Assignment 2: Image Classification

Key information

Deadlines

Submission: 11:59pm, 21 October, 2022 (Friday week 11, Sydney time)

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 guide for both the code and report is included at the end of this document.

The assignment can be completed individually or in groups of 2 students. No more than 2 students are allowed. See the submission details section for more information about how to submit.

Submission

You will submit three files: report as .pdf file, and your code in two versions: .ipynb and .pdf. There are three different submission boxes in Canvas.

The code for this assignment should be written in **Python** in the Jupyter Notebook environment. A Jupyter Notebook template with the data loading code is provided. Your core implementation of the algorithms and hyperparameter searching should use the same suite of libraries that we have used in the tutorials, such as **keras**, **sklearn**, **numpy**, and **pandas**. Other libraries may be utilised for minor functionality such as plotting, however please specify any dependencies at the beginning of your code submission.

To generate a .pdf of your notebook for submission, please use File> Download as > PDF or Print Preview -> Save as PDF. Remember to submit both the .ipynb and the .pdf of your code.

Before you submit, if you work in a group, you will need to create a group in Canvas. Under the "People" page on Canvas, select the "A2 Group" tab. You and your group partner should choose one of the empty groups listed under this tab, and both join it. Groups have a maximum of 2 members. **If you are completing the assignment individually, you don't need to create a group.**

The submission file should contain the SID number(s) and should be named like this:

Code:

- a2-SID.ipynb (.pdf) for a student working individually, where SID is the student's SID number
- a2-SID1-SID2.ipynb (.pdf) for a group of 2 students, where SID1 and SID2 are the SIDs of the two students

Report:

- a2-report-SID.pdf (individual) or a2-report-SID1-SID2.pdf (group)

Task

In this assignment, you will implement and compare several machine learning algorithms, including a Multi-layer Perceptron (MLP) and Convolutional Neural Network (CNN), 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.

Since chasing the best possible performance may require large models and specialised GPU hardware, the main objective of the assignment is not to achieve the highest accuracy possible, but rather to thoroughly justify your decisions and analyse your results.

1. Code

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.

Data loading, pre-processing, and exploration

The dataset for this assignment is Fashion-MNIST, so named because the images are in the same 28x28 greyscale format as the original MNIST dataset. Each image depicts an item of clothing or fashion accessory, with ten classes in total. Despite each image having the same format as MNIST, the task is significantly more difficult.

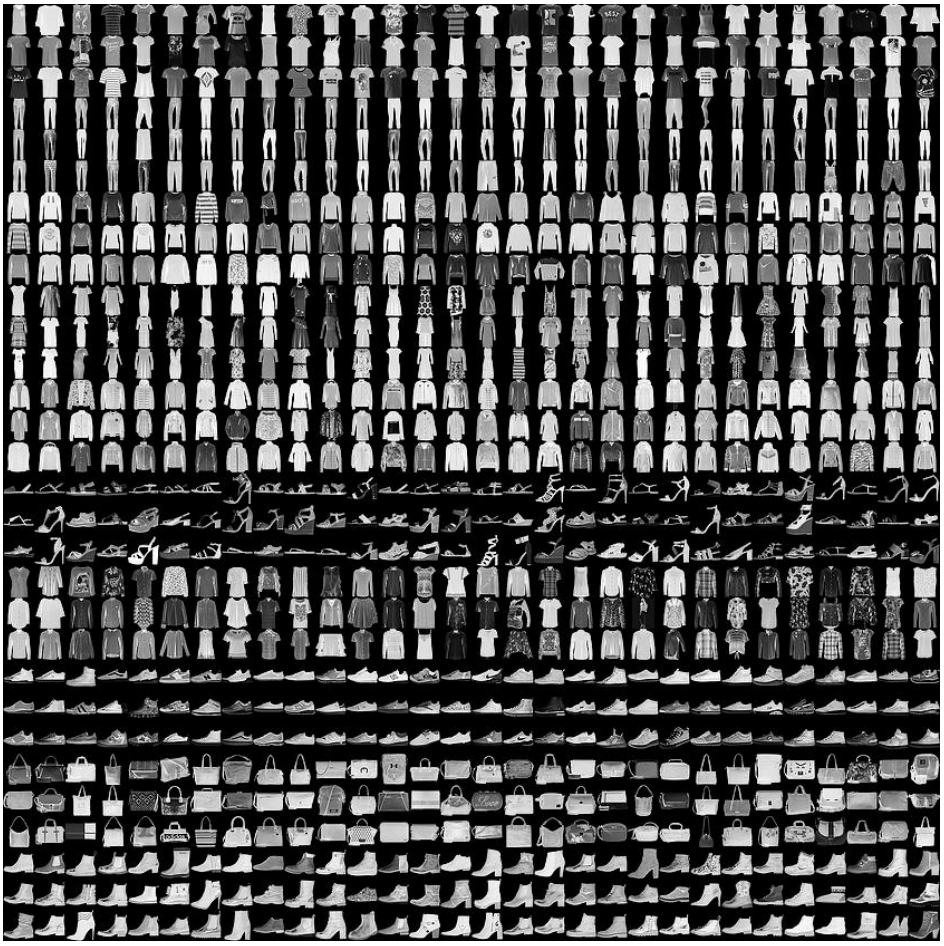

Figure 1: Samples from the Fashion-MNIST dataset

The code to load the training and test sets is provided in the .ipynb template. For further information on the dataset, you may be interested in the following link: https://github.com/zalandoresearch/fashion-mnist

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 fully connected neural network
3. A convolutional neural network

Due to runtime constraints, you will not be required to thoroughly tune over all possible neural network designs, 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**.

You will need to choose an appropriate number of epochs to use for your search, or use a stopping criterion.

Hyperparameter tuning

You may use cross validation via **sklearn** to search over different parameter combinations, or you may implement another reasonable search approach if you find the runtime of CV prohibitive (for example, a holdout procedure with the validation set). Please ensure you search over the following:

1. 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
2. Fully connected neural network:
    - At least three learning rate values
    - At least two appropriate activation functions for the hidden layers, as well as the case where **no** activation function is used
3. Convolutional neural network:
    - At least three learning rate values
    - At least two kernel sizes
    - At least two strides

Keep a record of the runtimes and results with each parameter combination (if using **sklearn**, consult the documentation) and use these to produce appropriate visualisations/tables of the trends in your hyperparameter search to aid the discussion in your report.

In your submission, keep the parameter search separate from other cells, and submit the .ipynb with the output preserved, as markers will not be able to run this code (it will take too long).

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 measure the performance of each model on 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.

Introduction

State the **aim** of your study. Outline the **importance** of your study. Even though this particular dataset may not be the most practically useful, consider the importance of comparing algorithms and their suitability for a given task, as well as the importance of image classification more broadly.

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 potential difficulties for classification such as classes with similar features and differences in pixel intensities between samples. Where relevant, you may wish to include some sample images to aid this discussion.

Pre-processing. Justify your choice of pre-processing either through your insights from the data exploration or with reference to other sources. Explain the effect/purpose of any pre-processing techniques you have applied. If you have not performed pre-processing or have intentionally omitted possible steps such as image centring, justify these decisions.

Methods

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

Summary. Briefly describe the main ideas of the three algorithms - your chosen algorithm, MLP and CNN. Explain why you chose your algorithm.

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**,** the hyperparameters you aim to tune and outline your search method. Briefly explain what each hyperparameter controls and how changing it may affect the algorithm. For example, consider the effects of changing the learning rate, or changing the stride of a convolutional layer.

Results and discussion

In this section, you should present and discuss your results.

Begin with the hyperparameter tuning results. Include appropriate tables or graphs (not code output) to illustrate the trends (performance, runtime etc.) across different hyperparameter values. Discuss the trends and provide possible explanations for their observation. Did they align with your predictions?

Next, present a table showing the best hyperparameter combination for each algorithm, its performance on the test set (e.g. accuracy and other performance measures), and the training runtime. Analyse and discuss the results. Refer to the stated strengths and weaknesses of the classifiers in the Methods section; did the results agree 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 look at which classes each algorithm confused.

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 (eg. 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.

Academic honesty – very 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 his/her work available for copying

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 takes several months**. Your mark will not be finalised until the investigation is completed. This may create **problems enrolling in other courses next semester** (COMP5318 is a pre-requisite for many courses) or **delaying your graduation**. Going through the investigation is also **very stressful**.

In addition, the Australian Government passed a new legislation last year (Prohibiting Academic Cheating Services Bill) that makes it a **criminal offence** to provide or advertise academic cheating services - the provision or undertaking of work for students which forms a substantial part of a student's assessment task.

Do not confuse legitimate co-operation and cheating! You can discuss the assignment with other students, but you (if you work individually) or your group (if you work in pairs) must write your own code.

To detect code similarity in this assignment, we will use TurnItIn and MOSS which are **extremely good**. If you cheat, the chances that you will be caught are very high.

Do not even think about engaging in plagiarism or academic dishonesty, it is not worth it. **Be smart and don't risk your future or break the law by engaging in plagiarism and academic dishonesty!**

Marking criteria

1. Code – 10 marks = 10% of your final mark

Requirement	Mark			
1. 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 **[0.5]**	Good; algorithm functions and is appropriate. Minor issues with design/implementation or other issues **[1]**	Excellent; algorithm is appropriate, well designed, and no issues with implementation **[1.25]**
2. Implements fully connected neural network (MLP)	No functioning algorithm/major issues with implementation **[0]**	Algorithm somewhat functions, but has serious issues with design or implementation **[0.5]**	Good; algorithm functions well. Minor issues with design/implementation or other issues **[1]**	Excellent; algorithm is appropriate, well designed, and no issues with implementation **[1.25]**
3. Implements CNN	No functioning algorithm/major issues with implementation **[0]**	Algorithm somewhat functions, but has serious issues with design or implementation **[0.5]**	Good; algorithm functions well. Minor issues with design/implementation or other issues **[1]**	Excellent; algorithm is appropriate, well designed, and no issues with implementation **[1.25]**
4. Algorithm of choice – hyperpara-meter search	No functioning hyperparameter search or completely irrelevant parameters **[0]**	Major issues with search method, parameters, or parameter values **[0.5]**	Good; minor issues with search method, parameters, or parameter values **[1]**	Excellent; well chosen search method, parameters and parameter values **[1.25]**
5. Fully connected neural network (MLP) - hyperparameter search	No functioning hyperparameter search or completely irrelevant parameters **[0]**	Major issues with search method, missing parameters, or parameter values **[0.5]**	Good; minor issues with search method, parameters, or parameter values. Not missing more than 1 of the 3 learning rates and 2 activation functions (plus none). **[1]**	Excellent; well chosen search method, all required params, and values appropriate **[1.25]**
6. CNN - hyperparameter search	No functioning hyperparameter search or completely irrelevant parameters **[0]**	Major issues with search method, parameters, or parameter values **[0.5]**	Good; minor issues with search method, parameters, or parameter values. Not missing more than 1 of the 3 learning rates 2 kernel sizes, 2 strides. **[1]**	Excellent; well chosen search method, all required params, and values appropriate **[1.25]**
7. Best hyperpara-meter combination of each model trained and evaluated on test set in separate cell	Not completed, or significant issues **[0]**		Completed with no issues **[0.5]**	
8. 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, not enough markdown text to easily read the notebook, poor variable names **[1]**	Good code quality; minor issues with one aspect such as comments, not enough markdown, or variable naming **[1.5]**	Excellent, readable code and markdown text in notebook **[2]**

2. Report – 15 marks = 15% of your final mark

Requirement	Mark			
Introduction – 1 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.) **[0.25]**	Excellent; aim of study is well discussed with no issues **[0.5]**	
2. Importance	Importance not discussed or very poor **[0]**	Importance partially discussed. Missing some aspects, such as importance of comparing classifiers or other issues **[0.25]**	Excellent; importance well justified and related to practical use **[0.5]**	
Data – 1 mark				
3. Data description and exploration	Dataset not described or very poor **[0]**	Limited dataset description with missing information, and/or no data exploration **[0.25]**	Minor issues with dataset description and/or exploration **[0.35]**	Thorough data description and exploration, including discussion of important features and challenges as mentioned in the assignment specification, with sample images where relevant. **[0.5]**
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 **[0.25]**	Minor issues with either pre-processing description or justification **[0.35]**	Excellent description of pre-processing techniques (if any) and their effect/purpose. Techniques used are justified from lectures, labs, or other sources. If no techniques are used, this must be well justified. Brief discussion of which pre-processing techniques were considered but not necessary. **[0.5]**
Methods – 6 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 **[0.5]**	Good description and justification of inclusion, with minor issues or missing detail **[0.75]**	Excellent description and justification of inclusion. Includes references where necessary. **[1]**
6. Fully connected neural network (MLP) – description	Description missing or very poor **[0]**	Major issues or omissions **[0.5]**	Good, with minor issues or missing detail. **[0.75]**	Excellent description, with sufficient detail to explain the advantages and disadvantages of the algorithm later. References included where necessary. **[1]**
7. CNN - description	Description missing or very poor **[0]**	Major issues or omissions**[0.5]**	Good, with minor issues or missing detail. **[0.75]**	Excellent description, with sufficient detail to explain the advantages and

				disadvantages of the algorithm later. Includes references where necessary. **[1]**
8. Comparison of strengths and weaknesses	Not included or very poor **[0]**	Major issues or omissions **[0.5]**	Good; minor issues including some relevant points of comparison missed. **[1]**	Excellent comparison of the relative strengths and weaknesses of the classifiers from a theory perspective. Sources cited where appropriate. **[1.5]**
9. Architecture and hyperparameter tuning description	Not included or very poor **[0]**	Major issues or omissions in description **[0.5]**	Good; architecture and search method are well described/justified and parameters chosen to search over explained. Minor issues or lacking detail. **[1]**	Excellent description and explanation/justification of architecture, search method, and chosen hyperparameter ranges. **[1.5]**

Results and discussion – 4 marks				
10. Hyperparameter tuning results presentation	No figures/tables, or only screenshots of code output. **[0]**	Figures or tables have major issues or omissions **[0.5]**	Good; figures or tables show trends/results from hyperparameter tuning, but there are minor issues with the presentation **[0.75]**	Excellent presentation of hyperparameter tuning results in figures or tables, with no presentation issues. If there are any relevant differences in runtime, these are presented. **[1]**
11. Hyperparameter tuning discussion	Not included or very poor **[0]**	Discussion has major issues or omissions **[0.5]**	Most important hyperparameter tuning results/trends are discussed. Includes comment on how the results aligned with predictions. Minor issues and/or lack of detail. **[0.75]**	Excellent discussion of hyperparameter results/trends, including possible explanations or reflections on how the results aligned with predictions. **[1]**
12. Results table (including at least best hyperparameter combination, test set performance, and training runtime)	Not included or very poor, including screenshots of code output **[0]**	Major issues with formatting or omission of multiple results **[0.5]**	Minor issues with formatting or omission of one result **[0.75]**	Excellent table with all required results and appropriate formatting **[1]**
13. Results discussion/analysis	Not included or very poor **[0]**	Discussion has major issues or omissions **[0.5]**	Most important trends in the results discussed, and compared to expectations based on theoretical properties. Minor omissions and/or lack of detail. **[0.75]**	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, precision vs. recall etc.) **[1]**
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 **[0.25]**	Minor issues with summary (e.g. does not consider runtime) or misses some relevant limitation(s) **[0.35]**	Excellent summary which considers factors such as runtime and practicality. Limitations identified are relevant. **[0.5]**
15. Future work suggestions	Not included or very poor **[0]**	Suggestions are not specific enough or do not address study limitations **[0.25]**	Minor issues with suggestions **[0.35]**	Suggestions are concrete and directly address the study limitations **[0.5]**
Reflection – 0.5 marks				
16. Reflection on most important thing learnt	Not included or very poor **[0]**	Reflection is lacking in depth or detail **[0.25]**	Excellent, relevant reflection with sufficient depth **[0.5]**	
Presentation – 1.5 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 **[0.35]**	Minor issues with structure or formatting **[0.6]**	No issues with report structure or formatting. Sections are clearly delineated and formatting is clean and legible. **[0.75]**
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 **[0.35]**	Several minor spelling or grammar mistakes that do not hinder the overall readability of the report, and/or non-academic language in some sections **[0.6]**	Very few minor spelling or grammar mistakes. Language is academic in style with clear sentences. **[0.75]**