

RESEARCH AND MODELING OF THE PROCESS OF SULFUR PRODUCTION IN THE CLAUS REACTOR USING THE METHOD OF ARTIFICIAL NEURAL NETWORKS

ZH.YE. SHANGITOVA¹, B.B. ORAZBAYEV², L.T. KURMANGAZIYEVA³, T.T. OSPANOVA⁴,
R.U. TULEUOVA⁵

^{1,4}Faculty of Information Technology, Department of Information Systems,
L.N. Gumilyov Eurasian National University, Nur-Sultan, Kazakhstan

²Faculty of Information Technology, Department of System Analysis and Management,
L.N. Gumilyov Eurasian National University, Nur-Sultan, Kazakhstan

³Faculty of Physics, Mathematics and Information Technology, Department of Software Engineering,
Kh. Dosmukhamedov Atyrau University, Atyrau, Kazakhstan

⁵Faculty of Physics, Mathematics and Information Technology, Department of Mathematics and Methods
of Teaching Mathematics, Kh. Dosmukhamedov Atyrau University, Atyrau, Kazakhstan

E-mail: ¹zhanna.shangitova@mail.ru, ²batyr_o@mail.ru

ABSTRACT

The Claus sulfur recovery process is the most important in natural gas desulfurization technology. Taking into account the large-tonnage facilities, it is urgent to solve the problem of effective management, which will allow obtaining a significant economic effect. When studying catalytic reactions and processes, mathematical modeling methods are most often used, which allow describing changes in the states of the system under study. An alternative approach to modeling chemical-technological processes can be the use of artificial neural networks, which make it possible to take into account the features of the processes under study as much as possible. The article is devoted to the study of the chemical-technological process of sulfur production in the Claus reactor by the method of artificial neural networks (ANN). This article describes the relevance of neural networks using in chemical-technological systems. Similar works on the research topic are presented. The analysis of the sulfur production unit as an object of management has been carried out. The main parameters influencing the process of sulfur production have been identified and investigated. The results of the study of sulfur production process using fuzzy logic are presented. Backpropagation algorithm is described. Based on the input data of a mathematical model with a multiple regression structure and real data from the Claus reactor, the Backpropagation Algorithm of multilayer neural networks in Python has been implemented. Based on the results of the program, the output values and errors dependences graphs have been built. As a result of the study, sufficient convergence of the results of modeling using mathematical models is shown on ANN and real production data.

Keywords: *Backpropagation Algorithm, Multi-layer Neural Networks, Python, Sulfur, Artificial Neural Networks*

1. INTRODUCTION

An increase in the depth of oil refining, an increase in the range of products of oil and gas processing is currently a priority area of oil and gas processing, and an increase in the processing of hydrocarbons with a high content of sulfur compounds leads to stricter environmental requirements for refineries. The Claus sulfur recovery unit is the most widespread process for the production of commercial sulfur from hydrogen sulfide.

During the study catalytic reactions and processes, methods of mathematical modeling are often used [1], which allow describing changes in the states of the system. However, for some processes, the construction of complete mathematical models that take into account all the features of the chemical-technological process is a very laborious task. In this case, when creating models, a number of simplifications and restrictions are included, which in turn add errors in numerical calculations and can significantly distort the results obtained.

An alternative approach to modeling chemical-technological processes can be the use of ANN,

which maximally take into account the features of the studied processes [2].

The ANN model is a network of elements (artificial neurons), interconnected and accumulating experimental knowledge, which are used for further processing [3]. To accumulate knowledge on a specific model, connections between neurons are used. The network processes the input information, and in the process of changing its state in time forms a set of output signals [4].

Искусственные нейронные сети по организации обучения можно классифицировать на типы:

Supervised neural networks, когда известно выходное пространство решений нейронной сети, а также предполагается, что имеются входные сигналы и эталонные реакции на них. В процессе обучения происходит целенаправленная модификация синаптических связей нейронной сети для достижения наилучшего соответствия между реальными выходными значениями сети и их эталонными значениями.

Nonsupervised neural networks. В этом случае нейронная сеть формирует выходное пространство решений только на основе входных воздействий.

A popular algorithm for supervised neural networks learning is the backpropagation algorithm [5], which belongs to supervised learning methods. The backpropagation algorithm is one of the best-known machine learning algorithms. The method of this algorithm is based on the use of the output error of the neural network to calculate the values of the correction for the weights of neurons in its hidden layers.

The learning algorithm was presented and detailed in 1974 by A.I. Galushkin, also independently and simultaneously by Paul J. Verbos. This algorithm considers the propagation of error signals from the network outputs to its inputs, in the direction opposite to the forward propagation of signals in normal operation. The advantages of the backpropagation algorithm include ease of implementation and resilience to anomalies and data outliers.

The goal of training a neural network when using the backpropagation algorithm is to adjust the weights of the neural network, which will allow, when a certain set of inputs is applied, to obtain the required set of outputs of neurons [6].

The advantage of ANN in modeling chemical-technological processes has been determined by the

ability to build flexible models of processes, which are described by complex mathematical dependences.

It should also be noted that neural networks are the most effective tool for solving forecasting and control problems under conditions when there is a connection between predictor variables (inputs) and predicted variables (outputs), even if this connection is of a very complex nature and is difficult to express in the usual terms of correlations or differences between groups. Therefore, the creation of algorithms for solving the optimal control problem based on the model of an ANN is an urgent scientific and technical problem. It seems appropriate to consider the possibility of using the algorithm for the object of study of this work, for the Claus reactor.

2. FORMULATION OF THE PROBLEM

The subject of this research is the algorithm of the neural network control system for the sulfur production unit and its software implementation.

The purpose of this study is to create and train multilayer neural networks in order to compare the results obtained with the results of research using fuzzy logic methods. This research will be implemented using the development of a Python program [7].

Based on the methods of system analysis using available information of various nature (theoretical information, statistical data, expert, fuzzy information), a mathematical model has been developed in the form of a fuzzy multiple regression equation, which describes the dependence of the volume of sulfur output from the Claus reactor on the input and operating parameters of the process [8].

Thus, the task of this article is to build a Python program for training of ANN, graphs and to compare the results obtained with the results of fuzzy logic.

3. RELATED WORK

In [9] the author considers an approach to modeling composite materials using neural network technologies and analyzes various methods of training neural networks.

In the article the authors examine an approach to modeling composite materials using neural network technologies and analyze various methods of training neural networks. Wavelet transform is proposed as an approach to solving of

finding the global attractor of energy function. The aim of the article is to compare the gradient methods for training neural networks.

In the work [10] algorithms for backpropagation of the error and types of neural networks have been considered. Two approaches are described for training the backpropagation algorithm: forward and backward. The work of algorithm is considered in detail.

In the work [11] provides an overview of the history of development and the current state of training methods for deep neural networks. A model of an ANN, algorithms for training neural networks, including an error backpropagation algorithm used for training deep neural networks, have been considered. The article provides an overview of the history of development and the current state of methods for teaching deep neural networks. A model of an artificial neural network, algorithms for learning neural networks, including an error backpropagation algorithm are considered. The development of neural network architectures is described: neocognitron, autoencoders, convolutional neural networks, limited Boltzmann machine, deep trust networks, long-short-term memory networks, controlled recurrent neural networks, and residual learning networks. The article discusses methods for training deep neural networks. An overview of popular deep learning neural network libraries is provided.

In the work [12] genetic algorithms are used to solve the problem. The results of the study showed that gradient methods had fast convergence only at the beginning of training, and the genetic algorithm - at the end. A hybrid algorithm based on the sequential use of gradient methods and a genetic algorithm has been proposed.

To solve optimization problems with a multi-extreme criterion, random search methods are used, which include genetic algorithms. To carry out a comparative characteristic of gradient methods and a genetic algorithm, software with a web interface has been developed. The problem of approximating the two-dimensional Rosenbrock function was used as a task for training the neural network. The results of the study showed that the gradient methods have a fast convergence only at the beginning of training, and the genetic algorithm - at the end. Thus, a hybrid algorithm based on the sequential use of gradient methods and a genetic algorithm is proposed.

In [13] an algorithm for finding the optimal temperature regime of a chemical process has been presented based on replacing the

mathematical model of the process in the form of differential equations model based on an ANN.

To solve the optimal control problem, a model of an ANN has been built, which allows solving the direct problem of chemical kinetics. Further, on the basis of the developed algorithm, the problem of theoretical optimization has been solved to find the optimal temperature regime of the process, taking into account the maximum product yield.

In the work [14] the formulation of the problem of optimal control of the catalytic reforming process has been completed. A hybrid mathematical model of the catalytic reforming process has been developed. An algorithm for determining the settings of the temperature controller at the output of the furnace in different modes of the catalytic reforming process has been synthesized. The method of optimal control of the catalytic reforming process according to the Bellman-Zade scheme has been developed. On the basis of the method of optimal control of the catalytic reforming process, a set of Pareto-optimal solutions has been built. The structure of the optimal control system for the catalytic reforming process has been developed on the basis of the developed models and algorithms. The assessment of improving the efficiency of the management of the catalytic reforming process has been carried out.

In the work [15] the proposed method is applicable for problems when the parameter of the training time of the neural network is a significant factor, and guarantees that the error function after the "splitting" of neurons will not increase.

However, in this case, the total number of neurons in a network built using a genetic algorithm for a given training sample may be larger than that of a network built using known algorithms. A genetic algorithm for synthesizing for architecture of a multi-layer feedforward network, determined by the number of layers and the number of neurons in the layer, is considered. The stages of the learning process using a genetic algorithm are described.

In the work [16] classification of neural networks by the types has been investigated. Based on the developed classification, it can be concluded that the basis for designing or choosing a neural network model for the task at hand is the structure of connections between the layers of neurons, the type of neurons, the rules for determining the weight coefficients when training the neural model.

4. DATA PREPARATION

Using the algorithm requires the data to be normalized. This means that the data must be processed in order to fit it to a single numerical scale.

The data has been normalized by dividing the numbers by the greatest, and the results have been all converted in the range from 0 to 1.

5. ENVIRONMENT

To implement the Python code, the Anaconda 3 software environment and the following auxiliary tools have been installed:

- Matplotlib – the main library for building scientific graphs in Python: <https://matplotlib.org/>
- NumPy – one of the main packages for scientific computing in Python. It contains functionality for working with multidimensional arrays, high-level math functions: <https://numpy.org/>
- Pandas – Python library for data processing and analysis: <https://pandas.pydata.org/>
- Scikit-learn – the most famous Python library for machine learning: <https://scikit-learn.org/>
- SciPy – a set of functions for scientific computing in Python: <https://www.scipy.org/>
- TensorFlow – open software library for machine learning for solving problems of building and training a neural network: <https://www.tensorflow.org/>
- Keras – open-source neural network library in Python, is an add-on over TensorFlow: <https://keras.io>

6. METHODS

During the study of the influence of input, operating parameters on the volume of sulfur produced at the output of the Claus reactor, ANN are used in this work. For ANN training a backpropagation algorithm and an iterative gradient learning algorithm are used. An iterative gradient learning algorithm minimizes the standard deviation of the current outputs from the required outputs of multilayer neural networks with serial connections [17].

The backpropagation algorithm is a popular algorithm for training multilayer

perceptrons. The method was first described in 1974 by A.I. Galushkin, as well as independently and simultaneously by Paul J. Verboos. It is an iterative gradient algorithm that is used to minimize the error in the multilayer perceptron and obtain the desired output.

Fig. 1 shows a neuron used as the main building block in backpropagation networks. A plurality of inputs are supplied, coming either from the outside or from the previous layer.

Each of them is multiplied by weight, and the products are added together:

$$NET = \sigma_1 w_1 + \sigma_2 w_2 + \dots + \sigma_n w_n \quad (1)$$

This amount must be calculated for each neuron in the network. After the NET value has been calculated, it is modified by the activation function and an OUT signal is obtained.

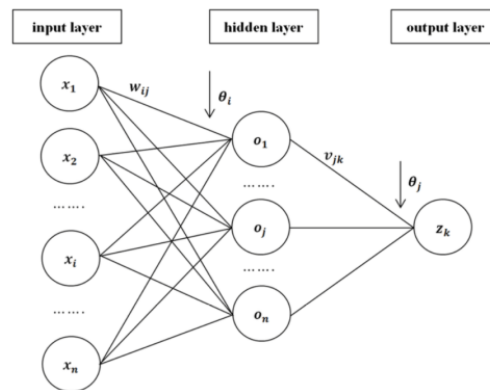


Figure 1: Structure of backpropagation networks

For backpropagation algorithms, the function is usually used

$$OUT = \frac{1}{1 + e^{-NET}} \quad (2)$$

Equation (2) describes the sigmoidal activation function. It decreases the range so that the OUT value is between zero and one. For the backpropagation algorithm, this function is most suitable, since in this case the function must be differentiable everywhere.

Consider a hierarchical network structure in which interconnected neurons are combined into several layers (Fig. 2).

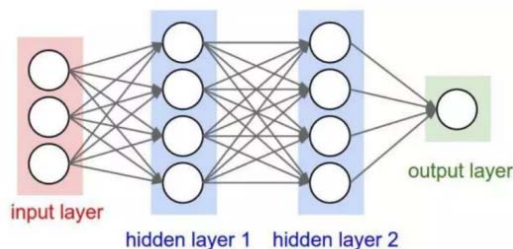


Figure 2: Hierarchical network structure

The perceptron is a network of several layers of neurons connected in series.

The goal of training a network is to adjust its weights in such a way that the application of some set of inputs leads to the required set of outputs.

Two approaches are proposed for all network layers to train the network using the back propagation algorithm: forward and backward. With a direct pass, the input vector is fed to the input layer of the neural network, after which it is divided along the network from layer to layer. The process produces a set of output signals that are the actual network response to the available input image.

During the forward pass, all synaptic weights of the network are fixed, and during the reverse pass, all synaptic weights are adjusted according to the error correction rule. The actual network output is derived from the desired output, resulting in an error signal. This signal then propagates through the network in the opposite direction to the direction of synaptic connections.

The object of the study is the sulfur production unit of Atyrau Oil Refinery LLP (Atyrau Oil Refinery) [18], designed to obtain sulfur from waste gases and acidic effluents from process plants by adsorbing sulfur with an amine solution and further catalytic conversion to crystalline sulfur.

In [19] a study has been carried out on the basis of fuzzy logic, the main technological parameters of the sulfur production process have been determined, and their influence on the quality of the Claus reactor has been studied. It has been revealed that the main input parameters of this technological process are: the volume of feedstock at the input of the Claus reactor, the temperature at the input to the Claus reactor, and the temperature in the Claus reactor.

In [20] the issues of the development of mathematical models of the thermoreactor and the Claus reactor of the sulfur production unit in the context of problems of deficiency and fuzziness of initial information were investigated. Based on the

collected and processed experimental-statistical and expert data and using the idea of the method of sequential inclusion of regressors, the structures of models for determining the sulfur output from reactor in the form of a system of multiple regression equations are identified. The models for assessing the quality of sulfur are identified in the form of linguistic models. The parametric identification of the regression coefficients is implemented by the known methods of parametric identification, based on the least squares methods using the REGRESS software package, and for the identification of fuzzy coefficients the methods of the theory of fuzzy sets are used, based on sets of level α . The Table 1 shows quantitative values of the input and output parameters of the reactors of the sulfur production unit [21].

Based on the above results, we will train an ANN in Python [22]. Figure 1 below shows a diagram of the learning algorithm for neural networks.

Stages of development of ANN

There are 5 main stages in the process of developing an ANN for solving the problem [23]:

1. Select the next training pair from the training set; submit the input vector to the input of the network;
2. Calculate the network output;
3. Calculate the difference between the output and required output
4. Correct the network weights to minimize the error;
5. Repeat the steps 1 – 4 for each vector of the training set until the error on the entire set reaches an acceptable level.

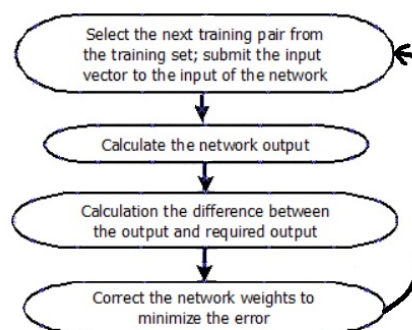


Figure 3: Diagram of the learning algorithm for neural networks

Table 1: A Table Of Quantitative Values Of The Input And Output Parameters Of The Reactor Of The Sulfur Production Unit.

№	Input, mode parameters											Output parameters						
	x1 – loading of raw materials into the F-001 thermoreactor	x2 – thermoreactor temperature F-001	x3 – thermoreactor outlet temperature	x4 – combustion air consumption on F-001	x5 – loading of raw materials into the Claus reactor R-001	x6 – temperature at the reactor inlet R-001	x7 – reactor outlet temperature R-001	x8 – boiler pressure 33-E-001	x9 – boiler pressure 33-E-002	x10 – outlet temperature 33-E-002	y1 – number of products - sulfur	y2 – sulfur quality - mass. proportion of sulfur, %	y3 – sulfur quality - mass. proportion of water, %	y4 – gray color	y5 – treated off gas	y6 – Processed	y7 – steamed water	
$\alpha = 0.5$																		
1	26.5	1350	250	650	7.3	245	295	5	4	195	26.3	99.9	0.23	0.93	0.5	0.3	0.4	
2	25	1350	270	650	7.3	245	295	5	4	195	24.7	99.96	0.24	0.93	0.5	0.3	0.4	
3	26.5	1400	250	650	7.3	245	295	5	4	195	26.3	99.9	0.17	0.95	0.5	0.3	0.4	

7. RESULTS

Using the data obtained as a result of research by the method of fuzzy logic, a program for the backpropagation algorithm in Python has been implemented. One of the most common forms of preprocessing neural network inputs is data normalization. The data has been normalized by dividing the column elements by the maximum data value [24].

Below is the code for a Python program. In this snippet we define the libraries:

```

import numpy as np
from numpy import *
import matplotlib.pyplot as plt
    
```

Initialize the input data:

```

layer_0 = np.array([
    [0.9464],
    [0.9554],
    [0.714286],
    [0.928571]])
    
```

Make transposition of the matrix of input data. It helps provide following operations with matrices:

```

layer_t_0 = layer_0.transpose()
    
```

```

i1 = 0.9464
i2 = 0.9554
i3 = 0.714286
i4 = 0.928571
    
```

Set the matrices of layer weights. Select the weights in a random way:

First layer weight matrix:

```

weights_0_1 = np.array([
    [0.2, 0.2, 0.3, 0.4],
    [0.5, 0.2, 0.3, 0.2],
    [0.3, 0.8, 0.3, 0.3],
    [0.2, 0.2, 0.4, 0.2],
    [0.5, 0.3, 0.2, 0.3]])
    
```

Second layer weight matrix:

```
weights_1_2 = np.array([
    [0.3, 0.3, 0.2, 0.2, 0.5],
    [0.2, 0.3, 0.6, 0.3, 0.4],
    [0.4, 0.4, 0.2, 0.3, 0.3],
    [0.4, 0.6, 0.6, 0.7, 0.7],
    [0.2, 0.2, 0.3, 0.3, 0.5]])
```

Last layer weight matrix:

```
weights_2_3 = np.array([0.2, 0.2, 0.2, 0.3, 0.3])
```

Select Learning rate value:

```
alpha = 0.2
```

The learning rate coefficient is a parameter of gradient neural network learning algorithms that allows you to control the amount of weight correction at each iteration. It is selected in the range from 0 to 1. It is pointless to indicate zero, since in this case the weights will not be adjusted at all. The choice of parameter is inconsistent. Larger values (0.7 - 1) will correspond to a larger value of the correction step. In this case, the algorithm will work faster (i.e., to find the minimum of the error function, fewer iterations are required). However, the accuracy of tuning the model to the minimum of the error function may decrease, which potentially increases the training error. Small values of the coefficient (0.1 - 0.3) correspond to a smaller step of the balance correction. In this case, the number of learning steps (or epochs) required to find the extremum, as a rule, increases, but the accuracy of the algorithm tuning to the minimum of the error function also increases, which potentially reduces the learning error. In practice, the learning rate factor is usually selected experimentally.

Target value:

```
goal = np.array([0.821875])
```

Set a general cycle for epochs:

```
x = 0
for x in range(0, 200):
```

The next snippet of code is the execution of Step 2 of the algorithm:

Perform summation of inputs with weights:

```
layer_1 = weights_0_1.dot(layer_0)
```

Calculate the sigmoid:

```
i = 0
```

for element in layer_1:

```
layer_1[i] = 1 / (1 + np.exp(-element))
```

```
i += 1
```

```
sigmoid_layer_1 = layer_1
```

```
layer_t_1 = sigmoid_layer_1.transpose()
```

```
deriv_sigmoid_layer_1 = sigmoid_layer_1 *
(1 - sigmoid_layer_1)
```

Likewise with the second layer:

```
layer_2 = weights_1_2.dot(sigmoid_layer_1)
```

```
j = 0
```

for element in layer_2:

```
layer_2[j] = 1 / (1 + np.exp(-element))
```

```
j += 1
```

```
sigmoid_layer_2 = layer_2
```

```
layer_t_2 = sigmoid_layer_2.transpose()
```

```
deriv_sigmoid_layer_2 = sigmoid_layer_2 *
(1 - sigmoid_layer_2)
```

Likewise with the output layer:

```
sum_output = weights_2_3.dot(sigmoid_layer_2)
```

```
sigmoid_layer_3 = 1 / (1 + np.exp(-sum_output))
```

```
output = sigmoid_layer_3
```

```
print("network output: ", output)
```

```
deriv_sigmoid_layer_3 = sigmoid_layer_3 *
(1 - sigmoid_layer_3)
```

The following piece of code is the execution of Step 3 of the algorithm:

Calculate the difference between the network output and the required output:

```
error = (output - goal) ** 2
```

```
print("network error: ", error)
```

The code written below implements the error finding algorithm (Step 4):

Output layer weights correction:

```
error_layer_2 = error * weights_2_3
```

```
error_layer_t_2 = error_layer_2.transpose()
```

Correction of the weights of the second layer:

```
error_layer_1 = error_layer_2.dot(weights_1_2)
```

```
error_layer_t_1 = error_layer_1.transpose()
```

Correction of the weights of the first layer:

```
w1 = weights_0_1[0] + error_layer_t_1[0] *
deriv_sigmoid_layer_1[0] * layer_t_0 * alpha
```

```
w2 = weights_0_1[1] + error_layer_t_1[1] *
deriv_sigmoid_layer_1[1] * layer_t_0 * alpha
```

```
w3 = weights_0_1[2] + error_layer_t_1[2] *
deriv_sigmoid_layer_1[2] * layer_t_0 * alpha
```

```
w4 = weights_0_1[3] + error_layer_t_1[3] *
deriv_sigmoid_layer_1[3] * layer_t_0 * alpha
```

```
w5 = weights_0_1[4] + error_layer_t_1[4] *
deriv_sigmoid_layer_1[4] * layer_t_0 * alpha
```

```
weights_0_1 = vstack([w1, w2, w3, w4, w5])
```

Correction of the weights of the second layer:

```
m1 = weights_1_2[0] + error_layer_t_2[0] *
deriv_sigmoid_layer_2[0] * layer_t_1 * alpha
```

```
m2 = weights_1_2[1] + error_layer_t_2[1] *
deriv_sigmoid_layer_2[1] * layer_t_1 * alpha
```

```
m3 = weights_1_2[2] + error_layer_t_2[2] *
deriv_sigmoid_layer_2[2] * layer_t_1 * alpha
```

```
m4 = weights_1_2[3] + error_layer_t_2[3] *
deriv_sigmoid_layer_2[3] * layer_t_1 * alpha
```

```
m5 = weights_1_2[4] + error_layer_t_2[4] *
deriv_sigmoid_layer_2[4] * layer_t_1 * alpha
```

```
weights_1_2 = vstack([m1, m2, m3, m4, m5])
```

Output layer weights correction:

```
weights_2_3 = weights_2_3 + error *
deriv_sigmoid_layer_3 * layer_t_2 * alpha
```

```
#print("Weights of the third layer:",
weights_2_3)
```

The results of the program are shown in Fig. 4,5. Build graphs [25] of dependencies of output values and errors on the number of epochs:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import *
import pandas as pd
fig = plt.figure()
Set the interval of values along the X axis (The
number of epochs is equal to 500):
x = np.arange(1, 501)
y = np.array([...])
ax = plt.gca()
plt.plot(x, y)
Indicate the names of the axes of the graph:
ax.set_xlabel("Epochs")
ax.set_ylabel("Output values")
plt.savefig('1.png', transparent=True)
```

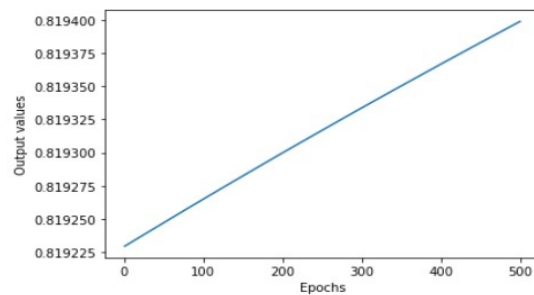


Figure 4: Graph of dependencies between output values and number of epochs

```
fig = plt.figure()
Set the interval of values along the X axis
(The number of epochs is equal to 200):
x = np.arange(1, 201)
y = np.array([...])
ax = plt.gca()
plt.plot(x, y)
ax.set_xlabel("Epochs")
ax.set_ylabel("Errors")
plt.savefig('2.png', transparent=True)
```

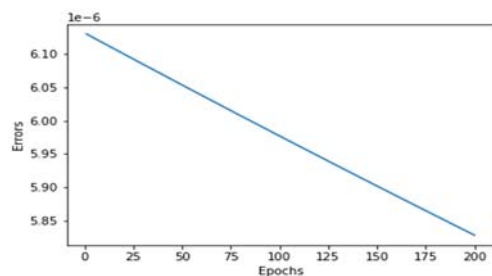



Figure 5: Graph of dependencies between errors and number of epochs

8. DISCUSSION

During the research, the backpropagation method has been used. After analyzing the graphs, it can be concluded that the results of the study using ann and fuzzy logic method converge. During the research, it was found that the learning error is minimal and the output value is as close as possible to its target value.

In a previous paper [21] the results of a fuzzy logic study were presented, in which the values of sulfur yield in the Claus reactor were obtained.

In this work, we have carried out a study of this process by the method of ann. Thanks to the results of both studies, we were convinced that both methods give similar results, thus we can say that the sulfur output is valid.

9. DIFFERENCE FROM PRIOR WORK

In the works [19, 20, 21] the technological process of the Claus reactor on the basis of fuzzy logic was investigated. The results are shown in Table 1.

In this work, in contrast to the work [21], a neural network approach is used to study the dependence of the input and output parameters of the technological process. An algorithm for backpropagation of an error is considered for training an artificial neural network based on the input parameters of the Claus reactor.

The results of applying the neural network approach have shown its effectiveness in the study of the technological process.

The novelty of the current work lies in the fact that the convergence of the results of methods of fuzzy logic and artificial neural network has been proven.

The investigated technological object is complex and it has a number of features and

limitations. The use of a neural network approach in modeling chemical-technological processes makes it possible to maximally take into account the features of the process under study, in connection with which the use of the method of artificial neural networks is important.

10. LIMITATIONS

Any research has limitations; therefore, this study has no exception. The study requires a survey with the decision maker of the plant, obtaining the necessary information related to the technological process at the plant. But due to the current situation in the world, at the moment there is no way to get access to the plant and fully work with specialists. In the future, when access to the plant is obtained, it is planned to work with specialists at the plant.

11. CONCLUSION

In this article, based on the use of the results of previous studies, a method of ANN has been proposed. The backpropagation algorithm has been chosen as the method. For this, a scheme for implementing the algorithm has been built. Based on the proposed algorithm, an ANN training program in Python has been developed.

Data normalization has been carried out, after which the network has been trained. Based on the training results, the graph of dependences between output values and number of epochs and the graph of dependences between errors and number of epochs have been built. Based on the results of the research, it can be concluded that the use of the backpropagation method is quite effective for solving the problems of controlling chemical-technological processes.

As a result of the study, sufficient convergence of the results of modeling using mathematical models, ANN and real production data have been shown. In the future, it is planned to use the results of this study for further research on other methods of ANN.

REFERENCES:

- [1] Hashemi, M., Pourfayaz, F., & Mehrpooya, M. (2019). Energy, Exergy, Exergoeconomic and Sensitivity Analyses of Modified Claus Process in a Gas Refinery Sulfur Recovery Unit. *Journal of Cleaner Production*, 220(5),

- 1071.
- [2] Tussupov, J., La, L., & Mukhanova, A. A. (2014). A Model of Fuzzy Synthetic Evaluation Method Realized by a Neural Network. *International Journal of Mathematical Models and Methods in Applied Sciences*, 8, 103–106.
- [3] Xialong, H. (2020). Global Finite-time Stability for Fractional-order Neural Networks. *Optical Memory and Neural Networks*, 77–99. <https://doi.org/https://doi.org/10.3103/S1060992X20020046>
- [4] Yakhyaeva, G. . (2006). *Fuzzy Sets and Neural Networks*. Internet-University of Information Technologies, BINOM, Laboratory of Knowledge.
- [5] Tijanić, K., Car-Pušić, D., & Šperac, M. (2020). Cost Estimation in Road Construction Using Artificial Neural Network. *Neural Computing and Application*, 32, 9343–9355.
- [6] Haghnegahdar, L., & Wang, Y. (2020). A Whale Optimization Algorithm-Trained Artificial Neural Network for Smart Frid Cyber Intrusion Detection. *Neural Computing and Application*, 32, 9427–9441. <https://doi.org/https://doi.org/10.1007/s00521-019-04453-w>
- [7] Saha, A. (n.d.). *Doing Math with Python*. San Francisco.
- [8] Kruglov, V. V., & V.V., B. (2002). *Artificial Neural Networks: Theory and Practice*. Moscow.
- [9] Fedosin, S. A., Ladyaev, D. A., & Maryina, O. A. (2010). Analysis and Comparison of training methods for neural networks. *Bulletin of Mordovian University*, 4, 79–88.
- [10] Gorelova, A. V., & Lyubimova, T. V. (n.d.). Error Backpropagation Algorithm. *North Caucasian Branch of the Belgorod State Technological University Named after V.G. Shukhov*, 151–156.
- [11] Sozykin, A. V. (2017). Overview of Methods for Training Deep Neural Networks. *Bulletin of South Ural State University*, 6(3), 28–59.
- [12] Belyavskiy, G. I., & Lila, V. B. (2012). Algorithm and Software Implementation of a Hybrid Training Method for Artificial Neural Network. *Software Products and Systems*, 4, 96–100.
- [13] Shaymukhametov, D. R., Mustafina, S. A., & Shaymukhametova, D. V. (2019). Creation of an Artificial Neural Network to Determine the Optimal Temperature Regime for Conducting a Chemical Process. *Bulletin of Bashkirian University*, 24(3), 551–556.
- [14] Dzhambekov, A. M., & Scherbatov, I. A. (2017). Optimal Control of the Catalytic Reforming of Gasoline Fractions. *Bulletin of the Tambov State Technical University*, 23(4), 557–571. <https://doi.org/https://doi.org/10.17277/vestnik.2017.04.pp.557-571>
- [15] Bozhich, V. I., Lebedev, O. B., & Shnicer, Y. L. (n.d.). Development of a Genetic Algorithm for Learning Neural Networks. *Proceedings of International Conference ICAD*, 170–174. Taganrog: Proceedings of the Taganrog State Radio Engineering University.
- [16] Gorbachevskaya, E. N. (2012). Neural Networks Classification. *Bulletin of the V. N. Tatishchev Volga State University*, 2(19), 128–134.
- [17] Protalinsky, O. M., Scherbatov, I. A., & Belyaev, I. O. (2010). Hybrid Neural Network Training Method for Classifying the Catalytic Stage of Claus Process. *Saratov State Technical University*, 2.
- [18] Serikov, T. P., & Orazbayev, B. B. (2018). *New Installations of the Atyrau Refinery: Sulfur Recovery Unit*. Evero.
- [19] Orazbayev, B. B., Shangitova, Z. Y., Orazbayeva, K. N., B.A., S., & Shagayeva, A. B. (2020). Studying the Dependence of the Performance Efficiency of a Claus Reactor on Technological Factors with the Quality Evaluation of Sulfur on the Basis of Fuzzy Information. *Theoretical Foundations of Chemical Engineering*, 54(6), 726–732.
- [20] Orazbayev, B. B., Shangitova, Z. Y., Kurmangazieva, L. T., & Kodanova, S. K. (2020). The Study of the Main Parameters Affecting the Efficiency of Claus Reactor and Construction of Linguistic Models for Assessing the Quality of Produced Sulfur. *Journal of Sulfur Chemistry*, 21(5), 498–507. <https://doi.org/https://doi.org/10.1080/17415993.2020.1759603>
- [21] Orazbayev, B. B., Shangitova, Z. Y., Orazbayeva, K. N., Kassenova, L. G., Zhanbirova, G. A., & Istayeva, N. (2019). Development of Mathematical Models of Thermoreactor and Claus Reactor of Sulfur Production Based on Hybrid Method. *Bulletin of L.N. Gumilyov Eurasian National University*, 127(2), 87–94.
- [22] Joshi, P. (2019). *Artificial Intelligence with Python*. Dialektika.

- [23] Wasserman, P. D. (1989). *Neural Computing: Theory and Practice Hardcover*. Coriolis Group.
- [24] Rojas, R. (1996). *Neural Networks*. Springer-Verlag.
- [25] Perrotta, P. (2020). *Programming Machine Learning. From Coding to Deep Learning*. North Carolina: The Pragmatic Bookshelf.