Representation models for parallel processes in parallel, cloud and distributed computing systems based on neuroprocessors

V A Romanchuk

1 Ryazan State University named for S.Yessenin, 46, Svobody Ave., Ryazan, 390000, Russia
E-mail: v.a.romanchuk@365.rsu.edu.ru

Abstract. We consider models of logical organization of information processing systems based on neuroprocessors, and analyze their effectiveness. The following study is based on their known general properties and principles of operation. To solve these problems, we employ the set-theoretic approach, the theory of graphs and finite automata, the theory of planning parallel computational processes, and the theory of mathematical and system analysis. We propose analytical expressions to describe the efficiency of the functioning of neuroprocessor systems as functions of the execution time of subroutines, or of the number of neurons emulated when implementing neural networks without/with feedback.

1. Introduction
Currently, many companies actively develop neurocomputing associated with neurocomputers, i.e., analog or digital computers whose main operating unit (central processor) is based on a neural network and implements neural network algorithms similar to the performance of the human brain.

Many companies have recently started manufacture of neurochips: RC “Module”, Qualcomm, IBM, Toshiba, Human Brain Project, KnuEdge Inc., Analog Devices, Texas Instruments, and others [1]. In 2014, IBM Research demonstrated its TrueNorth chip comprising a million digital neurons and 256 million synapses that are parts of 4,096 synaptic nuclei. At the demonstration event, the chip analyzed a video stream identifying automobiles, cyclists and pedestrians at a speed that was 100 times faster and consumed 1000 less energy as compared to existing computing devices built on the basis of the von Neumann architecture. The neurochips created have limited performance, so the manufacturers most often used cluster multiprocessor systems based on neurochips. The effectiveness of neurocomputer computing increases dramatically when using multiprocessor systems due to their architectural features and homogeneity of neurobase operations [2].

Among the works on fata parallelism in GPUs and NPUs we can list the works of the following authors: L. Bottou, S. O. Cyprien Noel, C. R. F. Niu, B. Retcht, S. J. Wright, Y. Jia, Caffe, E. C. B. P. Stephen Boyd, Neal Parikh, J. Eckstein, K. L. Xiaqing Li, Guangyan Zhang, W. Zheng [3-5].

Nevertheless, one of the problems that prevents creation of efficient multiprocessor structures based on neuroprocessors is the fact that the task of organization of specialized multiprocessor systems based on neuroprocessors is complicated and labor-consuming and, since unlike classical processors, neuroprocessors and neuroprocessing systems are not used as bases for required models, approaches, algorithms and software. Thus, we define the relevance of the research goal: the task of developing models of parallel processes in computing systems of various structures for a specialized hardware base
neuroprocessors [6, 7].

2. Model of parallel processes in a neuroprocessor system of the pipeline structure

Let $TO^{(j)}_i$ be the execution time of the $RO^{(j)}_i; i=1,...,q$ subroutine, where $q$ is the number of neuroprocessing devices – modules (NPSM). We draw a block diagram of the pipeline neural processors system (NPS) (Figure 1) [7, 9].

![Figure 1. Pipeline NPS.](image)

The $T_{c}^{(j)}$ cycle of the conveyor structure is defined as the duration of the maximum processing time, i.e. $T_{c}^{(j)}(konv) = \max_{i \in L} TO^{(j)}_i = \max_{i \in L} V_{i}^{(j)} * H_m, \forall l = 1, q$, where $H_m \in H$ is a certain technical characteristic that determines the time for execution of one stroke of an NPSM, is the number of steps of the subprogram; $V_i$ - the number of cycles of the subprogram $RO^{(j)}_i$. The first result of data processing during the implementation of the $A^{(j)}$ algorithm will be obtained at the pipeline output after time $t^{(j)} = T_{c}^{(j)} * q$. This value is the program execution time for two types of ANNs (artificial neural networks), i.e. $K_{T_{T_i}}^{(j)}(konv) = K_{DNPS}^{(j)}(konv) * _{i \in L} TO^{(j)}_i * q = \max_{i \in L} V_{i}^{(j)} * H_m * q, \forall l = 1, q$. For ANNs without feedback, each subsequent processing result will be obtained at the output after $T_{c}^{(j)}$ time, and then the time for obtaining the subsequent processing results $K_{T_{T_i}}^{(j)}(konv)$ can be calculated as follows:

$$K_{T_{T_i}}^{(j)}(konv) = \max_{i \in L} TO^{(j)}_i * q + (i - 1) * \max_{i \in L} TO^{(j)}_i,$$

$$K_{DNPS}^{(j)}(konv) = \max_{i \in L} V_{i}^{(j)} * H_m * q + (i - 1) * \max_{i \in L} V_{i}^{(j)} * H_m, \forall l = 1, q,$$

where $N$ is the ordinal number of the required processing result.

For ANNs with feedback: $K_{T_{T_i}}^{(j)}(konv) * = i * \max_{i \in L} TO^{(j)}_i * q = i * \max_{i \in L} V_{i}^{(j)} * H_m * q, \forall l = 1, q$. Figure 2 shows the sequence of loading subroutines to a distributed NPS (DNPS), where NPC is a neuroprocessor cluster.

We calculate the amount of data transferred for a DNPS, determined by the sum of the data transferred in each program cycle: $K_B = \sum_{i=1}^{q} K_{Bi}$. $K_{Bi}$ is determined by the input data set, that is, the set of ANN inputs $Net_X = \{net_{a1},...,net_{ai},...,net_{an}\}$, where each $net_{ai}$ input is characterized by its type and range of possible values. Let $NetX_{R_{i}}$ be the digit capacity of the $i$th input of the $Net_X = \{net_{a1},...,net_{ai},...,net_{an}\}$ ANN, determined by its maximum possible value: $NetX_{R_{i}} = \log_2 (\max (net_{ai}))$. 


Then, \( K_{BI} \) equals the total bit width of all input data: \( K_{BI} = \sum_{i=1}^{\infty} N_{Net} X_{K_i} = \sum_{i=1}^{\infty} \log_2(\max(net_{u_i})) \). For the cloud NPS (CNPS) in Figure 3, there are no delays during a conveyor run, but there are delays in the transmission of data from the \( U_i \in U \) user to the \( \Delta \) control node: \( T_s = T_{tU_i,0} + T_{0,U_i} \). Then the cycle of the CNPS conveyor is equal to the NPC conveyor cycle.

The first result of data processing in the implementation of an \( Z^{(j)} \) class of tasks will be obtained at the output of the pipeline after the \( T^{(j)}_c = T^{(j)}_{o} + T_s \) time. Then, taking into account the delay time, the \( PR^{(j)} \) execution time equals: \( K_{Tw}^{(j)}(o-\text{konv}) = (\max_{l\in L}TO_{l}^{(j)} + T_s)q, \forall l = 1, q \). Each subsequent processing result for an ANN without feedback will be obtained at the output after \( T^{(j)}_c \), and then we can calculate the time for obtaining the subsequent processing results as follows: \( K_{Tw}^{(j)} (o-\text{konv}) = (\max_{l\in L} TO_{l}^{(j)} + T_s)q + (N-1) * (\max_{l\in L} TO_{l}^{(j)} + T_s), \forall l = 1, q \). For an ANN with feedback: \( K_{Tw}^{(j)} (o-\text{konv}) = N * (\max_{l\in L} TO_{l}^{(j)} + T_s)q, \forall l = 1, q \).
3. Model of parallel processes in a vector neuroprocessor system

For a parallel system, the program execution time for a vector structure for two types of ANN is defined as the duration of the maximum processing stage, i.e.:

\[ K_{Tw}^{(j)}(\text{vect}) = \max_{i \in L} TO_{l}^{(j)} = \max_{i \in L} V_{l}^{(j)} \ast H_{m}, \forall l = 1, q. \]

For a vector-structure distributed complex, the data transmission delays can be calculated as follows:

\[ T_{s} = \sum_{i=1}^{q} T_{s_{i}}, \quad T_{s_{i}} = Tr_{i} + Tl_{i}, \]

\[ Tr_{i} = Tr_{0,i} + Tr_{r,0} \cdot Tl_{i} = Tl_{0,i} + Tl_{i,0}. \]

The \( PR^{(j)} \) program execution time in this case is the duration of the maximum processing stage and the overtime costs associated with data transfer:

\[ K_{Tw}^{(j)}(r - \text{vect}) = \max_{i \in L} TR_{l}^{(j)} + \max_{i \in L} T_{s_{i}}, \forall l = 1, q. \]

The \( PR^{(j)} \) program execution time for the cloud type is the duration of the maximum processing time stage and the overtime costs associated with data transfer:

\[ K_{Tw}^{(j)}(o - \text{vect}) = \max_{i \in L} TO_{l}^{(j)} + T_{s}, \forall l = 1, q. \]

4. Model of parallel processes in pipeline-vector neuroprocessor system

Let all adjacent subprograms where information is transmitted sequentially be combined into \( W \) sets:

\[ G_{i} = \{ RO_{1}^{(j)}, ..., RO_{W}^{(j)} \}, \forall i = 1, W. \]

The \( W \) number equals the number of contiguous NPSM united by vector parallelism. The number of subroutines in each group is different for each \( G_{i} \). The \( K_{Tw}^{(j)} \) execution time for the two ANN types can be defined as the time of the maximum processing time multiplied by the \( W \) number of sets:

\[ K_{Tw}^{(j)}(k - v) = K_{Tw}^{(j)}(k - v)* = W * T_{c}^{(j)} = W * \max_{i \in L} TO_{l}^{(j)} = \]

\[ = W * \max_{i \in L} V_{l}^{(j)} \ast H_{m}, \forall l = 1, q. \]

For a distributed conveyor-vector complex, the data transmission delays can be calculated in the same way as for a vector structure. The \( T_{c}^{(j)} \) cycle time equals the cycle time in the conveyor structure and is defined as the duration of the maximum time processing stage, taking into account the extra costs:

\[ T_{c}^{(j)}(r - k - v) = \max_{i \in L} TR_{l}^{(j)} + \max_{i \in L} T_{s_{i}}, \forall l = 1, q. \]

The execution time for the two ANN types can be defined as the maximum time of the processing stage, taking into account the time of data transmission costs, multiplied by the \( W \) number of sets:

\[ K_{Tw}^{(j)}(r - k - v) = W * T_{c}^{(j)} = W * (\max_{i \in L} TR_{l}^{(j)} + \max_{i \in L} T_{s_{i}}), \forall l = 1, q. \]

The \( T_{c}^{(j)} \) cycle time for the cloud type equals the cycle time in a conveyor structure and is defined as the duration of the maximum time of the processing stage, taking into account:

\[ T_{c}^{(j)}(o - k - v) = \max_{i \in L} TO_{l}^{(j)} + T_{s}, \forall l = 1, q. \]

The execution time for the two ANN types can be defined as the maximum time of the processing stage, taking into account the time of data transmission costs, multiplied by the \( W \) number of sets:

\[ K_{Tw}^{(j)}(o - k - v) = K * T_{c}^{(j)} = W * (\max_{i \in L} TO_{l}^{(j)} + T_{s}), \forall l = 1, q. \]

5. Model of parallel processes in a pipeline-vector neuroprocessor system

Let us suppose that all adjacent subroutines where information is transmitted sequentially are combined into \( W \) sets:

\[ G_{i} = \{ RO_{1}^{(j)}, ..., RO_{W}^{(j)} \}, \forall i = 1, W. \]

The number of subprograms in each group is different for each \( G_{i} \). The \( T_{c}^{(j)} \) cycle equals:

\[ T_{c}^{(j)}(v - k) = \max_{i \in L} TO_{l}^{(j)} = \max_{i \in L} V_{l}^{(j)} \ast H_{m}, \forall l = 1, q. \]

The \( K_{Tw}^{(j)} \) program execution time for the two ANN types is defined as the maximum duration of the processing stage multiplied by the maximum
size of the sets $G = \{ G_1, ..., G_V \}$, i.e.

$$K_{Tw}^{(j)}(v - k) = K_{Tw}^{(j)}(v - k)^* = \max_{i \in L} TO_i^{(j)} \ast \max_{G_i \in G} |G_i|^+ \ast \max_{i \in L} V_i^{(j)} \ast H_m \ast \max_{G_i \in G} |G_i|, i = 1, W, \forall l = \overline{1, q}.$$  

Then, the $K_{Tw}^{(j)}$ program execution time in a distributed system for the two ANN types is defined as the maximum duration of the time processing stage, taking into account the data transfer time costs, multiplied by the maximum value of the $G = \{ G_1, ..., G_V \}$ set size, i.e.

$$K_{Tw}^{(j)}(r - v - k) = (\max_{i \in L} T_{Tw}^{(j)} + \max_{T_S} T_{S_i}) \ast \max_{G_i \in G} |G_i|, i = 1, V, \forall l = \overline{1, q}.$$  

The runtime of a $T_{Tw}^{(j)}$ program in a cloud system for the two ANN types is defined as the maximum duration of the processing stage taking into account the data transfer time costs multiplied by the maximum value of the the set $G = \{ G_1, ..., G_V \}$ size, i.e.: 

$$K_{Tw}^{(j)}(o - v - k) = (\max_{i \in L} TO_i^{(j)} + T_S) \ast \max_{G_i \in G} |G_i|, i = 1, V, \forall l = \overline{1, q}.$$  

6. Conclusion
As a result of the study, we have considered options of possible multiprocessor structures based on neural processors, and the number of processor modules for each option is determined. It is shown that in order to implement a mechanism for choosing the optimum number of neuroprocessors and structures for the given class of tasks, we need criteria and methods for calculating the values of criteria. We propose analytical expressions describing the effectiveness of functional neuroprocessor systems, in particular, the execution time of the program presented on a neural network basis.

7. Acknowledgments
The research was carried out within the framework of the assignment for the performance of public works in the sphere of scientific activity within the framework of the initiative scientific project of the state task of the Ministry of Education and Science of the Russian Federation, No. 2.9519.2017 / BC, on the topic "Technologies for parallel processing of data in neurocomputer devices and systems".

References
[1] Barabash O, Kravchenko Y, Mukhin V, Kornaga Y and Leshchenko O 2017 International Journal of Intelligent Systems and Applications(IJISA) 9 1–9
[2] Kryuchkovsky V V and Usov A V 2013 Proceedings of Odessa National Polytechnic University 2 154-160
[3] Singh H, Lee MH, Lu G, Kurdahi FJ, Bagherzadeh N and Filho EMC 2000 IEEE Transactions on Computers 49(5) 465-481.
[4] Pande S., Morgan F., Cawley S., Bruintjes T., Smit G., McGinley D., Carrillo S., Harkin J. and McDaid L. 2013 Neural Process Lett. 38 131–153
[5] Bottou L 2012 Neural networks: Tricks of the trade 421–436
[6] Yakkalli R T and Raghava N S 2017 International Journal of Image, Graphics and Signal Processing(IJIGSP) 9 38–49
[7] 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) 45-50
[8] Romanchuk V A 2017 IOP Conference Series: Materials Science and Engineering 161 012033
[9] Goswami S, Chakraborty S, Saha H N 2017 International Journal of Intelligent Systems and Applications(IJISA) 9 20-30