

# Deep Learning: Eine Einführung für Coder

Dr. Johann Heller

# Was ihr hier lernt

- Intuitive Erklärung wie Deep Learning funktioniert
- Verständnis woher der Hype kommt
- Transfer-Learning
- Klassifikation mit CNNs
- Extrem stark vereinfachte Zusammenhänge über Deep Learning

# Was ihr hier **nicht** lernt

- Deep Learning
- Mathematische Erklärungen
- Deployen und Monitoren von Modellen
- Das *beste* Framework



```
X = load_my_dataset(...) # <- Your Data goes here
model = OneModelToRuleThemAll.fit(X)
model.solve_all_my_problems()
```

# Technologie

- Jupyter Notebooks mit Python3
- Google Colaboratory
  - Kostenlos für 12 Stunden am Stück
  - kein Aufwand zu konfigurieren
  - Alles was ihr hier macht, könnt ihr ohne Problem weiter verwenden
- Cloud-Service mit GPUs
- keine Konfigurierung der Hardware

# Über mich



- M.Sc. in Elektrotechnik
- Dr.-Ing. in Beschleunigerphysik
- Experte in Deep Learning, Angewandter Mathematik und Softwaredesign
- vorher angestellt KI Startup in HRO
- z. Zt. Gründung **Emergo Machine Learning**

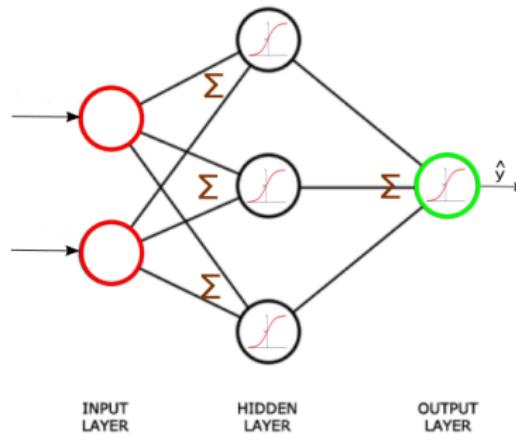
# Praktisches Beispiel

- Wir wollen ein Programm schreiben das optisch Hunde und Katzen unterscheiden kann
- Ursprünglich Kaggle Challenge
- 12500 Bilder pro Klasse



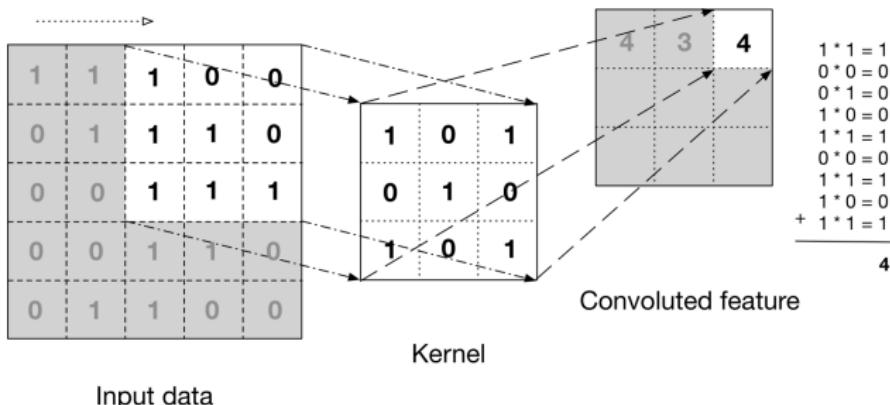
# Was ist ein neuronales Netz

- Das Herzstück modernen Deep Learnings
- jedes Neuron besteht aus gewichteter Summe der Inputs multipliziert mit nichtlinearer Aktivierungsfunktion



# Was ist ein Convolutional Neural Network

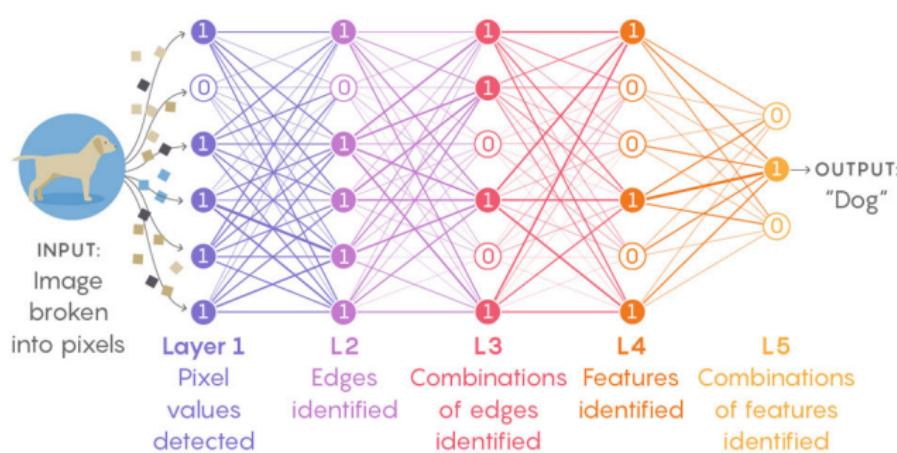
- Zusammensetzen einer Schicht aus vielen Faltungen



<sup>2</sup><http://www.davidsbatista.net/blog/2018/03/31/SentenceClassificationConvNets/>

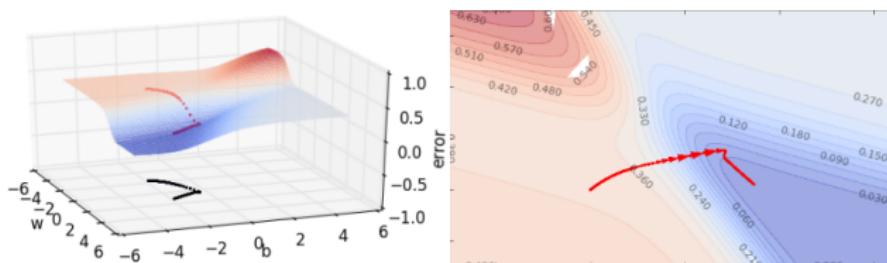
# Was tut ein Convolutional Neural Network

- Mehrere Schichten von solchen Faltungen, erlauben die Extraktion von immer komplexeren Features aus Datenströmen



# Stochastic Gradient Descent

- Wie optimiert man die Gewichte eines Netzes?
- Forward-Propagation der Daten, Back-Propagation des Fehlers, Optimierung der Gewichte so das Fehler minimal wird

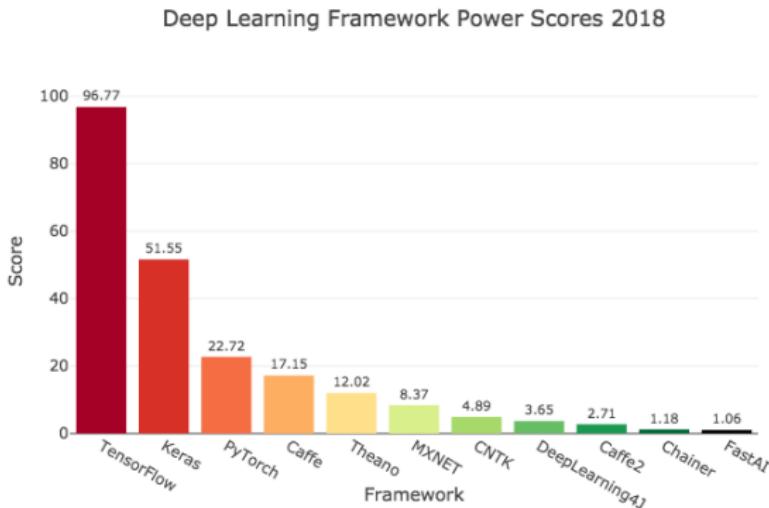


4

<sup>4</sup><https://towardsdatascience.com/why-visualize-gradient-descent-optimization-algorithms-a393806eee2>

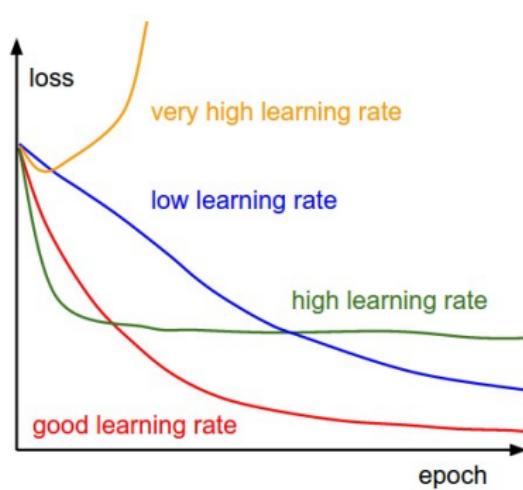
# Frameworks

- Es gibt extrem viele Frameworks, wie wählen wir das richtige aus
- Open-Source
- Fast alle werden über Python angesteuert



# Lernrate

- Die Lernrate beeinflusst die Konvergenzgeschwindigkeit
- Lernrate *optimal* wählen ist wichtig für finale Performance des Netzes

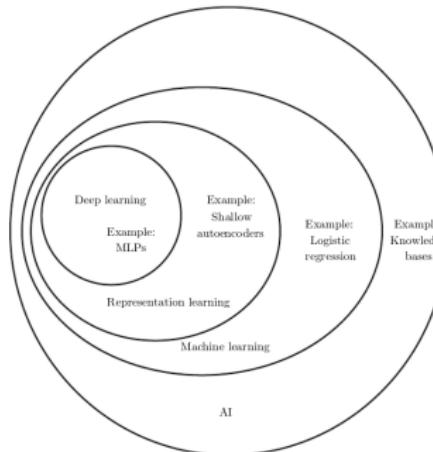


5

<sup>5</sup><https://www.hackerearth.com/blog/developers/3-types-gradient-descent-algorithms-small-large-data-sets/>

# Deep Learning vs Machine Learning vs KI

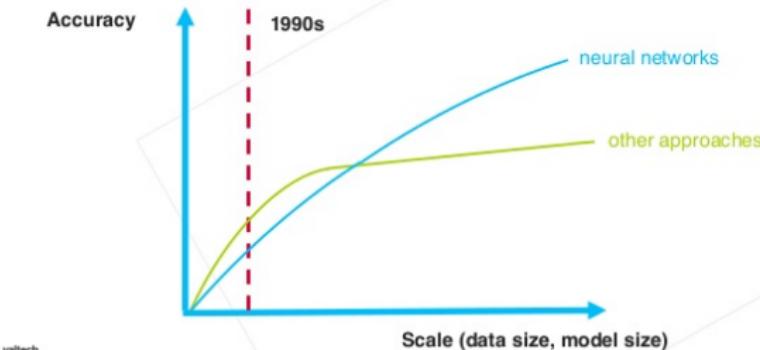
- Wie genau unterscheidet sich Deep Learning von Machine Learning
- KI umfasst alle Versuche Maschinen sinnvoll lernen zu lassen, DL ist nur eine von vielen Methoden dazu



# Skalierung

- Die Skalierung von DL Modellen ist einer der größten Vorteile gegenüber anderen Methoden
- Erlaubt das Erreichen sehr hoher Genauigkeiten

## More Data + Bigger Models



valtech.

<https://www.scribd.com/document/355752799/Jeff-Dean-s-Lecture-for-YC-AI>

# Lifecycle

- Lifecycle von ML Applikationen <sup>7</sup>



- Das eigentliche Training von Modellen macht nur 10-20% der Arbeitszeit aus

<sup>7</sup><https://towardsdatascience.com/data-science-as-software-from-notebooks-to-tools-part-2-e1d1bf99d067>

# Sinnvolle Skills für Machine und Deep Learning

- Mathematik
- Containerization und Orchestration (Docker und Kubernetes)
- Datenbanken (SQL und NoSQL)
- Datawarehousing
- Webscraping
- Webdevelopment
- Cloudcomputing z.B. AWS
- Schnelles Einarbeiten in verschiedene Verticals
- Continuous Intergration, Continuous delivery

# Zusammenfassung

- CNNs eignen sich hervorragend zur Klassifikation von Bildern
- Transfer-Learning
- Klassifikation ist eine nahezu gelöstes Problem
- Image-Augmentation ist eine von vielen Techniken um die Performance von CNNs zu verbessern

# Fragen

johann.heller@protonmail.ch

# GANs

- Generative Adverserial Networks
- Erfunden von Ian Goodfellow in 2015
- Erlaubt das erzeugen von Daten aus der selben Dichtefunktion wie die Trainingsdaten<sup>8</sup>



<sup>8</sup>Gutes Beispiel <https://thispersondoesnotexist.com/>

# Capsule Networks

- Erfunden von Geoffrey Hinton (einem der Pioniere des Deep Learnings)
- löst theoretisch eines der Probleme von CNNs
- Zur Zeit noch zu aufwändig zu trainieren



# OpenAI GPT-2

- Text-Generierung von OpenAI mit GPT-2
- Input: *Legolas and Gimli advanced on the orcs, raising their weapons with a harrowing war cry.*
- Output: *The orcs' response was a deafening onslaught of claws, claws, and claws; even Elrond was forced to retreat. "You are in good hands, dwarf," said Gimli, who had been among the first to charge at the orcs; it took only two words before their opponents were reduced to a blood-soaked quagmire, and the dwarf took his first kill of the night. The battle lasted for hours until two of the largest Orcs attempted to overwhelm Aragorn. When they finally stopped, they lay defeated and lifeless for miles and miles.*

# Künstliche Intelligenz

- Echte künstliche Intelligenz ist noch nicht in Sicht

**Yann LeCun**

March 14 at 3:15am ·

[Follow](#)

Statement from a Slashdot post about the AlphaGo victory: "We know now that we don't need any big new breakthroughs to get to true AI"

That is completely, utterly, ridiculously wrong.

As I've said in previous statements: most of human and animal learning is unsupervised learning. If intelligence was a cake, unsupervised learning would be the cake, supervised learning would be the icing on the cake, and reinforcement learning would be the cherry on the cake. We know how to make the icing and the cherry, but we don't know how to make the cake.

We need to solve the unsupervised learning problem before we can even think of getting to true AI. And that's just an obstacle we know about. What about all the ones we don't know about?

#deeplearning #AI #AlphaGo

# Gartner Hype Cycle

- Deep Learning auf der Spitze des Hype Cycles

