Proc. 5th IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI 2009). Thessaloniki, Greece (2009)

Model Identification in Wavelet Neural Networks Framework

A. Zapranis[1], A. Alexandridis[2]

Department of Accounting and Finance, University of Macedonia of Economics and Social Studies, 156 Egnatia St., P.O. 54006, Thessaloniki, Greece.

Abstract — The scope of this study is to present a complete statistical framework for model identification of wavelet neural networks (WN). In each step in WN construction we test various methods already proposed in literature. In the first part we compare four different methods for the initialization and construction of the WN. Next various information criteria as well as sampling techniques proposed in previous works were compared in order to derive an algorithm for selecting the correct topology of a WN. Finally, In variable significance testing the performance of various sensitivity and model-fitness criteria were examined and an algorithm for selecting the significant explanatory variables is presented.

1. Introduction

This study presents a complete statistical wavelet neural network (WN) model identification framework. Model identification can be separated in two parts, model selection and variable significance testing. Wavelet analysis (WA) has proved to be a valuable tool for analyzing a wide range of time-series and has already been used with success in image processing, signal de-noising, density estimation, signal and image compression and time-scale decomposition.

In [1] have demonstrated that it is possible to construct a theoretical description of feedforward NN in terms of wavelet decompositions. WN were proposed by [2] as an alternative to feedforward NN hoping to elevate the weakness of each method. The WN is a generalization of radial bases function networks (RBFN).WNs are one hidden layer networks that use a wavelet as an activation function instead of the classic sigmoid function. The families of multidimensional wavelets preserve the universal approximation property that characterizes neural networks. In

[1] Phone: +30 2310 891690, Fax: +30 2310 891689. e-mail: zapranis@uom.gr.
[2] Corresponding Author. Phone: +30 2310 891631, e-mail: aalex@uom.gr

[3] various reasons presented in why wavelets should be used instead of other transfer functions.

Wavelet networks have been used in a variety of applications so far. Wavelet networks were used with great success in short term load forecasting, [4], in time series prediction, [5], signal classification and compression, [6], static, dynamic [1] and nonlinear modeling [7], nonlinear static function approximation, [8]. Finally, [9] proposed WN as a multivariate calibration method for simultaneous determination of test samples of copper, iron, and aluminum.

In contrast to sigmoid neural networks, wavelet networks allow constructive procedures that efficiently initialize the parameters of the network. Using wavelet decomposition a wavelet library can be constructed. Each wavelon can be constructed using the best wavelet of the wavelet library. These procedures allow the wavelet network to converge to a global minimum of the cost function. Also starting the network training very close to the solution leads to smaller training times. Finally, wavelet networks provide information of the participation of each wavelon to the approximation and the dynamics of the generating process.

The rest of the paper is organized as follows. In section 2 we present the WN, we describe the structure of a WN and we compare different initialization methods. In section 3 we present a statistical framework in WN model selection and different methods are compared. Various sensitivity criteria of the input variables are presented in section 4 and a variable selection scheme is presented. Finally, in section 5 we conclude.

2. Wavelet Neural Networks for Multivariate Process Modeling

In [10] and [11] we give a concise treatment of wavelet theory. Here the emphasis is in presenting the theory and mathematics of wavelet neural networks. So far in literature various structures of a WN have been proposed [5] [8] [9] [7] [12] [13]. In this study we use a multidimensional wavelet neural network with a linear connection of the wavelons to the output. Moreover in order for the model to perform well in linear cases we use direct connections from the input layer to the output layer. A network with zero hidden units (HU) is the linear model.

The network output is given by the following expression:

$$\hat{y}(\mathbf{x}) = w_{\lambda+1}^{[2]} + \sum_{j=1}^{\lambda} w_j^{[2]} \cdot \Psi_j(\mathbf{x}) + \sum_{i=1}^{m} w_i^{[0]} \cdot x_i$$

In that expression, $\Psi_j(\mathbf{x})$ is a multidimensional wavelet which is constructed by the product of m scalar wavelets, $\mathbf{x}$ is the input vector, m is the number of network inputs, λ is the number of hidden units and w stands for a network weight. Following [14] we use as a mother wavelet the Mexican Hat function. The multidimensional wavelets are computed as follows:

$$\Psi_j(\mathbf{x}) = \prod_{i=1}^{m} \psi(z_{ij})$$

where ψ is the mother wavelet and

$$z_{ij} = \frac{x_i - w_{(\xi)ij}^{[1]}}{w_{(\zeta)ij}^{[1]}}$$

In the above expression, $i = 1, \ldots, m$, $j = 1, \ldots, \lambda+1$ and the weights w correspond to the translation ($w_{(\xi)ij}^{[1]}$) and the dilation ($w_{(\zeta)ij}^{[1]}$) factors. The complete vector of the network parameters comprises:

$$w = \left(w_i^{[0]}, w_j^{[2]}, w_{\lambda+1}^{[2]}, w_{(\xi)ij}^{[1]}, w_{(\zeta)ij}^{[1]} \right)$$

There are several approaches to train a WN. In our implementation we have used ordinary back-propagation which is less fast but also less prone to sensitivity to initial conditions than higher order alternatives. The weights $w_i^{[0]}$, $w_j^{[2]}$ and parameters $w_{(\xi)ij}^{[1]}$ and $w_{(\zeta)ij}^{[1]}$ are trained for approximating the target function.

In WN, in contrast to NN that use sigmoid functions, selecting initial values of the dilation and translation parameters randomly may not be suitable, [15]. A wavelet is a waveform of effectively limited duration that has an average value of zero and localized properties hence a random initialization may lead to wavelons with a value of zero. Also random initialization affects the speed of training and may lead to a local minimum of the loss function, [16]. In [2] the wavelons are initialized at the center of the input dimension of each input vector x_i.

The initialization of the direct connections $w_i^{[0]}$ and the weights $w_j^{[2]}$ is less important and they are initialized in small random values between 0 and 1.

The previous heuristic method is simple but not efficient. As it is shown in figure 2 the initial approximation is a bad approximation of the function *f(x)*. The heuristic method does not guarantee that the training will find the global minimum. Moreover this method does not use any information that the wavelet decomposition can provide. In literature more complex initialization methods have been proposed, [17] [14] [18]. All methods can be summed in the following three steps.

1. Construct a library *W* of wavelets
2. Remove the wavelets that their support does not contain any sample points of the training data.
3. Rank the remaining wavelets and select the best regressors.

The wavelet library can be constructed either by an orthogonal wavelet or a wavelet frame. However orthogonal wavelets cannot be expressed in closed form.

It is shown that a family of compactly supported non-orthogonal wavelets is more appropriate for function approximation, [19]. The wavelet library may contain a large number of wavelets. In practice it is impossible to count infinite frame or basis terms. However arbitrary truncations may lead to large errors, [20].

In [14] three alternative methods were proposed in order to reduce and rank the wavelet in the wavelet library namely the Residual Based Selection (RBS) a Stepwise Selection by Orthogonalization (SSO) and a Backward Elimination (BE) algorithm. In [21] the RBS algorithm is used for the synthesis of a WN while in [17] an algorithm similar to SSO is proposed. In [18] an orthogonalized residual based selection (ORBS) algorithm is proposed for the initialization of the WN.

All the above methods are used just for the initialization of the dilation and translation parameters. Then the network is further trained in order to obtain the vector of the parameters $w = w_0$ which minimizes the cost function.

The heuristic, the SSO, the RBS and the BE methods that constitute the bases for alternative algorithms and can be used with the batch training algorithm will be tested. We test these methods in two examples. The first example where the underlying function *f(x)* is:

$$f(x) = 0.5 + 0.4\sin(2\pi x) + \varepsilon_1(x) \quad x \in [0,1]$$

where x is equally spaced in [0,1] and the noise $\varepsilon_1(x)$ follows a normal distribution with mean zero and a decreasing variance:

$$\sigma_\varepsilon^2(x) = 0.05^2 + 0.1(1 - x^2)$$

Figure 1 show the initialization of all four algorithms for the first example. The network uses 2 hidden units with learning rate 0.1 and momentum 0. The use of a large learning rate or momentum might lead to oscillation between two points. As a result the WN would not be able to find the minimum of the loss function or it will be trapped in a local minimum of the loss function. It is clear that the BE and SSO algorithms starting approximation are very close to the target function *f(x)*. As a result less iterations and training time are needed. In order to compare the previous methods we use the heuristic method to train 100 networks with different initial conditions of the direct connections $w_i^{[0]}$ and weights $w_j^{[2]}$ to find the global minimum. We find that the smallest mean square error (MSE) is 0.031332. Using the RBS algorithm the MSE is 0.031438 and is found after 717 iterations. The MSE between the underlying function *f(x)* and the network approximation is 0.000676. The SSO needs 4 iterations and the MSE is 0.031332 while the MSE between the underlying function *f(x)* and the network approximation is only 0.000121. The same results achieved by the BE method. Finally, one implementation of the heuristic method needed 1501 iterations.

From the previous examples it seems the SSO and the BE algorithms give the same results and outperform both the heuristic and the RBS algorithm. In order to have a more clear view we introduce a more complex example where

$$g(x) = 0.5x\sin(x) + \cos^2(x) + \varepsilon_2(x) \;\; x \in [-6,6]$$

and $\varepsilon_2(x)$ follows a Cauchy distribution with location 0 and scale 0.05 and x is equally spaced in [-6,6]. While the fist example is very simple the second one proposed by [22] incorporates large outliers in the output space. The sensitive to the presence of outliers of the proposed WN will be tested.

The results for the second example are similar however the BE algorithm is 10% faster than the SSO. Using the RBS, SSO and BE algorithms the MSE is 0.004758, 004392 and 0.04395 and is found after 2763, 1851 and 1597 iterations respectively. The MSE between the underlying function $g(x)$ and the network approximation is 0.000609, 0.000073 and 0.000057 respectively.

One can observe in Figure 2 that the WN approximation was not affected by the presence of large outliers in contrast to the findings of [22]. In this study 8 hidden units were used for the network topology proposed by *v-fold* cross-validation while in [22] 10 hidden units were proposed by the FPE criterion. As it is shown in the next section the FPE criterion does not perform as well as sampling techniques and should not be used.

The previous examples indicate that SSO and BE perform similarly whereas BE outperforms SSO in complex problems. On the other hand BE needs the calculation of the inverse of the wavelet matrix which columns might be linear dependent, [14]. In that case the SSO must be used. However since the wavelets come from a wavelet frame this is very rare to happen, [14].

3. Model Selection

In this section we describe the model selection procedure. One of the most crucial steps is to identify the correct topology of the network. A network with less HU than needed is not able to learn the underlying function while selecting more HU than needed will result to an overfitting model. Several criteria exist for model selection, such as Generalized Prediction Error, Akaike's Information Criterion, Final Prediction Error (FPE), Network Information Criterion and Generalized Cross-Validation (GCV). These criteria are based on assumptions that are not necessarily true in the neural network framework. Alternatively we suggest the use of sampling methods such as bootstrap and cross-validation. The only assumption made by sampling methods is that the data are a sequence of independent and identically distributed variables. However, sampling methods are computationally very demanding. In this study we will test the FPE proposed by [14], the GCV proposed by [14], the bootstrap (BS) and the *v-fold* cross-validation (CV) methods proposed by [23] and [24]. These criteria will be tested with and without training of the network.

In both examples BS, FPE and CV propose similar models. In the first example 2 HU were needed to model the underlying function $f(x)$.On the other hand GCV suggests 3 hidden units. The MSE between the underlying function $f(x)$ and the

approximation of the WN using 3 HU is 0.000271 while using 2 HU is only 0.000121 indicating that the GCV suggested a more complex model than needed. In the second example BS and CV propose the same network topology (8 HU) while using the FPE criterion the prediction risk minimized in 7 HU and using the GCV criterion it is minimized in 14 HU. In order to compare the performance of each criterion the MSE between the underlying function *g(x)* and the approximation of the WN is calculated. The MSE is 0.000079, 0.000073 and 0.000101 for 7, 8 and 14 HU. Again the BS and CV gave correct results while the FPE performs satisfactorily.

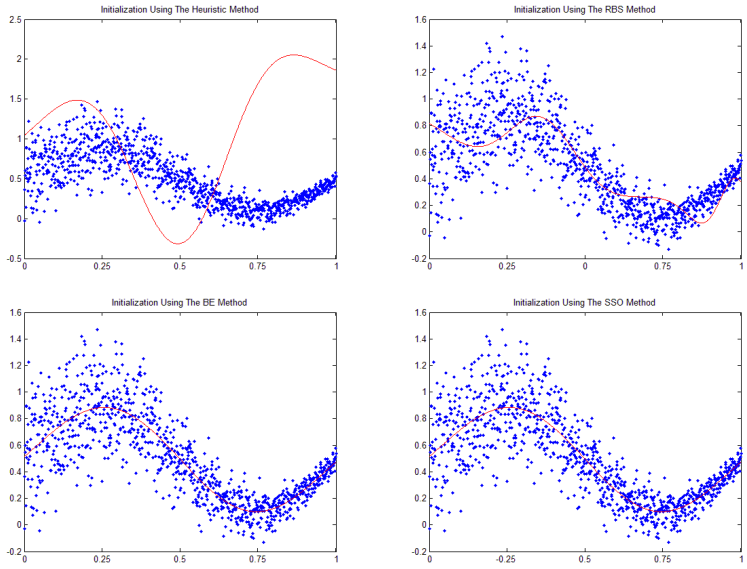

Fig. 1. Four different initialization methods.

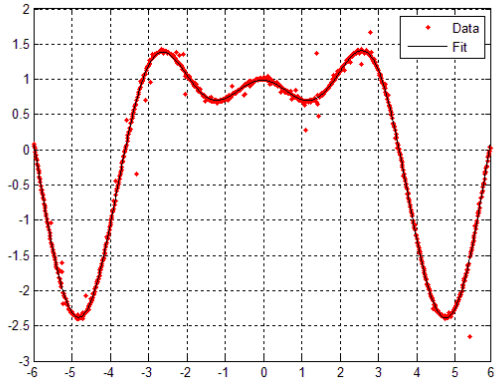

Fig. 2. Data and WN approximation using 8 hidden units.

In order to significantly reduce the training times [14] propose that since the initialization is very close to the underlying function the prediction risk can be calculated after the initialization. In the first example all information criteria gave the same results as in the previous case. However in the second example in all criteria more than 14 HU were needed proving that early stopping techniques does not perform satisfactory.

Since sampling techniques are very computationally expensive the FPE criterion can be used initially. Then BS or CV can be used in +/-5 HU around the HU proposed by FPE in order to define the best network topology.

4. Model fitness and sensitivity criteria.

In real problems it is important to define correctly the independent variables. In most problems there is a little information about the relationship of any explanatory variable with the dependent variable. As a result unnecessary independent variables included in the model reduce the predictive power of the model. In this section we will present eight different sensitivity criteria and one model fitness sensitivity (MFS) criterion for testing the significance of each explanatory variable.

First we create a second variable X_2 which was randomly drawn from the uniform distribution within the range (0,1). To fit the sample for the first example we use a WN with both X_1 and X_2 as inputs. Using the CV the prediction risk is minimized when 3 HU are used and it is 0.04194. The network approximation converges after 3502 iterations. Comparing the results with the findings in previous section it is clear that including an irrelevant variable to our model increases the training time while the predictive power of the model is reduced. Hence an algorithm that correctly identifies the insignificant variables is needed. For analytical expressions of each criterion we refer to [24].

In linear models the significance of each explanatory variable is determined by the value of the coefficient. In the WN case by observing the weights of the direct connections one concludes that X_2 is more significant than X_1. As expected the listed magnitudes are much larger for the first variable for all nine criteria. However the only information that Table 1 gives is how sensitive is the dependent variable to each independent variable. There is no information if X_2 should be removed from the model. In [24] a novel approach (parametric sampling) is presented in order to determine if a variable should be removed from the model. In parametric sampling new networks are created by bootstrapping the parameters of the initial network. In order to reduce training times [24] use local bootstrap. Wavelets are local function and local bootstrapping may cannot be used. Hence we sample from the training patterns. As *v-fold* cross validation performs better than bootstrap [23] we propose an approach where 50 new training samples are created according to *v-fold* cross validation. After the correct topology of the network is determined, the sensitivity criteria are calculated for each sample. Next the *p-values* for each criterion are computed and the variable with the largest *p-value*

is removed. The procedure is repeated until all explanatory variables have *p*-value less than 0.1 indicating significant variables.

First the standard deviation and the *p*-values for all sensitivity and model fitness measures for the two variables of the first example are calculated. As it was expected X_1 has a larger impact in the output *y*. However all eight sensitivity measures consider both variables as significant predictors. As discussed on [24] these criteria are application dependent while MFS criteria are much better suited for testing the significance of the explanatory variables. Indeed the *p*-value for X_2 using the SBP is 0.6019 indicating that this variable must be removed from the model. In the reduced model the *p*-value for X_1 using the SBP is 0 indicating that X_1 is very significant. Next the correctness of removing the X_2 should be tested. The prediction risk in the reduced model was reduced to 0.0396 from 0.0419 in full model. Moreover the adjusted R^2 increased to 70.8% from 69.7%.

The same analysis is repeated for the second example. In Table 2 the mean, the standard deviation and the *p*-values for all sensitivity and model fitness measures for the two variables of the second example are presented. A network with 10 HU was needed when both variables were included in the model. Only three criteria suggest that X_2 should be removed from the model, the SBP, the MaxDM and MinDM with *p*-values 0.1597, 0.4158 and 0.8433 respectively. However the MinDM wrongly suggests that X_1 should also be removed from the model with *p*-value 0.1795 in the reduced model. On the other hand the p-values for X_1 using the SBP and the MaxDM are 0 indicating a very significant variable in both full and reduced models. The reduced model needed only 8 HU and the prediction risk reduced to 0.0008 from 0.0033 that it was when X_2 was included as an input. Moreover the adjusted R^2 increased to 99.7% from 99.2%. The previous examples show that SBP can be safely used for the identification of irrelevant variables. On the other hand the sensitivity criteria are application dependent and extra care must be taken when used.

5. Conclusions.

This study presents a statistical framework for wavelet network model identification. To our knowledge this is the first time that a complete statistical framework for the use of WNs is presented. Several methodologies were tested in wavelet network construction, initialization, model selection and variable significant testing. We propose a multidimensional wavelet neural network with a linear connection of the wavelons to the output and direct connections from the input layer to the output layer. The training is performed by the classic back-propagation algorithm. Next four different methods were tested in wavelet network initialization. Using the BE and SSO the training times were reduced significantly while the network converged to the global minimum of the loss function.

Model selection is a very important step. Four techniques were tested with the sampling techniques to give more stable results than other alternatives. BS and CV found the correct network topology in both examples. Although FPE and GCV are

extensively used in the WN framework, due to the linear relation of the wavelets and the original signal, it was proved that both criteria should not be used in complex problems. Moreover using early stopping techniques in complex problems was found to be inappropriate.

A variable selection method was presented. Various sensitivity and model fitness criteria were tested. While sensitivity criteria are application dependent, MFS criteria are much better suited for testing the significance of the explanatory variables. The SBP correctly indentified the insignificant variables while their removal reduced the prediction risk and increased the adjusted R^2 implying the correctness of this decision.

Finally the partial derivatives with respect to the weights of the network, to the dilation and translation parameters as well as the derivative with respect to each input variable are presented. The construction of confidence and prediction intervals as well as a model adequacy testing scheme are left as a future work.

Table 1. Sensitivity measures for the first example.

	$w_i^{[0]}$	MaxD	MinD	MaxDM	MinDM	AvgD	AvgDM	AvgL	AvgLM	SBP
Full model										
(two variables)										
X_1	0.0161	1.3962	-1.3459	1.3962	0.0005	-0.0529	0.6739	0.2127	1.6323	0.0953
X_2	0.0186	0.4964	-0.7590	0.7590	0.0002	0.0256	0.0915	0.0781	0.1953	0.0001
Reduced model										
(one variable)										
X_1	0.1296	1.1646	-1.1622	1.1644	0.0014	0.0841	0.7686	0.3165	1.3510	0.0970

MaxD=Maximum Derivative
MinD=Minimum Derivative
MaxDM=Maximum Derivative Magnitude
MinDM=Minimum Derivative Magnitude
AvgD=Average Derivative
AvgDM=Average Derivative Magnitude
AvgL=Average Elasticity
AvgLM=Average Elasticity Magnitude
SBP=Sensitivity Based Pruning

References

1. Pati, Y., Krishnaprasad, P.: Analysis and Synthesis of Feedforward Neural Networks Using Discrete Affine Wavelet Transforms. IEEE Trans. on Neural Networks 4(1), 73-85 (1993)
2. Zhang, Q., Benveniste, A.: Wavelet Networks. IEEE Trans. Neural Networks 3(6), 889-898 (1992)
3. Bernard, C., Mallat, S., Slotine, J.-J.: Wavelet Interpolation Networks. In the proc. of ESANN '98, 47-52 (1998)

4. Benaouda, D., Murtagh, G., Starck, J.-L., Renaud, O.: Wavelet-Based Nonlinear Multiscale Decomposition Model for Electricity Load Forecasting. Neurocomputing 70, 139-154 (2006)
5. Chen, Y., Yang, B., Dong, J.: Time-Series Prediction Using a Local Linear Wavelet Neural Wavelet. Neurocomputing 69, 449-465 (2006)
6. Kadambe, S., Srinivasan, P.: Adaptive Wavelets for Signal Classification and Compression. International Journal of Electronics and Communications 60, 45-55 (2006)
7. Billings, S., Wei, H.-L.: A New Class of Wavelet Networks for Nonlinear System Identification. IEEE Trans. on Neural Networks 16(4), 862-874 (2005)
8. Jiao, L., Pan, J., Fang, Y.: Multiwavelet Neural Network and Its Approximation Properties. IEEE Trans. on Neural Networks 12(5), 1060-1066 (2001)
9. Khayamian, T., Ensafi, A., Tabaraki, R., Esteki, M.: Principal Component-Wavelet Networks as a New Multivariate Calibration Model. Analytical Letters 38(9), 1447-1489 (2005)
10. Zapranis, A., Alexandridis, A.: Modelling Temperature Time Dependent Speed of Mean Reversion in the Context of Weather Derivetive Pricing. Applied Mathematical Finance 15(4), 355 - 386 (2008)
11. Zapranis, A., Alexandridis, A.: Weather Derivatives Pricing: Modelling the Seasonal Residuals Variance of an Ornstein-Uhlenbeck Temperature Process With Neural Networks. Neurocomputing(accepted, to appear) (2007)
12. Becerikli, Y.: On Three Intelligent Systems: Dynamic Neural, Fuzzy and Wavelet Networks for Training Trajectory. Neural Computation and Applications 13, 339-351 (2004)
13. Zhao, J., Chen, B., Shen, J.: Multidimensional Non-Orthogonal Wavelet-Sigmoid Basis Function Neurla Network for Dynamic Process Fault Diagnosis. Computers and Chemical Engineering 23, 83-92 (1998)
14. Zhang, Q.: Using Wavelet Network in Nonparametric Estimation. IEEE Trans. Neural Networks 8(2), 227-236 (1997)
15. Oussar, Y., Rivals, I., Presonnaz, L., Dreyfus, G.: Trainning Wavelet Networks for Nonlinear Dynamic Input Output Modelling. Neurocomputing 20, 173-188 (1998)
16. Postalcioglu, S., Becerikli, Y.: Wavelet Networks for Nonlinear System Modelling. Neural Computing & Applications 16, 434-441 (2007)
17. Oussar, Y., Dreyfus, G.: Initialization by Selection for Wavelet Network Training. Neurocomputing 34, 131-143 (2000)
18. Xu, J., Ho, D.: A Basis Selection Algorithm for Wavelet Neural Networks. Neurocomputing 48, 681-689 (2002)
19. Gao, R., Tsoukalas, H.: Neural-wavelet Methodology for Load Forecasting. Journal of Intelligent & Robotic Systems 31, 149-157 (2001)
20. Xu, J., Ho, D.: A Constructive Algorithm for Wavelet Neural Networks. Lecture Notes in Computer Science(3610), 730-739 (2005)
21. Kan, K.-C., Wong, K.: Self-construction algorithm for synthesis of wavelet networks. Electronic Letters 34, 1953-1955 (1998)
22. Li, S., Chen, S.-C.: Function Approximation using Robust Wavelet Neural Networks. In the proc. of ICTAI '02, 483-488 (2002)
23. Efron, B., Tibshirani, R.: An Introduction to the Bootstrap. Chapman & Hall, USA (1993)
24. Zapranis, A., Refenes, A.: Principles of Neural Model Indentification, Selection and Adequacy: With Applications to Financial Econometrics. Springer-Verlag (1999)