Article ID: 1000 - 8152(2002)03 - 0435 - 03

New Neural Network Realization Algorithm for Neyman-Person Criterion

WANG Qi, SHEN Guofeng and ZHANG Zhaoli

(Department of Automation Measurement and Control Engineering, Harbin Institute of Technology·Harbin, 150001, P.R.China)

Abstract: Neyman-Person criterion in hypothesis testing is a method based on the probability rate for problems like classification, detection, and pattern recognition. Solutions through neural network to those problems would be very desirable. However, the traditional least square learning algorithms, like backpropagation, provide no guarantee for success. This paper intends to improve a kind of non-least-square learning algorithm, decide the criterion of the probability distribution and give a better algorithm based on the absolute error. Aside from theoretical argument, the proposed algorithm is examined on a simulated problem and compared with other algorithms. The simulative result proves that the new algorithm has fewer errors and is more suitable for the Neyman-Person criterion.

Key words: neural network; data fusion; hypothesis testing; Neyman-Person criterion

Document code: A

Neyman-Person 准则的神经网络实现新算法

王 祁 沈国峰 张兆礼

(哈尔滨工业大学自动化测试与控制系·哈尔滨,150001)

摘要: 假设检验中 Neyman-Person 准则是一种基于似然比的信号分类、检测、识别方法.神经网络是实现这种判定准则的优选方案,但是传统的最小平方学习算法,如 BP 算法等,往往不能取得全局最优解.本文针对一种非最小平方学习算法,提出了一种概率分配原则,并给出了一种 Neyman-Person 准则的神经网络实现新算法.文中对新算法在假设检验中的应用进行了仿真验证,结果表明新算法具有更小的误差,更加适用于 Neyman-Person 准则.

关键词: 神经网络; 数据融合; 假设检验; Neyman-Person 准则

1 Introduction

Hypothesis testing plays an important role in object detection, which is based on the statistic theory and has a lot of criterions, such as Bayes criterion, Neyman-Person criterion, min-max criterion and so on. These rules are widely used in radar signal detection, multi-sensor NDT (non-destructive testing) data fusion and medical diagnosis. Many research works have been carried out to realize these criterions and methods. Solutions through neural network to those problems would be more popular.

BP algorithm is a widely used algorithm for feed-forward neural network[1]. It is a least square learning algorithm and has some demerits[2,3] such as slow learning speed, less stabilization of the learning and memory, and so on. In some cases its operation result is likely to only get the local minimal point. Therefore it provides no guarantee for success to realize those criterions in hypothesis testing. Barnard[4] gave an example to illustrate that BP algorithm can not be used to realize the Bayes rule. In this paper, we improve one non-least-square learning algorithm to realize Neyman-Person criterion used in data fusion, decide the principle of the probability distribution and give a better algorithm based on the absolute error.

2 New neural network realization algorithm for N-P criterion

Neyman-Person criterion is very important in data fusion. Its object is to maximize the detection probability subject to a given false alarm probability. We will make some definitions before the following detailed discussion. H_1 and H_0 represent two hypotheses. β represents

Received date: 1999 - 11 - 12; Revised date: 2001 - 12 - 29.

the detection probability (p_d), which is the probability of deciding H_1 when H_1 is active. α represents the false alarm rate (p_f), which is the probability of deciding H_1 when H_0 is active.

As for the neural network is shown in Fig. 1, there are N neurons in the input layer, whose input is x_i, response function is $T(x_i)$, threshold is θ_i, and output is u_i. There is one neuron F in the ouput layer, whose response function is $T_F(x_i)$, threshold is θ_F, and output is u_F. The weights between the input neurons and the output neuron are w_i, and

$$T(x) = \frac{1}{1 + e^{-x}}, \tag{1}$$

$$T_F\{(u_i)\} = \sum_{i=1}^{N} w_i u_i,\ w_i > 0 \text{ for } i = 1,2,\cdots,N. \tag{2}$$

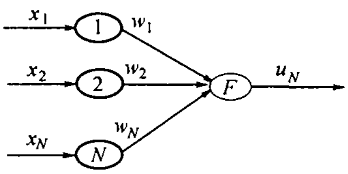

Fig. 1 Neural network structure

In the input layer, the input x_i can be considered as the ith sensor's output. The output of the input neuron can be calculated as follows

$$u_i = \begin{cases} 1, & T(x_i) \geqslant \theta_i, \\ 0, & T(x_i) < \theta_i. \end{cases} \tag{3}$$

In the output layer, neuron F is the fusion center. There must be a decision 'randomizing constant' r, thus we have

$$u_F = \begin{cases} 1, & \text{if} \sum_{i=1}^{N} w_i u_i > \theta_F, \\ 1 \text{ with probability } r, & \text{if} \sum_{i=1}^{N} w_i u_i = \theta_F, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where $r \in [0,1]$.

For the neural network shown in Fig. 1, the new algorithm is to assign the given P_f into the different stages in the data fusion according to some principles that the input layer realizes the $g(\alpha)$ and the output layer realizes $g'(\alpha)$. Then the samples $\{y_i\}_{i=1}^{KN}$ are input to get the trained parameters $\{\theta_1, \theta_2, \cdots, \theta_N, w_1, w_2, \cdots, w_N, \theta_F, r\}$. In the layer-by-layer (LBL) algorithm[5], both the input layer and the output layer realize the false alarm probability α. However, because of the uncertainty and inadequacy of the input samples, the problem becomes very complex. So the LBL algorithm has big error and need more training data. In the new algorithm, we denote $g'(\alpha) = \alpha$. If the data input to the input neurons have the same distribution, we have

$$g(\alpha) = C_C\left(1 - \frac{N_k}{k}\right)\alpha, \tag{5}$$

where $0 < C_C < 2$. If $N < k$, then $N_k = N$. If $N \geqslant k$, then $N_k = 1$. If the data transferred to the input neurons have different distributions, we have

$$g^i(\alpha) = w^i \alpha, \tag{6}$$

where $g^i(\alpha)$ is the false alarm probability realized by the neuron i. w^i is the parameter defined according to the distribution and $0 < w^i < 2$.

2.1 Determination of the thresholds of the input neurons

Under the following assumptions:

1) The input data x_i is independent.

2) For the function $f(x, l)$, $\exists\, k_0, k_1 (0 < k_0 < k_1 < \infty)$,

$$k_0(l - \theta)^2 \leqslant \mathrm{E}\{f(x, l)\}(l - \theta) \leqslant k_1(l - \theta)^2. \tag{7}$$

3) For parameter l: $\mathrm{var}\{f(x, l)\} \leqslant \sigma^2 < \infty$ and $\{c_n\}$ satisfies:

$$\sum_{n=1}^{\infty} c_n = \infty \text{ and } \sum_{n=1}^{\infty} c_n^2 \leqslant \infty, \tag{8}$$

Blum[6] certified that

$$l_{n+1} = l_n - c_n[f(x_n, l_n)],\ n = 1,2,3,\cdots \tag{9}$$

must converge to θ with probability one.

For the neural network shown in Fig. 1, we give a recursion formula:

$$\theta_{n+1}^i = \theta_n^i + c_n^i[u(y_n^i, \theta_n^i - g(\alpha)], \tag{10}$$

where y_n^i is the data deriving from H_0. θ_0^i is the random value of the neuron i. $c_n^i = c/n$, where c is a positive integer. According to formula (10), the first layer will realize false alarm probability when θ_n^i is converged. If the sensor's data have the same distribution, the training samples $\{y_j\}_{j=1}^{KN}$ are input into one neuron, and get the value of θ adapted to all input neurons. If the sensor's data have different distributions, the training samples $\{y_j^i\}_{j=1}^{KN}$ are input into each neuron, and get the threshold value θ_i.

2.2 Determination of the weights

Let $P_i(u_1, u_2, \cdots, u_N)$ be the multivariate probability distribution function of $u_1, u_2, \cdots, u_N$ under hypothesis $H_i, i = 0,1$. Define $g(\alpha) = a, g(\beta) = b$, we have

$$T_F\{(u_i)\} = \log \frac{P_1(u_1, u_2, \cdots, u_N)}{P_0(u_1, u_2, \cdots, u_N)} =$$

$$\sum \log \frac{P_1(u_i)}{P_0(u_i)} = \sum \log \frac{b_i^{u_i}(1 - b_i)^{1-u_i}}{a_i^{u_i}(1 - a_i)^{1-u_i}}, \tag{11}$$

i.e.:

$$T_F\{(u_i)\} = \sum_{i=1}^{N} w_i u_i = \sum_{i=1}^{N} \log \frac{b_i(1 - a_i)}{a_i(1 - b_i)} u_i. \tag{12}$$

Absorbing the constant $\sum \log[(1 - b_i)/(1 - a_i)]$, we have

$$w_i = \log \frac{b_i(1 - a_i)}{a_i(1 - b_i)}. \tag{13}$$

2.3 Determination of the threshold of the output neuron

In the training process, $\{y_i\}_{i=1}^{KN}$ are inputs to the neural network and the parameters $\theta_1, \theta_2, \cdots, \theta_N$, w_i do not change. According to formula (10), we have

$$\begin{cases} r_{n+1} = r_n - c_n'\{u_F(\theta_F, r_n) - g'(\alpha)\}, \\ \theta_F(n + 1) = \theta_{Fn} - [r_{n+1}], \text{ for } n = 1,2,\cdots,KN, \\ r_{n+1} = r_{n+1} - [r_{n+1}], \end{cases} \tag{14}$$

where $[r_{n+1}]$ represents the integer part. θ_{F1} is a random value. $c_n' = c'/n$, c' is a positive integer. In the second layer, $g'(\alpha) = \alpha$. Then we can determinate $\{\theta_1, \theta_2, \cdots, \theta_N, w_1, w_2, \cdots, w_N, \theta_F, r\}$.

3 Simulations and comparisons

In this section, the effect of the new algorithm on an actual hypothesis testing problem is simulated. The algorithms compared are Bayes optimization rule and LBL algorithm. In order to give a distinct result, we present a simple problem. Assume H_0, H_1 subject to the Gaussian distribution, and

$$f_0 = N(0,1), \ f_1 = N(1,1). \tag{15}$$

For the Bayes optimization problem realized by BP algorithm, we assume that $P(H_0) = P(H_1)$. The Bayes optimization fusion rule is shown as follows

$$\sum_{i=1}^{N} [u_i \log \frac{1 - P_{m_i}}{P_{f_i}} + (1 - u_i) \log \frac{P_{m_i}}{1 - P_{f_i}}] \gtrless \log \frac{P_1}{P_0}, \tag{16}$$

where $P_{m_i} = P(u_i = 0/H_1), P_{f_i} = P(u_i = 1/H_0)$. For the new algorithm, note that $c_c = 1$, so we have

$$g(\alpha) = (1 - \frac{N_k}{k})\alpha, \tag{17}$$

where $\alpha = 0.1$, $N_k = 8$. The simulation results are shown in Fig. 2 and Fig. 3.

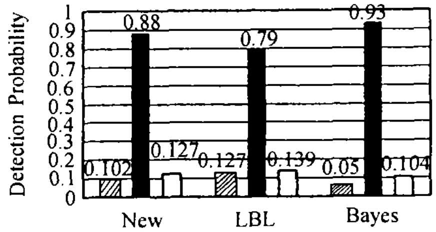

Fig. 2 Comparison of three algorithms

Figure 2 shows the result that when 1500 training samples are given, the new algorithm has bigger detection probability than LBL algorithm and the false alarm probability gained by the new algorithm is nearer to α.

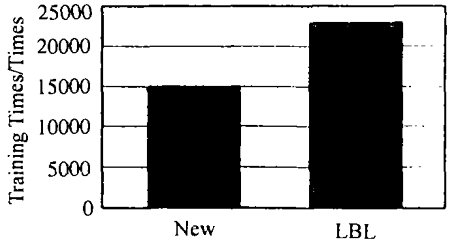

Fig. 3 Comparison result of two algorithms

Figure 3 shows the result when the false alarm probability is given. In order to realize the same false alarm probability, the speed of the new algorithm is faster than LBL algorithm

4 Conclusions

This paper presents a new non-least-squares neural network realization algorithm for the Neyman-Person criterion. The distribution principle of the false alarm probability is also discussed in this new algorithm. The simulation results illustrate that this new algorithm is more suitable to realize Neyman-Person criterion and provides a useful tool for multi-sensor data fusion based on the neural network.

References

[1] Werbos B J. Backpropagation troth time: What it does and how to do it [J]. Proc. IEEE, 1990, 78(10): 1550 - 1560

[2] Ruck D and Roges S. The multilayer perceptron as an approximation

(Continued on page 441)

References

[1] Liberzon D and Morse S A. Basic problems in stability and design of switched systems [J]. Control Systems Magazine, 1999, 19(5): 59 - 70

[2] McClamroch N H, Rui C, Kolmanovsky I, et al. Hybrid closed loop systems: a nonlinear control perspective [A]. In Proc. 36th Conf. Decision and Control [C], San Diego, California, USA, 1997, 114 - 119

[3] Boyd S, Ghaoui L E, Feron E, et al. Linear matrix inequalities in system and control theory [A]. SIAM Studies in Applied Mathematics [M]. Philadelphia, PA: SIAM, 1994, Vol. 15

[4] Brannicky M S. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems [J]. IEEE Trans. Automatic Control, 1998, 43(4): 475 - 482

本文作者简介

SUN Hongfei Born in 1970. He received the B. S. degree in mathematics in 1992 from Northeastern Normal University, China, and the M. S. degree in control theory and applications in 1998 at Northeastern University, China. His main research interests include hybrid systems, robust control. Email: zequn_sun@sina.com

LU Chunming Born in 1968. She received the B. S. degree in mathematics in 1992 from the Northeastern Normal University. Her main research interests include hybrid systems, intelligent control.

ZHAO Jun Born in 1957. He received the B. S. and M. S. degree in mathematics in 1982 and 1984, respectively, both from Liaoning University. He completed his Ph. D in control theory and applications in 1991 at Northeastern university, China. From February 1992 to 1993 he was a postdoctoral fellow at the same university. Since 1994 he has been with School of Information Science and Engineering, Northeastern University, China, where he is currently a professor. From February 1998 to February 1999, he was a visiting scholar at the Coordinated Laboratory, University of Illinois at Urbana-Champain. His main research interests include hybrid systems, nonlinear systems, geometric control theory, and robust control.

(Continued from page 437)

to Bayes optimal discriminant function [J]. IEEE Trans. Neural Networks, 1990, 1(4): 296 - 298

[3] Kanaya K and Miyaker S. Bayes statistical behavior and valid generalization of pattern classifying neural networks [J]. IEEE Trans. Neural Networks, 1991, 2(4): 471 - 475

[4] Barnard E and Casasent D. A comparison between criterion functions for linear classifiers with an application to neural nets [J]. IEEE Trans. Systems, Man, and Cybernetics, 1989, 19(6): 1030 - 1041

[5] Dimitris and Padons A. New nonleast-squares neural network learning algorithm for hypothesis testing [J]. IEEE Trans. Neural Networks, 1995, 6(3): 596 - 609

[6] Blum J R. Approximation methods which converge with probability one [J]. Annals of Mathematical Statistic, 1954, 25(4): 734 - 744

本文作者简介

王 祁 1944年生. 教授, 博士生导师. 目前主要研究方向是多传感器信息数据融合理论及应用. Email: wangqi@hope.hit.edu.cn

沈国峰 1973年生. 讲师, 哈尔滨工业大学自动化测试与控制系博士研究生. 主要从事多传感器数据融合理论方面的研究工作. Email: Shenguofeng@0451.com

张兆礼 1972年生. 哈尔滨工业大学自动化测试与控制系博士研究生. 主要从事神经网络, 传感器信息处理的研究工作.