International Journal of Advances in Engineering and Management (IJAEM)
Volume 2, Issue 12, pp: 167-172 www.ijaem.net ISSN: 2395-5252

Estimating Cloud Workload employing gradient descent back propagation in Deep Neural Networks

Bhagyashree Kamle[1], Anil Kumar[2]
[1]Research Scholar, [2]Assistant Professor
[1,2]Department of Computer Science and Engineering, Oriental University, Indore (M.P.)

Date of Submission: 25-12-2020 Date of Acceptance: 06-01-2021

ABSTRACT: Cloud Computing has revolutionized computing off late with several domains and applications resorting to the cloud architecture. However effective task scheduling and load balancing is critical for cloud based servers. This is typically a very challenging task keeping in mind the fact that cloud workload is a parameter that depends on several other parameters. Moreover, due to the enormity of the data and its complexity, the use of machine learning or artificial intelligence based techniques is important for cloud workload estimation. Forecasting future workloads with high accuracy is especially challenging due to the randomness of the cloud workloads and also the non-deterministic nature of the governing or affecting parameters. Hence, due to the size and complexity of the data involved, finding regular patterns is a challenging task at hand. The present work proposes a back propagation based deep neural network architecture for cloud workload forecasting. Two different approaches employing steepest descent i.e. the Levenberg training rule and the scaled conjugate training rule have evaluated. The experiment uses the NASA cloud data set. The performance evaluation parameters have been chosen as mean absolute percentage error (MAPE), Mean Square Error (MSE), number of iterations and regression. It has been found that the Levenberg's steepest descent approach outperforms the scaled conjugate gradient based approach in terms of the evaluation parameters.

Keywords – Cloud Workload Estimation, Artificial Neural Network (ANN), Back Propagation, Steepest Descent Approach, Mean Absolute Percentage Error (MAPE), Mean Square Error (MSE), Iterations, Regression.

I. INTRODUCTION

Cloud Computing has become one of the most sought after technologies which plays a pivotal role in several domains resorting to the high levels of data complexity, complex computation or applications needing hybrid platforms [1]-[2]. One of the most important aspects of cloud systems management is the fact that cloud servers sporadically face sudden surges in the number of requests often termed as clou d workload. This workload, if unforeseen can result in crash of the cloud server if alternate provisions are not made to handle the cloud workload [3]-[5]. This in term needs the estimate of cloud workloads in advance considering several governing factors. This is majorly critical especially for applications such as e-commerce and finance which may see sudden surges in requests. Thus there is a clear necessity of cloud workload prediction using models which can estimate cloud workloads with high accuracy. Statistical techniques are not found to be as accurate as the contemporary artificial intelligence and machine learning based approaches [6]. In this paper, a back propagation based approach for estimating cloud workload is proposed employing deep neural networks. [7].

II.DEEP NEURAL NETWORKS

Deep neural networks has evolved as one of the most effective machine learning techniques which has the capability to handle extremely large and complex datasets [8]. It is training neural networks which have multiple hidden layers as compared to the single hidden layer neural network architectures [9]-[10].

The architectural view of a deep neural network is shown in figure 1, [11]. In this case, the outputs of each individual hidden layer is fed as the input to the subsequent hidden layer. The weight adaptation however can follow the training rule decided for the neural architecture. There are various configurations of hidden layers which can be the feed forward, recurrent or back propagation etc.

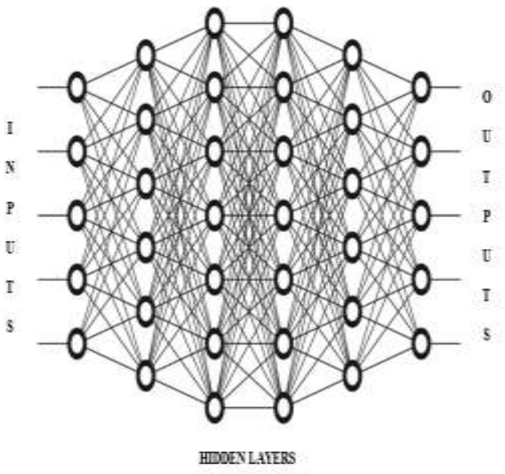

Figure. 1Architecture of Deep Neural Networks

The figure above depicts the deep neural network architecture with multiple hidden layers. The output of the neural network however follows the following neural networks rule:

$$\mathbf{Y = \sum_{i=1}^{n} X_i \,.\, W_i \; + \; \theta_i} \qquad \textbf{(1)}$$

Here,
Where,
X are the inputs
Y is the output
W are the weights
Ө is the bias.
Training of ANN is of major importance before it can be used to predict the outcome of the data inputs.

III. BACK PROPAGATION IN NEURAL NETWORKS

Back propagation is one of the most effective ways to implement the deep neural networks with the following conditions:

1) Time series behavior of the data
2) Multi-variate data sets
3) Highly uncorrelated nature of input vectors

The essence of the back propagation based approach is the fact that the errors of each iteration is fed as the input to the next iteration. [11]-[13]. The error feedback mechanism generally is well suited to time series problems in which the dependent variable is primarily a function of time along with associated variables. Mathematically,

$$\mathbf{Y = f(t, V_1 \ldots . V_n)} \qquad \textbf{(2)}$$

Here,
Y is the dependent variable
f stands for a function of
t is the time metric
V are the associated variables
n is the number of variables

Typically, the output or the dependent variable can be linked not only to the present inputs but also on the previous outputs. Hence back propagation becomes even more effective for multi-variate problems such as time series prediction problems. The back propagation based approach can be illustrated graphically in figure 2 [12].

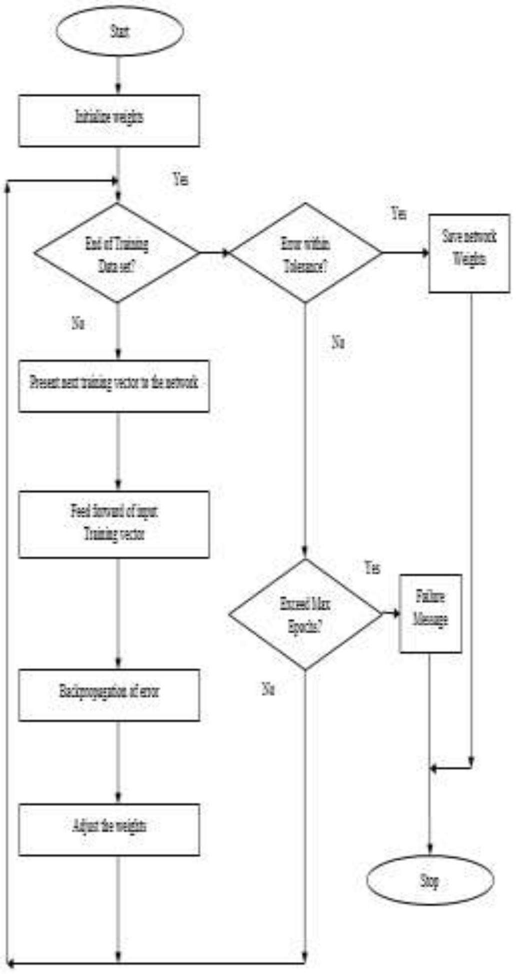

Figure.2 Back Propagation Mechnanism in Neural Networks

The In case of back propagation, the weights of a subsequent iteration doesn't only depend on the conditions of that iteration but also on the weights and errors of the previous iteration mathematically given by:

$$\mathbf{W_{k+1} = f(W_k, e_k, V)} \qquad \textbf{(3)}$$

Here,
W_{k+1} are the weights of a subsequent iteration
W_k are the weights of the present iteration
e_k is the present iteration error
V is the set of associated variables

In general, back propagation is able to minimize errors faster than feed forward networks, however at the cost of computational complexity at times. However, the trade off between the computational complexity and the performance can be clearly justified for large, complex and uncorrelated datasets for cloud data sets [1].

IV. PROPOSED APPROACH

The proposed approach applies the gradient descent approach in back propagation on the benchmark NASA dataset [12]. The gradient descent approach tries to effectively reduce the error in each iteration with the maximum rate of change with respect to each iteration. The gradient descent algorithms (GDAs) generally exhibit:

1) Relatively lesser memory requirement
2) Relatively faster convergence rate

The essence of this approach is the updating of the gradient vector g, in such as way that it reduces the errors with respect to weights in the fastest manner. Mathematically, let the gradient be represented by g and the descent search vector by p, then

$$\mathbf{p_0 = -g_0} \qquad (4)$$

Where,

g_0 denotes the gradient given by $\frac{\partial e}{\partial w}$

The sub-script 0 represents the starting iteration

The negative sign indicates a reduction in the errors w.r.t. weights

The tradeoff between the speed and accuracy is clearly given by the following relations:

$$\mathbf{W_{k+1} = W_k - \alpha g_x}, \quad \mathbf{\alpha = \frac{1}{\mu}} \qquad (5)$$

Here,

w_{k+1} is the weight of the next iteration

w_k is the weight of the present iteration

g_x is the gradient vector

μ is the step size for weight adjustment in each iteration.

The above equation shows stability in errors with monotonic decrease but needs higher number of iterations, specifically more in deep learning architectures due to direct computation of the Hessian Matrix of gradients. A faster approach is given by:

$$\mathbf{W_{k+1} = W_k - [J_K{}^T J_k]^{-1} J_K{}^T e_k} \qquad (6)$$

In this case, the number of iterations reduce at the cost of the stable monotonic reduction of the errors with respect to weights.

Here,

J_k represents the Jacobian Matrix given by $\frac{\partial^2 e}{\partial_w{}^2}$

J_k^T represents the transpose of the Jacobian Matrix.

The speed of convergence is due to the indirect computation of the Hessian Matrix by using the Jacobian computation given by:

$$\mathbf{H = J_k{}^T J_k} \qquad (7)$$

And

$$\mathbf{g = J_k{}^T e} \qquad (8)$$

Here,

H is the Hessian Matrix

Finally, the GDA with both speed and stability optimized is given by the Levenberg's Training rule [15]:

$$\mathbf{W_{k+1} = W_k - \left[J_K{}^T J_k + \mu I\right]^{-1} J_K{}^T e_k} \qquad (9)$$

Here,

The differentiating factor is the combination co-efficient μ which optimizes the GDA by adjusting the weights and thus the gradient.

K is the sub-script representing the iteration number

The activation function used for the algorithm is the tan-sig function mathematically defined as:

$$\mathbf{tansig(x) = \frac{2}{1+e^{-2x}} - 1} \qquad (10)$$

For the sake of comparison in terms of the accuracy of estimation, the other low space complexity scaled conjugate gradient (SCG) approach is also analyzed which is mathematically governed by the following training rule:

$$\mathbf{K_0 = -d_0} \qquad (11)$$

Here,

K represents the initial gradient search vector

d is the actual gradient

$$\mathbf{w_{k+1} = w_k + \mu_k d_k} \qquad (12)$$

Here,

w_{k+1} is the weight of the subsequent iteration of training

w_k is the weight of the present iteration of training

μ_k is the step combination co-efficient values

V. EXPERIMENTAL RESULTS

The data is bifurcated randomly into training and testing in the ratio of 70:30. The evaluation of the proposed approach is evaluated in terms of the following parameters:

1) Mean Absolute Percentage Error (MAPE), mathematically expressed as:

$$\mathbf{MAPE = \frac{100}{M} \sum_{t=1}^{N} \frac{E - E_t|}{E_t}} \qquad (13)$$

Here E_t and $E_t^{\sim}$ stand for the predicted and actual values respectively.

The number of predicted samples is indicated by N.

The MAPE facilitates in estimating the errors more accurately by nullifying the negative errors which may cancel out with values of the positive errors.

2) Mean Square Error (MSE), mathematically expressed as:

$$\mathbf{MSE} = \frac{1}{N}\sum_{i=1}^{N}(E - E_t)_i^2 \qquad \mathbf{(14)}$$

The mean square error is a metric to evaluate the training performance and deciding the epoch to terminate training. If the mse becomes stable for 6 or more consecutive iterations, the training is stopped, or if the maximum number of iterations is reached.

3) Regression

The extent of similarity between two variables is given by the regression where the maximum value is 1 and the minimum is 0.

The proposed scheme is tested using ordinarily image processing. From the simulation of the experiment results, we can draw to the conclusion that this method is robust to many kinds of watermark images.

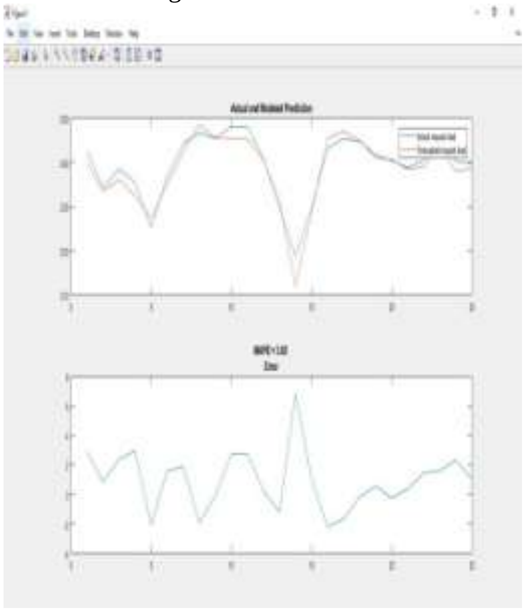

Figure.3 Obained MAPE for the Levenberg's steepest descent appraoch

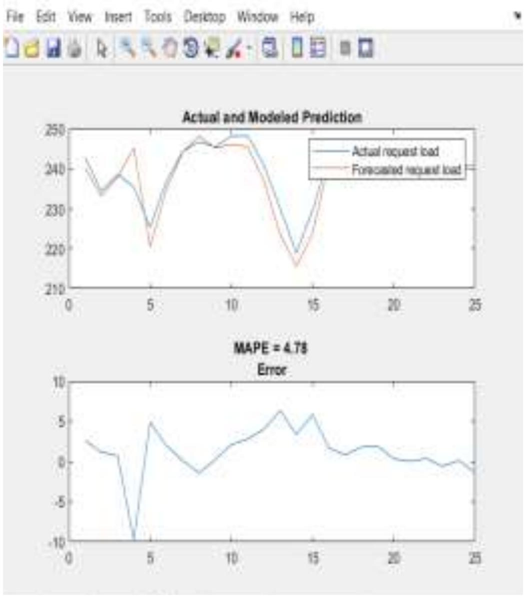

Figure.4 Obained MAPE for the Scaled Conjugae Gradient steepest descent appraoch

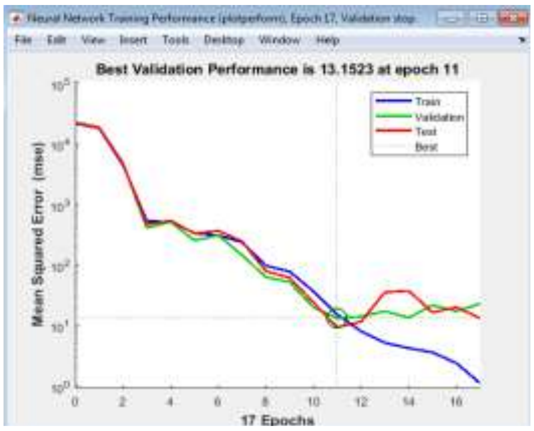

Figure 5. (a) MSE Plot for Levenberg's steepest descent

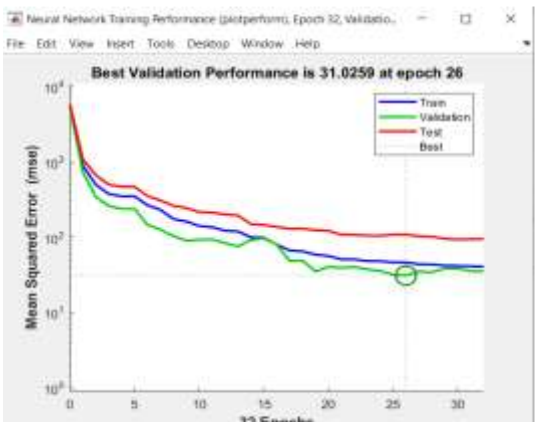

Figure 5(b). MSE Plot for Scaled Conjugate Gradient steepest descent

From figures 3 and 4, it can be clearly observed that while the Levenberg's steepest desent attains an MAPE value of 3.65%, the SCG's steepest descent attains an MAPE of 4.78%, thereby clearly indicating the fact that the previous appraoch is more accurate.

Figure 5(a) and 5(b) depict the comparison of the training progress in terms of the variation of MSE as a function of the epocts or iterations. Again, the Levenberg's steepest desent attains an MSE of 13.1523 at 11 iterations while the SCG based steepest descent attains an MSE of 31.0529 at 26 iterations. This indicates the the Levenberg's appraoch not only outperforms the SCG based appraoch in terms of time complexity, but also in terms of the accuracy of prediction and mean square error. The number of iterations is a critiical parameters keeping in mind the fact that as the number of iterations increase, the time complexity of the system also increases.

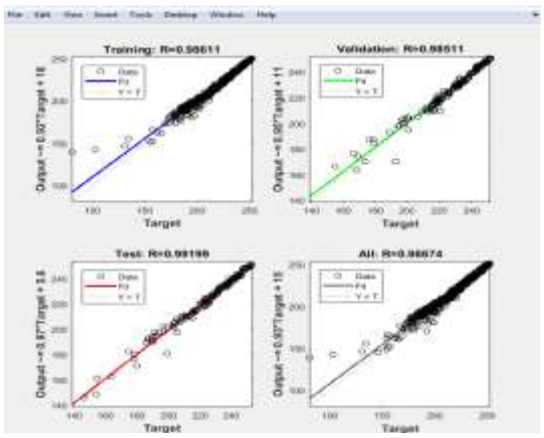

Figure 6. (a) Regression for Levenberg based steepest descent

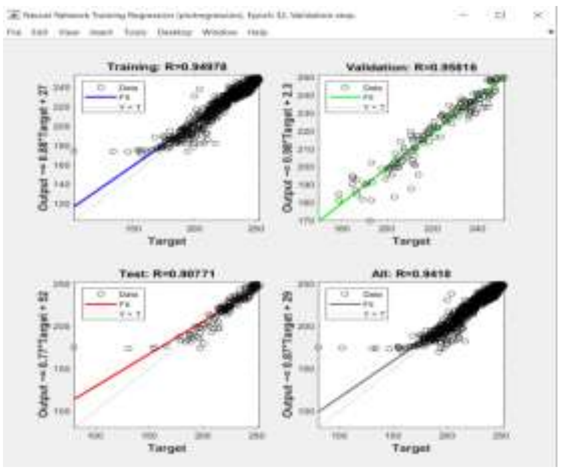

Figure 6. (b) Regression for SCG based steepest descent

Figure 6(a) and 6(b) compare the regression obtained by both the approaches for the training, testing, validation and overall instances. It can be clearly seen that the manifestation of lower MAPE and MSE is evident on the regression curves too. While the Levenberg steepest descent attains and overall regression of 0.98674, the SCG based steepest descent attains an overall regression of 0.9418 thereby clearly indicating that the previous approach is more accurate.

Table -1 Summary of Obtained Results

	Levenberg based Steepest Decent	**SCG based Steepest Descent**
MAPE	**3.65**	**4.78**
MSE	**13.1523**	**31.0259**
ITERATIONS	**11**	**26**
REGRESSION	**0.98674**	**0.9418**

Table 1 shows the comparative analysis of both the approaches in terms of the evaluation parameters for the estimation of cloud workloads. It can be clearly seen that the Levenberg based steepest descent outperforms the SCG based steepest descent in terms of all the evaluation parameters depicting speed of convergence of the algorithm as well as accuracy.

VI.CONCLUSION

It can be concluded form the previous discussions that the extensive use of cloud based platforms for various applications has made it necessary to estimate cloud workloads well in advance so as to manage constrained resources. The estimation is however non-trivial keeping in mind the staggering amount of data that is to be analyzed for cloud based platforms. Hence one of the most effective choices is to use machine learning based techniques for the purpose. In this work, two different machine learning approaches namely the Levenberg's backpropagation and the scaled conjugate gradient based approaches have been developed to estimate cloud workloads. The performance metrics have been chose to be the mean square error, mean absolute percentage error and the regression. It has been shown that the proposed approach attains a predication accuracy of over 95% for both cases and the Levenberg's approach outperforms the SCG.

REFERENCES

[1]. J. Kumar , A. k. Singh, "Workload prediction in cloud using artificial neural network and adaptive differential evolution",

Future Generation Computer Systems, Elsevier, Vol.81, pp: 41-52, 2018

[2]. L.Wang and E. Gelenbe, "Adaptive Dispatching of Tasks in the Cloud", IEEE Transcations on Cloud Computing, Vol.6, Issue.1, pp.33-45, 2018

[3]. M. Duggan, K. Mason, J. Duggan, E. Howley and E. Barrett, "Predicting Host CPU Utilization in Cloud Computing using Recurrent Neural Networks", 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), IEEE 2017

[4]. N. Liu, Z. Li, J. Xu, Z. Xu, S. Lin, Q. Qiu, J. Tang and Y. Wang, "A Hierarchical Framework of Cloud Resource Allocation and Power Management Using Deep Reinforcement Learning", 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), IEEE 2017

[5]. Y. Hu, B. Deng, F. Peng and D. Wang, "Workload Prediction for Cloud Computing Elasticity Mechanism", 2016 IEEE International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), IEEE, 2016.

[6]. J. Xue, F. Yan, R. Birke, L. Y. Chen, T. Scherer and E. Smirni, "PRACTISE: Robust Prediction of Data Center Time Series", 2015 11th International Conference on Network and Service Management (CNSM), IEEE 2015

[7]. M. Demirci "A Survey of Machine Learning Applications for Energy-Efficient Resource Management in Cloud Computing Environments", 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), IEEE 2015

[8]. A. Bashar, "Autonomic Scaling of Cloud Computing Resources using BN-based Prediction Models", 2013 IEEE 2nd International Conference on Cloud Networking (CloudNet), IEEE 2013

[9]. M. M. Taheri and K. Zamanifar, "2-Phase Optimization Method for Energy Aware Scheduling of Virtual Machines in Cloud Data Centers", 2011 International Conference for Internet Technology and Secured Transactions, IEEE 2011

[10]. T.V.T Duy, Y Sato, Y Inoguchi, "Performance evaluation of a green scheduling algorithm forenergy savings in cloud computing", 2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), IEEE 2010

[11]. Charu. C. Aggarwal, "Neural Networks and Deep Learning", Springer Publications.

[12]. MasteringMachine/Learning,

[13]. www.coursera.org