Tensorflow Course

Alket Cecaj Ph.D.

Outline

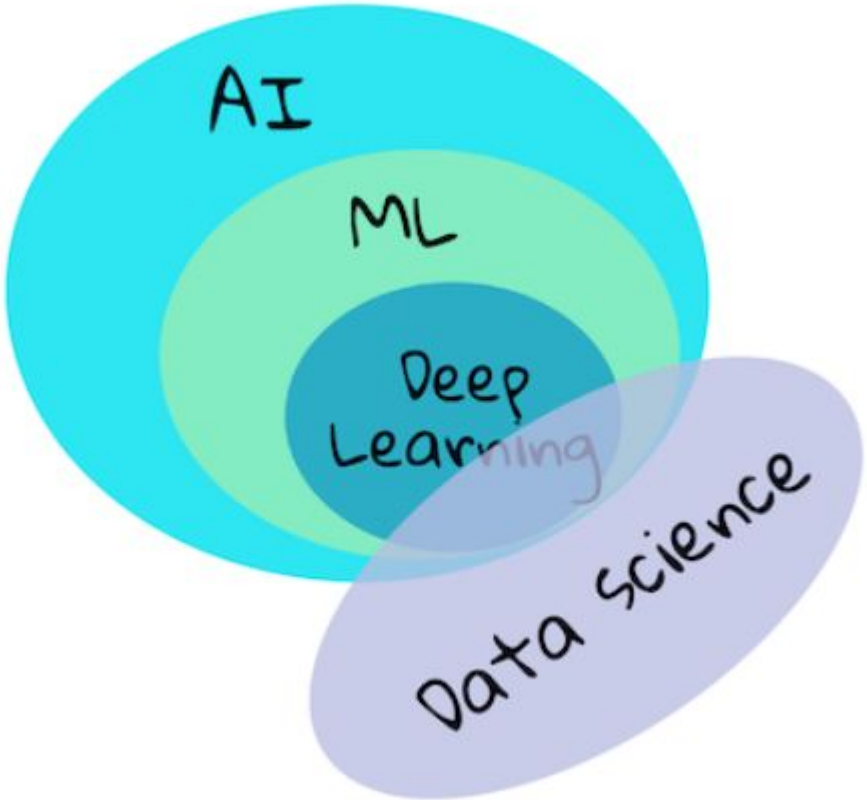
- Introduction to Machine Learning
- Introduction to Neural Networks and Tensorflow
- Predictive Modelling in Tensorflow: Regression and Classification problems.
- Tensorflow for Time Series Forecasting
- Computer Vision with Tensorflow
- Tensorflow for Natural Language Processing
- Transfer Learning with Tensorflow - TensorHub.
- Deployment of a Neural Network using Streamlit.

Introduction to Machine Learning

Orientation about terms

- In classic terms, machine learning is a type of **artificial intelligence** that enables self-learning from data and then applies that learning without the need for human intervention.

- In actuality, there are many different types of machine learning, as well as many strategies of how to best employ them.



History of Machine Learning

AI should be elegant → Neats vs. scruffies

"Artificial intelligence"

Shakey the Robot

DOES NOT COMPUTE

too complicated!

Hey Siri!

GO!

Need more power

CPU

1950 1956 1966 1970 1980 2000

"Machines that think"

Dartmouth Summer Research Proj.

HELLO, I AM ELIZA!

Blocks world

AI winter

Deep Learning

Big data

Alan Turing

Turing test

NLP

Natural Language Processing

THREE BLOCKS

"SHRDLU"

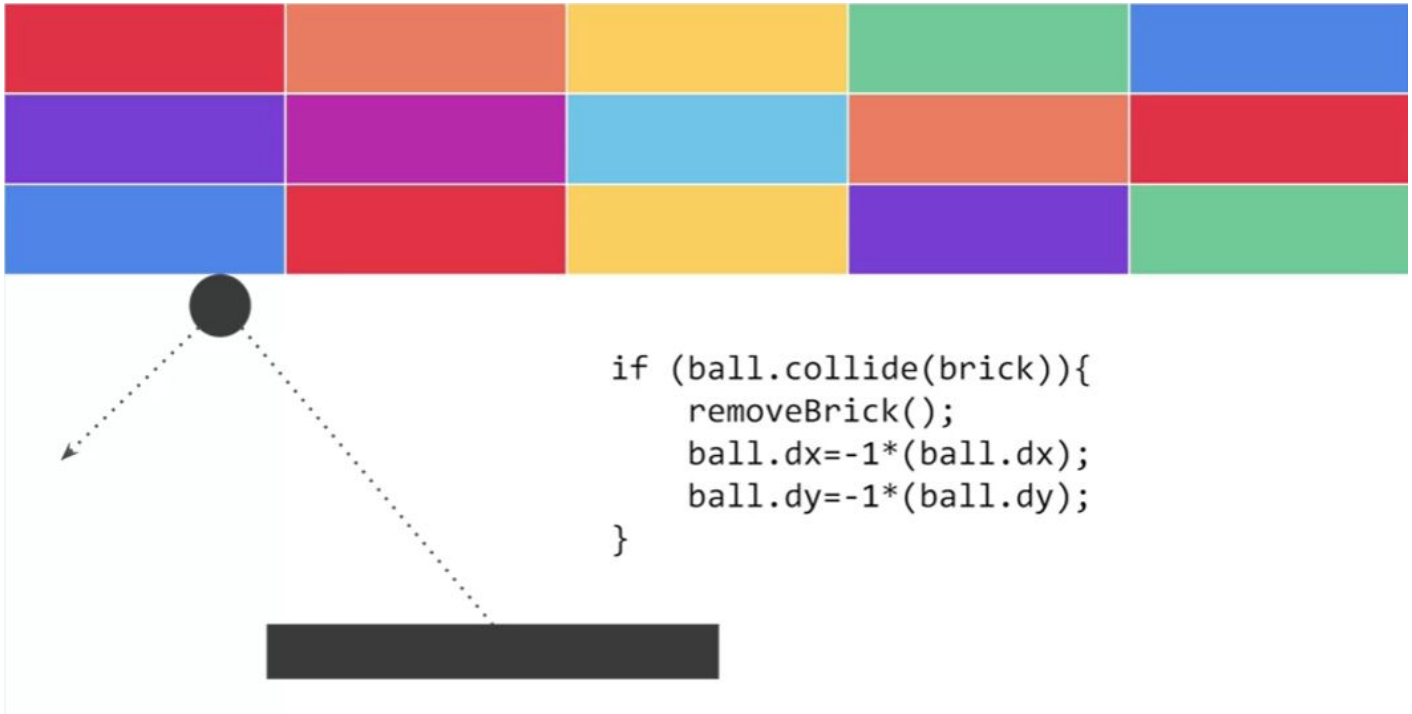
$2000 Ken

$19436 Watson

@AzureAdvocates
@girlie_mac

Traditional Computer Programming

Traditional Computer Programming - Video Games



```
if (ball.collide(brick)){
    removeBrick();
    ball.dx=-1*(ball.dx);
    ball.dy=-1*(ball.dy);
}
```

Traditional Computer Programming - Activity Recognition

```
if(speed<4){
    status=WALKING;
}
```

```
if(speed<4){
    status=WALKING;
} else {
    status=RUNNING;
}
```

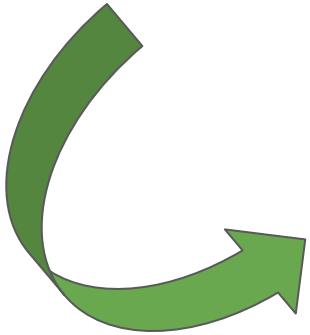
```
if(speed<4){
    status=WALKING;
} else if(speed<12){
    status=RUNNING;
} else {
    status=BIKING;
}
```

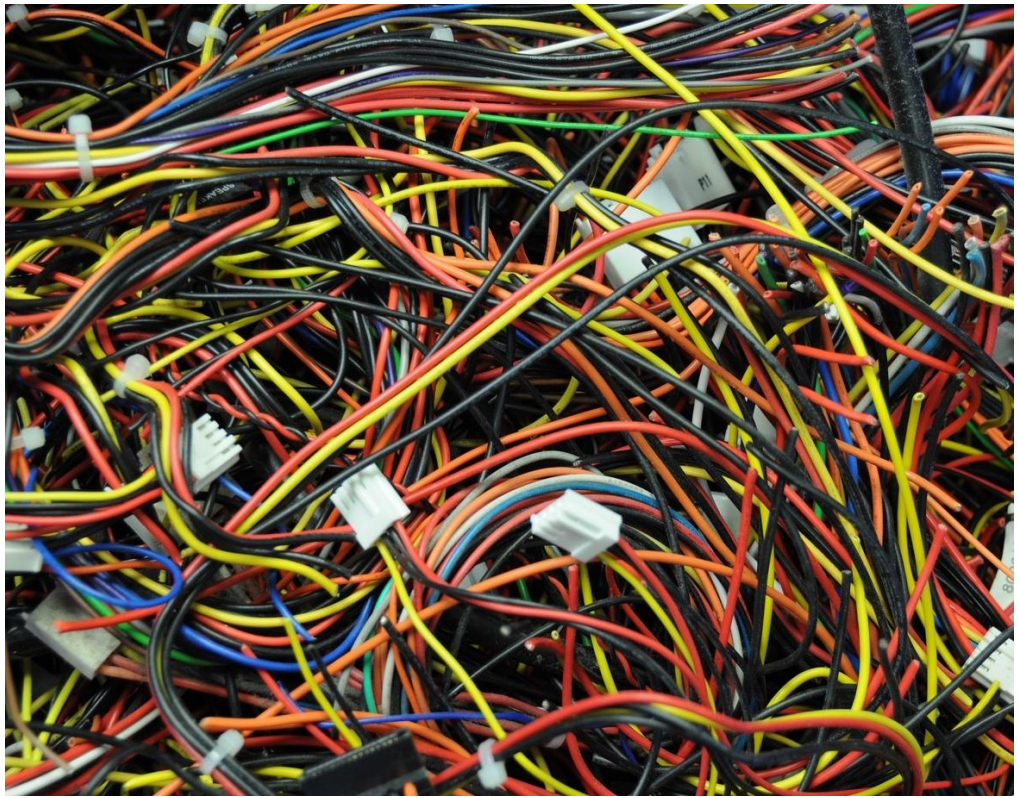
```
// Oh crap
```

Which wire you want me to cut?

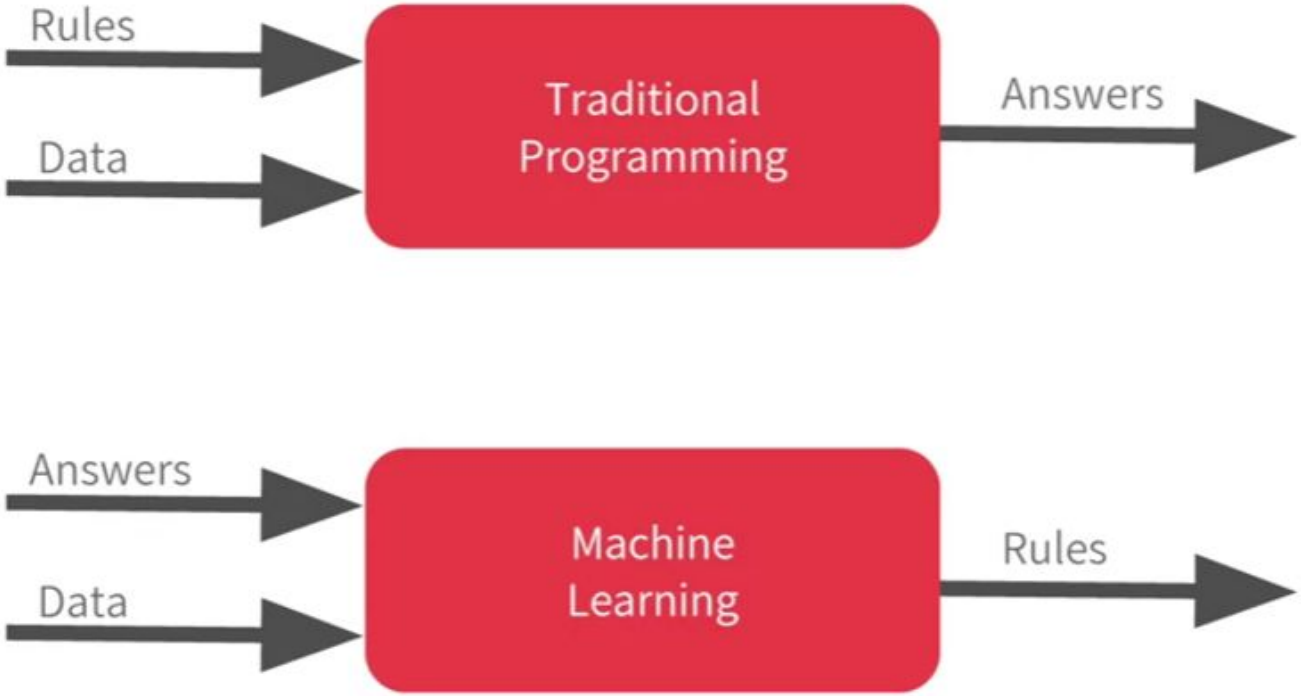
Red or Blue?

Why not the green one?!





Too many rules call for a paradigm shift



What does it mean to give data and answers to an algorithm?

Collect data while person is doing one of those activities and label it accordingly.

```
01010010101001010101
10010101010010111101
01001010100101010010
01010010101001010101
```

Label = WALKING

```
10101001010010101010
01010100100100100001
00100111110101011111
10101001001111010111
```

Label = RUNNING

```
10010100111110101010
11010101110101011110
10101011110101010111
11111100011110101010
```

Label = BIKING

```
11111111110100111010
00111111010111110101
01011101010101011110
10101010101001111110
```

Label = GOLFING

This paradigm shift is shown in two major events of the last 20 years.

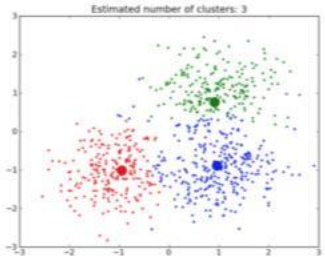
1996 - Deep Blue VS Kasparov

2016 - Lee Sedol VS AlphaGo

Machine Learning: Supervised and Unsupervised Learning

Unsupervised learning: is about finding possible groups and the hidden structure of the data

- Clustering algorithms: k-means, DBSCAN,
- Anomaly detection: local outlier factor
- PCA (Principal Component Analysis)



Supervised learning: leaning a function that maps and input to an output based on examples

- Regression : linear/logistic regression
- Trees: decision trees, random forest
- Naive Bayes: bayesian classifier
- Neural Networks: Perceptrons

How to Apply a Machine Learning Algorithm

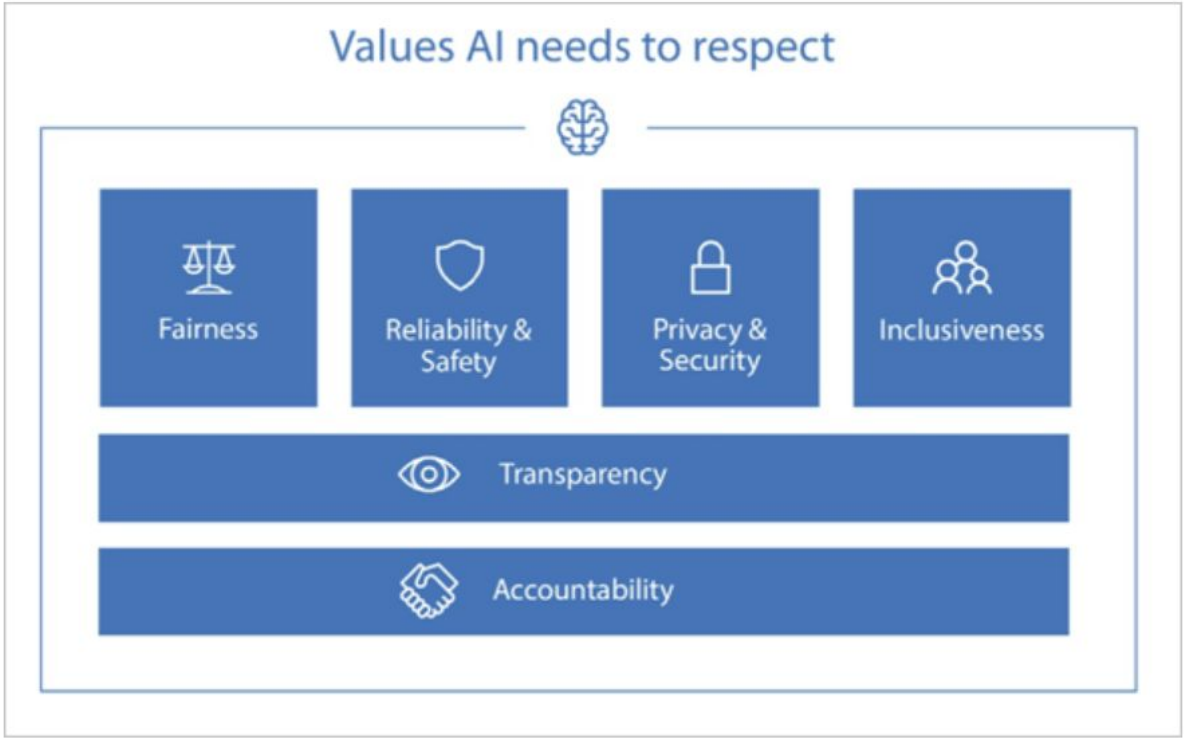
- Feature Engineering: prepare the data for the ML Algorithm
- Attention of data leakage during data transformation
- Train-Test split the data: prepare training set and test set.
- Measure the performance:
  - Is it a classification problem? → accuracy, precision and recall.
  - Is it a regression problem : MAE, RMSE or MAPE?
- Consider cross-validation to estimate the bias and variance of the model

Applications of Machine Learning

Finance, economics, earth science, space exploration, biomedical engineering, cognitive science, and even fields in the humanities have adapted machine learning to solve the arduous, data-processing heavy problems of their domain.

- To predict the likelihood of disease from a patient's medical history or reports.
- Speech Recognition: “Alexa turn on the light”
- To leverage weather data to predict weather events and manage emergency.
- To understand the sentiment of a text detect and stop hate speech.
- To detect fake news to stop the spread of propaganda.
- Satellite imagery and deep learning for crop yield prediction..
- Pay attention to human bias in the data!

Machine Learning & Human Ethics



Some important points to deal with in Machine Learning / Deep Learning models.

- **Parameter set optimization**: what are the best parameters for my model?
- **Overfitting**: did my model learn like a parrot?
- **Validation and testing**: how can I be sure that my model performs well?
- **Performance Measures**: the accuracy and the MAE.
- **Bias and Variance of a Machine Learning Model**: Why is my model giving me a different result each time?
- **Cross-validation** with K-fold dataset.
- **Data Leakage**: did I split the data without leaking anything?
- **Data Drift**: my data just changed format, did my model understand it?
- **Model Drift**: related to the previous point, data changes, models should change as well.
- **Interpretability** of the results in neural networks.

Tools and Software for Machine Learning Implementation.

- Programming languages: Python, Julia, R programming

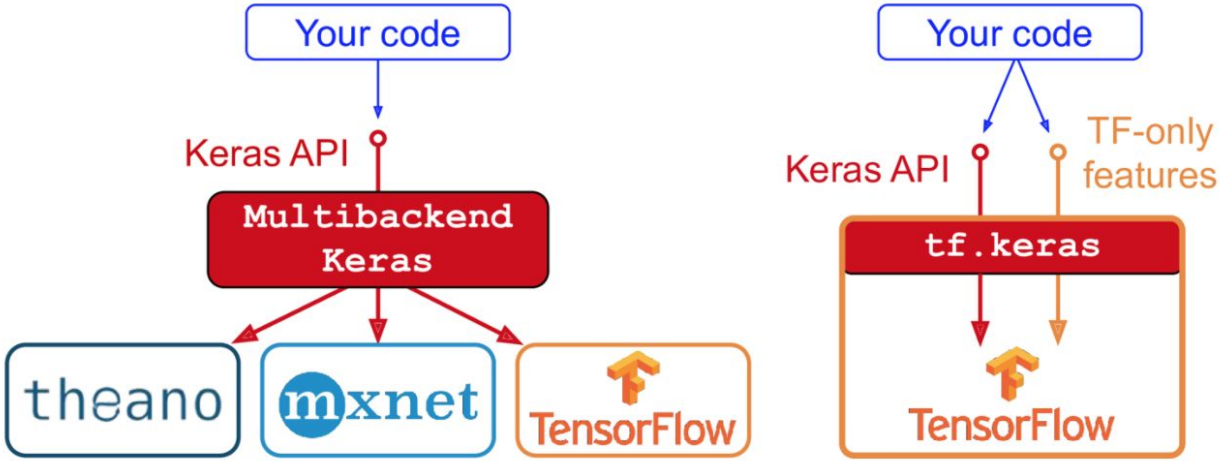
- GUI Software for data mining : KNIME, RapidMiner, Orange

Python and the Machine Learning / Data Science stack

- Anaconda
- Numpy
- Pandas
- Tensorflow
- Matplotlib and …
- Scikit-learn

The Implementation Part - Tensorflow and Keras

- TensorFlow is a powerful open source software library developed by the Google Brain team for deep neural networks.
- Its GitHub repository has more than 55,000 commits, with roughly 2,130 contributors. This in itself provides a measure of the popularity of TensorFlow.
- Tensorflow comes with a Keras interface that makes developer's life easier.



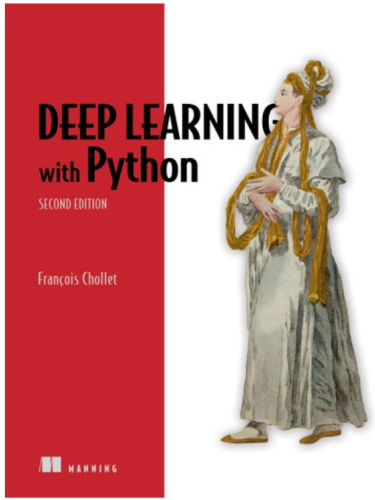
Recommended Books - Tensorflow & Keras

- Tensorflow implementations with Keras API.
- Easy to start with experimenting hands on.
- Insights for both novice and experienced machine learning practitioners.
- Lots of Jupyter notebooks with Python code included.
- Many examples of real world applications.

Deep Learning with Python, Second Edition

★★★★★ 1 review

By Francois Chollet



TIME TO COMPLETE:
15h 41m

TOPICS:
Deep Learning

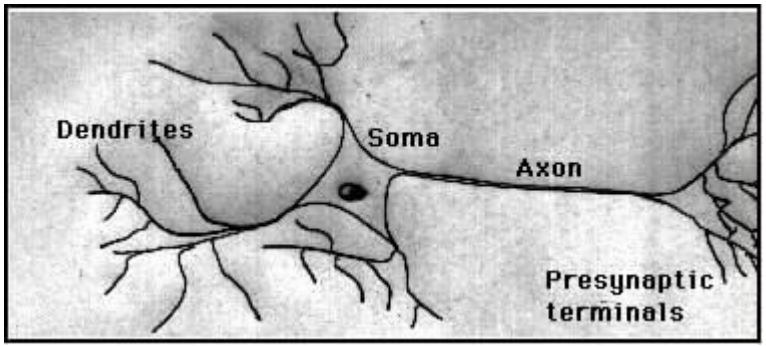
PUBLISHED BY:
Manning Publications

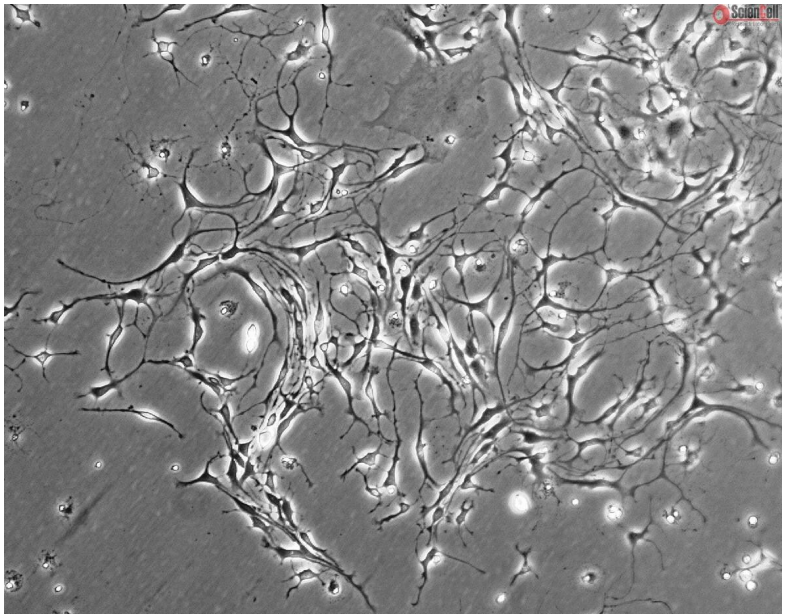
PUBLICATION DATE:
November 2021

PRINT LENGTH:
504 pages

Introduction to Neural Networks and Tensorflow

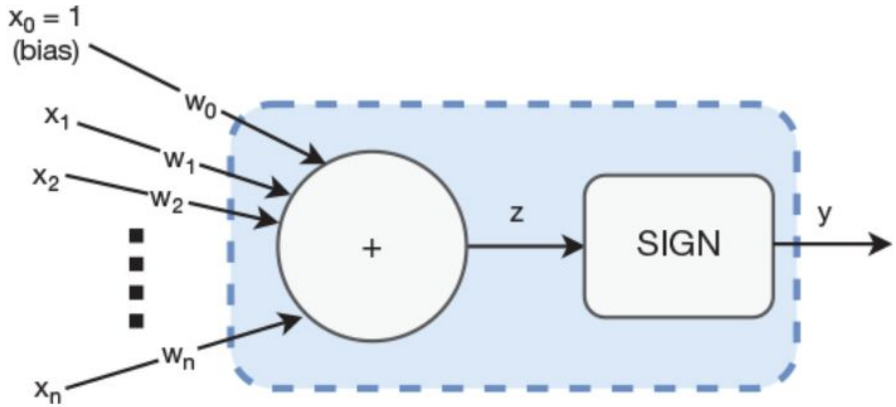
Biological Neural Networks

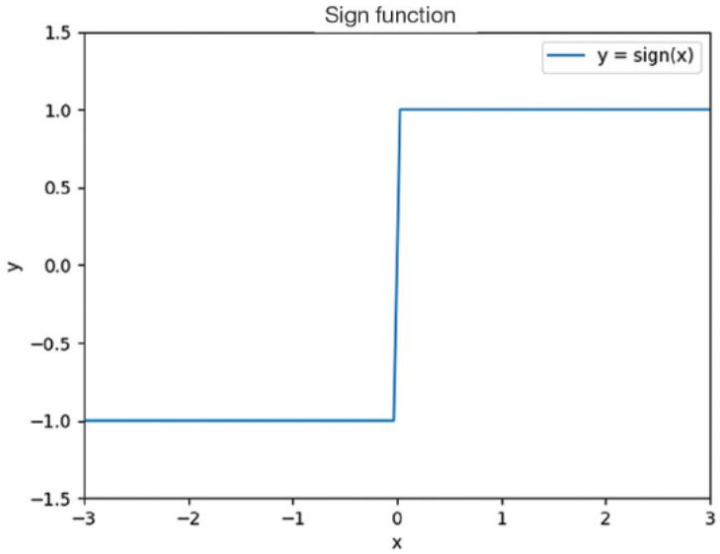




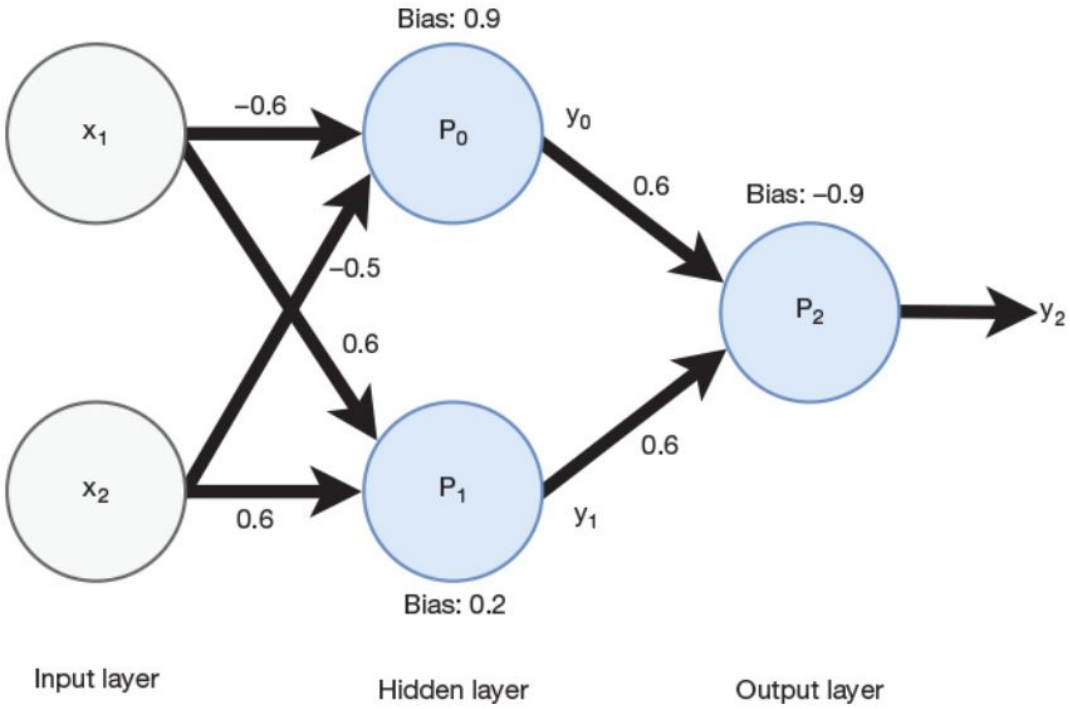
The Neuron - Computer Science Representation

Two functions: the main function and the activation function.

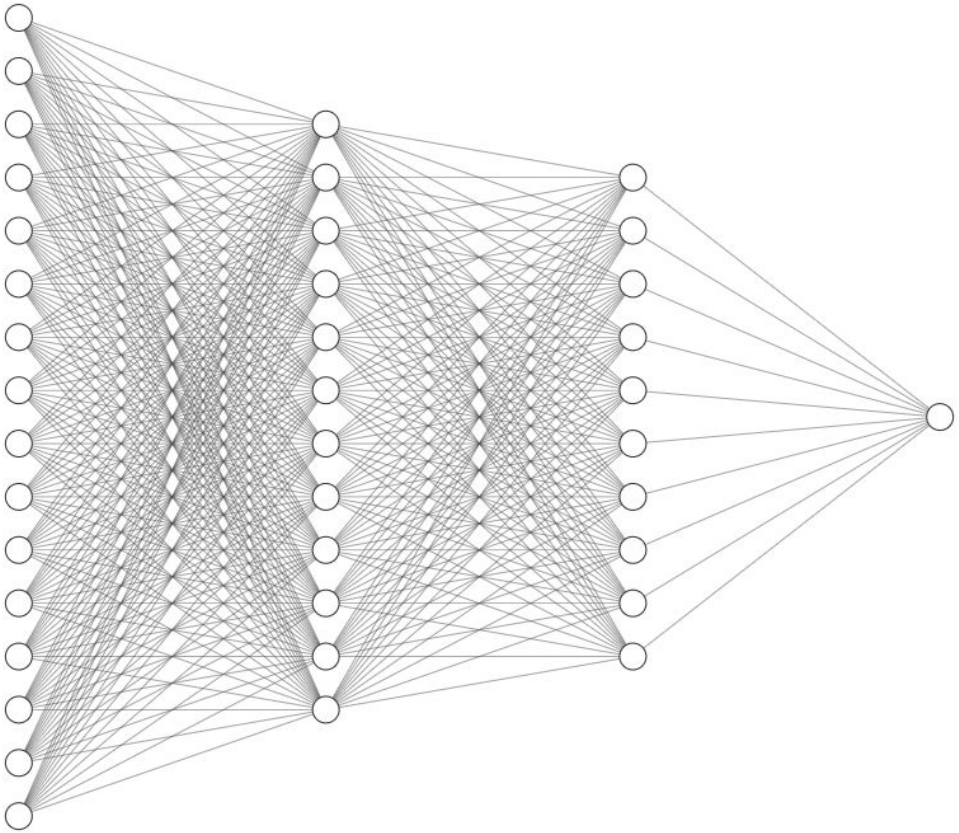




Neural Networks - The Computer Science Representation



Deep Neural Networks - NN with more layers and neurons



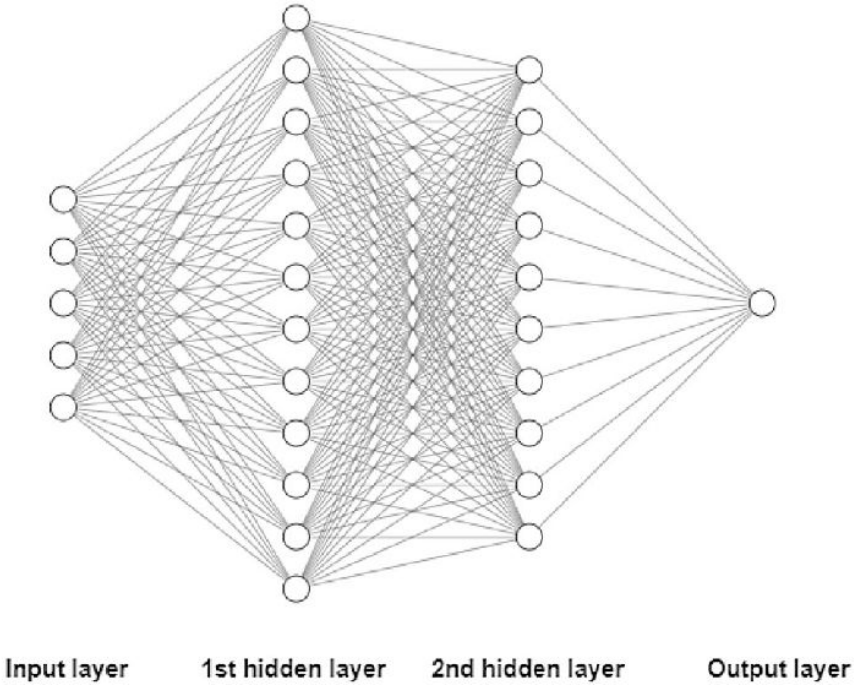
Types of Neural Networks

- Multilayer Perceptrons Neural Networks or MLP → General purpose.
- Convolutional Neural Networks or CNN → Work with images.
- Recurrent Neural Networks or LSTM → Good at mapping sequences.
- Generative Neural Networks (GAN) → Generative output (Deep Fakes)

MLP Multilayer Perceptron Networks

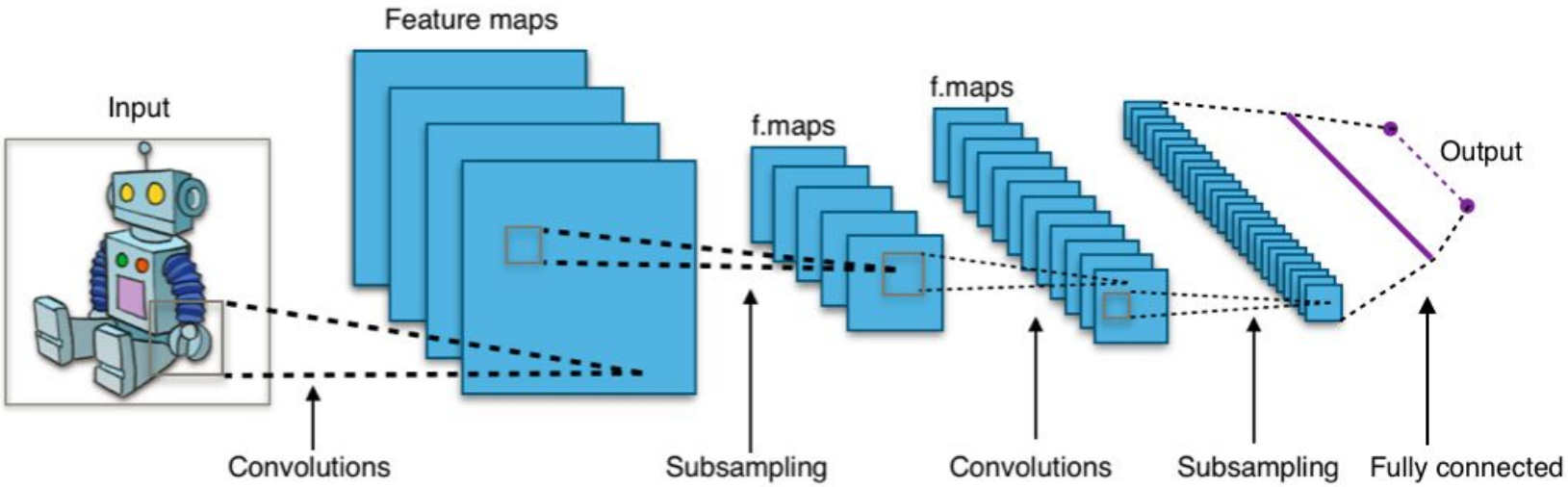
A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN).

The term MLP is used ambiguously, sometimes loosely to mean any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptrons.

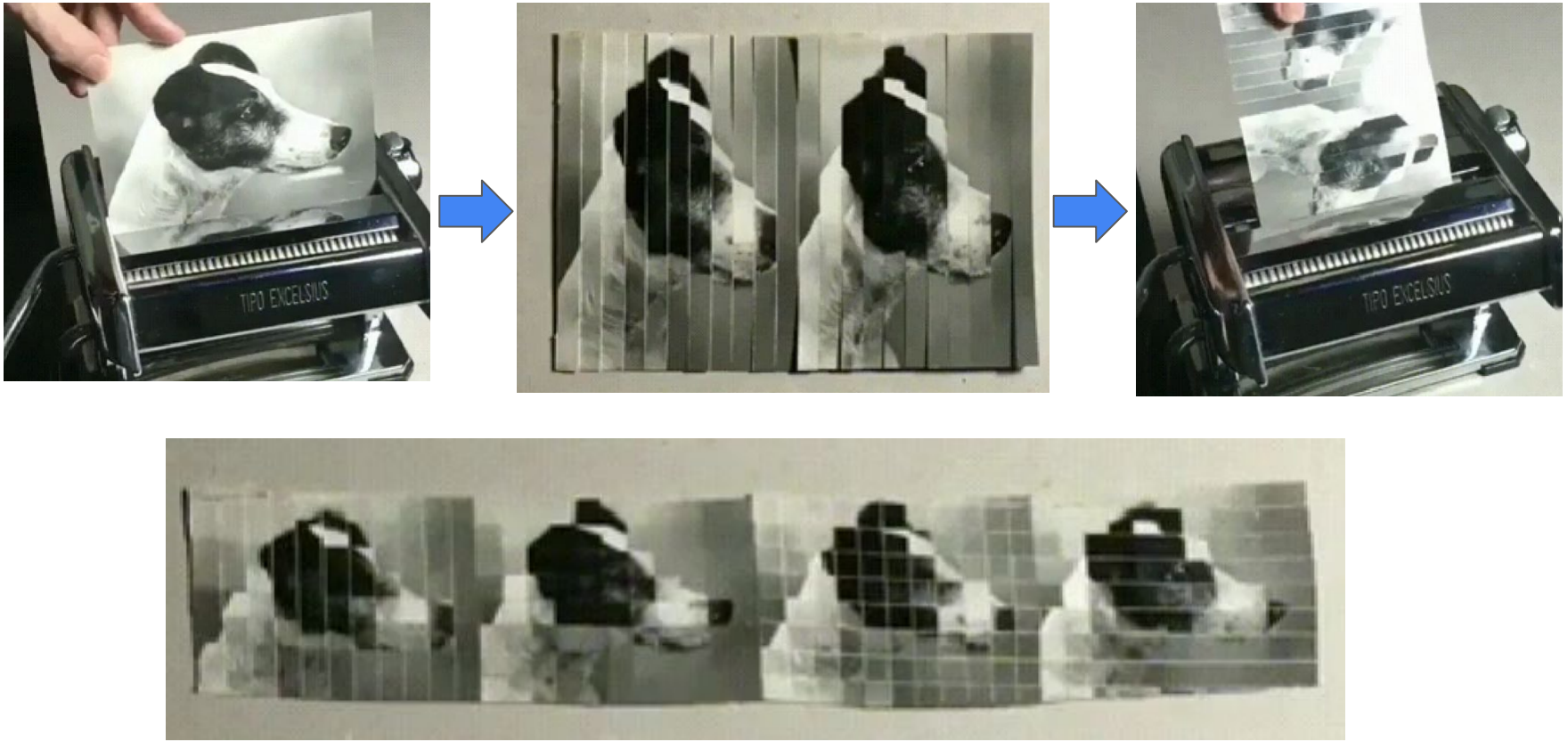


CNN - Convolutional Neural Networks

Composed of convolution kernels or filters that slide along input features and provide translation equivariant responses known as feature maps.

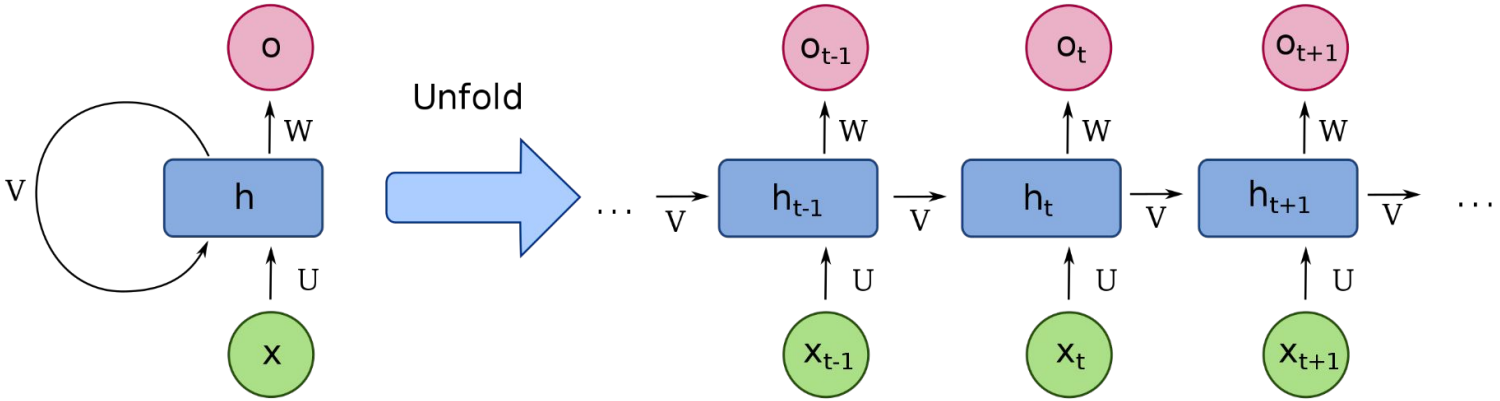


How the CNNs use Convolutions to process Images

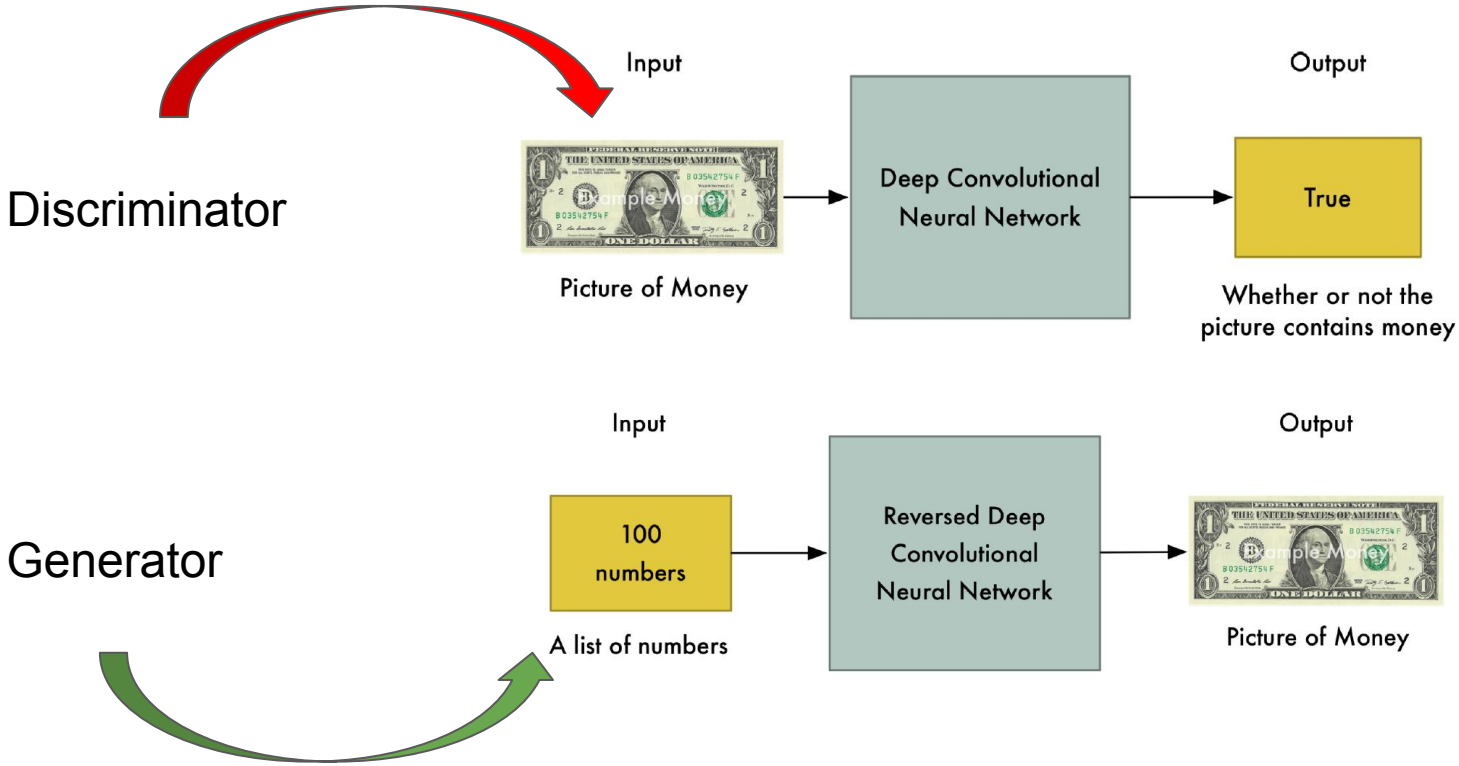


RNN - Recurrent Neural Networks

An RNN is a class of artificial neural networks where connections between nodes form a directed or undirected graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feedforward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs

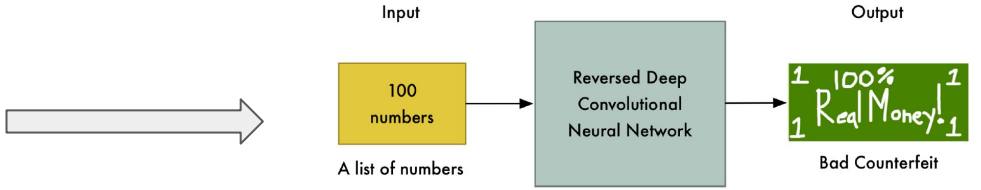


Generative Adversarial Networks - GANs

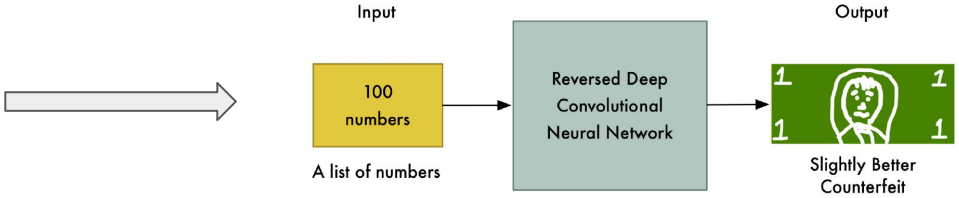


Generative Adversarial Networks - Improve as they go.

First interaction



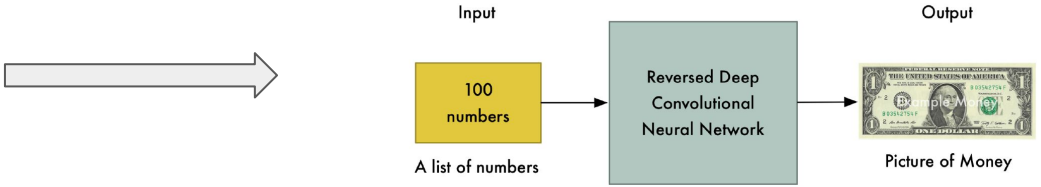
Second interaction



.....

.....

Final result



Example of Generative NN or GANs

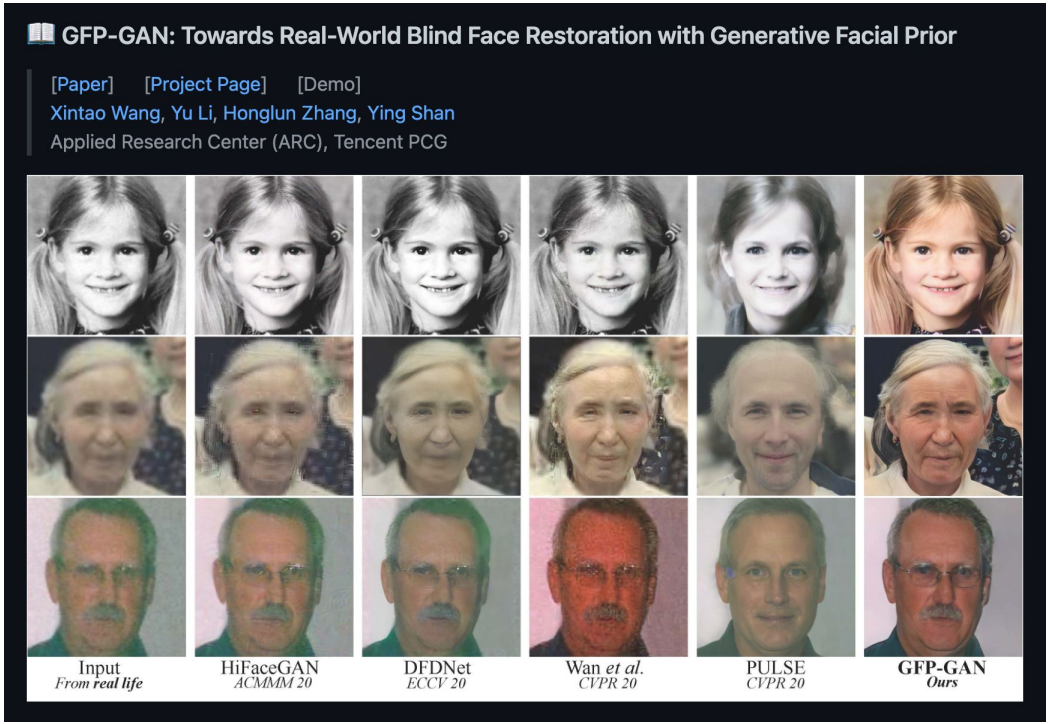
A generative adversarial network (GAN) is a class of machine learning frameworks designed by Ian Goodfellow and his colleagues in June 2014.

Two neural networks contest with each other in a game (in the form of a zero-sum game, where one agent's gain is another agent's loss).

Deep Convolutional Generative Adversarial Networks.

https://thispersondoesnotexist.com/

Useful Application of GAN - Restore Old Pictures



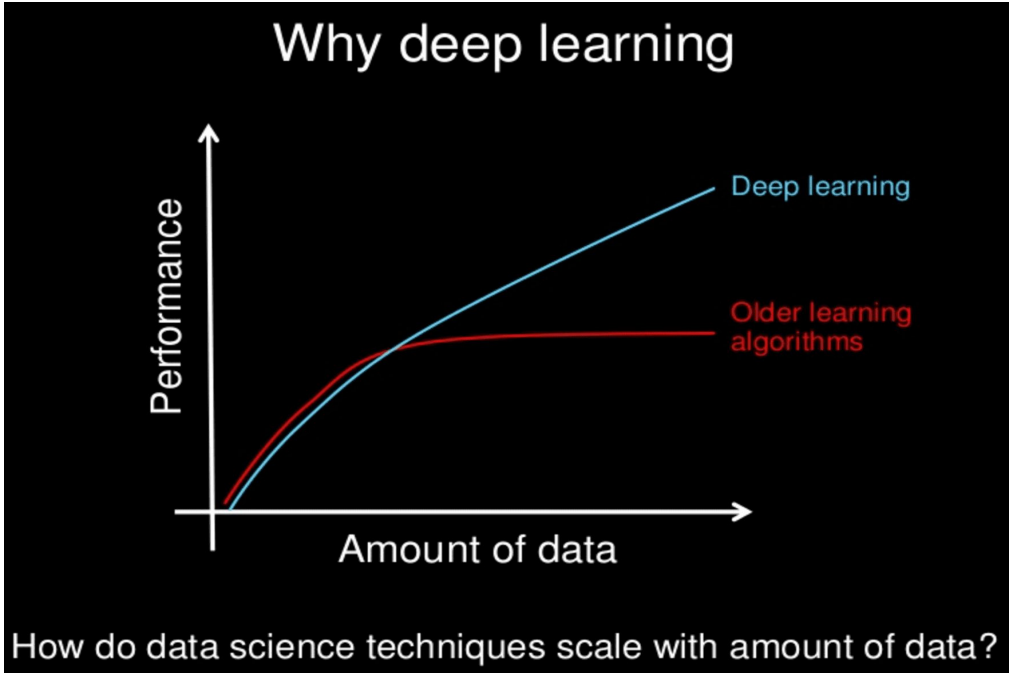
Is Machine Learning only About Neural Networks?

Not really! There are many other machine learning methods like:

1. Linear and Logistic Regression
2. Decision Trees (C4.5 and C5.0) and Random Forests
3. AdaBoost and XGBoost Algorithms
4. K-NearestNeighbors (kNN)
5. Support Vector Machines
6. Self-Organizing Map (SOM)
7. Naive Bayes and Gaussian Algorithms
8. Apriori and Eclat algorithms

Advantages of Neural Networks Over Older ML Algorithms

One of the main advantages of NN is a better capability to capture the complexity in the data .



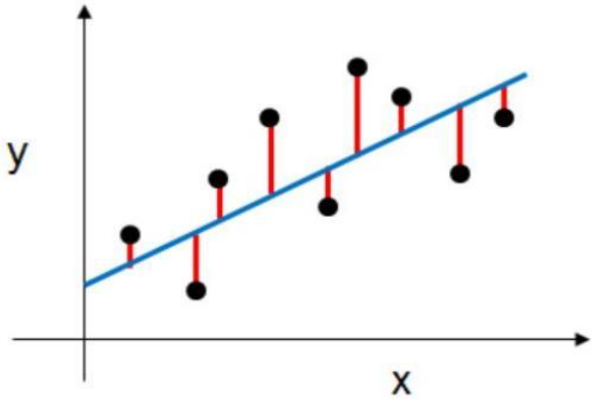
Predictive Modelling with TensorFlow: Regression and Classification Problems.

Predictive Modelling in Machine Learning/Deep Learning

- Regression Problems → Predict what is the value (number).
- Classification Problems. → Predict what is the classification (class).
- Both are part of predictive modeling: the problem of developing a model using historical data to make a prediction on new data where we do not have the answer.
- Predictive modeling can be described as the mathematical problem of approximating a mapping function (f) from input variables (X) to output variables (y). This is called the problem of function approximation.

$$f(X) = y$$

Explaining Linear Regression Intuitively



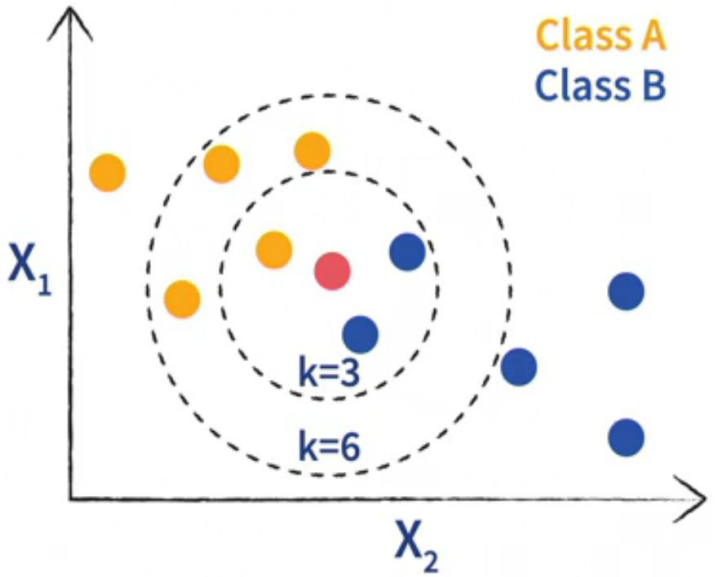
Model Prediction

$$SS_{residuals} = \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$$

Observed Result

Intuitive Explanation of a Classification Problem

- A classification problem requires that examples be classified into one of two or more classes.
- A classification can have real-valued or discrete input variables.
- A problem with two classes is often called a two-class or binary classification problem.
- A problem with more than two classes is often called a multi-class classification problem.
- A problem where an example is assigned multiple classes is called a multi-label classification problem.

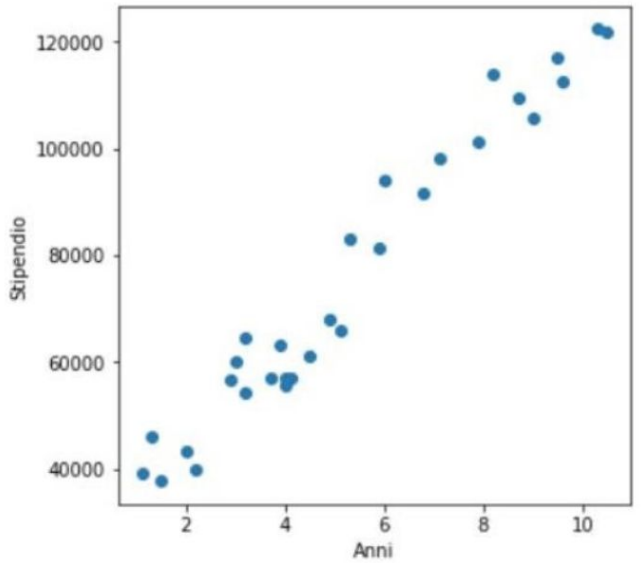


Tensorflow for modelling a Regression Problem.

The data model the relationship between years of experience and salary.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

dataset = pd.read_csv('../datasets/AnniStipendio.csv')
dataset.head()
```

	Anni_esp	Stipendio
0	1.1	39343.0
1	1.3	46205.0
2	1.5	37731.0
3	2.0	43525.0
4	2.2	39891.0



A First Implementation in Tensorflow Of a Regression Problem.

Use a Neural Network that can predict the salary based on the number of years of experience that a person has.

```
# Instantiate a Sequential model
model = tf.keras.models.Sequential()

# Add three Dense layers with a given number of neurons and relu activation function
model.add(tf.keras.layers.Dense(500, input_shape=(1,),activation='relu'))
model.add(tf.keras.layers.Dense(100, input_shape=(1,),activation='relu'))
model.add(tf.keras.layers.Dense(50, input_shape=(1,),activation='relu'))

# End your model with a Dense layer and no activation
model.add(tf.keras.layers.Dense(1))
```

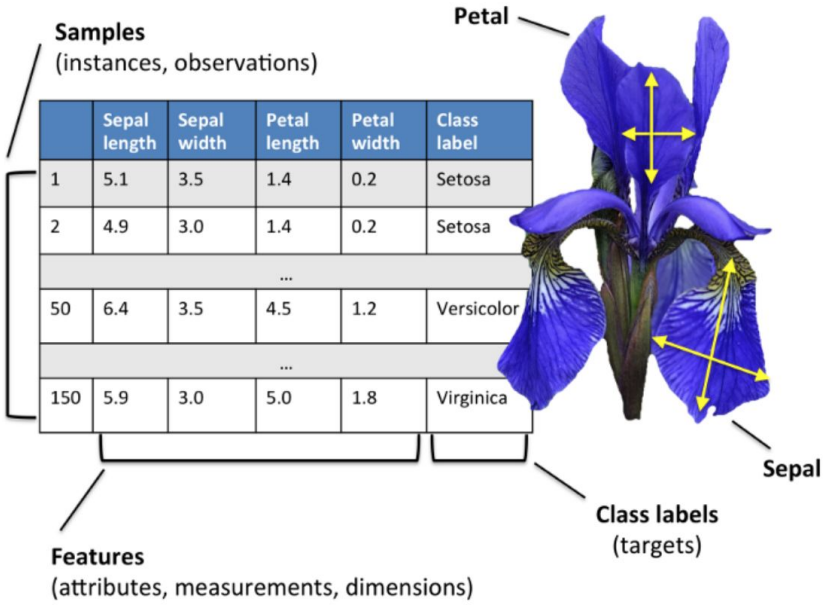
```
# Compile your model
model.compile(optimizer = 'adam', loss = 'mse')

# Fit your model on your data for 30 epochs
history = model.fit(anni_esperienza, salario, epochs = 500)
```

```
Training started..., this can take a while:
Epoch 1/500
1/1 [==============================] - 0s 459ms/step - loss: 64837448.0000
```

Tensorflow for modelling a Classification Problem.

- Iris dataset: 50 samples from each of three species of Iris (Iris setosa, Iris virginica and Iris versicolor) with 4 features for each sample: the length and the width of the sepals and petals, in centimeters.

- Based on the combination of these four features, it is possible to develop a linear discriminant model to distinguish the species from each other.

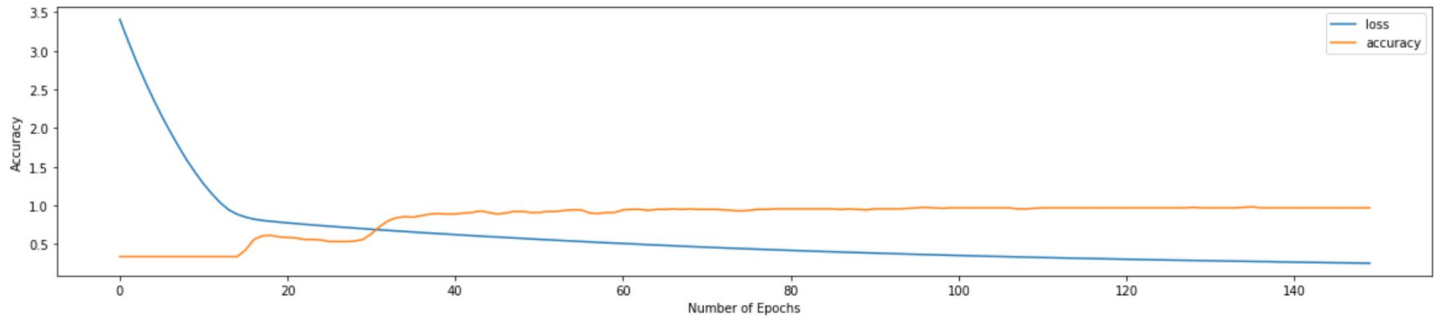


	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
			...		
50	6.4	3.5	4.5	1.2	Versicolor
			...		
150	5.9	3.0	5.0	1.8	Virginica

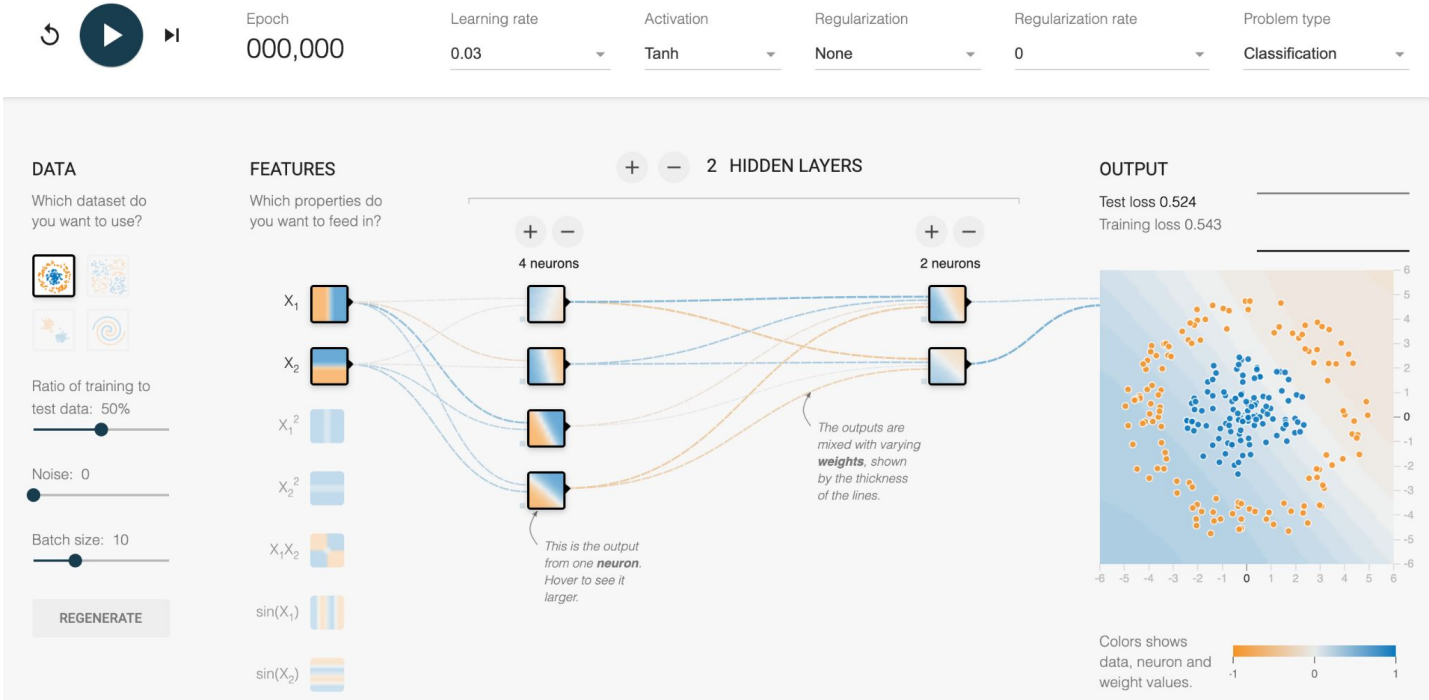
A First Implementation in Tensorflow Of a Classification Problem.

```
# create model
model = Sequential()
model.add(Dense(8, input_dim=4, activation='relu'))
model.add(Dense(3, activation='softmax'))
# Compile model
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
# fit the model
history = model.fit(X, dummy_y, epochs = 200, batch_size = 16)
```

```
Epoch 1/200
10/10 [==============================] - 0s 1ms/step - loss: 2.9310 - accuracy: 0.3289
```



Want to understand more about Neural Network Architecture?
Try [Tensorflow Playground]

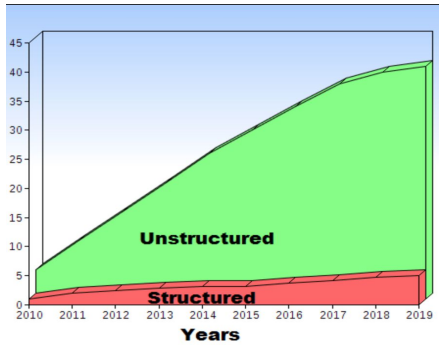


Tensorflow for Time Series Forecasting

Structured / Unstructured Data and How to Think About Them

Structured	Unstructured
List of people with their phone numbers	Text of Encyclopedia Britannica
Temperature in all rooms of a building at every minute for the last 20 years	File share with corporate documents
Data for age and gender of all people entering the building	Raw video feed from surveillance camera

About 90 % of the data generated each day, especially in social networks .. through images, audio, video and text is unstructured data.

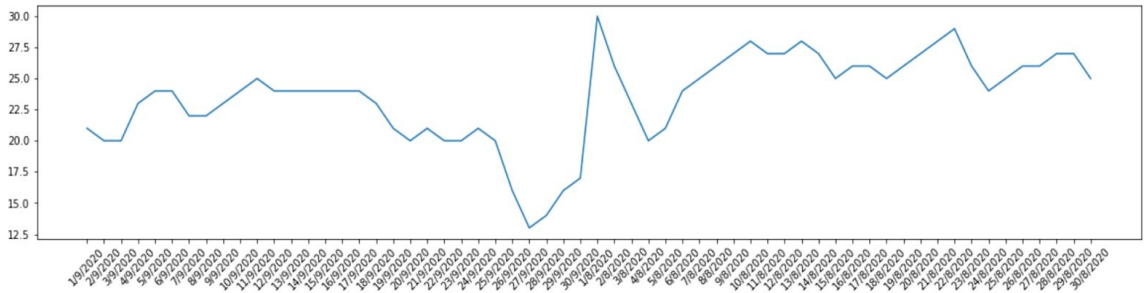


Time Series : Highly Structured Data

Time dependent data

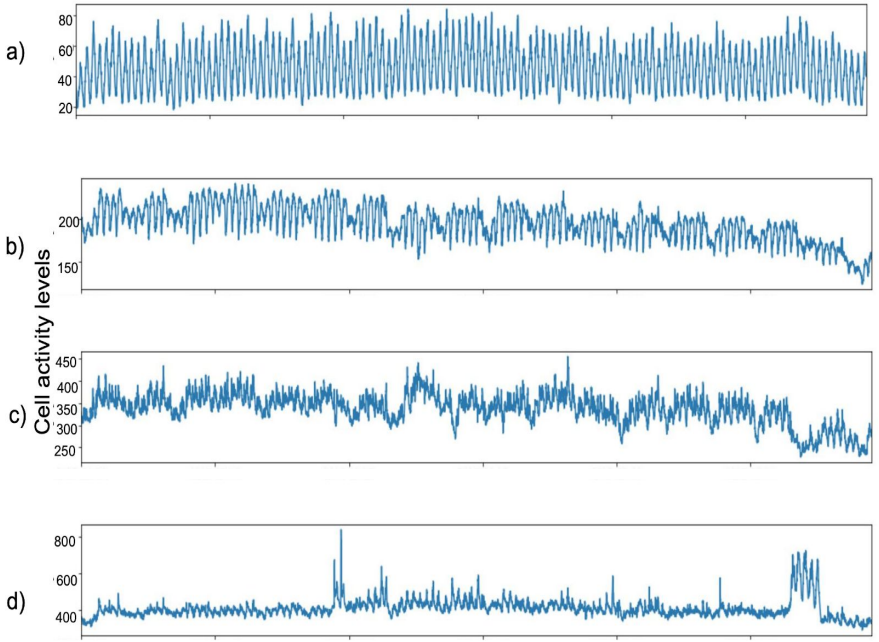
	LOCALITA	DATA	TMEDIA °C
0	Bologna	1/9/2020	21
1	Bologna	2/9/2020	20
2	Bologna	3/9/2020	20
3	Bologna	4/9/2020	23
4	Bologna	5/9/2020	24
...	...	...	...
56	Bologna	27/8/2020	26
57	Bologna	28/8/2020	27
58	Bologna	29/8/2020	27
59	Bologna	30/8/2020	25

What is a time series? A collection of observations made sequentially in time.



Time series - some (not so) famous examples

1. Number of cars passing by a road sensor.
2. Fitbit data on activity and sleep.
3. Observed levels of pollution in a given location.
4. Closing price of a stock each day.
5. The birth rate at all hospitals in a city each year.
6. Observed humidity/temperature in a given location
7. Product sales in units sold each day in a store/location.
8. Number of daily passengers in a train station.
9. Unemployment in a country for each quarter of the year.
10. Utilization demand on mobile phone network.
11. The everyday average price of gasoline in a city.



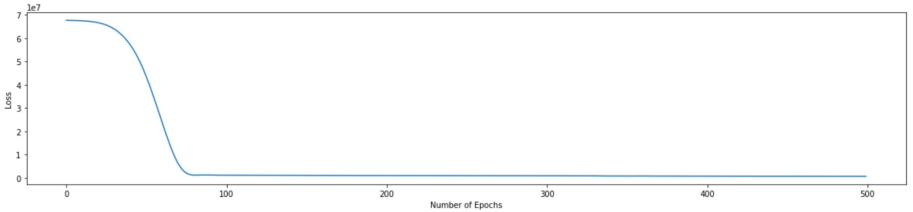
Forecasting Financial Data with Neural Networks and Deep Learning

How do neural networks learn? By mapping input sequences to output sequences.

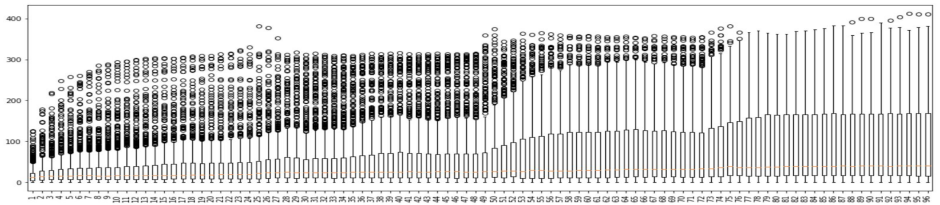
1- Prepare data for deep neural network

s	x	y
238.0	[238. 232. 231. 230. 233. 233. 230. 226. 224. 220. 218. 213.]	[213. 206. 204. 203. 203. 202.]
232.0	[232. 231. 230. 233. 233. 230. 226. 224. 220. 218. 213. 213.]	[206. 204. 203. 203. 202. 198.]
231.0	[231. 230. 233. 233. 230. 226. 224. 220. 218. 213. 213. 206.]	[204. 203. 203. 202. 198. 198.]
230.0	[230. 233. 233. 230. 226. 224. 220. 218. 213. 213. 206. 204.]	[203. 203. 202. 198. 198. 197.]
233.0	[233. 233. 230. 226. 224. 220. 218. 213. 213. 206. 204. 203.]	[203. 202. 198. 198. 197. 194.]
233.0	[233. 230. 226. 224. 220. 218. 213. 213. 206. 204. 203. 203.]	[202. 198. 198. 197. 194. 197.]
230.0	[230. 226. 224. 220. 218. 213. 213. 206. 204. 203. 203. 202.]	[198. 198. 197. 194. 197. 196.]
226.0	[226. 224. 220. 218. 213. 213. 206. 204. 203. 203. 202. 198.]	[198. 197. 194. 197. 196. 196.]

2- Train the model and find best parameters



3- Test the model and compute forecasting error



Multilayer Perceptrons for Time Series Forecasting

```
# split the data in train-test
split_size = int(len(X)* 0.75)
X_train, y_train = X[:split_size], y[:split_size]
X_test, y_test = X[split_size:],y[split_size:]
```

```
# model
model = Sequential()
model.add(Dense(500, activation='relu', input_dim=nr_steps_in))
model.add(Dense(nr_steps_out))
model.compile(optimizer='adam',  loss='mse' , metrics=['accuracy'])
history = model.fit(X_train, y_train, validation_split=0.33, batch_size=64, epochs=500, verbose=2)
```

```
# perform a prediction
x_input = np.array([16, 26, 25, 24, 32, 21])
x_input = x_input.reshape((1, n_steps_in))
yhat = model.predict(x_input, verbose=2)
print(yhat[0])
```

```
# compute error
predicted = []
for i in range(len(X_test)):
    x_input = X_test[i].reshape((1, nr_steps_in))
    yhat = model.predict(x_input, verbose = 0)
    predicted.append(np.around(yhat[0], decimals = 1))
```

Forecasting Performance Measures

How can we say which method is best? Yes, by measuring the forecasting error.

Forecasting Error → (Predicted - Real Value)

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T},$$

MAE → (Mean Absolute Error)

$$\text{MAE} = \text{mean}(|e_t|),$$

RMSE → (Root Mean Square Error)

$$\text{RMSE} = \sqrt{\text{mean}(e_t^2)}.$$

MAPE → (Mean Absolute Percentage Error)

$$\text{MAPE} = \text{mean}(|p_t|). \qquad p_t = 100e_t/y_t.$$

Computer Vision With Tensorflow - From CNN to Pretrained Models

Structured / Unstructured Data and How to Think About Them

Structured	Unstructured
List of people with their phone numbers	Text of Encyclopedia Britannica
Temperature in all rooms of a building at every minute for the last 20 years	File share with corporate documents
Data for age and gender of all people entering the building	Raw video feed from surveillance camera

About 90 % of the data generated each day, especially in social networks .. through images, audio, video and text is unstructured data.

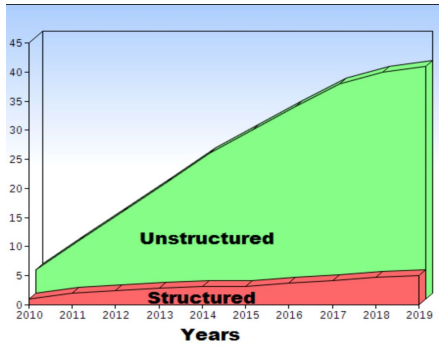


Image Processing and Classification using Tensorflow.

- (OCR) used in Google Translator:

这是对计算机视觉应用程序的测试

- QR codes

- Visual Search for Fashion Industry

- Lane Detection in Autonomous Vehicles

Applications of Computer Vision

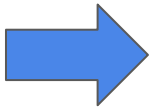
Object detection

Computer Vision Algorithm

Image processing : working with pixels and arrays on more that two dimensions

```
image = cv2.imread('images/img.jpg')
type(image)
plt.imshow(image[:,:,::-1])
print(image.shape)
```

```
(434, 500, 3)
```

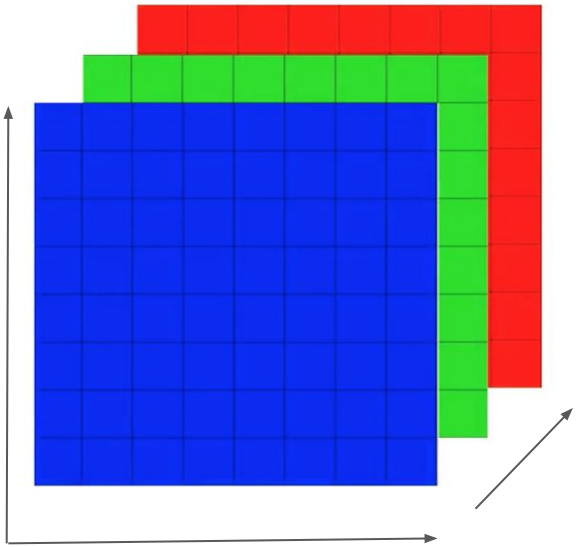


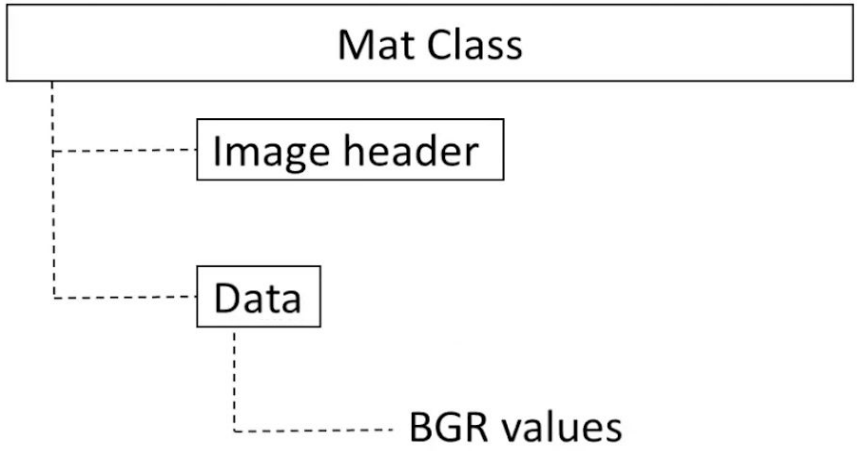
```
[[[250 245 242]
  [250 245 242]
  [250 245 242]
  ...
  [250 245 242]
  [250 245 242]
  [250 245 242]]

 [[250 245 242]
  [250 245 242]
  [250 245 242]
  ...
  [250 245 242]
  [250 245 242]
  [250 245 242]]

 [[250 245 242]
  [250 245 242]
  [250 245 242]
  ...
  [250 245 242]
  [250 245 242]
  [250 245 242]]]
```

.....

```
[[ 81  62  66]
 [ 82  63  67]
 [ 82  63  67]
 ...
 [ 80  63  69]
 [ 81  64  70]
 [ 83  67  73]]

[[ 78  59  63]
 [ 79  60  64]
 [ 79  60  64]
 ...
 [ 77  60  66]
 [ 79  62  68]
 [ 82  65  71]]

[[ 77  58  62]
 [ 77  58  62]
 [ 76  57  61]
 ...
 [ 78  61  67]
 [ 81  64  70]
 [ 86  69  75]]]
```

Image Processing: images and their dimensions





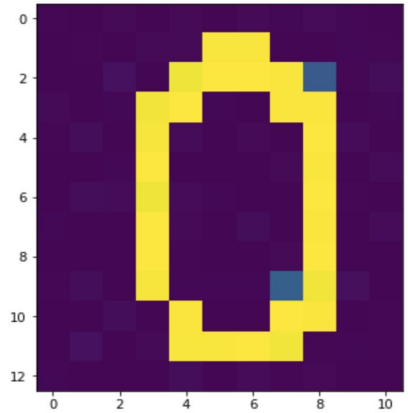
A black & white picture has only one layer (only two dimensions) as of next slide!

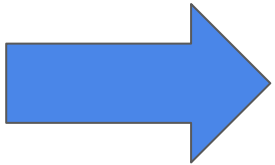
Images are made of a large number of pixels

A black & white picture of number zero

```
imagePath = "images/number_zero.jpg"
testImage = cv2.imread(imagePath, 0)

plt.figure(figsize = (6, 6))
plt.imshow(testImage)
```

```
<matplotlib.image.AxesImage at 0x7ffa96924950>
```





```
testImage.shape
```

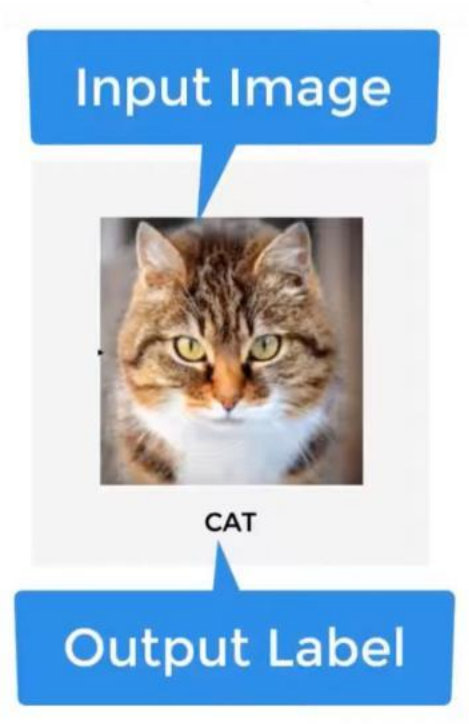
```
(13, 11)
```

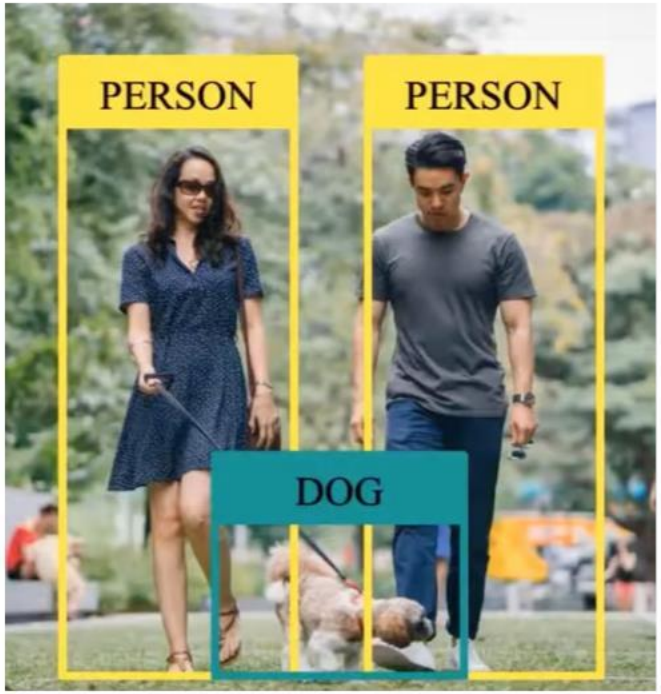
```
print(testImage)
```

```
[[  1   0   3   0   3   0   3   2   4   2   0]
 [  0   1   0   3   3 253 253   0   0   2   1]
 [  0   0   8   0 249 255 255 253  71   1   5]
 [  3   0   2 251 255   2   0 253 254   0   2]
 [  1   5   0 252   4   0   3   0 255   4   0]
 [  0   0   2 255   0   0   0   3 253   0   4]
 [  0   5   4 249   4   2   0   0 255   1   0]
 [  2   0   0 255   3   0   5   0 254   0   4]
 [  0   0   0 255   1   0   0   3 255   0   0]
 [  1   5   0 252   2   2   2  76 250   7   0]
 [  0   0   5   0 254   0   0 255 254   0   1]
 [  0   8   0   3 253 253 255 250   1   2   1]
 [  2   0   0   0   5   0   4   1   3   0   0]]
```

Working with black & white pictures is much easier and this is the reason why different applications perform a transformation into B&W first.

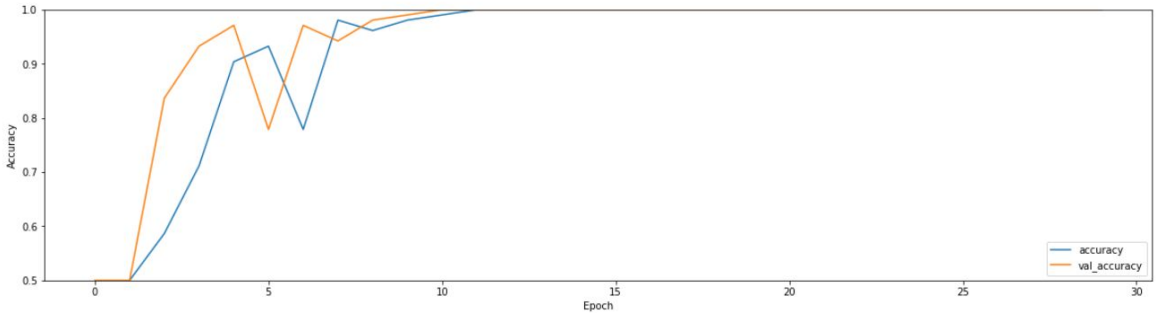
Image classification and object detection





Dog Breeds Classification: Border Collies Vs Labrador

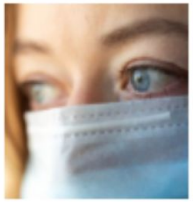
```
model = models.Sequential()
model.add(layers.Conv2D(16, (3, 3), activation='relu', input_shape=(300, 300, 3)))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(32, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))

model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(2))
```



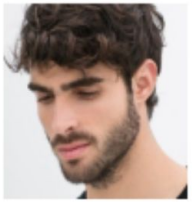
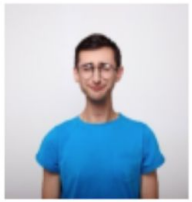
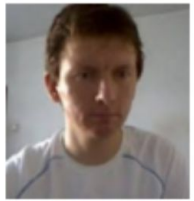
Mask-NoMask Classification with Convolutional Networks

Classifying pictures of people wearing (or not wearing) a mask!

Use CNN and train it with pictures from the two classes!

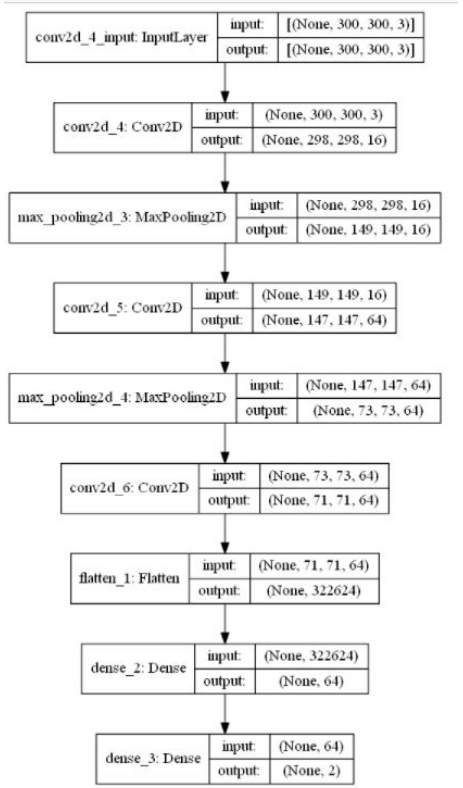





Architecture of Convolutional Network for Mask-NoMask Classification

```
model = tf.keras.models.Sequential([
    # Note the input shape is the desired size of the image 300x300 with 3 bytes color

    # This is the first convolution
    tf.keras.layers.Conv2D(16, (3,3), activation='relu', input_shape=(300, 300, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),

    # The second convolution
    tf.keras.layers.Conv2D(32, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    # The third convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    # The fourth convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    # The fifth convolution
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    # Flatten the results to feed into a DNN
    tf.keras.layers.Flatten(),

    # 512 neuron hidden layer

    tf.keras.layers.Dense(512, activation='relu'),
    # Only 1 output neuron. It will contain a value from 0-1 where 0 for 1 class ('Reps')
    tf.keras.layers.Dense(1, activation='sigmoid')
])
```



Tensorflow for NLP - The LSTM Networks

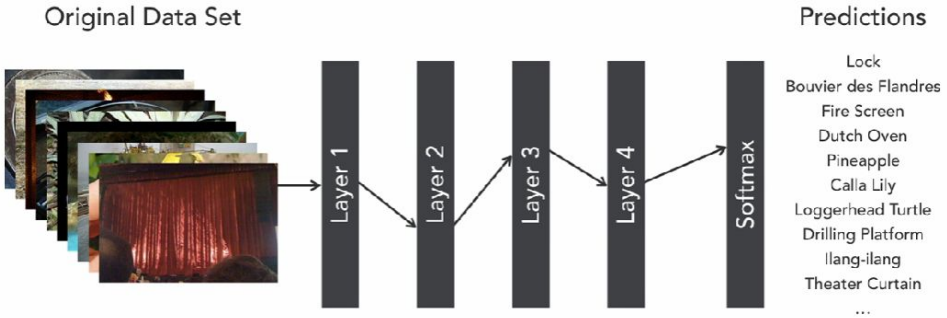
Transfer Learning with Tensorflow - the TensorHub.

What is Transfer Learning and why it matters.

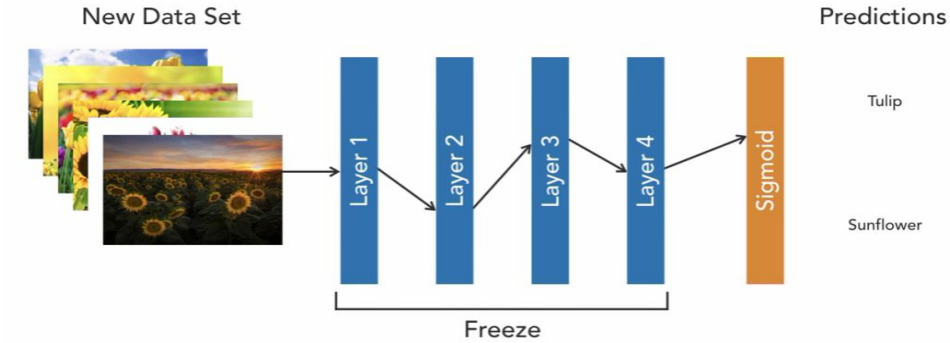
- Transfer learning for machine learning is not different from transfer learning in human learning.
- Transfer learning is about reusing an already trained neural network.
- This trained neural network would have been trained on a data set that is similar to the one you're working on.
- Many image classification models are trained on ImageNet, this has 1000 classes of objects and animals.
- If you're tryna tell the difference between different types of flowers, then transfer learning using a model that was trained on ImageNet would be a very good start

How to Implement Transfer Learning Examples

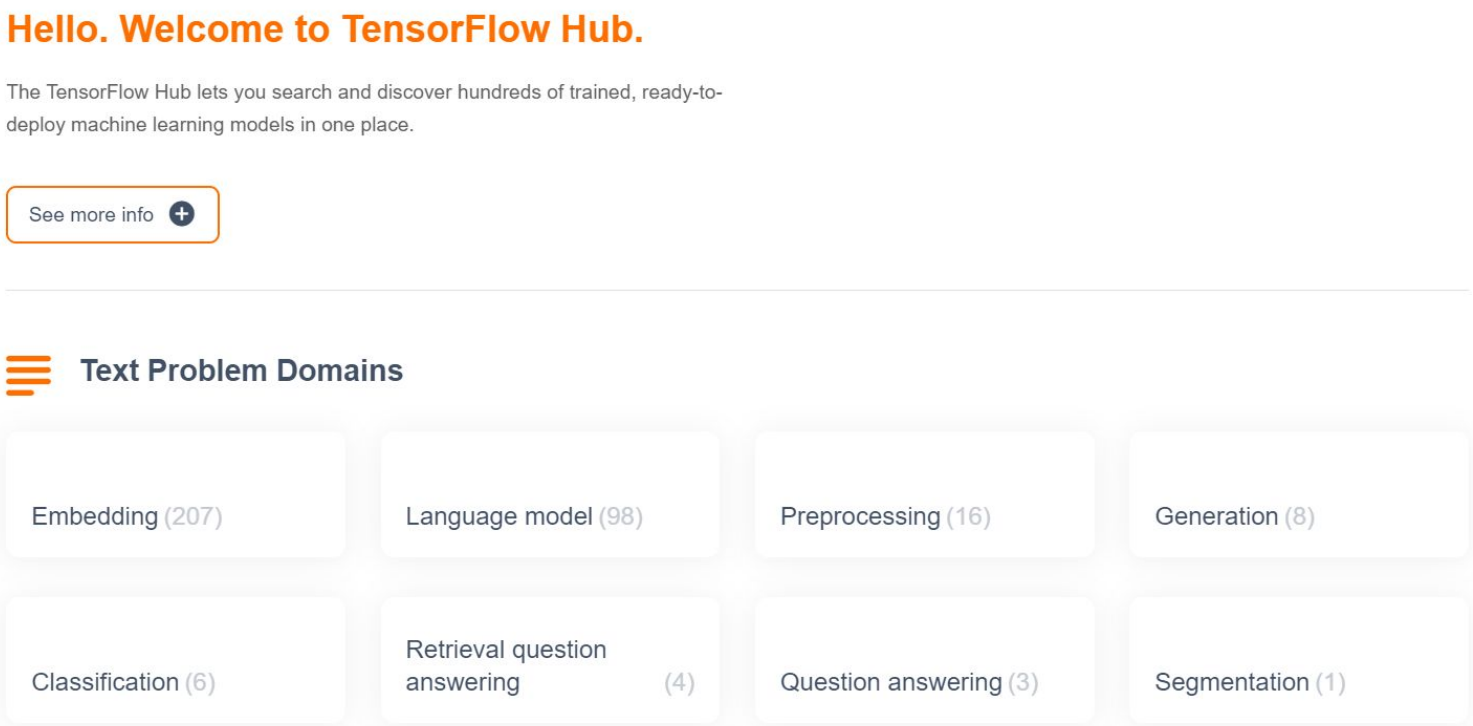
The original model that has beet train on ImageNet



The new model that we create based on the one originally trained on ImageNet



TensorFlow Hub is a repository of trained machine learning models.



Transfer Learning for NLP - Some Pretrained Models

- **BERT (Bidirectional Encoder Representations from Transformers).** BERT is a technique for NLP pre-training, developed by Google.

- **RoBERTa (Robustly Optimized BERT Pretraining Approach).** RoBERTa is an optimized method for the pre-training of a self-supervised NLP system.

- **OpenAI's GPT-3** is a transformer-based NLP model that performs translation, question-answering, poetry composing etc..

- **DeBERTa (Decoding-enhanced BERT with disentangled attention)** The authors from Microsoft Research propose DeBERTa, with two main improvements over BERT, namely disentangled attention and an enhanced mask decoder.

Transfer Learning For Computer Vision - Some Pretrained Models

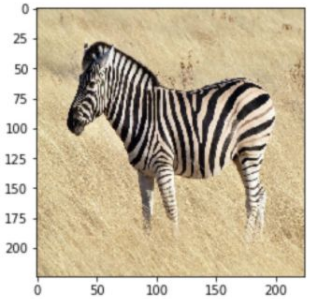
VGG16 is a deep neural network with either 16 or 19 layers.

ResNet-50 is a state of the art from 2015, it's a 50 layer neural network that manages to be more accurate, but uses less memory than the VGG design.

Inception v3 is another design from 2015 that also performs very well.

Google's MobileNet created in 2017 is designed specifically to be able to run well on low power devices.

Google's NASNet which was created at the end of 2017, is based on the the idea of having algorithms design neural networks.




```
# Normalize the input image's pixel values to the range used when traini
y = vgg16.preprocess_input(x)

# Run the image through the deep neural network to make a prediction
predictions = model.predict(y)

# Look up the names of the predicted classes. Index zero is the results
predicted_classes = vgg16.decode_predictions(predictions)

print("Top predictions for this image:")

for imagenet_id, name, likelihood in predicted_classes[0]:
    print("Prediction: {} - {:2f}".format(name, likelihood))
```

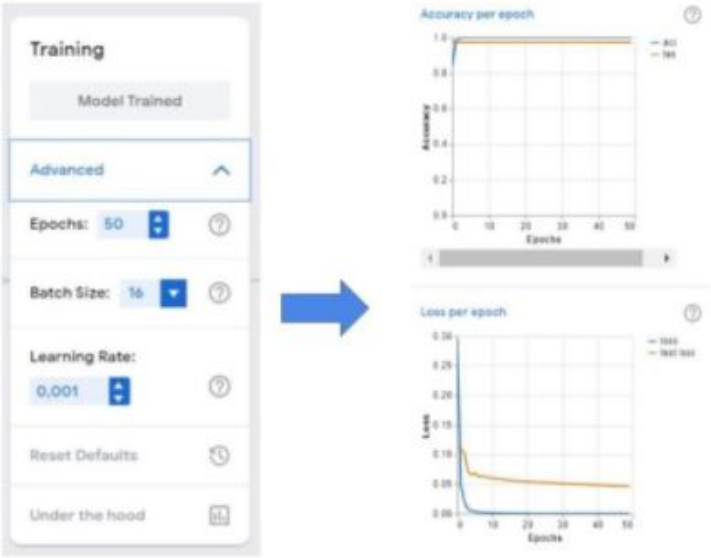
```
Top predictions for this image:
Prediction: zebra - 0.999603
Prediction: gazelle - 0.000199
Prediction: impala - 0.000122
```

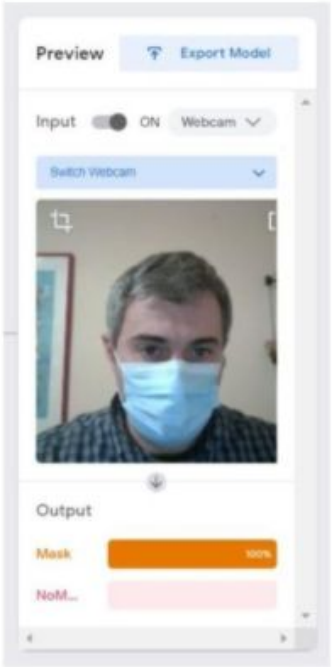
Transfer Learning for both CV and NLP: meet DALL-E.

DALL·E is a 12-billion parameter version of GPT-3 trained to generate images from text descriptions, using a dataset of text–image pairs. It has a diverse set of capabilities, like creating anthropomorphized versions of animals and objects, combining unrelated concepts in plausible ways, rendering text, and applying transformations to existing images.

Using Transfer Learning for Mask-NoMask Classification

https://teachablemachine.withgoogle.com/





Local Deployment of TensorFlow Models with Streamit Framework

Machine Learning Life Cycle - MLLC

1- Gather data

2- Data Preparation

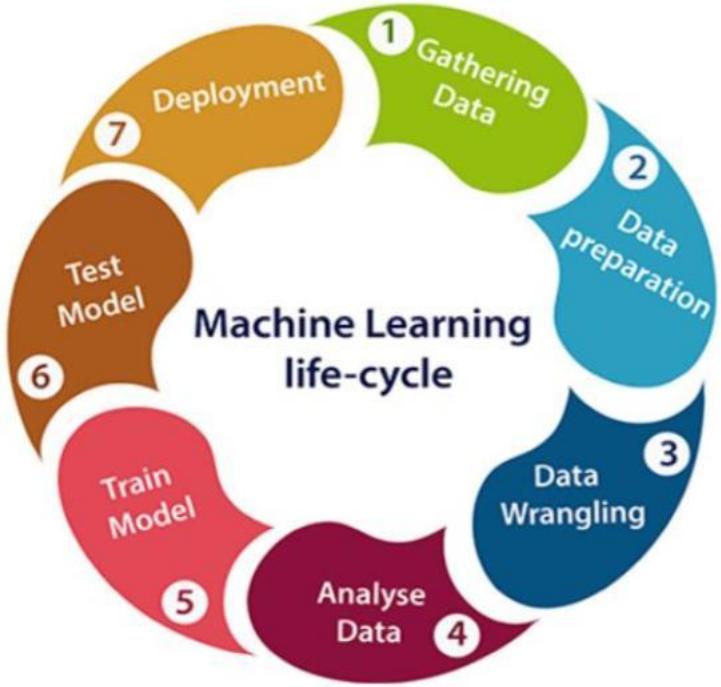
3- Data Wrangling

4- Analyse Data

5- Train Models

6- Test Models

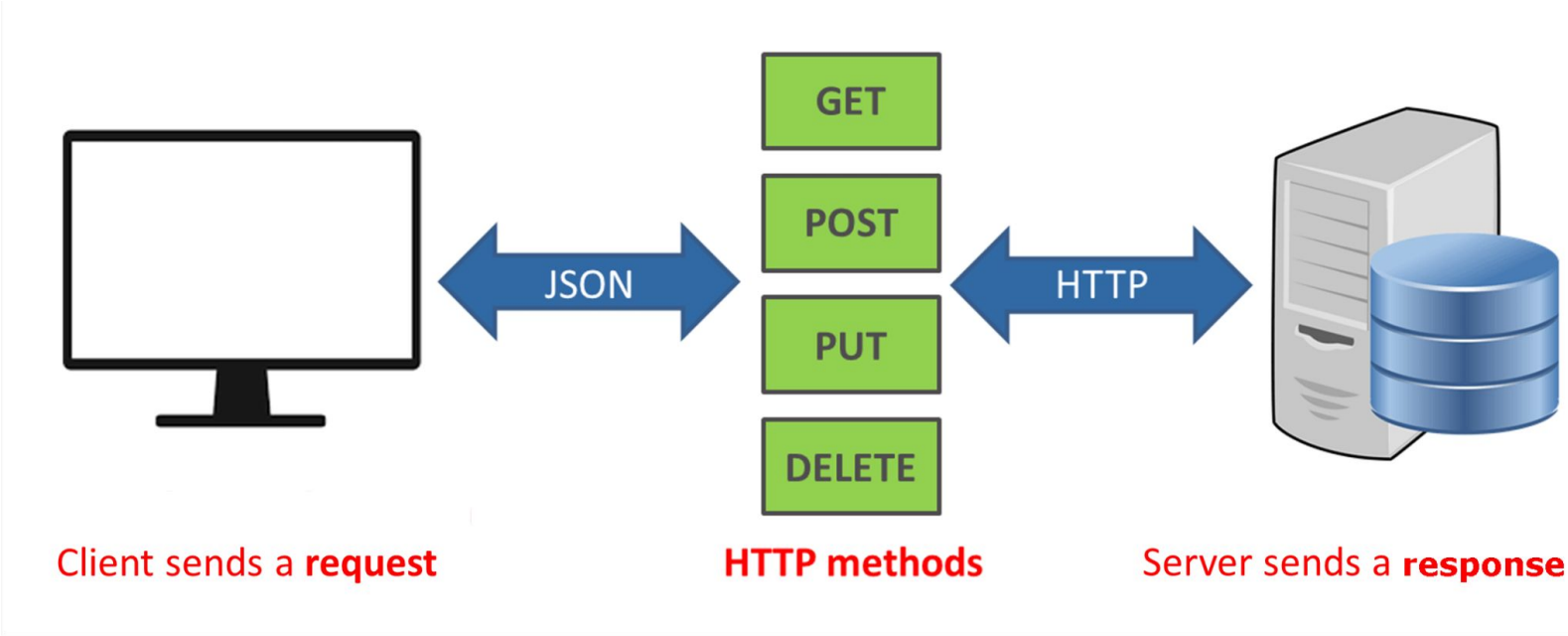
7- Deployment



Deployment of ML models - Different ways to deploy.

- deploying on a server (for example with a REST API in Flask)
- deploying in a container (such as Docker)
- serverless deployment
- model streaming — instead of REST APIs, all of the models and application code are registered on a stream processing engine like Apache Spark, Apache Storm, and Apache Flink.

Deployment of ML models with REST API in Flask



Introduction to Flask - How to use it for Model Deployment.

Contents

Quickstart

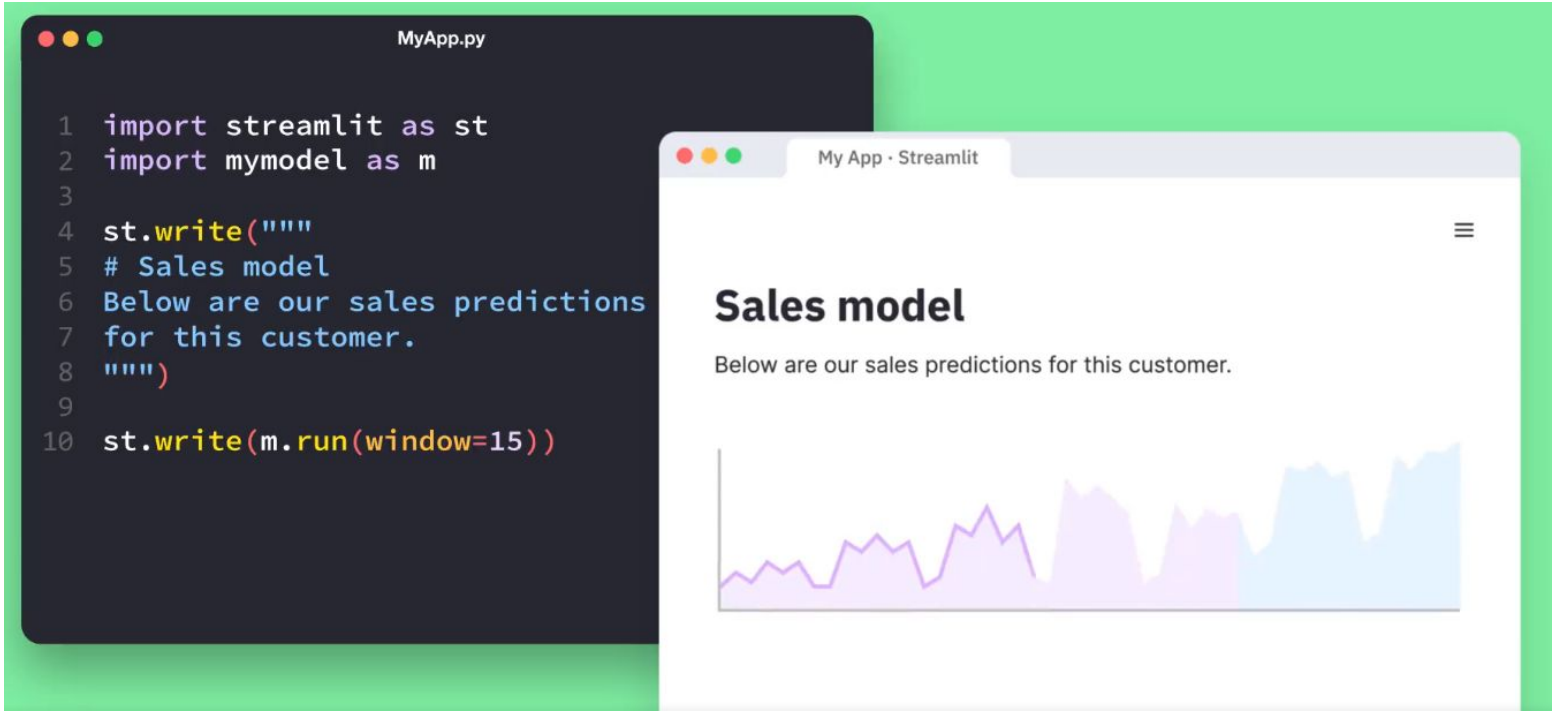
Eager to get started? This page gives a good introduction over to the Installation section.

A Minimal Application

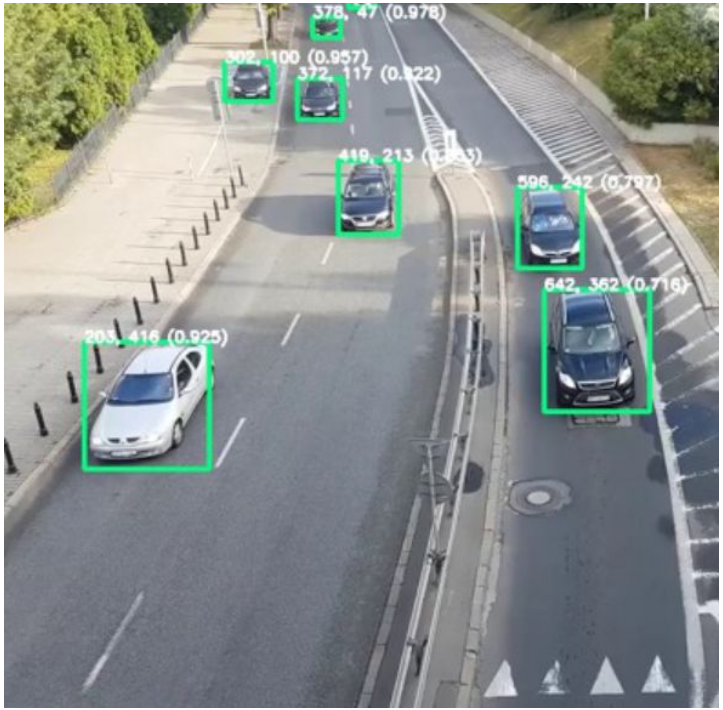
A minimal Flask application looks something like this:

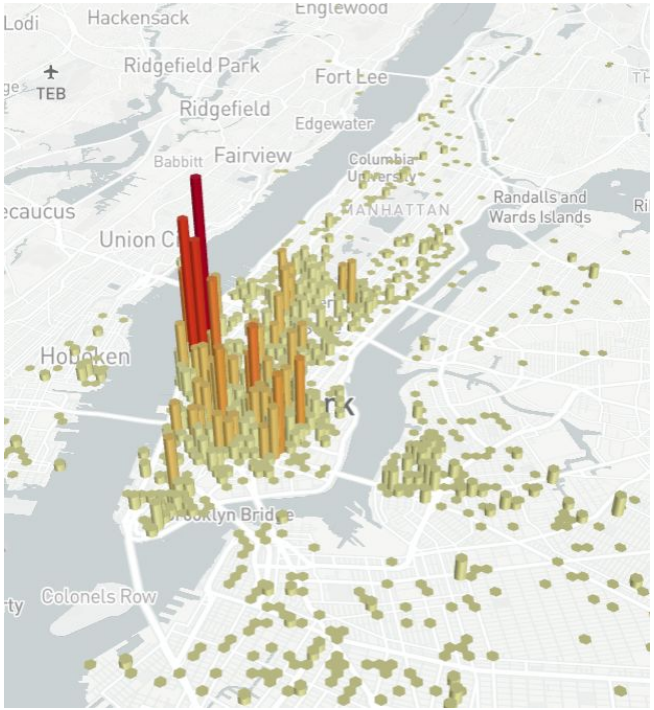
```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
```

Streamlit : Serverless application for machine learning deployment



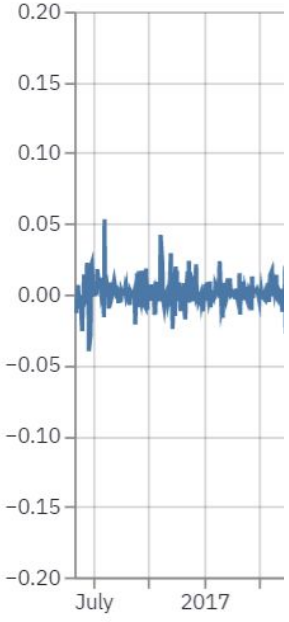
Streamlit : Serverless application for machine learning deployment



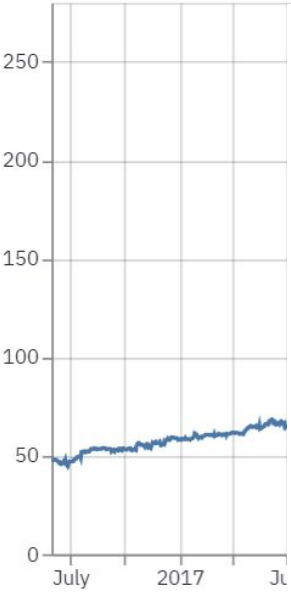


Using Streamlit with Yahoo Finance Data

Return as a graphics!



Visualize the stock timeseries for the last 5 years!



Visualize the volume of trading for MSFT stock!