Information Visualisation Analysis based on Neural Network Model With Bimodal Distribution Removal technology

Jun Han

Australian National University

Abstract. Information visualization analysis has been studied a lot in the past, the previous linear programming model can not accurately fit the relationship between human eye features and radial or hierarchical visualization. So the neural network is used to deal with the complex nonlinear correlation problem. It is useful to use the neural network model to solve the problem of information visualization. By adding bimodal distribution removal and deleting the data with too large error, the classification accuracy of the neural network is improved. In this paper, the validity of the BDR process for information visualization is tested. This paper discusses the application of cross-validation in training neural networks with small data sets. This paper also uses the recurrent Neural Network to deal with time series data. Then this paper analysis the result and the improvement.

Keywords: information visualisation · Artificial Neural Network · Bimodal Distribution Removal.· Deep Learning · Recurrent Neural Network · Long Short Term Memory Networks

1 Introduction

1.1 Background

Md Zakir Hossain Et al (2018) compare the efficiency of visual methods with an experimental method [9] . Using the eye tracker system to record the observer at 60hz, observe the chair's movement and adjust its position, track the position of the laptop screen, and then calibrate the eye tracker to begin testing. When monitoring at the same time, the observers should cooperate as much as possible, and the monitoring environment should be adjusted according to different observers to reduce the impact. In a challenging environment, observe a visual public data as an observer, and then record the observer's questions. From all the observers, 24 observers who were unfamiliar with the data were selected to answer six questions, and the accuracy rate was more than 80%.After the data is processed by neural network, the visual data is output, which is beneficial to the processing efficiency of large amount of data.

Visualization interface is widely used in modern society, which can provide users with a high level of interaction, extract and collect raw data, and then describe big data with the minimum cost[20] . Combined with certain data, this way of expression can transmit information to the observer to the greatest extent. The advantage of this way is that the observer has strong reception ability, so how to collect, process and draw a large amount of original data is a key problem.

1.2 Motivation

It is foreseeable that visualization will become the most important way for people to obtain information in the future. Car navigation is a kind of, for example, in the process of driving, and road conditions need to pay attention to the vehicle driving conditions guarantee the safe operation of the vehicle, so car navigation requires a great deal of information transmission efficiency, we can through the machine learning of human information processing, to deal with traffic navigation data such as the first, and then directly pass good data through image processing ways to the driver, ensure the driver can get enough information in a short time.

1.3 the Modelled Problem

I built a classification problem. I extracted 11 features from the dataset as the input mode of the model. I choose response time (s), no. of fixations, average saccade duration (MS), average fixation duration (MS), mean, STD, Max, min, range, first derivative, second derivative as features. Because these eleven features come from each observer's gaze and saccade, they are recorded and converted into data through the eye tribe (https://the eye tribe.com/) remote eye tracker system. Then extract a list of labels as the target for supervised learning. I choose the interface

parameter as the label. There are two categories, radial and hierarchical, to study the two categories. I choose interface as the label because the main research object of the experiment is two visualisations (radial and hierarchical). Radial visualization is a complex set of interrelated entities, highlighting the associated structure, while hierarchical visualization highlights each entity of major interest [9].

1.4 Neural network

In fact, it has been proved [25] [4]in the case of consistent architecture, feedforward network is consistent. If the performance of the non-parametric estimator is to be guaranteed, its superiority can be guaranteed only when the data set is infinite. Therefore, for a limited sample, the estimation of the number of non-parameters will have an impact on the sample, which leads to the variance of the estimator, which can be controlled by introducing part of the prior data into the estimator as a parameter. In the case that a sufficient number cannot be obtained as an example, a certain deviation can be allowed: a small number of cells can be added as dynamic nodes, with a large deviation [3][25][7].

Therefore, if there is a lot of "noisy" in the training center, use the check-out detection. In any case, if there is a large number of "noisy" in the training center, use LTS.Or the BDR approach can improve generalization by eliminating noise sources.

2 Method

First, divide the data set into train, validation, and test. Select the feature and label from the data and conduct a classification problem. Then the data is preprocessed, standardized, and the unused features are deleted. Then write recurrent neural network with LSTM and linear out [1], set the loss function and use adm as optimizer. And then you train it 200 times using a neural network, to get a rough model. Then cross validation is used to remove the overfit [15], along with the train set and validation set. Then the bimodal distribution removal function was written and added to the train to delete the data with large errors. Finally, the training model is obtained. Use the test set to verify its accuracy. The algorithm flow chart is shown in Figure 1.

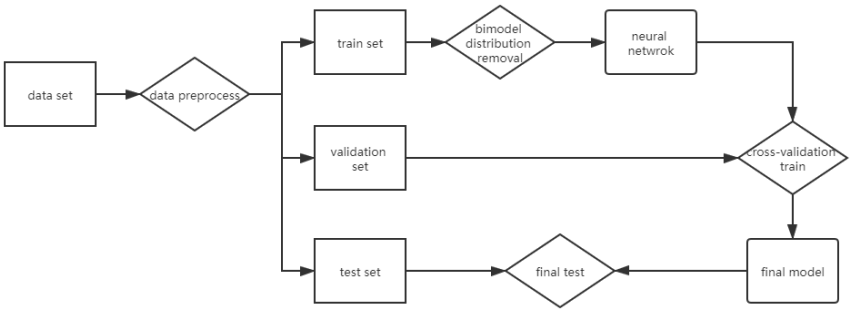

Fig. 1: The design of our model

2.1 Data Preprocessing

We choose the data of information visualization as the data set. The data set contains the judgment of 25 subjects on 12 groups of information and records the changes of pupil size with time in their left and right eyes. 12 groups of information can be divided into two categories: radial and hierarchical. Then we delete some useless personal information of the experimenter, such as age, gender, major, and so on. Filter out identifiers unrelated to classification tasks, such as index-q1 of information. For the missing value (Nan) for a dataset, we use 0 paddings. The missing value makes the uncertainty in the system more significant, and the certainty component contained in the system more difficult to grasp [2]. Data with null values can mess up the mining process, resulting in unreliable output [24]. At this time, we will simply fill it with 0 and discuss other methods later.

Then the data set is divided into three parts: training set, verification set, and test set. Using a random function to generate a a random index, all data are randomly divided into three groups. We use training sets to train models.

The validation set is used to test the generated model and adjust the super parameter of the model [23]. Use test sets to evaluate the final predictive power of the model.

Before neural network training, data need to be preprocessed. Data preprocessing includes data cleaning, normalization, transformation, feature extraction,, and selection. The result of data preprocessing is the final training set [13]. The reason why we need data preprocessing is to ensure the quality of instance data. If there is a lot of irrelevant, redundant information or noise, unreliable data in the data, a lot of errors will be produced in future training.

We first select the data. The number of original data sets is too large. There are not only 600 groups of data from 25 people to 12 groups of problems, but also the time series of each group of data is up to 2700. However, it is found that there are many default values when observing the later data of time series. The possible reason is that the time of data acquisition experiment is too long, and the experimenter fails to complete all 2700 pupil data acquisition. So we only select the first half of the time series, that is, 1500 times of time acquisition data. To reduce the training time of the model, we randomly selected 8 out of 25 experimenters and used their data to train the neural network.

We use standardized data for data preprocessing. First, use a standard scaler to process the data by subtracting the average and then dividing by the variance (or standard deviation). After data standardization, the data conforms to the standard normal distribution [18]. Then use the minmaxscaler method to scale the feature to a specific range between the given minimum and maximum values, or you can also convert the maximum absolute value of each feature to the unit size. This method is a linear transformation of the original data, which places the data in the middle of [0,1]. This also reduces the impact of missing data to 0. The preprocessed data can improve the prediction accuracy of the model.

2.2 Artificial Neural Network

The artificial neural network classification algorithm is composed of multi-layer neuron structure, each layer of neurons has input and output [8]. The artificial neural network consists of three layers: an input layer, an output layer, and a hidden layer. After data normalization, we need to build a neural network. The topological structure of our neural network is one input layer, two hidden layers, and one output layer. The connection mode is NN. Linear(). The excitation function of each layer is the Relu function. The NN. Batchnorm1d() method adds normal normalization to hidden layers. In this way, the network can be constrained to automatically adjust the intensity of the standardization in the training process, so as to accelerate the training speed and reduce the cost of weight initialization. Then 10, 15, and 20 hidden units are used in two hidden layers. The input layer passes in 11 features. The output layer outputs a two-dimensional vector, in which the first dimension is the label of the classification after passing the softmax function. Then through the correction of training samples, the weight matrix value of the neural network is obtained.a Finally, the trained artificial neural network model is obtained.

2.3 Bimodal Distribution Removal

For the error of bimodal distribution, bimodal distribution removal can be used to remove some large error modes. When selecting the mode to delete, the first step is to calculate the average variance errors of all modes in the training set [22]. Record high error peaks that are greater than the average in the subset of outliers. Then the average BSS and standard deviation MSS of the outlier set is calculated. From these two statistics, you can decide which modes to permanently delete from the training set. Delete error is greater than BSS + alpha * MSS mode, where alpha is a number between 0 and 1 [22]. Repeat every 50 epochs so that the network can be updated. Finally, variance error can be used as the termination condition of training. Once the variance error is less than a constant (usually 0.01), the training stops. Note that patterns are slowly removed to prevent overfitting and to avoid too few input patterns.

2.4 Cross Validation

Our data set has 288 data in total, and only about 190 data are divided into the train set. When the data set is small, we use cross-validation to "make full use of" the limited data to find the appropriate model parameters and prevent overfitting [21]. The training set is used to train the classifier, and then the verification set is used to verify the model. The final classification accuracy is recorded as the performance index of the classifier. Cross-validation can not only evaluate the model performance but also constantly adjust the super parameters to make the model perform best on the verification set. In the process of training, model. Train () and model. Eval () are used to train and verify each epoch.

2.5 Loss Function and Optimizer

The loss function in the feedforward neural network calculates the difference between the target value and the predicted value. The cross-entropy is used as the loss function in classification experiments. The evaluation is based on comparing the prediction generated by our artificial neural network with the actual target results[17]. We use the ADM function as the optimizer. The optimizer of the ADM function increases momentum (exponentially weighted average), the velocity in the direction of constant gradient becomes faster, and the velocity in the direction of gradient change becomes slower.

2.6 Recurrent Neural Network

Deep learning is to use a multi-level neural network, its network results are more complex and diverse. Convolution network (CNN), recurrent neural network (RNN), self encoder, and deep Boltzmann machine are commonly used in deep learning. CNN is a kind of feedforward neural network that contains convolution computation and has deep structure [14]. It has the ability of representation learning and can classify the input information according to its hierarchical structure. CNN is mainly used in image processing and natural language processing. The data of this experiment is about the classification of time series data, so it is not suitable to use CNN. The self encoder is a kind of artificial neural network that can learn the efficient representation of input data through unsupervised learning [12]. In this experiment, we have supervised learning to classify. The self encoder is not suitable to build the model of this experiment. The deep Boltzmann machine is a deep learning model based on the restricted Boltzmann machine (RBM) [19]. The algorithm complexity of deep Boltzmann's reasoning learning process is too high to be effectively applied to large-scale learning problems. Because we have a long time series, the deep Boltzmann opportunity leads to the training process is too complex, so we do not use it to build the network. RNN can deal with time series [16], so it is chosen as the network model.

The recurrent neural network (RNN) is a kind of recurrent neural network that takes sequence data as input and recursion in the evolution direction of the sequence, and all nodes (loop units) are linked by a chain. Because RNN has memory, parameter sharing, and Turing completion, it has some advantages in learning nonlinear features of sequences. So RNN can be used to process the time series data of this experiment. The network structure of RNN is shown in Figure 2. Bidirectional RNN, Bi RNN, and long short term memory networks (LSTM) are common recurrent neural networks.

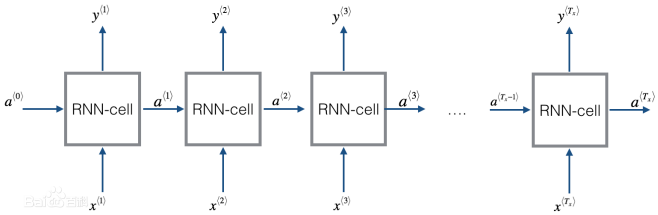

Fig. 2: The design of our model [11]

But RNN cannot solve the problem of long-term dependence. Long term dependency refers to the current system state, which may be affected by the system state a long time ago. In theory, RNN can learn long-term information by adjusting parameters. 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, we use LSTM to build the RNN network.

2.7 Long Short-Term Memory Network

Long short term memory (LSTM) is a kind of special RNN, which is mainly used to solve the problem of gradient disappearance and gradient explosion in the process of long sequence training [6]. In short, LSTM can perform better in a longer sequence than RNN. Compared with RNN, which has only one transfer state, LSTM has two transfer states, one cell state, and one hidden state. The specific structure of LSTM is shown in Figure 3. The left network in Figure 3 is the general RNN, and the right network is LSTM. Compared to RNN with only one transfer state, LSTM has two transfer states. Unlike RNN, LSTM is not a simple way of memory superposition. LSTM selects memory.

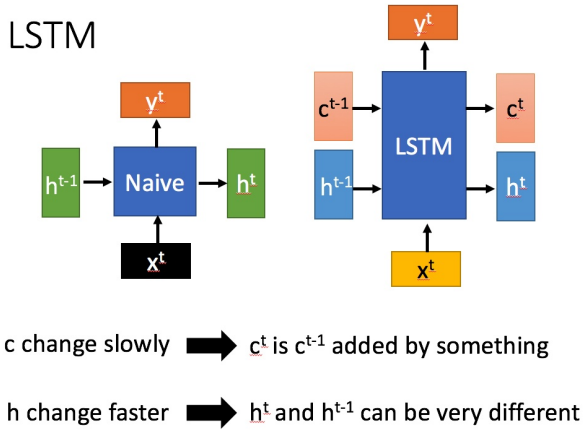

Fig. 3: The design of our model [11]

There are three stages in LSTM[5]: 1. Forgetting. This stage is mainly to selectively forget the input from the previous node. Use forget gating to control which part of the previous state needs to be forgotten. 2. Select the memory stage. In this stage, the input is selectively recorded and controlled by the selected gating signal . 3. Output stage. This stage will determine which information will be taken as the output of the current state . The output changes through a tanh activation function.

In this experiment, LSTM is used as the internal network structure of RNN, and a linear output layer is used. We set the number of hidden layers of RNN to 6. In the LSTM model, the input data must be a batch of data. We use the same meaning for batch data in LSTM and data loader.

3 Result and Discussion

3.1 Effectiveness of Different Neural network

We detect the effect of the different numbers of hidden layers on neural network training. We find that a single hidden layer cannot fit all features. So we use the neural network of two hidden layers. Then we try different number of hidden layer units, using 10 * 10, 15 * 10, 20 * 10, 25 * 10, and 20 * 15 respectively See Table 1 for the results. When the number reaches 20 * 10, the prediction effect does not change greatly. So for this dataset, the number of cells in the hidden layer of 20 * 10 is enough.

Then we detect the number of hidden layer units using 20 * 10, use ADM as the optimizer, and use the neural network training condition with a learning rate of 0.01. As shown in Figure 2. The average error of the training set decreases rapidly with the increase of training times. When epochs are 200, the decline curve of average error tends to be flat. But there are ups and downs. Because of the cross-validation of the validation set, the parameters of the training model are adjusted. Lead to changes in the model. However, due to the large validation error, the model can not be determined, so the loss of the training set is not stable.

index	1	2	3	4	5
Hidden Units	10*10	15*10	20*10	25*10	20*15
Accuracy(%)	49	55	61	58	63

Table 1: Testing accuracy by various numbers of hidden units

3.2 Effectiveness of Bimodal Distribution Removal

We do two different experiments, using the same structure of the neural network, different training sets, to detect the effect of BDR process on the prediction accuracy. In 1000 training epochs, every 100 times we count the correct

rate of the training set, the correct rate of the validation set, and the number of remaining input modes. The results are shown in Table 2. We find that the BDR process can help training accuracy. It can delete the data with too large error, reduce the overall error of the model, and increase the accuracy of classification. But with the input mode less than a certain number, the model error rises again. At this time, because the input mode is too few and the model training is not complete, the prediction accuracy decreases. You should adjust the threshold or change the stop condition.

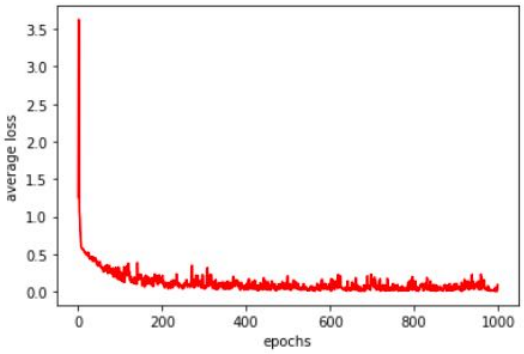

Fig. 4: Average Loss over epochs under Adam, learning rate = 0.01

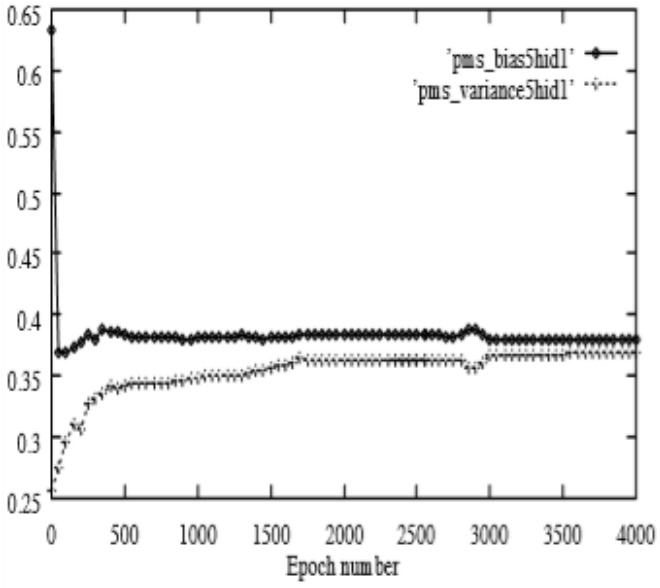

Fig. 5: Loss over epochs cite from "BIMODAL DISTRIBUTION REMOVAL [22]"

3.3 Effectiveness of Recurrent Neural Network

After the ANN and BDR processes are tried again, the prediction effect of the model is still poor. So we use the original data, use LSTM to build RNN, and train again. In order to save the training time, we selected the data of 8 experimenters. we seperately use 400 time series data and 1500 time series data into the RNN training process.

The sequence length of data is set to 20, and each group of 400 data is divided into 20 batches and input into RNN for training. The training epochs is 200, and the average loss and accuracy rate of each 20 output training sets. The results are shown in Table 2. The loss curve of the training set is shown in Figure 4. The average loss of the test set is 0.6989, and the accuracy is 41%.

Epoch	1st train set			2nd train set		
	Train Accuracy(%)	Validation Accuracy(%)	remain Input Size	Train Accuracy(%)	Validation Accuracy(%)	remain Input Size
1	85	57	181	78	46	192
101	93	57	181	87	64	192
201	93	64	173	96	72	186
301	98	52	173	97	68	182
401	98	52	173	97	70	182
501	99	64	173	93	64	181
601	99	64	172	99	62	179
701	95	57	172	98	62	179
801	98	60	172	97	66	179
901	99	57	171	97	62	178

Table 2: Experiment Group Result

Epoch	Average loss	Train Accuracy(%)
1	0.6356	48
21	0.5971	46
41	0.8211	54
61	0.7242	54
81	0.5226	48
101	0.7676	48
121	0.8082	57
141	0.6423	48
161	0.7749	54
181	0.9029	49

Table 3: Experiment 400 data Result

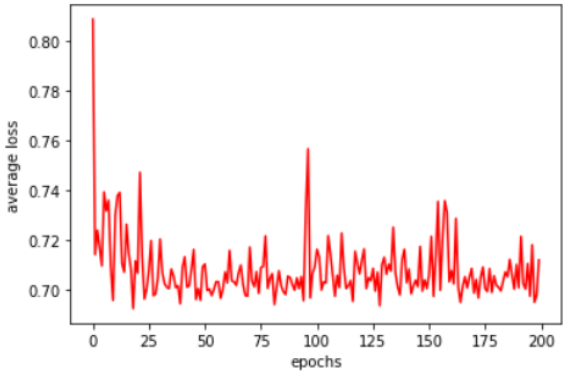

Fig. 6: Averge loss over epochs for 400 time series"

The sequence length of data is set to 30, and each group of 1500 data is divided into 50 batches and input into RNN for training. The training epochs is 200, and the average loss and accuracy rate of each 20 output training sets. The results are shown in Table 3. The loss curve of the training set is shown in Figure 5. The average loss of the test set is 0.6942, and the accuracy is 57%.

Epoch	Average loss	Train Accuracy(%)
1	0.7129	39
21	0.6933	41
41	0.8251	48
61	0.6519	49
81	0.7780	52
101	0.5743	58
121	0.6849	55
141	0.7693	61
161	0.7849	59
181	0.8681	63

Table 4: Experiment 1500 data Result

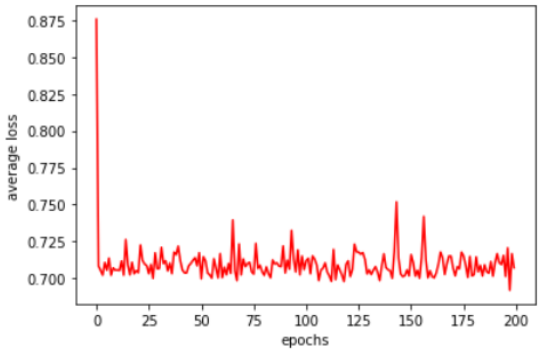

Fig. 7: Averge loss over epochs for 1500 time series"

Then we use BDR process during the RNN training. for every 50 epochs, the BDR will check the loss and delete the data with large loss. we still use 400 time series data to save time. we also use cross validation to improve the training. Every 20 epochs, we show the train set accuracy and test set accuracy. Then we use cross validation to improve RNN model. the result shows on table 5. the loss curve shows on figure 8.

From the results of three times model training, the RNN model can not fully learn all the data features, and the accuracy of classification is generally low. The classification accuracy is only 50% when using 400-time series data and 60% when using 1500 time series data. The correct rate of classification is only 55%. Generally speaking, ANN with the BDR process and RNN with the BDR process are not good at information visualization dataset classification.

The reason for the poor RNN effect may be that the time cycle of the data is not consistent with the sequence length set by the model. You can try data input of different sequence length multiple times. The reason may also be the processing of missing values in the data. We only use 0 instead of Nan (data missing value) in data preprocessing. However, 0 does not match the data before and after the time period. The loss of model training fails to converge. Instead of Nan, the nearest neighbor data of the missing time period can be used. Looking at the loss curve, we find that the loss will not decline until it drops to 0.7. This shows that the model training reaches the local optimal solution of loss. K-fold can be used for many experiments to break the local minimum. However, the reason why loss can't continue to decline may also be because of a saddle point. Because it's not a neural network problem to fall into local optimum. It's very difficult to form a local minimum in a very high-dimensional space because local minimum

Epoch	Average loss	Train Accuracy	Remain data numbers	Test Accuracy(%)
1	2.0557	58	60	54
21	4.4969	53	60	54
41	0.9090	52	60	54
61	0.0517	52	60	50
81	0.3793	48	60	54
101	1.1882	53	60	54
121	1.2754	42	60	54
141	0.8460	45	60	57
161	0.2843	52	60	50
181	0.2118	55	60	54

Table 5: Experiment 400 data Result with BDR

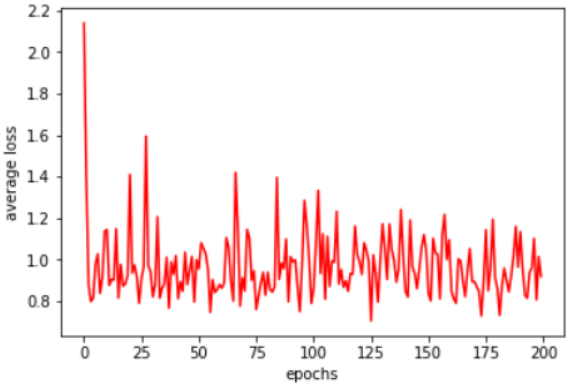

Fig. 8: Averge loss over epochs for 1500 time series”

requires the function to be local minimum in all dimensions. The function will fall on a saddle point [10]. We have used tanh function as the excitation function of the RNN output layer, so we can guarantee the homeomorphism of original space and transformed space. We can also try to use the colah function as the excitation function of the output layer.

3.4 Comparison with other papers

The training results of our model are compared with those in "bimodal distribution removal". The model training loss after BDR processing of "bimodal distribution removal" is shown in Figure 3[22]. It can be seen that the loss curves of the two models are stable in the 0.2 and 0.3 attachment regions. It shows that the training accuracy of our model meets the expectation. But the classification accuracy of our validation set and test set is low. This is caused by the model overfitting. Although cross-validation has been used to slow overfitting. However, due to the lack of data, the slowdown effect is not obvious. Therefore, k-fold can be used for cross-validation in the future to further slow down overfitting.

4 Conclusion

In this paper, we discuss how to enhance the learning of artificial neural networks by removing bimodal distribution. We use standardized data preprocessing and cross-validation to improve the learning accuracy of neural networks and avoid overfitting. The influence of a different number of hidden units on model training is compared. Then, based on the bimodal distribution of training error, the BDR process is used to delete the mode with too large error and remove the noise. The relationship between the number of schema deletions and the prediction accuracy is compared. Then we use RNN model to train the time series data. we still use the BDR on RNN training process. Then we compare the different lengths of series and different numbers of input data.

Future work: the reduced number of patterns in the BDR process directly affects the prediction results of the model. Too little deletion will cause too much noise. Too much deletion will lead to too few input modes and incomplete learning. Try to use the dip test to observe the peak structure of the error. When the bimodal disappears, the BDR process is stopped. This may achieve a more accurate model deletion. On the other hand, we only use simple cross-validation, but not cross thinking. K-fold cross-validation can be improved in the future. The original data is divided into k groups (generally average), and each subset data is verified once respectively, and the rest k-1 group subset data is used as a training set so that K models can be obtained. Then select the best model. This can effectively avoid the occurrence of overfitting and underfitting, and the final results are more convincing. we also can use other data values instead of zero to paddling NAN. We could change the network forward and backward methods to deal with local minimal.

References

1. Abdel-Nasser, M., Mahmoud, K.: Accurate photovoltaic power forecasting models using deep lstm-rnn. Neural Computing and Applications **31**(7), 2727–2740 (2019)
2. Acuna, E., Rodriguez, C.: The treatment of missing values and its effect on classifier accuracy. In: Classification, clustering, and data mining applications, pp. 639–647. Springer (2004)
3. Ash, T.: Dynamic node creation in backpropagation networks. Connection science **1**(4), 365–375 (1989)
4. Gallant, A.R., White, H.: A unified theory of estimation and inference for nonlinear dynamic models. Blackwell (1988)
5. Gensler, A., Henze, J., Sick, B., Raabe, N.: Deep learning for solar power forecasting—an approach using autoencoder and lstm neural networks. In: 2016 IEEE international conference on systems, man, and cybernetics (SMC). pp. 002858–002865. IEEE (2016)
6. Gers, F.A., Schraudolph, N.N., Schmidhuber, J.: Learning precise timing with lstm recurrent networks. Journal of machine learning research **3**(Aug), 115–143 (2002)
7. Harris, D., Gedeon, T.: Adaptive insertion of units in feed-forward neural networks. In: 4th Int. Conf. on Neural Networks and their Applications (1991)
8. Hassoun, M.H., et al.: Fundamentals of artificial neural networks. MIT press (1995)
9. 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)
10. Jin, W., Li, Z.J., Wei, L.S., Zhen, H.: The improvements of bp neural network learning algorithm. In: WCC 2000-ICSP 2000. 2000 5th international conference on signal processing proceedings. 16th world computer congress 2000. vol. 3, pp. 1647–1649. IEEE (2000)

11. Kamath, U., Liu, J., Whitaker, J.: Deep learning for nlp and speech recognition. Springer (2019)
12. Kitaev, N., Klein, D.: Constituency parsing with a self-attentive encoder. arXiv preprint arXiv:1805.01052 (2018)
13. Kotsiantis, S., Kanellopoulos, D., Pintelas, P.: Data preprocessing for supervised leaning. International Journal of Computer Science **1**(2), 111–117 (2006)
14. Lo, S.C.B., Li, H., Wang, Y., Kinnard, L., Freedman, M.T.: A multiple circular path convolution neural network system for detection of mammographic masses. IEEE transactions on medical imaging **21**(2), 150–158 (2002)
15. Ng, A.Y., et al.: Preventing" overfitting" of cross-validation data. In: ICML. vol. 97, pp. 245–253. Citeseer (1997)
16. Pascanu, R., Mikolov, T., Bengio, Y.: On the difficulty of training recurrent neural networks. In: International conference on machine learning. pp. 1310–1318 (2013)
17. Quan, K.: Bimodal distribution removal and genetic algorithm in neural network for breast cancer diagnosis. arXiv preprint arXiv:2002.08729 (2020)
18. Quemy, A.: Data pipeline selection and optimization. In: DOLAP (2019)
19. Salakhutdinov, R., Hinton, G.: Deep boltzmann machines. In: Artificial intelligence and statistics. pp. 448–455 (2009)
20. Sedig, K., Rowhani, S., Morey, J., Liang, H.N.: Application of information visualization techniques to the design of a mathematical mindtool: A usability study. Information Visualization **2**(3), 142–159 (2003)
21. Shao, J.: Linear model selection by cross-validation. Journal of the American statistical Association **88**(422), 486–494 (1993)
22. Slade, P., Gedeon, T.D.: Bimodal distribution removal. In: International Workshop on Artificial Neural Networks. pp. 249–254. Springer (1993)
23. 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)
24. Twisk, J., de Vente, W.: Attrition in longitudinal studies: how to deal with missing data. Journal of clinical epidemiology **55**(4), 329–337 (2002)
25. White, H.: Learning in artificial neural networks: A statistical perspective. Neural computation **1**(4), 425–464 (1989)