Research the adaptive neural networks using possibility in control systems for cycle chemistry at thermal power plants

Olga Yegoshina[1*], *Sofya* Lukutina[1], *Egor* Bezuglov[1], and *Alexandra* Ivanova[1]

[1]National Research University «Moscow Power Engineering Institute», 111250 Moscow, Russia

Abstract. The study is devoted to the analysis of the learning possibilities of adaptive neural networks in the context of water-chemical regime management systems. During a series of experiments on a laboratory stand with a disturbing effect, transient characteristics describing changes in the controlled value were obtained. The obtained mathematical models using the TunePID program served as the basis for the analysis and research of adaptive neural networks. Adaptive neural networks are a powerful tool capable of learning from the data obtained and dynamically changing their structure and parameters to accurately predict and control the behavior of the system. Unlike classical mathematical models, adaptive neural networks have the ability to process nonlinear dependencies and adapt to changing conditions without the need for manual reconfiguration. The stages of identification of disturbances and adaptation of compensator settings were described and illustrated. The study included an analysis of the work of a neural network with a different number of hidden neurons and various activation functions. The results showed that neural networks trained on data using an adaptive approach are able to predict and control the system with an accuracy comparable to the calculated parameters of the control coefficients calculated manually.

1 Introduction

Currently, control and regulatory systems require more effective approaches to ensure their accurate and stable operation. This is especially true for energy facilities such as thermal power plants (TPP), where any violations of the quality standards of the coolant of power units can lead to serious consequences [1, 2].

Modern thermal power plants often use traditional control methods, such as manual control or proportional control based on changes in feed water. However, these methods may be ineffective and insufficiently accurate, which may lead to violations of regulatory indicators and uneconomical use of reagents.

In this context, the study of the possibility of learning adaptive neural networks represents an important step in the development of control and regulation systems at thermal power plants. Adaptive neural networks have the ability to learn from data and dynamically

* Corresponding author: yegoshinao@yandex.ru

change their structure and parameters to more accurately predict and control the behavior of the system. The purpose of this study is to assess the learning potential of adaptive neural networks in the context of automatic dosing at thermal power plants. The main task is to develop and test neural network training methods based on data obtained during experiments on a laboratory bench simulating changes in the flow rate of treated water. In addition, a comparative analysis of the effectiveness of neural networks in comparison with traditional methods of management and regulation is carried out.

It is known [3] that the use of adaptive neural networks in control and regulation systems increases their efficiency and accuracy, which makes it possible to automate the control processes by dosing corrective reagents and reduce the likelihood of violation of coolant quality standards. Thus, the study of adaptive neural network learning is of great practical importance for improving the operation of energy facilities.

2 Materials and methods

The study of the possibility of learning adaptive neural networks is a significant step in the development of modern water-chemical regime management systems. Their ability to automatically adapt to changing conditions makes them very promising for use in the energy sector. Unlike traditional mathematical models, adaptive neural networks have the ability to analyze nonlinear connections and automatically adapt to changing conditions without requiring manual calibration.

As part of the study, a series of experiments was carried out on a laboratory stand with a disturbing effect on changing the flow rate of treated water from 20 to 50 l/h in increments of 3-4 l/h [4]. The data obtained from the pH meter and conductometer were processed and used to approximate mathematical models of the system using the TunePID tuning program. Figures 1-2 show the transient characteristics of 2 series of experiments out of 16 when the flow rate of treated water changes from 20 to 50 l/h and from 43 to 50 l/h.

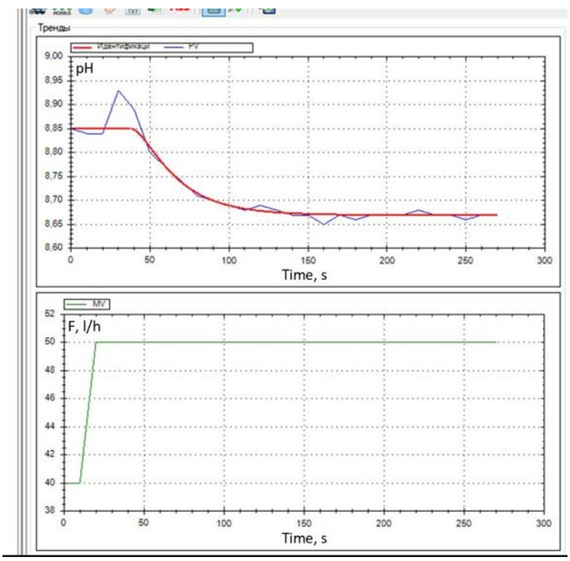


Fig. 1. Transient characteristics when changing the flow rate of treated water from 20 l/h to 50 l/h.

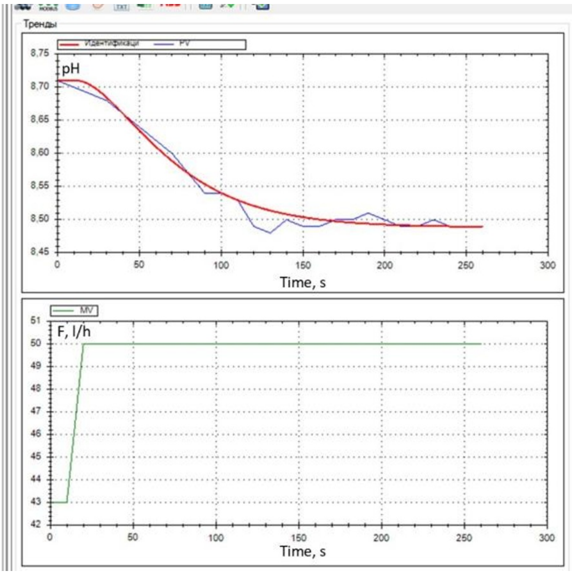
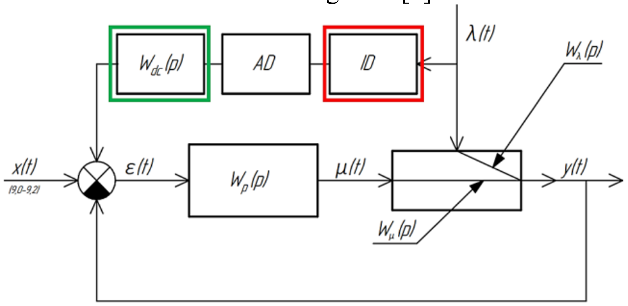


Fig. 2. Transient characteristics when changing the flow rate of treated water from 43 to 50 l/h.

A block diagram of an automatic dosing system using adaptive neural networks for effective control of disturbances is shown in Figure 3 [5].



x(t) – task; $W_p(p)$ – controller (on PI algorithm); W_μ (p) – transfer function of the object along the channel μ-y; W_λ (p) – transfer function of the object along the channel λ-y; y(t) – the output of the object; $W_{dc}(p)$ – the disturbance compensator

Fig. 3. A block diagram of automatic dosing using neural networks.

At the first stage of the study, disturbances are identified (ID), which is achieved by obtaining experimental characteristics of analyzers such as a pH meter and a conductometer, depending on the magnitude of the applied disturbance, followed by approximation of experimental data. The received information is transmitted to the adaptation unit (AD), where, based on the updated data, the optimal compensation parameters are calculated, after which the signal is sent from the compensator to the regulator [5]. Since the nature of the disturbance is changeable, the corresponding object along the disturbance channel is also subject to changes, which requires recalculation of the compensator parameters. Therefore, effective correction requires adaptation to the conditions of the object by adjusting it.

One of the key aspects of the study is to assess the learning potential of adaptive neural networks in the context of automatic dosing. Neural networks are a complex network of artificial neurons (Figure 4) that interact with each other and are able to automatically learn and adapt to changes in environmental conditions.

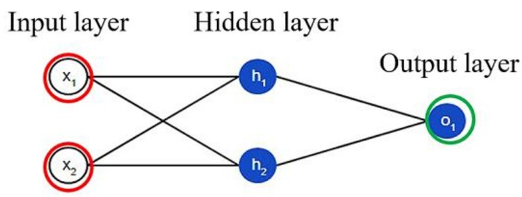
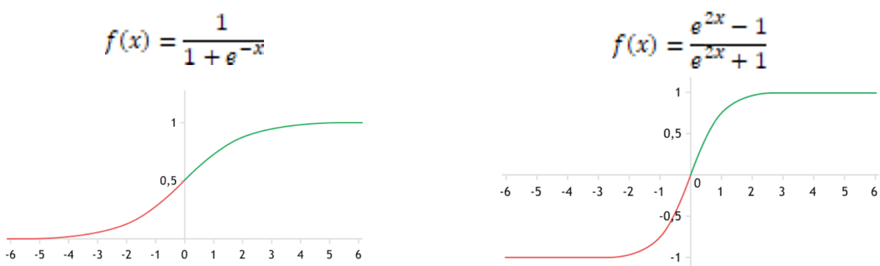


Fig. 4. An example of how a neural network works.

On the input layer of the neural network, the input parameters of mathematical models are accepted, such as coefficients k, τ, T_1, T_2, then information is processed on hidden layers, after which the output layer forms the settings of the compensator: K_d, T_d [6-7].

Figure 5 shows activation functions, which are a way to normalize input data. They allow you to limit the output values to the desired range, regardless of the size of the input data. Among the presented activation functions, the main attention is paid to the sigmoid (logistic) and hyperbolic tangent. These activation functions are characterized by different ranges of values, which allows you to adapt the output data to the necessary requirements. And they are also preferable for shallow neural networks with one hidden layer, while others (Rectified Linear Unit, Leaky ReLU, Softmax) are used to train deep neural networks with several layers of hidden neurons.



a) b)

Fig. 5. Activation functions: a) sigmoid and b) hyperbolic tangent.

To work with the neural network, Python code has been written in the form of a class with initialization, forward and reverse propagation functions. This class is a module ready to be called as part of working with a neural network. The initialization function sets the initial parameters and weights. The direct propagation function transmits input data through a neural network, calculating output values. The back propagation function adjusts the weights to minimize the error. This class is a key component of the neural network, ready for use for data processing and performing computational tasks. Figures 6 and 7 show neural networks with two activation functions [5].

```
# Sigmoidal activation function
def sigmoid(x):
    return 1 / (1 + np.exp(-x))

# Derivative of the sigmoidal activation function
def sigmoid_derivative(x):
    return x * (1 - x)
```

Fig. 6. The activation function of neural networks is sigmoid.

```
# Hyperbolic Tangent (Tanh) activation function
def tanh(x):
    return np.tanh(x)

# Derivative of the hyperbolic tangent function (Tanh)
def tanh_derivative(x):
    return 1 - np.tanh(x)**2
```

Fig. 7. The activation function of neural networks is hyperbolic tangent.

The data processing was carried out using a neural network consisting of three neurons in a hidden layer, each of which is activated by a sigmoid function. Figure 8 illustrates the dimensions of the input data: 4 (k, τ, T_1, T_2), the number of hidden layers (3) and the number of output data: 2 (expected settings of the compensator K_d, T_d). An instance can be considered as an object created based on a specific class (in this case, a neural network class) that contains specific values and methods specific to that object.

```
# Neural network parameters
input_size = 4
hidden_size = 3
output_size = 2

# Creating an instance of a neural network
model = NeuralNetwork(input_size, hidden_size, output_size)

# Training parameters
learning_rate = 0.1
epochs = 200001
```

Fig. 8. Creating an instance of neural networks with 3 hidden neurons (activation function – sigmoid).

The work cycle of a neural network is a key stage of its functioning, where operations are repeated to update internal parameters and improve the quality of forecasting. In all calculations, the neural network goes through 200,000 iterations, calculating the output values and estimating the error. Every 50,000 iterations, it provides a degree of error for monitoring and monitoring progress (evaluating the effectiveness or success of the neural network learning process). When the neural network goes through 50,000 iterations, the error is evaluated and analyzed to determine how well the model is currently learning. This allows you to track learning progress, identify problems, and adjust the learning process if necessary to achieve better results. This approach allows you to effectively configure the parameters of the neural network and improve its performance to achieve the desired results [8-9].

We apply a new set of input data, represented as an array of dynamic characteristics of the identification object (k, τ, T_1, T_2) to already prepared and trained neurons in our neural network (Figure 9) [10]. After passing through the network and processing these inputs, we get the calculated output values. This process provides the neural network with the ability to adapt to a variety of input data and effectively solve tasks, which ultimately allows it to obtain accurate and reliable results in accordance with its training.

```
# An example of using a trained neural network for new input data
new_inputs = np.array([[0.02073,30.2475,8.4257,27.9335]])
# Normalization of new input data
normalized_new_inputs = (new_inputs - min_inputs) / (max_inputs - min_inputs)
predicted_outputs = model.forward(normalized_new_inputs)
# Denormalization of output data
denormalized_predicted_outputs = predicted_outputs * (max_outputs - min_outputs) + min_outputs
print("Prediction of output values for new input data:", denormalized_predicted_outputs)
print("Expected output: [0.022424,65.725516]")
```

```
Prediction of output values for new input data: [[2.16009504e-02 6.04420502e+01]]
Expected output: [0.022424,65.725516]
```

Fig. 9. An example of using a trained neural network with response output.

The experiment investigated the effectiveness of a neural network with a different number of hidden neurons (3, 15, 30) and an activation function in the form of a sigmoid and a hyperbolic tangent. The proposal to expand the number of neurons in the hidden layer is due to the desire to improve the ability of the neural network to adapt and process diverse and more complex data.

3 Results

The results of calculations of the neural network with the sigmoid activation function (sigma) are presented in summary Table 1.

When using a neural network with 3 hidden neurons, high accuracy and convergence to the desired result were observed, which is confirmed by a low error rate of 0.003405 after 200,000 iterations.

An increase in the number of hidden neurons to 15 led to a further improvement in the quality of the neural network. The degree of error decreased to 0.002847, which indicates a more accurate response from the neural network (the difference between the expected and received response is minimal). The calculations were carried out on the same data sets. However, when using 30 hidden neurons, there is an increase in error to 0.002884. This indicates a retrain of the neural network when the model is too adapted to the training data and loses the ability to generalize to new data.

Thus, the selection of the optimal number of hidden neurons is an important point in training a neural network, since this affects its ability to generalize and the accuracy of predictions. In this context, calculations were performed using an alternative activation function (hyperbolic tangent- tanh), and the effect of using a different number of hidden neurons (3, 15 and 30) was also analyzed. The results are also summarized in table 1.

Table 1. The results of the calculation of a neural network with 3, 15, 30 hidden neurons and various activation functions.

Number of neurons	3 hidden neurons		15 hidden neurons		30 hidden neurons	
Settings Calculation method	K_d	T_d	K_d	T_d	K_d	T_d
Manual	0,022424	65,725	0,022424	65,725	0,022424	65,725
Neural network, sigm	0,024044	62,684	0,021597	61,113	0,021883	60,568
Neural network, tanh	0,024547	59,014	0,023069	61,134	0,023327	60,050

Note: «sigm» is the sigmoid activation function, «tanh» is the hyperbolic tangent activation function.

Based on calculations and analysis of Tables 1-2, transients of the dosing system of corrective reagents were constructed (Figure 10), the settings of which were calculated in various ways: manually using formulas and using neural networks. The results were verified using integral quality criteria (linear, modular, quadratic), which further confirms the observed differences in the effectiveness of various neural network configurations.

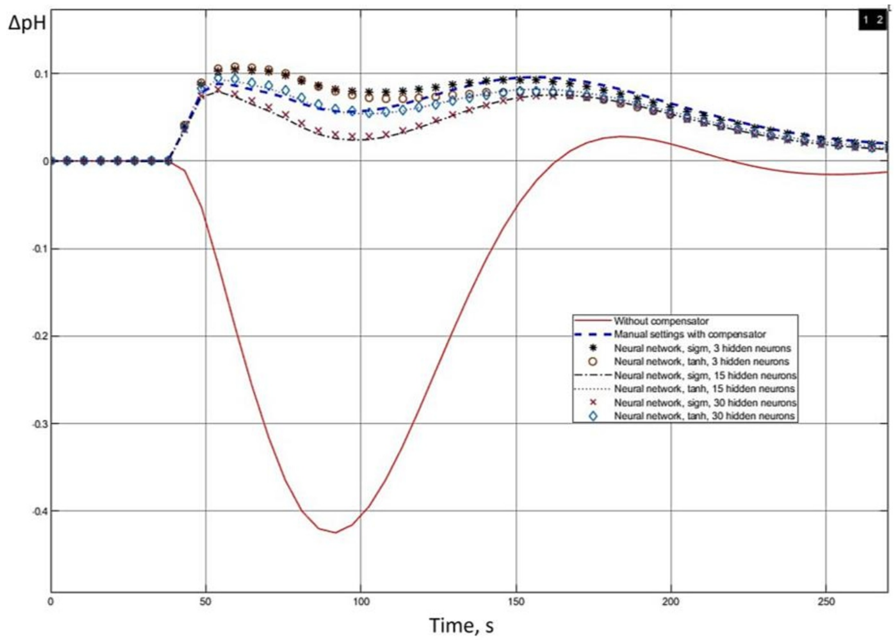


Fig. 10. Transients of the automatic dosing system of reagents.

Table 2 shows that the results of evaluating the quality of the automatic dosing system of corrective reagents with different numbers of neurons and activation functions have differences. In particular, the best result was shown by a neural network with 15 hidden neurons and a sigmoid activation function having the lowest values of integral quality criteria (I_{lin} = 11.65, $|I_{module}|$ = 11.65, I^2_q = 0.6061), which indicates a more efficient and accurate prediction. In comparison with another option, such as a neural network with 30 hidden neurons and a hyperbolic tangent activation function, it shows that although the number of hidden neurons has increased, the result has not improved, which indicates the need for a more careful choice of neural network architecture and activation function depending on the characteristics of the task and the available data.to specify for our case

Table 2. Integral quality criteria.

Calculation method	I_{lin}	$\lvert I_{module} \rvert$	I^2_q
Without compensator	-27,8	29,88	8,958
Manual settings with compensator	16,16	16,16	1,098
Neural network, sigm, 3 hidden neurons	17,2	17,2	1,295
Neural network, tanh, 3 hidden neurons	15,52	15,52	1,105
Neural network, sigm, 15 hidden neurons	**11,65**	**11,65**	**0,6061**
Neural network, tanh, 15 hidden neurons	14,46	14,46	0,9119
Neural network, sigm, 30 hidden neurons	11,81	11,81	0,6211
Neural network, tanh, 30 hidden neurons	14,37	14,37	0,9112

4 Discussion

From the analysis of the data presented on the graph, several conclusions can be drawn: tangential neural networks with 15 and 30 hidden neurons performed better in predicting coefficients, since the graphs are almost identical to the calculated ones. In our case, sinusoidal ones with 15 and 30 hidden neurons coped with the task better, since they have the lowest values according to integral quality criteria. Neural networks with 3 hidden neurons performed worse than the rest, 30 hidden neurons lead to retraining of the neural network.

5 Conclusions

1. The possibility adaptive neural networks synthesis is shown in cycle chemistry control systems. A neural network has been developed; tuning coefficients has been obtained for the compensator in the training data sample range. The difference has been established between systems with compensation and neural network adaptation. The difference lies in the software code for the controller.
2. The neural network using with a sinusoidal activation function and 15 hidden neurons showed the best result of the studied options in cycle chemistry control systems.

Acknowledgment

The work was carried out as part of the project «Research and development of a smart monitoring and control system for the cycle chemistry at new power units» with the support of a grant from «MPEI» for the implementation of the scientific research program «Priority 2030: Future Technologies» in 2022-2024.

References

1. O. Yegoshina, S. Zvonareva, N. Bolshakova / Thermal Engineering, 2023, **Vol. 70**, No. 5, pp. 362–369, doi: 10.1134/S0040601523050014.
2. O. Yegoshina, N. Bolshakova and S. Zvonareva, *The State and Perspectives of Corrective Reagents Dosing Systems as the Cycle Chemistry Monitoring at Power Plants*, 2022 VI International Conference on Information Technologies in Engineering Education (Inforino), 2022, pp. 1-4, doi:10.1109/Inforino53888.2022.9782961
3. V.A. Terekhov, D.V. Efimov, I.Yu. Tyukin. *Neural network control systems: a textbook for universities* //M., Higher School. – 2002.
4. V.V. Volgin, R.I. Karimov, A.S. Koretsky, // Thermal power engineering. 1970. No. **3.** pp. 25-30.
5. I. M. Sharavin, N. I. Smirnov, A. I. Repin, Industrial automated control systems and controllers. – 2012. – No. **4.** – pp. 27-32.
6. V.Ya. Rotach, Theory of automatic control: textbook. for universities. M.: Publishing House of the Moscow Institute of Economics, 2004.
7. G. P. Pletnev, Automatic control and protection of thermal power plants of power plants, – M.: Energoatomizdat – 1986.
8. I. M.Sharovin, *Development of mathematical and software for neural network algorithms of adaptive ASR* : dis. – National Research University of the Moscow Institute of Economics, 2013.
9. V. R. Sabanin, N. I. Smirnov, A. I. Repin, //Exponenta Pro: Mathematics in applications. - 2004. – No. **3-4.** – pp. 78-85.
10. A.I. Repin, N.I. Smirnov, V.R. Sabanin, A.N. Maksimov, A.N. Ignatenko, *Mathematical and software algorithms for identification and adaptation of ASR and their implementation in modern PTCs* (2020)