An Algorithm for Radial Basis Function Neural Networks

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

ABSTRACT
A radial basis function (RBF) neural network depends mainly upon an adequate choice of the number and positions of its basis function centers. In this paper we have proposed an algorithm for RBF neural network and the results may be reduced for artificial neural networks as particular cases.

Categories and Subject Descriptors
I.2.0 Artificial Intelligence

General Terms
Neural Networks, Radial Basis Function, Algorithm

Key Words
Radial Basis Function, Neural Networks, Algorithm

1. INTRODUCTION
The radial basis function neural networks are powerful function approximators for multivariate nonlinear continuous mappings. They have a simple architecture and the learning algorithm corresponds to the solution of a linear regression problem. The RBF network behavior strongly depends upon the number and the position of the basis functions at the hidden layer. Traditional method of determining the centers are randomly choose input vectors from the training data set or to obtain vectors from unsupervised clustering algorithms applied to input data or vectors obtain through a supervised learning scheme. In this paper we have proposed a new algorithm for RBF neural network and the results may be reduced for artificial neural networks as particular cases.

Let $x_1, x_2, ..., x_r$ is a set of points and $c_i; i = 1,2,...,r$ be the set of centers, then

$$\| x_i - c_i \| \quad ; \quad i = 1,2,\ldots,r$$

be the radial vectors in Euclidean sense. If the corresponding weights are

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

the radial strength of x_i will be

$$w_i \| x_i - c_i \| \quad ; \quad i = 1,2,\ldots,r$$

Then the combined probabilistic strength of type x_i will be

$$\frac{\{w_i \| x_i - c_i \|\}^{m_i}}{m_i!}$$

for m external input lines for n units of RBF neural network.

If we choose an activation function

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

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

$$\text{w} = (w_1, w_2, ..., w_r); \ (0 \le w_i \le 1)$$

may be taken as

$$f\left(\sum_{i=1}^{r} m_i\right) \quad \text{X}$$

$$\text{X} \coprod_{i=1}^{r} \frac{\{w_i \parallel x_i - c_i \parallel\}^{m_i}}{m_i!}$$

Similarly for second stage output

$$y_1, \ldots, y_s$$

We would have

$$g\left(\sum_{j=1}^{s} n_j\right) \coprod_{j=1}^{s} \frac{\left\{w_j' \parallel y_j - c_j \parallel\right\}^{n_j}}{n_j!}$$

Then for total output Z of multilayer RBF neural network, we may adapt an unified activation function

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

and therefore (Because in case of indefinite large numbers the probabilistic models are best suited for infinity).

$$\mathbf{Z} = \sum_{m_1,\ldots,m_r=0}^{\infty} \sum_{n_1,\ldots,n_j=0}^{\infty} h\left(\sum_{i=1}^{r} m_i + \sum_{j=1}^{s} n_j\right) f\left(\sum_{i=1}^{r} m_i\right) g\left(\sum_{j=1}^{s} n_j\right)$$

$$\coprod_{i=1}^{r} \frac{\{w_i \parallel x_i - c_i \parallel\}^{m_i}}{m_i!} \coprod_{j=1}^{s} \frac{\left\{w_j' \parallel y_j - c_j \parallel\right\}^{n_j}}{n_j!} .$$

Or

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

$$\frac{\left\{\sum_{i=1}^{r} w_i \parallel x_i - c_i \parallel\right\}^{m}}{m!} \frac{\left\{\sum_{j=1}^{s} w_j' \parallel y_j - c_j \parallel\right\}^{n}}{n!} \quad \ldots \; (1)$$

by using identity [9]

$$\sum_{m_1,m_2,\ldots,m_r=0}^{\infty} \sum_{n,n,\ldots,n_s=0}^{\infty} f\left(\sum_{i=1}^{r} m_i + \sum_{j=1}^{s} n_j\right) g\left(\sum_{i=1}^{r} m_i\right) h\left(\sum_{j=1}^{s} n_j\right) \prod_{i=1}^{r} \frac{x_i^{m_{ii}}}{m_i!} \prod_{j=1}^{s} \frac{y_j^{n_j}}{n_j!}$$

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

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

$$\mathbf{Z} = \mathbf{k} \sum_{m,n=0}^{\infty} \frac{\left\{\lambda \sum_{i=1}^{r} w_i \parallel x_i - c_i \parallel\right\}^{m}}{m!} \frac{\left\{\mu \sum_{j=1}^{s} w_j' \parallel y_j - c_j \parallel\right\}^{n}}{n!}$$

$$= \mathbf{k} \; e^{\lambda \sum_{i=1}^{r} w_i \|x_i - c_i\| + \mu \sum_{j=1}^{s} w_j' \|y_j - c_j\|}$$

Or

Z α

$$e^{\lambda \sum_{i=1}^{r} w_i \|x_i - c_i\| + \mu \sum_{j=1}^{s} w'_j \|y_j - c_j\|} \quad \textbf{... (2)}$$

i.e. the RBF neural network output is directly proportional to exponentially weighted sum of inputs and middle stage outcomes taken radially along with constraints λ and μ which may be suitably defined as per network configuration.

Particularly when we choose

$\mu = 0$ in (2), we have (Z$\rightarrow$ Y)

$$\mathbf{Y} \ \alpha \ e^{\lambda \sum_{i=1}^{r} w_i \|x_i - c_i\|} \quad \textbf{...} \ (3)$$

Or for $c_i = 0$

$$\mathbf{Y} \ \alpha \ e^{\lambda \sum_{i=1}^{r} w_i x_i}$$

$$\alpha \ e^{\lambda WX} \quad \ldots (4)$$

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

a result proposed by author [10] for artificial neural network.

2. CONCLUSION

In this paper we have proposed an algorithm for radial basis neural networks for the lack of availability of proper algorithm in the discipline. The further work is being carried out and may bring some fruitful results in near future.

3. 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.