Introduction TensorFlow 2

Introduction to TensorFlow 2

What is TensorFlow ?

TensorFlow is an end-to-end open-source platform developed by Google for machine learning

--> https://www.tensorflow.org/

TensorFlow 2.0

Introduction to TensorFlow 2

What is Keras ?

Keras is a high-level deep learning API (Application Programming Interface) running on top of TensorFlow.

It is an extension to TensorFlow that makes machine learning more user-friendly.

--> https://keras.io/

Introduction to TensorFlow 2

What is Google Colab ?

Google is a web IDE (Integrated Development Environment) for Python :

- Enables Machine Learning operations and storage on the cloud.
- Is a Jupyter notebook environment that requires no setup
- Enables access to Google drive
- Allows using code, text, and images
- Enables connection to GPU

--> https://colab.research.google.com/notebooks/mlcc/intro_to_neural_nets.ipynb

Introduction to TensorFlow 2

Overview of Colaboratory Features

Fichier Modifier Affichage Insérer Exécution Outils Aide

Partager

Sommaire

- Cells
  - Code cells
  - Text cells
  - Adding and moving cells
- Working with python
  - System aliases
  - Magics
  - Automatic completions and exploring code
  - Exception Formatting
  - Rich, interactive outputs
- Integration with Drive
  - Commenting on a cell
- Section

+ Code + Texte Copier sur Drive Connecter Modification

Cells

A notebook is a list of cells. Cells contain either explanatory text or executable code and its output. Click a cell to select it.

Code cells

Below is a **code cell**. Once the toolbar button indicates CONNECTED, click in the cell to select it and execute the contents in the following ways:

- Click the **Play icon** in the left gutter of the cell;
- Type **Cmd/Ctrl+Enter** to run the cell in place;
- Type **Shift+Enter** to run the cell and move focus to the next cell (adding one if none exists); or
- Type **Alt+Enter** to run the cell and insert a new code cell immediately below it.

There are additional options for running some or all cells in the **Runtime** menu.

```
[ ] a = 10
    a
```

```
10
```

Text cells

This is a **text cell**. You can **double-click** to edit this cell. Text cells use markdown syntax. To learn more, see our markdown guide.

You can also add math to text cells using LaTeX to be rendered by MathJax. Just place the statement within a pair of **$** signs. For example `$\sqrt{3x-1}+(1+x)^2$` becomes $\sqrt{3x-1}+(1+x)^2$.

Introduction to TensorFlow 2

Other libraries :

NumPy : Scientific computing package

--> https://numpy.org/

Matplotlib : Static, animated, and interactive visualisations in Python

--> https://matplotlib.org/stable/index.html

Scikit-Learn : Machine Learning and data analysis built on NumPy, SciPy, and Matplotlib

--> https://scikit-learn.org/stable/

Introduction to TensorFlow 2

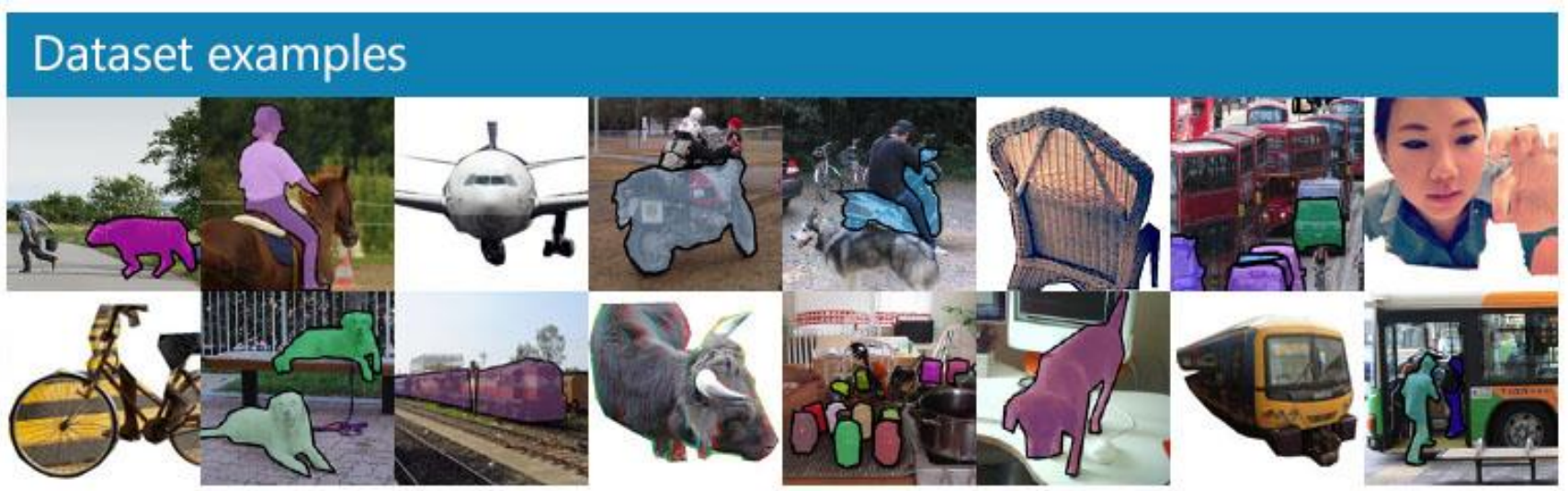
Datasets - A few examples :

- MNIST : Handwritten digits

Introduction to TensorFlow 2

Datasets - A few examples :

- COCO : Common Objects in COntext



Introduction to TensorFlow 2

Datasets - A few examples :

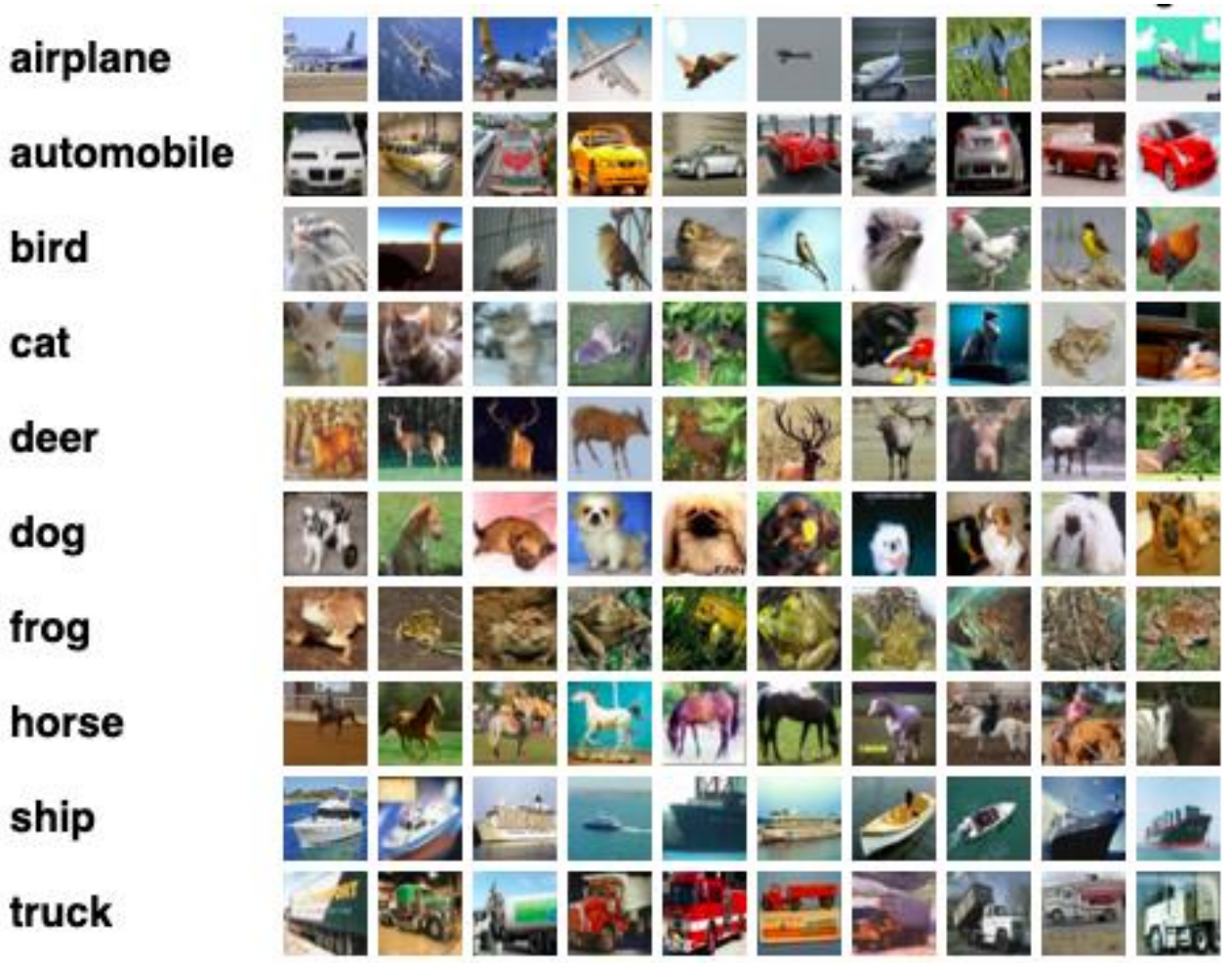
- ImageNet : Image dataset of different classes

Introduction to TensorFlow 2

Datasets - A few examples :

- CIFAR-10 and CIFAR-100 :

  Image datasets of different classes

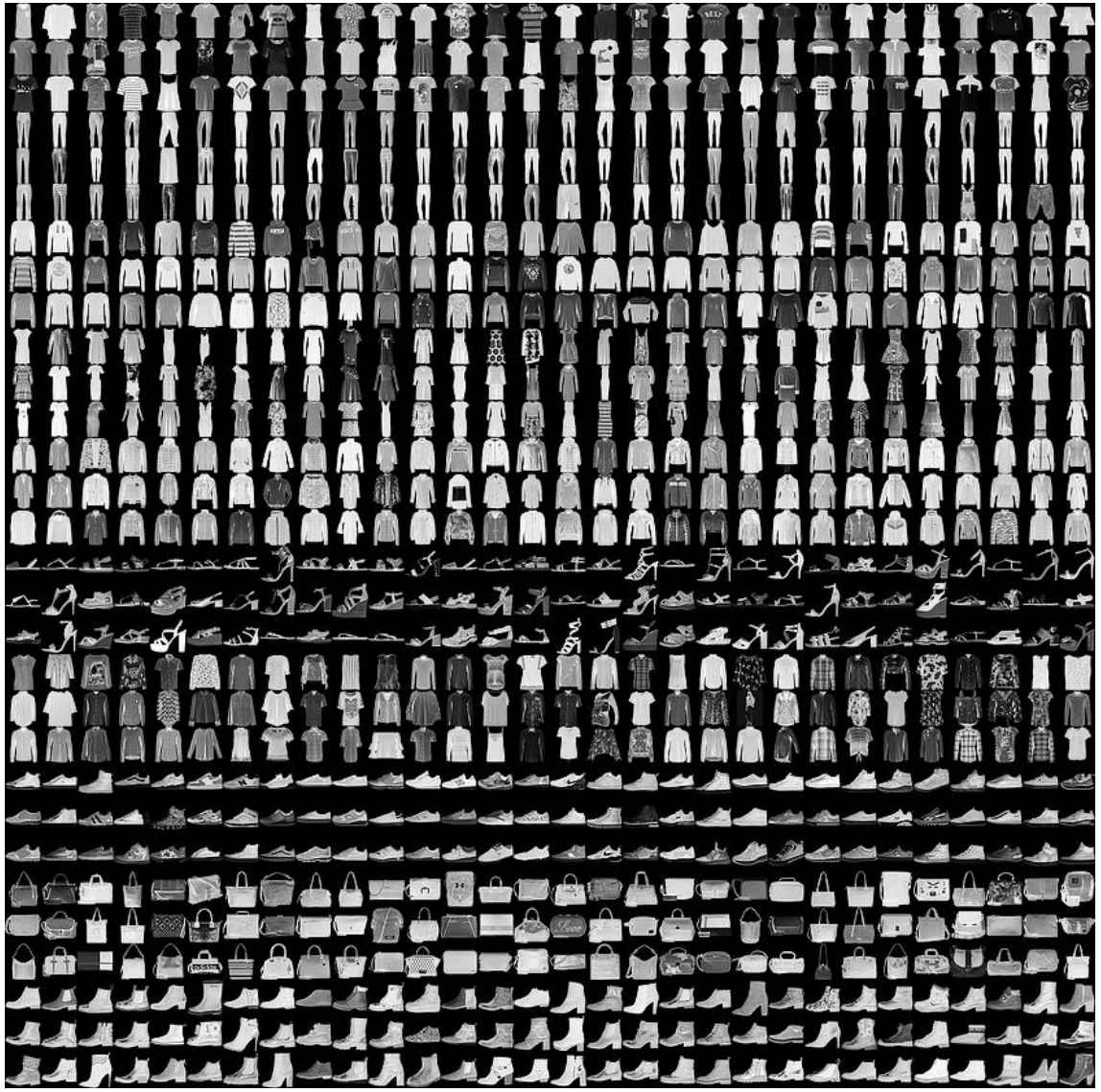


Introduction to TensorFlow 2

Datasets - A few examples :

- FASHION-MNIST :

  Image datasets of Zalando's article images



Introduction to TensorFlow 2

Datasets - A few examples :

- Wine Dataset : Wine quality classification based on features like color, alcohol level, etc.

alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue
14.23	1.71	2.43	15.6	127.0	2.80	3.06	0.28	2.29	5.64	1.04
13.20	1.78	2.14	11.2	100.0	2.65	2.76	0.26	1.28	4.38	1.05
13.16	2.36	2.67	18.6	101.0	2.80	3.24	0.30	2.81	5.68	1.03
14.37	1.95	2.50	16.8	113.0	3.85	3.49	0.24	2.18	7.80	0.86
13.24	2.59	2.87	21.0	118.0	2.80	2.69	0.39	1.82	4.32	1.04
...	...	...	...	...	...	...	...	...	...	...
13.71	5.65	2.45	20.5	95.0	1.68	0.61	0.52	1.06	7.70	0.64
13.40	3.91	2.48	23.0	102.0	1.80	0.75	0.43	1.41	7.30	0.70
13.27	4.28	2.26	20.0	120.0	1.59	0.69	0.43	1.35	10.20	0.59
13.17	2.59	2.37	20.0	120.0	1.65	0.68	0.53	1.46	9.30	0.60
14.13	4.10	2.74	24.5	96.0	2.05	0.76	0.56	1.35	9.20	0.61

Introduction to TensorFlow 2

More Datasets :

Datasets can be found on :

Kaggle

Google Dataset Search

Keras

Check this article for more --> Best Public Datasets for Machine Learning and Data Science

Introduction to TensorFlow 2

First Experience with Keras :

- The core data structures of Keras are models and layers
- The Sequential model is the simplest model type --> a linear stack of layers

```
from tensorflow.keras.models import Sequential

model = Sequential()
```

Introduction to TensorFlow 2

First Experience with Keras :

- To stack the layers, we use the *add()* method
- The Sequential model is the simplest model type --> a linear stack of layers

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

model = Sequential()

model.add(Dense(units=64, activation='relu'))
model.add(Dense(units=10, activation='softmax'))
```

Note :

'Dense' refers to a fully-connected layer

Introduction to TensorFlow 2

First Experience with Keras :

- Once the model configuration is done, we configure the training/learning process
- This is done using the *compile()* method

```
model.compile(loss='categorical_crossentropy',
              optimizer='sgd',
              metrics=['accuracy'])
```

Introduction to TensorFlow 2

First Experience with Keras :

- The *compile()* method allows us to configure the learning further
- Here, we configure the optimizer by specifying the *learning rate* and the *momentum*

```
import keras
model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.SGD(learning_rate=0.01, momentum=0.9))
```

Introduction to TensorFlow 2

First Experience with Keras :

- Train the model using the *fit()* method :

```
model.fit(x_train, y_train, epochs=5, batch_size=32)
```

Notes :

x_train = training data

y_train = training labels

Introduction to TensorFlow 2

First Experience with Keras :

- Evaluate model performance using the *evaluate()* method :

```
loss, accuracy = model.evaluate(x_test, y_test)
```

Notes :

x_test = test/evaluation data

y_test = test/evaluation labels

Introduction to TensorFlow 2

First Experience with Keras :

- Test on new data using the *predict()* method :

```
classes = model.predict(x_test)
```

Introduction to TensorFlow 2

Data Loading :

1. Dataset importation :

```
import pandas as pd
from sklearn.datasets import load_wine
wine_data = load_wine()
```

```
data_frame = pd.DataFrame(wine_data.data[:, :], columns=wine_data.feature_names[:])
data_frame['label'] = wine_data.target
data_frame
```

Introduction to TensorFlow 2

Data Loading :

2. Dataset loading from .csv file :

```
import pandas as pd
```

```
dataset = pd.read_csv('Churn_Modelling.csv')
#Kaggle database (source: https://www.kaggle.com/aakash50897/churn-modellingcsv?select=Churn_Modelling.csv)
dataset
```