UDC 004.054 DOI: 10.30748/zhups.2021.67.17

O. Tymochko[1], S. Osievskiy[1], O. Timochko[2], P. Bieliaiev[1]

[1]*Ivan Kozhedub Kharkiv National Air Force University, Kharkiv*
[2]*Kreditech Holding SSL GmbH, Hamburg, Germany*

METHOD OF TROUBLESHOOTING IN THE NEURAL NETWORK ENVIRONMENT OF INTELLECTUAL DECISION SUPPORTING SYSTEMS

The issues of error detection in artificial neural networks are considered. They are related to conflicting opinions of experts and / or limited (imperfect) description of the subject area. Approaches to their debugging are analyzed. Ways to improve existing approaches to debugging errors such as "forget about exclusion" are shown. Possible ways of application of the received decisions for debugging of errors "intersection of critical events" are shown. A formalized definition of the error of the neural network intelligent system is proposed, taking into account the requirements for efficiency and accuracy of information presentation. The issues of the influence of incorrect organization of machine learning on the accuracy of classification of elements of the neural network intellectual system are considered. The possibility of applying network contrast methods at the preparatory stage for testing the knowledge base of the intelligent decision support system is proved. This reduces the likelihood of errors of this type for these systems. The classification and analysis of algorithms for sampling knowledge from an artificial neural network are given. It is shown that to detect these types of errors it is advisable to use a modified GLARE algorithm with the adaptation procedure. Block diagrams of algorithms for debugging the knowledge base of an intelligent decision support system using the obtained theoretical solutions are presented. The scheme of the organization of testing process on levels of detailing for integration and modular testing is offered. This approach can be used to implement testing processes of Agile methodology, in particular: Agile Modeling, Agile Unified Process, Agile Data Method, Essential Unified Process, Extreme Programming, Feature Driven Development, Getting Real, Open UP, Scrum, Kanban.

Keywords: *automated control systems, decision support, artificial neural network, neural network debugging, knowledge base.*

Introduction

General Problem Statement. Increasing the functional and computational complexity of tasks implemented by software of intelligent systems (IS), continuously leads to an increase in their volume and complexity of development. Accordingly, as the complexity of the software increases, the number of detected and undetected defects and errors of the software increases. This significantly affects not only the quality of IS software, but also in general as a result of solving the tasks assigned to the intelligent decision support system (IDSS).

Significant expansion of the areas of application of IS has allowed to separate critical subject areas. Errors or poor quality of software in these areas can cause damage that far exceeds the positive effect of their use. For such critical areas, IS malfunctions are unacceptable in the event of any input changes, hardware failures or partial failures, and other emergency situations.

In [1] it was proved that the problem of error detection and elimination is exacerbated by increasing the complexity of the tasks solved by the software. It also threatens catastrophes in systems that perform critical functions of managing large and particularly important objects or processes.

This trend has an increasing dynamics over time. This entails the need to develop appropriate algorithms, models and methods. They will provide the required quality of software in general and software of intelligent systems that use the mechanism of neural networks, in particular. This statement is based on the current state of existing approaches to the design of IDSS.

Analysis of the Recent Research and Publications. General principles and approaches to the creation of control systems for complex systems are widely represented in the works of Burkov V, Novikov D., Galin D., Weinberg G., Richard C. Dorf, Bishop R., Dolinina O. [2–4] and others scientists. A number of scientific substantiations of the solution of the problem of debugging knowledge bases are reflected in the works of Suwa H., Scott A., Shortliffe A., Nguen T., and others [5–7].

The most striking, in terms of scientific results, are works based on methods of static debugging without obtaining solutions from the expert system. Unfortunately, the known scientific results have a rather subjective effect on the formalization of structural errors. This approach does not allow the software developer to claim that statically correct knowledge bases will guarantee the required quality of decisions (errors in knowledge caused by a duplicate system of reasoning). The most difficult to detect are errors such as “forgetting about exclusion” and “critical combination of events”. They lead to mistakes in decision-making and are often the cause of great material damage and human casualties [5].

Testing is used to detect errors in knowledge. However, automated testing for knowledge bases also requires additional time and money. They are aimed at formalizing the plan for the implementation of certain rules of a particular subject area, accompanied by the construction of appropriate models.

Another advanced way to create intelligent systems that also require further study is to use a three-layer perceptron as a type of artificial neural network. Therefore, the urgent issue is the development of adaptive methods of establishing knowledge bases that use existing forms of knowledge representation, and thus allow to achieve the required level of reliability of decision-making intelligent systems.

Aim of the Research. The aim of the article is to develop a method of debugging in the neural network environment of intelligent decision support systems. The complexity of solving this problem, according to many experts in the field of systems analysis, is primarily due to the presence in IDSS such properties as inconsistency of knowledge, activity, non-identity, and others.

Main part

In [5] the concept of error of the neural network intelligent decision support system is introduced. A detailed analysis of the applicability of this definition to various subject areas (in particular, to traffic control systems) showed the following. For knowledge bases that support the work of ACS in real time, this definition can be slightly modified, taking into account the requirements for efficiency and accuracy of expert information. Given the above, under the error of the neural network IDSS we understand the fact of erroneous assignment of an event (group of events) $X_i = \{x_1, x_2, ..., x_n\}$ to a class k_k that actually belongs to the class (classes) k_n (classes $k_1, ..., k_n$):

$$P(X_i) = k_k \wedge X_i \in X^K , \qquad (1)$$

where P – the function implemented by the neural network model;

$X_i = \{x_1, x_2, ..., x_n\}$ – the set of events;

$K = \{k_1, ..., k_n\}$ – the set of source classes.

Errors related to defects in the software implementation of neural network IDSS and failures of information processes, and methods for their detection are described in detail in the literature [6]. Therefore, it is assumed that these errors can be formally detected in the stages preceding testing. Errors caused by conflicting opinions of experts and / or limited (imperfect) description of the subject area are subject to more detailed consideration. These errors to some extent have certain features, but in general they are reduced to two classes:

1) errors associated with the incorrect organization of the process of machine learning neural network decision-making mechanism and processing of expert information;

2) errors associated with the peculiarities and inconsistencies of the subject areas. As a rule, this is manifested in the omission or disregard of a number of input factors. An incorrectly organized machine learning process is considered to be a learning process as a result of which the neural network has not reached acceptable classification accuracy or has lost the ability to generalize due to retraining [7–10]. To identify the fact of incorrect organization of the machine learning process, it is proposed to use a mechanism based on the methods described in [7].

Suppose there is a set Z of pairs of input and, respectively, output vectors:

$$Z = \{< X_1, Y_1 >, < X_2, Y_2 >, ..., < X_n, Y_n >\} . \qquad (2)$$

As components of the set Z we define the obligatory minimum number of subsets: educational Z_1, control Z_2 and validation Z_3:

$$Z = Z_1 \cup Z_2 \cup Z_3 . \qquad (3)$$

Based on the definition of the purpose of machine learning – minimization of the objective function of the total quadratic error of the neural network [11], which can be interpreted as the accuracy of classification:

$$E = \frac{1}{2} \sum_j \sum_s (y_j^s - d_j^s)^2 , \qquad (4)$$

where y_j^s – the obtained value of the output neuron j when applying to the network input the value of the observed event s, d_j^s – the expected value of the output neuron j when applying to the network input the value of the observed event s.

Note that in accordance with the accepted error search mechanism, in the process of machine learning, the system error E is checked not only on the subset Z_1, but also on the control subset Z_2. The total error in the training and control subsets takes a constant value and stops the decrease if the current configuration of the artificial neural network cannot learn to solve the problem. That is, the control mark of the transition to neural network retraining is determined by the fact of stabilization of the error value and stopping its reduction in the control subset Z_2. At the same time, it is necessary to pay attention to the constant control of the value of the error in the learning subset, as it can decrease and affect the value of the set Z as a whole. The accuracy of the classification achieved as a result of training and compliance with the requirements of the standards [12] must be verified on a validation subset Z_3 by testing [10]. The analysis of IDSS of the automated air traffic control system showed that the approaches to testing described in [8–10] require additional expansion of the elements of the set Z. To do this, it is proposed to use methods and algorithms for implementing cross-validation [14].

If the accuracy of the classification after training was higher than stated in the terms of reference for the development of IDSS, the neural network is considered successfully trained. But a trained neural network can make mistakes related to the incorrect organization of machine learning neural network mechanism. This is most often manifested in the incorrect classification of events. This situation is determined by the nature of these errors and the statistical similarity of belonging to the wrong class and belonging to the correct class. In a multidimensional feature space, this means that these errors are described by points that are close to the class boundaries:

$$\exists(k_i^*, k_n^* \in K), (|D(X_j, X^{k_j^*}) - D(X_j, X^{k_n^*})| < \varepsilon), \quad (5)$$

where X_j – the input vector of the *j*-th event;

$X^{k_j^*}$ – a subset of events related to the class k_i^*;

$D(X_j, X^{k_j^*})$ – a measure of the proximity of the event vector X_j and the required value of the event $X^{k_j^*}$;

ε – the allowable deviation value determined by the technical conditions and subject area of the IDSS knowledge base.

Depending on the value ε, you can detect all the errors identified by expression (5). In this case, for the events characterized by (5), the neural network system will make a decision, which is that for a particular input vector X_j to determine its belonging to an existing class is impossible. Thus, the correct learning of the artificial neural network, the absence of errors in the implementation of system software and the correct execution of computational processes in general do not indicate the absence of errors in decision-making due to external inconsistencies of the subject area and generate errors such as “forget about exclusion” and / or “ critical combination of events”. For errors of this type, the neural network decision-making mechanism, using an established set of facts $F = \{fact_1, fact_2, ..., fact_n\}$ of an event X_i, makes a conclusion about the belonging of this event to the class $k_i : X_i \in X^{(k_i)}$. This is a correct decision on the formal description of the structural and logical scheme of the knowledge base, but incorrect on the physical essence. This is caused by some combination of facts $F_x = \{fact_{x_1}, fact_{x_2}, ..., fact_{x_n}\}$, which in the disjunction (exclusive combination) uniquely X_i determines non-belonging to the class k_i:

$$fact_{x_1} \wedge fact_{x_2} \wedge, ..., \wedge fact_{x_n} \Rightarrow X_i \notin X^{(k_i)}. \quad (6)$$

The presence of a combination of facts of an exceptional combination (6) is a feature of the subject area. If such a combination of facts is not taken into account at the stage of formalization of the elements of the subject area, the IDSS will give an incorrect result $P(X_i)$, based on the derivation mechanism for the class k_i, based on the assessment obtained on the basis of statistical proximity X_i to the class $X^{(k_i)}$. In [6] the issue of formalization of the error of the decision-making mechanism based on an artificial neural network is considered. Taking into account the results presented in [6] and the judgments given above, we determine that the error of the type “forget about the exception” for the neural network IDSS occurs if:

$$\exists F_x = \{fact_{x_1}, fact_{x_2}, ..., fact_{x_n}\},$$
$$(fact_{x_1} \wedge fact_{x_2} \wedge, ..., \wedge fact_{x_n} \Rightarrow X_i \notin X^{(k_i)}) \wedge \quad (7)$$
$$\wedge P(X_i) = k_i.$$

Note that at the time of deciding on the unambiguous presence of this error, the knowledge base must be statically correct. That is, there are no errors in software implementation and errors associated with failures of information processes. In this case, to detect errors in the classes “forgetting about exclusion” and “critical combination of events” in the training set, it is enough to build a test set and conduct a test procedure of artificial neural network, making logically transparent structure of expertise in the trained neural network. Why is it necessary to extract the decisive rules. The scheme of the process of converting the data of the training set of the artificial neural network ISPR into the appropriate information structure, taking into account the value of the error level is shown in Fig. 1.

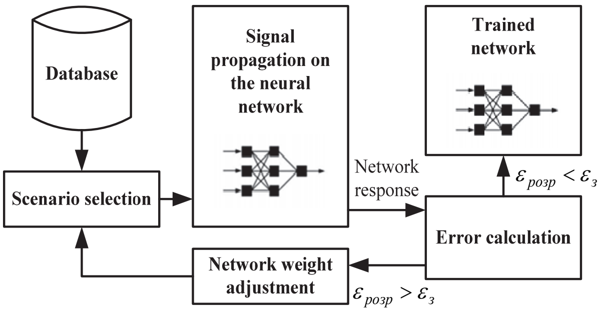
Database
Scenario selection
Signal propagation on the neural network
Trained network
Network response
$\varepsilon_{розр} < \varepsilon_з$
Error calculation
Network weight adjustment
$\varepsilon_{розр} > \varepsilon_з$

Fig. 1. Scheme of the process of converting the data of the training set of artificial neural network IDSS in the appropriate information structure, taking into account the value of the error rate

Taking into account the above and in order to form a logically transparent structure of expert knowledge and their selection at the first stage, it is proposed to apply methods based on neural network contrast algorithms [6; 15] (while increasing the interpretability of the neural network).

In the second stage, it is proposed to carry out direct sampling. In the context of the implementation of the first stage, attention should be paid to solving the problem of contrast during neural network training, which will significantly reduce the likelihood of errors described above.

The use of contrast techniques involves the removal of certain scales and neurons. Their influence on the output is insignificant and they provide expert information in the form of a sparse logically transparent neural network information structure. To implement the network contrast strategy, it is proposed to use methods that can be equally applied to both synapses and inhomogeneous inputs of neurons.

As an example, we can consider the method of adding a penalty term to the target function – the so-called "structural training". The decisions obtained by this method are based on fines for large (in absolute value) weight of the synapse. As the objective function is selected, for example, a function of the form:

$$H = \sum_{i=1}^{n} (y_i - y_i^*)^2 + \varepsilon \sum_{j,k} \left| w_{j,k} \right| . \qquad (8)$$

The criterion of least squares is taken as the first term (y – necessary, y^* – current output signal of a network on the *i*-th example of a training sample).

To simplify the neural network with several output signals in the first term for each example, the values of all output signals of the network are summed. Penalty for the weight of the synapses $w_{j,k}$ (second term) is included with the regulatory weight ε. Its value depends on the balance between the accuracy of the problem and the simplicity of the network structure.

Upon completion of the training, synapses are removed, the weight modulus of which is close to zero (does not exceed a small value σ). In other words, the contrast is performed, in fact, simultaneously with the training. In the second stage it is supposed to use algorithms that allow sampling the decision rules of the form MofN [15–16]:

IF at least / exactly / at least M from N conditions C_1, C_2, ..., C_N are satisfied, THEN ..., OTHERWISE (9)

According to the principle of operation, algorithms for sampling knowledge from the intellectual neural network can be formally classified into the following classes:

– algorithms that implement a direct interpretation of the weights of neurons [17–18];

– algorithms that distinguish ranges of values of input, intermediate and output neurons due to clustering of neural network information structure [19–20];

– algorithms that consider the information structure of the intelligent neural network as a "black box" and its corresponding reactions to changes in input values [21].

The analysis of the selected classes of algorithms for the sampling of knowledge from the intelligent neural network and the approaches implemented by them, showed the following. To detect errors such as "forget about exclusion" and "critical combination of events" it is necessary to use algorithms that allow machine output to obtain a symbolic description of the essence of the subject area. They are semantically similar to the descriptions provided by experts who analyze the same set of input features.

The obtained symbolic descriptions of entities must be directly interpreted in natural language and unambiguously define quantitative and qualitative concepts and characteristics of the subject area. From this point of view, the GLARE algorithm has proven itself well for sampling knowledge. As a result of its application the received information structures have a clear kind and close syntax of the description of rules in natural language formed by the expert.

GLARE does not impose restrictions on the size of the training sample and the number of input and output neurons.

The execution time of the algorithm depends linearly on the above parameters. In addition, this algorithm is characterized by the relative simplicity of software implementation and high speed.

The sequence of stages of debugging of the neural network ISPR is shown in Fig. 2.

If there are several knowledge bases in ISPPR, the testing process, including regression testing, is repeated for each of the knowledge bases of the system separately (Fig. 3).

Algorithm settings allow you to vary the complexity and accuracy of the decision rules in a wide range. The above allows us to conclude that to remove the decisive rules, it is advisable to use the GLARE algorithm as a base, for which adaptive procedures should be used. The system of decision rules obtained by the above steps is essentially a production knowledge base, which in the next step is reduced to a logic circuit that is used to generate a test set and conduct testing.

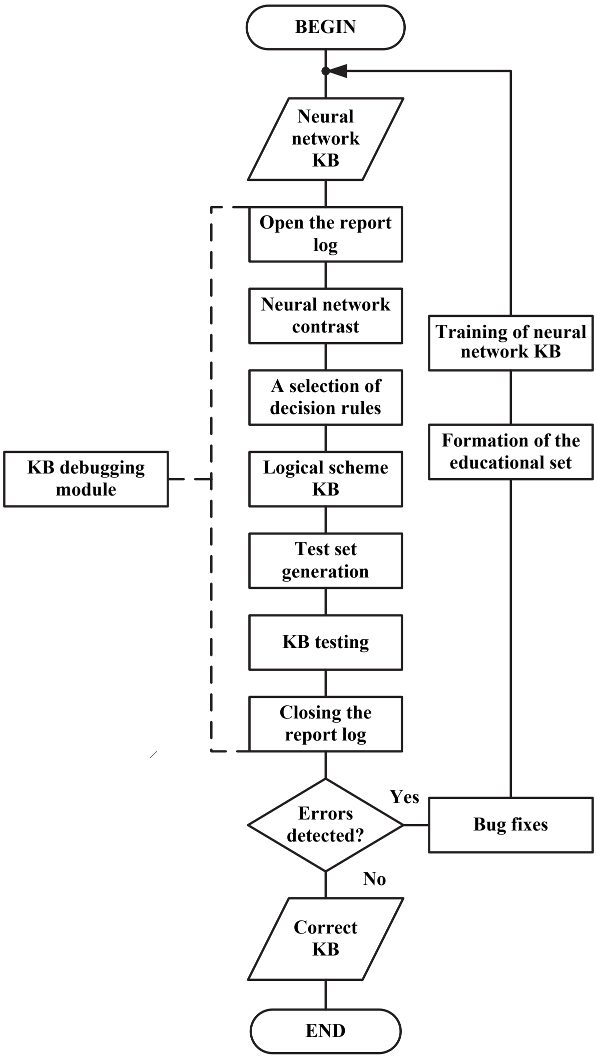

Fig. 2. The sequence of stages of debugging the neural network knowledge base

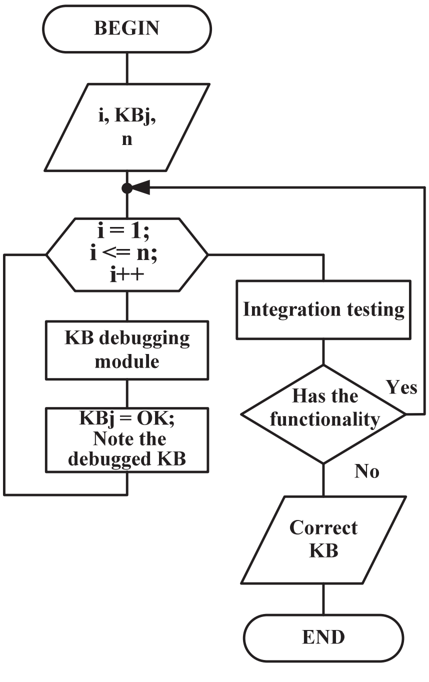

Fig. 3. The sequence of stages of debugging a group of neural network knowledge bases

Testing errors are used to correct the learning set and retrain the ISPPR. After completing the process of testing knowledge bases, the integration testing of the modules of the system of fig. 4. A mandatory element of the testing procedure is to keep a log of the test report and eliminate the identified errors.

The analysis of ways to implement the modular and integration testing approach in the design process of ICPS software showed that the most effective is the use of flexible design methodology Agile, which is adapted to the design process of software ICPS.

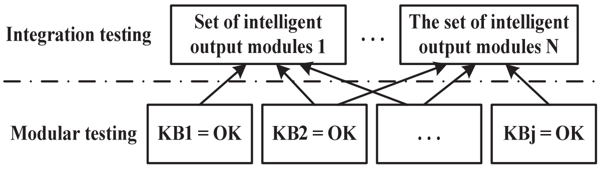
Integration testing
Set of intelligent output modules 1
. . .
The set of intelligent output modules N
Modular testing
KB1 = OK
KB2 = OK
. . .
KBj = OK

Fig. 4. Scheme of presentation of the testing process by the level of detail of the IDSS

This is due to the possibility of obtaining an actual assessment of the level of quality and reliability of IDSS software, respectively, for each backlog sprint. The actual assessment obtained must be verified for compliance with the requirements set by the experts. The scheme of adaptation of the received decisions in Agile methodology is given in fig. 5.

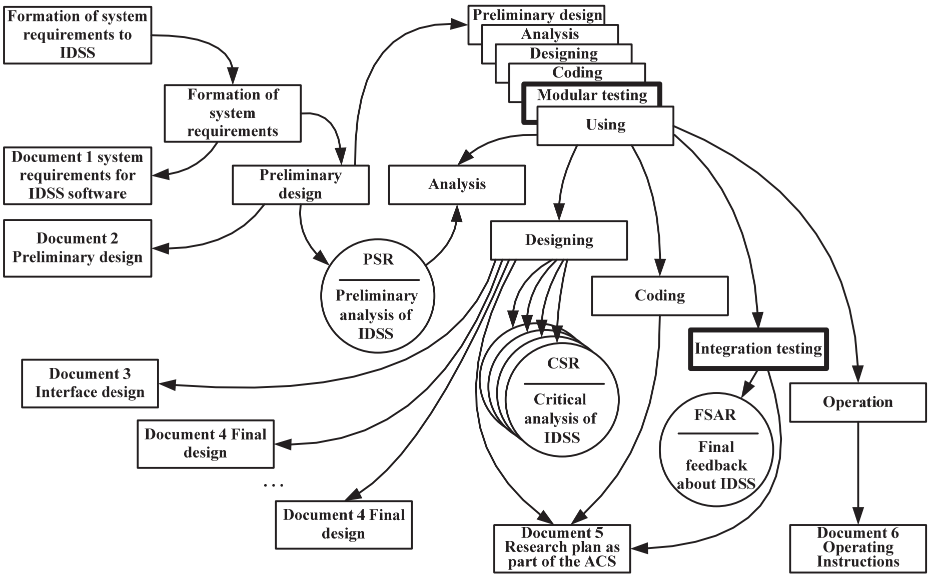
Formation of system requirements to IDSS
Formation of system requirements
Document 1 system requirements for IDSS software
Preliminary design
Document 2 Preliminary design
Preliminary design
Analysis
Designing
Coding
Modular testing
Using
Analysis
PSR
Preliminary analysis of IDSS
Designing
Coding
Integration testing
Operation
CSR
Critical analysis of IDSS
FSAR
Final feedback about IDSS
Document 3 Interface design
Document 4 Final design
. . .
Document 4 Final design
Document 5 Research plan as part of the ACS
Document 6 Operating Instructions

Fig. 5. Scheme of adaptation of the obtained solutions in the Agile methodology

In addition, the use of the proposed approach will allow to assess the compliance of time and financial costs in conventional units (Story Point) for each sprint.

Conclusions

In the course of research it is revealed that for intelligent decision-making systems the presence of the following classes of errors made at expert formation of knowledge bases is possible:

– internal contradiction, redundancy, incompleteness;

– errors in the rules of the knowledge base;

– errors related to the external contradiction of the subject area: “forgetting about exclusions”, “critical combination of events”.

Neural network knowledge bases based on a three-layer perceptron are characterized by an error such as “forget about exclusion”. Due to the balanced structure of the knowledge base after the learning stage, the errors of the class “incompleteness”, “redundancy” and “contradiction”, which are characteristic of the production knowledge base, are absent. The revealed features of errors in ISPPR knowledge bases allowed to offer algorithms of debugging of neural network knowledge bases.

They include: contrast of the neural network, selection of decision rules from the neural network based on the modified GLARE algorithm, which provides for the use of an adaptive procedure; bringing the decision rules to an equivalent logical network, on the basis of which a test set can be formed, followed by a testing procedure.

References

1. Introduction to Theory of Control in Organizations / V.N. Burkov, M. Goubko, N. Korgin, D. Novikov. – Boca Raton: CRC Press, 2015. – 346 p.

2. Galin D. Towards an inclusive model for the cost of software quality / D. Galin // Software quality Professional. – 2004. – № 6(4). – P. 25-31.

3. Weinberg G.M. Quality Software Management / G.M. Weinberg // First-Order Measurement. – NY: Dorset House Publishing, 1993. – 108 p.

4. Dolinina O.N. Method of Test Generation for Debugging of Knowledge Bases for Artificial Intelligence Systems [Electronic resource] / O.N. Dolinina // Software Engineering. – 2011. – С. 40-47. ─ Available at: http://novtex.ru/prin/eng/10.17587/prin._5_2011_6.html.

5. Suwa H. An Approach to Veryfing Consistency and Completeness in a Rule-Based Expert System / H. Suwa, A.C. Scott, A. Shortliffe // Rule-Based Expert Systems. – London: Addison-Wesley, 1984. – P. 159-170.

6. Checking Expert System Knowledge Bases for consistency and completeness / T. Nguen, W. Perkins, T. Laffey, W. Pecora // Proceedings of the 9th Internationally Joint Conference on AI. – Los Angeles, 1985. – P. 375-378.

7. Demidenko S. On-Line Testing In Digital Neural Networks / S. Demidenko, V. Piuri // Proceedings of the Fifth Asian Test Symposium. – Hsinchu, 20-22 November 1996. – P. 295-300. https://doi.org.10.1109/ATS.1996.555174.

8. Уоссермен Ф. Нейрокомпьютерная техника: Теория и практика / Ф. Уоссермен; пер. Ю.А. Зуева, В.А. Точенова. – М.: Мир, 1992. – 184 с.

9. Rumelhart D. Learning Internal Representations by Error Propagation [Electronic resource] / D. Rumelhart, G. Hinton, R. Williams. – Cambridge: MIT Press, 1989. – 362 p. – Available at: https://web.stanford.edu/class/psych209a/ReadingsByDate/02_06/PDPVolIChapter8.pdf

10. Rumelhart D. Learning Representations by Back-propagating Errors / D. Rumelhart, G. Hinton, R. Williams // Nature. – 1986. – № 323. – P. 533-578.

11. Vanwinckelen G. On Estimating Model Accuracy with Repeated Cross-Validation [Electronic resource] / G. Vanwinckelen // Proceedings of the 21-st Belgian-Dutch Conference on Machine Learning. – Ghent, 24-25 May 2012. – P. 39-44. – Available at: https://inlnk.ru/W43oj.

12. The Official site GITHUB.IO. How neural networks are trained. – Available at: https://ml4a.github.io/ml4a/how_neural_networks_are_trained.

13. The problem of structuring indicators of quality of decision software support system / O. Turinskyi, H. Pievtsov, M. Pavlenko, S. Osievskiy, S. Herasimov, V. Djus // International Journal of Advanced Trends in Computer Science and Engineering. – 2020. – № 9(5). – P. 7916-7923. https://doi.org.10.30534/ijatcse/2020/144952020.

14. The Official site CS.CMU. Cross Validation. – Available at: https://www.cs.cmu.edu/ ~schneide/tut5/node42.html.

15. Towell G. Extracting Refined Rules from Knowledge-Based Neural Networks / G. Towell, J. Shavlik // Working Paper. – 2000. – № 91. – 12 p.

16. Towell G. Interpretation of Artificial Neural Networks: Mapping Knowledge-based Neural Networks into Rules / G. Towell, J. Shavlik // Advances in Neural Information Processing Systems. – 1992. – № 4. – P. 977-984.

17. Setiono R. Extraction of Rules from Artificial Neural Networks for Nonlinear regression / R. Setiono, W. Leow, J. Zurada // IEEE Trans. of Neural Networks. – 2002. – № 13. – P. 564-577.

18. Ourston D. Theory refinement combining analytical and empirical methods / D. Ourston, R. Mooney // Artificial Intelligence. – 1994. – № 66. – P. 273-310.

19. Gupta A. Generalized Analytic Rule Extraction for Feedforward Newrak Networks / A. Gupta, S. Park, S. Lam // IEEE Transactions on Knowledge and Data Ingeneering. – 1999. – № 11. – P. 60-68.

20. Setiono R. Symbolic presentation of neural networks / R. Setiono, H. Liu // IEEE Computer. – 1996. – № 29. – P. 71-77.

21. Craven M. Extracting Tree-Structured Representations of Trained Networks / M. Craven, J. Shavlik // Advances in Neural Information Processing Systems. – Cambridge: MIT Press, 1996. – P. 88-93.

Надійшла до редколегії 10.11.2020
Схвалена до друку 05.01.2021

Відомості про авторів:

Тимочко Олександр Іванович
доктор технічних наук професор
професор кафедри
Харківського національного університету
Повітряних Сил ім. І. Кожедуба,
Харків, Україна
https://orcid.org/0000-0002-4154-7876

Осієвський Сергій Валерійович
кандидат технічних наук
доцент кафедри
Харківського національного університету
Повітряних Сил ім. І. Кожедуба,
Харків, Україна,
https://orcid.org/0000-0003-0861-9417

Information about the authors:

Oleksandr Timochko
Doctor of Technical Sciences Professor
Professor of Department
of Ivan Kozhedub Kharkiv National
Air Force University,
Kharkiv, Ukraine
https://orcid.org/0000-0002-4154-7876

Serhii Osievskiy
Candidate of Technical Sciences
Senior Lecturer
of Ivan Kozhedub Kharkiv National
Air Force University,
Kharkiv, Ukraine
https://orcid.org/0000-0003-0861-9417

Тімочко Олександр Олександрович
кандидат технічних наук
інженер фірми “Kreditech Holding SSL GmbH”,
Гамбург, Німеччина
https://orcid.org/0000-0003-0424-0426

Oleksander Timochko
Candidate of Technical Sciences
Expert Quality Assurance Engeneer of Kreditech Holding,
Hamburg, Germany
https://orcid.org/0000-0003-0424-0426

Бєляєв Павло Васильович
науковий співробітник
Харківського національного університету
Повітряних Сил ім. І. Кожедуба,
Харків, Україна
https://orcid.org/0000-0003-0650-6232

Pavlo Bieliaiev
Research Associate
of Ivan Kozhedub Kharkiv
National Air Force University,
Kharkiv, Ukraine
https://orcid.org/0000-0003-0650-6232

МЕТОД УСУНЕННЯ ПОМИЛОК В НЕЙРОМЕРЕЖЕВОМУ СЕРЕДОВИЩІ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ

О.І. Тимочко, С.В. Осієвський, О.О. Тімочко, П.В. Бєляєв

Розглянуті питання виявлення помилок в штучних нейромережах, що пов’язані з суперечливістю думок експертів та/або обмеженістю (недосконалістю) опису предметної області. Проаналізовано підходи щодо їх відлагодження. Показані шляхи вдосконалення існуючих підходів щодо відлагодження помилок типу “забування про виключення”. Показано можливі шляхи застосування отриманих рішень для відлагодження помилок “перетин критичних подій”. Запропоновано формалізоване визначення помилки нейромережевої інтелектуальної системи з урахуванням вимог до оперативності і точності подання інформації. Розглянуті питання впливу некоректної організації машинного навчання на точність класифікації елементів нейромережевої інтелектуальної системи. Доведено можливість застосування методів контрастування мереж на підготовчому етапі до тестування бази знань інтелектуальної системи підтримки прийняття рішень, що дозволяє зменшити ймовірність виникнення помилок даного типу для зазначених систем. Наведено класифікацію алгоритмів вибірки знань з інтелектуальної нейронної мережі, виконано їх аналіз та показано, що для виявлення зазначених типів помилок доцільно використовувати модифікований алгоритм GLARE з процедурою адаптації. Наведено блок-схеми алгоритмів відлагодження БЗ ІСППР, що використовують отримані теоретичні рішення. Запропоновано схему організації процесу тестування за рівнями деталізації для інтеграційного та модульного тестування, що може бути застосована для реалізації процесів тестування Agile-методології, зокрема: Agile Modeling, Agile Unified Process, Agile Data Method, Essential Unified Process, Extreme Programming, Feature Driven Development, Getting Real, Open UP, Scrum, Kanban.

Ключові слова: *автоматизовані системи управління, підтримка прийняття рішення, штучна нейронна мережа, відлагодження нейромереж, база знань.*

МЕТОД УСТРАНЕНИЯ ОШИБОК В НЕЙРОСЕТЕВОЙ СРЕДЕ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ

А.И. Тимочко, С.В. Осиевский, А.А. Тимочко, П.В. Беляев

Рассмотрены вопросы обнаружения ошибок в искусственных нейросетях, связанные с противоречивостью мнений экспертов и / или ограниченностью (несовершенством) описания предметной области. Проанализированы подходы к их отладке. Показаны пути совершенствования существующих подходов к отладке ошибок типа “забывания об исключении”. Показаны возможные пути применения полученных решений для отладки ошибок “пересечение критических событий”. Предложено формализованное определение ошибки нейросетевой интеллектуальной системы с учетом требований к оперативности и точности представления информации. Рассмотрены вопросы влияния некорректной организации машинного обучения на точность классификации элементов нейросетевой интеллектуальной системы. Доказана возможность применения методов контрастирования сетей на подготовительном этапе к тестированию базы знаний интеллектуальной системы поддержки принятия решений, что позволяет уменьшить вероятность возникновения ошибок данного типа для указанных систем. Приведена классификация алгоритмов выборки знаний интеллектуальной нейронной сети, выполнен их анализ и показано, что для выявления указанных типов ошибок целесообразно использовать модифицированный алгоритм GLARE с процедурой адаптации. Приведены блок-схемы алгоритмов отладки БЗ ИСППР, использующих полученные теоретические решения. Предложена схема организации процесса тестирования по уровням детализации для интеграционного и модульного тестирования, может быть применена для реализации процессов тестирования Agile-методологии, в частности: Agile Modeling, Agile Unified Process, Agile Data Method, Essential Unified Process, Extreme Programming, Feature Driven Development, Getting Real, Open UP, Scrum, Kanban.

Ключевые слова: *автоматизированные системы управления, поддержка принятия решения, искусственная нейронная сеть, отладка нейросетей, база знаний.*