Implementing Bimodal distribution on the noisy dataset using neural network

Prateek Arora, u674244@anu.edu.au

Research School of Computer Science, Australian National University

Abstract: A good dataset is essential as it helps the neural network to learn the relationship between the inputs and outputs easily. It is important to remove the outliers from the dataset as it will refine the dataset for the usage. In this project, we are implementing the Bimodal distribution removal algorithm to remove the outliers. Encoding and feature selection is an efficient way for the algorithm to understand the dataset. This report explains that how the BDR algorithm contribute in making the neural network model good. Initially, the neural network works with normal backpropagation method and then the BDR algorithm is applied on the training dataset on different neural network and comparison between them is done on the basis of the RMSE value in the report.

Keywords: Outlier Detection, BDR (Bimodal Distribution Removal) algorithm, Encoding, Feature Selection, COMP 1111 dataset, RMSE and variance

Introduction

Feedforward neural network is used to make the prediction from the given dataset. It helps the algorithm to learn the relationship between inputs and outputs to draw some inference and make really good predictions. Backpropagation applied in the neural network helps the algorithm to update the weights in the neural network which makes the model efficient. But outliers in the dataset makes the algorithm to overfit mostly. So, Bimodal distribution is used to remove the outliers from the dataset to improve the accuracy as well as decrease the error rate.

This report explains how the algorithm works and how the outliers are removed from the dataset. The given dataset contains the marks of the students who are enrolled in the COMP 1111 course at UNSW. Regression neural network model is used to predict the final exam marks based on the marks they got in the mid-semester, assignments as well as lab tests. But before the dataset is used for prediction by the neural network, we should do feature selection as well as pre-processing for the good performance.

Feature Selection

Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in [4]. It is important to consider feature selection as a part of the model selection process because if you don't, you may inadvertently introduce bias into your models which can result in overfitting [6]. The reason for not selecting the Registration number is, because it is unique for every student and we cannot extract any meaningful information from it. The feature named tutgroup and Crse/Prog is chosen because it could be that the program in which the student has enrolled is difficult for them or it could be that the tutor in the tutgroup is either good or bad at teaching which can indirectly affect marks as well as learning of the students. The marks of the students are also added in the training and testing set because the marks are the main key components on which prediction will be made in the neural network. In addition to it, some of the data points where all the values where empty including the tutgroup was removed from the dataset initially.

Encoding

The objective of encoding in the dataset is to extract the quality data and useful information from the data so that it can be easily interpreted by the algorithm [1]. Given dataset contains some incomplete rows which are replaced with the 0's using the encoding technique. This makes the dataset complete and almost ready for the neural network. There are some columns named 'Tutgroup' as well as Crse/Prog which has categorical labels. So, for them to be understood by the algorithm, we assign numeric labels using the Label Encoder.

The dataset is converted to numeric as most of the data given is in the form of string. Normalization is the next step towards making the dataset easy for the neural network to understand. In this dataset, normalization is needed because the dataset has values ranging from 0 to 100. So, interpretation of the dataset is difficult for the algorithm. So, to ease it, we normalize each column with its maximum value that could have been achieved. Now, the normalized data is ready to be used for the neural network prediction.

Implementation

Model

The network topology formed in this neural network is that there are 13 inputs with 2 hidden layers, with 15 nodes in the first hidden layer and 10 hidden neurons in the second hidden layer and the output node for that neural network is 1. The reason for choosing these number of hidden layers is because the number of nodes below that makes the model to underfit and neurons more than that makes the model to overfit. MSE error is used as the loss function as it performs best in the regression model. Adam is used as an optimizer in the neural network. The reason for choosing this optimizer is because it requires very less memory as well as it is well suited for the dataset having too many parameters

or features [5]. The learning rate of the Adam optimizer is kept to 1e-4 because if the learning rate is increased, the model tends to overfit. After all, the optimizer Adam converges quickly as well as doesn't generalize much which results in overfitting. The learning rate is not further decreased as the learning rate lower than that doesn't give good results and at 1e-2 learning rate, the results are optimum. Number of epochs is kept to 500 so that the model can be trained properly.

BDR Algorithm

BDR algorithm is an algorithm which helps in removal of the outliers from the dataset. It works much better than the least trimmed squared error since BDR produces lower bias and variance in the training data as compared to the least trimmed error [2]. Removing the outliers from the dataset makes the algorithm more robust towards predicting and learning the relationship between the inputs and output. In this algorithm, we are more concerned about the variance of the losses. We have to make sure that the initial normalized variance should be less than 0.1. After that, we can start training the model on the training dataset. Mean error and standard deviation should be calculated for each epoch. From that we can use the formula given below that helps us know to the threshold error

$$\text{Threshold} \geq \mu_{ss} + \alpha\delta_{ss}$$

[2]

Here, μ_{ss} is the mean of the error of the epochs, δ_{ss} is the standard deviation of the error of the epochs and α is the variable that lies between the range of 0 and 1. Over here, the value of α is set as 0.5 in the algorithm.

In the loop, when the error of any data point is greater than the threshold at any epoch, then that particular data point is removed from the dataset as it is considered as an outlier in the dataset. So, this procedure repeats until we get the variance less than 0.01. Now at this point, the training is stopped and then the prediction is made on the testing dataset that we have.

Results and Analysis

The main thing that matters in the regression model is to decrease the RMSE value. After the pre-processing of the model, we start training the feedforward neural network with the normal backpropagation implementation in the algorithm. This provides us with the good results as given below:

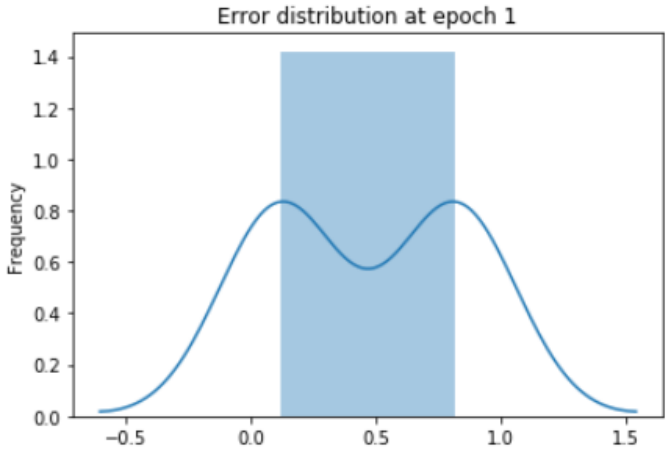


Fig 1.1

Training error at 1st epoch is 0.3776305615901947

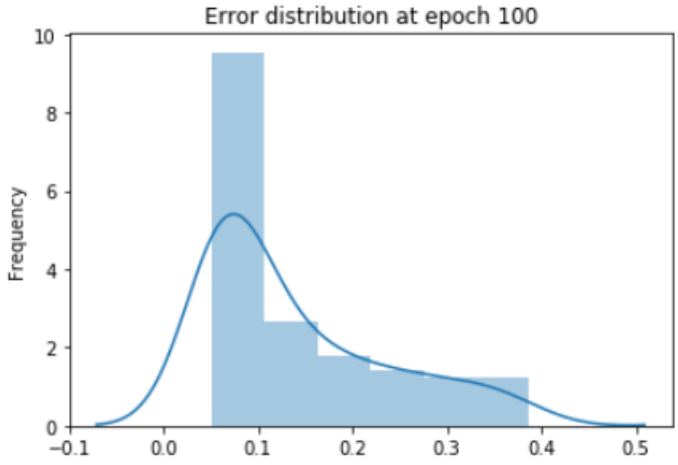


Fig 1.2

Training error at 100th epoch is 0.051100414246320724

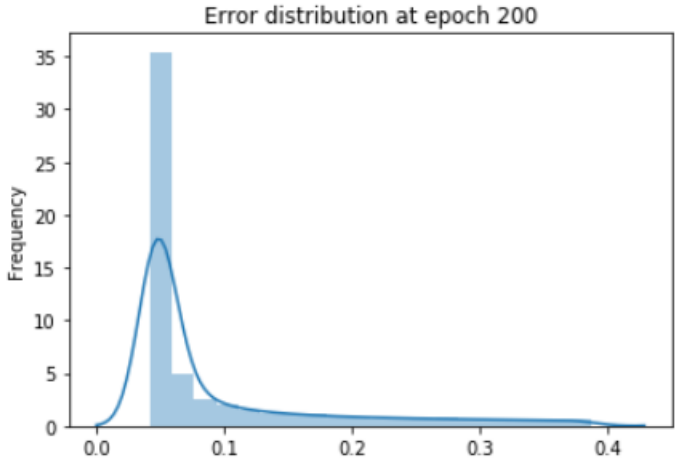


Fig 1.3

Training error at 200th epoch is 0.04178202152252197

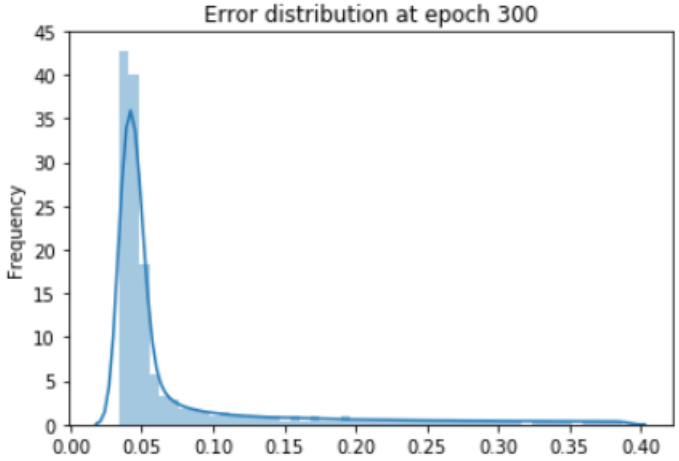


Fig 1.4

Training error at 300th epoch is 0.033965010195970535

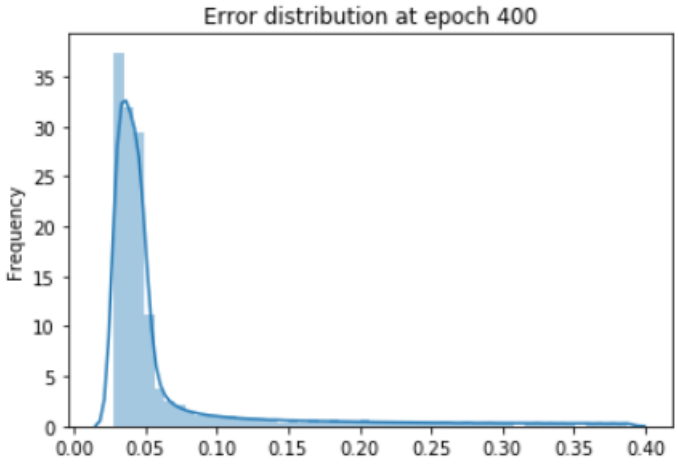
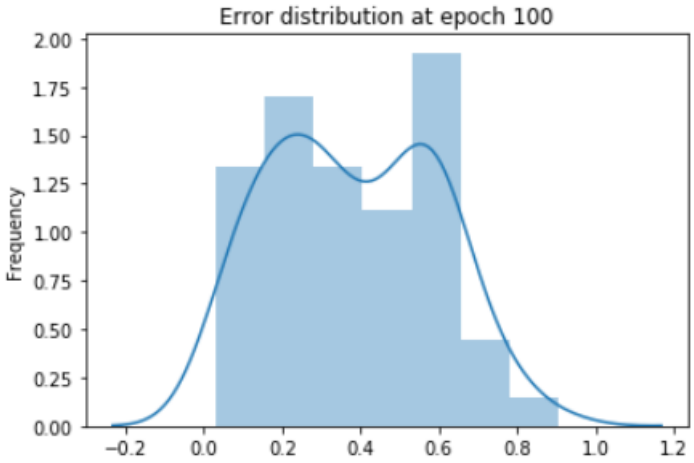


Fig 1.5

Training error at 400th epoch is 0.027345331385731697

Test loss for this model is: 0.030787

Our objective is to remove the outliers from the dataset to improve the performance of the model by using the BDR algorithm. When the algorithm is applied to the model, we start to lose some of the data points as they are the outliers in the dataset. So, the result that we get is as follows:



Training error at 100 is 0.18990333378314972

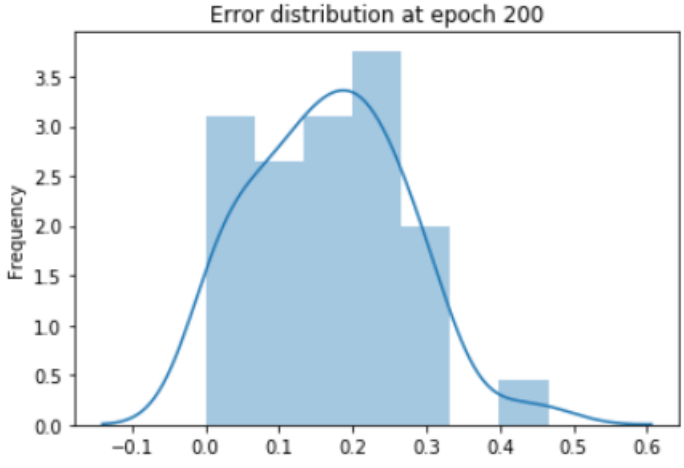
Epoch Number 100

Mean error 0.381531

Standard deviation error 0.2105646

Threshold error 0.4868132993578911

Number of data points removed from the dataset at 100th epoch is 40



Training error at 200 is 0.03849923610687256
Epoch Number 200
Mean error 0.16774964
Standard deviation error 0.10178062
Threshold error 0.21863995492458344
Number of data points removed from the dataset at 200th epoch is 22

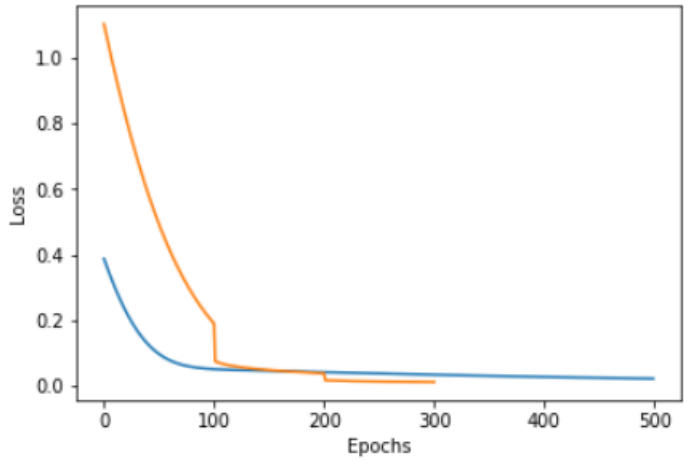


Fig 1.7
(Graph between the error of normal backpropagation algorithm and the error when the BDR algorithm is applied to the training dataset)

Now the training stops as the variance goes below 0.01. So, the training error after applying BDR is 0.03845 and after running the model on the test data, the error that we get is 0.065668. The training dataset initially consisted of 116 data points and now after the BDR algorithm, the training dataset gets decreased to 40 data points making 66 data points as the outliers. But as we observe from the results that there not much change in the training as well as the testing error even after we remove the outliers from the dataset. So, from this we can't draw a conclusion on whether BDR algorithm is good for this dataset or not.

So, to conclude whether BDR algorithm should be included or not, we can suppose that there are some points in the testing set which are outliers and if we try to remove those outliers also, we might get the good result. Initially BDR algorithm is applied on the whole dataset and then the data points which act as an outlier in the data are removed from the whole dataset. When the whole dataset is passed to the algorithm, the following result is obtained.

Training error at 50 is 0.25965946912765503
Epoch Number 50
Mean error 0.46656546
Standard deviation error 0.20488086
Threshold error 0.5485178053379058
Number of data points removed from the dataset at 50th epoch is 54

Training error at 100 is 0.07670805603265762
Epoch Number 100
Mean error 0.2491469
Standard deviation error 0.12097058
Threshold error 0.2975351244211197
Number of data points removed from the dataset at 100th epoch is 38

Now some data points are left in the dataset and that dataset is passed through the feed forward neural network and then following results are obtained:

Training error at 0 is 0.38308435678482056
Training error at 50 is 0.13775579631328583
Training error at 100 is 0.047838278114795685
Training error at 150 is 0.026347771286964417
Training error at 200 is 0.022755887359380722
Training error at 250 is 0.021498240530490875
Training error at 300 is 0.020419051870703697
Training error at 350 is 0.01939111016690731
Training error at 400 is 0.01840967684984207
Training error at 450 is 0.017472535371780396

Test loss: 0.018416

So, over here, we can infer that the BDR algorithm does really good when the BDR algorithm is applied before the train and test split of the data because the test loss as well as training loss is less than that of loss that we got when we used normal back propagation in the neural network.

Comparison

From the technique paper diagram and description of the dataset [2] as well as from the paper given with the dataset [3], we can infer that the dataset used for all three papers is same. We cannot compare much of the results as the technique paper as well as the dataset paper have solved the problem using the classification. The neural network regression model is chosen for this dataset because our priority is to predict the marks of the students not the grades as grades can be easily deduced from the system of grade marking provided in the technique paper. We can say that the BDR algorithm works good in the classification as well as regression. This is the commonality that both the report share.

Conclusion and Future Work

Initially, in the back propagation neural network, as our first epoch **(Fig 1.1)** showed that there are 2 peaks in the error graph. So, it is evident that the graph is bimodal as there are two modes in the error graph. So, BDR algorithm could be useful for this dataset. So, when BDR is applied on the training dataset, then training error as well as the testing error remains the same. But, when it is applied on the dataset before train test split, then it gives great results. So, at the end, I conclude that BDR algorithm should be used for the noisy datasets.

We can further extend this project by increasing size of the dataset as the dataset given to us is very small. Increasing the number of datapoints will also help the algorithm to learn more about the relationship between the input and output and moreover, it could happen that when we have huge dataset, we could get the training error and testing error difference in the starting only when we applied BDR algorithm on the training dataset.

References

[1] Dhairya Kumar, Introduction to Data Pre-processing in Machine Learning, Dec 9, 2019

[2] P.Slade, T.D.Gedeon , Bimodal Distribution Removal, June 01, 2005

[3] Edwin Che Yiu Choi and T.D.Gedeon, Compare the extracted Rules from Multiple Networks, August 06, 2002

[4] Raheel Shaikh, Feature selecting techniques in Machine Learning with Python, Oct 28, 2018

[5] Diederik P. Kingma, Jimmy Ba, Adam: A Method for Stochastic Optimisation, Dec 22, 2014

[6] Jason Brownlee, An introduction to feature selection, Oct 06, 2014