Genetic Algorithm and Network Reduction in an information visualisation classification

Yijian Fan

Research School of Computer Science
The Australian National University
Canberra, Australia
u6711966@anu.edu.au

Abstract. Genetic algorithm is a method to search for the optimal solution by simulating the natural evolution process. In the classification task of neural network, genetic algorithm can help us determine hyperparameters and improve model performance. Network reduction is also an alternative for improving the model performance. Criteria such as distinctiveness can be used to judge redundant units and to do network pruning. In this paper, we perform classification on an information visualisation dataset. We set up genetic algorithm to optimise the hidden size and learning rate in our model. We also utilise the distinctiveness to judge the less functional units. The results show that the hyperparameters found by genetic algorithms lead faster convergence and less training loss. We also found that there were still some redundant hidden neurones thus the distinctiveness could be used for a better performance.

Keywords: Genetic Algorithm · Distinctiveness · Classification.

1 Introduction

Information visualisation is an important direction of human-computer interaction, which helps people to complete cognitive work better and reduces the complexity of information acquisition[14]. In practical application, this method has a good performance in many fields, such as business decision-making, online transactions, data analysis and education. Information visualisation processes complex information and uses tables and pictures to create visual stories for the information, making it easier for people to understand the data[9]. More importantly, visualisation usually provides multiple ways to collect user feedback, some of which are significant features of physiology[10]. Valuable features such as eye gaze statistics can be recorded through some devices and be applied into evaluating the performance of visualisation and producing some predictions[8]. In this study we choose the dataset from Hossain et al.[7]. They collect the eye behaviour of 24 observers and design some questions to study the user's perception of information.

In this study we choose the dataset from Hossain et al.[7]. They collect the eye behaviour of 24 observers and design some questions to study the user's perception of information. For this dataset we intent to build up the relationship between eye gaze data and information acquisition to illustrate the effectiveness of original design. The analysis can be done with the assistance of neural network. Neural network is a mathematical model that simulates biological neural networks for information processing[12]. By adopting back-propagation algorithms in a feed-forward network[6], the model could be more precise step by step in the training process, and therefore gain a good performance in classification and prediction.

To optimise our model, we use the genetic algorithm as well as some network reduction techniques. Genetic algorithms are stochastic search method inspired by Darwin's natural selection theory[4]. By simulating the evolutionary process, GA can preserve those suitable features and eliminate negative genes. This can lead the population to better direction. The most important application of Genetic algorithms is the optimisation. In our setting it is used to optimise hyperparameters such as hidden size and learning rate. These parameters are usually difficult to determine by empiricism and will greatly influence the performance. Further, we study the possibility of improving the performance of our classification through the techniques of network reduction. The excessive hidden units may increase the time complexity and cause some overfitting problems, and thus should be avoided. There are various literatures study the steps to reduce the network size. Some method, such as Gedeon-Harris method[3], manage to extract patterns from original network and use different measurements like distinctiveness to determine the functionality of units. This kind of factors include contribution[11], badness[5] etc. In our settings we test the Gedeon's method on our dataset and try to measure its effectiveness.

2 Method

2.1 Dataset

In the data set of the paper[7], the researchers collected answers for 12 questions from 24 people and recorded their eye movements when viewing two types of visualisation, including fixation and saccades. there was a total of 288 lines data with 22 column features.They managed to distinguish the relationship between the hierarchical and radial structures (column I) and the data. In this experiment, in addition to the classification of column I, we further analysed the relationship between other label groups and data, such as Correct Response (column K) with eye data. The prediction of the CR can help us understand the relationship between the behaviour of the observed person and the accuracy rate of the answer.

2.2 Classification

When designing the input of the model, we removed some non-related features, such as the interviewee's language background and academic background. We mainly used physiological statistical features as input. We first preprocessed the data, removing the name of the table, normalising and encoding the input and label items, respectively. In the process, the data marked as abnormal is removed, so the input matrix of (264, 7) is finally obtained. The input size was 7 as we added 7 features to the model.

Due to the scale of the data set, we designed a simple neural network model with one hidden layer to perform the classification task. The activation function was simply sigmoid function as it could constrain the output to the range between 0 and 1. We used the cross-entropy loss to calculate the loss as it could measure subtle differences and be applied to binary classification problems. For the optimiser in this model, the SGD method was adapted for the good generation speed. We manually split the data set as 80% for training and 20%for test, and fixed them for the comparison. We also performed k fold validation for more accurate comparison. loss and accuracy were recorded and displayed. A confusion matrix was also established to represent the model capabilities.

The hyperparameter in this classification model included learning rate, hidden size, number of training epochs. In the first model for interface classification we compared several parameters and chose the best one. We set those parameters as following: hidden size =6, epochs=200, learning rate=0.06. In the second model for correct response classification, we set the parameters as following: hidden size=10, epochs=200, learning rate =0.01

For the optimisation of parameters we used GA algorithm to the combination of hidden size and learning rate, which will be discussed in the following section. We then reused the best combination of parameters to compare with the above results.

2.3 Genetic algorithm

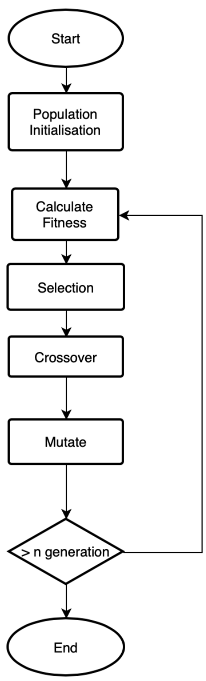

Fig. 1. The flow Genetic algorithm

Genetic algorithm was one of the most classic evolutionary algorithms. It created search space for finding the best solution in each generation and introducing randomness into the solution space to improve the solution generation by generation[1]. As shown in the Fig. 1, We used the genetic algorithm to simulate the evolutionary process. Each generation we performed calculation of fitness to select better parents and then generated group of children by doing the operation of crossover and mutate.

In our experiment. We used the idea of genetic algorithm to determine the suitable hyperparameters. Although the process should usually contain the process of encoding and decoding features, we simply generated random value between fixed range as the gene. Specifically, the range of hidden size was between 2 to 20, and the range of learning rate was between 0.01 to 0.1.

For the initial generation we randomly created 20 groups of hyperparameters containing hidden size and learning rate. These parameters were added into the NN model and returned the final loss. Then the fitness function was applied to determine the quality. If the loss was small, then the corresponding fitness would be high. After that, we used the roulette method to select the best solutions as parents. Higher fitness would gain more chance to be selected.

The children comes from two operations between parents: crossover and mutation. The rate of crossover was 0.7, if the random value was lower than 0.7, we exchanged the genes of two parents, which were hidden size and lr in this experiment. The mutation was to introduce some random values into the original solution space. We set the mutate rate 0.05. If the random value was lower than 0.05, we added random valuse onto the result of crossover.

By finalising the steps above, we completed one generation training and obtained 20 children. Then these children were reused in the next generation as new population. The generation number was 10, and the best parameters were printed out after 10 generations. The best hidden size and lr were taken into the original NN model to make a comparison of performance.

2.4 Distinctiveness

For the network reduction, we mainly applied Gedeon-Harris method[3] to determine the distinctiveness of hidden units. This method utilised the similarity of two pattern vectors to assess the hidden units. Each pattern vector, coming from a single hidden, represented the functionality of that unit. Basically it would contain same dimension with the input value. In this example, we had 264 inputs so that we should build a (264,1) vector to epitomise the pattern vector. Since we built a neural network with single hidden layer, we directly took the output of sigmoid function as the component of pattern vector.

$$\cos\theta = \frac{\boldsymbol{a}\cdot\boldsymbol{b}}{|\boldsymbol{a}||\boldsymbol{b}|} \tag{1}$$

After we obtained the pattern vector, we aimed to figure the similarity between two vectors. Our activation function had constrained the output of hidden units between 0 and 1, but we needed to do some normalising operation to set each value of component to between -0.5 an 0.5. As per Gedeon[3], this normalisation could lead the final angle to the range 0-180°, and therefore could support us find those similar vectors and complementary vectors. We then calculated the cosine similarity of two vectors, using equation(1). By changing the cosine value to the degree of angle, we got a vector angle table.

The thresholds for judging similarity and complementarity were set to 15°and 165° respectively, and the parameters were adjusted separately. If the angle between the two vectors was less than 15°, they were considered to have a similar adjustment effect on the input, so we could combine their weights and removed one of the units. If the angle was more than 165°, the two vectors could be seen as reciprocal, and their accumulative effect equaled to zero. They could be both removed from the network. By inspection of the angle and manually adjusted the weight, we could thus get a reduced network. In this experiment we used this method to find the possibility to do further network reduction after the hyper parameters being optimised by the genetic algorithm.

3 Result and Discussion

3.1 Interface Classification

The result of training is shown in Fig.2. We found that in our original settings of parameters, the average training accuracy of distinguishing the hierarchical and radial interface was around 51.56% (t=5, max=56.5%, min=47.4%), and the loss could be generalized to 0.693. The test accuracy was around 48% (t=5, max=57.50%, min=39.82%). The result showed that the hierarchical interface basically had an identical effect as the radial interface. According to Hossain et al.[7], most of the statics differed slightly from the two interfaces. So if we conducted a classification, the current data showed that it was hard to classify them.

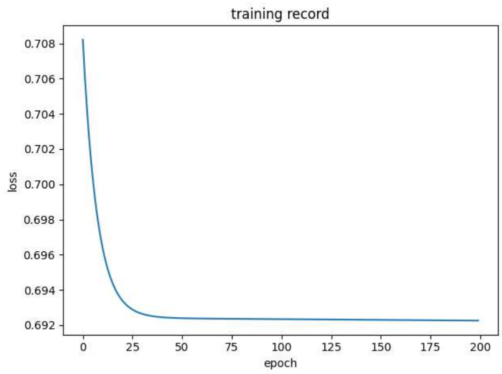

Fig. 2. Training record of Interface classification

We got the best combination of hyperparameters, in which the hidden size was 6 and the learning rate was 0.045. After we added those parameters into the NN model, we got the result(Fig. 3). The average training accuracy was rised to around 53.04%(t=5, max=57.5%, min=48.4%), and the loss was generalized to 0.69. The test accuracy was about 48.35% (t=5, max=57.50%, min=40.2%). From that we could see, the optimised parameters gained a better performance on the accuracy and loss. The convergence speed was basically same as the original one. However, we should note that the original parameters came from many trials, so the optimisation process using genetic algorithms could reduce that time for parameter decisions.

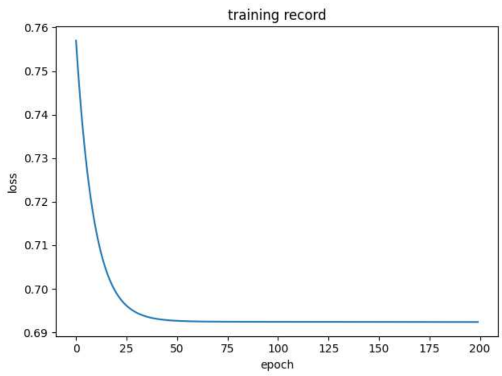

Fig. 3. The classification after GA

Then for network reduction, the data set is fixed to make a comparison of the effect of this technique. The original test accuracy is 48.17%, the time parameter is 0.1662. The distinctiveness showed hidden 3 and hidden 4 was similar and we combined them. The result after reduction showed the test accuracy increases to 50.08% and the time parameter decreased to 0.1420. The data is shown on Table 1.

Table 1. Network reduction of Interface classification

	original	$15°, 165°$
num of units	6	5
test accuracy	48.17%	50.08%
time	0.1662	0.1420

From the above process we found genetic algorithm could improve the model performance on the accuracy and could also help us reach an optimal solution more quickly. Moreover, we found that network reduction method could still found the redundant units and the pruning would also increase the accuracy slightly and lower the loss. In this experiment, both genetic algorithm and network reduction posted a good influence on our network.

3.2 CR Classification

This second classification is about Correct Response, that was to show the connection between the eye behaviour and the Correct Response they gave. The result of training is shown in Fig. 4. We found that the average training accuracy of this classification was around 80.40% (t=5, max=82.24%, min=79.56%), and the loss could be generalised to 0.53. The test accuracy is around 76.74% (t=5, max=82.08%, min=73.27%). The result also showed that this label was related to the data set and thus the test had a high accuracy. It showed the possibility that there would be different eye behaviour when people's response goes different.

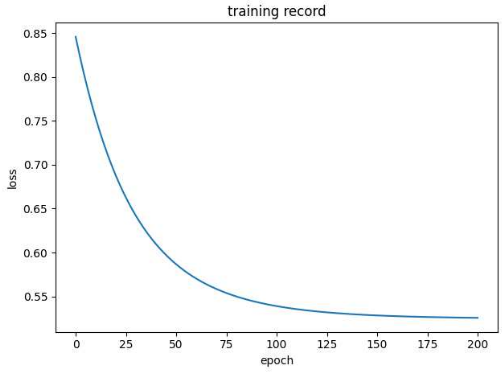

Fig. 4. Training record of CR classification

We got the best combination of hyperparameters, in which the hidden size was 17 and the learning rate was 0.045. After we added those parameters into the NN model, we got the result(Fig. 5). The average training accuracy and test accuracy raised slightly to around 81%, 78.2%. And the loss decreased to 0.52. From that we could see, the optimised parameters gained a better performance on the accuracy and loss. Moreover, after genetic alogorithm, the convergence speed greatly increase as you could see on the figure through x label.

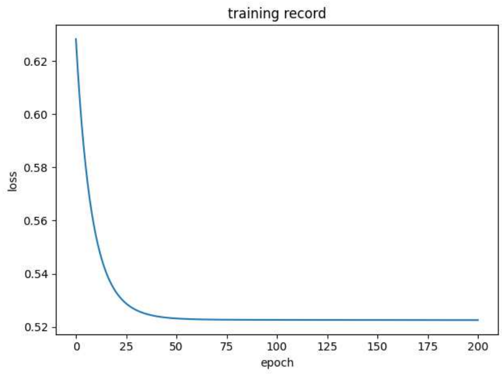

Fig. 5. The classification after GA

The result of network reduction was as following. The original test accuracy was 82%, the time parameter is 0.1831. The distinctiveness showed we should drop and combine several units. And the result after reduction showed the test accuracy was turning to 85.40% and the time parameter decreases to 0.1521. The data is shown on Table 2.

Table 2. Network reduction of CR classification

	original	$15^\circ, 165^\circ$
num of units	17	10
test accuracy	82%	85.40%
time	0.1831	0.1521

Since our method was performing the adjustment on hidden size and learning rate simultaneously, the hidden size found might not be the smaller one. From this experiment, the hidden size was 17. It provided us with the direction to reduce our hidden size by the Gedeon's methods. Although there were slight change in the performance. It would definitely decrease the time so it could help our model become efficient. From that point of view, both genetic algorithm and network reduction worked well and posted good influence on the original model.

3.3 Factors of GA

From our experiment, the genetic algorithm itself was influenced by some hyperparameters. For example, the generation number. The result of compared was listed on the Table 3 From comparison we could see the genetic

Table 3. The comparison of different generation numbers

	10	15	20
Best parameters	(7,0.049)	(18,0.112)	(18,0.057)
Loss	0.6924	0.6920	0.6925
time	14.20s	19.6743s	30.7470s

algorithm was influenced by the generation numbers. The intuitive thoughts of this parameter was that more generations led better results. However, it was not always true. Since the mutation would introduce some randomness, the evolutionary direction would also become worse. It indicated that when we performed GA, we needed to get a balance of generations number, considering the effect and efficiency.

We also found other factors relating to our model. For example, the initial size of population, if it was too small, our solution would not be good enough as the solution space was limited. However, a too large population would increase the processing time. Also, a too large population would require more precise fitness function to differentiate each loss.

3.4 Complexity of the method

The genetic algorithm in our experiment basically ran for reasonable time and with suitable complexity. However, if the data set was large, we needed to use batch training and we needed more careful consideration of each parameters. For network reduction, we basically set the weight manually to change the role of the unit. For multiple vectors that are similar or opposite at the same time, we mainly decide by inspection. But as the amount of data increases, the complexity of this method will increase. For example, the dimensions of the data will be very large, so storage will become more difficult. It also becomes difficult to traverse every two pairs of vectors and record their angles.

4 Conclusion and Future Work

In this experiment, we conducted two sets of experiments using eye behaviour data and related labels. Genetic algorithm were used to generate hyperparameters such as hidden size and learning rate. From the results we found that genetic algorithm had improved the model performance. The chosen parameters would lower the training loss and increased the convergence speed. However, the balance between the performance and time was really significant so we suggested that the number of generations and population size would be carefully designed. Moreover, we tested the Gedeon-Harris method to the NN model after the genetic algorithm. We found that some groups of hidden size determined were relatively large and thus the network reduction was necessary. By comparison of the results, the method was proved to be effective in improving accuracy and speed.

We study the hyperparameters in neural network using the genetic algorithms. The inspiration behind this is that we can also use the genetic algorithm to adjust our model structure. As stated in the paper [13], we can recombine the structure of our model by linking different neurones from different layers to generate new networks. If the fitness is of high quality, we can obtain a better NN model. For network reduction, we can also use a similar method to determine which input does not have a great impact on the final result. For these inputs, we can also filter to eliminate unimportant features. Some literatures have already obtained good results[2]. We notice that in a large-scale network, only use distinctiveness to judge the units may be difficult and inefficient. Some combined methods or some indicators that are easier to calculate may require further study.

References

1. Dasgupta, D., Michalewicz, Z.: Evolutionary algorithms in engineering applications. Springer Science & Business Media (2013)
2. Gedeon, T.D.: Data mining of inputs: analysing magnitude and functional measures. International Journal of Neural Systems **8**(02), 209–218 (1997)
3. Gedeon, T., Harris, D.: Network reduction techniques. In: Proceedings International Conference on Neural Networks Methodologies and Applications. vol. 1, pp. 119–126 (1991)
4. Gen, M., Cheng, R., Lin, L.: Network models and optimization: Multiobjective genetic algorithm approach. Springer Science & Business Media (2008)
5. Hagiwara, M.: Novel backpropagation algorithm for reduction of hidden units and acceleration of convergence using artificial selection. In: 1990 IJCNN international joint conference on neural networks. pp. 625–630. IEEE (1990)
6. Hecht-Nielsen, R.: Theory of the backpropagation neural network. In: Neural networks for perception, pp. 65–93. Elsevier (1992)
7. Hossain, M.Z., Gedeon, T., Caldwell, S., Copeland, L., Jones, R., Chow, C.: Investigating differences in two visualisations from observer's fixations and saccades. In: Proceedings of the Australasian Computer Science Week Multiconference. pp. 1–4 (2018)
8. 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)
9. Kucher, K., Kerren, A.: Text visualization techniques: Taxonomy, visual survey, and community insights. In: 2015 IEEE Pacific Visualization Symposium (PacificVis). pp. 117–121. IEEE (2015)
10. Lallé, S., Toker, D., Conati, C., Carenini, G.: Prediction of users' learning curves for adaptation while using an information visualization. In: Proceedings of the 20th International Conference on Intelligent User Interfaces. pp. 357–368 (2015)
11. Sanger, D.: Contribution analysis: A technique for assigning responsibilities to hidden units in connectionist networks. Connection Science **1**(2), 115–138 (1989)
12. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural networks **61**, 85–117 (2015)
13. Stanley, K.O., Miikkulainen, R.: Evolving neural networks through augmenting topologies. Evolutionary computation **10**(2), 99–127 (2002)
14. Ware, C.: Information visualization: perception for design. Morgan Kaufmann (2019)