Classification of Drugs by Feed-Forward Neural Network with the Implementation of Bimodal Detection Removal and Genetic Algorithm

Haohan Lian
Research School of Computer Science, Australian National University
u6022863@anu.edu.au

Abstract

This paper will focus on the classification on the Musk (Version 1) dataset to determine whether a molecule is musk or non-musk. To achieve this, the method used in this paper is a three layers feed-forward neural network based on the back-propagation algorithm with different kinds of other parameters. Some optimization techniques are used to improve the performance of the neural network, namely Genetic Algorithm (GA), Bimodal Detection Removal (BDR) and K-Fold cross validation. According to the result, neural network with BDR can achieve 90.4% accuracy on average while neural network with both BDR and GA can achieve 91.2% accuracy. The results in this paper are slightly higher than APR in relevant paper, which is 86.6%. [4].

Keywords: neural network, accuracy, Back-propagation, Genetic algorithm, Bimodal Detection Removal.

Introduction

1. The source of the data

The dataset used in this paper is chosen from UCI Machine Learning Repository, named Musk (Version 1) Data Set. It is created by AI Group at Arris Pharmaceutical Corporation. As for its initial motivation, it is used to predict the drug activity [2].

2. The description of the data

Most drugs are small molecules and they need to relate to some extremely large protein molecules. Some new molecules can be predicted by activity prediction. In order to achieve this goal, it is necessary to analyse the dataset which contains the training example of previously synthesised molecules as well as their activities when they are relevant to some medical interest. So that analysing and observation would take plenty of cost and time for chemists to synthesising the most promising candidate molecules due to the special nature of small molecules. However, the prediction of the molecules can be done with the help of neural network, by which time and expenditure can be saved. Meanwhile, with the help of the prediction of neural network, chemists can design better drugs because the shape of the binding site is hard to analyse for human beings. The neural network can make it much easier to predict the three-dimensional shape to support the design of the new medicine. Some approaches to avoid multi-instance problem [4]. One is that a representation which has no relationship with the changes in bond angle can be employed. The other one is that a representation which is shape-oriented can also be employed.

There are 168 attributes in dataset, each column of them represents a single feature of the molecules. The first two columns represent the symbolic name of each molecule and molecule number. While the attribute 1 to attribute 162 represent the distance features along rays. Moreover, the rest four attribute demonstrate the distance and displacement of oxygen atom in the molecule which is also called OXY_DIS [4]. The molecules in the dataset are processed by low-energy conformations, which left 476 molecules. So the instances in the dataset are based on the number of molecules. Obviously, there is only one column of target.

3. Model the problem

The reason of using this dataset is because it is a real world dataset with only 2 labels and the result can be easy to be measured and analysis. In order to solve this classification problem, the 166 attributes are regarding as the input neuron of the neural network. The first two columns which is not relevant to the classification model is removed from the dataset. Besides, the label is treated as the output neuron, which has only two values, 0 for non-musks and 1 for musks. The goal of the neural network model is to predict whether new molecules will be musks or not by analysing the given dataset. Meanwhile, it is also important to investigate the more proper model for this problem in order to gain an ideal prediction.

4. General description

Genetic algorithm is used to find the best combination of the features which could possibly improve the performance of the classification. After that, the neural network will implement Bimodal Detection Removal to further remove the feature that is not suitable for the classification. Then some comparisons and analysis of the results will be stated. Furthermore, some conclusions, limitations of these methods and some future works will be discussed in the final sections. Accuracy is regarded as the standard for measuring the performance of the neural network, which will be demonstrated in the form of figures and tables.

Method

1. Basic pre-process

Pre-processing the data is very common in the analysis, which can save a lot of time or expense. Before training the neural network, the dataset needs to be pre-processed. The format of the dataset needs to be changed as '.csv' profile for better analysing because the original format is not clear enough. Then, according to the description of the dataset, the first two columns need to be removed because they have nothing to do with the classification.

After reading the dataset, normalisation of data is also of vital importance because it can balance the weight of each column so that there is no column with a stronger significant influence than others. Besides, it is better to shuffle the data so that the training set will be in a random sequence, which is better for evaluate the training model. Furthermore, training set and test set are both important in the multi-instances classification. However, there is only one dataset so that the original data needs to be separated into two parts, one is training set and the other one is test set. In this paper, the proportion of the training set and test set is defined as 8:2, in other words, the training set takes 80% of the total dataset while test set takes 20%. Furthermore, the training set is divided into training set and validation set, which is used for the K-Fold validation in the following steps. After that, the input and the output need to be clarified before starting training and testing. This can be easily achieved by split the data according to its features.

2. Neural Network Model

As for the model in this paper, we assume a three layers feed-forward neural network trained using back-propagation, which is implemented with the hyper parameters pre-defined, which contains hidden neuron, learning rate and so on. To be more specific, the neural network includes an input layer, a hidden layer as well as an output layer, which are fully connected to each other.

Moreover, the inputs and outputs need to be wrapped before training due to the reason that Torch only trains neural network on Variables. Furthermore, a sigmoid activation function is used from the input neuron to the output neuron. After training, the output should be transformed into one column for comparison. The reason why sigmoid function is used is that it satisfies a property between the derivative and itself such that it is computationally easy to perform and it can turn the linear relationship to non-linear relationship.

$$\frac{d\left(sig(x)\right)}{d(x)} = sig(x)\left(1 - sig(x)\right) \tag{1}$$

Where: sig(x) $represents\ the\ sigmoid\ activation\ function$

$\frac{d\left(sig(x)\right)}{d(x)}\ represents\ the\ derivation\ of\ sig(x)$

Besides, the loss function and optimiser are also needed to be clarified. In this paper, we choose cross entropy loss function. A loss function is used to evaluate the difference between the actual models with the model implemented. Among these loss function. CrossEntropy is the best choice for classification. Contrasted with CrossEntropy, MSE could result in some non-convex problem. While finishing the all the steps above, the basic version of the training neural network is formed.

3. Genetic algorithm

In order to get a more accurate result of the classification, an optimisation technique named evolutional algorithm is used. The implementation of this method is inspired by a research paper, in which a correct classification rate between 80% and 100 % is achieved [5]. A brief introduction of the method will be given in the following. As can be seen from its name, GA is inspired by biological evolution named natural selection which is proposed by Charles Darwin on the basis of observation and research, [3]. The basic idea is that many better descendants will replace those individuals who are not good enough. Similarly, GA is an algorithm that can simulate the process of the natural selection and the process of the evolution to find out the optimal result. To be more specific, GA is started with a population represents the possible answers to the problem. The population is consisted of many individuals with a certain DNA code. After the initial generation is produced, natural selection will be applied to the algorithm. For the purpose of finding the more suitable features, fitness function is defined to calculate the adaptability of each individual, which, in other words, represents the survival probability. Therefore, some individuals with higher survival probability will survive through the selection. After the selection, crossover and mutation are used to generate new offsprings. As can be implied from the name, crossover means that new generation will be created by two or more parents randomly. Besides, mutation is used to increase the diversity of the population by creating different gene. Finally, the whole process above will be repeated according to the number of generation and when the last generation is produced, a best DNA can be chosen.

In this paper, GA is used to find the combination of features which is the most appropriate for classifying the molecules through the accuracy of training. In other words, DNA represents each possible combination of the input features and each DNA code will be transformed into a corresponding dataset with a random combination of features. The DNA code is generated by the random number between 0 and 1. 0 represents the corresponding feature should be removed and 1 represents the feature should be reserved. According to this technique, a new sub-dataset is generated and used in the neural network function. Then a set of accuracy will be returned through the neural network function that represents their ability of survival. During following process, fitness function, cross rate and mutation rate are used to simulate the natural selection. The fitness function is set as the default formula. The cross rate is set to 0.8 and mutation rate is set to 0.002. Otherwise, if the rate of mutation is too high, the whole system will not be stable. When the process of reproduction completed after the times of generation pre-defined, a most fitted DNA code is chosen on the basis of the maximum accuracy. Similarly, the DNA stands for the combination of features, will be written into a text file. Then the text will be read by the neural network model to perform the further process.

4. K-Fold Cross Validation

For the purpose of getting a better model, the 10-fold cross validation is used, the main reason of implementing k-fold cross validation is to evaluate the generalisation ability of the model so that a better model can be chosen. In this part, a brief introduction of k-fold cross validation will be given.

When training neural networks, the input data is usually divided into three sets, training set, validation set as well as test set. 10-fold cross validation is used after the finish of training because only after training, the weight of the hidden neuron can be adjusted to a proper weight so that the result of the 10-fold cross validation can be more accurate. The principle of it is to first split the training set into 10 parts, regarding 9 of them as new training sets and 1 as a new test set. The process will be calculated 10 times for different training set as well as test sets. The accuracy of 10-fold cross validation in this paper is the mean of the accuracy of these 10 results.

The reason why 10-fold cross validation is used here is because that 10-fold cross validation will take less time compared to the time cost of leave-one-out cross validation. The latter one use every row as a test set which will definitely increase the time cost. Meanwhile, 10-fold cross validation also has better performance than hold out cross validation.

5. Bimodal Detection Removal

In order to improve the performance of the model, outlier detection method is applied in this model. To be more specific, generally, there will be noisy in the original dataset, which could decrease the accuracy of training. Thus it is important to remove those data which is outliers in terms of the general data. The major difficulty of this method is the definition of the outlier. In other words, to what extent, the data can be regarded as an outlier. In the paper provided, a method called bimodal distribution removal is introduced [6].

The reason why Bimodal Detection Removal is used in this paper is because that it is better than other outlier detection methods such as Absolute Criterion Method, Least Median Squares and Least Trimmed Squares. There are some advantages of BDR. For example, the removal happens very slowly which will offer the neural network model enough time to extract the information. And the removal will not start until the neural network itself find the outliers in the training set [6].

The main procedure of this method is showed as follows, before choosing the patterns that need to be removed, the training set needs to be evaluated first by the variance of the error ϑts, if it lower or equal to 0.1, then it is necessary to conduct the following operation. The mean of the error for all patterns needs to be calculated first to get the $\acute{\delta} ts$.

After that, the patterns whose errors are larger than $\acute{\delta} ts$ have to be taken from the training set, which forms a small subset. Then, the mean error $\acute{\delta} ss$ and the standard deviation σss of the subset needs to be calculated. Finally, those patterns whose error is larger than the addition of $\acute{\delta} ss$ and σss needs to be removed from the origin training set

$$error \geq \acute{\delta} ss + \alpha \sigma ss \quad where 0 \leq \alpha \leq 1 \tag{2}$$

Here are some precautions that need to be paid attention to. The first removal should not start immediately when the training begins. Because, generally, the ϑts of the original training set is above 0.1. Thus it can be checked when the times of training exceed a pre-defined number, for example, in the research, they set the threshold as 200. After the first removal happens, Bimodal Detection Removal has to be continued every 50 training times [6].

In their paper, they also mention that the removal cannot be continued all the time because if not stop removal, the whole training set is likely to be removed, which is not what we are expected. Thus, a condition needs to be met to indicate when it is a good time to stop the training process. They proposed a method based on ϑts. It needs to be checked again to halt the training process. Once ϑts is below 0.01, the whole training set needs to be halted in case of the over removal.

6. Evaluation of the results

In order to have a better visualisation of the result, two kinds of evaluating methods are used in this paper. On the one hand, we directly show the accuracy of the result in the test set, which are used to compare the performance of neural network. On the other hand, in order to know the specific details about the result, confusion matrix is also implemented. Overall, the accuracy will demonstrate the performance in general while confusion matrix can show the data of performance in a more detailed way.

7. Display of the results

For a better demonstration of the conclusion get during the training and comparison, figures and tables are used in the part of result. All the accuracy stated below is the average accuracy in every ten results, which could make the result more reliable. Besides, the figures about the loss function below is obtained during the training while the other figures are created in Python manually.

Results

Test is necessary to be carried out to evaluate the performance of the neural network model. The following part will mainly concentrate on:

1. Show the effect of Bimodal Detection Removal.
2. The effect of the performance of hyper parameters.
3. The trend of the loss in the whole training process.
4. Compare the result between the test accuracy of neural network with GA and neural network without GA
5. Compare the result between different mutation rate.
6. Compare the result between the accuracy of neural network with both BDR and GA and neural network with BDR only.
7. The comparison of K-Fold cross validation of back propagation method with or without the use of Bimodal Detection Removal and GA.
8. The comparison of the accuracy in this paper with the accuracy in the relevant paper [4].

- **The effect of Bimodal Detection Removal.**

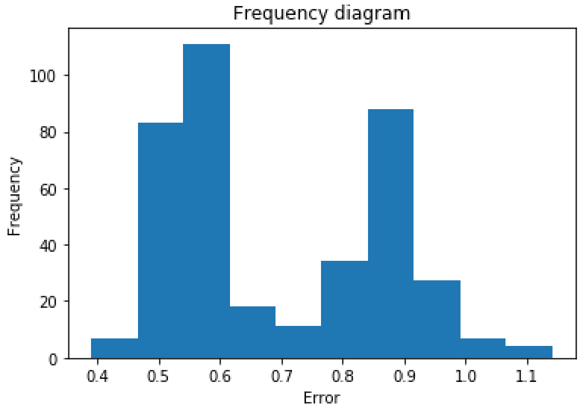

Figure.1 Distribution of error before BDR

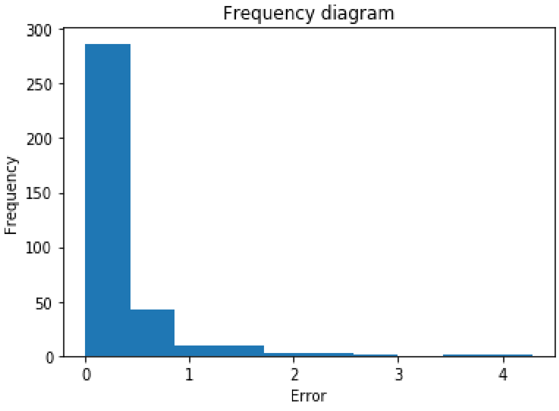

Figure.2 Distribution of error after BDR

Figure 1 represents the distribution of error in the initial training set. It can be seen that the distribution shaped like a bimodal, which indicates that there are some outliers in the original dataset that could have some effect on the performance of the training process. Figure 2 gives us a distribution of the error of the data after the training with BDR, which clearly presents the distribution of the error after BDR.

- **Tests about the influence of hyper parameters by the test accuracy**

  The size of the hidden neuron with the SGD optimiser, CrossEntropy loss function and learning rate = 0.01

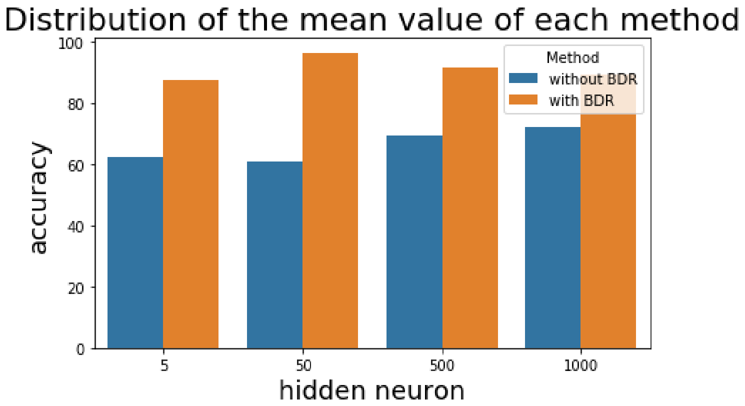

Figure. 3 comparison of number of hidden neuron between BDR and without BDR

From the figure above we can see that the test accuracy of the method without Bimodal Detection Removal is slightly increased with the increase of the hidden neuron. However, the accuracy of the method with BDR increases at first and when the neuron is big enough the accuracy begins to drop. Here are some aspects that are worth to be concentrated. The method without BDR performing poorly when the number of the hidden neuron is small, in some circumstance, the model even wrongly mark the label which results in only one label at last. This phenomenon stops when there are suitable number of hidden neurons.

The amount of learning rate with the SGD optimiser, CrossEntropy loss function and hidden neuron = 50

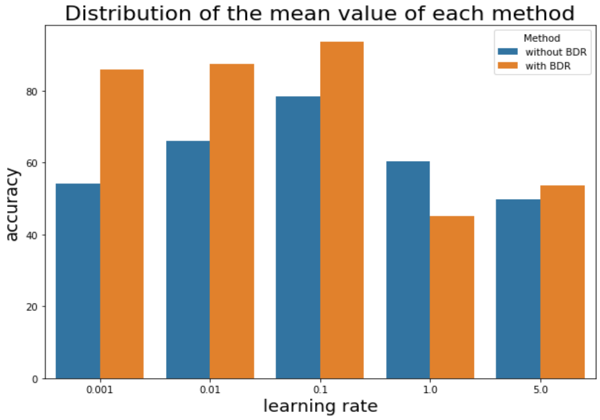

Figure. 4 comparison of learning rate between BDR and without BDR

From this figure, we can see that both of them stay increase before the learning rate reaches 0.1 and the accuracy of them seems not too bad. However, when the learning rate exceeds a threshold, the accuracy decreases sharply for both of them. Learning rate decide the speed of update of the weight. Thus, if the learning rate is too big, it is possible that the result has crossed over the best optimal result. On the contrary, if the learning rate is too small, the speed could be very slow.

- **The plot of the loss when the training is finished with the SGD optimiser, CrossEntropy loss function and hidden neuron = 50**

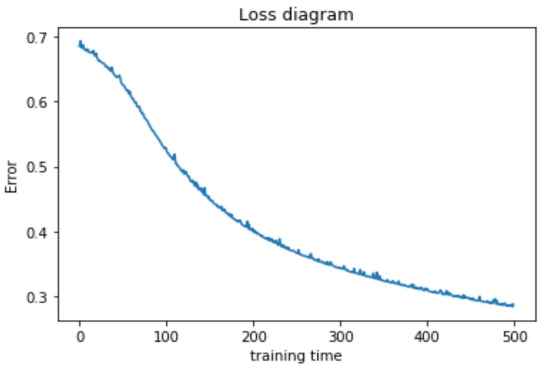

Figure.5 learning rate = 0.001

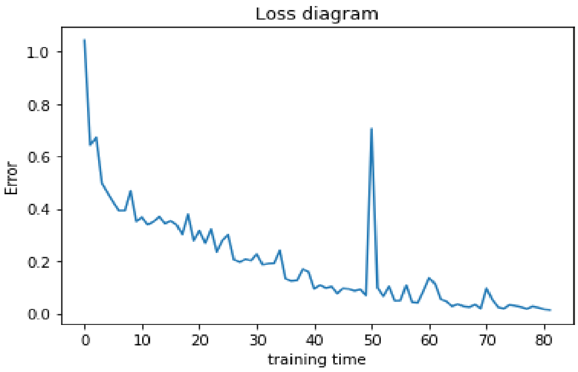

Figure.6 learning rate = 0.1

From Figure 5 and Figure 6, it can be indicated that the after the whole training finishes, the loss is decreased generally for both of them. However, when the learning rate is 0.1, the fluctuation seems to be more stable than the other one. When the learning rate is higher, the neural network will be more possible to miss the 0 slope point, which will make it more unstable. In addition, the model with 0.01 learning rate has cost more time than the model with 0.1 learning rate. Thus, the learning rate will also determine how fast the weight change.

- **Compare the result between the test accuracy of neural network with GA and neural network without GA**

Table. 1 Comparison of K-Fold in this paper with the relevant paper

Method	% correct
network with GA	75.3
network without GA	73.7

Remark:
GA represents the GA with N_GENERATIONS = 100, POP_SIZE = 100, CROSS_RATE = 0.8 and MUTATION_RATE = 0.002

According to Table. 1, the test accuracy of the neural network with GA is slightly higher than that of the neural network without GA

- **Compare the result between the range of test accuracy of GA with MUTATION_RATE = 0.002 and MUTATION_RATE = 0.9**

Table. 2 Comparison of K-Fold in this paper with the relevant paper

Method	% correct range
network with GA1	[61.0-78.6]
network with GA2	[56.2-80.7]

Remark:
GA1 represents the GA with N_GENERATIONS = 100, POP_SIZE = 100, CROSS_RATE = 0.8 and MUTATION_RATE = 0.002
GA2 represents the GA with N_GENERATIONS = 100, POP_SIZE = 100, CROSS_RATE = 0.8 and MUTATION_RATE = 0.9

It can be demonstrated from Table. 2 that the test accuracy range of GA with MUTATION_RATE = 0.9 is bigger than that of GA with MUTATION_RATE = 0.002. To be more specific, GA with MUTATION_RATE = 0.9 appears to have a lower lower bound and higher higher bound than another one.

- **Compare the result between the accuracy of neural network with both BDR and GA and neural network with BDR only**

Table. 3 Comparison of test accuracy of the neural network with BDR and neural network with both BDR and EA

Method	% correct
network with BDR	90.4
network with BDR and EA1	88.7
network with BDR and EA2	89.4
network with BDR and EA3	91.2
network with BDR and EA4	89.0

Remark:
GA1 represents the GA with N_GENERATIONS = 10, POP_SIZE = 10, CROSS_RATE = 0.8 and MUTATION_RATE = 0.002
GA2 represents the GA with N_GENERATIONS = 50, POP_SIZE = 50, CROSS_RATE = 0.8 and MUTATION_RATE = 0.002
GA3 represents the GA with N_GENERATIONS = 100, POP_SIZE = 100, CROSS_RATE = 0.8 and MUTATION_RATE = 0.002
GA4 represents the GA with N_GENERATIONS = 166, POP_SIZE = 166, CROSS_RATE = 0.8 and MUTATION_RATE = 0.002

From the results shown in the table above, it is obvious that the results of neural network implemented with both BDR and EA are better than the network with BDR only. The improvement is mainly because the use of genetic algorithm, in other words, the application of genetic algorithm in the neural network will improve the accuracy of the classification result. Moreover, when the size of the generation and the population increases, the results appear to be more accurate generally.

- **Compare the result of the 10-Fold cross validation in this paper with the results published in a research paper [4]**

Table. 3 Comparison of K-Fold in this paper with the relevant paper

Method	% correct

network with BDR	99.7
network with BDR and GA	99.2
original network	75.0
APR	92.4

Remark:
GA represents the GA with N_GENERATIONS = 100, POP_SIZE = 100, CROSS_RATE = 0.8 and MUTATION_RATE = 0.002

It can be concluded from the Table. 4, the accuracy of 10-Fold cross validation of the neural network with BDR or one with both BDR and GA are much higher than that of the original neural network. When comparing to the accuracy of APR [4], the result of neural network with BDR and the result of neural network with BDR and GA is better.

- **Compare the different test accuracy in this paper with that in the published paper. The test is implemented with hidden neuron = 129, learning rate = 0.1 [4]**

Table.4 Comparison of acccuracy in this paper with the relevant paper

Method	**% correct**
network with BDR	90.4
network with BDR and GA	91.2
original network	73.8
APR	86.6

Remark:
GA represents the GA algorithm with N_GENERATIONS = 100 and POP_SIZE = 100

The comparison between the test accuracy of different methods is provided in the Table. 5. And it can be seen that neural network with BDR or with both BDR and GA or APR [4] has better performance than the original network. Moreover,

Discussion

From the Figure 1, it can be indicated that the musk dataset is suitable for Bimodal Detection Removal due to the distribution of the error, whose shape is a bimodal. Figure. 1 and Figure. 2 clearly demonstrate the effect of the Bimodal Detection Removal on the dataset, which has a significant difference. This result is due to the reason that some outliers in the original dataset have been removed from the dataset so the distribution of the errors would be more likely in the same range.

According to the comparison of the results within different hyper parameters, it can be concluded that the number of the hidden neurons and the setting of the learning rate cannot be increased blindly. Otherwise, it will cause overfitting problem. However, the number cannot be too small either, which, on the other hand, will cause underfitting problem. As for the learning rate, as can be seen from these Figure.5 and Figure. 6, there are many waves in both of them. The reason for the fluctuation is mainly due to the use of Bimodal Detection Removal. Some training data are removed from the training set frequently, which would make it harder to train like the normal back propagation model. Figure. 4 indicates that different learning rates may result in different performance of the classification. Therefore, finding the suitable hyper parameters manually is trials-consuming. Some advanced techniques on finding proper hyper parameters will be discussed later.

When it comes to GA, in the light of the result in Table. 1, there is a slight improvement of the performance of neural network with GA. Thus, it can be concluded that GA can improve the performance by removing some trivial features in the original data. However, the change of the rate of cross rate and mutation rate is not obvious according to the test accuracy. It can be seen from the Table. 2 that higher mutation rate could lead to a better or worse result compared with the lower rate. That is to say, the stability of GA with higher mutation rate is not stable. Though it may lead to a better result, a stable program to perform classification is more preferred. To investigate a further research, GA is used in the neural network with DBR. However, Table.3 shows that the performance of the neural network with BDR is similar to that of the neural network with both BDR and GA. The reason of this result could be that BDR is more suitable for this dataset than GA so that the improvement of GA would not be so obvious compared with that of BDR.

Finally, according to the results in Table. 4 and Table. 5, the neural network with BDR only or with both BDR and GA can have more accurate result than the original neural network and APR method [4] in both the 10-Fold cross validation and the test accuracy. It is mainly because the removal of the some relatively irrelevant features, which could decrease some interference in the input data while training and testing.

Conclusion

After finishing the neural network and some tests, the neural network with both BDR and GA and the neural network with BDR appear to have higher classification accuracy, which is 91.2% and 90.4% respectively. The results in this paper is higher than that in the relevant paper. Moreover, there are some findings about these optimization techniques and the hyper parameters in the neural network. First of all, it is better to have the dataset pre-processed, which could make the data more suitable for training. Meanwhile, some bias or data with significant values can be balanced.

Besides that, there are many aspects that cannot be ignored. On the one hand, proper hyper parameters in the neural network is extremely essential because there will be a various differences among the different values of these parameters according to the result in this paper. On the other hand, K-fold validation is also an essential technique in neural network due to the reason that cross validation can access how the model performs outside the data, in other words, cross validation can evaluate the ability for the model to adapt to other kinds of dataset. Thus, it is an essential method when evaluating the model.

Furthermore, the optimisation techniques for improving the performance is also indispensable. For example, BDR and GA are good techniques that worth being used. As can be seen from the result, the performance of the classification can be improved a lot by removing the outliers of the input. Thus, using BDR and GA can help the scientists to classify the new molecules in a relatively high accuracy, which could save them a lot of time and expenses.

In conclusion, in order to gain a high performance feed-forward neural network, the follows states have to be satisfied.

1. Correct input neuron and output neuron.
2. A proper number of hidden neuron.
3. A proper number of training epoch.
4. A proper learning rate.
5. Cross validation is feasible but not compulsory.
6. An appropriate neural network model, such as back-propagation.
7. Some techniques need to be taken to process the input data or to modify the process in neural network.

Future work

1.1 Limitations

Although the average performance of the model in this paper is acceptable, there are still a lot of aspects that need to be improved. For example, in this paper, we only user three layers feed-forward neural network. Thus, some tests on the four or more layers neural network can be done to build a more robust network to improve the performance. Besides, the running time of the BDR in this paper is relatively slower than the basic back propagation, which is probably because of the use of the batch. Thus, in order to improve it data can be read line by line from the Dataframe instead of using batch from the data loader. Moreover, there are also some limitations in the genetic algorithm. Genetic algorithm is time-consuming in computation so when the problem become more complicated, the time would be a huge issue. Furthermore, random generating DNA code could also lead to the poor stability of the solution. There are also some other limitations on GA. For example, on the one side, the fitness function in GA can be replaced by some more intelligent and proper functions in classification. On the other hand, the hyper parameters in GA can also be chosen in a more suitable way instead of randomly modifying the number. There are also some limitations in the design in this paper except its intrinsic limitations. For example, the training epoch in the neural network function is not sufficient due to the limit of time, which would lead to a lower accuracy. In addition, some drawbacks about the back propagation algorithm which is implemented in this paper cannot be ignored either. For instance, the hyper parameters such as learning rate and the number of hidden neuron are adjusted manually, which does not obey the scientific principle.

1.2 Possible Extension

Fortunately, there are many researchers who have devoted themselves to researching the proper method to improve the performance of the neural network. A method called ADADELTA, it requires no manual adjustment of the learning rate, which works more robust. The main process of this method is to use first order logic which can dynamically adapt. Meanwhile the minimal computation is beyond stochastic gradient descend [9]. Moreover, the performance of neural network is also sensitive to the number of hidden neuron, which has been found in the testing part. Too few hidden neurons would lead to underfitting while too many neurons can result in overfitting. It can be solved by building a decision tree which can correctly divide the whole sample space. From the decision tree, important and related hidden neuron can be found and those which are not relevant can be removed. [7] Besides, gradient descend is the main method in back propagation to adjust the weight, which could result in the local minima problem [1]. As its name applies, the local minima means that there might be some local optimal solution but not the global one. Although it can be fixed by using random initial weight, it costs expensive computation time. Thus, a new method except back propagation needs to be find. According to their research, a simultaneous training method with a removal criteria can be used to improve the performance of the training. It can decrease the probability that local minimum happens. Meanwhile it can also use the resources efficiently. In addition, genetic algorithm can also be improved. For example, the probability of crossover and mutation can be adapted by using fuzzy logic [8]. A more suitable fitness function could be found in order to achieve better performance. For instance, a certain fitness function could be created for this classification tasks.

The methods that have been mentioned above can be used to improve the neural network in this paper. Besides these methods, there are also plenty of advanced methods that can be taken to improve the performance of the feed-forward neural network.

References

1. Atakulreka, A., & Sutivong, D. (2007, December). Avoiding local minima in feedforward neural networks by simultaneous learning. In *Australasian Joint Conference on Artificial Intelligence* (pp. 100-109). Springer, Berlin, Heidelberg.

2. Chapman,D, Jain,A UCI Machine Learning Repository [http://archive.ics.uci.edu/ml] AI Group at Arris Pharmaceutical Corporation

3. Darwin, C. (1951). On the Origin of Species (Vol. 71, No. 6, p. 473). LWW.

4. Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, *89*(1-2), 31-71.

5. Hope, D. C., Munday, E., & Smith, S. L. (2007, April). Evolutionary algorithms in the classification of mammograms. In Computational Intelligence in Image and Signal Processing, 2007. CIISP 2007. IEEE Symposium on (pp. 258-265). IEEE.

6. Slade, P., & Gedeon, T. D. (1993, June). Bimodal distribution removal. In International Workshop on Artificial Neural Networks (pp. 249-254). Springer, Berlin, Heidelberg.

7. Yuan, H., Xiong, F. and Huai, X. (2003). A method for estimating the number of hidden neurons in feed-forward neural networks based on information entropy. *Computers and Electronics in Agriculture*, 40(1-3), pp.57-64.

8. Zhang, J., Chung, H. S. H., & Lo, W. L. (2007). Clustering-based adaptive crossover and mutation probabilities for genetic algorithms. IEEE Transactions on Evolutionary Computation, 11(3), 326-335.

9. Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.