Introduction to Deep Learning

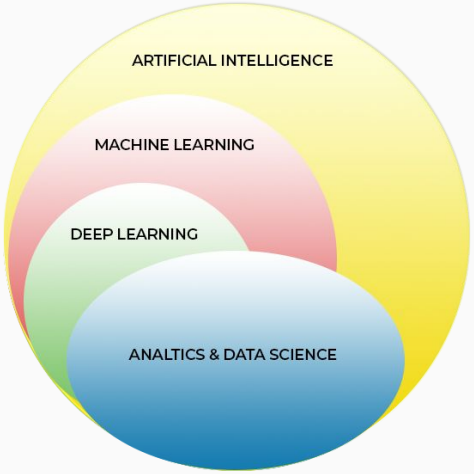
Deep Learning in Remote Sensing

Episode-1

İrem KÖMÜRCÜ
iremkomurcu.com
iremkomurcubm@gmail.com

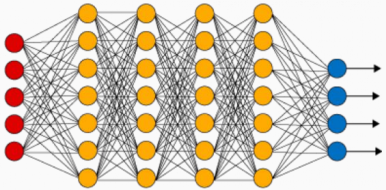
01

AI - DL - ML - Data Terms



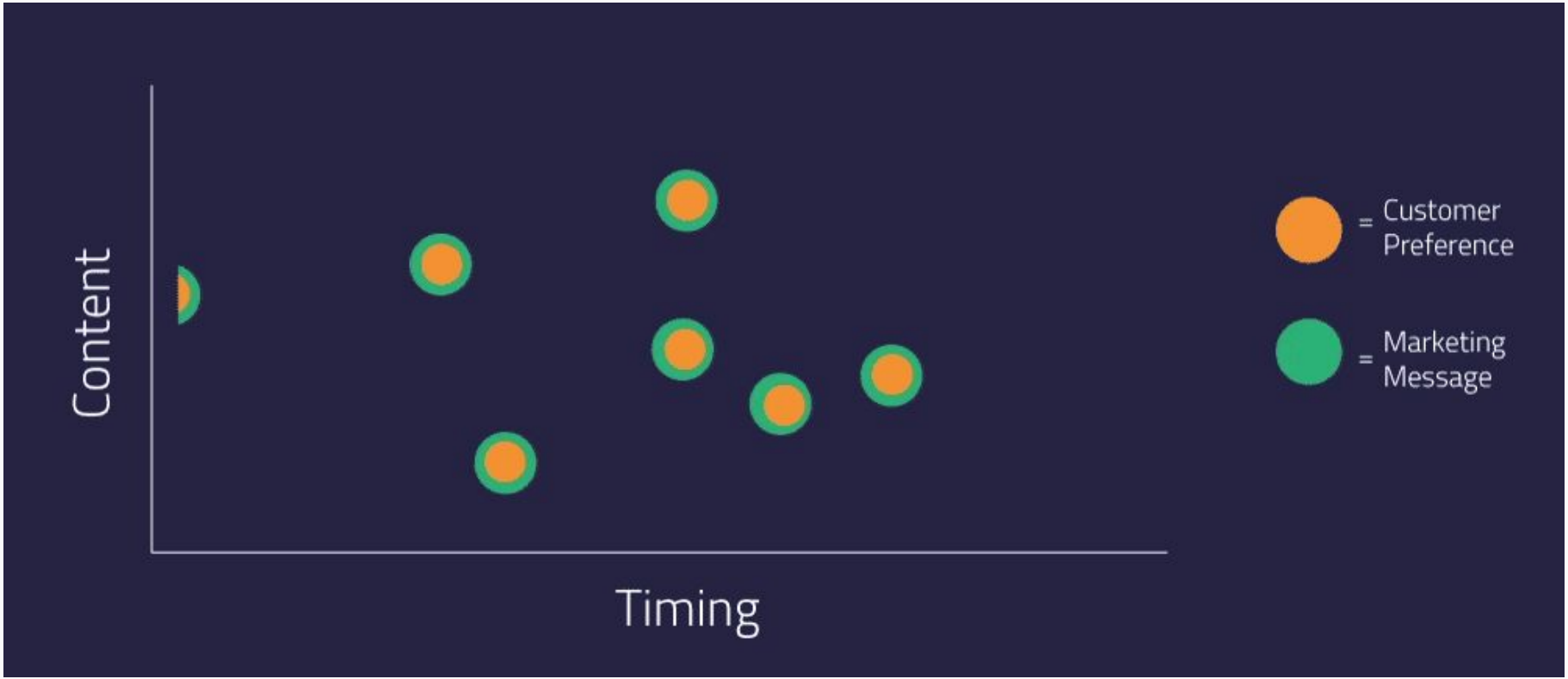
Artificial Neural Network

Machine Learning



Deep Learning

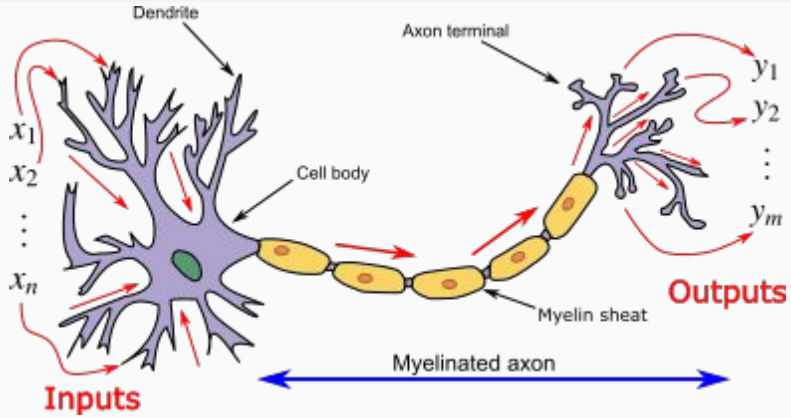
1950 1970 1980 1990 2010

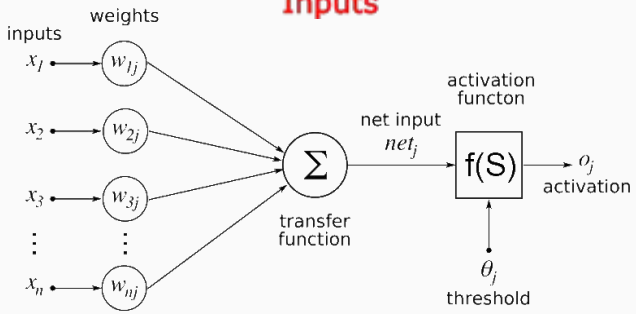

Content
Timing
= Customer Preference
= Marketing Message

Artificial Neuron

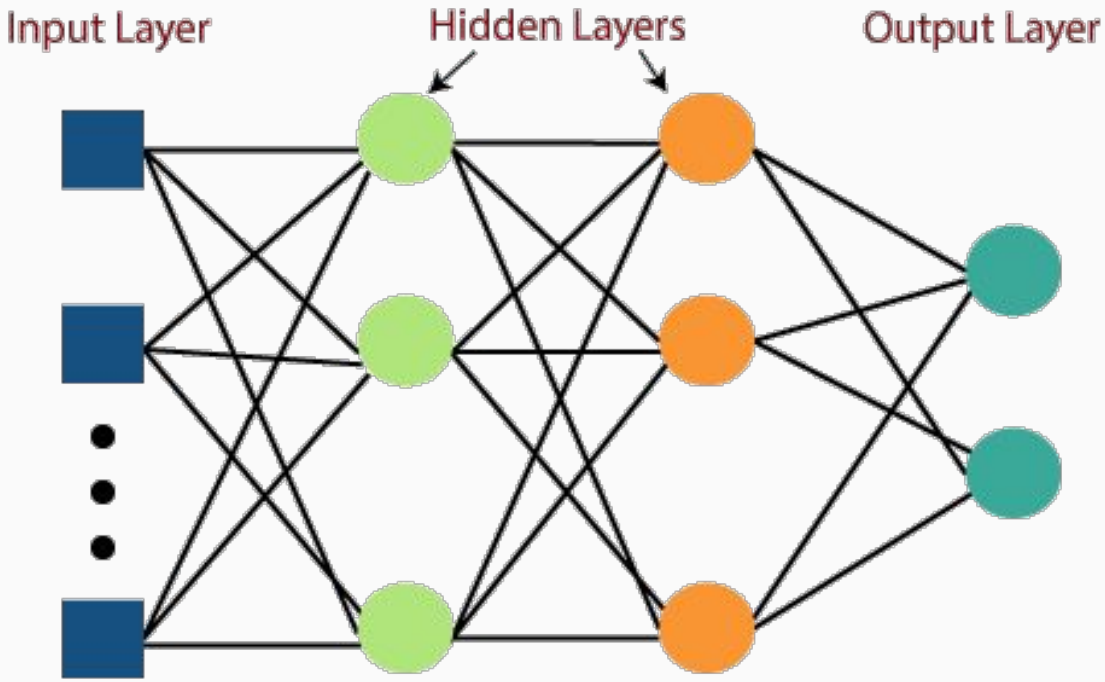
y = f (P i wi xi)
xi : inputs
wi : weights
f : non-linearity

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

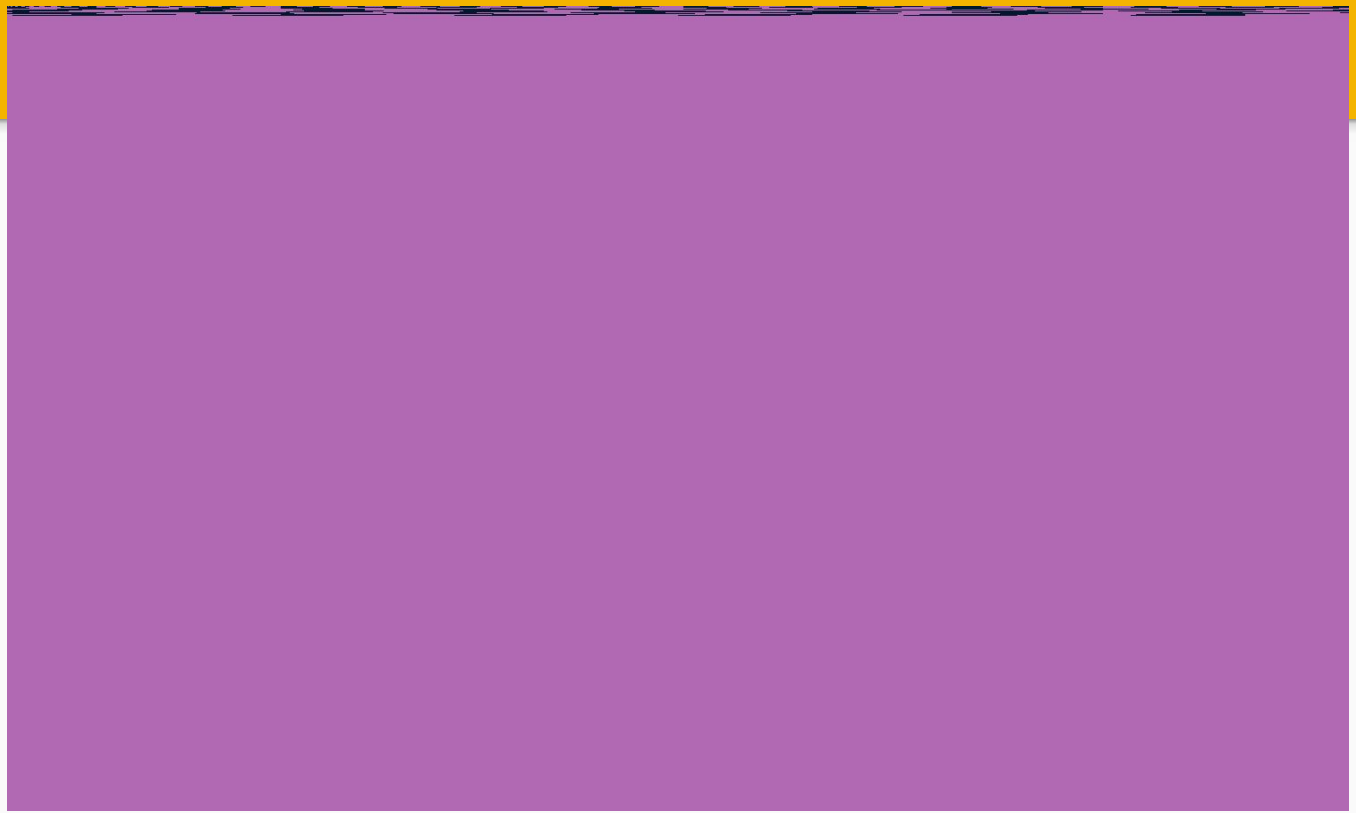




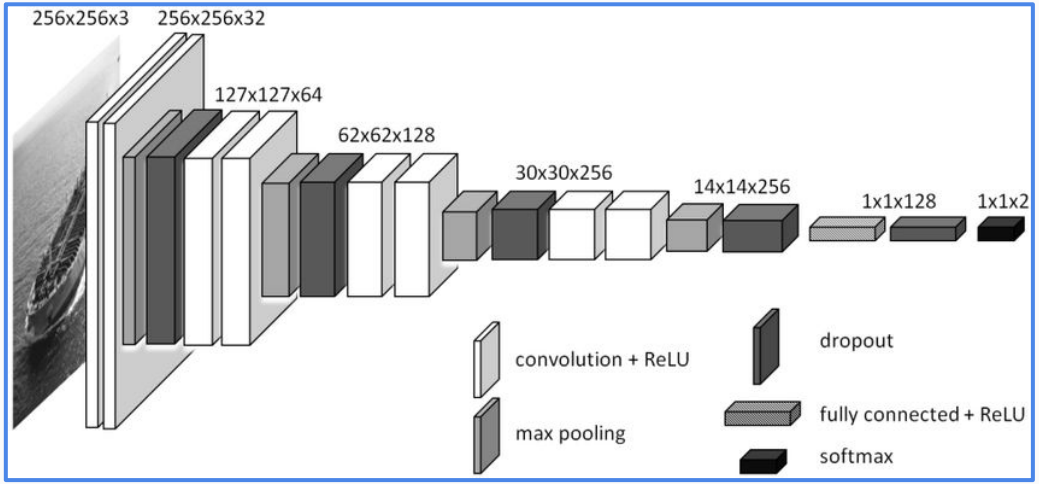
Organization in layers



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

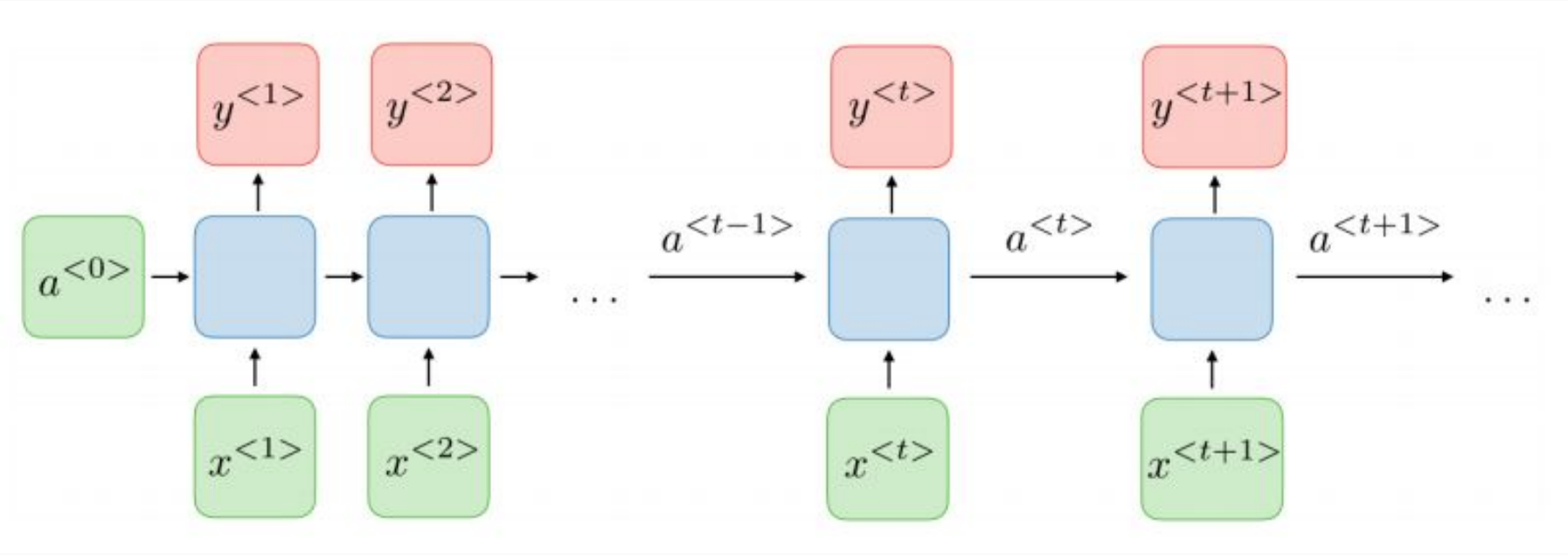


Image Source: Afshine Amidi and Shervine Amidi
November25, 2018

Creating Model - Example CNN

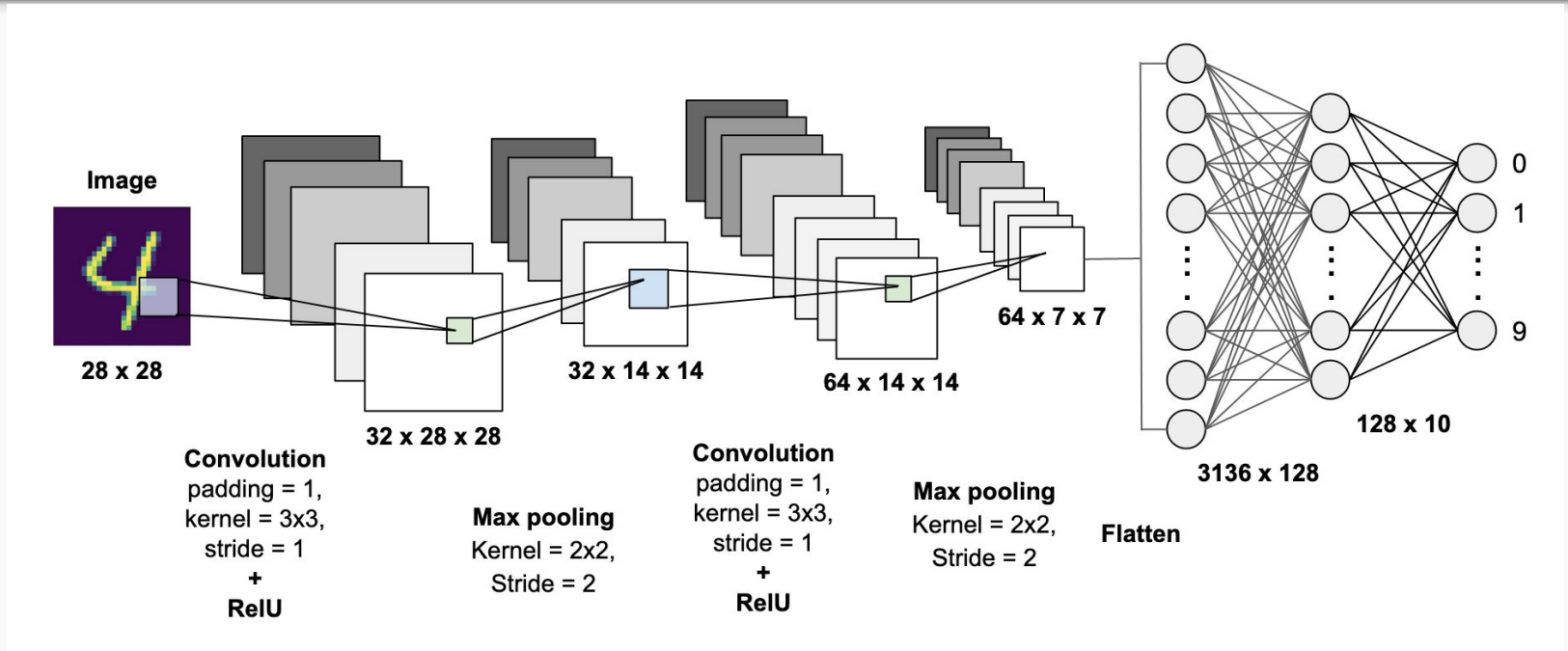
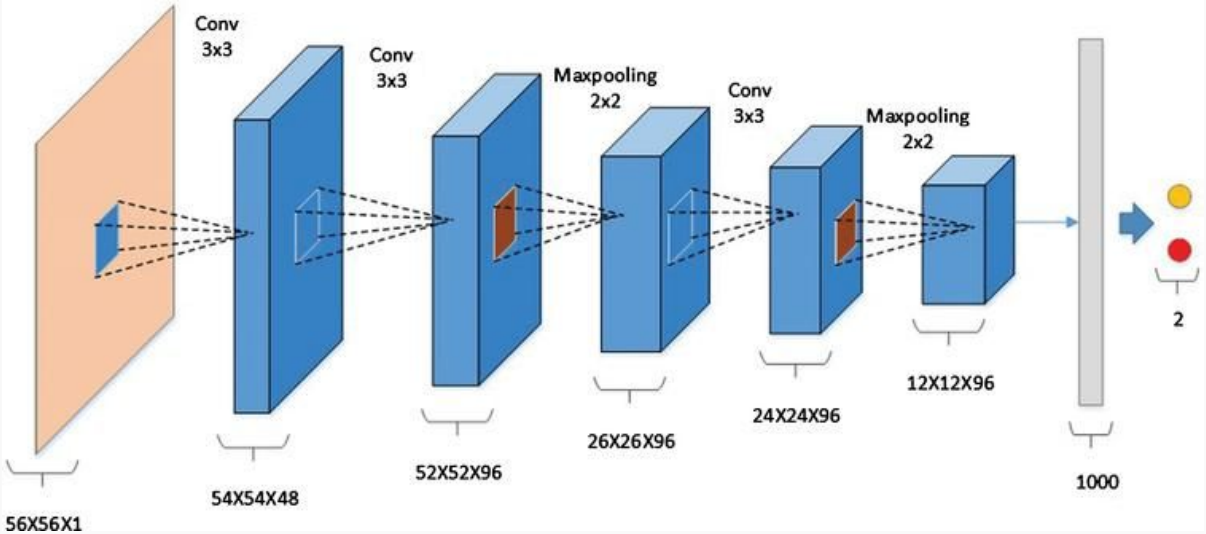
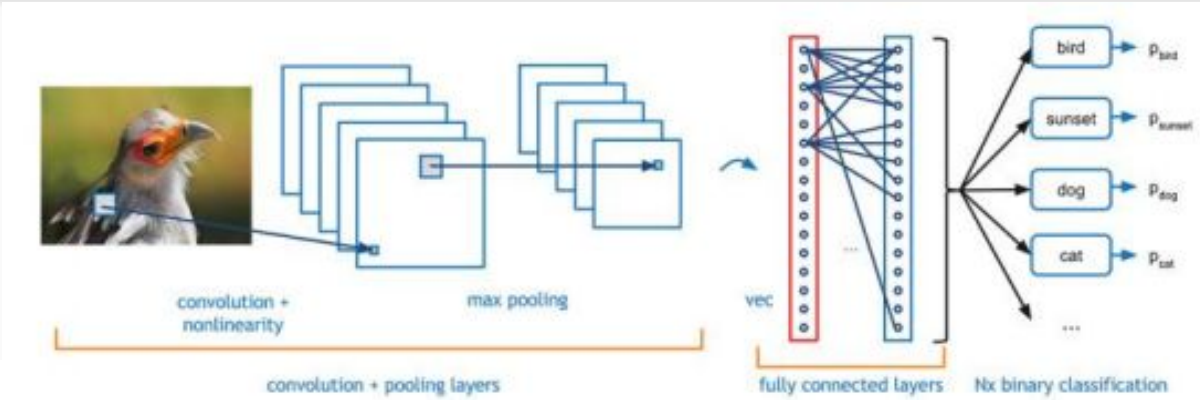


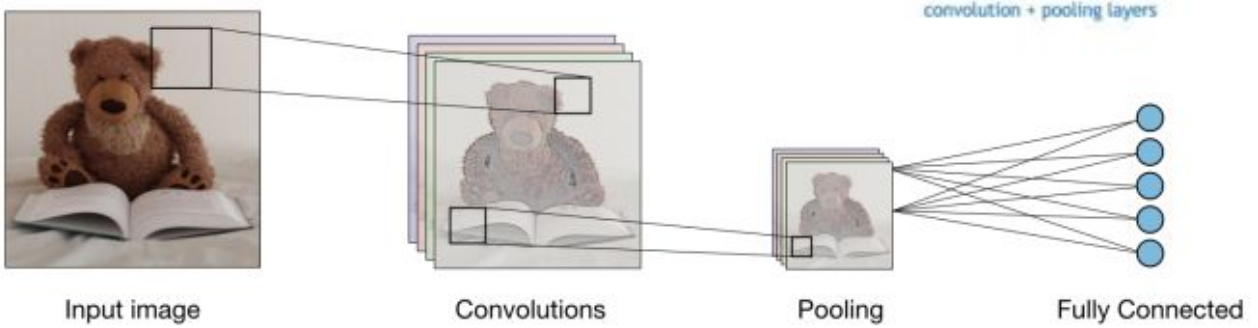
Image Source: Medium, becominghuman.ai , Shreyak

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



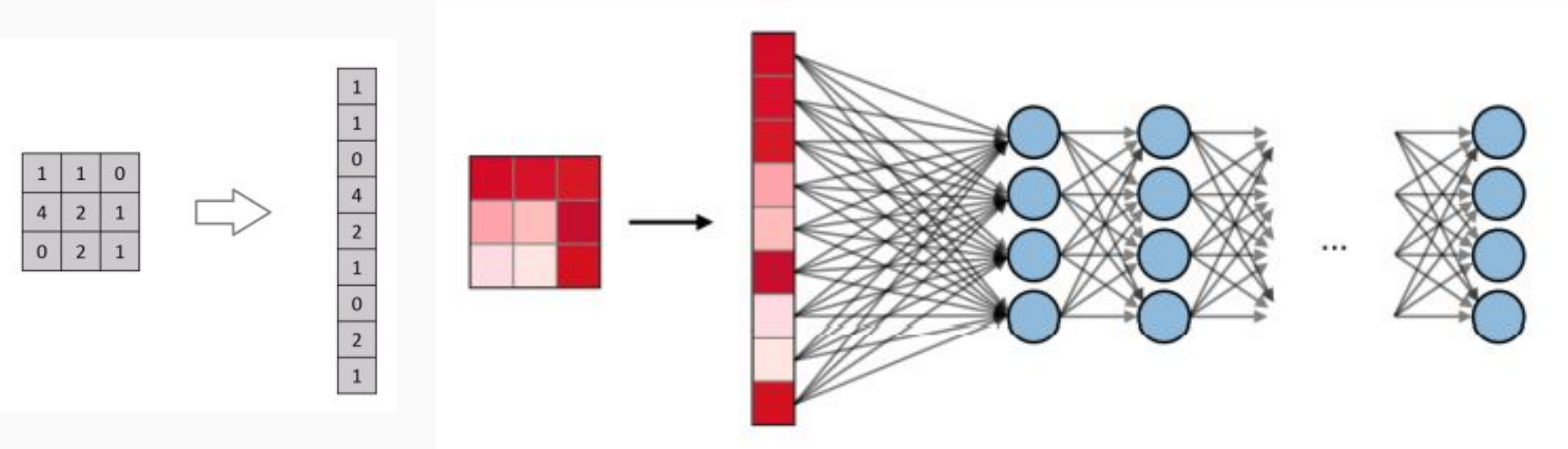
Convolutional Neural Networks





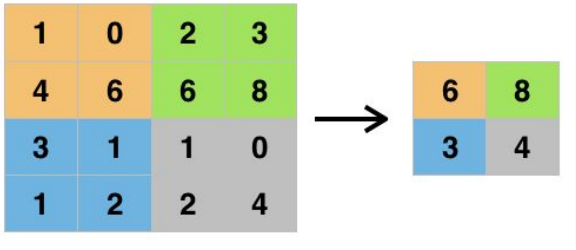
Source: Afshine Amidi and Shervine Amidi
November25, 2018

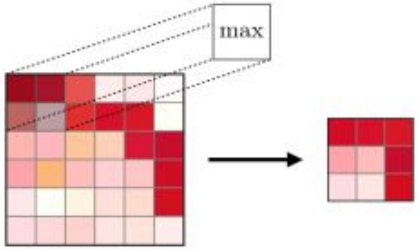
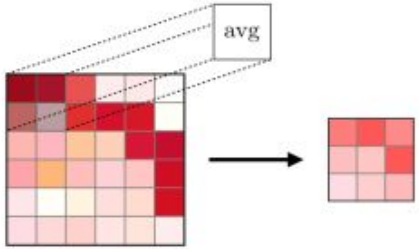
Flatten and Fully Connected



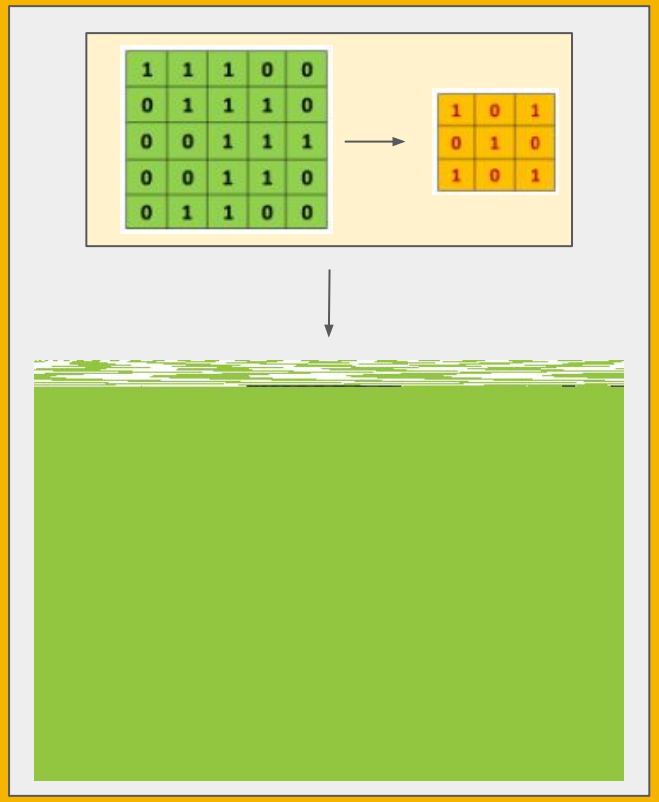
Source: Afshine Amidi and Shervine Amidi
November25, 2018

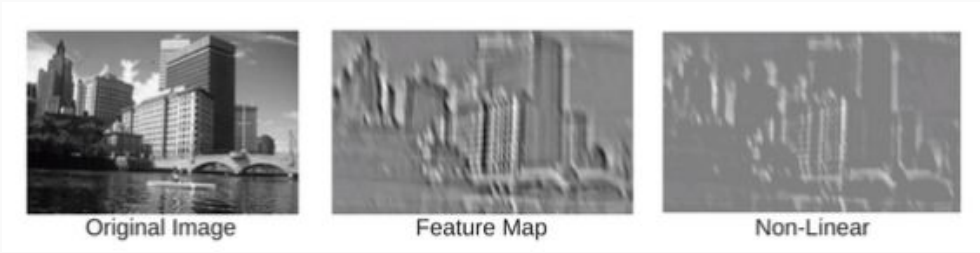
Pooling



	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		
Comments	- Preserves detected features - Most commonly used	- Downsamples feature map - Used in LeNet

Source 1: medium/@tuncerergin
Source 2: Afshine Amidi and Shervine Amidi
November25, 2018

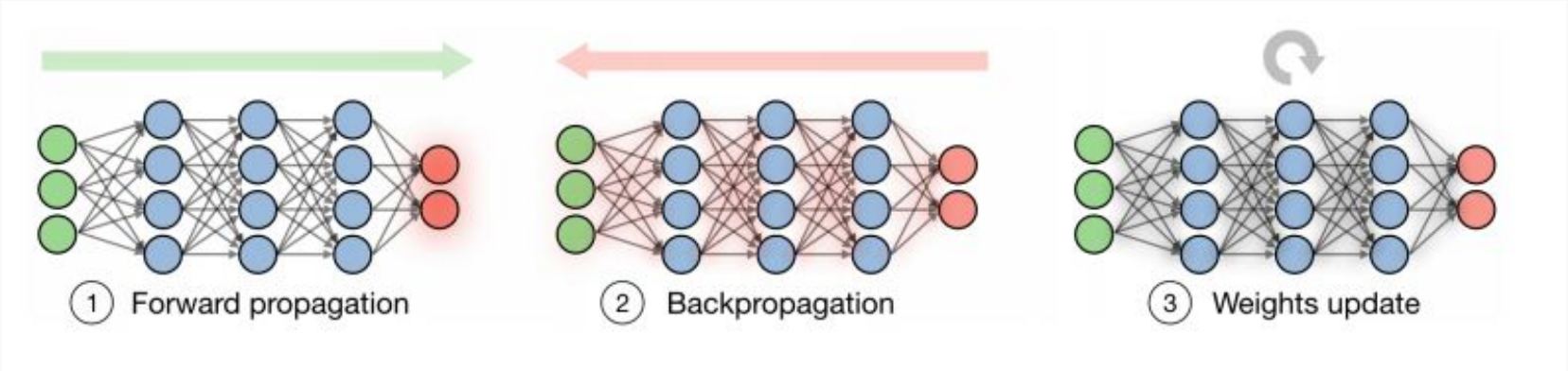


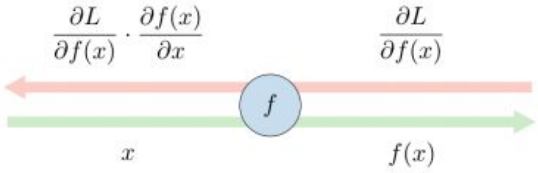


Source: medium/@tuncerergin

Backpropagation and Updating weights

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





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

$$w \longleftarrow w - \alpha \frac{\partial L(z,y)}{\partial w}$$

Source: Afshine Amidi and Shervine Amidi
November25, 2018

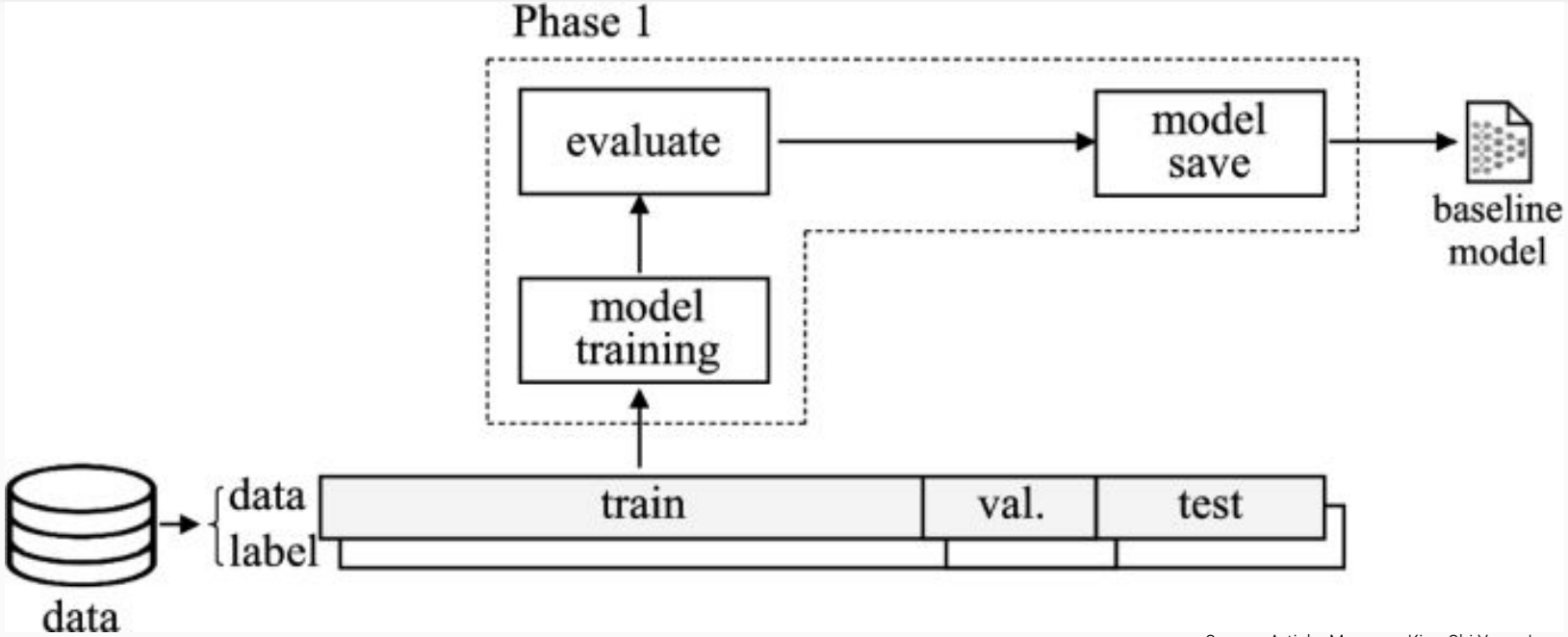
Data Augmentation

Original	Flip	Rotation	Random crop
			
- 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
			
- Nuances of RGB is slightly changed - Captures noise that can occur with light exposure	- 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

Source: Afshine Amidi and Shervine Amidi
November25, 2018

Create General Baseline Model



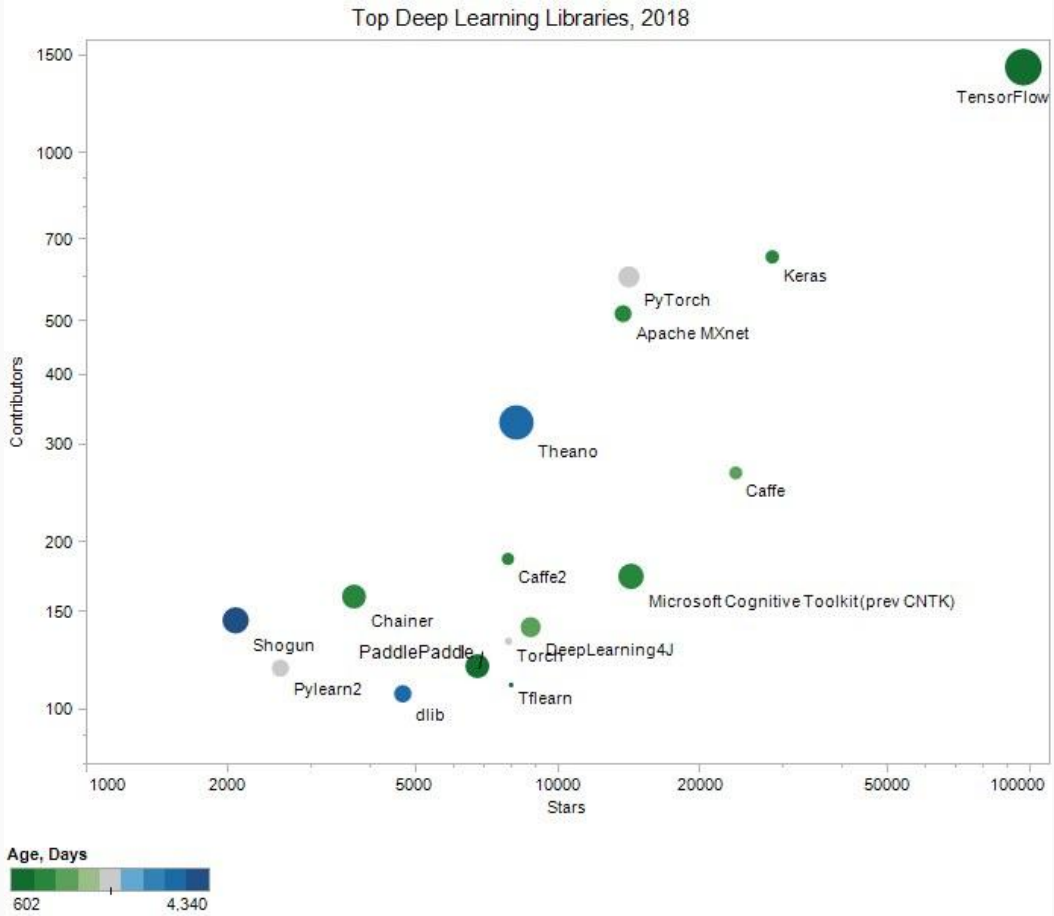
Source: Article, Mooseop Kim, Chi Yoon Jeong, Label-preserving data augmentation for mobile sensor data

02

DL Frameworks and Libraries

Some DL Frameworks

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

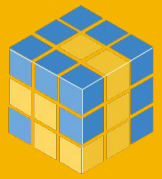


Some Most Popular and Useful Python Libraries

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

Numpy

The fundamental package for scientific computing with Python



NumPy

```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))
```

Source: cs231n.github.io

Pandas

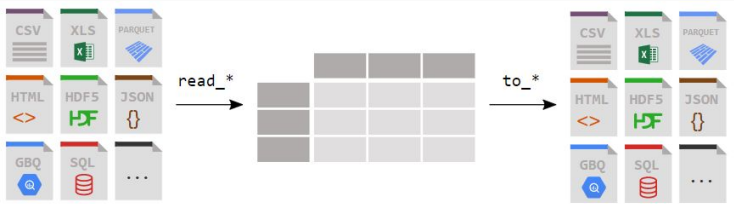
Pandas; open source data analysis and manipulation tool

pandas

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

```
In [3]: titanic
Out[3]:
     PassengerId  Survived  Pclass  ...     Fare Cabin  Embarked
0              1         0       3  ...   7.2500   NaN         S
1              2         1       1  ...  71.2833   C85         C
2              3         1       3  ...   7.9250   NaN         S
3              4         1       1  ...  53.1000  C123         S
4              5         0       3  ...   8.0500   NaN         S
..           ...       ...     ...  ...      ...   ...       ...
886          887         0       2  ...  13.0000   NaN         S
887          888         1       1  ...  30.0000   B42         S
888          889         0       3  ...  23.4500   NaN         S
889          890         1       1  ...  30.0000  C148         C
890          891         0       3  ...   7.7500   NaN         Q

[891 rows x 12 columns]
```



Source: pandas.pydata.org

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()
```

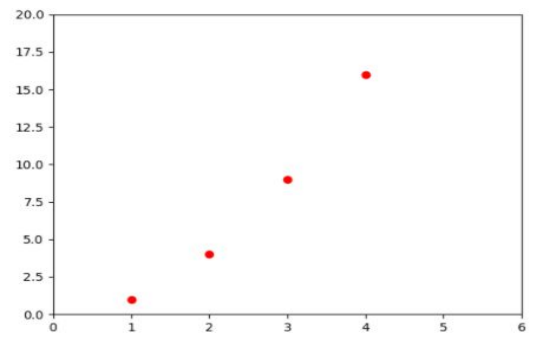
```
>>> print(digits.data)
[[ 0.   0.   5. ...   0.   0.   0.]
 [ 0.   0.   0. ...  10.   0.   0.]
 [ 0.   0.   0. ...  16.   9.   0.]
 ...
 [ 0.   0.   1. ...   6.   0.   0.]
 [ 0.   0.   2. ...  12.   0.   0.]
 [ 0.   0.  10. ...  12.   1.   0.]]
```

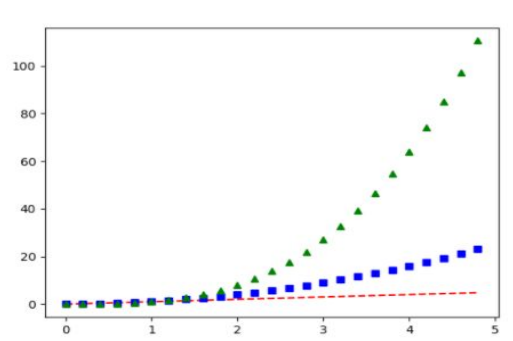
```
>>> digits.target
array([0, 1, 2, ..., 8, 9, 8])
```

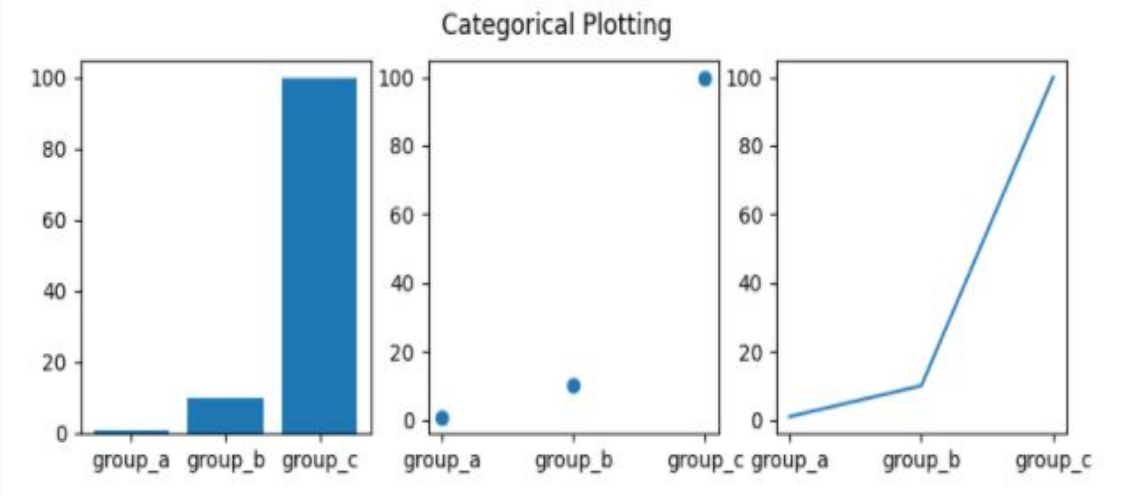
Source: scikit-learn.org

Matplotlib

Visualization with Python







Source: matplotlib.org

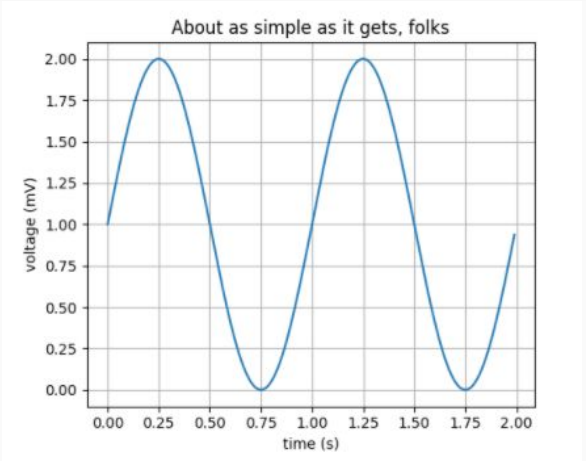
Matplotlib

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()
```



Source: matplotlib.org

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

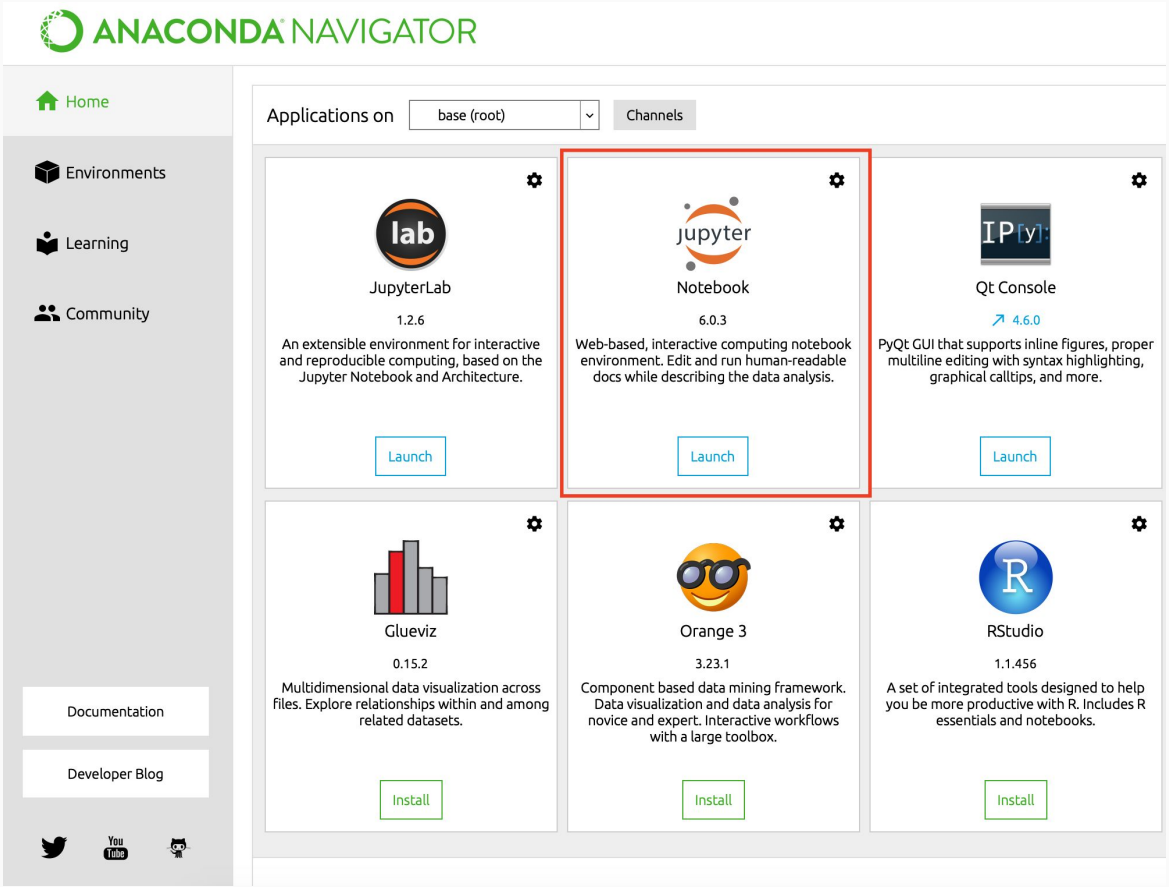
Computer Needed:

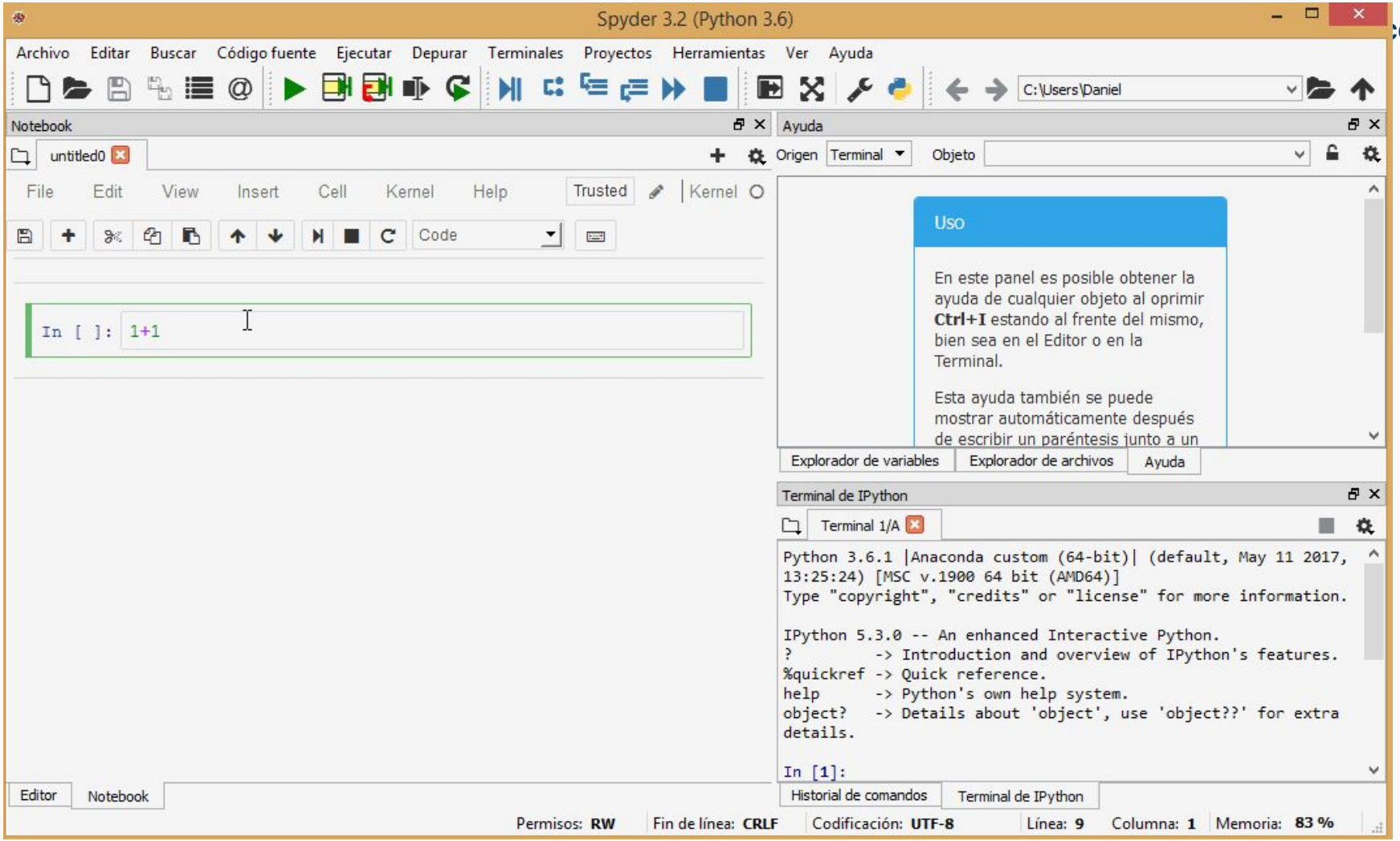
- GPU Supported, Cuda, cuDNN

Container Engine

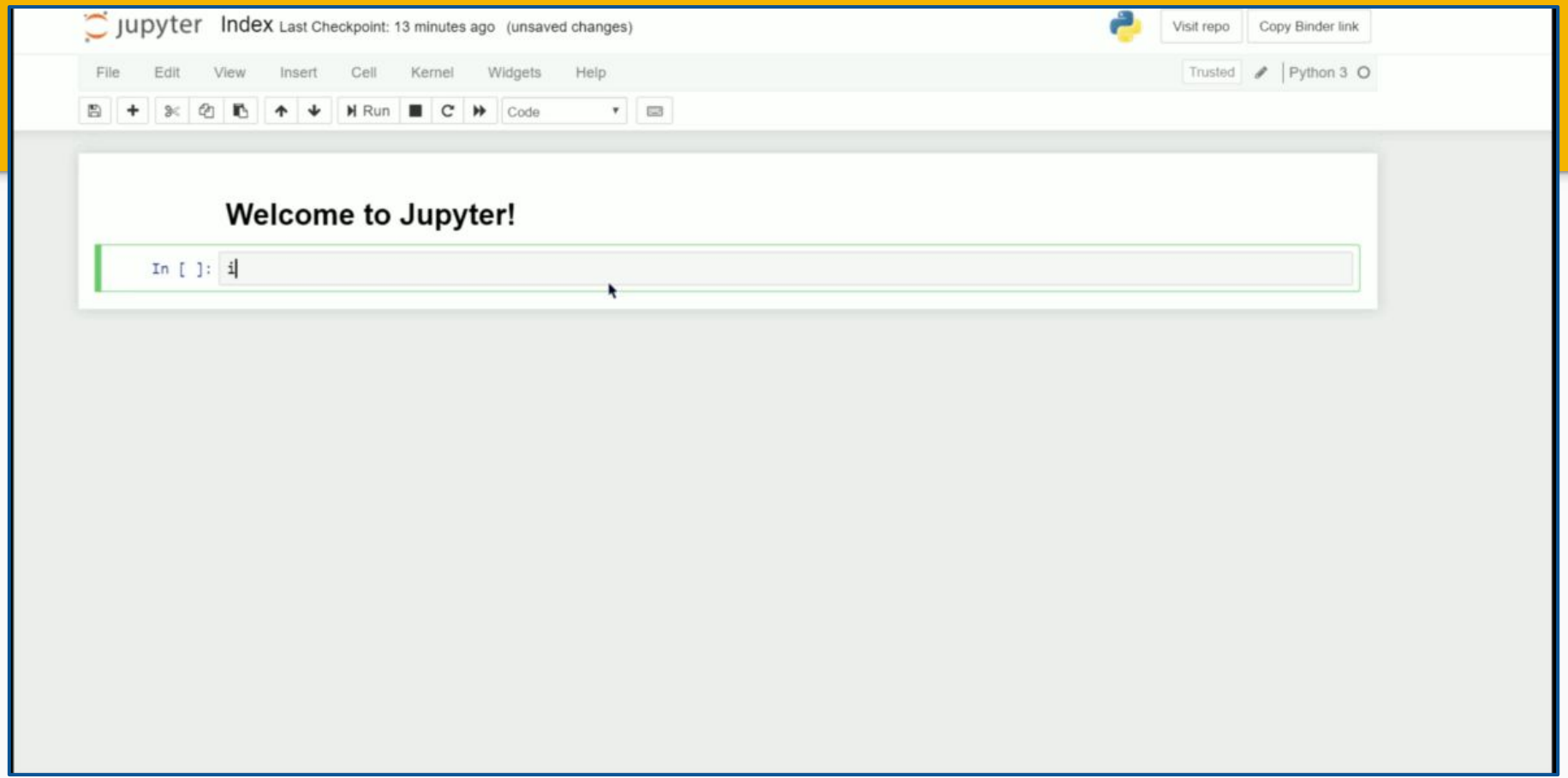
- Docker

Anaconda Navigator

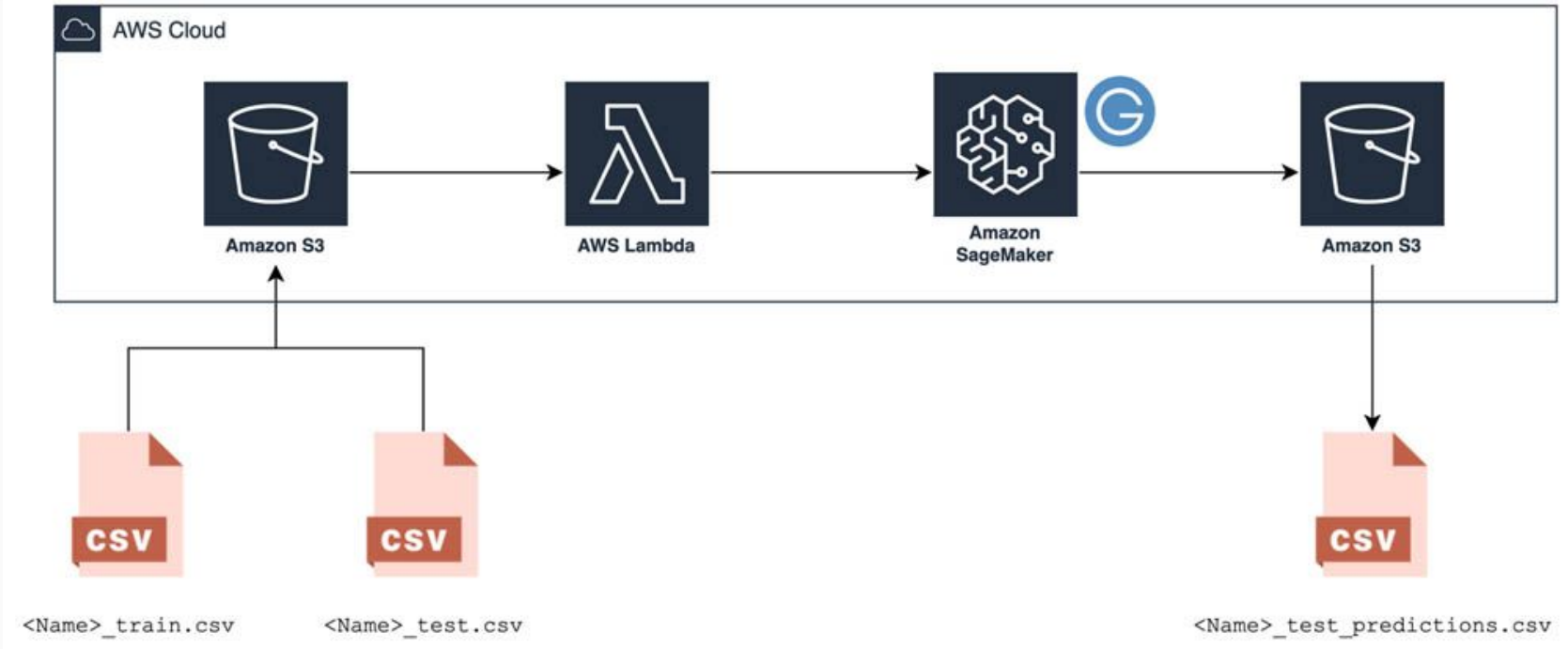



Spyder 3.2 (Python 3.6)
Archivo Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas Ver Ayuda
C:\Users\Daniel
Notebook
untitled0
File Edit View Insert Cell Kernel Help
Trusted
Kernel
Code
In [ ]: 1+1
Ayuda
Origen Terminal
Objeto
Uso
En este panel es posible obtener la ayuda de cualquier objeto al oprimir Ctrl+I estando al frente del mismo, bien sea en el Editor o en la Terminal.
Esta ayuda también se puede mostrar automáticamente después de escribir un paréntesis junto a un
Explorador de variables
Explorador de archivos
Ayuda
Terminal de IPython
Terminal 1/A
Python 3.6.1 |Anaconda custom (64-bit)| (default, May 11 2017, 13:25:24) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.
IPython 5.3.0 -- An enhanced Interactive Python.
? -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extra details.
In [1]:
Editor
Notebook
Historial de comandos
Terminal de IPython
Permisos: RW
Fin de línea: CRLF
Codificación: UTF-8
Línea: 9
Columna: 1
Memoria: 83 %

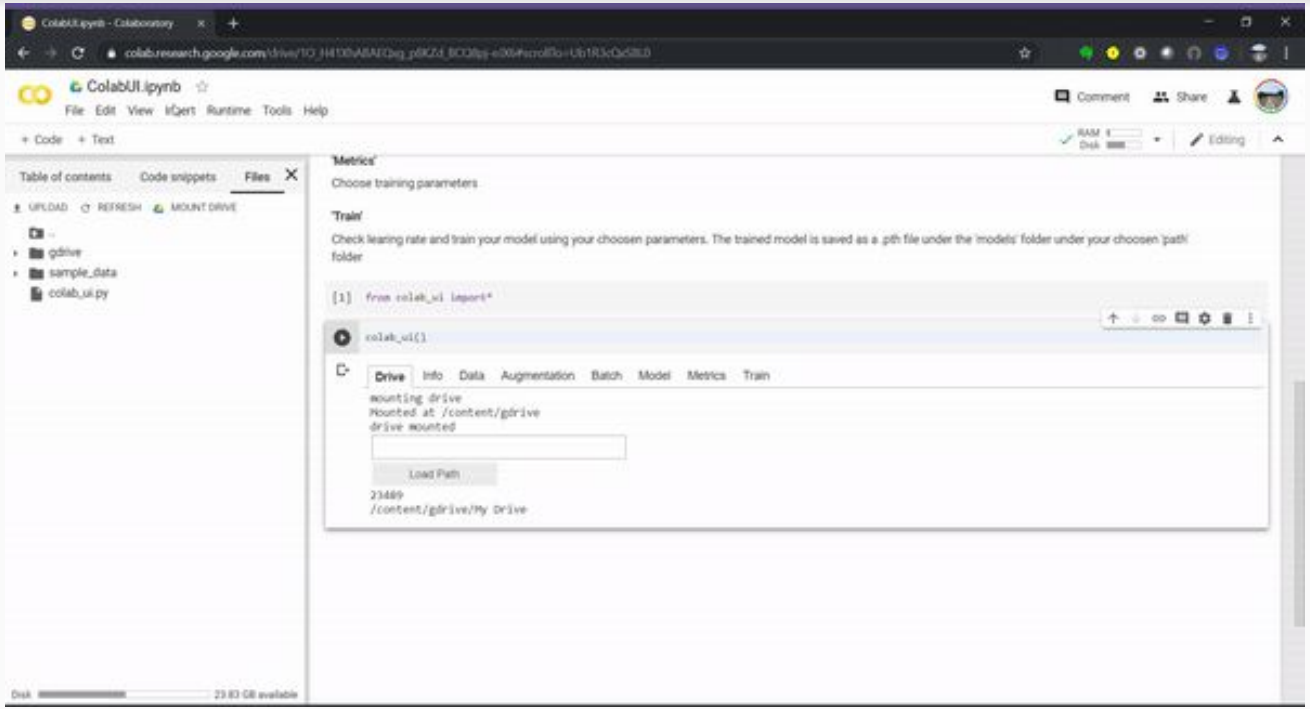
Jupyter Notebook



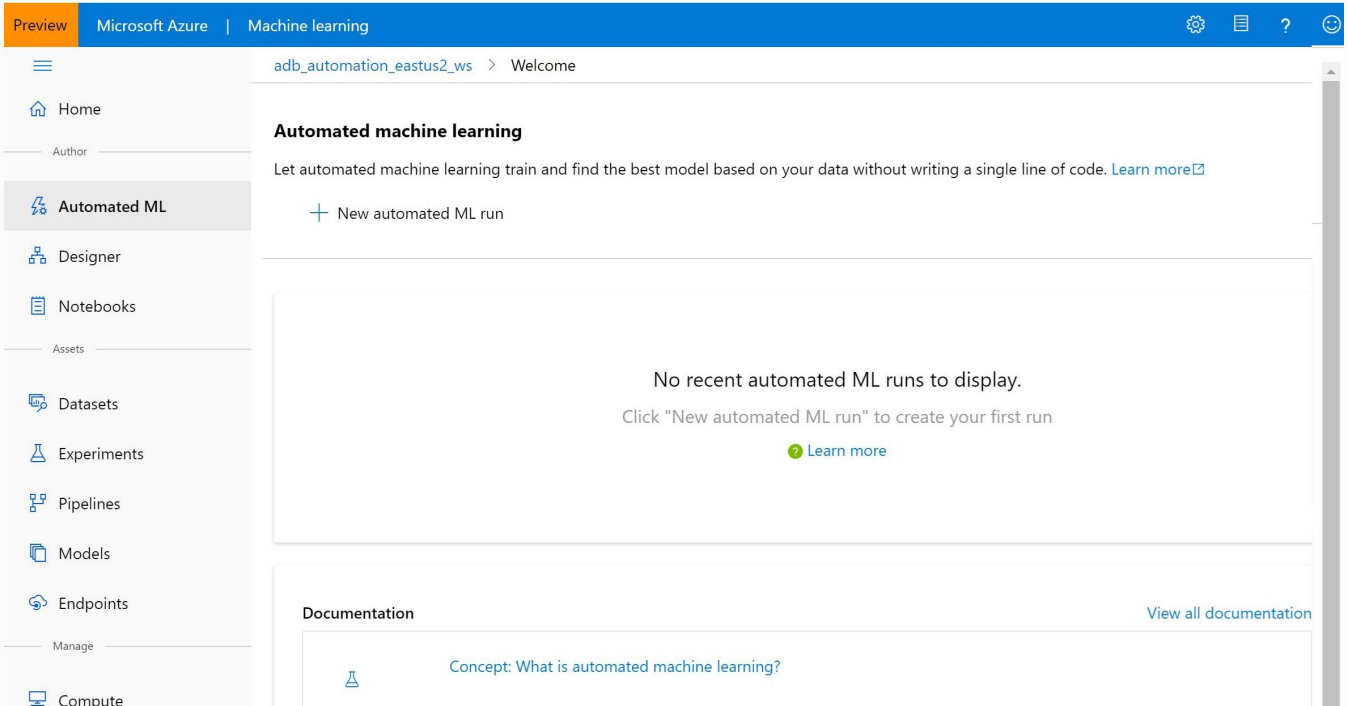
AWS


AWS Cloud
Amazon S3
AWS Lambda
Amazon SageMaker
Amazon S3
csv
csv
csv
<Name>_train.csv
<Name>_test.csv
<Name>_test_predictions.csv

Google Colab



Microsoft Azure



Docker

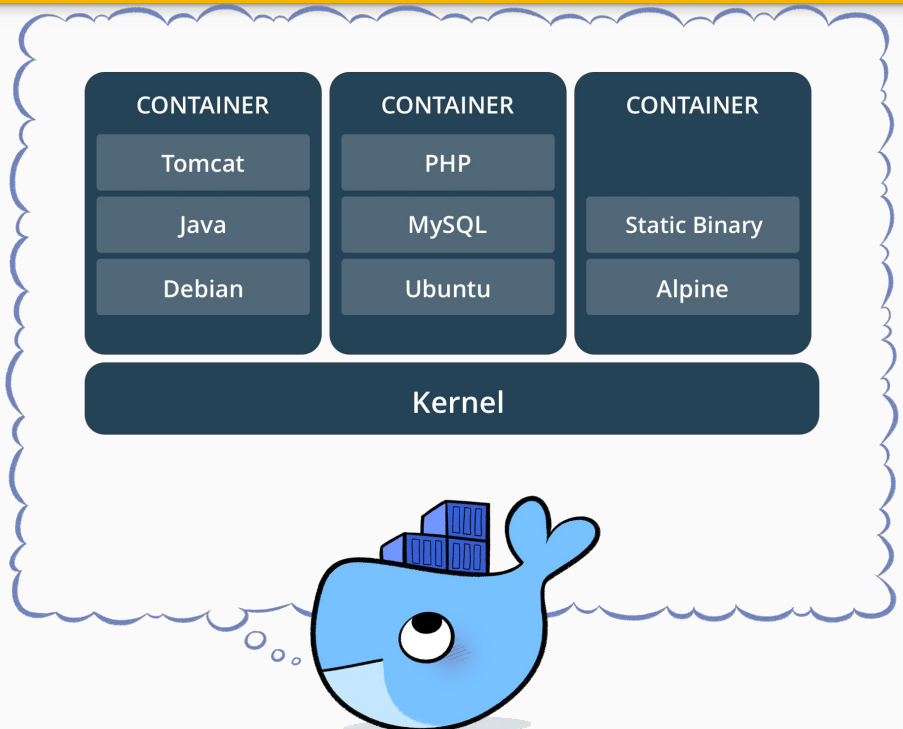


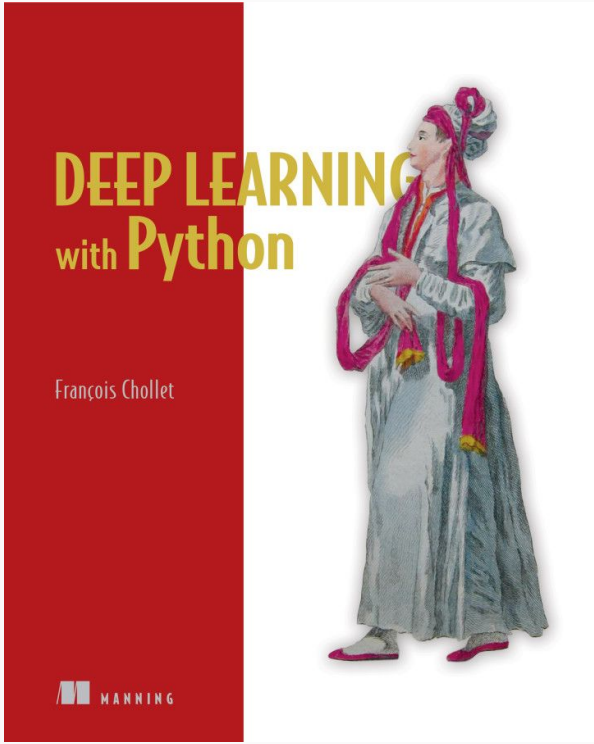
Image source: GeoNode

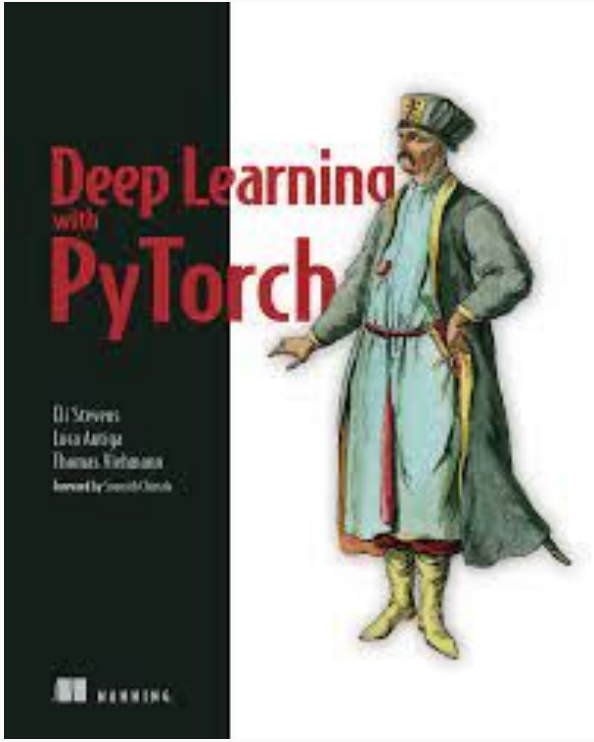
04

Suggested Resources

Suggested Resources -Book-

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





coursera

Browse > Data Science > Machine Learning

Offered By

Stanford

Machine Learning

★★★★★ 4.9 156.670 ratings • 40.081 reviews

Andrew Ng TOP INSTRUCTOR

Enroll for Free
Started Mar 15

Financial aid available

4,001,864 already enrolled

kaggle

stackoverflow

GitHub

THANKS

Does anyone have any questions?

iremkomurcubm@gmail.com
iremkomurcu.com