Neuronale Netze und Algorithmen

Modellierung dynamischer und adaptiver Systeme
WS 2017/18

Njomza Avdijaj

18. Januar 2018

Software and Systems Engineering

Agenda

1. Motivation
2. Grundlagen
3. Netztypen
4. Lernstrategien
5. Deep Learning
6. Anwendungen
7. Zusammenfassung
8. Quellen

Motivation

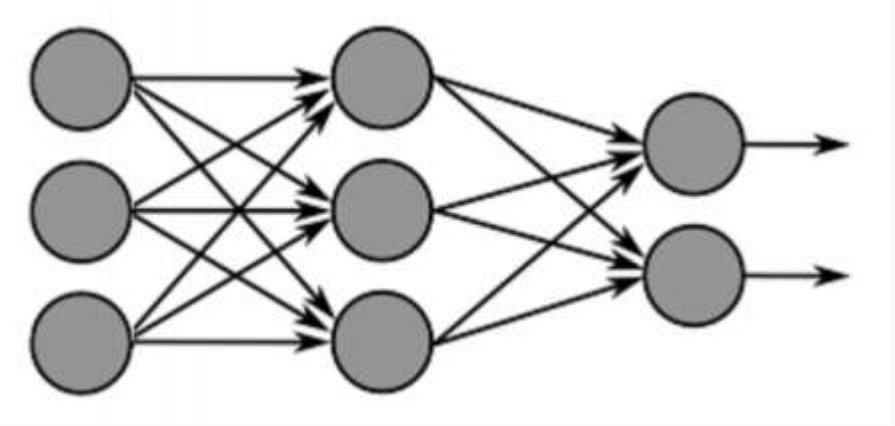
Motivation

- Berechnungsmodell mit biologischem Vorbild
- Interdisziplinärer Forschungsgegenstand der Informatik, Mathematik, Biologie und Physik
- Pionierleistungen der Forscher Warren McCulloch und Walter Pitts im Jahr 1943

Grundlagen

Neuronen und Verbindungen

Gerichteter Graph $G = (U, C)$
$U \rightarrow$ Units $\rightarrow$ Neuronen
$C \rightarrow$ Connections $\rightarrow$ Verbindungen



http://web.utk.edu/~wfeng1/spark/fnn.html

Schichten

- **Input Layer**:
  Neuronen können von der Außenwelt Informationen empfangen
- **Output Layer**:
  Neuronen können Informationen an die Außenwelt weitergeben
- **Hidden Layer**:
  - Neuronen ohne Kontakt zur Außenwelt
  - Interne Repräsentation

Gewichte

Jede Kante $(v, u) \in C$ trägt ein **Gewicht** w_{uv}
$\rightarrow$ Stärke der Verbindung zwischen dem sendenden Neuron v und dem empfangenden Neuron u

Positives Gewicht $(w_{uv} > 0) \rightarrow$ exzitatorischer, erregender Einfluss

Negatives Gewicht $(w_{uv} < 0) \rightarrow$ inhibitorischer, hemmender Einfluss

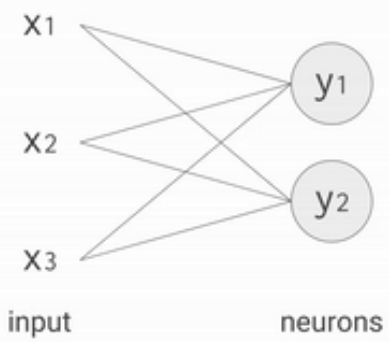
Gewicht von Null $(w_{uv} = 0) \rightarrow$ kein Einfluss

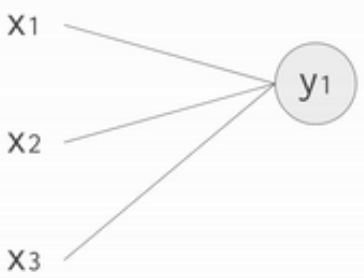
Definition

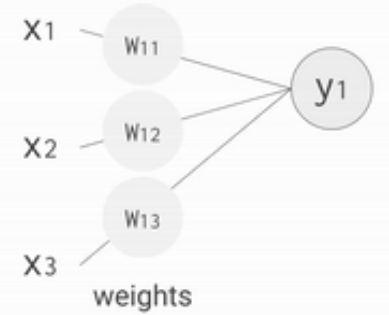
Das **Wissen** eines neuronalen Netzes ist in seinen Gewichten gespeichert.

Lernen wird als Gewichtsveränderung zwischen den Neuronen definiert.

Funktionsweise von Neuronen







$$y_1 = w_{11}x_1 + w_{12}x_2 + w_{13}x_3$$

https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu

Funktionsweise von Neuronen

https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu

Zusammenfassung:

1. Berechnung der einzelnen Inputs
2. Berechnung des Netzinputs mithilfe der einzelnen Inputwerte
3. Zuordnung des Netzinputs zu einem Aktivitästlevel

Funktionen

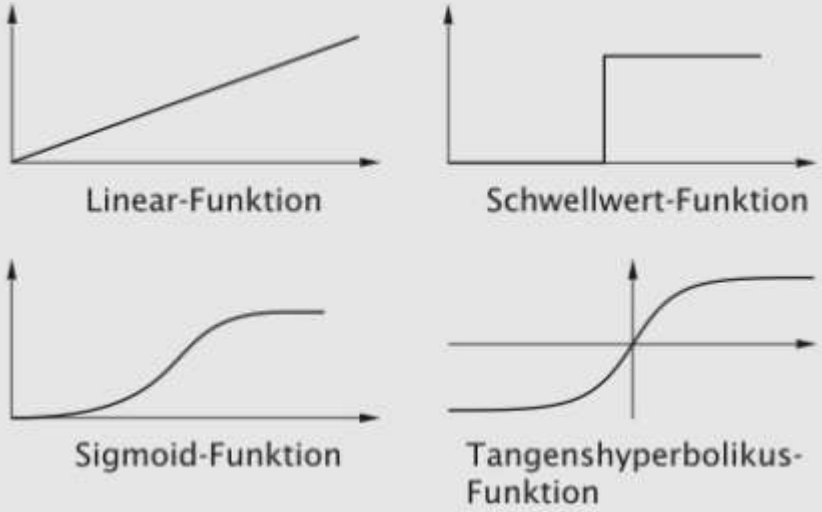
Input

$input_{uv} = a_v \cdot w_{uv}$

Netzinput

$$netinput_u = \sum_v input_{uv} = \sum_v a_v \cdot w_{uv}$$

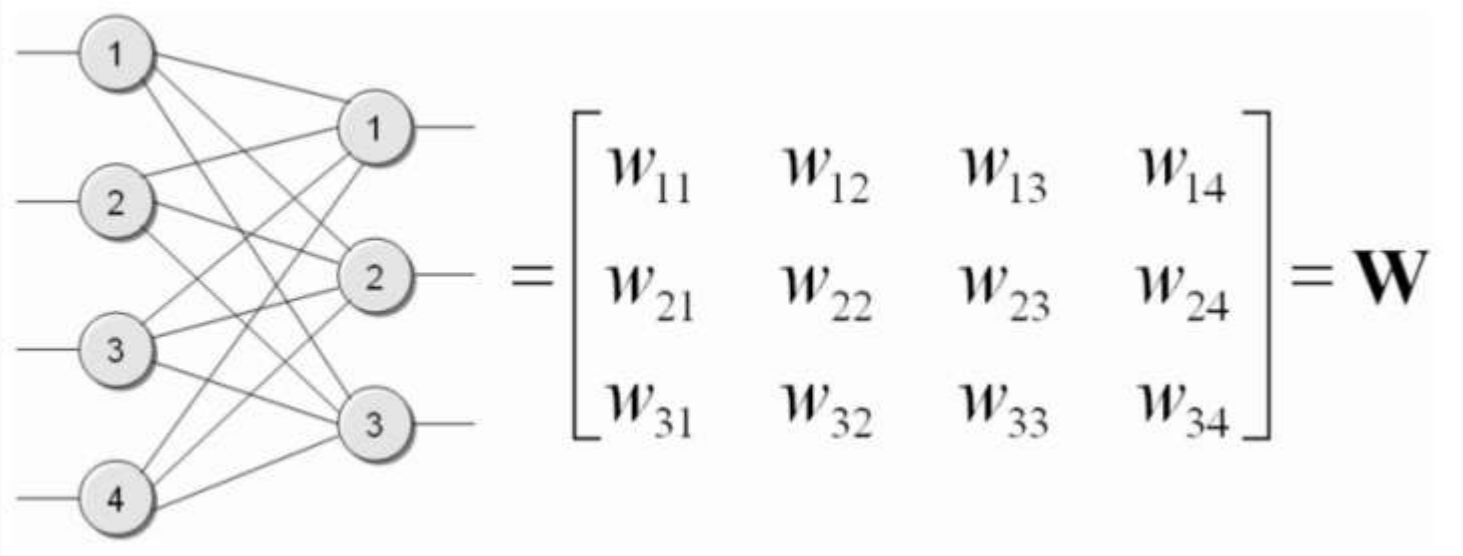
Aktivitätsfunktion



http://www.electroyou.it/nunziox/wiki/un-esempio-di-rete-neurale-a-percettore-multistrato

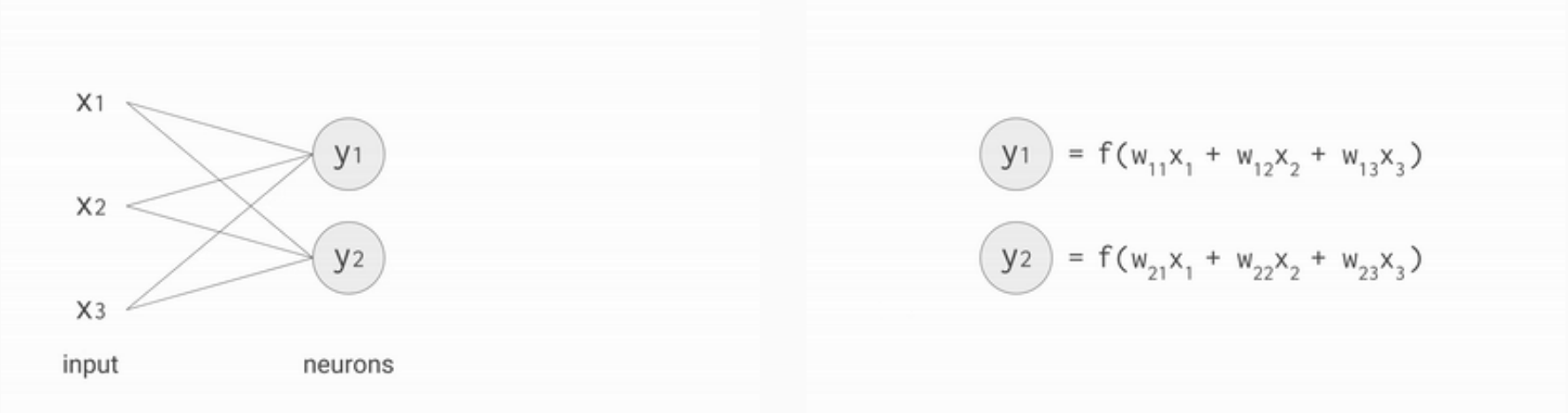
Matrizendarstellung

Gewichtsmatrix



http://www.neuronalesnetz.de/matrix.html

Berechnung



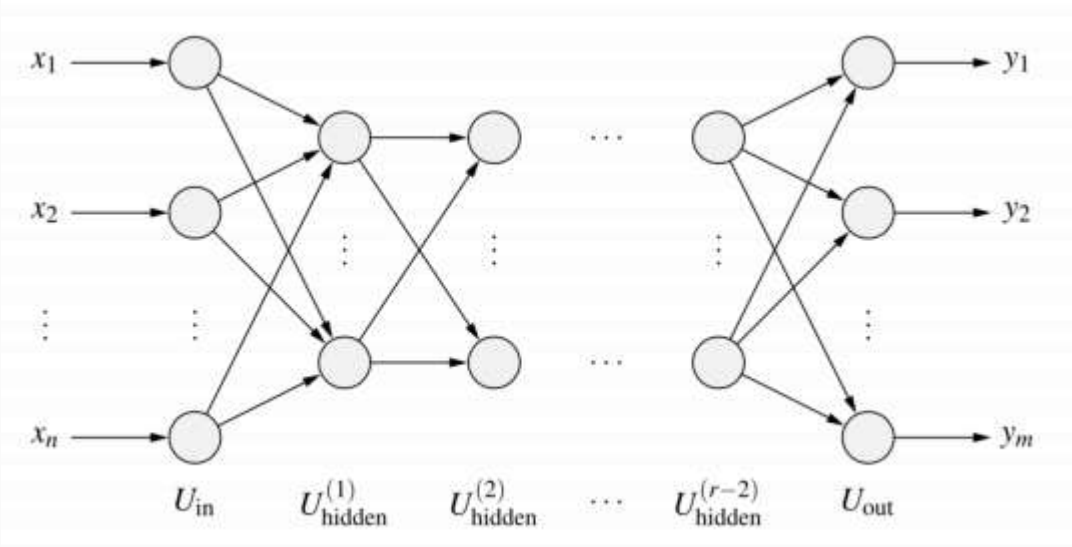
https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu

Berechnung durch Multiplikation der **Gewichtsmatrix** mit dem **Inputvektor**:

$$\begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} w_{11}x_1 + w_{12}x_2 + w_{13}x_3 \\ w_{21}x_1 + w_{22}x_2 + w_{23}x_3 \end{pmatrix}$$

Netztypen

Feedforward Neural Networks



Kruse, R., Borgelt, C., Klawonn, F., Moewes, C., Steinbrecher, M., and Held, P. (2013). Computational Intelligence: A Methodological Introduction. Springer Publishing Company, Incorporated.

Definition

Ein **Multilayer Perceptron** (R-Layer Perceptron) ist ein gerichteter Graph $G = (U, C)$, für den Folgendes gilt:

- eine Eingabe- und eine Ausgabeschicht
- $r - 2$ verdeckte Schichten
- Verbindungen nur zwischen aufeinanderfolgenden Schichten

Recurrent Neural Networks

Definition

Ein **Recurrent Neural Network** ist dadurch gekennzeichnet, dass **Rückkopplungen** von Neuronen einer Schicht zu anderen Neuronen derselben oder einer vorangegangenen Schicht existieren.

Rückkopplungsarten

- Direkte Rückkopplungen
- Indirekte Rückkopplungen
- Seitliche Rückkopplungen
- Vollständige Verbindungen

Lernstrategien

Supervised vs Unsupervised

Supervised Learning $\rightarrow$ *labeled training data*

- Vorgabe der gewünschten Ausgabe zu verschiedenen Eingabemustern
- Vergleich der tatsächlichen mit der vorgegebenen Ausgabe
- Anpassung der Gewichte zur Minimierung des Fehlers

$\Rightarrow$ *Backpropagation*

Unsupervised Learning $\rightarrow$ *unlabeled training data*

- Vorgabe der zu lernenden Eingabemuster ohne Zielwert
- Lernen durch Mustererkennung und Vergleich der Eingabedaten
- Gewichtsanpassung zur Erzeugung ähnlicher Ausgaben für ähnliche Eingaben

Backpropagation

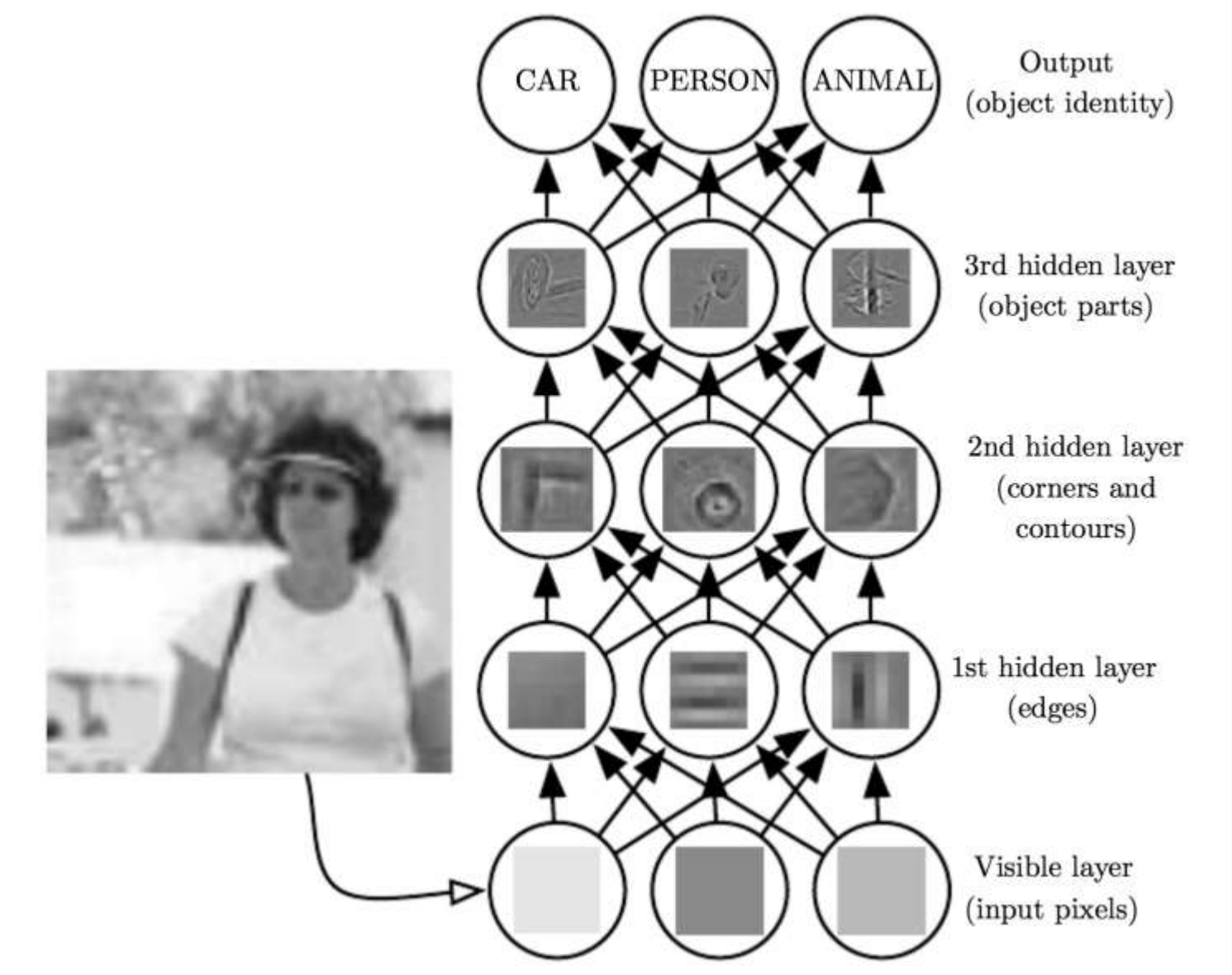
1. **Forward-Pass:**
   - Berechnung der Netzausgabe für jedes Eingabemuster
2. **Fehlerbestimmung:**
   - Vergleich der tatsächlichen mit der gewünschten Ausgabe durch eine Fehlerfunktion
   - $E = \sum_{l \subset L} \sum_{v \subset U_{out}} e_v^{(l)} = \sum_{l \subset L} \sum_{v \subset U_{out}} (t_v^{(l)} - o_v^{(l)})^2$
3. **Backward-Pass:**
   - Fehlerrückführung von der Ausgabe- bis hin zur Eingabeschicht
   - Anpassung der Gewichte abhängig von ihrem Einfluss auf den Fehler
   - Minimierung des Gesamtfehlers

Deep Learning

Deep Learning

- Klasse von Optimierungsmethoden künstlicher neuronaler Netze
- Netze mit einer tiefen inneren Struktur (zahlreiche Hidden Layer)
- Lösung von Problemen, die nicht formal beschrieben werden können
- Intuitives Wissen
- Lernen aus Erfahrung
- Hierarchie von Konzepten
- Steigerung des Abstraktionslevels von Schicht zu Schicht

Beispiel



http://www.deeplearningbook.org/contents/intro.html

Anwendungen

Anwendungsgebiete

Anwendungen, bei denen kein oder nur geringes explizites Wissen über das zu lösende Problem vorliegt:

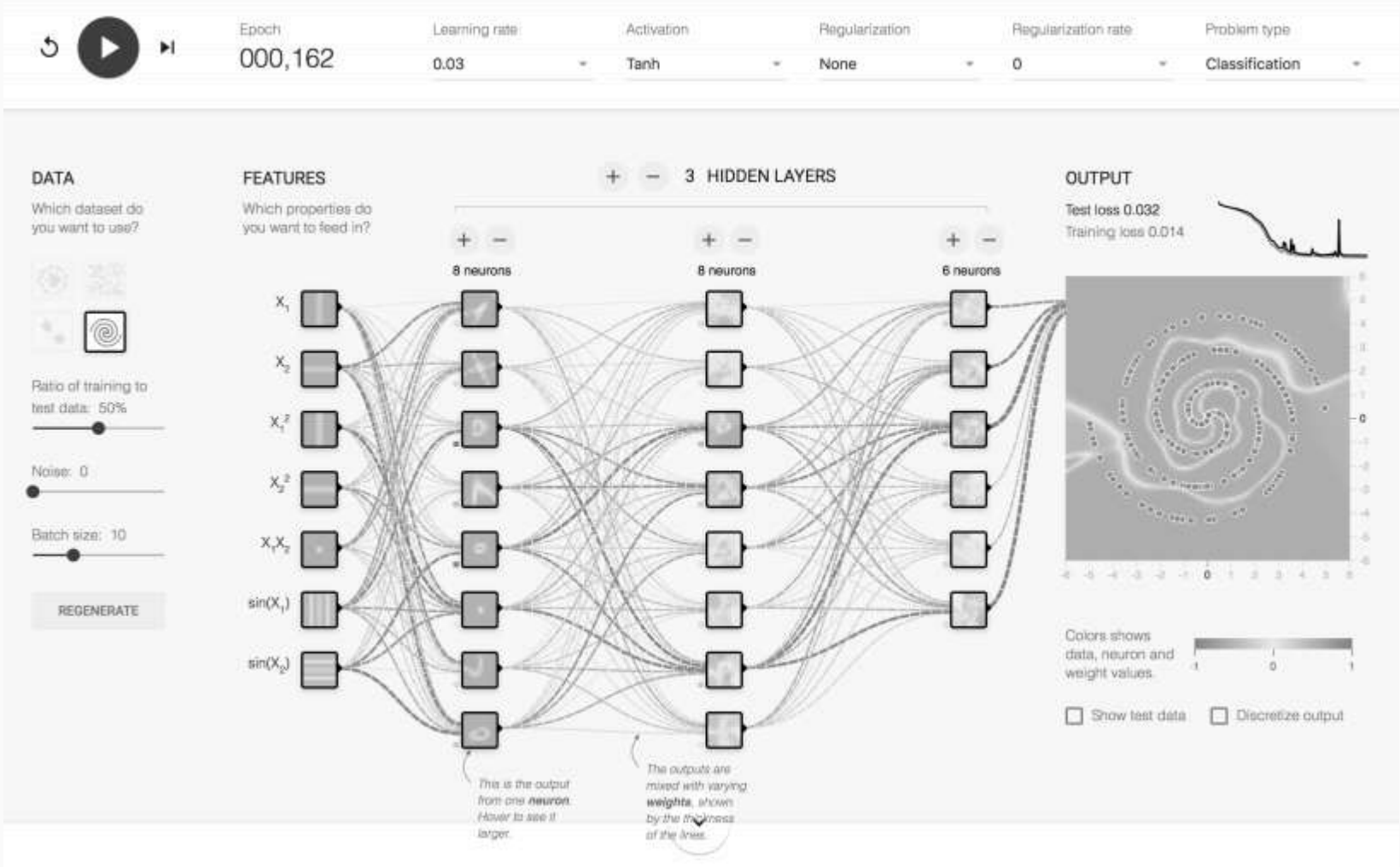
- Bildverarbeitung und Mustererkennung
  - Bilderkennung
  - Texterkennung
  - Schrifterkennung
  - Gesichtserkennung
  - Spracherkennung
  - Maschinenübersetzung
- Klassifikation
- Optimierung
- Medizinische Diagnostik
- Frühwarnsysteme
- Robotik
- Virtuelle Agenten

Hilfsmittel

- Verschiedene Bibliotheken
  - TensorFlow (https://www.tensorflow.org)
  - Keras (https://keras.io)
- Tools
  - Neural Network Toolbox (https://de.mathworks.com/products/neural-network.html)
  - Stuttgarter Neural Network Simulator (http://www.ra.cs.uni-tuebingen.de/SNNS/)
  - Simbrain (http://simbrain.net)

Beispiel

TensorFlow Playground



http://playground.tensorflow.org

Zusammenfassung

Zusammenfassung und Diskussion

Zusammenfassung

Neuronale Netze bekommen eine Struktur und ein Lernverfahren vorgegeben und müssen dann in einem Lernprozess selber die richtige Konfiguration zur Lösung eines Problems finden.

Probleme

- Sammlung oder manuelle Erzeugung von Trainings- und Testdaten notwendig
- Großer Rechenaufwand
- Möglicherweise keine optimale Lösung (lokales statt globales Optimum)
- Overfitting (keine Verallgemeinerung auf neue Daten möglich)
- Blackbox bei umfangreicher innerer Struktur
- Fragwürdige biologische Plausibilität

Quellen

Literatur

Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT Press. Online erhältlich unter http://www.deeplearningbook.org; 15. Januar 2018.

Kruse, R., Borgelt, C., Klawonn, F., Moewes, C., Steinbrecher, M., and Held, P. (2013). Computational Intelligence: A Methodological Introduction. Springer Publishing Company, Incorporated.

Nielsen, M. A. (2015). Neural Networks and Deep Learning. Online erhältlich unter http://neuralnetworksanddeeplearning.com; 8. Januar 2018.

Rey, G. and Wender, K. (2011). Neuronale Netze: eine Einführung in die Grundlagen, Anwendungen und Datenauswertung. Aus dem Programm Huber:Psychologie-Lehrbuch. Huber. Online erhältlich unter http://www.neuronalesnetz.de; 14. Januar 2018.

Rojas, R. (1993). Theorie der neuronalen Netze: eine systematische Einführung. Springer Lehrbuch. Springer.

Sato, K. (2016). Understanding neural networks with TensorFlow Playground. Online erhältlich unter https://cloud.google.com/blog/big-data/2016/07/understanding-neural-networks-with-tensorflow-playground; 15. Januar 2018.

Sato, K., Young, C., and Patterson, D. (2017). An in-depth look at Google?s first Tensor Processing Unit (TPU). Online erhältlich unter https://cloud.google.com/blog/big-data/2017/05/an-in-depth-look-at-googles-first-tensor-processing-unit-tpu; 4. Januar 2018.

Smilkov, D. and Carter, S. TensorFlow Playground. Online erhältlich unter

http://playground.tensorflow.org/; 15. Januar 2018.

Fragen?