BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 59, No. 1, 2011
DOI:

Adaptive synthesis of a wavelet transform using fast neural network

J. STOLAREK*

Institute of Information Technology, Technical University of Lodz, 215 Wólczańska St., 93-005 Łódź, Poland

Abstract. This paper introduces a new method for an adaptive synthesis of a wavelet transform using a fast neural network with a topology based on the lattice structure. The lattice structure and the orthogonal lattice structure are presented and their properties are discussed. A novel method for unsupervised training of the neural network is introduced. The proposed approach is tested by synthesizing new wavelets with an expected energy distribution between low- and high-pass filters. Energy compaction of the proposed method and Daubechies wavelets is compared. Tests are performed using sound and image signals.

Key words: wavelets, neural networks, filter parametrization, fast algorithms.

1. Introduction

A wavelet transform plays an important role in a signal analysis, compression and processing. During last years it has acquired a lot of attention from the researchers and it seems that is has become more popular than other linear transforms like DFT, DHT or DCT. Unlike these transforms Discrete Wavelet Transform (DWT) doesn't have one strictly defined set of basis functions. So far many wavelets have been designed, each with its unique properties. It is important that the chosen wavelet basis function precisely corresponds to characteristics of analysed signal. Therefore, it is necessary to develop methods for an adaptive synthesis of a wavelet best suitable for particular task.

Some attempts in that field have already been made. In [1] parametrization of Daubechies wavelets was proposed. This allowed to adjust wavelet's properties by changing the parameters. In [2] and [3] lattice structure for designing two-channel perfect reconstruction filters was presented. This approach was based on representing a filter bank in form of parametrized lattice structure. Parameters were optimized using well-known numerical methods (e.g. quasi-Newton method) and the resulting values, together with the lattice structure, defined the filter.

This paper presents a novel approach to adaptive synthesis of a wavelet transform. Generalized lattice structure and orthogonal lattice structure are presented. Fast neural network with topology based on this structures is discussed. Network's weights correspond to lattice structure parameters and they are modified during learning process leading to optimization of defined objective function. Main contribution of this paper is demonstration of effective method for unsupervised training of such multilayer network using backpropagation algorithm.

2. Lattice structure

Wavelet synthesis method presented in this paper is based on lattice structure introduced and described in [4]. Lattice structure is based on two-point base operations

$$D_k = \begin{bmatrix} w_{11}^k & w_{12}^k \\ w_{21}^k & w_{22}^k \end{bmatrix}, \tag{1}$$

where k stands for the index of operation (see Fig. 1a). Such two-point base operation can be written in form of a matrix equation

$$\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = D_k \cdot \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}. \tag{2}$$

Let us assume that D_k is invertible, i.e. condition

$$w_{11}^k w_{22}^k - w_{12}^k w_{21}^k \neq 0 \tag{3}$$

holds true. Hence there exists inverse transformation D_k^{-1} such that

$$D_k D_k^{-1} = I, \tag{4}$$

where I is the identity matrix (Fig. 1b).

Forward lattice structure is composed of $K/2$ stages, each containing D_k operations repeated $N/2$ times, where K and N are the lengths of the filter's impulse response and of a processed signal respectively (see Fig. 1c). On each stage of the lattice structure, elements of the signal are processed in pairs by D_k base operations. After each stage base operations are shifted down by one and the lower input of the last base operation in the current stage is connected to the upper output of the first base operation in the preceding stage (t_1 and t_2 in Fig. 1c).

Inverse lattice structure is created by reversing forward lattice structure and replacing each D_k operation with its inverse operation D_k^{-1}. Cyclic shift is performed in the opposite direction (Fig. 1d).

The presented lattice structure is used to calculate DWT. Upper outputs (b_1 in Fig. 1a) of base operations in last layer is referred to as the "low-pass outputs" and lower outputs (b_2 in Fig. 1a) will be referred to as the "high-pass outputs". In Fig. 1c and 1d all D_k operations within one layer are identical, however it is possible to design lattice structure in which operations within one layer are different.

*e-mail: jan.stolarek@p.lodz.pl

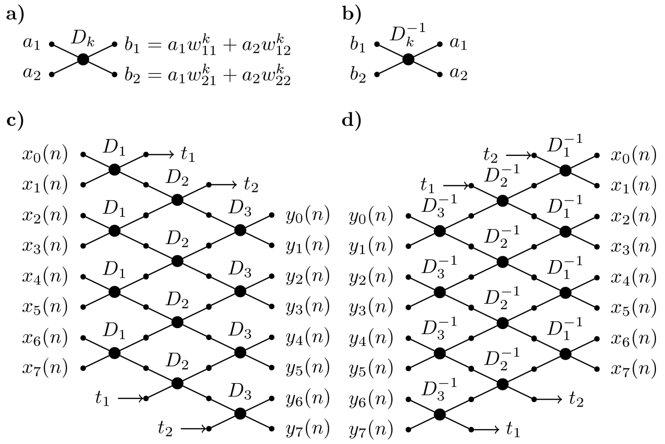

Fig. 1. a) forward base opertaion, b) inverse base opertaion, c) forward lattice structure for $K = 6$, $N = 8$, d) inverse lattice structure for $K = 6$, $N = 8$

3. Orthogonal lattice structure

Let us assume that D_k transform is orthogonal. By definition scalar product of D_k basis functions (i.e. rows or columns of D_k transform) equals zero:

$$w_{11}^k w_{21}^k + w_{12}^k w_{22}^k = 0. \tag{5}$$

Therefore

$$D_k \cdot D_k^T = D, \tag{6}$$

where D_k^T is transpose of D_k matrix and D is a diagonal matrix (entries outside the main diagonal are all zero). This means, that although orthogonal D_k transform can be inverted by simply transposing the transformation matrix, it doesn't preserve signal's energy. Energy is preserved however, when each of the basis functions (each row or column of D_k matrix) has unit norm, which implies that

$$D_k \cdot D_k^T = I, \tag{7}$$

where I is the identity matrix. Such transform is called *orthonormal*.

Equation (5) is explicitly satisfied when:

- $w_{21} = w_{12}$ and $w_{22} = -w_{11}$. This implies that transform is symmetric:

$$S_k = S_k^T = \begin{bmatrix} w_{11} & w_{12} \\ w_{12} & -w_{11} \end{bmatrix}. \tag{8}$$

If condition

$$w_{11}^2 + w_{12}^2 = 1 \tag{9}$$

holds true, then

$$S_k = S_k^T = S_k^{-1}. \tag{10}$$

- $w_{21} = -w_{12}$ and $w_{22} = w_{11}$. This implies that transform is asymmetric:

$$F_k = \begin{bmatrix} w_{11} & w_{12} \\ -w_{12} & w_{11} \end{bmatrix}, \quad F_k^T = \begin{bmatrix} w_{11} & -w_{12} \\ w_{12} & w_{11} \end{bmatrix}. \tag{11}$$

If Eq. (9) holds true, then

$$F_k^T = F_k^{-1}. \tag{12}$$

Matrices given by Eqs. (8) and (11) have different properties and not every transform can be represented in form of both of these matrices. Let us consider Haar transform [5]. It is a 2-tap transform, therefore it can be performed using one layer lattice structure. Haar low-pass filter is given by coefficients $\left[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}\right]$ and the high-pass filter is given by coefficients $\left[\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}\right]$. Therefore D_k transform corresponding to Haar transform is given by matrix

$$D_k = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}, \tag{13}$$

which is equivalent to Eq. (8) with $w_{11} = w_{12} = \frac{\sqrt{2}}{2}$. We notice that the Haar transform can't be represented in form of F_k matrix (Eq. (11)).

4. Fast neural network for wavelet synthesis

A fast neural network is used for determination of optimal lattice structure parameters, leading to synthesis of a wavelet. In this approach every D_k base operation is replaced by a pair of linear neurons, each of them with two inputs and one output, which guarantees a straightforward relation between the weights in a neural network and coefficients of the lattice structure[1]. All neurons within one layer have identical weights. To represent orthogonal lattice structure, orthogonal neural network [6] must be used. In such case each D_k base operation is represented using Basic Orthogonal Operation Neuron (BOON) corresponding to Eq. (8) or (11).

4.1. Training methods. In order to determine lattice structure coefficients using a neural network, objective function must be defined. This function is minimized during learning process and it shows how well network realizes transform of a signal.

First approach is the supervised teaching. In this case for each training pattern expected output value is known, which means that transform the network is supposed to learn must be

[1]In further text term "neuron" refers to a pair of linear neurons representing D_k base oparation.

known *a priori*. Objective function minimized in the learning process is a standard square error function [7]:

$$E = \frac{1}{2}\sum_{i=1}^{N}(d_i - y_i)^2, \tag{14}$$

where N denotes number of outputs from the network, y_i denotes signal value on i-th output and d_i denotes expected value for i-th output. This method however doesn't lead to synthesis of any new transforms and therefore it is only a proof of concept, that the network is able to learn a wavelet transform. It has been shown [8] that network with topology based on proposed lattice structure is able to learn Daubechies wavelets.

To synthesize a new wavelet unsupervised teaching must be used, since expected output values for patterns in a training set are unknown. In such a case square error, shown in Eq. (14), can't be used as an objective function. Therefore a new objective function must be designed. Following criteria for teaching a single neuron are proposed:

- each neuron preserves energy,
- energy ratio between the outputs of each neuron is fixed to some desired value.

Error function for a single neuron is given by formula

$$E = \frac{1}{2}\sum_{i=1}^{2}(d_i^2 - b_i^2)^2, \tag{15}$$

where b_i^2 is the energy of i-th output of a neuron and d_i^2 is the expected energy on that output. Therefore expression $(d_i^2 - b_i^2)$ is interpretated as error on i-th output. Given expected energy proportions h_1 and h_2, where $h_1 + h_2 = 1$, expected output values are determined using formula

$$d_i^2 = h_i \cdot (a_1^2 + a_2^2), \tag{16}$$

where $i \in \{1, 2\}$, a_1 and a_2 are neuron's inputs (see Fig. 1a).

Objective function for a single layer is given by formula

$$E = \frac{1}{2}\sum_{j=1}^{N/2}\sum_{i=1}^{2}(d_{ji}^2 - b_{ji}^2)^2, \tag{17}$$

where j is the index of neuron in a layer, b_{ji}^2 is the energy of i–th output of a j-th neuron and d_{ji}^2 is the expected energy on that output. Expected energies d_{ji} are calculated independently for each neuron, based on its actual inputs.

The above teaching method is suitable for one–layer network. The problem arises when multilayer network must be trained. One of the solutions to this problem is forward propagation of input signal through the network and then training layers independently with different energy proportion defined for each layer [9]. In this paper defining expected energy proportion only for the output layer and teaching the network using backpropagation algorithm is demonstrated. For a straightforward determination of objective function's gradient in respect to the weights Signal Flow Graphs (SFG) are used [7, 10]. Due to non-standard form of objective function, adjustment of backpropagation algorithm is required. Since each output of the network is raised to the power of two before comparing it to the expected value, it is necessary to multiply error value backpropagated for each output by $-2b_{ji}$ [7, 10].

Weights modification is performed according to the steepest descent algorithm:

$$w_{n+1} = w_n - \eta\nabla E(w), \tag{18}$$

where w_n is weights vector in n-th iteration, η is the learning step and $\nabla E(w)$ is error function's gradient calculated in respect to network's weights. However, this method doesn't preserve norm of weights, which is not acceptable in case of orthonormal transform, since the preservation of energy requires that weight vector for each base operation has unit norm. Therefore, for preservation of energy, weights must be normalized after each update.

It is important to notice, that it is not possible to find such weights of the neuron, that it produces expected energy proportions for each input signal. It is however possible to determine such weights that, for a given class of signals, energy proportions are true in a statistical sense. Therefore, it is important, that the network is trained using signals of some particular classes, e.g. image or sound.

5. Experimental validation

5.1. Network and dataset preparation. Neural network with topology based on lattice structure and orthonormal symmetric base operations was designed for experiments. Tests were carried out using two different classes of signals: sound and image. For each class two different data sets were prepared. One set – containing 400 patterns – was used to train the network, the other – containing 1000 patterns – was used for testing. Image data was taken from rows of a grayscale images. Sound data was taken from songs containing vocals, drums, two guitars and a bass guitar. Each pattern in a set was a 64-element vector with its coordinates scaled to fit into $[0,\ 1]$ range. Network's initial weights were chosen randomly from range $[-1,\ 1]$ and then normalized, so each row of base operation would have unit length.

Experiments were carried out using 4-tap, 6-tap and 8-tap transforms (two-, three-, and four-layer networks respectively). Network was trained using off-line teaching [7]. Optimal values of parameters (e.g. number of teaching epochs or learning step) may differ depending on number of layers in the network and desired energy distribution.

5.2. Results and discussion. Tables 1 and 2 present the results. First column shows expected percentage of input energy located on low-pass outputs of the network. Remaining amount of energy is located on high-pass outputs of network, summing up to give a total of 100%. Remaining columns show testing result obtained on both training and testing sets, expressed as actual percentage of energy located on low-pass outputs of network in a particular data set. Seven different energy distributions were tested. Presented results are average values obtained from 10 independent tests. Tables present also energy distribution for Daubechies wavelets.

Table 1
Results of training performed on image data

Expected energy of low-pass outputs	Actual results					
	4-tap transform		6-tap transform		8-tap transform	
	training	testing	training	testing	training	testing
0%	1.82%	3.1%	1.71%	3.8%	1.67%	3.91%
10%	8.14%	9.69%	7.95%	9.46%	8.39%	9.5%
30%	29.32%	29.43%	29.05%	30.48%	29.38%	29.01%
50%	49.99%	50.74%	49.9%	49.95%	49.99%	49.80%
70%	71.21%	71.21%	70.77%	71.23%	70.86%	69.87%
90%	91.6%	92.92%	91.77%	90.43%	91.87%	89.93%
100%	97.29%	98.7%	98.33%	97.09%	98.28%	95.71%
Daubechies	98.43%	97.5%	98.43%	97.37%	98.44%	97.31%

Table 2
Results of training performed on sound data

Expected energy of low-pass outputs	Actual results					
	4-tap transform		6-tap transform		8-tap transform	
	training	testing	training	testing	training	testing
0%	4.71%	4.07%	4.7%	4.14%	4.64%	4.1%
10%	6.04%	5.95%	6.01%	5.73%	5.44%	5.01%
30%	27.27%	28.21%	27.23%	28.15%	27.47%	26.81%
50%	49.62%	50.9%	50.03%	50.45%	50.88%	50.88%
70%	72.85%	74.08%	73.06%	72.67%	72.51%	73.93%
90%	94.18%	94.27%	94.27%	94.59%	94.6%	94.95%
100%	95.39%	95.86%	95.4%	95.84%	95.43%	95.88%
Daubechies	95.11%	96.34%	95.14%	96.23%	95.3%	96.06%

Results show, that proposed unsupervised training method for neural network based on lattice structure is effective. There is no problem with achieving equal energy distribution, with error less than 1%. In case of image data other obtained results are mostly within 2% from expected value. In case of sound, the results are not so accurate. If expected energy on low-pass outputs is less then 50%, the network has a tendency to allocate less energy to these outputs than necessary. Opposite tendency can be noticed when expected energy is greater than 50%. In case of both sound and image it is impossible to allocate all of signal's energy only to low- or high-pass outputs. Results also show, that proposed method performs similary to Daubechies wavelets in terms of energy compaction. In most cases Daubechies method outperforms presented neural network approach, but in a few cases neural network offers a slight improvement.

6. Conclusions

In this paper a new method for unsupervised training of neural network based on the orthogonal lattice structure was presented. A new objective function for estimating network's error was defined and an appropriate adjustment of the backpropagation algorithm was discussed. It was demonstrated that proposed method can be effectively used for the adaptive synthesis of a new wavelet with the desired energy distribution for a signal of a particular class, which is the most important contribution of this paper. Presented neural network approach was compared with Daubechies wavelets in terms of energy compaction. Results have shown that Daubechies wavelets in general perform slightly better, however in some cases they can be outperformed by presented adaptive method.

It was shown, that symmetric and asymmetric orthogonal base operations have different properties. Therefore within the further development of proposed orthogonal lattice structure it is necessary to determine relation between type of base operation and the class of orthogonal wavelet transforms possible to synthesize. Possibility of applying presented method to tasks other than signal compression should be investigated as well.

Acknowledgements. This paper is a revised and exteded version of a paper originally presented at XI International PhD Workshop OWD, Wisła, October 2009.

REFERENCES

[1] P. Lipiński and M. Yatsymirskyy, "On synthesis of 4-tap and 6-tap reversible wavelet filters", *Electrotechnical Review* 12, 284–286 (2008).

[2] P. Rieder, J. Gotze, J.S. Nossek, and C.S. Burrus, "Parameterization of orthogonal wavelet transforms and their implementation", *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Trans.* 45 (2), 217–226 (1998).

[3] P.P. Vaidyanathan and P.-Q. Hoang, “Lattice structures for optimal design and robust implementation of two-channel perfect-reconstruction QMF banks”, *Acoustics, Speech and Signal Processing, IEEE Trans.* 36 (1), 81–94 (1988).

[4] M. Yatsymirskyy, “Lattice structures for synthesis and implementation of wavelet transforms”, *J. Applied Computer Science* 17 (1), 133–141 (2009).

[5] J.S. Walker, *A Primer on Wavelets and Their Scientific Applications*, Chapman & Hall/CRC, 2008.

[6] B. Stasiak and M. Yatsymirskyy, “Fast orthogonal neural networks”, *Lecture Notes in Computer Science* 4029, 142–149 (2006).

[7] S. Osowski, *Neural Networks to Information Processing*, Warsaw University of Technology Publishing House, Warszawa, 2006.

[8] J. Stolarek, “Realization of Daubechies transform using lattice structure”, *The Collection of Scientific Works of ISDMCI* 2, 188–192 (2009).

[9] J. Stolarek and M. Yatsymirskyy, “Fast neural network for synthesis and implementation of orthogonal wavelet transform”, *Image Processing & Communications Challenges* 1, 87–97 (2009).

[10] S. Osowski, “Signal flow graphs and neural networks”, *Biological Cybernetics* 70 (4), 387–395 (1994).