CITY UNIVERSITY LONDON

EST 1894

IN3063/INM702: Mathematics and Programming for AI
Coursework

V1.0

Submission deadlines:
Report and Code: Sunday 20th December 2020, 5pm
Presentation: Sunday 12th January 2021, 5pm

Introduction

This coursework builds on the material covered in the tutorials and lecture. On completing this coursework, you should be able to implement and understand advanced neural network techniques from scratch. This coursework builds on the material covered in the tutorial Jupyter notebooks and lecture slides You will make use of the different concepts learned in the module:

- How to convert mathematical principles into algorithms
- How to implement those algorithms in Python
- How to organize your ideas in an appropriate code structure
- How to evaluate different algorithms

Python should be used for all implementations. Deliverables are:

- Written report of your work (max 10 pages, split in case of teamwork)
- Your practical implementation (code)
- For INM702 only: an individual presentation (10 minutes) recorded and submitted in January

Module marking:

INM702: 70% Coursework (Code and Report) and 30% Presentation.
IN3063: 100% Coursework

Teamwork

This coursework should be completed either in **groups of two** or **individually**; We encourage you to work in pairs, and no additional marks will be granted if you do the Coursework alone. If you decide to work in pair, you should declare it on the report.

All team members are expected to contribute to all parts of the work: both the coding and the report. Teamwork does NOT mean division of labour. You can distribute the leading role for

each assignment, but each of you must contribute to all the tasks. If you don't, you will not be evaluated for the tasks that you did not prepare. Distributing the assignments is considered a form of academic misconduct.

You are required to explain your personal contribution to each task in the coursework report, in the reflection part.

For MSc AI students, there is a maximum number of modules you can work within the same team. You cannot operate with the same team in more than 2 modules per term and no more than 4 in total.

The Coursework is divided into 4 different Tasks. Even if you are in a team, Task 1 should be solved individually, and no teamwork is allowed for the Task 1.

Submission

Submission is through Moodle, and no other method of submission will be accepted. You should submit the following files:

- Report number 1: Task 1, 2 pages
- Report number 2: Task 2-4, including a reflection on the work, 8 pages
- Zip file of the Code for Task 1
- Zip file of the Code for Task 2-4

In addition:

- your code must be developed and available on a git server (github or otherwise), with a full revision history indicating who has pushed what code. This repository should be available to Michael Garcia Ortiz and Atif Riaz.

Format for reports: pdf format, single column, standard A4 margins, standard default line spacing of 1.15, font Arial 11, including all figures. If you are working alone, you can present a single report and a single zip for the code.

Late submissions will score 0. You can upload work to Moodle more than once, so there is no need for last minute submission. Don't leave final submission to the last minute.

Presentation (INM 702 only)

You will be recording an individual presentation of 10 minutes. This is a free-form exercise where the important point is to present your work. During this presentation, you will present the results for the 4 tasks, and then go into details for either Task 2 or 4.

This is an individual exercise.

Feedback

In the labs and surgeries, we can check your progress and give formative feedback. Evaluative feedback and marks on your coursework will be give out after the presentations.

Coding

- Each task should be presented as an individual Jupyter Notebook with possibly additional modules and packages that you developed and are used by the notebooks.
- Code quality, clarity and organization will be taken into account in the marking.

The Tasks

In this coursework, you are expected to demonstrate what you have learned in the module in terms of Programming, Neural Networks, and Deep Learning. Additionally, you have the occasion to work on an optional problem of your choosing for additional points.

The maximum number of marks which can be scored is 100. You can gain up to 20 additional marks in Task 4, but the maximum of marks obtained for the Report and Code is capped at 100.

In all tasks, you can use the built-in libraries of python (math, random, …), numpy, and matplotlib. If you think that you might benefit from using another library, you can ask about it on Moodle.

You will use PyTorch in Task 3, and you are allowed to use any library in Task 4.

Note that you can use any library for the purpose of loading the training and testing set of the MNIST dataset for Task 2 and 3, but we advise to use torchvision.

Task 1: 25 marks

The first task tests your Python skills. You need to develop a simple game consisting of a rectangular grid (of size height x width) where each cell has a random integer value between 0 and 9. An agent starts at the upper-left corner of the grid and must reach the lower-right corner of the grid as fast as possible.

You can implement one of the two (or both, for no extra point) game modes:

- The time spent on a cell is the number on this cell

- The time spent on a cell is the absolute of the difference between the previous cell the agent was on and the current cell it is on

In order to solve this problem, you will provide 3 algorithms:

- A first baseline of your choosing. E.g. it can be any search algorithm, or an algorithm using heuristics. It doesn't have to perform fast or well, but should be better than random movements.
- Dijkstra's algorithm
- Ant colony optimization algorithm

You should describe the algorithms and compare them. Are they always solving the problem? How long do they take depending on the size of the maze?

Task 2: 50 Marks

The second task is about classifying handwritten digits. We will use the MNIST dataset for training and testing.

The point of this task is to develop a multi-layer neural network for classification using mostly Numpy:

- Implement sigmoid and relu layers (with forward and backward pass)
- Implement a softmax output layer
- Implement a fully parameterizable neural network (number and types of layers, number of units)
- Implement an optimizer (e.g. SGD or Adam) and a stopping criterion of your choosing
- Train your Neural Network using backpropagation.

Evaluate different neural network architectures and compare your different results.

You can also compare with the results presented in http://yann.lecun.com/exdb/mnist/

Task 3: 25 Marks

The third task is about comparing your results with architectures developed using PyTorch.

Compare the results obtained in Task 2 to the results obtained using the same architectures implemented in PyTorch.

Then, propose improvements and new architectures that make use of more advanced methods (e.g. Convolutional Neural Networks, dropout, …).

Compare the results.

Finally, present the confusion matrix of your best model.

Task 4: 20 Marks – optional

For this last optional task, you can work on a topic of your choosing. It must be related to the Module, and different from Tasks 1 to 3.

Reports

Each report must have an additional first title page (not included in the page count), and as many references as needed (not counting in the page total).

Your final report should cover each of the aspects above (and any other element of your work that you believe should be reported). Graphically illustration of your results is expected as well as numerical results.

You should present the results clearly and concisely and provide a discussion of the results, with conclusions related to problem being addressed. The conclusions section might propose some further work based on the results of this coursework.

Of particular importance, you should indicate on the report on estimate of the percentage of code that you borrowed from external sources for each task. This will participate in the evaluation of your work. Failure to do so will lead to a fail mark.

Reflection

In the case of teamwork, the reflection part should briefly address who did what. You might also report further conclusions and discussion of your work from your individual perspective in this section.

Note

You are not necessarily being marked on how good the results are. What matters is that you try something sensible and clearly describe the problem, method, what you did, and your interpretation of the results.

Coding & Referencing

This is, in large part, a coding assignment. If you use code (or other materials) written by someone else, you must **cite** that code (or other material). If you do not cite work appropriately you will have committed academic misconduct. Making superficial changes to the code does not make it yours. You are also expected to make a coding contribution, so if you use a large amount of code written by someone else, and cite it appropriately, your contribution will be low, and your work marked accordingly.

Extenuating Circumstances

If you are not able to submit your coursework on time for unforeseen medical reasons or personal reasons beyond your control you should contact the Programmes Office as soon as possible and fill an Extenuating Circumstances form. Strong evidence in the form of, for instance, medical certificates or legal statements will have to be produced.

Plagiarism

If you copy the work of others (either that of another team or of a third party), with or without their permission, you will score no marks and further disciplinary action will be taken against you. The same applies if you allow others to copy your work.