Classifying Real Smiles from Human Eye Gaze Data Using A Neural Network with Evolutionary Algorithms

Guangming Zeng

The Australian National University, Research School of Computer Science
Acton, ACT, Australia
u7100771@anu.edu.au

Abstract. When human beings are stimulated by different smile videos (displayer's smiles), their reactions to seeing real smiles and fake smiles are also different. Ten Asian observers (6 males and 4 females) underwent this experiment, receiving real and fake smile stimulation, respectively. The pupil diameter was recorded from these observers. Preliminary analysis of pupil data showed that in the case of male observers, the pupil size of fake smile stimuli increased more than that of real smile stimuli, while the situation was the opposite for female observers. We will use the pupil diameter dataset with gender to train the model. In this work, we consider using a three-layer neural network to distinguish the authenticity of a smile from human eye gaze data. In addition, evolutionary algorithms are used to screen better models and explore more possibilities. But because the data set is too small, this leads to large fluctuations in our results. If we need to fully test the reliability of the model, we still need to expand the data set.

Keywords: Eye gaze tracking, smile stimuli, neural networks, evolutionary algorithms

1 Introduction

The human smile is one of the complex facial expressions. We usually think that a smile is a positive expression that conveys the joy of one's heart. But in fact, people do not just smile for happiness, they also smile when they are anxious, embarrassed, frustrated, surprised, etc. In many cultures, happiness is mandatory, especially in workplace culture. To show a welcoming attitude, employees must smile as part of their work. This is a false smile, which cannot truly reflect happiness. Understanding the meaning behind the smile can help us better understand the true intentions of the displayer.

Previous work conducted by researchers Hossain, Gedeon, Sankaranarayana, Apthorp, and Dawel from The Australian National University shows that the pupil reaction caused by a real smile is different from a fake smile, and the pupil reaction of men and women is opposite. Using the data from the experiments run by Hossain, Gedeon, Sankaranarayana, Apthorp, and Dawel, we will train several neural network models to distinguish between real smile stimuli and fake smile stimuli. Given insufficient data, we do not guarantee the results.

2 The Data

The data used to train the NN came from the previous work conducted by ANU researchers (Hossain, Gedeon, Sankaranarayana, Apthorp, and Dawel, 2016). Each column represents the dynamic data of the pupil diameter of an observer. There are a total of ten observers, and each observer will receive a real smile stimulus and a fake smile stimulus, so the data set has 20 columns. The pupil diameter is measured every 10 seconds, a total of 541 records are recorded, so the data set has 541 rows.

541 rows of data mean that 541 features can be used, which is too large. Therefore we decided to compress the data, accumulate every 10 rows, and record the sum as a new feature value, totalling 54 feature points.

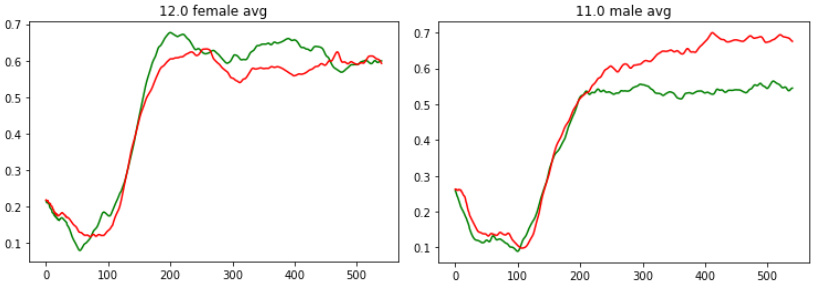

Fig. 1. Average change curve of pupil diameter when receiving smile stimulation (male, female)

The data of Fig. 1. are all from the same data set mentioned above. The pupil diameter changes of male and female observers were significantly different in the later stage of the smile stimulus. It can be seen that gender will be an important feature that affects the results. This feature requires us to compare the pupil data of real smile stimulation and fake smile

stimulation at the same time. In other words, we cannot indirectly express the gender feature from just a column of data. This means we need to add the 55th feature, gender.

ANU researchers' (Hossain, Gedeon, Sankaranarayana, Apthorp, and Dawel, 2016) reports indicate that the average pupil diameter of male observers receiving real smile stimulation is smaller than that of those receiving fake smile stimulation, while the opposite is true for female observers. Using this rule, we quickly added a new feature (gender) to each column.

So far, there are 55 features and 20 samples.

3 Method

Simple NN techniques were used in developing the model and testing its performance. A three-layer neural network was selected as the main model structure. In this structure, the Sigmoid is selected as the activation function for both hidden layers. In addition, we will use Dropout in both hidden layers to randomly delete some neurons to avoid overfitting. What's more, Cross-Entropy will be used as the loss function and Adam will be the optimizer of the model.

The hyperparameters of the model need to be carefully selected, including learning rate, dropout rate, epochs, the number of neurons in Hidden Layer 1, and the number of neurons in Hidden Layer 2. The range of hyperparameters is shown in Fig. 2.

Hyperparameter	**Min**	**Max**
Learning Rate	0.00001	0.01
Dropout Rate	0.25	0.75
Hidden Neurons 1	Input Neurons // 4	Input Neurons // 2
Hidden Neurons 2	2	Hidden Neurons 1 // 2
Epochs	200	1000
Input Neurons = 55 , Output Neurons = 2		

Fig. 2. The range of hyperparameters.

We randomly generate ten sets of hyperparameters that meet the above range, corresponding to ten different models. After model evaluation, we select the best set of fixed hyperparameters for subsequent experiments.

The model evaluation method is as follows:

1. We split the data set into a training set and a test set at a ratio of 8:2.
2. Subsequently, the K-fold Cross Validation method is applied to the training set. The data of one observer's real smile stimulus and fake smile stimulus is taken from the training set without repetition each time as the new test set (validation set), and the rest are used as the new training set.
3. Next, calculate the accuracy of the model on the test set using the new training set and the new test set.
4. Repeat step 2 and 3 until the training set is traversed.
5. After then, calculate the average accuracy rate (the accuracy of the validation set) acc1.
6. Calculate the accuracy of the model on the test set acc2 using the training set and the test set.
7. Finally, compare 10 sets of models, and select the best set of hyperparameters in which both acc1 and acc2 are at a high level.

So far, we have established a primary neural network model suitable for the data and found all the hyperparameters.

If we want to further optimize the model and seek more optimization possibilities, we can consider evolutionary algorithms. The algorithm mainly involves population initialization, natural selection, crossover, and mutation.

Before using this algorithm, we need to define DNA. With the help of PyTorch, we can extract the parameters in the neural network, including weights and bias. With the help of the NumPy library, we can quickly obtain the length and upper and lower limits of each parameter. In this way, we can construct real-numbered DNA sequences. However, this type of DNA is not conducive to handling mutations. To this end, we encode real-numbered DNA into longer binary DNA sequences. A binary DNA sequence of the same length can express a real number α on real-numbered DNA, ranging from 0 to 1 (both 0 and 1 can be taken). With the upper and lower limits obtained before, we can know the minimum Min and maximum Max of the original value of this position. Therefore the original value of this position is (Max-Min)* α + Min. With the help of the parameter template and the real-numbered DNA that has been decoded into the original value, we can get the model parameters (new weights and new bias).

With PyTorch, parameter templates can be directly obtained, and new parameters can be directly imported into the model. So far, we have built a bridge between binary DNA and model parameters.

The product of the total length of the parameter and the binary conversion length is the total length of the binary DNA. We also need to customize the population size, DNA crossover probability, mutation probability and generation size.

The evolutionary algorithm has officially started, the details are as follows:

1. Initialization: Generate POP_SIZE binary sequences. Each binary sequence is a random sequence of length DNA_SIZE.
2. Translated DNA: Convert each piece of binary DNA into different parameters.
3. Compute target function: Substitute each set of parameters into the model, generate different models, perform training, and return the value of the loss function.
4. Calculate fitness value: Obtain the fitness of DNA according to the loss value (the reciprocal of loss).
5. Selection: Select high-quality populations according to fitness.
6. Reproduction: Perform gene recombination with a set probability, and perform gene mutation with a set low probability. Obtain the next generation population.
7. Loop or terminate: Jump to step 2 and continue to execute until it reaches the expected generation size.
8. Extract model: Convert the obtained latest generation of binary DNA into model parameters, and finally get a new model.

The EA algorithm does not guarantee a better model, but it is still possible to give us a better solution.

4 Results

We generated ten sets of hyperparameters, and thus generated 10 different models. The details are shown in Fig. 3. We measure the performance of the model by calculating Cross-Validation and test accuracy.

Fig. 3. Model performance under different hyperparameters.

No.	learning_rate	dropout_rate	hidden_ neurons	hidden_ neurons2	epochs	Cross-Valiation	test accuracy
1	0.00442631	0.51650448	23	4	862	0.3125	0.75
2	0.00001853	0.28240800	17	7	679	0.5	0.5
3	0.00024036	0.25663719	23	8	656	0.3125	0.25
4	0.00228812	0.28100785	21	7	987	0.375	0.25
5	0.00017326	0.44335013	20	9	260	0.6875	0.75
6	0.00060210	0.55584729	19	6	549	0.4375	0.5
7	0.00029139	0.68407833	20	8	949	0.5625	0
8	0.00150108	0.48226477	17	4	874	0.375	0.25
9	0.00031584	0.54707048	18	6	509	0.625	0.5
10	0.00001323	0.69907686	22	6	912	0.5	0.5

Group 5 has the best model performance. Its validation set accuracy and test accuracy are both at the highest level. We will choose its hyperparameters.

Subsequently, we applied evolutionary algorithms to obtain the 50th generation model parameters with a population size of 20, a crossover rate of 0.8 and a mutation rate of 0.002.

```
Most fitted DNA:  [0 0 1 ... 1 0 0]
Most fitted DNA:  [0 1 1 ... 0 0 0]
Most fitted DNA:  [0 1 1 ... 0 0 0]
Most fitted DNA:  [0 1 0 ... 0 0 1]
Most fitted DNA:  [0 1 0 ... 0 0 1]
Most fitted DNA:  [0 1 0 ... 0 0 0]
Most fitted DNA:  [0 1 0 ... 0 0 1]
Most fitted DNA:  [0 1 0 ... 0 0 1]
Most fitted DNA:  [0 1 0 ... 0 1 1]
Most fitted DNA:  [0 1 0 ... 0 0 1]
Most fitted DNA:  [0 1 0 ... 0 0 1]
Most fitted DNA:  [0 1 0 ... 0 0 0]
Most fitted DNA:  [0 1 0 ... 0 1 0]
Most fitted DNA:  [0 1 0 ... 0 1 1]
Most fitted DNA:  [0 1 0 ... 0 0 1]
Most fitted DNA:  [0 1 0 ... 0 0 1]
Most fitted DNA:  [0 1 0 ... 0 0 1]
Most fitted DNA:  [0 1 0 ... 0 0 1]
Most fitted DNA:  [0 1 0 ... 0 0 1]
Most fitted DNA:  [0 1 0 ... 0 0 1]
```

Fig. 4. Most fitted DNA of each generation.

Evolutionary algorithms are time-consuming, and we cannot intervene in the process of execution. After the algorithm is executed, we obtain the binary DNA sequence with the sequence [0 1 0 ... 0 0 1] (the sequence is too long, replace it with an ellipsis). Using the weight and bias corresponding to the sequence, the accuracy of the new model can reach 75%.

5 Discussion

The three-layer neural network model can fully learn the effective information in the feature to a certain extent. The method of introducing dropout can prevent overfitting. Multiple methods are used together, so that the model not only avoids unfitting, but also avoids overfitting, and achieves a balance.

The ten sets of hyperparameters are randomly generated, which means that we may find different optimal hyperparameters in each experiment, which affects the performance of the next step. The best hyperparameters are not unique. But no matter what kind of hyperparameter, we can use it as long as the expected performance is achieved. We use the Cross Validation method to add an indicator to model evaluation so that we can measure model performance more comprehensively. Under the same conditions, the result given by the neural network is not certain. This means that we need to execute our code multiple times and eliminate outliers.

The evolutionary algorithm aims to find better model parameters, mainly optimizing the weights and biases of each layer. Since mutations are random, we cannot intervene. But we successfully screened out suitable DNA by measuring its fitness. Mutations have given various possibilities. And genetic recombination (crossover) further enriches the DNA sequence. After the DNA is defined, we can use selection, crossover, mutation and other functions almost without modification.

But we still cannot guarantee the effect of EA in this experiment. One is that EA itself has uncertainty because genetic mutations are completely random. Second, even in the same situation, it is difficult for neural networks to provide the same results. The third is that the data set is too small to provide robust data support.

Nevertheless, neural networks are still very powerful models. EA, as a way to exchange computation time for development time, is also easy to apply and simple to implement, although the algorithm requires a certain GPU performance.

6 Limitations and Future Work

At the end of the project, I found that the EA algorithm can be used not only for models but also for feature selection. In future work, I may expand the use of EA algorithms and help us better select features. Due to time constraints, we only tested 10 sets when looking for model hyperparameters. If time permits, we will expand. The biggest limitation of this experiment is that the amount of data is far from enough, resulting in very large fluctuations in our test results. Therefore, it is difficult for us to guarantee the results of models and algorithms. If possible, we need to expand the data. With the deepening of learning, we may also select models in the field of deep learning for research.

7 Conclusion

We use a three-layer neural network as the model, and also use the dropout function and K-fold Cross Validation method to avoid overfitting. Regarding the hyperparameters of the model itself, we have given sufficient degrees of freedom. As long as each hyperparameter is within the permitted range and the model under that hyperparameter performs well, it can be used as our alternative. Another focus of the research is evolutionary algorithms. We have designed a conversion method between model parameters and DNA. With only a small modification, we can use almost all functions in the evolutionary algorithm, including crossover, mutation, selection, etc. Limited by the size of the data, the highest accuracy of the model is around 75%.

References

1. Hossain, M. Z., Gedeon, T., Sankaranarayana, R., Apthorp, D., & Dawel, A. (2016). Pupillary responses of Asian observers in discriminating real from fake smiles: A preliminary study. In Measuring Behavior (pp. 170-176).
2. Rahman, J.S., Gedeon, T., Caldwell, S., Jones, R. and Jin, Z., 2021. Towards Effective Music Therapy for Mental Health Care Using Machine Learning Tools: Human Affective Reasoning and Music Genres. Journal of Artificial Intelligence and Soft Computing Research, 11(1), pp.5-20.

3. M.Z. Hossain1 , T.D. Gedeon1 , R. Sankaranarayana1 , D. Apthorp2 , A. Dawel. (2016). 11. Wang, J. T.-y., Spezio, M., Camerer, C. F. (2009). Pinocchio's Pupil: Using Eye tracking and Pupil Dilation To Understand Truth-telling and Deception in Sender-Receiver Games, American Economic Review, 100(3), 984-1007.
4. L.K. Milne1, T.D. Gedeon1 and A.K. Skidmore2. Classifying Dry Sclerophyll Forest From Augmented Satellite Data: Comparing Neural Network, Decision Tree & Maximum Likelihood
5. Luo Liangchen. (2018) From SGD to Adam-Overview of Deep Learning Optimization Algorithms (1) . Zhihu (https://zhuanlan.zhihu.com/p/32626442)