UDC 004.89 : 004.032.26 : 519.651 : 532.5.013.12 : 627.152

Yaroslav Khodnevych, PhD (Engineering), Research Scientist
ORCID ID: https://orcid.org/0000-0002-5510-1154 ***e-mail:*** ya.v.khodnevych@gmail.com

Dmytro Stefanyshyn, D. S. (Engineering), Senior Research Scientist
ORCID ID: https://orcid.org/0000-0002-7620-1613 ***e-mail:*** d.v.stefanyshyn@gmail.com

Institute of Telecommunications and Global Information Space of NASU, Kyiv, Ukraine

DO WE NEED A MORE SOPHISTICATED MULTILAYER ARTIFICIAL NEURAL NETWORK TO COMPUTE ROUGHNESS COEFFICIENT?

Abstract. *Artificial neural networks (ANNs) are one of the most rapidly growing fields of soft computing. Along with deep learning, they are currently the most widely used machine learning techniques. Artificial neural networks are especially suitable for problem-solving where a researcher deals with incomplete data sets and no algorithms or specific sets of rules to be followed.*

This article deals with a case of comparison of several modifications of neural networks that may be applied to compute Chézy's roughness coefficient. Neural network modelling is often started with one hidden layer. Having even one hidden layer, a neural network presents a powerful computing system to give good results. If it is necessary, the number of hidden layers may increase. Usually, two or three hidden layers of neurons are used. Diverse activation functions may also apply. The article aims to explore the necessity of developing sophisticated multilayer artificial neural networks to compute Chézy's roughness coefficient.

Under the study, the following modifications of the neural network computing Chézy's roughness coefficient were considered and analysed: (1) Application of two hidden layers of neurons; (2) Application of three hidden layers of neurons; (3) Use of a dropout algorithm for training neural networks by randomly dropping units during training to prevent their co-adaptation; (4) Apart from the sigmoid (logistic) activation function, the use of other artificial neuron transfer functions – hyperbolic tangent (tanh) and rectifying activation function (ReLU).

The training and testing of the considered neural network options were carried out using the actual hydro-morphological and hydrological data related to the channel section on the Dnieper River (downstream of Kyiv), the Desna River section near Chernihiv, and the Pripyat River section near the town of Turiv. The Python object-oriented programming environment was applied to build and train the neural networks. The test results confirm the acceptability and sufficiency of computing the Chézy roughness coefficient using the ANN of direct propagation with one hidden layer and a sigmoid logistic activation function. The formation of a qualitative set of training data, as well as data arrangement and choosing a relevant computing model based on empirical knowledge, are, as concluded, among more actual issues than creating more sophisticated neural networks.

Keywords: *activation functions, artificial neural networks, Chézy's roughness coefficient, comparison, dropout algorithm, hidden layers, modifications, neurons*

https://doi.org/10.32347/2411-4049.2023.4.170-182

1. Introduction

Artificial neural networks (ANNs, neural networks, neural computing) present one of the most popular fields of so-called soft computing that deals with approximate models and using inexact solutions [1-5]. Soft computing is not a single method of solving diverse tasks and problems using computing. It is a holistic approach that uses probabilistic models, multivalued and fuzzy logic, neural networks, and evolutionary (genetic) algorithms (Fig. 1) and their hybridizations [5-7].

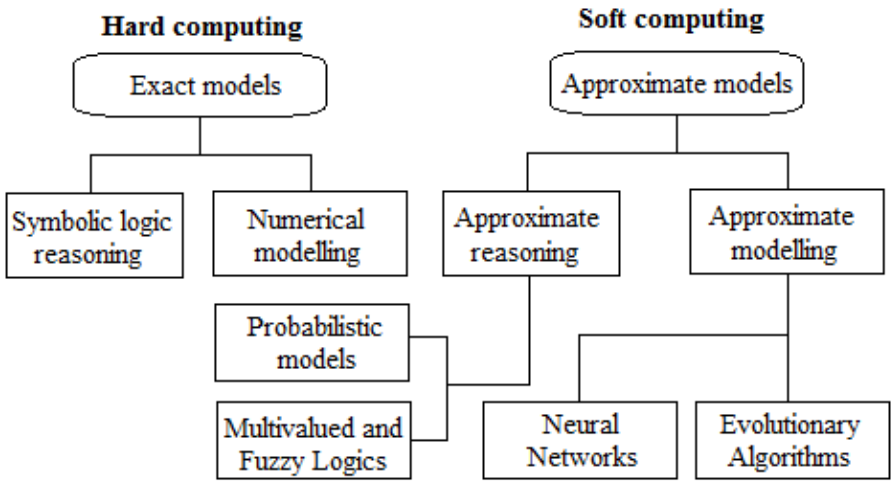

Fig. 1. Approaches to solving problems using computing

According to L. Zadeh [1], the point of departure in soft computing is the thesis that precision and certainty carry a cost and that computation, reasoning, and decision-making should exploit – wherever possible – the tolerance for imprecision and uncertainty. Soft computing is a flexible tool for mathematical modelling. Being a flexible tool for modelling and forecasting real-world phenomena [5], soft computing is used in various fields of human activity, such as economy, environment [5, 8, 9], engineering, science, medicine, etc. [1, 10-13].

Neural computing is one of the most rapidly growing fields of soft computing. Especially, ANNs are highly suitable for solving complex problems where there are no algorithms or specific sets of rules to be followed. Along with deep learning, they are currently the most widely used machine learning techniques [14-16]. Neural computing is developed as an integral part of artificial intelligence and data science [15, 17, 18]. The most typical application areas of ANNs are pattern and data recognition, image and speech processing and recognition, machine translation, and medical diagnosis. Essentially, ANNs have become a successful forecasting method used in diverse applications [8-10, 18-20]. It is because ANNs have several advantages compared with other forecasting methods [2, 3, 18-20]: results depend on the accuracy of available data; ANNs deal with incomplete data sets; neural computing is an adaptive method; ANNs can learn without any prior assumption; they are non-linear models with good generalization ability.

This article deals with the case of comparison of several modifications of ANNs to compute the Chézy coefficient. Chézy's roughness coefficient is a hydraulic integral empirical characteristic controlling the hydraulic resistance to open flows in river channels. The challenge is that it may not be determined directly using field measurements or experimentally. To determine the Chézy coefficient, many empirical formulas have been developed [21], usually dealing with incomplete data sets. As practice shows, the appropriate formula choosing can become a challenge for researchers. Therefore, in [22], we proposed to solve the problem using an artificial neural network. It was a multilayer perceptron of direct propagation with one hidden layer and a sigmoid logistic activation function. That network showed good predictive skills in the condition of correct studying of the subject area and relevant input data arrangements [22].

As known, the basic structure of a multilayer ANN (a multilayer perceptron) consists of an input layer, one or more hidden layers, an output layer, an activation function and a set of weights and biases. In a fully connected neural network, each neuron in one layer is connected to all neurons in the next layer. Neurons of the input layer in such networks transmit input signals to the first hidden layer without converting them. In hidden neurons, sequentially, layer by layer, there is a nonlinear conversion of signals. Each network neuron produces a weighted sum of its inputs, passes this value through the activation function and gives the output value. Signals from the last hidden layer arrive at the neurons of the output layer, which eventually form the ANN response [14, 19, 20, 22-24].

The modelling is often started with one hidden layer [14]. Having even one hidden layer, the ANN is a powerful computing system [25], giving good results [22]. If it is necessary, the number of hidden layers may increase. Usually, two or three hidden layers of neurons are used [14, 23-26]. Diverse activation functions may also be applied. That is how researchers try to improve the ANN performance. ***This article aims*** to answer the question of whether we need to develop a more sophisticated multilayer artificial neural network to compute Chézy's roughness coefficient within the computing model that was proposed in [22].

2. Computing model, the basic structure and considered modifications of the neural network computing Chézy's roughness coefficient

2.1. Computing model and the basic structure of the ANN computing Chézy's roughness coefficient

In [22], the following computing model was proposed for calculating the Chézy coefficient C within the fully connected ANN of direct propagation with one hidden layer and a sigmoid logistic activation function:

$$C = f(x_1,\ x_2), \tag{1}$$

$$x_1 \in \{n, \Delta, S_f, B\},\ x_2 \in \{h, R\}, \tag{2}$$

where n is the Gauckler-Manning roughness coefficient; Δ is the height of protrusions of roughness; S_f is the water surface slope; B is the average flow width; h is the average flow depth; R is the hydraulic radius. It was assumed that

multicollinearity between the hydro-morphological parameters n, S_f, B, Δ, h, and R could be neglected.

The basic structure of the ANN proposed in [22] for computing the Chézy coefficient C consists of two input neurons, four hidden layer neurons, and one output layer neuron (Fig. 2).

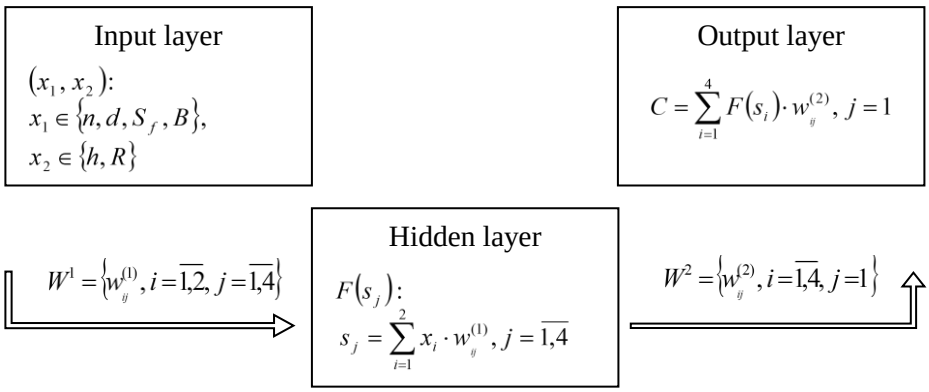

Fig. 2. Flow-chart showing the architecture of the ANN computing Chézy's coefficient C according to [22]

The ANN uses a direct propagation network model with a linear source neuron [22]; the sigmoid (logistic) activation function is applied to the neurons of the hidden layer [14, 22]:

$$F(s) = \frac{1}{1+e^{-\beta \cdot s}}, \tag{3}$$

and its first derivative

$$F'(s) = \beta \cdot F(s)\big(1 - F(s)\big), \tag{4}$$

where the parameter β influencing the steepness of the transition is equal to 1.

According to the ANN [22], receiving the (x_1, x_2) values, the input layer transmits them (without conversion) to the next layer of neurons using the weight matrix $W^1 = \left\{w_{ij}^{(1)},\ i = \overline{1,2},\ j = \overline{1,4}\right\}$ containing the weight values $w_{ij}^{(1)}$ of inputs for the hidden layer neurons. Afterwards, the hidden layer containing four neurons calculating the weighted sum $s_j = \sum_{i=1}^{2} x_i \cdot w_{ij}^{(1)}$ conducts this sum value through the activation function $F(s_j)$, and transmits the resulting value using the weight matrix $W^2 = \left\{w_{ij}^{(2)},\ i = \overline{1,4},\ j = 1\right\}$ to the output layer. The weight matrix W^2 contains the weight values $w_{ij}^{(2)}$ of relationships of each hidden layer neuron with the output neuron. The output layer contains one neuron in which the weighted sum of its inputs is calculated, and Chézy's coefficient C value is thereby determined.

2.2. Considered modifications of the ANN

Under the study, the following modifications of the neural network computing Chézy's roughness coefficient were considered and analysed:

1) Application of two hidden layers of neurons;

2) Application of three hidden layers of neurons;

3) Use of a dropout algorithm for training neural networks by randomly dropping units during training to prevent their co-adaptation [27];

4) Apart from the sigmoid (logistic) activation function (3), the use of other artificial neuron transfer functions (activation functions) – hyperbolic tangent (tanh) and rectifying activation function (ReLU – Rectified Linear Unit) [29].

Having the same *S*-shape, the hyperbolic tangent activation function (tanh) is very similar to the sigmoid (logistic) function (3).

The equation for tanh is:

$$F(s) = tanh(s) = \frac{2}{1+e^{-2s}} - 1. \quad (5)$$

Its first derivative is:

$$F'(s) = 1 - F(s)^2. \quad (6)$$

Compared to the sigmoid function, the hyperbolic tangent activation function produces a more rapid rise in result values [15, 16, 28].

The ReLU function is the most commonly used activation function in deep learning models. The function returns 0 if it receives any negative input, but for any positive value s it returns that value back. The equation for ReLU is:

$$F(s) = \begin{cases} 0, & s < 0 \\ s, & s \geq 0. \end{cases} \quad (7)$$

The ReLU first derivative is:

$$F'(s) = \begin{cases} 0, & s < 0 \\ 1, & s \geq 0. \end{cases} \quad (8)$$

The sigmoid and hyperbolic tangent activation functions cannot be used in networks with many layers due to the vanishing gradient problem. The rectified linear activation function overcomes the vanishing gradient problem inhering, allowing models to learn faster and perform better [29].

3. Materials and methods

The training and testing of the considered neural network options were carried out using the actual hydro-morphological and hydrological data related to the channel section on the Dnieper River (downstream of Kyiv), the Desna River section near Chernihiv, and the Pripyat River section near the town of Turiv. The chosen river sites are characterized by a straight earthen channel with a simple cross-sectional shape and calm flow (the Froude number, Fr<<1). The limits of change of

hydrological and hydro-morphological parameters are the following [22]: the water discharge Q = 48.8 ÷ 3665.0 m^3/s; the average flow velocity $V = Q/A$= 0.336 ÷ 0.968 m/s, where A is the cross-sectional area of the flow (m^2); the water surface slope S_f = 0.000036 ÷ 0.00016; the average flow depth h = 1.0 ÷ 6.2 m; the average flow width B = 122.0 ÷ 611.0 m; the Manning roughness coefficient n = 0.027 ÷ 0.045 (s/m$^{1/3}$); the identified (observed) Chézy roughness coefficient C_o = 27.0 ÷ 43.7 (m$^{1/2}$/s). The input data is summarised in Table 1.

Table 1. Input data used in the training and testing of the considered neural network options

Rivers, channel sections	Hydrological and hydro-morphological parameters:						
	Q_o (m^3/s)	A (m^2)	B (m)	h (m)	$S_f \cdot 10^3$	n (s/m$^{1/3}$)	C_o (m$^{1/2}$/s)
Dnieper, downstream of Kyiv (training)	657.4	1956	575	3.4	0.046	0.045	27.0
	1123	2403	586	4.1	0.054	0.040	31.4
	3665	3787	611	6.2	0.079	0.031	43.7
Dnieper, downstream of Kyiv (testing)	1763	2858	595	4.8	0.063	0.036	35.6
	2601	3320	604	5.5	0.071	0.033	39.7
Desna, Chernihiv (training)	188	501.8	125	4.0	0.036	0.041	31.1
	249.4	580	129	4.5	0.040	0.040	32.2
	403.7	742.4	135	5.5	0.046	0.039	34.2
	497.5	826.3	138	6.0	0.049	0.038	35.1
Desna, Chernihiv (testing)	321.2	660.3	132	5.0	0.043	0.039	33.3
Pripyat, Turiv (training)	48.8	122	122	1.0	0.16	0.032	31.6
	89	195.4	130	1.5	0.128	0.033	32.9
	248.6	437.3	146	3.0	0.087	0.034	35.1
Pripyat, Turiv (testing)	136,3	273	136	2.0	0. 109	0.033	33.8
	189.7	353.8	142	2.5	0. 097	0.034	34.5

Training data samples consisted of normalized values of the characteristics obtained with uniform linear interpolation in the vicinity of the observed values of parameters. Numerical data were converted in such a way as to obtain their model values varying in the range between 0 and 1 [22]. The test cases observed values were not included in the training samples.

The Python object-oriented programming environment [23, 24, 30] was applied to build and train the considered options of the neural network. The software implementation of the computational algorithms is given in [31].

The weight coefficients of the considered ANN options were adjusted on a series of real case examples of $(x_1,\ x_2)$ values in such a way as to achieve a reduction in the error between the predicted (computed) C_p and observed (reference) C_o estimates of Chézy's roughness coefficient C. The reference C_o values were calculated on actual data (Table 1) according to the Chézy formula:

$$C_o = \frac{Q_o}{A \cdot \sqrt{R \cdot S_f}}. \quad (9)$$

The initial values of the weight coefficients were set randomly, close to zero. At each iteration step (learning epoch), at the ANN entrance, training examples were input and the neural network outputs were computed. The obtained results were further compared with the reference values with error estimating. For the neurons of each hidden layer, the network error was calculated. The obtained error values were used to recalculate weight coefficients according to the inverse error propagation algorithm [14, 23, 24]. Then, the transition to the next learning epoch was performed. When the required number of epochs was fulfilled, or the computational error amounted to an acceptable value, the algorithm was stopped.

The testing of the considered ANN options was carried out according to the actual data of observations (Table 1), which were not used in the network training. The testing procedure consisted of a comparison of the actual (observed, gauged) Q_o and computed (predicted) Q_p water discharges:

$$Q_p = C_p \cdot A\sqrt{R \cdot S_f}, \quad (10)$$

where C_p is the predicted (computed using the ANN) Chézy coefficient value.

To assess the forecast (predictive) skill of the considered ANN options, the Nash-Sutcliffe model efficiency coefficient (*NSE*) was used [32]:

$$NSE = 1 - \frac{\sum_{i=1}^{6}(Q_{o,i} - Q_{p,i})^2}{\sum_{i=1}^{6}(Q_{o,i} - \bar{Q}_o)}, \quad (11)$$

where $Q_{o,i}$, $Q_{p,i}$ are observed and predicted water discharges for a river channel section i, $i = \overline{1,6}$; $\bar{Q}_o$ is the mean of the observed discharges $Q_{o,i}$.

4. Results

In general, a multi-layer direct propagation neural network for computing the Chézy coefficient within the model (1) and (2) works as follows. The input layer receives the predictors $(x_1,\ x_2)$ values and transmits them without conversion to the next layer of neurons using the weight matrix $W^1 = \left\{w_{ij}^{(1)},\ i = \overline{1,2},\ j = \overline{1,m}\right\}$ containing the weight values $w_{ij}^{(1)}$ of inputs for all hidden layer neurons, where m is the number of artificial neurons in the hidden layer. The hidden layer containing neurons, each of which computes a weighted sum $s_j = \sum_{i=1}^{2} x_i \cdot w_{ij}^{(1)}$, $j = \overline{1,m}$, relating to its input data, conducts this sum value through the activation function $y_j^{(1)} = F(s_j)$, and transmits the obtained value to the next layer, etc. The weight matrix $W^{\xi-1} = \left\{w_{ij}^{(\xi-1)},\ i = \overline{1,m},\ j = 1\right\}$ contains the value of the weight of connections of each neuron of the previously received layer with the original neuron, where ξ is the total number of layers of a multilayer ANN. For a neural network with two hidden layers, $\xi = 4$; with three hidden layers, $\xi = 5$. The output layer contains

one neuron in which the weighted sum of its inputs is calculated, and Chézy's coefficient value is determined: $C = \sum_{i=1}^{m} F(s_i) \cdot w_{ij}^{(\xi-1)}$, $j = 1$.

The results of using different options of a multilayer neural network of direct propagation for calculating Chézy's coefficient within the computing model framework according to equations (1) and (2) are given in Tables 2, 3, and 4.

Table 2. Results of using different options for the number of hidden layers and neurons with the sigmoid activation function (learning factor is 0.002)

The number of:		Activation function	The number of learning epochs	Duration of calculations*, sec	$\overline{C}_p$, $m^{1/2}/s$
hidden layers	neurons m				
1	4	Sigmoid, β = 1	100	<1	34.1149
2	4		100	<1	34.1151
2	8		49	<1	34.1166
2	16		24	<1	34.1166
3	4		100	~1	34.1157
3	8		47	<2	34.1166
3	16		18	<2	34.1170

*The duration of computing is conditional; it depends on the computer equipment's power

Table 3. Results of using different options for the number of hidden layers, neurons, and the different activation functions (learning factor is 0.002)

The number of:		Activation function	The number of learning epochs	Duration of calculations*, sec	$\overline{C}_p$, $m^{1/2}/s$
hidden layers	neurons m				
1	4	Sigmoid, β = 1	100	<1	34.1149
2	8	Sigmoid, β = 1	49	<1	34.1166
2	8	Sigmoid, β = 2	47	<1	34.1165
2	8	Sigmoid, β = 5	45	<1	34.1168
2	8	Sigmoid, β = 10	48	<1	34.1168
2	8	Sigmoid, β = 11	48	<1	34.1166
2	8	Sigmoid, β = 12	48	<1	34.1164
3	8	Sigmoid, β = 1	47	<2	34.1166
3	8	Sigmoid, β = 2	43	<2	34.1168
3	8	Sigmoid, β = 5	38	<2	34.1169
3	8	Sigmoid, β = 10	38	<2	34.1193
2	8	tanh	49	<1	≈ 0
3	8	tanh	47	<2	≈ 0
2	8	ReLU	49	<1	0
3	8	ReLU	47	<2	0

*The duration of computing is conditional; it depends on the computer equipment's power

Table 2 shows the results of using the considered neural network options for the number of hidden layers and neurons with the sigmoid activation function. Table 3 gives the results of using the different considered ANN options including the number of hidden layers, neurons, and the different activation functions – sigmoid with

various β values, hyperbolic tangent (tanh) and rectifying (ReLU) ones. The results of using a dropout algorithm for training the considered ANN options are given in Table 4.

Table 4. Results of using a dropout algorithm for training the considered ANNs by randomly dropping units during training

The number of:		Dropping units in hidden layers:	The number of learning epochs	$\overline{C}_p$, $m^{1/2}/s$
hidden layers	neurons m			
1	4	not applied	100	34.1149
1	8	applied	155	31.3485
1	16	applied	125	33.3661
2	8	not applied	49	34.1166
		applied, but separately for each layer	53	30.1830
		applied, same for all layers	53	30.7412
		applied, but only for the first hidden layer	45	34.1177
2	16	not applied	24	34.1166
		applied, but separately for each layer	25	31.1932
		applied, same for all layers	25	31.4216
		applied, but only for the first hidden layer	21	34.1187
3	8	not applied	47	34.1166
		applied, but separately for each layer	60	31.2421
		applied, same for all layers	55	31.2433
		applied, but only for the first hidden layer	43	34.1168
3	16	not applied	18	34.1170
		applied, but separately for each layer	30	33.9920
		applied, same for all layers	18	33.8473
		applied, but only for the first hidden layer	16	34.1140

For all the considered ANNs (Table 4), the sigmoid activation function (3) was applied. The learning factor was 0.002.

5. Discussion

Analysing the results, it can be emphasised that the standard hyperbolic tangent (tanh) and rectifying (ReLU) activation functions appeared to be irrelevant for approximating the Chézy coefficient according to the computing model of (1) and (2). In this case, additional settings and modifications are needed to use these functions to obtain the desired results when calculating the Chézy roughness coefficient. This task may be the aim of future research.

The increase in the number of hidden layers and neurons contributes to the fact that the neural network within the model of (1) and (2) learns faster (the number of learning epochs tends to decrease). However, this practically did not affect the results of the Chézy coefficient computing.

To approximate the Chézy roughness coefficient function $C = f(x_1,\ x_2)$, $x_1 \in \{n, \Delta, S_f, B\}$, $x_2 \in \{h, R\}$ [22], it may be enough to use the fully connected ANN of direct propagation with one hidden layer and a sigmoid logistic activation function. In particular, in this case study, the Nash-Sutcliffe model efficiency coefficient

(*NSE*) varied between 0.94-0.98. It can signify the forecast (predictive) skill of the ANN is quite high.

In turn, the relative prediction errors E_r (%) of forecasting water discharges varied between 0.9-13.9% depending on rivers and river channel sections (Table 1), where:

$$E_r = \frac{|Q_o - Q_p|}{Q_o} \cdot 100\%, \quad (12)$$

where Q_o and Q_p are the observed and predicted values of water discharges. It may also indicate the high performance of the ANN of direct propagation with one hidden layer and a sigmoid logistic activation function [22] in computing the Chézy roughness coefficient as an integral empirical characteristic of hydraulic resistance to open flows in river channels [21].

In addition, in applying the sigmoid activation function, the accuracy of forecasting the Chézy coefficient values was determined using the relative prediction errors E_r (%):

$$E_r = \frac{|\overline{C}_o - \overline{C}_p|}{\overline{C}_o} \cdot 100\%, \quad (13)$$

where $\overline{C}_o$ is the average observed (identified) value and $\overline{C}_p$ is the average predicted value of Chézy's coefficient.

When using the sigmoid logistic activation function, the relative prediction errors of forecasting the Chézy coefficient values varied between 3.56-3.58% depending on the considered ANN options. Thereby, the answer to the question of whether we need to develop a more sophisticated multilayer artificial neural network to compute Chézy's roughness coefficient may be the next. When applying the computing model of $C = f(x_1,\ x_2)$, $x_1 \in \{n, \Delta, S_f, B\}$, $x_2 \in \{h, R\}$ [22], it is recommended to use the ANN options of direct propagation with one or two hidden layers and a sigmoid activation function. More important in this case may be the formation of a qualitative set of training data.

6. Conclusions

Under the study, to explore the necessity of developing more sophisticated multilayer artificial neural networks to compute Chézy's coefficient, the following modifications of the neural computing were considered and analysed: (1) Application of two hidden layers of neurons; (2) Application of three hidden layers of neurons; (3) Use of a dropout algorithm for training neural networks by randomly dropping units during training to prevent their co-adaptation; (4) Apart from the sigmoid (logistic) activation function), the use of other artificial neuron transfer functions (activation functions) – hyperbolic tangent (tanh) and rectifying activation function (ReLU – Rectified Linear Unit). The training and testing of the considered neural network options were carried out using the actual hydro-morphological and hydrological data related to the channel section on the Dnieper River (downstream of Kyiv), the Desna River section near Chernihiv, and the Pripyat River section near Turiv.

The test results confirm the acceptability and sufficiency of computing the Chézy roughness coefficient using the ANN of direct propagation with one hidden layer and a sigmoid logistic activation function. The formation of a qualitative set of training data, as well as data arrangement and choosing a relevant computing model based on empirical knowledge, as revealed, are among more actual issues than creating more sophisticated neural networks.

REFERENCES

1. Zadeh, L.A. (1994). Fuzzy logic, neural networks and soft computing. Fuzzy Systems. Communications of the ACM, Vol. 37, No. 3, 77–84.
2. Ibrahim, D. (2016). An overview of soft computing. Procedia Computer Science, 102, 34–38. https://doi.org/10.1016/j.procs.2016.09.366
3. Bele, S.B. (2020). The concept of soft computing. International Journal of Recent Advances in Multidisciplinary Research, Vol. 07, Issue 03, 5623–5625.
4. Carlos, M. (2022). Characteristics of Soft Computing and its Applications. Int. J. of Swarm Intelligent and Evolutionary Comp. 11:275. https://doi.org/10.35248/2090-4908.22.11.275
5. Soft computing approach for mathematical modeling of engineering problems. (2022). Ed. by A. Ahmadian, and S. Salahshour. CRC Press. Taylor & Francis Group, London, N.Y., 203 p.
6. Singh, Ju. (2016). A Parameterized Comparison of Fuzzy Logic, Neural Network and Neuro-Fuzzy System: A Literature. International Journal of Computer Science and Mobile Computing, Vol. 5, Issue 5, 478-482.
7. Thakur, A., Dhiman, K., and Phansikar, M. (2021). Neuro-Fuzzy: Artificial Neural Networks & Fuzzy Logic. International Journal for Research in Applied Science & Engineering Technology, Vol. 9, Issue IX, 128-135.
8. Haghbin, M., Sharafati, A., Motta, D. et al. (2021). Applications of soft computing models for predicting sea surface temperature: a comprehensive review and assessment. Progress in Earth and Planetary Science, 8:4. https://doi.org/10.1186/s40645-020-00400-9
9. Adnan, R.M., Meshram, S.G., Mostafa, R.R., et al. (2023). Application of Advanced Optimized Soft Computing Models for Atmospheric Variable Forecasting. Mathematics, 11, 1213. https://doi.org/10.3390/math11051213
10. Pham, B.T., Nguyen, M.D., Al-Ansari, N. et al. (2021). A Comparative Study of Soft Computing Models for Prediction of Permeability Coefficient of Soil. Mathematical Problems in Engineering, Vol. 2021, 7631493. https://doi.org/10.1155/2021/7631493
11. Redjimi, H., and Tar, J.K. (2021). A Simple Soft Computing Structure for Modeling and Control. Machines, 9, 168. https://doi.org/10.3390/machines9080168
12. Afaq, H., and Saini, S. (2011). Swarm Intelligence based Soft Computing Techniques for the Solutions to Multiobjective Optimization Problems. International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 2, 498-510.
13. Soft Computing: Recent Advances and Applications in Engineering and Mathematical Sciences. (2023). Edited by P. Debnath, O. Castillo, and P. Kumam. CRC Press. Taylor & Francis Group, London, N.Y., 233 p.
14. Haikin, S. (2008). Neural Networks and Learning Machines (3rd Edition), Prentice Hall, 906 p.
15. Choi, R.Y., Coyner, A.S., Kalpathy-Cramer, J., Chiang, M.F., and Campbell, J.P. (2020). Introduction to machine learning, neural networks, and deep learning. Trans Vis Sci Tech., Special Issue, Vol. 9, No. 2, Article 14:2. https://doi.org/10.1167/tvst.9.2.14
16. Taye, M.M. (2023). Understanding of Machine Learning with DeepLearning: Architectures, Workflow, Applications and Future Directions. Computers, 12, 91. https://doi.org/10.3390/computers12050091

17. Russell, S.J., and Norvig, P. (2010). Artificial Intelligence: A Modern Approach. 3rd ed. Pearson Education, Inc.: Upper Saddle River, New Jersey, 1132 p.
18. Dhar, V. (2013). Data science and prediction. Communications of the ACM 56 (12): 64. https://doi.org/10.1145/2500499
19. Zhang, G., Patuwo, B.E., Hu, M.Y. (1998). Forecasting with artificial neural networks: The state of the art. International Journal of Forecasting, 14, 35–62. https://doi.org/10.1016/S0169-2070(97)00044-7
20. Montaño Moreno, J.J., Pol, A.P., and Gracia, P.M. (2011). Artificial neural networks applied to forecasting time series. Psicothema, Vol. 23, No 2, 322–329.
21. Stefanyshyn, D.V., Khodnevich, Y.V., Korbutiak, V.M. (2021). Estimating the Chezy roughness coefficient as a characteristic of hydraulic resistance to flow in river channels: a general overview, existing challenges, and ways of their overcoming. Environmental safety and natural resources, 39(3), 16–43. https://doi.org/10.32347/2411-4049.2021.3.16-43
22. Khodnevych, Y.V., Stefanyshyn, D.V. (2022). Data arrangements to train an artificial neural network within solving the tasks for calculating the Chézy roughness coefficient under uncertainty of parameters determining the hydraulic resistance to flow in river channels. Environmental safety and natural resources, 42(2), 59–85. https://doi.org/10.32347/2411-4049.2022.2.59-85
23. Chollet, F. (2018). Deep Learning with Python. Manning Publications Co., 384 p.
24. Trask, A.W. (2019). Grokking Deep Learning. Manning Publications Co., 336 p.
25. Cybenko, G.V. (1989). Approximation by Superpositions of a Sigmoidal function. Mathematics of Control, Signals and Systems, Vol. 2, No. 4, 303–314.
26. Keim, R. (2020). How Many Hidden Layers and Hidden Nodes Does a Neural Network Need? Available from https://www.allaboutcircuits.com/technical-articles/how-many-hidden-layers-and-hidden-nodes-does-a-neural-network-need/
27. Baldi, P., and Sadowski, P. (2014). The dropout learning algorithm. Artificial Intelligence, 210, 78–122. http://dx.doi.org/10.1016/j.artint.2014.02.004
28. Nielsen, M. Neural Networks and Deep Learning, 224 p. Available from https://static.latexstudio.net/article/2018/0912/neuralnetworksanddeeplearning.pdf
29. Brownlee, J. (2019). A Gentle Introduction to the Rectified Linear Unit (ReLU). In Deep Learning Performance. Available from https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/
30. Muller, A., and Guido, S. (2016). Introduction to Machine Learning with Python, Published by O'Reilly Media, 378 p.
31. Khodnevych, Ya. (2022). The software implementation of a neural network computational algorithm for predicting the Chézy roughness coefficient. Available from https://github.com/yakhodnevych/ANN_approximation_C
32. Nash, J.E., and Sutcliffe, J.V. (1970). River flow forecasting through conceptual models part I – A discussion of principles, Journal of Hydrology, 10 (3), 282–290. https://doi.org/10.1016/0022-1694(70)90255-6

The article was received 05.09.2023 and was accepted after revision 29.11.2023

Я.В. Ходневич, Д.В. Стефанишин

Чи потрібна нам більш складна багатошарова штучна нейронна мережа для обчислення коефіцієнта шорсткості?

Анотація. Штучні нейронні мережі є однією з найбільш швидко зростаючих областей м'яких обчислень. Поряд з глибоким навчанням вони натепер широко використовуються при машинному навчанні. Нейронні мережі особливо підходять для вирішення завдань, де дослідникові доводиться мати справу з неповними наборами даних і відсутні алгоритми або специфічні набори правил, яких слід дотримуватися.

У статті розглядається порівняння декількох модифікацій нейронних мереж, які можуть застосовуватися для обчислення коефіцієнта шорсткості Шезі. Моделювання нейронної мережі часто починається з одного прихованого шару. Навіть з одним

прихованим шаром нейронна мережа є потужною обчислювальною системою, яка може дати хороші результати. При необхідності кількість прихованих шарів може збільшуватися. Зазвичай використовуються два або три прихованих шари нейронів. Також можуть застосовуватися різні функції активації. Ця стаття має на меті дослідити, чи потрібно розробляти більш складну нейронну мережу для практичних обчислень коефіцієнта шорсткості Шезі.

В рамках проведеного дослідження було розглянуто та проаналізовано наступні модифікації нейронної мережі, яка обчислює коефіцієнт шорсткості Шезі: (1) застосування двох прихованих шарів нейронів; (2) застосування трьох прихованих шарів нейронів; (3) використання dropout алгоритму для навчання нейронних мереж шляхом випадкового скидання одиниць під час навчання, щоб запобігти їх спільній адаптації; (4) крім сигмоїдної (логістичної) функції активації – використання інших функцій – гіперболічного тангенса (tanh) і випрямляючої функції активації (ReLU).

Навчання та тестування розглянутих варіантів нейронної мережі проводилося з використанням фактичних гідро-морфологічних та гідрологічних даних на ділянках русла на річці Дніпро (нижче за течією Києва), річки Десна біля Чернігова та ділянки річки Прип’ять поблизу м. Турів. Для побудови та навчання нейронних мереж було застосовано об'єктно-орієнтоване середовище програмування Python. Отримані результати показують, що для обчислення коефіцієнта шорсткості Шезі може бути достатнім використовувати ШНМ прямого поширення з одним прихованим шаром і сигмоїдною функцією активації. Формування якісного набору навчальних даних, а також організація даних і вибір відповідної обчислювальної моделі, заснованої на емпіричних знаннях, є в цьому випадку однією з більш актуальних проблем, ніж створення більш складних нейронних мереж.

Ключові слова: функції активації; штучні нейронні мережі; коефіцієнт Шезі; порівняння; dropout алгоритм; приховані шари; модифікації; нейрони.

Стаття надійшла до редакції 05.09.2023 і прийнята до друку після рецензування 29.11.2023

Ходневич Ярослав Васильович
кандидат технічних наук, науковий співробітник Інституту телекомунікацій і глобального інформаційного простору НАНУ
Адреса робоча: 03186 Україна, м. Київ, Чоколівський бульвар, 13
ORCID: https://orcid.org/0000-0002-5510-1154 ***e-mail:*** ya.v.khodnevych@gmail.com

Стефанишин Дмитро Володимирович
доктор технічних наук, провідний науковий співробітник Інституту телекомунікацій і глобального інформаційного простору НАНУ
Адреса робоча: 03186 Україна, м. Київ, Чоколівський бульвар, 13
ORCID: https://orcid.org/0000-0002-7620-1613 ***e-mail:*** d.v.stefanyshyn@gmail.com