Journal of Applied Sciences

ISSN 1812-5654

Journal of Applied Sciences 13 (14): 2658-2663, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.2658.2663

Research of Improved LVQ Neural Network by AdaBoost Algorithm

[1]Li Xiang, [1]Zhu Quanyin and [2]Wang Zun
[1]Faculty of Computer Engineering, Huaiyin Institute of Technology, Huai'an 223003, China
[2]School of Electronic Engineering and Optoelectronic Technology,
Nanjing University of Science and Technology, Nanjing 210094, China

Abstract: In view that the traditional Learning Vector Quantization (LVQ) neural network is sensitive to initial value and has a lower stability of the algorithm, a new LVQ model with the AdaBoost algorithm was put forward to improve the forecasting accuracy and generalization ability. Firstly, the method performed the pre-treatment for the historical data and initialized the distribution weights of test data. Secondly, it selected different hidden layer nodes and network learning functions to construct weak predictors of LVQ neural network and trained the sample data repeatedly. At last, it made more weak predictors of LVQ neural network to form a new strong predictor by AdaBoost algorithm for classification. A simulation experiment for the 6 data sets of UCI was carried out. The results show that this method has improved the classification accuracy nearly 5% compared to the traditional LVQ neural network and has a better stability of the algorithm. This method provides references for the LVQ neural network.

Key words: LVQ, classification, iterative algorithm, adaboost algorithm

INTRODUCTION

LVQ neural network is a supervised forward neural network with simple structure and powerful function, of which the algorithm was evolved from the Kohonen (2001) competitive algorithm and is one of the most widely used neural network models at present. In the training process, LVQ neural network realizes the gradual convergence of the boundary between different categories of weight vector to the Bayes classification border through the continuous renewal of neuron weight vector and the continuous adjustment of learning rate. LVQ neural network determines the winning neuron by calculating the Euclidean distance between the input sample and the weight vector (An *et al*., 2011). Compared to the traditional BP neural network and ART neural network, LVQ neural network has a simpler network structure, a faster learning rate, a more reliable recognition rate and a better fault tolerance in the application of structure type selection (Luo *et al*., 2010). But LVQ also has some deficiencies: It is more sensitive to the initial value of the network in the process of training; The algorithm performance is unstable; For the input sample, each dimension attribute's information was used insufficiently and did not reflect that each dimension attribute had a different importance in the classification process (Qiao *et al*., 2012; Filippi and Jensen, 2007).

Many improvement researches have been proposed at home and abroad, aimed at the performance and algorithm of LVQ. Literatures (Zhu *et al*., 1995) have deeply researched on LVQ1 and LVQ2, strictly analyzing and deriving the learning step length of LVQ1 and the algorithm essence of LVQ2, respectively. Biehl *et al*. (2007) introduced the concept of the related factors into the Euclidean distance formula and put forward the algorithm of the Related Learning Vector Quantization (RLVQ). Eventually, this method can characterize the importance degree of each dimension attribute in the weight vector, thus improving the performance of the traditional LVQ algorithm but due to the limitation of learning rule in the algorithm, it needs normalization operation for the related factors after updating which makes the algorithm have a higher computational complexity and an unstable system. Hammer and Villmann, 2002.) and other people proposed the Generalized Learning Vector Quantization Algorithm (GRLVQ), this algorithm can ensure the convergence of the weight vector and has a higher recognition rate but it has a poorer generalization ability and stability.

For the above problems, the paper put forward to combine the LVQ neural network with the AdaBoost algorithm to form the strong predictor model of AdaBoost_LVQ and did the simulation experiment for the data set in the UCI data base to prove the validity of this method.

Corresponding Author: Li Xiang, Faculty of Computer Engineering, Huaiyin Institute of Technology, Huai'an, 223003, China

THE LVQ NEURAL NETWORK AND ADABOOST ALGORITHM

The principles of LVQ neural network: LVQ neural network is a supervised learning neural network which is used for training the competitive layer, generally composed of three layers of neurons that are input layer, hidden layer and linear output layer (Chang and Zhuang, 2007). The input layer and hidden layer are fully connected, the hidden layer and linear output layer are partially connected, the number of hidden layer neurons is always greater than that of linear output layer, each hidden layer neuron is only connected with a linear output layer neuron and the connection weights keep at 1 but each linear output layer neuron can be connected with multiple hidden layer neurons. The value of hidden layer neurons and linear output layer neurons can only be 1 or 0, when an input pattern is sent to the network, the hidden layer neuron closest to the input pattern is activated, with the status of "1" and all states of other hidden layer neurons are "0". Meanwhile, the status of linear output layer neuron connected with the activated neuron is "1" and all states of other linear out layer neurons are "0" (Jiang and Liu, 2011; Xu *et al*., 2012).

The structure of LVQ neural network is shown in Fig. 1, wherein p is the input pattern of R dimension, S^1 is the number of hidden layer neurons, $IW^{1,1}$ is the connection weight coefficient matrix between the input layer and hidden layer, n^1 is the input of competitive layer neurons, a^1 is the output of competitive lay neurons, $LW^{2,1}$ is the connection weight coefficient matrix between the hidden layer and linear output layer, n^2 is the input of linear output layer neurons and a^2 is the output of linear output layer neurons (Wang *et al*., 2008).

AdaBoost algorithm: AdaBoost algorithm can boost a group of weak predictors adaptively to a strong predictor and introduce a weight ω_i for each training sample, to realize training through the iterative process. Every time when training a weak predictor iteratively, it should have the minimum error rate under the current weight distribution. After each end of iteration, the weight of prediction error sample should be increased and the weight of prediction correct sample should be reduced, so that the wrong sample will be paid more attention to in the next selection of the iteration weak predictor (Li and Zhu, 2012). The advantage of AdaBoost algorithm is that it uses the selected training data after weighting instead of the randomly selected training samples, to combine the weak predictors and uses the weighted voting mechanism instead of the average voting mechanism (Pan *et al*., 2011).

The adaBoost algorithm improving the classification model of LVQ neural network: The article constructs many types of LVQ weak predictors by selecting different hidden layer nodes and network learning functions for the LVQ neural network, then constitutes more weak predictors to a new strong predictor by using AdaBoost algorithm.

The hidden layer nodes of LVQ neural network have a great influence on the prediction accuracy of network: Too few nodes will lead to a bad study effect of network, the training number needs to be increased and the training accuracy will be affected; Too many nodes will cause the training time to increase and the network is easy to have over fitting (Triguero *et al*., 2012). The best hidden layer nodes proposed in the literature (Shi *et al*., 2010) can refer to the Eq. 1, 2 and 3, wherein n is the nodes of input layer, l is the nodes of hidden layer, m is the nodes of output layer and a is the constants between 0~10:

$$l < n-1 \quad (1)$$

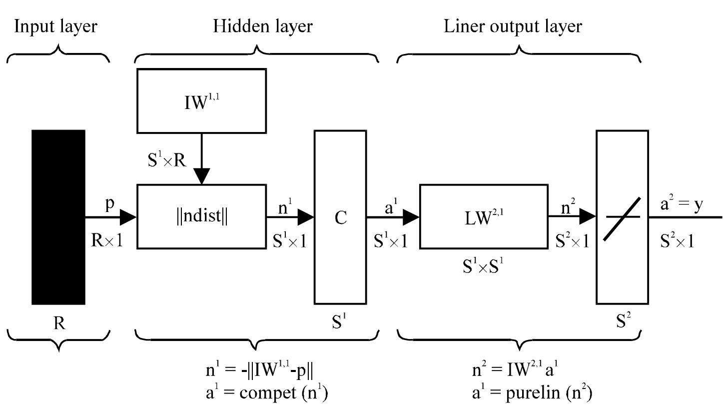

Fig. 1: Model of LVQ neural network

$$1 < \sqrt{m+n} + a \tag{2}$$

$$1 = \log_2 n \tag{3}$$

The algorithm of LVQ neural network can be divided into LVQ1 algorithm and LVQ2 algorithm. The basic idea of LVQ1 algorithm is to calculate the hidden layer neurons which are closest to the input vector, thus to find the linear output layer neurons that are connected to them. If the category of input vector is consistent with that of the linear output layer neurons, the corresponding weights of hidden layer neurons will move along the direction of the input vector but on the contrary, if the two categories are inconsistent, the corresponding weights of hidden layer neurons will move along the negative direction of the input vector (Gauri, 2010). In LVQ1 algorithm, only one neuron can win that is to say only the weights of one neuron can be updated. The LVQ2 algorithm is approaching the limit of Bayes based on the smooth move decision boundary, the "second winning" neuron is introduced, the weight vectors of "winning neuron" and "second winning neuron" are updated.

The classification algorithm flow of LVQ neural network based on the AdaBoost algorithm is shown in Fig. 2.

The detailed steps of the algorithm are explained as follows:

Step 1: Selection of sample data and initialization of network. Select m set of training data from the sample set randomly, the distribution weights of the initial test data is:

$$D_t(i) = \frac{1}{m}$$

design the network structure according to the dimension of the data input and output and set the initialization for the weights and threshold of LVQ

Step 2: Pretreatments of sample data

Step 3: Determine the number of LVQ neural network weak predictor, n

Step 4: Construct n LVQ neural network weak predictors by choosing different number of hidden layer neurons and learning functions

Step 5: Prediction of LVQ weak predictors. When training the t weak predictors, use LVQ to train the training data, to get the prediction error of the prediction sequence f_t and e_t, the error and formula are as shown in Eq. 4:

$$e_t = \sum_t D_i(i) \qquad i = 1, 2, \cdots, m \tag{4}$$

Step 6: Calculate the weights of the prediction sequence. Calculate the weights a_t of the sequence according to the prediction error e_t of the prediction sequence f_t, the formula of the weights is as shown in Eq. 5:

$$a_t = \frac{1}{2} \ln\left(\frac{1-e_t}{e_t}\right) \tag{5}$$

Step 7: Adjust the weights of the test data. Adjust the weights of new training samples according to the weights a_t of the prediction sequence and the adjusting formula is as shown in Eq. 6. Where, B_t as a normalized factor mainly makes the sum of distribution weights to be 1 when the weight proportion doesn't change:

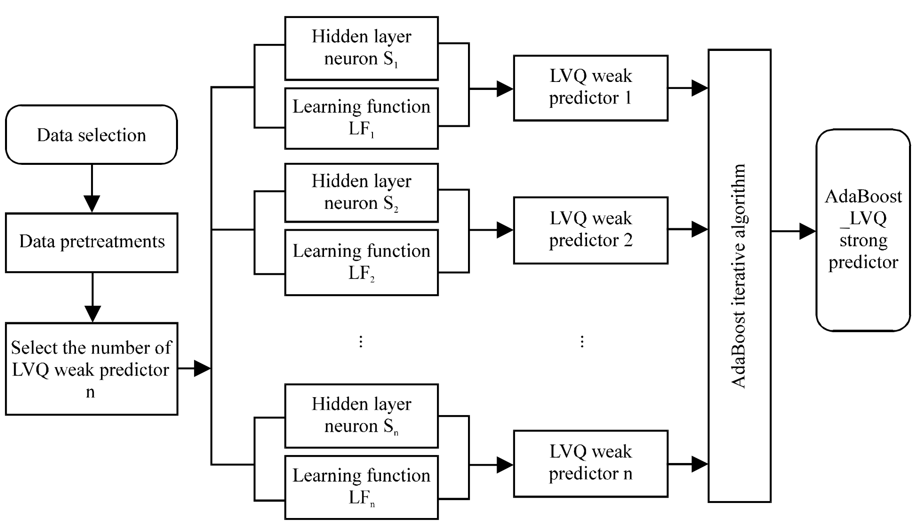

Fig. 2: Algorithm flow of LVQ neural network based on AdaBoost

$$D_{t+1}(i) = \frac{D_t(i)}{B_t} \times \exp\left[-a_t y_i g_t(x_i)\right] \qquad i = 1, 2, \cdots, m \qquad (6)$$

Step 8: Output strong predictor function. After T times of iterative algorithm, get T set of weak predictor function $f(g_t, a_t)$ and thus get strong predictor function G(x) by combining T set of weak predictors, as shown in Eq. 7:

$$G(x) = \frac{a_t}{\sum_{t=1}^{T} a_t} f(x) \qquad (7)$$

EXPERIMENT AND RESULT ANALYSIS

Experimental data: UCI database is a famous database that is provided by University of California, Irvine for machine learning, at present this database has 239 data sets in total and the number is still increasing. This experiment selected 6 data sets such as Wine and Abalone etc., from the UCI database for classification experiments, covering the situation of small samples or big samples, the situation of more or less category number, the situation of more or less feature dimension and the situation of even or uneven sample distribution, etc. The download address of data sets is http://archive.ics.uci.edu/ml/datasets/. In this experiment, the numbers of training samples and testing samples randomly selected from each data set are shown in Table 1.

In the experiment process, Wine data set was used as a detailed example and other data sets were experimented referring to the method of Wine data set. Wine data set is the common test data set for testing the accuracy rate of pattern classification, it has recorded the chemical compositions of three different wines from the same area of Italian and had 178 samples, each of which has 1 category label and 13 characteristic components that are the Class and Alcohol, MalicAcid, Ash, AlcalinityofAsh, Magnesium, TotalPhenols, Flavanoids, NonflavanoidPhenols, Proanthocyanins, ColorIntensity, Hue, OD280/OD315ofDilutedWines, Proline.

Experiment and result analysis: MATLAB (2013a) software is used for the simulation experiment, training samples and testing samples listed in Table 1 are taken randomly from the data sets. The constructor function of LVQ neural network in Matlab is:

net = newlvq (PR, S1, PC, LR, LF)

Wherein, PR is a R2 dimension input matrix, it determines the minimum and maximum value range of the input vector and R is the number of input vector; S1 is the number of hidden layer neurons; PC is the respective proportion of linear output layer's expectation category; LR is the learning rate and the default value is 0.01; LF is the learning function; The output parameter net is the generated LVQ neural network (Boubaker *et al*., 2010; Biehl, 2012).

The corresponding weight learning function of LVQ1 algorithm in Matlab is learnlv1 and its call function is:

[dW, LS] = learnlv1
(W, P, Z, N, A, T, E, gW, gA, D, LP, LS)

Wherein, dW is the weight change matrix; LS is the current learning state; W is the weight matrix; P is the input vector; Z is the weight vector of input layer; N is the input vector of network; A is the output vector of network; T is the target output vector; E is the error vector; gW is the weight gradient matrix that is related to the performance; gA is the output gradient matrix that is related to the performance; D is the distance matrix of neurons; LP is learning parameter and the default value is 0.01; LS is the initial learning state (Guo *et al*., 2012; Schneider *et al*., 2010).

The corresponding weight learning function of LVQ2 algorithm in Matlab is learnlv2 and its call function is:

[dW, LS] = learnlv2
(W, P, Z, N, A, T, E, gW, gA, D, LP, LS)

In addition to the different adjustment method of the weights, the meaning of parameter is the same as the learnlv1.

Different LVQ neural networks are constructed through the number of hidden layer neurons S1 of newlvq function and different values of learning functions LF. In

Table 1: Numbers of training samples and testing samples

Data sets	No. of categories	No. of characteristic dimension	Training samples	No. of test samples
Wine	3	13	100	78
Abalone	3	8	2500	1677
Image segmentation	7	19	1500	810
Breast cancer wisconsin	2	32	400	169
Adult	2	14	40000	8842
Dermatology	6	33	200	166

Table 2: Parameter selection of LVQ weak predictor

No.	S1	LF
1	1<n-1	learnlv1
2	$1<\sqrt{(m+n)}+a$	learnlv2
3	$1=\log_2 n$	learnlv1
4	l<n-1	learnlv2
5	$1<\sqrt{(m+n)}+a$	learnlv1
6	$1=\log_2 n$	learnlv2

Table 3: Classification accuracy rate and average classification accuracy rate of the wine data set in 5 experiments

NO.	AdaBoost_LVQ	BP	RBF	LVQ
1	98.72% (77/78)	94.87% (74/78)	96.15% (75/78)	94.87% (74/78)
2	94.87% (74/78)	97.44% (76/78)	94.87% (74/78)	92.31% (72/78)
3	97.44% (76/78)	92.31% (72/78)	93.59% (73/78)	93.59% (73/78)
4	98.72% (77/78)	87.18% (68/78)	92.31% (72/78)	94.87% (74/78)
5	97.44% (76/78)	94.87% (74/78)	94.87% (74/78)	94.87% (74/78)
$\bar{x}$	97.44%	93.33%	94.36%	94.10%

Table 4: Average classification accuracy rates of other 5 UCI data sets

Data sets	AdaBoost_LVQ (%)	BP (%)	RBF (%)	LVQ (%)
Abalone	97.28	90.17	94.16	92.86
Image segmentation	95.33	87.90	88.15	91.65
Breast cancer wisconsin (diagnostic)	94.08	87.81	90.18	87.81
Adult	94.35	88.24	92.66	90.17
Dermatology	92.17	89.76	95.78	87.23

this experiment, 6 LVQ neural networks are selected to constitute a weak predictor, the selections of S1 reference formula and LF function of these 6 LVQ neural networks are shown in Table 2.

The sample is randomly extracted in each experiment, so each data set is experimented five times to let the experimental results have a higher representativeness and then the average value of 5 classification accuracy rates is obtained as the final accuracy rate, the calculation formula of average classification accuracy rate is shown in Eq. 8.

Average classification accuracy rate $\bar{x}$:

$$\bar{x}=\frac{1}{n}\sum_{i=1}^{n}x_i \quad n=5 \tag{8}$$

In order to compare the classification effect of AdaBoost_LVQ, three kinds of neural networks are introduced into this experiment for comparison, such as BP, RBF and LVQ. Table 3 shows the detailed classification accuracy rate and average classification accuracy rate of the Wine data set in 5 experiments.

From Table 3, it can be seen the accuracy rates of AdaBoost_LVQ in 5 classifications of the Wine data set, respectively 98.72, 94.87, 97.44, 98.72 and 97.44% and the average classification accuracy rate reaches 97.44%. This shows in the Wine data set, the strong classifier model after improving LVQ based on AdaBoost algorithm has a higher forecasting accuracy and a better generalization ability.

The average classification accuracy rates of each neural network corresponding with other 5 UCI data sets are shown in Table 4.

It can be seen from Table 3 and 4, the average classification accuracy rates of AdaBoost_LVQ model are the highest in five data sets such as Wine, Abalone, Image Segmentation, Breast Cancer Wisconsin (Diagnostic) and Adult, the average classification accuracy rate in Dermatology data set is second only to RBF neural network. Thus it can be seen, the LVQ neural network model that is improved by AdaBoost has a higher classification accuracy rate, a better generalization ability and a stronger robustness which is a feasible method to improve the LVQ neural network.

CONCLUSION

LVQ neural network has been widely used in pattern classification and achieved a better effect but it is still sensitive to initial value and other problems. This article constituted a weak predictor sequence by selecting different hidden layer nodes and network learning functions for the traditional LVQ neural network, then constructed a new strong predictor by combining with AdaBoost algorithm, this method has effectively improved the prediction accuracy and stability of the LVQ neural network and provided a reference for the application of LVQ neural network.

ACKNOWLENDGMENT

This study was supported by the National Sparking Plan Project of China (2011GA690190) and the fund of Huaian Industry Science and Technology, China (HAG2012086).

REFERENCES

An, X., Z.W. Liu and Y.G. Shi, 2011. A weighted learning vector ouantization algorithm. Pattern Recognit. Artif. Intell., 24: 376-384.

Biehl, M., 2012. Admire LVQ-adaptive distance measures in relevance learning vector quantization. Kunstliche Intelligenz, 26: 391-395.

Biehl, M., A. Ghosh, B. Hammer and Y. Bengio, 2007. Dynamics and generalization ability of LVQ algorithms. J. Mach. Learn. Res., 8: 323-360.

Boubaker, M., M. Akil, K.B. Khalifa, T. Grandpierre and M.H. Bedoui, 2010. Implementation of an LVQ neural network with a variable size: Algorithmic specification, architectural exploration and optimized implementation on FPGA devices. Neural Comput. Appl., 19: 283-297.

Chang, C.Y. and D.F. Zhuang, 2007. A fuzzy-based learning vector quantization neural network for recurrent nasal papilloma detection. IEEE Trans. Circuits Syst., 54: 2619-2626.

Filippi, A.M. and J.R. Jensen, 2007. Effect of continuum removal on hyperspectral coastal vegetation classification using a fuzzy learning vector quantizer. IEEE Trans. Geosci. Remote Sensing, 45: 1857-1869.

Gauri, S.K., 2010. Control chart pattern recognition using feature-based learning vector quantization. Int. J. Adv. Manuf. Technol., 48: 1061-1073.

Guo, P., Z. Jiang, S. Lin and Y. Yao, 2012. Combining LVQ with SVM technique for image semantic annotation. Neural Comput. Appl., 21: 735-746.

Hammer, B. and T. Villmann, 2002. Generalized relevance learning vector quantization. Neural Networks, 15: 1059-1068.

Jiang, D. and C. Liu, 2011. Machine condition classification using deterioration feature extraction and anomaly determination. IEEE Trans. Reliab., 60: 41-48.

Kohonen, T., 2001. Self-Organizing Maps. 3rd Edn., Springer-Verlag, Berlin, Germany, ISBN: 9783540679219, Pages: 501.

Li, X. and Q.Y. Zhu, 2012. Tax forecasting based on Adaboost algorithm and BP neural network. J. Comput. Appl., 32: 3558-3560.

Luo, Z.Z., J. Xiong and Z.H. Liu, 2010. Pattern recognition of hand motions based on WPT and LVQ. Pattern Recognit. Artif. Intell., 23: 695-700.

Pan, H., B. Chen and Q.W. Li, 2011. Paper currency number recognition based on binary tree and adaboost algorithm. J. Comput. Appl., 31: 396-398.

Qiao, H., Y.Z. Zhou and S. Nan, 2012. Software reliability prediction based on learning vector quantization neutral network. J. Comput. Appl., 32: 1436-1438.

Schneider, P., K. Bunte, H. Stiekema, B. Hammer, T. Villmann and M. Biehl, 2010. Regularization in matrix relevance learning. IEEE Trans. Neural Networks, 21: 831-840.

Shi, F., X.C. Wang and L. Yu, 2010. Thirty Cases Analysis of MATLAB Neural Network. Beijing University Press, Beijing, China.

Triguero, I., J. Derrac, S. Garcia and F. Herrera, 2012. A taxonomy and experimental study on prototype generation for nearest neighbor classification. IEEE Trans. Syst. Man Cybernetics C, 42: 86-100.

Wang, C.Y., Z.T. Fu and J. Zhang, 2008. On automatic classification for agricultural products web pages in Chinese based on LVQ network and ontology. J. Comput. Inform. Syst., 4: 971-976.

Xu, Y., F. Shen and J. Zhao, 2012. An incremental learning vector quantization algorithm for pattern classification. Neural Comput. Appl., 21: 1205-1215.

Zhu, C., L.H. Li and T. Wang, 1995. The performance analysis of learning vector quantization algorithm. Acta Electronica Sinica, 23: 59-63.