



Professional training

Neural network model

Summary: In this Module, you will learn about neural network model (CNN).

Version: 1.00

Contents

| | | |
|-------------|---------------------------------------|-----------|
| I | Introduction | 2 |
| II | General instructions | 3 |
| III | Exercise 00 | 4 |
| IV | Exercise 01 | 5 |
| V | Exercise 02 | 6 |
| VI | Exercise 03 | 7 |
| VII | Bonus part | 9 |
| VIII | Submission and peer-evaluation | 10 |

Chapter I

Introduction

Greetings!



If you haven't already done so, read `en.toolkit.pdf`.

What this Module will cover:

In this module, we're about to embark on an exciting journey into the world of image classification, a powerful field of machine learning and computer vision. Using state-of-the-art deep learning techniques, we'll explore how to build Convolutional Neural Networks (CNNs) to label and classify images.

As we progress, we'll delve deeper into the world of CNNs, a class of neural networks specifically designed for image-related tasks. You'll learn how to design and train CNN architectures using industry-standard libraries like PyTorch or Keras. We'll explore techniques for feature extraction, model training, and fine-tuning to achieve impressive classification accuracy.

By the end of this module, you'll not only have a strong grasp of image classification principles but also the practical skills to apply CNNs to your own projects, whether it's identifying objects in images, recognizing handwritten digits.

Wishing you success in your learning journey.

Chapter II


General instructions

Unless explicitly specified, the following rules will apply every day of this Professional training.

- This subject is the one and only trustable source. Don't trust any rumor.
- This subject can be updated up to one hour before the turn-in deadline.
- The assignments in a subject must be done in the given order. Later assignments won't be rated unless all the previous ones are perfectly executed.
- Be careful about the access rights of your files and folders.
- Your assignments will be evaluated by your peers.
- You must not leave in your turn-in your workspace any file other than the ones explicitly requested By the assignments.
- You have a question? Ask your left neighbor. Otherwise, try your luck with your right neighbor.
- Every technical answer you might need is available in the **man** or on the Internet.
- By Thor, by Odin! Use your brain!!!

Chapter III


Exercise 00

| | |
|---|-------------|
|  | Exercise 00 |
| Data exploration | |
| Turn-in directory : <i>ex00/</i> | |
| Files to turn in : Advanced01.ipynb | |
| Allowed functions : pandas | |

For this first exercise, you'll need to load the data into colab and do some data exploration, i.e. you'll need to understand your data.

Chapter IV

Exercise 01

| | |
|---|-------------|
|  | Exercise 01 |
| Data Visualisation | |
| Turn-in directory : <i>ex01/</i> | |
| Files to turn in : Advanced01.ipynb | |
| Allowed functions : pandas, seaborn | |

In this exercise, to better understand your data, you need to display a bar chart to see if your data is balanced, and then display some images of your data set.

You should have something like this:

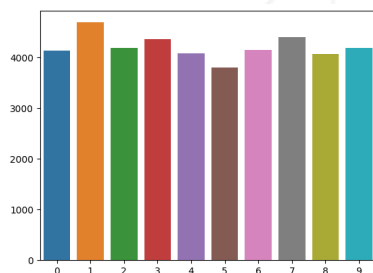


Figure IV.1: Distribution of Labels

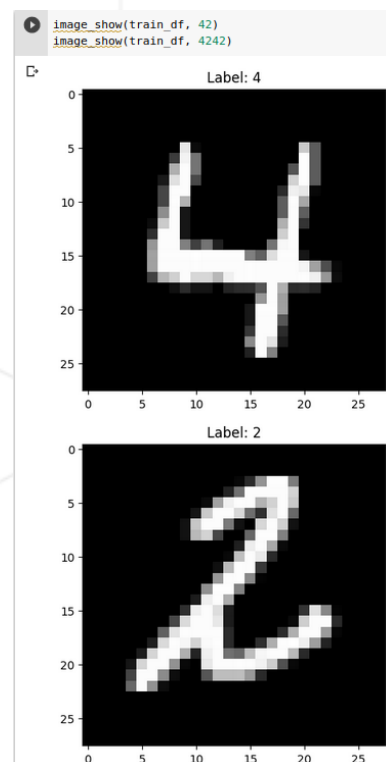



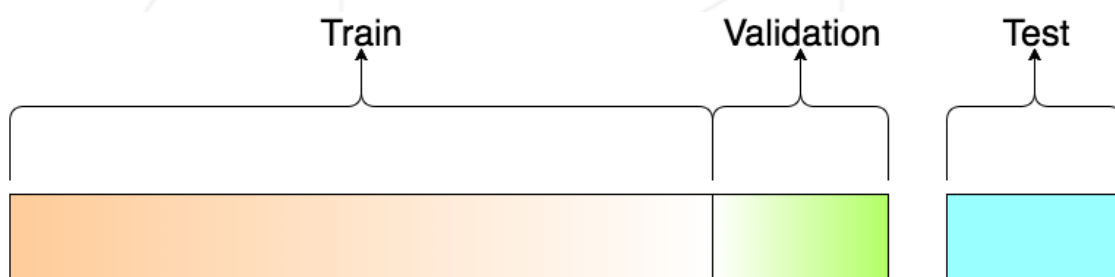
Figure IV.2: Numerical Attributes

Chapter V

Exercise 02


| | |
|---|-------------|
|  | Exercise 02 |
| Data Split | |
| Turn-in directory : <i>ex02/</i> | |
| Files to turn in : Advanced01.ipynb | |
| Allowed functions : pandas, seaborn | |

You have to separate your `train.csv` file into a training data set and a validation data set, and later we'll see your actual results on a test set.

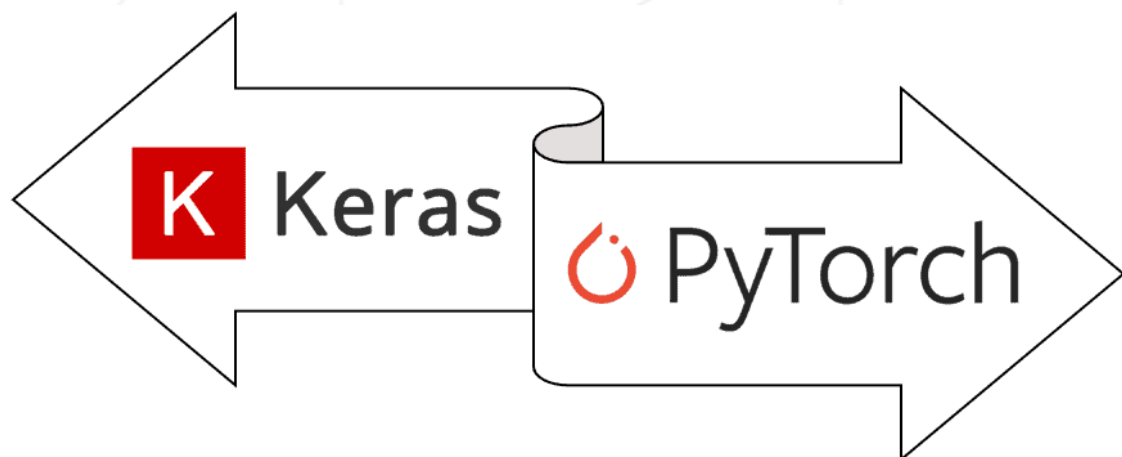


Chapter VI

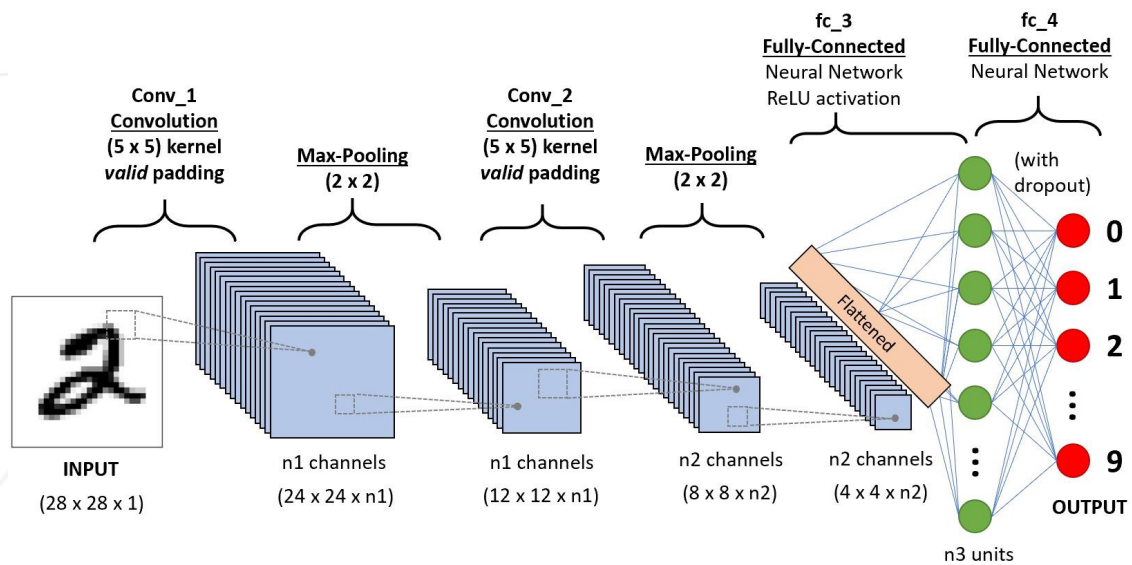
Exercise 03

| | |
|---|-------------|
|  | Exercise 03 |
| Models | |
| Turn-in directory : <i>ex03/</i> | |
| Files to turn in : Advanced01.ipynb | |
| Allowed functions : A11 | |

In this exercise, you will create a neural network using [Keras](#) or [PyTorch](#) (you can complete the exercise with both libraries if you want/have time; Keras is simpler for beginners).



Just like in the diagram below, create your own neural network with the requested layers.



You have to recreate each type of layer you require, making sure you understand the purpose of each layer and the activation function



You have to an accuracy greater than 0.95 on the Test set.

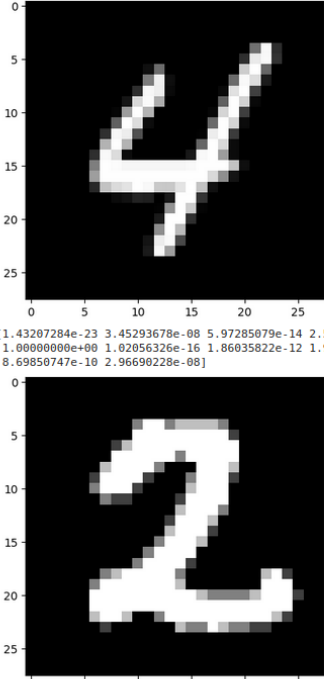
Chapter VII

Bonus part

If you have some time, you can try to validate this little bonus. It is not mandatory, so don't waste too much time doing it.

Display some images with your predictions

```
[40] image_show(test_df, 42, visualize=True, test_df=True)
      print(predictions[42])
      image_show(test_df, 4242, visualize=True, test_df=True)
      print(predictions[4242])
```



```
[1.43207284e-23 3.45293678e-08 5.97285079e-14 2.50143602e-25
 1.00000000e+00 1.02056326e-16 1.86035822e-12 1.94881139e-14
 8.69850747e-10 2.96690228e-08]

[9.1252943e-11 3.8889572e-03 1.0000000e+00 4.0728279e-10 1.8171478e-13
 2.0083014e-21 1.7261727e-14 1.4040934e-07 2.8609300e-05 4.8089895e-17]
```



The bonus part will only be assessed if the mandatory part is PERFECT. Perfect means the mandatory part has been integrally done and works without malfunctioning. If you have not passed ALL the mandatory requirements, your bonus part will not be evaluated at all.

Chapter VIII

Submission and peer-evaluation

- Create a `professional_training_Advanced` folder at the root of your home, and move around in it.
- Create a new `module01` folder and navigate to it.



Please note, during your defense anything that is not present in the folder for the day will not be checked.