Fast neural network for synthesis and implementation of orthogonal wavelet transform

Jan Stolarek[1] and Mykhaylo Yatsymirskyy[1]

Institute of Computer Science, Technical University of Lodz, ul. Wolczanska 215, 93-005 Lodz, Poland, stolarek@ics.p.lodz.pl, jacym@ics.p.lodz.pl

Summary. This paper presents lattice structure and orthogonal lattice structure for synthesis of a wavelet transform. Lattice structure is based on parametrized 2 × 2 operations. Adjustment of parameter values leads to synthesis of a new wavelet transform. Fast neural network with topology based on lattice structure is applied to determine values of the parameters. Factorization–based reduction in number of arithmetic operations is demonstrated.

Key words: signal processing, wavelet transform, neural networks

1 Introduction

During the last two decades, wavelet transform became an important tool in the area of compression, analysis and processing of signals. The most popular are the Daubechies wavelets [1] that are well suited for analysis of many natural signals and offer a straightforward interpretation of the result. Many other kinds of wavelets have been designed, each with its unique characteristics. However, there is no single wavelet perfectly suitable for every task. It is very important, that chosen wavelet family closely corresponds to characteristics of analysed signal. Therefore it is important to develop a method to adaptively synthesize a wavelet most suitable for particular task.

Some attempts in that field have already been made. In [10] parametrization of wavelet transform was proposed. This allows adjusting wavelet's properties by optimization of parameters. In [4] and [9] lattice structure for designing two–channel perfect reconstruction filters was presented. This approach was based on representing a filter bank in a form of parametrized lattice structure. Parameters were then 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 method for synthesis of a wavelet transform, based on generalized lattice structure [11]. Fast Neural Network is used to determine optimal values of parameters.

2 Lattice structure

Let us consider two–point base operation

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

where k stands for the index of operation. Let us assume that D_k is invertible, i.e. condition $d_{11}^k d_{22}^k - d_{21}^k d_{12}^k \neq 0$ holds true. Hence there exists inverse transformation D_k^{-1} such that $D_k D_k^{-1} = I$, where I is the identity matrix.

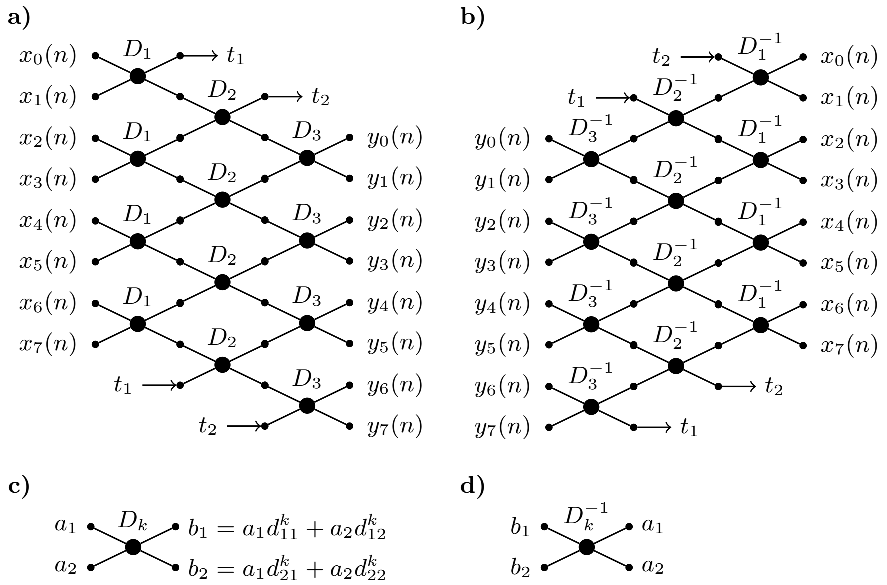
a)
b)
c)
d)
$x_0(n)$ $x_1(n)$ $x_2(n)$ $x_3(n)$ $x_4(n)$ $x_5(n)$ $x_6(n)$ $x_7(n)$
$y_0(n)$ $y_1(n)$ $y_2(n)$ $y_3(n)$ $y_4(n)$ $y_5(n)$ $y_6(n)$ $y_7(n)$
D_1 D_2 D_3
D_1^{-1} D_2^{-1} D_3^{-1}
t_1 t_2
a_1 a_2 D_k
$b_1 = a_1 d_{11}^k + a_2 d_{12}^k$
$b_2 = a_1 d_{21}^k + a_2 d_{22}^k$
b_1 b_2 D_k^{-1} a_1 a_2

Fig. 1. The lattice structure for $K = 6$, $N = 8$: a) forward, b) inverse. Base operations: c) forward, d) inverse

We introduce forward lattice structure that 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 1a). In the first stage the pairs of x_{2i} and x_{2i+1} samples are assigned to the inputs of each D_k operation, $i = 0, 1, \ldots, N/2 - 1$. Base operations in successive stages are shifted down by one position 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. In other words we perform the cyclic N–element shift of the mentioned outputs to the left (upward). The outputs of the last stage are the outputs y_i of the whole structure, where $i = 0, 1, \ldots, N-1$.

The inverse lattice structure emerges as the reversed forward structure, where base D_k operations are substituted by inverse operations D_k^{-1} and the cyclic shift is performed in the opposite direction. Figure 1b shows inverse structure for the structure presented in Figure 1a. The cyclic left (Figure 1a) and right (Figure 1b) shift operations are represented by t_1 and t_2 symbols.

In structures presented on Figure 1a and 1b all D_k operations within one stage are identical. However it is possible, that D_k operations within single stage are different. This allows for more precise adjustment of filter's properties.

Let us now estimate the number of arithmetic operations in the proposed structures. In a general case in order to execute $K/2$ stages while operating on N samples of a signal it is required to perform the number of $(N/2)(K/2)$ base operations D_k, where each operation requires two additions and four multiplications. Hence the forward and the inverse structures require $\alpha(K, N)$ additions and $\mu(K, N)$ multiplications separately where $\alpha(K, N) = KN/2$ and $\mu(K, N) = KN$. In other words for one pair of the processed elements each structure requires K additions and $2K$ multiplications. In order to achieve reduction in the number of arithmetic operations it is required to consider the particular base operations. This will be presented in the next section of this paper.

The main advantage of the proposed lattice structure is the ease of its realization in the pipelined scheme. In a general case the pipeline scheme contains $K/2$ blocks of base operations and $K/2 - 1$ blocks z^{-1} with delays by one element, see Figure 2.

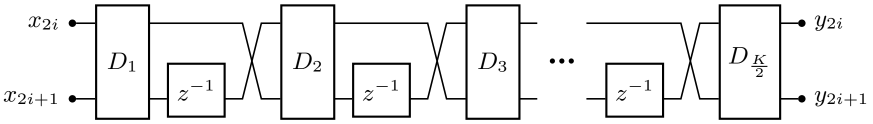

Fig. 2. The pipeline scheme for $K/2$–stage lattice structure

In the scheme in Figure 2 there is no cyclic shift which must be performed within the hardware or software implementation, e.g. by cyclic repetition of the first samples of input signal. The proposed pipeline scheme is simpler than the well-known lifting one. Firstly it reduces the number of required two-point base operations, e.g. by one for $K = 4$ and $K = 6$. Moreover, it utilizes the base operations of one type and all delays are placed outside base operations. In the lifting structure delays and z^1 shifts are enclosed within base operations.

3 Orthogonal lattice structure

Let us assume that D_k transformation is orthogonal, which implies that condition $d_{11}^k d_{21}^k + d_{12}^k d_{22}^k = 0$ holds true. We will consider two cases of such transformations :

- symmetric, when $S_k = S_k^{-1}$

$$S_k = S_k^{-1} = \begin{bmatrix} p_k & q_k \\ q_k & -p_k \end{bmatrix} \tag{2}$$

- asymmetric, when $F_k \neq F_k^{-1}$

$$F_k = \begin{bmatrix} p_k & q_k \\ -q_k & p_k \end{bmatrix} \ , \ \ F_k^{-1} = \begin{bmatrix} p_k & -q_k \\ q_k & p_k \end{bmatrix} \ , \tag{3}$$

where p_k and q_k are non-zero numbers. Each of transforms (2) and (3) still requires two additions and four multiplications. We will demonstrate with an example of S_k transform the technique of reducing the number of multiplications in structures from Figure 1 and 2. To achieve it, the following factorization is introduced :

$$S_k = E_k T_k \ , \tag{4}$$

where

$$T_k = \begin{bmatrix} 1 & t_k \\ t_k & -1 \end{bmatrix} \ , \ \ t_k = \frac{q_k}{p_k} \ , \ \ E_k = \begin{bmatrix} p_k & 0 \\ 0 & p_k \end{bmatrix} \ . \tag{5}$$

For instance, let us modify the pipeline scheme for the orthogonal transforms from Figure 2. Factorization (4) is applied to each stage of the transform. However we perform only T_k transformations while E_k transformations are grouped in the additional stage, see Figure 3.

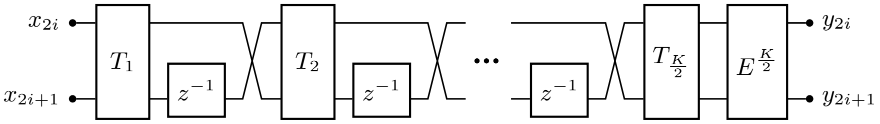

Fig. 3. The pipeline scheme for $K/2$–stage orthogonal lattice structure

As a result we obtain $K/2$ blocks of simplified base operations T_k, $K/2 - 1$ blocks with z^{-1} delays and one block $E^{K/2} = E_1 E_2 \dots E_{K/2}$. Each T_k operation requires two multiplications and two addition. Within $E^{K/2}$ block two multiplications are performed. Hence the scheme from Figure 3 requires K additions and $K + 2$ multiplications for one pair of output elements. In comparison to a general lattice structure we obtain almost twofold reduction in the number of multiplications preserving the same number of additions. It should be noted that $E^{K/2}$ block performs multiplications of it's inputs by the constant value and therefore it can be omitted by grouping the constants at successive factorizations of lengths $N/2, N/4, \ldots$. It results in reduction of two multiplications.

4 Synthesis of a wavelet transform

The values of a lattice structure coefficients can be determined by the means of two methods: the standard one and the one taking advantage of artificial neural networks. Standard method is based on determining the relations between impulse responses of H and G filters and coefficients of base operations. In this method we rewrite the impulse responses in the form of algebraic formulas depending on structure's coefficients. Next we follow the standard scheme that requires solving the system of equations including the orthogonality and the zero moments equations. This method is not simple since it requires solving the system of K non–linear equations and, in general, produces coefficients that are independent of specific requirements imposed on wavelet transform.

More interesting method utilizes the technique of artificial neural networks. This method was developed for the synthesis of fast algorithms for Fourier–like transforms [2, 5, 6]. In this method each stage of the lattice structure is replaced with one hidden layer of a linear neural network. Each D_k base operation is replaced with two 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 base operations. All pairs of neurons within one layer have identical weights, which means that such neural network is equivalent to a pipeline scheme. To represent orthogonal lattice structure, orthogonal neural network [5] must be used. In this case each D_k base operation is replaced with Basic Orthogonal Operation Neuron (BOON) corresponding to equation 2.

By training the neural network we determine values of lattice structure's coefficients. With supervised teaching, proposed network is capable of learning already existing wavelet transforms, e.g. Daubechies wavelets [7]. More interesting method relies on unsupervised teaching, when we don't know the exact expected output values. In order to teach the network using this method we must first define the objective function that will be optimized during the learning process. Our objective function will be based on energy — we expect that each BOON will preserve energy of its input signal and that energy ratio between the outputs of that neuron will be fixed to some desired value. This means, that for each BOON objective function is given by formula

$$E = \sum_{i=1}^{2} (d_i - b_i^2)^2 \quad , \tag{6}$$

where b_i^2 is the energy of i–th output of BOON and d_i is the expected energy on that output. Given expected energy proportions h and g, where $h + g = 1$, we determine expected output values : $d_1 = h(a_1^2 + a_2^2)$, $d_2 = g(a_1^2 + a_2^2)$. In case of multilayer network, signal is propagated forward through the network and then each layer is trained independently (different energy proportions can be defined for each layer).

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.

Network was trained using off–line teaching [3]. Weights modification was performed according to the steepest descent algorithm:

$$w_{n+1} = w_n - \eta \nabla E(w) \quad , \tag{7}$$

where w_n is weights vector in n–th epoch, η is the learning step and $\nabla E(w)$ is error function's gradient calculated in respect to network weights. It is known [8], that this method doesn't preserve norm of weights, which is not acceptable, since the preservation of energy requires that each weight vector has unit length. Therefore weights must be normalized after each update.

To explain teaching process in detail Haar transform will be used. Haar wavelet is a 2–tap transform, therefore it can be performed using one layer network. Haar low–pass filter is given by coefficients $[\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}]$ and the high–pass filter is given by coefficients $[\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}]$. Therefore D_k transform corresponding to Haar transform is given by matrix

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

which is equivalent to equation 2 with $p_k = q_k = \frac{\sqrt{2}}{2}$. We notice that

$$D_k = \begin{bmatrix} cos(45°) & sin(45°) \\ sin(45°) & -cos(45°) \end{bmatrix} = \begin{bmatrix} cos(45°) & -sin(45°) \\ sin(45°) & cos(45°) \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad , \tag{9}$$

which means that we can graphically interpret Haar transform, assuming that each pair of signal elements processed by one BOON is treated as coordinates of a point. Equation 9 shows that Haar transform is a superposition of scaling y coordinates of all points by -1 (reflection over x-axis) and rotation by 45 degrees. Daubechies wavelets and other n-tap wavelets $(n > 2)$ can be treated as multiple reflections and rotations with coordinates shift after each rotation. Therefore, teaching process can be viewed as search for optimal rotation angles that, for a given teaching set, provide output energy proportions closest to expected values.

5 Experimental validation

An orthogonal neural network with topology based an proposed lattice structure was designed for experiments. Teaching set consisted of 400 samples, each sample being 16–element vector taken from rows of an image. Testing set consisted of 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 D_k transform would have unit length. Only two weights for each D_k operation were randomized. Remaining two weights are defined by orthogonality of base operation (see Equation 2). Experiments were carried out using 2–tap and 4–tap transforms (one– and two–layer networks respectively).

Expected energy of	Rotation	Actual result	
upper outputs	angle	training set	testing set
2–tap transform			
0%	131.42°	2.27%	4.18%
10%	138.78°	2.03%	4.17%
30%	−11.99°	29.37%	31.01%
50%	90.86°	49.65%	48.83%
70%	13.67°	71.17%	71.03%
90%	−118.64°	91.18%	88.99%
100%	42.68°	97.98%	96.02%
4–tap transform			
[10%, 10%]	[120.83°, 90.51°]	7.4%	9.19%
[10%, 90%]	[136.01°, 16.32°]	90.29%	88.51%
[50%, 50%]	[−60.67°, 153.26°]	53.63%	53.77%
[90%, 10%]	[30.74°, 0.28°]	7.65%	9.44%
[90%, 90%]	[60.13°, −90.77°]	92.88%	90.86%

Table 1. Results for 2–tap and 4–tap transform

Table 1 presents the results of the learning process. First column shows expected percentage of input energy located on upper outputs of BOONs (i.e. b_1 outputs from Figure 1c). Remaining amount of energy is located on lower outputs of BOONs, summing up to give a total of 100%. Second column shows determined rotation angle. Remaining two columns show testing result obtained on both training and testing sets, expressed as actual percentage of energy located on upper outputs of BOONs. Columns in the 4–tap section show the same information, however expected energy proportions and determined rotation angles are given for both layers of the network, while testing results are given only for output layer. Results show that network is able to achieve desired energy distribution in case of image signals, sometimes with error less than 1%.

6 Conclusion

The lattice structure proposed in this paper can be characterized by the simplicity of its framework and the effectiveness of calculations. It is constructed on the basis of iterative repetition of simple two–point base operations and it can be implemented in the form of a simple pipelined scheme. As a result it gives extensive possibilities of software and hardware implementations of wavelet transforms with special indication on the integrated circuits.

Within the further development of the proposed lattice structure the relation between the orthogonal lattice structure and the orthogonal wavelet transform should be investigated in detail. It means that the class of wavelet transforms that can be implemented with the orthogonal lattice structures should be defined. From the practical point of view it is crucial to develop novel training techniques of multilayer neural networks with topologies based on the proposed lattice structure.

References

1. I. Daubechies. *Ten Lectures on Wavelets.* SIAM, 1992.
2. M. Jacymirski and P. Szczepaniak. Neural realization of fast linear filters. In Mislav Grgić, editor, *4th EURASIP-IEEE Region 8 International Symposium on Video/Image Processing and Multimedia Communications*, pages 153–157, 2002.
3. S. Osowski. *Sieci neuronowe do przetwarzania informacji.* Oficyna Wydawnicza Politechniki Warszawskiej, 2nd edition, 2006.
4. 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 Transactions on*, 45(2):217–226, February 1998.
5. B. Stasiak and M. Yatsymirskyy. Fast orthogonal neural networks. *Lecture Notes in Computer Science*, 4029:142–149, July 2006.
6. K. Stokfiszewski and P. Szczepaniak. An adaptive fast transform based image compression. In *Artificial Intelligence and Soft Computing*, pages 874–885. Springer, 2008.
7. J. Stolarek. Realization of Daubechies transform using lattice structure. In *The Collection of Scientific Works of ISDMCI 2009*, volume 2, pages 188–192, 2009.
8. R. Tadeusiewicz. *Sieci neuronowe.* Akademicka Oficyna Wydawnicza RM, 1993.
9. 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 Transactions on*, 36(1):81–94, January 1988.
10. M. Yatsymirskyy. Synteza przekształceń falkowych dla mobilnych systemów e-gospodarki. In *Monografia : Wybrane problemy elektronicznej gospodarki*, pages 273–281. 2008.
11. M. Yatsymirskyy. Lattice structures for synthesis and implementation of wavelet transforms. *Journal of Applied Computer Science*, 17(1):133–141, 2009.