Model for the representation of artificial neural networks and actions for their processing in the knowledge base

Mikhail Kovalev
Department of Intelligent Information Technologies
Belarusian State University of Informatics and Raddioelectronics
Minsk, Belarus
michail.kovalev7@gmail.com

***Abstract*—The article is dedicated to the problem of the integration of artificial neural networks with knowledge bases. The need for integration and development of a model for the representation of artificial neural networks (ANN) and actions for their processing in the knowledge base as the basis for such integration is justified. The subject domains and the corresponding ontologies of ANN and actions for processing ANN are described.**

***Keywords*—neuro-symbolic AI, ANN, knowledge base**

I. Introduction

Modern problem solvers of intelligent systems are increasingly faced with the need to solve complex problems using various traditional and intelligent methods of solving problems in a single information resource (in the limit – in a single knowledge base) [1].

One of the most popular intelligent methods of solving-problems is the method of solving problems usingartificial neural networks, which is caused, first of all,by the development of both the theory of ANN and thehardware capabilities of machines that are used for their training.

Neural network models show excellent results when working with unstructured data, but the traditional disadvantage of such models is the lack of human-understandable feedback, which could be called a reasoning chain. In other words, most neural network models work as a "black box" [2].

However, the complexity of modern intelligent systems that use neural network models, as well as the large amount of data processed by them, create the need to monitor, explain and understand the mechanisms of their work to verbalize the assessment and optimization of the activity of ANN.

Thus, we have, on the one hand, the need to use ANN in solving complex problems and, on the other hand, the need to explain how these problems were solved. In that context, it becomes relevant to develop neuro-symbolic approaches [3], [4], in particular, approaches for the integration of ANN and knowledge bases that use ontologies. Such integrated systems can combine:

- the possibility of semantic interpretation of the processed data, using the representation of the applied problems being solved by ANN as well as the specification of its input and output data;
- with a representation of the ANN structure itself, a description of its features and states, which make it easier to understand its operation [5].

There are two main directions of integration of ANN with knowledge bases:

- the building of intelligent systems that can use neural network methods jointly with other methods available in the system to solve complex problems. Such systems will be able to take into account the semantics of the problems at a higher level, which will make the solutions of these problems more structured and transparent;
- the building of an intelligent environment for the development, training and integration of various ANN compatible with knowledge bases through the representation of ANN with the help of ontological structures and their interpretation using knowledge representation. Such an environment will provide an opportunity for introspection of ANN, the ability to save the states of ANN after training and reconfiguration of the network, which will allow for a deeper analysis of its operation. Also, a formal description of the knowledge within the framework of the subject domain of ANN will help to lower the threshold for developers to enter the area of methods for solving problems with the help of ANN.

The development of both directions implies that the knowledge base stores knowledge (of various degrees of detail) about what neural network models it contains, how they are internally organized, what problems they can solve, etc. Therefore, the basis for the development of both directions is the development of a model for the representation of artificial neural networks and actions for their processing in the knowledge base. The purpose of this article is to develop such a model.

II. Proposed approach

The basis of the proposed approach is the usage of the OSTIS Technology and its basic principles for building a knowledge base [6].

Within the framework of the OSTIS Technology, powerful tools have been developed. Its allow describing any type of knowledge in a unified form, structuring the knowledge base according to various criteria as well as verifying its quality and editing the knowledge base directly while in operation [7].

The basis of the knowledge base built using the OSTIS Technology is a hierarchical system of subject domains and their corresponding ontologies. An ontology is interpreted as a specification of the system of concepts of the corresponding subject domain.

Further, the development of any subject domain and ontology will mean the development of a subject domain and ontology in the terminology of the OSTIS Technology.

Knowledge in knowledge bases built using the OSTIS Technology is represented using an SC-code. To record formalized texts, in the article, the variants of the external display of SC-code constructions are used – SCg (graphic version) and SCn (hypertext version). It is possible to examine in detail the representation of knowledge with the help of the SC-code here [8].

Thus, the specified problem of developing a model for the representation of artificial neural networks in the knowledge base and actions for their processing within the framework of the OSTIS Technology is reduced to the development of:

- the subject domain and the corresponding ontology of artificial neural networks;
- the subject domain and the corresponding ontology of actions for processing artificial neural networks.

III. Subject domain and the corresponding ontology of artificial neural networks

The subject domain of artificial neural networks contains a formal description of knowledge about particular artificial neural networks. The corresponding ontology contains a formal description of the concepts that are necessary for the representation of such knowledge.

The maximum class of objects of research in the subject domain of artificial neural networks is the ***artificial neural network***. This concept denotes a class, to which all entities that denote specific ANN belong.

An ***artificial neural network*** is a set of neural elements and connections between them [9]. The ANN consists of ***formal neurons***, which are interconnected by ***synaptic connections***. Neurons are organized in ***layers***. Each neuron of the layer receives signals from the synaptic connections that come in it, processes them in a single way using the ***activation function*** prescribed for it or for the entire layer and transmits the result to the synaptic connections that go out of it.

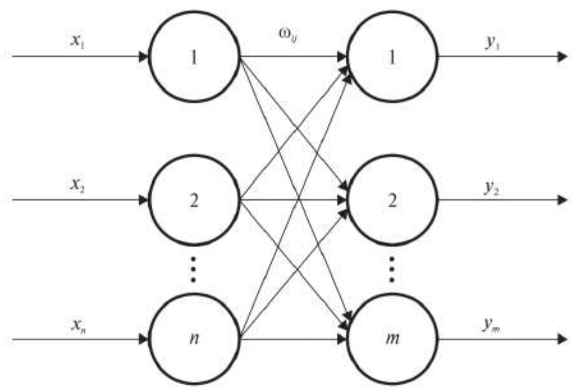

Figure 1. The general scheme of the artificial neural network [9]

Figure 1 shows the general scheme of the ANN.

We will call a set of information, which denotes data about the structure of ANN layers, formal neurons, synaptic connections and activation functions, i.e., something that can be trained and used to solve problems, the ANN architecture.

In accordance with the ANN architecture, it is possible to distinguish the following hierarchy of classes of ANN. Let us consider this hierarchy in the SCn-code.

artificial neural network
⇒ *subdividing**:
*Typology of ANN on the basis of the directivity of connections^*
= {• *ANN with direct connections*
⇒ *decomposition**:
{• *perceptron*
⇒ *decomposition**:
{• *Rozenblatt perceptron*
• *autoencoder ANN*
}
• *support vector machine*
• *ANN of radial-basis functions*
}
• *ANN with inverse connections*
⇒ *decomposition**:
{• *Hopfield ANN*
• *Hamming ANN*
}
• *recurrent artificial neural network*
⇒ *decomposition**:
{• *Jordan ANN*
• *Elman ANN*
• *multi-recurrent ANN*
• *LSTM-element*
• *GRU-element*
}
}

⇒ *subdividing**:

*Typology of ANN on the basis of completeness of connections^*

= {• *fully connected ANN*

• *weakly connected ANN*

}

Next, the key elements of the ontology of ANN, the concepts used to formalize the ANN architecture will be described.

A ***formal neuron*** is the main element of the *artificial neural network*, which applies its *activation function* to the sum of the products of input signals by weight coefficients (Golovko2017):

$$y = F\left(\sum_{i=1}^{n} w_i x_i - T\right) = F(WX - -T)$$

where $X = (x_1, x_2, ..., x_n)^T$ is an input signal vector; $W -$ $-(w_1, w_2, ..., w_n)$ – a vector of weight coefficients; T – a threshold value; F – an activation function. The role relation *formal neuron'* is used to denote the belonging of a particular formal neuron to any ANN.

A separate formal neuron is an artificial neural network with one neuron in a single layer.

formal neuron

:= [an artificial neuron]

:= [a neuron]

⊂ *artificial neural network*

⇒ *subdividing**:

{• *fully connected formal neuron*

:= [a neuron that has a complete set of connections with the neurons of the previous layer]

• *convolutional formal neuron*

⇒ *explanation**:

[A separate processing element of the ANN, which performs a functional transformation of the result of the operation of convolution of the matrix of input values using the activation function.]

⇒ *explanation**:

[A convolutional formal neuron can be represented by a fully connected formal neuron.]

• *recurrent formal neuron*

⇒ *explanation**:

[A formal neuron that has an inverse connection with itself or with other neurons of the ANN.]

}

Figure 2 shows the general scheme of a formal neuron.

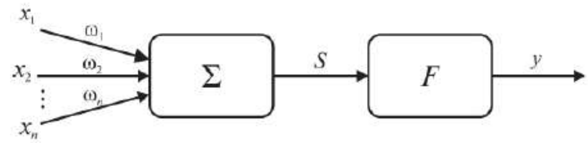

Figure 2. The general scheme of a formal neuron of the ANN [9]

The ***synaptic connection*** is a set of oriented pairs, the first components of which are the neurons, out of which the signal goes, and the second components are the neurons that receive this signal. When denoting the belonging of a particular synaptic connection to any ANN, the role relation *synaptic connection'* is used.

The ***weight sum**** is a non-role relation that connects a formal neuron with a number that is the sum of the products of input signals by the weight coefficients of the synaptic connections that come in the neuron.

The ***activation function**** is a non-role relation that connects a formal neuron with a function, the result of applying which to the weight sum of the neuron determines its output value. To describe such functions, in the knowledge base, classes of such functions (threshold, sigmoid, ReLU, etc.) and information about the cases, in which they are usually used, are described.

The ***ANN layer*** is a set of neural elements that receive information from other neural elements of the network in parallel at each clock period [9]. The activation function of a layer is the activation function of all the formal neurons of this layer. The description of the sequence of ANN layers with the configuration of each layer sets the ANN architecture. The configuration of the layer is set by the type, the number of formal neurons, the activation function. The description of the sequence of ANN layers with the configuration of each layer sets the ANN architecture. A separate layer is an artificial neural network with a single layer.

ANN layer

:= [layer]

:= [set of layers of artificial neural networks]

⊂ *artificial neural network*

⇒ *decomposition**:

{• *fully connected ANN layer*

:= [a layer, in which each neuron has a connection with each neuron of the previous layer]

• *convolutional ANN layer*

:= [a layer, in which each neuron is convolutional]

• *nonlinear transformation ANN layer*

:= [a layer that performs nonlinear transformation of input data]

⇒ *explanation**:

[As a rule, they are allocated into separate layers only in software implementations. In fact, they are considered as the final stage of calculating the output activity of any neuron – the application of the activation function.]

- *dropout ANN layer*

  := [a layer that implements the dropout regularization technique]

  ⇒ *explanation**:

  [This type of layer functions only during training the ANN.]

- *pooling ANN layer*

  := [a subsampling layer]

  := [a pooling layer]

:= [a layer that reduces the dimensionality of the input data]

- *ANN layer of the butch-normalization*

}

Also, within the framework of the subject domain, the hierarchy of ANN parameters is formalized.

ANN parameter

⊂ *parameter*

⇒ *decomposition**:

{• *configurable ANN parameter*

  := [an ANN parameter, the value of which changes during training]

  ⇒ *decomposition**:

  {• *weight coefficient of the synaptic connection*

  • *threshold value*

  • *convolution kernel*

  }

- *architectural ANN parameter*

  ⇒ *note**:

  [The ANN parameter that defines its architecture.]

  ⇒ *decomposition**:

  {• *number of layers*

  • *number of neurons*

  • *number of synaptic connections*

  }

}

The following concepts are also distinguished within the subject domain: *threshold formal neuron', input value of formal neuron*, output value of formal neuron*, distributing layer*, processing layer*, output layer**.

With the help of the allocated concepts, it becomes possible to formalize the architecture of a particular ANN in the knowledge base. Figure 3 shows an example of the formalization of a fully connected two-layer ANN with two neurons on the input layer and one neuron on the processing layer.

It should be noted that within the subject domain, it is possible to formalize knowledge about the ANN without formalizing knowledge about the ANN architecture. This is possible when the ANN is a sign of an external entity

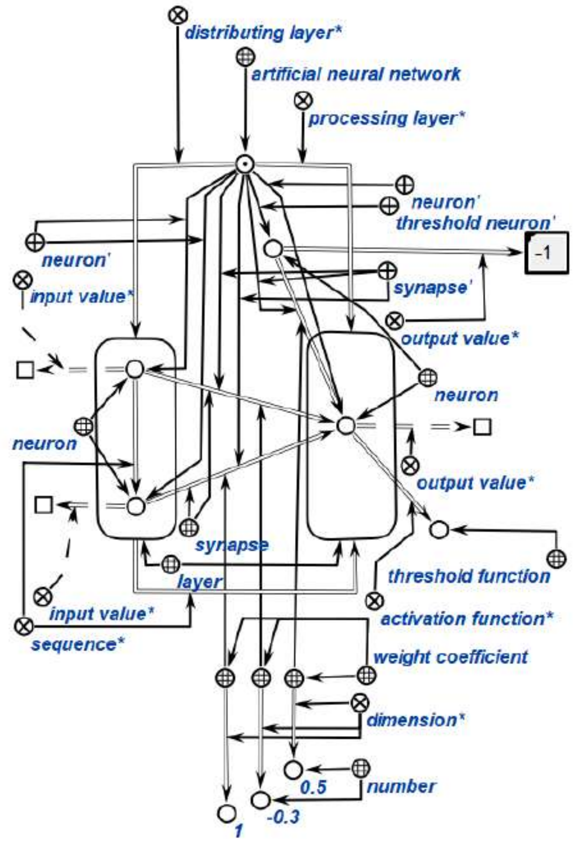

Figure 3. An example of the formalization of the architecture of an artificial neural network in the knowledge base

in relation to the knowledge base, for example, the ANN is implemented and works on a third-party service. In this case, it is sufficient to formalize the specification of such an ANN that includes:

- the ANN class;
- the ANN parameters;
- the type of problem being solved;
- the types of input and output data;
- the identifier, by which the specification of the ANN in the knowledge base can be correlated with the model in a third-party service;
- other knowledge that will help the system decide whether to use ANN as a method for solving a specific problem.

The description of each knowledge about ANN will allow the system that uses the subject domain under consideration to decide to use one or another ANN to solve a specific problem or to help the developer make such a decision when choosing an ANN to train for solving such a problem. For the same purposes, for each particular ANN, it is necessary to specify the problem that it can solve. The following classes of problems are identified, which can be solved with the help of the ANN with acceptable accuracy:

- the *classification problem*. The problem of creating a classifier, i.e., displaying $\tilde{c}: X \to C$, where $X \in \mathbb{R}^m$ is the attribute space of the input pattern, $C = C_1, C_2, ... C_k$ – a finite and usually small set of class labels.
- the *regression problem*. The problem of constructing an evaluation function based on examples $(x_i, f(x_i))$, where $f(x)$ is an unknown function. The *evaluation function* is the display of the form $\tilde{f}: X \to \mathbb{R}$, where $X \in \mathbb{R}^m$ is the attribute space of the input pattern.
- the *clustering problem*. The problem of dividing the set of input patterns into groups (clusters) according to some similarity metric.
- the *problem of reducing the dimensionality of the feature space*;
- the *control problem*;
- the *filtering problem*;
- the *detection problem*. It is a special case of a classification problem and a regression problem.
- the *problem with associative memory*.

IV. Subject domain and ontology of actions for processing artificial neural networks

In the previous section, we considered the subject domain and the corresponding ontology of ANN, within which the ANN architecture is described. However, the formalization of the ANN architecture is not enough to use ANN as a method of solving problems; the system should perform some *actions* on the configuration of the architecture, training and usage (interpretation) of ANN.

A. Action for processing the ANN

An ***action for processing the ANN*** is the maximum class of objects of research of the subject domain of actions for processing artificial neural networks. This concept denotes a class, to which all entities that denote specific actions for processing the ANN belong.

Within the framework of the *OSTIS Technology*, an *action* is defined as a purposeful *process* performed by one or more subjects (cybernetic systems) with the possible usage of certain tools [1].

Therefore, the *action for processing the ANN* is an action, the object of which is some ANN, the subject of which is an intelligent system, in the knowledge base of which this ANN is described.

Actions for processing the ANN are carried out by the appropriate group of agents [1].

Depending on whether the ANN is a sign of an entity external to the system memory, the elements of the set of the actions for processing the ANN are either elements of the set *action performed by a cybernetic system in its environment* or an element of the set *action performed by a cybernetic system in its memory*.

Let us consider the hierarchy of classes of actions for processing the ANN in the SCn-code.

action for processing the artificial neural network
:= [action for processing the ANN]
:= [action with the artificial neural network]
⊂ *action*
⇒ *decomposition**:
  {• *action for configuring the ANN*
   • *action for configuring the weight coefficients of the ANN*
   • *action for interpreting the ANN*
  }

action for configuring the ANN
⊂ *action for processing the ANN*
⇒ *decomposition**:
  {• *action for creating the ANN*
   • *action for editing the ANN*
   • *action for deleting the ANN*
   • *action for configuring the ANN layer*
    ⇒ *decomposition**:
     {• *action for adding a layer to the ANN*
      • *action for editing a layer of the ANN*
      • *action for deleting a layer of the ANN*
      • *action for setting the activation function of the neurons of the ANN layer*
      • *action for configuring a neuron in the ANN layer*
       ⇒ *decomposition**:
        {• *action for adding a neuron to the ANN layer*
         • *action for editing a neuron in the ANN layer*
         • *action for deleting a neuron from the ANN layer*
         • *action for setting the activation function of a neuron in the ANN layer*
        }
     }
  }

action for configuring the weight coefficients of the ANN
⊂ *action for processing the ANN*
⊃ *action for training the ANN*
⊃ *action for initializing the weights of the ANN*
  ⊃ *action for initializing the weights of neurons of the ANN layer*
    ⊃ *action for initializing the weights of the neuron of the ANN*

Since as a result of the actions for processing the ANN, the object of these actions, a particular ANN, can change significantly (the configuration of the network, its

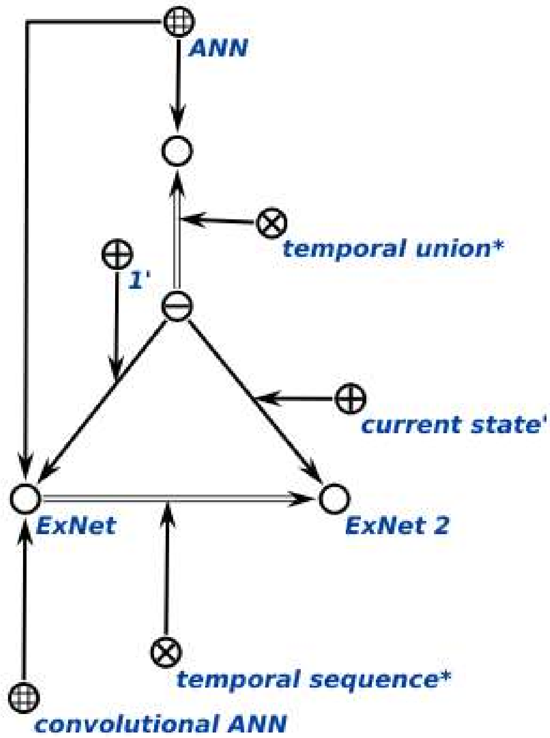

Figure 4. The representation of the artificial neural network as a temporal union of all its versions

weight coefficients change), the ANN is represented in the knowledge base as a temporal union of all its versions. Each version is the ANN and a temporal entity. On the set of these temporal entities, a temporal sequence is set with the indication of the first and last versions. Specific knowledge is described for each version. The knowledge common to all versions is described for the ANN, which is a temporal union of all versions.

Figure 4 shows an example of the representation of the ANN as a temporal union of all its versions.

B. Action for training the ANN

Special attention should be paid to the description of the ***action for training the ANN*** as a basic action, as a result of which it is possible to obtain an ANN suitable for usage as a method for solving problems.

An ***action for training the ANN*** is an action, during which a certain *method of training the ANN* is implemented with the specified *parameters of training the ANN*, *optimization method* and *loss function* on a given set of input data – a *training dataset*.

Let us consider the key concepts used to describe the action for training.

The ***method of training the ANN*** is a method of iterative search for optimal values of the configurable parameters of the ANN, which minimize some given loss function. It is worth noting that although the purpose of using the method of training is to minimize the loss function, the "utility" of the model obtained after training can be assessed only by the achieved level of its generalizing ability. The role relation *method of training'* is used to indicate the belonging of a particular method of training to any action for training the ANN.

Let us consider the hierarchy of methods of training the ANN in the SCn-code.

method of training the ANN
⊂ *method*
⊃ *method of training with a teacher*
  ⇒ *explanation**:
    [a ***method of training with a teacher*** is a method of training using the set target variables.]
  ⊃ *method of backward propagation of the error*
    := [MBPE]
    ⇒ *note**:
      [MBPE uses a certain optimization method and a certain loss function to implement the phase of backward propagation of the error and change the configurable ANN parameters. One of the most common optimization methods is the method of stochastic gradient descent.]
    ⇒ *note**:
      [It should also be noted that despite the fact that the method is classified as one of the methods of training with a teacher, in the case of using MBPE for training autoencoders, in classical publications, it is considered as a method of training without a teacher, since in this case there is no marked data.]
⊃ *method of training without a teacher*
  ⇒ *explanation**:
    [a ***method of training without a teacher*** is the method of training without using the set target variables (in the self-organization mode)]
  ⇒ *explanation**:
    [During the performance of the algorithm of the method of training without a teacher, useful structural properties of the set are revealed. Informally, it is understood as a method for extracting information from a distribution, the dataset for which was not manually annotated by a human [10].]

An ***optimization method*** is a method for minimizing the target loss function during training the ANN. When denoting the belonging of a particular optimization method to any method of training the ANN, the role relation *optimization method'* is used.

optimization method
⊂ *method*
⇒ *definition**:
[an ***optimization method*** is a method for minimizing the target loss function during training the ANN]
⇒ *inclusion**:
- *SGD*
  := [stochastic gradient descent]
  := [SGD]
- *Nesterov method*
- *AdaGrad*
  := [adaptive gradient]
- *RMSProp*
  := [root mean square propagation]
- *Adam*
  := [adaptive moments]

The ***loss function*** is a function used to calculate the error, which is calculated as the difference between the factual reference value and the predicted value being obtained by the ANN. When denoting the belonging of a particular loss function to any method of training the ANN, the role relation *loss function'* is used.

loss function
⊂ *function*
⇒ *inclusion**:
{• *MSE*
  := [mean square error]
 • *BCE*
  := [binary cross entropy]
 • *MCE*
  := [multi-class cross entropy]
}

A ***parameter of training the ANN*** is a group of the most general parameters of the method of training the ANN. The following parameters of training the ANN are distinguished:

- the ***learning rate*** is a parameter that determines the rate of change in the weight coefficients of synaptic connections of the ANN in the training process.
- The *moment parameter* is a parameter used in the training process to eliminate the problem of the "blocking" of the training algorithm in the local minima of the minimized loss function. When training the ANN, the situation of stopping of the process at a certain point of the local minimum without achieving the desired level of generalizing ability of the ANN is frequent. To eliminate such an undesirable phenomenon, an additional parameter (moment) is introduced, which allows the training algorithm to "jump" through the local minimum and continue the process.
- The ***regularization parameter*** is a parameter used to control the level of retraining the ANN. ***Regularization*** is the addition of extra restrictions to the rules for changing the configurable ANN parameters to avoid retraining.
- The ***size of the training group*** is the size of the group from the dataset that is used to change the weight coefficients of synaptic connections at each elementary step of training.
- The ***training epoch*** is one iteration of the training algorithm, during which all the images from the training dataset were used once.

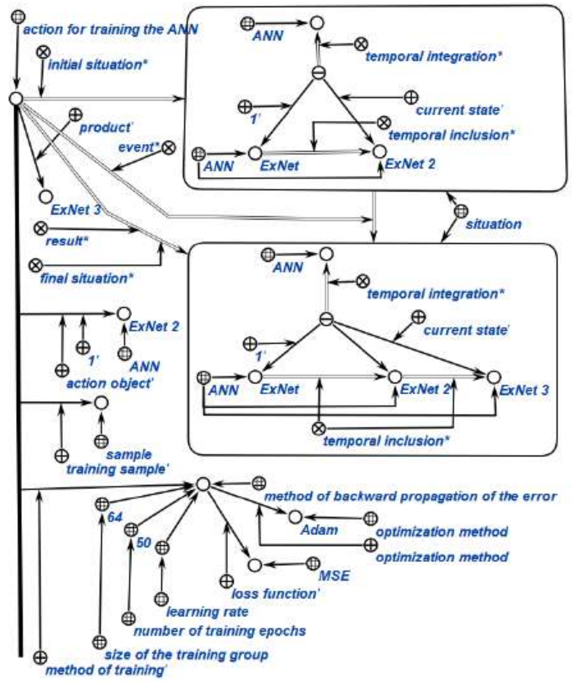

Figure 5. An example of the formalization of the action for training the artificial neural network in the knowledge base

With the help of the concepts highlighted in the subject domain under consideration, it becomes possible to formalize a specific action for processing the ANN in the knowledge base. Figure 5 shows an example of the formalization of the action for training the ANN.

Conclusion

A model for the representation of artificial neural networks and actions for their processing in the knowledge base on the ground of the OSTIS Technology has been developed. The model is represented in the form of subject domains and corresponding ontologies of artificial neural networks and their processing actions.

Within the framework of the described subject domains, with the help of their ontologies, it becomes possible to describe ANN in the knowledge base with all the variety of their architectures, classes, methods of training, etc.

The presence of such a possibility allows creating intelligent systems that can:

- use neural network methods along with other methods available in the system to solve complex problems in a common memory;
- configure, train and interpret ANN within the knowledge base for the purpose of their introspection and a deeper analysis of their work.

The described model creates the basis for further research in the field of developing:

- universal integration with the knowledge base of any neural network models, whose architecture is not formalized directly in the knowledge base itself;
- a group of agents capable of performing the described actions for processing ANN;
- an approach to automatic decision-making on the usage of a particular neural network model for solving system problems;
- an approach to using the knowledge base to improve the training results of artificial neural networks.

Acknowledgment

The author would like to thank the research groups of the Departments of Intelligent Information Technologies of the Belarusian State University of Informatics and Radioelectronics and the Brest State Technical University for their help and valuable comments. In particular, Vladimir Adamovich Golovko, Vladimir Vasilyevich Golenkov, Daniil Vyacheslavovich Shunkevich, Aleksandr Aleksandrovich Kroshchenko and Egor Vladimirovich Mikhno.

References

[1] D. Shunkevich, “Agentno-orientirovannye reshateli zadach intellektual’nyh sistem [Agent-oriented models, method and tools of compatible problem solvers development for intelligent systems],” in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed. BSUIR, Minsk, 2018, pp. 119–132.

[2] D. Castelvecchi, “Can we open the black box of ai?” *Nature News*, vol. 538, no. 7623, Oct 2016.

[3] T. R. Besold, A. d’Avila Garcez, S. Bader, H. Bowman, P. Domingos, P. Hitzler, K.-U. Kuehnberger, L. C. Lamb, D. Lowd, P. M. V. Lima, L. de Penning, G. Pinkas, H. Poon, and G. Zaverucha, “Neural-symbolic learning and reasoning: A survey and interpretation,” Nov. 2017, (accessed 2020, Jun). [Online]. Available: https://arxiv.org/pdf/1711.03902.pdf

[4] A. d’Avila Garcez, T. R. Besold, L. de Raedt, P. Földiak, P. Hitzler, T. Icard, K.-U. Kühnberger, L. C. Lamb, R. Miikkulainen, and D. L. Silver, “Neuralsymbolic learning and reasoning: Contributions and challenges,” *In: McCallum, A., Gabrilovich, E., Guha, R., Murphy, K. (eds.) Proceedings of the AAAI 2015 Propositional Rule Extraction under Background Knowledge 11 Spring Symposium on Knowledge Representation and Reasoning: Integrating Symbolic and Neural Approaches. AAAI Press Technical Report*, vol. SS-15-03, 2015.

[5] V. A. Golovko, V. V. Golenkov, V. P. Ivashenko, V. V. Taberko, D. S. Ivaniuk, A. A. Kroshchanka, and M. V. Kovalev, “Integration of artificial neural networks and knowledge bases,” in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual’nykh system [Open semantic technologies for intelligent systems]*, ser. 2, V. Golenkov, Ed. BSUIR, Minsk, 2018, pp. 133–145.

[6] V. Golenkov, N. Gulyakina, I. Davydenko, and D. Shunkevich, “Semanticheskie tekhnologii proektirovaniya intellektual’nyh sistem i semanticheskie associativnye komp’yutery [Semantic technologies of intelligent systems design and semantic associative computers],” *Otkrytye semanticheskie tehnologii proektirovanija intellektual’nyh sistem [Open semantic technologies for intelligent systems]*, pp. 42–50, 2019.

[7] I. Davydenko, “Semantic models, method and tools of knowledge bases coordinated development based on reusable components,” in *Otkrytye semanticheskie tehnologii proektirovanija intellektual’nyh sistem [Open semantic technologies for intelligent systems]*, V. Golenkov, Ed., BSUIR. Minsk , BSUIR, 2018, pp. 99–118.

[8] (2021, Jun) IMS.ostis Metasystem. [Online]. Available: https://ims.ostis.net

[9] V. A. Golovko and V. V. Krasnoproshin, *Nejrosetevye tekhnologii obrabotki dannyh [Neural network data processing technologies]*. Minsk: Publishing House of the BSU, 2017.

[10] J. Goodfellow, I. Benjio, and A. Courville, *Deep learning*. Moscow: DMK Press, 2017.

Модель представления искусственных нейронных сетей и действий по их обработке в базе знаний

Ковалёв М. В.

Статья посвящена рассмотрению проблемы интеграции искусственных нейронных сетей с базами знаний. Обоснована необходимость в интеграции и разработке модели представления искусственных нейронных сетей(и.н.с.) и действий по их обработке в базе знаний как основы такой интеграции. Описаны предметные области и соответствующие им онтологии и.н.с. и действий по обработке и.н.с.

Received 30.08.2021