Main Steam Temperature Prediction Modeling Based on Autoencoder and GRU

Ling Zheng*, Xinwei Cao and Fei Chen
School of Control and Computer Engineering, North China Electric Power University, Beijing, China
Email: zhengling@ncepu.edu.cn

Abstract. There is an important parameter in the normal operation of thermal power plant units, that is, the main steam temperature, too high or too low will have a great impact on production. For safety and efficiency considerations, the main steam temperature needs to be controlled within a stable and appropriate range. In this way, the safety of equipment and people is guaranteed, but also ensures that resources will not be wasted. The prediction of the main steam temperature can achieve this aim. In view of the numerous influencing factors and non-linear characteristics of the main steam temperature, this paper proposes a main steam temperature prediction model based on autoencoder and GRU. First, build an autoencoder model, use historical data for training, enable it to accurately extract the essential information of the data. Then multiple influence parameters are dimensional reduced by the trained autoencoder, and use the reduced data as the input of the neural network built using GRU for training. The better model is finally compared with RNN and LSTM to verify the effectiveness and accuracy of the method.

Keywords. Thermal power plant; prediction of the main steam temperature; autoencoder; gate recurrent unit.

1. Introduction
With economic development and technological progress, electricity has become an indispensable foundation in our lives. According to an analysis of the development situation of the domestic power industry in a paper published by scholars studying domestic energy [1], in 2018, China’s electricity production through fossil fuel-fired thermal power generation reached 4.92 trillion kilowatt-hours, the proportion was 70.4%. During the operation of the thermal power plant units, the main steam temperature can have a huge impact on the unit's safe production and economic performance. Excessive main steam temperature will reduce the service life of the equipment, which will cause safety problems in severe cases. Excessively low main steam temperature will reduce the thermal efficiency of the unit. According to relevant data, a 5-10° decrease in main steam temperature will cause 1% Reduced thermal efficiency will waste fuel and result in economic losses.

If the main steam temperature beyond reasonable range, it will have a bad effect, so people need to control it within a suitable range, and predicting the main steam temperature can better achieve this goal. However, the large delay, non-linearity and time-varying characteristics of the main steam temperature cause the traditional control theory to have insufficient control effect. With the further research of scholars, through combining advanced related theories with control strategies, many practical methods have been proposed. For example, using the advantages of predictive control “advanced”, many experts at home and abroad have studied the technology of predictive control It was...
applied to the unit control of thermal power plant [2]. After analyzing the complex characteristics of the main steam temperature system, experts and scholars proposed a corresponding control method. Against many shortcomings of traditional PID control, many experts and scholars combine fuzzy control theory with main steam temperature control [3]; some scholars have incorporated BP neural network into PID control [4].

The main steam temperature and some factors affecting it are time series data. Time series prediction refers to the estimation of a variable at a certain time or time in the future based on the current and past time series sample values. Traditional time series prediction models include autoregressive models (AR) [5], moving average models (MA) [6], and autoregressive moving average models (ARMA) [7]. These models have been applied in many fields since they were proposed. Some scholars use the ARMA model for earthquake ground motion simulation [8], and Torres also uses the ARMA model when predicting wind speed [9].

Artificial neural network (ANN) has developed rapidly after entering the 21st century. Due to its advantages of non-linearity, non-limitation, very qualitative and non-convexity, it has a good performance in various fields. Among them, the recurrent neural network (RNN) uses the output of the neuron at the previous moment as the input of the neuron at the current moment to form a feedback system, which effectively uses the dependencies between the data, so it is very suitable for the analysis of time series data. Greatly improved the time series prediction effect. However, there is a problem of gradient disappearance or explosion of neural networks in deep network. Later, short-term memory networks (LSTM) [10] came into being. It added three “gate” structures based on RNN. Effectively solved this problem. Later, there was a simplified version of LSTM GRU.

Main steam temperature is affected by many factors. If all the factors are input into the network for training and prediction is made for a long period of time, the entire network has a large amount of calculation and low efficiency. The traditional PCA algorithm can reduce the dimensionality of the data and reduce data redundancy, but it loses the information content of the data. In order to find effective solutions, this paper proposes a method based on autoencoders and GRU. Main steam temperature prediction model. The autoencoder can effectively reduce the dimensionality of numerous data and save the original nature of the data. The dimensional reduced data is input to the GRU network. The GRU network predicting time series data has good results. Being lost again ensures the timing of the model. Finally, through a lot experiments, the output value of the network is compared with the label, which shows that the model has a good effect.

2. Introduction to Autoencoder

The paper published by Rumelhart et al. in Nature first proposed a basic autoencoder [11]. An autoencoder is an unsupervised learning algorithm whose output can reproduce the input data. Generally, an autoencoder includes an encoding stage and a decoding stage, and the structure is symmetrical. In figure 1, there is a basic network structure. The specific calculation formula is as follows:

Encoding process: \( h = f_1(w_1x + b_1) \) \hspace{1cm} (1)

Decoding process: \( y = f_2(w_2h + b_2) \) \hspace{1cm} (2)

Among them, \( w_1 \) and \( b_1 \) are encoding weights and biases, \( w_2 \) and \( b_2 \) are decoding weights and biases, \( f_1 \) is a non-linear transformation, currently commonly used are sigmoid, tanh, etc. \( f_2 \) can be the same non-linear transformation or encoding process Identity transformation. The loss function of the basic autoencoder is to minimize the error between \( y \) and \( x \):

When \( f_2 \) is an identity function, the loss function takes the square error:

\[ L(x, y) = \|y - x\|^2 \] \hspace{1cm} (3)

When \( f_2 \) is a sigmoid function, the loss function takes the cross-entropy function:

\[ L(x, y) = -\sum_{i=1}^{n} [x_i \log(y_i) + (1 - x_i) \log(1 - y_i)] \] \hspace{1cm} (4)
For the structure shown in figure 1, there are three different forms in the hidden layer representation of the basic autoencoder [12]: compressed structure, sparse structure, and isodimensional structure. The input layer of the compressed structure has more neurons than the hidden layer. Conversely, the input layer of the sparse structure has fewer neurons than the hidden layer. In the equal-dimensional structure, they are equal.

![Figure 1. Structure of basic autoencoder.](image)

In the basic autoencoder, the weights in the encoding and decoding stages are trained separately, and there is no practical connection. However, if $w_1 = w_2^T$, that is, the same weights are used in the encoding and decoding stages, this type of autoencoder is called a Tiedweight auto-encoder (TAE). It should be noted here that the bias is not bound.

The purpose of the basic autoencoder is to make the output equal to the input. Such a transformation is actually meaningless. What we really care about is the expression of the hidden layer. An autoencoder receives an input signal and converts it into an efficient internal representation. Its dimensions are generally smaller than the input dimensions. If the model can also reconstruct the signal at this time, it means that the hidden layer expression is sufficient to represent the input signal. This expression is automatically learned by the model. The effective features come out, which plays a role in reducing the dimension.

With the increasing demand, many improved autoencoders have been born, such as noise reduction autoencoders, sparse autoencoders, and so on. Since the parameters of the unit are relatively stable and do not vibrate significantly, this article uses a basic autoencoder to meet the experimental requirements.

3. GRU Introduction

3.1. RNN
Because ordinary neural networks cannot use the time series information in the data, a recurrent neural network (RNN) was proposed. Recurrent neural network can well describe the relationship of data on the timeline. Therefore, it is widely used for processing and forecasting time series data. Figure 2 shows a typical recurrent neural network structure and its expanded view. As can be seen from figure 2, the input of the recurrent neural network has two parts, one is the state of the previous moment, and the other is the input sample of the current moment. For time series data, the input sample of each moment examples can be the value at the current time. In order to transform the current state into the final output, there is a fully connected network behind the recurrent neural network to complete this process. The recurrent neural network can also be thought of as copying a cyclic body multiple times on the timeline. The parameters in the cyclic body network structure are also shared at different times. In ordinary RNNs, the structure of this loop body is very simple, just a single tanh layer.
3.2. LSTM

Although the purpose of RNN was designed to learn long-term dependence, a lot of practice also shows that standard RNN is often difficult to achieve long-term preservation of information. Bengio et al. [13] proposed that the standard RNN suffers from gradient disappearance and gradient explosion. To solve these problems, Hochreiter [10] proposed a Long-Short Time Memory (LSTM) network to improve the traditional recurrent neural network model. After continuous evolution, the most widely used LSTM model cell structure is shown in figure 3. LSTM is implemented through a structure called a gate. The gate plays the role of filtering information. This function can be realized by multiplying the vector and the gate point by point. The output of the gate is a vector with elements ranging from 0-1, which indicates how much information can pass through. Each LSTM has three such gate structures, namely input gate, forget gate, and output gate, to implement protection and control information. Among them, the input gate controls how much new information is added to the cell, the forget gate determines what information is allowed to continue through the cell, the final output value is determined by the output gate. The specific formula of LSTM cells is as follows:

\[ f_t = \text{sigmoid}(W_{fx}x_t + W_{fh}h_{t-1} + b_f) \]  
\[ i_t = \text{sigmoid}(W_{ix}x_t + W_{ih}h_{t-1} + b_i) \]  
\[ g_t = \tanh(W_{gx}x_t + W_{gh}h_{t-1} + b_g) \]  
\[ o_t = \text{sigmoid}(W_{ox}x_t + W_{oh}h_{t-1} + b_o) \]  
\[ c_t = g_t \cdot i_t + c_{t-1} \cdot f_t \]  
\[ h_t = \tanh(c_t) \cdot o_t \]

Among them, \( f_t, i_t, g_t, o_t, h_t \) are forgotten gates, input gates, candidate vectors, output gates, hidden layer outputs, \( W_{fx}, W_{fh}, W_{ix}, W_{ih}, W_{gx}, W_{gh}, W_{ox}, W_{oh} \) are each gate and The matrix weight multiplied by the input \( x_t \) at the current moment and the output \( h_{t-1} \) of the hidden layer at the previous moment, \( b_f, b_i, b_g, b_o \) are the biases of the corresponding gates, respectively.

\[ \text{Figure 2. Structure of RNN.} \]

3.3. GRU

GRU (Gated Recurrent Unit) is a simplified variant of LSTM proposed by Cho et al. [14], and its structure is shown in figure 4. In GRU, there are only two gates: reset gate and update gate. It uses an
update gate to replace the forget gate and input gate in LSTM, and combines the cell state and the hidden state. In the case of a small amount of calculation, it can also have the same effect as the standard LSTM. The specific formula of GRU cells is as follows:

$$z_t = \text{sigmoid}(W_z \cdot [h_{t-1}, x_t])$$  \hspace{1cm} (11)

$$r_t = \text{sigmoid}(W_r \cdot [h_{t-1}, x_t])$$  \hspace{1cm} (12)

$$\tilde{h}_t = \tanh(W \cdot [r_t \ast h_{t-1}, x_t])$$  \hspace{1cm} (13)

$$h_t = (1 - z_t) \ast h_{t-1} + z_t \ast \tilde{h}_t$$  \hspace{1cm} (14)

Among them, \(r_t\) represents the reset gate and \(z_t\) represents the update gate. The reset gate determines how much forgotten previous state, and whether to update the hidden state to a new state depends on the update gate.

**Figure 4. Structure of GRU cell.**

4. Experiments and Results

4.1. Data Sources

The data used in this experiment was provided by Guodian Zhishen Control Technology Co., Ltd. The data is the historical data of the 47-day operating unit and the data sampling frequency is one per second. More than ten factors such as fuel quantity, main steam flow, flue gas flow, and main steam pressure are selected to predict the main steam temperature.

4.2. Data Preprocessing

In this experiment, since all data is floating-point data, no data conversion operation is required. At the same time, because the data has a delay phenomenon, but the delay time of the relevant variable has been given, the delay is also removed. Next, the method of deleting empty rows is used to process missing values. Different parameters have different measurement scales. Therefore, all parameters are dimensionless, that is, the data is normalized. The normalization formula is as follows:

$$\text{result} = (\text{value} - \text{min\_value})/(\text{max\_value} - \text{min\_value})$$  \hspace{1cm} (15)

Among them, result is the normalized value, value is the data to be normalized, the maximum value of the parameter is used by \(\text{max\_value}\), the minimum value of the parameter is used by \(\text{min\_value}\). After normalizing the data, the data are all between 0 and 1. The convergence rate and accuracy of the model can be greatly improved.

4.3. Model Training and Parameter Settings

First build a basic autoencoder with a network structure of 12-4-12, and pass the data through the autoencoder to reduce the 12 influencing factors into 4 dimensions for GRU input. The input data is then arranged as a sample in 1200 groups in a row of time, that is, the time step is 1200. The first 600 groups are used to obtain the unit operating status, and the last 600 groups are used to predict the
value. Finally, the total data is divided into a training set, a validation set, and a test set according to a ratio of 8: 1: 1.

In the design of the GRU network structure, after repeated experiments and adjustments, it was finally determined that the number of layers was set to 3 layers with 64 nodes in each layer. The optimization method of the model adopted the Adam optimization algorithm. The initial learning rate was 0.001 and the batch size was 20. The number of training epoch is 5000.

This experiment uses Python language and is implemented by Tensorflow deep learning framework under Windows system.

4.4. Evaluation Indicators
This paper uses the accuracy and root mean square error as the evaluation index of the model. Calculate the error between the predicted value and the true value. The percentage of points with a statistical error of less than 5 to all points is the accuracy rate. The specific formula of the root mean square error is:

\[
\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_{true}(i) - y_{pred}(i))^2}
\]  

Among them, \( y_{true} \) represents the true value, \( y_{pred} \) represents the experimental prediction value, the prediction duration is represented by \( n \).

4.5. Experimental Results
In the experimental test phase, a random sample of the test set is selected, and RNN, LSTM, and GRU are simultaneously input. The last 600 points of the model output are predicted values. The model output is compared with the real historical data, and the graph is visually drawn to show. Figure 5 shows the RNN model. Figure 6 shows the LSTM model, and figure 7 shows the GRU model. The red line is the real historical data and the blue line is the predicted value.

The accuracy and root mean square error of the three models are shown in table 1.
Table 1. Comparison of model results.

|     | RMSE   | ACC    |
|-----|--------|--------|
| RNN | 9.798  | 36.68% |
| LSTM| 5.477  | 52.34% |
| GRU | 3.606  | 65.76% |

5. Conclusion

For the main steam temperature control system, this paper proposes a main steam temperature prediction model based on autoencoder and GRU. Experiments were performed using the 47-day historical data of the operating units provided by Guodian Zhishen Control Technology Co., Ltd. The comparison with the methods of LSTM and RNN proves that this method has higher accuracy for the prediction of main steam temperature.

The advantage of this method is that the multi-dimensional influence factors are reduced by the auto-encoder, which takes into account the influence of multiple factors on the main steam temperature, and reduce the amount of calculation of the model a lot, so the training is faster. Processing to make the predicted data more accurate.

This prediction method is of great significance to the control of the unit’s main steam temperature. In the future, the model will be further improved to find the optimal parameters and put into practical application as soon as possible.

References

[1] Li J and Fan H X 2019 China’s power development situation in 2018 and outlook for 2019 China Energy 41 (02) 22-27.
[2] Ali A, Saleh M A and Syed M I 2006 Predictive control of an integrated PV-diesel water and power supply system using an artificial neural network Renewable Energy 32 (8) 1426-1439.
[3] Wu B and Zhang L Y 2015 Application of fuzzy self-tuning PID Control in main steam temperature control Computer Simulation 32 (02) 387-390.
[4] Xiao Z, Yu R and Wang Q 2011 Design and simulation of main steam temperature system based on PID neural network Journal of System Simulation 23 (10) 2195-2199.
[5] Yule G U 1927 On a method of investigating periodicities in disturbed series, with special reference to Wolf’s sunspot numbers Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 226 (636-646) 267-298.
[6] Walker G 1931 On periodicity in series of related terms Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences 131 (818) 518-532.
[7] Box G E and Jenkins G M 1976 Time series analysis: Forecasting and control rev. ed Oakland, California, Holden-Day 31 (4) 238-242.
[8] Polhemus N W and Cakmak A S 1981 Simulation of earthquake ground motions using autoregressive moving average (ARMA) models 9 (4) 343-354.
[9] Torres J L, A. García, Blas M D, et al. 2005 Forecast of hourly average wind speed with ARMA models in Navarre (Spain) Solar Energy 79 (1) 65-77.
[10] Hochreiter S and Schmidhuber J 1997 Long short-term memory Neural Computation 9 (8) 1735-1780.
[11] Rumelhart D E 1986 Learning representations by back-propagating errors Nature 23.
[12] Lekamalie C L, Yang Y, Huang G B, et al. 2016 dimension reduction with extreme learning machine IEEE Transactions on Image Processing 25 (8) 1-1.
[13] Bengio Y 1994 Learning long-term dependencies with gradient descent is difficult IEEE Transactions on Neural Networks 5.
[14] Cho K, Van Merrienboer B, Gulcehre C, et al. 2014 Learning phrase representations using RNN encoder-decoder for statistical machine translation Computer Science.