SYNTHESIS OF THE RNS-PROCESSOR FOR A NEUROCOMPUTER NETWORK TO PROCESS VIEWING AEROSPACE INFORMATION IN PROBLEMS OF ECOLOGICAL MONITORING

S.A. Ikonnikov

A.F. Mozhaiskii Military Space-Engineering Academy
Received May 30, 1995

We discuss here general principles of a neuronetwork functioning. A way for determining the correction vector for the weighting factors matrix is described based on linear reconstruction algorithm. An algorithm for RNS-processor optimal structure synthesis is proposed.

INTRODUCTION

The conceptions of using neural networks and nontraditional arithmetic based, for instance, on the residue-number-system (RNS) seem to be a very promising direction of investigations capable to lead to essential changes in the principles of creating computer systems for processing the aerospace viewing information.

Neural networks are often considered as one of the directions in developing artificial intellect and as an alternative to digital computers and algorithmic programming in solving problems of ecological monitoring. These definitions cannot be treated as absolutely correct, but preliminary investigations make it possible to judge that neural networks and the RNS-arithmetic can drastically change the situation when digital processing and algorithmic programming do not provide serious achievements. According to Ref. 11, some university laboratories (Illinois, California) and corporations (INTEL, G.ELECTRIC) actively conduct scientific research aimed at reduction of methods and means of neural networks to such serious applied problems as pattern recognition and reading of symbols.

In perspective, such an approach will give an essential increase in functional possibilities of computer design which must include various methods of information processing.

1. GENERAL PRINCIPLES OF NEUROCOMPUTER NETWORK OPERATION

Since many problems solved by a neurocomputer are connected with the processing of prime pithy information such as images, simple models of biological systems are chosen as architectural principles for constructing neural networks.

Neural network is an adaptive network which consists of inputs, outputs, and processor elements (PE) and is capable to minimize the cost function of the result desired.

Practically any type of coded information is permissible at the inputs and outputs depending on an applied problem.

PE "weighs" the input signal (r_i), i.e., finds its synoptic weight ($r_i W_i$) what gives the network a possibility to map input data at the output adequately and accurately; then the data are summed and pass through the NEURON (Fig. 1*a*). The neuron executes the function $F(r_i W_i)$ which can be linear, step-wise, or nonlinear sigma-shaped and it is chosen so that rapid convergence is achieved and the finite result R is obtained (Fig. 1*b*).

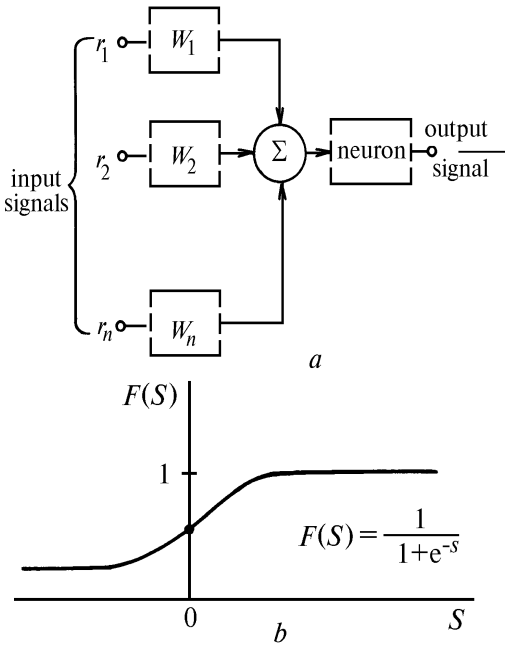

FIG. 1. Processor element structure (a) and sigma-shaped neuron function (b).

The networks already existing have two key architectural characteristics: every input is connected

0235-6880/96/01 74-06 $02.00

with every PE and the outputs of one level are inputs the next level in a network with more than one level (Fig. 2).

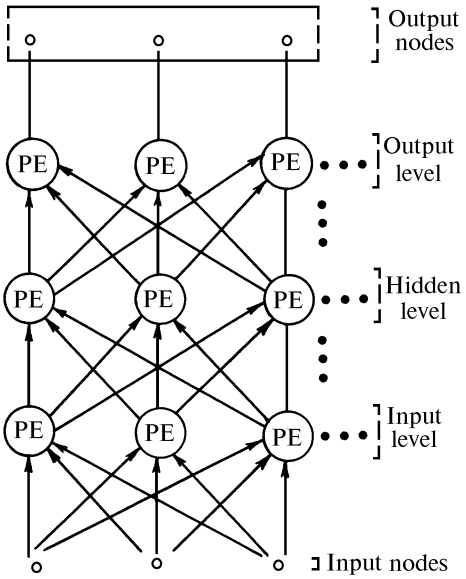

FIG. 2. Multilevel neural network structure

In order to find correct weights, one uses the training of a neural network, namely, error and trial method. A successful training is possible if the data have a correct format and cover the required number ranges, i.e., training data must include the whole range of the initial data for the network. Statistical analysis algorithm realized in a neural network will approximate a curve with the data interpolation in a domain not coinciding with the data values used in the network during the training. In other words, the data are "pumped" through the network during training process in order to catch correct values of synoptic weights. After training all input data can be mapped into a correct output values.

From the viewpoint of mathematical modeling, neural network is a dynamic system which can be simulated as a system of coupled differential equations. Such a system is potentially unstable because small oscillations of the weighting factor can lead to appearance of uncontrolled generation, spikes, standing and running waves and tend to a chaotic state.

If a network with its state changes is considered as an energy surface, the stable states will correspond to energy minima. Every stored pattern of the network, e.g. data file, corresponds to a stable state or a local minimum. This pattern is associated with an input pattern; so entering of the latter into the neural network will cause the network's choice of the corresponding stored pattern, similarly to the associative memory.

However, a complication occurs that in different models several energy minima can correspond to one and the same input pattern, and that means that one not necessarily obtains a correct output result by processing input data.

In order to decrease the probability of a false computer operation, one develops or uses already existing specialized algorithm of neural network training for each class of the problems to be solved. At present the training "PE ("ack Propagation of Error) algorithm [10] is most spread, its accuracy being 94.1%. The network training with this algorithm starts with the assignment of small random values to synoptic weights. For the first set of input vectors one obtains wrong output vectors. However, since correct values of the output vectors are known, one can compute differential values which are said to be correcting coefficients or delta-mistakes (Q).

The obtained Q-values for each synoptic weight W_i propagate backward along the network.

With their passage through the neural network levels, the values corresponding to every synoptic weight W change. Finally, the matrix of synoptic weights

$$W_{ij} = \begin{vmatrix} W_{11}, & W_{12}, & \dots, & W_{1n} \\ W_{21}, & W_{22}, & \dots, & W_{2n} \\ . & . & . & . \\ W_{m1}, & W_{m2}, & \dots, & W_{mn} \end{vmatrix}, \qquad (1.1)$$

where i is the nth synoptic weight of the jth level, and j is the mth neural network level, is fitted so that the network is capable to compute correct output vector for the first input vector. These values of weight factors are used in processing the second set of input vectors.

The process is repeated, and the matrix of synoptic weights W_{ij} is modified again and again until the optimal configuration for the whole data field is obtained.

In order to provide higher accuracy of calculating the matrix W_{ij} and depending on the class of problems solved, more sophisticated training algorithms are often used.[10] The below algorithms of training neural networks are among most widely spread:

LVQ – Learning Vector Quantization, 95.7%;

STLVQ – Shift-Tolerant Learning Vector Quantization, 98%;

R"F – Radial "asis Functions, 99%;

RCE – Restricted Coulomb Energy, 99%;

"DT – "ias Data Training, 99%.

Two basic conceptions of neural network construction, namely, the establishment of the links set and execution of iterations for adequate mapping of the input parameters to the output ones cause greater intensity of calculations necessary for solving certain problems. Therefore, whenever possible one tries to reduce the neural network after its initial configuration is determined. "ut the advantages reached by customizing and reduction of the network are usually very limited. So the computer hardware for speeding up computations is very often required.

2. DETERMINATION OF THE CORRECTION VECTOR (*Q*) USING LINEAR RECONSTRUCTION ALGORITHM

Tough restrictions to the allowable time for neural network learning and information processing, big bulks of transferred information, and rather small values of the noise factor (*Q*-mistake) demonstrate the expedience of applying the linear reconstruction algorithm. The convolution of actual readings with the weight function of the correcting element is the central and most laborious operation of linear algorithms.[2]

The properties of algebraic structures over which the convolution is calculated are of great importance for synthesizing fast convolution algorithms. In the case under study, we propose to consider the convolution over the residue class rings Z_m. These algebraic structures seem to be the most optimal for the neural network training and operation.

A linear convolution of an L-point sequence $\{r_i\}$ with an M-point sequence $\{W_i\}$ yields a dilated $K=L+M+1$- point sequence $\{S_i\}$.

Compactly, a one-dimensional linear convolution $S=r\otimes W$ of two integer-valued sequences $\{r_i\}$ and $\{Wi\}$ has the form

$$S(x) = r(x)\ W(x), \tag{2.1}$$

where the degrees of polynomials are: deg $r(x)$ $L-1$, deg $W(x)$ $M-1$, deg $S(x)$ $L+M-2$.

It is shown in Ref. 2 that the coefficients S of the polynomial $S(x)$ can be obtained from the expression

$$\sum_{i=0}^{N-1} S_i\, x^i = \sum_{i=0}^{L-1} r_i\, x^i \sum_{m=0}^{M-1} W_m\, x^m, \tag{2.2}$$

from which we have

$$S_i = \sum_{n=0}^{N-1} r_n\ W_{i-n}, \tag{2.3}$$

where r_i are discrete quantized readings from the first integer-valued sequence, e.g., the readings of the input signal; W_m are discretes of the second sequence, e.g., transmission characteristics of the synoptic weight; $i, n = 0, 1,\ldots, N$; $l = 0, 1, \ldots, L-1$; $m = 0, 1, \ldots, M-1$.

In order the coefficients W_{i-n} be always meaningful, one should periodically, with the period N, extend their values. This means that if the index $(i-n)$ does not belong to the interval $0<i-n<N-1$ one should add or subtract a number multiple of N for the inequality $0 < i - n + mN < N - 1$ to be fulfilled.

Then one can use the values

$$W_{i-n\,\pm\,mN} = W_{(i-n)\ (\mathrm{mod}\ N)}$$

when calculating the convolution.

The convolution $S=r\otimes h$ obtained in such a way is said to be a **cyclic convolution** and can be written in a polynomial form as a residue of a linear convolution to the polynomial $X-1$ modulus

$$S(x) = r(x)\ W(x)\ (\mathrm{mod}\ x-1) \tag{2.4}$$

or

$$\sum_{i=0}^{N} S_i\, x^i = \sum_{i=0}^{N} r_i\, x^i\ W_i\, x^i\ (\mathrm{mod}\ x-1), \tag{2.5}$$

what implies

$$S_i = \sum_{n=0}^{N} r_n\ W_{(i-n)\ (\mathrm{mod}\ N)}. \tag{2.6}$$

The sequences differing only by a cyclic displacement will correspond to the expression $W_{(i-n)(\mathrm{mod}N)}$ for different values S_i, $n = 0, 1, \ldots, N-1$. Therefore, the calculation of the difference $(i-n)(\mathrm{mod}\ N)$ results in periodic permutations of the elements W_i of the sequence of readings. This cyclicity is most apparently seen if the convolution is written in the matrix form:

$$\begin{bmatrix} S_0 \\ S_1 \\ \cdot \\ \cdot \\ \cdot \\ S_{N-1} \end{bmatrix} = \begin{bmatrix} W_0 & W_{N-1} & \ldots & W_1 \\ W_1 & W_0 & \ldots & W_2 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ W_{N-1} & W_{N-2} & \ldots & W_0 \end{bmatrix} \times \begin{bmatrix} r_0 \\ r_1 \\ \cdot \\ \cdot \\ \cdot \\ r_{N-1} \end{bmatrix}, \tag{2.7}$$

where $\begin{bmatrix} S_0 \\ S_1 \\ \cdot \\ \cdot \\ \cdot \\ S_{N-1} \end{bmatrix} = S$ is the column vector of the convolution results; $\begin{bmatrix} W_0 & \ldots & W_1 \\ \cdot & \cdot & \\ \cdot & & \cdot \\ \cdot & & \cdot \\ W_{N-1} & \ldots & W_0 \end{bmatrix} = W$ is the matrix of readings of the synoptic weight transmission characteristics; $[r_0\ r_2\ \ldots\ r_{N-1}]^T = R$ is the column vector of the current data sequence.

If the cyclic convolution is used for calculating linear convolution, the influence of the periodicity should be compensated. For computing the whole sequence $\{S_i\}$ by a cyclic convolution the length of the latter should be equal to N points. "esides, the sequences $\{r_i\}$ and $\{W_i\}$ should be continued up to N-point sequences. For this purpose it is sufficient to add the corresponding number of zeros to $\{r_i\}$ and $\{W_i\}$ and to construct the sequences

$$r_i = \begin{cases} r_i, & 0 \le i \le L-1, \\ 0, & L \le i \le N-1; \end{cases} \tag{2.8}$$

$$W_i = \begin{cases} W_i, & 0 \le i \le M-1, \\ 0, & M \le i \le N-1. \end{cases} \tag{2.9}$$

Then the values of cyclic and linear convolutions coincide on the interval $0<i<N-1$, i.e., in the polynomial form,

$$S'(x) = r'(x)\ W'(x)\ (\mathrm{mod}\ x - 1) = \\ = r(x)\ W(x) = S(x). \qquad (2.10)$$

In order to calculate the coefficients of the convolution over the residue class ring Z_m, one uses the expression

$$S_i = \left(\sum_{n=0}^{N-1} r_n\ W_{i-n} \right) (\mathrm{mod}\ m), \qquad (2.11)$$

for the linear convolution and

$$S'_i = \left(\sum_{n=0}^{N-1} r_n\ W_{(i-n)(\mathrm{mod}\ N)} \right) (\mathrm{mod}\ m) \qquad (2.12)$$

for the cyclic convolution, respectively.

The operation to the modulus m is unnecessary and all the calculations are performed by laws of usual arithmetic until the modulus m is greater than all the integers used in calculations.

Otherwise, the Z_m-arithmetic, i.e., the arithmetic to the modulus m comes into force. The values $S(S')$ obtained in such a way allow one to determine the correction vector $[Q_{ij}]$ for the weighting coefficient matrix $[W_{ij}]$ rather simply on the basis of the formulas (1.1) and (2.7) what makes it possible to create efficient specialized numerical algorithms of the neural network training.

3. THE ALGORITHM OF OPTIMAL RNS-PROCESSOR SYNTHESIS

To calculate cyclic convolution, the modulus m can be chosen so that there are sufficiently convenient convolution procedures in the ring Z_m.

It is desirable to use the advantages of the residue number system (RNS) for these purposes. RNS is a number system in which numbers are represented as a set of non-negative residues $a_1, a_2, \ldots, a_n$ by mutually prime moduli (bases) $m_1, m_2, \ldots, m_n$:

$$A = \{a_i\ (\mathrm{mod}\ m) = 1,\ n\}. \qquad (3.1)$$

The condition of pair-wise mutual simplicity of the chosen moduli $\{m_i\} = 1$, $i = \overline{1, n}$, provides an unambiguous representation of the number A in the range D which is equal to the product of these moduli

$$D = \prod_{i=1}^{n} m_i. \qquad (3.2)$$

Since the bases $\{m_i\}$ are mutually independent by definition, there exists the following isomorphism of the direct sum

$$x \cdot y = \sum \oplus\ (x \cdot y) \qquad (3.3)$$

for making the arithmetic operations $\cdot\ \{+, -, \times\}$. This isomorphism defines a one-to-one correspondence between a positive integer $a < m$ and a residue vector $(a_1, a_2, \ldots, a_n)$. The isomorphism enables one to process quantized video data in parallel and independently in every of the rings Z_{mj} by laws of the RNS-arithmetic. The residue

$$S_{ij} = S_i\ (\mathrm{mod}\ m) \qquad (3.4)$$

is calculated in every ring Z_{mj} and the integer result of the convolution, i.e. the result in the ring is obtained by formula

$$S_i = \left(\sum_{i=1}^{n} S_{ij}\ M_j\ N_j \right) (\mathrm{mod}\ m), \qquad (3.5)$$

where $M_i = m_i / m_j$, $M_j N_j = 1 (\mathrm{mod}\ m)$.

Thus the calculation of the convolution can be realized both over the ring Z_m and over the direct sum of rings $Z_{m_1} + Z_{m_2} + \ldots + Z_{m_n}$. Figure 3 shows the calculation scheme for the convolution of the sequences $\{r_i\}$ and $\{Wj\}$ over the ring Z_m and over the direct sum $Z_{m_1} + Z_{m_2} + \ldots + Z_{m_n}$ of residue class rings by pairwise prime moduli $m_1, m_2, \ldots, m_n$.

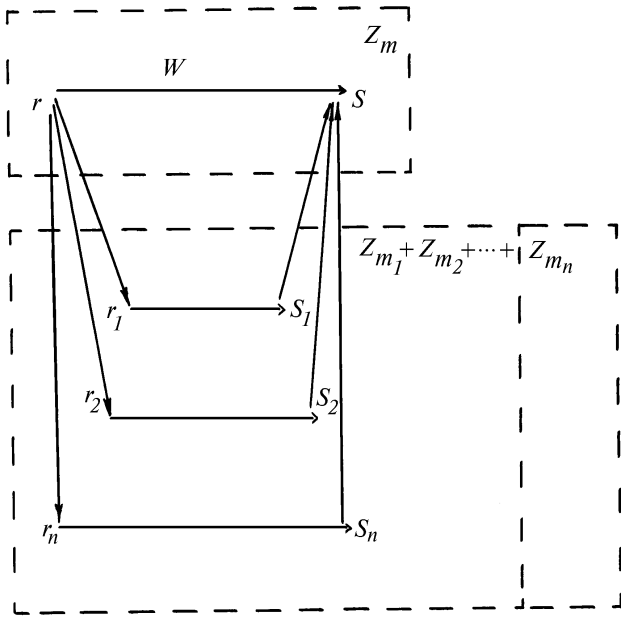

FIG. 3. Scheme of calculation of the convolution S = r W over the direct sum $Z_{m_1} + Z_{m_2} + \ldots + Z_{m_n}$

So it is assumed that the arithmetic operations can be executed over several finite rings and the result of the direct sum can be mapped into a larger modulus $\prod_{i=1}^{n} m_i$ ring. All the calculations are independent of others,

so they can be done in parallel what just takes place in the RNS-processor, i.e., the concurrence principle is realized almost completely.

The structure of the RNS-process is extremely simple; it is a regular matrix where the computations are reduced to a simple choice of the result from the jth node of the table (matrix) analyzed.

"y the jth node we mean the part of the matrix where the relation

$$C_i = \Phi(a, b)$$

takes place with a and b belonging to the same range, i.e., $D \le a^2$.

It is obvious that in the general case the number of nodes K of the matrix is determined by $K_i = m_i^2 + m_i$, for the whole set of moduli $\{m_i\}$

$$K = \sum_{i=1}^{n} (m_i^2 + m_i). \qquad (3.6)$$

Here each possible combination of the input values is realized, while the outputs of nodes are united by the right-hand side Φ.

The main disadvantage of the table version of the RNS-processor is a great increase in the volume of the equipment needed (the node number K) with increasing moduli values $\{m_i\}$.

In this connection, there arises a problem on optimal choice of $\{m_i\}$ providing the hardware expenditures minimization.

Analysis of existing criteria of the moduli set $\{m_i\}$ estimation performed by the author in Ref. 3 enables one to formulate possible approaches to the choice of moduli sets with the properties given beforehand and to develop practical methods of choosing the set $\{m_i\}$ optimal from the viewpoint of minimizing costs.

It is known that the amount of hardware for constructing the RNS-processor depends on the number of bases and their values. "esides, if the terms K and D in Eq. (3.6) correlate, the optimal amount of hardware can expressed by some function $K_{opt}=f(n_{max}, V)$, where V is the coefficient of range overlapping.

If one turns to Eq. (3.2), it is now expressed in the form

$$D = \left(\prod_{i=1}^{n} m_i \right) V, \qquad (3.7)$$

therefore, if one of the moduli m from the initial set $\{m_i\}$ is divided by V, the set $\{m_i\}$ will overlap the range D at $V \approx 1$.

Let m_n be chosen as a modulus to be diminished. This is caused by the fact that this base has the largest weighting coefficient in Eq. (3.2).

Since m_n can be only integer, the obtained base m'_n should be rounded up to the higher integer, i.e.,

$$m'_n =]m_n / V[. \qquad (3.8)$$

"esides, m'_n should satisfy the condition of pair-wise simplicity

$$(m_i, m_j) = 1, \text{ where } i, j \in Z. \qquad (*).$$

If m'_n does not satisfy the condition (*), it should be increased by one until this condition is fulfilled. Thus, the obtained set $\{m_1 + m_2 + \ldots + m'_n\}$ will be optimal for D in the sense of hardware costs.

The algorithm of choosing optimal modulus sets $\{m_i\}$ of the RNS-processor is as follows:

1. Assume the operation range D of the RNS-processor.

2. From the set $\{m_i\}$ of prime numbers, starting from the third, choose such that satisfy the condition

$$\prod_{i=1}^{n} m_i > D.$$

3. Find the coefficient of relative overlapping from the condition

$$V = \left(\prod_{i=1}^{n} m_i \right) / D.$$

4. Divide the largest base m_n from the set $\{m_i\}$ by V and round the result to the higher integer

$$m'_n =]m_n / V[.$$

5. Test m'_n for the fulfillment of the condition of pair-wise simplicity $(m_n, m'_n) = 1$.

6. The condition $(m_n, m'_n) = 1$ is satisfied:

a) if yes, the optimal set is found;

b) if no, increase m'_n by 1 until the condition is satisfied.

The proposed algorithm of choosing the moduli set $\{m_i\}$ is formulated from the viewpoint of the standard RNS-processor synthesis.

In order to estimate the amount of hardware and its optimization quantitatively in choosing the optimal $\{m_i\}$, the coefficient H is mathematically defined in Ref. 4. It is considered as the ratio of the node (processor elements of the matrix) number K_s of the synthesized RNS-processor to that of the conventionally ideal RNS-processor (K_{id})

$$H = \frac{K_s}{K_{id}} = \frac{\sum_{i=1}^{n} (m_i^2 + m_i)}{n \cdot D^{1/n} (1 + D^{1/n})}. \qquad (3.9)$$

CONCLUSION

The numerical experiments performed in Ref. 4 demonstrate that the proposed algorithm of the RNS-processor synthesis allows one to design a substantially optimal neural network from the viewpoint of hardware minimization and to estimate quantitatively the hardware costs.

"ut it seems to the author that optimization of the RNS-processor structure can be extended if the multistage residue number system (RNS-arithmetic) is applied.

REFERENCES

1. R.E. Blahut, *Fast Algorithms for Digital Signal Processing* [Russian translation] (Mir, Moscow 1989), 448 pp.
2. A.I. Brodovich and E.I. Shabakov, *Integer Processing of Video Signals and Images* (A.F. Mozhaiskii Military Space-Engineering Academy, St. Petersburg, 1993), Vol. 2, 152 pp.
3. S.A. Ikonnikov, in: *Proceedings of the International Conference "Informatics-94"*, (SPIIRAN, St. Petersburg, 1994), Vol. 3, pp. 30–33.
4. S.A. Ikonnikov, in: *Proceeding of the Scientific Practical Conference*, A.F. Mozhaiskii Military Space-Engineering Academy, St. Petersburg (1995), p. 220.
5. V.G. Labunets, *Algebraic Theory of Signals and Systems. Digital Processing of Signals* (Krasnoyarsk State University, Krasnoyarsk, 1984), 244 pp.
6. T.H. McClellan and C.M. Rader, *Number Theory in Digital Signal Processing* [Russian translation] (Mir, Moscow, 1983), pp.8–59, 186–202.
7. B.V. Titkov and E.I. Shabakov, Tekhnika Sredstv Svyazi. Ser. Tekhnika Televideniya, No. 4, 26–33 (1985).
8. D.E. Knuth, *The Art of Computer Programming,* Vol. 2, *Seminumerical Algorithms* [Russian translation] (Mir, Moscow, 1976), pp. 74–76, 411–419, 476.
9. A.S. Batrakov, A.I. Brodovich, and E.I. Shabakov, SPIE **1961**, 456–466 (1993).
10. Y. Shadle, Electronics Design, No. 4, 51–58 (1993).
11. Y. Till, Electronics Design, No. 1, 49–63 (1989).
12. C. Vollum, Electronics Design, No. 2, 38–51 (1989).