The implementation of the BDR technique and the Recurrent Neural Network in the Human Eye Gaze Pattern Recognition

Yihui Zheng
Research School of Computer Science, Australian National University
U5937192@anu.edu.au

Abstract. Neural Network is wild used in different classification tasks. In the first part of the report, a simple two-layer network with a cross-validation accuracy of 94.71% is constructed, then a data processing technique Bimodal Distribution Removal (BDR) will be applied to a neural network to remove outliers and improve the performance of the simple neural network. The results obtained that the original network can achieve an average training accuracy of 86.13% and a testing accuracy of 83.66%. After implementing the BDR technique, although the average training accuracy can be increased to 94.37, there is not an obvious improvement in the testing accuracy. It can be anticipated that the BDR can help to remove outliers, but it does not improve the learning of this network. In the second part of the report, a recurrent neural network (RNN) with a cross-validation accuracy of 88.24% is constructed. As a result, the RNN can achieve an average training accuracy of 90.27% and a testing accuracy of 89.53%. It can be concluded that, in this project, the RNN model has a better performance than the simple neural network. For future work, a larger dataset could be used to enhance the training process and early stop command could be used to optimise the BDR processes and we could add dropout layers or use specific regulation to improve the RNN model.

Keywords: Simple Neural Network, Bimodal Distribution Removal, Recurrent Neural Network, cross-validation, training accuracy, testing accuracy

1. Introduction

As an intelligent way to solve problems such as time series predictions, anomaly detection in data, and natural language understanding [1], neural networks are wild used in many fields. Based on the research data from the project: *A Hybrid Fuzzy Approach for Human Eye Gaze Pattern Recognition* [2], different neural networks can be designed and used to predict whether the vertical distance between fixations is recognised. In this project, there are two given datasets with 169 lines and 154 lines respectively and they will be used as the training set and testing set.

The project will start by setting up a simple two-layer neural network with backpropagation to fitful the basic requirements of recognition prediction. However, if there is not any technique to optimise the dataset and remove outliers, the accuracy of the model will be highly influenced. From the research *The Effects of Outliers Data on Neural Network Performance* [2], it can be obtained that the percentage outliers and the magnitude of outliers will both significantly influence the overall accuracy of the network model. Therefore, the Bimodal Distribution Removal Technique (BDR) is chosen to modify the original dataset and aim to improve the performance of the simple neural network model.

After that, the neural network in this project can also be upgraded by applying deep learning methods. According to the research *Learning to Diagnose with LSTM Recurrent Neural Networks* [7], the recurrent neural network can help to check whether an illness is correctly detected. In the *Human Eye Gaze Pattern Recognition*, we need to find that whether the vertical distance between fixations is recognised and the task is similar to illness diagnosis, which means RNN can also be constructed to achieve the pattern recognition.

In order to illustrate the difference between the performance of the original simple neural network, the models which are optimised by the BDR technique, and the RNN model, several comparisons will be made such as error distribution, loss, training and testing accuracy. Then, a conclusion will be drawn to determine whether the BDR technique and the RNN are useful or not. Besides, there will be some potential future developments, these will also be covered at the end.

2. Methodology

2.1 Data Pre-processing

By extracting the data from the original training and testing datasets, it can be obtained that all these 4 input values represent the distance between fixations and share the same unit; therefore, they will have the same dimension and there is no need to do further data standardization nor normalisation. Then, by checking the correlation of the dataset, it can be found that the dataset is balanced and there is no need to do further data alleviation. Besides, it is still necessary to decompose the training set for validation purposes. In this project, the 'KFold' command will be imported from the sklearn.model_selection and with the help of it, the validation datasets can be produced.

2.2 Simple Neural Network Construction

At first, a two-layer simple neural network will be constructed, since it is able to fulfil the need for prediction to a large extend. If a too complicated network were used at the beginning, it will face the problem of challenging optimization [4] and it will restrict the improvement methods. Based on the given datasets, the basic parameters can be set up. There will be 4 input neurons, 10 hidden neurons and 2 output neurons with the learning rate 0.01, and there will be 1000 epochs.

According to the dataset, it can be found that the prediction values are '0' and '1' which represents 'not recognised' and 'recognised'. The purpose of the classification is similar to a logistic regression problem; therefore, the Sigmoid function will be used as the activation function since the Sigmoid function is wildly used in binary classification. For the loss function, the cross-entropy loss function will be used. In order to do back-propagation, the function 'zero_grad' will be used first to clear the gradients, then the 'backpass' function could be applied. After that, the SGD optimiser will be used to make updates based on the gradient descent.

2.3 Bimodal Distribution Removal (BDR)

To apply the Bimodal Distribution Removal technique, it is important to find out whether there is an existing bimodal distribution in the error distribution. The errors of the training set can be computed by the absolute value of the difference between the actual values and the predicted values. When the normalised variance of the error is in the range from 0.01 to 0.1, the error distribution plot can be found as:

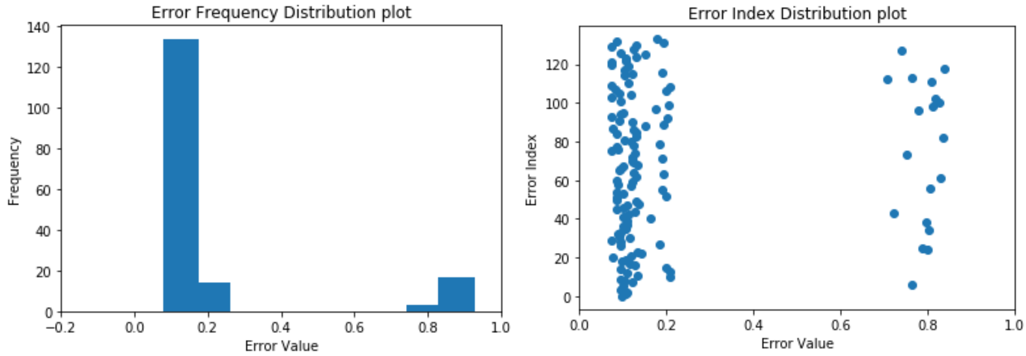

Fig.1 The error distribution from the original model

From figure 1, it is obvious that there is a bimodal distribution in the error distribution, because there are two distinct local maxima, one is around 0.15 and another is around 0.85. Based on that, the BDR could be applied in the model. According to the BDR technique introduction paper [5], after the normalised variance of the error is below 0.1, the mean error $\overline{\delta_{ts}}$ will be computed and all the error which is greater than $\overline{\delta_{ts}}$ will be taken to form a skewed subset. From the skewed subset, the mean $\overline{\delta_{ss}}$ and standard deviation σ_{ss} can be calculated, then there will be a threshold value $\overline{\delta_{ss}} + \alpha\sigma_{ss}$ $(0 \leq \alpha \leq 1)$. In this project, the value of α will be set as 0.5. The next step is to remove any patterns from the training set whose error is greater than $\overline{\delta_{ss}} + \alpha\sigma_{ss}$ ($0 \leq \alpha \leq 1$). This process will be repeated every 50 epochs until the normalised variance of the error is less than 0.01.

2.4 Recurrent Neural Network Construction (RNN)

Different from the simple neural network, there are connections between different hidden layers, according to the research *Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network* [8], the inner features of a recurrent neural network such as control accumulation nodes will have a great influence to the scaling of the data signals which means we cannot just use a basic sigmoid function to calculate the values. By import the built-in function of the recurrent neural network from PyTorch, we can set up a one-layer recurrent neural network by customising the basic parameters such as the input size, hidden size, and batch size. There are four different inputs, so the input size is set to be 4. In order to compare with the original model, we will set 1000 epochs to RNN to make sure the model is trained with the same number of epochs.

For the loss function, the cross-entropy loss function will also be used. However, for the optimiser, instead of using the SGD optimiser, the Adam optimiser is chosen, that is because in the recurrent neural network the gradient descent is changing more frequently because the new values depend on more connections between different variables compared to simple neural networks. According to the research *Adam: A Method for Stochastic Optimization* [9], SGD optimiser has the right direction, but it has a lower speed for convergence. On the contrary, Adam optimiser has a higher convergence speed and can make all the iterations more stable.

2.5 Cross Validation and Accuracy Measurement

As mentioned in the section 2.1, the 10-fold cross-validation will be used to validate the simple neural network model and the RNN model. The training set will be divided into 10 subsets. In the process of validation, the model will be trained ten times, and for each time, one of the 10 subsets will be used as the validation testing set and the other 9 subsets will be combined and used as the training set. After testing, the average testing accuracy will be computed and the value of it will point out whether the model is fit for the task and whether there is a problem of overfitting. The accuracy will be calculated by the number of the correct predictions divided by the number of the total predictions.

3. Results and Analysis

3.1 Cross Validation Results

The results of the 10-fold cross validation for models are shown as below:

Iteration Number	1	2	3	4	5
Test Accuracy	100%	100%	70.59%	100%	88.24%
Iteration Number	6	7	8	9	10
Test Accuracy	100%	100%	88.24%	100%	100%

Table.1 The accuracy results of 10-fold cross validation for the original simple neural network

Based on the result of testing accuracy in each iteration, the average testing accuracy can be calculated as 94.71% which is high enough and is able to prove that the original model is appropriate and can be used for further training purposes. Besides, it can be found that the difference between each training set is small which means the original model is less likely to face overfitting problems.

Iteration Number	1	2	3	4	5
Test Accuracy	100%	100%	70.59%	100%	70.59%
Iteration Number	6	7	8	9	10
Test Accuracy	100%	70.59%	70.59%	100%	100%

Table.2 The accuracy results of 10-fold cross validation for the recurrent neural network

Based on the result of testing accuracy in each iteration, the average testing accuracy can be calculated as 88.24% which is able to prove that the RNN model is appropriate and can be used for further training purposes. However, compared to the original model, there will be a higher possibility for the RNN model to face the overfitting problems since there is a bigger gap between each training set.

3.2 The Performance of the Original Simple Neural Network

After running the original simple neural network, the training accuracy and loss plot can be made:

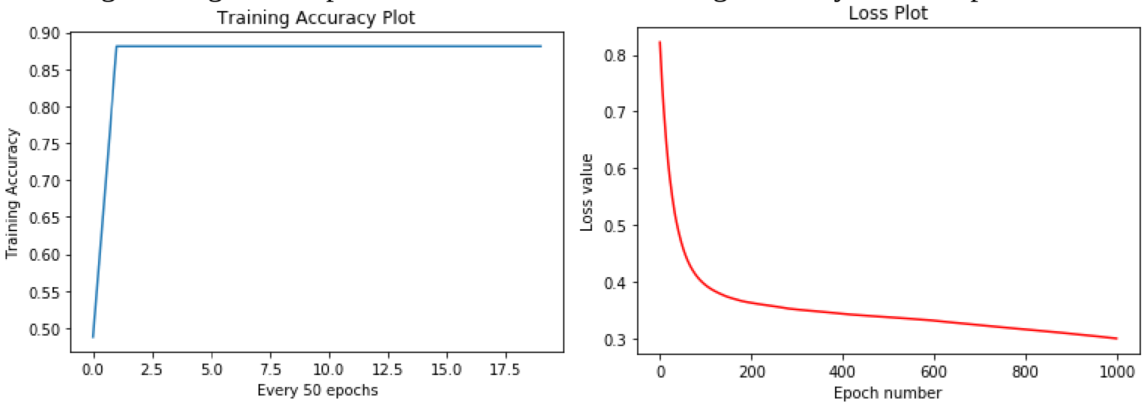

Fig.2 The Training Accuracy and Loss Plot for the Original Neural Network

From figure 2, it can be found that, at the beginning, the training accuracy is really low which is only about 48%, and after training it for 50 epochs the accuracy began to increase and converge to around 88 %. According to the cross-entropy loss, the loss of the model is keeping decreasing, and in the end the loss is decreased to 0.2835. Then, run the python code for 10 times, based on the average training accuracy and testing accuracy, the performance table can be drawn:

Neural Network Type	Average Training Accuracy	Average Testing Accuracy
Without BDR	86.13%	83.66%

Table.3 The Performance table for the Original Neural Network

The testing accuracy is not bad which means the original neural network did a good job, and for the next step, the BDR technique can be applied to find out whether it can have a positive influence on the training or the testing accuracy.

3.3 The Performance of the Neural Network with Bimodal Distribution Removal (BDR)

After running the neural network with BDR, the training accuracy and loss plot can be made:

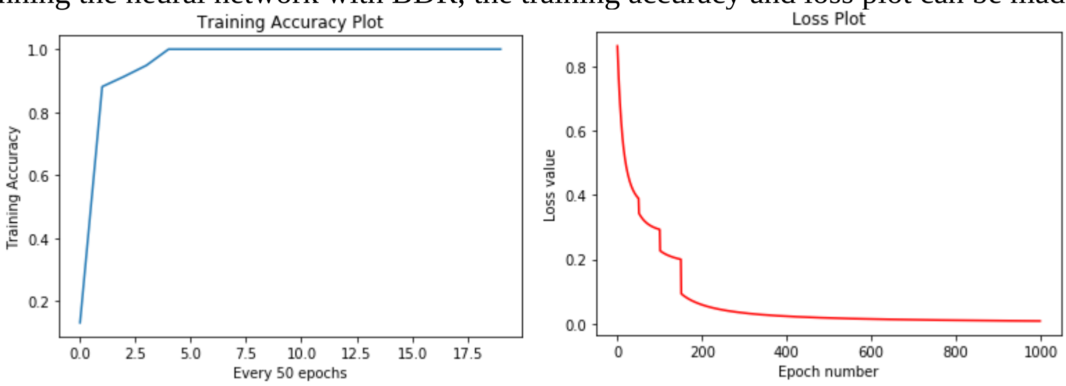

Fig.3 The Training Accuracy and Loss Plot for the Neural Network with BDR

From figure 3, it can be found the training accuracy is increased in different stages, in the first 50 epochs, the training accuracy is increased from 13.10% to 88.10%, then from epoch 51 to 101, the accuracy increased to 91.36% and in the next 50 epochs, the accuracy increased to 94.87%, and after that, the accuracy finically reached 100%. Because all the patterns with errors greater than $\overline{\delta_{ss}} + \alpha\sigma_{ss}$ are removed from the training set and they are actually the patterns which have the wrong prediction (should be 0, but predicted 1), so after bimodal distribution removal, only the correct predicted patterns are kept in the training set and then the training accuracy will finally converge to 100%. This process can also be shown in the loss plot. There are several stair-stepping parts that can represent the process of BDR. For every stair-stepping part, the skewed subset will be renewed and the threshold value $\overline{\delta_{ss}} + \alpha\sigma_{ss}$ $(0 \leq \alpha \leq 1)$ will be recalculated and some patterns will be removed. It can also be found that after 200 epochs, there is no more stair-stepping part which means all the 'outiler' patterns have been removed and only the correct patterns are left. Besides, compared with figure 2, the loss after applying BDR can be decreased to a smaller value which is around 0.0081. It can be evidence that the BDR helps to reduce loss during the training period.

The error distribution plot after using BDR can prove that as well:

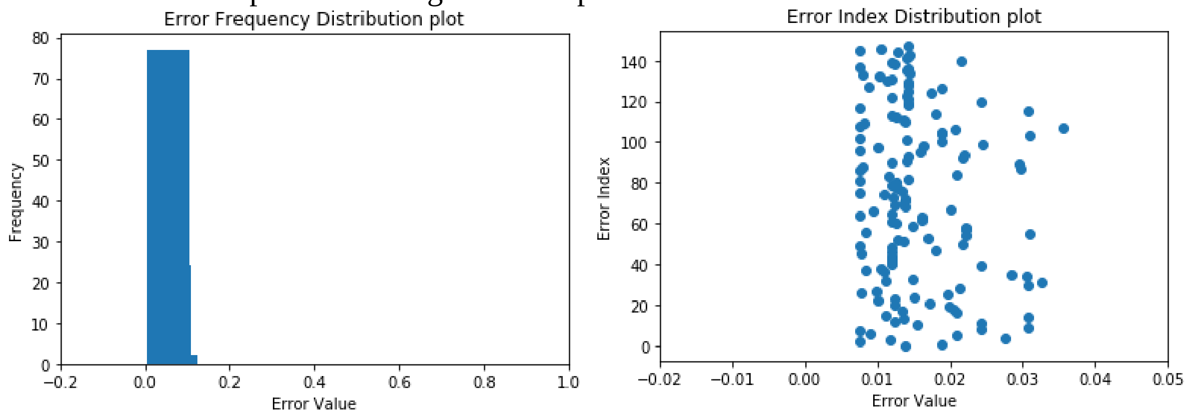

Fig.4 The error distribution after using BDR

with figure 1, the errors at the right-hand side (the local maxima around 0.8) were removed and there are no more obvious bimodal distribution problems. Besides, the remained patterns just had small errors and most of them were evenly distributed in a small range. Then, run the python code for 10 times, based on the average training accuracy and testing accuracy, the performance table of the Neural Network with BDR can be drawn:

Neural Network Type	Average Training Accuracy	Average Testing Accuracy
With BDR	94.37%	83.84%

Table.4 The Performance table for the Neural Network with BDR

By comparing table 3 and table 4, it can be found that the BDR technique can significantly increase the training accuracy and reduce the loss; however, it cannot efficiently increase the testing accuracy of the model. In other words, although a high training accuracy is achieved, the model was not trained better. One hypothesis could be that the BDR is only applied to the training dataset but not to the test dataset which means the bimodal distribution may still exist in the test dataset. When the test dataset is used to test the model, the model still cannot adjust its prediction. One possible solution is that the bimodal distribution can be applied to both training and testing set and there will be an increase in the testing accuracy after using the pre-processed testing set.

Another hypothesis could be that the provided dataset is not abundant enough and the number of patterns in the skewed subset is small which means the removal process just be repeated for several times and it is not enough for the model to be familiar with the removal process so the model cannot provide correct action to the test data. Besides, based on the paper *Review on Methods to Fix Number of Hidden Neurons in Neural Networks*[6], the number of hidden neurons may also have an influence on the results, if the fixed number hidden neurons can be set to a more proper value, there will be a training process with higher quality and if optimiser is changed from SGD to Adam there will also be a change on the accuracy.

3.4 The Performance of the Recurrent Neural Network (RNN)

After running the original neural network, the training accuracy and loss plot can be made:

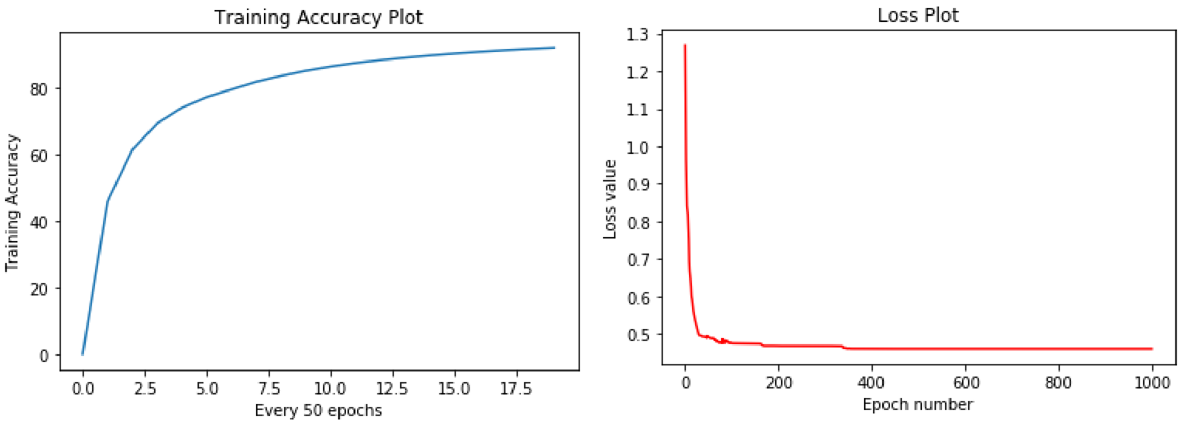

Fig.5 The Training Accuracy and Loss Plot for the Recurrent Neural Network

Compared with figure 2 and 3, it can be found that the training accuracy of the RNN is increasing in a steady and stable trend. In the first 500 epochs, there is a big increase in the training accuracy from 0% to about 86%, this is because at this period, the degree of change in weights will be big and the model will be adjusted substantially. In the last 500 epochs, the increasing speed becomes slow just from 86% to 91 % and the training accuracy will finally converge. The reason is that after the learning of the first 500 epochs, the weights have already been updated to a large extend, the degree of change will become smaller which lead to small improvement in training accuracy. According to the cross-entropy loss, the loss of the model is keeping decreasing from 1.291 to 0.4417. Run the python code for 10 times, based on the average training accuracy and testing accuracy, the performance table of the recurrent neural network can be drawn:

Neural Network Type	Average Training Accuracy	Average Testing Accuracy
RNN	90.27%	89.53%

Table.5 The Performance table for the Recurrent Neural Network

By comparing with table 3 and table 5, it can be found that the recurrent neural network has a higher training accuracy and testing accuracy than the simple two-layer neural work, there are around 4% and 6% improving respectively. By comparing table 4 and table 5, it can be found that: although the recurrent neural network has a lower training accuracy, it can achieve a higher testing accuracy than the network model with BDR technique and there are around 6% increasing in the testing accuracy. The cause of lower training accuracy (compared with the model with the BDR technique) is that, without an outlier removal technique, the outliers still exist in the training and testing datasets for the RNN model and they will affect the training accuracy. In a word, based on the given datasets from the *Human Eye Gaze Pattern Recognition* [2], the RNN model performances better than the simple neural network model and the model with the BDR technique.

However, there are some drawbacks to the RNN. Firstly, based on the cross-entropy loss and results of cross-validation, it can be found that the RNN has a bigger loss than the other two models and this might be caused by the outliers in the datasets (which is mentioned above) and the potential overfitting issues. Secondly, from the research *Combination of Deep Recurrent Neural Networks and Conditional Random Fields for Extracting Adverse Drug Reactions from User Reviews* [10], the recurrent neural network has a disadvantage that there is a vanishing gradient problem, in other words, it cannot process very long data sequences since 'tanh' function is used as it activation functions. If another long dataset is used, the RNN may not be able to handle it. Last but not least, it will cost more time to train a RNN model, for the original model and the model with the BDR technique, the training time will be several seconds but for the RNN model it will cost up to twenty seconds. For further improvement, some methods can be applied to address these issues, and they can be shown in the future work section below.

3.5 Comparison with the results in the research paper

The training and testing results from the research: *A Hybrid Fuzzy Approach for Human Eye Gaze Pattern Recognition* [2] is 80.23% and 80 % respectively with a mean square error of around 0.06. The original neural network in this project has a similar performance with the research's results; for the neural network with BDR, it has a higher training accuracy but there is still not a big difference in the testing accuracy; and for the RNN, it can achieve a higher training accuracy and a higher testing accuracy at the same time. The BDR technique is able to identify the possible outliers, but it is not efficient in improving the model's performance and it cannot achieve better test accuracy. Although the RNN has a slightly higher loss, it can successfully improve the prediction results and achieve better accuracy.

4. Conclusion

In conclusion, this project shows that the bimodal distribution removal technique does not have an obvious effect on the neural network training, but by using the recurrent neural network there will be a better prediction for the *Human Eye Gaze Pattern Recognition* task. In this project, with a given training set and testing set, three different neural networks (a simple neural network, a simple neural network with BDR technique and a recurrent neural network) are trained and tested. At the beginning of the project, 10-fold cross-validation is used. Based on the average validation accuracy of 94.71% and 88.24%, it can be proved that the two-layer network model and the recurrent neural network are appropriately constructed. Then after the training and testing process, the neural network without BDR and the neural network with BDR can achieve a testing accuracy of 83.66% and 83.84% respectively. Although the model with BDR has a relatively higher training accuracy which is around 94.37%, there is not a big difference in testing accuracy after applying the BDR technique which shows that the BDR does not improve the performance of this neural network. However, with an average training accuracy of 90.27% and an average testing accuracy of 89.53 %, the recurrent neural network has the best performance within these three models and further improvements can be applied to the RNN to reduce the loss and prevent potential overfitting issues. Therefore, the recurrent neural network can be a good potential choice for further research.

5. Future Work

To improve the model with the BDR technique: Firstly, for the original network itself, more layers could be added to enhance the training process and allow more chances for backpropagation. The number of hidden neurons could be increased to improve the efficiency of learning because in this project there are only 10 hidden neurons, some predictions are not good enough. Secondly, for the dataset, a larger dataset could be provided to train the network; therefore, the BDR process could be repeated more times to benefit the model training. Furthermore, for the implementation of the BDR technique, the value of the parameter α could be adjusted several times to find its influence on the training results. Besides, in order to make sure the BDR will not remove too many patterns and avoid removing some important patterns accidentally, an early stop command could be added into the model to protect the effectiveness of the remaining patterns.

To improve the RNN model: Firstly, in this project, we only use the one-layer recurrent neural network. In the future, we could use RNN with more layers and the parameters such as batch size and learning rate could be adjusted to find out whether there will be a large impact. Secondly, as mentioned in Section 2.4, in a recurrent neural network, the connection between layers will have a great influence on the final results and there might be too much data produced by the inner connections. Therefore, to simplify the model and prevent potential overfitting issues, we could add dropout layers into the model or use specific regulation. Besides, to increase the learning speed, the layer normalization method can also be applied to the recurrent neural network.

Reference

1. Shah, J., 2017. Neural Networks for Beginners: Popular Types and Applications. [online] Available at: <https://blog.statsbot.co/neural-networks-for-beginners-d99f2235efca>

2. Zhu, D., Mendis, B. S. U., Gedeon, T., Asthana, A., & Goecke, R. (2008, November). A hybrid fuzzy approach for human eye gaze pattern recognition. In International Conference on Neural Information Processing (pp. 655-662). Springer, Berlin, Heidelberg.

3. Azme Khamis, Zuhaimy Ismail, Khalid Haron and Ahmad Tarmizi Mohammed, 2005. The Effects of Outliers Data on Neural Network Performance. Journal of Applied Sciences, 5: 1394-1398.

4. Brownlee, J., 2019. Why Training A Neural Network Is Hard. [online] Machine Learning Mastery. Available at: <https://machinelearningmastery.com/why-training-a-neural-network-is-hard/>

5. Slade, P. and Gedeon, T., 1993. Bimodal distribution removal. New Trends in Neural Computation, pp.249-254.

6. Sheela, K. and Deepa, S., 2013. Review on Methods to Fix Number of Hidden Neurons in Neural Networks. Mathematical Problems in Engineering, 2013, pp.1-11.

7. Lipton, Z., Kale, D., Elkan, C. and Wetzel, R., 2015. Learning To Diagnose With LSTM Recurrent Neural Networks. [online] arXiv.org. Available at: <https://arxiv.org/abs/1511.03677>

8. Sherstinsky, A., 2020. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, p.132306.

9. Kingma, D. and Ba, J., 2015. Adam: A Method For Stochastic Optimization. [online] arXiv.org. Available at: <https://arxiv.org/abs/1412.6980>

10. Tutubalina, E. and Nikolenko, S., 2017. Combination of Deep Recurrent Neural Networks and Conditional Random Fields for Extracting Adverse Drug Reactions from User Reviews. Journal of Healthcare Engineering, 2017, pp.1-9.