

COMP222 - 2018 - Second CA Assignment
Individual coursework
Train Deep Learning Agents

Assessment Information

Assignment Number	2 (of 2)
Weighting	10%
Assignment Circulated	16 November 2018
Deadline	Friday December 28, 15:00
Submission Mode	Electronic
Learning outcome assessed	3. Ability to explain how deep neural networks are constructed and trained, and apply deep neural networks to work with large scale datasets 4. Understand reinforcement learning, and is able to develop deep reinforcement learning algorithms for suitable applications
Purpose of assessment	To design and implement deep learning agents for either image classification task or reinforcement learning task.
Marking criteria	The marking scheme can be found in Section 3.2
Submission necessary in order to satisfy Module requirements?	No
Late Submission Penalty	Standard UoL Policy.

1 Objective

This assignment requires you to *implement* one of the following tasks:

- two image classifiers with convolutional neural networks,
- a deep reinforcement learning model for a video game from OpenAI Universe.

Considering their different difficulties and the fact that the second needs at least an implementation of the first, we enable the possibility of getting 140% marks for the completion of the second.

In the following, the Requirement and Description and the Marking Criteria for the two tasks are explained separately.

2 CNN-based Image Classification

2.1 Requirement and Description

Language and Platform Python (version 3.5 or above) and Tensorflow (newest version). You may use any libraries available on Python platform, such as numpy, scikit-learn, panda, etc.

Dataset You can use any dataset which is convenient for you. Unless exceptional circumstance, it is recommended that the dataset is not too small (e.g., no less than 10,000 items) and not too big (e.g., no more than 100,000 items). The images in the dataset are not too large, as that will cost you too much time on training a good model. There are a few suggested datasets:

- MNIST handwritten dataset;
- CIFAR10 small image dataset;

but you are encouraged to use your preferred dataset.

Learning Task You need to train two classifiers on the chosen dataset.

DNN Architecture There are a few existing DNN architectures for such small scale image dataset, including

- LeNet, see tutorial e.g., goo.gl/S6RQiS
- AlexNet, see tutorial e.g., goo.gl/UkjY6u

But you can design your own architecture.

Assignment Tasks The implementation task (as suggested in the Objective) is to learn two models (of different architecture) from the dataset you select. You need to write a proper document explaining the architectures, your training parameters, and the results (e.g., accuracy).

Submission files Your submission needs to contain the following two files:

- a package containing your source codes (with the instruction on how to run them) and
- a document explaining your architectures, your training parameters, and the results. Detailed requirements on the document are given below.

2.2 Marking Criteria

The assignment is split in a number of steps. Every step gives you some marks. At the beginning of the submitted document, please include a check list indicating whether the below marking points have been implemented successfully. Unless exceptional cases, the length of the submitted document needs to be within 4 pages (A4 paper, 11pt or 12pt font size).

Step 1: Loading Data 10%

Successfully load the dataset and use python commands to display the dataset information, e.g., the number of data entries, the number of classes, number of data entries for each classes, etc.

Step 2: Write two CNN models 20%

Write your CNN models with tensorflow. To get the marks, you need to explain in the document what the inputs and outputs are and what the hidden layers are.

Step 3: Train your CNN models 30%

To get the marks, you need to explain in the document the training parameters, e.g., learning rate, loss function, etc, and the accuracy of your resulting models over testing dataset.

Step 4: Predict with Trained Model, 20%

You need to be able to predict the class label by giving an image. For example, predict the label for the 100th image in the MNIST test dataset. To get the marks, you need to clearly identify in the document which part of the code is for this purpose.

2.3 Extra 20%

You can see that marks for the steps described add up to 80%. In order to get 20% extra you have to be creative. For example (you are encouraged to not follow this example), you may adjust your models by adding or removing some layers and compare the performance of the resulting models. You are expected to have a clear explanation on what you have done and what you have observed.

3 Deep Reinforcement Learning

3.1 Requirement and Description

Language and Platform Python (version 3.5 or above) and Tensorflow (newest version). You may use any libraries available on Python platform, such as numpy, scikit-learn, panda, etc. You may also need to install OpenAI universe (<https://github.com/openai/universe>) if you choose the second challenge.

Dataset You can use any game in OpenAI universe or OpenAI gym.

Learning Task You need to train a deep reinforcement learning model to play the game you selected.

DNN Architecture There are a few existing deep reinforcement models, including

- Deep Q-Network
- Double Q-Learning
- Actor-Critic algorithm

You are able to find various tutorials and implemenations from Github, for example [goo. gl/1q6L31](https://github.com/goo11q6L31).

Assignment Tasks The implementation task (as suggested in the Objective) is to train a deep reinforcement agent for a simple video game. Here, the main task is not to design a new algorithm, but to get yourself familiar with the concept of reinforcement learning and understand how the existing methods work.

Submission files Your submission needs to contain the following two files:

- a package containing your source codes (with the instruction on how to run them) and
- a document explaining your connection to openAI universe/gym, your network architectures, your training parameters, and the results. Detailed requirements on the document are given below.

Note: please make sure that you either submit your dataset along with these files or provide clear instructions on how to download the dataset. Please keep in mind the markers won't have plenty of time to be spent on working out how to run your program. So to ensure that you get a fair mark, please provide clean and sufficient instructions.

3.2 Marking Criteria

The assignment is split in a number of steps. Every step gives you some marks. The submitted document should include the following headings (e.g., Step 1, Step 2, etc) and provide relevant information. At the beginning of the submitted document, please include a check list indicating whether the below marking points have been implemented successfully

Step 1: Import an OpenAI universe/gym game 20%

As the starting point, you need to import a game.

Step 2: Creating a network 20%

You need to use a deep neural network.

Step 3: Connection of the game to the network 10%

You will need to explicitly associate the observations, actions, and rewards of the game to the network's input and output. Clearly identify this part of the code in your document.

Step 4: Deep reinforcement learning model 30%

This part is for different deep reinforcement learning models. You need to clearly state which model you are using, the parameters of the model, and how do you train/update the model.

Step 5: Experimental results 20%

You may record a video demo to exhibit what your agent can do. Alternatively, you can describe in details with texts.

4 Deadlines and How to Submit

- Deadline for submitting the second assignment is Friday, 28 December 2018 at 3pm.
- Submission is via the departmental submission system accessible (from within the department) from
<http://intranet.csc.liv.ac.uk/teaching/modules/module.php?code=COMP219>.