# The Analysis of An Eye-tracking Experiment Based on Deep Neural Network and Genetic Algorithm

Jun Yang
u6767560

**Abstract.** The paper mainly explores the relationship inside of the eye-tracking experiment data set. Classifiers are built to predict the device screen size and search task type according to the search performance and search behaviours. First of all, the multi-label and multi-class classifiers are built based on similar deep neural networks. Then the selection of features and hidden neuron numbers are explored by the application of the genetic algorithm, the GA model is trained to find the optimal solution of choosing those parameters in order to find the optimal multi-label classifier and multi-class classifier. Comparing the basic models, it finds that multi-class models have a high test accuracy. Meanwhile, GA can increase the accuracy of both models. At last, the trend of feature selection was analysed, and one sparse feature is unwanted in both GA models.

**Keywords:** Deep Neural Network, Classifier, Genetic Algorithm

## 1 Introduction

Information searching weights more and more in our daily life in recent years. As people use different devices to access to the internet, search various information via different search engines more and more frequently, it is appealing to optimise the design of these web applications in order to improve the search performance. Given the certain device and search task, people will behave differently with respect to the search performance and behaviour metrics, it would be inspiring if the relation among device, task and performance could be found.

Eye-tracking uses a specific device to monitor the movement of our eyes. Research [1] believes eye-movement has some implications to help understand the search behaviours. According to [1], an experiment was designed and implemented. The experiment recorded the searching performances and behaviours of the participants, and it found that many results were similar with respect to different screen size and task type.

Based on the data set from [1], this paper explores using deep neural network and genetic algorithm to train and optimise models so that the models can reflect the relationship inside of the data set.

## 2 Basic Deep Neural Network Models

Deep neural network models are built to classify two targets: device size and task type. Two classifiers have a similar structure, and their outline is listed in table 1 and table 2.

**Table 1.** Multi-label classifier structure.

| | Type | Neuron Numbers | Activation |
|---|---|---|---|
| Input Layer | Linear | 26 | |
| 1st-Hidden Layer | Linear | 10 | Sigmoid |
| 2nd-Hidden Layer | Linear | 8 | Sigmoid |
| 3rd-Hidden Layer | Linear | 4 | Sigmoid |
| Output Layer | Linear | 2 | Sigmoid |
| Criterion | Binary Cross Entropy Loss | - | - |
| Optimiser | Adam | - | - |

**Table 2.** Multi-class classifier structure.

| | Type | Neuron Numbers | Activation |
|---|---|---|---|
| Input | Linear | 26 | |
| 1st-Hidden Layer | Linear | 10 | Sigmoid |
| 2nd-Hidden Layer | Linear | 8 | Sigmoid |
| 3rd-Hidden Layer | Linear | 4 | Sigmoid |
| Output | Linear | 4 | Log_softmax |
| Criterion | Cross Entropy Loss | - | - |
| Optimiser | Adam | - | - |

### 2.1 Eye-tracking Experiment Data Set Analysis

The data set contains 26 input features, 4 of which are search performance and are all numerical values. The rest of the features are search behaviours, and it contains one categorical value: DF, BF, MIX, which represents the search strategy. There are 2 targets in the data set, 'L' and 'S' represent large screen size and small screen size. 'ino' and 'nav' represent two different task types.

The input features are not all independent, feature 'Compressed/Minimal' is the ration of the other two features 'Minimal scanpath value' and 'Compressed scanpath value'. Some features are boolean values represented by 1 and 0.

For the basic DNN models, all categorical features and targets will be encoded by one-hot encoding. Other features will be regarded as numerical values.

### 2.2 Data Encoding

The input feature ‘Strategy’ contains 3 categorical values: DF, BF, MIX, so that each of them can map to a 3-bits value. The target contains two labels. They can be one-hot encoded separately or in a combined way.

For the multi-label classifier, two targets are encoded separately. For example, if the target is ‘S’ and ‘inf’, the two targets can be [0, 1] and [0, 1].

For the multi-class classifier, two targets are combined into 4 different classes, such as (‘S’,‘inf’), which can be represented by [0, 0, 0, 1].

### 2.3 Training and Results Analysis

The original experiment data set is encoded first. After that, the data is shuffled then split into training data set and testing data set, each of them accounts for 80% and 20% of all data respectively.

Based on the structure shown in table 1 and table 2, two DNN models are created.

Each DNN model is trained in 200 epochs. Stochastic gradient descent is used to optimise the models. For the multi-label classifier model, three accuracies are calculated, which are the accuracy of first label prediction, of the second label, and when both labels have correct predictions are the same time. For the multi-class classifier model, only one accuracy is calculated.

The training accuracy and loss are shown of the multi-label classifier are shown in Fig. 1.

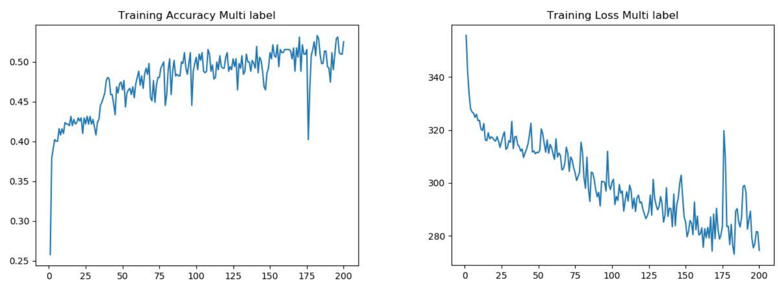


**Fig. 1.** Training accuracy and loss of multi label classifier

The testing accuracy of two correct label predictions at the same time is 39.84%. The accuracies of label 1 and label 2 predications are 64.06% and 64.84%.

The training accuracy and loss are shown of the multi-class classifier are shown in Fig. 2.

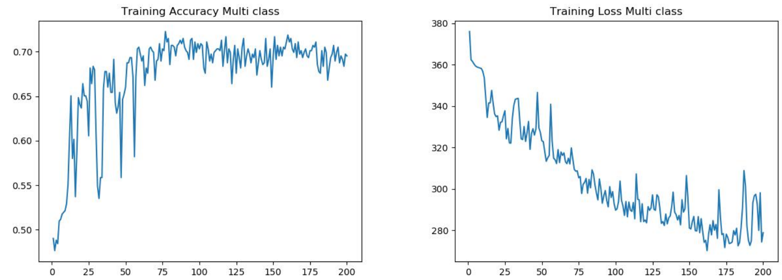


**Fig. 2.** Training accuracy and loss of multi class classifier

The testing accuracy of the model is 60.16%

From the results, it can be seen that multi-class classifier has a higher test accuracy, which is much better than the multi-label classifier. It might because the multi-class classifier has four neurons in the output layer, but in the multi-label classifier, there are only two, and more neurons allow better prediction.

## 3 Optimising Models with Generic Algorithm

Genetic algorithm and artificial neural network are combined to improve the classification accuracy [2], in which GA is used to optimise the feature selections and model parameters. Because in the previous section, all features are used as the input, which is not necessary. For the dependent features and the noise, it might have a better test accuracy when features are selected. Meanwhile, the neuron numbers in three hidden layers will be optimised as well.

In the GA model, the population is initialised first. Each individual in $k_{th}$ generation represents the feature selection and hidden neuron numbers. After the translation of the chromosome, a DNN model can be created and trained through the selected input features. After the training of DNN model, the test accuracy is applied to be the fitness value of the corresponding chromosome. The best chromosome and its fitness value in each generation are stored, after training the GA model, the best one is chosen.

The training process of the GA model can be slow if the population or the epoch is too large. In GA model training, there are several strategies to shrink the population [3]. Here a linear strategy is applied to decrease the population size during the training process.

Meanwhile, Mini-batch is used in the training of DNN to accelerate the training, which is different from basic DNN training in previous section.

### 3.1 Chromosome Encoding

Chromosome is a binary string that represent the variable being explored. GA is an optimising process aiming to find the best chromosome with the highest fitness value.

In the paper a 2-D array is used, each row is a list of integers in the range of [0 ,1], instead of a binary string. The length of the list, or the chromosome is 38, and is shown in Fig.3.

| | | | |
|---|---|---|---|
| 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 1 1 | 1 0 0 1 | 1 1 1 1 | 0 1 0 1 |
| Feature selection mask | n_h1 | n_h2 | n_h3 |

**Fig.3** Chromosome

The feature selection mask in the chromosome has 26 bits. When a bit is 1, the feature will be selected, otherwise it will be deleted. n_h1, n_h2 and n_h3 are the hidden neuron numbers, and 4-bits Gray encoding is used here. For example, in Fig.3, in the feature selection mask, the first digit is 1, then the first feature will be selected, n_h1: [1, 0, 0, 1] can be decoded as 14, which means the first hidden layer has 14 neurons.

### 3.2 Shrinking Population

In order to increase the speed of the training, population is decreasing during the training process. At the beginning, the population is 80, a linear decreasing method is used.

$$\mathrm{p}op\_size_{\mathrm{t}} = \mathrm{p}op\_size_{\mathrm{t-1}} - (5 \times \mathrm{t} / \mathrm{number}_{\mathrm{generation}}) \qquad \textbf{(1)}$$

A minimum population size is also set as 40, when the size is less than the minimum population size, set it to the minimum size and stop decreasing.

### 3.3 Operators

The operators contain selection and reproduction. Selection is used to select better chromosome after all fitness values are calculated in one generation. Reproduction is used to create next generation.

Proportionate selection is applied in the GA model, which enables the selection probability proportional to the corresponding fitness value. The probability $p_i$ is calculated in the following equation.

$$p_i = \frac{f_i}{\sum_{j=1}^{pop\_size} f_j} \qquad (2)$$

In this paper, reproduction contains one point crossover and uniform mutation. One point crossover is a crossover method, the operation contains the following steps: 1, randomly select a point in the chromosome; 2, switch the part after the point from two parents; 3, the crossover rate is the probability that previous two steps happen.

The uniform mutation means that each point is probably negated under a probability mutate rate.

### 3.4 Training and Results Analysis

After population initialisation, the GA model is trained, the training process is shown in Fig.4.

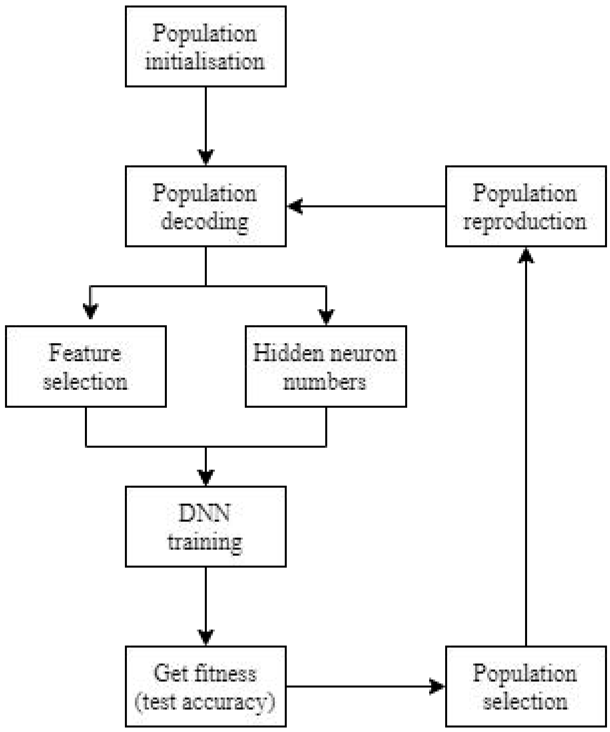


**Fig.4** GA model training process

For the multi-label classifier optimised by GA, the best fitness value is 58.59%, the corresponding translated chromosome is:

[[2, 7, 8, 12, 15, 16, 19, 20, 21, 22, 26], 12, 6, 8]

[2, 7, 8, 12, 15, 16, 19, 20, 21, 22, 26] are the feature indexes that deleted from the input features. 12, 6, 8 are the hidden neuron numbers. From all best chromosome in each generation, features 14 , 20 and 19 have the highest proportion to be deleted, which is 56%, 56% and 52%.

For the multi-class classifier optimised by GA, the best fitness value is 71.88%, the corresponding translated chromosome is:

[[1, 2, 5, 7, 11, 12, 18, 19, 20, 25, 26], 13, 14, 13]

From all best chromosome in each generation, features 15 , 1 and 19 have the highest proportion to be deleted, which is 63%, 63% and 54%.

Feature 19 is ‘Skip to 1 or 2’, and majority of the values are 0, which could be the reason that most chromosome decide to delete this feature, because it has less effect on the DNN training and the fitness value.

With the optimisation, the accuracies of both classifiers have been improved, especially the label classifier model.

## 4 Conclusion and Limitations

The testing accuracy is not high for both basic DNN models, which is in line with the conclusion in the research [1]. When training a DNN model based on such data set that the correlation between input and target are small, it is hard to get a good model. Under similar set-up, the basic multi-class model has a high prediction accuracy, which could suggest that when confront with similar problem, a multi-class model is preferred to the multi-label model.

The GA can be used to optimise the DNN model, the test accuracies in both scenarios have increased after optimisation. The drawback is optimisation process takes much more time than solely training a DNN model.

The limitation of this paper is obvious. In the first place, the different features have different meanings in real life, it is difficult to find an appropriate way to pre-process the data. Secondly, the training of GA model is not based on a large population size nor large number of generation, so that the solution might not be optimal. Thirdly, the data set only has less than 30 features which is too small, so it is more likely that the GA model returns a solution that delete the features that we concern.

## References


1. Kim, J., Thomas, P., Sankaranarayana, R., Gedeon, T., Yoon, H.: 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, 526-544 (2014).
2. Sharma, N., Gedeon, T.: Hybrid Genetic Algorithms for Stress Recognition in Reading. Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics. 117-128 (2013).
3. Hallam, J., Akman, O., Akman, F.: Genetic algorithms with shrinking population size. Computational Statistics. 25, 691-705 (2010).