Time Series Data Prediction Based on Sequence to Sequence Model

Chao Yang, Zhongwen Guo and Lintao Xian
Department of Computer Science and Technology, Ocean University of China, Qingdao, China
xianlintao@qq.com

Abstract. Home appliance test is an essential part of the R&D process. However, since home appliance test must be carried out under certain circumstances, it makes the development cycle longer. If it is possible to predicting the home appliance test data, it will improve the efficiency of home appliance development. Sequence to sequence model has been proved that it has a strong ability to map the input sequence and output sequence and it has been widely used in machine translation tasks. In order to solve the home appliance test data prediction problem, we firstly tried to apply sequence to sequence model to predict numerically continuous time series data. Our experiments proved that our model can predict the data of relatively long-time steps when inputting data of relatively short time steps. In our experiment, the mean absolute percentage error of the prediction of our model is about 0.1%. Our approach has a strong generalization ability and performs well in different experimental scenarios. Finally, we believe that the sequence to sequence method performs satisfactorily on prediction problems for continuous time series data. With this model, we can obtain more accurate results than traditional methods and it is meaningful for the R&D of home appliances.

1. Introduction
With the development of smart homes, the pattern of the home appliance manufacturing industry has changed from the enterprise productions decides the user’s demand to the user's demand decides the original enterprise productions. The home appliance manufacturing industry customizes the products according to the user's needs. Thus, the frequency of research and development of new products is becoming higher and higher. In the process of R&D of home appliances, product test is an indispensable part. Home appliances need to run continuously for hundreds to thousands of hours under certain working conditions, resulting in a long development cycle. Predicting the refrigerator test data shortens the test cycle, which will improve the research and development efficiency of new products.

To address prediction problems, traditionally we use linear regression, lasso regression, polynomial regression, etc. However, these kinds of algorithms depend on choosing the regression terms manually and do not have satisfying performances when the actual formula is extremely complicated. Thanks to the fast development of deep learning, we are able to use artificial neural networks (ANN) to address the regression problems [1]. Because of the character of ANN, researchers no longer need to set regression terms manually, but use an end-to-end approach that the model can automatically learn the features in the data and use it to complete the regression tasks. The simplest ANN, also called Multi-Layer Perception (MLP) [2], has been proven to be a general function approximation that can be used
to match the nonlinear part more accurately and have better performances comparing with the previous algorithms.

Unfortunately, when the data have time-series characters, MLP is not able to utilize the time features of the data. In order to take advantage of the time series information of the data, Hihi [3] proposed recurrent neural networks (RNN).

A standard RNN does not only focus on the current input, but also utilize the sequence features. I.e., it is easier for a standard RNN to map sequences to sequences [4,5].

But, when the length of the input sequence is different from the length of the output sequence, it will be challenging for RNN to map the two sequences. In addition, when the length of the sequence is quite long, RNN will be tricky to train and mostly suffer for gradient vanish [6]. Later, a new structure called Long Short-Term Memory (LSTM) [7] became popular. Thanks to the special design of the LSTM module, it can address the gradient vanish problem when facing a long-term sequence and it has a strong ability for continual prediction [8].

The principle of a LSTM is to estimate the conditional probability \( p(y_1, \ldots, y_T \mid x_1, \ldots, x_T) \) when given the input sequence \((x_1, \ldots, x_T)\). A standard LSTM formulation is:

\[
p(y_1, \ldots, y_T \mid x_1, \ldots, x_T) = \prod_{t=1}^{T} p(y_t \mid v, y_1, \ldots, y_{t-1})
\]

Thanks to LSTM, it is possible to predict a longer-term sequence. It is almost the best choice when addressing natural language processing problems. To make LSTM module more easily to train and implement, Cho et al. [9] proposed a novel structure called Gated Recurrent Unit (GRU) and it has been proved that GRU is faster for training than LSTM and has almost the same performance.

However, these kinds of structures are still hard to address the problems when the input length and output length are different. To address this problem, Cho et al. [9] developed sequence to sequence model.

The sequence to sequence model, often called seq2seq for short, is inspired by auto-encoder structure [10]. The difference is that instead of using multiple layers of neural network in the encoder and decoder, we use RNN module in the encoder part to change the sequence into an context vector, the context vector, which consist of the information of the input sequence, will be the input or the initial hidden state to send to the decoder part.

Generally, we choose the last time-step output of the encoder as the context vector. But it has been proved to be unreasonable and unsuitable for translation tasks. Graves [11] proposed a novel mechanism called attention which allowed the decoder to concentrate on different part of encoder’s hidden state to choose the necessary and important information for translation. This seq2seq with attention structure has been proven to have an excellent performance for sequence prediction tasks such as machine translation [12,13,14], speech recognition [15,16], emotion classification [17], video analysis [18], trajectory prediction [19], image captioning [20], etc.

2. The Model

In this paper, to address the home appliances test data prediction task, instead of the traditional methods, we propose a method based on seq2seq model.

The encoder consists with a two-layer GRU. Comparing with the model that usually uses in machine translation tasks, we abandon word2vec [21], also called embedding layers, in our encoder part because our data do not suffer for sparse input issues. The input will go through all the GRUs and then convert to the context vector.

In the middle of encoder and decoder we use a single fully connected layer network for adjusting the dimension of the data. In addition, it is kind of like attention mechanism that allowed the neural network to choose the information that are more useful for our task automatically.
The decoder consists with a two-layer GRU which has the same structure with the GRU which is used in the encoder. Following the GRU are fully connected layers. We use ReLU as the activation function in the middle layers of the neural network. Generally, we often use softmax at the last layer to force the neural network to choose the word with the maximum possibility. However, the test data of the appliances are continuous. In order to obtain continuous output, we abandon softmax and do not add any activation function in the last layer. Also, we do not add embedding layer in decoder part. The output of the GRUs in the decoder will then go through the fully connected layer and we will obtain the output of the model.

An illustration of the structure of our model can be seen in figure 1.

Figure 1. An illustration of the structure of our model

3. Experiments

3.1. Data Description and Processing
In this task, we will try to use the previous few sets of data to predict a large amount of subsequent data. We use refrigerator power consumption test data in our experiment. An overview of the data can be seen in table 1. In refrigerator power consumption test, we use seven sensors to capture the frozen temperature of the seven different position of refrigerator. Our test needs to be done in a certain environment temperature so we have two sensors to capture the environment temperature to make sure the environment temperature meets the requirement. Other sensors are used to capture the status of the refrigerator during the power consumption test.

We use a small length of data as input to predict some essential sensor data. In refrigerator power consumption test, we need to predict the power consumption and the temperatures that are important to the testers.

We collected experimental data from different kinds of refrigerator in different environments to make sure our model is robust and has generalization performance. We accumulated 105,000 time steps of data for training and 10,500 time steps of data for validation.

We have tried different length of the input and output, and to make sure the deviation is small enough and the length of the output is longer enough comparing with the length of the input, we set 20 as the max length of the input and set 500 as the max length of the output. We use normalization to make sure all the data are in similar range to avoid hurting the performance because of the range differences of the data. In this experiment, we have temperatures from 7 sensors, and we do care about
mean temperature. So, we just calculate the mean temperature and use it as the input to reduce the dimension of the input data.

| Feature       | Time Step | 1   | 2   | 3   | 4   | 5   | 6   |
|---------------|-----------|-----|-----|-----|-----|-----|-----|
| Frozen 1      |           | -17.40 | -17.40 | -17.30 | -17.20 | -17.20 | -17.20 |
| Frozen 2      |           | -13.20 | -13.10 | -13.00 | -13.00 | -13.20 | -13.40 |
| Frozen 3      |           | -13.80 | -13.80 | -13.60 | -13.70 | -14.10 | -14.40 |
| Frozen 4      |           | -16.30 | -16.30 | -16.20 | -16.20 | -16.10 | -16.10 |
| Frozen 5      |           | -16.30 | -16.30 | -16.20 | -16.20 | -16.10 | -16.10 |
| Frozen 6      |           | -16.10 | -16.10 | -16.00 | -15.90 | -15.90 | -15.90 |
| Frozen 7      |           | -17.00 | -16.90 | -16.90 | -16.80 | -16.70 | -16.70 |
| Environment 1 |           | 15.90 | 15.80 | 15.90 | 15.80 | 15.80 | 15.80 |
| Environment 2 |           | 16.10 | 16.10 | 16.10 | 16.10 | 16.10 | 16.10 |
| Voltage       |           | 99.80 | 99.80 | 99.80 | 99.80 | 99.90 | 99.90 |
| Electricity   |           | 1.476 | 1.442 | 1.415 | 1.413 | 1.424 | 1.427 |
| Power         |           | 77.40 | 70.60 | 65.00 | 65.30 | 66.00 | 66.60 |
| Frequency     |           | 50.00 | 50.00 | 50.00 | 50.00 | 50.00 | 50.00 |
| Consumption   |           | 0.1723 | 0.1729 | 0.1735 | 0.1740 | 0.1746 | 0.1751 |
| Humid Temp    |           | 13.50 | 13.50 | 13.50 | 13.50 | 13.50 | 13.50 |
| Power Factor  |           | 0.53 | 0.49 | 0.46 | 0.46 | 0.46 | 0.46 |

3.2. Training Details

We tried different kinds of RNN units in our encoder and decoder. We find GRU is easy and efficient to train comparing with LSTM. We use GRUs with two layers and each layer contains 256 cells. Although deep GRUs have been proved its significant performance [22], we want to balance the performance and the depth of our model to make sure it runs faster. We tried different number of layers and it turns out two-layer GRU structure is a good choice.

We reserve all the hidden states of the encoder instead of taking the hidden state of the last time-step to take full advantage of the input data. To make sure that the dimension is suitable for the input of the decoder, we add a single fully connected layer between the encoder and the decoder which is quite different from the traditional seq2seq model. It turns out this layer is quite useful and helps the model predict the output more precisely.

We have considered the fact that in different experiments, the kinds of data that the testers are interested in are different. So, for each kind of data, we trained a single model for it. In our test term, we let multiple models predict the data parallelly. This way has two advantages. Firstly, once the testers change the kinds of data that they are interested in, we just need to fine-tune our model to make it fit the requirement of the testers. Comparing with predicting all the kinds of data, it is cheaper in computation and more convenient to change the output. Secondly, according to our experiments, the model predicts a single kind of data more precisely than predicting multiple kinds of data. This may because the range of the different kinds of data is quite different, thus predicting all the data at the same time is trickier and more imprecise.

We use Adam optimizer and set 1e-6 as the learning rate. We trained our model for a total of 2000 epochs and our model converges.

4. Experiment Result

To evaluate the performance of our model more accurately and reasonably, we use two scores. Firstly, we use mean absolute error (MAE) to assess the deviation numerically. Secondly, we use mean absolute percent error (MAPE) to assess the quality of our prediction.

The specific calculation of MAE and MAPE can be seen in the following equations:
In these two equations, $y$ is the original data, $\hat{y}$ is the output of our model. MAE and MAPE are two common indicators to evaluate the regression performance. Because the numerical range of output may be highly different. Solely MAE or MAPE to assess our output is not reasonable. Therefore, we use both of them to evaluate the performance of our model. The MAE and MAPE at certain epoch of the training process are presented in table 2 and table 3.

The changes of MAPE and MAPE during the training process are presented in figure 2 and figure 3.

**Table 2.** Validation mean absolute error loss

|                      | Loss after 500 Epochs | Loss after 1000 Epochs | Loss after 1500 Epochs | Loss after 2000 Epochs |
|----------------------|------------------------|-------------------------|------------------------|------------------------|
| LR                   | 1.185242               | 1.06315                 | 1.049329               | 1.047863               |
| MLP                  | 0.011882               | 0.010381                | 0.009476               | 0.010658               |
| RNN                  | 0.006274               | 0.005227                | 0.004893               | 0.004010               |
| Our Model            | 0.006756               | 0.003951                | 0.002512               | 0.002136               |

**Table 3.** Validation mean absolute percentage error loss

|                      | Loss after 500 Epochs | Loss after 1000 Epochs | Loss after 1500 Epochs | Loss after 2000 Epochs |
|----------------------|------------------------|-------------------------|------------------------|------------------------|
| LR                   | 96.6006%               | 100.7325%               | 105.3969%              | 104.8149%              |
| MLP                  | 0.8833%                | 0.8086%                 | 0.8624%                | 0.7519%                |
| RNN                  | 0.4256%                | 0.3387%                 | 0.3008%                | 0.2471%                |
| Our Model            | 0.5276%                | 0.3032%                 | 0.1993%                | 0.1501%                |

**Figure 2.** Validation mean absolute error loss during training process

**Figure 3.** Validation mean absolute percentage error loss during training process

It turns out that based on our seq2seq model, after 2000 epochs, the MAPE is about 0.1% and the MAE is about 0.002, which is precise enough for the usage in the industry field. We can see that
comparing with linger regression, multiple layer perception and simple RNN, our model has a much fewer mean absolute error and mean absolute percentage error in validation state. The result suggests that our model has a better performance on predicting the sequence data than the previous methods.

5. Conclusion
In this paper, we showed that sequence to sequence model can learn the mapping from a sequence of a variable-length sequence to another variable-length sequence and it has a pretty good performance for continuous time series data prediction tasks.

We evaluate our model based on refrigerator power consumption test. We use MAE to optimize our model and use both MAE and MAPE to evaluate the performance because of the range distinction of the output data. Our experiment result shows that our model is able to extract and utilize the information efficiently. By using a short length of data, our model can predict a quite long length of data and it is helpful for the testers to reduce the experiment time and increase efficiency.

Because generally the deep learning algorithm is greedy for data, we do believe this time series data prediction method is suitable for predicting the time series data that are influenced by a various of factors. I.e., the input data should have enough dimensions as features to make sure the model has a satisfied performance.

We are excited about the ability of the sequence to sequence model to precisely predict such a long sequence. We believe sequence to sequence model has an excellent generalization and it is suitable for a various kinds of time series prediction tasks.

Acknowledgment
We would like to say thank you to all the colleagues in Ocean University of China that have helped us about our research. Chao Yang wants to thank his family for supporting his decision to dive in deep learning and for their support during his career.

References
[1] Kwok T Y and Yeung D Y 1997 Constructive algorithms for structure learning in feedforward neural networks for regression problems IEEE transactions on neural networks 8(3) pp 630-645
[2] Gardner M W and Dorling S R 1998 Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences Atmospheric environment 32(14-15) pp 2627-2636
[3] El Hihi S and Bengio Y 1996 Hierarchical recurrent neural networks for long-term dependencies In Advances in neural information processing systems pp 493-499
[4] Lipton Z C, Berkowitz J and Elkan C 2015 A critical review of recurrent neural networks for sequence learning arXiv preprint arXiv 1506.00019
[5] Jozefowicz R, Zaremba W and Sutskever I 2015 An empirical exploration of recurrent network architectures In International Conference on Machine Learning pp 2342-2350
[6] Hochreiter S 1998 The vanishing gradient problem during learning recurrent neural nets and problem solutions International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6(02) pp 107-116
[7] Hochreiter S and Schmidhuber Jü rgen 1997 Long short-term memory Neural Computation 9(8) pp 1735-1780
[8] Gers F A, Schmidhuber Jürgen and Cummins F 2000 Learning to forget: continual prediction with lstm Neural Computation 12(10) pp 2451-2471
[9] Cho K, Van Merrienboer B, Gulcehre C, Bahdanau, D, Bougares F and Schwenk H 2014 Learning phrase representations using rnn encoder-decoder for statistical machine translation arXiv preprint arXiv 1406.1078
[10] Hinton G E and Zemel R S 1994 Autoencoders, minimum description length and Helmholtz free energy In Advances in neural information processing systems pp 3-10
[11] Graves A 2012 Sequence transduction with recurrent neural networks. Computer Science
[12] Eriguchi A, Hashimoto K and Tsuruoka Y 2016 Tree-to-sequence attentional neural machine translation arXiv preprint arXiv 1603.06075
[13] Zhang J and Zong C 2015 Deep neural networks in machine translation: an overview IEEE INTELLIGENT SYSTEMS 30(5) pp 16-25
[14] Singh S P, Kumar A, Darbari H, Singh L, Rastogi A and Jain S 2017 Machine translation using deep learning: An overview. International Conference on Computer, Communications and Electronics (Comptelix) pp 162-167
[15] Chiu C C, Sainath T N, Wu Y, Prabhavalkar R, Nguyen P, Chen Z, and Jaitly N 2018 State-of-the-art speech recognition with sequence-to-sequence models IEEE International Conference on Acoustics, Speech and Signal Processing pp 4774-4778
[16] Graves A, Mohamed A R and Hinton G 2013 Speech recognition with deep recurrent neural networks IEEE international conference on acoustics, speech and signal processing pp 6645-6649
[17] Rana R 2016 Gated recurrent unit (GRU) for emotion classification from noisy speech arXiv preprint arXiv 1612.07778
[18] Venugopalan S, Rohrbach M, Donahue J, Mooney R, Darrell T and Saenko K 2015 Sequence to Sequence -- Video to Text IEEE International Conference on Computer Vision pp 4534- 4542
[19] Park S H, Kim B, Kang C M, Chung C C, and Choi J W 2018 Sequence-to-sequence prediction of vehicle trajectory via LSTM encoder-decoder architecture IEEE Intelligent Vehicles Symposium (IV) pp 1672-1678
[20] Wang C, Yang H, Bartz C and Meinel C 2016 Image captioning with deep bidirectional LSTMs In Proceedings of the 24th ACM international conference on Multimedia pp 988-997
[21] Mikolov T, Sutskever I, Chen K, Corrado G S and Dean J 2013 Distributed representations of words and phrases and their compositionality In Advances in neural information processing systems pp 3111-3119
[22] Pascanu R, Gulcehre C, Cho K and Bengio Y 2013 How to construct deep recurrent neural networks arXiv preprint arXiv 1312.6026