Cooper Sloan

Advisor: Alan Edelman

MEng Thesis Proposal

Deep Recurrent Neural Networks for Predicting Bus Arrival Times

Abstract

Have you ever had to run to catch a bus in the rain? Or have you waited in the rain for a bus that never came? Public transit is unreliable and unpredictable, but deep learning may have the answer. Current research shows that passenger waiting times can be reduced significantly through statistical modeling of bus arrival times. The current best modeling techniques use neural networks to predict bus arrival times. This research will apply deep learning and recurrent neural networks to improve the accuracy of bus arrival time predictions.

Problem

Increased use of public transit has the ability to significantly decreases traffic congestion and reduce the environmental impact of transportation. Nevertheless, most Americans commute via private cars, with only 10.8% using public transit[1]. One issue with public transit is that it is unreliable and unpredictable. Schedules are rarely followed due to the random nature of traffic networks. The unpredictability of traffic is compounded with other features of bus networks to make the problem even worse.

[1] McKenzie, Brian, and Melanie Rapino. Commuting in the united states: 2009. US Department of Commerce, Economics and Statistics Administration, US Census Bureau, 2011.

Bus networks have several unique characteristics which differentiate them from normal car networks. In particular, buses have to adhere to a schedule, and must arrive at stops at specified times. Additionally, several buses operate on the same route at the same time, and they influence one another. Specifically, buses exhibit a phenomenon called clumping. When a bus starts running behind schedule, more passengers arrive at future stops. The bus then has to wait additional time to pick up the extra passengers. This causes even more passengers to pile up at future stops, making the bus slow down even more. As this happens, buses at previous stops catch up to the delayed bus and clump up. The opposite effect happens when a bus starts running ahead of schedule. This complicated dependency between buses in a network makes predicting arrival times an interesting problem to study.

Accurate predictions can be used to make more reliable schedules and give passengers more ease of mind when using public transit. Additionally, predicting arrival times can significantly reduce waiting times for passengers. In my research I will focus on modeling the Boston bus network with an LSTM network to predict bus arrival times with high accuracy.

Related Work

Several different models have been used to predict bus arrival times. Basic regression modeling can reduce wait times for passengers[2]. Current research typically studies a single city over a period of up to a few months. Studies on predicting bus arrival times have been

[2] Patnaik, Jayakrishna, et al. 2004. Estimation of Bus Arrival Times Using APC Data. Journal of Public Transportation, 7 (1): 1-20.

conducted in the US, Canada[3], Brazil[4] and China[5]. In general, neural networks have shown the prediction best accuracy when compared to historical and linear regression models[6]. The neural networks used in most of the papers are simple, feed-forward neural networks with a few hidden layers. The most complex networks use basic time delay neural networks to process the sequential bus data[7].

The most famous study on bus arrival times comes out of Cuernavaca, Mexico. Researchers came to the somewhat surprising results that spacings between bus arrival times follow the Gaussian Unitary Ensemble (GUE)[8]. The GUE is a distribution which stems from random matrix theory. It models the spacing between eigenvalues of random matrices. It is not entirely obvious why this distribution would be well suited to model bus arrival times. However the distribution shows up in other traffic networks as well. This suggests that complicated interactions may be at play within bus networks.

All of the current models are relatively small, and are trained small datasets which span only a few months. This scale of data does not allow for training larger models or studying long-term effects. Furthermore, the current neural network techniques do not take utilize the fact that the

[3] Ibid. 2

[4] Weigang, Li, et al. "Algorithms for estimating bus arrival times using GPS data." Intelligent Transportation Systems, 2002. Proceedings. The IEEE 5th International Conference on. IEEE, 2002.

[5] Chien, S., Ding, Y., and Wei, C. (2002). "Dynamic Bus Arrival Time Prediction with Artificial Neural Networks." J. Transp. Eng., 10.1061/(ASCE)0733-947X(2002)128:5(429), 429-438.

[6] Jeong, Ranhee, and Laurence R. Rilett. "Bus arrival time prediction using artificial neural network model." Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on. IEEE, 2004.

[7] Shalaby, Amer & Farhan, Ali. 2004. Prediction Model of Bus Arrival and Departure Times Using AVL and APC Data. Journal of Public Transportation, 7 (1): 41-61.

[8] Milan Krbálek and Petr Seba "The statistical properties of the city transport in Cuernavaca (Mexico) and random matrix ensembles." 2000 J. Phys. A: Math. Gen. 33 L229

data is sequential. Recurrent neural networks are more suited to sequential data than standard feed forward neural networks because they have sequential sensitivity.

Approach

The dataset I will be using is GPS data from the Massachusetts Bay Transportation Authority (MBTA). The MBTA runs a server which can be queried to get the latitude and longitude of all of the buses in Boston, along with other metadata about routes and stops. This lab has collected several years worth of the GPS data. The dataset is several terabytes in raw XML, though it can be compressed to a more manageable size.

The GPS data then has to be converted into features which can be used as inputs to the neural network. One important features is travel time or the difference in arrival times between two adjacent stops. Another common feature used in the field is dwell time, or the amount of time the bus stays at each stops before leaving. Schedule adherence is also commonly used, which is the difference between the scheduled arrival time at a stop and the actual arrival time. These values can be computed from the data by comparing the GPS location of the bus with published GPS locations for each of the stops, and doing interpolation. Feature vectors can then be composed by combining each of the metrics for each bus over its entire trip on a route.

Bus trajectories are explicitly time series data. This type of data is well suited to recurrent neural networks. Furthermore, there is important state about the current traffic network which needs to be learned in order to predict arrival times. This suggest incorporating some state into the model to reflect the current traffic conditions. Long short term memory (LSTM) networks are well suited for modeling this type of interaction. Furthermore, the size of the dataset and the complicated nature of modeling traffic networks suggest that larger networks may be well suited to the problem. Therefore this research will apply deep RNNs model bus arrival times.

The model architecture will be an LSTM with several hidden layers, and a dense output layer. LSTM models are based on how human memory works. The following figure is from the very useful blog post by Christopher Olah[9]. It shows an LSTM gate, the basic building block of LSTM networks. It consists of three gates, a forget gate, and input gate and an update gate. These gates work together to store and update the memory of the unit. Several of these gate combine together to make up an LSTM network.

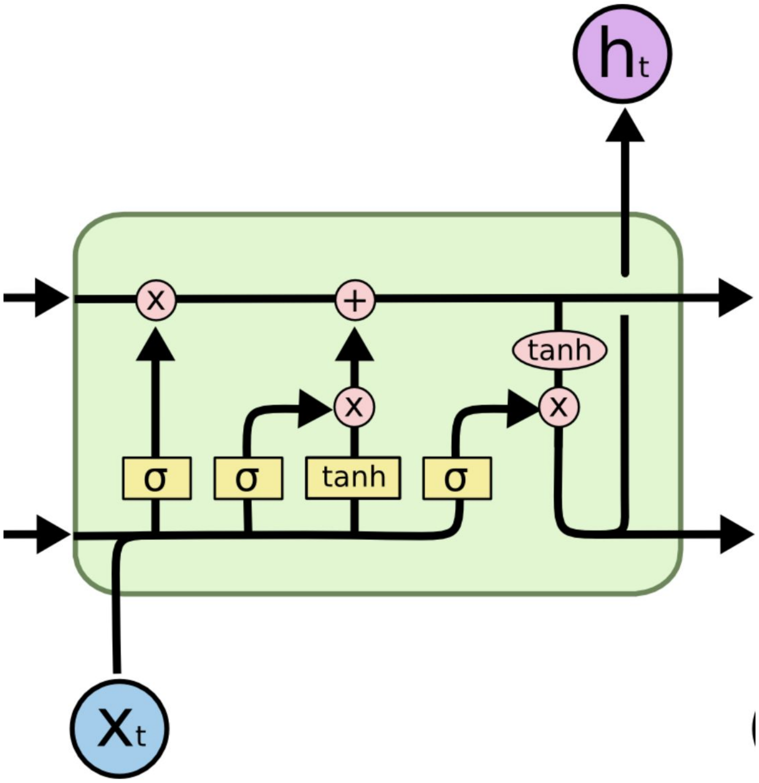

Figure 1: LSTM unit

[9] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Figure 2 shows an example of a simple LSTM. The blue boxes represent the network, green boxes represent training data, and the red boxes represent predicted probability distributions. LSTMs are just like normal feed forward models, except they have state which persists between consecutive passes through the network. The state is passed from one timestep to the next, and updated via a module called an LSTM gate. The state is combined with the input, passed through a series of linear and nonlinear transformations, and a probability distribution is the output. The probability distribution represents the networks belief about what the next input datapoint will be.

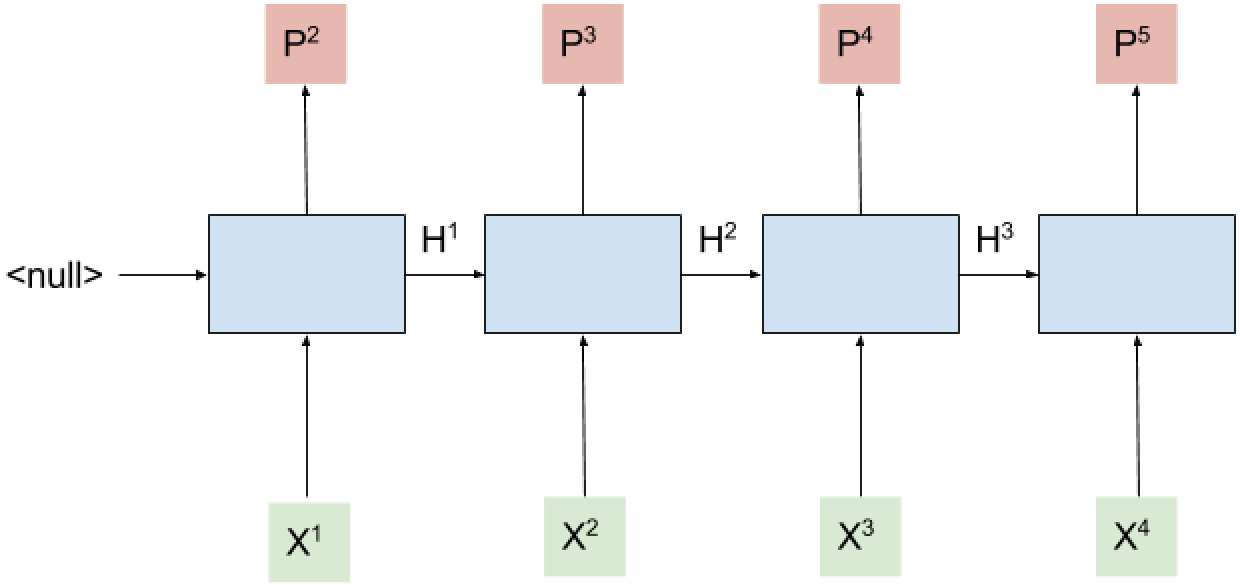

Figure 2

These types of networks can be trained in an unsupervised manner by feeding example training data through the network. The network can then be used to predict future sequences given a prefix. In this case the training data will be a sequence of features vectors for each stop on a route. Each feature vector will consist of the travel time, dwell time, and schedule adherence for each stop. The training data will then be used sequentially to train the model parameters via backpropagation through time. The dataset contains GPS data for all of the routes in Boston, so a different model will be trained for each route. The accuracy for predictions along each route can then be compared.

Additional characteristics which can be studied with this type of model are long-term and periodic effects. Traffic networks exhibit periodicity on a daily, weekly, and yearly basis. The size of the dataset allows for studying long-term behavior. Furthermore, LSTMs are well suited to modeling short term as well as long-term behavior. In order to study this behavior, temporal information can be added to the feature vector such as day of week, time of day, and year. The accuracy on test data can then be compared to the model trained without temporal data to determine the significance of periodic effects.

Evaluation

The output of the model will be continuous values, so a few important metrics to measure are mean square error, absolute error, R-squared score. Absolute error is an important metric because it reflects what passengers actually care about for a predicted arrival time. Mean square error is the metric which the model parameters are optimized for so it is also important to measure. R-squared is an important metric to measure in regression problems to determine how much the model is actually learning about the distribution. Even if absolute errors are small, the R-squared score shows if the model is actually explaining any of the variance in outputs. Specifically, these metrics will be evaluated for the training, validation, and test data. The training set will be used to select model parameters, the validation set will be used to select hyperparameters, and the test set will solely be used to evaluate the model. This ensures that test error will accurately reflect the expected error on novel data. The most important metric will be the test error.

Due to the size of the neural networks used, overfitting is a valid concern. The are several regularization techniques used for RNNs, including L1 and L2 penalties and dropout. Validation data can help in selecting the parameters for these regularization techniques.

Another important metric is the training time of the model, and the size of the model. If the model is computationally expensive to train, it will be less useful in applications. Therefore smaller models should be favored as long as they can achieve good accuracy. An important metric to measure will be the accuracy as a function of model size and training time.

On a similar vein, the dataset size will largely affect the accuracy of the model. If the dataset is too small, the model may not be able to generalize well. On the other hand if the dataset is too large the model may take too long to train, or may not learn at all.

Risks

There are several potential risks when tackling a machine learning application of this scale. The dataset is quite large, so training may be prohibitively slow. Additionally deep models are difficult to train in general, so even if the scale of the dataset in not an issue convergence may be an issue. Another significant issue, especially with larger models, is overfitting. Even with regularization techniques, there is no guarantee that a model which does well on training will generalize well.

The largest risk is modeling the bus prediction problem in the RNN framework. RNNs have significantly more architecture decisions than feed forward or even convolutional neural networks. How the data is framed, presented into the model, and how the model is trained have big impacts on the overall performance. Because current research has not used LSTMs for predicting arrival times, this is a significant risk.

Timeline

Fall 2017:
Work on processing GPS data and ensuring accuracy of measurements
Interpolate trajectories and compute features
Create baseline models and first iteration of LSTM
Begin writing thesis

IAP 2017:
Develop and iterate model and study behavior
Finalize model and finish with development

Spring 2017:
Interpret findings
Determine feasibility of LSTM models for bus networks

Relate to other work in the field

Finish writing thesis

Conclusion

Predicting bus arrival times can significantly decrease wait times for passengers. However current models are too simple to capture periodic and long-term effects. The MBTA dataset is unique in its time scale and size, and can be used to train larger models. LSTMs may hold the answer to improving prediction accuracy by modeling both short term and long-term periodic behavior.