Information, Computing and Intelligent Systems, 2025, No. 7, 39 – 48

UDC 004.75 (004.62) https://doi.org/10.20535/2786-8729.7.2025.341480

OPTIMIZATION NEURAL NETWORK FOR TIME SERIES PROCESSING

Oleksii Pysarchuk
https://orcid.org/0000-0001-5271-0248

Danylo Baran*
https://orcid.org/0009-0007-0361-6870

National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukrainee

*Corresponding author: danil.baran15@gmail.com

The article proposes the architecture of the optimization neural network and the model of test sample synthesis for the process of extrapolation of time series parameters. In particular, the addition of an input layer with the introduction of an optimization scheme of nonlinear trade-offs has been implemented. Extrapolation of the behavior of the time series was carried out according to a test sample, which is formed as a data model with the selection of the trend according to the method of least squares. The scientific novelty of the results obtained in the article is reflected in the essence of these decisions.

The aim of the research is to develop an optimization network architecture and data model for extrapolation, which allows to improve the accuracy and time of predicting the behavior of the time series outside the observation interval. Subject of research: architecture of an artificial neural network and methods of extrapolation of time series. Object of research: processes of architectural synthesis of an artificial neural network and extrapolation of time series behavior outside the observation interval.

The optimization layer provides mini-requirements for the approximation of training and test samples. This is especially appropriate for time series with stochastic noise and allows you to reduce the impact of random errors on time series prediction results. The use of model data for extrapolation allows you to determine the behavior of the time series outside the observation interval. At the same time, the forecasting time with acceptable accuracy characteristics increases. These solutions are reflected in the name of the optimization neural network, which is proposed by the authors. The study of the effectiveness of the proposed solutions was implemented by methods of simulation modeling on a modified artificial neural network. The results of the calculations proved an increase in the adequacy of data models and an increase in the accuracy of extrapolation.

Keywords: mathematical model, multi-criteria optimization, time series, artificial neural network.

1. Introduction

In the modern practice of processing Time Series, the direction of application of artificial neural networks is actively evolving. The deep learning paradigm in time series processing is to train a neural network on a sample of the current implementation (within the observation interval) and use it to predict the behavior of the time series in the observation area and in perspective (extrapolation) / retrospective (interpolation). The result is a generated time series with the most similar properties to the level of a separate implementation.

However, the use of neural networks for time series processing has a number of features and limitations. In particular, this is seen in taking into account random data errors and solving the problem of extrapolating the behavior of the time series outside the observation interval.

In connection with the above, the scientific and applied task of modifying the architecture of the neural network and the procedure for forming a test sample to predict the behavior of the time series outside the observation intervals is relevant.

2. Literature review and problem statement

The theory and practice of time series processing has three classes of methods: statistical learning; approximation and deep learning [1–4]. At present, deep learning methods using artificial neural networks have been rapidly developed. In this direction, convolutional, recurrent and artificial neural networks built on the architecture of large language models are actively used [5–8].

Classical methodologies and technologies of deep learning at Time Series have the specifics of predominantly empirical implementation and, as a result, the need to improve the following tasks [5–8].

1. Formation of metrics of losses minimized in the learning process. After all, the indicators of the efficiency of processing / reproduction / extrapolation of time series differ from the processes of sacrifice, for example, natural language and images. Time-series processing is task-oriented, and performance metrics may include requirements for minimizing random errors, or identity to inputs, etc.

2. The structure of the input layer and subsequent layers and the classical learning scheme are oriented towards the requirement of "rigid" identity/correspondence of the training and test samples. However, for time series processing this is not always acceptable, since it may include the requirement of minimizing random errors of the input data. In this sense, the requirement of minimax approximation of the training and test parts of the dataset is considered more acceptable. Currently, this is not presented in known sources [6–8]. Since the application of deep learning methods is oriented towards the repetition of the time series at the level of a separate implementation. The task of minimizing random noise is not considered and relies on statistical learning methods.

3. The architecture of a neural network in terms of the number of internal layers, neurons (except for the input and output layers), the type of neural network depends on the structure of the input data, the target processing task and the requirements for efficiency indicators. Architectural solutions, at present, are determined mostly empirically and do not have analytical models. The practice of modernity demonstrates the dominant use of three types of architectural solutions of neural networks: convolutional; recurrent; transformers of Large Language Models (LLM). However, an agreed analytical decision on the choice of an effective solution for their application has not yet been formed.

4. Time series processing by deep learning methods is not carried out sporadically on relatively small data structures. This is caused by the practical spheres of existence of time series. Therefore, data splitting into batches is used to replicate data. The partitioning process is also implemented mainly by empirical methods.

5. The main paradigm of neural networks is working with training pairs: training and test samples. Forecasting takes place on the current instance, which must belong to the set of the test sample. For problems of predicting the behavior of the time series outside the observation interval / extrapolation, this means the availability of data on the behavior of the time series outside the observation. This is problematic, since this is what the network is actually built for. Practice and current research solve this problem in the following areas. The use of recurrent networks, where the previous value is used as an element of the test sample for prediction, is an analogue of recurrent filtering with all its disadvantages [6–8]. Using sliding window algorithms. In this sense, the use of statistical training methods with all the modifications proposed in the work in combination with simulation modeling methods is seen as promising.

The specified tasks outline the general state of the issues of deep learning methods and the relevance of further research. The article is devoted to the development of approaches for preprocessing input data to take into account their random errors in the learning process and generating a test sample for predicting the behavior of time series.

3. The aim and objectives of the research

The purpose of the research is to develop an optimization network architecture and a data model for extrapolation, which allows to increase the accuracy and timeliness of forecasting the behavior of a time series outside the observation interval.

The achievement of the formed goal is realized in two directions:

– formation of the input layer of the neural network according to the optimization scheme of nonlinear trade-offs. This should provide a minimax approach in the requirements of the maximum approximation of the training and test sample in the conditions of random data errors;

– development of an effective approach in generating a test block of data outside the observation interval to obtain highly accurate predictive properties of the neural network.

4. Materials and methods for developing an optimization neural network for time series processing

The research is focused on theconvolutional neural network.This choice is due to the rather successful experience of their application for processing large Time Series arrays with training from the accumulated sample of measurements. This corresponds to the essence of the problem of predicting the activity of information content, when to solve the problem we have a sample of Time Series format as the only implementation from which a DataSet for training a neural network is formed.

The applied field of time series processing is the activity of information content in global information networks. The indicator of activity that forms a time series is the frequency of tonality of information messages $F_{j,area,tonality}$ [9].

The focus of the research is on modifying the structure and processes of the convolutional neural network in the following directions: adding an optimization layer to the neural network by preprocessing the input data; applying statistical learning methods in combination with simulation modeling methods and forming a test sample for extrapolation.

4.1. Formation of the input layer of the neural network using the optimization scheme of nonlinear compromises

It is well known that the basic ideology of training artificial neural networks is to achieve, ideally, the absolute identity of the training and test parts / training pair of the input data array / dataset. This is controlled by minimizing the loss metric. In the basic implementation of artificial neural networks, this is reflected in the additivity of input data that forms the output parameter of a single neuron. This is a kind of "hard" requirement of input-output identity for a dataset training pair. More natural is the "soft" requirement for the maximum approximation of the input – output of the training pair. These are the natural realities of learning and identification processes. This is especially justified when the inputs, for our Time Series case, are contaminated with random noise. Therefore, the input-output approximation in maximum similarity takes into account the probabilistic properties of the input data. It is proposed to implement the logic of "maximum similarity" as an alternative to the absolute identity of the training pair using the logic of multi-criteria optimization using a nonlinear scheme of compromises.

The paper proposes to add to the classical architecture of an artificial neural network *a layer of preprocessing of input data*, which is implemented by multi-criteria optimization logic. To distinguish neural networks formed in this way, the authors propose to call them *optimization neural networks*.

A nonlinear scheme of compromises for solving multi-criteria problems on discrete data is implemented according to the convolution of partial criteria [10]:

$$I = \sum_{l=1}^{b} \gamma_{0l} (1 - y_{0l})^{-1} \rightarrow min, \tag{1}$$

where $l = 1...b$ – the number of partial optimality criteria included in the convolution; γ_{0l} – normalized weight factor; y_{0l} – normalized partial criterion.

Convolution (1) allows obtaining an integrated estimate as an aggregation of a set of partial optimality criteria. The model (1) provides a nonlinear scheme of compromises under many competing criteria and, in comparison with known analogues, has proven properties [10]. In particular: the solution belongs to the Pareto area and implements a minimax scheme for solving a multi-criteria problem; on a limited interval of consideration of convolution criteria, unimodal.

Studies have proven that for the processing of Time Series, noise-contaminated weight coefficients γ_{0l} should be taken in inverse proportion to the standard deviation of the input data error $\gamma_{0l} = 1/\sigma_{y_{0l}}$. This achieves an objective measure of confidence in the input measurements, taking into account the random error of their receipt.

Convolution is minimizable, so it has different directions of extremum for values from the test sample for those approaching zero (minimization) and for those approaching one (maximization). For this purpose, it is proposed to implement the normalization of data that is reduced to one with the requirement to minimize them. Then the convolution for the conditional zero remains unchanged (1), and for the conventional unit it is transformed to the form:

$$I = \sum_{l=1}^{b} \gamma_{0l} (1 - (1/y_{0l}))^{-1} \rightarrow min. \tag{2}$$

The threshold of the conventional unit is proposed to be set, for example, at a level greater than 0.6.

Thus, the input data y_0 of the neural network is subject to preprocessing according to the transformation model:

$$I_{opt} = \begin{cases} \gamma_{0l} (1 - y_{0l})^{-1}, \ if \ y_0 < 0.6; \\ \gamma_{0l} (1 - (1/y_{0l}))^{-1}, \ if \ y_0 > 0.6. \end{cases} \tag{3}$$

The expression (3) de facto forms an additional input layer of the neural network to the classical architecture.

4.2. Generating test data outside the observation interval

A trained neural network interprets (predicts) data on the time interval of the existence of the test sample – on the observation interval. For extrapolation tasks – predicting the development of a controlled process outside the observation interval – it is also necessary to have a test sample. For this purpose, it is proposed to use statistical training methods with the modifications proposed in the work in combination with simulation modeling methods.

Then the test sample for extrapolation will have two components: the original component – the input data $F_{j,area,tonality}$ at the observation interval and the synthesized component at the prediction interval y_{model}:

$$y\_test\_ext = \left[F_{j,area,tonality}, y_{model} \right]. \tag{4}$$

The synthesized component $y_{test_{out}}$ is formed as an additive mixture of OLS estimates of input parameters – trend y_{model_0} and random normal error φ with the standard deviation of the original sample:

$$y_{model} = \left[y_{model_{0i}} + \varphi_i \right], i = 1 \dots m. \tag{5}$$

The proposed solutions regarding the loss metric, input layer and extrapolation process can be applied to any type of neural networks used for time series processing: convolutional, recurrent, LLM. The paper provides an example of the development and application of a convolutional neural network with the introduction of the proposed innovations.

Studies have proven the effectiveness of the architecture of an optimization neural network of the convolutional type and the structure of input data with parameters: dividing the input dataset into portions `batch_size=200` dimensions; the number of neurons of the input, hidden and output layers was: `n_neurons=[512,256,128,64]`. The architecture of the neural network is built in accordance with the regression of the power series at the base of two. The parameters of the structure and division of the input data array are established empirically and are the best for the problems of predicting the development of time series of the format $F_{j,area,tonality}$.

The architecture of a convolutional-type optimization neural network for processing, prediction, and extrapolation of the Time Series is shown in Fig. 1.

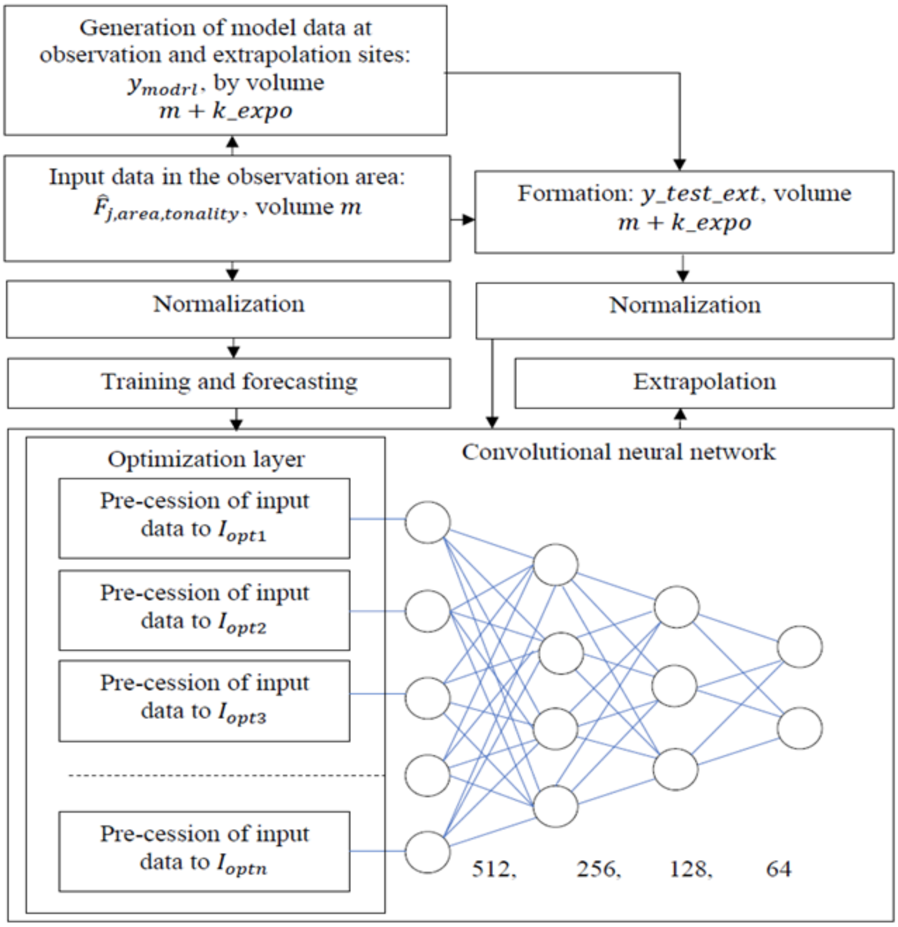

Fig. 1. Architecture of a convolutional optimization neural network to process, predicts, and extrapolates time series

The structural diagram of Fig. 1 includes all the necessary stages: data preparation and preprocessing, generation of a test sample for extrapolation, training, forecasting and extrapolation. The presence of an optimization first layer with preprocessing of data according to a nonlinear scheme of compromises (1), (2) is used to designate such a type of networks as optimization neural networks.

5. Results of verification and evaluation of the effectiveness of the application of an optimization neural network for time series processing

The study of the effectiveness of the proposed solutions was implemented by methods of mathematical modeling on real data.

Research conditions. A sample of 348 dimensions with a monotonous trend and random errors was to be processed: zero mathematical expectation; MSEdeviation $\sigma = 0.99$.

Extrapolation was carried out on 100 dimensions, so training was implemented within 248 incoming dimensions. The test sample for extrapolation was formed as an additive mixture of trend from least squares method (OLS) [2] and random normal noise.

The trend model of the input sample with parameters determined by the algorithm of the OLS with preprocessing described above is as follows:

$$F(t) = 28.03423083418919 + 0.004773157735137576t + \\ + 0.0008240042970011226t^2 - 3.204450289838111\,(\mathrm{e}-06)\,t^3 + \\ + 7.543074486030252\,(\mathrm{e}-10)\,t^4 + 6.87528764332385\,(\mathrm{e}-12)\,t^5. \qquad (6)$$

The dynamics of changes in the input sample is reflected in the graph Fig. 2.

On the graph of Fig. 2 and on subsequent graphs, the abscissa axis denotes the measurement number of the discrete sample / time – t, and the ordinate axis – the value of the monitored parameter $F(t)$ in the notations.

Parameters of the wrapping neural network in the model experiment: dividing the input dataset into portions `batch_size=200` dimensions; `epochs=100`; the number of neurons of the input, hidden, and output layers was: `n_neurons=[512,256,128,64]`. To generate a test sample to extrapolate the parameters of the studied process outside the observation interval to the OLS trend (6), a random normal error with a variable value was added $\sigma = 1.0, 0.9, 0.1$.

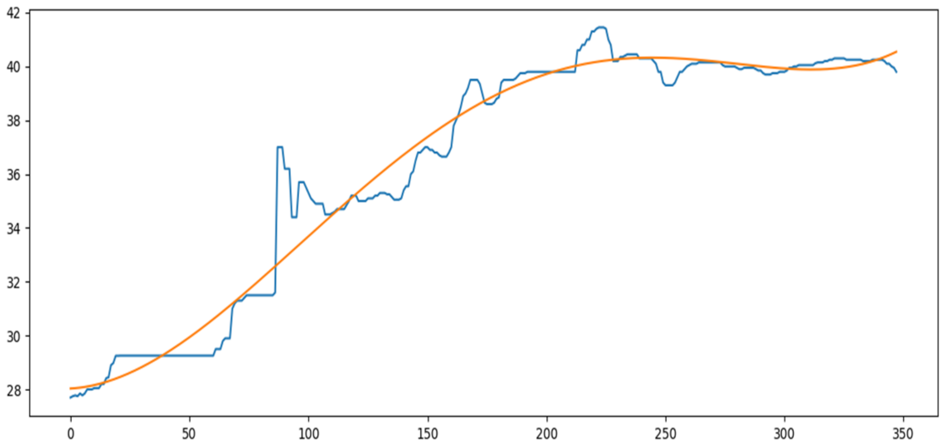

Fig. 2. Input data and trend of OLS

The training parameters of the neural network were formed in the range of 0-248 dimensions. The test sample for extrapolation was formed as a combination of real input data (interval 0-248) and synthesized data (interval 249–348). Extrapolation was provided for 100 measurements (interval 249-348) synthesized in the described way by a test extrapolation sample with efficiency assessment on a known input sample (348 measurements).

The graph of the test sample for extrapolation, as a combination of real and synthesized data is $\sigma = 1.0$ shown in the graph Fig. 3.

The controlled parameters of training progress and the effectiveness of prediction and extrapolation were: Mean Absolute Error (MAE); Mean Squared Error (MSE) / Root Mean Squared Error (RMSE); coefficient of determination / probability of approximation – R2; loss = MSE.

The results of the research are presented in Table 1 and in Fig. 4,5.

The results presented in general indicate the effectiveness of the proposed solutions from the standpoint of reproduction and forecasting of time series with high efficiency indicators.

6. Discussion of the obtained results of the study of the effectiveness of the optimization network architecture and the data model for extrapolation

The analysis of the presented results showed that the established neural network architecture and data preparation for the defined conditions are acceptable for the processes of prediction and

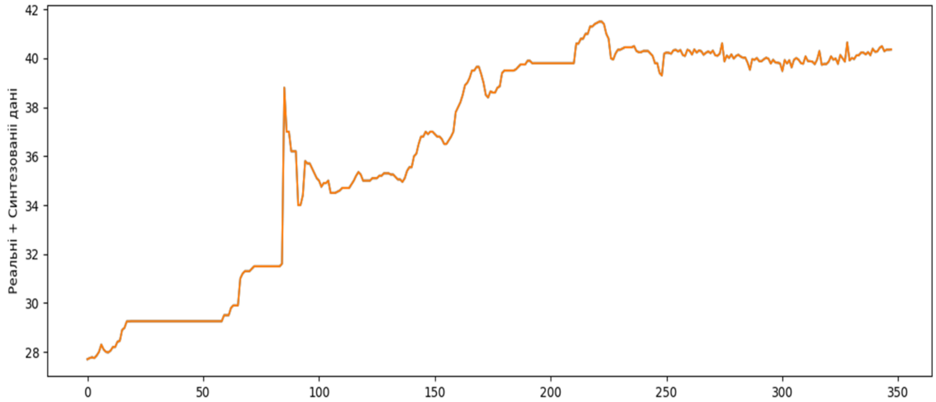

Fig. 3. Graph of the test sample for extrapolation

Table 1. Optimization Neural Network Research Results

Setting	Learning dynamics			Prediction	Extrapolation
	1	22	100		
σ = 1.0					
Convolutional neural network – a well-known analogue					
MAE	2.2293	0.0604	0.0080	0.00770	1.30636
MSE	21.4090	0.0072	9.3980e-05	14. 35234e-05	2.48986
RMSE	-	-	-	0.011980	1.57792
R2	-50.7250	0.9825	0.9998	0.999523	0.86950
loss	21.4090	0.0072	9.3980e-05	-	-
With optimization layer					
MAE	2.2253	0.3043	0.0034	0.00287	0.66466
MSE	21.2251	0.1233	2.6881e-05	3.41680e-05	0.73367
RMSE	-	-	-	0.00584	0.85654
R2	-50.2806	0.7022	0.9999	0.99990	0.96154
loss	21.2251	0.1233	2.6881e-05	-	-

extrapolation of time series parameters of at least 0.5 observation intervals. This is evidenced by high indicators of the concordance coefficient (kept at the level of 0.9) and a relatively low level of model error (MSE) and MAE.

In the extrapolation section, there is a typical increase in errors in the MSE and MAE indicators, while maintaining the value of the probability of approximation at a high level of 0.9. Therefore, the formed solutions and the results obtained are suitable for practical application.

For the observation and extrapolation interval, there is a gain in all indicators exactly from the use of an optimization neural network. An increase in the value of the random component of the error in the data enhances the gain, which ranges from 5 to 20%, depending on the indicator and the error of the input data.

The use of OLS is proposed as a test sample. The combination of the method of statistical and deep learning has positive results and significant prospects, which are due to the following. A neural network is able to accumulate significant amounts of information about the implementation of a time series as a random process. This achieves a flexible mechanism for reproducing predicted and extrapolated data with an analogy with the accumulation of static information using the Monte Carlo statistical test method. That is, a neural network is able to contain the experience of a significant number of implementations of random processes of the time series format. At the same time, the

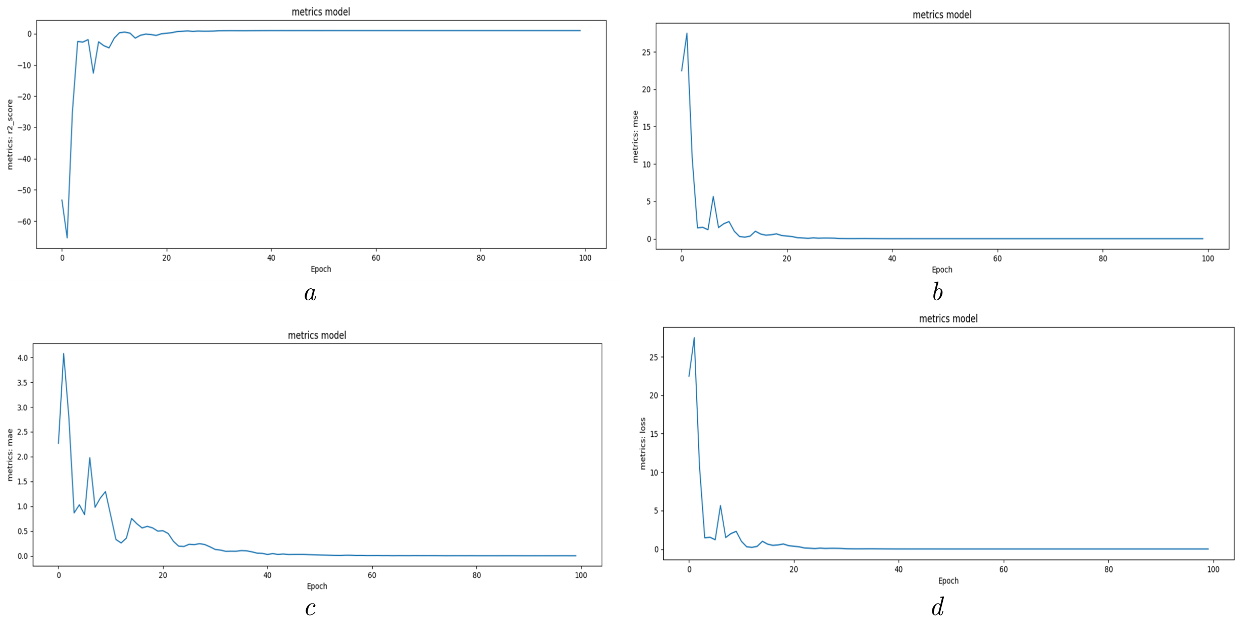

Fig. 4. Dynamics of learning effectiveness indicators: *a* – Coefficient of determination; *b* – Average absolute error; *c* – RMSE error; *d* – Loss

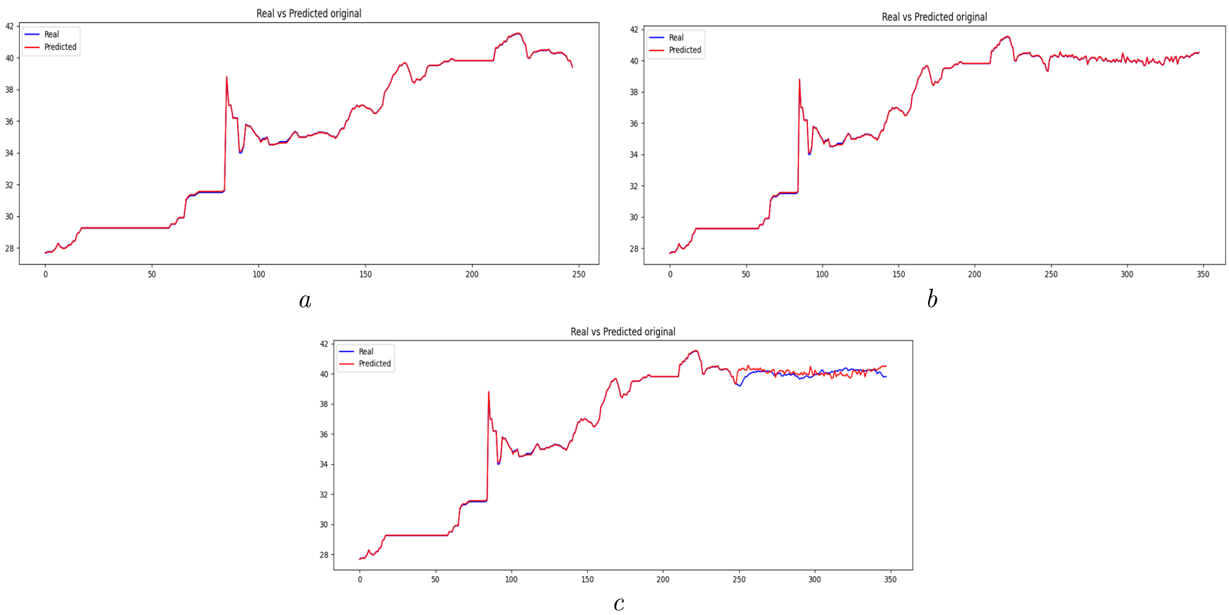

Fig. 5. Graph of forecasting and extrapolation results: *a* – Forecasting results; *b* – Extrapolation results; *c* – Extrapolation results and real data

MNC successfully complements the neural network with its predictive properties with one single implementation of a random process. The above studies are implemented with the training of a neural network for one implementation of the time series. On a limited data warehouse, training is implemented in the traditional way of dividing input data into portions – `batch_size`. We should expect positive results in the optimization layer and in the support of OLS for training neural networks on big data. For the practical task of predicting the activity of information content, this is seen in the accumulated experience of solving such a problem in a neural network based on retrospective data.

The experience of using the optimization layer only at the input level has proven the stability of the results to random data errors. In the future, it is assumed to synthesize neural networks with a convolution operation that spreads to other layers at the neural level.

Conclusions

Thus, in the course of the research, optimization neural network architecture and a data model for extrapolation were developed, which allows increasing the accuracy and time of forecasting the behavior of the time series outside the observation interval.

1. The architecture of the neural network, compared to known solutions, is supplemented with an optimization input layer. This is implemented by pre-processing the input data using a nonlinear trade-off scheme. Such a modification allows taking into account random errors of the input data and increasing the accuracy of the time series estimation.

2. To determine the behavior of the time series outside the observation area, it is proposed to use simulation modeling. In this case, the data trend is determined by the least squares method. Random normal and anomalous errors with the input data parameters are added to the trend. This improves the predictive properties of artificial neural networks.

3. The results of the study of the effectiveness of the proposed solutions proved the gain in accuracy in terms of the standard deviation of the error and the reliability of the model up to 30 percent on the extrapolation interval. The comparison was made with known analogues - a convolutional neural network without an optimization layer. At the same time, the forecasting time increased from several measurements to half the observation interval.

References

[1] E. Egrioglu, M. Khashei, C. H. Aladag, I. B. Turksen, and U. Yolcu, "Advanced Time Series Forecasting Methods," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–2, 2015. https://doi.org/10.1155/2015/918045

[2] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning," *Springer*, New York, 2009. https://doi.org/10.1007/978-0-387-84858-7

[3] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, Jan. 2003. https://doi.org/10.1016/s0925-2312(01)00702-0

[4] M. Khashei and M. Bijari, "A new class of hybrid models for time series forecasting," *Expert Systems with Applications*, vol. 39, no. 4, pp. 4344–4357, Mar. 2012. https://doi.org/10.1016/j.eswa.2011.09.157

[5] A. Tealab, "Time series forecasting using artificial neural networks methodologies: A systematic review," *Future Computing and Informatics Journal*, vol. 3, no. 2, pp. 334–340, Dec. 2018. https://doi.org/10.1016/j.fcij.2018.10.003

[6] P. Singh, M. K. Singh, R. Singh, and N. Singh, "Federated Learning: Challenges, Methods, and Future Directions," *EAI/Springer Innovations in Communication and Computing*, Springer International Publishing, pp. 199–214, 2022. https://doi.org/10.1007/978-3-030-85559-8_13

[7] L. Su, X. Zuo, R. Li, X. Wang, H. Zhao, and B. Huang, "A systematic review for transformer-based long-term series forecasting," *Artif Intell Rev*, vol. 58, no. 3, Jan. 2025. https://doi.org/10.1007/s10462-024-11044-2

[8] R. Velastegui, L. Zhinin-Vera, G. E. Pilliza, and O. Chang, "Time Series Prediction by Using Convolutional Neural Networks," *Advances in Intelligent Systems and Computing*, Springer International Publishing, pp. 499–511, Oct. 31, 2020. https://doi.org/10.1007/978-3-030-63128-4_38

[9] D. Baran and O. Pysarchuk, "Mathematical Model of Clustering of Informational Messages with Indicators of Activity for the Information Content by Tone and Areas of Society Activity," *Inf. Comput. and Intell. syst. j.*, no. 6, pp. 230–241, Sept. 2025. https://doi.org/10.20535/2786-8729.6.2025.339127

[10] A. N. Voronin, "A method of multicriteria evaluation and optimization of hierarchical systems," Cybern Syst Anal, vol. 43, no. 3, pp. 384–390, May 2007.https://doi.org/10.1007/s10559-007-0060-8

УДК 004.75 (004.62)

ОПТИМІЗАЦІЙНА НЕЙРОННА МЕРЕЖА ДЛЯ ОБРОБКИ ЧАСОВИХ РЯДІВ

Олексій Писарчук
https://orcid.org/0000-0001-5271-0248

Данило Баран
https://orcid.org/0009-0007-0361-6870

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна

В статті запропоновано архітектура оптимізаційної нейронної мережі та модель синтезу тестової вибірки для процесу екстраполяції параметрів часових рядів. Зокрема реалізовано додавання вхідного прошарку з впровадженням оптимізаційної схеми нелінійних компромісів. Екстраполяцію поведінки часового ряду здійснено за тестовою вибіркою, що формується як модель даних з виділенням тренду за методом найменших квадратів. Наукова новизна отриманих в статті результатів відображається суттю зазначених рішень.

Мета досліджень полягає в розробці архітектури оптимізаційної мережі та моделі даних для екстраполяції, що дозволяє підвищити точність та час прогнозування поведінки часового ряду поза межами інтервалу спостереження. Дослідження стосується процесів архітектурного синтезу штучної нейронної мережі та екстраполяції поведінки часових рядів за межами інтервалу спостереження. Предметом дослідження є архітектура штучної нейронної мережі та методи екстраполяції часових рядів.

Оптимізаційний прошарок забезпечує мінімаксні вимоги до наближення навчальної та тестової вибірок. Це особливо доречно до часових рядів із стохастичним шумом і дозволяє зменшити вплив випадкових похибок на результати прогнозування часового ряду. Використання для екстраполяції модельних даних дозволяє визначити поведінку часового ряду поза межами інтервалу спостереження. При цьому збільшується час прогнозування із прийнятними характеристиками точності. Зазначені рішення відображені в назві оптимізаційної нейронної мережі, яка запропонована авторами. Дослідження ефективності запропонованих рішень реалізовано методами імітаційного моделювання на модифікованій згортковій штучній нейронній мережі. Результати розрахунків довели підвищення адекватності моделей даних та збільшення точності екстраполяції.

Ключові слова: математична модель, багатокритеріальна оптимізація, часовий ряд, штучна нейронна мережа.