COMSM0098: Applied Deep Learning Coursework

Re-Produce Published Research Paper

(Task for up to 3 students per team, coursework, 100%)

Forming your Teams:

Register your team of up to 3 people (i.e. one, two, or three students) online at: https://forms.office.com/r/WbfpMAsj5c (only one person per team needs to do this).

Post registration, teams can split but cannot merge, to avoid copying of code or ideas.

Each member of the team should submit an exact copy of the final submission on Blackboard by the deadline. The report (see below) should note the full names and usernames of all members in the team as well as a signed agreement (this can be digital) that all members contributed roughly equally. All members of the group will be given the same mark.

It is up to each team to decide their best strategy to tackle this coursework, i.e. whether to divide the tasks below, or to work together on all tasks. Contributions of team members need not be explicitly stated.

However, by submitting a group coursework, you are implicitly acknowledging that all members of the team contributed approximately equally. If this is not the case, you should email the unit director with any issues encountered during the coursework (also see Appendix A).

Note, that in the past we have found no benefit in working as part of a group, there is no correlation between mark and group size. Keep in mind the communication overhead of working in a group compared to working solo.

Task Brief

This assignment gives you the opportunity to appreciate the work required in replicating published research from a publicly available dataset and manuscript. It allows you to reflect on the experience of reproducing published results and potentially outperforming on your replication.

Gathering all the knowledge you acquired from the lectures and labs, read the paper below carefully and replicate the required results (Note: you are not required to re-produce all the paper's results). Feel free to take any pieces of code from the labs as a baseline, but the rest of the code should be originally yours.

The Paper

Schindler, Alexander, Thomas Lidy, and Andreas Rauber. "Comparing Shallow versus Deep Neural Network Architectures for Automatic Music Genre Classification." FMT. 2016.
https://publik.tuwien.ac.at/files/publik_256008.pdf

Note that our choice of paper is based on its simplicity and similarity to your labs, rather than its superior performance or exceptional novelty.
Please read the following information carefully **before** attempting the replication:

1. Architecture

For this coursework, you will only be asked to implement the shallow CNN architecture (Figure 1 from the paper). You should only implement the deep CNN architecture (Figure 2) as an extension – see section J for more details. **Failure to include the shallow CNN architecture will result in 0 marks for the coursework!**

2. Selected Dataset

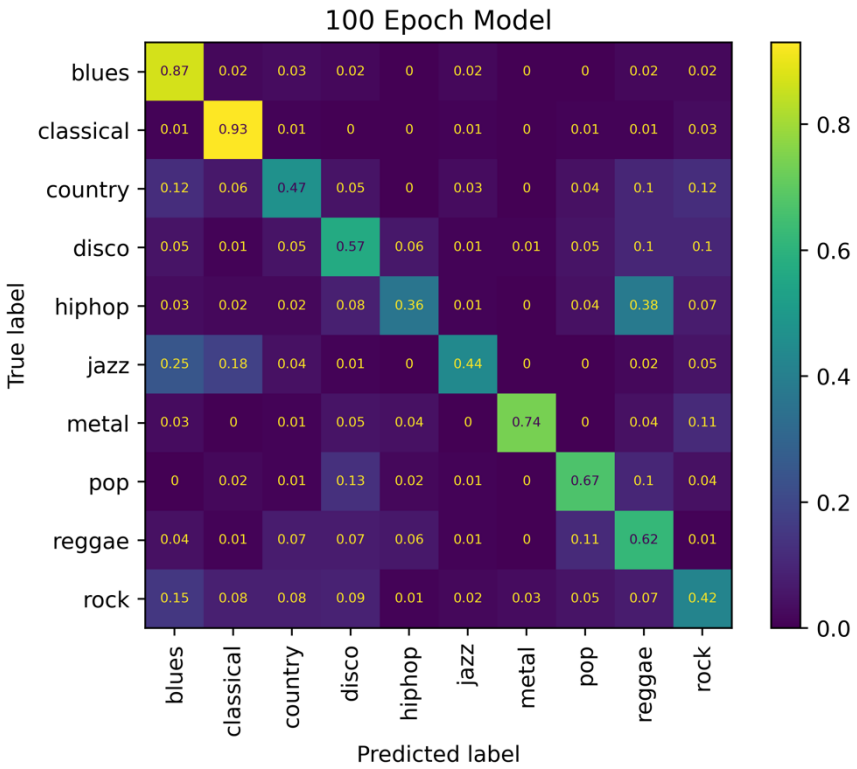
Within the paper the authors test on four different datasets. We ask only that you use the GTZAN dataset for the main part of the coursework. The paper additionally uses stratified four-fold cross validation, which we ask you **not** to use. Instead, please use the train/test split that we have provided for you to use (see Dataset and Useful Code).

3. Required Results

In replicating results, we expect you to provide the code to your results for the shallow Model on GTZAN using the **raw accuracy** for both 100 and 200 epochs, marked here from Table 2 in the paper:

D	Model	raw	max	maj	ep
GTZAN	shallow	66.56 (0.69)	78.10 (0.89)	77.80 (0.52)	100
	deep	65.69 (1.23)	78.60 (1.97)	78.00 (2.87)	100
	shallow	67.49 (0.39)	80.80 (1.67)	80.20 (1.68)	200
	deep	66.19 (0.55)	80.60 (2.93)	80.30 (2.87)	200
	shallow aug	66.77 (0.78)	78.90 (2.64)	77.10 (1.19)	100
	deep aug	**68.31** (2.68)	**81.80** (2.95)	**82.20** (2.30)	100

You should also be able to create a confusion matrix showing the performance of all 10 classes for either model. An example of what such a confusion matrix might look like can be seen below:



Your code to generate the confusion matrix should be a part of your submission.

4. Other details

In replicating the method, there were some implementation details left unspecified by the authors. We give a list of these below:

- The output shapes of the feature maps of each stream after the pooling layers are 80x4 and 4x80 respectively. You may notice this requires shapes 80x80 before the max-pooling layers with kernel sizes 1x20 and 20x1 (i.e., the input and output size of the convolutional layer are the same). PyTorch supports matching the outputted feature map to the size of the input of a convolutional layer by using the argument *"padding='same' "*. **NOTE**: **You should not specify the padding size with an int tuple.**
- There is a missing layer within Figure 1 of the paper, which you will need to map the 200 units to the number of classes for the output.
- Following typical architecture design, and for consistency with the deep architecture/contents of the text, we also add a LeakyReLU with alpha=0.3 after the 200 unit fully connected layer, before dropout, which is not shown in Figure 1.
- Note that the position of Dropout in Figure 1 may cause confusion, the dropout is applied **AFTER** the 200 unit **FULLY CONNECTED LAYER** as they say in the text, not before/after the merge as they show in the figure.
- The paper specifies that they use L_1 regularisation with weight decay 0.0001. You **CANNOT** simply add *"weight_decay=0.0001"* to the optimizer, as this will implement L_2 regularisation, not L_1. As such, you will need to implement L_1 regularisation yourself.
  TIP: To obtain a tensor containing all the weights of the model, you can use:
  "weights = torch.cat([p.view(-1) for n, p in model.named_parameters() if ".weight" in n])"

- Weight initialisation and batch size are not specified in this paper. As such, we leave you to explore which initialisation schemes/batch sizes give you the best performance.
- Note, in this case, “validation” and “test” are used synonymously. In most cases outside of this coursework, there will be three splits: train/validation/test. It is worth noting the distinction between validation and test, where validation is a small subset of the training set where you have access to the ground truth, but you don’t train on it and rather use to it to tune/optimize hyperparameters of your model. Whereas the test set is used to assess the model’s ability to generalise to unseen data. You can think of the validation set almost as a “simulation” or “approximation” to the test set.

BC4 Notes

If you try running interactive jobs (i.e., using *“srun”*) you may be allocated a GPU that is being utilised by other processes (you can verify this by typing *“nvidia-smi”*, if there is a high GPU util on one of the GPUs on the node, it is likely being used by another process). This gives a high chance of your code producing “CUDA Error: Out of memory”. As such, it is recommended to run your code by submitting job scripts (i.e., using *“sbatch”*) rather than requesting interactive jobs, which will ensure your process is the only one allocated to that GPU node. Interactive jobs may still be useful to quickly test things such as code compilation, but if you get the CUDA error, it will likely be solved by using a job script. Our implementation was able to run and fully train the model using with following resources in the .sh script: “--time 0-00:30 --mem 16GB --gres gpu:1”. You may require more time/memory for extensions tasks.

5. Our Replicated Results

Replicating results from papers rarely produces the exact results as advertised. We have first created a PyTorch implementation of the paper using the data files and train/test split available to you. Our results can be found below:

Model	Raw	Epochs
Shallow	63.49 (±0.69)	100
Shallow	64.64 (±0.378)	200

As our re-implementation performs worse on the chosen splits, these are the values we want you to reproduce.

Note: the reported results are the averages of the raw accuracy on the final epoch for 5 runs (i.e., not necessarily the best epoch). We have included the margin of error to help you gauge what your average accuracy should roughly be.

6. Dataset and Helpful Code

To get you started and focused on training the method, we've prepared resources for this project which you can find here: https://uob-my.sharepoint.com/:f:/g/personal/jc17360_bristol_ac_uk/EuVWze7HrUxOhwGPFfxjvFoBRdE9K3mL5yEUoOgcLKs7hw.

The OneDrive directory includes a README which you can refer to understand what each file contains. The dataset contains a train split and a test split. All of your final results should be produced on the test split. **Do not train your model on data from the test split!**

Final Submission

Within your final submission you should submit the following:

1. Original code written in Python and PyTorch (other software/libraries will NOT be accepted) that replicates the published paper. You can use lab code from any or all group members. **We will run your code on Blue Crystal, so ensure that it compiles and runs**.
2. A report in the IEEE format (we recommend conference format) which can be found here: https://template-selector.ieee.org/secure/templateSelector/publicationType. The report should be no longer than **5 pages including references**. The report should include the following sections:
    a. **Title and Team Members**: (names and usernames) in addition to an agreement that all members gave an almost equal contribution with signatures (See Appendix A). Note that for single person groups only the name/username is required.
    b. **Introduction**: Definition of the problem addressed by Schindler et al. (in your own words!)
    c. **Related Work**: A summary of more recent published work (i.e., after Schindler et al. was published in 2016) attempting to address the same problem (up to 3 works).
    d. **Dataset**: A description of the dataset used, training/test split size, and labels.
    e. **CNN Architecture (Schindler et al.)**: Describe the architecture(s) that you have recreated and all of its details.
    f. **Implementation Details**: Provide a summary of the steps you have undertaken to replicate the results, train the data, and obtain the results. Do not include pieces of code, but you can use pseudo code if you find this helpful.
    g. **Replicating Quantitative Results**: You need to provide your version of Table 2 from the paper with the corresponding rows given above **AND** provide the confusion matrix.
    h. **Training Curves**: Provide the train/test loss curves and accuracy curves and comment on any over/underfitting you find within your training. These curves should be the same that you use in **Section G above** and can be directly gathered from tensorboard.
    i. **Qualitative Results**: This section should include sample cases where your method worked well and where it struggled based on your algorithm. You

can present examples showing the spectrogram along with the ground truth and predicted classes. We expect 1 good example where your prediction works and up to 2 examples where your prediction can be criticised.

j. **Improvements (if doing extension)**: In this section you should give information about up to two improvements that you have made to the method (see the mark scheme for details). You should not provide code when describing these improvements, but you may use pseudo code if this helps your explanation. Your choice of improvements should be justified both theoretically and experimentally.
k. **Conclusion and Future Work**: Summarise what your report contains in terms of content and achievements. Suggest future work that might extend, generalise, or improve the results in your report.

Marking Guideline

Note: code and report will be checked for plagiarism/academic misconduct. Proven plagiarism will result in a coursework grade of 0 for the whole team.

Up to 55%

To pass this assignment, you must produce original complete (compiles and runs on BC4 using batch mode command and PyTorch) code that replicates the results in the paper. You should produce a report with sections A-F correct and satisfactory. A partial attempt at including sections G-I, K is given. Replication results are within 5% of those given by us above on either the 100 epoch or 200 epoch model (including the margin of error).

Up to 60%

In addition to the above, sections G-I, K are complete and reflective of your understanding of the code and implementation. All sections are completed to an acceptable standard. Replication results are within 1% of those given by us above on either the 100 epoch or 200 epoch model (including the margin of error).

Up to 65%

In addition to the above, provide a single extension to the method that has been listed within Schindler et al. Section J includes results of this extension and discussion. Examples of extensions from within the paper include: Implementing the Deep CNN architecture, Data augmentation (either time stretching or pitch shifting), implementing BOTH file based maximum probability and majority vote accuracies, using stratified four-fold cross validation, evaluating your method on one of the other dataset.

Up to 75%

In addition to the above, provide a single extension to the method that **has not** been listed within Schindler et al. and Section J includes results of this extension and discussion. Note you must have at most 2 extensions. We will ignore all others when marking. Your extension should show at least a marginal improvement (i.e., be strictly greater than your base results without the extension). All sections of the report should be completed to a very good standard with good discussion of the results and method.

Up to 80%

In addition to the above, the report should be submittable to a B-class peer review conference or venue, i.e., it shows excellent understanding, correct and complete showcasing of the approach. Statements are concise, and any jargon cut out of implementation details. The chosen related work focuses on current state of the art for this problem. Extensive evidence of analysis, creativity, and originality in concise content presentation should be shown. Code is commented and could be easily understood and re-used by the reader.

Up to 100%

In addition to the above, the code and report are exemplary, and could be given as an example for an attempt to replicate this published work. Improvements to the results are beyond marginal (i.e., greater than 1% of baseline performance without improvement).

Universal Coursework Details

Deadline

The deadline for submission of all optional unit assignments is 13:00 on Thursday 8th of December (the University discourages Friday deadlines!). Students should submit all required materials to the "Assessment, submission and feedback" section of Blackboard - it is essential that this is done on the Blackboard page related to the "With Coursework" variant of the unit.

Time commitment

You are expected to work on both of your optional unit courseworks in the 3-week coursework period as if it were a working week in a regular job - that is 5 days a week for no more than 8 hours a day. The effort spent on the assignment for each unit should be approximately equal, being roughly equivalent to 1.5 working weeks each. It is up to you how you distribute your time and workload between the two units within those constraints.

You are strongly advised NOT to try and work excessive hours during the coursework period: this is more likely to make your health worse than to make your marks better. If you need further pastoral/mental health support, please talk to your personal tutor, a senior tutor, or the university wellbeing service.

Academic Offences

Academic offences (including submission of work that is not your own, falsification of data/evidence or the use of materials without appropriate referencing) are all taken very seriously by the University. Suspected offences will be dealt with in accordance with the University's policies and procedures. If an academic offence is suspected in your work, you will be asked to attend an interview with senior members of the school, where you will be given the opportunity to defend your work. The plagiarism panel are able to apply a range of penalties, depending on the severity of the offence. These include: requirement to resubmit work, capping of grades and the award of no mark for an element of assessment.

Extenuating circumstances
If the completion of your assignment has been significantly disrupted by serious health conditions, personal problems, periods of quarantine, or other similar issues, you may be able to apply for consideration of extenuating circumstances (in accordance with the normal university policy and processes). Students should apply for consideration of extenuating circumstances as soon as possible when the problem occurs, using the following online form:
https://www.bristol.ac.uk/request-extenuating-circumstances-form

You should note however that extensions are not possible for optional unit assignments. If your application for extenuating circumstances is successful, it is most likely that you will be required to retake the assessment of the unit at the next available opportunity.

Appendix A: Working in a Group

- **What if I can't find members for a group?** As it is optional to work within a group it is expected that you can complete the coursework by yourself if needs be. You can ask Michael or Tilo for help finding a group member, but we will not force groups together.
- **How will individual marks be assigned to each member in the group?** All members in the group will receive the same marks – the coursework is marked regardless of group size in the first place.
- **What if one or more of my team members doesn't engage with the coursework?** We expect each member of the group to contribute equally (though this could be in different areas – maybe one student is working on the report, another on the base model, another researching and implementing an extension). If this isn't the case, you should contact the unit director ASAP. By submitting the report, it is assumed that all group members are happy with the contribution of their team members with the signed agreement.
- **What if one or more of my team members becomes ill/has extenuating circumstances and has to resit?** In this case it is expected that the remaining members of the group will finish and complete the coursework on their own. Remember, it is optional to work as a group, so have a contingency plan to have to work with fewer members if needed.
- **What is the signed agreement?** You can include the following at the beginning of your report: "We agree that all members have contributed to this project (both code and report) in an approximately equal manner" with signatures of all group members below it.