Journal of Theoretical and Applied Information Technology
30th April 2021. Vol.99. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MACHINE LEARNING APPLICATIONS TO MOBILE NETWORK PERFORMANCE MODELING

[1]BRANDON LITWIN, [2]KHALED ELLEITHY, [3]LAIALI ALMAZAYDEH

[1,2]University of Bridgeport, Department of Computer Science and Engineering, USA

[3]Al-Hussein Bin Talal University, Department of Software Engineering, Jordan

E-mail: [1]blitwin@my.bridgeport.edu, [2]elleithy@bridgeport.edu, [3]laiali.almazaydeh@ahu.edu.jo

ABSTRACT

Network optimization is a highly relevant solution to the growing demands of mobile communications. Increasing the efficiency of the existing spectrum and infrastructure drives down costs and improves provider networks' usability. This paper focuses on applying data mining to the evaluation of network performance as a viable tool for resource allocation. Network performance parameters are evaluated and passed through an artificial neural network to determine the categorization of high vs. low performance of the network environment and will detail parameters such as algorithm run time, accuracy and system requirements using a variety of different machine learning techniques.

Keywords: *Machine Learning, Mobile Network, Performance Modeling, Resource Allocation, Data Mining, Artificial Neural Network.*

1. INTRODUCTION

Network efficiency remains and will remain an important topic throughout the development of mobile communications. Telephones have dramatically changed the way society operates. With the increasing demands of both consumers and industry, it comes to an even greater need for faster and more robust data transmission.

Changing functionality of devices has driven processing power requirements to meet these demands, congesting the existing bandwidth and pushing providers to research new schemes and bands of the spectrum to squeeze out higher transmission rates and lower loss. These changes are often costly and spread across a massive scale, upgrading entire base stations and mobile devices, pushing costs higher for both providers and consumers [1]. To continue the upward trend in mobile communications requirements and demands, increasing the existing spectrum's efficiency is a cost-effective and easily implementable solution to both research and infrastructure upgrades [2]. This paper focuses on applying data mining to networking schemes to find a cost-effective solution to the growing demand by optimizing the existing spectrum and networking schemes.

There has been a great deal of research into utilizing data mining and machine learning algorithms to optimize network efficiency, with varying success. Network environments change rapidly and randomly, requiring effective algorithms that require high processing power, all of which create significant challenges in the field. Existing research has relied largely on Lagrangian relaxation and greedy methods to solve these optimization problems, and given the approximation associated with both, and they have only moderately succeeded [3], [4]. Other research has utilized a solution set search and comparison methods [5]. Large amounts of network performance data have been compiled into performance features, and these features are compared to real-time environments using decision trees to allow for resource allocation [6]. This has proven to be a better approach.

Artificial neural network (ANNs)-based mobility prediction schemes have considered from [7]-[11] to predict effectively different applications such as resource management, handover management and location-based applications.

2. PROBLEM IDENTIFICATION

The performance of mobile networks is an important field for both providers and consumers. Providers require streamlining for better profits and higher quality service, and customers need higher services with low latency and high accuracy. Network efficiency is, therefore, a relevant field regardless of the speed of networks. With Comcast

as the largest provider in the US and most criticized by consumers, increasing efficiency is of high priority.

Analyzing networks for performance is computationally expensive and complex. However, data mining offers a cost-effective and feasible solution. Providing a method of real-time computation for incoming data, streams provide an avenue for self-sustaining resource allocation with low overhead and high reward. Current methods rely on sampling large sets of data, which increases runtimes and requires high processing power [12]-[13].

This paper is organized in the following format: Section 3 defines the study design, shortfalls, and errors in the execution of the proposal and section 4 focuses on the results and comparisons of the empirical data, Section 5 explains the conclusions, and Section 6 focuses on future work.

3. PROPOSED SOLUTION

The current proposal is to train a large neural network with real-time data using an artificial neural network to allow for feature learning and the creation of a classification problem. This is similar in practice to a previously researched approach by Liu et al. [2] but using a different methodology. Instead of training the network using feature data, then classifying, the network will learn its own features. The algorithm's performance will be evaluated to determine the feasibility of implementation in a real-world environment, given a set of processing parameters run time and accuracy.

Suppose we utilize the latency as our output parameter. In that case, we can categorize each set of input parameters X as part of a single latency category E where E = X_1 , X_2 , X3 …Xn, and where n = number of data points or inputs to a base station or antenna. Each E is part of a latency category set T, where T = E_1, E_2, E_3…En is the total set representing all of the system's latency categories at time t. These inputs can be taken as a batch from a larger input set S, and categorized using a neural network, turning the loss optimization problem with Loss = (Actual – Expected)^2 , and finding the minima with respect to the weight given to each input parameter X_n, into a categorization problem. Once categorized correctly, the gradient for the sum of each category per batch sampled versus the previous batch, that is $\partial Et/\partial t$ dt can give information regarding the future category state of the system for the next batch or set of batches, with the assumption that the overall performance state of the system is MAX(T), where T, as above, is the set representing all of the categories of the system at time t.

By finding the system's overall performance state in this way, computation is simplified, and by finding the gradient of T with respect to the previous T, that is $\partial Tt/\partial t$ dt will give useful information for time t+1. Therefore, calculating the system's overall performance category will allow for a generalized single categorization value for T, and given the reliance of the MAX T ~ some E_n, solving for T is solving for E_n, where E_n is the largest set in T.

By having a trained algorithm that can model network performance based on input to a single categorical value, finding the MAX count of each category is solving for the system. Solving for $\partial E/\partial t$ dt gives insight into the future state of the system. Therefore the equation for the future state is then:

$$MAX(E_t + \partial E_t/\partial t\ dt) \quad \ldots\ldots (1)$$

For each category, we can determine which state the system will be in at time t+r, where r is an arbitrary unit of time. Using the state of the system as a scaling factor for each category gives insight into the probability of the future state, and accounts for uncertainty in the training system. The equation above then becomes:

$$MAX(E_t + (P_E * \partial E_t/\partial t\ dt)) \quad \ldots\ldots (2)$$

where P_{En} is the precision of the model for category E_n.

4. PROCEDURE

4.1 Study Designs and Shortfalls

Phase 1: GSN3 network simulation software was downloaded from [14], and the software ware trialed to produce a dataset of performance parameters for a given network of 10 nodes and two switches. The software was found to be unusable for the given function and was abandoned in use of an existing dataset found on Colorado Information Marketplace [15]. A second dataset was obtained from the same site and used as the test data. The dataset consisted of mobile device performance tests performed by a carrier from the Office of Information Technology and contained latency parameters, signal strength, download, and upload speed, among others. The datasets are 20,000 and 5000 data points, respectively.

Latency was used as the primary output parameter and was categorized into seven distinct

Journal of Theoretical and Applied Information Technology
30th April 2021. Vol.99. No 8

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

categories based on the average latency per mobile user. The data was preprocessed and cleaned first to remove none values and non-distinct fields from the set, and the categorical parameters were parsed to numeric integer values for use in the Keras architecture using Python. The network was run on a Windows 10 system using an i7 SkyLake processor with six cores, and a GeForce 1060 GPU with the potential for parallel streams using CUDA.

Various neural network architectures were employed and compared for accuracy and runtime, as network performance can change rapidly and unexpectedly. Run-time was considered a critical factor in determining a feasible algorithm for performance determination, given the importance of response time to network performance optimization. The weighted average accuracy of the categories was used to reduce bias towards rare categories, and a breakdown of occurrence and contribution to accuracy was tabulated. The future state of the system was calculated using Eq. 2.

4.2 Results and Comparisons

Using Keras with Tensorflow, several instances and accuracies were obtained with various architectures. Firstly, a dense, fully connected neural network was utilized with two layers, each containing 1000 and 7 neurons in the hidden and final layers, respectively. Results of the accuracy of precision, recall, and f1 score weighted averages were calculated. Using a sigmoid activation function produced very low weighted average accuracy for both precision and recall (0% and 6% resp.) using a training dataset of 18000 on a test of 2100 data points is shown in Figure 1. The runtime of 17us per epoch was considered acceptable. The Tanh activation function gave greater accuracy in both, as shown in Figure 2.

This was then optimized using an ADAM optimization function, which increased the model's accuracy without significantly sacrificing runtime, as shown in Figure 3.

	precision	recall	f1-score	support
0	0.00	0.00	0.00	72
1	0.00	0.00	0.00	132
2	0.00	0.00	0.00	876
3	0.06	1.00	0.11	119
4	0.00	0.00	0.00	280
5	0.00	0.00	0.00	113
6	0.00	0.00	0.00	507
accuracy			0.06	2099
macro avg	0.01	0.14	0.02	2099
weighted avg	0.00	0.06	0.01	2099

Epoch 10/10
26054/26054 [==============================] - 0s 17us/step - loss: 0.2266

Figure 1: Output for trial 1

	precision	recall	f1-score	support
0	0.03	0.60	0.06	72
1	0.00	0.00	0.00	132
2	0.39	0.36	0.37	876
3	0.00	0.00	0.00	119
4	0.00	0.00	0.00	280
5	0.00	0.00	0.00	113
6	0.00	0.00	0.00	507
accuracy			0.17	2099
macro avg	0.06	0.14	0.06	2099
weighted avg	0.16	0.17	0.16	2099

Epoch 10/10
26054/26054 [==============================] - 0s 12us/step - loss: 1.0003

Figure 2: Output for trial 2. Tanh with no optimization

	precision	recall	f1-score	support
0	0.00	0.00	0.00	72
1	0.00	0.00	0.00	132
2	0.42	0.96	0.58	876
3	0.04	0.02	0.02	119
4	0.20	0.03	0.06	280
5	0.00	0.00	0.00	113
6	0.00	0.00	0.00	507
accuracy			0.40	2099
macro avg	0.09	0.14	0.09	2099
weighted avg	0.20	0.40	0.25	2099

Epoch 10/10
26054/26054 [==============================] - 0s 15us/step - loss: 1.1614

Figure 3. Output For Trial 3. Tanh With ADAM Optimization

In an effort to increase accuracy, an experiment was run using the softmax activation function as given in Figure 4. Average accuracy of 17% for precision and 42% for the recall was obtained given a training set of 18000 data points on a test set of 2100 data points, with a run time per epoch of 18 us. One thousand neurons were used in this method, which added greatly to the runtime of the experiment per epoch. Dropping the number of neurons from 1000 to 100, as shown in Figure 5, reduced the run time without sacrificing the accuracy.

In optimizing the model, utilizing the Adagrad function brought the precision, that is, the correctly predicted positive observations / total positive observations, to 27%, ten percentage points higher than with Adam, with a modest drop in the recall, or correctly predicted observations / total observations, but more importantly, there is a greater accuracy across all classes in categorization with only a modest change in the runtime as shown in Figure 6.

We used a different data set and using the data above as data set 1 = time t. The dataset set 2 at time t+1 is shown in Figure 7. We can calculate the delta for the max category to determine the system's future performance. The Adagrad implementation

was used for this purpose, given the higher diversity of the accuracy surrounding precision.

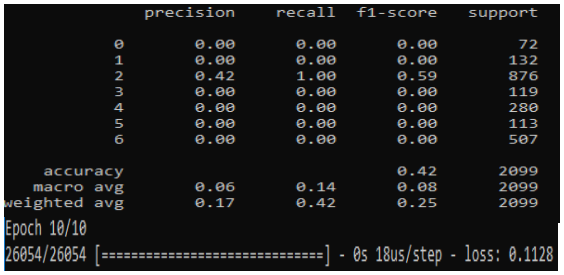

Figure 4. Output for trial 4. Softmax with ADAM optimization

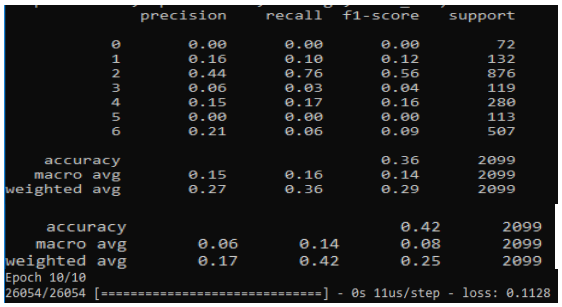

Figure 5. Output for trial 5. Softmax with ADAM Optimization with 100 neurons

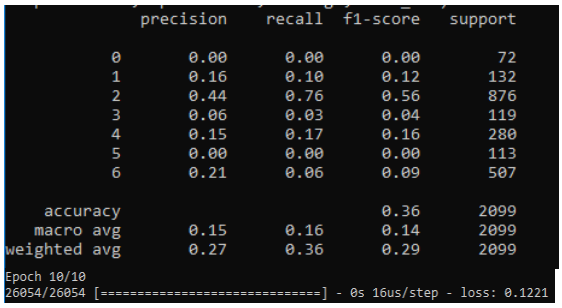

Figure 6. Softmax with ADAGRAD optimization

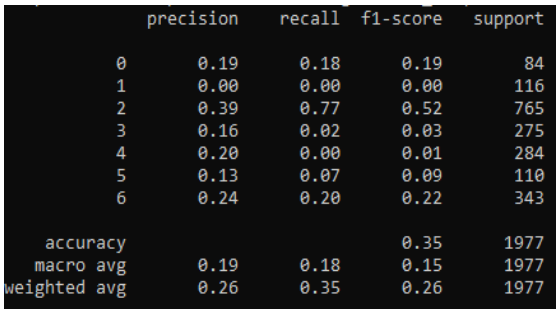

Figure 7. Softmax with ADAGRAD optimization

The system's performance is categorized as latency set 2 given the count data above, with the delta being [765 + 0.39*765-876/t+1 – t] = 721.71 at time t + r. It was calculated for all categories, as shown in Table 1.

Therefore, compared with the other deltas, the system will be in latency state 2 for the next time cycle.

Table 1: Predicted Future Latency

Category	Future Count
0	86.28
1	116
2	721.71
3	299.96
4	284.8
5	122.61
6	303.64

5. CONCLUSIONS

The activation function and optimization functions greatly changed the accuracy without sacrificing runtime. With an average mobile latency of 4G of 50us, these algorithms provide rapid performance analysis infractions of the average latency period, allowing for rapid potential response times for resource allocation. This study proves that with a simple machine running a six-core processor at 3.2 MHz and using historical data as a precursor for training, a 42% weighted accuracy can be achieved to classify real-time data to the tune of 2000 data points in under 1 second when ten epochs are utilized. An algorithm can easily be implemented to handle future performance calculations and input data to the trained model to predict future latency states of the network.

This kind of computation would have to be placed on the provider at *base* stations or antennas to allow for the correct allocation of resources and provide the processing power needed. This may be considered a hardware upgrade from existing schemes and may not be considered the best option for implementing a real setting.

6. FUTURE WORK

A different step would be to employ another type of network architecture for machine learning, that is, instead of a dense network, utilizing a Siamese or convolutional network on the dataset.

The utilization of CUDA coding schemes can also be experimented with employing parallel processing on the GPU rather than CPU and could reduce runtime for the algorithm.

For future work, given the dataset's optimization is solved, it would be to devise responses to low-performance environment classes. This would be analogous to resource allocation. The output of these responses could then be modeled to solve the optimization problem of resource allocation to networks.

Machine learning principles applied to mobile network performance classification presents an interesting approach to network efficiency optimization. Future work will design data generators in a user-friendly and readily producible format for research in network design and analysis.

REFRENCES:

[1] Grijpink, F. Menard, A. Sigurdsson, H. Vucevic, N. "The Road to 5G: The inevitable growth of infrastructure cost," in McKinsey & Company, 2018.

[2] J. Liu, Y. Xu and Z. Li, "Resource Allocation for Performance Enhancement in Mobile Ad Hoc Networks," in IEEE Access, vol. 7, 2019, pp. 73790-73803.

[3] Wang, J-B. Wang, J. Wu, Y. Wang J. Zhu, H. Lin, M. Wang, J. "A Machine Learning Framework for Resource Allocation Assisted by Cloud Computing," IEEE Network, Vol. 32, 2018, pp. 144-151.

[4] D. Prangchumpol, "Improving the performance of network traffic prediction for academic organization by using association rule mining," Fourth edition of the International Conference on the Innovative Computing Technology (INTECH 2014), Luton, 2014, pp. 93-96.

[5] Zhang, H. and Dai, L. "Mobility Prediction: A Survey on State-of-the-Art Schemes and Future Applications," in IEEE Access, Vol. 7, 2018, pp. 802-822.

[6] J. Liu, G. Bian, C. Qin and W. Lin, "A fast multi-pattern matching algorithm for mining big network data," in China Communications, Vol. 16, no. 5, 2019, pp. 121-136.

[7] S. Parija, R. K. Ranjan, and P. K. Sahu, ‘‘Location prediction of mobility management using neural network techniques in cellular network,’’ in Proc. Int. Conf. Emerg. Trends VLSI, Embedded Syst., Nano Electron. Telecommun. Syst. (ICEVENT), Jan. 2013, pp. 1–4.

[8] S.-C. Liou and Y.-M. Huang, ‘‘Trajectory predictions in mobile networks,’’ Int. J. Inf. Technol., vol. 11, no. 11, pp. 109–122, 2005.

[9] S. Parija, P. K. Sahu, S. K. Nanda, and S. S. Singh, ‘‘A functional link artificial neural network for location management in cellular network,’’ in Proc. Int. Conf. Inf. Commun. Embedded Syst. (ICICES), Feb. 2013, pp. 1160–1164.

[10] S. Parija, S. Nanda, P. K. Sahu, and S. S. Singh, ‘‘Novel intelligent soft computing techniques for location prediction in mobility management,’’ in Proc. Students Conf. Eng. Syst. (SCES), Apr. 2013, pp. 1–4.

[11] D. S. Wickramasuriya, C. A. Perumalla, K. Davaslioglu, and R. D. Gitlin, ‘‘Base station prediction and proactive mobility management in virtual cells using recurrent neural networks,’’ in Proc. IEEE Wireless Microw. Technol. Conf. (WAMICON), Apr. 2017, pp. 1–6.

[12] C. He, Y. Zhou, G. Qian, X. Li & D. Feng, “ Energy Efficient Power Allocation Based on Machine Learning Generated Clusters for Distributed Antenna Systems,” IEEE Access, Vol. 7, 2019, pp. 59575-59584.

[13] J. Wang, Y. Wu, H. Zhu & M. Lin, “A Machine Learning Framework for Resource Allocation Assisted by Cloud Computing,” IEEE Network, Vol. 32, 2018, pp. 144-151.

[14] GNS3 Network Simulator, https://www.gns3.com/software/download, accessed 12/24/2019.

[15] Colorado Information Marketplace https://data.colorado.gov/Telecommunications/AT-TUnusablePts/jwf3-czny,accessed 12/24/2019.