An Attention-based Recurrent Neural Network for Resource Usage Prediction in Cloud Data Center

Huidan Xi¹, Cheng Yan², Huixi Li*³ and Yinhao Xiao*³
1 School of Information Engineering, Hunan Industry Polytechnic, Changsha, 410208, China
2 School of Computer and Information, Qiannan Normal University for Nationalities, Duyun, 558000, China
3 School of Information Science, Guangdong University of Finance and Economics, Guangzhou, 510320, China
*Corresponding author’s e-mail: lihuixi@gdufe.edu.cn, xyh3984@gmail.com

Abstract. To manage the computing resources efficiency is crucial for the Cloud data centers (CDCs). By predicting the resource usages of certain virtual machine (VM), the workload could be balanced before the resource overusing occurs. In this paper, we propose an attention-based time series prediction model, which contains an encoder of input attention mechanism and a decoder of a temporal attention mechanism, to optimize the efficiency of cloud data center. Experiments on Alibaba CDC VM trace dataset demonstrates that our proposed methods can outperform the classic LSTM method, especially when the resource usage data is lack of relationship.

1. Introduction
In Cloud data centers (CDCs), high energy efficiency is achieved through flexible allocation of computing resources. Due to the dynamic changes of user behavior, it is a huge challenge to optimize the resource allocation according to the user's real-time demand for physical resources. Therefore, Cloud service providers (CSPs) can implement dynamic server consolidation by predicting various devices' usage of computing resources in the CDC. These devices range from virtual machines (VMs), containers to physical hosts.

Due to the diversity and complexity of business, the demands of various computing resources by VMs or containers fluctuate drastically over time and lacks obvious rules in CDC. At this time, traditional prediction methods, such as automatic regression, cannot accurately capture the workload changes of modern CDCs. When the information is related in time series, the recurrent neural network (RNN) has been proved to be able to predict the state information for a period of time in the future more accurately than traditional prediction methods in many situations. In CDC, changes of device's workload are obviously related in time sequence. In essence, it is a time series-based behavior.

Time series forecasting algorithms have been widely used in many fields, such as stock price forecasting, weather forecasting and action forecasting. Autoregressive moving average (ARMA)[1] mechanism has shown effectiveness in many real scenarios, but it lost the advantages when describing nonlinear relationships. In order to solve this problem, a model[2] that combines linear regression and wavelet neural network is proposed to predict the short-term future workload. However, this method faces the problem of large errors when predicting the workload for a long period of time in the future.
Kumar et al. [3] proposed a prediction model that integrates artificial neural networks and adaptive differential evolution algorithms. Compared with linear regression and backpropagation-based methods, this model exhibits higher accuracy. [4] proposed a cluster-based learning method to improve the accuracy of prediction, but this method is difficult to determine the appropriate learning rate. In [5], based on the $k$-NN method, the author proposes two cooperative filtering techniques to allow the workload prediction model to achieve better performance on multi-core systems.

![Figure 1. CPU trace of m_633.](image1.png)

![Figure 2. Memory trace of m_633.](image2.png)

![Figure 3. I/O trace of m_633.](image3.png)
Recurrent neural network (RNN) [6] emphasizes the connectivity of neurons between hidden layers in order to effectively deal with sequence problems by using historical memory in the neural network. In the past few years, RNN models have been used to solve the workload prediction problem in Cloud computing. Zhang et al. [7] proposed an RNN-based model to improve the accuracy of predicting workloads. Similarly, [8, 9] used a classic recurrent neural network structure to predict the workload of the cloud data center in the future. Analyzing the conclusions of the above research, it can be concluded that the recurrent neural network has a better performance in solving the prediction problem of short-term dependence. However, the research in [10, 11] shows that RNN cannot effectively guarantee the solution of long-term prediction problems. This is caused by the fact that traditional recurrent neural networks cannot solve the problem of gradient disappearance during training. When the distance between the information and the predicted value increases, the recurrent neural network will lose its ability to connect and use meaningful information. This problem is also known as the long-term dependency problem. In order to solve this problem, the improvement of recurrent neural network, long short-term memory network (LSTM) [6] appeared. Song et al. [12] used LSTM to predict the workload of the server, which is a great improvement compared to the method based on the traditional recurrent neural network proposed by them [13]. An LSTM model using correlation learning was developed to capture the relationship between different computing resource indicators and predict future workloads more accurately [14]. Compared with LSTM, GRU [15] can converge faster and has fewer parameters to set. Chen et al. [16] proposed a prediction method based on automatic decoder and GRU to predict the workload of VMs in cloud data centers. They first use the
sparse encoder to extract the features of historical data, and then make predictions based on the GRU model.

In summary, most of the existing RNN-based prediction methods are based on the traditional recurrent neural network structure, so it is difficult to avoid the problem of gradient disappearance, and it is unable to capture long-term memory dependence in historical workloads. In order to solve this problem, Bahdanau et al. [17] proposed an attention-based encoder-decoder network to select parts of hidden states through the time steps. Moreover, in a production environment, the usage of a certain resources by the VM is not independently related to the resource itself. For example, when a VM uses a CPU to perform computing tasks, it usually uses a large amount of memory to store intermediate data at the same time. Fig. 1–5 shows such a pattern, where the data comes from the trace of a VM (No. m_633) in the Alibaba Cloud data center. Fig. 1 shows the CPU usage of the VM, and Fig. 2 shows the memory usage of the VM. It can be seen that there is a strong correlation between the two. Fig.3–5 demonstrate the usages of disk I/O and network in/out of the VM, respectively. Therefore, it is worth considering to predict the CPU usage by using the trace of the VM’s use of multiple resources.

Based on this, this paper proposes a multi-level attention-based RNN to predict the amount of CPU resource used by the VMs in CDC. One time step is divided into two folds. First, by referring to the previous encoder hidden state, we extract the relevant driving series. Then, we select relevant encoder hidden states. Each fold is deployed with the attention mechanism. The above mentioned two attention models are integrated within an LSTM network and can adaptively select the most relevant input features as well as capture the long-term temporal dependencies in the time series. To evaluate the efficiency of the proposed model, we compare it with classic LSTM mechanism by using Alibaba CDC VM trace dataset [18].

2. Attention-based LSTM

In this section, first we introduce the used notation of the study, and then present attention-based LSTM time series workload prediction in CDC.

2.1. Notation and Problem Description

Given \( n \) driving series which presents \( n \) resource usage traces, i.e., \( X = (x^1, x^2, ..., x^n)^T = (x_1, x_2, ..., x_T) \), where \( T \) is the window size. We use \( x^k = (x^k_1, x^k_2, ..., x^k_T)^T \) to present a \( T \) length driving series and use \( x_t = (x^1_t, x^2_t, ..., x^n_t)^T \) to denote a vector of exogenous input series at time \( t \).

Given the previous values of the target series \( (y_1, y_2, ..., y_{T-1}) \) and the past and current values of \( n \) driving series \( (x_1, x_2, ..., x_T) \), our problem is to learn a nonlinear mapping to the current value of the target series \( y \): \( \hat{y}_t = F(y_1, y_2, ..., y_{T-1}, x_1, x_2, ..., x_T) \), where \( F(\cdot) \) is the target nonlinear mapping function.

2.2. Model

The leveraged attention-based mechanism is two-fold. In the encoder, an attention module is used to adaptively select the relevant series. Then in the decoder, the relevant encoder hidden states is selected via another attention module.

2.3. Encoder

For workload time series prediction, given the input workload sequence \( X = (x_1, x_2, ..., x_T) \), the encoder is to learn a mapping from \( x_t \) to \( h_t = f_1(h_{t-1}, x_t) \), where \( h_t \) is the hidden state of the encoder at time \( t \), and \( f_1 \) is a non-linear activation function. We use a GRU as a memory cell with the state \( s_t \) at time \( t \). Accessing it is controlled by three gates: forget gate \( f_t \), input gate \( i_t \) and output gate \( o_t \). The following describes the update of a GRU:
\[ f_t = \sigma(W_f[h_{t-1}; x_t] + b_f) \]
\[ i_t = \sigma(W_i[h_{t-1}; x_t] + b_i) \]
\[ o_t = \sigma(W_o[h_{t-1}; x_t] + b_o) \]
\[ s_t = f_t \odot s_{t-1} + i_t \odot \tanh(W_o[h_{t-1}; x_t] + b_s) \]
\[ h_t = o_t \odot \tanh(s_t), \]

where \([h_{t-1}; x_t]\) is the concatenation of the previous hidden state \(h_{t-1}\) and the current input \(x_t, W_f, W_i, W_o, b_f, b_i, b_o, b_s\) are learning parameters, \(\sigma\) is a logistic function, and \(\odot\) is an element-wise multiplication.

Given the \(k\)-th input driving series \(x^k = (x^k_1, x^k_2, ..., x^k_T)^T\), an input attention mechanism is constructed based on a deterministic attention model by referring the previous hidden and the cell states with \(e_t^k = v^k_d \tanh(W^k_d[h_{t-1}; s_{t-1}] + U^k_d x^k)\) and \(a_t^k = \frac{\exp(e^k_t)}{\sum_{i=1}^{n} \exp(e^k_i)}\), where \(v^k_d, W^k_d\) and \(U^k_d\) are learning parameters, \(a_t^k\) is the weight that measures the importance of the \(k\)-th input feature at time \(t\). With the weight, the driving series are extract with \(\tilde{x}_t^k = (a_t^1 x_t^1, a_t^2 x_t^2, ..., a_t^n x_t^n)^T\). We then update the hidden state at time \(t\) as \(h_t = f_1(h_{t-1}, \tilde{x}_t^k)\), where \(f_1\) is a GRU.

### 2.4. Decoder

To predict \(y_T\), we also use an attention mechanism in the decoder to select relevant encoder hidden states. The procedure is very similar to the encoder. First, we calculate the attention weight with \(l_t^i = v^i_d \tanh(W^i_d[d_{t-1}; c_{t-1}] + U^i_d h_t)\) and \(\beta^i_t = \frac{\exp(l^i_t)}{\sum_{j=1}^{T} \exp(l^j_t)}\), where \(v^i_d, W^i_d\) and \(U^i_d\) are learning parameters, \(\beta^i_t\) is the weight that measures the importance of \(i\)-th encoder hidden state. The context vector \(c_t = \sum_{i=1}^{T} \beta^i_t h_i\) is the weighted sum of all encoder hidden states. Then, all such vectors are combined with the given series \((y_1, y_2, ..., y_{T-1})\) as \(\hat{y}_{t-1} = w^T[y_{t-1}; c_{t-1}] + b\), where \([y_{t-1}; c_{t-1}]\) is a concatenation of \(y_{t-1}\) and \(c_{t-1}\), \(w\) and \(b\) map the concatenation to the decoder input's size. After this, we can obtain \(d_t = f_2(d_{t-1}, \hat{y}_{t-1})\), where \(f_2\) is a GRU. And then we have

\[ f_t' = \sigma(W_f'[d_{t-1}; \hat{y}_{t-1}] + b_f') \]
\[ i_t' = \sigma(W_i'[d_{t-1}; \hat{y}_{t-1}] + b_i') \]
\[ o_t' = \sigma(W_o'[d_{t-1}; \hat{y}_{t-1}] + b_o') \]
\[ s_t' = f_t' \odot s_{t-1}' + i_t' \odot \tanh(W_o'[d_{t-1}; \hat{y}_{t-1}] + b_s') \]
\[ d_t = o_t' \odot \tanh(s_t'). \]

### 2.5. Hence

\[ y_T = F(y_1, ..., y_{T-1}, x_1, ..., x_T) = v_T^W [d_T; c_T] + b_w \]
3. Evaluation

3.1. Dataset and Evaluation Metrics

we use Alibaba CDC dataset for empirical studies. This dataset contains the traces of 4,023 VMs in the Alibaba CDC. ALibaba CDC collected the usage of resources in every 5 minutes (a time slice) and the trace of a VM contains 60,000 time slices. Their usages of resources are roughly divided into three patterns. In pattern $a$, the resources are used periodically in long term and with sharp fluctuations in short term. An example is shown in Fig.6, and its corresponding index in Alibaba CDC dataset is $m_{1392}$. In pattern $b$, the resources are used periodically in long term and without sharp fluctuations in short term. An example is shown in Fig.7, and its corresponding index in Alibaba CDC dataset is $m_{225}$. Pattern $c$ is lack of regularity in the use of resources. An example is shown in Fig.8, and its corresponding index in Alibaba CDC dataset is $m_{3}$.

We compare the proposed attention-based method against the classic LSTM method. To measure the effectiveness of the two methods for CPU workload prediction, we use mean absolute error (MAE).
3.2. Results

We use the traces of \( m_{1392}, m_{225} \) and \( m_3 \) to evaluate the performance. In the experiment, the first 50\% data of certain VM's trace are leveraged for training, then 20\% of them are used as validation set. The last 30\% data (17,500) are used as the test set. Fig. 9, 10, 11 demonstrate the comparisons of the predictions results by our methods and the real traces of the VMs, respectively.

![Figure 9. Comparison of the predictions results by our methods and the real trace of pattern a.](image-url)
Figure 10. Comparison of the predictions results by our methods and the real trace of pattern b.

Figure 11. Comparison of the predictions results by our methods and the real trace of pattern c.

Table 1. Prediction MAE results ($\times 10^{-2}$%) over Alibaba CDC dataset (pattern a, pattern b, and pattern c)

|            | pattern a | pattern b | pattern c |
|------------|-----------|-----------|-----------|
| Our attention-based method | 1.58      | 0.66      | 3.63      |
| LSTM       | 1.81      | 0.97      | 7.51      |

In Table 1, we observe that the MAE of our proposed method is better than that of LSTM in all three patterns. For pattern a and pattern b, which present obvious periodical regularity, LSTM can capture the relationship between the past and the future to some extent. For pattern c, which is lack of obvious regularity, our attention-based prediction method shows its high effectiveness.

4. Conclusion

In this paper, we presented an attention-based time series prediction model, which contains an encoder of input attention mechanism and a decoder of a temporal attention mechanism, to optimize the efficiency of cloud data center. Experiments on Alibaba CDC VM trace dataset demonstrates that our
proposed methods can outperform LSTM method, especially when the resource usage data is lack of relationship.

This paper is focus on the prediction of CPU usage. In the future, our work can take the multiple resource usage prediction into consideration. CPU is no longer the only primary computing resource in today's CDC. The usage of memory, disk I/O, and network must be accurately predicted in the future.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.62002067), Natural Science Foundation of Education of Guizhou Province (No.KY[2017]351) and the Department of Education of Guangdong Province (No.2020KTSCX039).

References

[1] D. Asteriou and S. G. Hall, “Arima models and the box–jenkins methodology,” Applied Econometrics, vol. 2, no. 2, pp. 265–286, 2011.
[2] X. Tang, X. Liao, J. Zheng, and X. Yang, “Energy efficient job scheduling with workload prediction on cloud data center,” Cluster Computing, vol. 21, no. 3, pp. 1581–1593, 2018.
[3] J. Kumar and A. K. Singh, “Workload prediction in cloud using artificial neural network and adaptive differential evolution,” Future Generation Computer Systems, vol. 81, pp. 41–52, 2018.
[4] Y. Yu, V. Jindal, F. Bastani, F. Li, and I.-L. Yen, “Improving the smartness of cloud management via machine learning based workload prediction,” in 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), vol. 2. IEEE, 2018, pp. 38–44.
[5] S. Salaria, A. Drozd, A. Podobas, and S. Matsuoka, “Predicting performance using collaborative filtering,” in 2018 IEEE International Conference on Cluster Computing (CLUSTER). IEEE, 2018, pp. 504–514.
[6] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, Deep learning. MIT press Cambridge, 2016, vol. 1, no. 2.
[7] W. Zhang, B. Li, D. Zhao, F. Gong, and Q. Lu, “Workload prediction for cloud cluster using a recurrent neural network,” in 2016 International Conference on Identification, Information and Knowledge in the Internet of Things (IIKI). IEEE, 2016, pp. 104–109.
[8] Z. Huang, J. Peng, H. Lian, J. Guo, and W. Qiu, “Deep recurrent model for server load and performance prediction in data center,” Complexity, vol. 2017, 2017.
[9] M. Duggan, K. Mason, J. Duggan, E. Howley, and E. Barrett, “Predicting host cpu utilization in cloud computing using recurrent neural networks,” in 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST). IEEE, 2017, pp. 67–72.
[10] H. Shi, M. Xu, and R. Li, “Deep learning for household load forecasting—a novel pooling deep rnn,” IEEE Transactions on Smart Grid, vol. 9, no. 5, pp. 5271–5280, 2017.
[11] W. Kong, Z. Y. Dong, Y. Jia, D. J. Hill, Y. Xu, and Y. Zhang, “Short-term residential load forecasting based on lstm recurrent neural network,” IEEE Transactions on Smart Grid, vol. 10, no. 1, pp. 841–851, 2017.
[12] B. Song, Y. Yu, Y. Zhou, Z. Wang, and S. Du, “Host load prediction with long short-term memory in cloud computing,” The Journal of Supercomputing, vol. 74, no. 12, pp. 6554–6568, 2018.
[13] Q. Yang, Y. Zhou, Y. Yu, J. Yuan, X. Xing, and S. Du, “Multi-step-ahead host load prediction using autoencoder and echo state networks in cloud computing,” The Journal of Supercomputing, vol. 71, no. 8, pp. 3037–3053, 2015.
[14] S. Kumar, N. Muthiyani, S. Gupta, A. Dileep, and A. Nigam, “Association learning based hybrid model for cloud workload prediction,” in 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, 2018, pp. 1–8.
[15] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” arXiv preprint arXiv:1412.3555, 2014.
[16] Z. Chen, J. Hu, G. Min, A. Y. Zomaya, and T. El-Ghazawi, “Towards accurate prediction for high-dimensional and highly-variable cloud workloads with deep learning,” IEEE Transactions on Parallel and Distributed Systems, vol. 31, no. 4, pp. 923–934, 2019.

[17] Bahdanau, D., Chorowski, J., Serdyuk, D., Brakel, P., & Bengio, Y. (2016, March). End-to-end attention-based large vocabulary speech recognition. In 2016 IEEE international conference on acoustics, speech and signal processing (ICASSP) (pp. 4945-4949). IEEE.

[18] C. Lu, K. Ye, G. Xu, C.-Z. Xu, and T. Bai, “Imbalance in the cloud: An analysis on alibaba cluster trace,” in 2017 IEEE International Conference on Big Data (Big Data). IEEE, 2017, pp. 2884–2892.