

## NEURAL NET DECODERS FOR LINEAR BLOCK CODES

*V.N. Dumachev<sup>1</sup>, A.N. Kopylov<sup>1</sup>, V.V. Butov<sup>1</sup>*

<sup>1</sup>Voronezh Institute of the Ministry of Internal Affairs of Russia, Voronezh, Russian Federation,  
E-mails: dumv@comch.ru, k-a-n-777@yandex.ru, butovvladislav@yandex.ru

The work is devoted to neural network decoders of linear block codes. Analytical methods for calculating synaptic weights based on a generator and parity-check matrices are considered. It is shown that to build a neural net decoder based on a parity-check matrix was sufficiently four layers feedforward neural net. The activation functions and weight matrices for each layer are determined, as well as the number of weights for the neural net decoder. An example of error correction with uses of the BCH neural net decoder is considered. As a special case of a neural network decoder built on the basis of a parity-check matrix, a model for decoding Hamming codes has been proposed. This is the two-layer feedforward neural net for with a neuron number equal to the length of the codeword and a number of weight coefficients equal to the square of the codeword length. The graphs of the number of a synaptic weight of neural net decoders based on the generator and parity-check matrices, on the number of bits and the number of corrected errors, are shown.

*Keywords: error-correction codes; neural network decoders; neural network classification.*

## Introduction

Recently, artificial intelligence is increasingly associated with neural networks. Neural networks allow solving a wide range of classical problems for which alternative approaches do not give satisfactory results. However, the specifics of neural networks construction, the complex selection of optimal parameters show that many classical problems with simple decision algorithms, become cumbersome for neural network implementation. One example of such problems is the construction of neural net decoders for error-correction codes  $(n, k)$ .

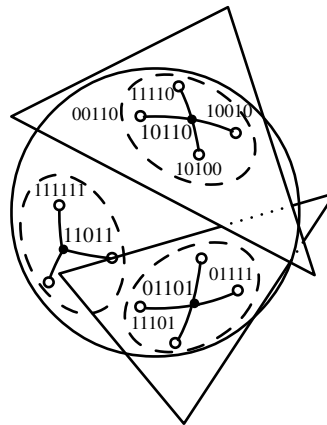
The idea of using neural networks for decoding error-correction codes was offered back in the 80's. For example, in [1] a two-layer recurrent neural network model of the Hamming decoder is proposed. The disadvantages of this model include the exponential growth of neurons with an increase of  $k$ , as well as the delays associated with the features of the recurrent layer this network. In [2], an approach related to the use of a check matrix is proposed for decoding Hamming codes. The first two layers of the neural network were needed to determine the error vector, the third and fourth to restore message. Decoding of linear block codes was also considered in [3–5]. In [3, 5] two-layer neural networks of direct propagation with  $2^k$  neurons in the hidden layer are proposed. The obvious drawback of these models is exponential growth of neurons with increasing  $k$ . In [4], a fundamentally different approach to the construction of neural net decoders is considered. The authors expand the feature space and the weight coefficients are calculated using genetic algorithms. The disadvantages of this approach are determined by the shortcomings of genetic algorithms. In [6–8] neural net decoders BCH(63,36), BCH (63,45), BCH (127,106) are considered. In this case, for decoding was used deep neural networks.

In this paper we use an analytic approach for constructing neural net decoders of matrix codes. In Section 1 we consider neural net decoders which are built on the base of

the generator matrix. In Sections 2 and 3 we propose models of decoders constructed on the base of the parity-check matrix.

### 1. Neural Net Decoder Built on the Basis of the Generator Matrix

If an error occurs, the valid codeword transforms to corrupt. Each word can be represented by a point in a Hamming space. For binary words the Hamming space is a sphere. In this case each original codeword, together with its own distorted words, can be grouped into clusters. Since all cluster points are located on the Hamming sphere, they belong to the same spherical segment (Fig. 1), i.e. can be separated from each other by cutting planes. The number of such separating planes for the code  $(n, k)$  is equal to  $2^k$ .



**Fig. 1.** The separating planes that cut off the Hamming sphere into clusters with centers corresponding to corrected codewords

Consider the neural network implementation of the described approach to decoding of error-correction codes. Let  $\mathbf{G}$  is a generator matrix,  $\mathbf{a} = (a_1, a_2, \dots, a_k)^T$  is an information word, then the corresponding codeword will be have the form:  $\mathbf{m} = \mathbf{G}\mathbf{a}$ . It is obvious that from the set of all  $2^k$  information words  $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{2^k})$  we can create  $2^k$  cluster centers  $\mathbf{M} = \mathbf{G}\mathbf{A}$  (allowed codewords). The decoder model is shown in Fig.2. Quantity neurons in the first layer is determined by the number of allowed codewords and in the second layer by the number of information bits. This model can be implemented as a perceptron or as a radial-basis functions network [3, 5].

In the first case, if we send to the input bipolar values  $\tilde{x}_i$  of received code combination  $\mathbf{x}$ , i.e.  $\tilde{x}_i = 2x_i - 1$  ( $\tilde{x}_i \in \{-1, 1\}$ ), then matrices  $\mathbf{W}^1$  and  $\mathbf{W}^2$  can be defined as  $\mathbf{W}^1 = 2\mathbf{M} - \mathbf{J}$ ,  $\mathbf{W}^2 = \mathbf{A}^T$ , where  $\mathbf{J}$  is all-ones matrix. As the activation functions of the first layer neurons one can take  $f^1(z) = \theta(z - z_{\max})$ , where  $\theta(z)$  is the Heaviside function with condition  $\theta(0) = 1$ ,  $z_{\max} = \max(\mathbf{W}^1 \tilde{\mathbf{x}})$ . For  $f^2(z)$  it suffices to take a linear function or ReLU. The second case was considered in detail in [5].

Note that to configure the neural network we need to calculate  $\mathbf{W}^1[n \times 2^k]$  and  $\mathbf{W}^2[2^k \times k]$ . In this case, we need to determine  $N = (n + k) \cdot 2^k$  synaptic weight.

### 2. Neural Net Decoder Based on a Parity-Check Matrix

In this section, we will consider an algorithm of analytic adjustment of a 4-layer neural net decoder for binary error-correction code  $(n, k)$ , which correcting up to  $t$  errors (Fig. 3).

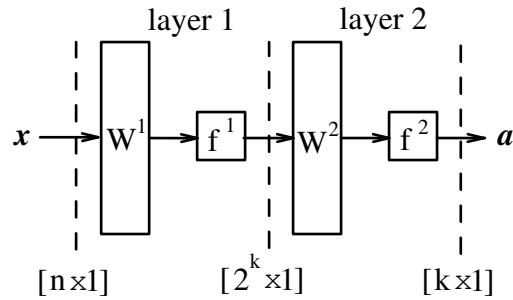


Fig. 2. Neural net decoder built on the basis of the generator matrix

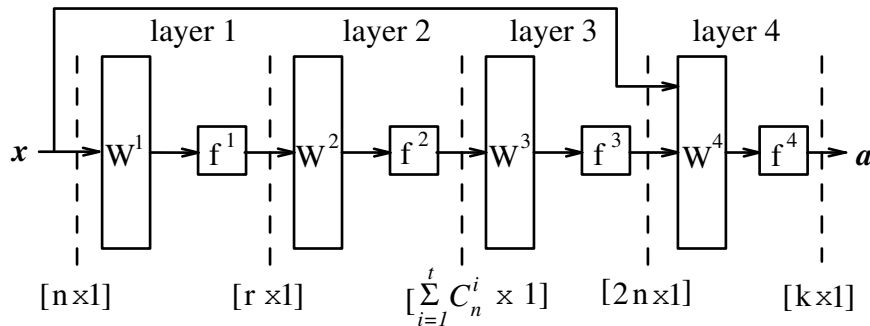


Fig. 3. Neural net decoder based on a parity-check matrix

Let  $\mathbf{H}$  is a parity-check matrix of the code, then for first layer we put  $\mathbf{W}^1 = \mathbf{H}$ ,  $f^1(z) = 2(z \bmod 2) - 1$ . The output of the first layer will give us a binary representation of the syndrome, which determines the error. To localize the error we use the syndrome matrix, which we denote by  $\mathbf{W}^2$ . The output of the second layer will be the vector that determines the position of the syndrome in matrix  $\mathbf{W}^2$ . Moreover,  $\mathbf{W}^2$  can be obtained using the mapping  $\mathbf{W}^2 = 2(\mathbf{H}\mathbf{W}^3)^T - \mathbf{J}$ , where  $\mathbf{W}^3$  is a matrix consisting of all possible error vectors that can be corrected. As activation functions for neurons of the second and third layers we take  $f^2(z) = \theta(z - r)$  and  $f^3(z) = z$ . In the fourth layer, the original code sequence and the error vector are added module 2 and the parity symbols are rejected. The activation function is  $f^4(z) = z \bmod 2$ . Note that to configure this neural network, we need to calculate  $\mathbf{W}^1 [r \times n]$ ,  $\mathbf{W}^2 \left[ \sum_{i=1}^t C_n^i \times r \right]$ ,  $\mathbf{W}^3 \left[ n \times \sum_{i=1}^t C_n^i \right]$  и  $\mathbf{W}^4 [k \times 2n]$ . In this case, we need to determine

$$N = (r + n) \cdot \sum_{i=1}^t C_n^i + n \cdot (r + 2k)$$

synaptic weight.

**Example 1.** Consider the BCH(31,21) code for information sequence  $\mathbf{a} = (1100100\dots 0_{21})^T$ . For matrix  $\mathbf{G} = (\mathbf{I}, \mathbf{G}'')^T$ , which generated by polynomial  $p(x) = 1 + x^3 + x^5 + x^6 + x^8 + x^9 + x^{10}$  we put in correspondence the parity-check matrix  $\mathbf{H} = (\mathbf{G}'', \mathbf{I})$ . Assume during transmission in the codeword  $\mathbf{m} = (1100100\dots 0101110101_{31})^T$  an errors occurred in the third and fourth symbols, i.e.  $\mathbf{x} = (1111100\dots 0101110101_{31})^T$ .

At the output of the first layer we have a vector-syndrome

$$\mathbf{y}^1 = f^1(\mathbf{W}^1 \mathbf{x}) = (1, 1, -1, -1, 1, -1, 1, 1, 1, 1)^T.$$

At the output of the second layer we have vector

$$\mathbf{y}^2 = f^2(\mathbf{W}^2 \mathbf{y}^1) = (000...001_{91}00...000_{496})^T,$$

which determines the position of the error vector in the syndrome matrix  $\mathbf{W}^2$ . We describe the algorithm for constructing this matrix. Define  $\mathbf{W}^3$  as a matrix whose columns are all possible errors that can be corrected:

$$\mathbf{W}^3 = \begin{pmatrix} 1 & & & & & & & & & & \dots & 1 & 1 & 1 & 1 & 1 & \dots & & & & & & & \dots \\ & 1 & & & & & & & & & \dots & & 1 & & & & \dots & & & & & & & \dots \\ & & 1 & & & & & & & & \dots & & & 1 & & & \dots & 1 & 1 & 1 & & & & \dots \\ & & & 1 & & & & & & & \dots & & & & 1 & & \dots & & & & & & & \dots \\ & & & & 1 & & & & & & \dots & & & & & 1 & & & & & & & & \dots \\ & & & & & 1 & & & & & \dots & & & & & & & & & & & & & \dots \\ & \dots \\ & \dots \end{pmatrix}.$$

As it was shown earlier  $\mathbf{W}^2 = 2(\mathbf{H}\mathbf{W}^3)^T - \mathbf{J}$ .

Since in vector  $\mathbf{W}^2 \mathbf{y}^1$  the maximum element is at 91 place, which corresponds to  $\mathbf{y}^2 = (000...001_{91}00...000_{496})^T$ , then we need to take the 91 column from matrix  $\mathbf{W}^3$ . It will coincide with the error vector for the received code combination:

$$\mathbf{y}^3 = \mathbf{W}^3 \mathbf{y}^2 = (001100...000_{31})^T.$$

At the output of the fourth layer we get valid information message

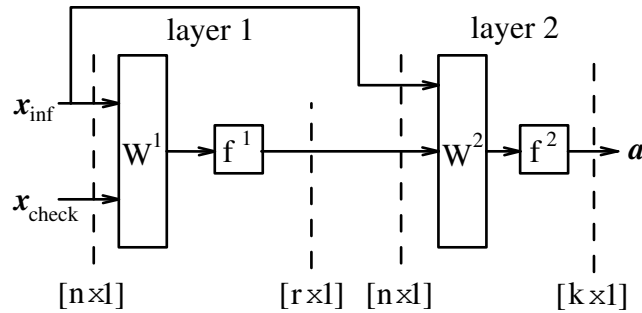
$$\mathbf{y}^4 = f^4 \left( \mathbf{W}^4 \cdot \begin{pmatrix} \mathbf{x} \\ \mathbf{y}^3 \end{pmatrix} \right) = (1100100...000_{21})^T = \mathbf{a}.$$

### 3. Neural Net Hamming Decoder on the Parity-Check Matrix Base

Note that in special cases, the number of layers in a neural net decoder can be reduced. As an example we consider the decoder of the Hamming codes. Let the input of a neural net decoder (Fig. 4) is vector  $\mathbf{x} = \begin{pmatrix} \mathbf{x}_{\text{inf}} \\ \mathbf{x}_{\text{check}} \end{pmatrix}$ , where  $\mathbf{x}_{\text{inf}}$  and  $\mathbf{x}_{\text{check}}$  are information and parity-check vectors respectively. Let  $\mathbf{H}$  is the parity-check matrix of the code written in a systematic form, then for the first layer we have  $\mathbf{W}^1 = \mathbf{H} = (\mathbf{G}'', \mathbf{I})$ . Function activation of neurons for this layer is given as follows:

$$f^1(z) = \frac{\varepsilon}{r} (2(z \bmod 2) - 1),$$

where  $\varepsilon$  is some positive number close to zero. The output of the first layer will give us a binary representation of the error position in the codeword (it is a syndrome), expanded to negative values and multiplied by  $\varepsilon/r$ . Next,  $\mathbf{x}_{\text{inf}}$  is combined with the syndrome and



**Fig. 4.** Two-layer neural net decoder of the Hamming codes, built on the basis of the parity-check matrix

transmitted to the second layer. The matrix of the weight for the second layer is equal to  $\mathbf{W}^2 = (\mathbf{I}, 2\mathbf{G}'' - \mathbf{J})$ . As a function of activation of neurons of the second layer, we take

$$f^2(z) = \begin{cases} 1 & \text{if } \varepsilon \leq z < 1 + \varepsilon, \\ 0 & \text{otherwise.} \end{cases}$$

Consider the restrictions on  $\varepsilon$  for correct work of the neural net. If we multiply matrix  $2\mathbf{G}'' - \mathbf{J}$  on the syndrome we will have a vector, the maximum of which is  $\varepsilon$ . This corresponds to an corrupted information bit. If corrupted bit is "0", then activation function  $f^2(0 + \varepsilon)$  equal unit. Otherwise, if the corrupted bit equal to "1", then  $f^2(1 + \varepsilon)$  will give "0".

For the correct operation of the network, it is necessary that the following condition is satisfied:  $1 - \varepsilon \geq \varepsilon$  (otherwise, correct bit "1" will be corrected to "0"). Thus  $\varepsilon$  must satisfy condition  $\varepsilon \leq 1/2$ . In particular, if  $\varepsilon = 1/2$ , then the activation functions of neurons of the first and second layers will take the form

$$f^1(z) = \frac{1}{r} \left( z \bmod 2 - \frac{1}{2} \right) \quad \text{and} \quad f^2(z) = \begin{cases} 1 & \text{if } 0,5 \leq z < 1,5, \\ 0 & \text{otherwise.} \end{cases}$$

If the half-interval in  $f^2(z)$  to shift on  $\frac{1}{2} \cdot \frac{\varepsilon}{r}$  to the left, then for  $\varepsilon = 1/2$  we will have

$$f^2(z) = \begin{cases} 1 & \text{if } 0,5 - \frac{1}{4r} \leq z < 1,5 - \frac{1}{4r}, \\ 0 & \text{otherwise.} \end{cases}$$

Note that to configure this neural network we need to calculate  $\mathbf{W}^1[r \times n]$  and  $\mathbf{W}^2[k \times n]$ . In this case we need to determine  $N = (r + k) \cdot n = n^2$  synaptic weight.

**Example 2.** Consider the Hamming (31,26) code for information sequence  $a = (1100100\dots 0_{26})^T$ . Generator matrix  $\mathbf{G} = (\mathbf{I}, \mathbf{G}'')^T$  is generated by polynomial  $p(x) = 1 + x^2 + x^5$ . Assume during transmission in codeword  $\mathbf{m} = (1100100\dots 01100_{31})^T$  an error occurred in the third symbols, i.e.  $\mathbf{x} = (1110100\dots 01100_{31})^T$ . At the output of the first layer we have vector

$$\mathbf{y}^1 = f^1(\mathbf{W}^1 \mathbf{x}) = 0,1 \cdot (-1, 1, 1, -1, -1)^T.$$

The input of second layer is vector

$$\begin{pmatrix} \mathbf{x}_{\text{inf}} \\ \mathbf{y}^1 \end{pmatrix} = (1, 1, 1, 0, 1, 0, 0, \dots, 0, 0,1, -0,1, 0,1, 0,1, -0,1)_{31}^T.$$

The output of this layer has the form

$$\mathbf{y}^2 = f^2 \left( \mathbf{W}^2 \cdot \begin{pmatrix} \mathbf{x}_{\text{inf}} \\ \mathbf{y}^1 \end{pmatrix} \right) = (1100100\dots0_{26})^T = \mathbf{a}.$$

Note that the weight matrices  $\mathbf{W}^1$  and  $\mathbf{W}^2$  include the unit matrices of orders  $r$  and  $k$ , respectively. It follows that to adjust the neural network is enough to determine  $N = 2rk$  synaptic weight.

## Conclusion

The paper considers three models of neural net decoders for linear block codes. Matrices of weight coefficients are analytically determined for each model. Note that when choosing a model for decoding linear block codes, two factors should be considered: the number of layers of the neural network and the number of synaptic connections. Fig. 5 shows typical graphs of number of synaptic connections for neural net decoders constructed on the base of the generator matrix (graphs 1, 2) and on parity-check matrix base (graphs 3, 4), in dependence from the number of information bits and the number of corrected errors. Graph 5 corresponds to the neural net decoder for the Hamming codes, built on the parity-check matrix base.

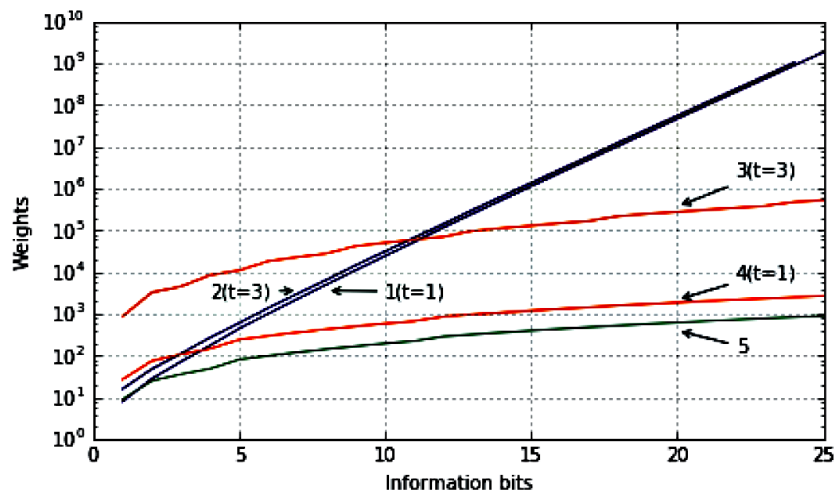


Fig. 5. Dependence of the number of synaptic connections on the code parameters

## References

1. Zeng G., Hush D., Ahmed N. An Application of Neural Net in Decoding Error-Correcting Codes. *IEEE International Symposium on Circuits and Systems*, 1989, vol. 2, pp. 782–785. DOI: 10.1109/ISCAS.1989.100467
2. Htay M.M. *A Computational Framework for Efficient Error Correcting Codes Using an Artificial Neural Network Paradigm*. PhD Dissertation. Louisiana State University and Agricultural and Mechanical College, 1992.
3. Ortuó I., Ortuó M., Delgado J. Error Correcting Neural Networks for Channels with Gaussian Noise. *IJCNN International Joint Conference on Neural Networks*, Baltimore, 1992, vol. 4, pp. 295–300.

4. Ja-Ling Wu, Yuen-Hsien Tseng, Yuh-Ming Huang. Neural Network Decoders for Linear Block Codes. *International Journal of Computational Engineering Science*, 2002, vol. 3, no. 3, pp. 235–255. DOI: 10.1142/S1465876302000629
5. Berezkin A.A. [Construction of Optimal Neural Decoders Block Codes]. *St. Petersburg Polytechnic University Journal*, 2008, no. 5, pp. 34–41. (in Russian)
6. Nachmani E., Beéry Y., Burshtein D. Learning to Decode Linear Codes Using Deep Learning. *54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, Monticello, 2016, pp. 341–346. DOI: 10.1109/ALLERTON.2016.7852251
7. Nachmani E., Marciano E., Burshtein D., Beéry Y. *RNN Decoding of Linear Block Codes*, 2017. Available at: arXiv.1702.07560.
8. Lugosch L., Gross W.J. Neural Offset Min-Sum Decoding. *IEEE International Symposium on Information Theory (ISIT)*, Aachen, 2017, pp. 1361–1365. DOI: 10.1109/ISIT.2017.8006751

Received July 12, 2018

УДК 004.032.26

DOI: 10.14529/mmp190111

## НЕЙРОСЕТЕВЫЕ ДЕКОДЕРЫ ЛИНЕЙНЫХ БЛОЧНЫХ КОДОВ

*В.Н. Думачев<sup>1</sup>, А.Н. Копылов<sup>1</sup>, В.В. Бутов<sup>1</sup>*

<sup>1</sup>Воронежский институт МВД России, г. Воронеж, Российская Федерация

Работа посвящена нейросетевым декодерам линейных блочных кодов. Рассмотрены аналитические методы расчета синаптических весов, базирующиеся на использовании порождающей и проверочной матриц. Показано, что для построения нейросетевого декодера на основе проверочной матрицы достаточно четырехслойной нейронной сети прямого распространения. Определены функции активации и весовые матрицы для каждого из слоев, а также количество весовых коэффициентов нейросетевого декодера. Рассмотрен пример исправления ошибок приведенным декодером при использовании кода БЧХ. В качестве частного случая нейросетевого декодера, построенного на основе проверочной матрицы, предложена модель для декодирования кодов Хэмминга. Данная модель представляет собой двухслойную нейронную сеть прямого распространения с числом нейронов, равным длине кодового слова, и числом весовых коэффициентов, равным квадрату длины кодового слова. Приведены графики зависимостей количества синаптических связей нейросетевых декодеров, построенных на основе порождающей и проверочной матриц, от числа информационных бит и числа исправляемых ошибок.

*Ключевые слова:* помехоустойчивое кодирование; нейросетевые декодеры; нейросетевая классификация.

## Литература

1. Zeng, G. An Application of Neural Net in Decoding Error-Correcting Codes / G. Zeng, D. Hush, N. Ahmed // *IEEE International Symposium on Circuits and Systems*. – 1989. – V. 2. – P. 782–785.
2. Htay, M.M. A Computational Framework for Efficient Error Correcting Codes Using an Artificial Neural Network Paradigm. PhD Dissertation / M.M. Htay. – Baton Rouge: Louisiana State University; Agricultural and Mechanical College, 1992. – URL: [https://digitalcommons.lsu.edu/gradschool\\_disstheses/5455](https://digitalcommons.lsu.edu/gradschool_disstheses/5455).

3. Ortuó, I. Error Correcting Neural Networks for Channels with Gaussian Noise / I. Ortuó, M. Ortuó, J. Delgado // IJCNN International Joint Conference on Neural Networks. V. 4. – Baltimore, 1992. – P. 295–300.
4. Ja-Ling Wu. Neural Network Decoders for Linear Block Codes / Ja-Ling Wu, Yuen-Hsien Tseng, Yuh-Ming Huang // International Journal of Computational Engineering Science. – 2002. – V. 3, № 3. – P. 235–255.
5. Березкин, А.А. Построение оптимальных нейронных декодеров блоковых кодов / А.А. Березкин // Научно-технические ведомости СПбГПУ. – 2008. – № 5. – С. 34–41.
6. Nachmani, E. Learning to Decode Linear Codes Using Deep Learning / E. Nachmani, Y. Beéry, D. Burshtein // 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton). – Monticello, 2016. – P. 341–346.
7. Nachmani, E. RNN Decoding of Linear Block Codes / E. Nachmani, E. Marciano, D. Burshtein, Y. Beéry. – 2017. – URL: arXiv.1702.07560.
8. Lugosch, L. Neural Offset Min-Sum Decoding / L. Lugosch, W.J. Gross // IEEE International Symposium on Information Theory (ISIT). – Aachen, 2017. – P. 1361–1365.

Владислав Николаевич Думачев, кандидат физико-математических наук, доцент, кафедра «Математика и моделирование систем», Воронежский институт МВД России (г. Воронеж, Российская Федерация), dumv@comch.ru.

Алексей Николаевич Копылов, кандидат технических наук, доцент, кафедра «Математика и моделирование систем», Воронежский институт МВД России (г. Воронеж, Российская Федерация), k-a-n-777@yandex.ru.

Владислав Вячеславович Бутов, адъюнкт, кафедра «Математика и моделирование систем», Воронежский институт МВД России (г. Воронеж, Российская Федерация), butovvladislav@yandex.ru.

*Поступила в редакцию 12 июля 2018 г.*