Data-driven Traffic Forecasting Project Milestone

Jennifer Der, Shane Timmerman
Khoury College of Computer Sciences
Northeastern University
Boston, MA 02115
{der.je, timmerman.sh}@husky.neu.edu

1 Introduction

1.1 Summary

Forecasting traffic is a task that requires complex learning models that are able to account for mistakes and changes in traffic density over time. The traffic prediction problem on road networks can be cast as a spatiotemporal forecasting problem. While other models have attempted to accurately predict traffic patterns, many fail to accurately predict far ahead in time, to capture traffic diffusion behaviors, or to accurately adjust to non-stationary events. This paper proposes an implementation of a diffusion convolutional recurrent neural network (DCRNN) to capture the spatiotemporal dependencies in traffic forecasting. In order to model the changes in traffic density over time and spatial dependencies, the model uses a recurrent neural network, and a bidirectional graph random walk respectively. To improve the long term performance of the diffusion convolutional recurrent neural network, a scheduled sampling technique is used. The model was trained on two datasets, the METRA-LA dataset and another on the PEMS-BAY dataset, and experiments were then conducted to test its forecasting performance, modeling of spatial dependency, and modeling of temporal dependency against other state of the art models, comparing the mean absolute error, mean absolute percentage error, and root mean squared error for each between the forecasted traffic at 15, 30 and 60 minutes ahead to the actual traffic at those times. Against all models tested, DCRNN had the lowest error, indicating it was able to most accurately forecast traffic conditions and model the spatial and temporal dependencies of traffic. These results suggest DCRNN could be a good candidate for modeling other spatio-temporal dependant systems.

1.2 Other papers

There are many ways to represent traffic forecasting through space and time, and “deep learning approaches have been widely and successfully applied to various traffic tasks nowadays” [3] The deep learning approach uses a convolutional neural network to capture adjacent relations among the traffic network while also using a recurrent neural network to represent time. Data is commonly collected through traffic detectors or toll collectors aggregated over five minutes. [3, 4] The deep learning model is able to account for traffic growing forward and backward by using random walks to adjacent nodes in the graph, and the graph is sparsely populated, using weights as the distances so that it is more flexible than other representations. The model is trained not only on real data but also on its own predictions, which helps improve its predictive quality and reduces error propagation at more temporally distant times. [3]

1.3 Weaknesses

Current learning models lack “spatial components” and represent the areas between intersections as atomic distances and “spatial correlation between traffic links follows a more sophisticated pattern, which is not captured simply by the distance rule.” [4] A DCRNN model is able to account for spatial dependencies, but it does not account for nonstationary changes in traffic due to external

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

factors, such as weather, and does not have a means for indicating certain events are outliers due to these non-stationary events. For instance, heavy rainfall and snowfall can lead to slower traffic condiditions[11]. Moreever, the model proposed in the paper [1] only used traffic data as input was only trained on data collected over a 6 month period, which could introduce a temporal bias in the traffic predictions. There might be a bias towards traffic conditions in that period, and predictions could become less accurate in the opposite 6 month period of the year because traffic is dependent on weather and seasons (e.g. lighter traffic in the summer, more traffic because of snow). So far, a DCRNN model has been applied to traffic networks in California, which has a relatively moderate climate compared to New England.

1.4 Addressing weaknesses

Weather events and natural disasters significantly impact travel times and traffic throughout the world. In urban areas especially, where people rely on specific routes to get them from source to their destinations, it is difficult to know to what extent big storms that close down roads will impact traffic. Weather events can also be predictors for higher rates of other non-stationary events, such as increased accidents rates. Moreover, if we can model traffic throughout a city while preemptively deciding which roads will be closed, then we can understand the bottlenecks in traffic and understand the flow of traffic during crucial hours, create emergency routes, and have more stable travel times during weather emergencies. There may also be potential in simulating hypothetical road maps for urban planning and measuring how they perform before development. Predictive behavior for how closing down roads or beginning development on how that would impact traffic flow will be invaluable to city planners and officials to be able to plan around natural weather disasters and construction plans.

2 Proposed idea

2.1 How to solve the weather vs. traffic prediction problem

In order to improve upon traffic prediction, we plan to introduce data from traffic sensors in Los Angeles when certain roads were closed for construction or there was a natural weather event/disaster that closed certain roads. We will build a neural network learning model that will efficiently and accurately represent the spatiotemporal traffic between intersections. We will analyze the results and traffic predictions for different seasons, and the days during and following major weather events.

2.2 Novelty of solving the problem

Predicting traffic during emergency weather events could prove beneficial to city planners, and emergency personnel in order to serve the cities and urban areas that they serve. City planners can use these models in order to best schedule road construction/closing times that will interrupt the normal flow of traffic the least. Real-time simulation of road traffic when an edge (or road) is closed is a novel graph flow problem that has not been applied to traffic simulations and is not readily available to the general public.

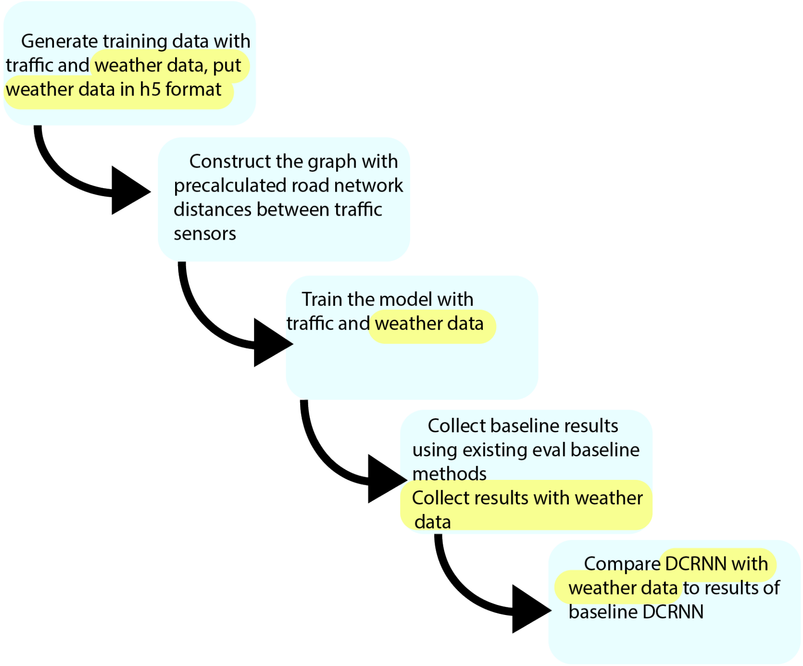
2.3 Datasets and baselines

We will be using a dataset for Los Angeles traffic data and an hourly weather dataset. The LA traffic data used will be the same METR-LA dataset used by the Yu et al. [1]. Weather data will be provided by a historical dataset collected by Open Weather Map from 2012 to 2018 [9]. We will only be looking at the data provided for Los Angeles. Attributes of the dataset include wind speed and direction, humidity, pressure, temperature, and a weather description enumeration. We will focus on a subset of that data collected from January 2017 through December 2017, so we can mitigate seasonal bias.

2.4 Measures of success

We will measure the mean absolute error (MAE), mean absolute percentage error (MAPE), and root mean squared error (RSME) between traffic forecasts from a weather-dependent diffusion convolution recurrent neural network (WDDCRNN) and the actual traffic conditions 30 minutes, 1 hour, and 2 hours ahead. We will compare the error rates to two other models: one where predicted changes in weather over time are not taken into consideration (weather at time t will be the same assumed weather at time t+60), and DCRNN, which will not take into consideration weather at all.

2.5 Project workflow


Generate training data with traffic and weather data, put weather data in h5 format
Construct the graph with precalculated road network distances between traffic sensors
Train the model with traffic and weather data
Collect baseline results using existing eval baseline methods
Collect results with weather data
Compare DCRNN with weather data to results of baseline DCRNN

3 Experiment

3.1 Datasets and pre-processing the data

For the datasets, we used the existing METR-LA traffic data form the original research [1] to train the model in combination with Los Angeles hourly weather data. [9] We are leveraging the graph construction and data partitioning from the original model's implementation. For preprocessing the weather data, the data needed to be pruned and reformatted into a hierarchical data format (h5 format) to be able to use it when training the model, and copartitioned with the corresponding traffic data into training, testing, and validation sets. These datasets have different time increments (5 min vs 1 hour), so there is some data duplication for the weather dataset. We inferred the weather over the course of an hour by duplicating the weather data given for 30 minute and one hour over 5 minute increments.

The weather data come in the following format:

datetime	humidity	pressure	temp	description	wind speed	wind direction
10/1/12 13:00	88	1013	291.87	mist	0	0
10/1/12 14:00	88	1013	291.87	sky is clear	0	0
10/24/12 8:00	72	1015	290.05	light rain	0	0
1/26/14 14:00	68	1025	280.84	few clouds	1	149
11/22/13 2:00	81	1010	287.37	thunderstorm	1	230
12/20/13 16:00	82	1009	280.83	heavy intensity rain	1	230

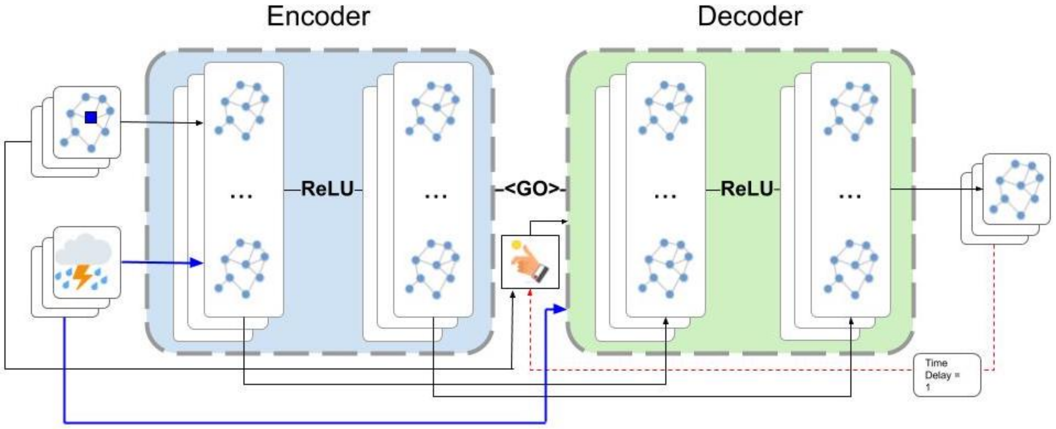
Since machine learning models are unable to handle categorical columns like the descrption, we transformed that column into continuous measures of cloudinees, raininess, and fogginess. For instance, a light rain would be 1 on the raininess scale, a moderate rain would be 2, and so forth. Similarly scattered clouds are a 1 on the cloudiness scale, few clouds are 2, etc. Sunny weather gets translated to a 0 in each.

datetime	humidity	pressure	temp	cloudiness	raininess	fogginess	wind spd	wind dir.
10/1/12 13:00	88	1013	291.87	0	0	1	0	0
10/1/12 14:00	88	1013	291.87	0	0	0	0	0
10/24/12 8:00	72	1015	290.05	0	1	0	0	0
1/26/14 14:00	68	1025	280.84	1	0	0	0	0
11/22/13 2:00	81	1010	287.37	0	4	0	1	230
12/20/13 16:00	82	1009	280.83	0	3	0	1	230

3.2 Experiment setup

Link to repository: https://github.com/jennder/DCRNN_PyTorch [1]

The model we propose is similar to the orginal DCRNN model; however, where it takes in only traffic data, this model would take in traffic and weather data, and output the predicted traffic conditions.

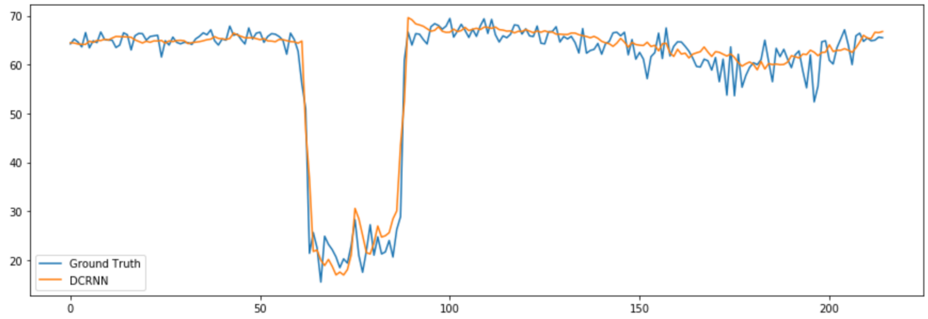


System architecture for the Weather Aware Diffusion Convolutional Recurrent Neural Network. The historical time series data for both weather and traffic are fed into an encoder, whose final states are used to initialize the decoder. The decoder makes predictions based on either previous ground truth or the model output, along with the ground proof weather conditions.

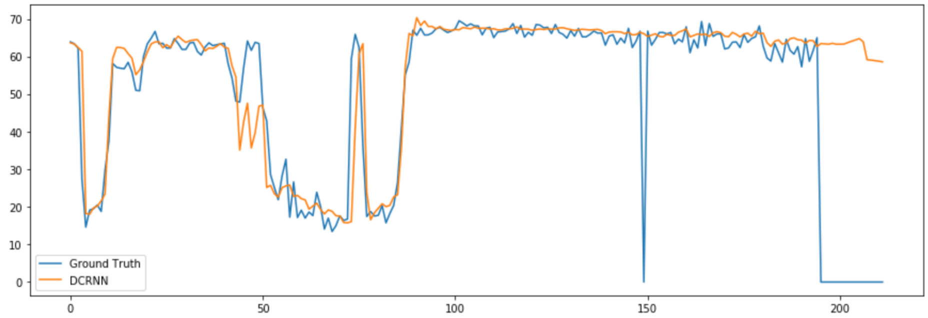
The proposed model is distinct from the DCRNN in its ingestion of both weather and traffic data. While traffic data is input to the decoder using the scheduled sampling approach, the ground truth for weather is fed in each time, as (1) the model only make predictions of traffic conditions and (2) in a real world scenario, such a model would have access to future weather forecast data to help it make predictions. The idea being to maintain the ability to predict traffic based on current traffic conditions, which DCRNN has been proven to do very well, while leveraging the predictive nature of weather events on traffic conditions to help better forecast traffic in the face of shifts in weather, which the current model in theory would be unable to handle as well. Specifically, the mean absolute error, mean absolute percentage error, and the root mean squared error will be collected for 30 min, 1 hour, and 2 hour ahead predictions with both the DCRNN model and our proposed model. We have opted to test accuracy at 2 hours rather than 15 minutes since this is where we would expect to see the largest weather based traffic changes, as weather does not often change significantly in shorter periods of time. In addition, similar to how response to peak hours was analyzed for the DCRNN, an analysis will be performed for periods of non-static weather events, such as a shift from sun to rain, so see if there is any bonus gained in the weather-aware model's ability to forecast traffic in such events.

3.3 Baseline results from existing methods

The existing model included in the repository uses a TensorFlow implementation of the DCRNN, and methods that are built to use with PyTorch, so we were not able to get baseline results using the existing trained models. We expect that traffic condition predictions one to multiple hours in advance computed using a model trained with weather data will produce more accurate results than the existing spatiotemporal traffic forecasting DCRNN implementation. The predictions made with the model trained with weather data is expected to perform better and be more accurate, especially when the weather changes rapidly throughout the day, or there is significant precipitation or changes in visibility (i.e. smog, rain, snow). [10]



Baseline results for DCRNN, only traffic data; predictions are relatively accurate. [1]



Baseline results for DCRNN only traffic data; predictions are not as accurate. [1]

South Pasadena
Alhambra
Temple City
San Gabriel
Rosemead
El Monte
West Hollywood
Beverly Hills
Los Angeles
Monterey Park
South El Monte
Culver City
Vernon
Montebello
Huntington Park
Maywood
Pico Rivera
Whittier
Inglewood
Bell
Cudahy
Bell Gardens
Santa Fe Springs
South Gate
Lynwood
Downey
El Segundo
Hawthorne
La Mirada
Compton
Manhattan Beach
Gardena
Paramount
Bellflower
Hermosa Beach
Buena Park
Lakewood
Artesia
Cerritos
Redondo Beach
La Palma
405
10
110
10
710
605
105
5
405
605
2
60
90
42
42
39
60
90
91
91

Reported "worse-than-usual traffic" due to heavy rainfall, traffic was slow and dense due to flooding in the area. [10]

3.4 Results from Experiment

We proposed to alter a pre-existing machine learning model that could predict traffic forecasting, and improve it by training the model with additional weather data. This problem is interesting because many data-driven approaches to spatiotemporal traffic forecasting do not incorporate external factors such as weather or precipitation. These external factors are not taken into account when making predictions for traffic many hours or days in advance, and could improve the accuracy of traffic prediction greatly. This is especially important in areas where weather can be turbulent, and/or one traffic obstruction or road closing causes hours of bumper-to-bumper traffic.

Unfortunately we were unable to develop, train, and test our model in time for this. We ran into significant issues in setting up the DCRNN to be compatible with Windows and setting up a training environment for our models. We also hit a number of road blocks navigating and iterating on the original code. Particularly, setting up the model to ingest the data from two separate sources, which is required in order for scheduled sampling, was something we could not figure out.

4 Conclusion

4.1 Summary

In the end, we were unable to produce a traffic forecasting model utilizing weather condition data. A significant portion of our time was spent familiarizing ourselves with the tools needed for development, namely PyTorch, and setting up our environment for training our models. In the end we decided to train the models on a windows machine with a dedicated GPU, as it seemed like it would take more time to setup a cloud environment to train the model. At the time, we were not aware of Northeastern's dedicated discovery clusters on AWS, and that we would have access to them. Training the model might have become feasible using the discovery cluster, however we also do not have experience setting up research projects such as this one in cloud computing resources. Once we had the environment set up on a desktop computer, we ran into a number of roadblocks modifying the model to work with weather data. Since the repository did not have any associated test suite, it was difficult to be sure if the changes we were making were working or just breaking code. This, mixed with our lack of experience with PyTorch and machine learning in general significantly hampered our development.

4.2 Take Aways

During the course of working on this project, we learned cloud computing resources such as Google Collab. While we were not successful in training our altered machine learning model, and running predictions in the cloud, it is still helpful to know that resources like this exist for the sole purpose of research. Going into this project, we knew that we would need significant computing resources in order to train our model, but we did not foresee to what extent we would need a computer with a dedicated graphics card, or how long it would take to train the model on even a small data set. Before this project, neither of us had hands-on experience with machine learning in any capacity. This includes setting up the environment, formatting the data sets, training the model, and making predictions. We learned about data formatting and how it needs to be prepared so it is comparable with machine learning systems. We also learned a lot about machine learning architecture and the benefits of some models over others. Analyzing the paper forced us to familiarize ourselves with recurrent neural networks and gated recurrent units and understand why they are potentially suited for the task of weather prediction. Similarly, the project made us learn about the encoder-decoder structure employed by many machine learning systems, how data needs to be set up for training, testing, and validation, and how a process like scheduled sampling can help develop more robust models. In addition, we learned about to work with existing machine learning projects, specifically through PyTorch, and how to iterate on existing models. Our project required us to change the model's configuration, layer interaction, and data ingestion processes. While it took a long time to familiarize ourselves with these tasks, and although we were unsuccessful in our attempt, the project provided a good introduction to the space of machine learning development and machine learning research.

References

[1] Li, Y., Yu, R., Shahabi, C. & Liu, Y. (2018) Diffusion Convolutional Recurrent Neural Network: Data-Drive Traffic Forecasting. *ICLR 2018*

[2] Chen, C., Liu, Z., Lin, W.H., Li, S. & Wang, K. (2013) Distributed Modeling in a MapReduce Framework for Data-Driven Traffic Flow Forecasting. *IEEE Transactions on Intelligent Transportation Systems* **, vol. 14** pp. 22–33., DOI: 10.1109/tits.2012.2205144.

[3] Yu, B., Yin, H. & Zhu, Z. (2018) Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting. *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, DOI: 10.24963/ijcai.2018/505.

[4] Ermagun, A. & Levinson, D. (2018) Spatiotemporal traffic forecasting: review and proposed directions. *Transport Reviews*, 38:6, 786-814, DOI: 10.1080/01441647.2018.1442887

[5] Ali U. & Mahmood T. (2018) Using Deep Learning to Predict Short Term Traffic Flow: A Systematic Literature Review. it In: Kováčiková T., Buzna L'., Pourhashem G., Lugano G., Cornet Y., Lugano N. (eds) Intelligent Transport Systems – From Research and Development to the Market Uptake. INTSYS 2017. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering , vol 222. Springer, Cham.

[6] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P.S. (2019) A Comprehensive Survey on Graph Neural Networks. *ArXiv 1901.00596*

[7] Zhang, J., Shi, X., Xie, J., Ma, H., King, I. & Yeung, D.Y. (2018) GaAN: Gated Attention Networks for Learning on Large and Spatiotemporal Graphs. *ArXiv 1803.07294*

[8] Traffic Related Data, (2019) Department of Innovation and Technology, https://data.boston.gov/dataset/traffic-related-data

[9] Historical Hourly Weather Data 2012-2017 (2018) Kaggle, https://www.kaggle.com/selfishgene/historical-hourly-weather-data

[10] Los Angeles Times, (2015) Los Angeles' Big Rain Day: Miserable Commutes, Power Outages, River Rescues. www.latimes.com/local/lanow/la-me-rain-traffic-live-updates-htmlstory.html.

[11] Jägerbrand, A. K., & Sjöbergh, J. (2016). Effects of weather conditions, light conditions, and road lighting on vehicle speed. SpringerPlus, 5, 505.