Deep neural networks application in next-generation intelligent computer systems

Aliaksandr Kroshchanka
Brest State Technical University
Brest, Belarus
kroschenko@gmail.com

***Abstract*—In the article, an approach to building hybrid next-generation intelligent computer systems (NGICS) based on the integration of pre-trained models of deep neural networks and logical models developed using the OSTIS technology is proposed. To reduce the requirements for the size of the training dataset, the authors propose an alternative method for pre-training deep models. To achieve the interpretability of neural network models, the authors used methods from the Explainable AI (XAI) field.**

***Keywords*—Neuro-symbolic approach, OSTIS, deep neural networks, Explainable AI, SHAP, hybrid intelligent systems**

I. Introduction

The implementation of next-generation intelligent computer systems is one of the most promising objectives of the present and near future of AI science. New approaches built at the intersection of various directions of artificial intelligence allow eliminating or minimizing the impact of the shortcomings of individual methods, while enhancing the overall efficiency of intelligent systems. For example, when combining the capabilities of logical and neural network models in the context of implementing a neuro-symbolic approach in AI, we benefit from each model used. From logical models – the possibility of explaining the results for a user of the system who is not expert in the subject domain of the problem, from neural networks – the possibility of solving problems that are difficult to formalize (for example, data analysis and computer vision) [1]. Well-known scientists and researchers in certain fields of AI are increasingly declaring the need for compatibility between logical and neural network approaches (for example, [2]).

It should be noted that progress has already been made in these separate fields of research. For example, thanks to logical approaches in artificial intelligence (for example, an OSTIS technology [3]), systems for automating the work of manufacturing enterprises [4] are being developed. On the other hand, in the last decade, there has been a tendency towards the active usage of machine learning methods (and neural networks) to solve various problems. Thanks to the development of deep learning theory, new approaches and models have solved problems, that earlier have been solved successfully only by humans, and in some cases even outperform them (for example, [5]). Such tendencies definitely give grounds to further actively explore neural network models, applying these approaches in new, still poorly explored or high-cost fields.

The organization of the neural network training process is the cornerstone of the achievements obtained using these models. It should be noted that most of the research work currently is based on the usage of so-called pre-trained neural networks. These are networks that have already been trained, and they have been retrained for a new problem being solved (transfer learning [6]). Thus, model training is put on stream, making the threshold for entering the field lower than ever before. However, this potentially leads to a serious commercialization of the field of neural networks with the impossibility (primarily via the hardware lack) for ordinary researchers to train neural networks from the scratch. In addition, not in all cases, the usage of pre-trained models and transfer learning can help in solving new problems [7].

These circumstances form the need to develop new methods for training deep neural networks that reduce the requirements for hardware and the size of used dataset.

Despite the success of neural network models, there remains a certain caution in their usage, mainly due to the closed nature and non-interpretability of these models. We see a solution to this problem in the development of hybrid approaches.

In hybridization, a collision with the problem of integrating different models is inevitable. In the analysis of neural network models it is necessary to proceed from the output data generated by the model. The direct usage of the output data in the development of hybrid intelligent systems is possible and gives results that allow talking about their effectiveness [8]. However, in this case, the neural network model is used in the "black box" mode, without the possibility of interpreting the impact that the input data has on the final result. A successfully implemented interpretation would allow the integration of models at a qualitatively new level, supplementing the logical subsystem with new rules based on the identified patterns. In this case, the neural network part could be used in the process of training an intelligent system to form rules, and then, if necessary, "disable" and generate predictions only based on logical rules. On the other hand,

it is possible to use models in combination, getting the primary result from the neural network subsystem, and then form an interpretation for the user, which would allow leveling the wariness in using neural networks. In addition, it becomes possible to obtain information that allows more accurately assessing the quality of the model in the face of possible leaks and shifts in the data.

The authors of this article propose to implement such integration using the results of research obtained in the field of Explainable AI (XAI), which make it possible to assess the influence of the values of the features on outputs of the neural network. Such an evaluation can be represented as a sorted list of features that most strongly influence the result, as well as the formation of feature change intervals, within which the features make the greatest contribution to the results obtained at the output of the neural network.

Within this article, a solution to the problem of integrating neural networks and the OSTIS technology based on Explainable AI and deep learning methods is proposed.

The next sections are organized as follows: Section II describes the problem definition for the development of a next-generation intelligent computer system; Section III provides an overview of existing and proposed approaches in the field of training deep neural networks; Section IV gives a brief description of the SHAP method; the main practical results obtained by the authors are represented in Section V; Section VI summarizes the main conclusions of the proposed approach and describes the possibilities for its evolution.

II. Problem definition

Based on the abilities of next-generation intelligent systems, which are given in [3], we formulated a set of basic requirements for such systems:

- semantic compatibility with existing approaches in the field of artificial intelligence;
- evolution of the system;
- replaceability of system components – the ability to replace the active components of the system directly in the process of system work, for example, in the process of selecting a solution to a given problem;
- (self-)extensibility, simplicity in making changes to the existing set of components with the complication of their functionality;
- no side effects when using the system;
- ability to explain decisions;
- adaptive interface.

Semantic compatibility primarily refers to the possibility of using various AI technologies as system components without the need for constant redesign of the system in the face of changing theories and requirements.

The core of the next-generation system is the concept of compatibility of approaches in the field of AI. Fundamentally, this means the possibility of coexistence of approaches developed in different fields of science within the same system. For example, logical and fuzzy models or logical and neural network models, etc. At the same time, a fundamental boundary should be drawn between systems (or approaches on the basis of which intelligent systems were implemented) of the previous generation, where developers used hybrid methods – for example, neural networks, which can act as separate individuals of a population and thus participate in the implementation of a certain genetic algorithm. In such cases, it was about hybrid methods, but such methods did not include components that provide the necessary level of reflection of the intelligent system. Next-generation intelligent systems are able to explain the decisions they make. For such systems, one of the main requirements is their **evolution**, i.e. the ability to change not only their state but their qualities, which is the most valuable property. Given the presence of semantic compatibility, such a system is able to do this in the most natural way, and **replacement of components**, even if they have different nature but solving the same problem, is not difficult.

Extensibility in the presence of these properties is only a matter of developer competence. The **self-extensibility** of the system becomes the development of extensibility, which is represented in the possibility of generating its own components by the system based on the available knowledge. This part is the most valuable and even revolutionary, since the components offered by the system itself are unique ways to solve problems, as well as ones that may not have been known until now.

In addition to the listed properties, the system must be **free of side effects**, that is, it must not do anything for which it was not designed and intended. This functionality can be considered in the context of some “stop tap”, a set of directives of direct and unconditional action that determine the purpose setting of the system and its internal value system.

A property directly related to the previous one is the **ability to explain decisions**. Since the NGICS evolves and acquires the ability to create its own components, the correct interpretation of the obtained decisions is very important. Based on the purposes and directives available in the system, the system should describe the procedure for making decisions with the explaining for each step. Thus, the components created by the system will be documented by the system itself.

Finally, an **adaptive interface** forms a convenient user access to the NGICS. Such interface is not limited to adaptation to the user device through which the system is accessed but also to adaptation to the users themselves, to their physical abilities and limitations, habits and interests. Only such an interface can be called fully adaptive.

All these requirements must be taken into account when

creating a model of NGICS.

The general view of the proposed model is shown in Fig. 1.

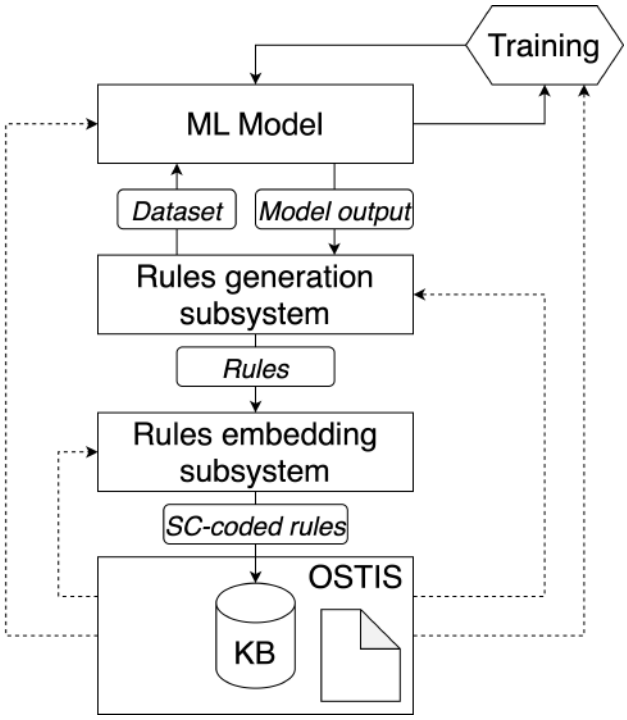

Figure 1. A model of a hybrid system (dashed lines indicate control actions)

Here, it is necessary to emphasize the fact that neural networks are ideally suited as component of next-generation intelligent computer systems. This is achieved mainly by the fact that these models are adaptive and can be used to solve various problems. In addition, such models support additional training during work. As the initial version of the neural network model, a model pre-trained on a small dataset can be used.

It should be noted that the neural network subsystem can also be placed in the OSTIS system and interact with the knowledge base as an agent. In this article, we propose a simplified architecture, focusing on the idea of interpretability of the neural network subsystem.

III. Deep neural networks training

Today, there are two main approaches to train deep neural networks: the first involves pre-training according to Hinton, the second involves special types of activation functions (ReLU), a large available training dataset, and some special regularization techniques (for example, dropout).

At the same time, it is necessary to distinguish between pre-training as a pre-executed procedure in accordance with the Hinton approach based on greedy layer-wise unsupervised learning with Restricted Boltzmann Machine as base trained network (let us call it pre-training of type I – Fig. 2) and pre-training as the process of preparing a pre-trained neural network that can be retrained on a different dataset to solve other problems using transfer learning (pre-training of type II). In the second case, traditional learning techniques can be used (for example, stochastic gradient descent with ReLU activation functions).

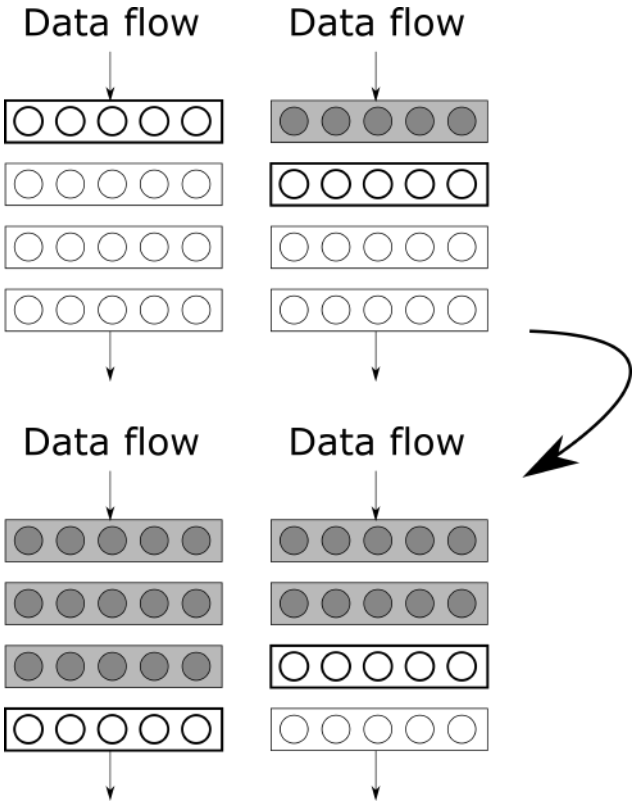

Figure 2. Greedy layer-wise pre-training

The choice of one or another approach to training deep neural networks depends on the size of the training dataset. So, if the dataset is large, pre-training of type II is applied. Otherwise, pre-training of type I is used. For small training datasets, this method overcomes overfitting [9].

The purpose of applying pre-training (both the first and second types) is to achieve some “good” initial initialization of the parameters of the neural network model. This allows starting the retraining process with a lower generalization error and speed it up.

In this article, a variant of the pre-training method based on the Hinton type is used. Further, the pre-training method proposed by Hinton will be called the classical method.

Let us consider a model of a restricted Boltzmann machine.

This model consists of two layers of stochastic binary neurons, which are interconnected by bidirectional symmetrical connections (Fig. 3). The input layer of neurons is called visible layer (X), and the output layer is called

hidden layer (Y). The restricted Boltzmann machine can generate any discrete distribution if enough hidden layer neurons are used [10].

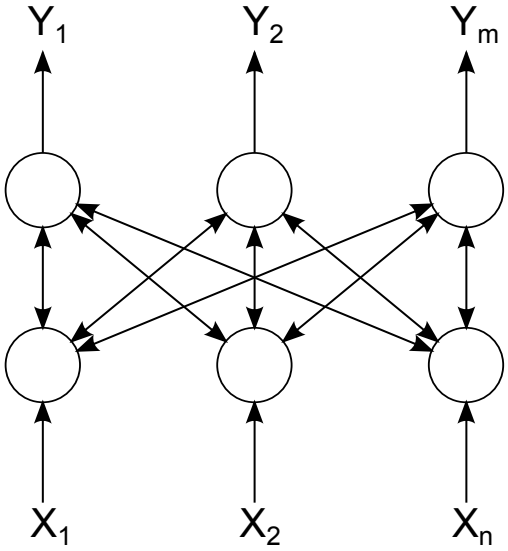

Figure 3. A Restricted Boltzmann Machine

This network is a stochastic neural network in which the states of visible and hidden neurons change in accordance with the probabilistic version of the sigmoid activation function:

$$p(y_j|x) = \frac{1}{1+e^{-S_j}},\ S_j = \sum_i^n w_{ij}x_i + T_j$$

$$p(x_i|y) = \frac{1}{1+e^{-S_i}},\ S_i = \sum_j^m w_{ij}y_j + T_i$$

where w_{ij} are the weight coefficients of the neural network, S_i, S_j are the weighted sums calculated for the neurons of the visible and hidden layers, respectively, T_i, T_j are the thresholds of the visible and hidden layers.

The rules for online learning of a restricted Boltzmann machine proposed in the classical method are as follows [11]:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(1)y_j(1))$$

$$T_i(t+1) = T_i(t) + \alpha(x_i(0) - x_i(1))$$

$$T_j(t+1) = T_j(t) + \alpha(y_j(0) - y_j(1))$$

where $x_i(0), x_i(1)$ are the original data of the visible layer and data, which been restored by the neural network, $y_i(0), y_i(1)$ are the original data of the hidden layer and data, which been restored by the neural network.

The last equations are obtained using the Contrastive Divergence algorithm with the parameter $k = 1$.

Rules for an arbitrary natural k can be obtained similarly:

$$w_{ij}(t+1) = w_{ij}(t) + \alpha(x_i(0)y_j(0) - x_i(k)y_j(k))$$

$$T_i(t+1) = T_i(t) + \alpha(x_i(0) - x_i(k))$$

$$T_j(t+1) = T_j(t) + \alpha(y_j(0) - y_j(k))$$

Rules for batch learning are as follows (case CD-1):

$$w_{ij}(t+1) = w_{ij}(t) + \frac{\alpha}{L}\left(\sum_{l=1}^{L}(x_i^l(0)y_j^l(0) - x_i^l(1)y_j^l(1))\right)$$

$$T_i(t+1) = T_i(t) + \frac{\alpha}{L}\left(\sum_{l=1}^{L}(x_i(0) - x_i(1))\right)$$

$$T_j(t+1) = T_j(t) + \frac{\alpha}{L}\left(\sum_{l=1}^{L}(y_j(0) - y_j(1))\right)$$

It should be noted that in order to obtain these rules, Hinton was guided by the idea of maximizing the likelihood function of the form:

$$P(x) = \sum_y P(x,y)$$

where $P(x,y)$ is the probability for a case of a visible and hidden neuron in the state (x,y), determined on the basis of the Gibbs distribution $P(x,y) = \frac{e^{-E(x,y)}}{Z}$, $Z = \sum_{x,y} e^{-E(x,y)}$ is the probability normalization parameter, E is the energy of the system in the state (x,y).

Finally, the function will take the form:

$$P(x) = \sum_y \frac{e^{-E(x,y)}}{Z} = \frac{\sum_y e^{-E(x,y)}}{\sum_{x,y} e^{-E(x,y)}}$$

Previously, the authors proposed an approach that generalizes the classical approach and demonstrated its effectiveness for some problems (for example, [12]).

The rules for online learning in accordance with the proposed approach for CD-1 are as follows:

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) \\ &\quad - \alpha((y_j(1) - y_j(0))F'(S_j(1))x_i(1) + \\ &\qquad (x_i(1) - x_i(0))F'(S_i(1))y_j(0)), \\ T_i(t+1) &= T_i(t) - \alpha(x_i(1) - x_i(0))F'(S_i(1)), \\ T_j(t+1) &= T_j(t) - \alpha(y_j(1) - y_j(0))F'(S_j(1)). \end{aligned}$$

The rules for batch learning in accordance with the proposed approach for CD-1 are as follows:

$$\begin{aligned} w_{ij}(t+1) &= w_{ij}(t) \\ &\quad - \frac{\alpha}{L}\left(\sum_{l=1}^{L}\Delta y_j^l(1)x_i^l(1)F'(S_j^l(1)) + \right. \\ &\qquad \left. \Delta x_i^l(1)y_j^l(0)F'(S_i^l(1))\right), \end{aligned}$$

$$T_j(t+1) = T_j(t) - \frac{\alpha}{L}\left(\sum_{l=1}^{L} \Delta y_j^l(1) F'(S_j^l(1))\right),$$

$$T_i(t+1) = T_i(t) - \frac{\alpha}{L}\left(\sum_{l=1}^{L} \Delta x_i^l(1) F'(S_i^l(1))\right),$$

where $\Delta y_j^l(1) = y_j^l(1) - y_j^l(0)$, $\Delta x_i^l(1) = x_i^l(1) - x_i^l(0)$

When obtaining these rules, the authors were guided by the idea of minimizing the mean squared error of the network (the case of using CD-1):

$$E_s(1) = \frac{1}{2L}\left(\sum_{l=1}^{L}\sum_{j=1}^{m}(\Delta y_j^l(1))^2 + \sum_{l=1}^{L}\sum_{i=1}^{n}(\Delta x_i^l(1))^2\right)$$

where $\Delta y_j^l(1) = y_j^l(1) - y_j^l(0)$, $\Delta x_i^l(1) = x_i^l(1) - x_i^l(0)$, L – size of the training dataset.

It is possible to prove the identity of these learning rules to the classical ones by using neurons with a linear activation function.

Thus, the following theorem can be proved:

Theorem. Maximizing the likelihood function of data distribution $P(x)$ in the space of synaptic connections of a restricted Boltzmann machine is equivalent to minimizing the mean squared error of the network E_s in the same space using linear neurons.

IV. Interpretability of ANN

The problem of interpretability of machine learning models is currently quite effectively solved by the Explainable AI methods [13].

In XAI methods such as LIME [14] and SHAP [15], only the data feeded to the model and the output returned by the model are used in the analysis. This type of methods belongs to the model-agnostic type, i.e. they can be applied to any machine learning model.

Our hybrid intelligent system model uses the SHAP (SHapley Additive exPlanations) approach to interpret the results obtained by the neural network.

The SHAP method is based on an attempt to explain changes in the predictions of the model caused by a change in the input features or the appearance of some information about the input feature. In this case, the contribution of each feature to the prediction of the model is calculated.

The SHAP method is based on the game theory. The key quantities used in evaluating the contribution of each feature to the overall output of the model are the Shapley values.

In this case, the players are features (the presence of the *i*-th player means the current value of the *i*-th feature in the example *x*, the absence of the *i*-th player means the undefined value of the *i*-th feature), and all the represented features define a set of players, called coalition.

Denote by $f : X \to Y$ the model under study, $x \in X$ is the selected test example for which the output value of the model is interpreted, $X \in \mathbb{R}^N$ is the feature space, N is the number of features (players), ν is a characteristic function that assigns a number to each coalition of players – its efficiency.

Further, assuming that some of the features in the example x are known and some are omitted (have undefined values), we obtain the vector x_S corresponding to the known features.

The Shapley values for each player are calculated using the following formula:

$$\phi(i) = \sum_{S \in 1,2,\ldots,N} \frac{|S|!(|N| - |S| - 1)!}{N!}\Delta(i, S)$$

where S defines the coalition of players and $\Delta(i, S)$ is the efficiency gained from adding player i to the coalition of players S:

$$\Delta(i, S) = \nu(S \cup i) - \nu(S)$$

In the SHAP method, the conditional expectation is used as the characteristic function for the set of features *S* of the example *x*:

$$\nu(S) = E[f(x)|x_S]$$

In practice, when calculating the characteristic function given by the last formula, simplifications are used (for example, Kernel SHAP modification).

V. Experimental results

For the experimental part of the research, we chose the well-known Fisher Irises dataset.

The size of this dataset is 150 examples, which are divided into train (120 examples) and test (30 examples) datasets. The examples describe the geometric shape of the iris flower. Each example contains 4 features (sepal length, sepal width, petal length, petal width) and a class label (0–2). It is required to classify the example according to the type of flower (Iris setosa, Iris virginica, Iris versicolor).

The usage of such simple dataset made it possible, on the one hand, to demonstrate the effect of pre-training on a dataset of a limited size and, on the other hand, to show the process of interpreting the model with the construction of simple rules, for which checking their corectness is not difficult.

For this problem, a series of experiments was carried out with the following training options:

- no pre-training – in this case we used backward propagation to train neural network from the scratch;
- with pre-training by the classical method;
- with REBA-based pre-training.

The training process used a model with a structure (4 – 10 – 10 – 3) with ReLU activation functions on all layers, except for the last one.

Table I
PRE-TRAINING PARAMETERS

mini-batch size	momentum	epochs count	train rate
4	0.5	5	0.01/0.04

Table II
TRAINING PARAMETERS

mini-batch size	momentum	epochs count	train rate
4	0.9	10	0.01

Tables I and II show the main parameters of pre-training and training stages.

A series of 100 computational experiments was carried out, the results of which were averaged. The results are represented in Table III.

Table III
RESULTS

pre-training method	test efficiency, %
RBM	91.0
REBA	92.6
Without pre-training	83.73

During the implementation of the SHAP method, Shapley values were obtained, on the basis of which visualizations were drawn. In Fig. 4, 5, and 6, the cumulative influence of individual features on irises type classification is shown.

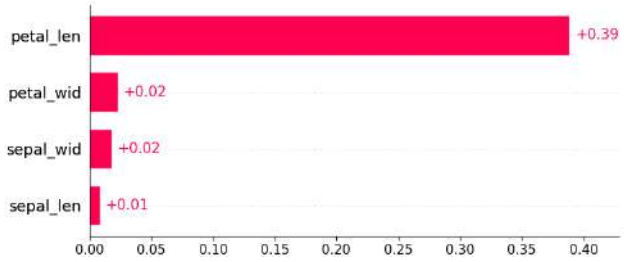

Figure 4. Influence of features on class 0 for identification (Iris setosa)

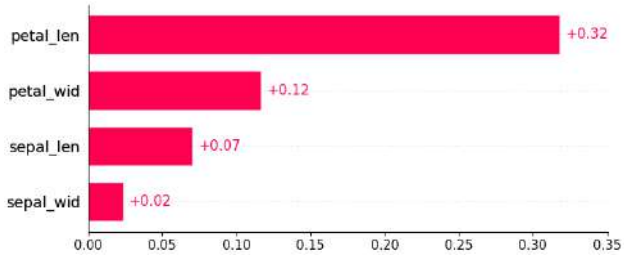

Figure 5. Influence of features on class 1 for identification (Iris virginica)

According to these images, it can be seen that the petal length feature has the greatest influence on determining

Figure 6. Influence of features on class 2 for identification (Iris versicolor)

the type of flower. This is confirmed by a more detailed study of the values dependence from this feature on the Shapley values (Fig. 7, 8, 9).

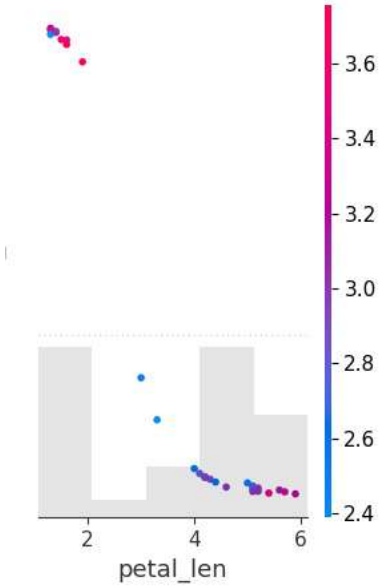

Figure 7. Dependences of feature values on Shapley values (class 0)

As can be seen from the represented visualizations, all values of the **petal length** feature are concentrated in 3 main intervals that directly affect the class identification ([1, 2], [2, 5], [5, 6]). The boundaries of the ranges for simplification are defined approximately. Based on these data, rules can be formulated:

1) If the value of the petal length feature is in the range from [1, 2], then define the class of the flower as **Iris setosa**
2) If the value of the petal length feature is in the range from [2, 5], then define the flower class as **Iris virginica**
3) If the value of the feature petal length is in the range from [5, 6], then define the flower class as **Iris versicolor**

These rules take into account only the value of one feature, in order to improve the characteristics of

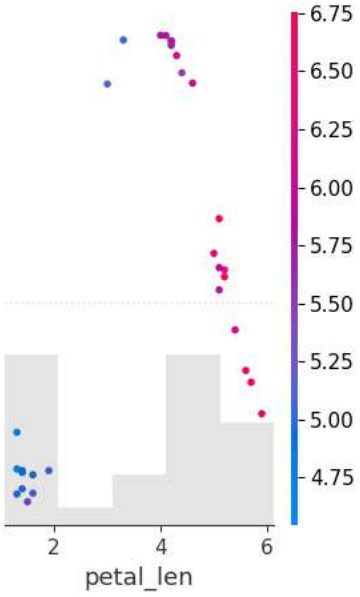

Figure 8. Dependences of feature values on Shapley values (class 1)

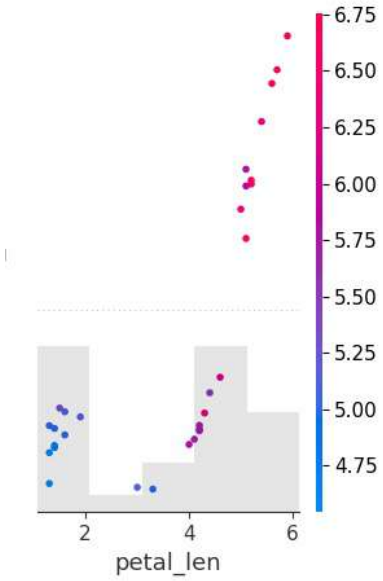

Figure 9. Dependences of feature values on Shapley values (class 2)

the classifying algorithm; the number of rules can be expanded by analyzing changes in other main features.

Finally, after forming a natural language representation of the rules, they can be easily represented in the SC-code or illustrated using its visual representation in SCg (Fig. 10).

VI. Conclusion

In the article, an approach to the implementation of next-generation intelligent computer systems is proposed, which allows integrating neural network and logical models created using the OSTIS technology. The proposed

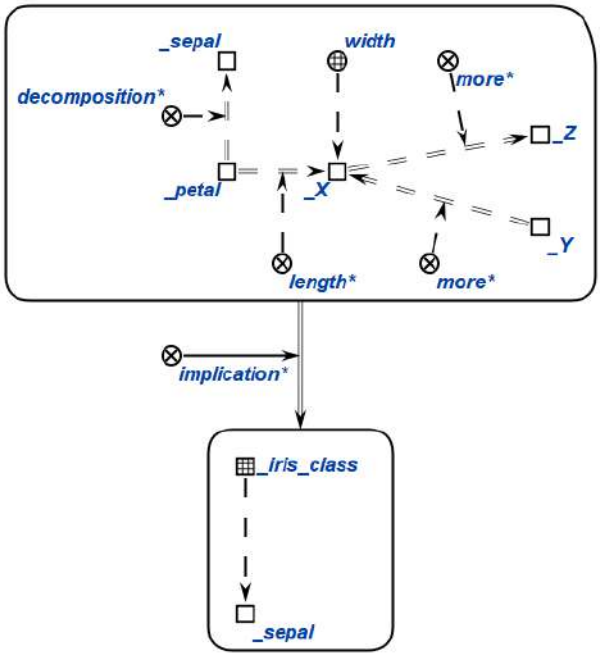

Figure 10. Representation of the rule for determining the type of flower in SCg

approach is based on the application of deep neural network pre-training methods and Explainable AI. The effectiveness of the proposed approaches to the pre-training of a deep neural network, as well as the approach to integration, is shown on the example of solving classification problem.

This approach can be used in the development of next-generation intelligent computer systems, for which the small amount of available training data and high requirements for the interpretation of the results often become critical factors.

As directions for further work, the authors see the development of the proposed approach in the context of studying the applicability to convolutional models, as well as studying the possibilities of interpreting models with homogeneous inputs (for example, when solving the problem of recognizing objects in an image, where the role of an individual pixel or superpixel is difficult to formalize).

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 in the work and valuable comments, in particular, Vladimir Golovko, Vladimir Golenkov, and Daniil Shunkevich.

The work was supported by the Belarusian Republican Foundation for Fundamental Research BRFFR, project F22KI-046.

References

[1] A. Kroshchanka, V. Golovko, E. Mikhno, M. Kovalev, V. Zahariev, and A. Zagorskij, "A Neural-Symbolic Approach to Computer Vision," in *Open Semantic Technologies for Intelligent Systems*, V. Golenkov, V. Krasnoproshin, V. Golovko, and D. Shunkevich, Eds. Cham: Springer International Publishing, 2022, pp. 282–309.

[2] Y. Cun, *Quand la machine apprend: La révolution des neurones artificiels et de l'apprentissage profond*. Odile Jacob, 2019. [Online]. Available: https://books.google.by/books?id=78m2DwAAQBAJ

[3] V. V. Golenkov, N. A. Gulyakina, D. V. Shunkevich, *Open technology for ontological design, production and operation of semantically compatible hybrid intelligent computer systems*, G. V.V., Ed. Minsk: Bestprint, 2021.

[4] V. Taberko, D. Ivaniuk, N. Zotov, M. Orlov, O. Pupena, and N. Lutska, "Principles of building a system for automating the activities of a process engineer based on an ontological approach within the framework of the Industry 4.0 concept," in *Otkrytye semanticheskie tekhnologii proektirovaniya intellektual'nykh system [Open semantic technologies for intelligent systems]*, vol. 5, Minsk, 2021, pp. 209–218.

[5] L. E. van Dyck, R. Kwitt, S. J. Denzler, and W. R. Gruber, "Comparing Object Recognition in Humans and Deep Convolutional Neural Networks—An Eye Tracking Study," *Frontiers in Neuroscience*, vol. 15, 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2021.750639

[6] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," 2019. [Online]. Available: https://arxiv.org/abs/1911.02685

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, no. 140, pp. 1–67, 2020. [Online]. Available: http://jmlr.org/papers/v21/20-074.html

[8] V. Golovko, A. Kroshchanka, M. Kovalev, V. Taberko, and D. Ivaniuk, "Neuro-Symbolic Artificial Intelligence: Application for Control the Quality of Product Labeling," in *Open Semantic Technologies for Intelligent System*, V. Golenkov, V. Krasnoproshin, V. Golovko, and E. Azarov, Eds. Cham: Springer International Publishing, 2020, pp. 81–101.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, no. 521 (7553), pp. 436–444, 2015.

[10] Y. Bengio, "Learning deep architectures for ai," *Foundations and Trends in Machine Learning*, no. 2(1), pp. 1–127, 2009.

[11] G. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural computation*, vol. 18, pp. 1527–54, 08 2006.

[12] V. Golovko, A. Kroshchanka, and E. Mikhno, "Deep Neural Networks: Selected Aspects of Learning and Application," in *Pattern Recognition and Image Analysis*. Cham: Springer International Publishing, 2021, pp. 132–143.

[13] A. Thampi, *Interpretable AI: Building Explainable Machine Learning Systems*. Manning, 2022. [Online]. Available: https://books.google.by/books?id=ePN0zgEACAAJ

[14] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You?: Explaining the Predictions of Any Classifier," 2016. [Online]. Available: https://arxiv.org/abs/1602.04938

[15] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf

Применение глубоких нейронных сетей в интеллектуальных компьютерных системах нового поколения

Крощенко А. А.

Статья посвящена модели гибридной интеллектуальной системы нового поколения, базирующейся на интеграции предобученных глубоких нейросетевых моделей и логических моделей технологии OSTIS. Для снижения влияния объема обучающей выборки на процесс обучения модели авторами предлагается альтернативный подход к предобучению глубоких нейронных сетей. Для достижения цели интерпретируемости нейросетей использовались методы из области Explainable AI.

Received 14.11.2022