ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 13-20

A Comparison Between Weighted Radial Basis Functions and Wavelet Networks

M. Sgarbi†, V. Colla*, L.M. Reyneri◇

◇ Dipartimento di Elettronica, Politecnico di Torino,
C.so Duca degli Abruzzi 24, 10129 TORINO, ITALY
* Scuola Superiore Sant'Anna,
Via Carducci 40, 56127 PISA, ITALY
† Via Aquilaia, 302 - 58054 Scansano (GR) - ITALY

Abstract. In the present paper, Wavelet Networks, are proven to be, as well as many other neural paradigms, a specific case of the generic paradigm named *Weighted Radial Basis Functions* Networks. Moreover, a fair comparison between Wavelet and more traditional WRBF networks for function approximation is attempted, in order to demonstrate that the performance depends only on how good the chosen mother/activation function "fits" the function itself.

1. Introduction

Wavelet Networks and Neural Networks (in particular, Radial Basis Functions) are very often used as non-parametric estimators in the fields of function approximation and system modeling.

Wavelet Networks (**WNs**) [3, 4] are an implementation of Wavelet Decomposition, a technique which has recently emerged as a powerful tool for many applications in the field of signal processing, such as data compression and function approximation. The basic idea of Wavelet Decomposition is to expand a generic signal $f(\mathbf{x}) \in L^2(\mathbf{R}^n)$ into a series of functions obtained by dilating and translating a single function $\psi(\mathbf{x})$, the so–called *mother wavelet.*

Radial Basis Functions (**RBFs**) [7, 2] are a class of neural networks particularly suited to function approximation and interpolation. The basic idea is to expand a generic function $f(\mathbf{x})$ into a series of identical radial basis functions (called *activation functions*), each one centered on a different point in the input space (usually, on a multi-dimensional lattice of points). The most commonly used activation functions are monotonically decreasing when moving away from the centers.

So far WNs and RBFs have been considered as two rather different approaches to the task of function approximation, and most paper published on the subject are willing to prove that one method is far better than the other due to some hot point specific of the method. In practice, it has been proven [2] that many neural and fuzzy paradigms are nothing but specific cases of a generic

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 13-20

paradigm called *Weighted Radial Basis Functions* (**WRBFs**), which therefore behaves as a neuro-fuzzy unification paradigm. In this paper we will show that also WNs are a specific case of WRBFs, therefore it can easily be shown that WNs and RBFs can behave (when properly designed) exactly alike.

By extending to WNs the results presented in [2], it can be shown that also the initialization and learning rules of WNs and RBFs can be unified, implying that any rule which can be applied to one of the networks can immediately be applied also to the other type of networks.

As the equivalence of WNs and RBFs is not yet widely known, it often happens that WNs and RBFs are compared with each other under non equivalent conditions. Therefore, all the differences between WNs and RBFs pointed out in literature do not depend on intrinsic peculiarities of either method, but on side effects of the chosen initialization procedure, training rule or mother/activation function.

Aim of this work is to compare WNs and WRBFs fairly, that is, using the same initialization and learning rules for both types of networks. The main results of the work is to prove that the network which performs the best is neither of the two a-priori, but it depends only on which of the chosen mother/activation functions best "fits" the function to be approximated. For instance, a periodic function and an exponential function are better approximated by a WN with an "oscillating" mother wavelet and by a WRBF with Gaussian activation function, respectively. This proves that, in practice, WNs and RBFs both behave as parametric estimators.

The paper is organized as follows: in Sec. 2. WN are presented, while Sec. 3. introduces WRBF networks and shows that WN are a specific case of this latter paradigm. In Sec. 4. some simulation are presented in order to compare the performances of the two examined networks.

2. Wavelet Networks

In the following we shall consider only *radial* wavelets in $L^2(\mathbf{R}^n)$, for which $\psi(\mathbf{x}) = g(\| \mathbf{x} \|)$ where $g : \mathbf{R} \to \mathbf{R}$. Radial functions are characterised by a radial Fourier transform; a function is admissible as a wavelet if $C_\psi = (2\pi)^n \int_0^\infty \frac{|\hat{\psi}(h\omega)|^2}{h} dh < \infty$ and C_ψ is independent of ω.

For the Discrete Wavelet Transform, the parameters which determine the dilation and translation of the mother wavelet are discretised, namely a countable set is extracted, such that the corresponding wavelet family

$$\left\{ \psi_k = \det(\mathbf{D}_k^{1/2}) \psi[\mathbf{D}_k(\mathbf{x} - \mathbf{t}_k)] : \mathbf{t}_k \in \mathbf{R}^n, \mathbf{D}_k = \mathrm{diag}(\mathbf{d}_k), \mathbf{d}_k \in \mathbf{R}_+^n, k \in \mathbf{N} \right\} \tag{1}$$

is a basis for the functions in $L^2(\mathbf{R}^n)$. To this aim, additional conditions are required both on ψ and on the parameters discretisation. The obtained basis is not necessarily orthonormal and can be somehow redundant: in this latter case family (1) is more correctly referred as *frame*. Usually the parameters set is a regular lattice, namely $\mathbf{d}_k = [\alpha^{p1} ... \alpha^{pn}]$, with $p1...pn \in \mathbf{Z}$, and $\mathbf{t}_k = \mathbf{m}\beta$,

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 13-20

where $\mathbf{m} \in \mathbf{Z}^n$, and α and β are positive scalars ($\alpha > 1$) which define the step sizes of the dilation and translation discretisations. The choiche $\alpha = 2$ (in this case the lattice is is often referred as dyadic) and $\beta = 1$ is particularly convenient from the computational point of view and is widely adopted [3, 6]. In practice a signal f is approximated by the weighted sum of a finite number of functions in (1) plus a bias which helps the approximation of functions with nonzero mean value:

$$g(\mathbf{x}) = \sum_{k=1}^{K} a_k \psi[\mathbf{D}_k(\mathbf{x} - \mathbf{t}_k)] + b \tag{2}$$

which is analogous to the output of a 2–layer neural network, provided that the activation function of the hidden neurons are wavelets [4]. Such network has been named *Wavelet Network* (WN). WN with radial wavelets presents the main advantage of an efficient initialisation procedure derived from the wavelet decomposition [5]. Furthermore a fast procedure based on the Orthogonal Least Squares (OLS) algorithm, a method already applied to RBF networks [8], is provided for choosing among all the basis functions those which give the greatest contribution to the approximation. Depending on the form of the function to be approximated, the expansion of a signal into a wavelet series can be more efficient than other solutions, in the sense that fewer basis functions can be needed for achieving a fixed approximation error. This is due to the time–frequency local properties of most wavelets, which make them particularly suitable to represent short–time high–frequency signal features. Fewer basis functions and more efficient initialisation lead to smaller networks and fast training. On the other hand some signal features are better represented by the linear combination of different function, thus WN are not suitable to fit any curve.

3. WRBF Networks

A WRBF neuron is associated to a set of parameters: an *order* $m \in \mathcal{R}$, defining the neuron's *metric*; a *weight vector* $\mathbf{w}$, a *center vector* $\mathbf{c}$, a *bias* b and an *activation function* $F(z)$. The mathematical model of a WRBF neuron is:

$$y = \mathcal{H}_m^{F(z)}(\mathbf{x}; \mathbf{c}, \mathbf{w}, b) \triangleq F\left(\mathbf{w}^T \mathbf{\Delta}_m(\mathbf{x} - \mathbf{c}) + b\right) \tag{3}$$

where $\mathbf{w} = [w_1, w_2, ..., w_n]^T$ and $\mathbf{\Delta}_m(\mathbf{x} - \mathbf{c})$ is a vector of $\mathbf{R}^n$ whose entries are:

$$\mathbf{\Delta}_{m,i}(\mathbf{x} - \mathbf{c}) \triangleq \begin{cases} (x_i - c_i) & \text{for } m = 0 \\ |x_i - c_i|^m & \text{for } m \neq 0 \end{cases} \qquad i = 1, \ldots, n \tag{4}$$

and $F(z)$ is a generic *activation function*; tipical choiches for it are sigmoidal, exponential and linear functions as well as radial wavelets.

The MLP and RBF neural paradigms can be reconduced to WRBF networks [2]; here we underline that also WN are a specific case of WRBF, provided that we define $\mathbf{d} = [w_1^2, w_2^2, ..., w_n^2]^T$, $\mathbf{D} = \text{diag}(\mathbf{d})$, $\psi(\mathbf{x}) = F(\| \mathbf{x} \|)$, $m = 2$ and $b = 0$. Under these hypotheses, we have:

$$F\left(\mathbf{w}^T \mathbf{\Delta}_2(\mathbf{x} - \mathbf{c})\right) = F\left(\mathbf{\Delta}_2(\mathbf{d}^T(\mathbf{x} - \mathbf{c}))\right) = \psi(\mathbf{D}(\mathbf{x} - \mathbf{c})) \tag{5}$$

and the analogy with the functions in the set defined in (1) is thus evident.

Tab. 1 summarises all the unification results.

Kind of layer	**Parameters of the equivalent WRBF layer**
MLP	m=0, **c=0**, F(z) sigmoidal or linear
RBF hidden	m=2, b=0, F(z) exponential
WN hidden	m=2, b=0, F(z) wavelet
RBF and WN output	m=0, **c=0**, F(z) linear

Table 1: The layers of several network structures as particular instances of a WRBF layer.

The generalised learning rule based on the gradient descent algorithm, which has been defined for WRBF networks [2], can be applied to WN as well. This is also true for all the methods which optimise the backpropagation algorithm, such as Rprop and Conjugate Gradient Method [1]. On the other hand, the fast initialisation procedures adopted for RBF networks and WN can be applied to WRBF networks with different activation functions and metrics. Consequently we expect that, provided that the initialisation procedure is fixed, the approximation capabilities of a WRBF network depend on how the chosen activation function "fits" the data.

4. Numerical Results

Our first two simulations concerns the approximation of the following two functions:

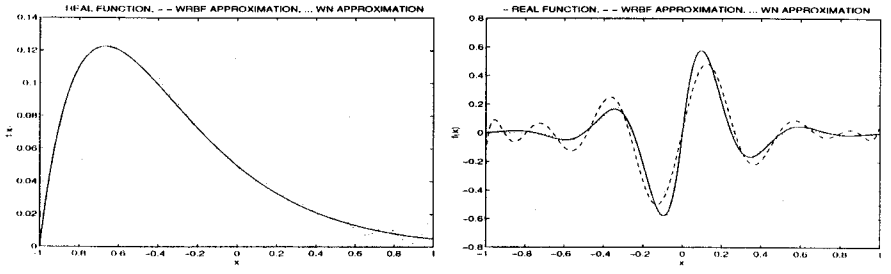
$$y = (x + 1)e^{-3x-3} \tag{6}$$

$$y = sin(4\pi x)e^{-|5x|} \tag{7}$$

by means of a two 2-layer WRBF networks. For both networks the hidden layers have $m = 2$ and $b = 0$, but the former has exponential activation function while the latter adopts a the wavelet known as *mexican hat*, whose expression is $\psi(\mathbf{x}) = (n - \| \mathbf{x} \|^2)exp(- \| \mathbf{x} \|^2 / 2)$ where $\mathbf{x} \in \mathbf{R}^2$. Both networks have linear output layers, i.e. $m = 0$, $c = 0$ and $F(z)$ linear; therefore the latter network could also be referred as a WN, as discussed in the previous sections.

The initialisation algorithm for both networks is the one described in [6], namely a library is builded by dilating and translating the original function so that translation and dilation parameters form a dyadic grid. This is not the only possible solution. A different approach could consist either in clustering

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 13-20

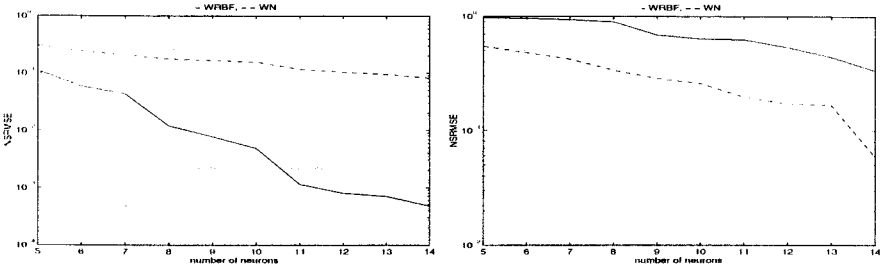


a) b)

Figure 1: a) Comparison between the function (5) and its approximations obtained by means of two WRBF networks having exponential and wavelet activation functions in the hidden layer. b) Comparison between the function (5) and its approximations obtained by means of two WRBF networks having exponential and wavelet activation functions in the hidden layer.

the training set and choosing one or more functions for each cluster or in considering one or more functions for each data point [7]. This latter option is computationally expensive, especially when the dimension of the data set is considerable. Once the library has been built, the most contributive functions in the library are selected via the OLS algorithm until a fixed maximum number of functions is reached [8].

Fig.1.a and Fig.1.b show the results obtained with 14 neurons in the hidden layer. It is evident that function (6) is more suitable than (7) to be approximated by sum of exponentials and therefore in this case the performance of the WN are worse; the contrary holds for function (7). In Fig.s 2.a2.b and we com-



a) b)

Figure 2: a) NSRMSE as a function of the number of neurons in the hidden layer for function (5). b) NSRMSE as a function of the number of neurons in the hidden layer for function (6).

pare the performances of the two kind of networks by varying the number of neurons in the hidden layer: as a performance index we adopt the *Normalised*

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 13-20

Square Root Mean Square Error (NSRMSE) defined in [4], namely:

$$\text{NSRMSE} = \frac{1}{\sigma_y}\sqrt{\sum_{i=1}^{M}[\hat{f}(x_i) - y_i]^2} \tag{8}$$

where M is the number of the data in the training set, (x_i, y_i) are the sample points, σ_y is the standard deviation of the output values and $\hat{f}$ is the function estimate obtained with the network.

We perform some simulations also in the two-dimensional case; we try to approximate the following two functions of the independent variables x and y:

$$z = 2\sin\left(\pi e^{-x^2-y^2}\right) \tag{9}$$

$$z = 2(1 - x^2 - y^2)e^{-x^2-y^2} + 4\sin\left[(x^2 + y^2)e^{-(x^2+y^2)/2}\right] \tag{10}$$

The results in terms of NRSMSE are reported in Tab.s 2.a and 2.b: as expected, function (10) is more suitable than (9) to be approximated by a sum of two–dimensional mexican hat functions.

No. neurons	exp.	Wav.
30	0.018	0.24
40	0.013	0.20
50	$1.7\ 10^{-3}$	0.16
60	$7.7\ 10^{-4}$	0.15

a)

No. neurons	exp.	Wav.
30	0.4	0.25
40	0.37	0.20
50	0.36	0.17
60	0.35	0.1

b)

Table 2: a) NSRMSE obtained with the two different networks for functon (8). b) NSRMSE obtained with the two different networks for functon (9).

5. Conclusions

In the present work we show that WN are a specific case of WRBF networks; WN have been widely adopted in function approximation, thus we guess that also WRBF networks in their more general form can be efficiently applyed to this purpose. The experimental results confirm that the performance of a WRBF network heavily depend on how well its activation function "fits" the function to be approximated, thus traditional WRBF are as good approximators as WN but with different kind of functions.

Acknowledgments

This work has been partially supported by the contract ASI–ARS 96.138 entitled "Optimized Structures for Intelligent Controllers for Flexible Arms."

ESANN'1998 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 22-23-24 April 1998, D-Facto public., ISBN 2-9600049-8-1, pp. 13-20

References

[1] M. Riedmiller: "Advanced Supervised Learning in Multi-layer Perceptrons From Backpropagation to Adaptive Learning Algorithms," *Int. Journ. Computers Standards and Interfaces*, No. 5, 1994.

[2] L.M. Reyneri, "Unification of Neural and Fuzzy Computing Paradigms", in *Proc. of AT-96, 1-st Int'l Symposium on Neuro-Fuzzy Systems*, Lausanne, August 1996.

[3] I. Daubechies: "Ten Lectures on Wavelets," Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, 1992.

[4] Q. Zhang, A. Benveniste: "Wavelet Networks," *IEEE Transactions on Neural Networks*, Vol. 3, No. 6, pp. 889–898, November 1992.

[5] Q. Zhang: "Using Wavelet Network in Nonparametric Estimation," *IEEE Transactions on Neural Networks*, Vol. 8, No. 2, pp. 227–236, March 1997.

[6] Q. Zhang: "Using Wavelet Network in Nonparametric Estimation" *Publication Interne n° 833 de l'Institut de Recherche en Informatique et Systèmes Aléatoires*, Campus Universitaire de Beaulieu, Rennes Cedex, France, June 1994.

[7] E.S. Chng, H. Yang, S. Bos: "Supervised learning of Radial Basis Function Network using adaptive orthogonal least squares learning algorithm,"

[8] S. Chen, C.F.N. Cowan, P.M. Grant: "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks," *IEEE Transactions on Neural Networks*, Vol. 2, No. 2, pp. 302–309, March 1991.