CYBERNETICS and COMPUTER TECHNOLOGIES

UDC 519.85 DOI:10.34229/2707-451X.21.4.8

M. KOSOVETS, L. TOVSTENKO

DEVELOPMENT OF A CLUSTER WITH CLOUD COMPUTING BASED ON NEURAL NETWORKS WITH DEEP LEARNING FOR MODELING MULTIDIMENSIONAL FIELDS

The process of modeling multidimensional electromagnetic fields, which are described by a system of differential equations, has been studied. A cluster with synchronous links between clusters and asynchronous links inside with access to cloud computing is used as instrumentation. The cluster is made in the form of a multiprocessor based on neural network technology with deep learning. Biomimetic principles are used in the architecture of the modeling complex. Thus, the central system module plays the role of a reticular system, which interacts with almost all computing structures. It is assumed that the functional metastable structures of the phase space of the neural network are model representations of a multidimensional system.

***Keywords**: cognitive space, deep learning, convolutional neural network, neural network architectures.*

Introduction. We live in an unstable electromagnetic environment, which is described by a system of equations of mathematical physics in partial derivatives. The information loading of the signal spectrum is growing rapidly. The common resource is provided by stations that can be taught to be passive monitoring stations, active LPI location stations, base telecommunication stations and use in military, security, environmental monitoring, medical and others [1]. User access to the resource is simplified. The intellectual component of real-time signal processing is complicated, which requires the use of neural networks with deep learning. Particular attention is paid to the use of bionic principles in the processing of multidimensional signals. A cluster computing with cloud computing is proposed to create a modeling complex for processing multidimensional signals and debugging the target system.

We are not currently considering sensors for capturing multidimensional information, this is a separate topic. The multidimensionality of the system speaks of the multiplicity of qualitative characteristics of the field: power, signal shape, coordinates of the measuring point, time characteristics and others. The systems are very complex to design and therefore expensive. Before making real systems, the field and received signals are simulated. Mathematical models generate fields with initial and graphical conditions. In the future, we replace them with real, recorded on media. To simplify processing, the process itself is broken down into computing units. This allows you to sort out the application of processing algorithms using cloud computing. The use of artificial intelligence elements to process multidimensional signals is as natural as the processes they describe. A neurocomputer for processing multidimensional signals is a multiprocessor with a heterogeneous network of processors. Moreover, the set of processor modules has both a different architecture and a system of connections between them. The idea of creating a neurocomputer for processing multidimensional signals is not new, but the introduction of a cluster system with cloud computing for modeling multidimensional gives new possibilities for constructing field processing systems. Traditionally, these systems are primarily used in military and security: target detection at long distances, tracking airborne objects, including stealth-based hypersonic aircraft and ballistic missiles, drones; automatic detection of parameters of movement of air targets (range, azimuth, altitude, speed).

1. Bionic aspects of building the architecture of a neurocomputer with deep learning for modeling multi-dimensional fields

Humanity has always used bionic principles when building computer systems. Nature has developed these principles for millions of years, so copying it is appropriate. But copying should be reasonable, taking into account the technology of production of components of computer technology, the development of mathematical, algorithmic, software for signal processing and the need for such computer systems. Now this direction in science is called biomimetics. Practical use of the principles of the brain was previously impossible due to the lack of sufficient research and technical support to implement it. Today, an intermediate stage is used: the joint use of old and new principles of constructing calculators. Thus, a biological neuron has dendrites for receiving input signals, while artificial neurons have received several input channels. A biological neuron has a cell body that processes its input and an artificial neuron has a function that maps input to output. Likewise, a biological neuron has an output axon through which multiple axon terminals transmit output to different channels, while an artificial neuron has an output layer to communicate with multiple neurons. The difference between the human brain and deep learning (DL) lies in the ability of the brain to optimize global cost functions in layered networks by adjusting or assigning each neuron its own contribution to the global outcome.

In all aspects of activity, optimization determines biological states – these are locomotion, respiration, nervous activity and even genetic evolution. There are many functions that are easily customizable and shaped by evolution and the needs of the body. In contrast, DL assumes that there is a global optimization function on which the network is focusing. So, the theory, formulates self-organization and learning without a teacher, which can directly eliminate the need for multi-level credit appropriation [2]. Variations of Hebben with nonlinearities can lead to self-organized optimization of cost functions [3].

We have outlined how deep learning differs from the architecture of the human brain. Let's look at the differences from a learning perspective. An important assumption underlying most connectionist models such as deep learning is: what can be learned from statistical patterns in the input data. There are many different forms of connectionism, but the most common are neural network models. Neural networks are flexible in approximating functions and require large amounts of data [4]. Bayesian Programming (BPL) introduces the concept where simple stochastic programs – structured procedures – generate new execution examples [5]. The BPL allows learning like “model building” by building generative models, and has been shown to work well with one-time training. The three basic tenets that are required for the success of a BPL are compositionality, causality, and learning for learning.

So far, there is no targeted work on introducing compositionality into deep learning of the network, with the exception of work – this provides a unique way to provide inductive bias using the model structure, which leads to significant performance in understanding the scene. Another famous work in this area, which proposes a neural modular network that is composed structurally and dynamically using representative copying and routing. We believe that Capsule Networks uses a layering concept, similar to the visual system of the human brain, to display a transformation matrix and represent 3D translation, scale and rotation. Capsule networks capture spatial relationships between objects in an image using information about the scene. The increase in publications on different models of the functioning of neural fences is explained by the increased interest in the problems of modeling artificial intelligence, which display images of dynamic and statistical information flows. The results of the introduction of chaotic neurons into the network structures are discussed.

The neurodynamic concept provides the possibility of building modes of localization, synchronization, stabilization of metastable chaotic structures in the reticular environments of the neural system. Complex biological systems cannot be accurately described due to the continuous chaotic change in the parameters of such systems in the phase space of states. Such dynamics is characteristic of neural brain processes that form model representations of mental images. In the reticular neural environment, there are processes similar to those observed in dissipative dynamical systems. These processes are accompanied by the emergence of simple structures that correspond to their own discrete values of the nonlinear problem. The mode of organization of complex structures is established, which has a metastable character in the phase space of a nonlinear problem. In the reticular neural environment, this mode corresponds to the creation of metastable structures of the ensemble of neurons in the phase space.

Consider in more detail the response of the reticular environment with neurons in the nodes to the input signal, which comes in the form of perturbations of the neural environment. These disturbances can be not only external but also purely internal. The result of their influence is the activation of neurons in response to perturbations. Activated neurons create a mosaic picture of simple structures. Depending on the parameters of the perturbation signal, the fundamental lengths of the created simple structures may intersect, in which case complex structures of simple ones are created in the system. Some of them are characterized by metastability in the phase space of the dynamic model in the system of independent variables as functions of time.

The process of relaxation triggers the reflection in the cognitive space of the created metastable structures – model representations of images. Model representation of an image is a metastable structure in the phase space of states. Therefore, we can assume that cognitive space is a phase space of metastable states – model representations of images, which are displayed in the form of functional coded "units" – functional modes. At any time, functional modes can be activated by the information flow. It can be argued that model representations of mental images in the form of functional modes constitute a static diversity of phase space as a subset of the cognitive space of memory. As a result, the phase space of a dynamic task can be replenished with metastable memory structures. Metastable phase states are being restructured, model representations of mental images are being reformed, taking into account information that was previously "stored". Cognitive computing only partially replicates the work of the brain in structuring multidimensional information and processing, using the principles of deep machine learning, developing new rules and algorithms for working with real-time data in computers with non-von Neumann architecture.

Artificial neural networks (ANN) are a highly connected network of elementary processors running in parallel. Each elementary processor calculates one output based on the information it receives. The two main elements make up ANN: the neuron model used to build the network, and then the network architecture. Each artificial neuron is an elementary processor that receives several neural inputs. Each of these inputs has an associated weight that represents the strength of the connections between the corresponding neurons. This puts forward two specific characteristics of each neuron: a "potential" equal to the sum of the input weights, and an "activation function" that outputs the neuron according to its "potential".

In the structure of feedback neural networks, the only suitable connections are between the outputs of each level and the input of the next level. A neural network with a backpropagation algorithm is one of the known methods of creating a trained machine or system that can provide the final solution for classification through a number of learning processes. It was developed using the Neural Networks tool provided by MATLAB. The backpropagation algorithm is an extension of the perception of multilayer neural networks. Thus, the backpropagation algorithm uses three or more levels of processing (neurons).

2. The concept of developing the cluster architecture with cloud computing for modeling multidimensional fields

There is no infrastructure for building multidimensional information calculations. We offer the construction of a modeling complex in the form of a cluster with cloud computing and debugging functions. Points that attract but do not cross the trajectory of a dynamic system are called attractors. Analysis at an early stage allows you to predict the behavior of the studied system. Such points that make up the trajectory are, in fact, a mathematical image of a complex motion, called strange attractors. The strangeness of the attractor is that, unlike a conventional attractor, which characterizes the stability of a dynamic system, all trajectories around it are dynamically unstable, and this instability is manifested in the mixing of trajectories in phase space. It can analyze not only linear but also nonlinear dynamic systems. An example of an attractor is the propagation of energy from the antenna to the target and vice versa. Power and propagation path are constantly changing, but the flow of energy is generally stable in space and does not go beyond certain limits.

Clusters with cloud computing to model multidimensional fields use synchronous structures, within which the processes are asynchronous. Each block of calculations is separated by steps, output registers, switched by a synchronizer. This is very convenient for the point (pixel) processing of radio fields and the reduction in the future into a block of global operations. Most of the multidimensional fields today boil down to 3D radio imaging, although this is not required. A synchronous technique is used in timing optimization problems to determine critical time windows for resources with asynchronous latency. From here, an estimate of the typical latency of each configuration can be obtained. In addition, a special technique for allocating resources for specific operations has been developed, which uses information about the allocation of time, together with information obtained from the data flow graph. Using both sources of information, the distribution problem is very often solved by a heuristic algorithm.

It is difficult to investigate all possible configurations for the implementation of this scheme, because the number of possible solutions grows exponentially with the size of the data flow graph. A number of precise and heuristic filters have been implemented to reduce the size of the schema. These filters are very effective in reducing circuit learning time and include: reducing circuit size when implicit edges are detected, removing redundant circuits from consideration, efficiently finding a solution with minimal latency, and detecting when maximum configuration is reached without examining the entire branch of design space. Implementing circuit design is convenient through a computer-aided design (CAD) tool for experimenting with various automatic timing methods, extrapolating the design parameter space, and the effect of the proposed filter. It can take an abstract model of the data flow graph, and from it form both the optimal structural representation of the information channel for this scheme, and the necessary behavioral control for coordinating this information channel. The generated structural view consists of a block diagram of related functional blocks, latches, controls, and multiplexers.

The biomimetic principle is applied: for the visual systems, the brain is organized in the form of deep architecture, and perception is presented in the form of several levels of abstraction. DL architectures are characterized by artificial neural networks, usually involving more than two layers. Deep neural networks use data-driven representations of functions. However, they do not require hand-crafted functions, which are mostly developed based on knowledge of a specific subject area. It is also impractical to consider the need to account for all the details embedded in all forms of real data, hand-generated functions. Rather than relying on small, hand-designed functions, DL methods can automatically learn informative representations of raw input with multiple levels of abstraction. These learned functions have been successful through their use in many machine vision applications. Deep learning is widely represented in scientific literature: deep belief networks are a generative graphical model, or, one of the types of deep neural networks, consisting of several hidden layers, in which neurons within one layer are not connected to each other, but are connected to neurons in an adjacent layer through the use of Restricted Boltzmann Machines (RBM).

We emphasize two important aspects: one concerns the classification by points (pixels) for multidimensional signals, and the other concerns the processing of a full-dimensional signal (for example, a frame) with high resolution. The first is related to determining which category each pixel in a given frame of the radio image belongs to, and the second is aimed at automatically assigning a semantic label to each frame.

$$map_{l,j}^{x,y} = f\left(\sum_{m}\sum_{h=0}^{H_l-1}\sum_{w=0}^{W_l-1} k_{l,j,m}^{h,w} map_{(l-1),m}^{(x+h),(y+w)} + b_{l,j}\right),$$

where $(h,\ w)$ is the value the position of the core connected to the $(l-1)$-th layer. H_i and W_i are the height and width of the core respectively, and $b_{l,j}$ is the offset of the j-th layer on map of objects in the l-st layer. Such convolutional layers introduce a weight sharing mechanism within the same function maps, which helps to greatly reduce the number of parameters otherwise required. For example, Lin and Chen proposed a network within a network, replacing the conventional convolutional layer with a multilayer perceptron composed of several fully connected layers. A fully connected layer is basically the same as a traditional neural network (backpropagation network).

The loss function or energy function measures the reconstruction z at a given input x,

$$J(\theta) = \frac{1}{2M}\sum_{m=1}^{M}\left\| z^{(m)} - x^{(m)} \right\|_2^2,$$

where M denotes the number of training samples. The goal is to find parameters $\theta = (W,\ by,\ bz)$ that can minimize the difference between output and input over the entire training set $X = [x(1),\ x(2)\ \ldots,\ x(m),\ \ldots,\ X(M)]$, and this can be efficiently implemented using a stochastic gradient descent algorithm (Johnson & Zhang, 2013). These approaches can be divided into three main categories, which will use spectral information, spatial information, and spectral-spatial information, respectively.

DL models have the ability to automatically extract features from raw input, and such functions are high-level and abstract functions. The DL pretrained network can be used as a function extractor for any type of multidimensional signal, since the functions learned by the network are less dependent on the end application and can be used for a variety of tasks [6]. Alternatively, we can use the pretrained CNN as a local feature extractor and combine it with feature coding to generate the final scene representation. After introducing, from two main points of view, the classification of images by pixels and by radio images by scenes, we will supplement the consideration of DL for signal processing according to spectral, spatial characteristics and joint spectral and spatial characteristics, using both controlled and uncontrolled methods of feature extraction.

To effectively detect details in the panoramic sensor, multiple candidate points are created around the radar origin. It then uses a widely used deep learning approach to detect and localize the left and right corners of target vehicles [7].

3. Principles of implementation of high-performance built-in m-processor cluster

Processing of multidimensional signals requires large computational resources. Taking into account the functional orientation of computations to perform a variety of different types of operations, we use neural networks built on the basis of streaming computations implemented on signal processors and FPGAs. The required computational resources are obtained by flexible restructuring of the multiprocessor resources [8]. The proposed model consists of a main and backup communication channel and *m*-multi-microprocessor clusters (fig. 1).

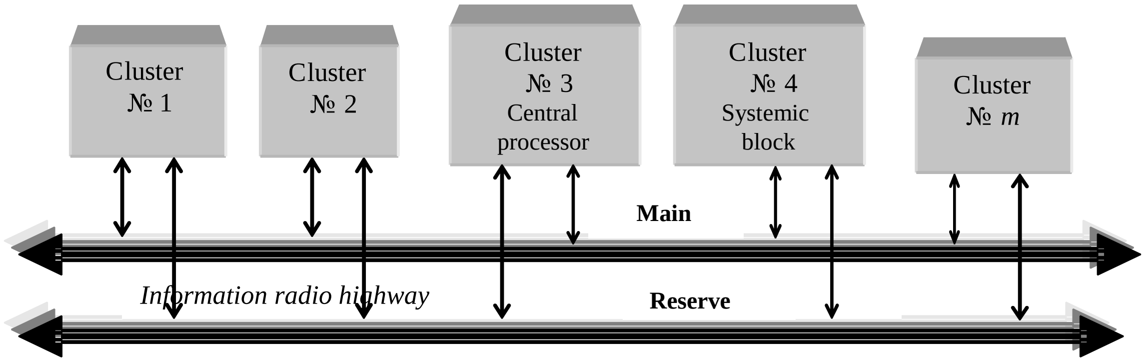

Fig. 1. The architecture of an *m*-cluster fault-tolerant real-time multiprocessor

In networked systems consisting of many processing elements, the exchange with each other is carried out through the transmission of messages. The tasks of processing multidimensional signals require very high performance, and, on the other hand, an efficient organization of a parallel computational process. Structurally, an *m*-processor cluster consists of m executive processors, system memory, control and communication processors. The first of them is the cluster administrator, records the state of the process and distributes tasks. The second one controls access to the communication medium, encoding / decoding and conflict resolution when implementing multiple access. All clusters are interconnected by means of a high-speed data exchange bus, multiple access to the communication medium. The system is easily reconfigurable taking into account the specifics of the task, the technical state of the multiprocessor system as a whole or individual clusters [9]. The dimension m of the cluster depends on the applied problem. The task is divided into subtasks and distributed among the processor modules. The technical implementation of the cluster system is shown in fig. 2, which consists of the System block, which is responsible for redistributing tasks within the system and generating a terahertz radar sounding signal; a central processor that simulates a deep learning neural network, and a communication cluster for communicating with the cloud and loading executive programs into the system. The multidimensional array of information comes from the receiving matrix of the terahertz radar.

Fig. 2. Cluster with cloud computing for modeling multidimensional fields

In a multiprocessor, each of the clusters can equally effectively perform supervisory functions that "flow" from one cluster to another. However, any cluster can perform all or most of the system-wide functions. Any task in the process of execution can be processed in different clusters. System-wide control is continuously redistributed between clusters: at a time, only one cluster can be a control one. A symmetric processing multiprocessor is best built with a fully distributed operating system, i. e., with the storage of copies of the operating system in all processors. When clusters fail, there is a gradual decrease in performance (system degradation). Recovery is carried out by redistributing the task between clusters. For a multiprocessor with *m*-processor clusters, we use a distributed OS, which is a collection of cluster OSs that monitor the state of computing resources for the purpose of planning the computational process, maintaining reconfiguration, an apparatus for returning to checkpoints, etc. A multiprocessor can have various OSs of clusters. So, a service cluster can have an OS focused on exchange with external devices, and a functional cluster – an OS focused on the maximum computation speed and exchange within the cluster. Each functional cluster contains a multitasking OS kernel.

To simulate a multiprocessor, we propose to take as a basis the apparatus of Colored Petri Nets (CPN), which is most suitable both for simulating protocols in multiprocessor systems and for simulating parallel processes in such systems. Colors can be used to distinguish between different processes, even if their subnets are collapsed into a single network. The system model has significantly fewer states than the equivalent model based on conventional Petri nets. The problem of choosing an architecture lies in the optimal mapping of multidimensional signal processing tasks to the structure of the computing environment of a multiprocessor, since the quality of the display predetermines performance. The highest performance is possible if the structure of the problem and the topology of the system are adequate.

The signal processing cluster includes a central service module, executive processor modules, an administrator processor, system RAM, a high-speed computer network communication processor, and accelerator coprocessors. The exchange between multiprocessor clusters is carried out through the connection space. To transfer a message, the processor – the sender prepares the message in a local buffer and "negotiates" with the recipient about the allocation of buffer space for a copy of the message and, if the "agreement" is reached, transmits data over the bus in packets at the maximum speed [10]. Considering the asynchronous method of transmitting messages with fuzzy information processing, there is no arbitration on the bus, the main requirement is that the receiver's processor has free resources to process this message.

Conclusions. This article discusses a fault-tolerant real-time multiprocessor for solving problems of processing multidimensional signals. The features of the multiprocessor operation when processing large amounts of information are shown. The parameters of information flows were determined, the tasks were redistributed between specialized coprocessors-accelerators (neural networks) for signal processing and processor control elements, testing and diagnostics. The requirements for the system backbone are determined, the throughput of which has ceased to be a limiting factor limiting the performance of a computing system.

On the example of 3D Radar Imaging, a cluster of modeling, processing and debugging of multidimensional signals was studied. Calibration of the modeling complex was possible thanks to the use of a neurocomputer with deep learning and the possibility of using feedback with an antenna. The calibrations used a small metal plate and several measurement cycles to average out the noise. It is shown that the measurement accuracy is influenced by the width of the radiation pattern, a decrease in the number of measurement cycles at one point, the accuracy of positioning and movement of the antenna during measurements, and the time interval between calibration.

References

1. Ranneberger O., Fischer P., Brox T. U-net: Convolutional networks for biomedical image segmentation. Springer International Publishing. 2015. P. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28

2. Kappel D., Nessler B., Maass W. STDP installs in Winner-Take-All circuits an online approximation to hidden Markov model learning. *PLoS Computational Biology*. 2014. 10(3): e1003511. https://doi.org/10.1371/journal.pcbi.1003511
3. Masci J., Meier U., Dan C., Schmid H.J. Stacked convolutional autoencoders for hierarchical feature extraction. Paper presented at International Conference on Artificial Neural Networks, Espoo, Finland, 2011. P. 52–59. https://doi.org/10.1007/978-3-642-21735-7_7
4. Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A., Fei-Fei L. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Computer Vision*. 2015. 115. P. 211–252. https://doi.org/10.1007/s11263-015-0816-y
5. Lake B., Salakhutdinov M., Tenenbaum J. Human Level Concept Learning through Probabilistic Program Induction. *Science*. 2015. **350** (6266). P. 1332–1338. https://doi.org/10.1126/science.aab3050
6. Andreas J., Rohrbach M., Darrell T., Klein D. Learning to Compose Neural Networks for Question Answering. Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL). 2016. P. 1545–1554. https://doi.org/10.18653/v1/N16-1181
7. Kang D., Kum D. Camera and radar sensor fusion for robust vehicle localization via vehicle part localization. *IEEE Access*. 2020. 8. 75223–75236. https://doi.org/10.1109/ACCESS.2020.2985075
8. Palagin A. On computers with virtual architecture. *Upraliayushie sistemy i mashiny*. 1999. 3. P. 33–43. (in Russian)
9. Kosovets M.A., Kosovets L.N. On hardware for increasing the reliability of fault-tolerant microprocessor systems. *Cybernetics and Computer Engineering*. 1993. 99. P. 102–104. (in Russian).
10. Kosovets M. Features of the architecture and structure of micro-computers with a variable ratio of performance and reliability. *Sat: Special electronics*. 1990. Series 10, issue I (26). P. 23–28. (in Russian)

Received 17.12.2021

Mykola Kosovets,
Leading Constructor, Scientific-production enterprise “Quantor”, Kyiv, Ukraine,
https://orcid.org/0000-0001-8443-7805
quantor.nik@gmail.com

Lilia Tovstenko,
Leading Software Engineer, V.M. Glushkov Institute of Cybernetics of the NAS of Ukraine, Kyiv.
https://orcid.org/0000-0002-3348-6065
Scopus Author ID: 56439972800
115lili@incyb.kiev.ua

УДК 519.85

М. Косовець [1 *], Л. Товстенко [2]

Розробка кластеру з хмарними обчисленнями на базі нейромереж з глибоким навчанням для моделювання багатовимірних полів

[1] *Науково-виробниче підприємство «Квантор», Київ, Україна*
[2] *Інститут кібернетики імені В.М. Глушкова НАН України, Київ*
[*] *Листування: quantor.nik@gmail.com*

Вступ. Розглянуто моделювання багатомірних полів на мультипроцесорах, з нейромережевою архітектурою, яка перебудовується у процесі вирішення задачі шляхом глибокого навчання. Така архітектура обчислювача використовує пристрій для вирішення задач пасивної локації, моніторингової станції, LPI активної локаційної станції, базової телекомунікаційної станції одночасно. Особливу увагу приділено використанню біонічних принципів при роботі з багатовимірними сигналами. Пропонується кластерний обчислювач з хмарними обчисленнями для створення моделюючого комплексу обробки багатомірних сигналів і відлагодження цільової системи.

Кластер виконаний у вигляді мультипроцесора на основі технології нейронної мережі з глибоким навчанням. Кожен із кластерів може рівноцінно ефективно виконувати супервізорні функції, які "перетікають" з одного кластера в інший. Разом з тим будь-який кластер може виконувати всі або більшість загальносистемних функцій. Будь яка задача у процесі виконання може оброблятися в різних кластерах.

Мета роботи. Створення моделюючого комплексу як кластера з хмарними обчисленнями і використанням нейромереж з глибоким навчанням. Кластер являє собою нейромультипроцесор, який перебудовується в процесі роботи.

Результати. У процесі роботи нам вдалося створити моделюючий комплекс з архітектурою мультипроцесора, який у процесі обчислень перебудовується і використовує хмарні обчислення. Комплекс апробовано при моделюванні терагерцового сканера 3D Imager.

Висновки. У процесі виконання роботи створено комплекс для моделювання багатовимірних полів. За основу обчислювача використано мультипроцесор, який перебудовується у процесі роботи. Обчислювальна база мультипроцесора – нейромережі з хмарними обчисленнями.

Ключові слова: когнітивний простір, глибоке навчання, згортка нейронної мережі, архітектура нейронних мереж, кластер.