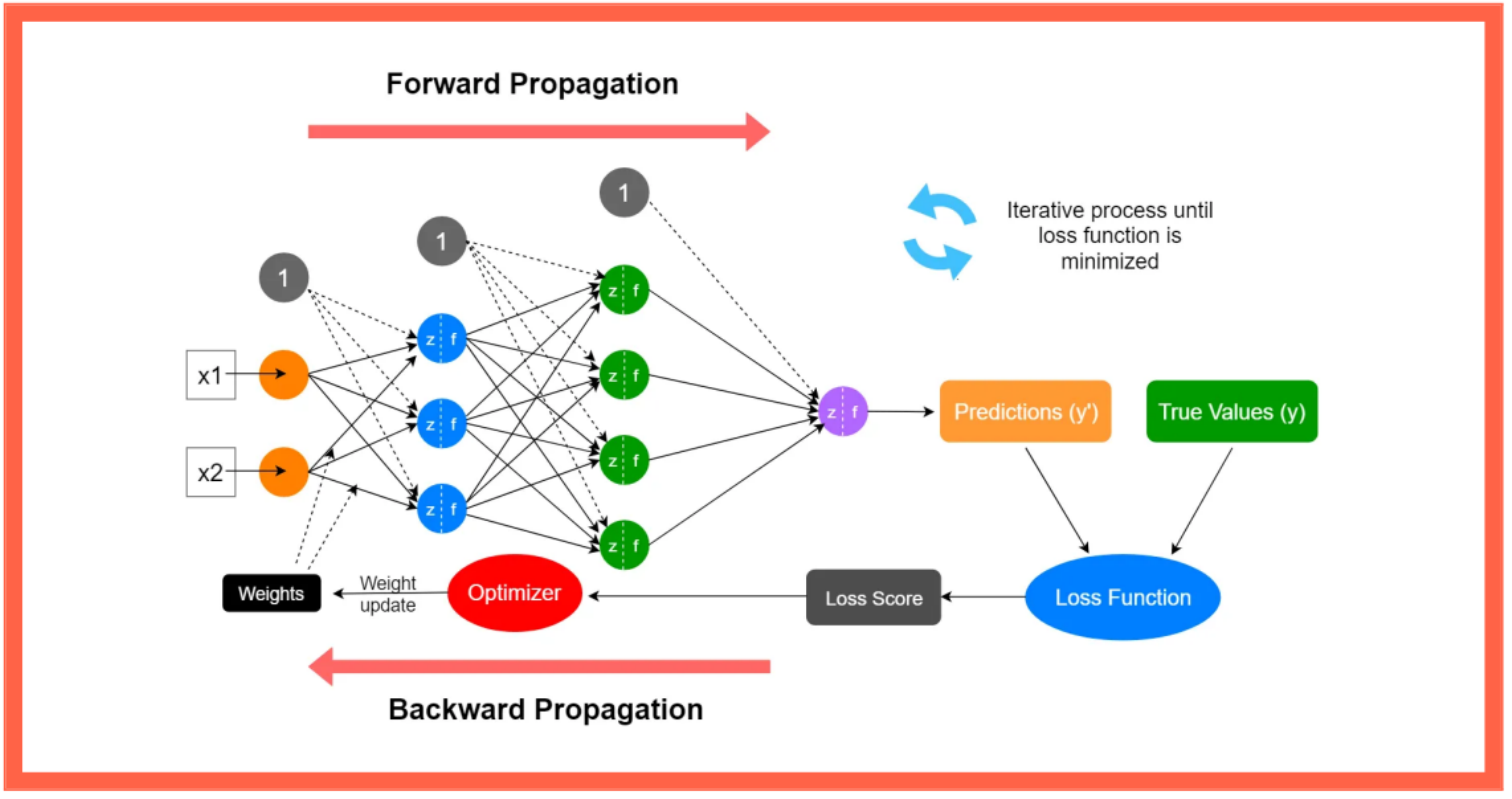
The mathematics of AI

Or: Learning = forward, loss, backward

~~Accept~~Change what you cannot ~~change~~accept



What is machine learning?

- Task Identify handwritten digits
- We can see this as a function in the following way:
  - Convert the pictures into grayscale values, e.g. 28×28 grid of numbers
  - Flatten the result into a vector, e.g. $28 \times 28 \mapsto$ a vector with $28^2 = 784$ entries
  - The output is a vector with 10 entries
- We thus have a function $\mathbb{R}^{784} \to \mathbb{R}^{10}$

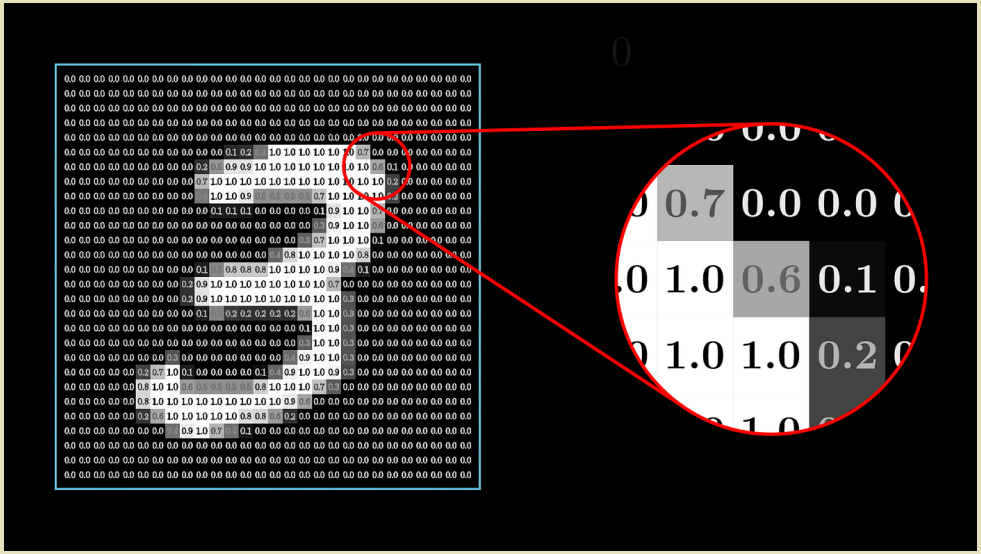
What is machine l

- Task Identify
- We can see thi
  - Convert th
  - Flatten the
  - The outpu
- We thus have a

28 grid of numbers

ith $28^2 = 784$ entries

Input example



Output example

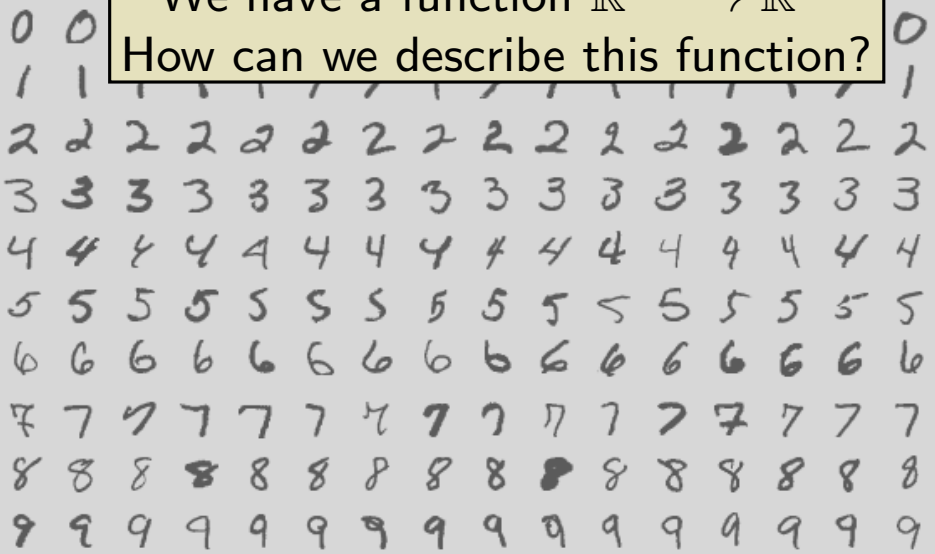
0.1	0
0.0	1
0.0	2
0.6	3
0.0	4
0.0	5
0.1	6
0.0	7
0.0	8
0.2	9

$\Rightarrow$ with 60% certainty this is a 3

What is machine learn

Task – rephrased

We have a function $\mathbb{R}^{784} \to \mathbb{R}^{10}$
How can we describe this function?

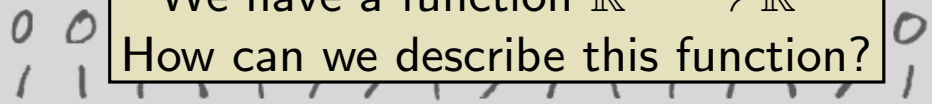


- **Task** Identify handwritten digits
- We can see this as a **function** in the following way:
  - Convert the pictures into grayscale values, e.g. 28×28 grid of numbers
  - Flatten the result into a vector, e.g. $28 \times 28 \mapsto$ a vector with $28^2 = 784$ entries
  - The output is a vector with 10 entries
- We thus have a function $\mathbb{R}^{784} \to \mathbb{R}^{10}$

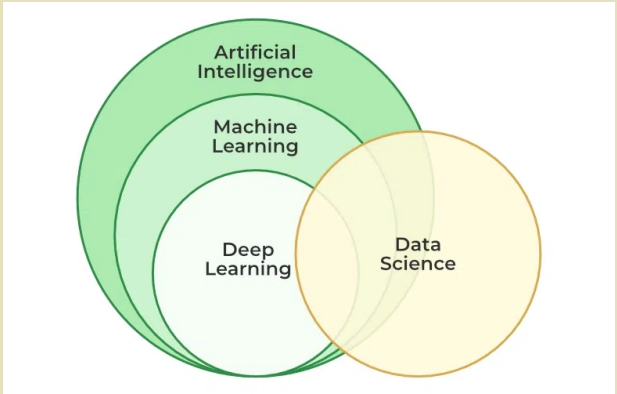
What is machine learn

Task – rephrased

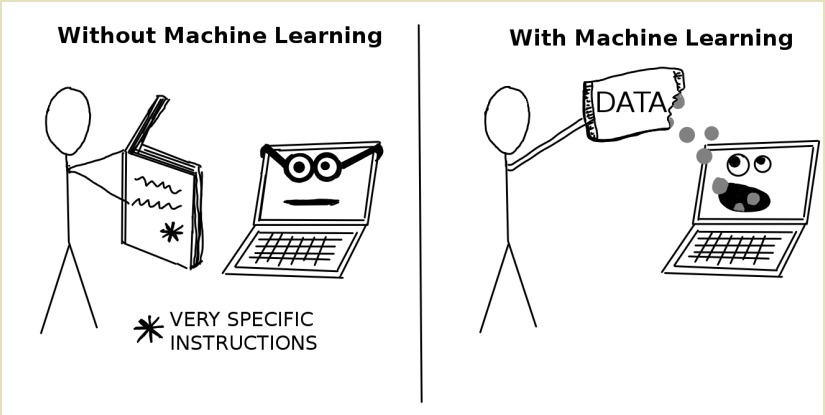
We have a function $\mathbb{R}^{784} \to \mathbb{R}^{10}$
How can we describe this function?



Machine/deep learning (today's topic)



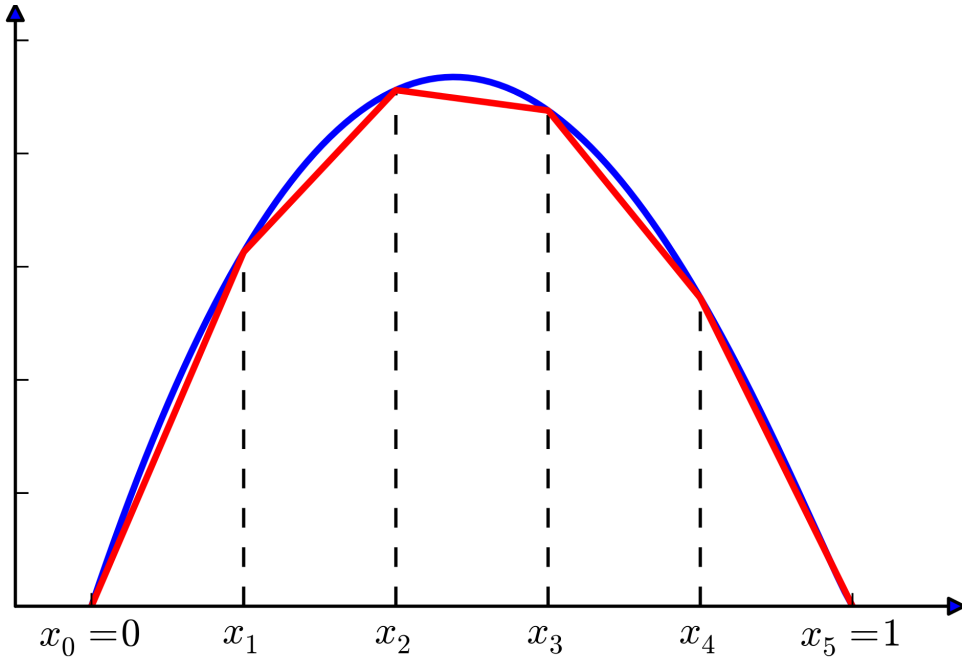
tries to answer questions of this type
letting a computer detect patterns in data



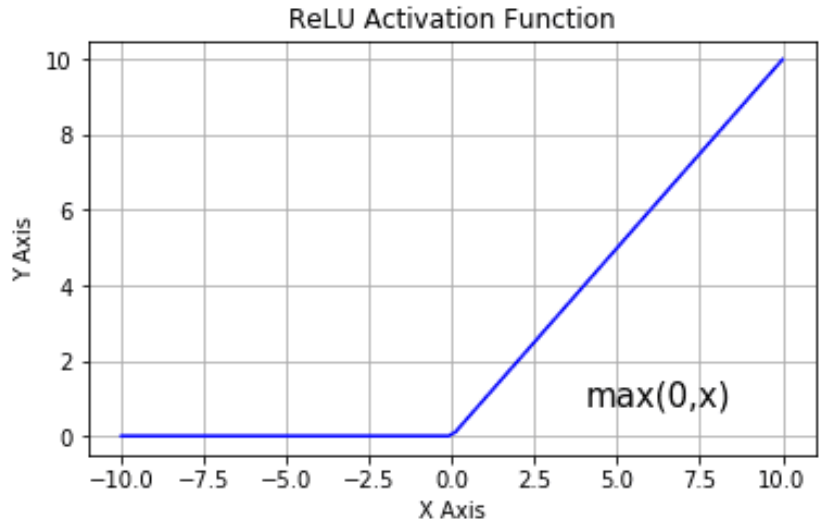
Crucial In ML the computer performs tasks without explicit instructions

- T
- W
  - umbers
  - entries
- W

What is machine learning?

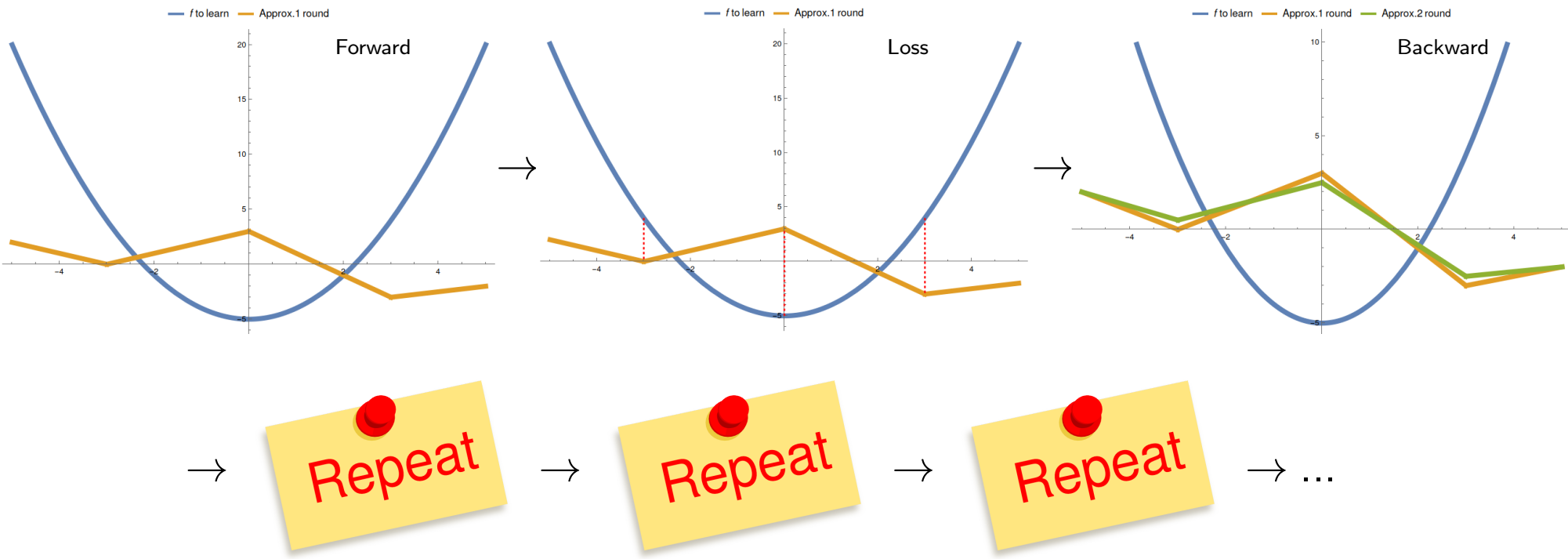


,



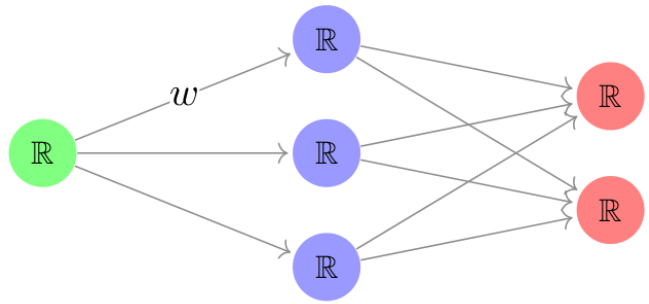
- **Idea** Approximate the unknown function $\mathbb{R}^{784} \to \mathbb{R}^{10}$
- **Neural network** = a piecewise linear approximation (matrices + PL maps)
- The **matrices** = a bunch of numbers (weights) and offsets (biases)
- The **PL maps** = usually ReLU

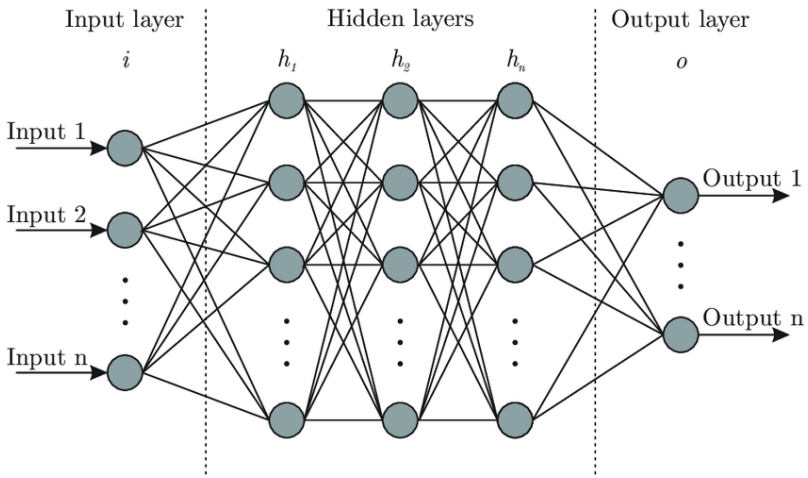
What is machine learning?



- Machine learning mantra
  - Forward = calculate an approximation (start with random inputs)
  - Loss = compare to real data
  - Backward = adjust the approximation

What is a neural network (nn)?

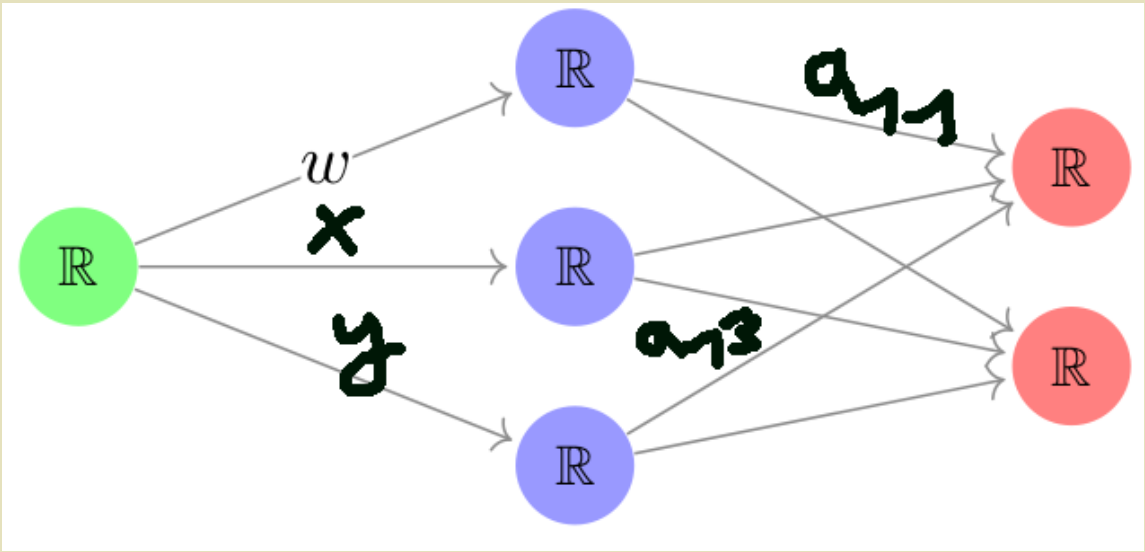




- NN = a directed graph as above
- The task of a nn is to approximate an unknown function
- It consist of neurons = entries of vectors, and weights = entries of matrices

What is a neural network (nn)?

Example



Here we have two matrices, a 3-by-1 and a 2-by-3 matrix

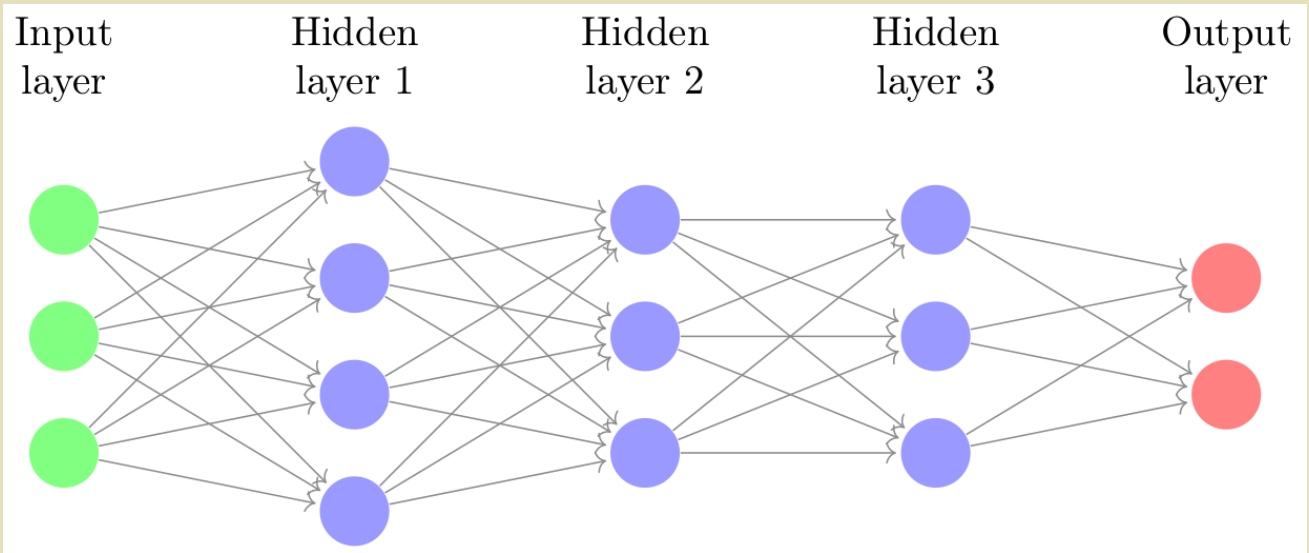
$$\begin{pmatrix} w \\ x \\ y \end{pmatrix} \text{ and } \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

plus two bias terms as in $y = matrix \cdot x + bias$

- NN = a
- The task of a nn is to approximate an unknown function
- It consist of neurons = entries of vectors, and weights = entries of matrices

What is a neural network (nn)?

Example



- Here we have four matrices (plus four biases), whose composition gives a map

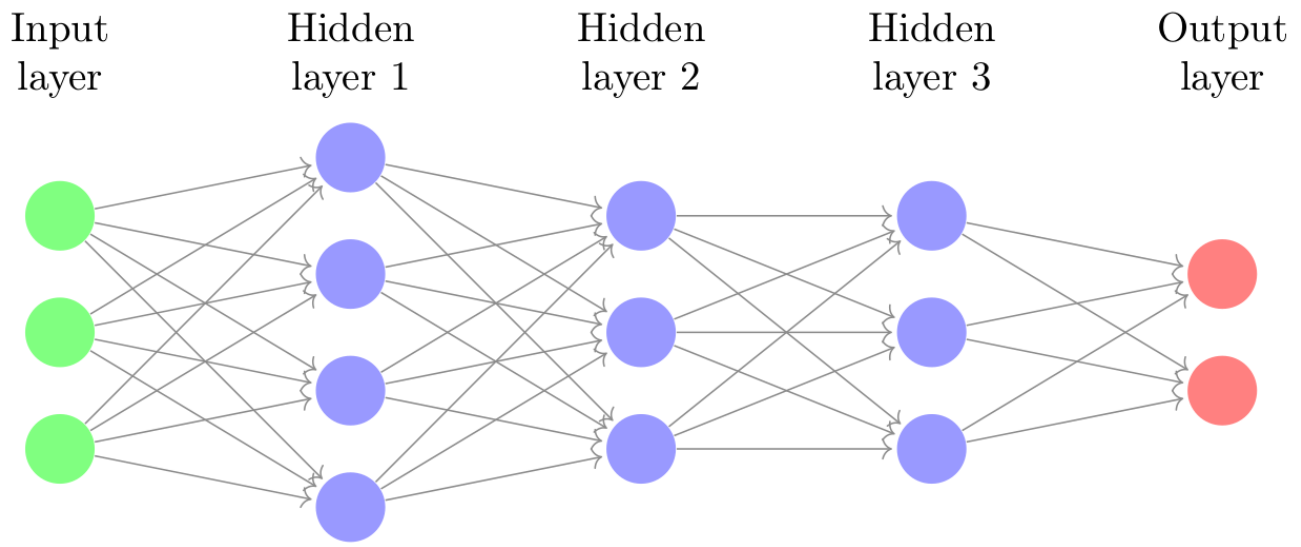
$$\mathbb{R}^3 \to \mathbb{R}^4 \to \mathbb{R}^3 \to \mathbb{R}^3 \to \mathbb{R}^2$$

- The task of a nn is to approximate an unknown function
- It consist of neurons = entries of vectors, and weights = entries of matrices

What is a neural network (nn)?

Actually...

we need nonlinear maps as well, say ReLU applied componentwise



Here we have four matrices, whose composition gives a map

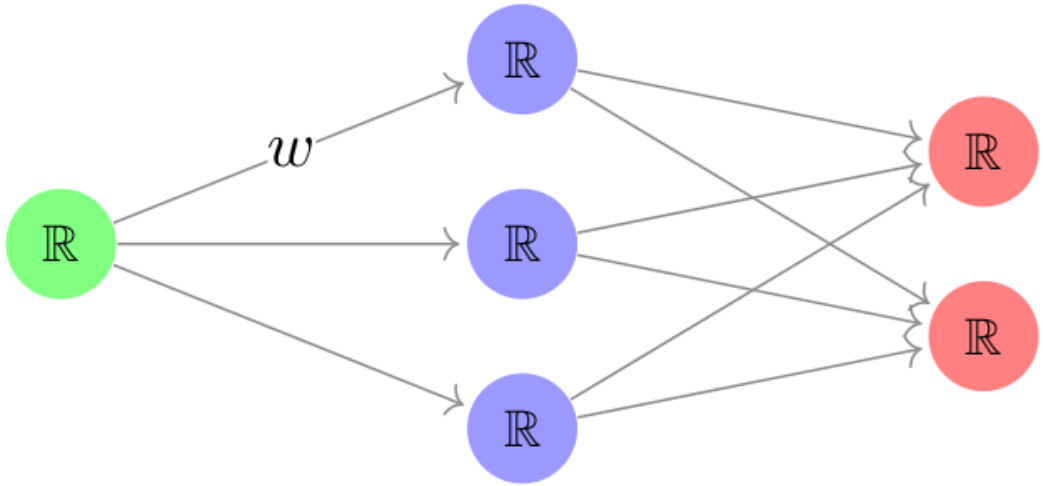
$$\mathbb{R}^3 \xrightarrow{\text{ReLU}\circ\text{matrix}} \mathbb{R}^4 \xrightarrow{\text{ReLU}\circ\text{matrix}} \mathbb{R}^3 \xrightarrow{\text{ReLU}\circ\text{matrix}} \mathbb{R}^3 \xrightarrow{\text{ReLU}\circ\text{matrix}} \mathbb{R}^2$$

- NN = a
- The task
- It consist of neurons = entries of vectors, and weights = entries of matrices

But ignore that for now

ReLU doesn't learn anything, its just brings in nonlinearity

What is a neural network (nn)?



$$\left(\begin{array}{c|c} a^1_{11} & b^1_1 \\ a^1_{12} & b^1_2 \\ a^1_{13} & b^1_3 \end{array}\right), \quad \left(\begin{array}{ccc|c} a^2_{11} & a^2_{12} & a^2_{13} & b^2_1 \\ a^2_{21} & a^2_{22} & a^2_{23} & b^2_2 \end{array}\right)$$

- The a^k_{ij} and b^k_i are the parameters of our nn
- k = number of the layer
- Deep = many layers = better approximation

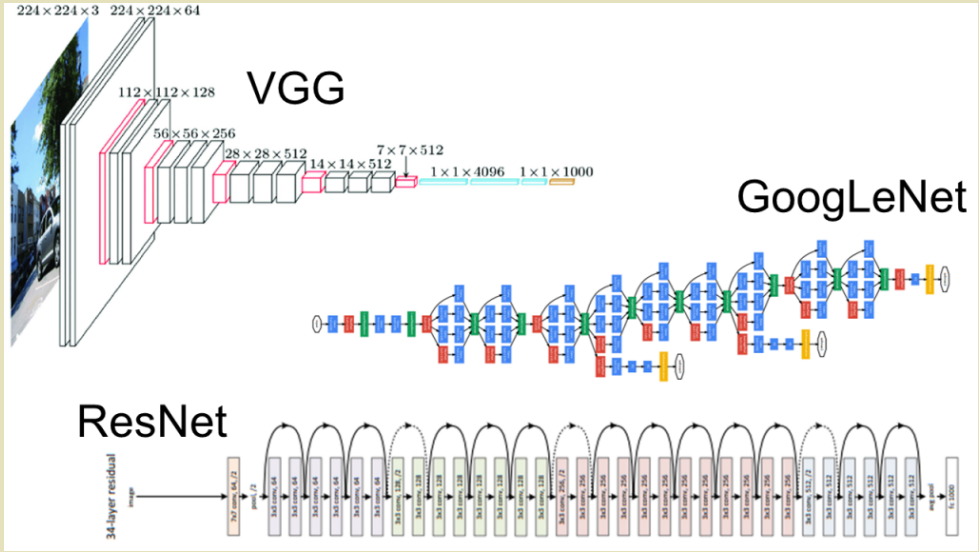
What is a neural network (nn)?

The point

Many layers → many parameters

These are good for approximating real world problems

Examples



ResNet-152 with 152. layers (used in transformer models such as ChatGPT)

VGG-19 with 19 layers (used in image classification)

GoogLeNet with 22 layers (used in face detection)

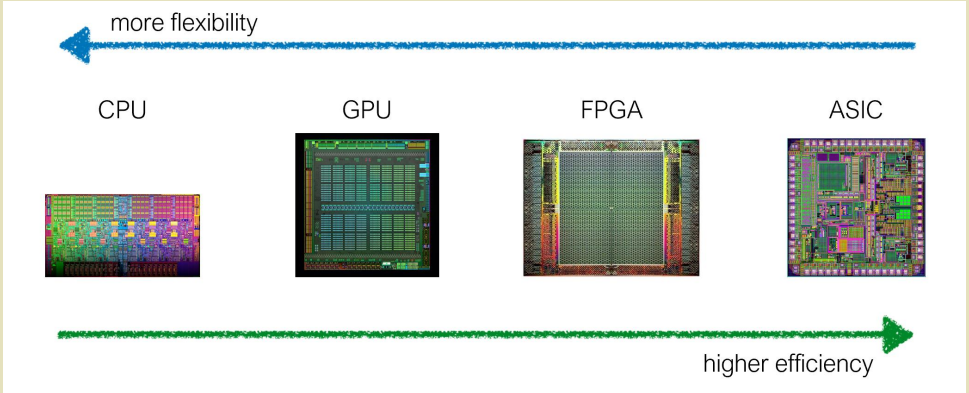
- k = number of the layer
- Deep = many layers = better approximation

What is a neural network (nn)?

Side fact

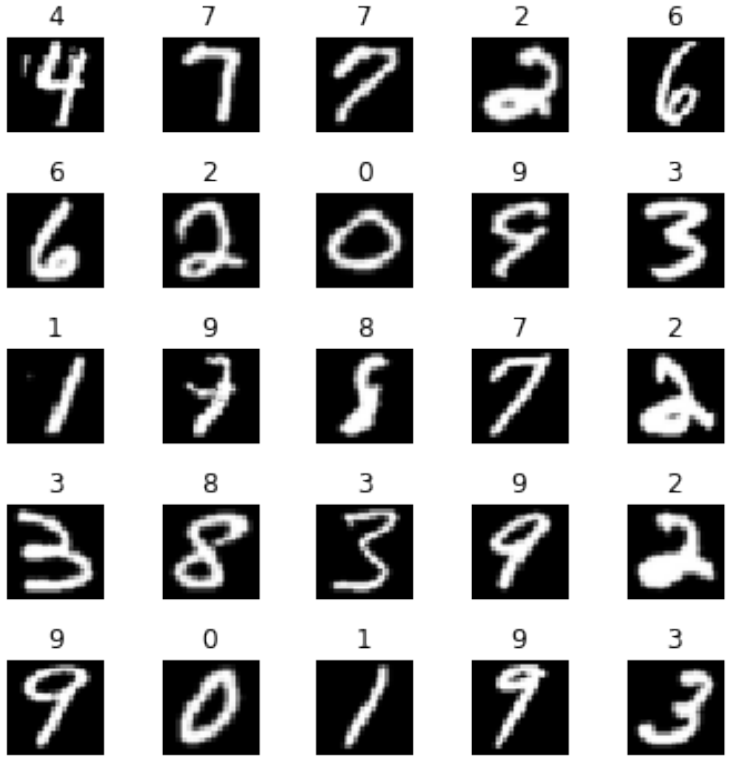
Gaming has improved AI!?

A GPU can do e.g. matrix multiplications faster than a CPU and lots of nn run on GPUs



- The a_{ij}^k and b
- k = number
- Deep = many layers = better approximation

How learning works



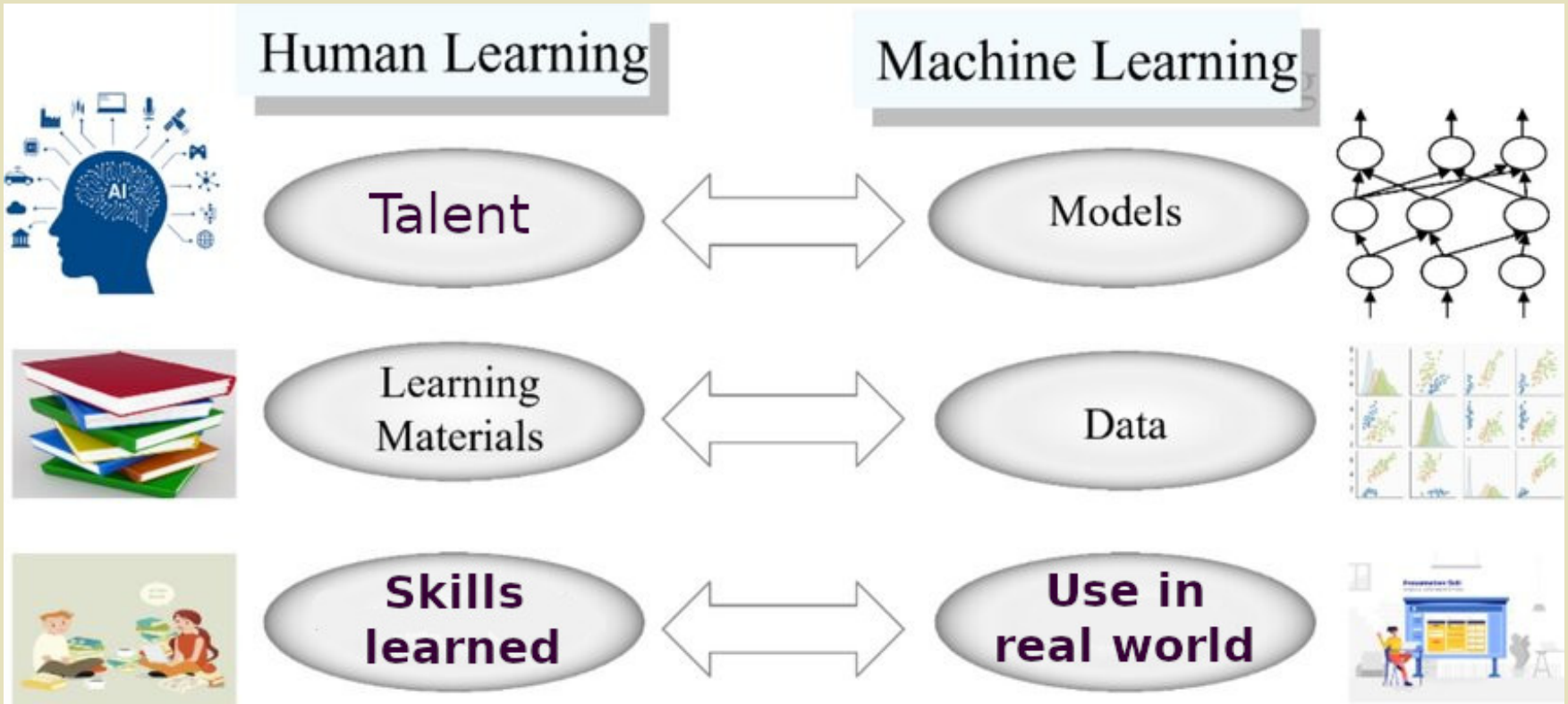
- Supervised learning Create a dataset with answers, e.g. pictures of handwritten digits plus their label
- There are other forms of learning e.g. unsupervised, which I skip
- Split the data into ≈80% training and ≈20% testing data

Idea to keep in mind

How to train students?
There are lectures, exercises etc., the **training data**

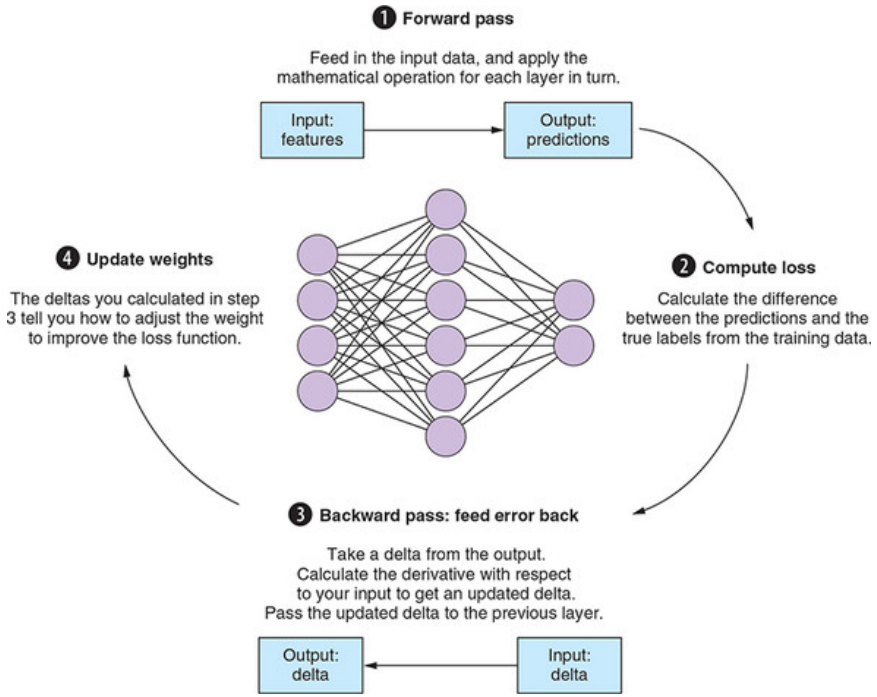
There is a final exam, the **testing data**

Upon performance, we **let them out into the wild**



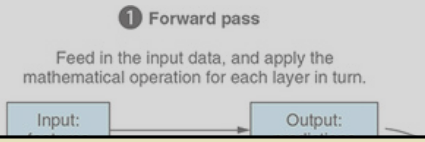
- Split the data into ≈80% training and ≈20% testing data

How learning works



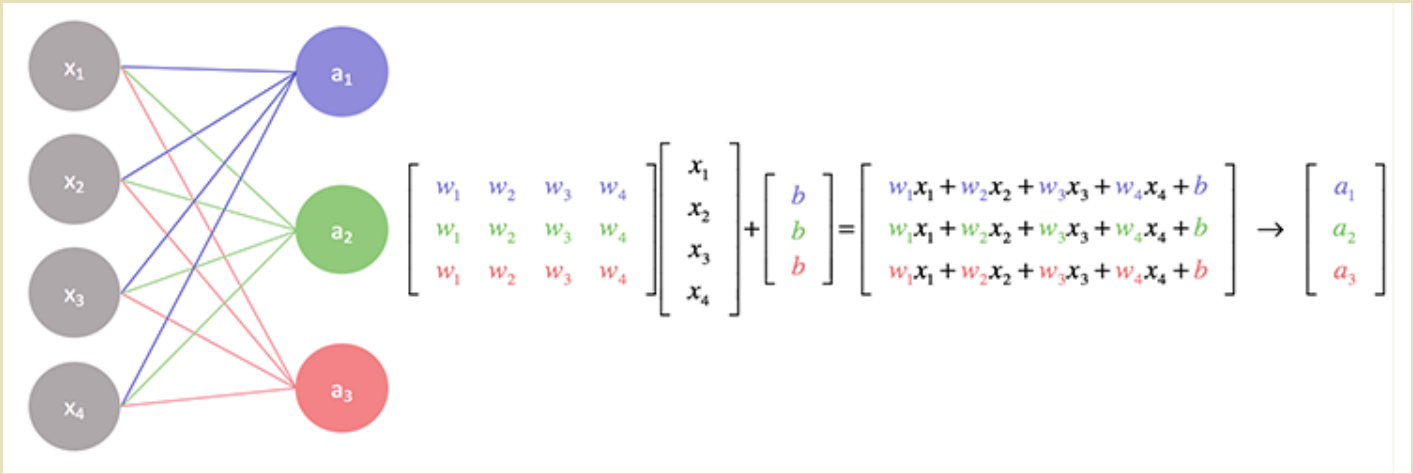
- Forward Run the nn = function on the training data
- Loss Calculate the difference "results - answers" ($\Rightarrow$ loss function)
- Backward Change the parameters trying to minimize the loss function
- Repeat

How learning works



Forward

Boils down to a bunch of matrix multiplications



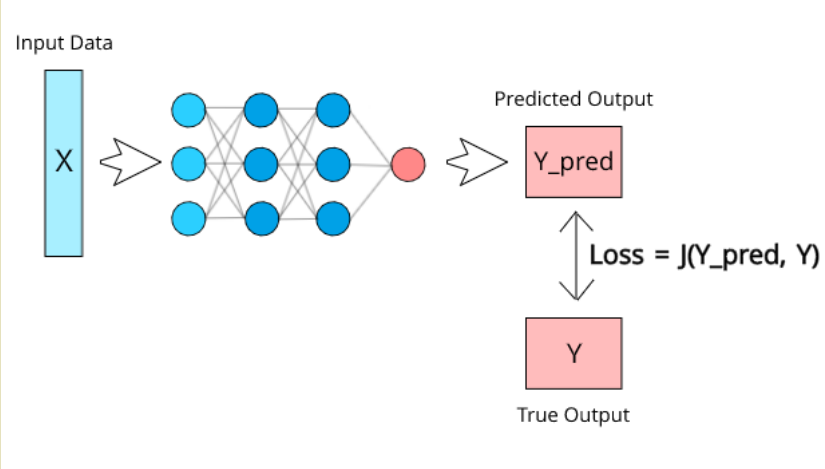
followed by the nonlinear activation e.g. ReLU

- Forw
- Loss
- Backward Change the parameters trying to minimize the loss function
- Repeat

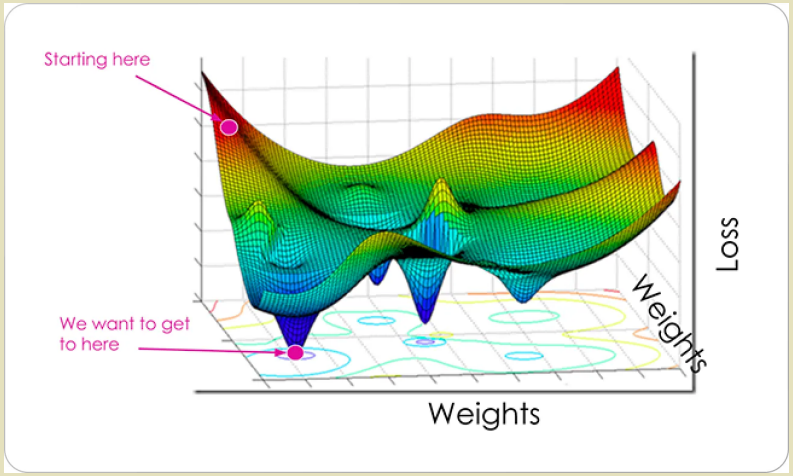
How learning wo

Loss

The difference between real values and predictions



Task Minimize loss function



- **Forward** Ru
- **Loss** Calcul
- **Backward** C
- **Repeat**

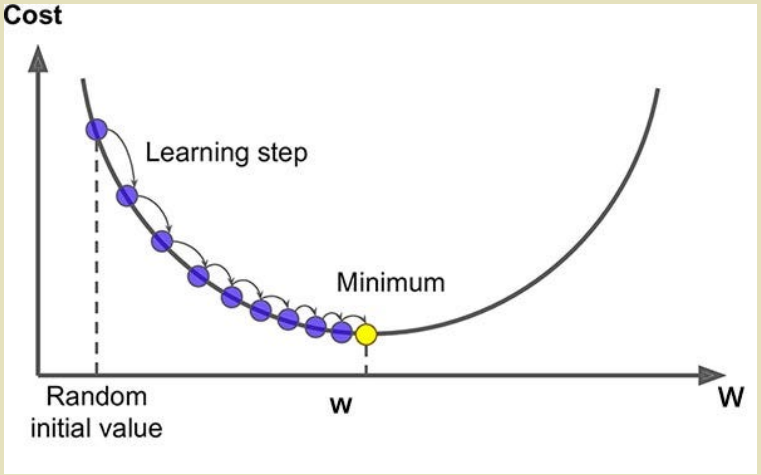
unction)

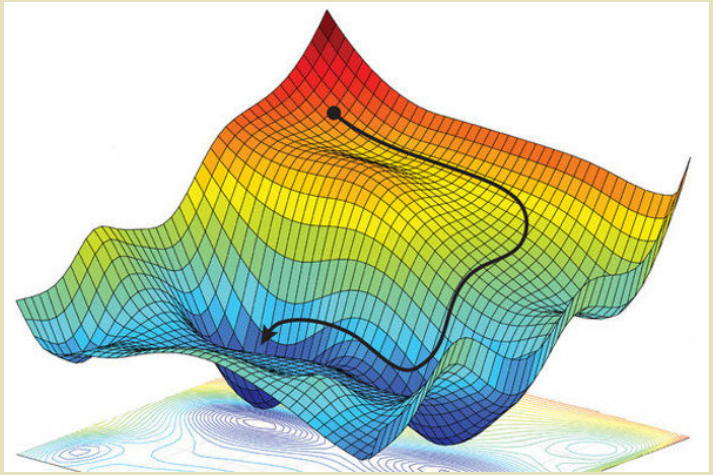
ss function

How learning

Backward

This is running gradient descent on the loss function





Slogan Adjust parameters following the steepest descent

- Forward
- Loss Ca tion)
- Backwar function
- Repeat

How learning works

1. Forward pass
Feed in the input data, and apply the

And what makes it even better:

You can try it yourself

My favorite tool is PyTorch but there are also other methods

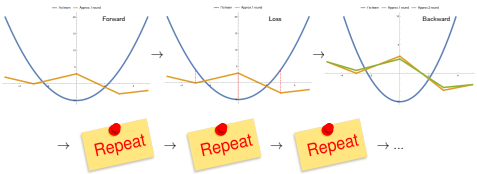
Let us see how!

- Forwar
- Loss C on)
- Backward Change the parameters trying to minimize the loss function
- Repeat

What is machine learning?

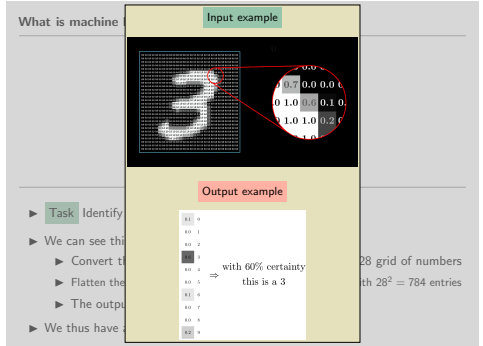
- Task Identify handwritten digits
- We can see this as a function in the following way:
  - Convert the pictures into grayscale values, e.g. 28×28 grid of numbers
  - Flatten the result into a vector, e.g. $28 \times 28 \mapsto$ a vector with $28^2 = 784$ entries
  - The output is a vector with 10 entries
- We thus have a function $\mathbb{R}^{784} \to \mathbb{R}^{10}$

What is machine learning?



- Machine learning mantra
  - Forward = calculate an approximation (start with random inputs)
  - Loss = compare to real data
  - Backward = adjust the approximation

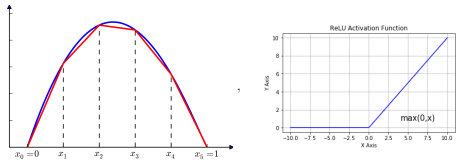
What is machine learning?



- Task Identify
- We can see thi
  - Convert t
  - Flatten the
  - The outpu
- We thus have

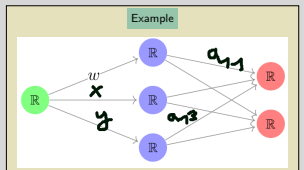
28 grid of numbers
th $28^2 = 784$ entries

What is machine learning?



- Idea Approximate the unknown function $\mathbb{R}^{784} \to \mathbb{R}^{10}$
- Neural network = a piecewise linear approximation (matrices + PL maps)
- The matrices = a bunch of numbers (weights) and offsets (biases)
- The PL maps = usually ReLU

What is a neural network (nn)?



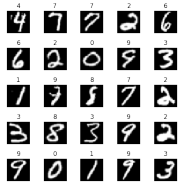
Here we have two matrices, a 3-by-1 and a 2-by-3 matrix

$$\begin{pmatrix} w \\ x \\ y \end{pmatrix} \text{ and } \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{pmatrix}$$

plus two bias terms as in $y = matrix \cdot x + bias$

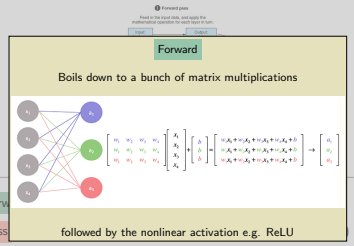
- NN = a
- The task of a nn is to approximate an unknown function
- It consist of neurons = entries of vectors, and weights = entries of matrices

How learning works



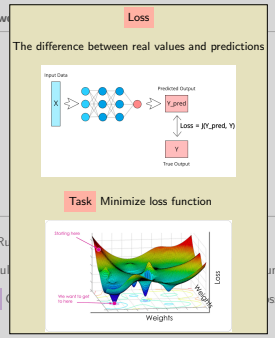
- Supervised learning Create a dataset with answers, e.g. pictures of handwritten digits plus their label
- There are other forms of learning e.g. unsupervised, which I skip
- Split the data into ≈80% training and ≈20% testing data

How learning works



- Forw
- Loss
- Backward Change the parameters trying to minimize the loss function
- Repeat

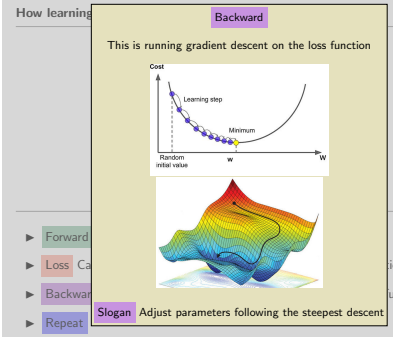
How learning w



- Forward Ru
- Loss Calcul
- Backward
- Repeat

unction)
ss function

How learning



- Forward
- Loss Ca
- Backwar
- Repeat

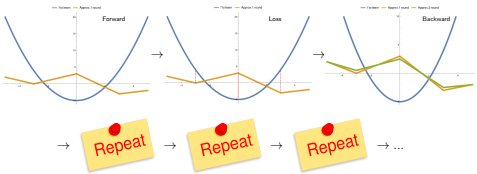
ion)
unction

There is still much to do...

What is machine learning?

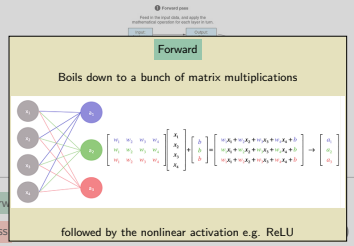
- Task Identify handwritten digits
- We can see this as a function in the following way:
  - Convert the pictures into grayscale values, e.g. 28×28 grid of numbers
  - Flatten the result into a vector, e.g. $28 \times 28 \mapsto$ a vector with $28^2 = 784$ entries
  - The output is a vector with 10 entries
- We thus have a function $\mathbb{R}^{784} \to \mathbb{R}^{10}$

What is machine learning?



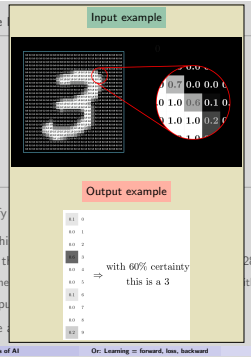
- Machine learning mantra
  - Forward = calculate an approximation (start with random inputs)
  - Loss = compare to real data
  - Backward = adjust the approximation

How learning works



- Forward
- Loss
- Backward Change the parameters trying to minimize the loss function
- Repeat

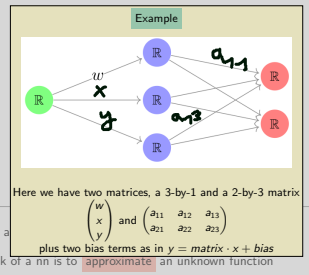
What is machine



- Task Identify
- We can see thi
  - Convert t
  - Flatten the
  - The outpu
- We thus have

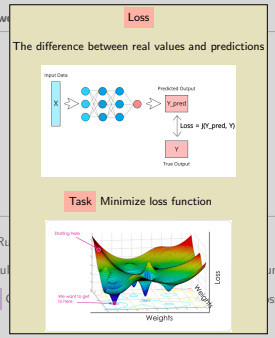
28 grid of numbers
th $28^2 = 784$ entries

What is a neural network (nn)?



- NN = a
- The task of a nn is to approximate an unknown function
- It consist of neurons = entries of vectors, and weights = entries of matrices

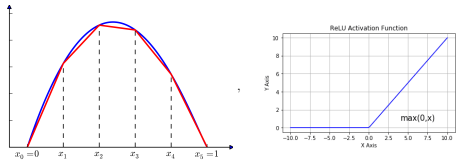
How learning w



- Forward Ru
- Loss Calcul
- Backward C
- Repeat

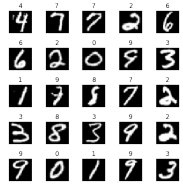
unction)
ss function

What is machine learning?



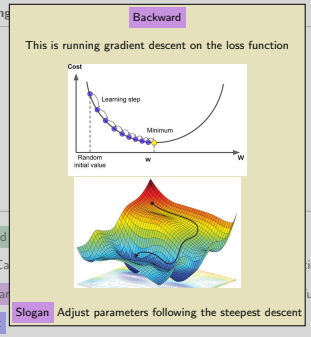
- Idea Approximate the unknown function $\mathbb{R}^{784} \to \mathbb{R}^{10}$
- Neural network = a piecewise linear approximation (matrices + PL maps)
- The matrices = a bunch of numbers (weights) and offsets (biases)
- The PL maps = usually ReLU

How learning works



- Supervised learning Create a dataset with answers, e.g. pictures of handwritten digits plus their label
- There are other forms of learning e.g. unsupervised, which I skip
- Split the data into ≈80% training and ≈20% testing data

How learning



- Forward
- Loss Ca
- Backwar
- Repeat

ion)
unction

Thanks for your attention!