**DOI: https://doi.org/10.15276/aait.03.2021.3**
**UDC 004.922**

# The use of control theory methods in neural networks' training based on a handwritten text

**Andrii V. Smorodin**
ORCID: https://orcid.org/0000-0002-3370-3197; andrey.v.smorodin@op.edu.ua
Odessa National Polytechnic University. 1, Shevchenko Ave. Odessa, 65044, Ukraine

## ABSTRACT

The paper shows the importance of reducing the neural networks' training time at present stage and the role of new optimization methods in neural networks' training. The paper researches a modification of stochastic gradient descent, which is based on the idea of gradient descent representation as a discrete dynamical system. The connection between the extreme points, to which the gradient descent iterations tend, and the stationary points of the corresponding discrete dynamical system is a consequence of this representation. The further applied stabilizing scheme with predictive control, for which a theoretical apparatus was developed by means of geometric complex analysis together with solving optimization tasks in a set of polynomials with real coefficients, was able to train a multilevel perceptron for recognizing handwritten numbers many times faster. The new algorithm software implementation used the PyTorch library, created for researches in the field of neural networks. All experiments were run on NVidia graphical processing unit to check the processing unit's resource consumption. The numerical experiments did not reveal any deviation in training time. There was a slight increase in the used video memory, which was expected as the new algorithm retains one additional copy of perceptron internal parameters. The importance of this result is associated with the growth in the use of deep neural network technology, which has grown three hundred thousand times from 2012 till 2018, and the associated resource consumption. This situation forces the industry to consider training optimization issues as well as their accuracy. Therefore, any training process acceleration that reduces the time or resources of the clusters is a desirable and important result, which was achieved in this article. The results obtained discover a new area of theoretical and practical research, since the stabilization used is only one of the methods of stabilization and search for cycles in control theory. Such good practical results confirm the need to add the lagging control and the additional experiments with both predictive and lagging control elements.

**Keywords**: Saddle Points; Neural Networks; Discrete Systems; Chaos Control

***For citation***: **Smorodin A. V. "The use of control theory methods in neural networks' training based on a handwritten text".** ***Applied Aspects of Information Technology.*** **2021; Vol. 4 No. 3: 243–249. DOI: https://doi.org/10.15276/aait.03.2021.3**

## INTRODUCTION. PROBLEM STATEMENT

Artificial neural networks ANNs ([18]) were born from attempts to mathematically model the process taking place not only in the human brain, but also in the brains of other living organisms. Today, this section is the most promising direction of Artificial Intelligence (AI) together with the appearance of new cheap resources (powerful central processing units, large HDDs [SSDs – Solid State Drives] and other technological improvements). These have opened the additional opportunities, both for researchers and for business, to make a new breakthrough not only in the application of well-known Machine Training algorithms, but also to create new architectures of deep neural networks (up to a billion neurons).

Bloomberg researches on the growth of interest and resource use in deep training projects from 2012 till 2015 have shown a rapid growth in this section of Artificial Intelligence (Fig. 1).

Since 2015, the growth has only accelerated with the appearance of growing interest from a wide range of industries, including medical, entertaining, technological and manufacturing.

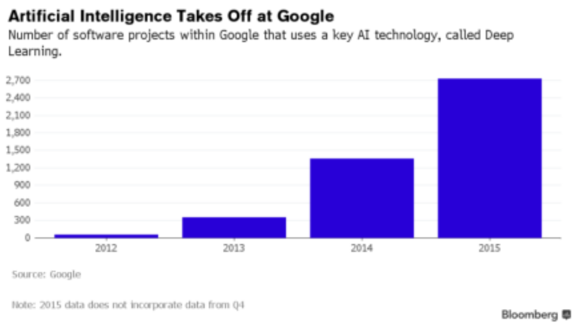



*Fig. 1.* **Number of software projects within Google that uses a key AI technology, called Deep Learning**

***Source:*** **https://www.bloomberg.com/news/articles/2015-12-08/why-2015-was-a-breakthrough-year-in-artificial-intelligence**

Along with the growing interest, the requirements for the resources required for this research and the creation of various products based on neural networks have grown. Recent researches have shown a 300,000-fold increase in resources [17] from 2012 till 2018. Such an increase in power requirements puts the neural networks' training process efficiency task in one of the first places, often ahead of the model accuracy. Training one neural network to a level comparable with the level of a specialist in a particular field can cost about 10 million US dollars, spent only on electricity and excluding other development costs and finding the optimal hyperparameters.

In addition to the direct costs of electricity, cooling and data-centers, the majority of costs are spent on building a platform for network training and automatic hyper-parameterization. The salaries of such specialists are only growing; the reason for such growth is their training complexity. The specialist in neural networks should not only be well versed in the brain structure and the construction of specialized network architectures for various applied tasks, but also should be versed in technology. The company is forced to keep a special development team. All this greatly increases the already expensive product (neural network).

The difficulty in the training process is the frequent absence of the required amount of training information. This forces to implement special processes for generating augmented data and/or training process, when some neural networks create input information for other networks, which are trained on their basis. The latter option, based on the Monte Carlo method, was used by a Google unit (DeepMind) for neural networks' training to play Go [11]. This network was trained to play Go, playing with itself for 40 days on 4 TPU (Tensor Process Unit, a special processor from Google, created to effectively train neural networks).

In addition to the networks described above, which are not directly related to solving important practical tasks, we can provide a list of really important issues that are effectively solved by deep neural networks' technologies. These are the tasks associated with processing signals from a variety of smart devices inside "smart homes" [30], recognizing emotions from images of faces or about the public health state of civil population [32].

Optimization algorithms are also the important factor in the networks' training speed and directly affect this speed, which means they require new algorithms' development and additional optimization of existing ones.

## LITERATURE OVERVIEW

Various modifications of the gradient descent algorithm are used in the process of neural networks' training [6], [8, 9], [10].

Such diversity is associated with various situations that one has to face in the networks' training practice.

Let's consider various modifications of the gradient descent algorithm that can be met in practice:

- Stochastic Gradient Decent (SGD).
- SGD with Nesterov momentum and/or modification.
- Adagrad ([9]).
- Adadelta ([10]).
- Adam ([14]).

The listed modifications basically correct some of the issues of their predecessors and use different approaches to averaging or transferring information about the gradient values in the previous steps.

For example, AdaGrad takes into account the change in each parameter of the neural network independently and thus gives an advantage in cases where some parameters of the network are more important than others. In this regard, the algorithm calculates $l_2$ the norm for each network parameter and divides the training coefficient by this norm. This leads to a constantly decreasing coefficient at the gradient, hence to a decrease in the training speed with an increase in the training time itself.

To solve this issue, another Adadelta algorithm was developed, which uses a floating window of a certain size within which the gradient changes are averaged.

The last considered algorithm Adam uses first- and second-order gradient changes with exponential decay to control both gradient changes and automatic training coefficient changes.

Many of these algorithms, notably Adam algorithm, rapidly reduce neural network prediction errors and, as some studies show, do not always generalize the result best [13].

The multidimensionality of functions that represent neural networks is another issue. Consequently, the surface of the loss function is also located in a very high dimensional space. Thus, many extreme points of this surface are "saddle" points, and not extreme in all dimensions at once [11].

Consequently, the relationship shown in the work [12] can be used again to modify gradient descent and use it to train various neural networks. One of such numerical experiments has been shown in the work [1] and it has shown excellent results.

## THE AIM AND OBJECTIVES OF THE RESEARCH

**The aim of this study** is to develop a new optimization algorithm with an innovative view of gradient descent as a discrete dynamic system and to compare the training speed of this neural network for handwriting recognition with the SGD algorithm (based on the MNIST handwritten digit image database).

In his article [1] the author considered a modification of the Standard Gradient Decent (SGD) for accelerating the CNN training process in the task of tooth segmentation on panoramic X-ray images. This study has shown the advantage of the new algorithm on a certain class of neural networks, and therefore, in this study, the task was set to test the possibilities of a new gradient descent modification in training neural networks of a different architecture.

Multilayer Perceptron, but not CNN networks, is usually used for handwriting recognition and since the network architecture greatly changes the shape of surfaces in which it is necessary to find extreme or saddle points, the question arises about testing the capabilities of the new algorithm in finding these extreme points on this class of neural networks.

In the article, the research was carried out according to the following plan:

1) to implement Multilayer Perceptron based on one of the neural network architecture libraries (PyTorch or TensorFlow);
2) to train the neural network using Stochastic Gradient Descent, as well as a new modification;
3) to compare training outcomes with each other.

## MAIN PART. THE NEURAL NETWORK TRAINING FOR HANDWRITING RECOGNITION

In 2006, another wave of neural networks' popularity has begun; an article by Hinton ([2]) can be considered its beginning. This article demonstrated the deep neural networks' capabilities in handwritten digits recognition in comparison with Support Vector Machines (SVM) on the MNIST dataset. This database contains 60,000 images for training and 10,000 for checking the results. Let's have a look at an example of images from this database (Fig. 2.)

*Fig. 2.* **Sample images from MNIST test dataset**
***Source*: https://en.wikipedia.org/wiki/MNIST_database**

A multilevel perceptron of the following architecture was implemented to conduct a comparative experiment for testing the performance of new modifications of the gradient descent algorithm

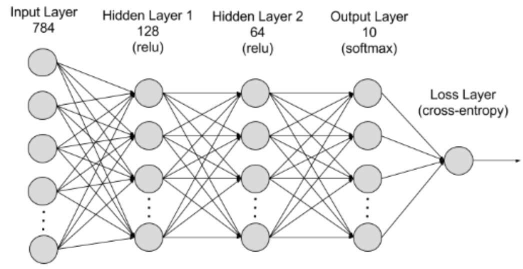



*Fig. 3.* **Architecture of multilayer perceptron**
***Source*: https://en.wikipedia.org/wiki/MNIST_database**

Let's describe the way for obtaining a modification of gradient descent as a discrete dynamical system for mini-batch gradient decent.

The main formula for changing the neural network parameters $\Theta_{t+1}$ at the $t+1$ step looks like the following

$$\Theta_{t+1} = \Theta_t - \frac{1}{B}\gamma\nabla_\Theta \sum_{i=1}^{B} L\left(f\left(\bar{x}_i;\Theta_t\right),\bar{y}_i\right), \qquad (1)$$

where $\gamma$ is a training coefficient, $f\left(\bar{x}_i;\Theta_t\right)$is a call to a function that a neural network implements with $\Theta_t$ internal parameters' values for $i$ training vector of $\bar{x}_i$; the function $L(f\left(\bar{x}_i,\Theta_t\right),\bar{y}_i)$ estimates the error between the predicted by neural network $f\left(\bar{x}_i,\Theta_t\right)$ at the $t$ step for the training vector of $\bar{x}_i$ and the expected result for it $\bar{y}$, and $B$ is a size of mini data packet.

Let's enter the notation

$$G(\Theta) = \Theta - \frac{1}{B}\gamma\nabla_\Theta \sum_{i=1}^{B} L\left(f\left(\bar{x}_i;\Theta\right),\bar{y}_i\right).$$

Then the formula (1) will be written as a discrete dynamical system

$$\Theta_{t+1} = G\left(\Theta_t\right). \qquad (2)$$

If a cycle of length 1 or a stationary point $\Theta^*$ will be found for system (2), then it can be easily shown that this point will be an extreme point for the average error, $\sum_{i=1}^{B} L\left(f\left(\bar{x}_i;\Theta_t\right),\bar{y}_i\right)$ which is the required result.

Now, when the search task has been reduced to the search for unstable stationary points of discrete dynamical systems, it is possible to apply a variety of stabilization cycles' methods developed in control theory. The work of Polyak [20] may be considered as one of the first works in this direction, in which an innovative method of predictive control was proposed. In 1992, together with this work, Pyragas proposed the so-called delayed feedback control (DFC) for continuous-time systems. This approach was transferred to the discrete case by Ushio [20] and Morgül [21].

Stabilization methods were transferred to discrete dynamical systems in vector spaces in the work [22]. The further research of the integro-differential circuits' stability led to some optimization tasks of complex analysis for polynomials. The search for optimal polynomials in the works [23, 24], [25, 26], [27, 28], [29] made it possible to obtain optimal coefficients for semilinear control.

Let's apply the semilinear control described in work [18], formula 3 for $T = 1$, to the search for a stationary point for (2). Then the gradient descent will look like the following

$$\Theta_{t+1} = ((1-\lambda)a + \lambda b)\Theta_t + (1 - ((1-\lambda)a + \lambda b))\Theta_{t-1} - (1-\lambda)\frac{1}{B}\gamma\nabla_\Theta \sum_{i=1}^{B} L\left(f\left(\bar{x}_i; a\Theta_t + (1-\lambda)\Theta_{t-1}\right), \bar{y}_i\right). \quad (3)$$

Algorithm (3) was implemented as an extension of the *torch.optim* interface for use as an optimizer within the PyTorch library. The multilevel perceptron was implemented using PyTorch tools.

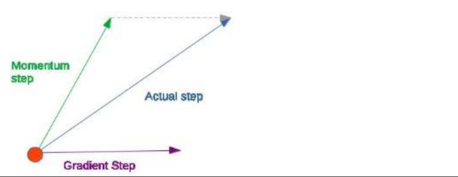
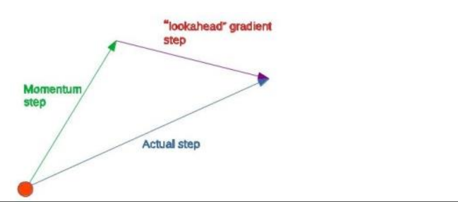
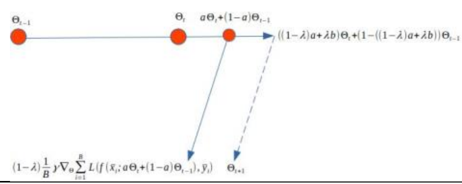
Let's consider geometrically the difference between the main modifications of gradient descent Table.

The graphs show that all algorithms behave differently in the process of predicting the next position of the optimal point $\Theta_{t-1}$. The new algorithm (3) stabilizes the search trajectory using a more flexible set of parameters, the values of which can be chosen experimentally for different neural network architectures. The optimality is determined by the ability to train the neural network faster than SGD.

## NUMERICAL EXPERIMENTS

All numerical experiments were carried out on a computer with an integrated NVidia graphical processing unit. The processing unit was equipped with 6GB RAM, which was enough for carrying out all the experiments.

*Table.* **Visual representation of next step prediction for three SGD modifications**

| Algorithm | Graph of movement |
|---|---|
| SGD with momentum |   |
| SGD with Nesterov |   |
| Algorithm (3) |   |

***Source:*** **compiled by author**

The contents of $\Theta_0$ has been saved to a file and the pseudo random number generator was initialized with one value during the first start, in order to ensure a correct comparison and start all runs from the neural network's same internal state $\Theta_0$, as well as to split the training set into the same mini-packets. All these precautions ensured the experiments' repeatability.

Only one control parameter was possible for the SGD algorithm – the $\gamma$ training factor, which was set to 0.01. The same value was used for algorithm (3). This was done only to equalize the "odds" of both algorithms.

A few numerical experiments were performed to get good values for parameters *a*, *b* and $\lambda$. The range for the enumeration was suggested by the developed theory based on geometric complex analysis and the optimization tasks solving by the special class of polynomials. These polynomials' coefficients were potential candidates for the best parameters for the algorithm (3).

The numerical experiments have shown that the best values are the following: *a = 1.2*, *b = 0.5* and $\gamma = -0.68$. The final experiment was carried out using these values; its results are shown below in Fig. 4.

It follows from the graph that the new algorithm is many times outperforms the standard gradient descent's capabilities and can reduce the time and resource costs for the neural network's training.

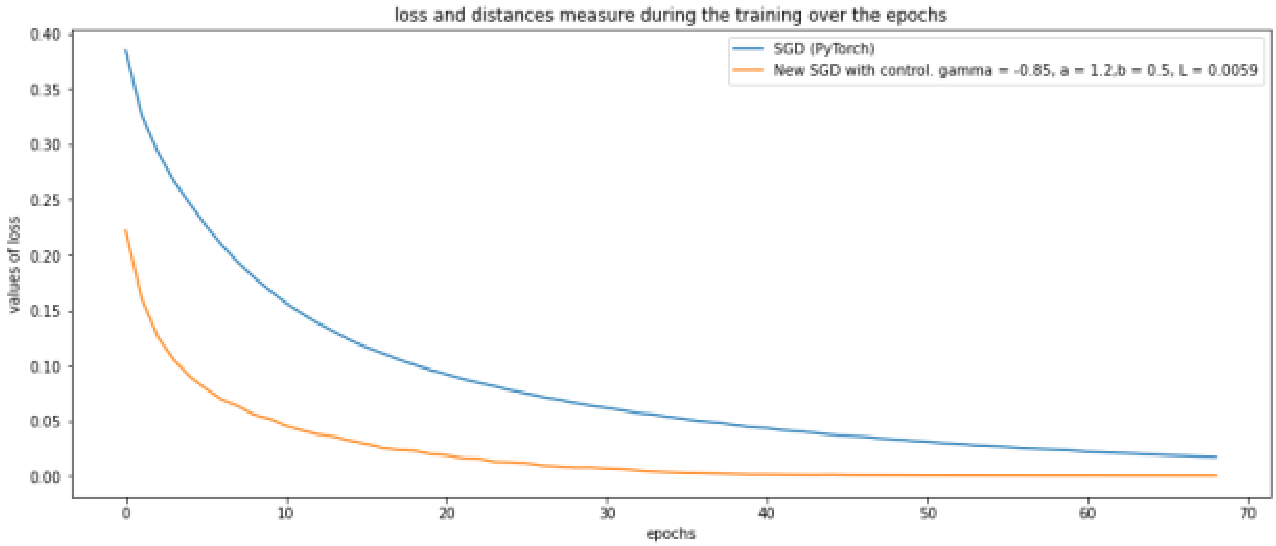



*Fig. 4.* **Comparison of performance between SGD and algorithm (3)**
***Source*: compiled by the author**

A more accurate comparison between the algorithms can be seen in another graph on Fig. 5, which reflects the epoch when algorithm (3) first calculated a value less than the value obtained by the SGD algorithm. This was the 21st epoch of training. Therefore, algorithm (3) outperforms SGD by at least 3 times.

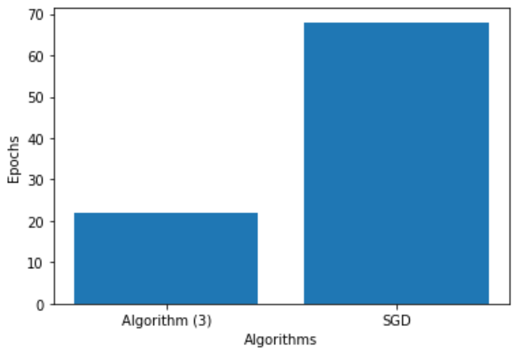



*Fig. 5.* **Bar graph of performance differences between algorithm (3) and SGD**
***Source*: compiled by the author**

## CONCLUSIONS

The article shows the connection between gradient descent and discrete dynamical systems. This relation reduces the task of neural network's iterative training for handwritten digit recognition to the search for a discrete dynamic system's stationary point.

The search for a stationary point requires the use of stabilization, since the high multidimensionality of the function describing the neural network increases the likelihood of a large number of saddle points, but not the extreme ones.

The developed predictive control methods were used to search for unstable extremes, which showed excellent performance in comparison with the SGD algorithm.

Therefore, theoretical predictions were confirmed by numerical experiments, which increases the chances of positive training outcomes for other neural network architectures and extending the control method (3) by adding some averaged historical values of the network parameters to it, similar to the averages in the AMSGrad or Adagrad algorithms.

The development of the PyTorch library optimization algorithms' majority extension with the algorithm (3) can be considered as an additional result of this work. This is a multipurpose extension and it allows using this algorithm for training any neural network architectures and solving optimization tasks, which can be solved using PyTorch.

## REFERENCES

1. Smorodin, A. "The Use of Control Theory Methods in Training Neural Networks on the Example of Teeth Recognition on Panoramic X-Ray Images". *Automation of Technological and Business Processes*. 2021; Vol. 13 No. 2.

2. Hinton, G. E., Osindero, S. & Teh, Y. "A Fast Learning Algorithm for Deep Belief Nets". *Neural Computation 18*. 2006. p. 1527–1554.

3. Rumelhart, D., Hinton, Geoffrey E. & Williams, R. J. "Learning Representations by Back Propagating Errors". *Publ. Nature*. 1986; Vol. 323: 533–536.

4. Hinton, G. E. & Shallice, T. "Lesioning an Attractor Network: Investigations of Acquired Dyslexia". *Psychological review*. 1991; 98(1): 74 p.

5. McCulloch, W. S. & Pitts, W. "A Logical Calculus of Ideas Immanent in Nervous Activity". *Bulletin of Mathematical Biophysics*. 1943; 5: 115–133.

6. Sutskever, I., Martens, J., Dahl, G. & Hinton, G. Proceedings of the 30th International Conference on Machine Learning. 2013; PMLR 28(3): 1139–1147.

7. Silver, D., Schrittwieser, J., Simonyan, K. et al. "Mastering the Game of Go without Human Knowledge". *Publ. Nature*. 2017; 550: 354–359. DOI: https://doi.org/10.1038/nature24270.

8. Polyak, B. T. & Juditsky, A. B. "Acceleration of Stochastic Approximation by Averaging". *Socity for Industrial and Applied Mathematics*. SIAM J. Control Optim. 1992; 30(4): 838–855.

9. Duchi, J., Hazan, E. & Singer, Y., "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". *Journal of Machine Learning Research.* 2011; 12(61): 2121–2159. – Available from: http://jmlr.org/papers/v12/duchi11a.html. – [Accessed: Oct. 2020].

10. Matthew, D. Zeiler. "ADADELTA: An Adaptive Learning Rate Method". 2012. – Available from: https://arxiv.org/abs/1212.5701. – [Accessed: Oct. 2020].

11. Li, H., Xu, Z., Taylor, G. & Studer, C., Goldstein. "Visualizing the Loss Landscape of Neural Nets". 2018. – Available from: https://arxiv.org/abs/1712.09913. – [Accessed: Oct. 2020].

12. Smorodin, A. "Predictive Control Methods in Tasks of Searching Saddle Points". *Proceedings of Odessa Polytechnic University*. 2020; 3(62): 80–90. DOI: https://doi.org/10.15276/opu.3.62.2020.10.

13. Ashia, C. Wilson, Roelofs, R., Stern, M., Srebro, N. & Recht, B. "The Marginal Value of Adaptive Gradient Methods in Machine Learning". 2018. – Available from: https://arxiv.org/abs/1705.08292. – [Accessed: Oct. 2020].

14. Diederik, P. Kingma & Ba, J. "Adam: A Method for Stochastic Optimization". 2017. – Available from: https://arxiv.org/abs/1412.6980v5. – [Accessed: Oct, 2020].

15. Dmitrishin, D., Lesaja, G., Skrinnik, I. & Stokolos, A. "A New Method for Finding Cycles by Semilinear Control". 2019; 383, 16: 1871–1878. DOI: https://doi.org/10.1016/j.physleta.2019.03.013.

16. Schwartz, R., Dodge, J., Noah, A. Smith & Etzioni, O. "Green AI". 2019, arXiv:1907.10597.

17. Rosenblatt, F. "The Perceptron: a Probabilistic Model for Information Storage and Organization in the Brain". *Psychol Rev*. 1958 Nov; 65(6): 386–408. PMID: 13602029. DOI: https://doi.org/10.1037/h0042519.

18. Dmitrishin, D., Iacob, I., Skrinnik, I. & Stokolos, "A. Finding, Stabilizing, and Verifying Cycles of Nonlinear Dynamical Systems". 2017. – Available from: https://arxiv.org/abs/1712.06035. – [Accessed: Oct. 2020].

19. Polyak, B. T. "Stabilizing Chaos with Predictive Control". *Autom Remote Control 66*. 2005. 1791–1804. DOI: https://doi.org/10.1007/s10513-005-0213-z.

20. Ushio, T. "Limitation of Delayed Feedback Control in Nonlinear Discrete-Time Systems". *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*. Sept. 1996; Vol. 43 No. 9: 815–816. DOI: https://doi.org/10.1109/81.536757.

21. Morgül, Ö. "On the Stability of Delayed Feedback Controllers". *Physics Letters A*. 2003. Vol. 314 Issue 4: 278–285. DOI: https://doi.org/10.1016/S0375-9601 (03)00866-1.

22. Dmitrishin D., Franzheva E., Skrinnik I. & Stokolos A., "Generalization of Nonlinear Control for Nonlinear Discrete Systems". *Bulletin of NTU "KhPI"*. Series: System Analysis, Control and information technology (in Russian, English summary). 2017; No. 28 (1250): 3–18. ISSN 2079-0023.

23. Dmitrishin, D., Smorodin, A. & Stokolos, A. "On a Family of Extremal Polynomials". *Comptes Rendus Mathematique*. 2019; 357(7): 591–596.

24. Dmitrishin, D., Smorodin, A., Stokolos, A. & Tohaneanu, M. "Symmetrization of Suffridge Polynomials and Approximation of T-symmetric Koebe Functions". *Journal of Mathematical Analysis and Applications*. 2021; 503(3): 125350 p.

25. Dmitrishin, D., Dyakonov, K. & Stokolos, A., "Univalent Polynomials and Koebe's One-Quarter Theorem". *Anal. Math. Phys*. (2019), arXiv:1812 .08311. DOI: https://doi.org/10.1007/s13324-019-00305-x.

26. Dimitrov, D. "Extremal positive trigonometric polynomials". In: B. Bojanov (Ed.). *Approximation Theory: a Volume Dedicated to Blagovest Sendov*. Darba: 2002. p. 1–24.

27. Dmitrishin, D., Hagelstein, P., Khamitova, A., Korenovskyi, A. & Stokolos, A. "Fejér Polynomials and Control of Nonlinear Discrete Systems". *Constr. Approx*. (2019), arXiv:1804.04537. DOI: https://doi.org/10.1007/s00365-019-09472-3.

28. Dmitrishin, D. V. & Khamitova, A. D., "Methods of Harmonic Analysis in Nonlinear Dynamics". *C. R. Acad. Sci. Paris:* 2013; Ser. I 351: 367–370.

29. Rogosinski, W.W. & Szegö, G. "Extremum Problems for Non-Negative Sine Polynomials". *Acta Sci. Math*. Szeged 12 (1950) 112–124.

30. Lobachev, I. M., Antoshchuk, S. G. & Hodovychenko, M. A. "Distributed Deep Learning Framework for Smart Building Transducer Network". *Applied Aspects of Information Technology. Publ. Nauka i Tekhnika*. Odessa: Ukraine. 2021; Vol. 4 No. 2: 127–139. DOI: https://doi.org/10.15276/aait.02.2021.1.

31. Petrosiuk, D. V., Arsirii, O. O., Babilunha, O. Ju. & Nikolenko, A. O. "Deep Learning Technology of Convolutional Neural Networks for Facial Expression Recognition". *Applied Aspects of Information Technology. Publ. Nauka i Tekhnika*. Odessa: Ukraine. 2021; Vol.4 No.2: 192–201. DOI: https://doi.org/10.15276/aait.02.2021.6.

32. Arsirii, O. O. & Manikaeva, O. S. "Models and Methods of Intellectual Analysis for Medical-Sociological Monitoring Data Based on the Neural Network with a Competitive Layer". *Applied Aspects of Information Technology. Publ. Science i Technical*. Odessa: Ukraine. 2019; Vol.2 No.3: 173–185. DOI: https://doi.org/10.15276/aait.03.2019.1.

Conflicts of Interest: The authors declare no conflict of interest

Received 09.01.2021
Received after revision 12.03.2021
Accepted 15.03.2021

**DOI: https://doi.org/10.15276/aait.03.2021.3**

**УДК 004.922**

# Використання методів теорії управління при вивченні нейронних мереж на прикладі рукописного тексту

**Андрій Вячеславович Смородін**
ORCID: https://orcid.org /0000-0002-3370-3197; andrey.v.smorodin@op.edu.ua
Одеський національний політехнічний університет, пр. Шевченка, 1. Одеса, 65044, Україна

## ABSTRACT

У статті чисельно досліджується модифікація стохастичного градієнтного спуску, яка була отримана через уявлення градієнтного спуску як дискретної динамічної системи. Наслідком цього подання є зв'язок між екстремальними точками, до яких прагнуть ітерації градієнтного спуску, і стаціонарними точками дискретної динамічної системи, які відповідають йому. Застосована далі стабілізуюча схема з предикативним контролем, для якої був розроблений теоретичний апарат за допомогою геометричного комплексного аналізу разом з рішенням оптимізаційних завдань у безлічі поліномів з дійсними коефіцієнтами, змогла набагато швидше навчити багаторівневий персептрон розпізнавати рукописні цифри. Програмна реалізація нового алгоритму використовувала бібліотеку PyTorch, створену для досліджень в області нейронних мереж. Всі експерименти запускалися на графічному прискорювачі компанії NVidia для перевірки споживання ресурсів прискорювача. Чисельні експерименти не виявили жодних відхилень за часом навчання. Було відзначено невелике збільшення використовуваної відео-пам'яті, як і очікувалося, оскільки новий алгоритм зберігає одну додаткову копію внутрішніх параметрів персептрону. Важливість отриманого результату пов'язана з ростом застосування технологій глибоких нейронних мереж, яке збільшилося у триста тисяч разів з 2012 по 2018 роки, та пов'язаного з цим збільшенням споживання ресурсів. Ця ситуація змушує індустрію розглядати питання оптимізації навчання на рівні з його точністю. Отже, будь-яке прискорення навчального процесу, яке скорочує час або зменшує ресурси кластерів, є бажаним і важливим результатом, якого і було досягнуто у цій статті. Отримані результати відкривають нову область теоретичних та практичних досліджень, оскільки використана стабілізація є лише одним з методів стабілізації та пошуку циклів в теорії управління. Такі хороші практичні результати підтверджують необхідність додавання запізнілого контролю і додаткових експериментів як з предикативними, так і з запізнілими елементами контролю.

**Ключові слова:** точки сідла; нейронні мережі; дискретні системи; управління хаосом

## ABOUT THE AUTHOR

**Andrii V. Smorodin,** PhD student of the Institute Computer Systems. Odessa National Polytechnic University. 1, Shevchenko Ave. Odessa, 65044,Ukraine
ORCID: https://orcid.org/0000-0002-3370-3197; andrey.v.smorodin@opu.ua
***Research field*:** Deep Learning; Neural Network; Dynamical Systems

**Андрій В'ячеславович Смородін,** PhD студент Інституту комп'ютерних систем. Одеський національний політехнічний університет, пр. Шевченка, 1. Одеса, 65044, Україна