The Artificial Neural Network for Solving the Classification Problem with Domain Constraints

Ye Thu Aung
Applied Mathematics and Artificial Intelligence
National Research University Moscow Power Engineering Institute (MPEI)
Moscow, Russia
yethuaungg55@gmail.com

Mikhaylov Ilya Sergeevich
Applied Mathematics and Artificial Intelligence
National Research University Moscow Power Engineering Institute (MPEI)
Moscow, Russia
fr82@mail.ru

Zayar Aung
Applied Mathematics and Artificial Intelligence
National Research University Moscow Power Engineering Institute (MPEI)
Moscow, Russia
zayaraung53@gmail.com

Abstract*— One of the most powerful methods for solving the forecasting problem in data mining is the use of artificial neural networks (ANN). ANNs are considered non-linear statistical data modeling tools that model complex relationships between input and output data or discover patterns. As the initial data of the problem, the values of the parameters controlled by the flow meter recorded at oil wells in the Perm Region are considered. In this paper, we propose to create different neural network architectures and perform a comparative analysis of their work on the data set under consideration. As a result, a neural network will be determined that provides the maximum accuracy of predicting the calculated parameters. Thus, the computational complexity of the multiphase flow meter operation will be reduced.

Keywords—*data mining; artificial neural network; nrural network architectures; oil wells.*

I. INTRODUCTION

The neural network is a universal model, as it is able to approximate any surface. The corresponding theorem was formulated in 1957 by Andrey Kolmogorov. Theorem (A. N. Kolmogorov, 1957) Every continuous function a (x) given on a unit cube of a d-dimensional space is representable as

$$a(x) = \sum_{i=1}^{2d+1} \sigma_i \left(\sum_{j=1}^{d} f_{ij}(x_j) \right)$$

Where $x = [x_1, \dots, x_d]^T$- the vector of the object description, the functions $\sigma_i(.)$ and $f_{ij}(.)$ are continuous, and f_{ij} does not depend on the choice of a.

According to the formulation of the theorem, the function a(x) is defined only on the unit cube, and, therefore, all the elements of the sample must lie in it. Thus, it is necessary to scale the features so that on the training sample, each feature takes values from the segment [0,1].

Normalization is a method of changing ranges of values — scaling. Scaling is particularly useful in machine learning, because different attributes can be measured in different ranges, or the values of a single attribute vary too much. For example, one attribute has a range from 0 to 1, and the second one has a range from 1 to 1000. For a regression problem, the second attribute would have a big impact on learning, although it is not a fact that it is more important than the first. Normalization involves changing the ranges in the data without changing the shape of the distribution.

To solve the feature scaling problem, you can use the following formula, which must be applied for each vector that is used in training the model:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

II. NEURAL NETWORK ARCHITECTURES FOR VARIOUS TASKS

The processing of multidimensional data, which includes the tasks of classification, creation of a new structure of the feature space and interpretation, storage, and transmission through communication channels, presents certain difficulties and is an urgent task of interest to researchers. The solution of these problems is greatly simplified if the dimension of the feature space is compressed. This compression is possible because in most cases the features are interrelated(correlated), hence the data is redundant in terms of information, and this redundancy is completely determined by the correlation matrix of the original X variables. However, such compression always leads to the loss of some information, which is a significant disadvantage of classical methods of processing multidimensional data.

To determine the neural network architectures applicable to the construction of a regression model, the possibility of using different types of networks in different classes of problems was analyzed. The Hopfield network, counter-propagation network, radial basis network, Kohonen map, multilayer perceptron, bidirectional associative memory, Hamming network, probabilistic network, adaptive resonance network were considered.

The results of the comparison are shown in Table 1 (the advantages mean the possibility of applying a neural network of this type to the solution of the corresponding problem).

TABLE I. COMPARISON OF DIFFERENT TYPES OF NEURAL NETWORKS

Type of neural network	Associative memory and recognition	Data compression	Forecasting	Optimization	Classification, expert reports	Clustering	Function approximation	Smoothing (generalization)
Hopfield Network	+			+				
Counter-distribution Network	+							
Radial Basis Network (RBF)			+		+		+	+
Kohonen Map				+	+	+		
Multi-layer Perceptron (MLP)	+		+		+		+	+
Bidirectional associative memory	+							
Hamming Network	+				+			
Probability Network PNN					+			
Adaptive Resonance Network ART2					+	+		

III. MULTI-LAYER PERCEPTRON

A multi-layer neural network consists of several layers. Each layer consists of neurons. A two-layer neural network is shown in Fig.1.

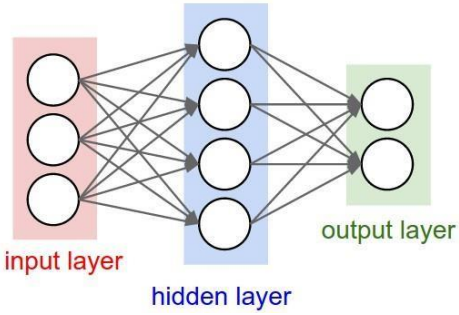

Fig 1. Two-layer neural network

The same is true for more layers. The input layer is the data $(matrix(n,m))$Layers that are not input or output are called hidden layers. When solving a regression problem, there is usually one neuron on the output layer that returns the predicted numbers (for each object by a number). In the case of a classification problem, the output layer usually has one neuron, if the problem is binary classification, and K neurons, if the problem is class classification.

A. Network parameter stabalization

By the learning rate, we can judge the correspondence of the sample and the neural network (see Fig 2.):

- If the neural network is too complex, it will quickly retrain, as, for example, the yellow line shows.
- If the sample is complex or very noisy for the neural network, the neural network will learn slowly, as shown by the blue line.
- If the sample complexity corresponds to the neural network, then most likely we will see a red line – a good learning rate.
- It is important to keep track of the difference between the values of the error function in training and in control. This difference should not be significant. If it is large, it means that the neural network has been retrained and its complexity should be changed.

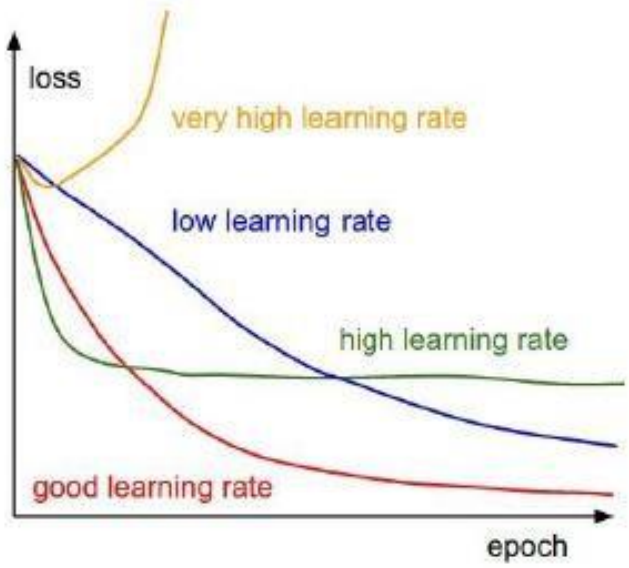

Fig.2. Graph illustrating the dependence of loss on the epoch (curves for different situations)

A significant difference between the values of the error in training and in control indicates that the neural network is overfitted (see Fig. 3.).

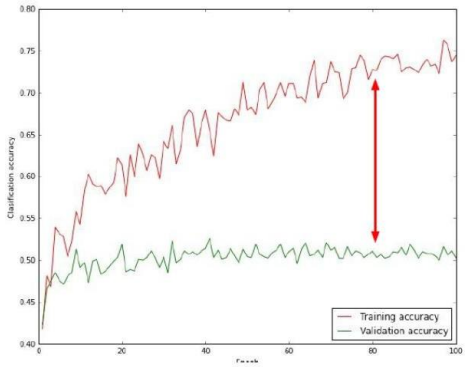

Fig.3. Retrained neural network

IV. Comparison Of Neural Networks

For all 3 types, we will minimize the functional of the MSE sum (otherwise the numbers will be very close to 0, the neural network will stop learning), then we will convert it to a standard MSE (divide by the number of elements participating in the MSE) to check the results. Graphs in Fig.4-6 show the results of 3 models: a Neuron, a Perceptron, and a Multilayer Perceptron. Judging by the graphs, we can conclude that the multi-layer perceptron finds a solution well in comparison with simpler models. The neuron and perceptron models give rather poor results, and they are also prone to falling into the local minimum of the error functional, so the model may stop learning For a more detailed analysis of the architecture, it is necessary to consider the MLP graph closer .

Dependent variable "Liquid [m^3/30 sec]" at about 100 epochs, the values of test and train loss intersect, most likely, in this case, it is worth finishing training the model. On the 100th epoch of train and test, the mse loss is approximately 0.05, which is a fairly good result.

Dependent variable "Gas [m^3/30 sec]" training can be stopped at about 10 epochs (however, this is due to the fact that 25% of the sample takes zero values, so it cannot be argued that this is the best approximation), and most likely, the model is too complex, since there are jumps in the graph. On the 100th epoch of train and test, the mse loss is approximately 0.1, which is a bad result.

Dependent variable "Water cut (OW) [%]". At about 250 epochs, the best result, the variable is predicted quite well (mse is about 0.002).

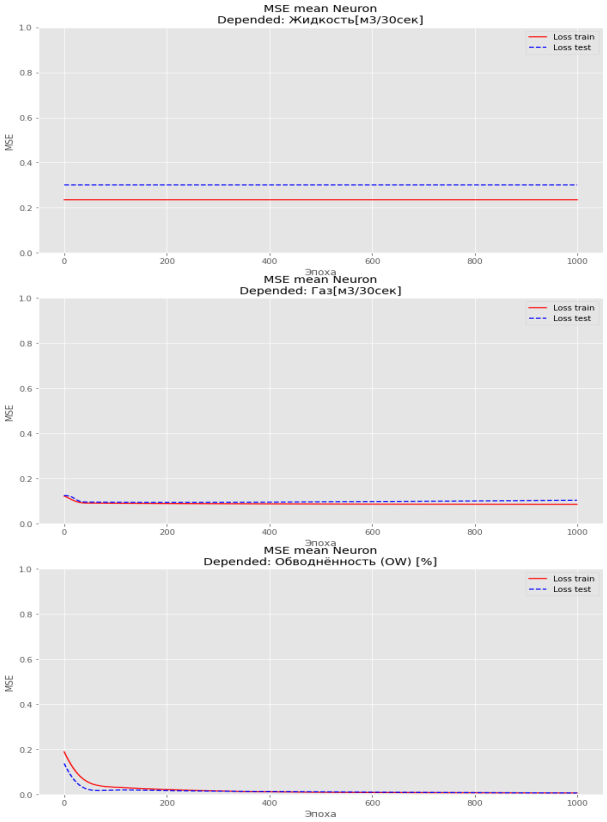

Fig.4. MSE graphs for the "Neuron" model»

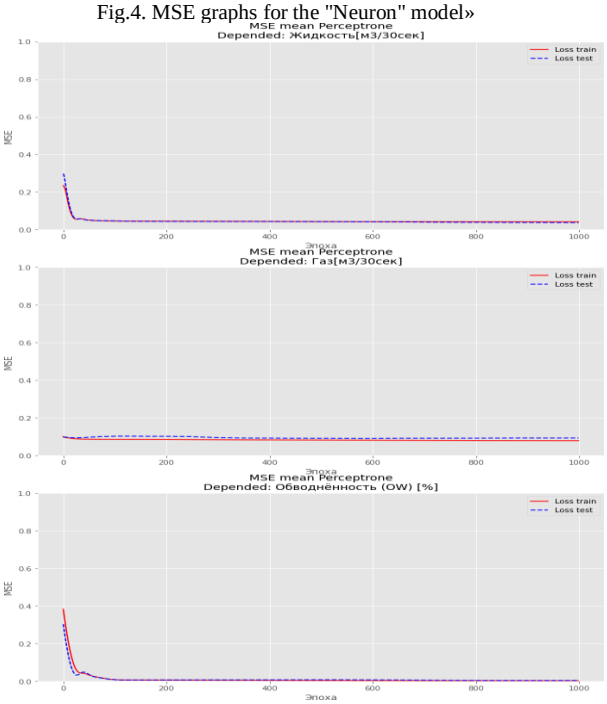

Fig.5. MSE graphs for the "Perceptron" model»

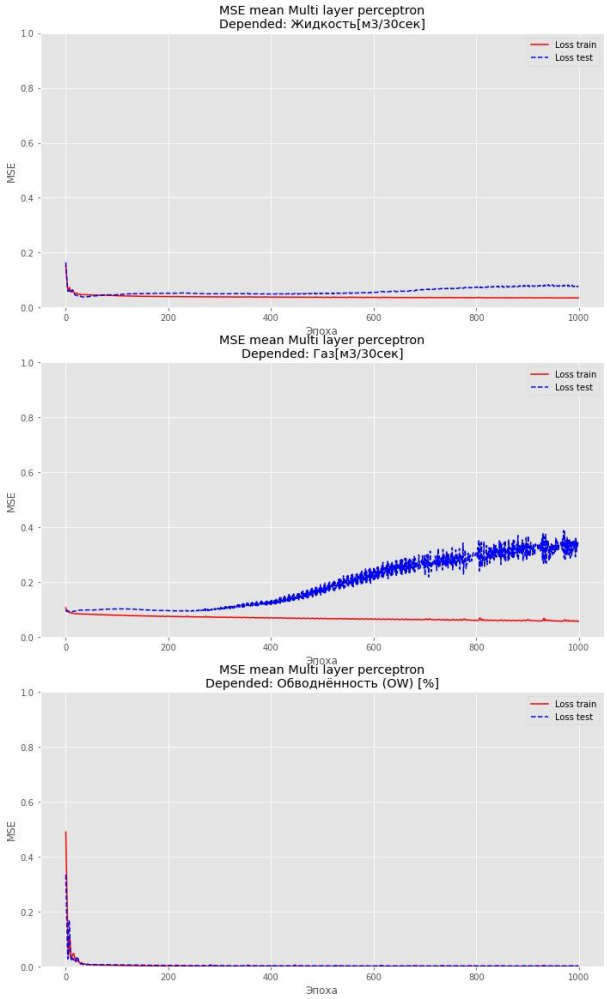

Fig.6. MSE graphs for the "Multilayer Perceptron" model»

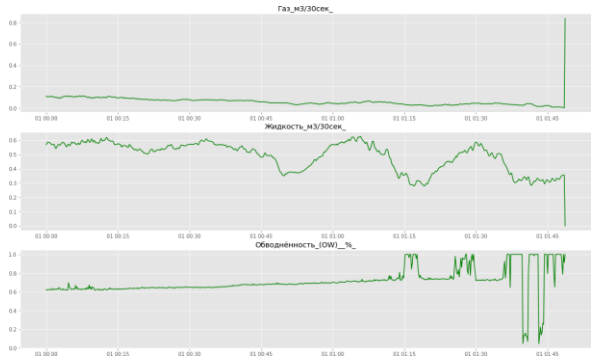

Fig.7. Comparison of values of predicted variables for a certain period of time

V. Conclusion

A. *Research results 1*

In the course of the study, the results were obtained:

* The main architectures of neural networks for solving the regression problem are investigated

* The models "Neuron", "Perceptron", "Multilayer perceptron" and "Network of radial basis functions" are applied for the regression problem.

As a result of research, it was found that the best solution to this problem is provided by a "Multi-layer perceptron". The "network of radial basis functions" gives approximately the same results for the MSE metric, however, judging by the display of results on the graph with real values, the results are worse for each dependent variable. Also, the "Multi-layer perceptron" can be trained quite quickly using the available free video card from the Google Colab service, RBFN is very difficult to parallelize.

The best results were obtained for the dependent variable "Water cut", most likely due to the fact that this variable does not have 0 values (sharp drops). It is necessary to change the approach to solving the problem, since all 3 dependent variables in this study also depend on each other. Thus, it is necessary to find not 3 models for solving each problem, but one model, for example, a multilayer perceptron with 3 outputs, in each of which a dependent variable will be predicted.

Thus, to solve this problem, additional research is needed, as a result of which a neural network should be obtained that simultaneously predicts 3 independent variables.

B. *Research results 2*

The study conducted a more thorough analysis of the data with a variety of visualizations. It was found that the distributions of the initial features are not normal (according to the Kolmogorov-Smirnov statistical method). Accordingly, the correlations of the features were checked using the Spearman method. It is found that the water content strongly (approximately 0.99) correlates with one of the variables, although it is almost a cavity linearly described by it.

It also turned out that it is incorrect to use one model to predict three specified target variables. It is necessary to consider separately models for 3 variables (or one model for Water Content and one for Liquid/Gas).

Zeros for Liquid and Gas were studied separately using the XGBoost model, since this algorithm overwhelmingly wins over all others in the classification problem in various studies and, in particular, in the well-known international competition Data Scientist's Kaggle. It was found out that the model does not work very well for classification (especially Liquid, judging by the Confusion Matrix).The Roc-Auc model is also not large enough to consider the classification successful. However, it is worth trying to additionally find the best thresholds for solving this problem. You can also try adding new features.

For further research, it is necessary to do Feature Engineering and use the RFE algorithm to remove insignificant features. To improve the result, it is also recommended to increase the sample, since it is possible that a given number of objects cannot describe the general population due to the fact that the data is quite difficult to study, a task in which the data is so strongly experiencing jumps is very rare in real life.

References

[1.] UDC 004.414.23 K. V. Gulakov "CHOICE OF NEURAL NETWORK ARCHITECTURE FOR solving PROBLEMS OF approximation AND regression analysis OF experimental DATA"
[2.] V. I. Shmyrov BACHELOR'S THESIS " Application of radial-basis neural networks in process control problems».
[3.] HTTPS://HABR.COM/RU/COMPANY/ANTIPLAGIAT/BLOG/528384/
[4.] CHRIS MCCORMICK, RADIAL BASIS FUNCTION NETWORKS HTTPS://MCCORMICKML.COM/2013/08/15/RADIAL-BASIS-FUNCTION- NETWORK-RBFN-TUTORIAL/
[5.] RBF implementationhttps: //github.com/ZilinXiang/Radial-Basis Function-Network
[6.] Deep Learning Course, MIPT, 2019 https://github.com/DLSchool
[7.] Specialization “Machine learning and data analysis", from the partner MOSCOW INSTITUTE OF PHYSICS and TECHNOLOGY, YANDEX, E-LEARNING DEVELOPMENT FUND https://www.coursera.org/specializations/machine-learning-data-analysis
[8.] SHAP interpretation of ML models https://habr.com/ru/post/428213/
[9.] Interpreting models using the Captum library https://medium.com/pytorch/introduction-to-captum-a-model-interpretability- library-for-pytorch-d236592d8afa
[10.] Official website of the Captum library https://captum.ai/
[11.] Game theory for model interpretation https://towardsdatascience.com/the-ultimate-guide-using-game-theory-to-interpret- machine-learning-c384cbb6929
[12.] XGBoost algorithm https://neerc.ifmo.ru/wiki/index.php?title=XGBoost
[13.] A histogram and a box with a mustache on your fingers https://habr.com/ru/post/267123/
[14.] ViolinPlot https://en.wikipedia.org/wiki/Violin_plot
[15.] Scatter plot https://ru.wikipedia.org/wiki/Диаграмма_рассеяния
[16.] Recursive Feature Elimination Algorithm https://habr.com/ru/company/otus/blog/528676/ https://tproger.ru/translations/feature-engineering-in-machine-learning/
[17.] Interesting clustering algorithms, part two: DBSCAN https://habr.com/ru/post/322034/
[18.] Volzhina Elena Grigoryevna " Detection and classification of magnetic field anomalies using clustering algorithms” Course work Supervisor: Grafeeva N. G. https://math.spbu.ru/SD_AIS/documents/2014-12-341/2014-12-tw-25.pdf
[19.] Spearman's criterion https://medstatistic.ru/methods/methods9.html
[20.] Search for anomalies with One-Class SVM https://otus.ru/nest/post/888/
[21.] The Shapiro-Wilk test http://www.machinelearning.ru/wiki/index.php?title=Criteria_shapiro-Wilka
[22.] INTRODUCTION TO MATHEMATICAL STATISTICS GORITSKY Yu. A. Textbook on the course "Probability Theory and Mathematical Statistics" Moscow MEI Publishing House 2015 UDC 519
[23.] PCA Principal component method