Вісник Тернопільського національного технічного університету
https://doi.org/10.33108/visnyk_tntu
Scientific Journal of the Ternopil National Technical University
2024, № 2 (114) https://doi.org/10.33108/visnyk_tntu2024.02
ISSN 2522-4433. Web: ***visnyk.tntu.edu.ua***

UDC004.8

A MULTIVARIATE METHOD OF FORECASTING THE NONLINEAR DYNAMICS OF PRODUCTION NETWORK BASED ON MULTILAYER NEURAL MODELS

Vasyl Martsenyuk[1,2]; Nataliia Kit[2]

[1]University of Bielsko-Biala, Bielsko-Biala, Poland
[2]Ternopil Ivan Puluj National Technical University, Ternopil, Ukraine

Summary. *Design of production network based on multilayer neural models is considered in this paper. Design of production network is crucial because it determines the optimal location of production and logistics facilities, affects cost efficiency, customer service level and overall competitiveness in the global market. Multi-layer neural networks play an important role in this process, using advanced algorithms, machine learning models and optimization techniques to analyze huge amounts of data. Special attention is focused on qualitative analysis of dynamic behavior, dynamic lattice model. The model includes rate constants and initial conditions affecting the model trajectories, which can be classified as a stable site, limit cycle, or chaotic attractor. We aim to solve the problem of qualitative behavior of the model as a problem of multilayer neural models. A multivariate method of predicting nonlinear dynamics was used to construct the training data set. Neural networks defined by regenerative architectures with linear and non-linear outputs were analyzed and compared. As a result of the analysis, it was found that architectures with linear outputs show better correspondence between expected and predicted values. Architectures with non-linear outputs, despite their complexity, exhibit less accuracy and more deviation compared to linear ones. The single-layer architecture with linear outputs shows the best accuracy, although the two-layer architecture with linear outputs has the lowest rms error. Architectures with non-linear outputs have faster training times but poor accuracy, while architectures with linear outputs require more training time but have lower errors. The results obtained in the work indicate the importance of choosing the right architecture of the neural network depending on the tasks and requirements for accuracy and training time of the model.*

Key words: *Design of production networks, lattice model, qualitative analysis, multivariate forecasting method, multilayer neural models.*

https://doi.org/10.33108/visnyk_tntu2024.02.039 *Received 13.02.2024*

Statement of the problem. Design of production network is of crucial importance, as it determines the optimal placement of production and logistics facilities, affects cost efficiency, the level of customer service and overall competitiveness in the global market [1]. This makes it possible for companies to position strategically their activities to meet the demands, minimize costs and adapt quickly to changing market conditions. Modern production network design with the integration of multilayer neural networks is aimed at optimizing the location of logistics and production sites.

Analysis of available investigation results. Multilayer neural networks play an important role in the process of designing production network using advanced algorithms, machine learning models, and optimization techniques to analyze huge amounts of data [2]. In the field of big data, with increasing computing capabilities, neural networks showed great strength in solving data classification and regression problems [3]. One of the key achievements is the application of predictive analytics to forecast demand patterns, allowing companies to locate strategically their facilities closer to high-demand regions [4].

Machine learning techniques represented by multilayer neural networks have attracted considerable attention due to their powerful capabilities in pattern classification, speech recognition, image processing, etc. [5–6]. In addition, simulations driven by multilayer neural networks make it possible to plan scenarios and analyze sensitivity, assisting manufacturers in making reasoned decisions about network configuration [7]. Robotics and automation integrated with multilayer neural networks for the improvement of operational efficiency in these facilities, provides higher level of adaptability and scalability [8–9]. In fact, integration of multilayer neural networks into manufacturing network design is transforming the industry by providing more manageable, flexible, and optimized operations, ultimately resulting in increased competitiveness and profitability [10].

Objective of the paper is to consider the manufacturing network design based on multilayer neural models; to carry out qualitative analysis of the dynamic behavior of the dynamic lattice model; to build the training data set using multivariate method for predicting nonlinear dynamics; to analyze and compare neural networks defined by corresponding architectures with linear and nonlinear outputs.

Materials and methods. *A. Lattice Model Design.* The following model is developed manufacturing network design problems arising from the transportation of raw materials between logistics sites on rectangular grid using lattice differential equations with delay.

The terminology of the model comes from [11]. The model is based on a number of assumptions. Assume the manufacturing network incorporates the logistic-production sites that are located at the nodes of squared lattice (i,j), $(i,j)=\overline{1,N}$. Let us accept that for t, at the given time, $V_{i,j}(t)$ is the resources used for manufacturing the product and currently located on the site, in turn $F_{i,j}(t)$ is the finished product that is produced and stored at (i,j).

The model takes into consideration the following parameters of the production and transportation processes for arbitrary logistics and production site (i,j):

1. Resources appear (can be «mined») within production site with probability $\beta > 0$.
2. The producing of the product unit requires $\gamma > 0$ units of the resource.
3. The use of resource is limited by coefficient $\delta_v > 0$, which makes it possible to strive for the level of throughput for $V_{i,j}(t)$.
4. Let us suppose that the transfer of resources can be possible from four neighboring sites $(i-1,j)$, $(i+1,j)$, $(i,j-1)$, $(i,j+1)$ (Fig. 1) with vertices $D_{i,j}^{i-1,j}\Delta^{-2}$, $D_{i,j}^{i+1,j}\Delta^{-2}$, $D_{i,j}^{i,j-1}\Delta^{-2}$, $D_{i,j}^{i,j+1}\Delta^{-2}$, where $D_{i,j}^{k,m} > 0$, $i,j,k,m=\overline{1,n}$ and $\Delta > 0$ is the distance between the sites.
5. The production can be rejected with probability $\mu_f > 0$.
6. As a result of delays and unconsidered consequences, we observe the increase in the cost of the resource required for products manufacturing to the probability level of. $\eta\gamma$.
7. Manufacturing tends to a certain throughput with probability $\delta_f > 0$.
8. Let us assume $\tau > 0$ is the time required to produce the unit of output.

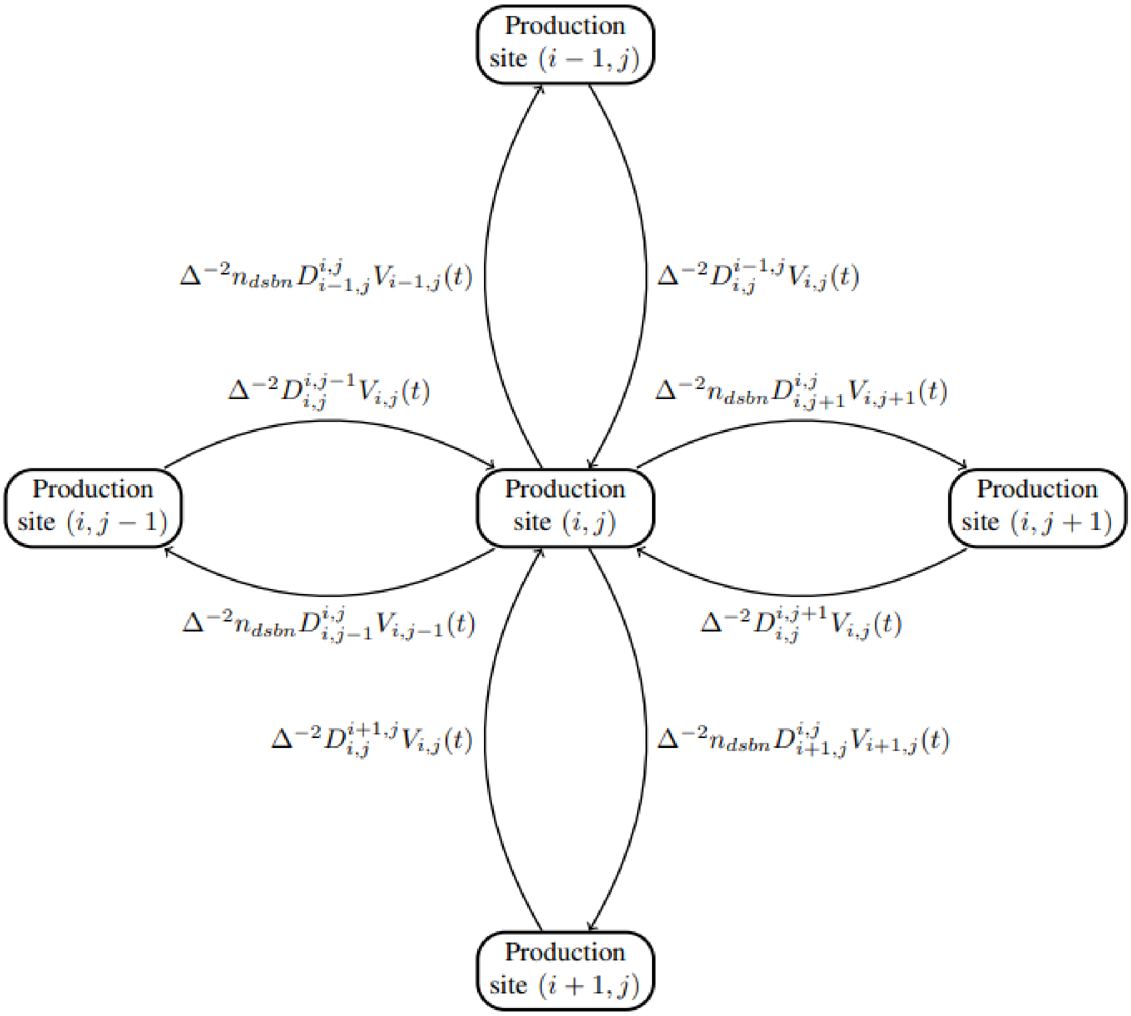

Figure 1. Square lattice presenting four neighboring logistic-manufacturing sites

Based on the above-mentioned assumptions, we consider the increase in the cost of resources in the site (i, j) during the time $\Delta t : \Delta V_{i,j}(t) = V_{i,j}(t+\Delta t) - V_{i,j}(t)$, taking into account the following assumptions:

a) increasing on the value $\beta V_{i,j}(t)\Delta t$ caused by «mining» of a new resource;

b) decreasing on the value $-\gamma F_{i,j}(t-\tau)V_{i,j}(t)\Delta t$, which is explained by the resources required for products manufacturing at site (i, j) at instant $(t-\tau)$;

c) increasing on the value $-\Delta\delta_v V_{i,j}(t-\tau)V_{i,j}(t)\Delta t$ due to resource throughput capacity.

The number of products in site (i, j) also depends on the distribution of raw materials between four neighboring sites, which is taken into account while calculating the spatial operator $\hat{S}\{V_{i,j}(t)\}\Delta t$ in the form (4).

Based on the above mentioned assumptions, the increase in the number of resources $\Delta V_{i,j}(t)$ in site (i, j) over a certain period of time can be written in the following form $\Delta V_{i,j}(t)\Delta t$.

Dividing the left and right sides of the whole equation by Δt and adding $\Delta t \to 0$, we get the equation for determining the amount of raw resources $V_{i,j}$, $t > 0$.

$$\frac{dV_{i,j}}{dt} = \beta V_{i,j}(t) - \gamma F_{i,j}(t-\tau)V_{i,j}(t) - \delta_V V_{i,j}(t-\tau)V_{i,j}(t) + \hat{S}\{V_{i,j}(t)\} \qquad (1)$$

The growth of products in site (i,j) for the period of time Δt is

$$\Delta F_{i,j}(t) = F_{i,j}(t+\Delta t) - F_{i,j}(t)$$

According to the assumptions stated above, the following analogs affect the change in product at the production site (i,j) for a period of time Δt:

a) reduce by value $-\mu_f F_{i,j}(t)\Delta t$ due to the number of defective products.

b) increase by the amount $\eta\gamma V_{i,j}(t-\tau)F_{i,j}(t)\Delta t$ determined by the resources required to produce the output unit at the given time $(t-\tau)$ (τ – the value of the delay (time spent on the production of the output unit));

c) decrease by value $-\delta_f F_{i,j}^2(t)\Delta t$, which is caused by the decrease in the speed of manufacturing new products δ_f due to their approaching to the saturation limit.

From the above mentioned reasons, the growth of products $\Delta F_{i,j}(t)$ in site (i,j) over period of time Δt can be written as follows $\Delta F_{i,j}(t) = -\mu_f F_{i,j}(t)\Delta t + \eta\gamma V_{i,j}(t-\tau)F_{i,j}(t)\Delta t - \delta_f F_{i,j}^2(t)\Delta t, t > 0$.

Dividing the left and right parts of the last equation by Δt with direction $\Delta t \to 0$, we get the equation for determining the quantity of production:

$$\frac{dF_{i,j}}{dt} = (-\mu_f + \eta\gamma V_{i,j}(t-\tau) - \delta_f F_{i,j}(t))F_{i,j}(t), t > 0 \qquad (2)$$

Operator (4) includes constant n_{dsbn} that describes possible imbalances between the input and output flows of raw materials.

Each site is affected by the resources produced by four neighboring sites – two sites in each direction, separated by the same distance Δ.

$$\hat{S}\{V_{i,j}\} = \begin{cases} \Delta^{-2}\left[D_{1,2}^{1,1}V_{1,2} + D_{2,1}^{1,1}V_{2,1} - n_{dsbn}D_{1,1}^{1,2}V_{1,1} - n_{dsbn}D_{1,1}^{2,1}V_{1,1}\right] & i,\ j = 1 \\ \Delta^{-2}\left[D_{2,j}^{1,j}V_{2,j} + D_{1,j-1}^{1,j}V_{1,j-1} + D_{1,j+1}^{1,j}V_{1,j+1} - n_{dsbn}D_{1,j}^{2,j}V_{1,j} - n_{dsbn}D_{1,j}^{1,j-1}V_{1,j} - n_{dsbn}D_{1,j}^{1,j+1}V_{1,j}\right] & i = 1,\ j \in \overline{2, N-1} \\ \Delta^{-2}\left[D_{1,N-1}^{1,N}V_{1,N-1} + D_{2,N}^{1,N}V_{2,N} - n_{dsbn}D_{1,N}^{1,N-1}V_{1,N} - n_{dsbn}D_{1,N}^{2,N}V_{1,N}\right] & i = 1,\ j = N \\ \Delta^{-2}\left[D_{i-1,N}^{i,N}V_{i-1,N} + D_{i+1,N}^{i,N}V_{i+1,N} + D_{i,N-1}^{i,N}V_{i,N-1} - n_{dsbn}D_{i,N}^{i-1,N}V_{i,N} - n_{dsbn}D_{i,N}^{i+1,N}V_{i,N} - n_{dsbn}D_{i,N}^{i,N-1}V_{i,N}\right] & i \in \overline{2, N-1},\ j = N \\ \Delta^{-2}\left[D_{N-1,N}^{N,N}V_{N-1,N} + D_{N,N-1}^{N,N}V_{N,N-1} - n_{dsbn}D_{N,N}^{N-1,N}V_{N,N} - n_{dsbn}D_{N,N}^{N,N-1}V_{N,N}\right] & i = N,\ j = N \\ \Delta^{-2}\left[D_{N-1,j}^{N,j}V_{N-1,j} + D_{N,j-1}^{N,j}V_{N,j-1} + D_{N,j+1}^{N,j}V_{N,j+1} - n_{dsbn}D_{N,j}^{N-1,j}V_{N,j} - n_{dsbn}D_{N,j}^{N,j-1}V_{N,j} - n_{dsbn}D_{N,j}^{N,j+1}V_{N,j}\right] & i = N,\ j \in \overline{2, N-1} \\ \Delta^{-2}\left[D_{N-1,1}^{N,1}V_{N-1,1} + D_{N,2}^{N,1}V_{N,2} - n_{dsbn}D_{N,1}^{N-1,1}V_{N,1} - n_{dsbn}D_{N,1}^{N,2}V_{N,1}\right] & i = N,\ j = 1 \\ \Delta^{-2}\left[D_{i-1,1}^{i,1}V_{i-1,1} + D_{i+1,1}^{i,1}V_{i+1,1} + D_{i,2}^{i,1}V_{i,2} - n_{dsbn}D_{i,1}^{i-1,1}V_{i,1} - n_{dsbn}D_{i,1}^{i+1,1}V_{i,1} - n_{dsbn}D_{i,1}^{i,2}V_{i,1}\right] & i \in \overline{2, N-1},\ j = 1 \\ \Delta^{-2}\left[D_{i-1,j}^{i,j}V_{i-1,j} + D_{i+1,j}^{i,j}V_{i+1,j} + D_{i,j-1}^{i,j}V_{i,j-1} + D_{i,j+1}^{i,j}V_{i,j+1} - n_{dsbn}D_{i,j}^{i-1,j}V_{i,j} - n_{dsbn}D_{i,j}^{i+1,j}V_{i,j} - n_{dsbn}D_{i,j}^{i,j-1}V_{i,j} - n_{dsbn}D_{i,j}^{i,j+1}V_{i,j}\right], & i,\ j \in \overline{2, N-1} \end{cases} \qquad (4)$$

The boundary condition $V_{i,j}=0$ for indices array $i,j=0,N+1$ is used.

B. Multivariate parameters method. The method was developed in [12] and applied in [13–14] to various models of dynamical systems. Here, it was used for (1)–(4) for the purpose of regression based on model parameters. The method includes a number of steps.

1. Setting up parameter areas D, Δ, n, β, γ, δ_v, δ_f, η, μ_f, τ, namely $D_{\min}\le D\le D_{\max}$, $\Delta_{\min}\le\Delta\le\Delta_{\max}$, $n_{\min}\le n\le n_{\max}$, $\beta_{\min}\le\beta\le\beta_{\max}$, $\gamma_{\min}\le\gamma\le\gamma_{\max}$, $\delta_{v,\min}\le\delta_v\le\delta_{v,\max}$, $\delta_{f,\min}\le\delta_f\le\delta_{f,\max}$, $\eta_{\min}\le\eta\le\eta_{\max}$, $\mu_{f,\min}\le\mu_f\le\mu_{f,\max}$, $\tau_{\min}\le\tau\le\tau_{\max}$

2. Construction of tuples training set $\Pi=(D, \Delta, n, \beta, \gamma, \delta_v, \delta_f, \eta, \mu_f, \tau, C)^{\mathrm{T}}$,

where C is class attribute. Here, we consider three values for the classes corresponding to stable site, limit cycle, or chaotic attractor, respectively. Construction of the training dataset is based on modeling the parameter values using multivariate prediction method.

The whole model is defined by equations (1), (2) and initial functions:

$$V_{i,j}(t)=V^0_{i,j}(t)\in 0, F_{i,j}(t)=F^0_{i,j}(t)\ge 0,\\ t\in[-\tau,0), V_{i,j}(0), F_{i,j}(0)>0 \tag{3}$$

For square array, the following operator of spatially variable discretely distributed diffusion is used [15].

3. Setting up the values of the parameters for obtaining trajectories $X(\Pi)=(X_0,X_1,...,X_{t_{\max}})$, where

$$X=(V_{1,1},V_{1,2},...,V_{1,N},V_{2,1},V_{2,2,},...,V_{2,N},...,V_{N,N},\\ F_{1,1},F_{1,2},...,F_{1,N},F_{2,1},F_{2,2,},...,F_{2,N},...,F_{N,N})^{\mathrm{T}}$$

$t_{\max}$ is maximum considered moment of time.

4. Obtaining the attribute – the largest Lyapunov exponent.

5. Application of the backward error propagation algorithm for adding model parameters and use of 10-fold cross-validation to evaluate the performance of the neural network. Return the neural network with the highest efficiency.

Investigation results. Specified parameter areas $D_{\min}=0.009$, $\Delta_{\min}=0.270$, $n_{\min}=0.810$, $\beta_{\min}1.800$ $\gamma_{\min}=1.800$, $\delta_{v,\min}=0.450$, $\delta_{f,\min}0.450$, $\eta_{\min}0.360$, $\mu_{f,\min}=0.900$, $\tau_{\min}=0.020$, $D_{\max}=0.011$, $\Delta_{\max}=0.330$, $n_{\max}=0.990$, $\beta_{\max}=2.200$, $\gamma_{\max}=2.200$, $\delta_{v,\max}=0.550$, $\delta_{f,\max}=0.550$, $\eta_{\max}=0.440$, $\mu_{f,\max}=1.100$, $\tau_{\max}=0.280$

As the result of the experiment, we constructed the training dataset with 1000 tuples.

Neural networks defined by given architectures were considered.

1 One hidden layer with 8 neurons and linear outputs.

The resulting optimal model for the root mean square error is shown in Figure 2.

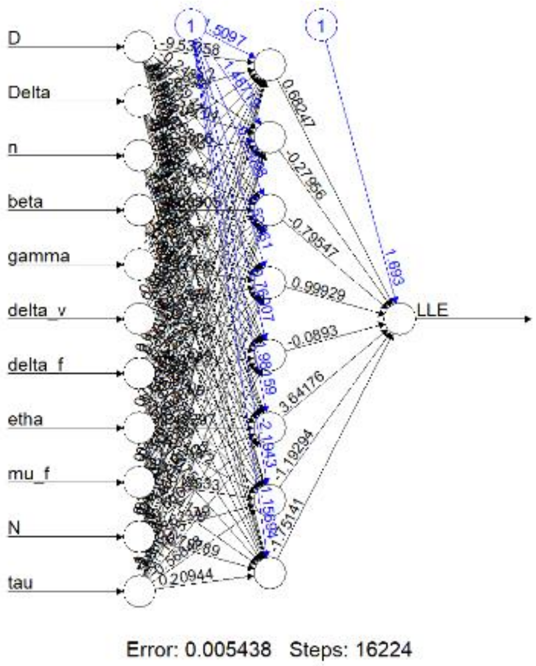

Figure 2. The optimal model in terms of root mean square error, for one hidden layer with the number of neurons 8 and linear outputs

The coincidence of expected and predicted values for the optimal model is shown in Figure 3.

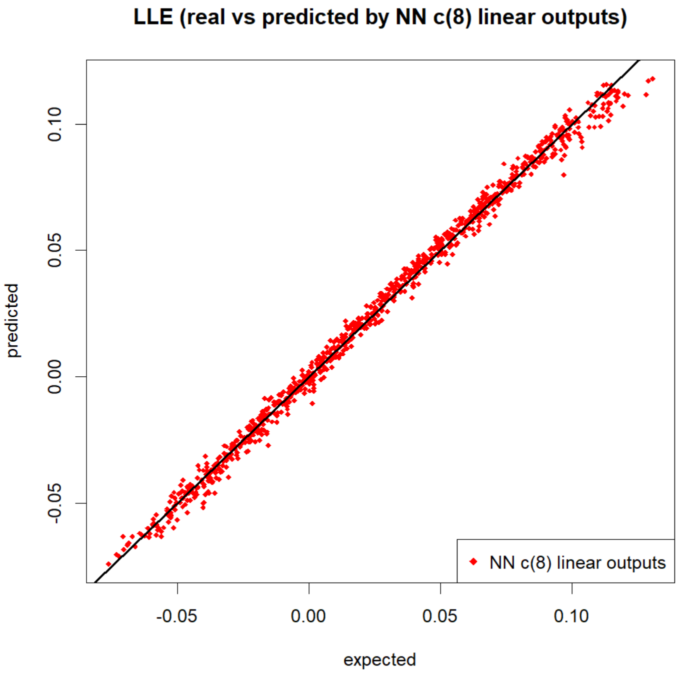

Figure 3. Coincidence of expected and predicted values for the optimal model, where there is one hidden layer with the number of neurons 8 and linear outputs

2. One hidden layer with 8 neurons and nonlinear outputs.
The resulting optimal model for the root mean square error is shown in Figure 4.

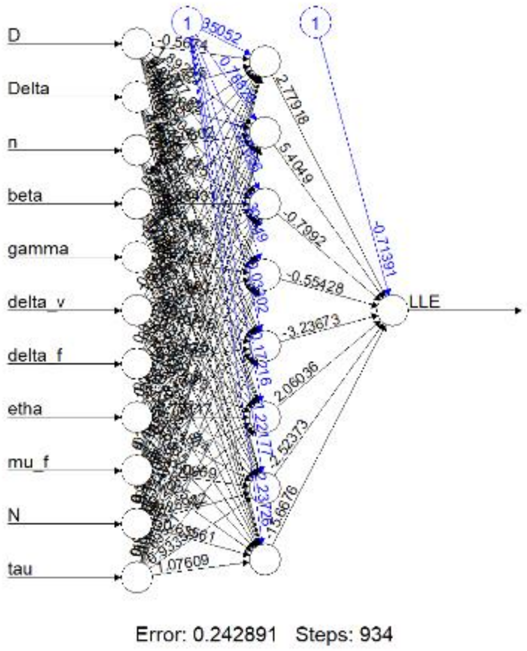

Figure 4. The optimal model for the root mean square error, for one hidden layer with the number of neurons 8 and non-linear outputs

The coincidence of expected and predicted values for the optimal model is shown in Figure 5.

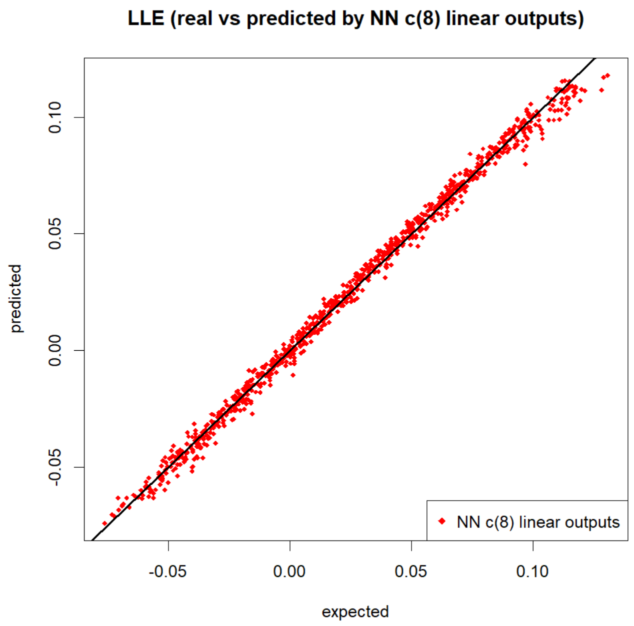

Figure 5. Coincidence of expected and predicted values for the optimal model, where there is one hidden layer with the number of neurons 8 and non-linear outputs

3. Two hidden layers with 8 and 5 neurons and linear outputs.
The resulting optimal model for the root mean square error is shown in Figure 6.

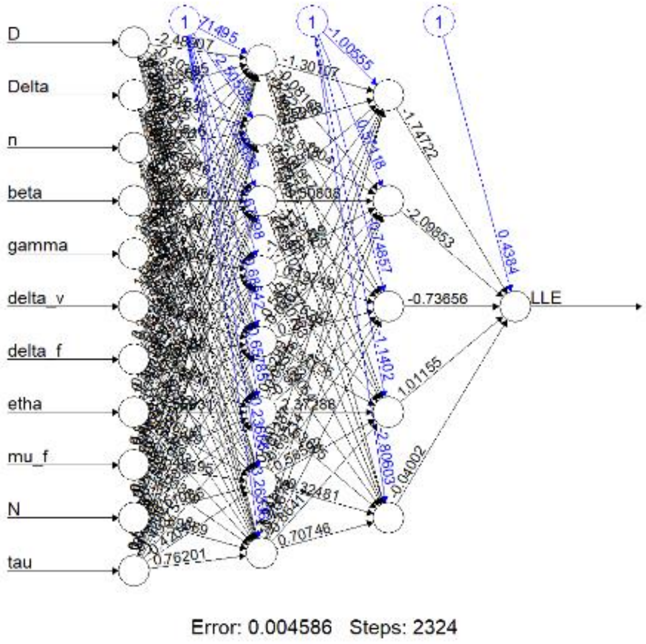

Figure 6. The optimal model with respect to the root mean square error, for two hidden layers with the number of neurons 8 and 5 and linear outputs

The coincidence of expected and predicted values for the optimal model is shown in Figure 7.

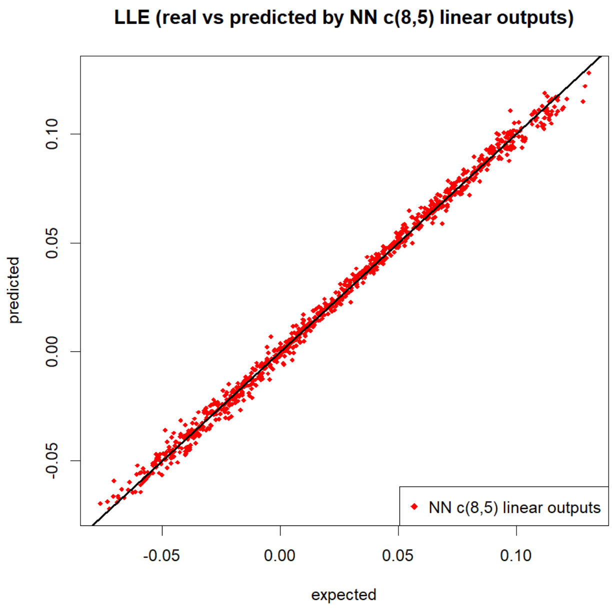

Figure 7. Coincidence of expected and predicted values for the optimal model, where there are two hidden layers with the number of neurons 8 and 5 and linear outputs

4. Two hidden layers with 8 and 5 neurons and nonlinear outputs.
The resulting optimal model for the root mean square error is shown in Figure 8.

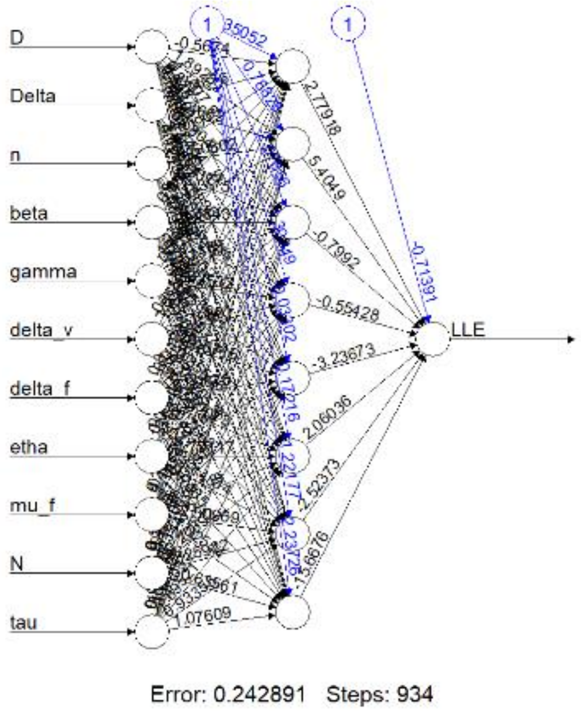

Figure 8. The optimal model for the root mean square error obtained for two hidden layers with the number of neurons 8 and 5 and nonlinear outputs

The coincidence of expected and predicted values for the optimal model is shown in Figure 9.

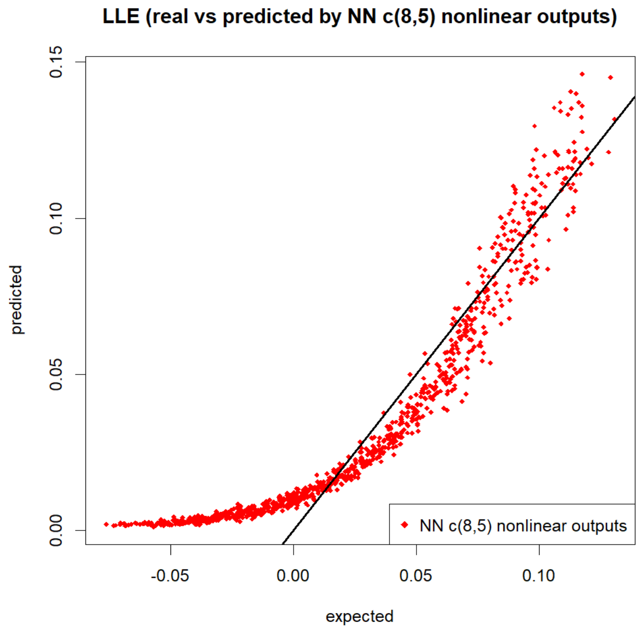

Figure 9. Coincidence of expected and predicted values for the optimal model, where there are two hidden layers with the number of neurons 8 and 5 and nonlinear outputs

The constructed optimal models were compared with respect to two criteria. Firstly, comparison of the optimal models of the above mentioned architectures in terms of the root mean square error is shown in Fig. 10.

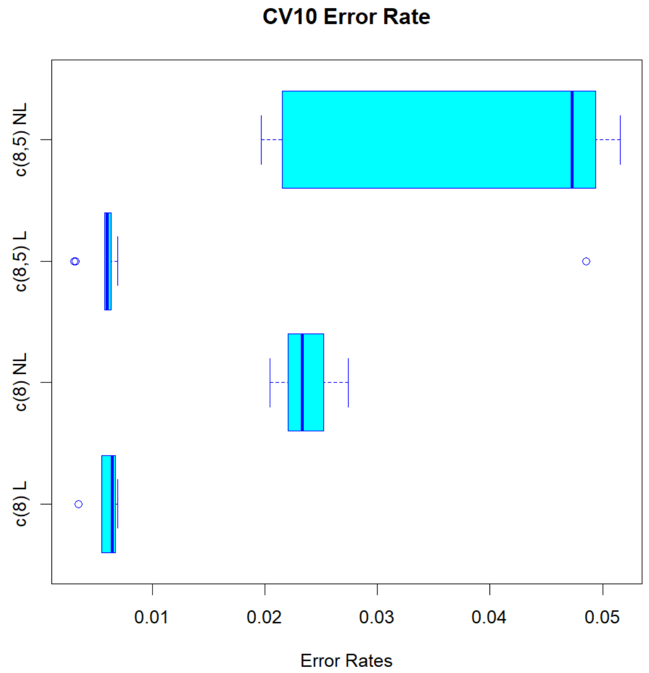

Figure 10. Comparison of the optimal models of the above-mentioned architectures with respect to root mean square error

Secondly, comparison of the optimal models in terms of learning time is shown in Figure 11.

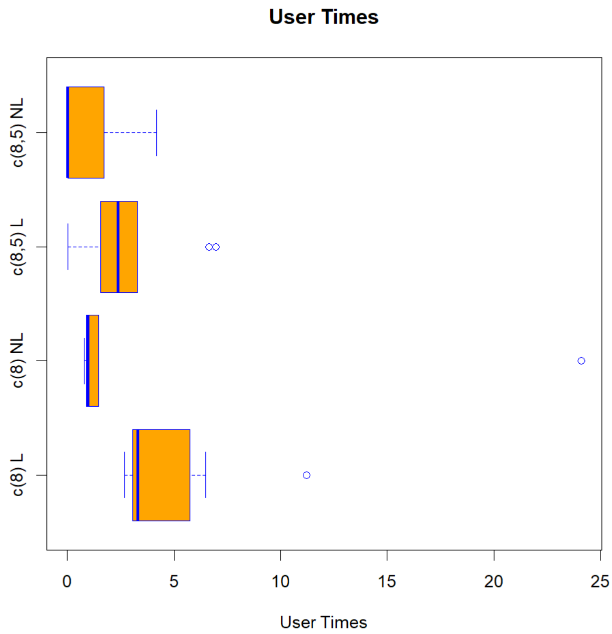

Figure 11. Comparison of optimal models regarding learning time

Conclusions. Analyzing the neural network architectures, we can see that architectures with linear outputs showed much better compliance of expected and predicted values (Figures 3, 7). Architectures with nonlinear outputs show lower accuracy and higher deviation compared to linear ones (Figures 5, 9). We compared the optimal models of the above mentioned architectures in terms of root mean square error (Figure 10), and concluded that two-layer architecture with linear outputs has the best error, but at the same time it has the ability to fly out accidentally and make a large error, so it is not the most accurate one. The worst accuracy rates are shown by the two-layer architecture with nonlinear outputs, and the complexity of this architecture only worsens the indicators. The single-layer architecture with nonlinear outputs did not show high accuracy rates as well. Therefore, the best accuracy is shown by the single-layer architecture with linear outputs. Comparisons of the optimal models in terms of leatning time showed that architectures with nonlinear outputs have poor accuracy but learn quickly, and architectures with linear outputs take longer time to learn but have smaller errors.

References

1. Jelena Milisavljevic-Syed, Janet K. Allen, Sesh Commuri, Farrokh Mistree, Design of networked manufacturing systems for Industry 4.0, Procedia CIRP, vol. 81, 2019, pp. 1016–1021. https://doi.org/10.1016/j.procir.2019.03.244
2. Mourtzis D., Doukas M., Psarommatis F., Manufacturing Network Design for Mass Customisation using a Genetic Algorithm and an Intelligent Search Method, Procedia CIRP, vol. 7, 2013, pp. 37–42 https://doi.org/10.1016/j.procir.2013.05.007
3. Weipeng Cao, Xizhao Wang, Zhong Ming, Jinzhu Gao,A review on neural networks with random weights, Neurocomputing, Volume 275, 2018, pp. 278–287, ISSN 0925-2312. https://doi.org/10.1016/j.neucom.2017.08.040
4. Tomasz Dudek, Tygran Dzhuguryan, Justyna Lemke, Sustainable production network design for city multi-floor manufacturing cluster, Procedia Computer Science, vol. 159, 2019, pp. 2081–2090. https://doi.org/10.1016/j.procs.2019.09.381
5. Xiaona Song, Peng Sun, Shuai Song, Vladimir Stojanovic,Event-driven NN adaptive fixed-time control for nonlinear systems with guaranteed performance, Journal of the Franklin Institute, vol. 359, issue 9, 2022, , pp. 4138–4159, ISSN 0016-0032. https://doi.org/10.1016/j.jfranklin.2022.04.003
6. Fei Han, Jing Jiang, Qing-Hua Ling, Ben-Yue Su, A survey on metaheuristic optimization for random single-hidden layer feedforward neural network, Neurocomputing, vol. 335, 2019, pp, 261–273, ISSN 0925-2312. https://doi.org/10.1016/j.neucom.2018.07.080
7. Cristiana L. Lara, David E. Bernal, Can Li, Ignacio E. Grossmann, Global optimization algorithm for multi-period design and planning of centralized and distributed manufacturing networks, Computers & Chemical Engineering, vol. 127, 2019, pp. 295–310. https://doi.org/10.1016/j.compchemeng.2019.05.022
8. Jiya Yu, Jiye Zhang, Aijing Shu, Yujie Chen, Jianneng Chen, Yongjie Yang, Wei Tang, Yanchao Zhang,Study of convolutional neural network-based semantic segmentation methods on edge intelligence devices for field agricultural robot navigation line extraction, Computers and Electronics in Agriculture, vol. 209, 2023, 107811, ISSN 0168-1699. https://doi.org/10.1016/j.compag.2023.107811
9. Ahmed Temtam, Abdraouf Abusoua, Khaled Benyounis, Abdalmonem Tamtam,Use of neural networks and artificial intelligence tools for modeling, characterization, and predicting in material engineering, Reference Module in Materials Science and Materials Engineering, Elsevier, 2023, ISBN 9780128035818. https://doi.org/10.1016/B978-0-323-96020-5.00088-1
10. Yohanes Kristianto, Angappa Gunasekaran, Petri Helo, Maqsood Sandhu, A decision support system for integrating manufacturing and product design into the reconfiguration of the supply chain networks, Decision Support Systems, vol. 52, iss. 4, 2012, pp. 790–801. https://doi.org/10.1016/j.dss.2011.11.014
11. Alexandra Birkmaier, Bernhard Oberegger, Andreas Felsberger, Gerald Reiner, Wilfried Sihn, Towards a robust digital production and logistics network by implementing flexibility measures, Procedia CIRP, vol. 104, 2021, pp. 1310–1315. https://doi.org/10.1016/j.procir.2021.11.220
12. Martsenyuk V. and Klos-Witkowska A. “Computation Model of Cyber-Physical Immunosensor System”, in IEEE Access, vol. 7, pp. 62325–62337, 2019. https://doi.org/10.1109/ACCESS.2019.2915946
13. Koch Y., Wolf T., Sorger P., Eils R., Brors B. Decision-tree based model analysis for efficient identification of parameter relations leading to different signaling states (2013) PLoS ONE, 8 (12), art. no. e82593. https://doi.org/10.1371/journal.pone.0082593

14. Martsenyuk V., Warwas K., Augustynek K., Klos-Witkowska A., Karpinskyi V., Klymuk N., Mayhruk Z. On multivariate method of qualitative analysis of Hodgkin-Huxley model with decision tree induction (2016) International Conference on Control, Automation and Systems, 0, art. no. 7832365, pp. 489–494. https://doi.org/10.1109/ICCAS.2016.7832365
15. Lyapandra A. S., Martsenyuk V. P., Gvozdetska I. S., Szklarczyk R., Rajba S. A. Qualitative analysis of compartmental dynamic system using decision-tree induction (2015) Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2015, 2, art. no. 7341391, pp. 688–692. https://doi.org/10.1109/IDAACS.2015.7341391

УДК004.8

БАГАТОВАРІАЦІЙНИЙ МЕТОД ПРОГНОЗУВАННЯ НЕЛІНІЙНОЇ ДИНАМІКИ ВИРОБНИЧОЇ МЕРЕЖІ НА ОСНОВІ БАГАТОШАРОВИХ НЕЙРОННИХ МОДЕЛЕЙ

Василь Марценюк[1,2]; Наталія Кіт[2]

[1]Університет в Бельсько Бяла, Бельсько Бяла, Польща
[2]Тернопільський національний технічний університет імені Івана Пулюя, Тернопіль, Україна

Резюме. *Розглянуто проектування виробничої мережі на основі багатошарових нейронних моделей. Проектування виробничої мережі має вирішальне значення, оскільки визначає оптимальне розміщення виробничих і логістичних потужностей, впливає на ефективність витрат, рівень обслуговування клієнтів та загальну конкурентоспроможність на світовому ринку. Багатошарові нейронні мережі відіграють важливу роль у цьому процесі, використовуючи передові алгоритми, моделі машинного навчання та методи оптимізації для аналізу величезної кількості даних. Зосереджено увагу на якісному аналізі динамічної поведінки, динамічної решітчастої моделі. Модель включає константи швидкості та початкові умови, що впливають на траєкторії моделі, які можна класифікувати як стабільний вузол, граничний цикл або хаотичний атрактор. Проблему якісної поведінки моделі прагнемо вирішити як проблему багатошарових нейронних моделей. Для побудови навчального набору даних використано багатоваріаційний метод прогнозування нелінійної динаміки. Проаналізовано та проведено порівняння нейромереж, які задані відповідними архітектурами, з лінійними та нелінійними виходами. В результаті аналізу виявлено, що архітектури з лінійними виходами демонструють кращу відповідність між очікуваними та прогнозованими значеннями. Архітектури з нелінійними виходами, незважаючи на свою складність, проявляють меншу точність та більше відхилення порівняно з лінійними. Найкращу точність показує одношарова архітектура з лінійними виходами, хоча двошарова архітектура з лінійними виходами має найменшу середньоквадратичну похибку. Архітектури з нелінійними виходами характеризуються швидшим часом навчання, але й поганою точністю, в той час як архітектури з лінійними виходами вимагають більше часу для навчання, але мають менші похибки. Отримані результати вказують на важливість вибору правильної архітектури нейронної мережі залежно від поставлених завдань та вимог до точності та часу навчання моделі.*

Ключові слова*: проектування виробничих мереж, решітчаста модель, якісний аналіз, багатоваріаційний метод прогнозування, багатошарові нейронні моделі.*

https://doi.org/10.33108/visnyk_tntu2024.02.039 *Отримано 13.02.2024*