UDC 004.855.5(045)
DOI:10.18372/1990-5548.85.20428

[1]**V. M. Sineglazov,**
[2]**P. A. Chynnyk**

RE-UPLOADING DATA IN TENSOR NETWORK

[1]Aviation Computer-Integrated Complexes Department, Faculty of Air Navigation Electronics and Telecommunications, State University "Kyiv Aviation Institute", Kyiv, Ukraine
[2]Institue for Applied System Analysis, National Technical University of Ukraine "Ihor Sikorsky Kyiv Polytechnic Institute," Kyiv, Ukraine
E-mails: [1]svm@nau.edu.ua ORCID 0000-0002-3297-9060,
[2]chynnyk@vivaldi.net ORCID 0009-0002-6787-5429

Abstract—*In this paper, we present an approach for enhancing quantum tensor networks through the method of data re-uploading. The proposed framework integrates multiple layers of classical data encoding into tensor network architectures, thereby improving their approximation capacity and reducing the impact of barren plateaus in training. The model construction relies on tree tensor networks combined with RX, RZ, and RY rotational gates and CNOT entanglement, while optimization is performed using differential evolution as a gradient-free algorithm. Experimental evaluation was carried out on the iris and wine datasets, comparing baseline tensor networks with architectures incorporating one to three re-uploading layers. The results demonstrate a consistent reduction in training and test loss, with accuracy, recall, and precision reaching 100% on the iris dataset for three layers and improvements of up to 40% in prediction quality on the wine dataset. These findings confirm that data re-uploading significantly enhances the performance and expressiveness of tensor network-based quantum models.*

Keywords—Machine learning; quantum computing; quantum machine learning; re-uploading; tensor network; barren plateaus; differential evolution; quantum neural network.

I. INTRODUCTION

Today, one of the strongest challenges in quantum machine learning is the barren plateau phenomenon. Many researchers are struggling to overcome it. One of the recent reviews of this problem is given in the article [1]. The authors of the study spent many years to comprehensively approach the study of this phenomenon. Based on this article, the main problems of barren plateaus are not noise-resistant quantum computers, high dimensionality of the scheme, many quantum gates that would turn the quantum scheme into a regular random number generator [2].

During our research, we also encountered the problem of barren plateaus. We tried many architectures, but we could not overcome the error limit. Based on the research of the authors of the articles [1], [3], we decided to use the method of re-uploading data for tensor networks. Tensor networks were chosen because they best demonstrate the quality of training a quantum artificial intelligence model and the result of this training [4].

The idea behind data reloading is to introduce multiple layers of encoding for classical input data throughout the depth of the circuit.

This method increases the expressiveness of a quantum model even with a small number of qubits. Unlike the traditional model of encoding classical data only once, reloading inserts data multiple times at different points in the quantum chain, increasing the capacity of the model and helping to avoid expressiveness bottlenecks.

In our work, we propose to investigate data reloading in quantum tensor networks. Tensor networks offer an efficient framework for representing quantum systems and are known for their scalability and ability to reflect entanglement patterns. We aim to assess whether integrating reloading into tensor network-based models can reduce training errors and improve classification performance.

II. LITERATURE REVIEW

One of the central challenges in quantum machine learning (QML) is the barren plateau phenomenon, which leads to vanishing gradients during the training of variational quantum circuits. This issue has been comprehensively analyzed in recent surveys, where it was shown that barren plateaus often arise due to high circuit depth, noise in near-term devices, or random parameter initialization [1], [2]. As a result, training quantum neural networks (QNNs) becomes inefficient or even infeasible for large-scale problems.

Tensor networks have emerged as an effective framework for representing large quantum systems and have been successfully applied in quantum-inspired classical machine learning. Architectures such as matrix product states (MPS), tree tensor networks (TTNs), and multiscale entanglement renormalization ansatz (MERA) provide scalable representations of entanglement structures. Recent studies have also demonstrated the potential of tensor-network quantum circuits for image classification and other supervised learning tasks [4], [10], [14].

To address the expressiveness limitations of variational quantum circuits, the method of data re-uploading was proposed. Pérez–Salinas et al. showed that even a single qubit can serve as a universal classifier when classical data is repeatedly embedded at multiple stages of the quantum computation [3]. This approach enhances model capacity without significantly increasing the number of qubits. Later works extended the idea, integrating re-uploading into more complex QML architectures.

Another important research direction is the use of gradient-free optimization methods. Traditional gradient-based approaches often face difficulties due to barren plateaus and the high cost of evaluating quantum gradients. Algorithms such as differential evolution [5] and other evolutionary strategies have been proposed as alternatives, enabling robust training of QNNs on both simulated and experimental hardware [6], [20].

In summary, prior research highlights three main directions relevant to our study:

- the challenges of barren plateaus in QML [1], [2];
- the efficiency of tensor networks for scalable quantum architectures [4], [10], [14];
- the advantages of data re-uploading and gradient-free optimization in improving model expressiveness and training stability [3], [5], [6], [20].

Building on these works, we investigate the integration of data re-uploading into tensor network models as a means to improve classification performance and mitigate barren plateaus.

III. METHODOLOGY

In quantum machine learning algorithms, there are the following steps:

1) Data Embedding: transforming classical data into quantum space.

2) Model construction: selecting and building a quantum machine learning model.

3) Model training: training a quantum machine learning model

4) Evaluation: model validation

A. Quantum Data Embedding

The first step is very important in quantum machine learning problems. The quality of learning a quantum model will depend on it. There are many ways to represent classical data in quantum space.

The most common:

- amplitude embedding;
- basis embedding;
- angle embedding.

Amplitude Embedding is one of the basic methods of encoding classical data into quantum states, which appeared in the early stages of Quantum Machine Learning. It was first systematically described in the works of Maria Schuld and Francesco Petruccione [22].

Let us have a classical vector

$$x = \left(x_0, x_1, \ldots, x_{N-1}\right) \in R^N. \tag{1}$$

We normalize it so that

$$\sum_{i=0}^{N-1} \left|x_i\right|^2 = 1. \tag{2}$$

Then the quantum state in the basis $|i\rangle$ will have the form

$$\left|\psi(x) = \sum_{i=0}^{N-1} x_i \right| i, \tag{3}$$

where x_i are the normalized components of the vector, which act as the amplitudes of the quantum state.

Basis Embedding is the simplest way to encode classical data into quantum states: each integer or bit directly corresponds to a basis state $|i\rangle$ [22].

Let us have a classical vector

$$x = \left(x_1, x_2, \ldots, x_n\right), \quad x_i \in 0,1. \tag{4}$$

Then the encoding into the quantum state is done as follows:

$$\left|\psi(x) = \right| x_1 x_2 \ldots x_n \rangle, \tag{5}$$

where $|x_1 x_2 \ldots x_n\rangle$ is the tensor product of the computational basis states:

$$|x_1 x_2 \ldots x_n\rangle = |x_1\rangle \otimes |x_2\rangle \otimes \cdots \otimes |x_n\rangle, \tag{6}$$

In Angle Embedding the numerical features are converted into rotation angles of parameterized quantum gates [22].

For a scalar parameter x_i:

$$|\psi(x_i)\rangle = R_\alpha(x_i)|0\rangle, \quad (7)$$

where $R_\alpha(\theta)$ rotation operator (for example, R_x, R_y, R_z):

For vector $x = (x_1, x_2, \ldots, x_n)$:

$$|\psi(x)\rangle = \bigotimes_{i=1}^{n} R_\alpha(x_i)|0\rangle. \quad (8)$$

In our work we decided to use Angle Embedding. We decided to use quantum gates *RX*, *RZ* to convert classical data into quantum data. First we apply *RX* layer, then *RZ* layer and so on until we have features.

$$|\psi(x)\rangle = \bigotimes_{i=1}^{n} R_z(z_i) R_x(x_i)|0\rangle. \quad (9)$$

B. Model Construction

In the second stage, we chose quantum tensor networks.

Tensor networks are a class of structured variational quantum circuits inspired by condensed-state physics. They represent large quantum systems using a connected network of smaller tensors, enabling efficient modeling and training.

In particular, tree-like tensor networks (TTNs) and matrix-product-of-states (MPSs) are well suited for one-dimensional and hierarchical data structures. We use a TTN-style architecture, where each node in the tree is implemented using rotation (RY) and entanglement (CNOT) gates.

$$|\psi(x)\rangle = \mathrm{CNOT}\big(R_y(\theta_0)|0\rangle\big) \otimes \big(R_y(\theta_1)|0\rangle\big). \quad (10)$$

You can see an example of such a network in Fig. 1.

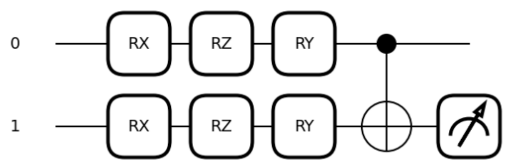

Fig. 1. Example of a quantum tensor network. *RX*, *RZ* are used to transform classical data into a quantum representation, and *RY* and CNOT are used to construct the tensor

C. Model Training

Next, we need to choose an algorithm for training a quantum artificial intelligence model. We chose differential evolution [5], because gradient methods have their drawbacks in quantum machine learning. Due to the complexity of expressing the gradient of a quantum circuit, Parameter Shift Rules, and its modifications are mainly used for training [6]. This method requires more time and resources for training, compared to free-gradient algorithms for quantum computing. Therefore, we settled on the differential evolution algorithm.

D. Evaluation

After training, we need to check the quality of training on the test sample. It is customary to divide the sample 80 by 20, or 70 by 30.

In our work, we divided the sample 70 by 30, 70 percent – training, 30 percent – test.

E. Motivation for Data Re-uploading in Tensor Networks

Now let's move on to the idea of re-uploading data.

The idea of data re-uploading in quantum machine learning was introduced by Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre in their paper "Data re-uploading for a universal quantum classifier" [3]. In their paper, the authors demonstrate that even a single qubit can be used to build a universal quantum classifier if a classical subsystem is added to the quantum processing and multiple data uploads are used. That is, instead of the traditional division of a quantum algorithm into the stages of "data upload → processing → measurement", it is proposed to periodically re-upload classical data into a quantum register during the computational process.

$$U(\phi, x) = U(\phi_N) U(x) \ldots U. \quad (11)$$

where $U(\phi_N)$ is a parametric unitary operator that parameterizes a quantum machine learning model; $U(x)$ is a unitary operator that is responsible for the feature map.

In the article, the authors demonstrated the advantages of a single qubit.

Using this idea, we hypothesize that such layering will not only improve expressiveness but also mitigate the barren plateau effect due to better gradient flow along the contour. To test this hypothesis, we designed experiments comparing tensor networks with different numbers of reloaded layers and evaluating their learning efficiency.

IV. RESULT OF EXPERIMENTS

This experiment involved data from the iris dataset. We took two classes from the iris dataset. The sample size is 100. The training sample included 70 observations, and the training sample included 30 observations. The tensor network architecture was used as shown in Fig. 2

As mentioned earlier, for the feature map, the *RX*, *RZ* gates are used (Fig. 3).

We created three models for the experiment. The first model had one layer of feature map and tensor network Fig. 4. The second model had two recurrent layers in the feature map and tensor network Fig. 5. And the third model had three such layers Fig. 6. The training was carried out using differential evolution.

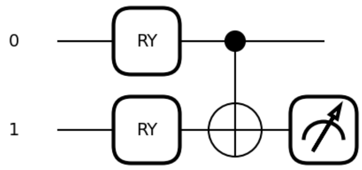

Fig. 2. Architecture of the tensor network that participated in the experiment

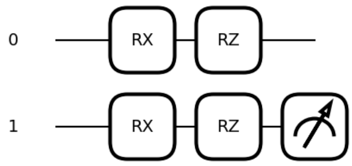

Fig. 3. Feature map, for converting classical data into quantum data

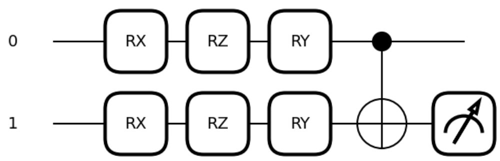

Fig. 4. Architecture of the first quantum neural network model

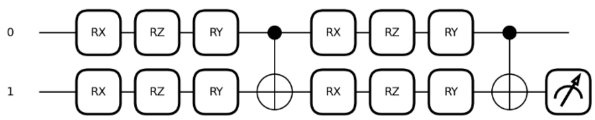

Fig. 5. Architecture of the second quantum neural network model

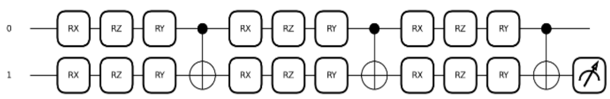

Fig 6. Architecture of the third quantum neural network model

The results of the experiment are shown in Tables 1 and 2.

As we can see from the experimental results, the best model is with the number of layers of 3.

Now consider a more complex experiment. Let's take the wine dataset. Also two classes and 100 observations. Let's divide the sample 70 by 30. We will train in the same way using differential evolution. The first model will be just a tensor network without re-uploading data Fig. 7. The second model will be with re-uploading data. In the second model, we used re-uploading data three times Fig. 8.

TABLE I. RESULT OF EXPERIMENTS (IRIS DATASET)

Model	Loss Train	Loss Test
1 layer	0.28041	0.27966
2 layers	0.14657	0.14379
3 layers	0.02582	0.06253

TABLE II. RESULT OF EXPERIMENTS (IRIS DATASET)

Model	Accuracy Train	Accuracy Test	Recall Train	Recall Test	Precision Train	Precision Test
1 layer	0.96	0.97	0.97	0.92	0.95	1.0
2 layers	1.0	1.0	1.0	1.0	1.0	1.0
3 layers	1.0	1.0	1.0	1.0	1.0	1.0

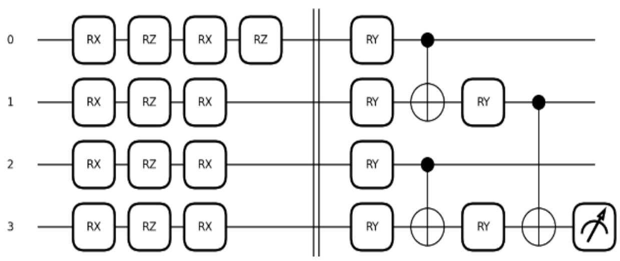

Fig. 7. Quantum neural network architecture without re-uploading data

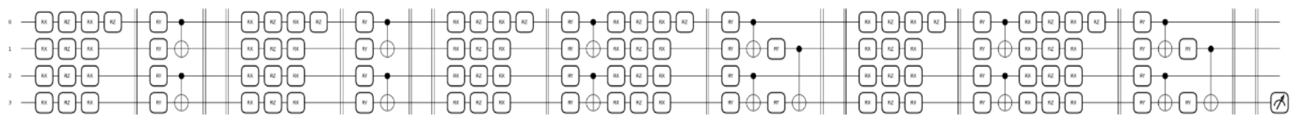

Fig. 8. Quantum neural network architecture using multiple re-uploading data

The results of the experiments can be seen in Tables 3 and 4.

As we can see, adding the re-uploading data method improves the characteristics of the quantum artificial intelligence model several times. According to the experimental results, it can also be seen that there is overfitting, but when using re-uploading data, although overfitting is preserved, if we evaluate the metrics, the quality of prediction has increased several times. If we evaluate these two models, the model with re-uploading data is better than the usual tensor network.

TABLE III. RESULT OF EXPERIMENTS (WINE DATASET)

Model	Loss Train	Loss Test
1 layer	0.92102	1.10513
3 layers	0.56452	0.834496

TABLE IV. RESULT OF EXPERIMENTS (WINE DATASET)

Model	Accuracy Train	Accuracy Test	Recall Train	Recall Test	Precision Train	Precision Test
1 layer	0.57	0.44	0.6	0.47	0.53	0.38
3 layers	0.84	0.69	0.93	0.94	0.76	0.59

So, according to the results of the experiments, we can see that adding re-uploading data to a quantum tensor network improves its performance several times.

V. CONCLUSION

In this paper, we consider the data re-uploading method in quantum tensor networks as a way to improve the quality of training and overcome the barren plateau phenomenon.

Experimental results on the classification task showed that increasing the number of re-uploading layers consistently reduces the loss and improves the accuracy, completeness, and precision. The model with three layers achieved ideal results on both training and test sets.

On a simple dataset (iris), we have excellent results. As the number of layers increases, the quality of the model increases. On a more complex dataset (wine), when adding the re-uploading data layer, the quality of the model also increases. Using the re-uploading data method, we were able to improve the quality of the model by almost 40 percent.

These results indicate that data re-uploading plays a significant role in improving the quality of training in quantum neural networks, especially in tensor network architectures.

But of course, there is another side to the coin. As the number of layers of re-uploading data increases, the complexity of the network increases, which in turn increases the complexity of its computation and makes it slow to learn. These experiments also raise the following questions: is it possible to improve the algorithm for training tensor networks with re-uploading data, are there other ways to use re-uploading data, and what is the optimal way to use re-uploading data in more complex examples, such as the wine dataset. It also remains unclear how this method will scale with larger datasets, more complex classes, or more qubits.

We plan to explore and present the results of our research in future work on all these questions.

REFERENCES

[1] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, "A review of barren plateaus in variational quantum computing," arXiv preprint, 2024. [Online]. Available: https://arxiv.org/html/2405.00781v1

[2] J. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature Communications*, vol. 9, no. 1, p. 4812, 2018. https://doi.org/10.1038/s41467-018-07090-4

[3] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, "Data re-uploading for a universal quantum classifier," *Quantum*, vol. 4, p. 226, 2020. https://doi.org/10.22331/q-2020-02-06-226

[4] A. Araujo, V. Madhok, and A. Datta, "Practical overview of image classification with tensor-network quantum circuits," *Machine Learning: Science and Technology*, vol. 3, no. 1, p. 015002, 2022. https://doi.org/10.1088/2632-2153/ac41bb

[5] R. Storn and K. Price, "Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341–359, 1997. https://doi.org/10.1023/A:1008202821328

[6] J. Schuld, B. Sweke, and M. Schuld, "General parameter-shift rules for quantum gradients," *Quantum*, vol. 5, p. 410, 2021. https://doi.org/10.22331/q-2021-03-15-410

[7] M. Schuld and N. Killoran, "Supervised learning with quantum enhanced feature spaces," *Physical Review Letters*, vol. 122, no. 4, p. 040504, 2019. https://doi.org/10.1103/PhysRevLett.122.040504

[8] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Physical Review A*, vol. 99, no. 3, p. 032331, 2019. https://doi.org/10.1103/PhysRevA.99.0323

[9] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, "Variational quantum algorithms," arXiv preprint, 2019. [Online]. Available: https://arxiv.org/pdf/2012.09265.pdf

[10] L. Wright, F. Barratt, J. Dborin, V. Wimalaweera, B. Coyle, and A. G. Green, "Deterministic tensor

network classifiers," arXiv preprint, 2022. [Online]. Available: https://arxiv.org/pdf/2205.09768.pdf

[11] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," arXiv preprint, 2014. [Online]. Available: https://arxiv.org/pdf/1412.6980

[12] T. Hur, L. Kim, and D. K. Park, "Quantum convolutional neural network for classical data classification," arXiv preprint, 2022. [Online]. Available: https://arxiv.org/abs/2108.00661

[13] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost-function-dependent barren plateaus in shallow parametrized quantum circuits," *Nature Communications*, vol. 12, no. 1, p. 1791, 2021. https://doi.org/10.1038/s41467-021-21728-w

[14] J. Biamonte and V. Bergholm, "Tensor networks in a nutshell," arXiv preprint, 2017. [Online]. Available: https://arxiv.org/abs/1708.00006

[15] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, "Quantum embeddings for machine learning," arXiv preprint, 2020. [Online]. Available: https://arxiv.org/abs/2001.03622

[16] A. Skolik, S. Jerbi, A. Elben, D. Garcia-Pintos, and H. J. Briegel, "Layerwise learning for quantum neural networks," *Quantum Machine Intelligence*, vol. 5, p. 5, 2023. https://doi.org/10.1007/s42484-023-00080-z

[17] E. Grant, M. Benedetti, S. Cao, A. Hallam, A. Lockhart, V. Stojevic, L. Wossnig, and S. Severini, "Hierarchical quantum classifiers," *npj Quantum Information*, vol. 4, p. 65, 2018. https://doi.org/10.1038/s41534-018-0116-9

[18] M. Schuld and F. Petruccione, *Supervised Learning with Quantum Computers*. Springer International Publishing, 2018. https://doi.org/10.1007/978-3-319-96424-9

[19] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," arXiv preprint, 2015. [Online]. Available: https://arxiv.org/pdf/1509.04279

[20] A. Arrasmith, M. Cerezo, L. Cincio, and P. J. Coles, "Effect of barren plateaus on gradient-free optimization," *Quantum*, vol. 5, p. 558, 2021. https://doi.org/10.22331/q-2021-10-05-558

[21] E. M. Stoudenmire and D. J. Schwab, "Supervised learning with quantum-inspired tensor networks," *Advances in Neural Information Processing Systems*, vol. 29, 2016. [Online]. Available: https://arxiv.org/abs/1605.05775

Received April 28, 2025

Sineglazov Victor. ORCID 0000-0002-3297-9060. Doctor of Engineering Science. Professor. Head of the Department.
Aviation Computer-Integrated Complexes Department, Faculty of Air Navigation Electronics and Telecommunications, State University "Kyiv Aviation Institute", Kyiv, Ukraine.
Education: Kyiv Polytechnic Institute, Kyiv, Ukraine, (1973).
Research area: Air Navigation, Air Traffic Control, Identification of Complex Systems, Wind/Solar power plant, artificial intelligence.
Publications: more than 750 papers.
E-mail: svm@nau.edu.ua

Chynnyk Petro. ORCID 0009-0002-6787-5429. PhD Student.
Institute for Applied System Analysis, Faculty of Artificial Intelligence, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.
Education: National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine, (2023).
Research interests: artificial intelligence, quantum computing, big data.
Publications: 1
E-mail: chynnyk@vivaldi.net

В. М. Синєглазов, П. А. Чинник. Перезавантаження даних в тензорній мережі

У статті представлено підхід до покращення квантових тензорних мереж за допомогою методу повторного завантаження даних. Запропонований фреймворк інтегрує кілька шарів класичного кодування даних в архітектури тензорних мереж, тим самим покращуючи їх апроксимаційну здатність та зменшуючи вплив безплідних плато на навчання. Побудова моделі спирається на деревоподібні тензорні мережі в поєднанні з обертальними вентилями RX, RZ та RY та заплутаністю CNOT, тоді як оптимізація виконується з використанням диференціальної еволюції як безградієнтного алгоритму. Оцінка була проведена на наборах даних iris та wine, порівнюючи базові тензорні мережі з архітектурами, що включають від одного до трьох шарів повторного завантаження. Результати демонструють послідовне зменшення втрат навчання та тестування, при цьому точність, повнота та прецизійність досягають 100% на наборі даних iris для трьох шарів та покращують якість прогнозування до 80% на наборі даних wine. Ці результати підтверджують, що повторне завантаження даних значно підвищує продуктивність та виразність квантових моделей на основі тензорних мереж.

Ключові слова: машинне навчання; квантові обчислення; квантове машинне навчання; повторне завантаження; тензорна мережа; безплідні плато; диференціальна еволюція; квантова нейронна мережа.

Синєглазов Віктор Михайлович. ORCID 0000-0002-3297-9060. Доктор технічних наук. Професор. Завідувач кафедри авіаційних комп'ютерно-інтегрованих комплексів. Факультет аеронавігації, електроніки і телекомунікацій, Державний університет «Київський авіаційний інститут», Київ, Україна.
Освіта: Київський політехнічний інститут, Київ, Україна, (1973).
Напрям наукової діяльності: аеронавігація, управління повітряним рухом, ідентифікація складних систем, вітроенергетичні установки, штучний інтелект.
Кількість публікацій: більше 750 наукових робіт.
E-mail: svm@nau.edu.ua

Чинник Петро Анатолійович. ORCID 0009-0002-6787-5429. Аспірант.
Кафедра штучного інтелекту, Інститут прикладного системного аналізу, Національний технічний університет Україні «Київський Політехнічний Інститут імені Ігоря Сікорського», Київ, Україна.
Освіта: Національний технічний університет Україні «Київський політехнічний інститут імені Ігоря Сікорського», (2023).
Напрям наукової діяльності: штучний інтелект, квантові обчислення, великі данні.
Кількість публікацій: 1.
E-mail: chynnyk@vivaldi.net