Elektrotehniški vestnik 70(4): 221–226, 2003
Electrotechnical Review, Ljubljana, Slovenija

Matrix Formulation of the Multilayered Perceptron with a Denoising Unit

Uroš Lotrič, Andrej Dobnikar

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Laboratorij za adaptivne sisteme in paralelno procesiranje, Tržaška cesta 25, Ljubljana, Slovenija
E-mail: uros.lotric@fri.uni-lj.si, andrej.dobnikar@fri.uni-lj.si

Abstract. A denoising unit, based on the wavelet multiresolution analysis, is integrated into the multilayered perceptron. It is shown that by its mathematical formulation the denoising unit is an analogy to the layers of the multilayered perceptron. Denoising is treated as a part of the model and the same cost function is used for setting all the free parameters, i. e., those of the denoising unit and those of the multilayered perceptron. It is illustrated that the proposed model outmatches the multilayered perceptron in noisy time-series prediction problems.

Key words: wavelet multiresolution analysis, neural networks, denoising

Matrični zapis večnivojskega perceptrona z enoto za glajenje

Povzetek. Odstranjevanje šuma na podlagi valčne multiresolucijske analize smo povezali z večnivojskim perceptronom v enovit model. Z zapisom v matrični obliki smo pokazali, da lahko enoto za odstranjevanje šuma obravnavamo kot nivoje večnivojskega perceptrona. Za nastavljanje prostih parametrov enote in nevronske mreže uporabimo isto kriterijsko funkcijo. Predlagani enoviti model omogoča uspešnejše napovedovanje zašumljenih časovnih vrst, saj se odstranjevanje šuma prilagodi problemu tako, da je napoved najboljša.

Ključne besede: valčna multiresolucijska analiza, nevronske mreže, odstranjevanje šuma

1 Introduction

Neural networks are general function estimators creating associations between input–output samples from a given data set. Their ability to generalize makes them useful in the noisy time-series analysis. Although they inherently remove, to a certain extent, noise from a time series, denoising of input samples can greatly improve their ability to capture valuable information. Different methods for denoising can be used, such as filtering [9] or denoising based on the wavelet multiresolution analysis [8]. In this paper it is shown that the latter can be formulated analogously with the layered neural networks. Denoising can therefore be treated as a part of the model in architectural sense, and the same cost function used for setting free parameters of the neural network and the denoising unit.

Several examples of combining neural networks with wavelets can be found in literature. Zhang [13] proposed wavelets as activation functions of neurons. Wavelets have also been used to reduce the complexity of the problem. Thomason [11] decomposed a time series into two, modeled them separately and summed them up. On the other hand, Aussem and Murtagh [1] performed the modelling in the space of wavelet coefficients. Procházka, Mudrová and Štorek [10] applied wavelet denoising with statistical criteria as preprocessing in combination with layered neural networks.

The approach given in this paper is similar to the last two. In addition to the approach of Aussem and Murtagh, wavelet based denoising is also applied in the space of wavelet coefficients. Contrary to the approach of Procházka, Mudrová and Štorek, wavelet-based denoising is not used to preprocess the entire time series, but, rather, is integrated in the layered neural network, the multilayered perceptron in this case, working on each input sample separately.

A brief background of wavelet-based denoising is given in the next section. In the third section, wavelet denoising is integrated in the multilayered perceptron and a gradient-based learning algorithm for the thresholds is presented. In the fourth section, the performance of the uniform model is compared to the performance of the multilayered perceptron. The main conclusions are drawn in the last section.

2 Wavelet-Based Denoising

In the wavelet multiresolution analysis [3] a time series is observed on different time scales or at different resolutions. On each time scale j the time series is decomposed

Received 10 October 2002
Accepted 5 April 2003

into an approximation described by the scaling functions $\phi_{j,k}(t)$ and the remaining details given by the wavelets $\psi_{j,k}(t)$, $j,k \in \mathsf{Z}$. Scaling functions and wavelets are obtained by scaling and translating the father wavelet $\phi(t)$, $\phi_{j,k}(t) = 2^{-j/2}\phi(2^{-j}t - k)$, and the mother wavelet $\psi(t)$, $\psi_{j,k}(t) = 2^{-j/2}\psi(2^{-j}t - k)$. The largest scale J to which the time series is decomposed is usually limited. In this case the time series $x(t)$ can be written as

$$x(t) = \sum_{k\in\mathsf{Z}} a_k^J \phi_{J,k}(t) + \sum_{j\le J}\sum_{k\in\mathsf{Z}} d_k^j \psi_{j,k}(t) \ , \quad (1)$$

where the sum with the wavelet coefficients a_k^J presents the approximation on the scale J, while the sums with the wavelet coefficients d_k^j present the details on the scales considered. When specific technical details are fulfilled [3], the wavelet coefficients a_k^J and d_k^j, $j,k \in \mathsf{Z}$ form the wavelet transform of the time series $x(t)$.

The wavelet coefficients a_k^j and d_k^j on adjacent scales are related by the decomposition

$$a_k^{j+1} = \sum_{n\in\mathsf{Z}} h^*_{n-2k} a_n^j \ , \quad (2)$$

$$d_k^{j+1} = \sum_{n\in\mathsf{Z}} g^*_{n-2k} a_n^j \ , \quad (3)$$

as well as by the reconstruction

$$a_n^j = \sum_{k\in\mathsf{Z}} (h_{n-2k} a_k^{j+1} + g_{n-2k} d_k^{j+1}) \ , \quad (4)$$

where $h_i = \int_{\mathsf{R}} \phi(t)\phi_{-1,i}(t)dt$, $g_i = (-1)^i h^*_{1-i}$, the asterix denoting complex conjugates. The coefficients h_i and g_i of the most popular wavelets are given in reference [3].

In further discussion, a discrete time series with the values $x_1, \ldots, x_N$ acquired at a constant sampling rate is considered [3]. To simplify the calculation of wavelet coefficients on a discrete time series, $a_k^0 = x_k$, $k = 1, \ldots, N$, is set. In this case, the number of decompositions J is limited to the largest integer, smaller than or equal to $\log_2 N$, $J = \lfloor \log_2 N \rfloor$.

The decomposition and the reconstruction of a time series are schematically presented in Fig. 1, whereas an example is shown in Fig. 2.

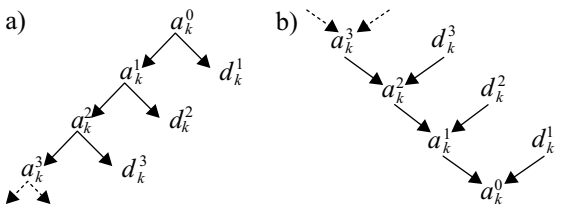

Figure 1. Time series a) decomposition and b) reconstruction

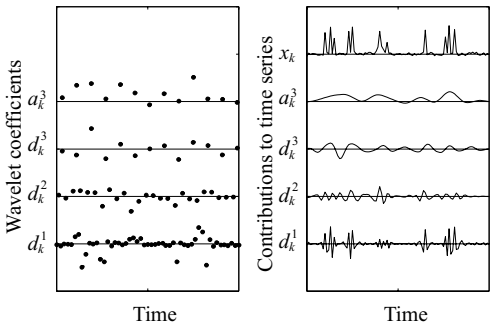

Figure 2. Coefficients a_k^3 and d_k^j, $j = 1, \ldots, 3$, and their contributions to time series x_k, $k \in \mathsf{Z}$

Usually, the decomposition results into many small wavelet coefficients d_k^j of the details, mainly contributing to noise [4]. Such compact coding can also be observed in Fig. 2. Furthermore, the decomposition is linear and orthogonal operation, transforming white noise in the time space into white noise in the space of the wavelet coefficients [4]. Based on these properties, Donoho and Johnstone [4] introduced wavelet-based denoising. The soft thresholding function

$$T(d_k^j, \tau_j) = d_k^j + \frac{1}{2}\left(\sqrt{(d_k^j - \tau_j)^2 + s} - \sqrt{(d_k^j + \tau_j)^2 + s}\right) \quad (5)$$

removes the coefficients d_k^j smaller than the threshold τ_j and reduces the absolute values of the coefficients d_k^j larger than the threshold τ_j (Fig. 3). The parameter $s \ge 0$ determines the smoothness of the soft thresholding function [7].

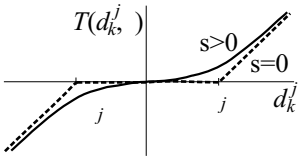

Figure 3. Soft thresholding function

The denoised time series $\tilde{x}(t)$ with values $\tilde{x}_k$ is obtained when the thresholded wavelet coefficients

$$\tilde{d}_k^j = T(d_k^j, \tau_j) \quad (6)$$

are used instead of the original ones in Eq. (1). Denoising methods differ in the way thresholds are estimated. Several statistically justified denoising criteria have been applied: Donoho and Johnstone used the minimax principle [4], and Cherkassky and Shao the Vapnik-Chervonenkis theory [2], for example.

3 Model

One of the most frequently used layered neural networks is the multilayered perceptron [5]. It owes its popularity to the mathematical formulation enabling derivation of

efficient learning algorithms used to adapt the model free parameters. In order to simplify its task of finding the best associations between the given input–output samples, the denoising unit is set in front of the first neural layer.

3.1 Model Structure

Wavelet-based denoising can be represented as two specialized layers of the neural network (Fig. 4). In the

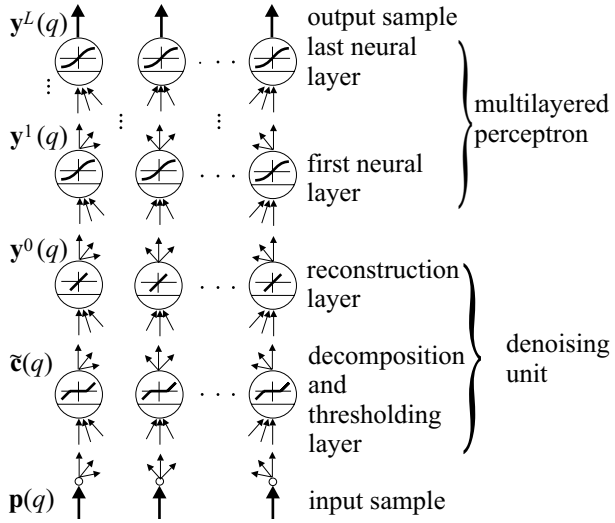

Figure 4. Multilayered perceptron with a denoising unit

decomposition and thresholding layer the input sample is first decomposed into the wavelet coefficients which are then thresholded. Using the reconstruction layer, the thresholded wavelet coefficients are reconstructed to form a denoised sample which is then further processed through neural layers of the multilayered perceptron.

3.1.1 Decomposition and Thresholding Layer

Wavelets considered in this work have finite support with the coefficients h_i and g_i, differing from zero only for $i = 1, \ldots, M$, with $M/2 > 0$ being the wavelet order. Besides, on the scale j only C_j wavelet coefficients a_n^j, $n = 1, \ldots, C_j$, of the approximation differ from zero. The decomposition formula given in Eq. (2) can thus be rewritten as

$$a_k^{j+1} = \sum_{n=n_{\mathrm{L}}(k)}^{n_{\mathrm{U}}(k)} h^*_{n-2k} a_n^j \,, \tag{7}$$

with the lower bound $n_{\mathrm{L}}(k) = \max(2k+1, 1)$ and the upper bound $n_{\mathrm{U}}(k) = \min(2k+M, C_j)$, giving wavelet coefficients a_k^{j+1}, $k = 1 - M/2, \ldots, \lfloor (C_j - 1)/2 \rfloor$.

To reduce the edge effects arising from the finite sample lengths [9], the sample is commonly extended at the beginning and at the end prior to transformation. In this case the index n ranges from $n_{\mathrm{L}}(k) = 2k+1$ up to $n_{\mathrm{U}}(k) = 2k+M$. The smallest index $n_{\mathrm{L}}(1 - M/2)$ and the largest index $n_{\mathrm{U}}(\lfloor (C_j - 1)/2 \rfloor)$ involved in calculation imply the extension of the sample with $M-2$ values at the beginning and $M-1$ values at the end.

Introducing the vector notation, the extended sample on the scale j, $\mathbf{e}_j = (e_1^j, \ldots, e_{C_j+2M-3}^j)^{\mathrm{T}}$ is obtained from the vector $\mathbf{a}_j = (a_1^j, \ldots, a_{C_j}^j)^{\mathrm{T}}$ of the wavelet coefficients a_k^j using the extension matrix $\mathbf{E}_j$,

$$\mathbf{e}_j = \mathbf{E}_j \, \mathbf{a}_j \,. \tag{8}$$

For example, when the sample is symmetrically extended on both edges, the matrix $\mathbf{E}_j$ is given by

$$\mathbf{E}_j = \begin{pmatrix} \mathbf{P}_{\mathrm{L}}(M-2, C_j) \\ \mathbf{I}(C_j) \\ \mathbf{P}_{\mathrm{F}}(M-1, C_j) \end{pmatrix} \tag{9}$$

where the matrix $\mathbf{P}_{\mathrm{L}}(r, c)$ represents the last r rows and the matrix $\mathbf{P}_{\mathrm{F}}(r, c)$ the first r rows of the matrix

$$\mathbf{P}(c) = (\, \mathbf{O}(c), \mathbf{I}(c), \ldots, \mathbf{O}(c), \mathbf{I}(c), \mathbf{O}(c) \,)^{\mathrm{T}} \,. \tag{10}$$

The identity matrix $\mathbf{I}(c)$ and the matrix $\mathbf{O}(c)$ are quadratic matrices with the elements $\mathrm{I}_{i,i} = 1$ and $\mathrm{O}_{i,c+1-i} = 1$, $i = 1, \ldots, c$, differing from zero.

In the extended vector $\mathbf{e}_j$, the vector of the wavelet coefficients $\mathbf{a}_j$ of the approximations is shifted by $M-2$ indices. Additionally, to allow for the matrix notation, the indices of the wavelet coefficients a_k^{j+1} should increase by $M/2$. Considering the extension of the sample and the two modifications of the indices, Eq. (7) can be written as

$$a_{k+M/2}^{j+1} = \sum_{n=2k+1}^{2k+M} h^*_{n-2k} e^j_{n+M-2} \,. \tag{11}$$

Introducing the index $k' = k + M/2$ and the index $n' = n + M - 2$, the matrix $\mathbf{F}^j(f)$ with the elements $\mathrm{F}^j_{k',n'} = f_{n'-2k'+2}$, $k' = 1, \ldots, C_{j+1}$, $n' = 2k'-1, \ldots, 2k'+M-2$, is defined. Using the matrix $\mathbf{A}_j^{\mathrm{D}} = \mathbf{F}^j(h^*)\mathbf{E}_j$, the decomposition formula (Eq. (11)) is given as

$$\mathbf{a}_{j+1} = \mathbf{A}_j^{\mathrm{D}} \, \mathbf{a}_j \,. \tag{12}$$

Similarly, introducing the matrix $\mathbf{D}_j^{\mathrm{D}} = \mathbf{F}^j(g^*)\mathbf{E}_j$ and the vector of the wavelet coefficients of the details $\mathbf{d}_j = (d_1^j, \ldots, d_{C_j}^j)^{\mathrm{T}}$, the decomposition formula given by Eq. (3) can be written as

$$\mathbf{d}_{j+1} = \mathbf{D}_j^{\mathrm{D}} \, \mathbf{a}_j \,. \tag{13}$$

Finally, the wavelet transform $\mathbf{c}(q) = (\mathbf{d}_1^{\mathrm{T}}, \ldots, \mathbf{d}_J^{\mathrm{T}}, \mathbf{a}_J^{\mathrm{T}})^{\mathrm{T}}$ of the q-th input sample $\mathbf{p}(q) = (p_1(q), \ldots, p_{N_0}(q))^{\mathrm{T}}$ is obtained by

$$\mathbf{c}(q) = \mathbf{W}^{\mathrm{D}} \, \mathbf{p}(q) \tag{14}$$

where the decomposition matrix is defined as

$$\mathbf{W}^{\mathrm{D}} = \begin{pmatrix} \mathbf{D}_0^{\mathrm{D}} \\ \mathbf{D}_1^{\mathrm{D}}\mathbf{A}_0^{\mathrm{D}} \\ \mathbf{D}_2^{\mathrm{D}}\mathbf{A}_1^{\mathrm{D}}\mathbf{A}_0^{\mathrm{D}} \\ \vdots \\ \mathbf{D}_{J-2}^{\mathrm{D}}\mathbf{A}_{J-3}^{\mathrm{D}}\cdots\mathbf{A}_0^{\mathrm{D}} \\ \mathbf{D}_{J-1}^{\mathrm{D}}\mathbf{A}_{J-2}^{\mathrm{D}}\cdots\mathbf{A}_0^{\mathrm{D}} \\ \mathbf{A}_{J-1}^{\mathrm{D}}\mathbf{A}_{J-2}^{\mathrm{D}}\cdots\mathbf{A}_0^{\mathrm{D}} \end{pmatrix} . \tag{15}$$

Following the thresholding rule given by Eqs. (5) and (6) and defining the vector of thresholds $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_J)^{\mathrm{T}}$, representing free parameters of the model, the vector of the denoised wavelet coefficients is given by

$$\begin{aligned}\tilde{\mathbf{c}}(q) &= T(\mathbf{c}(q), \boldsymbol{\tau}) = T(\mathbf{W}^{\mathrm{D}}\,\mathbf{p}(q), \boldsymbol{\tau}) = \\ &= \mathbf{c}(q) + \frac{1}{2}\left(\sqrt{(\mathbf{c}(q)-\mathbf{t})^2+\mathbf{s}} - \sqrt{(\mathbf{c}(q)+\mathbf{t})^2+\mathbf{s}}\right)\end{aligned} \tag{16}$$

where $(\cdot)^2$ and $\sqrt{\cdot}$ denote the square and square root of the vector elements, respectively. The vector $\mathbf{t}$ is constructed from thresholds

$$\mathbf{t} = (\underbrace{\tau_1, \ldots, \tau_1}_{C_1}, \ldots, \underbrace{\tau_J, \ldots, \tau_J}_{C_J}, \underbrace{0, \ldots, 0}_{C_J})^{\mathrm{T}} , \tag{17}$$

whereas the vector $\mathbf{s} = s\,(1, \ldots, 1)^{\mathrm{T}}$ with the smoothing parameter s determines the smoothness of the thresholding function.

3.1.2 Reconstruction Layer

Considering the finite sample size and finite number of coefficients h_i and g_i, Eq. (4) can be written as

$$a_n^j = \sum_{k=k_{\mathrm{L}}(n)}^{k_{\mathrm{U}}(n)} (h_{n-2k}a_k^{j+1} + g_{n-2k}d_k^{j+1}) \tag{18}$$

with the lower bound $k_{\mathrm{L}}(n) = \lfloor (n-M+1)/2 \rfloor$ and the upper bound $k_{\mathrm{U}}(n) = \lfloor (n-1)/2 \rfloor$. Since all the coefficients h_i and g_i are used in calculation of each wavelet coefficient, the lower and the upper bound are obtained from the relations $n - 2k_{\mathrm{L}}(n) \leq M$ and $n - 2k_{\mathrm{U}}(n) \geq 1$, respectively.

Owing to the shift of wavelet coefficients in the vectors $\mathbf{a}_{j+1}$ and $\mathbf{d}_{j+1}$ made in the decomposition, the reconstructed wavelet coefficients a_n^j need to be shifted back. Having in mind that $n - 2k$ decreases with step two when index k increases, the smallest index $n = M + 1$, is obtained from the relation $n - 2k = \min(M-1, M)$, $k = 1$, yielding the shift $-M$. Introducing the index $n'' = n - M$, the matrix $\mathbf{Q}^j(q)$ with the elements $\mathrm{Q}^j_{n'',k} = q_{n''+M-2k}$, $k = k_{\mathrm{L}}(n''+M), \ldots, k_{\mathrm{U}}(n''+M)$, $n'' = 1, \ldots, C_j$, can be constructed. Defining the matrices $\mathbf{A}_j^{\mathrm{R}} = \mathbf{Q}^j(h)$ and $\mathbf{D}_j^{\mathrm{R}} = \mathbf{Q}^j(g)$, the one-level reconstruction formula (Eq. (18)) is given as

$$\mathbf{a}_j = \mathbf{A}_j^{\mathrm{R}}\,\mathbf{a}_{j+1} + \mathbf{D}_j^{\mathrm{R}}\,\mathbf{d}_{j+1} . \tag{19}$$

Finally, the inverse discrete wavelet transform is given by

$$\mathbf{p}(q) = \mathbf{W}^{\mathrm{R}}\,\mathbf{c}(q) , \tag{20}$$

with the matrix $\mathbf{W}^{\mathrm{R}}$ constructed in the following way:

$$\begin{aligned}\mathbf{W}^{\mathrm{R}} \;=\; & (\mathbf{D}_0^{\mathrm{R}}, \mathbf{A}_0^{\mathrm{R}}\mathbf{D}_1^{\mathrm{R}}, \mathbf{A}_0^{\mathrm{R}}\mathbf{A}_1^{\mathrm{R}}\mathbf{D}_2^{\mathrm{R}}, \ldots, \\ & \mathbf{A}_0^{\mathrm{R}}\cdots\mathbf{A}_{J-3}^{\mathrm{R}}\mathbf{D}_{J-2}^{\mathrm{R}}, \mathbf{A}_0^{\mathrm{R}}\cdots\mathbf{A}_{J-2}^{\mathrm{R}}\mathbf{D}_{J-1}^{\mathrm{R}}, \\ & \mathbf{A}_0^{\mathrm{R}}\cdots\mathbf{A}_{J-2}^{\mathrm{R}}\mathbf{A}_{J-1}^{\mathrm{R}}) .\end{aligned} \tag{21}$$

Applying the discrete inverse wavelet transform to the vector of thresholded wavelet coefficients $\tilde{\mathbf{c}}(q)$, the denoised sample

$$\mathbf{y}^0(q) = \mathbf{W}^{\mathrm{R}}\tilde{\mathbf{c}}(q) \tag{22}$$

is obtained, being further passed to the neural layers.

3.1.3 Neural Layers of the Multilayered Perceptron

Neurons in the multilayered perceptron are arranged in L layers and each of the N_l neurons in the l-th layer is connected to all neurons in the adjacent layers [5]. When the q-th input–output sample is presented to the multilayered perceptron, the internal net activity level is first calculated:

$$\mathbf{v}^l(q) = \boldsymbol{\omega}^l \begin{pmatrix} -1 \\ \mathbf{y}^{l-1}(q) \end{pmatrix} . \tag{23}$$

The vector $\mathbf{y}^{l-1}(q)$ incorporates the outputs $y^{l-1}(q)$ from the previous layer, whereas the vector $\mathbf{v}^l(q)$ represents internal net activity levels $v_n^l(q)$ of all neurons on the l-th layer. The elements $\omega_{n,i}^l$ of the weight matrices $\boldsymbol{\omega}^l$, $l = 1, \ldots, L$, called weights, are additional free parameters of the model. Bias is introduced into the internal activity level by adding the constant -1 to the outputs from the previous layer. The outputs of the neurons in the l-th layer are given by

$$\mathbf{y}^l(q) = \varphi^l(\mathbf{v}^l(q)) \tag{24}$$

where φ^l is the activation function, acting element wise. Throughout this work the hyperbolic tangent $y_n^l(q) = \tanh(v_n^l(q))$ is used.

Although Eqs. (14) and (16) of the decomposition and thresholding layer are very similar to Eqs. (23) and (24) of the neural layers, the layers apparently differ in the positions of the free parameters. While the thresholds, as the free parameters of the decomposition and thresholding layer, determine the shape of the thresholding functions (Eq. (16)), the weights, being the free parameters of the neural layers, determine the internal net activity levels (Eq. (23)).

3.2 Learning

In the time series modelling, the model usually connects the value of a time series x_t at time t with N_0 previous values. To establish such connection, the model is

trained on a set of known input–output pairs. From a time series with values $x_1, \ldots, x_N$, the input–output pairs $\{\mathbf{p}(q), \mathbf{r}(q)\}$, $q = 1, \ldots, N - N_0$, can be constructed with the elements $p_i(q) = x_{i+q-1}$, $i = 1, \ldots, N_0$, and $r_1(q) = x_{q+N_0}$.

In the backpropagation learning [5], the free parameters of the model are iteratively updated, separately for each input–output pair. Minimizing the cost function, defined as Euclidian norm $\mathcal{E}(q) = ||\mathbf{r}(q) - \mathbf{y}^L(q)||_2$, the backpropagation learning approximates the steepest descent rule. The weights of neural layers are updated by the following equation:

$$\boldsymbol{\omega}^l \leftarrow \boldsymbol{\omega}^l - \eta \nabla_{\boldsymbol{\omega}^l} \mathcal{E}(q) \ , \ l = 1, \ldots, L \ , \qquad (25)$$

with the cost function gradients with respect to the weights $\nabla_{\boldsymbol{\omega}^l} \mathcal{E}(q) = (\ldots, \partial \mathcal{E}(q)/\partial \omega^l_{i,j}, \ldots)^{\mathrm{T}}$ calculated by the recurrence relation [5].

The backpropagation learning is designed for optimization of non-constrained free parameters. For the threshold τ_j, however, only nonnegative values smaller than or equal to absolutely the largest wavelet coefficient $d_{j,\max}$ are reasonable [8]. To extend the backpropagation learning to the denoising unit, unlimited thresholds τ_j^∞ are introduced:

$$\tau_j = d^j_{\max}(1 + e^{-\tau_j^\infty})^{-1} \ , \ j = 1, \ldots, J \ . \qquad (26)$$

Defining the vector of unlimited thresholds $\boldsymbol{\tau}^\infty = (\tau_1^\infty, \ldots, \tau_J^\infty)^{\mathrm{T}}$, the update equation becomes

$$\boldsymbol{\tau}^\infty \leftarrow \boldsymbol{\tau}^\infty - \eta \nabla_{\boldsymbol{\tau}^\infty} \mathcal{E}(q) = \boldsymbol{\tau}^\infty - \eta \nabla_{\boldsymbol{\tau}} \mathcal{E}(q) \mathbf{T} \ , \quad (27)$$

with the diagonal elements of the matrix $\mathbf{T}$ differing from zero, $\mathrm{T}_{j,j} = \tau_j(1 - \tau_j/c_{j,\max})$, $j = 1, \ldots, J$. The thresholds τ_j define the shape of the thresholding function, and so an approach similar to [12] is used to calculate the cost function gradient with respect to thresholds $\nabla_{\boldsymbol{\tau}} \mathcal{E}(q) = (\partial \mathcal{E}/\partial \tau_1, \ldots, \partial \mathcal{E}/\partial \tau_J)^{\mathrm{T}}$ [6]. Additionally, the smoothness parameter $s > 0$ should be used to prevent the minimization from unintentional stopping [6].

4 Results

The proposed multilayered perceptron with a denoising unit (DMLP) was applied on two real-world one-step ahead prediction problems: prediction of the sunspot number and prediction of the quality of the rubber compounds. To allow for a comparison, the classical multilayered perceptron (MLP) was applied to the same problems.

From each time series input–output pairs were prepared. The first 85% of the input–output pairs were included in the training set and were used to set free parameters of the models. The remaining 15% of the input–output pairs formed the test set and were used for the model comparison. The number of model inputs N_0 was limited to 20 and the number of free parameters to 30% of the input–output pairs. In the DMLP model, wavelets from the symlet family [3], $S^{M/2}$, were used with $M \leq N_0$.

The normalized root mean squared error $\mathrm{NRMSE} = (\sum_{q=1}^{Q}(r(q) - y_1^L(q))^2/(\sigma^2 Q))^{1/2}$ was considered as a performance measure, with σ denoting the standard deviation of a time series and Q the number of input–output pairs in the observed set. For each model configuration the learning was repeated 20 times and only configurations with the smallest performance measure were used in further analysis.

4.1 Sunspot Number

Sunspots are dark temporary areas of concentrated magnetic fields on the sun, contracting and expanding as they move across the surface. Their number is highly correlated with the sun activity and as such observed for centuries.

Fig. 5 shows prediction of the yearly average of the sunspot number, based on the data collected between the years 1700 and 2001. The average period of the most obvious solar cycle is approximately 11 years. Except for the maximum and successive minimum around 1960, both models give adequate predictions of the future values.

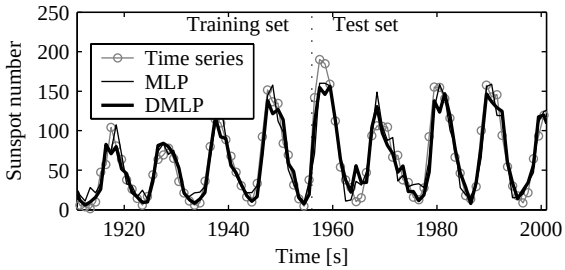

Figure 5. Prediction of the sunspot number

A comparison of the two models is given in Table 4.1. For each model the structure having led to the smallest error on the test set is given. The structure is described as the number of inputs and computational units in each layer. The DMLP model is slightly in favor because of the severe denoising of the details on the first scale. At the end of the learning process, namely, the thresholds of the DMLP model were 94%, 14% and 0% of the absolutely

Model	MLP	DMLP
Structure	8-4-1	S^2, 8-(15-8)-4-1
Free parameters	41	44
NRMSE, training set	0.29	0.28
NRMSE, test set	0.46	0.44

Table 1. Model comparison in prediction of the sunspot number

maximal wavelet coefficient of the detail on each of the first three scales, respectively.

4.2 Hardness of the Rubber Compound

An important characteristic determining the quality of the rubber compound is its hardness. It is measured in Shore units on the scale ranging from 0 to 100. Variations in hardness for successive mixings of a compound used for bicycle and motorcycle tubes are shown in Fig. 6.

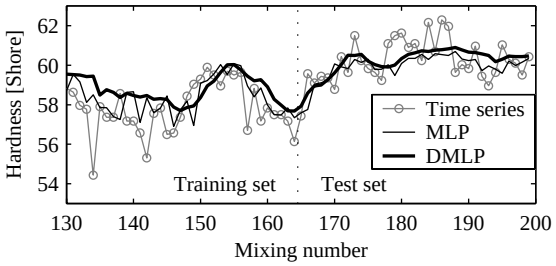

Figure 6. Prediction of the rubber quality

A detailed comparison of the models is given in Table 4.2. Considering the NRMSE on the test set, the impact of the denoising unit is more pronounced in this problem compared to the previous one. At the end of the learning process the thresholds were 87%, 67%, 13% and 54% of the absolutely maximal wavelet coefficient of the detail on each of the first four scales.

Model	MLP	DMLP
Structure	18-1-1	S^5, 19-(53-19)-1-1
Free parameters	21	26
NRMSE, training set	0.78	0.87
NRMSE, test set	0.67	0.63

Table 2. Model comparison in prediction of the rubber quality

5 Conclusions

A model integrating a wavelet-based denoising unit and multilayered perceptron is proposed. Using a matrix formulation of both, the denoising unit and the multilayered perceptron, it was shown that in the architectural sense the denoising unit is equivalent to the layers of the multilayered perceptron. The denoising layers apparently differ in the positions of free parameters. Thresholds, the free parameters of the denoising unit, determine the shape of the thresholding function, whereas the weight matrices, the free parameters of the multilayered perceptron, determine the internal net activity levels. Nevertheless, a gradient-based learning algorithm for simultaneous setting of all continuous free parameters using the same cost function can be derived.

An illustration of the model performance in a noisy time series prediction is given. Since in this case the model uses prediction error as the denoising criterion, the thresholds of the denoising unit are tuned for optimal prediction. In both examples this leads to improvements in prediction.

Acknowledgement

This work is partially supported by the Slovenian Ministry of Education, Science and Sport under the grant Z2-3040.

6 References

[1] A. Aussem and F. Murtagh, Combining neural network forecasts on wavelet-transformed time series, *Connection Science*, 9, pp. 113–121, 1997.

[2] V. Cherkassky and X. Shao, Signal estimation and denoising using vc-theory, *Neural Networks*, 14, pp. 37–52, 2001.

[3] I. Daubechies, *Ten Lectures on Wavelets*, Philadelphia, SIAM, 1992.

[4] D. L. Donoho and I. M. Johnstone, Adapting to unknown smoothness via wavelet shrinkage, *Journal of the American Statistical Association*, 90(432), pp. 1200–1224, 1995.

[5] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd edition, New Jersey, Prentice-Hall, 1999.

[6] U. Lotrič, Wavelet based denoising integrated into multilayered perceptron, *Neurocomputing*, submitted for publication, 2002.

[7] U. Lotrič and A. Dobnikar, Uporaba valčne analize in nevronskih mrež pri napovedovanju časovnih vrst, *Organizacija*, 35(1), pp. 33–42, 2002.

[8] U. Lotrič and A. Dobnikar, Wavelet based smoothing in time series prediction with neural networks, In V. Kůrková, N. Steele, R. Nerudá, and M. Kárný, *Artificial Neural Nets and Genetic Algorithms: Proceedings of the International Conference in Prague*, Wien, Springer, 2001, pp. 43–46.

[9] T. Masters, *Neural, Novel & Hybrid Algorithms for Time Series Prediction*, Toronto, John Wiley & Sons, 1995.

[10] A. Procházka, M. Mudrová, and M. Štorek, Wavelet use for noise rejection and signal modelling, In A. Procházka, J. Uhlíř, P. J. W. Rayner, and N. G. Kingsbury, *Signal Analysis and Prediction*, Boston, Birkhauser, 1998, pp. 215–226.

[11] M. R. Thomason, Financial forecasting with wavelet filters and neural networks, *Journal of Computational Intelligence in Finance*, 5(2), pp. 27–32, 1997.

[12] E. Trentin, Networks with trainable amplitude of activation functions, *Neural Networks*, 14, pp. 471–493, 2001.

[13] Q. Zhang, Using wavelet network in nonparametric estimation, *IEEE Transactions on Neural Networks*, 8(2), pp. 227–236, 1997.

Uroš Lotrič received his Ph. D. degree in computer science from the University of Ljubljana, Slovenia in 2000. He currently holds the position of teaching assistant at the Faculty of Computer Science, University of Ljubljana. His research interests include neural networks, wavelets and their applications.

Andrej Dobnikar received his Ph. D. degree in computer science from the University of Ljubljana in 1984. He currently holds the position of professor at the Faculty of Computer Science, University of Ljubljana. His research interests cover soft computing, parallel processing and adaptive systems.