INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Regressor Selection and Wavelet Network Construction

Qinghua Zhang

N° 1967

April, 1993

PROGRAMME 5

Traitement du signal,
automatique
et productique

ISSN 0249-6399

INRIA
RENNES

Regressor Selection and Wavelet Network Construction*

Qinghua Zhang

Programme 5 — Traitement du signal, automatique et productique
Projet AS

Rapport de recherche n° 1967 — April, 1993 — 21 pages

Abstract: The wavelet network [22, 23] has been introduced as a special feedforward neural network supported by the wavelet theory. Such network can be directly used in function approximation problems, and consequently can be applied to nonlinear system modeling by means of nonlinear black-box identification. In this paper the construction of feedforward neural networks is discussed from both identification and regressor selection points of view. This reveals that the wavelet network structure is well suited for developing constructive methods for feedforward networks. An efficient initialization procedure of the wavelet network based on the orthogonal least squares (OLS) method is then proposed. The efficiency of the wavelet network and the proposed procedure for nonlinear system modeling is illustrated by a numerical example.

Key-words: neural network, wavelet transform, nonlinear system identification, function approximation.

(Résumé : tsvp)

*This study was performed during the author's visiting year in the Department of Electrical Engineering, Linköping University, Sweden.

Unité de recherche INRIA Rennes
IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex (France)
Téléphone : (33) 99 84 71 00 – Télécopie : (33) 99 38 38 32

Sélection des Regresseurs et Construction du Réseau d'Ondelettes

Résumé : Le réseau d'ondelettes [22, 23] a été introduit comme un réseau de neurone spécial supporté par la théorie des ondelettes. Il peut être directement utilisé pour approcher des fonctions, et par conséquent appliqué à la modélisation de systèmes non linéaires par identification de type boîte noire non linéaire. Dans ce document la construction des réseaux de neurones dits "feedforward" est discutée des points de vue de l'identification et de la sélection de regresseurs. Ceci révèle que la structure du réseau d'ondelettes est bien adaptée pour développer des méthodes constructives des réseaux "feedforward". En suite une procédure basée sur la méthode des moindres carrées orthogonales est présentée. La performance du réseau d'ondelettes ainsi que la procédure d'initialisation proposée est illustrée par un exemple numérique pour la modélisation de systèmes non linéaires.

Mots-clé : réseau de neurones, transformée en ondelettes, identification de systèmes non linéaires, approximation.

1 Introduction

Neural networks have increasingly received considerable attentions in various areas such as signal processing, pattern recognition and automatic control. One of the attractive topics concerning neural networks is using *multi-layer feedforward networks* to approximate static mappings [17, 20]. By performing such approximations the feedforward neural networks are explored as a *nonlinear black-box* tool to model static nonlinear systems, and hopefully, to model dynamic nonlinear systems [15] by means of identification. Among feedforward networks, the most often studied networks are constructed with *sigmoid neurons*. Although such networks have demonstrated some empirical successes, very few useful theoretical results have been known. So far the most important theoretical result on this subject is, probably, that any continuous function can be approximated in its finite definition domain with an arbitrary accuracy, provided the number of *hidden neurons* is sufficiently large. This is true even for networks with only one hidden layer [5, 11]. However, this theoretical result cannot satisfactorily explain the empirical successes of the feedforward neural networks for approximation problems. On the other hand, it does not tell us, even not suggest us, how to construct the neural network given an approximation problem. In other words, it is only an existence theorem. The lack of theoretical result leads to the lack of efficient constructive method. The commonly used *back-propagation procedure* [9] for neural network training is a gradient descent method which minimizes the square error between the output of the network and the desired output. Before the back-propagation procedure is run, selecting the network structure (the number of layers and the number of neurons in each layer) is a delicate problem for which no useful theoretical indication is available. Further more, there is another difficulty even after the network structure is selected, that is less often discussed in the literature. This difficulty is the initialization of the network. Usually the networks are randomly or mostly randomly initialized[21], this is because there is no useful theoretical indication on this subject, and very few intuition suggests the way in which neurons should be organized in a network. Because the output of such network is highly nonlinear in its parameters, the gradient descent minimization of the square error combined with random initializations may be stuck to bad local minima, or the initial parameter of the network may be quite far from the nearest "good"

minimum. Consequently such gradient based high dimensional minimization procedures are usually very slow.

It is obvious that some more theoretical investigations are needed in order to get some solutions of the above described problems and to have a deeper understanding of the functioning of the neural networks. For this purpose, the *wavelet theory* [13, 14, 8, 6] has been found to be a useful tool. Several authors have independently realized the possible connection between the feedforward neural networks and the wavelet theory. See, for example, [22, 23, 16, 10]. Previously reported works on this subject have already introduced the concept of *wavelet networks* [22, 23] which are feedforward neural networks consisting of wavelets and make the connection between the feedforward neural networks and the wavelet theory. The theorems on the approximation ability of the wavelets [6, 23] are better than that for the "classical" feedforward neural networks in a sense that they explain better how the wavelets (neurons) work. By selecting wavelets with good *spatio-spectral* (or *time-frequency* in the terminology of signal processing) localization properties as neurons, we know that each neuron (or wavelon) contributes to the desired global approximation locally in both spatial and spectral domains. And further more, this understanding inspired some efficient initialization procedure of the network [23]. This results in more constructive methods of the feedforward network.

In this paper, first the construction of the feedforward neural network is discussed from both identification and regressor selection points of view. It will conclude that the wavelet network structure is well suited for developing more constructive methods to build feedforward network for approximation and modeling problems. Then as a result of this discussion, the *orthogonal least squares* (OLS) method [3] is used to develop a new initialization procedure of the wavelet network.

2 Feedforward neural network construction: different points of view

In this section, the construction of feedforward neural networks is investigated from both identification and regressor selection points of view. This investigation will reveal that the *wavelet network* [22, 23] which results from

a combination of the feedforward neural network and the wavelet theory is a suitable network structure for developing constructive methods to build feedforward network for function approximation and system modeling problems.

2.1 The identification point of view

A multilayer feedforward neural network is a nonlinear parametric model. Generally the modeling with neural networks consists of two steps: determining the network structure (the number of layers and the number of neurons in each layer) and adjusting the parameters of the network of the determined structure. Let us focus for the moment our attention on the second step. Assume that, for a given feedforward network structure, the network can be expressed by the following equation

$$z = g(\theta_1, \theta_2, \ldots, \theta_{N'}; x) \tag{1}$$

where $x \in \mathbb{R}^n$ is the input of the network, $z \in \mathbb{R}$ is the output, $\theta_1, \theta_2, \ldots, \theta_{N'} \in \mathbb{R}^m$ are parameters of the neurons[1], N' is the number of neurons constituting the network, and g represents the mapping realized by the network which is determined by the network structure and parameterized by the θ's. Usually the parameter vector θ_i $(1 \leq i \leq N')$ of each neuron can take any real values, thus equation (1) defines a class of mappings or models with all the possible values of the parameters, whenever the network structure (the form of g) is determined. Assume that a set of *training patterns* is given, that is, a set of input and desired output pairs

$$T = \{(x_k, y_k) : x_k \in \mathbb{R}^n, y_k \in \mathbb{R}, k = 1, 2, \ldots, K\} \tag{2}$$

The data vectors x_k in T are often referred to as *input patterns*, and y_k *output patterns*. The problem of *training the network* is to adjust the parameters of the network to minimize some cost function (usually the square error between the output of the network and the desired output). This is a typical nonlinear identification problem. The usual method to do this in the domain of neural networks is the back-propagation (gradient descent) procedure following some random initialization as mentioned in the introduction section. Such methods

[1] Assume that all the neurons in the net have the same number of parameters. In fact in many cases the neurons of a net are identical except the values of their parameters and their "topological" positions in the net.

are usually not efficient because of the high nonlinearity of the output of the network in its parameters and the large number of parameters to be adjusted.

2.2 The regressor selection point of view

Now we investigate the neural network construction problem from another point of view. For the sake of simplicity, we consider only networks with one hidden layer. The discussion of this subsection can be straightforwardly generalized into more general case of multi-layer feedforward network, though only the result on single hidden layer network will be used for the construction of the wavelet network.

Assume that the neurons in the hidden layer are all identical except the values of their parameters. Therefore, these hidden neurons (or equivalently, their parameters) are mutually exchangeable. In other words, if we express the single hidden layer network by the following formula

$$z = g(\theta_1, \theta_2, \ldots, \theta_N, \xi; x) \tag{3}$$

where $\theta_1, \theta_2, \ldots, \theta_N$ are the parameter vectors of the hidden neurons, N is the number of hidden neurons, and ξ represents the rest of the parameters of the network, then the order of the θ's is meaningless. The commonly studied single hidden layer network has the following form

$$z = \sum_{i=1}^{N} w_i h_{\theta_i}(x) \tag{4}$$

where $h_{\theta_i}(\cdot)$ represents the hidden neuron parameterized by θ_i, and $w = (w_1, \ldots, w_N)^T$ is usually considered as the parameter vector of the *output neuron*. Because this output neuron performs only a linear combination of the outputs of the *hidden neurons* (it does not have a nonlinear activation function), it is less like a neuron than the others. Therefore, we would rather call w the *linear combination weights* of the (hidden) neurons or *linear weights* for short. Note that h represents the *form* or *structure* of the hidden neurons parameterized by $\theta_i \in \mathbb{R}^m$.

At this stage it should be remarked that equation (4) can be considered as a linear regression if the parameter θ's are fixed. In this case one neuron $h_{\theta_i}(x)$ is nothing but a variable in the regression equation that is also referred to as a *regressor*.

Assume that the form of neurons h is chosen, and let us consider the family of all possible neurons h_θ parameterized by $\theta \in \mathbb{R}^m$

$$\{h_\theta : \theta \in \mathbb{R}^m\}. \tag{5}$$

Given a set of training patterns T as defined in formula (2), and assume that the number of (hidden) neurons N is already (somehow) chosen, then the problem of constructing a neural net of form (4) with N neurons for fitting the training patterns T can be solved by selecting N neurons from family (5) according to some optimal criterion and then determining the linear weights w. If we consider family (5) as a set of possible regressors, then the above problem can be interpreted as the selection of "best" regressors in regressions (4). This is a well known subject in nonparametric statistics (see, e.g., [7]). This is an alternative of the identification point of view discussed in the previous subsection. We are going to show that this point of view will provide a useful insight and lead to some efficient network construction method.

In order to be able to select the "optimal" neurons from family (5) for a some given training data, we have to first define the meaning of the "optimality". As for the problem of regressor selection, the most computationally convenient way for doing this is to use the least squares criterion on the fitting errors. However, the problem is not easy to solve if we want to get a globally optimal solution. For instance, if h_{θ_1} is optimally selected for $N = 1$, usually the optimal $h_{\theta'_1}, h_{\theta'_2}$ for $N = 2$ do not include the previously selected h_{θ_1}. This is because the selections of $h_{\theta'_1}$ and $h_{\theta'_2}$ are not independent. Therefore, for a chosen N, all the N neurons should be simultaneously selected from family (5). It turns out that the difficulty is the same as that from the identification point of view.

Facing to this difficulty, we have to content ourselves with some sub-optimal solution. The first problem is that we are trying to select regressors from the *continuously* parameterized family (5). Indeed, even for $N = 1$, this is a difficult non convex minimization problem. If we discretize the parameter θ of family (5), the situation may be improved by selecting regressors from a finite or countable family. Thus the question is how to reasonably discretize family (5) in order to get a good enough sub-optimal solution of the original problem. In fact this idea has been implicitly used in works for constructing radial basis function (RBF) networks [18, 4]. In the construction of radial basis function networks the selection of the centers of the radial basis functions

is a difficult problem and can be considered as the selection of radial basis functions from a continuously parameterized family. In practice these centers are usually selected within the input patters of the training data (x_k's in T). It is not guaranteed that the selection within such a subset of the radial basis function family can give the global optimal solution, but it is believed that this constrained selection can result in reasonable sub-optimal solutions.

Consider again the question how to reasonably discretize family (5). The answer depends, of course, on the structure of the network, more precisely, on the form of the neuron h_θ. For the most commonly used network structure with sigmoid neurons [5, 1, 12], we have no idea about this. The existence theorem of the approximation ability of the sigmoid neural networks [5, 11] does not inspire anything on this subject. For the radial basis function networks, the situation is better. The input patters of the training data are usually used as the candidates of the centers of the radial basis functions, but it is not clear how to choose the radius (or scale) parameter of the radial basis functions.

A more suitable network structure should be supported by some better established theory. The *wavelet transform* and *wavelet decomposition* [13, 14, 8, 6] offer such a theoretical background. The wavelet family used in the *continuous wavelet transform* is related to the continuously parameterized neuron family (5), and the *discrete wavelet transform* is related to the discrete family of neurons if we use wavelets as neurons and choose the network structure similar to that of the discrete wavelet decomposition. This connection between the feedforward neural network and the wavelet theory has inspired the development of the *wavelet network* [22, 23]. In section 3 we briefly recall the wavelet network structures introduced in [22, 23].

Before finishing this section, we should remark that the strategy of regressor selection based on a discrete family of neurons can only lead to a sub-optimal solution of the training data fitting problem. We can further, however, consider thus obtained solution as the start point of a training procedure. In other words, we only consider the regressor selection procedure as an initialization method of the network before training procedures such as the back-propagation procedure are run. Such a strategy has proved to be more efficient than the strategy of random initialization + gradient descent training. In section 4 we present such an initialization procedure based on the regressor selection using orthogonal least squares method.

3 Wavelet network structures

Wavelet networks are feedforward neural networks with only one hidden layer consisting of wavelets. In this section we briefly recall the wavelet network structures introduced in [22, 23]. The reader may refer to these two references for more details.

Let us start with one dimensional input networks, that is, networks with their inputs $x \in \mathbb{R}$. In this case, the wavelet network structure is expressed by the following formula

$$z = \sum_{i=1}^{N} w_i \psi(d_i(x - t_i)), \quad w_i, d_i, t_i \in \mathbb{R} \tag{6}$$

where $\psi(\cdot)$ is the *mother wavelet function*, $d_i \in \mathbb{R}$ are *dilation parameters*, t_i are *translation parameters*, w_i are *linear weights* and N is the number of wavelets. This is a single hidden layer feedforward network as a particular case of network (4). Here a hidden neuron is a dilated and translated wavelet. This network structure was directly inspired by the one dimensional discrete wavelet decomposition which has the following form

$$f(x) = \sum_{s \in \mathbb{Z}_+, r \in \mathbb{Z}} \mathcal{W}(s, r) \alpha^{-\frac{s}{2}} \psi(\alpha^{-s} x - \beta r) \tag{7}$$

where $\mathcal{W}(s, r)$ is the *discrete wavelet transform* [2] of $f(x)$, and $\alpha, \beta \in \mathbb{R}_+$ are dilation and translation step sizes respectively. If the wavelet family

$$\{\psi_{s,r} = \alpha^{-\frac{s}{2}} \psi(\alpha^{-s} x - \beta r)) : s, r \in \mathbb{Z}\} \tag{8}$$

constitutes an orthonormal basis of $L^2(\mathbb{R})$, then selecting elements of this family to construct wavelet network of form (6) will obviously be able to approximate any function of $L^2(\mathbb{R})$. More generally family (8) may constitute a *frame* of $L^2(\mathbb{R})$ instead of a basis. In this case family (8) is redundant to span the $L^2(\mathbb{R})$ space. It is more convenient in practice to use a redundant wavelet family than an orthonormal wavelet basis for constructing the

[2] Usually the wavelet transform of a function is defined as the inner product of the function with elements of the wavelet family. If the discrete wavelet family is not an orthonormal basis of L^2 space, the noted $\mathcal{W}(s, r)$ in this formula is not such defined wavelet transform, but can be computed from the latter.

wavelet network, because nobody has found any easy analytical form of wavelet function which has good spatio-spectral localization properties and can generate orthonormal basis of L^2. The investigations in [6] give sufficient conditions for family (8) to constitute frames of $L^2(\mathbb{R})$. The theorem given in [23] generalizes this result to the general n-dimensional case. Therefore, wavelet networks can be used for approximating all functions in L^2. In addition to this justification of the approximation ability , if the chosen wavelet function ψ is well concentrated in both spatial and spectral domains, then each dilated and translated wavelet contributes to the global approximation realized by network (6) locally in both spatial and spectral domains. This offers us a deep insight into how each wavelet in a network works. This also inspires us how to construct the network. The wavelet network initialization procedure presented in [23] and another one presented in section 4 of this paper are results of this inspiration.

Formula (6) defines the *one dimensional wavelet net* which can only approximate functions of $L^2(\mathbb{R})$ space. It must be generalized to multi-dimensional case in order to approximate functions in $L^2(\mathbb{R}^n)$. For this purpose in [22] the direct product form was used to construct multi-variable wavelets from single-variable ones. This resulted in the *direct product wavelet network*. Because the nonlinear computation operated in each wavelet of direct product form is proportional to the dimension n, it is much more expensive to construct high dimensional direct product wavelet net than the classical sigmoid neural net for which the nonlinear computation (sigmoid function) is independent of the dimension. Therefore the direct product wavelet net introduced in [22] is not suitable for approximation of functions of large number of variables. The extension of approximation (6) with radial wavelets introduced in [23] is an alternative of the direct product wavelet net and has a computational complexity similar to that of the classical sigmoid neuron for any dimension case. Therefore, the radial wavelet network is better suited for approximation problems of large dimensions.

In order to easily capture linear properties of nonlinear approximation problems, in [23] some additional terms were introduced to the network. This resulted in the following network structure

$$g(x) = \sum_{i=1}^{N} w_i \psi(\mathrm{diag}(d_i)(x - t_i)) + c^T x + b \tag{9}$$

where ψ is a radial wavelet function, $d_i \in \mathbb{R}^n$ are dilation parameters, $t_i \in \mathbb{R}^n$ are translation parameters, $w_i \in \mathbb{R}$ are linear weights, N is the number of wavelets, $c \in \mathbb{R}^n$ is the additional *direct linear combination parameters* (also called *direct connection parameters*), and $b \in \mathbb{R}$ is the *bias parameter*.

In the sequel we only discuss this type of wavelet network.

4 Initialization by orthogonal least squares method

As we mentioned in the end of section 2, the initialization of the network can be performed in a regressor selection manner. In this section, first we choose the set of regressor within which the selection will be performed for initializing the radial wavelet network. Then the orthogonal least squares method is described in detail for performing this selection.

4.1 The set of candidate wavelets

If we consider the wavelet network (9) as a linear regression, the discrete wavelet decomposition suggests us to choose as the set of possible regressors (also referred to as *the set of candidate wavelets* here) the following wavelet family

$$\{\psi_{s,r}(x) = \alpha^{-\frac{1}{2}ns}\psi(\alpha^{-s}x - \beta r) : \ s \in \mathbb{Z}, r \in \mathbb{Z}^n\} \tag{10}$$

where $\psi(x)$ is the mother wavelet, $x \in \mathbb{R}^n$, $\alpha, \beta \in \mathbb{R}_+$. The theorem given in [23] states that this family can constitute *frames* of $\mathrm{L}^2(\mathbb{R}^n)$ for properly chosen ψ, α and β. Selecting a finite number of wavelets of family (10) to construct (or initialize) a wavelet network is similar to keeping a finite number of terms in the discrete wavelet decomposition to approximate the decomposed function. Selecting wavelets from the countable set (10) is generally a difficult problem. we first truncate it into a finite set of wavelets. This needs, of course, a priori knowledge on the given approximation or modeling problem. The same truncation is needed for discrete wavelet decomposition in practice. Roughly speaking, this truncation should only keep those wavelets of family (10) with their "supports"[3] falling inside the domain of interest of

[3] strictly speaking, the chosen wavelet function may infinitely spread but rapidly vanish. In this case, the term "support" here should be understood in an approximate manner.

the given approximation problem and with their dilation parameters corresponding to the desired resolution level. Usually the truncated family is in the form of a regular pyramid[4] that we denote by

$$\{\psi_{s,r}(x) = \alpha^{-\frac{1}{2}ns}\psi(\alpha^{-s}x - \beta r) : \ s \in S, r \in R\} \tag{11}$$

with some chosen $S \subset \mathbb{Z}$ and $R \subset \mathbb{Z}^n$.

4.2 Refinement of the set of candidate wavelets

The truncated set (11) usually has the form of a regular pyramid, and the number of wavelets in this set increases approximately exponentially as the dimension n increases. Therefore, the number of candidate wavelets chosen in this way for the initialization procedure may be very large. In practice, for some approximation or modeling problems, especially for those of large dimensions, the available training patterns are not uniformly distributed in the domain of interest of the approximation problem. In family (11) the wavelets of different scale levels are uniformly distributed in the domain of interest of the approximation. It turns out that some wavelets in set (11) may have no or very few training patterns falling in their "supports"[5]. Surely these wavelets are useless for fitting the training patterns. Therefore, these wavelets should be first eliminated from set (11). This may considerably reduce the number of wavelets as candidates for the wavelet network construction.

One way for this elimination is to count, for each wavelet in set (11), the number of training patterns falling in its "support" and to compare this number to a threshold. Because the number of wavelets in set (11) may be very large, this wavelet oriented sorting may be very slow. We can avoid this difficulty by approximately taking hyper-cubes cut along the axes of $\mathbb{R}^n$ space as the "supports" of the wavelets and by employing a training pattern oriented strategy. In this way, by examining the value x_i of each training pattern, we can determine the wavelets whose "supports" contain this training pattern. If the number of training patterns is much smaller than the number of wavelets in set (11), this method is much more efficient.

[4]By "pyramid" we mean a set of wavelets in which the wavelets with the same value of dilation parameter are uniformly translated in the domain of interest, and lager the dilation parameter is, denser the wavelets of the corresponding scale are.

[5]See footnote 3.

In the sequel we refer to the thus refined set as the *set of candidate wavelets* and denote it by

$$\{\psi_{\theta_i}(x) : i = 1, 2, \ldots, M\} \tag{12}$$

where θ_i represents both dilation and translation parameters, and M is the number of wavelets in the set. Note that usually discrete wavelet families are indexed by the dilation and translation parameters, here we use only one (arbitrarily defined) index i for the convenience of presentation.

4.3 The OLS procedure for wavelet selection

The next question is then how to select wavelets from the candidate wavelet set (12). Given N the number of wavelets to be selected, in principle this selection can be performed by examining all the subsets of N elements of set (12). Because the number of these subsets is a combination number $\binom{M}{N}$ which is generally very large, in practice such an exhaustive examination is computationally very expensive. Therefore we have to make a trade-off between the optimality and the efficiency of the methods of selection. In principle this is a regressor selection problem. The classical methods for regressor selection should be helpful, but they are not directly applicable because of the particularities of the wavelet network: large number of regressor candidates and the possibility to use a training procedure after the selection of the wavelets.

One method was proposed in [23] which uses a backward elimination strategy. With that method the wavelet network is first constructed with all the wavelets in the set of candidate wavelets, then the *least contributive* wavelet in a sense for fitting the training data is removed from the net, and then this procedure is repeated to remove more wavelets until some stop criterion affects. A practically easy to use method was developed under some assumptions and approximations to determine the least contributive wavelet. However, this method, though efficient for some tested examples, still has some drawbacks. First it assumes that the training patterns (observations on the function to be approximated) are reasonably uniformly distributed and dense enough. This requirement may not be satisfied in some applications. The second drawback is that it starts by constructing the network with all the candidate wavelets and then uses an iterative procedure to remove the least contributive wavelets. Because the number of wavelets in the candidate set

may be quite large compared to the number of wavelets needed to construct the final network, this elimination strategy is computationally expensive.

The initialization procedure we present in this subsection will remedy these two drawbacks. Assume that we want to select N wavelets from set (12) for constructing the wavelet network. Unformally speaking, first we select one wavelet which is "optimal" (for fitting the given training patterns) for the case $N = 1$, then we select the second one such that it is "optimal" while working with the previously selected one, then the third one is similarly selected, and so on. With such a stepwise or incremental strategy, the selected N wavelets ($N > 1$) is not guaranteed to be "optimal" among all the subsets of size N, but experiments have shown quite satisfactory results.

It is convenient to use the orthogonal least squares (OLS) algorithm for implementing such a selection procedure. The OLS algorithm has been used by Chen et al. [3, 4] for regressor selection problems. In [4] the OLS algorithm is used for selecting the centers of radial basis functions for constructing radial basis function (RBF) networks. Here we closely follow this method, but we can do better with the wavelet network. For the RBF network, taking the input patters in the training data set as the candidates of the centers of the radial basis functions is purely intuitive and the choice of the radius or (scale) parameter of the radial basis functions is based on experiences. In contrast, the set of candidate wavelets (11) was suggested by the discrete wavelet decomposition and ensures a "uniform" distribution of the candidate wavelets in both spatial and spectral domains. In the following we describe in detail the procedure based on the OLS algorithm for initializing the wavelet network.

The wavelet network constructed with N wavelets has the form (for the moment we omit the direct linear connections and the bias)

$$z = \sum_{i=1}^{N} w_{l_i} \psi_{\theta_{l_i}}(x) \tag{13}$$

where $\{l_1, l_2, \ldots, l_N\}$ is a subset of $\{1, 2, \ldots, M\}$. Assume further that we have a set of training data T as defined in formula (2), then we can write

$$y = \Psi w + e \tag{14}$$

where

$$\begin{aligned}
\Psi &= \begin{bmatrix} \psi_{\theta_{l_1}}(x_1) & \cdots & \psi_{\theta_{l_N}}(x_1) \\ \vdots & \vdots & \vdots \\ \psi_{\theta_{l_1}}(x_K) & \cdots & \psi_{\theta_{l_N}}(x_K) \end{bmatrix} \\
w &= (w_{l_1} w_{l_2} \dots w_{l_N})^T \\
y &= (y_1, y_2, \dots, y_K)^T \\
e &= (e_1 e_2 \dots e_K)^T.
\end{aligned}$$

e is the residuals or errors of the approximation problem defined by the training data T. The goal of the approximation is to minimize the sum of square errors

$$C \triangleq e^T e = \sum_{k=1}^{K} e_k^2. \tag{15}$$

The linear weights w can obviously be determined by the least squares method. Now the question is how to select N wavelets in set (12) such that C is minimized.

For the convenience of notation, we define

$$p_i \triangleq [\psi_{\theta_i}(x_1) \dots \psi_{\theta_i}(x_K)]^T$$

for $i = 1, 2, \dots, M$. Then we can write

$$\Psi = [p_{l_1} p_{l_2} \dots p_{l_N}]. \tag{16}$$

Generally the vectors $p_1, p_2, \dots, p_M$ are not mutually orthogonal. the Gram-Schmidt method [2] can be applied to decompose Ψ as

$$\Psi = QA$$

where Q is a $K \times N$ matrix with orthogonal columns and A is a $N \times N$ triangle matrix with 1's on the diagonal and 0's below the diagonal, that is,

$$A = \begin{bmatrix} 1 & \alpha_{12} & \alpha_{13} & \cdots & & \alpha_{1N} \\ 0 & 1 & \alpha_{23} & \cdots & & \alpha_{2N} \\ 0 & 0 & 1 & \cdots & & \alpha_{3N} \\ \vdots & \vdots & \ddots & \ddots & & \vdots \\ 0 & 0 & \cdots & 0 & 1 & \alpha_{N-1N} \\ 0 & 0 & \cdots & & 0 & 1 \end{bmatrix}.$$

Then equation (14) can be rewritten as

$$y = Qv + e \tag{17}$$

and the vector $v = Aw$ can be determined by the least squares solution

$$v = (Q^T Q)^{-1} Q^T y. \tag{18}$$

Let us denote

$$Q = [q_1 q_2 \dots q_N]$$

with q_i columns of Q. Because the columns of Q are orthogonal, equation (18) becomes

$$\begin{aligned} v &= (v_1 v_2 \dots v_N)^T \\ v_i &= q_i^T y/(q_i^T q_i), \quad i = 1, 2, \dots, N \end{aligned}$$

and it turns out that

$$y^T y = \sum_{i=1}^{N} v_i^2 q_i^T q_i + e^T e \tag{19}$$

The value of $y^T y$ is determined by the set of training data, thus minimizing $C = e^T e$ is maximizing the sum $\sum_{i=1}^{N} v_i^2 q_i^T q_i$. Note that each term of the sum $\sum_{i=1}^{N} v_i^2 q_i^T q_i$ corresponds to one wavelet in network (13) or one column of the matrix Ψ. Therefore, if we want to select one by one N wavelets ($N \leq M$) from set (12) to construct network (13), the i-th selected wavelet should maximize the corresponding term $v_i^2 q_i^T q_i$.

Now we summarize this selection procedure using the Gram-Schmidt orthogonalization method as follows.

Step 1 For $i = 1, 2, \dots, M$ compute

$$\begin{aligned} q_1^{(i)} &= p_i \\ v_1^{(i)} &= (q_1^{(i)})^T y / \left((q_1^{(i)})^T q_1^{(i)} \right) \\ J_1^{(i)} &= (v_1^{(i)})^2 (q_1^{(i)})^T q_1^{(i)} \end{aligned}$$

and select

$$q_1 = q_1^{(i_1)}$$

such that

$$i_1 = \arg \max_{i=1,\ldots,M} J_1^{(i)}.$$

Note

$$J_1 = J_1^{(i_1)}.$$

Step k $(k \geq 2)$ For $1 \leq i \leq M$ and $i \neq i_1, \ldots, i \neq i_{k-1}$, compute

$$\begin{aligned} a_{jk}^{(i)} &= v_j^T p_i / (v_j^T v_j), \quad j = 1, 2, \ldots, k-1 \\ q_k^{(i)} &= p_i - \sum_{j=1}^{k-1} a_{jk}^{(i)} q_j \\ v_k^{(i)} &= (q_k^{(i)})^T y / \left((q_k^{(i)})^T q_k^{(i)} \right) \\ J_k^{(i)} &= (v_k^{(i)})^2 (q_k^{(i)})^T q_k^{(i)} \end{aligned}$$

and select

$$q_k = q_k^{(i_k)}$$

such that

$$i_k = \arg \max_i J_1^{(i)} \text{ with } 1 \leq i \leq M \text{ and } i \neq i_1, \ldots, i \neq i_{k-1}.$$

Note

$$J_k = J_k^{(i_k)}.$$

The space spanned by the selected $q_1, q_2, \ldots, q_N$ is the same space spanned by $p_{i_1}, p_{i_2}, \ldots, p_{i_N}$, therefore, the wavelets indexed by $i_1, i_2, \ldots, i_N$ in set (12) are selected.

This iterative procedure can be terminated at $k = N$ with N a prefixed number. The choice of N is a model order selection problem. It is possible to use the normalized square errors

$$c(k) = 1 - \frac{\sum_{i=1}^{k} J_i}{(y - \bar{y})^T (y - \bar{y})} \quad \text{with} \quad \bar{y} = \frac{1}{K} \sum_{j=1}^{K} y_j$$

to decide whether the iterative procedure should be stopped. While using this stop criterion, we should remember that this is only an initialization procedure, the error of the network may be further improved by a training procedure.

This selection procedure determines the dilation parameters d_i and the translation parameters t_i of the selected wavelets, meanwhile the linear weights w_i can also be computed by solving the triangle system of linear equations $Aw = v$.

The above procedure is for the wavelet network without the linear combination terms and the bias $c^T x + b$. In order to select these terms together with the wavelets, we define $n+1$ additional p_i's as follows.

$$[p_{M+1} p_{M+2} \cdots p_{M+n+1}] = [\mathbf{1}\ [x_1 x_2 \cdots x_K]^T]$$

where $\mathbf{1}$ is a column vector consisting of K 1's. Then in the above selection procedure, the range $1 \le i \le M$ is modified to $1 \le i \le M+n+1$.

5 Numerical example

The used mother wavelet function in our implementation is

$$\psi(x) = (x^T x - n) e^{-\frac{1}{2} x^T x}, \quad n = \dim(x).$$

The proposed method was applied to the identification of a robot arm to illustrate its efficiency. The arm is hydraulically controlled. What we want to model is the dynamic relationship between the position $u(t)$ of the valve through which the oil streams and the oil pressure $p(t)$. Both $u(t)$ and $p(t)$ are measured, and 1024 samples of the measurements are depicted in figure 1. It is assumed that the input-output relationship between $u(t)$ and $p(t)$ can be described by

$$p(t) = f(p(t-1), p(t-2), p(t-3), u(t-1), u(t-2)) + e(t) \qquad (20)$$

where $f(\cdot)$ is a unknown function of 5 variables and $e(t)$ represents the modeling error.

A wavelet network $g(\cdot)$ of 6 wavelets was used to approximate this unknown $f(\cdot)$. For this purpose the first 512 samples of the data shown in figure 1 were used as training data to construct the wavelet network, the rest of the data were used to validate the established model. To validate the model, we used the measured $u(t)$ in the second half of the data to simulate the output $p(t)$. More precisely, we generated a signal $\hat{p}(t)$ in the following way

$$\hat{p}(t) = g(\hat{p}(t-1), \hat{p}(t-2), \hat{p}(t-3), u(t-1), u(t-2)) \qquad (21)$$

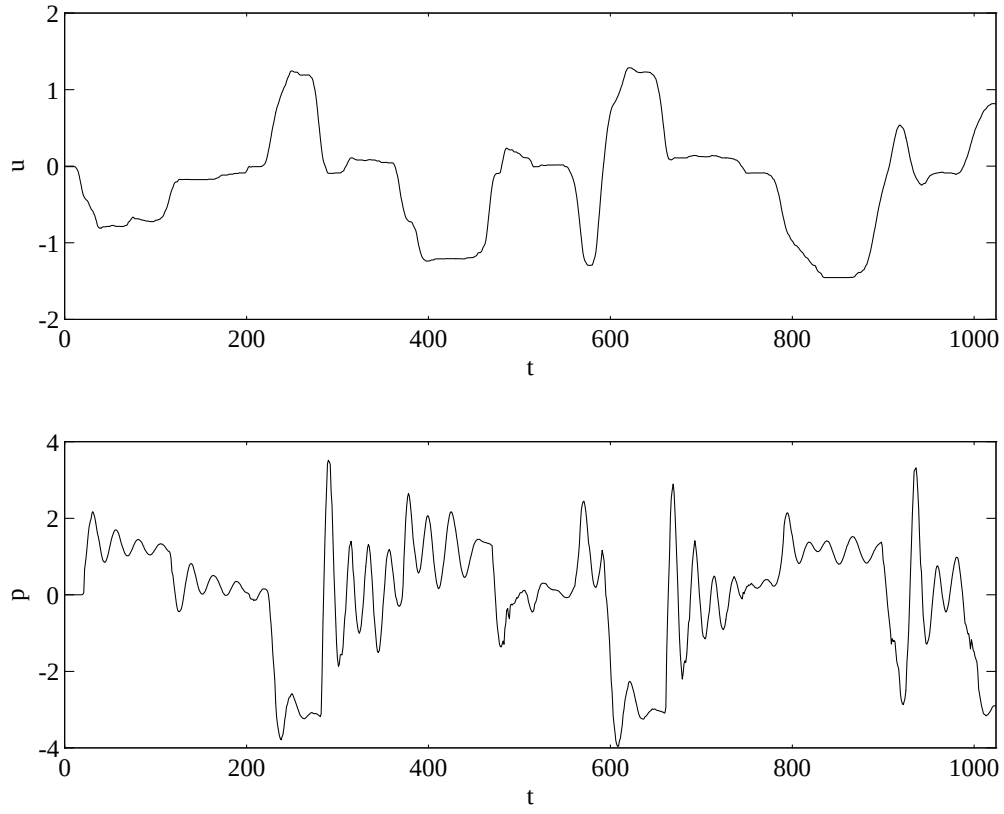


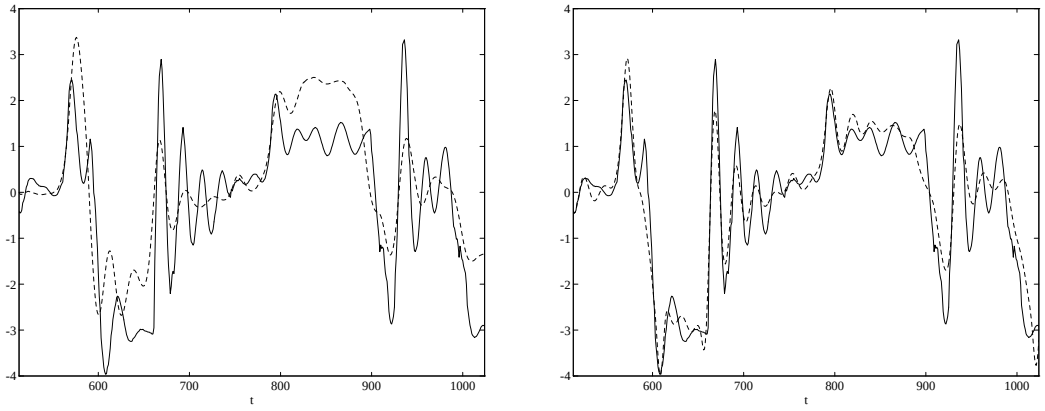
Figure 1: Measured data $u(t)$ and $p(t)$

where $g(\cdot)$ is the constructed wavelet network, and the first three values of $\widehat{p}(t)$ are set to the values of the corresponding $p(t)$. If the thus simulated $\widehat{p}(t)$ is closed to the corresponding real measurement $p(t)$, then the model can be considered as appropriate.

In order to compare the performance of the obtained model to that of linear models, we also tried to fit the same data with the following linear ARX model

$$p(t) = a_1 p(t-1) + a_2 p(t-2) + a_3 p(t-3) + b_1 u(t-1) + b_2 u(t-2)) + e'(t) \quad (22)$$

and do the same simulation with it.



simulation with the ARX model simulation with the wavelet network

The solid lines represent the real measurements and the dashed lines represent the results of simulation.

Figure 2: Simulated $\widehat{p}(t)$ compared to measured $p(t)$

The simulation results compared to the real measurements are shown in figure 2. The model with wavelet network performs effectively much better than the linear regression model. Note that the linear regression model has 5 parameters and the number of parameters of the used wavelet network composed of 6 wavelets is 72. The wavelet network was constructed with the OLS selection procedure and no training procedure was used in order to show the result of the proposed method. The selection procedure programmed in Matlab used less than 20 seconds on a workstation of Sun Sparc 2. Note that to get comparable results with traditional neural networks, the time needed for network training is in the order of minutes or even hours.

6 Conclusion

In this paper the construction of the feedforward neural networks was discussed from both identification and regressor selection points of view. This revealed that the wavelet network structure is well suited for developing constructive methods of neural networks. An initialization method based on

the OLS algorithm was proposed for constructing wavelet networks. Such constructions of the wavelet network are very different from the traditional constructions of neural networks by random initialization + training procedure. Experiments have shown the efficiency of the proposed method in nonlinear modeling.

Acknowledgment The author gratefully acknowledges Jonas Sjöberg and Svante Gunnarsson from Linköping University for providing the data of the robot arm.

References

[1] A. R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function", Tech. Rep. #58, Dept. of Statistics, Univ. of Illinois at Urbana-Champaign, 1991.

[2] A. Björck, "Solving linear least squares problems by Gram-Schmidt orthogonalization", *Nordisk Tidskr. Information-Be-Handling*, vol.7 pp. 1-21. 1967.

[3] S. Chen, S.A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification", *Int. J. Control*, Vol 50, No. 5, pp 1873-1896. 1989.

[4] S. Chen, C.F.N. Cowan and P.M. Grant "Orthogonal least squares learning algorithm for radial basis function networks", *IEEE Trans. on Neural Networks*, Vol. 2, No. 2 pp. 302-309. March 1991.

[5] G. Cybenko, "Approximation by superposition of a sigmoidal function", *Mathematics of control, signals and systems* (1989) 2:303-314.

[6] I. Daubechies, "Ten Lectures on Wavelets", SIAM, Philadelphia, Pennsylvania, 1992.

[7] N. Draper, H. Smith "Applied regression analysis, second edition", Wiley Series in Probability and Mathematical Statistics. 1981.

[8] A. Grossmann and J. Morlet, "Decomposition of Hardy functions into square integrable wavelets of constant shape", *SIAM J. Math. Anal.*, vol. 15, pp. 723–736, 1984.

[9] R. HECHT-NIELSEN, "Theory of the backpropagation neural network", *Proc. IJCNN*, Washington D.C., June 18-22, 1989, I-593.

[10] J. HONG, "Identification of stable systems by wavelet transform and artificial neural networks", Ph.D. thesis, University of Pittsburgh, Pittsburgh, PA. 1992.

[11] K. HORNIK, "Multilayer Feedforward networks are universal approximators", *Neural networks*, Vol. 2, 1989.

[12] L.K. JONES, "Constructive approximations for neural networks by sigmoidal functions", *Proceeding of the IEEE*, Vol. 78, No. 10, October 1990.

[13] S.G. MALLAT, "Multiresolution approximation and wavelets orthonormal bases of $\mathcal{L}^2(\Re)$", *Trans. Amer. Math. Soc.,* 315, No. 1 , 69-88, 1989.

[14] Y. MEYER, "Wavelets and operators", *Proceedings of the Special year in modern Analysis*, Urbana 1986/87, published by Cambridge University Press, 1989. See also Y. MEYER, *Ondelettes et Opérateurs*, Hermann, Paris, 1990.

[15] K.S. NARENDRA and K. PARTHASARTHY "Identification and control pf dynamical systems using neural networks", *IEEE Trans. on Neural Networks*, Vol. 1, No. 1, pp 4-27, March 1990.

[16] Y.C. PATI, "Wavelets and time-frequency methods in linear systems and neural networks", Ph.D. thesis, University of Maryland, College Park. 1992.

[17] T. POGGIO and F. GIROSI, "Networks for approximation and learning", *Proceeding of the IEEE*, Vol. 78, No. 9, September 1990.

[18] M.J.D. POWELL, "Radial basis function approximations to polynomials", Proc. 12th Biennial Numerical Analysis Conference, Dundee, pp. 223-241.

[19] J. SJÖBERG, L. LJUNG, "Overtraining, regularization, and searching for minimum in neural networks", Tech. Report LiTH-ISY-I-1297, Department of Electrical Engineering, Linköping University, Sweden. December 1991.

[20] E.D. SONTAG, "Feedforward nets for interpolation and classification", *J. Comp. Syst. Sci.* (1991 or 1992)

[21] L.F.A. WESSELS, E. BARNARD, "Avoiding false local minima by proper initialization of connections", *IEEE Trans. on Neural Networks*, Vol. 3, No. 6, pp.899-905. November 1992.

[22] Q. ZHANG and A. BENVENISTE, "Wavelet networks", *IEEE Trans. on Neural Networks*, Vol. 3, No. 6, pp.889-898. November 1992.

[23] Q. ZHANG, "Wavelet networks: the radial structure and an efficient initialization procedure", *Technical Report of Linköping University*, LiTH-ISY-I-1423. October 1992.

Unité de recherche INRIA Lorraine, Technôpole de Nancy-Brabois, Campus scientifique,
615 rue de Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, IRISA, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399