Introduction – Deep Learning Bootcamp

Technische Hochschule Ingolstadt

KI-basierte Optimierung in der Automobilproduktion

Technische Hochschule Ingolstadt

About this Course

Organisation

- Lecturer: Alexander Schiendorfer
- 3 Days, 9:00 – 17:00
- 1,5 Hour Break for Lunch and Coffee

- **Goals**: You should get a basic understanding of the presented techniques, being able to program is not necessary for it!

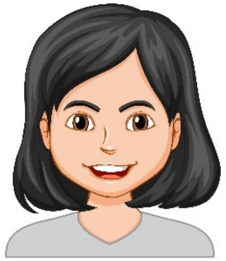
- **Topics**:
  - Introduction to *Artificial Intelligence* in general, *Machine Learning* and ***Deep Learning***
  - How can we solve an image classification problem with Deep Learning?
  - Basic Feed Forward Neural Networks
  - Convolutional Neural Networks
  - Transfer Learning
  - **Discussion**: What‘s the impact of AI on society?

Introduction Round

- What's your name?

Please switch on your cameras for this introduction round

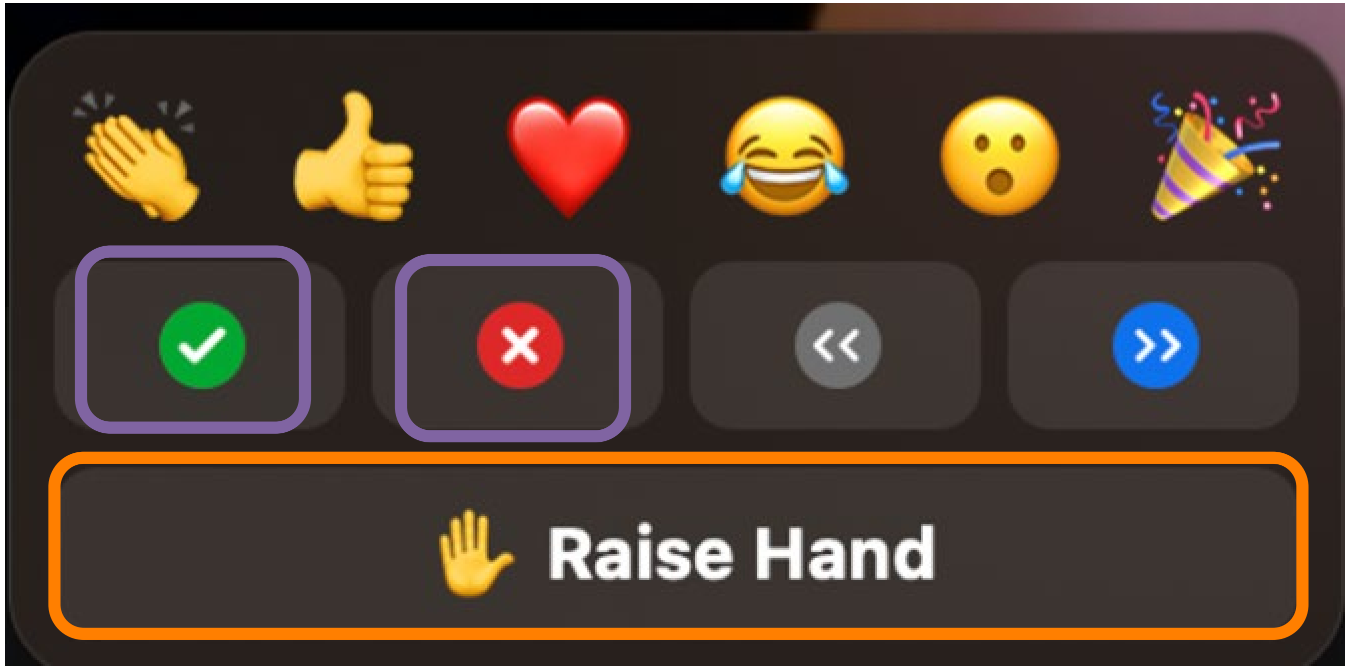
- Tell us something you heard about AI
  - that is exciting / interesting / useful
  - that you think is problematic / dangerous / annoying



https://www.vecteezy.com/vector-art/416880-men-and-women-with-different-emotions

https://www.vecteezy.com/vector-art/242746-face-emoticon-boy-with-glasses-vector-collection

Game plan for the course



When you have questions or remarks

Bei Abstimmungen

Your ideal setup

On one screen:

- Open the browser for exercises

On the other screen:

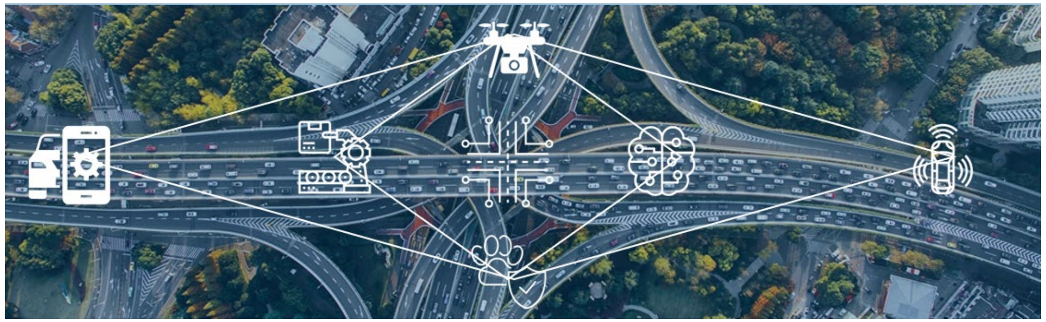
- Watch the lectures and instructions on my shared screen
- Open the PDF of the course

You could use your cell phone to listen on Zoom, too

Key facts „AImotion Bavaria“

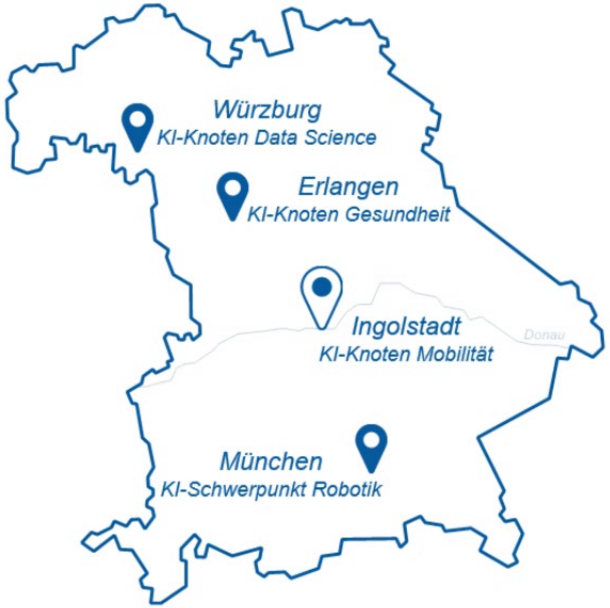
Bavarian AI-node for Mobility @ THI

AImotion Bavaria



- **20 new research professorships**
  - 10 from Hightech Agenda Bavaria
  - 10 from industry funding (among others: Audi, Stadt Ingolstadt, Klinikum, etc.)
  - *Key topics*: Autonomous mobility (driving, flying), **AI-based manufacturing**
    - BSc / MSc Theses available!
  - ***Goal***: Growth to 120 scientists / PhD researchers (maybe yourself?)

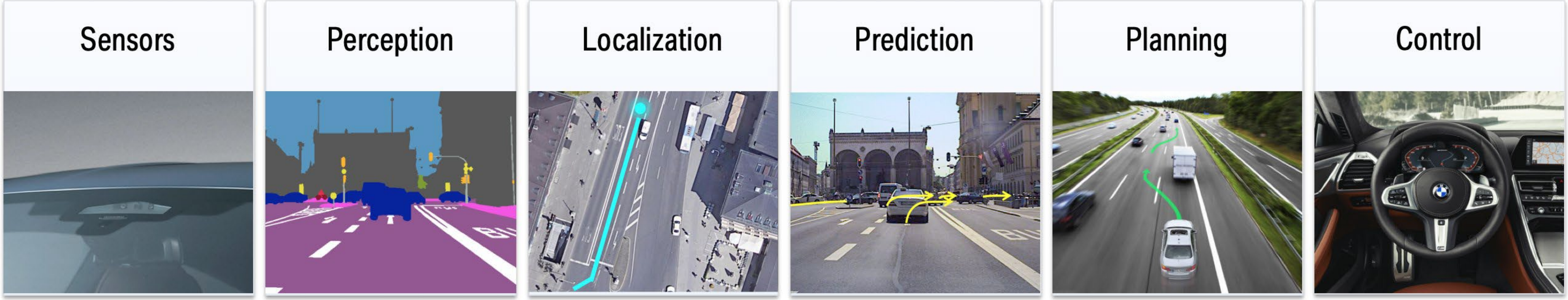
- **Soon located in the digital building (Kavalier Dalwigk)**
  - 4.000 m² area, classrooms, AI research labs
  - Office space for startups



What is Artificial Intelligence (AI)?

Artificially Intelligent, Self-driving Cars?

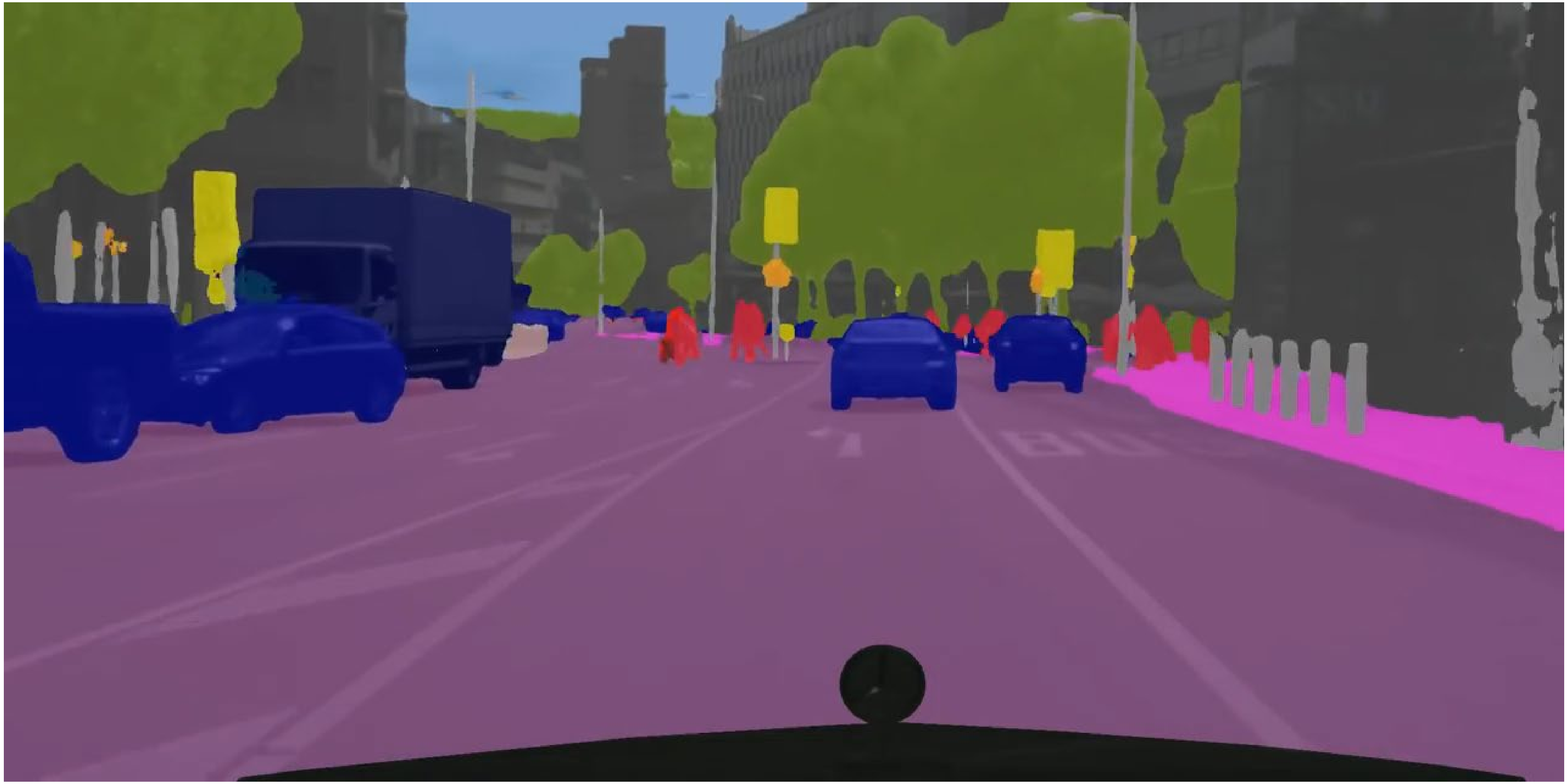
How can an autonomous car navigate through a city? What aspects does this involve?



Most steps are heavily influenced by deep learning, nowadays!

Image: @haltakov
https://twitter.com/haltakov/status/1382014488174530563

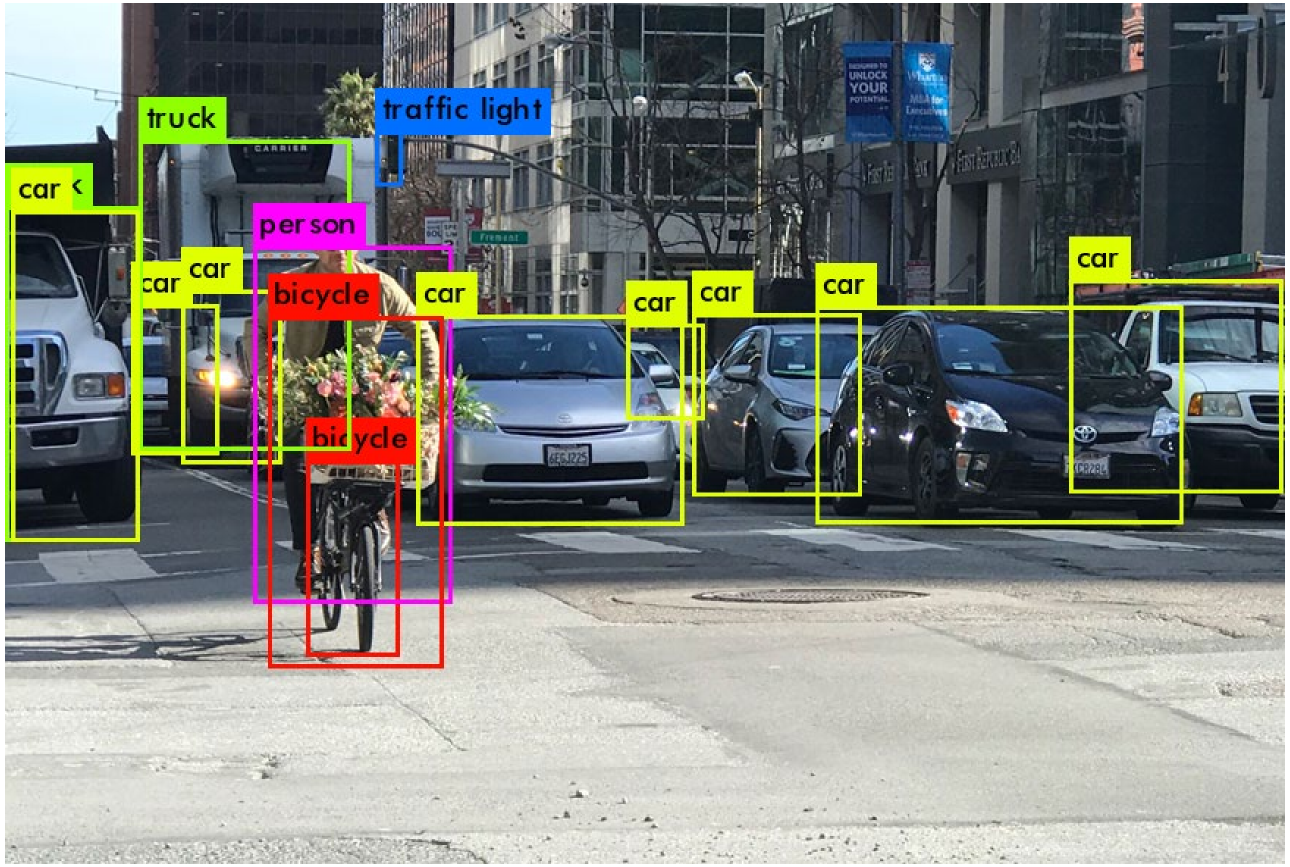
Semantic Segmentation

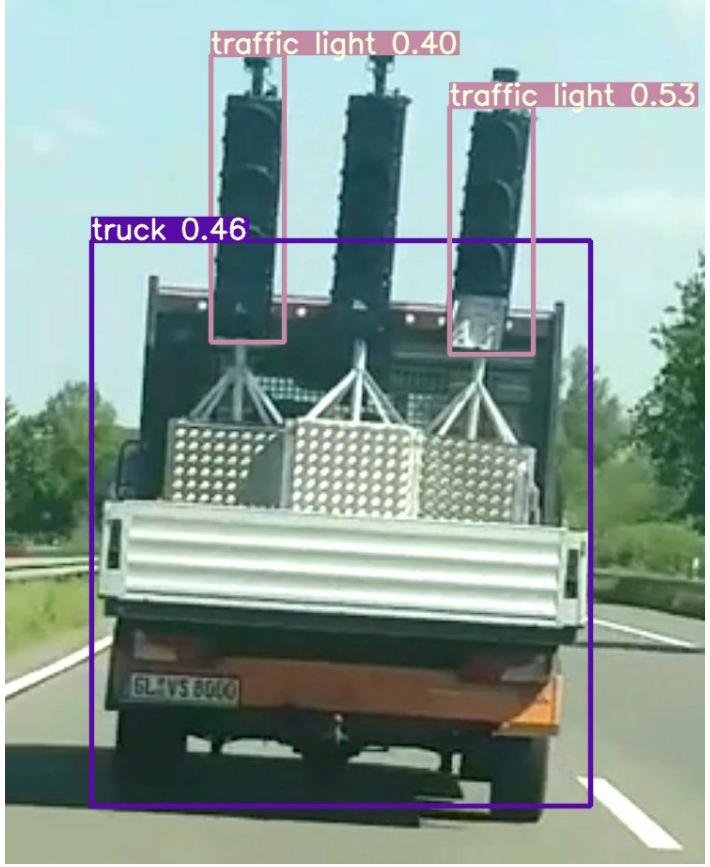


[Pohlen, Hermans, Mathias, Leibe, 2016]

Object Detection

→The car must recognize other cars, pedestrians, street lines, etc.





https://neilnie.com/2018/11/18/implementing-yolo-v3-object-detection-on-the-autonomous-vehicle

Traffic sign recognition: A step in the vision part

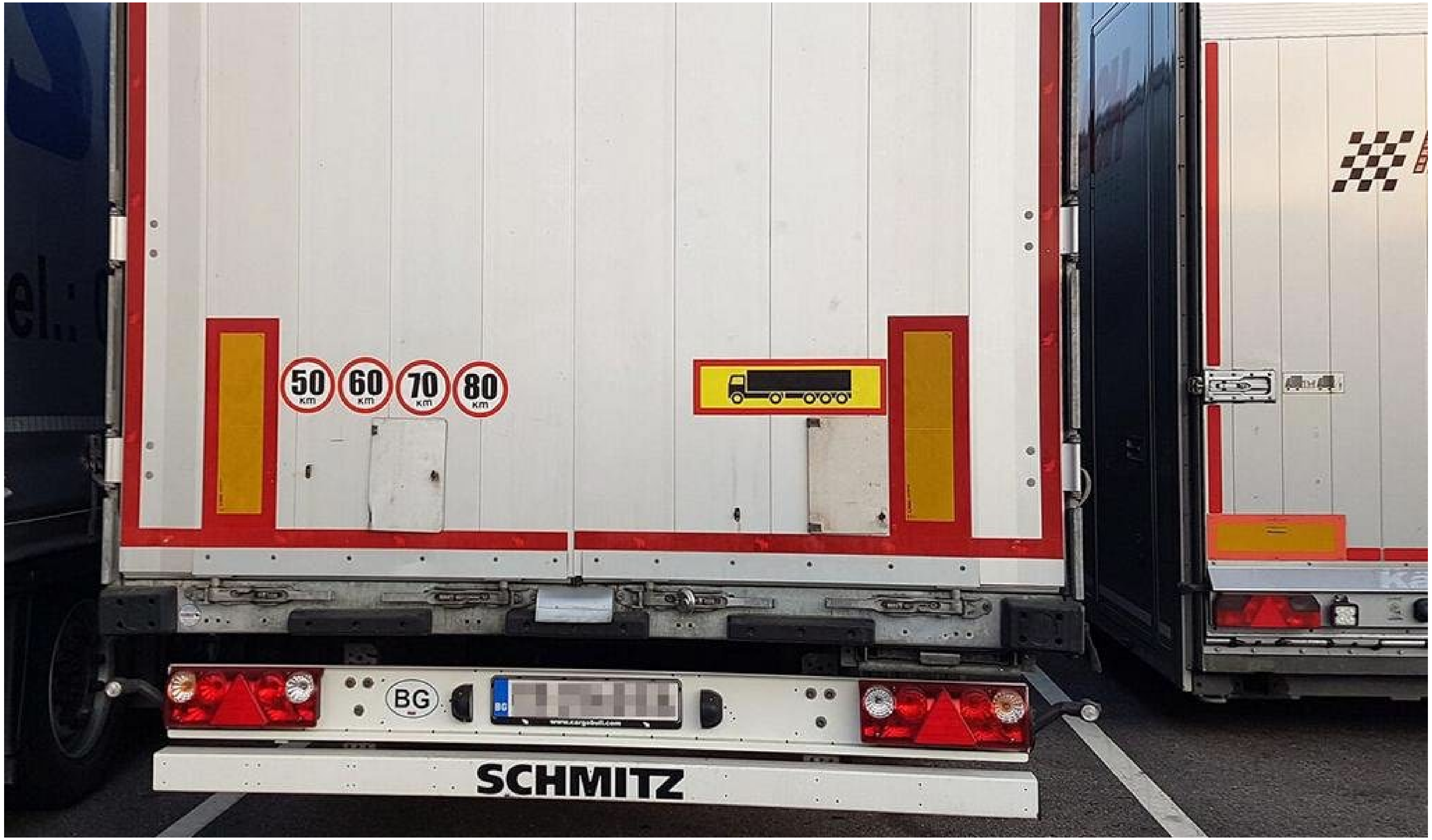
→It also must recognize traffic signs!

→We will build this part in the bootcamp!

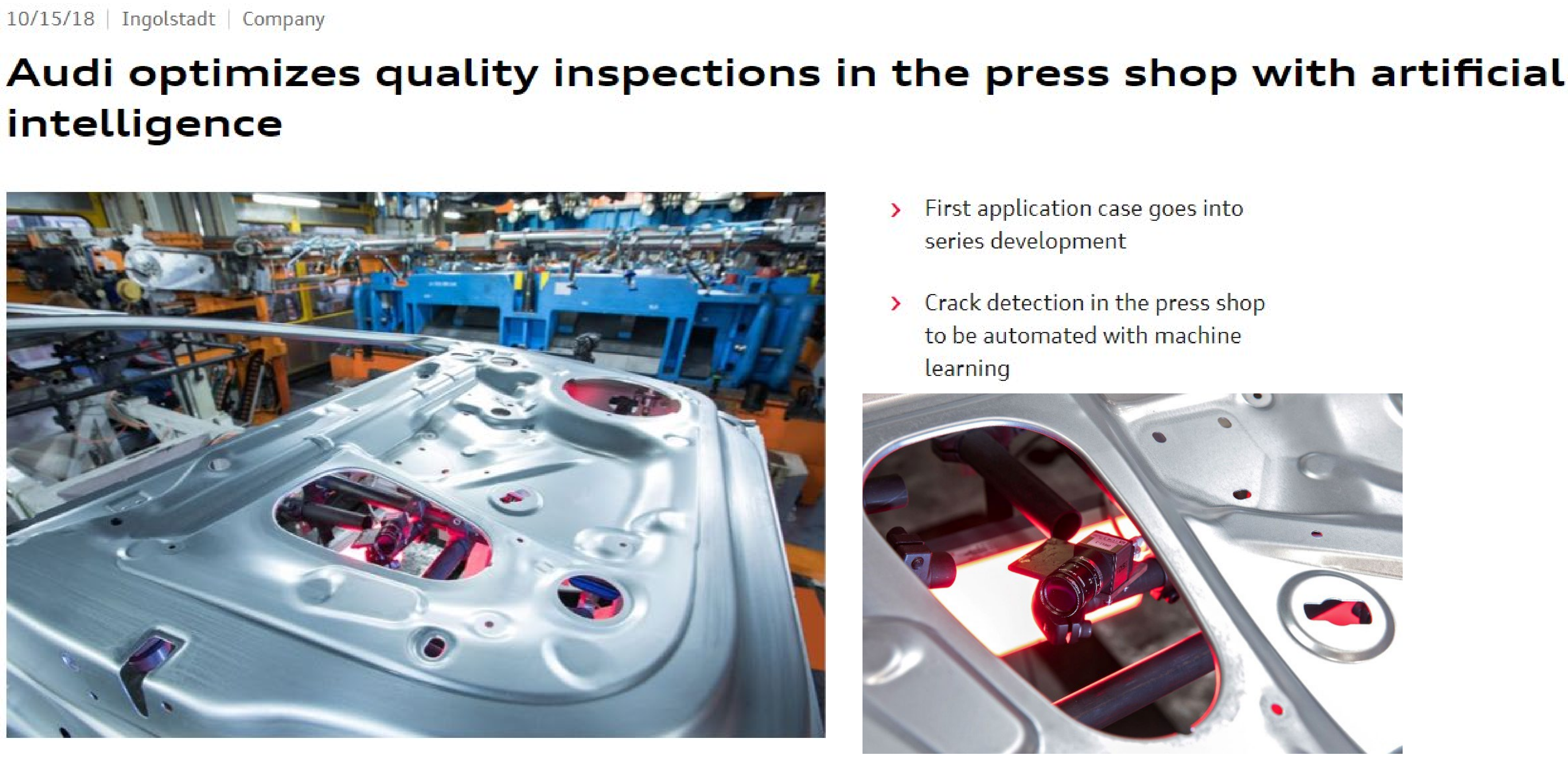
How can the car recognize the traffic signs?

⍰ Deep Learning and Computer Vision can solve this problem

Trickier than it might seem …



Other application: Visual quality inspections at Audi



10/15/18 | Ingolstadt | Company

Audi optimizes quality inspections in the press shop with artificial intelligence

- First application case goes into series development
- Crack detection in the press shop to be automated with machine learning

https://www.audi-mediacenter.com/en/press-releases/audi-optimizes-quality-inspections-in-the-press-shop-with-artificial-intelligence-10847

Questions

- Do you see AI as a positive or a negative topic?

- Do you think AI will have positive, negative or no effects on your life?

- Would you use AI in your life?

- What do you know about AI?

https://docs.google.com/presentation/d/1CMsPfX2wpTX4Tb_jaSEmwDFiQQl80jaL/edit?usp=sharing&ouid=112995901544340943491&rtpof=true&sd=true

How can we define AI?

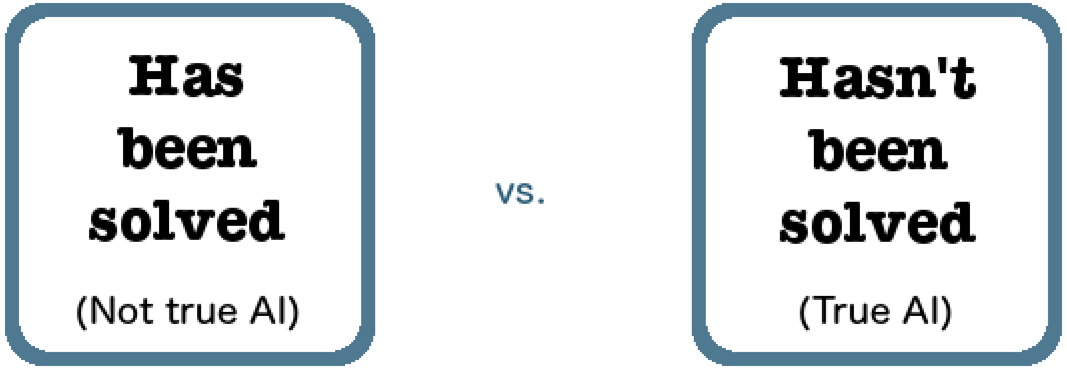
AI = the theory and development of computer systems able to perform tasks normally requiring **human intelligence**, such as visual perception, speech recognition, decision-making, and translation between languages. [Oxford Dictionary]

- Multiplying two large numbers?
  - $786\ 234\ 578\ 193\ 148\ 237\ 230\ 534 \cdot 234\ 578\ 786\ 148\ 237\ 193\ 230\ 534 = ?$
- Memorizing and displaying a large document?
- Calculating the fastest way to travel from A to B?
- Recognizing images?

The „AI Effect“

In the past, we would have said only a superintelligent AI could drive a car or beat a human at Jeopardy! or chess. But once AI did each of those things, we considered that achievement obviously mechanical and hardly worth the label of true intelligence. **Every success in AI redefines it.**

Kevin Kelly, 2014



Has been solved

(Not true AI)

vs.

Hasn't been solved

(True AI)

Problematic Definition?

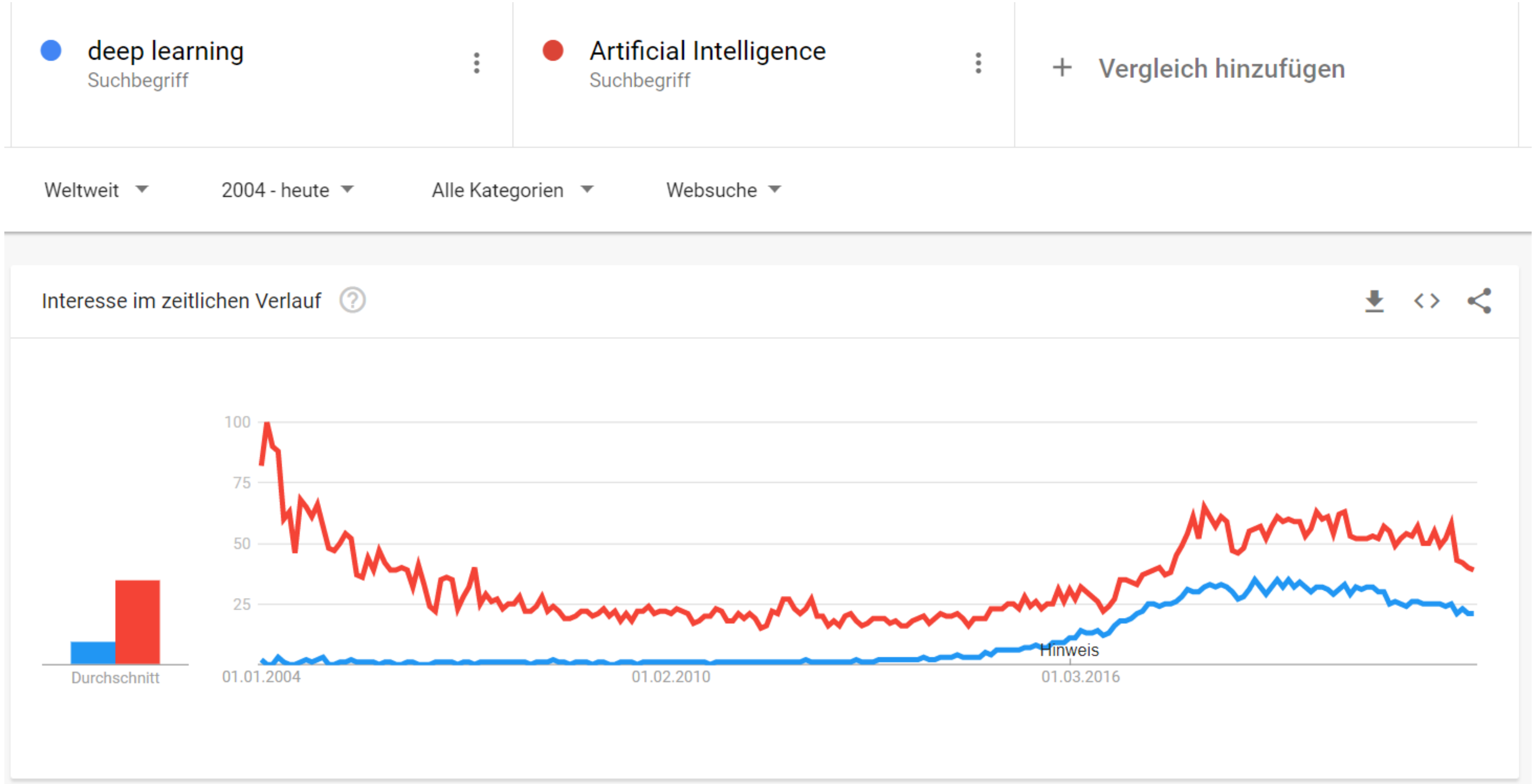
> Artificial Intelligence is the study of how to make computers do things at which, **at the moment,** people are better.

Elaine Rich

Nowadays as well: Solve problems that humans cannot do due to cognitive limitations:

- Play millions of games of Go to beat world champions
- Learn better strategies for some problems than any preprogrammed solution (by smart trial & error)
- Semi-automatically find correlations & patterns in vast amounts of data

Interest in Deep Learning?

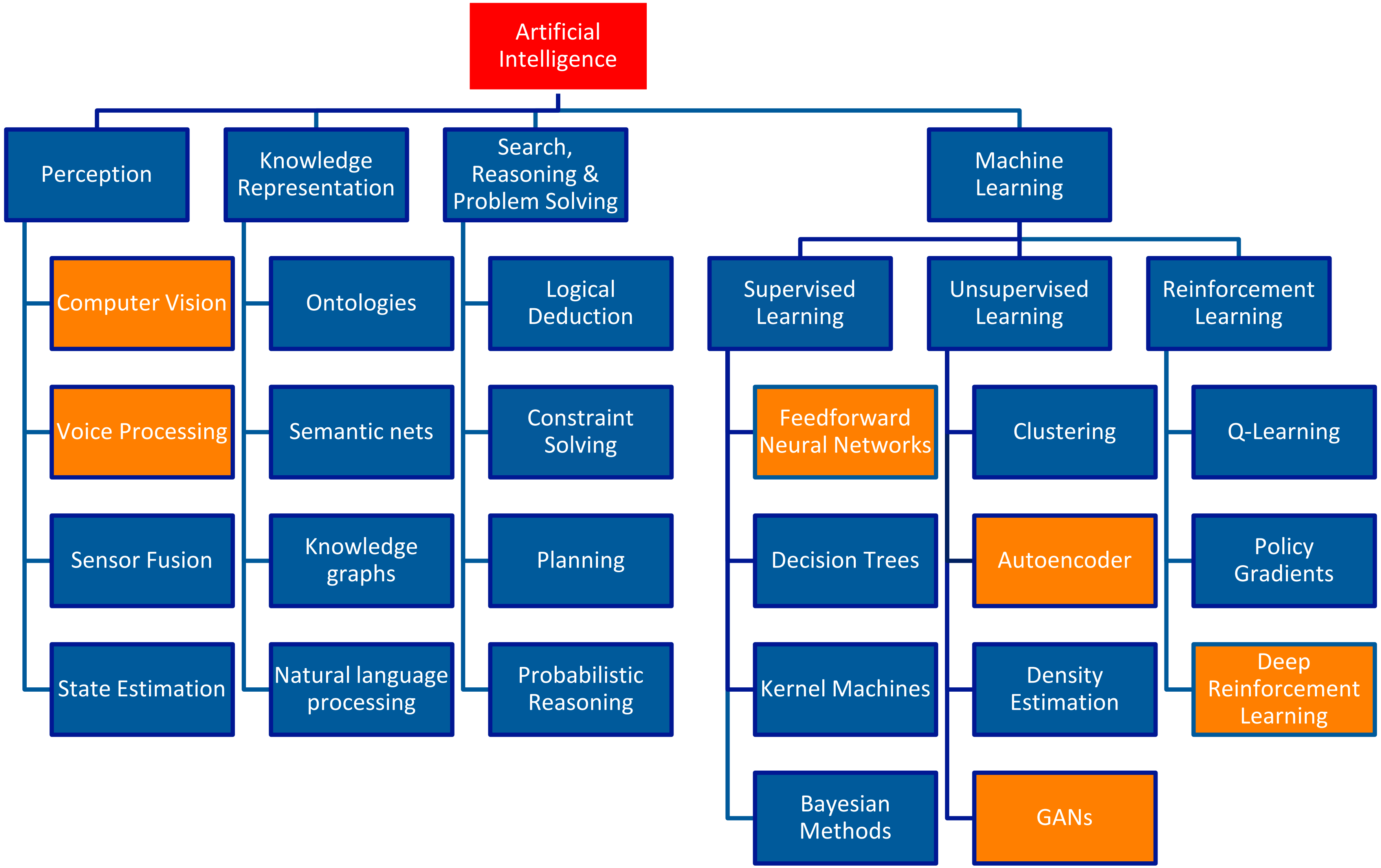


● deep learning
Suchbegriff

● Artificial Intelligence
Suchbegriff

Weltweit ▾ 2004 - heute ▾ Alle Kategorien ▾ Websuche ▾

https://trends.google.com/trends/explore?date=all&q=deep%20learning,Artificial%20Intelligence



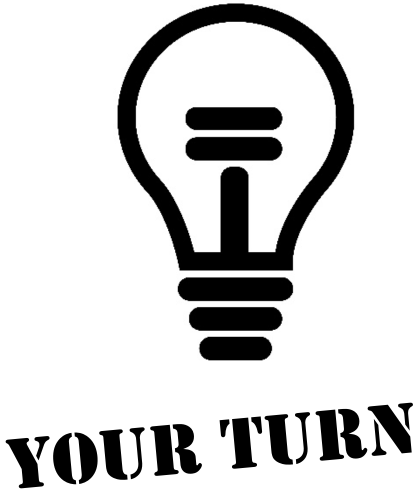
- Artificial Intelligence
  - Perception
    - Computer Vision
    - Voice Processing
    - Sensor Fusion
    - State Estimation
  - Knowledge Representation
    - Ontologies
    - Semantic nets
    - Knowledge graphs
    - Natural language processing
  - Search, Reasoning & Problem Solving
    - Logical Deduction
    - Constraint Solving
    - Planning
    - Probabilistic Reasoning
  - Machine Learning
    - Supervised Learning
      - Feedforward Neural Networks
      - Decision Trees
      - Kernel Machines
      - Bayesian Methods
    - Unsupervised Learning
      - Clustering
      - Autoencoder
      - Density Estimation
      - GANs
    - Reinforcement Learning
      - Q-Learning
      - Policy Gradients
      - Deep Reinforcement Learning

Deep Learning

Task: State of the Art

Play with the following demos in your Webbrowser and tell us about your experiences!

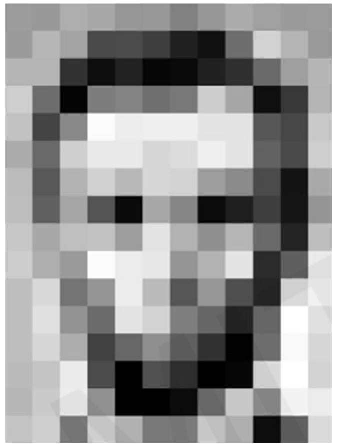
- http://gaugan.org/gaugan2/
- https://www.nvidia.com/research/inpainting/index.html
- https://quickdraw.withgoogle.com/



Machine Learning & Deep Learning

Why Machine Learning?

- Could we define **rules** why we see Abraham Lincoln in the highly pixelated image?
- Could we define **rules** why two traffic signs mean something different?
- How does a dog differ from a cat?



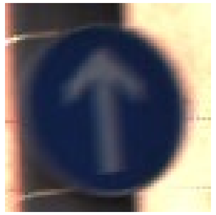
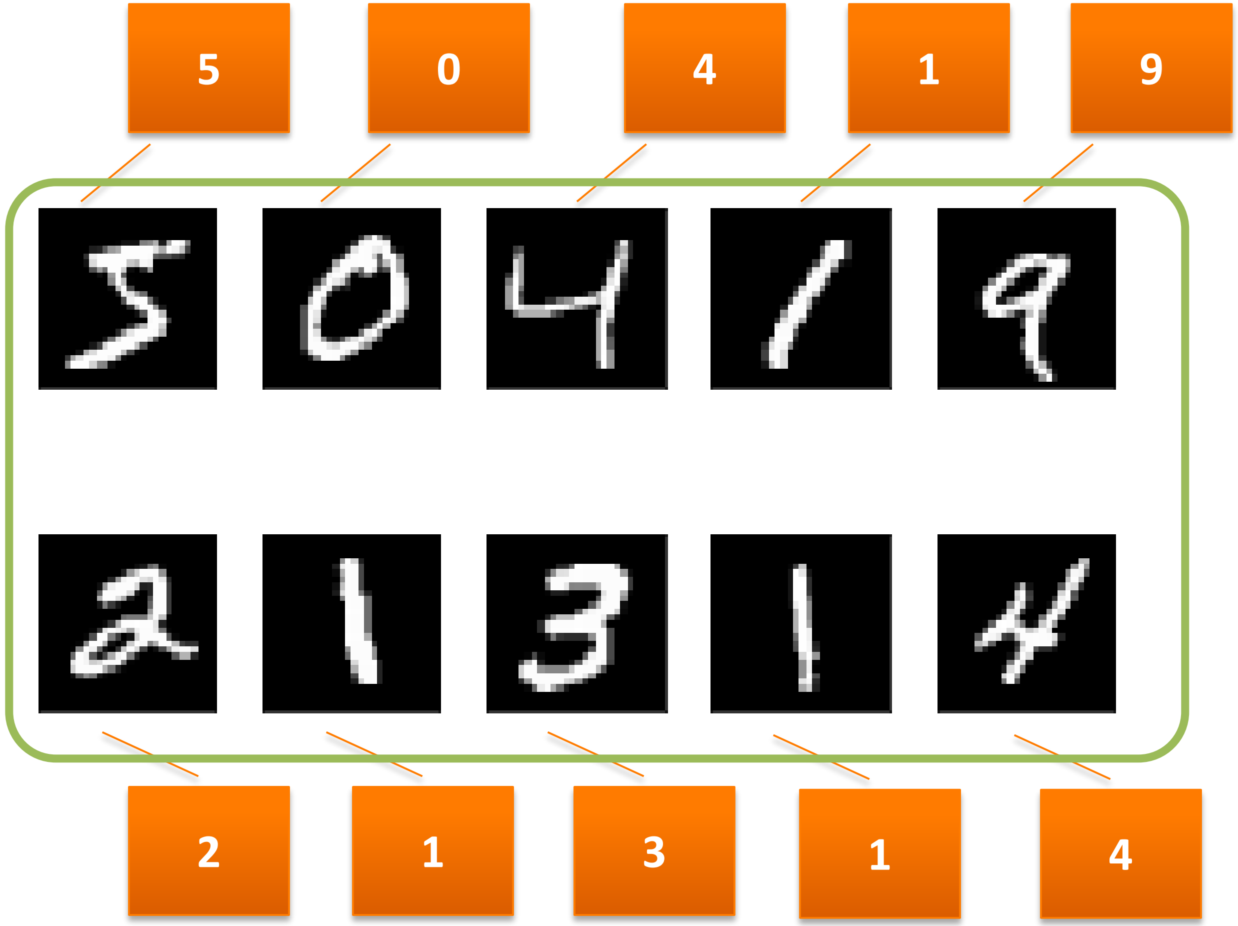
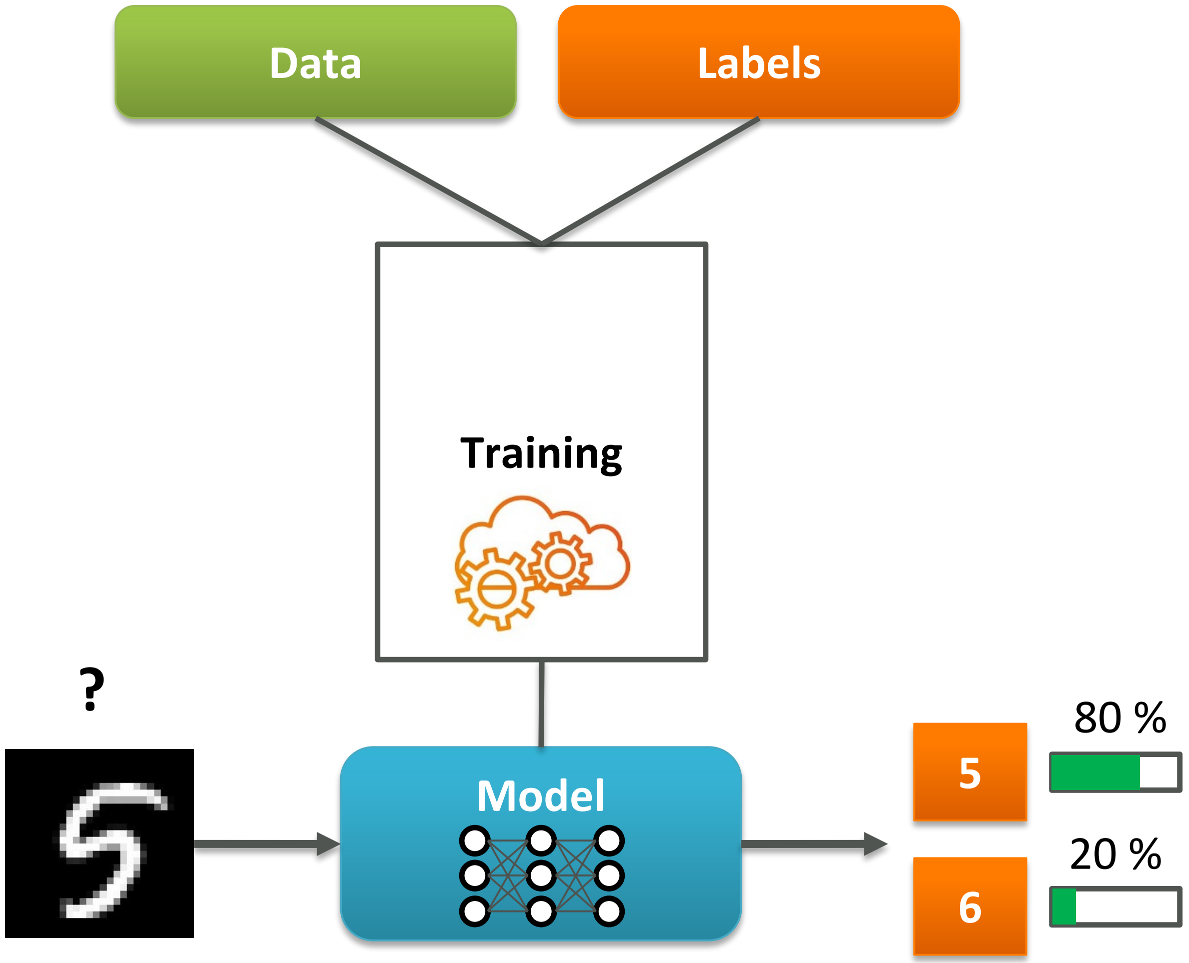


Foto: photology1971 – stock-adobe.com

Machines that learn rules from data (Supervised Learning)





Machine Learning and Deep Learning

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E." – Mitchell, 1997

Supervised Learning

Data: Pairs with input x and target t

Infer approximate functional mapping
$f: x \mapsto t$

Classification
Discrete targets
(finitely many)

Regression
Continuous targets
(real-valued)

Unsupervised Learning

Data: Only input x

Pattern recognition, Clustering, Dimensionality reduction, Embeddings/representations

Reinforcement Learning

Data: Receive reward r in state s at time t for taking action a

Learn a policy for a Markov decision process; Given state s, which action should I take?

Machine Learning and Deep Learning

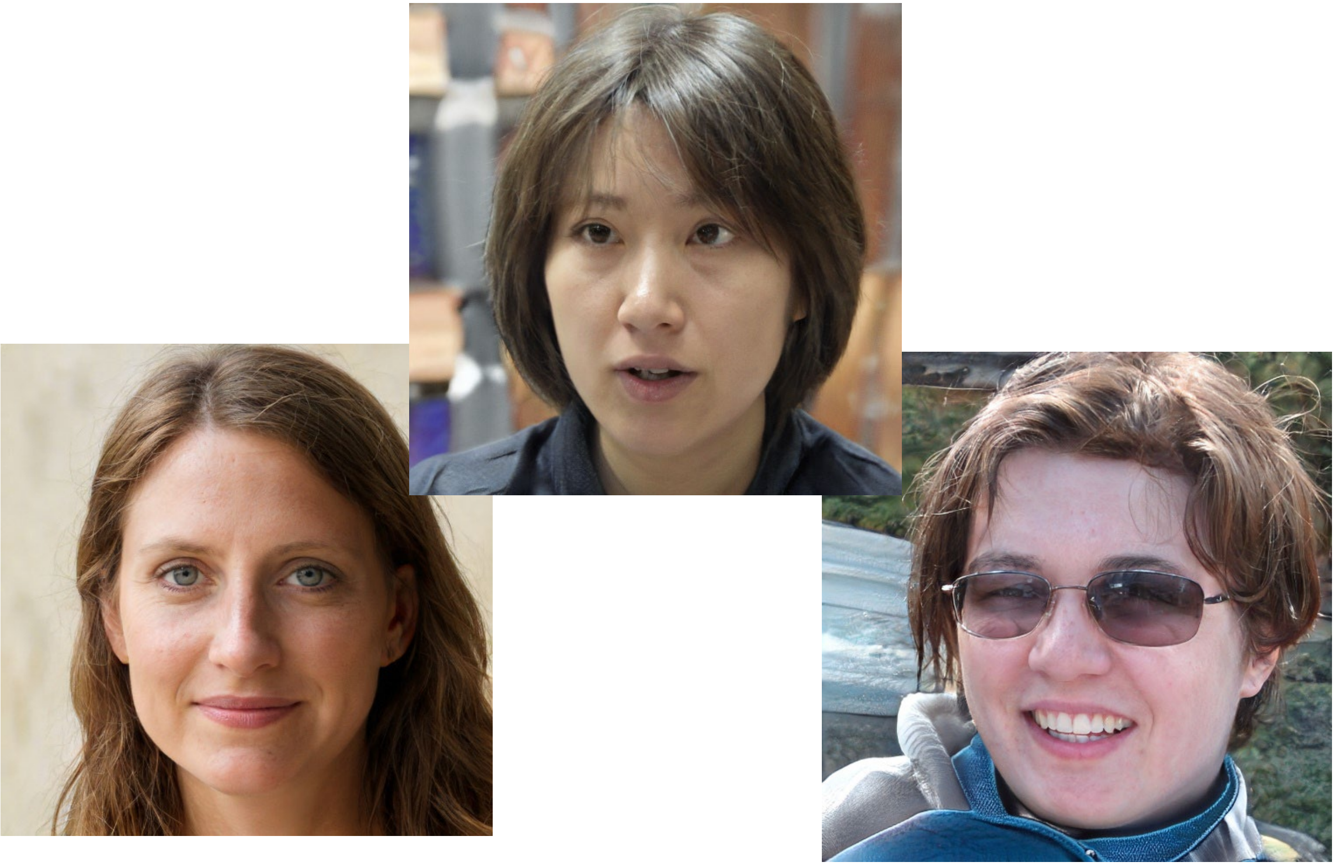
- The field of ML has been an active research area since the 1940s
- Historically: 3 waves of machine learning excitement

Cybernetics	Connectionism	Currently: „Third wave of neural networks“
Ca. 1940s-1960s Foundation of **control theory** Only simple, linear models Limited by simple non-linear problems	Ca. 1980s-1990s More complex, layered models **(= artificial neural networks, ANN)** Limited by computational resources	Since ca. 2006 **Very deep** and complex models State of the art in several areas(computer vision, language processing, …)

State of the Art

- Modern AI (very often in form of neural networks) can produce impressive results

- Why “AI = Deep Learning”?



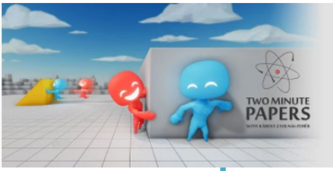
Images from thispersondoesnotexist.com

Task: State of the Art

Search for current state of the art Deep Learning models and what they can achieve!

Breakout room #1: Autonomous driving in action

Breakout room #2: Deep Learning Solutions for Automotive Manufacturing



Breakout room #3: Learning to play hide & seek

Breakout room #4: Free choice of topic

Breakout room #5: Free choice of topic

YOUR TURN

Youtube Links für Breakout-Sessions

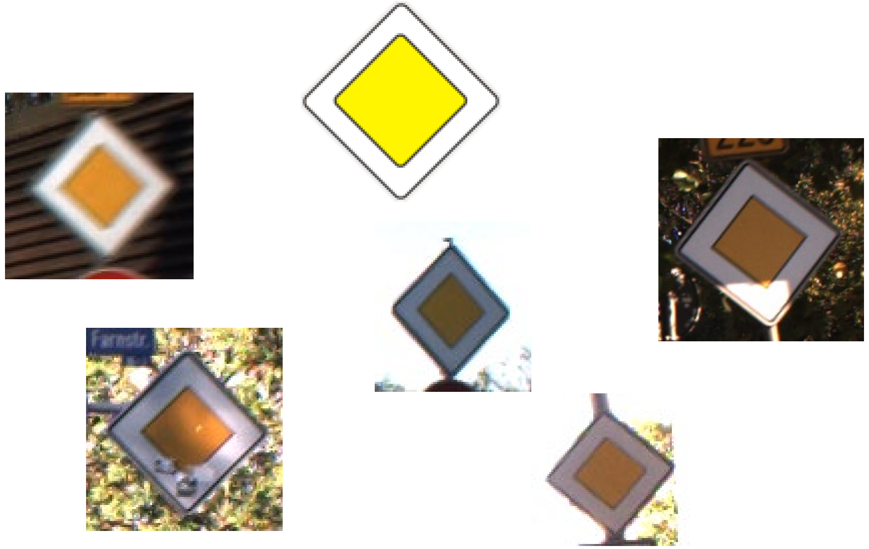
- Breakout #1: https://youtu.be/PhSooO33Eus?t=232
- Breakout #2: https://www.youtube.com/watch?v=krd49sG05no
- Breakout #3: https://www.youtube.com/watch?v=Lu56xVlZ40M

Image Classification

This course: German Traffic Sign Recognition Benchmark

- We want to develop solutions for the German Traffic Sign Recognition Benchmark (GTSRB) Dataset

- Dataset was released by Ruhr-Uni Bochum in 2013 as a competition
- Consists of more than 50 000 images of 43 different German traffic signs

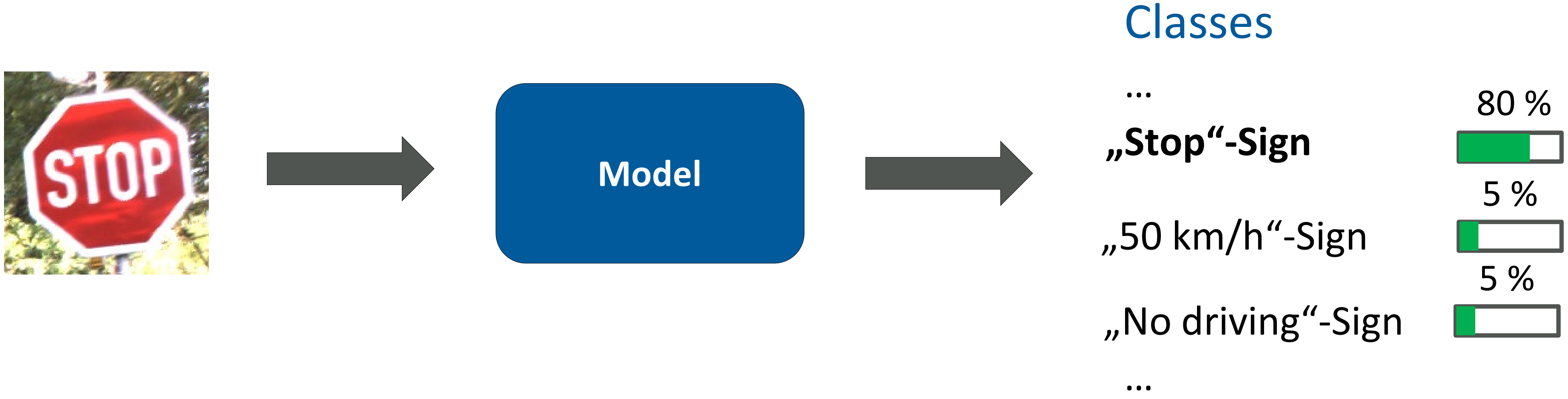


What‘s the task?

Our Goal: Solve the classification problem using neural networks

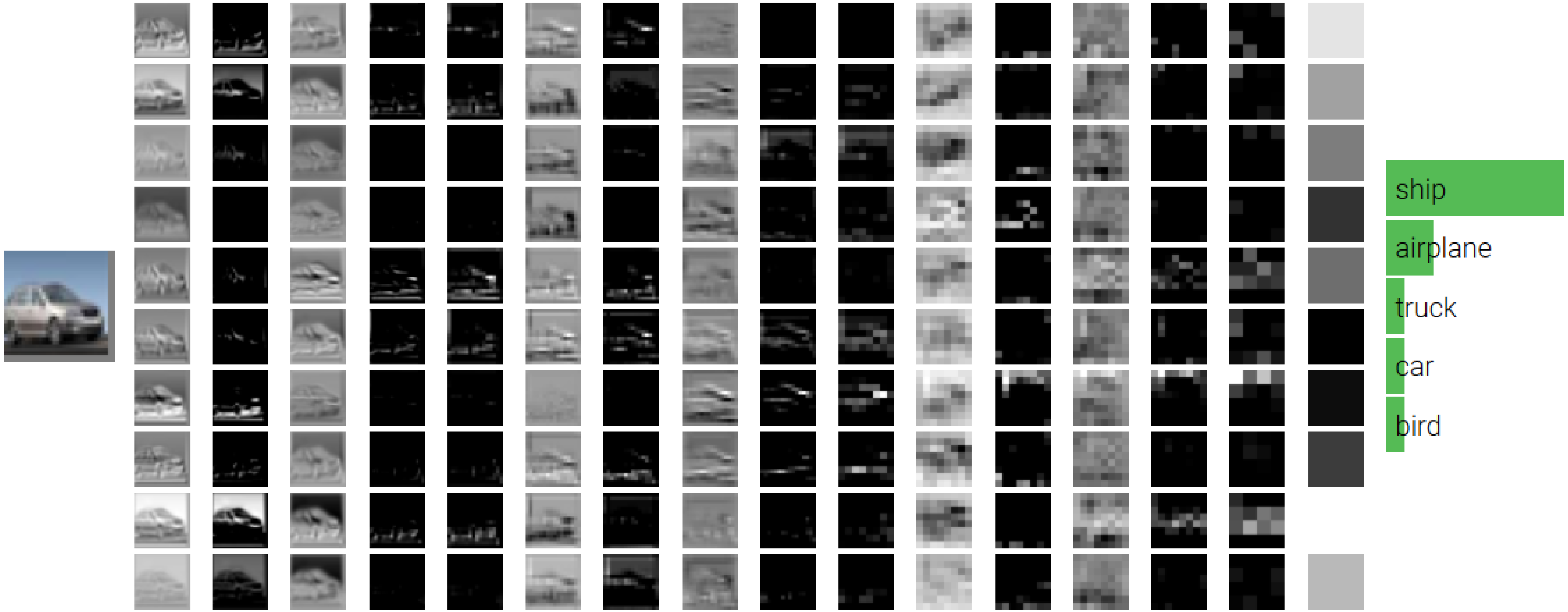
Classification Problem:

- *Informal*: We want to assign the correct label to an image
- *Formal*: Our model acts as a function f that takes an instance x as input and outputs a label y: $f: x \mapsto y$



What‘s the task?

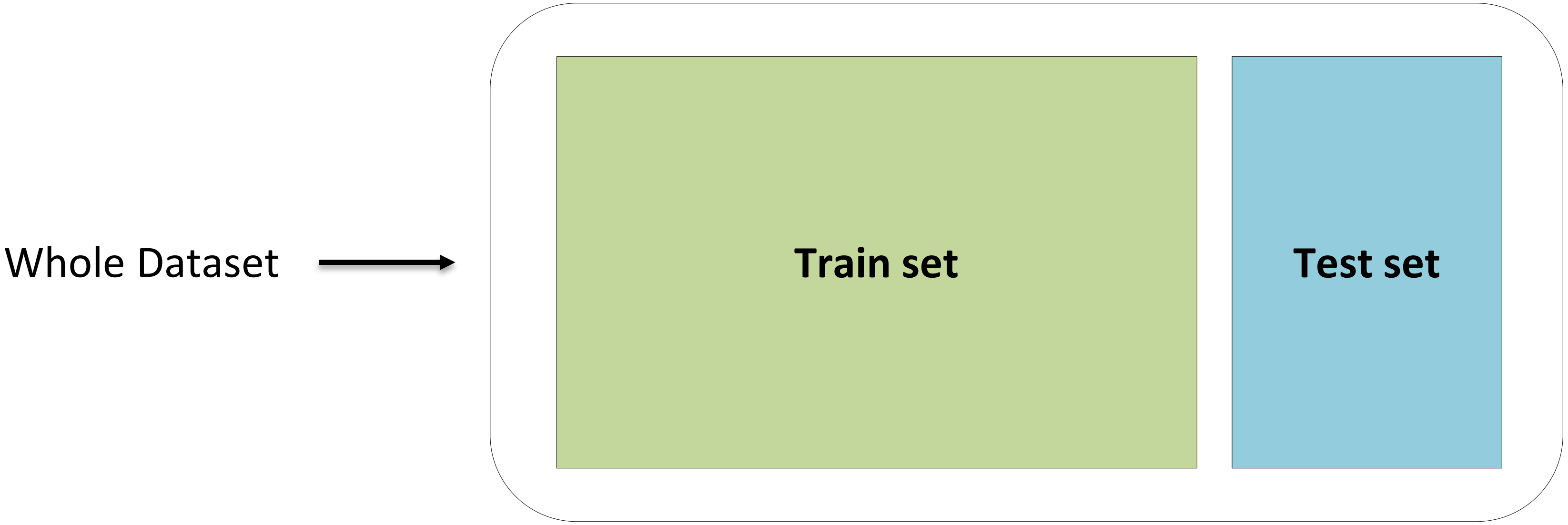
Have a look at: http://cs231n.stanford.edu/



Evaluation of Classifiers

We want to use the training data to learn a model that classifies *unknown* instances correctly („Generalization”)

→We don‘t use the whole dataset for training, but train and test splits!



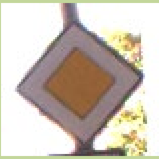
Evaluation of a Classifier

Train set Accuracy: 5/6 = 83.33 %

↦ Priority ✓

↦ Priority ✗

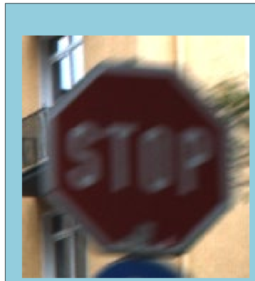
↦ Stop ✓


↦ Stop ✓

↦ Priority ✓

↦ Stop ✓

Test set Accuracy: 2/3 = 66.66 %


↦ Priority ✗

↦ Stop ✓

↦ Priority ✓

Accuracy is not the only metric!

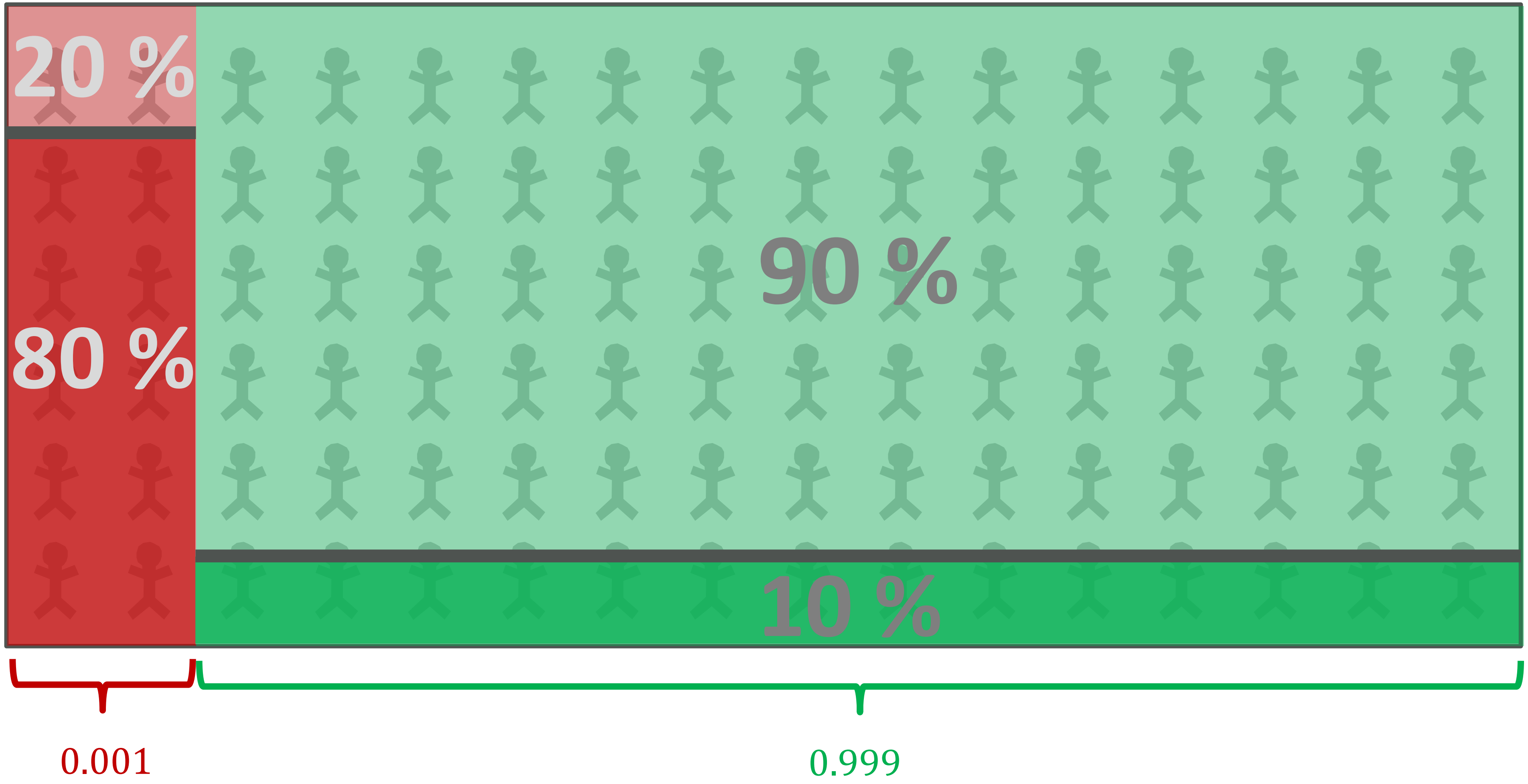
Assume 10 000 people

Actually healthy people
(9 990 / 10 000)

Tested „negative“
(healthy)

Tested „positive“
(sick)

Actually sick people
(10 / 10 000)



Evaluation

We use statistical methods to evaluate the performance of our model

→Many metrics available, but only confusion matrix and accuracy are important for this course!

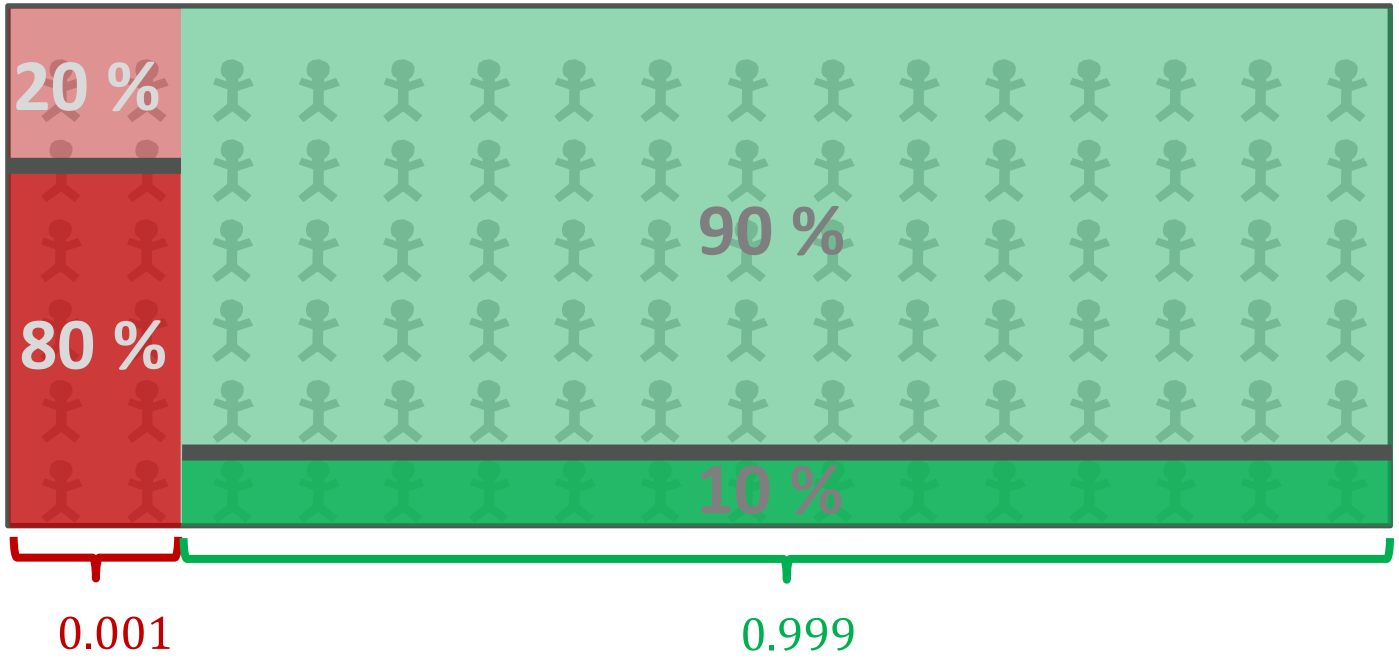
Prediction → **↓True Class**	**True**	**False**
True	# True Positive	# False Negative
False	# False Positive	# True Negative

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} = \frac{\#\ \text{Correct}}{\#\ \text{Samples}}$$

More specialized metrics can be calculated using the confusion matrix!

Relevant proportions of population

Actually sick people
(10 / 10 000)



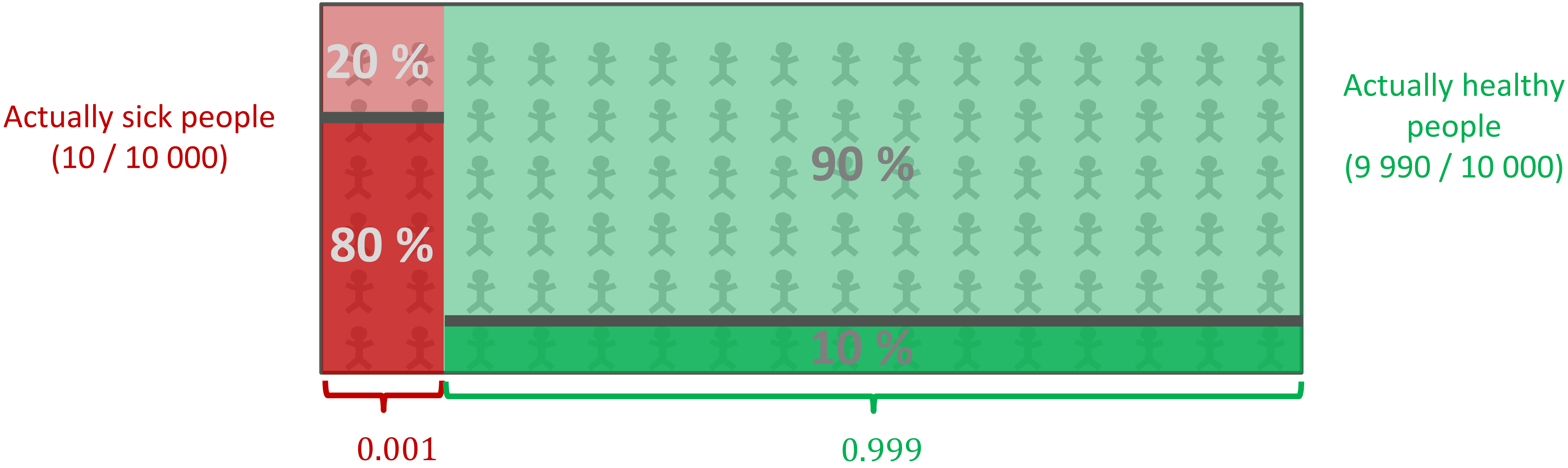
Actually healthy
people
(9 990 / 10 000)

8 sick people are classified as sick, 2 as healthy
→ 8 True Positives, 2 False Negatives

8991 healthy people are classified as healthy, 999 as sick
 8991 True Negatives, 999 False Positives

Prediction → ↓True Class	Sick	Healthy
Sick	8	2
Healthy	999	8991

Relevant proportions of population



$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$= \frac{8 + 8991}{8 + 999 + 8991 + 2} = 0{,}8999$$

Prediction → **↓True Class**	**Sick**	**Healthy**
Sick	8	2
Healthy	999	8991

Deep Learning: Details

Before we start

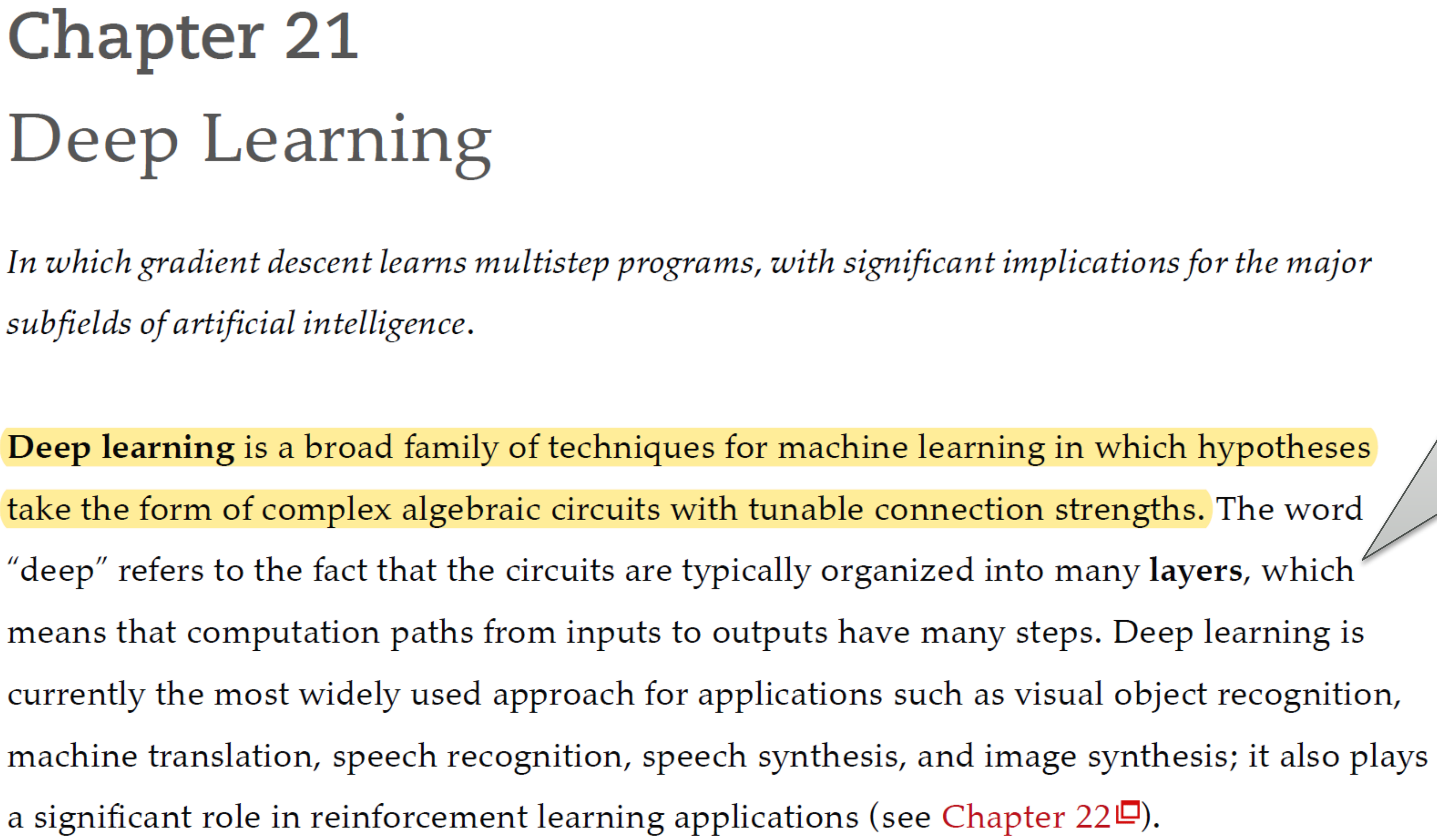
We want you to...

- Get an intuition for the topics AI, ML and DL
- Get a feeling for AI projects so that you can realistically assess possibilities and limits of AI
- Learn what well suiting problems and applications for AI are
- Understand that AI isn’t black magic (… well, mostly)

We don’t expect you to…

- Remember *every* mathematical detail
- Program the tasks completely alone

Definition of Deep Learning



Chapter 21

Deep Learning

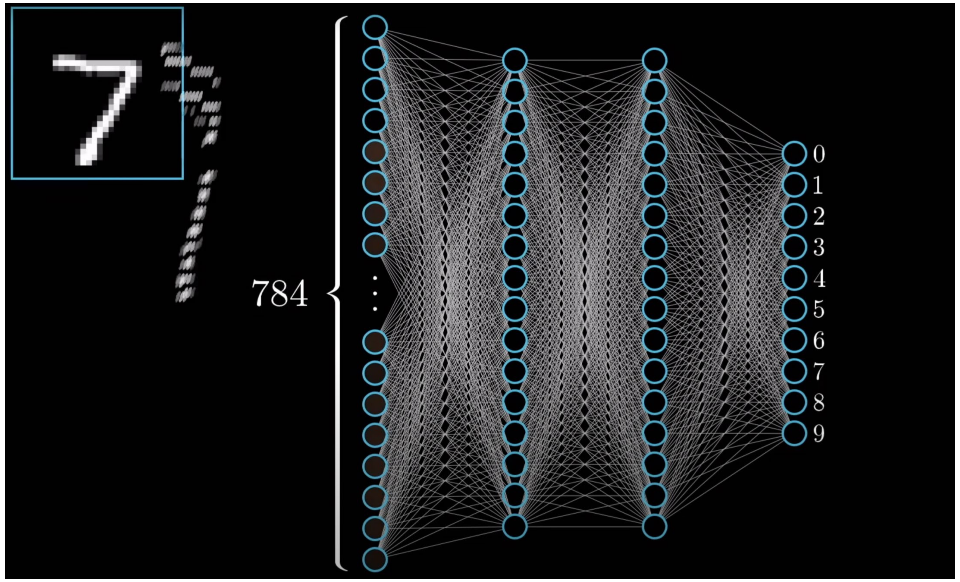
In which gradient descent learns multistep programs, with significant implications for the major subfields of artificial intelligence.

Deep learning is a broad family of techniques for machine learning in which hypotheses take the form of complex algebraic circuits with tunable connection strengths. The word "deep" refers to the fact that the circuits are typically organized into many **layers**, which means that computation paths from inputs to outputs have many steps. Deep learning is currently the most widely used approach for applications such as visual object recognition, machine translation, speech recognition, speech synthesis, and image synthesis; it also plays a significant role in reinforcement learning applications (see Chapter 22).

Then called „(artificial) neural networks"!

[Russel & Norvig: AI, A modern Approach, 4th ed 2021]

Self-Study Time – “What are neural networks?”

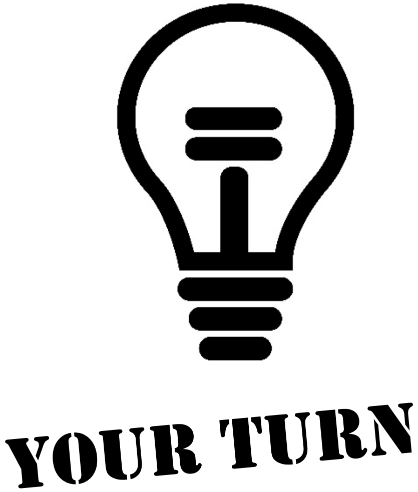


3blue1brown – Neural networks series:

https://www.youtube.com/watch?v=aircAruvnKk

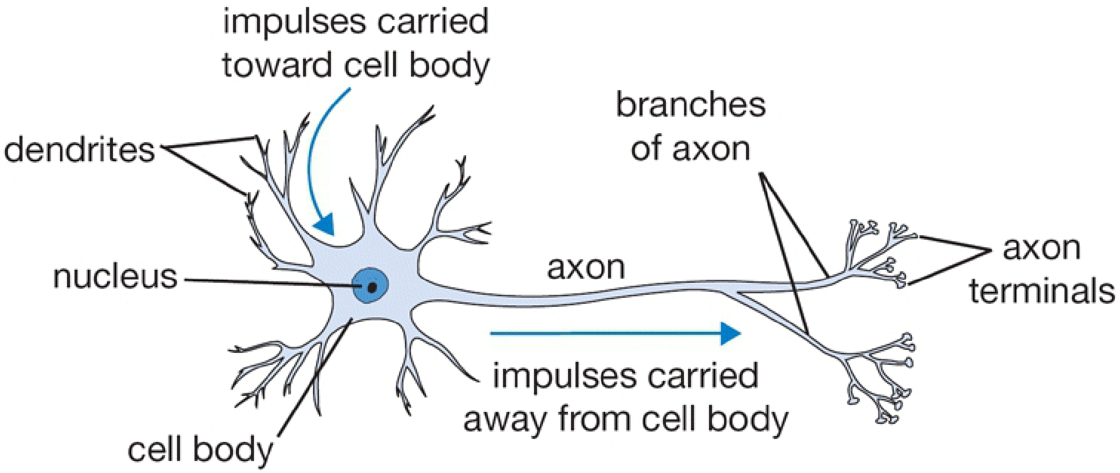
Tasks:

- Write a short summary of the video for your own notes
- Note 2 core intuitions explained by 3blue1brown

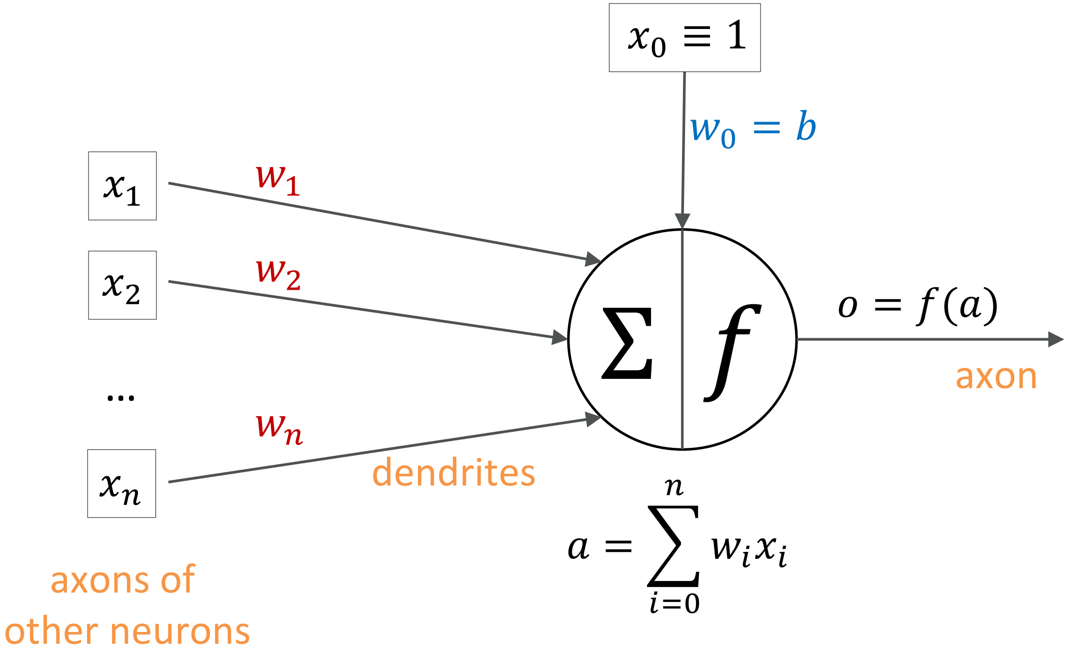


Inspiration for (Artifical) Neural Networks comes from Biology

- Deep Learning initially tried to model the human brain
- Neural Networks are trained using optimization techniques



impulses carried toward cell body
dendrites
branches of axon
axon terminals
nucleus
axon
impulses carried away from cell body
cell body

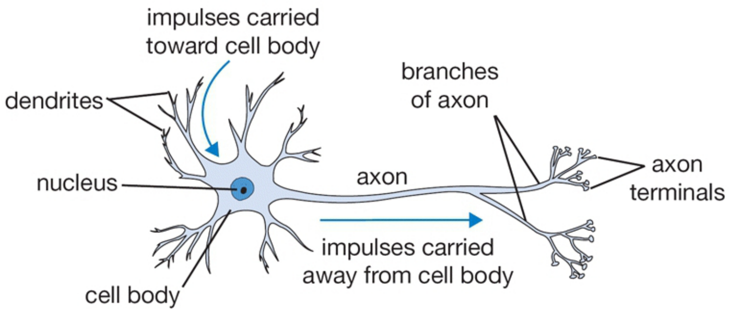


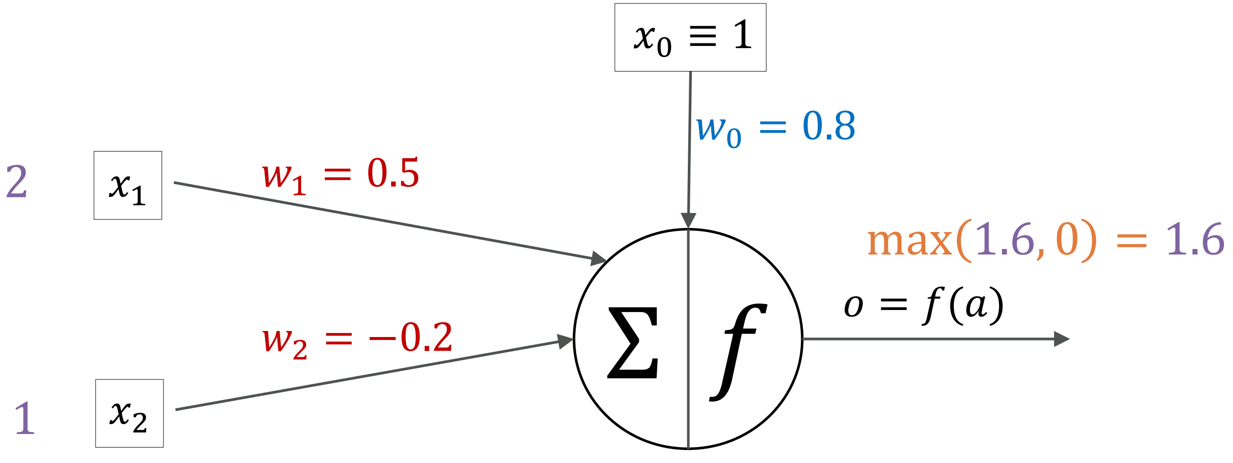
$$a = \sum_{i=0}^{n} w_i x_i$$

McCulloch and Pitts, 1943

x_i:	inputs	w_i:	weights
b:	bias weight	$f(z)$:	activation function
a:	activation	o:	output

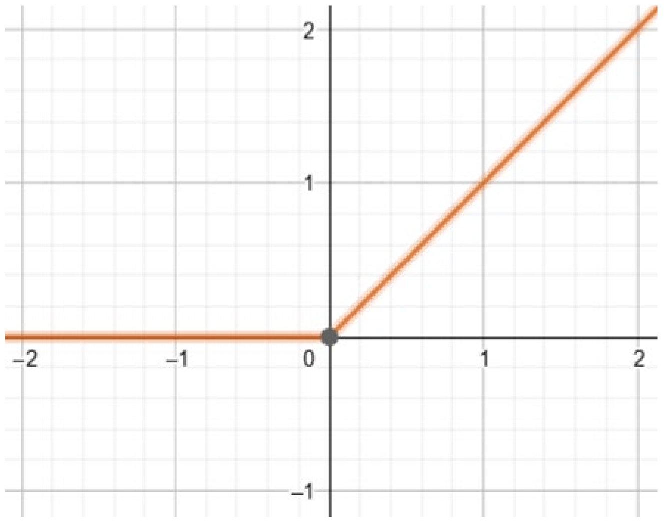
Example (single) Artificial Neuron


impulses carried toward cell body
dendrites
branches of axon
nucleus
axon
axon terminals
impulses carried away from cell body
cell body



$$a = \sum_{i=0}^{n} w_i x_i$$

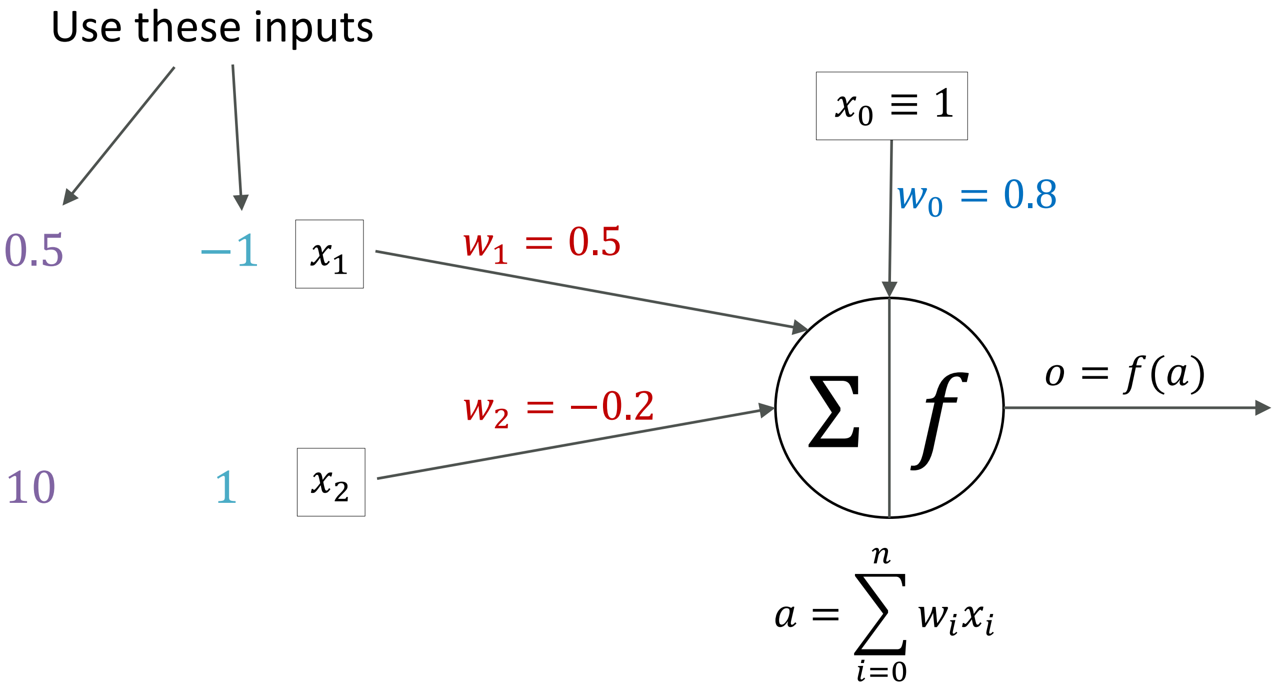
$$2 \cdot 0.5 + 1 \cdot (-0.2) + 0.8 = 1.6$$



$f(z) = \max(z, 0)$

x_i:	inputs	w_i:	weights
b:	bias weight	$f(z)$:	activation function
a:	activation	o:	output

Try it yourself!



Use these inputs

0.5 -1 x_1 $w_1 = 0.5$

10 1 x_2 $w_2 = -0.2$

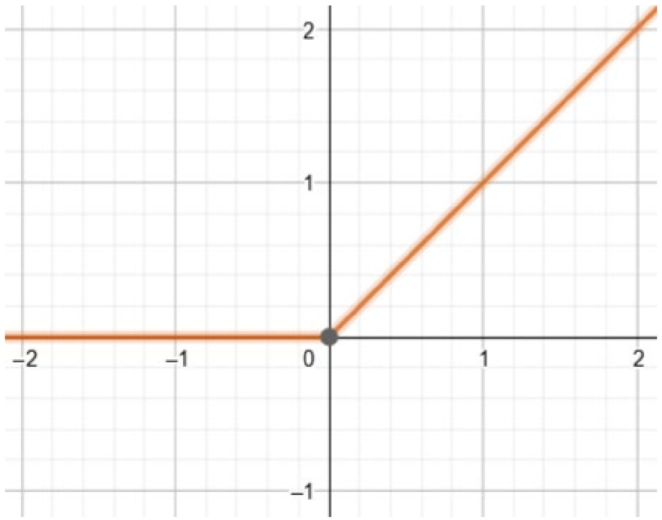
$x_0 \equiv 1$

$w_0 = 0.8$

Σ f

$o = f(a)$

$$a = \sum_{i=0}^{n} w_i x_i$$



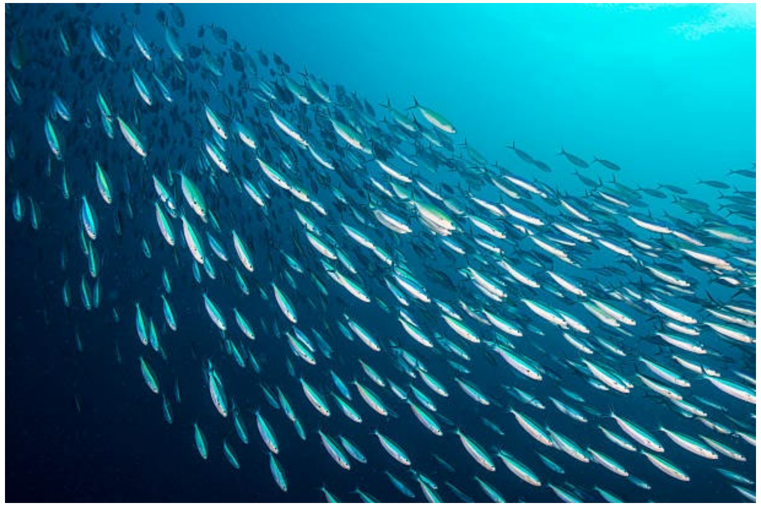
$f(z) = \max(z, 0)$

x_i:	inputs	w_i:	weights
b:	bias weight	$f(z)$:	activation function
a:	activation	o:	output

Why Neural *Networks*?

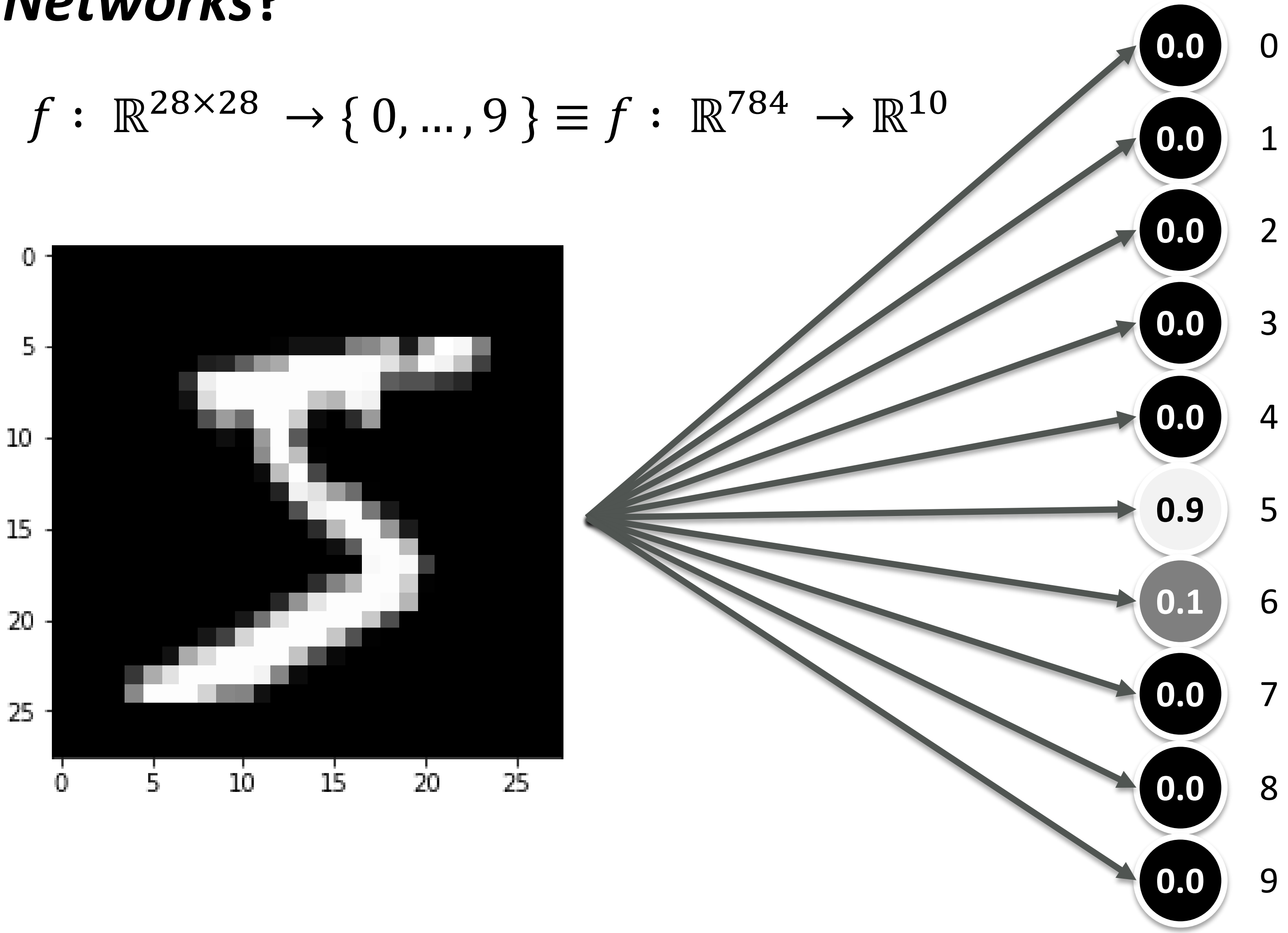
- Important finding during period of connectionism: Multiple perceptrons are more powerful than a single perceptron

- Similar to examples from nature:



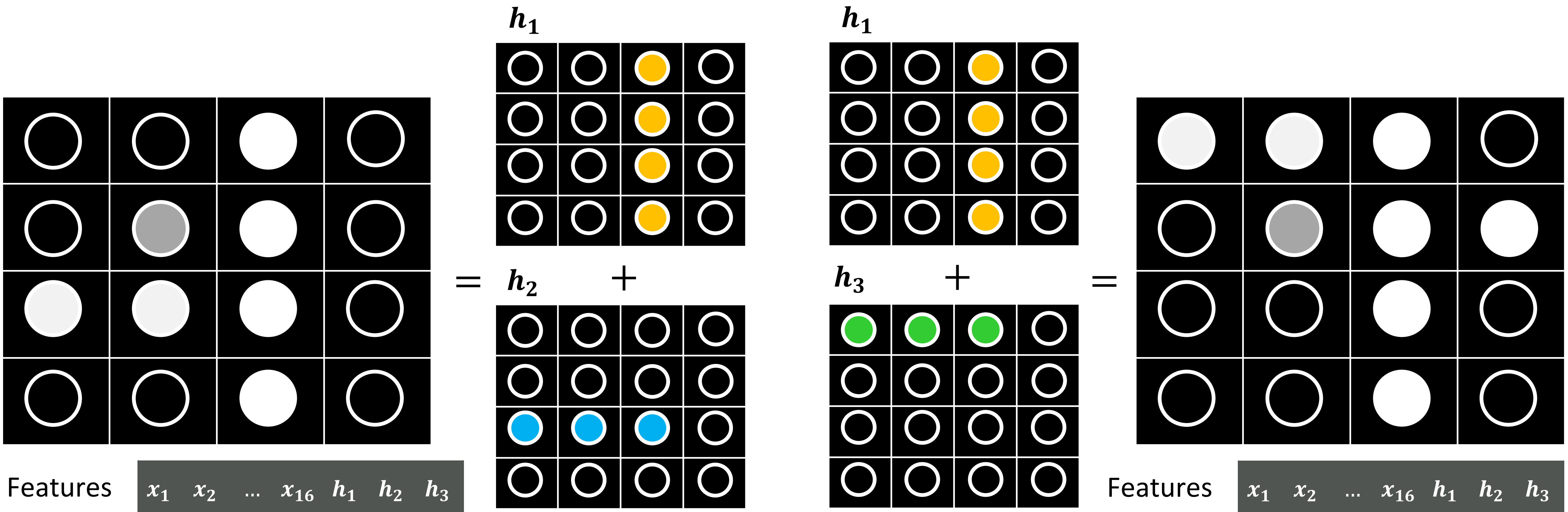
Why Neural *Networks*?

$$f : \mathbb{R}^{28\times 28} \rightarrow \{\, 0, \ldots, 9 \,\} \equiv f : \mathbb{R}^{784} \rightarrow \mathbb{R}^{10}$$



Why Neural *Networks*?

What makes a „4“ a „four“ and a „7“ a „seven“?

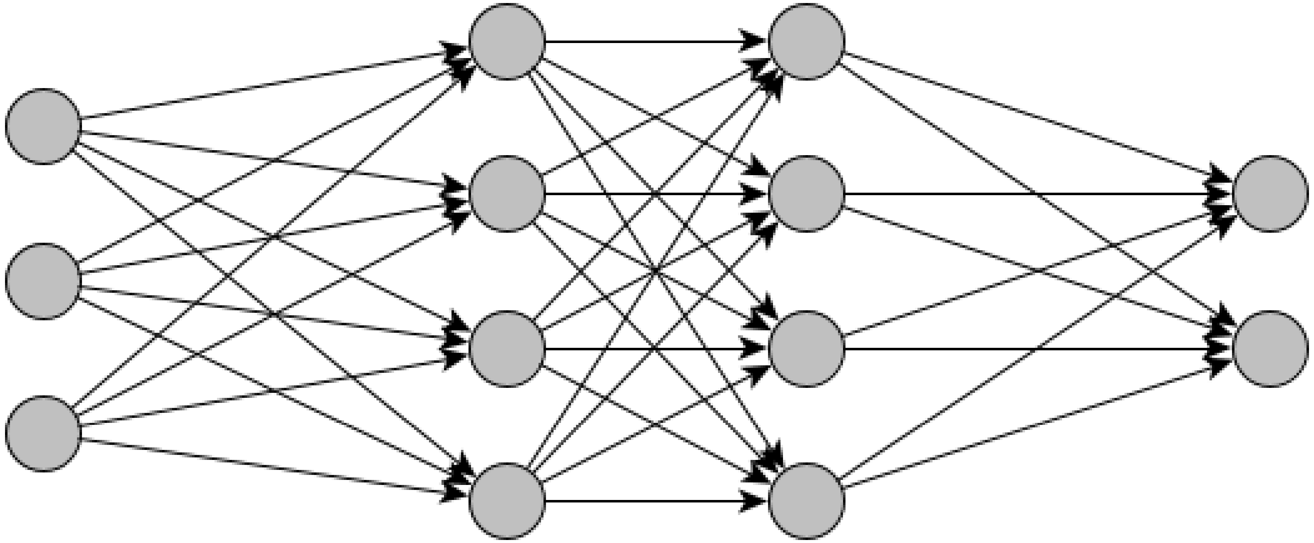


Features	x_1	x_2	…	x_{16}	h_1	h_2	h_3
Values	0.0	0.0	…	0.0	**1.0**	**1.0**	**0.0**

Features	x_1	x_2	…	x_{16}	h_1	h_2	h_3
Values	0.8	0.8	…	0.0	**1.0**	**0.0**	**1.0**

Why Neural *Networks*?

- We use multiple neurons in parallel („layer“)
- We can stack multiple layers
- In the simplest form, each neurons from one layer is connected with all neurons from the previous and following layer

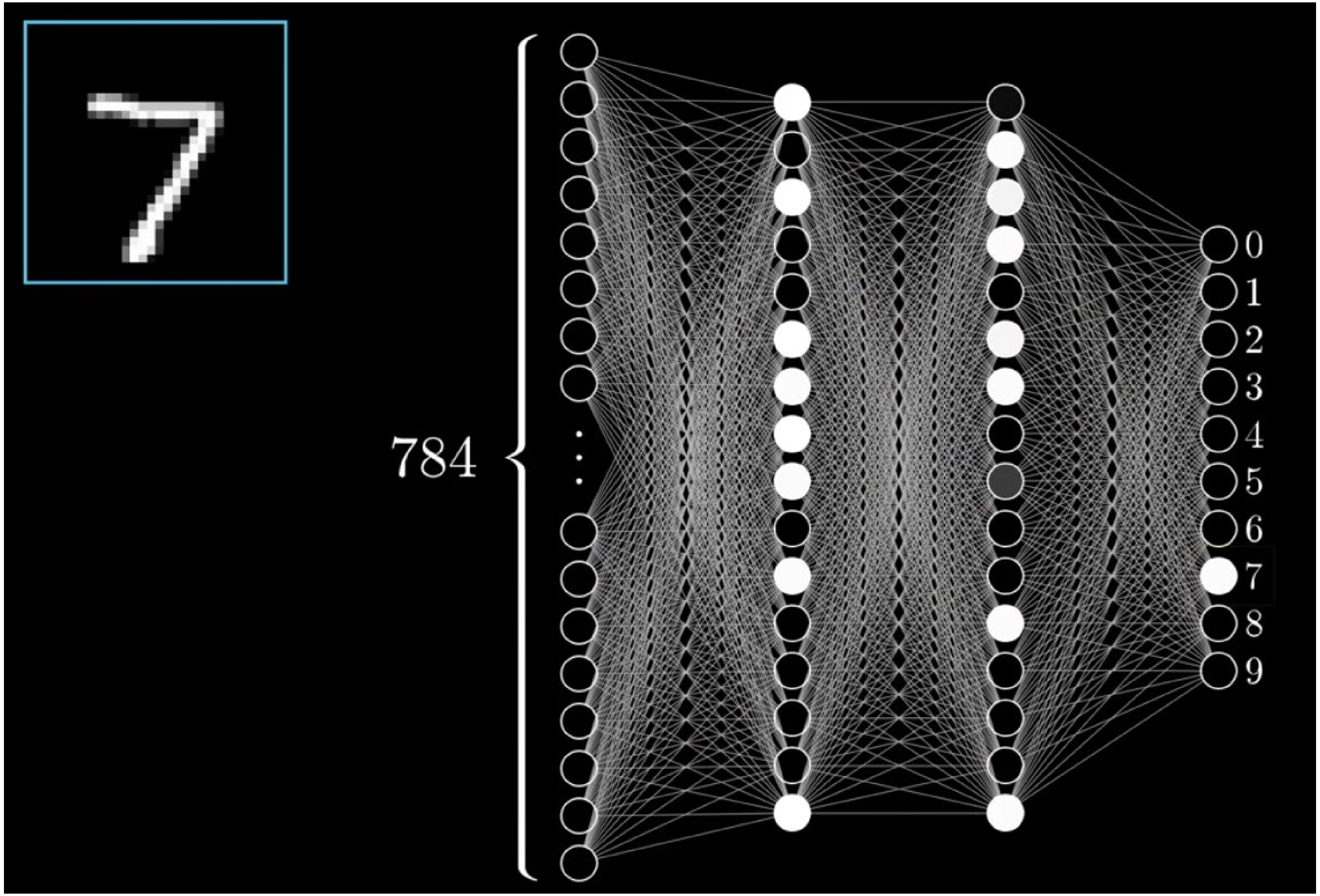


Input layer

Hidden layer

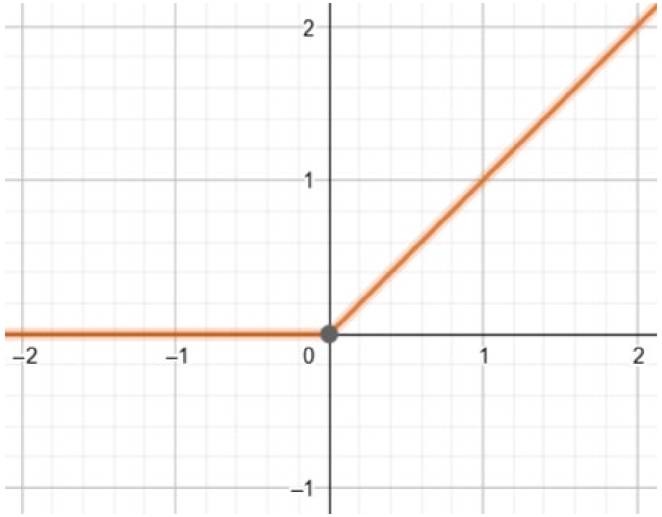
Output layer, Units are also called „logits“ in classification

Why Neural *Networks*?

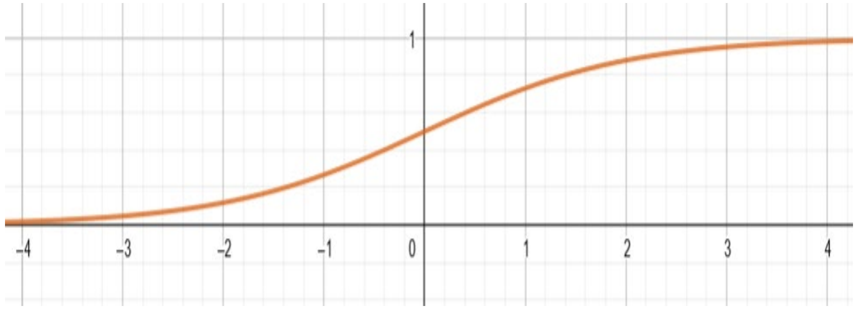


Activation Functions

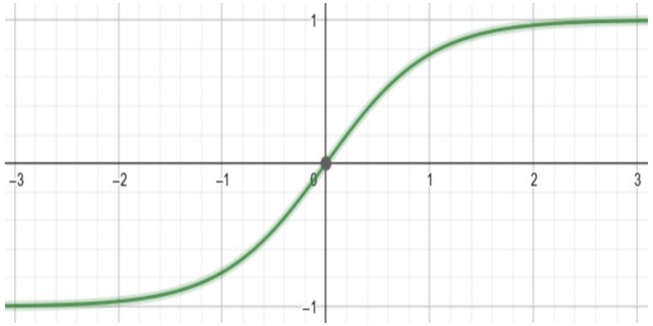
- The usage of activation functions is also motivated from nature
- Historic activation function: Step-function, Sigmoid
- Mathematical properties are not ideal, therefore we use continuous activation functions
- **Examples**:



ReLU



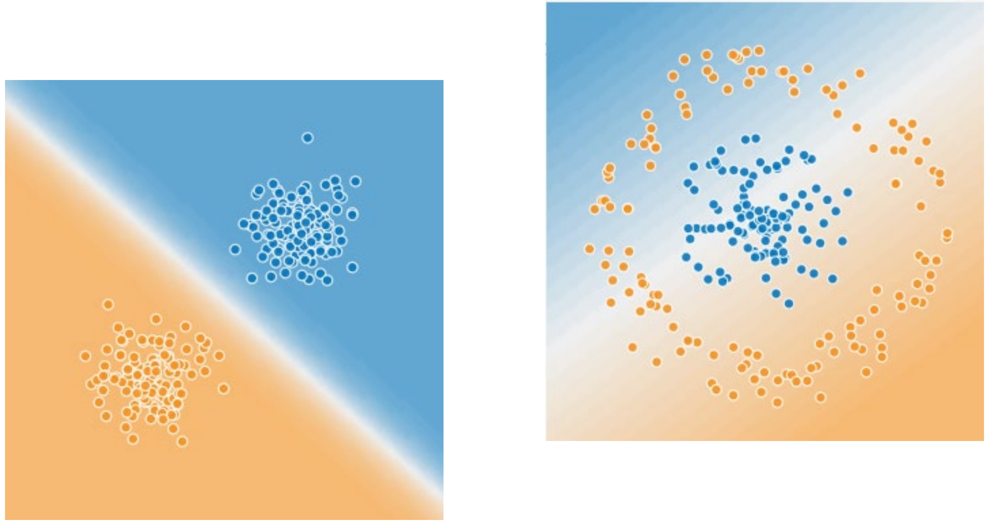
Sigmoid



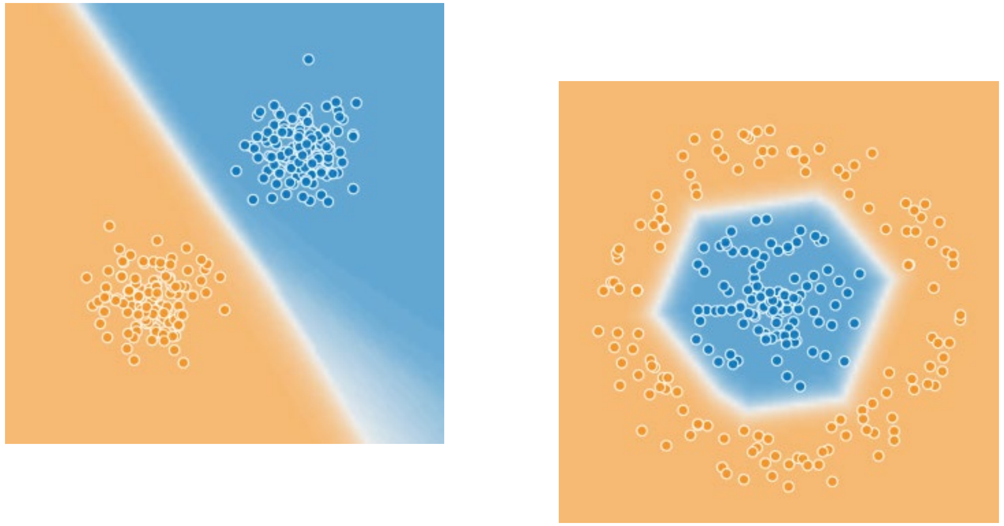
tanh

Activation Functions

- Using nonlinear activation functions is very important!
- Stacked linear functions are still *a linear function*
- With a linear function, we can only solve linearly separable decision problems



Linear activation function



Non-Linear activation function

Small discussion of Deep Learning

+ Neural Networks outperform „classic“ ML algorithms in many areas
+ Deep Learning allows for new areas of application
+ Neural Networks work especially well with implicit knowledge, i.e., many examples with labels instead of written “rules”

- Not always rational to use neural networks
- Often need more data than „classic“ ML algorithms
- Training and inference of reasonable sized neural networks for image or voice processing are often computationally heavy

How are neural networks programmed?

- Neural Networks and their training are complex mathematical problems
- Computational optimizations are often crucial

- Fortunately, no one has to write neural networks from scratch!

scikit learn
NumPy
Keras
TensorFlow
PyTorch

How are neural networks programmed?

- **Problem**: High requirements for memory and computational resources
- Trick for speeding up computations: **GPUs**!
  - ⍰ Neural Networks involve many parallelizable computations

- Alternative: Use cloud providers like Google, Microsoft or Amazon
- Special Case: Google Colab ⍰ Cloud service for iPython Notebooks

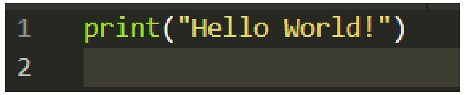
Azure

aws

Machine Learning in Python

- Python is the most widely used (frontend) language for ML/DL!

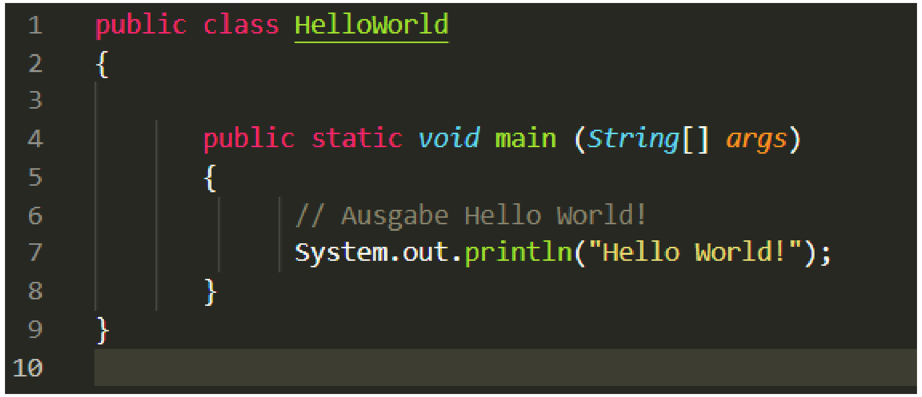
- **Characteristics**:
  - Beginner friendly
  - Simple syntax, focus lies on readability– no need to specify data types
  - Powerful statements: You can often write multi-line expressions from other languages in 1 line



```python
print("Hello World!")

```

Python



```java
public class HelloWorld
{

    public static void main (String[] args)
    {
        // Ausgabe Hello World!
        System.out.println("Hello World!");
    }
}

```

Java

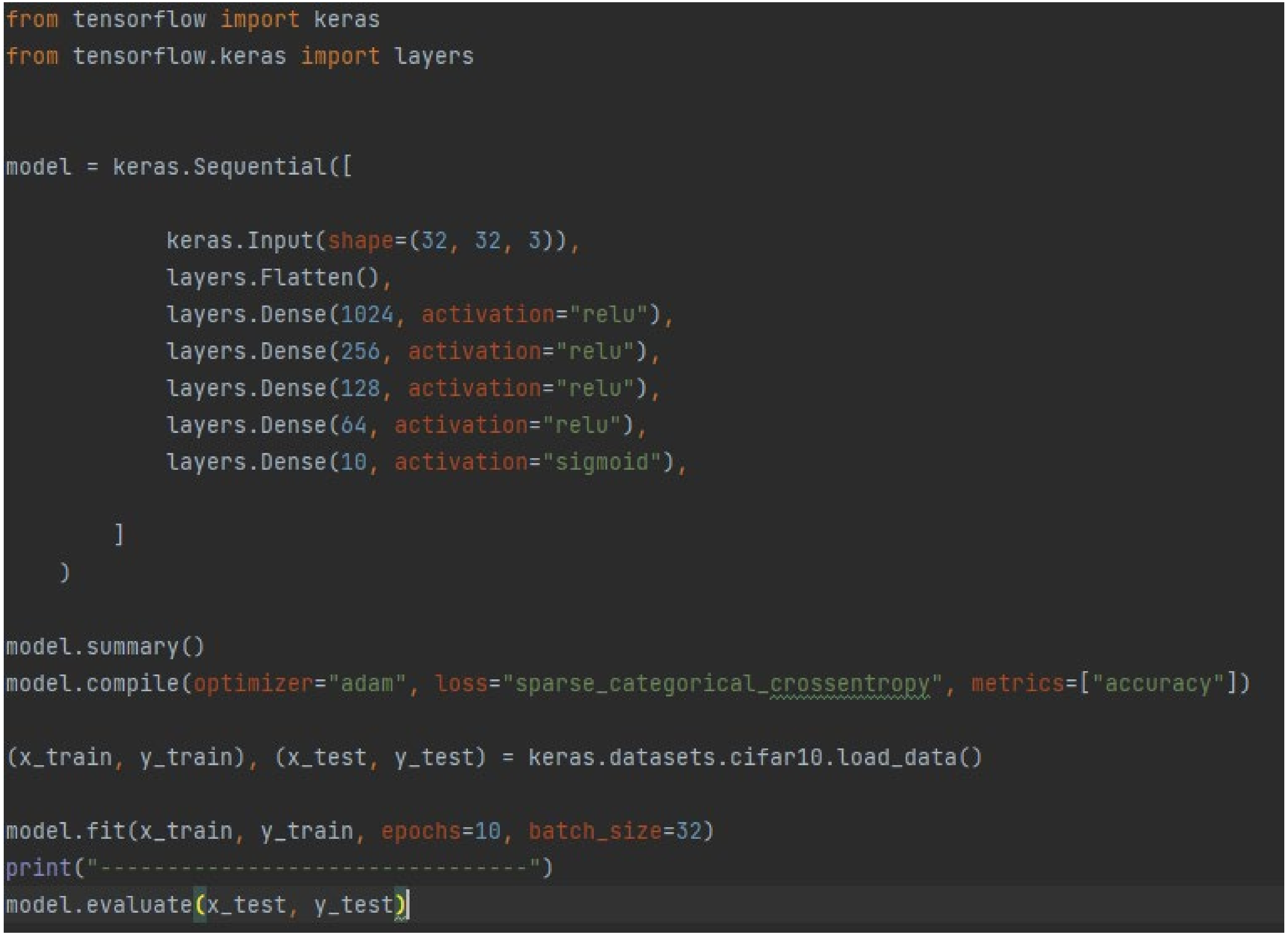
ML & DL Frameworks in Python

- *Numpy*: Base Framework for mathematical computations (matrices, vectors)
- Scikit-learn: general ML Framework, especially important for classic ML and for evaluation

- „Clash of Frameworks“: Tensorflow/Keras (Google) vs PyTorch (Facebook)
  - Both have their pros and cons
  - Opinion: Keras for beginners, Tensorflow for production systems, PyTorch for research

→We will use **Keras** (integrated in Tensorflow)

→ *(in our own research, we tend to use PyTorch)*



```python
from tensorflow import keras
from tensorflow.keras import layers


model = keras.Sequential([

            keras.Input(shape=(32, 32, 3)),
            layers.Flatten(),
            layers.Dense(1024, activation="relu"),
            layers.Dense(256, activation="relu"),
            layers.Dense(128, activation="relu"),
            layers.Dense(64, activation="relu"),
            layers.Dense(10, activation="sigmoid"),

        ]
    )

model.summary()
model.compile(optimizer="adam", loss="sparse_categorical_crossentropy", metrics=["accuracy"])

(x_train, y_train), (x_test, y_test) = keras.datasets.cifar10.load_data()

model.fit(x_train, y_train, epochs=10, batch_size=32)
print("--------------------------------")
model.evaluate(x_test, y_test)
```

Link to Notebook

https://colab.research.google.com/github/AImotion-Bavaria/dl-bootcamp/blob/main/%5BSolution_FCN%5DCIFAR_10.ipynb

Solving the GTSRB: Loading the data

Preliminary Task

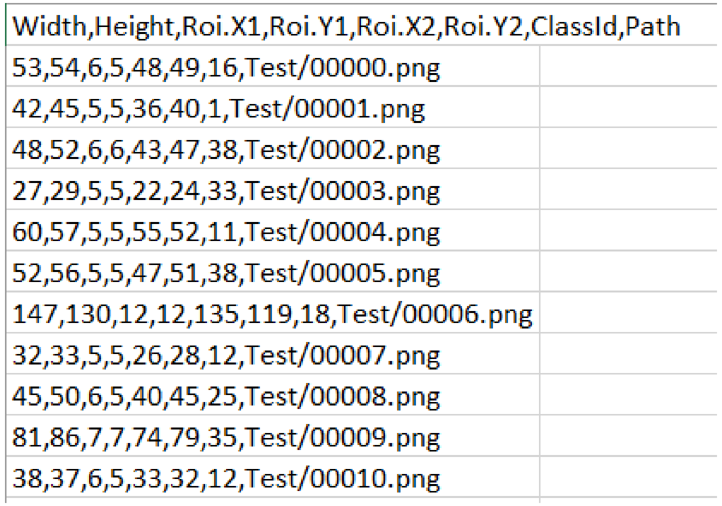
- First step in every ML Project: ***Data exploration***!

- Task: Download the data set from the following link:
https://moodle.thi.de/mod/resource/view.php?id=315239

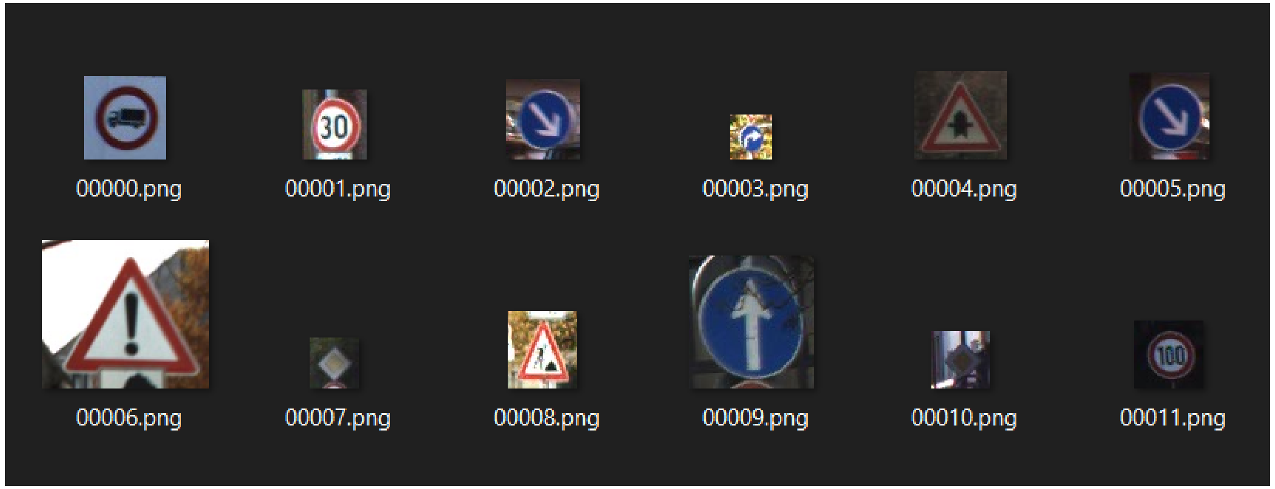
- Extract the files on your computer and have a look at the dataset:
  - How is the data set structured?
  - Where are the necessary files?
  - What additional information is given?

The data set

- The GTRSB data set is structured in the following way:
  - Meta: synthetic examples for each class
  - Test: unstructured examples for testing our model
  - Train: labelled examples (one folder per class) for training
  - Meta.csv, Train.csv, Test.csv: Additional information about the images
    - ⍰ Very important for us!



```
Width,Height,Roi.X1,Roi.Y1,Roi.X2,Roi.Y2,ClassId,Path
53,54,6,5,48,49,16,Test/00000.png
42,45,5,5,36,40,1,Test/00001.png
48,52,6,6,43,47,38,Test/00002.png
27,29,5,5,22,24,33,Test/00003.png
60,57,5,5,55,52,11,Test/00004.png
52,56,5,5,47,51,38,Test/00005.png
147,130,12,12,135,119,18,Test/00006.png
32,33,5,5,26,28,12,Test/00007.png
45,50,6,5,40,45,25,Test/00008.png
81,86,7,7,74,79,35,Test/00009.png
38,37,6,5,33,32,12,Test/00010.png
```

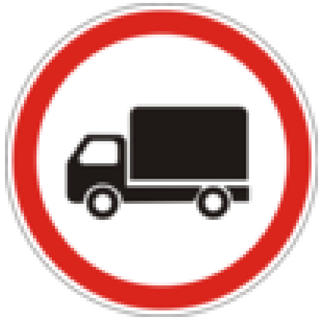


The data set

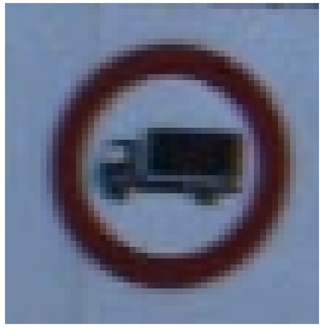
From Test.csv:

Width,Height,Roi.X1,Roi.Y1,Roi.X2,Roi.Y2,ClassId,Path

53,54,6,5,48,49,16,Test/00000.png



Meta/16.png



Test/00000.png

Data exploration part 2

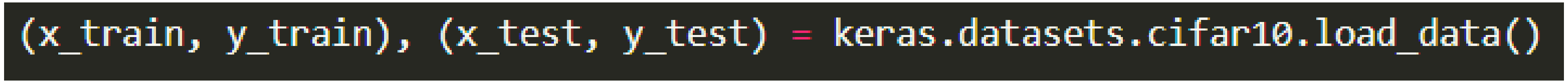
- Now have a look at the files themselves

- What do you think:
  - Where will problems (w.r.t. classification) occur?
  - Which instances/classes are similar?
  - Do you think it is a hard classification problem?

 Write down your answers, we will compare them with the results!

Loading the data

- Remember this line? We have to do it by ourselves now!



```
(x_train, y_train), (x_test, y_test) = keras.datasets.cifar10.load_data()
```

- Idea: The *.csv files provide us the paths an labels for the images
  - → We want to read the *.csv files
  - → We want to map the file paths to loaded images
  - → We want to extract the labels

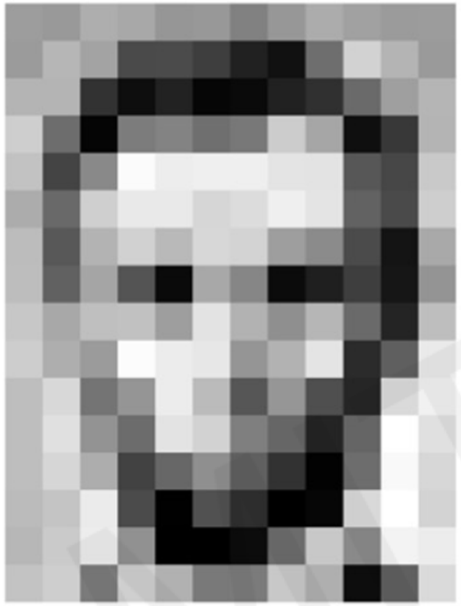
How can computers see?

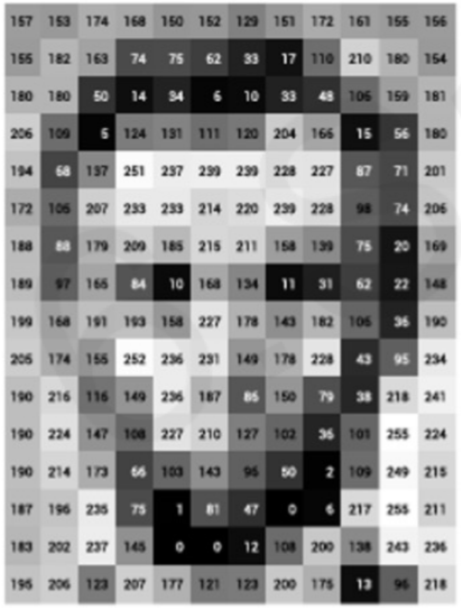
- Computers use pictures taken by a digital camera
- A digital picture is usually represented as a tensor:
  - (Height, Width, 1) in case of grayscale pictures
  - (Height, Width, 3) in case of RGB pictures

Tip: RGB means „Red – Green – Blue“. Each color is represented by a value in [0, 255], e.g. White = (255, 255, 255)

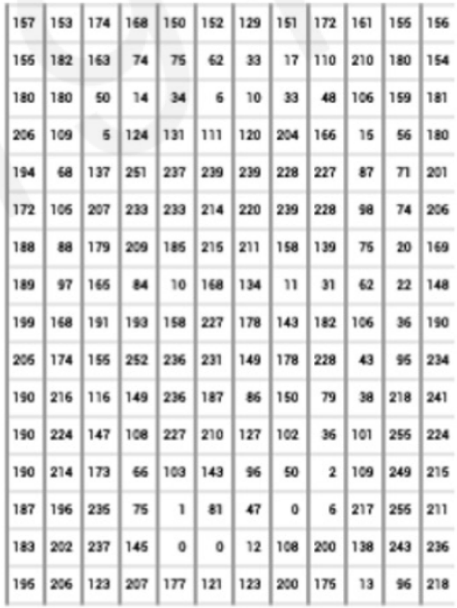
→ 256*256*256 ≈ 16,8 mio different color can be respresented

→ In Grayscale it‘s just one value from [0, 255]. 0 means „black“, 255 means „white“





157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	95	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

In case of RGB pictures: 2 additional planes, each representing a RGB channel

Image Credit: introtodeeplearning.com – Lec. 3 Deep Computer Vision

Input Image In Channels

Original image (RGB)

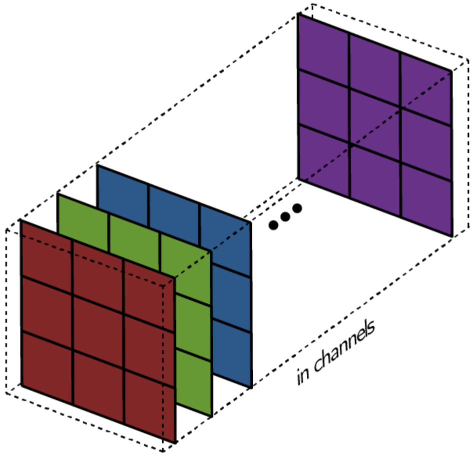
R channel

G channel

B channel

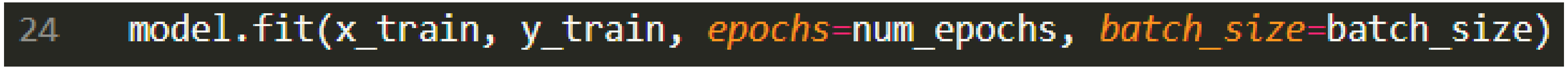
An image is a tensor with dimensions: $\mathbb{R}^{w \times h \times d}$

w(idth)
h(eight)
d(epth)



Loading the data

- If we load an image, it is represented as a matrix/tensor of values
- How should we represent the whole dataset?



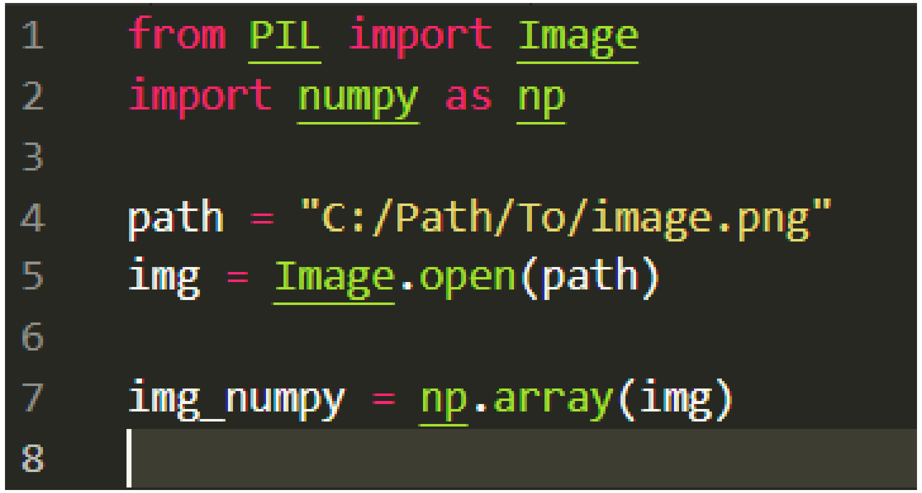
```
model.fit(x_train, y_train, epochs=num_epochs, batch_size=batch_size)
```

- Keras allows these to be numpy arrays of all instances

→We want to construct a huge array:
 (#samples, img_width, img_height, 3)

Loading the data

- How can we load images into a Python program and turn it into a numpy array?



```python
from PIL import Image
import numpy as np

path = "C:/Path/To/image.png"
img = Image.open(path)

img_numpy = np.array(img)

```