

Comparison between Beta Wavelets Neural Networks, RBF Neural Networks and Polynomial Approximation for 1D, 2D Functions Approximation

Wajdi Bellil, Chokri Ben Amar, and Adel M. Alimi

Abstract—This paper proposes a comparison between wavelet neural networks (WNN), RBF neural network and polynomial approximation in term of 1-D and 2-D functions approximation. We present a novel wavelet neural network, based on Beta wavelets, for 1-D and 2-D functions approximation. Our purpose is to approximate an unknown function $f: \mathbb{R}^n \rightarrow \mathbb{R}$ from scattered samples $(x_i; y = f(x_i))$ $i=1 \dots n$, where first, we have little a priori knowledge on the unknown function f : it lives in some infinite dimensional smooth function space and second the function approximation process is performed iteratively: each new measure on the function $(x_i; f(x_i))$ is used to compute a new estimate \hat{f} as an approximation of the function f . Simulation results are demonstrated to validate the generalization ability and efficiency of the proposed Beta wavelet network.

Keywords—Beta wavelets networks, RBF neural network, training algorithms, MSE, 1-D, 2D function approximation.

I. INTRODUCTION

COMBINING the wavelet transform theory with the basic concept of neural networks [1]-[2], a new mapping network called wavelet neural network or wavenets (WNN) is proposed as an alternative to feedforward neural networks for approximating arbitrary nonlinear functions. Kreinovich prove in [3] that if we use a special type of neurons (wavelet neurons), then the resulting neural networks are *optimal* approximators in the following sense: as $\varepsilon \rightarrow 0$, the number of bits that is necessary to store the results of a 3-layer wavelet neural network approximation, increases slower than for any other approximation scheme. The wavenet learning algorithms consist of two processes: the self-construction of networks and the minimization of error. In the first process, the network structures applied for representation are determined by using

wavelet analysis [4]-[6]. The network gradually recruits hidden units to effectively and sufficiently cover the time-frequency region occupied by a given target. Simultaneously, the network parameters are updated to preserve the network topology and take advantage of the later process. In the second process, the approximations of instantaneous errors are minimized using a regressor algorithm. The parameter of the initialized network is updated using the steepest gradient-descent method of minimization. Each hidden unit has a square window in the time-frequency plane. The optimization rule is only applied to the hidden units where the selected point falls into their windows. Therefore, the learning cost can be reduced.

Wavelets occur in family of functions and each is defined by dilation a_i which controls the scaling parameter and translation t_i which controls the position of a single function, named the mother wavelet $\psi(x)$. Mapping functions to a time-frequency phase space, WNN can reflect the time-frequency properties of function more accurately than the RBFNN. Given an n -element training set, the overall response of a WNN is:

$$\hat{y}(w) = w_0 + \sum_{i=1}^{N_p} w_i \Psi_i \left(\frac{x - t_i}{a_i} \right) \quad (1)$$

where N_p is the number of wavelet nodes in the hidden layer and w_i is the synaptic weight of WNN. A WNN can be regarded as a function approximator which estimates an unknown functional mapping:

$$y = f(x) + \varepsilon \quad (2)$$

where f is the regression function and the error term ε is a zero-mean random variable of disturbance. There are a number of approaches for WNN construction [7]-[9], we pay special attention on the model proposed by Zhang [10].

This paper is structured in 4 sections. After a brief introduction and some basic definitions, we present in section 2 the Beta function and we prove that it is a mother wavelet.

Wajdi Bellil is with University of Sfax, National Engineering School of Sfax, B.P. W, 3038, Sfax, Tunisia (phone: 216-98212934; e-mail: Wajdi.bellil@ieee.org).

Chokri Ben Amar is with University of Sfax, National Engineering School of Sfax, B.P. W, 3038, Sfax, Tunisia (phone: 216-98638417; e-mail: chokri.benamar@ieee.org).

Adel M. Alimi is with University of Sfax, National Engineering School of Sfax, B.P. W, 3038, Sfax, Tunisia (phone: 216-98667682; e-mail: adel.alimi@ieee.org).

This function will be the heart of the new Beta Wavelet Neural Network (BWNN). In section 3 we present the classic problem in regression analysis; we focalized on tow algorithms: the residual based selection and the stepwise selection by orthogonalization. In the last section (section 4), we present some results and tables related to the application of our new Beta Wavelet Neural Network in 1-D approximation, 2-D approximation and some others regressor as RBF neural network and polynomial approximation.

II. BETA WAVELETS

The Beta function [11-13] is defined as: if $p > 0, q > 0, (p, q) \in \mathbb{IN}$

$$\beta(x) = \begin{cases} \left(\frac{x-x_0}{x_1-x_0}\right)^p \left(\frac{x_1-x}{x_1-x_0}\right)^q & \text{if } x \in [x_0, x_1] \\ 0 & \text{else} \end{cases} \quad (3)$$

where, $x_c = \frac{px_1 + qx_0}{p+q}$

A. Derivatives of Beta Function

We prove in [14]-[15] that all derivatives of Beta function $\in L^2(\mathbb{IR})$ and are of class C^∞ . The general form of the n^{th} derivative of Beta function is: $\Psi_n(x) = \frac{d^{n+1}\beta(x)}{dx^{n+1}}$

$$= \left[(-1)^n \frac{n! p}{(x-x_0)^{n+1}} - \frac{n! q}{(x_1-x)^{n+1}} \right] \beta(x) + P_n(x) P_1(x) \beta(x) + \sum_{i=1}^{n-1} C_n^i \left[(-1)^{n-i} \frac{(n-i)! p}{(x-x_0)^{n+1-i}} - \frac{(n-i)! q}{(x_1-x)^{n+1-i}} \right] P_i(x) \beta(x) \quad (4)$$

$$\text{where: } P_1(x) = \frac{p}{x-x_0} - \frac{q}{x_1-x} \quad (5)$$

$$P_n(x) = P_{n-1}'(x) + P_{n-1}(x) P_1(x) \quad \forall n > 1$$

The first (Beta 1), second (Beta 2) and third (Beta 3) derivatives of Beta wavelet are shown graphically in Fig. 1.

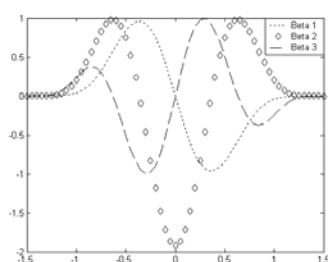


Fig. 1 First, second and third derivatives of Beta function

B. Proposition

if $p = q$, for all $n \in \mathbb{IN}$ and $0 < n < p$ the functions $\Psi_n(x) = \frac{d^n \beta(x)}{dx^n}$ are wavelets [14]-[15].

III. 1-D APPROXIMATION FORMULATION

The mathematical formulation of 1-D function approximation is given by:

Let $E = \{(x_k, y_k) | k = 0, 1, \dots, K\}$. We want to find N samples of $f(x)$ satisfying $f(x_k) = y_k$ for $k = 0, 1, \dots, K$.

A. Polynomial Approximation: Lagrange Method

We should find a polynomial $P_K(x_k)$ ($k = 0, \dots, K$) equal to $f(x_k)$ for all the known $(K+1) x_k$.

The general formula of Lagrange polynomial is :

$$L_k(x) = \prod_{\substack{j=0 \\ j \neq k}}^K \frac{x - x_j}{x_k - x_j} \quad (6)$$

$$P_K(x) = \sum_{k=0}^K L_k(x) f(x_k) = \sum_{k=0}^K \left[\prod_{\substack{j=0 \\ j \neq k}}^K \frac{x - x_j}{x_k - x_j} \right] f(x_k)$$

B. RBF Neural Network Approximation

A neural network with one output y , d inputs $\{x_1, x_2, \dots, x_d\}$ and L nodes can be parameterized as follows:

$$\hat{y}(x) = \sum_{i=1}^L w_i \phi(x) \quad (7)$$

$$x = [x_1, x_2, \dots, x_d]$$

where w_i is the weight of neurons and ϕ is the activation function.

C. Wavelet Neural Network Approximation

Given an n -element training set, the overall response of a WNN is:

$$\hat{y}(x) = \sum_{i=1}^{N_p} w_i \Psi_i \left(\frac{x - t_i}{a_i} \right) \quad (8)$$

Where N_p is the number of wavelet nodes in the hidden layer and w_i is the synaptic weight of WNN. A WNN can be regarded as a function approximator which estimates an unknown functional mapping:

$$y = f(x) + \varepsilon \quad (9)$$

Where f is the regression function and the error term ε is a zero-mean random variable of disturbance. There are a number of approaches for WNN construction [16]-[17], we pay special attention on the model proposed by Zhang [18].

D. Selecting Best Wavelets Regressors

Selecting the best regressor from a finite set of regressor candidates is a classical problem in regression analysis [18]. In our case the set of regressor candidates is the wavelets library W as defined by Zhang [10]. The problem is then to select a number $M < L$ of wavelets from W , the best ones based on the training data O_1^N , for building the regression.

$$f_M(x) = \sum_{i \in I} u_i \Psi_i(x) \quad (10)$$

where I is an M -elements subset of the index set $\{1, 2, \dots, L\}$, $u_i \in \mathbb{R}$

We denote by W the set of wavelets resulting from the refining of family:

$$\begin{aligned} & \{\Psi(a_0^n x - mt_0) : n \in S_a, m \in S_i(n), (S_a, S_i) \subset \mathbb{Z}^2\}, \text{ which called} \\ & \text{it the library of wavelet regressor candidates.} \\ & W = \{\Psi_i : \Psi_i(x) = \alpha_i \Psi(a_i(x - t_i))\} \\ & \alpha_i = \left(\sum_{k=1}^N [\Psi(a_i(x_k - t_i))]^2 \right)^{-\frac{1}{2}}, i = 1, \dots, L \end{aligned} \quad (11)$$

E. Regression Selection Problem

Given a wavelet library W defined by (3.2) and a set of training data O_1^N , let I_M be the set of all the M -elements subset of $\{1, 2, \dots, L\}$, $M < L$. the problem is to find $I \in I_M$ that minimizes:

$$J(I) = \min_{u_i, i \in I} \frac{1}{N} \sum_{k=1}^N \left(y_k - \sum_{i \in I} u_i \Psi_i(x_k) \right)^2 \quad (12)$$

In principle such a selection can be performed by examining all the M -elements subsets of W , but the combination of all the possible subsets is usually very large and exhaustive examination may not be feasible in practice. Some sub-optimal and heuristic solutions have to be considered. In the following we propose to apply several such heuristic procedures.

F. The Residual based Selection

The idea of this method [20]-[21] is to select, for the first stage, the wavelets in W that best fit the training data O_1^N , and then iteratively select the wavelet that best fit the residual of the fitting of the previous stage. In the literature of the classical regression analysis, it is considered as a simple, but not effective method, for example in [19] where it is called stage-wise regression procedure. It is also similar to the projection pursuit regression (PPR) [22]-[23], but much simpler than the latter. Because for classical regression the number of regressor candidates is relatively small, some more

complicated may reach several hundreds or even more, the computational efficiency becomes more important and the simplicity of this method is of interest. Recently it is also used in some similar problems, such as the matching pursuit algorithm of S. Mallat and Z. Zhang [10] and the adaptive signal representation of S. Qian and D. Chen [17]. Define the initial residual $\gamma_0(k) = y_k$, $k = 1, \dots, N$, with y_k the output observations in O_1^N .

Set $f_0(x) \equiv 0$; at stage I ($I=1, \dots, M$), search among W the wavelet Ψ_j that minimizes:

$$J(\Psi_j) = \frac{1}{N} \sum_{k=1}^N (\gamma_{i-1}(k) - u_j \Psi_j(x_k))^2 \quad (13)$$

where $u_j = \left(\sum_{k=1}^N (\Psi_j(x_k))^2 \right)^{-1} \sum_{k=1}^N \Psi_j(x_k) \gamma_{i-1}(k)$ (14)

and $\gamma_{i-1}(k)$ ($k = 1, \dots, N$) are the residual of stage $i-1$.

note $l_i = \arg \min_{1 \leq j \leq L} J(\Psi_j)$

then Ψ_{l_i} is the wavelet selected at stage i . Update f_i and γ_i :

$$\begin{aligned} f_i(x) &= f_{i-1}(x) + u_{l_i} \Psi_{l_i}(x) \\ \gamma_i(k) &= \gamma_{i-1}(k) - u_{l_i} \Psi_{l_i}(x_k) \quad k = 1, \dots, N \end{aligned}$$

G. Stepwise Selection by Orthogonalization

The above residual based selection procedure [24]-[26] does not explicitly consider the interaction or the non orthogonality of the wavelets in W . The idea of this alternative method is to select, for the first stage, the wavelet in W that best fits the training data O_1^N , and then iteratively select the wavelet that best fits O_1^N while working together with the previously selected wavelets. Recently this method has been used for training radial basis function neural networks (RBFNN) and other nonlinear modeling problems by Chen et al. [17]-[18].

At stage I of this procedure, assume that $i-1$ already selected wavelets correspond to the vectors $v_{1,1}, \dots, v_{i-1,i-1}$. In order to select the I^{th} wavelet, we have to compute the distance from y to the space $\text{span}(v_{1,1}, \dots, v_{i-1,i-1}, v_j)$ for each $j=1, \dots, L$ and $j \neq 1, \dots, i-1$. For computational efficiency, we orthogonalize the later selected vectors v_j to the earlier selected ones.

IV. EXPERIMENTS

In this section, we present some experimental results of the proposed Beta Wavelet Neural Networks on approximating three 1-D functions using the Stepwise selection by orthogonalization training algorithms. First, simulations on the 1-D function approximation are conducted to validate and

compare the proposed BWNN with some others wavelets and some others method such as RBF neural networks and polynomial approximation. The input x is constructed by the uniform distribution, and the corresponding output y is functional of $y = f(x)$. The training and test data are composed of 50 points and 500 points, respectively. The same functions are used on RBF neural networks and polynomial approximation in order to compare these approximators method. Second, we approximate four 2-D functions using Beta wavelet networks and some others wavelets networks to illustrate the robustness of the proposed wavelets family.

We compare the performances of the three kind of regressor by the Mean Square Error (MSE) defined by:

$$MSE = \frac{1}{M} \sqrt{\sum_{i=1}^M \left[\hat{f}(x_i) - y_i \right]^2} \quad (15)$$

A. 1-D approximation using the Stepwise Selection by Orthogonalization Algorithm

These results are given, using the Stepwise selection by orthogonalization algorithm, on a wavelet neural network using 6 wavelets, 4 levels decomposition, 300 iterations and 50 uniform spaced points for training.

The table below gives the mean square error using traditional wavelets and Beta wavelets:

TABLE I

COMPARISON OF MSE FOR BETA WAVELETS NEURAL NETWORKS AND SOME OTHERS USING THE STEPWISE SELECTION BY ORTHOGONALIZATION ALGORITHM IN TERM OF 1-D FUNCTIONS APPROXIMATION

Functions	Mex-hat	Beta 1	Beta 2
F1	3.03e-005	1.44e-004	4.00e-005
F2	9.82e-006	3.35e-005	8.21e-006
F3	4.87e-005	2.15e-004	2.84e-005

Functions	Beta 3	Beta 4	Beta 5
F1	1.30e-005	2.76e-005	2.58e-005
F2	2.72e-005	1.99e-006	4.28e-005
F3	2.09e-005	4.94e-005	1.57e-003

Functions	Beta 6	Polywog2	Slog1
F1	3.76e-005	3.86e-005	4.79e-005
F2	1.02e-004	1.09e-005	2.10e-004
F3	2.46e-005	1.32e-004	1.16e-004

The best approximated functions F1, F2 and F3 are displayed in Figs. 2. For F1 the Mean Square Error (MSE) of the Mexican hat WNN is 3.03e-005, comparing to 1.30e-005 the Beta 3 WNN achieved. The fourth derivative of Beta wavelet approximate the second function F2 with an MSE equal to 1.99e-006 where the MSE using the Mexican hat wavelet network is 9.82e-006. Finally we can see that the MSE is equal to 2.09e-005 for Beta 3 WNN comparing to 4.87e-005 for Mexican hat WNN.

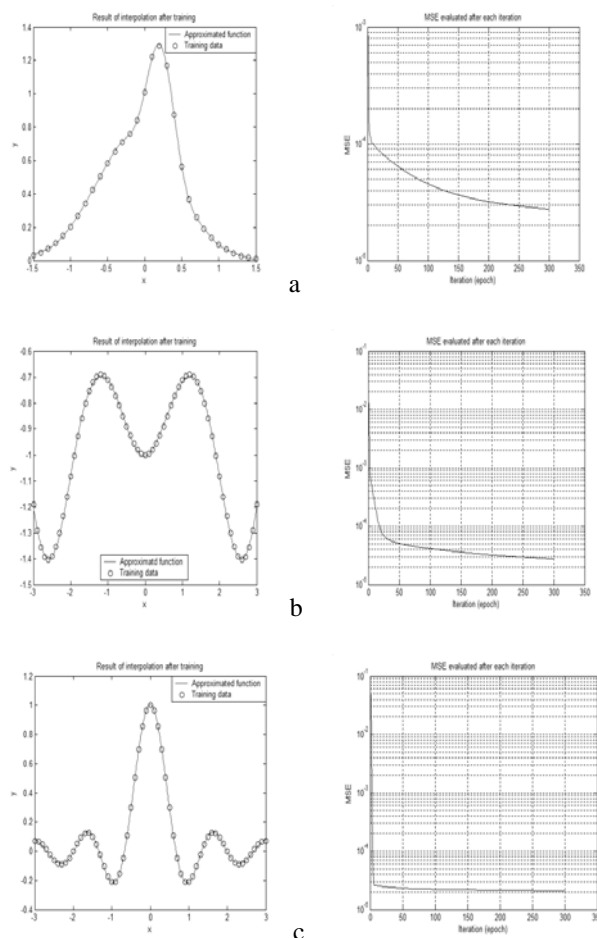


Fig. 2 Approximated 1-D function using wavelet neural network and MSE evolution

a- Approximation of F1 with Beta 3 WNN $p=45, q=45$

b- Approximation of F2 with Beta 4 WNN $p=55, q=55$

c- Approximation of F3 with Beta 2 WNN $p=24, q=24$

B. 1-D Approximation using Neural Network

The table below gives results of approximation on the functions F1, F2 and F3 using RBF neural networks constructed with 6 neurones on the hidden layer and 300 iterations for learning.

TABLE II

THE MSE USING RBF NEURAL NETWORKS IN TERM OF 1-D FUNCTIONS APPROXIMATION

Functions	F1	F2	F3
MSE	1.134e-005	4.446e-006	1.016e-003

C. 1-D polynomial approximation

In the case of polynomial approximation we change the polynomial degree to evaluate the MSE criteria. The table below gives the MSE for different polynomial degree for the functions F1, F2 and F3.

TABLE III
EVOLUTION OF MSE IN TERM OF POLYNOMIAL DEGREE FOR 1-D FUNCTIONS
APPROXIMATION

Functions	F1		
Deg	8	9	10
MSE	7.71e-003	4.26e-003	4.25e-003

Functions	F2		
Deg	6	7	8
MSE	1.43e-003	1.43e-003	2.36e-005

Functions	F3		
Deg	8	9	10
MSE	1.39e-003	1.39e-003	1.40e-003

From these simulations we can deduce the superiority of the wavelet neural networks over neural networks and polynomial approximation in term of 1-D functions approximation. The WNN based on Beta wavelets have the best approximators because these wavelets family have the particularity of the adjustment of the parameters p , q x_0 and x_1 .

D. 2-D Approximation using the Stepwise Selection by Orthogonalization Algorithm

These results are given, using the Stepwise selection by orthogonalization algorithm, on a wavelet neural networks using 9 wavelets, 4 levels decomposition, 100 iterations to approximate some 2-D functions (S1, S2, S3 and S4 given on figure 4) using a uniform distribution of 11x11 points and 11x11 randomly distribution (see Fig. 3).

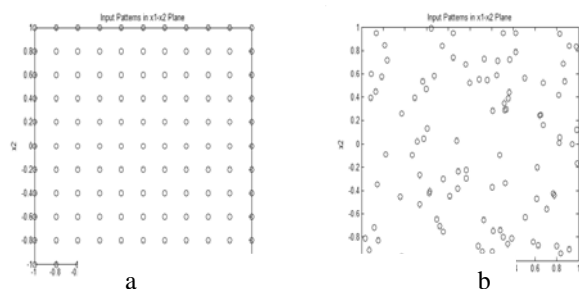


Fig. 3 Distribution points or approximation
a- Uniform distribution of 11x11 points
b- Randomly distribution of 11x11 points

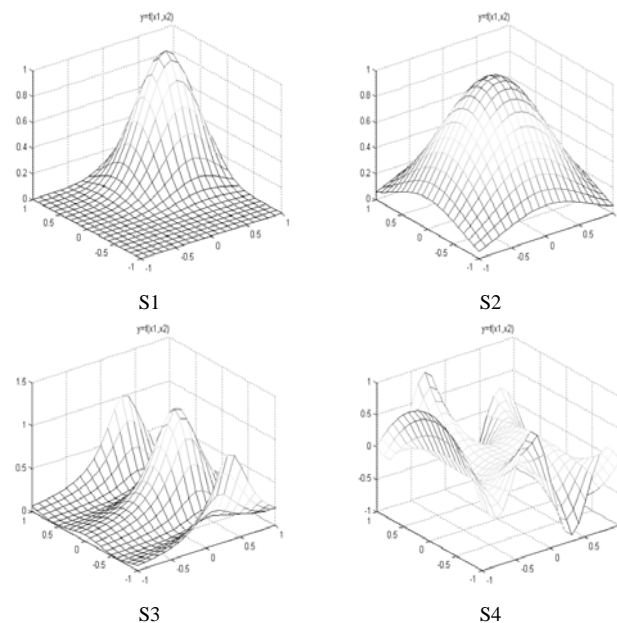


Fig.4: 2-D functions

$$S_1(x, y) = e^{-\frac{81}{16}[(x-0.5)^2 + (y-0.5)^2]}$$

$$S_2 = \left(\frac{x+2}{2}\right)^5 \left(\frac{2-x}{2}\right)^5 \left(\frac{y+2}{2}\right)^5 \left(\frac{2-y}{2}\right)^5$$

$$S_3(x, y) = \frac{3.2(1.25 + \cos(5.4y))}{6 + 6(3x-1)^2}$$

$$S_4(x, y) = (x^2 - y^2) * \sin(5x)$$

The table below gives the MSE of the 2-D function approximation, using WNN, for the tow distributions points.

TABLE IV
THE MSE FOR BETA WAVELETS NEURAL NETWORKS AND SOME OTHERS
USING THE STEPWISE SELECTION BY ORTHOGONALIZATION ALGORITHM IN
TERM OF 2-D FUNCTIONS APPROXIMATION USING DISTRIBUTION 1 AND 2

Surface	Distribution	Mex-hat	Beta 1	Beta 2
S1	1	3.32e-004	6.82e-007	2.20e-007
S2	1	1.2 e-004	2.88e-007	1.21e-007
S3	1	3.25 e-003	4.81 e-004	2.65 e-004
S4	1	4.13 e-002	4.92 e-004	1.00 e-003

Surface	Distribution	Beta 3	Beta 4	Beta 5
S1	1	4.62e-007	1.43e-006	1.01e-005
S2	1	2.22e-007	6.05e-007	1.46e-005
S3	1	3.37e-004	7.94 e-004	1.92 e-003
S4	1	5.20 e-004	1.64 e-003	6.26e-004

Surface	Distribution	Beta 6	Polywog 2	Slog1
S1	1	4.07e-004	3.44e-006	2.10e-005
S2	1	4.50e-006	6.02e-006	7.33e-005
S3	1	9.91 e-004	6.26e-005	4.33 e-003
S4	1	1.99 e-003	1.91 e-003	8.71 e-003

Surface	Distribution	Mex-hat	Beta 1	Beta 2
S1	2	1.63e-004	3.83e-007	1.38e-007
S2	2	4.64e-005	2.60e-007	6.83e-006
S3	2	2.18 e-003	1.71 e-004	7.91 e-004
S4	2	1.05 e-002	2.25 e-004	3.94 e-004

Surface	Distribution	Beta 3	Beta 4	Beta 5
S1	2	5.36e-007	4.41e-007	5.85e-007
S2	2	4.27e-007	1.04e-006	3.31e-006
S3	2	4.37e-004	1.64e-004	5.01e-004

S4	2	4.85e-004	5.17e-004	5.16e-004
Surface	Distribution	Beta 6	Polywog 2	Slog1
S1	2	4.12-007	5.79e-006	9.68e-005
S2	2	8.14e-006	5.37e-006	5.46e-005
S3	2	7.41e-004	1.78e-004	1.56e-003
S4	2	4.64e-004	1.08e-003	6.37e-003

We display in Fig. 4 the 2-D data used in our tests. From table 4 we can see that Beta wavelet networks are more suitable for 2-D function approximation than the others wavelets neural networks. For example we have an MSE equal to 2.20e-007 when we use a uniform distribution input patterns point for training to approximate the surface S1 using Beta 2 wavelet neural networks over 3.32e-004 if we use Mexican hat wavelet neural networks and an MSE for Beta 1 and Mexican hat respectively of 1.38e-007 and 1.63e-004 when we use a randomly input patterns point.

V. CONCLUSION

We present two experimental results of the proposed Beta Wavelet Neural Networks (BWNN) on approximating 1-D and 2-D functions using the Stepwise selection by orthogonalization training algorithms. First, simulations on the 1-D function approximation on which we prove the superiority of Beta wavelets in term of MSE over the classical wavelets network and the RBF neural networks and the polynomial approximation method. Second, the two-dimension functions are approximated with the Beta wavelets and some others one to illustrate the robustness of the proposed wavelets family. This new wavelets family can be used to approximate volume using the 2-D 1-D 2-D technique.

ACKNOWLEDGMENT

The author would like to acknowledge the financial support of this work by grants from the General Direction of Scientific Research and Technological Renovation (DGRSRT), Tunisia, under the ARUB program 01/UR/11/02.

REFERENCES

- [1] S.Li and E. Leiss, "On Noise-immune RBF Networks, in Radial Basis Function Neural Networks: Recent Developments in Theory and Applications", Editors: Howlett, R. J. and Jain, L.C., Springer Verlag, pp. 95-124, 2001.
- [2] P. Cristea, R. Tuduce, and A. Cristea, "Time Series Prediction with Wavelet Neural Networks", Proceedings of IEEE Neural Network Applications in Electrical Engineering, pp. 5-10, 2000.
- [3] V. Kreinovich, O. Sirisaengtaksin. S. Cabrera, "Wavelets compress better than all other methods: a 1-dimensional theorem University of Texas at El Paso", Computer Science Department, Technical Report, June 1992.
- [4] D.Lee, "An application of wavelet networks in condition monitoring", IEEE Power Engineering Review 19, 69-70, 1999.
- [5] M.Thuillard, "Applications of wavelets and wavenets in soft computing illustrated with the example of fire detectors", SPIE Wavelet Applications VII, April 24-28 2000.
- [6] J.Fernando Marar, E.C.B.Carvalho Filho, G.C.Vasconcelos, "Function Approximation by Polynomial Wavelets Generated from Powers of Sigmoids", SPIE 2762, 365-374. 1996.
- [7] Y. Oussar, I. Rivals, L. Personnaz & G. Dreyfus, "Training Wavelet Networks for Nonlinear Dynamic Input-Output Modeling", Neurocomputing, 1998.

- [8] J.Echaz, "Strategies for Fast Training of Wavelet Neural Networks, 2nd International Symposium on Soft Computing for Industry", 3rd World Automation Congress, Anchorage, Alaska, May 1998.
- [9] V.Kruger, A. Happe, G. Sommer, "Affine real-time face tracking using a wavelet network", Proc. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, 26-27 Sept. 1999 (Corfu), IEEE, 141-8, 1999.
- [10] Q. Zhang, "Using Wavelet Network in Nonparametric Estimation", IEEE Trans. Neural Network, Vol. 8, pp.227-236, 1997.
- [11] M.A Alimi, "The Beta System: Toward a Change in Our Use of Neuro-Fuzzy Systems", International Journal of Management, Invited Paper, no. June, pp. 15-19, 2000.
- [12] M.A Alimi, "The Beta-Based Neuro-Fuzzy System: Genesis and Main Properties", TASK Quarterly Journal, Special Issue on "Neural Networks" edited by W. Duch and D. Rutkowska, vol. 7, no. 1, pp. 23-41, 2003.
- [13] C. Aouiti, M.A Alimi, and A. Maalej, "Genetic Designed Beta Basis Function Neural Networks for Multivariable Functions Approximation, Systems Analysis, Modeling, and Simulation", Special Issue on Advances in Control and Computer Engineering, vol. 42, no. 7, pp. 975-1005, 2002.
- [14] W.Bellil, C.Ben Amar et M.Adel Alimi, "Beta Wavelet Based Image Compression", International Conference on Signal, System and Design, SSD03, Tunisia, vol. 1, pp. 77-82, Mars, 2003.
- [15] W.Bellil, C.Ben Amar, M.Zaied and M.Adel Alimi, "La fonction Beta et ses dérivées : vers une nouvelle famille d'ondelettes", First International Conference on Signal, System and Design, SCS'04, Tunisia, vol. 1, P. 201-207; Mars 2004.
- [16] S.Qian and D.Chen, "Signal representation using adaptative normalized gaussian function", Signal processing, vol.36, 1994.
- [17] S.Chen, C. Cowan, and P. Grant, "Orthogonal least squares learning algorithm for radial basis function networks", IEEE Trans. On Neural Networks, vol. 2, pp.302-309, March 1989.
- [18] S.Chen, S. Billings, and W. Luo, "Orthogonal least squares learning methods and their application to non-linear system identification", Ind. J. Control, vol. 50, pp.1873-1896, 1989.
- [19] N. Draper and H. Smith, Applied regression analysis, Series in probability and Mathematical Statistics", Wiley, Second edi. 1981.
- [20] K. Hornik, M. Stinchcombe, H. White and P. Auer, "Degree of Approximation Results for Feedforward Networks Approximating Unknown Mappings and Their Derivatives", Neural Computation, 6 (6), 1262-1275, 1994.
- [21] C.S.Chang, Weihui Fu, Minjun Yi, "Short term load forecasting using wavelet networks", Engineering Intelligent Systems for Electrical Engineering and Communications 6, 217-23, 1998.
- [22] J.Frieman and W. Stuetzle, "Projection pursuit regression", J. Amer. Stat. Assoc., vol. 76, pp.817-823, 1981.
- [23] P. Huber, Projection pursuit", Ann. Statist., vol. 13, pp 435, 1985.
- [24] Q. Zhang and A. Benveniste, "Wavelet Networks", IEEE Trans. on Neural Networks 3 (6) 889-898, 1992.
- [25] J. Zhang, G. G. Walter, Y. Miao and W. N. Wayne Lee, "Wavelet Neural Networks For Function Learning", IEEE Trans. on Signal Processing 43 (6), 1485-1497, 1995.
- [26] Q. Zhang, "Using Wavelet Network in Nonparametric Estimation", IEEE Trans. on Neural Networks, Vol. 8, No. 2, pp. 227-236, 1997.