Evaluation of storage and functioning characteristics of artificial neural networks on the basis of a neurocomputer device

To cite this article: V A Romanchuk 2020 IOP Conf. Ser.: Mater. Sci. Eng. 734 012009

View the article online for updates and enhancements.
Evaluation of storage and functioning characteristics of artificial neural networks on the basis of a neurocomputer device

V A Romanchuk
Department of computer science, computer engineering, and methods of teaching of computer science, Ryazan State University named for S. Yesenin, Ryazan, Russia

E-mail: v.romanchuk@365.rsu.edu.ru

Abstract. A solution to the problem of evaluating the temporal and quantitative characteristics of the storage and processing of artificial neural networks based on a neurocomputer device is proposed. The most popular and used topologies of artificial neural networks are considered, for which analytical relationships are presented that allow evaluating the training cycle of an artificial neural network, the amount of memory needed and the amount of transmitted data. The proposed relationships differ in that approaches and characteristics unique to neuroprocessor devices are offered for evaluation when implementing the calculations presented in the neural network logical basis.

1. Introduction
Neurocomputing is a scientific field dealing with the development of sixth generation computing systems, neurocomputers, which consist of a large number of parallel working simple computing elements (neurons). Elements are interconnected, forming a neural network. They perform uniform computing operations and do not require external control. A large number of parallel computing elements provide high performance [1].

Currently, neurocomputers are being developed in most industrialized countries (companies Module, Qualcomm, IBM, Toshiba, Human Brain Project, KnuEdge Inc., Analog Devices, Texas Instruments, Darwin, Google, NVidia, Fujitsu, Eyeriss, Intel). Neurocomputers make it possible to solve a number of intellectual problems with high efficiency. These are the tasks of pattern recognition, adaptive control, forecasting, diagnostics, etc. [2].

However, at present, there is a lack of a single mathematical apparatus for describing and analyzing the functioning of the whole multitude of neurocomputer devices.

The aim of the research is developing a mathematical apparatus for evaluating some temporal and quantitative characteristics of storage and functioning of common artificial neural networks in their implementation on the basis of neurocomputer devices of different architectures.

2. Materials and methods
Let $Z_{j}^{(i)}$ be some j-th class of neural network problems, which is an artificial neural network when implemented on a microprocessor platform. An artificial neural network can be defined as a tuple of parameters and characteristics [3]:

...
- multiple inputs of the artificial neural network $Net_X = \{net_{x_1},...,net_{x_i},...,net_{x_m}\}$, where each input $net_{x_i}$ is characterized by a type and range of possible values;
- multiple outputs of the artificial neural network $Net_Y = \{net_{y_1},...,net_{y_i},...,net_{y_n}\}$, where each output $net_{y_i}$ is characterized by a type and range of possible values;
- a lot of neurons $Net_N = \{n_1,...,n_i,...,n_e\}$, each of which must be emulated to solve some class of tasks $Z_{INS}^{(j)}$;
- a lot of weighting coefficients $Net_W = \{net_{w_1},...,net_{w_i},...,net_{w_{nw}}\}$, characterized by the type and range of possible values;
- topology of artificial neural network $Net_T = \{SLP, MLP, RBF, Hop, Ham, BAM, RMLP, Elm, ART, CPN,...\}$, including the number of layers $SI$ and the number of neurons in each of the layers;
- activation function of neurons $F$;
- method of setting weight coefficients;
- and other parameters.

The currently used criteria for evaluating systems, which is an artificial neural network based on a neurocomputer device, reflect two main methodological approaches. In the first case, the studied system is considered as an element of the supersystem, within the framework of which its functional purpose is realized. This approach involves the analysis of the functioning of the supersystem, the study of its functional relationships and systems. Efficiency criteria in this case are, as a rule, temporary indicators of user interaction with the system, for example, access time to resources, execution time, downtime, etc.

In the second approach, the parameters of the system under study are considered. Moreover, the characteristics of the system’s connections with the supersystem appear only in the form of limitations. An example of performance criteria of this kind can be characteristics such as speed, availability, load factor, etc. However, they have the same meaning and can serve as a measure of comparison only in systems having the same architecture, logic, internal language, etc. Their application is justified in evaluating a system with a basic architecture. With any variation of the system parameters, these criteria may lose their meaning.

By studying the complex structures, such as artificial neural networks based on a neurocomputer device, it is advisable to use the criteria selected on the basis of the first approach to assessing the effectiveness of the system.

For storing a trained artificial neural network, two tasks must be solved:
1. Keeping the architecture of an artificial neural network: keeping the architecture and keeping the weighting coefficients. The most important parameter is the amount of memory for storing weighting coefficients for various topologies of the artificial neural network.
2. Processing the artificial neural network, where the important parameters are the time characteristic (training cycle of the artificial neural network) and the quantitative characteristic of the artificial neural network (the amount of data transmitted).

3. Topology of the perceptron
A perceptron is a simple neural network whose weights and offsets can be adjusted to solve the task of classifying input vectors. The single-layer perceptron (SLP) consists of a single layer comprising $nc$ neurons [4, 5].

For this topology, we have the sets:
- multiple inputs $Net_X = \{net_{x_1},...,net_{x_i},...,net_{x_m}\}$;
- multiple outputs $Net_Y = \{net_{y_1},...,net_{y_i},...,net_{y_n}\}$;
- a lot of neurons $Net_N = \{n_1,...,n_i,...,n_e\}$. 

- a lot of weighting coefficients \( \text{Net}_W = \{ \text{net}_{n1}, \ldots, \text{net}_m, \ldots, \text{net}_{nm} \} \);

The training cycle of an artificial neural network is equal to:

\[
K_T(SLP) \leq \frac{x_n \cdot n_c}{H_{\text{CUPC}}} + \max_{i=1}^{L} (\frac{n_c \cdot K_{R_l}}{H_{\text{MAC}} \cdot H_{m}}), \forall l = 1, L,
\]

where \( H_{\text{MAC}} \) is the number of emulated neurons per one cycle of NSAIDs, \( H_{m} \) is the number of cycles of NSAIDs per second, \( K_{R_l} \) - neural processor core load factor.

The required amount of memory can be calculated as follows:

\[
K_D(SLP) = \log_2 (\max(\text{net}_w)) \cdot n_w = \log_2 (\max(\text{net}_w)) \cdot x_n \cdot n_c.
\]

The total amount of data transmitted is equal to:

\[
K_B(SLP) = \sum_{i=1}^{m} \log_2 (\max(\text{net}_w)) + \sum_{i=1}^{L} \sum_{j=1}^{n} \log_2 (\max(f_j(\sum_{i=1}^{n} w_i \cdot x_i + w_0))).
\]

4. Topology of the multilayer perceptron

Under the multilayer perceptron (MLP), two different types are understood: the multilayer perceptron according to Rosenblatt and the multilayer perceptron according to Rumelhart. The Rosenblatt multilayer perceptron contains more than one layer of neurons. The Rumelhart multilayer perceptron is a special case of the Rosenblatt multilayer perceptron, but with feedback. Let \( Sl \) be the number of perceptron layers.

The total number of neurons is equal to: \( n_c(MLP) = \sum_{i=1}^{Sl} n_{c_i} \), where \( n_{c_i} \) is the number of neurons in the \( i \) layer.

The training cycle of an artificial neural network is equal to:

\[
K_T(MLP) \leq \frac{x_n \cdot n_c + \sum_{i=1}^{Sl-1} n_{c_j} \cdot n_{c_{i+1}}}{H_{\text{CUPC}}} + Sl \cdot \max_{i=L}^{1} (\frac{n_{c_i} \cdot K_{R_l}}{H_{\text{MAC}} \cdot H_{m}}), \forall l = 1, L.
\]

The required amount of memory can be calculated as follows:

\[
K_D(MLP) = \log_2 (\max(\text{net}_w)) \cdot (x_n \cdot n_{c_1} + \sum_{i=1}^{Sl-1} n_{c_j} \cdot n_{c_{i+1}}).
\]

The total amount of data transmitted is equal to:

\[
K_B(MLP) = \sum_{i=1}^{m} \log_2 (\max(\text{net}_w)) + \sum_{i=1}^{L} \sum_{j=1}^{n} \log_2 (\max(f_j(\sum_{i=1}^{n} w_i \cdot x_i + w_0))).
\]

5. Topology of radial basis networks

In general, the Radial Basis Function Network (RBF Network) refers to a two-layer network without feedback, which contains a hidden layer of radially symmetric neurons. The RBF network learning rate is much higher than that of the perceptron, but there are a number of drawbacks, the main of which is the deterioration of the approximation accuracy. A network has good generalizing ability only for a limited class of approximable functions, for example, for clustering classes in a feature space.

The total number of neurons is equal to: \( n_c(RBF) = n_{c_1} + n_{c_2} \).

The total number of synapses is: \( n_w(RBF) = x_n \cdot n_{c_1} + n_{c_1} \cdot n_{c_2} \).
The training cycle of an artificial neural network is equal to:

\[ K_T(RBF) = \frac{xn \ast nc + nc_1 \ast nc_2}{H_{CUPC}} + 2 \ast \max_{l \in L} \left( \frac{nc_1 \ast K_{Bl}}{H_{MAC} \ast H_{ml}} \right), \forall l = 1,2. \]

The required amount of memory can be calculated as follows:

\[ K_o(RBF) = \log_2 \left( \max(\text{net}_{w1}) \right) \ast (xn \ast nc_1) + \log_2 \left( \max(\text{net}_{w2}) \right) \ast (nc_1 \ast nc_2), \]

where \( \text{net}_{w1} \) are the hidden layer neuron weights, \( \text{net}_{w2} \) are the output layer neuron weights. The total amount of data transmitted is equal to:

\[ K_B(RBF) = \sum_{i=1}^{m} \log_2(\max(\text{net}_{wi})) + \sum_{i=1}^{m} \sum_{j=1}^{m} \log_2(f_j(\sum_{i=1}^{m} w_{ij}x_i + w_0)). \]

6. The Hopfield topology

The Hopfield network has a recurrent artificial neural network architecture. A characteristic feature of the architecture of a recurrent network is the presence of blocks of dynamic delay and feedback, which allows such networks to process dynamic models.

The number of inputs, the number of neurons, the number of outputs are equal and depend on the number of input parameters:

\[ nc(HopN) = xn(HopN) = yn(HopN). \]

The total number of synapses is:

\[ nw(HopN) = xn \ast nc_1 = xn^2. \]

The values of the input, output sets and neurons can take values only "1" or "-1":

\[ \text{Net}_X = \left\{ \text{net}_{x1},... \text{net}_{xi},... ,\text{net}_{xn} \right\}; \text{net}_{xi} \in \{1,-1\}; i = 1, xn, \]

\[ \text{Net}_Y = \left\{ \text{net}_{y1},... \text{net}_{yi},... ,\text{net}_{yn} \right\}; \text{net}_{yi} \in \{1,-1\}; i = 1, xn, \]

\[ \text{Net}_N = \left\{ \text{net}_{n1},... \text{net}_{ni},... ,\text{net}_{nn} \right\}; \text{net}_{ni} \in \{1,-1\}; i = 1, xn. \]

The training cycle of an artificial neural network is equal to: \( K_T(HopN) = \frac{xn^2}{H_{CUPC}}. \)

The required amount of memory can be calculated as follows:

\[ K_o(HopN) = \log_2 \left( \max(\text{net}_{wi}) \right) \ast nw = \log_2 \left( \max(\text{net}_{wi}) \right) \ast xn^2. \]

The total amount of data transmitted is equal to: \( K_B(HopN) = xn + R \ast 2 \ast q - 1. \)

7. The Hamming topology

The network consists of two layers. The first and the second layers have the same number of neurons equal to the number of samples. Neurons of the first layer have \( xn \ast xn \) synapses connected to the network inputs (which form a dummy zero layer). Neurons of the second layer are interconnected by inhibitory (negative feedback) synaptic connections. The only synapse with positive feedback for each neuron is connected to its own axon.

The number of neurons for the Hamming network is: \( nc(Ham) = 3 \ast xn. \)

The total number of synapses is: \( nw(Ham) = 3 \ast xn^2. \)
The weighting factors can take values: \( w(Ham) \in (0, 1) \).

The number of neurons in each layer is the same: \( n_c = n_{c_2} = n_{c_3} = n \).

The training cycle of an artificial neural network is equal to: \( K_T = \frac{3 * x_n^2}{H_{CUPC}} \).

The required amount of memory can be calculated as follows:

\[
K_0 = \log_2 (\max(\text{net}_i)) * x_n^2 + \log_2 (\max(\text{net}_{i_2,i_3})) * 2 * x_n^2.
\]

The total amount of data transmitted is equal to:

\[
K_B = \sum_{i=1}^{x_n} \log_2 (\max(\text{net}_i)) + R \sum_{i=1}^{5} \sum_{j=1}^{x_n} \log_2 (\max(f_j (\sum_{i=1}^{n} w_i x_i + w_0))).
\]

8. The topology of adaptive resonance theory

Several types of neural networks based on adaptive resonance theory (ART) have been developed, in particular, the ART-1 network and the ART-2 network. ART-1 network is designed to work with binary input images or vectors, and ART-2 network is designed to classify both binary and multidimensional vectors. Although the details of the architecture and algorithms for the ART-1 network and the ART-2 network are different, however, they share a common basic architecture. The structure of any ART network contains a single layer of neurons. The number of input variables is equal to the number of binary or real features characterizing the object. The number of outputs is not constant, there are no outputs at all when the network starts functioning. Gradually, their number increases with each new unfamiliar input image, forming, in fact, a new cluster.

For ART network: \( |Net_x| = |Net_y| \cdot n_c = x_n \).

The total number of synapses is: \( w = 2 * x_n * n_c \).

The training cycle of an artificial neural network is equal to:

\[
K_T (ART) = \frac{2 * x_n^2 * n_c}{H_{CUPC}} + \max_{l=1}^{L} (\frac{n_c * K_{RL}}{H_{MAC} * H_{m}}), \forall l = 1, L.
\]

The required amount of memory can be calculated as follows:

\[
K_0 (ART) = \log_2 (\max(\text{net}_i)) * 2 * n_c * x_n.
\]

The total amount of data transmitted for the ART-1 network is equal to:

\[
K_B (ART1) = x_n + \sum_{j=1}^{n_c} \log_2 (\max(f_j (\sum_{i=1}^{n} w_i x_i + w_0))).
\]

The total amount of data transmitted for the ART-2 network is equal to:

\[
K_B (ART2) = \sum_{j=1}^{x_n} \log_2 (\max(\text{net}_i)) + \sum_{j=1}^{n_c} \log_2 (\max(f_j (\sum_{i=1}^{n} w_i x_i + w_0))).
\]

9. The topology of the convolutional neural network

Convolutional neural network (CNN) is a special architecture of artificial neural networks, proposed by Jan Lekun and aimed at effective image recognition, is part of deep learning technologies. The idea of convolutional neural networks lies in the alternation of convolutional layers and subdirectory layers (layers subsample). The structure of the network is unidirectional (without feedbacks), fundamentally multilayer. For training, standard methods are used, most often the method of error back propagation. The activation function of neurons (the transfer function) could be any, at the option of the researcher.
The total number of neurons is equal to: \( nc(CNN) = nc_s + nc_p + nc_{ps}, \) where \( nc_s \) is the number of neurons in the convolution layer, \( nc_p \) is the number of neurons in the pooling layer, \( nc_{ps} \) is the number of neurons in the fully connected layer.

If you do not consider the option of sharing parameters, then:
\[
n_{c_s} = ((Iw - F + 2P) * S + 1) * ((Ih - F + 2P) * S * 1) * Ic,
\]
where \( F \) is the size of the convolution layer filter, \( S \) is the step with which the filters are moved, \( P \) is the number of zeros on the border of the input image.

\[
n_{c_p} = nc_s / 2 .
\]

\[
n_{c_{ps}} = \sum_{i=1}^{s} nc_i , \text{ where } nc_i \text{ is the number of neurons in the } i \text{ layer.}
\]

The total number of synapses is:
\[
nw(CNN) = (F * F * Ic) * nc_s * 1.5 + xn * nc_1 + \sum_{i=1}^{s-1} nc_i * nc_{i+1} .
\]

The training cycle of an artificial neural network is equal to:
\[
K_T(CPN) \leq \frac{3 * xn * nc}{H_{CUPC}} + 3 * \max_{i=1}^{L} \left( \frac{nc_1 * K_{R1}}{H_{MAC} * H_m}, \frac{nc_p * K_{R2}}{H_{MAC} * H_m}, \frac{nc_{ps} * K_{R3}}{H_{MAC} * H_m} \right).
\]

The required amount of memory can be calculated as follows:
\[
K_O(CPN) = \log_2 (\max (net_{wi})) * 2 * (F * F * Ic) * nc_s * 1.5 + xn * nc_1 + \sum_{i=1}^{s-1} nc_i * nc_{i+1} .
\]

The total amount of data transmitted is equal to:
\[
K_p(CPN) = \sum_{i=1}^{n} \log_2 (\max (net_{yi})) + \sum_{j=1}^{nc_p} \log_2 (\max (f_j (\sum_{i=1}^{n} w_i x_i + w_0))) + \\
\sum_{j=1}^{nc_{ps}} \log_2 (\max (f_j (\sum_{i=1}^{n} w_i x_i + w_0))) + \sum_{j=1}^{nc_{ps}} \log_2 (\max (f_j (\sum_{i=1}^{n} w_i x_i + w_0))).
\]

10. Results and conclusion

In the course of the study were obtained the analytical formulas for estimation of quantitative and temporal parameters: the training time of an artificial neural network, the required memory and the amount of data transmitted for different, most common topologies of artificial neural networks: a single layer perceptron, a multilayer perceptron, a radial basis artificial neural network, the Hopfield network, the Hamming network, BAM network, the Jordan network, the Elman network, the ART network, the Grossberg star, the Kohonen network, the network of a counter propagation, convolutional network. The given analytical relations can be further refined for the cases of solving problems of emulation of artificial neural networks of various types and using specific architectural models of modern neuroprocessor devices.

Acknowledgements

The study was carried out with the financial support of the Government of the Ryazan Region and the Russian Federal Property Fund in the framework of the research project No. 18-47-623001r_mol_a.

References

[1] Pande S, Morgan F, Cawley S, Bruintjes T, Smit G, McGinley D, Carrillo S, Harkin J and McDaid L 2013 Modular neural tile architecture for compact embedded hardware spiking
neural network *Neural Process Lett.*, 38, 131–53

[2] Izeboudjen N, Larbes C and Farah A 2014 A new classification approach for neural networks hardware: from standards chips to embedded systems on chip *Artif. Intell. Rev.*, 41, 491–534

[3] Ruchkin V, Romanchuk V and Sulitsa R 2013 Clustering, restorability and designing of embedded computer system based on neuroprocessors *Proceedings of the 2nd mediterranean conference on embedded computing* (Montenegro: Budva) pp 58-62

[4] Ruchkin V, Fulin V, Kostrov B, Romanchuk V and Ruchkina E 2015 Parallelism in embedded microprocessor system based on clustering *Proceedings of the 4nd mediterranean conference on embedded computing* (Montenegro: Budva) pp 45-50

[5] Avetisyan A, Grushin D and Ryzhov A 2002 Systems of cluster management *Proceedings of the Institute for System Programming of Russian Academy of Sciences* pp 140-7

[6] Ivutin A and Larkin E 2014 Estimation of latency in embedded real-time systems *Proceedings of the 3rd mediterranean conference on embedded computing* (Montenegro: Budva) pp 236-9

[7] Diken E, Jordans R, Corvino R, Jozwiak L, Corporaal H and Chies F 2014 Construction and exploitation of VLIW ASIPs with heterogeneous vector-widths *Microprocessors and Microsystems*, 38, 947–59