Implementation of reinforcement learning strategies in the synthesis of neuromodels to solve medical diagnostics tasks

Serhii Leoshchenko[a], Andrii Oliinyk[a], Sergey Subbotin[a], Viktor Lytvyn[a] and Oleksandr Korniienko[a]

[a] *National university "Zaporizhzhia polytechnic", Zhukovskogo street 64, Zaporizhzhia, 69063, Ukraine*

Abstract

The highlevel of accuracy of the functioning of artificial neural network (ANN) diagnostic models described at the resources indicates the prospects for the use of ANN in various fields of medicine for the diagnosis and forecasting of diseases. The implementation of diagnostic neuromodels into clinical practice can provide effective results during making medical decisions, contribute to improving the accuracy of diagnosis of diseases, and speed up process of examination of the patient. It is also worth noting that ANN can be used as models of the subject area under consideration. By changing the input data of the neural network model, observing the behavior of the output signals, it is possible to research the subject area under consideration, identify and investigate medical patterns that the ANN extracted during training. However, medical tasks become more complicated every time: the nature of clinical data about the patient changes, the data is constantly updated, the volume of data increases, as well as the hidden connections in the data. An additional challenge is the increased requirements for the adaptability and sensitivity of the neuromodel for a particular patient or disease. Using a reinforcement learning approach demonstrates good training results on incomplete data or in areas of high specificity. The paper investigates the possibility of using reinforcement learning strategies for the synthesis of high-precision neuromodels for subsequent use in medical diagnostics.

Keywords 1

medical diagnostics, neuromodel, synthesis, reinforcement learning, penalty and reward, duel

1. Introduction

ANNs are increasingly used in intelligent medical systems every year. Among the possible use cases are [1], [2]:

- methods that look for deviations in MRI images, mammography, X-rays. Before the pandemic, developers often created such programs to help doctors diagnose cancer. Since the beginning of the pandemic, they have been changed for the diagnosis of COVID-19;
- analysis of medical records, patient complaints. The doctor enters data about the patient into the database: test results, examination data, and the program offers treatment tactics. This is how one of the most famous programs in this industry works: Watson from IBM;
- control of medical staff. It is extremely important for the head of the clinic to understand whether the doctors prescribe the procedures and treatment correctly. The patient's medical history has everything: what he came with, what tests he was prescribed, what treatment. The method looks for anomalies and points to histories where excessive treatment is prescribed, too many procedures. Or to those where it is less than in similar cases.

IDDM-2021: 4rd International Conference on Informatics & Data-Driven Medicine, November 19-21, 2021, Valencia, Spain
EMAIL: sergleo.zntu@gmail.com (S. Leoshchenko); olejnikaa@gmail.com (A. Oliinyk); subbotin@zntu.edu.ua (S. Subbotin); lytvynviktor.a@gmail.com (V. Lytvyn); al.korn95@gmail.com (O. Korniienko)
ORCID: 0000-0001-5099-5518 (S. Leoshchenko); 0000-0002-6740-6078 (A. Oliinyk); 0000-0001-5814-8268 (S. Subbotin); 0000-0003-4061-4755 (V. Lytvyn); 0000-0003-4812-5382 (O. Korniienko)

Moreover, researchers often have to work with more specific tasks that are not so common in mass practice [3-6]:

- methods for detecting signs of early-stage Alzheimer's disease on MRI images;
- a method that looks for anomalies in X-ray images;
- methods for the control of bedridden patients. There are cameras in the wards that are connected to a program that can recognize a specific situation: a patient falling out of bed. If this happens, the nurses are automatically notified;
- methods for monitoring the workload of operating tables. The program determines how evenly the load is distributed on medical teams in different operating rooms;
- medical reference book with artificial intelligence (the doctor enters data about the patient, the program suggests a solution).

However, all these tasks are characterized by similar problems in the process of implementing ANN. For example, getting data. To create a neuromodel designed for any task, it must be trained on data. To teach her to see an anomaly on an X-ray or to determine that it is cancer and not pneumonia, she needs to show a lot of such pictures (thousands, hundreds of thousands, millions). The diagnosis must be correctly signed on all the pictures, otherwise the program will make more mistakes [7-10].

So, many researchers agree that the main difficulty of developers is: the lack of homogeneous and high-quality data. A developer can't just come to a hospital and take medical data about patients. Even taking into account the fact that they are depersonalized, for example, X-rays without a first and last name [7-10]. These data are protected by several legal laws at once: on medical secrecy, on personal data, etc. Large Western universities often provide developers with arrays of data to guarantee the ability to train a model. But then there is a problem with data compatibility. For example, the developers received a database with postoperative X-rays: control images, which are made after surgery in the patient's supine position. However, to analyze the results of screening studies, the pictures are taken most massively when the patient is standing, it is impossible to apply a system trained on such data. The patient's X-rays lying down and standing are two very different pictures. There are also always doubts about the reliability and accuracy of other people's data. It is difficult to train models that prompt a doctor to make a decision based on text data: approaches to the treatment of certain diseases may differ in each country [11-13].

Reinforcement learning is a approach of machine learning method in which a model is trained that has no information about the system, but has the ability to perform any actions in it. Actions move the system to a new state, and the model receives some reward from the system. Therefore, such a strategy can be a useful practice for solving medical problems.

So the main goal of the work will be to develop a new method of neuroevolutionary synthesis of neuromodels for medical diagnostics with the borrowing of strategies and mechanisms of reinforcement learning methods. This approach will eliminate most of the disadvantages of neuroevolutionary methods.

2. Related Works

In recent years, researchers have observed significant qualitative growth in reinforcement learning technologies. If initially this approach demonstrated good results in game tasks, then at the moment neuromodels trained with reinforcement learning methods are actively used for pattern recognition, agent management in robotics and decision-making in continuous tasks [14-20].

Sometimes reinforcement learning is distinguished not as a separate strategy, but as an offshoot from the strategy of learning with a teacher. This is due to the formulation of the assignment: a real or virtual environment acts as a teacher. However, this is the main mistake of this classification. After all, the environment in this case reacts to the agent dynamically and each time the reaction may be different. Thus, during the training process, the agent receives information from the external environment about where there is no exit, thus, he studies the surrounding world and learns to find a way out [14-20].

It should be noted that several factors have influenced the rapid development of the reinforcement learning approach [14-20]:

- increased computing speed (using powerful distributed and parallel computing systems, the use of many lightweight threads of modern GPUs);
- a significant increase in the amount of suitable data for training models in open repositories (for example, ImageNet);
- dissemination of new ANN topologies (CNN, LSTM, GRU);
- expansion and distribution of computing infrastructures (Linux, TCP/IP, Git, ROS, PR2, AWS, AMT, TensorFlow, etc.).

So in general, It can be concluded that the main impetus for recent progress is not new ideas and methods, but the intensification of computing, sufficient data, mature infrastructure. And, despite significant practical results, their theoretical basis still remains simple [16-20].

The most common and researched reinforcement learning method is Policy Gradient (PG) [21-28]. The popularity of this method is explained by theoretically supported rules for optimizing the expected reward:

- clear policy;
- transparent rules.

In general, PG can be represented in the likeness of the diagram in Fig. 1. Then basically the method will consist of performing 4 basic steps [21-28]:

1. run N scenarios τ_i with a strategy $\pi_\theta(a \mid s)$. At the same time, τ_i is a certain scenario, that is, a sequence of agent states (s_i) and actions performed in these states (a_i): $\tau = (s_1, a_1; s_2, a_2; s_3, a_3; ..., s_n, a_n)$, and the behavior of the agent (further states and actions) is determined by its stochastic strategy: $\pi_\theta(a_n \mid s_n)$;
2. calculate the arithmetic mean $\nabla_\theta J(\theta) \leftarrow \frac{1}{N}\sum_{i=1}^{N}\left(\sum_{t=1}^{T^i}\nabla_\theta \log \pi_\theta\left(a_t^i \mid s_t^i\right)\right)\left(\sum_{t=1}^{T^i} r\left(a_t^i \mid s_t^i\right)\right)$, where $J(\theta)$ is a function of the maximized mathematical expectation of the sum of the agent's winnings θ, and $\nabla_\theta J(\theta)$ is the gradient of this function. Then, $r\left(a_t^i \mid s_t^i\right)$ is the gain gained from the action a_t^i in the state s_t^i at the step T at which the transition to the terminal state occurred;
3. $\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$;
4. if result not agree to the extreme, repeat from point 1.

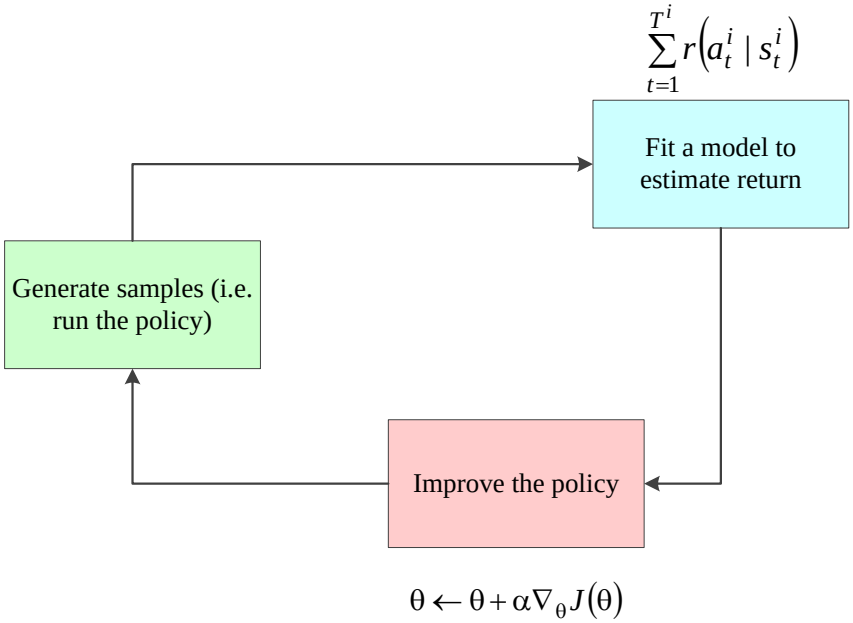

Figure 1: General scheme of the PG method

Further researches of reinforcement learning methods was found in the more complete and advanced Q-learning method [21-28]. Q-learning is a method that researched values from a special table that measures in what quality level it will be performed a certain action in any state (it can be

measured this with a simple scalar value, so the larger the value, the better the action). The values which stored in the table are called "Q-values". These are estimates of the amount of future awards. In other words, they estimate how much more reward it could be get before the end of the game by being in the (s_i) state and performing the (a_i) action. This method allows to get more information about the environment at every step. This information is used to update the values in the table [21-28].

The basic concept of Q-learning is based on the Bellman equation:

$$Q(s,a) = r + \gamma \max_{a'} Q(s',a'), \quad (1)$$

Q is a Q-Values for the state given a particular state;

s_i is a sequence of agent states (s_i);

a_i is a sequence of agent and actions;

r is a expected discounted cumulative reward;

γ is a the award in the future, devaluing future awards.

The equation states that the value of Q for a certain state-action pair should be the reward received when moving to a new state (by performing this action), added to the value of the best action in the next state. And to resolve the conflict, when the hypothesis works that receiving an award right now is more valuable than receiving an award in the future, γ number is used from 0 to 1 (usually from 0.9 to 0.99), which is multiplied by the award in the future, devaluing future awards [21-28].

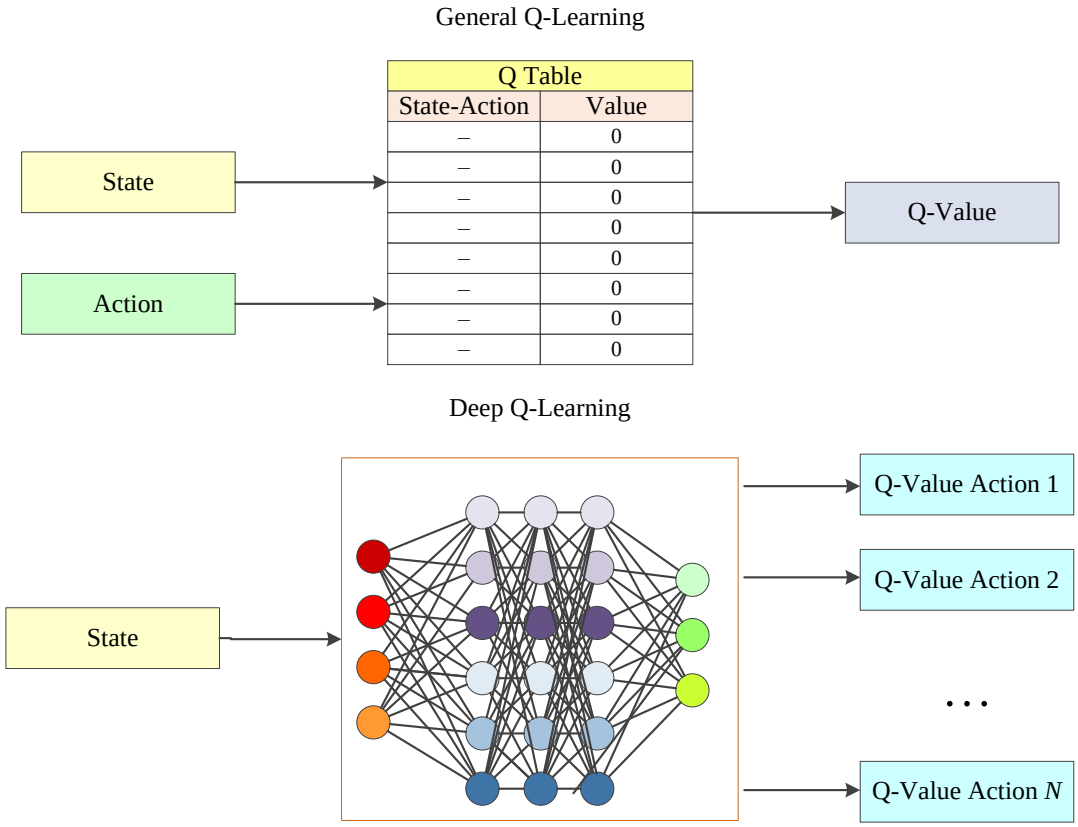

Figure 2: General scheme of the Q-learning method

Dueling Double Deep Q-Learning (Dueling DDQN) this is the most modern approach to the synthesis of neuromodels based on the principles of reinforcement learning. To approximate the optimal function of the action value (a_i), it could be used a deep Q-network: $Q(s,a,\theta)$ with the parameter θ. To evaluate this network, firstly should be optimized the following sequence of function dropouts on iteration i: $L_i(\theta_i) = E_{s,a,r,s'}\left[\left(y_i^{DQN} - Q(s,a,\theta_i)\right)^2\right]$; $y_i^{DQN} = r + \gamma \max Q(s',a',\theta')$, updating the parameters of the descent gradient such that $\nabla\theta_i L_i(\theta_i) = E_{s,a,r,s'}\left[y_i^{DQN} - Q(s,a,\theta_i)\right)\nabla\theta_i Q(s,a,\theta_i)\right]$ [21-28].

Dueling DDQN is a special state-of-the-art deep Q learning algorithm consisting of separate duel architectures that share streams of value and benefits in deep Q networks to determine the value of the

next state. Prioritizing experience reproduction, i.e. sampling mini-experience packages that have a large expected impact on learning, further increases efficiency [21-28].

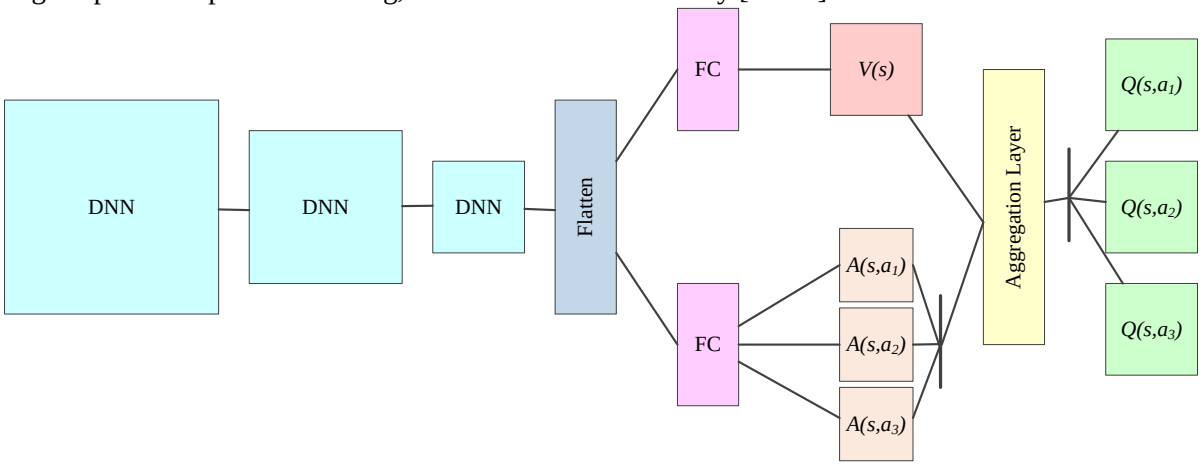

Figure 3: General architecture of the Q-learning method

3. Proposed method

As it was presented in the previous section, reinforcement learning methods have great prospects for solving problems that are poorly formalized, with incomplete data or with a dynamic environment. In our work, we propose a method based on strategies of reinforcement learning methods [29].

So it is proposed:

1. taken the neuroevolutionary synthesis of neuromodels as a basis, but with the addition of 2 separate neural networks: a network for evaluating and monitoring the environment (NN_{glob_crit}) and a network for duplicating the parameters of the best agent at the next step (NN_{best_clone});
2. the rest of the population will be a set of different individual agents ($NN_{ind's} = \{NN_{ind_1}, NN_{ind_2}, ..., NN_{ind_n}\}$);
3. during the synthesis and modification of the structure of individual individuals considered as agents in the environment, all information will be forwarded to the global network NN_{glob_crit}, whose task is to compare the current results of agents with the reference results of training data and adjust the penalty or reward for each agent;
4. at the same time, after evaluating the actions of all agents, the agent with the best results is selected at the iteration ($Q \to \max, \varepsilon_{outNN_{ind}} \to \min$) structurally and parametrically duplicated in the clone network: $NN_{best\_clone} = NN^{i}_{ind_n}(Struct_{NN}, Param_{NN})$.

The main goal of this step is to evaluate the results in the next iteration with the previously best ones.

This synthesis approach also assumes the presence of an additional identifier: the evaluation of the reward growth step $mark_{Q-lev}$ [30-34]. Such an identifier will help to avoid areas of local extremes, since if the reward value decreases less than the specified one, it is possible to change the best agent in the population peratively.

The general progress of the method is shown in Fig. 4.

4. Results and Discussions

A data set was selected for testing based on the characteristics of patients with pneumonia, which was recently presented by authors M.-A. Kim, J. Seok Park,C. W. Lee, and W.-I. Choi [35]. Total sample size: 77490 values. Table 1 shows the characteristics of the set date.

For this task, the development of neuromodels will make it much easier to determine the further diagnosis of a person after collecting data on their well-being. Given that pneumonia is one of the most important signs and complications of COVID-19 [36], [37], after additional training on advanced data, this model can be used to diagnose patients or predict the further development of disease dynamics.

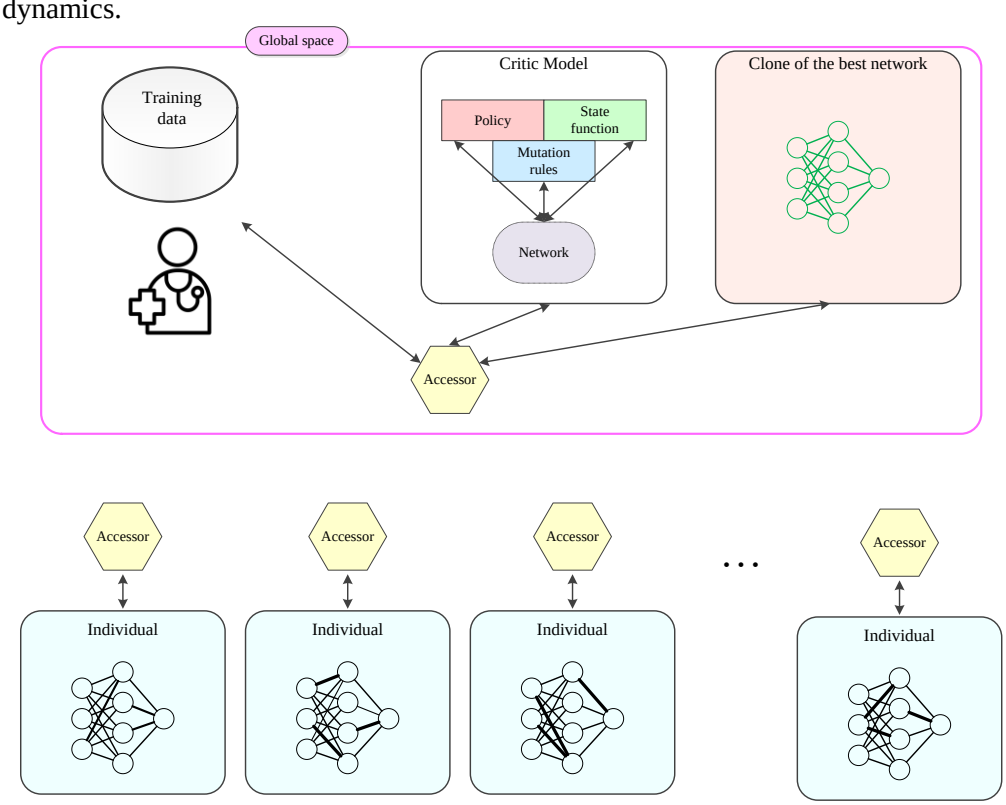

Figure 4: The general scheme of the method

Table 1
General characteristics of the data set

Total number of values	77490	Number of attributes	54
The type of the data	Numeric (after consideration of)	Number of instances'	1435

It will be compared the work of the proposed method reinforcement learning (RLNE) with the modified neuroevolution genetic algorithm method (MGA) which synthesis tasks will be RNN and DNN [14], [15], [34]. For methods compared, will be used next characteristics of the metaparameters.

Table 2
Metaparameters for methods

Metaparameter of the method	Characteristics of the metaparameter
Number of individual in population (size)	100
Elite size of population	5%
Neurons activation function	hyperbolic tangent
Probability of the mutation (for the MGA)	25%
Type of the crossover	two-point
Reward	[-1;0;1]
Types of mutation	deleting a connection between neurons removing a neuron (hidden layer) adding a connection between neurons

adding a neuron (hidden layer)
changing the type of activation function

The test results for the synthesis task are shown in Table 3.

Table 3
General results on data set

Methodof synthesis	Synthesis Time, s	Error at the training sample	Error at the test sample
RLNE	7352	0.021	0.147
MGA RNN	7173	0.03	0.157
MGA DNN	8031	0.019	0.134

Analyzing the results, it can be concluded that the proposed method has well demonstrated the synthesis time in comparison with the use of MGA for the synthesis of DNN. This is due to the fact that topologically synthesized neuromodels were simpler and their modifications required less effort. However, the time results are inferior in time to MGA for RNN synthesis. A possible explanation may be that during RNN synthesis, there was no need to clone the best individuals to compare the results, since the presence of recurrent connections makes this process easier.

Another important characteristic is the accuracy of the synthesized solutions. So the solutions obtained by RLNE were more accurate both on training and test data, but the difference in error with MGA RNN is not so significant. And the results of MGA DNN were even better. It is likely that deep networks allow encode hidden connections between data more accurately.

The second stage of the study of experimental results was focused on the characteristics of resource consumption during the synthesis of solutions. So special attention was paid to measuring the load on the CPU and RAM [38]. Such monitoring allows more accurately determine the load distribution at different iterations of the method execution. The CPU and RAM load graphs are shown in Fig. 5 and 6, respectively.

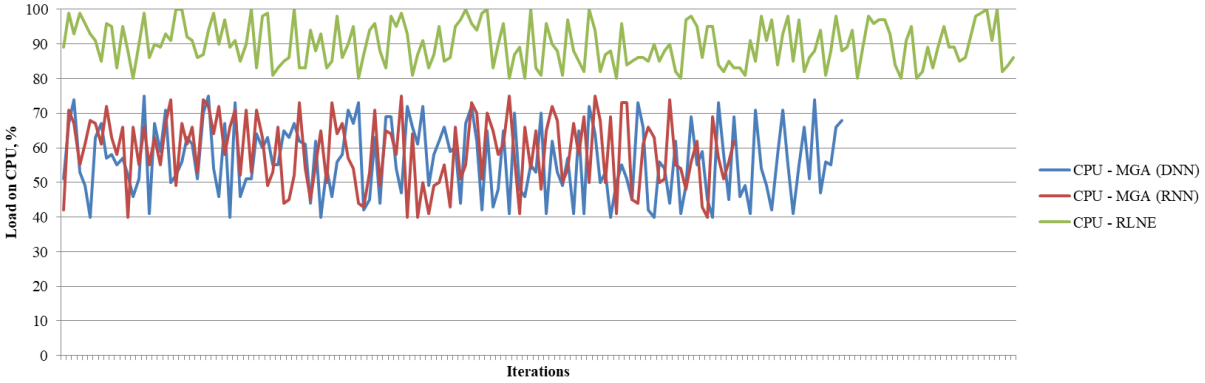

Figure 5: Load on the CPU during synthesis

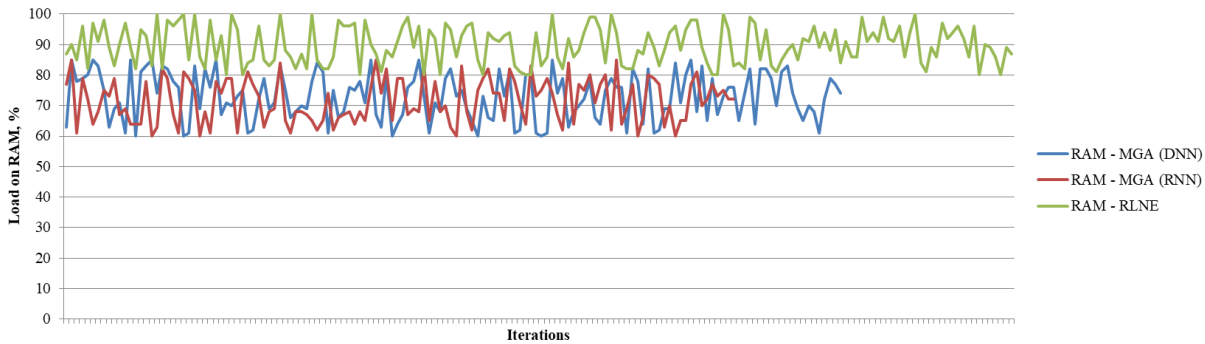

Figure 6: Load on the RAM during synthesis

During the use of MGA in both cases, the load on the CPU and RAM was more abrupt, but did not exceed the mark of 81-82% on average. When using RLNE, the load distribution was more systematic, but it often reached 100%. These indicators are important when designing a parallel approach in synthesis using methods. So a relatively low load allow implement MGA on highly productive GPUs, but the high resource consumption of RLNE, on the contrary, limits this possibility.

5. Conclusion

The proposed strategies and method demonstrated the accepted level of work. Thus, the accuracy of the resulting solution was increased by 6.4% (from 0.157 to 0.147). It was also possible to reduce the synthesis time: in comparison with analogues by 8.5% (from 8031 s to 7352 s). However, a high level of resource consumption limits the parallelization of the method, which in turn can significantly limit the genetic diversity of individuals. In the future, it is possible to implement the main strategies of the proposed method in parallel implementations of neuroevolutionary methods for the purpose of intellectual maintenance and control of populations of solutions.

Also, an important option for further research may be to simplify the proposed strategy by extracting a clone of the best result at the iteration and replacing this approach with the use of individual agents with recurrent connections, but by tightening the control of the import of the barrier from the external global critic network. On the other hand, this approach will allow to focus the work of the critic's network on the external data of the environment.

6. Acknowledgements

The work was carried out with the support of the state budget research projects of the state budget of the National University "Zaporozhzhia Polytechnic" “Intelligent methods and software for diagnostics and non-destructive quality control of military and civilian applications” (state registration number 0119U100360) and “Development of methods and tools for analysis and prediction of dynamic behavior of nonlinear objects” (state registration number 0121U107499).

7. References

[1] V.M. Adamović, D.Z. Antanasijević, M.Đ. Ristić, et al., An optimized artificial neural network model for the prediction of rate of hazardous chemical and healthcare waste generation at the national level, J Mater Cycles Waste Manag 20, pp. 1736–1750 (2018). doi: 10.1007/s10163-018-0741-6

[2] M. Johnson, A. Albizri, S. Simsek, Artificial intelligence in healthcare operations to enhance treatment outcomes: a framework to predict lung cancer prognosis, Ann Oper Res (2020). doi: 10.1007/s10479-020-03872-6

[3] A.W.Y. Khang, J.A.J. Alsayaydeh, S.M. Idrus, J.A.B.M. Gani, W.A. Indra, J.B. Pusppanathan, Resource efficient for hybrid fiber-wireless communications links in access networks with multi response optimization algorithm, ARPN Journal of Engineering and Applied Sciences, vol. 16(1) (2021) 45-50.

[4] J.A.J. Alsayaydeh, A. Aziz, A.I.A. Rahman, S.N.S. Salim, M. Zainon, Z.A. Baharudin, M.I. Abbasi, A.W.Y. Khang, Development of programmable home security using GSM system for early prevention, vol. 16(1) (2021) 88-97.

[5] J.A.J. Alsayaydeh, W.A.Y. Khang, W.A. Indra, J.B. Pusppanathan, V. Shkarupylo, A.K.M. Zakir Hossain, S. Saravanan, Development of vehicle door security using smart tag and fingerprint system, ARPN Journal of Engineering and Applied Sciences, vol. 9(1) (2019) 3108-3114.

[6] J.A.J. Alsayaydeh, W.A.Y. Khang, W.A. Indra, V. Shkarupylo, J. Jayasundar, Development of smart dustbin by using apps, ARPN Journal of Engineering and Applied Sciences, vol. 14(21) (2019) 3703-3711.

[7] G. Paragliola, M. Naeem, Risk management for nuclear medical department using reinforcement learning algorithms, J Reliable Intell Environ 5, pp. 105–113 (2019). doi: 10.1007/s40860-019-00084-z
[8] Z. Tian, X. Si, Y. Zheng, et al., Multi-step medical image segmentation based on reinforcement learning, J Ambient Intell Human Comput (2020). doi: 10.1007/s12652-020-01905-3
[9] A. Kishor, C. Chakraborty, W. Jeberson, Reinforcement learning for medical information processing over heterogeneous networks, Multimed Tools Appl 80, pp. 23983–24004 (2021). doi: 10.1007/s11042-021-10840-0
[10] M. Chitsaz, C. Seng Woo, Software Agent with Reinforcement Learning Approach for Medical Image Segmentation, J. Comput. Sci. Technol. 26, pp. 247–255 (2011). doi: 10.1007/s11390-011-9431-8
[11] I. Izonin, R. Tkachenko, V. Verhun, K. Zub, An approach towards missing data management using improved GRNN-SGTM ensemble method, Engineering Science and Technology, an International Journal, vol. 24(3) (2021), pp. 749-759. doi: 10.1016/j.jestch.2020.10.005.
[12] I. Izonin, R. Tkachenko, I. Dronyuk, P. Tkachenko, M. Gregus, M. Rashkevych, Predictive modeling based on small data in clinical medicine: RBF-based additive input-doubling method, Math Biosci Eng. 18(3) (2021), pp. 2599-2613. doi: 10.3934/mbe.2021132. PMID: 33892562.
[13] R. Tkachenko, I. Izonin, P. Tkachenko, Neuro-Fuzzy Diagnostics Systems Based on SGTM Neural-Like Structure and T-Controller, in: Babichev S., Lytvynenko V. (eds), Lecture Notes in Computational Intelligence and Decision Making. ISDMCI 2021. Lecture Notes on Data Engineering and Communications Technologies, vol 77 (2022), Springer, Cham, pp. 685-695. doi: 10.1007/978-3-030-82014-5_47
[14] J.A. Kumar, S. Abirami, Ensemble application of bidirectional LSTM and GRU for aspect category detection with imbalanced data, Neural Comput & Applic (2021). doi: 10.1007/s00521-021-06100-9
[15] A. Khan, A. Sarfaraz, RNN-LSTM-GRU based language transformation, Soft Comput 23, pp. 13007–13024 (2019). doi: 10.1007/s00500-019-04281-z
[16] M. Lu, Z. Shahn, D. Sow, F. Doshi-Velez, L.H. Lehman, Is Deep Reinforcement Learning Ready for Practical Applications in Healthcare? A Sensitivity Analysis of Duel-DDQN for Hemodynamic Management in Sepsis Patients, Proceedings of the AMIA Annual Symposium, Rockville, 2020, pp. 773-782. Published 2021 Jan 25.
[17] M. Hausknecht, P. Stone, Deep Recurrent Q-Learning for Partially Observable MDPs, AAAI Fall Symposia (2015), pp. 1-7.
[18] T.P. Lillicrap, J.J. Hunt, A. Pritzel, N.M. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, CoRR (2016), pp.1-14.
[19] A. Liu, N., Liu, Y., Logan, B. et al. Learning the Dynamic Treatment Regimes from Medical Registry Data through Deep Q-network. Sci Rep 9, 1495 (2019). doi: 10.1038/s41598-018-37142-0
[20] B. Guo, X. Zhang, Q. Sheng, H. Yang, Dueling Deep-Q-Network Based Delay-Aware Cache Update Policy for Mobile Users in Fog Radio Access Networks, IEEE Access, Vol. 8 (2020), pp. 7131-7141. doi: 10.1109/ACCESS.2020.2964258.
[21] A pair of interrelated neural networks in Deep Q-Network, 2020. URL: https://towardsdatascience.com/a-pair-of-interrelated-neural-networks-in-dqn-f0f58e09b3c4
[22] G. Delétang, Mixing policy gradient and Q-learning, 2019. URL: https://towardsdatascience.com/mixing-policy-gradient-and-q-learning-5819d9c69074
[23] Karpathy, Deep Reinforcement Learning: Pong from Pixels, 2016. URL: http://karpathy.github.io/2016/05/31/rl/
[24] G. Kesari, Catch me if you can: A simple english explanation of GANs or Dueling neural-nets, 2018. URL: https://towardsdatascience.com/catch-me-if-you-can-a-simple-english-explanation-of-gans-or-dueling-neural-nets-319a273434db
[25] C. Yoon, Dueling Deep Q Networks. Dueling Network Architectures for Deep Reinforcement Learning, 2019. URL: https://towardsdatascience.com/dueling-deep-q-networks-81ffab672751
[26] Improvements in Deep Q Learning: Dueling Double DQN, Prioritized Experience Replay, and fixed, 2018. URL: https://www.freecodecamp.org/news/improvements-in-deep-q-learning-dueling-double-dqn-prioritized-experience-replay-and-fixed-58b130cc5682/

[27] Oppermann, Self Learning AI-Agents Part II: Deep Q-Learning, 2018. URL: https://towardsdatascience.com/self-learning-ai-agents-part-ii-deep-q-learning-b5ac60c3f47
[28] L. Vazquez, Understanding Q-Learning, the Cliff Walking problem, 2018. URL: https://medium.com/@lgvaz/understanding-q-learning-the-cliff-walking-problem-80198921abbc
[29] S. Leoshchenko, A. Oliinyk, S.Subbotin, V. Shkarupylo, Using the Actor-Critic method for population diversity in neuroevolutionary synthesis, in: Proceedings of the 2nd International Workshop on Intelligent Information Technologies and Systems of Information Security (IntelITSIS'2021), CEUR-WS (2021), pp. 99–107.
[30] A. Oliinyk, I. Fedorchenko, A. Stepanenko, M. Rud, D. Goncharenko, Combinatorial optimization problems solving based on evolutionary approach, Proceedings of the 15th International Conference on the Experience of Designing and Application of CAD Systems, IEEE, Lviv, Ukraine, 2019, pp. 41–45. doi: 10.1109/CADSM.2019.8779290
[31] A. Oliinyk, I. Fedorchenko, A. Stepanenko, A. Katschan, Y. Fedorchenko, A. Kharchenko, D. Goncharenko, Development of genetic methods for predicting the incidence of volumes of emissions of pollutants in air, in: Proceedings of the 2nd International Workshop on Informatics & Data-Driven Medicine, IDDM 2019, CEUR-WS, 2019, pp 340-353.
[32] S. Leoshchenko, A. Oliinyk, S. Subbotin, T. Zaiko, A Using modern architectures of recurrent neural networks for technical diagnosis of complex systems. Proceedings of the 5th International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PICST2018), IEEE, Kharkiv, Ukraine, 2018, pp. 411–416. doi: 10.1109/INFOCOMMST.2018.8632015
[33] S. Leoshchenko, A. Oliinyk, S. Subbotin, N. Gorobii, T. Zaiko, Synthesis of artificial neural networks using a modified genetic algorithm, in: Proceedings of the 1st International Workshop on Informatics & Data-Driven Medicine, IDDM 2018, CEUR-WS, 2018, pp. 1-13
[34] S. Leoshchenko, A. Oliinyk, S. Subbotin, T. Zaiko, Using Recurrent Neural Networks for Data-Centric Business, in: D. Ageyev, T. Radivilova, N. Kryvinska (eds), Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 42. Springer, Cham, pp. 73-91. doi: 10.1007/978-3-030-35649-1_4
[35] M.-A. Kim, J. Seok Park,C. W. Lee, W.-I. Choi, Pneumonia severity index in viral community acquired pneumonia in adults, PLoS One, vol. 14(3) (2019). doi: 10.1371/journal.pone.0210102
[36] G. Raghu, K. C Wilson, COVID-19 interstitial pneumonia: monitoring the clinical course in survivors, The Lancet Respiratory Medicine, vol. 8(9) (2020) 839-842. doi: 10.1016/S2213-2600(20)30349-0
[37] C. Hani, N.H. Trieu, I. Saab, S. Dangeard, S. Bennani, G.Chassagnon, M.-P. Revel, COVID-19 pneumonia: A review of typical CT findings and differential diagnosis, Diagnostic and Interventional Imaging, vol. 101(5) (2020), 263-268. doi: 10.1016/j.diii.2020.03.014
[38] Resource Monitor, 2011. URL: https://docs.microsoft.com/en-gb/archive/blogs/yongrhee/how-to-pull-the-information-that-resource-monitor-resmon-exe-provides