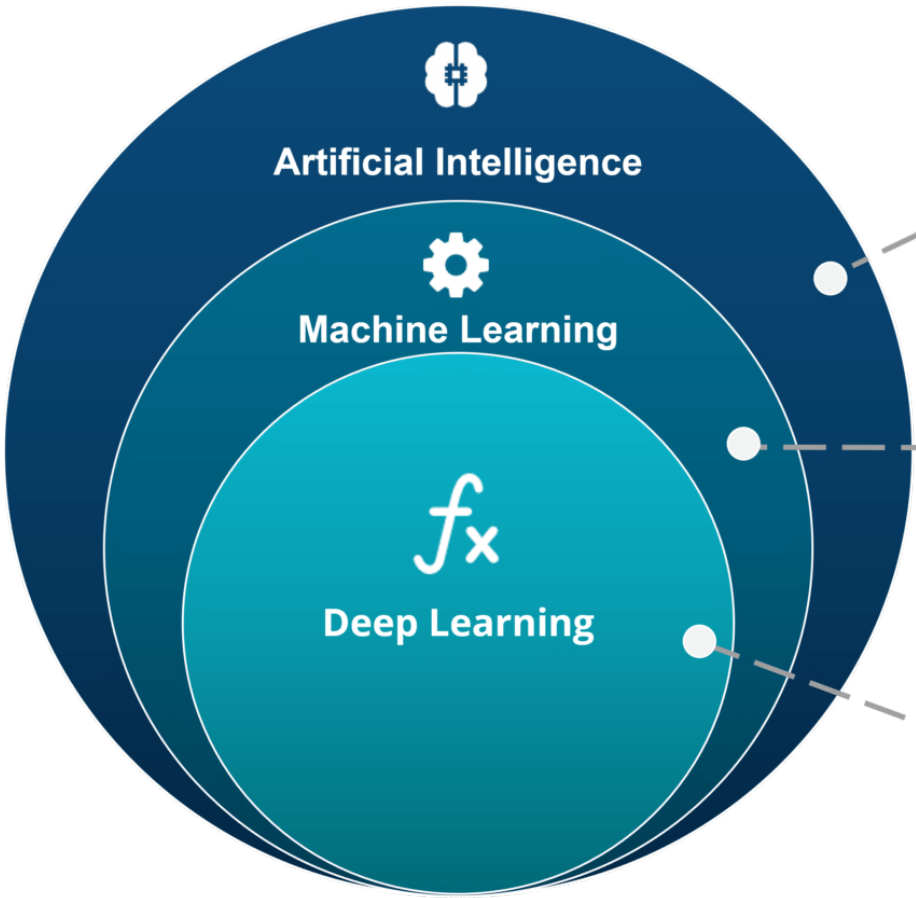
ML Refresher and History of Deep Learning

38-616, Spring 2023

Olexandr Isayev
Department of Chemistry, CMU
olexandr@cmu.edu



ARTIFICIAL INTELLIGENCE

A technique which enables machines to mimic human behaviour

MACHINE LEARNING

Subset of AI technique which use statistical methods to enable machines to improve with experience

DEEP LEARNING

Subset of ML which make the computation of multi-layer neural network feasible

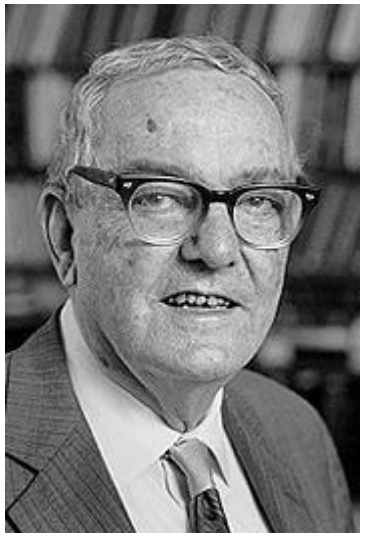
Artificial Intelligence (AI) the theory and development of computer systems able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.

Machine learning (**ML**) is the study of algorithms and techniques that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead.

Machine learning (**ML**) is the collection of algorithms that learn from experiences and their outcomes and is able to predict the outcome of new experience

Machine Learning

- **Herbert Alexander Simon**:
  "Learning is any process by which a system improves performance from experience."
- "Machine Learning is concerned with computer programs that automatically improve their performance through experience. "

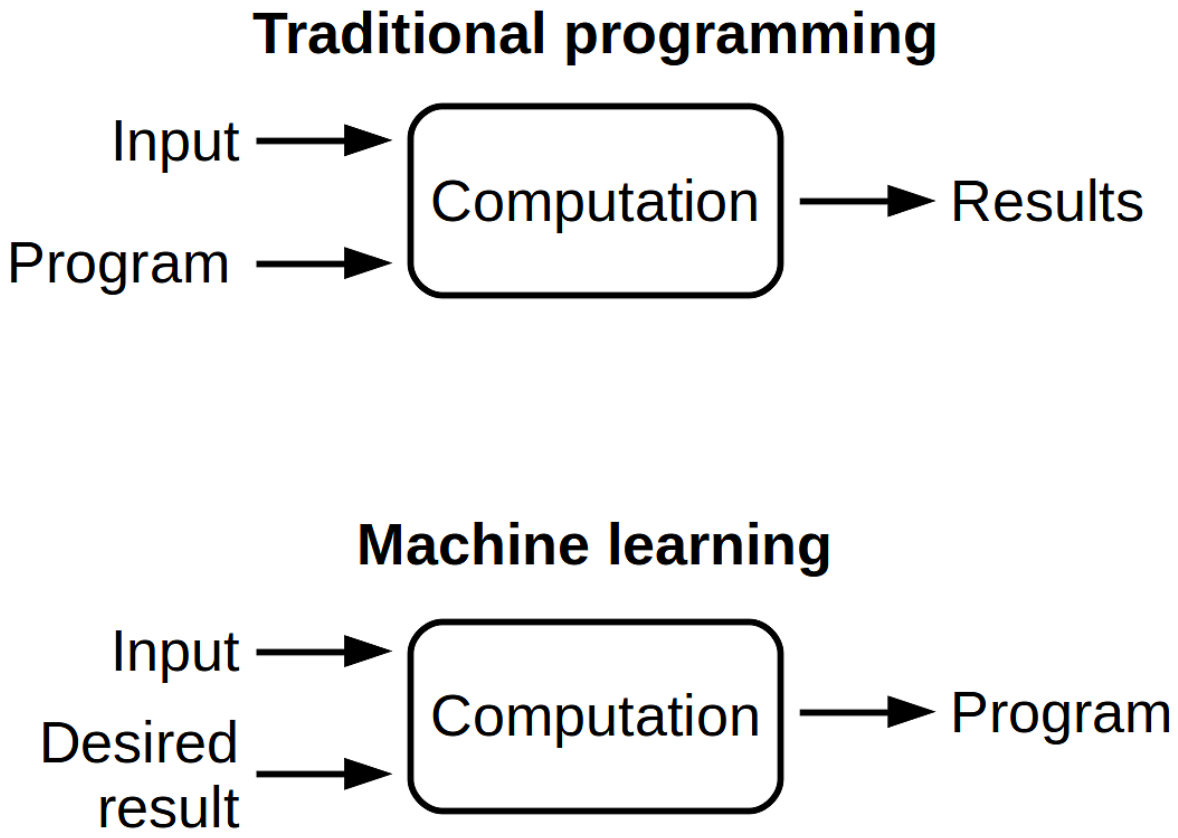


Herbert Simon
Turing Award 1975
Nobel Prize in Economics 1978

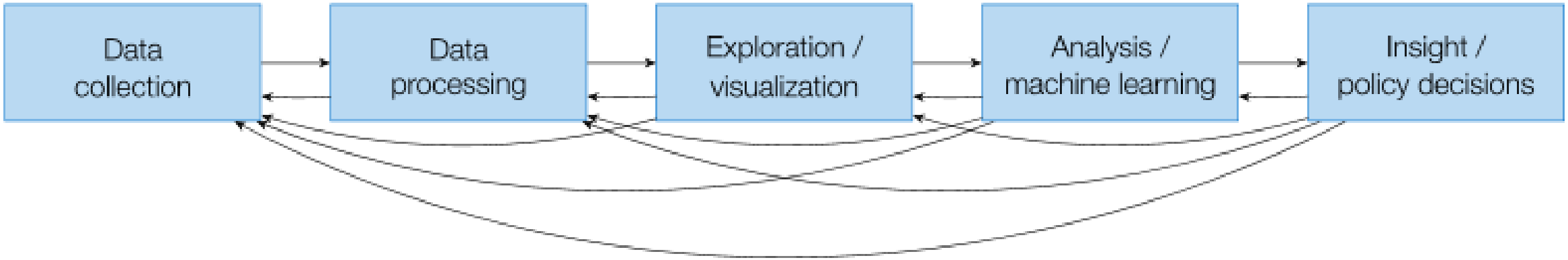
Machine Learning

- Machine Learning
  - Study of algorithms that
  - improve their performance
  - at some task
  - with experience
- Optimize a performance criterion using example data or past experience.
- Role of Statistics: Inference from a sample
- Role of Computer science: Efficient algorithms to
  - Solve the optimization problem
  - Representing and evaluating the model for inference

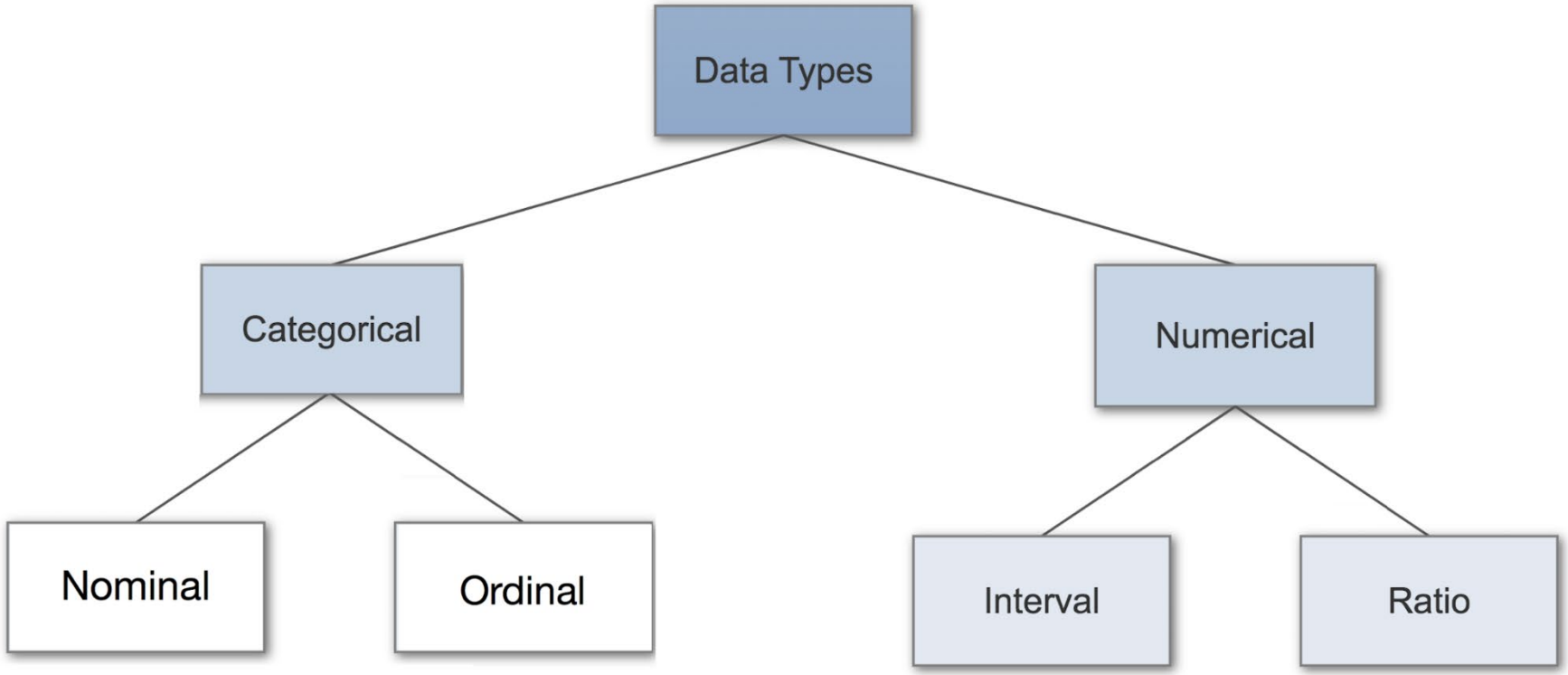
ML vs Programming



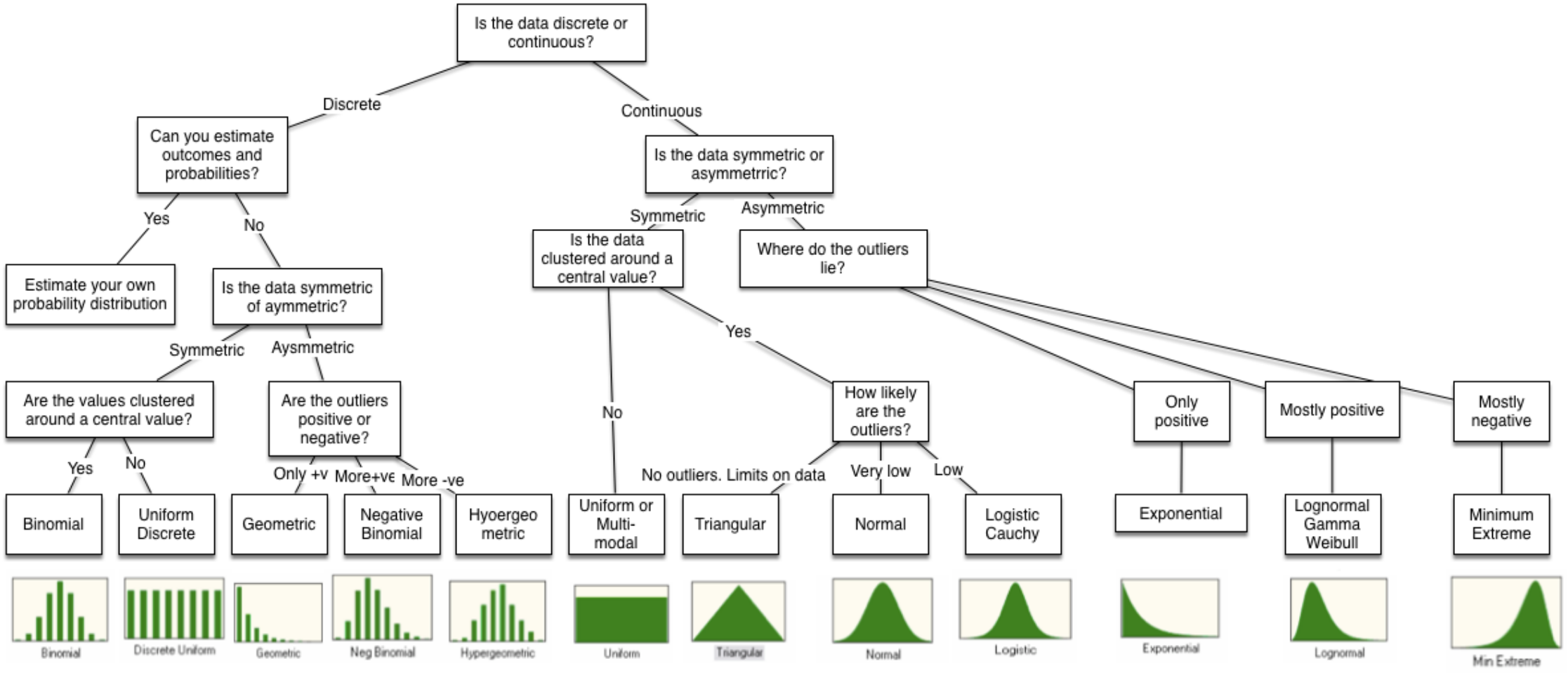
Course point of view: Data Pipeline



Basic Data Types in Statistics



Data and Distributions



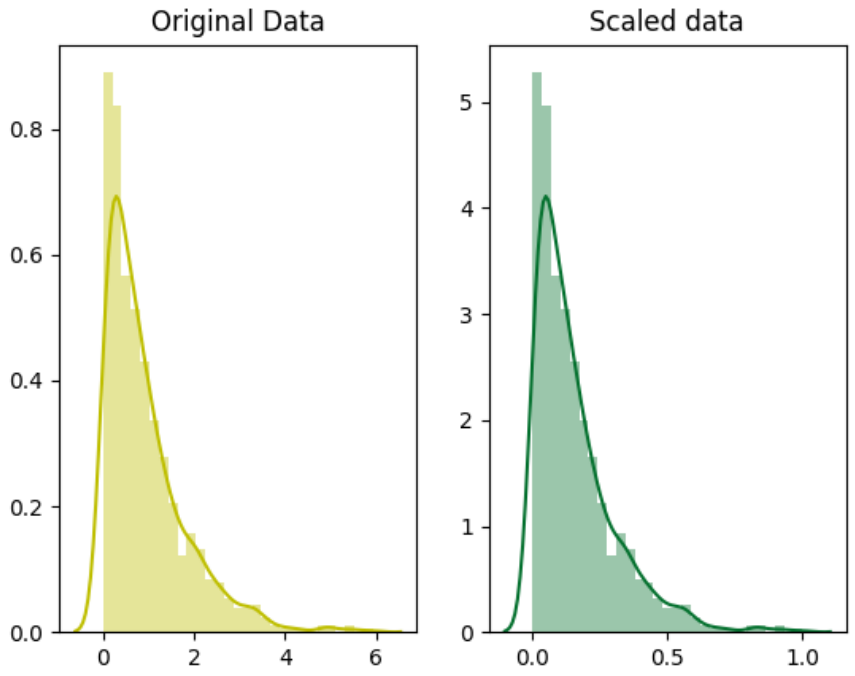
Summary Statistics

Provides:	Nominal	Ordinal	Interval	Ratio
The "order" of values is known		✔	✔	✔
"Counts," aka "Frequency of Distribution"	✔	✔	✔	✔
Mode	✔	✔	✔	✔
Median		✔	✔	✔
Mean			✔	✔
Can quantify the difference between each value			✔	✔
Can add or subtract values			✔	✔
Can multiple and divide values				✔
Has "true zero"				✔

Scaling

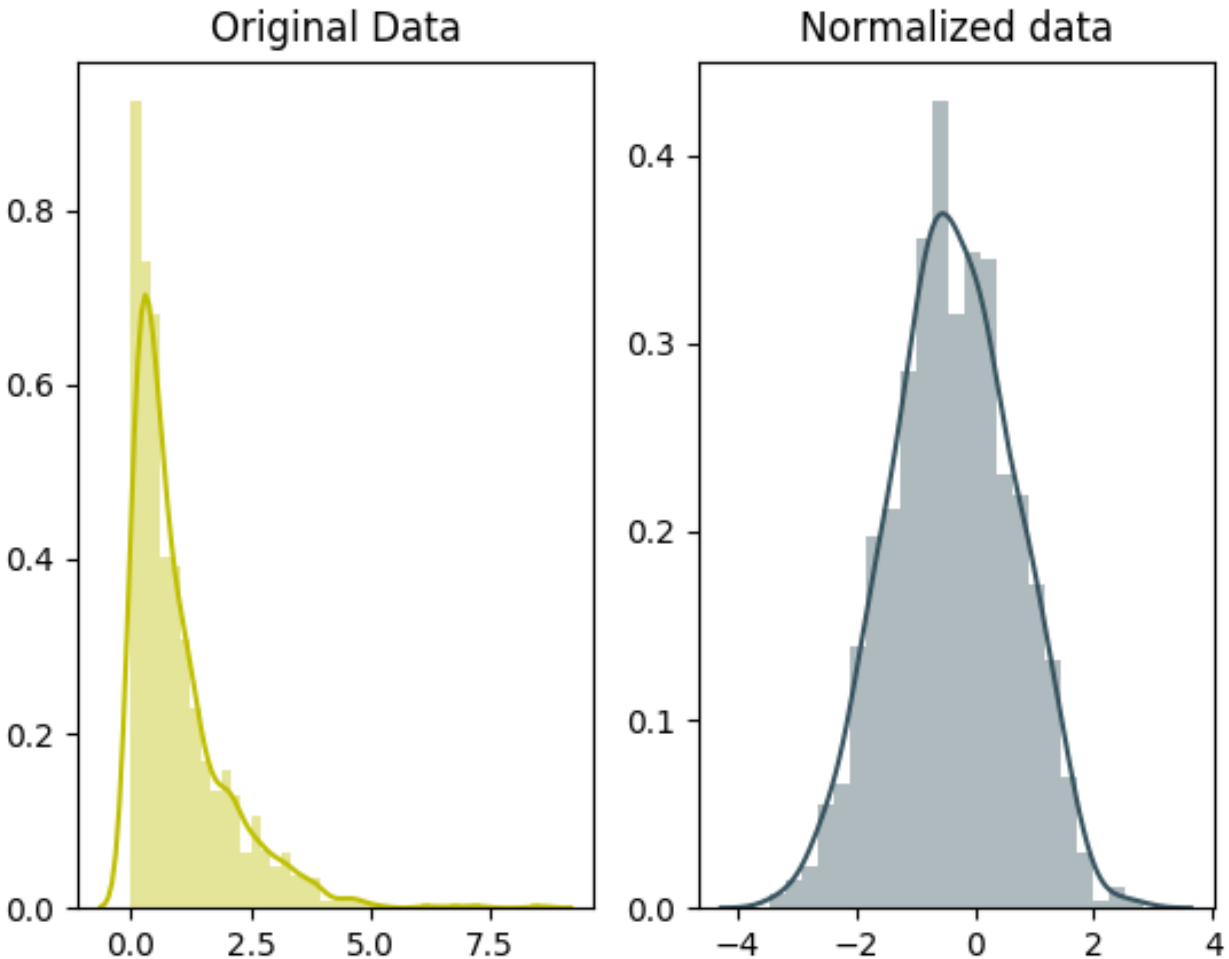
In scaling *(also called **min-max scaling**)*, you transform the data such that the features are within a specific range e.g. [0, 1].

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$



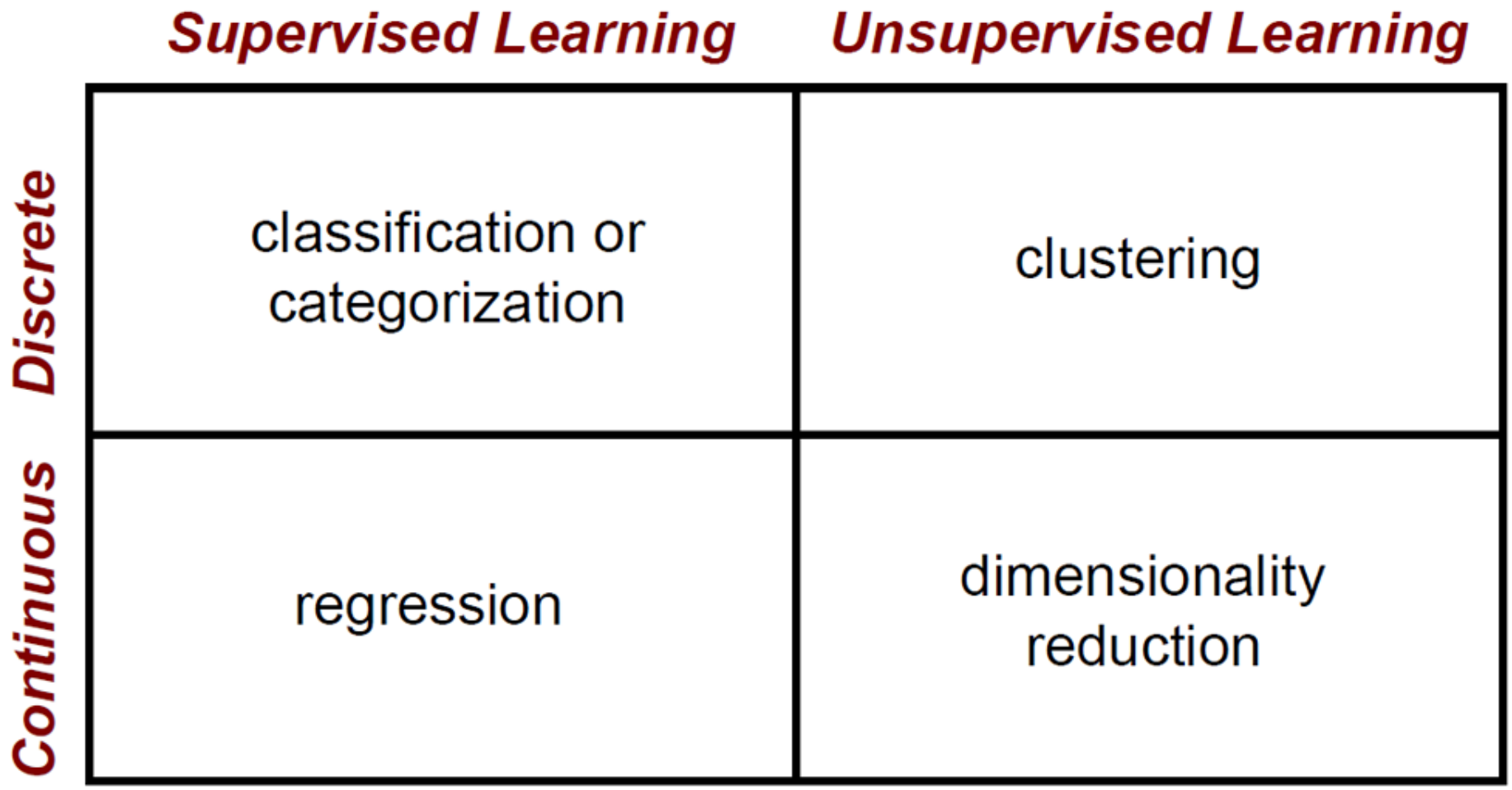
Good practice: always scale continuous data

Standardization



$$x' = \frac{x - x_{mean}}{\sigma}$$

Machine Learning Problems



Machine Learning

- **Supervised:** We are given input samples (X) and output samples (y) of a function y = f(X). We would like to "learn" f, and evaluate it on new data. Types:
  - **Classification:** y is discrete (class labels).
  - **Regression:** y is continuous, e.g. linear regression.

- **Unsupervised:** Given only samples X of the data, we compute a function f such that y = f(X) is "simpler".
  - **Clustering:** y is discrete
  - Y is continuous: **Matrix factorization, Kalman filtering, unsupervised neural networks**.

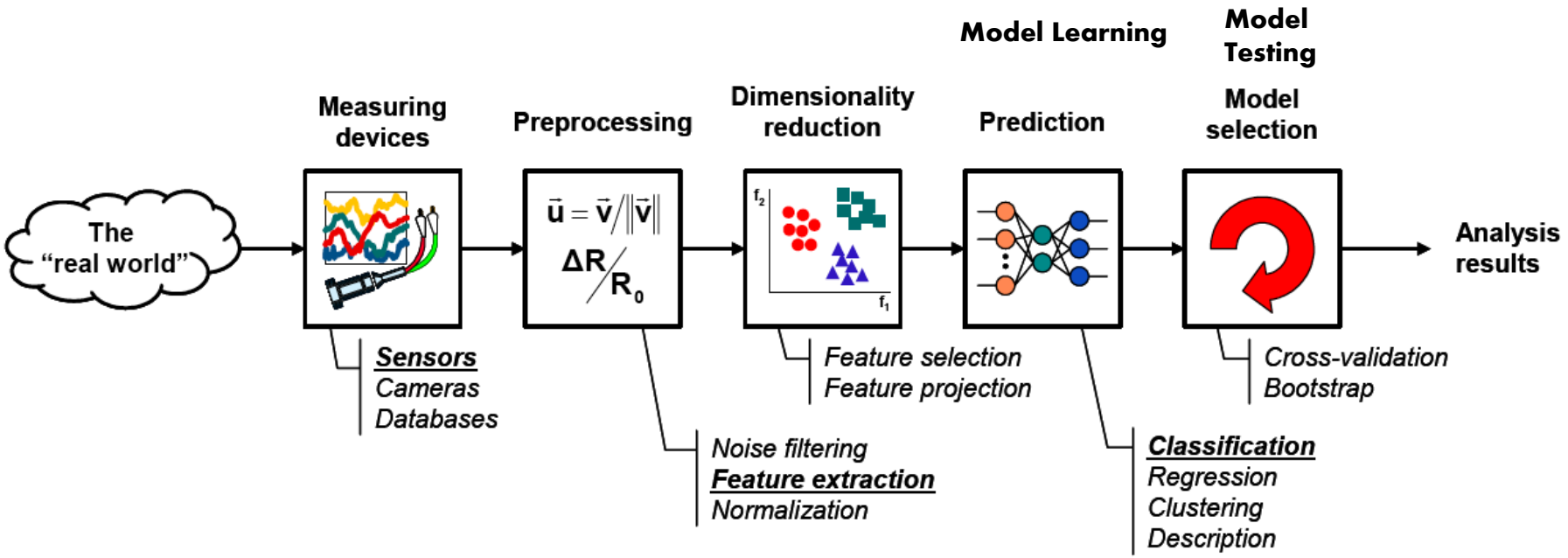
Supervised learning vs. unsupervised learning

- Supervised learning: discover patterns in the data that relate data attributes with a target (class) attribute.
  - These patterns are then utilized to predict the values of the target attribute in future data instances.
- Unsupervised learning: The data have no target attribute.
  - We want to explore the data to find some intrinsic structures in them.

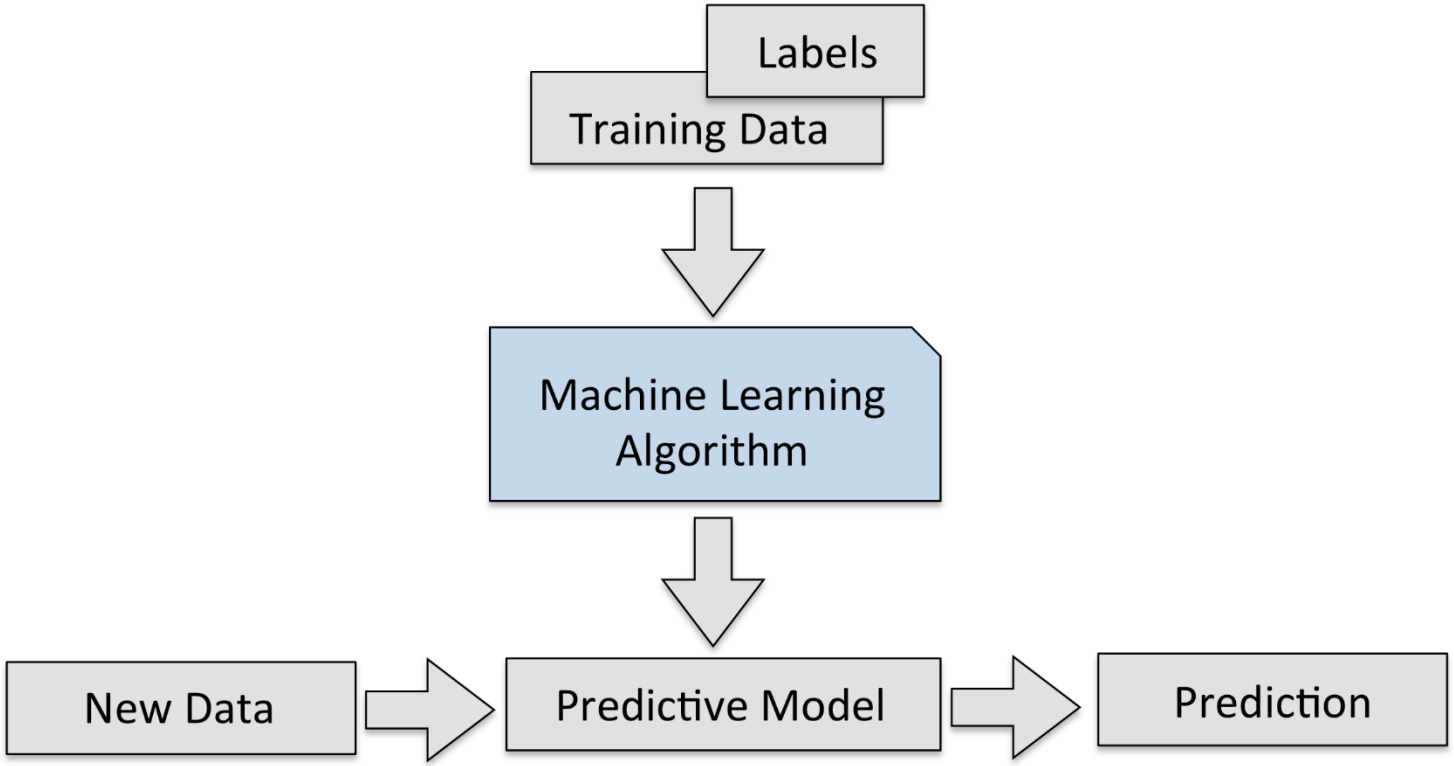
Clustering

- Clustering is a technique for finding similarity groups in data, called **clusters**. I.e.,
  - it groups data instances that are similar to (near) each other in one cluster and data instances that are very different (far away) from each other into different clusters.
- Clustering is often called an **unsupervised learning** task as no class values denoting an *a priori* grouping of the data instances are given, which is the case in supervised learning.
- Due to historical reasons, clustering is often considered synonymous with unsupervised learning.
  - In fact, association rule mining is also unsupervised
- This lecture focuses on clustering.

The Learning Process



Making predictions about the future with supervised learning



Predicting from Samples

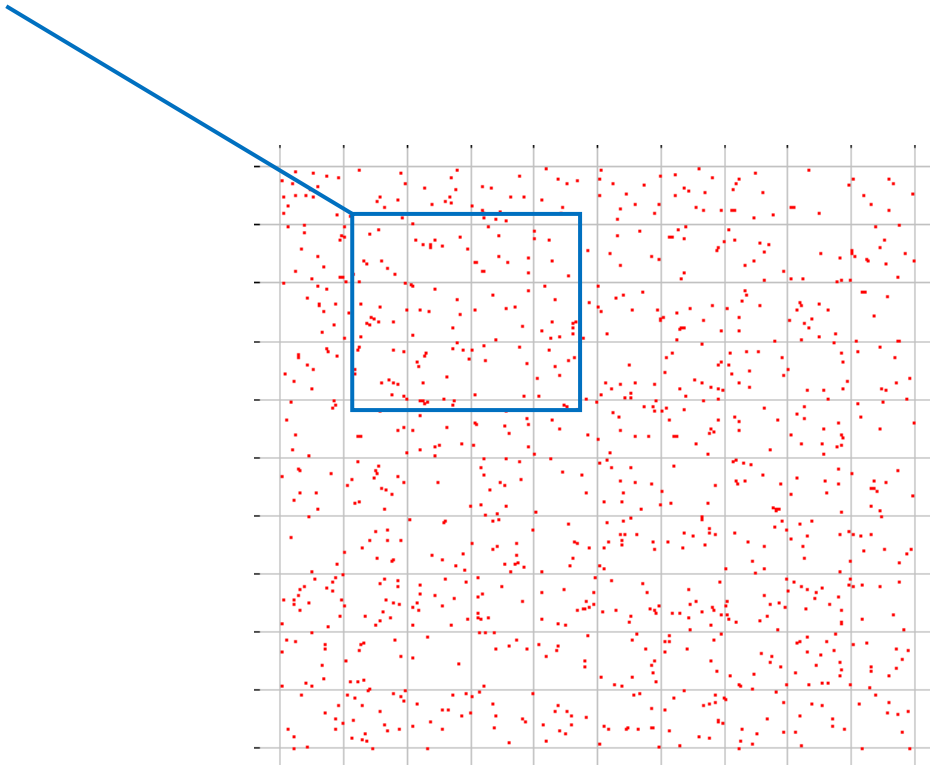
- Most datasets are **samples** from an **infinite population**.
- We are most interested in **models of the population**, but we have access only to a **sample** of it.

For datasets consisting of (X,y)

- features X + label y

a model is a prediction y = f(X)

We train on a training sample D
and we denote the model as $f_D(X)$



Bias and Variance Tradeoff

There is usually a bias-variance tradeoff caused by model complexity.

Complex models (many parameters) usually have lower bias, but higher variance.

Simple models (few parameters) have higher bias, but lower variance.

Bias and Variance Tradeoff

The total expected error is

$$Bias^2 + Variance$$

Because of the bias-variance trade-off, we want to **balance** these two contributions.

If $Variance$ strongly dominates, it means there is too much variation between models. This is called **over-fitting**.

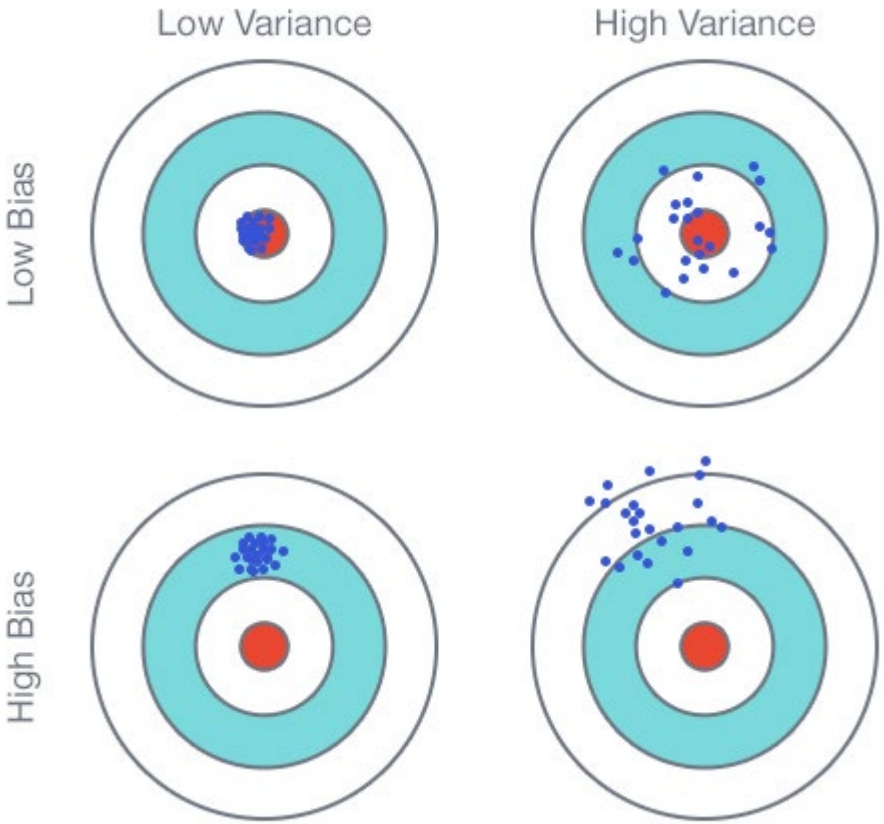
If $Bias$ strongly dominates, then the models are not fitting the data well enough. This is called **under-fitting**.

Bias and Variance

Bias and variance can be visualized with a classic example of a dartboard. We have four different dart throwers, each with different combinations of low/high bias and low/high variance. We represent the locations of each of their dart throws as blue dots:

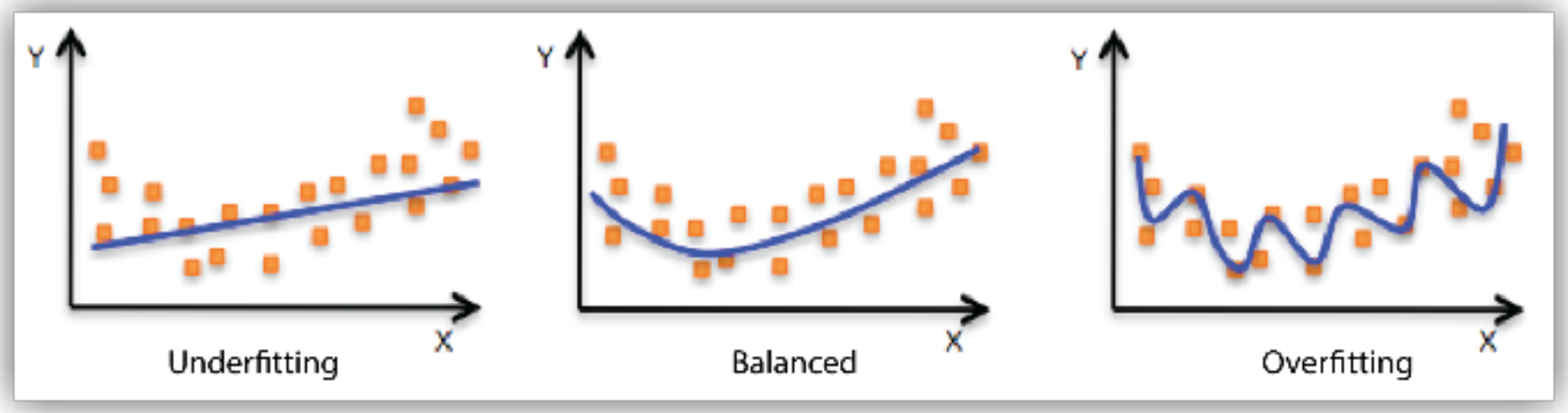
The low bias players' darts tend to be centered around the center of the dart board, while the high bias players' darts are centered in a different location. In this case, their darts are "biased" toward the top of the dart board.

The high variance dart throwers have their darts more spread out; they are less able to successfully place the dart where they're aiming (in the case of the biased player, they are aiming at the incorrect location).

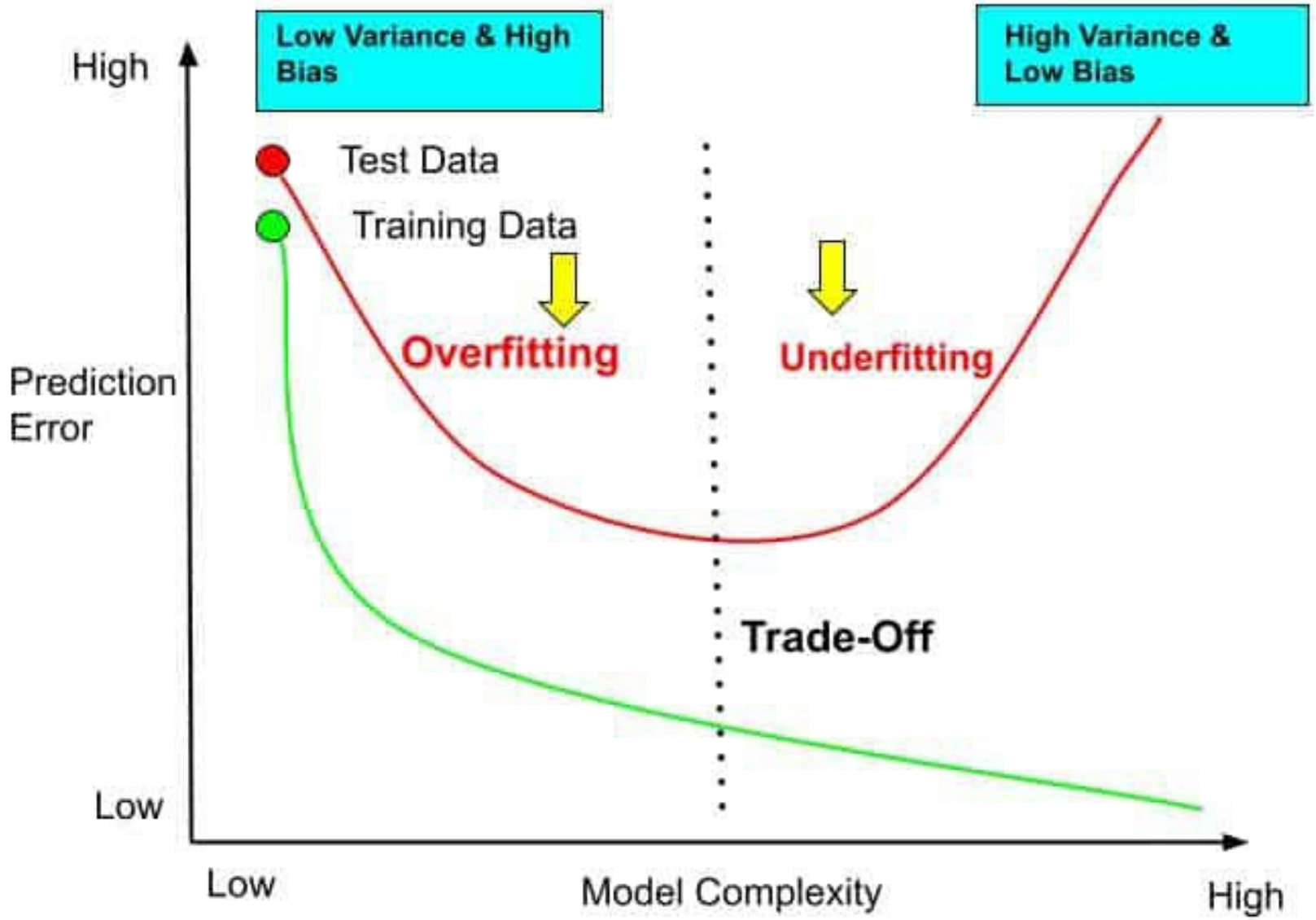


Underfitting vs. Overfitting

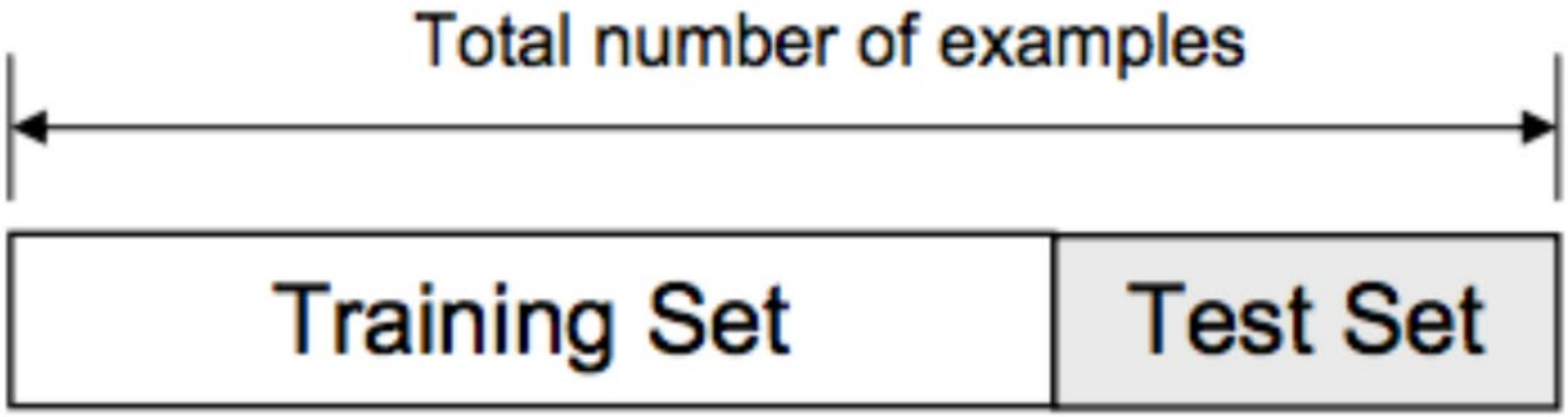
Understanding model fit is important for understanding the root cause for poor model accuracy. This understanding will guide you to take corrective steps. We can determine whether a predictive model is underfitting or overfitting the training data by looking at the prediction error on the training data and the evaluation data.



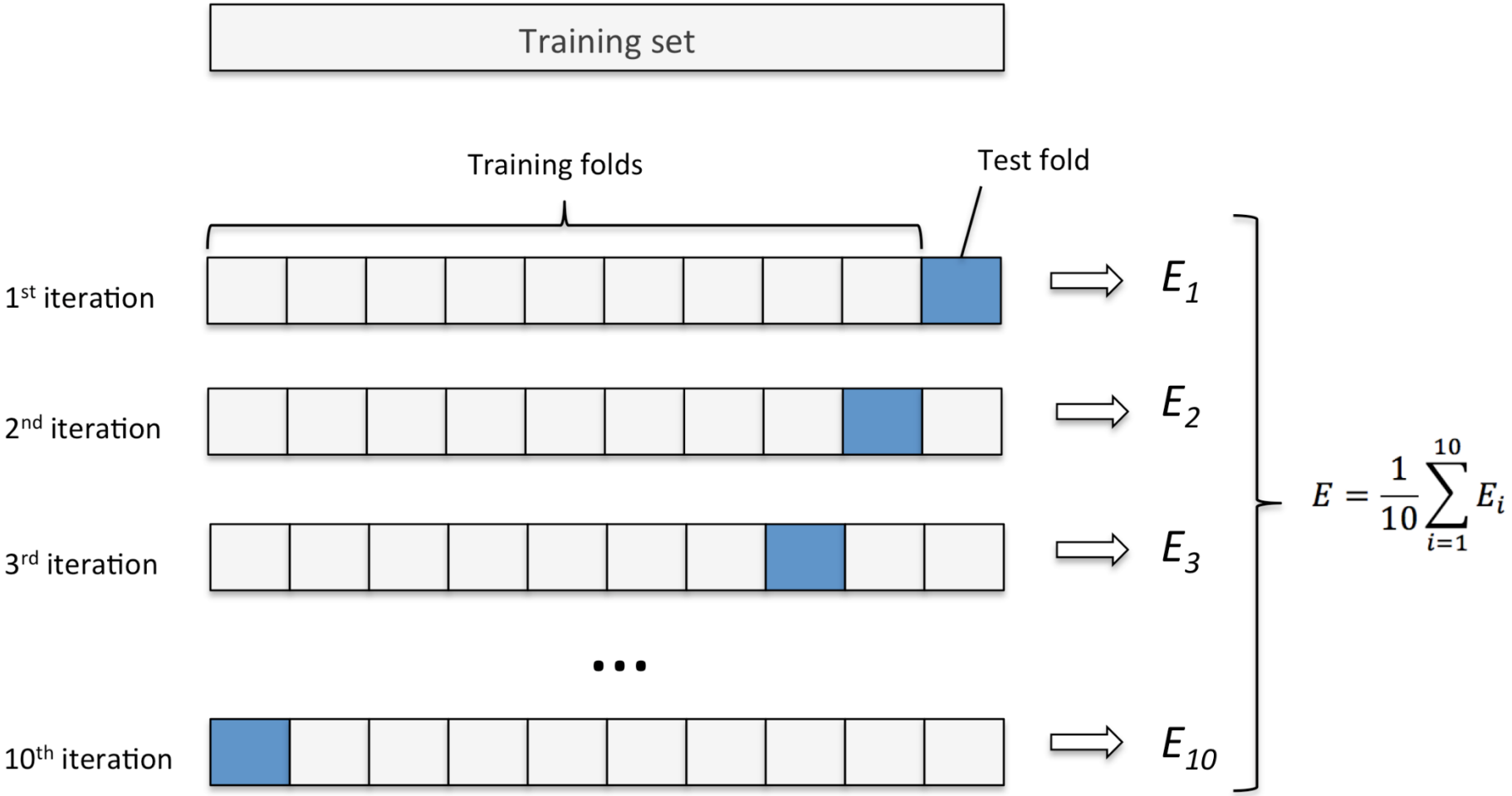
Your model is underfitting the training data when the model performs poorly on the training data. This is because the model is unable to capture the relationship between the input examples (often called X) and the target values (often called Y). Your model is overfitting your training data when you see that the model performs well on the training data but does not perform well on the evaluation data. This is because the model is memorizing the data it has seen and is unable to generalize to unseen examples.


Low Variance & High Bias
High Variance & Low Bias
High
Test Data
Training Data
Overfitting
Underfitting
Prediction Error
Trade-Off
Low
Low
Model Complexity
High

Training-Test Split



K-fold cross-validation



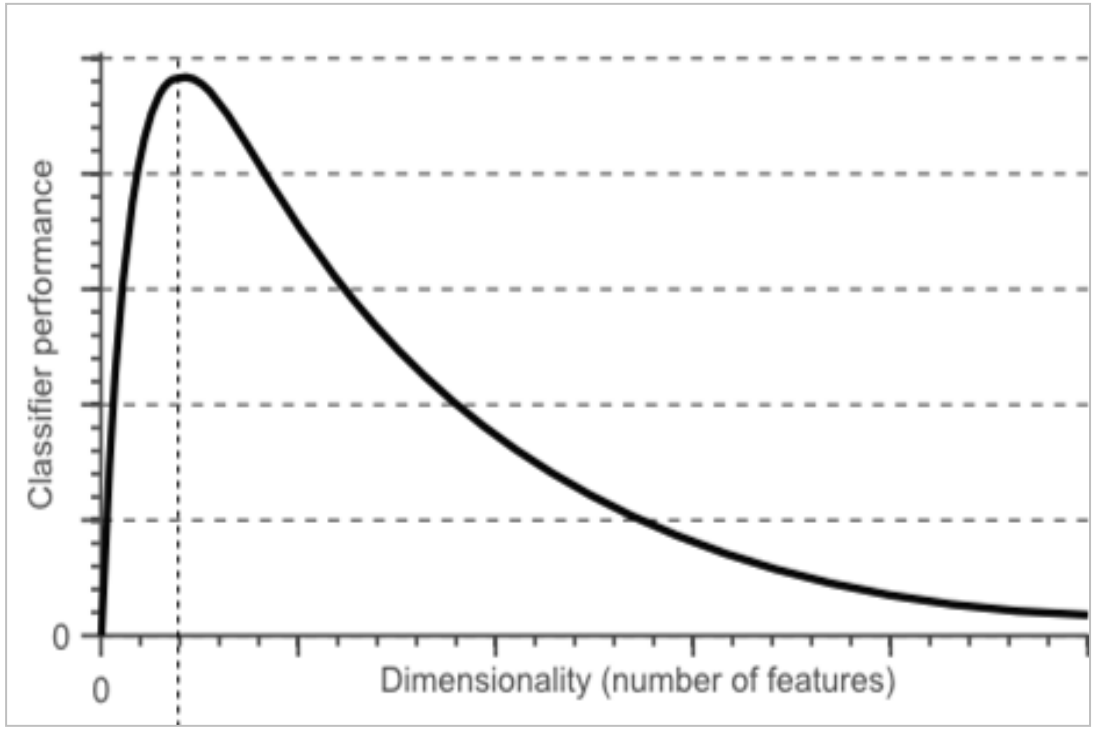
Huge number of dimensions causes problems

Data becomes very sparse, some algorithms become meaningless (e.g. density based clustering)

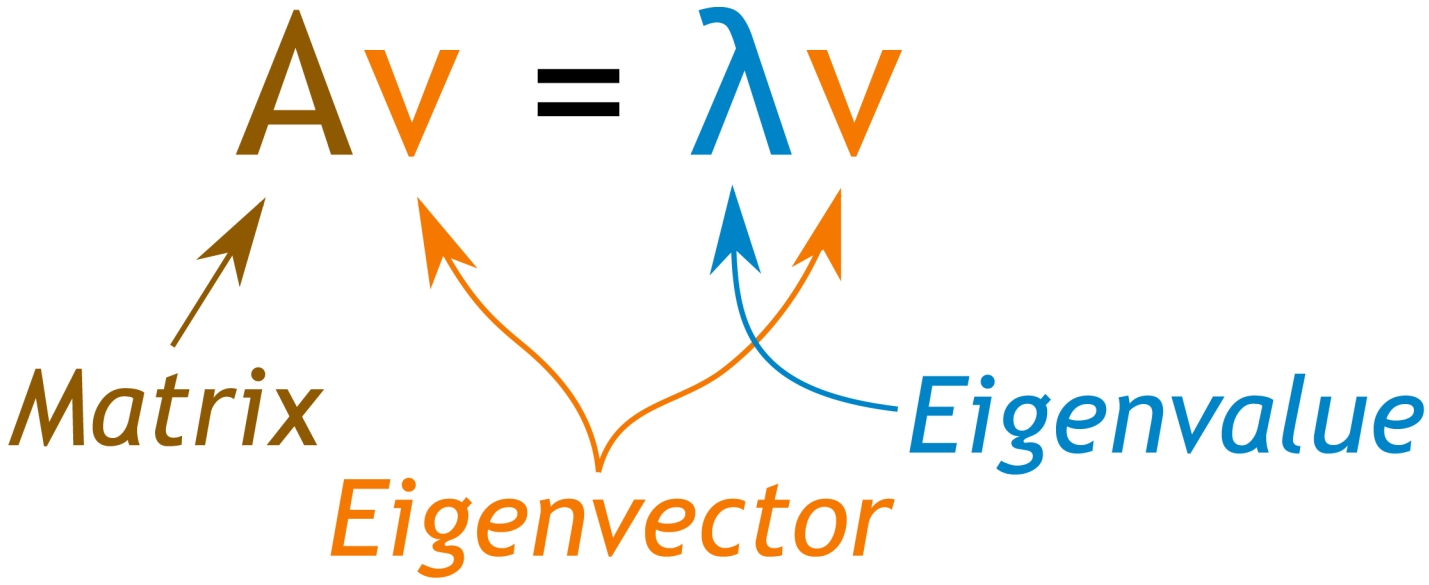
The complexity of several algorithms depends on the dimensionality and they become infeasible.

Curse of Dimensionality

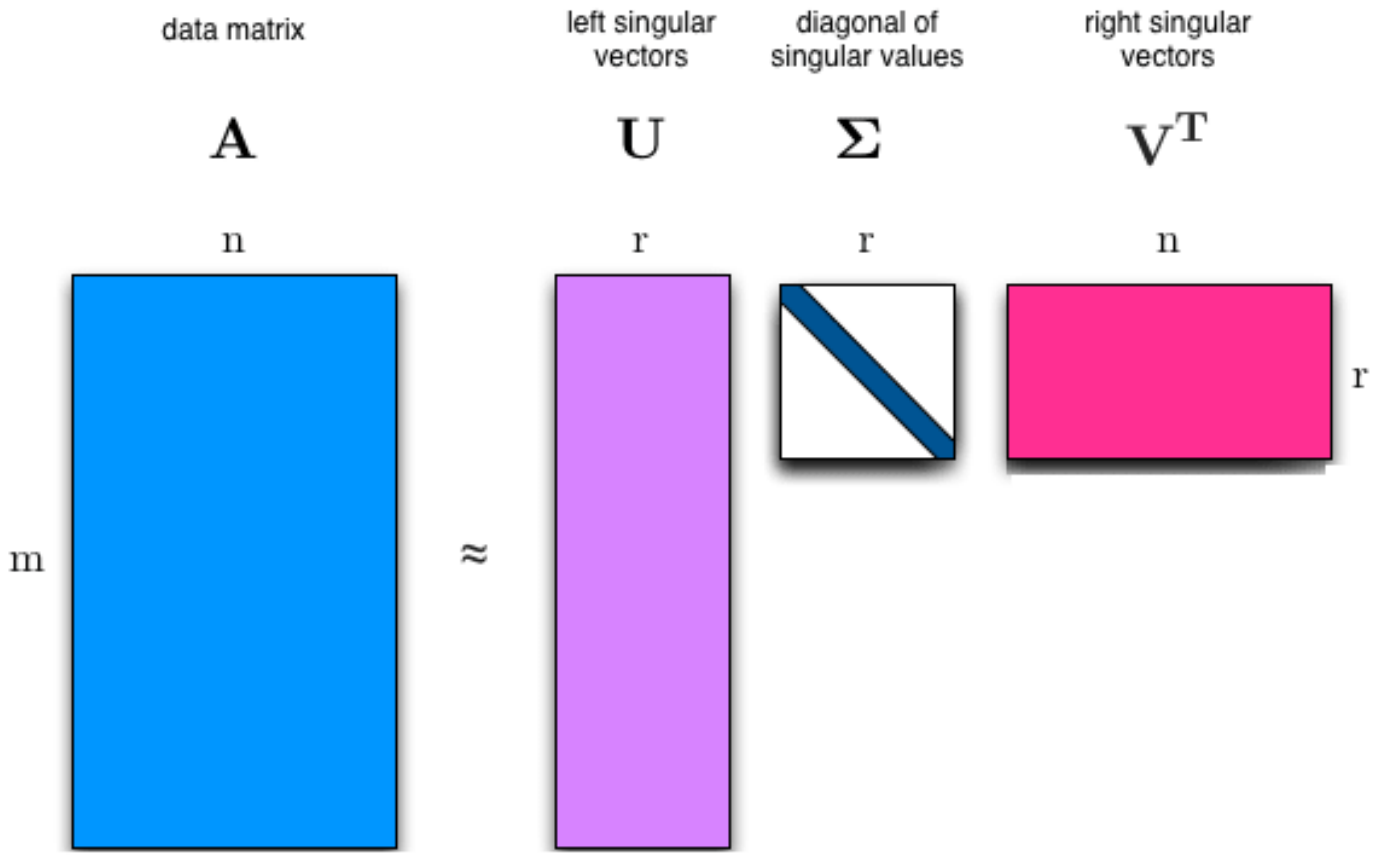
Machine learning algorithms tend to over-fit data when the sample has a lot of predictor variables. There is a number of features above which the performance of a ML will degrade rather than improve.



Eigenvalues and Eigenvectors

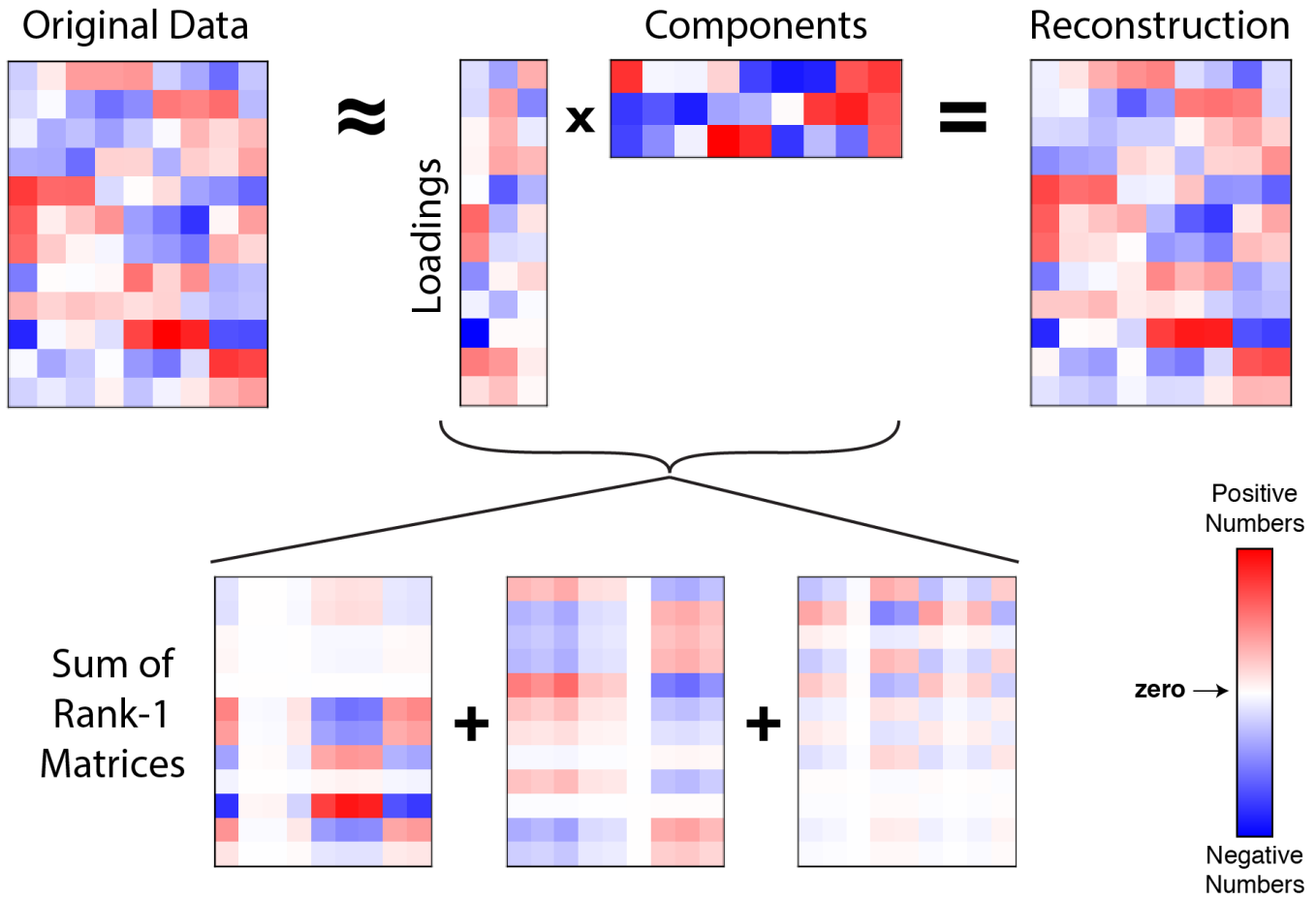


Singular Value Decomposition



The singular value decomposition (SVD) is a factorization of a real or complex matrix that generalizes the eigendecomposition of a square normal matrix to any m×n matrix.

SVD/PCA and Rank-k approximations



Milestones of deep learning

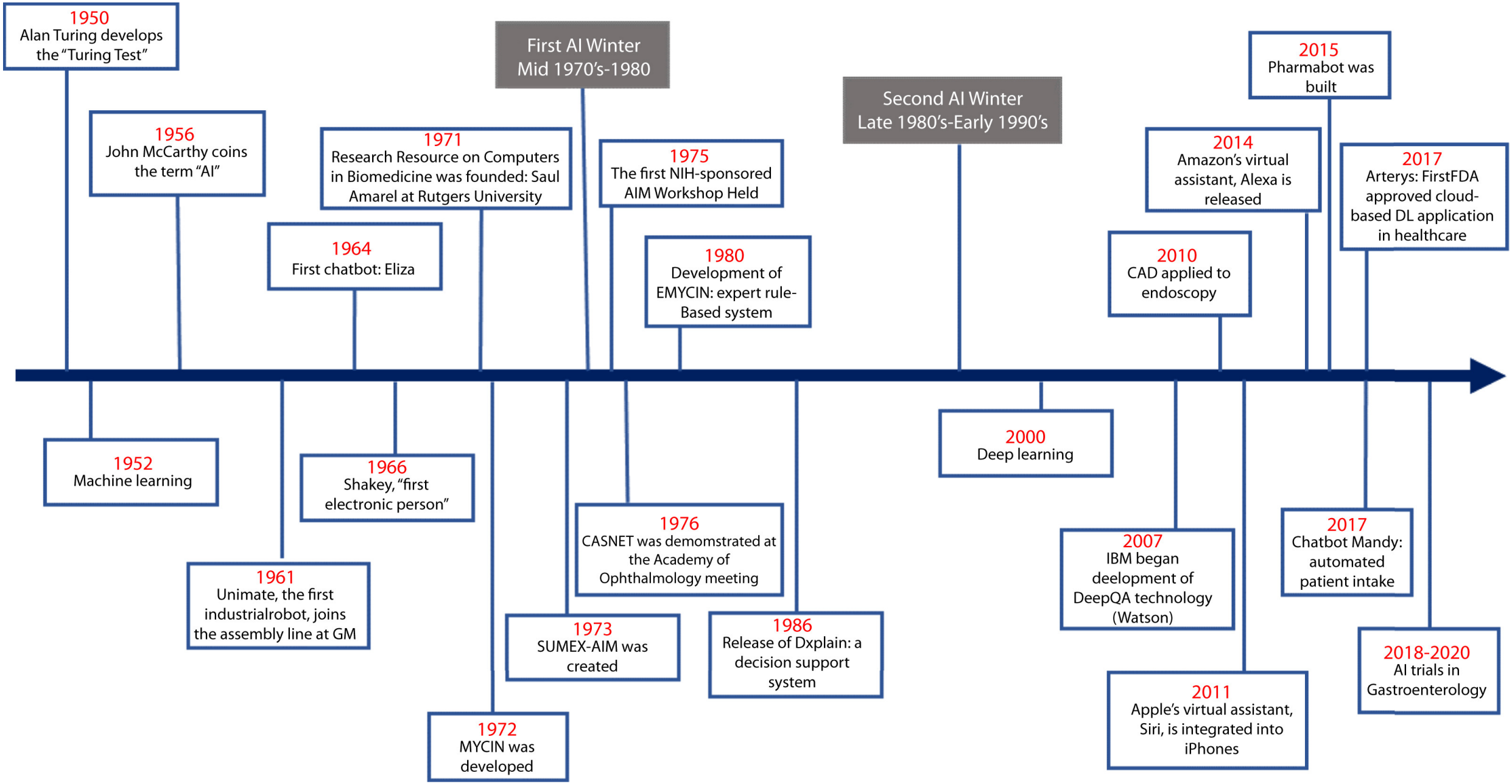
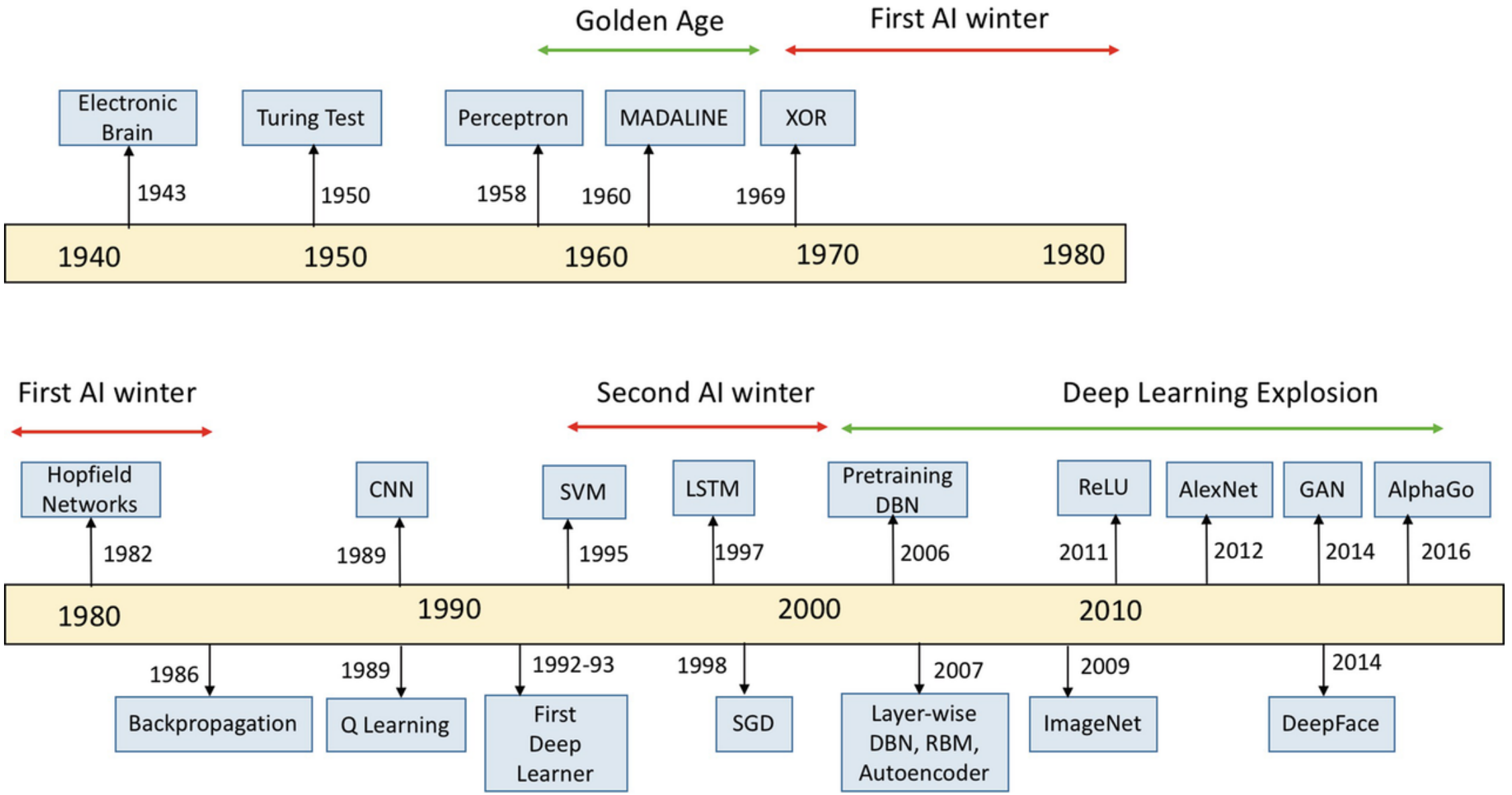
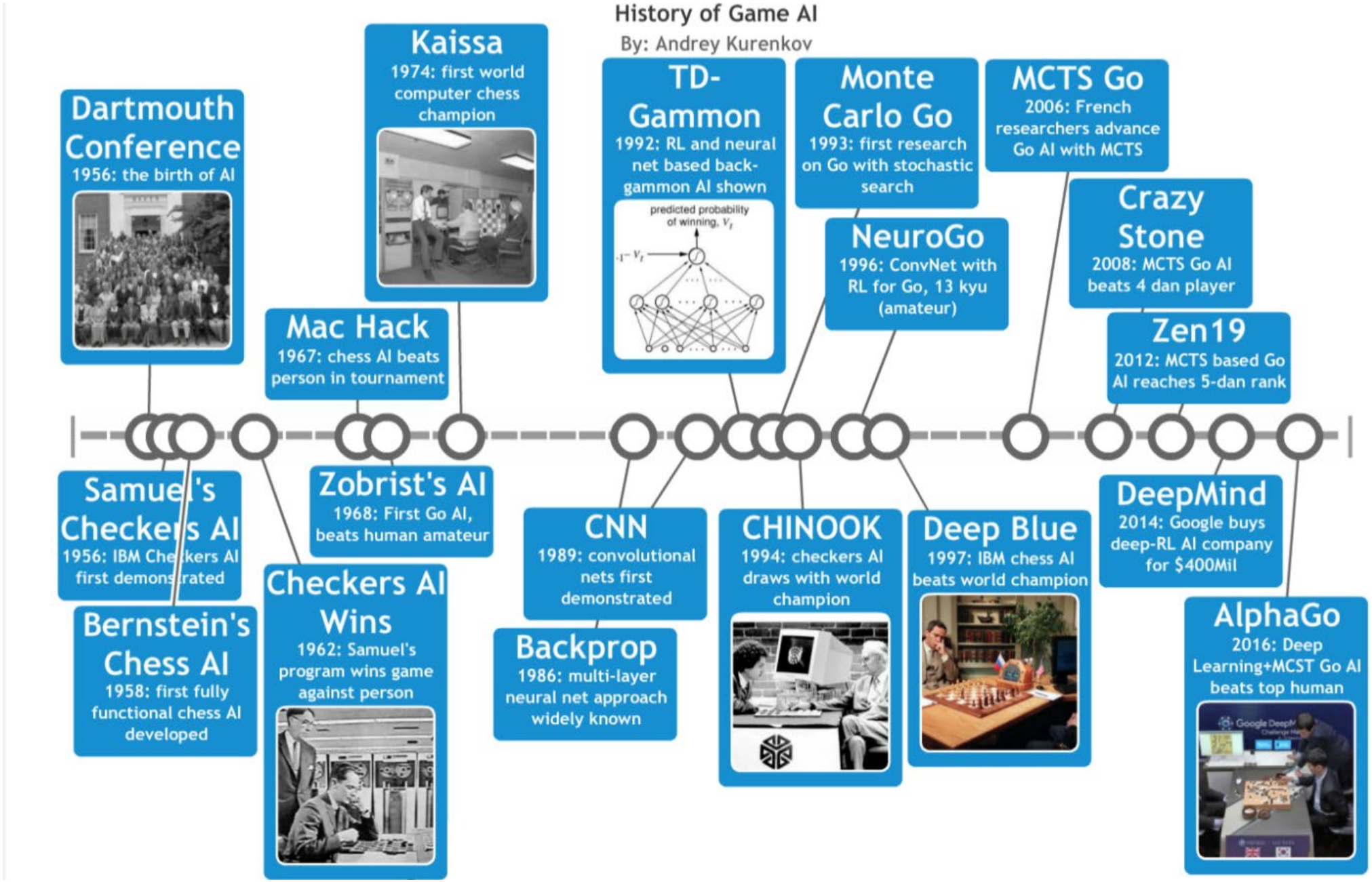


Image credit: https://doi.org/10.1016/j.gie.2020.06.040

Highlights in deep learning research



From: Deep Learning for NLP and Speech Recognition



[A 'Brief' History of Game AI Up To AlphaGo]

A few historical milestones

1958 New York Times…

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

The embryo—the Weather Bureau's $2,000,000 "704" computer—learned to differentiate between right and left after fifty attempts in the Navy's demonstration for newsmen.

The service said it would use this principle to build the first of its Perceptron thinking machines that will be able to read and write. It is expected to be finished in about a year at a cost of $100,000.

Dr. Frank Rosenblatt, designer of the Perceptron, conducted the demonstration. He said the machine would be the first device to think as the human brain. As do human beings, Perceptron will make mistakes at first, but will grow wiser as it gains experience, he said.

Dr. Rosenblatt, a research psychologist at the Cornell Aeronautical Laboratory, Buffalo, said Perceptrons might be fired to the planets as mechanical space explorers.

Without Human Controls

The Navy said the perceptron would be the first non-living mechanism "capable of receiving, recognizing and identifying its surroundings without any human training or control."

The "brain" is designed to remember images and information it has perceived itself. Ordinary computers remember only what is fed into them on punch cards or magnetic tape.

Later Perceptrons will be able to recognize people and call out their names and instantly translate speech in one language to speech or writing in another language, it was predicted.

Mr. Rosenblatt said in principle it would be possible to build brains that could reproduce themselves on an assembly line and which would be conscious of their existence.

In today's demonstration, the "704" was fed two cards, one with squares marked on the left side and the other with squares on the right side.

Learns by Doing

In the first fifty trials, the machine made no distinction between them. It then started registering a "Q" for the left squares and "O" for the right squares.

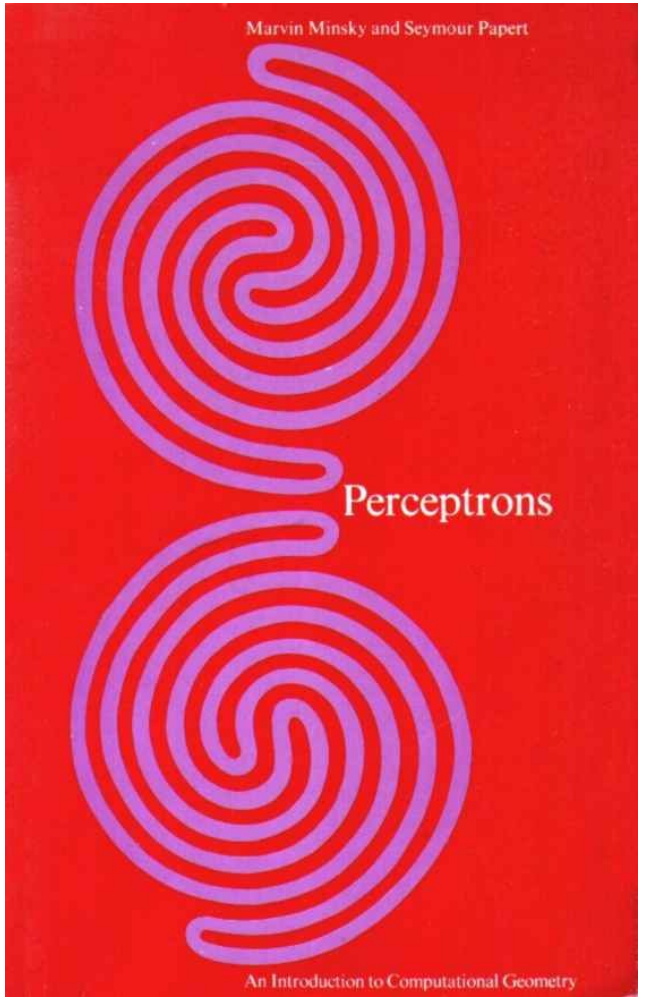
Dr. Rosenblatt said he could explain why the machine learned only in highly technical terms. But he said the computer had undergone a "self-induced change in the wiring diagram."

The first Perceptron will have about 1,000 electronic "association cells" receiving electrical impulses from an eye-like scanning device with 400 photo-cells. The human brain has 10,000,000,000 responsive cells, including 100,000,000 connections with the eyes.

Frank Rosenblatt (1928-1971)

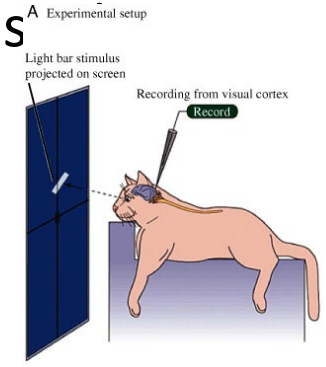
A few historical milestones

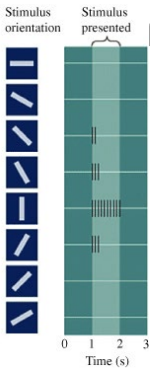
- 1958: Rosenblatt's perceptron
- 1969: Minsky and Papert Perceptrons book
  - Fascinating reading: M. Olazaran, A Sociological Study of the Official History of the Perceptrons Controversy, *Social Studies of Science*, 1996



A few historical milestones

- 1958: Rosenblatt's perceptron
- 1969: Minsky and Papert Perceptrons book
- 1980: Fukushima's Neocognitron
  - Video (short version)
  - Inspired by the findings of Hubel & Wiesel about the hierarchical organization





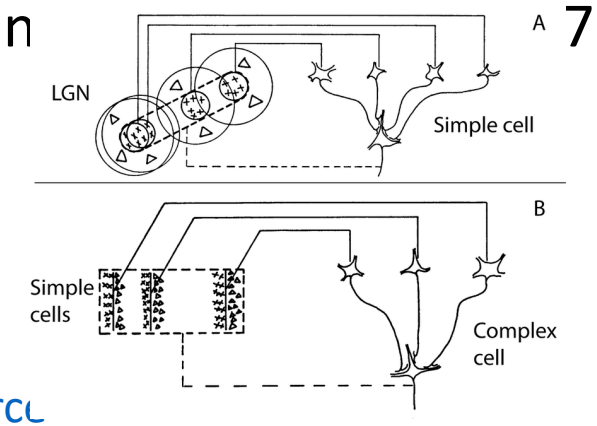
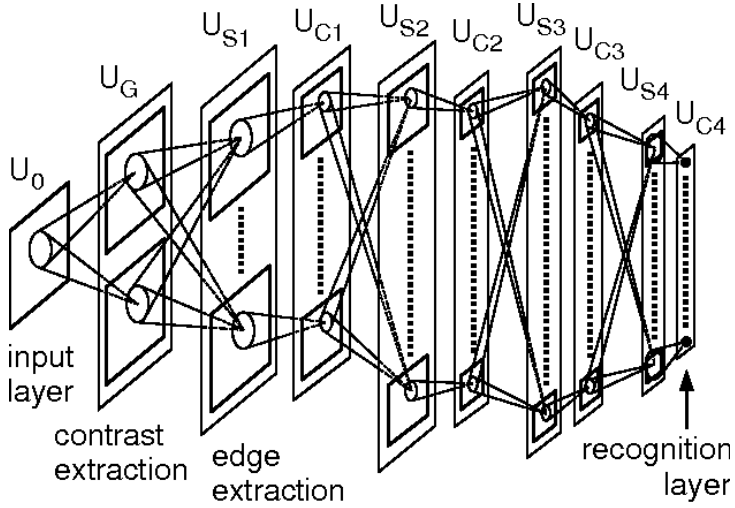


Image source

Kunihiko Fukushima



A few historical milestones

- 1958: Rosenblatt's perceptron
- 1969: Minsky and Papert Perceptrons book
- 1980: Fukushima's Neocognitron
- 1986: Back-propagation
  - Origins in control theory and optimization: Kelley (1960), Dreyfus (1962), Bryson & Ho (1969), Linnainmaa (1970)
  - Application to neural networks: Werbos (1974)
  - Popularized by Rumelhart, Hinton & Williams (1986)

A few historical milestones

- 1958: Rosenblatt's perceptron
- 1969: Minsky and Papert Perceptrons book
- 1980: Fukushima's Neocognitron
- 1986: Back-propagation
- 1989 – 1998: Convolutional neural networks
  - LeNet to LeNet-5

Yann LeCun
2018 ACM Turing Award winner
(with Hinton and Bengio)

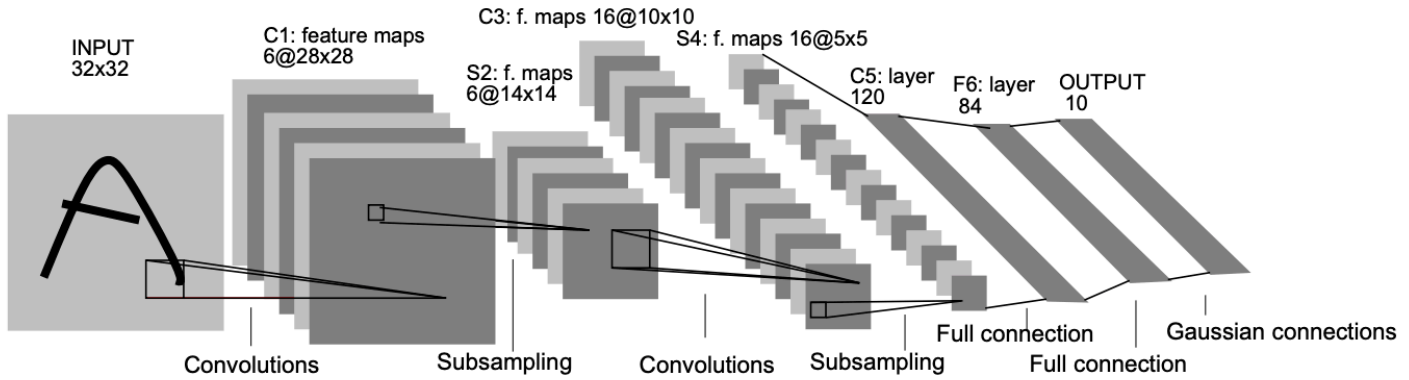


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

A few historical milestones

- 1958: Rosenblatt's perceptron
- 1969: Minsky and Papert Perceptrons book
- 1980: Fukushima's Neocognitron
- 1986: Back-propagation
- 1989 – 1998: Convolutional neural networks
- 2012: AlexNet

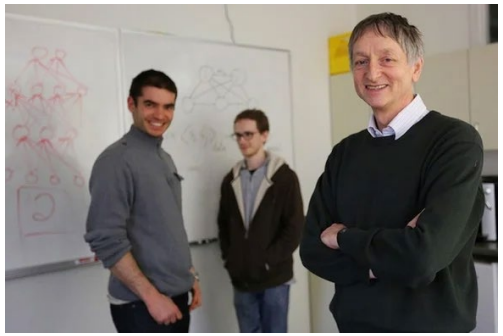
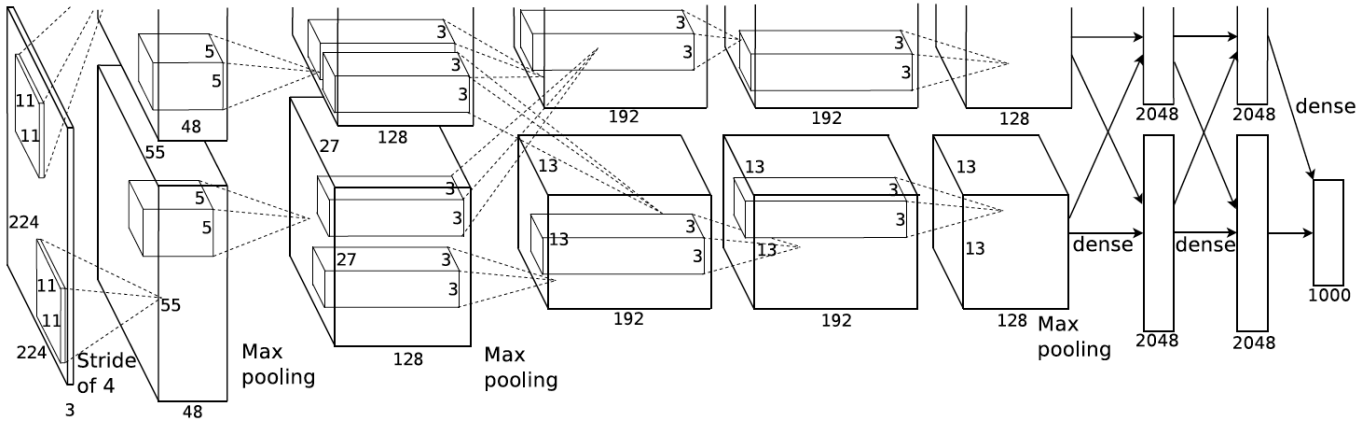
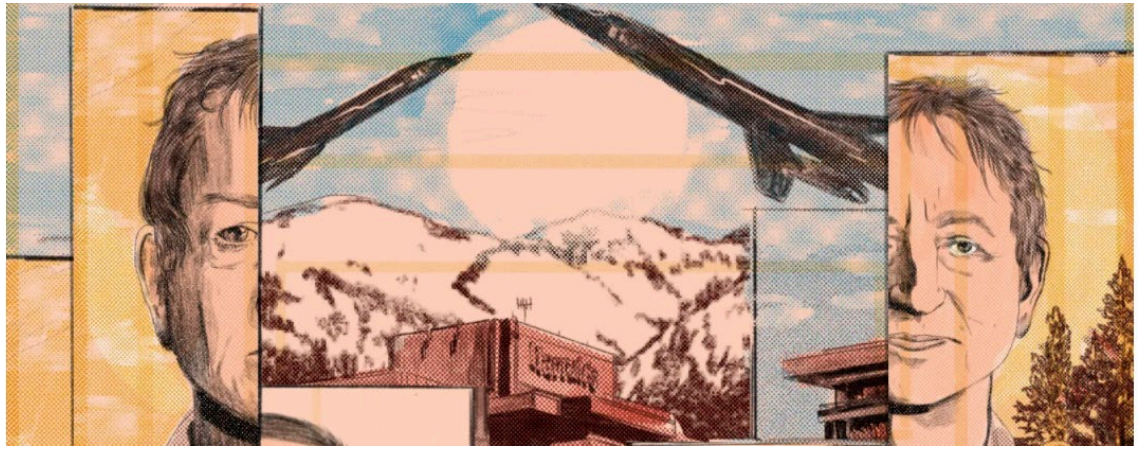


Photo source



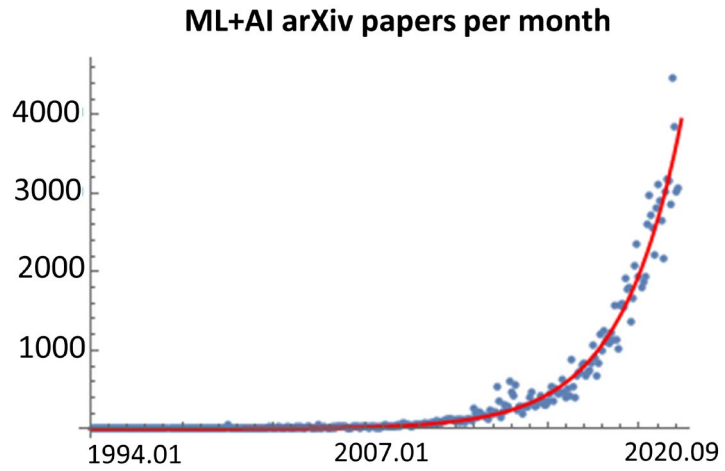
A few historical milestones

- 1958: Rosenblatt's perceptron
- 1969: Minsky and Papert Perceptrons book
- 1980: Fukushima's Neocognitron
- 1986: Back-propagation
- 1989 – 1998: Convolutional neural networks
- 2012: AlexNet
  - Fascinating reading: The secret auction that set off the race for AI supremacy, Wired, 3/16/2021



A few historical milestones

- 1958: Rosenblatt's perceptron
- 1969: Minsky and Papert Perceptrons book
- 1980: Fukushima's Neocognitron
- 1986: Back-propagation
- 1989 – 1998: Convolutional neural networks
- 2012: AlexNet
- 2012 – present: deep learning explosion

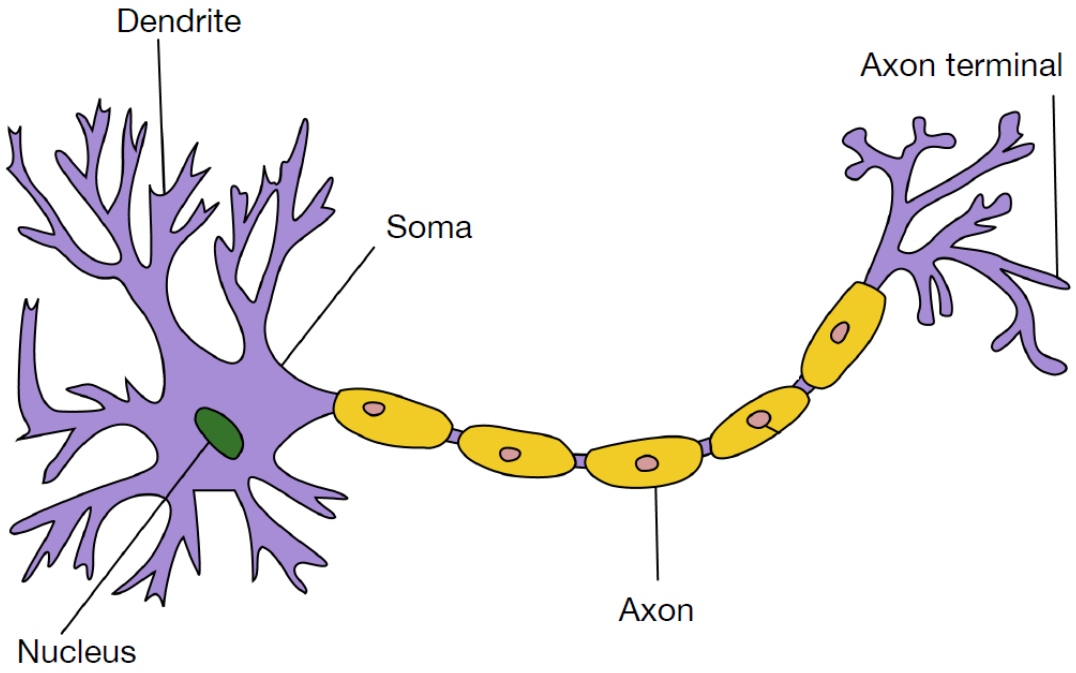


Source, via J. Johnson

Why ”neural” networks?

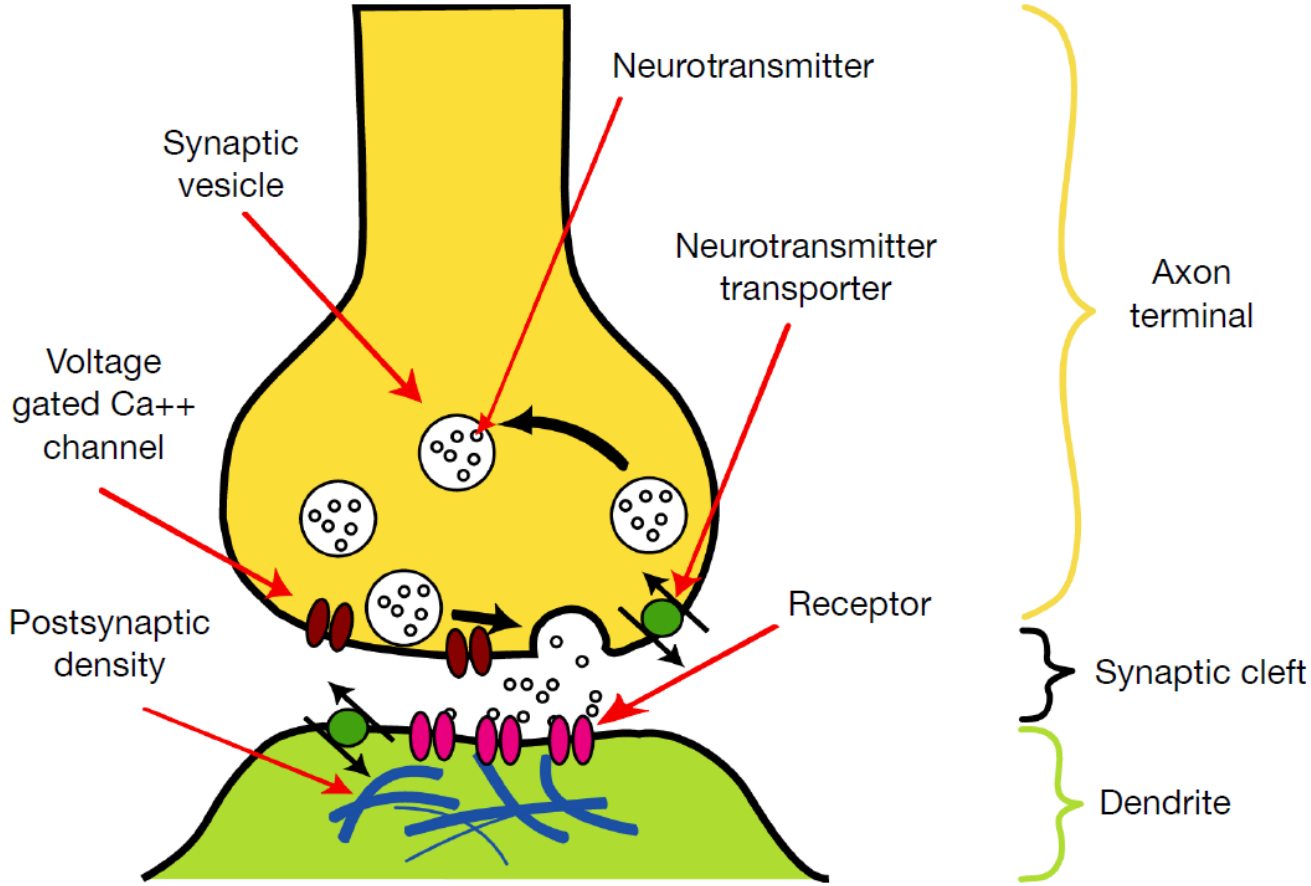
Neuron in the Brain

The human brain is made up of about 100 billion neurons

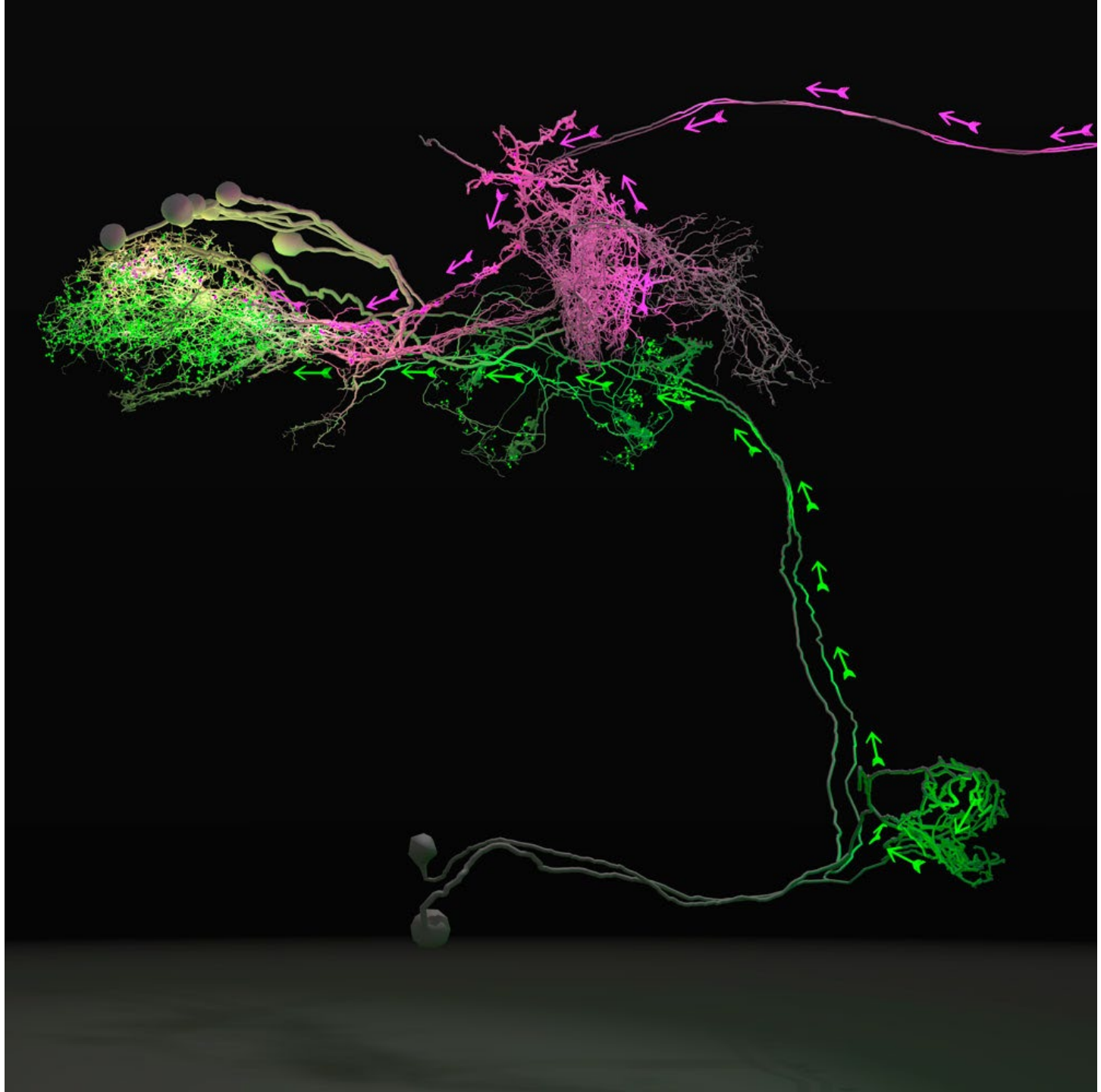


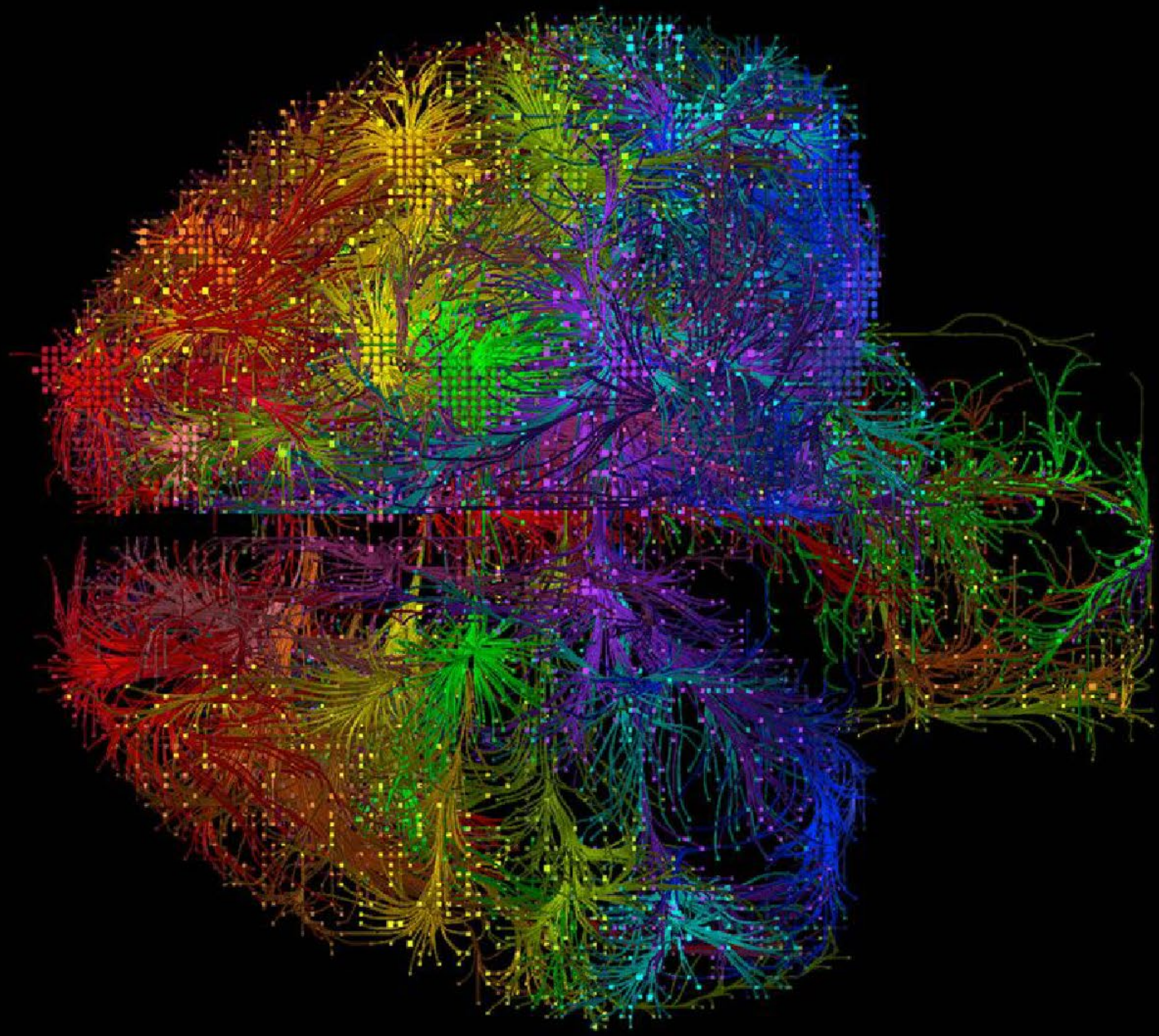
Neurons receive electric signals at the dendrites and send them to the axon

Neural Communication



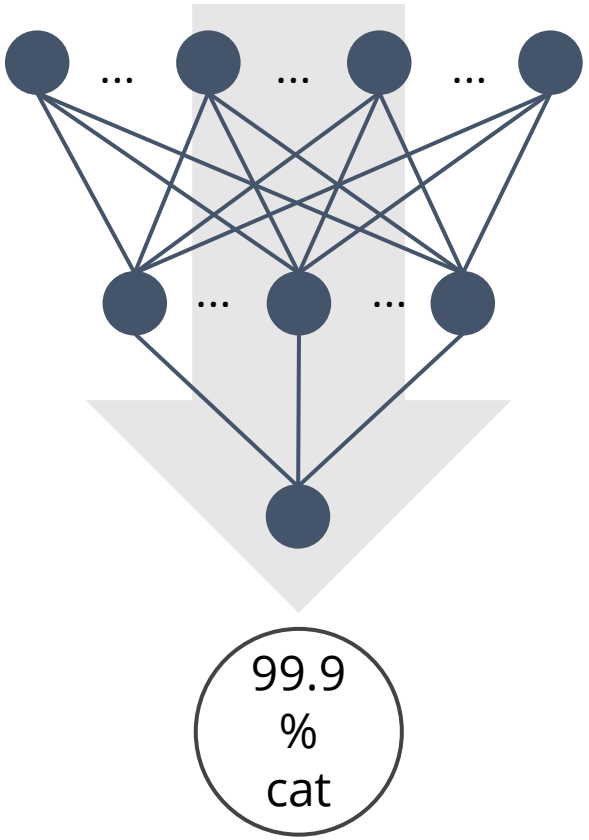
The axon of the neuron is connected to the dendrites of many other neurons





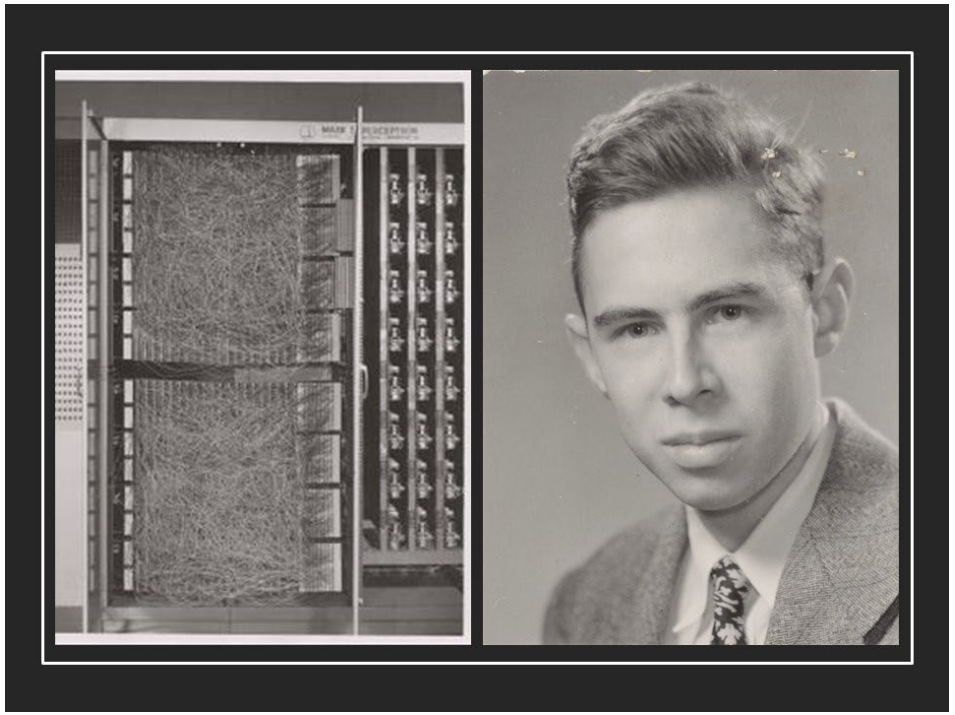
Artificial Neural Networks

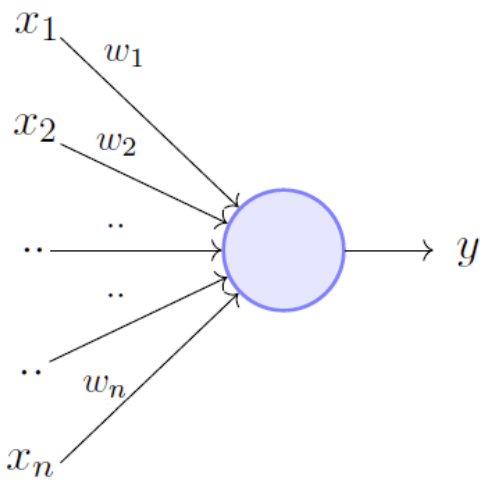
1. Machine Learning Algorithm
2. Very simplified parametric models of our brain
3. Networks of basic processing units: neurons
4. Neurons store information which has to be learned (the weights or the state for RNNs)
5. Many types of architectures to solve different tasks
6. They can scale to massive data



Perceptron

Frank Rosenblatt, 1958





$$y = 1 \quad if \sum_{i=1}^{n} w_i * x_i \geq \theta$$

$$= 0 \quad if \sum_{i=1}^{n} w_i * x_i < \theta$$

Rewriting the above,

$$y = 1 \quad if \sum_{i=1}^{n} w_i * x_i - \theta \geq 0$$

$$= 0 \quad if \sum_{i=1}^{n} w_i * x_i - \theta < 0$$

Perceptron algorithm

```
Initialize w⃗ = 0⃗                                  // Initialize w⃗. w⃗ = 0⃗ misclassifies everything.
while TRUE do                                      // Keep looping
    m = 0                                          // Count the number of misclassifications, m
    for (x_i, y_i) ∈ D do                          // Loop over each (data, label) pair in the dataset, D
        if y_i(w⃗^T · x⃗_i) ≤ 0 then                  // If the pair (x⃗_i, y_i) is misclassified
            w⃗ ← w⃗ + yx⃗                            // Update the weight vector w⃗
            m ← m + 1                              // Counter the number of misclassification
        end if
    end for
    if m = 0 then                                  // If the most recent w⃗ gave 0 misclassifications
        break                                      // Break out of the while-loop
    end if
end while                                          // Otherwise, keep looping!
```

Geometric Intuition of Perceptron

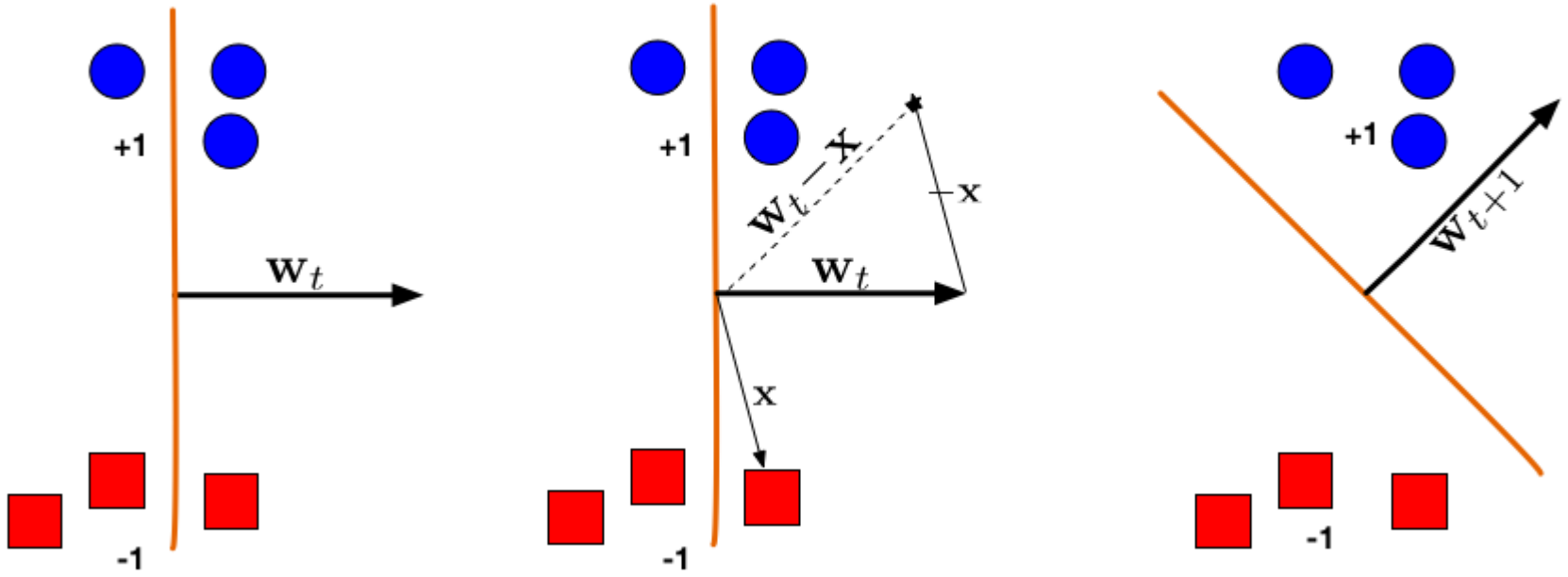
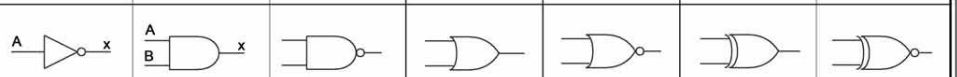
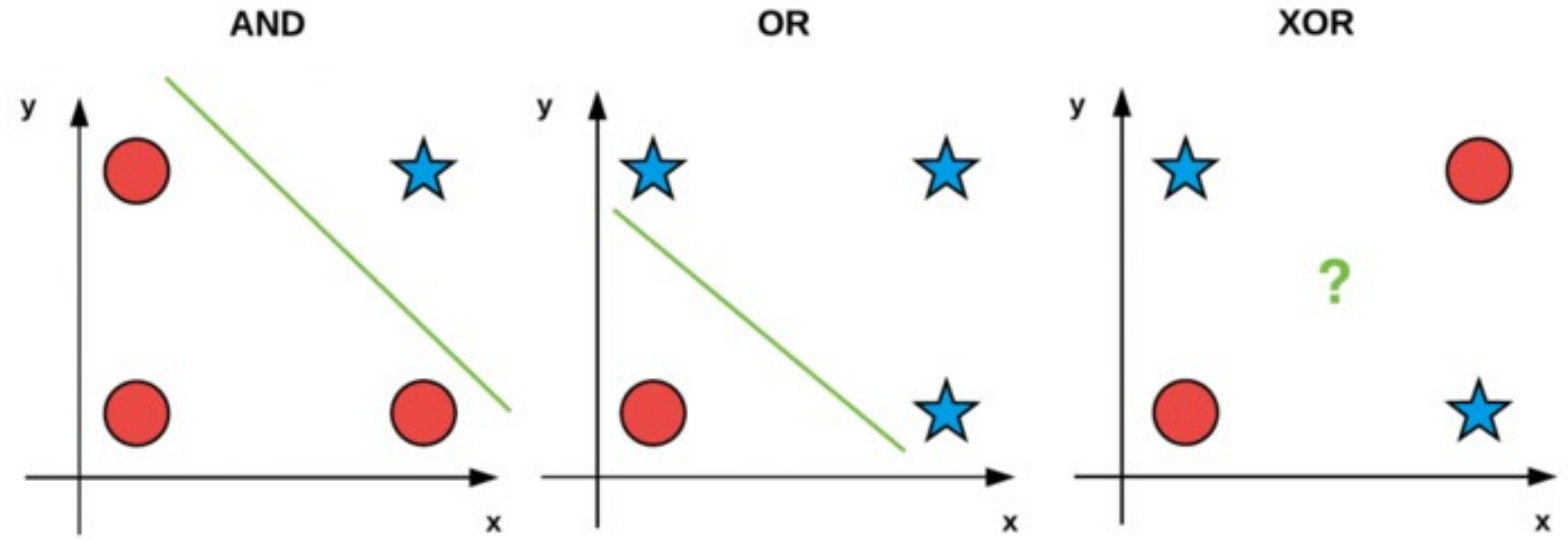


Illustration of a Perceptron update. (Left:) The hyperplane defined by w_t *misclassifies one red (-1) and one blue (+1) point. (Middle:) The red point* x *is chosen and used for an update. Because its label is -1 we need to* ***subtract*** x *from* w_t*. (Right:) The udpated hyperplane* $w_{t+1} = w_t - x$ $w_{t+1} = w_t - x$ *separates the two classes and the Perceptron algorithm has converged.*

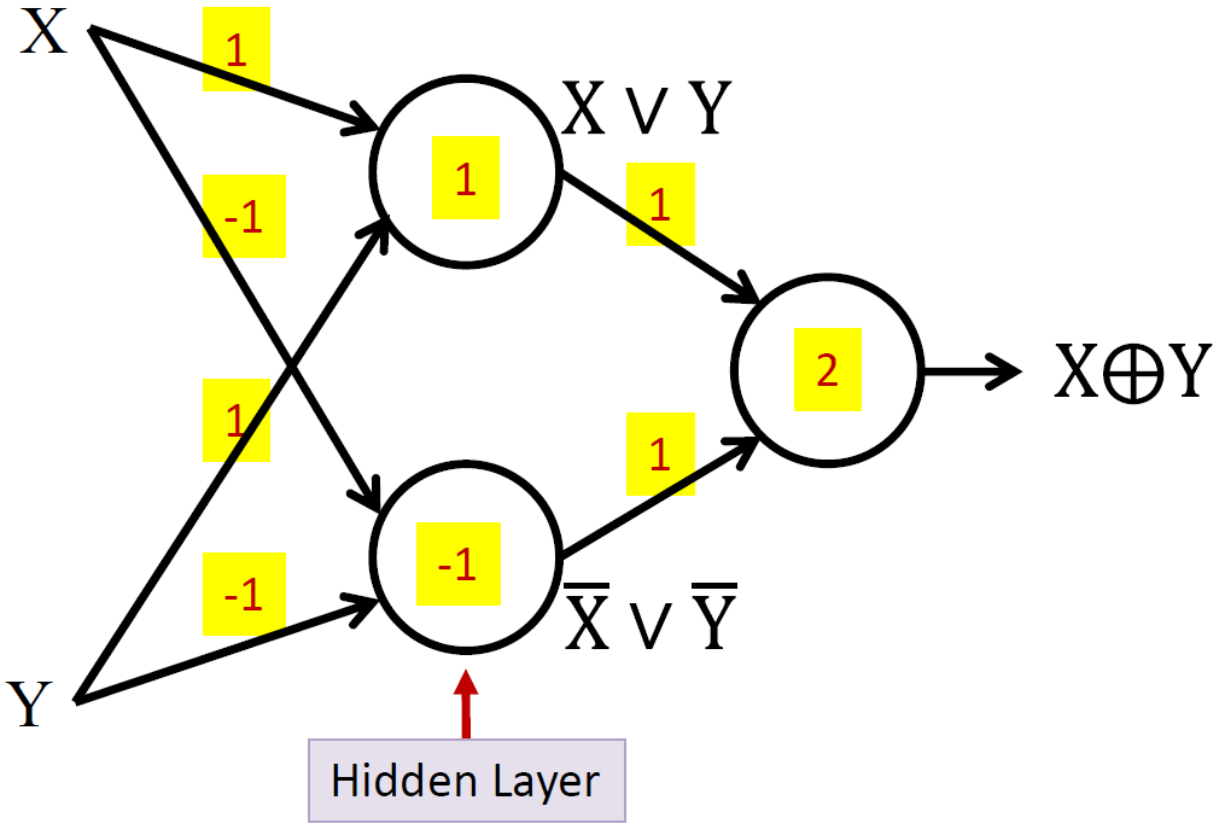
Perceptron & Logic Operations

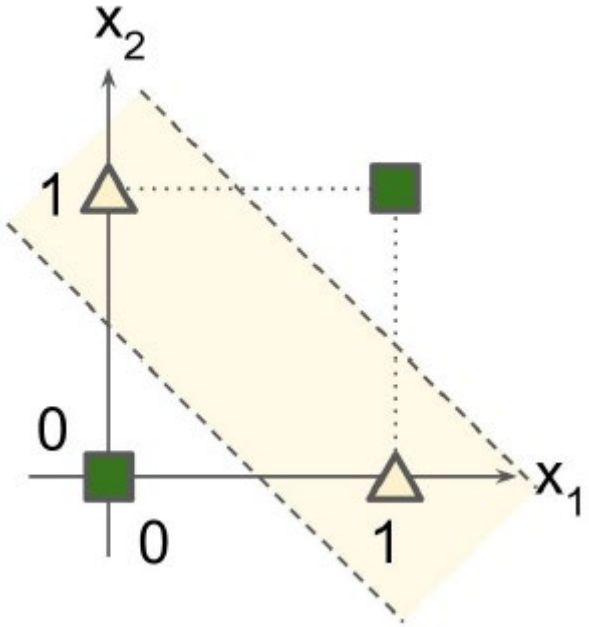
Logic Gates

Name	NOT	AND	NAND	OR	NOR	XOR	XNOR
Alg. Expr.	$\overline{A}$	AB	$\overline{AB}$	$A+B$	$\overline{A+B}$	$A \oplus B$	$\overline{A \oplus B}$
Symbol							
Truth Table	A X 0 1 1 0	B A X 0 0 0 0 1 0 1 0 0 1 1 1	B A X 0 0 1 0 1 1 1 0 1 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 1	B A X 0 0 1 0 1 0 1 0 0 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 0	B A X 0 0 1 0 1 0 1 0 0 1 1 1

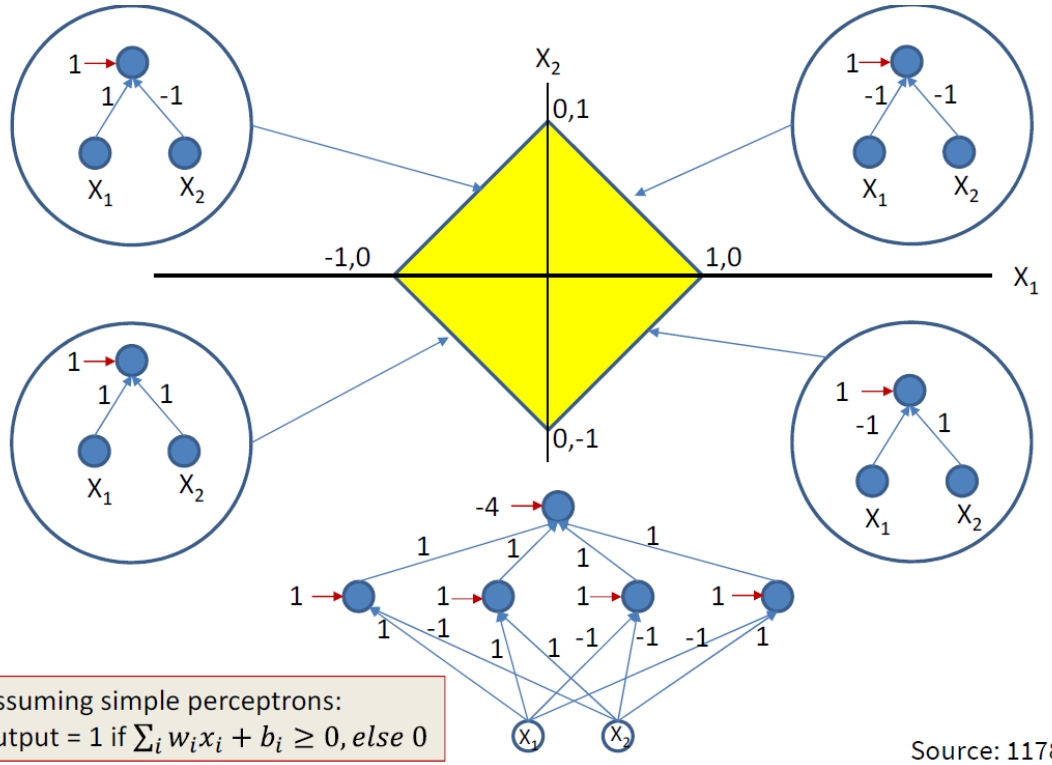


Multi-layer perceptron XOR



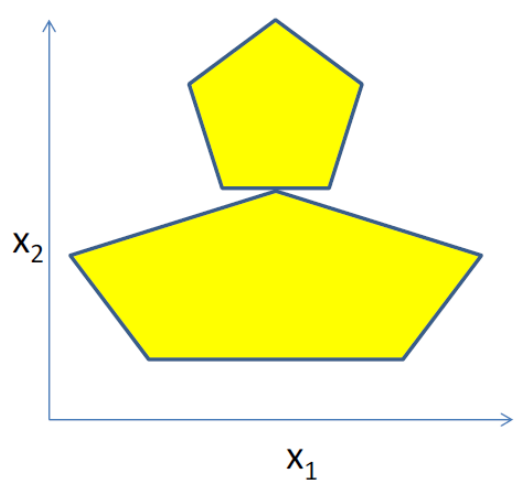


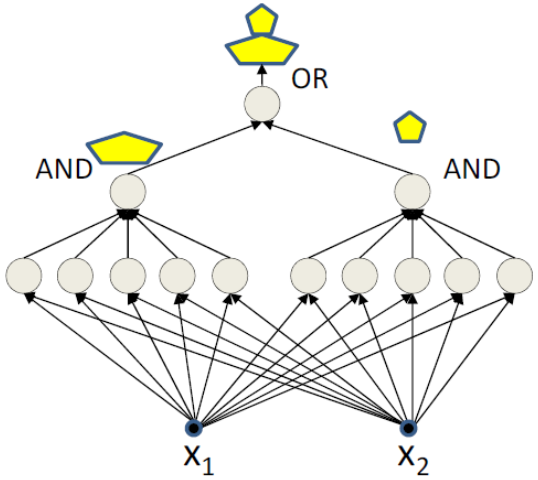
Complex Decision Boundaries



Assuming simple perceptrons:
output = 1 if $\sum_i w_i x_i + b_i \geq 0, else\ 0$

Source: 11785 lecture notes

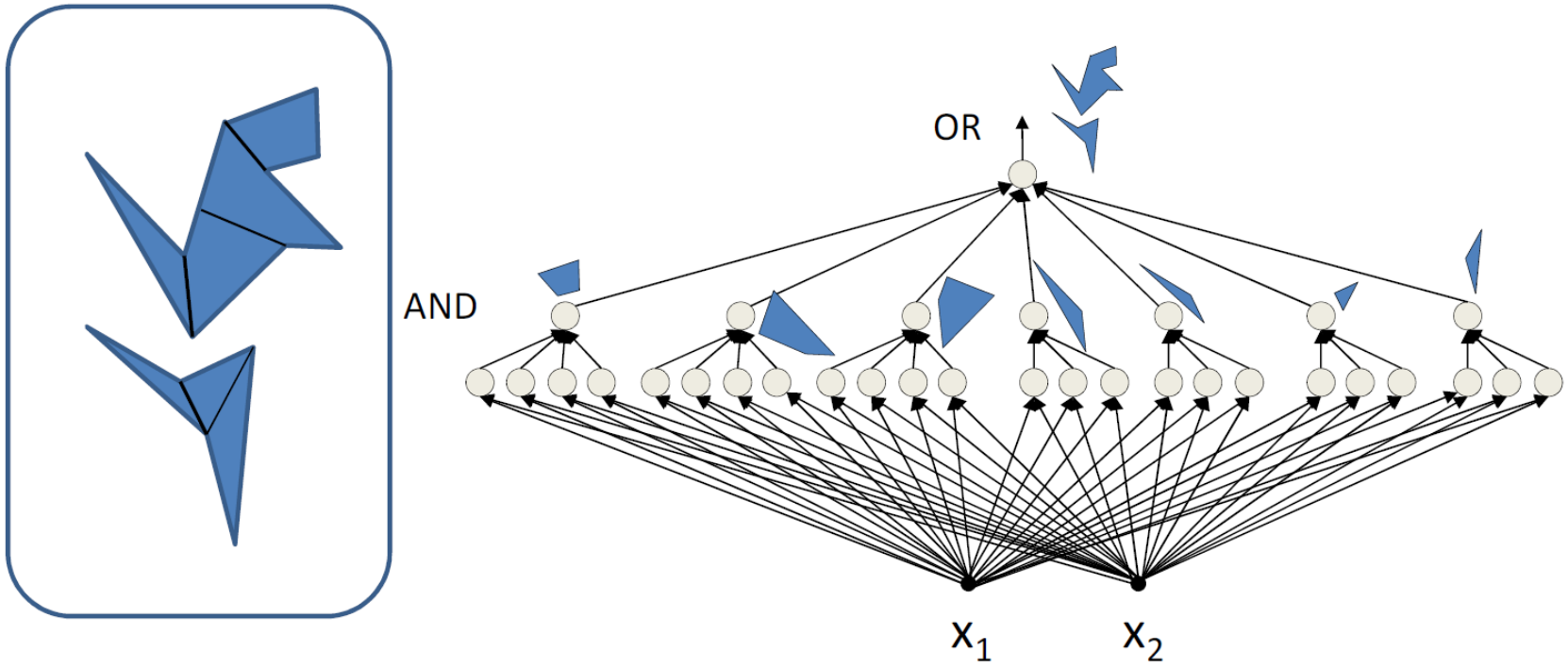




Network to fire if the input is in the yellow area

- "OR" two polygons
- A third layer is required

Even More Complex Decision Boundaries!



- Can compose *arbitrarily* complex decision boundaries

Source: 11785 lecture notes

MLP as a continuous-valued regression