OGRU: An Optimized Gated Recurrent Unit Neural Network

Xin Wang¹, Jiabing Xu², Wei Shi³*, and Jiarui Liu⁴

¹College of Computer Science and Technology, Harbin Engineering University, Harbin, Heilongjiang, 150001, China
²Yantai Research Institute, China Agricultural University, Yantai, 264670, China
³Beijing Aerospace Smart Technology Development Co., Ltd, Beijing, 100039, China
⁴School of Economics and Management, Harbin Engineering University, Harbin, Heilongjiang, 15001, China

*Corresponding author’s e-mail: shiwei142@163.com

Abstract. Due to the structural problem, the traditional neural network models are prone to problems such as gradient explosion and over-fitting, while the deep GRU neural network model has low update efficiency and poor information processing capability among multiple hidden layers. Based on this, this paper proposes an optimized gated recurrent unit (OGRU) neural network. The OGRU neural network model proposed in this paper improves information processing capability and learning efficiency by optimizing the unit structure and learning mechanism of GRU, and avoids the update gate being interfered by the current forgetting information. The experiment uses Tensorflow framework to establish prediction models for LSTM neural network, GRU neural network and OGRU neural network respectively, and compare the prediction accuracy. The results show that the OGRU model has the highest learning efficiency and better prediction accuracy.

1. Introduction

With the rapid development of the global information industry, artificial intelligence has become a hot spot for many scholars. Deep learning is one of machine learning and one of the methods to solve artificial intelligence problems [1]. Deep learning has overcome many computer problems that have been difficult to solve in the past, and has been widely used in speech recognition[2], machine translation[3,4], video motion recognition[5,6], image processing and other research tasks. As one of the important forms of deep learning, RNN has made great progress in the fields of natural language processing and time series prediction. However, RNN is difficult to achieve long-distance preservation of time series information. And there are problems such as gradient disappearance and gradient explosion [7], so it has not been widely used in early.

Based on this, Hochreiter et al. [8] proposed a long-Short Time Memory (LSTM) neural network model to improve the traditional RNN model. The network structure is more complex. By adding linear rules, redundant information can be selectively discarded. To a large extent, this has alleviated the problems of long-term dependence on information and the disappearance of gradients. RNN also has a variety of other variants, such as the Gated Recurrent Unit (GRU), which reduces the number of gating units on the LSTM model and optimizes the network structure, which is now widely used in industrial practice. However, GRU models still have problems such as slow convergence rate and low learning efficiency, resulting in too long training time and even under-fitting. To this end, Schuster et
al. [9] proposed a bidirectional recurrent neural network (BRNN), which can be learned in two time
dimensions, improving learning efficiency. Lei et al. [10] proposed a simple recurrent unit (SRU) to
improve the training speed of the GRU model, and proposed the model representation ability under the
large data volume. Oliva et al. [11] proposed a non-gating unit, the Statistical Recurrent Unit (TSRU),
which only needs to record the moving average of statistical data to maintain the long-term
dependence of data and achieve associative memory, greatly reducing the complexity of the model.

In order to improve the learning efficiency and prediction accuracy of the deep GRU neural
network model, this paper proposes an optimized Gated Recurrent Unit (OGRU) model, which
optimizes the learning mechanism of the standard GRU model gating unit. The output of the gate is
used to filter the current input information to avoid interference from redundant information. The
output of update gate is adjusted by the reset gate, which can speed up the convergence, and suppress
the gradient attenuation problem, thereby improving the learning efficiency.

2. The Standard GRU Neural Network and the OGRU Neural Network

The GRU neural network is a special variant of the recurrent neural network, which can maintain a
longer-term information dependence and has been widely used in industry. However, GRU still has
the disadvantages of slow convergence and low learning efficiency. So we proposed an optimized
gated recurrent unit (OGRU) neural network model. The OGRU model uses the reset gate to optimize
the learning mechanism of GRU, improving the learning efficiency and prediction performance.

2.1. The standard GRU neural network model

The LSTM neural network model is composed of three gate units, such as the forgetting gate, the input
gate and the output gate. The design of the gate unit is used to process the time series data. Although
the gradient disappears to a certain extent, the parameters are more likely to lead to training. Longer
time. The GRU neural network model is a variant of the LSTM neural network. It optimizes the
structure of the LSTM neurons and combines the three gating units of the LSTM into two gating units,
name the update gate and the reset gate. Therefore, the parameters of the GRU model are relatively
small, the training overhead is reduced, and the information dependency of a longer distance can be
maintained. The neuronal structure of the standard GRU is shown in Figure 1.

![Figure 1. The neuronal structure of GRU](image)

The GRU model consists of the input layer, the output layer and the implicit layers. The hidden
layers are composed of GRU neurons. The input data of the GRU neural network is the data at the
time t after data preprocessing. It should be noted that the input data is time series data. Suppose the
input sequence is \((x_1, x_2, ..., x_t)\), then at time t, the calculation formulas for the update gate, reset
gate, and standard GRU unit are as follows:

\[
\begin{align*}
    r_t &= \sigma \left( W_r \ast [h_{t-1}, x_t] \right) \\
    z_t &= \sigma \left( W_z \ast [h_{t-1}, x_t] \right)
\end{align*}
\]
\[ n_t = \tanh \left( W * \left[ r_t * h_{t-1}, x_t \right] \right) \] \hspace{1cm} (3)

\[ h_t = (1 - z_t) * h_{t-1} + z_t * n_t \] \hspace{1cm} (4)

\[ y_t = \sigma \left( W_o * h_t \right) \] \hspace{1cm} (5)

Where \( r_t \) represents the output of the reset gate at time \( t \); \( z_t \) represents the output of the gate at time \( t \); \( W_z \) represents the weight between the input and \( h_{(t-1)} \) in the update gate; \( W_r \) represents the weight between the input and \( h_{(t-1)} \) in the reset gate, where \( h_{(t-1)} \) represents the standard GRU unit output at time \( t-1 \); \( x_t \) represents the input at time \( t \); \( n_t \) represents a new candidate value vector created with the tanh layer at time \( t \), ie hidden state, and added in the current state; \( W \) represents the update gate's output \( z_t \) and the weight between the inputs; \( h_t \) represents the output of the standard GRU unit at time \( t \), used to update the current neuron state; the previous layer hidden state \( h_{(t-1)} \) and Multiply \((1-z_t)\), discard the redundant information, and add the product of \( z_t \) and \( n_t \) to form the hidden state \( h_t \) at the current \( t \) time; \( y_t \) represents the output of the GRU neural network at time \( t \), that is, the predicted result, and \( W_o \) represents the weight of \( h_t \); for sigmoid activation function, sigmoid and tanh are two commonly used neuron activation functions.

It can be seen from the above formula that the GRU model achieves long-distance preservation of key information by simplifying the number of gating units, continuously discarding redundant information, and utilizing the hidden state to store information dependencies.

2.2. The OGRU neural network model

Considering that GRU still has problems such as slow convergence rate and low learning efficiency, and the complex state of time series data, it is often impossible to discard enough redundant state information in one screening. Therefore, we propose the OGRU neural network, that is, the update gate of the GRU neural unit is improved, the \( x_t \) in the original update gate input is changed to \( x_t \) multiplied by \( r_t \), and the output of the reset gate is used to feedback adjust the update gate. By filtering the current input information \( x_t \) by the reset gate, the adverse effects caused by the redundant information are avoided to a greater extent, thereby accelerating the convergence speed and achieving the purpose of efficient learning. The deep OGRU neural network originates from the GRU neural network, and its neuron structure diagram is shown in Fig. 2.

Let us assume the input sequence to be \((x_1, x_2, ..., x_t)\), then update the gate at \( t \), reset the gate, and the standard OGRU unit output calculation formula is as follows:

\[ r_t = \sigma \left( W_r * \left[ h_{t-1}, x_t \right] \right) \] \hspace{1cm} (6)

\[ z_t = \sigma \left( W_z * \left[ h_{t-1}, x_t * r_t \right] \right) \] \hspace{1cm} (7)

\[ n_t = \tanh \left( W * \left[ r_t * h_{t-1}, x_t \right] \right) \] \hspace{1cm} (8)

\[ h_t = (1 - z_t) * h_{t-1} + z_t * n_t \] \hspace{1cm} (9)

\[ y_t = \sigma \left( W_o * h_t \right) \] \hspace{1cm} (10)

Among them, symbols such as \( z_t \) and \( r_t \) in the formula indicate the same meaning as standard GRU neurons. As shown in FIG. 2, the OGRU neuron is different from the GRU neuron in that, at the update gate \( z_t \), the \( r_t \) is multiplied and then multiplied with the previous time to hide the state weight, so that the reset gate re-screens the current input information \( x_t \), that is, The output of the reset gate is used to adjust the update gate to optimize the neuron structure. It can be seen from Fig. 2 and formula (7) that the neuron structure of the OGRU neural network is more reasonable than that of the GRU, and the hidden state at each moment can be simplified, and the gradient attenuation is suppressed to some extent. Therefore, the OGRU model can maintain a larger distance information dependency, and its learning efficiency and prediction accuracy are higher.
Figure 2. The neuronal structure of OGRU.

The deep OGRU neural network consists of the input layer, the output layer and the hidden layer. The hidden layers are composed of OGRU neurons. By optimizing the learning mechanism of the GRU model, the recursive transmission of information between neurons is promoted, and the information preservation ability is stronger.

3. Experimental verification

In this paper, a deep LSTM neural network prediction model, a standard GRU neural network prediction model and a deep OGRU neural network prediction model are established on the classical time series data set. By comparing the prediction accuracy of the above three models, the algorithm performance of the proposed OGRU model is verified. The experimental environment uses the deep learning framework tensorflow 1.10.0, the programming language is Python 3, the programming tool is PyCharm, the computer operating system is Ubuntu 16.4, the basic configuration is: CPU is Intel Core i5-6300 3.20GHZ, memory is 32G, graphics card is Nvidia GeForce GTX 980Ti.

3.1. Experimental datasets

Above all, we selected the Shampoo Sales Dataset (SSD), the Minimum Daily Temperatures Dataset (MDTD) in DataMarket datasets, and the Ozone Level Detection Dataset (OLDD) in the UCI datasets. Among them, the SSD contains a monthly sales volume of shampoo for a period of 3 years, and MDTD records the daily lower temperature of Melbourne City, Australia for 10 years (1981-1990), both of which are univariate time series data sets. The OLDD is a multivariate time series dataset that records ground-level ozone concentration data over a six-year period. The purpose is to predict whether it is an “ozone day”, which is a problem of time series classification prediction. The nature, number of features and number of samples of the datasets selected in this paper are different, which is beneficial to test the performance of each model.

Since there are many features in datasets and contain certain duplicate data, this will interfere with the model fitting of neural networks and waste much time. Therefore, the datasets need to be preprocessed, including dimensions reduction, cleaning and normalization. For each dataset after preprocessing, the first 80% is taken as the training set, and the remaining is taken as the verification set.

3.2. Process of establishing the prediction model

In the field of deep learning, the performance verification of the neural network models needs to establish a prediction model. The main processes include data preprocessing, training neural network and model verification. The flow chart is shown in Figure 3. This paper uses this modeling process to establish LSTM, GRU and OGRU neural network prediction models.
Beginning

Obtaining datasets and preprocessing (Dimension reduction, cleaning, normalization)

Training set

Training deep neural network

Establishing deep neural network prediction model

Validation set

End

Figure 3. The flow chart for establishing neural network prediction model

The preprocessing of the data set has been described in the previous section, and the remaining steps of the modeling are explained below. First, in order to make the experimental comparison more convincing, this paper unifies the structure of the LSTM, GRU and OGRU neural network models. We set all three types of network input and output layers to one layer, and the number of hidden layers is set to five. At the same time, the initial learning rate of the three kinds of neural networks is set to 0.5, and the weight distribution is based on random numbers.

After determining the algorithm parameters such as input layer, hidden layer and output layer neurons, the LSTM, GRU and OGRU neural network models are constructed in the TensorFlow environment, and the three preprocessed data sets are read separately for model training. Finally, in the verification model stage, the verification set is input, the output of the model is inversely normalized and compared with the actual value, and the experimental results are compared. The prediction accuracy of the OGRU model and the LSTM and GRU models under three data sets is obtained.

It should be pointed out that the mean of error ratio and sum of squares of error are used as the evaluation criteria of prediction accuracy. They are common indicators to measure the difference between actual values and predicted values, and can accurately reflect the prediction accuracy of the model. The error ratio mean and sum of error squares are shown in formulas (11) and (12):

\[
AE = \frac{1}{N} \sum_{i=1}^{N} \left( \frac{y_i' - y_i}{y_i} \right)
\]  

(11)

\[
SSE = \sum_{i=1}^{N} (y_i' - y_i)^2
\]  

(12)

Among them, AE and SSE represent the mean of error ratio and sum of error squares respectively; M is the total number of samples in the verification set, \(y_i'\) represents the predicted output, and \(y_i\) represents the actual output. The smaller AE and SSE are, the higher the prediction accuracy is.

3.3. Experimental result

After the training of three models is completed, the above three verification sets are input into each model respectively, and the prediction results of GRU, LSTM and OGRU models can be obtained.
Table 1. AE values of three prediction models in different datasets.

| Datasets | Neural network prediction models |  |
|----------|---------------------------------|---|
|          | LSTM | GRU | OGRU |  |
| SSD      | 0.06243 | 0.0287 | 0.0266 |  |
| MDTD     | 0.05159 | 0.0324 | 0.0249 |  |
| OLDD     | 0.08091 | 0.0498 | 0.0413 |  |
| Average value of AE | 0.06498 | 0.0370 | 0.0309 |  |

Table 1 shows the AE values of three models under the data sets of SSD, MDTD, OLDD, and table 2 shows the SSE values of three models. From Table 1 and Table 2, compared with LSTM and GRU, the AE and SSE values of OGRU neural network prediction model remain the lowest, indicating that its prediction accuracy is high and its prediction performance is more stable.

Figure 4. Comparison of AE and SSE values of three models.

Figure 4 shows the trend of average AE and SSE values of the three models under different data sets. The average AE value of OGRU was 52.4% lower than LSTM and 16.4% lower than GRU, while the average SSE value of OGRU was 36.7% lower than LSTM and 16.5% lower than GRU. The results show that the generalization ability of OGRU model is stronger.

Figure 5. Predictive output and actual output of three models under verification set.
Figure 5 shows the predicted and actual output of three models on the MTDTD verification set for a certain period of time. Among them, the OGRU model curve is the closest to the actual output curve, which further verifies that the proposed OGRU model has higher learning efficiency.

Table 2. SSE values of three prediction models in different datasets.

| Datasets | LSTM | GRU | OGRU |
|----------|------|-----|------|
| SSD      | 337.0| 281.4| 195.1|
| MTDTD    | 392.5| 276.0| 243.7|
| OLD      | 485.6| 364.3| 330.8|
| Average value of SSE | 405 | 307.2 | 256.5 |

4. Conclusions

In this paper, an optimized gated recurrent unit neural network model is proposed to solve the problems of low learning efficiency and prediction accuracy of standard GRU neural network. Experiments on three classical time series data sets show that compared with traditional LSTM and standard GRU prediction models, the average AE value of the OGRU neural network model proposed in this paper decreases by 16.4% - 52.4%, and the average SSE value decreases by 16.5% - 36.7%. Therefore, OGRU neural network model can fully capture the information dependence relationship in time series data, has high prediction accuracy, and its generalization ability is more prominent. It is an excellent variety of cyclic neural network.

References

[1] Wu, J., Long, J., & Liu, M. (2015). Evolving rbf neural networks for rainfall prediction using hybrid particle swarm optimization and genetic algorithm. Neurocomputing, 148, 136-142.

[2] Graves, A., Mohamed, A. R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks.

[3] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks.

[4] Meng, F., Lu, Z., Tu, Z., Li, H., & Liu, Q. (2015). Neural transformation machine: a new architecture for sequence-to-sequence learning.

[5] Donahue, J., Hendricks, L. A., Guadarrama, S., Rohrbach, M., Venugopalan, S., & Saenko, K., et al. (2015). Long-term Recurrent Convolutional Networks for Visual Recognition and Description. AB initio calculation of the structures and properties of molecules. Elsevier.

[6] Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. In Proceedings of the IEEE international conference on computer vision (pp. 4489-4497).

[7] Chatterjee, S., & Bandopadhyay, S. (2012). Reliability estimation using a genetic algorithm-based artificial neural network: An application to a load-haul-dump machine. Pergamon Press, Inc.

[8] Hochreiter, S., & Schmidhuber, Jürgen. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780.

[9] Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. IEEE Transactions on Signal Processing, 45(11), 2673-2681.

[10] Lei, T., Zhang, Y., Wang, S. I., Dai, H., & Artzi, Y. (2017). Simple recurrent units for highly parallelizable recurrence.

[11] Oliva, J. B., Poczcos, B., & Schneider, J. (2017). The statistical recurrent unit.