EE599 Deep Learning – Project Propsal

©Tianchen Yuan, Xitong Wang, Zifan Wang

April 15, 2019

Project Title: Estimation of Origin to Destination Matrices using Link Flow Measured Data from Transportation Network

Project Team: Tianchen Yuan, Xitong Wang, Zifan Wang

Project Summary: In this project, we propose to estimate Origin to Destination (OD) matrices with traffic flow data at the Los Angeles International Airport (LAX) area. We will collect real-world traffic flow data from the Los Angeles Department of Transportation (LADOT), and use traffic simulation software VISSIM to generate training data. We will then build a Recurrent Neural Network (RNN) model to find the mapping between traffic flows and OD matrices. The estimation of real OD matrices can be obtained by putting the real-world traffic flows through the RNN model. The performance can be measured again by VISSIM.

Data Needs and Acquisition Plan: The real-world traffic flow data includes hourly traffic flows of 13 road segments and 22 parking entrances/exits within LAX area in April 2016. There are 32 OD locations so the size of each OD matrix is $32 * 32$. The $ij-th$ entry represents the traffic demand from location i to location j. Although OD matrices are of large size, they are sparse. There are approximately only 150 fixed nonzero entries that we need to estimate. To produce training data, we will first generate a random OD matrix based on the real data at each hour, and that will be 720 random OD matrices in total. We will then run the Dynamic Traffic Assignment (DTA) for each OD matrix in VISSIM to collect the corresponding traffic flows. These are somehow fake flows but will help us find the relationship between flows and OD matrices through RNN. Training data production is very slow given the data size we need. To tackle this problem, we will do Principal Component Analysiss (PCA) to both training input and output. Hopefully with PCA, the training data length can be reduced to 10800 (must be a multiple of 720).

Primary References and Codebase: We propose to build on the approach used in

- Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation," Conference on Empirical Methods in Natural Language Processing (EMNLP, 2014), Doha, Qatar.

- F.A. Gers, J. Schmidhuber, and F. Cummins "Learning to forget: continual prediction with LSTM," International Conference on Artificial Neural Networks (ICANN 1999), Edinburgh, UK.

- H. Yang, Y. Wang and D. Wang, "Dynamic Origin-Destination Estimation without Historical Origin-Destination Matrices for Microscopic Simulation Platform in Urban Network," International Conference on Intelligent Transportation Systems(2018), Maui, Hawaii, United States.
- Lecture Slide: Recurrent Neural Networks
- Traffic Simulation Software: PTV VISSIM 10
- GitHub codebases: RNN Tutorial, LSTM Neural Network for Time Series Prediction

Architecture Investigation Plan: The neural network that we plan to build takes time-sequence flow vectors as inputs and outputs time-sequence OD matrices. To model this time-sequence feature, we will utilize the GRU and LSTM architectures based on the above reference. We will also modify the network structure to make sure it is adaptive to our problem.

Estimated Compute Needs: According to the data set size we have and the benchmarks in LSTM Benchmarks for Deep Learning Frameworks, the training time per batch for our initial GRU or LSTM architecture will take about 13 ms on a machine with a Xeon W-2195 CPU and a NVIDIA GTX 1080 graphics card, which is close to the GPU resource in AWS p3.2xlarge instance. With about $3.06 per hour, we expect $0.2 per training run. The development of GRU or LSTM may make the model more complex, so our estimation is $0.5 per training. To improve our GRU or LSTM model, we will tune hyper-parameters in some provisional runs, which roughly costs $10. In addition, we expect to train our model for about 10 runs, which brings our total computing cost around $15.

Team Roles: The following is the rough breakdown of roles and responsibilities we plan for our team:

- Tianchen: Data Collection, VISSIM Simulation.
- Xitong: Data Augmentation, Slides.
- Zifan: Data Preprocessing, Get existing codebase running on an AWS instance.

All team members will work on the RNN architecture development, final presentation, and report.

Milestone: We are about to finish data collection and data preprocessing.

- **Data Collection:** For generating random OD matrices, we define 11 constraints based on the real-world flow data, including nonnegative entries, zero diagonal entries, no demand from parking to parking, etc. These constraints guarantee that our OD matrices make sense in the real world. Then we solve a constrained linear programming problem to obtain one random OD matrix. In one round, we have 720 hourly flow data, so we can generate 720 random OD matrices. In total we need 10800 OD matrices, so we repeat the process 15 times. The next step is to run DTA for each OD matrix in VISSIM to get the corresponding flows. At this time, we will have 10800 flow vectors as the training input and 10800 OD matrices as the training output.

- **Data Preprocessing:** Simulation of time-sequence flow vectors requires time length. We choose 4 days, i.e. 96 hours, as our time length, because 4 days are reasonable to monitor the flow change between weekdays and weekends. The length of each flow vector is 35, including 13 road segments and 22 parking entrances/exits in LAX. Therefore, each data sample is a $96 * 35$ dimensional matrix. In reality, there must be some correlations between traffic flows in main roads and nearby parking lots. As a result, we perform PCA to flow matrices and find that the top ten principal components have most of the energy. Then we reduce the dimension of our data sample to $96 * 10$.