Binary Classification Model for Eye Gaze: Tuning Parameter for Shallow Neural Network with 3-Layer Structure*

Ruiyi Sun[1 [u6233314]]

Research School of Computer Science,The Australian National University
u6233314@anu.edu.au

Abstract. Due to the rapid development of the Internet, much information has poured into people's vision, and people's demand for the efficiency of network search has gradually increased. The search engine's ability to clearly distinguish between different types of tasks will improve user efficiency. This article compare a simple 3-layer model that predicts the performance of different tasks by analysing data such as eye movements and user behaviours with an improved 3-layer neural network by GA algorithm. Changing hyper-parameters (the number of hidden neurons, the number of epochs,loss function,optimiser and the learning rate), using k-fold cross- validation , changing the threshold of output and tuning hyper-parameters by GA algorithm effectively make the simple 3-layer neural network achieve 90% accuracy in test set and nearly 100% accuracy in training set. Through usage of change the output threshold of the 3-layer model, the test accuracy and train accuracy slightly improved. The GA algorithm increases accuracy by nearly 10%, but this comes at the cost of time. .

Keywords: Binary Classification · 3-Layer Neural Network · Eye Gaze · Tuning Parameter · GA algorithm.

1 Introduction

Internet search has become an important part of modern society. Due to the explosive growth of information, the demand for efficient search is becoming stronger. From the perspective of web search research, the improvement of the performance of web search engines is inseparable from the study of users' search behaviour (Buscher, Cutrell Morris, 2009) [2]. This article starts with a simple 3-layer neural network and proves that it can be used to predict the data set. Then,the model is improved by adjusting output threshold. This technology improves the accuracy of the model. Finally, the model compares with a optimal model with GA algorithm.

2 Methods

2.1 introduction of Dataset

This chapter will briefly introduce the attributes and terminology involved in the dataset "Jae-First_Exp_data2". It will help readers quickly understand this data set.

Eye-tracking technology is a widely used technology. It involves many complex areas of knowledge. In this chapter, knowledge and terminology related to web search performance in the past eye movement research field will be briefly introduced.

Eye-tracking technology is a technique that captures eye movements which allow researchers to know where the subject is looking and the sequence of change in gaze (Poole, Alex & Ball,2006) [7]. This article will explore how eye-tracking technology can be used to determine the intrinsic connection between search performance and search behaviour in web search. Search performance is a reference value to measure the quality of web search. Kim, Thomas, Sankaranarayana, Gedeon & Yoon (2015) claim that search performance includes two dimensions: search speed and search accuracy [5]. According to previous research, there is a great correlation between some factors such as type of tasks and search behaviour (Hsieh- Yee, 2001) [3]. The search behaviour maybe all the behaviour of the subject during the experiment. For example, the time when the link is first clicked and the time to complete the entire task.

Therefore, there are two branches of "Search Speed" and "Search Accuracy" under "Search Performance" in the data set "Jae-First_Exp_data2". "Search Speed" contains the time of the first click on the link and the time of completion of a single task in this data set. Considering the accuracy of the research data, Kim et.al (2015)

* Supported by Research School of Computer Science, The Australian National University.

describe that they will not have the first time the subject of the experiment reported the answer data, then the search accuracy will be 0 [5].

There are two states of eye tracking, fixation and saccades. Kim et.al (2015) conclude that fixation is a relatively static state in which the eyeball is kept while acquiring certain information and Saccades is a glance state where the eyeball is outside the fixation [5]. The special arrangement consisting of fixation and saccades is called scanpath. In the data set, the fixation is only the average value of the time the eyeball enters the fixation time in a single task and the average value of the fixation time of a single click. “Scanpath” in this data set has three attributes. The value of the third attribute is the value of the first attribute (the length of minimal scanpath) divided by the value of the second attribute (the length of compressed scanpath). This means that there are two ways to express scanpath. They are “Minimal scanpath” and “Compressed scanpath”. According to the description of A, the minimal scanpath is to delete the rank of the link that the recurrent eyeball is watching. For example, the result of 2-2-3-4-2 is 2-3-4; while the compressed scanpath will only continuously appear the value Delete, for example, the result of 2-2-3-4-2 is 2-3-4-2.

Scanning direction is an analysis of scanpath. ”Complete” can observe whether the subject stays in each rank. ”Linear” indicates that the stay of the subject is increasing. ”Strictly Linear” means that compressed scanpath will strictly follow the growth trend. ”Regression” means that the back-rank returns to the front rank. Skip means to skip the middle rank. Kim et.al (2015) define the search strategy as a summary of a person ’s search behaviour: depth-first (DF) focus on finding a possible link and then selects it; breadth-first (BF) traverse all links and select the most appropriate one; mixed reside between the two and the subject browses a part of the linked content and selects [5]. ”Trackback” is the number of links that a subject is read before making a decision. There are two kinds of task in this dataset: informational and navigational. The navigational task only needs a definite link and the informational task needs information from one or more links.

The target attributes of this data set are evenly distributed. In Figure 1, the image shows distributions of all features. It can be clearly seen from the figure that the distribution of ”size” and ”task” is very even. The ”Regression to 1 or 2”, ”wrong answer” and ”Skip” attributes are relatively sparse.

2.2 A Simple Neural Network

The simple neural network can have high accuracy with the tuning of parameters. **First and foremost, The input encoding should be applied to this data set.** Firstly, the string format in the data is converted to int. This includes “size”, “task” and “Strategy”. Then, switching the positions of 0 and 1 in “wrong answer”. This is because 1 in the previous item “Accuracy” is the correct answer and 0 is the wrong answer. To prevent confusion caused by unclear definitions, 1 means correct and 0 means wrong. Finally, Normalization will be tried in the input data set to explore whether the attributes in this data set require normalization. However, for unknown reasons, normalization reduces the accuracy of the entire dataset. This should be due to the same weighting of noise and target-related attributes. Therefore, normalization is not used.

This neural network uses “task” as the target and other values as input. Since “task” contains only two kinds of values,it is a binary classification. This neural network uses only a 3-layer structure (one input layer, one output layer, and one hidden layer) which looks like a logistic regression. Since this is a logistic regression problem, the activation function will be chosen between sigmoid and softmax.This is just a binary classification problem. The softmax function is only a extension for sigmoid function in the multi-classification problem (Bouchard,2008) [1]. Therefore, the activation function is sigmoid. The optimiser is SGD. The loss function is a combination of cross entropy and sigmoid. The loss function and optimiser use only the most common functions here. In the following content, these hyper-parameters will be optimized. This is a balance data set and the two types of tasks have no weight before, so recall, precision and F1 score do not need to be considered. This data set was initially divided into 80% training set and 20% test set. The standard for evaluating this NN was originally the Confusion Matrix. However, as shown in Table 1., the accuracy of the model’s training set does not perform well. Then, the evaluation of

Table 1. Confusion Matrix of Training Set.

	True	False
Positive	229	91
Negative	182	138

the loss value of this model is also added. Figure 2 clearly shows that the loss value is not monotonously decreasing.

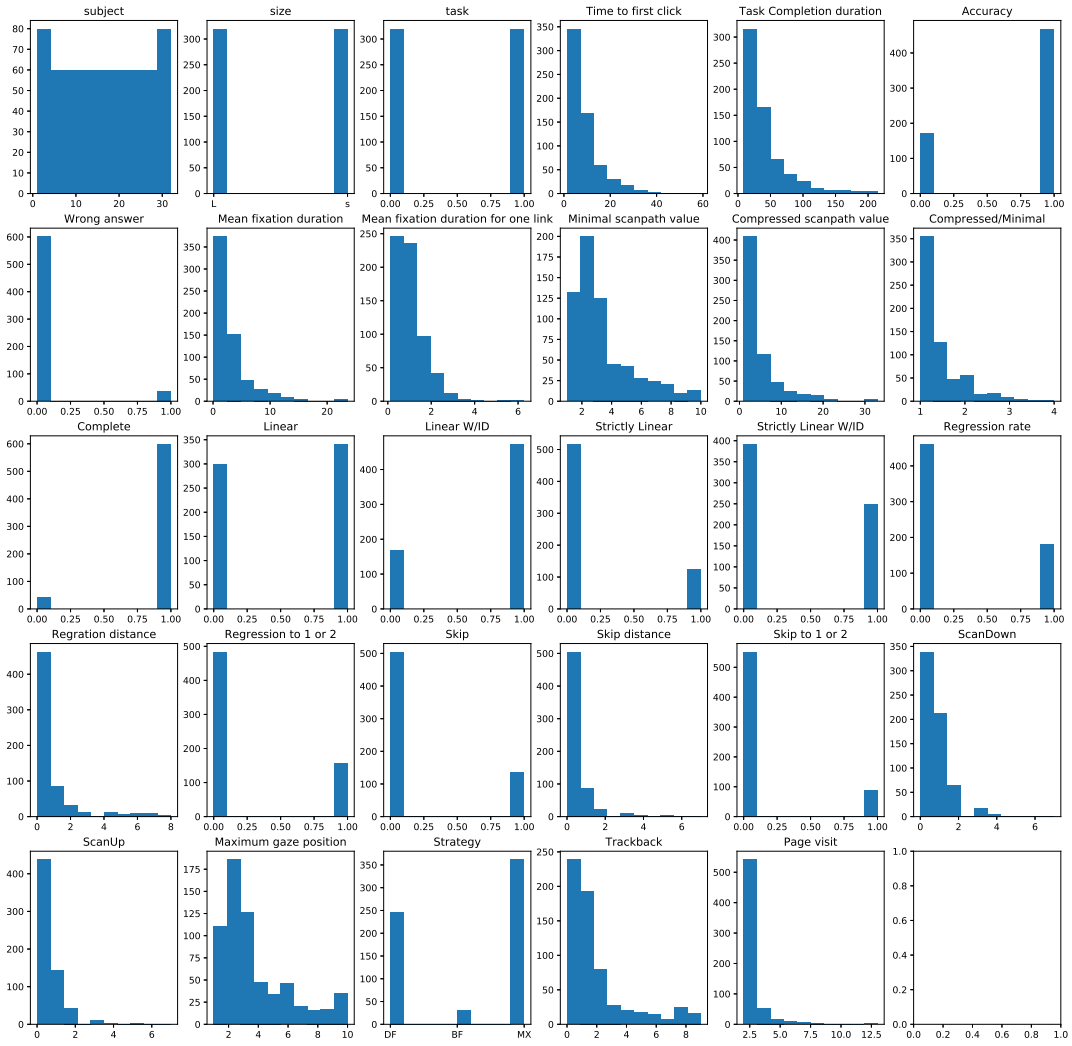

Fig. 1. Distribution of All Attributes.

For example,in the 100th epoch,the loss value rises to 0.525. In summary, this model not only has the problem that the accuracy is too low (the correct rate of 0.57 is only a little better than the probability of random guessing), but there may be situations where the accuracy is unstable (it cannot be proven that the accuracy of this data set will always maintain this number).

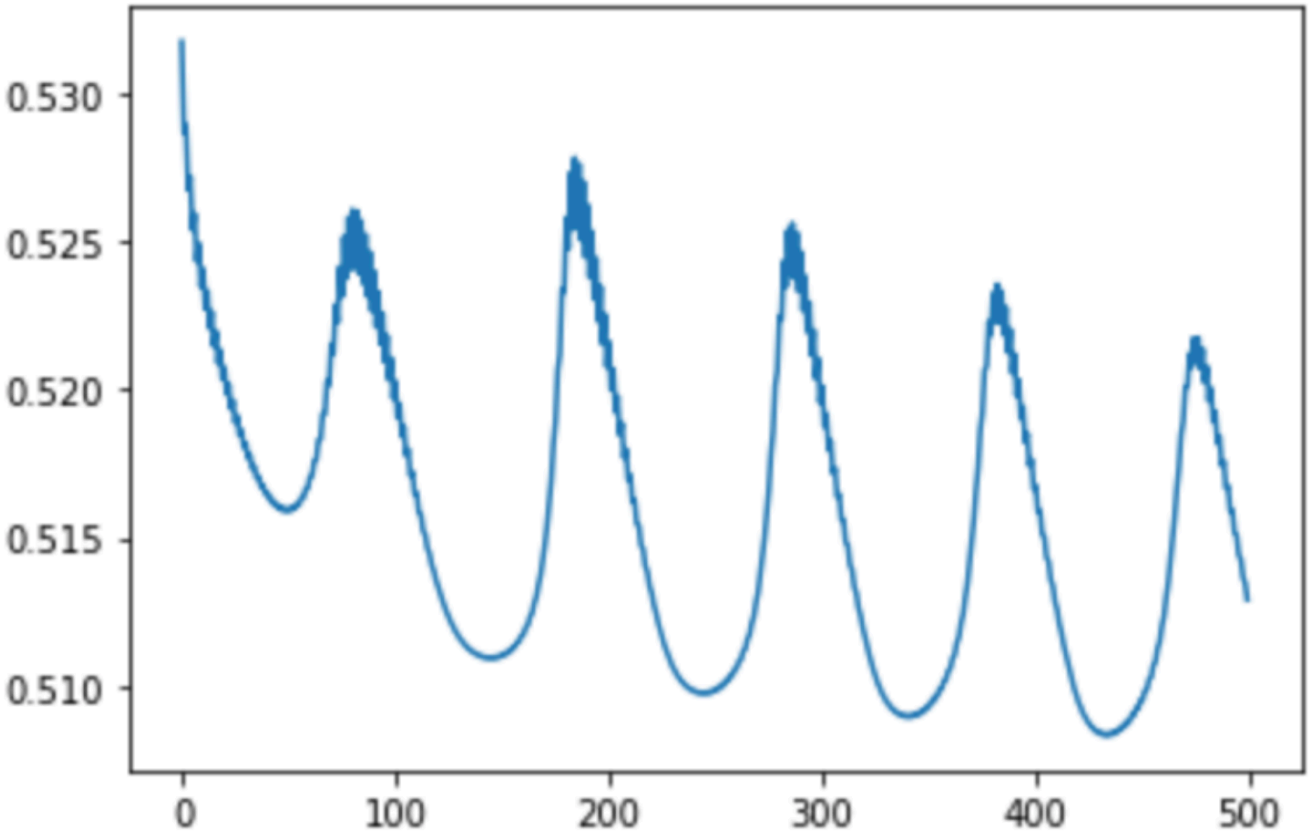

Fig. 2. The Loss Trend Per Epoch.

To prove that accuracy is trustworthy, 10 times 5-fold cross validation is used to separate the test set and training set and the epoch number is increased from 500 to 600. At first, the accuracy of the model is still not high. Then, the model performs better by modifying some parameters, such as the number of hidden neurons and the learning rate. Eventually, the average value of MSE is stable at 1.7 (round one decimal) and the average value of accuracy in the test data set is 69.7 (round one decimal). The reason for the poor overall performance of this model may be due to improper parameter settings. Next, some techniques will be used to improve this model.

Milne, Gedeon and Skidmore (1995) introduce a way to improve the final performance of the model by modifying the final output threshold of the binary classification [6]. They conclude that the output threshold found in Kogan's research does not necessarily have to be 0.5 and can be adjusted within a certain range. Meanwhile, there is an output neuron in the original neural network. The result of this output neuron is a number between 0 and 1. If following the original threshold (0.5), then an output greater than 0.5 will be predicted as an informational type task, and an output no greater than 0.5 will be determined as a navigational type.

Table 2 shows the impact of different thresholds on the accuracy of test dataset and training dataset. As can be seen from the table, the highest accuracy in the test dataset is 0.3, and the lowest is 0.8. There is a 5% difference between them. This method improves accuracy. However, the accuracy of the test set is still not satisfactory. Fig. 3 shows the changing trend of the accuracy of the test dataset in 10 attempts when 0.3 is the threshold. The x axis is 50 = k fold (5) * the number of repeat time (10). It can be seen from the figure that the accuracy of the test dataset fluctuates between 50% and 80% (Fig. 3). This means that the model does not predict a part of the data set well at the beginning. However, the accuracy improved over time. This may be caused by the model not fully training the data set or the convergence speed is too slow. Fig. 4 shows that the maximum accuracy of the training dataset does not exceed 75%. This illustrates the problem of underfitting (the number of hidden neurons is 4 now).

Table 2. Test Accuracy by Varying Output Thresholds.

Threshold	The average accuracy of the test data set	the average accuracy of the training data set
0.3	72.50%	73.92%
0.4	72.0%	74.66%
0.5	71.75%	73.20%
0.6	70.46%	71.09%
0.7	69.51%	71.39%
0.8	68.29%	68.81%

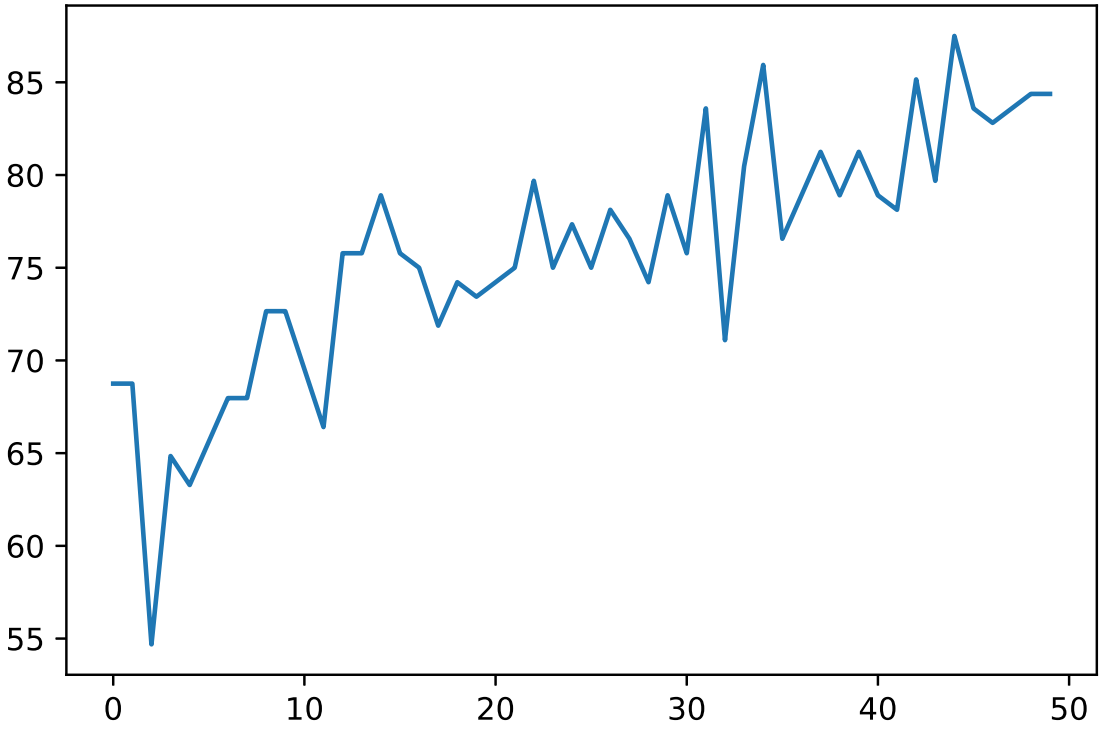

Fig. 3. The Test Accuracy Trend Per Epoch.

Therefore, the number of hidden neurons increases to 27. Currently, the average accuracy of the training set and test set were 76.93% and 79.71%, respectively. However, there is a certain difference between the average accuracy

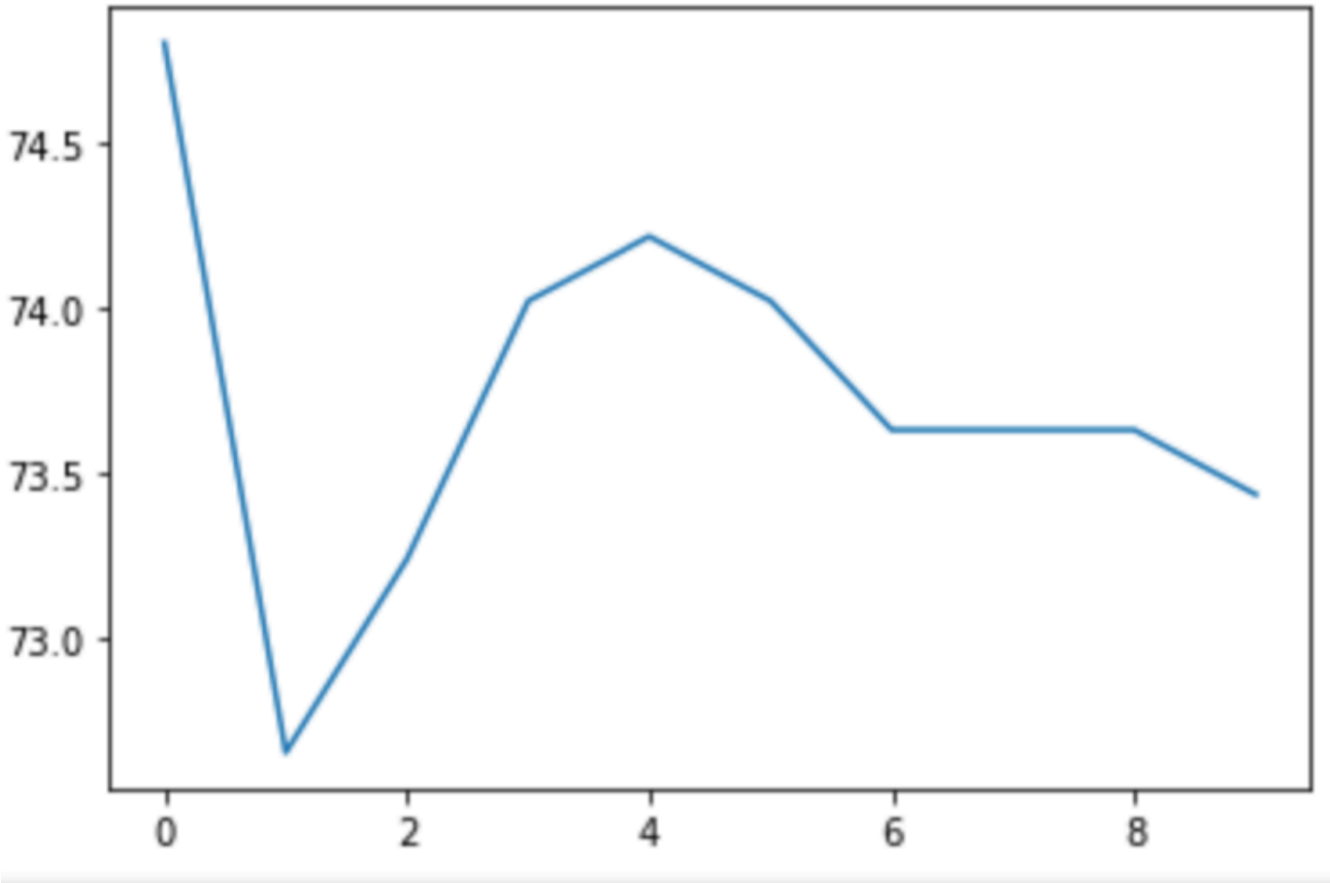

Fig. 4. The Train Accuracy Trend.

of the test set and the training set. This may cause by overfitting issue. According to the fact that the number of hidden neurons is randomly changed above, this seems to be caused by the excessive number of hidden neurons selected or the number of layers of the model is too small to extract feature. So, how to find a suitable hidden neuron quantity?

Table 3 clearly shows the relationship between different numbers of hidden neurons and data accuracy. When there are only 9, 6 and 3 hidden neurons, the accuracy of the data set will decrease but the gap between the training set and the test set will decrease. This can effectively prevent overfitting. Then, according to the performance of 12 to 27, the gap between the training dataset and the test dataset did not exceed 4%. And when the number of hidden neurons is increased to 100, the prediction accuracy of the test set of this data set is improved to 80%. This proves that a simple 3-layer neural network can better predict the data model. However, regardless of the number of hidden neurons, the accuracy of the test set stays at 80%. In summary, the choice of the number of hidden neurons should be a trade-off between the accuracy of the data set and overfitting. The accuracy of the data set is positively correlated with the probability of overfitting. From the above, we obtained good data accuracy through a simple

Table 3. Test Accuracy by Varying Output Thresholds.

Number of Neurons	100	27	24	21	18	15	12	9	6	3
Average Accuracy Training set	85.27	79.71	80.29	79.81	80.12	78.27	78.70	75.25	75.22	73.76
Average Accuracy of test set	81.35	76.46	76.82	76.68	76.40	75.85	76.17	73.20	73.15	71.65
Difference of Training and Test	3.92	3.25	3.47	3.13	3.72	2.42	2.53	2.07	2.07	2.11

3-layer neural network. Then, with the weight of this simple model, the features related to “task” will be found out. The total weight matrix can be obtained by multiplying the input weight matrix and output weight matrix of hidden neurons. Then take its absolute value. Finally, they are normalized.

2.3 GA algorithm for Tuning hyper parameters

The selection of hyper-parameters has not been given a detailed analysis in the previous simple model. Therefore, the accuracy of the model may still be improved. This chapter uses genetic algorithm (GA) to analyze and find the most suitable hyper-parameters and functions of the model. GA is an optimization algorithm. Its purpose is to seek the optimal solution.

This GA algorithm consists of several steps. First, the GA algorithm needs to initialize the initial population. For the GA algorithm to measure the model, the number of hidden neurons, the type of loss function and the type of optimiser should be constructed here by randomly selecting individuals. In this model, the initial number of population is 10. Then, the GA algorithm needs to create children to expand the population. The crossover in this model uses only a simple two point crossover. Next, the fitness function is used to evaluate all individuals for screening purposes. The average of the accuracy of the test set of the model is used here. Fourth, some offspring will consider genetic mutations. Here the probability of mutation is set to 80%. Steps 2 to 4 will be repeated until the repeated conditions fail. The method of looping 100 times has been adopted here to reduce unnecessary calculation time waste.

In this model, the number of hidden neurons should not be more than 100 as previously known. Therefore, the number of hidden neurons in GA algorithm is randomly selected between 2 and 100.This GA algorithm uses the test accuracy as a benchmark to find the loss function and optimiser that are most suitable for this data set. The candidate loss functions are 'L1Loss', 'MSELoss', 'BCEWithLogitsLoss'. Candidate optimisers are 'Adam' and 'SGD'. The adam function can perform weight decay so that the learning rate will not be too large and the final result will fail to converge. However, although the training result of adam function will be better than SGD, the generalization ability of adam is not as good as SGD(Keskar&Socher,2017) [4]. This may cause this model to fail to predict new data well. The remaining hyper-parameters, such as the learning rate, are consistent with the previous three-layer neural network. The structure of the entire neural network is unchanged. This means that the learning model is still a 3-layer neural network. In the end, almost all of the most promising results were "BCEWithLogitsLoss" and "Adam". The range of the number of hidden neuron is 10 to 27. In the end, comparing the results, adam is indeed more suitable for this model than SGD and GA optimization is better than manual adjustment.

2.4 Comparison of Two Models

Judging by the results of the above two models. The optimization of GA algorithm can make the model have higher accuracy. However, this is actually a way to trade time for accuracy. The GA algorithm actually consumes a lot of time. Using this algorithm may require sufficient computing resources and a certain amount of time. Fig. 5 and Fig. 6 show that the GA result which changes the loss function and optimiser perform better than the original model. As can be seen from these two pictures, the model optimized by the GA algorithm has obvious advantages. The above experiment proves that GA optimizes the accuracy of the model. However, the varying threshold technique does not improve models that have been optimized by the GA algorithm. It can be seen from the Fig. 7 that this technology only plays a weak role in improving the accuracy of the new model. Compared with Table 2,this technique slightly improves the accuracy of the two models.

3 Conclusion

This article uses the varying threshold technique to improve a simple three-layer model. In the end, the accuracy of this model has been slightly improved. By adjusting the hyper-parameters (loss function, optimiser and hidden neuron number) through the GA algorithm, the accuracy of the model is improved from about 70% to about 80%. This effectively proves that a simple 3-layer model can also achieve very good accuracy.

4 Future Work

There are still areas for improvement in this article. First, the generalization ability of the model in this article has not been evaluated. In future research, the validation set should be added. Then, the way of input encoding

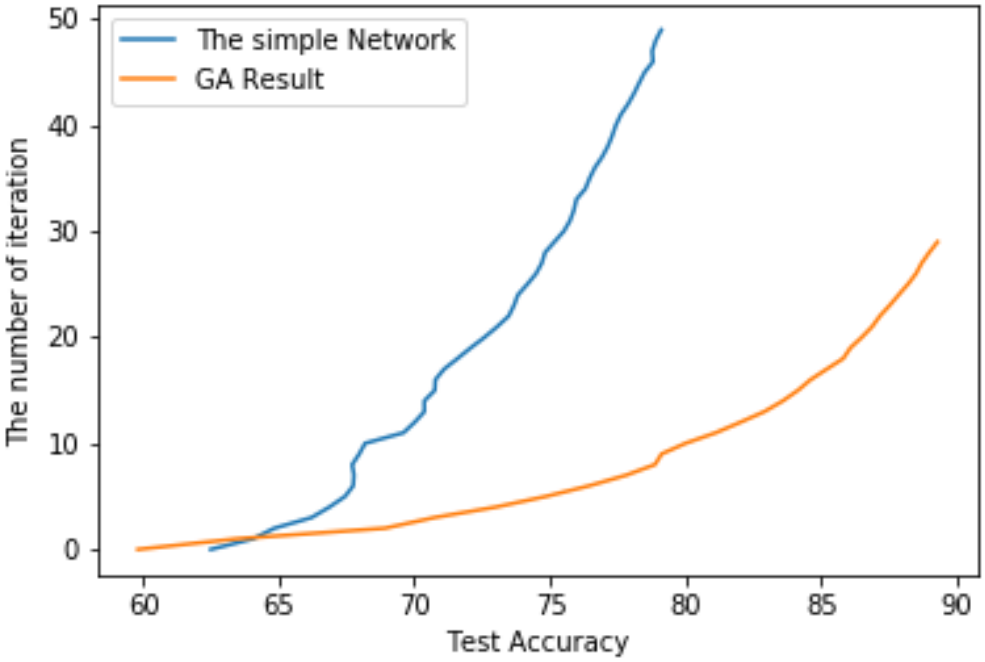

Fig. 5. Compare the Test Accuracy of two Method.

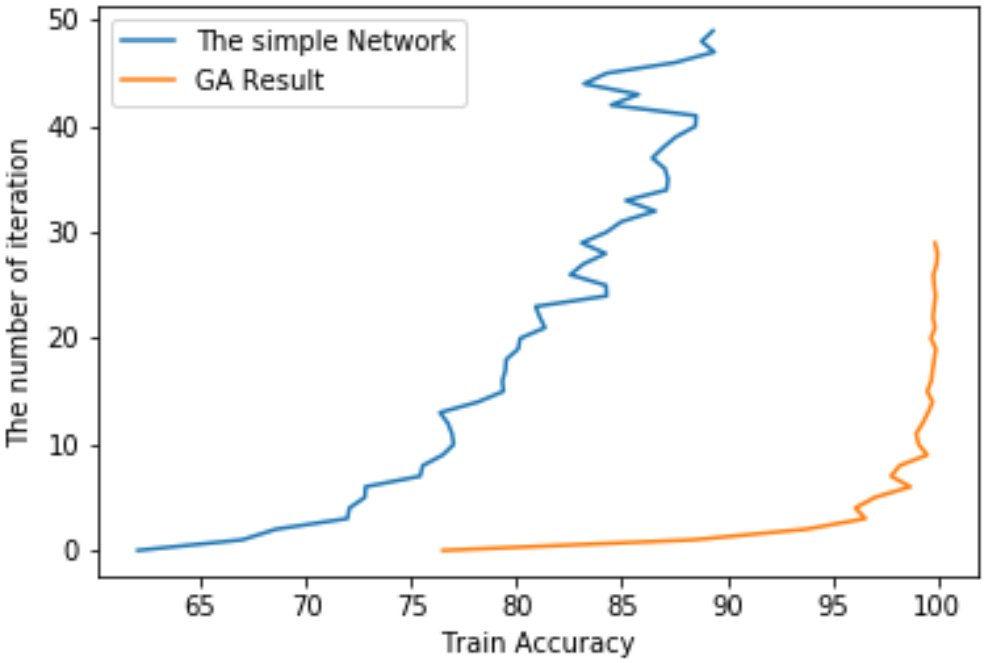

Fig. 6. Compare the Train Accuracy of two Method.

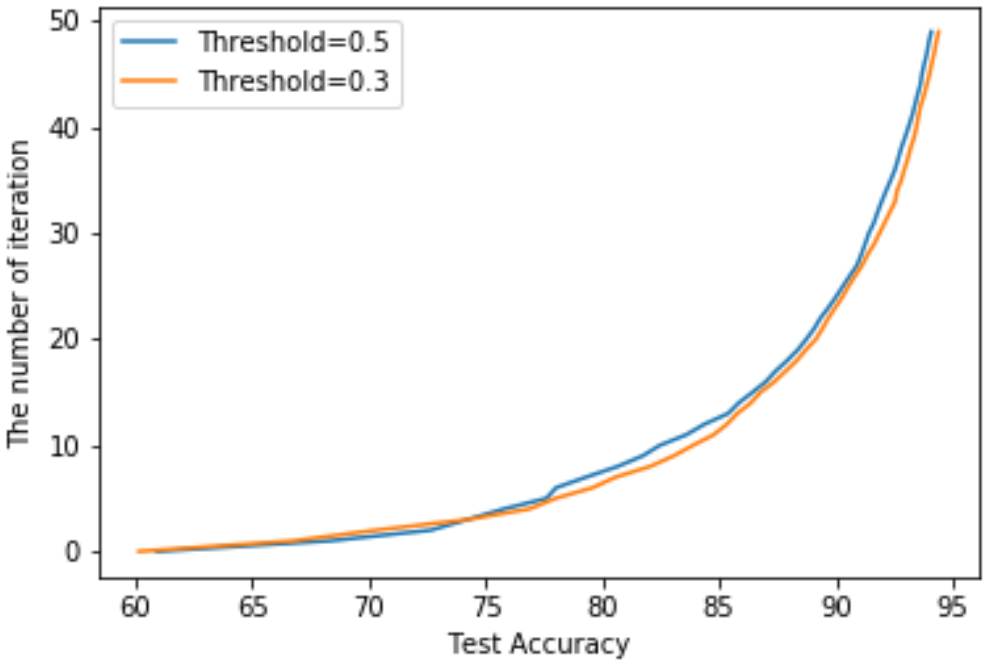

Fig. 7. Compare New Model Test Accuracy with different Thresholds.

needs to be improved. Integer encoding applies to ordinal data rather than categorical data. Using one-hot encoding will reduce the error. Finally, the model does not perform feature selection. If feature selection is performed, the accuracy of the model may still be improved.

References

1. Bouchard, G.: Efficient bounds for the softmax function and applications to approximate inference in hybrid models. In: Proceedings of the Presentation at the Workshop For Approximate Bayesian Inference in Continuous/Hybrid Systems at Neural Information Processing Systems (NIPS), Meylan, France. vol. 31 (2008)
2. Buscher, G., Cutrell, E., Morris, M.R.: What do you see when you're surfing? using eye tracking to predict salient regions of web pages. In: Proceedings of the SIGCHI conference on human factors in computing systems. pp. 21–30 (2009)
3. Hsieh-Yee, I.: Research on web search behavior. Library & Information Science Research **23**(2), 167–185 (2001)
4. Keskar, N.S., Socher, R.: Improving generalization performance by switching from adam to sgd. arXiv preprint arXiv:1712.07628 (2017)
5. Kim, J., Thomas, P., Sankaranarayana, R., Gedeon, T., Yoon, H.J.: Eye-tracking analysis of user behavior and performance in web search on large and small screens. Journal of the Association for Information Science and Technology **66**(3), 526–544 (2015)
6. Milne, L., Gedeon, T., Skidmore, A.: Classifying dry sclerophyll forest from augmented satellite data: Comparing neural network, decision tree & maximum likelihood. training **109**(81), 0 (1995)
7. Poole, A., Ball, L.J.: Eye tracking in hci and usability research. In: Encyclopedia of human computer interaction, pp. 211–219. IGI Global (2006)