Hirianskyi, Bohdan; Bulakh, Bogdan

Article

A review of practice of using evolutionary algorithms for neural network synthesis and training

Technology audit and production reserves

Reference: Hirianskyi, Bohdan/Bulakh, Bogdan (2023). A review of practice of using evolutionary algorithms for neural network synthesis and training. In: Technology audit and production reserves 4 (2/72), S. 22 - 26. https://journals.uran.ua/tarp/article/download/286278/280634/661675. doi:10.15587/2706-5448.2023.286278.

6. Yakovchuk, O., Cherneha, A., Zhelezniakov, D., Zaytsev, V. (2020). Methods for Lines and Matrices Segmentation in RNN-based Online Handwriting Mathematical Expression Recognition Systems. *2020 IEEE Third International Conference on Data Stream Mining & Processing (DSMP)*. doi: https://doi.org/10.1109/dsmp47368.2020.9204273
7. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I. (2019) *Language Models are Unsupervised Multitask Learners*. Available at: https://www.semanticscholar.org/paper/Language-Models-are-Unsupervised-Multitask-Learners-Radford-Wu/9405cc0d6169988371b2755e573cc28650d14dfe
8. Child, R., Gray, S., Radford, A., Sutskever, I. (2019). *Generating Long Sequences with Sparse Transformers*. doi: https://doi.org/10.48550/arXiv.1904.10509
9. Vaswani, A., Shazeer, N., Parmar, N. (2017). *Attention Is All You Need*. doi: https://doi.org/10.48550/arXiv.1706.03762
10. Brown, B., Mann, B., Ryder, N., Subbiah, M. (2020). *Language Models are Few-Shot Learners*. Available at: https://arxiv.org/pdf/2005.14165.pdf

✉ ***Oleg Yakovchuk,*** *Assistant, Postgraduate Student, Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, ORCID: https://orcid.org/0000-0002-9842-9790, e-mail: olegyakovchuk@gmail.com*

Maksym Vasin, *Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, ORCID: https://orcid.org/0009-0005-1903-9874*

✉ *Corresponding author*

UDC 004.032.26
DOI: 10.15587/2706-5448.2023.286278

Bohdan Hirianskyi,
Bogdan Bulakh

A REVIEW OF PRACTICE OF USING EVOLUTIONARY ALGORITHMS FOR NEURAL NETWORK SYNTHESIS AND TRAINING

The object of this research is the application of evolutionary algorithms for the synthesis and training of neural networks. The paper aims to select and review the existing experience on using evolutionary algorithms as competitive methods to conventional approaches in neural network training and creation, and to evaluate such existing solutions for further development of this field.

The essence of the obtained results lies in the successful application of genetic algorithms in conjunction with neural networks to optimize parameters, architecture, and weight coefficients of the networks. The genetic algorithms allowed improving the performance and accuracy of neural networks, especially in cases where backpropagation algorithms faced difficulties in finding optimal solutions.

These results can be attributed to the fact that genetic algorithms are efficient methods for global optimization in parameter space. They help avoid local minima and discover more reliable and stable solutions. The obtained findings can be practically utilized to enhance the performance and quality of neural networks in various classification and prediction tasks. The use of genetic algorithms enables the selection of optimal weight coefficients, network connections, and identification of significant features from the dataset. However, they come with the limitation of additional time costs for evaluating the entire population according to the selection criteria.

It is worth noting that the application of genetic algorithms is not a universal method for all tasks, and the algorithm parameters should be individually tuned for each specific problem. Further research could focus on refining the combination methods of genetic algorithms and neural networks, as well as exploring their application in new domains and tasks.

Keywords: *neural networks, evolutionary algorithms, genetic algorithms, hybrid approach, optimization neural network architecture.*

Received date: 03.07.2023
Accepted date: 23.08.2023
Published date: 28.08.2023

How to cite
Hirianskyi, B., Bulakh, B. (2023). A review of practice of using evolutionary algorithms for neural network synthesis and training. Technology Audit and Production Reserves, 4 (2 (72)), 22–26. doi: https://doi.org/10.15587/2706-5448.2023.286278

1. Introduction

The increasing demand for high-precision systems, intelligent applications, and smart technologies in the modern world poses a challenge for researchers and engineers to explore innovative approaches for neural network (NNs) synthesis and training. NNs, as powerful tools for modeling and understanding complex tasks, find diverse applications in various fields, such as medicine, finance, industry, data analysis, and robotics. They have also gained significant

popularity and success in computer vision, natural language processing, adaptive conversational models, audio recognition, data analysis, and autonomous robotic systems.

Efficient neural network training and synthesis are critical for achieving optimal performance and accuracy in problem-solving. Searching for the optimal network structure and weight parameters remains a significant challenge due to NNs' structural complexity, numerous parameters, and high demands on computational resources and time.

The creation of NN is associated with several problems that do not have universal solutions. Some difficulties are [1–4]:

– Overfitting – NN perform exceptionally well on the training data but fails to generalize effectively to new, unseen data.
– Underfitting – inability to capture the underlying patterns and complexities present in the training data.
– Vanishing gradients – the gradients of the loss function become extremely small during the backpropagation process.
– Choosing the type of network, its dimensions, and the number of parameters.
– Computational complexity – as the number of parameters and connections in the network grows, computations require significant resources and time.
– Adapting the model to changed training data requires complete retraining rather than adjusting existing weights.

In this context, evolutionary algorithms (EAs) are becoming increasingly attractive for the synthesis and training of neural networks. The use of these algorithms enables optimization and search for solutions in a vast parameter space, which is particularly relevant for NNs with complex structures and a large number of layers and neurons. Based on natural principles of selection, mutation, and inheritance, EAs efficiently explore, discover, and improve potential solutions.

This article aims to comprehensively investigate and comprehend the practice of using EAs in the synthesis and training of NNs, evaluating their effectiveness, stability, implementation challenges, and added value compared to other methods.

The practical side of the research is to determine the effectiveness of the combination of these groups of algorithms for solving the problems of machine learning and artificial intelligence to accelerate the creation of models and increase the accuracy of their work.

The relevance of using EAs for the synthesis and training of NNs lies in their potential to provide global search in the model's parameter space, helping to find more accurate and efficient architectures for specific tasks. Inspired by natural processes, EAs can effectively optimize neural networks, avoid local minima difficulties, and achieve high model robustness.

2. Materials and Methods

The object of research is the use of evolutionary algorithms in the context of synthesis and learning of neural networks. It is focused on the analysis and evaluation of the application of these algorithms for the optimization and search of optimal weight coefficients, structure of the NNs, and their ability to solve tasks with high accuracy. The description of the object of research includes various aspects and options of combining EAs in the learning of NNs, as well as their advantages and limitations.

During the research process, scientific materials obtained from open sources were utilized for writing the article. These sources encompass publications ranging from the 1970s to the present day, including scientific articles, conference papers, journal publications, and academic research, focusing on the application of evolutionary algorithms in the context of neural network training and artificial intelligence. Analyzing such sources provided a well-founded basis for analysis and drawing conclusions.

2.1. Artificial Neural Networks. Artificial Neural Networks (ANNs) are machine learning models built based on the organization of neurons in the brains of living organisms [5]. ANNs consist of interconnected nodes, or artificial neurons. Each neuron receives signals from other neurons, processes them, and can transmit signals to other neurons it is connected to. The connections between neurons have weights that change during the learning process. Neurons can have thresholds, so a signal is sent only if the aggregated signal crosses this threshold. Typically, neurons are organized into layers, and different layers may perform different transformations on their inputs. Signals of artificial neurons propagate from the input layer to the output layer, possibly passing through multiple layers multiple times.

ANNs are trained by processing examples with known inputs and outputs, forming weighted associations between neurons, which are stored in the network. The training of ANNs from an example typically involves determining the difference between the network's output and the target output, which is referred to as the error. The network adjusts its weights according to the learning rule and the magnitude of the error. Sequential corrections compel the network to produce an output that increasingly resembles the target output. After a sufficient number of such training corrections, the learning process can be completed based on specific criteria.

2.2. Evolutionary algorithms. Evolutionary algorithms are a subset of computational methods that simulate the processes of biological evolution to solve optimization problems [6]. They utilize mechanisms such as reproduction, mutation, recombination, and selection. With this approach, candidate solutions to the optimization problem are randomly generated and interact as individuals within a large population.

In general, the EAs steps can be summarized as follows:

1. *Initialization*: Generate an initial population of solutions (chromosomes). Evaluate the fitness of each solution.
2. While the stopping criteria are not met:
2.1. *Selection*: Select parents for reproduction using a selection probability based on their fitness.
2.2. *Crossover*: Combine the genetic information of parents to create new solutions (offspring).
2.3. *Mutation*: Randomly alter some genes in the solutions with a low probability.
2.4. *Fitness Evaluation*: Evaluate the fitness of the newly created population.
2.5. *Replacement*: Replace the old population with the newly created.
3. After completing the main EA loop, select the best individual from the final population based on their fitness values.

At their core, ANNs and EAs differ both in the tasks they are used for and the methods they are based on. Neural networks are models for decision-making with inputs and outputs. On the other hand, genetic algorithms (GAs) progressively improve existing solutions over time. In theory,

ISSN 2664-9969

this allows for the exploration and evolution of connections in NNs, weight optimization, and feature selection from data using GAs.

3. Results and Discussion

3.1. Results. Careful planning at each stage is necessary for developing NNs models for specific applications. First, the domain of interest should be selected based on theoretical, empirical, or applied interests. Afterward, careful consideration of the network architecture is crucial, determining the structure of the network, the number of layers, neurons, and connections between them.

Network design is a challenge, especially considering the diversity and complexity of options. Finding the optimal balance among various performance criteria, including learning speed, compactness, and generalization ability, is crucial.

Therefore, an automated approach based on genetic algorithms can open new perspectives in the research and development of neural networks, particularly in terms of their design and training.

There are several assumptions. The first one that the application of GAs can lead to unique network architecture with inaccessible connections for conventional algorithms. The second, that EAs can effectively solve the problem of training NNs associated with local optima and gradient vanishing. The third indicates that by combining evolutionary algorithms to search for weight initialization values with conventional methods, faster convergence of the algorithm can be achieved.

The potential use of Genetic Algorithms for constructing NNs architectures was evaluated as early as 1989 [7]. They described the neural network as a string of bits, where each bit indicated the presence of a connection between neurons, forming the genotype population processed by the genetic algorithm (GA). A simple crossover was utilized, randomly selecting a segment that exchanged values between parents. The backpropagation method was employed for weight coefficient search. The primary criterion for success was the model's ability to learn. The authors tested the effectiveness on three tasks: using a neural network to predict the outputs of XOR logical gates with two binary inputs, the four-quadrant problem, and the copying pattern task [7].

Initially, they obtained results that indicated the need for much iteration to solve a simple problem. Specifically, it took 10 iterations and 500 evaluations to construct a network capable of representing the properties of the XOR function, although the fitness value reached 0.75 already in the first iteration. During the research, they obtained unexpected architectural solutions for networks with high accuracy. In the second task, it took approximately 200 epochs to achieve the desired accuracy, but they found that the network architecture was novel compared to previous proposals, having fewer layers but additional connections. For the copying task, the authors used a 10-bit string that the model had to duplicate during the learning process. On average, the researchers achieved convergence of the algorithm in 6 epochs [7].

The study demonstrates that the utilization of GAs can be applied for establishing connections within a network. Such an approach can generate more unique architectures, but depending on the task, it may require a significant number of iterations. The authors emphasize the importance of the crossover function, which influences the changes between generations, and suggest conducting further experiments in this direction [7].

In the study [8] aimed to train NNs to recognize handwritten digits and enhance accuracy by applying a genetic algorithm to search for adjusting coefficients. The author chose a hybrid approach that combines neural networks with a genetic algorithm. This method is used to determine optimal weight parameters, which are multiplied by the outputs of the networks as coefficients. The training process occurred in two stages. Initially, three different NNs were trained, each using its own features for image recognition and achieving accuracy up to 90 % within 1000 iterations. Subsequently, the GA was applied to find the optimal parameters for combining these networks. The GA achieved an accuracy of 97.9 %, the method of averaging achieved 97.15 %, and the Borda method achieved 96.7 %. The author also compared their results with solutions from other researchers, where the highest correct response rate was 97.10 %. This approach demonstrated higher accuracy compared to the arithmetic mean and Borda methods, with faster and more stable convergence using a one-percent mutation rate as opposed to a five-percent [8].

The task of extracting the most significant features from a set of data requires time-consuming retraining of classifiers on new training datasets, but results in a more compact model that demands less computational power. To solve this problem, it was used [9] genetic algorithms to identify the essential significant features, reducing the initial 155 features to 33 for the k-nearest neighbors classifier. In another study [10], was compared multiple data classification methods for electromyogram data. The algorithms included feedforward NNs, the K-means method, the Kohonen neural network, and GA with a classification system.

Since the task was found to be linearly inseparable, the *k*-means algorithm showed poor results. After evaluating the accuracy of the algorithms, the authors proposed to combine them all into a single system and provide the user with the values from each classifier [10]. This approach aimed to compensate for the drawbacks of each individual model by utilizing a combination of different methods.

More than 30 different combinations of cases on using Genetic Algorithms with NNs were investigated in [11]. It is noted that genetic algorithms are less effective to the best gradient algorithms in most cases methods, and in those cases when the genetic algorithm was proven to be competitive, it required more time. However, GAs can be valuable where gradient methods are not directly applicable and obtaining gradient information is challenging. Some researchers utilize GAs for weight initialization and training optimization. The findings from their work suggest that using genetic algorithms for large networks requires computational optimization and parallel implementation to accelerate the learning process [11].

The study [12] compared eight combinations of evolutionary algorithms and neural methods applied to 15 classification problems. These problems included both binary and multiclass datasets with feature sizes ranging from 4 to 60. The experiments involved the use of feedforward neural networks with a single hidden layer. The accuracy was evaluated using a 2-fold cross-validation with five iterations. Neural networks were trained using both binary and real-valued encoding in combination with GAs. The GA was responsible for generating the initial values, and the weights were further adjusted using the backpropagation method with varying numbers of epochs. To assess the models, 30 % of the dataset was used. As a result, most combinations of GAs and

NNs demonstrated similar good performance, with accuracy comparable to the backpropagation method. Specifically, the methods that utilized GA for designing network structures had a weaker impact on the results compared to using GA for training the networks. The networks trained using simple backpropagation and simple binary GA demonstrated competitive performance compared to more complex methods. Authors of [12] see potential in exploring new combinations of GA and NN, such as ensembles and methods that simultaneously change both weights and connections between neurons. However, they emphasize the importance of thoroughly testing all experiments on various datasets and using different methods to obtain the most objective and reliable results.

In [13], it was noted that there are no universal parameters for genetic algorithms, such as crossover probability, mutation rate, and population size that would be suitable for all tasks. Typically, these parameters were determined through trial and error. The success of the genetic algorithm heavily relied on the initial architecture of the neural network – the initial population. Many research studies focus on feedforward neural networks due to their recognition capabilities, effectiveness in image recognition compared to other networks, and simplicity [13, 14].

In [15], it is mentioned that the biogeography-based optimization algorithm is the most suitable for optimizing multilayer perceptrons. It is compared this algorithm with five other optimization algorithms: swarm optimization, genetic algorithm, ant colony, population-based incremental learning, evolutionary strategy, and backpropagation. Six datasets were used to evaluate the effectiveness of these algorithms on different functions: sigmoid, cosine, sine, sphere, Griewank, and Rosenbrock. The study also observed the productivity of this algorithm and the achieved accuracy of the models. Additionally, the author conducted research on the efficiency of using the BBO algorithm for training radial basis functions. The experiment addressed problems from 12 well-known datasets and compared them with 11 other widely-used methods in the literature, including gradient descent, evolutionary, and swarm algorithms. The effectiveness was evaluated using a statistical test, confirming the efficiency of the proposed algorithm. The author highlights several main advantages of the algorithm, including its speed, high accuracy, capability to avoid local optima, efficiency in training networks for classification tasks, and its ability to handle networks with varying numbers of neurons [15].

Another research paper [16] investigated the effectiveness of Genetic Algorithms and Neural Networks on a real stock market forecasting problem. It is utilized Backpropagation Neural Network (BPNN) and a combination of Neural Network with Genetic Algorithm (NNGA) as intelligent techniques for building prediction models. The model validation was performed on highly volatile data. NNGA demonstrated more reliable forecasts compared to BPNN for almost all datasets. The evaluation metrics used to assess the results included Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), and Root Mean Squared Error (RMSE). In this problem, BPNN struggled to determine the weight coefficients leading to a high percentage of correct responses, while NNGA achieved a prediction accuracy of over 97 % for the 5-day forecast horizon on both datasets. As the forecasting horizon transitioned from short-term to long-term, the performance of both models decreased, but NNGA consistently made fewer mistakes [16].

3.2. Discussion. The main intent of this study was to collect, structurize and generalize the existing state-of-the-art knowledge of different means to apply GA to NN synthesis and usage, and based on that to find the most perspective cases of such a specific combination. Overall, similar conclusions are observed in all analyzed works: GA algorithms can be utilized for the data preprocessing, training, and network architecture modification stages. The convergence of the genetic algorithm cannot be evaluated until the research is conducted. Upon reaching a certain threshold of accuracy, the genetic algorithm can be halted, and the model can be further fine-tuned using conventional methods. Thus, the advantages of both approaches are combined. In cases where conventional algorithms fail, GA may create and train a neural network, but it is likely to require more time and resources.

The practical significance of this work lies in elucidating the possibilities and limitations of using evolutionary algorithms for training and constructing neural networks. The results of this study provide crucial insights for researchers and engineers working in the field of machine learning and artificial intelligence.

Key successful combinations of GA and NN that can be applied in practice, along with their main pros and cons, are given in Table 1.

Table 1

Variants of combining genetic algorithms and neural networks

No.	Method of combination	Advantages	Disadvantages
1	GA for feature selection	Automation of feature selection from raw data based on their impact on the outcome	High computational costs
2	Initialization and optimization of weight coefficients	Reduced dependency on initial conditions, search for values across the parameter space	Time-consuming evaluation of training accuracy
3	Initialization and modification of neuron connections	Improvement of network structure, optimization of computations	Difficulty in scaling to large networks
4	Parameter tuning for standard algorithms: step size, momentum	Automation of learning rule selection	Dependency on the initially initialized network

Limitations of the study: For the swift and facile application of GA to NN, the lack of well-established libraries readily available for use is evident. Currently, in most cases, the implementation of algorithms needs to be done manually for each specific task. Additionally, results may vary across different studies due to dependencies on the initial population, its creation method, crossover, and other parameters. Tasks involving high-dimensional feature data and complex network architectures demand significant resources. One potential solution lies in employing parallel computing, clusters, or cloud computing. The ultimate implementation directly hinges upon the researcher's goals, skills, and available resources.

Impact of military conditions: This article mostly dealt with the existing expertise in the field of application of GA to NN, and the impact of the martial law conditions on the authors' research was minimal. However, in order to continue this investigation with practical numerical experiments (see further research directions below), a robust network connection must be present and the computing resources must be available, which can potentially be an issue due to the war currently ongoing. It must be also noted that the current

ISSN 2664-9969

study showed that AI based on NN can benefit from the using of GA in many ways, which can contribute both to civil applications and the defense industry.

Prospects for further research: This study can be a starting point for the next research steps including, but not restricted to: the development of new algorithms specialized for controlling neural network training, optimization of GA for handling vast datasets, creation of hybrid systems integrating GA and conventional algorithms, investigation into scalability and computation optimization.

4. Conclusions

While analyzing and researching the application of evolutionary algorithms for training and synthesizing neural networks, it is identified that it can be profitable across a wide spectrum of tasks. The most prevalent scenarios for employing Genetic Algorithms (GAs) with Neural Networks (NNs) involve addressing issues related to algorithms getting stuck in local optima, gradient Vanishing, optimizing computations through network connection simplification, and selecting critical parameters from a set of values while combining them with already established algorithms through the initialization of initial parameters and improvement through conventional methods. This means that the combination of GAs and NNs can be successfully applied to solve problems in image classification, prediction, optimization, robotics, medicine, finance, and various other fields.

However, this study revealed several critical issues to be considered. Genetic algorithms can solve many problems as they search for the best option in the solution space by going through a large number of combinations, and this is both an advantage and a disadvantage of this method. In some problems, algorithm convergence can be rapid, while in others, it may demand significant time and resources. This characteristic restricts the scalability of this approach, particularly for tasks involving a large number of hyperparameters and neurons. There exist no universal parameters for GAs, and they must be tailored individually for each specific task and network. To determine optimal combinations of GAs and neural networks and assess their effectiveness, rigorous research involving various testing methods and diverse datasets is imperative.

Summarizing all of the above, the obtained results indicate the potential of GAs in the field of neural network training and their ability to enhance the performance and quality of neural networks. The application of GAs represents a promising approach for optimizing the parameters and structure of neural networks. However, it is essential to note that genetic algorithms are not a universally applicable method for all tasks. Their effectiveness may vary depending on the specific problem, and further research and experimentation are necessary to determine their suitability for different scenarios.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this study, including financial, personal, authorship, or any other, that could affect the study and its results presented in this article.

Financing

The study was conducted without financial support.

Data availability

The study has no associated data.

References

1. Hawkins, D. M. (2003). The Problem of Overfitting. *Journal of Chemical Information and Computer Sciences, 44 (1),* 1–12. doi: https://doi.org/10.1021/ci0342472
2. Geman, S., Bienenstock, E., Doursat, R. (1992). Neural Networks and the Bias/Variance Dilemma. *Neural Computation, 4 (1),* 1–58. doi: https://doi.org/10.1162/neco.1992.4.1.1
3. Hanin, B. (2018). Which neural net architectures give rise to exploding and vanishing gradients? *Advances in neural information processing systems,* 31.
4. Hu, X., Chu, L., Pei, J., Liu, W., Bian, J. (2021). Model complexity of deep learning: a survey. *Knowledge and Information Systems, 63 (10),* 2585–2619. doi: https://doi.org/10.1007/s10115-021-01605-0
5. Basic Learning Principles of Artificial Neural Networks (2007). *Foreign-Exchange-Rate Forecasting With Artificial Neural Networks,* 27–37. doi: https://doi.org/10.1007/978-0-387-71720-3_2
6. Katoch, S., Chauhan, S. S., Kumar, V. (2020). A review on genetic algorithm: past, present, and future. *Multimedia Tools and Applications, 80 (5),* 8091–8126. doi: https://doi.org/10.1007/s11042-020-10139-6
7. Miller, G. F., Todd, P. M., Hegde, S. U. (1989). Designing neural networks using genetic algorithms. *Proceedings of the third international conference on Genetic algorithms.* San Francisco: Morgan Kaufmann Publishers Inc., 379–384.
8. Cho, S.-B. (1999). Pattern recognition with neural networks combined by genetic algorithm. *Fuzzy Sets and Systems, 103 (2),* 339–347. doi: https://doi.org/10.1016/s0165-0114(98)00232-2
9. Chang, E. I., Lippmann, R. (1990). *Using Genetic Algorithms to Improve Pattern Classification Performance.* NIPS.
10. Schizas, C. N., Pattichis, C. S., Middleton, L. T. (1992). Neural networks, genetic algorithms and the K-means algorithm: in search of data classification. *Proceedings COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks.* Baltimore, 201–222. doi: https://doi.org/10.1109/cogann.1992.273938
11. Schaffer, J. D., Whitley, D., Eshelman, L. J. (1992). Combinations of genetic algorithms and neural networks: a survey of the state of the art. *Proceedings COGANN-92: International Workshop on Combinations of Genetic Algorithms and Neural Networks.* doi: https://doi.org/10.1109/cogann.1992.273950
12. Cantu-Paz, E., Kamath, C. (2005). An Empirical Comparison of Combinations of Evolutionary Algorithms and Neural Networks for Classification Problems. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics), 35 (5),* 915–927. doi: https://doi.org/10.1109/tsmcb.2005.847740
13. Chiroma, H., Noor, A. S. M., Abdulkareem, S., Abubakar, A. I., Hermawan, A., Qin, H., Hamza, M. F., Herawan, T. (2017). Neural Networks Optimization through Genetic Algorithm Searches: A Review. *Applied Mathematics & Information Sciences, 11 (6),* 1543–1564. doi: https://doi.org/10.18576/amis/110602
14. Bishop, C. M. (1995). Neural networks for pattern recognition.
15. Mirjalili, S. (2019). Evolutionary Algorithms and Neural Networks. *Studies in Computational Intelligence.* doi: https://doi.org/10.1007/978-3-319-93025-1
16. Sharma, D. K., Hota, H. S., Brown, K., Handa, R. (2021). Integration of genetic algorithm with artificial neural network for stock market forecasting. *International Journal of System Assurance Engineering and Management, 13 (S2),* 828–841. doi: https://doi.org/10.1007/s13198-021-01209-5

✉***Bohdan Hirianskyi,*** *Postgraduate Student, Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, ORCID: https://orcid.org/0009-0000-6580-7268, e-mail: giryanskbogdan@gmail.com*

Bogdan Bulakh, *PhD, Associate Professor, Department of System Design, National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Kyiv, Ukraine, ORCID: https://orcid.org/0000-0001-5880-6101*

✉*Corresponding author*