Investigation of Hybrid Neo-fuzzy Neural Networks in the Problem of Pandemic Forecasting

Yuriy Zaychenko[a], Helen Zaichenko[a] and Galib Hamidov[b]

[a] *Igor Sikorsky Kyiv Polytechnic Institute, Peremohy av., 37, Kyiv, 03056, Ukraine*
[b] *Information Technologies Department, Azershiq, K.Kazimzade 20, Baku, AZ12008, Azerbaijan*

Abstract

The problem of covid-19 forecasting was considered and investigated. Review of different models and methods of pandemic forecasting are presented. For short-term forecasting indicators of covid-19 the application new class neural networks – hybrid neo-fuzzy networks based on GMDH is suggested. The application of GMDH enables to construct the structure of hybrid network and accelerate the speed of learning neural weights. The experimental investigations were carried out during which the optimal parameters of hybrid network were found sliding window size, forecasting interval and network architecture. The efficiency of hybrid neo-fuzzy network in the pandemic forecasting problem was estimated and compared with Back Propagation neural network.

Keywords [1]

Covid 19 forecasting, hybrid neo-fuzzy network, GMDH, network parameters and structure optimization

1. Introduction

The global problem of the beginning of the XXI century is the spread of the infectious disease covid-19. The pandemic affects not only human life and health, but also the global economy as a whole. In order to take the necessary measures to curb the development of a pandemic and to preserve the life and health of the population, it is extremely important to develop and apply effective models and methods for predicting the development and spread of a pandemic. Forecasting pandemic processes is a difficult task. There are many factors that affect the course of the disease: the density of population in a country, the age of the nation, the state of immunity, the environment, the time of year, the social status of the individual. Information on the spread of coronavirus in Ukraine and around the world is presented on the website of the World Data Center "Geoinformatics and Sustainable Development" [1]. The prevalence of covid-19 can be assessed as a whole throughout Ukraine and separately by region. The dynamics of mortality, the number of confirmed cases, suspected, dead, recovered are presented.

Official information on the spread of covid-19 in Ukraine is provided by the Office of the National Security and Defense Council of Ukraine in the form of the Coronavirus Epidemic Monitoring System [2], which has the form of a board with a map and tables. It contains convenient filters for working with data and comparing the epidemiological situation in Ukraine and the world.

Consider the main classes of models that are currently used to predict the development of the Covid-19. One of the simplest and most frequently used models is the model based on the Euclidean network SIR [3], which is often used in the prediction of epidemiological processes. The entire population of the country is taken into account, which is divided into groups:

S – Susceptible; *I* – Infected; *R* – Removed.

It's assumed that all population size is left constant. After disease a person comes from class of "susceptible" to the class of "infected" and after that to the class "recovered". Thus:

$$S + I + R = 1. \tag{1}$$

II International Scientific Symposium «Intelligent Solutions» IntSol-2021, September 28–30, 2021, Kyiv-Uzhhorod, Ukraine
EMAIL: zaychenkoyuri@ukr.net (A. 1); syncmaster@bigmir.net (A. 2); galib.hamidov@gmail.com (A. 3);
ORCID: 0000-0001-9662-3269 (A. 1); 0000-0002-4630-5155 (A. 2); 0000-0002-9942-1950 A. 3)

In a subsequent study [4] to predict morbidity and mortality in the United States for long-term, the researchers used the SEIR model to simulate possible trajectories of severe acute coronavirus syndrome and the effects of non-drug interventions at the state level from September 22, 2020 to February 28, 2021. Using the SEIR model and forecasts of critical covariances (seasonality of pneumonia, mobility, per capita testing and use of masks), scenarios of social distancing and the level of mask use were assessed. Due to its simple structure and speed of use ARIMA model was widely used in forecasting [5]. The model takes into account changing trends, periodic changes and random distortions in time series. The forecast made for Italy, France and Spain showed that covid-19 did not show seasonal patterns. The best models for Italy, Spain and France were ARIMA (0.2,1), ARIMA (1,2,0) and ARIMA (0,2,1), respectively.

Deep learning models are widely used for Covid-19 forecasting with good results [6]. In particular, simple recurrent networks (RNN), long-term short memory (LSTM), bidirectional LSTM (BiLSTM), gated recurrent units (GRU) and variational autoencoder (VAE) were used to predict covid-19. In a study of six countries (Italy, Spain, France, China, the United States and Australia), the best result was shown by the variation autoencoder (VAE). Data since January 22 to May 31, 2020 were taken for the study. Initially, each model was trained on a training sample.

Comparing the forecasting results, conclusion was made that VAE model showed the best result in almost all countries except Italy. The average absolute error for Italy is 5.90%, Spain - 2.19%, France - 1.88%, China - 0.128%, Australia - 0.236% and the United States - 2.04%.

The study [7] used Back Propagation Neural Network based on a "sliding window" mechanism for short-term forecasting of four and five days. The absolute average error in Ukraine is 0.65%, and in Kyiv - 1.33%. The forecasting error appeared to be insignificant.

Artificial intelligence methods have been used in the research [8]. The proposed method tracks time differences in outbreaks in different countries. Data were collected about 250 countries. The neural network used is a topological autoencoder. Time series of confirmed cases in many countries are submitted for entry. The system develops a two-dimensional map that projects the topological structures of dynamics. When forecasting, the system uses a map to find a reference country with similar dynamics, and then trains the LSTM neural network. The network is taught "with a teacher" or "without a teacher", or by two methods. Forecasting was carried out for a short period of three days. Comparing the forecasting results, one can see that in different cases the results vary significantly. Thus, for Brazil the error reached 3.66%, for Singapore - 2.39%, for New Zealand - 3.72%.

In the investigation [9], a generalized logistic growth model (GLM), a Richards model and a sub-epidemic model were used for short-term forecasting of coronavirus. The study was conducted in China to predict new cases in different provinces. For example, for Guangdong, all three models showed similar results for the five-day period. The ten-day forecast suggests a small increase in the incidence. The standard error of forecasting in Guangdong for the GLM model is 36.95-37.63%; for Richard's model - 35.87-36.95%; for the sub-epidemic model - 35.68-37.46%.

Last year’s new class of deep learning networks- hybrid neuro-fuzzy networks [21] and hybrid neo-fuzzy neural networks [22] were developed and investigated for solution of various problems: forecasting in macroeconomy and financial sphere, pattern recognition, etc. They are based on self-organization method GMDH and unlike conventional neural networks enable to train not only weights, but the network structure as well. Besides, due to a small number of tunable parameters the training process is accelerated. So, it presents the great interest to apply the hybrid neural networks to the problem of forecasting the indicators of covid -19 in the process of evolution.

The goal of this paper is to investigate the hybrid neo-fuzzy networks for short-term forecasting of pandemic indicators in Ukraine, estimate their efficiency and compare with alternating method – neural network Back Propagation.

2. Hybrid GMDH-Neuro-Fuzzy System Architecture construction

The general architecture of the hybrid GMDH-network is shown in Fig. 1. The structure hybrid network is constructed using the principle of self-organization in the following evolution process. To the input system layer $(n \times 1)$-dimensional vector of input signals $x = (x_1,\ x_2,\ \dots,\ x_n)^T$ is fed.

Then this signal is fed to the first hidden layer that contains $n_1 = c_n^2$ nodes-neurons, each of which has only two inputs. At the node outputs $N^{[1]}$ of the first hidden layer the output signals $\hat{y}_l^{[1]}$, $l = 1, 2, \ldots, 0{,}5n(n-1) = c_n^2$ are formed. Then these signals are fed to the selection block of the first hidden layer $SB^{[1]}$, that selects among the output signals $\hat{y}_l^{[1]}$ n_1* most precise by accepted criterion (mostly by the mean squared error $\sigma^2_{y_l^{[1]}}$).

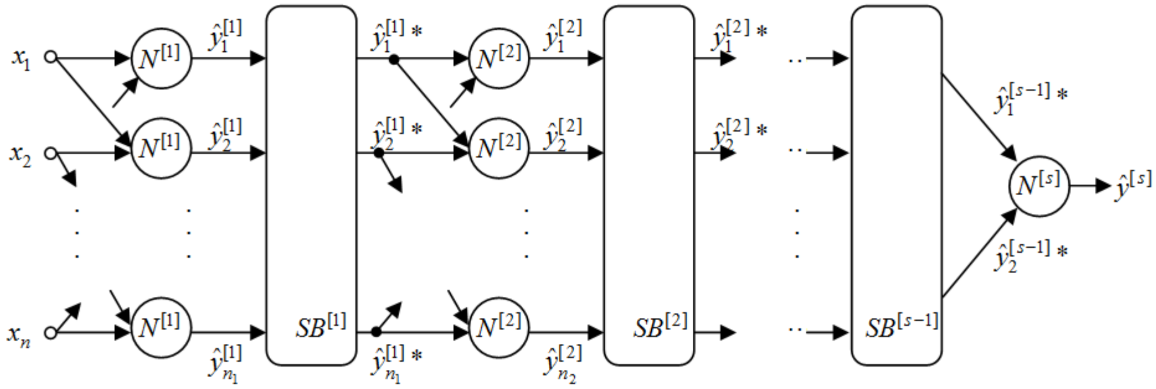

Figure 1: Evolving GMDH-system.

From these n_1* best outputs of the first hidden layer $\hat{y}_l^{[1]} * n_2$ pairwise combinations $\hat{y}_l^{[1]} *, \hat{y}_p^{[1]} *$ are formed, that are fed to the second hidden layer, formed by neurons $N^{[2]}$.

Among the signals of this layer $\hat{y}_l^{[2]}$ the selection block $SB^{[2]}$ selects F best neurons by accuracy (e.g., by value of $\sigma^2_{y_l^{[2]}}$) if the best signal of the second layer is better than the best one of the first hidden layer $\hat{y}_1^{[1]} *$. Other hidden layers forms signal similarly to the second layer.

The system evolution process continues until the best signal of the selection block $SB^{[s+1]}$ would be worse than the best signal of the previous sth layer, that is $\sigma^2_{y_l^{[s+1]}} > \sigma^2_{y_l^{[s]}}$.

Then we return to the previous layer and choose its best node neuron $N^{[s]}$ in order to form the system output signal $\hat{y}^{[s]}$. And moving from this neuron (node) along its connections backward and sequentially passing all previous layers we determine the structure of GMDH-neuro-fuzzy network. As nodes of hybrid fuzzy network neo-fuzzy neurons with inputs are used [20]. Their main advantage is the lack of necessity to train membership functions and only weights of rules are adapted during the learning. It should be stressed that we obtain not only optimal network structure but also well-trained network due to the GMDH algorithm. Besides, since the training is performed sequentially layer by layer the problems of high dimensionality as well as vanishing or exploding gradient are absent. This is very important for deep learning networks.

3. Software and initial data description

The programming language was chosen on the basis of the following criteria: ease of performing mathematical calculations, prevalence, cross-platform and purity of syntax. The most suitable for these criteria was the Python programming language [14, 15], which has gained great popularity in recent years. It is worth noting that the Python language gained popularity at a time when developers began to actively use neural networks to solve various problems, resulting in a large number of libraries and modules that simplify mathematical calculations or implement the logic of individual algorithms.

The development environment was chosen in terms of the cost of the license, the ease of writing software code and the availability of extensions to optimize performance. As a result, the PyCharm environment was chosen [15]. It is also important that PyCharm is operating system independent and has useful extensions that help reduce code writing time and control development. Extensions can be installed as needed or run without their use.

Scikit learn was chosen as a library for machine learning [18]. It is the most common for machine learning problems. This library allows you to work with tensors, perform complex mathematical operations for the scientific and technical field, provides tools for visualization of data processing and analysis.

The SciKit-Fuzzy library [19], which is essentially a set of algorithms, was chosen to work with fuzzy logic. The benefit of this library is that with its use it is possible to use a set of common membership functions, to introduce fuzzy, to perform operations of clarity.

Thus, the main tools for creating a software product that are free and popular were chosen.

The program consists of three main modules:

Module for working with data sets and auxiliary functions

Module for working with GMDH neo fuzzy

Module for working with Back Propagation

Statistical data is processed in several successive stages: obtaining data from the resource, preparing and saving a file with time series, generating graphs based on the obtained time series and creating data sets for training. The API of the National Security and Defense Council of Ukraine was chosen as a resource for obtaining statistical data [20]. When contacting this service, it is possible to obtain detailed information about the level of COVID-19, for forecasting the following indicators were selected: the number of confirmed cases, recovered and dead. Values of these indicators for the period from March 12, 2020 to April 19, 2021 were used.

Thus, absolute indicators were obtained, and a graph of their dynamics was constructed on the time interval and presented in the Fig. 2.

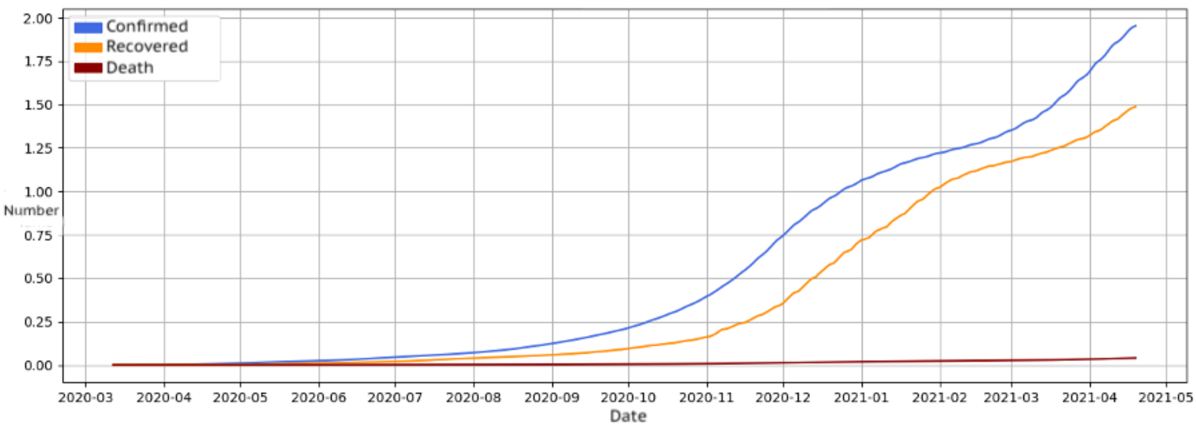

Figure 2: Flow chart of absolute values of covid-19 indicators evolution in Ukraine

The daily increase of indicators was also chosen for forecasting, for which the necessary requests to the service were created, the obtained data were saved, and the corresponding flow chart was constructed (Fig. 3).

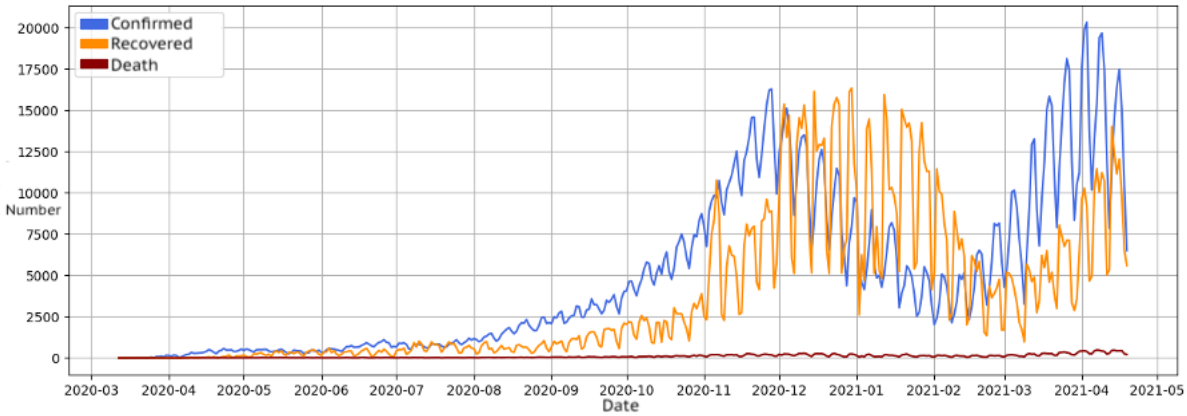

Figure 3: Daily increase of COVID-19 indicators values in Ukraine

Before training, the data were normalized, i.e. subjected to some transformation, so that it was possible to apply mathematical algorithms to work with them.

After training neural networks, as well as after forecasting, it is possible to generate reports that contain detailed information about the processes that took place and the results of work. Such reports include a table comparing forecast values with real ones and a calculated criterion MAPE.

4. Experimental investigations of hybrid neo-fuzzy network

To obtain the most accurate prediction, the optimal neural network parameters were determined, at which the MAPE values were the best. For this goal the following parameters were successively changed: membership function (bell-shaped, Gaussian, sigmoid, triangular), number of inputs (3, 4, 5, 6) and short-term forecast period (5, 6, 7, 8). It should be noted that the number of inputs is a small number, because it depends on the size of the neural network, which affects the computational costs of training. Determination of optimal parameters was performed for absolute values and daily growth of indicators.

4.1. Investigations of membership functions influence

In the first experiment the influence of membership functions on prediction accuracy was explored. The corresponding results for the absolute values of the number of confirmed cases, is presented in the Table 1. As it follows the best value of MAPE was found when using the Gaussian function (1). Bell-shaped and sigmoidal functions showed almost identical results.

Table 1
MAPE values for different membership functions in predicting the absolute values of the number of confirmed cases

Membership functions	MAPE
Bell-wise	3.15
Gaussian	2.37
sigmoidal	3.21
triangular	4.93

In the Table 2 MAPE criteria for prediction absolute values of the number of recovered and in the Table 3 – for prediction number of deaths are shown. As it follows the most accurate forecast was achieved with the bell-wise membership function.

The value of MAPE was the lowest when using the sigmoidal membership functions to predict the absolute values of the number of deaths (Table 3). While forecasting the daily increase in the number of confirmed cases, the best MAPE value was obtained using the Gaussian function (Table 4).

The forecast of the daily increase in the number of recovered was the most accurate for the triangular membership function (Table 5). The best option is also to use the Gaussian function in this case and the worst results were shown by the sigmoidal function.

Table 2
MAPE values for different membership functions in predicting the absolute values of the number of recovered

Membership functions	MAPE
Bell-wise	3.79
Gaussian	4.45
sigmoidal	5.17
triangular	4.79

Table 3
Values of MAPE for different membership functions in predicting the absolute values of the number of deaths

Membership functions	MAPE
Bell-wise	7.83
Gaussian	8.42
sigmoidal	7.71
triangular	8.14

Table 4
MAPE values for different membership functions in predicting the daily increase in the number of confirmed cases

Membership functions	MAPE
Bell-wise	49.05
Gaussian	31.85
sigmoidal	51.37
triangular	33.79

Table 5
MAPE values for different membership functions in predicting the daily increase in the number of recovered

Membership functions	MAPE
Bell-wise	44.39
Gaussian	40.12
sigmoidal	50.88
triangular	38.82

To predict the daily increase in the number of deaths, the lowest value of MAPE was for the triangular membership function, but the bell-shaped function also showed not much worse results. So, after these experiments the adequate membership functions were determined to predict all indicators of covid-19.

4.2. Investigation of optimal number of inputs

In the next series of experiments the investigation of optimal inputs number for forecasting covid-19 indicators were performed. The results of experiments- dependence MAPE values on inputs number for absolute number of confirmed cases are presented in the table 6, for absolute number of deaths- in the table 7 and for absolute number of recovered- in the table 8.

Table 6
MAPE values dependence on number of inputs when predicting the absolute values of confirmed cases numbers

Number of inputs	MAPE
3	2.37
4	2.35
5	4.16
6	4.27

Table 7
MAPE values when changing the number of inputs when predicting the absolute values of the number of recovered

Number of inputs	MAPE
3	4.84
4	4.57
5	3.72
6	3.41

Table 8
MAPE values dependence on number of inputs when predicting the absolute values of the number of deaths

Number of inputs	MAPE
3	8.12
4	7.84
5	7.67
6	7.26

The MAPE criterion turned to be the best for 6 inputs when forecasting the absolute values of the number of recovered and number of deaths (Table 7, 8).

In the next experiments the investigations of dependence of inputs number on forecasting accuracy for daily increase of covid-19 indicators were performed. The corresponding results are presented in the Table 9.

In the study of forecasts of daily increase in the number of confirmed cases, the best value of MAPE was obtained for 4 inputs, but for 3 and 6 inputs, the forecast was also satisfactory (Table 10).

Table 9
The value of MAPE when changing the number of inputs in predicting the daily increase in the number of confirmed cases, deaths and recovered

Number of inputs	MAPE number of confirmed cases	MAPE for number of deaths	MAPE for number of recovered
3	31.85	63.76	38.82
4	22.14	61.98	37.72
5	45.96	62.44	28.31
6	32.78	59.94	21.68

For the daily increase in the number of recovered and died forecast the optimal number of inputs is equal to 6. As result of these experiments the optimal values of inputs number were determined to forecast of absolute values and daily increase of indicators.

4.3. Estimation of forecast accuracy for different short-term periods

In the subsequent experiments the forecasting period (interval) was investigated for absolute and daily indicators of covid-19. The corresponding results are presented in the Tables10-12. To predict the absolute values of the number of recovered the optimal period was 6 days, but the accuracy for all periods was approximately the same (Table 11).

Table 10
MAPE values dependence on number of inputs when predicting the absolute values of confirmed cases numbers

Period, days	MAPE
5	2.35
6	2.79
7	2.94
8	3.01

Table 11
MAPE values when predicting the absolute values of the number of recovered for different short-term period

Period, days	MAPE
5	3.64
6	3.41
7	3.42
8	3.67

Table 12
MAPE values when predicting the absolute values of the number of deaths for different short-term period

Period, days	MAPE
5	3.26
6	3.87
7	3.69
8	3.48

The forecast of daily increase in the number of confirmed cases was the best for the period of 5 days (Table 12). The corresponding results when forecasting daily increase in covid-19 indicators are presented in the Table 13. The predicted values of the daily increase in the number of recovered were more accurate for the period of 5 days, and the worst were the periods of 7 and 8 days, and the values of MAPE for them are approximately the same, but they differ significantly from the best case.

Table 13
The value of MAPE in predicting the daily increase of various covid -19 indicators for different short-term period

Period, days	MAPE for number of confirmed cases	MAPE for number of deaths	MAPE for number of recovered
5	22.13	59.87	21.67
6	35.85	61.83	27.12
7	50.01	66.78	40.11
8	40.47	65.87	40.44

5. Neural network Back Propagation for Covid-19 forecasting

5.1. Experimental investigations of NN Back Propagation

The neural network Back Propagation (NNBP) is a universal approximator because it can approximate any continuous nonlinear bounded function from n variables Y = F (x1, x2,…, xn). It's property is used to build a forecast.

The architecture of NNBP which was considered in this paper consists of four layers: input layer: the number of neurons depends on the training sample, namely the value of the sliding window; hidden layer 1: contains 16 neurons; hidden layer 2: contains 8 neurons; output layer: has one output.

The activation functions for latent layer neurons and the source layer neuron are the same and are sigmoidal. A gradient descent algorithm was used to train the neural network. To assess the accuracy of the forecast, we used the same criterion of MAPE like hybrid neo-fuzzy network.

The forecast of absolute values and daily growth of indicators using the optimum parameters determined in previous section 3 was carried out. That is, the value of the sliding window was taken as the number of inputs for the case. The forecasting period was similarly chosen. Each case was explored at a different percentage of the training sample in order to obtain better forecasting results. The percentage of the training sample was 60%, 70%, 80%, and the test sample remained, respectively, 40%, 30%, 20% of the whole data sample.

To determine the best forecast in addition to the value of MAPE as an additional criterion was taken the number of epochs for learning, but in the selection of the best forecast it was not taken into account. The number of epochs of learning was considered to be the number of the last epoch, after which the value of the loss function didn't not improve. In this case training was stopped, and the weights of the neurons were restored to the corresponding values. The experimental results for forecasting covid-19 indicators for NN Back Propagation are presented in Tables 14-16.

Table 14

MAPE values for the number of confirmed cases for a period of 5 days with a sliding window of 4 points for different percentages of the training sample

Training sample,%, %	Number of epochs	MAPE
60	741	3.87
70	87	3.16
80	34	2.75

Table 15

MAPE values when predicting the number of recovered for a period of 6 days with a sliding window of 6 points for different percentages of the training sample

Training sample,%, %	Number of epochs	MAPE
60	99	3.12
70	179	2.67
80	36	2.27

The highest accuracy of the forecast of the number of recovered was achieved at 80% training sample, and training lasted 36 epochs.

Table 16

MAPE values when predicting the number of deaths for a period of 5 days with a sliding window of 6 points for different training samples

Training sample,%, %	Number of epochs	MAPE
60	99	4.59
70	98	4.31
80	36	3.86

In the case of predicting the number of deaths, the most accurate forecast was using 80% training sample (Table 16).

5.2. Comparison of forecasting results of hybrid NFN and Back Propagation

For the convenience of forecasting efficiency estimation the comparison of the forecasting results by hybrid neo-fuzzy neural network and NN Back Propagation are presented in the Table 17.

Analyzing the results of Table 17 the following conclusions are made.

Hybrid neo-fuzzy network appears to be better than NN BP when forecasting absolute (integral) values of covid indicators while NN Back propagation is better for forecasting daily increase in covid-19 indicators.

It should be noted that the values of MAPE in predicting the daily increase in the number of confirmed cases and recovered have a relatively small difference for both neural networks. A significant difference is observed for MAPE in predicting the absolute and daily increase values of the number of died. If to pay attention to the size of the sliding window, one may conclude that its smaller value ensures greater accuracy in the case of forecasting using GMDH- neo fuzzy network.

Table 17

Comparison of GMDH neo fuzzy and Back Propagation forecasting results

COVID-19 Indicators		Period (forecasting interval)	Sliding window size	MAPE	
				GMDH-neo-fuzzy	Back Propagation
Absolute values	Confirmed	5	4	2.35	2.75
	Recovered	6	6	2.41	2.27
	Died	5	6	3.26	3.86
Daily increase	Confirmed	5	4	22.13	27.58
	Recovered	5	6	21.67	18.69
	Died	5	6	59.87	28.33

6. Synthesis of optimal structure of hybrid GMDH neo fuzzy network

The GMDH method was used to synthesize the structure of the hybrid network based on the principle of self-organization. The successive increase in the number of layers is carried out until the value of the external criterion of optimality MSE begins to increase for the best model of the current layer. In this case it is necessary to return to the previous layer, to find there the best model with the minimum value of criterion. Then we move backward, go through its connections, find the corresponding neurons of the previous layer. This process continues until we reach the first layer and the corresponding structure is automatically determined.

The process of synthesis of the network structure in the forward direction is shown in Fig. 3 where in green color the outputs which passed through selection block (SB)are shown while in red color - outputs which were dropped (excluded) by SB.

The process of restoring the desired structure in the backward direction is shown in Fig. 4. In yellow color nodes and their connections selected by this process are indicated.

The corresponding optimal synthesized structure of the hybrid network for this forecasting problem is shown in Fig. 5.

It consists of 3 layers: first layer has 3 neo-fuzzy neurons, second layer - two neurons and the last-one neuron.

7. Conclusions

In this paper the problem of short-term covid-19 forecasting is considered and investigated. For its solution a new class of NN – hybrid neo-fuzzy network based on self- organization is suggested.

The experimental investigations of hybrid NFN were carried out. In the results of experiments the optimal parameters of hybrid network were found: membership functions, number of inputs and forecasting interval. Optimal structure of hybrid neo-fuzzy network was constructed using GMDH method.

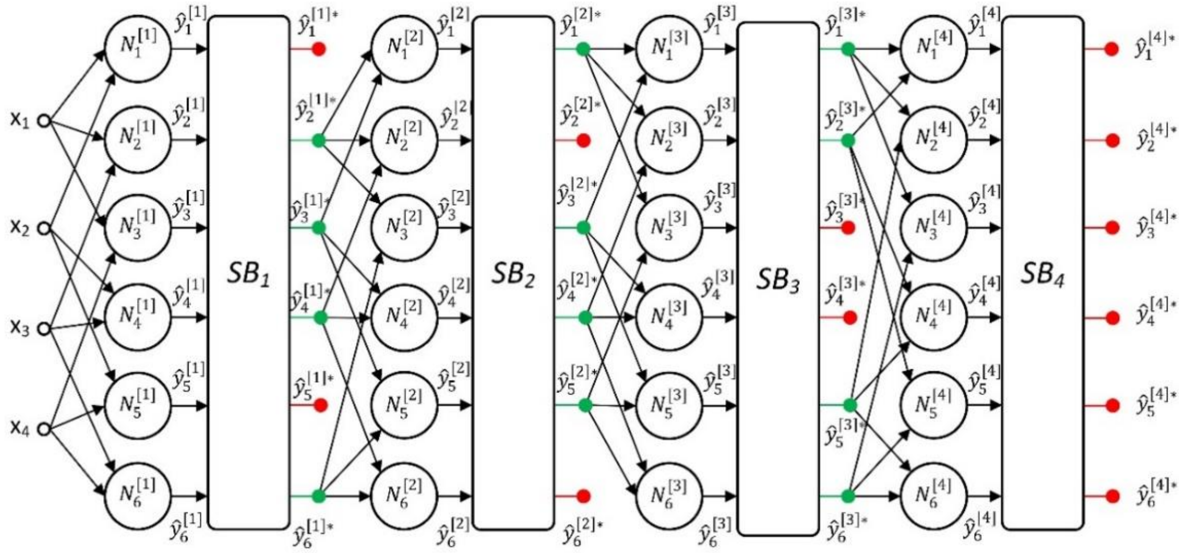

Figure 3: Hybrid network structure construction using GMDH method

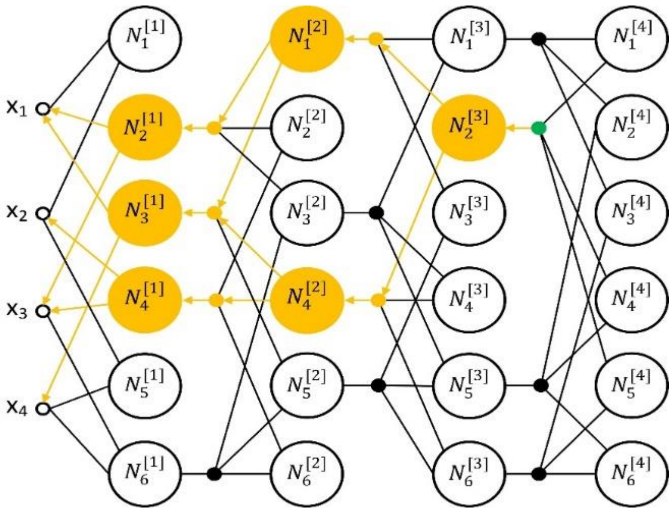

Figure 4: Process of restoring found optimal structure in backward direction

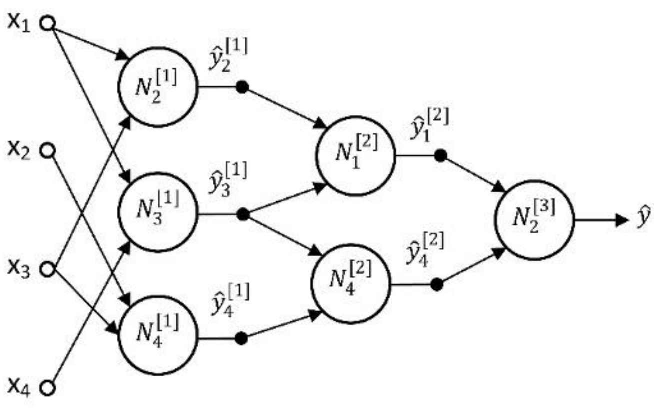

Figure 5: Sythesized structure of hybrid network for covid forecast

The experimental investigations of hybrid NFN in short-term forecasting covid-19 indicators, incl. confirmed cases, number of recovered and died -were performed and forecasting efficiency was estimated. The forecasting accuracy of hybrid NFN was compared with NN Back Propagation.

As a whole hybrid neo-fuzzy neural network appeared to be the efficient tools for short-term pandemic indicators forecasting.

8. References

[1] World Data Center for Geoinformatics and Sustainable Development. URL: https://wdc-ukraine.maps.arcgis.com/apps/dashboards/07b38b42264d4a2ea5a05d578e235559 (Last accessed: 25.03.2021).

[2] Staff of the National Security and Defense Council of Ukraine. Coronavirus epidemic monitoring system. URL: https://covid19.rnbo.gov.ua (Last accessed: 25.03.2021).

[3] Covid-19 spread: Reproduction of data and prediction using an SIR model on Euclidean network. URL: https://arxiv.org/pdf/2003.07063.pdf (Last accessed: 25.03.2021).

[4] Reiner, R.C., Barber, R.M., Collins, J.K. et al. Modeling COVID-19 scenarios for the United States. URL: https://doi.org/10.1038/s41591-020-1132-9 (Last accessed: 25.03.2021).

[5] Zeynep C. Estimation of COVID-19 prevalence in Italy, Spain, and France. URL: https://www.sciencedirect.com/science/article/pii/S0048969720323342 (Last accessed: 25.03.2021).

[6] Zeroual A., Harrou F., Dairi A., Sun Y. Deep learning methods for forecasting COVID-19 time-series data: A Comparative study. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7362800 (Last accessed: 25.03.2021).

[7] Short-term spread forecast COVID-19 (22.08.20-26.08.20) based on Back Propagation Neural Network. URL: http://wdc.org.ua/uk/covid19-short-term-19 (Last accessed: 25.03.2021).

[8] Hartono P. Similarity maps and pairwise predictions for transmission dynamics of COVID-19 with neural networks. Informatics in Medicine Unlocked. URL: https://www.sciencedirect.com/science/article/pii/S2352914820302689 (Last accessed: 27.03.2021).

[9] Roosa, K., Lee, Y., Luo, R., Kirpich, A., Rothenberg, R., Hyman, J. M., Yan, P. Short-term Forecasts of the COVID-19 Epidemic in Guangdong and Zhejiang, China: February 13-23, 2020. Journal of clinical medicine. 9(2), 596. URL: https://doi.org/10.3390/jcm9020596 (Last accessed: 27.03.2021).

[10] Mandic D.P., Chambers J.A. Recurrent Neural Networks for prediction. Learning algorithms, architectures and Stability. England: John Wiley&Sons, Ltd, 2001. 308 p.

[11] Patterson J., Gibson A. Deep Learning. A practitioner’s approach. Sebastopol, CA: O’Reilly, 2017. 507 p.

[12] Aggarwal C. Neural Networks and Deep Learning. Springer, 2018. 497 p.

[13] Yong Y., Xiaosheng S. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. Massachusetts Institute of Technology. URL: https://www.mitpressjournals.org/doi/pdf/10.1162/neco_a_01199 (Last accessed: 29.03.2021).

[14] About Python. URL: https://www.python.org/about (Last accessed: 29.03.2021).

[15] PyCharm: The Python IDE for Professional Developers by JetBrains. URL: https://www.jetbrains.com/pycharm (Last accessed: 29.03.2021).

[16] Why TensorFlow. URL: https://www.tensorflow.org/about (Last accessed: 29.03.2021).

[17] Keras: The Python deep learning API. URL: https://keras.io (Last accessed: 29.03.2021)

[18] Scikit-learn: machine learning in Python. URL: https://scikit-learn.org/stable (last accessed: 09.06.2021).

[19] Skfuzzy 0.2 docs. URL: https://pythonhosted.org/scikit-fuzzy (last accessed: 09.06.2021).

[20] API Coronavirus. URL: https://api-covid19.rnbo.gov.ua (Last accessed: 29.03.2021).

[21] Yuriy Zaychenko, Yevgeniy Bodyanskiy, Oleksii Tyshchenko, Olena Boiko, Galib Hamidov. Hybrid GMDH-neuro-fuzzy system and its training scheme. Int. Journal Information theories and Applications, 2018. vol.24, Number 2.-pp. 156-172.

[22] Evgeniy Bodyanskiy[1], Yuriy Zaychenko[2], Olena Boiko[1], Galib Hamidov[3] The hybrid GMDH-neo-fuzzy neural network in forecasting problems in financial sphere. International conference IEEE SAIC 2020 .- Kiev 2020.