**Resource Usage Prediction of CloudWorkloads using Deep Bidirectional Long Short Term Memory Networks**

Abstract

Resource usage prediction is an important aspect for achieving optimal resource provisioning in cloud. The presence of long range dependence in cloud workloads makes conventional time series resource usage prediction models unsuitable for prediction. In this paper, we proposed to use multivariate long short term memory (LSTM) models for prediction of resource usage in cloud workloads. We analyze and compare the predictions of LSTM model and bidirectional LSTM model with fractional difference based methods. The proposed LSTM models have been evaluated and compared with the state-of-the-art existing methods on Google cluster trace [1]. The experimental results show that the proposed algorithms outperform state-of-the-art algorithms.

1. Introduction

Cloud computing enables rapid access to a shared pool of hosted services. Different number of users enter and leave the cloud environment dynamically at varying instants to access the services offered by the cloud [2]. Due to the uncertainty observed in the number of users accessing the cloud services at different points in time, it is difficult for the service provider to efficiently allocate resources to their clients. This makes resource provisioning a challenging problem. Inefficiency in provisioning of optimal number of resources results in over/under-provisioning where the service provider ends up allocating either too many or too few resources, respectively. Predicting the needs of clients aids the service providers in better capacity planning and to dynamically scale the number of active machines to achieve an service level agreement (SLA) aware energy efficient resource provisioning system. Future resource usage prediction is formulated as a time series analysis problem where the past trends in the resource usage are analyzed to predict expected resource usage trends in future. Different statistical methods have been explored for predicting future resource usage patterns. For example, Markov models [3], [4], autoregressive integrated moving average models (ARIMA) [5] and neural networks [6] have been explored for prediction of future resource requirements. Most of these methods assume that the observations separated by a long time span are unrelated to each other. But it has been observed that dependence does exist between distant observations. In [7], fractional differencing based time series prediction models are used to analyze and capture long range dependence in cloud workloads. Long short term memory (LSTM) models are used in [8] to model the long range dependence in cloud workloads. In [8], univariate LSTM models are used for predicting future trends in resource usage where the CPU load is predicted with only CPU usage values. Multivariate time series prediction exploits the interactions between different features for generating future predictions. Multivariate time series prediction helps to analyze the affect of other features also on the desired resource metric. To the best of our knowledge, the use of multivariate LSTM has not been reported in literature for predicting future resource usage in cloud workloads. In our work, we explore multivariate analysis and analyze the effect of other features on CPU resource usage. The usage is predicted using different features namely, number of running jobs, memory usage, assigned memory usage, unmapped page cache, total page cache, maximum memory usage observed over a measurement interval, disk I/O time, disk space used, maximum CPU usage observed over a measurement interval, maximum disk I/O time observed over a measurement interval, cycles per instruction and memory accesses per instruction along with CPU usage itself [1]. Forward forecasting models analyze the trends involved in the past history to predict future patterns. A bidirectional prediction architecture results in reduced prediction errors because of the coupling of past dependencies with the future dependencies [9]. To the best of our knowledge, the use of bidirectional LSTM (BLSTM) has not been reported in literature for predicting future resource usage in clouds. In this work, we propose to build a BLSTM model for prediction of future trends in resource usage and we also propose to analyze and compare the predictions of different fractional difference based methods with univariate and multivariate LSTM and BLSTM models for prediction of future trends in resource usage. We use Google cluster trace [1] in our work for modeling cloud workloads and analyze for prediction of resource usage values. The main contributions of this paper are twofold. First contribution is multivariate LSTM model for future prediction of cloud resource usage. This is in contrast to [8], where univariate LSTM model is used. Our second contribution is BLSTM model to analyze the effect of forward and back ward prediction networks for future prediction of resource usage. The structure of the paper is organized as follows. In Section 2, we review the time series models used for resource usage prediction in cloud network. Section 3 presents the architecture of LSTM and BLSTM models. In Section 4, the results of the experiments performed for validating the proposed methods are presented. Conclusions and future work are presented in Section 5.

2. Review of existing models for resource usage prediction in cloud

One of the most important goals of resource usage prediction in cloud is to improve the quality of predictions to achieve more realistic predictions. Different methods have been explored to forecast future usage of different resources. In this section, we review some prominent approaches that have explored the domain of resource usage prediction in cloud using time series prediction methods. Gong et al. [3] uses Markov model for prediction of resources in Google cluster trace. The usage of the desired resource metric is divided into n bins and a transition probability matrix P of size n x n is computed. The probability of the next bin is predicted using Chapman-Kolmogorov equation as \_t = \_t􀀀1P (1) where \_t and \_t􀀀1 denote the probability of a state at time t and t 􀀀 1 respectively. AGILE [4] uses wavelet transform along with Markov models for prediction of resources in cloud. Wavelet transform provides multi-resolution analysis in both time and frequency domains which captures all the frequencies in time series along with their location. The basic idea of AGILE is to decompose the time series into wavelet based signals. Markov model is applied on each signal to generate future predictions. The final outof- sample predictions are generated by reconstructing the original signal from wavelets. Zhang et al. [5] uses a linear model called ARIMA method for prediction of resource consumption values. The value at time instant t, x 0 t is predicted as a weighted sum of previous lags. x 0 t = a0xt􀀀1 +: : :+am􀀀1xt􀀀m +b0\_t􀀀1 +: : :+bn􀀀1\_t􀀀n (2) where xi is the actual value at time instant i, a; b are the weights of the ARIMA model. m and n are the number of lags used and \_ are the error terms associated with the model. ARIMA model is a widely used time series forecast model but it cannot effectively capture non-linear patterns in time series data. iOverbook, [6] uses feed-forward neural network for resource usage prediction. The analytic equation of a neural network model is expressed as: x 0 t = Xn k=1 \_kg Xm i=1 \_ikxi ! (3) where x denotes the input time-series. m is the number of lags, n is the number of neurons in the hidden layer, g(:) is the activation function and \_, \_ represent the connection weights that will be learned during training. The approaches presented above assume the time series to be stationary and memoryless. However, it has been observed that long range dependence is present in cloud workloads [7]. Long range dependence is a phenomenon in time series analysis where an increase/decrease in the next step value is most likely affected by several past time lags in the series. In long memory time series, the dependence between successive observations decays more slowly as compared to a normal time series. The presence of long range dependence in cloud workloads is studied in [7] using the rescaled range analysis method and Hurst parameter value. To capture the effect of long memory in the data, fractional differencing method is applied. Fractional differencing operator is defined as an infinite binomial series expansion x 0 t = xt 􀀀dxt􀀀1 + d(d 􀀀 1) 2! xt􀀀2 􀀀 d(d 􀀀 1)(d 􀀀 2) 3! xt􀀀3::: (4) where d is the difference parameter, that can take fractional values. Fractionally differenced data is passed to resource prediction model and the predictions generated by the model are post processed. This method performed better than methods without fractional differencing but includes an extra overhead of preprocessing and post-processing the data. By nature, LSTM models are suitable handling long range dependencies. In [8], the future requirement of resources is predicted using LSTM network. However, the future requirement of the CPU resource usage is predicted by analyzing the trends in the past history of CPU resource alone. It has been observed in [10] for prediction of emergency department demand in Western Australia that using multivariate vector autoregressive models gives better predictions than using univariate autoregressive models. In our work, we propose to predict the future CPU resource usage by using multivariate analysis. We consider the effect of other features like memory usage, page cache, number of running jobs etc. along with the past CPU resource usage for predicting future trends. In the present work, we propose to build multivariate LSTM for resource usage prediction in cloud workloads. Most studies use unidirectional computational dependencies to predict future events from past information. In this work, the bidirectional extension of a multivariate LSTM model is explored and we evaluate its effectiveness for out-ofsample resource usage predictions. In [11], bidirectional neural network is evaluated by performing time series prediction on laser data. Experimentally, it was observed that integration of future-past dependencies in the bi-directional model produces a good advantage in training ability and prediction quality of the model as compared to the unidirectional model. Another work [9], uses bidirectional prediction architecture to perform time series predictions on sunspots data and observed reduced prediction errors by combining both past and future dependencies. A work [12], utilizes a bidirectional extreme learning machine mechanism for wind power forecasting and observed better forecasts after using the bidirectional mechanism. To the best of our knowledge, the use of BLSTM has not been reported for future prediction of resources by analyzing cloud workloads. In this paper, we propose to analyze and compare the predictions generated by BLSTM network with the LSTM model and the other fractional difference resource prediction methods.

3. Bidirectional long short-term memory (LSTM) network architecture

A major limitation of conventional time series prediction models is that they can only remember information from the recent past. In the training of neural network model, the local minimum of the error function is found by iteratively taking small steps in the negative direction of gradient. During the back-propagation phase, the gradient signal are iteratively multiplied by the weights of the recurrent hidden layers. If the weights are very small, the gradient gets so small that learning becomes very slow or stops working altogether. This is called vanishing gradients problem. It makes it difficult to learn the long-term dependency in the data. Long short-term memory [13] (LSTM) have been proved efficient in dealing with time series that have long range dependency. They overcome the vanishing gradient problem in neural networks by enforcing constant error flow through the internal states of special units called memory cells [13]. Univariate resource prediction models analyze the temporal relations involved in the past historical values of the desired resource metric only to learn patterns and generate the future out-of-sample predictions. Multivariate time series prediction helps to analyze the affect of other features also on the desired resource metric. Studying many related variables together rather than just one helps to gain a better understanding by analyzing the variations in other time series features in addition to a single time series [14]. In this work, we explore multivariate LSTM where the time series X is represented as a multivariate, continuous time series X = (x1; x2; : : : ; xN) where xi 2 RD 8i = 1; : : : ;N describing the time evolution of usage of different resources and different performance metrics. Here N is the number of time series observations and D is the dimension of each observation. The main goal of a multivariate LSTM is to forecast the value of a resource xt j at time instant t, as a function of itself as well as usage of other resources and different performance metrics xt j = f1(xt􀀀1 j ) + XD k=1;k6=j f2(xt􀀀1 k ) (5) where f1 and f2 are two functions. The major challenge with modeling multivariate data lies in capturing the temporal variations within each feature as well as the correlations between different features at each time step. Figure 1 shows the structure of an LSTM block. The LSTM architecture consists of different cells and each cell comprises of three gates that are used to regulate the flow of information in and out of the cell memory. The input gate regulates the flow of input information into the memory, (ii) the forget gate regulates the information that should be stored in memory and (iii) the output gate regulates the output activation of the block. The rectangular boxes in the figure represent a layer of sigmoidal/hyperbolic tangent neurons. The forget gate in the LSTM cell can be formulated as: ft = sig(Wxfxt +Whfht􀀀1 + bf ) (6) where ft is the output of the forget gate and sig(:) denotes the sigmoid activation function. Wxf and Whf represent the weights between input (xt􀀀1) and forget gate , weights between hidden layer (ht􀀀1) and the forget gate respectively. bf represents the bias of the forget gate. The old cell state is multiplied by ft to forget the part of state information. The new state is defined in two steps where the input gate determines the values that are to be updated and a vector of new candidate values is obtained using the tanh layer. The cell state ct is then achieved as: ct = ftct􀀀1 + it~ct (7) where ct􀀀1 represents the previous cell state at time t 􀀀 1, it is the output of input gate and ~ct is a vector of new candidate values given as: ~ct = tanh(Wxcxt +Whcht􀀀1 + bc) (8) where Wxc and Whc represent the weights of cell state with the input and previous hidden layer respectively. bc represents the corresponding bias. The input gate of the cell, it is defined as: it = sig(Wxixt +Whiht􀀀1 + bi) (9) where Wxi and Whi are the weights associated with the inputs and the previous hidden layer for the input gate. bi indicates the bias for the input gate. The output gate of the cell computes the output for the cell using the previous hidden layer and the present input. The outcome of the output gate is used to compute the outcome of the hidden layer at time t, ht. ot = sig(Wxoxt +Whoht􀀀1 + bo) (10) where Wxo represent weights for input in output gate and Who represent weights for past hidden layer in output gate. bo represents the bias of the gate. The output of the hidden layer is computed as: ht = g(ot tanh(ct)) (11) where g(:) is the activation function. The output Yt is computed as: Yt =Whyht + by (12) where Why and by represent the weights and bias respectively. [15]. Unidirectional networks predict future outputs by learning the trends from only past inputs. Bidirectional networks on the other hand, integrate dependencies from future as well as the past to learn the time varying patterns. Bidirectional LSTM learns the patterns in the time series by using both forward and backward dependencies. Combining the models going both forward and backward in time aids in reaping more benefits for capturing different time series patterns. A bidirectional model processes data in both directions where the forward layer iterates from t = 1; : : : ; T and the backward layer iterates from t = T; : : : ; 1 and learning the trends in the reverse time series. Therefore, intuitively they are expected to be better than unidirectional LSTM. Figure 2 shows the structure of BLSTM network. These models analyze the time-series in both forward and reverse directions with two separate hidden layers. These hidden layers are then passed as input to the same output layer. The output Yt is then computed as: Yt =Wb hyhbt +Wf hyhf t + by (13) where hbt , hf t represent the output of the backward and forward hidden layers at time t. Wb hy and Wf hy denote the weights between the final output and the backward and forward hidden layers, respectively. by represents the bias.

4. Results and discussions

In this section, we present the results of the studies performed for predicting CPU utilization in cloud workloads. The experiments performed in this study are conducted on Google cluster trace [1]. Google cluster trace is based on a cluster of about 12500 machines and provides run time information of different jobs arriving to the cluster for a 29 day period. The workload provides data on arrival, execution and termination of different jobs along with their time-stamps. This trace gives an insight into the real cloud environment. In this work, we analyze and predict the CPU resource usage metrics. We took 60480 samples (7 days) for training the resource prediction models, the next 20 (3 minutes) samples of the time series are used as validation data for selection of appropriate parameters. We generate and analyze the out-of-sample predictions for the succeeding 60 (10 minutes), 120 (20 minutes) and 180 (30 minutes) steps ahead. Resource usage values are aggregated at 10 seconds time interval. To visualize the existence of long range dependence in the data, autocorrelation plot of CPU resource usage is analyzed for the 40 lags as shown in Figure 3. The lags vs autocorrelation plot gives correlation between the successive observations of a time series [7]. We can see that there exists a very slow decay in correlation between successive observations indicating the presence of long range dependence . Because of the presence of long range dependence, conventional time series prediction models (ARIMA, Markov models) are not suitable for future resource usage prediction as they can only model exponentially decaying correlations [16]. Therefore, we use LSTM and BLSTM models in our work for resource usage prediction in cloud. In the present work, we generate out-of-sample CPU utilization predictions using both univariate and multivariate LSTM and BLSTM models. To build the models, we analyze both stateless and stateful variations of the models. If the model is stateless, the cell states are reset at each sequence. However, in the stateful model, all the states are propagated to the next batch. We divide the data in L batches of size bs, then X = (X1; : : : ;XL) XL = (x((L􀀀1)\_bs+1); : : : ; x(L\_bs)); xi 2 RD In the stateful model, all the states are propagated to the next batch. The state of the observation located in batch (L􀀀1) is used in the computations of the next batch XL . Figure 4, presents a comparison of stateless vs stateful model using one layer LSTM at different neurons. It can be observed that the error in the RMSE generated by stateful LSTM is less as compared to stateless LSTM model. As the stateful models can retain memory between batches, they work better than stateless models. We use a layer-wise construction scheme to build multilayered LSTM and BLSTM models and explored upto five layers in the models. Although we have observed that in our case, the optimum is achieved for three layers. Upto two layers, the model appeared to be underfit and above three layers, we have observed that the model is overfit in nature. Deep networks are formed by piling several hidden layers above each other. The output of one layer is passed as input for the next layer. In this study, the number of LSTM cells in a hidden layer are varied from 16 to 128. The LSTM blocks use the logistic sigmoid for the input. Keras [17] with Theano based framework [18] at its backend is used for the implementation of models. The training of the models is performed on a NVIDIA Tesla K80 GPU. Mean square error is used as a loss function where error et is given as: et = PT t=1(yt 􀀀 ^ yt)2 T (14) where, yt denotes the actual resource usage value at time instant t and ^ yt denotes the resource usage value predicted at time instant t. We compare and analyze these time series predictions with fractional difference based time series prediction models. In case of fractional difference method, we use rescaled range and Hurst exponent value to compute the value of fractional difference parameter (d) [7]. Different resource prediction models viz. PRESS, AGILE, ARIMA and NARNN have been implemented with fractional differencing for generating future multi-step ahead time series predictions. Table 1 presents the multi-step ahead RMSE of the out-ofsample predictions of (i) PRESS (ii) AGILE (iii) ARIMA and (iv) NARNN v) LSTM-U (univariate LSTM) (vi) BLSTM-U (univariate BLSTM) (vii) LSTM-M (multivari- ate LSTM) and (viii) BLSTM-M (multivariate BLSTM) resource prediction models. From the Table, it is seen that NARNN generates best predictions from the set of four of fractional differenced based models. Long short term memory networks model long range dependencies in data well and their RMSE of predictions is lower than fractional differenced time series prediction models. Multivariate LSTM and BLSTM networks generate better predictions than all univariate models and multivariate BLSTM performs the best among all the resource prediction models. Figures 5 and 6 present the out-of-sample predictions generated by multivariate LSTM and multivariate BLSTM models respectively. We analyze the iterative multi-step predictions of the resource prediction models at (a) 60 (b) 120 and (c) 180 steps ahead. From the figure, we can see that the models capture the trends in resource usage very closely. The BLSTM model exploits the coupling in both forward and backward time series dependencies and hence generates better predictions than the LSTM model. Comparing the fractional differencing and LSTM based models, we can say that LSTM based models generate better out-of-sample predictions but have more training overhead in terms of training time and computational resource requirements.

5. Conclusions and Future work

Forecasting resource usage is valuable for making better scheduling and load balancing in cloud. Although numerous time-series models are available, the research for generating better resource usage forecasts has never stopped. In our work, we explored the LSTM and BLSTM models for resource usage prediction. We observed that the LSTM networks model long range dependencies in time series based resource usage data well and generate better out-ofsample predictions. The proposed multivariate extensions of LSTM and BLSTM models generate better predictions than univariate models. We compared the predictions of stateful and stateless LSTM models and observed that as the stateful LSTM can retain memory between batches, they performed better than the stateless models for resource usage predictions. Comparing the predictions of BLSTM and LSTM on CPU resource usage prediction, we can say that coupling between future and past prediction transformations provides good prediction ability over the unidirectional models. In the future, we plan to extend our work to utilize the predictions generated by resource prediction models for dynamic scheduling of resources in cloud.