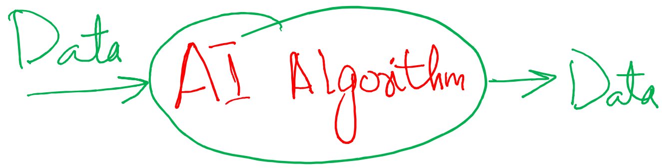
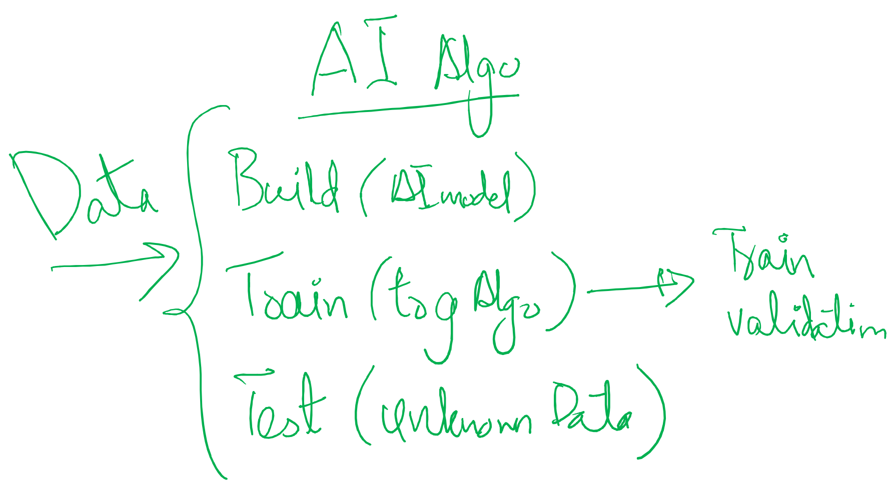
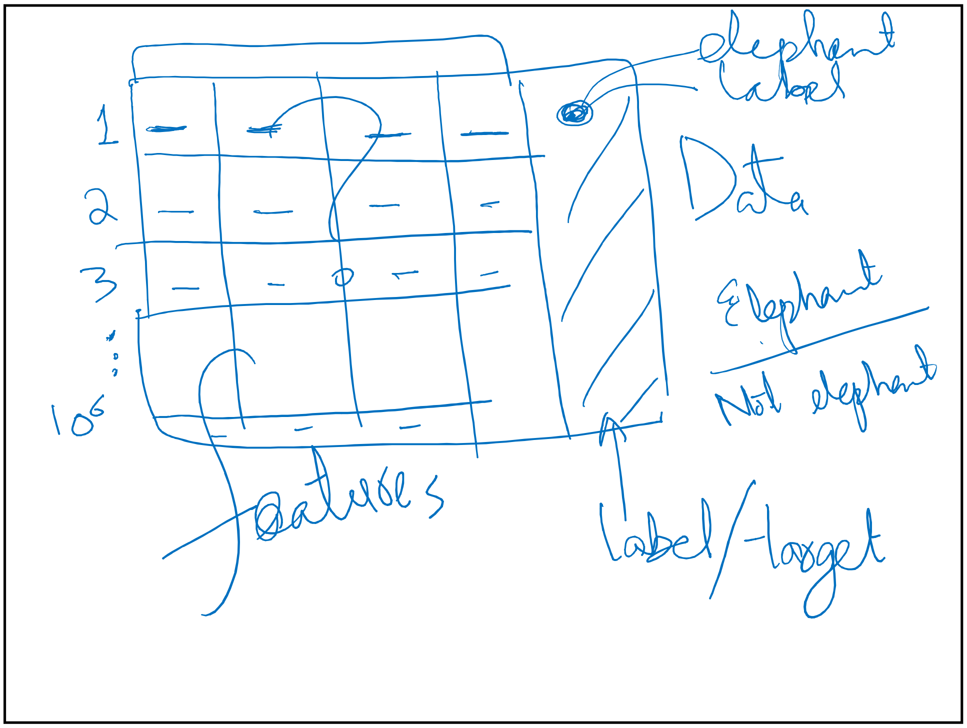
# DATA, AI ALGO, DATA

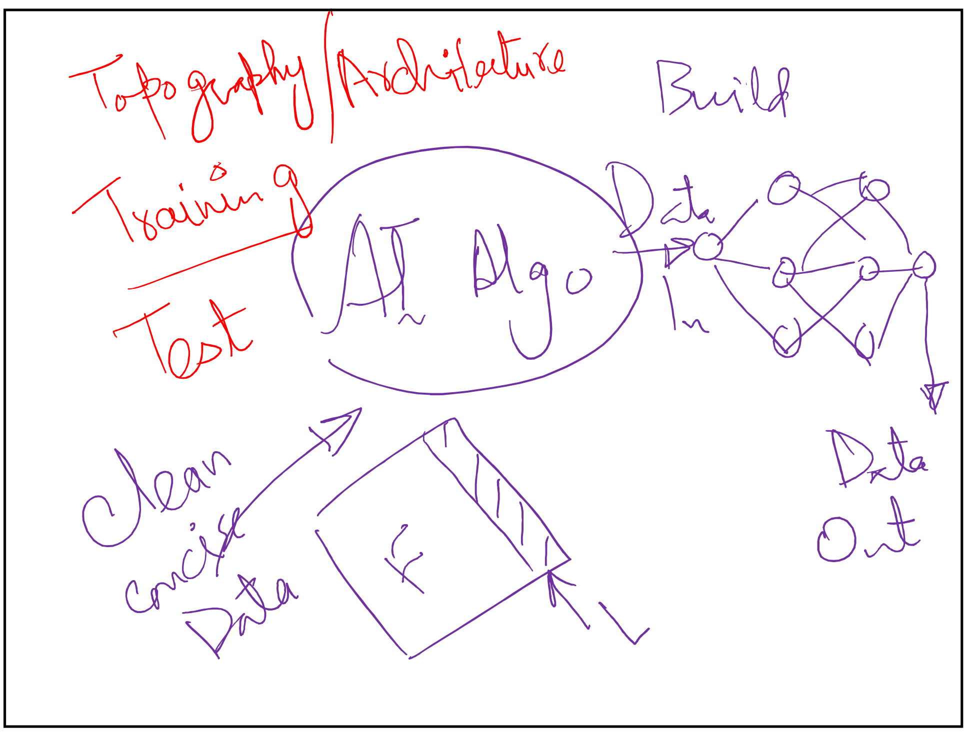


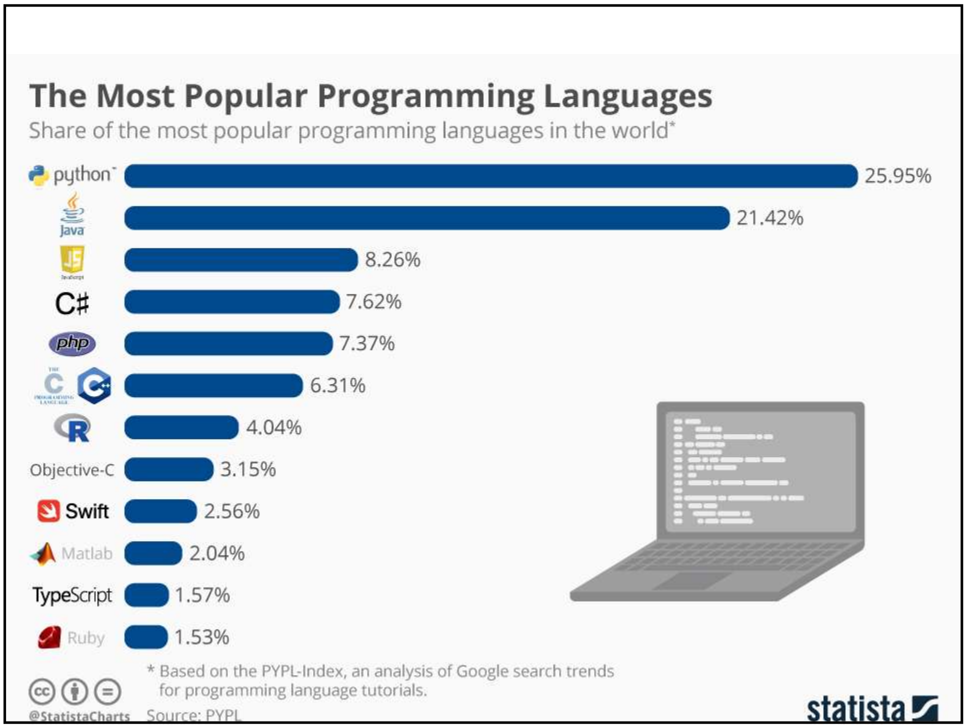

Databases
Kaggle
UCI
Github
Private sources

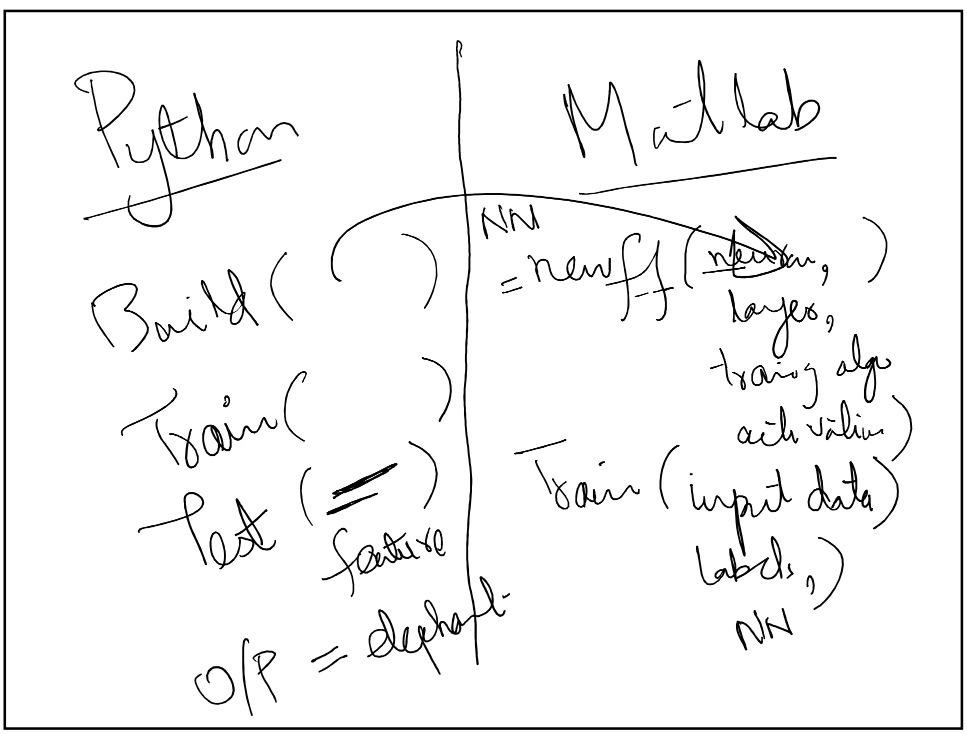
REMIND TO SHOW
AFTER GC exercise




elephant
label
1
2
3
$10^6$
Data
Elephant
Not elephant
features
label/target


Topography/Architecture
Build
Training
Test
AI Algo
Data In
Data Out
Clean concise Data
F
L


The Most Popular Programming Languages
Share of the most popular programming languages in the world*
python
25.95%
Java
21.42%
8.26%
C#
7.62%
php
7.37%
6.31%
4.04%
Objective-C
3.15%
Swift
2.56%
Matlab
2.04%
TypeScript
1.57%
Ruby
1.53%
* Based on the PYPL-Index, an analysis of Google search trends for programming language tutorials.
@StatistaCharts
Source: PYPL
statista


Python
Matlab
Build( )
Train( )
Test ( )
feature
O/P = elephant
NN
= newff(
layers,
Train (input data
labels,
NN

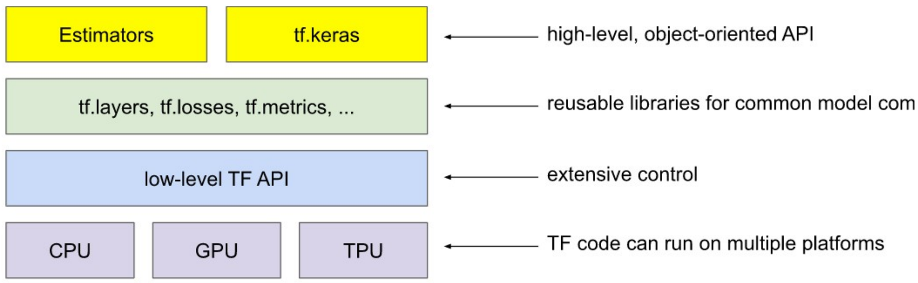
## INTRODUCTION TO TENSORFLOW

- TensorFlow, an end-to-end open source platform for machine learning
- Its APIs are arranged hierarchically, high-level APIs built on the low-level APIs
- ML researchers use the low-level APIs to create and explore new ML algorithms

## INTRODUCTION TO TENSORFLOW

- Keras is a high-level, deep learning API developed by Google for implementing neural networks
- It is written in Python and is used to make the implementation of neural networks easy

## TENSORFLOW TOOLKIT HIERARCHY




## PROGRAMMING EXERCISES

- ML concepts are into practice by coding models in tf.keras.
- **Colab** used as a programming environment, Google's version of **Jupyter Notebook** (https://jupyter.org/)
- Jupyter Notebook, Colab provides an interactive Python programming environment that combines text, code, graphics, and program output

## PROGRAMMING EXERCISES

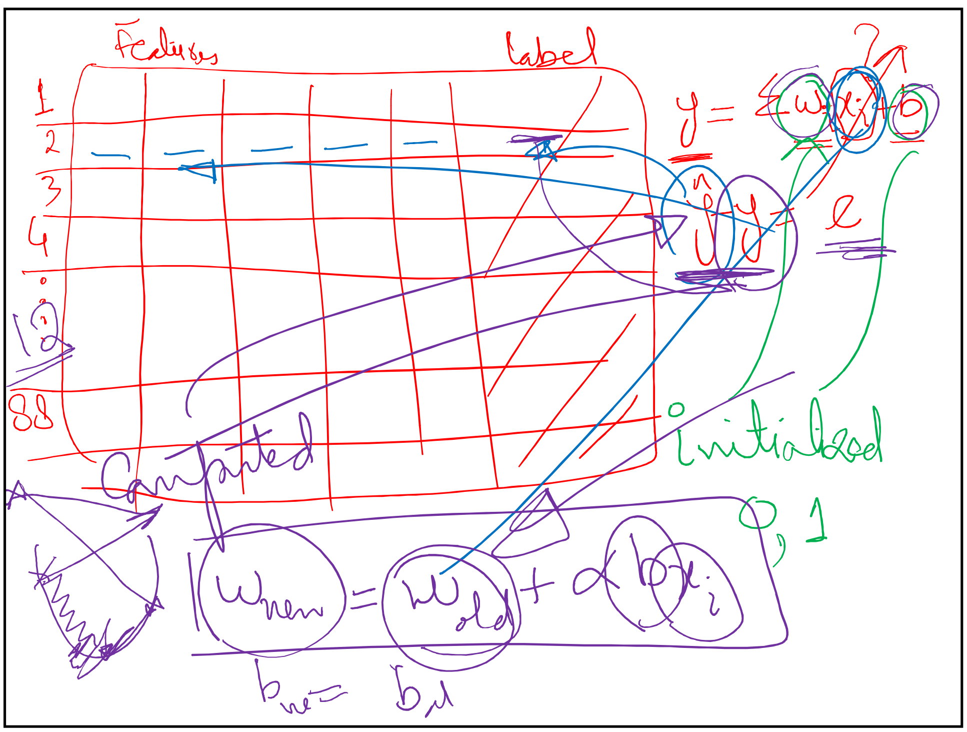
- Using tf.keras requires at least a little understanding of the following two open-source Python libraries:
- NumPy, which simplifies representing arrays and performing linear algebra operations.
- pandas, which provides an easy way to represent datasets in memory.

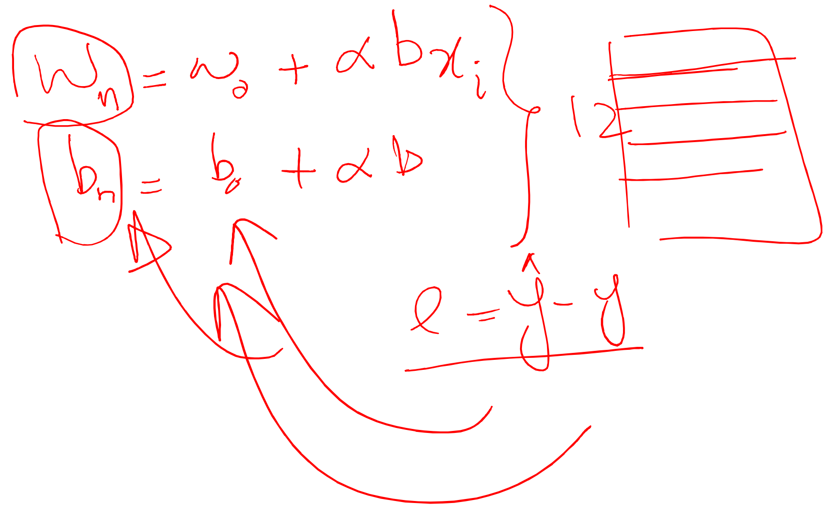
## PROGRAMMING EXERCISES

- Learn enough about NumPy and pandas to understand tf.keras code.
- Learn how to use Colabs.
- Become familiar with linear regression code in tf.keras.
- Evaluate loss curves.
- Tune hyper-parameters.

## PROGRAMMING EXERCISES

- NumPy Ultraquick Tutorial
- pandas UltraQuick Tutorial
- To explore linear regression and hyper-parameter tuning in tf.keras:
- Linear Regression with Synthetic Data Colab exercise, which explores linear regression with a toy dataset.
- Linear Regression with a Real Dataset Colab exercise for analysis you should do on a real dataset.





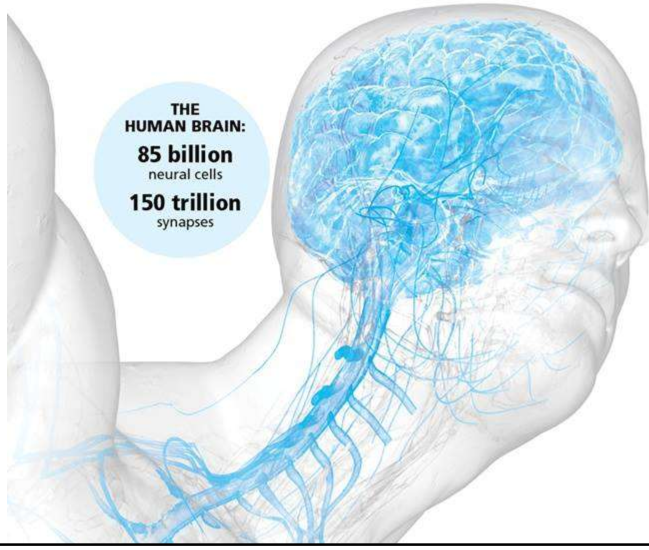

# BEST INTELLIGENCE

لَقَدْ خَلَقْنَا الْإِنسَانَ فِي أَحْسَنِ تَقْوِي

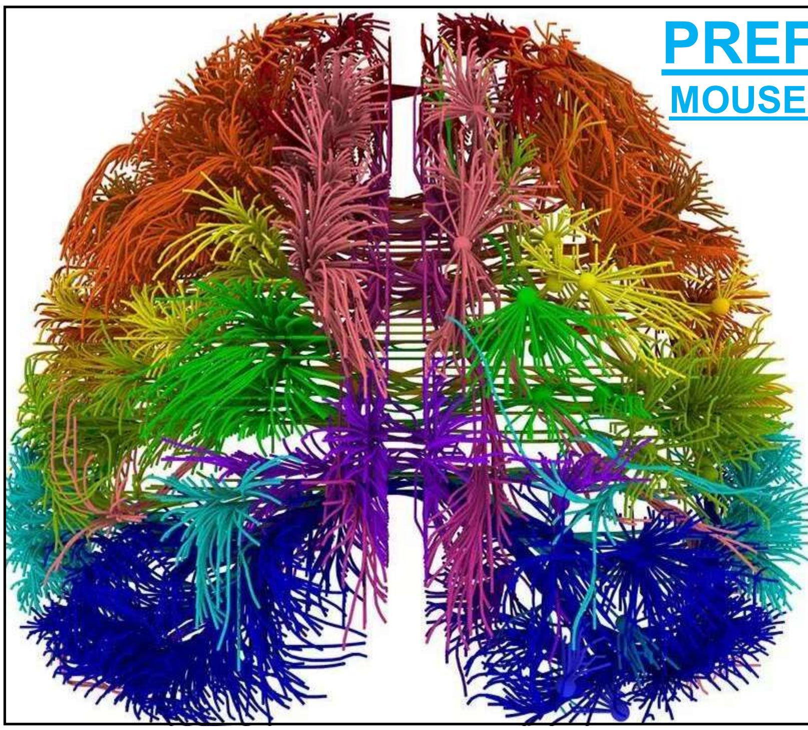
*Indeed Allah has created the human in the best of forms. (Surah Teen, verse 4)*

This verse indicates to Insaan (humans) being the best and most noble creation of Allah.

## THE BEST AI
## INSPIRED BY HUMAN BRAIN




## PREFACE
### MOUSE BRAIN



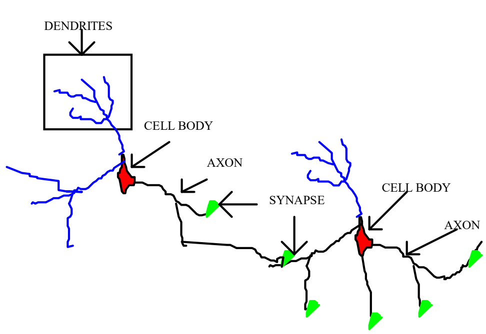

The most complex object in the known universe: **BRAIN**,

only uses **20 watts** of power. It would require a nuclear power plant to energize a computer the size of a city block to mimic your brain, and your brain does it with just 20 watts.

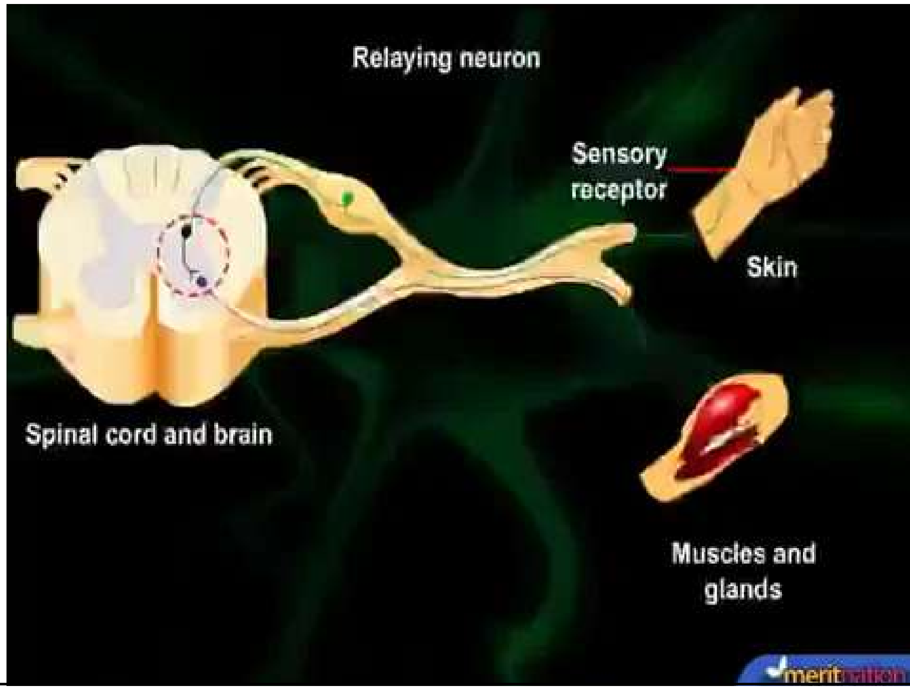
@michiokaku, physicist

# Human and Artificial Neurons (FUNCTIONALITIES)

- Much is still unknown
  - How brain train itself/process information
  - Theories abound




- Dendrites→ input
- Cell Body
  - soma
- Axon→ output
  - Electrical pulses
- Synapses
  - junction
- Biochemical activities
- Brain's functionality
  - Thought, emotion and cognition




# IMPORTANT TIMELINES

- Roots of work on AI/ANN are **Neurobiological studies** (more than one century ago):
  - How do nerves behave when stimulated by different magnitudes of electric current?
  - Is there a minimal threshold needed for nerves to be activated?
  - Given that no single nerve cell is long enough, how do different nerve cells communicate among each other?

1. Phy Connections

medical

# IMPORTANT TIMELINES

- **Psychological studies**: 2. Learning
  - How do animals learn, forget, recognize and perform other types of tasks?
- **Psycho-physical** experiments helped to understand how individual neurons and groups of neurons work.

3. Modelling

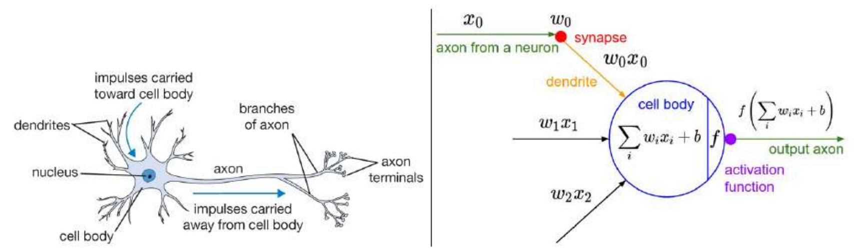
- **McCulloch and Pitts** introduced the first mathematical model of single neuron, widely applied in subsequent work.

# IMPORTANT TIMELINES

4. Simulations

- Initial simulations using formal logic
  – logic functions such as "a or b" and "a and b“
- McCulloch and Pitts (1943)
  – several assumptions about neurons
    - binary devices
    - fixed thresholds

# Artificial Neural Networks




A cartoon drawing of a biological neuron (left) and its mathematical model (right).

# IMPORTANT TIMELINES

ENGR & PHYSICIANS

- Computer simulations matured
  - Two groups (Farley and Clark, 1954 (IBM Engrs); Rochester, Holland, Haibit and Duda, 1956 (McGill Uni neuroscientists))
  - Whenever Engrs models did not work, they consulted the neuroscientists
  - Multidisciplinary trend continues today

## IMPORTANT TIMELINES

- Psychologists entered for Training Logics
  - Rosenblatt (1958) stirred considerable interest and activity by Perceptron
    - The Perceptron had three layers with the middle layer known as the association layer
    - This system could learn to connect or associate a given input to a random output unit

## IMPORTANT TIMELINES

- ADALINE (ADAptive LInear Element)
- developed in 1960 by Widrow and Hoff of Stanford University
  - Least-Mean-Squares (LMS) learning rule

## IMPORTANT TIMELINES

In 1969 Minsky and Papert generalized the limitations of

In the book they said:

"...our intuitive judgment that the extension (to multilayer systems) is sterile". NOT CAPABLE

The significant result of the book

- eliminate funding
- Disenchantments of researchers
- Prejudice against this field

## IMPORTANT TIMELINES

- **Innovation in the 70's: (even funding was minimal)**
  - Individual researchers continue laying foundations
  - von der Marlsburg (1973): competitive learning and self-organization

**Big neural-nets boom in the 80's and re-emergence**

- Books, conferences, university programmes, funding by major ctry
  - Grossberg: adaptive resonance theory (ART)
  - Hopfield: Hopfield network
  - Kohonen: self-organising map (SOM)

# IMPORTANT TIMELINES

- Oja: neural principal component analysis (PCA)
- Ackley, Hinton and Sejnowski: Boltzmann machine
- Rumelhart, Hinton and Williams: backpropagation

**Diversification in 90's, 2000's and Till Today:**

- Machine learning: mathematical rigor, Bayesian methods, information theory, ***support vector machines (now state of the art!)***, ...
- Computational neurosciences: To understand low-level compartmental models of individual neurons to large-scale brain models
- **Deep learning evolution, last few years**

# Some of most common ANN models

| Period | Inventors | Name of the model | Applications | Learning Mode |
|---|---|---|---|---|
| 1957-1960 | F. Rosenblatt | Perceptron | Type character recognition and Classification | Supervised |
| 1959-1962 | B. Widrow<br>M. E. Hoff | LMS | Prediction, noise cancellation | Supervised |
| 1971-1994 | I. Aleksander<br>J. G. Taylor<br>T. G. Clarkson<br>D. Gorse | RAM model and PRAM (Weightless neurons) | Pattern recognition | Supervised reinforcement |
| 1974-1986 | P. Werbos,<br>D. Parker<br>D. Rumelhart | Back propagation | Pattern, recognition, Prediction, etc. | Supervised |
| 1975-1983 | K. Fukushima | Neocognitron | Pattern recognition | Supervised/ Unsupervised |
| 1978-1986 | G. Carpenter<br>S. Grossberg | Adaptive Resonance Theory (ART) | Recognition: classification of complex pattern | Supervised/ Unsupervised |
| 1980 | T. Kohonen | Self Organizing Map | Image recognition | Unsupervised |
| 1982 | B. Wilkie<br>J. Stonham<br>I. Aleksander | WISARD | Pattern and image recognition | RAM based model |
| 1982-1984 | J. Hopfield | Associative Memory | Speech Processing | Association (Hebbian) |
| 1985 | B. Kosko | Bi-directional Associative Memory (BAM) | Image Processing | Association (Hebbian) |
| 1980-1993 And so on | M. J. D. Powell<br>J. E. Moody<br>C. J. Darken<br>S. Renals<br>T. Poggio<br>F. Girosi | Radial Base Function (RBF) (Hybrid system) Hardware systems | Prediction /Recognition | Supervised and unsupervised |

# THE EVOLUTION

- Perceptron
- Multi layer perceptron
- Feed-forward back-propagation ANN
- Recurrent ANN
- Long-short term memory (LSTM)
- Convolutional ANN
- Attentions Nets aka Transformers aka Regenerative networks aka Large language models