

Article

An Automatic Classification Method of Well Testing Plot Based on Convolutional Neural Network (CNN)

Hongyang Chu ^{1,2,†}, Xinwei Liao ^{1,2}, Peng Dong ^{1,2,†}, Zhiming Chen ^{1,2,*}, Xiaoliang Zhao ^{1,2} and Jiandong Zou ^{1,2}

¹ College of Petroleum Engineering, China University of Petroleum, Beijing 102249, China

² State Key Laboratory of Petroleum Resources and Engineering, Beijing 102249, China

* Correspondence: zhimingchn@cup.edu.cn

† These authors contributed equally to this work.

Received: 16 June 2019; Accepted: 21 July 2019; Published: 24 July 2019



Abstract: The precondition of well testing interpretation is to determine the appropriate well testing model. In numerous attempts in the past, automatic classification and identification of well testing plots have been limited to fully connected neural networks (FCNN). Compared with FCNN, the convolutional neural network (CNN) has a better performance in the domain of image recognition. Utilizing the newly proposed CNN, we develop a new automatic identification approach to evaluate the type of well testing curves. The field data in tight reservoirs such as the Ordos Basin exhibit various well test models. With those models, the corresponding well test curves are chosen as training samples. One-hot encoding, Xavier normal initialization, regularization technique, and Adam algorithm are combined to optimize the established model. The evaluation results show that the CNN has a better result when the ReLU function is used. For the learning rate and dropout rate, the optimized values respectively are 0.005 and 0.4. Meanwhile, when the number of training samples was greater than 2000, the performance of the established CNN tended to be stable. Compared with the FCNN of similar structure, the CNN is more suitable for classification of well testing plots. What is more, the practical application shows that the CNN can successfully classify 21 of the 25 cases.

Keywords: convolutional neural network; well testing; tight reservoirs; pressure derivative; automatic classification

1. Introduction

Well testing generally has two major categories: Transient rate analysis and transient pressure analysis. For the transient pressure analysis, its main purpose is to identify the type of target reservoir and further quantitatively determine the reservoir properties. Muskat [1] first proposed a method of estimating the initial reservoir pressure and parameters using a buildup test plot. Due to the fact that compressibility of the formation fluid is difficult to study, this method only can qualitatively analyze the results. Van Everdingen and Hurst [2] used the Laplace integral method to obtain the analytical solution of the transient diffusion equation, which gives the mathematical theoretical basis of well testing. Based on this truth, Horner et al. [3] developed a classic “semi-log” analysis method, which can determine the permeability, skin factor, productivity index, and other parameters. These methods make full use of the mid and late period data in well testing, but a common disadvantage is that the early data of the well testing is ignored.

In order to make reasonable use of the early data in well testing, Ramey et al. [4] first proposed a “plate analysis method” of the log–log type plot. Further, Gringarten et al. [5] extended this method to various well test models such as the dual-porosity model and fractured well model, and a combination of different parameters were used to greatly reduce the difficulty in curve classification

and interpretation, which indicated that the well testing interpretation was widely used around the world. Bourdet et al. [6] found that different types of reservoirs had distinct responses in the pressure derivative curve, so the pressure derivative curve was introduced into the “plate analysis method”. Compared to the pressure dynamic, the application of the pressure derivative curve makes the classification of reservoir types, and the overall curve fitting, easier. Therefore, the pressure derivative plot is the most critical part of the large-scale application of well testing interpretation methods.

Recently, with the advancement in machine learning technology and the vast datasets in the petroleum industry, the broad prospects of machine learning technology in the petroleum industry have gradually been proven, and it has been applied to different aspects of the petroleum industry [7–13].

Awoleke et al. [12] combined self-organizing maps, the k-means algorithm, the competitive-learning-based network (CLN), and the feed-forward neural network (FFNN) to predict the well water production in Barnett shale. The expected misclassification error was about 10% for CLN and the average prediction error was between 10% and 26% for FFNN, which depended on the quality of the training data set.

Akbilgic et al. [14] used a neural network-based model to predict the steam-to-oil ratio in oil sands reservoirs. Porosity, permeability, oil saturation, reservoir depth, and thickness characterized by well logging and core data were used as data sets for the models.

With deep neural networks (DNNs), Wang et al. [11] used production data from 2919 wells in Bakken shale reservoirs to forecast well productivity. Results show that the predicted oil production of DNNs for both six months and 18 months was acceptable and the average proppant placed per stage was the most important factor in affecting productivity.

In numerous research studies about machine learning in the petroleum industry, Al-Kaabi and Lee [15] firstly used a three-layer FCNN to determine the well test interpretation model. In their work, the pressure derivative and corresponding time were entered into the FCNN with 60 input nodes. Additionally, different well test models were exported, and the accuracy of the prediction was verified by two field examples. This meaningful work demonstrates meaningful guidelines for later work on well test plot identification by neural networks. Following that, a series of researchers [16–18] utilized a more complex network structure and data set to optimize FCNN’s recognition result of well testing curves.

Although a large number of scholars have done some meaningful research, due to the previous limitations of CPU performance and mathematical theory basis, there are also several shortcomings as follows: (a) The number of training samples and input nodes in present neural network models are relatively insufficient, which greatly restricts the generalization ability of neural network models. (b) There is no corresponding method to overcome the over-fitting and local minimum problem, which is the phenomenon that exists widely in the fitting process of neural network models. (c) Almost all the current research is limited to the FCNN, and the newly proposed CNN has not been considered in research.

Nowadays, CNN is one of the most popular methods in the field of machine learning. Compared with FCNN, the CNN has a better performance in the domain of image recognition [19–22]. Since the different forms of pressure derivative curves represent various reservoir types, flow regimes, and outer boundary properties, in this paper, an automatic classification method of well testing curves is proposed based on CNN. By summarizing the buildup test data in low permeability reservoirs, the vertically fractured well model, dual-porosity model, and radial composite model were selected as the base model, which were used to generate 2500 theoretical curves of five different types. To overcome the problem of overfitting and local minimum, the regularization technique, Adam optimization algorithm, ReLU activation function, and mini batch technique were used to optimize the established CNN. The model input nodes were 488, which ensured that the information of the curve is completely input. Further, we compared the training performance of CNN and FCNN. The analysis of confusion matrix showed that the *Score* for CNN and FCNN on the validation set were 0.91 and 0.81 respectively, which

means that the CNN had a better prediction result than FCNN. Finally, 25 buildup test curves from Ordos Basin were used to verify the generalization ability of the CNN noted above.

2. Background

The Ordos Basin is the second largest sedimentary basin in China and it contains abundant oil and gas reserves. In terms of geology, the Ordos Basin is a large-scale multicycle craton basin with simple tectonics, which is made up of the Yimeng uplift, Weibei uplift, the Western margin thrust belt, the Tianhuan depression, and the Jinxi flexure belt [23,24]. This Basin is a typical low-permeability formation with an average permeability of less than 1 mD. Except for the Chang 6 reservoir with developed horizontal bedding [25,26], the horizontal stress in most areas of the basin is greater than the vertical stress, which means that the fractures generated by hydraulic fracturing are mainly vertical fractures [27–30].

3. Theory

3.1. Concept of CNN

Traditional neural networks (like the FCNNs) use the matrix multiplication to describe the connection between input nodes and output nodes. Wherein, each individual weight of the weights matrix describes the interaction between an input unit and an output unit. For traditional neural networks, when the number of input nodes is quite large, the number of weights will also become very huge, and the training efficiency will drop drastically. To address this issue, the convolution method of reducing the number of weights is needed to reduce training costs. The two main advantages of the convolution method are weight sharing and sparse connections, which effectively improve the situation. The calculation process of convolution method is shown in Figure 1. The filter contains the weights to be optimized, and the forward propagation process of the filter is a process of calculating the output data by using the inner product matrix multiplication between the weights in the filter and the input data. In a CNN, the filter weights used by each convolutional layer (CONV layer) are the same. The sharing of filter weights can make image content unaffected by local feature and reduce the number of weights. Then, the data are convolved and output through the activation function.

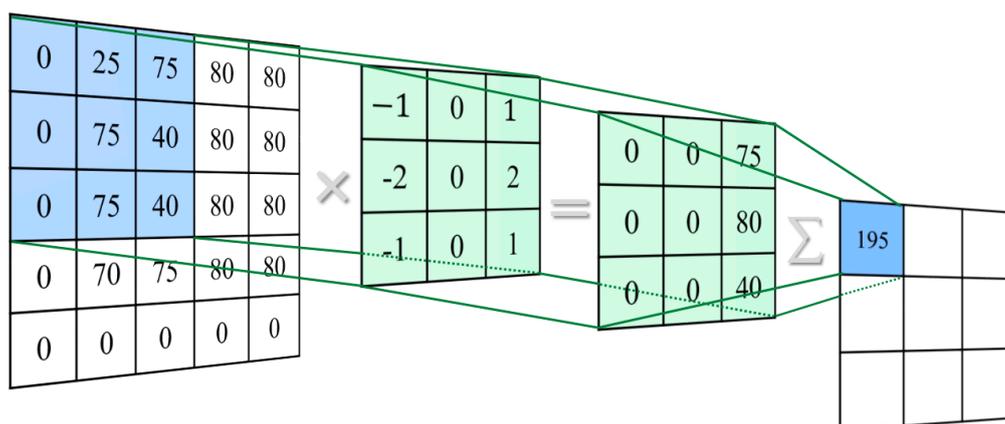


Figure 1. Schematic diagram of convolution in convolutional neural network (CNN) (the elements in the matrix represent the pixel values of the input data and weights).

In addition to the CONV layer, the network often uses a pooling layer, which can adjust the output structure of the layer and reduce the size of the model. With the pooling layer, the calculation speed and the robustness of the model are improved. The pooling layer usually includes a max-pooling layer and an average-pooling layer, as shown in Figure 2, which is used to output the maximum value and the average value in the filter area. So, no weights exist in the filter of pooling layer.

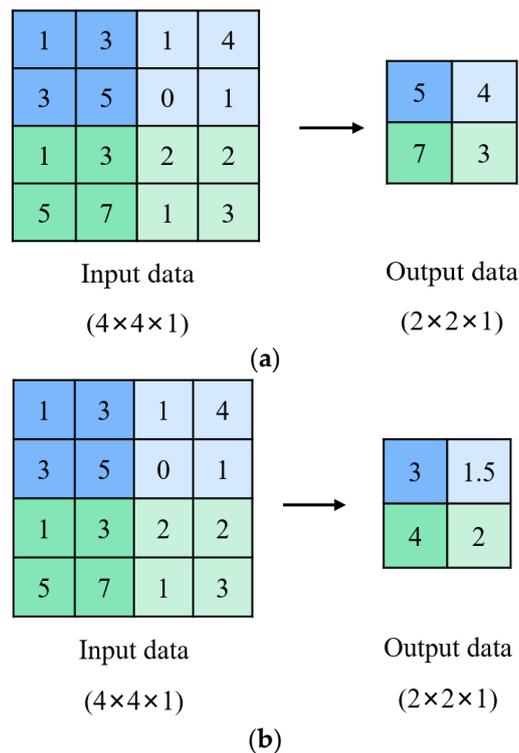


Figure 2. Schematic diagram of pooling layer calculation process (a) max-pooling layer (b) average-pooling layer (the elements in the matrix are various pixels).

To achieve different test tasks, different layers need to be connected to form a CNN. The AlexNet is a typical CNN proposed by Krizhevsky et al. [31], which has a simple model structure but accurate image recognition rate. The AlexNet fully demonstrates the superior performance of CNN in dealing with complex data. As shown in Figure 3, the structure of AlexNet is 8 layers with weights, including 5 layers of CONV layers and 3 layers of fully connected layers (FC layers). Three max-pooling layers are utilized to adjust the output shape. Additionally, to reduce the dimension of curve probabilistic prediction data, a flatten method is used before the FC layer. Finally, the FC layers are used to achieve the data dimensional reduction and output the final results. In the calculation process of FC layers, the softmax function is usually chosen to calculate the probability of the data after dimension reduction. The picture with the highest probability is the final result of CNN. Equation (1) gives the mathematical expression of the softmax function.

$$\text{softmax}(l) = \frac{e^{a_l}}{\sum_{k=1}^c e^{a_k}} \quad (1)$$

where a_l is the output value of the l th node of the output layer, c is the total number of sample classes.

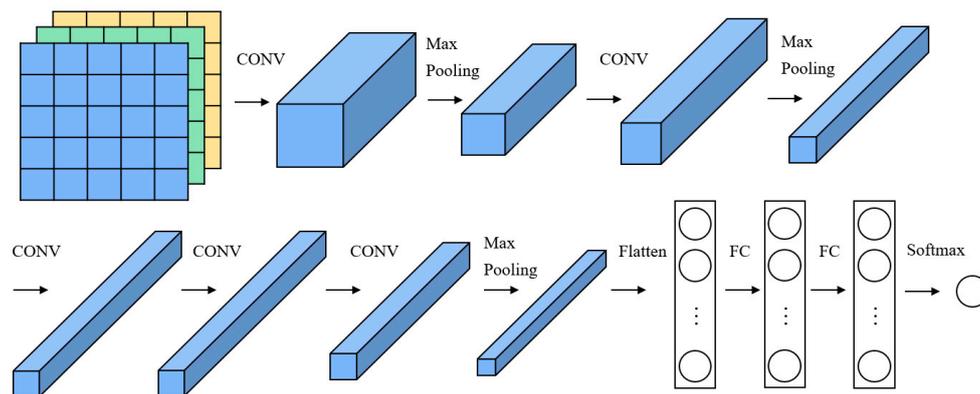


Figure 3. Schematic diagram of the AlexNet structure.

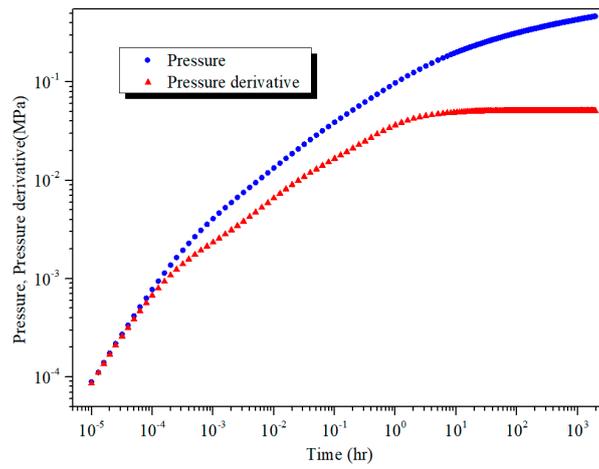
3.2. Model of CNN

3.2.1. Sample Obtaining

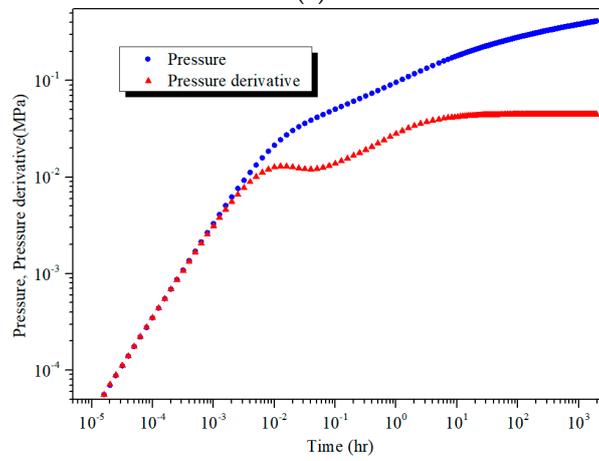
The type curve of well testing is the log–log plot, which is based on the analysis of the time, pressure, and its derivative in the log–log coordinates. The reservoir types are determined by different shapes of the curve, and they are very critical to well testing interpretation results. Due to the non-uniqueness in interpretation results, it is difficult to quickly and accurately determine the reservoir type corresponding to a large amount of interpretation data. Automatic identification of well test curve types based on CNN can significantly reduce the workload of identification, and it provides a reliable basis for accurate parameter inversion.

Production wells in unconventional reservoirs represented by the Ordos Basin are generally hydraulically fractured, so the vertically fractured model is one of the commonly used well test interpretation models in unconventional reservoirs. At the same time, due to hydraulic fracturing, the natural fractures in the formation are activated, and the resulting considerable amounts of buildup test data of Ordos basins are characterized by a dual-porosity model. On the other hand, large-scale hydraulic fracturing significantly improves the permeability of the near-well region, which means that the radial composite model is also used as the reservoir model for well test interpretation in unconventional reservoirs. The mathematic expressions of the above well testing models are given in the Appendices B–D. With these mathematic expressions, Figure 4 shows that the typical well test curves for the above models can be roughly divided into the following categories. In the same reservoir conditions, there is no doubt that the radial composite model with mobility ratio >1 and dispersion ratio >1 in the five well test models has the greatest productivity. The reason is that this model assumes that the area around the production well has been adequately stimulated by the hydraulic fracturing, so an inner zone of high permeability is formed around the production well, which contributes to the largest productivity.

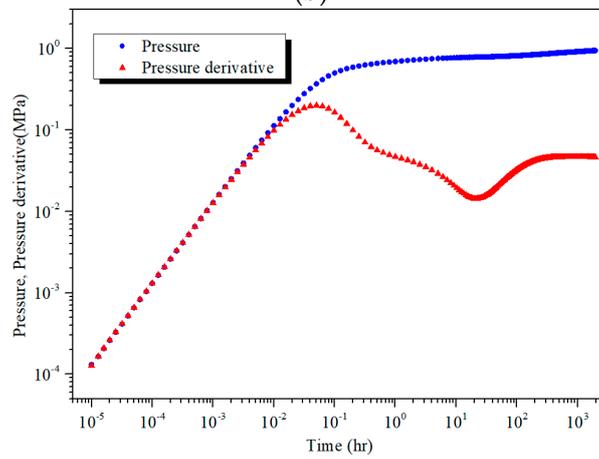
In this paper, the training set included 2725 well test curves for five well test models, and 25 field buildup test cases were used to evaluate the generalization ability of CNN. The pressure derivative-time curve data for each training sample was used for classification. There were 545 curves of each well test model type and Table 1 shows the range of corresponding parameters for five well test models.



(a)



(b)



(c)

Figure 4. Cont.

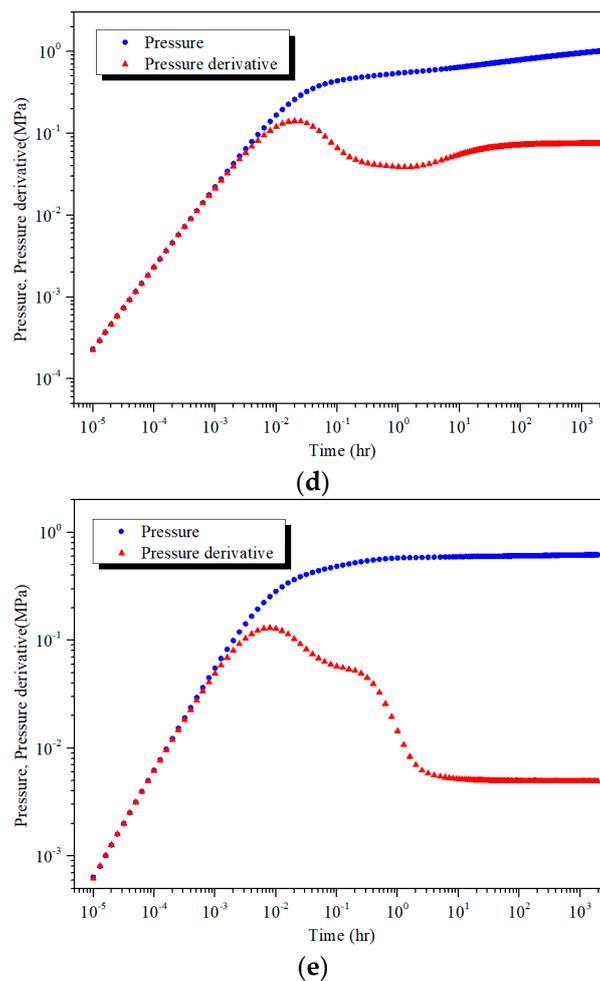


Figure 4. The typical well test curves for models used in this work. (a) Infinite-conductivity vertically fractured model without skin factor (Model 1); (b) infinite-conductivity vertically fractured model with skin factor (Model 2); (c) dual-porosity model with pseudo-steady state (Model 3); (d) radial composite model with mobility ratio >1 and dispersion ratio >1 (Model 4); (e) radial composite model with mobility ratio <1 and dispersion ratio >1 (Model 5).

Table 1. The range of model parameters of various well test models in this paper.

Model	1	2	3	4	5
Wellbore storage coefficient (m^3/MPa)	0–0.25	0–0.25	0–0.25	0–0.25	0–0.25
Skin factor	0–0.05	0.05–2	0–1	0–1	0–1
Fracture half length (m)	20–80	20–80	/	/	/
Initial pressure (MPa)	15–35	15–35	15–35	15–35	15–35
Permeability (mD)	0.10–50	0.10–50	0.10–50	0.10–50	0.10–50
Thickness (m)	9.14	9.14	9.14	9.14	9.14
Porosity	0.10	0.10	0.10	0.10	0.10
Omega	/	/	0.01–0.60	/	/
lambda	/	/	10^{-6} – 10^{-9}	/	/
Mobility ratio	/	/	/	1–20	0–1
Dispersion ratio	/	/	/	1–20	0–1
Composite radius (m)	/	/	/	10–200	10–200

Before training, improving, and evaluating the CNN model for well test plots, it was necessary to divide the training data sets into training set, validation set, and test set. Their quantities respectively accounted for 90.909%, 8.182%, and 0.909% of the total number of samples. The primary role of the validation set was to compare the performance of different neural network models. The test set was used to verify the generalization ability of the model based on the mine data. The validation set and test set were not involved in the training process of the network. The first time they were entered into the network was in the process of network verification. In total, 2500 of the theoretical curves were determined as training sets, and the remaining 225 curves were chosen as the validation set. Additionally, 25 field buildup test cases from the Ordos Basin were used as a test set. Figure 5 is a schematic diagram of training data partition.

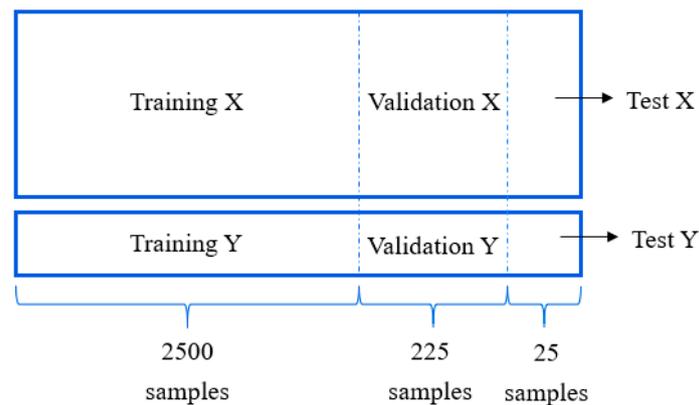


Figure 5. Data partition including training set, validation set, and test set.

3.2.2. Structure of Neural Network Model

The neural network model has strong ability of nonlinear representation, as its basic unit is a neuron. Through the design of different numbers of neurons and different layers, various mapping relations can be characterized.

Model Building of CNN

The CNN constructed in this paper was a five-layer deep network with weights, in which three layers were CONV layers and two layers were FC layers, as shown in the Figure 6. There was also the max-pooling layer and average-pooling layer between various CONV layers, which was used to compress the input data and reduce overfitting problem. Table 2 shows the number of network weights in the different layers and the total number of weights was 76,583. In order to minimize the weights number of CNN, the input layer of the CNN was the data point of pressure derivative time plot, rather than the curve picture. Since the input data point of pressure derivative time curve was one-dimensional data with respect to time, we used the layers containing one-dimensional (1D) filter, including CONV1D, max-pooling1D. Layers containing two-dimensional (2D) filters (such as CONV2D, max-pooling2D, and average-pooling2D) were used to transform 1D data into 2D data needed for convolutional calculations. In the final layer, flatten method and softmax activation function were used to output the result of CNN.

Table 2. The layer shape and weights number of CNN.

Layer	Layer Shape (Output Shape)	Weights Number
Input	(2,244)	0
Conv1D	(38,80)	418
Max-Pooling1D	(38,38)	0
Conv2D	(17,17,64)	1664
Max-Pooling2D	(5,5,64)	0
Conv2D	(2,2,128)	73856
Average-Pooling2D	(1,1,128)	0
Flatten	128	0
FC (Output)	5	645

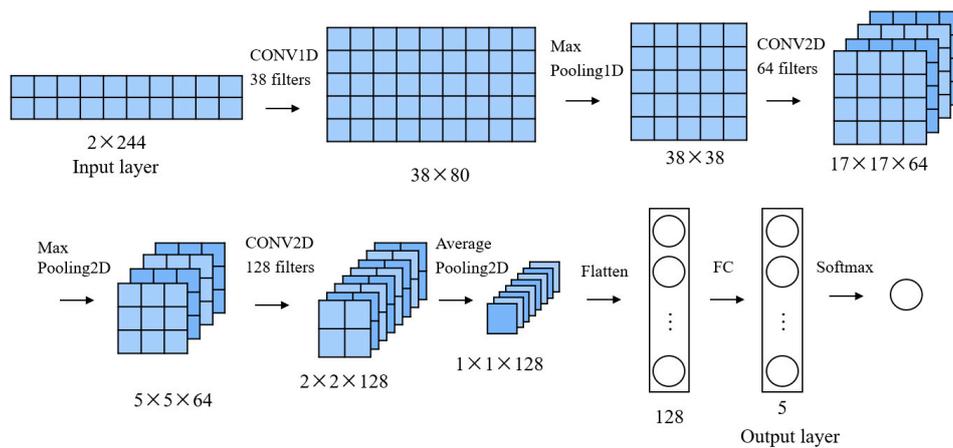


Figure 6. Schematic diagram of CNN structure.

Model Building of FCNN

In contrast, a FCNN was established, which had a similar number of weights as CNN. The input layer, hidden layer, and output layer for FCNN had 488, 106, and 5 neurons respectively. Figure 7 shows that the input layer consisted of 488 nodes that accepted the 244 data points (t, dp). Table 3 demonstrates that the FCNN had a total of 76,575 weights.

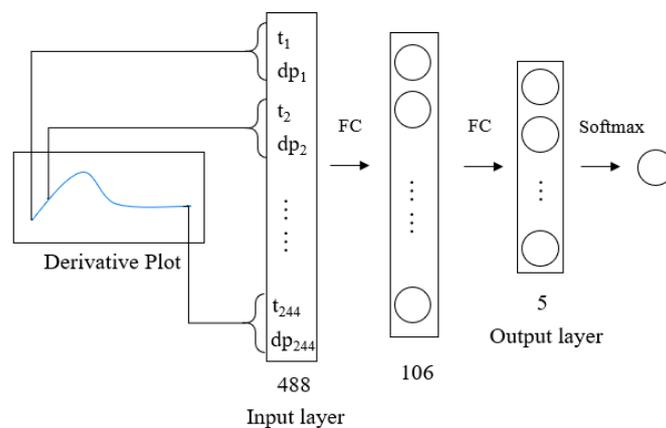


Figure 7. Schematic diagram of fully connected neural networks (FCNN) structure.

Table 3. The layer shape and weights number of FCNN.

Layer	Layer Shape (Output Shape)	Weights Number
Input	488	0
FC	106	75,795
FC (Output)	5	780

Evaluation Results for the CNN and FCNN

During the training process of the FCNN and CNN, the maximum value of the output corresponded to the type of curves being predicted, which was recorded as \hat{y} . In order to optimize the weights in models, the cross-entropy function values of the predicted and the theoretical curve type were calculated. As shown in Equation (2), the cross-entropy function is usually recorded as the loss function L . When the loss function value is the smallest, the ratio of the number of accurately predicted training samples to the total number of samples (called accuracy) is the largest, which means that the network model has the highest performance.

$$L(\hat{y}, y) = -\sum_{i=1}^m y_i \log \hat{y}_i \tag{2}$$

where y_i the type of the i th training samples, \hat{y}_i is the probability of i th training samples, and m is the number of training samples. In order to obtain a robust and fast CNN, the newly proposed one-hot encoding, Xavier normal initialized model, ReLU activation function, L2 regularization method, Adam optimization algorithm, and mini batch technique were combined to further construct the CNN.

3.2.3. One-Hot Encoding

In the training tasks of machine learning, the variety of sources of training data led to more complex data types. The training data can be roughly divided into type data and numerical data. The training process of the neural network model was performed on the basis of numerical data. Therefore, in the classification task, the type data needed to be converted into numerical data and were further used to train the neural network model. The one-hot encoding method is a commonly used method of encoding type data into numerical data, which encodes the type data into a binary vector with at most one valid value. As shown in Table 4, each column represents each category in the training sample data and the unit containing “1” represents the category to which the sample data belongs.

Table 4. The schematic diagram of one-hot encoding.

	Class1	Class2	Class3	Class4	Class5
Sample1	0	1	0	0	0
Sample2	1	0	0	0	0
Sample3	0	0	1	0	0
				
Sample2724	0	0	1	0	0
Sample2725	0	0	0	0	1

3.2.4. Determination of Model Initialization

During the training process of the neural network model, proper initialization of the weights was essential to establish a robust neural network model. The proper initialization of the weights will cause the weights to be distributed over a reasonable range. If the initial weight value is small, the effective information in the backpropagation process will be ignored and the training process of neural network model maybe invalidated. When the initial weight values are large, the weight fluctuations in the backpropagation process will increase, which may lead to the instability or even

collapse of the model. The commonly used initialization methods of neural network model include four categories and they are as follows: (1) Randomnormal method; (2) Randomuniform; (3) Xavier normal; (4) Xavier uniform [32]. In this work, we compared the effects of four initialization methods on the training results. After 100 iterations of the model, the Xavier normal initialized model had the highest accuracy in the training set and the validation set (Figure 8).

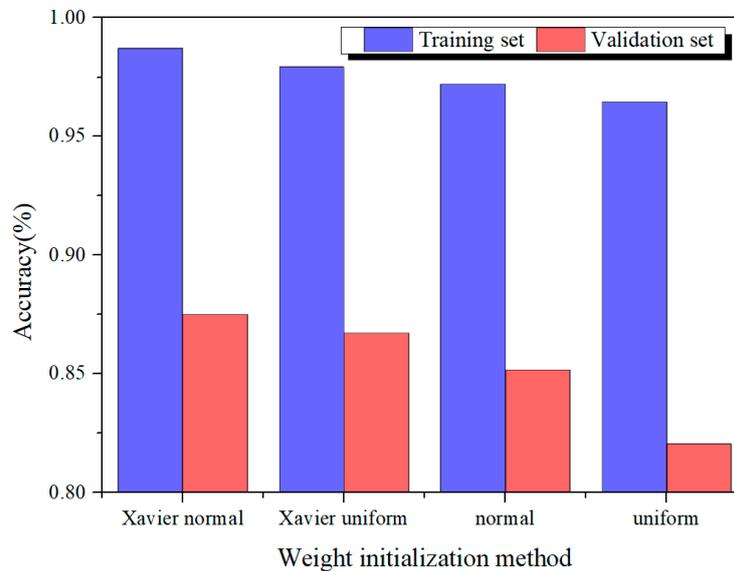


Figure 8. Comparison results of CNN on training set and validation set under different initialization methods.

3.2.5. Selection of Activation Functions

The activation function of the neural network has a significant impact on the prediction effect of the model. When no activation function is used (i.e., $f(x) = x$), the input of each node is a linear function of the output result of the node in upper layer. In this case, regardless of the number of layers in the neural network, the output result only is a linear combination of input results, and the hidden layer does not work. Only when the neural network model uses a nonlinear activation function are the output results no longer a linear combination of input results and it can approximate an arbitrary function. Table 5 shows the five commonly used activation functions (i.e., linear, tanh, sigmoid, ELU, and ReLU). As shown in Figure 9, the comparative results showed that the neural network model had a better effect when the ReLU function was used in the middle layer.

Table 5. The mathematical expression of five commonly activation functions.

Type	Equation
linear	$f(x) = x$
tanh	$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
sigmoid	$f(x) = \frac{1}{1 + e^x}$
ELU	$f(x) = \begin{cases} e^x - 1, & x < 0 \\ x, & x \geq 0 \end{cases}$
ReLU	$f(x) = \max(0, x)$

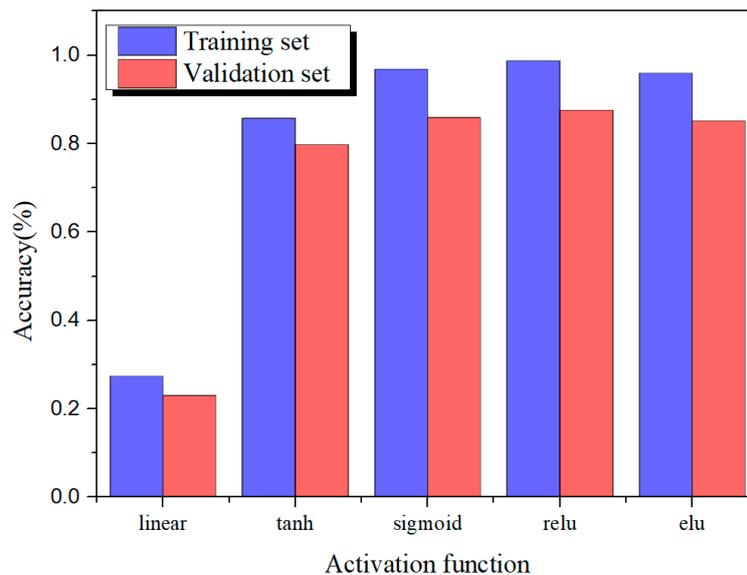


Figure 9. Comparison results of CNN on training set and validation set under different activation functions.

3.2.6. Regularization Technique

The overfitting is a common problem in the training process of neural network models and it greatly reduces the generalization ability of neural network models [33]. The main reasons for the overfitting problem are insufficient training samples and a complex structure of networks. To overcome this problem, the dropout method and L2 regularization method are used to dynamically adjust the network structure, which can effectively avoid the overfitting problem. (1) In the process of forward propagation, the dropout method can make various nodes stop working in a certain probability p (called dropout rate) and the relative importance of each node is balanced. After the introduction of the dropout method, each node of the neural network model contributes more equally to the output results and avoids the situation where a few high-weight nodes fully control the output results. (2) For the L2 regularization method, the sum of the squared value for weight squares is added to the loss function, which can constrain the size of the weight values to reduce complexity of the model. Therefore, Equation (2) is rewritten as Equation (3).

$$L(\hat{y}, y) = -\sum_{i=1}^m y_i \log \hat{y}_i + \lambda \sum_{j=1}^n w_j^2 \quad (3)$$

where λ is the super-parameter, which is used to control the level of weight decay. n is the amount of weights. In contrast, the results of the classification accuracy of the model with dropout method, L2 regularization method, and without regularization method are compared. It can be seen from Figure 10 that the model had the highest accuracy in the validation set when using the dropout method and the accuracy in validation set was close to the that for the training set.

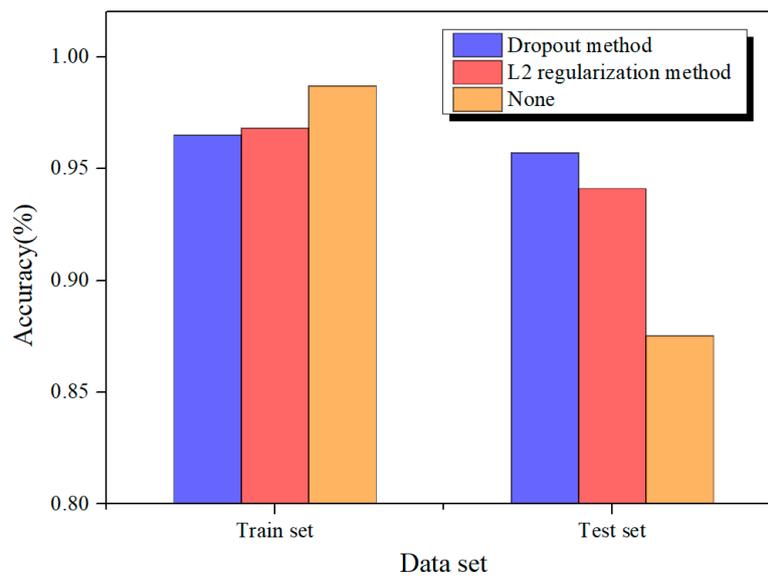


Figure 10. Comparison results of CNN on training set and validation set under different regularization techniques.

3.2.7. Adam Optimization Algorithm

To obtain the minimum value of loss function of the model, the weights in the network model need to be updated at each iteration step. Among various optimization algorithms for weight updating, the Adam optimization algorithm proposed by Kingma and Ba, [34] has the highest performance [35,36]. Compared to the classical gradient descent method, this method can avoid the oscillation of the loss function and the model with Adam optimization algorithm has a higher convergence speed. The Adam optimization algorithm updates the network model weights in the form of Equation (4).

$$w_j^t = w_j^{t-1} - \frac{\eta^t}{\sqrt{v_j^t + \varepsilon}} \omega_j^t \quad (4)$$

where η^t is the learning rate at the t th time step, w_j^t is the network weight of the j th feature of the training sample data under the t th time step, ε is a small constant to avoid a zero denominator.

In Equation (5),

$$\omega_j^t = \frac{\beta_1 \omega_j^{t-1} + (1 - \beta_1) g_j^t}{1 - (\beta_1)^t} \quad (5)$$

$$v_j^t = \frac{\beta_2 v_j^{t-1} + (1 - \beta_2) (g_j^t)^2}{1 - (\beta_2)^2} \quad (6)$$

where β_1 and β_2 are the exponential decay rates for the moment estimates, g_j^t is the gradient in the j th parameter under the t th time step. In this work, we used the parameters recommended by Kingma and Ba, [34]: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\omega_j^0 = 0$, $v_j^0 = 0$ and $\varepsilon = 10^{-8}$.

3.2.8. Mini Batch Technique

The premise of machine learning is that it requires a huge sample size. During each iteration of the model, the optimization algorithm needs to fit the model to all training samples at once, so the requirement for CPU will be enormous. In order to reduce the requirements for CPU and improve computational efficiency, the mini batch technique was utilized as it can randomly select a small portion of the training samples in the training set for each iteration process of the model. Meanwhile,

the random selection of training samples also made the mini batch technology effectively avoid the neural network model falling into local minimum problems during the training process. For the mini batch technique, the gradient g^t in Equations (5) and (6) is as follows:

$$g^t = \frac{1}{b} \sum_{k=1}^b g_k^t \quad (7)$$

$$g_k^t = \frac{1}{s} \sum_{r=1}^s \nabla L(w^{t-1}; x_{i_r}; y_{i_r}) \quad (8)$$

$$b = \left\lceil \frac{m}{s} \right\rceil \quad (9)$$

where b represents the number of iterative step of the mini batch method from $t - 1$ th time step to t th time step. g_k^t is the gradient of the k th iterative step from $t-1$ th time step to t th time step. w^{t-1} is the weights at the $t-1$ th time step. s is the number of training samples in one mini batch. i_1, \dots, i_s are a random number between 1 and m . x_{i_r} is the pressure derivative-time curve data of the i_r training sample. y_{i_r} the type of the i_r training sample. m is the total number of training samples.

4. Results and Discussions

4.1. Comparison of Classification Performance for FCNN and CNN

We used the same techniques (including regularization technique, Adam optimization algorithm, activation functions, and initialization methods) to optimize the FCNN model. Table 6 and Figure 11 compare the errors of different models. The error of the test set verified the performance of the well test plot classification in the field buildup test cases and demonstrated the generalization ability of the CNN. For the FCNN, after 100 iterations, the loss function value was 0.44, and the classification accuracy was 91.2%. In the validation set, its accuracy was 89.8%. For the CNN, the loss function and accuracy for training set and validation set respectively were 0.19, 96.6%, and 95.6%. The comparing results of FCNN and CNN showed that the CNN had a higher accuracy when the number of weights of two models were close (76,583 weights and 76,575 weights).

As shown in Figures 12–14, the confusion matrix analysis is a method for judging the classification performance of neural network model, which shows the accuracy result of classification. Mukhanov et al. [37] used the confusion matrix to evaluate the classification result of the waterlogging curve by the support vector machine technology. The confusion matrix separately calculates the number of misclassification classes and the number of correct classification classes in the model. It can be seen from Figures 12 and 13, and Tables 7 and 8 that the FCNN had different classification capabilities for various types of well test curves in the training set and validation set. For the CNN, its classification results of the 5 well test models were 0.98, 0.94, 0.97, 0.95, 0.98 for training set and 0.97, 0.93, 0.96, 0.93, 0.98 for validation set, which were generally better than the FCNN results. Figure 13 and Table 8 also showed that the FCNN forecasting results of class1, class3, and class5 in the validation set were basically correct, but there was a large error for the curves of class2 and class4. For CNN, the stability of forecasting result was high, and the prediction errors of various types of curves were almost the same, indicating the reliability of CNN. Through the confusion matrix, the recall rate (Equation (11)) and precision rate (Equation (10)) of the model could be calculated. The precision rate represents the number of correctly predicted samples in a class (TP) to all actually retrieved items (the sum of TP and FP). The recall rate refers to the TP as a percentage of all items (the sum of TP and FN) that should be retrieved. F_1 value is the harmonic mean of the precision rate and recall rate. The average of the F_1 values for all training samples is determined as $Score$ (Equation (13)). Table 8 summarizes the performance of different network models in validation sets. It can be seen that the $Score$ of the FCNN

model was 0.81 and the value of CNN was 0.91, indicating that the overall performance of CNN was better than FCNN.

$$Precision = \frac{TP}{TP + FP} \quad (10)$$

$$Recall = \frac{TP}{TP + FN} \quad (11)$$

$$F_1 = 2 \frac{Precision \times Recall}{Precision + Recall} \quad (12)$$

$$Score = \left[\frac{1}{c} \sum_{k=1}^c (F_1)_k \right]^2 \quad (13)$$

where TP is the number of correctly predicted samples in a class. For a certain type of training sample, FN is the difference between the number of successfully predicted training samples and the total number of training samples, FP is the difference between the number of successfully predicted training samples and the predicted number of samples for a certain type, c is the total number of sample classes.

Finally, the classification ability of CNN was verified on 25 field buildup test data, among which 21 samples are successfully classified. Table 9 and Figure 14 show the confusion matrix of the model in the test set, and its $Score$ was 0.69. Appendix A shows the data of the 25 field buildup test data.

Table 6. Model prediction accuracy.

	Loss Function	Accuracy (%)
CNN train set	0.19	96.6
CNN validation set	/	95.6
FCNN train set	0.44	91.2
FCNN validation set	/	89.8

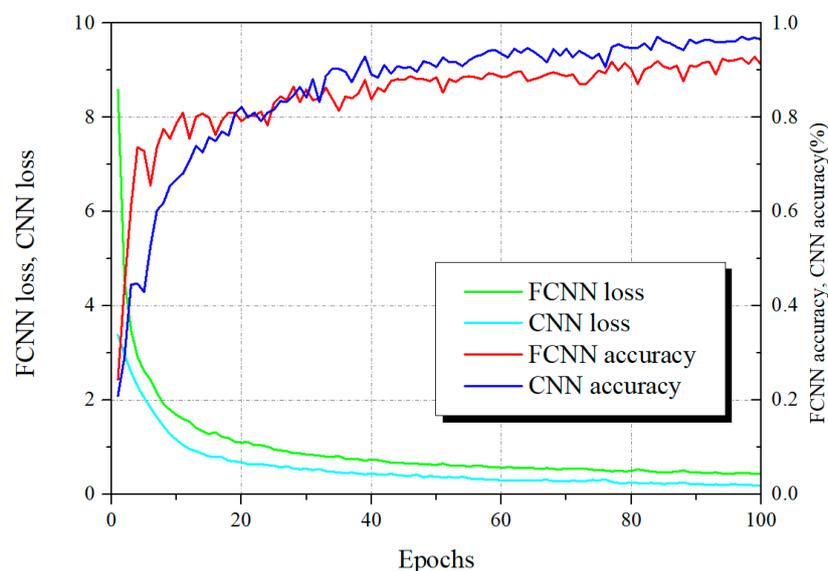


Figure 11. The changes in the accuracy and loss function curve for FCNN and CNN on training set as the number of iterations increases.

CNN		Predict				
		1	2	3	4	5
Ture	1	495	0	5	0	0
	2	4	453	12	25	6
	3	10	0	489	0	1
	4	0	13	0	483	4
	5	0	1	0	4	495

(a)

FCNN		Predict				
		1	2	3	4	5
Ture	1	461	13	26	0	0
	2	9	411	24	41	15
	3	26	17	456	0	1
	4	0	25	8	463	4
	5	0	8	0	3	489

(b)

Figure 12. The confusion matrix of CNN and FCNN on training set. (a) CNN confusion matrix; (b) FCNN confusion matrix.

CNN		Predict				
		1	2	3	4	5
Ture	1	43	2	0	0	0
	2	0	41	0	4	0
	3	0	0	43	0	2
	4	0	0	2	43	0
	5	0	0	0	0	45

(a)

FCNN		Predict				
		1	2	3	4	5
Ture	1	39	5	0	1	0
	2	1	33	0	11	0
	3	0	0	45	0	0
	4	0	4	1	40	0
	5	0	0	0	0	45

(b)

Figure 13. The confusion matrix of CNN and FCNN on validation set. (a) CNN confusion matrix; (b) FCNN confusion matrix.

CNN		Predict				
		1	2	3	4	5
Ture	1	3	1	0	0	0
	2	0	5	0	1	0
	3	0	0	2	0	0
	4	0	1	0	4	0
	5	0	0	1	0	7

Figure 14. The confusion matrix of CNN on test set.

Table 7. The evaluation result of FCNN and CNN on training set.

Model	Index	Class1	Class2	Class3	Class4	Class5	Score
FCNN	Precision (%)	92.94	86.71	88.72	91.32	96.07	0.83
	Recall (%)	92.20	82.20	91.20	92.60	97.80	
	F ₁ Score	0.93	0.84	0.90	0.92	0.97	
CNN	Precision (%)	97.25	97.00	96.64	94.34	97.83	0.93
	Recall (%)	99.00	90.60	97.80	96.60	99.00	
	F ₁ Score	0.98	0.94	0.97	0.95	0.98	

Table 8. The evaluation result of FCNN and CNN on validation set.

Model	Index	Class1	Class2	Class3	Class4	Class5	Score
FCNN	Precision (%)	97.50	78.57	97.83	76.92	100	0.81
	Recall (%)	86.67	73.33	100	88.89	100	
	F ₁ Score	0.92	0.76	0.99	0.82	1.00	
CNN	Precision (%)	100	95.35	95.56	91.49	95.74	0.91
	Recall (%)	95.56	91.11	95.56	95.56	100	
	F ₁ Score	0.97	0.93	0.96	0.93	0.98	

Table 9. The evaluation result of test set for CNN.

Index	Model 1	Model 2	Model 3	Model 4	Model 5	Score
Recall	75	83.3	100	80	87.5	0.69
Precision	100	71.4	66.7	80	100	
F ₁ Score	0.86	0.77	0.80	0.80	0.93	

4.2. Effects of Parameters on Classification Results

Sensitivity analysis is a key step in testing CNN performance and determining the impact of input parameters on the predictive results [38–43]. In order to study the influence of a series of CNN parameters on the prediction results and further optimize the established CNN, a sensitivity analysis was conducted.

4.2.1. Effect of the Learning Rate

The loss function is a function of weights, and the learning rate determines the update speed of the weights in the CNN and determines the value of the loss function. If the learning rate is too large, it will cause the loss function to oscillate and the CNN is hard to converge. When the learning rate is a small value, the updated value of the weights also is small and the model converges slowly. Therefore, there is an optimized learning rate for each CNN. In order to determine the optimal learning rate, we changed the value of the learning rate from 0.0001 to 0.03 and remaining values were constant. Figure 15 shows that the CNN had the highest accuracy on both the validation set and the training set when the learning rate was 0.005.

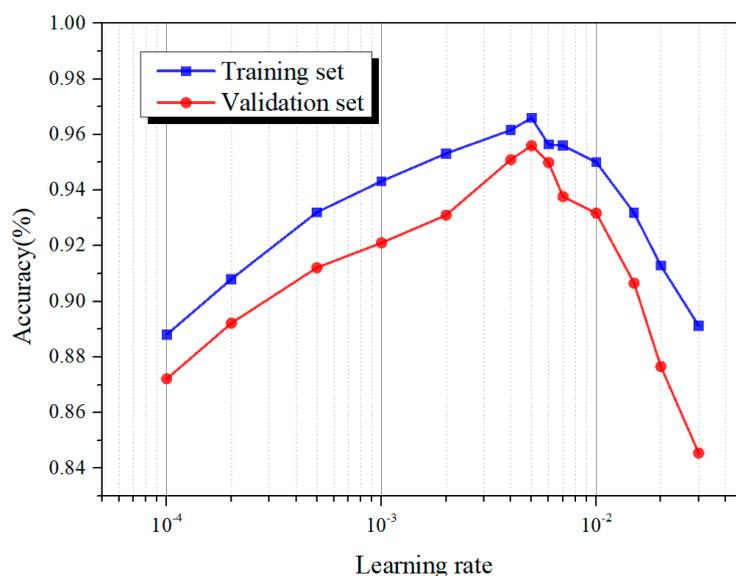


Figure 15. Comparison of the training results of CNN on training set and validation set under different learning rates.

4.2.2. Effect of the Dropout Rate

The key point of the dropout method to prevent the overfitting problem is to make some nodes of the CNN to stop working in a probability of dropout rate. Therefore, the value of the dropout rate has a significant impact on the training effect of the CNN. As shown in Figure 16, as the dropout rate increased, the accuracy of CNN on the training set continued to decrease. For the value for the validation set, it increased firstly and then decreased when the dropout rate increased. Meanwhile, the accuracy difference of the training set and the validation set was large in the case of small dropout value, indicating that the overfitting phenomenon had occurred. As the dropout rate became larger, the accuracy difference was very small, meaning that the CNN had not been well fitted to the training data. For the CNN in this paper, the optimal value of the dropout rate was 0.4.

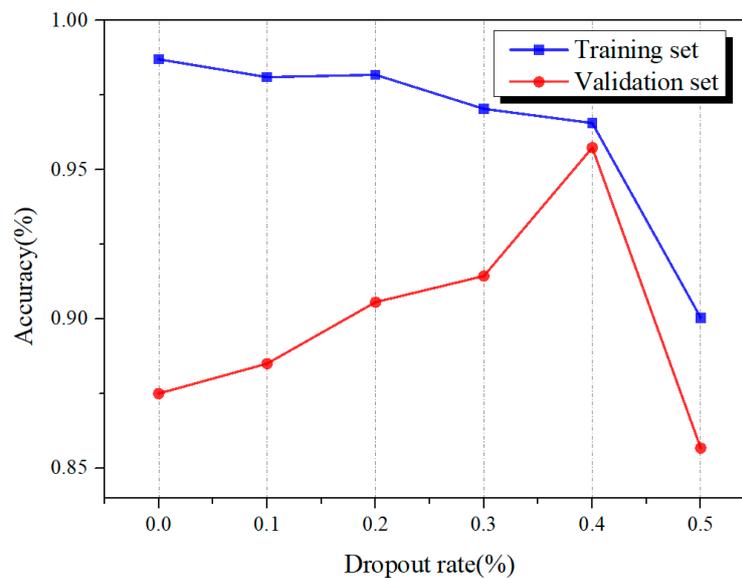


Figure 16. Comparison of the training results of CNN on training set and validation set under different dropout rates.

4.2.3. Effect of the Number of Training Samples

The performance of CNN is strongly controlled by the number of training samples. A small sample size can make the training process of the CNN difficult to converge. In general, CNN require a pretty large training sample size and the negative effect is that this large sample size usually increases the requirement for CPU. To select as few samples as possible while ensuring the best performance of CNN, the impact of sample size on CNN learning curves was investigated. Figure 17 shows that as the number of training samples increased, the accuracy of CNN model on the validation set also increased. When the number of training samples was greater than 2000, the increase of the accuracy on the validation set tended to be flat. Meanwhile, the CNN had a similar accuracy on both the training set and the validation set. Therefore, the number of training samples was finally determined to be 2500.

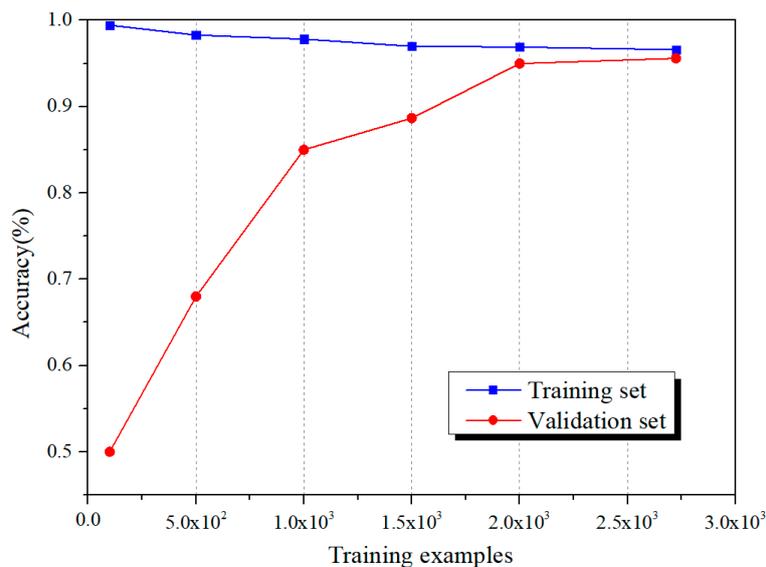


Figure 17. Comparison of the training results of CNN on training set and validation set under different numbers of training samples.

5. Conclusions

In this paper, a CNN model was developed to classify the well testing curves. In order to obtain the best test curve classification effect, before the training, we optimized the CNN model from several aspects such as regularization technology, activation function, and optimization algorithm. The results show that the Xavier normal initialization worked best in the four optimization methods. Among the five activation functions, the CNN model had the best performance when the activation function of the convolution layer and the output layer was chosen as the ReLU function. Compared to the L2 regularization method, the dropout method had a better performance in avoiding overfitting problem. In addition, the utilization of mini batch technique and Adam optimization algorithm made the model not fall into local minimum and fast convergence. Further, the impacts of key parameters in the CNN model on work performance were studied. It was found that when the learning rate was 0.005, the CNN had the highest precision in the validation set and the training set. For the dropout rate, CNN could better fit the training data without over-fitting phenomenon in the case of 0.4. The analysis of training sample numbers showed that the accuracy difference between the training set and the validation set could be ignored when the number of training samples was 2500. Finally, the classification results of CNN and FCNN with similar structures on well testing curves were compared. For the validation set, the *Score* of FCNN and CNN were 0.81 and 0.91, respectively, indicating that the CNN had a more robust performance in the classification results of the well test curve. The 25 field cases from the Ordos Basin showed that the trained CNN could successfully classify 21 cases and the robustness of the model was further proved.

Author Contributions: Every author has contributed to this work. Conceptualization, C.H. and D.P.; Methodology, C.H.; Software, D.P.; Validation, C.H., D.P. and L.X.; Formal Analysis, L.X.; Investigation, L.X.; Resources, L.X. and C.Z.; Data Curation, L.X.; Writing-Original Draft Preparation, C.H.; Writing-Review & Editing, C.H., D.P. and C.Z.; Visualization, Z.J.; Supervision, Z.X.; Project Administration, L.X.; Funding Acquisition, L.X., Z.X. and C.Z.

Funding: This research was funded by the Joint Funds of the National Natural Science Foundation of China (Grant No. U1762210), National Science and Technology Major Project of China (Grant No.2017ZX05009004-005), Science Foundation of China University of Petroleum, Beijing (Grant No. 2462018YJRC032), and Post-doctoral Program for Innovation Talents (BX20180380).

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, and in the decision to publish the results.

Nomenclature

<i>CNN</i>	Convolutional Neural Network
<i>FCNN</i>	Fully Connected Neural Network
<i>CONVlayer</i>	Convolutional Layer
<i>FC layer</i>	Fully Connected Layer
<i>1D</i>	One Dimensional
<i>2D</i>	Two Dimensional
<i>TP</i>	True Positive
<i>FP</i>	False Positive
<i>FN</i>	False Negative
<i>w</i>	Network Weight
<i>g</i>	Gradient
<i>n</i>	Number of Network Weight
<i>b</i>	Number of Iterative Step in Mini Batch Technique
<i>s</i>	Number of Training Samples in Mini Batch Technique
<i>c</i>	Number of Sample Classes
<i>x</i>	Sample Matrix
<i>y</i>	Real Sample Label Matrix
\hat{y}	Predictive Sample Label Matrix
<i>a</i>	Output Value of Neural Network
<i>m</i>	Number of Training Samples

Greek

η	Learning Rate
β_1, β_2	Exponential Decay Rates in Adam Algorithm
ω, v	Momentum in Adam Algorithm
ε	Constant
λ	L2 Regularization Parameter

Subscript

<i>i</i>	<i>i</i> -th Sample
<i>j</i>	<i>j</i> -th Feature
<i>k</i>	Iteration

Superscript

<i>t</i>	<i>t</i> -th Time Step
----------	------------------------

Appendix A. Field Cases Used in This Work

Table A1. Parameter distribution of field cases used in this work.

	Thickness (m)	Porosity (%)	Permeability (mD)	Initial Pressure (MPa)	Wellbore Storage Coefficient	Skin Factor	Composite Radius (m)	Mobility Ratio	Dispersion Ratio	Fracture Half Length (m)	Omega	Lambda	Curve Type
Case1	9.4	10.94	0.82	15.06	0.19	0.05	/	/	/	23.1	/	/	1
Case2	9.56	13.12	0.02	13.56	0.12	-5.88	/	/	/	59	/	/	1
Case3	13.12	9.08	7.53	14.02	2.12	0.02	/	/	/	112	/	/	1
Case4	5.68	11.04	0.5	17.2	0.01	0.01	/	/	/	76	/	/	1
Case5	16	11.61	0.2	14.28	0.03	0.63	/	/	/	11.1	/	/	2
Case6	4.3	13.68	0.41	10.29	0.6	0.11	/	/	/	46	/	/	2
Case7	13.7	12.56	0.09	26.82	0.07	0.18	/	/	/	16.4	/	/	2
Case8	9.7	10.8	1.14	20.05	1.96	0.26	/	/	/	128	/	/	2
Case9	13.3	11.28	0.17	12.29	0.33	0.21	/	/	/	56.5	/	/	2
Case10	8.5	11.77	1.08	22.61	0.12	0.3	/	/	/	126	/	/	2
Case11	9.67	10.13	1	19.13	0.01	0.02	/	/	/	/	0.29	1.4×10^{-8}	3
Case12	10.78	13.03	0.27	40.55	0.07	-0.81	/	/	/	/	0.08	9.4×10^{-4}	3
Case13	13.2	11.1	0.84	17.08	0.24	-3.77	40.6	7.36	3.61	/	/	/	4
Case14	5.4	12.3	0.34	13.47	0.16	-4.58	12.3	7.71	12.8	/	/	/	4
Case15	16.3	10.95	0.13	12.41	0.14	-3.54	31	2.32	1.4	/	/	/	4
Case16	11.6	13.12	0.25	16.02	0.15	-3.64	31	2	3.06	/	/	/	4
Case17	11.2	9.63	0.78	13.24	0.14	-2.23	13.3	9.56	8.03	/	/	/	4
Case18	8.2	11.25	0.25	20.87	0.15	-2.82	33.26	0.82	0	/	/	/	5
Case19	9.2	14.06	0.91	18.52	0.66	-1.37	52.1	0.44	0	/	/	/	5
Case20	27.2	13.12	0.4	24.64	0.32	-1.72	13.9	0.36	0	/	/	/	5
Case21	9.7	10.8	1.14	20.05	1.96	0.26	94.3	0.75	0.1	/	/	/	5
Case22	8.2	13.36	0.42	25.82	0.89	-3.78	59	0.51	0	/	/	/	5
Case23	11.2	10.31	0.1	23.21	0.26	-3.11	41	0.03	0	/	/	/	5
Case24	7.3	14.6	1.35	24.68	0.61	-3.56	92.2	0.72	0.01	/	/	/	5
Case25	7.6	12.74	0.66	23.66	1.13	-3.45	18.9	0.98	0	/	/	/	5

Appendix B. Infinite-Conductivity Vertically Fractured Model

At first, a series of dimensionless variables need to be defined:

$$p_{wD} = \frac{kh\Delta p}{1.842 \times 10^{-3} q\mu B} \quad (\text{A1})$$

$$t_D = \frac{3.6kt}{\varphi\mu C_t L^2} \quad (\text{A2})$$

$$C_D = \frac{0.1592C}{\varphi h C_t L^2} \quad (\text{A3})$$

$$y_D = \frac{y}{L} \quad (\text{A4})$$

$$x_D = \frac{x}{L} \quad (\text{A5})$$

$$r_D = \frac{r}{L} \quad (\text{A6})$$

where k is the permeability, φ is the porosity, C is the wellbore storage coefficient, L is the reference length, C_t is the compressibility, t refers to time, μ refers to the viscosity, B is the volume factor, q refers to the flux rate, p is the pressure, h is the formation thickness. After dimensionless treatment, the diffusion equation in Laplace domain can be expressed as:

$$\frac{d^2 \bar{p}_D}{dr_D^2} + \frac{1}{r_D} \frac{d\bar{p}_D}{dr_D} = u \bar{p}_D \quad (\text{A7})$$

where \bar{p}_D is the dimensionless pressure in Laplace domain, u refers to the Laplace variable, and r_D is the dimensionless distance. The initial condition is:

$$\bar{p}_D(r_D, 0) = 0. \quad (\text{A8})$$

The internal boundary condition and exterior boundary respectively are:

$$\lim_{r_D \rightarrow 0} \left[r_D \frac{d\bar{p}_D}{dr_D} \right] = -1 \quad (\text{A9})$$

$$\bar{p}_D(\infty, t_D) = 0. \quad (\text{A10})$$

The general solution of Equation (A7) is shown as following:

$$\bar{p}_D = \frac{1}{u} K_0(r_D \sqrt{u}) \quad (\text{A11})$$

where K_0 is the first class zero order Bessel function. With pressure superposition method, the pressure solution of the infinite conductivity vertically fractured model is obtained:

$$\bar{p}_D = \frac{1}{u} \int_{-1}^1 K_0 \left(\sqrt{(x_D - x_i)^2 + y_D^2} \sqrt{u} \right) da \quad (\text{A12})$$

Appendix C. Dual-Porosity Model with Pseudo-Steady State

In the dual-porosity model, the corresponding dimensionless variables are:

$$p_{wD} = \frac{k_f h \Delta p}{1.842 \times 10^{-3} q \mu B} \quad (\text{A13})$$

$$t_D = \frac{3.6kt}{(\varphi C_t)_{f+m} \mu L^2} \quad (\text{A14})$$

$$C_D = \frac{0.1592C}{(\varphi C_t)_{f+m} \mu L^2} \quad (\text{A15})$$

$$\lambda = aL^2 \frac{k_m}{k_f} \quad (\text{A16})$$

$$\omega = \frac{(\varphi C_t)_f}{(\varphi C_t)_f (\varphi C_t)_m} \quad (\text{A17})$$

where subscript f is the natural fracture system, subscript m refers to the matrix system, w is the wellbore system, λ is the interporosity flow coefficient, ω refers to the storage ratio. The diffusion equation of the pseudo-steady state in Laplace domain can be expressed as:

$$\frac{d^2 \bar{p}_{fD}}{dr_D^2} + \frac{1}{r_D} \frac{d\bar{p}_{fD}}{dr_D} = \lambda (\bar{p}_{fD} - \bar{p}_{mD}) + \omega u \bar{p}_D \quad (\text{A18})$$

$$(1 - \omega) u \bar{p}_{mD} = \lambda (\bar{p}_{fD} - \bar{p}_{mD}). \quad (\text{A19})$$

The initial condition is:

$$\bar{p}_{fD}(r_D, 0) = \bar{p}_{mD}(r_D, 0) = 0. \quad (\text{A20})$$

The boundary conditions are:

$$C_D u d\bar{p}_{wD} - \frac{d\bar{p}_{fD}}{dr_D} \Big|_{r_D=1} = \frac{1}{u} \quad (\text{A21})$$

$$\bar{p}_{wD} = \left(\bar{p}_{fD} - S \frac{d\bar{p}_{fD}}{dr_D} \right) \Big|_{r_D=1} \quad (\text{A22})$$

$$\bar{p}_{fD} = \bar{p}_{mD} = 0 \Big|_{r_D \rightarrow \infty} \quad (\text{A23})$$

where S is the skin factor and C_D is the dimensionless wellbore storage coefficient. Combining Equations (A18) and (A19), the general solution is determined as [44–46]:

$$\bar{p}_{wD} = \frac{K_0(\sqrt{f(u)u}) + S \sqrt{f(u)u} K_1(\sqrt{f(u)u})}{u \sqrt{f(u)u} K_1(\sqrt{f(u)u}) + C_D u^2 K_0(\sqrt{f(u)u}) + S C_D u^2 \sqrt{f(u)u} K_1(\sqrt{f(u)u})}. \quad (\text{A24})$$

In Equation (A24),

$$f(u) = \frac{\omega(1 - \omega) \times u + \lambda}{(1 - \omega) \times u + \lambda}. \quad (\text{A25})$$

Appendix D. Radial Composite Model

For the radial composite model, the dimensionless variables are defined as follows:

$$p_{irD} = \frac{k_{ir}h\Delta p}{1.842 \times 10^{-3}q\mu_{ir}B} \quad (\text{A26})$$

$$p_{erD} = \frac{k_{er}h\Delta p}{1.842 \times 10^{-3}q\mu_{er}B} \quad (\text{A27})$$

$$p_{wD} = \frac{k_{ir}h\Delta p_{wf}}{1.842 \times 10^{-3}q\mu_{ir}B} \quad (\text{A28})$$

$$t_D = \frac{3.6k_{ir}t}{\varphi\mu_{ir}C_tL^2} \quad (\text{A29})$$

$$C_D = \frac{0.159C}{\varphi h C_t L^2} \quad (\text{A30})$$

$$r_D = \frac{r}{L} \quad (\text{A31})$$

$$r_{fD} = \frac{r_f}{L} \quad (\text{A32})$$

$$M = \frac{(k/\mu)_{ir}}{(k/\mu)_{er}} \quad (\text{A33})$$

$$W = \frac{(\varphi C_t)_{ir}}{(\varphi C_t)_{er}} \quad (\text{A34})$$

where p_{irD} and p_{erD} are the dimensionless pressure in inner region and outer regions, r_{fD} is the dimensionless radius of the interface, M is the mobility ratio, W is the dispersion ratio. The diffusion equations in Laplace domain for the inner and outer regions of the composite model can be written as:

$$\frac{1}{r_D} \frac{d}{dr_D} \left(r_D \frac{d\bar{p}_{irD}}{dr_D} \right) = u\bar{p}_{irD} \quad (\text{A35})$$

$$\frac{1}{r_D} \frac{d}{dr_D} \left(r_D \frac{d\bar{p}_{erD}}{dr_D} \right) = \frac{W}{M} u\bar{p}_{erD}. \quad (\text{A36})$$

Correspondingly, the inner and outer boundary conditions are:

$$C_D u d\bar{p}_{wD} - \frac{d\bar{p}_{irD}}{dr_D} \Big|_{r_D=1} = \frac{1}{u} \quad (\text{A37})$$

$$\bar{p}_{wD} = \left(\bar{p}_{irD} - S \frac{d\bar{p}_{irD}}{dr_D} \right) \Big|_{r_D=1} \quad (\text{A38})$$

$$\bar{p}_{erD} \Big|_{r_D \rightarrow \infty} = 0. \quad (\text{A39})$$

There is an interface between the inner and outer regions. For this interface, the pressure and pressure derivative meet the following requirements:

$$\bar{p}_{irD} = \bar{p}_{erD} \Big|_{r_D=r_{fD}} \quad (\text{A40})$$

$$\frac{d\bar{p}_{irD}}{dr_D} = \frac{1}{M} \frac{d\bar{p}_{erD}}{dr_D} \Big|_{r_D=r_{fD}}. \quad (\text{A41})$$

Therefore, the solution can be determined as:

$$\bar{p}_{irD} = AK_0(r_D \sqrt{u}) + BI_0(r_D \sqrt{u}). \quad (\text{A42})$$

To satisfy the conditions of the interface, the value of A and B can be obtained:

$$A = \bar{q}_D \quad (\text{A43})$$

$$B = \frac{\bar{q}_D MK_0(r_{irD} \sqrt{Mu/W}) K_1(r_{irD} \sqrt{u}) \sqrt{u} - \bar{q}_D MK_1(r_{irD} \sqrt{Mu/W}) K_0(r_{irD} \sqrt{u}) \sqrt{Mu/W}}{I_0(r_{irD} \sqrt{u}) K_1(r_{irD} \sqrt{Mu/W}) \sqrt{Mu/W} - MI_1(r_{irD} \sqrt{u}) K_0(r_{irD} \sqrt{Mu/W}) \sqrt{u}}. \quad (\text{A44})$$

References

1. Muskat, M. The flow of homogeneous fluids through porous media. *Soil Sci.* **1938**, *46*, 169. [[CrossRef](#)]
2. Van Everdingen, A.F.; Hurst, W. The application of the Laplace transformation to flow problems in reservoirs. *J. Pet. Technol.* **1949**, *1*, 305–324. [[CrossRef](#)]
3. Horner, D.R. Pressure build-up in wells. In Proceedings of the 3rd World Petroleum Congress, The Hague, The Netherlands, 28 May–6 June 1951.
4. Ramey, H.J., Jr. Short-time well test data interpretation in the presence of skin effect and wellbore storage. *J. Pet. Technol.* **1970**, *22*, 97–104. [[CrossRef](#)]
5. Gringarten, A.C.; Ramey, H.J., Jr.; Raghavan, R. Unsteady-state pressure distributions created by a well with a single infinite-conductivity vertical fracture. *Soc. Pet. Eng. J.* **1974**, *14*, 347–360. [[CrossRef](#)]
6. Bourdet, D.; Ayoub, J.A.; Pirard, Y.M. Use of pressure derivative in well test interpretation. *SPE Form. Eval.* **1989**, *4*, 293–302. [[CrossRef](#)]
7. Zhou, Q.; Dillmore, R.; Kleit, A.; Wang, J.Y. Evaluating gas production performances in Marcellus using data mining technologies. *J. Nat. Gas. Sci. Eng.* **2014**, *20*, 109–120. [[CrossRef](#)]
8. Ma, Z.; Leung, J.Y.; Zanon, S.; Dzurman, P. Practical implementation of knowledge-based approaches for steam-assisted gravity drainage production analysis. *Expert Syst. Appl.* **2015**, *42*, 7326–7343. [[CrossRef](#)]
9. Lolon, E.; Hamidieh, K.; Weijers, L.; Mayerhofer, M.; Melcher, H.; Oduba, O. Evaluating the relationship between well parameters and production using multivariate statistical models: A middle Bakken and three forks case history. In Proceedings of the SPE Hydraulic Fracturing Technology Conference, The Woodlands, TX, USA, 9–11 February 2016.
10. Wang, S.; Chen, S. Insights to fracture stimulation design in unconventional reservoirs based on machine learning modeling. *J. Pet. Sci. Eng.* **2019a**, *174*, 682–695. [[CrossRef](#)]
11. Wang, S.; Chen, Z.; Chen, S. Applicability of deep neural networks on production forecasting in Bakken shale reservoirs. *J. Pet. Sci. Eng.* **2019b**, *179*, 112–125. [[CrossRef](#)]
12. Awoleke, O.; Lane, R. Analysis of data from the Barnett shale using conventional statistical and virtual intelligence techniques. *SPE Reserv. Eval. Eng.* **2011**, *14*, 544–556. [[CrossRef](#)]
13. Chu, H.; Liao, X.; Zhang, W.; Li, J.; Zou, J.; Dong, P.; Zhao, C. Applications of Artificial Neural Networks in Gas Injection. In Proceedings of the SPE Russian Petroleum Technology Conference, Moscow, Russia, 15–17 October 2018.
14. Akbilgic, O.; Zhu, D.; Gates, I.D.; Bergerson, J.A. Prediction of steam-assisted gravity drainage steam to oil ratio from reservoir characteristics. *Energy* **2015**, *93*, 1663–1670. [[CrossRef](#)]
15. Al-Kaabi, A.U.; Lee, W.J. Using artificial neural networks to identify the well test interpretation model (includes associated papers 28151 and 28165). *SPE Form. Eval.* **1993**, *8*, 233–240. [[CrossRef](#)]
16. Sultan, M.A.; Al-Kaabi, A.U. Application of neural network to the determination of well-test interpretation model for horizontal wells. In Proceedings of the SPE Asia Pacific Oil and Gas Conference and Exhibition, Melbourne, VIC, Australia, 8–10 October 2002.
17. Kharrat, R.; Razavi, S.M. Determination of reservoir model from well test data, using an artificial neural network. *Sci. Iran.* **2008**, *15*, 487–493.

18. AlMaraghi, A.M.; El-Banbi, A.H. Automatic Reservoir Model Identification using Artificial Neural Networks in Pressure Transient Analysis. In Proceedings of the SPE North Africa Technical Conference and Exhibition, Cairo, Egypt, 14–16 September 2015.
19. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 815–823.
20. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
21. Gatys, L.A.; Ecker, A.S.; Bethge, M. A neural algorithm of artistic style. *arXiv* **2015**, arXiv:1508.06576. [[CrossRef](#)]
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
23. Liu, D.; Li, Y.; Agarwal, R.K. Numerical simulation of long-term storage of CO₂ in Yanchang shale reservoir of the Ordos basin in China. *Chem. Geol.* **2016**, *440*, 288–305. [[CrossRef](#)]
24. Chu, H.; Liao, X.; Chen, Z.; Zhao, X.; Liu, W. Estimating carbon geosequestration capacity in shales based on multiple fractured horizontal well: A case study. *J. Pet. Sci. Eng.* **2019**, *181*, 106179. [[CrossRef](#)]
25. Meng, X. Horizontal Fracture Seepage Model and Effective Way for Development of Chang 6 Reservoir. Ph.D. Thesis, Southwest Petroleum University, Chengdu, China, 2018.
26. Chu, H.; Liao, X.; Chen, Z.; Zhao, X.; Liu, W.; Dong, P. Transient pressure analysis of a horizontal well with multiple, arbitrarily shaped horizontal fractures. *J. Pet. Sci. Eng.* **2019**, *180*, 631–642. [[CrossRef](#)]
27. Jingli, Y.; Xiuqin, D.; Yande, Z.; Tianyou, H.; Meijuan, C.; Jinlian, P. Characteristics of tight oil in Triassic Yanchang formation, Ordos Basin. *Pet. Explor. Dev.* **2013**, *40*, 161–169.
28. Hua, Y.A.N.G.; Jinhua, F.; Haiqing, H.; Xianyang, L.I.U.; Zhang, Z.; Xiuqin, D.E.N.G. Formation and distribution of large low-permeability lithologic oil regions in Huaqing, Ordos Basin. *Pet. Explor. Dev.* **2012**, *39*, 683–691.
29. Li, Y.; Song, Y.; Jiang, Z.; Yin, L.; Luo, Q.; Ge, Y.; Liu, D. Two episodes of structural fractures: Numerical simulation of Yanchang Oilfield in the Ordos basin, northern China. *Mar. Pet. Geol.* **2018**, *97*, 223–240. [[CrossRef](#)]
30. Guo, P.; Ren, D.; Xue, Y. Simulation of multi-period tectonic stress fields and distribution prediction of tectonic fractures in tight gas reservoirs: A case study of the Tianhuan Depression in western Ordos Basin, China. *Mar. Pet. Geol.* **2019**, *109*, 530–546. [[CrossRef](#)]
31. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105.
32. Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, Sardinia, Italy, 13–15 March 2010; pp. 249–256.
33. Ghaderi, A.; Shahri, A.A.; Larsson, S. An artificial neural network based model to predict spatial soil type distribution using piezocone penetration test data (CPTu). *Bull. Eng. Geol. Environ.* **2018**, 1–10. [[CrossRef](#)]
34. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
35. Chen, X.; Liu, S.; Sun, R.; Hong, M. On the convergence of a class of adam-type algorithms for non-convex optimization. *arXiv* **2018**, arXiv:1808.02941.
36. Reddi, S.J.; Kale, S.; Kumar, S. On the convergence of adam and beyond. *arXiv* **2019**, arXiv:1904.09237.
37. Mukhanov, A.; Arturo Garcia, C.; Torres, H. Water Control Diagnostic Plot Pattern Recognition Using Support Vector Machine. In Proceedings of the SPE Russian Petroleum Technology Conference, Moscow, Russia, 15–17 October 2018.
38. Hamdia, K.M.; Silani, M.; Zhuang, X.; He, P.; Rabczuk, T. Stochastic analysis of the fracture toughness of polymeric nanoparticle composites using polynomial chaos expansions. *Int. J. Fract.* **2017**, *206*, 215–227. [[CrossRef](#)]
39. Shahri, A.A.; Asheghi, R. Optimized developed artificial neural network-based models to predict the blast-induced ground vibration. *Innov. Infrastruct. Solut.* **2018**, *3*, 34. [[CrossRef](#)]

40. Saltelli, A.; Ratto, M.; Andres, T.; Campolongo, F.; Cariboni, J.; Gatelli, D.; Saisana, G.; Tarantola, S. *Global Sensitivity Analysis: The Primer*; John Wiley & Sons: Chichester, UK, 2008; ISBN 9780470725184.
41. Saltelli, A. Sensitivity analysis for importance assessment. *Risk Anal.* **2002**, *22*, 579–590. [[CrossRef](#)]
42. Vu-Bac, N.; Lahmer, T.; Zhuang, X.; Nguyen-Thoi, T.; Rabczuk, T. A software framework for probabilistic sensitivity analysis for computationally expensive models. *Adv. Eng. Soft.* **2016**, *100*, 19–31. [[CrossRef](#)]
43. Shahri, A.A. An optimized artificial neural network structure to predict clay sensitivity in a high landslide prone area using piezocone penetration test (CPTu) data: A case study in southwest of Sweden. *Geotech. Geol. Eng.* **2016**, *34*, 745–758. [[CrossRef](#)]
44. Chen, Z.; Liao, X.; Sepehrnoori, K.; Yu, W. A Semianalytical Model for Pressure-Transient Analysis of Fractured Wells in Unconventional Plays With Arbitrarily Distributed Discrete Fractures. *SPE J.* **2018**, *23*, 2041–2059. [[CrossRef](#)]
45. Chen, Z.; Liao, X.; Zhao, X.; Lyu, S.; Zhu, L. A comprehensive productivity equation for multiple fractured vertical wells with non-linear effects under steady-state flow. *J. Pet. Sci. and Eng.* **2017**, *149*, 9–24. [[CrossRef](#)]
46. Zongxiao, R.; Xiaodong, W.; Dandan, L.; Rui, R.; Wei, G.; Zhiming, C.; Zhaoguang, T. Semi-analytical model of the transient pressure behavior of complex fracture networks in tight oil reservoirs. *J. Nat. Gas Sci. Eng.* **2016**, *35*, 497–508. [[CrossRef](#)]



© 2019 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 (<http://creativecommons.org/licenses/by/4.0/>).