

Article

Research on Prediction of Movable Fluid Percentage in Unconventional Reservoir Based on Deep Learning

Jiuxin Wang ¹, Yutian Luo ^{1,2}, Zhengming Yang ^{1,2,*}, Xinli Zhao ^{1,2,3} and Zhongkun Niu ^{1,2,3}

¹ Enhanced Oil Recovery Research Center, Research Institute of Petroleum Exploration & Development, Beijing 100083, China; wangjxriped@163.com (J.W.); luoyutian@petrochina.com.cn (Y.L.); ucaszxl@163.com (X.Z.); niuzhongkun19@mails.ucas.ac.cn (Z.N.)

² Institute of Porous Flow and Fluid Mechanics, Chinese Academy of Sciences, Langfang 065007, China

³ School of Engineering Science, University of Chinese Academy of Sciences, Beijing 100049, China

* Correspondence: yzhm69@petrochina.com.cn

Abstract: In order to improve the measurement speed and prediction accuracy of unconventional reservoir parameters, the deep neural network (DNN) is used to predict movable fluid percentage of unconventional reservoirs. The Adam optimizer is used in the DNN model to ensure the stability and accuracy of the model in the gradient descent process, and the prediction effect is compared with the back propagation neural network (BPNN), K-nearest neighbor (KNN), and support vector regression model (SVR). During network training, L_2 regularization is used to avoid over-fitting and improve the generalization ability of the model. Taking nuclear magnetic resonance (NMR) T_2 spectrum data of laboratory unconventional core as input features, the influence of model hyperparameters on the prediction accuracy of reservoir movable fluids is also experimentally analyzed. Experimental results show that, compared with BPNN, KNN, and SVR, the deep neural network model has a better prediction effect on movable fluid percentage of unconventional reservoirs; when the model depth is five layers, the prediction accuracy of movable fluid percentage reaches the highest value, the predicted value of the DNN model is in high agreement with the laboratory measured value. Therefore, the movable fluid percentage prediction model of unconventional oil reservoirs based on the deep neural network model can provide certain guidance for the intelligent development of the laboratory's reservoir parameter measurement.

Keywords: deep learning; NMR T_2 spectrum; movable fluid percentage; deep neural network; unconventional reservoir



Citation: Wang, J.; Luo, Y.; Yang, Z.; Zhao, X.; Niu, Z. Research on Prediction of Movable Fluid Percentage in Unconventional Reservoir Based on Deep Learning. *Appl. Sci.* **2021**, *11*, 3589. <https://doi.org/10.3390/app11083589>

Academic Editor: Kun Sang Lee

Received: 23 March 2021

Accepted: 13 April 2021

Published: 16 April 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The fluids in unconventional oil reservoirs can be divided into two categories according to their existence states: one is bound fluid (immovable fluid), and the other is free fluid (movable fluid) [1]. The bound fluid exists in the extremely tiny pores and walls of the larger pores. The bound fluid in the smaller pores is difficult to flow due to the large capillary force; the fluid in the middle of the larger pores is subject to the smaller capillary force. It can flow under a certain driving pressure, so it is called movable fluid. The presence of boundary fluids often leads to the reduction of seepage space in the reservoir pores and the increase of seepage resistance. The more movable fluid in the reservoir, the stronger seepage capacity of the corresponding reservoir, and more oil and gas resources can be recovered [2]. In conventional reservoir evaluation, researchers generally use porosity and permeability as the characterization parameters of reservoir physical properties. However, the experimental evaluation of movable fluids and unconventional reservoir core experiments show that: there is a good positive correlation between the oil displacement efficiency and the percentage of movable fluid [2], which also proves that movable fluid percentage can better reflect the development potential of unconventional reservoirs than permeability. At present, the most reliable measurement of the movable

fluid percentage of the reservoir is the core nuclear magnetic resonance technology [3]. The current disadvantages of this technology are the long experimental period and the large amount of manpower required, which make it difficult to synchronize the evaluation of the reservoir with the development and deployment of the oil field.

In recent years, with the development of science and technology, artificial intelligence methods have been increasingly applied to the petroleum industry: Yushu and Qidi used the XGBoost algorithm to identify complex carbonate rock lithology with an accuracy rate of 88.18% [4,5]. Mohamed used machine learning methods to study lithology classification, and concluded that the classification accuracy of supervised learning algorithms was better than that of unsupervised algorithms [6]; Liuqing predicted the porosity of sandstone reservoirs based on deep neural network and logging data. The correlation between the predicted value of the model and the actual porosity is as high as 0.9725 [7]. Yuyang combined the NMR transverse relaxation time spectrum and mercury intrusion data to intelligently predict the permeability of sandstone reservoirs through BP neural network and achieved a good prediction accuracy [8]; Dongxiao realized the automatic generation of logging curves based on cyclic neural network [9]. Ye predicted unconventional reservoir saturation based on NMR logging data and machine learning methods [10]. Deep learning was first proposed by Hinton in 2006. The ability to achieve complex nonlinear fitting through artificial neural networks with multiple hidden layers had greatly improved the prediction and classification accuracy of artificial intelligence models [11]. NMR logging data are affected by many factors in the reservoir, and the effective information contained in NMR logging often has large deviations. In order to improve measuring speed and accuracy of the movable fluid percentage of unconventional reservoirs, this article is based on the deep learning method and the unconventional reservoir core NMR T_2 spectrum data of laboratory measurement to predict movable fluid percentage of unconventional oil reservoirs.

2. Data Source and Experimental Methodology

2.1. Correlation Analysis between NMR T_2 Spectrum and Percentage of Movable Fluid

By comparing and studying the NMR T_2 spectrum of oil-saturated cores with different degree of tightness in Figure 1, it was found that: (1) with the increase of reservoir tightness, the left peak of the NMR T_2 spectrum of oil-saturated cores gradually rose and shifted to the left, the right peak gradually decreased or even disappeared; (2) the proportion of movable fluid in the reservoir gradually decreased, and the boundary fluid gradually increased. Through the above comparative studies, there was a strong correlation between the shape characteristics of the oil-saturated core NMR T_2 spectrum of unconventional oil reservoirs and movable fluid percentage of reservoirs. Therefore, this article predicted the percentage of movable fluid in unconventional oil reservoirs based on the shape characteristics of the oil-saturated core NMR T_2 spectrum.

2.2. Data Source and Preprocessing

In this paper, a total of 580 unconventional reservoir cores NMR T_2 spectrums and the corresponding movable fluid percentage were collected, all of which were obtained through the laboratory measurement. The movable fluid percentage of the reservoir was predicted based on the shape characteristics of the core NMR T_2 spectrum of unconventional oil reservoirs. By discretizing the NMR T_2 spectrum of the core, the horizontal axis T_2 relaxation time of each discrete point was fixed. At this time, all discrete points' T_2 distribution values could represent the shape characteristics of the core NMR T_2 spectrum [11]. The result of the discretization processing of different cores' NMR T_2 spectrum is shown in Figure 2. In the unconventional reservoir NMR T_2 spectrum, T_2 distribution values of discrete points with a relaxation time greater than 1000ms is basically 0. Therefore, for the nuclear magnetic resonance T_2 spectrum of each core, collecting T_2 distribution values of the first 55 discrete points could completely extract the shape characteristics of the core

NMR T_2 spectrum. Before model training, the data needed to be standardized. The specific standardized processing methods were as follows:

$$a_i = \frac{x_i - \mu}{\sigma}, \tag{1}$$

where: a_i and x_i are the parameter value after standardization and the original parameter value, μ is the average value of input parameters, and σ is the standard deviation of input parameters.

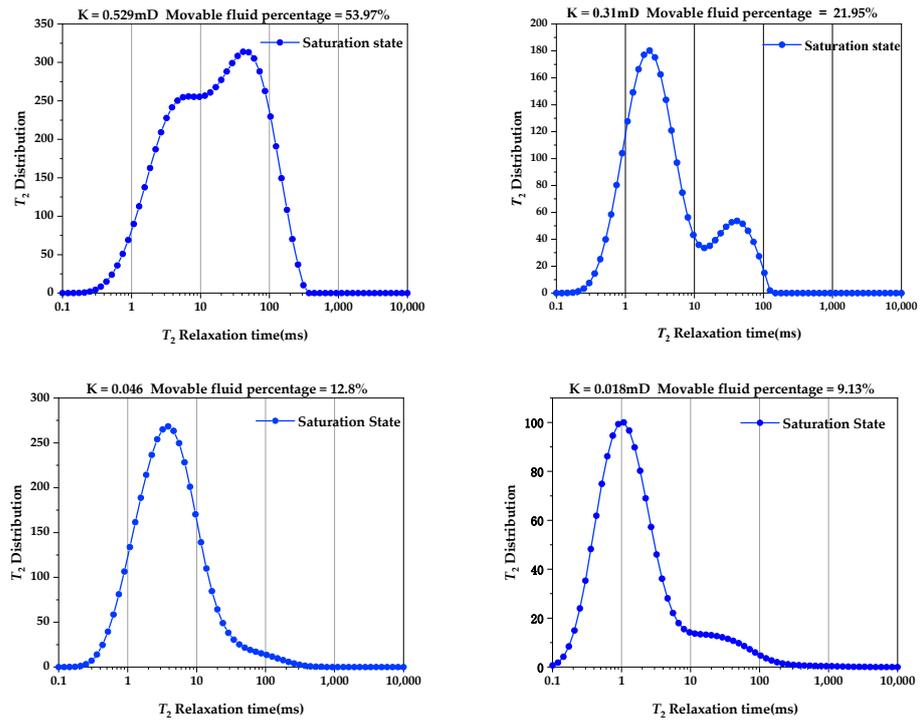


Figure 1. NMR T_2 spectrums of cores with different tightness.

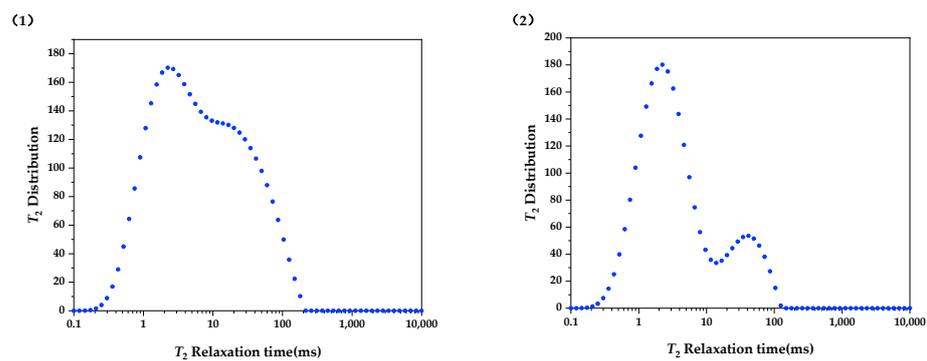


Figure 2. Discretization of the core NMR T_2 spectrum. (1) Core sample 1; (2) Core sample 2.

The original data set was divided into a training set and a test set. The training set was mainly used for model learning, and the test set was used for evaluating the effect of model learning. The training set contained 500 cores NMR data, and the test dataset contained 80 cores NMR data.

2.3. Principles of Deep Neural Networks

2.3.1. Feedforward Algorithm of Deep Neural Networks

In the deep neural network, the final output value of the model is obtained by complex nonlinear operations on the input vector, weight vectors and bias vectors [12]. Assuming that the deep neural network has a total of six layers of neurons, this article took the i -th neuron in the L -th layer as an example for calculation:

$$\begin{cases} z_i^L = \sum_{j=1}^{M_{l-1}} w_{ij}^{(L)} a_j^{(L-1)} + b_i^{(L)} \\ a_i^{(L)} = f_L(z_i^{(L)}) \end{cases} \quad (2)$$

where: z_i^L represents the input value of the i -th neuron in the L -th layer, $w_{ij}^{(L)}$ is the weight of $a_j^{(L-1)}$ connected to z_i^L , $b_i^{(L)}$ is the bias of $a_i^{(L-1)}$ connected to z_i^L , $a_i^{(L)}$ is the output value of the i -th neuron in the L -th layer, $f_L(\cdot)$ represents the activation function of the i -th neuron in the L -th layer, when $L = 6$, $a_i^{(L)}$ represents the output vector.

2.3.2. Back Propagation Algorithm of Deep Neural Networks

The backpropagation algorithm was used to calculate the partial derivative of the loss function $\zeta(y, \hat{y})$ to the model parameters, which was used to update the model's parameters. Because the calculation of $\frac{\partial \zeta(y, \hat{y})}{\partial w_{ij}^{(l)}}$ involves the partial differentiation of the vector to the matrix, the calculation process is cumbersome and complicated, so the backpropagation algorithm obtained according to the chain rule could be greatly simplified the calculation process [13]. The meaning of the backpropagation algorithm is: the error term of the l -th layer was obtained by multiplying the weight of the error term of the neurons of the $(l + 1)$ -th layer and the gradient of the activation function of the neurons of the l -th layer [14]. The Equation (3) is the calculation formula of the sensitivity error term of the l -th layer neurons. After calculating the sensitive error term of the l -th layer, the partial derivative of the loss function to the weight and bias of the neuron of the l -th layer could be obtained to achieve the parameter update. Equations (3)–(5) are the calculation formula for the above process.

$$\delta^l = \frac{\partial \zeta(y, \hat{y})}{\partial z^{(l)}} = f'_l(z^{(l)}) \odot \left((W^{(l+1)})^T \delta^{(l+1)} \right), \quad (3)$$

$$\frac{\partial \zeta(y, \hat{y})}{\partial W^l} = \delta^l (a^{(l-1)})^T, \quad (4)$$

$$\frac{\partial \zeta(y, \hat{y})}{\partial b^{(l)}} = \delta^l, \quad (5)$$

where: l represents the l -th neuron layer, δ^l the sensitive error term of the l -th neuron layer, $a^{(l-1)}$ is the output value of the $(l-1)$ -th neuron layer, and $f'_l(\cdot)$ presents the derivative of activation function of the l -th neuron layer, \odot represents the vector product, W^l represents weights value of the l -th neuron layer, and $b^{(l)}$ represents all the biases of the l -th neuron layer.

2.3.3. Adam Optimization Algorithm

In the deep neural network's training process, Adam was selected as the optimizer for model parameter update. Adam is a fusion of momentum method [14] and RMSprop algorithm [15]. It not only uses momentum as the direction of parameter update, but also adjusts the learning rate adaptively to ensure the accuracy and stability of the gradient descent during training [16]. The Adam optimizer calculates the exponentially weighted average of the gradient square g_t^2 on the one hand (similar to the RMSprop algorithm), and

on the other hand calculates the exponentially weighted average of the gradient g_t (similar to the momentum method). The Adam optimization is expressed by:

$$\begin{cases} M_t = \beta_1 M_{t-1} + (1 - \beta_1) g_t \\ G_t = \beta_2 G_{t-1} + (1 - \beta_2) g_t \times g_t \end{cases} \quad (6)$$

where: β_1 and β_2 are the attenuation rates of the two moving averages, usually $\beta_1 = 0.9$, $\beta_2 = 0.99$, M_t is the exponentially weighted average gradient, and G_t is the square of the average gradient.

When $M_0 = 0$, $G_0 = 0$, the value of M_t and G_t will be smaller than the true mean and variance at the beginning of the iteration, especially when β_1 and β_2 are both close to 1, the deviation will be greater, so the deviation should make corrections:

$$\begin{cases} \hat{M}_t = \frac{M_t}{1 - \beta_1^t} \\ \hat{G}_t = \frac{G_t}{1 - \beta_2^t} \end{cases} \quad (7)$$

where: \hat{M}_t and \hat{G}_t are deviation correction values of the exponentially weighted average gradient and the average gradient, β_1 and β_2 are attenuation rates of the two moving averages, and t is the time step.

Finally, the modified gradient value was used to update parameters of the model, and the update formula is:

$$\theta = \theta - \frac{\alpha}{\sqrt{\hat{G}_t + \epsilon}} \hat{M}_t \quad (8)$$

where: α is the learning rate, ϵ is the constant of stable value, $\epsilon = 1 \times 10^{-8}$, θ represents parameters of the model.

2.4. Experimental Comparison Models

2.4.1. BP Neural Network Model

Back propagation neural network (BPNN) is a forward-propagation neural network model trained based on the back error propagation algorithm. Through training, it learns the inherent feature relationship between the input vector and the output vector, and continuously updates the model weights through the gradient descent algorithm to achieve non-linear mapping between input features and output values [14]. The BPNN model in this experiment consisted of an input layer, a hidden layer, and an output layer. The neuron node of input layer was set to 55, the number of hidden layer neurons was 200, the output layer neuron node was set to 1, and the learning rate was set to 0.005. The Relu(Rectified Linear Units) function is used as the activation function of the hidden layer. The maximum number of training iterations was 1000 times.

2.4.2. K-Nearest Neighbor Regression Model

The K-nearest neighbor (KNN) model is a simple supervised learning algorithm. The input of the K-nearest neighbor method is the feature vector of the instance, it corresponds to the point in the feature space, and the output is the predicted value of the instance [17]. When the K-nearest neighbor model is used as regression model, it is assumed that a training data set is given, and the label value of each data has been calibrated; The KNN model calculates the average value of label values of the K nearest neighbor training instances of the new instance as the output of the model. In this experiment, the K value of the KNN model was set to 10, and the method of calculating the distance between different instances was the Euclidean distance.

2.4.3. Support Vector Regression Model

Support vector regression (SVR) model is one of the most widely used models in machine learning. It was proposed by former Soviet Union scientists Vladimir Vapnik and Alexey Chervonenkis in 1963 and 1995 respectively [18]. For a sample data (x, y) , general

regression models are usually based on the direct difference between the model's predicted value $f(x)$ and the true label value y to calculate the loss. Only when the predicted value $f(x)$ is completely equal to the y value, the regression model's loss function is 0. There is a big difference between the SVR model and the general regression model, which is SVR allows a maximum deviation of ϵ between the predicted value $f(x)$ and y . Only when the direct deviation between $f(x)$ and y is greater than ϵ , SVR will calculate the error between the two. It is equivalent to taking $f(x)$ as the center and establishing an interval band with a width of 2ϵ . When the predicted value of the sample falls within the interval band, the prediction is considered accurate [19]. SVR tries to find the optimal hyperplane to minimize the deviation from all sample points to the optimal hyperplane. Seeking the optimal hyperplane is equivalent to finding the maximum interval. In this experiment, the support vector regression model used the radial basis function as the kernel function, the regularization constant C was set to 5 and gamma was set to 0.02.

2.5. Model Evaluation Method

This article used the root mean square error function (RMSE) and the R_2 coefficient to measure the prediction accuracy of the model. The R_2 coefficient is a method to measure the correlation between true values and predicted values. The formula is as follows:

$$R_2 = 1 - \frac{\sum_{i=1}^m (f(x_i) - y_i)^2}{\sum_{i=1}^m (y_i - \bar{y})^2} \quad (9)$$

where: $f(x_i)$ represents the predicted value of the i -th sample's the movable fluid percentage, y_i represents the true movable fluid percentage of the i -th sample, and \bar{y} represents the average value of true movable fluid percentage of all samples.

RMSE reflects the error between the true movable fluid percentage of the reservoir and the predicted movable fluid percentage. The formula is as follows:

$$RMSE = \sqrt{\sum_{i=1}^N (y_i - f(x_i))^2 \times \frac{1}{N}}, \quad (10)$$

where: y_i is the true movable fluid percentage of the i -th sample, $f(x_i)$ is the predicted movable fluid percentage of the i -th sample. N is the total number of samples.

3. Optimization of Deep Neural Network's Hyperparameters

This experiment used Tensorflow developed by Google as the implementation platform. Tensorflow supports automatic derivation and doesn't need to manually write the derivation code, and the neural network's structure can be freely designed [20]. Hornik found that any function can be approximated when using a neural network with more than three layers through research [21]. In the training process of the model, it is necessary to optimize the hyperparameters, otherwise the model is prone to high deviation or high variance. In this experiment, L_2 regularization was selected as a method to prevent the model overfitting, and the regularization coefficient was set to 0.01. Relu was selected as the activation function to accelerate the update of parameters

3.1. Optimization of Learning Rate

The learning rate is an important hyperparameter in deep neural networks' training. In the gradient descent method, the value of the learning rate is very critical. If the learning rate is too large, the model cannot converge, and if the learning rate is too small, the convergence speed of the model is too slow. The experiment began to set DNN model parameters based on experience, given the hidden layer $n = 2$, the neural network structure was 55-200-160-1, and the number of training times was 1000. Figure 3 shows changes of RMSE in training set under different learning rates during training. When the learning rate was 0.01, the model's training error value dropped rapidly in the early stage of training, and it could also converge at the end of training. Compared with other curves, the root

mean square error curve with a learning rate of 0.01 was smoother and had less fluctuation, so 0.01 was selected as the optimal learning rate for this experiment.

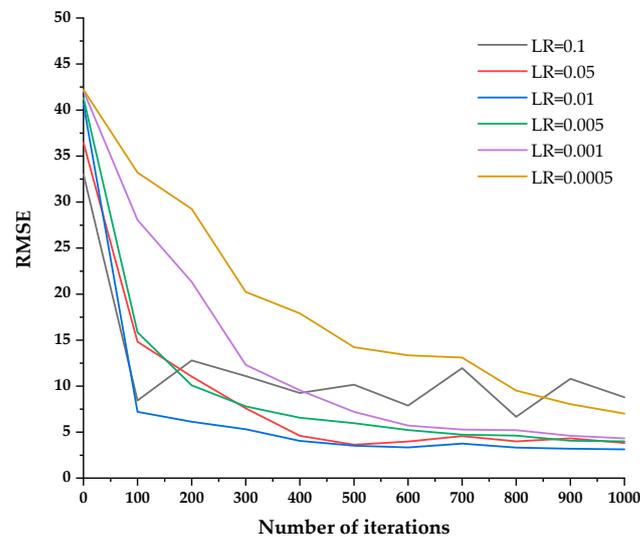


Figure 3. The change curves of root mean square error (RMSE) on training dataset under different learning rates.

3.2. Optimization of Hidden Layer Neuron Nodes

In order to determine the optimal number of neuron nodes in the hidden layer, this article used a grid search method to optimize the number of neuron nodes. Grid search is a method to find a suitable set of hyperparameter configurations by trying all the combinations of hyperparameters. Assuming that there are a total of K hyperparameters, and the K -th hyperparameter can take m_k values, then the total number of configuration combinations is $m_1 \times m_2 \times \dots \times m_k$. When there are too many hyperparameters or when a certain hyperparameter takes more values, the number of hyperparameter configuration combinations will increase explosively, which leads to a significant increase in the time cost for the optimal number of hidden layer neuron nodes. In order to reduce the time for searching the optimal number of hidden layer neurons, this paper adopted the following two methods:

(1) Set the possible value of each hyperparameter at an interval of 20, and the value range of each hyperparameter was 20 to 300. Taking the neuron node of the first hidden layer as an example, the possible values of the neuron node were 20,40,60, ,300.

(2) On the premise that the values of the hyperparameters that were optimized by grid search are fixed, the values of other hyperparameters were further optimized. Taking the optimization of the number of neuron nodes in the third hidden layer as an example, it was assumed that the optimal number of neuron nodes in the first and second hidden layer after grid search was 200 and 160. When optimizing the neuron node of the third hidden layer, the neuron node of the first hidden layer was set to 200 and the neuron node of the second hidden layer was set to 160, then the grid search method was used to select the optimal neuron node of the third hidden layer. The method for optimizing the number of neurons in other hidden layers was similar to the above process.

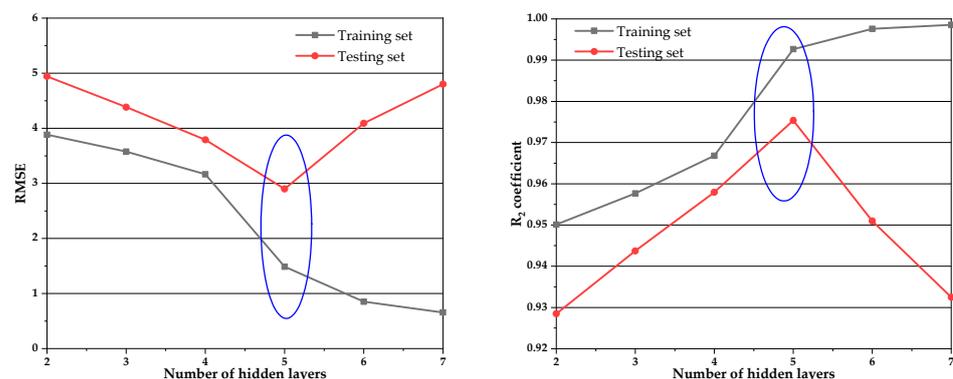
After grid search optimization, the optimal structure of DNN models with different hidden layers was finally obtained as shown in the Table 1:

Table 1. The number of optimal neurons in different hidden layers.

Number of Hidden Layers	The Number of Neurons in Each Hidden Layer
n = 2	200-160
n = 3	200-160-120
n = 4	200-160-120-80
n = 5	200-160-120-80-60
n = 6	200-160-120-80-60-20
n = 7	200-160-120-80-60-20-20

3.3. Optimal Number of Hidden Layers

In order to explore the inherent relationship between the prediction accuracy of movable fluid percentage of unconventional reservoirs based on the DNN model and the depth of the neural network, this experiment carried out a sensitivity analysis between the prediction accuracy and the depth of the neural network. Learning rates of experimental models with different hidden layers were all set to 0.01. The number of hidden layer neurons in different experimental models is shown in Table 1. It can be seen from Figure 4 that as the number of hidden layers increased, the prediction error of the deep neural network model on the test dataset continued to decrease, and the R_2 correlation coefficient continued to increase. When $n = 5$, the prediction result of the deep neural network model on the test dataset after training was the best (RMSE = 2.901, R_2 coefficient = 0.9753). This is because the complexity of the model increased as the depth of the model increased, the DNN model's ability to fit the mapping relationship between input features and output parameters was also increasing. When the number of hidden layers of the deep neural network exceeded 5, the prediction effect of the model began to deteriorate. When $n = 7$, the prediction accuracy of the model was even lower than that of $n = 3$. This is because the complexity of the model was too high, which led to the model's ability to fit the training set too strong. The model's strong ability to fit the training set led to the deterioration of the model's robustness, which made the DNN model's prediction accuracy on the test set worse. Based on the prediction accuracy of the model in Figure 4, a deep neural network with five hidden layers was selected as the best model to predict percentage of movable fluid of unconventional reservoirs.

**Figure 4.** Prediction effect of DNN models under different hidden layers on test dataset.

4. Experimental Result

4.1. Training and Evaluation Results of Different Models

Deep neural network and three contrast regression models were used to predict the percentage of movable fluid in unconventional reservoirs. The prediction results are shown in Table 2. Because the K-nearest neighbor model did not have an explicit training process [22], so KNN model could not express the RMSE and R_2 coefficients on the training

set. It can be obtained from the data in Table 2: whether it was on the training set or the test set, the deep neural network achieved better prediction results than the other three comparison models. Compared to the BPNN model, KNN model, and SVR model, the prediction errors of the deep neural network model on the test dataset were reduced by 45.89%, 61.74% and 61.84%, and the predicted correlation coefficients R_2 were increased by 6.51%, 17.29% and 17.40%. In summary, the deep neural network model had a good ability to extract the shape features of the core NMR T_2 spectrum. After models learned the training set data, the DNN model had the smallest prediction error (RMSE = 2.901) and the highest prediction correlation coefficient ($R_2 = 0.9745$) for the test dataset, which also showed that the DNN model had the best robustness.

Table 2. Evaluation results of different models.

Model	Data Set	RMSR	R_2 Correlation Coefficient
DNN	Training set	1.487	0.9926
	Testing set	2.901	0.9745
BPNN	Training set	4.359	0.9371
	Testing set	5.362	0.9158
KNN	Training set	—	—
	Testing set	7.583	0.8316
SVR	Training set	5.822	0.8878
	Testing set	7.602	0.8308

4.2. Application Results of the Deep Neural Network Model

In order to further verify the prediction effect of the deep neural network model on the percentage of movable fluid in unconventional reservoirs, we performed the prediction of the percentage of movable fluid in 10 unconventional reservoir cores from Changqing Oilfield. Firstly, the core was saturated with oil, and the nuclear magnetic resonance T_2 spectrum data of the saturated oil core were measured by the laboratory's nuclear magnetic resonance instrument. Then the laboratory method of measuring the core movable fluid percentage was used to measure the true movable fluid percentage of this 10 cores. Secondly, the trained deep neural network used the nuclear magnetic resonance T_2 spectrum data of oil-saturated cores to predict the percentage of movable fluid. The predicted value of the model and the true value obtained in the laboratory are shown in Figure 5. From the Figure 5, it can be drawn: (1) the prediction result obtained by the deep neural network model (DNN) was the closest to the result measured by the laboratory method, followed by the BP neural network model, and the worst prediction model was the SVR; (2) on the whole, the predicted value of the machine learning model was greater than the movable fluid percentage measured in the laboratory; (3) compared with the BPNN, KNN, and SVR models, the prediction RMSE of the DNN model was reduced by 39.65%, 51.18%, and 54.35%.

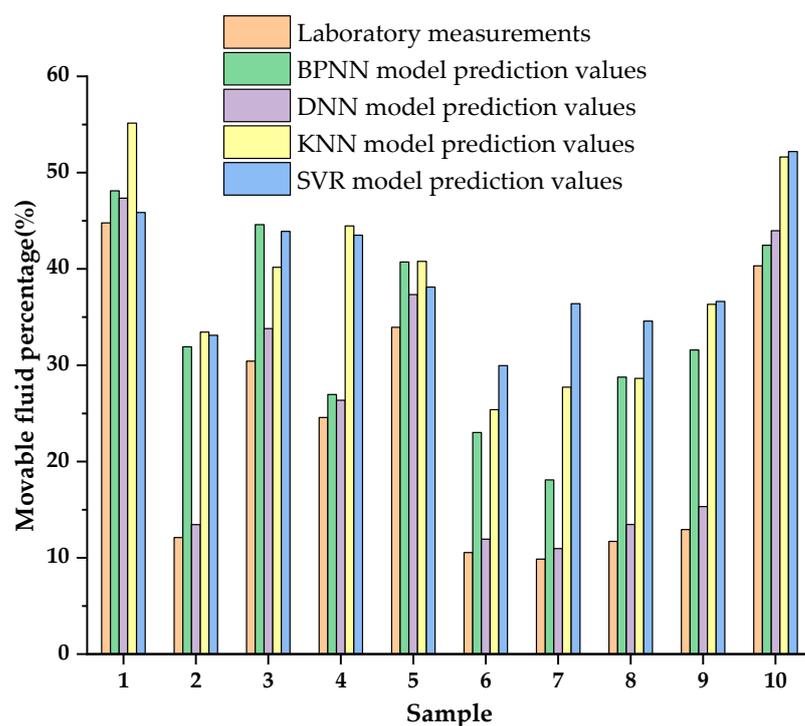


Figure 5. Prediction results of different models.

5. Conclusions

1. The deep neural network model achieved the complex non-linear mapping from the core NMR T_2 spectrum to the movable fluid percentage, and the prediction effect of DNN model was compared with that of BPNN, KNN and SVR model. The experimental results demonstrated that for the 10 core data of Changqing Oilfield, the R_2 correlation coefficient between the predicted value of the DNN model and the real movable fluid percentage of core is as high as 0.9632. The prediction RMSE of the DNN model is reduced to 2.447, and a good prediction effect is achieved.

2. Compared with the method of predicting unconventional reservoir saturation based on logging data, the method proposed in this article to predict the percentage of movable fluid in unconventional reservoirs based on laboratory NMR data has achieved better prediction results and faster prediction speed. Therefore, this method can provide certain guidance for the intelligent development of laboratory reservoir parameter measurement.

3. The study found that the prediction accuracy of DNN model gradually decreased when the number of hidden layers of the deep neural network model was greater than five. The reason for the above phenomenon may be that there are fewer training data and the model's overfitting in the later stage of training. In the future research work, the above two aspects will be optimized: (1) increasing the core NMR data of the training set; (2) taking a variety of methods to solve the overfitting in the later stage of model training.

Author Contributions: Conceptualization and methodology, Z.Y., J.W. and Y.L.; investigation, J.W.; software, J.W.; formal analysis, J.W., X.Z., Z.N.; data curation, J.W., Y.L., Z.N.; writing—original draft preparation, J.W.; funding acquisition, Y.L., Z.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by PetroChina Company Limited, grant numbers 2019D-500809.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are available on request the corresponding author (wangjxriped@163.com).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, W.; Guo, H.; Ye, C. Evaluation of the development potential of low-permeability oilfields by using nuclear magnetic resonance movable fluid. *Acta Pet.* **2001**, *22*, 40-44+44-43.
2. Yang, Z.; Miao, S.; Liu, X.; Huang, D.; Deng, C. Movable fluid percentage parameter and its application in ultra-low permeability reservoir. *J. Xi'an Shiyou Univ. (Nat. Sci. Ed.)* **2007**, *2007*, 96-99+178-179.
3. Xiao, L. Research Progress and Application of Rock Nuclear Magnetic Resonance. *Logging Technol.* **1996**, *1*, 27–31.
4. Han, Q.; Zhang, X.; Shen, W. Lithology identification technology based on gradient boosting decision tree (GBDT) algorithm. *Bull. Mineral. Petrol. Geochem.* **2018**, *37*, 1173–1180.
5. Sun, Y.; Huang, Y.; Liang, T.; Ji, H.; Xiang, P.; Xun, X. Log identification of complex carbonate lithology based on XGBoost algorithm. *Lithol. Reserv.* **2020**, *32*, 98–106.
6. Mohamed, I.M.; Mohamed, S.; Mazher, I.; Chester, P. Formation lithology classification: Insights into machine learning methods. In Proceedings of the SPE Annual Technical Conference and Exhibition, Calgary, AB, Canada, 30 September–2 October 2019.
7. Yang, L.; Cha, B.; Chen, W. Prediction method of reservoir porosity based on deep neural network. *China Sci.* **2020**, *15*, 73–80.
8. Huang, Y.; Feng, J.; Song, W.; Guan, Y.; Zhang, Z. Improved intelligent prediction method of sandstone reservoir permeability based on NMR transverse relaxation time spectrum and mercury intrusion data. *Comput. Tech. Geophys. Geochem. Explor.* **2020**, *42*, 338–344.
9. Zhang, D.; Chen, Y.; Meng, J. Synthetic well logs generation via Recurrent Neural Networks. *Pet. Explor. Dev.* **2018**, *45*, 598–607. [[CrossRef](#)]
10. Ye, S.-J.; Scribner, A.; McLendon, D.; Ijisan, O.; Chen, S.; Shao, W.; Balliet, R. Method of determining unconventional reservoir saturation with NMR logging. In Proceedings of the SPE Annual Technical Conference and Exhibition, Calgary, AB, Canada, 30 September–2 October 2019.
11. Zhu, L.; Zhang, C.; He, X.; Wu, Z.; Zhou, X.; Di, S.; Li, Y. Permeability prediction of tight sandstone reservoir based on improved BPNN and T₂ full-spectrum. *Geophys. Prospect. Pet.* **2017**, *56*, 727–734.
12. Bishop, C. *Pattern Recognition and Machine Learning*; Springer: New York, NY, USA, 2006.
13. Qiu, X. *Neural Network and Deep Learning*; Machinery Industry Press: Beijing, China, 2019.
14. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [[CrossRef](#)]
15. Tieleman, T.; Hinton, G. *Divide the Gradient by a Running Average of its Recent Magnitude*. Coursera: *Neural Networks for Machine Learning*; Technical Report; 2017; Available online: <https://www.classcentral.com/course/neuralnets-398> (accessed on 16 April 2021).
16. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
17. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [[CrossRef](#)]
18. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [[CrossRef](#)]
19. Drucker, H.; Burges, C.J.; Kaufman, L.; Smola, A.; Vapnik, V. Support vector regression machines. *Adv. Neural Inf. Process. Syst.* **1997**, *9*, 155–161.
20. Zheng, Z. *TensorFlow: Practical Google Deep Learning Framework*; Electronic Industry Press: Beijing, China, 2018.
21. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [[CrossRef](#)]
22. Li, H. *Statistical Learning Methods*; Tsinghua University Press: Beijing, China, 2012.