

ACS61011 Deep Learning Assignment: Individual Project

General Assignment Information

Assignment weighting

20%

Assignment summary

The assignment is to design, implement and evaluate an automated speech recognition (ASR) system using deep learning methods.

Assignment start date

Week 5

Assignment due date

The assignment is due at 23:59 pm on Monday 21st March (start of week 7).

Assignment supporting materials and data

All assignment instructions, supporting material and data are on Blackboard in the ACS61011 course pages, under the Coursework/Quizzes section

Submission

You will have to submit a short Technical Report as a Word document in Blackboard under the Turnitin link in the ACS61011 Coursework/Quizzes section by the due date.

Penalties for Late Submission

Late submissions will incur the usual penalties of a 5% reduction in the mark for every working day (or part thereof) that the assignment is late and a mark of zero for submission more than 5 working days late.

Unfair Means

The assignment should be completed individually. You should not discuss the assignment with other students and should not work together in completing the assignment. The assignment must be wholly your own work. Any suspicions of the use of unfair means will be investigated and may lead to penalties. See <http://www.shef.ac.uk/ssid/exams/plagiarism> for more information.

Special Circumstances

If you have medical or personal circumstances which cause you to be unable to submit this assignment on time or that may have affected your performance, follow the guidance at <https://www.sheffield.ac.uk/ssid/forms/circs>

Help

This assignment briefing and the lecture notes provide all the information that is required to complete this assignment. It is not expected that you should need to ask further questions. However, if you need clarifications on the assignment then please discuss the issue with me after a lab, or email me s.anderson@sheffield.ac.uk or make a virtual appointment to meet.

Specific assignment information and instructions

Data

Speech files are provided on Blackboard in the Coursework/Quizzes section in the file speechDataReduced.zip. There are about 20 different words and 1000 examples of each word, plus a folder with some background noise. Unzip the folder and put it into your working directory for the project.

Initial Code and Matlab versus Python Tools

Initial Matlab code to get started is provided on Blackboard in the file main.m

It is expected that most people will probably use Matlab and the deep learning tools within Matlab to complete this project but there is no requirement to do so. The task and mark scheme (shown over the page) should apply to any implementation environment.

Matlab requirements: you will need

- **Matlab,**
- **the Deep Learning Toolbox and**
- **the Audio Toolbox**

If you feel confident using Python-based tools then you have the freedom to do that. You should use the dataset provided on Blackboard for the initial tasks regardless of whatever environment you choose to use in order that everyone's results are comparable.

Note that extracting audio-based spectrograms from the raw .wav files of speech will require some digital signal processing. If you choose to use Python you may wish to use Matlab (and the code provided in Matlab) to perform the audio signal processing steps and then switch to Python for the Deep Learning part.

Useful background reading

Here is a source that provides useful background reading:

- The use of convolutional neural networks for speech recognition is described in Abdel-Hamid, O., Mohamed, A. R., Jiang, H., Deng, L., Penn, G., & Yu, D. (2014). Convolutional neural networks for speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10), 1533-1545.

Technical report

Words: For each distinct task or subtask specified in the mark scheme (over the page) you should write just enough to explain what you have done and justify your design choices – a bullet point list is fine. Use up to 100 words in the report per task – for example:

- Level 2: write no more than 100 words for the whole report (limit of 100 words per task)
- Level 3: write no more than 300 words for the whole report (limit of 100 words per task)
- Level 4: write no more than 400 words for the whole report (limit of 100 words per task)
- Level 5: write no more than 400 words + 100 words per additional open ended task.

Figures: You should include a variety of figures for each task – about four per task should be about right.

- To evidence your network design you can visualise the network with analyzeNetwork in Matlab.
- To demonstrate training/validation performance you should use a screenshot of the training/validation plot that the Deep Learning toolbox produces during training.
- To demonstrate performance you should also include a confusion plot for each task.
- Any other figures as necessary
- Example figures are given at the end of this report – note that some Matlab figures cannot be exported so you can use a print screen instead.

Code: You should include all your code in the Word document – copy and paste it into the Word document. This helps for both marking and plagiarism detection in Turnitin.

Tasks and Mark Scheme

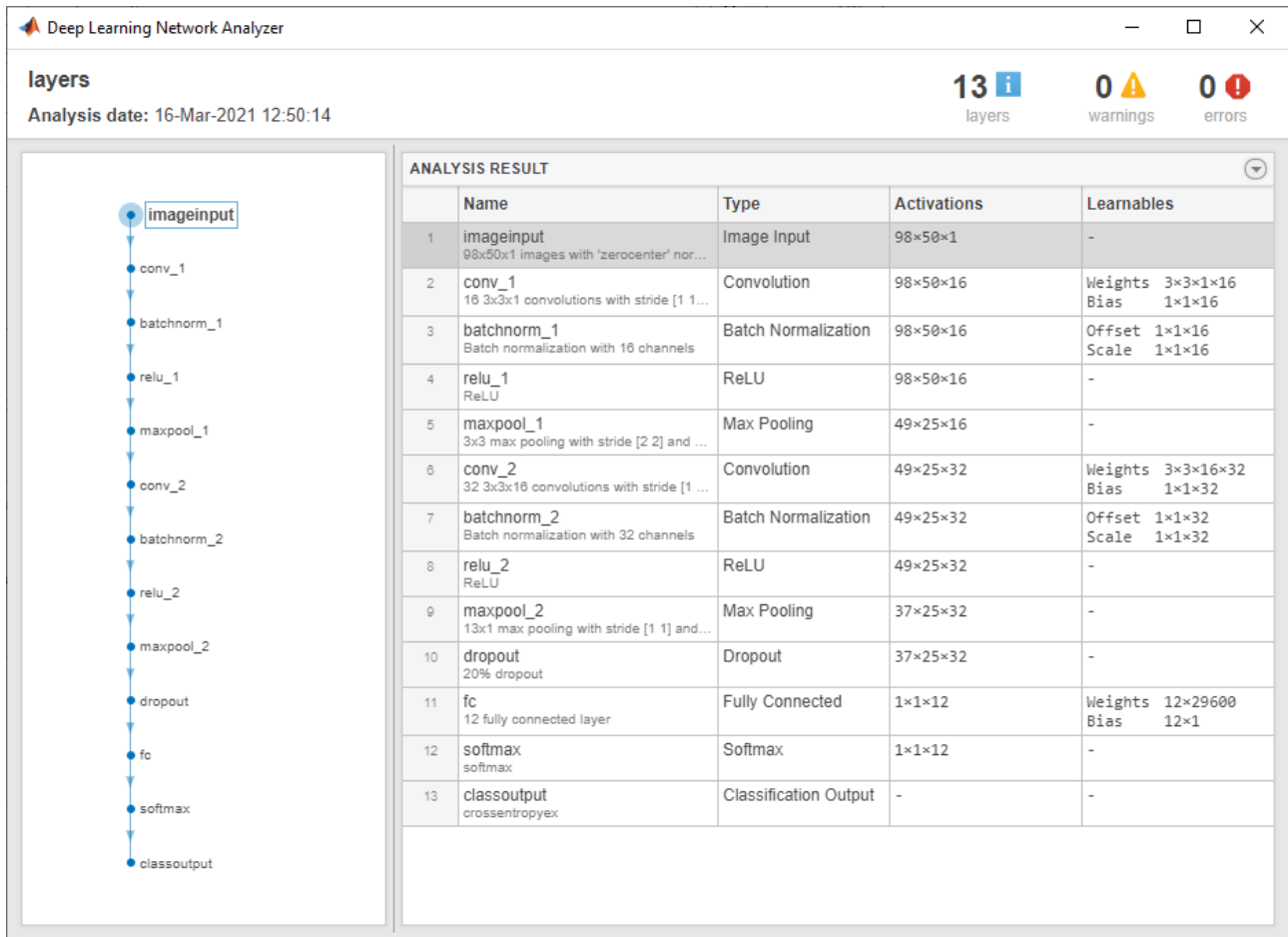
The specific tasks and corresponding mark scheme are given in the table below. It is up to you to choose what amount of work you do.

For each task, the mark within a grade boundary will be moderated based on your results and code. Note, I reserve the right to mark at a lower level if the tasks are done very poorly (or e.g. presented poorly).

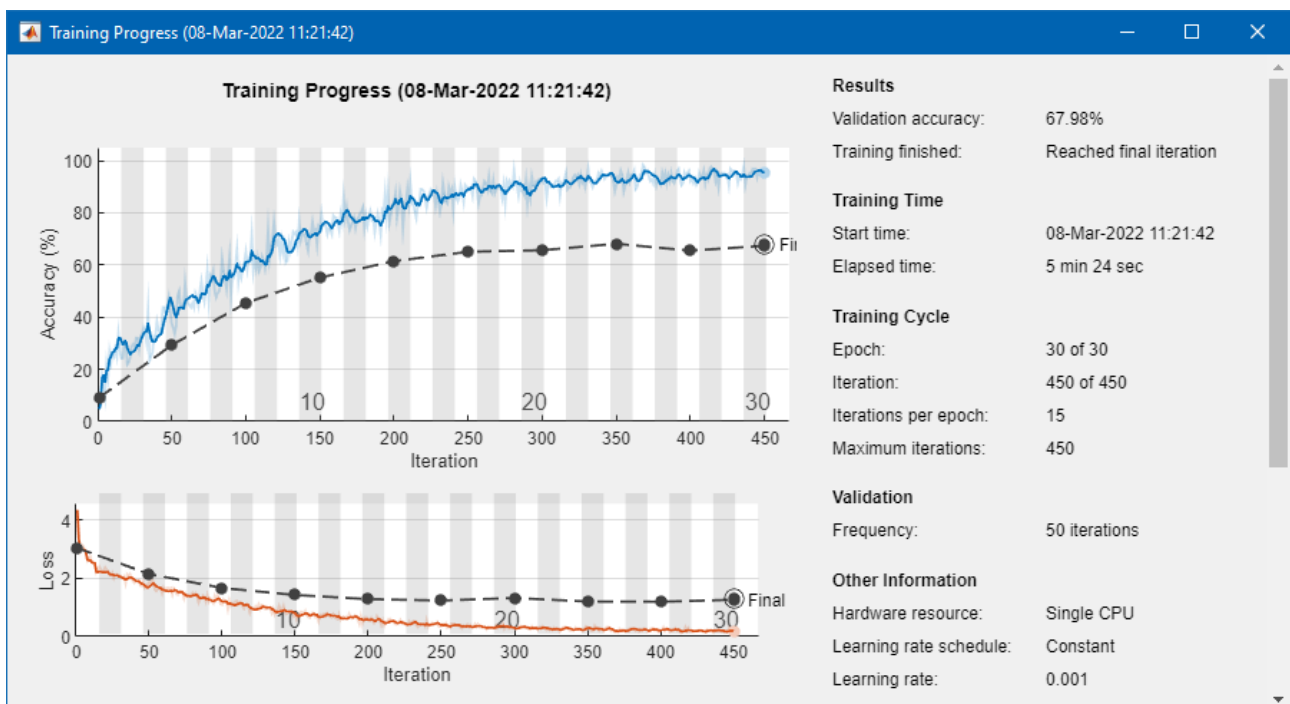
| Level of achievement | Mark Range | Task/Assessment Description |
|----------------------|------------|--|
| 1 | 0-49% | An attempt at the project to design, implement and evaluate a basic deep CNN for speech recognition, which achieves an accuracy of <60% on the validation data. Little or no results/code/evidence of model accuracy. |
| 2 | 50% | Design, implement and evaluate a basic deep CNN for speech recognition, using the data set and initial code provided, to achieve an accuracy of >60% on the validation data. Use the data already processed in the file dataPreProcess.m provided on Blackboard and do not change the file at this stage! |
| 3 | 50-60% | Achieve level 2 plus any two tasks from the following list in Matlab: -Perform a systematic investigation into the effect of adding layers and number of filters per layer to the model from level 2 – use a 3x3 grid search -Design, implement and evaluate a bagging (model averaging) scheme to regularise the network from level 2 to improve generalisation -Add at least 5 additional words to the data set, then redesign, implement and evaluate this new model (Now you can change the dataPreProcess.m file to include more data) Alternatively to carrying out two tasks from above , instead repeat the basic deep learning design with the implementation in Python |
| 4 | 60-70% | The same as level 3 but do all three of the tasks from level 3 specified above in Matlab (so if you have already done two of the tasks from level 3, just add the third one – make it clear in your report) Alternatively , instead -do the basic deep learning design with a network implementation in Python (same as Python task as at level 3 so you might have already done this at level 3; if so no need to repeat it) and Plus do one of the following in Matlab or Python, your choice: -Perform a systematic investigation into the effect of adding layers and number of filters per layer to the model – use a 3x3 grid search -Design, implement and evaluate a model averaging scheme to regularise the network and improve generalisation performance |
| 5 | 70-100% | Do level 4 and then do an open-ended extension of your choice , e.g.: -Try and make the model more accurate – e.g. you could use more advanced model designs such as GoogLeNet for this. (But don't just retrain GoogLeNet – design your own model) -perform a more sophisticated hyperparameter search using e.g. Bayesian optimization. -You can ask me if you want to check your idea, or if unsure, at a drop-in session or email s.anderson@sheffield.ac.uk |

Example figures for the technical report

Analyzenetwork figure



Training/validation plot



Confusion plot

