A Deep Recurrent Network for Web Server Performance Prediction

S.Umamaheswari, Department of Computer Application,

Mr. R. Sathish Kumar, Assistant professor,

Krishnasamy College of Engineering and Technology,

Cuddalore.

ABSTRACT

Internet is growing day by day so it is important to increase the performance of a web server. Web server is a server software or hardware dedicated to run software that can serve contents to the World Wide Web. Recurrent neural network(RNN) has been widely applied to many sequential tagging tasks such as natural language process(NLP), and it has been proved that RNN works well in those areas. Classical methods focus on building relation between performance and time domain, which can't capture the essence of web server performance. In this paper, we analyze the log of nginx web servers which contains user's url access sequence, and predict the performance of the servers by using RNN-LSTM. Experiment result shows that our model gets a good performance in predicting web server performance on the data set which has been deployed in online service.

Key Words: Deep recurrent network, Web server, recurrent natural network, LSTM…

1.INTRODUCTION

The function of web server is to store, process and deliver web pages to clients. Today the number of users of internet is increasing day by day and all fields are automated so it is so important to predict the performance of the web server hence to increase its performance. There are multiple ways existing to predict web server performance. The performance of a web server can be predicted by using its response time, response time is the time needed to get the first output to the query by the client. As the response time decreases web server performance increases. The GEO LIGO is an existing system to predict the performance of the web server which is only 33% performance improvement and another one is AADMLSS, for which the result is only 10%

accurate. The performance of web server can be predicted using a recurrent neural network; a recurrent neural network is a class of artificial neural network where connections between nodes form a directed graph along a sequence.

2.EXISTING SYSTEM

One is focusing on building the relation between performance and time, such as neural network(MLP) and linear regression weighted multivariate linear regression(MVLR) and recurrent neural network(RNN) are used. Another one does not consider the sequential effects, and predict the performance by analyzing the workload.

2.1 DISADVANTAGES

Both of these two kinds of method may not explore the essence of the problem. Since the fluctuation of server's performance is caused by user's url request sequence .

3. PROPOSED SYSTEM

Our work is the first one to apply RNN-LSTM network to predict the performance of Web servers.We proposed to investigate the relation between users' url requests sequence and web sever performance, which previous researches didn't pay much attention to.

3.1 ADVANTAGES

Model achieves a good performance and generalization on predicting the performance of nginx web sever. Since the fluctuation of server's performance is caused by user's url request sequence .

4. IMPLEMENTATION

The object design consists of 3 modules.

MODULES

4.1 Datasets

4.2 Nginx log files

4.3 Web Server Performance

MODULES DESCRIPTION

4.1. Datasets

The data set for evaluating the performance of our model contains the log files of 191 web servers nodes in one day. The web servers are set up using nginx and have been deployed in production environment, which means the records in log files are real.

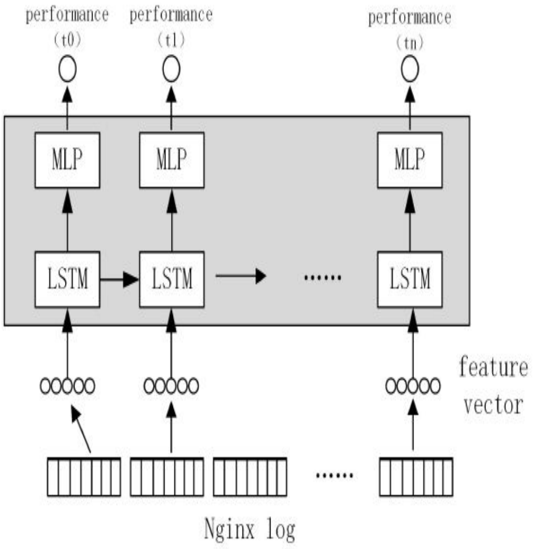
The data set of this model is log files. The servers are made with nginx and can be used for prediction because the data in the files are real. The memory capacity is limited hence we choose to use log files of several random nodes which has the biggest log files to train the model. The urls of user requests are filtered and Find the valid request.The data set of valid request is formed in chronological order

4.2.Nginx log files

Nginx log files are the raw data source of our model, and the main idea of our model is predicting the web server performance by analyzing the log files. Each url request has an one-to-one id in the form of integer which is stored in a dictionary. The dictionary is generated by collecting all the unique url request string in the whole data set. Using this id, the url requests during a time window can be abstracted to a vector with d-dimensional, every dimension of which means the number of times user request the url during the time-window.

4.3.Web server Performance

Three performance including request error rate, throughput and request delay are simulated. We use sum squared resid (ˆ yi − yi)2 to measure the efficiency of the model. The features of the network which has the best result on the valid set are saved, and then we perform a final test to see the performance of our network by using the test set. environment, which means the records in log files are real.



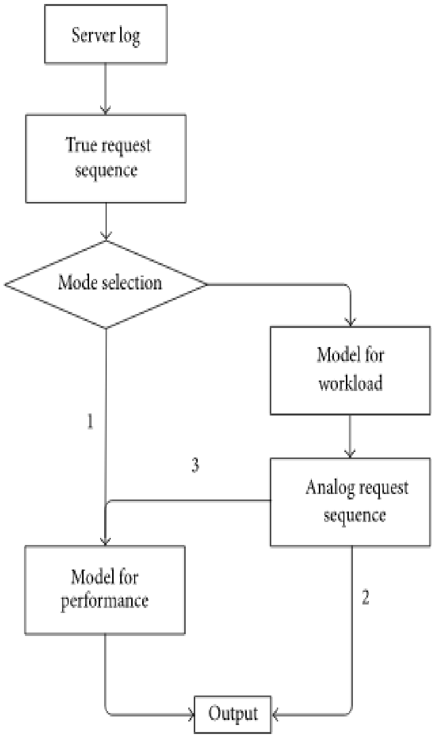
5. TRAINING AND APPLICATION FRAMEWORK

By using the model for workload prediction, request sequence with the original load features can be generated. This model of usage can meet the needs such as when the data of new log is too little, and the performance under a long-term workload is required. With these 3 kinds of options to apply these two models, the models can be used more flexibly and adapted to more situations.

The main aim of our model is to predict the performance of web server based on urls. Nginx log files are our data files. Each url entered has a unique id and each url are stored in the directory.

The dictionary is generated by collecting all the unique url request string in the whole data set. Using this id, the url requests during a time window can be abstracted to a vector with d-dimensional, every dimension of which means the number of times user request the url during the time-window.

Take vector $v = (p1,\ p2,\ ...pd)$ as an example, pi in v means users requested the url whose id is i altogether pi times in the time window. This initialization step is regarded as constructing embeddings for url requests.



6.EXPERIMENT

The length of the Lstm network is set as 10 because it is assume that a url request can’t affect the web server performance after 10 seconds. So the requests in each 10 seconds are organized into one sequence, and 3423 sequences of url requests are generated finally. Our model is trained and tested on the GPU: NVIDIA Tesla K20c, and the model was developed on the framework of theano with CUDA to accelerate calculation. It took about 2 to 3 hours to finish the training of the model on the GPU. As a comparison, it will take about more than 15 hours to complete this job on a CPU.

7. FUTURE SCOPE

Improving the performance of predicting web server performance, we apply RNN-LSTM network with url embedding to this task. Both of these two kinds of method may not explore the essence of the problem. Since the fluctuation of server’s performance is caused by user’s url request sequence .

8.CONCLUSION

The performance of a web server can be predicted using different ways, it is very important to find the performance of a web server because the usage of internet is growing and it must be fast in the fast world. we propose to use

RNN-LSTM to predict web server performance and workload. Model for performance prediction is composed of RNN-LSTM and Multilayer Perceptron (MLP), and the one for workload prediction consists of RNN-LSTM and softmax layer. Doing the research based on events is a new way in this prediction area. Doing the research based on events is a new way in this prediction area. The models can extract features automatically during the learning process without any prior knowledge or hand-generated features for segmentation.

REFERENCES

[1] T. Berners-Lee, R. Cailliau, J.-F. Groff, and B. Pollermann, "World-wide web: The information universe," Internet Research, vol. 20, no. 4, pp. 461–471, 2010.

[2] R. Buyya, K. Ramamohanarao, C. Leckie, R. N. Calheiros, A. V. Dastjerdi, and S. Versteeg, "Big data analytics-enhanced cloud computing: Challenges, architectural elements, and future directions," in Proceedings of the 21st IEEE International Con- ference on Parallel and Distributed Systems, *ICPADS 2015*, pp. 75–84, December 2015.

[3] S. Islam, J. Keung, K. Lee, and A. Liu, "Empirical prediction models for adaptive resource provisioning in the cloud," Future Generation Computer Systems, vol. 28, no. 1, pp. 155–162, 2012.

[4] I. Davis, H. Hemmati, R. C. Holt, M. W. Godfrey, D. Neuse, and
S. Mankovskii, "Storm prediction in a cloud," in Proceedings of the 2013 5th International Workshop on Principles of Engineering Service-Oriented Systems, PESOS 2013, pp. 37–40, May 2013.

[5] R. Girshick, "Fast R-CNN," in Proceedings of the 15th IEEE International Conference on Computer Vision (ICCV '15), pp. 1440–1448, December 2015.

[6] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," Learning, vol. 3, 2015.

[7]. Wu, S. Pan, X. Zhu, C. Zhang, and X. Wu, "Positive and Unlabeled Multi-Graph Learning," IEEE Transactions on Cyber- netics, vol. 47, no. 4, pp. 818–829, 2016.

[8]. X. Ma and E. Hovy, "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF," in Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1064–1074, Berlin, Germany, August 2016.