Implementation of RBF type networks by MLP networks

Bogdan M. Wilamowski and Richard C. Jaeger
Department of Electrical Engineering, Auburn University
Auburn, AL 36849-5201
wilam@uwyo.edu, jaeger@eng.auburn.edu

Abstract

Simple transformations of input patterns onto a hypersphere in augmented space are presented and experimentally verified. This approach allows the MLP networks to perform the same functions as RBF networks. Two transformations are described. In the first one, dimensionality is increased by one, and only one additional variable has to be computed. In the second approach dimensionality is doubled, but this leads to a simple implementation of the transformation with sigmoidal type neurons. The modified network has a relatively simple structure, and it is able to perform very complicated nonlinear operations. The power of this network is demonstrated with examples including the two spiral problem.

1. Introduction

Artificial neural networks use a biological neuron as a prototype of the processing element. In biological neurons the input is an integrated and weighted sum of the incoming pulses and the output is a pulse train with a frequency that is dependent upon the input excitation. The transfer function of such a biological neuron is approximated by a squashed step function, usually a sigmoidal function. In the majority of artificial neural network implementations, this pulse nature of the biological neuron is replaced by simple static sigmoidal type characteristics. This simple artificial neuron model is used in Multi Layer Perceptron (MLP) networks. All neurons in MLP networks compute the weighted sum of the incoming signals $net = \mathbf{w}^T\mathbf{x}$, and then the sigmoidal type of activation function is applied to the *net* value to compute the neuron output

$$o = f(net) \approx \frac{1}{1+\exp(-a\,net)} \tag{1}$$

In the last several years it was found that another type of network, known as the Radial Basis Function or RBF network is, in many cases, more convenient and more powerful [1][2][3]. A simple RBF "neural" network architecture is shown in Fig. 1(a). Each input "neuron" of the RBF network computes a distance between stored pattern **t** and applied pattern **x** in multidimensional input space, $net = \|\mathbf{x}-\mathbf{t}\| = [(\mathbf{x}-\mathbf{t})^T(\mathbf{x}-\mathbf{t})]^{1/2}$, and then a Gaussian type activation function is applied

$$o \approx \exp\left(-a\,net^2\right) = \exp\left[-a\,(\mathbf{x}-\mathbf{t})^T(\mathbf{x}-\mathbf{t})\right] \tag{2}$$

In order to sustain processing ability it is not necessary that neurons will have the exact activation functions as given by (1) or (2). It is sufficient that MLP neurons have something like a squashed step function, and that RBF "neurons" are able to create a "bump" in multidimensional input space.

The RBF networks are very powerful and easy to use; however they are usually criticized for not reassembling biological neurons. On the other hand, biological neurons are not able to compute a distance in multidimensional input space. They are able only to linearly separate this space. The purpose of this presentation is to show that with slight modifications the MLP can replace the RBF networks.

In the past, many attempts have been made in this direction. In Winner Take All (WTA) [4] and in Self-Organizing Mapping (SOM) [5] architectures, developed by Kohonen, input patterns and weights in the first layer are normalized, so they are projected on a hypersphere with unit radius. The sigmoidal neuron dividing a hyperspace by hyperplane can separate a cluster on this hyperspherey in a manner similar to the way a fraction of an apple's surface can be separated by a single straight cut with a knife. The Kohonen approach has one a significant drawback. By normalization of the input pattern, the information about its magnitude is lost.

A similar approach was used in counterpropagation networks developed by Hecht-Nilsen [6]. For binary digital patterns, normalization is not necessary so no information is lost, and the counterpropagation network, or its simplified version as described by Zurada [7], are very simple and powerful tools for binary pattern clustering

$$w_i = \begin{cases} t_i & (i = 1,2,\ldots,N+1) \\ \mathbf{t}^T\mathbf{s} & (i = N+2) \end{cases} \tag{7}$$

then the neuron separates all patterns around center of the cluster t *within the radius* ξ

Proof: The separation hyperplane of the neuron is given by a linear equation with weights as coefficients

$$w_1y_1 + w_2y_2 + \cdots + w_{n+1}y_{n+1} + w_{n+2} = 0 \tag{8}$$

The hyperplane is normal to the **t** vector and passes through point **s**. From point and normal vector formula, the equation for a separation hyperplane can be written as

$$\mathbf{t}^T(\mathbf{y}-\mathbf{s}) = 0 \quad \text{or} \quad \mathbf{t}^T\mathbf{y} - \mathbf{t}^T\mathbf{s} = t_1y_1 + t_2y_2 + \cdots + t_{n+1}y_{n+1} - \mathbf{t}^T\mathbf{s} = 0 \tag{9}$$

Equation (7) can be obtained from comparison of (8) and (9).

Assuming that center and side patterns are given, then weights can be found by transforming these known patterns from initial input space to the transformed space using (4) and choosing r such that $r^2 > \mathbf{x}^T\mathbf{x}$.. Then using (7), actual neuron weights can be found. Note, the that first N weights are equal to the coordinates of the center of a cluster in the initial input space. Using this transformation, the RBF network can be replaced by an MLP network as shown in Fig. 1. The method can be illustrated with a few simple examples. Although the approach is general, only examples with two inputs will be used in order to be able to present results graphically.

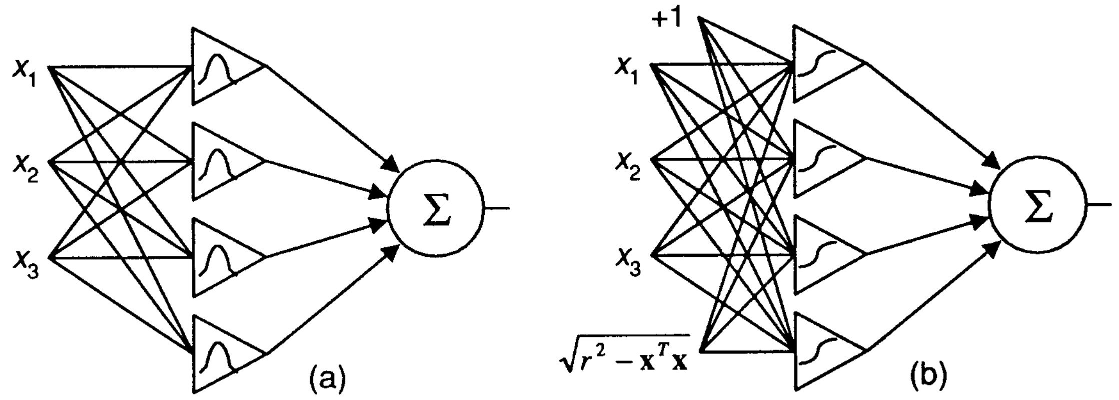


Fig. 1. Networks diagram (a) RBF network, (b) MLP equivalent.

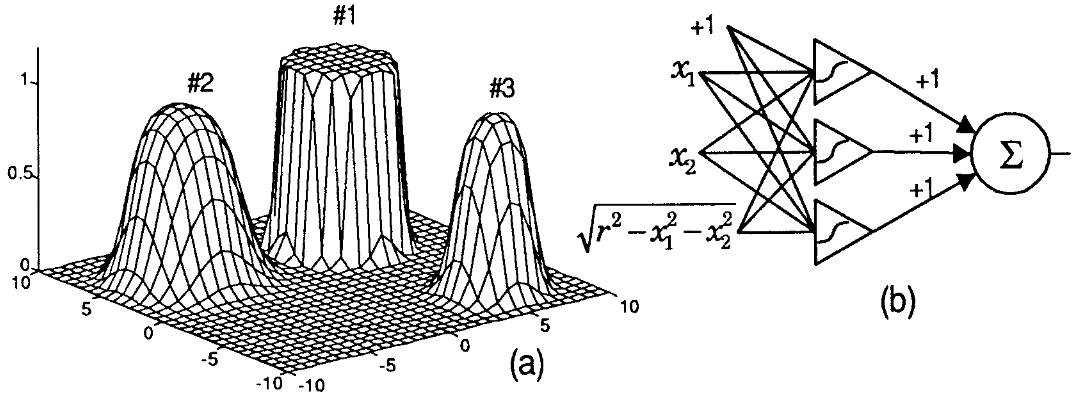


Fig. 2. Separation of three clusters (a) input-output mapping, (b) network diagram

Example 1. Input patterns belong to two categories: the first category has three clusters with a given location and size; patterns of another cluster are equally distributed in the remaining area. Using (4) and (7) the weights can be analytically computed. The resultant input-output mapping, as presented in Fig. 2(a), clearly shows that sigmoidal type neurons behave like the RBF "neurons" with the proposed transformation. Various cluster radiuses (ξ = 3, 3, 2) and neuron gains (a = 1, 0.125, 0.25) were used to demonstrate the network performance.

and classification in the minimum Hamming distance sense. Analog patterns in counterpropagation networks face the same difficulty as in Kohonen networks.

Attempts were also made to introduce extra dimension to the input so the information about input vector magnitude will not be lost. Wilensky and Manukian proposed and patented the Projection Neural Network [8] where two different transformations from an N-dimensional input to N+1 dimensional transformed input space were introduced. Both transformations use rational, square and square root functions and all dimensions have to be recalculated. Wilamowski [9] and Ota and Wilamowski [10] presented different trigonometric transformations for the same purpose. With additional transformations a simple sigmoidal neuron connected to an N+1 dimensional input linearly separates patterns which are not separable in the initial N dimensional input space. Linear separation in augmented space corresponds to separations by circles, spheres or hyperspheres. Therefore, a sigmoidal type neuron with transformed input pattern can act as a RBF "neuron".

A slightly different approach is used in Casasent Networks [11][12]. The Casasent networks also increase the input dimensionally by introducing an additional variable x_{n+1} equal to the square of the magnitude of the input pattern

$$x_{n+1} = \mathbf{x}^T\mathbf{x} = x_1^2 + x_2^2 + \cdots + x_n^2 \tag{3}$$

Sarajedini and Hecht-Nilsen presented excellent theoretical proof that a RBF network with a single hidden layer can be replaced by a Casasent network with two hidden layers[12]. In Casasent networks, the input pattern is projected on a cylindrical hyperparabola, and a cluster on the top of this cylindrical parabola can be easy separated by a linear cut of sigmoidal neuron.

In this paper, a transformation onto a hypersphere in augmented N+1 space is proposed. Contrary to previous approaches with transformation to hypersphere [4][8][9][10], only one additional input variable in augmented space has to be computed. The approach is similar to the Casasent network, but the input space is transformed to a hypershpere, not to a cylindrical hyperparabola. This approach is more general. It allows for future increase of input space dimensionally which can be useful when implementing the transformations in hardware. This generalized approach leads to a MLP implementation of both transformation and RBF networks. In previous approaches [4][5][8][9][10][11][12] input data normalization or transformation has to be done off-line.

2. Input data transformation to N+1 dimensional space

The first theorem presents the transformation from an initial N dimensional input to a hypersphere in N+1 dimensions, as given by equation (4). The second theorem gives a formula for neuron weights.

Theorem 1: *Let input pattern* x ∈ R^N *and transformed pattern* y ∈ R^{N+1} *and*

$$y_i = \begin{cases} x_i & (i = 1,2,\ldots,N) \\ \sqrt{r^2 - \mathbf{x}^T\mathbf{x}} & (i = N+1) \end{cases} \tag{4}$$

Then transformed pattern y *lays on a hypersphere with radius r*

Proof:

$$\mathbf{y}^T\mathbf{y} = y_1^2 + y_2^2 + \cdots + y_n^2 + y_{N+1}^2 = x_1^2 + x_2^2 + \cdots + x_N^2 + \left(\sqrt{r^2 - x_1^2 + x_2^2 + \cdots + x_N^2}\right)^2 = r^2 \tag{5}$$

or the same with shorter notation:

$$\mathbf{y}^T\mathbf{y} = \mathbf{x}^T\mathbf{x} + \left(\sqrt{r^2 - \mathbf{x}^T\mathbf{x}}\right)^2 = r^2 = const \tag{6}$$

All transformed patterns as defined by (5) satisfy condition $\mathbf{y}^T\mathbf{y} = r^2 = const$ and therefore they are located on a hypersphere with a radius r.

Theorem 2: *Let* (y,t,s) ∈ R^{N+1} *where* y *is a transformed input pattern,* t *is a pattern corresponding to the center of a cluster, and* s *is any pattern located on a side (edge) of the cluster. The radius of this cluster* ξ *is equal to the distance between* t *and* s, $\xi = \|t-s\|$. *The Neuron has* N+1 *inputs connected to transformed input signals* y_i (i = 1, 2, ..., N+1) *and the bias weight* w_{n+2}. *If weights are given by*

Example 2 consists of two neurons in which the outputs are subtracted and a crescent shape is created as Fig. 3 shows.

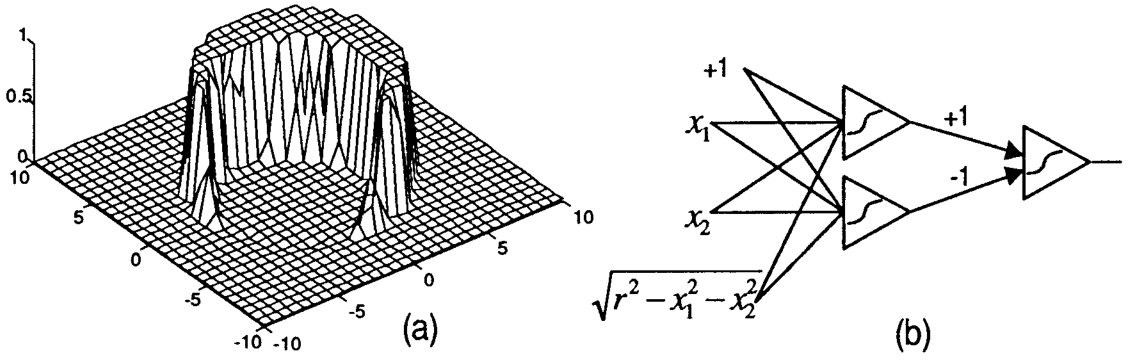


Fig. 3. Network with two neurons capable of separating crescent shape of patterns (a) input-output mapping, (b) network diagram

3. Transformation to higher dimensions

Although the transformation presented in the previous section requires only computation for one additional input, it is rather computationally intensive and therefore has to be done off line. In order to also use MLP for the required transformations, let us consider the following transformation of input patterns

$$y_i = \begin{cases} x_i & (i = 1,2,\cdots,N) \\ \sqrt{r^2 - x_{i-N}^2} & (i = N+1, N+2, \cdots, 2N) \end{cases} \tag{10}$$

This transformation also projects input patterns onto a hypersphere but $\mathbf{y} \in \mathrm{R}^{2N}$. Proof that transformation (10) is correct is similar to the proof of the **Theorem 1**. It can be easily shown that

$$\sum_{i=1}^{2N} y_i^2 = r^2 = const \tag{11}$$

and therefore **y** lays on a 2N dimensional hypersphere. Half of inputs need not to be changed, and the other half require the following function to be implemented in hardware

$$y_i \approx r \tanh\left[\pi\left(1 - \frac{x_{i-N}}{r}\right)\right] \qquad (i = N+1, N+2, \cdots, 2N) \tag{12}$$

Figure 4 illustrates the approximation. Note, that in order to use this approximation the input variables must be positive and preferably within a range of 0.1 to 0.8 where the second derivative has a large magnitude. This is not a serious limitation because this range can be easily set by properly scaling the weights (and threshold).

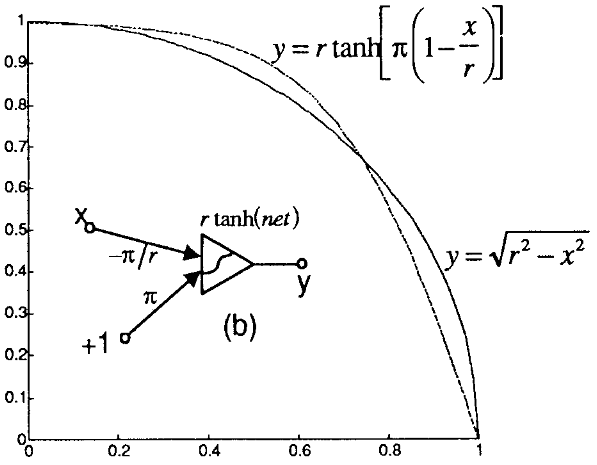
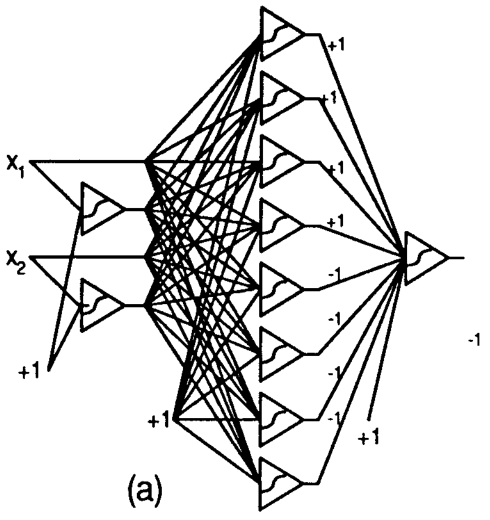


Fig 4. Function approximation: both function comparison and network to implement the approximation are shown.

The two spiral problem is considered to be one of the most difficult to handle by MLP networks. By an extension of the crescent shape synthesis shown in Fig. 3, the two spiral problem can be easily solved. With two neurons in the transformation layer and eight neurons in the hidden layer, the network is capable of separating a spiral. The network and the input-output mapping are shown in Fig. 5.



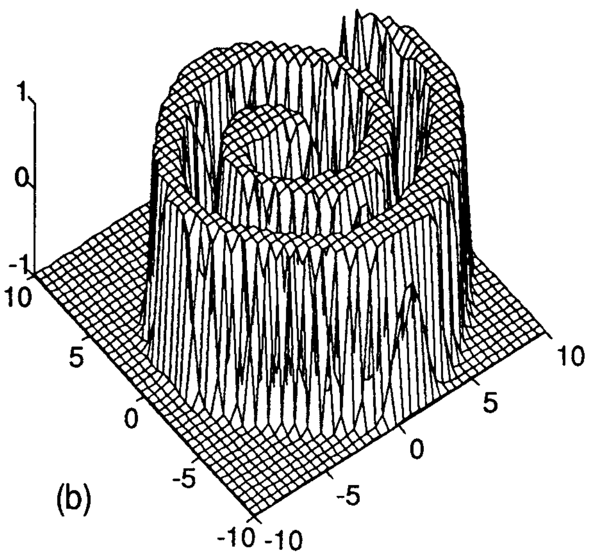
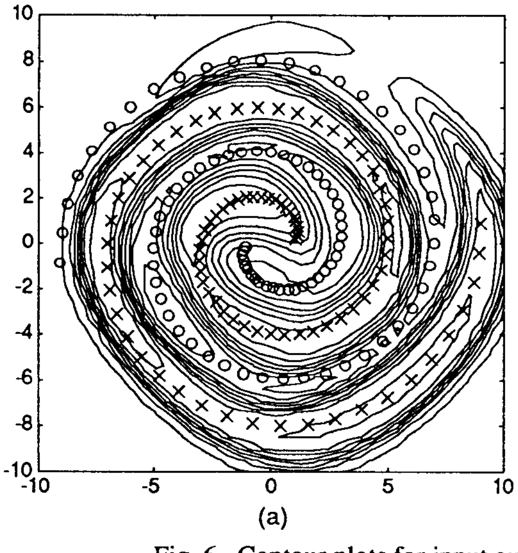


Fig. 5. Spiral problem solved with sigmoidal type neurons (a) input-output mapping, (b) network diagram

Weights for the network shown in Fig. 5 were computed analytically using equations (4)(7), and (12) for ideal hyperspheres. The presented results in the figure are of the actual network. Since the transformation obtained by sigmoidal neurons is only an approximation to the actual one, the obtained results are not perfect. This can be easily corrected by final network tuning. Using any supervised training method such as the Error Back Propagation [14] or the Levenberg-Marquardt method [15]. Note, that in this particular problem weights in the first and the last layer need not to be changed, and also the required outputs from the hidden layer are known. Therefore the tuning process can be simplified to the one-layer training problem, and the very efficient modified regression algorithm as described in [13] can used. Fig 6 shows comparison of the input-output mapping for the network before and after tuning.



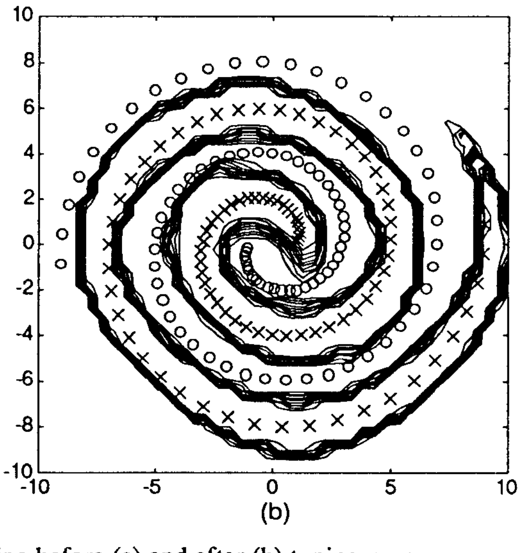


Fig. 6. Contour plots for input output mapping before (a) and after (b) tuning process

4. Conclusion

Two transformations to a hypersphere were presented. The first transformation to $N+1$ dimensions is suitable for off-line training where only one additional input for augmented space has to be computed. Another transformation to $2N$ dimensions is suitable for hardware implementation. Both approaches allow for replacing the RBF networks with MLP networks. The modified MLP network has a relatively simple structure, and it is able to perform very complicated nonlinear operations. The power of this network was demonstrated with several examples including the two spiral problem.

The network weights can be found by any supervised training algorithm, but because of the highly nonlinear character of the network the proper solution can be guaranteed only when initial weights are close to the required one. Knowing cluster locations, the network weights can be found analytically using (4)(7), (10)(7), or (12)(7). Therefore, the key issue is to find the size and cluster location. Unsupervised clustering methods such as SOM [5], ART[16] with its derivatives, and mountain clustering[17] can be used to find these clusters. Unfortunately the unsupervised algorithms can perform well only when all clusters are clearly defined and separated. Even when the information about desired output patterns is known, it is not used in the unsupervised clustering. Therefore, the supervised clustering method as described in [18] is an attractive alternative and usually leads to better results.

References

[1] J. Moody and C. J. Darken, "Fast learning networks of locally-tuned processing units," *Neural Computation* vol. **1**, no. 2, pp. 281-294, 1989.

[2] E. Hartman, J. D. Keeler, and J. M. Kowalski, "Layered neural networks with Gaussian hidden units as universal approximations." *Neural Computation*, vol. **2**, no. 2, pp. 210-215, 1990.

[3] F. Girosi, T. Poggio, and B. Caprile, "Extension of a theory of networks for approximation and learning: outliners and negative examples." *Advances in Neural Information Processing Systems* 3, eds. R. P Lippmann, J. E. Moody, and D. S. Touretzky, pp. 750-756, San Mateo, California: Morgan Kaufmann 1991.

[4] T. Kohonen, "Adaptive, associative, and self-organizing functions in neural computing," *Applied Optics*, vol. **26**, pp. 4910-4918, 1987.

[5] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. **78**, no. 9, 1990.

[6] R. Hecht-Nilsen, *Neurocomputing*, pp. 147-153, Addison-Weseley, Reading, MA, 1991

[7] J. Zurada, *Introduction to Artificial Neural Systems*, pp. 410-414, 447-449, West, St. Paul, MN, 1992

[8] G. Wilensky and N. Manukian, The Projection Neural Networks, *International Joint Conference on Neural Networks*, **II**, 358-367, 1992.

[9] B. Wilamowski, Fast Algorithms for Neural Network Design and Training, *Proc. of the Third Intelligent Information Systems Workshop*, Wigry, Poland, pp. 443-458, June 6-10, 1994.

[10] Y. Ota and B. Wilamowski, "Input data transformation for better pattern classification with less neurons," *Proc. of World Congress on Neural Networks*, San Diego, California, vol. 3, pp 667-672, 1994.

[11] D. Casasent and E. Barnard, Adaptive clustering neural net for picewise nonlinear discriminant surfaces, *International Joint Conference on Neural Networks*, **I**, 423-427, 1992.

[12] A.Sarajedini, R. Hecht-Nielsen, The best of both worlds: Casasent networks integrate multilayer perceptrons and radial basis functions, *International Joint Conference on Neural Networks*, **III**, 905-910, 1992.

[13] T. Andersen , B. Wilamowski, A modified regression algorithm for fast one layer neural network training, *World Congress on Neural Networks*, **I**, 687-690, Washington DC, July 17-21, 1995.

[14] D. E. Rumelhard, J. McClelland, *Parallel Distributed Processing*, vol. 1: Foundations, MIT Press, 1986.

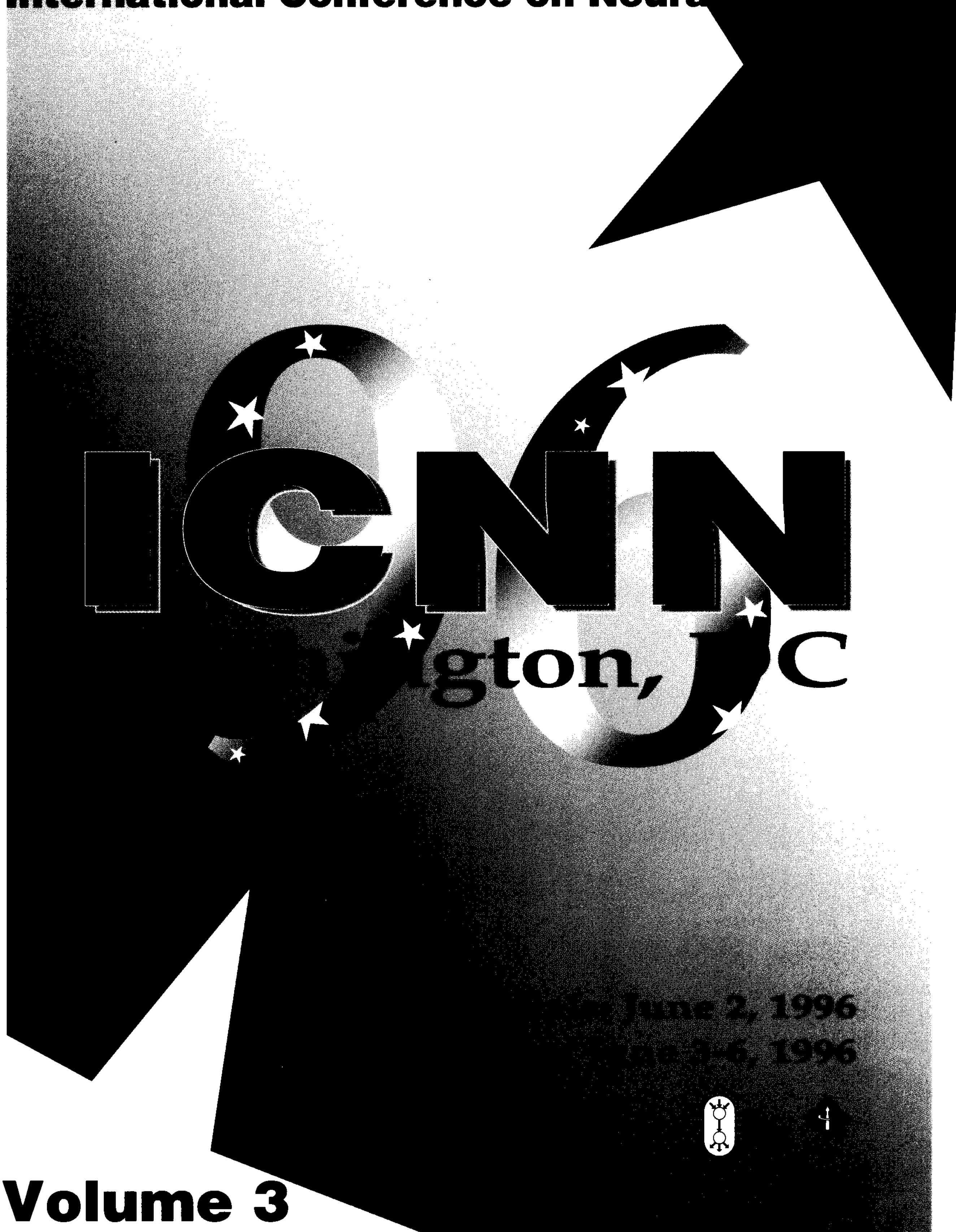
[15] D. W. Marquardt, *J. Soc. Ind. Appl. Math.*, vol. 11, pp. 431-441, 1963.

[16] G. A. Carpenter, S Grossberg, D. Rosen, ART 2-A: An adaptive resonance algorithm for rapid category learning and recognition, *Neural Networks*, vol. **4**, 493-504, 1991.

[17] R. Yager and D. Filev, Generation of fuzzy rules by mountain clustering, *Journal of Intelligent and Fuzzy System*, vol. 2, no. 3, pp. 209-219, 1994.

[18] B. Wilamowski and K. Vieira, Clustering of patterns using radial base function networks, *Artificial Neural Networks in Engineering ANNIE'95*, pp. 109-115, St. Louis, Nov. 1995.

Proceedings of the International Conference on Neura



Volume 3

Sheraton Washington Hotel, Washington, D.C., USA