Heart Disease Detection Using Deep Neural Networks

Ahmed Tameem

Abstract

In this report two different supervised learning approaches are discussed to handle the binary classification problem of finding out if a patient has a heart disease or not in an explainable manner. The first approach is a simple logistic regression classifier (LRC), the second approach is a deep neural network (DNN) classifier.

1 Introduction

1.1 The Problem

This is a binary classification supervised learning problem which has 13 attributes. These attributes are used to predict whether a patient suffers from a heart disease or not. The data set used has 1025 labelled samples. The list of attributes is shown in the figure below.

1. age
2. sex
3. chest pain type (4 values)
4. resting blood pressure
5. serum cholestoral in mg/dl
6. fasting blood sugar > 120 mg/dl
7. resting electrocardiographic results (values 0,1,2)
8. maximum heart rate achieved
9. exercise induced angina
10. oldpeak = ST depression induced by exercise relative to rest
11. the slope of the peak exercise ST segment
12. number of major vessels (0-3) colored by flourosopy
13. thal: 0 = normal; 1 = fixed defect; 2 = reversable defect

1.2 Data Collection and Pre-processing

The data-set was downloaded from kaggle [1]. It has no missing/null fields so no pre-processing is needed. 80% of the data was used for training and the other 20% was used for validation.

1.3 Logistic Regression Classifier

The LRC was used to determine the feasibility of such a problem while being computationally cheap, this allows for rapid and efficient prototyping. The LRC also helps explain the results produced, this

ELEC 400M (2021-2022), UBC.

is because the magnitude of the weight assigned to each attribute shows how important this attribute is.

1.4 Deep Neural Network

The DNN is less explainable than the LRC by nature but it has a higher learning capacity which should lead to a better accuracy at the expense of a less explainable result, more computational power used, and less efficient development due to slower training.

2 Logistic Regression Classifier

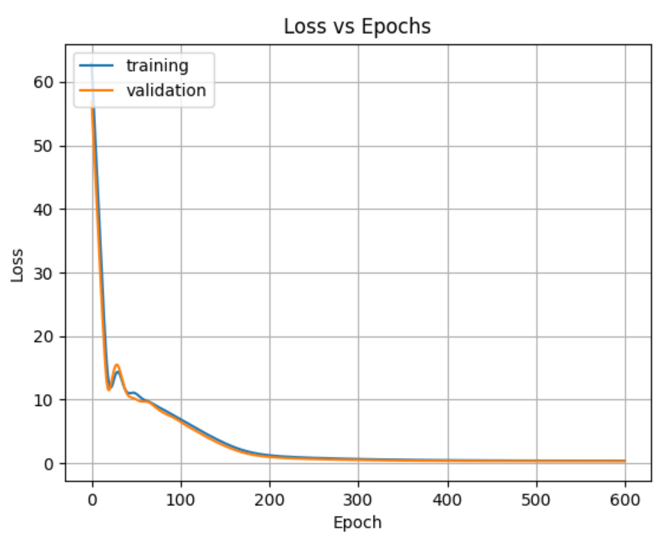
2.1 Methodology

The LRC has 13 input neurons (one for each feature). It utilizes a "Sigmoid" activation function and utilizes "He Normal Initialization" to initialize its weights and biases. It also utilizes "Binary Cross-entropy Loss". Finally, it uses ADAM as its optimizer.

During training, a learning rate of 0.01 was used for the first 50 epochs and then the learning rate was decayed to 0.001 for the rest of training. The model was trained for 600 epochs using a batch size of 1025 (all the samples at the same time). The reason the learning rate decay was used is that after the first 50 epochs the loss (both training and validation) plateaus at a value close to 0. Decaying the learning rate sometimes results in finding a better local minimum after the loss plateaus, but in this case it didn't reduce the loss as it was already very small.

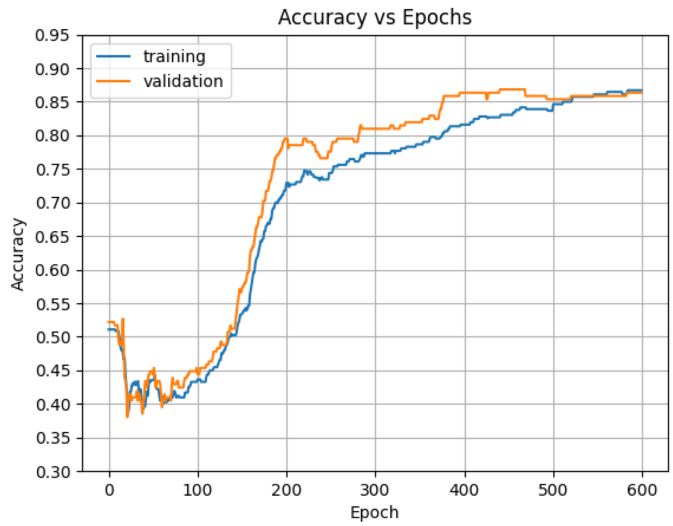
2.2 Results

The loss curves of both the training and validation plateaued at a value very close to 0, suggesting that a more complex model can help improve the accuracy.



The training accuracy achieved was 86.71% and the validation accuracy achieved was 86.34%. This indicates that the model is generalizing well and is not over-fitting.

The confusion matrix yielded the following results: 78 TP, 99 TN, 20 FP, and 8 FN. The area under the curve (AUC) of the ROC curve is 0.86 which suggests that the model is good at distinguishing between the 2 classes. Finally, the F1 score is 0.88, suggesting that the model is not over-fitting and that it has a low bias and low variance.



3 Deep Neural Network

3.1 Methodology

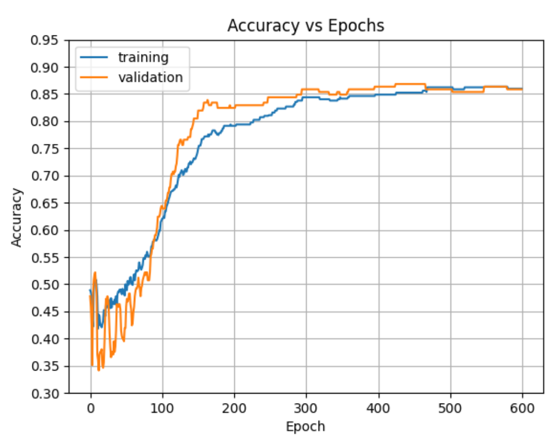
The DNN had the exact same parameters as the LRC with an addition of a "ReLU" activated 13-unit hidden-layer between the input and output layer. At first, data was not standardized, but after standardizing the data by subtracting its mean and dividing by its standard deviation large improvements were seen. Many learning rate schedules were tested but none of them ended up affecting the results positively or negatively, therefore, a constant learning rate of 0.01 was used. The most likely reason learning rate scheduling did not affect the result is that the loss was already very low.

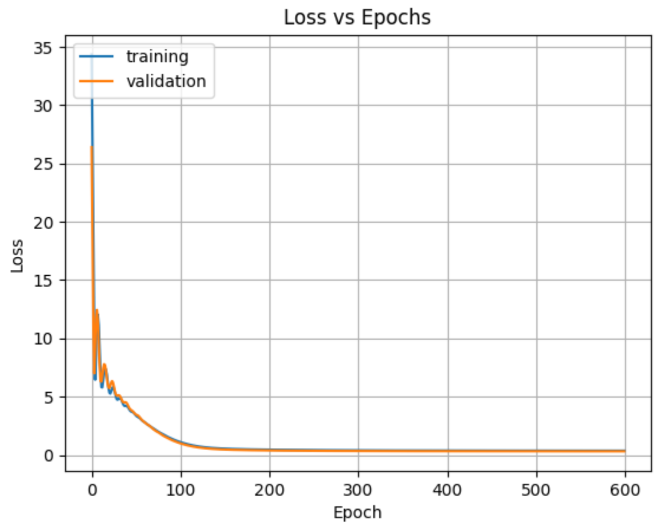
3.2 Results

3.2.1 Non-standardized Input

The training accuracy was 86.83% and validation accuracy was 86.34%.

The confusion matrix yielded the following results: 76 TP, 101 TN, 22 FP, and 6 FN. The AUC of the ROC curve is 0.86 which suggests that the model is good at distinguishing between the 2 classes. Finally, the F1 score is 0.88, suggesting that the model is not over-fitting and that it has a low bias and low variance.

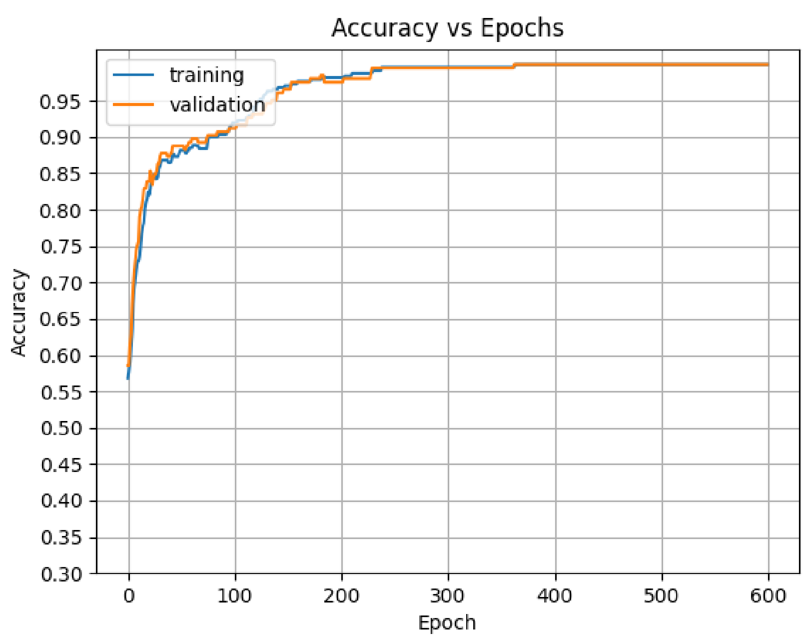


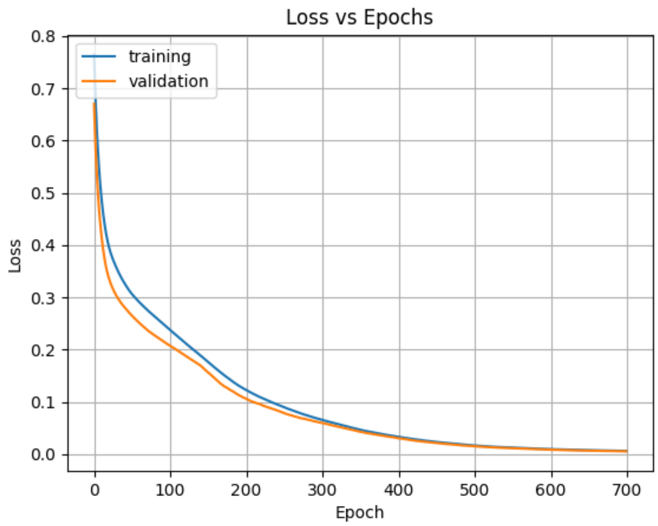


3.2.2 Standardized Input

The training and validation accuracy was 100%.

The confusion matrix yielded the following results: 98 TP, 107 TN, 0 FP, and 0 FN. The AUC of the ROC curve is 1 which suggests that the model is good at distinguishing between the 2 classes. Finally, the F1 score is 1, suggesting that the model is not over-fitting and that is has a low bias and low variance.

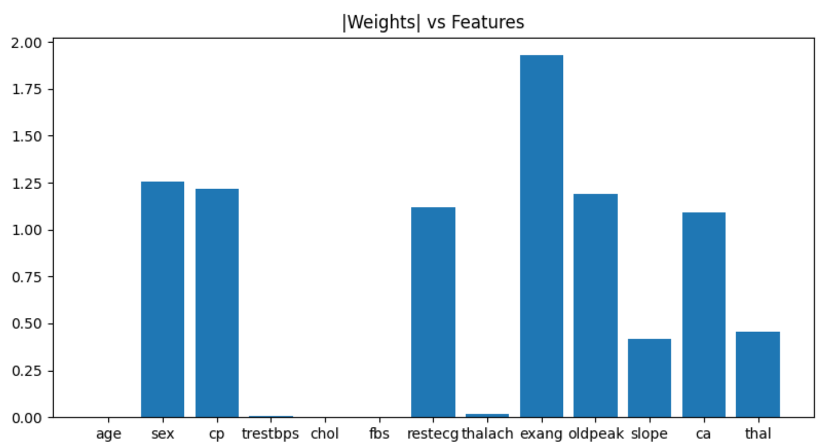




4 Discussion

Overall, The DNN performed better than the LRC, especially after standardizing the data. It is clear that more data is needed to verify how well the DNN generalizes. While the DNN trained on standardized data achieved an accuracy of 100%, it would have certainly had a lower accuracy had there been more data available.

The LRC gave some valuable insights as to what attributes are the most important.



For example, we can see that patients with exercise induced angina (exang) are much more likely to have a heart disease. We can also see that males are much more likely to develop heart diseases compared to females.

5 References

[1] Lapp, D. (2019, June 6). Heart disease dataset. Kaggle. Retrieved April 17, 2022, from https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset?resource=download