Begriffbildung, Konzepte und Überblick

Seminar Deep Learning

Alexander Strätz

Universität Leipzig

12. Januar 2018

UNIVERSITÄT LEIPZIG

Inhaltsverzeichnis

Motivation

Wo ist deep learning anzufinden?:

- Websuche
- Filterfunktionen in sozialen Netzwerken
- Empfehlungen im Bereich e-commerce
- Kameras/Smartphones

Motivation

deep learning wird genutzt um:

- Objekte in Bildern zu identifizieren
- Sprache zu Text zu übersetzen
- Produkte auf Nutzerinteressen abzustimmen
- relevante Suchergebnisse festzustellen

Motivation

Konventionelles Machine-learning:

- Fachkompetenz im Anwendungsbreich nötig
- transformieren der rohen Daten in eine geeignete interne Repräsentation

Deep learning:

- rohe Eingabedaten
- automatisches Entdecken der geeigneten Repräsentation
- mehrere Abstraktionslevel der Repräsentation

Aufbau künstlicher neuronaler Netze

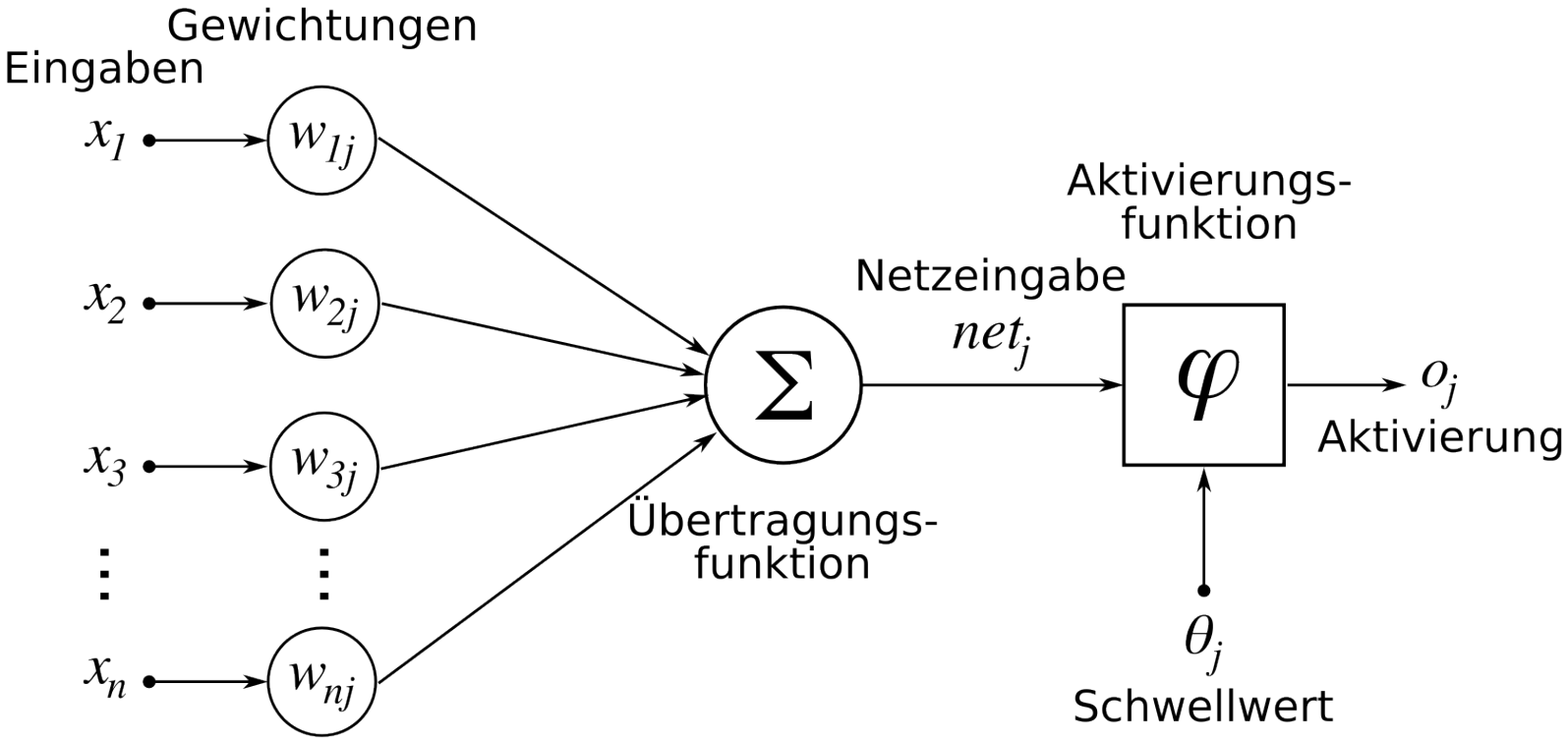


Abbildung: Aufbau des künstlichen Neurons

Aufbau künstlicher neuronaler Netze

- Aktivierungsfunktion: Zusammenhang zwischen Netzinput und Aktivitätslevel eines Neurons
- Sigmoid Funktion: $\sigma(z) = \frac{1}{1+e^{-z}}$

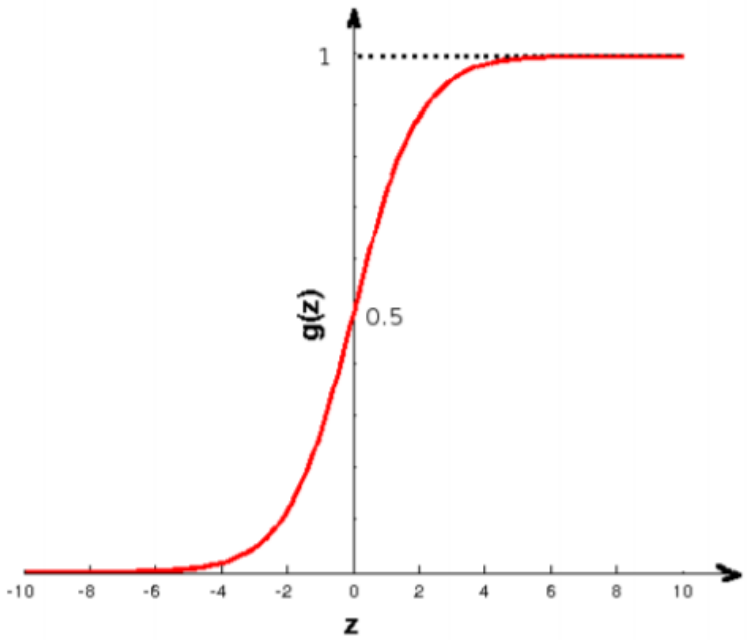


Abbildung: Sigmoid-Funktion

Aufbau künstlicher neuronaler Netze

- empirisch oft bessere Ergebnisse: rectified linear units (ReLU)
- $ReLU(z) = max(0, z)$

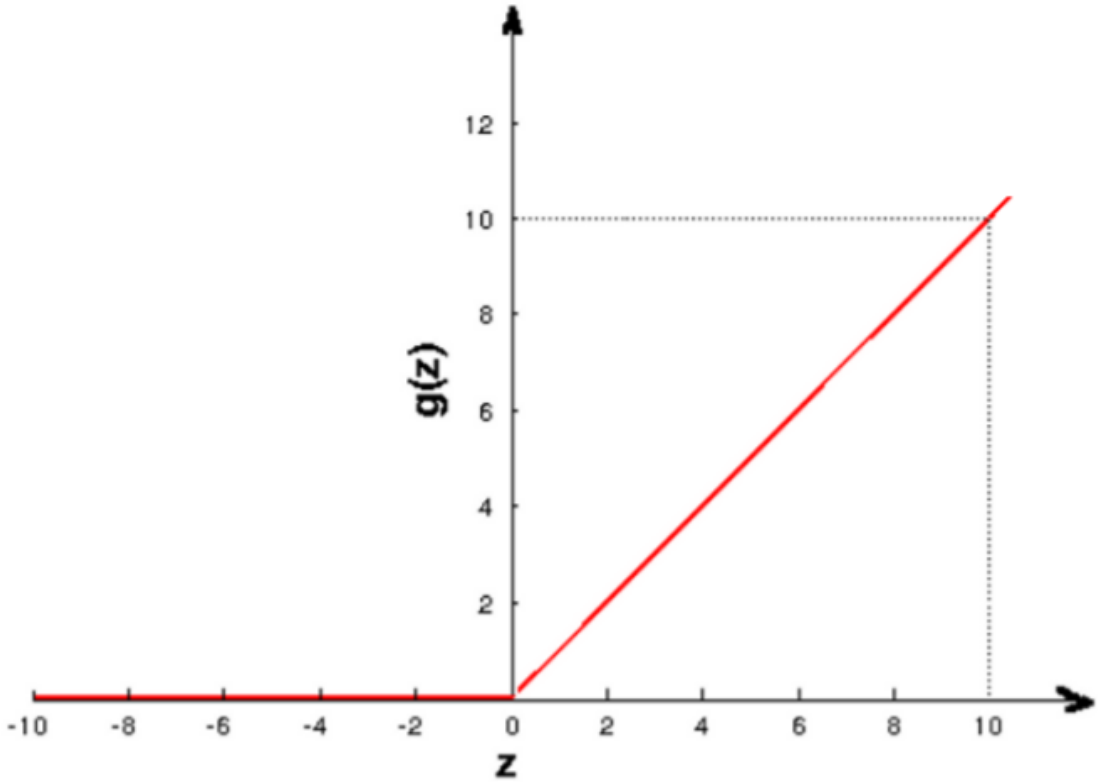


Abbildung:

Aufbau künstlicher neuronaler Netze

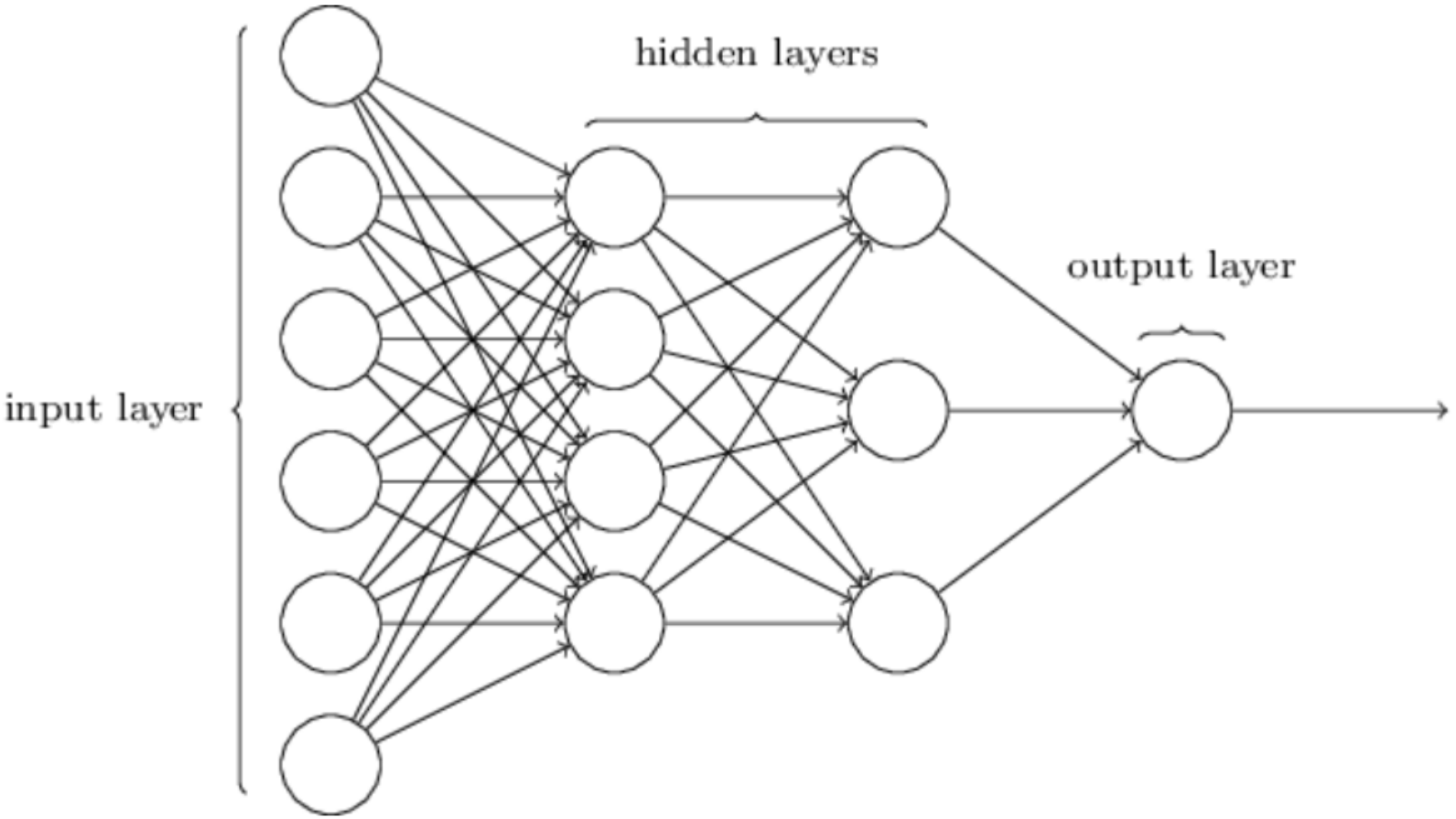


Abbildung: Schichten im KNN

Aufbau künstlicher neuronaler Netze

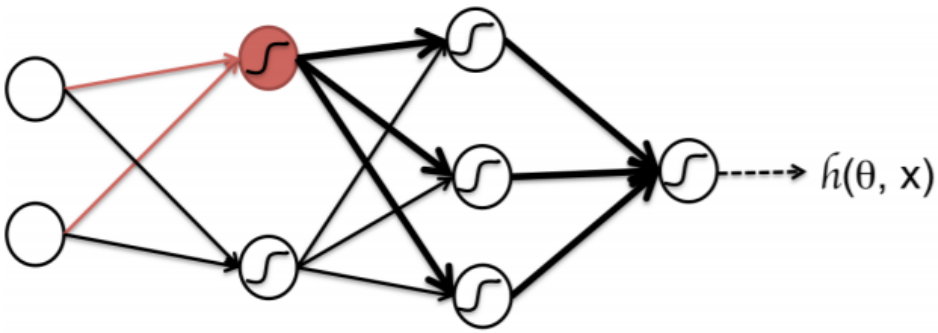


Abbildung: Tiefes Netz

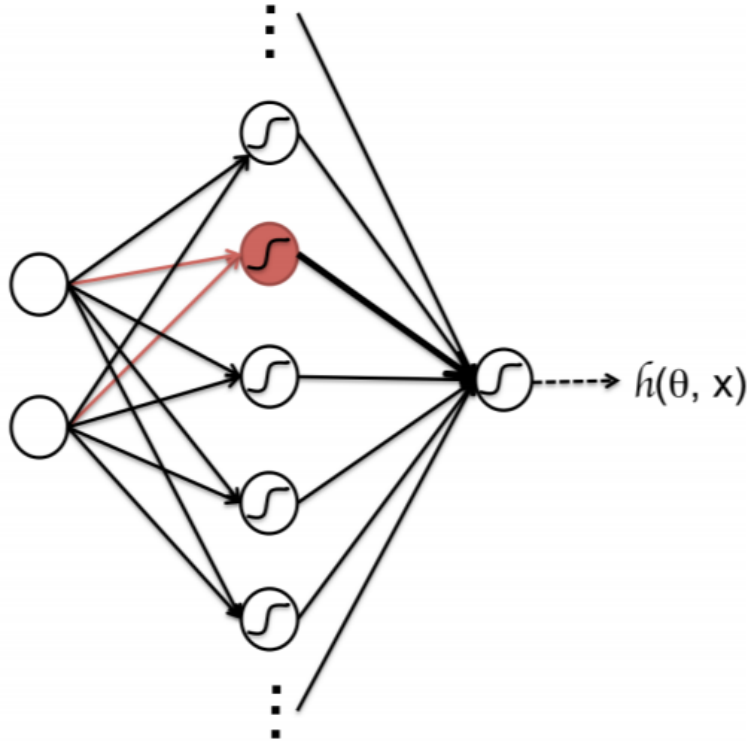


Abbildung: Flaches Netz

Aufbau künstlicher neuronaler Netze

- Convolutional Neural Network (CNN)
  - Anwendungsbeispiel: Verarbeitung von Bilddaten
  - besteht aus Convolutional Layer und Pooling Layer
  - z.b. zuerst Kantenerkennung, anschließend Reduktion auf Maximum

- Recurrent Neural Network (RNN)
  - Anwendungsbeispiel: Spracherkennung
  - Verbindungen von Neuronen einer Schicht zu Neuronen derselben oder einer vorangegangenen Schicht
    $\rightarrow$ Verarbeitung von Sequenzen

Funktionsweise: Beispiel Filmempfehlung

Movie name	Mary's rating	John's rating	I like?
Lord of the Rings II	1	5	No
...	...	...	...
Star Wars I	4.5	4	Yes
Gravity	3	3	?

Abbildung:

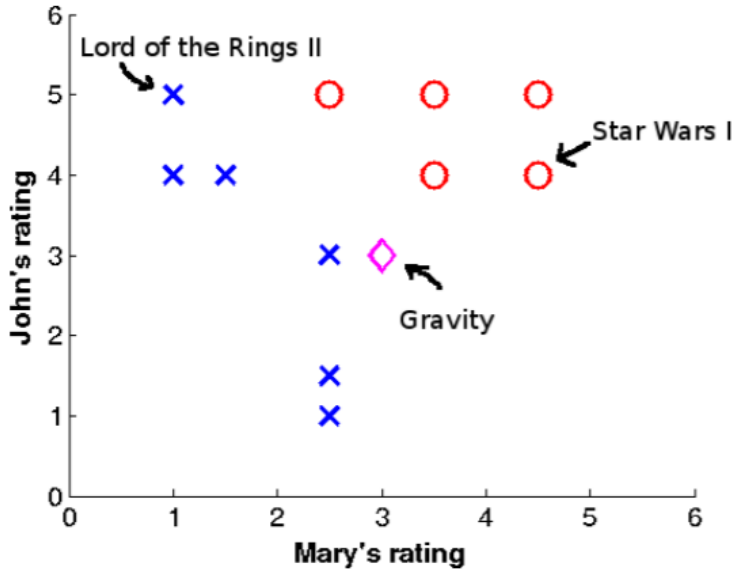


Abbildung:

Funktionsweise: Beispiel Filmempfehlung

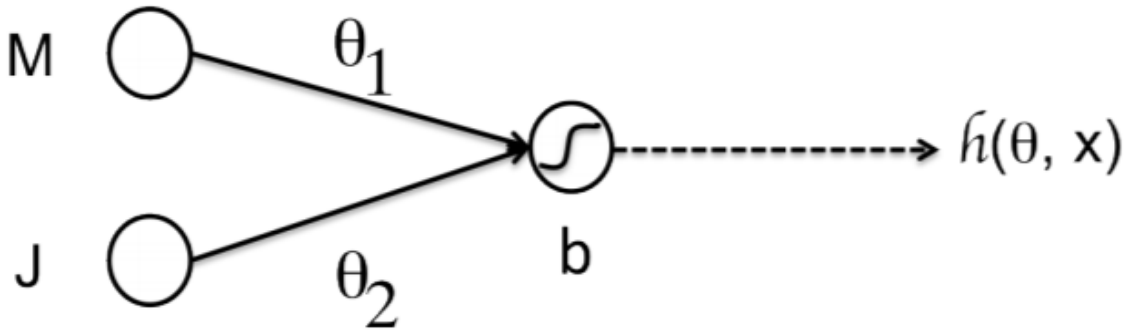


Abbildung:

- Initialisiere θ und b zufällig
- Ziel: $h(x^{(1)}; \theta, b) \approx y^{(1)}$
  $h(x^{(2)}; \theta, b) \approx y^{(2)}$
  ...

Funktionsweise: Gradientenverfahren

- minimiere Kostenfunktion: $\sum_{i=1}^{m}(h(x)^{(i)};\theta,b)-y^{(i)})^2$
- stochastisches Gradientenabstiegsverfahren:
  $\theta_1 = \theta_1 - \alpha\Delta\theta_1$
  $\theta_2 = \theta_2 - \alpha\Delta\theta_2$
  $b = b - \alpha\Delta b$
- α: Lernfaktor (kleines nicht-negatives Skalar)
- schätze Gradienten mit zufällig ausgewählten Trainingsbeispielen

Funktionsweise: Gradientenverfahren

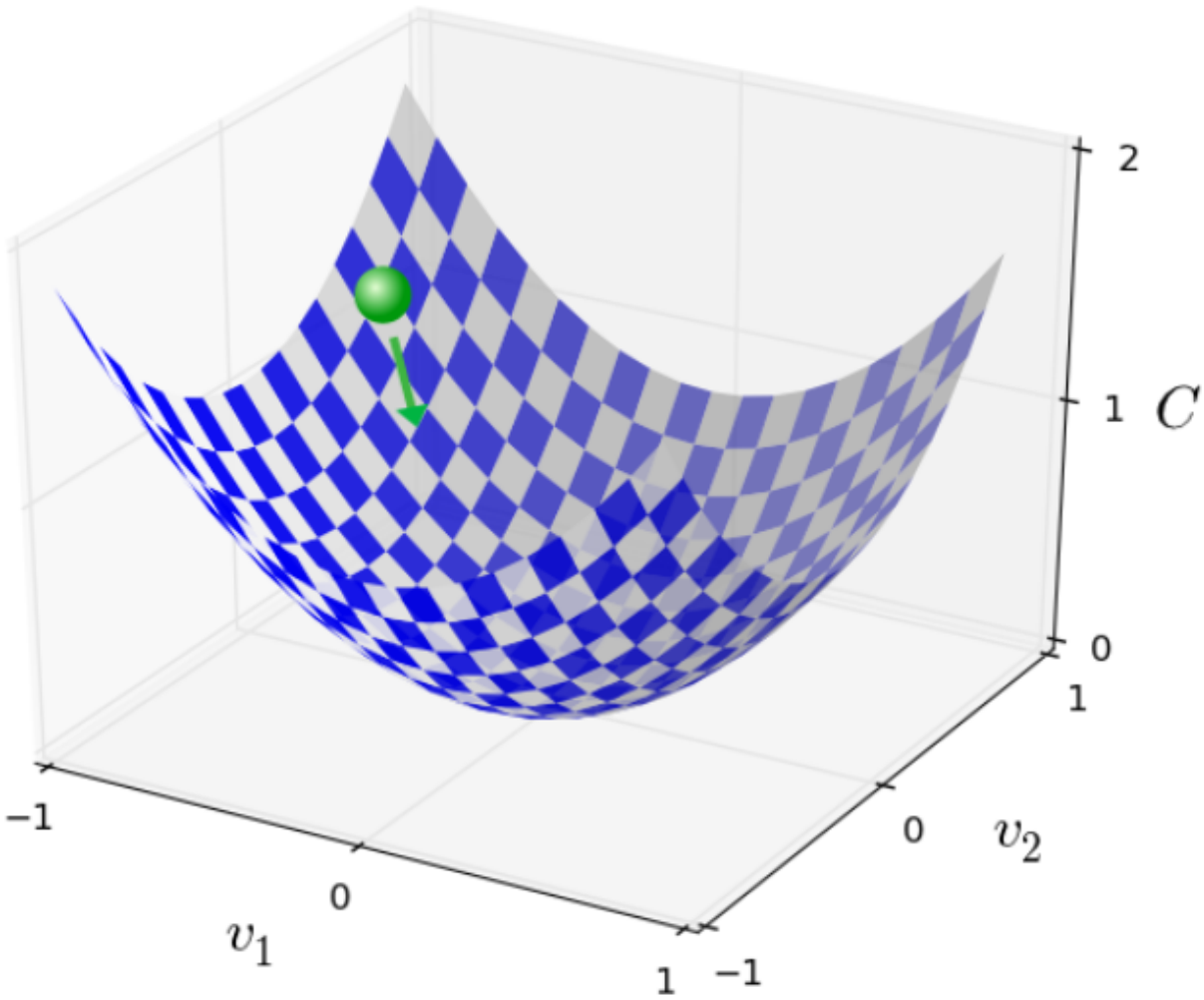
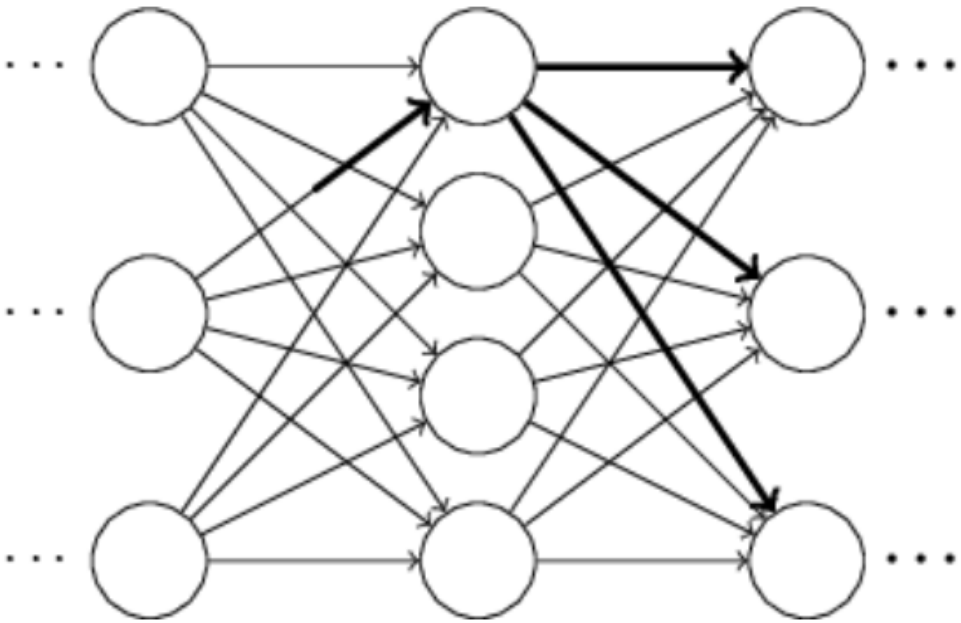


Abbildung: Gradientenverfahren

Backpropagation Algorithmus

- Ziel: Berechne $\partial C/\partial w$ und $\partial b/\partial b$
- **Input**: Setze die Aktivierung für den Input-Layer
- **Feedforward**: Berechne Input $z^l = w^l a^{l-1} + b^l$ und Aktivierung $a^l = \sigma(z^l)$
- **Output Fehler**: $\delta^l = \frac{\partial C}{\partial a_L^j} \sigma'(z_j^L)$
- **Backpropagation des Fehlers**: $\delta^l = (w^{l+1}\delta^{l+1}) * \sigma'(z^l)$
- **Output**: Berechne $\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1}\delta_j^l$

Funktionsweise



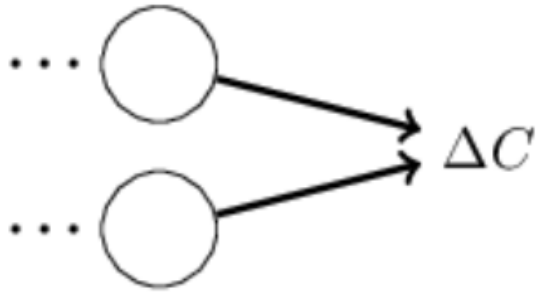


Abbildung:

Funktionsweise

- Stopp des Gradientenverfahrens, wenn sich Parameter nicht mehr ändern oder nach festgelegter Anzahl von Iterationen
- Entscheidungsfunktion $h(x; \Theta, b)$ bestimmt ob man den Film mag oder nicht
- $h > 0,5$: ja, $h < 0,5$: nein

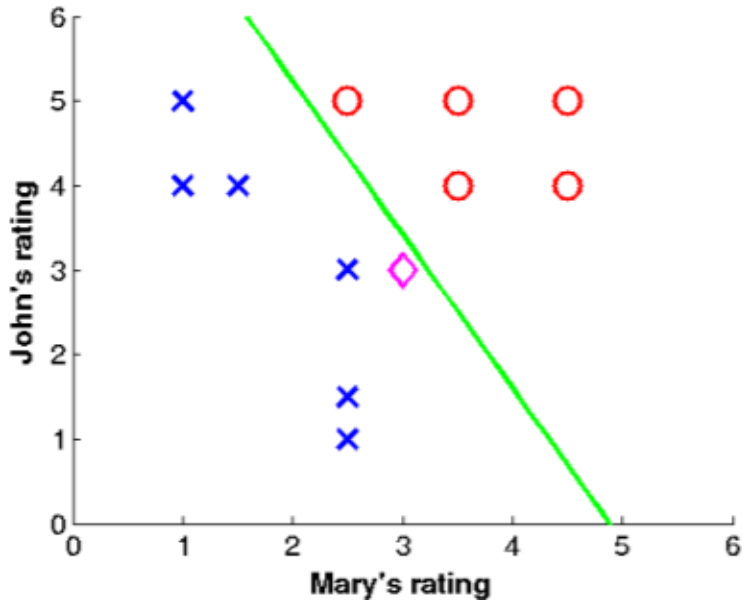
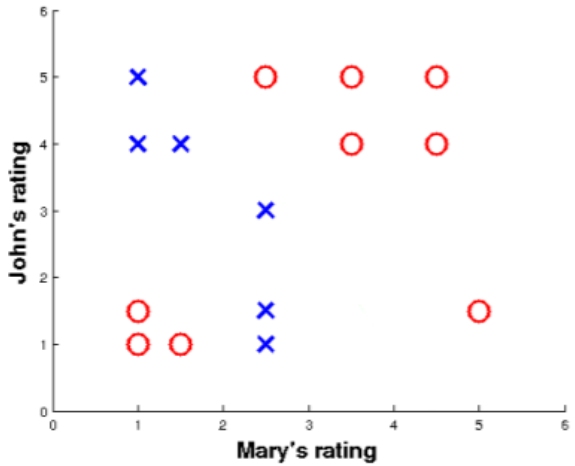


Abbildung:

Funktionsweisen



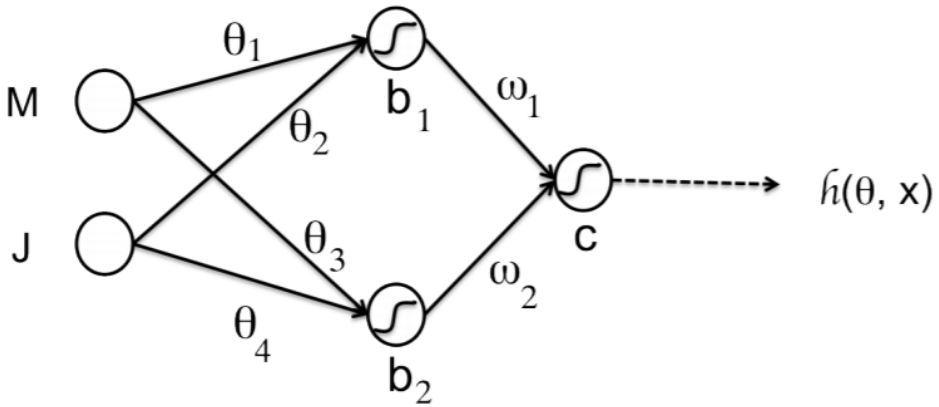


Abbildung:

Funktionsweise: Beispiel

- realistischeres Beispiel: Erkennen von handgeschriebenen Zahlen

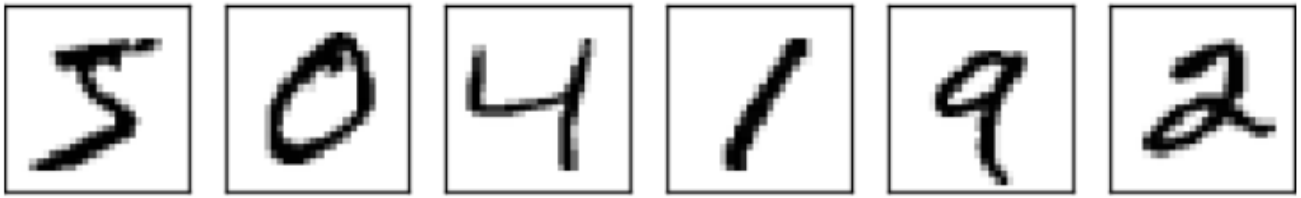


Abbildung:

- Zahlen bestehen aus 64x64 Graustufenbildern
  $\rightarrow$ Input-Layer aus $64x64 = 4096$ Neuronen zwischen 0 (weiß) und 1 (schwarz)
- 10 Output-Neuronen, die jeweils eine Ziffer repräsentieren

Funktionsweise: Beispiel

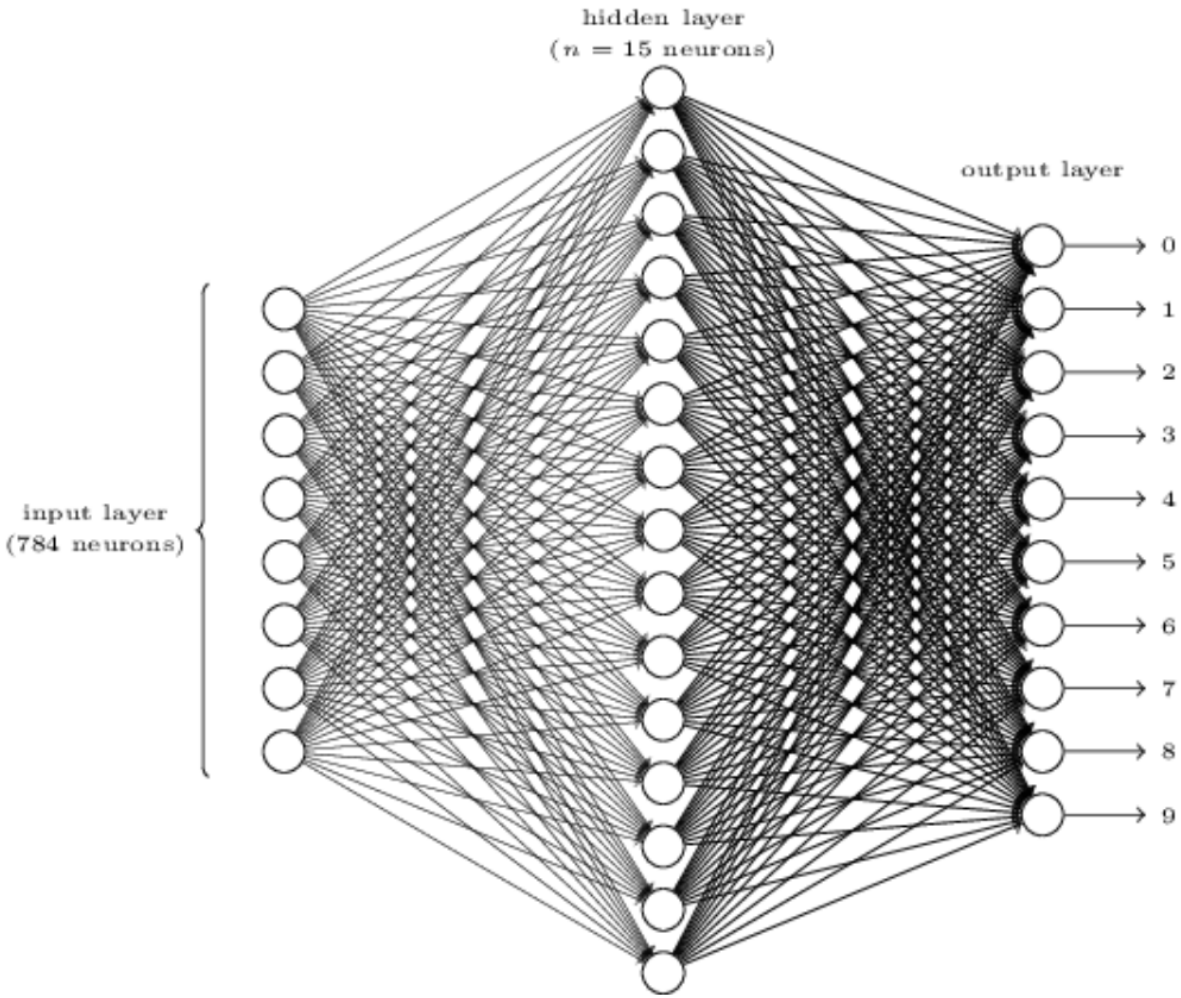


Abbildung:

Probleme/Lösungen: Overfitting

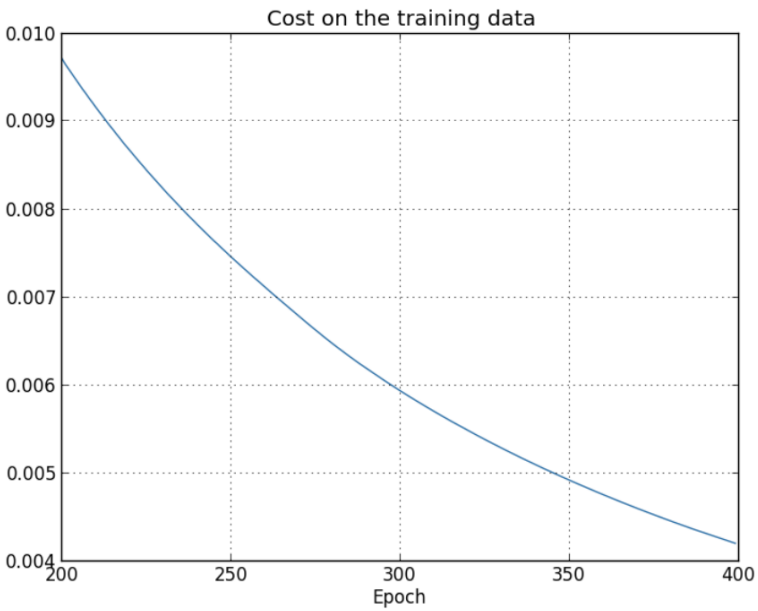


Abbildung: Gesamtkosten auf den Trainingsdaten

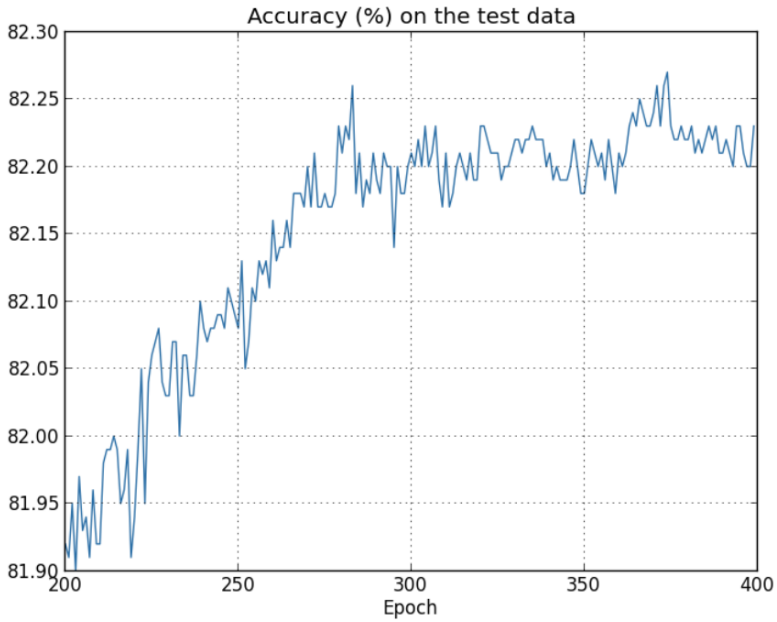


Abbildung: Genauigkeit der Klassifikation auf den Testdaten

Overfitting

Lösungsmöglichkeiten:

- Berechnen der Klassifikationsgenauigkeit nach jeder Epoche auf Validierungsdaten
  $\rightarrow$ Stoppen des Trainings nachdem die Klassifikationsgenauigkeit nicht mehr zunimmt

- L2-Regularisierung: $C_{neu} = C_{alt} + \sum_w w^2$
  $\rightarrow$ kleine Gewichte werden vorgezogen

- L1-Regularisierung: $C_{neu} = C_{alt} + \sum_w |w|$
  $\rightarrow$ Schrumpfen der Gewichte um konstanten Betrag

Overfitting

Lösungsmöglichkeiten:

- **Dropout:** Bei jedem Trainingsdurchgang lösche die Hälfte der hidden Neuronen
- Anschließend halbiere die Gewichte, die von den hidden Neuronen ausgehen

- **Erweitere künstlich die Trainingsdaten:** Modifiziere leicht die originalen Trainingsdaten
- zum Beispiel durch Rotation und Translation bei Bilddaten

Initialisierung der Gewichte

- Initialisierung der Gewichte mit Mittelwert 0 und Standardabweichung 1
- 1000 eingehende Neuronen, jeweils 50% besitzen Wert 0 und 1

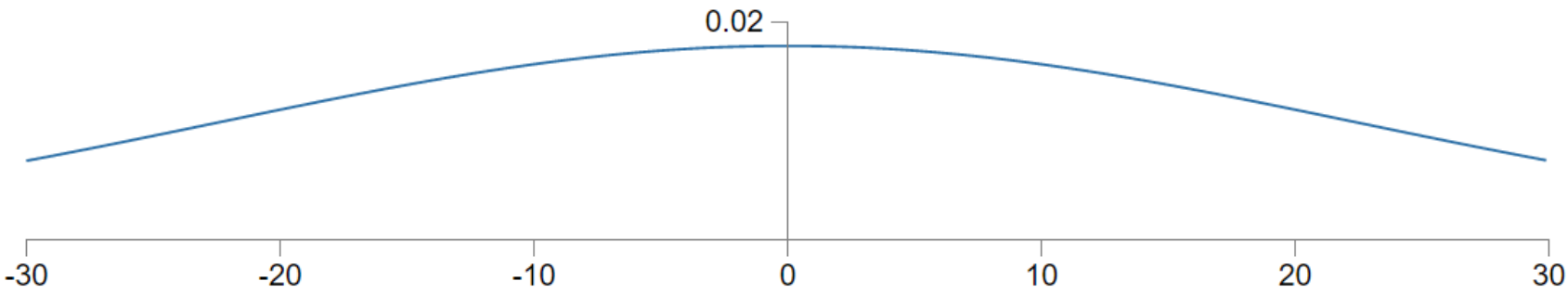


Abbildung:

$\rightarrow$ sehr hohe Wahrscheinlichkeit für $z << 1$ oder $z >> 1$

Initialisierung der Gewichte

- Initialisiere Gewichte mit Mittelwert 0 und Standardabweichung $1/\sqrt{n_{in}}$

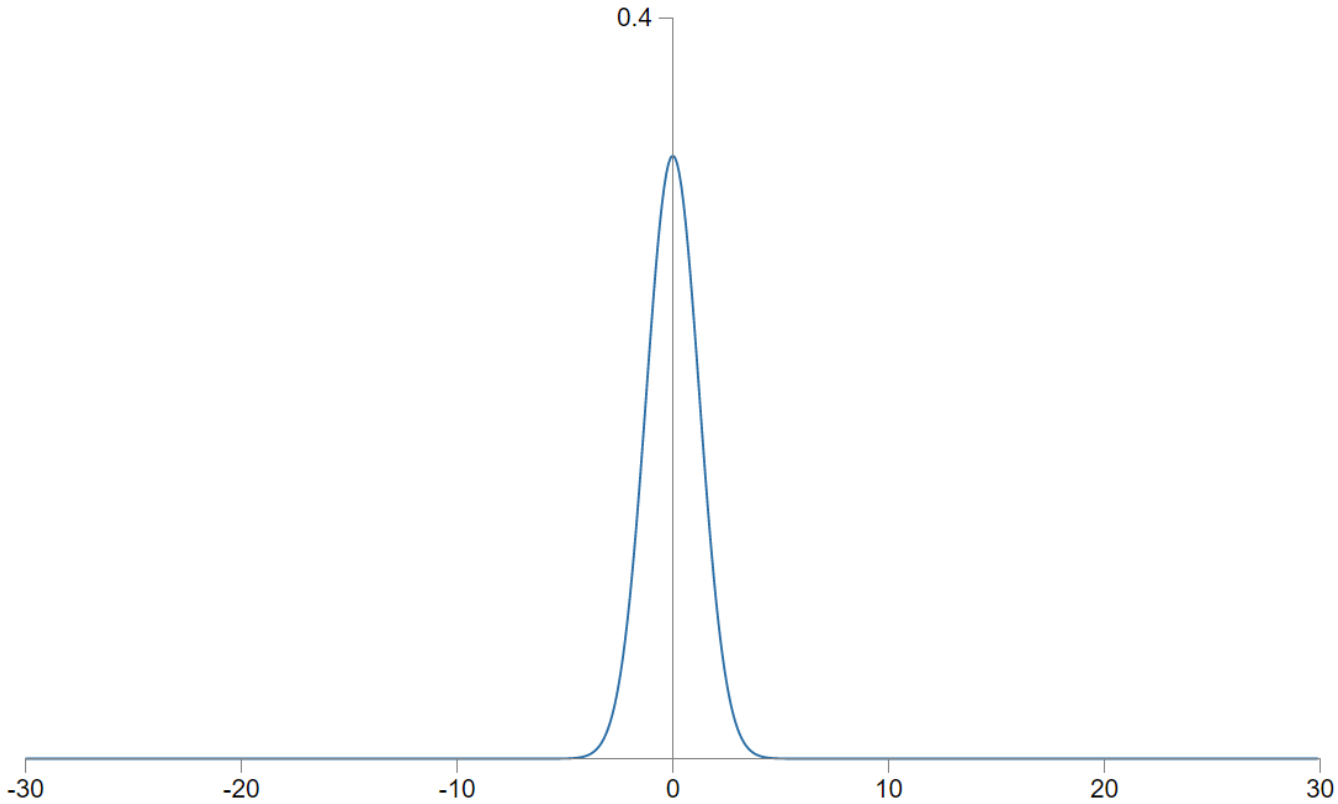


Abbildung:

Problem des verschwindenden Gradienten

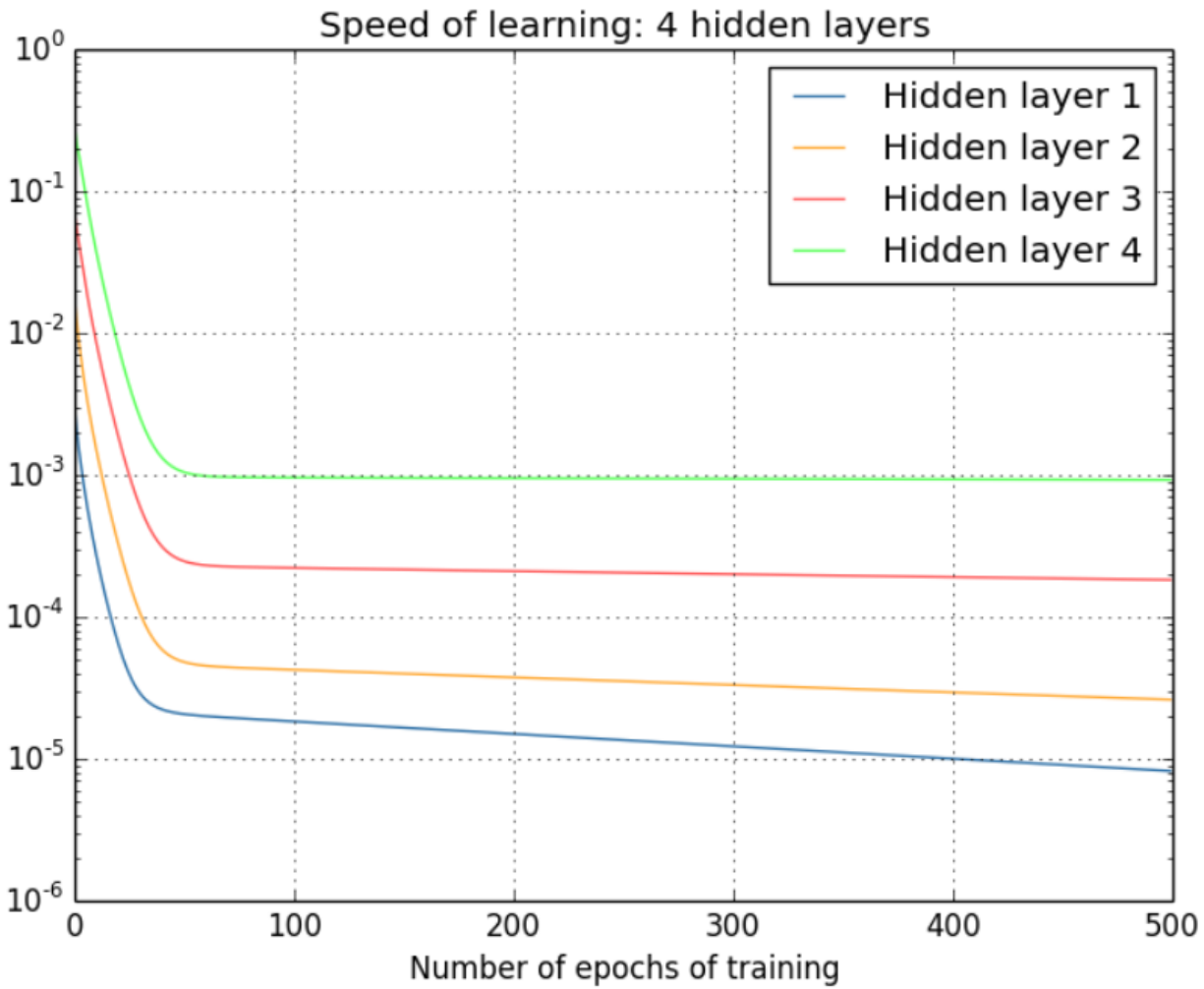


Abbildung:

Problem des verschwindenden Gradienten

$$\frac{\partial C}{\partial b_1} = \sigma'(z_1) \times w_2 \times \sigma'(z_2) \times w_3 \times \sigma'(z_3) \times w_4 \times \sigma'(z_4) \times \frac{\partial C}{\partial a_4}$$

Abbildung:

- Gradient in den frühen Schichten ist Produkt der Terme aus den späteren Schichten
  $\rightarrow$ unstabile Situation
- besser: verwende ReLU statt Sigmoid-Funktion

Zusammenfassung

- Neuronale Netze sind vor allem für Aufgaben gut geeignet, die vom Menschen intuitiv gelöst werden
  z.B.: Spracherkennung, Gesichtserkennung...
- Deep Learning nutzt eine Reihe hierarchischer Schichten, um die Eingabedaten interpretieren zu können
- viele Richtlinien bei der Verwendung neuronaler Netze basieren nur auf empirischen Erkenntnissen

$\rightarrow$ weitere Forschung nötig

Vielen Dank für die Aufmerksamkeit