MPHY0041: Machine Learning in Medical Imaging
Assessed Coursework 2 – 2022/23

Available on 17th March 2023
Submission before 16:00 – 5th May 2023, at UCL Moodle submission section

WHAT NEEDS TO BE SUBMITTED

Code (20%)

- ✓ Python code: all python code used to complete the tasks [10 marks for completeness[1]].
- ✓ "Instruction.pdf": a text document that outlines the steps to reproduce the results [10].

Scientific Report (80%)

- ✓ "report.pdf": This coursework requires to summarise the completed tasks (in the Tasks section) in a scientific report format. The Lecture Notes in Computer Science template is recommended and please use the font sizes and page margin specified in the template without modifications. The report should not exceed 6 pages excluding references, submitted in PDF format. The report should include the following sections to *cohesively* address what is described in the Project section.
  - Introduction [15]
  - Methods [15]
  - Experiments [25]
  - Results [20]
  - Discussion and Conclusion [5]

Peer Assessment

There is no mandatory peer assessment. The same grade on the code from each group will be assigned to all group members. However, cases of negligence or any other academic malpractice, if reported, will be investigated and assessed per UCL regulations, on a case-by-case basis.

THE PROJECT

Marking environment

IMPORTANT: The conda-based *marking environment*, updated from the one used in the module repository, will also be used for marking the submitted Python code and for downloading the provided data set. In this project, you can install an additional library nibabel for reading and writing nifti images, as follows:

```
conda create --name mphy0041-cw2 -c conda-forge numpy nibabel matplotlib
tensorflow=2.10 pytorch=1.12
```

Data set

This project uses a publicly open database from UCL, which includes 589 T2-weighted MR images of male pelvic region, with eight anatomical structures segmented. See further details in the Zenodo page and the associated paper. Additional public datasets can also be used, which need to be specified in your code and report.

[1] Square bracket contains the marks in each task.

Which tasks can help which task(s)

Multi-task learning has been motivated by an intuition that learning a good representation during one or more auxiliary tasks may benefit a different task, the main task. This is particularly interesting when learning the main task is challenging due to, for example, data scarcity and labels with high variance. Medical image segmentation is such an example. In this project, segmentation of a particular set of regions of interest (ROIs) in T2-weighted MR images of pelvic is useful for imaging-based surgical planning for prostate cancer intervention. This main task may be assisted by segmenting a number of other surrounding structures or classification tasks to discriminating whether certain ROIs are present.

This project aims 1) to motivate and justify a main segmentation task, including understanding the data set and its processing; 2) to hypothesize what can be a viable set of auxiliary tasks that improve the main task; 2) to design experiments to test the hypothesis; 3) to develop deep neural networks for multi-task learning for these experiments; and 4) to summarise and report relevant findings. The submission is expected to contain a set of experimental results, summarised in the submitted report and reproducible by the submitted code.

The goal of this project is to assess your ability to implement deep learning applications, run experiments using real clinical image data, and evaluate the developed deep learning models, akin to conducting a scientific or technical research study. Your findings are to be compiled in a report which will follow a similar structure to a research paper. The report together with the submitted code will be marked for its scientific soundness, technical accuracy, completeness, presentation and critical appraisal. Novelty, such as new method for improving networks and well-designed clinically relevant metrics, is encouraged but will be limited to 20% of each section.

The following describes example considerations in each section. However, this is a list that is neither comprehensive nor compulsory.

1. **Introduction**:
    a. Background and existing work.
    b. The identified main (segmentation) task and auxiliary tasks, and their motivations.
    c. A summary of the project.
2. **Methods**: A description of the adopted multi-task learning algorithms, any adaptation that is desirable for the intended tasks.
3. **Experiments**: that train and test such neural networks, in order to answer the following two study questions, at minimum:
    a. Demonstrate whether the identified auxiliary tasks benefit the main task.
    b. Any adaptation and changes that can be made for improving the multi-task benefit.
4. **Results**: clearly summarise and, if useful, illustrate quantitatively and qualitatively the key results support any conclusions the project can draw.
5. **Discussion and Conclusion**: summary of the results, limitations, future extensions.

OTHER IMPORTANT INFORMATION

Python with TensorFlow/PyTorch Development

Python is required in this coursework with either TensoFlow2 or PyTorch, as in the module tutorials. The coursework marking environment is specified above.

- You can use up to 3 pip-installable external packages in this coursework. If you do, please include an instruction text file to details of the installation and inclusion of these packages.
- Only *.py files need to be submitted to avoid large files, without unnecessary system-dependent configurations.
- Please do not submit the downloaded data set!

- Notebook files will not be marked.
- It is the group's responsibility to ensure that the submitted code can be successfully run using the marking environment. It is recommended to have the provided **data.py file** in the unzipped folder with other ***.py files** and the **Instruction.pdf** document.
- For technical support for work with such an environment, please refer to the documents available in the module repository.

Computing Resource

Groups may consider to use CMIC HPC Cluster hosted at CS Department and other publicly available parallel computing facilities, such as Google Colab. The use of the parallel computing and other high-performance computing is not assessed in this coursework. With proper configurations, each network training in this coursework is expected to take no more than several hours on a modern Nvidia GPU card. Training on CPU may take a significantly longer time but is largely dependent on the hardware. However, it is feasible to develop, debug and run the inference/testing procedures on a personal computer.

Patient Data

Although the image data this project has access to are anonymised and with patient consents for research purposes. These are still sensitive clinical data and need to be treated with highest ethical standards and caution. Care must be taken when storing, transferring and processing the images and labels and, **under no circumstances**, these can these be used for any other purposes outside this coursework, academic research or be distributed further. Upon the completion of the project, all the copies of the data should to be removed completely from any computer or cluster storage.

THE CHECKLIST

This is a list of things that help you to check before submission.

- ✓ All the code are zipped in a single file, e.g. code.zip
- ✓ The zip file does not contain files other than *.py Python code and Instruction.pdf.
- ✓ The zip file does not contain downloaded data.
- ✓ All the scripts run in the unzipped folder, within the marking environment without requiring additional libraries or packages.
- ✓ A single PDF report file is submitted, e.g. report.pdf
- ✓ The report has the sections described above and appropriate references.

Academic Integrity

UCL Guides for Academic Integrity includes guides to references, citations and avoiding plagiarism.

References

Ruder, S., 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Zhang, Y. and Yang, Q., 2021. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, *34*(12), pp.5586-5609.