161-172

ISBN : 978-602-96426-0-5

PROCEEDING of

IndoMS International Conference on Mathematics and Its Applications (IICMA) 2009

Yogyakarta - Indonesia, October 12th- 13th, 2009

Published by
IndoMS (Indonesian Mathematical Society)

Secretariat :
Department of Mathematics
Faculty of Mathemathics & Natural Sciences
Gadjah Mada University
Kampus Sekip Utara
Yogyakarta - Indonesia, 55281
Phone : +62 - 274 - 552243 ; 7104933
Fax. : +62 - 274 555131

Proceedings of IICMA 2009
Analysis, pp. 161-172.

WAVELET NEURAL NETWORK ON MULTIRESOLUTION ANALYSIS WITH PARTICLE SWARM OPTIMIZATION

JULAN HERNADI

Abstract. The main issue in the artificial neural network (ANN) includes the architecture, the activation function and the appropriate learning algorithm. So far, the application of wavelet in ANN are concerned with the wavelet (mother wavelet), instead of the activation function. This paper proposes some new ideas to the architecture of ANN based on the structure of multiresolution analysis (MRA), where the scaling function (father wavelet) plays a key role for approximation. These ideas also offer that both types the mother and the father wavelet to be used simultaneously as bridge connecting between the hidden layer and the output layer. An interesting optimization technique, particle swarm optimization (PSO) is also introduced for training the network. In addition, the numerical simulation is presented to show the performance of the model.

Key words and Phrases: Artificial Neural Network, Wavelet, PSO.

1. Introduction

The wavelet theory is a rapidly developing branch of mathematics which has found many application, for instance, in numerical analysis and signal processing, e.g. see Mallat [14]. This theory is very attractive and has offered very efficient algorithms for analyzing, approximating and estimating function or signal. On the other hand, the neural networks are a class of computational architecture that are composed of interconnected nodes or neurons. Due to the similarity between wavelet theory, specially discrete inverse wavelet transform and one-hidden layer neural network, the idea of combining both wavelets and neural networks has been proposed by Zhang and Benveniste [17]. Thereafter, the combination of those two topics created a new research field which is called wavelet neural network (WNN). One of the characteristic WNN is the presence of the wavelet bases, instead of the

2000 Mathematics Subject Classification: 78M32, 63M45, 42C40, 65D15.

activation function such as sigmoid function relating between the hidden unit and the output unit, e.g. Zhang [18], Chen at al [2, 3], Kobayashi and Torioka [12].

Generally, there are two ways to introduce wavelet, through continuous wavelet transform (CWT) and mutiresolution analysis (MRA). The first-mentioned concerned with a single function which oscillates and has zero mean. This function is called wavelet or mother wavelet. On the MRA view of point, there is another functions involved, the scaling function (father wavelet) which has a unity area property, beside the wavelet function as in CWT. In practice, the scaling function to be used for approximation, whereas the wavelet to capture the residual or detail, usually in the oscillation form. Not all wavelet function has a counterpart scaling function, but every scaling function of MRA there is always a corresponding wavelet function, see Daubechies [6].

So far, the wavelet bases used in WNN is only derived from mother wavelet, see Zhang [17] and Galvao at al [7]. Chen at al [2, 3] even allow all the scaling and translating parameters running on the real line to be optimized. It is redundance, instead, the parameters are enough to be taken from a set of dyadic points. The wavelets commonly used in WNN are mexican-hat and Gaussian wavelet. Unfortunately, these wavelets are not of compactly supported so that the computation becomes inefficiency because the matrix representing function evaluation of input data becomes dense. Instead of wavelet function, the scaling function of MRA with compactly supported has chance of success to be applied in WNN. The basis from scaling function has been successfully applied to approximate some operator equations, see Hernadi [9].

The another crucial issue in ANN is the learning algorithm, an algorithm to determine the network weights via a given training sample. The algorithms common used involve the gradient method, e.g. gradient steepest descent, or the quasi-Newton method, e.g. the BFGS algorithm. But its disadvantages are slow convergence and easy trap at local minimum. Particle swarm optimization (PSO) is a population based optimization method first proposed by Kennedy and Eberhart [10] in 1995. This method incorporates swarming behavior observed in flocks of birds, school of fish, or swarms of bees, and even human social behavior, from which the idea was emerged, see Clerk [4] for detail explanation. The main strength of PSO is its fast convergence, which compares with global optimization algorithms like genetic algorithm (GA), Abraham at al [1].

This paper offers a slightly different WNN model than previous results. Here, the output are based on the structure of MRA. Hence, the scaling function or father wavelet plays key role in the model. The parameters of translation and dilation to be considered here are dyadic points and characterized by two integers j and k, instead of whole real values such as those in CWT. In the computation, the paper introduces the way of choosing the translation parameters k's adaptively depending on the scaling parameter j and the domain of input data.

This paper is organized as follows: Section 2 introduces an overview of wavelet theory and muliresolution analysis. The models of wavelet neural network on MRA is described in section 3. The particle swarm optimization used for learning is presented in section 4. The computation procedures and some numerical experiments is given in section 5. Finally, concluding remarks and some discussions are described in the last section.

2. The Wavelet Theory: An Overview

There are two ways to introduce wavelet, through the continuous wavelet transform (CWT) and multiresolustion analysis (MRA). Let ψ be a function in $\mathcal{L}^2(\mathbb{R})$. Furthermore, the collection of functions $\psi_{a,b}$ are defined as

$$\psi_{a,b}(x) := \frac{1}{\sqrt{|a|}}\psi\left(\frac{x-b}{a}\right) \text{ where } a, b \in \mathbb{R}, a \neq 0.$$

For each $a, b \in \mathbb{R}, a \neq 0$, the function $\psi_{a,b}$ is called atomic and its norm does not depend on both parameters a and b, indeed

$$\|\psi_{a,b}\|_{\mathcal{L}^2(\mathbb{R})} = \|\psi\|_{\mathcal{L}^2(\mathbb{R})}.$$

The continuous wavelet transform (CWT) of $f \in \mathcal{L}^2(\mathbb{R})$ is the function $\mathcal{W}f$ depending on parameters a and b and defined by

$$(\mathcal{W}f)(a,b) := (f, \psi_{a,b}) = \frac{1}{\sqrt{|a|}}\int_{-\infty}^{\infty} f(x)\psi\left(\frac{x-b}{a}\right)dx. \tag{1}$$

If ψ satisfies the admissibility condition, i.e. $C_\psi = \int_{-\infty}^{\infty}\frac{|\hat{\psi}(\omega)|^2}{\omega}d\omega < \infty$ then the CWT is invertible, see Daubechies [6] or Tang at al [16]. In addition, if $\psi \in \mathcal{L}^1(\mathbb{R})$ then the admissibility condition implies

$$\hat{\psi}(0) = \int_{-\infty}^{\infty}\psi(x)dx = 0 \tag{2}$$

that is the function ψ oscillates, hence this function is called **wavelet** or 'small wave'. As a result, in engineering practical, the wavelet is often defined as the function $\psi \in \mathcal{L}^2(\mathbb{R})$ which satisfies (2). This definition is not exact in mathematics, however, it is harmless in the application of wavelet. Two most popular wavelets in WNN are Gaussian wavelet $\psi_\alpha(x) = -\frac{x}{4\alpha\sqrt{\pi\alpha}}e^{-\frac{x^2}{4\alpha}}$ and Mexico-hat wavelet $\psi_M(x) = \frac{1}{\sqrt{2\pi}}(1-x^2)e^{x^2/2}$.

In most application the function f in (1) to be reconstructed by only discrete values instead of whole continuous values of a and b. The most often choice is the dyadic points, $a = 2^{-j}$ and $b = k2^{-j}$, $j, k \in Z$. Furthermore, $\psi \in \mathcal{L}^2(\mathbb{R})$ is called wavelet if the set of functions $\{2^{j/2}\psi(2^jx-k) : j, k \in Z\}$ constitutes an orthonormal basis of $\mathcal{L}^2(\mathbb{R})$. Consequently, each $f \in \mathcal{L}^2(\mathbb{R})$ can be represented as

$$f(x) = \sum_{j\in Z}\sum_{k\in Z} d_{j,k}\psi_{j,k}(x) \tag{3}$$

where

$$d_{j,k} = (f, \psi_{j,k}) = \int_{-\infty}^{\infty} f(x)\psi_{j,k}(x)dx. \tag{4}$$

The sequence of coefficients $(d_{j,k})$ in (4) is defined as the discrete wavelet transform of f with respect wavelet ψ.

Recall that the equality in (3) does work when $f \in \mathcal{L}^2(\mathbb{R})$ since the convergence on the right hand side is valid in $\mathcal{L}^2(\mathbb{R})$, but not for all f in $L^\infty(\mathbb{R})$ or $L^1(\mathbb{R})$. For example, if $f(x) = 1$, and because of the oscillation property then all the coefficients $d_{j,k} = 0$ and (3) gives $1 = 0$, a contradiction. This fact says that the wavelet basis can not capture the constant part of any function. In the other hand, the bases arising from ϕ could be more appropriate to be used in approximation than wavelet itself.

In some cases, the wavelet ψ can be derived from ϕ through MRA. The MRA was first introduced by Mallat [13] set up from a function $\phi \in \mathcal{L}^2(\mathbb{R})$ which satisfies

$$\int_{-\infty}^{\infty} \phi(x)dx = 1. \tag{5}$$

Then, a ladder closed subspaces $\{0\} \subset ... \subset V_{-1} \subset V_0 \subset V_1 ... \subset \mathcal{L}^2(\mathbb{R})$ where V_j is defined as

$$V_j := \overline{\text{span}\{\phi_{j,k}; k \in Z\}}$$

where $\phi_{j,k}(x) := 2^{j/2}\phi(2^j x - k)$ and $\overline{E}$ stands for the closure of E is an MRA of $\mathcal{L}^2(\mathbb{R})$. Moreover, if $P_j : \mathcal{L}^2(\mathbb{R}) \to V_j$ is the orthogonal projection onto V_j then for each $f \in \mathcal{L}^2(\mathbb{R})$ we have $\lim_{j\to\infty} P_j f = f$ in $\mathcal{L}^2(\mathbb{R})$. This means that every function in $\mathcal{L}^2(\mathbb{R})$ can be approximated as accurate as possible by functions in V_j's.

Let $\{V_j\}_{j=-\infty}^{\infty}$ be a MRA of $\mathcal{L}^2(\mathbb{R})$. For every $j \in Z$, let us define W_j the orthogonal complement of V_j in V_{j+1}, that is

$$V_{j+1} = V_j \oplus W_j. \tag{6}$$

Moreover, there always exists a function $\psi \in \mathcal{L}^2(\mathbb{R})$ such that $\{2^{j/2}\psi(2^j x - k)\}_{k\in Z}$ constitutes an orthonormal basis of W_j. Such a function ψ is called wavelet and W_j is defined as

$$W_j = \overline{\text{span}\{\psi_{j,k}; k \in Z\}}$$

is called detail space. By limiting process $\mathcal{L}^2(\mathbb{R})$ is decomposed as

$$\bigoplus_{j=-\infty}^{\infty} W_j = \mathcal{L}^2(\mathbb{R}).$$

This expression tells us that $\{\psi_{j,k} : j, k \in Z\}$ is an orthonormal basis of $\mathcal{L}^2(\mathbb{R})$. Biorthonormal wavelets often to be used in applications, because of a weaker condition and more flexible.

According to the definition of MRA, any function $f \in V_J$ can be represented in the form of

$$f(x) = \sum_{k\in Z} c_{J,k}\phi_{J,k}(x) \tag{7}$$

for some coefficients $\{(c_{J,k})\}$. On the other hand, by relation (6) we have another representation as

$$f(x) = \sum_{k\in Z} c_{J-1,k}\phi_{J-1,k}(x) + \sum_{k\in Z} d_{J-1,k}\psi_{J-1,k}(x). \qquad (8)$$

Later, representations (6) and (8) to be used in the model.

3. The Network Models

This paper considers a model of WNN based on MRA where the model output has the form

$$y_i = \sum c_{j,k}\phi_{j,k}(x_i),\ i = 1, \cdots, N \qquad (9)$$

where the basis function $\phi_{j,k}$'s are derived from scaling function ϕ, $\mathcal{O}_1^N := \{(x_i, y_i) : i = 1, \cdots, N\}$ is the training data set, j is the approximation level and $c_{j,k}$'s are network weights to be determined. Beside that, there may be another data set of pairs which is used to test or to validate . Graphically, this model is described by following figure:

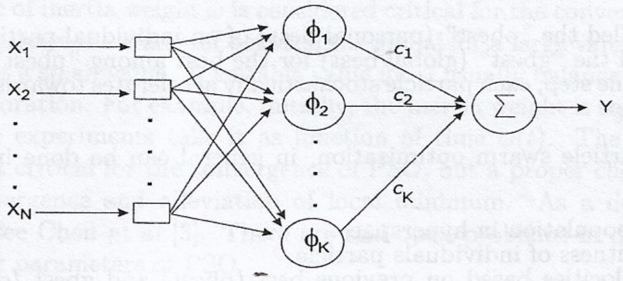

Figure 1. Model governed by the scaling function

Another model could be proposed here is based on (6) where both types bases are involved. This model has the form

$$y_i = \sum_k c_{j,k}\phi_{j,k}(x_i) + \sum_k d_{j,k}\psi_{j,k}(x_i),\ i = 1, \cdots, N \qquad (10)$$

Here, two groups of weights $\{c_{j,k}\}$ and $\{d_{j,k}\}$ are trained concurrently. Figure 2 shows the architecture of such model.

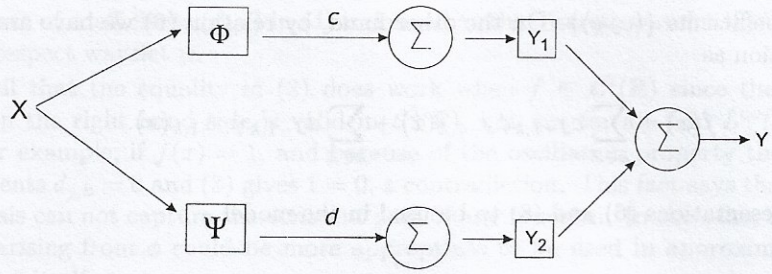

Figure 2. Model based on decomposition of MRA

4. Particle Swarm Optimization

Particle swarm optimization is a form of evolutionary computation technique developed by Kennedy and Eberhart [10]. It incorporates swarming behavior observed in flocks of birds, schools of fish, or swarm of bees, even human social behavior, from which the idea is arisen. A "swarm" is an apparently disorganized collection or population of moving individuals that tend to cluster together, while each individual seems to be moving in random direction.

Commonly, a features on particles swarm optimization is described as follows,

- Population is initialized by assigning random positions and velocities; particles representing the potential solutions are then flown through hyperspace as a search space.
- Each particle keeps track of its "best" (highest fitness) position in hyperspace.
- This is called the "pbest" (personal best) of an individual particle.
- It is called the "gbest" (global best) for the best among "pbest's".
- At each time step, each particle stochastically accelerates towards its "pbest" and "gbest".

The process of particle swarm optimization, in general can be done in following steps:

1. Initialize population in hyperspace.
2. Evaluate fitness of individuals particle.
3. Modify velocities based on previous best (pbest) and gbest (or lbest for neighborhood version of PSO), then update positions.
4. Terminate by some criteria, e.q. the number of iterations is reached.
5. Go to step 2.

Now, we are going to formalize PSO mathematically. Assume, a swarm of particles of size S fly through an $N-$ dimensional search space where each particle represents a potential solution to the optimization problem. Each particle a in swarm $\xi = (X_1, X_2, \cdots, X_a, \cdots X_S)$ is represented by the following characteristics:

$x_{a,j}(t)$: j^{th} component of the position of particle a, at time t.
$v_{a,j}(t)$: j^{th} component of the velocity of particle a, at time t.
$y_{a,j}(t)$: j^{th} component of the personal best(pbest)of particle a, at time t.
$\hat{y}_j(t)$: j^{th} component of the global best (gbest) position of swarm, at time t.

Let f be a *fitness* function to be optimized. WLOG, assume that the objective is to find the minimum of f in N-dimensional space. The personal best of particle a at time t, $y_{a,j}(t)$ is updated at time $t+1$ as

$$y_{a,j}(t+1) = \begin{cases} y_{a,j}(t) & \text{if } f(x_a(t+1)) > f(y_a(t)) \\ x_{a,j}(t+1) & \text{else} \end{cases} \tag{11}$$

for $j = 1, 2, \cdots, N$. Then, gbest is taken as

$$\hat{y}(t) = y_{gbest}(t) = \min\left(y_1(t), \cdots, y_S(t)\right).$$

In each iteration, positional updates are performed for each component $j = 1, \cdots, N$ and for each particle $a = 1, \cdots, S$ by

$$\begin{aligned} v_{a,j}(t+1) &= \omega v_{a,j}(t) + c_1 r_{1,j}(t)\Big(y_{a,j}(t) - x_{a,j}(t)\Big) \\ &\quad + c_2 r_{2,j}(t)\Big(\hat{y}_j(t) - x_{a,j}(t)\Big) \qquad (12) \\ x_{a,j}(t+1) &= x_{a,j}(t) + v_{a,j}(t+1). \qquad (13) \end{aligned}$$

where ω is called the inertia factor, r_1 and r_2 are random numbers which used to maintain the diversity of the population, and are uniformly distributed in the interval $[0, 1]$, c_1 is a positive constant, called as coefficient of the self-recognition component, c_2 is a positive constant, called as coefficient of the social component.

The main problem of application PSO in optimization is how to select the best parameters so that the iteration converges fastly. So far, the selection is determined by trial and error experimentations without scientific consideration. Abraham [1] stated the role of inertia weight ω is considered critical for the convergence behavior of PSO. It regulates the trade-off between the global for a large value, and the local exploration for a small value. A suitable value for ω usually balance between global and local exploration. For example, initially, the inertia weight is set as a constant, however some experiments take it as function of time $\omega(t)$. The parameters c_1 and c_2 are not critical for the convergence of PSO, but a proper choice may result in faster convergence and alleviation of local minimum. As a default, usually, $c_1 = c_2 = 2$, see Chen at al [3]. There are still open problems in determining the best values for parameters of PSO.

Kennedy [11] also proposed another version for velocity updating as

$$v_{a,j}(t+1) = \chi\Big(\omega v_{a,j}(t) + c_1 r_{1,j}(t)(y_{a,j}(t) - x_{a,j}(t)) + c_2 r_{2,j}(t)(\hat{y}_j(t) - x_{a,j}(t))\Big) \tag{14}$$

where parameter χ controls the magnitude of v, whereas the inertia ω weights the magnitude of the old velocity. At the end of this section, the pseudo-code for PSO algorithm is illustrated.

```
Initialize the size of the swarm, e.g. s, and other parameters.
Initialize the positions and velocities for all particles randomly.
While (the stopping criteria in not met) do
t = t + 1;
```

```
Calculate the fitness value of each particle;
x* = argmin{f(x*(x − 1), f(x_i(t)), i = 1, ··· S}
  For i = 1 : S
      x# = argmin{f(x#(x − 1), f(x_i(t)), i = 1, ··· S}
      For j = 1 : N
          Update the j-th dim value of v_i according to (12) or (14), then x_i
      Next j
  Next i
End While.
```

5. Computation Procedure and Numerical Experiments

5.1 Bases on input domain

Let $\mathcal{O}_1^N := \{(x_i, y_i) : i = 1, \cdots, N\}$ be a given training data set, and assume that for each i, x_i lies in a closed interval $[x_a, x_b]$. The minimal such interval $[x_a, x_b]$ to be called domain of input. Moreover, let $\phi_{j,k}$ defined by

$$\phi_{j,k}(x) := 2^{j/2}\phi(2^j x - k)$$

be wavelets bases generated by a scaling function ϕ. Since the scaling function must be compactly-supported, it is enough to assume that supp $\phi = [a, b]$. Now, we want to determine which bases contributing on domain $[x_a, x_b]$.

Theorem 1. *Let the domain of training data input be included in some closed interval $[x_a, x_b]$ and support of the scaling function ϕ is $[a, b]$. Then, for fixed level j the bases $\phi_{j,k}$ contributing to the domain are corresponding to $k \in \{k_{min}, \cdots, k_{max}\}$ where*

$$k_{min} = \lfloor 2^j x_a - b + 1 \rfloor, \quad \text{and } k_{max} = \lceil 2^j x_b - a - 1 \rceil$$

where notations $\lfloor\ \rfloor$ and $\lceil\ \rceil$ symbolize to flooring and ceiling functions, respectively. Consequently, the number of hidden nodes will be $k_{max} - k_{min} + 1$.

Proof. Since supp $\phi = [a, b]$ then supp $\phi_{j,k} = [\frac{a+k}{2^j}, \frac{b+k}{2^j}]$. In order that support intersects domain, the following situations must be satisfied

$$\frac{b + k_{min}}{2^j} > x_a \text{ and } \frac{a + k_{max}}{2^j} < x_b.$$

Thus, k_{min} is the first integer that greater than $2^j x_a - b$. On account of flooring function $\lfloor x \rfloor$ as the greatest integer n which is less than or equal to x, then the first assertion is proven. The second proof is similar. □

5.2 The fitness function

The fitness function in the neural network is used to measure the quality of learning process. Commonly, the root mean square error (RMSE) is taken as fitness, i.e.

$$RMSE := \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

where y_i is the output of training data and $\hat{y}_i$ is the model output (estimator). According to model (9), it is needed to evaluate the value of basis functions

$$\hat{y}_i = \sum_{k=k_{min}}^{k_{max}} c_{j,k}\phi_{j,k}(x_i), \; i = 1, \cdots, N.$$

In order to keep computation efficiently, it is recommended to write the basis functions evaluation in the matrix-vector form as

$$\hat{Y} = \Phi \, \mathbf{c}$$

where $\hat{Y} := (\hat{y}_i)^T_{i=1,\cdots,N}$, $\Phi := (\phi_{j,k}(x_i))^{k=k_{min},\cdots,k_{max}}_{i=1,\cdots,N}$ and c: $= (c_{j,k})^T_{k=k_{min},\cdots,k_{max}}$. Furthermore, the mass matrix Φ could be stored as pre-determined. A similar procedure when model 2 is taken.

5.3 Examples

For numerical implementation, we have two examples of nonlinear function approximation,

$$y = \frac{(x-2)(2x-1)}{1+x^2}, \; x \in [-8, 12]. \qquad \text{(ex.1)}$$

This example was taken by Sun at all [15] using RBFNN and PSO approach. Next example is

$$y = \sin(3x)\cos(5(x-0.5)), \; x \in [-1, 1]. \qquad \text{(ex.2)}$$

This example was implemented by Kobayashi [12] using WNN with Gaussian wavelet and SERWANN algorithm.

On this paper we use model 1 and PSO of version (14) with parameters follow Chen at al [3], i.e. $\chi = 0.8; w = 0.7; c_1 = 2; c_2 = 2$. The scaling function to be used in implementation arises from B-spline family defined recursively as

$$\phi_N(x) = \frac{x}{N}\phi_{N-1}(x) + \frac{N+1-x}{N}\phi_{N-1}(x-1)$$

where $\phi_0(x) := \chi_{[0,1)}(x)$ is a characteristic function on $[0, 1)$, see Chui [5] for detail or Hernadi [9] for closed-form definition. The level approximation j affects on the number of hidden nodes of the network. Theoretically, the more hidden nodes the more accurately but the less efficiency. Thus, j should be taken as small as possible, but still gives an accurate result. Here, the experiment results.

Table 1. Summary of experiment results

	Example 1	Example 2
Scaling function	quadratic B-spline	cubic B-spline
Approximation level	$j = 0$	$j = 3$
Number of hidden nodes	22	19
Swarm size	100	50
Number of inputs	50 randomly	25 randomly
RMSE	0.005	0.004
Number of iterations	200	276

This results is much better than Sun at al [15] which only gives RMSE = 0.0332 with 33 hidden nodes. Figure 3 shows the result of method.

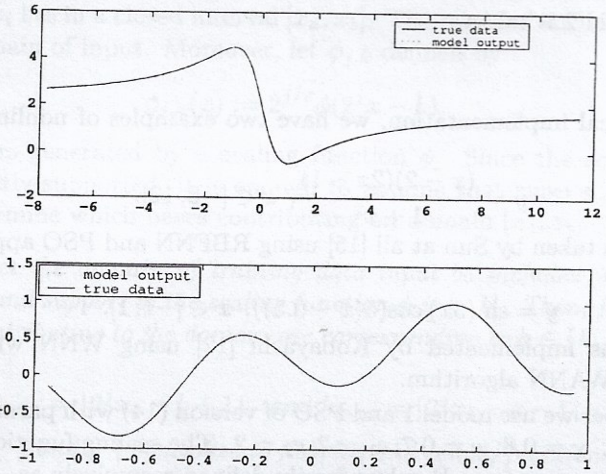

Figure 3. Results of method: example 1 (above) and example 2 (bottom)

As view in figure, the model approximates data accurately, not only for training data but also for test data. For instance, in example 1, the 75 test data was taken randomly on interval $[-8, 12]$. As a result, RMSE for this data test is 0.0158.

Intuitively, the larger size of swarm the faster convergence of iteration. According to the numerical experiments this is only taken place in the first few iterations as be shown in the figure 4.

Why this phenomenon could be happened, let's recall to the iteration (12), (13) and (14). If the velocities are very close to zero, then all particles will stop moving once they catch up with the global best particle, which lead to premature convergence, and no further improvement. This stagnation phenomenon is a weakness of PSO iteration and will be regarded as an open problem.

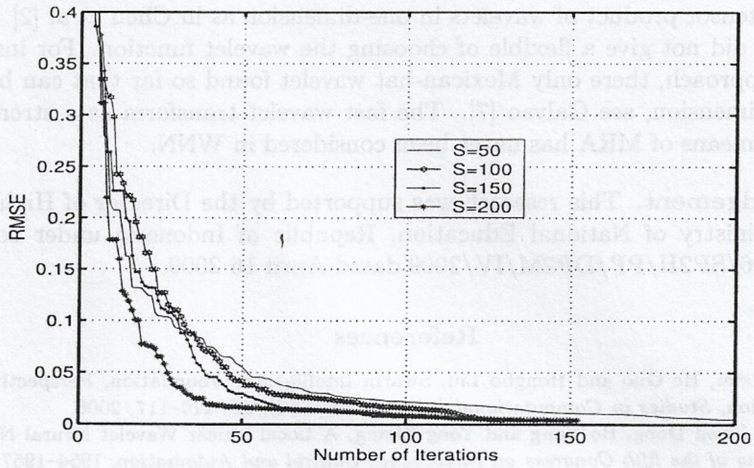

Figure 4. Convergence rate of several swarm sizes

6. Concluding Remarks and Discussion

This paper has proposed a different kind of WNN where scaling function was used in the model, instead of wavelet as it has been considered by several authors. Training using PSO algorithm gave satisfactory results, even better than previous method.

So far, only model 1 has been implemented with B-spline scaling functions. Model 2 could be more attractive because it combines both scaling function and wavelet. It is a strength of MRA theory that the scaling function is used to approximate, while the wavelet to capture the detail or residual in consequence of approximation. The application of other type of wavelets bases should be tried, e.g. Daubechies's wavelets could be interesting one because they have many good approximation properties such as regularity, orthonormality, vanishing moment and narrow support.

Since the wavelets basis on MRA usually of compactly supported, it is a good idea to take network weights depend on input data. The main reason is there will be only a few input data included in any support of basis function so that the corresponding weight only depend on those input data, hence efficiently is kept. This idea has been done by Chen at al [2, 3] but in the form of linearly depending and the support was not compact so that all inputs contribute to each weight.

Many challenging issues corresponding to this method is concerned with study of PSO algorithm, in particular choosing the best parameters to be applied in WNN. A collaboration between PSO and GA may be considered as the method for improving of PSO, in particular to cope the stagnation phenomenon as mentioned above. An extension to higher dimension is still to be a problem in the wavelet neural network. So far, this problem is just coped by a radial approach as in Galvao

[7] or by a tensor product of wavelets in one-dimension as in Chen at al [2]. These approaches did not give a flexible of choosing the wavelet function. For instance, by radial approach, there only Mexican-hat wavelet found so far that can be used in higher dimension, see Galvao [7]. The fast wavelet transform as a strength of wavelet by means of MRA has never been considered in WNN.

Acknowledgement. This research was supported by the Director of Higher Education, Ministry of National Education, Republic of Indonesia under contract number: 106/SP2H/PP/DP2M/IV/2009 dated April 16 2009.

References

[1] A. Abraham, He Guo and Hongbo Liu, Swarm Intelligence: Foundation, Perspectives and Application, *Studies in Computational Intelligence (SCI)*, **26**, 110–117, 2006.

[2] Y.Chen, Jiwen Dong, Bo Yang and Yong Zhang, A Local Linear Wavelet Neural Network, *Proceeding of the fifth Congress on Intelligence Control and Automation*, 1954–1957, 2004.

[3] Y.Chen, Bo Yang and Jiwen Dong, Time-series Prediction using a local linear wavelet neural network, *Neurocomputing*, **69**, 449–465, 2006.

[4] M. Clerc, *Particle Swarm Optimization*, ISTE Ltd., London, 2006.

[5] Charles K. Chui, *Wavelets: A Mathematical Tool for Signal Analysis*, SIAM, Philadelphia, 1997.

[6] I. Daubechies, *Ten Lectures on Wavelets*, Academic press, New York, 1992.

[7] Roberto K.H. Galvao, Victor M. Becerra, Joao M.F.Calado and Pedro M. Silva, Linear Wavelet Network, *Int. J. Appl. Math. Comput. Sci.*, **24**, 221–232, 2004.

[8] Venu G. Gudise and Ganesh K. Venayagamoorty, Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Network, *Proceeding of IEEE Swarm Intelligence Symposium*, 110–117, 2003.

[9] J. Hernadi, *Wavelet Projection Methods for Solving Operator Equations*, Dissertation, Gadjah Mada University, 2004.

[10] J. Kennedy and R.C. Eberhart, Particle Swarm Optimization, *Proc. IEEE Int. Conf. Neural Network IV*, 1954–1957, 1995.

[11] J. Kennedy, Small World and Mega-Mind: effects of Neighborhood Topology on Particle Swarm Optimization, *Proc. of the 1999 Congress of Evolutionary Computation*,3, 1931–1938, 1999.

[12] K. Kobayashi and T. Torioka, A Wavelet Neural Network for Function Approximation and Network Optimization , *Proceeding of ANNIE 94*, **4**, 505–510, 1994.

[13] S. Mallat, Multiresolution Approximation and Wavelet Orthonormal Bases of $\mathcal{L}^2(\mathbb{R})$, *Trans. Amer. Math. Soc.*, **315**, 69–88, 1989.

[14] S. Mallat, *A Wavelet Tour of Signal Processing, second edition*, Academic Press, San Diego, 1999.

[15] T-Y Sun, Chan-Cheng Liu, Chun-Ling Li, Sheng-Ta Hsieh and Cheng-Sen Huang, A Radial Basis Function Neural Network with Adaptive Structure via Particle Swarm Optimization in A. LAZINICA(ed), *Particle Swarm Optimization* , 423–436, 2009.

[16] Y.Y. Tang, L.H. Yang, J. Liu, and H.Ma, *Wavelet Theory and Its Application to Pattern Recognition*, World Scientific, Singapore, 2000.

[17] Q. Zhang and Benveniste, Wavelet Neural Network, *IEEE Trans. Neural Network*, **3**, 889–896, 1992.

[18] Q. Zhang, *Using Wavelet Network in Nonparametric Estimation*, Technical Report, **2321**, IRISA, Campus Universitaire de Beaulieu, France, 1994.

JULAN HERNADI: Department of Mathematics Education, Muhammadiyah University of Ponorogo, East Java, Indonesia. E-mail: julan_hernadi@yahoo.com