

Künstliche Intelligenz mit neuronalen Netzen

Michael Hartmann

Kaffeeseminar

22. Juni 2016



"Jede hinreichend fortgeschrittene Technologie ist von Magie nicht mehr zu unterscheiden."

Arthur C. Clarke

Überblick

- ① Entwicklungen
- ② Erkennen von handgeschriebenen Ziffern
- ③ Ausblick

Entwicklungen: Was war



Windows Vista™

Video: Microsoft voice recognition

Entwicklungen: Was ist



Video: Siri

Entwicklungen: Was ist



Video: Googles selbstfahrende Autos

Entwicklungen: Was ist

Google algorithm busts CAPTCHA with 99.8 percent accuracy

Google engineers have defeated CAPTCHA thanks to a Street View algorithm designed to decipher blurry street addresses.



By [Liam Tung](#) | April 17, 2014 -- 12:37 GMT (13:37 BST) | Topic: [Security](#)

Google algorithm busts CAPTCHA

Entwicklungen: Was war



Entwicklungen: Was ist



Ehre wem Ehre gebührt

Neural Networks and Deep Learning

von

Michael Nielsen



Erkennen von Ziffern

504192

Erkennen von Ziffern

504192

Schwierigkeit für

- Menschen: extrem einfach
- (klassische) Algorithmen: extrem kompliziert

Erkennen von Ziffern

504192

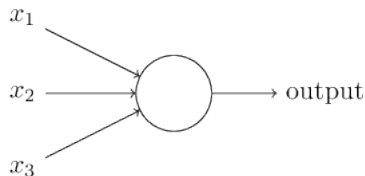
Schwierigkeit für

- Menschen: extrem einfach
- (klassische) Algorithmen: extrem kompliziert

Idee:

Computer soll selbstständig lernen das Problem zu lösen

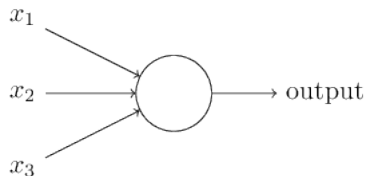
Perceptron



- mehrere binäre Eingänge x_1, x_2, \dots
- ein binärer Ausgang: output
- Gewichte w_1, w_2, \dots
- Schwellwert: threshold

$$\text{output} = \begin{cases} 0 & \text{if } \sum_j w_j x_j \leq \text{threshold} \\ 1 & \text{if } \sum_j w_j x_j > \text{threshold} \end{cases}$$

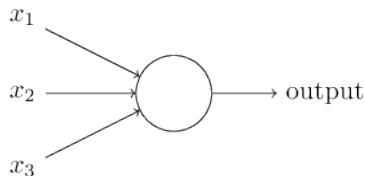
Perceptron



- mehrere binäre Eingänge x_1, x_2, \dots
- ein binärer Ausgang: output
- Gewichte w_1, w_2, \dots
- Schwellwert: threshold

$$\text{output} = \begin{cases} 0 & \text{if } \mathbf{w} \cdot \mathbf{x} \leq \text{threshold} \\ 1 & \text{if } \mathbf{w} \cdot \mathbf{x} > \text{threshold} \end{cases}$$

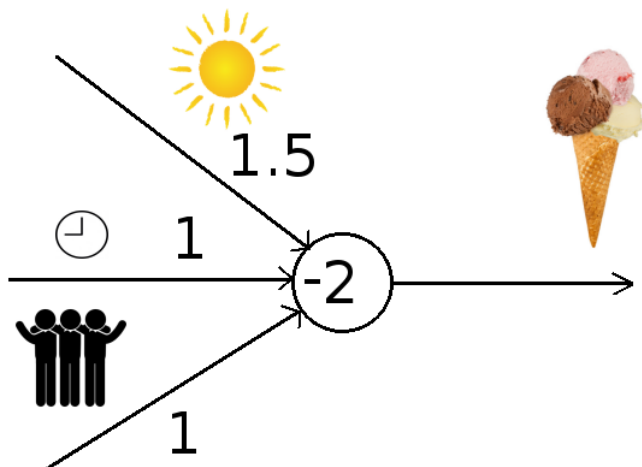
Perceptron



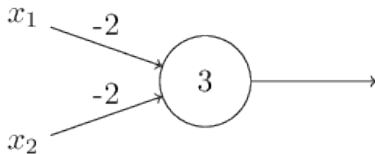
- mehrere binäre Eingänge x_1, x_2, \dots
- ein binärer Ausgang: output
- Gewichte w_1, w_2, \dots
- Schwellwert: threshold

$$\text{output} = \begin{cases} 0 & \text{if } \mathbf{w} \cdot \mathbf{x} + b \leq 0 \\ 1 & \text{if } \mathbf{w} \cdot \mathbf{x} + b > 0 \end{cases}$$

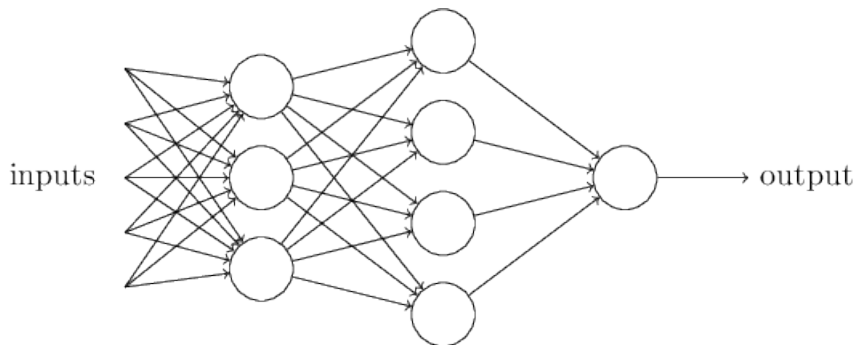
Perceptron: Beispiel



Perceptrons sind universell



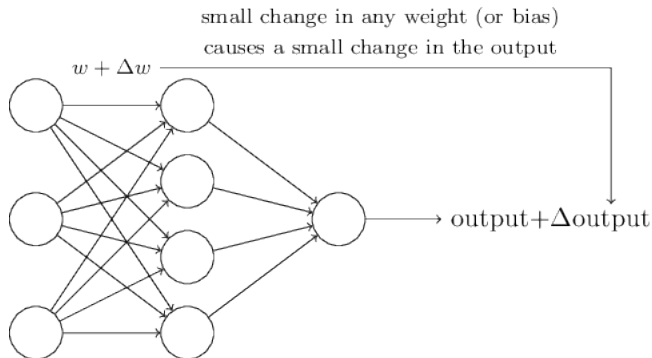
Netzwerk aus Perceptronen



Maschinelles lernen

Grundidee

- 1 Neuronales Netz erstellen
- 2 Gewichte und Schwellwerte (bias) durch Training anpassen



Sigmoids

Problem:

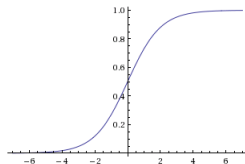
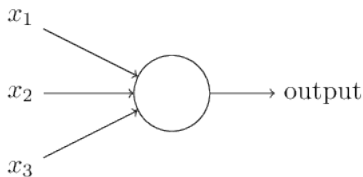
Kleine Änderung der Gewichte/Schwellwerte kann komplettes Verhalten ändern

Sigmoids

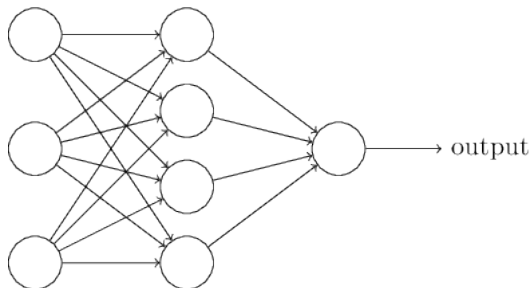
Lösung: Sigmoids

- kontinuierliche Eingangssignale $x_j \in [0, 1]$
- Gewichte w_j , Schwellwert b
- Ausgabe: $\sigma(\mathbf{w} \cdot \mathbf{x} + b)$ mit $\sigma(z) = \frac{1}{1+\exp(-z)}$

$$\text{output} = \frac{1}{1 + \exp(-\mathbf{w} \cdot \mathbf{x} - b)}$$

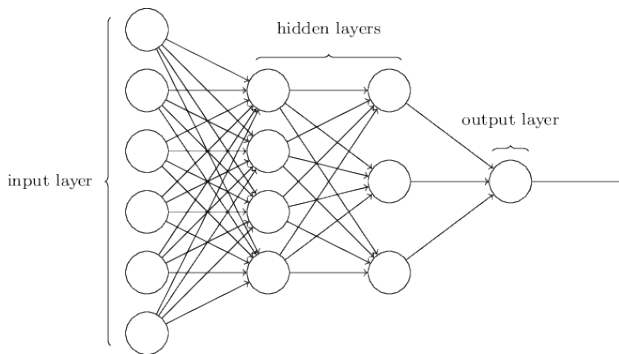


Netzwerk



- linke Schicht: input layer
- rechte Schicht: output layer
- mittlere Schichten: hidden layers

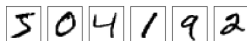
Netzwerk



- linke Schicht: input layer
- rechte Schicht: output layer
- mittlere Schichten: hidden layers

Erkennen von Ziffern

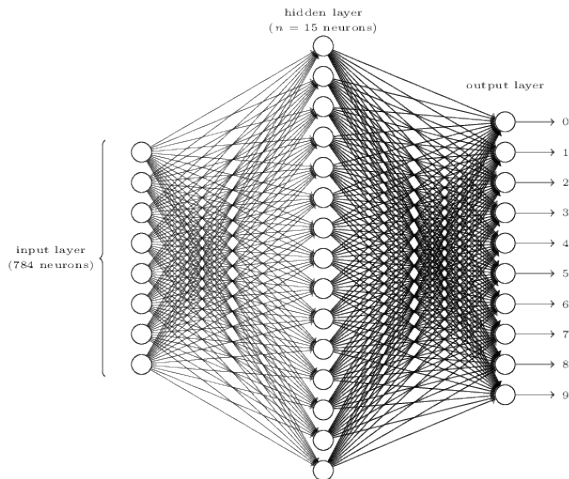
- Segmentation problem:



gar nicht so leicht zu lösen...

- Erkennen der Ziffer
Dieses Problem lösen wir. :)

Netzwerk



- 28^2 Eingänge, $\mathbf{x}_{\text{in}} \in \mathbb{R}^{784 \times 1}$
- 10 Ausgänge, $\mathbf{x}_{\text{out}} \in \mathbb{R}^{10 \times 1}$
- $f : \mathbb{R}^{784 \times 1} \rightarrow \mathbb{R}^{10 \times 1}$

Trainingsdaten

MNIST: 60 000 + 10 000 handgeschriebene Ziffern



Kostenfunktion

$$C(\mathbf{w}_1, \dots, \mathbf{w}_n, b_1, \dots, b_n) = \frac{1}{2N} \sum_{\mathbf{x} \in \mathcal{T}} \|f(\mathbf{x}) - \mathbf{a}\|$$

- \mathcal{T} : Menge aller (Eingangs-)Trainingsdaten, $N = |\mathcal{T}|$
- \mathbf{a} : erwartetes (richtiges) Ergebnis für \mathbf{x}
- f : Wirkung des neuronalen Netzes

Kostenfunktion

$$C(\mathbf{p}) = \frac{1}{2N} \sum_{\mathbf{x} \in \mathcal{T}} \|f(\mathbf{x}) - \mathbf{a}\|$$

- \mathcal{T} : Menge aller (Eingangs-)Trainingsdaten, $N = |\mathcal{T}|$
- \mathbf{a} : erwartetes (richtiges) Ergebnis für \mathbf{x}
- f : Wirkung des neuronalen Netzes
- $\mathbf{p} = (\mathbf{w}_1, \dots, \mathbf{w}_n, b_1, \dots, b_n)$

Kostenfunktion

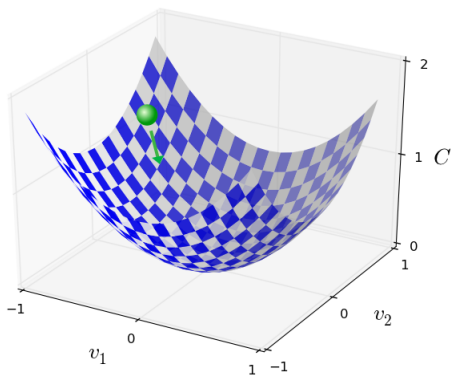
$$C(\mathbf{p}) = \frac{1}{2N} \sum_{\mathbf{x} \in \mathcal{T}} \|f(\mathbf{x}) - \mathbf{a}\|$$

- \mathcal{T} : Menge aller (Eingangs-)Trainingsdaten, $N = |\mathcal{T}|$
- \mathbf{a} : erwartetes (richtiges) Ergebnis für \mathbf{x}
- f : Wirkung des neuronalen Netzes

\Rightarrow Minimiere $C(\mathbf{p})$!

Gradientenabstieg

$$\mathbf{p}_{j+1} = \mathbf{p}_j - \eta \nabla C(\mathbf{p}_j)$$



Demo

Demo

online: <http://myselfph.de/neuralNet.html>

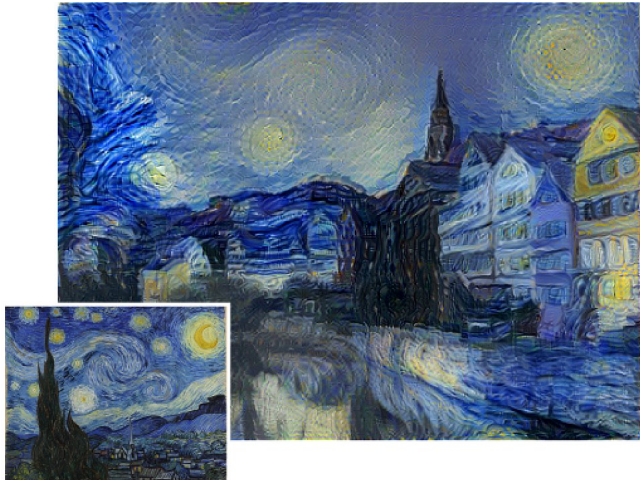
Probleme

- Lernprozess konvergiert gegen lokales (nicht globales!) Minimum
- Anpassung des Neuronalen Netzes auf Muster in den Trainingsdaten (z.B. hell/dunkel)
- Überanpassung (Auswendig lernen)
- Qualität der Trainingsdaten (z.B. 1)
- Daten müssen sinnvoll aufbereitet sein (z.B. Bild richtig gedreht und zugeschnitten)
- Netz muss für Problem geeignet sein

Kopieren von Stilen



Kopieren von Stilen



Kopieren von Stilen



Video: Picasso Odyssey

Deep dreaming



deepdreamgenerator.com

Deep dreaming



deepdreamgenerator.com

Deep dreaming



deepdreamgenerator.com

Vielen Dank für die Aufmerksamkeit :)

