Development of neural network for control production process in oil and gas fields

M I Sharipov
Ufa State Petroleum Technological University, Branch of the University in the City of Sterlitamak 453118, Russian
sharipovm@mail.ru

Abstract. For a long time there has been a tendency to increase field productivity, therefore, increasing oil recovery is the main task for fuel and energy complex. Currently, neural networks are increasingly used in various industries. The advantage of neural networks is to work with a large amount of data, however, it must have sufficient data sets collected and prepared for its operation, thereby achieving high decision accuracy. When developing oil and gas fields, the main task is to ensure maximum production from an economic and physical point of view. Oil production at oil and gas fields varies in volume, complexity, operating conditions, etc., therefore, it is necessary to find the optimal production conditions for each field. At the moment, the main problems in oil production at oil and gas fields are: the long processing time of data collected from wells, the increased risks of operating these wells, as well as the low amount of oil produced. The main objective of this study is to develop a control method us in artificial intelligence to control the production process in oil and gas fields, taking into account all factors. The resulting neural network, without reconfiguring weighted connections, generates output signals when applied to the input to the network. The resulting neural network expresses the patterns that are present in the input data. This network turns out to be the functional equivalent of some model of dependencies between variables.

1. Introduction
The increase in oil recovery of wells is currently the main requirement in the development and operation of oil fields. However, this is hampered by a number of reasons, such as the small volume of oil produced, the slow transfer of information to control points and the low efficiency of the water injection process into the formation. All these problems can be solved with artificial intelligence. The neural network should reduce the number of wells and analyses carried out to determine the characteristics of the deposit, which will reduce financial costs and save time.
Artificial neural networks (ANN) find new applications every day. They are sets of elementary neuron-like information converters (neurons) connected to each other by information exchange channels for their collaboration. Network data is capable of processing a large amount of data at high speed.
In order to build a neural network, it is necessary to prepare data sets that will be used to solve the problems.
After data collection, the neural network training process takes place, where the values of weight coefficients for individual nodes are clarified on the basis of gradual increase of input and output information volume. Learning occurs until the error reaches an acceptably low level. The obtained ANN model is then tested using independent examples.

2. Description of technological process

![Figure 1. Explored area.](image)

All field production data necessary for neural network training is obtained from the oil companies exploration and production department, so this information depends on the real data obtained during the well operation. An example of the grid-map of the investigated area from which the data is taken is shown in Figure 1.

The input values are as follows:
1. Number of injection wells - 1 - 5.
2. The number of production wells is 1 - 10.
3. Injection factor (ratio of injection wells to production wells) - 0.1 - 5.
4. Crude oil fraction - 0.833667 - 1.0.
5. Quantity of injected water, m$^3$/day - 61685 - 26994661.
6. Quantity of pumped water, m$^3$/day - 521,2129 - 3896482.

The main problems of this process are the low amount of oil produced and the low efficiency of the water injection process into the formation.

These problems can be solved if a neural network is used to control the mining process. Since the rapid transfer of information to the control points increases the efficiency of use of the well stock, reduces operating costs, optimizes the injection of water into the formation, as well as increases the volume and speed of production.

Development of a Neural Network Based Control System

The development of the system in the MatlabR2015b environment began with the collection of data that are used as input data sets, including production data from known wells. The best continuous production conditions were then chosen, excluding maintenance periods, interruptions, etc.

In order to collect data, on the map of well location, a point is selected, relative to which operation of other wells is considered, then data are collected from all considered wells.

The essence of the method is that data obtained from several wells at once are processed, not from each well separately. Due to this and the ability of the neural network to process large values at one time, data processing time is significantly reduced [1-6].
3. Data collection

To create a training set for the neural network under development, we will use the data and formula obtained by experimental means:

\[ V = \frac{(V_z - V_v)}{n} k_n k_d f, \]  

where \( V \) - is the amount of oil produced; \( N \) is the injection factor (ratio of injection wells to production wells); \( V_z, V_v \) - is amount of the gone and rolled out water respectively; \( k_n, k_d \) - number of injection and production wells, respectively; \( f \) - Oil fraction in crude oil composition.

The outputs of the neural network must indicates:
1. The amount of oil produced;
2. Data processing and preparation;
3. Using formula (1), create a table with a learning dataset. It will consist of 500 examples.
4. 6 values will be supplied to the neural network input:
   1. Number of delivery wells.
   2. Quantity of production wells.
   3. Injection factor (ratio of injection wells to production wells).
   4. Crude oil fraction.
   5. Amount of the pumped water.
   6. Amount of the extorted water.
5. At the output, the neural network shall calculate:
   1. Amount of the extracted oil

| №  | Number of delivery wells | Quantity of production wells | Injection coefficient | Crude oil fraction | Amount of the gone water, m³/day | Amount of the rolled-out water, m³/day | Quantity of produced oil, m³/day |
|----|--------------------------|----------------------------|----------------------|-------------------|----------------------------------|--------------------------------------|----------------------------------|
| 1  | 5                        | 4                          | 1,25                 | 0,978875          | 9115227                          | 2118141                             | 17123181                         |
| 2  | 5                        | 1                          | 5                    | 0,998705          | 12541155                         | 2581183                             | 24867685                         |
| 3  | 2                        | 2                          | 1                    | 0,977392          | 9029387                          | 2242921                             | 2653215                          |
| 4  | 4                        | 6                          | 0,7                  | 0,955479          | 14140388                         | 823860                              | 21375754                         |
| 5  | 3                        | 4                          | 0,8                  | 0,943319          | 10538808                         | 2708967                             | 7090596                          |
| 6  | 5                        | 8                          | 0,6                  | 0,910067          | 16324000                         | 963036                              | 33550815                         |
| 7  | 5                        | 1                          | 5                    | 0,924735          | 18851974                         | 647894                              | 42084875                         |
| 8  | 2                        | 9                          | 0,2                  | 0,998692          | 11473443                         | 166511                              | 4065171                          |
| 9  | 2                        | 5                          | 0,4                  | 0,911984          | 7441365                          | 1130555                             | 2302143                          |
| 10 | 5                        | 6                          | 0,8                  | 0,9582            | 7986699                          | 1967742                             | 13841675                         |

Table 1. Training data set.
4. Selection of neural network type and architecture

The direct propagation neural network shown in Figure 2 has a straight-line structure, it transmits information from input to output. Neurons of one layer are linked to neurons of the other layer, but are not linked to each other. Network learning takes place by the reverse propagation method of the error train Fcn = 'trainbr', in which the network receives a plurality of input and output data. If the network has enough hidden neurons, it simulates the interaction between input and output data [7-16].

![Figure 2. Neural network structure.](image)

5. Building and Learning a Neural Network at Matlab

We set the maximum number of training epochs equal to 1000, which determines the interval of time after which training will be terminated, the number of epochs between screenings is equal to 5, the criterion of training completion at which training will be considered as completed we specify 0.0001 [17-24].

Next we implement and train the neural network in Matlab. To do this, enter the uiopen command. The neural network input is called input and the output is called output. Enter the number of inputs and outputs and the number of hidden layers (Figure 3).

The nnstart command will allow you to enter the Neural Network Learning tab.

![Figure 3. Entering the number of hidden neurons.](image)

![Figure 4. Neural Network Training Schematic Diagram.](image)
After the network training, we obtain a graph (Figure 4) showing the change of the standard error in relation to the epochs. The graph shows that with the increase in the number of eras for training and testing the neural network, the error rate decreases. Mean square error $1000, 0959 \times 10^{-12}$ has been reached in 5 epochs.

Figure 5 shows the Training State graphs. The final value of the gradient factor per 1000 eras is $6.6271 \times 10^{-5}$, which is very close to zero. The smaller the gradient factor, the better the training and testing of the neural network will be.

The final training parameter $\mu$ is 50 per 1000 epochs. The Mu plot shows how the regularization variables ($\mu$) of the Bayesian regularization method we selected changed. Regularization is the range of numerical values required to adjust the values of the learning sample and the retraining of the neural network.

The "valfail" graph shows a learning error that is 0 per 1000 eras. Training error is an indicator of the accuracy of model setting on the training set and can be used as a condition of training stop when the specified value is reached [25-30].

![Network Training Schedules](image)

**Figure 5.** Network Training Schedules.

![Comparison between the quantity of oil obtained by formula and neural network](image)

**Figure 6.** Comparison between the quantity of oil obtained by formula and neural network.
From the graph shown in Figure 6, it can be seen that the data obtained by the neural network is close to the data obtained by calculations, from which it follows that the neural system is trained correctly.

6. Check of neural network
In order to check the received neural network, we will input 6 values with the command sim (net, [3; 6; 0, 5; 0.966961; 18792218; 1273693]), finally obtain a value of 1, 5246 * 10^7, which is approximately equal to the value of 15245757, which is obtained by formula 1. It can be concluded from this that the neural network is trained correctly.

7. Conclusion
The method developed in this article to control the process of oil production using artificial intelligence helps to increase the volume of produced oil, reduces operating costs and allows to increase the efficiency of the process of water injection into the formation. All this will open up new opportunities for development of new deposits or more efficient use of existing infrastructures.

References
[1] Muravyova E A 2011 Autoregressive neural network for model predictive control of multivariable cracking catalyst calcinato Optical Memory & Neural Networks p 213–16
[2] Nielsen M 2015 Neural networks and deep learning (Publisher: «Determination Press») p 224
[3] Rashid T 2016 Make your own neural network (Publisher: «CreateSpace») p 222
[4] Muravyova E A and Timerbaev R R 2018 Application of artificial neural networks in the process of catalytic cracking Optical Memory & Neural Networks (Information Optics). p 203–8
[5] Muravyova E A and Uspenskaya N N 2018 Development of a neural network for a boiler unit generating water vapour control Optical Memory & Neural Networks (Information Optics. p 297–307
[6] Kriesel D 2014 A brief introduction to neural networks (Publisher: «Autoedicin») p 244
[7] Samarasinghe S 2016 Neural networks for applied sciences and engineering (Publisher: «CRC Press») p 570
[8] Evmenov V 2009 Intelligent control systems (Moscow: book house "Librokom") p 290
[9] Muravyova E and Almakaev I 2019 Electrical Heating Reactor Control System Using Neural Network and the Fuzzy Controller International Multi-Conference on Industrial Engineering and Modern № 8934383
[10] Muravyova E A and Sharinov M I 2018 Intelligent Control System for Process Parameters Based on a Neural Network 14th International Scientific-Technical Conference on Actual Problems of Electronic Instrument Engineering p 256–60
[11] Haikin S 2006 Neural networks: a complete course (Moscow: Publishing house «Williams») p 1104
[12] Callan R 2003 Basic concepts of neural networks ( M.: Publishing house «Williams») p 291
[13] Nikolenko S, Kadurin A and Arkhangelsk E 2018 Deep learning Immersion in the world of neural networks (SPb.: «Peter») p 477
[14] Muraveva E A, Kayashev A I and Gabitov R F 2010 Control of the furnace for calcining zeolite-containing catalysts for cracking petroleum products using the floating horizon method using a neural network model Automation, telemechanization and communication in the oil industry p 19
[15] Mahmud F and Tarek A A 2011 Identification and adaptive control of dynamic non-linear installations Intelligent Control and Automation 02(03) p 176–81
[16] Li H Dzh and CHen M 2008 Design of decoupling PID controller for a kind of practical engineering Control Engineering 15(3) p 275–8.
[17] Cheng Q M and Zheng Y 2007 Multi-variable PID neural network control systems and their application to coordination control *East China Electric Power* 11 p 54–8

[18] Vasilyev V I and Ilyasov B G 2000 Intelligent control systems using genetic algorithms *Appendix to the journal "Information Technology"* № 12 p 392

[19] Andreychikov A V and Andreychikova O N 2004 *Intelligent Information Systems: A Textbook* (M.: Finance and statistics) p 423

[20] Vasilyev V I and Ilyasov B G 2009 *Intelligent control systems. Theory and practice* (M.: Radio engineering) p 392

[21] Muravyova E A and Gabitov R F 2019 Economic features to optimize the catalyst calcinations process 2018 International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2018 № 8602535

[22] Muravyova E A, Sharipov M I and Gabitov R F 2019 Scada-system based on multidimensional precise logic controller for the control of a cement kiln 2018 International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2018 № 8602589

[23] Muravyova E A, Sharipov M I and Bondarev A V 2019 Method for increasing the speed and reducing the error of multidimensional precise logic controller 2018 International Multi-Conference on Industrial Engineering and Modern Technologies, FarEastCon 2018 № 8602643

[24] Heo S and Lee J H 2019 Parallel neural networks for improved nonlinear principal component analysis *Computers & Chemical Engineering* 1274 p. 1–10

[25] Huang S, Zhang J and Hu C K 2019 Effects of external stimulations on transition behaviors in neural network with time-delay *Physica A: Statistical Mechanics and its Applications* 53615 № 122517

[26] Sheng D, Wei Y, Chen Y and Wang Y 2019 Convolutional neural networks with fractional order gradient method *Neurocomputing* In press, corrected proof Available online

[27] Lin S, Huang Y and Ren S 2018 Analysis and pinning control for passivity of coupled different dimensional neural networks *Neurocomputing* 32110 p. 187–200

[28] Kobayashi M 2018 Twin-multistate commutative quaternion Hopfield neural networks *Neurocomputing* 3203 p. 150–6

[29] Berg J and Nystrom K 2018 A unified deep artificial neural network approach to partial differential equations in complex geometries *Neurocomputing* 31723 p. 28–41

[30] Jiang N, Xu J and Zhang S 2018 Neural network control of networked redundant manipulator system with weight initialization method *Neurocomputing* 30713 p. 117–129