Journal of Information & Computational Science 12:17 (2015) 6553–6562 November 20, 2015
Available at http://www.joics.com

Patten Classification Based on a Hybrid Neural Network ⋆

Jiandun Li*

School of Electronic Information, Shanghai Dianji University, Shanghai 201306, China

Abstract

Artificial Neural Network (ANN) has been applied to pattern classification for years, which is mostly due to its nonlinear capability. However, most of the approaches using ANN, supervised or unsupervised, converge slowly. In this paper, a pattern classification method based on ANN, named HBG, is introduced, in which a hybrid learning strategy comprising back propagation and Gaussian weights, is presented. The benchmark test shows that, holding the accuracy, the HBG method converges faster, compared with several ANN-based approaches.

Keywords: Pattern Classification; Artificial Neural Network (ANN); Back Propagation (BP)

1 Introduction

The goal of pattern classification is to partition the feature space into decision regions, mostly one region for each class [1]. And Artificial Neural Network (ANN) emerges to be one of the most applied technologies ever since its birth in 1940s. However, there are still some drawbacks which have been proved to be obstacles to overpass for most ANN-based classifiers. Take associative memory, Back Propagation (BP) and Gaussian weight for instance. The memory volume is very limited to avoid interference and degeneration of the stored information as referred to associative memory; as to BP, the number of hidden layers and the number of neurons within hidden layers are difficult to determine, and the situation that falling into local minimums is often encountered [2]; as to Gaussian weight, which is a family member of Adaptive Resonance Theory (ART) and field theory, the convergence process is quite time-consuming, sometimes intolerable, although it has been mathematically proved to be capable of stably learning a recognition code in response to an arbitrary sequence of binary input patterns until it utilizes its full memory capacity. The cause mostly lies in the adjustment of the attracting basin which is a resonant process [3]. Compared with Zhihua Zhou et al. [3], our contribution lies in the converging part, i.e., focuses more on how to shorten the initiating pain process when learning. Therefore, a hybrid ANN model, named

⋆This work is supported by Shanghai Leading Academic Discipline Project (No. J50103), Shanghai Funding Program for Young Teacher Training of Colleges (No. 14AZ23), Shanghai Dianji University Key Academic Program (No. 13XKJ01), and Shanghai Dianji University Scientific Research Starting Funds (No. 13QD01).

*Corresponding author.
Email address: lijd@sdju.edu.cn (Jiandun Li).

DOI: 10.12733/jics20106986

HBG, is proposed in this article, which integrates Gaussians searching capability across the entire problem space and BPs weight sensitive as well as fast adjusting gift.

This paper follows with five sections. In Section 2, some background knowledge is given; in Section 3, details about the HBG method is presented; benchmark tests are presented in Section 4; related work is reviewed in Section 5; finally, Section 6 concludes the paper.

2 Background Knowledge

- **Gaussian weight** Gaussian weight was introduced by Zhihua Zhou et al. in FANNC model [3], which organically exploits the advantages of both ART and field theory. As to the structure of FANNC, besides one input layer and one output layer, there are two hidden layers, which represent the combination of input attributes and output classes respectively. Although Hopfield has proved that one single hidden layer is enough for almost every non-linear function, one more hidden layer does loosen the connection between neurons. Moreover, the structure of this model is adaptively adjusted by adding new neurons to the model. Gaussian weights were applied between the input layer and the first hidden layer, where neurons from each layer are fully connected. One neuron in the first hidden layer and its connections to the input layer represent a particular attracting basin, represented by responsive center and responsive width, which is designed to cover special patterns.

- **Back propagation strategy** Back Propagation (BP) is an important learning strategy in learning history of ANNs. It is extensively applied in feed-forward ANNs, such as Multilayer Perceptron (MLP) models [4]. It consists of two phases: 1) the forward phase, where the activations are propagated from the input layer to the output layer, and 2) the backward phase, where the error between the observed actual and the requested nominal value are propagated backwards so as to modify the weights and bias values.

3 The HBG Method

3.1 Structure

The structure of HBG, which has four layers, is generally inspired by ART and FANNC. The first layer (Layer-I) is the input layer, and the number of neurons is exactly equal to the number of input attributes. The second (Layer-S) and the third layer (Layer-T) are hidden layers, which represent the combination of input attributes and output classes respectively. There is only one neuron in every hidden layer when the first pattern is going to be learned, and with the learning procedure moving on, new neurons are added into the hidden layers adaptively. The last layer (Layer-O) is the output layer, the number of which is equal to the number of target classes. Layer-I and Layer-S are fully connected, and the connections are Gaussian weights, represented by responsive center and responsive width. The rest of connections are ordinary weights, with BP strategy (See Fig. 1 for the structure).

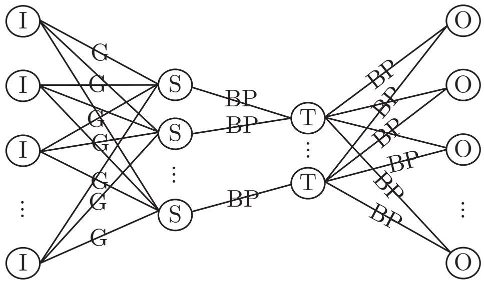

Fig. 1: Structure

3.2 Learning Strategy

At the very beginning, there are only Layer-I and Layer-O in the model. The learning procedure is comprised of two phases as BP strategy advised, forward learning and backward tuning.

3.2.1 Forward Learning

First, patterns are submitted into the system via Layer-I and the result can be received from Layer-O. The learning phase is depicted in Fig. 2.

For convenience, patterns is formalized as

$$P_q = (f_1, f_2, \cdots, f_{NOF}), \tag{1}$$

where P_q represents the q-th input pattern, f_i represents the i-th feature, and NOF stands for the number of features per input pattern.

When the first patter P_1 is about to be launched, two neurons S_1 and T_1 are added into Layer-S and Layer-T respectively, which are directly connected. The threshold value of S_1 and T_1 and their connection weight W_{11}^{ST} are randomly set from $(-1, 1)$. Every neuron S_j within Layer-S and its connections to neurons of Layer-I construct an attractive field AF_j, which is called attracting basin [3]. The value sends from I_i (the i-th neuron in Layer-I) to S_j is

$$O_{ij}^{IS} = e^{-\left(\frac{f_i - \theta_{ij}}{\alpha_{ij}}\right)^2}, \tag{2}$$

where α_{ij} and θ_{ij} are the responsive center and responsive width of AF_j.

During the learning phase, if one pattern belongs to one of the attractive field, the attracting field would adjust itself so as to reduce the gap between this pattern and the responsive center; otherwise, another attractive field which is nearer to this input pattern than others may be adjusted to enclose this pattern.

Every neuron of Layer-S sums the value sent from Layer-I, and applies the Sigmoid Function to calculate its output value. The equation is listed below.

$$O_j^S = f\Big(\sum_{i=1}^{NOF} O_{ij}^{IS} - \theta_j^S\Big), \tag{3}$$

where f is the Sigmoid Function.

When the result of each neuron in Layer-S is ready, there is a competition named leakage competition, which means neurons whose output value preponderate over a particular value L can send its result forward. It also indicates that there may be one or more neurons that can pass its value to the next layer at the same time.

In Layer-T, every neuron who has received value from Layer-S sums the value and then processes it using the Sigmoid Function just as Layer-S does. The equation is listed below.

$$O_k^T = f\Big(\sum_{i=1}^{NOF} O_i^S - \theta_k^T\Big). \tag{4}$$

Next, all neurons in this layer come to another competition, called the king competition which means the neuron whose output value outperforms others, can forward its value. It differs from the leakage competition in that, there is only one winner left.

The leakage competition and king competition play the same role, just like the mechanism in ART, where F1 or F2 can go on when it gets two or more pluses out of three.

Neurons in Layer-O receive value from Layer-T and process it by

$$O_l^O = W_{wi}^{TO} O_w^T. \tag{5}$$

The final result can be represented by the winner of neuron O_w in Layer-O, whose output value outperforms others, which means the input pattern P_q can be classified into class w. When the system is in testing mode, the next pattern should be learned; otherwise, it will go on with the next phase. The forward learning phase can be seen clearly in Fig. 2.

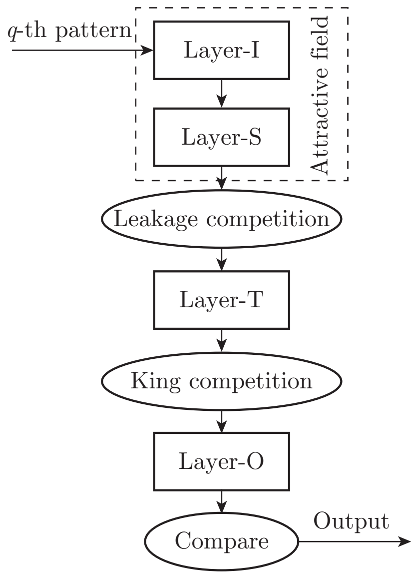

Fig. 2: Forward learning

3.2.2 Backward Tuning

When the system is in learning mode, error of the output layer would be calculated by equation

$$Error = \frac{\sum_{l=1}^{NOC} (e_l - O_l^O)^2}{NOC}. \tag{6}$$

According to the comparison between the error value and the default $ERROR$, there are different ways to tune the system backward using BP algorithm.

(1) If the error value is less than the $ERROR$ and the final result O_w is what the pattern P_q desired to be, which means one particular attractive field AF_j has covered P_q, the responsive center would be adjusted to hold the current pattern tightly.

Step 1 Select neuron S_b^w in Layer-S whose output value outperforms others in the same layer that has connection with the winner neuron T_w in Layer-T.

Step 2 Tune the responsive center and the responsive width by equations below [3].

(2) If the error outperforms the ERROR, calculate the error of each neuron in Layer-T by equation

$$Error = \frac{\sum_{l=1}^{NOC} (W_{tl}^{TO} - e_l)^2}{NOC}, \tag{7}$$

and then pick up the neuron with the least error T_m. After that, tune the connection weights and the thresholds from Layer-O to Layer-S based on BP strategy. The connection weight from T_k to O_l is tuned as

$$W_{kl}^{TO} = W_{kl}^{TO} - \tau(O_l^O - e_l)O_j^T, \tag{8}$$

the threshold of T_k can be tuned as

$$\theta_k^T = \theta_k^T - \tau O_k^T(1 - O_k^T) \sum_{l=1}^{NOC} W_{kl}^{TO}(O_l^O - e_l), \tag{9}$$

the connection weights from S_i to T_j can be tuned as

$$W_{ij}^{ST} = W_{ij}^{ST} - \tau O_i^S O_j^T(1 - O_j^T) \sum_{l=1}^{NOC} W_{jl}^{TO}(O_l^O - e_l), \tag{10}$$

and the threshold of S_i can be tuned as

$$\theta_i^S = \theta_i^S - \tau O_j^T(1 - O_j^T) \sum_{l=1}^{NOC} W_{jl}^{TO}(O_l^O - e_l). \tag{11}$$

Step 1 If the error value of T_m is less than the desired $ERROR$, sort out the neuron in Layer-S with the biggest output value S_b^m among the neurons who has directly connected to T_m. If this neuron S_b^m was one of the winners of leakage competition, tune the corresponding attractive field so as to cover it. It indicates that although the attractive field has not covered P_q, still we can adjust it to cover it. The equation is listed below:

$$\theta_{im} = \theta_{im} + \delta(f_i - \theta_{im}), \delta \in (0, 1). \tag{12}$$

If the neuron S_b^m was not one of the winners before, add a new neuron S_n into Layer-S. It means that none of the current attractive fields have covered P_q and cannot any more. The new added neuron S_n directly connects to neuron T_m. The threshold value and connection weight to T_m should be randomly set from $(-1, 1)$. Moreover, S_n and its Gaussian connections compose a new attractive field AF_n to cover the current patter P_q by set f_i as its responsive center θ_{in}. The responsive width is set as default value.

Step 2 If the error value of T_m outperforms the $ERROR$, two neurons S_n and T_n could be added into Layer-S and Layer-T respectively, which are directly connected to each other. The threshold values and the connection weights are randomly set from $(-1, 1)$. The new added neuron S_n and its connections to Layer-I compose a new attractive field AF_n which can cover the current pattern P_q by set its responsive center by f_i. The connection weight between T_n and the output layer neuron O_l is set as e_l.

After two new neurons have enrolled, connection weights from Layer-O to Layer-S along with its thresholds are tuned according to BP learning strategy (Refer Fig. 3).

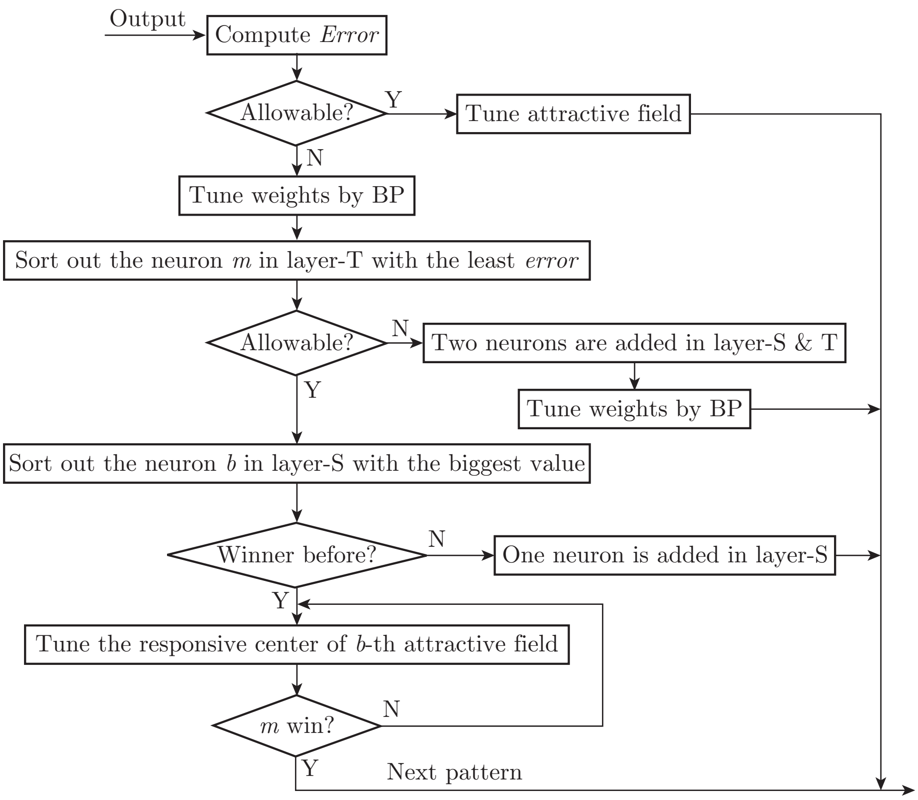

Fig. 3: Backward tuning

4 Benchmark Test

We perform a benchmark test called circle-in-the-square, which is one of the famous benchmark tests for ANNs. The aim of this test is to classify the nodes in the square into two categories, inside or outside the circle. Apparently, this is a non-linear problem.

We conduct this test on a PC, with CPU (Intel Core2 Duo P8400) 2.26 GHz and Memory 0.98 Mbt. The developing environment is Visual C++ 6.0. The learning set size is ranged from 10 to 50, and the testing set size is set as 1000. The input pattern in the learning set or the testing set is got through randomly coordinates $x, y \in (-1, 1)$. For comparing purpose, we take three models to perform this test. After a couple of experiments, we get some useful results shown in Fig. 4 – Fig. 6, and Table 1, plus the testing time for across three methods is approximate 1 s.

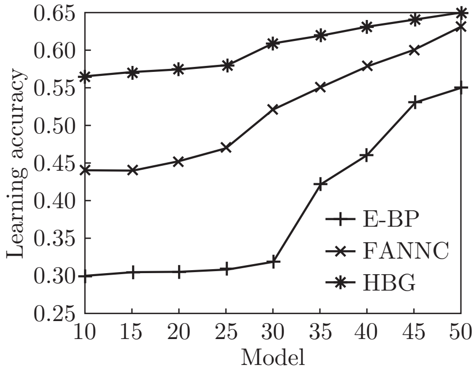

Fig. 4: Learning accuracy

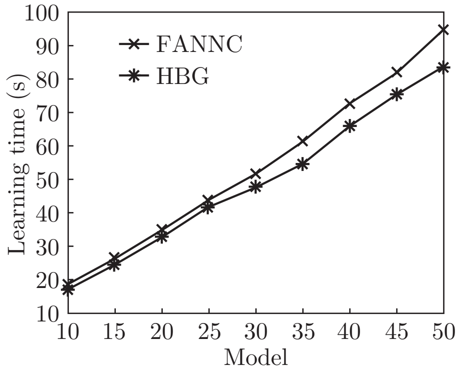

Fig. 5: Learning time

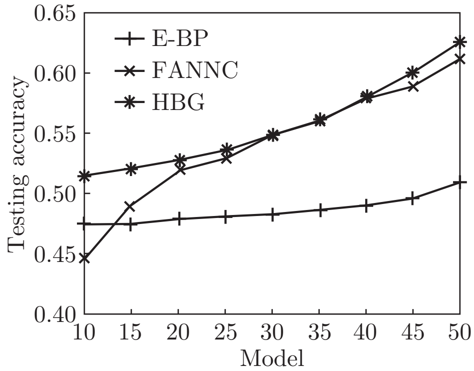

Fig. 6: Testing accuracy

Table 1: Learning time (s)

Model	E-BP	FANNC	HBG
10	273.6	17.5	18.1
15	436.1	26.3	24.9
20	582.5	34.7	32.6
25	762.9	43.5	41.8
30	958.2	51.6	47.3
35	1174.5	61.8	54.2
40	1365.4	72.4	66.3
45	1617.7	82.1	74.9
50	1849.3	94.5	82.8

As related to learning, Gaussian weights based solutions, i.e. FANNC and HBG, rule the arena no matter accuracy or time efficiency is assumed as a metric, leaving E-BP way behind, as depicted through Fig. 4, Fig. 5 and Table 1. The phenomenon can be explained by digging down to the born differences between ART and delta-rule, i.e., due to the inherent slow convergence drawback of BP, E-BP is not adequately enough to compete with ART-based hybrid solutions. For versus scenario between FANNC and HBG, we can see HBG is more accurate than FANNC, especially in the very beginning of the learning phase, which could be the leveraging profits of the deep adaption found on neuron thresholds and connection weights (tune the tunnel to the past misplaced class down to let it be less attractive, and tune the other tunnel up to make it more attractive) brought by back-propagation. With approaching the curve border between classes (two, in this experiment), every pattern on hand is getting clearer on where it might be clustered, which means attractive filed miss has low probability to happen and learning time from then on is highly saved, meanwhile the accuracy margin is also shrinking.

For testing, the time required for classify 1000 points is almost the same, 1 second approximately. Nevertheless, the accuracy is nearly paralleled with the learning phase with the exception on E-BP, which can be analyzed as long distance to reach its convergence point (see Fig. 4 and Fig. 6 for detail). For comparison between FANNC and HBG, accuracy differs around the initial part and the terminal part. When the first batch of 2D points is launched, typically border oriented approaching, the HBG model would necessarily confuse and usually misjudge them. FANNC can self-adapt its architecture to cover this accident, but not as profound as HGB does (threshold of neurons in hidden layers and related connection weights embraced), and this is what happen on the very beginning. Yet, this radical solution may have side effects for potential over tuned afterwards, and this drawback might account for the subsequent equivalent or slightly fall behind to FANNC around the central part. Misplacing, adapting, over-tuned, readapting and all the swing procedures, make the converging process goes a little faster to see in advance the convergence point not far away than conservatively balanced solutions, e.g., FANNC (check Fig. 6).

Note that, with the rising of learning set size, FANNC would approach HBG on both accuracy and time efficiency metrics, which implies HBG is not superior to FANNC near mature point.

5 Related Work

Considering pattern classification or recognition field, great numbers of diverse ANN models have born, what followed is a short list of recent work which are closely related to our research.

- **Feed-forward ANNs** B. Sokouti et al. [5] proposed a LevenbergCMarquardt feed-forward MLP ANN, and successfully applied it to cervical cancer classification in reality with 100% correct rate. Facing that evaluating fundamental changes in a firms technology adoption performance would always be costly and complex, S. Saberi and R. M. Yusuff [6] developed a feed-forward ANN training with BP algorithm, which can cluster companies into three groups with 72% accuracy rate. H. Hasanlou et al. [7] used a multilayer perceptron feed-forward ANN with stop training method to predict quality parameters of many industrial variables. To be noted, principal component analysis technique was applied to improve performance of generated ANN models.

- **Back-propagation ANNs** In computer-aided diagnosis, M. A. Mazurowskia et al. [8] focused on the effects of imbalanced datasets on ANN training between back-propagation and particle swarm optimization, and the result on real clinic data showed that the classical BP method is generally preferable over particle swarm optimization for imbalanced datasets. Additionally, in control area, back-propagation ANN had been trained to obtain the accurate control signal and achieve the precise control purpose [9]. For Chinese character recognition within vehicle license plates, F. Yang et al. [2] presented a parallel BP ANN, which can enhance the accuracy and performance of automated reading of license plates.

- **Adaptive resonance theory ANNs** M. Svaco et al. [10] proposed a novel ANN architecture ARTgrid based on adaptive resonance theory, and it is capable of clustering 2D object structures for specific robotic applications. For gesture spotting problem, P. K. J. Park et al. [11] proposed an adaptive distributed prediction technique to detect the start and end points of a continuous moving gesture, and the result showed that it is 2000 times faster than alternative methods. Aiming at improving the vigilance adaption of ART, L. Meng [12] introduced two methods, the Activation Maximization Rule (AMR) and the Confliction Minimization Rule (CMR), both of which can upgrade the robust performance of fuzzy ART to the vigilance parameter.

- **Associative memory ANNs** In order to study the effect of delay in dynamic behavior of associative memory ANN, X. Li [13] considered an n+1-dimensional bidirectional associative memory ANN model with multi-delay and explored an effective model by taking delay as a parameter, using the global Hopf bifurcation existence theorem and the Bendixson non-existent theorem. M. Wang and T. Zhou [14] focused on the Multidirectional Associative Memory (MAM) ANN which was said more advanced to realize associative memory. Based on the Brouwer fixed point theorem and Dini upper right derivative, it is confirmed that two of the equilibriums are exponentially stable.

6 Conclusion

Facing the slow convergence problem of ANN classifiers, a hybrid method, integrating BP strategy and Gaussian weights, named HBG is introduced. With this model, the initial pain of pattern learning could be alleviated, while retaining learning accuracy. Benchmark test verified that the method is feasible and effective for 2D pattern classification scenarios.

References

[1] F. M. Ham, I. Kostanic, Principles of Neurocomputing for Science and Engineering, McGraw-Hill College, 2000

[2] F. Yang, F. Yang, Character recognition using parallel BP neural network, Proc. of International Conference on Audio, Language and Image Processing, 2008, 1595-1599

[3] Z. Zhou, S. Chen, Z. Chen, FANNC: A fast adaptive neural network classifier, Knowledge and Information Systems, 2(1), 2000, 115-129

[4] M. Rezaeian-Zadeh, S. Zand-Parsa, H. Abghari, M. Zolghadr, V. P. Singh, Hourly air temperature driven using multi-layer perceptron and radial basis function networks in arid and semi-arid regions, Theoretical and Applied Climatology, 109(3), 2012, 519-528

[5] B. Sokouti, S. Haghipour, A. D. Tabrizi, A framework for diagnosing cervical cancer disease based on feedforward MLP neural network and ThinPrep histopathological cell image features, Neural Computing and Applications, 24(1), 2014, 221-232

[6] S. Saberi, R. M. Yusuff, Neural network application in predicting advanced manufacturing technology implementation performance, Neural Computing and Applications, 21(6), 2012, 1191-1204

[7] H. Hasanlou, N. Mehrdadi, M. T. Jafarzadeh, H. Hasanlou, Performance simulation of H-TDS unit of Fajr industrial wastewater treatment plant using a combination of neural network and principal component analysis, Journal of Water Resource and Protection, 4(5), 2012, 311-317

[8] M. A. Mazurowskia, P. A. Habasa, J. M. Zuradaa, Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance, Neural Networks, 21(2), 2008, 427-436

[9] Y. Cheng, K. Chen, L. Bai, M. Dai, Back propagation neural network based control for the heating system of a polysilicon reduction furnace, Review of Scientific Instruments, 84(12), 2013, 12508-12513

[10] M. Svaco, B. Jerbic, F. Suligoj, ARTgrid: A two-level learning architecture based on adaptive resonance theory, Advances in Artificial Neural Systems, 2014, 1-9

[11] P. K. J. Park, J. H. Lee, C. W. Shin, H. S. Ryu, Gesture recognition system based on adaptive resonance theory, Proc. of the 21st International Conference on Pattern Recognition (ICPR 2012), 2012, 3818-3822

[12] L. Meng, A. H. Tan, D. C. Wunsch, Vigilance adaptation in adaptive resonance theory, Proc. of the 2013 International Joint Conference on Neural Networks (IJCNN), 2013, 1-7

[13] X. Li, Dynamical quality in bidirectional associative memory neural network model, Journal of Beijing University of Aeronautics and Astronautics, 39(3), 2013, 371-375

[14] M. Wang, T. Zhou, Multistability in a multidirectional associative memory neural network with delays, Journal of Applied Mathematics, 2013, 1-10