An Algorithm for Multistage Artificial Neural Network

B.M.Singhal
Department of Computer Application, ITM–University,
Gwalior, India.

ABSTRACT
We may presume the neural networks are simplified models of the biological neurons system. The Artificial Neural Network (ANN) is an information processing system which is inspired by brain learning system. It is assumed that brain is composed of a large number of highly interconnected processing elements working in groups to solve specific problems. Various networks and algorithms have been proposed to enhance the machine learning process and to achieve some thing new. In this paper we have proposed a moderate algorithm for multistage artificial neural network.

(Key Words- Algorithm, Neural Networks)

1. INTRODUCTION
Artificial Neural Network is a precise tool to represent a technology that is in the disciplines of neurosciences, mathematics, physics, computer science and engineering. The advent of neural networks saw the area of artificial intelligence grows rapidly as a **tool** for many undecided events. These studies have huge possibilities to design a competent Artificial Neural Network resembling and enhancing brain efficiencies. To achieve the goal many algorithm in this field have been proposed by various authors. Here we propose a moderate algorithm for multistage Artificial Neural Network.

Let $x_1, x_2, ..., x_r$ is a set of inputs and the corresponding weights are

$$w_i ; i = 1, ..., r;$$

Then the strength of x_i will be

$w_i \; x_i$; i= 1,2,…r and let we have m_i neurons of type x_i. Then the combined probabilistic strength of type x_i will be

$$\frac{(w_i x_i)^{m_i}}{m_i!}$$

If we choose an activation function

$$f\left(\sum_{i=1}^{n} m_i \right)$$

The system output y for first stage of network for the weight vector

w = ($w_1, ..., w_n$)

may be taken as ($0 \le w_i \le 1$),

$$f\left(\sum_{i=1}^{r} m_i\right) \prod_{i=1}^{r} \frac{(w_i x_i)^{m_i}}{m_i!}$$

Similarly for second stage output

$$y_1, ..., y_s$$

we would have

$$g\left(\sum_{j=1}^{s} n_j\right) \prod_{i=1}^{s} \frac{\left(w_j{}' \, y_j\right)^{n_j}}{n_j!}$$

Then the total output Z with an unified activation function

$$h\left(\sum_{i=1}^{r} m_i + \sum_{j=1}^{s} n_j\right)$$

may be taken as (Because in case of indefinite large numbers the probabilistic models are best suited for infinity)

$$\mathbf{Z} = \sum_{m_1,...,m_r=0}^{\infty} \sum_{n_1,...,n_j=0}^{\infty} h\left(\sum_{i=1}^{r} m_i + \sum_{j=1}^{s} n_j\right) f\left(\sum_{i=1}^{r} m_i\right) g\left(\sum_{j=1}^{s} n_j\right) \prod_{i=1}^{r} \frac{(w_i x_i)^{m_i}}{m_i!} \prod_{i=1}^{s} \frac{\left(w_j{}' \, y_j\right)^{n_j}}{n_j!}$$

$$= \sum_{m,n=0}^{\infty} \text{h(m+n) f(m) g(n)} \frac{\left(\sum_{i=1}^{r} w_i x_i\right)^m}{m!} \frac{\left(\sum_{j=1}^{s} w_j{}' \, y_j\right)^n}{n!} \quad .... (1)$$

by using identity [9],

$$\sum_{m_1,m_2,\ldots m_r=0}^{\infty} \sum_{n,n,\ldots n_s=0}^{\infty} f\left(\sum_{i=1}^{r} m_i + \sum_{j=1}^{s} n_j\right)$$

$$g\left(\sum_{i=1}^{r} m_i\right) h\left(\sum_{j=1}^{s} n_j\right) \prod_{i=1}^{r} \frac{x_i^{m_{ii}}}{m_i!} \prod_{j=1}^{s} \frac{y_j^{n_j}}{n_j!}$$

$$= \sum_{m,n=0}^{\infty} f(m+n)\ g(m)\ h(n)$$

$$\frac{\left(\sum_{i=1}^{r} x_i\right)^m}{m!} \frac{\left(\sum_{j=1}^{s} y_j\right)^n}{n!}.$$

In case of symmetrical layers h(m+n) may be taken as constant, say k. Also a group of neurons are similar in structure, the f(m) and g(n) in (1) may be taken as modular parameters λ^m and μ^n say. So we have

$$Z = \sum_{m,n=0}^{\infty} \frac{\left(\lambda \sum_{i=1}^{r} w_i x_i\right)^m}{m!} \frac{\left(\mu \sum_{j=1}^{s} w_j' y_j\right)^n}{n!}$$

$$= k\, e^{\lambda \sum_{i=1}^{r} w_i x_i + \mu \sum_{j=1}^{s} w_j' y_j}$$

Also if the network configures with $\lambda = \mu = 1$ then

$$Z = k\, e^{\sum_{i=1}^{r} w_i x_i + \sum_{j=1}^{s} w_j' y_j}$$

$$= k\, e^{WX + W'Y} \qquad \ldots (2)$$

where $WX = \sum_{i=1}^{r} w_i x_i$ and

$$W'Y = \sum_{j=1}^{s} w_j' y_j$$

Or

$$Z \ \alpha \ e^{WX + W'Y} \qquad \ldots \qquad (3)$$

i.e. the second stage output Z will be proportional to exponential weighted sum of inputs and first stage outputs.

Choosing Y = O, We get

$$Z \propto e^{WX}$$

A result obtained by the author [10].

REFERENCES

[1]. J.Moody and C. Darken, "Fast learning in Networks of locally- tuned Processing units, Neural Computation , 1:281-294, 1989.

[2]. N.B.Karayiannis and G.W. Mi, "Growing radial basis Neural Networks: merging supervised and unsupervised learning with network growth techniques " IEEE Trans. On Neural Networks.

[3]. D.Dasgupta and S. Forest, "Artificial Immune System in Industrial Applications" Proc. Of the IPMM'99, 1999.

[4]. P.Hajela and J.S.Yoo, "Immune Network Modeling in design Optimization ".In new Ideas in Optimization,(Eds) D Corne, M.Dorigo & F. Glover, McGraw Hill, London, pp. 203-215, 1999.

[5]. L.N.De Castro and F.J.Von Zuben, "An Evolutionary Immune Network for data clustering ", Proc. Of the IEEE Brazelian Symposium on Neural Networks, pp. 84-89, 2000b.

[6]. D.S.Broomhead and D.Lowe, "Multivariate functional Interpolation and adaptive Networks", Complex Systems, 2:321-355, 1988.

[7]. M.J.D. Powell, "Radial Basis Functions for multivariable Interpolation", A reviw in IMA Conference, Algorithm for Appr. Of Functions and Data, J.C. Mason & M.G. Cox (eds.), Oxford , U.K.: Oxford Univ. Press, 143-167, 1987.

[8]. C.A. Michelli, "Interpolation of Scattered Data: Distance Matrices and conditionally Positive definite Functions", Const.Approx.,2: 11-22, 1986.

[9]. B.M. Singhal and B.M. Agrawal, "On Multiple Integrals Involving Hypergeometric Functions of two Variables", Jnanabha Sect. A. Vol. 4, July 1974.

[10]. B.M. Singhal, "A proposed Algorithm for Multivariate Artificial Neural Network", accepted for publication, IEEE Conference Feb.2010 Indian Institute Of Science, Banglore , India. *2010 International Journal of Computer Applications (0975 – 8887) Volume 1 – No. 3 pp 56-57*