Synthesis of a wavelet transform using neural network

Jan Stolarek
Institute of Computer Science
Technical University of Lodz
ul. Wolczanska 215, 93-005 Lodz, Poland
e-mail: stolarek@ics.p.lodz.pl

Abstract

Wavelet transform has a wide area of application in signal processing. However there is no single wavelet perfectly suitable for every task. In practice Daubechies 4 is the most commonly used wavelet, since it is well suited for analysis of many natural signals and it offers a straightforward interpretation of the results. It would be very useful to develop a method for adaptive synthesis of a wavelet transform suitable for particular task. Artificial neural networks offer such ability. So far this approach wasn’t explored. This paper presents neural network for synthesis of orthogonal wavelet transform and a method of unsupervised training of this network.

2 Lattice structure

Wavelet synthesis method presented in this paper is based on lattice structure introduced and described in [11]. Lattice structure is based on two-point base operations

\[ D_k = \begin{bmatrix} w_{11}^k & w_{12}^k \\ w_{21}^k & w_{22}^k \end{bmatrix}, \]

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

\[ \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = D_k \cdot \begin{bmatrix} a_1 \\ a_2 \end{bmatrix}. \]

Let us assume that \( D_k \) is invertible, i.e. condition \( w_{11}^k w_{22}^k - w_{12}^k w_{21}^k \neq 0 \) holds true. Hence there exists inverse transformation \( D_k^{-1} \) such that \( D_k D_k^{-1} = I \), where \( I \) is the identity matrix (Figure 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 the processed signal respectively (see Figure 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 \( t_2 \) on Figure 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 (Figure 1d).

On Figure 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.
Figure 1: Base operations: a) forward, b) inverse.
Lattice structure for $K = 6, N = 8$: c) forward, d) inverse.

Presented lattice structure can be used to calculate DWT. Therefore, upper outputs ($b_1$ on Figure 1a) of base operations in last layer will be referred to as “low-pass outputs” and lower outputs ($b_2$ on Figure 1a) will be referred to as “high-pass outputs”.

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:

$$w_{11}^k w_{12}^k + w_{12}^k w_{22}^k = 0 \quad (3)$$

Therefore:

$$D_k \cdot D_k^T = D \quad (4)$$

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 length:

$$D_k \cdot D_k^T = I \quad (5)$$

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

Equation 3 is explicitly satisfied when:

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

$$F_k = \begin{bmatrix} w_{11} & w_{12} \\ -w_{12} & w_{11} \end{bmatrix}, \quad F_k^{-1} = \begin{bmatrix} w_{11} & -w_{12} \\ w_{12} & w_{11} \end{bmatrix}. \quad (7)$$

Matrices given by equations 6 and 7 have different properties and not every transform can be represented in form of both of these matrices. Let us consider Haar transform [9]. It is a 2–tap transform, therefore it can be performed using one layer lattice structure. Haar low–pass filter is given by coefficients $[\sqrt{2}, \sqrt{2}]$ and the high–pass filter is given by coefficients $[\sqrt{2}, -\sqrt{2}]$. Therefore $D_k$ transform corresponding to Haar transform is given by matrix

$$D_k = \begin{bmatrix} \sqrt{2} & \sqrt{2} \\ \sqrt{2} & -\sqrt{2} \end{bmatrix}, \quad (8)$$

which is equivalent to equation 6 with $w_{11} = w_{12} = \sqrt{2}$. We notice that Haar transform can’t be represented in form of $F_k$ matrix (equation 7).

4 Fast neural network for wavelet synthesis

Fast Neural Network is used for determination of optimal lattice structure parameters, leading to synthesis of optimal 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 the coefficients of lattice structure. All pairs of neurons within one layer have identical weights. To represent orthogonal lattice structure, orthogonal neural network [4] must be used. In such case each \( D_k \) base operation is replaced with Basic Orthogonal Operation Neuron (BOON) corresponding to equation 6 or 7.

### 4.1 Teaching methods

In order to determine lattice structure coefficients using 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 known \( a \) \( p r i o r i \). Objective function minimized in the learning process is a standard square error function [2, 7]. This method 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 [5] 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 case square error can’t be used as objective function. Therefore new objective function must be designed. Following criteria for teaching the neuron are proposed:

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

Objective function for a single layer is given by formula

\[
E = \sum_{j=1}^{N/2} \sum_{i=1}^{2} (d_{ji} - b_{ji}^2)^2 ,
\]

where \( j \) is the number of neuron in the layer, \( b_{ji}^2 \) is the energy of \( i \)-th output of a \( j \)-th neuron and \( d_{ji} \) is the expected energy on that output. Given expected energy proportions \( h \) and \( g \), where \( h + g = 1 \), expected output values are determined:

\[
d_{1j1} = h \cdot (a_{j1}^2 + a_{j2}^2), \\
d_{1j2} = g \cdot (a_{j1}^2 + a_{j2}^2)
\]

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

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 [6]. In this paper defining expected energy proportion only for the output layer and teaching the network using backpropagation algorithm is proposed. For a straightforward determination of objective function’s gradient in respect to the weights Signal Flow Graphs (SFG) are used [1, 2]. 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 \(-2h_{ji} \) [1, 2].

Weights modification is performed according to the steepest descent algorithm:

\[
w_{n+1} = w_n - \eta \nabla E(w) ,
\]

where \( w_n \) is weights vector in \( n \)-th iteration, \( \eta \) is the learning step and \( \nabla E(w) \) is error function’s gradient calculated in respect to network weights. It is known [7], that 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 length. Therefore, for preservation of energy, weights must be normalized after each update.

### 5 Experimental validation

An orthogonal neural network with topology based on proposed orthogonal lattice structure with orthonormal symmetric base operations was designed for experiments. Teaching set contained 400 training patterns, each pattern being 16–element vector taken from rows of an image. Testing set contained 1000 16–element vectors taken from different image. Values in both sets were normalized 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).

Table 1 presents the results of the learning process. First column shows expected percentage of input energy located on low–pass outputs of 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. Results presented in the table are average from 10 independent tests.

Network was trained using off–line teaching [2]. Optimal values of parameters (e.g. number of teaching epochs or learning step) may differ depending on number of layers in the network. Results clearly show that proposed network is able to achieve desired en-
energy distribution in case of image signals. However, it is impossible to distribute 100% of energy to low-pass or high-pass outputs.

6 Conclusion

Neural network presented in this paper can be used for adaptive synthesis of a wavelet with desired energy distribution for a signal of particular class. Presented teaching method allows to effectively train multilayer network in an unsupervised learning process given only expected energy ratio between low-pass and high-pass outputs of the lattice structure.

It was demonstrated, 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. It is also necessary to develop training methods that would allow to effectively train neural network corresponding to multilevel wavelet analysis.

References


Author

mgr inż. Jan Stolarek
Technical University of Lodz
Institute of Computer Science
Wólczańska 215
90-924 Łódź, Poland
phone: +48 42 6329757
e-mail: stolarek@ics.p.lodz.pl