Practical work 10 – Functional API and Transfer Learning

Objectives

The objective of this PW is to practice with some applications of Convolutional Neural Networks (CNN) using transfer learning. Another objective is to perform experiments with the **functional API** of Keras that allows to build more complex network structures.

Submission

— **Deadline** : W11 (in 1 week), before the start of the lecture.

— **Format** : Zip with report and iPython notebook.

Exercise 1 Functional API of Keras

Make sure first that you understand the principles of Keras functional API (see lecture slides). Using the CIFAR10 dataset :

a) Re-use one of your best CNN architecture from the previous PWs and transform the model definition from sequential to the functional API. Observe that you get the same number of parameters and similar performances between the sequential and functional models.

b) Experiment with non-sequential strategies such as *multiple features* or *multiple paths* as described in the lecture slides.

c) Optional : install GraphViz to use the Keras function `plot_model()`. If working[1], use it to generate graphs of the architectures.

d) Use *callbacks* to save the best trained models according to a monitoring of the accuracy on the test set.

Report on your experiments and describe your best configuration through experimenting with 3-4 different architectures. Use a table similar to the example given below and provide the hyper-parameters used for your configurations.

1. The installation may not be so obvious.

Model	Architecture	Callback	Acc. train %	Acc. test %
1	Description of architecture...	yes	84.2%	78.4%
2	...	...	...	...
3	...	...	...	...
4	...	...	...	...
5	...	...	...	...

TABLE 1 – Performances on CIFAR10 with different sequential and non-sequential configurations. In the Architecture column, D is the number of filters, w h are the width and height in pixels, S is the stride value, P is the padding value, etc.

Exercise 2 Transfer learning

Keras provides many pre-trained architectures on ImageNet. Such architectures are able to extract robust image features and can generally be used for many different image tasks. The objective of the exercise is to check if such features lead to good results on the Food101 dataset. This dataset contains 101'000 color images of meals corresponding to 101 classes. In order not to have too long trainings, only the first 20 classes will be used. The following tutorial can probably help you with the exercise : https://keras.io/guides/transfer_learning/#build-a-model

a) Review the lecture slides on using Keras for a transfer learning task.

b) Chose one of the architecture presented here : https://keras.io/applications/. Beware that some architectures are using large memory and lots of cpu (so better move to gpu) !

c) Download the provided notebook `ex2_transfer_learning_stud.ipynb`.

d) Implement the preprocessing and the training phases.

e) Experiment with one or several architectures with different classification heads (hidden layers, ...).

f) Describe the results in your report which architecture and parameters works best.

Exercise 3 Optional : Review Questions

a) Why (or when) do we need the functional API of Keras ?

b) What are the benefits of using transfer learning ?