Application of neural network with genetic algorithm and pruning optimizer in face features classification

Yuying Yan

Research School of Computer Science,
Australian National University, Canberra Australia
u6844740@anu.edu.au

Abstract. Neural network plays a significant role in the fields varies from speech synthesis (Sejnowski and Rosenberg, 1987), to bond underwriting (Dutta and Shekhar, 1988), the most common field is image recognition, many applications use Neural network in face recognition function. In order to help neural network model to learn better, in this project, we used genetic algorithm and pruning to train a basic neural network model for analyzing and learning the facial distance data, we will discover how this trained neural network model performed when identifies whether two faces belong to the same person through the data. As result, the model trained in this experiment achieves about 72%~81% accuracy in the prediction of test data.

Keywords: Neural Network， Facial features, Genetic Algorithm, Pruning

1 Introduction

Photos, as a special medium, carry and store a great amount of information. In the process of social change, a lot of photos, especially face photos, have lost relevant information, no one knows who these people are. We can identify whether some photos are the same person manually to find the corresponding information, but the efficiency is very low. Therefore, training neural network with a genetic algorithm to compare face information will be a good solution for handling the “lost” images.

Dr Sabrina Caldwell (2021) studied in her report about how to compare the data displayed on the face in the two pictures to determine whether the two pictures belong to the same person, she also provided two data sets about face features, and I chose the distance data set for this experiment. As Dr Sabrina Caldwell explained the way of analyzing facial feature is to identify fixed 14 marking points of each photo (2021), which indicates the position of right exocanthion, right endocanthion, left endocanthion, left exocanthion, nasion, sub-nasale, right alare, left alare, labiale superius, stomion, labiale inferius, pogonion, supramentale and menton, and the distance data set shows the distance between each parking point. Each column represents the distance between two mark-points. Each row is the data corresponding to two pictures. The last column using string “1” and “0” to indicates whether the two groups of data belong to the same person, which “1” means they are the same person, “0” means not.

2 Method

Genetic Algorithm (GA) is one of the most basic evolutionary algorithms. It is an optimization model that simulates Darwin's biological evolution theory. It was first proposed by Professor J. Holland in 1975. In the genetic algorithm, each individual of the population is a feasible solution in the solution space. By simulating the evolution process of organisms, the optimal solution is searched in the solution space. As Siddique and Tokhi (2001) stated in their study that there are many problems and limitations in training neural networks. Generally, neural network training is easily trapped in local minimums, and changes in network parameters (such as the number of hidden layer neurons and learning rate) are unreliable. Genetic algorithm performs better because it is good at exploring huge and complex search spaces to overcome these problems. In order to achieve higher accuracy of prediction, this project trained pruned neural network with genetic algorithm for analysis the facial features which represent in 36 images.

2.1 Data

The data set includes 36 records, and each record includes the distance of all the corresponding marked points of two pictures. It can be seen that there are 14 marked points, which are indicates the position of right exocanthion, right endocanthion, left endocanthion, left exocanthion, nasion, sub-nasale, right alare, left alare, labiale superius, stomion, labiale inferius, pogonion, supramentale and menton. Thus, there will be 14 * 13 = 182 distance data one by one, which are recorded in 182 columns respectively, represent the distance between two marked points. The data in the last column is “0” or “1” to indicate whether the two pictures represented by this row belong to the same person, “1” represent those two photos are belong to the same person but “0” represent they aren’t. Table 1 shows the general data format:

Table 1.

	Difference of distance of maker1 and 2 in photo1 and photo2	Output
1-1&1-2	192.22…	‘0’
1-2&1-3	59.22…	‘1’
1-1&1-3	132.05…	‘0’

But the data is provided in .xlsx format, it needs to be converted into a CSV file for model to read in, and the data should be preprocessed properly.

First, the xlrd library is imported to read the xlsx file, and loop all the sheets and convert them into a CSV file, which are stored in the folder named “csv” to facilitate subsequent reading. The processing codes are shown below:

```
# pre-processing the data, transfer the sheet into csv file

sheet_names=data.sheet_names() for worksheet_name in sheet_names:

# loop all the sheets


data_xls=pd.read_excel(file_path,worksheet_name,index_c ol=None)

# get the current direction
dir_path=os.path.abspath(os.path.dirname(file_path))

# save the sheet as a csv file and put it in the folder named "csv"

csv_path=dir_path+'\\csv\\'     if not os.path.exists(csv_path):

os.mkdir(csv_path)

data_xls.to_csv(csv_path+worksheet_name+'.csv’, index=None,encoding='utf-8')

data_dis = pd.read_csv(csv_path+'distances.csv',  header=None)
```

After reading the CSV file, we need to preprocess the data. There is no missing or abnormal data in this dataset, so we don't need to deal with it. Just delete the meaningless header of the first column and the first row, and then divide it into 25 rows for training model, and 11 rows for testing model. In addition to deleting the useless columns and rows, we still need to convert the output from string into numeric value using the function named “to_numeric” in pandas. Last but not the least, we used “OneHotEncoder” function in “sklearn.preprocessing” module to convert each element of the classification feature into a directly calculateable value. After all these preprocessing, the dataset is ready for the model training.

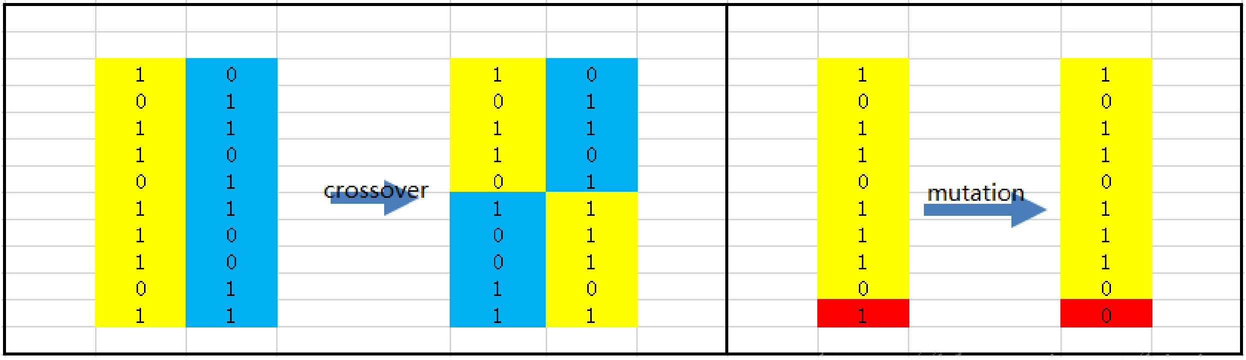
2.2 Neural network with genetic algorithm and pruning.

Genetic Algorithm is a computational model that simulates the biological evolution process of natural selection and genetic mechanism of Darwin's biological evolution theory. It is a method of searching for the optimal solution by simulating the natural evolution process. The genetic algorithm starts from the set of potential solutions (a population) that represents the problem, and the population is composed of a certain number of individuals coded by a gene (an individual can be understood as a kind of Feasible solution).

Each individual is actually an entity with chromosome characteristics. As the main carrier of genetic material, chromosome is a collection of multiple genes. Its internal performance (genotype) is a certain combination of genes, which determines the external performance of an individual's shape. For example, the characteristics of black hair are controlled by the chromosome and determined by a certain combination of genes for a characteristic.

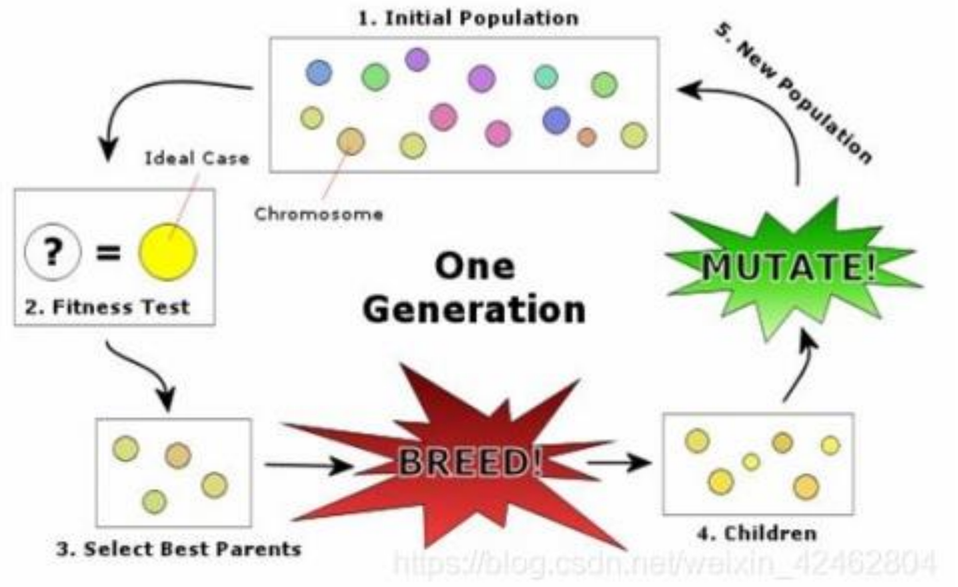
In neural network training, a lot of feasible solutions are randomly selected to form a population, and each solution (individual) is composed of a chromosome (such as 1001101101) containing only two genes :0 or 1. Chromosomes are different between different individuals, but they can exchange part of the gene fragment with each other and mutate themselves, for example, 0 becomes 1 randomly. The process of exchange and randomly change the values are crossover and mutation in genetic algorithm.

Image 1.



1	0	1	0	1	1
0	1	0	1	0	0
1	1	1	1	1	1
1	0	1	0	1	1
0	1	0	1	0	0
1	1	1	1	1	1
1	0	0	1	1	1
1	0	0	1	1	1
0	1	1	0	0	0
1	1	1	1	1	0

After all the steps above, we have a new population, and then start a new round of calculation. In each calculation, the moderately poor ones are eliminated, and then the surviving individuals have the right to cross or mutate. It goes over and over again, meeting a certain number of iterations or a certain threshold to exit the loop.



In detail, we built a basic neural network in the experiment for further training, which contains two hidden layers and uses sigmoid activation function. There are 182 neurons in the input layer, equal to the number of data attributes. The number of neurons in the hidden layer is set to 10, and the number of neurons in the final output layer is 2, representing two output results, which are 0 or 1. But we have to use genetic algorithm to train it, so we don’t have to set the learning rate and number of epochs.

Instead, we set parameters such as population number, mutation rate and number of genetic iterations in the genetic algorithm to gradually optimize the model. In the experiment, we set the mutation rate to 0.2, crossover rate to 0.4 and the retain rate to 0.4, the initial number of populations is 30 and the evolution process will be repeated for 1000 times.

After building the basic neural network, the project starts to build the pruning optimization algorithm. In this part, the project refers to the pruning optimization method introduces by T.D. Gedeon and D. Harris in the article "Network Reduction Techniques" of Brunel University, that find meaningless neurons, and then remove them. When two neurons are very similar, it is meaningless and can be removed. Thus, we loop all the units in hidden layers and use the cos angle formula of vector to calculate the angle between each vector. The formula is shown in Figure 1.

When the included angle is less than 15 ° If the two neurons are too similar, one of them can be removed by setting the vector to 0.

Figure 1

$$\cos < \vec{a}, \vec{b} >= \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|} = \frac{a_1 b_1 + a_2 b_2 + a_3 b_3}{\sqrt{a_1^2 + a_2^2 + a_3^2} \cdot \sqrt{b_1^2 + b_2^2 + b_3^2}}$$

3 Results and Discussion

After the model was completed, I recorded the data of 10 times of evolutionary process. Since I have recorded the accuracy of the model obtained every 10 times of evolution, for 1,000 times of evolution, there is a lot of data, and not every evolution will reflect the accuracy of the model, so I have summarized the data and streamlined, only the changed data is recorded. The following three tables include more typical situations in all experiments:

Table 1

iteration number	Current accuracy	Test accuracy (%)
1	64.00%	72.72727272727273
11	76.00%	72.72727272727273
21	84.00%	63.63636363636363
91	84.00%	72.72727272727273
111	88.00%	72.72727272727273
201	92.00%	72.72727272727273
261	92.00%	63.63636363636363
301	92.00%	72.72727272727273
501	92.00%	63.63636363636363
991	92.00%	72.72727272727273

Table 2

iteration number	Current accuracy	Test accuracy (%)
1	64.00%	72.72727272727273
11	88.00%	72.72727272727273
21	80.00%	54.54545454545454
41	88.00%	72.72727272727273
61	92.00%	72.72727272727273
991	92.00%	72.72727272727273

Table 3

iteration number	Current accuracy	Test accuracy (%)
1	36.00%	27.27272727272727
11	64.00%	72.72727272727273
31	88.00%	81.81818181818183
61	88.00%	63.63636363636363
81	92.00%	72.72727272727273
311	92.00%	81.81818181818183
991	92.00%	81.81818181818183

Excluding the data of these ten regular records, there was a special record appeared when I ran the code, it was an evolution failure case. I recorded this data, as shown in Table 4 below. It is a pity that the probability of this failure case is very small, and it is difficult to reproduce for researching the reason.

Table 4

iteration number	Current accuracy	Test accuracy (%)
1	36.00%	27.27272727272727
11	64.00%	72.72727272727273
31	84.00%	63.63636363636363
81	88.00%	63.63636363636363
101	96.00%	63.63636363636363
121	92.00%	63.63636363636363
191	96.00%	63.63636363636363
571	96.00%	72.72727272727273
861	100.00%	45.45454545454545
991	100.00%	45.45454545454545

In conclusion, the prediction accuracy of the neural network model which trained based on the genetic algorithm and pruning, basically increases with the iteration of genetics. As for the performance of the prediction model, the accuracy rate can reach up to 72.7% to 81.8%. Compared to the result that Dr Sabrina Caldwell recorded in the report, “This accuracy varied from 55% on a ‘bad’ split to 91% on a ‘good’ split.”, the performance of my model is relatively stable and better.

4 Conclusion and Future Work

In this experiment, the method of training neural network model based on both genetic algorithm and pruning optimizer was proposed, and the distance data set was used for training. As the result of this experiment shoed, this trained model can complete a certain degree of accurate prediction of facial data, up to about 72.7% or 81.8%.

Although the current neural network model can be applied to the currently used data set and perform pretty good, there are also some shortcomings that need to be further improved in the future. First of all, the data set is too small. There are only 36 pieces of data in total. 25 pieces of data are used to train the model and 11 pieces of data are used to test. The amount of training data and the test data are not large enough. It is difficult to guarantee the accuracy of this prediction model in practical application when facing larger amount of data. It is necessary to further expand the size of the data set and expose more defects to further optimize the model. Secondly, the dataset we used is imbalanced, but we did not deal the problem properly. The imbalance of sample data will result in too few features in the classification with a small sample size, and it is difficult to extract rules from it, and it is easy to produce over-reliance and limited data samples and lead to over-fitting problems. When it comes to new data, the accuracy of the model will be very poor. In the future work, the problem of sample imbalance can be solved by changing the weight of the majority class sample and the minority class sample data during training. This means that when training the model, the minority class sample is given a larger weight, which is the majority class sample is given a smaller weight.

References

1. Caldwell, S. (2021) "Human interpretability of AI-mediated comparisons of sparse natural person photos," CSTR-2021-1, School of Computing Technical Report, Australian National University.
2. Dutta, S, Shekhar,S, “Bond rating: A non-conservative application of neural networks,” IEEE Int Conf on Neural Networks, vol. II, pp. 443-450, 1988.
3. Sejnowski, TJ, Rosenberg, CR, “Parallel networks that learn to pronounce English text,” Complex Systems, vol. 1, pp. 145-168, 1987.
4. Siddique, M. N. H., & Tokhi, M. O. (2001, July). Training neural networks: backpropagation vs. genetic algorithms. In IJCNN'01. International Joint Conference on Neural Networks. Proceedings (Cat. No. 01CH37222) (Vol. 4, pp. 2673-2678). IEEE.
5. Holland, J. (1975). Adaptation in natural and artificial systems: an introductory analysis with application to biology. Control and artificial intelligence.
6. T.D. Gedeon, D. Harris. “Network Reduction Techniques,” Department of Computer Science, Brunel University.