Deep Learning for Well Data History Analysis

Yuanjun Li
Department of Energy Resources Engineering
Stanford University
yuanjun@stanford.edu

Ruixiao Sun
Department of Energy Resources Engineering
Stanford University
ruixiaos@stanford.edu

Abstract

The rapid development of deep learning algorithms and the massive accumulation of well data from continuous monitoring has enabled new applications in the oil and gas industries. Data gathered from well sensors are a foundation of the oilfield digitization and data-driven analysis. The main objective of this work is to predict well pressure based on flow rate recorded from well surface or downhole gauges. Essentially we hope models will be able to learn the complex underground geological composition and structure. Then the deep learning model can be served as a simulation tool to simulate ideal condition which can not be realistic for practical control. With advanced RNN methods, engineers can interpret long-term reservoir performance information from responses estimated by the deep learning model, instead of performing costly well tests or shut-ins. Here, we describe a deep learning approach to predict the long-term well performance based on a moderate duration of well monitoring data. Specifically, we used LSTM and GRU as baseline models and further implemented LSTNet(Long- and Short-term Time-series network) structure.

1 Introduction

1.1 Motivation and Objective

Conventionally in oil and gas industry, pressure and flow rate data are obtained from physical models, which often require comprehensive geological and reservoir properties information. Besides, the calculation of reservoir parameters requires information from wells long shut-in or long constant flow rate period of time, which is not realistic and will lead to economic losses. What is more, sensors especially high-accuracy permanent downhole gauges(PDG) are very expensive[4]. Therefore, we want to explore a model which can utilize data from sensors to generate the rest data. Here, we describe a deep learning approach to predict the long-term well performance based on a moderate duration of well monitoring data.

The main objective is to predict well pressure based on flow rate with deep learning approaches, which can significantly contribute to reservoir monitoring, management, and optimization[2]. The input definitely contains gas/oil flow rate(q), and performance of adding other features like time, time intervals as input will be explained in later chapters. The output is well bottom hole pressure (BHP). Both of input and output are time-series data.

1.2 Data

The data is displayed in Fig.1. (a) is flow Rate (q). The flow rate used in this study is from a gas field. Raw pressure is pre-processed to be more continuous and accurate. (b) shows the raw measurement

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

of from pressure gauge. Due to the properties of gas and facility operation requirement, there might be discontinuous or inaccurate measurement. So based on physical equations, we generated the simulated pressure data to substitute the raw records. (c) Simulated pressure - constant time step. (d) Simulated pressure - unequal time step. Pressure in (d) is what we used in this project.

The input and output data are pre-processed and normalized. Our train-validation-test data is split 80%-10%-10%

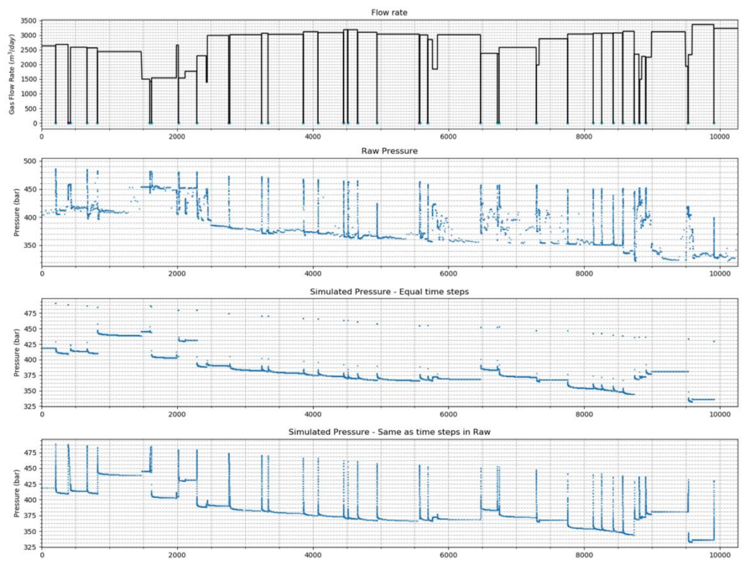

Figure 1: (a)Input flow rate; (b),(c),(d) Output pressure.

2 Methods

Considering the BHP and q are time series type of data, recurrent neural network(RNN) is preferred to solve the problem. It can process any length of input and can compute current output with using information from many steps back. In the project, we investigated various deep learning models to forecast well BHP: Gated Recurrent Unit (GRU), Long Short Term Memory (LSTM), and Long- and Short-term Time-series network (LSTNet)[1]. It turns out that LSTnet is less familiar and works better than GRU/LSTM, so we will focus on LSTNet in illustration of architecture, hyperparameter choices, and results analysis.

2.1 GRU/LSTM

Here we applied two RNN(GRU+LSTM) layers and one dense layer. Hyperparameters are tuned and selected in Table1. The result is displayed in next chapter. Because of the limit of pages, we will not explain hyperparameter tuning process in detail.

Batch-size	8
GRU unit	512
LSTM unit	512
Loss	Mean squared error
Optimizer	Adam(learning rate=0.001)

Table 1: Hyperparameter for GRU/LSTM architecture

2.2 LSTNet model

LSTNet[1] is applied in this project, which can capture short-term and long-term dependency in dataset. It is designed for multivariate time series forecasting, whose architecture is illustrated in 2. It is composed of convolutional layers, recurrent layers (GRU in [1]), novel recurrent-skip structure and fully-connected layers. The convolutional layer helps to capture local dependency patterns; the

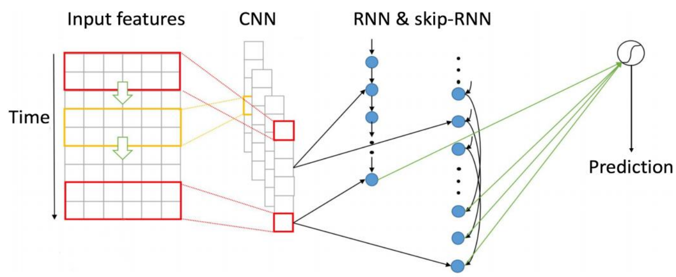

Figure 2: An overview of the Long and Short Term Time-series Network

recurrent layers is applied to for long-term dependency; the recurrent skip leverages the periodic pattern in dataset and makes the optimization easier. Besides skip-RNN, attention function is an alternative method applied to learn weighted combination of hidden representations at each window position of the input.

We applied two evaluation metrics in LSTNet model: relative squared error (RSE) in and relative absolute error (RAE) in Eqn.2.2

$$RSE = \frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{\sum_{i=1}^{n} (\overline{y} - y_i)^2} \qquad RAE = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{\sum_{i=1}^{n} |\overline{y} - y_i|} \tag{1}$$

3 Experiments/Results/Discussion

3.1 Hyperparameter tuning

In this project, we investigated and tuned the LSTNet model by considering input features, and number of past time intervals n_past as input, learning rate, CNN kernels size, dropout and evaluation metrics. We discuss each hyperparameter performance individually in the following.

3.1.1 Input features

Various groups of input features were tested. We firstly compared performance of Group1 $\{\Delta q, q, \Delta t\}$ and Group2 $\{\Delta q, q, \Delta t, t\}$ and found out that RSE and RAE is reduced obviously if current time t considered. Adding t as input can help to capture the tendency of pressure decreasing in long-term time period, which greatly improve model performance. Then, we considered increasing t's influence in the model and applied t^2 term in Group3 $\{\Delta q, q, \Delta t, t, t^2\}$. In addition, more input features are explored $\{\Delta q, q, \Delta t, t, t^2\ \Delta q \Delta t, \Delta q / \Delta t, \Delta q log(\Delta t)\}$ as Group4. Results are displayed in Fig3, from which we find out that input Group3 has best performance in val data set.

3.1.2 Number of past time step

Features of past time steps n_past are used as input of CNN. In the experiments, we explored n_past value from 5 to 200, and found out that loss is smaller as n_past is small. The validation dataset RSE and RAE are 0.2905 and 0.1262 while n_past is 5. We found only limited number of past time step can help to improve performance. When the number becomes larger, it prevents the model focusing on the current time features, which reduces performance. When n_past is smaller, LSTNet model can capture more portion of features of current time step, and less of previous time steps, which can improve the performance of predicting corresponding current pressure. Validation set losses are displayed in Fig.4.

Input								Output	Val	
									RSE	RAE
Δq	q	Δt						p	0.4568	0.2523
Δq	q	Δt	t					p	0.2503	0.0980
Δq	q	Δt	t	t^2				p	0.2332	0.0831
Δq	q	Δt	t	t^2	Δq*Δt	Δq/Δt	Δq*log(Δt)	p	0.2584	0.0849

Figure 3: Performances of four groups input features

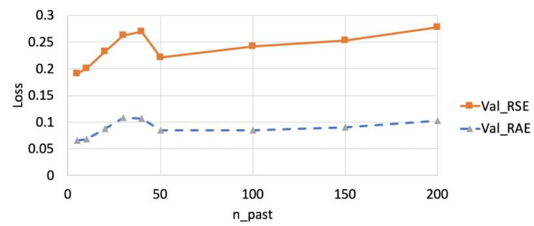

Figure 4: Performances of various n_past values

3.1.3 Other hyperparameters

In order to achieve possible best models, other hyperparameters are also tuned. We showed the performance of influential hyperparameters in Fig.5, within which optimal choices have been highlighted. We found out that dropout sensitivity and learning rate have more effects. Dropout can randomly remove some neurons to help to generalize the model. However, removing too much neurons (dropout=0.5) reduces the model performance and make validation set loss flip up. As for learning rate, it is tested that it have most effects on the result. Large learning rate can have a large update, but it will result in problems in convergence and get bad optimization results. Very small learning rate will consume more computation time before getting optimal results. Here we present

Hyperparameter		Val	
		RSE	RAE
Dropout	0.2	0.3254	0.1324
	0.3	0.2906	0.1144
	0.4	0.2226	0.0773
	0.5	0.2438	0.0838
Loss	L1	0.2329	0.0820
	L2	0.2378	0.0876
Learning rate	0.001	0.2708	0.1022
	0.01	1.7538	0.7779
# of CNN kernels	5	0.2533	0.0963
	10	0.2074	0.0775
	20	0.2210	0.0757
	30	0.2270	0.0857

Figure 5: Performances of various hyperparameters

our hyperparameter choices in Table2

3.2 Result and Discussion

With tuned optimal hyperparameters applied, we ran GRU/LSTM and LSTNet models with various input features and the results are visualized in Fig.6 (the first plot is GRU/LSTM; the rest are LSTNet with various input features).

From Fig6 and Fig7, we can see GRU/LSTM model cannot capture the whole decreasing trend, nor the large pressure variations; the same goes for LSTNet (4 features), and its prediction is more oscillating than LSTM/GRU results; LSTNet (4 features + time) has much better performance with adding real time as input, which can capture pressure decreasing trend and become more stable; LSTNet(4 features + time + 3 physics-based feature) improves the model performance in large pressure variation.

Batch-size	Dropout	Loss	lr	kernel size	RNN layers
32	0.4	L2	0.001	10	3

Table 2: Hyperparameter for LSTNet

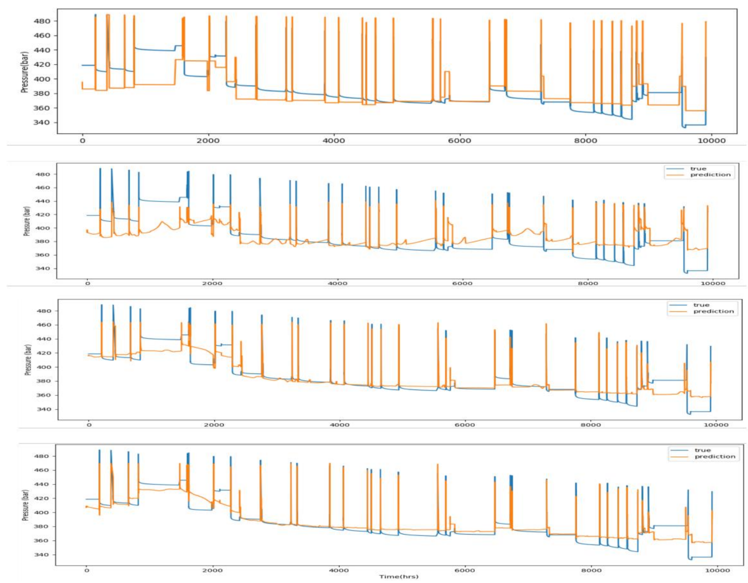

Figure 6: Results of GRU/LSTM model (first) and LSTNet model with various features

The study found that the input of real time values is really important to reflect the trend of pressure decreasing in the long term. Besides, the length of short term captured is worse when it's too long. When the short term contains too much previous time information, the model will focus less on current time step and the performance will be decreased.

4 Conclusion/Future Work

In this project, we explored two time-sequential models in deep learning to represent dynamic flow-rate and pressure in reservoir well. We implemented LSTM/GRU and LSTNet individually to predict well pressure given flow rate, which is many-to-many mapping. We found out that LSTNet which can capture both short and long term time periods has better performance, especially for capturing extreme variations in pressure. In addition, as for the input features of LSTNet, adding real time value can help to capture the whole pressure decreasing tendency and achieve better prediction results.

In the future, our work can be divided into two stages. First, we will focused on how to get better predictions for large pressure variations, which can help to reduce errors significantly. Then, we will implement continuous RNN-GAN model. Continuous RNN-GAN [3] is designed to learn the distribution behind well production and generate realistic data. It can help us generate more well flow rate data in real reservoir production, and better analyze and optimize the reservoir future performance.

Structure	Test_RAE	Test_RSE
LSTNet (4 features)	0.4723	0.2601
LSTNet (4 features + time)	0.3392	0.1512
LSTNet (4 features + time + 3 physics-based feature)	0.2802	0.1002

Figure 7: Test data loss for LSTNet model with three different input features

5 Contributions

Yuanjun worked on data collection, building GRU/LSTM network, and testing cases in GPU. Ruixiao worked on applying and modifying LSTNet, and adding functions to LSTNet. Hyperparameters design are discussed and cooperated.

6 Additional Information

We would like to express our great appreciations to our TA Sarah Najmark for insightful discussions and helpful suggestions provided to our project.

References

[1] Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 95–104. ACM, 2018.

[2] Yuanjun Li, Andrei Popa, Andrew Johnson, Iraj Ershaghi, Steve Cassidy, et al. Dynamic layered pressure map generation in a mature waterflooding reservoir using artificial intelligence approach. In *SPE Western Regional Meeting*. Society of Petroleum Engineers, 2018.

[3] Olof Mogren. C-rnn-gan: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*, 2016.

[4] Chuan Tian, Roland N Horne, et al. Applying machine-learning techniques to interpret flow-rate, pressure, and temperature data from permanent downhole gauges. Society of Petroleum Engineers, 2019.