COMP222 - 2019 - Second CA Assignment
Individual coursework
Train Deep Learning Agents

Assessment Information

Assignment Number	2 (of 2)
Weighting	10%
Assignment Circulated	Thursday 14 November 2019
Deadline	Monday 16 December 2019, 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
Purpose of assessment	To design and implement deep learning agents for classification task
Marking criteria	The marking scheme can be found in Section 2.2
Submission necessary in order to satisfy Module requirements?	No
Late Submission Penalty	Standard UoL Policy.

1 Objective

This assignment requires you to *implement* deep neural networks for the two datasets, i.e.,

- Optical recognition of handwritten digits dataset
- RCV1 dataset

from `https://scikit-learn.org/stable/datasets/index.html`, and *apply* the model evaluation methods to compare them with the two models in Assignment 1. Please make sure that you select the same dataset as you did for the Assignment 1, if you completed the Assignment 1.

2 DNN-based Classification

2.1 Requirement and Description

Language and Platform Python (version 3.5 or above) and Tensorflow or Keras (latest version). You can use some libraries available on Python platform, including numpy, scipy, scikit-learn, and matplotlib. If you intend to use libraries other than these, please consult the demonstrator or the lecturer.

Learning Task You can choose either classification (preferred) or regression, but needs to be the same choice as your Assignment 1 submission.

Assignment Tasks You need to implement the following functionalities:

f1 design and build two different deep neural networks, one with convolutional layer and the other without convolutional layer;

f2 apply model evaluation on the learned models. For the materials on model evaluation, you may take a look at the metrics explained in the lecture "model evaluation". You are required to implement by yourself (i.e., do not call built-in libraries)

(a) the cross-validation of 5 subsamples,

(b) the confusion matrix, and

(c) the ROC curve for one class vs. all other classes

for

(a) the two neural networks you trained in **f1**, and

(b) the two traditional machine learning algorithms in the first assignment.

Please also summarise your observation on the results.

Additional Requirements We have *additional requirements* that,

1. the marker can run your code directly, i.e., see the results of functionality **f1** by loading the saved models, without training.
2. You need to provide clear instructions on how to train the two models. The instructions may be e.g., a different command or an easy way of adapting the source code.

Documentation You need to write a proper document

1. detailing how to run your program, including the software dependencies,
2. explaining how the functionalities and additional requirements are implemented, and
3. providing the details of your implementation, including e.g., the meaning of parameters and variables, the description of your model evaluation, etc.

Submission files Your submission should include the following files:

- a file for source code,
- two files for saved models, and
- a document.

Please see Section 3 for instructions on how to package your submission files, and read the Q&A on whether to upload the two trained models from the first assignment.

2.2 Marking Criteria

The assignment is split in a number of steps. Every step gives you some marks.

Note 1 *At the beginning of the 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 font size).*

Note 2 *The marking of a functionality will also consider the quality of coding and the quality of documentation. A run-able implementation alone will have up to 50% of the marks.*

functionality f1: 50%

For each model (with and without convolutional layer), 20% will be for the model construction and 5% will be on the model saving and the model file in the submission.

functionality f2: 50%

The model evaluation between will include

- cross validation (10%)
- confusion matrix (10%)
- ROC curve (20%)
- discussion on the discovery (10%)

For each of the four parts, 80% of the marks are for deep learning models, while 20% are for the traditional models in the first assignment. For example, for cross validation part, if you only do deep learning models, your marks are capped at 8% instead of 10%.

The marker will mark according to the quality of both your evaluation and the documentation.

3 Deadlines and How to Submit

- Deadline for submitting the first assignment is given at the beginning of this document.
- Please submit all the files in a single compressed file with the filename

  $''\langle studentnumber\rangle.tar'' \quad or \quad ''\langle studentnumber\rangle.zip''$

  For example, "201191838.tar" or "201191838.zip" if your student number is 201191838. Submissions with other filename will not be accepted. Also, in the submission files, please do not include your name.
- Submission is via VITAL Turnitin system.

4 Q&A

Q: The ROC curve we taught in the lecture is for binary classification, but the models we trained are for multiple classes. What can we do?

A: As indicated, you can have one class vs. all other classes, where all other classes are deemed as a single class.

Q: My models in the first assignment can output a classification but not a confidence probability. What can we do for ROC curve?

A: If you think some functionality is hard to implement, please explain in the document. The marker will then evaluate your explanation to give you a reasonable mark.

Q: Since we are requested to evaluate the two models from our first assignment, shall we upload again?

A: You can upload them again if needed. Note that, the marker won't be able to access the first assignment when they are marking the second assignment.