

Use of Neural Networks in the Adaptive Testing System

Ekaterina Vitalevna Chumakova¹, Tatiana Alexandrovna Chernova², Yulia Aleksandrovna Belyaeva³
Dmitry Gennadievich Korneev⁴, Mikhail Samuilovich Gasparian⁵
Moscow Aviation Institute, Moscow, Russia^{1, 2, 3}
Plekhanov Russian Economic University, Moscow, Russia^{4, 5}

Abstract—The paper examines the issues of the use of adaptive testing systems in terms of their incorporation in artificial neural network modules designed to solve the problem of choosing the next question, thereby forming an individual testing trajectory. The study presents an analysis of data affecting the quality of problem-solving, proposes a general modular structure of a system, and describes the main data flows at the input of an artificial neural network. The solution proposed for the problem of choosing the difficulty of the question is to use feedforward neural networks. Different architectures and parameters of training artificial neural networks (weight update mechanisms, loss functions, the number of training epochs, batch sizes) are compared. As an alternative, the option of using recurrent long-short term memory networks is considered.

Keywords—Adaptive testing system; artificial neural network; machine learning

I. INTRODUCTION

Adaptive testing is a technology of determining the level of knowledge of the tested subject, in which each upcoming question is automatically selected based on previous answers. The advantage of such testing, as seen by specialists, is the opportunity to determine the testee's level of knowledge more comprehensively and accurately. The problems of developing adaptive tests are topical not only as part of testing students, for example, for the purpose of developing individual learning trajectories, but also in other spheres that require assessment of the subject's competencies and personal intellectual and psychophysiological characteristics. Increasing interest in adaptive testing is demonstrated, for instance, by HR specialists in large companies concerned with recruiting new specialists and testing current employees.

At the heart of adaptive testing systems is an intelligent way to select questions individually for each test subject, based on the answers in all previous steps of the test. The degree of adaptation of the test depends on the number of parameters considered, such as the level of complexity and the number of tasks that are proposed to be completed [1]. Of the greatest interest for research are flexible adaptive systems that allow one to achieve a large variability of tests with high accuracy and reliability in determining the level of training and make the testing process itself look like an oral exam with a teacher. Therefore, the purpose of the study is to organize the process of intellectual choice of a topic (a thematic block of questions) and determine the complexity of the next

question, considering previous answers and the complexity of previously asked questions, as well as the connectivity of topics (blocks) and response time as a factor in guessing or searching for an answer in part of the adaptive testing system.

With the advancement of smart technologies, the development of new methods and the resolution of particular problems using computerized adaptive testing (CAT) is attracting ever-increasing interest from specialists.

At present, we can note three major directions of research, along which CAT methods are being developed:

- Item Response Theory (IRT). IRT is a set of related psychometric theories that serves as a foundation for subject assessment. At the basis of this theory lie mathematical models and logical functions characterizing the relationship between the subject's features (characteristics, knowledge) and the probability of correct answers.
- Bayesian Belief Network (BBN). BBN is a formal graphical language for representing and conveying decision scenarios that require reasoning under uncertainty.
- Artificial Neural Networks (ANNs). ANN is an information processing paradigm based on mechanisms similar to the operation of neurons in the human brain. An ANN is comprised of a certain number of interconnected nodes (neurons) that process information and transmit signals to other neurons based on the results of processing.

It can be stated that the majority of current theoretical and applied research in CAT concerns primarily the use of ANNs. In the field of testing, ANNs are most often proposed to be used as the final module for scoring. In several works [1-3], there were attempts to solve the problem of intellectual choice of a question in the form of determining the level of difficulty of the next question based on one previous one based on the correct or incorrect answer.

Currently, there are many different frameworks for creating ANNs, which makes this mechanism more accessible. However, creating and training a neural network that could offer a substantial advantage over traditional testing requires advanced theoretical knowledge and a fair number of experiments.

At present, researchers do not have a universal approach to creating neural networks for CAT. There arise questions related to the choice of the type and architecture of the network, as well as the number of training examples needed for an acceptable quality of recommendations generated by the network. Contemporary studies only offer general recommendations on these issues. For instance, Golovko and Krasnoproshin [2] suggest quite a large interval of the number of hidden layers.

It is also worth noting the lack of comprehensive works which, having an in-depth look at the entire process of creating a neural network, also describe the applied network learning technologies, which are an important element in ensuring the performance of an ANN.

Approaches to choosing the type of network also vary. Researchers employ both the "classical" feedforward neural networks, in which the signal goes sequentially from layer to layer [3, 4, 5], and recurrent neural networks, in which there is feedback between neurons, and the output signal can be transmitted to the input of the neurons of the previous layer [6]. What should be noted as interesting ideas published in a number of papers is the application of the methods of open systems in the creation of neural networks, in particular, the creation of ANN according to the modular principle [6].

II. METHODS

In the course of the study, the process of conducting a knowledge test in technical disciplines was analyzed from the point of view of its general organization and the identification of indicators that affect the course of testing. As disciplines, such disciplines as "Databases", "Informatics", and "Computer Graphics", read in different educational institutions for students of different courses and specialties, were chosen, and four teachers acted as experts.

At the initial stage, a classification of the tasks solved by the ANN within the process was carried out. Studies were carried out on the use of various types of networks for solving CAT problems, on the basis of which the types of network architectures for further research were determined. The place of the ANN in the overall structure of the adaptive testing system was determined. As a result, a general modular structure of the system was proposed and the main data flows entering the ANN input were described. To achieve the universality of the approach, i.e. regardless of the subject of the test, the choice of the next question was proposed to be carried out in two stages using two ANNs: to select a topic and select the level of difficulty of a question for a particular topic.

At the next stage, to solve the problem of determining the complexity of the question, a feed-forward network was considered. A comparison was made of various ANN architectures and training parameters (weight update algorithms, loss functions, number of training epochs, packet sizes).

In accordance with the results obtained at the previous stage, it was proposed to use six input neurons, which are fed with normalized average values of the correctness of answers to questions and their complexity, the number of questions

already asked, the average time of deviation from the expected response time, the assessment of the answer to the last question asked, and its complexity. At the output of the network there are five output neurons corresponding to the difficulty levels of the questions. Thus, for the simplest feed-forward network with one hidden layer of neurons, the 6-m-5 architecture was chosen and, in accordance with general heuristic recommendations, experiments were carried out for $m = 9, 12, 15, 18, 21$.

All results were obtained using the high-level Keras library, which allows you to quickly start at the initial stages of research and get the first results. SGD, Adam, NAdam and RMSprop implemented in Keras were compared as optimizers to achieve faster convergence. The loss function MSE (mean square error) was used together with the optimizer. Training was carried out on a training set of 1500 sets, which accounted for 80% of the general sample, 10% each were validation and test samples prepared by experts. Traditionally, training was carried out for a large number of epochs (50, 100, 20, 350, and 500), experimentally obtained graphs of accuracy versus the number of epochs for a different number of neurons in the hidden layer to determine the most appropriate architecture. The resulting graphs were constructed using cubic spline interpolation.

In order to determine the effect of the data packet size on the learning process for the 6-12-5 architecture, several experiments were carried out with packets of various sizes (5, 10, 15, 20, 25, 30, 50 and 100) and the optimal size for this task was determined.

Similar training experiments on the same general sample were carried out when switching to network architectures that included two and three hidden layers within 9-21 neurons.

At the final stage, as an alternative, the possibility of using a recurrent ANN LSTM (Long-Short Term Memory) network was considered. In accordance with the feature of this type of network, the number of input neurons was reduced to four, to which the question number, answer score, question complexity, and temporal deviation from the normal were applied. For training, the same tools and approaches were used as in training the feedforward network.

III. MODULAR STRUCTURE OF THE ADAPTIVE TESTING SYSTEM

The main objective solved by an adaptive testing system is the identification of a reliable "profile" of the examinee's knowledge in a particular area. In this case, adaptivity is understood not only as the intellectual selection of questions depending on the level of knowledge demonstrated by the subject, but also the extensibility and universality of the system as a whole [7, 8]. It is thereby clear that such a system has to be constructed by the modular principle, which will ultimately give the structure of the system greater flexibility and versatility.

Of particular interest is the smart selection of the next question. The approaches used in practice differ: assigning the subject an equal number of questions on all topics with different levels of complexity; giving the subject more questions on the topics they made mistakes in; or selecting the

questions using a clear preset algorithm [9-13]. In this case, these methods are difficult to regard as smart approaches.

In this paper, by the subject's profile of knowledge in the given area, we understand the level of mastery over the material in each considered topic. The level of mastery has to be determined in accordance with the difficulty of the assigned questions and the accuracy of the given answers. Therefore, all questions from the bank should not only belong to a certain topic but be characterized by a specific level of complexity.

Obtaining a reliable profile of knowledge through testing presupposes selecting the topic and complexity of the question to be assigned next, meaning that the question is selected for a specific subject in view of the number of questions assigned by topic, their association with one another, level of complexity, and the accuracy of answers given at the previous stages. This task is what an ANN is intended to solve.

The choice of the group of the next question is influenced by the following parameters, which serve as input data for the ANN:

- The topics of the question assigned previously;
- The number of the assigned questions (in each topic);
- The difficulty of the assigned questions (in each topic);
- The accuracy (grading) of the given answers (by the topics considering the levels of complexity);
- The relatedness of the topics to each other;
- Response time to the questions assigned previously.

Let us describe the form in which these data can be stored in the system. It is proposed to store a vector (an array with

the dimensionality equal to the number of questions assigned) of structures for each test taker containing:

- Question – topic, number, difficulty;
- Response time, or rather its positive or negative deviation from the expected norm, i.e. the time sufficient to read the question and give a meaningful answer;
- Grade – the degree of response accuracy (1 – correct, 0 – incorrect).

The relatedness of topics is set via the matrix $M[N, N]$ of coefficients varying in the range from 0 to 1, where 0 – the topics are completely unrelated, 1 – related to the highest degree. The coefficient at the intersection of the i -th row and the j -th column shows how related the i -th and the j -th topics are. The matrix is symmetric, with ones on the main diagonal, so the only really significant input values are $(N^2 - N)/2$.

With a large number of test questions (30-50), the number of input parameters is not only large but also constantly changes as new answers are given (growing in an arithmetic progression). This creates major challenges in determining the architecture of an ANN, as well as complicates the preparation of training and test data sets and the training process itself. In addition, the number of input parameters also changes when so does the number of questions in the test, which eliminates all possible universality.

To resolve this issue, we use two ANNs, one determining the topic of the next question and the other setting the difficulty level. To preserve the number of input parameters and make it constant, we integrate the ANNs with each other by means of algorithmic modules, which perform preliminary mathematical preparation of input values. The resulting general structure of the testing system, which has a hybrid architecture, is presented in Fig. 1.

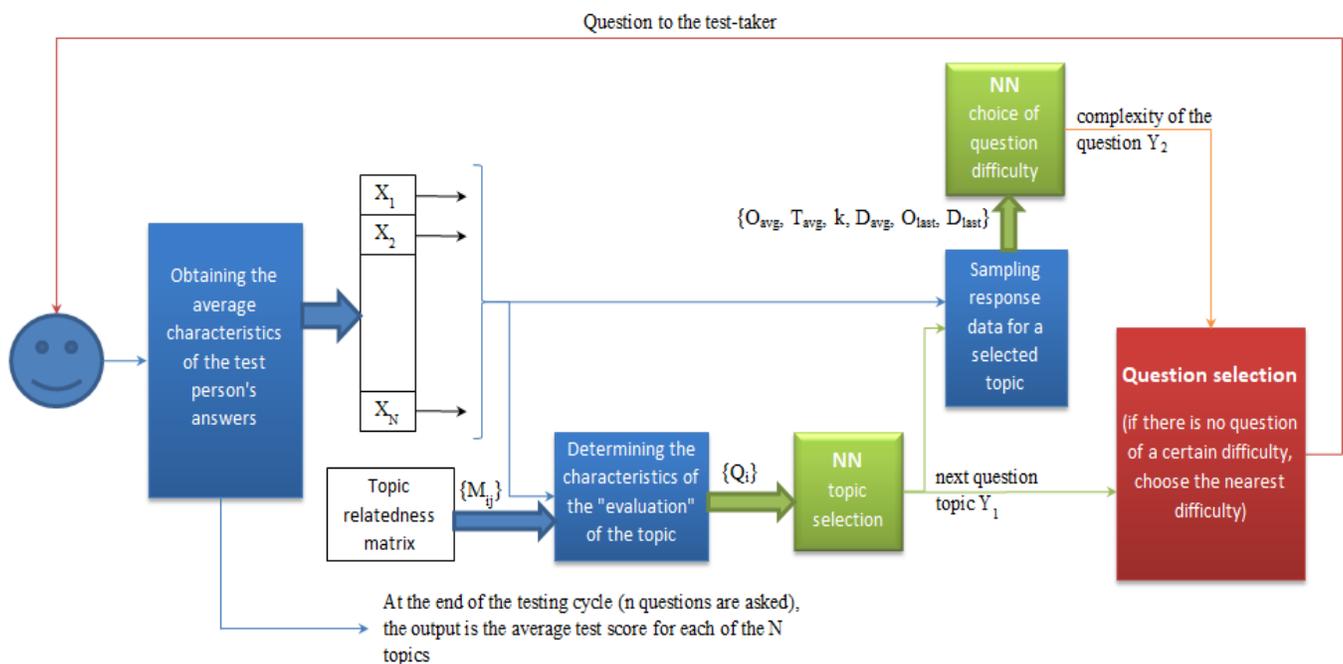


Fig. 1. General Scheme of Functioning of the Adaptive Testing System.

In the case of a feedforward neural network, consideration of all answers given to all of the questions is proposed to be performed by inputting in the networks the average parameters of answers for each topic $\{X_i\}$, for which purpose a respective module is included in the system. In the case of a recurrent ANN, the aforementioned module can be absent due to the capacity of the network itself to account for its previous states. In both cases, the number of inputs with a large number of topics is reduced insignificantly, but stays constant at all stages of testing for any number of questions. The efficiency of using particular types of ANNs is suggested to be assessed in further research.

The choice of the topic in the process of creating the individual testing trajectory needs to take into consideration both the subject's answers and the relatedness of the topics in order to, first, test proficiency in the material across different topics and, second, optimize the total number of questions asked in each topic. At the input of the topic selection ANN, it is proposed to put a vector of "assessment" coefficients of the topics $\{Q_i\}$, which are essentially derived by summing up the share of each answer minding the level of difficulty and relatedness of the topics. Mathematically, this can be expressed as the product of the topic connectivity matrix described earlier and the vector of averaged difficulty-weighted answers.

Once the topic is selected, the second ANN module should determine the difficulty of the upcoming question based on the averaged data for the already specified topic. The logic of decreasing or increasing the difficulty of questions is laid down during training based on the training sets provided by experts according to the given requirements.

As a result, the test taker receives a question selected from the question bank, the difficulty and topic of which are individually selected depending on the test taker's previous answers.

IV. NEURAL NETWORK MODULE FOR DETERMINING THE LEVEL OF DIFFICULTY OF A QUESTION

At the first stage, we design and train an ANN module responsible for selecting the difficulty level. Specification of the difficulty of the question after its topic is already selected largely decreases the number of input parameters that can affect the choice of the question. In this case, it is necessary to consider the parameters of the subject's answers only in one specific topic. For this very reason, this ANN module is preceded by the block of selection of the test taker's performance on a specific subject topic.

In the course of the study, the feasibility of using various ANN models was analyzed from the point of the correspondence of this task to the specific class of tasks solved by particular types of ANNs. This task, however, cannot be unequivocally attributed to a single type since, on the one hand, determining the difficulty of the next question is a classification task, i.e. determining the class of difficulty based on the subject's performance, while on the other, this task involves predicting the real level of knowledge based on the previous answers.

Although there are no specific architectures designed to solve classification tasks, the most commonly used type, in this case, is a multilayer feedforward neural network. For tasks based on sequences, a special type of ANN – a recurrent network – is used. It is impossible to determine in advance exactly which of the architectures is best suited for the task. Therefore, we focus on a more detailed study of two variants of networks, specifically:

- A feedforward neural network, in which all layers are connected with one another directly and sequentially – without feedback or delay lines.
- A recurrent Long-Short Term Memory (LSTM) network, which receives information from the previous passes, thereby being capable of learning long-term dependencies.

The well-known and obvious disadvantage of the latter is their high demands for hardware and resources, both in training (the training process takes a significant amount of time) and in startup.

Next, we will more closely consider the option of using a feedforward neural network. To account for all the previous answers on the topic, the number of which at a certain stage can be random, the input fed to the ANN should include the average values of the accuracy of the given answers and the difficulty of the questions, the number of questions already asked, as well as the average deviation from the expected answer time as a kind of indicator of guessing or searching for answers. In addition to the average values, which do not provide complete information for decision-making in this task even for a human, the network input also includes the mark for the last question answered and its complexity.

Thus, the input layer contains 6 neurons, the output layer – depending on the number of question difficulty levels. In our case, there are 5 neurons, which are aggregated into the last layer containing one neuron. For the final determination of the network architecture, it is necessary to determine the number of hidden layers and the number of neurons in them.

Following the recommendations of Golovko and Krasnoproshin [2], for a network with n-m-p architecture and training sample volume L, the number of neurons in the hidden layer should meet the following condition:

$$\log_2(p) < m < (L - p) / (n + p + 1) \quad (1)$$

Herein, the upper bound is derived from the condition that the training sample size exceeds the number of adjustable parameters. At the same time, there are heuristic rules according to which the size of the training sample must, at least, by an order of magnitude, exceed the number of adjustable parameters to obtain an error of 10%, and the number of hidden layer neurons must, at least, exceed the size of the input by 1.5-2 times [14].

If the use of a perceptron with one hidden layer fails to provide the required accuracy and generalizability of the network, then a neural network with more than one hidden layer is used. The optimal network architecture is also determined via genetic and evolutionary algorithms, which

have their own practical features and limitations. Therefore, at the first stage, we examine a network with architecture n - m - p , for which m and L should satisfy the conditions described earlier.

Since there are no exact methods for estimating the complexity of the problem to be solved and the learning algorithm, which are the determining factors for choosing the amount of data in machine learning, the sufficient amount of data cannot be determined in advance. On the basis of the above recommendations, it is possible to roughly estimate the required size of the general population of raw data and the number of neurons in a hidden layer. We will proceed from the fact that the minimum number of neurons in a hidden layer should be $m=9-12$, suggesting that the volume of the training sample should be, on the one hand, $L>110-150$, and on the other hand, $L\geq 1150-1500$, i.e. an order higher than the number of adjustable parameters for the given m . In the absence of other points of reference, the size of the training sample is set to be 1500 items. According to the generally accepted ratios, the training sample must be 80% of the general population, the verification (validation) sample – 10% (150 items), and the test (control) sample – 10% (150 observations). For the chosen training set, it is advisable to study a number of architectures for $m = 9, 12, 15, 18, 21$ neurons.

The requirements taken into account when preparing the samples are that they should contain a sufficient number of unique examples, should not contain duplicates, contradictions, omissions, and anomalous values, and that the numerical ratio of objects of different classes in each sample should be the same as in the initial general population [15].

In particular, the data structure is affected by the method of network learning. In our case, the “teacher – student” method is employed, as this method is the most commonly used for classification and prediction tasks. Training without a teacher is used in statistical and language models, as well as, for example, in the tasks of clustering and data compression, which does not correspond to the conditions of our task.

Two libraries (frameworks) are considered for the ANN modeling – Keras and PyTorch. These libraries differ in the levels of API and the ways of describing and running network training. Nevertheless, they produced similar results for the described network architecture and training on the same training sets. All the presented results are obtained with the Keras library, which makes it possible to easily create networks and simplify testing of training models, offering additional convenience in the initial stages of research and obtaining first results.

The quality of training of the model is determined not only by its structure and training set but a number of training parameters: the weight update algorithm (optimizer), the loss function, the number of training epochs, and batch size. Below we compare the most popular optimization methods, SGD, Adam, NAdam, and RMSprop, implemented in Keras to achieve faster convergence. The analysis shows that for the problem under study, the best results in terms of accuracy are demonstrated by Adam. The MSE loss function (root-mean-square error) is used for all optimizers.

Traditionally, training is performed for a large number of epochs, which is usually determined experimentally and is sufficient to obtain minimal error and high accuracy [16-18]. In this study, the training of networks with different numbers of hidden layer neurons is performed in the span of 50, 100, 200, 350, and 500 epochs. The results reveal dependencies of accuracy on the number of training epochs for networks with different numbers of neurons in the hidden layer, which are presented graphically below (Fig. 2).

The last noted parameter is the batch size (`batch_size`), i.e. the number of examples in the sample run through the network after which the weight coefficients are updated. Keras implements mini-batch gradient descent with the recommended batch size being 32. Meanwhile, the generalization ability can decline not only when the batch size, which is chosen experimentally, is reduced, but also when it is increased, which is due to the inner noise in the gradient estimation [19]. Several experiments with different batch sizes (5, 10, 15, 20, 25, 30, 50, and 100) are conducted for the 6-12-5 network, resulting in the best generalization ability obtained with a batch size of 50.

The network with the 6-15-5 architecture shows the best accuracy, the obtained learning curves are shown in the graph below (Fig. 3).

An increase of the number of neurons in the hidden layer, the number of hidden layers, and training epochs, as well as mixing the data and change of the learning rate, by means of Keras does not result in an increase in network performance, accuracy rates remain at an average level of 83-85%. The effect of retraining is also not observed. The conclusion from the conducted experiments is that to further improve the accuracy, the general sample of examples needs to be analyzed in terms of the completeness and complexity of the model.

Next, we examine a recurrent network with a similar number of neurons per layer [20, 21]. Proceeding from the fact that an LSTM network processes the temporal sequence of input data while preserving the internal state obtained when processing the previous items, it is not necessary to calculate averaged values to account for all previously received responses. The number of network inputs can be reduced to 4: question number, answer score, question difficulty, and response time deviation from normal. In general, the set of input parameters of the LSTM network will not differ from the previously considered case for the feedforward neural network.

The exception is the form in which training and test sets are presented, each of which, in fact, is a sequence of answers to the questions. In this case, the optimization methods and the functions of activation and evaluation of training results used for the LSTM network will be the same as for the feedforward neural network.

A number of experiments yield similar results (model accuracy of 95% and accuracy on the test sample of 80%) and learning curves, one of which is shown below (Fig. 4). The retraining effect is observed already at 100-150 epochs of training, indicating that the network remembers all the examples and its training requires large samples.

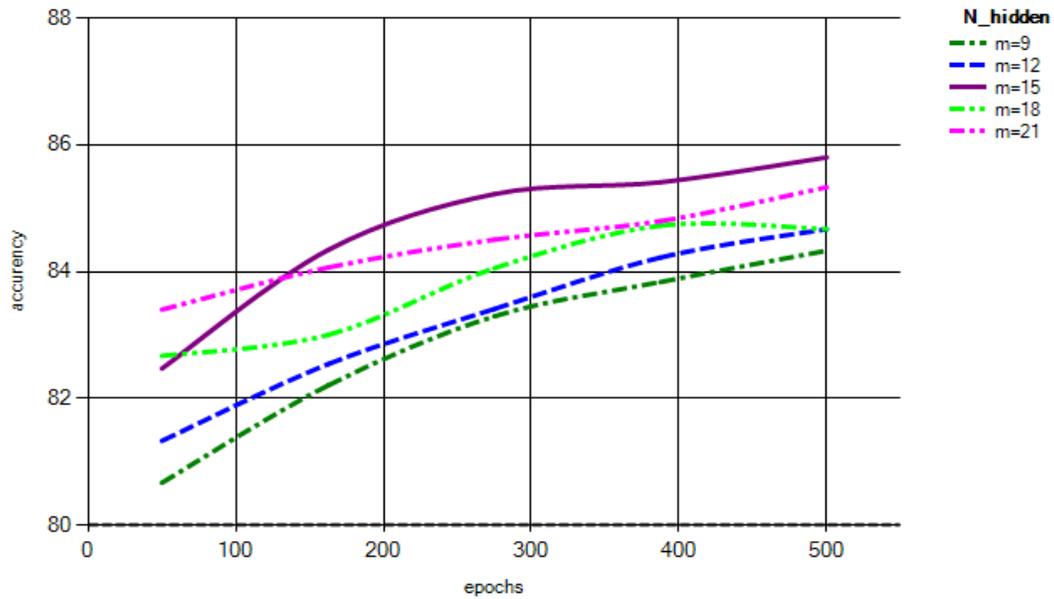


Fig. 2. Dependencies of Accuracy on the Number of Training Epochs for different Numbers of Neurons in the Hidden Layer.

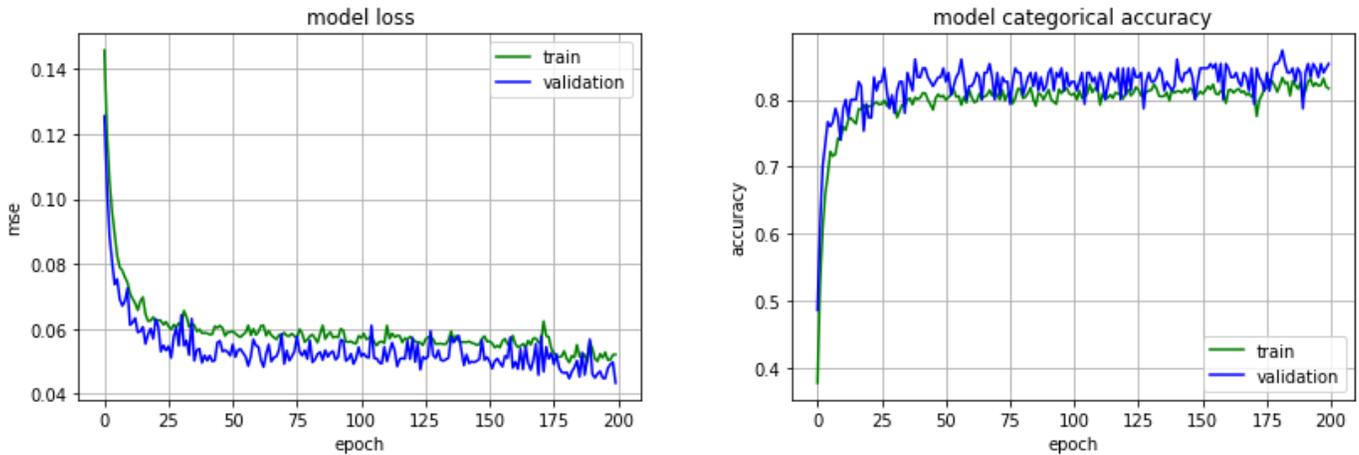


Fig. 3. Learning Curves for Networks with 6-15-5 Architecture.

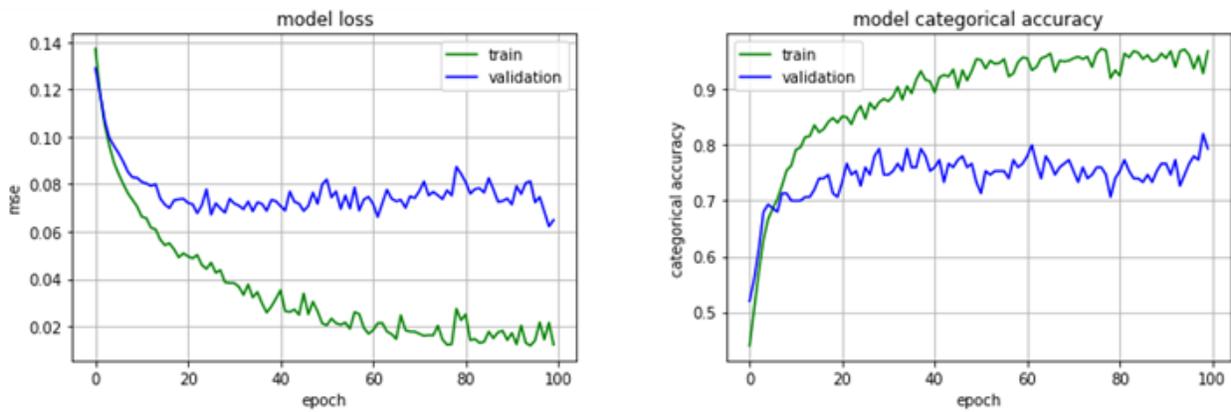


Fig. 4. LSTM Network Learning Curve.

It is not necessary to talk about the advantage of one of the types of networks (forward propagation networks and LSTM networks) for solving the problem of determining the next question in adaptive testing problems. Both types of networks require more detailed preparation of the training set. For direct propagation networks, this is primarily due to the complexity of the model, namely, the nature of the network input data – the averaged values of all previous answers. In this case, to complete the description of the model, more unique and consistent training and test cases are required. The form of the input data of the recurrent network is simpler, but the task of determining the complexity of the next answer based on all the previous ones remains, and, therefore, the training data must represent different testing trajectories at different stages. In addition, the number of adjustable parameters is several times greater than that of the direct distribution network, which also increases the requirements for the volume and quality of the training sample. Preliminary analysis of the training set revealed the presence of gaps in certain data ranges, as well as a certain uneven representation, which, if randomly mixed, can lead to learning failures.

V. DISCUSSION

In general, based on the study, we can conclude that the obtained accuracy of the direct propagation network of 83-85% is quite sufficient for its use in the adaptive testing system. A well-known and obvious disadvantage of using LSTMs is their demands on equipment and resources, both during training (the training process takes a significant time) and during startup, in our case, it is supplemented by increased requirements for the training set and, despite the obtained accuracy of 80%, puts questioning the expediency of further study of LSTM networks in solving this problem.

Using this solution to determine the level of complexity of the next question will consider all the answers given to the subjects, the levels of complexity of the questions, as well as the time of answers within each thematic block, and in the future and considering their thematic connection, in contrast to the previously proposed solutions, in which, only one previous test step is considered, and the accuracy for the recurrent network is 75%.

VI. CONCLUSION

The paper proposes a new structure for organizing an adaptive testing system based on the use of neural network modules, training of the neural network responsible for determining the level of complexity of the question asked is performed. The results obtained can be used in the construction of the second module of the ANN of the system, which is responsible for choosing a topic.

In addition to selecting the topic of the next question, an ANN can be entrusted with the task of moving to the next test stage (assigning the next question), i.e. deriving a reliable knowledge profile with an individual long test trajectory. The study of this possibility is interesting in terms of how optimal the number of given questions will be, whether the system will not go into infinite test mode, or the tests will be too short.

In addition, it is necessary to answer the question of the need to improve the efficiency of an already implemented

network, and, consequently, to conduct research on methods to improve the efficiency of networks, including finer tuning of parameters and learning algorithms, as well as architecture.

In general, the introduction of the proposed tools will allow organizing the process of adaptive testing, with an intelligent selection of questions depending on the demonstrated level of knowledge of the test person to form an individual testing trajectory in order to determine the reliable level of knowledge of the test subject for the optimal number of questions asked.

REFERENCES

- [1] R. P. Chalmers, "Generating Adaptive and Non-Adaptive Test Interfaces for Multidimensional Item Response Theory Applications", *Journal of Statistical Software*, vol. 71(5), 2016, pp. 1–38. <https://doi.org/10.18637/jss.v071.i05>.
- [2] V. A. Golovko, and V. V. Krasnoproshin, "Neural Network Technologies of Data Processing" [Neurosetevye tekhnologii obrabotki dannikh], Minsk: BSU, 2017.
- [3] E. Iu. Savchenko, "Application of modified neural network learning algorithms for adaptive testing tasks" ["Primenenie modifitsirovannykh algoritmov obucheniia neironnykh setei v zadachakh adaptivnogo testirovaniia"], *Nauchnyi aspekt*, no. 4, 2012. <https://na-journal.ru/4-2012-tehnicheskie-nauki/159-primenenie-modifitsirovannykh-algoritmov-obucheniia-neironnykh-setej-v-zadachah-adaptivnogo-testirovaniia>.
- [4] A. Petrovskaya, D. Pavlenko, K. Feofanov, V. Klimov, "Computerization of learning management process as a means of improving the quality of the educational process and student motivation", *Procedia Computer Science*, vol. 169, 2020, pp. 656-661 <https://doi.org/10.1016/j.procs.2020.02.194>.
- [5] A. Grigorev, and V. Mamaev, "On the application of neural networks in knowledge testing" ["O primeneniia neironnykh setei v testirovanii znaniia"], *Nauchnoe Priborostroenie*, vol. 4, no 26, pp. 77–84, 2016.
- [6] S. Sh. M. Jafri, "Computerized adaptive testing using neural networks", 2007. https://www.researchgate.net/publication/228720628_Computerized_adaptive_testing_using_neural_networks.
- [7] O. Iu. Nikiforov, "Use of adaptive computer-based testing systems" ["Ispolzovanie adaptivnykh sistem kompiuternogo testirovaniia"], *Gumanitarnyye nauchnyye issledovaniya*, vol. 4, 2014. <https://human.snauka.ru/2014/04/6274>.
- [8] D. S. Zhadaev, A. A. Kuzmenko, and V. V. Spasennikov, "Peculiarities of neural network analysis of students' training level in the process of adaptive testing of their professional competencies" ["Osobennosti neurosetevogo analiza urovnia podgotovki studentov v protsesse adaptivnogo testirovaniia ikh professionalnykh kompetentsii"], *Vestnik Bryanskogo gosudarstvennogo tekhnicheskogo universiteta*, vol. 2, no. 75, pp. 90-98, 2019.
- [9] V. A. Pesoshin, V. V. Zvezdin, A. N. Iliukhin, R. R. Saubanov, and R. R. Saubanov, "Automated testing system as a tool to improve the quality of knowledge assessment" ["Avtomatizirovannaia sistema testirovaniia kak instrument povysheniia kachestva otsenki znaniia"], *Mashinostroyeniye i komp'yuternyye tekhnologii*, vol. 7, pp. 137–142, 2016.
- [10] J. Rodríguez-Cuadrado, D. Delgado-Gómez, J.C. Laria, S. Rodríguez-Cuadrado, "Merged Tree-CAT: A fast method for building precise computerized adaptive tests based on decision trees", *Expert Systems with Applications*, vol. 143, 1 April 2020, 113066. <https://doi.org/10.1016/j.eswa.2019.113066>.
- [11] D. Pavlenko, L. Barykin, S. Nemeshev, E. Bezverhny, "Individual approach to knowledge control in learning management system", *Procedia Computer Science*, vol. 169, 2020, pp. 259-263. <https://doi.org/10.1016/j.procs.2020.02.162>.
- [12] H. Özyurt, Ö. Özyurt, A. Baki, B. Güven, "Integrating computerized adaptive testing into UZWEBMAT: Implementation of individualized assessment module in an e-learning system", *Expert Systems with Applications*, vol. 39, Issue 10, August 2012, pp. 9837-9847. <https://doi.org/10.1016/j.eswa.2012.02.168>.

- [13] D. A. Pominov, L.S. Kuravsky, P.N. Dumin, G.A. Yuriev, "Adaptive trainer for preparing students for mathematical exams", *International Journal of Advanced Research in Engineering and Technology*, vol 11, Issue 11, November 2020, pp. 260-268.
- [14] S. Haykin, "Neural networks: a complete course" ["Neironnye seti: polnyi kurs"], Moscow: Viliams, 2019.
- [15] X. Xu, T. Liang, J. Zhu, D. Zheng, T. Sun, "Review of classical dimensionality reduction and sample selection methods for large-scale data processing", *Neurocomputing*, vol. 328, 7 February 2019, pp. 5-15 <https://doi.org/10.1016/j.neucom.2018.02.100>.
- [16] K. M. Ang, W.H. Lim, S.S. Tiang, C.K. Ang, E. Natarajan, M.K.A. Ahamed Khan, "Optimal training of feedforward neural networks using teaching-learning-based optimization with modified learning phases", *Lecture Notes in Electrical Engineering*, vol. 770, 2022. pp. 867-887.
- [17] K. Gorshkova, L. Tugashova, V. Zueva, M. Kuznetsova, "Optimizing deep learning methods in neural network architectures", *International Review of Automatic Control*, vol. 14, Issue 2, 2021, pp. 93-101.
- [18] Y. Zhao, "Research on management model based on deep learning", *Complexity*, vol. 2021, 2021, pp. 9997662.
- [19] N. Shirish Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, P. T.P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima", Published as a conference paper at ICLR 2017 <https://arxiv.org/pdf/1609.04836.pdf>.
- [20] Y. Huo, D.F. Wong, L.M. Ni, L.S. Chao, J. Zhang, "Knowledge modeling via contextualized representations for LSTM-based personalized exercise recommendation", *Information Sciences*, vol. 523, June 2020, pp. 266-278. <https://doi.org/10.1016/j.ins.2020.03.014>.
- [21] Md. Kowsher, A. Tahabilder, Md. Z. Sanjid, N. J. Prottasha, Md. S. Uddin, Md A. Hossain, Md. A. K. Jilani, "LSTM-ANN & BiLSTM-ANN: Hybrid deep learning models for enhanced classification accuracy", *Procedia Computer Science*, vol. 193, 2021, pp. 131-140. <https://doi.org/10.1016/j.procs.2021.10.013>.