

# Eine kleine Einführung in Deep Learning

Sarah Böning

DLR Jena, Institut für Datenwissenschaften

Sarah.Boening@dlr.de

T-Systems MMS Jena

11.09.2020



Wissen für Morgen



# Agenda

- Vorstellung DLR
- Quiz
- Deep Learning – Theorie
- Deep Learning – Praxis
- Kleine Demos
- Hilfreiche Links



# Das DLR

- Forschungsbereiche
  - Luftfahrt
  - Raumfahrtforschung & -technologie
  - Verkehr
  - Energie
  - Sicherheit
  - Digitalisierung

Circa 9.000 Mitarbeiterinnen und Mitarbeiter arbeiten in 49 Instituten und Einrichtungen an 28 Standorten.

Auslandsbüros in Brüssel, Paris, Tokio und Washington.



# Das DLR in Jena – Institut für Datenwissenschaften

- Datenmanagement und Analyse
  - Datenmanagementtechnologien
  - Klimainformatik
  - Visuelle Analyse
  - Maschinelles Lernen
- Sichere Digitale Systeme
  - Digitale Produktionsplattformen
  - Sichere Softwaretechnik
- Bürgerwissenschaften
  - Bürgerwissenschaften





# Über mich

- 2014 – 2017: TU Clausthal, Informatik, B.Sc.
  - *Entwicklung einer Webservice-Schnittstelle zur Bereitstellung von Ergebnissen aus der Kombination manueller und automatisierter Textanalyseverfahren*
- 2018 – 2020: Bauhaus-Universität Weimar, Computer Science for Digital Media, M.Sc.
  - *Assessing the State-of-the-Art for Source-Code Suggestions*
- Seit 1. Juli 2020: DLR Jena, Arbeitsgruppe Digitale Produktionsplattformen
- Forschungsinteressen:
  - Natural Language Processing (NLP)
  - Machine & Deep Learning
  - Software Engineering



## Ein kleines Quiz: Mensch oder Maschine?







<https://www.themoviedb.org/person/928572-scott-eastwood?language=de-DE> , aufgerufen 07.09.2020



Karras, T., Aila, T., Laine, S., and Lehtinen, J. 2017. "Progressive Growing of GANs for Improved Quality, Stability, and Variation"







[https://www2.hm.com/de\\_de/productpage.0905945001.html](https://www2.hm.com/de_de/productpage.0905945001.html) , aufgerufen 07.09.2020



<https://datagrid.co.jp/all/release/386/> , aufgerufen 07.09.2020



```
1 public int compare(Object o1, Object o2) {  
2     if (o1 instanceof ICompletionProposal) {  
3         return ((ICompletionProposal) o1).compareTo(o2);  
4     }  
5     return -1;  
6 }
```



```
1 private ArrayList<DashboardStep> endStep(ArrayList<DashboardStep> steps, Feature feature){  
2     if (AuthToken.getMember().equals(feature.getMember())){  
3         DashboardStep end = new DashboardStep(Context.tr("End"), StepState.COMPLETE, Context.tr("You created this feature"));  
4         steps.add(end);  
5     }  
6     return end;  
7 }
```



```
1 public int compare(Object o1, Object o2) {  
2     if (o1 instanceof ICompletionProposal) {  
3         return ((ICompletionProposal) o1).compareTo(o2);  
4     }  
5     return -1;  
6 }
```

```
1 private ArrayList<DashboardStep> endStep(ArrayList<DashboardStep> steps, Feature feature){  
2     if (AuthToken.getMember().equals(feature.getMember())){  
3         DashboardStep end = new DashboardStep(Context.tr("End"), StepState.COMPLETE, Context.tr("You created this feature"));  
4         steps.add(end);  
5     }  
6     return end;  
7 }
```



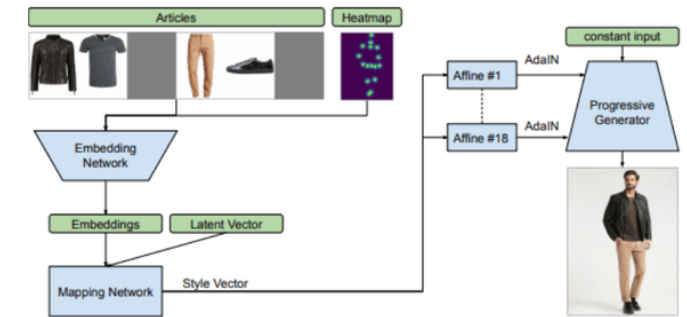
S. Böning, Assessing the State-of-the-Art for Source-Code Suggestions, 2020.





# Für was kann man ML einsetzen?

- Jede Aufgabe (die lernbar ist)
- Keine allgemeine AI (bis jetzt)
- Keine Lotto-Zahlen vorhersagen
- Grundlage: ausreichende Datenmenge, die ein erkennbares Muster hat



<https://syncedreview.com/2019/08/29/ai-creates-fashion-models-with-custom-outfits-and-poses/> aufgerufen 07.09.2020



<https://www.analyticssteps.com/blogs/weather-forecasting-how-do-big-data-analytics-magnify-it> aufgerufen 07.09.2020

SYSTEM PROMPT (HUMAN-WRITTEN) *In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*

MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)

The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

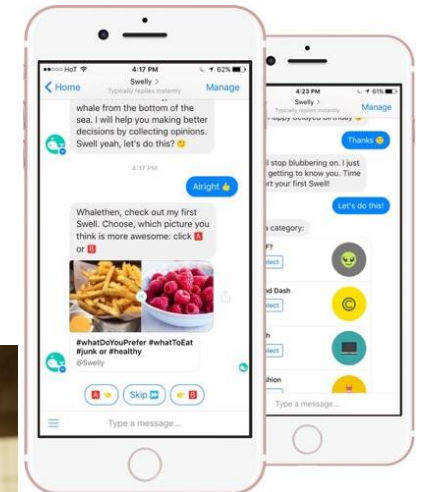
Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

<https://openai.com/blog/better-language-models/> aufgerufen 07.09.2020



<https://medium.com/theblock1/eventum-alpha-emotion-recognition-meets-blockchain-5acd8fa95c8>, aufgerufen 07.09.2020



<https://botsociety.io/blog/2018/03/chatbot-examples/>, aufgerufen 07.09.2020

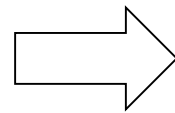
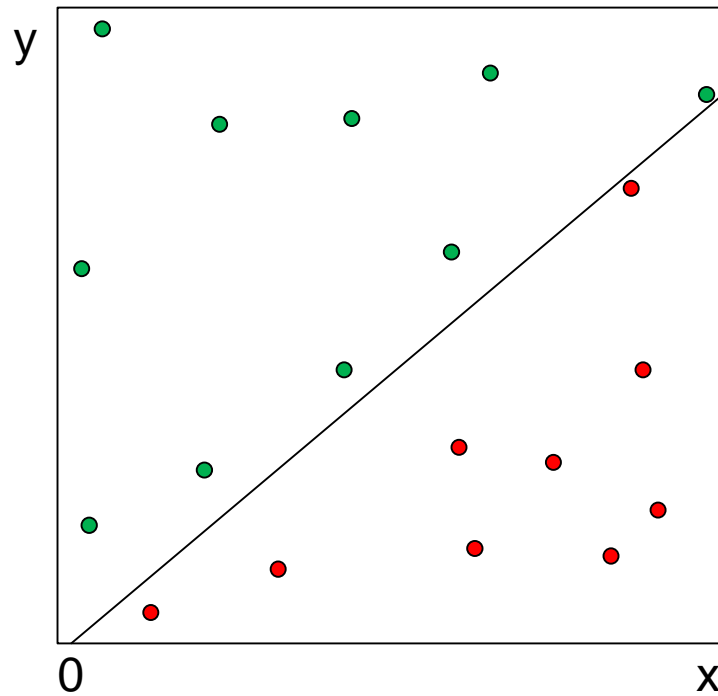
# Deep Learning – Die Theorie



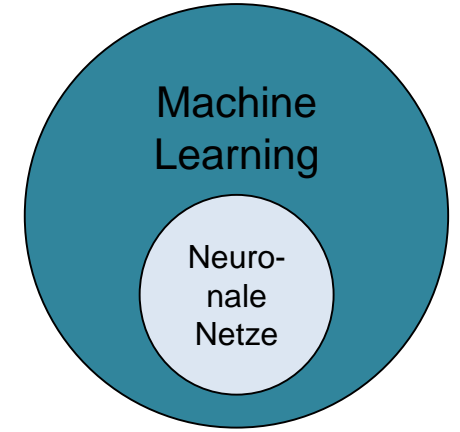
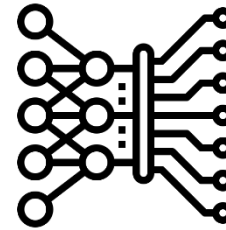
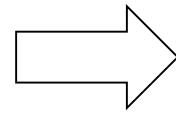
Wissen für Morgen



# Die Grundlagen von Machine Learning



$y(x)$

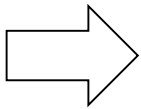


- Ziel: Funktion, die meine Daten beschreibt, aber auch neue Beispiele richtig erkennt
- Training: Suche nach den besten Parametern der Funktion

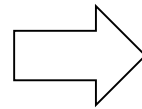


# Die Grundbegriffe von Machine Learning

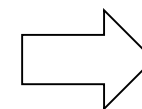
- Fehler: Differenz zwischen Output der ML-Funktion und dem realen Wert
- Backpropagation: Anpassen der Parameter mit Hilfe des Fehlers → Lernen! (nur bei NNs)



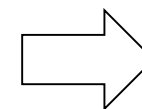
255	255	255	255
255	123	56	255
255	58	70	255
255	255	255	255



Netzwerk



[4]



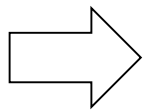
Flugzeug





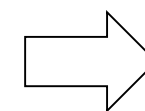
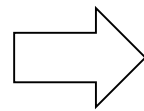
# Die Grundbegriffe von Machine Learning

- Fehler: Differenz zwischen Output der ML-Funktion und dem realen Wert
- Backpropagation: Anpassen der Parameter mit Hilfe des Fehlers → Lernen! (nur bei NNs)
- Input-Vektor: numerische Repräsentation der Daten
- Output-Vektor: (numerische) Vorhersage des Algorithmus' / Netzwerks

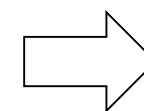


255	255	255	255
255	123	56	255
255	58	70	255
255	255	255	255

Input-Vektor



[1]



Katze

Output-Vektor

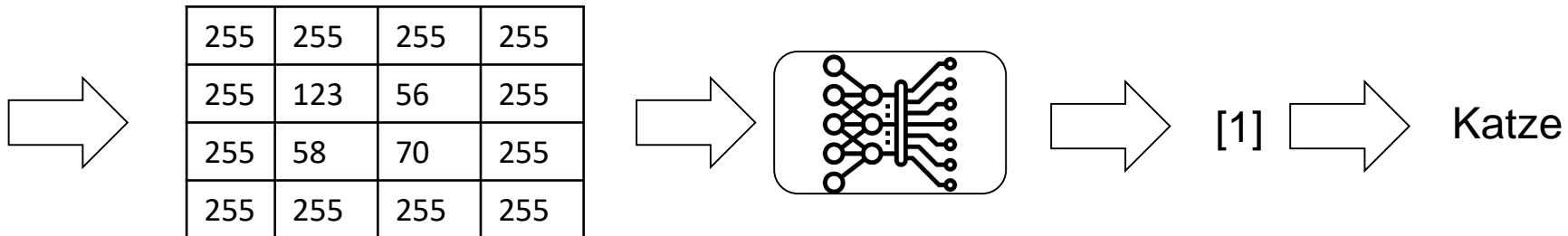


# Die Grundbegriffe von Machine Learning

- Fehler: Differenz zwischen Output der ML-Funktion und dem realen Wert
- Backpropagation: Anpassen der Parameter mit Hilfe des Fehlers → Lernen! (nur bei NNs)
- Input-Vektor: numerische Repräsentation der Daten
- Output-Vektor: (numerische) Vorhersage des Algorithmus' / Netzwerks
- Supervised Learning: Trainingsdaten haben vorgegebenen Output (z.B. Klassifikation)
- Unsupervised Learning: kein vorgegebener Output (z.B. Clustering)
- Deep Learning: Training von Neuronalen Netzwerken



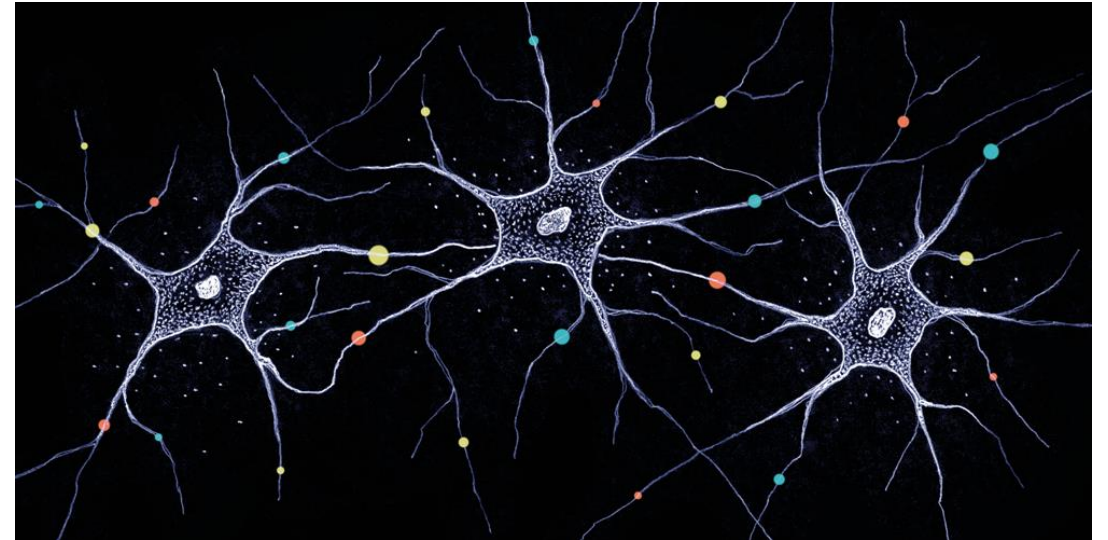
$y = 1$  (Katze)



# Der Biologische Hintergrund

- Neuronen tauschen Informationen aus
- Impuls wird nur weitergeben, wenn Schwellwert erreicht wird
- Oft mehrere Signale von benachbarten Neuronen nötig
- Lernen durch Abschwächen/Stärken bestimmter Verbindungen

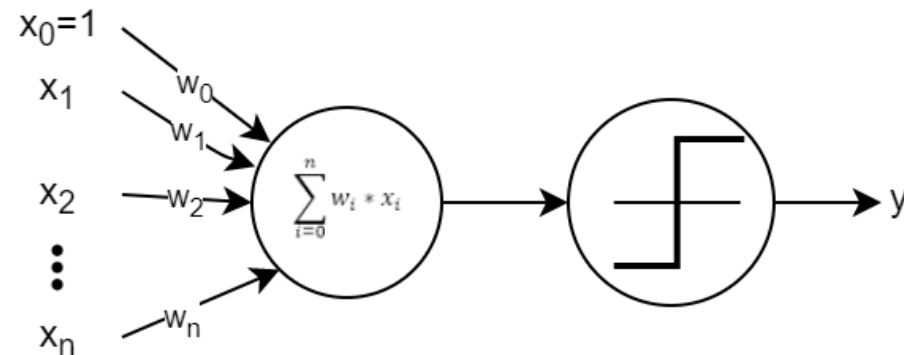
→ Grundidee aller künstlichen neuronalen Netze



<https://www.unibas.ch/de/Aktuell/News/Uni-Research/Vielfalt-im-Gehirn.html>,  
aufgerufen 07.09.2020

# Das Perceptron

- F. Rosenblatt, 1958
- Basiert auf Funktionsweise von einem Neuron
- Gewichtete Summe des Inputs
- Schwellenwert zur Aktivierung
- $$\text{output}(x) = \begin{cases} 1, & \text{wenn Schwellenwert erreicht} \\ -1, & \text{sonst} \end{cases}$$



- Lernen = finde beste Gewichte basierend auf den vorhandenen Daten

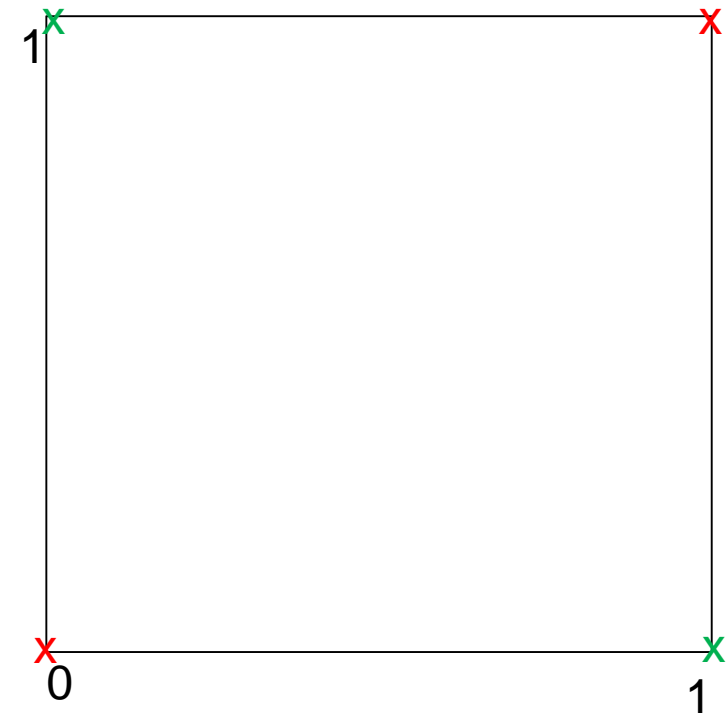




# Warum brauchen wir Deep Learning? – Das XOR Problem

- Problem bei der Klassifikation

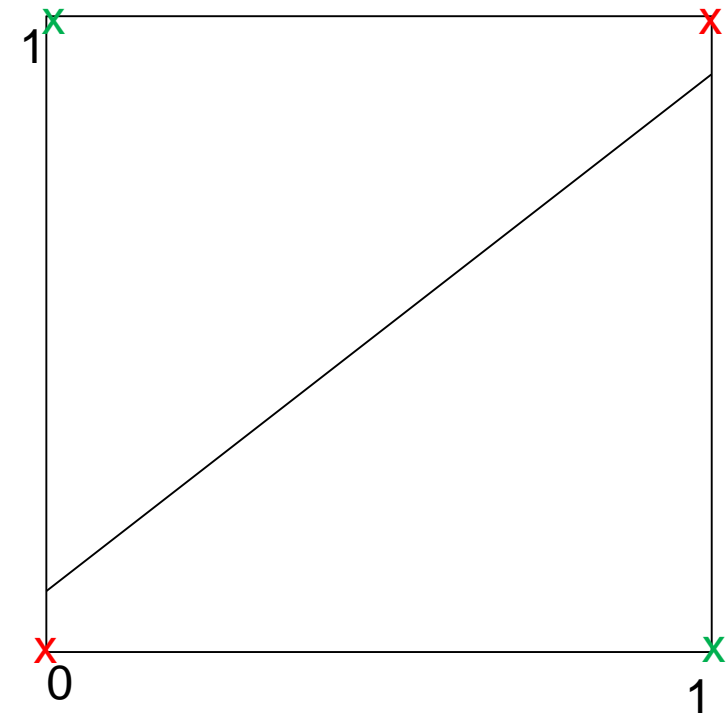
X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0



# Warum brauchen wir Deep Learning? – Das XOR Problem

- Problem bei der Klassifikation

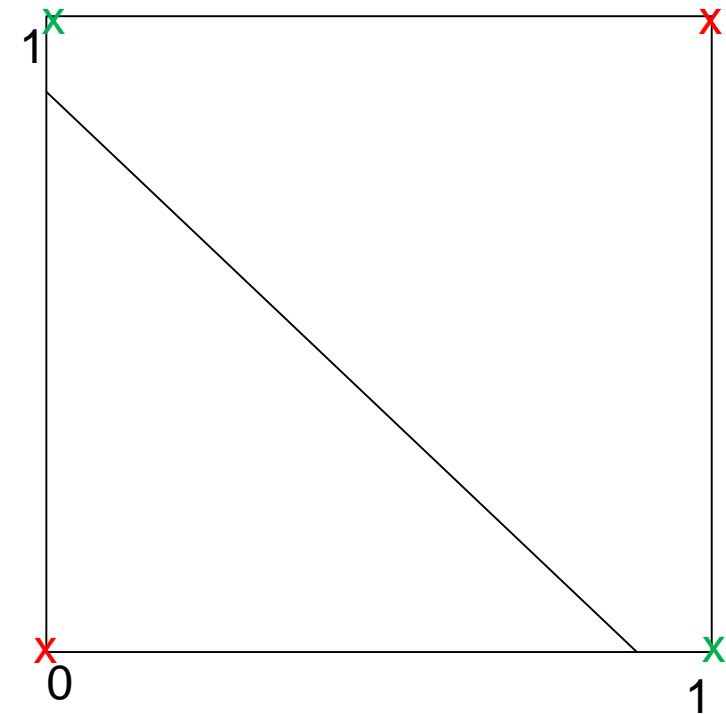
X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0



# Warum brauchen wir Deep Learning? – Das XOR Problem

- Problem bei der Klassifikation

X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0

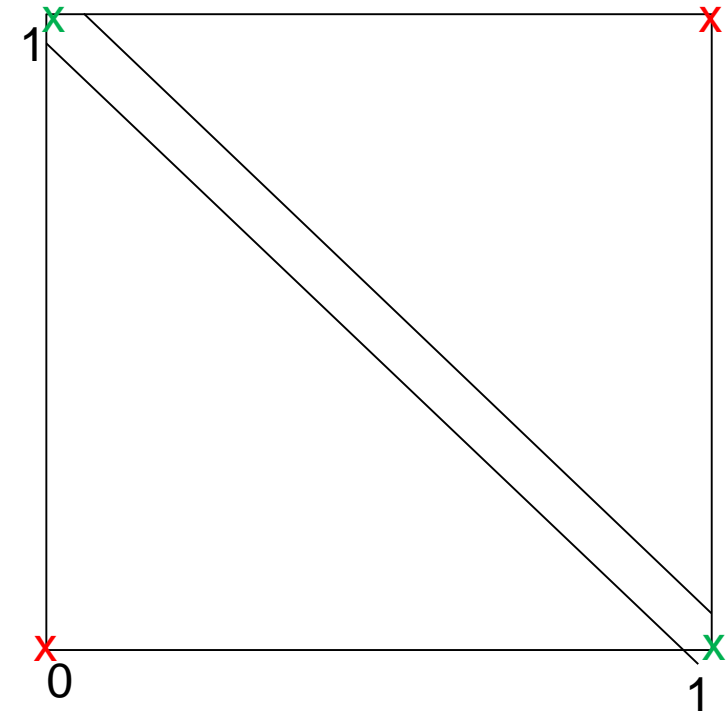


# Warum brauchen wir Deep Learning? – Das XOR Problem

- Problem bei der Klassifikation

X	Y	XOR
0	0	0
0	1	1
1	0	1
1	1	0

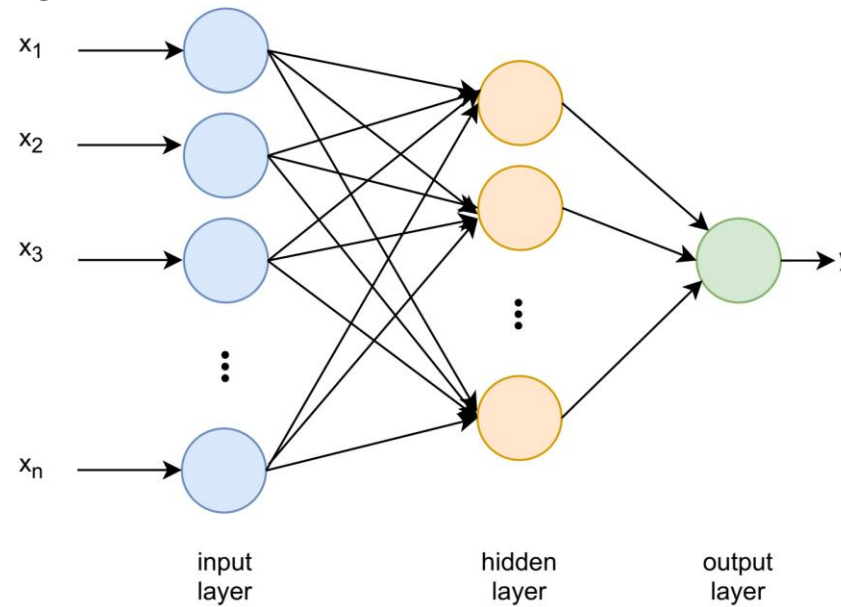
→ Multi-Layer Perceptron & Backpropagation





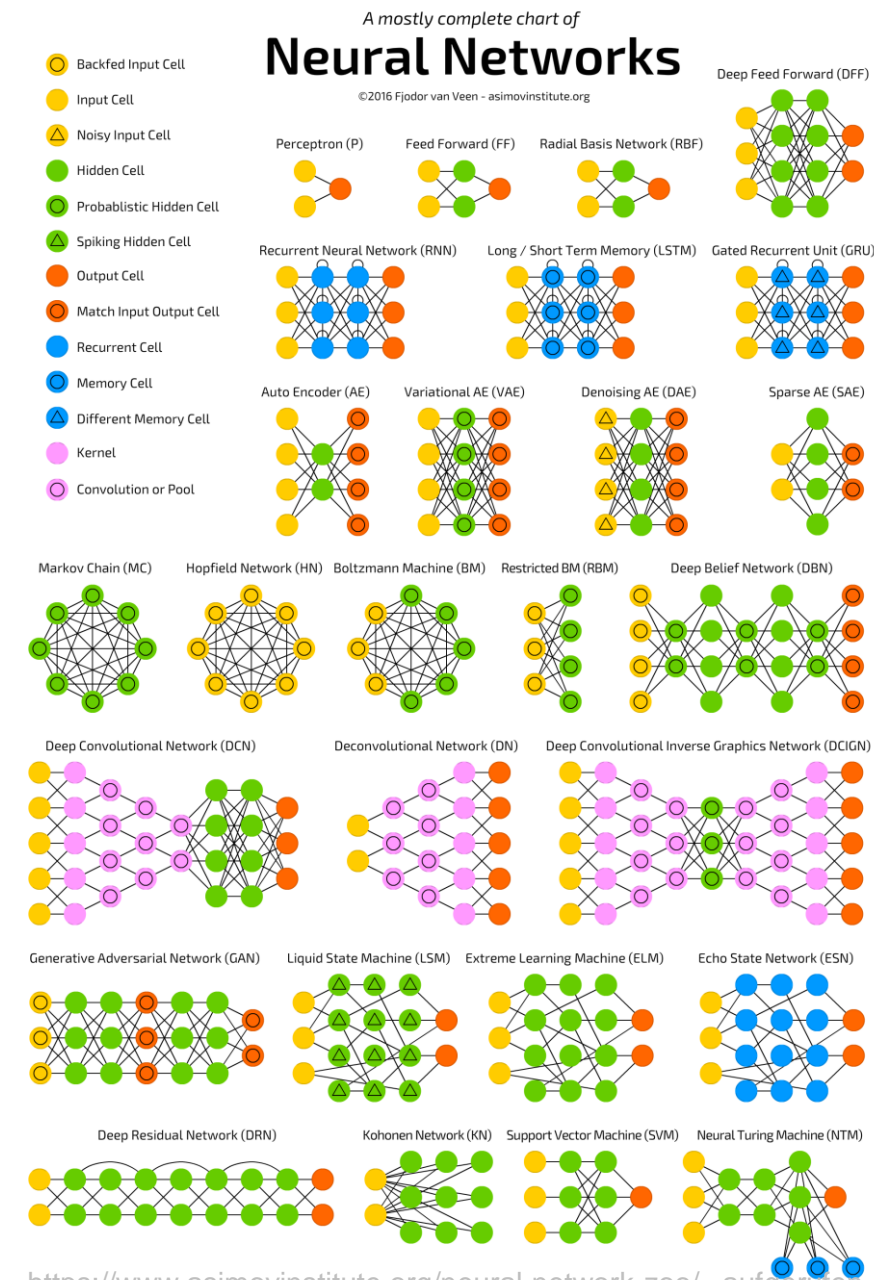
# Multi-Layer Perceptron aka Neuronales Netzwerk

- D. Rumelhart et. al. 1986
- 3 Layer-Typen
- Künstliches neuronales Netzwerk mit mehreren Layern = Deep Neural Network
- Der Durchbruch: Backpropagation: Fehleranteil pro Neuron (partielle Ableitungen)



# Andere Netzwerktypen

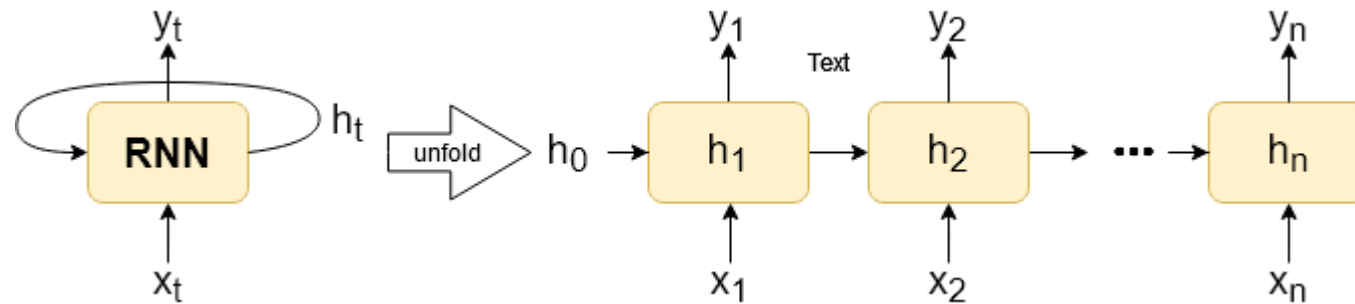
- Sequentielle Daten (Text, Audio, etc.): **Recurrent Neural Networks**, Attention, Transformer
- Bilder: **Convolutional Neural Networks**, GaNs
- Graph-Daten (Optimierung): Graph-Netzwerke
- ...



<https://www.asimovinstitute.org/neural-network-zoo/>, aufgerufen  
07.09.2020

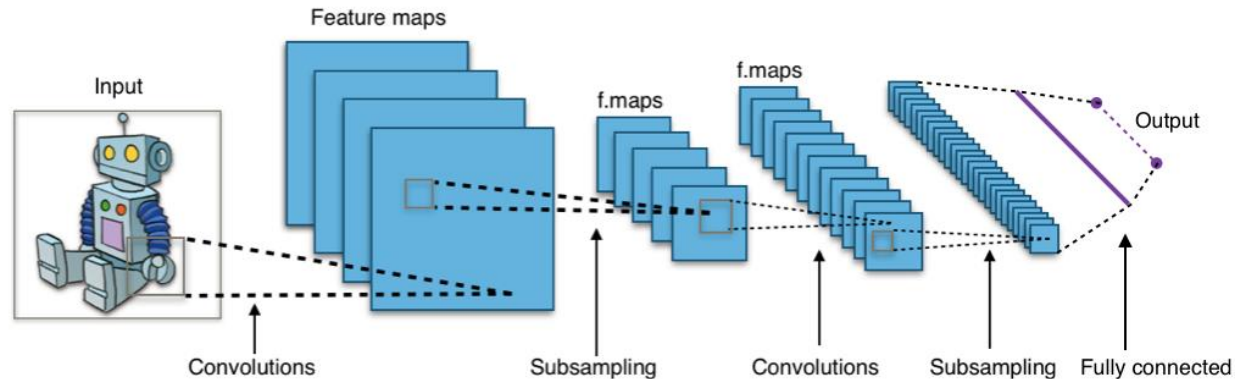
# Recurrent Neural Networks

- D. Rumelhart, 1988
- Netzwerk hat „Erinnerung“ aus vorherigem Schritt
- Weiterentwicklungen: LSTM, GRU
- Einsatz: Textgenerierung, Handschrifterkennung, Übersetzungen, etc.



# Convolutional Neural Networks

- Y. LeCun et. al. 1989
  - Convolution = Faltung
  - Faltung (Filter) von Bild für kleinere Netzwerkarchitektur
  - Braucht „klassischen“ letzten Layer
- 
- Einsatz: Objekterkennung, Bildklassifizierung, etc.

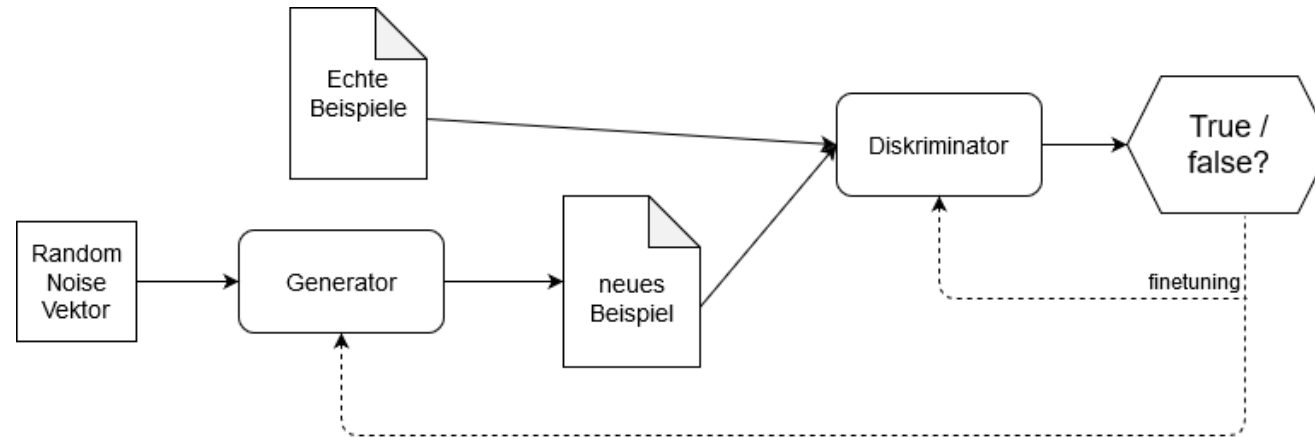


<https://jaai.de/convolutional-neural-networks-cnn-aufbau-funktion-und-anwendungsgebiete-1691/> , aufgerufen 07.09.2020



# Generative Adversarial Networks

- Ian Goodfellow et. al., 2014
- Spieltheorie: Spieler & Gegenspieler
- Spieler: erzeugt Ausgabe
- Gegenspieler: Versucht Ausgabe von realen Beispielen zu unterscheiden
- Einsatz: Generierung von realistischen Medien (Bilder, Musik, Text, etc.)





## Fragen zur Theorie?



# Deep Learning – Die Praxis



Wissen für Morgen



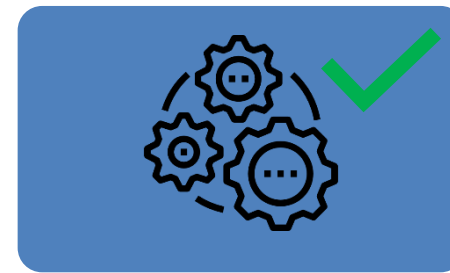
# Typischer Ablauf



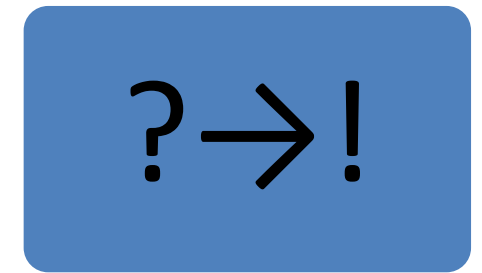
1. Daten



2. Training



3. Fertiges Modell



4. Anwenden



# Frameworks

- Python Deep-Learning-Bibliotheken:
  - PyTorch, Tensorflow, Keras
- Andere nützliche Python-Bibliotheken:
  - Numpy, Pandas
  - Scikit-learn
  - Matplotlib
  - OpenCV
- Java:
  - Java Machine Learning Library, Deep Java Library, Deeplearning4j, ...
- C++:
  - TensorFlow 146k, Caffe, Microsoft Cognitive Toolkit, ...
- Grafikkarten-Unterstützung: NVIDIA CUDA toolkit

 PyTorch

 Keras   
TensorFlow

Tensorflow.org, pytorch.org, keras.io , aufgerufen  
07.09.2020





## Kleine Demo



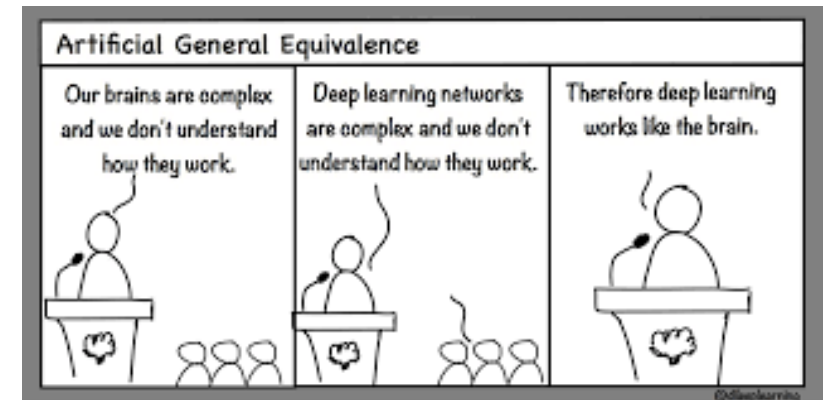
Wissen für Morgen





# Take-Home-Message

- Guter Datensatz ist wichtig!
- Netzwerk-Architektur: größer und komplizierter ist nicht immer besser
- Es gibt nicht das eine perfekte Netzwerk für alles
- Netzwerk dem Problem anpassen, nicht anders herum, **aber**
- Es muss auch nicht jedes Mal das Rad neu erfunden werden
- Oft kann Finetuning schon ausreichen (→ Transfer Learning)
- Ausprobieren!



<https://twitter.com/dileeplearning/status/1107148753884995584> , aufgerufen 07.09.2020

## Hilfreiche & Interessante Links

- <https://www.deeplearningbook.org/> - Online-Version des Buches Deep Learning von Ian Goodfellow et. al.
- <https://machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-improve-generalization-error/> - Details zu Over- und Underfitting
- <https://pytorch.org/tutorials/> - PyTorch Tutorials
- <https://www.tensorflow.org/tutorials/quickstart/beginner> - Tensorflow Tutorials
- <https://deepart.io/> - Fotos im Stil berühmter Kunstwerke mit GANs
- <https://openai.com/projects/> - Übersicht über KIs des Forschungslabors OpenAI
- <https://transformer.huggingface.co/> - Online-Demo von Transformers-Netzwerken
- <https://paperswithcode.com/> - Sammlung von aktuellen (AI-) Papern mit Code-Repositories



**Vielen Dank für Eure Aufmerksamkeit.**

**Gibt es noch Fragen?**



# Quellenverzeichnis

- Sarah Böning. Assessing the State-of-the-Art for Source-Code Suggestions. Master's Thesis, Bauhaus-Universität Weimar, 2020.
- Ian Goodfellow, et al. Generative adversarial Nets. Advances in neural information processing systems. 2014.
- Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., USA, 1 edition, 1997. ISBN 0070428077.
- David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. *Learning Representations by Back-Propagating Errors*, page 696–699. MIT Press, Cambridge, MA, USA, 1988.
- Prof. Dr. Benno Stein. Introduction to Machine Learning, Vorlesungsskript, Bauhaus-Universität Weimar, WiSe 18/19.
- Jacek M Zurada. *Introduction to artificial neural systems*, volume 8. West St. Paul, 1992.
- Bilder, Icons, GIFs falls nicht angegeben: privat, Iconfinder.com, Flaticon.com, giphy.com

