COMP5329 - Deep Learning

Assignment-1

Due: 11-May-2018 5:00 p.m. (Week 9)

1. Task description

Based on the codes given in Tutorial: Multilayer Neural Network, you are required to accomplish a multi-class classification task on the provided dataset. Candidate modules to establish the neural network and their corresponding marks are given in the table below.

	Module	Mark
Optional	More than one hidden layer	5
	ReLU activation	5
	Weight normalization	5
	Momentum in SGD	5
	Dropout	10
	Softmax and cross-entropy loss	10
Bonous	Mini-batch training Batch Normalization Other advanced modules	

You must guarantee that the submitted codes are self-complete, and the newly implemented modules can be successfully run in common python3 environment.

You are **NOT** allowed to use Deep Learning frameworks (e.g. Tensorflow, Caffe, and KERAS), or any kinds of auto-grad tools (e.g. autograd) Scientific computing packages, such as NumPy and SciPy, are acceptable.

If you have any question about the assignment, please contact:

Dalu Guo dguo8417@uni.sydney.edu.au

Jiayan Qiu jqiu3225@uni.sydney.edu.au

2. Dataset

The dataset can be downloaded from **Canvas**

There are 10 classes in this dataset. The dataset has been splited into training set and test set, where the training set has 60,000 examples and the test set has 10,000 examples.

To read the hdf5 file and load the data into a numpy array, use the following code:

```
with h5py.File('train_128.h5','r') as H:
    data = np.copy(H['data'])
with h5py.File('train_label.h5','r') as H:
    label = np.copy(H['label'])
```

The performance of your neural network will be evaluated in terms of the accuracy metric, i.e.

$$accuracy = \frac{\text{number of correct classifications}}{\text{total number of test examples}} * 100\%$$

3. Instructions to hand in the assignment

3.1 Go to Canvas and upload the following files/folders compressed together as a zip file

a) Report (a pdf file)

The report should include each member's details (student ID and name)

b) Code (a folder)

i. Algorithm (a sub-folder)

Your code (could be multiple files or a project)

ii. Input (a sub-folder)

Empty. Please do NOT include the dataset in the zip file as they are too large. We will copy the dataset to the input folder when we test the code.

iii. Output (a sub-folder)

"Predicted_labels.h5" - This file contains the predicted labels of test exampels and must be in the output folder. We will use this file for grading.

If you work as a group, only one student needs to submit the zip file which must be named as student ID numbers of all group members separated by underscores. E.g. "xxxxxxxx_xxxxxxxx_xxxxxxxx .zip"

3.2 Your submission should include the report and the code. A plagiarism checker will be used. Clearly provide instructions on how to run your code in the appendix of the report.

3.3 The report must clearly show (i) details of your modules, (ii) the predicted results from your classifier on test examples, (iii) run-time, and (iv) hardware and software specifications of the computer that you used for performance evaluations.

3.4 There is no special format to follow for the report but please make it as clear as possible and similar to a research paper.

3.5 A penalty of MINUS 1 (one) points per each day after the due date. Maximum delay is 7 (seven) days, after that assignments will not be accepted.

3.6 Remember, the due date to submit them on Canvas is **11-May-2018, 5:00PM**

4. Marking scheme

Category	Criterion	Marks	Comments
Report [50]	Introduction [5] - What's the aim of the study? - Why is the study important?		

	Methods [15] - Pre-processing (if any) - The principle of different modules		
	Experiments and results [15] - Accuracy - Extensive analysis		
	Discussion [5] - Meaningful and relevant person reflection		
	Conclusions and [5] - Meaningful conclusions based on results		
	Other [5] - At the discretion of the marker: for impressing the marker, excelling expectation, etc. Examples include fast code, using LATEX, etc.		
Basic Modules [40]	More than one hidden layer [5]		
	ReLU activation [5]		
	Weight normalization [5]		
	Momentum in SGD [5]		
	Dropout [10]		
	Softmax and cross-entropy loss [10]		
Bonous [+]	Mini-batch training		
	Batch Normalization		

	Other advanced modules		
	Top-10 Accuracy		
Code [10]	Code runs within a feasible time [5]		
	Well organized, commented and documented [5]		
Penalties [-]	Badly written code: [-20]		
	Not including instructions on how to run your code: [-30]		
	Late submission: [-1] for each day late		