

Machine Learning Practical: Coursework 2

Release date: Wednesday 2nd November 2016

Due date: 16:00 Thursday 24th November 2016

Introduction

The aim of this coursework is to use a selection of the techniques covered in the course so far to train accurate multi-layer networks for MNIST classification. It is intended to assess your ability to design, implement and run a set of experiments to answer specific research questions about the models and methods covered in the course.

You should choose **three** different topics to research. Our recommendation is to choose one simpler question and two which require more in-depth implementations and/or experiments.

Examples of what might constitute a simpler question include

- How effective are early stopping methods at reducing overfitting?
- Does combining L1 and L2 regularisation offer any advantage over using either individually?
- How does training and validation set performance vary with the number of model layers?
- How does the choice of the non-linear transformation used between affine layers (e.g. logistic sigmoid, hyperbolic tangent, rectified linear) affect training set performance?
- Does applying a whitening preprocessing to the input images help increase the rate at which we can improve training set performance?

Similarly some ideas of more complex topics you could investigate are (there are various questions you could pose on these topics - we leave choosing appropriate ones up to you)

- data augmentation (beyond the random rotations covered in lab 5),
- models with convolutional layers (lectures 7 and 8),
- models with 'skip connections' between layers (such as residual networks / deep residual learning, mentioned at the end of lecture 8)
- batch normalisation (lecture 6).

You are welcome to come up with and investigate your own ideas, these are just meant as a starting point.

Note that it is in your interest to start running the experiments for this coursework as early as possible. Some of the experiments may take significant compute time.

Mechanics

Marks: This assignment will be assessed out of 100 marks and forms 25% of your final grade for the course.

Academic conduct: Assessed work is subject to University regulations on academic conduct:

<http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct>

Late submissions: The School of Informatics policy is that late coursework normally gets a mark of zero. See <http://web.inf.ed.ac.uk/infweb/student-services/ito/admin/coursework-projects/late-coursework-extension-requests> for exceptions to this rule. Any requests for extensions should go to the Informatics Teaching Office (ITO), either directly or via your Personal Tutor.

Report

The main component of your coursework submission, on which you will be assessed, will be a report. This should follow a typical experimental report structure, in particular covering the following for each of the three topics investigated

- a clear statement of the research question being investigated,
- a description of the methods used and algorithms implemented,
- a motivation for each experiment completed (e.g. initial pilot runs, further investigation of observations from previous experiments)
- quantitative results for the experiments you carried out including relevant graphs,
- discussion of the results of your experiments and any conclusions you have drawn.

The report should be submitted in PDF. You are welcome to use what ever document preparation tool you prefer working with to write the report providing it can produce a PDF output and can meet the required presentation standards for the report.

Of the total 100 marks for the coursework, 20 marks have been allocated for the quality of presentation and clarity of the report. A good report, will clear, precise, and concise. It will contain enough information for someone else to reproduce your work (with the exception that you do not have to include the values to which the parameters were randomly initialised).

You will need to include experimental results plotted as graphs in the report. You are advised (but not required) to use `matplotlib` to produce these plots, and you may reuse code plotting (and other) code given in the lab notebooks as a starting point.

Each plot should have all axes labelled and if multiple plots are included on the same set of axes a legend should be included to make clear what each line represents. Within the report all figures should be numbered (and you should use these numbers to refer to the figures in the main text) and have a descriptive caption stating what they show.

Ideally all figures should be included in your report file as [vector graphics](#) rather than [raster files](#) as this will make sure all detail in the plot is visible. Matplotlib supports saving high quality figures in a wide range of common image formats using the `savefig` function. **You should use `savefig` rather than copying the screen-resolution raster images outputted in the notebook.**

Figures saved as a PDF file using `fig.savefig('file-name.pdf')` can be included as graphics in [LaTeX](#) compiled with `pdflatex` and in Apple Pages and [Microsoft Word](#) documents. If you are using Libre/OpenOffice you should use Scalable Vector Format plots instead using `fig.savefig('file-name.svg')`. If the document editor you are using for the report does not support including either PDF or SVG graphics you can instead output high-resolution raster images using `fig.savefig('file-name.png', dpi=200)` however note these files will generally be larger than either SVG or PDF formatted graphics.

If you make use of any any books, articles, web pages or other resources you should appropriately cite these in your report. You do not need to cite material from the course lecture slides or lab notebooks.

Code

You should run all of the experiments for the coursework inside the Conda environment [you set up in the first lab](#).

A branch `mlp2016-7/coursework2` intended to be a starting point for your code for the second coursework is available on the course [Github repository](#) on a branch `mlp2016-7/coursework2`. To create a local working copy of this branch in your local repository you need to do the following.

1. Make sure all modified files on the branch you are currently on have been committed ([see details here](#) if you are unsure how to do this).

2. Fetch changes to the upstream `origin` repository by running
`git fetch origin`
3. Checkout a new local branch from the fetched branch using
`git checkout -b coursework2 origin/mlp2016-7/coursework2`

You will now have a new branch `coursework2` in your local repository.

The only additional code in this branch beyond that already released with the sixth lab notebook is:

- A notebook `Convolutional_layer_tests` which includes a skeleton class definition for a convolutional layer and associated test functions to check the implementations of the layer `fprop`, `bprop` and `grads_wrt_params` methods. This is provided as a starting point for those who decide to experiment with convolutional models - those who choose to investigate other topics may not need to use this notebook.
- A new `ReshapeLayer` class in the `mlp.layers` module. When included in a multiple layer model, this allows the output of the previous layer to be reshaped before being forward propagated to the next layer.

Submission

Your coursework submission should be done electronically using the [submit](#) command available on DICE machines.

Your submission should include

- your completed course report as a PDF file,
- the notebook (`.ipynb`) file(s) you use to run the experiments in
- and your local version of the `mlp` package including any changes you make to the modules (`.py` files).

Please do NOT include a copy of the other files in your `mlpractical` directory as including the data files and lab notebooks makes the submission files unnecessarily large.

There is no need to hand in a paper copy of the report, since the pdf will be included in your submission.

You should EITHER (1) package all of these files into a single archive file using `tar` or `zip`, e.g.

```
tar -zcf coursework2.tar.gz notebooks/Coursework_2.ipynb mlp/*.py reports/coursework2.pdf
```

and then submit this archive using

```
submit mlp 2 coursework2.tar.gz
```

OR (2) copy all of the files to a single directory `coursework2` directory, e.g.

```
mkdir coursework2
cp notebooks/Coursework_2.ipynb mlp/*.py reports/coursework2.pdf coursework2
```

and then submit this directory using

```
submit mlp 2 coursework2
```

The `submit` command will prompt you with the details of the submission including the name of the files / directories you are submitting and the name of the course and exercise you are submitting for and ask you to check if these details are correct. You should check these carefully and reply `y` to submit if you are sure the files are correct and `n` otherwise.

You can amend an existing submission by rerunning the `submit` command any time up to the deadline. It is therefore a good idea (particularly if this is your first time using the DICE submit mechanism) to do an initial run of the `submit` command early on and then rerun the command if you make any further updates to your submission rather than leaving submission to the last minute.

Backing up your work

It is **strongly recommended** you use some method for backing up your work. Those working in their AFS homespace on DICE will have their work automatically backed up as part of the [routine backup](#) of all user homespaces. If you are working on a personal computer you should have your own backup method in place (e.g. saving additional copies to an external drive, syncing to a cloud service or pushing commits to your local Git repository to a private repository on Github). **Loss of work through failure to back up does not constitute a good reason for late submission.**

You may *additionally* wish to keep your coursework under version control in your local Git repository on the `coursework2` branch. This does not need to be limited to the coursework notebook and `mlp` Python modules - you can also add your report document to the repository.

If you make regular commits of your work on the coursework this will allow you to better keep track of the changes you have made and if necessary revert to previous versions of files and/or restore accidentally deleted work. This is not however required and you should note that keeping your work under version control is a distinct issue from backing up to guard against hard drive failure. If you are working on a personal computer you should still keep an additional back up of your work as described above.

Marking Scheme

- Experiment 1 (20 marks). Marks awarded for experimental hypothesis / motivation, completeness of implementation, experimental methodology, experimental results, discussion and conclusions.
- Experiment 2 (25 marks). Marks awarded for experimental hypothesis / motivation, completeness of implementation, experimental methodology, experimental results, discussion and conclusions. Weighted by difficulty of task.
- Experiment 3 (25 marks). Marks awarded for experimental hypothesis / motivation, completeness of implementation, experimental methodology, experimental results, discussion and conclusions. Weighted by difficulty of task.
- Presentation and clarity of report (20 marks). Marks awarded for overall structure, clear and concise presentation, providing enough information to enable work to be reproduced, clear and concise presentation of results, informative discussion and conclusions.
- Additional Excellence (10 marks). Marks awarded for significant personal insight, creativity, originality, and/or extra depth and academic maturity.