

Neuronale Netzwerke in der Computerlinguistik

Eine Einführung

Dennis Wegerhoff

13. Juni 2024

Institut für Linguistik (BUW)

Einführung

Biologische neuronale Netze

Künstliche Neuronale Netze

Anwendung und Kernproblematik in der Computerlinguistik

Einführung

- Künstliche neuronale Netze sind dem menschlichen neuronalen System nachempfunden. Sie sind ein grundlegendes Konzept im maschinellen Lernen (Machine Learning)
- Maschinelles Lernen $\approx$ Vorgang, bei dem ein Algorithmus sich durch Feedbackschleifen bei der Datenauswertung selbst optimiert
- Computerlinguistik $\approx$ Interdisziplinäre Wissenschaft, die sich mit der **maschinellen Verarbeitung natürlicher Sprachen** auseinandersetzt.

Biologische neuronale Netze

- Neuronales Netz = beliebige Anzahl miteinander verknüpfter Neuronen

Neuron

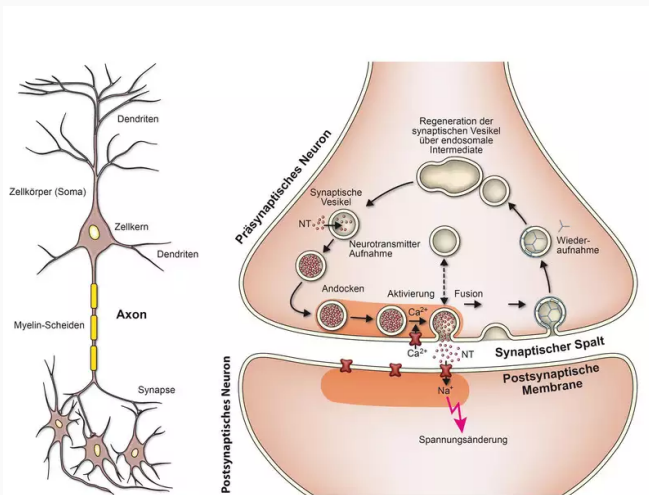


Abbildung 1: Aufbau einer Nervenzelle (MPI 2016)

Aktionspotential

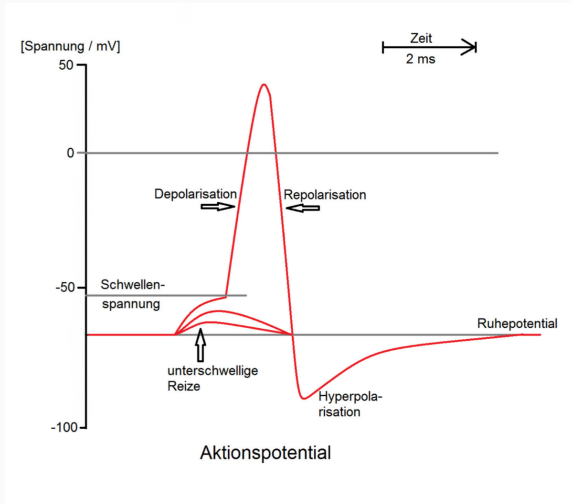


Abbildung 2: Aktionspotential

Hebbsche Regel: Kommt es zwischen zwei Neuronen A und B wiederholt und dauerhaft zur Erzeugung von Aktionspotentialen, so steigt die Effizienz, mit der Neuron A bei Neuron B ein Aktionspotential erzeugt.

Künstliche Neuronale Netze

- Je nach Struktur und Aufbau gibt es verschiedene künstliche neuronale Arten von Netzwerken (Perceptron, Multi-Layer Perceptron, rekurrente Netze...)
- Alle haben gemein: Sie berechnen Funktionen aus Input- und Outputdaten

Aufbau eines neuronalen Netzwerks

- **Eingabeschicht:** Eingabedaten aufnehmen.
- **Versteckte Schichten:** Komplexere Berechnungen zur Mustererkennung
- **Ausgabeschicht:** Vorhersagen oder Klassifikationen basierend auf den Eingabedaten.

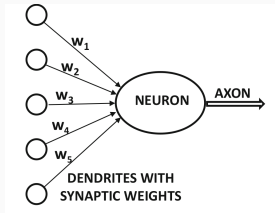


Abbildung 3: Schematisches künstliches Neuronales Netzwerk (Aggarwal 2023, S. 2)

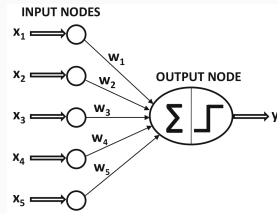


Abbildung 4: Ein einschichtiges Perceptron (Aggarwal 2023, S. 6)

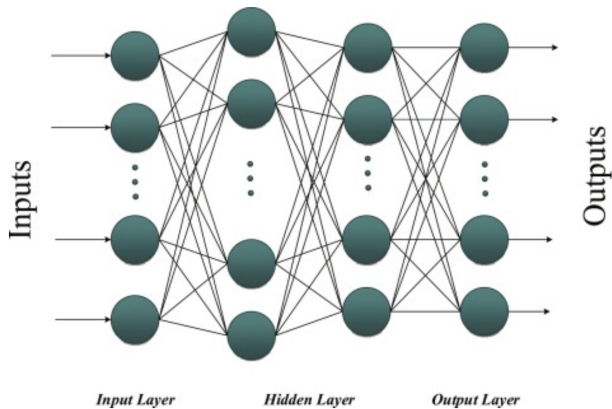


Abbildung 5: Multi Layer Perceptron

$$\begin{array}{ccccc} \text{Inputvariable(n) } (\overline{X}) & \rightarrow & \text{Funktion} & \rightarrow & \text{Zielvariable(n) } (y) \\ \overline{X} = [x_1, x_2, \dots, x_d] & & y = f_{\overline{W}}(\overline{X}) & & y \end{array}$$

Ein künstliches neuronales Netz ist nichts anderes als eine Funktion, die einem Inputvektor $\overline{X}$ in Abhängigkeit von einem Gewichtsvektor $\overline{W}$ einen Zielwert y zuweist.

Grundproblem des (maschinellen) Lernens

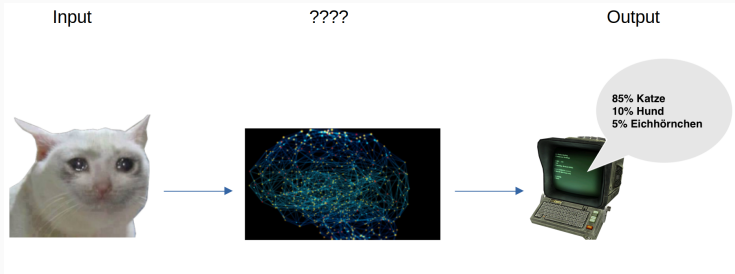


Abbildung 6: Exemplarisches Schema eines Entscheidungsproblems

Berechnung eines binären Entscheidungsproblems mithilfe eines KNN

$$\hat{y} = f(\bar{X}) = \text{sign}\{\bar{W} \cdot \bar{X}^T\} = \text{sign}\left\{\sum_{j=1}^d w_j x_j\right\}$$
$$\Rightarrow \hat{y} = \text{sign}\{x_1 \cdot w_1 + x_2 \cdot w_2 + x_3 \cdot w_3 + \dots + x_d \cdot w_d\}$$

Trainingsproblem künstlicher neuronaler Netze:

Minimiere die Abweichung zwischen $f_{\overline{W}}(\overline{X})$ und y durch Anpassung von $\overline{W}$.

Hauptbestandteile eines künstlichen neuronalen Netzwerks

- Verschiedene neuronale Schichten mit gewichteten Verbindungen
- Kostenfunktion zur Berechnung der Abweichung
- Optimierer zur Minimierung der Abweichung
- ggf. versch. Aktivierungsfunktionen, um Daten auf nichtlineare Weise zu transformieren

- Die Abweichung von dem Zielwert wird i.d.R. in Form einer **Kostenfunktion** (a.k.a. loss function, error function) ermittelt.
- die Kostenfunktion quantifiziert die Abweichung zwischen $\hat{y}$ und y
- die Kostenfunktion soll den Algorithmus für hohe Abweichungen in Form von Kosten bestrafen
- Es existieren verschiedene Kostenfunktionen, die je nach zu lösendem Problem unterschiedlich gewählt werden sollten.
- Beispiele:

Wie hoch sind die erwarteten Einnahmen meines Limonadenstands, wenn die Temperatur morgen x beträgt?

- Bei Regressionsproblemen wird häufig die mittlere quadratische Abweichung (MSE) als Kostenfunktion verwendet
- $$MSE = \frac{\sum_{i=1}^n (y_i - \bar{y}_i)^2}{n}$$

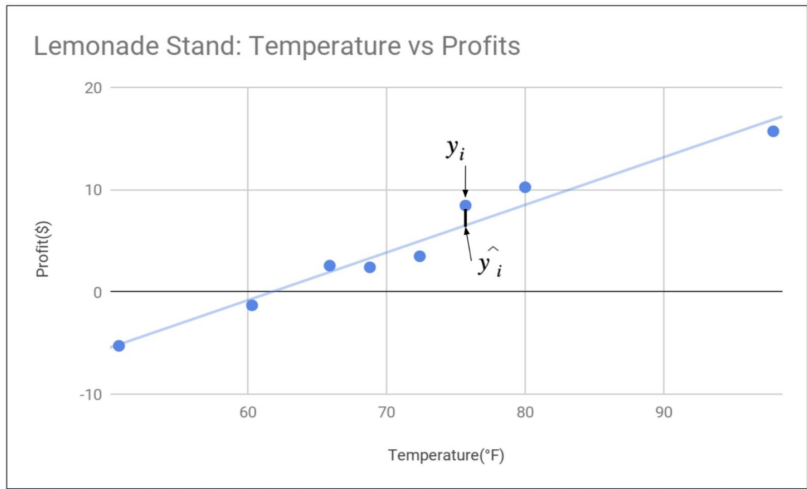
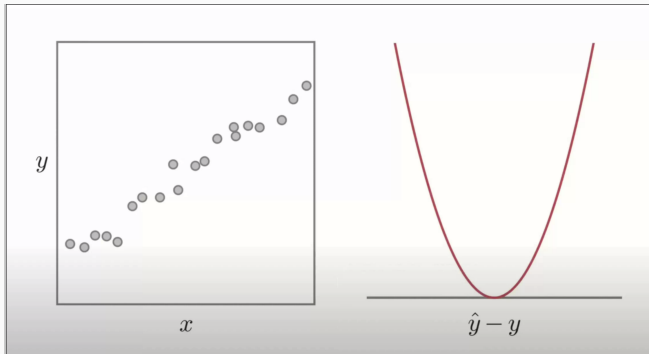


Abbildung 7: Mittlere quadratische Abweichung



Ist das gezeigte Bild ein Hund oder eine Katze?

- Kreuzentropie (engl. Cross-Entropy) ist eine gängige Kostenfunktion bei der Klassifikation mit neuronalen Netzwerken.
- Sie misst den Unterschied zwischen der tatsächlichen Verteilung der Klassen und der vorhergesagten Verteilung.
- Insbesondere für binäre Klassifikation und Multi-Klassen-Klassifikation nützlich.

Kostenfunktion: Kreuzentropie bei Klassifikationsproblemen

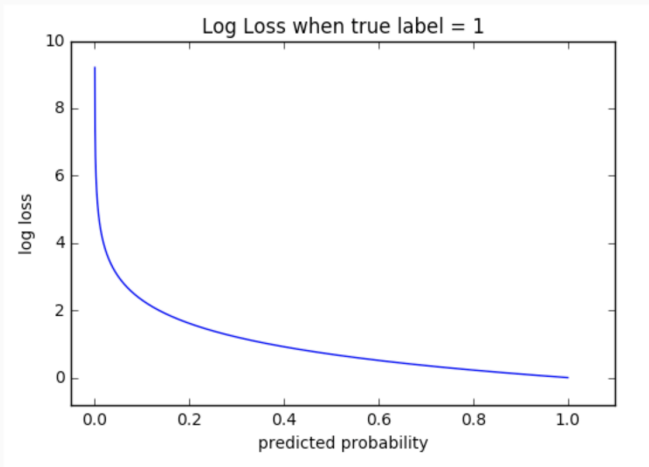


Abbildung 8: Kreuzentropie-Kostenfunktion bei binärem Klassifikationsproblem (z.B. Hund vs. Katze)

- Für eine binäre Klassifikation:

$$L(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

- Für eine Multi-Klassen-Klassifikation:

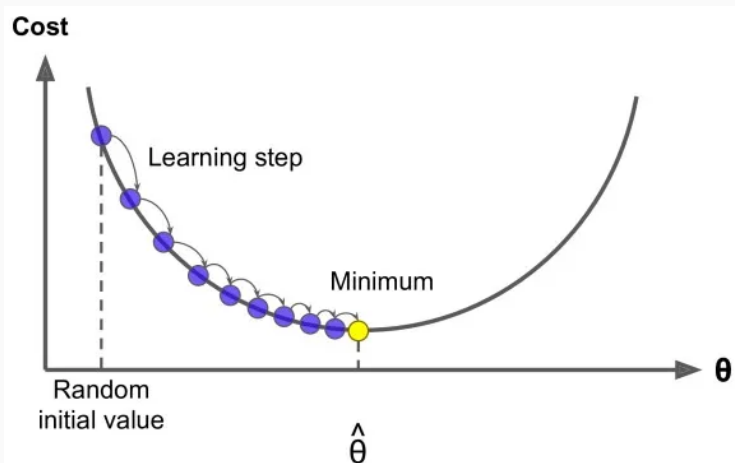
$$L(y, \hat{y}) = - \sum_{i=1}^C y_i \log(\hat{y}_i)$$

Rekapitulation

- Ein KNN ist eine Funktion, die in Abhängigkeit des Inputs und der internen Gewichtung eine **Vorhersage** macht
- Ein KNN löst i. d. R. Regressionsprobleme oder Klassifikationsprobleme
- Abweichungen zwischen dem Vorhersagewert und dem Zielwert werden mithilfe einer Kostenfunktion ‚bestraft‘
- die Kostenfunktion variiert je nach Problem und Datenbeschaffenheit

- Backpropagation $\approx$ Anpassung der Gewichtungen auf Basis der Abweichung vom Zielwert
- Optimizer haben die Aufgabe, die Gewichtungen des neuronalen Netzwerks so anzupassen, dass die Kosten möglichst gering sind
- Gradientenabstiegsverfahren (Gradient Decent) als klassische Lösung des Optimierungsproblems

Gradientenabstiegsverfahren



- Transformieren die Ausgaben von Neuronen auf nicht-lineare Weise
- Ermöglichen die Einführung von Nichtlinearität in das Netzwerk
- Nichtlinearität ist notwendig, um komplexe Muster und Abstraktionen in den Daten zu erfassen

Aktivierungsfunktionen

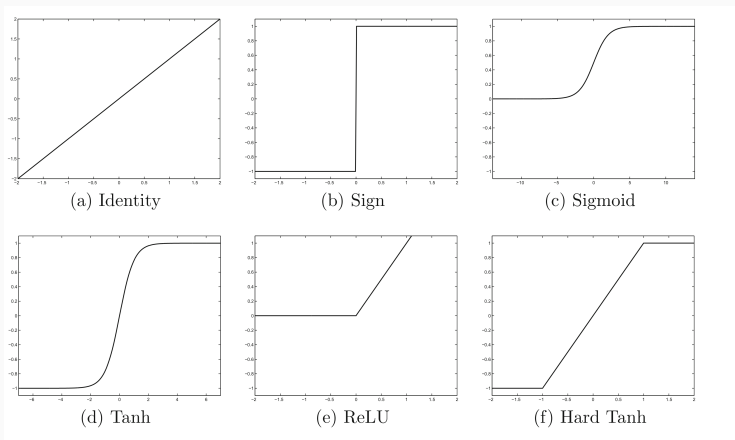


Abbildung 9: Einige Aktivierungsfunktionen (Aggarwal 2023, S. 13)

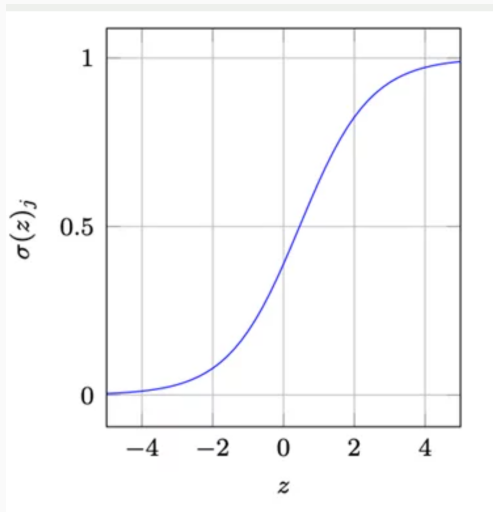


Abbildung 10: Softmax-Aktivierungsfunktion

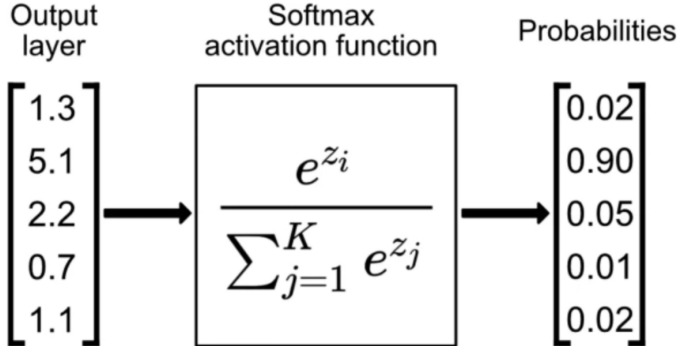


Abbildung 11: Softmax-Aktivierungsfunktion in Anwendung

Anwendung und Kernproblematik in der Computerlinguistik

- Maschinelle Übersetzung
- Sprachmodellierung und Textgenerierung
- Spracherkennung
- Sentiment-Analyse
- Named Entity Recognition
- Parsing und Syntaxanalyse
- Textklassifikation
- Sprachsynthese
- Dialogsysteme und Chatbots

Beispiel: Sentiment-Analyse

Kannst Du mir den Wert der Apple-Aktie für morgen vorhersagen?



Entschuldigung, aber ich kann keine zukünftigen Aktienkurse vorhersagen. Die Börse ist von vielen Faktoren abhängig, darunter wirtschaftliche Bedingungen, Unternehmensleistung, Nachrichten und Marktstimmung, die alle sehr volatil sein können. Es ist am besten, sich auf fundierte Analysen und Expertenmeinungen zu verlassen, um Investitionsentscheidungen zu treffen.

Toll, das ist ja wirklich großartig.



Es tut mir leid, wenn das nicht die Antwort war, die du erwartet hast. Wenn du weitere Fragen hast oder etwas anderes wissen möchtest, stehe ich gerne zur Verfügung!



Abbildung 12: Beispiel für gelungene Sentiment-Analyse

Beispiel: Sentiment-Analyse

Kannst Du mir den Wert der Apple-Aktie für morgen vorhersagen?



Es tut mir leid, aber ich kann keine spezifischen Vorhersagen über den zukünftigen Wert von Aktien machen. Die Börse ist von vielen Faktoren abhängig, darunter wirtschaftliche Indikatoren, Unternehmensleistung, Marktstimmung und externe Ereignisse, die schwer vorhersehbar sind. Es ist immer ratsam, sich an einen Finanzberater oder Analysten zu wenden, um Investitionsentscheidungen zu treffen.

Toll, das ist ja wirklich absolut großartig. Ich bin richtig begeistert



Es freut mich, dass meine Antwort hilfreich war! Wenn du Fragen zu anderen Themen hast oder etwas Bestimmtes besprechen möchtest, stehe ich gerne zur Verfügung.



Abbildung 13: Beispiel für misslungene Sentiment-Analyse

SoftBank's new AI makes angry customers sound calm on phone

By TAKESHI NARABE/ Staff Writer

June 11, 2024 at 16:56 JST

[f](#) Share [T](#) Tweet [P](#) Print [list](#)



Employees of SoftBank Corp. who work on the development of AI voice-conversion technology. From left, Toshiyuki Nakatani, Toshio Imamura and Shuhei Takahara. (Provided by SoftBank Corp.)

- Inputlayer kann nur numerische Vektoren entgegen nehmen
- Sprache muss daher in numerische Vektoren umgewandelt werden
- Wörter werden i. d. R. auf morphemähnliche Weise als numerische **Tokens** repräsentiert.

Methoden zur Umwandlung von Sprache in numerische Vektoren

- **One-Hot Encoding**
 - Jedes Wort wird durch einen binären Vektor repräsentiert.
- **Bag-of-Words (BoW)**
 - Wörter werden durch ihre Häufigkeiten in Dokumenten dargestellt.
- **Term Frequency-Inverse Document Frequency (TF-IDF)**
 - Gewichtet Wörter basierend auf ihrer Häufigkeit und Seltenheit.
- **Word Embeddings**
 - Dichtere Vektoren, die semantische Ähnlichkeiten erfassen (z.B. Word2Vec, GloVe, FastText).
- **Contextual Embeddings**
 - Kontextabhängige Vektoren (z.B. BERT, GPT, ELMo).
- **Sentence Embeddings**
 - Vektoren für ganze Sätze (z.B. Universal Sentence Encoder, Sentence-BERT).

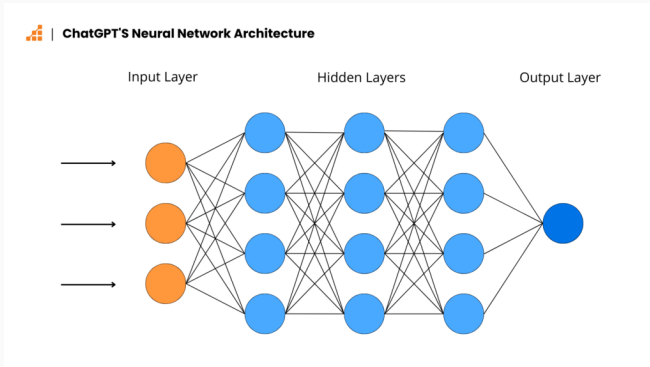


Abbildung 14: ChatGPT-3.5-Architektur: 175 mrd. Parameter, 96 Schichten

Beispiel: GPT als Transformer

- GPT basiert auf der bahnbrechenden Transformer-Architektur von Vaswani u. a. (2017)
- **Schlüsselkomponenten:**
 - **Self-Attention:** Gewichtung der Bedeutung von Eingabeelementen untereinander
 - **Multi-Head Attention:** Mehrere Aufmerksamkeitsebenen für vielfältige Kontextinformation
 - **Residual Connections:** Verbessern die Gradientenausbreitung
 - **Positional Encoding:** Fügt Reihenfolgeninformationen hinzu

Literatur



Aggarwal, Charu C. (2023). **Neural Networks and Deep Learning. A Textbook.** 2. Aufl. Springer Cham.



Hebb, Donald O. (1949). **The organization of behavior: A neuropsychological theory.** New York: Wiley. ISBN: 0-8058-4300-0.



Vaswani, Ashish u. a. (2017). **“Attention is All you Need”.**
In: *Neural Information Processing Systems*. URL:
<https://api.semanticscholar.org/CorpusID:13756489>.

Bildquellen

- Abb. 1: <https://www.mpg.de/synapse>
- Abb. 2: <https://www.doccheck.com/de/detail/photos/20984-aktionspotential>
- Abb. 5 <https://www.geeksforgeeks.org/multi-layer-perceptron-learning-in-tensorflow/>
- Abb. 7:
<https://www.machinelearningworks.com/tutorials/mean-squared-error-cost-function>
- Abb. 8. https://ml-cheatsheet.readthedocs.io/en/latest/loss_functions.html
- Animation: John Barron,
<https://www.youtube.com/watch?v=BmNKbnF69eY>
- Abb. 10 <https://databasecamp.de/ki/softmax>
- Abb. 11 <https://www.singlestore.com/blog/a-guide-to-softmax-activation-function/>