

How to recognize fake AI-generated images



Kyle McDonald Dec 5, 2018 · 7 min read



In 2014 machine learning researcher Ian Goodfellow introduced the idea of generative adversarial networks or GANs. “Generative” because they output things like images rather than predictions about input (like “hotdog or not”); “adversarial networks” because they use two neural networks competing with each other in a “cat-and-mouse game”, like a cashier and a counterfeiter: one trying to fool the other into thinking it can generate real examples, the other trying to distinguish real from fake.

The first GAN images were easy for humans to identify. Consider these faces from 2014.





How to recognize fake AI-generated images



Kyle McDonald Dec 5, 2018 · 7 min read



In 2014 machine learning researcher Ian Goodfellow introduced the idea of generative adversarial networks or GANs. “Generative” because they output things like images rather than predictions about input (like “hotdog or not”); “adversarial networks” because they use two neural networks competing with each other in a “cat-and-mouse game”, like a cashier and a counterfeiter: one trying to fool the other into thinking it can generate real examples, the other trying to distinguish real from fake.

The first GAN images were easy for humans to identify. Consider these faces from 2014.

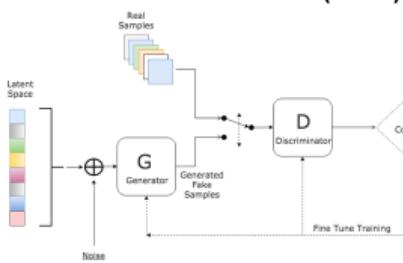


8 Machine Learning III

- Specialized Areas in Machine Learning

(4) Generative Adversarial Networks GAN

Generative Adversarial Networks (GAN)



A Generative Adversarial Network (GAN) is a class of machine learning frameworks designed by Goodfellow and his colleagues in 2014.

Two neural networks contest with each other in a game. Given a training set, this technique learns to generate new data with the same statistics as the training set.

The core idea of a GAN is based on the "indirect" training through the discriminator, which itself is also being updated dynamically. This basically means that the generator is not trained to minimize the distance to a specific image, but rather to fool the discriminator. This enables the model to learn in an unsupervised manner. (Wikipedia)

StyleGAN: Motivation Style Transfer

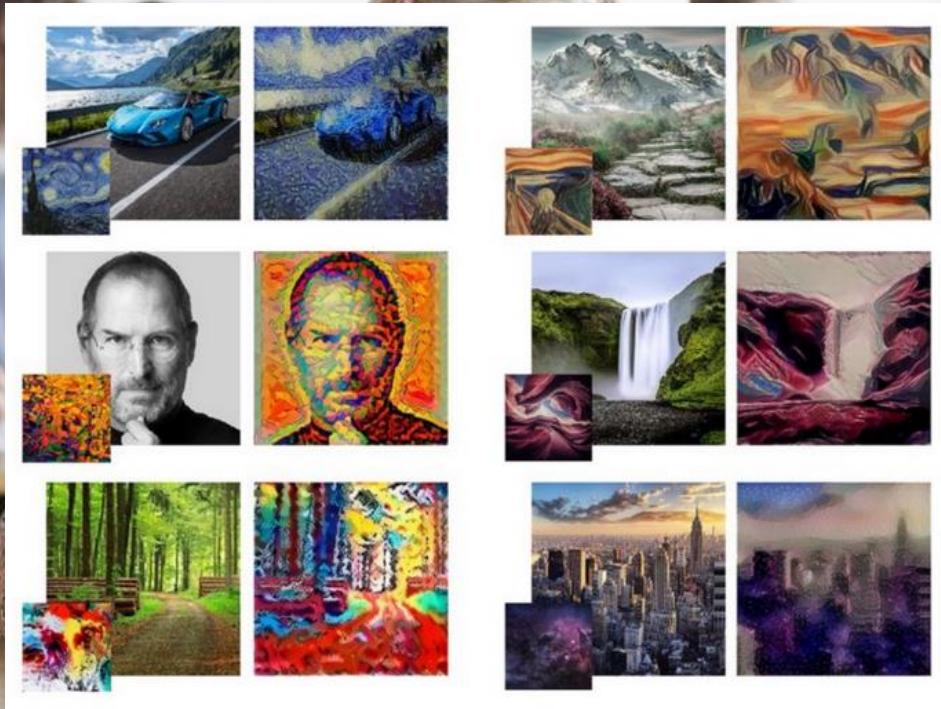


Intuitive Guide to Neural Style Transfer, 2019

towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-neural-style...

Which machine learning technique is used in this case (and in Deepfakes)?

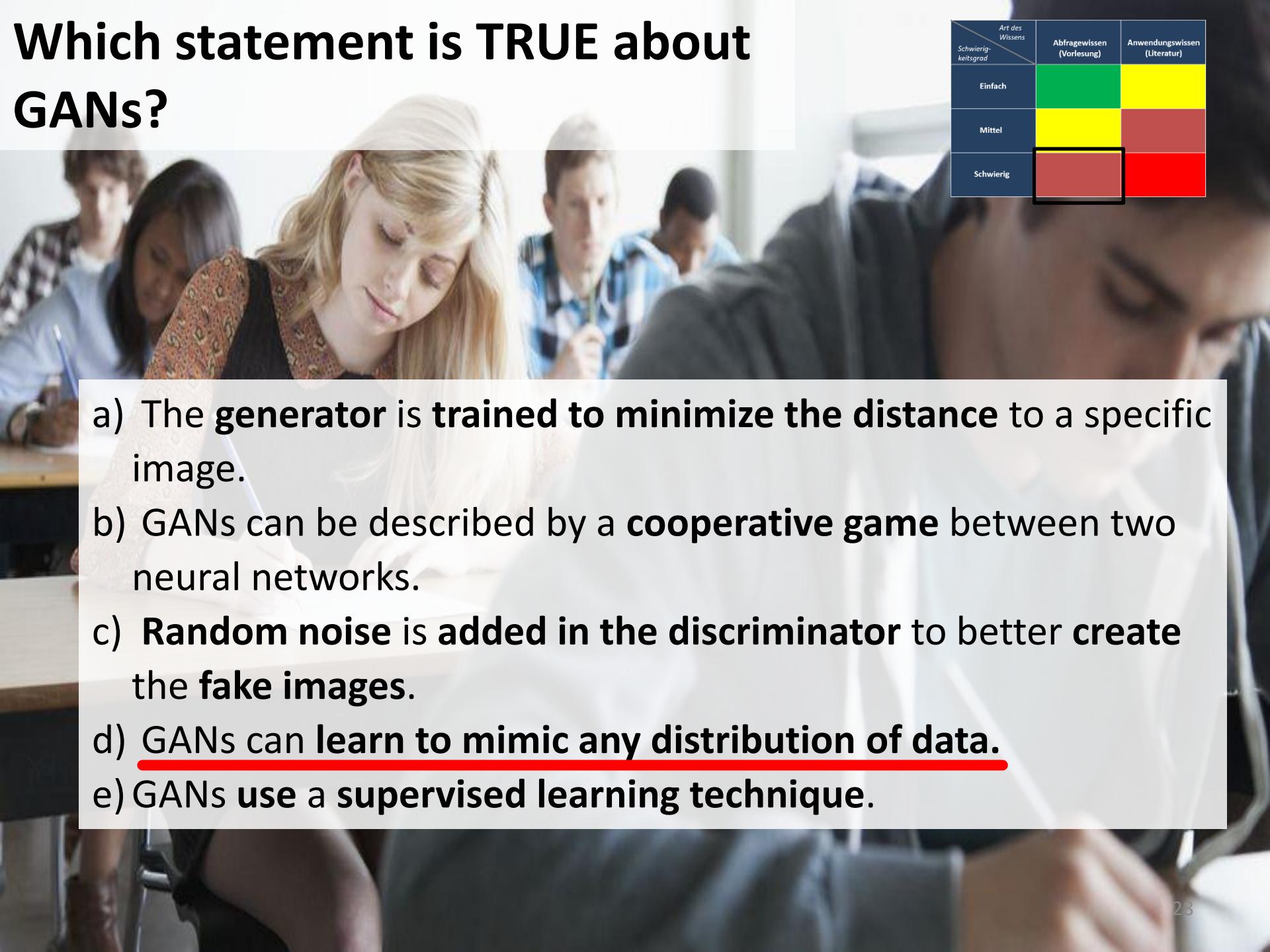
Schwierigkeitsgrad	Art des Wissens Abfragewissen (Vorlesung)	Anwendungswissen (Literatur)
Einfach	Green	Yellow
Mittel	Yellow	Red
Schwierig	Red	Red



- a) Convolutional Neural Networks
- b) Transfer Learning
- c) Generative Adversarial Networks
- d) Reinforcement Learning
- e) Transformer

Which statement is TRUE about GANs?

Schwierigkeitsgrad	Art des Wissens	Abfragewissen (Vorlesung)	Anwendungswissen (Literatur)
Einfach		Green	Yellow
Mittel		Yellow	Red
Schwierig		Red	Red

- 
- a) The **generator** is trained to minimize the distance to a specific image.
 - b) GANs can be described by a **cooperative game** between two neural networks.
 - c) **Random noise** is added in the discriminator to better **create** the **fake images**.
 - d) **GANs** can **learn to mimic any distribution of data**.
 - e) GANs **use a supervised learning technique**.

Explore our schools to find your perfect Nanodegree Program

Learn in-demand skills, build incredible projects, and gain an industry-valued Nanodegree program.



SCHOOL OF
Data Science



SCHOOL OF
Artificial
Intelligence



SCHOOL OF
Programming



SCHOOL OF
Autonomous
Systems



SCHOOL OF
Cloud Computing



SCHOOL OF
Business

Your path to the right job

■ Machine Learning
Engineer

Machine Learning Engineer

Deep Learning
Engineer

Artificial Intelligence
Specialist

Machine learning is becoming a fundamental skill as software development is entering a new era. This path will enable you to start a career as a Machine Learning Engineer. First learn the fundamentals of programming in Python, linear algebra, and neural networks, and then move on to core Machine Learning concepts.

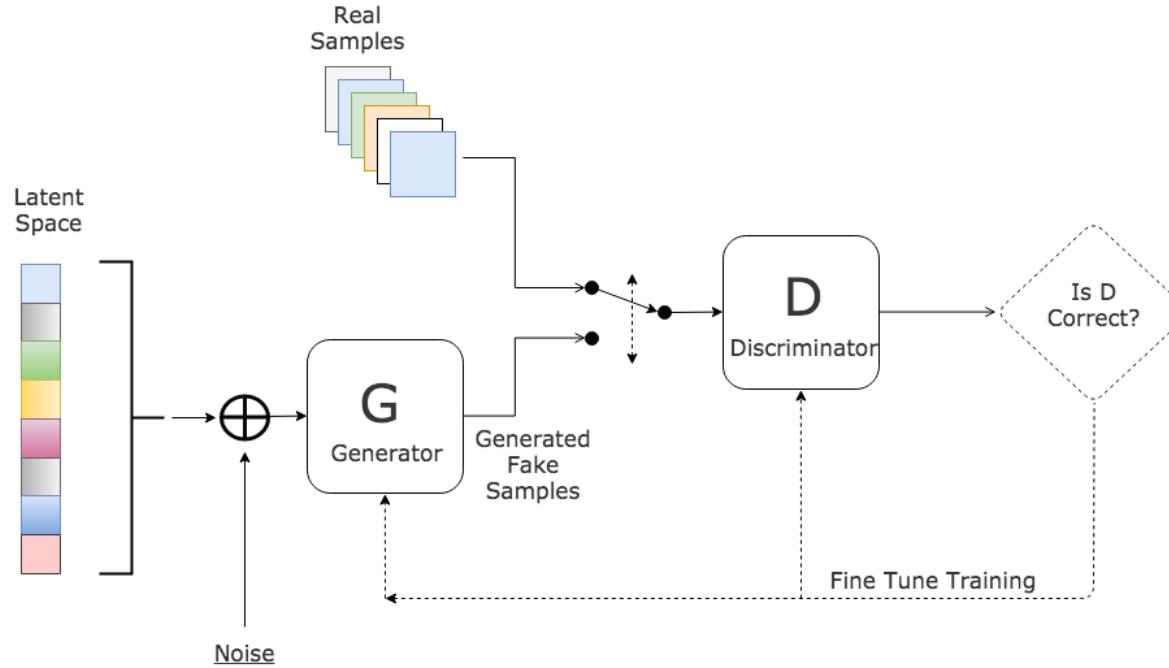
RECOMMENDED PROGRAMS

ALUMNI SUCCESS



Jeremy Jordan

Generative Adversarial Networks (GAN)



A **Generative Adversarial Network (GAN)** is a class of **machine learning** frameworks designed by **Goodfellow** and his colleagues in **2014**.

Two neural networks contest with each other in a game. Given a **training set**, this technique learns to **generate new data** with the same statistics as the **training set**.

The **core idea** of a GAN is based on the "indirect" training through the **discriminator**, which itself is also being **updated dynamically**. This basically means that the **generator** is **not trained to minimize the distance to a specific image**, but rather to **fool the discriminator**. This enables the model to learn in an **unsupervised manner**. (Wikipedia)¹⁸

Transfer Learning with CNNs (Convolutional NN)

OpenAI

OpenAI Microscope

We're introducing a collection of visualizations for layer and neuron of "neuron organisms" which increase interpretability. Microscope analyzes the features learned by neural networks, and is a research community for understanding them.

April 14, 2020
2 minute read

Mit OpenAI Microscope sieht man, dass frühe Schichten eines Modells abstrakte, späte Schichten konkrete Muster lernen. Kernelement beim Transfer Learning: Abstraktes Wissen weiterverwenden, spezifisches Wissen nachtrainieren.

Microscope

MODELS ABOUT

Models

AlexNet

The OpenAI Microscope is a collection of visualizations for significant neurons in neural vision models.

Inception v1

Inception v1 (Places)

VGG 19

Inception v3

Inception v4

ResNet v2 50

Inception v1

mixed4c

Unit 447

FEATURE VISUALIZATION

An artificial, optimized image that maximizes activations of the given unit. [Read more](#).

CHANNEL OPTIMIZATION objective results in a repeating pattern.

NEURON OPTIMIZATION objective shows spatial preferences.

DATASET SAMPLES

Pieces of images from the training dataset that result in the largest activations from the given unit.

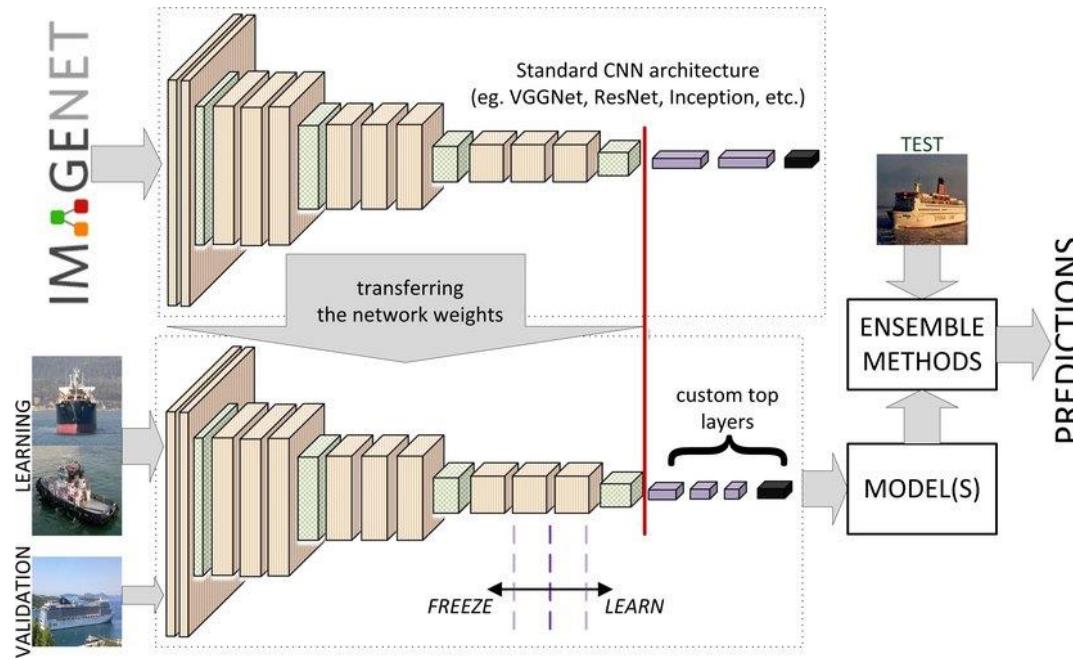
These images are cropped and downsize samples from the [ImageNet](#) research dataset. Unlike our other visualizations, they are not CC-BY-SA because they are derived from ImageNet.

maxpool0

Unit 45 Unit 46 Unit 47

7

Transfer Learning: Options when applying

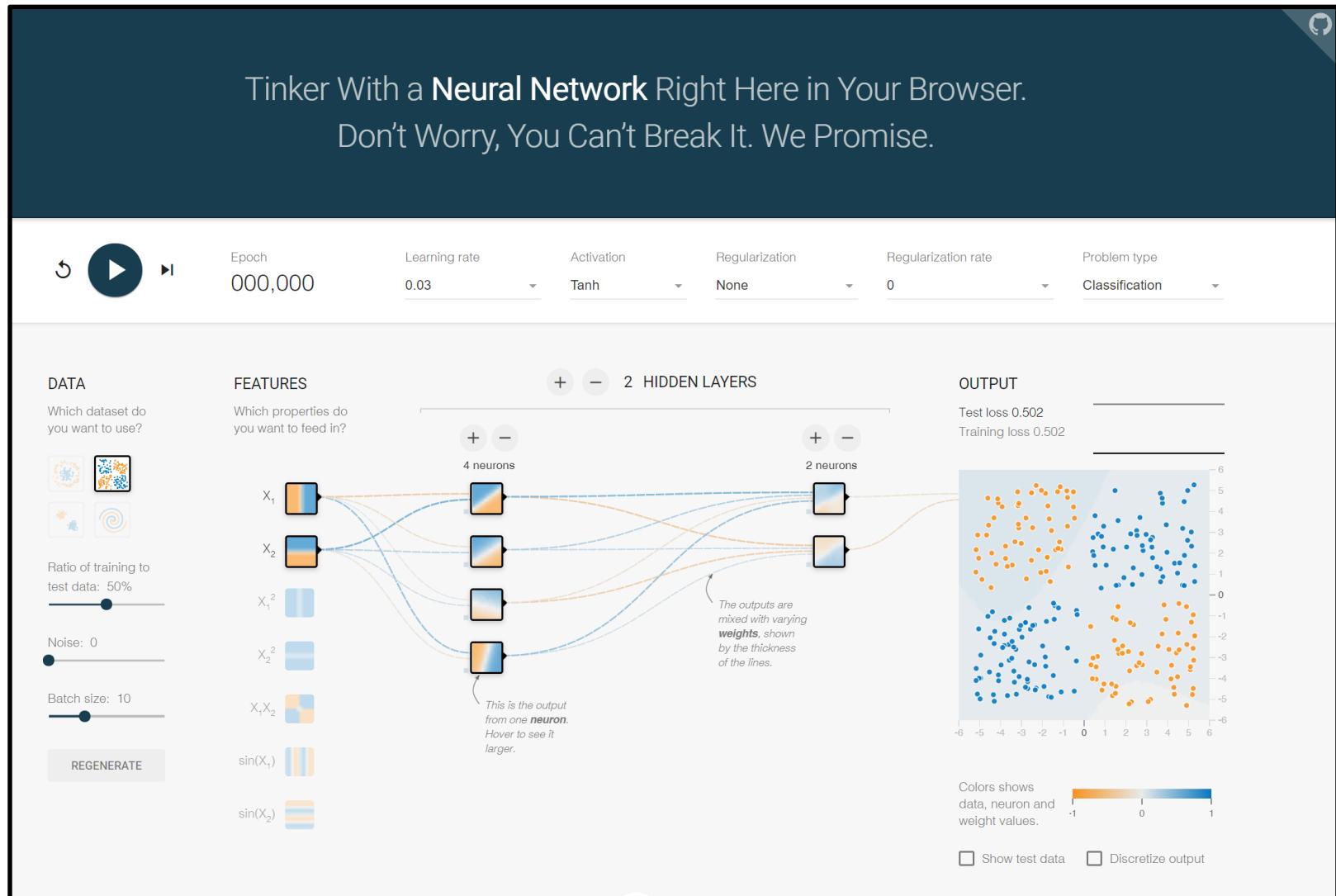


2 Options When applying Transfer Learning in Neural Networks:

- **Freeze the Weights and Bias on Initial Few Layers** and train only the **Last Few Layers** and **Fully Connected**.
In this case you **don't** need to **re-train** the **whole Network Model** again.
- **Re-train the whole Network**, **initializing** from the **learned Weights and Bias**.
Keep the Learning Rate very low so that the **original Weights** don't deviate drastically.

8 Machine Learning I

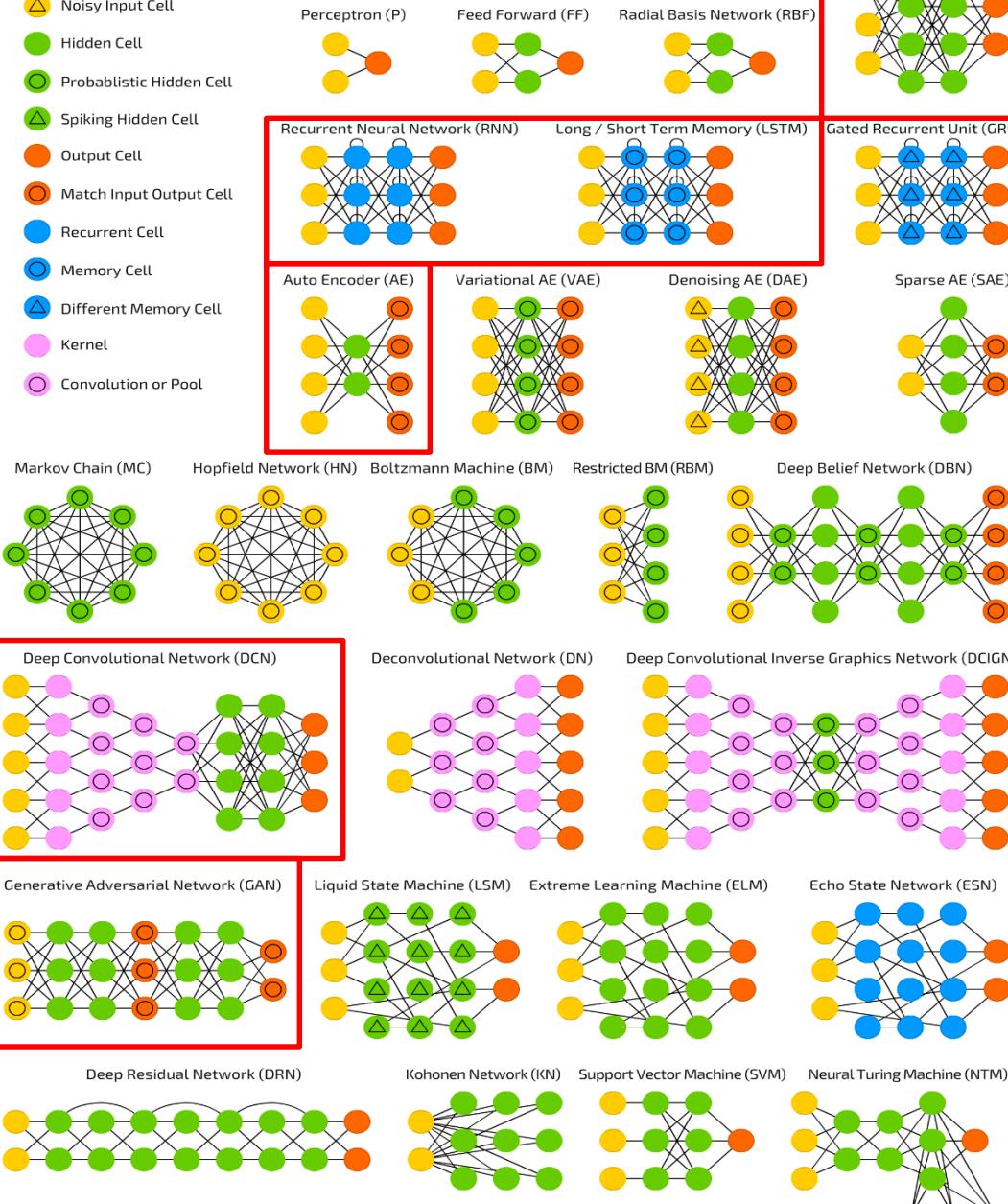
- Neural Networks & Deep Learning (3) TensorFlow Playground



Neural Networks

©2016 Fjodor van Veen - [asimovinstitute.org](http://www.asimovinstitute.org)

- Backfed Input Cell
- Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- △ Different Memory Cell
- Kernel
- Convolution or Pool



THE NEURAL NETWORK ZOO, www.asimovinstitute.org/neural-network-zoo/

Neuronale Netze stellen einen zentralen Baustein für Deep Learning dar.

Im Laufe der Jahre wurden zahlreiche **Architekturen** für Neuronale Netze entwickelt.

Die Wahl der **Architektur** hängt von der **Datenstruktur**, den **Dateninhalten** und der **Aufgabenstellung** ab.

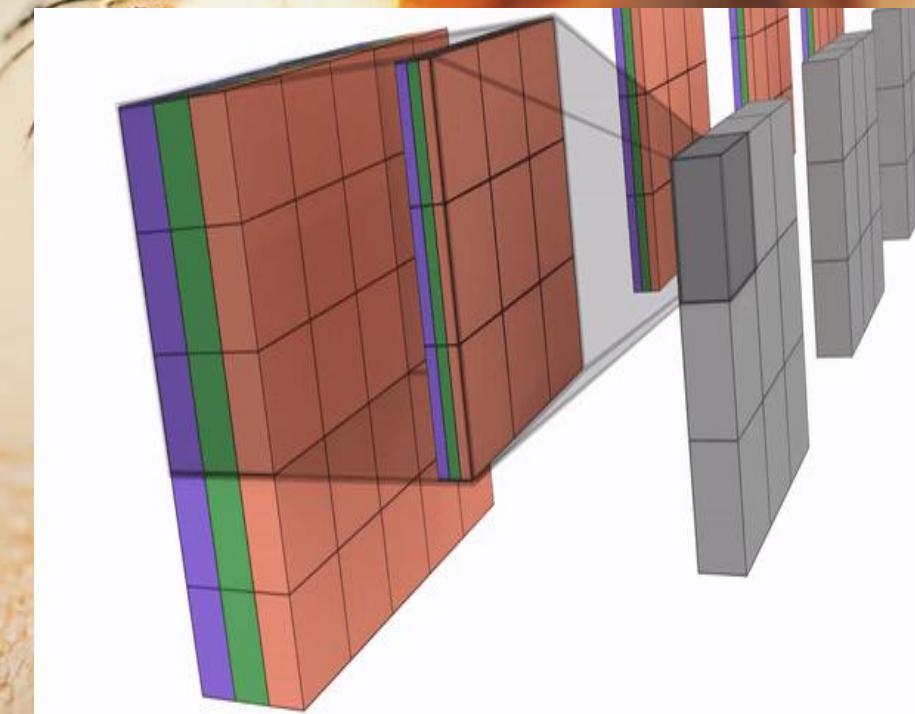
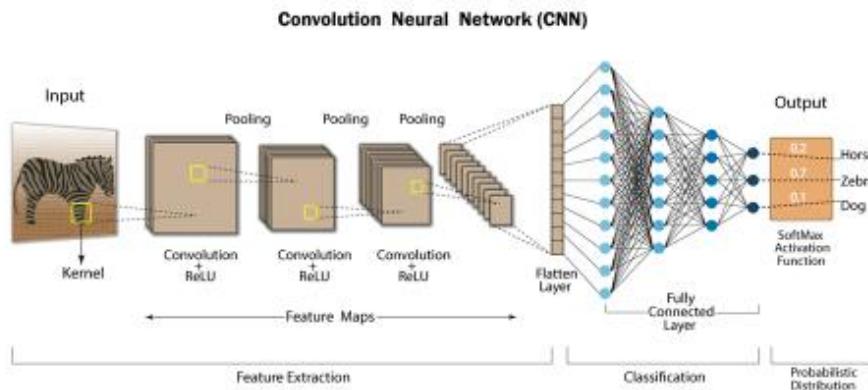
Aktuell wichtige Klassen sind u.a. **Convolutional Neural Networks (CNN)**, **Recurrent Neural Networks (RNN)**, **Autoencoder**, **Generative Adversarial Networks (GAN)**.

8 Machine Learning I

- Neural Networks & Deep Learning

(4) Basics of Convolutional Neural Networks (CNN)

CNN: Grundstruktur am Beispiel





RESEARCH ARTICLE

10.1029/2020MS002301

Key Points:

- Machine learning is successfully applied to the warm-rain parameterization problem
- Training and testing data for the warm-rain kinetic collection equation are provided using the superdroplet method
- Standard training methods show some limitations for the resulting ODE system

Supporting Information:

- Supporting Information S1

Correspondence to:

A. Seifert,
axel.seifert@dwd.de

Citation:

Seifert, A., & Rasp, S. (2020). Potential and limitations of machine learning for modeling warm-rain cloud

microphys-

Advances

12, e2020

10.1029/

Received

Accepted

Accepted

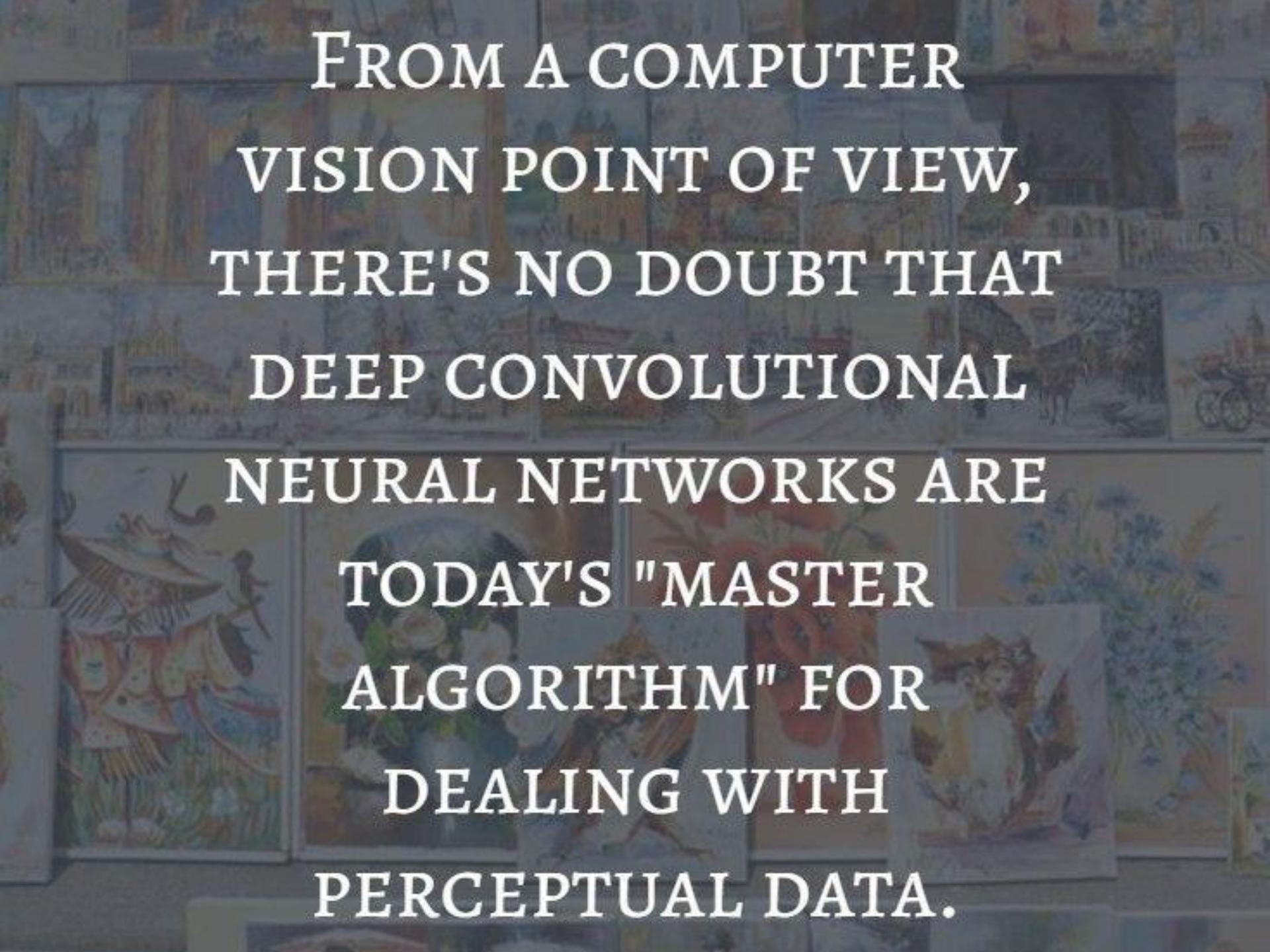
Potential and Limitations of Machine Learning for Modeling Warm-Rain Cloud Microphysical Processes

Axel Seifert¹  and Stephan Rasp² 

¹Deutscher Wetterdienst, Offenbach, Germany, ²TU München, Munich, Germany

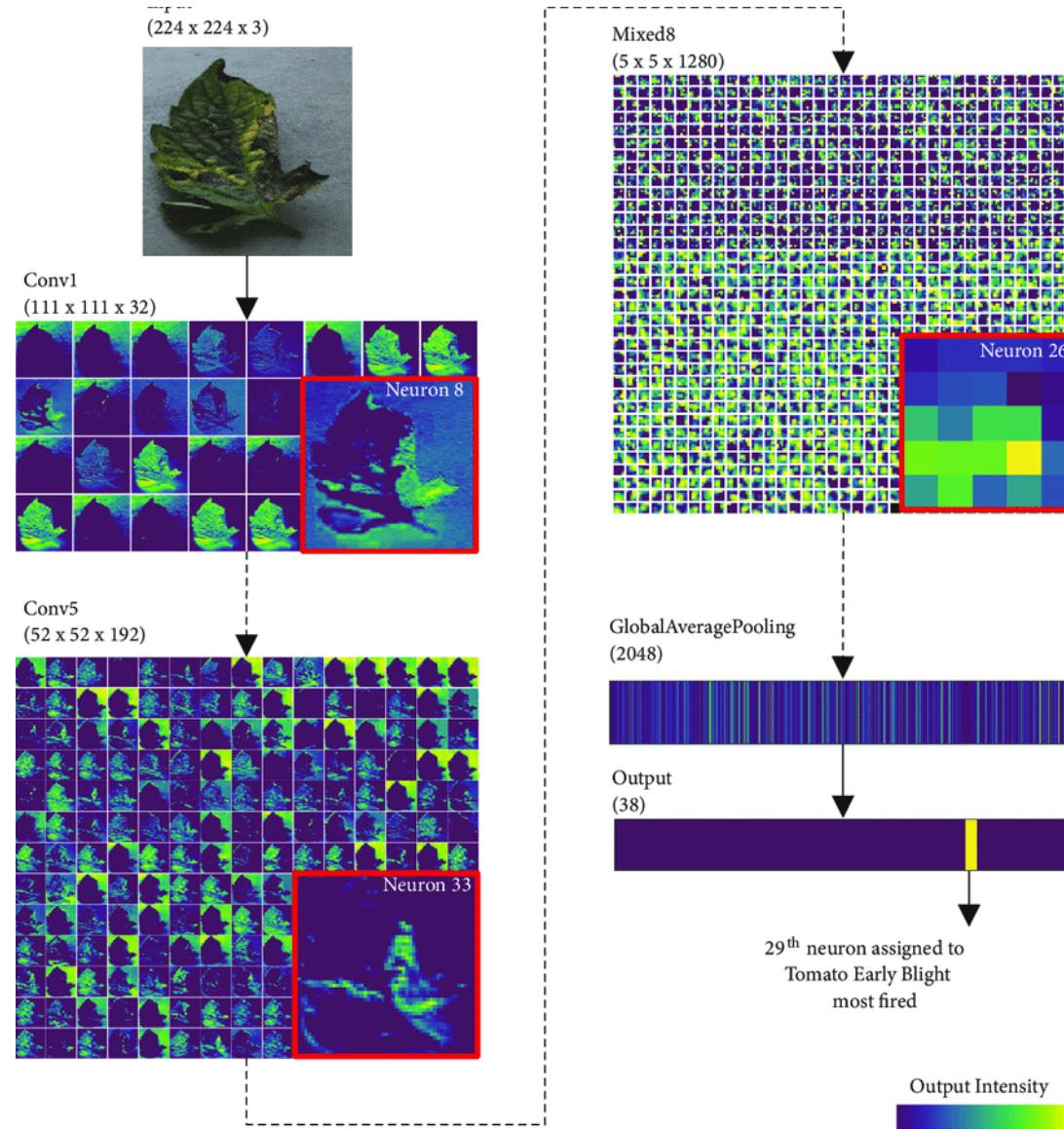
Abstract The use of machine learning based on neural networks for cloud microphysical parameterizations is investigated. As an example, we use the warm-rain formation by collision-coalescence, that is, the parameterization of autoconversion, accretion, and self-collection of droplets in a two-moment framework. Benchmark solutions of the kinetic collection equations are performed using a Monte Carlo superdroplet algorithm. The superdroplet method provides reliable but noisy estimates of the warm-rain process rates. For each process rate, a neural network is trained using standard machine learning techniques. The resulting models make skillful predictions for the process rates when compared to the testing data. However, when solving the ordinary differential equations, the solutions are not as good as those of an established warm-rain parameterization. This deficiency can be seen as a limitation of the machine learning methods that are applied, but at the same time, it points toward a fundamental ill-posedness of the commonly used two-moment warm-rain schemes. More advanced machine learning methods that include a notion of time derivatives, therefore, have the potential to overcome these problems.

Plain Language Summary In our work, we are trying to teach a computer how rain forms in clouds. We show that computer hundreds of cases in the form of data. To be honest, the data are not real data but only results of simulations with a more complicated computer model. This complicated model can track the collisions of 10,000 of droplets, and we save all that data about the growth of the droplets into larger raindrops. This is what we then give to the simpler computer model to teach it something about clouds and rain. Afterward, it can make pretty good predictions about which clouds will rain and how long it will take them to produce the first rain. Unfortunately, the current machine learning methods are still a bit stupid because they only learn from the data but do not understand the mathematics and the physics behind the data. Therefore, the new computer model is still not as good at predicting rain as some clever mathematical formulas that were developed 20 years ago. Maybe we first have to teach the computer model more about calculus before it can learn to predict rain.



FROM A COMPUTER
VISION POINT OF VIEW,
THERE'S NO DOUBT THAT
DEEP CONVOLUTIONAL
NEURAL NETWORKS ARE
TODAY'S "MASTER
ALGORITHM" FOR
DEALING WITH
PERCEPTUAL DATA.

CNN: Eine Beispielanwendung



8 Machine Learning I

- Neural Networks & Deep Learning

Content:

1. Motivation
2. Basics of
 Neural Networks (NN)
3. **TensorFlow Playground**
4. Basics of **Convolutional
 Neural Networks (CNN)**
5. Deep Learning (DL) by
 Deepmind: AlphaGo, Zero...
6. Deep NN in
 Tesla Autonomous Driving
7. Summary

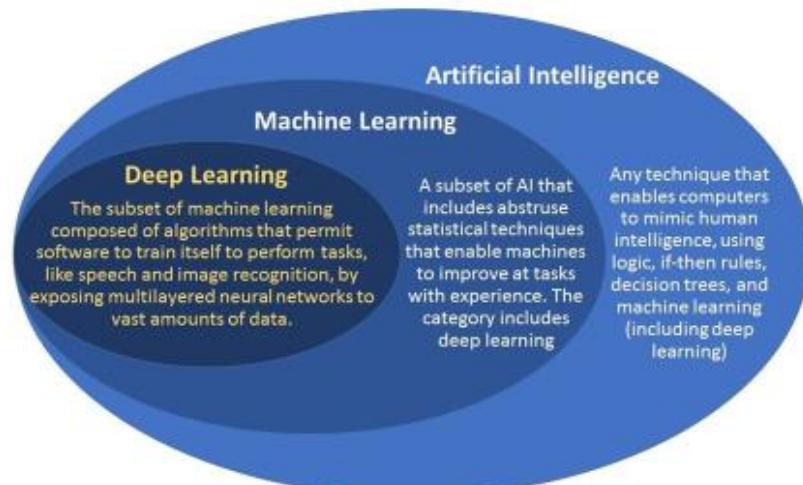
0
1
2
3
4
5
6
7
8
9

8 Machine Learning I

- Neural Networks & Deep Learning

(1) Motivation & Intro Machine Learning

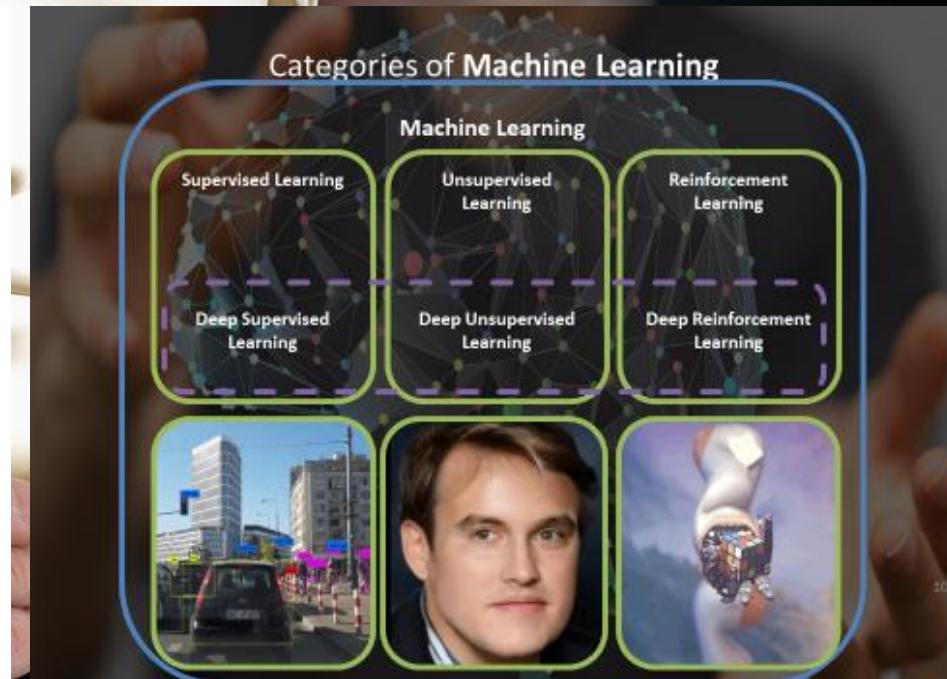
AI vs Machine Learning vs Deep Learning



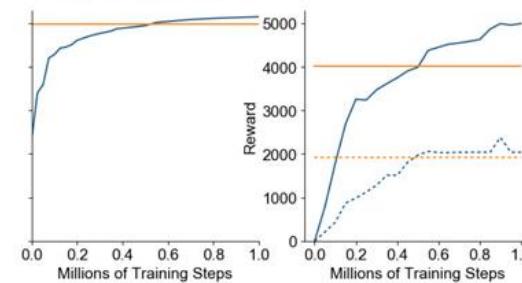
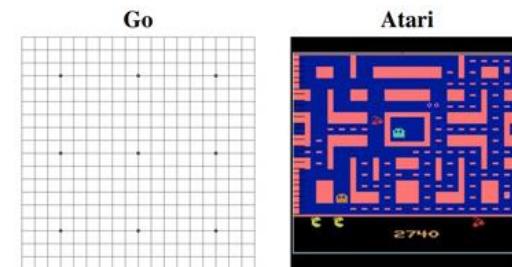
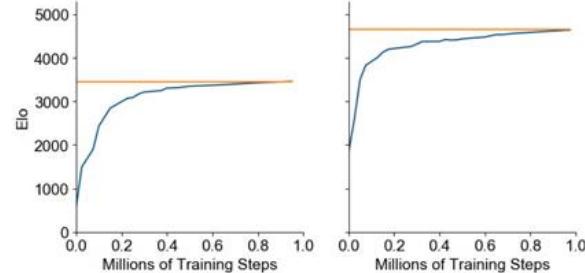
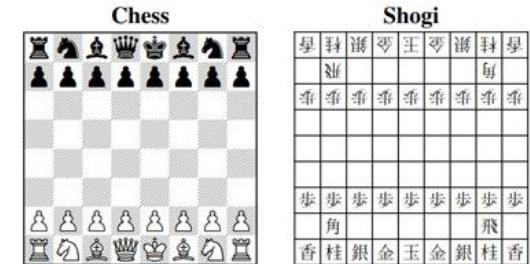
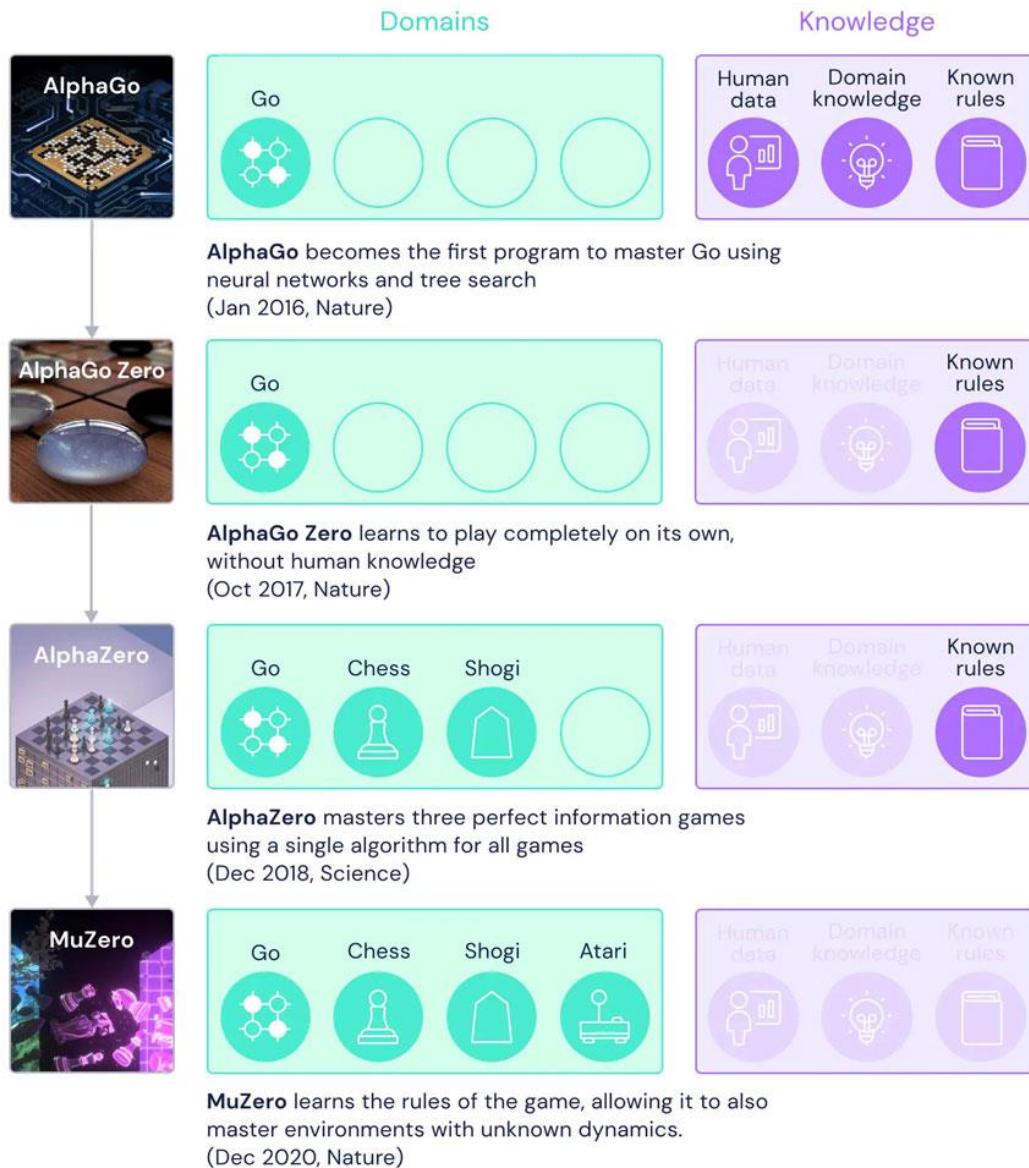
What is the difference between AI, machine learning and deep learning?

www.geospatialworld.net/blogs/difference-between-ai-machine-learning-and-deep-learning/

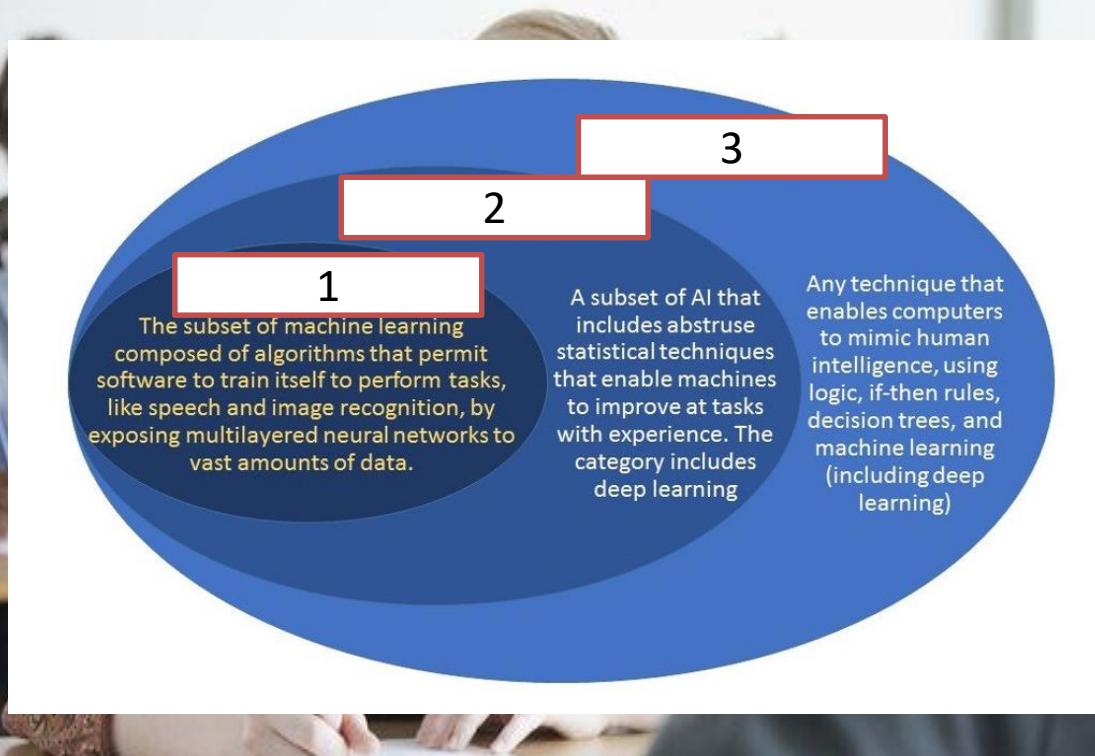
Categories of Machine Learning



Deepmind: Von AlphaGo über AlphaZero zu MuZero

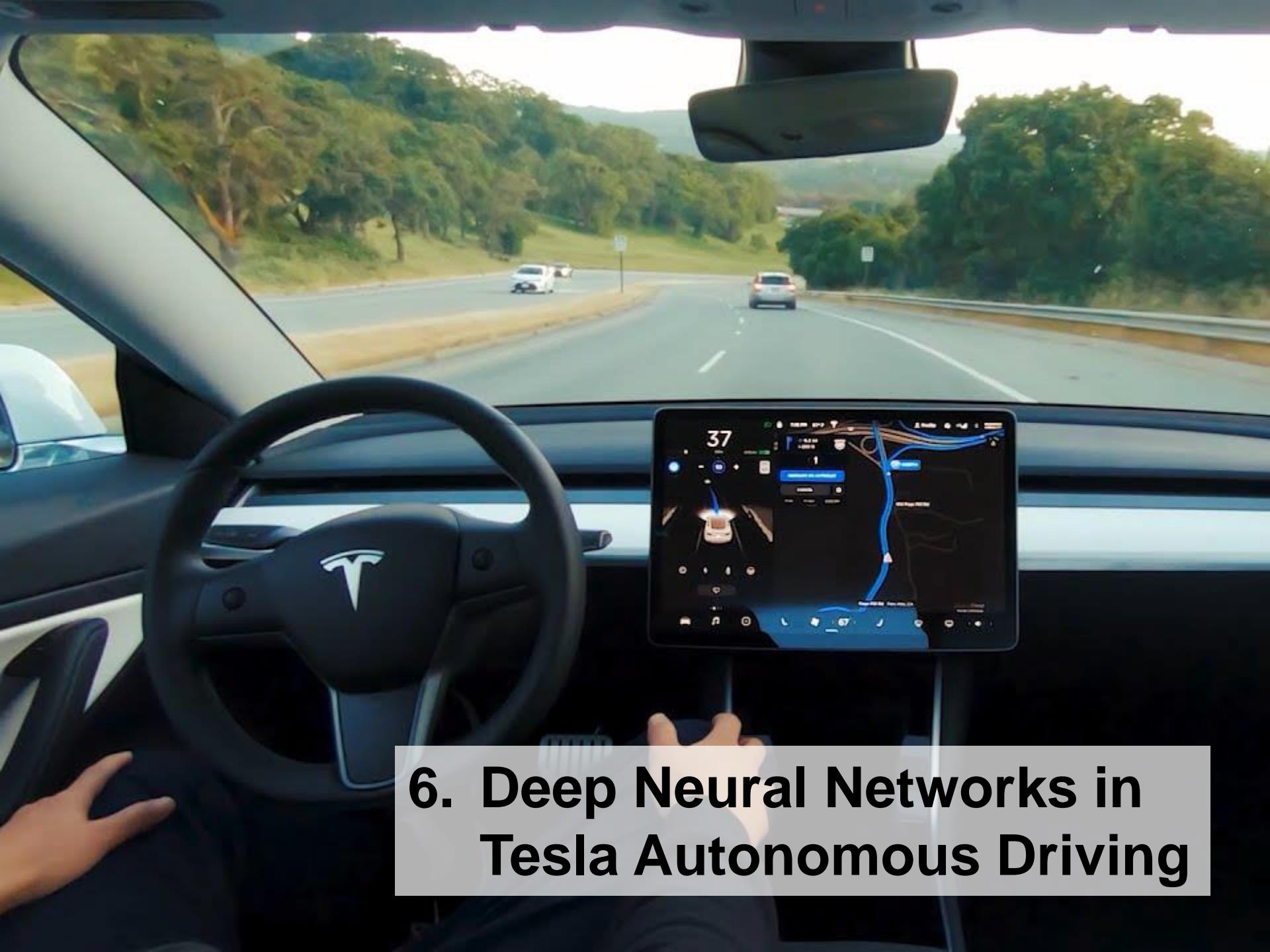


Fill in the blank



		Art des Wissens	Abfragewissen (Vorlesungen)	Anwendungswissen (Literatur)	
		Schwierigkeitsgrad	Einfach	Mittel	Schwierig
Einfach	Abfragewissen (Vorlesungen)	[Red Box]			
	Einfach				
Mittel	Mittel				
Schwierig	Schwierig				

- a) Artificial Intelligence | Machine Learning | Deep Learning
- b) Artificial Intelligence | Deep Learning | Machine Learning
- c) Machine Learning | Deep Learning | Artificial Intelligence
- d) Deep Learning | Machine Learning | Artificial Intelligence
- e) Machine Learning | Artificial Intelligence | Deep Learning



6. Deep Neural Networks in Tesla Autonomous Driving

8 Machine Learning I

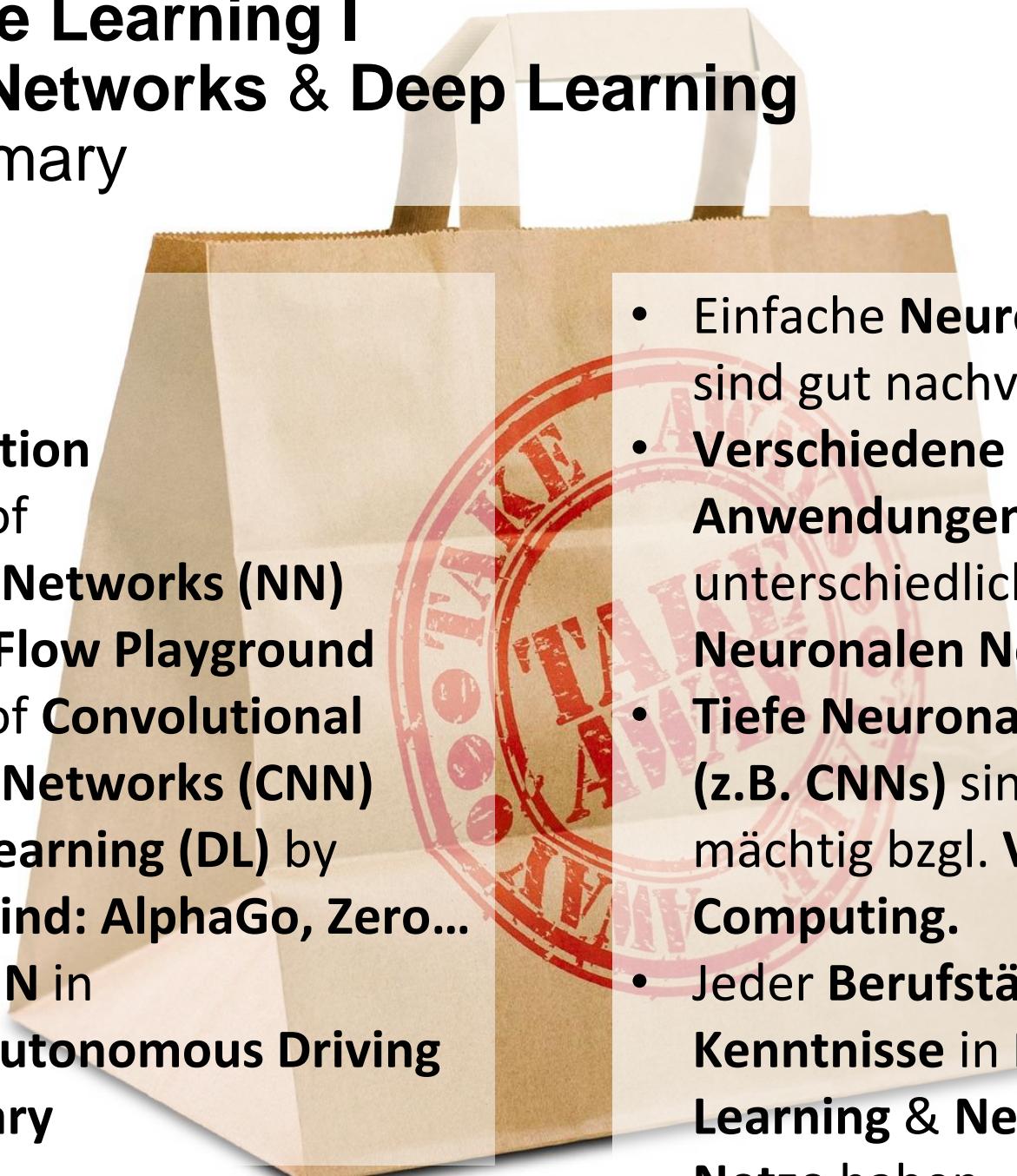
- Neural Networks & Deep Learning

(7) Summary

Content:

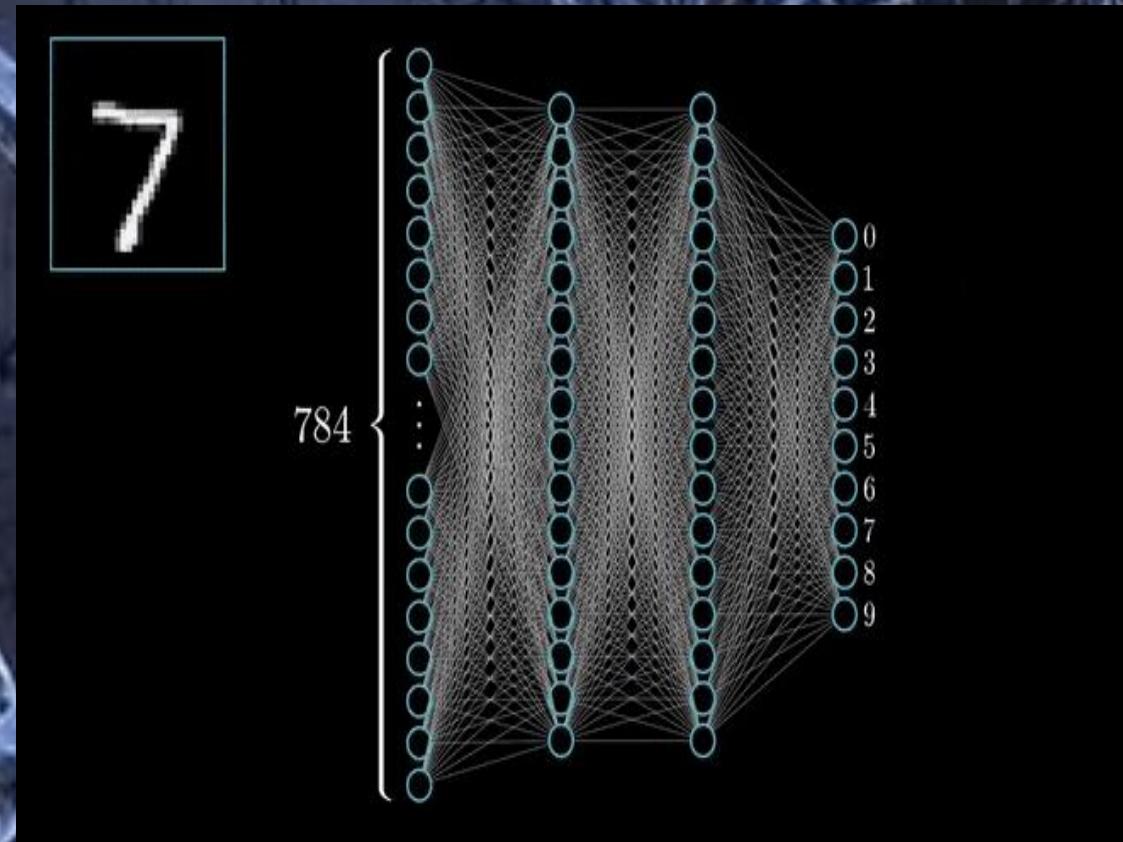
1. Motivation
2. Basics of Neural Networks (NN)
3. TensorFlow Playground
4. Basics of Convolutional Neural Networks (CNN)
5. Deep Learning (DL) by Deepmind: AlphaGo, Zero...
6. Deep NN in Tesla Autonomous Driving
7. Summary

- Einfache **Neuronale Netze** sind gut nachvollziehbar.
- Verschiedene Anwendungen benötigen unterschiedliche **Neuronale Netze**.
- Tiefe **Neuronale Netze** (z.B. CNNs) sind sehr mächtig bzgl. **Visual Computing**.
- Jeder **Berufstätige** sollte Kenntnisse in **Machine Learning & Neuronale Netze** haben.



8 Machine Learning I

- Neural Networks & Deep Learning (2) Basics of Neural Networks (NN)



MIT Introduction to Deep Learning 6.S191

The screenshot shows a YouTube video player. The main content area displays a diagram of a deep learning architecture. It starts with an 'INPUT' image of a car key. This is followed by a sequence of operations: 'CONVOLUTION + RELU', 'POOLING', 'CONVOLUTION + RELU', and another 'POOLING'. These steps are grouped under the heading 'FEATURE LEARNING'. The output of these features is then processed through a 'FLATTEN' layer, followed by a 'FULLY CONNECTED' layer, and finally a 'SOFTMAX' layer to produce classification results for 'CAR', 'TRUCK', 'VAN', and 'BICYCLE'. Below the diagram, applications of this architecture are listed: 'Detection', 'Semantic segmentation', and 'End-to-end robotic control'. To the right of the diagram, there is a video frame showing a man (Alexander Amini) speaking at a podium. The video player interface includes a progress bar (29:47 / 37:20), a 'Applications' link, and various interaction buttons like 'ABONNIEREN' (Subscribe) and 'CHATWIEDERGABE ANZEIGEN' (Show Chat Replay). The MIT Deep Learning logo is visible in the top right corner of the video frame.

An Architecture for Many Applications

INPUT CONVOLUTION + RELU POOLING CONVOLUTION + RELU POOLING

FEATURE LEARNING

Detection
Semantic segmentation
End-to-end robotic control

CLASSIFICATION

CAR TRUCK VAN
BICYCLE

MIT Deep Learning

IntroToDeepLearning.com

Massachusetts Institute of Technology

6.S191 Introduction to Deep Learning
introtodeeplearning.com @MITDeepLearning

1/28/20

29:47 / 37:20 • Applications >

Alexander Amini
86.200 Abonnenten

ABONNIEREN

CHATWIEDERGABE ANZEIGEN

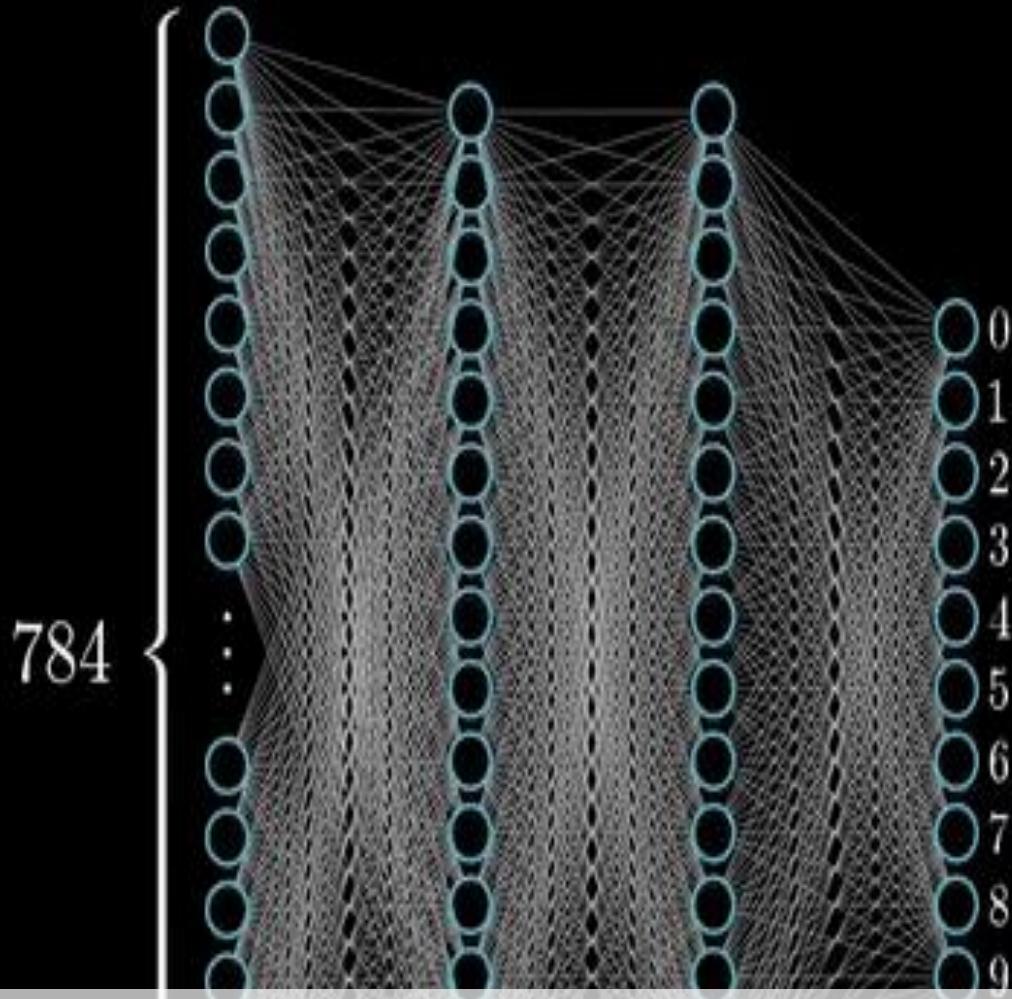
Text im Video suchen

3128 24 TEILEN SPEICHERN ...

MIT Introduction to Deep Learning 6.S191: Lecture 3

109

7



Machine Learning I – Neural Networks & Deep Learning

Michael Amberg

Todays Content:

- 1. Motivation**
- 2. Basics of Neural Networks (NN)**
- 3. TensorFlow Playground**
- 4. Basics of Convolutional Neural Networks (CNN)**
- 5. Deep Learning (DL) by Deepmind: AlphaGo, Zero...**
- 6. Deep NN in Tesla Autonomous Driving**
- 7. Summary**



Artificial Intelligence (AI)

The **theory and development** of **computer systems** able to **perform tasks normally requiring human intelligence**, such as visual perception, speech recognition, decision-making, and translation between languages.

Machine Learning (ML)

The **use and development** of **computer systems** that can **learn and adapt without following explicit instructions**, by using **algorithms** and **statistical models** to **analyze** and **draw inferences** from patterns in data.

Neural Networks (NN)

Use a **network of functions** to **understand** and **translate** a **data input** of one **form** into a **desired output**. Neural Networks are one **approach to machine learning** that **learn a representation by themselves** and can **vary in depth**.

Deep Learning (DL)

A **subset of machine learning** **based on neural networks** in which **multiple layers** of processing are used to **extract progressively higher-level features** from **data**. Without neural networks, there would be no deep learning. But deep learning may comprise other techniques from machine learning.



Artificial intelligence



From Wikipedia, the free encyclopedia

"AI" redirects here. For other uses, see [AI \(disambiguation\)](#) and [Artificial intelligence \(disambiguation\)](#).

Artificial intelligence (AI), is intelligence demonstrated by machines, unlike the **natural intelligence** displayed by **humans and animals**, which involves consciousness and emotionality. The distinction between the former and the latter categories is often revealed by the acronym chosen. 'Strong' AI is usually labelled as AGI (Artificial General Intelligence) while attempts to emulate 'natural' intelligence have been called ABI (Artificial Biological Intelligence). Leading AI textbooks define the field as the study of "intelligent agents": any device that perceives its environment and takes actions that maximize its chance of successfully achieving its goals.^[3] Colloquially, the term "artificial intelligence" is often used to describe machines (or computers) that mimic "cognitive" functions that humans associate with the **human mind**, such as "learning" and "problem solving".^[4]

As machines become increasingly capable, tasks considered to require "intelligence" are often removed from the definition of AI, a phenomenon known as the **AI effect**.^[5] A quip in Tesler's Theorem says "AI is whatever hasn't been done yet."^[6] For instance, **optical character recognition** is frequently excluded from things considered to be AI,^[7] having become a routine technology.^[8] Modern machine capabilities generally classified as AI include successfully **understanding human speech**,^[9] competing at the highest level in **strategic game systems** (such as **chess** and **Go**),^[10] autonomously operating cars, intelligent routing in content delivery networks, and **military simulations**.^[11]

Artificial intelligence was founded as an academic discipline in 1955, and in the years since has experienced several waves of optimism,^{[12][13]} followed by disappointment and the loss of funding (known as an "**AI winter**").^{[14][15]} followed by new approaches, success and renewed funding.^{[13][16]} After **AlphaGo** successfully defeated a professional Go player in 2015, artificial intelligence once again attracted widespread global attention.^[17] For most of its history, AI research has been divided into sub-fields that often fail to communicate with each other.^[18] These sub-fields are based on technical considerations, such as particular goals (e.g. "**robotics**" or "**machine learning**"),^[19] the use of particular tools ("**logic**" or **artificial neural networks**), or deep philosophical differences.^{[22][23][24]} Sub-fields have also been based on social factors (particular institutions or the work of particular researchers).^[18]

The traditional problems (or goals) of AI research include **reasoning**, **knowledge representation**, **planning**, **learning**, **natural language processing**, **perception** and the ability to move and manipulate objects.^[19] **General intelligence** is among the field's long-term goals.^[25] Approaches include **statistical methods**, **computational intelligence**, and **traditional symbolic AI**. Many tools are used in AI, including versions of **search** and **mathematical optimization**, **artificial neural networks**, and methods based on **statistics**, **probability** and **economics**. The AI field draws upon **computer science**, **information engineering**, **mathematics**, **psychology**, **linguistics**, **philosophy**, and many other fields.

Part of a series on

Artificial intelligence

Major goals	[show]
Approaches	[show]
Philosophy	[show]
History	[show]
Technology	[show]
Glossary	[show]

V · T · E

Main page

Contents

Current events

Random article

About Wikipedia

Contact us

Donate

Contribute

Help

Learn to edit

Community portal

Recent changes

Upload file

Tools

What links here

Related changes

Special pages

Permanent link

Page information

Cite this page

Wikidata item

Print/export

Download as PDF

Printable version

In other projects

Wikimedia Commons

Wikibooks

Wikiquotes



Deep learning

From Wikipedia, the free encyclopedia

Deep learning (also known as **deep structured learning**) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised.^{[1][2][3]}

Deep-learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, machine vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance.^{[4][5][6]}

Artificial neural networks (ANNs) were inspired by information processing and distributed communication nodes in **biological systems**. ANNs have various differences from biological brains. Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analog.^{[7][8][9]}

The adjective "deep" in deep learning comes from the use of multiple layers in the network. Early work showed that a linear **perceptron** cannot be a universal classifier, and then that a network with a nonpolynomial activation function with one hidden layer of unbounded width can on the other hand so be. Deep learning is a modern variation which is concerned with an unbounded number of layers of bounded size, which permits practical application and optimized implementation, while retaining theoretical universality under mild conditions. In deep learning the layers are also permitted to be heterogeneous and to deviate widely from biologically informed **connectionist** models, for the sake of efficiency, trainability and understandability, whence the "structured" part.

Part of a series on
Machine learning
and
data mining

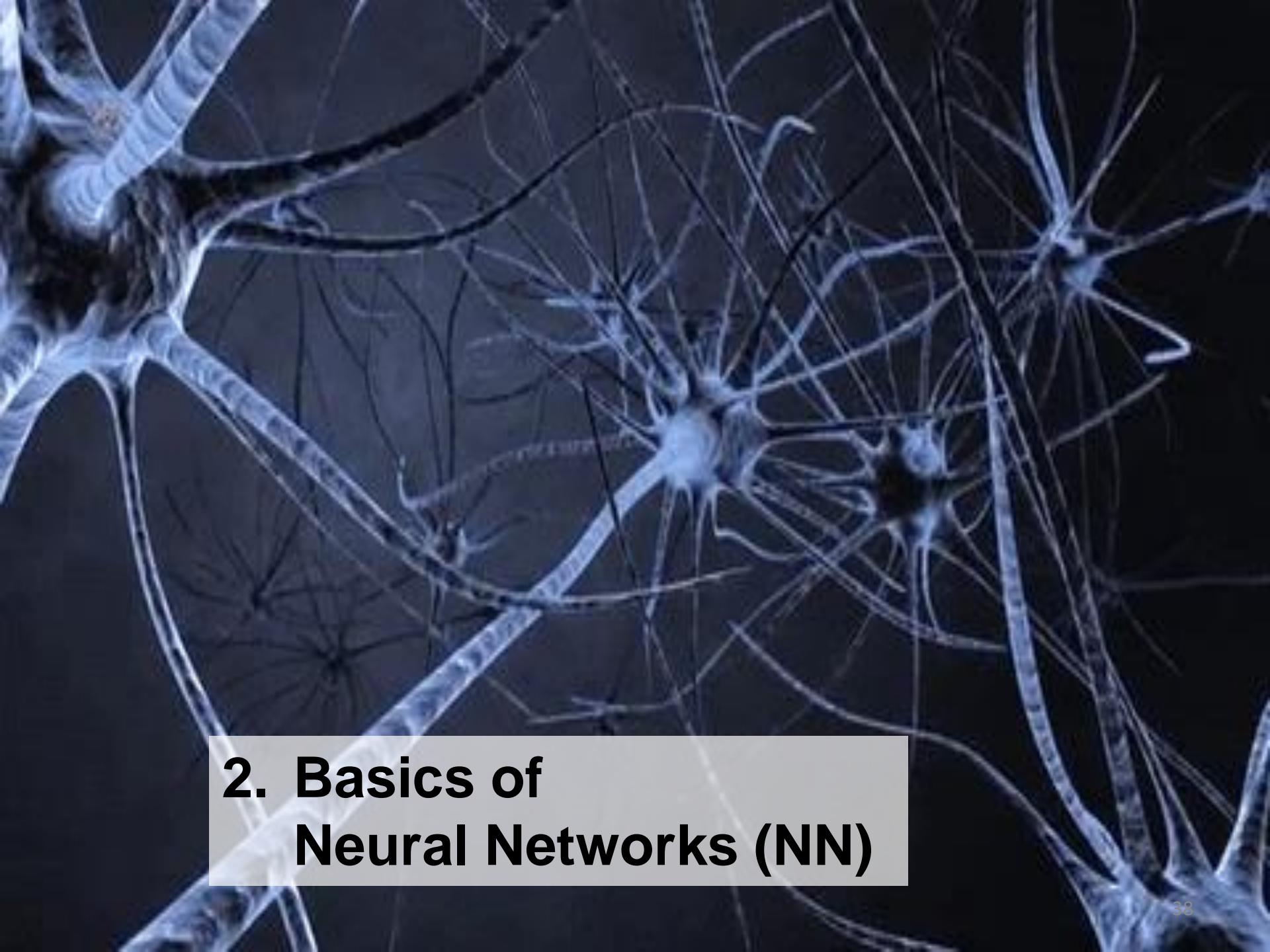
Problems	[show]
Supervised learning	[show]
(classification • regression)	
Clustering	[show]
Dimensionality reduction	[show]
Structured prediction	[show]
Anomaly detection	[show]
Artificial neural network	[show]
Reinforcement learning	[show]
Theory	[show]
Machine-learning venues	[show]
Glossary of artificial intelligence	[show]
Related articles	[show]

V · T · E

Part of a series on
Artificial intelligence

Major goals	[show]
--------------------	------------------------

V · T · E



2. Basics of Neural Networks (NN)

Extraktion und Selektion von Merkmalen in *RapidMiner Studio*

The screenshot shows the RapidMiner Studio interface with the following components:

- Repository:** Contains "Training Resources (connected)", "Samples", "processes" (including "Titanic" and "Toyota"), "Templates", "Time Series", "Tutorials", and "Local Repository (Legacy)".
- Operators:** Shows categories like "Data Access (53)", "Blending (82)", "Cleansing (29)", "Modeling (166)", "Scoring (14)", "Validation (30)", "Utility (85)", and "Extensions (2)".
- Process View:** Displays a process flow:
 - "Retrieve Toyota" (blue icon) has an output port "out" connected to the "Select Attributes" operator.
 - "Select Attributes" (purple icon) has three output ports: "exa", "exa", and "on".
 - "Set Role" (purple icon) has two output ports: "exa" and "on".
 - "Split Data" (purple icon) has three output ports: "exa", "par", and "par".
 - "Decision Tree" (green icon) receives input from "Split Data" and has three output ports: "tra", "mod", and "wei".
 - "Apply Model" (green icon) receives input from "Decision Tree" and has two output ports: "mod" and "lab".
 - "Performance" (yellow icon) receives input from "Apply Model" and has four output ports: "lab", "per", "per", and "exa".
- Parameters View:** Shows process parameters like "logverbosity: init", "logfile", "resultfile", "random seed: 2001", "send mail: never", and "encoding: SYSTEM".
- Recommended Operators:** A list of operators with their compatibility status:
 - Retrieve: 12%
 - Select Attributes: 6%
 - Set Role: 6%
 - Apply Model: 4%
 - Filter Examples: 4%

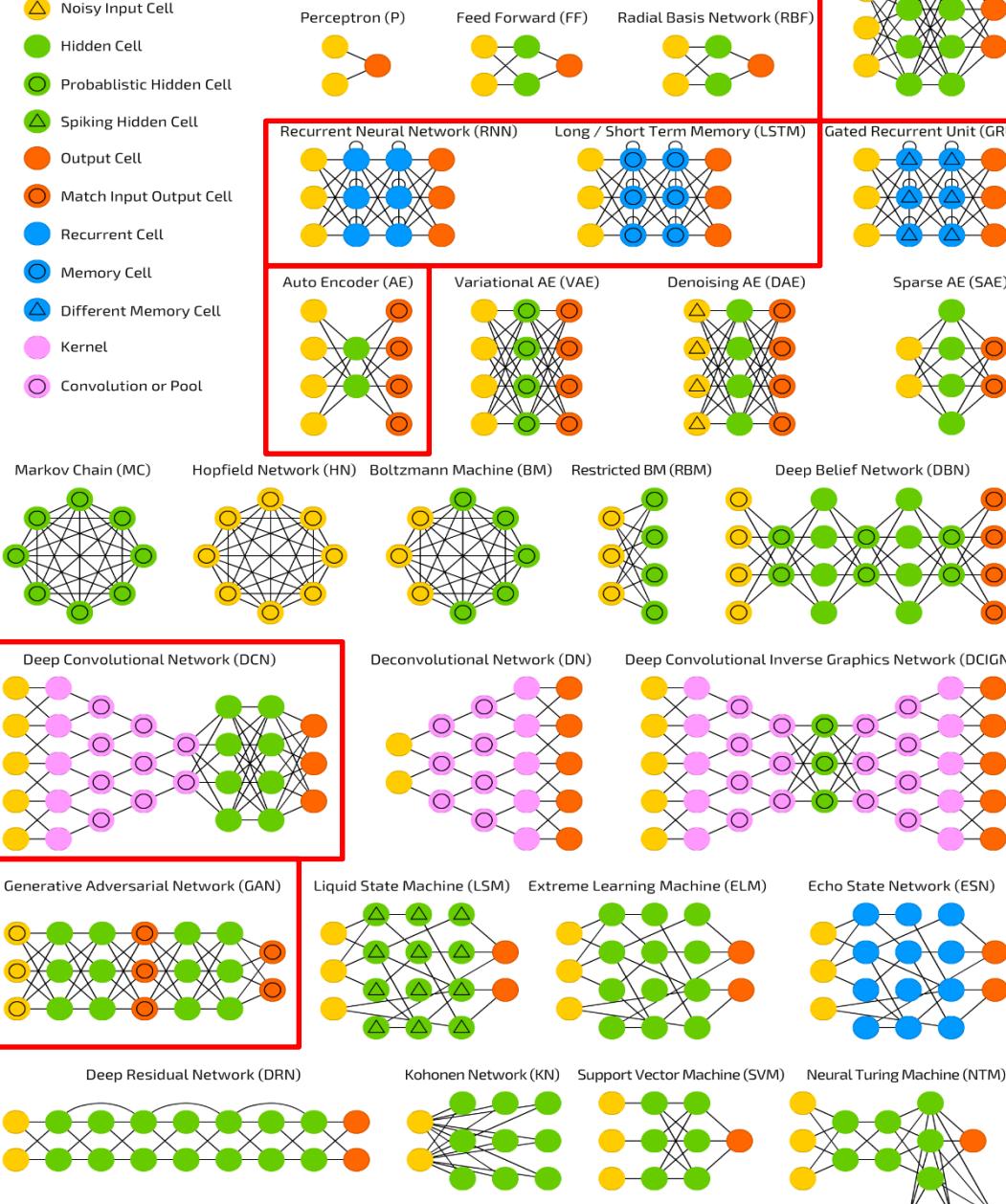
Text Labels:

- Entscheidungsbaum als Beispiel für eine klassische Methode des Maschine Learning**
- Auswahl der unabhängigen Merkmale (Features) entfällt bei der Verwendung von Neural Networks.**

Neural Networks

©2016 Fjodor van Veen - www.asimovinstitute.org

- Backfed Input Cell
- Input Cell
- △ Noisy Input Cell
- Hidden Cell
- Probabilistic Hidden Cell
- △ Spiking Hidden Cell
- Output Cell
- Match Input Output Cell
- Recurrent Cell
- Memory Cell
- △ Different Memory Cell
- Kernel
- Convolution or Pool



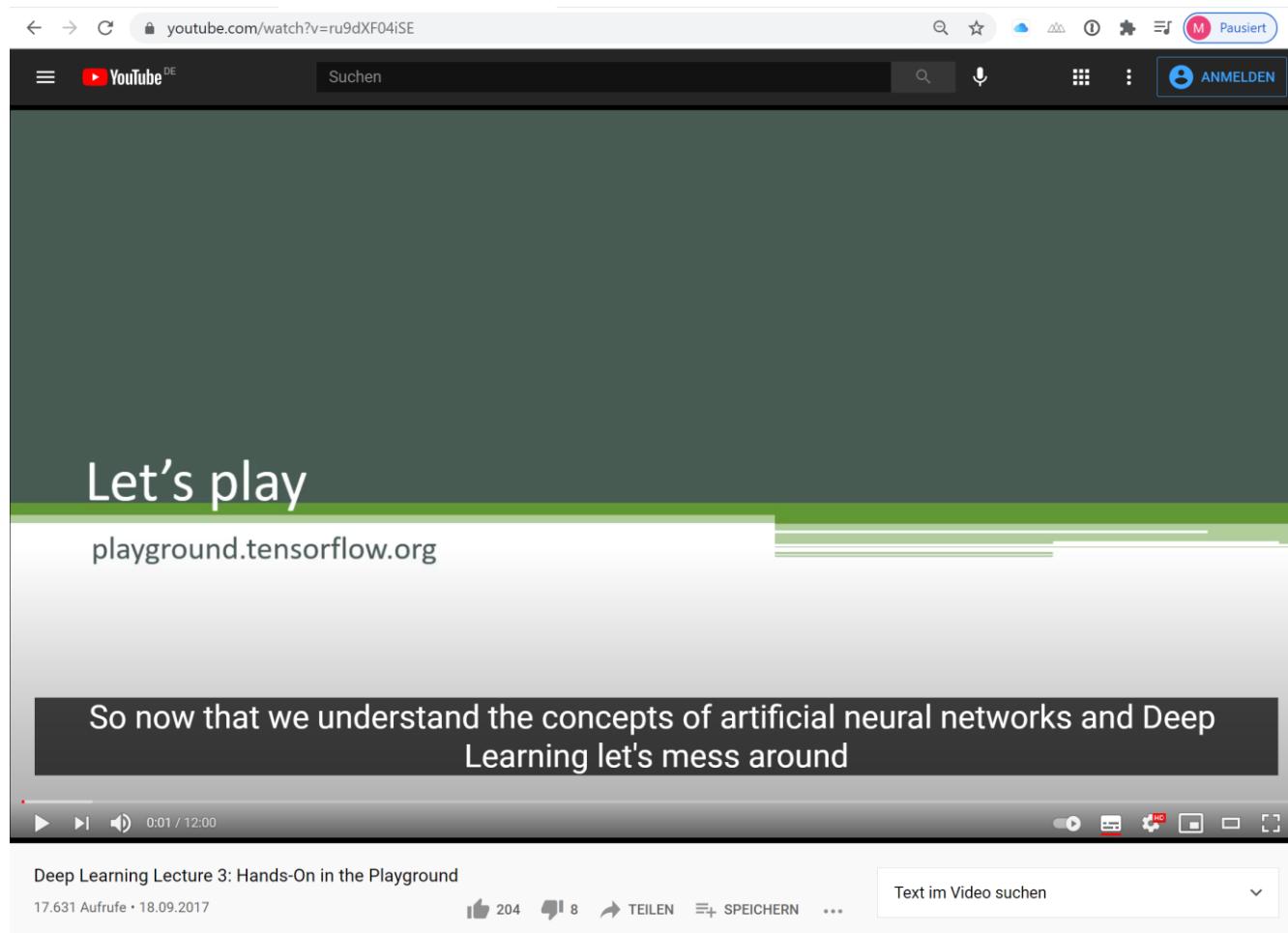
Neuronale Netze stellen einen zentralen Baustein für Deep Learning dar.

Im Laufe der Jahre wurden zahlreiche **Architekturen** für Neuronale Netze entwickelt.

Die Wahl der **Architektur** hängt von der **Datenstruktur**, den **Dateninhalten** und der **Aufgabenstellung** ab.

Aktuell wichtige Klassen sind u.a. **Convolutional Neural Networks (CNN)**, **Recurrent Neural Networks (RNN)**, **Autoencoder**, **Generative Adversarial Networks (GAN)**.

TensorFlow Playground Tutorial



Deep Learning Lecture 3: Hands-On in the Playground 2017 (12min)

<https://www.youtube.com/watch?v=ru9dXF04iSE>

TensorFlow Playground Help

The screenshot shows the TensorFlow Playground interface with the following configuration:

- Epoch: 000,000
- Learning rate: 0.03
- Activation: Tanh
- Regularization: None
- Regularization rate: 0
- Problem type: Classification

The interface includes sections for DATA, FEATURES, and OUTPUT.

DATA: Which dataset did you feed in? Lex Fridman: Neural networks learning spirals 2020. www.youtube.com/watch?v=i3ZnDRrmFjg

FEATURES: 2 HIDDEN LAYERS. 2 neurons. The outputs are mixed with varying weights, shown by the colors of the lines. This is the output from one neuron. Hover to see it larger.

OUTPUT: Test loss 0.502. Training loss 0.502. A scatter plot shows blue and orange data points separated by a decision boundary.

Bottom controls: REGENERATE, sin(X_1), sin(X_2), Colors shows data, neuron and weight values, -1, 0, 1, Show test data, Discretize output.

Machine Learning Crash Course – A self-study Guide

The screenshot shows a web browser displaying the Google Machine Learning Crash Course website at developers.google.com/machine-learning/crash-course/introduction-to-neural-networks/playground-exercises. The page title is "Neural Networks: Playground Exercises". The left sidebar, titled "Crash Course", contains a navigation menu with several sections, each with a red horizontal bar underneath it. The sections include "Quick Links", "ML Concepts", "Neural Networks", "Playground Exercises" (which is currently selected and highlighted in blue), and "ML Engineering". The main content area displays the "Neural Networks: Playground Exercises" page, which includes a "Google is committed to advancing racial equity for Black communities. See how." banner, a breadcrumb navigation (Home > Products > Machine Learning > Courses), and a "Rate and review" button with thumbs up and down icons. Below the main title, there is a "Send feedback" button. A callout box indicates an "Estimated Time: 20 minutes". The main content starts with a section titled "A First Neural Network" and describes the task of training a neural network to learn nonlinear models without explicit feature crosses. It then lists four tasks for experimenting with neural networks, including increasing the number of neurons, changing activation functions like ReLU, and testing model quality.

Machine Learning Crash Course

Courses Practica Guides Glossary

Search

Crash Course Problem Framing Data Prep Clustering Recommendation Testing and Debugging GANs

Quick Links

- Overview
- Prerequisites and Prework
- Exercises

ML Concepts

- Introduction to ML (3 min)
- Framing (15 min)
- Descending into ML (20 min)
- Reducing Loss (60 min)
- First Steps with TF (65 min)
- Generalization (15 min)
- Training and Test Sets (25 min)
- Validation Set (35 min)
- Representation (35 min)
- Feature Crosses (70 min)
- Regularization: Simplicity (40 min)
- Logistic Regression (20 min)
- Classification (90 min)
- Regularization: Sparsity (20 min)

Neural Networks (65 min)

- Video Lecture
- Structure
- Playground Exercises

- Programming Exercise
- Training Neural Nets (10 min)
- Multi-Class Neural Nets (45 min)
- Embeddings (50 min)

ML Engineering

Google is committed to advancing racial equity for Black communities. [See how.](#)

Home > Products > Machine Learning > Courses

Rate and review

[Send feedback](#)

Estimated Time: 20 minutes

Neural Networks: Playground Exercises

A First Neural Network

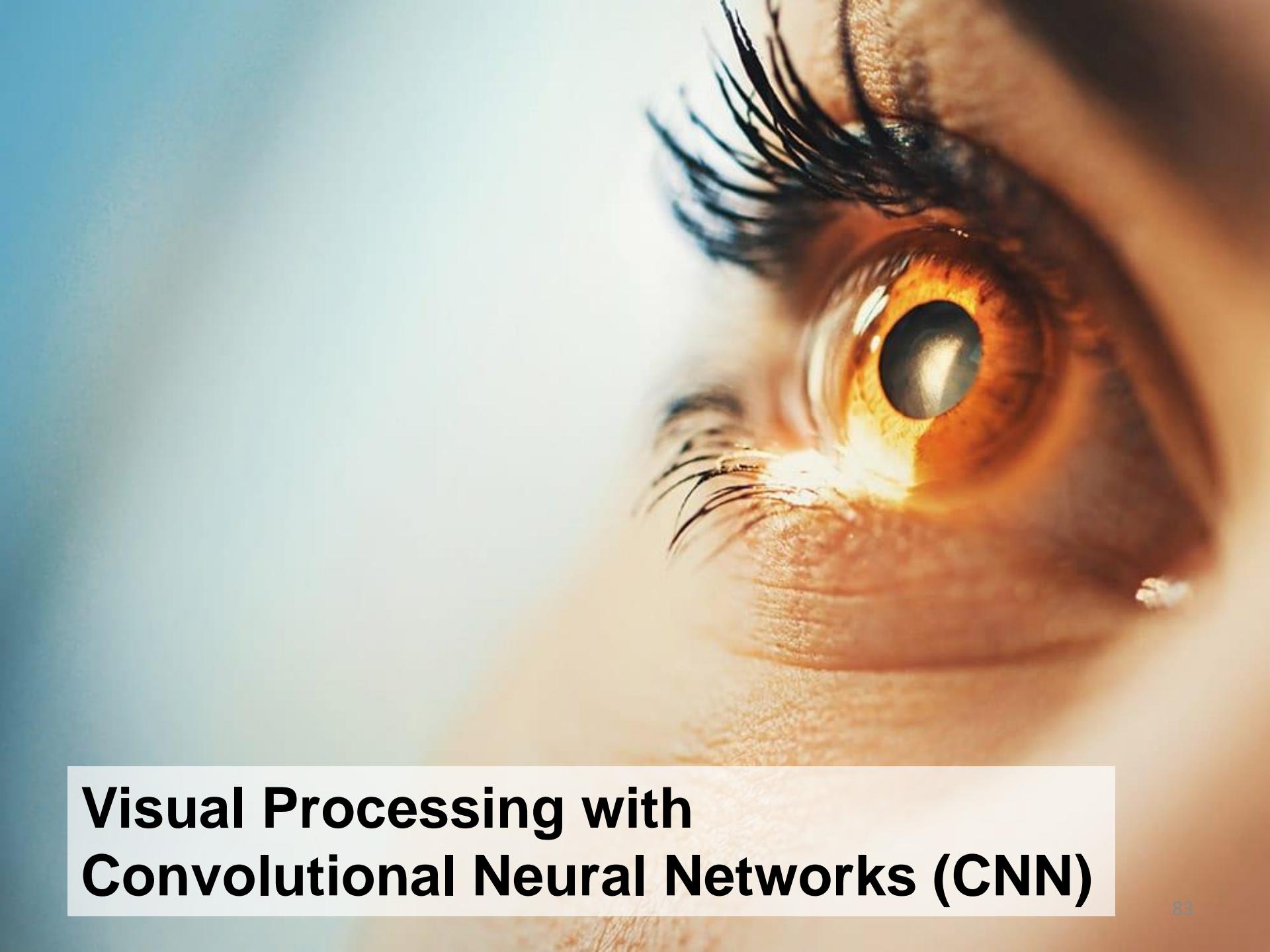
In this exercise, we will train our first little neural net. Neural nets will give us a way to learn nonlinear models without the use of explicit feature crosses.

Task 1: The model as given combines our two input features into a single neuron. Will this model learn any nonlinearities? Run it to confirm your guess.

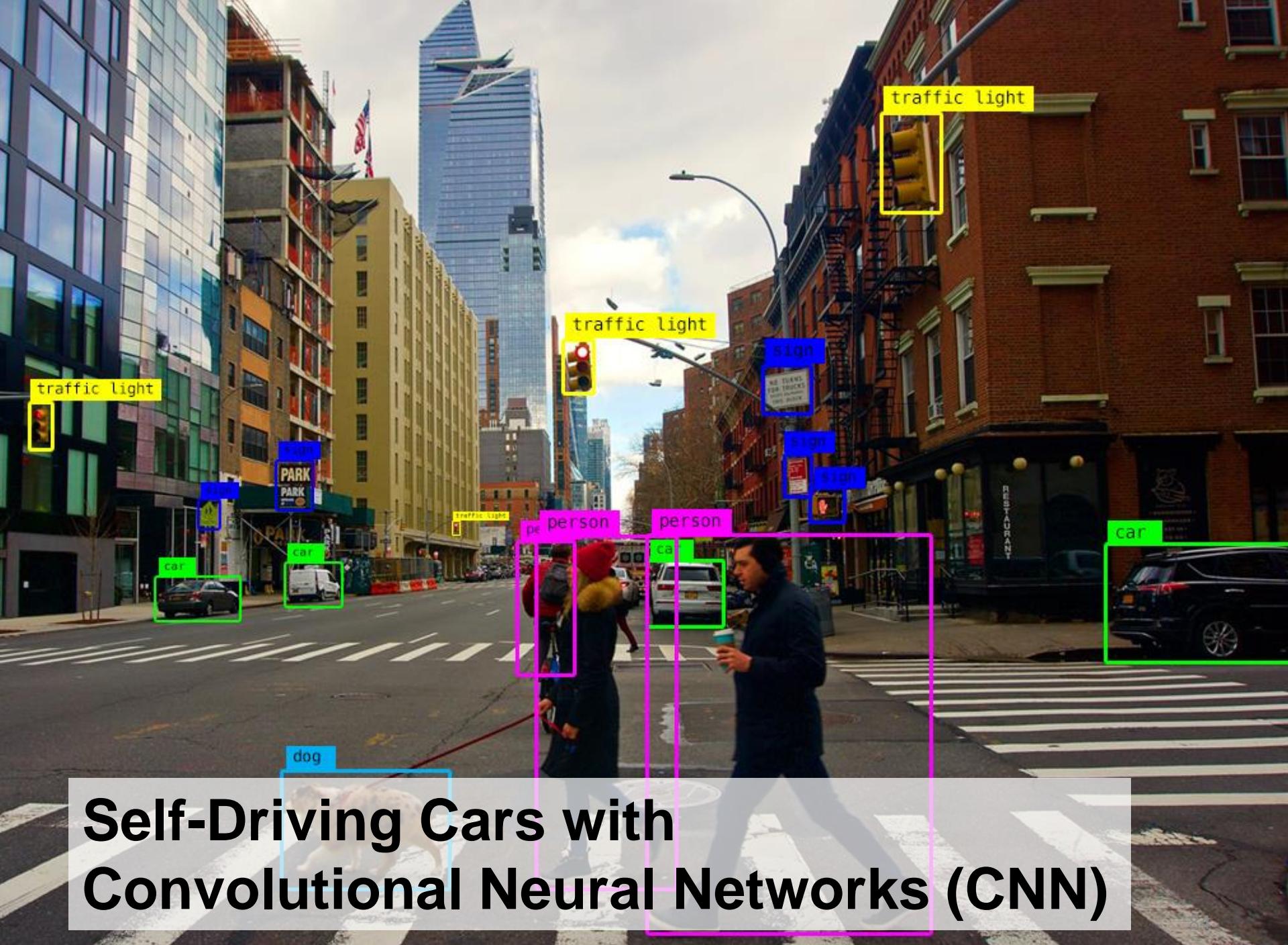
Task 2: Try increasing the number of neurons in the hidden layer from 1 to 2, and also try changing from a Linear activation to a nonlinear activation like ReLU. Can you create a model that can learn nonlinearities? Can it model the data effectively?

Task 3: Try increasing the number of neurons in the hidden layer from 2 to 3, using a nonlinear activation like ReLU. Can it model the data effectively? How model quality vary from run to run?

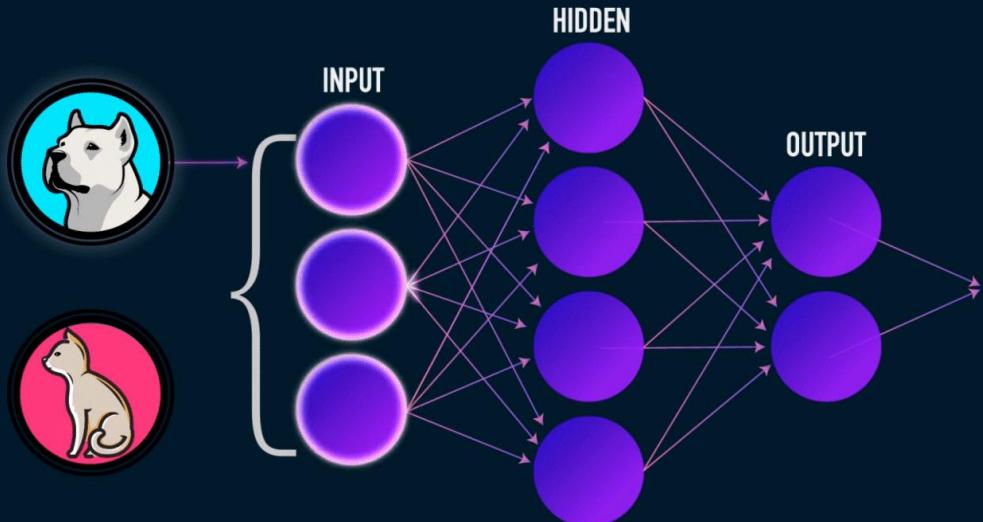
Task 4: Continue experimenting by adding or removing hidden layers and neurons per layer. Also feel free to change learning rates, regularization, and other learning settings. What is the *smallest* number of neurons and layers you can use that gives test loss of 0.177 or lower?



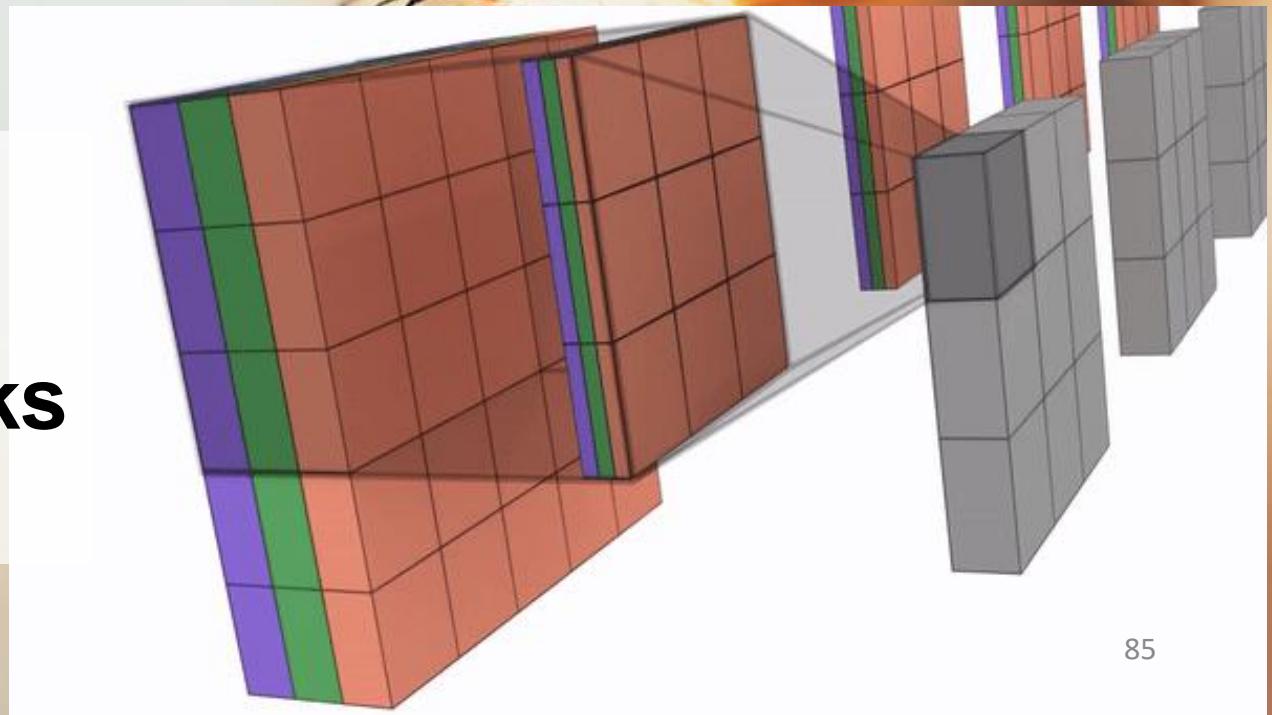
Visual Processing with Convolutional Neural Networks (CNN)



Self-Driving Cars with Convolutional Neural Networks (CNN)

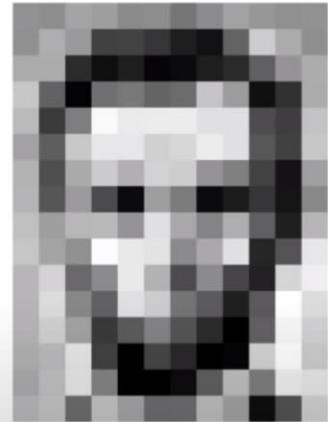


Basics of Convolutional Neural Networks (CNN)



CNN: Grundlegende Problemstellung (1/3)

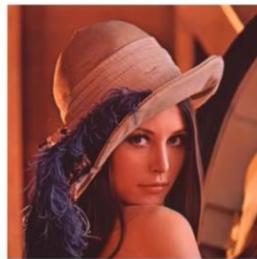
Was Computer sehen: Ein Bild ist eine Folge von Bildpunkten (Pixel).



What the computer sees															
157	153	174	168	160	162	129	151	172	161	155	156	155	182	163	74
156	182	163	74	75	62	93	17	110	210	180	154	180	180	50	14
180	180	50	14	34	6	10	33	48	106	159	181	206	109	5	124
206	109	5	124	131	111	120	204	166	15	56	180	194	68	137	251
194	68	137	251	237	239	239	228	227	87	71	201	172	106	207	233
172	106	207	233	233	214	220	239	228	98	74	206	188	88	179	209
188	88	179	209	185	215	211	158	139	75	20	169	189	97	165	84
189	97	165	84	10	168	134	11	31	62	22	148	199	168	191	193
199	168	191	193	158	227	178	143	182	106	36	190	205	174	195	252
205	174	155	252	236	231	149	178	228	43	95	234	190	216	116	149
190	216	116	149	236	187	84	150	79	38	218	241	190	224	147	108
190	224	147	108	227	210	127	102	35	101	255	224	190	214	173	66
190	214	173	66	103	143	95	50	2	109	249	215	187	196	235	75
187	196	235	75	1	81	47	0	6	217	255	211	183	202	237	145
183	202	237	145	0	0	12	108	200	138	243	236	195	206	123	207
195	206	123	207	177	121	123	200	175	13	96	218				

An image is just a matrix of numbers [0,255]!
i.e., 1080x1080x3 for an RGB image

Let's identify key features in each image category



Nose,
Eyes,
Mouth



Wheels,
License Plate,
Headlights



Door,
Windows,
Steps

CNN: Grundlegende Problemstellung (2/3)

Eine Objekterkennung in Bildern ist nicht einfach,
da das Umfeld und weitere Rahmenbedingungen dies erschweren.

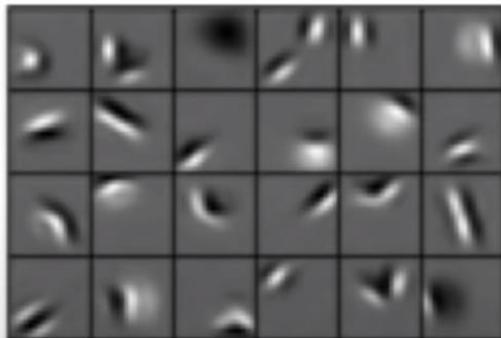


CNN: Grundlegende Problemstellung (3/3)

Kann ein **Computer** die **relevanten Aspekte (Features)** zur **Objekterkennung** aus **Daten** **selbstständig erlernen**, ohne sich von den Rahmenbedingungen „**ablenken**“ zu lassen?

Can we learn a **hierarchy of features** directly from the data instead of hand engineering?

Low level features



Edges, dark spots

Mid level features



Eyes, ears, nose

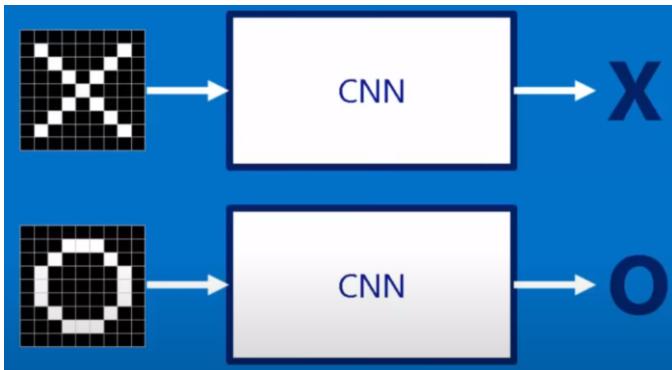
High level features



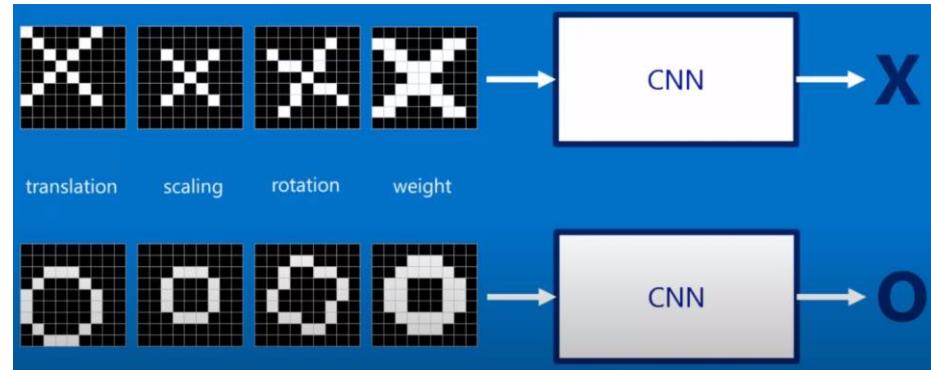
Facial structure

CNN: Kernidee am Beispiel Objekterkennung (1/5)

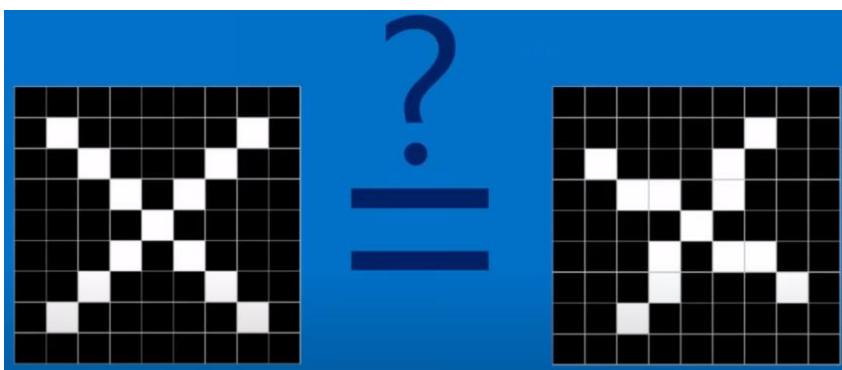
Problemstellung



Herausforderung



Problem



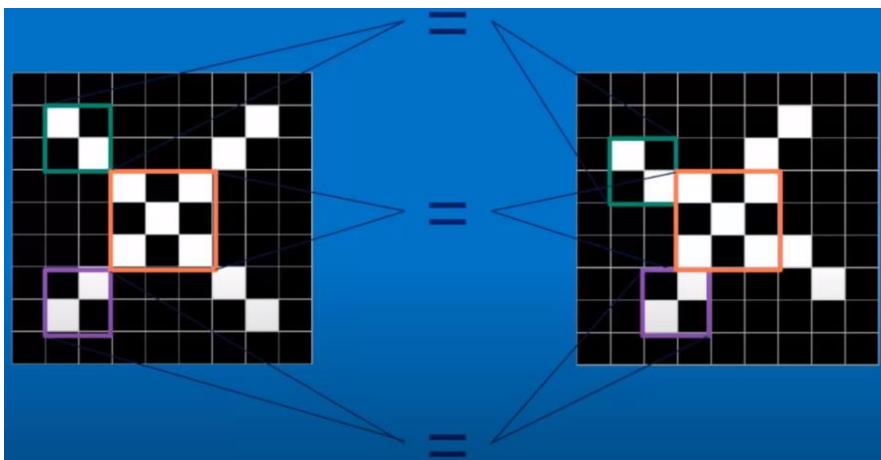
Problem (Computerdarstellung)

This diagram provides a numerical representation of the convolution process. The input image is a 4x4 matrix of -1s and 1s. The kernel is a 3x3 matrix with a central value of 1. The output is a 2x2 matrix where the central element is highlighted in blue, representing the result of the convolution step.

-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	1	-1
-1	-1	1	-1	-1	-1	1	-1	-1
-1	-1	-1	1	-1	1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	-1	-1	1	-1	-1	-1
-1	-1	-1	1	-1	-1	1	-1	-1
-1	1	-1	-1	-1	-1	1	-1	-1
-1	-1	-1	-1	-1	-1	-1	1	-1

CNN: Kernidee am Beispiel Objekterkennung (2/5)

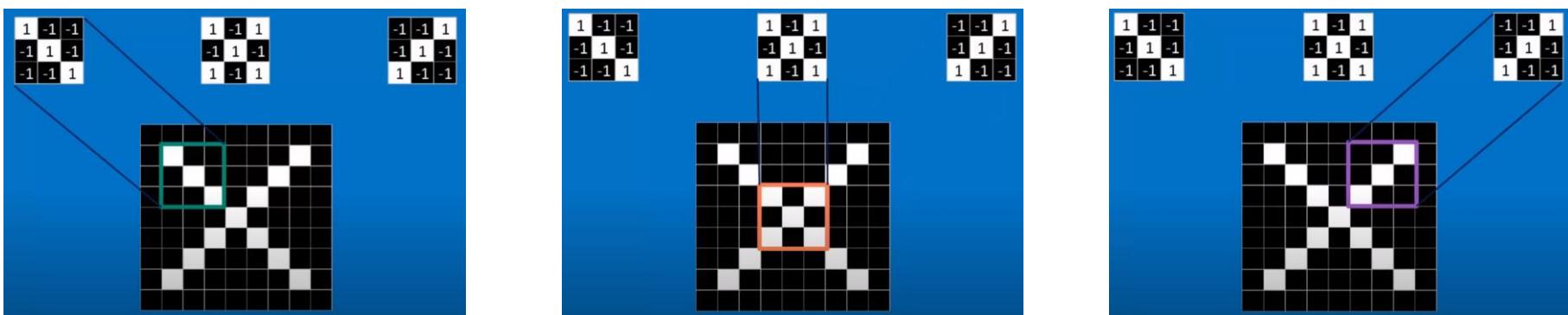
Idee: Features (Filter) helfen



Relevante Features (Filter) identifizieren

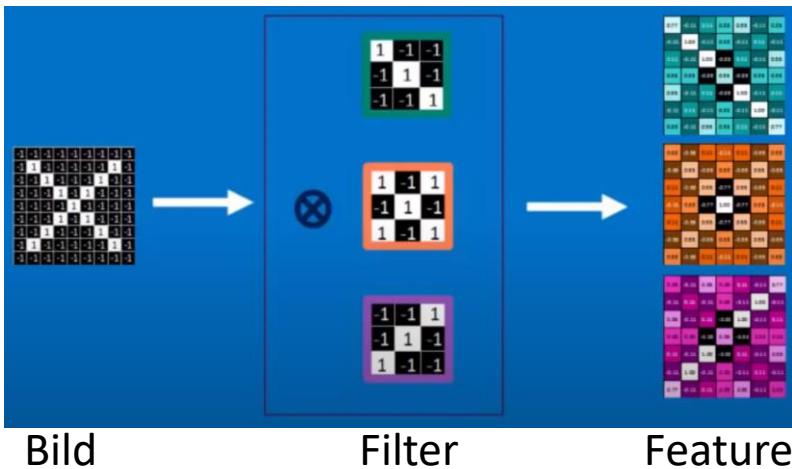
$\begin{matrix} 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{matrix}$	$\begin{matrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{matrix}$	$\begin{matrix} -1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & -1 \end{matrix}$
---	---	---

Features (Filter) anwenden: Suche nach Features im Bild



CNN: Kernidee am Beispiel Objekterkennung (3/5)

Stapel von mehreren Features (Filtern)



Pooling (Komplexität reduzieren & ungenauer werden)



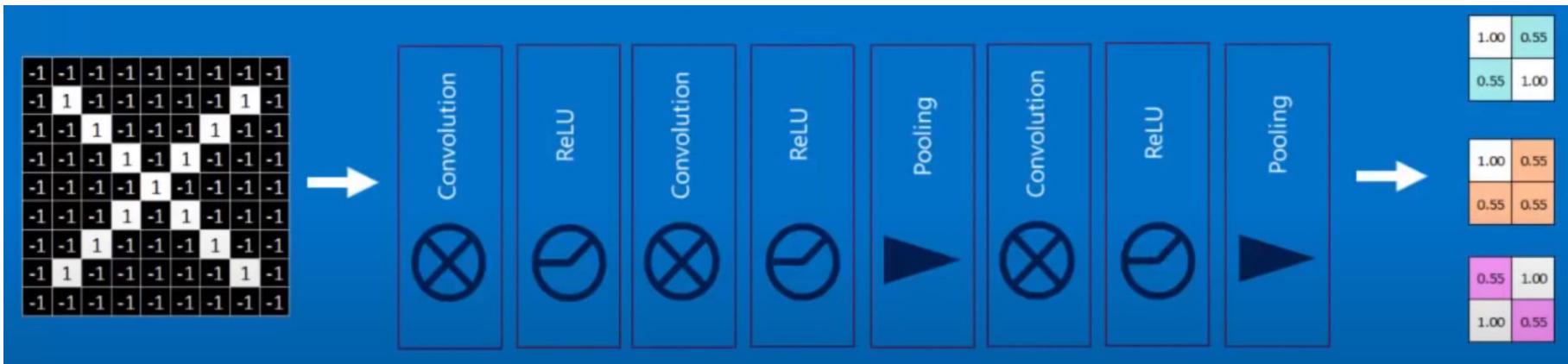
ReLU (Unwichtiges entfernen)



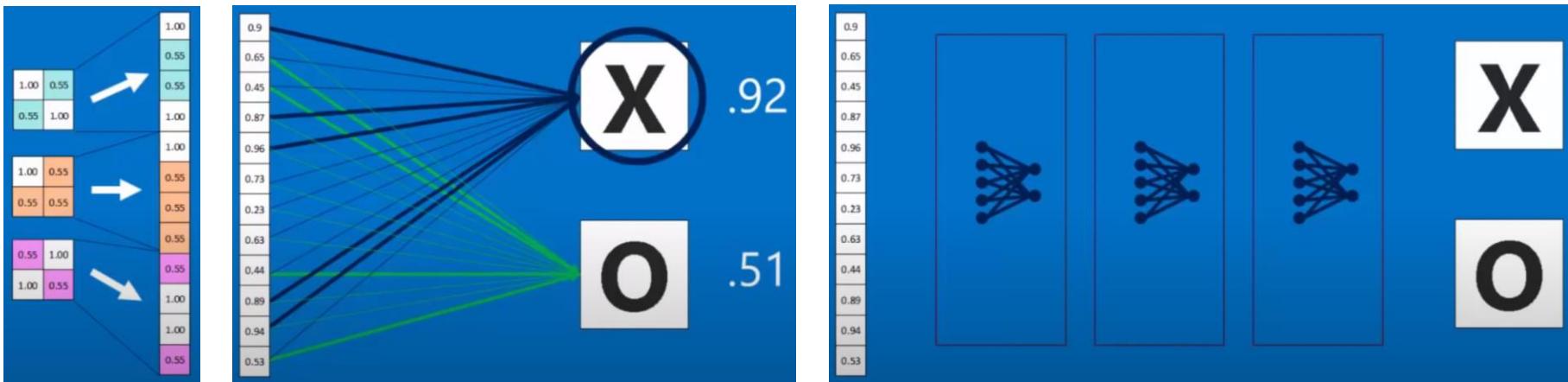
CNN: Kernidee am Beispiel Objekterkennung (4/5)

Diese Operationen können mehrfach wiederholt werden:

Features suchen (Conv./Falten), Ergebnis bereinigen (ReLU) und komprimieren (Pooling).



Ergebnis (die finalen Features) ebnen (engl. flatten) und bewerten (evtl. mehrstufig)



How Convolutional Neural Networks work, 2016

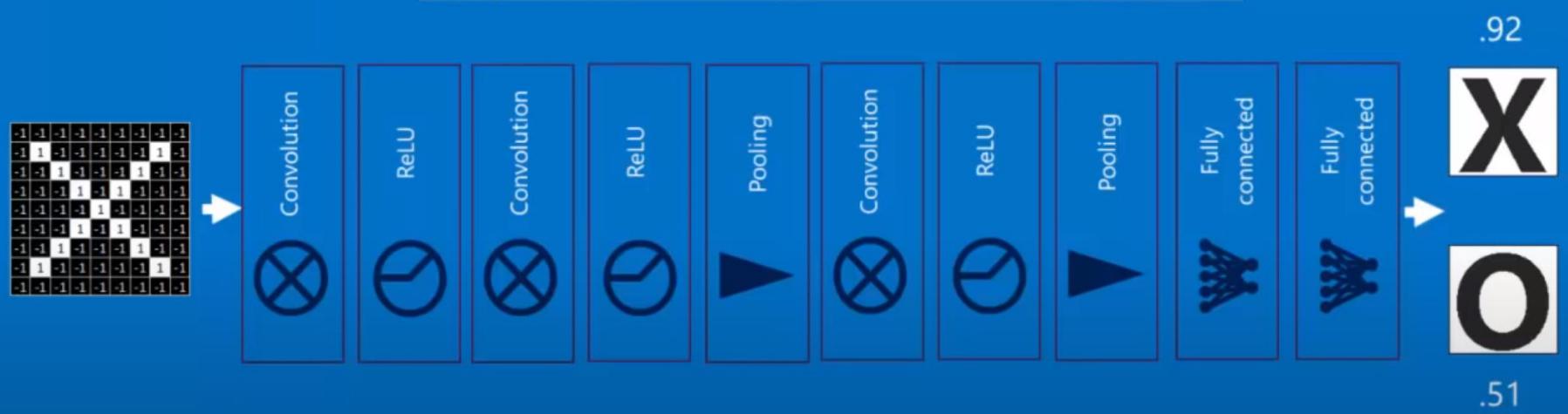
www.youtube.com/watch?v=FmpDlaiMleA

CNN: Kernidee am Beispiel Objekterkennung (5/5)

Backpropagation (Trainieren)

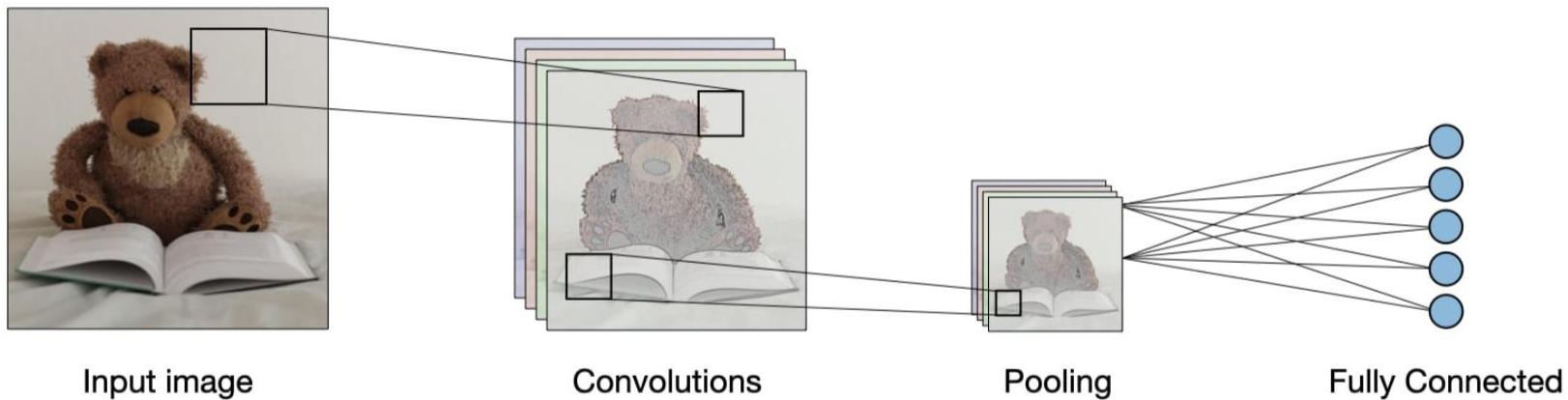
Backprop

	Right answer	Actual answer	Error
X	1	0.92	0.08
O	0	0.51	0.49



Zentrale Bausteine klassischer CNN am Beispiel

- **Architecture of a traditional CNN** — Convolutional neural networks, also known as CNNs, are a specific type of neural networks that are generally composed of the following layers:



The convolution layer and the pooling layer can be fine-tuned with respect to hyperparameters that are described in the next sections.

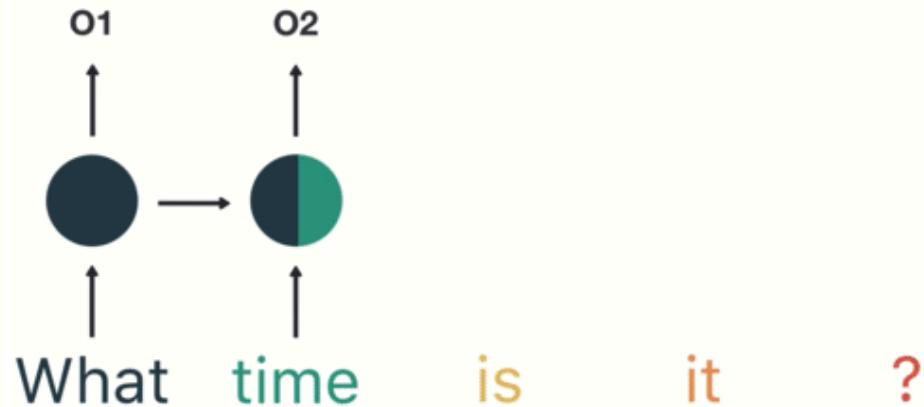
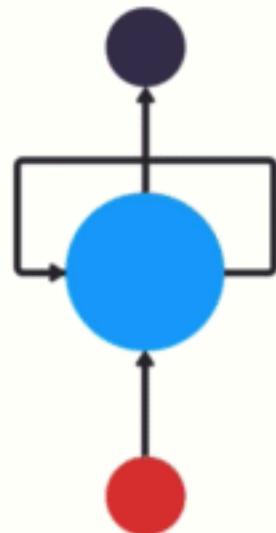
Recurrent Neural Networks (RNN)

Textanalyse mit einem RNN (Beispiel)

Output

Neuron

Input



Neuronen in RNNs haben jeweils einen Speicher (Internal Memory / Hidden State).

RNNs eignen sich gut für **kurzfristige Erinnerungen**,
weniger gut für **langfristige Erinnerungen** (Vanishing Gradient Problem).

OpenAI Microscope

OpenAI

OpenAI Microscope

We're introducing a collection of visualizations for layer and neuron of "neural organisms" which increase interpretability. Microscope analyzes the features learned by neural networks, and helps the research community understand them.

April 14, 2020
2 minute read

OpenAI Microscope
microscope.openai.com/models

Microscope

MODELS ABOUT

Models

AlexNet

The OpenAI Microscope is a collection of visualizations for significant neurons in Inception v1, VGG 19, Inception v3, Inception v4, and ResNet v2 50.

LEARN MORE

Inception v1

mixed4c

Unit 447

FEATURE VISUALIZATION

An artificial, optimized image that maximizes activations of the given unit. [Read more.](#)

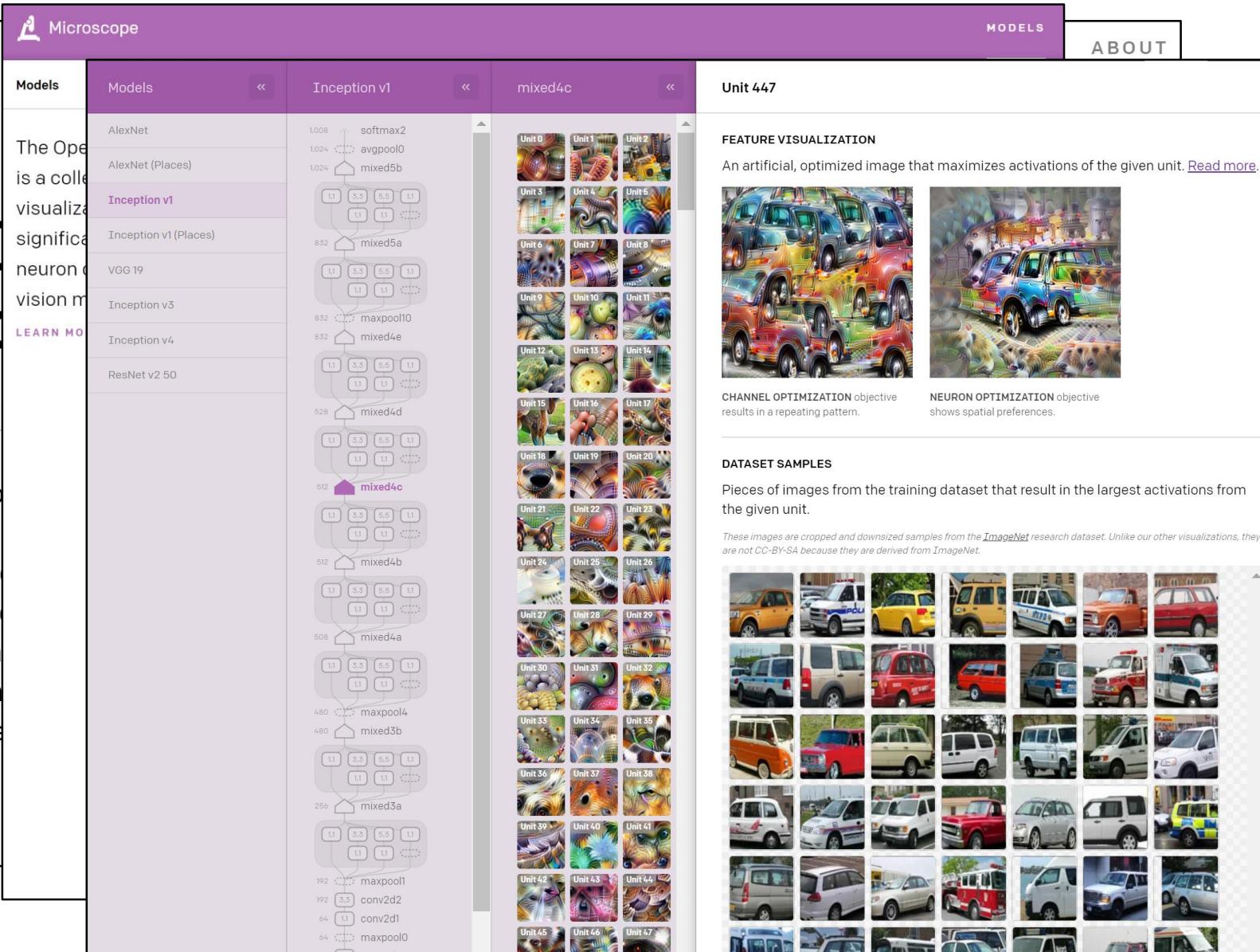
CHANNEL OPTIMIZATION objective results in a repeating pattern.

NEURON OPTIMIZATION objective shows spatial preferences.

DATASET SAMPLES

Pieces of images from the training dataset that result in the largest activations from the given unit.

These images are cropped and downsize samples from the [ImageNet](#) research dataset. Unlike our other visualizations, they are not CC-BY-SA because they are derived from ImageNet.



Google, 2017: Attention is All You Need

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar
Google Research
nikip@google.com

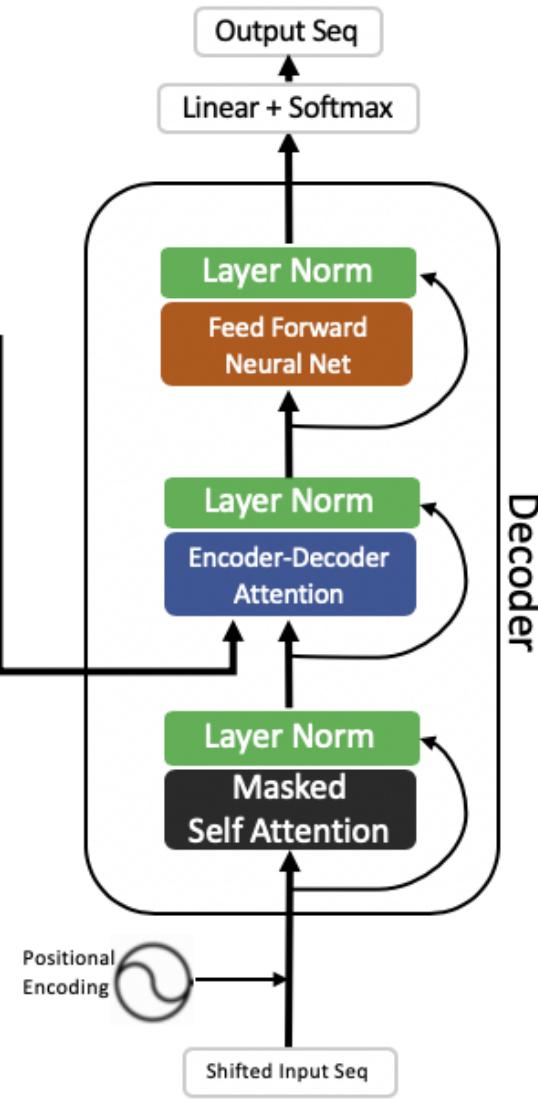
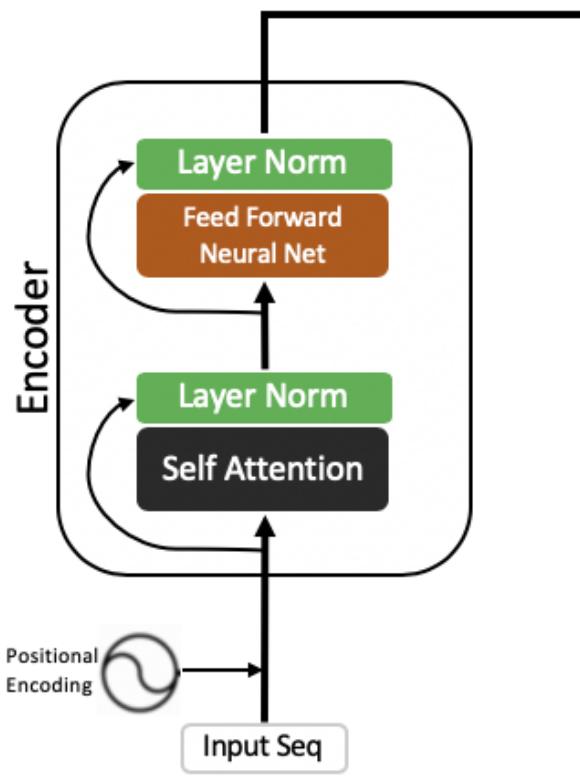
Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on recurrent convolutional neural networks that include an encoder and decoder. Recurrent models also connect the encoder and decoder via a shared recurrent mechanism. We propose a new simple network architecture based solely on attention mechanisms, dispensing with recurrence entirely. Experiments on two machine translation tasks show that our model is superior in quality while being more parallelizable and requiring less time to train. Our model achieves 28.4 BLEU on the English-to-German translation task, improving over the existing best ensembles, by over 2 BLEU. On the WMT 2014 English-to-French task, our model establishes a new single-model state-of-the-art BLEU score after training for 3.5 days on eight GPUs, a small fraction of the best models from the literature. We show that the Transformer can also succeed on other tasks by applying it successfully to English constituent ordering and limited training data.



OpenAI: Image GPT, Juni 2020

Generative Pretraining from Pixels

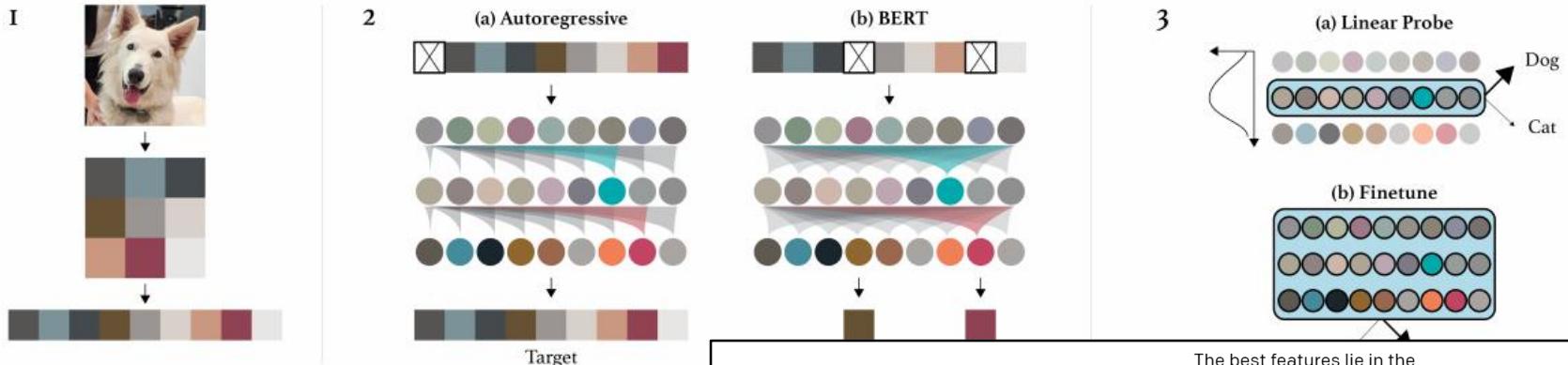


Figure 1. An overview of our approach. First, we pre-process raw images into a 16x1 vector. We then chose one of two pre-training objectives, auto-regressive or BERT, to learn representations. Finally, we fine-tuned the representations learned by these objectives with linear probes.

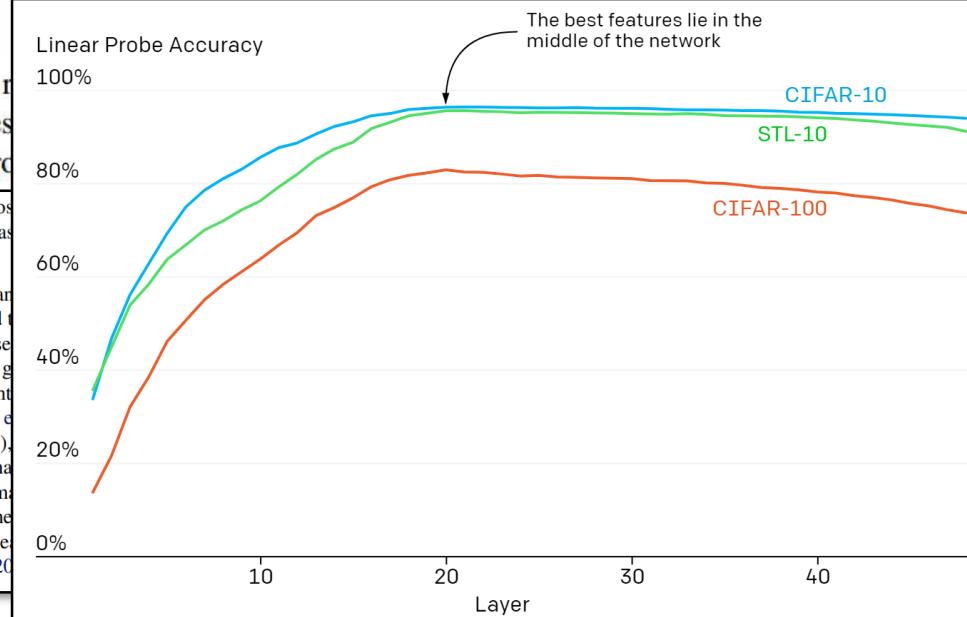
ture of ImageNet and web images is competitive with self-supervised benchmarks on ImageNet, achieving 72.0% top-1 accuracy on a linear probe of our features.

prediction of corrupted inputs, closely matching the Denoising Autoencoder, which was trained on raw images.

As a higher dimensional, noisier, and more complex than text, images are believed to require more sophisticated modeling. Here, self-supervised learning objectives encourage the modeling of more general features. Recent work (Ordóñez et al., 2015) have shown significant improvements in the quality of new training objectives (Ordóñez et al., 2015). In addition, new architectures (Gomez et al., 2017), and new training strategies (Kolesnikov et al., 2019) have led to achieve state of the art performance (Hénaff et al., 2019) and sometimes even surpass supervised representations in transfer learning (Misra & van der Maaten, 2019).

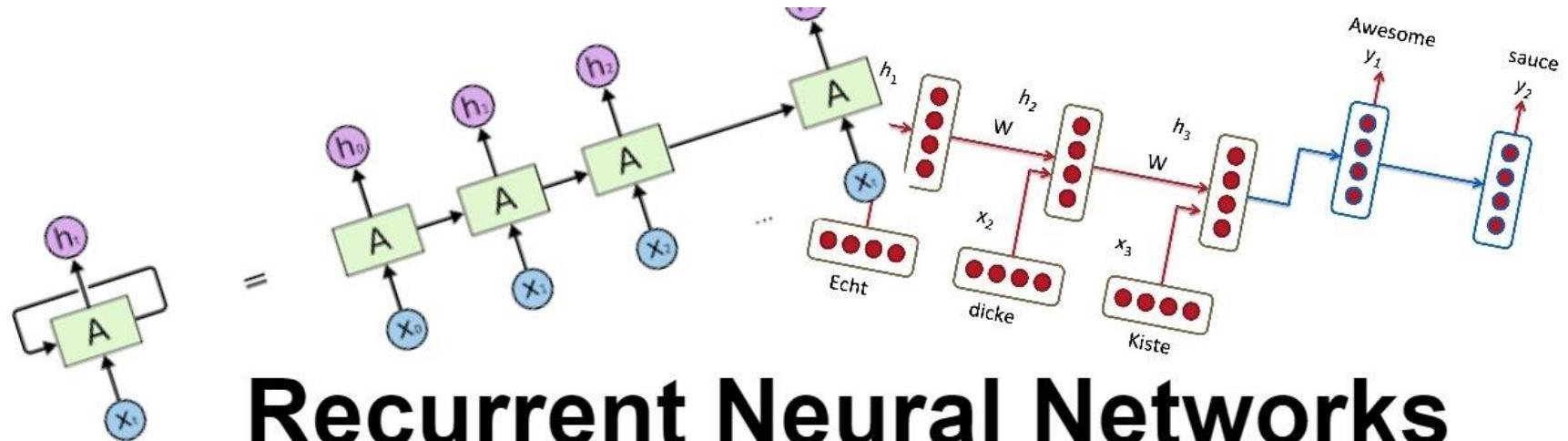
1. Introduction

Unsupervised pre-training played a central role in the resurgence of deep learning. Starting in the mid 2000's, approaches such as the Deep Belief Network (Hinton et al., 2006) and Denoising Autoencoder (Vincent et al., 2008) were commonly used in neural networks for computer vision (Lee et al., 2009) and speech recognition (Mohamed et al., 2009). It was believed that a model which learned the data distribution $P(X)$ would also learn beneficial fea-

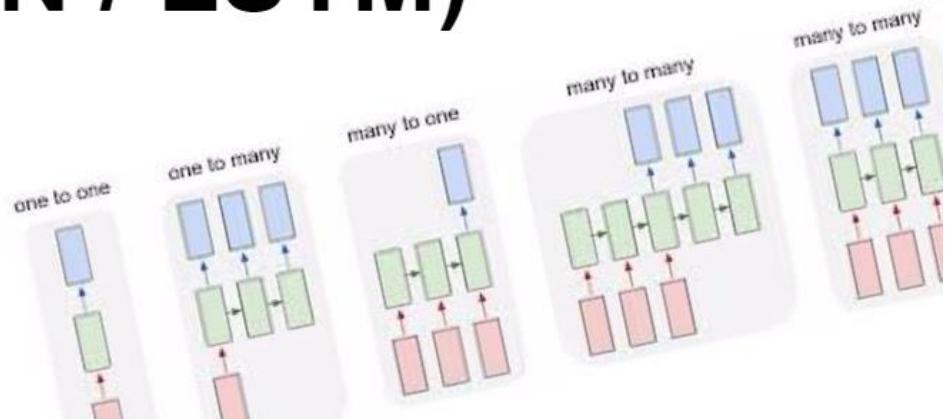


Feature quality depends heavily on the layer we choose to evaluate. In contrast with supervised models, the best features for these generative models lie in the middle of the network.

3. RNN & LSTM Networks



Recurrent Neural Networks (RNN / LSTM)



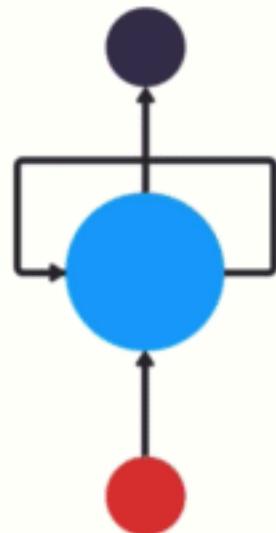
Recurrent Neural Networks (RNN)

Textanalyse mit einem RNN (Beispiel)

Output

Neuron

Input



Neuronen in RNNs haben jeweils
einen Speicher (Internal Memory /
Hidden State).

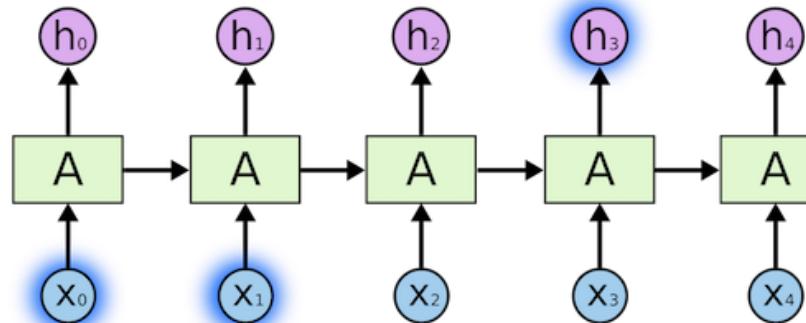
Es gibt verschiedene Arten von RNN-Netzen.

Illustrated Guide to Recurrent Neural Networks, 2018

towardsdatascience.com/illustrated-guide-to-recurrent-neural-networks-79e5eb8049c9

Recurrent Neural Networks (RNN)

Output:



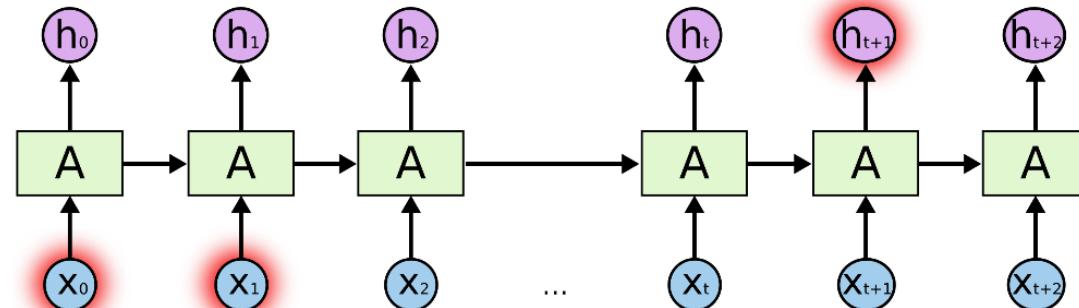
Beispiel
Textvorhersage
(wenig Kontext)

Input: The clouds in the

Output: French

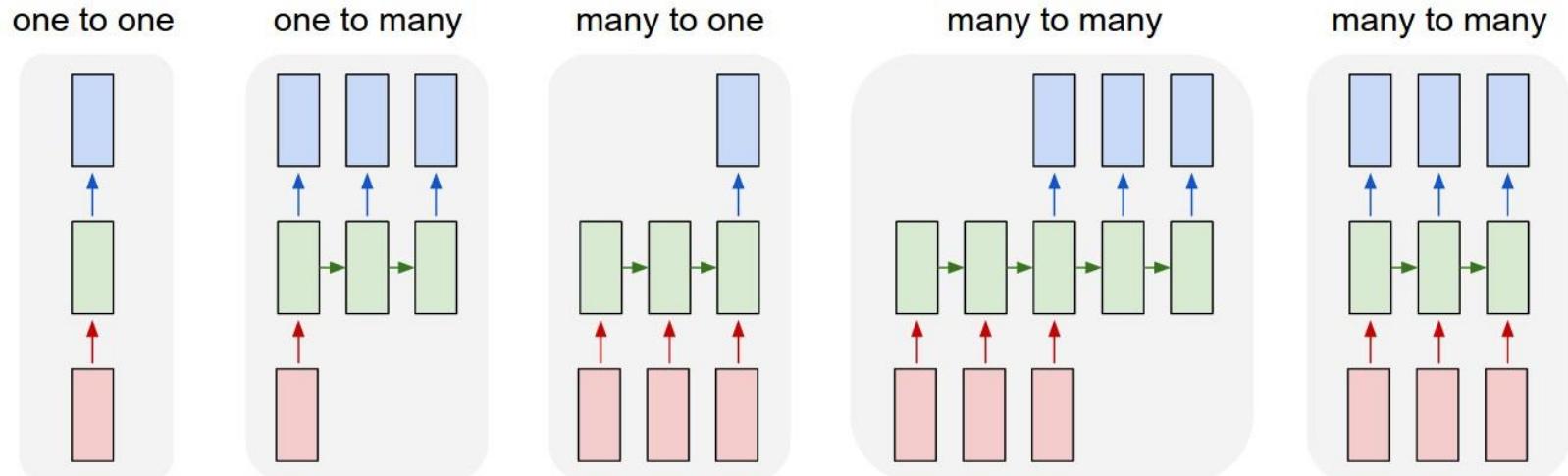
Beispiel
Textvorhersage
(mehr Kontext)

Input: I grew up in France ... I speak fluent



RNNs eignen sich gut für **kurzfristige Erinnerungen**,
weniger gut für **langfristige Erinnerungen** (Vanishing Gradient Problem).

Grundlegende RNN-Architekturen



One-To-One: Für einfache Anwendungen

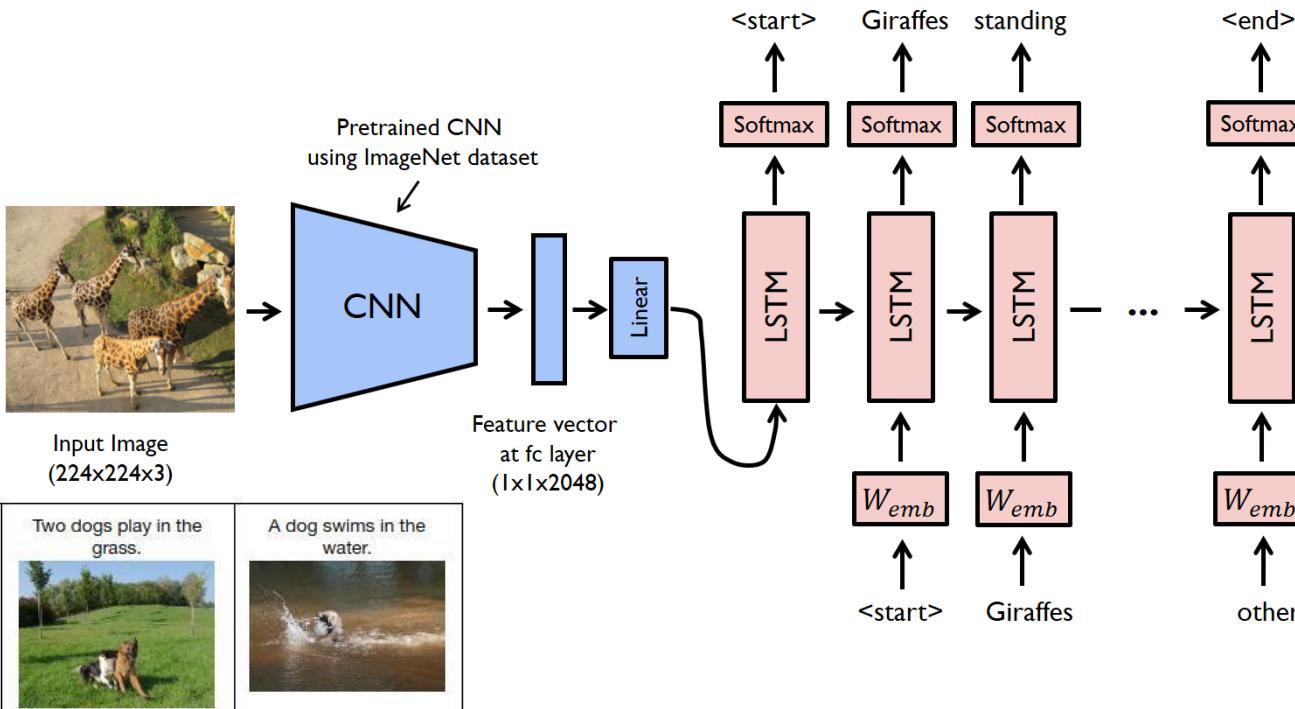
One-To-Many: Z.B. **Image Captioning** (Image -> Sequenz von Worten)

Many-To-One: Z.B. **Sentiment-Klassifikation** (Sequenz von Worten -> Sentiment)
Textvorhersage (Sequenz von Worten -> Wort)

Many-To-Many: Z.B. **Übersetzung** (Sequenz von Worten -> Sequenz von Worten)

**Many-To-Many:
(synched)** Z.B. **Video-Klassifikation** von Frames

Kombinierter Einsatz von RNN und LSTM



RNN und LSTM können auch mit **Feed-Forward-Netze** (z.B. **Convolutional Neural Networks**) kombiniert werden. Sie bringen damit „Memory“ hinein.

Anwendungen sind z.B.:
Textuelle Beschreibung von Bildern (**Image Captioning**) oder Automatische Generierung von Untertiteln (**Video Captioning**).

Generierung von Musik mit RNN und LSTM

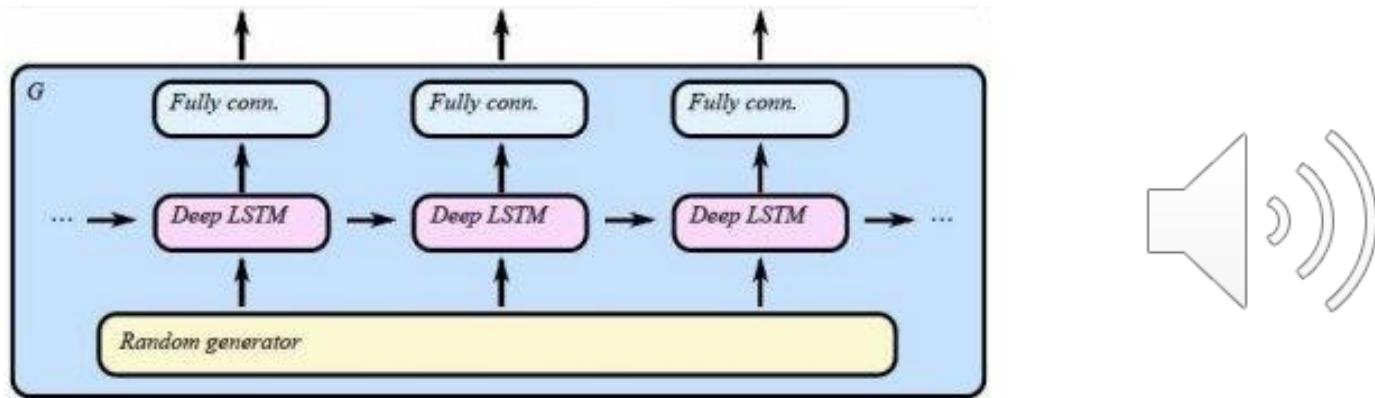


Fig. 7.27 C-RNN-GAN architecture.



Fig. 7.28 C-RNN-GAN generated examples.

Musik ist auch eine Sequenz (Sequence). Mit RNN und LSTMs kann man Musik generieren.

Music Composition using Recurrent Neural Networks

web.stanford.edu/class/archive/cs/cs224n/cs224n.1174/reports/2762076.pdf

Transformer Networks (machine learning model)

Transformer (machine learning model)

From Wikipedia, the free encyclopedia

The **Transformer** is a deep learning model introduced in 2017, used primarily in the field of natural language processing (NLP).^[1]

Like recurrent neural networks (RNNs), Transformers are designed to handle sequential data, such as natural language, for tasks such as [translation](#) and [text summarization](#). However, unlike RNNs, Transformers do not require that the sequential data be processed in order. For example, if the input data is a natural language sentence, the Transformer does not need to process the beginning of it before the end. Due to this feature, the Transformer allows for much more [parallelization](#) than RNNs and therefore reduced training times.^[1]

Transformers have rapidly become the model of choice for NLP problems,^[2] replacing older recurrent neural network models such as the [long short-term memory](#) (LSTM). Since the Transformer model facilitates more parallelization during training, it has enabled training on larger datasets than was possible before it was introduced. This has led to the development of [pretrained systems](#) such as [BERT](#) (Bidirectional Encoder Representations from Transformers) and [GPT](#) (Generative Pre-trained Transformer), which have been trained with huge general language datasets, such as Wikipedia Corpus, and can be fine-tuned to specific language tasks.^{[3][4]}

Google, 2017: Attention is All You Need

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Llion Jones*
Google Research
llion@google.com

Noa Golany*
Google Brain
noamg@ai.goo

Aidan老人
University of
Edinburgh
aidan@inf.ed.ac.uk

Clémentine

Niki Dyer*

Torko Elmiussin*

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Google, 2017: Attention is All You Need

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkorei
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukasz.kaiser@google.com

Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

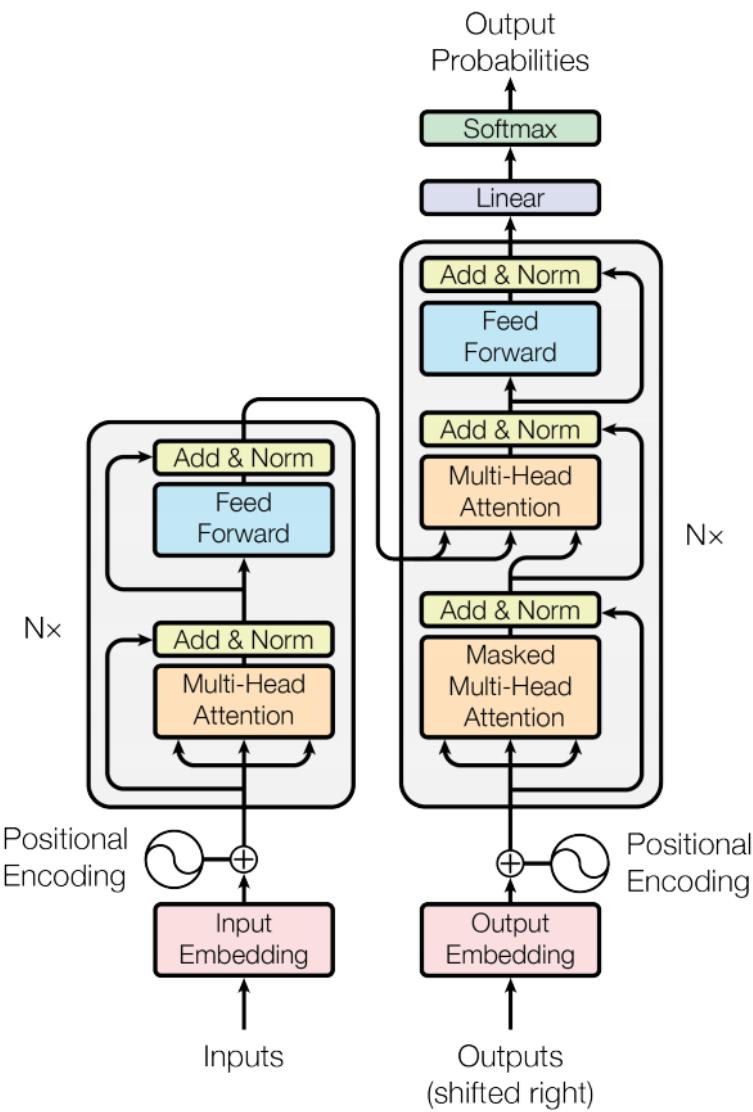


Figure 1: The Transformer - model architecture.

GPT-3: Trainingsdaten

Since Neural Networks are **compressed/compiled version** of the training data, the size of the dataset has to scale accordingly with the size of the model. GPT-3 175B is trained with 499 Billion tokens. Here is the breakdown of the data:

Dataset	# Tokens (Billions)
Total	499
Common Crawl (filtered by quality)	410
WebText2	19
Books1	12
Books2	55
Wikipedia	3