Missing Log Data Interpolation and Uncertainty Analysis via Deep Learning

Introduction

Reservoir modeling provides a simulation basis for the development of oil and gas fields and plays an important role in the evaluation of reservoirs. The distribution of underground lithology can be understood by analyzing the logging and seismic data of the target reservoir. Geologists usually use mathematical statistics to build reliable geological models based on logging data (Bader et al., 2018). However, instrument failure or lack of key types of logging data in the drilling will bring many difficulties to the analysis of reservoir characteristics.

In actual tasks, the method of re-drilling to obtain missing logging data is very expensive. To reduce the cost of obtaining missing logging data, geologists explore nonlinear relationships from existing logging data to establish an empirical model. However, the different types of underground rocks limit the application of empirical models, i.e., rely on manual selection. Deep learning (DL) technology is an expression of machine learning (Li et al., 2020). DL extracts the characteristics of information by structuring multiple hidden layers, and then combines these characteristics into a higher-level representation. In sequence prediction tasks, long short-term memory (LSTM) networks are widely used. Chen et al. (2020) proposed a multi-layer LSTM to predict reservoir porosity and tested it in logging data of different regions and depths. Lai et al. (2018) proposed a long- and short-term time-series network (LSTNet) combining multiple frames to extract nonlinear and linear features in time-series. In the estimation task of missing logging data, a combination of multiple frameworks may have better performance.

In this paper, we construct two data feature extraction methods, i.e., non-linear and linear feature extraction. The nonlinear feature extraction module consists of two 1D convolutional layers and two LSTM layers. The two LSTM layers are the conventional LSTM layer and the LSTM layer (skip-LSTM) with a time skip mechanism. The 1D convolution layer is used to initially extract and reorganize the features of the input logging data. The introduction of the skip mechanism will enhance the feature extraction capability of the network when the logging depth span is large. Besides, we introduce an attention mechanism in the nonlinear module to adaptively adjust the network weights. The linear module consists of an autoregressive (AR) model. This kind of depth-series LSTM driven by logging data is called LSTD. Our proposed method has verified its effectiveness and accuracy in real logging data experiments.

Method

The 1D convolution layer is placed on the first layer of the LSTD to initially extract the features of the input logging data and reorganize them in the form of feature maps. The output $h_i$ of the $i$th filter is:

$$h_i = \text{ReLU}(W_i \ast X + b_i),$$

where $X$ denotes a input matrix, $\ast$ denotes a convolution operation, $W_i$ and $b_i$ denote the training parameters of the network. ReLU is a nonlinear activation function.

After the initial extraction of the 1D convolutional layer, the information is passed into the forget gate of the LSTM and skip-LSTM layers respectively. The LSTM and skip-LSTM layers update information in the same way. However, the time step $t$ of the skip-LSTM layer includes a custom skip multiple $n$ and a period parameter $p$ to increase the connection between shallow- and depth-series. The forget gate $f^{(t)}$ in the skip-LSTM can be expressed as:

$$f^{(t)} = \text{sigmoid}(W_{fs}s^{(t-np)} + W_{fh}h^{(t)} + b_f),$$
where $W_{fs}$ and $W_{fh}$ denote the weight matrices to be trained, $b_f$ denotes the bias vector added in the forget gate.

The cell $\tilde{c}^{(t)}$ in the input gate will filter the information to update the status of the current cell $c^{(t)}$. The input gate $i^{(t)}$, $\tilde{c}^{(t)}$, and current $c^{(t)}$ can be expressed as:

$$i^{(t)} = \text{sigmoid}(W_{is}s^{(t-np)} + W_{ih}h^{(t)} + b_i),$$  

$$\tilde{c}^{(t)} = \text{ReLU}(W_{cs}s^{(t-np)} + W_{ch}h^{(t)} + b_{\tilde{c}}),$$  

$$c^{(t)} = f^{(t)} \odot c^{(t-np)} + i^{(t)} \odot \tilde{c}^{(t)},$$  

where $\odot$ denotes the element-wise product. Finally, the output gate uses the sigmoid and ReLU activation functions to determine the information state of the current cell $c^{(t)}$ and transmits it backward. The output gate $o^{(t)}$ and the final output $s^{(t)}$ are as follows:

$$o^{(t)} = \text{sigmoid}(W_{os}s^{(t-np)} + W_{oh}h^{(t)} + b_o),$$  

$$s^{(t)} = o^{(t)} \odot \text{ReLU}(c^{(t)}).$$

The structure of our proposed method is shown in Figure 1. It contains two 1D convolutional and LTM-based layers. After each LSTM layer, an attention mechanism is introduced to adjust the weight of the network, and then the nonlinear module and the AR-based linear module are combined to obtain the final output. The number $f$ of filters in the convolution layer is 16, and the size of the convolution kernel $s$ is the type of input logging data, i.e., $s=5$. The number of cells in both LSTM and skip-LSTM layers is 48. It is worth noting that the LSTD uses an early stopping mechanism and an adaptive learning rate optimization scheme. In other words, the initial learning rate is set to 0.001, which is reduced by 3 times every 20 epochs. When the loss of the validation set for 20 consecutive epochs does not decrease, the network will stop iterating and save the optimal training model.

![Figure 1](image.png)

**Figure 1** The architecture of our proposed method.

**Examples**

To verify the robustness of our proposed method, two main experiments are carried out: (1) missing log data estimates in different intervals, (2) uncertainty analysis. All the experiments are performed with the Tensorflow and Keras frameworks in Python on PC (Intel Core i7-6700 2.6-GHz CPU, 8 GB memory, and a NVIDIA GeForce GTX 960 GPU).
Missing log data estimates in different intervals

In this experiment, ten wells are used to predict the missing sonic (AC) and density (DEN). The data set A comes from an oil field in northern China, and 10 wells contain a total of 65,600 samples. The well contains seven logs: DEN, AC, grammy ray (GR), resistivity (RS), spontaneous potential (SP), shale (SH), and compensated neutron (CNL), with a logging interval of 0.125 m. The hyperparameters of the LSTD are measured through cross-validation experiments. Among them, the period value $p$ and the LSTM-skip multiple $n$ are set to 16 and 5, respectively, i.e., the logs within 10 m before each sampling point during network training participate in the training of the current depth-series. This experiment uses LSTM as a comparison model, and the hyperparameter setting rules of the LSTM are consistent with the LSTD. We select a well A10 that does not participate in training as the test well from the data set, and assume that AD and DEN in this well are missing at 200-350 m and 700-850 m. The interpolated results of missing logging data are shown in Figure 2. From the left side of Figure 2 (red dashed frame boxes), it can be seen that LSTD has a good performance in missing logging data interpolation at different intervals. Our proposed method extract depth-series features from different wells in the same block, and then use this feature to effectively predict the missing logs of a blind well. The right side of Figure 2 is the cross-plot of the LSTD estimated and true values in the two missing logging intervals.

![Figure 2](image)

**Figure 2** The interpolated performance of our proposed method in A10 well. Left: Comparison of the interpolation results of the two methods in AC and DEN. Right: Cross-plots of our propose method estimated and true values in different missing logging intervals.

Uncertainty analysis

In many tasks based on DL, the uncertainty research of the model is often ignored, which will cause ambiguity in task accuracy. This experiment uses A11 well (from the same region as A10 well) to test the uncertainty of LSTD in the interpolation task. Among them, the input parameters of the network are RS, GR, DEN, and TVD, and the target parameters are the other six types of logs from A11 well. We select the last 20% from the target logs in A11 as missing data. It can be seen from the left side of Figure 3 that our proposed method has a good performance in interpolating different types of logging data. The right side of Figure 3 (yellow dashed frame box) shows the relationship between epistemic uncertainty and prediction error of LSTD in this experiment. We use the dropout approximation method to generate uncertainty.
Epistemic uncertainty can be used to detect whether the prediction error of the model is within an acceptable range. It is worth noting that there is an obvious positive correlation between the predicted epistemic uncertainty and the prediction error of each well. The intersection of prediction error and model uncertainty is stable, which shows that the method we propose is robust.

Figure 3 Prediction results of the A11 well. Left: the estimation results of the six logs. Right: the relationship between uncertainty and prediction error.

Conclusions

In this paper, we perform missing logging data interpolation tasks by combining two 1D convolutional and two different LSTM layers. The skip-LSTM layer with multiple period parameters can effectively explore the changing trend of logging data when the logging depth span is large. The introduction of attention and layer normalization mechanisms can accelerate network training and adaptively adjust the weights of network iterations. In addition to building nonlinear components, we also introduced an AR component to the network to explore the linear relationship between logging data. This combination of nonlinear and linear modules can effectively predict missing logging data even with fewer data samples, thereby reducing the cost of re-drilling.

Acknowledgements

This research are supported by the National Key R&D Program of China (No. 2019YFC0312003) and the Strategic Cooperation Technology Projects of CNPC and CUPB (ZLZX2020-03).

References


