Development of Telecommunication System Units in Parallel Neural Network Basis

A A Smirnov¹, P A Sakhnyuk²

¹North-Caucasus Federal University, Institute of Mathematics and Natural Sciences, Pushkina str. 1, Stavropol, Russia Federation, 355009
²National Research University Higher School of Economics, 20, Myasnitskaya str., Moscow, 101000

E-mail: shursun@mail.ru

Abstract: The article suggests the integration of a neural network as a parallel element base in a telecommunication system. In this case, the ability to learn or adapt to external conditions is applied as the main advantage. For telecommunication systems in conditions when it is possible, this ability will improve noise immunity, reliability, operability, etc. The article considers an example of the integration of a neural network into a discrete matched signal filter. It is noted that the use of parallel mathematical methods in signal processing leads to the maximum effect of increasing the quality parameters of such telecommunication elements.

Keywords: Neural Network, Matched Filter, Residue Number System, Learning Algorithms, Activation Function, Telecommunications.

1. Introduction

Nowadays, in the era of global informatization and multiple increases in the volume of information transmission, data transmission networks develop rapidly. The problems such as “Integration of various types of networks”, “Optimization of the information under existing channels”, “Adaptation of signals regarding the quality of channels”, etc. [1-5] arise on the agenda.

Along with the increase in the volume of information, a problem of the processing rate, speed and quality of transmission arises. One of the ways to increase these indicators is to enhance the clock rate or operating frequencies of telecommunications elements. However, at present, this direction has been exhausted by the technological limitations of the conductor dimensions in units of nanometers. Thus, the only way to increase the quality parameters is to parallelize the process of transmission and processing of information [6-8, 13]. To construct systems for transmitting and processing parallel data, “parallel mathematics” is required, for example, a system of residual classes, a parallel element base, such as neural networks, is needed. Parallel communication channels are also needed based on various compression methods, algorithms, etc. This work is devoted to the introduction of mass parallelism in telecommunications. Let us consider the directions of possible integration of neural networks into telecommunication elements [2, 7, 8-11, 14, 15].
2. Possible directions of the neural network technologies integration

To determine the possibility of using neural networks in telecommunication elements, we consider the basic mathematics of neural networks and compare them with mathematical models of telecommunication elements [2, 4, 15].

2.1. Elemental base of neural networks

The basis for constructing neural networks is a neuron, as a physical prototype of a human brain’s biological neuron. Functionally, it is represented in Figure 1, where \( x_i \) is neuron input vector (outputs of previous neurons); \( w_i,j \) is weight vector; \( F(u) \) is activation function (determined by the type of solvable problem).

\[
\text{NET} = \sum_{i=1}^{n} x_i w_{i,j}
\]

\[
F(\text{NET}) = F\left(\sum_{i=1}^{n} x_i w_{i,j}\right)
\]

\[
\text{OUT} = \frac{1}{1 + e^{-(\sum_{i=1}^{n} x_i w_{i,j})}}
\]

The activation function is considered as a nonlinear amplification characteristic of an artificial neuron that mathematically implements the following operation:

\[
\text{OUT} = F\left(\sum_{i=1}^{n} x_i w_{i,j}\right).
\]

The gain is calculated as the ratio of the increment of the output value to the small increment of the input value that caused it. It is expressed by the slope of the activation function curve. There are many activation functions, for example, a sigmoidal activation function (Figure 2), which is described by the expression:

\[
F(u) = \left[1 + \exp(-u)\right]^{-1}.
\]

Another activation function adapted to the system of residual classes is a piecewise continuous function (Figure 3), which operates in the ring of integers:
\[ F_2(u) = \frac{10}{1 + e^{-0.5u}} \]  \hspace{1cm} (3)

This function allows smoothly to adjust the weight coefficients, eliminating the influence of random deviations in the received signal, i.e., it is more suitable for cases of constant interference.

In the adder, the input vectors of the neuron are multiplied by the vectors of weight coefficients, this sum is equal to: \( NET = x_1w_1 + x_2w_2 + \ldots + x_nw_n = \sum_{i=1}^{n} x_i w_j \). Then the function takes the form:

\[ OUT = F(NET) = F(\sum_{i=1}^{n} x_i w_{i,j}) = \left[ 1 + \exp(-\sum_{i=1}^{n} x_i w_{i,j}) \right]^{-1}. \]  \hspace{1cm} (4)

One of the advantages of neural networks is their ability to self-learn. It means the ability to adapt weights \( w_{ij} \) to the actual conditions of propagation of radio waves. The difference between real conditions and the analytical model is most clearly expressed in those models where one of the input factors is the external environment, internal thermal noise, etc. The network is trained to give the desired set of outputs for some set of inputs. During training, the weights of the network gradually become such that each input vector produces an output vector. There are learning algorithms with and without a teacher. Learning with a teacher assumes that for each input vector there is a target vector representing the desired output. Comparison with the standard allows you to determine the error and adjust the weight coefficients of the neural network.

Learning without a teacher does not need a target vector for exits and does not require comparison with predetermined ideal answers. The training set consists only of input vectors. The training algorithm adjusts the network weights so that consistent output vectors are obtained, i.e. so that the presentation of sufficiently close input vectors gives the same outputs. Most modern learning algorithms grew out of the concepts of D. Habb, who in 1949 proposed the law of learning, which was the starting point for the learning algorithms of artificial neural networks.

A generalization of the neuron learning algorithm (delta rule) is formulated as

\[ w_{i,j}(n+1) = w_{i,j}(n) + \Delta_{i,j}, \]  \hspace{1cm} (5)

where \( w_{i,j}(n+1) \) is weight value after correction, \( w_{i,j}(n) \) is value weight \( i \) before correction, \( \Delta_{i,j} \) is correction associated with \( i \)-th input \( x_i \).

\[ \Delta_{i,j} = \alpha \delta y_j, \]  \hspace{1cm} (6)

where \( \alpha \) is coefficient «learning rate», \( \delta \) is the difference between desired or target output \( T \) and real output \( Y \).

\[ \delta = (T - Y). \]  \hspace{1cm} (7)
It follows that $\delta$ is multiplied by the value of each input $x_i$ and this product is added to the corresponding weight. The learning coefficient is selected depending on the type of problem being solved and the type of signal. Obviously, the learning algorithm (5 - 7) can be implemented in the system of residual classes. For this, we will determine the weighting coefficients of the neural network according to (5), calculated on the basis of the corresponding mutually simple bases $p_i$ and taking into account the proposed activation functions (Fig. 3). According to (6), when choosing an integer value of the learning speed and the initial values of the weight coefficients, the final learning result will also be integer.

$$w_j(n+1) = [w_j(n) + \alpha \Delta_{i,j}], \quad w_j(n+1) = (w_j(n) + \alpha \Delta_{i,j}) \mod p_i. \quad (8)$$

In the matrix of weights, we get integers, not superior to the corresponding bases.

Note that it is impossible to see in neurocomputers a panacea for solving all problems. They are designed to solve difficult formalized and non formalized problems. As it turned out, the majority of such problems in the theory of telecommunications. Moreover, to formalize them, they are forced to resort to serious restrictions (the depth of feedback, the width of the signal spectrum, the use of standard radiation patterns, the limitation of the correcting capabilities of the code, etc.), to the detriment of quality.

Let us outline the main directions in the theory of telecommunications that require the use of neural networks and the development of the analytical foundations of their construction.

2.2. Neural networks in digital filtering

A digital filter is a device or signal processing algorithm implemented programatically on an electronic computer. Mathematically, the operation of a digital filter is described by the equation in finite differences:

$$Sv(nT) = -\sum_{i=0}^{m-1} a_i \cdot Sv(nT - jT) + \sum_{j=0}^{k-1} b_j \cdot S(nT - jT), \quad (9)$$

where $a_m$ and $b_k$ are coefficients, determined by signal processing tasks; $Sv(nT)$ is output values of discrete signal; $S(nT)$ is output signal; $m$ is feedback depth coefficients. Under condition $m = k = 1$, a circuit that implements (9) can be performed by a single feedback neuron [14]. Moreover, multipliers of neural network coefficients will play as coupling amplifiers. As can be seen, expression (9) can easily be implemented in a neural network basis, since it contains the basic operations of this basis - addition and multiplication (4). An even greater effect is achieved when implementing digital filters in parallel mathematics.

2.3. Signal correlation filtering

The synthesis of correlation filters is based on the correlation properties of a signal having a large central response and small side lobes. The decision to receive a signal is made when the correlation response is exceeded:

$$R(k) = \sum_{i=0}^{n} h_i x_{i+k}, \quad (10)$$

where: $h_i$ is impulse characteristic; $x_{i+k}$ are elements $n$-th signal with $k$-th shift, some threshold analogue of the activation function.

Correlation analysis in the residue number system is equivalent to correlation analysis in ordinary mathematics of a positional number system. The limitation in the development of correlation filter technologies is the number of admissible calculations (n) with simultaneous operation in real-time. In addition, there is the problem of creating continuous delay lines. The elimination of these problems is possible when implementing filters on a neural network basis inspired by the residue number system. This will allow for correlation analysis of any complexity signals in real-time. Analysis (10) shows that matched filters are a special case of neural
networks. That is, such schemes can be implemented as a result of training a neural network, however, unlike a rigid circuit implementation, the latter are capable of adapting to real interference conditions.

2.4. The harmonic analyses, Fourier transform

The spectral analysis of discrete signals and filters is based on a discrete Fourier transform:

$$C(k\Omega) = \frac{1}{N} \sum_{n=0}^{N-1} S(nT) \exp(-j2\pi k) \quad S(nT) = \sum_{k=0}^{N-1} C(k\Omega) \exp(jnk\Omega T),$$

(11)

where: $S(nT)$ is original discrete signal; $N$ is the number of counting in the signal cycle; $T$ is discretization interval, $\Omega = 2\pi/(NT)$ is the main frequency of transform, $C(k\Omega)$ are coefficients of Fourier series.

The number of row coefficients is equal to the number of discrete signal samples (N). A limitation of existing circuits is the number of samples in signal N. The increase in the number of samples is limited by the capabilities of modern computing facilities. Therefore, fast Fourier transform algorithms are used. In this case, the analysis error either increases or the technical implementation of the analysis devices is complicated. An analysis of expression (11) shows that the discrete Fourier transform also contains the basic operations of neural networks (addition, multiplication). Therefore, the implementation of spectral analysis devices can easily be performed by neurocomputers and parallel mathematics.

2.5. Noiseless coding

Noiseless coding is based on the use of redundancy in the transmitted information. Information symbols are divided into intersecting groups and a verification symbol is assigned to each group so that each group has an even number of units. The mathematical form of this procedure is implemented by:

$$b_{i_{\text{var}}} = \sum_{i=1}^{k} \gamma_{i_{\text{var}}} a_{i}, \mod 2,$$

(12)

where $a_{i}$ is information symbols; $b_{i}$ is verification symbols; $\gamma_{i_{\text{var}}}$ is determined by the coding equation.

Analysis (12) shows that the coding scheme can also be implemented by a neural network in the residue number system based on modular operations. Such an implementation will allow receiving signals as a whole, combining the first and second decisive schemes.

2.6. Adapted communication channels

There is a direction of research on the creation of adaptive communication channels based on the “backward channel” method in real-time. The method is based on considering the effects of phase distortions in the front of the received wave in real-time and their compensation (for example, decameter channels by the inverse ionosphere method). This problem is formulated only for a multipath model of the communication channel.

Obviously, this problem relates either to non-formalizable or difficult to formalize. Neurocomputers are designed specifically for solving such problems.

2.7. Phased array

The synthesis and operation of phased array is based on the common-mode addition of signals received or transmitted by individual emitters $S_{i}(t)\exp(-j\varphi_{i})$. It can be written analytically as:

$$\hat{S}(t) = \sum_{i=1}^{n} S_{i}(t)\exp(-j(\omega t + \varphi_{i})), $$

(13)
where \( \varphi_i \) is phase offset at \( i \)-th radiator. The limitation in the implementation of phased arrays is the complexity of the analytical description of non-standard radiation patterns, adaptable to real conditions. For example, the formation of zeros in the direction of interference at the maximum signal-to-noise ratio. That is the solution of difficult formalized problems. Analysis (13) shows that phased arrays can be implemented in a neural network basis. Since the main operations in (13) are the sum and the product, their implementation in the system of residual classes will become even more efficient [16].

3. Modeling a discrete matched signal filter in a neural network basis

Existing discrete matched signal filters (Figure 4) have several disadvantages. This is low noise immunity due to the lack of consideration of the influence of constant signal distortions \( S_k \) caused by external additive and multiplicative noise, distortions caused by technical features of the implementation of delay lines, especially on the capacitive element base.

Figure 4. Filter matched to the signal

Another disadvantage is their rigid implementation and the inability to adapt to real signal distortions, which leads to a decrease in the correlation response and, therefore, to a decrease in noise immunity. In addition, discrete matched filters significantly lose their immunity when irreducible errors occur in the signal, since their multipliers \( h_i \) reflect the impulse characteristics of the expected signal and are fixed, and in case of distortion of the received signal, the response of such filters is significantly reduced, reducing their noise immunity.

The elimination of these shortcomings is possible only by adding neural network blocks to a discrete matched filter. Note that the structure of the matched filter has a parallel view, and the delay line is due to the need to receive a serial signal. The matched filter solves the problem of calculating the cross-correlation function \( R(k) \) of the received signal \( S = \{x_i\} \) and the impulse response \( H = \{h_i\} \) of the expected signal depending on the shift \( k \) according to expression (10). The maximum \( \max[R(k)] \) is determined at the time the signal is detected. Obviously, distortion of the received signal \( S = \{x_i\} \) can significantly reduce the correlation response \( R(k) \). In fig. 4 elements of the input discrete signal \( S_k = \{x_1, x_2, \ldots, x_n\} \) are delayed in the delay line, pass through the attenuators \( h_i \) or multipliers and, folding in the adder (\( \Sigma \)), give a response at the filter output. The value of the attenuators is determined by the impulse response of the signal. Therefore, the response of the autocorrelation function is formed at the filter output. Exceeding a certain threshold value by this response means signal detection. The Solver, which is set to the response threshold, decides to detect the signal. If you pay attention to the structure of the matched filter (Fig. 4), you can notice that it corresponds to the scheme of a trained neuron (Fig. 1) that implements function (10). In fact, you can replace the multipliers with a self-learning neural network. This will lead to increased noise immunity when working in communication channels with interference and significantly reduce the effect of internal interference. The neural network has pattern recognition properties, smoothing distortion, which allows you to compensate for various types of interference and distortion and increase noise immunity.
Figure 5 shows one of the options for such an implementation, where a neural network is introduced at the impulse response level. At the same time, attenuators acting as an impulse response \( h_i \) can be taken into account when choosing the activation functions \( F(u_j) \) of a neural network. The role of the output adder and the deciding device, respectively, performs a common output neuron. A delay line is necessary to convert a serial signal in parallel. Obviously, if you use a parallel data format in the system of residual classes, then the delay line can be excluded.

For a sign-positive digital signal, we select the sigmoid function as the activation function (Figure 2). The solver, based on the Neumann-Pearson criterion, makes decision on the reception of the signal. This criterion takes into account the probability of a false alarm and can be taken into account in the output activation function. The value of the coupling weights is calculated according to the training algorithm of the neural network according to the Habib rule, for which the training set is the input signal. The learning algorithm, its learning speed and the activation function of the calculation unit can be selected depending on the specific types of signals, signal transmission speed and other tasks. The implementation of a neural network with training allows, due to its pattern recognition properties, to eliminate significant distortions of the input signal, eliminate fatal errors, eliminate the influence of internal interference of reception circuits and other interference, which gives the discrete matched filter adaptation properties. Finally, a mathematical model of an adaptive matched filter can be obtained by substituting expressions (2) and (4) into expression (10) for a conventional matched filter. Thus, we finally obtain:

\[
R(k) = \sum_{i=1}^{n} \left( \frac{1}{1 + \exp \left[ \sum_{j=1}^{m} x_{j+i} w_{i,j} \right]} \right) h_i. \tag{14}
\]

4. Conclusion

The resulting expression (14) can be directly used in modeling, as well as in the synthesis of adaptive matched filtering schemes. The introduction of mass parallelism in telecommunications becomes feasible with the integration of neural networks into telecommunication elements. Efficiency becomes maximum when applying simultaneously parallel mathematics of the system of residual classes.

References

[1] Schmidhuber J 2015 Deep Learning in neural networks: An overview Neural Networks 61 85–117

[2] Takefuji Y 1992 A Neural Network Parallel Algorithm for Channel Assignment Problems in Cellular Radio Networks IEEE Trans. Veh. Technol. 41 430–7
[3] Funabiki N, Takefuji Y and Lee K C 1992 A neural network model for traffic controls in multistage interconnection networks *IJCNN - International Joint Conference on Neural Networks* (Publ by IEEE)

[4] Smirnov A A, Bondar V V, Belokon L V, Saieg T kh., Mirzoyan M V and Smirnova O N 2018 The synthesis of the computing systems elements in neural network basis *Mod. High Technol.* 98–102

[5] Karras D A and Zorkadis V 2003 On neural network techniques in the secure management of communication systems through improving and quality assessing pseudorandom stream generators *Neural Networks* vol 16 (Elsevier Ltd) pp 899–905

[6] Smirnov A A, Bondar V V, Rozhenko O D, Mirzoyan M V. and Derzhaniya A D 2019 Mersenne numbers in the foundations of the system of residual classes when transmitting data in serial communication channels *The results of science and technology.* pp 87–94

[7] Aazhang B, Paris B P and Orsak G C 1992 Neural Networks for Multiuser Detection in Code-Division Multiple-Access Communications *IEEE Trans. Commun.* 40 1212–22

[8] Hiramatsu A 1989 ATM communications network control by neural network *International 1989 Joint Conference on Neural Networks* (Publ by IEEE) pp 259–66

[9] Tavanaei A, Ghodrati M, Kheradpisheh S R, Masquelier T and Maida A 2019 Deep learning in spiking neural networks *Neural Networks* 111 47–63

[10] Herzog S, Tetzlaff C and Wörgötter F 2020 Evolving artificial neural networks with feedback *Neural Networks* 123 153–62

[11] Smirnov A A and Sakhnyuk P A 2018 Neural network implementation of an optimal receiver in binary communication channels *Des. Technol. Electron. tools* 3–9

[12] He G, Tang P and Pang X 1996 Neural network approaches to implementation of optimum multiuser detectors in code-division multiple-access channels *Int. J. Electron.* 80 425–31

[13] Smirnov A A, Sakhnyuk P A, Sayeg T K, Vasilchenko A S and Yaroshuk A A 2019 Evaluation of noise immunity of communication channels with fading during neural network integration *Des. Technol. Electron. tools* 3–9

[14] Smirnov A A, Sakhnyuk P A and Senik V V 2004 Synthesis of digital signal processing circuits based on training of non-network circuits *Neurocomputers, Dev. Appl.* 91–4

[15] Brown T X 1989 Neural network for switching *IEEE Commun. Mag.* 72–81