Assignment 2: Image Classification

Key information

Deadlines

Submission: Sunday week 11 (26 Oct), 11.59pm

Late submissions policy

Late submissions are allowed for up to 3 days late. A penalty of 5% per day late will apply. Assignments more than 3 days late will not be accepted (i.e. will get 0 marks). The day cut-off time is 11:59pm.

Marking

This assignment is worth 25 marks = 25% of your final mark. It consists of two components: code (10 marks) and report (15 marks). A marking rubric on 100-point scale for this assignment is included at the end of this document.

The assignment can be completed in groups of 4 or 5 students. See the submission details section for more information about how to submit.

Submission

Three files are required to be submitted in the relevant submission portals on Canvas:

- Your report as a .pdf file
- Your Jupyter notebook as a .ipynb file

Name your files with the following format:

- Report:
  - A2-report-SID1-SID2-SID3-SID4.pdf
- Code:
  - a2-code-SID1-SID2-SID3-SID4.ipynb

where SIDx are the SIDs of the four students in your group.

Before you submit, you need to create a group in Canvas. Under the "People" page on Canvas, select the "A2 groups" tab. You and your group partners should choose one of the empty groups listed under this tab, and join it.

Important: The "A2 groups" registration is for Assignment 2. You need to register a group under "A2 groups" even if you work with the same partner as in Assignment 1. Please be careful as otherwise your mark may not be recorded correctly. There may also be a mark deduction for not following the instructions.

Then you need to submit your assignment on behalf of the group in the corresponding submission box. Only one student from the group needs to submit, not all.

Code information

The code for this assignment should be written in Python in the Jupyter Notebook environment. **Please follow the structure in the template notebook provided as part of the assignment materials**. Your implementation of the algorithms should predominantly utilise the same suite of libraries we have introduced in the tutorials (Keras, scikit-learn, numpy, pandas etc.). Other libraries may be utilised for minor functionality such as plotting, however please specify any dependencies at the beginning of your code submission. While most of your explanation and justification can be included in the report, please ensure your code is well formatted, and that there are sufficient comments or text included in the notebook to explain the cells.

You may choose to run your code locally or on a cloud service (such as Google Colaboratory), however your final submission needs to be able to run on a local machine. Please submit your notebook with the cell output preserved and ensure that all the results presented in your report are demonstrated in your submitted notebook. Your code may also be rerun by your marker, so please ensure there are no errors in your submitted code and it can be run in order.

Task

In this assignment, you will implement and compare several machine learning algorithms, including a Multilayer Perceptron (MLP), Convolutional Neural Network (CNN) and an appropriate algorithm of your choice from the first 6 weeks, on an image classification task. You will need to demonstrate your understanding of a full machine learning pipeline, including data exploration and pre-processing, model design, hyperparameter tuning, and interpreting results. Moreover, the assignment will require you to consolidate your knowledge from the course so far to effectively discuss the important differences between the algorithms.

While better performance is desirable, it is not the main objective of this assignment. Since chasing the best possible performance may require large models and specialised GPU hardware, you should focus on thoroughly justifying your decisions and analysing your results. **Please see the marking criteria at the end of the specification for how you will be assessed. There are no marks allocated for reaching a particular performance or having highly complex models.**

1. Code

As mentioned above, your code submission for the assignment should be an .ipynb Jupyter notebook in similar fashion to the tutorials, including well commented code and sufficient text to explain the cells. Please follow the structure provided in the template notebook available as part of the assignment materials.

Data loading, pre-processing, and exploration

The dataset you will use in this assignment is CIFAR-10. It consists of 60000 32x32 colour images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images. More information on the dataset can be found in the associated paper here: https://www.cs.toronto.edu/~kriz/cifar.html. **Please refer to the provided dataset on Canvas rather than downloading from the CIFAR site.** Code to load these files is provided in the template notebook.

The images in this dataset have relatively low dimensionality to aid keeping your runtimes short. You can increase/decrease the dimensionality of the data or make use of a subset of the training data as required, with justification in the report.

To better understand the properties of the data and the preprocessing that may be appropriate, you should begin with some exploration of the data. You may like to explore what the different classes are, the number of examples from each class, and consider characteristics of the images such as whether they are centred, size of different features in the images, pixel intensities across different images, key differences between classes etc. Explore if there are factors that might make the task more difficult, such as classes with similar features. You should include anything you feel is relevant in this section.

Based on your insight from the data exploration and/or with reference to other sources, you should apply appropriate preprocessing techniques. Your choice of preprocessing needs to be justified in the report. You may apply different preprocessing techniques for the different algorithms, with justification in the report.

Consider if you need to make any additional splits of the data and think carefully about how each part of the data should be utilised to evaluate hyperparameter combinations and compare the performance of the different models.

Algorithm design and setup

You will be required to design and implement three algorithms that we have covered in the course using the **sklearn** and/or **keras** libraries, in order to investigate their strengths and weaknesses.

1. An appropriate algorithm of your choice from the first 6 weeks of the course
2. A feedforward multilayer perceptron neural network (MLP)
3. A convolutional neural network (CNN)

In this section, implement an instance of each model before tuning hyperparameters, and set up any functions you may require to tune hyperparameters in the next section.

Due to runtime constraints, it is not feasible to consider every possible neural network architecture when designing your models, however you should justify your design decisions in the report, and you may wish to conduct a few rough experiments to converge on a reasonable design. Remembering that the focus of the assignment is to demonstrate your understanding of and compare the algorithms, not to achieve state-of-the-art performance, **keep your models appropriately small such that you can complete the hyperparameter tuning in a reasonable time on your hardware of choice**.

Although you may like to reference external sources when designing your algorithms, **you must implement your neural network models yourself**, rather than import prebuilt models from Keras (such as those available in keras.applications).

Hyperparameter tuning

Perform a search over relevant hyperparameters for each algorithm using an appropriate search strategy of your choice (eg. cross validation, validation set grid search, Bayesian search, etc.). You may use scikit-learn and/or keras-tuner to perform your searches, and you are suggested to look through the documentation of these packages to see the available utilities and decide on your implementation. Remember there are many factors that can affect the runtime of your search, such as the number of epochs, optimisers, and the particular hyperparameters included in the search. Please ensure you search over the following:

- Algorithm of your choice from the first 6 weeks:
    - You will need to choose an appropriate set of hyperparameters to search over based on what was covered in the tutorials or other references
- Multilayer feedforward and convolutional neural networks:
    - Tune over at least 3 different hyperparameters

You will need to justify your search method, choices of hyperparameters to search over, and the values included in the search as part of your report (see below).

Keep a record of the runtimes and results with each hyperparameter combination (you may need to consult documentation to see how to extract this information) and use these to produce appropriate visualisations/tables of the trends in your hyperparameter search to aid the discussion in your report.

Please preserve the output of these cells in your submission and keep these hyperparameter search cells independent from the other cells of your notebook to avoid needing to rerun them, as markers will not be able to run this code (it will take too long), i.e. **ensure the later cells can be run if these grid search cells are skipped.**

Final models

After selecting the best set of hyperparameters for each model, include cells which train the models with the selected hyperparameters **independently of the parameter search cell**, and use these implementations to compare the performance (and other relevant properties) of the different models using the test set.

2. Report

An approximate outline of the report requirements is provided below, but make sure to reference the marking criteria also for the mark distribution. You may write your report in MS Word or LaTeX, but please ensure that it is well-formatted and submit it as a pdf.

Please stick to the structure (headings and subheadings) outlined below, as these align with the marking criteria.

Please do not include screenshots of your code in the report. The report should not be focussed on your code and describing exactly how you have implemented things, but rather on the content outlined below.

Introduction

State the aim of your study and outline the importance of your study. You can consider both the importance of this dataset itself, but also the importance of comparing algorithms and their suitability for the task more generally.

Data

In this section, you should describe the dataset and pre-processing.

Data description and exploration. Describe the data, including all its important characteristics, such as the number of samples, classes, dimensions, and the original source of the images. Discuss your data exploration, including characteristics/difficulties as described in the relevant section above and anything you consider relevant. Where relevant, you may wish to include some sample images to aid this discussion. For information about the original dataset, such as the class names, the CIFAR site (https://www.cs.toronto.edu/~kriz/cifar.html) may be useful.

Pre-processing. Justify your choice of pre-processing either through your insights from the data exploration or with reference to other sources. Explain how the preprocessing techniques work, their effect/purpose, and any choices in their application. If you have not performed pre-processing or have intentionally omitted possible preprocessing techniques after consideration, justify these decisions.

Methods

In this section, you should explain the classification methods you used.

Theory. For each algorithm, explain the main theoretical ideas (this will be useful as a framework for comparing them in the rest of the report). Explain why you chose your particular algorithm from the first six weeks.

Strengths and weaknesses. Describe the relative strengths and weaknesses of the algorithms from a theory perspective. Consider factors such as performance, overfitting, runtime, number of params and interpretability. Explain the reasons; e.g. don't simply state that CNNs perform better on images, but explain why this is the case.

Architecture and hyperparameters – State and explain the chosen architectures or other relevant design choices you made in your implementation (e.g. this is the place to discuss your particular neural network configurations)**.** Describe the hyperparameters you will tune over, the values included in the search, and outline your search method. Briefly explain what each hyperparameter controls and the expected effect on the algorithm. For example, consider the effects of changing the learning rate, or changing the stride of a convolutional layer. Justify why you have made these choices about the search method, and hyperparameters included.

Results and discussion

In this section, you should present and discuss your results. **Please do not include screenshots of raw code outputs when presenting your results. Instead tabulate/plot any results in a manner more appropriate for presentation in the report.**

Begin with the hyperparameter tuning results. Include appropriate tables or graphs to illustrate the trends (performance, runtime etc.) across different hyperparameter values. Discuss the trends and provide possible explanations for their observation. Consider if the results aligned with your predictions.

Next, present a comparison of the models you have implemented for this task (with the best hyperparameters found). Include a table showing the best hyperparameter combination for each algorithm, the performance on the test set (e.g. accuracy and other performance measures), and the runtime(s). Analyse and discuss the results, referring back to the theoretical properties and strengths/weaknesses of the classifiers discussed in the Methods section. Consider if the results aligned with your expectations. What factors influenced the runtime (time per epoch, total number of epochs required etc.)?

Include anything you consider interesting and/or relevant. For example, you may like to comment on the types of mistakes particular models made via confusion matrices etc.

Conclusion

Summarise your main findings, mention any limitations, and suggest future work. When making your conclusions, consider not only accuracy, but also factors such as runtime and interpretability. Ensure your future work suggestions are concrete (e.g. not in the spirit of "try more algorithms") and justify why they would be appropriate.

Reflection

Write one or two paragraphs describing the most important thing that you have learned while completing the assignment. Every student should write their own reflection.

References

Include references to any sources you have utilised in completing the code and/or report. You may choose any appropriate academic referencing style, such as IEEE.

Academic honesty – important

Please read the University policy on Academic Honesty very carefully:
https://sydney.edu.au/students/academic-integrity.html

Plagiarism (copying from another student, website or other sources), making your work available to another student to copy, engaging another person to complete the assignments instead of you (for payment or not) are all examples of academic dishonesty. Note that when there is copying between students, both students are penalised – the student who copies and the student who makes their work available for copying.

Please do not confuse legitimate co-operation and cheating. You can discuss the assignment with other students, this is a legitimate collaboration, but you cannot complete the assignment together unless you are from the same group. Every group must write their own code and report.

The University penalties are severe and include: 1) a permanent record of academic dishonesty in the university database, 2) mark deduction, ranging from 0 for the assignment to Fail for the course and 3) in severe cases, suspension from the University and cancelling of your student visa.

The University penalties are severe and include: 1) a permanent record of academic dishonesty on your student file, 2) mark deduction, ranging from 0 for the assignment to Fail for the course and 3) expulsion from the University and cancelling of your student visa.

If there is a suspected case, the investigation may take several months. Your mark will not be finalised until the investigation is completed. This may create problems enrolling in other courses next semester for which this course is a pre-requisite or delaying your graduation. Going through the investigation is also very stressful.

We will use similarity detection software to compare your assignments with other sources and with each other. If you cheat, the chances that you will be caught are very high.

Be smart and don’t risk your reputation by engaging in plagiarism and academic dishonesty.

Marking criteria

1. Code – 40 marks of the assignment mark (10% of your final mark)

Requirement	Mark			
1. Implements appropriate preprocessing techniques	Preprocessing does not function or has significant implementation issues **[0]**	Preprocessing code runs but has moderate implementation issues **[1]**	Good; preprocessing techniques implemented with minor issues **[2]**	Excellent; preprocessing performed appropriately with no implementation issues **[3]**
2. Implements appropriate algorithm of choice from first 6 weeks of course	No functioning algorithm/significant issues with implementation **[0]**	Algorithm functions, but is not appropriate or has serious issues with design or implementation **[2]**	Good; algorithm functions and is appropriate. Minor issues with implementation or other issues **[3]**	Excellent; algorithm is appropriate, and no issues with implementation **[4]**
3. Sets up and implements MLP	No functioning algorithm/major issues with implementation **[0]**	Algorithm somewhat functions, but has serious issues with design or implementation **[2]**	Good; algorithm functions well. Minor issues with Implementation or other issues **[3]**	Excellent; algorithm is appropriate and there are no issues with implementation **[4]**
4. Sets up and implements CNN	No functioning algorithm/major issues with implementation **[0]**	Algorithm somewhat functions, but has serious issues with design or implementation **[2]**	Good; algorithm functions well. Minor issues with implementation or other issues **[3]**	Excellent; algorithm is appropriate and there are no issues with implementation **[4]**
5. Algorithm of choice – hyperparameter search	No functioning hyperparameter search or completely irrelevant hyperparameters **[0]**	Major issues with search method, hyperparameters, or hyperparameter values **[2]**	Good; minor issues with search method, hyperparameters, or values. Tunes over key hyperparameters for chosen algorithm. **[4]**	Excellent; well implemented search method, hyperparameters and hyperparameter values **[5]**
6. MLP - hyperparameter search	No functioning hyperparameter search or completely irrelevant parameters **[0]**	Major issues with search method, missing hyperparameters, or hyperparameter values **[2]**	Good; minor issues with search method, hyperparameters, or hyperparameter values. Tunes over at least 3 hyperparameters appropriately. **[4]**	Excellent; well chosen search method, sufficient and appropriate hyperparameters and values included in search **[5]**

7. CNN - hyperparameter search	No functioning hyperparameter search or completely irrelevant parameters **[0]**	Major issues with search method, missing hyperparameters, or hyperparameter values **[2]**	Good; minor issues with search method, hyperparameters, or hyperparameter values. Tunes over at least 3 hyperparameters appropriately. **[4]**	Excellent; well chosen search method, sufficient and appropriate hyperparameters and values included in search **[5]**
8. Best hyperparameter combination of each model trained and evaluated on test set in separate cell	Not completed, or significant issues **[0]**		Completed with no/minimal issues **[2]**	
9. Code quality	Very poor code quality, e.g. some code does not run, no comments, no markdown text, very poor variable names **[0]**	Poor code quality, e.g. poor comments or not enough markdown text to easily read the notebook, poor variable names **[4]**	Good code quality; minor issues with one aspect such as comments, not enough markdown, or variable naming **[6]**	Excellent, readable code and overall notebook **[8]**

2. Report – 60 marks of the assignment mark (15% of your final mark)

Requirement	**Mark**			
	Introduction – 4 mark			
1.Aim	Aim is not discussed or very poor **[0]**	Good; minor issues, such as not highlighting all aspects of study (comparison, hyperparameter tuning etc.) **[1]**	Excellent; aim of study is well discussed with no issues **[2]**	
2.Importance	Importance not discussed or very poor **[0]**	Good; importance partially discussed. Missing some aspects, such as importance of comparing classifiers or other issues **[1]**	Excellent; importance well justified and related to practical use **[2]**	
	Data – 4 mark			
3. Data description and exploration	Dataset not described or very poor **[0]**	Limited dataset description with missing information, and/or no data exploration **[1]**	Minor issues with dataset description and/or exploration **[1.5]**	Thorough data description and exploration, including discussion of important features and challenges as mentioned in the assignment specification, with sample images where relevant. **[2]**

4. Pre-processing description and justification	Pre-processing not mentioned or very poor **[0]**	Pre-processing mentioned but not described well and/or missing/poor justification **[1]**	Minor issues with either preprocessing description or justification **[1.5]**	Excellent description of preprocessing techniques (if any) and their effect/purpose. Techniques used are justified from lectures, labs, or other sources. Brief discussion of which pre-processing techniques were considered but not necessary. **[2]**
Methods – 22 marks				
5. Algorithm of choice – description and justification for inclusion	Both description and justification missing or very poor **[0]**	Major issues description or justification, including poorly chosen algorithm **[2]**	Good description and justification of inclusion, with minor issues or missing detail **[3]**	Excellent description with sufficient detail to explain the advantages and disadvantages of the algorithms later. Appropriate justification of inclusion is provided. References included where appropriate. **[4]**
6. MLP – description	Description missing or very poor **[0]**	Major issues or omissions **[2]**	Good, with minor issues or missing detail. **[3]**	Excellent description, with sufficient detail to explain the advantages and disadvantages of the algorithm later. References included where necessary. **[4]**
7. CNN - description	Description missing or very poor **[0]**	Major issues or omissions **[2]**	Good, with minor issues or missing detail. **[3]**	Excellent description, with sufficient detail to explain the advantages and disadvantages of the algorithm later. Includes references where necessary. **[4]**
8. Comparison of strengths and weaknesses	Not included or very poor **[0]**	Major issues or omissions **[2]**	Good; minor issues including some relevant points of comparison missed. **[4]**	Excellent comparison of the relative strengths and weaknesses of the classifiers from a theory perspective, and considering this particular dataset in the comparison. Sources cited where appropriate. **[5]**

9. Architecture and hyperparameter tuning description	Not included or very poor **[0]**	Major issues or omissions in description **[2]**	Good; architecture choices and search method are well described/justified and hyperparameters chosen to search over explained. Minor issues or lacking detail. **[4]**	Excellent description and explanation/justification of architecture, design choices, search method, and chosen hyperparameters and values. **[5]**
Results and discussion – 22 marks				
10. Hyperparameter tuning results presentation	No figures/tables, or only screenshots of code output. **[0]**	Figures or tables have major issues or omissions **[2]**	Good; figures or tables are appropriate and show trends/results from hyperparameter tuning. Minor issues with the presentation **[3]**	Excellent presentation of hyperparameter tuning results in appropriate figures or tables, with no presentation issues. If there are any relevant differences in runtime, these are presented. **[4]**
11. Hyperparameter tuning discussion	Not included or very poor **[0]**	Discussion has major issues or omissions **[2]**	Most important hyperparameter tuning results/trends are discussed. Includes comment on how the results aligned with predictions. Minor issues and/or lack of detail. **[3]**	Excellent discussion of hyperparameter results/trends, including possible explanations or reflections on how the results aligned with predictions. **[4]**
12. Results table	Not included or very poor, including screenshots of code output **[0]**	Major issues with formatting or omission of multiple results **[2]**	Minor issues with formatting or omission of one result **[3]**	Excellent table with all required results and appropriate formatting **[4]**
13. Results discussion/analysis	Not included or very poor **[0]**	Discussion has major issues or omissions **[2]**	Most important trends in the results discussed, and compared to expectations based on theoretical properties. Minor omissions and/or lack of detail. **[4]**	Excellent analysis of the trends in the results, with comparison to expectation based on theoretical properties. Differences in runtime are discussed and justified. Possible exploration of further trends beyond the tabulated results (e.g. differences by class accuracy, types of mistakes made, precision vs. recall etc.) **[6]**
Conclusion and future work – 1 mark				

14. Summary of main findings and identification of study limitations	Not included or very poor **[0]**	Major omissions or issues in summary and/or limitations **[1]**	Minor issues with summary (e.g. does not consider runtime) or misses some relevant limitation(s) **[1.5]**	Excellent summary which considers factors such as runtime and practicality of the algorithms for this particular task. Limitations identified are relevant. **[2]**
15. Future work suggestions	Not included or very poor **[0]**	Suggestions are not specific enough or do not address study limitations **[1]**	Minor issues with suggestions **[1.5]**	Suggestions are concrete and directly address the study limitations **[2]**
Reflection – 2 marks				
16. Reflection on most important thing learnt by each group member	Not included or very poor **[0]**	Reflection is lacking in depth or detail or is missing for some group members **[1]**	Excellent, relevant reflection with sufficient depth for all group members **[2]**	
Presentation – 6 marks				
17. Report formatting and structure	Serious issues with formatting or structure that make the report difficult to read **[0]**	Unclear structure or formatting issues, but report is still readable **[1]**	Minor issues with structure or formatting **[2]**	No issues with report structure or formatting. Sections are clearly delineated and formatting is clean and legible. Code snippets are not included inappropriately in the report. **[3]**
18. Language	Serious spelling or grammatical issues in all aspects of report that make the report difficult to read **[0]**	Many minor spelling or grammar mistakes that hinder the overall readability of the report, and/or non-academic language in many sections **[1]**	Several minor spelling or grammar mistakes that do not hinder the overall readability of the report, and/or non-academic language in some sections **[2]**	Very few minor spelling or grammar mistakes. Language is academic in style with clear sentences. **[3]**

Dataset attribution

Krizhevsky, A., & Hinton, G. (2009). Learning multiple layers of features from tiny images.