Parallel Implementation of Neural Networks for Solving the Problem of Oil Production

Aksultan A. Mukhanbet [1] and Bazargul Matkerim [1]

[1]Al-Farabi Kazakh National University, 71 Al-Farabi Avenue, Almaty, 050040, Kazakhstan

Abstract

Oil production is a pressing challenge in the modern energy sector. Artificial intelligence and neural networks are widely used to enhance the efficiency of the oil and gas extraction processes. However, processing large volumes of data related to oil production requires the efficient parallel implementation of machine learning algorithms. This research addresses the issue of parallel implementation of neural networks for solving oil production tasks. We used data-level parallelism, splitting, and processing in parallel. In addition, parallelism was employed to distribute the training across multiple nodes (processes) and gather the training results. For this purpose, a dataset was created using the Buckley-Leverett model, which allowed us to obtain extensive data on oil reservoirs. The parallel implementation of machine-learning algorithms significantly accelerates the training process of neural networks and enhances the accuracy of their data analysis in oil production. Our work makes a significant contribution to optimizing the oil extraction industry and demonstrates the successful application of parallel data processing in solving complex tasks in this field. MPI technology was used for parallelization, resulting in a twofold acceleration time. The accuracy of the neural network is 98%.

Keywords

Oil production, Neural networks, Parallel implementation, Machine learning, Optimization.

1. Introduction

In recent decades, neural networks have been widely used in various scientific and industrial fields. One such field is the oil industry, in which the application of neural networks can significantly improve oilfield development and management. In this article, we explore the parallel implementation of neural networks for solving oil production tasks and provide an overview of scientific papers dedicated to this topic.

Oil production, as a crucial component of modern energy, faces constant challenges and optimization tasks related to the extraction of oil and gas from Earth's depths. With each passing day, the volume of data associated with oil production is growing, and innovative methods are required for effective industrial management. In this context, artificial intelligence and neural networks have reached the rescue stage, offering powerful tools for analyzing and optimizing production processes.

However, we are faced with a complex task: processing vast amounts of data related to oil production in real time while maintaining a high accuracy of analysis. The answer to this question lies in the efficient parallel implementation of machine learning algorithms designed specifically to work with oil production data. Several studies have been conducted on parallel implementation of these algorithms. In [1] proposes an edge computing health model using P2P-based deep neural networks. The model utilizes multiple edge nodes connected to a deep neural network model to process health big data in parallel, reducing response time delays. In [2], systematic methods were proposed based on a graph-theoretical approach for mapping neural networks onto cellular SIMD arrays. The authors have achieved significant results over time. In works [3-4], the implementation of the full training procedure of artificial neural networks (ANN) for speech recognition using the backpropagation algorithm in block mode and an intelligent

DTESI 2023: Proceedings of the 8th International Conference on Digital Technologies in Education, Science and Industry, December 06–07, 2023, Almaty, Kazakhstan

mukhanbetaksultan0414@gmail.com (A.A. Mukhanbet); bazargul.matkerim@gmail.com (B. Matkerim)

0000-0003-4699-0436 (A.A. Mukhanbet); 0000-0002-5336-687X (B. Matkerim)

classifier for two-dimensional objects was considered. By training acoustic models for speech recognition with a large vocabulary, a 6-fold reduction in the time required to train large real networks and a significant reduction in both training and recognition times were achieved. Reference [5] compared parallel and sequential implementations of feedforward neural networks. The modeling results show that for small networks, sequential implementation outperforms parallel implementation. However, as the network and training dataset sizes increase, parallel implementation leads to shorter training times than those of sequential implementation. The parallel implementation of complex models such as deep neural networks (DNN) was evaluated in [6]. The distributed training performance of the Tianhe-3 prototype for deep neural networks using an improved LeNet model, classic models AlexNet, VGG16, and ResNet18 in a ported distributed PyTorch environment, and a dynamic selection algorithm for the main communication mechanism showed significant time reductions.

The implementation of neural networks in oil production tasks has also been demonstrated in works [7-8]. These studies discussed various types of machine learning and artificial intelligence methods that can be used for data processing and interpretation in various sectors of the oil and gas industry. Works [9-10] consider convolutional neural networks (CNN) and long short-term memory (LSTM) networks as a comparative approach to forecasting oil production rates in an Iranian oil field. The highest performance was observed in the results of the rough-NN (CNN) with a determination coefficient of 0.82 for forecasting test data. In addition, [11] compared machine learning classifiers, neural networks, and recurrent neural networks. The results showed that the Gradient Boosting classifier and neural network demonstrated high accuracy of 99.99% and 97.4%, respectively. Research [12] focused on predicting oil production rates using the Levenberg-Marquardt backpropagation algorithm for training artificial neural networks (BPANN) and decline curve analysis methods (DCAM). All of the results discussed in these studies show significant and rapid improvements in both time and accuracy. However, parallel neural networks have not been specifically used in oil production tasks.

In this study, we address one of the key challenges in the field of oil production: the parallel implementation of neural networks. To achieve this goal, we compiled an extensive dataset using the Buckley-Leverett model, which allowed us to obtain detailed information about the oilfields. Parallel data processing combined with modern machine learning algorithms accelerates the training of neural networks and improves their accuracy in oil production data analysis.

This study represents a significant contribution to the optimization of the oil production industry and demonstrates how parallel data processing can effectively solve complex tasks in this strategically important field.

2. Methods and materials

In this study, the following methods were used for parallel implementation of neural networks to solve oil production tasks:

- Data collection using Buckley–Levett model.
- Parallel data processing.
- Development of parallel training algorithms.
- Optimization.
- Evaluation and analysis of results.

These points were considered separately. The Buckley–Levett model was selected for the successful implementation of parallel data processing in oil production tasks. This model is a mathematical tool widely used in geology and geophysics for modeling hydrocarbon reservoirs such as oil and gas. The Buckley-Leverett model allows us to describe the physical and geological characteristics of underground formations, which is necessary for understanding the behavior of oil reservoirs. The formula and implementation of this model are presented in [11].

Data collection began with the creation of a reservoir model based on the known geological and hydrodynamic parameters. These parameters included the geometry of the reservoir, rock properties, permeability, and viscosity. The Buckley-Leverett model was used to simulate the

behavior of fluids in underground formations over time. During the modeling process, an extensive database containing information on the physical properties and state of the reservoir at different depths and times was created. The data collected were as follows:

	Class	etta	Kviews	PORO	TIME	VISC_OIL
0	1.0	0.004140	1.0	0.130890	0.0	0.414938
1	1.0	0.033697	1.0	0.130890	200.0	0.414938
2	1.0	0.060086	1.0	0.130890	400.0	0.414938
3	1.0	0.085833	1.0	0.130890	600.0	0.414938
4	2.0	0.111229	1.0	0.130890	800.0	0.414938
...	...	...	...	...	...	...
403435	9.0	0.839421	6.0	0.107296	7000.0	0.115207
403436	9.0	0.852769	6.0	0.107296	7200.0	0.115207
403437	9.0	0.865994	6.0	0.107296	7400.0	0.115207
403438	9.0	0.879127	6.0	0.107296	7600.0	0.115207
403439	9.0	0.892180	6.0	0.107296	7800.0	0.115207

Figure 1: Data from the Buckley-Leverett model

The final data was obtained from the Bakley-Leverett model, and the collected data provided extensive information about the geological structure and dynamics of the deposits. The total number of data used was 403,439. This dataset has become a key foundation for training neural networks and analyzing data in the context of parallel implementation of machine learning algorithms.

Parallel data processing. Parallel data processing is a method that allows the simultaneous processing of large volumes of information by dividing them into smaller parts and processing each of them on separate computing nodes or processors. In the context of solving oil production tasks, this method plays an important role in data collection and preparation for training the neural networks. First, data visualization is performed, as shown in Figure 2.

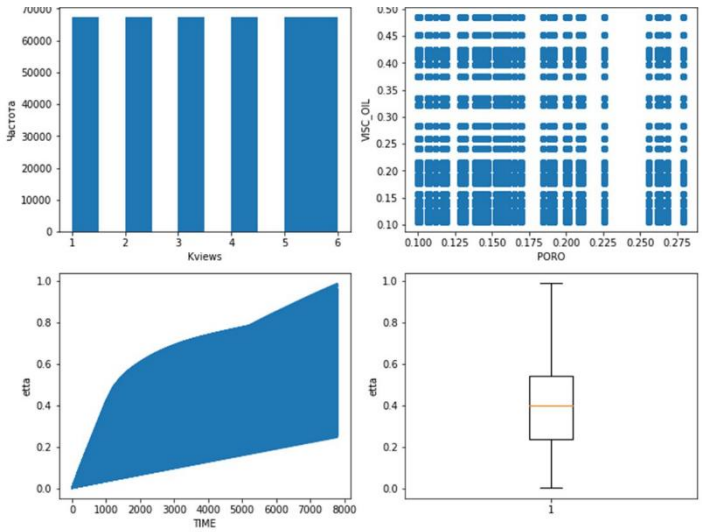

Figure 2: Data Visualization

Data Distribution. A large volume of data on oil fields obtained using the Buckley-Leverett model was divided into multiple parts. These data segments can be evenly distributed among different computational nodes and processors. Data parallelism is used in this study. Data parallelism in neural networks is a method for optimizing the training and execution of neural

networks, in which data are divided into batches and processed in parallel on different computational devices or processor cores. This speeds up training and execution because different data segments can be processed independently. Data parallelism can be implemented at the following levels.

Data Parallelism: In this case, each data batch is sent to a separate computational node (e.g., GPU), where it is processed independently. The gradients were then computed for each batch and summed to update the neural network parameters.

Model Parallelism: Here model is divided into several parts, and each part is processed using different computational nodes. This is particularly useful when the model is too large for fitting the memory of a single device.

Task Parallelism: In this case, different tasks related to neural networks are executed in parallel. For example, one part may be responsible for training, whereas the other handles inferences (predictions).

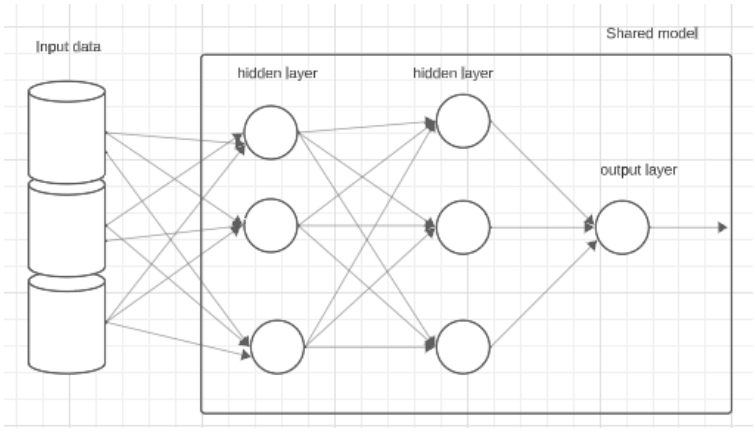

Figure 3: The Architecture of Neural Network

Data parallelism was used in this study. The main concept of data parallelism is that data are divided into parts and each part is processed in parallel. This is particularly useful in situations where there are large volumes of data to process, such as training neural networks, analyzing large datasets, or performing distributed computing. An example could be the parallel processing of images; if one has a set of images to process (e.g., apply a filter to each image), each image can be processed on a separate processor or core. This allows for faster processing because multiple images can be processed simultaneously. In this study, 403,440 data points were divided. The architecture of a traditional neural network is as follows.

The architecture of parallel data splitting can be represented as follows:

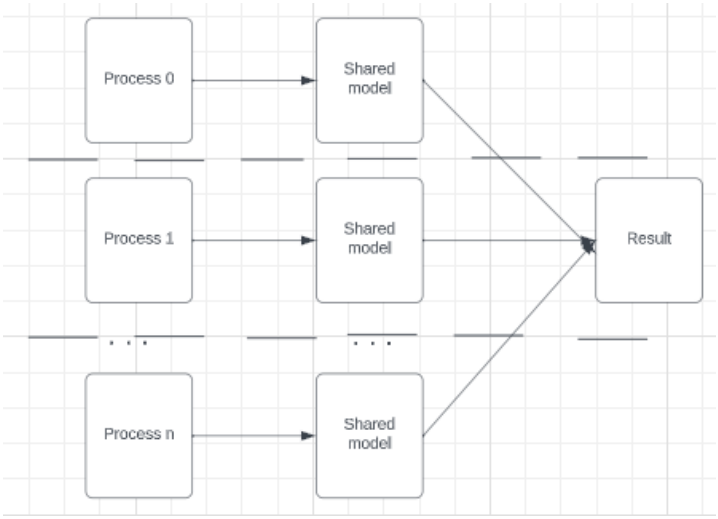

Figure 4: Data-Level Parallelism Architecture

In this architecture, the following stages are involved:

- Parallel Processing: Each computational node or processor operates independently to process data in parallel, significantly increasing the efficiency of data collection and preparation.
- Data Exchange: Information exchange between nodes may occur during processing and can be performed in parallel to reduce processing time.
- Results Aggregation: After data processing is completed on each node or processor, the results are aggregated into a unified solution.

The implementation of these methods allows for efficient training of neural networks on large volumes of oil production data, reducing training time and increasing model accuracy, ultimately contributing to process optimization in the oil industry.

In this implementation, parallelism was utilized to distribute training across multiple nodes, with the results collected during the main process. An MPI communicator is then created to link all the processes. During training, the training data and test data are passed to all processes, with each process receiving its local training dataset using slices. A sequential neural network model was then created, consisting of three layers: two layers with rectified linear unit (ReLU) activation functions and one output layer without activation. The model is compiled with the 'mean_squared_error' loss function and the 'adam' optimizer and trained on all processes. However, these results were not obtained. This process is illustrated in Figure 5.

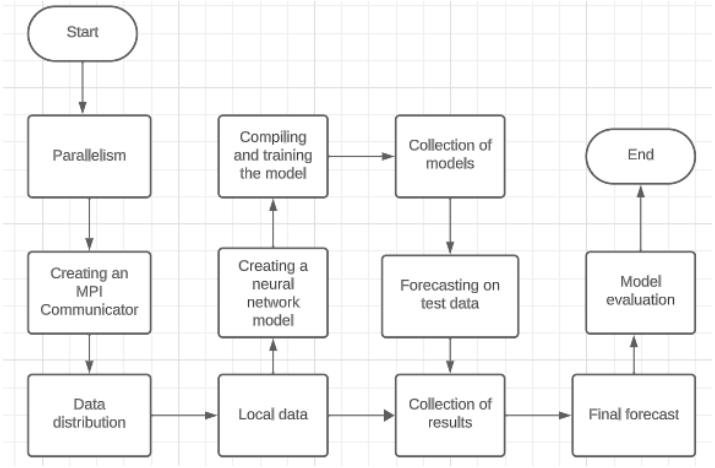

Figure 5: Parallel Training of Neural Networks Using MPI and Result Aggregation

Currently, a model collection process is in progress. The models from each process were gathered for the main process (rank=0). Then, in the main process (rank=0), predictions were made on the test data for each model, and the prediction results were collected in the main process. Next, the final prediction was performed by calculating the average of all predictions.

3. Results

Parallelism in this code is attained by apportioning the training of the model among different processes, and then consolidating the findings of the primary process for aggregation and model evaluation. Following training, the outcomes were as follows:

	Тестовые данные	Прогноз
0	0.486674	0.485748
1	0.082572	0.080607
2	0.360197	0.360908
3	0.175699	0.174224
4	0.590396	0.589551
...	...	...
13443	0.024453	0.021270
13444	0.306247	0.304377
13445	0.564379	0.564172
13446	0.374263	0.373374
13447	0.481870	0.480828

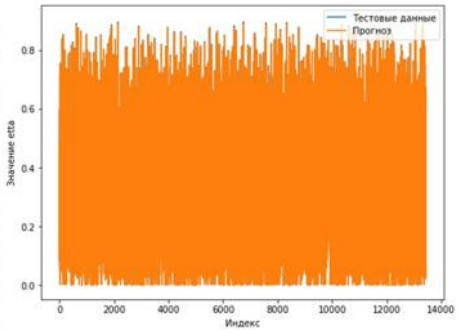

Figure 6: Visualization and Forecast of Results

Subsequently, a forecast for etta was made for a single iteration. The results are as follows:

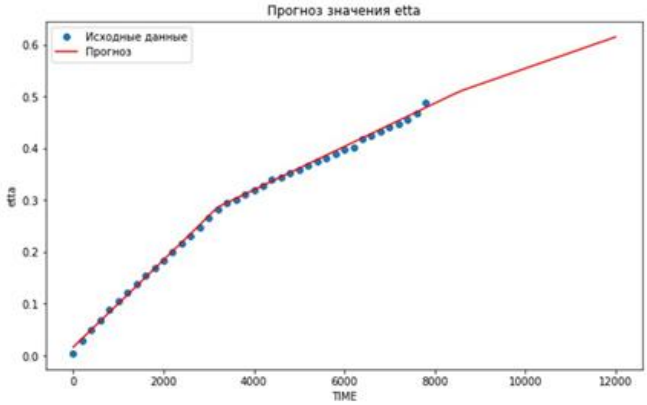

Figure 7: Forecast of the eta value for one iteration

It can be observed from the figure that the forecasted values of eta have high accuracy and are more similar to the real data. To compare the model with parallelized networks, it was necessary to assess the accuracy and speed improvement estimates.

The speed improvement estimate is an important aspect of parallel implementation. This involves comparing the time required to train neural networks on parallel computing nodes with the time required to train a single node. Similarly, the time required for the prediction of parallel nodes and a single node were compared. This allows us to determine how much faster and more efficient models have become owing to parallel implementation.

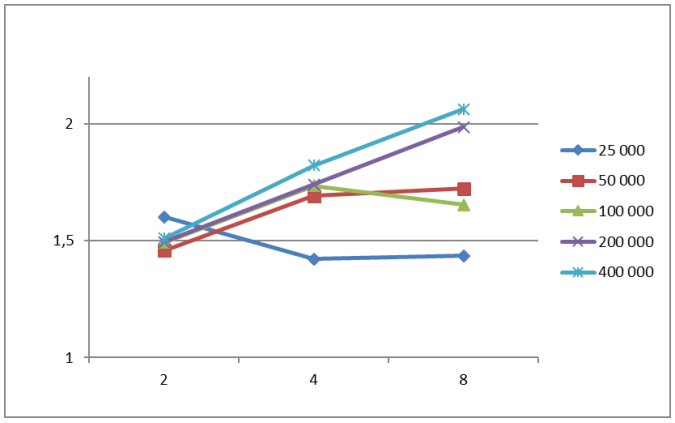

Figure 8: Speedup of MPI NN parallel program vs. the number of processes

The test results for different packet sizes when the data size is 400,000 are as follows:

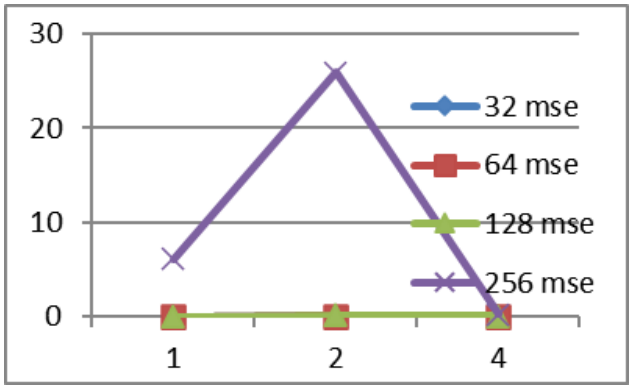

Figure 9: MSE of MPI NN parallel program

The change in the R^2 score when distributing tasks among processes depends on how the distribution is performed, as well as on the data and neural network architecture.

Parallel Model Training. In this study, several neural networks were trained in parallel using different subsets of data. This led to a change in the coefficient of determination (R^2) because each neural network was trained on different data. Therefore, the R^2 score for each model, as well as the final R^2 score after averaging the predictions, differed from the value obtained when training on the entire dataset.

R^2 evaluation results. The R^2 evaluation shows how well the model fits the data. If the distribution of data among processes leads to a better generalization of the model, then the R^2 score may increase. Figure 11 shows that, as the number of processes increased, the R^2 score decreases. This can be explained as follows. If the number of processes is too large compared to the size of the data, each model may overfit. Overfitting occurs when the model adjusts too closely to the training data and loses its ability to generalize new data. This can lead to a decrease in the generalization ability of the models and the R^2 score.

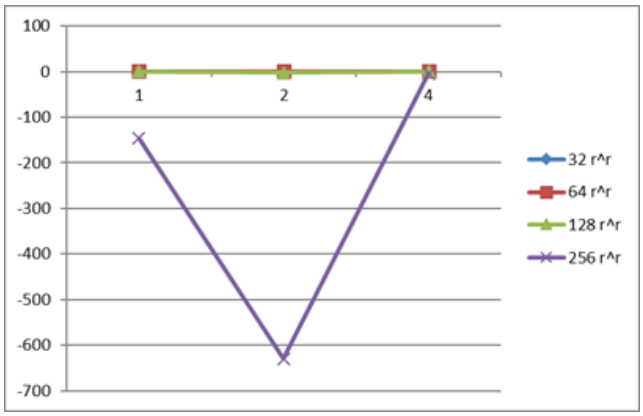

Figure 10: R2 score of MPI NN parallel program

Analysis of Model Accuracy. After completing the training process of the neural networks, their accuracy and efficiency were analyzed. This includes evaluating the ability of the models to predict various aspects of oil production, such as the production volume, oil quality, process optimization, and other key parameters. Comparative analyses of the model accuracy before and after parallel implementation were performed to determine the extent to which it improved owing to the application of parallel methods. The figure shows that the accuracy after training the neural network is 98%.

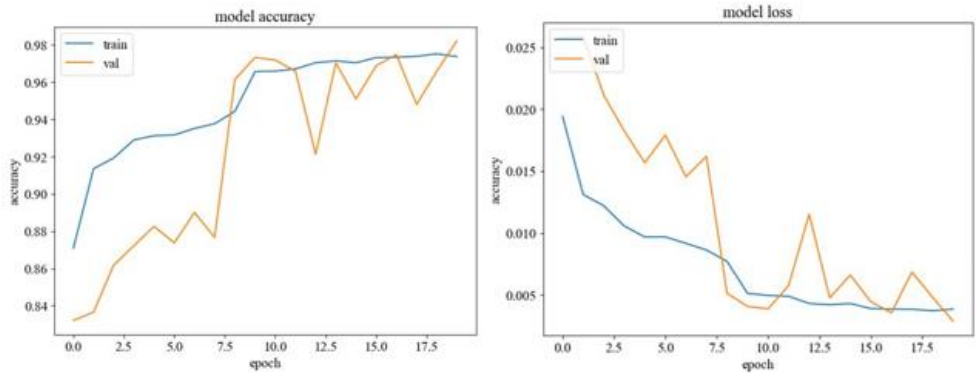

Figure 11: Visualization of Accuracy and Error Results

However, the accuracy of the parallel training of the neural network was 94%. The comparative statistics are presented in Table 1.

Table 1
Comparative Table of Results with Parallelization and without Parallelization

Neural networks and metrics	Accuracy	Execution time (in seconds)	Error
Neural network without parallelization	98%	10.21234	0.020
Neural network with parallelization	94%	1.974637	0.027

The results indicate that there is a trade-off between model accuracy and training time when using parallel computing. We now consider each aspect in greater detail.

Model Accuracy:

Without parallelization: A model trained without parallel computing achieves higher accuracy, meaning that it better fits the training data and can generalize better to new data. This is reflected in a lower error and a higher R^2 score.

With parallelization, the models trained using parallel computing exhibited lower accuracy. This may be attributed to differences in the data, overfitting of smaller subsets of data for each process, or other aspects related to parallelization.

Model Errors:

Without parallelization: A non-parallelized model has a lower error, indicating that its predictions are closer to actual data. This indicates the model's ability to make accurate forecasts.

With parallelization, the models trained using parallel computing had higher errors, implying that their predictions deviated more from the actual data. This could indicate issues related to overfitting or the limited amount of data available for each process.

Training Time:

Without parallelization, the training time was longer because the entire training process was executed sequentially on a single process or core. However, this approach can be inefficient when dealing with large volumes of data or complex models.

Parallelization significantly accelerates the training process. The training time was reduced by a factor of five, making parallelization highly advantageous for large datasets or complex models.

In conclusion, parallelization reduces the training time, but models trained in parallel may have slightly lower accuracy and higher errors compared to models trained sequentially. However, this difference was not statistically significant. The final analysis of the results allows us to draw conclusions about the success of the parallel implementation of neural networks for oil production tasks and the practical benefits they can bring to the energy and oil industry.

4. Conclusions

In this study, we focused on optimizing oil production processes using parallel implementation of neural networks. Oil production continues to be one of the key industries in the energy sector, and the effective use of modern technologies, such as artificial intelligence and machine learning, plays an important role in increasing productivity and reducing costs. Our study demonstrated that the parallel implementation of neural networks at the data level can significantly speed up the learning process and improve the accuracy of oil production data analysis. We successfully created a dataset using the Buckley-Leverett model, which allowed us to enrich our initial data and improve the quality of training. The use of MPI for parallelization has led to a significant increase in data processing speed, reducing the time required for analysis and decision-making. The overall result of our study is an important step towards optimizing the oil industry. We demonstrated that parallel data processing and machine learning can work in symbiosis to speed up processes and improve outcomes. The proposed neural network achieved an accuracy of 98%, thereby confirming the successful application of the proposed approach. Our study provides new insights into the effective use of modern technologies to optimize oil production and create a basis for future research and development in this area.

5. Acknowledgements

This research was carried out at the expense of a grant from the Science Committee of the Ministry of Education and Science of the Republic of Kazakhstan under project № BR18574136.

6. References

[1] Chung, K., Yoo, H. (2020). Edge computing health model using P2P-based deep neural networks. Peer-to-Peer Netw. Appl. 13, 694–703. https://doi.org/10.1007/s12083-019-00738-y.

[2] Wojtek Przytula, K., Prasanna, V.K. & Lin, WM. (1992). Parallel implementation of neural networks. J VLSI Sign Process Syst Sign Image Video Technol 4: 111–123 https://doi.org/10.1007/BF00925117.

[3] Stefano Scanzio, Sandro Cumani, Roberto Gemello, Franco Mana, P. Laface. PARALLEL IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORK TRAINING. Conference: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2010, 14-19, Sheraton Dallas Hotel, Dallas, Texas, USA.

[4] Enrique V. Carrera, Cicero Pereira. A Parallel Implementation of a Neural-Network Based Object Classifier. February 1999. Conference: 3rd Workshop on Cybernetic Vision At: Campinas - SP, Brazil Volume: 1.

[5] Shou King Foo, P. Saratchandran, N. Sundararajan. (1995). Comparison of parallel and serial implementation of feedforward neural networks. Journal of Microcomputer Applications. 18(1): 83-94.

[6] Wei J, Zhang X, Ji Z, Li J, Wei Z. (2021). Deploying and scaling distributed parallel deep neural networks on the Tianhe-3 prototype system. Sci Rep. 12;11(1):20244. doi: 10.1038/s41598-021-98794-z. PMID: 34642373; PMCID: PMC8511035.

[7] Husam H. Alkinani, Abo Taleb T. Al-Hameedi, Shari Dunn-Norman, and Ralph E. Flori. Applications of Artificial Neural Networks in the Petroleum Industry: A Review. Missouri University of Science and Technology; Mortadha T. Alsaba, Australian College of Kuwait; Ahmed S. Amer, Newpark TechnologyCenter/ Newpark Drilling Fluids. 18-21 March 2019.

[8] Anirbid Sircar, Kriti Yadav, Kamakshi Rayavarapu, Namrata Bist, Hemangi Oza. (2021). Application of machine learning and artificial intelligence in oil and gas industry. Petroleum Research, 6(4): 379-39. https://doi.org/10.1016/j.ptlrs.2021.05.009.

[9] Kwon S, Gayoung Park, Jang Y, Jinhyung Cho, Chu M, Baehyun Min. (2021). Determination of oil-well placement using a convolutional neural network coupled with robust optimization under geological uncertainty. Journal of Petroleum Science and Engineering. 201. p.108118. https://doi.org/10.1016/j.petrol.2020.10811.

[10] Amin Sheikhoushaghi, Narges Yarahmadi Gharaei, Amirhossein Nikoofard. (2022). Application of Rough Neural Network to forecast oil production rate of an oil field in a comparative study. Journal of Petroleum Science and Engineering. 209. p.109935. https://doi.org/10.1016/j.petrol.2021.109935.
[11] Daribayev, B., Mukhanbet, A., Nurakhov, Y., Imankulov, T. (2021). Implementation of the solution to the oil displacement problem using machine learning classifiers and neural networks. Eastern-European Journal of Enterprise Technologies, 5 (4 (113)), 55–63. https://doi.org/10.15587/1729-4061.2021.241858.
[12] Marfo, S.A. and Kporxah, C. (2020), Predicting Oil Production Rate Using Artificial Neural Network and Decline Curve Analytical Methods, Proceedings of 6th UMaT Biennial International Mining and Mineral Conference, Tarkwa, Ghana, pp. 43-50.