THE IMPLEMENTATION OF BIMODAL DISTRIBUTION REMOVAL ON NEURAL NETWORK AND EXPERIMENT ON ANGER DATASET

Yunyuan Yu

Research School of Computer Science,
Australian National University,
u6092441@anu.edu.au

Abstract. The quality of data can highly affect the performance of the back-propagation neural network. Outliers are one of the factors that can greatly reduce the accuracy of the neural network. This led to the implementation of many outlier detection methods which including Bimodal Distribution Removal (BDR). However, while dealing with a real-world dataset, the study on the effectiveness of BDR for the small dataset is lacking. This paper evaluates the effectiveness of BDR technique by compares the performance of the normal back-propagation neural network and normal back-propagation neural network with BDR. The original dataset is capable to use for training a neural network and achieve 95% of accuracy. The performance of BDR is measure upon a baseline model of neural network with 69.625% of accuracy using data with outliers. However, the result shows BDR does not increase the accuracy of neural network modeling with 67%~72% of accuracy after BDR perform outliers detection and removal. Given the ineffectiveness of BDR in this case, this paper also examined the effectiveness of BDR in terms of detecting outliers. The results show BDR is indeed effective in finding the outliers but it does not necessarily benefit the neural network training when the given dataset is small and the proportion of outliers is high.

Keywords: Outlier detection, bimodal distribution removal, neural networks, binary classification, small dataset, error measurement.

1 Introduction

Neural network modeling can be used to solve many classification problems. However, the performance of the neural network can be influenced by the presence of outliers in the dataset. Outliers usually can be seen as a subset of data which appears to be statistically inconsistent with the majority of the rest of the data [1]. A back propagation neural network is trained by adjusting the weight of each neurons base on the difference between network predicted output and actual desired output. Thus, the presence of outliers can greatly change the weight as the goal of the model is to minimize the sum of error of each pattern. [2] have shown that with 5 to 30% of outliers can statistically significant affect the training accuracy in term of a neural network modeling. Therefore, many outlier detection methodologies have been introduced and experiment by various researchers [3]. One of the preprocessing techniques for outlier detection is Bimodal Distribution Removal (BDR) based on the research by [4]. It removes the outliers using the distribution of the errors by identifying the high error peak during the training.

However, reviews have shown that there is no extensive study on the performance of BDR in a neural network with a small amount of data. BDR remove the outliers permanently during the training, therefore, it might become a problem as the removal of data within small dataset will have a higher impact compared to remove the same amount of data within a bigger dataset. There are many experiments with the general performance of BDR exists, but it is still lacking in the study of using BDR in a normal back propagation neural network with a little amount of data. In this work, the goal is to analyze the performance of BDR by examining the effectiveness of removing outliers in a real-world classification problem.

The data chosen for this work are from [5], the dataset consists the non-conscious responses from the testing candidate when they watched a short video that contains a person is either posed anger or real anger. The original work uses pupillary response patterns to predict the veracity of anger in the video. The original data have been tested and achieve 95% of accuracy when detecting the genuineness of the anger. In this research, data is extracted and computed into 400 rows, 6 valid features, and 1 label. Those data will be used in the training of the neural network to compete for the result of [5].

In this paper, I will first introduce the method I used, including the basic concept of how Bimodal Distribution Removal algorithm works, the baseline neural network model I am using. Next, I will discuss the process to determine the performance of BDR. Then, I will present an analysis of the results of the experiments and discuss the effectiveness of BDR with supporting evidence. Based on the research on the implementation of BDR in the neural network, I have also explored the usage of BDR in another two ways: examined the neural network performance after performing BDR on the training set data; examined the neural network performance after BDR on all data. Finally, I will have my conclusion and discuss the area of interest that can be researched further in the future.

2 Method

2.1 Pre-processing the Data

The very first task in this work is to reshape the data into the form that most suitable for neural network training. I choose to use Python and Panda data frame in this work as they are very user-friendly in transforming the data and easy to model a neural network. By looking at the data, the very last column “label” is removed and store at the predicted label as it contains the desired result of predict for each data pattern with either “Genuine” or “Posed” as value. It is translated into numbers to do neural network modeling, thus, assign 0 as "Genuine" and 1 as "Posed". Besides, the given data file contains 2 columns (Video number, ID) that are redundant because only pupillary response information will be used in this modeling, thus, it is removed from the data frame. On the other hand, each column except the label column has a different value range. However, all training data which should be normalized into the same scale in order for the neural network to treat them equally, in this paper, I will use the following formula for each column to scale them into a value between 0 to 1.

$$z_i = \frac{X_i - min(X)}{max(X) - min(X)} \quad . \tag{1}$$

Where X is the data column, X_i is the single data cell in data column X, and z_i is the resulting data cell. In all, after the data-pre-processing is done, and we have a 400 row of data with 6 features and 400 desired output labels. It will then randomly split the data into the train set and test set for train testing of the neural network.

2.2 Bimodal Distribution Removal

The idea of BDR is to perform outlier detection using the distribution of error during the training of the neural network. BDR is an effective approach in selecting the outliers during the training. However, if the initial error distribution during the first few hundred epochs does not form 2 peaks like in [4], it greatly increases the chance of overfitting as BDR runs when the model is still untrained.

During the experiment of using BDR in the neural network, one finding is that the choice of computation of error does affect the performance of BDR. In this is a particular binary classification problem, if the error is calculated by the absolute difference between prediction and actual, which is 1 minus predicted probability of choosing the desired output. The distribution of error can be very different from [4], instead, it forms one large peak in the distribution of error. This is because the error distribution of the initial neural network training is normally distributed. Therefore, the center distribution can be very close to 50% in the early stage and have a variance below 0.1(the trigger condition of BDR) which trigger BDR to detect the outliers. Since the normalized value of error can be very close to each other and this caused BDR incorrectly identify the outliers pattern. The experiment on BDR shows that it can hardly differentiate the outliers using the absolute difference between prediction and actual as shown in Figure 1.

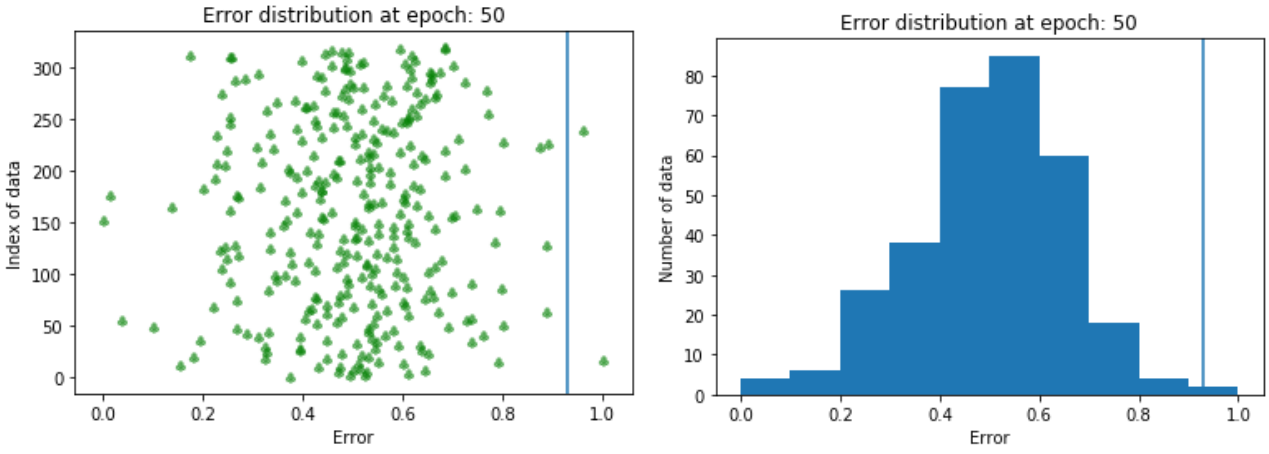

Fig. 1. Error distribution in scatter plot and histogram. Error is calculated by absolute differences between prediction and actual. Errors are normalized between 0 and 1 and the blue line is $\overline{\delta_{ss}} + \alpha\sigma_{ss}$ $(\alpha = 0.5)$ that used as the cut-off line to remove data patterns.

In Figure 1, the first run of BDR is at epoch 50 where the neural network is not well-trained. Although it has a normalized variance of error less than 0.1, it is not ready for outlier removal as the error form one cluster in the middle. Therefore, a better way to calculate the error and its variance is required as the BDR have to wait longer for the model to be well trained, and also know the difference between the correct prediction and incorrect prediction. Therefore, there are

two ways to change this, either create another condition such as wait until the epoch is greater than a certain number or change the error calculation. In order to increase the variance during the initial training and reduce the chance of such undesired removal to correct prediction data. I have come out with the following formula to calculate the error for each pattern:

$$error(x) = \begin{cases} -x \; if \; prediction(x) = \; True \\ x \; if \; prediction(x) = False.' \end{cases} . \tag{2}$$

Where x (>0.5) is chosen from the higher probability of the prediction each pattern produces for each output (either 1 or 0). Therefore, if the model prediction is true, the error for the pattern is a negative value between -0.5 to -1 and if the model prediction is false, the error is positive between 0.5 to 1.This allows the error rate to lie between -1 and 1 and then normalized into a range of 0 to 1. This leaves more space between correct prediction and incorrect prediction for the range of -0.5 and 0.5, it allows the initial variance of the error to be bigger than Figure 1 and also easier for BDR to differentiate the outliers. The result of the distribution of error using error calculation of formula 2 is shown in Figure 2 where there is a clear segmentation of correct prediction on the left peak and incorrect prediction on the right peak. Therefore, this ensures the BDR find only the outliers within the incorrect prediction.

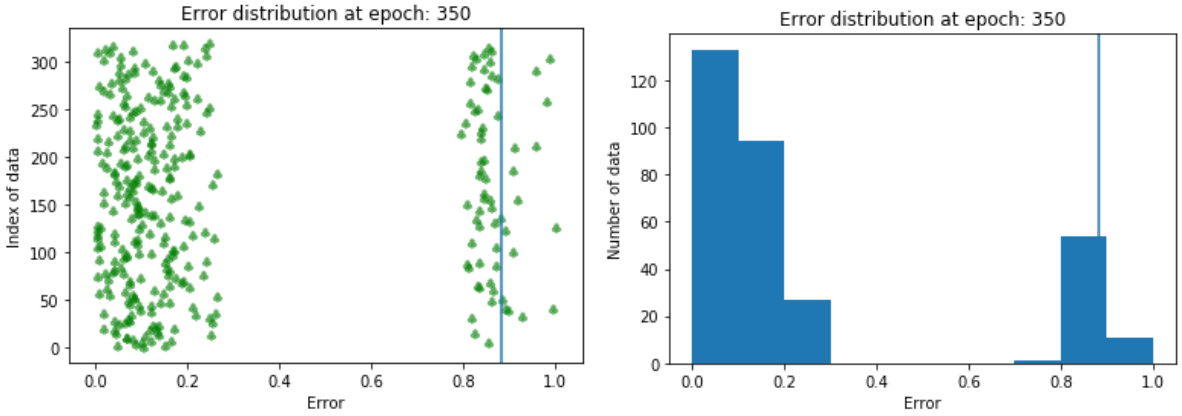

Fig. 2. Error distribution in scatter plot and histogram. Error is calculated by formula 2. Error is normalized between 0 and 1 and the blue line is $\overline{\delta_{ss}} + \alpha\sigma_{ss}$ $(\alpha = 0.5)$ that used as the cut-off line to remove data patterns.

The BDR operation waits until the neural network is generally well-trained which the normalized variance of errors fall under 0.1. This proves that the model has already learned something where the majority of patterns lies in the lower error peak of the distribution, but due to the presence of outliers, there is another error peak which consists the outliers and thus, those patterns shall be removed permanently from training. To calculate the outliers that need to be removed, following formal is used:

$$\overline{\delta_{ss}} + \alpha\sigma_{ss} \; (0 \leq \alpha \leq 1) \; . \tag{3}$$

The algorithm will compute the subset of data where its error is greater than the mean training data error. Then use the subset to calculate the mean error of subset $\overline{\delta_{ss}}$ and standard deviation of subset σ_{ss}. In this paper, α is chosen as 0.5 for all experiment by tuning α value and looking at the cut of the distribution. In figure 2 above, it shows the error distribution occurs at the first time of normalized variance of error fall under 0.1. The blue line is formula 2 where it represented the cut-off line to remove the data point on the right. This process then runs every 50 epoch and terminate the entire training process when the normalized variance of error falls under 0.01.

At each run of BDR, only a small number of outliers is removed from the high error peak which is the left group from Figure 2 of the distribution, and it moves the overall mean of the training data towards the low error peak of the distribution. This plot fits well in the argument from [4] as the neural network creates an almost bimodal error distribution, with the low error peak containing patterns the network has learned well, and the high error peak containing the outliers.

2.3 Back-Propagation Neural Network

When we experiment BDR performance in the neural network, we need to devise a neural network that reasonably works well but with plenty of room for improvement in order to test it with BDR. I have developed a simple three layers neural network with 6 input neurons for 6 input features in the input layers, a hidden layer with 20 neurons, and an output layers with 2 neurons that each represent one of the output labels. It is because the gradient of the Sigmoid function become flat on each side of the Y-axis. This means the network is not really learning because the gradient becomes close to zero. Generally, in a binary classification program, Sigmoid function is used as an activation function, however, the performance of the neural network is better when rectified linear unit (ReLU) is used after several experiments. Therefore, I choose ReLU as my activation function because it is also the common choice to deal with computer vision database on finding from [6]. The performance of each epoch is determined by the difference between the actual labels and prediction

outputs, and here I use the cross-entropy loss function. To do back-propagation, the neural network clears the gradients so that the gradient does not stack and accumulate and then perform backward pass to compute gradients of the loss with respect to all the learnable parameters of the model. Next, update the weights in hidden layers by tracking error backward using Adam optimizer as it is an efficient optimization that only requires little memory requirement [7]. Below are the initial parameters for the neural network:

- Input neurons: 6
- Hidden neurons: 20
- Output neurons: 2
- Learning rate: 0.005
- Number of epochs: 1000

2.4 Method of Evaluation

To evaluate the performance of the neural network on a classification problem, a confusion matrix is usually used [8]. However, given that it is a binary classification problem, the prediction outcome is either 1 or 0. The easier way to measure would be simply computing the accuracy score on the prediction on the test set. In this paper, we divide the dataset into 80% of train data and 20% of test data. Since the performance of the neural network can be varied by each run, we compute the average accuracy on N runs to measure the average performance.

The formula of calculating the accuracy:

$$\text{Accuracy} = \frac{1}{N}\sum_{n=1}^{N}\frac{Correct\_Predictoin_n}{Predictoin_n} = \frac{Ture\ Positive+True\ Negative}{True\ Positive+True\ Negative+False\ Positive+False\ Negative} \quad (4)$$

3 Results and Discussion

3.1 Initial Result for Normal Back-Propagation Neural Network

After running the normal back-propagation neural network on a randomly split training set using parameters discussed on 2.3. The average testing accuracy and training accuracy for each epoch are calculated among N=10 test run. In Figure 3, it is clear that the neural network does not learn much after 200 epochs and turns to be overfitting the training data after. However, the number of epoch in this experiment is still set to 1000 to ensure the number of BDR will run during the training of the neural network. The final evaluation score is 69.625%, calculated using the formula 4 in 2.4.

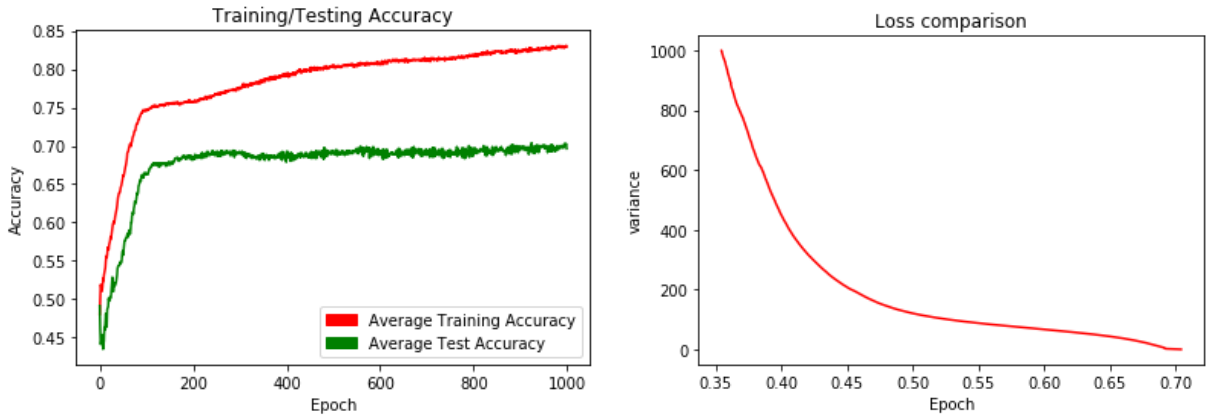

Fig. 3. The left plot shows the average training with testing accuracy. The right plot shows the average cross-entropy loss of running the normal back-propagation neural network

3.2 Bimodal Distribution Removal on Neural Networks

The data is split into testing data and training data, the neural networks will feed in the training data and calculate the accuracy score based on its prediction on the testing data. Due to the nature of BDR, as it terminates the training once the normalized variance of error falls under 0.01, it is not feasible to compute the average accuracy score and training score among 10 runs. Therefore, N=1 is chosen for multiple tests. Total 10 trials in done on the same neural network as 3.1, the final evaluation score is computed as a range within 67%-73%. The example used in the following presentation has an accuracy score of 71.25% using the formula 4 in 2.4 and 70 patterns were removed from train data.

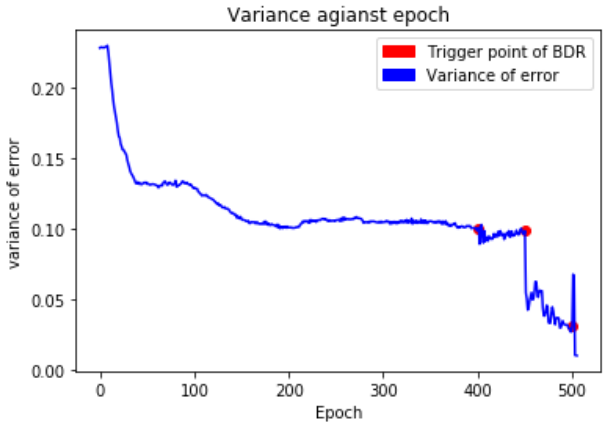

Fig. 4. The normalized variance of error for each epoch as the blue line and the redpoint represent trigger point of BDR where BDR identify and remove the outliers from the training set.

In Figure 4, it shows one of the tests runs to implement BDR in a neural network. Where the blue line plot the progression of the normalized variance of error and the redpoint presented as the trigger point of BDR. From the above plot, it is clear that the neural network progresses its learning by reducing the variance of error during first 200 epoch, this is similar to 3.1 as the normal back-propagation neural network greatly reduces the learning speed after 200 epoch. The iteration of BDR happened at 400, 450, and 500 epochs where the normalized variance of error fall under 0.1. This proves the [4]'s work as the process of BDR usually happened within 200-500 epochs. From each of the trigger point, we can see that BDR help to reduce the normalized variance of error when it removes the outliers of from the training set. Focusing on the trigger point at 450, the variance falls from 0.1 to below 0.05 which proved that the outlier contribute the most when calculating the normalized variance of error, therefore, it also shows that presence of outliers does affect the learning of neural network.

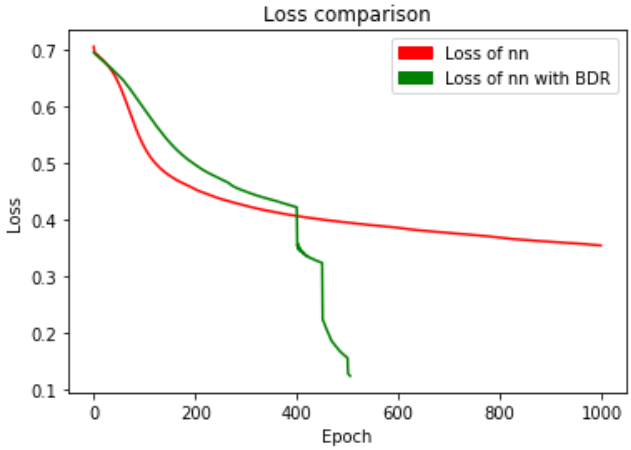

Fig. 5. Comparison of cross-entropy loss from run the normal neural network and with BDR.
Similar to Figure 4, the cross-entropy loss drops significantly after the outlier is removed from the training data. However, the cross-entropy loss is calculated using the training data and labels, thus, it might worsen the performance of the model in terms of accuracy score as the model become incapable of recognizing and predicting the patterns outliers from the test set.

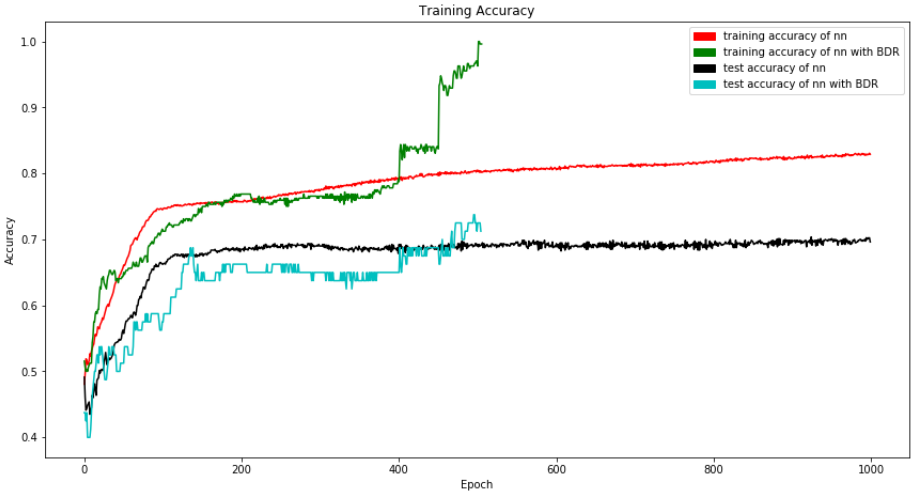

Fig. 6. Comparison of training accuracy and testing accuracy between normal neural network and neural network with the implementation of BDR.

Figure 6 compares the progression of training accuracy and testing accuracy of both normal neural network and neural network with BDR, it is unexpected that the training accuracy of BDR rise close to 100% while the testing accuracy is similar to 3.1 around 70%. After several runs on the neural network, the normalized variance of error and loss is reduced significantly but the accuracy score of each run does not show the effectiveness in terms of accuracy score. Generally, such implementation of BDR does not favor the performance of the neural network in this binary classification problem.

Table 2. The result of the neural network with BDR and normal neural network

Model	Accuracy
Normal Neural Network	69.625% (average)
Neural Network with BDR	67%-72%

This is possible because BDR removes the outliers from the training set and cause the neural network model to overfit the rest of the training data. However, it is known that the neural network learned more than 400 epoch using the original train data, therefore, the effectiveness of BDR is questioned since the removal of outliers might have less effect on the neural network because the reduced training dataset only has gone through less than 200 epochs. On the other hand, given that outliers can exist on both training data and testing data, it is also interesting to examine the result of running BDR on all data and use the new training and testing data into a new neural network model. By doing so, we can evaluate the capability of BDR to find the outliers. If the neural use the BDR-processed data as both train and test data, the resulting accuracy should be higher as the outliers are also removed from the test set.

3.2 Bimodal Distribution Removal on Training Data before Neural Networks

The hypothesis is that neural network had already learned about the training data set which include the outliers before the removal of BDR if we treat BDR as only a data preprocessing techniques, the result produces by feed to new training dataset into a new neural network is still unknown. In this stage, I will use the same mechanism as 3.1 except I have no interest in the testing accuracy and training accuracy. After the neural network with BDR training stop, I use the training data that left and feed them into a new neural network without BDR. Therefore, the new neural network will only learn from training data without the outliers. I will then calculate the testing accuracy from the new neural network. There are 2 possible outcomes from this experiment: the new network perform better than 3.1 neural networks with BDR; the new network perform no better than 3.1 neural networks with BDR. In the following experiment, 259 out of 320 training pattern is used for new neural network training.

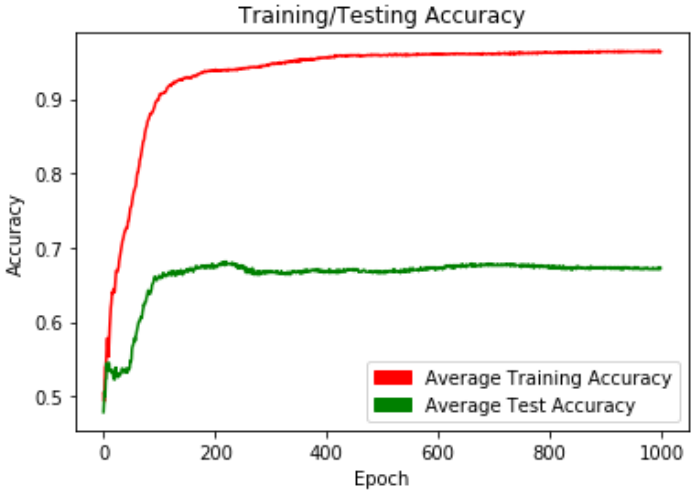

Fig. 7. The average training with testing accuracy.

Base on the comparison of Figure 7 and Figure 3, the result of this experiment is nearly identical to the normal neural network without BDR. It has an average test accuracy of 67.25% which is even worse than the above experiment. However, given that the training dataset is shortened by the previous BDR operation, it is still considered an acceptable result. This answer my previous question on whether the effect of remove outliers will have a bigger impact when applying the new dataset into a new model. This experiment shows that the outliers found by BDR are effective as they are redundant for the neural network training, the removal of those patterns does not affect the test accuracy very much.

3.3 Bimodal Distribution Removal on Training and Testing Data before Neural Networks

From the result of 3.1, the neural network model is overfitting the training data. However, the operation of remove outliers is only done with training data. It is an intuition to assume the outliers exist in testing data. The hypothesis is that if the BDR is used with both training and testing data. The neural network model should have a higher accuracy score using the resulting dataset. As above, the set up for the neural network model and training is identical from 3.1 except all data are considered as training data. It is also can be seen as a data pre-processing technique to remove the outliers from both training and testing data. In the following experiment, 321 out of 400 patterns is left for new neural network training and testing. 256 patterns were used in training the neural network.

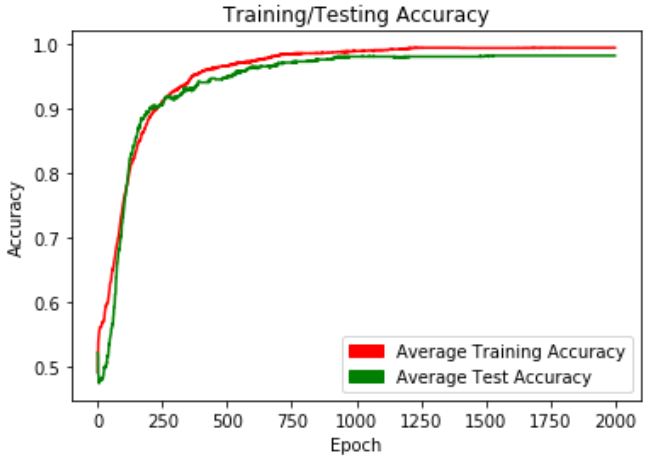

Fig. 8. The average training with testing accuracy.

Base on the comparison of Figure 8, the result is very surprising as it achieved an average accuracy of 98.4615% when predicting the test data. However, consider this result is produced from only 80% of the total data, it is not feasible to compare this result with [5]. By removing the outliers and use a same neural network to achieve a much higher result, this can be further proving the ability of BDR in selecting outliers, and the improvement it can produce for the neural network if the outliers are excluded from all data. Last but not least, this result of the experiment should not be used as the accuracy of the trained model as it is biased and only learn and test in a smaller size of the dataset.

4 Limitation

There are many factors that affect the performance of the neural network and also BDR. Outliers detection can be hard to measure, therefore, more outlier detection method should be deployed and test in order to compare with the outcome of BDR. Given the amount of outliers BDR found are relatively around 25%, it is considered a very large amount of data. Thus, it is possible that part of the patterns identified by BDR as outliers are not outliers, instead, they have their own behavior and isolated because they are hard to train for the model.[3] have proposed various outliers detections method, and comparison of the number of outliers detected can be used to examine the effectiveness of BDR on such small dataset.

Moreover, the parameters of the neural network can also affect the performance of BDR, in this paper, only one set of parameters of the neural network is used. However, there can be more explore and comparison between different kind of neural network and how BDR works with a different kind of neural network.

5 Conclusion and Future Work

In summary, my study shows that BDR does not benefit the neural network training with a small dataset for a binary classification problem. In this paper, I examined the performance of the neural network with the implementation of BDR to solve a binary classification problem. The dataset consists of 400 rows of data with 6 features and 1 label. The formula used for error measurement is design specially for this task in formula (2). The normal back-propagation neural network achieves an average accuracy of 69.625% and the neural network with the implementation of BDR has accuracy float between 67% and 72%. This shows the implementation of BDR does not benefit the training of a neural model. However, BDR is still effective in selecting outliers in general base on the experiment done in 3.2, 3.3.

Compare to the result from [5] where they achieve 95% of the accuracy of the input data is not feasible as it is unknown to their data structure and machine learning techniques. In the future, more works should be done on this same problem using different parameters, it is reasonable to investigate the inference of various parameters. The normal back-propagation neural network can be designed and tuned using a different setting. In addition, in BDR, the α value is set to 0.5 for all experiment, it is still unknown for different value of α affect the final outcome of BDR.

References:

1. 4. F. E. Grubbs, “Procedures for detection outlying observations in samples,” Technometrics 11(1), 1–21 (1969).
2. Azme Khamis, Zuhaimy Ismail, Khalid Haron, and Ahmad Tarmizi Mohammed. “The Effects of Outliers Data on Neural Network Performance”. Journal of Applied Sciences, 5: 1394-1398 (2005)
3. D. Coursineau, “Outliers detection and treatment: a review,” Int. J. Psychol. Res. 3(1), 58–67 (2010).
4. Slade, P. and Gedeon, T.D., 1993, June. “Bimodal distribution removal.” In International Workshop on Artificial Neural Networks (pp. 249-254). Springer, Berlin, Heidelberg.
5. Chen, L., Gedeon, T., Hossain, M. Z., & Caldwell, S. (2017, November). : “Are you really angry?: detecting emotion veracity as a proposed tool for interaction.” In Proceedings of the 29th Australian Conference on Computer-Human Interaction (pp. 412-416). ACM.
6. Caglar Gulcehre, Marcin Moczulski, Misha Denil, Yoshua Bengio, “Noisy Activation Functions,” Proceedings of Machine Learning Research Volume 48: International Conference on Machine Learning, 20-22 June 2016, New York, New York, USA
7. Kingma, D. and Ba, J. (2015). “ADAM: A Method for Stochastic Optimization.” 3rd International Conference for Learning Representations, San Diego, 2015
8. J. T. Townsend. (1971). "Theoretical analysis of an alphabetic confusion matrix." In: Attention, Perception, & Psychophysics, 9(1).