

Introduction to Deep Learning using Keras

FPRO 2018/2019

Pedro M. Ferreira
INESC TEC, PDEEC

10 January 2019

What is Machine Learning?

What is Machine Learning?

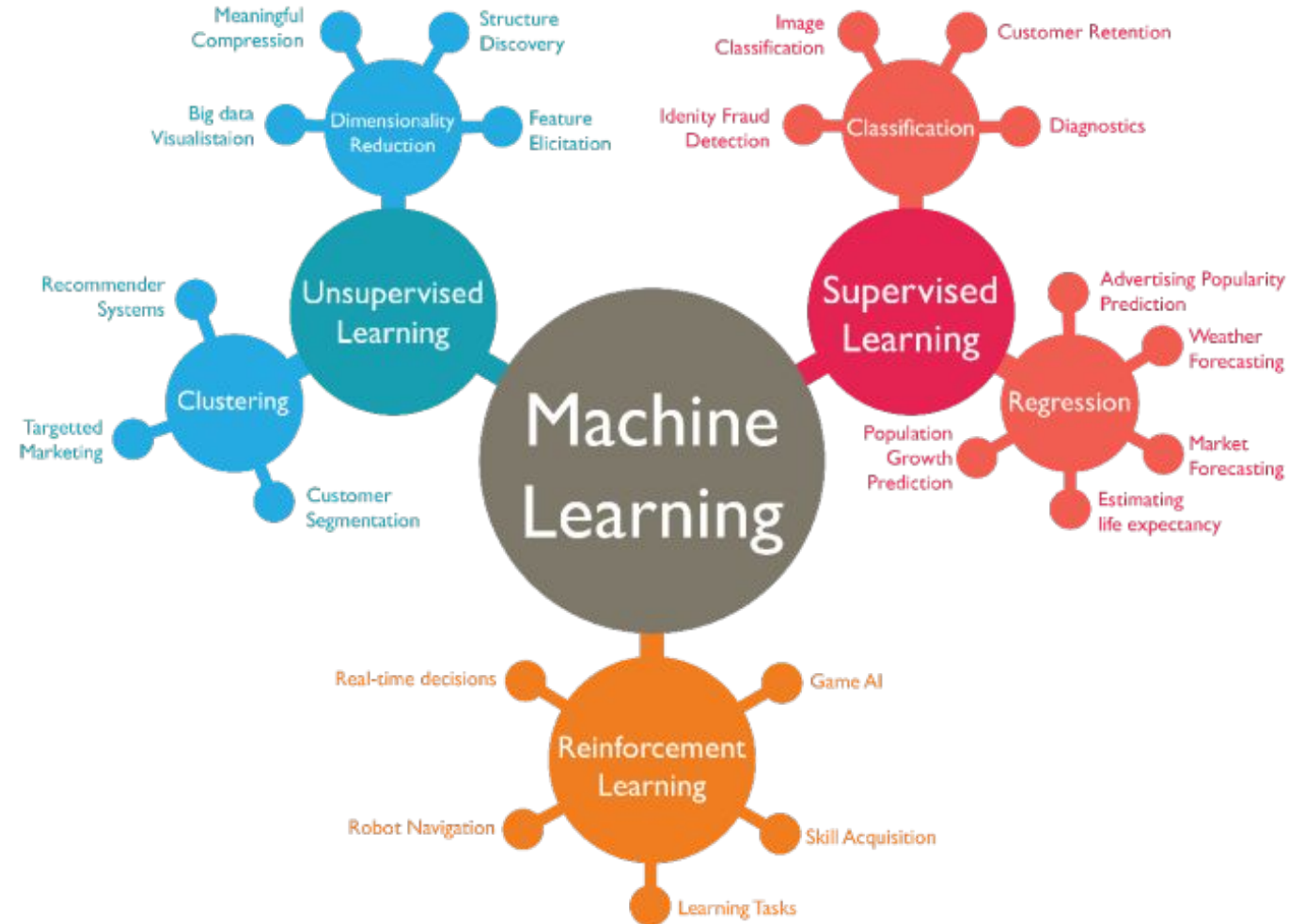
by Arthur Samuel, 1959

"the field of study that gives computers the ability to learn without being explicitly programmed."

by Tom M. Mitchell, 1997

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P , if its performance at tasks in T , as measured by P , improves with experience E ."

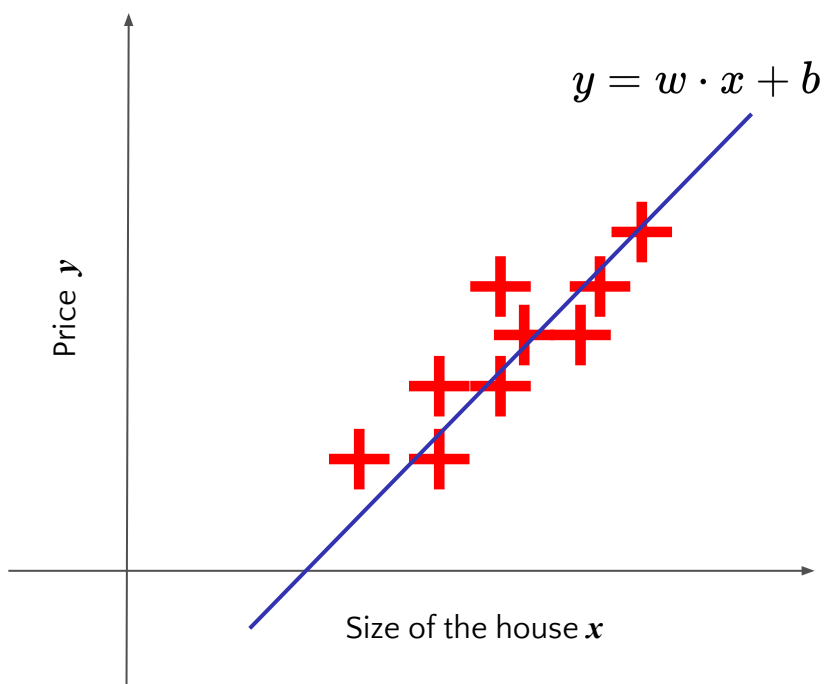
Machine Learning Problems



Deep Learning

What is a Neural Network?

[EXAMPLE] Housing Price Prediction

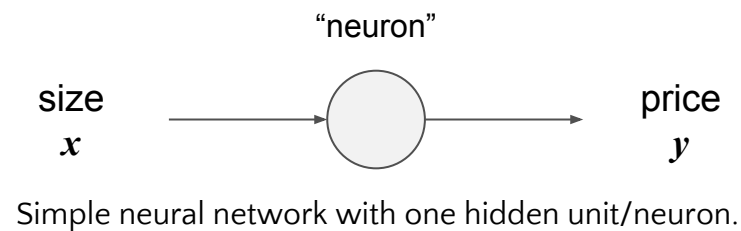


> Using **linear regression**, you can fit a linear function to predict the price y as a function of the size of the house x :

$$y = w \cdot x + b$$

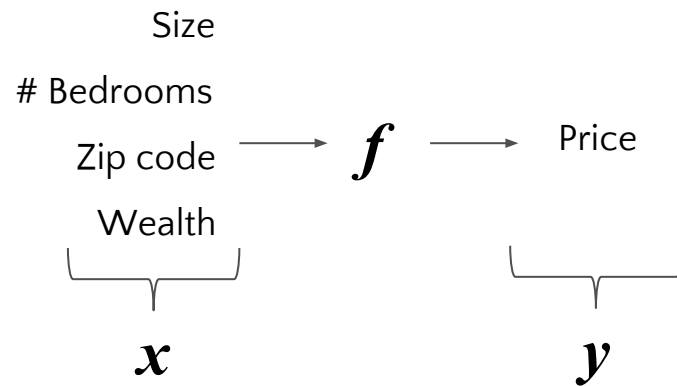
where w and b are learned from the data.

> You can think of this function as a **very simple neural network**!

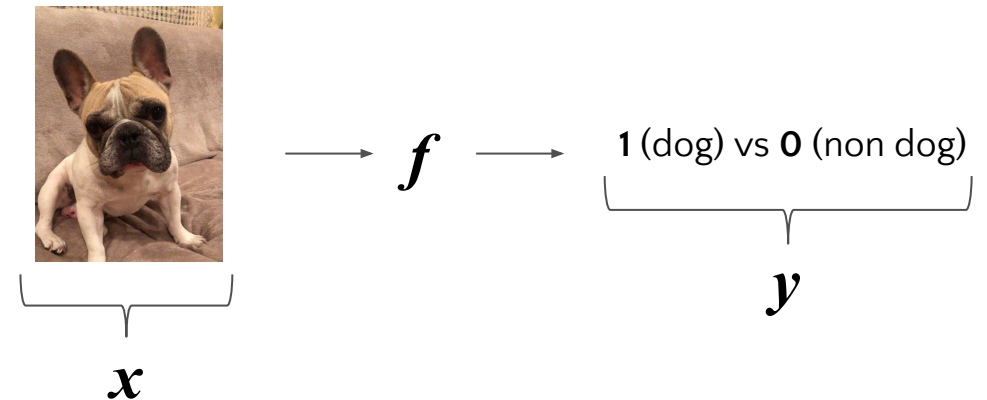


What is a Neural Network?

So, what type of functions f should we fit to the data when we don't know the relation/structure of the data?



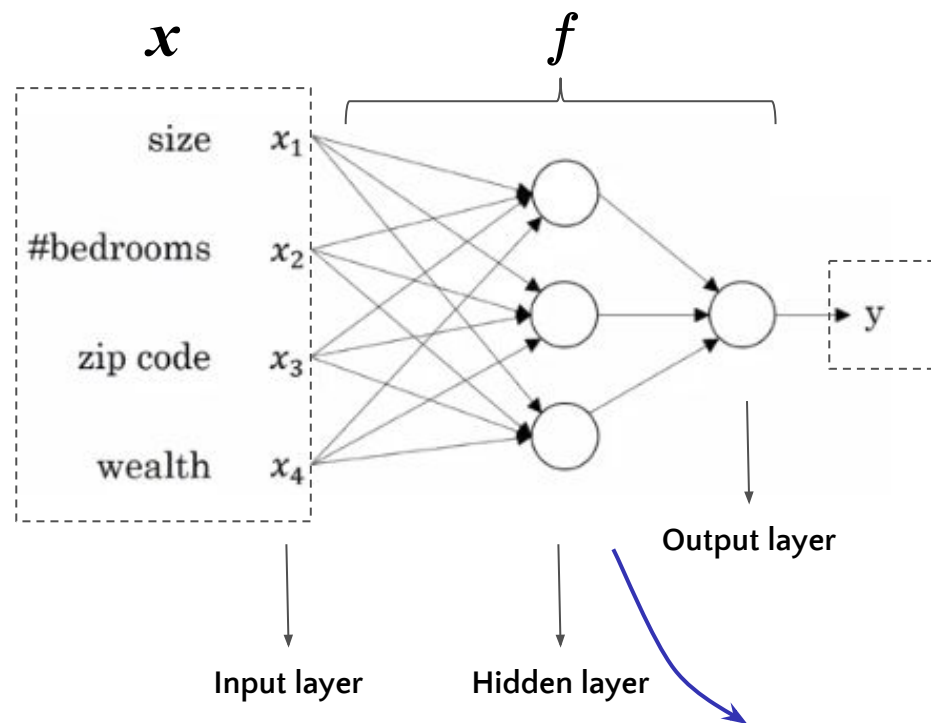
Housing Price Prediction



Binary image classification

What is a Neural Network?

So, what type of functions f should we fit to the data when we don't know the relation/structure of the data?



> Given enough training examples with both x (observations) and y (labels), **neural networks are remarkably good at figuring out functions f that accurately map from x to y !**

$$f : x \longrightarrow y$$

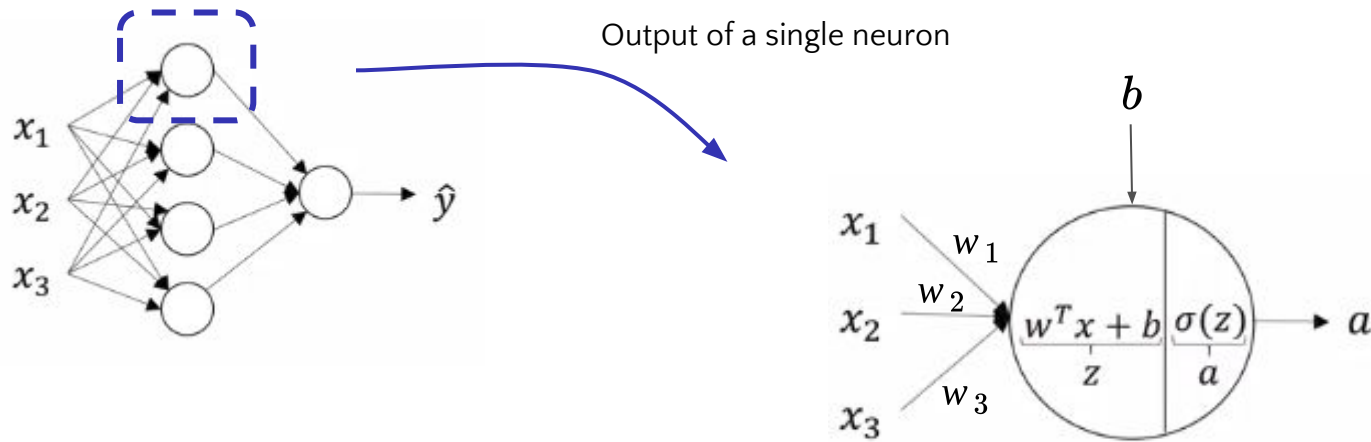
> **Building neural networks is analogous to Lego bricks:**

we take individual neurons and stack them together to create complex neural networks.

We talk in **deep learning** when we have a **neural net** with **two or more hidden layers!**

What is a Neural Network?

Neural Network Representation



$$z = w^T x + b$$

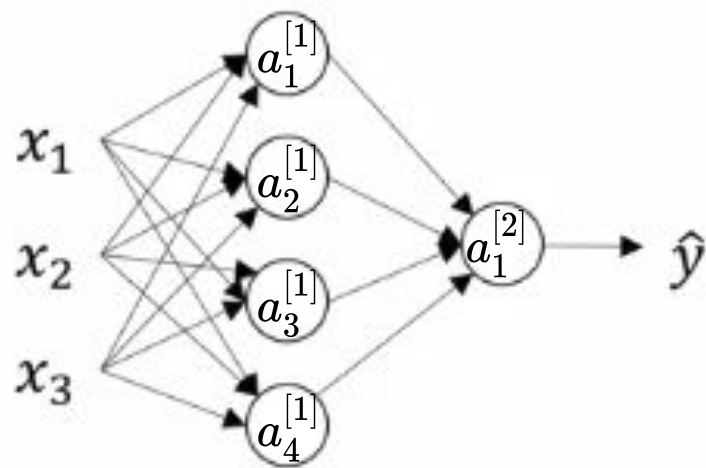
$$a = \sigma(z)$$

$$w^T = [w_1, w_2, w_3], x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Activation function/Non-linearity

What is a Neural Network?

Neural Network Representation



> Given X : $z^{[l]}$ and $a^{[l]}$ of layer l are given by:

$$z^{[1]} = W^{[1]}x + b^{[1]}$$

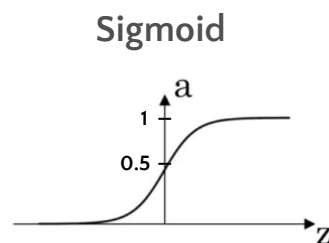
$$a^{[1]} = \sigma(z^{[1]})$$

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$$

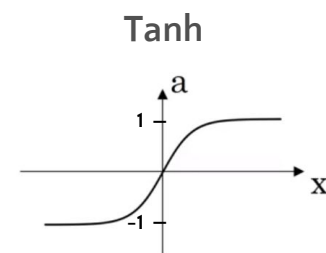
$$a^{[2]} = \sigma(z^{[2]})$$

$$\hat{y} = a^{[2]} \longrightarrow \text{Predicted output}$$

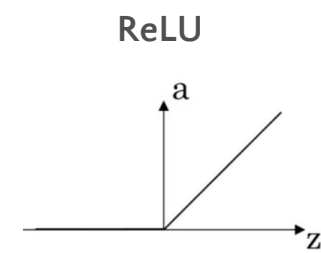
> Why non-linear activation functions are important?



$$a = \frac{1}{1+e^{-z}}$$



$$a = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

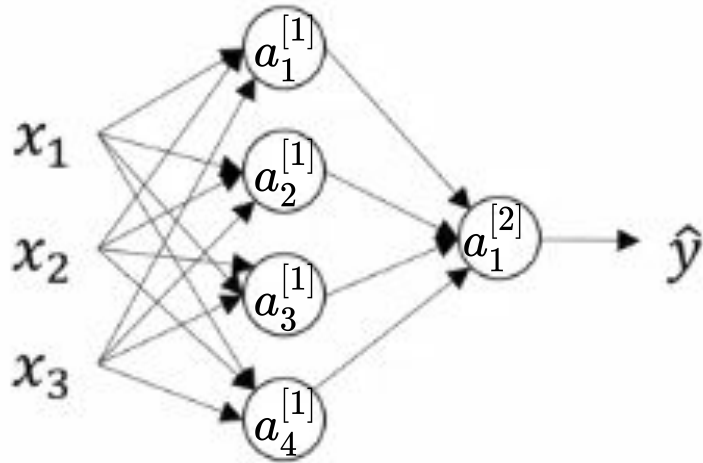


$$a = \max(0, z)$$

- Because the composition of two linear functions is still a linear function!
- Unless you throw a non-linear there, you are not computing more interesting functions as you go deeper in the network!

What is a Neural Network?

Neural Network Representation



> Given X : $z^{[l]}$ and $a^{[l]}$ of layer l are given by:

$$z^{[1]} = W^{[1]}x + b^{[1]}$$

$$a^{[1]} = \sigma(z^{[1]})$$

$$z^{[2]} = W^{[2]}a^{[1]} + b^{[2]}$$

$$a^{[2]} = \sigma(z^{[2]})$$

$$\hat{y} = a^{[2]} \longrightarrow \text{Predicted output}$$

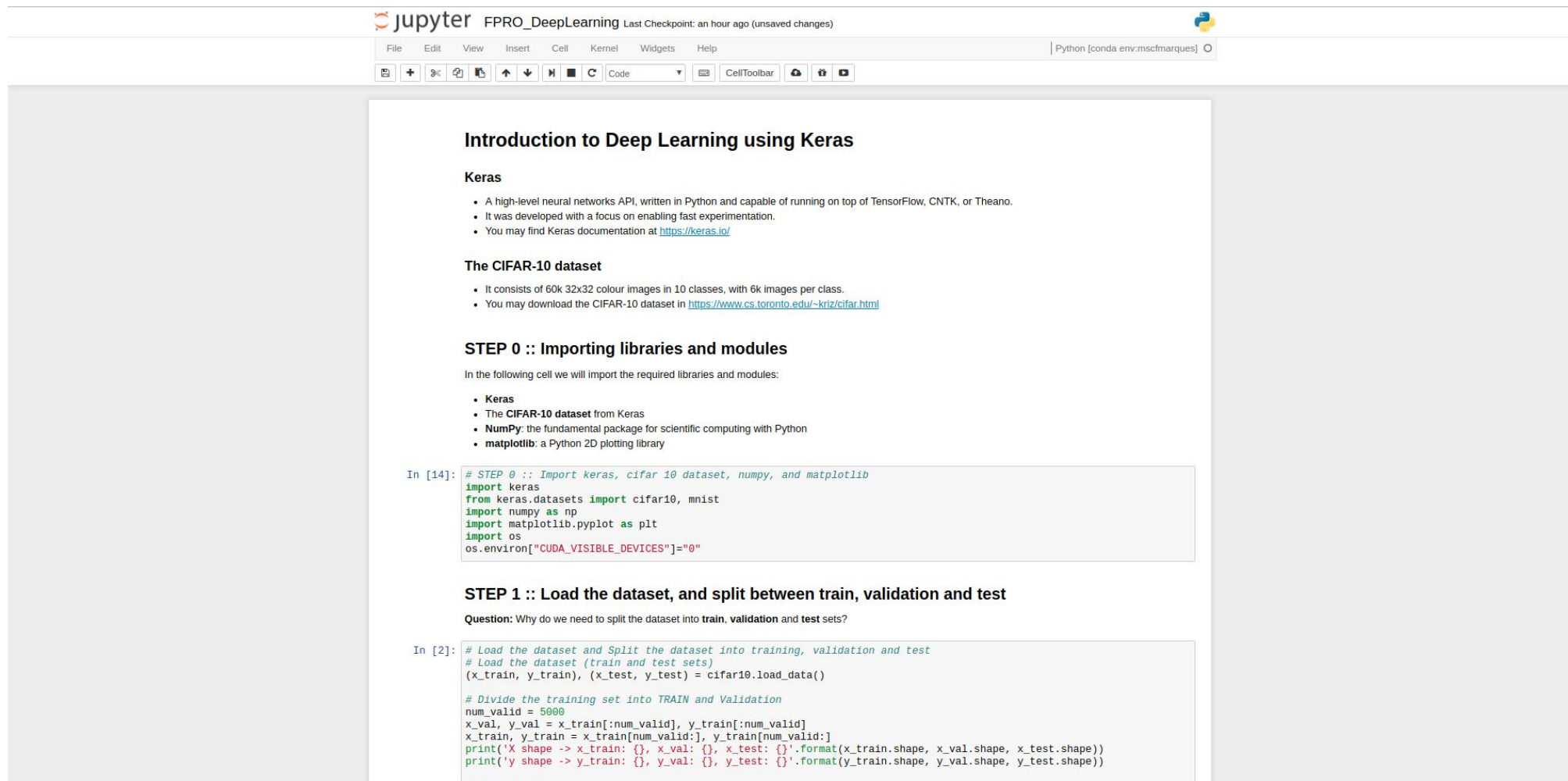
How can we learn the best possible parameters W , so that the function learned by the neural network accurately maps from x to y ?



This is called the **training/fitting process** of the neural network!

Implementing Neural Nets using Keras

Implementing Neural Networks



The screenshot shows a Jupyter Notebook interface with the title 'FPRO_DeepLearning' and a status bar indicating 'Last Checkpoint: an hour ago (unsaved changes)'. The notebook is running Python in a conda environment named 'mscsmarques'.

Introduction to Deep Learning using Keras

Keras

- A high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano.
- It was developed with a focus on enabling fast experimentation.
- You may find Keras documentation at <https://keras.io/>

The CIFAR-10 dataset

- It consists of 60k 32x32 colour images in 10 classes, with 6k images per class.
- You may download the CIFAR-10 dataset in <https://www.cs.toronto.edu/~kriz/cifar.html>

STEP 0 :: Importing libraries and modules

In the following cell we will import the required libraries and modules:

- **Keras**
- The **CIFAR-10 dataset** from Keras
- **NumPy**: the fundamental package for scientific computing with Python
- **matplotlib**: a Python 2D plotting library

```
In [14]: # STEP 0 :: Import keras, cifar 10 dataset, numpy, and matplotlib
import keras
from keras.datasets import cifar10, mnist
import numpy as np
import matplotlib.pyplot as plt
import os
os.environ["CUDA_VISIBLE_DEVICES"]="0"
```

STEP 1 :: Load the dataset, and split between train, validation and test

Question: Why do we need to split the dataset into **train**, **validation** and **test** sets?

```
In [2]: # Load the dataset and Split the dataset into training, validation and test
# Load the dataset (train and test sets)
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

# Divide the training set into TRAIN and Validation
num_valid = 5000
x_val, y_val = x_train[num_valid:], y_train[num_valid:]
x_train, y_train = x_train[:num_valid], y_train[:num_valid]
print('X shape -> x_train: {}, x_val: {}, x_test: {}'.format(x_train.shape, x_val.shape, x_test.shape))
print('y shape -> y_train: {}, y_val: {}, y_test: {}'.format(y_train.shape, y_val.shape, y_test.shape))
```

EXERCISES

- Moodle activity at: [LE26: Deep Learning](#)