO

$$\Phi(v) = \max\{v, 0\}$$

2.3.2.3.2 hard tanh Aktivierung

TODO

$$\Phi(v) = \max\{\min[v, 1], -1\}$$



(a) Linear (b) Sign Function (c) Sigmoid

(d) Tanh (e) ReLU (f) Hard Tanh

Abbildung 3: Aktivierungsfunktionen

2.3.3 Schichtenmodell

TEXT FOLGT...

2.3.4 Loss-Function

2.4 Wie sind Neuronen miteinander verknüpft

Hier sollen die Weights erklärt werden

2.4.1 Weights

TEXT FOLGT...

2.5 Fehler / Backpropagation Einführung

Nur eine sehr knappe Einführung, da eigenes Kapitel für dieses Thema reserviert

3 Gradientenverfahren

Inhalte des *Gradientenverfahren*

- Wofür braucht mann das Gradientenverfahren?
- Grundkonzepte des Gradientenabstiegsverfahren
- Gefährliche Fehlerquellen

Das Kapitel Gradientenverfahren stellt die Grundlagen dar, die für das Verständnis des Lernprozesses eines neuronalen Netzwerks im nachfolgenden Kapitel erforderlich sind.

3.1 Wofür braucht mann das Gradientenverfahren?

Das Gradientenverfahren wird genutzt, um das Minimum der Verlustfunktion zu finden. Dort ist der optimale Trainingszustand des Modells gefunden, weil dort der Fehler minimal ist.

3.1.1 Was ist ein Gradient?

Ein Gradient ist ein Spaltenvektor der ersten partiellen Ableitung einer Funktion mit "n" Variablen. Dieser beschreibt die Richtung und Stärke der größten Steigung in einem Punkt p in Form eines Vektors.[JH20]

3.2 Grundkonzepte des Gradientenverfahrens

3.2.1 Grundlage für den Fehlerrückführungs-Algorithmus - Wozu dient er?

Warum soll das Minimum der Verlustfunktion gefunden werden? Das spätere Lernen geschieht durch Anpassung der Gewichte, es wird die Differenz aus der tatsächlichen und der korrekten Ausgabe bestimmt. Auf die Fehlerbestimmung wird in Kapitel 4 - Backpropagation eingegangen. Dafür ist es notwendig, eine Möglichkeit zu finden, wie der Fehler verringert werden kann.

3.2.2 Wie funktioniert das Gradientenverfahren?

Bei der Delta Regel(bzw. allgemein beim Gradientenabstiegsverfahren) vergleicht man gewünscht und berechnete (Ausgabe-) Werte und nimmt mit Hilfe dieses Delta-Terms sukzessive Gewichtsveränderungen vor. Die Gewichte werden mit dem Gradientenabstiegsverfahren (bzw. im speziellen der Delta-Regel) modifiziert. Ziel ist ein Minimum der Fehlerfunktion per Näherungsverfahren zu finden.

Das Verfahren durchläuft folgende Schritte:

- Wahl eines (zufälligen) Startpunktes
- Festsetzung eines Lernparameters
- Festlegung des Abbruchkriteriums
  - Fixierung der kritischen Differrenz der Gewichtsveränderungen, die nicht unterschritten werden darf
  - Spezifizierung der maximalen Anzahl an Iterationen (Wiederholungen), die vorgenommen werden sollen.
- Berechnung des Gradienten
- Veränderung der Gewichte

Der vierte und der fünfte Punkt werden solange wiederholt, bis mindestens eines der beiden Abbruchkriterien erfüllt ist (siehe dritter Punkt). Das Gradientenverfahren beginnt mit einer zufälligen Gewichtskombination, die die Startposition auf der Kurve bzw. in einer n-dimensionalen "Gebirgslandschaft"makiert. Von dieser Position aus soll nun das "tiefste Talïn der "Hügellandschaft"gesucht werden.[GR10]

TODO: **Graphische Veranschaulichung, in 3-dim Raum **

3.2.3 Mehrere Dimensionen

Das Gradientenverfahren beginnt mit einer zufälligen Gewichtskombination, die die Startposition auf der Kurve bzw. in einer n-dimensionalen "Gebirgslandschaft"makiert. Von dieser Position aus soll nun das "tiefste Talïn der "Hügellandschaft"gesucht werden. Im zweidimensionmalen Raum kann ein Abstieg notwendigerweise nur nach links oder rechts erfolgen, während man sich im dreidimensionalen Raum einmal um seine eigene Achse drehen muss, um den steilsten Abstieg bestimmen zu können.

Mathematisch ist der steilste Abstieg druch den sogenannten Gradienten(daher der Name Gradientenverfahren) repräsentiert bzw. genauer gesagt durch den negativen Gradienten, da der Gradient selbst den stärksten Anstieg in der "Hügellandschaft"makiert. Der Gradient gibt nicht nur die Richtung, sondern zugleich auch die Steigung des "Hügelsän und stell folglich einen n-1-dimensionalen Vektor dar.[GR10]

3.3 Gefährliche Fehlerquellen

3.3.1 Befindet man sich wirklich im globalen Minimum?

Gradientenabstiegs- und Suchverfahren finden in der Regel nur lokale Minima, abhängig vom gewählten Startpunkt. Durch die fehlende Kenntnis der gesamten n-dimensionalen

"Hügellandschaft", die sich hinter einem "Nebelschleier"verbirgt, ist de facto nie, ausgenommen der gesamte Fehlerterm liegt bei Null (In diesem Fall ist gewährleistet, das es sich um ein globales Minimum handelt) sichergestellt, dass das Verfahren das "tiefte Tald.h. das globale Minimum - findet.

Eine Fehlerquelle ist, dass in der Praxis oft lokale Minima auftreten, da die Fehlerfunktion oft nicht konvex ist, wodurch das Erreichen eines globalen Minimums verhindert wird.

3.3.2 Steckt man in einem lokalen Minimum fest?

Neuronale Netze können in einem lokalen Minimum feststecken, insbesondere bei der Verwendung von nicht-konvexen Kostenfunktionen. Ein lokales Minimum tritt auf, wenn das Netzwerk in einem Punkt des Fehlergradienten auf eine niedrigere Fehlerfunktionsebene trifft, aber in der Nähe dieses Punktes gibt es einen anderen Punkt mit noch niedrigerem Fehler. Da Neuronale Netze häufig große Anzahlen von Parametern haben, kann die Suche nach dem globalen Minimum eine schwierige Aufgabe sein.[HS97]

3.3.3 Wie löst man dieses Problem?

TEXT FOLGT...

3.3.4 Wie sollen Initialisierungs-Werte der Gewichte gewählt werden?

Wenn alle Gewichte mit 0 gewählt werden, ist ein Lernen nicht möglich, also TODO....

4 Backpropagation

Inhalt

Die Backpropagation umfasst folgende Elemente:

- Wie lernen Neuronale Netze?
- Grundidee Backpropagation
- Fehlerfunktion finden
- Gewichtsanpassung

> Eine Einleitung muss auch durch die Arbeit führen. Sie muss dem Leser helfen, sich in der Arbeit und ihrer Struktur zu Recht zu finden. Für jedes Kapitel sollte eine ganz kurze Inhaltsangabe gemacht werden und ggf. motiviert werden, warum es geschrieben wurde. Oft denkt sich ein Autor etwas bei der Struktur seiner Arbeit, auch solche Beweggründe sind dem Leser zu erklären[a]:.

[a] [BBoJ], S. 6

4.1 Wie lernen Neuronale Netze?

4.2 Grundidee Backpropagation

4.2.1 Fehlerrückführung

TEXT FOLGT...

4.2.2 Methode mit Matrizenmultiplikation

TEXT FOLGT...

4.2.3 Forward / Backward Phase erklären

TEXT FOLGT...

4.3 Fehlerfunktion finden

4.4 Gewichtsanpassung (Maybe)

5 Trainieren und Testen von Neuralen Netzen

todo

5.1 Trainingsdaten

todo

5.2 Testdaten

todo

6 Quellenverzeichnis

6.1 Literatur

[SW11] Stickel-Wolf, Christine; Wolf, Joachim (2011): Wissenschaftliches Lernen und Lerntechniken. Erfolgreich studieren—gewusst wie!. Wiesbaden: Gabler.

[CA18] Aggarwal, Charu C. (2018): Neural Networks and Deep Learning: A Textbook. Springer.

[TR17] Rashid, Tariq (2017): Neuronale Netze selbst programmieren. In O'Reilly eBooks. NY, USA

6.2 Internetquellen

[BBoJ] Bertelsmeier, Birgit (o. J.): Tipps zum Schreiben einer Abschlussarbeit. Fachhochschule Köln-Campus Gummersbach, Institut für Informatik. http://lwibs01.gm.fh-koeln.de/blogs/bertelsmeier/files/2008/05/abschlussarbeitsbetreuung.pdf (29.10.2013).

[HR08] Halfmann, Marion; Rühmann, Hans (2008): Merkblatt zur Anfertigung von Projekt-, Bachelor-, Master- und Diplomarbeiten der Fakultät 10. Fachhochschule Köln-Campus Gummersbach.http://www.f10.fh-koeln.de/imperia/md/content/pdfs/studium/tipps/anleitungda270108.pdf (29.10.2013).

[JH20] Harrer, J. (2020): Künstliche Neuronale Netze.https://pxldeveloper.eu/assets/docs/KuenstlicheNeuronaleNetzeJulianHarrer.pdf (20.04.2023)

[GR10] Günter Daniel Rey und Karl F Wender(2010): Neuronale Netze, eine Einführung in die Grundlagen, Amwemdung und Datenauswertung

[HS97] Hochreiter, S. and Schmidhuber, J. (1997): Flat minima. Neural Computation, 9(1), 1-42.

A Anhang

A.1 Unterabschnitt von Anhang

TEXT FOLGT...

Erklärung über die selbständige Abfassung der Arbeit

Ich versichere, die von mir vorgelegte Arbeit selbständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht.
Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

_____

(Ort, Datum, Unterschrift)

Hinweise zur obigen *Erklärung*

- Bitte verwenden Sie nur die Erklärung, die Ihnen Ihr **Prüfungsservice** vorgibt. Ansonsten könnte es passieren, dass Ihre Abschlussarbeit nicht angenommen wird. Fragen Sie im Zweifelsfalle bei Ihrem Prüfungsservice nach.
- Sie müssen **alle abzugebende Exemplare** Ihrer Abschlussarbeit unterzeichnen. Sonst wird die Abschlussarbeit nicht akzeptiert.
- Ein **Verstoß** gegen die unterzeichnete *Erklärung* kann u. a. die Aberkennung Ihres akademischen Titels zur Folge haben.