# Application of Function Link Net to Recognition of Radar Targets

Huang Deshuang, Mao Erke, Han Yueqiu

Beijing Institute of Technology, Beijing and National Laboratory of Pattern rexognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 100081, P.R.China

## ABSTRACT

This paper studies the mechanism for classification of Feedforward Neural Networks from the geometric viewpoints. It is pointed out that the MLPNs realize hyperplance divisions in the pattern space, and the FLN realize hypercurved divisions. We give a form of Generalized Function Link Nets (GFLN), and discuss the application of a special GFLN to recognition of radar targets, and give several experimental results.

## 1. INTRODUCTION

Multilayer Percetron Network (MLPN) is a kind of feedforward network. For 3—LPN, it is disclosed that the first hidden layer realizes hyperplane divisions in pattern space; the second hidden layer makes logic "AND" computation among outputs of the first hidden layer , and effect space divisions on the divided pattern space; the third hidden layer(output layer) conducts logic "OR" computation among outputs of the second hidden layer, i.e., makes the same sorts of hyperplanes cluster [1].

As is shown in Fig. 1, the $i$th node for the first hidden layer in 3—LPN receives total input (called linear affine function)[2]:

$$z_i = \sum_{j=1}^{n} W_{ij} x_j + \theta_i = W_i^T X \qquad (1)$$

Where $W_i = [W_{i1}, W_{i2}, \cdots, W_{in}, \theta_i]^T$; $X = [x_1, x_2, \cdots, x_n, 1]^T$

Assume that the nonlinear transfer function in the first hidden layer is step—function, then the corresponding output is[2]:

$$y_i = f(z_i) = \begin{cases} 1 & z_i > 0 \\ 0 & z_i = 0 \\ -1 & z_i < 0 \end{cases} \qquad (2)$$

When $z_i = 0$, $W_i$ is called the normal vector. The distance between the hyperplane corresponding to (1) and the origin is defined as $|z_i| / \|W_i\|$.

If we define a hypersphere $S(x_0, r)$ in pattern space $R^n$ [2]:

$$S(x_0, r) = \{x / \|x - x_0\| \leqslant r,\ x, x_0 \in R^n;\ r \in R^+\} \qquad (3)$$

Assume that $x_0$ is the origin in $R^n$, let $z_i = 0$ in (1), a hyperplane $HP_W$ through the origin is obtained, as is shown in Fig. 2.

For any pattern vector $X$, let the hyperplane orthogonal to it be $HP_X$ then the angle $\theta$ between $X$ and $W$ is[2]:

$$\theta = cos^{-1} \frac{W^T X}{\|W\| \|X\|} \quad (0 \leqslant \theta \leqslant 180^\circ) \qquad (4)$$

Obviously, $0 \leqslant \theta \leqslant 90^\circ$, then $z_i > 0$ corresponds to the plus aspect $HP_W^+$ of the $HP_W$; $90^\circ \leqslant \theta \leqslant 180^\circ$, then $z_i < 0$ corresponds to the minus aspect $HP_W^-$ of the $HP_W$. From the above analyses, the trained weights for the first layer in MLPN decide on the space direction for the hyperplane. As the training goes on, the hyperplanes will constantly change the



0-7803-2431-5/95 $4.00 © 1995 IEEE

direction. If the units in the first hidden layer are large enough, the linear regions divided will become unlimited.

These hyperplanes divisions are no utility to those irregular pattern distributions which need many many hidden nodes and much training time to gain stationary hyperplanes. Sometimes it is very difficult to separate out certain pattern from the pattern space, i. e., the trained network doesn't converge. So we need look for other hyperspace divisions such as hypersphere、hyperellipsoid、hyperparaboloid, etc. As is shown in Fig. 3, these hypercurved divisions are suited to those irregular pattern distributions which are usual situation. So the studies of the hypercurved division networks are more significant.

## 2. GENERALIZED FLNs

### 2. 1. Hypercurved Divisions and FLNs

In order to attain those hypercurved divisions, the method is to produce all kinds of nonlinear input terms in the input units of the networks. They form the nonlinear affine functions for inputs to the first hidden layer, which are tranformed by the I/O function for the units in the first hidden layer to produce all kinds of hypercurved divisions.

Assume that the nonlinear affine function receivbed by the $i$th unit in the first hidden layer is:

$$z_i = \sum_{k_1=1}^{d} W_{ik_1}^{(1)} x_{k_1} + \sum_{k_1=1}^{d}\sum_{k_2=1}^{k_1} W_{ik_1}^{(2)} x_{k_1} x_{k_2} + \cdots + \sum_{k_1=1}^{d}\sum_{k_2=1}^{k_1}\cdots\sum_{k_p=1}^{k_{p-1}} W_{ik_1}^{(p)} x_{k_1} x_{k_2} \cdots x_{k_p} + \sum_{k_1=1}^{d} W_{ik_1}^{(p+1)} f_1(x_{k_1}) + \sum_{k_1=1}^{d}\sum_{k_2=1}^{k_1} W_{ik_1}^{(p+2)} f_1(x_{k_1}) f_2(x_{k_2}) + \cdots + \sum_{k_1=1}^{d}\sum_{k_2=1}^{k_1}\cdots\sum_{k_q=1}^{k_{q-1}} W_{ik_1}^{(p+q)} f_1(x_{k_1}) f_2(x_{k_2}) \cdots f_q(x_{k_p}) + \theta_i \qquad (5)$$

Where $p$、$q$ are the high-order number; $f_1(.)$, $f_2(.)$, $\cdots$, $f_q(.)$ are the nonlinear transform functions. We call the networks composing of them as Generalized Function Link Nets(GFLN).

Here, we select $q = 0$, $f_1(.) = f_2(.) = \cdots = f_q(.) = 1$ to produce the special FLN as shown in Fig. 4 to conduct the experiment about the recognition of radar targets.

### 2. 2. RLS—BP Algorithm

We use the fast Recursive Least Square Backpropagation learning algorithm(RLS—BP) to train the weights of the FLN. As is shown in Fig. 4, We define the weighted erroneous cost function[2]:

$$J(n) = \frac{1}{2}\sum_{t=1}^{n}\lambda^{n-t}\sum_{k=1}^{M}e_k^2(t) = \frac{1}{2}\sum_{t=1}^{n}\lambda^{n-t}\sum_{k=1}^{M}[d_k(t) - y_k(t)]^2 \qquad (6)$$

$$y_k(t) = \sum_{i=1}^{N} W_k^{(i)}(n) x_1(t) x_2(t) \cdots x_i(t) \qquad (7)$$

Where $e_k(t)$, $d_k(t)$, and $y_k(t)$ are the error, the desired and actual output signal of the kth node in the output layer, $\lambda$ is the forgotten factor used as weighted square errors.

By taking the partial derivative of $J(n)$ W.r.t. $W_k(n)$ and setting $\frac{\partial J(n)}{\partial W_k(n)} = 0$. We can obtain recursive least square updating formulas of the link weight vectors as follows[2][3][4][5]:

$$K(n) = \frac{P(n-1)X(n)}{\lambda + X^T(n)P(n-1)X(n)} \qquad (8)$$

$$P(n) = \frac{1}{\lambda}[P(n-1) - K(n)X^T(n)P(n-1)] \qquad (9)$$

$$W_k(n) = W_k(n-1) + K(n)[d_k(n) - X^T(n)W_k(n-1)] (1 \leqslant k \leqslant M) \qquad (10)$$

$$J(n) = \lambda J(n-1) + \frac{1}{2}\sum_{k=1}^{M}[d_k(t) - X^T(n)W_k(n-1)]^2 \qquad (11)$$

Obviously, we can see that the RLS—BP algorithm obtains the accurate solutions of the link weight vectors at each time, and the updating doesn't need any external parameters, it is superior to the

LMS—BP algorithm.

## 3. EXPERIMENTAL RESULTS

We use one—dimensional cross image datum of the five kinds of airplanes (Plane1, Plane2, Plane3, Plane4 and Plane5) as the experimental datum to be classified. Assume the orientation of the airplanes' head to be 0°, the one—dimensional cross images are imaged from the model datum in the range of 0° ~ 100° (the dimension for each vector is 30) where 30 training samples are selected every 3° (Plane5 every 0.5°). The remainder testing samples for Plane1、Plane2、Plane3、Plane4 and Plane5 are 138、150、113、162 and 18, respectively. Fig. 5 shows the testing samples distribution for the Plane2.

We use the 30 × 5 training samples to train the link weights in the network so that the erroneous weighted cost functions in the output layer descend to $10_3$. After the link weights in the network are obtained, we use the 14 testing samples to get the rate of recognition: 96.4%, 98.7%, 97.6%, 99.7%, 98.5% for the five kinds of planes. Fig. 6 gives the confidence curve of recognition for Plane1.

## 4. REFERENCE

〔1〕Y. H. Pao, Adaptive Pattern Recognition and Neural Networks, Addison—Wesley Publishing Company Inc., 1989.

〔2〕Huang Deshuang, "the Study of Recognition Technique Based on One—dimensional Image of Radar Targets", Ph.D dissertation, Xidian University, 1992.12.

〔3〕M. R. A. Sadjadi and R. J. Liou, "Fast Learning Process of Multilayer Neural Networks Using Recursive Least Squares Method", IEEE Trans. Vol. SP—40, №. 2, pp. 446~450, Feb. 1992.

〔4〕S. Kollias and D. Anastassiou, "Adaptive Training of Multilayer Neural Networks Using a Least Squares Estimation Technique", IJCNN, Vol. I, pp. 383~390, 1989.

〔5〕S. Kollias and D. Anastassiou, "An Adaptive Least Squares Algorithm for the Efficient Training of Artifical Neural Networks", IEEE Trans. Vol. CAS—36, №. 8, pp. 1092~1101, Aug. 1989.

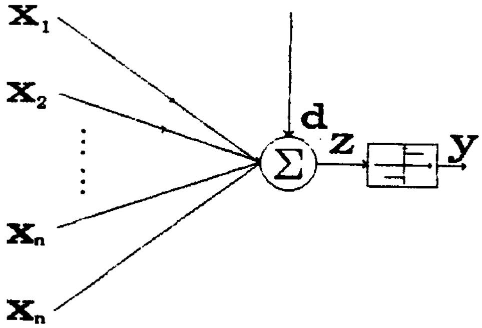


Fig. 1 The $i$ th Node in the First Hidden Layer

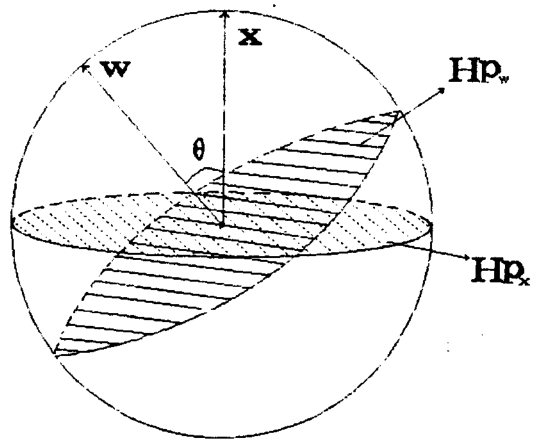


Fig. 2 Hypersphere and Hyperplane

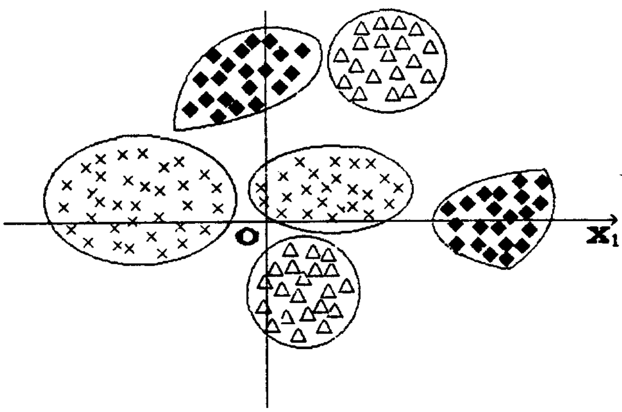


Fig. 3 Two Dimensional Curved Division

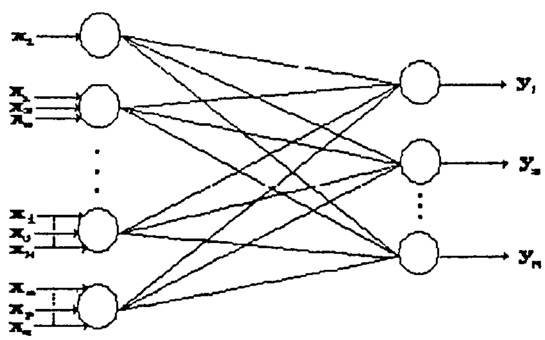

Fig. 4 Generalized Function Link Net

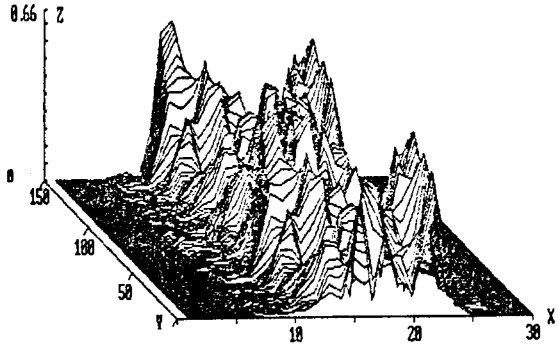


Fig. 5 Testing Sample Distribution for Plane2

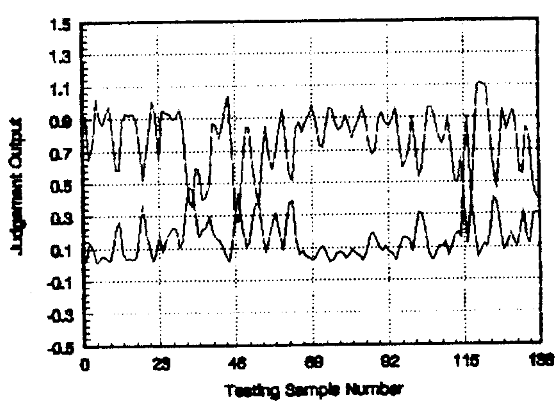


Fig. 6 Confidence Curve for Plane1