2. Programming Tasks

2.1 Fork DeepDIVA

For your semester project you will need to work on DeepDIVA. Since eventually with the publication your code would become open source, it is good practice to fork the main repo (of DeepDIVA) and put it in your own repository. NOTE: do this only *ONCE* per group! Then report the link to the repository in the pdf of your solutions. If you implement or improve something that you think it might beneficial for the whole community (and not only for your project) do no hesitate do make a pull request.

2.2 Kick off!

Synchronise within your group and decide on the details of the project. In particular decide which task are you going to tackle, the dataset and the approach you are going to use, the timeline, who does what, etc (for some project these might be already defined). Keep in mind that you don't have to teach a NN to solve the task end-to-end but you can have (even multiple) stages of pre/post processing. Finally, formalize your thoughts in a simple 2-3 slides presentation (informal!) that you will use next week to present.

2.1 Transfer Learning from ImageNet

In this task you are going to observe the effect of transfer learning on a (once) very popular achitecture.

- Train AlexNet on CIFAR10 with seed=42 and report the accuracy on the test set.
- Now train again, with the same setting but this time you will add the flag –pretrained. Report the accuracy on the test set.
- Explain (briefly!) what is the difference between the two runs and why there is a difference in performance.

2.2 Transfer Learning from MNIST

In this task you are going to perform transfer learning on your own, without benefitting of a pretrained network from PyTorch zoo.

- (1) Train CNN_basic on SVHN. Report the test test accuracy.
- (2) Train CNN_Basic on MNIST. Report the test test accuracy.
- Load the model from (2) and then train it (fine tune) on SVHN. Report the test set accuracy.

In the third step you are performing transfer learining from MNIST to SVHN.