Anger Detection Analysis By Bimodal Distribution Removal Neural Network Model

Ziang Xu

Australian National University

Abstract. Facial expressions can reflect people's emotions, and anger is the strongest expression. People have two states of anger: real anger and fake anger. These two states correspond to different facial expressions. We extract and record human facial features. Using neural network to analyze the data set, a classification model is obtained to judge whether others are angry. We find that when using a simple neural network model, the accuracy of the training set is as high as 95%, while the accuracy of the test set is only 50%, there is over fitting. The accuracy of the training set is reduced to 88% and the accuracy of the test set is increased to 70% when the verification set is added to the cross verification training. The accuracy of the test set is improved to 80% by using the bimodal distribution algorithm. Then the network model is changed and the time series data is trained by using the recurrent neural network. The accuracy of the training set is 95% and the accuracy of the test set is 89%. Compared with the results of "are you really angry? Detecting displacement veracity as a proposed tool for interaction", the accuracy of neural network classification is closed. Finally, the shortcomings and improvements of the neural network model are analyzed.

Keywords: Anger Detection · Artificial Neural Network · Bimodal Distribution Removal · Recurrent Neural Network · Long Short Term Memory Networks.

1 Introduction

Anger is one of the most common human feelings. Anger is expressed through people's faces. But sometimes people choose to pretend to be angry. This is a face recognition expression that mimics the expression when angry. It can be that others mistakenly think he is angry. Lu, Tom, MD Zakir and Sabrina (2017) auger that the expression of behavioral anger, the expression of which does not have real feelings, attempts to manipulate the perceiver of their behavioral anger [1]. They did an experiment to collect people's angry faces in different states. Through experiments, the authors want to test how strong the ability of human beings to consciously detect the authenticity of anger is, and further test their ability to unconsciously detect anger, which is reflected in their pupil response.

We hope that by analyzing the data of face recognition in this experiment, we can figure out how to make computers recognize whether people are really angry or pretend to be angry. Zhihong, Jilin and others (2004) point out that the effective computing in HCI can track the user's effective state to obtain the corresponding feedback, which could judge the user's real emotion[18]. We can use neural network to build classification model, and then improve the accuracy of classification model to identify whether people are angry.

1.1 motivation of this research

By analysis this anger data set, we can make the machine judge people's anger by their facial expressions. Through this machine learning, we can better understand how to express personal feelings. It helps the AI machine to grasp user psychology more accurately.

1.2 the problem that I modelled

Participants will watch 20 videos at a time and judge whether the people in the video are really angry. In the data set, the subjects' eyes pupil changes when they watch the video and judge whether they are really angry. Data time series. The title of the data table records whether the people in this video are really angry. Use true and fake for both categories. In another data set, the deviations of participants' physical indicators were recorded. Participants were asked to wear E4 pneumothorax and measure skin conductance, blood volume pulse, and heart rate on hands and wrists. Then we use eye gaze tracker to measure eye gaze and pupil size. The data set is the mean value, standard deviation, and the largest two eigenvalues after principal component analysis of pupil size.

We design a classification problem using Angel dataset. First, the last label column in the dataset is used as a label for classification. The data is divided into two categories: true and fake. Then we delete the empty data in the dataset. Use tags to establish supervised learning. The neural network model of the learning input vector is established. Use the learning model to predict the classification of data in the test set.

1.3 Neural network

A neural network is a group of algorithms, loosely imitating the human brain, designed to recognize patterns [3]. They can analyze input data by machine sensing, labeling, or clustering the original input. Simon still adds that the neural network is digital and contained in the vector[3]. All the input data must be converted into a vector, including image, sound, text, or time series.

In my network model, I first use regularization to preprocess the data to make the sum of squares of each row 1. Then delete the useless part of the data and recode some of the data. Divide the data into features and labels. Then the validation set and test set are divided. The multi-layer neural network model is constructed to learn the training set, and the validation set is used to modify it. Finally, the test set is used to detect the accuracy and error of the model. Then, the model is improved by using recurrent neural network to improve the accuracy of model classification. the model also is improved by adding bimodal distribution removal to improve the robust of model.

2 Methods

First, we classify the data and divide them into the train set and the test set. Then the features and targets are divided for supervision training. Then preprocess the data. Construct an artificial neural network. Add bi model distribution removal to eliminate the data with large errors. Use k fold method to train cross-validation. Select the best model to test set. the design of our model is shown in Fig 1.

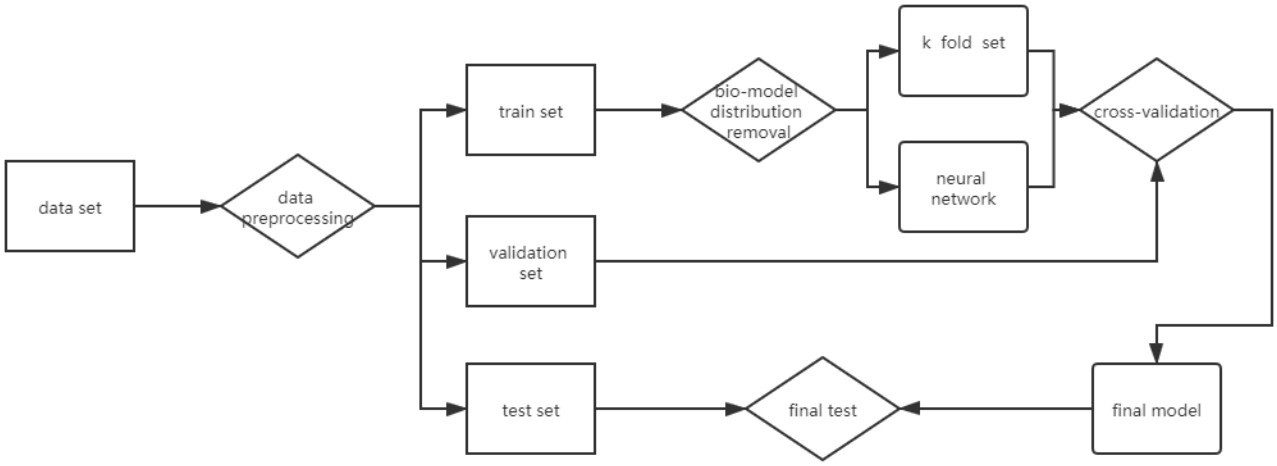

Fig. 1: The design of our model

2.1 Data Preprocessing

Data preprocessing refers to some data processing before the main processing. Because the data may contain incomplete, noisy, inconsistent, redundant, imbalance, outliers, and duplicate. These problems will lead to a large deviation in machine learning. So when there is low-quality data, we can significantly affect the quality and reliability of subsequent automatic discovery and decision-making through appropriate preprocessing steps [12].

Here I choose standard-scaler for data preprocessing. Standardization refers to the state that the value of a column of numerical characteristics in the training set is scaled to a mean value of 0 and a variance of 1. After standardization, the range of data is not necessarily between 0-1, and the data is not necessarily standard normal distribution [17], because the distribution of data will not change after standardization. If the data itself is normal distribution, it is standard normal distribution after standardization. Standardization can make the training process faster and improve convergence speed. Standardization can make the optimization process of the optimal solution smooth and easy to converge to the optimal solution. And improve the accuracy of the model, because, for the distance-based algorithm, the dimension of each feature directly determines the prediction results of the model. The neural network uses the error of reducing the distance to update the model parameters.

Then the data set is divided into a training set, verification set, and test set. The random number is used to randomly divide these three subsets from the dataset. A train set is used for data samples of model fitting [6]. The validation set is a set of samples set aside separately in the model training process[11], which can be used to adjust the super parameters of the model and to conduct a preliminary evaluation of the model's ability[4] . A test set is used to evaluate the generalization ability of the final model. But it can't be used as the basis of parameter adjustment, feature selection, and other algorithms related selection. Validation sets participate in the process of manual parameter adjustment (super parameter). In neural networks, a super parameter is the number of hidden units and it can be adjusted by cross-validation using validation sets[16] . Because the test set is independent of the training process, it can be used to examine the real classification data capability of a model.

In the improved experiment, I used to observe the size of the distance between the left and right pupils of the patient with time. This time, the data is a time series, which aims to classify anger. I did three different data preprocessing.

For the first time, I used the average pupil changes in all the experimenters in different videos. The data were averaged for all subjects and left and right pupil sizes. The data is simple, only 22 groups. I use 0 to fill in the default. Then the data is divided into a training set and test set. I use true and fake as labels, and convert them into 0 and 1. 22 groups of data are pupils distance on time series. They have been the same order of magnitude and do not need regularization [19]. All of them can be directly input into RNN model training with regularization.

The second time, I randomly selected all the participants from eight videos and collected the pupil spacing of their left and right eyes. Still use 0 to fill in the default value, using the first 180 data in the time series. Although there are 188-time data in total, the final sample part is unbalanced and there are many default values for the final part data. For the convenience of segmentation when the RNN model is input later, only 180 data are extracted. The left eye and right eye data of all the participants in the 8 videos were 320 sets of data (160 sets of the left eye and 160 sets of the right eye). I use true and fake as labels, and convert them to 0 and 1. but for this time, i will use normalization. because i use 0 instead of NAN. the zero and other pupils are not at the same order of magnitude. The data is regularized then input into RNN model training.

In the third experiment, according to the classification of the left eye and right eye, I read the pupil changes of all the experimenters for all the videos. I gave up all the empty time series in the data. Because this group of data failed to measure the pupil changes of the experimenter. Then I add up the left and right eye data of each experimenter on the same video to find the average. In this way, 390 sets of data (mean values of left and right eyes) are obtained, each set of data includes 186 values on time series. I still use 0 to fill in the default value, using the first 180 data in the time series. I use true and fake as labels, and convert them to 0 and 1. The data is regularized then input into RNN model training.

2.2 Artificial Neural Networks

Artificial neural network abstracts the neural network of the human brain from the perspective of information processing, establishes a simple model, and forms different networks according to different connection ways. Here we use a neural network with two hidden layers. There are n scalars in the input layer. Each neuron can take two states 1 or - 1, which are composed of N carrier neurons [10]. After the input layer excitation function, there is an output of hidden1 and it is transmitted to the first layer hidden layer. The corrected linear unit (Relu) is selected as the excitation function because it is more efficient in gradient descent and backpropagation, which can avoid gradient explosion and gradient disappearance. And the calculation of the Relu function is simple. Then, in the first hidden layer, using the linear model, through the Relu excitation function, the hidden2 vector is output. In the process, the batch norm is used to make the input of each layer of the neural network keep the same distribution in the process of deep neural network training. Then the hidden2 vector is input into the second hidden layer, and the linear model is also used to output the 2 output vector through the Relu excitation function. Using the softmax function to classify the first value of the output vector, we get two classes: "genuine" and "posted".

2.3 Cross Validation

Cross-validation in the given modeling samples, take out most of the samples to build the model, leave a small part of the samples to use the just built model for prediction, and calculate the prediction error of the small part of the samples and record their sum of squares. This process continues until all samples are predicted once and only once. Sum the square of the prediction error of each sample. Because in the actual training, the training results are usually good for the fitting degree of the training set, which can reach 95%, but the fitting degree of the data outside the training set is not so satisfactory. This may lead to two problems in the statistical model: the independency of residuals and the overfitting of data-dependent structures[13]. Therefore, we do not use all the data sets for training

but divide them into parts (this part does not participate in the training) to test the parameters generated by the training set and judge the degree of compliance of these parameters with the data outside the training set relatively objectively. Using the verification set to test the training model, as a performance index to evaluate the model classification. Then adjust the parameters based on this indicator, and you can perform early termination or dropout operations. This can reduce overfitting and increase the accuracy of the model on the test set. So for each cycle, first train the model with the train set, then adjust the parameters with the validation set to reduce overfitting.

2.4 Bimodal Distribution Removal

In the early stage of training, errors are almost normal distribution, which greatly reduces the errors of most training sets through model learning. However, there is a relatively high error mode in the later training period. This is the bimodal error distribution. At this time, we need to select the appropriate threshold to delete the patterns with too large error. bi model distribution removal uses the average value of error distribution as the first threshold [15]. All input patterns whose prediction error is greater than are included in the candidate set. Use the mean plus standard deviation as the second threshold. Delete input patterns with errors higher than the second threshold. Because the error image conforms to the bimodal distribution, the error can be reduced. Then set the endpoint of bimodal distribution removal, and the deletion of the endpoint will no longer take effect. If the deletion process has no termination point, it will continue to be deleted. Deleting too many input patterns results in insufficient information provided to the model. In this way, the accuracy of model learning will also decline. When the cumulative loss is lower than the threshold value, the bimodal distribution no longer exists, and the BDR process should be stopped actively. And patterns should be removed slowly, giving the network enough time to extract information from them [15].

2.5 Hyper Parameters and Optimizer

The neural network consists of seven input neurons, each neuron corresponds to a feature, and two output neurons correspond to two kinds of classified outputs. Firstly, SGD was selected as an optimizer for training. Then it is found that it is difficult to achieve more than 60% of the test set accuracy under any learning rate configuration because the random gradient descent is difficult to get rid of the local optimal solution. Then we chose Adam as the optimizer. Adam algorithm is based on adaptive low order moment estimation[8]. Because of the invariance of gradient's diagonal scaling, all Adam's are suitable for solving problems with large-scale data or parameters. In Adam algorithm, super parameters can be explained intuitively, and only a few parameters need to be adjusted. After using Adam, the classification accuracy of the training set can reach 95%, and the classification accuracy of the test set can reach 70%. The effect is better than the SGD optimizer. The result is better than 0.05 when the learning rate is 0.01 because the model training will lead to a more significant oscillation of the loss curve under the learning rate of 0.05.

2.6 Deep Learning

Deep learning is a method to get results by multi-level analysis and calculation [14]. The depth of the neural network is the number of layers from the input layer to the output layer. The deeper the network, the stronger the learning ability of the model. All can use deep learning to process large data sets. The common algorithms of deep learning are convolutional neural networks and cyclic neural networks [7]. CNN is a neural network that uses convolution instead of matrix multiplication. CNN is used to process multidimensional data, such as image processing. This experiment mainly deals with the pupil spacing in time series and classifies them. So it is more suitable to use a cyclic neural network.

2.7 Recurrent Neural Networks

Recurrent neural networks (RNNs) can ensure the information to exist continuously by continuously cycling the information[2]. It can be extended to longer time series and can also handle variable-length series. the figure 2 shows how a RNN works.

It can be seen that a is a group of neural networks (which can be understood as a network's self cycle), whose work is to receive and output constantly. It can be seen from the figure that allows the information to be continuously recycled internally so that it can ensure that the previous information is saved in each step of the calculation.

But RNN has a problem: long dependence. Long term dependence refers to the state of the current system, which may be affected by the state of the system a long time ago. It is an unsolvable problem in RNN. In theory, RNN

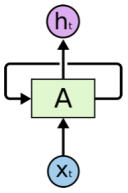

Fig. 2: The model of RNN [5]

can learn long-term information by adjusting parameters [20]. However, the conclusion in practice is that RNN can hardly learn this kind of information. RNN will lose the ability of learning time and expense information, leading to long-term memory failure. To solve this problem, I use LSTM to build an RNN network.

2.8 Long Short Term Memory Networks

Long short term memory networks (hereinafter referred to as Lstms), a special RNN network, is designed to solve the problem of long dependence.

Like RNN, Lstms has a chain structure. its structure can be seen on figure 3. But the repeat unit of Lstms is different from that of the standard RNN network, which has only one network layer. There are four network layers in it.

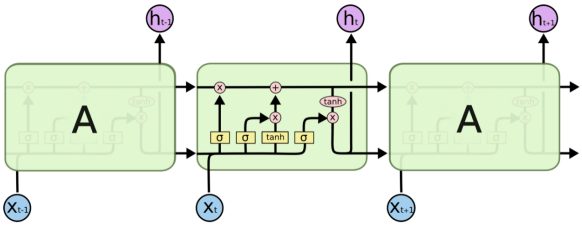

Fig. 3: The model of LSTMs [5]

First of all, the sigmoid unit of "forgetting gate" is used in LSTM to determine what information the cell state needs to discard. Then decide which information to update for the operation of the input door. The old cell information will be updated to new cell information. The updated rule is to forget part of the old cell information by forgetting gate selection and get new cell information by inputting part of the candidate cell information by gate selection. Finally, the information is passed through a sigmoid layer called the output gate to get the judgment conditions. Then, the state of the cell is passed through the "tanh" layer to get a vector between - 1 and 1, and the output of the RNN cell is obtained by multiplying the vector with the judgment conditions of the output gate.

3 Result and Discussion

3.1 Effectiveness of Neural network

We first need to determine the appropriate number of hidden layers and cells. When there is only one hidden layer, the model is difficult to learn. When there are two hidden layers, the model can learn all features well and make classification predictions. Then select the number of hidden cells. The right number can balance enough accuracy and complex network structure. We use 10, 15, and 20 hidden units for the two hidden layers respectively, and then test the accuracy of the artificial neural network classification. See Table 1 for the summary results. Although the accuracy of the result is very close, 20 * 10 hidden cells seem to be the best configuration of this model.

Then we choose the cross-entropy function to find the loss and use ADM as the optimizer. The k-fold method with k = 6 is used to separate the train set and validation set for cross-validation. Use the bimodal distribution removal method to delete data with large errors. After 500 repetitions of training, we use the validation set to detect the classification accuracy. The correct rates of the six groups were 77%, 57%, 62%, 81%, 57%, respectively. Select

index	1	2	3	4	5
Hidden Units	10*10	15*10	20*10	10*15	20*15
Accuracy	61	66	76	71	73

Table 1: Testing set accuracy obtained by various numbers of hidden units

the training model with the highest accuracy as the final model. Then output the change of accumulated loss in the training process. We can find that the loss curve of the training stage usually reaches its bottom in about 400 stages, and further training leads to an increase of loss, as shown in Figure 4.

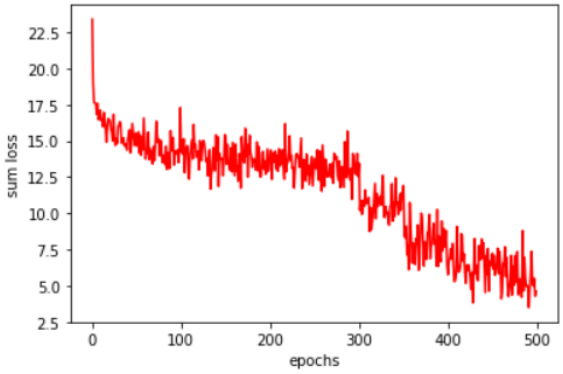

Fig. 4: Loss over epochs under Adam, learning rate = 0.01

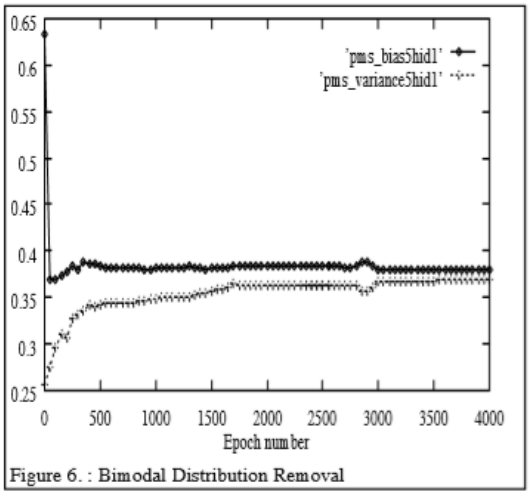

Fig. 5: Loss over epochs cite from "BIMODAL DISTRIBUTION REMOVAL " [15]

3.2 Effectiveness of Bimodal Distribution Removal

The bimodal distribution removal is added to the constructed artificial neural network with cross-validation. We found that as the training went on, some big error data was deleted. Using variance threshold (0.01) as the termination point, beyond which we will stop bimodal distribution removal to avoid excessive deletion of input mode. Then the standard deviation from mean to standard is used as the deleted standard. If the error of a pattern is greater than

the threshold value (0.01), it is classified as an exceptional value. Compare the outliers with the deviation from mean to standard var. Delete data larger than the deviation.

We do bimodal distribution removal for all six training sets and count the average error after every 100 epochs, the accuracy of the model classification, and the number of remaining input modes. The results are shown in Table 2. The results show that when the number of remaining input patterns is too small, the accuracy of model classification will also decline. The number of initial input modes is about 330. When using the bimodal distribution removal method to delete about 50 data, the effect is the best. At this time, the number of remaining input modes is about 280, the accuracy of training is about 95%, and the accuracy of the verification set is 77%. The model is the best. Therefore, the BDR process can effectively remove the outliers in the training set, to give better prediction results for the test set. But pay attention to the stop node of the BDR process. When the BDR process is deleted too much, the input mode is too few, and the accuracy of prediction results will still decline.

Epoch	1st train set			2nd train set		
	Average loss	Accuracy(%)	remain Input Size	Average loss	Accuracy(%)	remain Input Size
1	0.7502	68	337	0.6099	65	327
101	0.4051	89	314	0.3850	84	301
201	0.1844	99	281	0.5472	87	301
301	0.6158	100	252	0.4636	86	301
401	0.4646	100	245	0.2036	92	285
Epoch	3rd train set			4th train set		
	Average loss	Accuracy (%)	remain Input Size	Average loss	Accuracy(%)	remain Input Size
1	0.7214	67	331	0.6040	62	342
101	0.6552	82	307	0.4591	85	312
201	0.3315	87	307	0.9197	88	312
301	0.5265	90	307	0.2954	90	312
401	0.3095	97	278	0.1428	97	278
Epoch	5th train set			6th train set		
	Average loss	Accuracy(%)	remain Input Size	Average loss	Accuracy(%)	remain Input Size
1	0.7534	68	334	0.5853	70	329
101	0.2794	85	311	0.6236	87	300
201	0.3738	86	311	0.3167	92	300
301	0.3505	89	311	0.2968	92	300
401	0.4982	93	294	0.0745	99	276

Table 2: Experiment Group Result

3.3 Effectiveness of Recurrent Neural Networks

When we use the ANN model to train data, the final prediction accuracy is low because of the small number of data and the mismatch of the model. We have used cross-validation to solve the overfitting and BDR algorithm to reduce the bimodal distribution of data. Because of the poor training results, we will replace the neural network model. We use the RNN model, and use the change of pupil distance in time as new data. So we have more raw data. We use LSTM to build RNN and linear output. We try different datasets and compare their results.

Using the mean data We used the mean pupil spacing of all the experimenters as the usage data for a video. We only have 22 sets of data. The experiment set was brought into RNN model training, the input time series length was set to 9, each group of data was input in 18 batches. After 500 epochs training, the predicted value and the average value of all groups of target data are output every 50 times. The experimental results show that the average loss is 0.7419 and the accuracy is 27% at the first epoch. When the 51th epoch, the average loss is 0.0058, and accuracy is 100%. After that, the average loss is 0.0000, and the accuracy is 100%. Observe the test set again. The average loss was 0.0671, and the accuracy was 57%

The effect of RNN training is good, and the correct rate of training set can reach 100%. But then there is overfitting. Because we only used 22 sets of data. In order to increase the accuracy of the test set, we use the pupil changes of each experimenter as the input data. This increases the amount of input data.

Using the sample from different video We selected eight video data from the original data set, and collected the left pupil and right pupil data of the experimenter respectively. Intercept the first 180 data of time series. This time use a different time series length. Set the input time series length to 20, and input each group of data in 9 batches. At this time, our input data amount reaches 257 groups. Input the data into the RNN model for 50 epochs training, and output the predicted value and the average value of all groups of target data every 5 times. The experimental results show that the average loss is 0.5724 and the accuracy is 66% at the first epoch. At the 5th epoch, the average loss is 0.0004, and accuracy is 100%. After that, the average loss is 0.0000, and the accuracy is 100%. Observe the test set again. The average loss is 0.000 and the accuracy is 100%. The curve of average loss during training is shown in Figure 6.

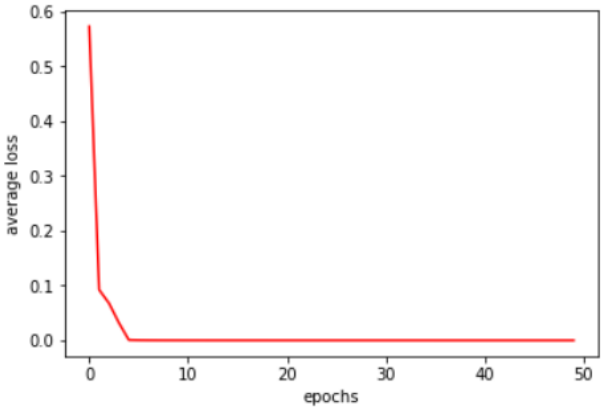

Fig. 6: Loss over epochs under Adam, learning rate = 0.01

RNN model can get the correct model in a small amount of cyclic training. And the accuracy of the test set is very high, even up to 100%. Different time series lengths were used in this experiment. According to the results, a time series length of 20 can meet the training requirements. Compared with the previous experiments, the different length of input time series has little effect on the results. Because our data is the pupil change value of the experimenter when watching the video. There is no obvious time cycle for the data. This experiment standardized the data and reduced the impact of 0 data, but the effect was not obvious. This experiment uses some video data, which has some limitations. Next, we use all the original data, that is, all the experimenters' pupil changes for all the videos.

Using the whole data We use the whole data set to take the average of the left and right pupil and input it into the RNN model. The time series length is set to 20. The number of cycles is set to 200. Other parameters remain unchanged. The training results are shown in Table 3. The train loss are shown in Figure 7. The accuracy of the training set is at 95%. The accuracy of the test set is 89%. It shows that there are enough data in training, which reduces the overfitting problem.

The results show that the RNN model can be used to classify this dataset. The curve surface of train loss rises many times after falling rapidly. It shows that the stability of the data set is poor in RNN model training. There is a peak of loss at the 100th epoch. The loss curve is similar to a bimodal distribution. But in many experiments, there are many peaks or no second peak in the loss curve. You can try to use the BDR algorithm or other algorithms again to enhance the stability of the model. The details will be discussed in future work.

3.4 Comparison with Other Paper

We compared the training results of the model with the experiments of "bimodal distribution removal"[15]. The experimental results of Slade and Gedeon are shown in Figure 5. In the figure, the abscissa is the epochs trained and the ordinate is the average error of the predicted value of the model. It can be seen that the bimodel distribution

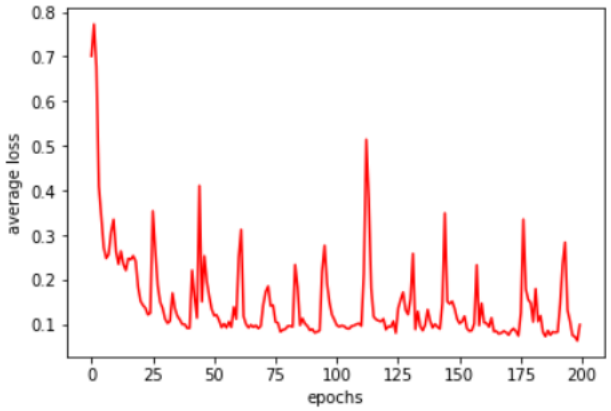

Fig. 7: Loss over epochs under Adam, learning rate = 0.01

Epoch	Average loss	Train Accuracy(%)
1	0.7013	50
21	0.1521	91
41	0.0919	95
81	0.0962	97
121	0.0886	95
161	0.1042	95

Table 3: Experiment whole data Result

removal process improves the general artificial neural network, because the deviation and variance in the training process are low, and it is the result of data-driven termination conditions. Compared with our experiments, Slade's and Gedeon's models are more stable, and the float of loss is smaller. But their iterations are much more than ours. It shows that the model tends to be more stable with the number of iterations increasing. Comparing the deviation value of the BDR method with the normal backpropagation shows that even if the BDR method performs well, noise data points are also useful in this case. When there are some noise data points, or there are relatively many data points, the prediction accuracy will increase.

After replacing ANN with RNN, the prediction accuracy was further improved to 89%. The prediction accuracy of Slade and Gedeon is 95%. It shows that the prediction accuracy of the model is almost the same under the same data set. But Slade's and Gedeon's models are still more stable. The stability of our model needs to be further strengthened.

4 Conclusion

We discuss the application of artificial neural networks (contains ANN and RNN) to data classification and the effectiveness of the bimodal distribution removal method in improving the generalization of the neural network. Our data comes from people's judgment of angry faces. We discuss whether cross-validation can effectively improve classification accuracy and reduce overfitting. The premise of BDR application is analyzed, that is, the bimodal distribution of training error comes from the mixture of two unimodal distributions. The number of input modes and prediction accuracy, as well as the selection of the best performance termination point in the BDR process, are discussed. We find that cross-validation and the BDR process can improve the classification accuracy and effectively reduce the scale of the neural network. after using the original data set, we get a high accuracy RNN model. it means RNN can deal with time series classification problems. But the number of data will impact the RNN model. The time series length also can influent the accuracy of the RNN model. By changing hyperparameters, the RNN model can get the best result.

Future work can explore whether the BDR process adapts to other machine learning methods, so that they can produce better results, or whether they can organically reduce the performance differences of selected features. We have obtained that the BDR process reduces the difference between selected features and directly affects the prediction results of the model. Error's variance threshold may not be a good termination point for the BDR process

because it can cause an excessive deletion. We can use the tilt test to determine whether the bimodal distribution still exists as a termination mechanism. When the bimodal distribution disappears, the BDR process is ended, and whether it improves the prediction performance is observed. Also, it is the future work to enhance the robustness of model training and the stability of prediction. We may use pruning and cascading networks to increase the robustness of artificial neural networks. we also can use tune fine to improve the RNNs model. we can add the BDR process or dynamic surface control during the RNN training [9]. it may can improve the robust of model.

References

1. Chen, l., Gedeon, T., Hossain, M., Caldwell, S.: Are you really angry?: detecting emotion veracity as a proposed tool for interaction. pp. 412–416 (11 2017). https://doi.org/10.1145/3152771.3156147
2. Graves, A., Mohamed, A.r., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE international conference on acoustics, speech and signal processing. pp. 6645–6649. IEEE (2013)
3. Haykin, S.: Neural networks: a comprehensive foundation. Prentice Hall PTR (1994)
4. James, G., Witten, D., Hastie, T., Tibshirani, R.: An introduction to statistical learning, vol. 112. Springer (2013)
5. Kamath, U., Liu, J., Whitaker, J.: Deep learning for nlp and speech recognition. Springer (2019)
6. Langford, J.: Combining train set and test set bounds. In: MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-. pp. 331–338 (2002)
7. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)
8. Liu, Y., Ling, X., Shi, Z., Lv, M., Fang, J., Zhang, L.: A survey on particle swarm optimization algorithms for multimodal function optimization. JSW **6**(12), 2449–2455 (2011)
9. Miao, B., Li, T., Luo, W.: A dsc and mlp based robust adaptive nn tracking control for underwater vehicle. Neurocomputing **111**, 184–189 (2013)
10. Mladenov, V., Koprinkova-Hristova, P., Palm, G., Villa, A., Apolloni, B., Kasabov, N.K.: Artificial Neural Networks and Machine Learning–ICANN 2013: 23rd International Conference on Artificial Neural Networks, Sofia, Bulgaria, September 10-13, 2013, Proceedings, vol. 8131. Springer (2013)
11. Prechelt, L.: Automatic early stopping using cross validation: quantifying the criteria. Neural Networks **11**(4), 761–767 (1998)
12. Ramírez-Gallego, S., Krawczyk, B., García, S., Woźniak, M., Herrera, F.: A survey on data preprocessing for data stream mining: Current status and future directions. Neurocomputing **239**, 39–57 (2017)
13. Roberts, D.R., Bahn, V., Ciuti, S., Boyce, M.S., Elith, J., Guillera-Arroita, G., Hauenstein, S., Lahoz-Monfort, J.J., Schröder, B., Thuiller, W., et al.: Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. Ecography **40**(8), 913–929 (2017)
14. Schmidhuber, J.: Deep learning in neural networks: An overview. Neural networks **61**, 85–117 (2015)
15. Slade, P., Gedeon, T.D.: Bimodal distribution removal. In: International Workshop on Artificial Neural Networks. pp. 249–254. Springer (1993)
16. Terry, A.M., McGregor, P.K.: Census and monitoring based on individually identifiable vocalizations: the role of neural networks. In: Animal Conservation forum. vol. 5, pp. 103–111. Cambridge University Press (2002)
17. Van Wegberg, M.: Standardization process of systems technologies: creating a balance between competition and cooperation. Technology Analysis & Strategic Management **16**(4), 457–478 (2004)
18. Zeng, Z., Tu, J., Liu, M., Zhang, T., Rizzolo, N., Zhang, Z., Huang, T.S., Roth, D., Levinson, S.: Bimodal hci-related affect recognition. In: Proceedings of the 6th International Conference on Multimodal Interfaces. p. 137–143. ICMI '04, Association for Computing Machinery, New York, NY, USA (2004). https://doi.org/10.1145/1027933.1027958, https://doi.org/10.1145/1027933.1027958
19. Zhang, J., Man, K.: Time series prediction using rnn in multi-dimension embedding phase space. In: SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 98CH36218). vol. 2, pp. 1868–1873. IEEE (1998)
20. Zhang, Y., Xiong, R., He, H., Pecht, M.G.: Long short-term memory recurrent neural network for remaining useful life prediction of lithium-ion batteries. IEEE Transactions on Vehicular Technology **67**(7), 5695–5705 (2018)