Practical Work 14 – 02/06/2022
Keras advanced - Transfer learning - MLOps

Objectives

The first objective of this PW is to practice with non-sequential deep architectures. The second objective is to experiment with transfer learning. Finally, MLOps considerations and tools will be optionally used. To provide your solutions, you can use either the *functional API* of Keras (as seen in class) or build an equivalent using Pytorch.

Submission

— **Deadline** : Wednesday 13th of June, 12 :00 (noon)

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

Exercise 1 Non-sequential architectures

Using the CIFAR10 dataset :

a) Re-use one of your best CNN architecture from PW06 or PW07 and transform the model definition to use non-sequential strategies such as *multiple features* or *multiple paths* as described in the slides.

b) Optional : generate graphs of the architectures as shown in class. For that, install GraphViz to use the Keras function `plot_model()` (note – you may need to launch the installation of pydot first with `pip install pydot`).

c) 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.

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

With Keras and Pytorch, you have access to 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 here.

For the one using Keras, the following tutorial can probably help you with the exercise : https://keras.io/guides/transfer_learning/#build-a-model

a) Review the slides on using transfer learning.
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 if you chose large models) !
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 MLOps pipelines

Being able to reproduce the experiments you create is crucial in research and industrial settings. Automation is a key aspect of achieving this goal. While Jupyter Notebooks are useful to perform experiments and quick visualizations, they are not well suited when creating "serious" models. In this exercise, your goal is to transform the Jupyter Notebook from exercise 2 into a small, locally run, DVC pipeline. You can find a detailed documentation of DVC at https://dvc.org/doc/user-guideaswellastheslidesofthecourse.

— Split your Jupyter Notebook into multiple command line tools. The final step should produce a markdown report as its output (keep it simple)

- Run the tools manually from the command line to reproduce the results of your Jupyter Notebook
- Create a DVC pipeline that chains your individual steps
- Optional : Output metrics that can be used by DVC and run multiple experiments to compare their results. Check the documentation at `https://dvc.org/doc/start/data-management/metrics-parameters-plots`

Submission : a pdf with screenshots of your results or a zip file including the markdown report, command line tools etc.

Exercise 4 Optional : Review Questions

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

b) What are the benefits of using transfer learning ?

c) What does *freezing* mean in a transfer learning strategy ? What is the influence of the training data set size on the *freezing* ?

d) What are the principles of self-supervised learning ?

e) In which situations would you prefer to use self-supervised learning instead of using pre-trained networks to extract features ?

f) What are the main advantages of using MLOps tools such as DVC ?