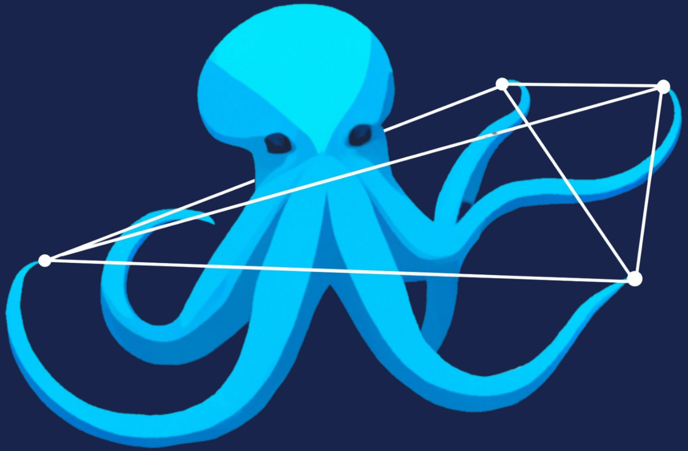
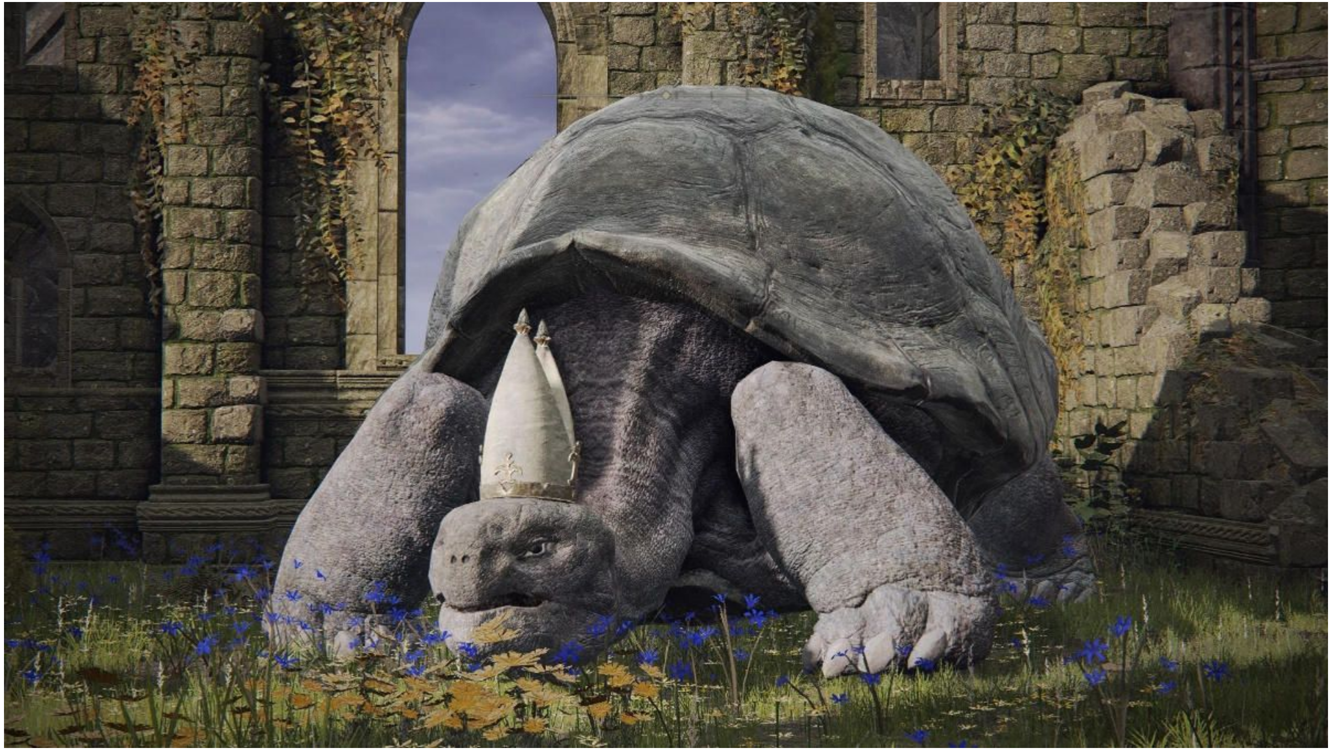


# Machine Learning Month

# Neural Networks with Tensorflow

# Plan for today

- 15:00 - 15:30 – Some Theory
- 15:30 - 16:00 – Some Practical Theory
- 16:00 - 16:15 – Break + QnA about the competition
- 16:15 - 17:00 – Practical



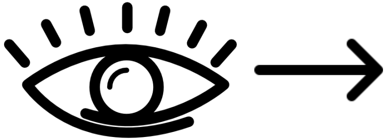
Could this be a dog?

# How do you know?

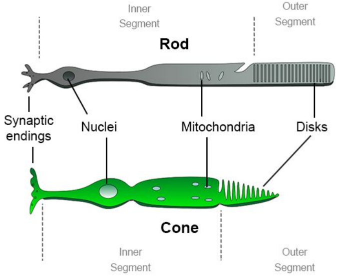
# How do you know?

Vision

# How do you know?

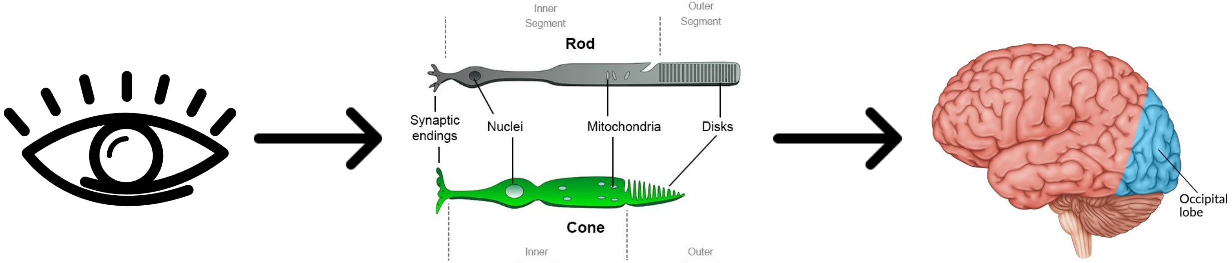


Vision



Edge detection

# How do you know?



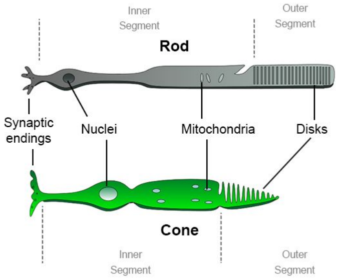
Vision

Edge detection

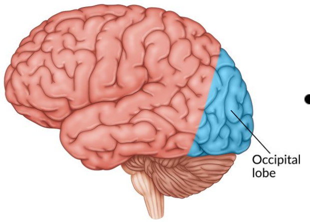
Feature extraction

# How do you know?

Vision



Edge detection



Feature extraction

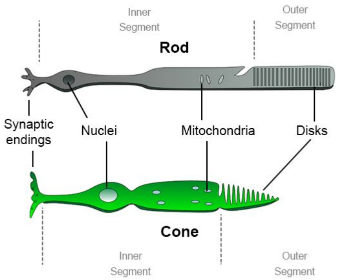
Answer

# Can a Machine do the same?

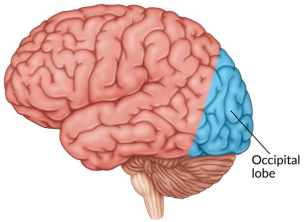
Yes!

# Science Power Go!

Vision →



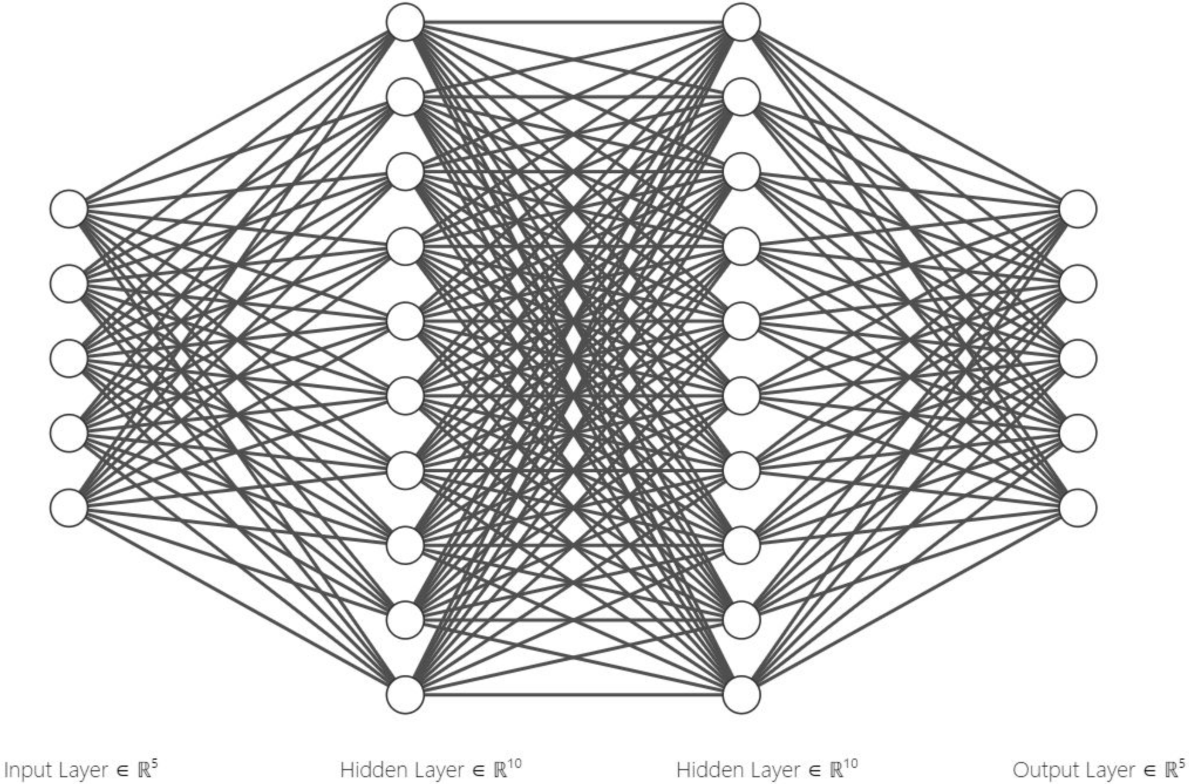
Edge detection →



Feature extraction →

Answer

# Neural Networks



# Welcome to the Third Lecture

Our Objectives:

- Overview of Neural Networks (NNs)
- Why should we use them?
- What are Keras and Tensorflow?
- NN Training
- Practical Session

# Overview of Neural Networks

# Why Neural Networks?

# Why Neural Networks?

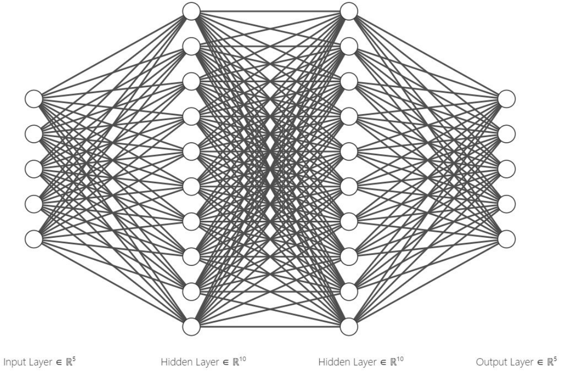
- Humans are smart! Can we copy them?

# When do we use NNs?

- We need high accuracy
- The data does not have an obvious pattern
- We have a LOT of data
- Explainability is not important

# What are Neural Networks?

- These are a series of algorithms that **attempt** to mimic the human brain


Input Layer ∈ $\mathbb{R}^5$
Hidden Layer ∈ $\mathbb{R}^{10}$
Hidden Layer ∈ $\mathbb{R}^{10}$
Output Layer ∈ $\mathbb{R}^5$

≈

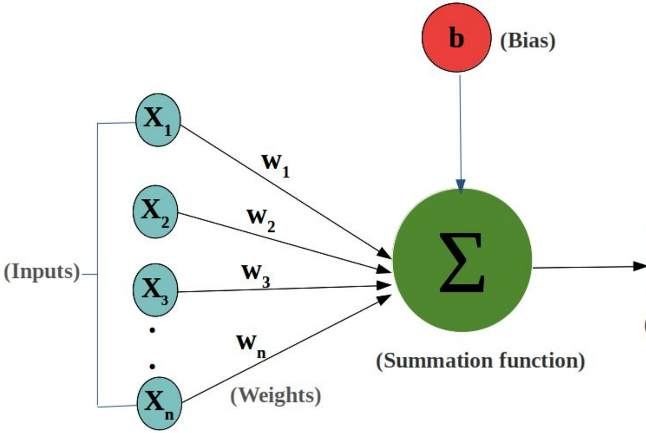
# Formally

- NNs are layers of nodes where each node can be expressed as the result of the previous layer.

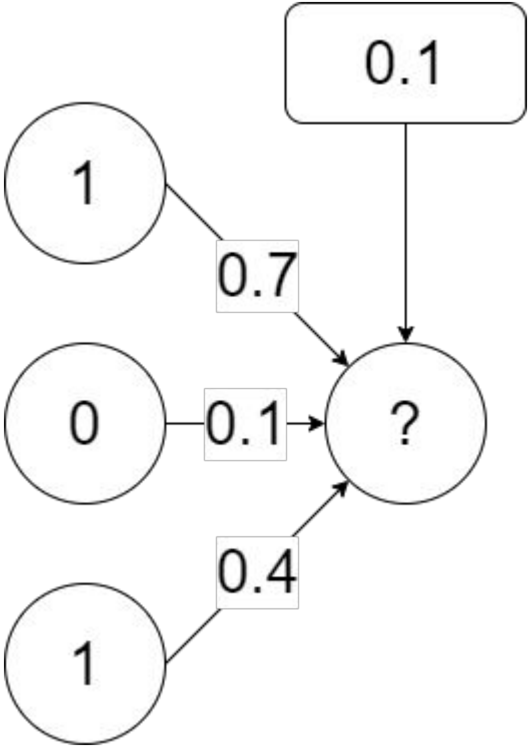
$$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

# Weights and Biases

- The contribution of each neuron is known as a weight.
- There is also an offset value in a neuron known as a bias
- Together they determine if the neuron activates



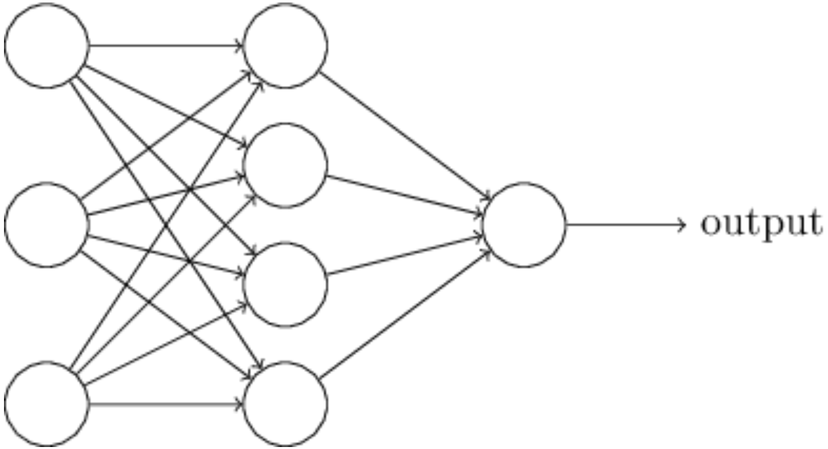
# Example:



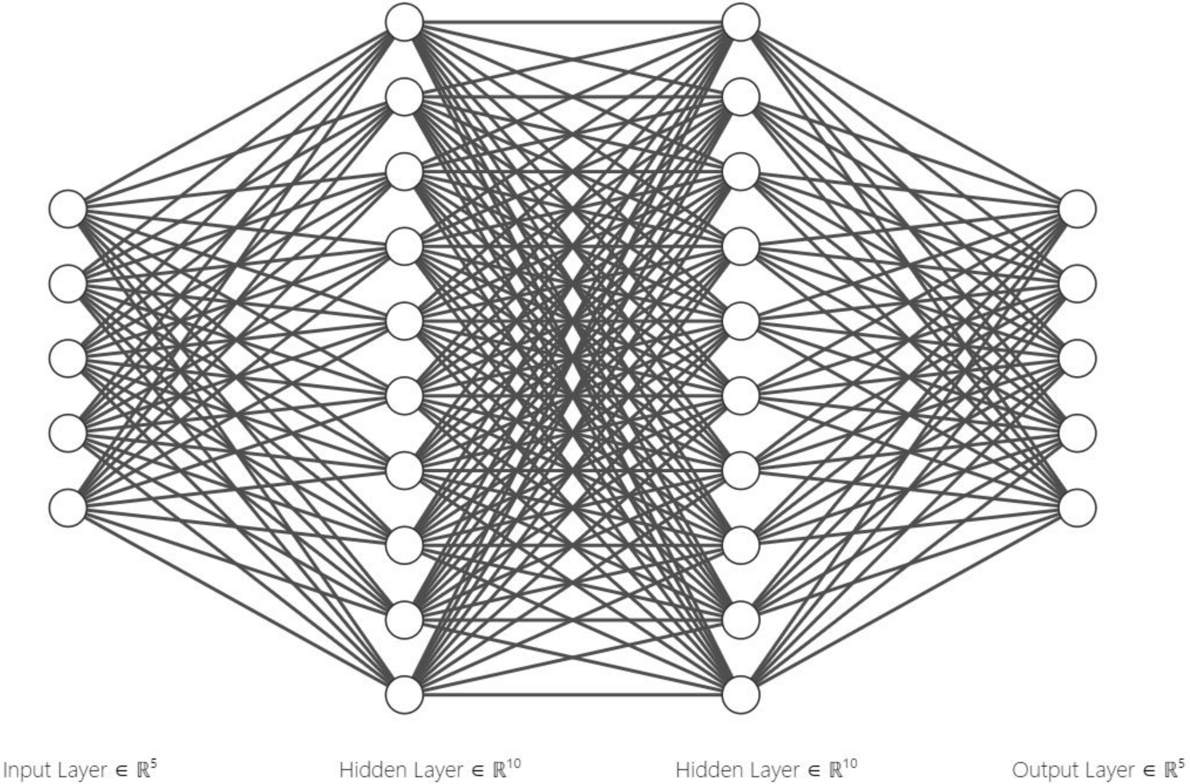
The neuron activates if a threshold of 1 is reached

# Fully Connected Neural Networks

- NN in which all nodes in one layer are connected to all nodes in the next layer

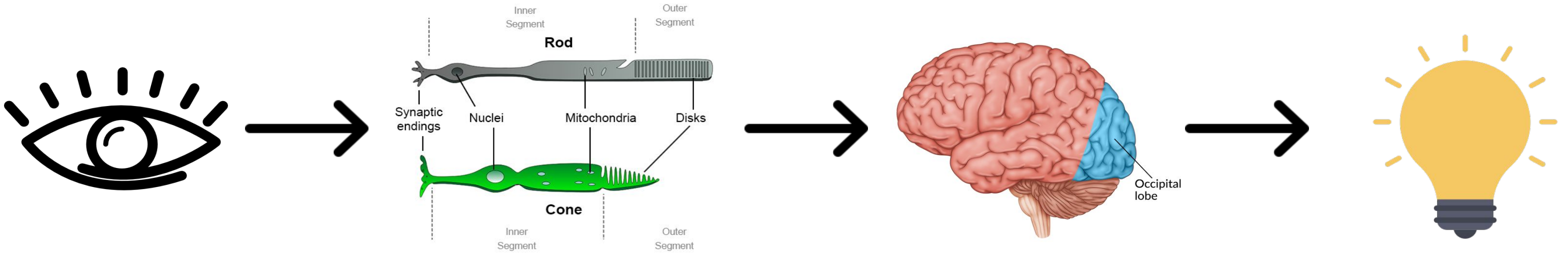


# Layers


Input Layer ∈ ℝ⁵
Hidden Layer ∈ ℝ¹⁰
Hidden Layer ∈ ℝ¹⁰
Output Layer ∈ ℝ⁵

# Idea

- Each subsequent layer in the Neural Network represents an increase in abstraction



Vision

Edge detection

Feature extraction

Answer

Increase in complexity

# Definitions

# Activation Functions

These determine whether or not a neuron fires

Examples:

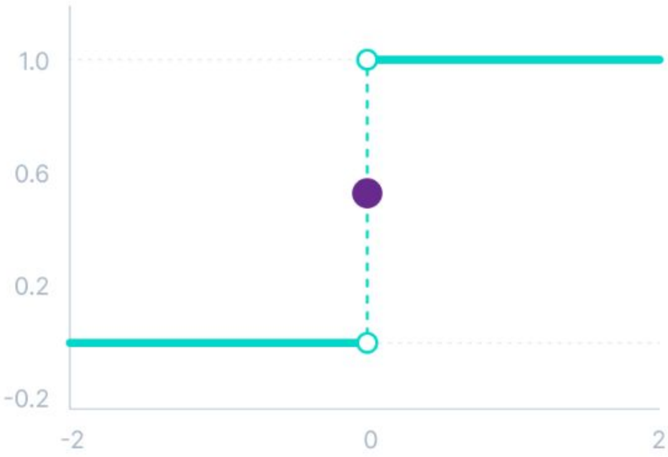
- Binary
- Linear
- Sigmoid
- ReLu

$$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 + bias$$

Remember

# Binary

Binary Step Function

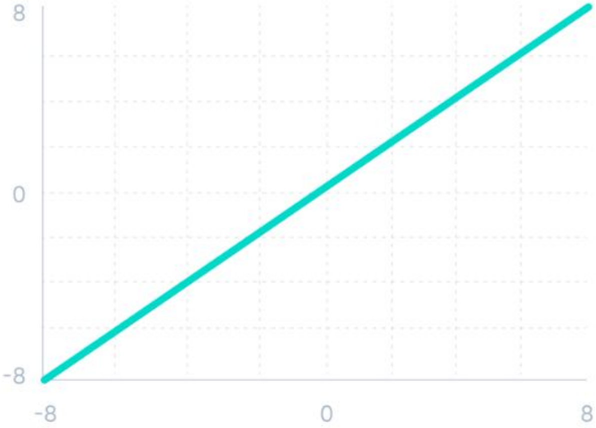


*Binary step*

$$f(x) = \begin{cases} 0 & for\ x < 0 \\ 1 & for\ x \geqslant 0 \end{cases}$$

# Linear

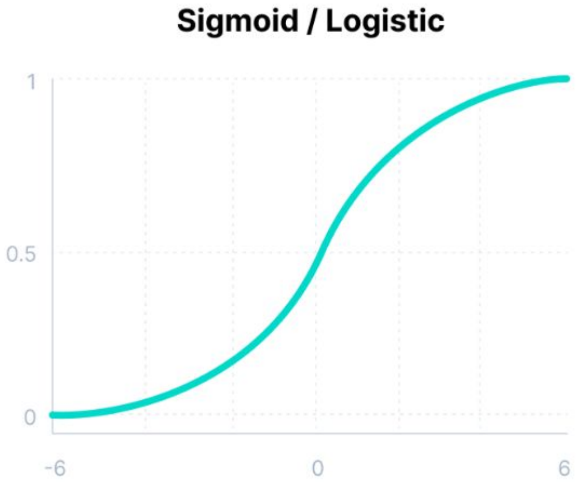
Linear Activation Function



*Linear*

$$f(x) = x$$

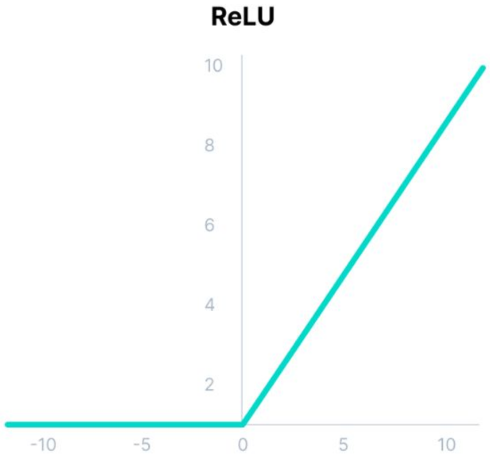
# Sigmoid



*Sigmoid / Logistic*

$$f(x) = \frac{1}{1 + e^{-x}}$$

# ReLu



*ReLU*

$$f(x) = max\ (0, x)$$

# Optimizers

These determine how we update the weights

Examples:

- Gradient Descent
- Stochastic Gradient Descent
- Adam
- RMS prop

# Optimizers

These determine how we update the weights

Examples:

- Gradient Descent
- Stochastic Gradient Descent
- Adam
- RMS prop

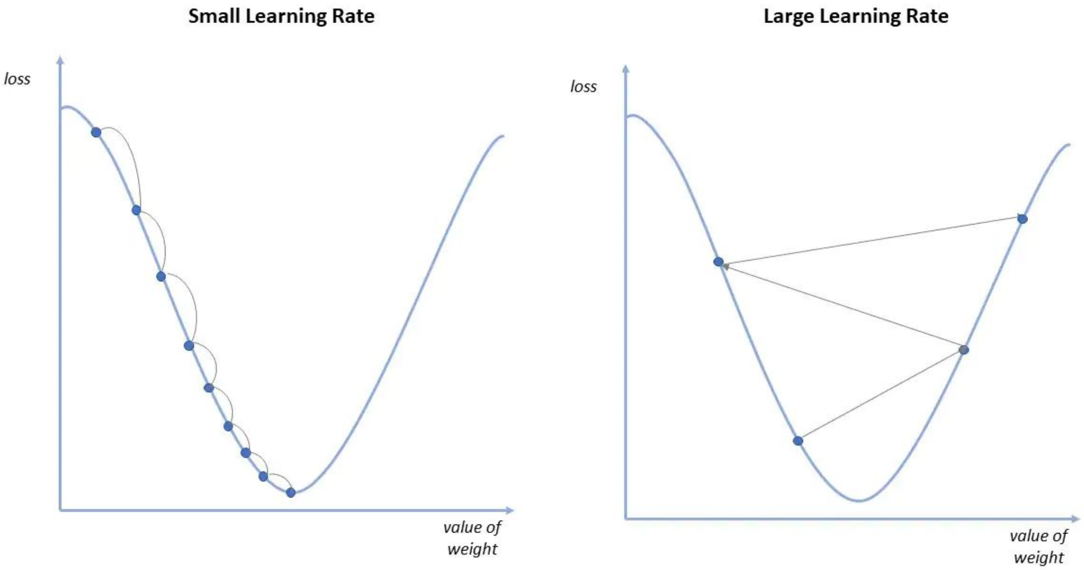
# Gradient Descent

- Minimizes error by calculating the slope of the loss function

$$W_{new} = W_{old} - \alpha * \frac{\partial(Loss)}{\partial(W_{old})}$$

α = Learning Rate
W = Neuron Weight
Loss = Loss function (MAE, MSE, Accuracy …)

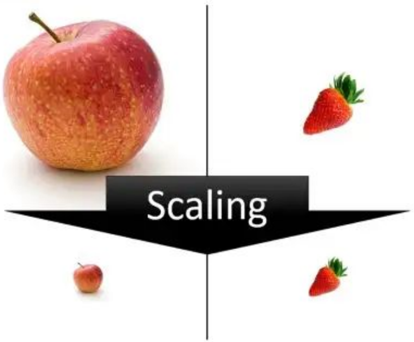
# We need to select a carefully



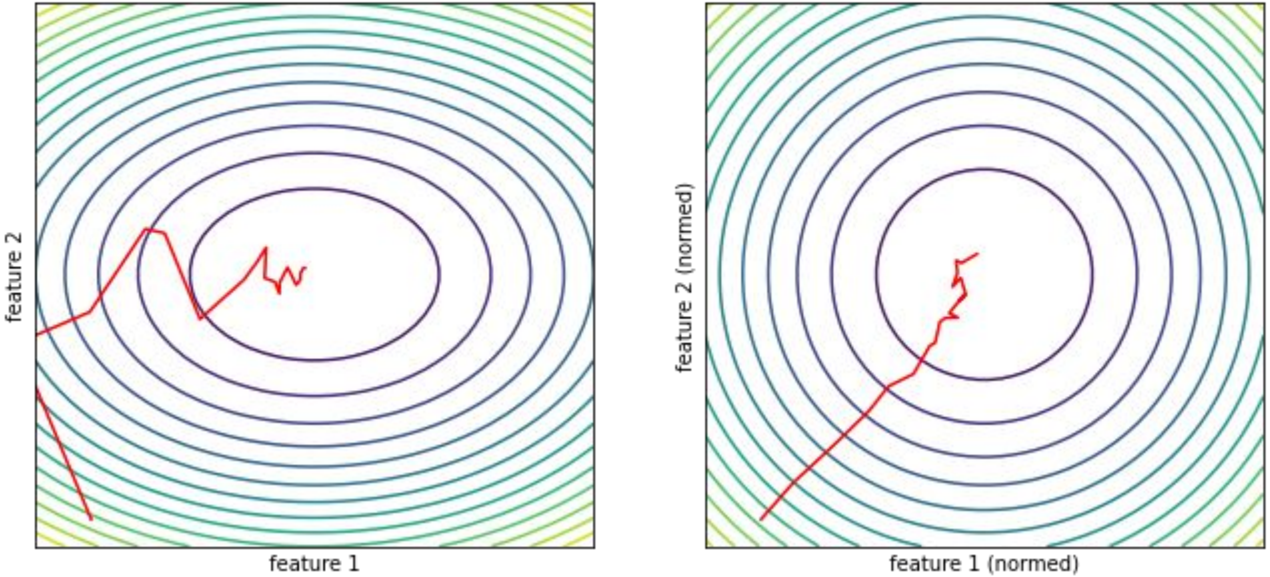
# Additional Pre-processing

# Scaling

- We should ensure that all data posses the same scale
- Otherwise, variables with greater distance between their values would given higher weightage



# Effect of Scaling



# Tools

# How will build NNs?

- Tensorflow is an open source python library created by Google

- Algorithms are implemented in C++ for blazing fast performance

TensorFlow

# How will build ANNs?

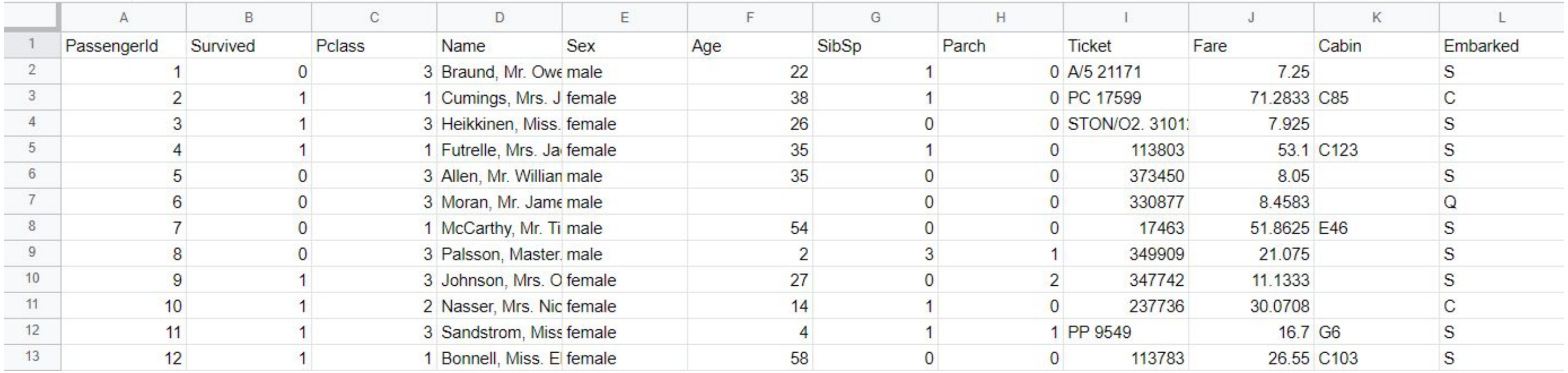
- Keras is a high level API for tensorflow

- It is designed to be human readable and simple

Keras

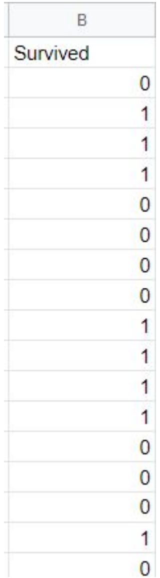
Easy to work with

# Let’s try an example



| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
| 2 | 1 | 0 | 3 | Braund, Mr. Owe | male | 22 | 1 | 0 | A/5 21171 | 7.25 | | S |
| 3 | 2 | 1 | 1 | Cumings, Mrs. J | female | 38 | 1 | 0 | PC 17599 | 71.2833 | C85 | C |
| 4 | 3 | 1 | 3 | Heikkinen, Miss. | female | 26 | 0 | 0 | STON/O2. 3101 | 7.925 | | S |
| 5 | 4 | 1 | 1 | Futrelle, Mrs. Ja | female | 35 | 1 | 0 | 113803 | 53.1 | C123 | S |
| 6 | 5 | 0 | 3 | Allen, Mr. Willian | male | 35 | 0 | 0 | 373450 | 8.05 | | S |
| 7 | 6 | 0 | 3 | Moran, Mr. Jame | male | | 0 | 0 | 330877 | 8.4583 | | Q |
| 8 | 7 | 0 | 1 | McCarthy, Mr. Ti | male | 54 | 0 | 0 | 17463 | 51.8625 | E46 | S |
| 9 | 8 | 0 | 3 | Palsson, Master. | male | 2 | 3 | 1 | 349909 | 21.075 | | S |
| 10 | 9 | 1 | 3 | Johnson, Mrs. O | female | 27 | 0 | 2 | 347742 | 11.1333 | | S |
| 11 | 10 | 1 | 2 | Nasser, Mrs. Nic | female | 14 | 1 | 0 | 237736 | 30.0708 | | C |
| 12 | 11 | 1 | 3 | Sandstrom, Miss | female | 4 | 1 | 1 | PP 9549 | 16.7 | G6 | S |
| 13 | 12 | 1 | 1 | Bonnell, Miss. E | female | 58 | 0 | 0 | 113783 | 26.55 | C103 | S |

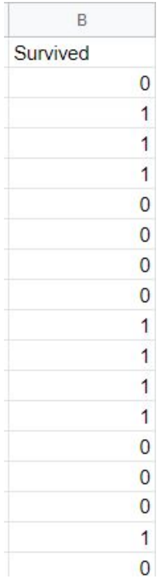
# Analysis



| B |
|---|
| Survived |
| 0 |
| 1 |
| 1 |
| 1 |
| 0 |
| 0 |
| 0 |
| 0 |
| 1 |
| 1 |
| 1 |
| 1 |
| 0 |
| 0 |
| 0 |
| 1 |
| 0 |

# Analysis

- Our target is binary
- It does not make sense for a person to be 0.4 alive
- We need to re-evaluate our methods

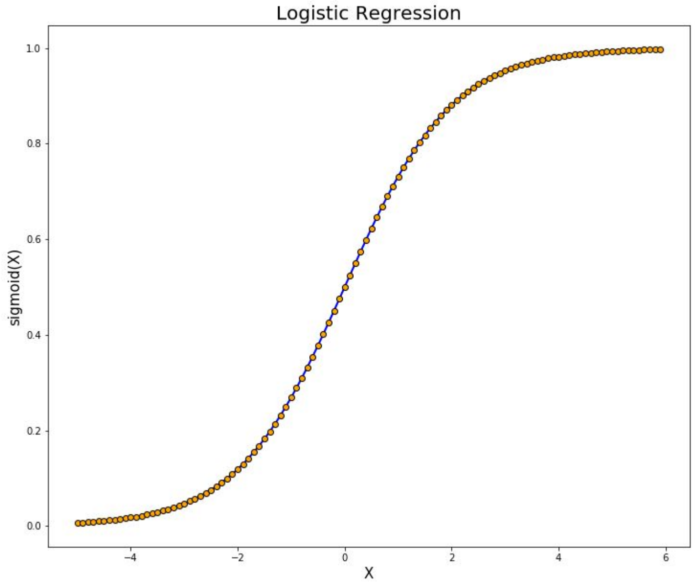


| B |
|---|
| Survived |
| 0 |
| 1 |
| 1 |
| 1 |
| 0 |
| 0 |
| 0 |
| 0 |
| 1 |
| 1 |
| 1 |
| 1 |
| 0 |
| 0 |
| 0 |
| 1 |
| 0 |

# Logistic Regression

- What if we calculated the probability of an item being a value?

- If P(alive) > P(dead), then let the person be alive



# How do we do it in Keras?

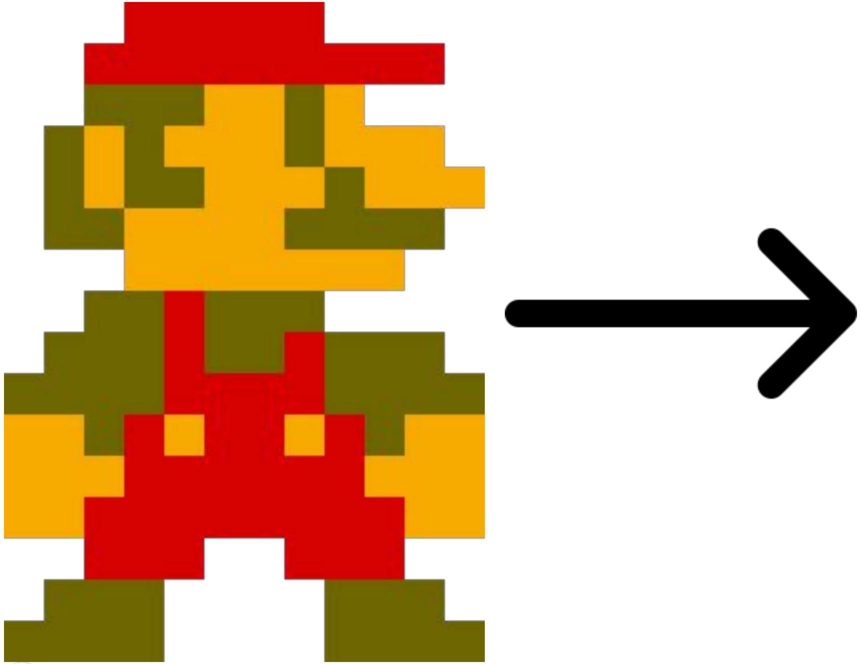
Refer to Colab Notebook

# Practice Session

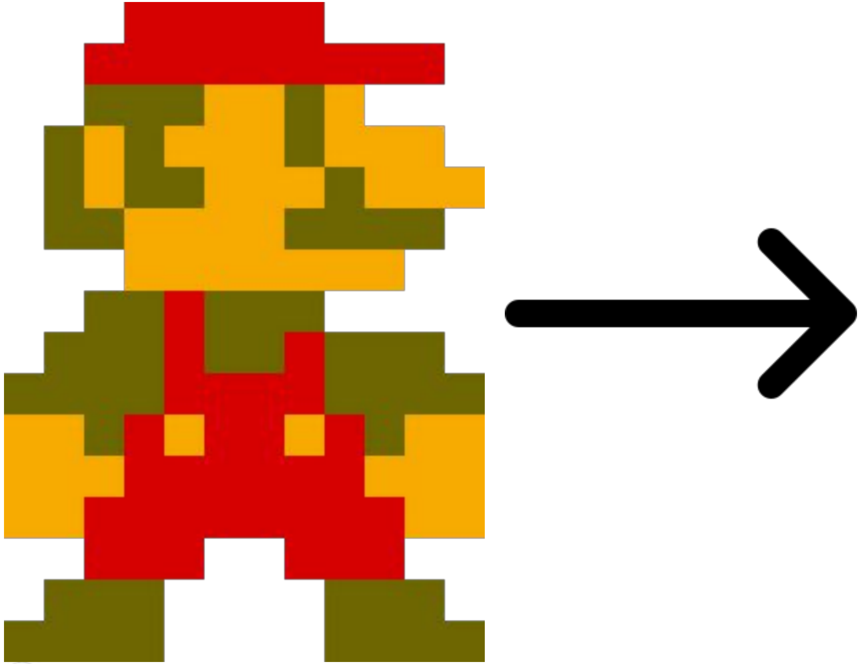
# Let’s build an image classifier

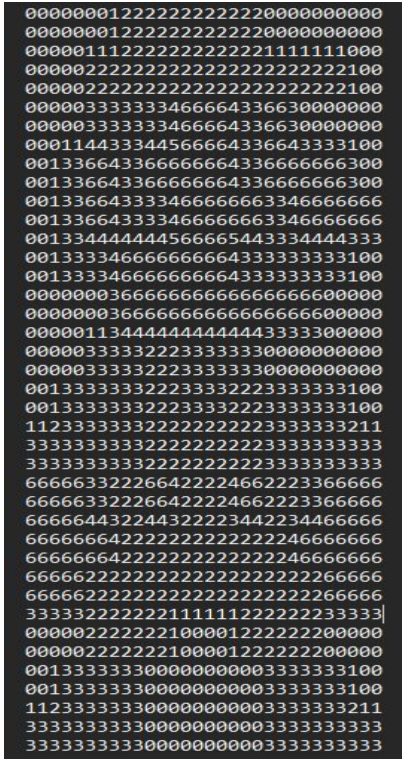
## But How?

# How do we represent images?



# Images as 2d arrays

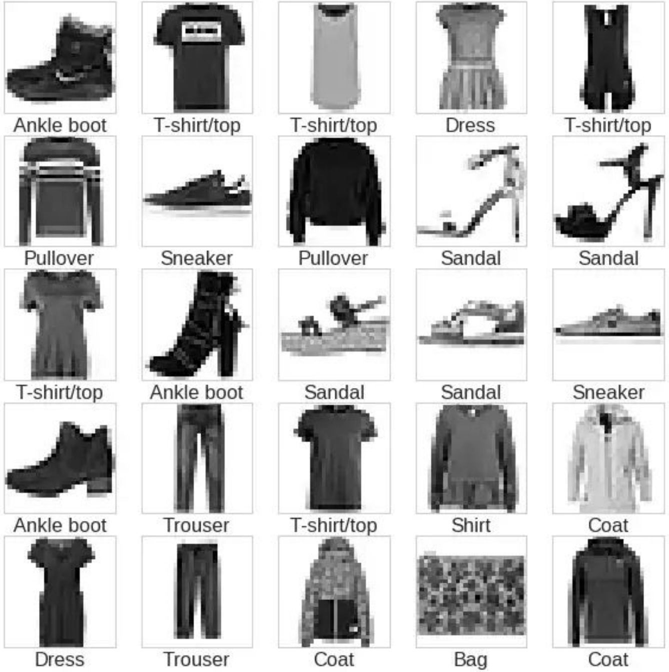


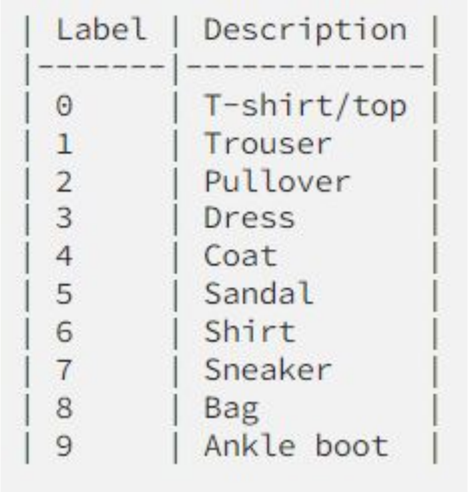


```
000000012222222222220000000000
000000012222222222220000000000
000001112222222222221111111000
000002222222222222222222222100
000002222222222222222222222100
000003333333466664336630000000
000003333333466664336630000000
000114433344566664336643333100
001336643366666664336666666300
001336643366666664336666666300
001336643333466666663346666666
001336643333466666663346666666
001334444444566665443334444333
001333346666666664333333333100
001333346666666664333333333100
000000036666666666666666600000
000000036666666666666666600000
000001134444444444443333300000
000003333322233333330000000000
000003333322233333330000000000
001333333322233332223333333100
001333333322233332223333333100
112333333322222222223333333211
333333333322222222223333333333
333333333322222222223333333333
666663322266422224662223366666
666663322266422224662223366666
666664432244322223442234466666
666666642222222222222246666666
666666642222222222222246666666
666662222222222222222222266666
666662222222222222222222266666
333332222222111111222222233333
000002222222100001222222200000
000002222222100001222222200000
001333333300000000003333333100
001333333300000000003333333100
112333333300000000003333333211
333333333300000000003333333333
333333333300000000003333333333
```

# Clothing Identification

- Dataset - Fashion MNIST
- 28 * 28 Pixels
- Grayscale (0 - 255)





| Label | Description |
|-------|-------------|
| 0 | T-shirt/top |
| 1 | Trouser |
| 2 | Pullover |
| 3 | Dress |
| 4 | Coat |
| 5 | Sandal |
| 6 | Shirt |
| 7 | Sneaker |
| 8 | Bag |
| 9 | Ankle boot |

# Practical Time

Link to Notebook

# Reminder!

## Deadline: 7th December

# Thank you for your attention!

Do you have any more questions? Join our Discord server