
NN and CNNs for Image Classification/Recognition tasks

A) OBJECTIVES

1. Understand the implementation of a simple Deep NN implemented from scratch.
2. Learn how to implement and train a toy CNN with Keras for Image Classification
3. Fine-tune a well-known Image Classification CNN architecture with Keras

Optional task goals: variations of the DNNs and CNNs, segmentation, Pytorch

B) INTRODUCTION: In this practical session, we'll train and evaluate several deep neural networks for image classification. You'll use both the CIFAR-10 you've used in previous lab1 and your own tiny-dataset.

PREVIOUS TO THE SESSION: Select and download sample images for your own toy-5-class image classification problem. Pick 5 classes from some of these options (you can download just a few images, ~25-50, per class if you have constraints):

- https://www.kaggle.com/kmader/food41/version/5#
- http://www.robots.ox.ac.uk/~vgg/data/pets/
- http://www.robots.ox.ac.uk/~vgg/data/flowers/
- **yours?**

C) EVALUATION: Submit your notebooks to Moodle **by the corresponding deadline (2 weeks after your lab session).** To do this, just download the colab document as .ipynb and upload that to moodle. All your answers should be stored in the .ipynb. So please!! MAKE SURE the **notebook saves the outputs of your cells**. (If you didn't disable it, by default it's done this way. You can download the notebook and make sure of it before closing your session).

The following will be evaluated:
- Correctness and extent of the work. (NO NEED for HIGH accuracies! These are toy-examples. We just want correctness on what you have done!)
- Correctness in the use of technical terms in the answers.
- Analysis and discussion of the results.
- Organization and cleanliness of the code.

REMEMBER:
- **To opt for the maximum grade, REQUIRED to submit all mandatory tasks + 2 other points of your choice** among the given list, if you work in a group of two people.

- You are **allowed 5 *late-days* in total for all the labs**. You can use them when you want, e.g. you can submit 1 day late each lab, or up to four days late one lab.

D) DESCRIPTION OF MANDATORY TASKS for LAB4:

For each task, **answer ALL the questions within the notebook, add the code to fill the gaps and run the requested subtasks**:

1. Simple 2 layer NN implemented from scratch. Use notebook **Simple2NN.ipynb**.
- 1.1. Understand the example. Run it to train the model with CIFAR-10 and answer the questions.

2. Simple CNN implemented in Keras. Use notebook **SimpleCNN.ipynb**.
- 2.1. Understand the example and run it to train your own toy-dataset.
- 2.2. What happens if you remove the dropout layers? Add the code necessary to include data augmentation**.**
  Re-train the CNN on your toy-dataset to analyze what these changes do and briefly discuss the results of the CNN with/without augmentation, with/without dropout.

3. Fine-tune existing CNN with Keras. Use notebook **CNN_Finetunning.ipynb**.
- 3.1. Fine-tune an existing well-known CNN architecture for Image classification on your toy-dataset.
- 3.2. Compare and discuss the results you obtain for your toy-dataset on this task and in previous one/s.

E) DESCRIPTION OF OTHER TASKS (choose 2 points to do out of these):

- **Extension of task 1 (**Make a copy of the notebook you did in task 1 and edit there)

1.2. (1 POINT) Train the simple NN on your own 5-class toy-dataset. **Code and display the confusion matrix for the validation data**.
Discuss briefly the results on your toy-dataset (comment on the loss and metrics plots, on why it's good/bad, …) and what can you analyze from the confusion matrix (include your answer at the end of the notebook)

- **Extension of task 2.** (Make a copy of the notebook you did in task 2 and edit there)

2.3. (1 POINT) Train the CNN with CIFAR-10 data (load it as in task 1). Train two additional variations of the model (e.g. varying the optimizer, the loss, the regularization, …). How do these results compare with your CIFAR-10 results in task 1? Discuss and compare all the models you have tried for CIFAR-10 in task1 and 2. (include your answer at the end of the notebook)

- **Semantic segmentation.** Use the OPT_TorchVision_SemanitSeg.ipynb notebook

4.1. (1 POINT) Read, modify and analyze the results of different semantic segmentation models.