Deep Learning

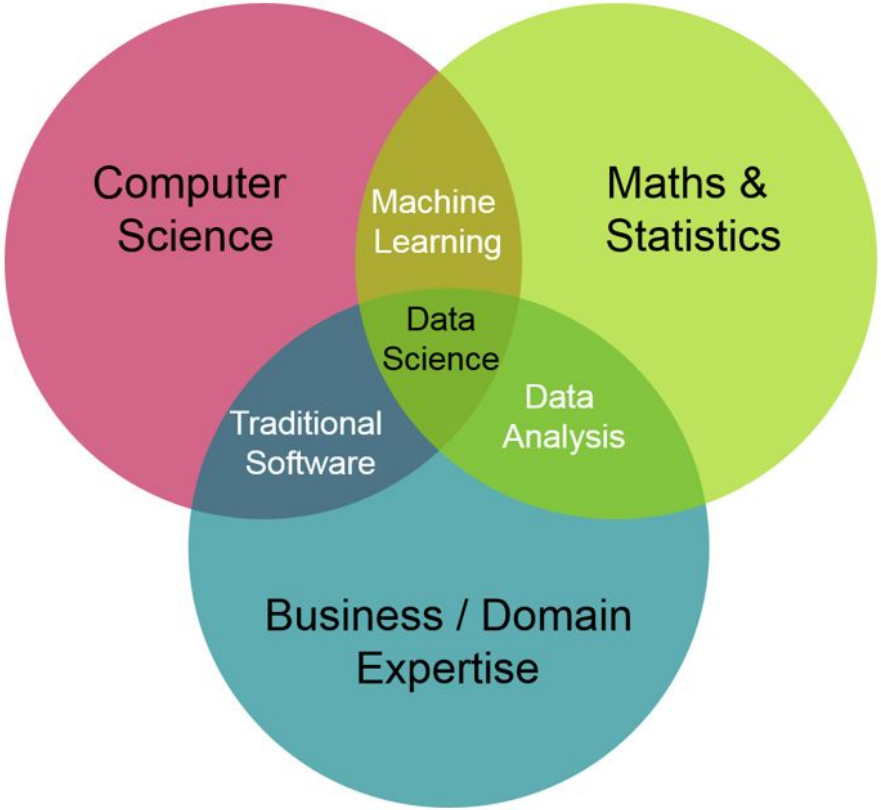
HSE University

Aziz Temirkhanov

2024

Intro

What is Deep Learning?

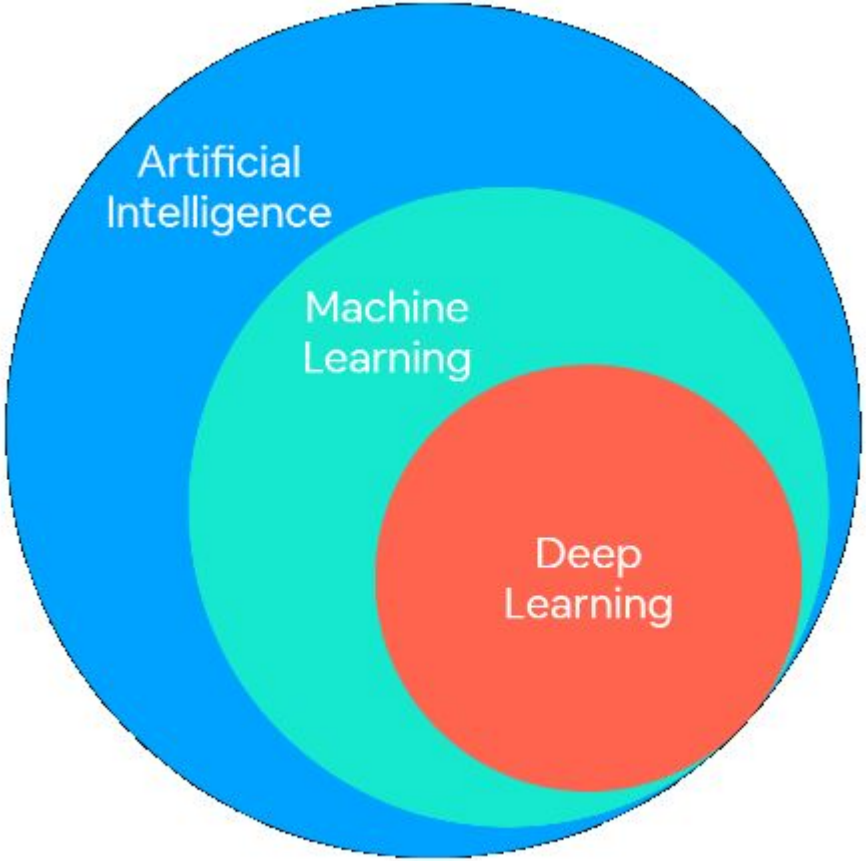


What is Deep Learning?

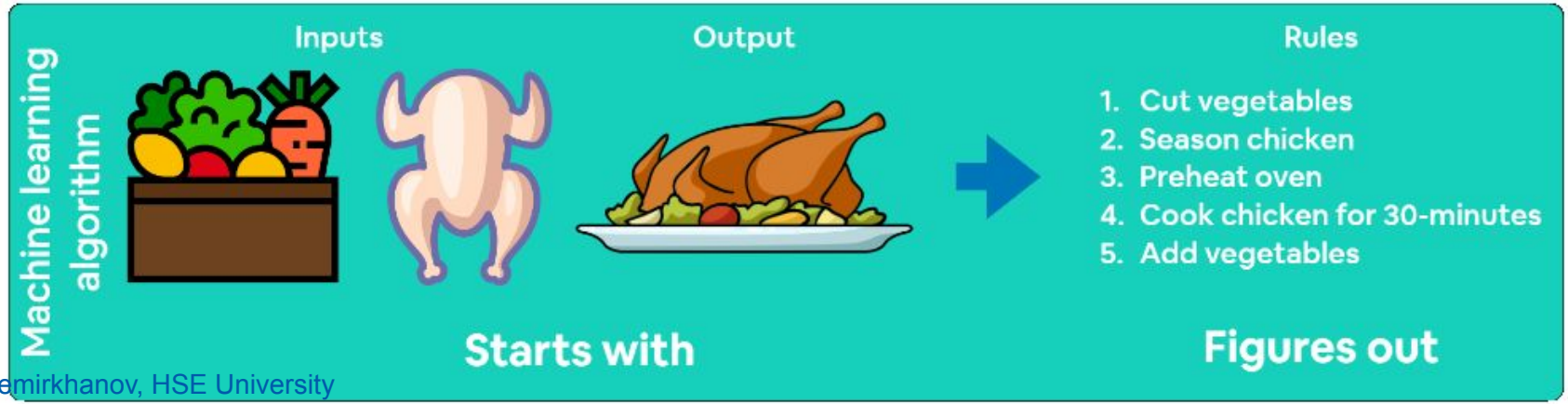
Machine learning is turning things (data) into numbers and finding patterns in those numbers.

The computer does this part.
How?
Code & math.
We're going to be writing the code.

What is Deep Learning?



What is Deep Learning?



ML vs DL

ML

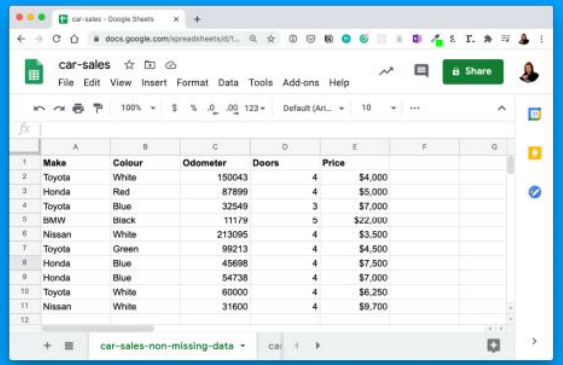
- Simple
- Explainable
- Less data required
- Feature engineering is required

DL

- Complex
- Unexplainable
- A lot of data is required
- No feature engineering
- Higher quality on complex tasks, but more computation cost

ML vs DL

Machine Learning



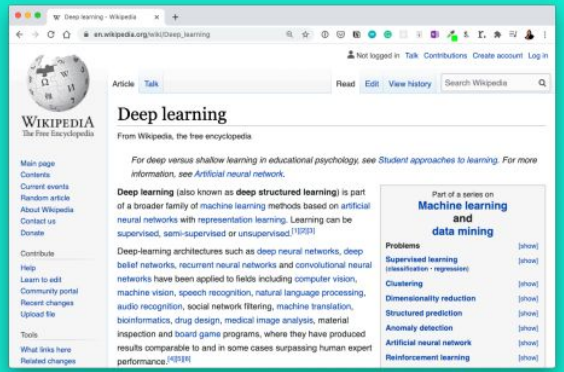
	Make	Colour	Odometer	Doors	Price
2	Toyota	White	150043	4	$4,000
3	Honda	Red	87899	4	$5,000
4	Toyota	Blue	32549	3	$7,000
5	BMW	Black	11179	5	$22,000
6	Nissan	White	213095	4	$3,500
7	Toyota	Green	99213	4	$4,500
8	Honda	Blue	45698	4	$7,500
9	Honda	Blue	54738	4	$7,000
10	Toyota	White	60000	4	$6,250
11	Nissan	White	31600	4	$9,700

Algorithm: gradient boosted machine

dmlc XGBoost

Structured data

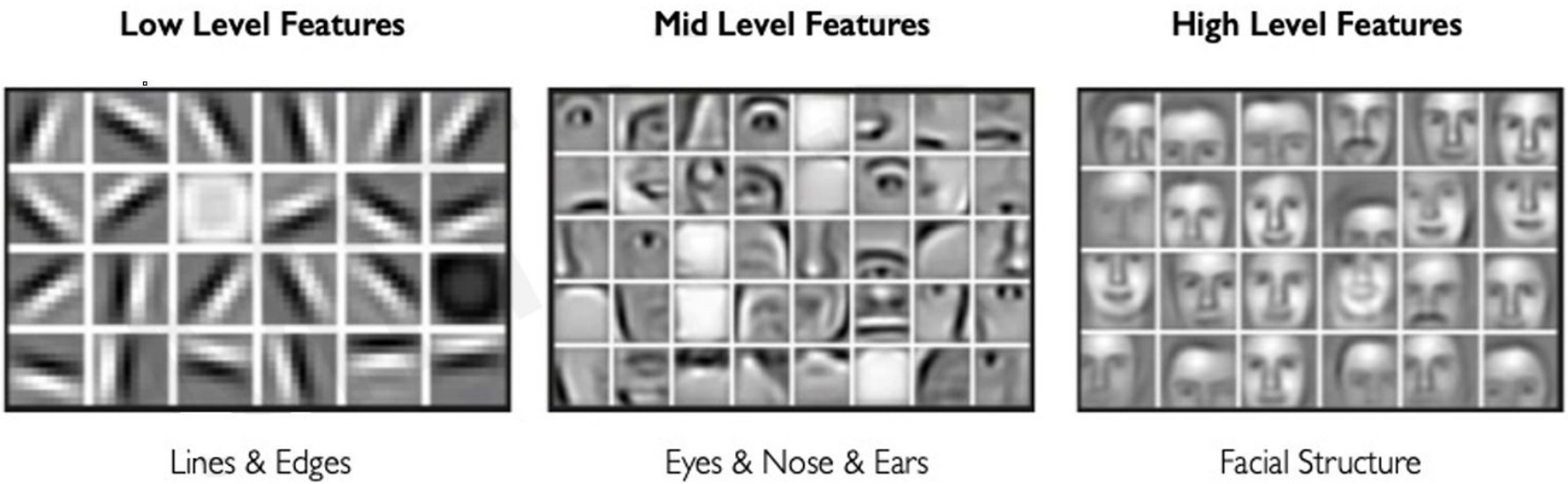
Deep Learning



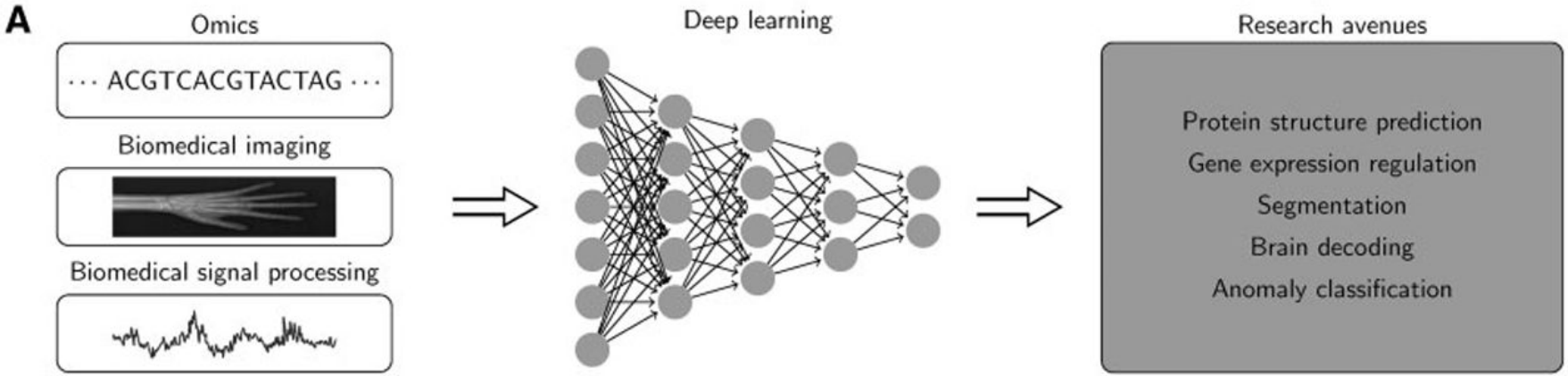
Algorithm: neural network

Unstructured data

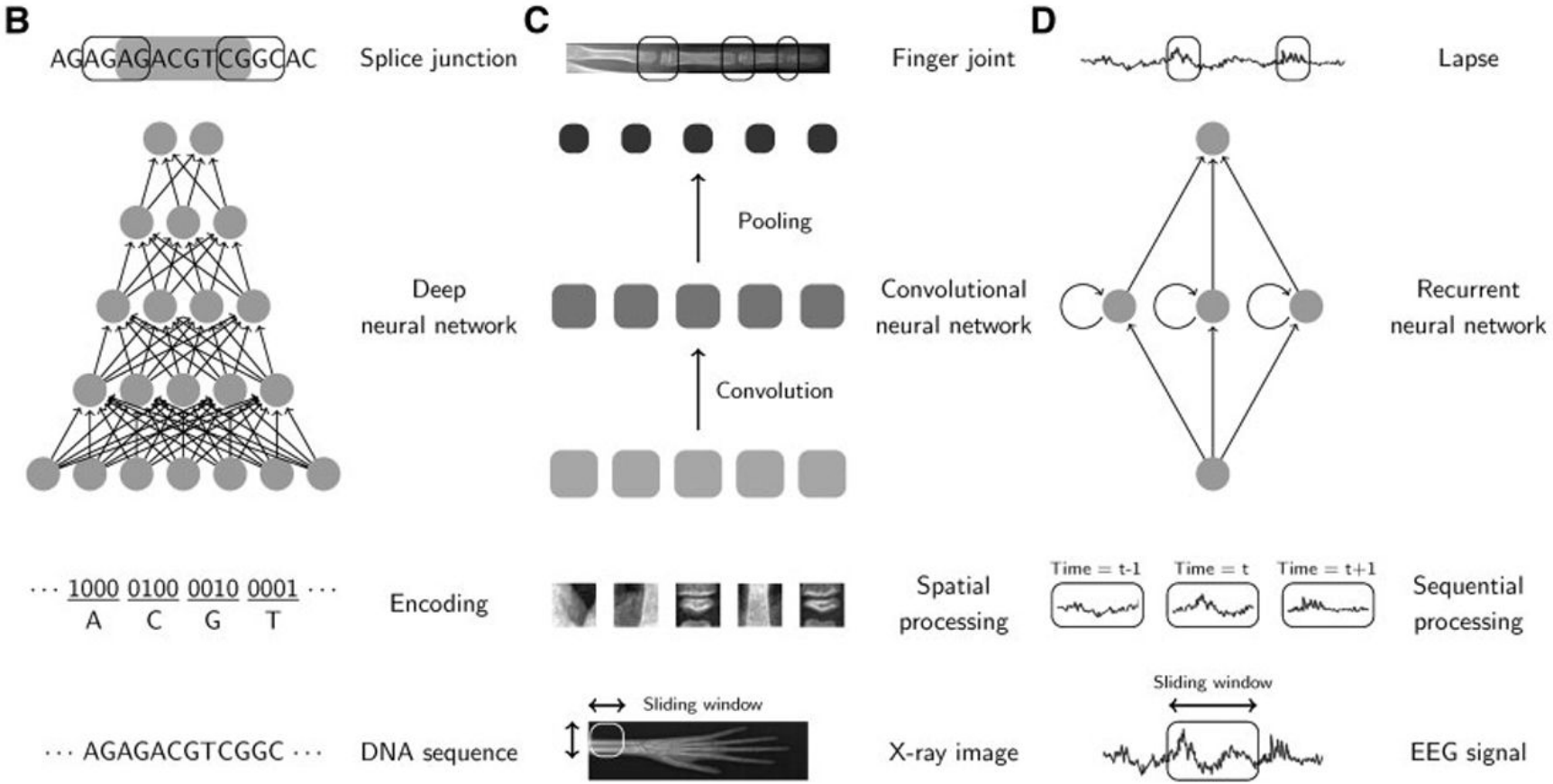
Feature engineering



Applications



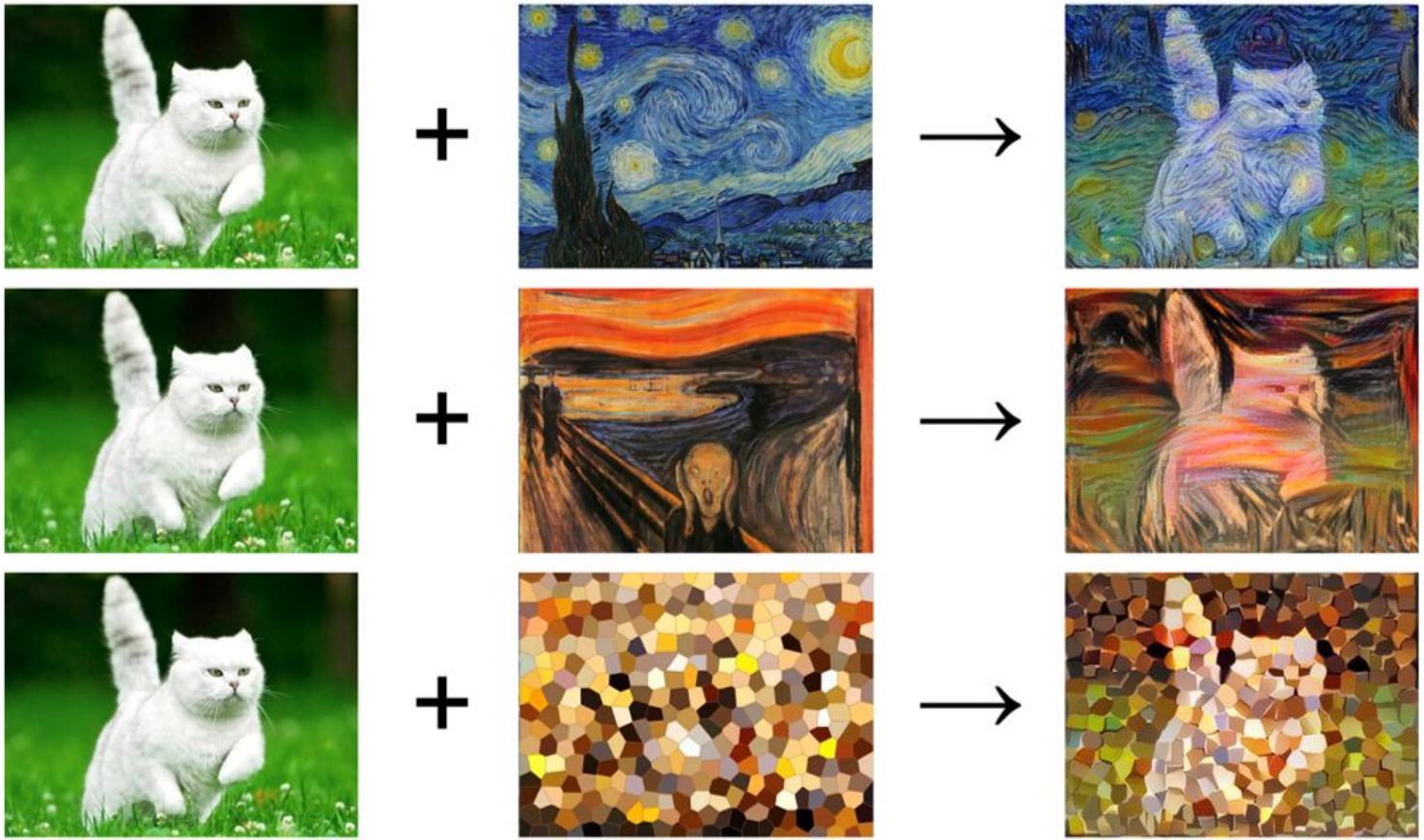
Applications



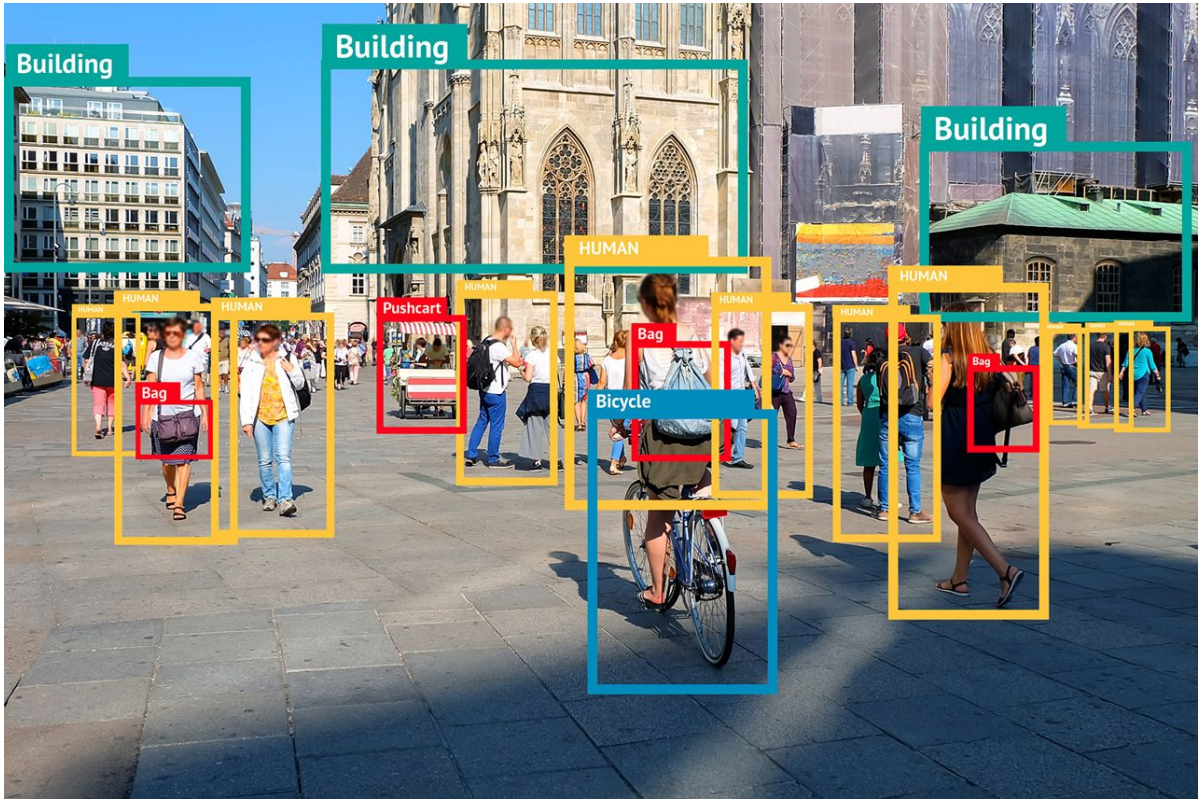
Applications

Source: Goodfellow et al., 2014; Radford et al., 2016; Liu & Tuzel, 2016; Karras et al., 2018; Karras et al., 2019; Goodfellow, 2019; Karras et al., 2020; AI Index, 2021; Vahdat et al., 2021

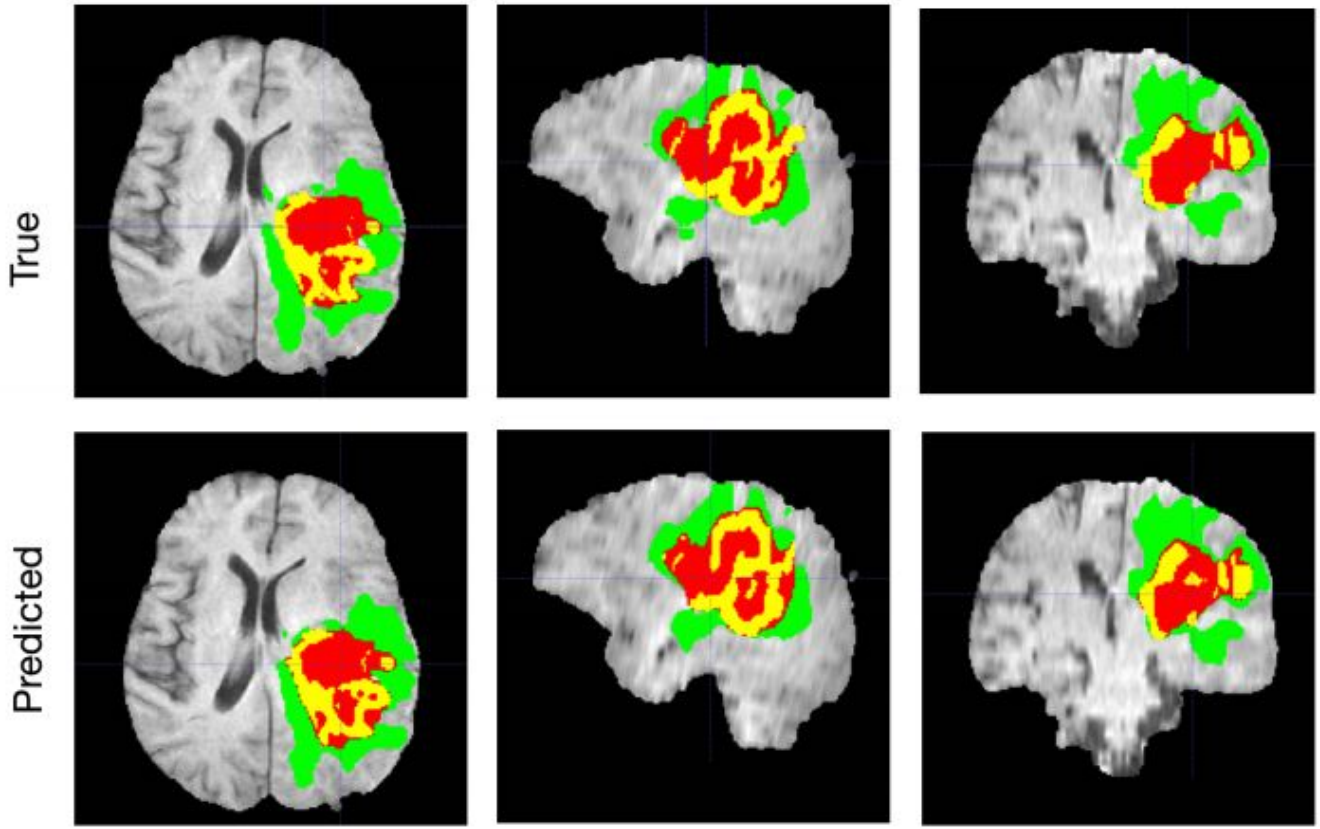
Applications



Applications

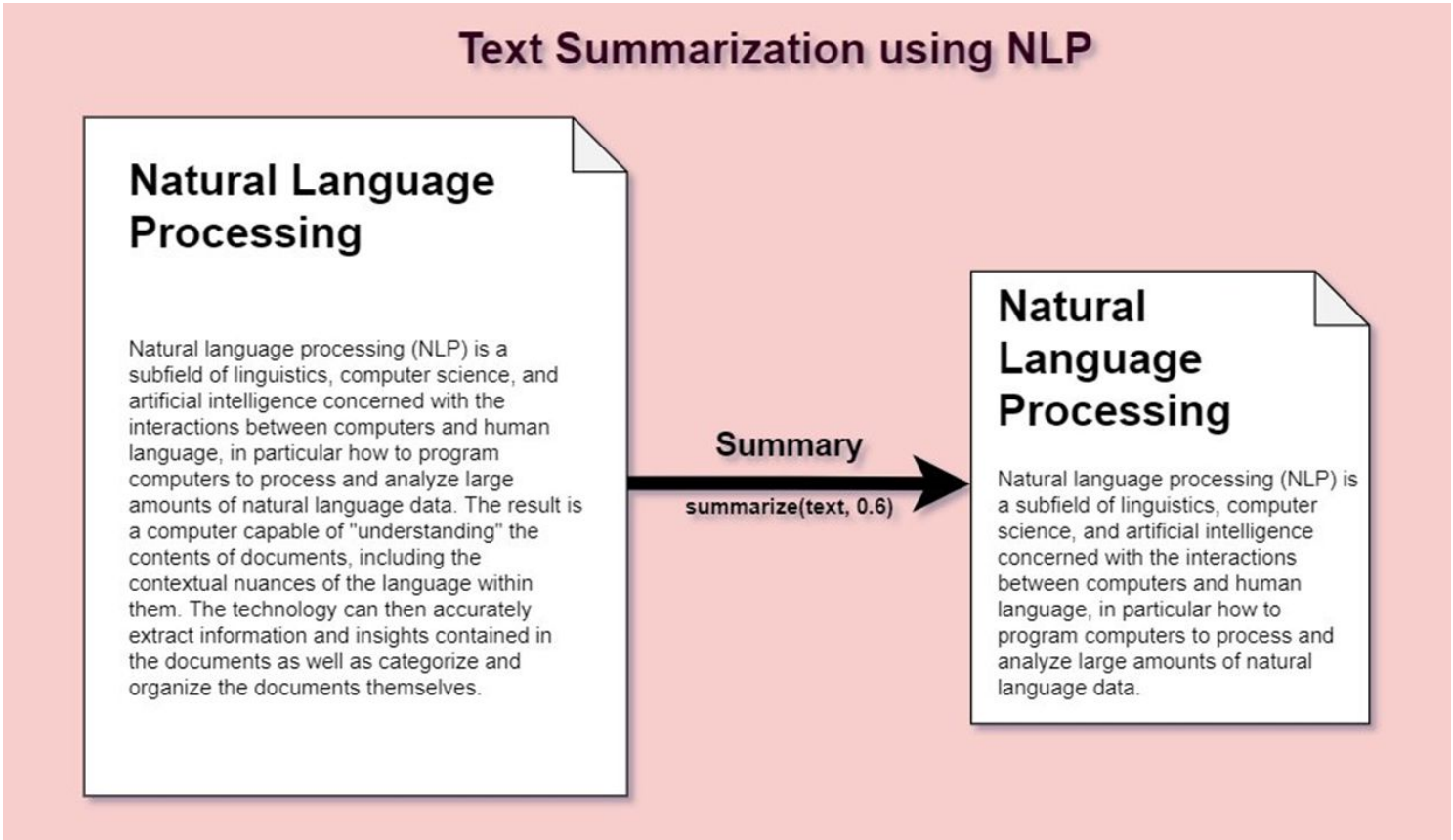


Applications



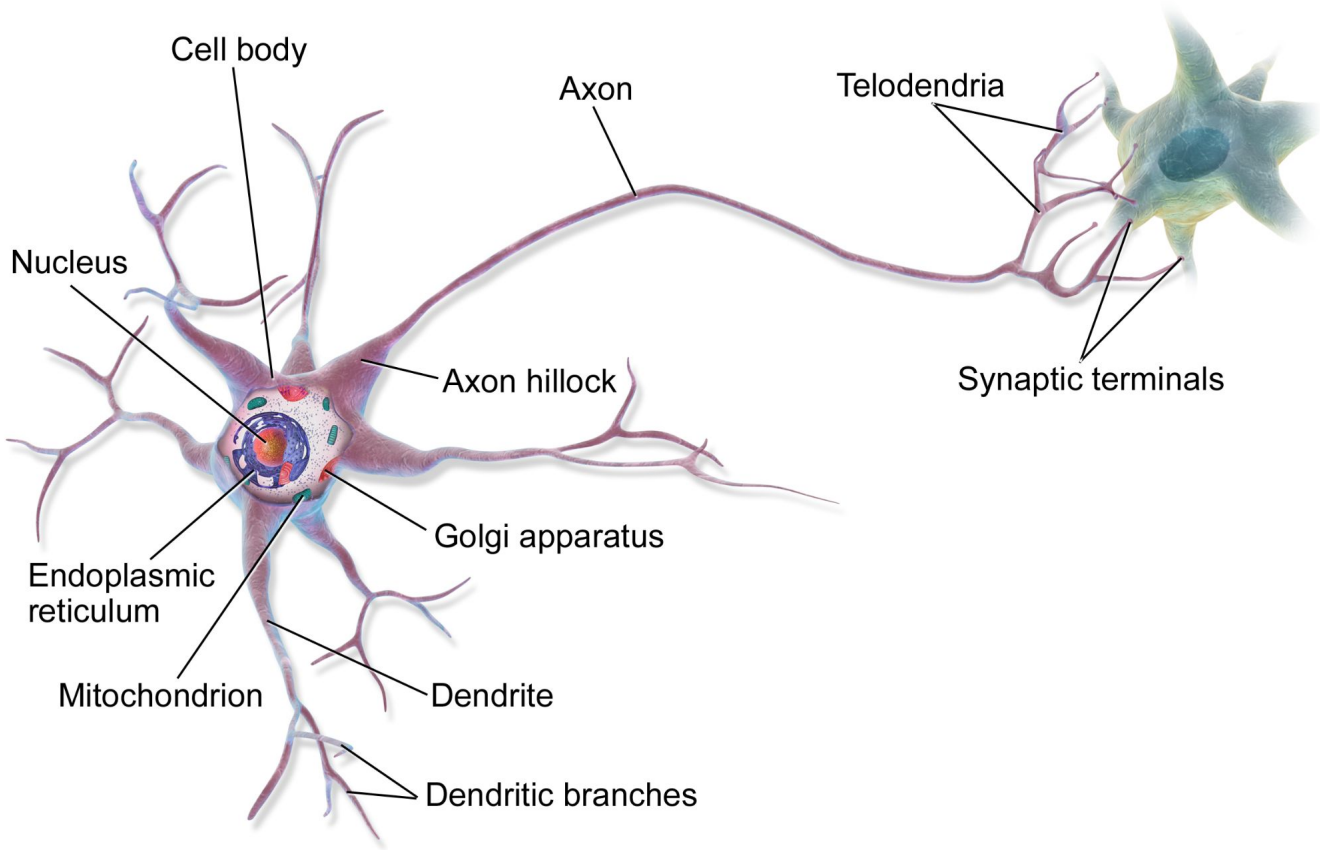
https://developer.nvidia.com/blog/automatically-segmenting-brain-tumors-with-ai/

Applications

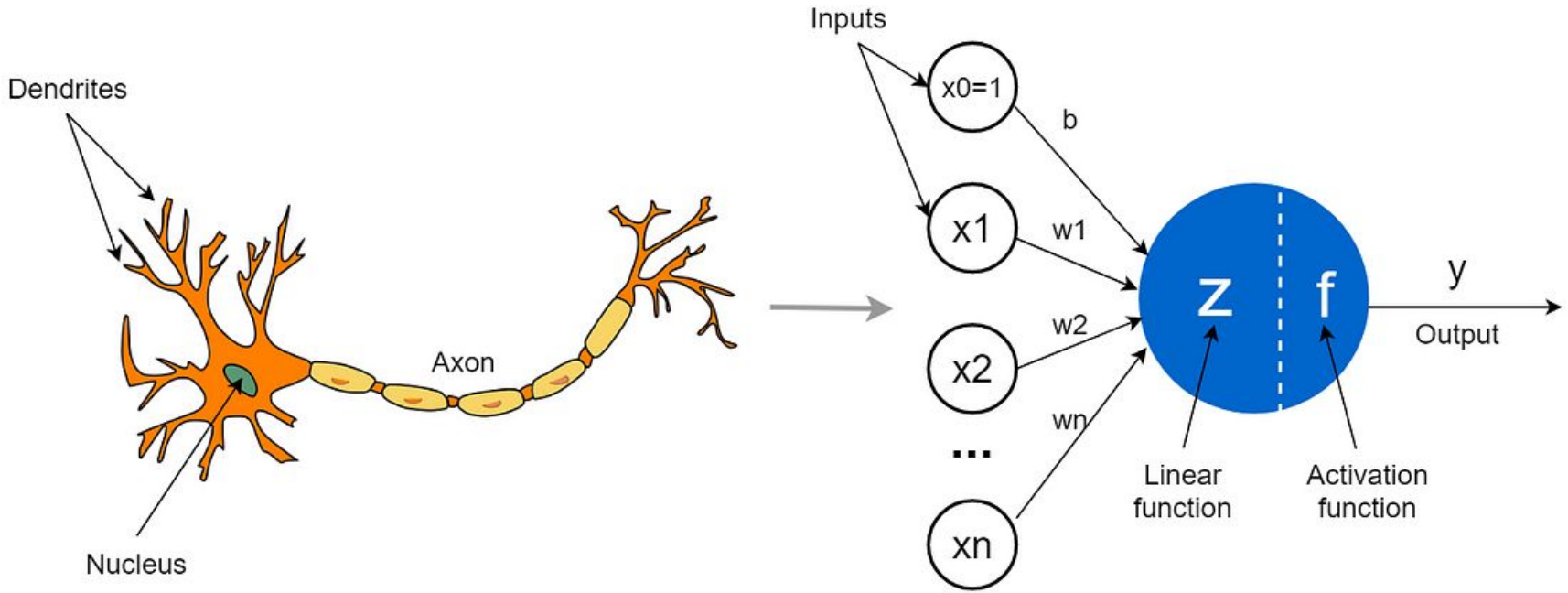


Perceptron

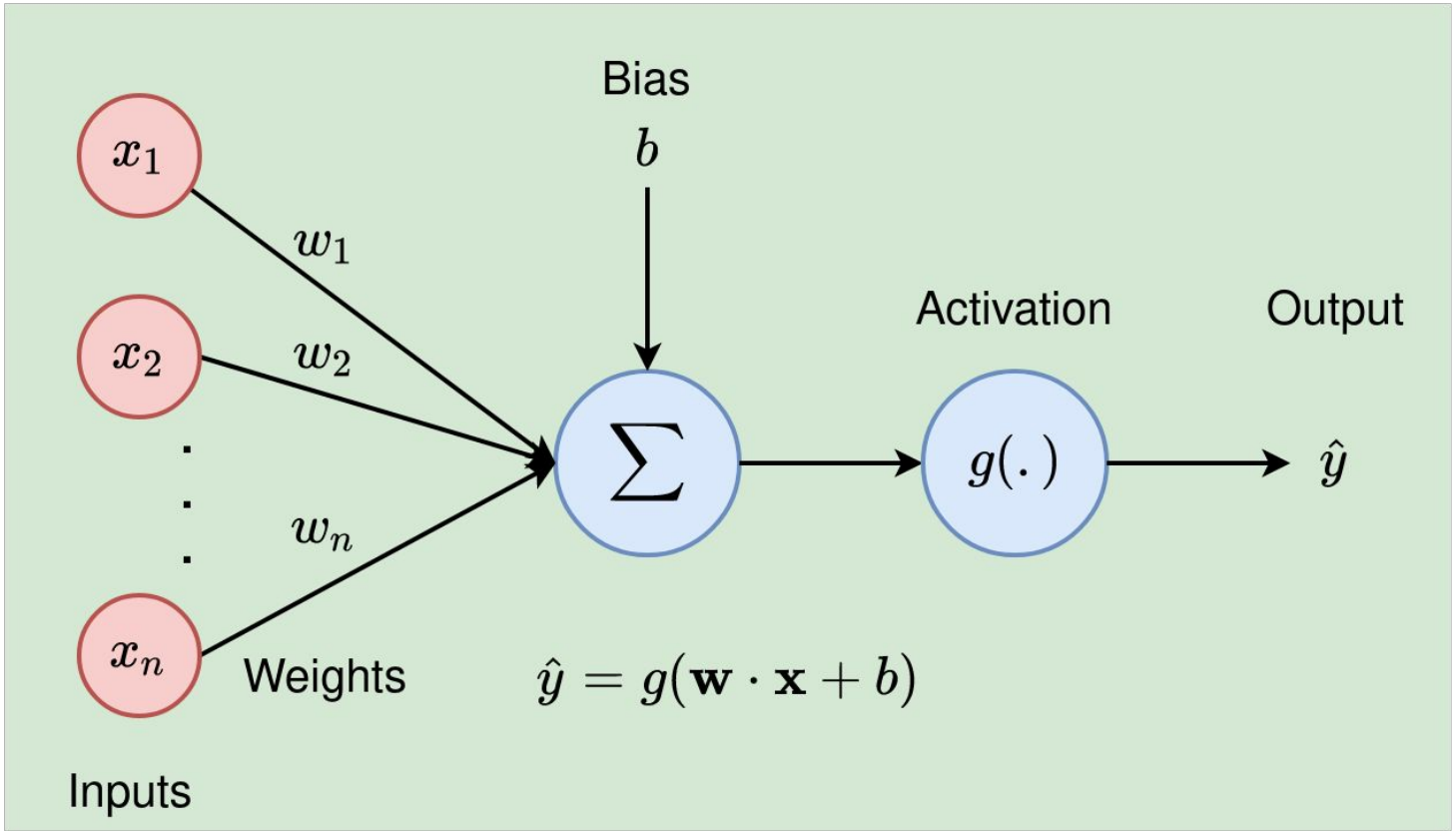
Neuron


Cell body
Axon
Telodendria
Nucleus
Axon hillock
Synaptic terminals
Golgi apparatus
Endoplasmic reticulum
Mitochondrion
Dendrite
Dendritic branches

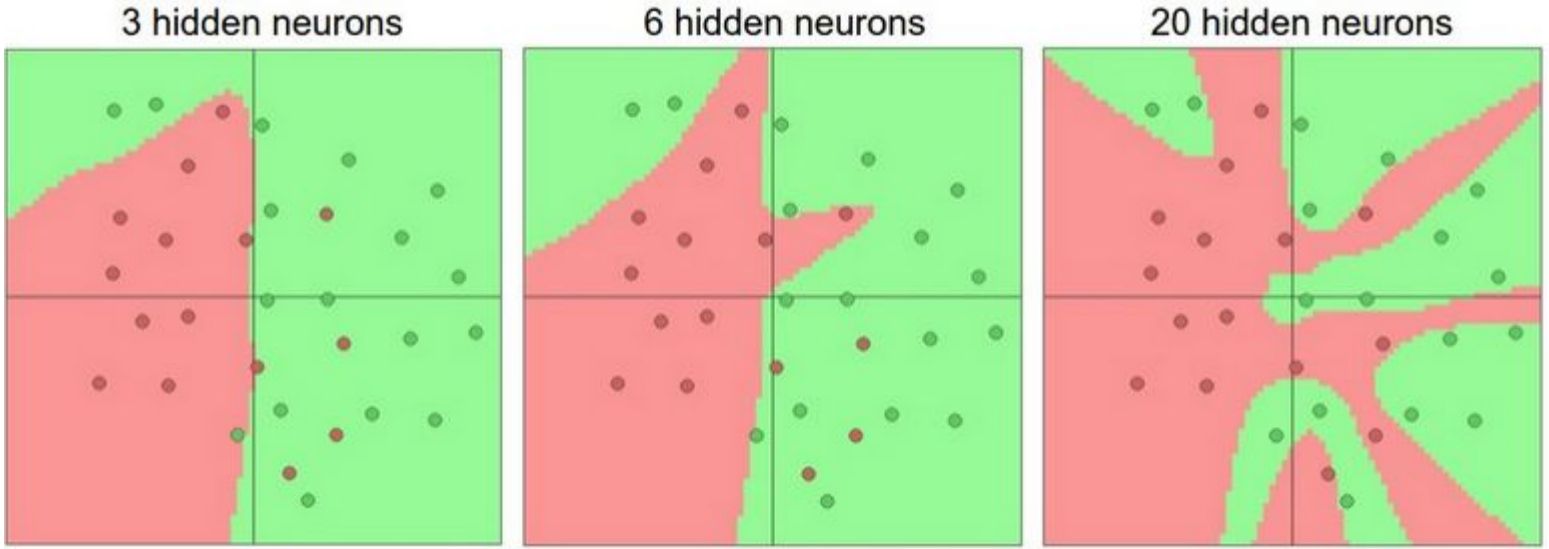
Perceptron


Dendrites
Axon
Nucleus
Inputs
x0=1
x1
x2
...
xn
b
w1
w2
wn
z
f
y
Output
Linear function
Activation function

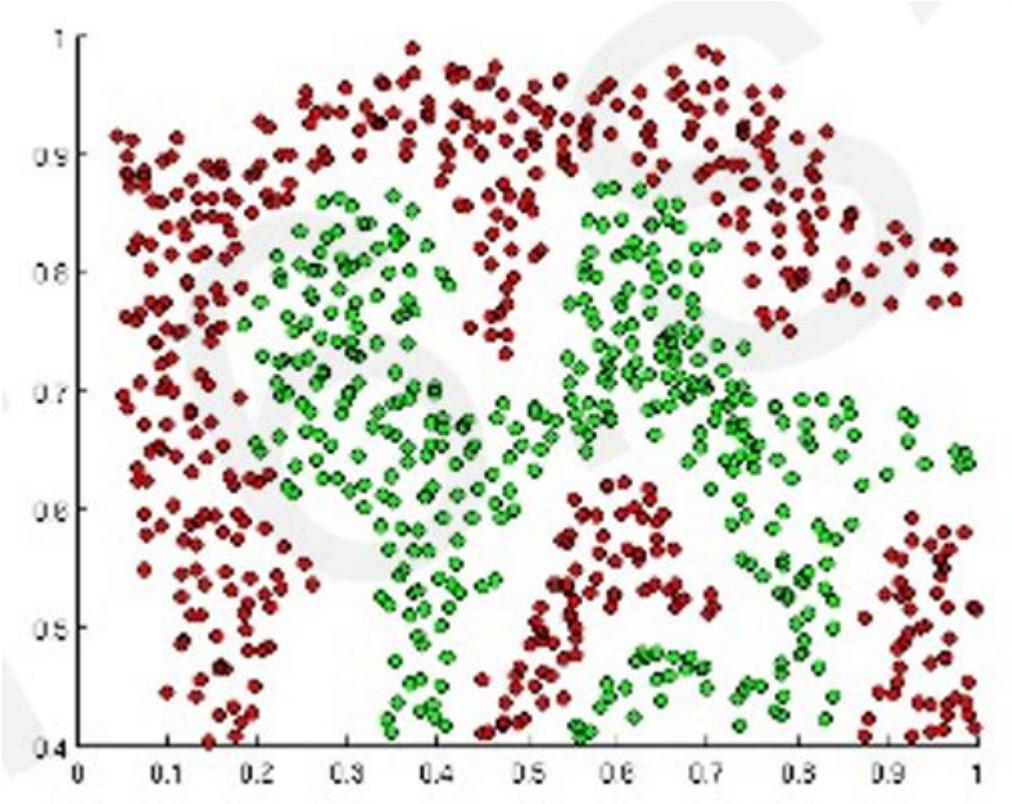
Perceptron


Bias
b
x_1
x_2
x_n
w_1
w_2
w_n
$\sum$
Activation
$g(.)$
Output
$\hat{y}$
Weights
$\hat{y} = g(\mathbf{w} \cdot \mathbf{x} + b)$
Inputs

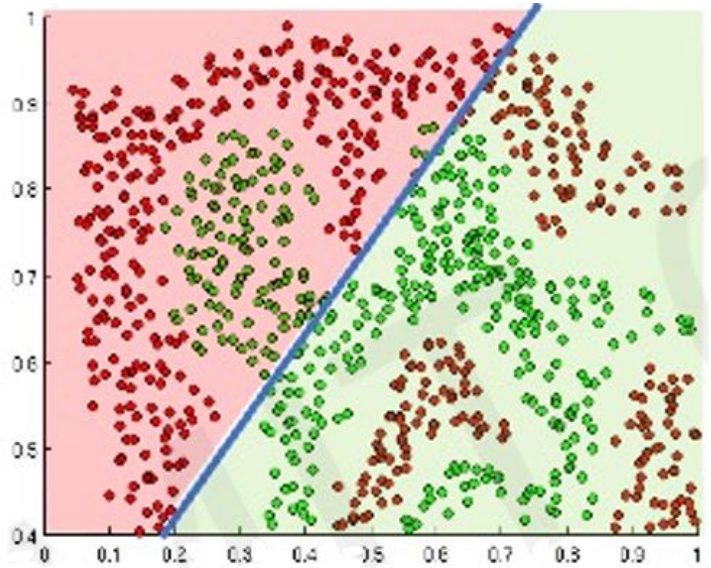
Multilayer Perceptron

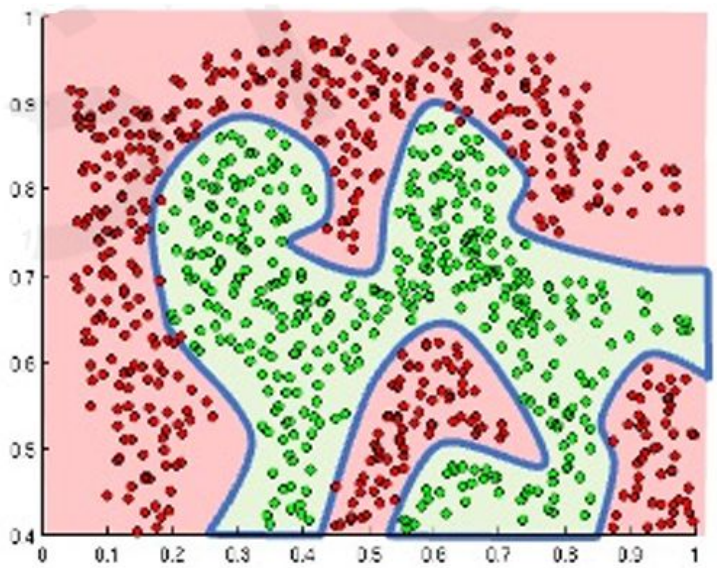


Deep Networks



Deep Networks

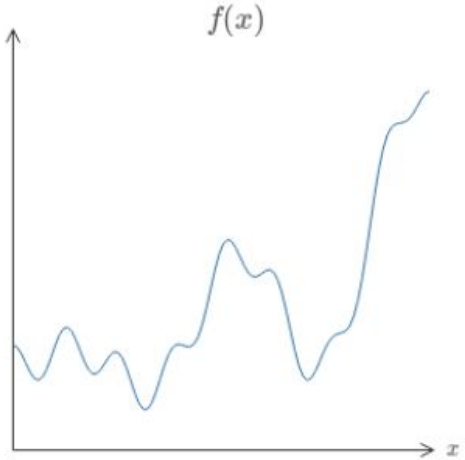


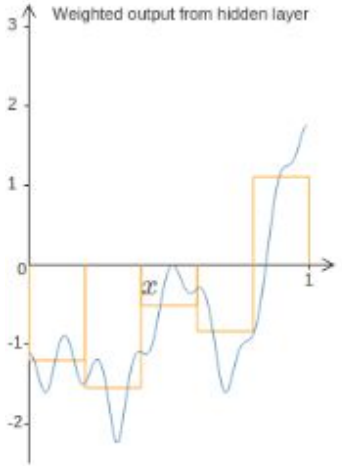


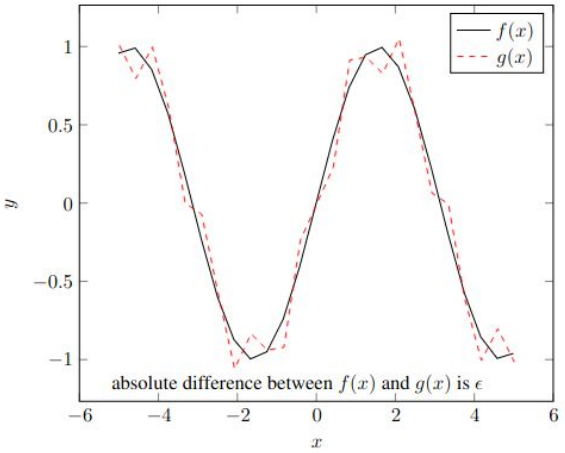
Universal approximation theorem

Theorem 3. *(Universal Approximation Theorem for Width-Bounded ReLU Networks). For any Lebesgue-integrable function $f : \mathbb{R}^n \to \mathbb{R}$ and any $\epsilon > 0$, there exists a fully-connected ReLU network A with width $d_m \leq n + 4$, such that the function F_A represented by this network satisfies*

$$\int_{\mathbb{R}^n} |f(x) - F_A(x)| dx < \epsilon$$







How to train NN

Training

- Define a problem (e.g. classification, ranking, etc.)
- Find a dataset
- Prepare data (cleaning, labeling, augmentation, etc.)
- Set an objective function
- Choose an evaluation metric
- Set or choose a baseline model
- Record every experiment