DOI: https://doi.org/10.15276/hait.04.2021.1
UDC 004.93.1

Accelerating the learning process of a neural network by predicting the weight coefficient

Viktor O. Speranskyy[1)]
ORCID: https://orcid.org/my-orcid?orcid=0000-0002-8042-1790; speranskiyva@ukr.net. Scopus Author ID: 54401618900
Mihail O. Domanciuc[1)]
ORCID: https://orcid.org/0000-0002-9191-1357; weti369@gmail.com
[1)] Odessa National Polytechnic University, 1, Shevchenko Ave. Odessa, 65044, Ukraine

ABSTRACT

The purpose of this study is to analyze and implement the acceleration of the neural network learning process by predicting the weight coefficients. The relevance of accelerating the learning of neural networks is touched upon, as well as the possibility of using prediction models in a wide range of tasks where it is necessary to build fast classifiers. When data is received from the array of sensors of a chemical unit in real time, it is necessary to be able to predict changes and change the operating parameters. After assessment, this should be done as quickly as possible in order to promptly change the current structure and state of the resulting substances. Work on speeding up classifiers usually focuses on speeding up the applied classifier. The calculation of the predicted values of the weight coefficients are carried out using the calculation of the value using the known prediction models. The possibility of the combined use of prediction models and optimization models was tested to accelerate the learning process of a neural network. The scientific novelty of the study lies in the effectiveness analysis of prediction models use in training neural networks. For the experimental evaluation of the effectiveness of prediction models use, the classification problem was chosen. To solve the experimental problem, the type of neural network "multilayer perceptron" was chosen. The experiment is divided into several stages: initial training of the neural network without a model, and then using prediction models; initial training of a neural network without an optimization method, and then using optimization methods; initial training of the neural network using combinations of prediction models and optimization methods; measuring the relative error of using prediction models, optimization methods and combined use. Models such as "Seasonal Linear Regression", "Simple Moving Average", and "Jump" were used in the experiment. The "Jump" model was proposed and developed based on the results of observing the dependence of changes in the values of the weighting coefficient on the epoch. Methods such as "Adagrad", "Adadelta", "Adam" were chosen for training neural and subsequent verification of the combined use of prediction models with optimization methods. As a result of the study, the effectiveness of the use of prediction models in predicting the weight coefficients of a neural network has been revealed. The idea is proposed and models are used that can significantly reduce the training time of a neural network. The idea of using prediction models is that the model of the change in the weight coefficient from the epoch is a time series, which in turn tends to a certain value. As a result of the study, it was found that it is possible to combine prediction models and optimization models. Also, prediction models do not interfere with optimization models, since they do not affect the formula of the training itself, as a result of which it is possible to achieve rapid training of the neural network. In the practical part of the work, two known prediction models and the proposed developed model were used. As a result of the experiment, operating conditions were determined using prediction models.

Keywords: Neural network; learning process efficiency; prediction error; classification tasks; prediction models; optimization methods

***For citation*: Speranskyy V.O., Domanciuc M.O. Accelerating the learning process of the neural network by predicting the weight coefficient. *Herald of Advanced Information Technology.* 2021; Vol. 4 No. 4: 295–302. DOI: https://doi.org/10.15276/hait.04.2021.1**

INTRODUCTION, FORMULATION OF THE PROBLEM

Since the beginning of the 21st century, neural networks have become especially relevant, as information technologies began to develop, where it is necessary to process a huge amount of data and neural networks perfectly cope with classification tasks (for example, building a space of diagnostic features based on a large amount of input data [1]) and pattern recognition (for example, for fast online extracting of faces in a real-time video stream with use of serverless computing for the purpose of subsequent analysis]. As a result of the development of neural networks, various areas of science continued to grow actively (mathematics, physics, biology, chemistry, and, most importantly, medicine). Unfortunately, neural networks also have a disadvantage, because of which it is not always possible to use. This is the so-called "curse of dimension". It is formed due to such reasons as a large amount of input data and a large structure of the neural network. As a result, there is a need for significant computing power and a significant increase in training time (up to a year on the most voluminous tasks), as well as optimizing existing developments in order to reduce time and energy

© Speranskyy V., Domanciuc M., 2021

This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/deed.uk)

costs. Thus, the purpose of the work is to increase the neural network training speed due to accelerating the learning process, which in turn is an urgent scientific and technical task with subsequent practical application. The following task is to apply existing time series prediction models. To analyze the effectiveness of applying prediction models, they will be compared with existing methods for optimizing neural networks such as "Adagrad", "Adadelta" and "Adam". These methods make it possible to repeatedly increase the speed of neural network learning process. A feature of using prediction models is that the model data is not tied to the input data; the synapse value itself serves as coefficients in the models, which allows the models to be used on different types of neural networks.

1. LITERATURE REVIEW

The process of selecting a model for building a neural network is caused on the one hand by the simplicity of implementation, on the other hand by the accuracy of the obtained results and the magnitude of errors [3].

Solutions to various problems using neural networks involve the following steps [4]:

- data collection for training;
- data preparation and normalization;
- network topology selection;
- experimental selection of network characteristics;
- experimental selection of training parameters;
- actually, training;
- verification of the training adequacy;
- adjustment of parameters, final training;
- verbalization of the network for further use.

A study of the applicability of deep learning models for classification tasks was considered in [5], and clarification for practical tasks related, for example, to chemistry was obtained to optimize predictions of chemical patterns in [6, 7], [8].

2. MODELS ANALYSIS

Further, models and their adaptation for organization of neural network training and work with experimental data sets are considered.

2.1. "Seasonal Linear Regression" model

A "Seasonal Linear Regression" model is a model that defines a relationship between two or more correlated variables. It is used to predict the value of one variable based on the value of other variables. Relationships are usually established based on observed data [10, 14], [17, 25].

The future meaning of the synapse is expressed by equation (1):

$$w = b + k \times Error, \tag{1}$$

where: w – predicted value of the future synapse; $Error$ – the value of the global finite root-mean-square error; the coefficients k and b are calculated using the least squares method [11].

$$k = \frac{a_1 - m_{Error} * m_w}{a_2 - {m_{Error}}^2}, \tag{2}$$

$$b = m_w - k * m_{Error}, \tag{3}$$

where: a_1 – the first mixed starting moment, which is calculated using the formula (4); a_2 – the second starting moment, which is calculated by the formula (5); m_{Error} – mathematical expectation for an error, which is calculated by the formula (6); m_w – mathematical expectation for synapses, which is calculated using the formula (7).

$$a_1 = \frac{\sum_{i=0}^{n} Error_i * w_i}{n}, \tag{4}$$

$$a_2 = \frac{\sum_{i=0}^{n} {Error_i}^2}{n}, \tag{5}$$

$$m_{Error} = \frac{\sum_{i=0}^{n} Error_i}{n}, \tag{6}$$

$$m_w = \frac{\sum_{i=0}^{n} w_i}{n}, \tag{7}$$

where: $Error_i$– global root mean square error values; w_i – synapse values; n – the epoch value when the model was applied.

2.2. "Simple Moving Average" Model

The "Simple Moving Average" model is one of the most widely used models for time series prediction. From a mathematical point of view, the prime mean is the arithmetic mean at the interval [9, 15], [16, 21].

To calculate its value, the following formula is used (8):

$$w_{n+1} = w_0 + \frac{\sum_{k=1}^{n} w_k}{n}, \tag{8}$$

where: w_{n+1} – predicted synapse value; w_0 – the initial synapse value; w_k – the synapse values; n – the epoch value when the model was applied.

2.3. "Jump" Model

The "Jump" model was developed independently and works as follows: the model is divided into two parts, namely, the calculation of the delta of synapses and the rate of change in global error [14].

Delta calculation (9) occurs by subtracting the older synapse from the previous one.

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

$$\Delta w = w_n - w_{n-1}, \quad (9)$$

where: w_n – the value of the synapse in the last epoch; w_{n-1} – the synapse value in the previous epoch; n – the epoch value when the model was applied.

The calculation of the change rate of the global error is as follows: first, the delta of the global error is calculated (10), which is caused by the change of the global error in the last two epochs, and to calculate the speed (11) we divide the global error in the last epoch by the obtained delta.

$$\Delta Error = Error_n - Error_{n-1}, \quad (10)$$

where: $Error_n$ – the global error value in the last epoch; $Error_{n-1}$ – the global error value in the penultimate epoch; n – last epoch value.

$$v = \frac{Error}{\Delta Error}, \quad (11)$$

where: $Error$ – the global error value in the last epoch; $\Delta Error$ – the global error delta value.

By calculating the speed and delta of the synapses, you can multiply these values. This method makes it possible to approach the correct synapse values several times faster. It should be noted that this model applies to all synapses both at the input / output and in hidden layers [12; 18].

2.4. Experiment conditions

These models work with high efficiency, but there are exceptions, namely, when the shape of the synapse change takes on a harmonic form, then there is a high probability of obtaining a result with a large error. It is recommended to use 10-15 epochs to increase the accuracy of the prediction model. The indicated number of epochs is due to the fact that after a given number of epochs the shape of the synapse plot takes a linear form (Fig. 1).

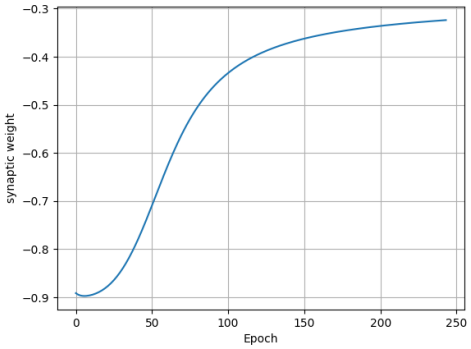


Fig. 1. **The dependence of the change in synaptic weight on the epoch**
Source: **compiled by the authors**

As a result of the analysis of the obtained information on prediction models, it became possible to adapt them to the format of weight coefficient prediction. As a result, the experimental part can be started.

A neural network model was developed to test the models (Fig. 2).

Neural network model data [13, 19], [20, 22], [23, 24]:

– number of inputs – 3;
– number of outputs – 1;
– number of hidden layers – 1;
– number of neurons in the hidden layer – 4.

A general view of the neural network is shown in Fig. 2.

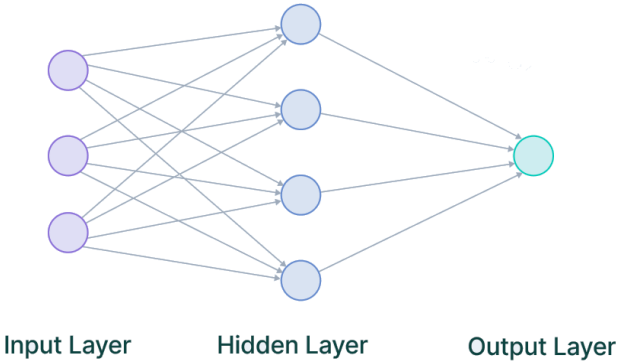


Fig. 2. **Neural network structure**
Source: **compiled by the authors**

To obtain experimental data, we will use a simple classification task, but with a learning rate of 0.5 (Table 1). The learning rate coefficient is 0.5 this is because such a value is optimal, since at a large value there will be a large correlation step, which in turn will accelerate learning, but increase the error value. If the learning rate coefficient is less than 0.5, then the correlation step will be much smaller and there will be a low error value, while the learning time will be longer. Also, to accurately determine the effectiveness of prediction models, the initial values of synapses will have fixed values. This is necessary so that the final number of eras in one experiment does not fluctuate from the random starting values of synapses. The experiment involves comparing prediction models and optimization methods, as well as the possible use of these methods simultaneously. To check the operation of the neural network, a standard technique was used: cross-checking, for which the model gave results of the same quality on the test sample as on the training sample [26].

3. EXPERIMENTAL RESULTS

An experiment was conducted where the required number of eras was measured without using the model and using models. The results of experiments are shown in Table 1.

Table 1. **Results of experiments**

Model	Number of epochs
Without model	243
“Jump”	184
“Simple Moving Average”	138
“Seasonal Linear Regression”	199

Source: **compiled by the authors**

All of the above models were used once during the entire training period in order to be able to see the difference in the number of epochs required to achieve the required accuracy.

Analysis of the use of models for the “multi-layer perceptron” were performed.

As it can be seen in Fig. 3, the plot of the change has a smooth appearance without jumps and sharp drops in the error value.

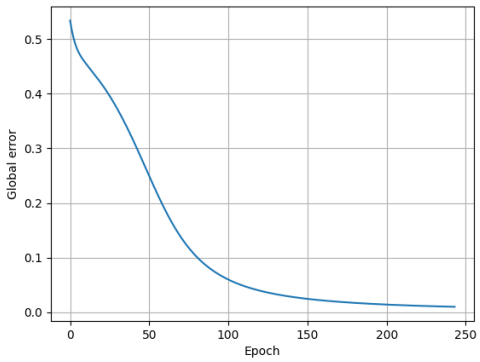


Fig. 3. **The dependence plot of the error on the epoch without using the model**
Source: **compiled by the authors**

Figure 4 shows the operation with the «Jump» model, the plot shows a sharp break in the error value.

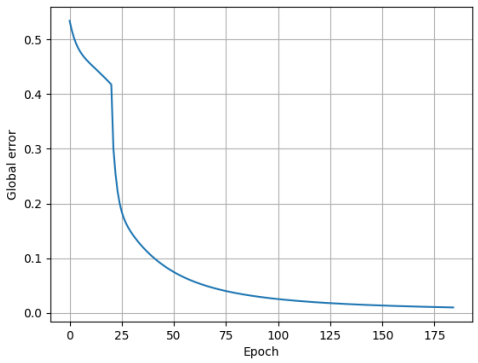


Fig. 4. **The dependence plot of the error on the epoch using the “Jump” model**
Source: **compiled by the authors**

Figure 5 shows the result of the “Simple Moving Average” model, where a jump is seen, at which the error value increased.

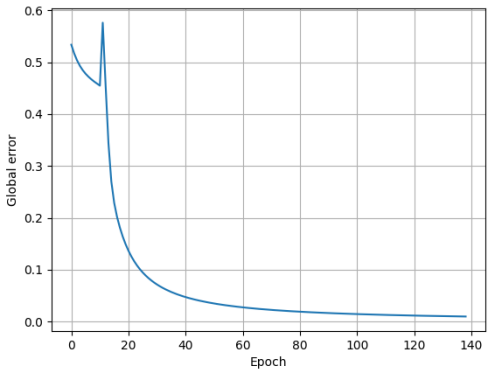


Fig. 5. **The dependence plot of the error on the epoch using the “Simple Moving Average” model**
Source: **compiled by the authors**

Figure 6 shows a sharp cut in the plot, which occurs due to the operation of the “Seasonal Linear Regression” model.

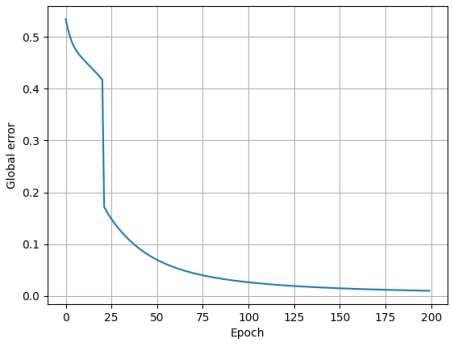


Fig. 6. **The dependence of the error on the epoch using the “Seasonal linear regression” model**
Source: **compiled by the authors**

Further, the model “Simple Moving Average” was used; it was used every 5 epochs (Fig. 7). The plot shows how quickly the neural network was able to learn, which leads to savings in both time and energy.

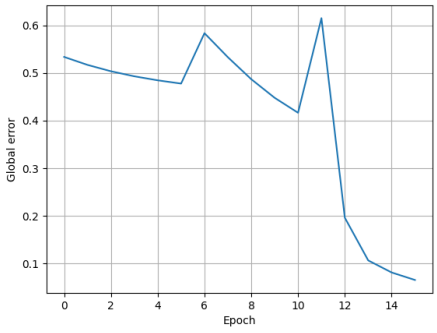


Fig. 7. **Dependences of the error on the epoch using the “Simple Moving Average” model every 5 epochs**
Source: **compiled by the authors**

Figure 8 shows the work of the model “Jump”, the plot shows a great jump in the error value.

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

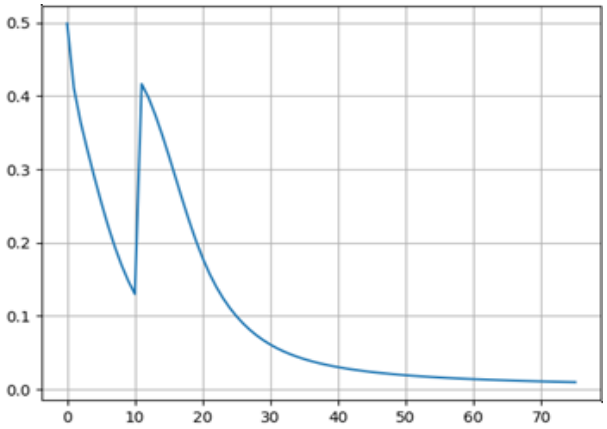


Fig. 8. **Dependences of the error on the epoch using the "Jump" model**
Source: **compiled by the authors**

An experiment was carried out where the required number of epochs was measured without use of optimization methods and using optimization methods (Table 2).

Table 2. **Results of experiments**

Optimization method	Number of epochs
No Optimization	243
"Adagrad"	8
"Adadelta"	10
"Adam"	7

Source: **compiled by the authors**

An experiment was carried out, where the required number of epochs was measured with the simultaneous use of optimization methods and prediction methods (Table 3).

Table 3. **Results of experiments**

Model / Method	"Jump"	"Simple Moving Average"	"Seasonal Linear Regression"
"Adagrad"	8	8	8
"Adadelta"	5	7	14
"Adam"	5	7	8

Source: **compiled by the authors**

Further, the measurements of the relative classification error were carried out using prediction models. As it can be seen from Table 4, the "Simple Moving Average" model has the smallest relative error. The methodic of calculation is written in [27].

Table 4. **Results of experiments**

Model	Relative error, %
Without model	14.28
"Jump"	24.95
"Simple Moving Average"	13.24
"Seasonal Linear Regression"	13.79

Source: **compiled by the authors**

Further, the measurements of the relative classification error were carried out using optimization methods. The results of using the "Adadelta" method have the smallest relative error (Table 5).

Table 5. **Results of experiments**

Optimization method	Relative error, %
No Optimization	14.28
"Adagrad"	14.13
"Adadelta"	10.66
"Adam"	22.35

Source: **compiled by the authors**

Measurements were made of the relative classification error when using the forecasting models and optimization methods together. As you can see from the Table 6, the most accurate case is when used together such combinations as:

– the "Adagrad" method together with the "Simple Moving Average" model;

– the "Adadelta" method together with the "Simple Moving Average" model;

– the "Adadelta" method together with the "Seasonal Linear Regression" model, but it should be noted that this combination increased the number of epochs for training from 7 to 14.

Table 6. **Results of experiments**

Model / Method	"Jump"	"Simple Moving Average"	"Seasonal Linear Regression"
"Adagrad"	28.68	13.63	24.27
"Adadelta"	34.96	7.71	3.44
"Adam"	27.39	22.54	22.98

Source: **compiled by the authors**

CONCLUSION

An analysis of the effectiveness of applying prediction models in training neural networks was

performed. As a result of the study, the effectiveness of using prediction models in predicting weight coefficients of the neural network was revealed. Effectiveness depends on the number of learning epochs and the final relative error. The idea was proposed and models were used, which can allow to repeatedly reduce the training time of the neural network. The work used two known prediction models and the proposed developed model. As a result of the experiments, it was revealed that the "Simple Moving Average" model can be used every 5 eras, and the "Jump" and "Seasonal Linear Regression" models can be used a limited number of times (no more than 2) because according to the plot and the obtained data, they reach the level of permissible global error equal to 0.01 in fewer epochs. In the case of using these two methods, the neural network will begin to retrain, which will cause an increase in the number of epochs for learning and error, which will tend to 100%. In addition, it was found that when using the two mentioned models, it is necessary to allocate a gap in synapse values. This is due to a certain probability that the values may be on the bend of the plot.

As a result of experiments, the efficiency of combining prediction models and optimization methods was proved. The most effective prediction model in combined use was the "Simple Moving Average" model. This model showed the greatest efficiency since with a decrease in epochs, the error also decreased.

Also, it was found that the additional memory used in the applications of prediction models and optimization models is used equally in cases of an experimental problem this number is equal to 64MB. When using GPU for calculations, additional 15MB memory was used.

The results of the study will be used to train a neural network to predict certain chemical compounds, namely metal alloys. Prediction will be based on input physicochemical parameters as; density; thermal conductivity; hardness; electrical conductivity, etc. At the outlet, the properties of such alloys will be obtained.

REFERENCES

1. Medvedew, Andrey, Fomin, Oleksandr, Pavlenko, Vitaliy & Speranskyy, Viktor. "Diagnostic features space construction using Volterra kernels wavelet transforms". *Proceedings of the 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications.* Bucharest: Romania. 2017; Vol. 2: 1077–1081. DOI: https://doi.org/10.1109/IDAACS.2017.8095251.

2. Kalnauz, D. & Speranskyy, V. "Productivity estimation of serverless computing". *Applied Aspects of Information Technology.* 2019; Vol. 1 (2): 20–28. DOI: https://doi.org/10.15276/aait.02.2019.

3. "Neural Networks: A Comprehensive Foundation". 2nd ed. *Publ. Williamce*. Moscow: Russian Federation. 2008. 1104 p. ISBN 0-13-273350-1.

4. Demuth, H. B., Beale, M. H., De Jess, O. & Hagan, M. T. "Neural network design. Martin Hagan, Stillwater". 2nd edition 2016. – Available from: https://hagan.okstate.edu/NNDesign.pdf. [Accessed: 12th Oct 2020].

5. Kruse, R., Borgelt, C., Braune, C., Mostaghim, S. & Steinbrecher, M. "Computational Intelligence: a methodological introduction". Second Edition. *Publ. Springer*. 2016. ISBN 978-1-4471-7294-9.

6. Cova, T. F. G. G. & Pais, A. A. C. C. "Deep learning for deep chemistry: optimizing the prediction of chemical patterns". *Frontiers in Chemistry*. 2019; Vol. 7: 809–831. DOI: https://doi.org/10.3389/fchem.2019.00809.

7. Galushka, M., Swain, C., Browne, F. et al. "Prediction of chemical compounds properties using a deep learning model". *Neural Computing & Applications.* 2021; 33: 13345–13366. DOI: https://doi.org/10.1007/s00521-021-05961-4.

8. Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P. & Aspuru-Guzik, A. "Automatic chemical design using a data-driven continuous representation of molecules". *ACS Central Science*. 2018; 4 (2): 268–276. DOI: https://doi.org/10.1021/acscentsci.7b00572.

9. "Forecasting algorithms. Simple moving average". SAP integrated business planning for supply chain. – Available from: https://help.sap.com/viewer/feae3cea3cc549aaa9d9de7d363a83e6/2108/en-US/a6cb81566f2adc5fe10000000a441470.html. – [Accessed: 12th Oct 2020].

10. "Forecasting algorithms. Seasonal linear regression". SAP integrated business planning for supply chain. – Available from: https://help.sap.com/viewer/feae3cea3cc549aaa9d9de7d363a83e6/2108/en-US/2a90abfcfc64409ab47b1f8e35c357a8.html. – [Accessed: 12th Oct 2020].

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)

11. Linnik, Yu. V. “Least squares method and foundations of mathematical and statistical theory of observation processing”. 2nd ed. Moscow: Russian Federation. 1962.

12. “Neural networks”. – Available from: https://www.it.ua/ru/knowledge–base/technology–innovation/iskusstvennye–nejronnye–seti–ins. – [Accessed: 12th Oct 2020].

13. Tarik, R. “Creating a neural network”. *Publ. Dialectics*. Kyiv: Ukraine. 2020. 272 p. ISBN 978-5-9909445-7-2.

14. Eileen, N. “Practical analysis of time series”. *Predicting with Statistics and Machine Learning*. – *Publ. Dialectics*. Kyiv: Ukraine. 2021. 544 p. ISBN 978-617-76-74-49-1.

15. Anasse, B., Mohamed, C. & Tommy, Yu. “Analytical predicting for teapots”. *Publ. Dialectics*. Moscow: Russian Federation. 2020. 480 p. ISBN 978-5-907203-0705.

16. Saul H. Hymans. “Forecasting and econometric models”. – Available from: https://www.econlib.org/library/Enc/ForecastingandEconometricModels.html. – [Accessed: 12th Oct 2020].

17. “Overview of economic forecasting methods”. – Available from: http://www.fhi.sk/files/katedry/kove/predmety/Prognosticke_modely/Methods_basics.pdf. – [Accessed: 12th Oct 2020].

18. Pragati Baheti. “The essential guide to neural network architectures”. – Available from: https://www.v7labs.com/blog/neural-network-architectures-guide. – [Accessed: 22th Oct 2020].

19. “Neural network: architecture, components & top algorithms”. – Available from: https://www.upgrad.com/blog/neural-network-architecture-components-algorithms – [Accessed: 12th Oct 2020].

20. Culurciello, E. “Neural network architectures”. – Available from: https://towardsdatascience.com/neural-network-architectures-156e5bad51ba. – [Accessed: 12th Oct 2020].

21. Stock, J. H. “Time Series: Economic Forecasting. International Encyclopedia of the Social & Behavioral Sciences”. Second Edition. *Elsevier.* 2015. p. 337–340. DOI: https://doi.org/10.1016/B978-0-08-097086-8.42087-8.

22. Charu, A. “Neural networks and deep learning”. *Publ. Dialectics*. Moscow: Russian Federation. 2021. 752 p. ISBN 978-5-907203-013.

23. Rashka, S. & Mirjalili, V. “Python and machine learning”. *Publ. Dialectics*. Kyiv: Ukraine. 2019. 656 p. ISBN 978-5-907114-52-4.

24. Russell, S. & Norvig, P. “Artificial intelligence. modern approach”. *Publ. Williams*. Kyiv: Ukraine. 2007. 1408 p. ISBN 978-5-907114-65-4.

25. John, H., Dean, W. & Arthur, R. “Business forecasting”. *Publ. Williams*. Kyiv: Ukraine. 2016. 56 p. ISBN 978-5-8459-2115-4.

26. Prashant Gupta. “Cross-validation in machine learning”. – Available from: https://towardsdatascience.com/cross-validation-in-machine-learning-72924a69872f. – [Accessed: 12th Oct 2020].

27. Pavlenko, V. D. & Pavlenko, S. V. (2018) “Deterministic identification methods for nonlinear dynamical systems based on the Volterra Model”. *Applied Aspects of Information Technology.* 2018; Vol. 1 (1): 11–32. DOI: https://doi.org/10.15276/aait.01.2018.1.

Conflicts of Interest: the authors declare no conflict of interest

Received 23.11.2020
Received after revision 26.11.2021
Accepted 14.12.2021

DOI: https://doi.org/10.15276/hait.04.2021.1
УДК 004.93.1

Прискорення процесу навчання нейронної мережі шляхом прогнозування вагового коефіцієнта

Віктор Олександрович Сперанський[1)]
ORCID: https://orcid.org/my-orcid?orcid=0000-0002-8042-1790; speranskiyva@ukr.net. Scopus Author ID: 54401618900

Михайло Олександрович Доманчук[1)]
ORCID: https://orcid.org/0000-0002-9191-1357; weti369@gmail.com
[1)] Одеський національний політехнічний університет, пр. Шевченка, 1. Одеса, 65044, Україна

АНОТАЦІЯ

Мета даного дослідження - аналіз та імплементація прискорення процесу навчання нейронної мережі шляхом передбачення вагових коефіцієнтів. Визначається актуальність прискорення навчання нейронних мереж, а також можливість застосування моделей прогнозування в широкому колі завдань, де потрібна побудова швидких класифікаторів. Наприклад, коли дані надходять у реальному часі від системи датчиків для подальшої оцінки структури та стану речовин. Роботи з прискорення класифікаторів зазвичай присвячені прискоренню класифікатора, що застосовується. Розрахунок прогнозованих значень вагових коефіцієнтів відбувається за допомогою розрахунку значень за допомогою відомих моделей прогнозування. Була перевірена можливість комбінованого застосування моделей прогнозування та моделей оптимізації для прискорення процесу навчання нейронної мережі. Наукова новизна дослідження полягає в аналізі ефективності застосування моделей прогнозування при навчанні нейронних мереж. Для експериментальної оцінки ефективності використання моделей прогнозування було обрано завдання класифікації. Для вирішення експериментальної задачі був обраний тип нейронної мережі «Багатошаровий персептрон». Експеримент розділений на кілька етапів: початкове навчання нейронної мережі без моделі, а потім вже з використанням моделей прогнозування; початкове навчання нейронної мережі без методів оптимізації, а потім за допомогою методів оптимізації; початкове навчання нейронної мережі з використанням комбінацій моделей прогнозування та методів оптимізації; вимірювання відносної погрішності використання моделей прогнозування, методів оптимізації та комбінованого використання. В експерименті використовувалися такі моделі, як «Сезонна лінійна регресія», «Просте середнє» і «Скачок». Модель «Скачок» була запропонована та розроблена за результатами спостереження залежностей змін значень вагового коефіцієнта від епохи. Для навчання нейронної та наступної перевірки комбінованого використання моделей прогнозування з методами оптимізації були обрані такі методи, як «Адаград», «Ададельта», «Адам». В результаті дослідження виявлена ефективність застосування моделей прогнозування при прогнозуванні вагових коефіцієнтів нейронної мережі. Представлена ідея та використані моделі, які можуть дозволити багатократно скоротити час навчання нейронної мережі. Ідея використання моделей прогнозування полягає в тому, що модель зміни вагового коефіцієнта від епохи є тимчасовим рядом, яка в свою чергу прагне до визначеного значення. В результаті дослідження було виявлено, що можна комбінувати моделі прогнозування та моделі оптимізації. Також моделі прогнозування ніяк не протидіють моделям оптимізації, так як не діють на формулу самого навчання, внаслідок чого можна досягнути швидкого навчання нейронної мережі. У практичній частині роботи були використані дві відомі моделі прогнозування та запропонована розроблена модель. В результаті експерименту були визначені умови експлуатації при застосуванні моделей прогнозування.

Ключові слова: нейронна мережа; ефективність процесу навчання; похибка прогнозування; задачі класифікації; моделі прогнозування; методи оптимізації

ABOUT THE AUTHORS



Viktor O. Speranskyy – Candidate of Engineering Sciences, Associate Professor of Computerized Control Systems Department. Odessa National Polytechnic University, 1, Shevchenko Ave. Odessa, 65044, Ukraine
ORCID: https://orcid.org/my-orcid?orcid=0000-0002-8042-1790; speranskiyva@ukr.net.
Scopus Author ID: 54401618900
***Research field*:** Nonlinear systems; modeling; identification; software development; neural networks

Віктор Олександрович Сперанський – кандидат технічних наук, доцент кафедри Комп'ютеризованих систем управління. Одеський національний політехнічний університет, пр. Шевченка, 1. Одеса, 65044, Україна



Mihail O. Domanciuc – master Student of Computerized Control Systems Department. Odessa National Polytechnic University, 1, Shevchenko Ave. Odessa, 65044, Ukraine
ORCID: https://orcid.org/0000-0002-9191-1357; weti369@gmail.com
***Research field*:** Software development; neural networks

Михайло Олександрович Доманчук – магістр кафедри Комп'ютеризованих систем управління, Одеський національний політехнічний університет, пр. Шевченка, 1, Одеса, 65044, Україна

ISSN 2663-0176 (Print)
ISSN 2663-7731 (Online)