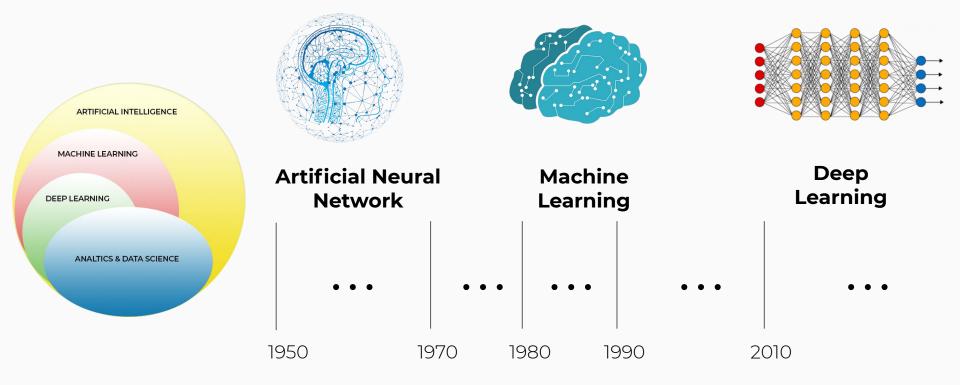
## Introduction to Deep Learning

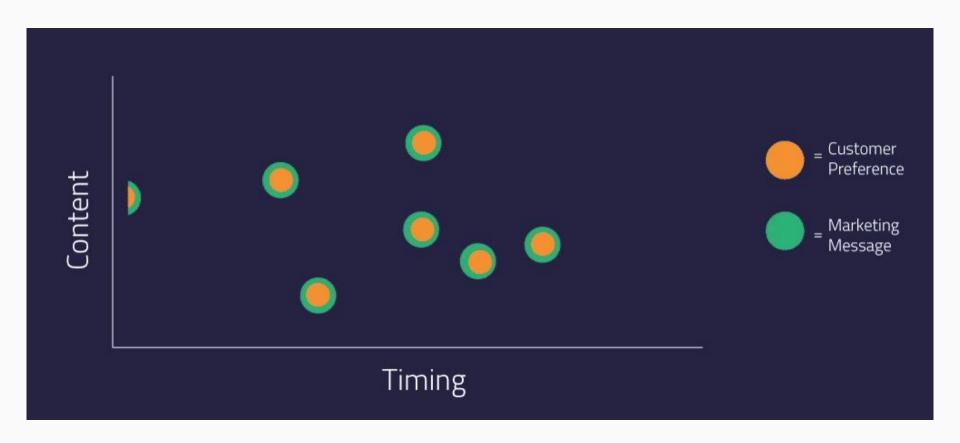
Deep Learning in Remote Sensing

Episode-1

İrem KÖMÜRCÜ iremkomurcu.com iremkomurcubm@gmail.com

## O 1 AI - DL - ML - Data Terms







#### **Artificial Neuron**

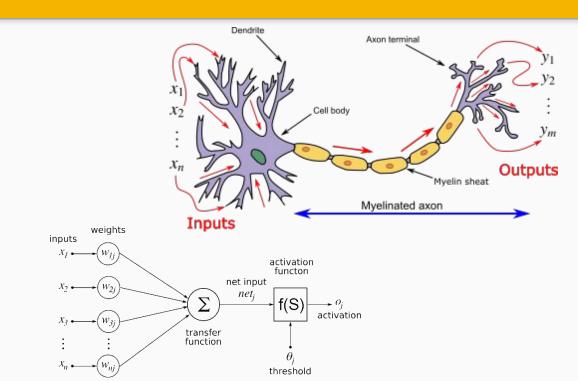
y = f(Piwixi)

xi:inputs

wi:weights

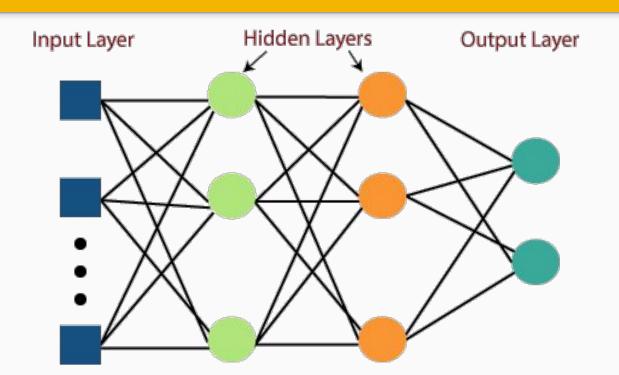
f: non-linearity

 $Y = \sum (weight * input) + bias$ 





#### Organization in layers



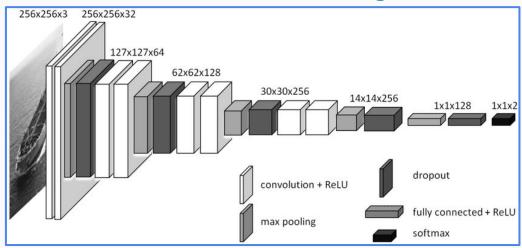


#### Deep Learning and Classification



# Types of Deep Learning Algorithms

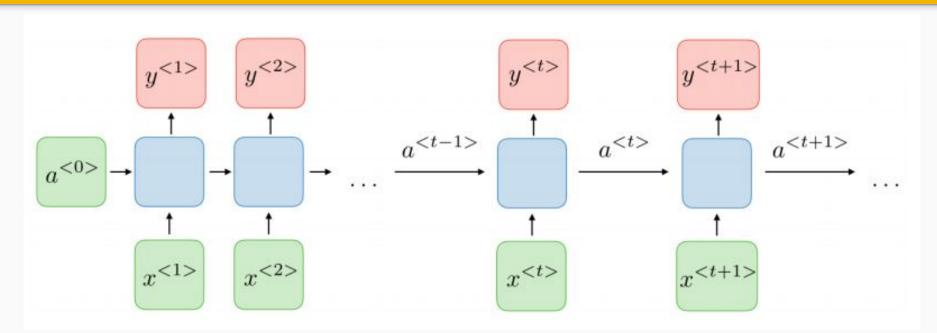




- Convolutional Neural Networks (CNNs)
- Long Short Term Memory Networks (LSTMs)
- Recurrent Neural Networks (RNNs)
- Generative Adversarial Networks (GANs)
- Radial Basis Function Networks (RBFNs)
- Multilayer Perceptrons (MLPs)
- Self Organizing Maps (SOMs)
- Deep Belief Networks (DBNs)
- Restricted Boltzmann Machines (RBMs)

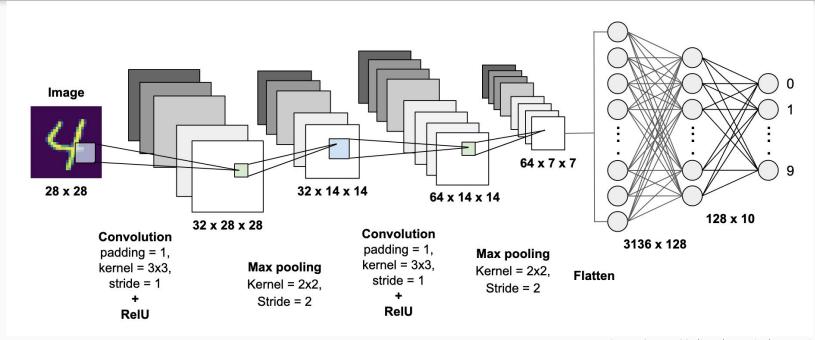


#### **Recurrent Neural Networks**



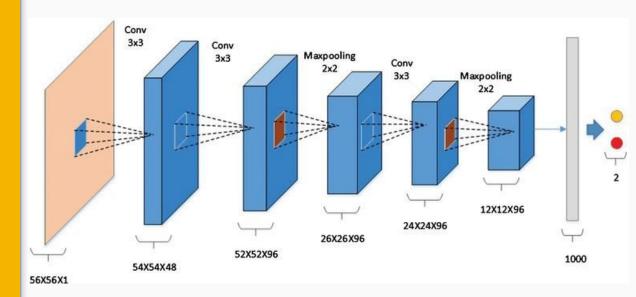


#### **Creating Model - Example CNN**



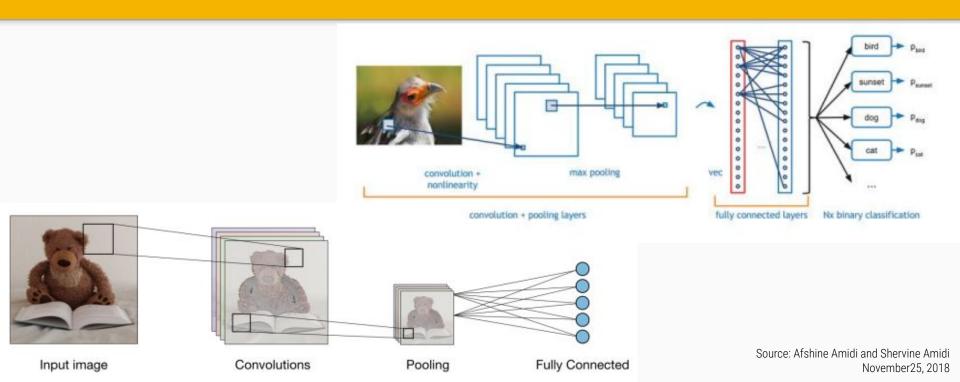


- Convolutional Layer
- Non-Linearity Layer
- Pooling (Downsampling Layer)
- Stride
- Padding
- Flattening Layer
- Fully-Connected Layer



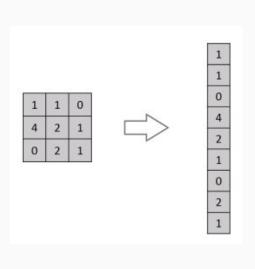


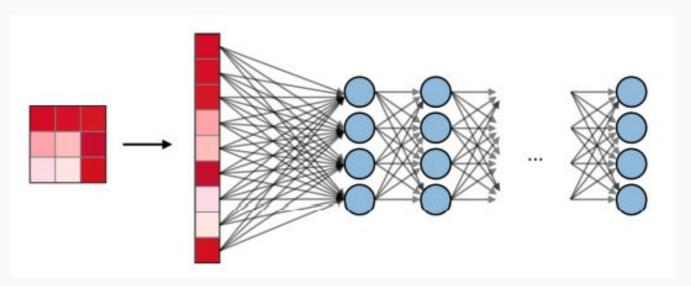
#### **Convolutional Neural Networks**





#### Flatten and Fully Connected





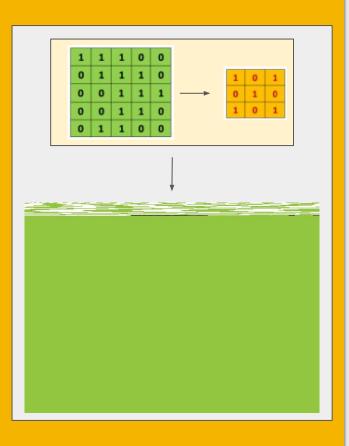


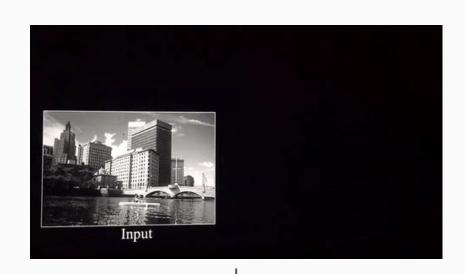
#### **Pooling**

1	0	2	3			
4	6	6	8		6	i
3	1	1	0	<b>—</b>	3	
1	2	2	4		5252	

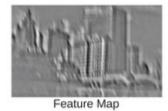
	Max pooling	Average pooling		
Purpose	Each pooling operation selects the maximum value of the current view	Each pooling operation averages the values of the current view		
Illustration	max	avg		
Comments	- Preserves detected features - Most commonly used	- Downsamples feature map - Used in LeNet		

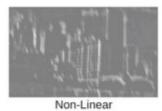
#### iremkomurcu.com









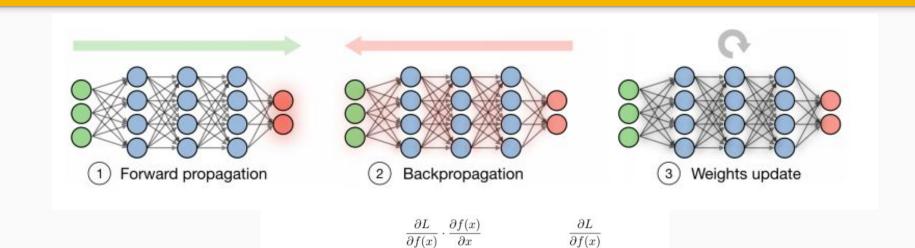


Non-Emea



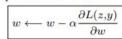
#### Backpropagation and Updating weights

$$Y = \sum (weight * input) + bias$$



Using this method, each weight is updated with the rule:

x



f(x)

### Data Augmentation

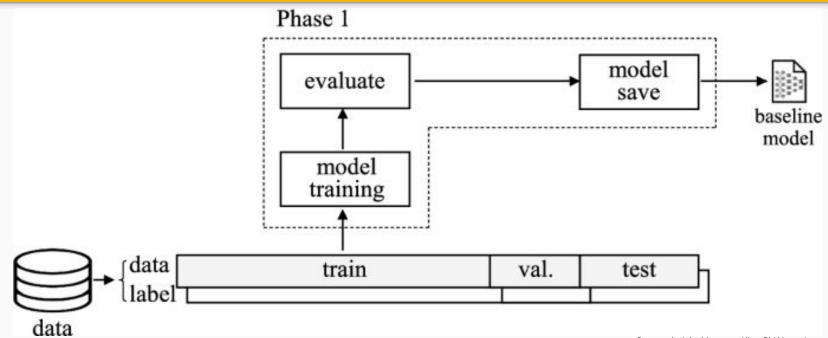


Original	Flip	Rotation	Random crop
-	4		8
- Image without any modification	- Flipped with respect to an axis for which the meaning of the image is preserved	- Rotation with a slight angle - Simulates incorrect horizon calibration	- Random focus on one part of the image - Several random crops can be done in a row

Color shift	Noise addition	Information loss	Contrast change
\$	4		
<ul> <li>Nuances of RGB is slightly changed</li> <li>Captures noise that can occur with light exposure</li> </ul>	- Addition of noise - More tolerance to quality variation of inputs	- Parts of image ignored - Mimics potential loss of parts of image	- Luminosity changes - Controls difference in exposition due to time of day



#### Create General Baseline Model



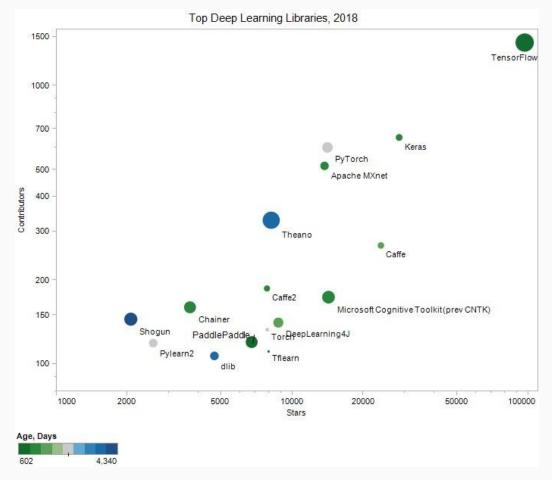


## **OZ DL Frameworks and Libraries**

#### Some DL Frameworks

- Tensorflow
- Keras
- Pytorch
- MXNet
- Theano
- Caffe





#### Some Most Popular and Useful **Python Libraries**

- Scikit-learn
- Pandas
- Numpy
- OpenCV
- Matplotlib



















#### Numpy

The fundamental package for scientific computing with Python



```
import numpy as np
x = np.array([[1,2],[3,4]])
y = np.array([[5,6],[7,8]])
v = np.array([9,10])
w = np.array([11, 12])
# Inner product of vectors; both produce 219
print(v.dot(w))
print(np.dot(v, w))
# Matrix / vector product; both produce the rank 1 array [29 67]
print(x.dot(v))
print(np.dot(x, v))
# Matrix / matrix product; both produce the rank 2 array
# [[19 22]
# [43 50]]
print(x.dot(y))
print(np.dot(x, y))
```

#### Pandas

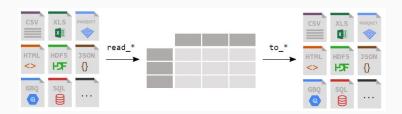
Pandas; open source data analysis and manipulation tool





```
In [2]: titanic = pd.read_csv("data/titanic.csv")
```

```
In [3]: titanic
Out[3]:
     PassengerId
                  Survived Pclass
                                              Fare Cabin
                                                           Embarked
                                            7.2500
                                                     NaN
0
                                                                  5
                                          71.2833
                                                     C85
                                            7.9250
                                                     NaN
                                           53,1000
                                                    C123
                                            8.0500
                                                     NaN
                                          13,0000
886
             887
                                                     NaN
887
             888
                                           30.0000
                                                     B42
888
                                           23,4500
                                                     NaN
             889
                                           30,0000
                                                     C148
889
             890
890
             891
                                            7.7500
                                                     NaN
                                                                  Q
[891 rows x 12 columns]
```



#### Scikit-learn

- Simple and efficient tools for predictive data analysis
- Accessible to everybody, and reusable in various context
- Built on Numpy, SciPy, and Matplotlib



```
$ python
>>> from sklearn import datasets
>>> iris = datasets.load_iris()
>>> digits = datasets.load_digits()
```

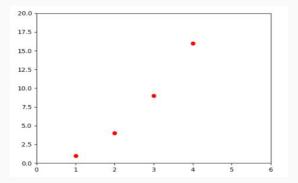
```
>>> digits.target array([0, 1, 2, ..., 8, 9, 8])
```

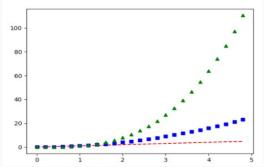
#### Matplotlib

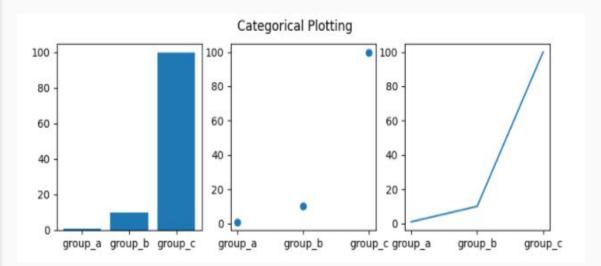
Visualization with Python













#### Matplotlib

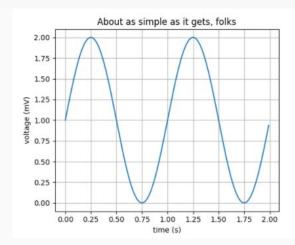
Visualization with Python



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

t = np.arange(0.0, 2.0, 0.01)
s = 1 + np.sin(2*np.pi*t)
plt.plot(t, s)

plt.xlabel('time (s)')
plt.ylabel('voltage (mV)')
plt.title('About as simple as it gets, folks')
plt.grid(True)
plt.savefig("test.png")
plt.show()
```



#### OpenCV

OpenCV is an open source library used in real-time computer vision and machine learning software library.

OpenCV has C++, Python, Java and MATLAB interfaces.



```
# read data
image = cv2.imread(self.images_fps[i])
image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
mask = cv2.imread(self.masks_fps[i], 0)
```

## 03 DL Working Environments

### Deep Learning - Working Environment

Some environments that can be used while developing a deep learning model

#### Local Environment:

- Anaconda
- Spyder
- Jupyter Notebook

#### Cloud Environment:

- Google Colab
- AWS
- Azure

#### Computer Needed:

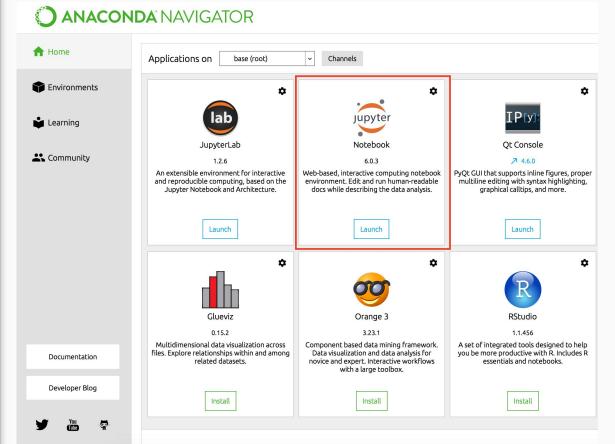
GPU Supported, Cuda, cuDNN

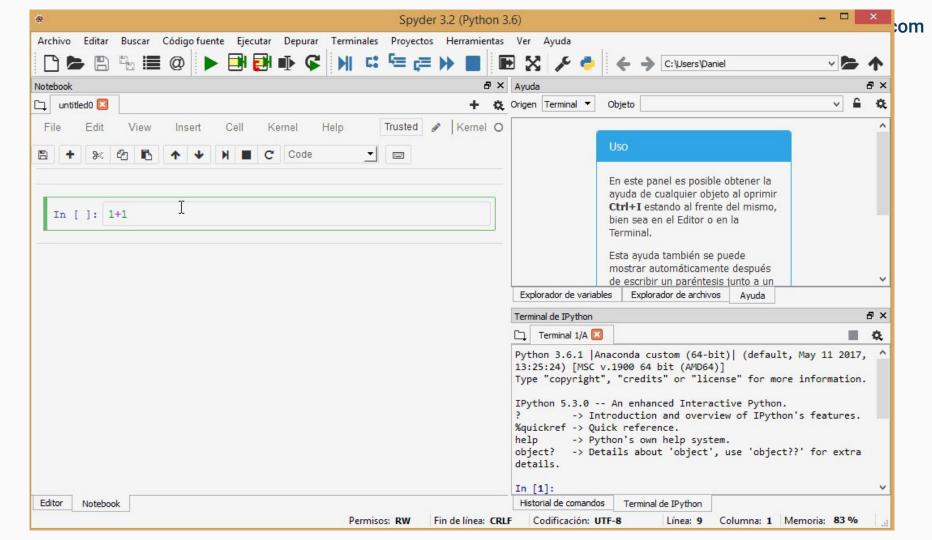
#### **Container Engine**

Docker

#### Anaconda Navigator

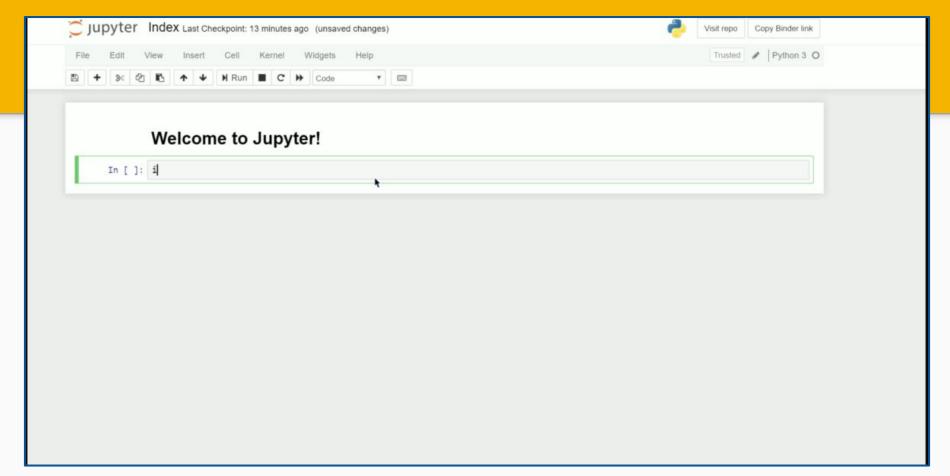








#### Jupyter Notebook





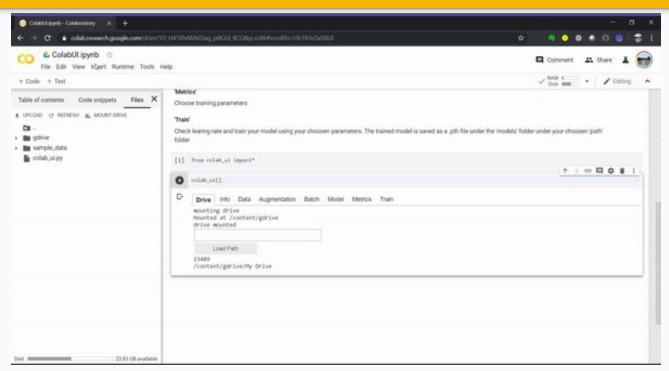






#### Google Colab

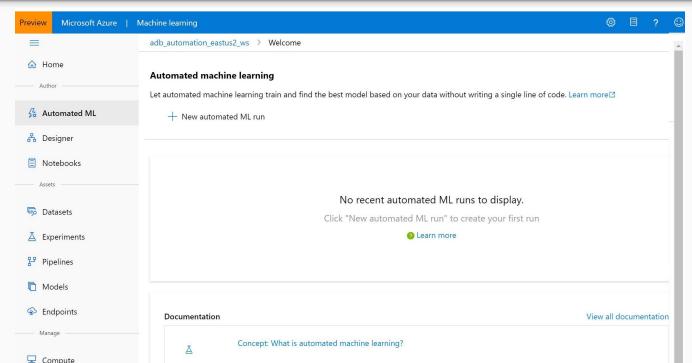






#### Microsoft Azure

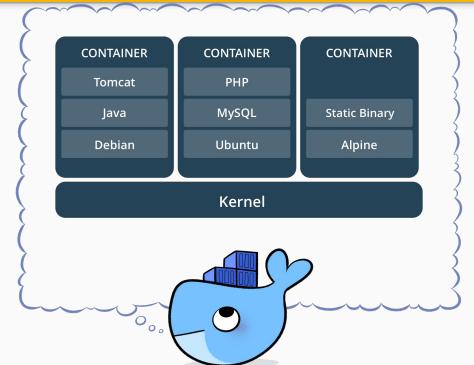






#### Docker





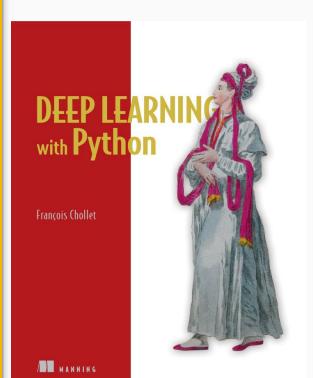


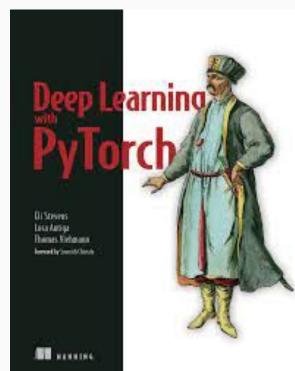
### 04 Suggested Resources



Suggested Resources -Book-

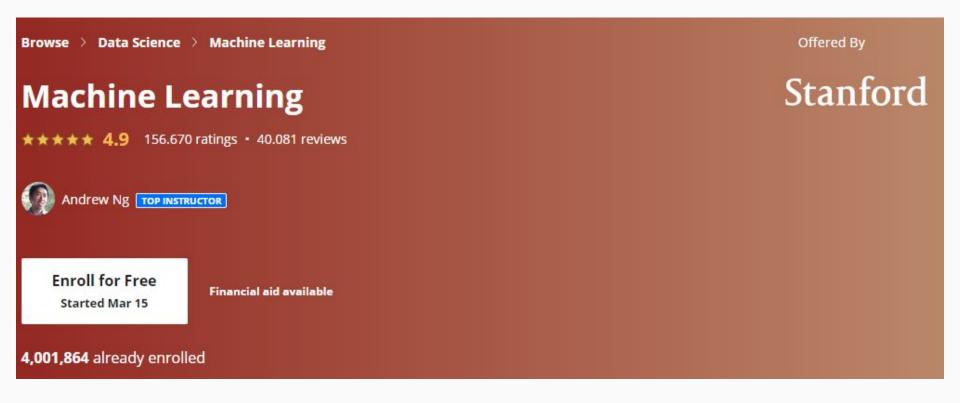
Recommended for those learning deep learning and getting started with PyTorch and TensorFlow/Keras











### kaggle





#### **THANKS**



Does anyone have any questions?

iremkomurcubm@gmail.com iremkomurcu.com







