1000-9825/2000/11(11)1451-09 ©2000 Journal of Software 软件学报 Vol.11, No.11

Research of Field Theory Based Adaptive Resonance Neural Network*

ZHOU Zhi-hua, CHEN Zhao-qian, CHEN Shi-fu

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

E-mail: daniel@aiake1.nju.edu.cn

http://aiake1.nju.edu.cn/~zhou

Received Feb. 9, 1999; accepted July 20, 1999

Abstract: In this paper, a Field Theory based adaptive resonance neural network algorithm FTART, which combines the advantages of Adaptive Resonance Theory and Field Theory, is proposed. FTART employs a unique approach to solve the conflicts between instances and extend classification regions dynamically. So that it does not need user to manually configure hidden units, and achieves fast training speed and high predictive accuracy. Moreover, a method named Statistic based Producing and Testing, which has the ability of extracting comprehensive and accurate symbolic rules from trained FTART, is proposed. Experimental results show that the symbolic rules extracted via this method can commendably describe the function of FTART.

Key words: neural networks; machine learning; rule extraction; adaptive resonance theory; field theory; knowledge acquisition; online learning; incremental learning

Adaptive Resonance Theory (ART)[1] is an important class of competitive neural learning models. The memory mode of these models is similar to that of life forms, and their memory capacity can increase as the learning instances increase. Moreover, ART models can perform real time online learning and can work under dynamical environments. So, these models have promising application prospect. On the other hand, Field Theory[2] is a class of relaxation models, which are the only neural models that need only one round training currently. And they are good heteroassociative classifiers, which have large memory capacities and can perform real time supervised learning with fast speed. In this paper, a new neural learning algorithm named FTART is proposed based on Adaptive Resonance Theory and Field Theory. FTART employs a unique approach to solve the conflicts between instances and extend classification regions dynamically. It overcomes the disadvantage of traditional feed-forward neural networks, which need user to manually configure hidden units, and achieves fast

* Project is supported by the National Natural Science Foundation of China under Grant No. 69875006 (国家自然科学基金) and the Natural Science Foundation of Jiangsu Province, China under Grant No. BK99036 (江苏省自然科学基金).
ZHOU Zhi-hua was born in 1973. He is a Ph.D. canadidate at the Department of Computer Science and Technology, Nanjing University. He received B.S. and M.S. degrees in computer science from Nanjing University in 1996 and 1998 respectively. His current research areas include neural networks, machine learning, evolutionary computing and data mining. **CHEN Zhao-qian** was born in 1940. She is a professor and doctoral supervisor of the Department of Computer Science and Technology, Nanjing University. Her current research areas include machine learning, knowledge engineering, and neural networks. **CHEN Shi-fu** was born in 1938. He is a professor and doctoral supervisor of the Department of Computer Science and Technology, Nanjing University. His current research areas include knowledge engineering, machine learning, image processing, and distributed artificial intelligence.

training speed and high predictive accuracy. Benchmark tests show that FTART is far better than BP in training time cost and predictive accuracy.

Because artificial neural network has stupendous ability of generalizing and dealing with nonlinear problems, it gets outstanding achievements that traditional symbolic mechanism cannot attain in many domains. But there exists an inherent disadvantage in ANNs, that is, concepts learned by ANNs are hard to understand, and it is difficult to give an explicit explanation for the reasoning process, because knowledge is represented in large assemblages of connection weights in the network. This has cumbered user from understanding the function of neural models, and limited them in applying those models to the task of knowledge discovery and knowledge refinement. We can overcome this disadvantage if we can extract comprehensible symbolic rules from neural networks. Nowadays, more attention has been paid to this field, and many fruits have been achieved[3~5]. In this paper, we propose a method named SPT (statistic based producing and testing), which comes from the view of functionality, to extract symbolic rules from trained FTART network. Experimental results show that the rules extracted via SPT are comprehensible and accurate, and can commendably describe the function of FTART.

The rest of this paper is organized as follows. In Section 1, we describe the Field Theory based adaptive resonance neural network algorithm FTART. In Section 2, we present the rule extraction algorithm SPT. In Section 3, we report on some benchmark tests and comparisons with FTART. In Section 4, we give out rule extraction experimental results and comparisons with SPT. And finally in Section 5, we conclude the paper.

1 Field Theory based Adaptive Resonance Algorithm FTART

1.1 Architecture

The FTART network is composed of four layers of units. Figure 1 shows its architecture. The activation function of hidden units is Sigmoid function, and Gaussian weights connect the input units with the second layer units. FTART uses the second layer units to classify inputs internally, uses the third layer units to classify outputs internally, and sets up associations between those two layers to implement supervised learning.

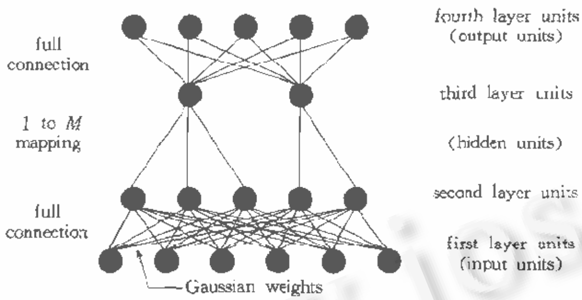

Fig. 1 The architecture of FTART

Except the connections between the first and second layer units, all connections are bi-directional. The feedback connections, whose function is just transmitting feedback signal, are always set to 1. In fact, this architecture is a typical competitive neural network classifier, which can partition the instance space into arbitrary shapes.

1.2 Mathematical descriptions

FTART introduces the notion of attracting basin, which is proposed in Field Theory[2]. It adopts Gaussian weights as the connections between the first and second layer units. Assume that pattern input to the first layer units is $A_k=(a_1^k,a_2^k,\ldots,a_n^k)$ $(k=1,2,\ldots,m)$, then the output of the second layer unit j is:

$$b_j=\sum_{i=1}^{m} e^{-\left(\frac{a_i^k-\theta_{ij}}{\alpha_{ij}}\right)}, \tag{1}$$

where θ_{ij} and α_{ij} are respectively the responsive central value and the responsive characteristic width of the Gaussian weight that connecting the input unit i with the second layer unit j. They determine a geometrical attracting basin jointly. The relationship between b_j and other parameters satisfies Eq. (2):

$$\begin{cases} a_i^k = \theta_{ij} \Rightarrow b_j = 1 \\ a_i^k \neq \theta_{ij} \Rightarrow b_j \to 0 \end{cases} \tag{2}$$

Because the dynamical property of a Gaussian weight is entirely determined by its responsive central value and responsive characteristic width, learned knowledge can be encoded in the weight through only modifying those two parameters. Thus, during the training process, if the input pattern is located in the attracting basin determined by θ_{ij} and α_{ij} of a Gaussian weight, no parameter will be changed. Else the nearest attracting basin will be found and modulated according to the relationship between the input pattern and the original basin, so that the input pattern could be covered by the basin. The modulation is performed by adjusting both θ_{ij} and α_{ij}, as shown in Eqs. (3) and (4):

$$\theta'_{ij} = \begin{cases} \dfrac{\theta_{ij} + 0.3\alpha_{ij} + a_i^k}{2} & a_i^k \in (-\infty, \theta_{ij} - 0.3\alpha_{ij}) \\ \theta_{ij} & a_i^k \in [\theta_{ij} - 0.3\alpha_{ij}, \theta_{ij} + 0.3\alpha_{ij}] \\ \dfrac{\theta_{ij} - 0.3\alpha_{ij} + a_i^k}{2} & a_i^k \in (\theta_{ij} + 0.3\alpha_{ij}, +\infty) \end{cases} \tag{3}$$

$$\alpha'_{ij} = \begin{cases} \dfrac{\theta_{ij} + 0.3\alpha_{ij} - a_i^k}{2} & a_i^k \in (-\infty, \theta_{ij} - 0.3\alpha_{ij}) \\ \alpha_{ij} & a_i^k \in [\theta_{ij} - 0.3\alpha_{ij}, \theta_{ij} + 0.3\alpha_{ij}] \\ \dfrac{a_i^k - \theta_{ij} + 0.3\alpha_{ij}}{2} & a_i^k \in (\theta_{ij} + 0.3\alpha_{ij}, +\infty) \end{cases} \tag{4}$$

The modulation involves feedback signals and may involve iterative adjusting. However, because the typical attractor of the basin will be more and more close to the input pattern, the adjusting resonance procedure is to stabilize at a point, where the input pattern is covered by the attracting basin. This stabilization property is a characteristic that FTART inherits from Adaptive Resonance Theory.

The output of the third layer unit h is computed according to Eq. (5):

$$c_h = f(b_j v_{jh} - \theta_h), \tag{5}$$

where f is a Sigmoid function as shown in Eq. (6):

$$f(u) = \frac{1}{1 + e^{-u}}. \tag{6}$$

In Eq. (5), θ_h is the threshold of unit h. b_j is the output of the second layer unit j, which is the winner among the second layer units connecting with unit h. v_{jh} is the weight connecting unit j with unit h. Attention should be paid to that v_{jh} is always equal to 1. The relationship between c_h and other parameters satisfies Eq. (7):

$$\begin{cases} b_j v_{jh} - \theta_h \gg 0 \Rightarrow c_h \to 1 \\ b_j v_{jh} - \theta_h \ll 0 \Rightarrow c_h \to 0. \end{cases} \tag{7}$$

Different from traditional feed forward neural network algorithm BP, FTART does not need user to manually configure hidden units before training because it possesses the characteristic of Adaptive Resonance Theory, which could dynamically increase units. When new patterns are fed, if necessary, FTART will append several new units in the hidden layers, connect them to a part of existing units, and adjust connection weights, in order that the new patterns could be covered. On the other hand, BP uses supercubes to partition the instance space. When training examples do not fully cover the instance space, the learning result is the simple mergeing of empty holes. This leads to the need of iterative learning. Comparatively, FTART uses super-ellipsoids to partition the instance space, which is an advantage of Field Theory. The responsive central values of Gaussian weights are corresponding to the characteristics of components of the learned patterns. Through adopting such framework, the learning system can make complicated partitions for instance space by only using the first layer units.

So, FTART needs only one round training and its training speed is much faster than BP. Moreover, the predictive accuracy of FTART is better than BP because the super-ellipsoids have better filling effect than supercubes.

1.3 Algorithm description

FTART algorithm is described in Table 1, and its flowchart is shown in Fig. 2.

Table 1 Field theory based adaptive resonance neural network algorithm FTART

1.	Create the initial network based on the components of the input and output vectors. And send input pattern of the first training instance to the input units.
2.	Input units send input values to the second layer units through Gaussian weights. Compute outputs of the second layer units according to Eq. (1).
3.	Carry out winner-take-all style competition among the second layer units connected with the same third layer unit. The winner sends its output to the third layer unit. Compute the output value of the third layer unit according to Eq. (5).
4.	Carry out winner-take-all style competition among all the third layer units. The winner sends its output value to the fourth layer units.
5.	Compute the error between real network output and the expected output. If the error is in the allowance range, goto step 6, else goto step 7.
6.	The output units release a stimulus signal and feed it back layer by layer to the second layer unit which is not only a winner but also connected with the winner of the third layer units. Then, the corresponding θ_{ij} and α_{ij} of the Gaussian weights are adjusted according to Eqs. (3) and (4). Goto step 10.
7.	Find out the third layer unit whose error is the minimum through comparing the expected network output with the characteristic output of all the third layer units. If the error is in the allowance range, goto step 8, else goto step 9.
8.	Find out the second layer unit which is not only a winner but also connected with the selected third layer unit. Adjust corresponding θ_{ij} and θ_{ij} according to Eqs. (3) and (4) until the selected third layer unit can win its competition. Goto step 10.
9.	Append two units in the hidden layers, one in the second layer, the other in the third layer. The new second layer unit is connected with all the input units. The θ_{ij} of the Gaussian weights are set to the current input values of the first layer units, and the α_{ij} are set to a default value. The new third layer unit is connected with all the output units. The weights are set to the output components of the current instance. Furthermore, the two new units are connected with each other. All the feed-forward and feedback weights between them are set to 1.
10.	If all training instances have been fed, the training procedure terminates. Else send the input of the next instance to input units. Goto step 2.

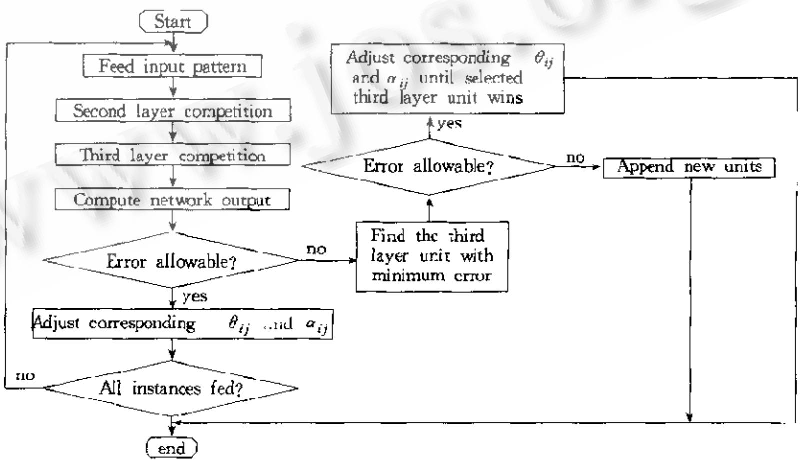

Fig. 2 Flowchart of field theory based adaptive resonance neural network algorithm FTART

2 Extracting Rules from FTART Network

Although FTART is a good neural learning algorithm, it still has a disadvantage, that is, the learned knowledge is concealed in large amount of real value connections. So, the knowledge encoded in the neural network is hard to understand by users, and the reasoning process is implicit and hard to be validated. If we can extract comprehensible symbolic rules from trained FTART network, we can not only help users to understand the function of FTART, but also push FTART into more and more application fields.

The training of neural network needs large amount of data. If we use a trained network to classify the examples in an instance set, there will be both correct and error results. Nonetheless, those results sufficiently reflect the function of the network. We can construct a new example set through combining the original inputs with the neural network classifications. If we can extract symbolic rules from this example set, we can not only use those rules to describe the function of the trained neural network but also give out comprehensible symbolic explanation for the neural network reasoning process. This is the start point of proposing SPT method.

The neural network training examples unnecessarily fully cover the whole instance space, and there are possibilities of introducing noise in data collection. So, extracting rules from only the original inputs and their neural network classifications can not guarantee that those rules effectively describe the generalization ability of the neural network. To address this problem, we introduce statistics into SPT. Only when the classifications for some extra new examples determined by the extracted rules are equivalent to those determined by the neural network to some extent, we accept the rules. Else we reject those rules and start to search for other possible rules.

Continuous features are often inevitably occurring and playing important roles in real application fields. One of the most distinct advantages of neural networks against symbolic learning is that neural networks can commendably deal with continuous features. However, extracting rules from neural networks that have continuous inputs is very difficult, and there is no very successful method. Some researchers discretize all continuous inputs before rule extraction[5]. Although this is helpful to some extent, it suffers from the subjective factor, because no one knows how many subclasses the values of a continuous attribute should be clustered into. Comparatively, SPT does not perform discretization at the beginning. It always extracts rules from discrete features. And only when necessary, a continuous feature that has the best clustering effect is discretized to be a new discrete feature. This style of processing reduces the combinatorial complexity and computational cost because it decreases the rule space involving continuous features.

Attention should be paid to that the rules extracted through SPT are ranked. The firstly extracted rule has the highest rank and the latest the lowest rank. This characteristic makes the rule set have a compact appearance.

The SPT algorithm is described in Table 2, and its flowchart is shown in Fig. 3. The λ in the algorithm, which defines the support of the current rule, is computed according to Eq. (8):

$$\lambda=\frac{S_{XY}}{S_X}, \tag{8}$$

where S_{XY} is the amount of instances in S' that possess the discrete input attribute combination X and fall into class Y, S_Y is the amount of instances in S' that possess X.

Table 2 Rule extraction algorithm SPT

1.	Split the instance set S, which is large enough, into two parts: S_1 and S_2. Regard S_1 as training set to train FTART, and leave S_2 alone. The amount of instances in S_2 is more than that in S_1. This is because we use more instances for rule extraction than for training.
2.	Use trained FTART network to determine the classifications of instances in S_2. Construct a new instance set S' by combining the original input with the network output. Set λ to 100%.
3.	If there exists in S' a discrete input attribute combination X, which makes instance that possesses it fall into a certain class with probability λ, construct a rule R_X by regarding the combination and class as the rule antecedent and consequent respectively. (If more than one combination exists, select the one that covers the largest amount of instances.) Goto step 4. Else goto step 6.
4.	Fix the values of attributes appearing in X; create N new instances by randomly varying the values of other input attributes. Use trained FTART network and extracted rule set to determine the classifications of the new instances. If the fidelity of classifying results is not satisfying, reject R_X and goto step 3 to find other discrete input attribute combination. Else accept R_X and goto step 5.
5.	Delete instances covered by R_X from S'. If S' is empty, goto step 8. Else goto step 3.
6.	If there exist in S' undiscretized continuous input attributes, select the one (A_c) that has the best clustering effect to discretize, goto step 3. Else goto step 7.
7.	Decrease λ. If λ is less than a pre-set value λ_{pre}, goto step 8. Else recover all the values of discretized continuous attributes of instance in S', goto step 3.
8.	Merge the rules that have not only successive ranks but also the same rule consequent.
9.	End.

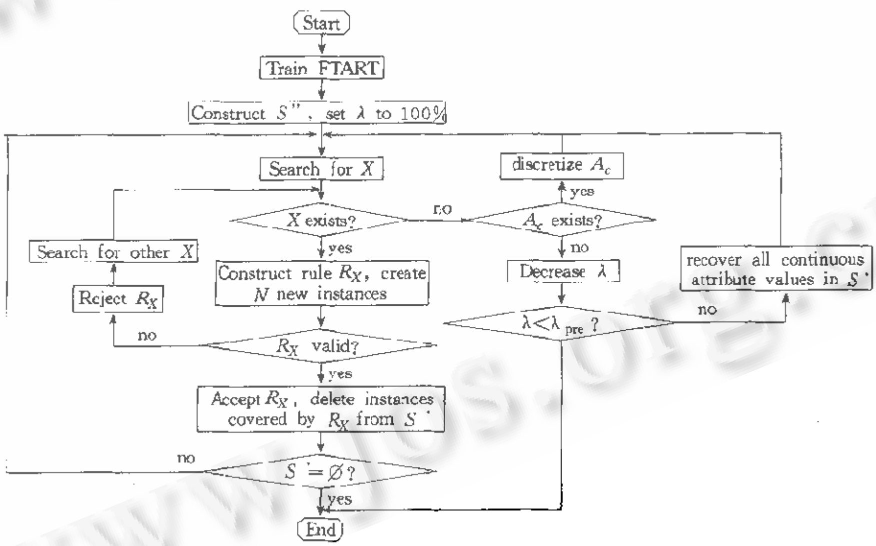

Fig. 3 Flowchart of rule extraction algorithm SPT

3 FTART Benchmark Tests

3.1 Benchmark test 1: circle-in-the-square

Circle-In The Square is specified as a benchmark test problem for system performance evaluation in the DARPA ANNT (artificial neural network technology) program[6]. The details of this task are described in Ref. [6]. The testing results of FTART and standard BP[7] are shown in Table 3 (The machine used is Pentium MMX 200MHz, 32MB RAM). In order to control the experimental time, we limit the iterative epochs of BP to at most 200. Since it is very difficult to determine the amount of hidden units of BP, in each experiment we use

3 BP networks whose amount of hidden units is respectively 5, 15 and 25. The data of BP in Table 3 are the average value of those 3 networks.

Table 3 shows that the training set accuracy of FTART is always 100%. And the testing set accuracy of FTART achieves higher than 98.9% when only 500 training examples are fed. Attention should be paid to that the training time of FTART is about one magnitude of order less than BP even we limit the largest iterative epochs of BP to 200. So, the learning accuracy, generalization ability and efficiency of FTART are obviously superior to those of BP. Figure 4 shows the testing result of FTART when 1,000 training examples are fed. In the figure, black and white points respectively represent instances that are judged to be in the circle or out of the circle by trained FTART. There are also some gray spots existing in Fig. 4, which denote undetermined areas that cannot be distinguished by trained FTART because the training examples do not well spread in the instance space.

Fig. 4 The testing result of FTART when 1,000 training examples are fed. The classification accuracy is 99.1%

Table 3 Comparison of circle-in-the square testing results

	Training set size	Training set accuracy (%)	Testing set accuracy (%)	Training time (second)
BP	200	72.0	71.3	4 769
	500	93.5	93.5	11 273
	1,000	96.1	95.0	22 353
FTART	200	100	94.0	752
	500	100	98.9	2 012
	1,000	100	99.1	3 976

3.2 Benchmark test 2: tell-two-spirals-apart

Tell-Two-Spirals-Apart is a neural network benchmark test proposed by Wieland[8]. This task is described in detail in Ref. [8]. The testing results of FTART and standard BP are shown in Table 4 (The machine used is Pentium MMX 200MHz, 32MB RAM). We still limit the largest iteration epochs of BP to 200, and use 3 BP networks for training whose amount of hidden units is respectively 5, 15 and 25. The data of BP in Table 4 are the average value of those 3 BP networks.

Table 4 Comparison of tell-two-spirals-apart testing results

	Training set accuracy (%)	Testing set accuracy (%)	Training time (second)
BP	52.0	51.3	4 413
FTART	100	100	1 024

Table 4 shows that the training set accuracy and testing set accuracy of FTART are 100%; both are far superior to BP. Moreover, the training time of FTART is far less than BP. Obviously, FTART has stronger learning ability than BP.

4 Examples of Rule Extraction

We use Human Race Classifying problem to demonstrate the experimental results of extracting rules from trained FTART via SPT. The instance set used in this task is an extension of that used in Ref. [9]. There are 160 instances in this set, and each instance has two discrete attributes and two continuous attributes, which are respectively "Hair Color", "Eye Color" and "Height", "Weight". Because there are continuous attributes existing, SPT may perform discretization in the rule extraction procedure. Here we only use the simplest c-Means clustering method. Since it is difficult to evaluate the accuracy of the extracted rules, we use 5-fold cross validation by dividing the instance set into 5 subsets and correspondingly constructing 5 FTART networks. For each we use four subsets to extract rules and the rest one to test. The rules extracted via SPT are slightly varied because the training sets are different in the 5 experiments. A typical extracted rule set is shown in Table 5.

Table 5 Rules extracted for Human Race Classifying problem via SPT

Rule No.	Rules
1	(hair=gray)→Black
2	(hair=blond) ∨ (hair=red) ∨ (height>194.1)→White
3	((hair=dark) ∧ (height<163.8)) ∨ ((hair=dark) ∧ (eye=dark))→Yellow
4	(weight<98.7) ∨ (hair=dark)→Black

We compare the testing set accuracy of FTART, C4.5 decision tree[10] and the rule set extracted by SPT at each experiment. Figure 5 shows the comparison of the results.

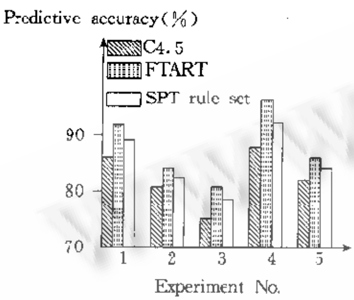

Fig. 5 Comparison of human race classifying problem testing set accuracy

Figure 5 reveals that the neural learning model has the highest accuracy. The reason is that FTART has better generalization and noise resistance ability, so that it could achieve satisfying learning effect even though the training examples are noisy and do not fully cover the whole instance space. Figure 5 also shows that the testing set accuracy of the rule set extracted via SPT is very close to that of FTART. This illustrates the validity of SPT.

Attention should be paid to that the testing set accuracy of the rule sets extracted via SPT is superior to that of C4.5 decision trees in all the 5 experiments. This result is consistent with that of Setiono and Craven[4,5]. We believe that it is because the extracted rules may benefit from the generalization ability of FTART, so that they can do better predictions than decision trees.

5 Conclusions

This paper has proposed a Field Theory based adaptive resonance neural network algorithm FTART, which combines the advantages of Adaptive Resonance Theory and Field Theory. FTART does not need to manually configure hidden units, and achieves fast learning speed and strong generalization ability. Benchmark tests show that FTART is far better than BP in both learning accuracy and training speed. Moreover, this paper has also proposed an algorithm named Statistic based Producing and Testing, which is designed to extract symbolic rules from trained FTART network. Experimental results show that SPT can extract comprehensive and accurate symbolic rules, which can do great help to improve the comprehensibility and reasoning transparency of FTART.

Acknowledgements The authors would like to thank WEI Wen-long and SUN Chen for their fruitful work. The comments and suggestions from the anonymous reviewers greatly improve this paper.

References:

[1] Carpenter, G.A., Grossberg, S. The ART of adaptive pattern recognition by a self-organizing neural network. Computer, 1988,21(3):77~88.

[2] Wasserman, P.D. Advanced Methods in Neural Computing. New York: Van Nostrand Reinhold Press, 1993. 14~34.

[3] Sestito, S., Dillon, T. Knowledge acquisition of conjunctive rules using multilayered neural networks. International Journal of Intelligent Systems, 1993,8(7):779～805.

[4] Setiono, R., Liu H. Understanding neural networks via rule extraction. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence. Montreal, Canada: Morgan Kaufmann Publishers, Inc., 1995. 480～485.

[5] Craven, M.W., Shavlik, J.W. Extracting Tree-Structured Representations of Trained Networks. In: Touretzky, D., Mozer, M., Hasselmo, M. eds. Advances in Neural Information Processing Systems (volume 8). Cambridge, MA: MIT Press, 1996. 24～30.

[6] Carpenter, G., Grossberg, S., Markuzon, N. *et al*. Fuzzy ARTMAP: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Transactions on Neural Networks, 1992,3(5):698～713.

[7] Rumelhart, D., Hinton, G., Williams, R. Learning representation by backpropagating errors. Nature, 1986,323(9):533～536.

[8] Lang, K.J., Witbrock, M.J. Learning to tell two spirals apart. In: Touretzky, D., Hinton, G., Sejnowski, T. eds. Proceedings of the 1988 Connectionist Models Summer School. San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1989. 52～59.

[9] Chen Z, Liu H, Zhou R, *et al*. A hybrid algorithm for multi-concept acquisition & its application. Chinese Journal of Computers, 1996,19(10):753～761 (in Chinese).

[10] Quinlan, J.R. C4.5: Programs for Machine Learning. San Mateo, CA: Morgan Kaufmann Publishers, Inc., 1993.

附中文参考文献:

[9] 陈兆乾,刘宏,周戎,等.一种混合型多概念获取算法HMCAP及其应用.计算机学报,1996,19(10):753～761.

基于域理论的自适应谐振神经网络研究

周志华, 陈兆乾, 陈世福

(南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093)

摘要:提出了一种基于域理论的自适应谐振神经网络算法FTART,有机结合了自适应谐振理论和域理论的优势,以一种独特的方式解决了示例间冲突和分类区域的动态扩展,不仅不需要手工设置隐层神经元,还可以获得了较快的训练速度和较高的预测精度.同时还提出了一种可以从训练好的FTART网络中抽取可理解性好、精度高的符号规则的方法,即基于统计的产生测试法.实验结果表明,用该方法抽取的符号规则可以较好地描述FTART的功能.

关键词:神经网络;机器学习;规则抽取;自适应谐振理论;域理论;知识获取;在线学习;增量学习

中图法分类号:TP183　　**文献标识码**:A