Rocznik Ochrona Środowiska

Volume 24 Year 2022 ISSN 2720-7501 pp. 505-519

https://doi.org/10.54740/ros.2022.036 open access

Received: 09 December 2022 Accepted: 28 December 2022 Published: 29 December 2022

Comparison of MLP and RBF Neural Networks in the Task of Classifying the Diameters of Water Pipes

Zurab Gvishiani

Faculty of Civil Engineering, Georgian Technical University, Georgia
https://orcid.org/0000-0001-5134-0089

*Jacek Dawidowicz**

Faculty of Civil Engineering and Environmental Sciences,
Bialystok University of Technology, Poland
https://orcid.org/0000-0003-1742-9463

**corresponding author's e-mail: j.dawidowicz@pb.edu.pl*

Abstract: Hydraulic calculations of water distribution systems are currently performed using computer programs. In addition to the basic calculation procedure, modules responsible for evaluating the obtained calculation results are introduced more and more often into the programs. This article presents the results of research on artificial neural networks with a radial base function (RBF) and a multilayer perceptron (MLP), aimed at determining whether they can be used to model the relationship between the variables describing the computational section of the water distribution system and the diameter of the water pipe. The classification capabilities of the RBF and MLP networks were analyzed according to the number of neurons in the hidden layer of the network. A comparative analysis of RBF networks with multilayer perceptron (MLP) networks was performed. The results showed that the MLP networks have much better classification properties and are better suited for the task of assessing the selected diameters of the water pipes.

Keywords: water distribution system, hydraulic calculations, selection of diameters of water pipes, artificial neural networks, radial basis function, multilayer perceptron

1. Introduction

Hydraulic calculations of water distribution systems are currently performed using computer programs (Lansey & Mays 2000, Rossman 2000, Walski et al. 2003). In addition to the basic calculation procedure, modules responsible for evaluating the obtained calculation results are introduced more and more often into the programs. It should be noted that increasing the computing power of computers alone will not solve complex problems. Only the introduction of appropriate calculation methods allows you to get the right results. It seems that classical algorithms with a formalized course can be supplemented with much more advanced techniques of artificial intelligence, which include, among others, artificial neural networks (Bishop 1995, Konar 2006, Rutkowski 2008). There are many attempts to apply this solution to water supply issues (Czapczuk et al. 2015, Dawidowicz et al. 2018a, Dawidowicz 2018a), including calculations of water distribution systems (Czapczuk et al. 2017, Lingireddy & Brion 2005, Piasecki et al. 2018, Zhu et al. 2002). One of the tasks that can be carried out using artificial neural networks is the assessment of the correctness of the selection of the diameters of water pipes. Previously, the use of MLP multilayer perceptron networks (Dawidowicz 2018b) and Kohonen networks (Dawidowicz et al. 2021), as well as other artificial intelligence methods (Dawidowicz et al. 2018b) were considered for this purpose.

In this article, artificial neural networks with a radial basis function (RBF) and a multilayer perceptron (MLP) were tested to determine whether they can be used to model the relationship between the variables that describe the computational section of the water distribution system and the diameter of the conduit. The number of neurons in the hidden layer of RBF and MLP neural networks was analyzed. Then, a comparative analysis of RBF networks with MLP networks was performed.

2. Introduction to the Problems of Artificial Neural Networks

It is difficult to describe artificial neural networks in a few sentences. As a preliminary general definition of an artificial neural network, it can be stated that it is a network of neural cell models capable of acquiring, storing, and using knowledge resulting from previous operations. Artificial neural networks have in many cases turned out to be a useful tool for the implementation of practical tasks. They can be used anywhere there are problems related to prediction, classification, design, or control. The subject of artificial neural networks is very extensive. Many models of neurons and structures of neural networks are known and new ones are still appearing. In this article, the use of two types of neural network is considered:

- with radial basis functions in hidden layer neurons (RBF),
- multilayer perceptron (MLP).

Neural networks work in two stages. The first is learning based on a set of data that describe the problem to be solved. In fact, the network itself is useless until it is properly trained to perform a computational task. This is done using learning methods that allow one to determine the appropriate values of network parameters, namely weights and threshold values. Learning algorithms vary in computational complexity and efficiency. The second stage is the proper operation, in which the neural network should solve new tasks, using data not previously involved in teaching the network.

The training set has a limited number of training examples (patterns). However, this set should be representative, so it can be treated as a random sample taken from the population of all possible examples for a given problem. The network trained on a limited training set should generalize knowledge and determine the correct values for data that were not used in the learning process but belong to a specific population. The neural model should reflect the general form of the relationship sought, and not only adapt to points from the training set. This feature is defined as the ability to generalize (generalize) the acquired knowledge. In order to control the ability to generalize the network during its training, the data set is divided into three parts (Bishop 1996):

- training set – used to teach the network,
- validation set – cases from this set are not used to modify network parameters in the learning process, but for an independent assessment of quality and ability to generalize, carried out in parallel with the learning process,
- test set – it is not used at all during training, it allows for the final evaluation of the network operation quality after the training process is completed.

Different learning algorithms are used to train artificial neural networks of different types.

In the case of the model classification task, the neural network should assign the examples described by the components of the input vector to one of the predetermined classes of the nominal variable. The parameter evaluating the quality of neural network classification is the classification accuracy η defined as:

$$\eta = \frac{n_{cor}}{n_{all}} \tag{1}$$

where:

n_{cor} – the number of correctly classified training examples,

n_{all} – the number of all training examples subjected to classification.

2.1. Neural networks with radial basis function (RBF)

Networks, with a Radial Basis Function, belong to the group of feed forward neural networks. The radial network is a structure containing an input layer to which the signals are described by the input vector X= [x1, x2,…, x_N]^T, a hidden layer with K radial neurons and an output layer. The output layer has neurons with a linear activation function.

A radial neuron performs a function that varies radially around a fixed "center" C in the multidimensional R_N space. The radial function assumes nonzero values only in the vicinity of the "center", at a distance described by the threshold value of neuron b. The threshold value of neuron b represents the radius, i.e. the given maximum deviation, above which the distance is considered to be so great that the output signal takes values close to zero. Therefore, a radial neuron represents a hypersphere (hypersphere) that divides the space R_N around point C into a circle. A function of this type is generally denoted by the following formula:

$$\phi\|\mathbf{X}-\mathbf{C}\| \tag{2}$$

where:
$\|\cdot\|$ – the distance of the input vector X from the centre of the radial C neuron in the N-dimensional space.

The coordinates of the "center" C of the radial neuron of the R_N space are stored in its weight vector, that is:

$$\mathbf{C} = \mathbf{W} = [w_1, w_2,\ldots,w_i,\ldots, w_N]^T \tag{3}$$

Whereas the vector X means the input data to the neuron:

$$\mathbf{X} = [x_1, x_2,\ldots,x_i,\ldots, x_N]^T \tag{4}$$

Assuming that the vectors X and C describe two points that lie in the multidimensional space R_N, the radial neuron first calculates the distance r between the vectors X and C according to the adopted measure. The Euclidean measure is most often used to calculate distances:

$$r = \|\mathbf{X}-\mathbf{C}\| = \sqrt{\sum_{i=1}^{N}\left(x_i - w_i\right)} \tag{5}$$

The point C is fixed, while the point X is variable, because it marks successive input vectors. The output value y of a radial neuron is determined based on the distance r using one of the so-called radial functions. One of them is the bell-shaped Gaussian function, whose shape is governed by a parameter σ called width (a parameter of smoothness or flatness):

$$\varphi(r) = \exp\left(-\frac{r^2}{2\sigma_i^2}\right) \tag{6}$$

where:
σ – the shape factor.

The role of the neuron, hidden in the RBF networks, consists in the radial mapping of the space around the C centre (Fig. 1). The location of the neuron centre, in the N-dimensional space, is determined in the network learning process. The superposition of signals, coming from all K neurons, hidden in the RBF network, allows all network centres to be mapped thus:

$$F(X)=\sum_{i=1}^{K}W_i\varphi\left(\left\|X-C_i\right\|\right) \tag{7}$$

where
K – the number of neurons in the hidden radial network.

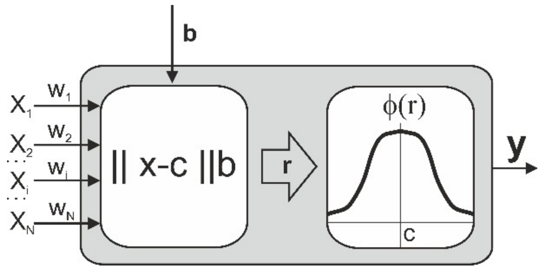

Fig. 1. Diagram of a radial neuron

The number of K neurons in the hidden layer and C centers should be determined in the process of the neural network training. However, the condition required for the RBF network to create an effective model for any function is the provision of a sufficient number of radial neurons in the network structure. The number of radial neurons cannot be too large because it will not have the ability to generalize.

The training of radial networks consists of two stages:
- selection of centres and deviation from the centre, i.e. the smoothing coefficient,
- selection of neuron weights in the output layer.

The selection of C centers of the radial neurons was performed using the k-means algorithm (Hartigan & Wong 1979). The size of the deviation, that is, the smoothing coefficient, was determined using the k-nearest-neighbor method.

The selection of weight values of the output neurons was made using the standard linear optimization technique, which is a pseudo-inversion algorithm, that is, decomposition, according to singular values (Bishop 1996). The quality of training of the output layer was evaluated using the sum-squares error function SOS, as described by the formula:

$$E_{SOS}=\sum_{i=1}^{P}(y_i-t_i)^2 \tag{8}$$

where:
P – the number of training cases,
y – the value at the output of the neural network,
t_i – the target value of the network for the i-th learning case.

2.2. Multilayer perceptron neural networks (MLP)

The most commonly used type of unidirectional artificial neural network is the multilayer perceptron (MLP), which consists of neurons arranged in layers. Neurons are connected between layers on a peer-to-peer basis, while in one layer there are no connections between neurons. A weighting factor is assigned to each connection. The above connection method is used, because in the learning process the weights of irrelevant connections will be zeroed, which means that there is no connection. There are three basic types of layers: input layer, hidden layer, and output layer. The signals of hidden-layer neurons are not directly accessible. Unidirectional multilayer networks always have an input layer and an output layer and at least one hidden layer. Neurons in the hidden and output layers use a logistic activation function (Fig. 2):

$$y = \frac{1}{1+e^{-\beta S}} \qquad y \in (0...+1) \tag{9}$$

where:
β – a numerical factor usually equal to 1,
S – aggregate value of the neuron.

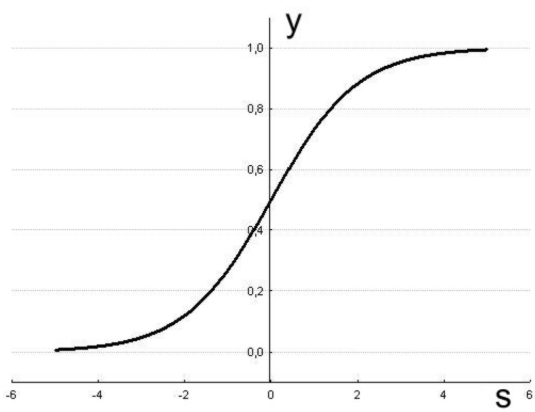

Fig. 2. Logistic function graph at β = 1

In the neurons of the output layer in a multilayer perceptron adapted to solve classification problems, the Softmax activation function is used. It is an exponential function, the value of which is additionally normalized in such a way that the activation sum of all M neurons in the output layer of the network is equal to 1. In addition to the fact that signals from the network are the basis for recognizing the appropriate class, additionally, the output values of individual neurons can be interpreted as estimating the probabilities of a given output signal belonging to particular classes (Bridle 1990):

$$y = \frac{e^{S}}{\sum_{m=1}^{M} e^{S_m}} \qquad y = (0...+1) \tag{10}$$

Multilayer perceptron networks are trained using a strategy with the teacher that is iterative and involves repeatedly presenting a network of teaching examples {X, d}. The examples presented boil down to starting the network for subsequent input vectors $X = [x_1, x_2,..., x_i,..., x_N]^T$, and then comparing the values obtained in the output y with the values set as standards of correct answers d. The error function E is then calculated, which is a measure of the discrepancy between the current states of the network outputs y and the standard values d. In one cycle, the presentation of all the examples in the learning collection is called an epoch. Learning algorithms work for a certain number of epochs.

The method of data processing by a neural network depends on the weights of connections between neurons, hence it is very important to properly determine their values. Teaching methods for unidirectional multilayer perceptron networks and with radial basis functions modify only the weights, leaving the network architecture unchanged. It follows that the first thing to decide is the number and type of neurons in each layer. Then, in the learning process, the weights of all neurons are determined, in an effort to minimize the error function. This task is performed by an appropriate learning algorithm. The learning process starts with initializing all network weights as non-zero numbers with small values.

There are many methods to teach a multilayer perceptron network, determining when and how the weights of connections between neurons change:

- back propagation,
- quick propagation,
- Conjugate Gradient Descent,
- Levenberg-Marquardt,
- Quasi-Newton (BFGS variable metric method).

Training of a multilayer perceptron performing the classification task consisting of recognizing one of several classes, where each class corresponds to one neuron in the output layer (M > 1) is performed using the entropy multiple error function:

$$\mathbf{E_{EME}} = -\sum_{t=1}^{T} \sum_{m=1}^{M} d_m^{(t)} \log y_m^{(t)} \tag{11}$$

where:
T – the number of training cases,
t – the number of the training case,
M – number of neurons in the output layer,

m – neuron number in the output layer,
y – the value at the output of the neural network,
d – the target value of the network.

3. Studies of network structures to classify the diameters of water pipes

To compile a data set for training neural networks, information was collected on 33 existing water supply systems. Hydraulic calculations were made for them for various variants of water intake from nodes and water pipes, to obtain the widest possible range of data for training neural networks. The calculations were performed using a program that takes into account the sectional water intake of the water pipes. Due to the large amount of data, a procedure has been developed to transform the calculation results for individual sections into the appropriate format and save them in a set of training examples. Calculations were made for different values of the absolute roughness coefficient and for the internal diameters of the water pipes. Based on the results of hydraulic calculations, 13923 training examples were obtained, containing input variables and the corresponding output variables, necessary for supervised learning. Each calculation variant was checked and corrected for irregularities in the calculation.

The input vector X consists of the following variables:

- Q_b – flow rate at the beginning of the calculation section [l/s],
- Q_e – flow rate at the end of the calculation section [l/s],
- q_{sec} – sectional output, water cutting along the length of the section [l/s],
- k – absolute roughness coefficient [mm],
- L – length of the calculation section of the water supply network [m].

The output variable in the set of training examples is the nominal variable DIA consisting of classes describing the nominal diameters of water pipes: DN90, DN110, DN160, DN225, DN250, DN300, DN350, DN400, DN450, DN500.

Artificial neural networks for different numbers of neurons in the hidden layer for multilayer perceptrons and networks with radial base functions were analyzed. The initial number of neurons in the latent layer was calculated based on Kolmogorov's theory, where at the N-dimensional input vector X, the number of hidden neurons is (Bishop 1996):

$$K = 2 \cdot N + 1 \tag{12}$$

where:
N – the number of variables in the input vector of the neural network X,
K – the number of neurons in the hidden layer of the neural network.

Then, the number of neurons in the inner layer was increased by the initial value.

3.1. Studies of network structures with radial basis functions (RBF)

Neural networks with radial activation function were trained, starting with the number of neurons in the inner layer of 11. The maximum number of neurons in the inner layer is 1265.

Table 1. Neural networks with radial base functions for estimating the diameters of water pipes on computational sections

Basic data of the neural network:

Number of inputs: 5
Input variables: Q_b, Q_e, q_{sec}, k, L
Output variable: DIA (nominal)
Hidden layer activation function: radial
Output Layer Activation Function: Linear
Error function: the sum-squares error function E_{SOS} by (8)

List of neural networks tested:

No.	K	E_{SOS} (L)	E_{SOS}(V)	E_{SOS}(T)	η(L)	η(V)	η(T)
1	11	0.2683137	0.2690769	0.2678924	0.4476368	0.4421143	0.4530307
2	22	0.2533979	0.2564669	0.2530448	0.5530815	0.5354783	0.5486929
3	33	0.2453031	0.2471132	0.2457670	0.6066657	0.5909221	0.6047113
4	44	0.2415342	0.2433118	0.242149	0.6431547	0.6236714	0.6308532
5	55	0.2366534	0.2388809	0.238103	0.6198822	0.5960931	0.6150531
6	110	0.2228965	0.2270652	0.2259353	0.6914236	0.6627406	0.6851479
7	165	0.2160266	0.22005	0.2195365	0.7125413	0.6894571	0.7006607
8	220	0.212763	0.2172975	0.2173149	0.727338	0.6995116	0.7178972
9	275	0.2088242	0.2150937	0.2148252	0.7418474	0.7147371	0.7270899
10	330	0.2069361	0.2125766	0.2124427	0.7467318	0.7176099	0.7331227
11	440	0.2083044	0.2138254	0.2134634	0.7322224	0.7066935	0.7210572
12	550	0.1927873	0.1999336	0.2004794	0.7924149	0.7563918	0.7667337
13	660	0.1828443	0.1934866	0.1927762	0.8195662	0.7882792	0.7940247
14	770	0.2113783	0.2156025	0.2324807	0.8346502	0.787992	0.8026429
15	990	0.1939883	0.2016957	0.2144034	0.8524637	0.8086757	0.8118357
16	1100	0.3162278	0.3162278	0.3162278	0.1370493	0.1378914	0.1519678
17	1265	0.3162278	0.3162278	0.3162278	0.1370493	0.1378914	0.1519678

K – number of neurons of the inner layer, E_{SOS}(L) – the error for the training subset, E_{SOS}(V) – the error for the validation subset, E_{SOS}(T) – the error for the test subset, η(L) – relevance of the classification for the training subset, η(V) – relevance of the classification for the validation subset, η(T) – relevance of the classification for the test subset.

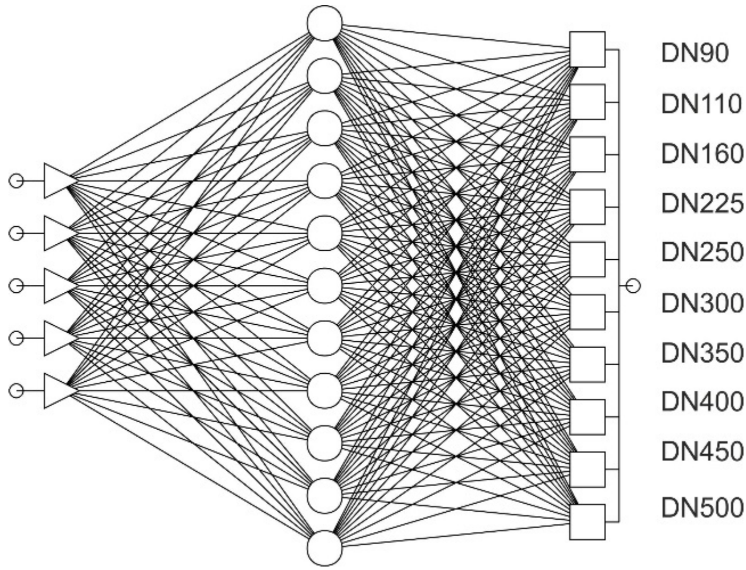

Fig. 3. Diagram of an exemplary network with radial neurons to classify the diameters of water pipes

3.2. Studies of multilayer perceptron structures (MLP)

In multiclass problems, the number of output neurons is equal to the number of object types available for classification, in this case the nominal diameters of water pipes of the nominal variable DIA. Determination of belonging to one of the classes consists of selecting a neuron of the output layer, in which a value close to 1.0 appears, and in the remaining neurons there should be values close to 0.0. The ideal state would be if the values obtained in individual neurons of the output layer were exactly 1.0 or 0.0, but in practice it is impossible to obtain. For this reason, two threshold values were introduced: the acceptance threshold and the rejection threshold, with which the level of activation of neurons of the output layer is compared. The activation level above the acceptance threshold results in accepting the object's belonging to the class, while the activation value below the rejection threshold proves that the object does not belong to the class. In a multiclass task, one neuron characterizing the selected class should be above the acceptance threshold, the rest below the rejection threshold. If this condition is not met, the case is described as undefined, i.e. the network is not able to classify the object into any class. In this task, the acceptance threshold of 0.95 and the rejection threshold of 0.05 were adopted.

The networks were first trained using the backpropagation method. Case mixing was used, randomly changing the order of introducing teaching cases from epoch to epoch. The backpropagation method in the initial stage gives the best learning results with the learning coefficient η in the range of 0.1-0.05 and the torque coefficient μ in the range of 0.3-0.1. This ensures that the value of the error function decreases along the main curvature of the error surface without falling into local minima. In the final stage, the best results were obtained using the quasi-Newton method or the backpropagation method with the lowest possible values of the learning coefficient η = 0.0001 and the torque μ = 0.0001.

Five multilayer perceptrons (Table 2) were obtained, in which the classification quality of the test set ranged from 0.887676 to 0.9376616. Networks with more neurons in the inner layer were prone to overfitting. Figure 4 shows an example of a multilayer perceptron to classify the diameters of water pipes with marked activation of neurons.

Table 2. Multilayer perceptron neural networks to estimate the diameters of water pipes in computational sections

Basic data of the neural network:

Number of entrances: 5
Input variables: Q_b, Q_e, q_{sec}, k, L
Output variable: DIA (nominal)
Hidden-layer activation function: Logistic
Output layer activation function: Softmax by (10)
Error function: Entropy (multiple) E_{EME} by (11)
Acceptance threshold: 0.95 Rejection threshold: 0.05

List of neural networks tested:

No.	K	E_{EME} (L)	E_{EME}(V)	E_{EME}(T)	η(L)	η(V)	η(T)
1	11	0.2310275	0.2611295	0.2718562	0.8810516	0.8787705	0.887676
2	22	0.2219108	0.2814595	0.3344539	0.8982905	0.8948578	0.9072106
3	33	0.2118822	0.2604122	0.3685451	0.9281712	0.9103706	0.9230106
4	44	0.209799	0.249185	0.2863754	0.9127999	0.9043378	0.9184142
5	55	0.19162	0.266959	0.3767445	0.9408131	0.9247343	0.9376616

K – number of neurons of the inner layer, E_{EME}(L) – the error for the training subset, E_{EME}(V) – the error for the validation subset, E_{EME}(T) – the error for the test subset, η(L) – relevance of the classification for the training subset, η(V) – relevance of the classification for the validation subset, η(T) – relevance of the classification for the test subset.

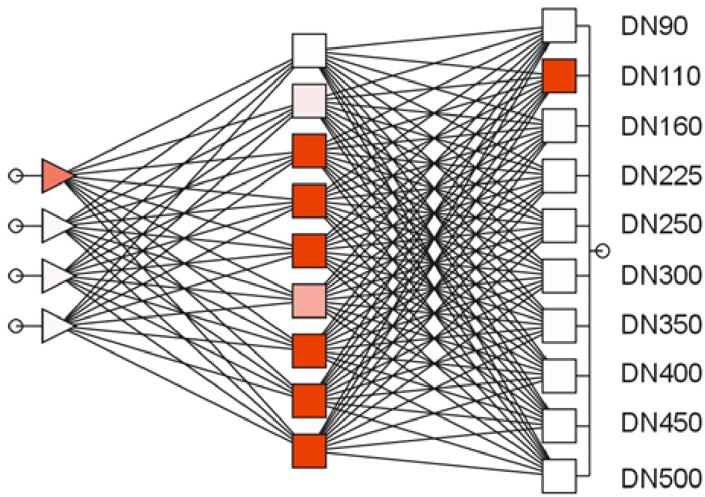

Fig. 4. Scheme of an example MLP neural network for pipe diameter classification with neuron activation

4. Comparative analysis of MLP and RBF networks for the classification of water pipe diameters

Based on the simulation of neural networks, the MLP multilayer perceptron and radial basis functions RBF were used to compare the classification accuracy η for the same number of neurons in the hidden layer (Fig. 5). It is clearly visible that the multilayer perceptron is characterized by a much better classification quality than the network with radial basis functions.

Research on the RBF network shows that the increasing number of neurons increases the classification accuracy, but does not exceed the value for the multilayer perceptron. For the number of neurons 1100 and above, a step deterioration in the quality of the network operation with radial base functions is observed (Fig. 6).

The lowest classification accuracy value for a multilayer perceptron is higher than the highest value for networks with a radial basis function. The best multilayer perceptron with 55 neurons in the hidden layer is characterized by the classification accuracy for the test subset at the level of 0.9376616.

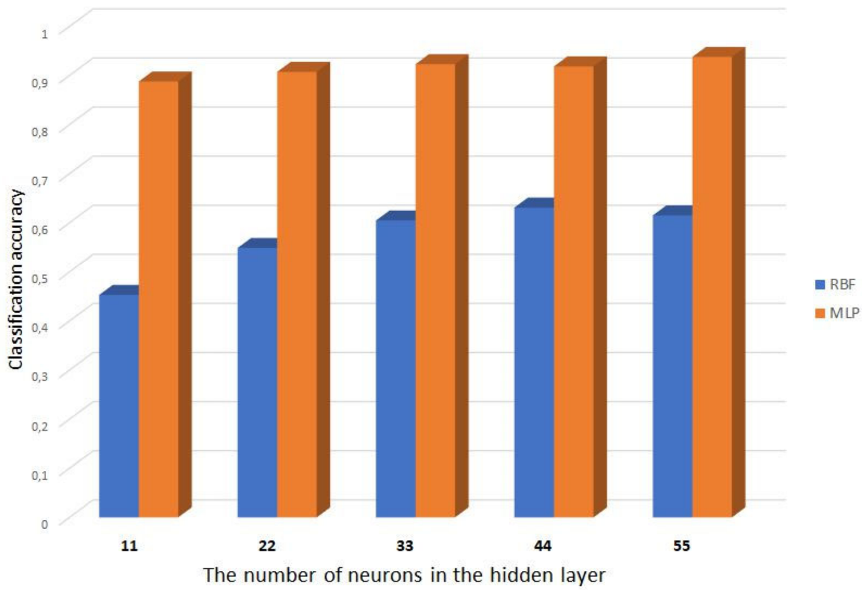

Fig. 5. Classification accuracy for the multilayer perceptron and RBF network for the test subset

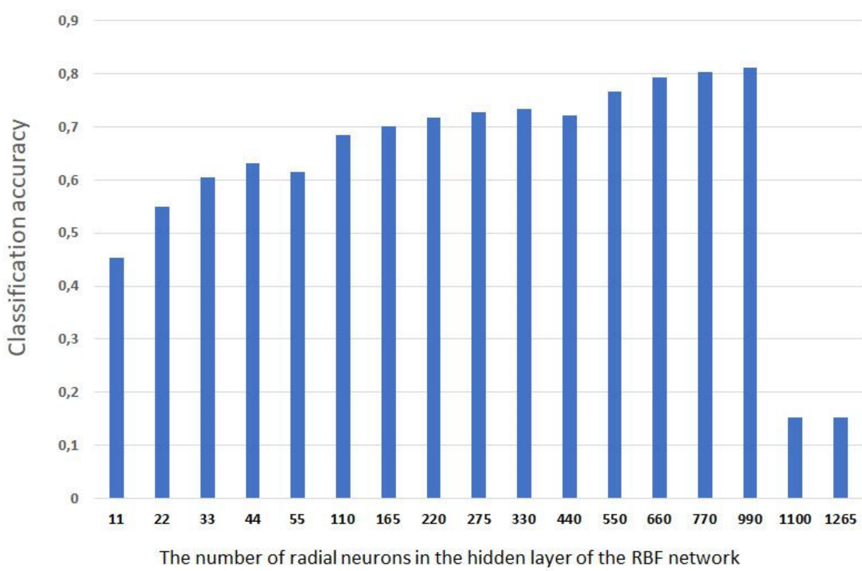

Fig. 6. Classification accuracy η for the RBF network for the test subset

Summary

The motivation for the research was the idea of creating additional modules to evaluate the results obtained in the computer software for the calculation of water distribution systems. In order to obtain the correct solution, calculations are conducted many times, and hence intelligent solutions would certainly be very useful at the design stage.

The article describes two types of unidirectional neural networks. Using the training data set, 17 neural networks with radial basis functions and 5 multilayer perceptron networks were prepared. The results obtained indicate that the multilayer perceptron is a much better classifier in the task of selecting the diameter of a water pipe on the basis of variables describing it from hydraulic calculations. Increasing the number of neurons in the hidden layer of the RBF network slightly increases the classification accuracy, but after exceeding the value of 1100 neurons, the quality of the network decreases significantly.

References

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Oxford: Oxford Univ. Press.

Bridle, J.S. (1990). Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition, In: *Neurocomputing: Algorithms, Architectures and Applications* (227-236). Berlin, Heidelberg: Springer-Verlag.

Czapczuk, A., Dawidowicz, J., Piekarski, J. (2015). Artificial Intelligence Methods in the Design and Operation of Water Supply Systems. *Rocznik Ochrona Srodowiska*, *17*, 1527-1544 (in Polish).

Czapczuk, A., Dawidowicz, J., Piekarski, J. (2017). Application of Multilayer Perceptron for the Calculation of Pressure Losses in Water Supply Lines. *Rocznik Ochrona Srodowiska*, *19*, 200-210.

Dawidowicz, J. (2018a). Evaluation of a pressure head and pressure zones in water distribution systems by artificial neural networks. *Neural Computing & Applications*, *30*(8), 2531-2538. DOI: 10.1007/s00521-017-2844-8

Dawidowicz, J. (2018b). A Method for Estimating the Diameter of Water Pipes Using Artificial Neural Networks of the Multilayer Perceptron Type. In: *2nd International Conference on Artificial Intelligence: Technologies and Applications (ICAITA 2018)*, (50-53). Atlantis Press. DOI: 10.2991/icaita-18.2018.13

Dawidowicz, J., Czapczuk, A., Kruszynski, W. (2021). Kohonen Artificial Networks for the Verification of the Diameters of Water-pipes. *Rocznik Ochrona Srodowiska*, *23*, 835-844. DOI: 10.54740/ros.2021.057

Dawidowicz, J., Czapczuk, A., Piekarski, J. (2018a). The Application of Artificial Neural Networks in the Assessment of Pressure Losses in Water Pipes in the Design of Water Distribution Systems. *Rocznik Ochrona Srodowiska*, *20*, 292-308.

Dawidowicz, J., Kruszynski, W., Andraka, D., Czapczuk, A. (2018b). Assessing the Diameters of Water Pipes Using the k-Nearest Neighbours Method in the Calculations of Water Distribution Systems. *Rocznik Ochrona Srodowiska*, *20*, 528-537.

Hartigan, J.A., Wong, M.A. (1979). Algorithm AS 136: K-Means Clustering Algorithm. *Applied Statistics*, *28*, 100-108.

Konar, A. (2006). *Computational intelligence: principles, techniques and applications*. Springer Science & Business Media. New York.

Lansey, K., Mays, L.W. (2000). Hydraulics of water distribution systems. In: Mays LW (ed) *Water distribution systems handbook*. New York: McGraw-Hill.

Lingireddy, S., Brion G.M. (2005). Artificial neural networks in water supply engineering. In: Lingireddy S, Brion GM (eds) *Artificial neural networks in water supply engineering*, (1-9). Cincinnati: ASCE.

Piasecki, A., Jurasz, J., Kaźmierczak, B. (2018). Forecasting daily water consumption : a case study in Torun, Poland. Periodica Polytechnica. Civil Engineering. 62(3), 818-824.

Rossman, L.A. (2000). *EPANET 2 users manual*. Cincinnati: U.S. Environmental Protection Agency, Office of Research and Development, National Risk Management Research Laboratory.

Rutkowski, L. (2008). *Computational intelligence: methods and techniques*. New York: Springer Science & Business Media.

Walski, M.T., Chase, D.V., Savic, D.A., Grayman, W.M., Beckwith, S., Koelle, E. (2003). *Advanced water distribution modeling and management*. Waterbury: Haestad Methods Solution Center, Haestad Press.

Zhu, D., Zhang, T., Mao, G. (2002). Back-propagation artificial neural networks for water supply pipe line model. *Tsinghua Sci Technol*, *7*(5), 527-531.