The model of multifunctional neural element of intelligent systems

«The model of multifunctional neural element of intelligent systems»

by Leonid KUPERSHTEIN; Tetiana MARTYNIUK; Andriy KOZHEMIAKO; Antonina BUDA

Source:
Education - Technology - Computer Science (Edukacja - Technika - Informatyka), issue: 2 / 2013, pages: 366-371, on www.ceeol.com.

Tetiana MARTYNIUK, Andriy KOZHEMIAKO, Antonina BUDA
Vinnytsia National Technical University, Vinnytsia, Ukraine
Leonid KUPERSHTEIN
Vinnytsia Financial Economical University, Vinnytsia, Ukraine

The model of multifunctional neural element of intelligent systems

Introduction

The using of neuro- and neuro-fuzzy technology has quickly increased in the intelligent systems of wide application last time [Haykin 1999; Jones 2003]. It applies to such areas of application as robotic systems, identification systems, protection systems of telecommunications and computer networks, medical and technical diagnostic systems [Jones 2003; Osowski 2000]. The actual task is hardware implementation of neural components for such systems. For example, the neural network segment reasonably hardware implement as the neural chip for neural-fuzzy part of specialized fraud management system in telecommunications networks [Васюра 2008].

The basic neural operations and components

The determination of basic operations is an important stage in the hardware implementation of compute nodes. It allows to specify the nomenclature of necessary basic components in the future and thus ensure regularity of synthesized structures [Царев 2000].

The dot product and nonlinear transformation are required to allocate among basic neural operations [Haykin 1999; Osowski 2000]. The multiinput accumulator and multiplier line are necessary selected from a well-known basic functional components [Царев 2000] for their hardware implementation.

On the other hand, the threshold activation function or Heaviside function is the particular interest among the known activation functions with a glance of application areas [Haykin 1999; Osowski 2000]. This function, in spite of its limited capabilities, is the most frequently used in the implementation, for example, of the single layer perceptron (the threshold linear classifier) [Haykin 1999; Osowski 2000].

By the hardware implementation of the single layer perceptron both dedicated neural operations are performed sequentially in such basic node as the multiplier line, the multiinput accumulator, the comparator, which are implemented the following operations for each neutron layer:

$$S_i = \sum_{j=1}^{n} w_{ij} x_{j,} \quad i = 1, \ ..., m, \tag{1}$$

$$y_i = f(S_i) = \begin{cases} 1, \ if \ S_i \geq \theta, \\ 0, \ if \ S_i < \theta, \end{cases} \tag{2}$$

where x_j is a j-th component of input vector X; w_{ij} is a weight of j-th input i-th neuron; S_i is the state of i-th neuron; yi is output signal of i-th neuron; f(•) is the activation function, θ is a threshold; m is the number of neurons; n is a dimension of input vector X.

In most cases the operation (1) is performed on the multiplier-and-accumulator with following formation of output signal y_i using a comparator. Thereat the multioperand summation operation of paire multiplications $w_{ij}x_j$ type (1) is the most difficult for the parallelize and long on time. This multioperand summation operation is performed using a pyramidal structure of the two-input summator for the acceleration [Царев 2000, King-Sun Fu 1984].

The difference cuts processing

An alternative to such approach is the multioperand processing of the vector array elements through the difference cuts (DC) [Martynyuk 2005]. This method is based on a sequent formation vector arrays in the form of the DC A_j, starting with the first DC A_0 of dimension n in kind of

$$A_0 = \{a_{1,0}, ..., a_{n,0}\} = \{a_{i,0}\}_{i=1}^{n}, \tag{3}$$

where index j is j-th processing cycle, j = 1, ... , N.
Each current DC A_j is formed in following way:

$$A_j = \{a_{i,j}\}_{i=1}^{n} = \{a_{i,j-1} - q_j\}_{i=1}^{n}, \tag{4}$$

and

$$q_j = \min_i a_{i,j-1}. \tag{5}$$

Thus, each current DC is A_j consisted of elements $a_{i,j-1}$ of previous DC A_{j-1}, which are reduced to the minimal element q_j of this DC, i.e. is formed on the differences values in kind of $(a_{i,j-1} - q_j)$ for DC A_j:

$$A_j = A_{j-1} - q_{j,} \tag{6}$$

where A_{j-1}, A_j is a vector arrays (DC) of dimension n which are formed in the according (j-1)-th and j-th cycles.

As a result of such processing in each j-th cycle one of the elements $a_{i,j}$ of DC is set to zero, and the processing is completed when zeroize all elements $a_{i,N}$ of the last N-th cycle. Thus, the maximum number of cycles $N_{\max}$ is not exceeded the dimension of the first DC A_0, i.e.

$$N_{\max} \le n, \tag{7}$$

and the average number of cycles N_{avg} has the following dependence if there are the equal elements in the DC:

$$N_{\tilde{n}\check{\partial}} = n - \sum_{r=1}^{R} (m_r - 1), \tag{8}$$

where m_r, R is a random real number.

The difference cuts processing feature is the fact that has formed in each j-th cycle the values q_j (5) can be used to perform several operations: a) the computing of the partial sums S_j; b) forming of the a_i^S elements of the sorted vector array A_0^S ; c) the restoring of the first vector array A_0 (3).

For the computing of partial sums S_j of DC elements A_j is necessary to analyze all elements $a_{i,j}$ this DC, and to form a vector F_j of binary signs. Each element $f_{i,j}$ of vector F_j is defined as follows:

$$f_{i,j} = \begin{cases} 1, \ if \ a_{i,j} \ge 0, \\ 0, if \ a_{i,j} < 0. \end{cases} \tag{9}$$

Thus, a partial sum S_j as the sum of all nonnegative elements $a_{i,j}$ of DC A_j is computed by the formula:

$$S_j = q_j \sum_{i=1}^{n} f_{i,j} = q_j \cdot b_j, \tag{10}$$

where b_j is a number of nonnegative elements of DC A_j.

By-turn, the gradual accumulation of the partial sums S_j is allowed to obtain the sum S, or all the elements $a_{i,j}$ convolution of the first vector A_0, i.e.:

$$S = \sum_{i=1}^{n} a_{i,0} = \sum_{j=1}^{N} S_j. \tag{11}$$

At a time can be compared the partial sum S_j (10) with the external threshold θ in the next (j+1)-th cycle. Thereat it is necessary to consider the paire multiplications $w_{ij}x_j$ in the formula (1) as the element $a_{i,0}$ in the i-th input of multiinput accumulator, i.e.

$$a_{i,0} = w_{i,j} \cdot x_j. \tag{12}$$

In this case, the final value of the output signal y_i for the i-th neuron is possible to obtain not after the formation of sum S (11), which is corresponded to the sum S_i in the formula (1), but by the performed of condition [Martynyuk 2005]:

$$\Delta_j = \Delta_{j-1} - S_j \leq 0,\ j = 1,...,N, \tag{13}$$

where $\Delta_0 = \theta$.

Thereby, the response time of each i-th neuron of single layer perceptron, i.e. the formation time of output unit signal y_i (2) is not depended on and is not matched with a response time of other neurons. As a result, the number of cycles N for each neuron can vary from 1 to n, taking into account the values m_r, R and especially value θ of threshold processing by DC [Martynyuk 2005]. Such response is adequated to the biological neuron response [Haykin 1999], since the lower the value of the threshold θ the faster the neuron response.

Realizations models

The computer simulation of describing method of neural-like data processing by DC was performed. The dependence of the average number N_{avg} of the cycles of array number processing from the array dimension n and mean square deviation σ of array elements and threshold value θ was researched. The graph of function $N_{avg}=F(n,\sigma)$ is presented on fig. 1 [Васюра 2008].

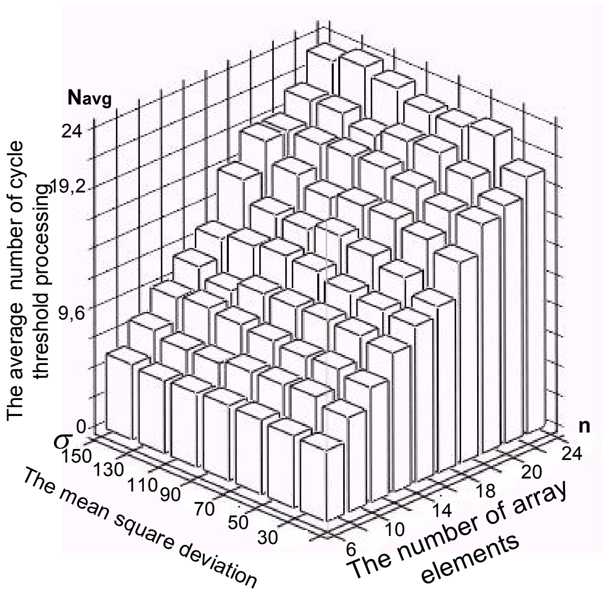

Fig. 1. The graph of function $N_{avg}= F(n,\sigma)$ for the neural-like data processing by DC

The threshold value for 6 elements array is chosen 3000, for 8 elements is chosen 4000 etc., that are corresponded to the expression ($\mu \cdot n$), since the expectation value is $\mu = 500$ for all arrays. The analysis of graphic dependence is proved neural processing effectiveness by DC, since the number of the processing cycles is less, than the traditional sequential summation method. In addition, the availability of equal operands in the input array is increased the processing speed on 10–30% [Васюра 2008].

The expressions for the values $a_{i,j}$ (4) and S (11) are indicated the recursion presence in the main equations (4) and (11) when the vector data array processing by DC. This is allowed to create an appropriate linear systolic array [Timchenko 1999], using a known methodology of synthesizing systolic arrays [Kung 1988]. By-turn, the linear systolic array is provided a natural expansion (increase) computation modules of such array when its hardware implementation [Kung 1988].

The structure such array is realized in the form of the parallel-pipeline processor as the neural element on the base of progressive base elements of neural computers – the programmable logic devices. The complex programmable logic device XC95288XL-6-BG256 is used. The implementation results are proved the ability of realization and effective using multilayer neural networks or their fragments with the multiinput threshold neurons, what work by the DC method, on the PLD Xilinx base of large logical capacity [Васюра 2008]. Analyzing the results of implementation on PLD Xilinx of neural chip with fragment layer of neural network on the base of the parallel-pipeline processor, the maximal time of threshold processing are estimated, which equal 0.23 μs [Васюра 2008]. It allows to concede that such neural chip on the base of the proposed parallel-pipeline processor will function in real time.

Literature

Haykin S. (1999), *Neural Networks: A Comprehensive Foundation*, Second Edition. Prentice Hall, Inc.

Jones T. (2003), *All Application Programming*, Charlies River Media, Inc.

King-Sun Fu (1984), *VLSI for Pattern Recognition and Image Processing*, Springer-Verlag Berlin Heidelberg.

Kung S.Y. (1988), *VLSI Array Processors*, Prentice Hall, Inc.

Martynyuk T.B. (2005), *A Threshold Neuron Model Based on the Processing of Difference Slices. Cybernetics and Systems Analysis*, Vol. 41, №4, pp. 541–550.

Osowski S. (2000), *Sieci neuronowe do przetwarzania informacji*, Warszawa.

Timchenko L.I. (1999), *Approach to Organization of the Multistage Scheme of Systolic Calculations*/L.I. Timchenko, T.B. Martyniuk, L.V. Zagoruyko//*Engineering Simulation*, Vol. 16, pp. 581–590.

Васюра А.С. (2008), *Методи та засоби нейроподібної обробки даних для систем керування*/ А.С. Васюра, Т.Б. Мартинюк, Л.М. Куперштейн – Вінниця: УНІВЕРСУМ – Вінниця, – 175 с.

Царев А.П. (2000), *Алгоритмические модели и структуры высокопроизводительных процессоров цифровой обработки сигналов.* – Szczecin, Informa, – 237 с.

Abstract

The article shows the features of realization of multioperand processing in neural structures on the base of difference cuts, that allow to expand functional capabilities and to reduce time consumptions in neural processing. The structural organization of the parallel-pipeline processor for neural-like vector data processing on the DCs base are proposed. This parallel-pipeline processor on CPLD base are implemented, which allow realize neural chip with a fragment of the neural network layer.

Key words: neural element, threshold parallel processing, parallel-pipeline processor, neural network, perceptron, neural chip, difference cut, programmable logic device.