Applying Deep Learning Approaches for Network Traffic Prediction

Vinayakumar R, Soman KP and Prabaharan Poornachandran
Centre for Computational Engineering and Networking, Center for Cyber Security Systems and Networks, Amrita University, India.

ICACCI
September 13-16, 2017, Manipal, India

Abstract

Network traffic prediction aims at predicting the subsequent network traffic by using the previous network traffic data. This can serve as a proactive approach for network management and planning tasks. In this paper, recurrent neural network (RNN) is employed to predict the future time series based on the past information.

Introduction

In modern society, the Internet and its applications has become a primary communication tool for all types of users to carry out daily activities. This can create a large volume of network traffic. Traffic matrix provides the abstract representation of the volume of traffic flows from all possible sets of origin to a destination point in a network for definite time interval. The source and destination point can be routers, points-of-presence, internet protocol (IP) prefixes and links. A network service provider has to know the future trends of network parameters, routers and other devices information in order to proceed with the traffic variability. To deal with the problem of predicting the future trends of network parameters, routers and other devices information in real time network, prediction approaches have been employed.

RNN Feeding technique

We use the publically available and most well-known data set such as GÉANT backbone networks. The GÉANT network includes 23 peer nodes and 120 undirected links. 2004-timeslot traffic data is sampled from the GÉANT networks by 15 minute time interval. From the 10,772 traffic matrices, we choose arbitrarily 1200 traffic matrices. Each traffic matrix TM is transformed to vector of size 23*23 = 529 TMV. These vectors are concatenated and formed a new traffic matrix TM of size 1200*529. The traffic matrix TM is randomly divided in to two matrices such as training TMtrain 900*529 and testing TMtest 300*529.

There are various ways exist for predicting the future traffic. One way is to feed the vectors TMV in TM to prediction algorithms and predict one value of TMV at a time. This method was not correct due to the fact that the OD traffic is dependent on other ODs. Thus capturing the patterns exist in previous traffic can enhance the performance of prediction of traffic matrix. We use the slide window to create a training data set, as shown in Fig. 1. The number of gray color boxes is the length of the slide window with time-slots TS and OD pair. The values exist in right side of the slide window is the predicted value by prediction algorithms. We obtain TS − SW + 1 training records by sliding the window for the OD pair with TS time slots.

Experiments

All experiments of RNN and its variant network are trained using backpropogation through time on graphics processing unit enabled TensorFlow computational framework in Ubuntu 14.04. The optimal parameters and network structures for RNN networks is chosen by following the hyper parameter selection method. The performance of RNN and its variants networks on the test data set Tmtest is reported in Table 1.

Fig 1 Sliding window

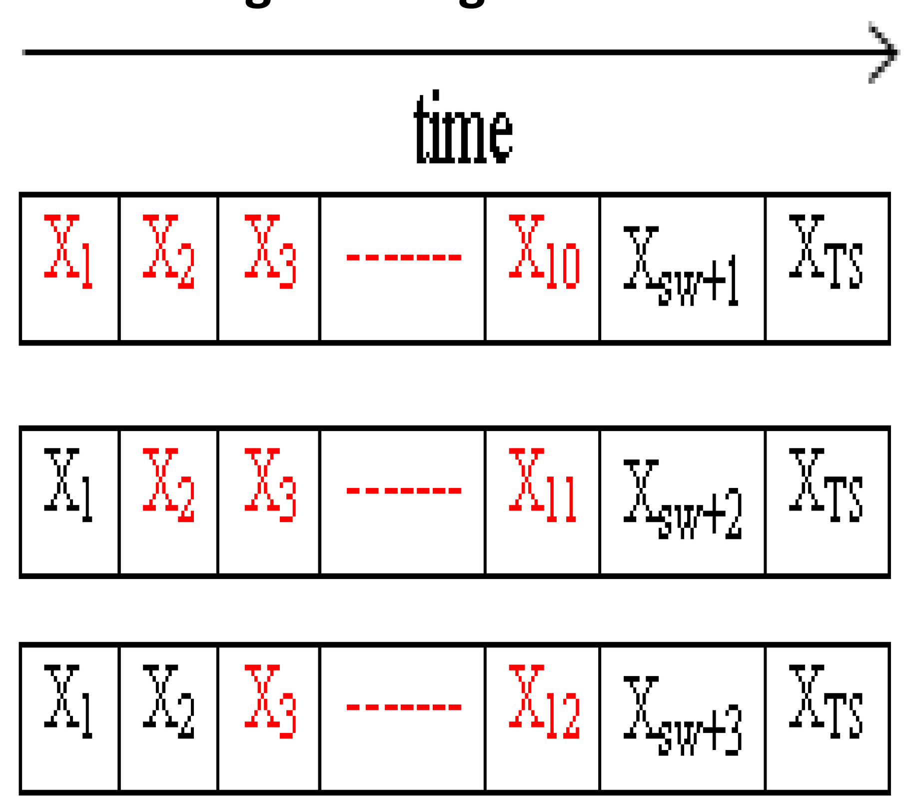

Table 1 Evaluation Results

Algorithm	MSE
FFN	0.091
RNN	0.067
LSTM	0.042
GRU	0.051
IRNN	0.059

Conclusions

This paper has discussed data preprocessing and feeding techniques for RNN and it evaluated the effectiveness of various RNN architectures for network traffic matrix prediction using GÉANT backbone networks. LSTM has performed well in comparison to the other RNN, and FFN methods.

The discussed RNN models are all same except the computational unit in the recurrent hidden layer. RNN are fundamentally complex networks that consist of many interactions between the computational units in the recurrent hidden layers to carry out a certain task from one layer to the other. Despite the fact that the effectiveness of various RNN has analyzed on traffic matrix prediction data sets under different experiments, the information about the internal procedure of the operation in the network is partly demonstrated. This can be considered as one of the direction towards the future work.

Acknowledgements

The authors would like to thank all at Centre for Computational Engineering and Networking, Coimbatore and Center for Cyber Security Systems and Networks, Amritapuri for their direct and indirect support.