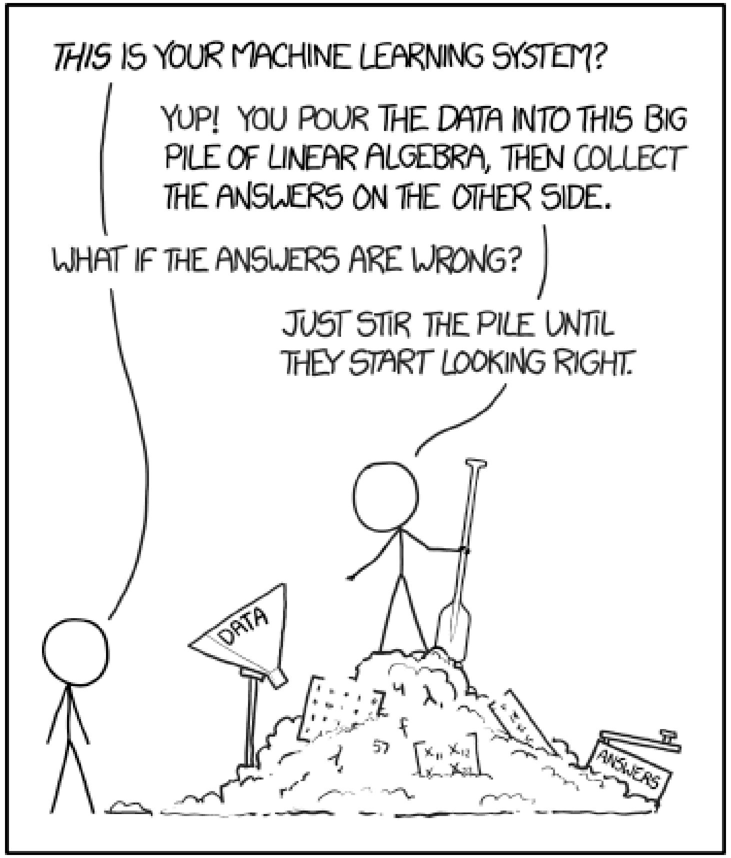


Image: xkcd

Image: Cover, Nature Vol 518, No. 7540, 26 Feb. 2015

Machine Learning is the science "concerned with the question of how to construct computer programs that automatically improve with experience"

- Tom Mitchell (1997) CMU

AI101

Lecture 11: Introduction into Machine Learning and Neural Networks

Recap
Machine Ethics

Challenges:

- Balancing autonomy and control.
- Addressing bias and fairness in algorithms.
- Ensuring transparency and accountability.
- Navigating privacy concerns.
- Accountability and liability for AI actions.
- Privacy preservation in data-driven systems.
- Fairness and avoiding discrimination in algorithmic decision-making.
- Development of ethical guidelines and codes of conduct for AI practitioners.
- Integration of ethical considerations in the design process.
- Collaboration between researchers, policymakers, and industry to establish standards.

Emerging Issues:

- Ethical dilemmas in AI decision-making.
- Impact on employment and societal structures.
- Global perspectives on machine ethics.

What is Learning?

00

Learning. What is Learning?

1. Learning is essential for dealing with unknown environments
   - What happens if our agent is not omniscient?

2. Learning is useful as a system construction method
   - Expose the agent to reality rather than trying to write it down.

3. Learning modifies the agent decision mechanisms to improve performance
   - “Insanity is doing the same thing over and over again and expecting different results”.

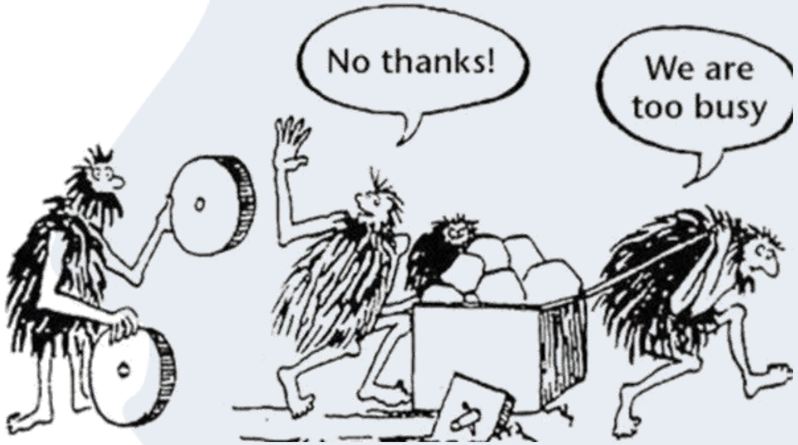


Image: original source unknown, https://www.jackvinson.com/blog/2016/8/11/no-thanks-too-busy

Learning

How are Things Learned?

Memorization (Declarative Knowledge)

- Accumulation of individual facts
- Limited by
    - Time to observe facts
    - Memory to store facts

Generalization (Imperative Knowledge)

- Deduce new facts from old facts
- Limited by accuracy of deduction process
    - Essentially a predictive activity
    - Assumes relations between past and future

Generalization, not Memorization

Example:

Day 1: 10h

Day 2: 10h → **How long did you study on Day 2?**

Day 3: 9h

Day 4: 2h

…

Day 31: 5h

→ **How long did you study on Day 32?**

Learning
Inductive Learning

Inductive Learning

Inductive Learning is the simplest form of learning. It learns a function from examples.

Idea:

- *f* is the (unknown) **target function** we want to learn. Then *f* (x) is called target, label or y
- Examples are defined as (x, f(x)), i.e. (1, Rain)

Problem: find a hypothesis *h*, such that *h* ≈ *f*

- Given a training set of examples
- On *all* examples

Example:

Day 1: 10h

Day 2: 10h → $f(2) = 10$

Day 3: 9h

Day 4: 2h

…

Day 31: 5h → $f(32) = 6$

Learning
Predicting the future

Construct a „rule set“ *h* to agree with *f* on training set

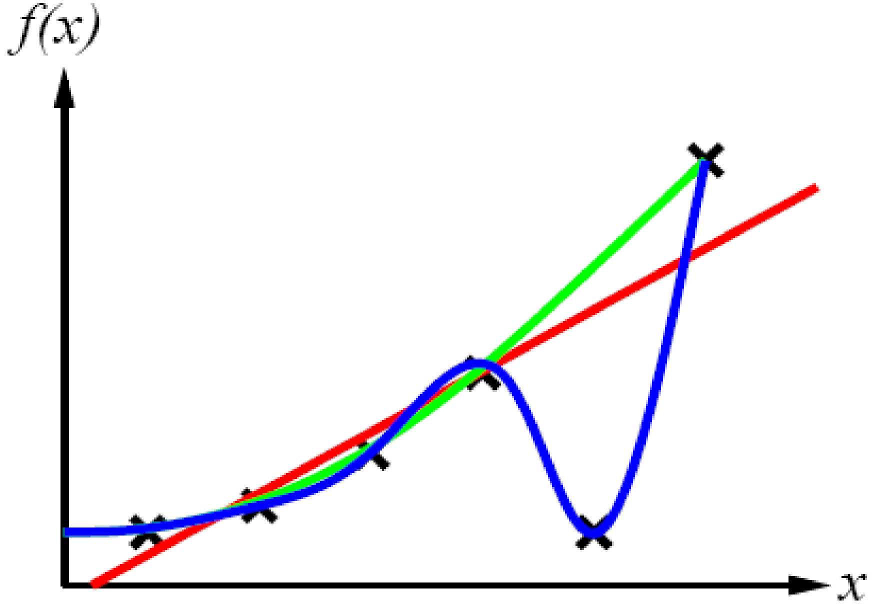
- *h* is consistent if it agrees with *f* on all examples

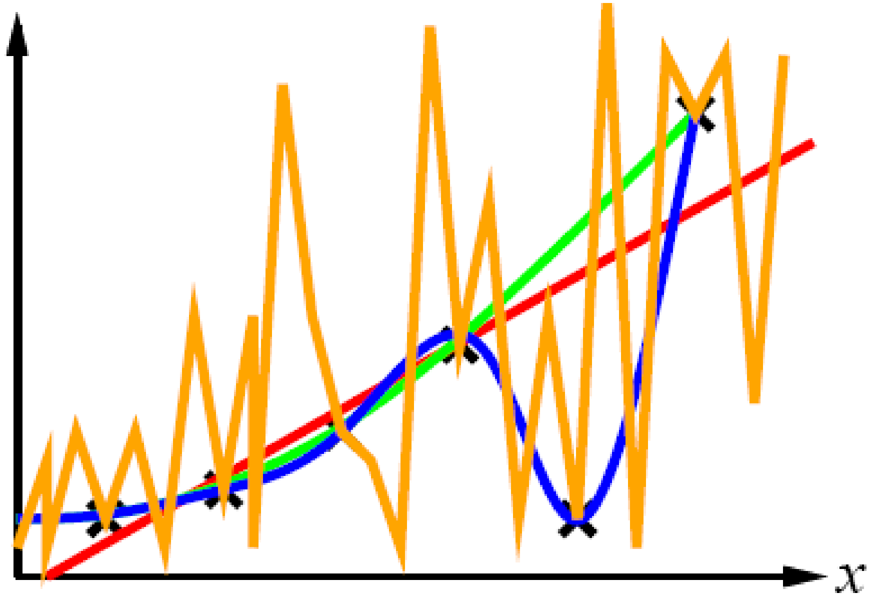
Ockham‘s Razor

- The best explanation is the simplest explanation that fits the data

Overfitting Avoidance

- Maximize the combination of consistency and simplicity



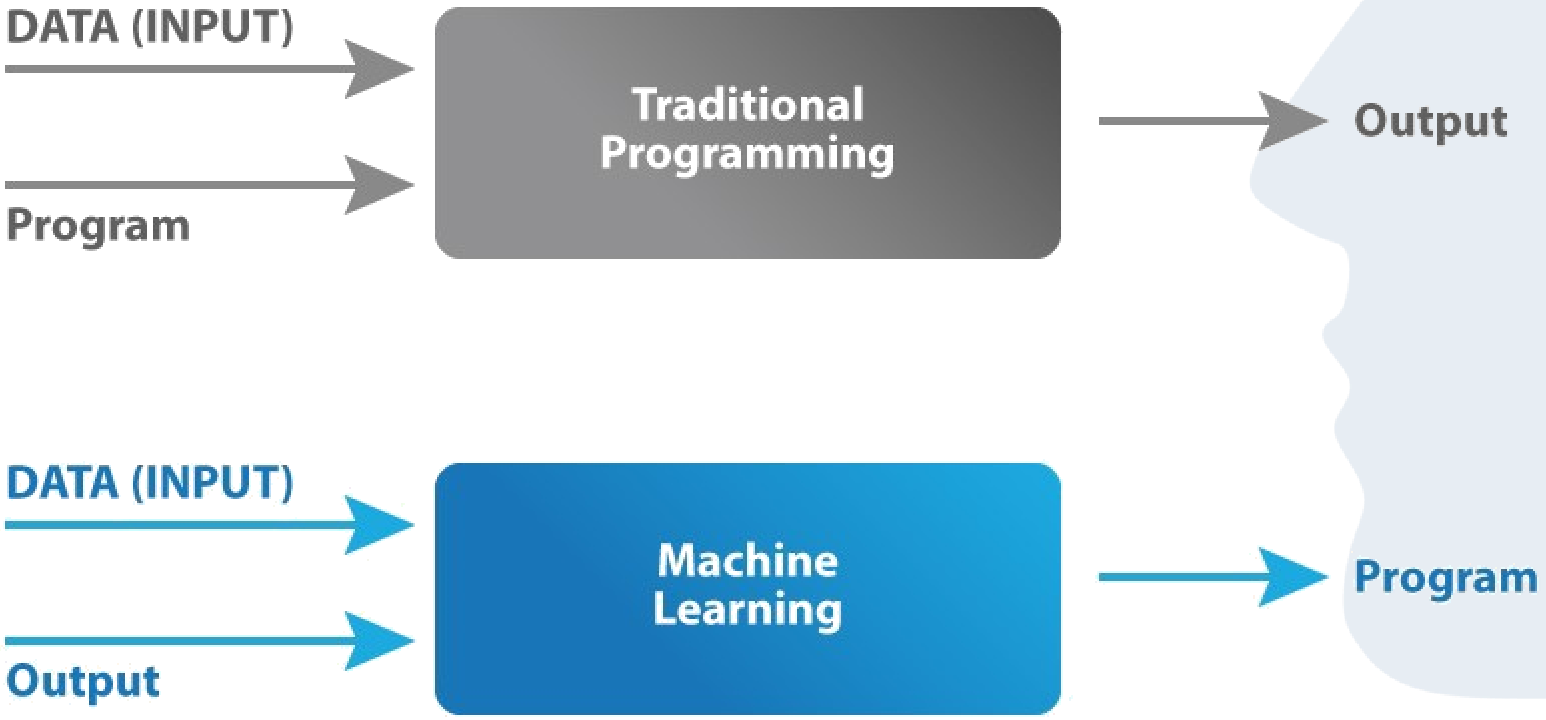


What is Machine Learning?

01

Machine Learning
What is Machine Learning

Machine Learning approach: Program an algorithm to automatically learn a program from data or experience



Machine Learning
Machine Learning and AI

Nowadays machine learning is often used as a synonym for artificial intelligence (AI) even if these are not the same (as you already know)!

AI does not always imply a learning-based system

- Search, CSP, Logical Inference, Rule-based Systems, Planning, …

Machine Learning focuses on learning based systems while often extracting knowledge from data.

Machine Learning
Machine Learning and Human Learning

Human Learning is (often)

- …very data- and knowledge-efficient
- …a complete multitasking, multi-modal system
- …time-inefficient, i.e. takes a lot of it

Machine Learning is often inspired by human learning but the goal is not to rebuild human learning.

- It may borrow ideas from biological systems, e.g. neural networks.
- It may perform better or worse than humans, it is far from perfect.

Machine Learning
Applications

- Web Search
- Computational Biology
- Speech Recognition, Machine Translations
- Image Recognition
- Robotics
- Finance and Stock Market
- Medical Diagnosis
- Information Extraction, Visual Analytics
- Traffic Prediction
- Software development
- …

Learning

Designing a Learning System

1. Do I need a learning approach for my problem?
   - Is there a pattern to detect?
   - Can I solve the problem analytically?
   - Do I have data to train on?
2. What type of problem do we have?
   - How to represent it?
   - Choose an algorithm based on the situation
3. Gather and organize your data
   - Preprocessing is important
4. Fitting/Training your model
5. Optimization
6. Evaluate and iterate back to step 2

Learning

Designing a Learning System

1. Do I need a learning approach for my problem?
   - Is there a pattern to detect?
   - Can I solve the problem analytically?
   - Do I have data to train on?
2. What type of problem do we have?
   - How to represent it?
   - Choose an algorithm based on the situation
3. Gather and organize your data
   - Preprocessing is important
4. Fitting/Training your model
5. Optimization
6. Evaluate and iterate back to step 2

ML is not the solution to all of our problems!

Get to a solution

02

Machine Learning

Types of Learning

Supervised Learning

Learning based on labeled datasets. It learns to map inputs to outputs based on the pairs in the dataset used in the learning process.

Unsupervised Learning

Unlike supervised training, unsupervised training uses unlabeled data to work with. It searches for patterns and similarities in data.

Reinforcement Learning

In RL an agent learns by interacting with its environment and getting a positive or negative reward.

Often there is a type called “semi-supervised Learning” which is a combination of the first two. Here only a subset of the examples are labeled

Machine Learning
Types of Learning

Supervised Learning

Learning based on labeled datasets. It learns to map inputs to outputs based on the pairs in the dataset used in the learning process.

Unsupervised Learning

Unlike supervised training, unsupervised training uses unlabeled data to work with. It searches for patterns and similarities in data.

Reinforcement Learning

In RL an agent learns by interacting with its environment and getting a positive or negative reward.

More next week

Often there is a type called "semi-supervised Learning" which is a combination of the first two. Here only a subset of the examples are labeled

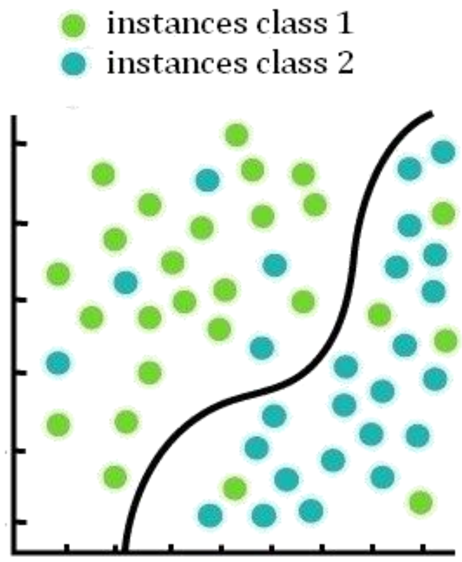
Machine Learning
Supervised Learning: Classification and Regression Tasks

Given a dataset, $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$

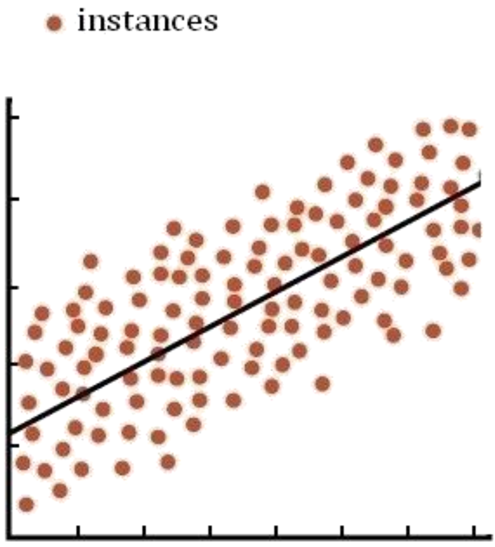
the **goal** is to learn a function *h(x)* to predict *y* given *x*

Regression Task
- *y* is a continious value
- Example: What is the temperature tomorrow?

Classification Task
- *y* is a discrete class label
- The algorithm tries to predict a continious value describing the label probability
- Example: Will it be above or below 0°C tomorrow?



Classification function



Regression function

Image: https://www.javatpoint.com/regression-vs-classification-in-machine-learning

Machine Learning
Representation

Machine Learning algorithms need to handle a lot of different data (e.g. images, audio, …)

Idea: Represent your input as an **input vector** (in R^n)

- Vectors are great since we can use linear algebra

Representation

Mapping your input to another space that is easy to manipulate.

Feature

Features are nothing but the <u>independent variables</u> in machine learning models.

Model

The representation of what our algorithm has learned from the data it used in the training process. The model is the output representation of the learned “rule set”

Machine Learning
Feature Engineering

Raw Data

```
0 : {
 house_info : {
 num_rooms: 6
 num_bedrooms: 3
 street_name: "Shorebird Way"
 num_basement_rooms: -1
 ...
 }
}
```

Raw data doesn't come to us as feature vectors.

Feature Engineering

Feature Vector

```
[
 6.0,
 1.0,
 0.0,
 0.0,
 0.0,
 9.321,
 -2.20,
 1.01,
 0.0,
 ...,
]
```

Process of creating features from raw data is **feature engineering**.

Label | Features

Example	Weather	Location	Date	Temperature	Precipitation
1	Rainy	Darmstadt	11.04.22	12.3	44.2
2	Sunny	Hamburg	11.04.22	19.2	12.6
3	Rainy	Darmstadt	12.04.22	16.7	67.3
4	Cloudy	Heidelberg	11.04.22	17.3	22.2
…	…	…	…	…	…

Feature Engineering

Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in within our learning approach.

Machine Learning
Feature Engineering

Know your data

- How is your data distribtution?
- Do you have outliers?
- Does your data reflect reality?
- Is your data biased?
- …

„Garbage In, Garbage Out“: Using bad data results in bad models, though noise per se may not be a problem

Machine Learning
Common Approaches

Regression:

- Linear Regression
- Multiple Linear Regression
- Regression Trees
- Non-linear Regression
- Polynomial Regression
- …

Classification:

- Random Forest
- Decision Trees
- Logistic Regression
- Naïve Bayes
- Support Vector Machines
- …

There is no single best model that works best for all problems.

More information about why there is no single best model: https://machinelearningmastery.com/no-free-lunch-theorem-for-machine-learning/

Evaluating your model

03

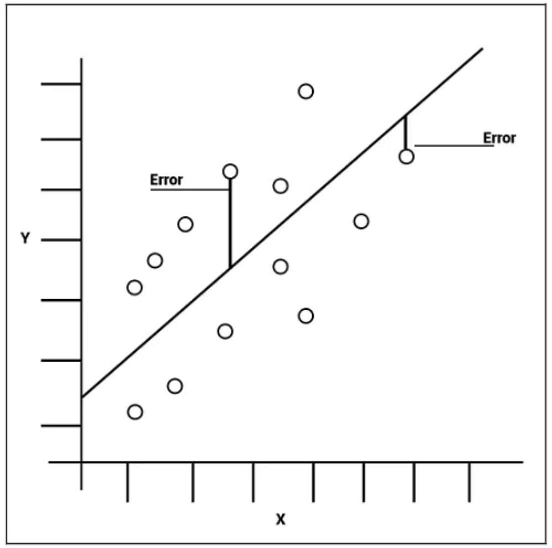
Machine Learning
Evaluation

To measure quality, you need to know the goal!

There are a lot of options:

- Accuracy
- Precision
- Recall
- Mean Squared Error
- …

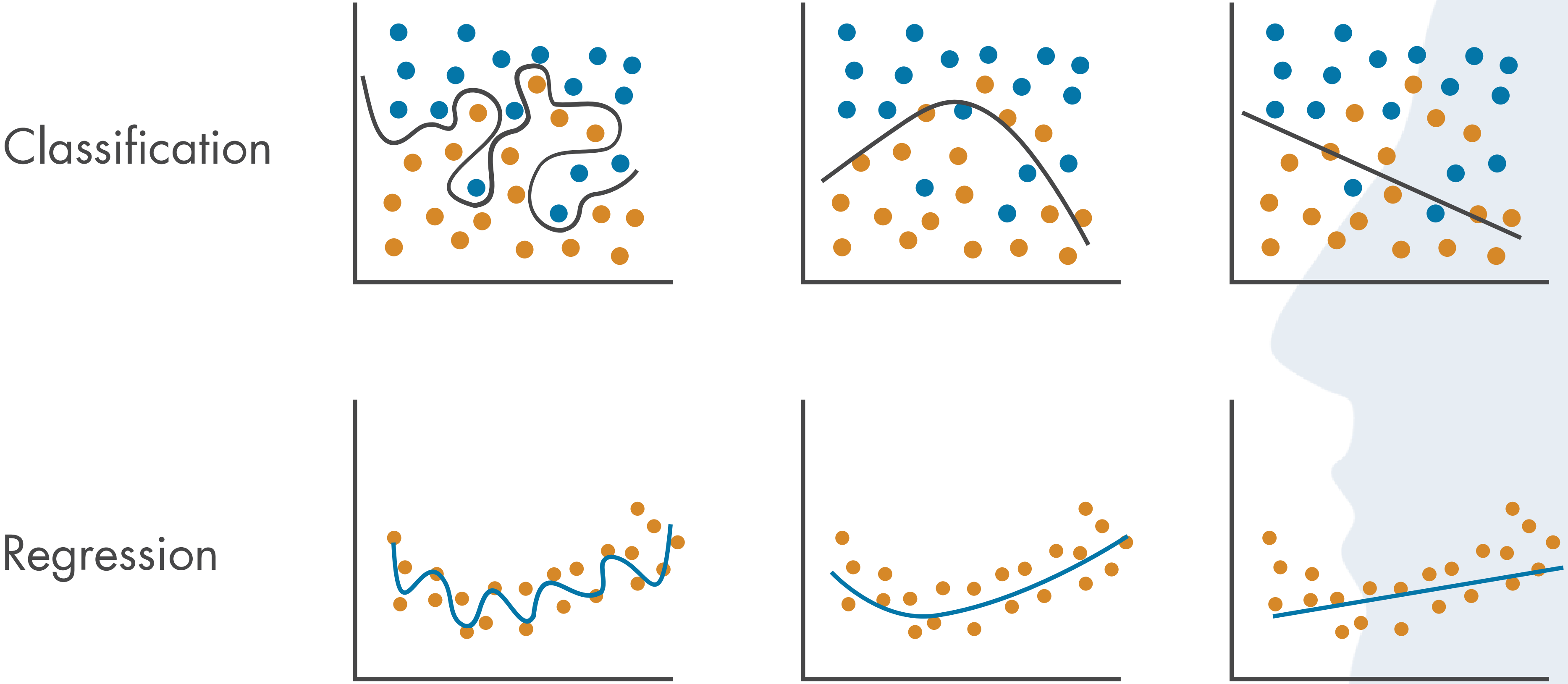
Mean Squared Error



$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Choosing the correct metric depends on the goal

Machine Learning
The good, the bad and the ugly?



Which do you prefer?

Image: https://www.mathworks.com/discovery/overfitting.html

Machine Learning
Overfitting

Overfitting

Overfitting means that the model we trained has trained “too well” and has memorized the dataset while loosing its ability to generalize, i.e. perform on unknown/new data.

How to deal with overfitting?

- Split your data
- Regularization
- Use more data, augment your data, i.e. adding noise
- Select different Features
- Cross-validation
- Ensemble methods
- …

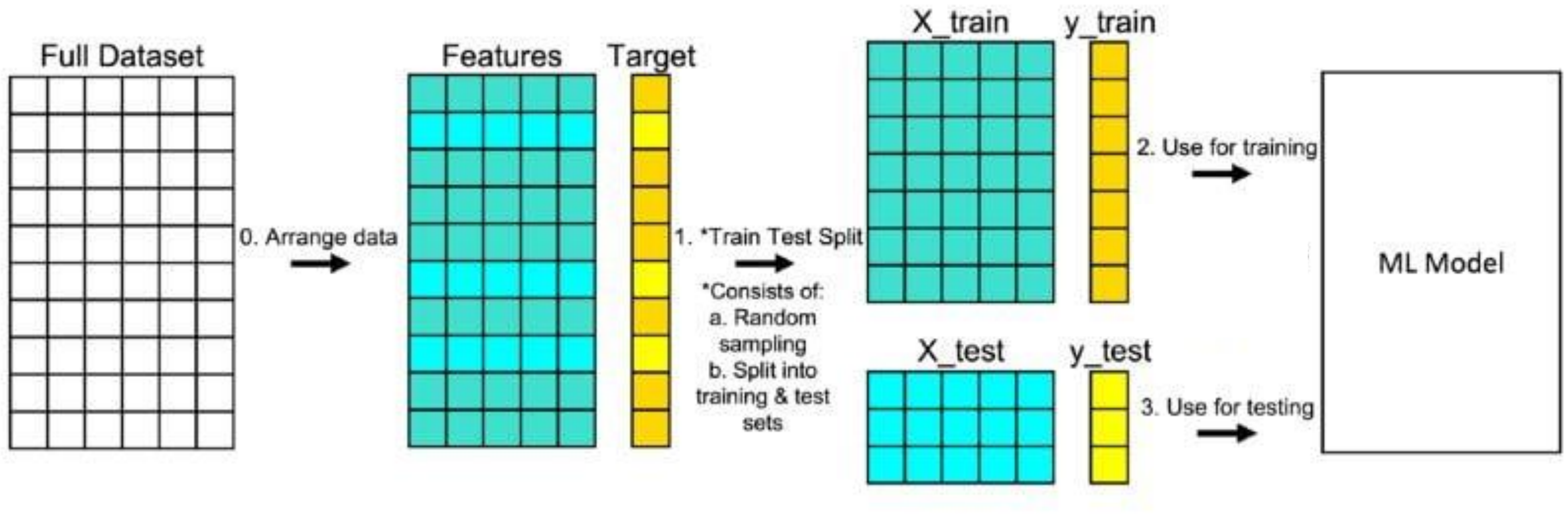
A musical explaination of overfitting by Michael Littman and Charles Isbell: https://www.youtube.com/watch?v=DQWI1kvmwRg

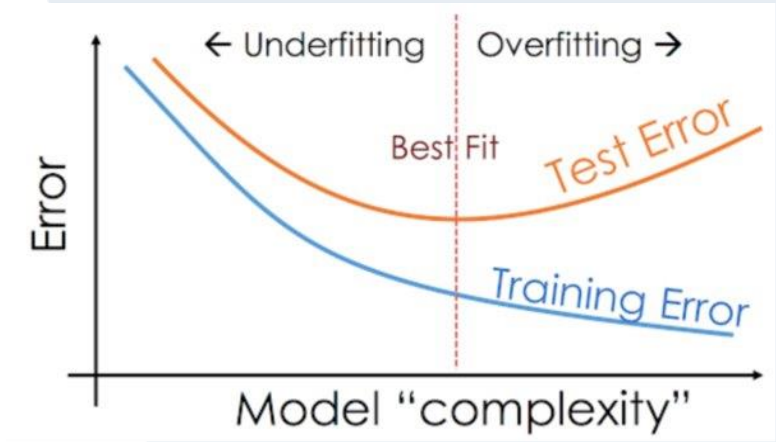
Machine Learning
Evaluation: Train Test Split

Memorization or Generalization?

When learning a model, we want it to perform not only on data we used in training

Idea:
Split your data into Training and Testing. Use Training to train and Testing to evaluate.





04

Artificial **Neural Networks**

Neural Networks
Introduction into Deep Learning

Why Deep Learning?

- Hand-engineered features are time consuming, brittle and not scalable in practice

Idea of Deep Learning

- Can we learn the underlying features directly from data without specifying them?

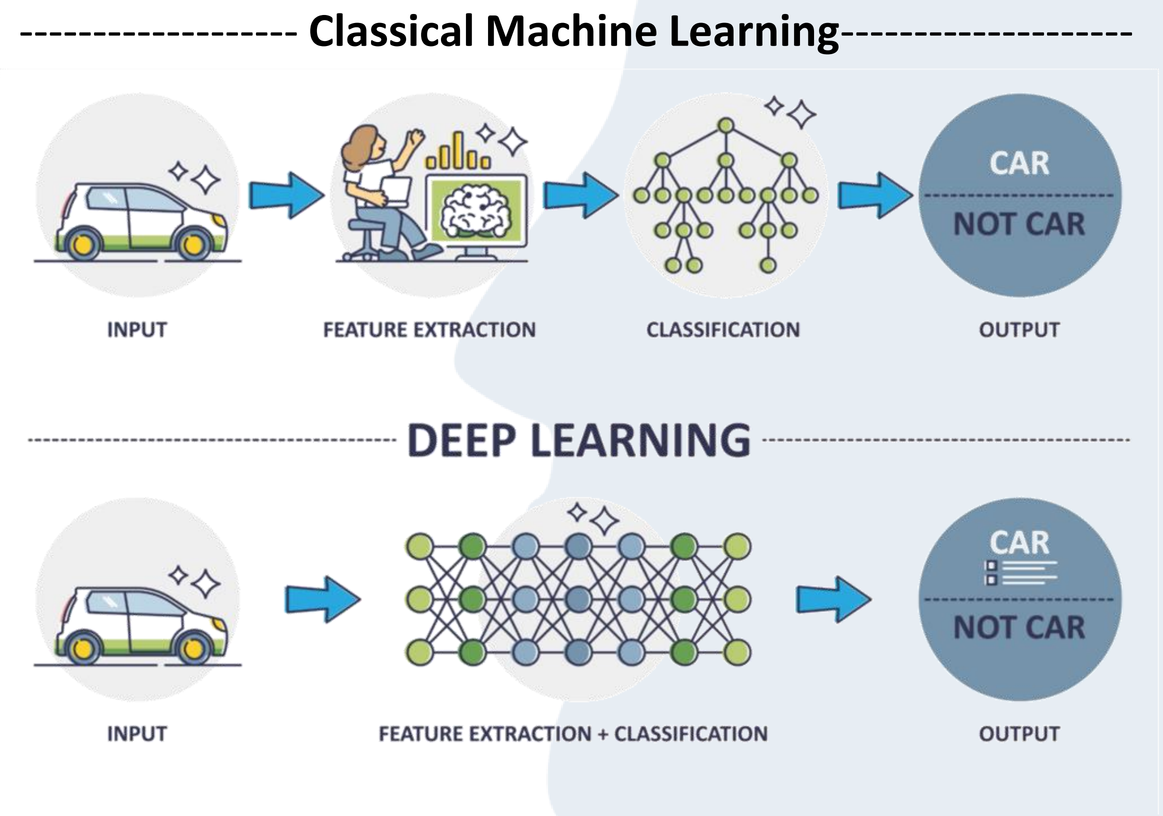


Image: https://www.ait.de/de/deep-learning/

Neural Networks
The Perceptron: An Artificial Neuron

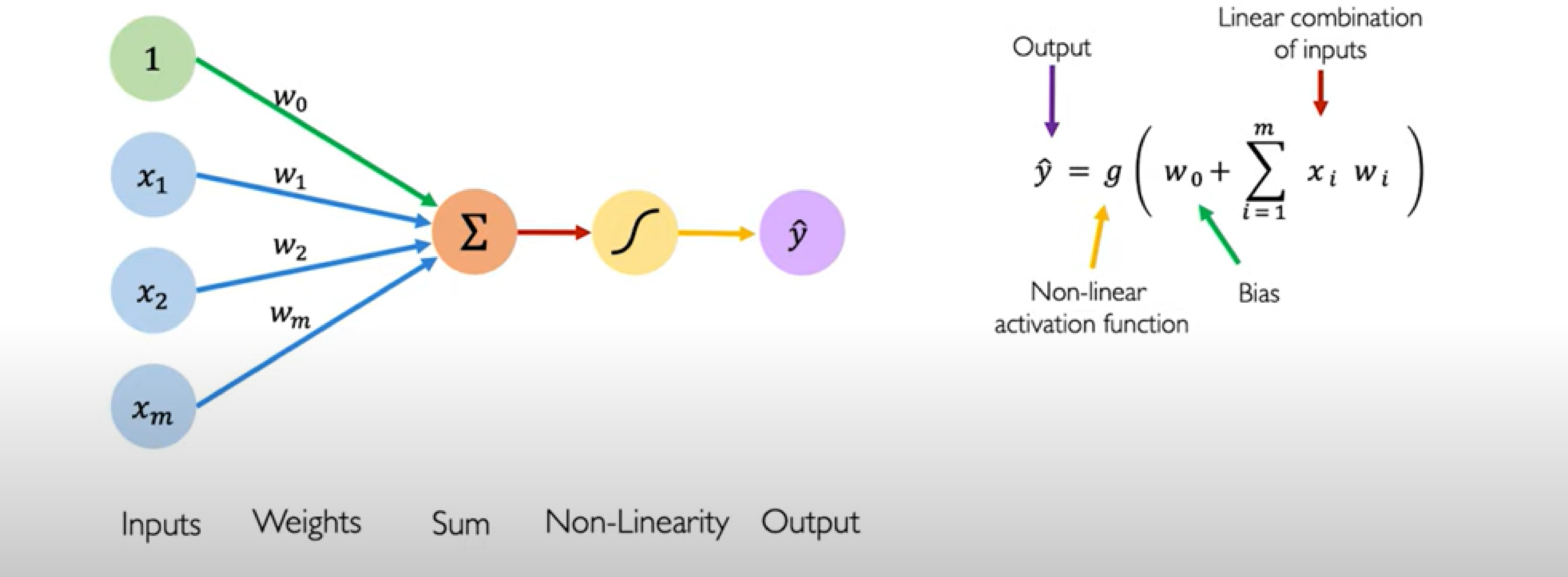
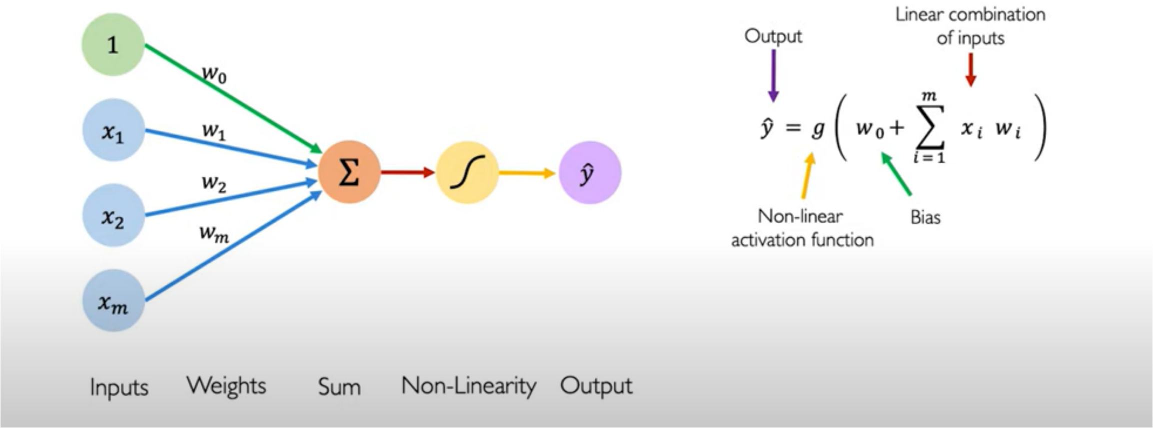


Image: MIT 6.S191 Introduction to Deep Learning

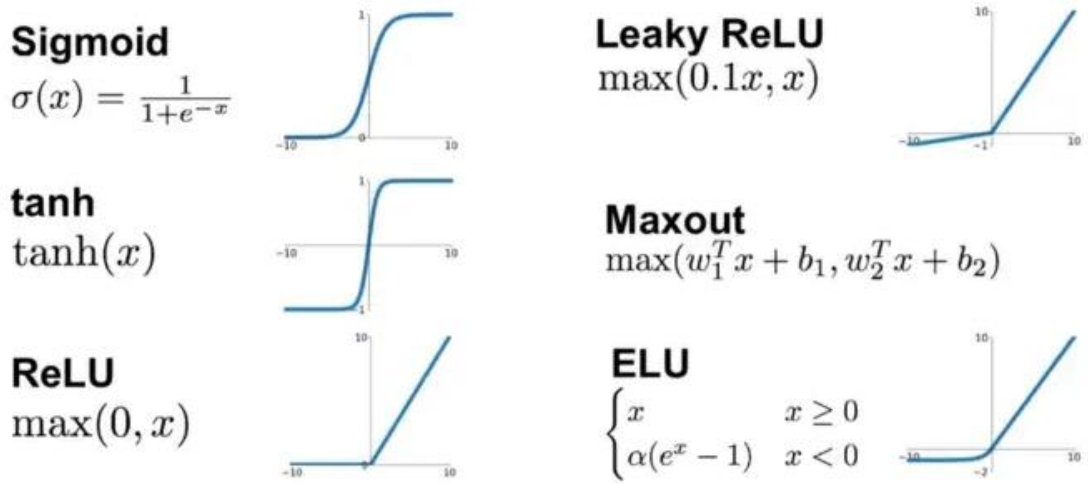
Neural Networks
The Perceptron: An Artificial Neuron



- Neurons correspond to nodes or units
- A link from unit *j* to unit *i* propagates activation *y* from *j* to *i*
- The weight $w_{j,i}$ of the link determines the strength and sign of the connection
- All weights together are called ***W*** or θ and describe our model
- The total input activation is the sum of the input activations
- The output activation is determined by the activiation function *g*

- Image: MIT 6.S191 Introduction to Deep Learning

Neural Networks
Activation Functions

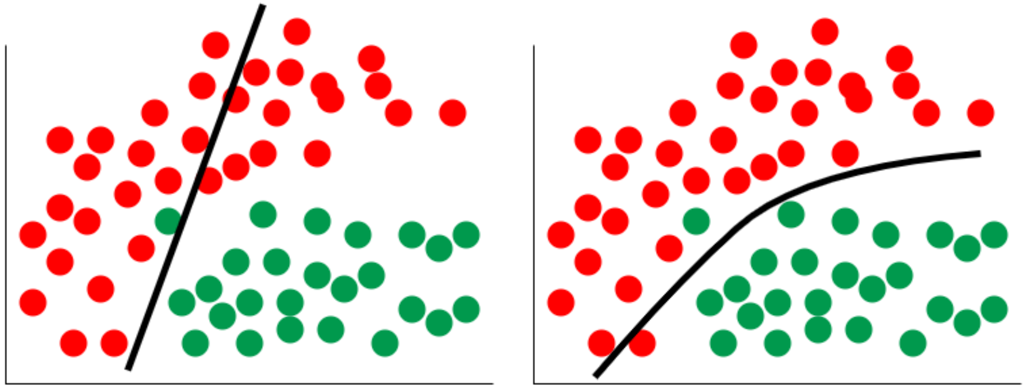


What is an activation function?

- Decides if a neuron should be active
- Nowadays mostly non-linear function

Why do we need an activation function?

- It adds non-linearity to the neural network
- Without it we have a simple linear regression model
- Allows us to use backpropagation (which will be introduced in a later slide)



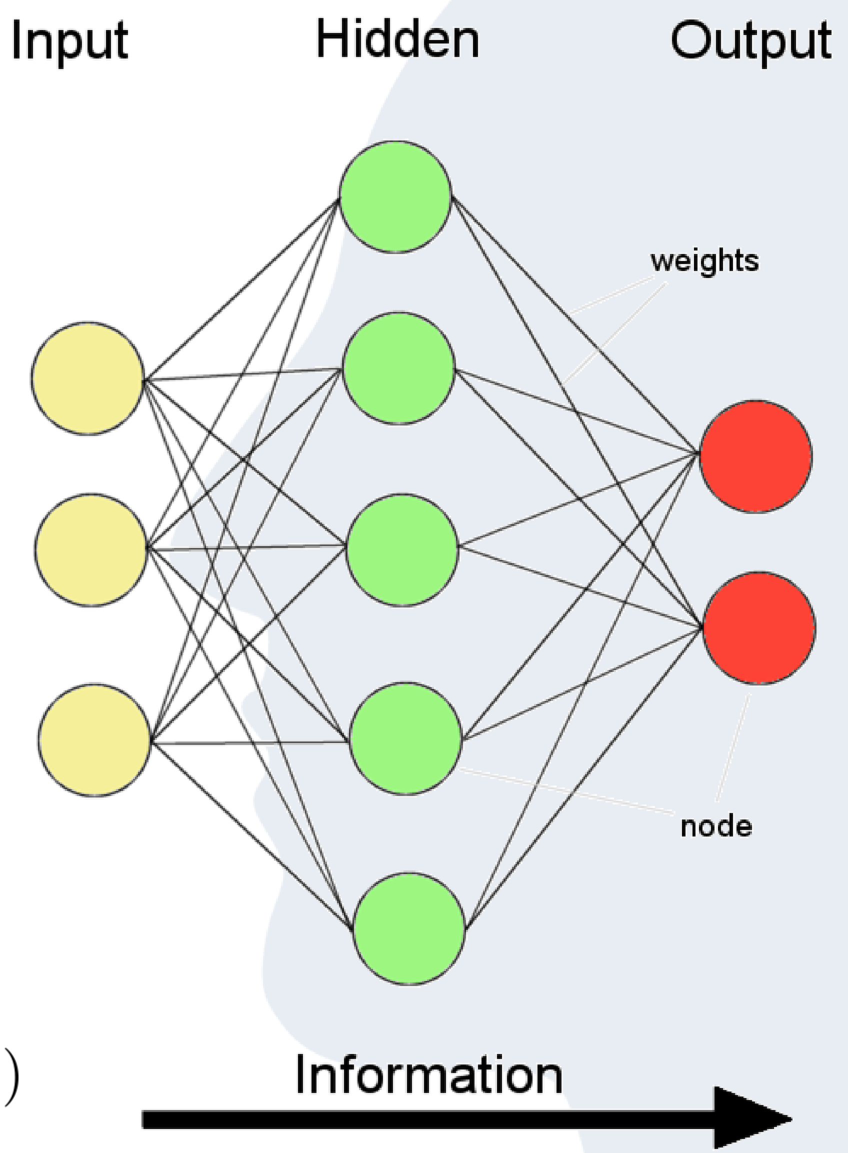
From linear to non-linear patterns

For more information how to choose an activation function: https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/

Neural Network
From a perceptron to a neural network

- Perceptrons may have multiple output nodes
- The output nodes can be combined with other perceptrons
- We can build networks of these nodes, i.e. **Multilayer Perceptrons (MLP)**
- In a Multilayer Perceptron information flow is unidirectional
- Information is distributed and processed in parallel
- We can use the size (i.e. number of layers) to model the expressiveness of an MLP
- Following the definition of a perceptron we know for each hidden node

$$z_{k,i} = \sigma\left(w_{0,i}^{(k)} + \sum_{j=1}^{n_{k-1}} z_{k-1,j} \cdot w_{j,i}^{(k)}\right)$$

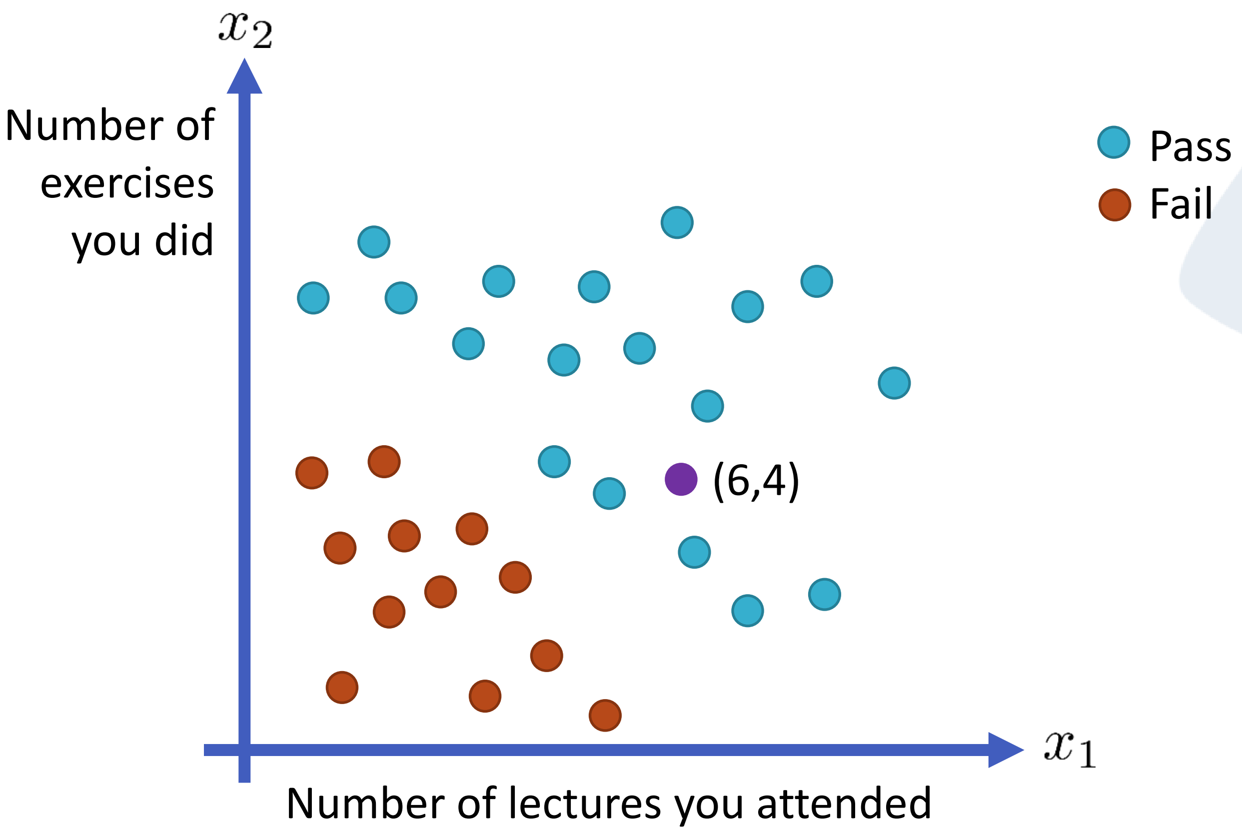


Prediction or Forward Propagation

05

Neural Network
Applying a neural network

Example: Will the purple dot pass the exam?

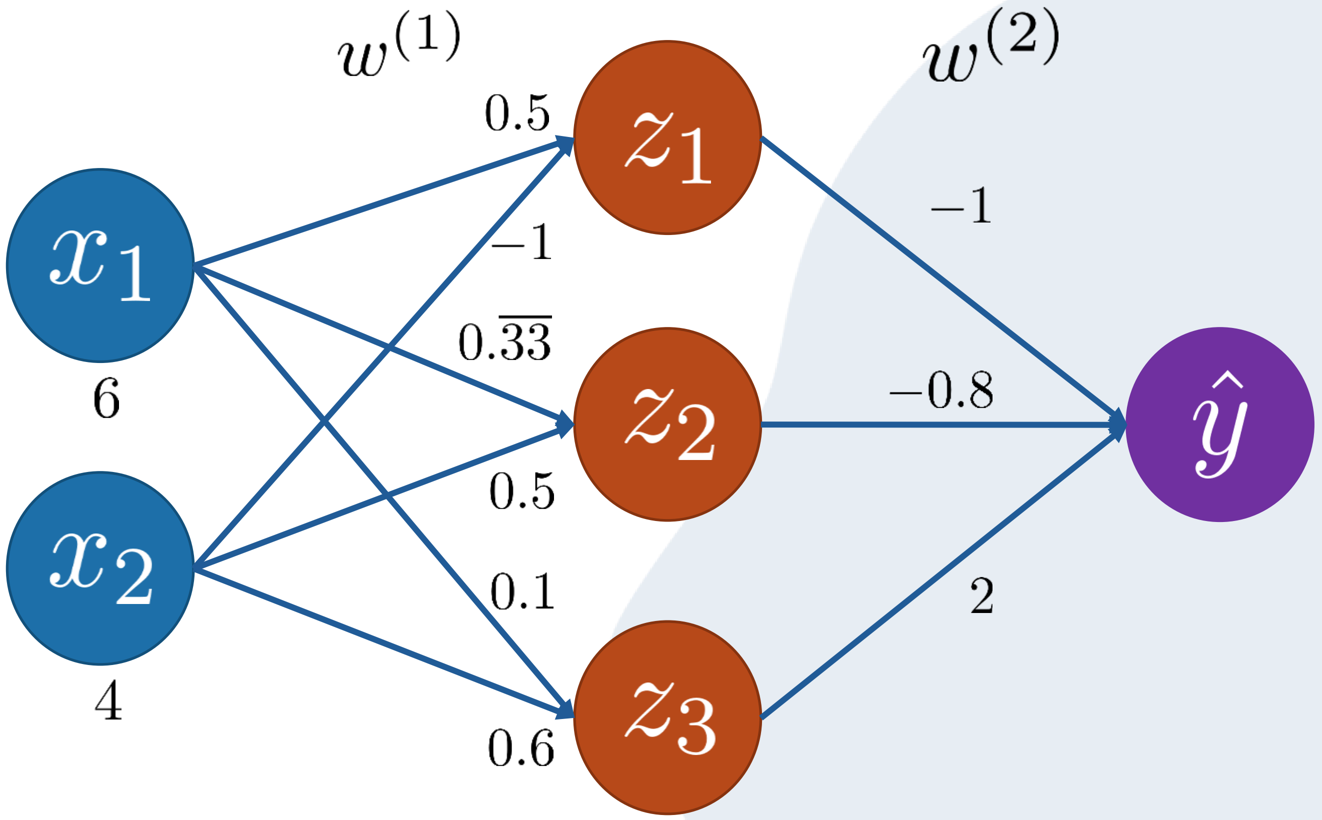


Neural Network
Applying a neural network

Let's assume our input is $x_1 = 6, x_2 = 4$

And the weights are

$$w_{1,1}^{(1)} = 0.5 \quad w_{2,1}^{(1)} = -1 \quad w_{1,1}^{(2)} = -1$$
$$w_{1,2}^{(1)} = 0.\overline{33} \quad w_{2,2}^{(1)} = 0.5 \quad w_{2,1}^{(2)} = -0.8$$
$$w_{1,3}^{(1)} = 0.1 \quad w_{2,3}^{(1)} = 0.6 \quad w_{3,1}^{(2)} = 2$$



Further we assume that the bias $w_{0,i}^{(k)}$ is 0

Then

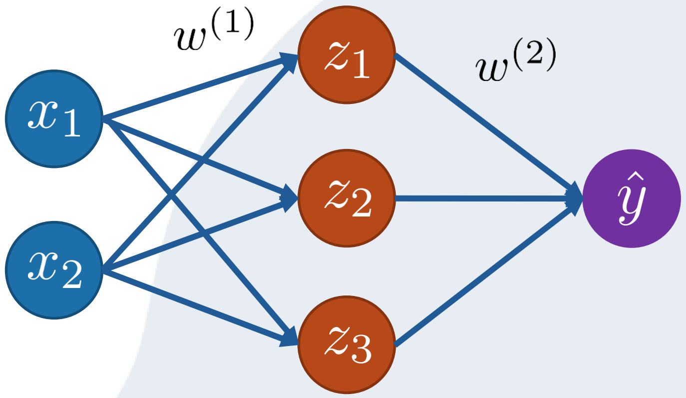
$$z_1 = g(w_{0,1}^{(1)} + \sum_{j=1}^{2} x_j \cdot w_{j,1}^{(1)}) = g(x_1 w_{1,1}^{(1)} + x_2 w_{2,1}^{(1)}) = g(3 + (-4)) = g(-1)$$

$$z_2 = g(2+2) = g(4), z_3 = g(0.6 + 2.4) = g(3)$$

σ

Neural Network

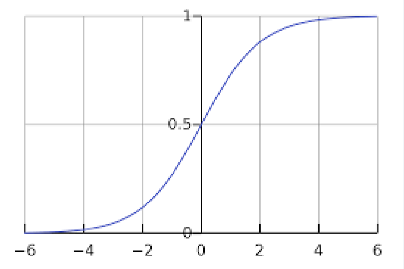
Applying a neural network



$$w^{(1)}_{1,1} = 0.5 \quad w^{(1)}_{2,1} = -1 \quad w^{(2)}_{1,1} = -1$$
$$w^{(1)}_{1,2} = 0.\overline{33} \quad w^{(1)}_{2,2} = 0.5 \quad w^{(2)}_{2,1} = -0.8$$
$$w^{(1)}_{1,3} = 0.1 \quad w^{(1)}_{2,3} = 0.6 \quad w^{(2)}_{3,1} = 2$$

$$z_1 = g(-1), z_2 = g(4), z_3 = g(3)$$

We use the sigmoid activation function: $\sigma(x) = \dfrac{1}{1+e^{-x}}$, Sigmoid Function



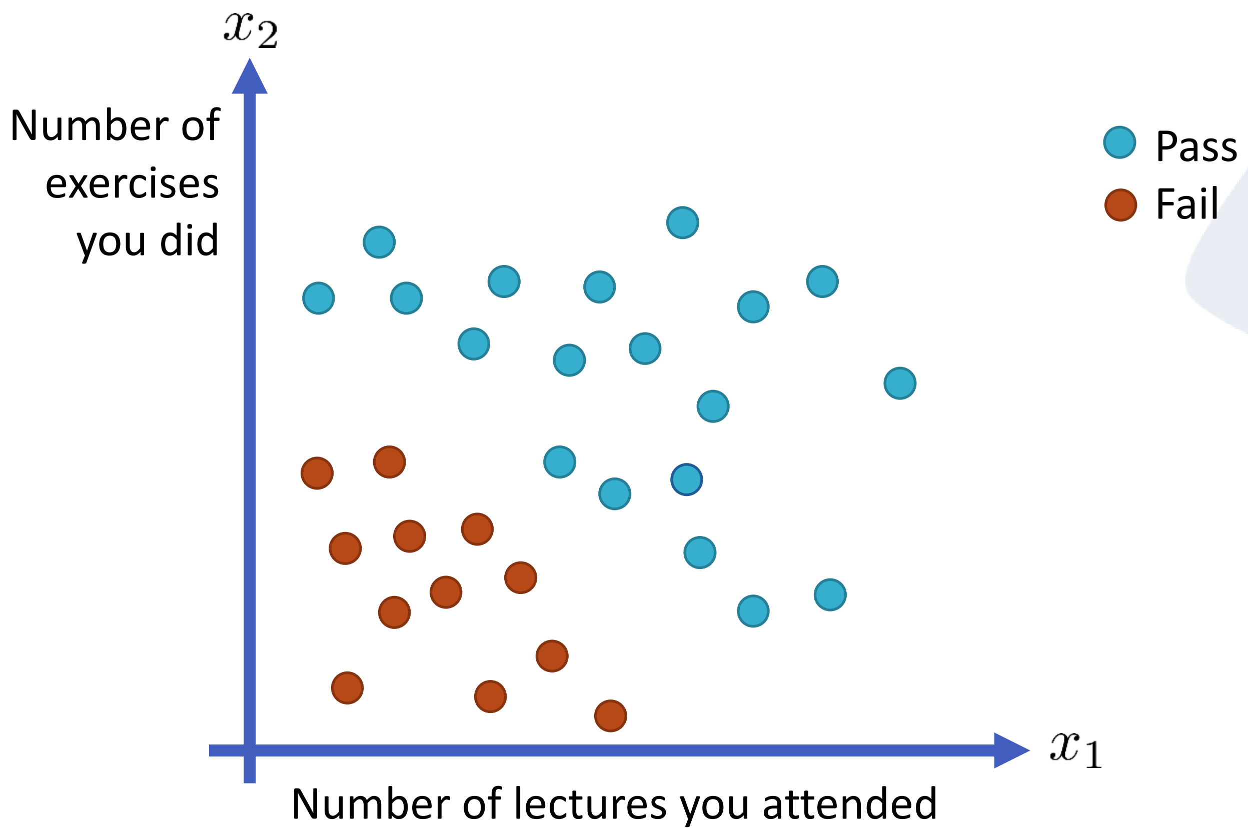
Then $z_1 = 0.26, z_2 = 0.98, z_3 = 0.95$

$$\hat{y} = \sigma(w^{(2)}_{0,1} + \textstyle\sum_{j=1}^{3} z_j \cdot w^{(2)}_{j,i}) = \sigma(w^{(2)}_{0,1} + z_1 w^{(2)}_{1,1} + z_2 w^{(2)}_{2,1} + z_3 w^{(2)}_{3,1})$$

$$\hat{y} = \sigma((-1) \cdot 0.26 + (-0.8) \cdot 0.98 + 2 \cdot 0.95) = \sigma(0.856) = 0.7$$

Neural Network
Applying a neural network

Example: Will the purple dot pass the exam?



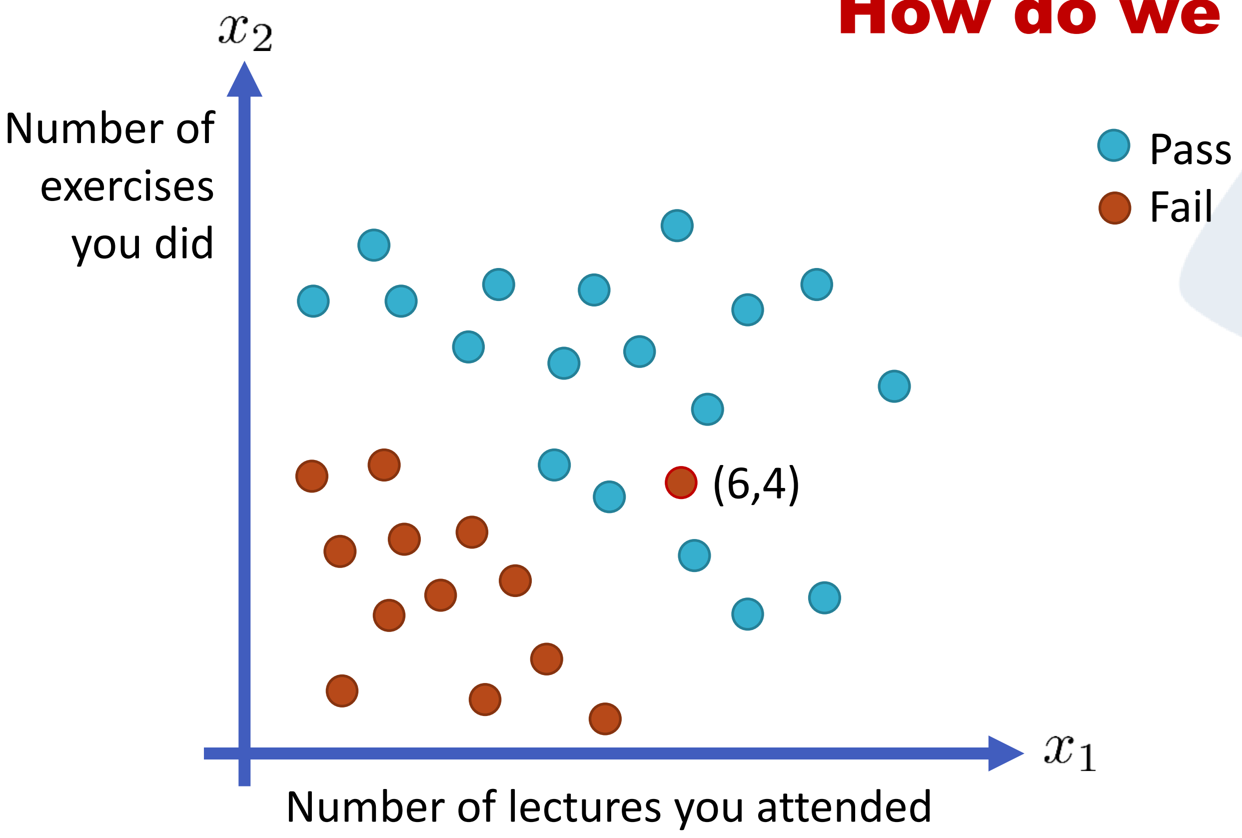
Training a NN or Backpropagation

06

Neural Network
Training a neural network

Lets say we have different weights and instead of 0.7 we get 0.14…

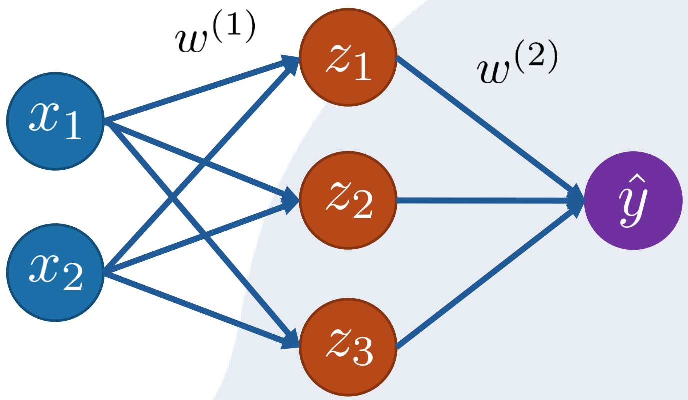
How do we measure quality?

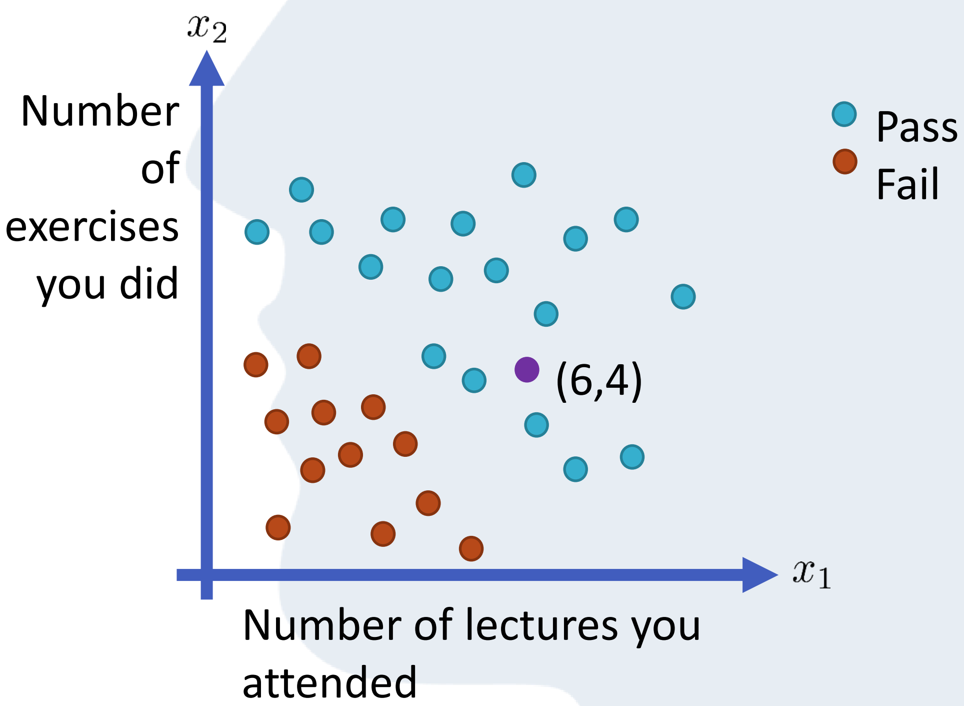


Neural Network
Loss function

Let’s assume we have different weights

→ Now the result is $\hat{y} = 0.14$





Quantifying Loss $\mathcal{L}(f(x^{(i)}; W), y^{(i)})$

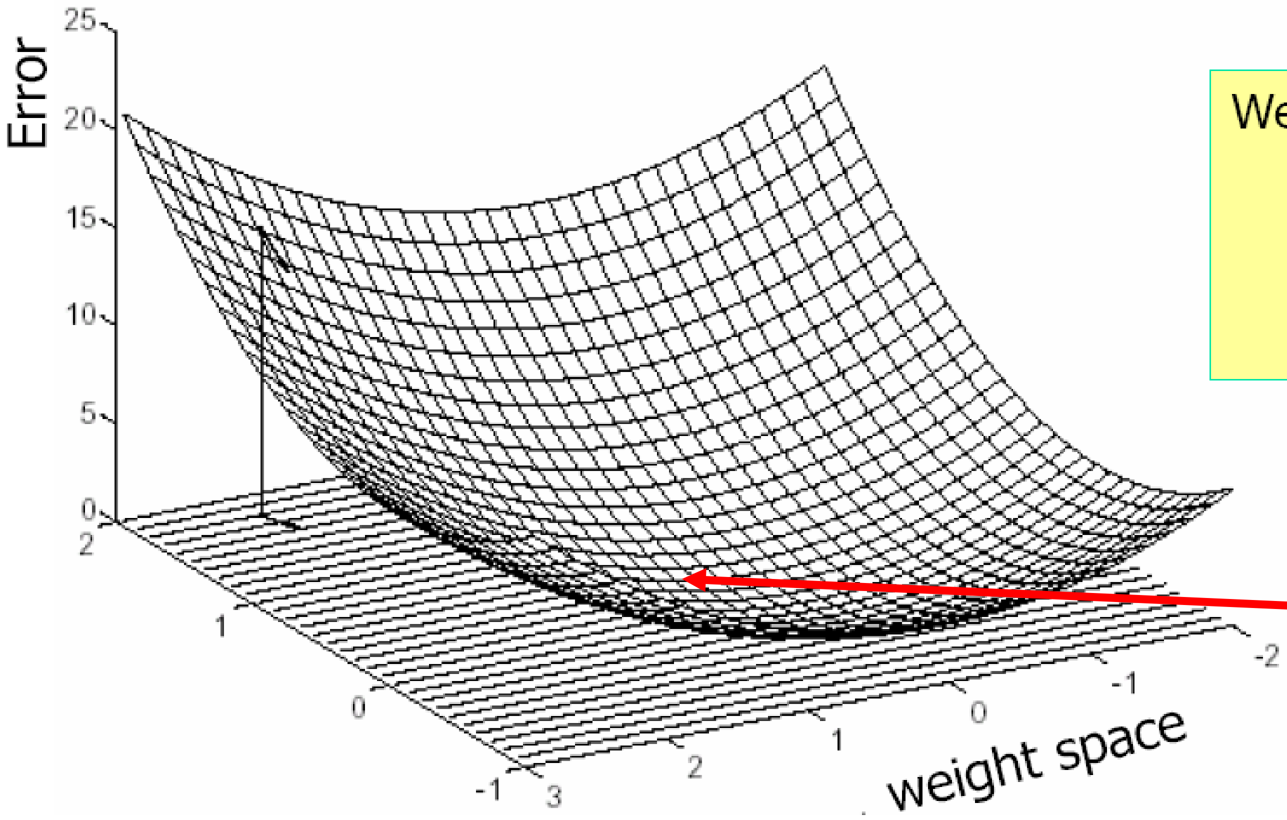
- Describes the cost of incorrect predictions

Empirical Loss $J(W) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x^{(i)}; W), y^{(i)})$

- Measures the total loss over our dataset
- Also known as objective function, cost function or empirical risk

Neural Network
Training our Network

Overall goal: Minimize the loss $W^* = \underset{W}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x^{(i)}; W), y^{(i)})$



Weight space is N-dimensional, where N is the total number of weights in the network

W^* describes the weight setting where the error or loss is minimal for the data set

Neural Network

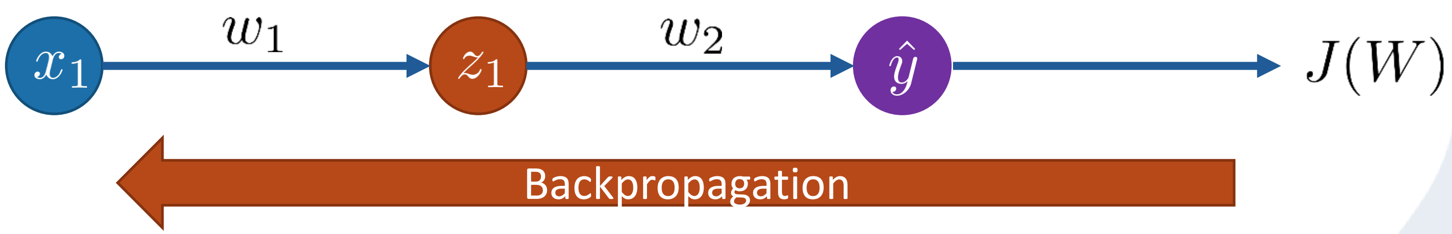
Training our Network: Gradient Descent

Algorithm:

1. Initialize weights randomly
2. Loop until convergence
   - Compute gradient: $\frac{\partial J(W)}{\partial W}$
   - Update weights: $W \leftarrow W - \alpha \frac{\partial J(W)}{\partial W}$
3. Return weights

Neural Network

Backpropagation: How to compute $\frac{\partial J(W)}{\partial W}$



Question: How much do my weights affect the outcome, i.e. the final loss? $\frac{\partial J(W)}{\partial w_i}$

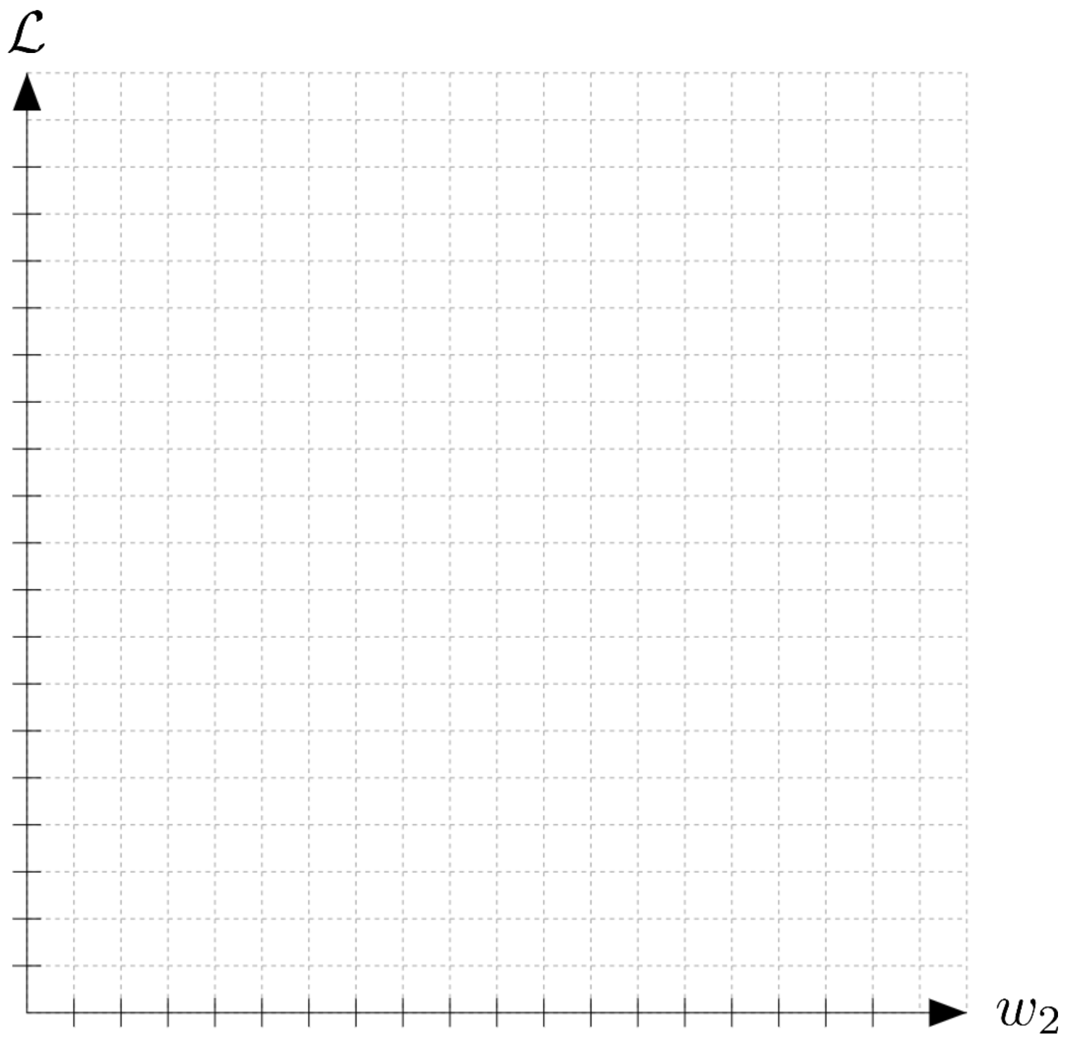
Using the chain rule we can describe the problem as:

$$\frac{\partial J(W)}{\partial w_2} = \frac{\partial J(W)}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial w_2} \qquad \frac{\partial J(W)}{\partial w_1} = \frac{\partial J(W)}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial z_1} \cdot \frac{\partial z_1}{\partial w_1}$$

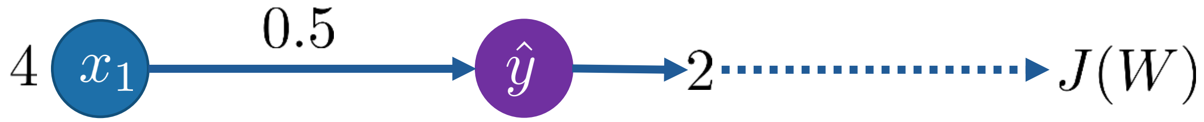
We can repeat this for every weight in the network using the gradients from later layers

- Propagate the error back to all nodes, trough the network

Neural Network
Training our Network: Gradient Descent



Neural Network
Backpropagation: Example



Given $g() = \mathrm{ReLU}(x) = max(0, x), x = 4, y = 1, \hat{y} = 2, \mathcal{L} = (\hat{y} - y)^2, J(W) = \mathcal{L}$

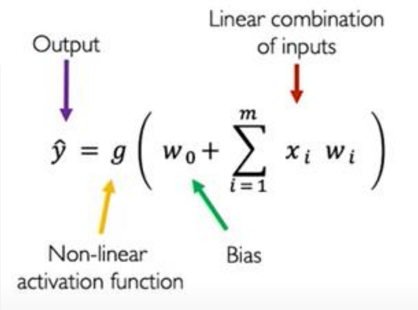
We want $\dfrac{\partial J(W)}{\partial w_1} = \dfrac{\partial J(W)}{\partial \hat{y}} \cdot \dfrac{\partial \hat{y}}{\partial w_1}$

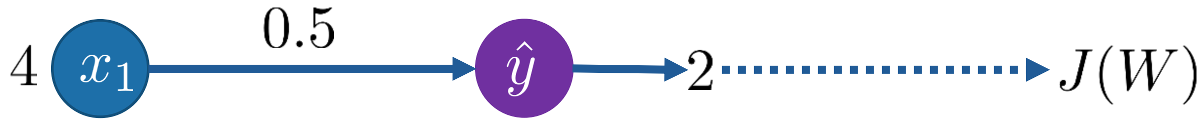
$$J(W) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x^{(i)}; W), y^{(i)})$$

Step 1: $\dfrac{\partial J(W)}{\partial \hat{y}} = 2(\hat{y} - y) = 2(2 - 1) = 2$

A good video explaining this: https://www.youtube.com/watch?v=khUVIZ3MON8

Neural Network
Backpropagation: Example





Given $g() = \mathrm{ReLU}(x) = max(0, x), x = 4, y = 1, \hat{y} = 2, \mathcal{L} = (\hat{y} - y)^2, J(W) = \mathcal{L}$

We want $\dfrac{\partial J(W)}{\partial w_1} = \dfrac{\partial J(W)}{\partial \hat{y}} \cdot \dfrac{\partial \hat{y}}{\partial w_1}$

Step 1: $\dfrac{\partial J(W)}{\partial \hat{y}} = 2(\hat{y} - y) = 2(2 - 1) = 2$

Step 2: $\dfrac{\partial \hat{y}}{\partial w_1} = \mathrm{ReLU}'(w_0 + \sum_{i=1}^{n} w_i x_i) \cdot x_1 = 1 \cdot 4 = 4$

Step 3: $\dfrac{\partial J(W)}{\partial w_1} = \dfrac{\partial J(W)}{\partial \hat{y}} \cdot \dfrac{\partial \hat{y}}{\partial w_1} = 2 \cdot 4 = 8$

$$\hat{y} = \mathrm{ReLU}(w_1 x_1)$$

$$f = g(h(x))$$
$$f' = g'(h(x)) \cdot h'(x)$$

$$\mathrm{ReLU}'(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$

A good video explaining this: https://www.youtube.com/watch?v=khUVIZ3MON8

Neural Network

Update the Weights: $W \leftarrow W - \alpha \frac{\partial J(W)}{\partial W}$

Step 3: $\frac{\partial J(W)}{\partial w_1} = 8$

Step 4: $W_{new} = W_{current} - \alpha \frac{\partial J(W)}{\partial W}$

$$W_{new} = 0.5 - 0.05 \cdot 8$$

$$W_{new} = 0.1$$

$$\alpha = 0.05$$

Alternative Step 4

$$W_{new} = 0.5 - 0.5 \cdot 8$$

$$W_{new} = -3.5$$

$$\alpha = 0.5$$

Neural Networks

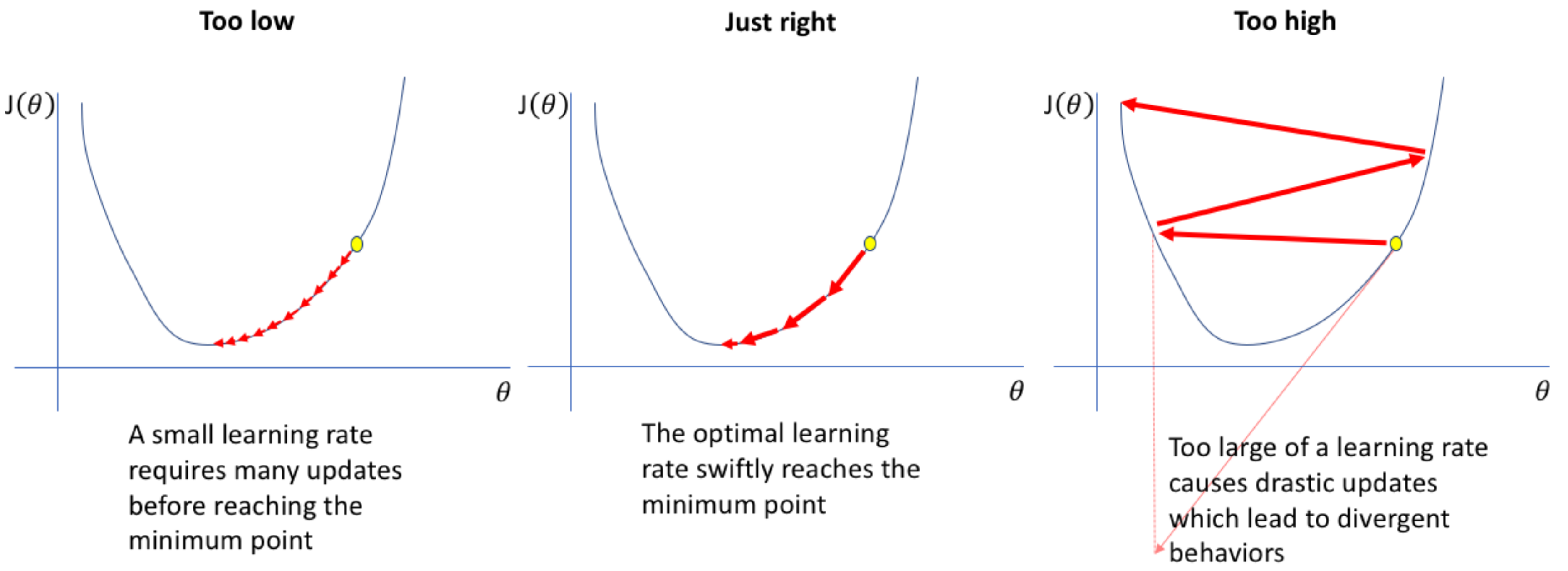
Update the Weights: $W \leftarrow W - \alpha \frac{\partial J(W)}{\partial W}$

Loss Functions can be difficult to optimize.

- Hard to find a global minimum

Idea: Change weights into the direction of the steepest descent of the error function giving a step size.

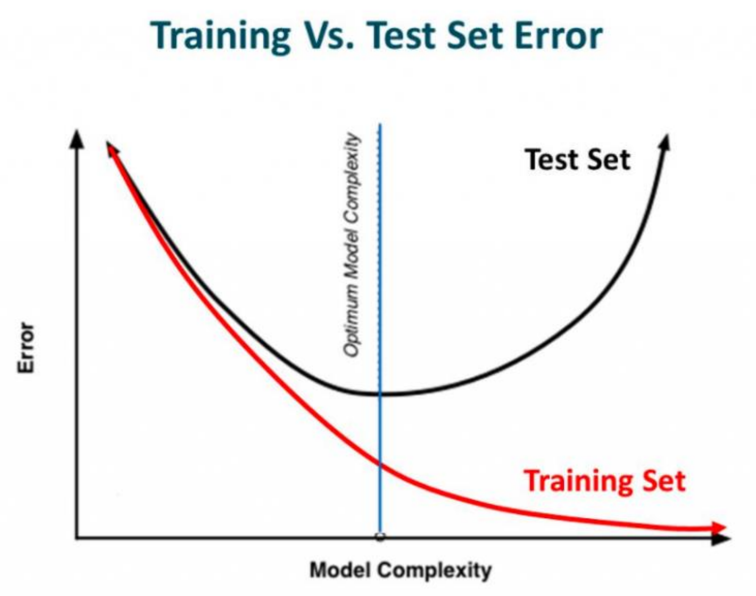
- This step size is called **learning rate** (α)
- Finding a good learning rate is difficult



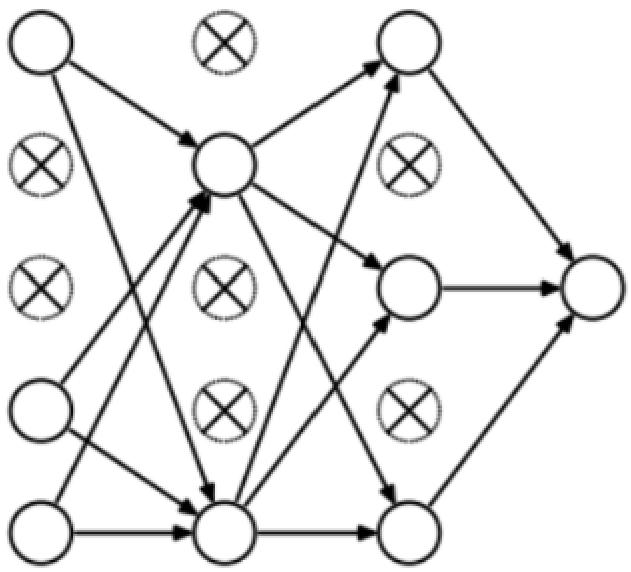
Neural Network
Overfitting

Regularization

Regularization is a set of techniques that can prevent overfitting in neural networks and thus improve the accuracy of a Deep Learning model when facing completely new data from the problem domain



Early Stopping



Dropout

What's next?

07

Neural Networks
Further Architectures

MLP are only the first step in the field of neural network architectures

- How many layers should I use?
- What if my input is very complex, e.g. images?
- How many features should I use?
- Can I use images, and text at the same time as inputs?
- How can I do sequences of data, e.g. sentences, time-series,…
- …

Neural Networks
Convolutional Neural Network (CNN)

- CNNs uses **convolutional layers** to extract features from the input
- Features in the first layers refer to edges, borders, shapes,…
- On higher levels it detect patterns and at some point specific objects
- It uses pooling layers to decrease the computational requirements and extract more dominant features
- Compared to MLPs
    - They have feyer connections, i.e. weights
    - Are easier to train
    - Can have a lot of layers without
- Very popular in fields like Computer Vision and NLP

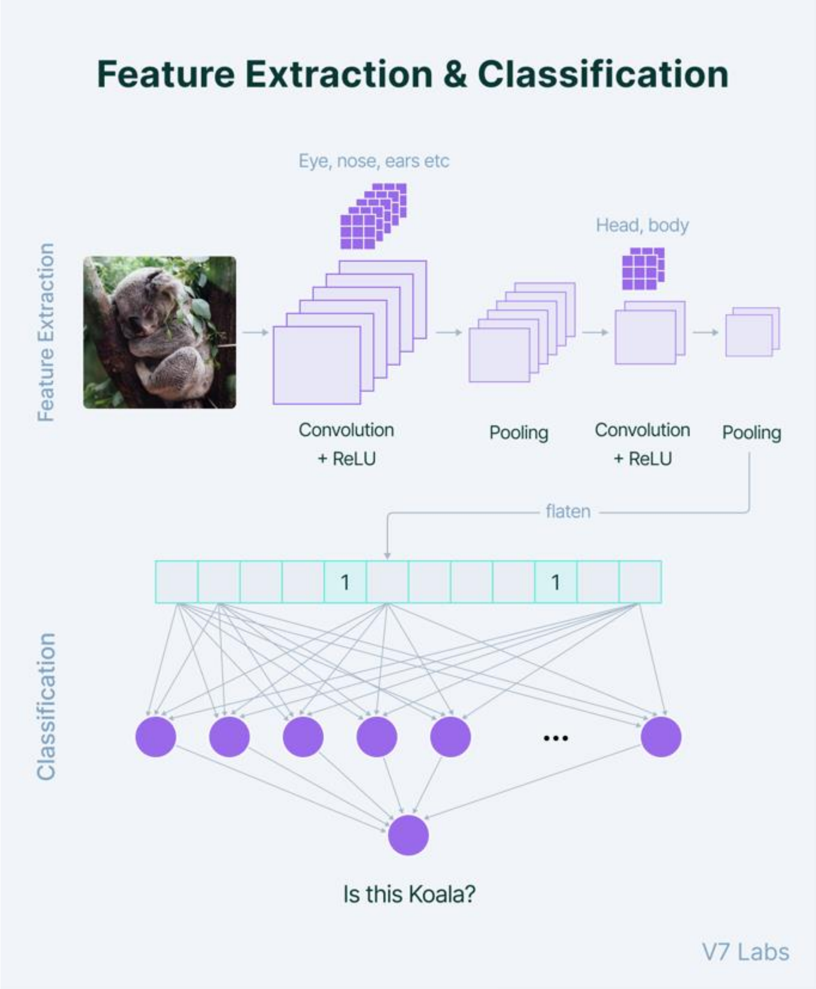


Image: https://www.v7labs.com/blog/neural-network-architectures-guide

Neural Networks
Performance Processing

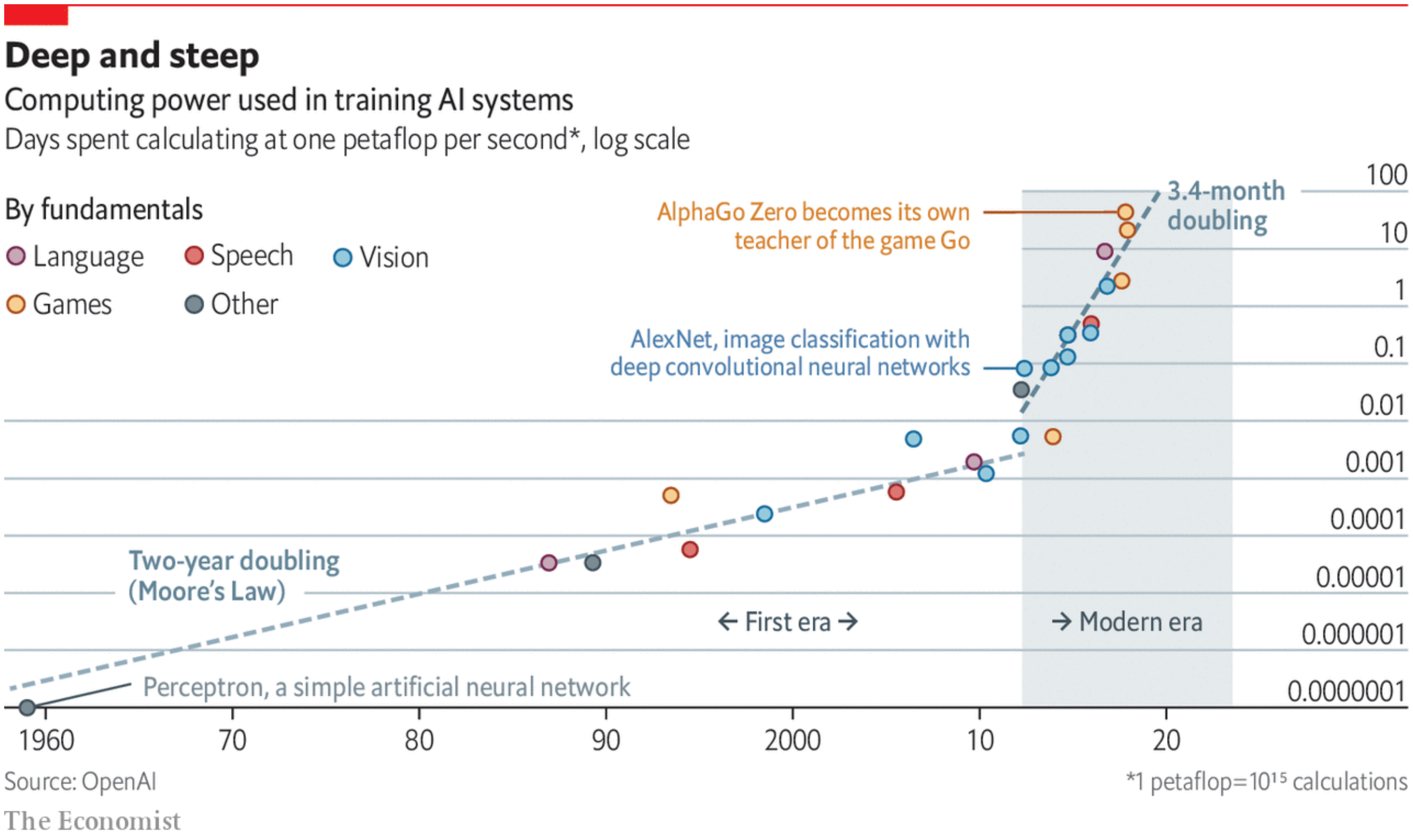


Image: https://www.economist.com/technology-quarterly/2020/06/11/the-cost-of-training-machines-is-becoming-a-problem

Follow up

Additional Sources

MIT S6.191 Introduction into Deep Learning
https://www.youtube.com/playlist?list=PLtBw6njQRU-rwp5__7C0oIVt26ZgjG9NI

Stanford CS229: Machine Learning
https://www.youtube.com/playlist?list=PLoROMvodv4rMiGQp3WXShtMGgzqpfVfbU

Ted Talk Lecture Friends with ML
https://www.youtube.com/playlist?list=PLRKtJ4IpxJpDxl0NTvNYQWKCYzHNuy2xG

Pattern Recognition and Machine Learning (Bishop, 2006)
https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf

Introduction into Pytorch for Deep Learning
https://www.learnpytorch.io/

Lectures in Darmstadt

- Deep Learning: Architectures & Methods (DLMA)
- Data Mining and Machine Learning (DMML)
- Satistical Machine Learning (SML)
- Probabilistic Graphical Models (PGM)
- Computer Vision (CV)
- Reinforcement Learning (RL)
- Continual Machine Learning (ContML)
- Deep Learning for NLP (DL4NLP)
- Deep Learning for Medical Imaging (DLMB)
- Robot Learning
- Deep Generative Models (TGM)

...

Summary

- What is Learning
- Types of Learning
- What is Machine Learning
- How to deal with overfitting
- Machine Learning on the example of neural networks
- Training a neural network

You should be able to:

- describe learning agents
- distinguish between different types of learning
- Give multiple solution how to solve overfitting
- Describe the Perceptron architecture
- Calculate a Forward Propagation and Backpropagation

Next Week: Reinforcement Learning