COS 711 Assignment 2 Report

Nathan Opperman

u21553832@tuks.co.za

***Abstract*—In order to compare and contrast different training algorithms I believe it is appropriate to, you know what fuck this is stupid The abstract should briefly summarise the purpose and findings of the report.**

***Index Terms*—This probably is not necessary**

I. 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

II. 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.

III. Experimental Set-Up

In this section, I will attempt to outline the approaches and strategies utilized throughout this assignment and provide rationale for the various choices made regarding data preparation, hyperparameters and hybrid learning.

A. Software Stack

To aid in this assignment, I used a Docker container to host and run Jupyter notebooks. Specifically, I used the "jupyter/scipy-notebook_ 64-ubuntu-22.04" Docker image. For the machine learning and neural network processes, I decided to use the PyTorch framework, known for its flexibility, efficiency and more importantly its simplicity. Additionally, I incorporated essential libraries such as Pandas and NumPy for data manipulation and preparation, along with scikit-learn (sklearn) for various preprocessing and modeling tasks. This software stack provided a comprehensive set of tool for addressing the task at hand.

B. Dataset Description

The "Almond Types Classification" dataset contains a total of 2803 almonds that are described via fourteen attributes (Id and Type included). The almonds were captured using 2D images each of which in one of three different positions: upright, on its side and laid on its back. The goal of this dataset is to classify each almond into one of either three types: SANORA, MAMRA and REGULAR.

C. Data Preparation

Data preparation is an important aspect of any learning process and is vital to enhancing the performance of any supervised learning process.

1) Outliers: Outliers can have a significant impact on the performance of the model and so it now requires attention. In order to detect outliers I used the InterQuartile Range (IQR) method. With this method the rows outside the 25^{th} and 75^th percentile are removed from the dataset. Missing values are excluded from this and are not removed. This is performed for each column excluding the target attribute "Type". After removing the outliers we are left with

2) Missing Values: Now that the outliers have been removed we can focus on the main issue of this dataset regarding the attributes: Length, Width and Thickness, whereby each almond is missing exactly one of these attributes. Specifically 30.57% are missing Length, 33.61% are missing Width and 35.82% are missing Thickness. This is clearly an extensive amount of missing information, to address this I decided to utilize the iterative imputer (IterativeImputer) from the scikit-learn library. This is a multivariate imputer that estimates each feature from all the others. It uses a strategy for imputing values by modeling each feature with missing values as a function of others in a round-robin like fashion.

There is still the issue concerning the attributes Roundness, Aspect Ratio and Eccentricity. These features are dependent on Length, Width, and Thickness. Therefore, including these dependent attributes in the iterative imputation process would be pointless, since their values are absent when the corresponding independent features are missing. Therefore, I have decided to exclude these dependent features from the imputation process. This leaves the following attributes:

Length, Width and Thickness. The correlation matrix below demonstrates that these features are interrelated, suggesting that they will be effective for iterative imputation.

TABLE I
CORRELATION MATRIX

Features	Length	Width	Thickness
Length	1.000	0.837	0.458
Width	0.837	1.000	0.595
Thickness	0.458	0.595	1.000

Lastly, I noticed an issue specific to the ”Roundness” attribute. Unlike the other dependent attributes this attribute is calculated from both the Length and Area. This is an issue as Area is clearly influenced by the angle of the almond in the 2D image from which it is extracted. Therefore, relying on the imputed Length and original Area to compute Roundness may not be sufficient. To address this, I created a temporary attribute called Area_ Length, which equals Area when Length is present and is NaN when Length is not present. This approach aims to provide a more reliable estimate for calculating Roundness.

Once the missing values have been imputed they are then used to recalculate the Roundness, Aspect Ratio and Eccentricity.

3) Encoding: Before we can utilize neural networks we need to ensure that the data is the correct format. This is especially true for the target attribute ”Type”. This is a nominal value, hence there is no inherit order to the category. Therefore I utilized one-hot encoding. This ensures that the data is in a numerical format and that the model cannot interpret any inherent order from the data.

TABLE II
ONE-HOT ENCODING REPRESENTATION FOR ALMOND TYPES

Almond Type	MAMRA	REGULAR	SANORA
MAMRA	1	0	0
REGULAR	0	1	0
SANORA	0	0	1

4) Bias: Neural networks are data-driven models, meaning their performance and predictions are highly dependent on the data they are trained on. This means that if our training data contains bias, the model will likely learn this bias and propagate it in its predictions on unknown data. This is a particularly important issue, luckily this dataset appears to represent each class equally with 943 (33.64%) SANORA almonds, 933 (33.39%) MAMRA almonds and 927 (33.07%) REGULAR almonds. Therefore each class is represented almost equally and so class bias should not be an issue.

5) Data Splitting: In order to train, optimize and evaluate the model I split the data into the standard 70:20:10 ratio, with 70% of the data allocated to training the model, 20% allocated to validation, and 10% allocated to testing. To ensure that the target attribute is represented in each set I used stratified sampling. With this approach the proportion of samples for each class remains the same across the training, validation, and testing sets.

6) Data Scaling: To prevent saturation for activation functions like sigmoid or tanh I utilized Z-Score normalization to scale all input features to zero mean and unit variance which helps avoid the issue of neurons becoming saturated. The training data transformation is also applied to all three data sets to produce the most realistic test error estimates.

D. NN Architecture

E. Weight Initialization

F. Objective Function

G. Learning Rate, Epochs and Batch Size

H. Training Algorithms

I. Hybrid Training Algorithm

IV. 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?”

V. 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.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[8] D. P. Kingma and M. Welling, “Auto-encoding variational Bayes,” 2013, arXiv:1312.6114. [Online]. Available: https://arxiv.org/abs/1312.6114

[9] S. Liu, “Wi-Fi Energy Detection Testbed (12MTC),” 2023, gitHub repository. [Online]. Available: https://github.com/liustone99/Wi-Fi-Energy-Detection-Testbed-12MTC

[10] “Treatment episode data set: discharges (TEDS-D): concatenated, 2006 to 2009.” U.S. Department of Health and Human Services, Substance Abuse and Mental Health Services Administration, Office of Applied Studies, August, 2013, DOI:10.3886/ICPSR30122.v2

[11] K. Eves and J. Valasek, “Adaptive control for singularly perturbed systems examples,” Code Ocean, Aug. 2023. [Online]. Available: https://codeocean.com/capsule/4989235/tree