Neural Networks Assignment

University of Sussex, Spring 2024, A2

Objective

Submit a research-style report, using the provided Jupyter Notebook, in which you build a neural network to investigate three small experiments and present/discuss the results.

How to submit

Please submit your assignment as a zip file that includes the .ipynb file containing your report, along with any additional image files that are used for figures in your report. Please name the zip file XXXXXX.zip, where XXXXXX is your candidate number. Give the Notebook the filename XXXXXX.ipynb. Submit through Canvas Online (not Turnitin).

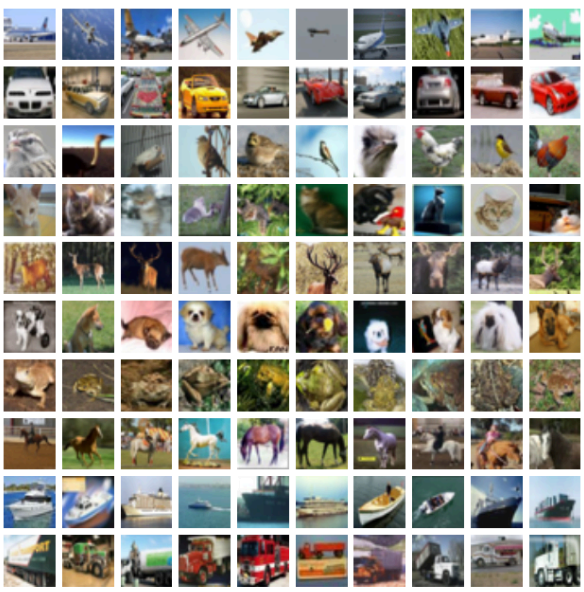
Grading

This assignment determines 100% of your grade for Neural Networks. The Notebook specifies how many marks you will be awarded for different parts of the report. There are 150 marks available in total, and your final grade will be given as a percentage. In general, marks will be awarded for clear descriptions of your model and results, well-reasoned, scientific explanations of the model choices and methods, clear presentation of the results, scientifically reasoned discussion of your results, and references to relevant academic publications that support your explanations/discussions.

Task Summary

The main task is to develop a deep convolutional neural network to perform multi-class classification. The dataset for this assignment is the CIFAR10 dataset (link), which is also available from the torchvision library in PyTorch (link). The dataset is made up of 32x32 colour images, which are split into a training set of 50,000 images and a test set of 10,000 images. The images have labels (one label per image) from 10 classes: 1) airplane; 2) automobile; 3) bird; 4) cat; 5) deer; 6) dog; 7) frog; 8) horse; 9) ship; 10) truck. The network should be trained to predict the labels using the training data, and its generalisation performance is evaluated using the test data. You will be asked to perform three experiments in which you analyse the influence of different methods on the performance and functionality of your neural network.

Task details



In this assignment, you need to implement a deep convolutional neural network, which you will later build upon. The baseline version of your CNN must have an input, three convolutional hidden layers, then two fully connected layers, where the second fully connected layer is 10-dimensional and provides the output classifications of your network. Each convolutional layer block must include pooling and a non-linear activation function of your choice. You can use PyTorch (recommended), or any other deep learning frameworks (e.g. TensorFlow) with automatic differentiation. The neural network should be trained to classify the images from the CIFAR10 dataset. You can use the modules available in packages like PyTorch to load the dataset (e.g. in PyTorch, you can use torchvision.datasets.CIFAR10) and to build the layers in your model. The assignment is split into three experiments:

Experiment 1	Choose an appropriate split of the training data to include a subset used for validation. Investigate the effect that the learning rate has on your model's performance, i.e. it's classification accuracy, or error rate. Compare the performance of your model for 5 different learning rates. The learning rates you choose must be high enough to allow the performance on the validation data to saturate, and they must be low enough to prevent your model from becoming unstable. For each learning rate, run the model at least 5 times, using different seeds to initialise any random number generators (e.g. for weight initialisation) such that each run of the model produces different results. You can then use the average of the model runs to plot the model's mean performance with respect to the training episodes for the training and validation data (these plots are also called learning curves). Using these results, design a learning rate scheduler that reduces the learning rate during training, again running the model at least 5 times to build up a picture of its average performance. Plot the learning curves for this new version of the model. Compare performance on the test data, and in terms of the generisability gap between training and validation data, between the learning rate scheduler model and the best performing model that didn't use a learning rate scheduler (out of the five you tested previously).
Experiment 2	Investigate the impact that regularisation has on the performance of your model. **First**, split the training data into two halves. One half will be used for training, the other half for validation. Then implement Dropout in the fully connected layers of your model. Investigate how the dropout rate affects average performance on the test data, and the generalisation gap between the training and validation data. Choose 5 Dropout rates, one of which must have a value of zero (i.e. no Dropout). Remember to run each version of the model 5 times to compute the mean performance. **Second**, swap around your two datasets, so that the training data is now used for validation, and the validation data is now used for training, and investigate how your model performs in a transfer learning task. To do this, freeze all of the weights in the convolutional layers (i.e. prevent them from being updated during optimisation), re-initialise the weights in the fully connected layers, and retrain the two fully connected layers. Apply only two models to the transfer learning task: one without Dropout, and the other using a non-zero Dropout rate that gave the best average performance. How does the average performance of these retrained models compare with the average performance of the models in which all layers were trained together? Refer to their average performance on the test data, as well as the generalisation gap between the training data and validation data. ***Provide learning curve plots to back up your analyses of these models.***
Experiment 3	Investigate the quality of gradient flow through your network. **First**, for the model that does not use Dropout, record gradients for the optimised parameters (i.e. dL/dw) in each layer of your model. Record the gradients over the first 5 episodes of training, and build a separate record of gradients over the final 5 episodes of training. Produce plots to show how the mean and standard deviation of gradients change as a function of layer number, for both the beginning (first 5 episodes) and the end (final 5 episodes) of training. **Second**, using the best performing, non-zero Dropout rate from Experiment 2, perform the same calculations to compute the mean and stand deviation of the gradients for all layers, and over the first and final 5 episodes of training. Does Dropout affect gradient flow in your model? If so, how are the gradients affected? **Third**, without including Dropout, add batch normalisation to each hidden layer of your model. Again, compute and plot the mean and standard deviation of the gradients as you had done before. How is gradient flow affected by batch normalisation in your model? **Fourth**, using learning curves for training and validation data, show how batch normalisation affects the performance of your model, referring to its average performance on the test data as well as to the generalisation gap between the average training and validation learning curves.

Report details

You will be guided through the process of writing a research report in the Notebook called XXXXXX.ipynb. All of your code and all of your written report must be produced within that one Notebook, using code cells for building the model, running the experiments, and plotting your results, and using Markdown cells for the written components of your report. The Markdown cells must not exceed a word limit of 2500 words, as determined by the Python script, `count_jupyter_nb_words.py`, that is provided with this assessment. Guidance for how to produce figures and how to use Latex markup within the Notebook is provided in the separate Notebook, `useful_code_for_figures_equatons.ipynb`.

Word count: An example of how to use `count_jupyter_nb_words.py` is provided as preamble comments within the file. The script outputs the number of words you have added to the document, as well as a short string starting from where the word count is exceeded so that you can locate where that happens, if it happens.

Your report will include the following components:

1. **Abstract/Introduction**
   Present a short summary of your project, including brief statements about the topic and the CIFAR10 classification task, your model, your results, and your conclusions. Reference relevant publications (academic papers or books). As a guide for what constitutes an academic publication: don't reference it if it doesn't appear in Google Scholar.

2. **Methodology**
   Describe any preprocessing of the input data, including how you split the training data (e.g. did you use a single train-validation split or did you do cross validation). Describe all the details of your modelling approach such that someone else would be able to replicate your model without referring to your code. The Notebook breaks this section down into several sections in which you are to describe:
   - the dataset, and any pre-processing you conducted.
   - neural network architecture, including a schematic.
   - the loss function.
   - the optimiser.
   - the methods specific to each experiment you have been tasked to perform.

   Make sure you state the values for any hyperparameters, e.g. layer sizes, filter sizes, nonlinear activation function details, optimiser hyperparameters etc. If multiple values were tested for certain hyperparameters, as you will need to do for two of the experiments, provide all of the values within the Methodology. Provide explanations for the methods you use, citing two or more academic publications where relevant. As a guide for what constitutes an academic publication: don't reference it if it doesn't appear in Google Scholar. You may wish to restate some methodological details in the Results section, e.g. to introduce the experiment, but those methods must also be described in the Methodology section in order to gain some of the available marks. **Remember: the methodology should include enough details for someone else to replicate your work.**

3. **Results**
   Briefly introduce and present the results of the three experiments. Use figures to plot your models' learning curves, i.e. the performance (accuracy or error rate) with respect to training episodes on training and validation datasets. Performance on the test data can be reported in the text or shown in the figure. Ensure that you have given your model sufficient training episodes/epochs for the performance to plateau. If your performance curves for validation data are still increasing/decreasing at the end of training, you haven't trained it for long enough. Remember to label figure axes, and to provide a figure caption that explains everything the reader needs to

know to understand the figure (note, this does not mean interpreting the results in the figure caption: you only need to provide a detailed description of what the figure is showing). **Make sure that you refer to figures (e.g. by calling it Figure 1) within your report text in Markdown cells, and that those figures (e.g. Figure 1) is generated as the output of a code cell.** For each set of results, the text must include a brief description of what you were investigating, and a summary of the results (for example: "performance increased monotonically with the learning rate", or "method A produced better performance than method B, which is consistent with the role of method A in….. [<citation>]".

4. **Conclusions and Discussion**
   Use this section to summarise the results and to draw conclusions. Discuss possible, scientifically reasoned explanations for the results you obtained, particularly if your results were different from what you expected. Describe two modifications you could make to your model to improve its performance on the test data. Explain what effect each modification might have on the results from your Experiments, e.g. how might those modifications affect which learning rate produces the best performance? Don't be afraid to point out flaws (if any) in the choices you have made for your model, and discuss alternatives that may produce better/more meaningful results, in hindsight. Make reference to two or more relevant publications. As a guide for what constitutes an academic publication: don't reference it if it doesn't appear in Google Scholar.

5. **References**
   List references in the format suggested. Any references to articles on popular science websites, including medium.com, towardsdatascience.com, geeksforgeeks.org etc. etc. will not be awarded marks as they do not constitute academic literature. If it doesn't appear in a Google Scholar search, then don't reference it.

6. **Code**
   All code to build, run, and analyse your model, as well as to plot results, must be provided in the code cells provided. If, for the sake of organisation, you need to add more code cells, add them immediately after the cells provided. **Do not add any code cells out of order and mixed between other Markdown/text cells, unless you are using Google Colab and you need to insert an image for a figure**. This one exception is allowed to make it easier for you to prepare your file for submission (see the guidance in `useful_code_for_figures_equatons.ipynb` for more information). Important: the person marking your report will not run any of your code. Therefore, all figures must be generated and visible in the Notebook when you submit to Canvas, or they will not gain marks. In order for those marking your report to determine whether or not you have correctly completed the Experiments, they may need to look at your code. To help us make sense of your code, and therefore to help us improve our feedback to you, please **provide plenty of comprehensible comments**, including general overviews for each section of code, and shorter comments every few lines of code (if not on every line, if necessary).

Important points:

- You are not expected to produce the perfect model with competitive performance. What we are looking for is your scientific scrutiny of the model and the techniques you have employed. The only way you can score high marks is to include explanations and/or rational justification for your methods, results, and conclusions.
- Your work will be assessed by humans who get tired/hungry/frustrated/intrigued/curious. While we make every effort to assess your work fairly, those humans are not perfect. Help them to assess your work, and to realise how talented you really are, by making your presentation clear and concise. Check that references to figures are consistent, check for spelling, grammar, and consistent terminology. Every little helps.
- Use of Large Language Models: we have given this project description (and paraphrased versions) to different Large Language Models (e.g. chatGPT, Gemini, etc.) and generated numerous responses from it. We therefore have a good idea of the kind of content it will produce for this assignment. Be very cautious about what these tools provide. **These are generative models**, which means that **they have been trained to give responses that appear plausible, and have not been trained to give factually correct responses**. It is not difficult to get a LLM to

generate factually incorrect responses about technical details.

Preparing your submission

Provide all necessary files in a single .zip file called XXXXXX.zip, where XXXXXX is your candidate number. The zip file should not have any directories within it. The zip file needs to include the Notebook, which should have the filename XXXXXX.ipynb, and any image files that are used within the Notebook. Make sure that those image files are correctly referenced in your Notebook. If you have used Google Colab and have followed the guidance (provided in `useful_code_for_figures_equatons.ipynb`) for including images in figures, then be sure to remove any occurrences of "/content/" in the file path.

Marking criteria

The marks that are available have been stated for each section of your report. For example, describing specific architectural details will gain marks; you will gain marks for providing scientifically reasoned explanations, e.g. for describing advantages/disadvantages for particular model choices. The word limit is 2500 words. **Any part of your report that exceeds that limit will not be marked**. Any code or text provided in additional files will not be marked. Any figures that do not appear in the Notebook cannot be marked. As a more general guide, the percentages below are applied to the available marks in each section to determine your actual mark. For example, if 10 marks are available for one section, and the quality of your work is deemed to be 80%, then the number of marks you achieve will be calculated by multiplying the marks available (10) by the quality of your work, 80%, meaning that you will obtain 8 marks for that section.

- **70%-100% - excellent:**
  - Shows very good understanding, supported by evidence that the student has extrapolated from what was taught, through extra study or creative thought. Work at the top end of this range is of exceptional quality. Report will be excellently structured, with proper references and proper discussion of existing relevant work. The report will be neatly presented, interesting and clear, with a disinterested critique of what is good and bad about the approach taken, and thoughts about where to go next with such work.
- **60-69% - good:**
  - The work will be very competent in all respects. Work will evidence substantially correct and complete knowledge, though will not go beyond what was taught. Report should be well-structured and presented with proper referencing and some discussion/critical evaluation. Presentation will generally be of a high standard, with clear written style and some discussion of related work.
- **50-59% - satisfactory:**
  - The work will be competent in most respects. There may be minor gaps in knowledge, but the work will show a reasonable understanding of fundamental concepts. The report will be generally well-structured and presented with references, though may lack depth, appropriate critical discussion, or discussion of further developments, etc.
- **40-49% - borderline:**
  - The work will have some significant gaps in knowledge but will show some understanding of fundamental concepts. The report should cover the fundamentals but may not cover some aspects of the work in sufficient detail. The work may not be organised in the most logical way and its presentation may not always be appropriate. There will be little or no critical evaluation or discussion. References may be missing, etc.
- **30-39% - fail:**
  - The work will show inadequate knowledge of the subject. The work is seriously flawed, displaying a major lack of understanding, irrelevance, or incoherence. The report is badly organised and incomplete, possibly containing irrelevant material. The report may have missing sections, no discussion, etc.
- **Below 30% - unacceptable (or not submitted):**
  - The work is either not submitted or, if submitted, so seriously flawed that it does not constitute a bona-fide report/script.