IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Revue d'Intelligence Artificielle
Vol. 34, No. 1, February 2020, pp. 51-57
Journal homepage: http://iieta.org/journals/ria

Big Data Mining Optimization Algorithm Based on Machine Learning Model

Changyi Jiao

School of Economics and Management, Hebi Polytechnic, Hebi 458030, China

Corresponding Author Email: jcy586@126.com

https://doi.org/10.18280/ria.340107

Received: 3 July 2019
Accepted: 8 November 2019

Keywords:
big data, machine learning, BP neural network, least mean square, imbalanced classification, batch learning

ABSTRACT

In a big data mining optimization algorithm, the classification algorithm plays an important role. At present, there are many popular classification algorithms based on machine learning. Aiming at the problems of existing big data classification algorithms, two improved strategies and implementation methods are proposed in this paper. First of all, before the data training, the orders of magnitude of the original data are normalized to achieve better data pre-processing and classification. Then with the least mean square algorithm and the BP neural network classification algorithm as the basic algorithms, an improved batch learning BP algorithm is designed, based on the rules of batch learning. The experimental results indicate that the improved batch learning BP algorithm can better solve the imbalanced classification problem in big data.

1. INTRODUCTION

With the continuous technological development, the ways data are used and collected are significantly improved, but the explosive growth of data has also caused a lot of problems. For example, various kinds of data, when mixed together, can hardly reflect their original values. How to extract information from the massive data and convert it to useful and valuable knowledge, and how to improve the speed of information extraction in the process, have become urgent problems to be solved.

Data mining refers to the process of searching the hidden information from massive data through an intelligent algorithm [1]. It is a decision support process mainly based on artificial intelligence and machine learning. Through highly automatic analysis of massive data, it can perform inductive reasoning, and extract potential patterns.

Data mining takes the following steps: information collection and data preparation, data mining, result interpretation and evaluation [2]. In the stage of information collection and data preparation, we need to summarize some representative features, and then use appropriate methods to collect information and store it in the database. Data preparation, as a key link in the data mining process, plays a role in the quality and speed of information extraction in the subsequent mining process.

Data preparation can also be subdivided into data integration, data reduction and data pre-processing. Data integration refers to integrating data with different features to achieve data sharing; data reduction refers to selecting core data in the data set to reduce the amount of subsequent data processing; and data pre-processing means improving the accuracy and speed of data mining by minimizing the conversion of data dimensions and forms.

The main function of data mining is to find specific patterns of data, including descriptive and predictive patterns. The descriptive pattern mining task is to characterize the common features of data, while the predictive one is to summarize data and make predictions.

Classification is a big branch of the data mining technology. It uses given class labels to analyze the objects in the dataset. Usually a training set is used, where all the objects have been associated with the known class labels. The classification algorithm learns from the training set and constructs a model, and then uses the model to classify new objects.

This paper mainly studies the classification algorithm in data mining. Based on the framework of artificial neural network, it designs an improved batch learning BP algorithm by using the least mean square algorithm and the traditional BP algorithm according to the rules of batch learning.

2. RELATED WORKS

Classification is an important method in data mining. It is a process of finding the classifier, with some constraints used to assign objects in the dataset to different classes [3]. Classification uses given class labels to analyze the objects in the dataset. Usually a training set is used, where all objects have been associated with the known class labels. The classification algorithm learns from the training set and constructs a model, and then applies such model to classify new objects. In other words, classification is a process of classifying data into different classes [4].

Currently there have been a number of mature classification algorithms, such as K-Nearest Neighbor [5], Support Vector Machine [6], Decision Tree [7], Artificial Neural Network [8] and so on. These classification algorithms have been successfully applied in people's life, production, transportation and other fields. Data in these fields share a common feature, which is data imbalance. Such problem is usually called the imbalanced data classification problem [9].

Imbalanced classification is a common classification problem. It means that there are far fewer samples in one class of a data set than in other classes. In an imbalanced classification problem, the traditional classification algorithm

can easily result in over-fitting due to the imbalance of the classes, and thus the effect of data classification is not desired. In order to improve the over-fitting problem, Karaçali and Krim [10] proposed a structural risk minimization strategy, under which, over-fitting can be improved to some extent although it cannot be avoided completely. In view of the shortcomings of the empirical risk minimization strategy and the structural risk minimization strategy, several other methods have been proposed successively to improve over-fitting, but they still cannot solve this problem well [11-14]. Brownfield et al. [15] compared seven different non-parametric classifiers - radial basis function neural network, multilayer perceptron neural network, support vector machine, classification and regression tree, Chi-square automatic interaction detection, quick, unbiased and efficient statistical tree algorithm and random forest. The results showed that random forest has the highest accuracy, sensitivity and specificity. Palanisamy et al. [16] then established a three-layer neural network to classify data sets.

However, each algorithm above has its advantages and disadvantages, and there is no data mining classification algorithm that can solve all problems perfectly. Neural network algorithm has strong robustness and fault tolerance to noise data. Compared with other algorithms, it has a strong ability to process noise data. Moreover, because of its strong learning ability, it can easily find the classification mode of the original data. At the same time, neural network can continuously improve its performance to improve the classification accuracy and prediction ability. Therefore, this paper chooses the neural network classification algorithm as the research object, and designs a representative BP neural network algorithm to improve the performance for big data analysis.

3. AN IMPROVED METHOD OF DATA PRE-PROCESSING FOR CLASSIFICATION ALGORITHM

Data pre-processing is a very important step in the process of data mining. Before data analysis, it is important to know how to express data and ensure the quality of data. This section mainly studies the improved strategy and method of data pre-processing in the classification algorithm based on artificial neural network.

For the original data, if there is a lot of irrelevant and redundant information, or there are noise and unreliable data, knowledge discovery in the training stage will become more difficult, and as a result, data preparation and filtering will take a lot of time, which is why data pre-processing is important. The purpose of data pre-processing in a classification algorithm is to obtain the final training set.

Data pre-processing consists of the following steps: data cleaning, data integration, data normalization and data transformation [17]. The process of data pre-processing is shown in Figure 1.

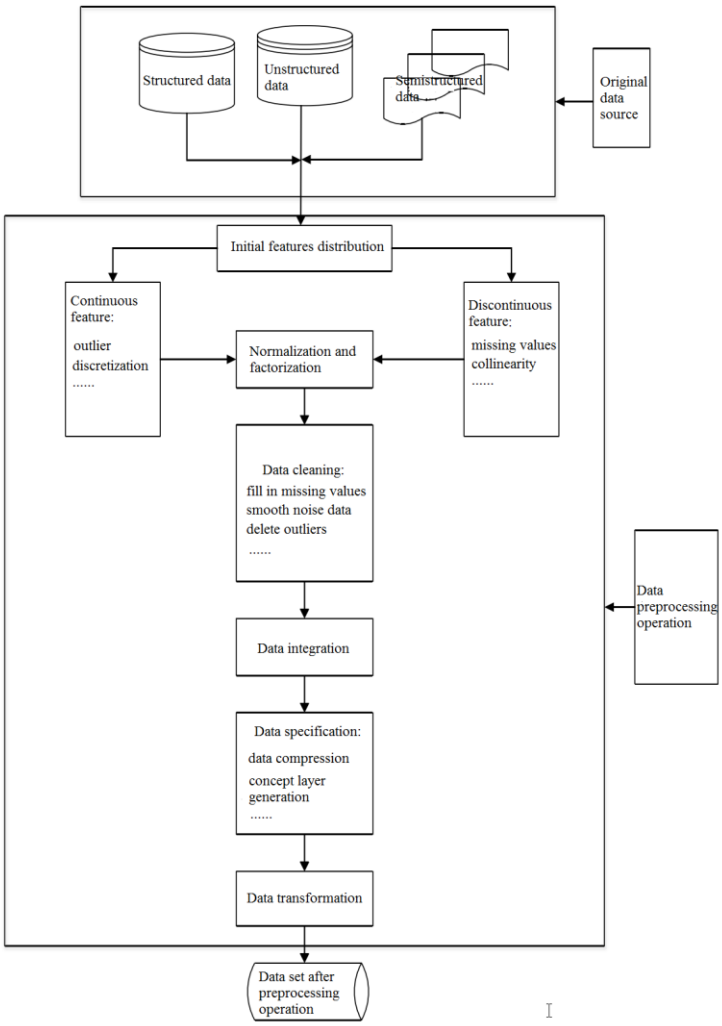

Figure 1. Data pre-processing process

The improvement strategy of data pre-processing in this paper is to normalize the original data before the training of data so that the values of all data are of the same order of magnitude. This strategy will directly affect the effectiveness and accuracy of the subsequent data mining process.

The original data are usually disordered, and those in the training set may contain data at all levels, or are even different on the order of magnitude. As different levels of features have different impacts on the neural network, training of such data will not only take a very long time, but also produce inaccurate training results. Therefore, before pre-processing the input data, we need to perform some operations on the input data and normalize the training data so as to facilitate data pre-processing and analysis of different types of samples and improve the speed of training and accuracy of the prediction model.

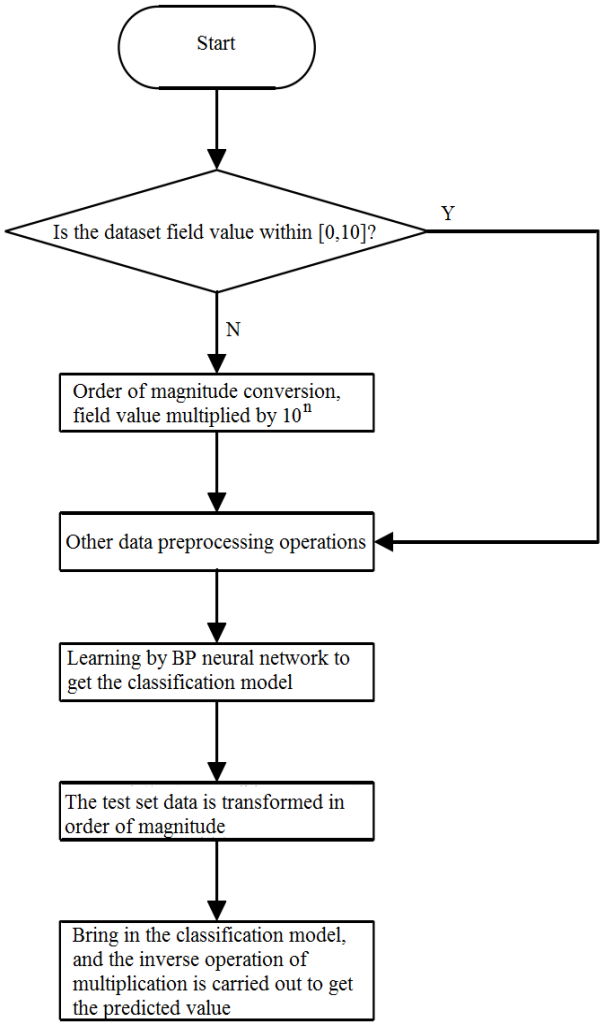

Figure 2. Process flow of data pre-processing for the classification algorithm

Those data with great differences in magnitude may be numerically normalized, but there are still great differences among them even after normalization, so the prediction model obtained through numerical normalization cannot fit with the reality very well. For an artificial neural network, the pre-processing of input data will not have any negative impact on the training network, because data training is only for numerical analysis. Therefore, this paper proposes normalizing the data of different levels within the range of (0,10) before data input to reduce the impact of large data on the training network, then preprocessing the normalized data by the pre-processing method and finally training the data. The specific processing steps are as follows:

Step 1: extract the eigenvalues of each sample, which are different in size. Each eigenvalue needs to be transformed in the order of magnitude to make it within the range of (0, 10). This range is selected based on the existing research on data pre-processing [18, 19].

Step 2: multiply the extracted eigenvalues by 10^n, where the value of n depends on the difference between the orders of magnitude.

Step 3: obtain the classification model. When the test set is used in testing, the data of the test set must also be transformed into the same order of magnitude. Perform prediction and comparison after the transformation.

Step 4: For the output results after training with this model, perform an inverse transformation, that is, an inverse operation of the multiplication so that the real prediction data corresponding to the original data can be obtained.

The mathematical expression of the corresponding processing method is as follows:

Training set D contains P samples, and each sample has M features and M output values. X_i represents the input vector, Y_i the output vector, and Z_i the converted data feature value, whose value is between 0 and 10. $X_p = (X_{i1}, X_{i2}, \dots, X_{iM})\ i = 1,2, \dots, P$, $Y_i = (Y_{i1}, Y_{i2}, \dots, Y_{iM})\ i = 1,2, \dots, P$, and $Z_i = X_{iP} * 10^n$, $Z_{iP} \in [0,10]$, where the value of n depends on the difference between orders of magnitude.

The specific process flow of data pre-processing for the classification algorithm is illustrated in Figure 2.

4. AN CLASSIFICATION ALGORITHM BASED ON THE IMPROVED BATCH LEARNING BP NEURAL NETWORK

The batch learning BP algorithm proposed in this paper is built on the back propagation algorithm, and the least mean square algorithm is used as the linear adaptive filtering algorithm to modify the parameters to make the output response closest to the expected one.

The multi-layer perceptron (MLP) model is the most widely applied neural network structure used in classification methods. The main objective of the proposed improved algorithm is to obtain the best variable parameters of the MLP model, so that the model can apply the batch learning BP algorithm to classify the given data set [20].

4.1 Adaptive filtering algorithm based on least mean square

The least mean square (LMS) algorithm is an adaptive learning algorithm. By establishing a cost function $E(w)$ that is continuously differentiable to the weight vector w, it is used to describe the difference between the output response and the expected one. The purpose is to find an optimal weight vector w^*, and for any w, $E(w^*) \leq E(w)$.

Self-adaption means automatically adjusting the processing method, sequence, parameters, boundary conditions or constraints according to the features of the processed data in the process of processing and analysis, to make them adaptive to the statistical distribution and structural features of the

processed data so as to achieve the best processing effect. There is an adjustment time before the system enters into stability. The gradient descent optimization method can be used to make the cost function surface rapidly converge to the local minimum value, which is controlled by the convergence factor δ of the algorithm. If δ is increased in the LMS algorithm, the adjustment time will be reduced in a certain range, but the system will not converge when it exceeds this range.

Based on the LMS learning rules, the weight vector v is defined as follows:

$$v_{q+1} = v_q + \Delta v_q = v_q - \delta \nabla E(v_q) = v_q - \delta \frac{\partial E}{\partial v}|_q \quad (1)$$

The LMS algorithm is updated in real time. Every time the data in the training set are trained, the weight vector q in the network will be modified, so the subscript q can be omitted.

In a calculation cycle, the training data pair (x, d) is randomly selected from the training data, where x is the input, d is the mathematical expectation, and the data pair is substituted into the activation function $f(u)$. According to the weighted input of neurons, u can be expressed in matrix form as follows:

$$u = v^T x \quad (2)$$

Substitute u into the activation function, and the output of neurons is obtained:

$$o = f(u) = f(v^T x) \quad (3)$$

In order to determine the error size of the weight v of the data pair (x, d), the expected output d is calculated and compared with the output o of the neuron directly. The error signal is the difference between the expected response and the output.

$$err = d - o = d - f(v^T x) \quad (4)$$

The error err will be used to measure and adjust the weights of neurons, so as to minimize the cost function E of the overall network weights.

The formula of E is defined as follows:

$$\boldsymbol{E} = \frac{1}{2} e^2 = E(v) \quad (5)$$

$\nabla E(v)$ represents the partial derivation of cost function E for each element of weight vector v,

$$\nabla E(v) = \frac{\partial E}{\partial u} \frac{\partial u}{\partial v} \quad (6)$$

where, $\frac{\partial E}{\partial u}$ represents the error signal, which is used to measure the change degree of error when the input of u changes, and $\frac{\partial u}{\partial v}$ is used to measure the impact on weight vector v when the specific input u is calculated.

By applying the chain rule to Eq. (6), we have:

$$\nabla E(v) = \frac{\partial E}{\partial e} \frac{\partial e}{\partial o} \frac{\partial o}{\partial u} \frac{\partial u}{\partial v} \quad (7)$$

Differentiate e on both sides of Eq. (5), and then:

$$\frac{\partial E}{\partial e} = e \quad (8)$$

At the same time, differentiate o on both sides of Eq. (4), and then:

$$\frac{\partial E}{\partial e} = -1 \quad (9)$$

Differentiate u on both sides of Eq. (3), and then:

$$\frac{\partial o}{\partial u} = f'(u) \quad (10)$$

Finally, differentiate v on both sides of Eq. (2), and then

$$\frac{\partial u}{\partial v} = x \quad (11)$$

Substitute the above equations into Eq. (7), and the first partial derivative of the cost function e to weight vector v can be expressed as:

$$\nabla E(v) = -e f'(u) x \quad (12)$$

Therefore, LMS learning rules can be written as follows:

$$\begin{aligned} v_{q+1} = v_q + \Delta v_q &= v_q - \delta \nabla E(v_q) \\ &= v_q + \delta e_q f'(u_q) x_q \end{aligned} \quad (13)$$

Since LMS is applied to the BP algorithm, it is necessary to specify error signal term ϑ for the output layer in the formula of the LMS algorithm. The error signal is given by the following formula:

$$d = e f'(u) = (d - o) f'(u) \quad (14)$$

Then Eq. (13) can be redefined as follows:

$$v_{q+1} = v_q + \Delta v_q = v_q + \delta \vartheta_q x_q \quad (15)$$

Therefore, the LMS learning algorithm of linear neuron is given by the following formula:

$$\begin{aligned} v_{q+1} = v_q + \Delta v_q &= v_q + \delta (d_q - o_q) x_q \\ &= v_q + \delta e_q x_q \end{aligned} \quad (16)$$

4.2 Improved batch learning BP neural network

The BP neural network designed and improved in this paper consists of two stages: feed-forward stage and back propagation stage. The input samples are transferred from the input layer to the output layer after being processed layer by layer in the hidden layer. When the actual output value of the output layer is inconsistent with the expected result, it is corrected by back propagation of error, which is to propagate the output error through the hidden layer to the input layer, and then distribute the error to all the neurons passing through each layer to obtain the error signal of each layer. Then the error signal is used to correct the weight of each neuron. When the

output error is within the controllable range, that is, the error is less than a certain set threshold, the loop will end.

This paper introduces the momentum scalar factor to define the weight. At the same time, it also uses the batch learning technology to make the weight more accurate. The specific mathematical expression is defined as follows:

The training data X with P samples is given by the following formula:

$$X = \{x_q, d_q\}, \qquad q = 1,2,\dots,P \tag{17}$$

where, x_q represents the input vector of sample q with n features and d_q is the vector of the expected output.

In the feed-forward stage, for the I neurons in the hidden layer, the (P, I) dimensional input matrix u is calculated as follows:

$$u = XV \tag{18}$$

where, X is the $(P, n+1)$-dimensional matrix of the training dataset, and V is the $(I, n+1)$-dimensional weight vector matrix in the hidden layer. In the algorithm, the hyperbolic tangent function and the logistic sigmoidal function are used as the activation function. The output matrix of the hyperbolic tangent activation function in the hidden layer is as follows:

$$z = 2/(1+\exp(-u)) - 1 \tag{19}$$

Therefore, the derivative of the error signal in the hidden layer is:

$$z' = \frac{1}{2}(1-z^2) \tag{20}$$

For the sigmoidal activation function, the equivalent equation is as follows:

$$z = 1/(1+\exp(-u)) \tag{21}$$

The derivative of the error signal in the hidden layer is:

$$z' = z(1-z) \tag{22}$$

If the output layer neuron is linear, calculate the output layer weight vector W with $(I+1, K)$ dimensions directly by using the pseudo inverse algorithm.

$$W = z_b^* d \tag{23}$$

where, z_b^* is the pseudo inverse of z.

Then calculate the input matrix u_0 of the output layer of neural network as follows:

$u_0 = z_b W$.

Because the neurons in the output layer are linear, the output of the neurons in the output layer is u_0.

This paper designs and improves the batch learning BP algorithm by using the *k*-fold cross validation cycle, which means that the data set is randomly divided into average *k* parts, of which the *k*-1 part is the training set and the rest part is the test set. The process is repeated *k* times. Because every experiment has a certain error rate, the average of the *k*-times error rates can be used to measure the accuracy of the algorithm. Therefore, to optimize and improve the neural network model, it is necessary to find the three best variable parameters of the neural network - the first is the number of hidden neurons, the second is the best learning rate, and the last variable parameter is the number of iterations of the k-fold cross validation cycle.

The process of the improved batch learning BP neural network is defined as follows:

Input: Training sample set D and the set of class labels of tuples in D, vector I_0 of the number of neurons in the hidden layer, learning rate vector β_0, learning steps iteration number vector S_0 and threshold T.

Output: Classification model of corresponding samples.

Step 1: Select I neurons from vector I_0;

Step 2: The hidden layer weight vector V and the output layer weight vector W are initialized to an $(n+1, I)$ -dimensional matrix and a $(J+1, K)$ -dimensional matrix, respectively. The fixed parameter kw is used to initialize the hidden layer weight vector V in the range of $(-kw, +kw)$.

Step 3: err is the scalar of the model error calculated with a specific combination of variable parameters (I_0, β_0, S_0). err is reset to 0.

Step 4: Apply the 10-fold cross validation method to the preprocessed training data and substitute specific variable parameters. At the beginning of each iteration in the cross validation process, set V and W to the initial values.

Step 5: For each iteration of 10-fold cross validation, firstly, apply the batch BP algorithm by substituting the learning rate β and the number of iterations into the training dataset, and calculate the V and W weights. Then, train the existing models on the test set data to evaluate V and W and calculate the error value of err. Finally, calculate the cumulative value of err obtained through ten cycles.

Step 6: Calculate the cumulative error percentage of all training data, and then save the error percentage under the specific variable parameter in the three-dimensional array i_Err.

Step 7: Go to Step 2, repeat the above steps for all I_0, β_0 and S_0, and save their error percentages in the array i_Err.

Step 8: Find the minimum error percentage of the i_Err array, and then extract its related variable parameter (I_0 , β_0, S_0).

Step 9: Establish the classification model by substituting the best variable parameters obtained in Step 8, and train the training set data to determine the weight vector V and W.

Step 10: Use the classifier model to train all the training data of a specific test set, and calculate the error percentage to measure the accuracy of the classification model.

5. EXPERIMENT AND ANALYSIS

The sample data used in the experiment is from the UCI machine learning library. In the experiment, two methods are used for comparative training. The original data of a set of experiments are directly preprocessed without being normalized to the same order of magnitude. In another set of experiments, the original data are normalized to the same order of magnitude, and then pre-processed. The experimental results of two sets of experiments are observed. The neural network consists of 9 input layer nodes, 5 hidden layer nodes and 5 output layer nodes. The total sample size is 17.

The errors between the predicted results obtained by the two methods and the real values are compared, as shown in Figure 3.

As shown in Figure 3, the experimental results show that the

errors are large between the predicted values and the real values of the experimental group where data are not normalized to the same order of magnitude. However, for the experimental group where data are normalized to the same order of magnitude, the errors are small between the predicted values and the real values obtained after learning by the BP neural network algorithm. Therefore, the classification algorithm has higher accuracy and better learning performance thanks to the improved data pre-processing method.

In order to verify the effectiveness of the improved batch learning BP neural network proposed in this paper, some data sets in the UCI machine learning database are used in the experiment, whose descriptive information is listed in Table 1.

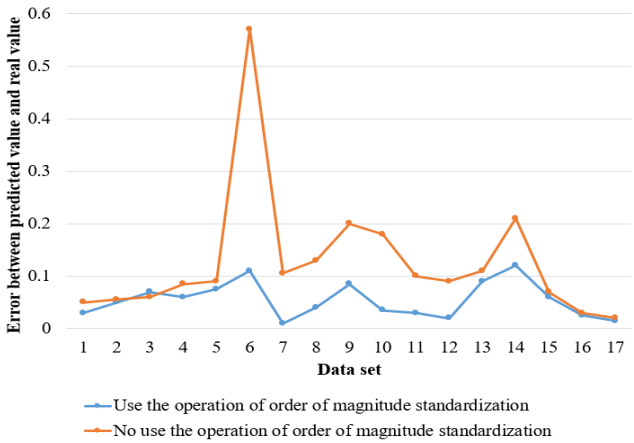

Figure 3. Comparison of errors between the predicted results obtained by the two methods and the real values methods

Table 1. Descriptive information of the experimental data set

Data set	Data scale	Number of attributes	Number of classes
Wine	238	7	5
Head	177	10	4
Animal	223	8	6
Letter	375	32	7
Arm	152	7	4
Plane	198	12	6
Bean	230	17	2
Face	549	8	6
Car	452	9	7
Craw	986	22	5
Vote	2269	15	8

In the experiment, 11 sample data sets are used, of which ten are used as the training set and the other one as the test set. The average error percentage is obtained and compared with the results of the traditional BP neural network.

The values of the fixed parameters used in the experiment are as follows: the momentum factor ϑ_m=0.75, and the initial weight vector in the hidden layer kw=0.1. Variable parameters include the number of neurons in the hidden layer I, the learning rate β and the number of iterations in the training stage.

By using these data sets, the improved batch learning BP algorithm and the traditional BP algorithm are trained and studied respectively, and the accuracy of the experimental results are shown in Table 2.

As can be seen from Table 2, the improved batch learning BP algorithm proposed in this paper has higher classification accuracy than the traditional BP algorithm for most data sets.

Table 2. Classification accuracy of the improved batch learning BP algorithm and the traditional BP algorithm

Data set	Accuracy of the traditional BP algorithm	Accuracy of the improved batch learning BP algorithm
Wine	97.38%	99.01%
Head	96.68%	98.17%
Animal	88.93%	94.81%
Letter	88.02%	91.72%
Arm	100%	100%
Plane	98.38%	98.76%
Bean	92.73%	95.21%
Face	89.32%	90.25%
Car	91.29%	92.33%
Craw	98.57%	98.27%
Vote	97.89%	97.81%

6. CONCLUSIONS

This paper presents an improved method of data pre-processing. Before pre-processing, the data are normalized so that the values of the data are basically on the same order of magnitude, which facilitates pre-processing, and directly improves the effect and accuracy of the subsequent data mining process. And then, with the least mean square algorithm and the BP neural network classification algorithm are taken as the basic algorithms, this paper designs and implements an improved batch learning BP algorithm using the rules of batch learning and introducing the momentum factor. Finally, the effectiveness of the improved method is verified through experimental data analysis.

REFERENCES

[1] Senousy, Y., Hanna, W.K., Shehab, A., Riad, A.M., El-Bakry, H.M., Elkhamisy, N. (2019). Egyptian social insurance big data mining using supervised learning algorithms. Revue d'Intelligence Artificielle, 33(5): 349-357. https://doi.org/10.18280/ria.330504

[2] Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. (1996). From data mining to knowledge discovery in databases. AI Magazine, 17(3): 37-54. https://doi.org/10.1609/aimag.v17i3.1230

[3] Subhash, P., Choudhary, N. (2018). Predicting instructor performance using naïve Bayes classification algorithm in data mining technique. International Journal of Computer Applications, 179(22): 9-12. https://doi.org/10.5120/ijca2018916409

[4] Kumar, R., Verma, R. (2012). Classification algorithms for data mining: A survey. International Journal of Innovations in Engineering and Technology (IJIET), 1(2): 7-14.

[5] Kanj, S., Abdallah, F., Denoeux, T., Tout, K. (2016). Editing training data for multi-label classification with the k-nearest neighbor rule. Pattern Analysis and Applications, 19(1): 145-161. https://doi.org/10.1007/s10044-015-0452-8

[6] Moezzi, S., Jalali, M., Forghani, Y. (2019). TWSVC+: Improved twin support vector machine-based clustering. Ingénierie des Systèmes d'Information, 24(5): 463-471. https://doi.org/10.18280/isi.240502

[7] Safavian, S.R., Landgrebe, D. (1991). A survey of decision tree classifier methodology. IEEE Transactions

on Systems, Man, and Cybernetics, 21(3): 660-674. https://doi.org/10.1109/21.97458
[8] Kohzadi, N., Boyd, M.S., Kermanshahi, B., Kaastra, I. (1996). A comparison of artificial neural network and time series models for forecasting commodity prices. Neurocomputing, 10(2): 169-181. https://doi.org/10.1016/0925-2312(95)00020-8
[9] Xu, Z., Watada, J., Wu, M., Ibrahim, Z., Khalid, M. (2014). Solving the imbalanced data classification problem with the particle swarm optimization based support vector machine. IEEJ Transactions on Electronics, Information and Systems, 134(6): 788-795. https://doi.org/10.1541/ieejeiss.134.788
[10] Karaçali, B., Krim, H. (2003). Fast minimization of structural risk by nearest neighbor rule. IEEE Transactions on Neural Networks, 14(1): 127-137. https://doi.org/10.1109/TNN.2002.804315
[11] Nansen, C., Geremias, L. D., Xue, Y., Huang, F., Parra, J.R. (2013). Agricultural case studies of classification accuracy, spectral resolution, and model over-fitting. Applied Spectroscopy, 67(11): 1332-1338. https://doi.org/10.1366/12-06933
[12] Jin, D.B., Xu, S.Q., Tong, L.J., Wu, L.Y., Liu, S.M. (2020). A deep learning model for striae identification in end images of float glass. Traitement du Signal, 37(1): 85-93. https://doi.org/10.18280/ts.370111
[13] Liu, R., Gillies, D.F. (2016). Overfitting in linear feature extraction for classification of high-dimensional image data. Pattern Recognition, 53: 73-86. https://doi.org/10.1016/j.patcog.2015.11.015
[14] Li, D.Z., Wang, W., Ismail, F. (2015). A selective boosting technique for pattern classification. Neurocomputing, 156: 186-192. https://doi.org/10.1016/j.neucom.2014.12.063
[15] Brownfield, B., Lemos, T., Kalivas, J.H. (2018). Consensus classification using non-optimized classifiers. Analytical chemistry, 90(7): 4429-4437. https://doi.org/10.1021/acs.analchem.7b04399
[16] Palanisamy, T., Sadayan, G., Pathinetampadiyan, N. (2019). Neural network–based leaf classification using machine learning. Concurrency and Computation: Practice and Experience, e5366. https://doi.org/10.1002/cpe.5366
[17] Gutierrez-Osuna, R., Nagle, H.T. (1999). A method for evaluating data-preprocessing techniques for odour classification with an array of gas sensors. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 29(5): 626-632. https://doi.org/10.1109/3477.790446
[18] Torii, A.J., De Faria, J.R. (2017). Structural optimization considering smallest magnitude eigenvalues: a smooth approximation. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 39(5): 1745-1754. https://doi.org/10.1007/s40430-016-0583-x
[19] Costea, A., Nastac, I. (2005). Assessing the predictive performance of artifIcial neural network-based classifiers based on different data preprocessing methods, distributions and training mechanisms. Intelligent Systems in Accounting, Finance & Management: International Journal, 13(4): 217-250. https://doi.org/10.1002/isaf.269
[20] Kou, Z.C., Fang, Y.J. (2019). A semi-supervised sparse representation neural network for error estimation of electricity meters with insufficient tagged samples. Instrumentation Mesure Métrologie, 18(6): 591-594. https://doi.org/10.18280/i2m.180611