Integration of artificial neural networks and knowledge bases

Golovko Vladimir Adamovich
Kroshchanka Alexsander Alexsandrovich
Brest State Technical University
Brest, Belarus
gva@bstu.by
kroschenko@gmail.com

Golenkov Vladimir Vasilyevich
Ivashenko Valerian Petrovich
Kovalev Mikhail Vladimirovich
Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus
golen@bsuir.by
ivashenko@bsuir.by
michail.kovalev7@gmail.com

Taberko Valery Vasilyevich
Ivaniuk Dzmitry Sergeevich
JSC "Savushkin product"
Brest, Belarus
id@pda.savushkin.by

***Abstract*—This article reviews the questions and directions of integration of artificial neural networks with knowledge bases. There are two main directions of integration:**

1) the inputs and outputs of artificial neural network to use integration of knowledge bases and artificial neural networks for solutions of application problems;

2) by artificial neural network representation on the basis of ontological structures and its interpretation by means of knowledge processing in the knowledge base providing an intelligent environment for the development, training and integration of different artificial neural networks compatible with knowledge bases.

The knowledge bases that are integrated with artificial neural networks are built on the basis of homogeneous semantic networks and multiagent approach to represent and process knowledge.

***Keywords*—ANN, knowledge base, integration, frameworks**

INTRODUCTION

At the moment, the idea that various areas of artificial intelligence should not develop in isolation has great acknowledgement. Fortunately, using synergetic approach, we can rely on not only the solutions of particular problems, but also on the achievement of ambitious goals such as the development of strong artificial intelligence systems.

The popularity of problem solving methods based on machine learning is inspired by the development of contemporary theoretical models of artificial neural networks and high-performance hardware platforms for their implementation. The variety of architectures, methods, directions and applications of artificial neural networks has accumulated and continues to be updated.

The complexity of contemporary intelligent systems that use artificial neural network models, as well as big data processing, require means of monitoring, understanding and explanation of the mechanisms of their work to verbalize and optimize their activity.

Therefore, it becomes actual to develop integration approaches for artificial neural networks and knowledge bases based on ontologies. Such integrated systems are able to combine the possibility of semantic interpretation of data that processed by an artificial neural network (ANN) including input and output data specification and the representation of ANN solvable applied problems with structure representation and the description of characteristics and states of artificial neural network making easier an understanding of its operation.

This article will describe the following questions:

- Is an artificial neural network a kind of knowledge and what kind if yes?
- What classes of problems are convenient to solve with the help of artificial neural networks, and which ones are not?
  - What classes of problems are convenient to solve by integrating artificial neural networks and knowledge bases?
  - What are the benefits of embedding an artificial neural network and its input and output data into knowledge base?
- How artificial neural network interacts with knowledge base?
- How to present artificial neural network in knowledge base?
- What is the structure of the domain of artificial neural networks?
- What agents are needed to process artificial neural networks in the knowledge base?

I. ARTIFICIAL NEURAL NETWORKS AND KNOWLEDGE

The appearance of artificial neural networks is associated with the works of W. McCulloch and W. Pitts [2], in which, however, preferred the term *neurons net*. The term *artificial neural network* has been fixed since the works of K. Fukushima [3]. Especial interest is the use of deep artificial neural networks for the formation of abstractions. These networks have an important advantage that distinguishes its from superficial models – the ability to form a complex hierarchy of characteristics (Figure 1 [4]). For the first time, an effective algorithm for training deep artificial neural networks was proposed by J. Hinton in 2006 [5]. Since that moment, a lot of work has been devoted to training this type of networks (for example, [6], [7]). Also, fundamentally new approaches were proposed([8], [9]).

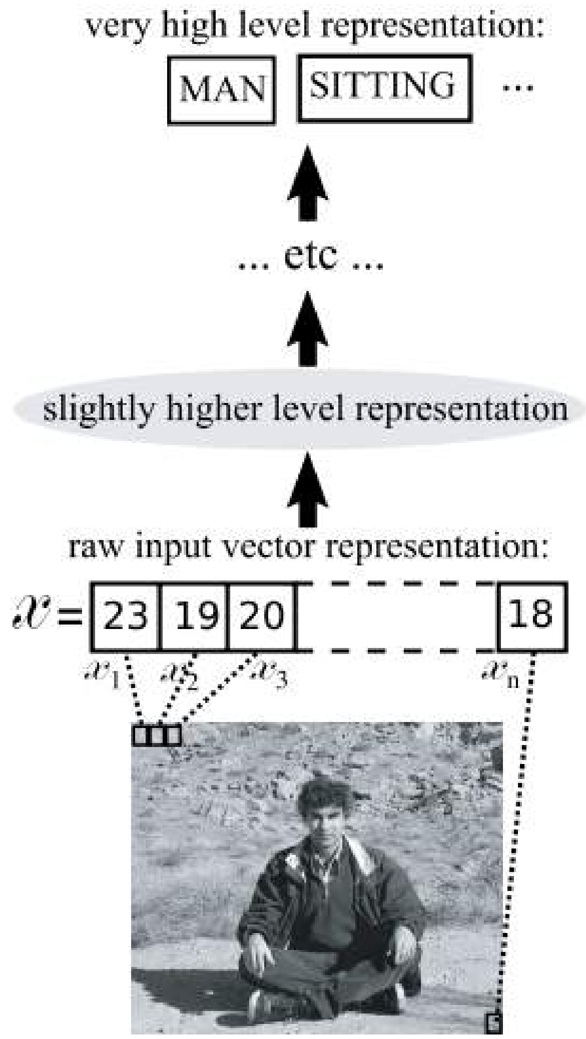

Figure 1. Hierarchy of features.

In this article, we consider artificial neural networks based on the concept of a formal neuron [10]. The formal neuron is specified by the composition of two functions:

1) function of synaptic transformation;
2) neuron activation function.

Artificial neural network that based on formal neurons is specified by four components:

1) set of vertices of artificial neural network (V).
2) set of edges of artificial neural network (E ($E \subseteq V \times V$)).
3) set of formal neurons, their properties and parameters of artificial neural network (N).
4) set of mapping between the union of vertex sets and edges of artificial neural network and the set of formal neurons, their properties and parameters of artificial neural network S

$$S \subseteq 2^{(V \cup E) \times N} \tag{1}$$

.

Artificial neural networks can be correlated with the concept of knowledge in accordance with different approaches to understanding knowledge. Correlation based on the [11] characteristics:

1) connectivity(artificial neural networks are based on a connected oriented graph, but not always strongly connected);
2) complex structure(there are artificial neural networks with a complex structure, consisting of layers set, vertices of artificial neural network and back connection);
3) interpretability(there are software implementations of artificial neural networks, where they are interpreted);
4) activity(software implementations of artificial neural networks have activity);
5) semantic metric (semantic metric in the general form and in the case when vertices of artificial neural networks is absent).

In accordance with IDEF-5 [12] knowledge is presented as relations, properties, kinds and attributes. In accordance with this, artificial neural networks have only numerical attributes and, possibly, sometimes kinds and relations, given with the help of predicates. However, in the general case, the predicates functions are not defined and are not known.

From the structural approach point of view, the artificial neural network is a mathematical structure(model) that corresponds to the principles of connectionism, which corresponds to the information process(becoming) that takes place in the technical(cybernetic) system expressed as a mapping or function (the dependence of the output data on the input).

From a pragmatic-mathematical point of view, the data processed by artificial neural network are elements of certain set, often interpreted as a sign set (image, attributes of some entity), and set itself as a feature(vector) space. There are binary, nominal, ordinal and quantitative features. Future is a numeric, quantitative attribute of an entity, an entity that can be expressed, specified by a mapping or function.

II. Classification of tasks that can be solved by artificial neural networks

Artificial neural networks are used to solve tasks. The task can be specified by the relation between the description of the initial situation and the description of a set of target situations.

Classify the tasks can be in accordance with different characteristics.

Since set of formal neurons are open, outline a class of tasks that are convenient to solve using artificial neural networks is problematic. Along with this, the accumulated experience shows that artificial neural networks are successfully used to solve tasks of the following types (see Table 1).

Table I
CLASSIFICATION OF ARTIFICIAL NEURAL NETWORK TASKS.

conceptually-pragmatic	**mathematical**
	approximation
processes management	- optimization
processes prediction	- extrapolation
images generation	
images matching detection of anomalous phenomena	
logical inference	logical inference
images translation	
- images clustering	- vector quantization
- images compression	- decrease in dimension
- linear division of images	- increase in dimension
- images association	
- images classification	
- images recognition	

Nevertheless, based on general methodological principles, it is possible to specify the signs of classes of tasks that are convenient to solve using artificial neural networks:

1) difficult-formalizable tasks, the solution of which has natural(vector) parallelism or data parallelism;

2) tasks, the solution of which is stable to the presence of NOT-factors in data and knowledge;

Also, there are list of possible signs of classes of tasks that are unprofitable to solve only with the help of artificial neural networks:

1) a complex conceptual description in task;

2) the presence of NOT-factors and the instability of solution to its;

3) the dominance of complexly described successive processes in the solution of the task;

4) need for introspective analysis and explanation of the obtained results in complex artificial neural networks.

Among the classes of tasks, it is possible to distinguish the classes of tasks that are conditionally specialized and the classes of tasks conditionally being abstract-fundamental [13].

III. USING OF INTEGRATION OF ARTIFICIAL NEURAL NETWORKS AND KNOWLEDGE BASES FOR SOLVING APPLIED TASKS

Intelligent systems with knowledge bases for solving applied problems with the help of artificial neural network algorithms can be used both for internal tasks, such as training neural network or optimizing its operation, and for solving a target problem, by processing input and output parameters of artificial neural network. Such intelligent systems can use artificial neural network methods on a par with other methods available in the system for solving only one, determined by system from subtasks during solving applied task.

Further, we will describe examples of use integration of artificial neural networks and knowledge bases for solving applied tasks.

A. Implementation of knowledge base integration into marking recognition system at JSC "Savushkin product"

One of the derived tasks at JSC "Savushkin product" is goods marking recognition task [14]. There is artificial neural networks [15], [16] based project to analyze image from camera, located above the production line (Fig. 2).

Figure 2. Production line with the camera unit

The unit recognize information from the bottles cover as string with the date of manufacture and the expiration date (Fig. 3).

Figure 3. Yogurt bottles with marked covers

This project is used by the engineer of Electrical Control & Instrumentation (EC&I engineer). The simplest way of integration into the bottle line system is to stop the line and inform about this the EC&I engineer using emergency sound and light alarm in case of detection n (e.g. 5) bottles with errors in recognition from the bottles cover. He should determine the cause of the problem and resolve it. In this case the module recognize and send the result as string to the knowledge base, where it should be processed according to this rules:

1) if the resulting string does not match the reference and it is only the one occurrence thus we should just discard this bottle (if there is a rejection device);

2) if the resulting string is empty and it is repeated for n occurrence consecutive bottles that is the ink has run out and we need to stop the line and notify the EC&I engineer to refill the printer;

3) if the resulting string is not empty but does not match the reference and it is repeated for n occurrence consecutive bottles that is nozzles are jammed and we need stop the line and notify the EC&I engineer to clean the printer unit.

Such rules can be a lot. In addition, we must be able to view and edit them (possibly even by EC&I engineer).

Since consumer packaging is constantly changing (and marking too), the next question arises: who and how will tune this system to a new kind of marking? Suppose today Arabic numerals are printed, and tomorrow Marketing department will decide to use Roman numerals. What needs to be done is to create a training set, train a new artificial neural network, launch the modified project. Such job can not be done by an EC&I engineer even if he wants (there are not enough knowledge in this area). But if there is some agent between them in the form of an intelligent system, then everything can look like this: the intelligent system receives information from artificial neural network that it is not possible to recognize the marking string, and from the user (engineer of the instrumentation and automation) information that everything in order and it was the launch a new type of labeling, then it can independently perform the tasks listed earlier to adapt the recognition module to this new type of labeling.

Proceeding from the above classification of tasks, this task can be classified as task of the detection of anomalous phenomena using the tasks of the images recognition. Also this task has signs of tasks described above, which are unprofitable to solve only with the help of artificial neural networks, because task has a dominance of difficult to describe successive processes in the solution. Using knowledge base integration allows to use artificial neural network algorithms to solve this task despite the presence of unprofitable factors for this.

B. Application of artificial neural network to support and update knowledge base

Today researchers proposed architectures, that consist from networks of different types. One of the ways of integration of artificial neural networks and knowledge bases lies in the use of such hybrid architectures, that form system for processing streams of knowledge to solve complex tasks.

An example of this structure is a hybrid network that uses convolutional neural networks (CNN) and recurrent LSTM networks (long short-term memory) [17].

This architecture (also known as LRCN – Long-term Recurrent Convolutional Network) is depicted in Fig. 4 [17]. Here are present in fact two variants such architecture, each of them allows to solve own task. First variant assumes averaging of results LSTM-network and forming compact description for depicted action (for example, HighJump). This variant can be consider as recognition of action, depicted on series of images. For second variant we need sequentially apply LSTM-network to form text description for image. Both variants uses two types of artificial neural networks. First type is convolutional neural network (CNN), which is used to process original image and form features. Second type is LSTM-network, which gets features and generates text description.

According to classification from section II, this is patterns recognition task (for first variant of model) and patterns generation task (for second variant of model).

The final text description can be analyzed, formulated into rules and integrated in the semantic network. Semantic network accumulates the knowledge in the form of specific objects and relations between them, and is able to perform a logical conclusion. In addition to image analysis, a similar architecture is used to generate text descriptions of the video stream.

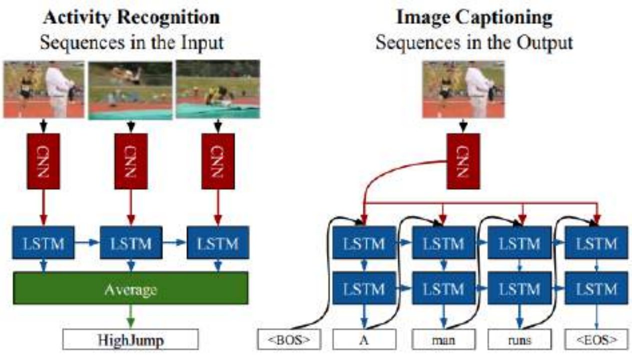

Figure 4. Long-term Recurrent Convolutional Network

Another use case is based on the word2vec approach [18]. This method uses for a semantic analysis of the text. The method is based on the use of simple shallow artificial neural network, which provides the formation of context based on one central word (skip-grams model – Fig. 5 [18]), or the formation of a central word based on the context (CBOW – Continuous Bag of Words model). The context is a set of words, which surrounding the central word and taken within a certain window. The trained network performs mapping of the data (words) from one-hot form to lower dimensional space, which then used to estimate the semantic similarity of words. The resulting embedding can be used for prediction semantic relations. For example, **king for queen** is also that **father for ?**.

Here is an example of the visualization obtained by us for a training set of 100000 English Wikipedia documents and vocabulary size of 50000 words. In this experiment used a simplified architecture of a skip-grams, including 50000 input neurons corresponding to the central word, 300 hidden and 50000 output neurons corresponding to the context word. Training set consist of pairs in form **(central_word, context_word)**, which fed to artificial neural network in mini-batches for 128 pairs in each. After training we applied t-SNE for dimensionality reduction of data. The resulting two dimensional map of semantic similarity depicted in Fig. 6.

Word2vec can be used to form knowledge base in certain subject domain (for example, [19]) and to extract semantic relations in common [20].

According to classification from section II, this is dimensionality reduction task.

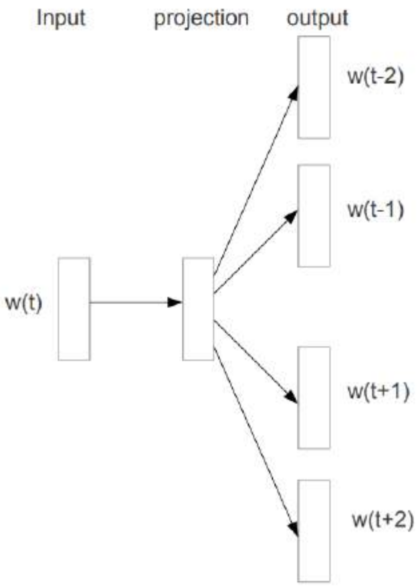

Figure 5. Skip-grams model

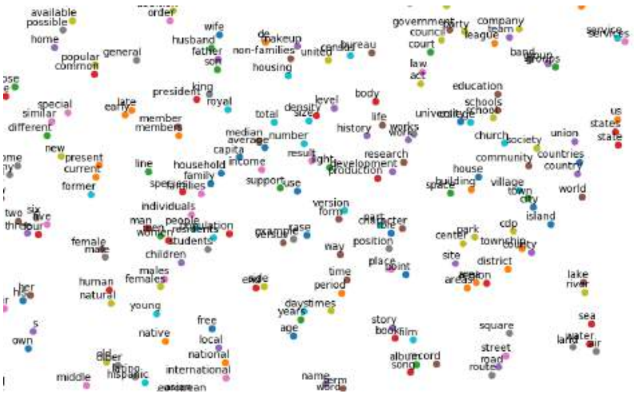

Figure 6. Map of semantic similarity

C. *Using knowledge base for artificial neural network training*

Interesting variation for integrating artificial neural networks and knowledge bases is to use knowledge base for artificial neural network training. With this approach, knowledge base comprise description of subject domain ontologies with in which task is solved, and with which you can describe artificial neural network inputs and outputs values. It also means the presence of knowledge-processing machine that can interpret the stored rules of **validation**(checking whether submerged value corresponds to certain conditions), checking for **non-consistency**(checking that new knowledge does not violate logical rules stored in knowledge base) and **adjustment**(replacing some knowledge with others) knowledge. This integration can be used to resolve the following tasks:

1) Validation and adjustment of training, test and examination samples. All elements of the samples are put to knowledge base and checked for consistency. Also in system can be stored a set of rules for validation and adjustment of features values of input elements.

2) Check consistency and correctness of artificial neural network output during training phase and implement appropriate adjustment of training. Training of artificial neural network is adjusted depending on intermediate results, which, with help of the reference to knowledge base, are checked for validity, consistency and plausibility. Also, result can be adjusted, for which knowledge base must have appropriate rules. Depending on the result of call to knowledge base, the weights calculation algorithm can be changed, which will allow training algorithms to be expanded by directly taking into account semantics.

3) Validation and adjustment of the results of trained artificial neural network. Artificial neural network outout can also be checked for validity, consistency and plausibility, and also adjusted with the help of rules used during training , and with the help of another specialized set of rules.

For resolving these tasks, knowledge base of a certain intelligent system can be used, as well as knowledge base, which has been specially developed for training particular artificial neural network.

Let us consider an example of classification task. In general, the statement of classification task is next: there is a group of objects. Each object has **n** features, it is necessary to assign each object to one of the **p** specified classes. Accordingly, artificial neural network is trained on some sample of objects with known features, and depending on type of training(with or without a teacher), with a known belonging to the class.

The training process can be improved by describing object classes and its features in knowledge base. Then, on training stage, for example, without a teacher, artificial neural network classified object to class **j**. Since knowledge base stores a description of this class, it will be possible to check consistency of object features with description of class that was detected. And already knowing this, it will be possible to do conclusion about need and method of synaptic connection weights adjustment of artificial neural network, thus expanding training algorithm.

Let's take the conditional task of defining the functional text style. There is text and author. The text extracts such conventional features as average length of sentences, average length of paragraphs, presence of direct speech, frequency of using verbs and nouns. It is indispensable to determine which of the five traditional styles – official, scientific, publicistic, newspaper and belles-lettres – related to text. Input features are very conventional and are used only as an illustration of proposed method. Figure 7 shows a schematic representation of artificial neural network inputs and outputs that solves this task.

Each element of training sample set is store in knowledge base. Semantic neighborhood of immersed elements are expanded by already existing knowledge in system. Figure 8 shows a fragment of knowledge base that describes submerged sample of texts to determine its style. Its semantic neighbor-

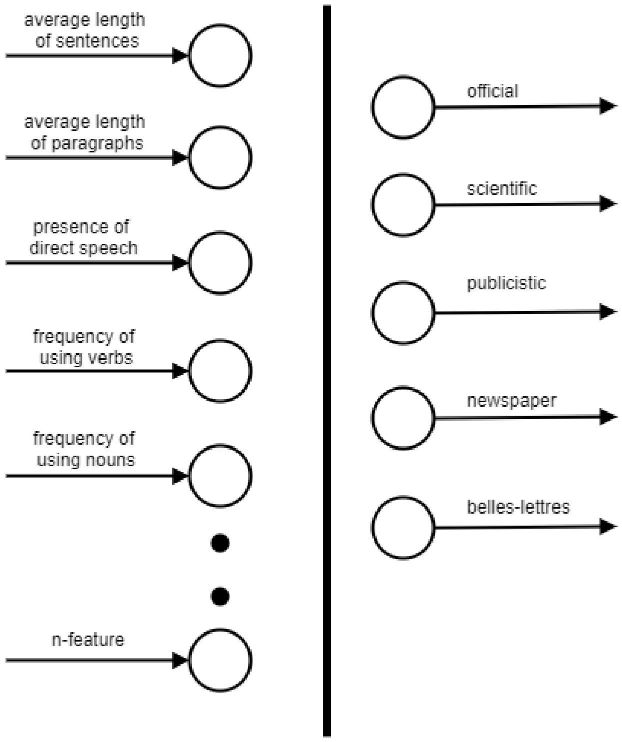

Figure 7. Schematic representation of artificial neural network inputs and outputs for solving text style determining task

hood is expanded by already existing in the system knowledge about text author and styles that he uses in its texts.

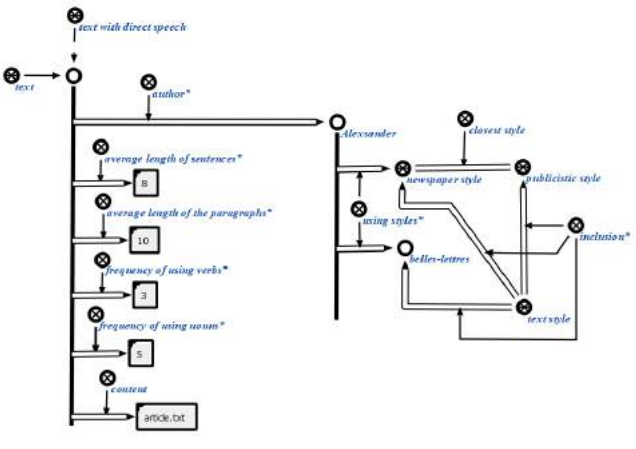

Figure 8. Knowledge base fragment that describes submerged sample of texts to determine its style

This information can be used to validate and adjust artificial neural network outputs during training. For example, during training, artificial neural network defined text style as publicistic. Knowledge base has validation rule, in which only style that author has previously used can be defined. Also, system has rule for adjustment the result, in which if text style that author has not previously used is defined, then it can be replaced with a similar one that author used. From knowledge base fragment that presented on Figure 8, system will be able to apply described rules of validation and adjustment, and replace the publicistic style to similar newspaper style. And depending on whether validation and adjustment rules were applied, as well as results of applying these rules, the algorithm of synaptic connection weight calculation in artificial neural network can change.

Described approach will help to increase degree of semantics influence during training, to discard or adjust deliberately unsuccessful sample elements, as well as to check artificial neural network outputs on plausibility.

IV. Intelligent environment of artificial neural network processing

A. *Review of existing frameworks*

The main drawback of all developed special frameworks is the requirement for knowledge about model structures, which should be used for each specific task. Despite the success of recent years, which reflects in the emergence of high-level frameworks with support of using graphics card (GPU) as hardware platform to calculations, and frameworks that are able to parallelize computations on different devices as a single computer or a whole cluster of computers [21], these developments are still only quantitative, but on the quality of such systems and their usability and comprehensibility for the end-user still need to perfect.

Despite the aforementioned drawback of existing frameworks, the threshold for occurrence in field of artificial neural networks, now more than ever low. Today, a large number of different libraries implement solutions based on intelligent algorithms. However, it is necessary to have the knowledge and skills to modify and improve the standard solution.

Here is a short overview of currently used open-source frameworks.

TensorFlow is one of the most popular libraries. Was developed by Google (2015). It allows to run training of models on several CPU and/or GPU devices. Available for different platforms, and supports different programming languages (C++, R, Python).

The main Tensorflow features includes

1) Multi-GPU support
2) Training across distributed resources (for example, on cloud)
3) Visualizing Tensorflow graph using TensorBoard – specific tool, which supports training process visualization and training data visualization
4) Model checkpointing – users of Tensorflow may stop training process and restart it from certain checkpoint.

Caffe/Caffe2 one of the first deep training libraries. It is written in C++, has a Python interface. Mainly oriented on training and the use of multi-layers and convolutional networks. It's created many pre trained networks for Caffe. In 2017 was proposed a new version Caffe2 by Facebook, which offers greater flexibility in building high-performance deep models. Can be used to develop architectures for run on mobile devices. It's well documented.

The Caffe features include the following:

- Trainable model is described in a special file with the extension "prototxt". In this file is stores the architecture of model and its main parameters.
- In the file with the same extension stores the parameters for the model training (number of training epochs, momentum, the training rate, weight decay, etc.)
- The training data are saved in files of special format (hdf5, lmdb) or in a text file with list of individual items (useful for images in separate files).
- The trained network is saved in a file with the extension "caffemodel". In the future it may be used directly to solve the basic tasks of machine training (classification, regression, etc.), and as an part of more complex models (for example, to solve the problem of detection of objects in images).
- To use the model, you need to have deploy-file that has the extension "prototxt", which, in fact, coincides completely with the file that contains the description of the model, except for settings that have a direct relation to the training (format of data, minimized function, etc.).

Theano was one of the first libraries that implement algorithms for deep training. It has some problems with scalability and computing on a cluster of graphics cards. Overall enjoyed sustained popularity with the experts in the field of deep training. It's often used as underlying framework for more high-level libraries that provide API wrappers (for example, Keras).

Keras provides a simplified interface to work with Theano, Tensorflow or CNTK. Very lightweight and easy to train and use. It's well documented. Allows you to create and train artificial neural network in a few lines of code. Written in Python.

Torch is framework based on Lua. It is possible to use libraries of C/C++ and CUDA. A very simple library, process of model building as simple as possible. There is a more modern implementation for Python, called PyTorch.

Additionally we need to refer on other frameworks, which are active use in last years (Fig. 9). These are Microsoft CNTK, MXNet and based on it Gluon, ONNX, released in September 2017 and presents open format to represent deep training models [22] and more others.

Number of frameworks is progressively to grow, but in fact each of them are completely repeat others and special differences include only additionally models and support of new hardware features.

B. The development of intelligent environment of artificial neural network processing

Intelligent environment is a environment with ability of automatic parameters selection of model (and, ideally, whole model) depending on tasks with minimal user involvement. This system allows people who are not specialists in the

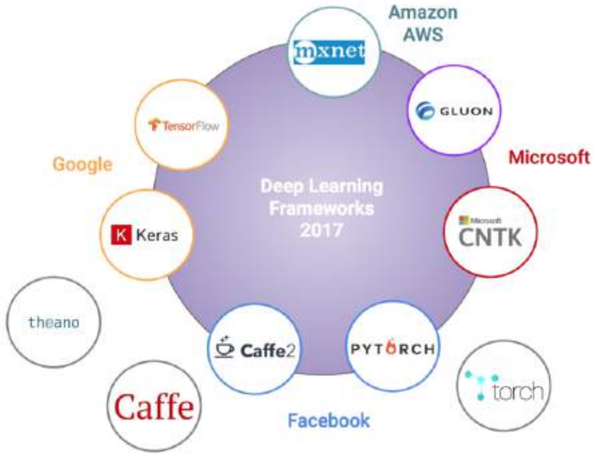

Figure 9. Open-source frameworks in 2017 [22]

field of machine training and artificial neural networks, to successfully apply the latest developments in their activity.

Obviously the relevance of such technologies is high. However, to develop similar software systems need to determine the main criteria for the selection of a particular model. It is known that the choice of artificial neural network to solve the task — is creative and empirical process, as it involves getting results through the selection and evaluation of the effectiveness of different architectures of networks.

But already developed approaches for the automatic generation of artificial neural network models [23], which show the advantage in comparison to manual selection of parameters.

V. Subject domain of artificial neural networks and its denotational and operational semantics

The main part of the system that developed using OSTIS([14], [24]) technology is the ontological model (knowledge base sc-model), which is built on the basis of the ontological approach. This approach includes building of ontologies as systems of absolute and relative concepts that describe a particular subject domain. In the OSTIS technology, the concept of ontology is defined as a specification of the subject domain [1], [25], its typology is specified.

A. Representing neural networks in knowledge base

The proposed approach is based on the use of knowledge bases corresponding to the model of the unified semantic representation of knowledge. This model uses homogeneous semantic networks which are semantic networks based on the basic set-theoretic semantic interpretation. This interpretation built on the (situtational) (non-)member-of relation whose links are denoted by sc-arc's. This relation links elements to a set and are relation of single basic type only [1]. The languages that included in the model of the unified semantic representation of knowledge are called sc-languages, the texts of which consist of sc-elements, and formed from its situative sets, structures and ontological representations, ontological models are called sc-sets, sc-structures and sc-models. The languages semantics of the unified model of knowledges semantic representation corresponds to the model of situative sets. Situative sets represent a more flexible and adequate apparatus for representing knowledge than classical sets, allowing to consider NOT-factors of knowledge and to adapt to the represented problem area while preserving the ontological model and its semantics. This is achieved not only by considering temporal properties, which makes it possible to interpret the system of situative sets as the development of systems of L-fuzzy sets, but also due to the fact that within the model of unified semantic representation of knowledge the process of using the apparatus of situative sets can be dynamic that support by dynamic of alphabetic labels. Therefore, it seems expedient to use situative sets and its advantages to represent artificial neural networks using the model of unified semantic representation of knowledge. It should also be noted that semantics of the texts of sc-languages in the model of unified semantic representation of knowledge is a model one, but it is possible to describe it in denotational form, in this case we can talk about the denotational semantics of sc-languages on basis of situative sets. Thus, when representing artificial neural network, each node (vertex) that is not receptor can be treated as a denotation of the situative set (sc-set) of all its vertices from which the signal comes to this node (Fig. 10.).

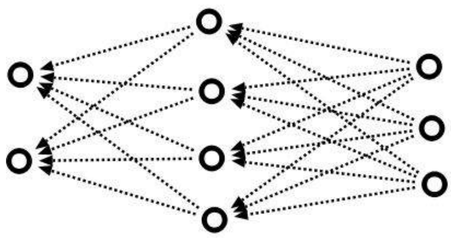

Figure 10. Representation of multilayer artificial neural network

Each receptor vertex is denotation of situative set of attributes (components of a features description) of the entities it expresses. Artificial neural network is represented as a situative set of sc-elements, denoting vertices, relations, its parameters, functions, properties, links and correspondence between them (Fig. 11, Fig. 12).

The synaptic connection weights and outputs of neural elements are represented as a situative measured value of a certain quantity that obtained by measuring the corresponding elements of artificial neural network (Fig. 13).

To specify the functions of the synaptic transformation and activation (Fig. 14), synaptic connection weights(Fig. 13) and outputs signals of neural elements, the key elements are used: *synaptic transformation function*, *activation function**, *synapse*, *neuron node* denoting situative relations.

Mathematical relationships can be established between these values, expressed through mathematical operations and relations, such as sum and product (Fig. 15).

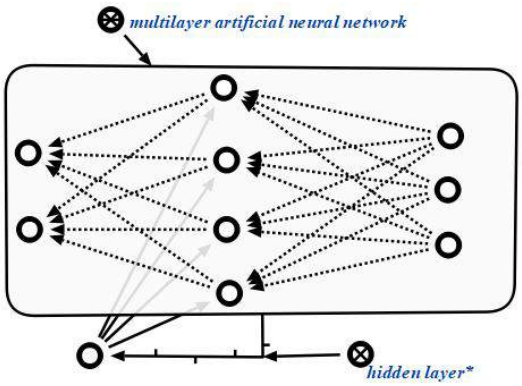

Figure 11. Representation of multilayer artificial neural network with a hidden layer

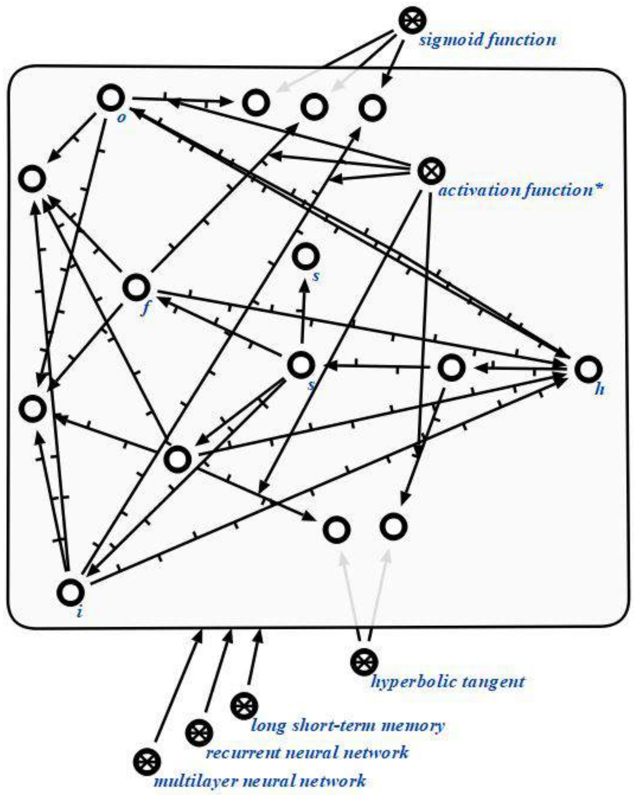

Figure 12. Long short-term memory representation

These mathematical relations can be given in general form by a sentence or program (Fig. 16).

Representation of artificial neural networks with complex structure makes it possible to represent its separate parts, such as hidden layers in multilayer artificial neural networks (Fig. 11), counter-connections in artificial neural networks of counter propagation (Fig. 17), reverse connections in recurrent artificial neural networks (Fig. 12) and etc.

As has already been said, each artificial neural network specifies a mapping between its inputs and outputs (Fig. 18, Fig. 19), a description of this fact is represented using the key

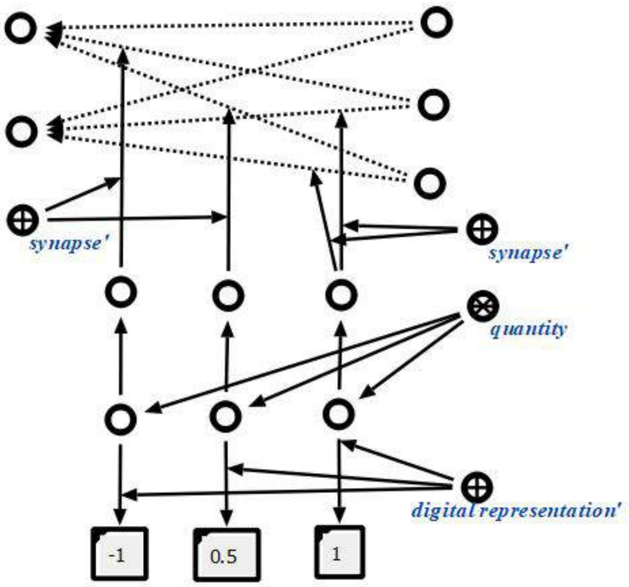

Figure 13. Representation of synaptic connection weights of artificial neural network

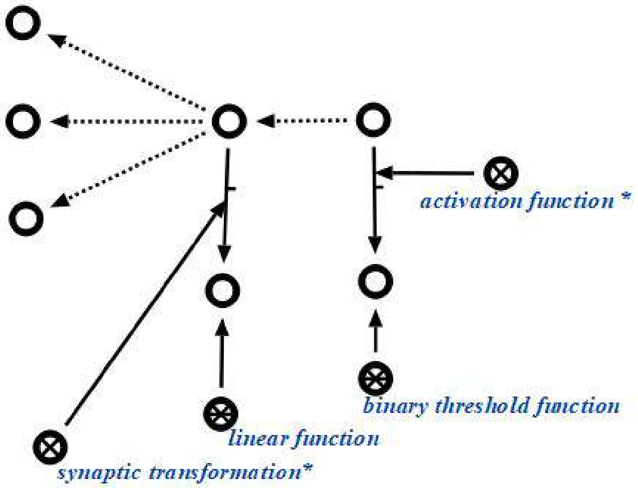

Figure 14. Representation of functions of synaptic transformation and activation of artificial neural network nodes

elements *mapping*, *domain of mapping*, *domain´*, *mapping of domain´*, *range of mapping*, *range´*, *mapping of range´*.

To clarify the values of input neural elements of artificial neural network, the key elements are *domain´*, *domain*, *relation´*. The listed key elements, along with others, are used for the specification of features(Fig. 20), whose values are the values of the input and output neural elements (receptors and effectors) of artificial neural network (Fig. 19).

B. The structure of subject domain

Subject domain is a structure that is defined on a set of sc-elements. On subject domains the same relations are defined as on sc-structures, which can be connected by inclusion relations (situative subset), isomorphic embedding, etc. These relations

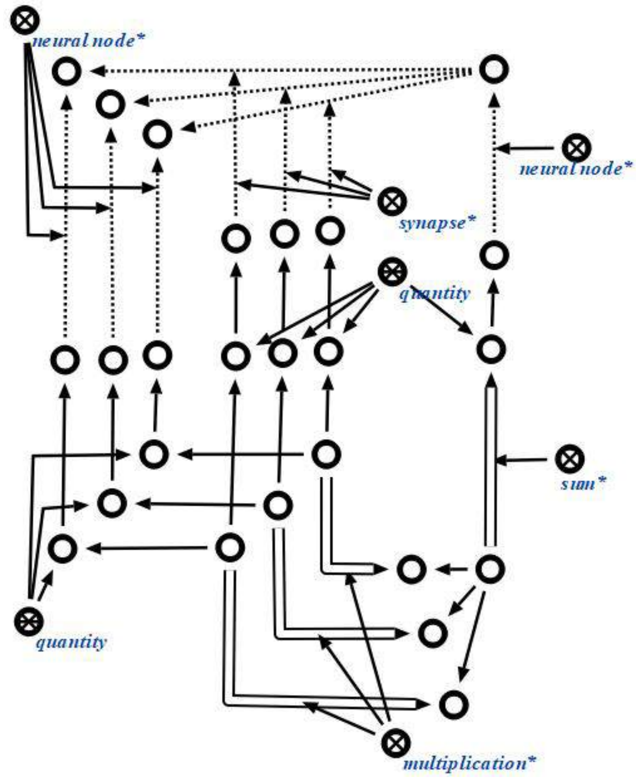

Figure 15. Representation of the output values of elements of artificial neural network and its mathematical relations

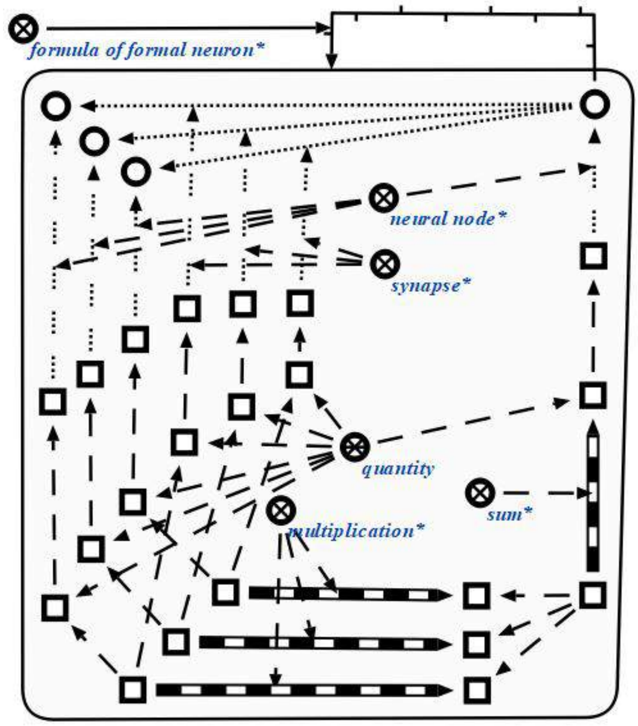

Figure 16. Representation of the program, a formal description of the dependence of the output values of the elements of artificial neural network

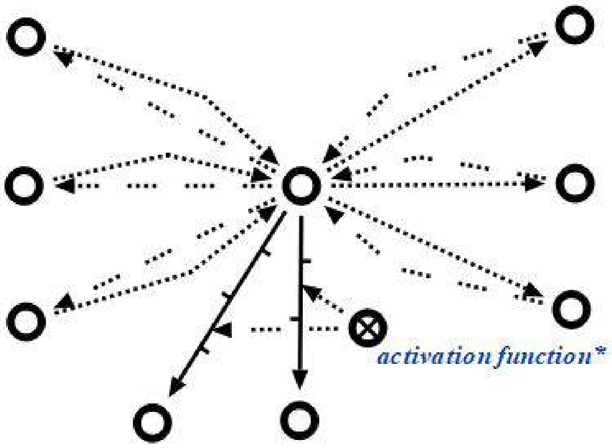

Figure 17. Representation of multilayer artificial neural network with a hidden layer

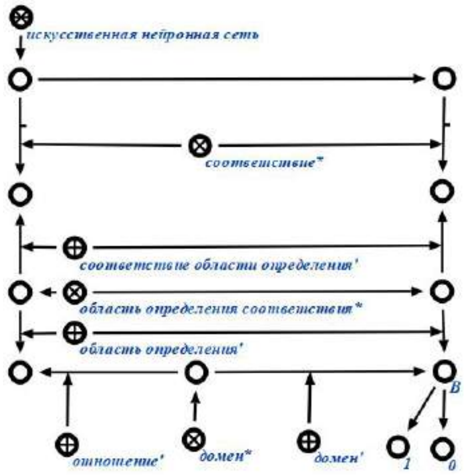

Figure 19. Representation of input artificial neural network element

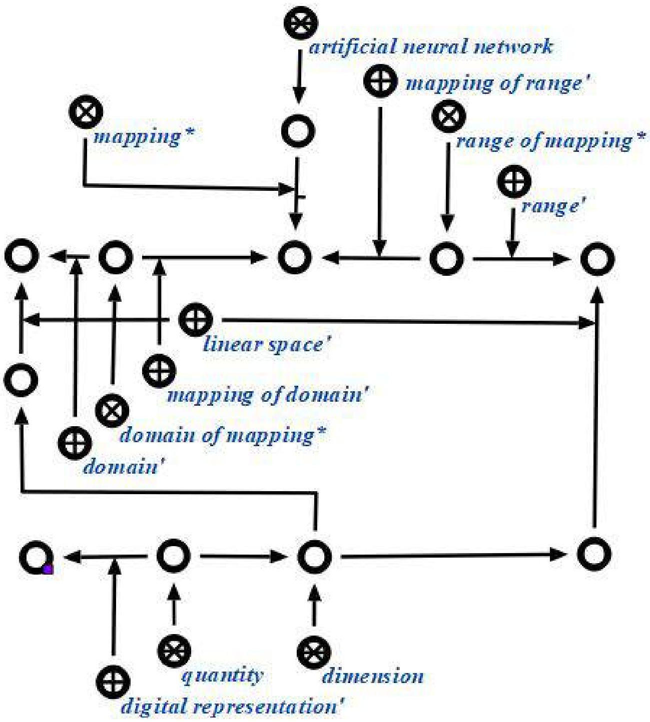

Figure 18. Representation of inputs and outputs of artificial neural network

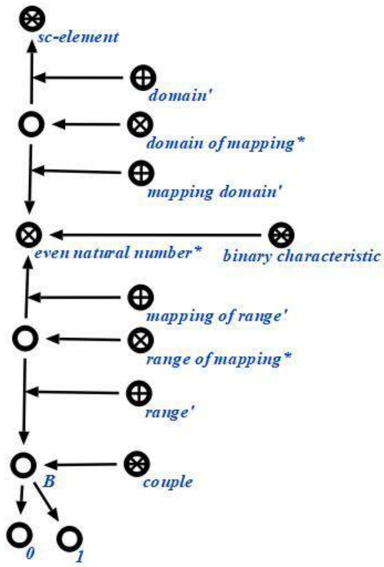

Figure 20. Representation of the binary feature

are analogous to the relations of the knowledge specification model [29]. It is necessary to distinguish these relations from situative relations of temporary inclusion, hypothetical inclusion, temporary isomorphic embedding and hypothetical isomorphic embedding, which are caused by the presence of such NOT-factors as incompleteness and uncertainty.

Subject domain **A** is a partial subject domain of subject domain **B** if it is included in **B** and its maximum investigated set is included in the maximal investigated set **B**.

The most important part in the development of the OSTIS knowledge base is the formation of a hierarchy of subject domains (based on the concept of a partial subject domain) that determines the structure of knowledge base and its specification [25]. Consideration of knowledge base from the position of the relationship with subject domain allows us to consider the objects under study at different levels of detailing, which are reflected in different types of ontologies that describe a certain direction of description the objects characteristic within the described subject domain. Such ontologies include: structural specification of subject domain, logical ontology, logical system of concepts and its definitions, set-theory ontology.

Artificial neural networks:

- Finite-valued artificial neural networks
  - •• Binarized artificial neural networks

- ••• Binary artificial neural networks
- ••• Bipolar artificial neural networks
- •• Ternary artificial neural networks
- Complex-numerical neural networks
  - •• Real-numerical artificial neural networks
    - ••• Rationally-numerical artificial neural networks
      - •••• Integer artificial neural networks
- Rational neural networks
- Dimension-increasing artificial neural networks
- Dimension-decreasing artificial neural networks
- Discontinuous artificial neural networks
- Continuous artificial neural networks
- Differentiable artificial neural networks
- Nondifferentiable artificial neural networks
- Homogeneous artificial neural networks
- Heterogeneous artificial neural networks
- Artificial neural networks without contextual neurons
- Artificial neural networks with contextual neurons
- Artificial neural networks with contextual connections
- Artificial neural networks without contextual connections
- Artificial neural networks with hidden neurons
- Artificial neural networks without hidden neurons
- Multilayer artificial neural networks
- Single-layer artificial neural networks
- Stochastic artificial neural networks
- Deterministic artificial neural networks
- Relaxation artificial neural networks
  - •• Relaxation networks with chaotic behavior
  - •• Relaxation networks with stable behavior

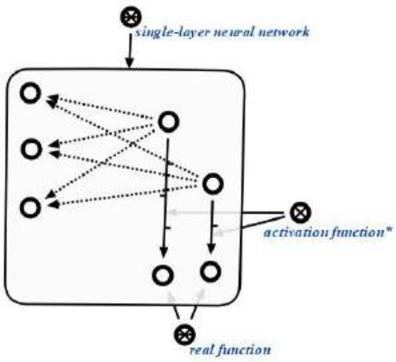

Figure 21. Single-layer artificial neural network

In addition to the distinguished classes of artificial neural networks (Figure 21, Figure 22.) and corresponding subject domains, it is possible to single out tasks, classes of artificial neural networks and corresponding subject domains associated with the training of artificial neural networks. There are two main classes:

- Training of artificial neural networks based on the target values of artificial neural network output elements.

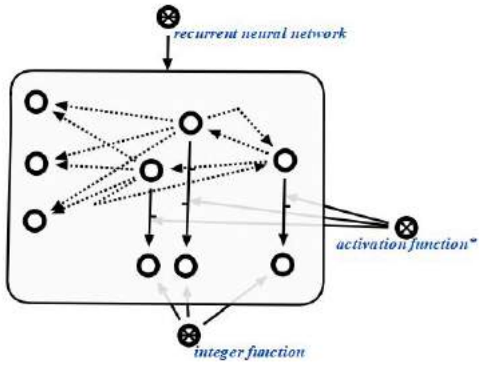

Figure 22. Recurrent artificial neural network

- Training of artificial neural networks without using the target values of artificial neural network output elements.

The first class includes subclasses:

- Training of artificial neural networks based on delta rule.
- Training of artificial neural networks based on gradient methods.
  - •• Training of artificial neural networks using the method of back propagation of an error.
  - •• Training of artificial neural networks by the conjugate gradient method.
  - •• training of artificial neural networks using the variable metric method.

The private classes of second class are:

- training of artificial neural networks without using target values based on the Hebb rule [26].
- training of artificial neural networks without using reference values based on the WTA rule [27].

In addition, the intelligent system may require knowledge of the following subject domains:

- Subject domain of the images and the features description of entities.
- Subject domain of set theory.
- Subject domain of logic.
- Subject domain of the natural numbers arithmetic(Peano).
- Subject domain of the integer number field.
- Subject domain of rational numbers field.
- Subject domain of dual rational numbers field.
- Subject domain of polynomial expressions.
- Subject domain of algebraic numbers field.
- Subject domain of dual algebraic numbers field.
- Subject domain of symbolic differentiation of polynomial expressions.
- Subject domain of functions and formal series.
- Subject domain of numerical sequences.
- Subject domain of common topology.
- Subject domain of graphs.
- Subject domain of measure theory.
- Subject domain of differentiable functions.
- Subject domain of linear spaces.
- Subject domain of tensor algebra.

- Subject domain of differentiable programming space.
- Subject domain of probability theory.
- Subject domain of cascades.
- Subject domain of cascades and dynamic systems.
- Subject domain of discrete optimization.
- Subject domain of optimization tasks.
- Subject domain of training artificial neural networks.

C. Agents of intelligent environment of artificial neural network processing

Operational semantics of knowledge bases that constructed in correspondence with model of unified semantic representation of knowledge is expressed in commands of knowledge processing sc-machine [28]. Each sc-machine corresponds to a formal information processing model, which language is some sc-language. Also, each sc-machine has initial information structure and set of operations that it implements, which can be programmed in procedures form.

In accordance with multiagent approach, each sc-machine can be implemented as a collection of agents (sc-agents). Operational semantics of artificial neural network in knowledge base is reduced to operational semantics of sc-agents that implements it and operational semantics of which, in turn, reduces to operational semantics of programs(commands) of its operations.

All operations of sc-agents are performed asynchronously, i. e. its are implemented in such a way that its joint work is reduced to its consistent work. All sc-agents interact via shared memory, passing each other data in form of semantic network constructs (sc-language texts that consist of sc-elements).

The condition for starting operation(initiating sc agent) is some event in the shared (graph) memory. Such events are changes in temporary non-belonging to temporal (actual) belonging of element to situative set, which is treated as set of commands for initiating sc-agents. Each command is data that will be processed by the agent. Such data can be a single sc-element (Figure 23) and its semantic neighborhood, available in shared memory, or some structure (sc-structure), denoted by such a sc-element (Figure 24).

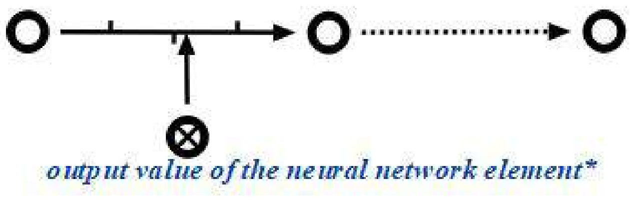

Figure 23. Data exchange between sc-agents

After operation start and complete, the temporary accessory is replaced with a temporary (actual) non-belonging (Figure 25), but new temporary (actual) accessories may appear that will initiate work of other sc-agents. Thus, work of artificial neural network and processes of processing knowledge in knowledge base are reduced to certain order of changes in temporal belonging to non-belonging and vice versa. Agents that provide semantic logging [30] artificial neural networks work form meta-descriptions as structures of special kind that provide ability to search and analyze order of artificial neural networks operations, which reduces to the interaction of sc-agents in shared graph memory.

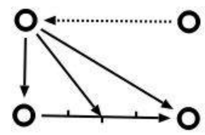

Figure 24. Transmission of data as a structure

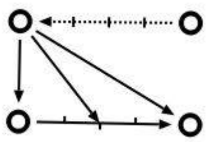

Figure 25. Processed data as a structure

For implementation of artificial neural networks, following agents classes was distinguished:

1) Agents for interpretation and processing of artificial neural networks inputs and outputs:
    - Agents of receptoral signals recording
    - Agents of receptoral signals validation and adjustment
    - Agents of effectoral signals reading
    - Agents of effectoral signals validation and adjustment
    - Agents of interpretation artificial neural networks nodes
    - Agents of interpretation artificial neural networks layers
    - Agents of semantic logging of artificial neural networks work
    - Agents of validation semantic logging of operation of artificial neural networks operations
2) Agents of training of artificial neural networks
    - Agents of artificial neural network training management
        - Agents of stochastic values generation
        - Agents of global training parameters changing
        - Agents of errors calculations
        - Agents of samples management
    - Agents of training processes semantic logging
3) Agents of integration of different artificial neural networks
    - Agents of cloning of artificial neural network
    - Agents of artificial neural network receptors search
    - Agents of artificial neural network effectors search

- Agents of synthesis of artificial neural network
  - Agents of synthesis of artificial neural network layers
- Agents of synthesis of multiple neurons agents that synaptically connected to neurons set
  - Agents of synthesis of multiple neurons agents that synaptically completely connected to neurons set
  - Agents of synthesis multiple neurons agents that synaptically incompletely connected to neurons set
- Agents of synthesis agents of backpropagation dual network
- Agents of synaptic connections deleting
- Agents of artificial neural network deleting
- Agents of translating ontological representation of artificial neural network to programming languages
- Agents of semantic logging of artificial neural networks integrations

CONCLUSION

The considered directions of integration of artificial neural networks with knowledge bases will help to solve more high-level tasks, making the solution of these tasks more structured and transparent. The implementation of the described intellectual system for the theory of artificial neural networks, as well as intelligent environment, helps to reduce the requirements to the skills of developers for methods of solving tasks using artificial neural networks. The possibility of artificial neural network introspection provided by the intelligent environment with the ability to memorize the state of artificial neural network during learning allows a deeper analysis of its work.

REFERENCES

[1] Horoshevskij B.F., Gavrilova T.A., Bazy znanij intellektual'nyh sistem(Intelligent Systems Knowledge Base), St.Petersburg, Piter, 2001, 384 p.

[2] McCulloch W., Pitts W. A Logical Calculus of Ideas Immanent in Nervous Activity, Bulletin of Mathematical Biophysics, 1943, pp. 115–133.

[3] Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, Biological Cybernetics, 1980, pp. 193-202.

[4] Bengio, Y. Learning deep architectures for AI. Foundations and Trends in Machine Learning, Foundations and Trends® in Machine Learning, 2009, Vol. 2: No. 1, pp 1-127

[5] Hinton G.E., Osindero S., Teh Y. A fast learning algorithm for deep belief nets, Neural Computation, 2006, pp. 1527–1554.

[6] Glorot, X., Bordes, A., Bengio, Y. Deep sparse rectifier neural networks, 14th International Conference on Artificial Intelligence and Statistics, 2011, vol. 15, pp. 315–323.

[7] Golovko V.A., Kroshchanka A. A., Rubanau U., Jankowski S., Learning Technique for Deep Belief Neural Networks, Communication in Computer and Information Science, 2014. vol. 440. . – pp. 136-146.

[8] Golovko, V.A. Deep learning: an overview and main paradigms // Optical Memory and Neural Networks (Information Optics), 2017, Volume 26, Issue 1, pp 1–17.

[9] Golovko, V. The nature of unsupervised learning in deep neural networks: A new understanding and novel approach / V. Golovko, A. Kroschanka // Optical memory and neural networks. – 2016. – No 3. – p.127–141.

[10] Rosenblatt F., Principles of neurodynamics; perceptrons and the theory of brain mechanisms, Washington Spartan Books, 1962, pp. 24-25.

[11] Pospelov D.A. Iskusstvennyj intellekt: V 3 knigah. Kniga 2. Modeli i metody: Spravochnik(Artificial Intelligence: In 3 books. Book. 2. Models and methods: Handbook), Moscow, Radio i svyaz', 1990, 304 p.

[12] Information Integration for Concurrent Engineering (IICE), IDEF5 Method Report, Texas, USA, Knowledge Based Systems, Inc., 1994.

[13] Graves A., Wayne G., Reynolds M., Harley T., Danihelka I., Grabska-Barwińska A., Colmenarejo S.G., Grefenstette E., Ramalho T., Hybrid computing using a neural network with dynamic external memory, Nature, 2016, pp. 471–476.

[14] Golenkov, V. V. Ontology-based Design of Batch Manufacturing Enterprises, Open Semantic Technologies for Intelligent Systems, 2017, pp. 265-280.

[15] Diederik P. Kingma, Jimmy Lei Ba., Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs.LG], December 2014.

[16] Klambauer G., Unterthiner T., Mayr A., Hochreiter S., Self-normalizing neural networks, CoRR, arXiv:1706.02515, 2017

[17] Donahue J., Long-Term Recurrent Convolutional Networks for Visual Recognition and Description, in IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, vol. 39, no. 4, pp. 677-691.

[18] Mikolov T., Chen K., Corrado G., Dean J., Efficient estimation of word representations in vector space, ICLR conference, arXiv preprint arXiv:1301.3781.

[19] Xiong S., Wang X., Duan P., Yu Z. and Dahou A. Deep Knowledge Representation based on Compositional Semantics for Chinese Geography , In Proceedings of the 9th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART, 2017, pp. 17–23.

[20] Pelevina M., Arefyev N., Biemann C., Panchenko A. Making Sense of Word Embeddings, 2017, arXiv preprint arXiv:1708.03390.

[21] Goldsborough P., A Tour of TensorFlow, ICLR conference, 2016, arXiv preprint arXiv:1610.01178.

[22] Battle of the Deep Learning frameworks – Part I: 2017, even more frameworks and interfaces. Available at: https://towardsdatascience.com/battle-of-the-deep-learning-frameworks-part-i-cff0e3841750 (accessed 2017, Dec)

[23] Liu H., Simonyan K., Vinyals O., Fernando C., Kavukcuoglu K., Hierarchical Representations for Efficient Architecture Search, ICLR conference, 2017, arXiv preprint arXiv:1711.00436.

[24] IMS. Available at: http://ims.ostis.net (accessed 2017, Nov)

[25] Davydenko I.T., Tekhnologiya komponentnogo proektirovaniya baz znanij na osnove unificirovannyh semanticheskih setej(The technology of component design of knowledge bases on the basis of unified semantic networks), Open Semantic Technologies for Intelligent Systems, 2013, pp. 185-190.

[26] Hebb D.O., The Organization of Behavior, New York, Wiley & Sons, 1949, 383p

[27] Grossberg S., Contour enhancement, short-term memory, and constancies in reverberating neural networks, Studies in Applied Mathematics, 52:213, 1973

[28] Shunkevich, D.V, Models and Means of Component Design of Knowledge-Processing Machines on the Basis of Semantic Networks. In: Open semantic technologies for designing intelligent systems (OSTIS2013): materials of the International Science and Technology Conf. Minsk, February 21-23, 2013) – Minsk: BSUIR, 2013, – P. 269–280

[29] Ivashenko, V.P., Modeli i algoritmy integracii znanij na osnove odnorodnyh semanticheskih setej(Models and algorithms of knowledge integration based on homogeneous semantic networks), Open Semantic Technologies for Intelligent Systems, 2015, pp 111-132.

[30] Ivashenko, V.P., Semanticheskoe protokolirovanie processov obrabotki znanij(Semantic logging of knowledge processing), Open Semantic Technologies for Intelligent Systems, 2017, pp. 110 – 111.

ИНТЕГРАЦИЯ НЕЙРОННЫХ СЕТЕЙ
С БАЗАМИ ЗНАНИЙ

Головко В.А., БГТУ
Голенков В.В., БГУИР
Ивашенко В.П., БГУИР
Таберко В.В., ОАО "Савушкин продукт"
Иванюк Д.С., ОАО "Савушкин продукт"
Крощенко А.А., БГТУ
Ковалёв М.В., БГУИР

Статья посвящена вопросам и направлениям интеграции искусственных нейронных сетей с базами знаний. Выделены два основных направления интеграции:

1) через входы и выходы искусственной нейронной сети, с целью использования такой интеграции баз знаний и искусственных нейронных сетей для решения прикладных задач;

2) через представление искусственных нейронных сетей с помощью онтологических структур и их интерпретацию средствами представления знаний в базе знаний, с целью создания интеллектуальной среды по разработке, обучению и интеграции различных искусственных нейронных сетей, совместимых с базами знаний.

Базы знаний, с которыми интегрируются искусственные нейронные сети, построены на основе однородных семантических сетей, а обработка знаний в них осуществляется с помощью многоагентного подхода.