

# Intelligente Systeme Praktikum

## Aufgabe 2 - Lernen

1. November 2019

Adrian Helberg, Rodrigo Ehlers

# Inhalt

- 1 Thema
- 2 Neuronale Netze
- 3 MNIST
- 4 Aktivierungsfunktion - Überlegung
- 5 Lernregel

# Thema

## Beschreibung

Handschrifterkennung mittels künstlichem neuronalem Netz (KNN)

## Überlegungen

Wir benötigen:

- Neuronales Netz: Hier findet das Lernen statt
- Lerndaten: Datensatz, durch den das KNN lernt
- Testdaten: Validieren des KNN, Bestimmen der "Genauigkeit"
- Anwendungsdaten: Möglichkeit handgeschriebene Zahlen dem KNN zu "übergeben"

# Neuronale Netze - Allgemeine Beschreibung

## Definition

Als neuronales Netz wird in den Neurowissenschaften eine beliebige Anzahl miteinander verbundener Neuronen bezeichnet, die als Teil eines Nervensystems einen Zusammenhang bilden, der einer bestimmten Funktion dienen soll

## Eignet sich die Problemstellung?

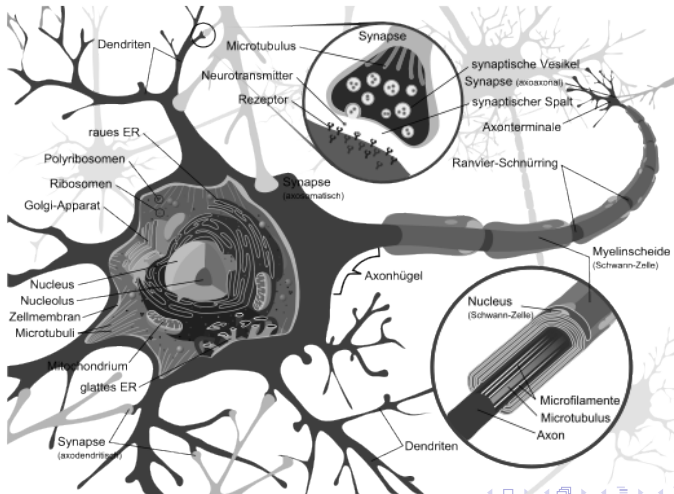
Eignet sich ein KNN überhaupt für das Lösen des Problems, oder lässt es sich auch durch das direkte Implementieren in einer Programmiersprache lösen?

# Neuronale Netze - Überlegung

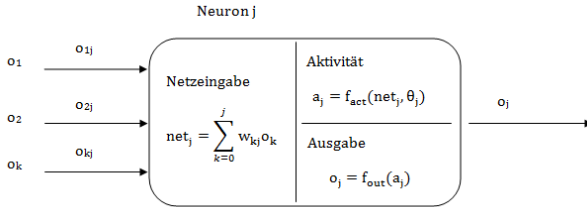
## Eignet sich die Problemstellung?

Da sich das Erkennen von handschriftlichen Zeichen nicht direkt in einem Flussdiagramm ableiten lässt, und sich daher nicht direkt in einer Programmiersprache implementieren lässt, eignet sich ein Neuronales Netz gut für das Problem. Weiter eignet sich ein KNN besonders gut für ein Problem, das sich stetig verändert und dessen Lösung eine Anpassung verlangt

# Neuronale Netze - Neuron



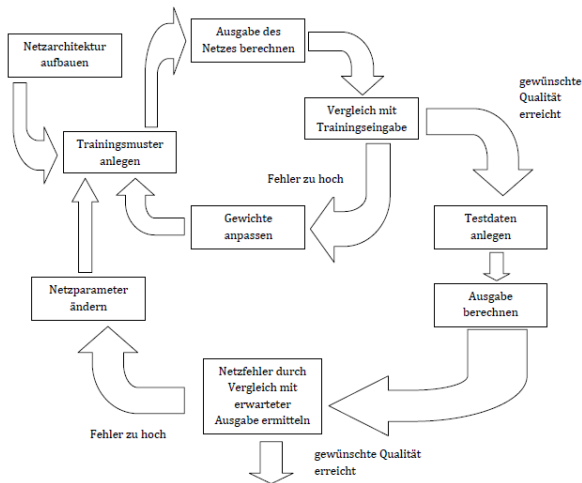
# Neuronale Netze - Formales Neuron



## Beschreibung

- Netzeingabe: Propagieren aller Eingabeinformationen mit den Gewichten der Verbindungen zu einer einzigen Netzeingabe
- Aktivität: Ermitteln eines Aktivierungszustands  $a_j$  mittels Aktivierungsfunktion  $f_{\text{act}}$
- Ausgabe: Weitergeben des Aktivierungszustands  $o_j$

# Neuronale Netze - Lernprozess





# Neuronale Netze - Lernen

Welche Lernmethode eignet sich gut?

Überwachtes, nicht überwachtes oder bestärkendes Lernen?

Lösung

Die "MNIST-Datenbank mit handgeschriebenen Ziffern" stellt sowohl einen Datensatz mit potentiellen Lerndaten, als auch einen Datensatz zum Testen zur Verfügung

# MNIST

## The MNIST database

The MNIST database of handwritten digits, available from this page, has a training set of 60,000 examples, and a test set of 10,000 examples. It is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image. It is a good database for people who want to try learning techniques and pattern recognition methods on real-world data while spending minimal efforts on preprocessing and formatting.

---

<http://http://yann.lecun.com/exdb/mnist/>

# Aktivierungsfunktion

## Biologisches Vorbild

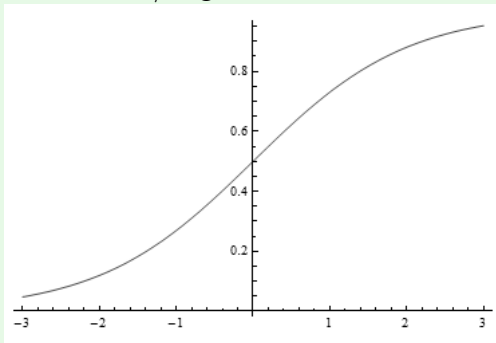
- Biologische Neuronen arbeiten analog, d.h. sie können auf kontinuierliche Schrittfolgen reagieren
- Computer sind digitale Rechenmaschinen mit nur zwei Zuständen, 0 oder 1 (Quantencomputer ausgeschlossen 😊)
- Künstliche Neuronen müssen deshalb analoge Schwellwertfunktionen nachbilden

# Aktivierungsfunktion

## Sigmoidfunktionen

Schwellwertfunktion / logistische Funktion

$$f_{\text{logistic}}(x) = \frac{1}{1 + e^{-cx}}$$



# Lernregel - Beschreibung

Die Lernregel gestattet es, dass ein KNN durch eine gegebene Aufgabe (weitgehend) selbständig aus Beispielen lernt

## Die Lernregel bestimmt

- die Entwicklung neuer Verbindungen
- das Löschen existierender Verbindungen
- die Modifikation der Stärke  $w_i$  von Verbindungen
- die Modifikation des Schwellenwertes von Neuronen
- die Modifikation der Aktivierungs-, Propagierungs- oder Ausgabefunktion
- die Entwicklung neuer Neuronen
- das Löschen von Neuronen

# Lernregel - Überlegung

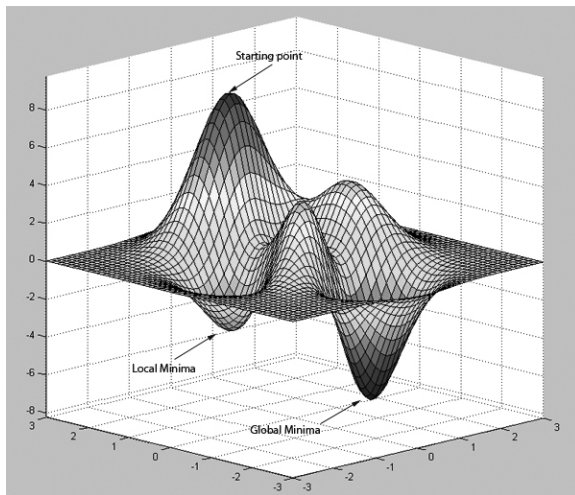
## Überlegung

Da wir eine nicht-lineare Aktivierungsfunktion gewählt haben, muss die Lernregel semilinear, d.h. monoton und differenzierbar, sein

## Backpropagation

Die Backpropagation-Regel erfüllt die Kriterien

# Lernregel - Backpropagation I



# Lernregel - Backpropagation II

## Korrektur von Gewichten

Die partielle erste Ableitung der Fehlerfunktion nach einer Gewichtsvariablen kann für die Korrektur dieses Gewichtes verwendet werden:

$$\Delta w_{ij} = -\gamma \frac{\partial E}{\partial w_{ij}}$$

→

$$\Delta w_k(i) = -\gamma \frac{\partial E}{\partial w_k} + \alpha \Delta w_k(i-1)$$

Mit  $E$  als mögliche Fehlerkurve  $E(w_1, w_2)$  für zwei Gewichte;  
Lernrate  $\gamma$ , als Grad der Änderung,  $\alpha$ , als Momentum