International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 09 | Sep 2018 www.irjet.net p-ISSN: 2395-0072

Web server Performance Prediction using a Deep Recurrent network

Anjitha P

Department of Computer Science, St. Joseph's College (Autonomous), Irinjalakuda, Thrissur, Kerala

---

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. There exist several methods for predicting server performance but they are not efficient methods. This paper is intended to give a flash for a deep recurrent network for predicting the performance of a web server. A deep recurrent network is a network which has many connected hidden layers. A recurrent natural network with long short term memory can be used to find the performance of a web server.*

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

1. INTRODUCTION

A web server is a hardware or software that can serve contents to the World Wide Web. 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. The input of recurrent network is not the current input it is also affected by the decision in the previous step. RNN has a loop, the information are carried by neurons .The recurrent neural network remembers past result because it has an internal memory. The RNN has a loop and it takes decision based on the input and also the previous input. A recurrent neural network has a short term memory and combining with a long short term memory it has also long-term memory. If we input a string if the network forgotten the previous inputs it can not to predict next input. A recurrent neural network can predict the next input because of its internal memory. This paper gives a flash on how to predict the performance of a web server using a recurrent neural network. The prediction of future behavior and reliability is key for virtual management system. A virtual system must balance load in a cloud. The paper is organized into different sub sections

2. MODEL

2.1 LSTM network

Long short-term memory (LSTM) units are layers of a recurrent neural network. A RNN made up of LSTM are called as LSTM network. A LSTM network is made with input gate, output gate and forget gate. The cells store data hence it can be considered as memory. The three gates can be considered as conventional artificial neuron as in a neural network. An LSTM network stores a value temporary or for a long period As shown in Figure 1, the basic structure of a LSTM unit is composed of a memory cell $ct \in Rd$ and three essential gates: Input Gate $it \in Rd$, Output Gate $ot \in Rd$ and Forget Gate $ft \in Rd$.

The formulas for upgrading the state of each gate and cell

in a LSTM unit using the input of xt, $yt-1$, and $ct-1$ are

said as below $zt = g(Wzxt + Rzyt-1 + bz)$ (1)

$ft = \sigma(Wfxt + Rfyt-1 + bf)$ (2)

$it = \sigma(Wixt + Riyt-1 + bt)$ (3)

$ot = \sigma(Woxt + Royt-1 + bo)$ (4)

$ct = it \_ zt + ft \_ ct-1$ (5)

$yt = ot \_ h(ct)$ (6)

Here xt indicates the input feature vector at time t. Similarly,

$yt-1$ and $ct-1$ is the output vector and cell state at time $t-1$.

And all of them are a d-dimensional value. W in formulas

Can be defined the weight matrices of the input parts in the gates and cell of LSTM network, and R are of the recurrent parts. The $\sigma(x)$, $g(x)$, and $h(x)$ functions are the activation functions of every part in LSTM, which determine the amount information that can be passed. And here we use sigmod as the activation function of three gates ($\sigma(x)$ in the formulas), and tanh as the functions $g(x)$ and $h(x)$ in the formulas. So the final output of LSTM is in a [-1, 1] range. And b indicates

Bias vectors of each formula. As for the _ mark, it means

Point-wise multiplication. With this structure, LSTM network is robust with respect to exploding and vanishing gradient problems [10], so it is able to learn long-term dependencies which RNN cannot perform very well and

makes the model can be trained without hand-generated features.

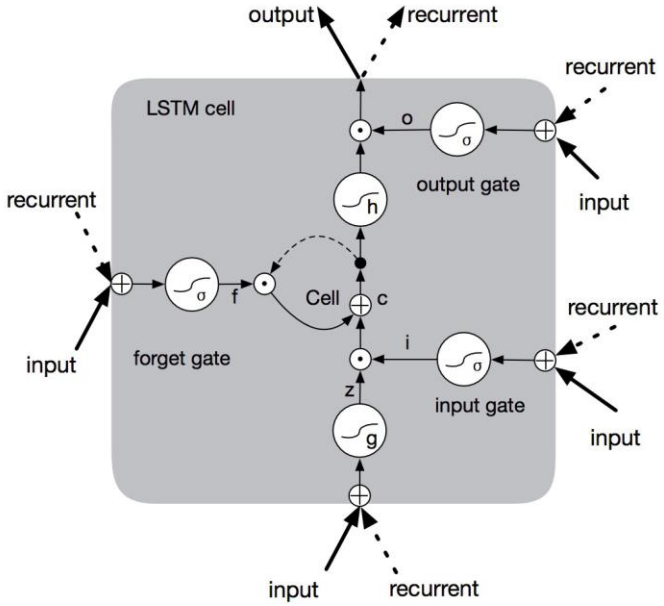

Chart -1: Schematic of LSTM unit [12]

2.2 Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

2.3 Regression

In normal output layer in LSTM are designed as a classifier here the output layer is designed as a regression task. The aim was to predict the request delay and throughput based on the urls hence the output layer is designed as a regression layer in a Multi-layer Perceptron (MLP)

Suppose the input vector is xi, and $\hat{}yi$ is the value predicted by the model, and θ means the network parameters, we use the variance between the real value of the performance and the predicted one ($\hat{}yi - yi$) 2 as the cost of $\hat{}yi$.

So the total loss function of the model is defined as follow:

$J(\theta) = m$

–

$i=1$

($\hat{}yi - yi$)2 (7)

And we apply the RMSPROP gradient descent algorithm when training the network which is an improved version of SGD algorithm and have batter performance with mini batches

3. TRAINING METHOD

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.

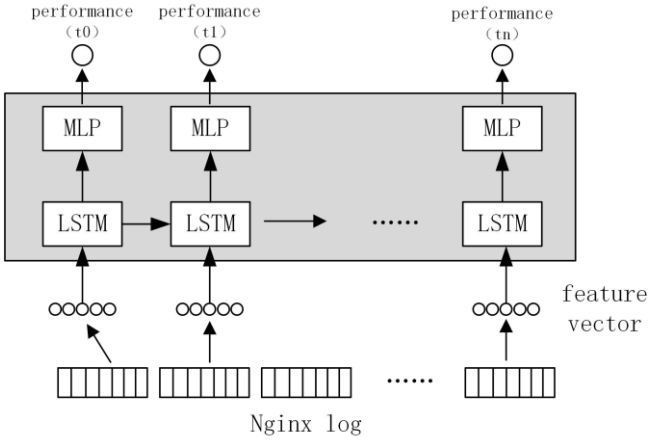

Chart -1: LSTM model for predicting web server performance

Datasets

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. EXPERIMENT

4.1 SETUP

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.

5. 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. The paper gives a flash on predicting performance of a web server using a recurrent network it is different from the existing networks the web server performance can be calculated using client server and browser server performance[2] it has a disadvantage that is is not easy to update the data base. The prediction of web server performance using a neural network is a fresh idea .The model can extract data from the learning process without any prior knowledge.

REFERENCES

[1] Jiajun Peng, Zheng HuangJie Cheng, "A Deep Recurrent Network for web server performance prediction," 2017 IEEE Second International Conference on Data Science in CyberspaceJ. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68-73.

[2] J.-f. Tu and R.-f. Guo, "The application reseach of mixed program structure based on client-server, browser-server and web.