50.039 – Theory and Practice of Deep learning

Alex

Week 01: Discriminative ML - quick intro

[The following notes are compiled from various sources such as textbooks, lecture materials, Web resources and are shared for academic purposes only, intended for use by students registered for a specific course. In the interest of brevity, every source is not cited. The compiler of these notes gratefully acknowledges all such sources.]

Due: week3 monday, 6pm

Run a small ML project. When I get a new dataset, I have to do these steps 5 times per year at least.
The goal is to let you do a bit recap on how to run a machine learning project in principle.

- Download the 17 flowers images. `http://www.robots.ox.ac.uk/~vgg/data/flowers/17/` or from edimension (should be faster bcs university internal).
- Download a set of features from me for those in the homework link . Are they up ?
- Split the images into 40 per class for training, 20 per class for validation, 20 per class for final test. You are not allowed to use the VGG splits. What is the difference to a random $50 - 25 - 25$ split of the whole data? Why I asked you to split classwise ? Explain in at most 8 sentences.
- idea: use a linear SVM (e.g. scikit-learn) with python3 interface. If you use another kernel, please notify clearly in the report.
- It is a multi-class dataset with mutually exclusive labels. Thus you can train 17 binary svms, one for each class in one-vs-all manner. Each svm is trained on the training dataset using all the training data.
- At test time you predict the class-label as the index of the svm with the highest prediction score.

- This method has one free parameter - the regularization constant. Find the best regularization constant from the set $0.01, 0.1, 0.1^{0.5}, 1, 10^{0.5}, 10, 100^{0.5}$ by repeatedly training on the training set and measuring performance on the validation set. Use as performance measure the class-wise accuracy averaged over all 17 classes.
- Once you have the best value for C, train on train + validation. then report performance of that classifier on the test set
- Submit your code (python3), including the code for splitting, and your saved train val test splits (.npy),
- Submit a short report showing the validation accuracies which lead to the selected C and the final test accuracy. The report should also show 1 or 2 fail case image per class (if there are less errors, then less)

Useful things in python:

```
File handling:
os.path.basename(...)
os.path.join(...)
os.path.isdirectory(...)
os.makedirs()

also helps to extract parts from a filename:
position=pythonstring.find(someotherstring)


loading of numpy arrays:
np.save(...)
np.load(...)

sklearn.svm.*
yourwhateversvm.predict(...) does not help you here,
you need the real valued scores,
not the label from one one-vs-all SVM
```