UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

COS711 Assignment 2

Hybrid Learning in Neural Networks for Almond Classification

Due date: 29 September 2024, at 23h30

1 General instructions

For this assignment, you have to submit (1) a Jupyter notebook containing all of your code, and (2) a PDF document, containing a scientific report wherein you describe what you have done, present and discuss your findings. Guidelines for writing the report are provided in this specification document. The report will be checked for plagiarism using the Turnitin system, and must be submitted through ClickUp. You are advised but not required to typeset your report in LaTeX. Your Jupyter notebook should be hosted in a public github repository. You must provide a link to the notebook in your submitted PDF report.

2 Almond Classification

For this assignment, you will work with a dataset that contains numeric features extracted from images of almonds. The dataset contains examples belonging to three different types of almonds. Your task is to optimise and train a neural network (NN) to perform almond classification. As part of this task, you will need to preprocess the data, optimise NN's hyperparameters, and perform a comparative study between various gradient-based training algorithms. You will also develop a hybrid training algorithm that averages the gradient information across multiple training algorithms.

2.1 Data set

Download the data set from ClickUP. The data set was borrowed from Kaggle. The data is provided in a `csv` format, where each row represents a data set entry, and each column represents a data attribute. There are 2803 rows in total. Each almond is described via 12 features, such as length, width, thickness, etc. You will see that for the first three features (length, width, and thickness), only two of the three are present for each entry. The reason for this is that the data was extracted from 2D images where an almond was either held upright, laid on its side, or laid on its back. Therefore, only two of the three characteristics could be measured at any one time.

The last column lists one of the three almond types: Mamra, Sanora, or Regular. Your task is to construct a NN to perform almond *classification* into one of these three types.

(Convert L, W, H to L, W, and add one-hot encoded attribute to signify the dimension that was removed)

2.1.1 Data preparation

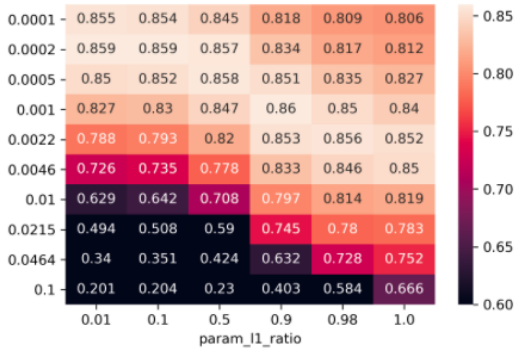
The given data contains numeric attributes that lie in various ranges. Analyse the data set, and pre-process it in a way that will make it possible for a NN to effectively discover the hidden relationships between inputs and outputs. Give extra thought to the following data set properties:

1. The dataset contains missing values. Should columns/rows with missing values be excluded, or can the missing values be imputed in a meaningful way?
2. Features such as aspect ratio and eccentricity were derived from other features (refer to the Kaggle page for formulas). What implications does this have for missing value imputation?

2.2 Hyperparameter Optimisation

As discussed in class, the performance of your NN model greatly depends on various hyperparameters, such as the NN architecture, activation functions, error (loss) function, training algorithm, etc. You will have to choose the hyperparameter values for your NN model. Your report **must** contain a section justifying all hyperparameter choices. Two justifications are acceptable: (1) theoretical insight; (2) empirical evidence. I.e., if you cannot decide on a value for a certain hyperparameter analytically, you have to run some experiments to see which value performs better than others.

You must empirically compare at least two different hyperparameters of your choice, excluding the training algorithm (see next section). While you are welcome to thoroughly optimise more that just two hyperparameters, it is required of you to perform a **grid search** over any two hyperparameters of your choice. Grid search implies selecting a set of values for each hyperparameter, and evaluating the model for every combination of values between the two sets. Visualise the results of the grid search using a heatmap such as the one shown below, where x-axis represents the values of hyperparameter A, y-axis represents the values of hyperparameter B, and each cell is colourised according to the model performance for a combination of (A,B):



NB: to compare the performance of any two hyperparameter values, you must obtain average performance for each hyperparameter across a few independent runs, as well as standard deviation. K-fold cross-validation is an excellent technique to perform hyperparameter optimisation. Discuss whether there is statistical evidence that one hyperparameter value is more performant than another value. Consider using appropriate hypothesis testing to make an informed conclusion. Hint: if standard deviations between two hyperparameter values do not overlap, you can safely conclude that the difference in performance is significant.

Since we are working with a classification task, remember to compare your models in terms of accuracy in addition to the NN loss value.

2.3 Hybrid Learning

For this assignment, you will compare different gradient-based NN training algorithms on the task of almond classification. It is up to you how many methods you compare, but you must compare **at least two**, and one of the methods must be resilient backpropagation (RProp): a simple variation of standard backpropagation that takes oscillatory behaviour into account by considering the change in the sign of the gradient for each weight, and adjusting the learning rate accordingly.

Finally, once you have two or more training algorithms working, implement a *hybrid learning approach*: use multiple algorithms to derive the weight updates, and for each weight, calculate and apply the **average** update across multiple training algorithms. Questions to ask yourself here are: how correlated are the weight updates between different algorithms? Are there cases where the algorithms point in opposite directions? Should both (all) algorithms contribute equally to the weight update?

Collect sufficient data (i.e. results from multiple independent runs) for each of your training algorithms, including the hybrid. How do the algorithms compare? Discuss and interpret all your results thoroughly. Remember that simply pasting a table with numbers in it, or a graph with no explanation, will **not yield any mark**s. Visualise the experimental data where appropriate: it is much easier to analyse your results when you can see them plotted next to one another in different colours. If you see that one approach is doing better than another, provide a hypothesis for why it is the case. Running the experiments is only half of the research process, the other and more important half is interpretation. Aim to derive as many insights from your results as you can.

3 Notes

- Implementation
  - You are required to use Python programming language for this assignment, and submit your code in Jupyter notebook format.
  - You may use a machine learning/neural network library/framework.
- Report
  - You must report on all data preparation steps taken.
  - You must report on all NN hyperparameters used, and substantiate your choices.
  - Training and testing errors have to be reported. Remember to report means with the corresponding standard deviations, and report how many independent runs have been performed.
  - To compare individual training algorithms, plot their training and testing loss values over multiple epochs.

4 Marking and general guidelines

For this assignment you have to submit a **research report** where you discuss your findings. Your report **must** contain a link to a Jupyter notebook in a public github repository. Your reports must follow the IEEE conference format (https://www.ieee.org/conferences/publishing/templates.html). You may use the Latex or the Word template, however, it will serve as good academic writing practice to utilise LaTeX. There is also a strict page limit of **8 pages** for this assignment. Given the imposed two column format it would require a substantial amount of writing to exceed this limit.

This is not a course in technical report writing; however, you should make your report as readable as possible. You are more likely to obtain a higher mark if your report generates a good impression with the marker and is void of general errors such as spelling and grammar mistakes. A typical research report would consist of the following sections:

1. **Abstract**

   The abstract should briefly summarise the purpose and findings of the report.

2. **Introduction**

   The introduction sets the stage for the remainder of your report. You usually have very general statements here. The introduction prepares the reader for what to expect from reading your report. In general, the introduction should either contain or be a summary of your ENTIRE report. Keep the introduction concise, try to limit it to 1 page maximum.

3. **Background**

   A very high level discussion on the problem domain and the algorithms and/or approaches that you have used. This section is typically where the "base cases" of concepts that appear throughout the remainder of your report are discussed. It is also an ideal place to refer a reader to other sources containing relevant information on the topic which is outside the scope of your assignment. Remember to discuss very generally. After reading this section the marker should be able to determine whether or not you **understand** the techniques that you are using. Try to limit this section to 1 page maximum. Make sure you reference the relevant sources when discussing the building blocks of your project.

4. **Experimental Set-Up**

   In this section you discuss how you approached, implemented and solved your assignment. Mention the values considered for the hyperparameters, how many simulations you have run, etc. After reading this section (in addition to the background) the reader should be able to **replicate** your experiments to obtain similar results to those obtained by you. Be very specific in your discussions in this section.

5. **Research Results**

   This is the section where you report your results obtained from running the experiments as discussed in the experimental set-up section. You have to give, at the very least, the averages and the standard deviations for all the experiments/simulations. Graphing your results is advisable, and no conclusions regarding the superiority of one approach over another can be made without some form of statistical reasoning. Training and testing errors have to be reported. Thoroughly discuss the results that you have obtained, and reason about why you obtained the results that you have. Answer questions like "are these results to be expected?" and "why these results occurred?" and "would different circumstances lead to different results?"

6. **Conclusions**

   Very general conclusions about the assignment that you have done. This section "answers" the questions and issues that you have raised and investigated. This section is, in general, a summary of what you have done, what the results were, and finally what you concluded from these results. This is the final section in your document, so be sure that all the issues raised up until now are answered here. This is also the perfect section to discuss what you have learnt in doing this assignment.

7. **Bibliography**

   Every research report must be supported by valid bibliographic sources, or references. Make sure that your bibliography lists valid academic peer-reviewed sources rather than unverified online tutorials.

Please **remember** to include a link to the Jupyter notebook containing all of your code! The link must feature in the *Experimental Set-Up* section. Reports that do not provide a Jupyter notebook link will have their total mark halved.

4.1 Marking

The following general breakdown will be used during the assessment of this assignment:

Category	**Mark Allocation**
Report structure	5 marks
Background	5 marks
Data preparation	10 marks
Experimental setup	10 marks
Hyperparameter optimisation	20 marks
Comparison between RProp and other optimisers	20 marks
Hybrid learning implementation and evaluation	20 marks
Conclusions	5 marks
Bibliography	5 marks
Penalty for not including a Jupyter notebook link	-50% of total marks
TOTAL	100 marks

Upload the PDF file to the appropriate assignment slot on ClickUp. Multiple uploads are allowed, but only the last one will be marked. The deadline is **29 September 2024, at 23h30**.