Cloud computing resource elasticity scaling method based on neural network time series prediction

Lingxiao Wang*, Jian Zhang
Faculty of Information Technology, Beijing University of Technology Beijing, China

*Corresponding author: wangle92@163.com

Abstract. With the development of cloud computing, the requirements for dynamic distribution of cloud computing according to the real-time load are getting higher and higher. It is necessary to predict the load of cloud computing resources to achieve the goal of real-time elastic scaling. Cloud computing load will constantly change with time and actual demand. Therefore, lstm-attention model is used to collect data and model and predict cloud computing resource load for a long time, so as to achieve accurate prediction of cloud computing resource load in the future. It can be concluded that the lstm-attention model performs well in cloud computing resource load prediction.

1. Introduction
Cloud computing[1] is an emerging computing mode with the development of the Internet. It combines a large number of hardware equipment and software resources with the high-speed transmission capacity of the Internet and provides them to all kinds of users in the form of resource sharing. The biggest advantage of this calculation mode is elastic scaling[2], which can be scaled according to actual needs. This can not only effectively improve the utilization of resources, but also greatly reduce the use cost of software and hardware. But the cloud computing resources is real-time changes with time and the needs of users, and the existing cloud computing elastic expansion model based on threshold to trigger, relative to the real-time change of load the pattern is not very good forecast resources under a moment load, so the whole cloud computing resources utilization rate is not high, many resources are idle most of the time, and maintain the normal operation of cloud computing resources is also need to consume huge cost, so it is necessary to cloud computing resources load for accurate prediction to improve the flexibility of cloud computing to performance.

Cloud computing resource load changes over time, generating a series of time-based resource usage data -- time series data. However, there are various prediction methods based on time, such as the traditional time series prediction model based on autoregressive models[3], there are machine learning methods like xgboost[4], random forest[5], there are also BP neural network prediction models[6], but autoregressive models require a lot of manual work and are not very usable, for the machine learning prediction model, a large number of features are needed to train the model well to achieve the final prediction effect. However, the cluster resource load has relatively few features, only CPU and memory utilization based on time changes, which cannot give full play to the advantages of machine learning. As BP neural network has no memory function, it cannot accurately predict the future load based on the past time series data. With the development of deep learning technology, it provides a new way to predict the load of cluster resources. Recurrent neural network is used to predict the time series data. Recurrent
Neural Network can store the information of the past period and participate in the data calculation of the next stage. Based on the above analysis, this paper uses short and long memory network to predict the cloud computing resource load.

2. Load prediction algorithm based on lstm-attention model

2.1. Recurrent Neural Network

Recurrent Neural Network [7] is a kind of widely used network structure, based on input layer, hidden layer and output layer as the main structure of the three layer network model, which have different weights between layers. Through the input of a sequence, the three-layer network model is used to output a sequence, and the gradient descent method is used to optimize the model. The structure of the cyclic neural network model is shown in figure 1.

![Fig 1. Recurrent Neural Network Model](image)

In figure 1, \(t\) is the time, \(x_t\) is the input at time \(t\), \(h_t\) is the hidden state at time \(t\), \(o_t\) is the output at time \(t\), \(U, W, V\) represent the weights of the output layer, the hidden layer and the output layer respectively, and the output expression at time \(t\) is:

\[
h_t = f(Ux_t + Wh_{t-1})
\]

\[
o_t = g(Vh_t) = g(Vf(Ux_t + Wh_{t-1}))
\]

Where \(f, g\) are activation functions of two layers respectively.

2.2. Long Short Term Memory

LSTM [8] is a kind of Recurrent Neural Network, which is an optimization of RNN. It solves the problem that the gradient disappears in the recurrent neural network and the prediction results are only related to the recent data. Compared with the previous recursive neural network, LSTM has a door structure inside each logical unit, namely the forgotten door, the input door and the output door, there is also a state unit for memorizing information at different times of the whole training process, which exists in a straight line on it alone and will not be affected by other factors, three different gate structures are used to add or delete information at different times to the state unit. Each gate is composed of the transformation function sigmoid and a bitwise multiplication operation, selectively letting useful information through. The model structure of LSTM is shown in figure 2.
Fig 2. Long Short Term Memory Model

As shown in figure 2, $x_t$ is the input at time $t$, $h_t$ is the hidden state, and $y_t$ is the output at time $t$. The entire LSTM consists of three layers, namely the input layer, the hidden layer and the output layer. Each layer is composed of multiple identical modules in series. Here, if one of the modules is enlarged, it can be seen that it is composed of four parts. For weight reuse $W$ between different layers or between different hidden states, for example, $W_{xh}$ represents the weight matrix between $x$ layer and $h$ layer, and $W_{hh}$ represents the weight matrix between different two hidden states.

Note: * in this article refers to the multiplication of the corresponding elements.

1. Input layer and hidden layer are connected by transformation function.

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

2. The hidden layer and the output layer are connected by the transformation function.

$$\hat{y}_t = g(W_{hy}h_t + m)$$

3. LSTM determines what useless information needs to be removed from the memory unit through the forgetting gate.

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$

4. LSTM determines the need to add some useful information to the memory unit through the input gate and candidate memory cells.

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$

$$C_t = \text{tanh}(W_{xc}x_t + W_{hc}h_{t-1} + b_c)$$

5. Update of memory unit.


\[ C_t = f_t \cdot C_{t-1} + i_t \cdot C_t \tag{8} \]

(7) LSTM determines the output information and the output of the hidden layer state through the output gate.

\[ o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{9} \]

\[ h_t = o_t \cdot \tanh(C_t) \tag{10} \]

The initial model of time series prediction is completed by repeating the above process for many times. Next, the model is optimized by back propagation. We define the loss function here as the mean square error:

\[ L = \sum_{i=1}^{n} (y_i - o_i)^2 \tag{11} \]

Train the model by using the minimized target function.

The Attention mechanism was added to further optimize the network model, the output data of the hidden layer was temporarily stored by LSTM, the Attention mechanism is used to allocate Attention weight to the hidden state to form the intermediate value. The output of the final result is selected by calculating different attentional weights. The LSTM-Attention model is shown in figure 3.

![Fig 3. The Diagram Of LSTM-Attention model](image-url)

The calculation formula is:

\[ C_t = \sum_{i=0}^{n} m_{ti} f(x_i) \tag{12} \]

\[ m_{ti} = \frac{\exp(e_{ti})}{\sum_{j=1}^{n} \exp(e_{tj})} \tag{13} \]

Where, the parameter \( m_{ti} \) represents the probability to indicate the importance of \( h_t \) to \( C_t \), and \( e_{ti} \) represents the matching degree between each element. The Attention mechanism was added to better predict the effect of the input sequence on the results.
3. Experimental and Result

3.1. Preparation of Experimental Data
Collect CPU and memory data through cluster monitoring. The load was recorded every 10 minutes and the average value was calculated to obtain 2000 pieces of data. The final data set was obtained by normalizing the data. The whole data set is divided into 4 training sets and 1 validation set by means of cross validation to train and verify the model.

3.2. Analysis of experimental results

Fig 4. The Diagram of Experimental Result

Fig 5. The Diagram of Experimental Result

Figures 4 and 5 show the predicted results of the lstm-attention model for CPU and memory utilization. It can be seen that the actual value and the predicted value still fit well. The above results show that LSTM-Attention can accurately predict the load of cluster resources in the future when it forecasts the time series data.
4. Conclusion
This paper proposes a model based on LSTM - Attention to key load indicators to predict the cluster resources, and through the experiment to compare the real value and predictive value, the fitting degree between the can see LSTM - it is desirable that the Attention of the whole fitting degree, this article preliminary demonstrated the feasibility of the load forecast for cloud computing resources, in a follow-up study to improve prediction accuracy and predict in actual working environment, in order to improve the practicability of this method.

Reference
[1] VasanthaAzhagu A K, Gnanasekar J M. Cloud Computing Overview, Security Threats and Solutions-A Survey[C]//Proceedings of the International Conference on Informatics and Analytics. 2016: 1-6.
[2] Ting Chen, Yongjian Wang, Yinan Ren, Cheng Luo, Depei Qian, and Zhongzhi Luan. 2009. R-ECS: reliable elastic computing services for building virtual computing environment. In Proceedings of the 2nd International Conference on Interaction Sciences: Information technolog, Culture and Human (ICIS '09). ACM, New ork, NY, USA, 866-872.
[3] Benjamin, Michael A., Robert A. Rigby, and D. Mikis Stasinopoulos. "Generalized autoregressive moving average models." Journal of the American Statistical association 98.461 (2003): 214-223.
[4] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A scalable tree boosting system." Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016.
[5] Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.
[6] Li, Jing, et al. "Brief introduction of back propagation (BP) neural network algorithm and its improvement." Advances in computer science and information engineering. Springer, Berlin, Heidelberg, 2012. 553-558.
[7] Mikolov, Tomáš, et al. "Recurrent neural network based language model." Eleventh annual conference of the international speech communication association. 2010.
[8] NHUAN T, THANG N, BINH H, et al. A multivariate fuzzy time series resource forecast model for clouds using lstm and data correlation analysis[J]. Procedia Computer Science, 2018, 126: 636-645.