F20BC/F21BC - Coursework Group Signing Sheet

INSTRUCTIONS

Each group should print one copy of this sheet, fill it in, sign it and hand it in together with the CW.

The group is to agree the % of total effort put in by each of the group member's to the development of the coursework that should sum up to a 100%. Each member must sign to show they agree the % of total effort and the sheet must be handed in together with the CW.

No marks will be issued until we have this signed copy.

Group Number: ____________

PRINT NAME AND STUDENT ID NUMBER	SIGNATURE	CONTRIBUTION TO PROJECT	% TOTAL EFFORT
SHAHIDULLA VANTELA	V. S	We as partners discussed and worked together. I worked upon creating ANN architecture, cost functions, activation functions as well as necessary functions and acquired results. We both worked on report writing.	50%
PRASANNA NIMMAGADDA	N.V.L. Prasanna	We as partners worked simultaneously. I worked on logistic regression, accuracy scores, encoding and we worked together on report writing as well	50%
Total:			100%

DECLARATION 1: We agree that the above information genuinely reflects the effort put in by group members and we **WISH / DO NOT WISH** (circulate as appropriate) marks to be allocated equally.

DECLARATION 2: We declare that the coursework has not been copied or plagiarised in any way.

NOTE: Unless indicated otherwise, all marks will be allocated equally. In case of any dispute on group contribution/mark allocation, the course leader will be final arbiter.

Please refer to the student handbook on notes on plagiarism. All cases of detected plagiarism will be reported to the disciplinary committee for consideration.

Student Declaration of Authorship

Course code and name:	F21BC-Biologically Inspired Computation
Type of assessment:	**Group**
Coursework Title:	Coursework 1
Student Name:	Shahidulla Vantela
Student ID Number:	H00397767

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Shahidulla Vantela*

Date: *28/10/2022*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Student Declaration of Authorship

Course code and name:	F21BC-Biologically Inspired Computation
Type of assessment:	**Group**
Coursework Title:	Coursework 1
Student Name:	Vijaya Lakshmi prasanna nimmagadda
Student ID Number:	H00395559

Declaration of authorship. By signing this form:

- **I declare** that the work I have submitted for individual assessment OR the work I have contributed to a group assessment, is entirely my own. I have NOT taken the ideas, writings or inventions of another person and used these as if they were my own. My submission or my contribution to a group submission is expressed in my own words. Any uses made within this work of the ideas, writings or inventions of others, or of any existing sources of information (books, journals, websites, etc.) are properly acknowledged and listed in the references and/or acknowledgements section.

- I confirm that I have read, understood and followed the University's Regulations on plagiarism as published on the University's website, and that I am aware of the penalties that I will face should I not adhere to the University Regulations.

- I confirm that I have read, understood and avoided the different types of plagiarism explained in the University guidance on Academic Integrity and Plagiarism

Student Signature *(type your name):* *Vijaya Lakshmi prasanna nimmagadda*

Date: *28/10/2022*

Copy this page and insert it into your coursework file in front of your title page.
For group assessment each group member must sign a separate form and all forms must be included with the group submission.

Your work will not be marked if a signed copy of this form is not included with your submission.

NAME vijaya Lakshmi prasanna nimmagadda

NAME shahidulla vantela

INTRODUCTION

In machine learning, Artificial Neural Networks (ANN) are one of the most important techniques used in classification tasks. They are brain-inspired systems that aim to mimic how humans learn, using multiple nodes as found in their layers. Basically, they consist of an input layer, an output layer, and one or more hidden layers consisting of units that transform the input into something the output layer can use. This report discusses the various steps we adopted in building a multi-layer ANN that mimics how the human brain learns. We also investigated how different hyper-parameters could impact the generalization of the model or classifier using a given dataset.

PROGRAMME RATIONALE

The rationale behind this project is to implement a Multi-layer ANN architecture, using a given dataset to predict if breast cancer is either Malignant(M) or Benign(B). The project also aims to investigate how various parameters, e.g. the weight and bias of a neuron and hyperparameters such as activation functions, learning rate, number of hidden layers and nodes, epoch size and more, could help improve or decrease the performance of an ANN model.

The design and implementation of the Multi-layer ANN architecture was done using python programming language, without the use of any Neural network libraries and framework. It was built from scratch to allow one to understand how the ANN learns, using both the feedforward and feed-backward mechanisms.

METHODS

The dataset used in this work comprises UCI breast cancer dataset obtained from []. The following methods were adopted to build the Multi-layered ANN architecture.

I. Data loading: The dataset was loaded to investigate the nature of the dataset, the size of the data using pandas [1], if they were any columns with missing or non-sensitive values, and also to investigate which column relates to the label and features.

II. Data Preprocessing: After loading the data, the dataset was split into label and feature columns. The label corresponds to columns with 'M or B' as values, whereas others were considered as features. Upon split, some features, such as image_id, were dropped since it doesn't have any importance to the task. Furthermore, the labels were converted into numerical values since they were found to contain categorical values, which could result in errors while building the model. Lastly, most features were found to have high values. Hence, we performed scaling to standardize the data within a given range with a mean of 0 and a variance of 1. This was done using the StandardScaler library provided by sklearn [2].

III. Model Building: The model Building was done using feedforward and feed backward propagation [3]. The feedforward propagation was used to determine the neural network's output using different activation function combinations. For this task, we used a sigmoid activation function[reference] as the activation function of the output layer since it is best suited for the binary classification task. However, several combinations of activation functions, including ReLu [4], tanh [5], and sigmoid, were used as the activation function for the hidden layers to ascertain the best combination of activation functions that best minimizes the error of the final output layer. The backward propagation was carried out to determine gradients which would be used to update the model's parameters to help minimize significant errors accumulated during the forward propagation. This was done using the gradient descent algorithm with a range of learning rates.

IV. Model Training – The model was trained by fitting the model to the UCI breast dataset which were earlier pro-processed as discussed. On the first attempt, it was trained using the entire dataset. Later on, the dataset was split into train and test dataset. It is worth mentioning to note that data were standardized after we had split the data to avoid data-leakage [6]. Training was done on the train dataset alone using a given EPOCH size.

V. Model Evaluation: Model Evaluation was done on both the entire dataset, and the test dataset after splitting. Several loss functions were tested to check how the errors were minimized at each stage of training. Furthermore, performance metrics such as accuracy_score and confusion matrix were used to determine how well the model learned on both seen(trained) and unseen(test) data. In this Scenario, our aim was to investigate if the model was performing well, overfitting, or underfitting.

Accuracy score basically represented the percentage of correct prediction from the dataset. The confusion matrix allowed us to investigate True positive values, true negative values, and both the false positive and false negative values.

RESULTS

After following the methods we discussed above, we realized the following results. At first, upon loading the dataset, we noticed the total number of records found was 569, and 32 Columns. After the data preprocessing, 30 columns were selected as features, 1 was dropped, and 1 was used as the label.

The Weights was initialized to an array of random numbers, while the bias was initialized to an array of zeros.

The following results were obtained upon building and training the model using various sets of hyperparameters. The results are as follows using the entire dataset.

Table: Performance of model using different Hyper-parameters

Epoch	Learning rate	Hidden layers	Hidden node +Activation function	Input node	Output node + Activation function	Accuracy
30000	0.001	1	10 - ReLu	30	2 - sigmoid	98.41
30000	0.001	1	10 - tanh	30	2 - sigmoid	98.41
30000	0.001	1	10 - sigmoid	30	2- sigmoid	93.50
30000	0.001	2	30 -Relu	30	2 -sigmoid	98.54

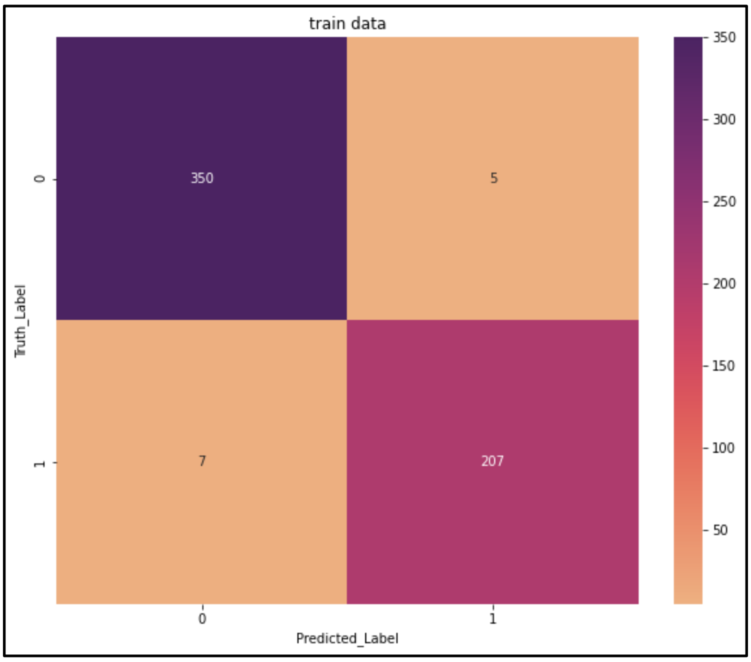

Fig 1.0: A confusion matrix showing reports generated on the training dataset using the relu hidden activation function, sigmoid output activation function, 10 hidden nodes, and a learning rate of 0.01. (Correctly classified data = 557, incorrectly classified data = 12)

DISCUSSION AND CONCLUSION

As shown in the table above, hyper-parameters greatly influenced the model's performance [7]. The table shows that ReLu activation function outperforms both tanh and sigmoid activation functions when used at the hidden layers. Also, although the computation time was not represented in the table, I noticed that the training time for ReLu was less than other activation functions used. Furthermore, the learning rate greatly affected the accuracy of the model. For this model, the best accuracy was observed while using a learning rate within 0.01 – 0.001, learning rate above 1 tends to explode the gradient, resulting in NaN (not a number) while trying to compute the cost function. Hence, for the best performance of an ANN model, a combination of learning rates within the range of 0.01 – 0.001, reLu activation function in the hidden layer, and sigmoid activation function allow the best to generalize well. There was not much increase in performance while increasing the number of nodes in the hidden layer; however, I noticed the training time increased.

REFERENCES

1. McKinney, W., *pandas: a foundational Python library for data analysis and statistics.* Python for high performance and scientific computing, 2011. **14**(9): p. 1-9.
2. Bisong, E., *Introduction to Scikit-learn*, in *Building machine learning and deep learning models on Google cloud platform*. 2019, Springer. p. 215-229.
3. Abiodun, O.I., et al., *State-of-the-art in artificial neural network applications: A survey.* Heliyon, 2018. **4**(11): p. e00938.
4. Agarap, A.F., *Deep learning using rectified linear units (relu).* arXiv preprint arXiv:1803.08375, 2018.
5. Lau, M.M. and K.H. Lim. *Review of adaptive activation function in deep neural network*. in *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*. 2018. IEEE.
6. Samala, R.K., et al. *Hazards of data leakage in machine learning: a study on classification of breast cancer using deep neural networks*. in *Medical Imaging 2020: Computer-Aided Diagnosis*. 2020. SPIE.
7. Breuel, T.M., *The effects of hyperparameters on SGD training of neural networks.* arXiv preprint arXiv:1508.02788, 2015.