

Introduction to Deep Learning

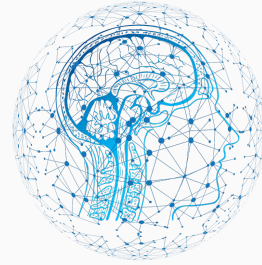
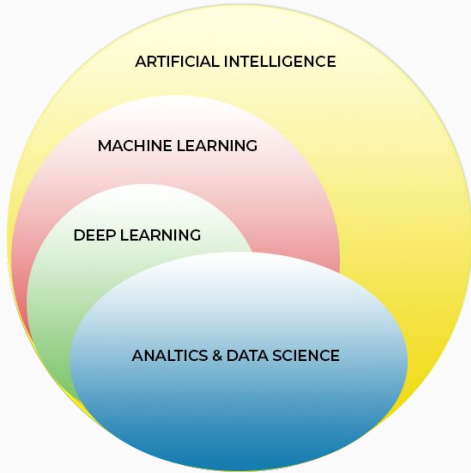
Deep Learning in Remote Sensing

Episode-1

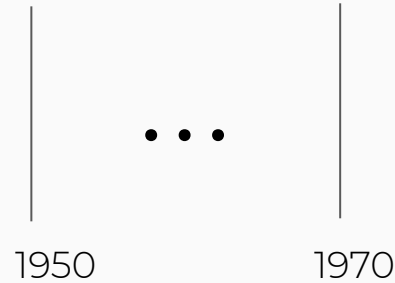
İrem KÖMÜRCÜ
iremkomurcu.com
iremkomurcubm@gmail.com

01

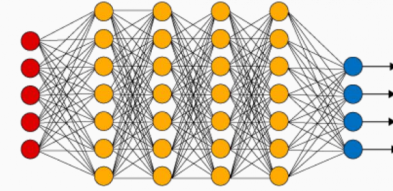
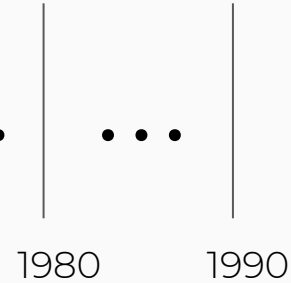
AI - DL - ML - Data Terms



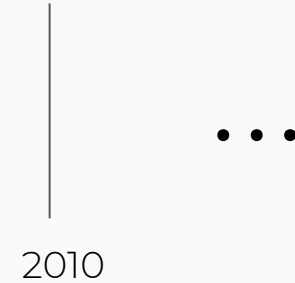
Artificial Neural Network

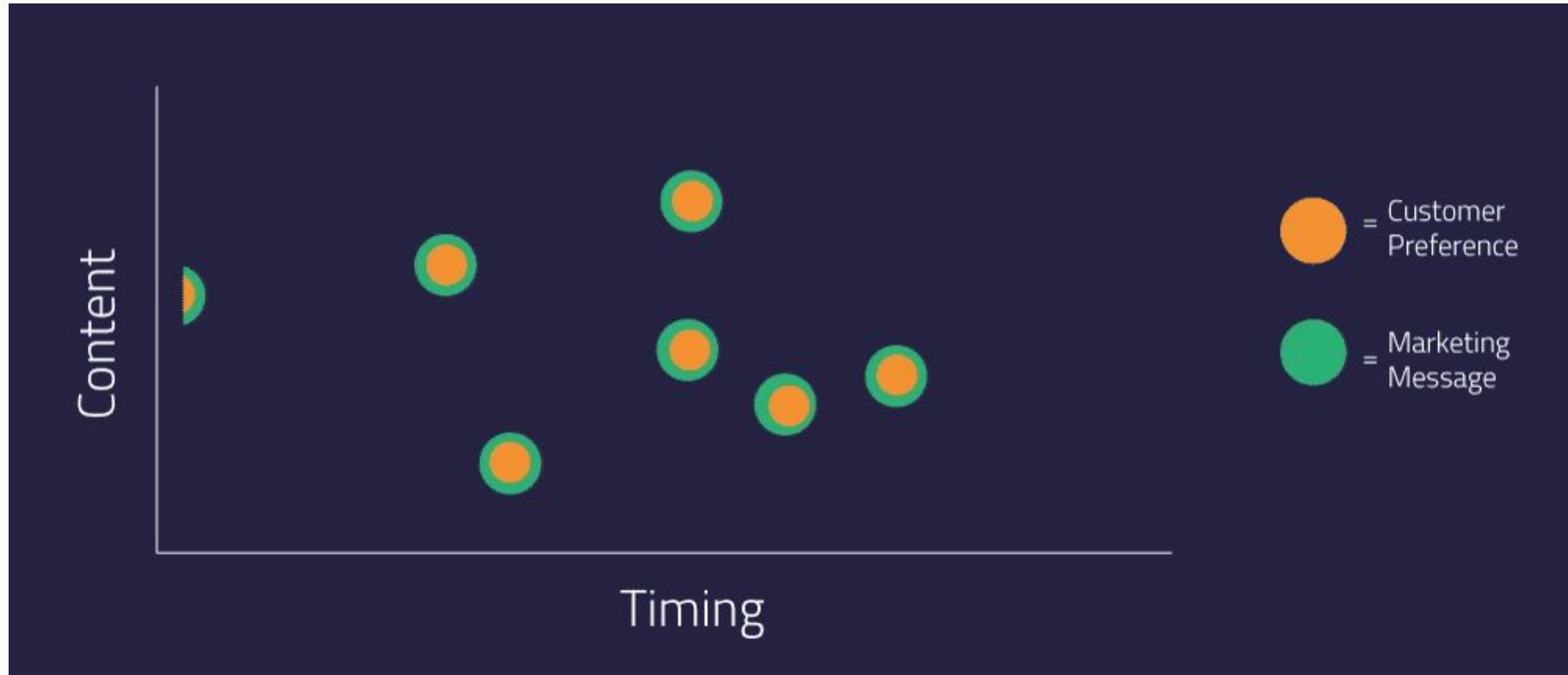


Machine Learning



Deep Learning

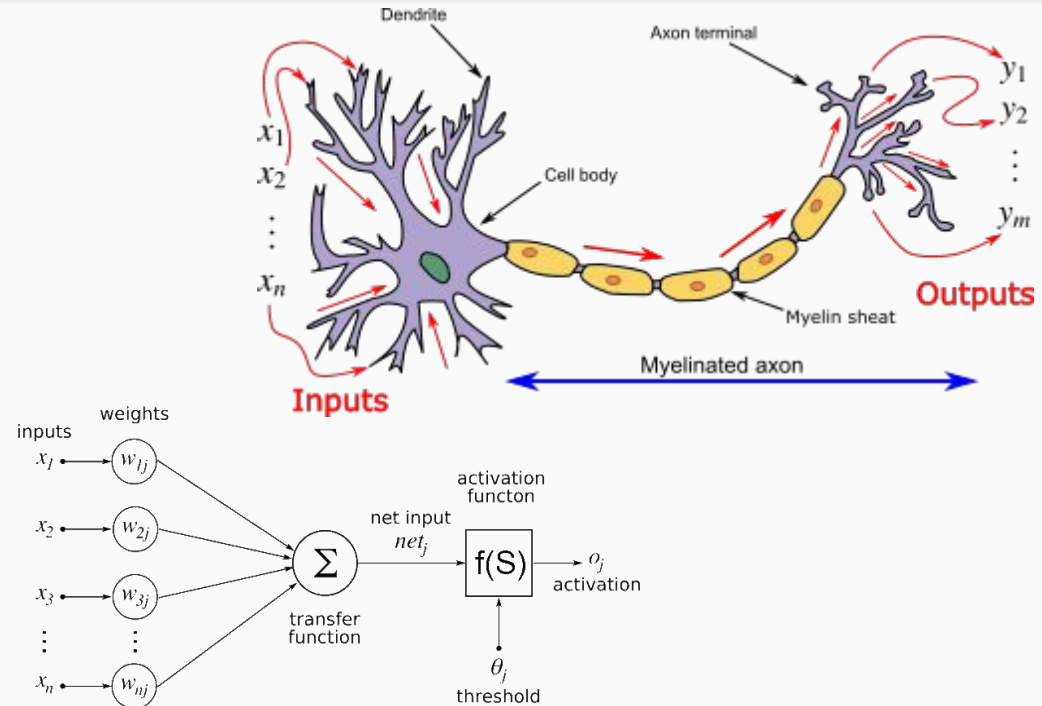




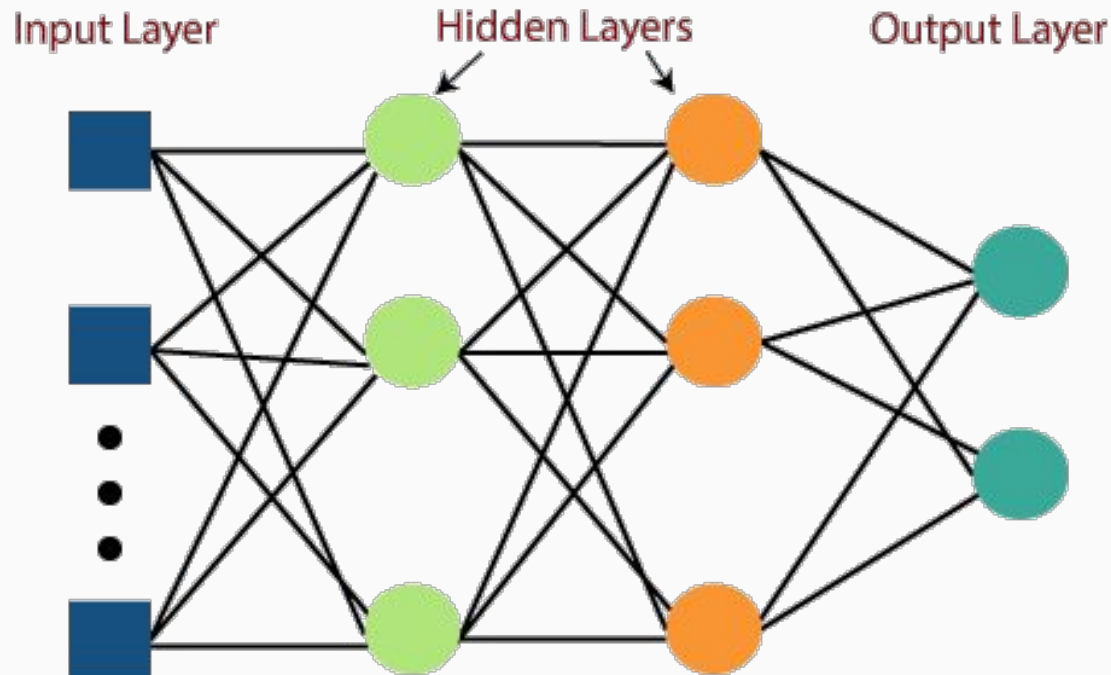
Artificial Neuron

$y = f(\sum w_i x_i)$
 x_i : inputs
 w_i : weights
 f : non-linearity

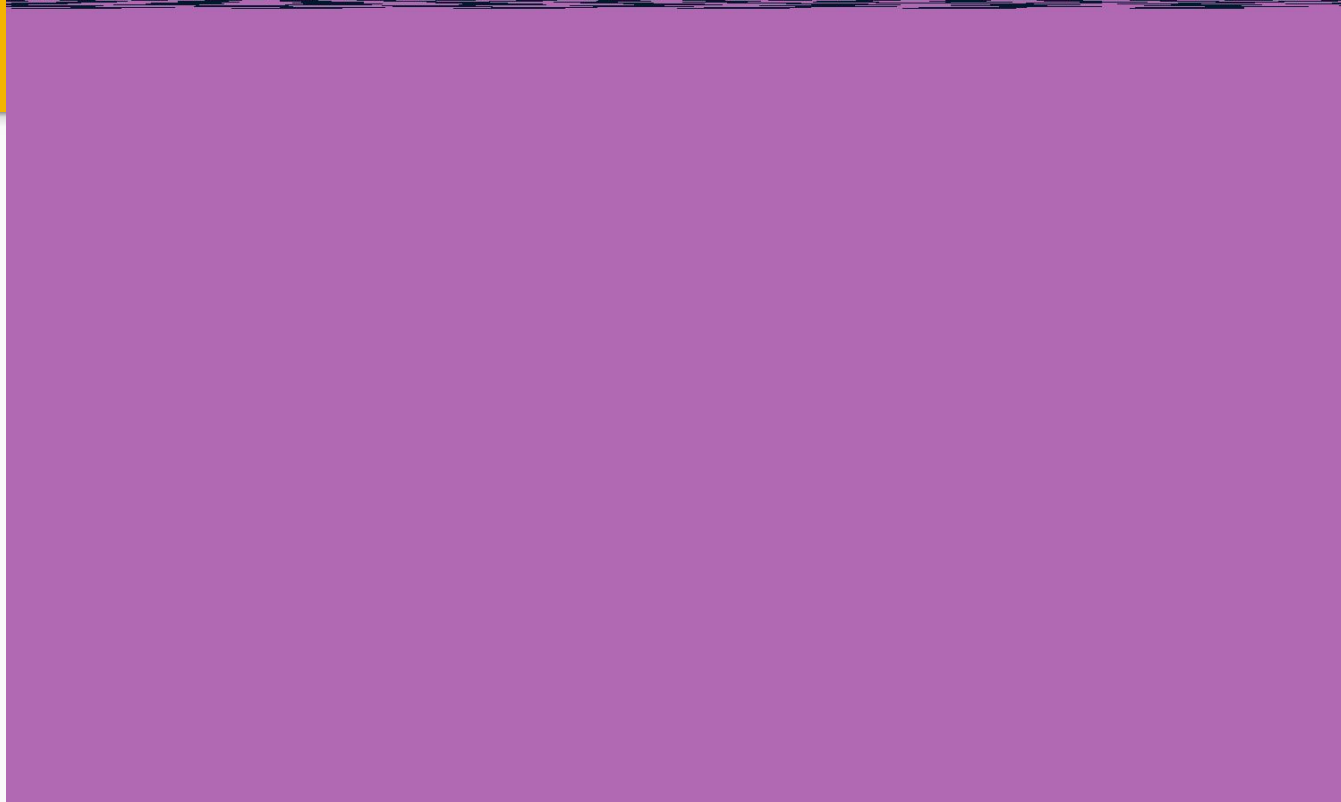
$$Y = \sum (\text{weight} * \text{input}) + \text{bias}$$



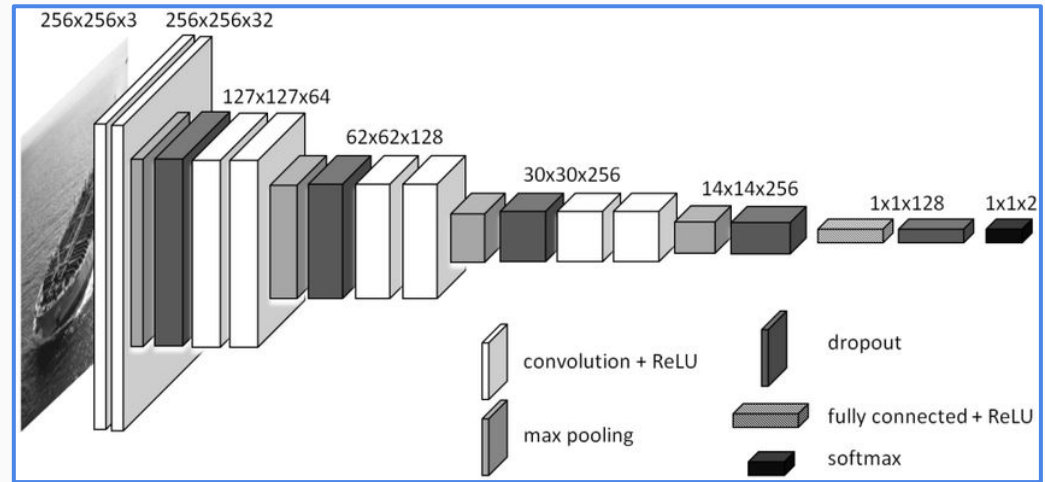
Organization in layers



Deep Learning and Classification

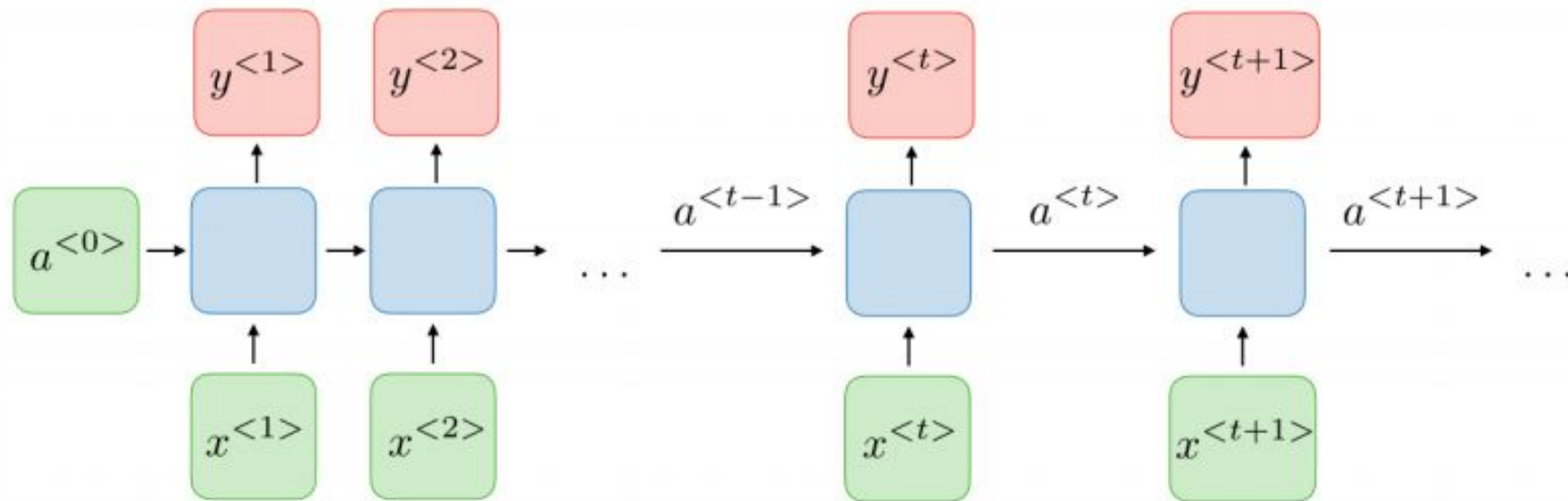


Types of Deep Learning Algorithms

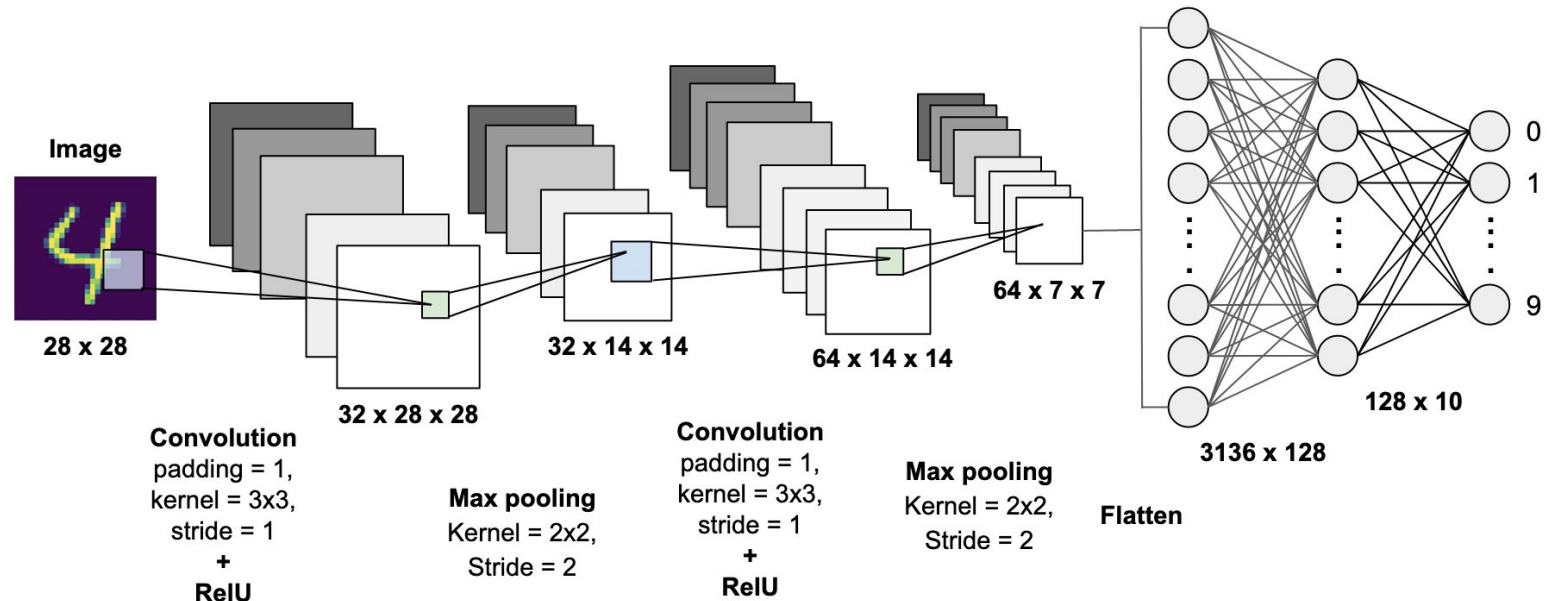


- Convolutional Neural Networks (CNNs)
- Long Short Term Memory Networks (LSTMs)
- Recurrent Neural Networks (RNNs)
- Generative Adversarial Networks (GANs)
- Radial Basis Function Networks (RBFNs)
- Multilayer Perceptrons (MLPs)
- Self Organizing Maps (SOMs)
- Deep Belief Networks (DBNs)
- Restricted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

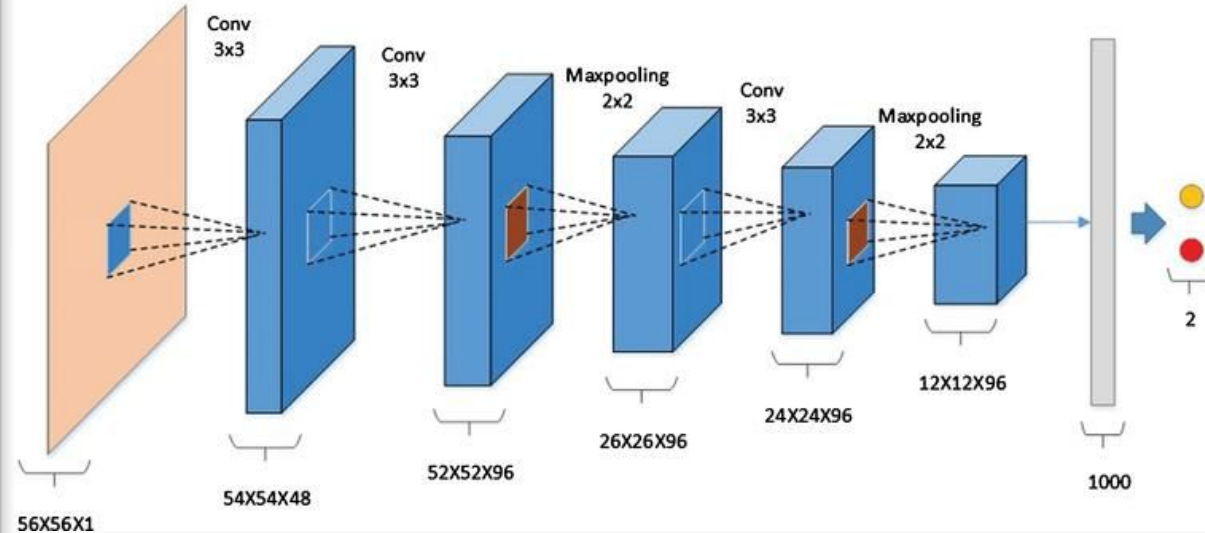
Recurrent Neural Networks



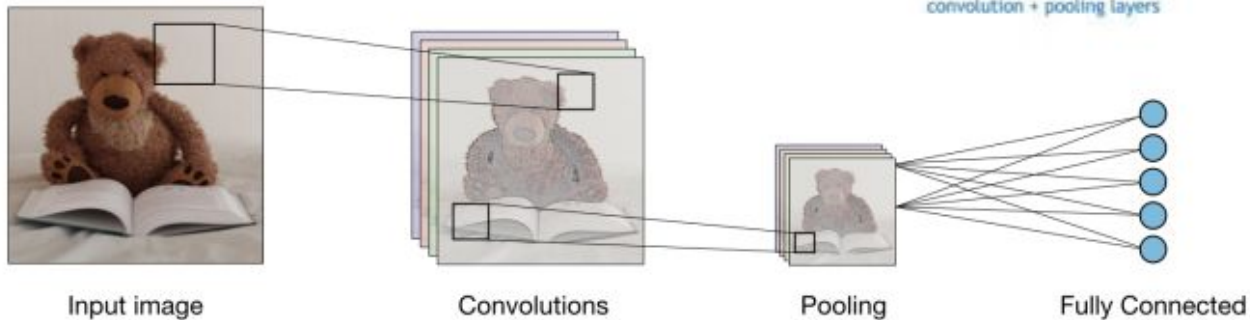
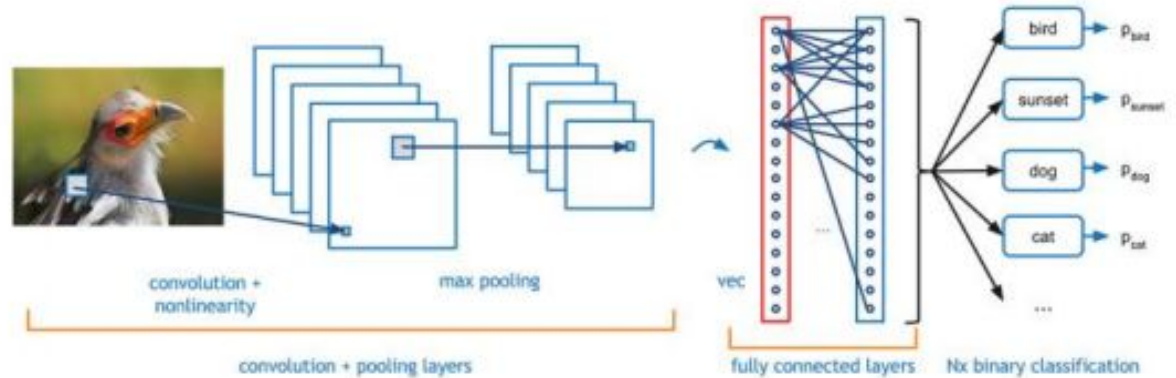
Creating Model - Example CNN



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



Convolutional Neural Networks

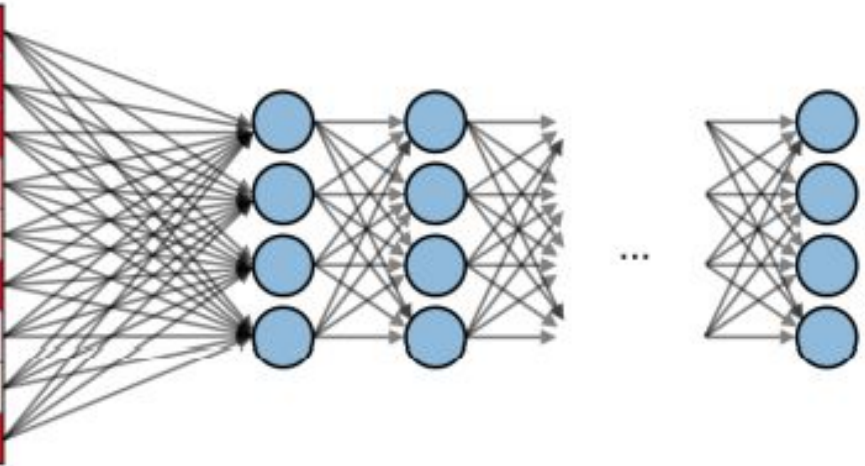
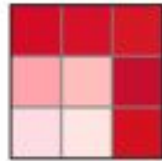


Flatten and Fully Connected

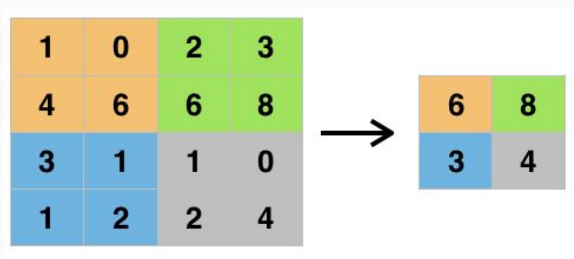
1	1	0
4	2	1
0	2	1

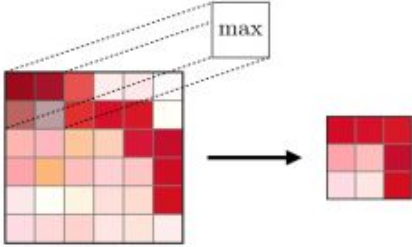
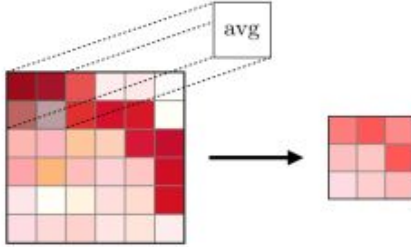


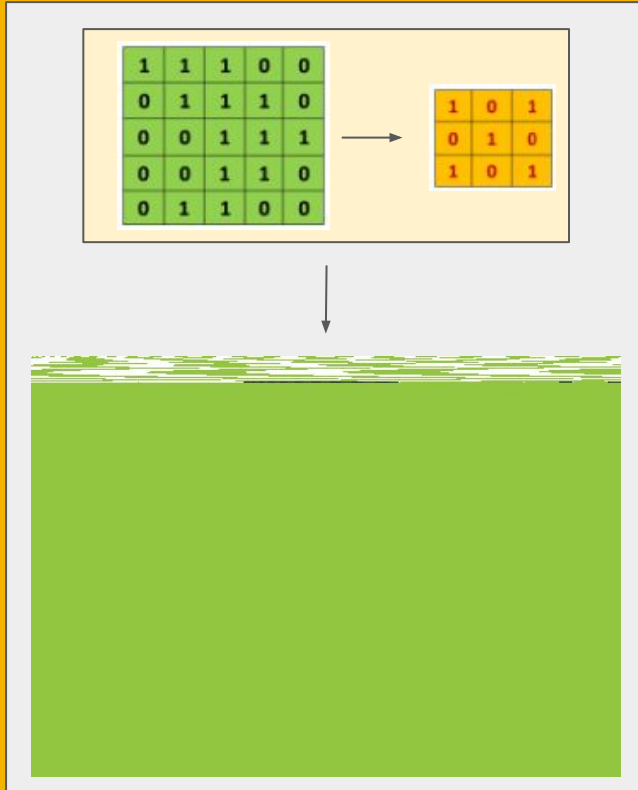
1
1
0
4
2
1
0
2
1



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	<ul style="list-style-type: none"> - Preserves detected features - Most commonly used 	<ul style="list-style-type: none"> - Downsamples feature map - Used in LeNet



Original Image



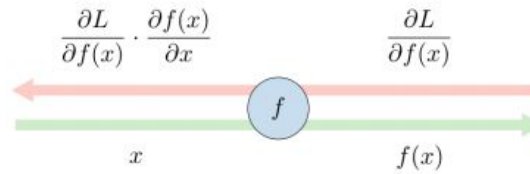
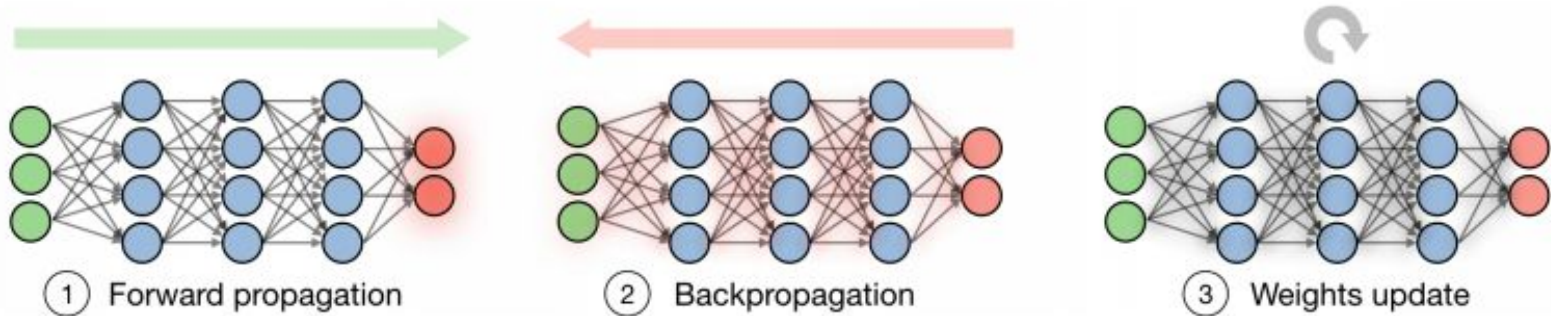
Feature Map



Non-Linear

Backpropagation and Updating weights





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







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

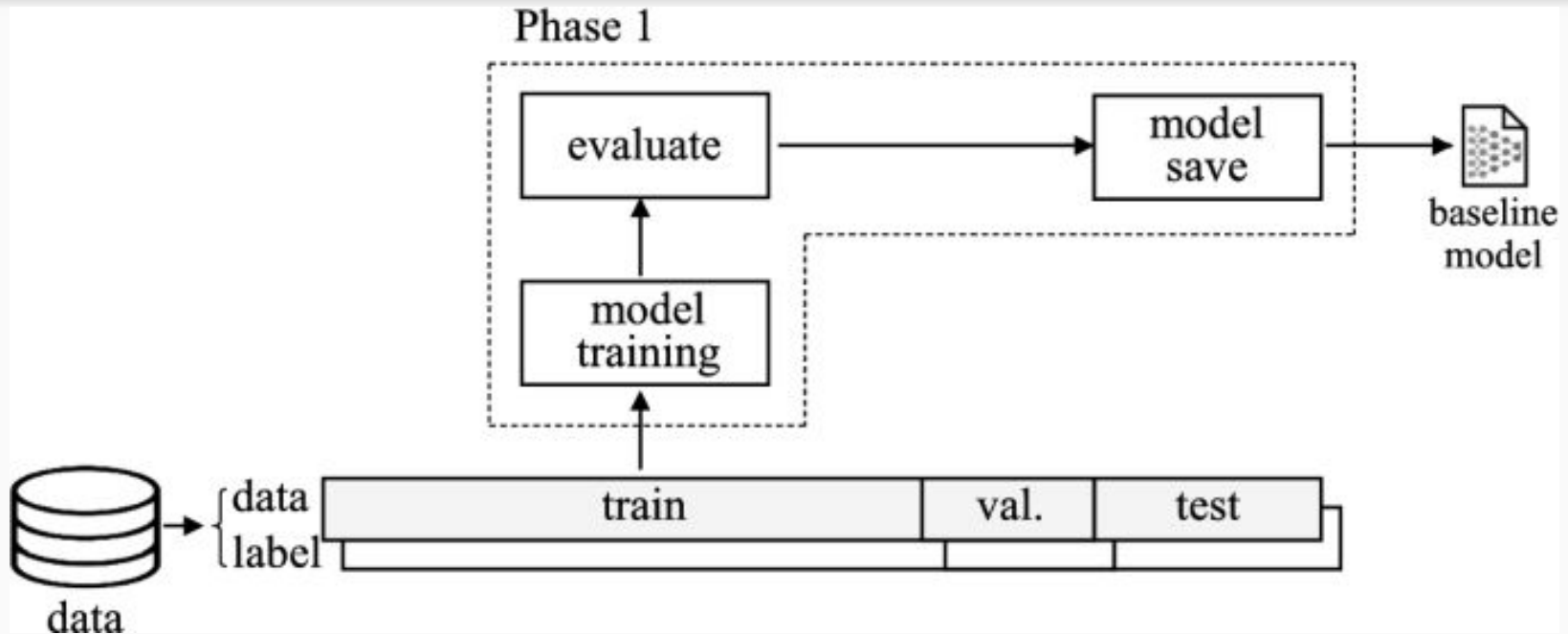
$$w \leftarrow w - \alpha \frac{\partial L(z,y)}{\partial w}$$

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

Create General Baseline Model

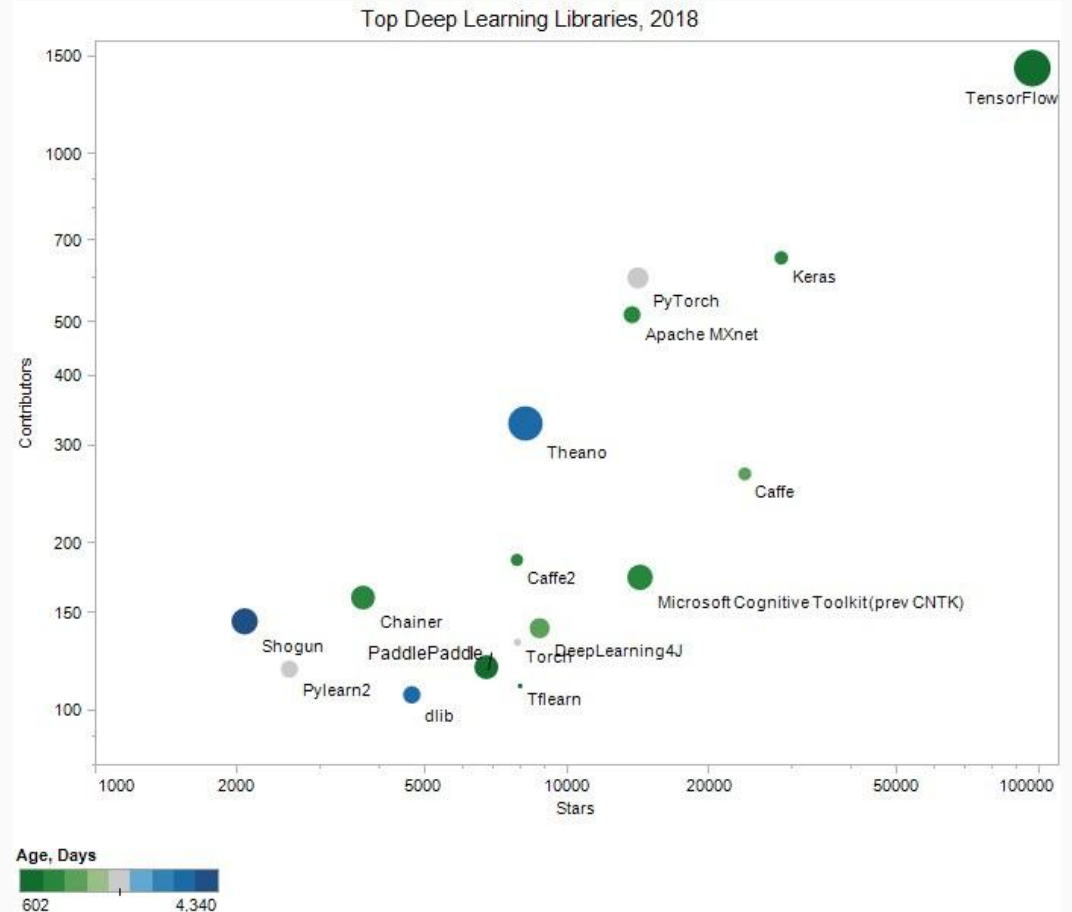


02

DL Frameworks and Libraries

Some DL Frameworks

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

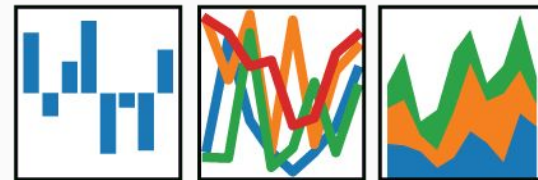


Some Most Popular and Useful Python Libraries

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



pandas

$$y_i t = \beta' x_{it} + \mu_i + \epsilon_{it}$$


NumPy

matplotlib

Numpy

The fundamental package for scientific computing with Python



NumPy

```
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



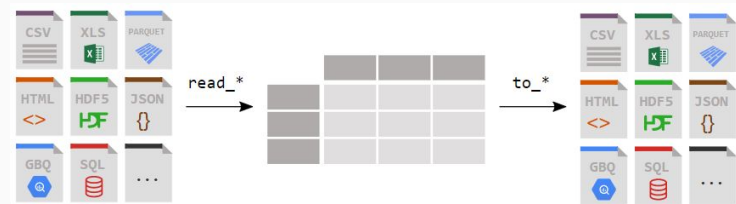
```
In [2]: titanic = pd.read_csv("data/titanic.csv")
```

```
In [3]: titanic
```

```
Out[3]:
```

	PassengerId	Survived	Pclass	...	Fare	Cabin	Embarked
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]
```



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()
```

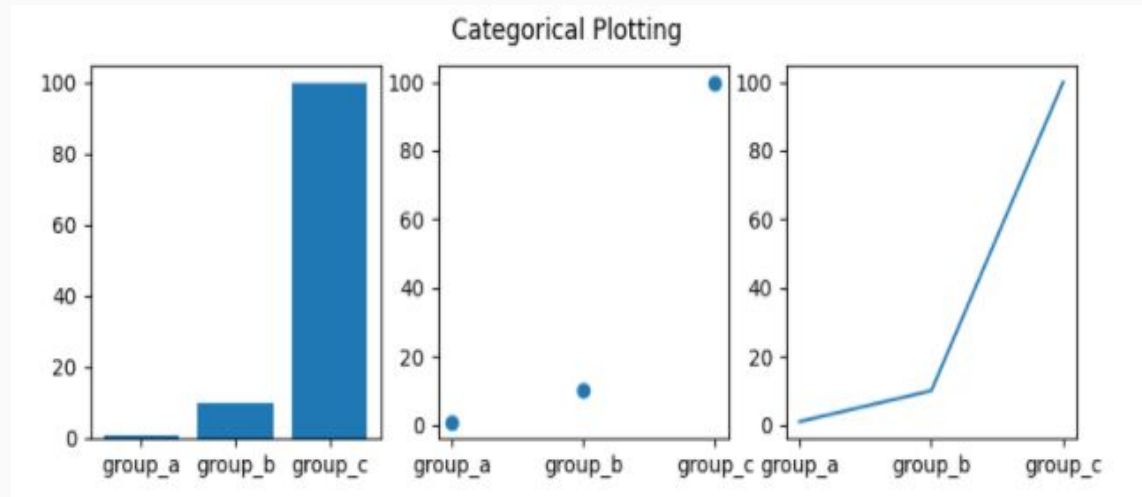
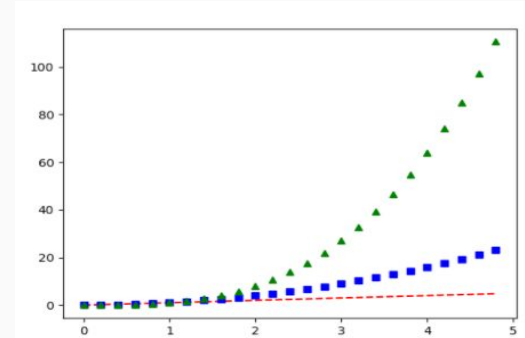
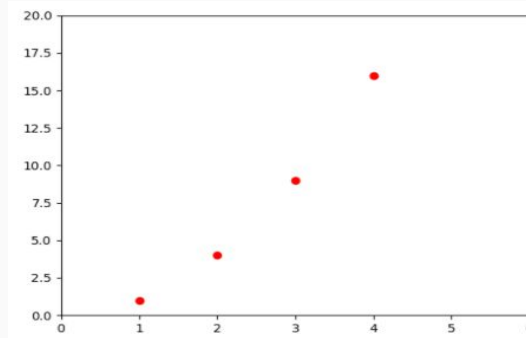
```
>>> print(digits.data)
[[ 0.  0.  5. ...  0.  0.  0.]
 [ 0.  0.  0. ... 10.  0.  0.]
 [ 0.  0.  0. ... 16.  9.  0.]
 ...
 [ 0.  0.  1. ...  6.  0.  0.]
 [ 0.  0.  2. ... 12.  0.  0.]
 [ 0.  0. 10. ... 12.  1.  0.]]
```

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


Matplotlib

Visualization with Python

matplotlib



Matplotlib

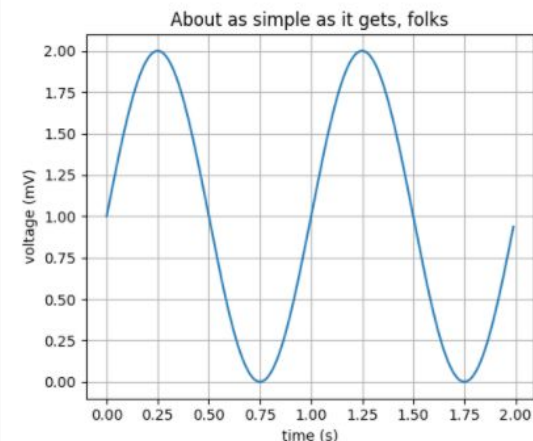
Visualization with Python

matplotlib

```
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

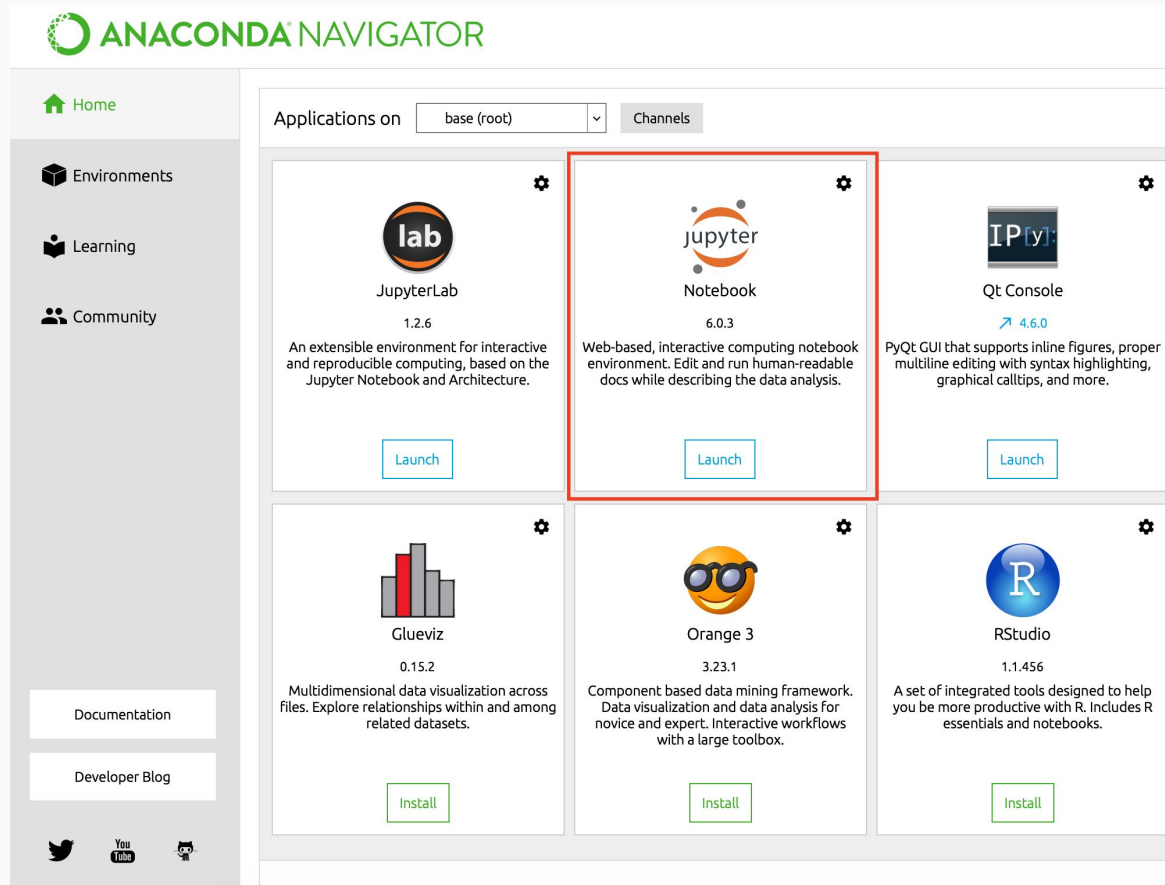
Computer Needed:

- GPU Supported, Cuda, cuDNN

Container Engine

- Docker

Anaconda Navigator



The image shows the Anaconda Navigator application interface. On the left is a sidebar with navigation options: Home (selected), Environments, Learning, and Community. Below these are links for Documentation and Developer Blog, and social media icons for Twitter, YouTube, and GitHub. The main area displays a grid of applications available on the 'base (root)' channel. The applications are: JupyterLab (1.2.6), Jupyter Notebook (6.0.3), Qt Console (4.6.0), Glueviz (0.15.2), Orange 3 (3.23.1), and RStudio (1.1.456). Each application card includes its logo, name, version, a brief description, and a button to either 'Launch' or 'Install' it. The Jupyter Notebook card is highlighted with a red border.

ANACONDA NAVIGATOR

Applications on base (root) Channels

- JupyterLab** 1.2.6
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
[Launch](#)
- Jupyter Notebook** 6.0.3
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
[Launch](#)
- Qt Console** 4.6.0
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
[Launch](#)
- Glueviz** 0.15.2
Multidimensional data visualization across files. Explore relationships within and among related datasets.
[Install](#)
- Orange 3** 3.23.1
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
[Install](#)
- RStudio** 1.1.456
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
[Install](#)

Archivo Editar Buscar Código fuente Ejecutar Depurar Terminales Proyectos Herramientas Ver Ayuda



Notebook Ayuda

untitled0 + Origen Terminal Objeto

File Edit View Insert Cell Kernel Help Trusted Kernel

Code

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]:

Historial de comandos Terminal de IPython

Editor Notebook

Permisos: RW

Fin de línea: CRLF

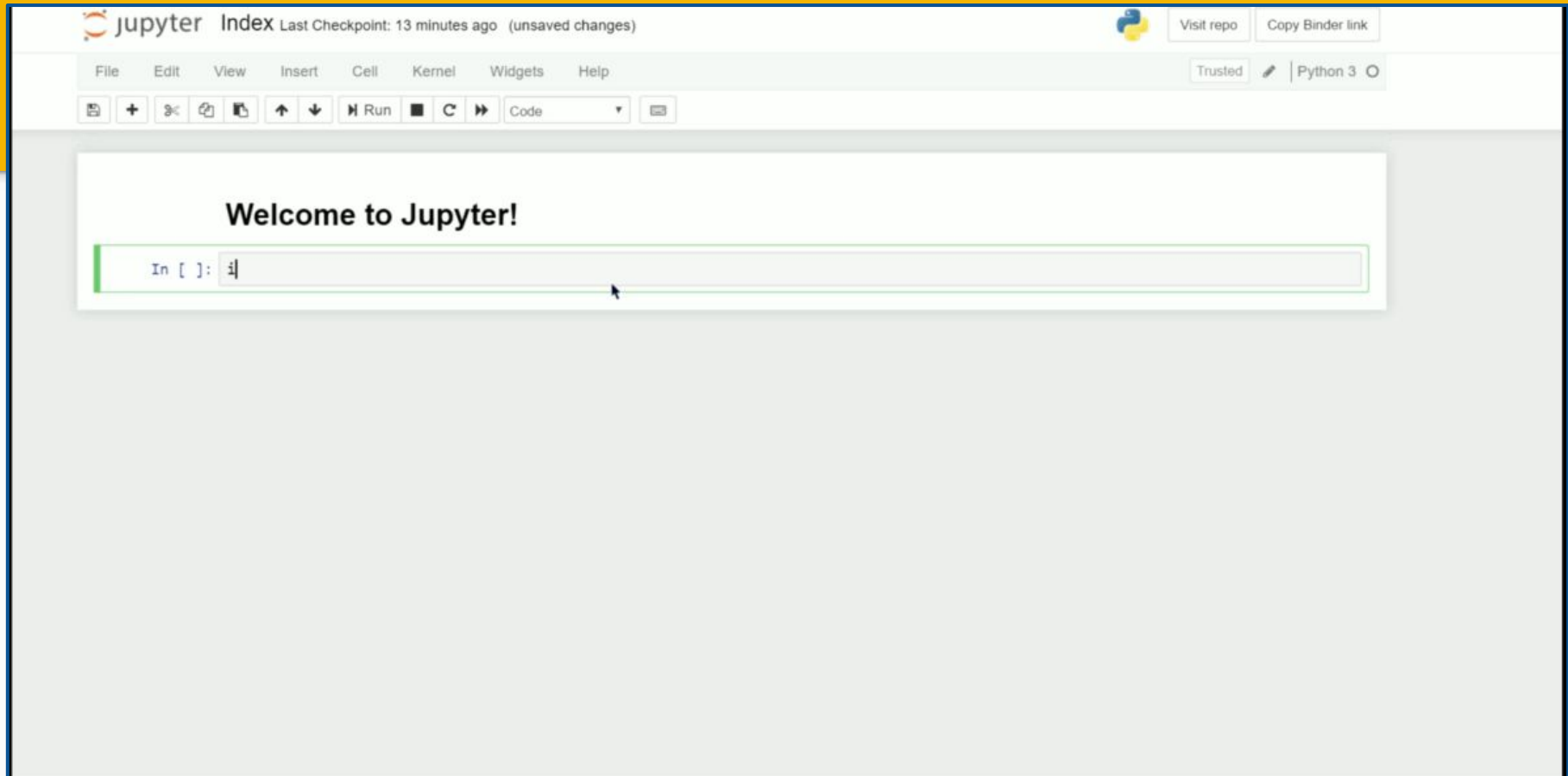
Codificación: UTF-8

Línea: 9

Columna: 1

Memoria: 83 %

Jupyter Notebook



The image shows a Jupyter Notebook interface. At the top, the header bar displays the Jupyter logo, the word "Index", and the text "Last Checkpoint: 13 minutes ago (unsaved changes)". On the right side of the header, there are two buttons: "Visit repo" and "Copy Binder link". Below the header is a menu bar with the following items: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. To the right of the menu bar, there are two buttons: "Trusted" and "Python 3". Below the menu bar is a toolbar with various icons for file operations (save, copy, paste, etc.), cell navigation (up, down), and execution (run, stop, restart). The main area of the notebook is a large white box with the text "Welcome to Jupyter!". Below this text is a code input area with the prompt "In []:" followed by a cursor. The background of the notebook is a light gray.

jupyter Index Last Checkpoint: 13 minutes ago (unsaved changes)

Visit repo Copy Binder link

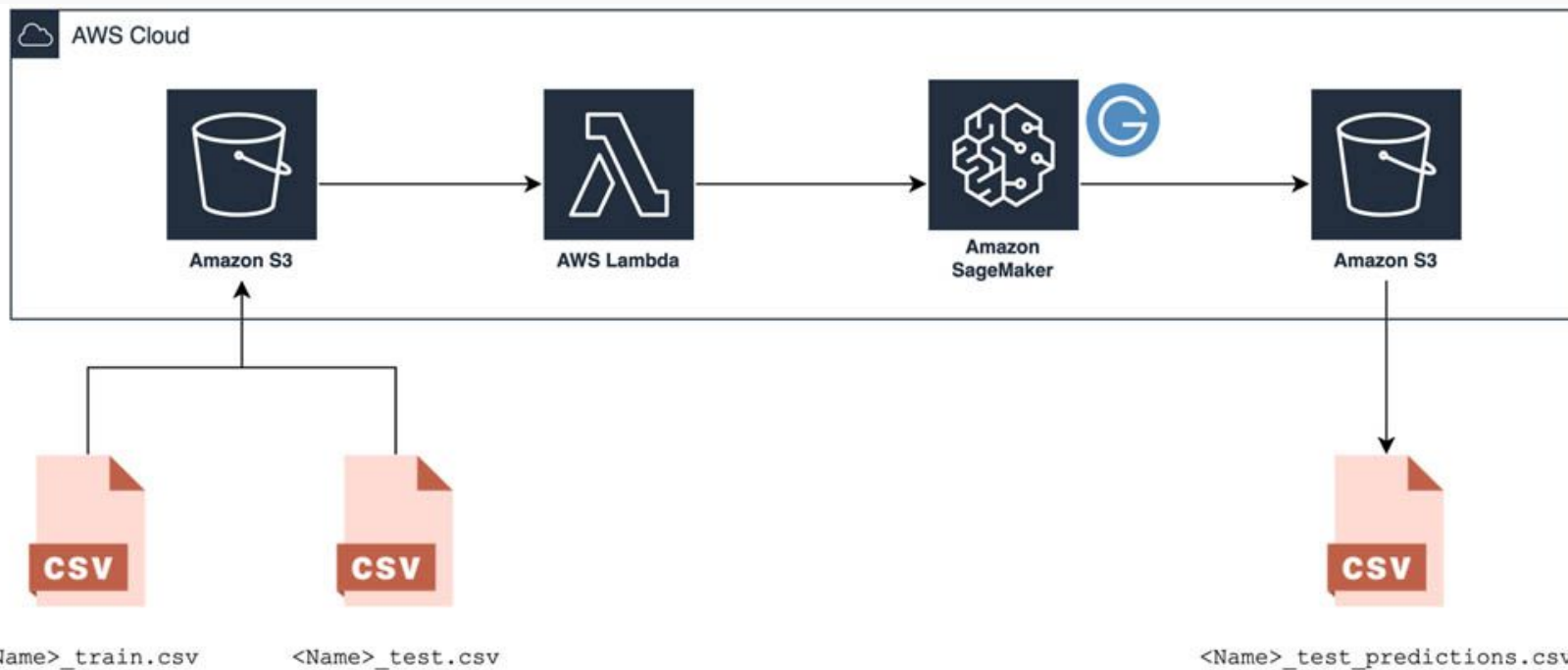
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run

Welcome to Jupyter!

In []: |

AWS



Google Colab



The screenshot displays the Google Colab web interface. At the top, the browser address bar shows the URL `colab.research.google.com/drive/1OJH100ABN1Qag_p6KZd_8CO0qg_e05Hnc0f0-Ub1R3-Qc08L0`. The Colab logo and the file name `ColabUI.ipynb` are visible in the top left. The interface includes a menu bar with `File`, `Edit`, `View`, `Insert`, `Runtime`, `Tools`, and `Help`. On the right side of the top bar, there are buttons for `Comment`, `Share`, and a user profile icon. Below the menu bar, there are tabs for `+ Code` and `+ Text`. The left sidebar contains a `Table of contents` panel with tabs for `Table of contents`, `Code snippets`, and `Files`. The `Files` tab is active, showing a file explorer with a tree view containing `gdrive`, `sample_data`, and `colab_ui.py`. The main area is divided into two sections: `Metrics` and `Train`. The `Metrics` section has a sub-header `Choose training parameters`. The `Train` section has a sub-header `Check learning rate and train your model using your chosen parameters. The trained model is saved as a pth file under the 'models' folder under your chosen 'path' folder`. Below this, there is a code editor with a single line of code: `[1] from colab_ui import*`. A dropdown menu is open, showing the file `colab_ui.py` selected. Below the dropdown, there is a `Drive` tab with sub-tabs `Info`, `Data`, `Augmentation`, `Batch`, `Model`, `Metrics`, and `Train`. The `Drive` tab is active, showing a `mounting drive` section with the text `Mounted at /content/gdrive` and `drive mounted`. Below this, there is a `Load Path` button and a text input field containing `/content/gdrive/My Drive`. At the bottom left, there is a `Disk` section showing `23.83 GB available`.





Microsoft Azure




Preview

Microsoft Azure


Machine learning







Home


Author


 Automated ML


 Designer


 Notebooks


Assets

 Datasets


 Experiments

 Pipelines

 Models

 Endpoints


Manage

 Compute

adb_automation_eastus2_ws > Welcome


Automated machine learning

Let automated machine learning train and find the best model based on your data without writing a single line of code. [Learn more](#)


 New automated ML run

No recent automated ML runs to display.

Click "New automated ML run" to create your first run

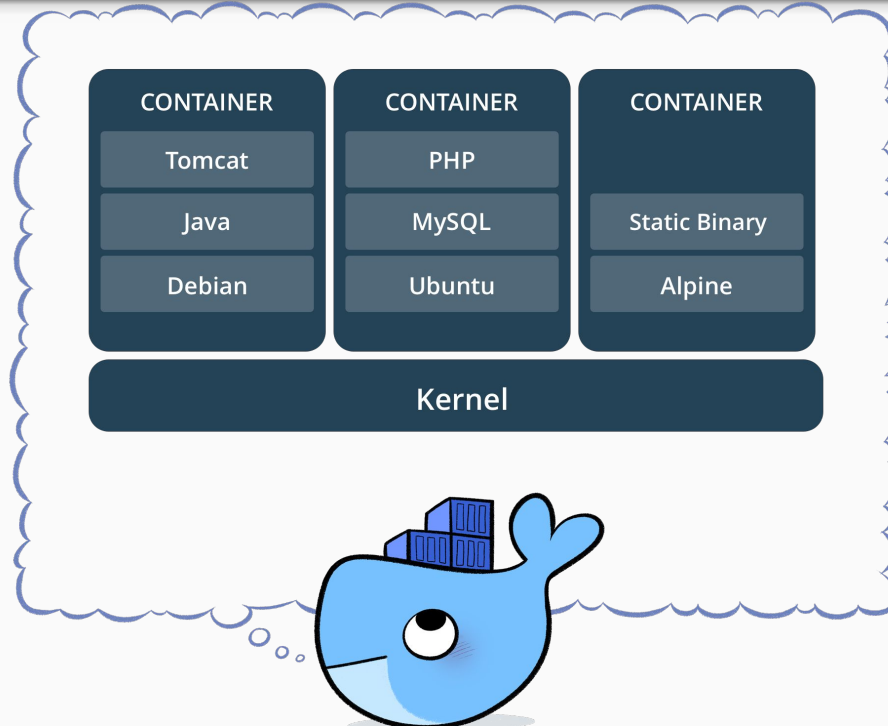
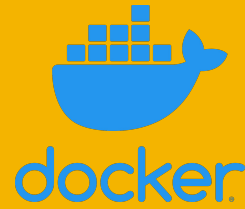
 [Learn more](#)

Documentation

 [Concept: What is automated machine learning?](#)

[View all documentation](#)

Docker

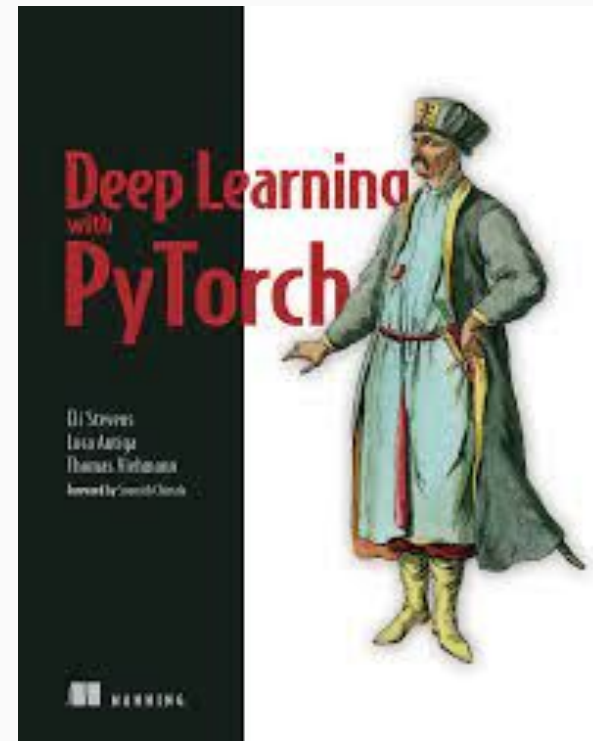
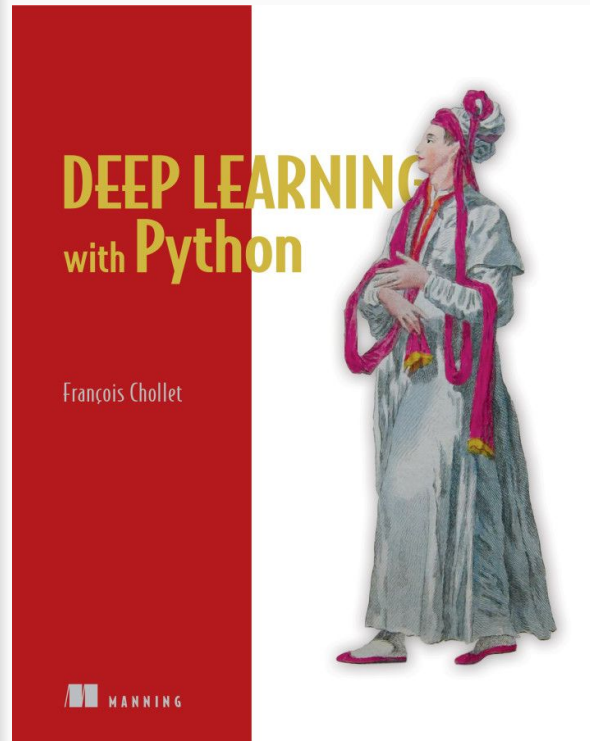


04

Suggested Resources

Suggested Resources -Book-

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



Browse > Data Science > Machine Learning

Offered By

Machine Learning

Stanford

★★★★★ 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



stack overflow



GitHub

THANKS



Does anyone have any questions?

iremkomurcubm@gmail.com

iremkomurcu.com



[iremkomurcu](https://twitter.com/iremkomurcu)



[irem-komurcu](https://github.com/irem-komurcu)



[iremkomurcubm](https://www.instagram.com/iremkomurcubm)



[iremkomurcu](https://www.youtube.com/iremkomurcu)