Predicting the distribution of car demand CS230-Spring 2018

HyoungJu SEO
Department of Aeronautics & Astronautics
Stanford University
hjseo@stanford.edu

Abstract

This paper aims predicting car demands in New York city from weather, date and historical car demand data. I experimented with various parameters (Time sequential model length, Input features, Time step, Categorization, Depth and attention layer) of LSTM to maximize the accuracy of car demand. From these experiments, I predicted car demand with 3.49% error.

1 Introduction

Mobility-on-Demand services(e.g. carsharing and ridesharing) have expanded numerously in several years. On top of that, rapid autonomous vehicle technique development has significantly influence on new paradigms for future urban mobility systems. One of these paradigms is coordinating a fleet of self-driving vehicles which serve customers with on-demand transportation service, also called autonomous mobility-on-demand (AMoD). An AMoD system may replace service with positive effects of safety, parking infrastructure, and congestion.[6] One of the most challenging problem of AMoD systems is being out of balance by asymmetric demand : If some areas are more popular than others, vehicles will stack on those regions limiting the availability of other areas.[3] Furthermore, reducing congestion on the road with AMoD systems is crucial problem.[6] There are various methods for optimizing control of an AMoD system with rebalance and congestion strategy (e.g., using a fluidic model, a queueing network model or a road network model)[6][3][5]. Previous work mostly concentrated on the control of routes, schedules and the number of the autonomous vehicles on the road using steady-state values, not the time varying values like distribution of car demand. However, in the previous work we couldn't efficiently solve a key challenge of the AMoD system with asymmetric demand because those strategies are reactive : they do not rebalance the cars in the real time under rapidly varying traffic phenomena.[1] Thus, if we knew information about future car demand, autonomous vehicles could be efficiently allocated to meet demand over time and minimize customer wait times by rescheduling . To coordinate the system more effectively, the prediction of car demand is needed. Accordingly, I will predict the number of car demand at certain time from time series data of weather, date, and car demand using a Long short-term Memory(LSTM) model, while finding optimal LSTM model for forecast car demand.

2 Related work

As forecasting car demand has positive effects on not only Mobility-on-Demand platform but also future urban transportation system, AMoD system, there is an increasingly gained attention in predicting transportation demand. Recently, as deep learning architecture developed significantly, forecast with deep learning architecture is popular. [2] used a LSTM model to predict Uber demands especially at special holidays. [4] utilized deep learning architecture, RNN and LSTM and deep LSTM, to predict human mobility and transportation pattern (e.g. future movements, transportation choice, and destination) from GPS record and transportation mode. Because [2] more concentrated on special holidays and [4] more dedicated to mobility and transportation pattern, the research to find the most efficient input and model structure for prediction is not done and is needed. In this paper I will experiment with various parameters that can affect forecasting accuracy in LSTM and will find optimal one.

CS230: Deep Learning, Spring 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

3 Dataset and Features

As we can easily think that the number of daytime cars demanded, weather, and holiday have an impact on evening car demand, forecasting car demand model should be trained with historical data and diverse features relevant to the model. In this project, I used the database which includes Yellow and Green Taxi Pickup data in New York City from 2014 to 2017, found in NYC Taxi & Limousine Commission (TLC), `http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml`. Data from 2014 to 2016 is used for training and data from 2017 is used for evaluation. For the model's input, I analyzed car demand history, day information(holiday, day, time), and weather of NYC (temperature, precipitation, and weather description). Here is an example of the dataset.

Table 1: **Example of data**

pick up time	*Temp*	*Windspeed*	*Precipitation*	*Clear*	*Clouds*	*Fog*	*Rain*	*Snow*	*Thunderstorm*	*Holiday*	*Hour*	*Day*	*pick up num*
2014 − 01 − 01 01 : 00	268.90	2.0	1	0	0	0	0	0	0.0	1	2	1	33482
2014 − 01 − 01 02 : 00	268.48	3.0	1	0	0	0	0	0	0.0	2	2	1	36280
2014 − 01 − 01 03 : 00	268.33	3.0	1	0	0	0	0	0	0.0	3	2	1	33154
2014 − 01 − 01 04 : 00	268.10	3.0	1	0	0	0	0	0	0.0	4	2	1	28520
2014 − 01 − 01 05 : 00	268.02	1.0	1	0	0	0	0	0	0.0	5	2	1	20703

4 Methods

In this project, I predict the car demand distribution from time series data. To catch time-series characteristics, Recurrent Neural Network (RNN) is effective because it could memorize previous data, except a long-term memory. Long Short-Term Memory (LSTM) neural network overcomes the disadvantage of RNN. LSTM makes it possible to learn the long time-series by determining the optimal time lags for prediction.
A LSTM model has units composed of a memory cell, an input gate, an output gate and a forget gate. It computes a relation from an input sequence $X = (x_1, \ldots, x_T)$ to an output sequence $Y = (y_1, \ldots, y_T)$ with following equations.

$$i_t = \sigma(W_{ix}x_t + W_{im}m_{t-1} + W_{ic}c_{t-1} + b_i)$$

$$f_t = \sigma(W_{fx}x_t + W_{fm}m_{t-1} + W_{fc}c_{t-1} + b_f)$$

$$o_t = \sigma(W_{ox}x_t + W_{om}m_{t-1} + W_{oc}c_{t-1} + b_o)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g(W_{cx}x_t + W_{cm}m_{t-1} + b_i)$$

$$m_t = o_t \odot h(c_t)$$

$$y_t = \phi(W_{ym}m_t + b_y)$$

where i, f and o are the input gate, forget gate and output gate respectively. c and m are the activation vectors for each cell and memory block, and the weigh matrices W and bias vectors b are building a connection between the input layer, output layer and memory block. Also, $\odot$ means the scalar product of two vectors, $\sigma(\cdot)$ represents the sigmoid function, and $g(\cdot)$, $h(\cdot)$ and $\phi(\cdot)$ are the input, cell output and network output activation function. LSTM units architecture is following Figure 1.

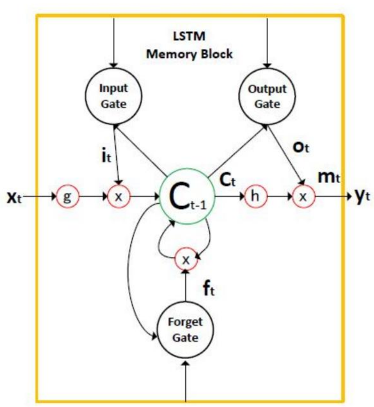

Figure 1: LSTM unit

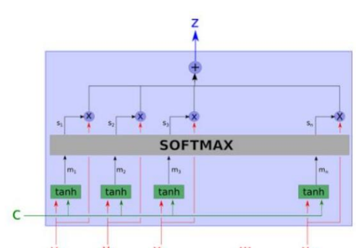

Figure 2: Attention model

However, LSTM model has a problem with connecting relevant inputs and outputs. An attention model is a method that takes n arguments $y_1, \ldots, y_n$ and a context c. It returns a vector representing summary of arguments : relevance of each argument and given context. The attention model's architecture is above Figure 2 and following equations.

$$m_i = tanh(W_{cm}c + W_{ym}y_i)$$

$$softmax(m1, \ldots, m_n) = (\frac{e^{m_i}}{\sum_j e^{m_j}})_i$$

5 Experiments/Results/Discussion

I predicted the total number of cars demanded over time in New York City with various parameters. This approach is able to figure out the appropriate LSTM model for predicting car demand. I experimented with respect to input condition, time sequential model length (the number of LSTM hidden layers), data categorization (depending on day and time), time step and structure of LSTM (the depth of LSTM and attention layer). Following Figure 3 is the architecture of my experiments.

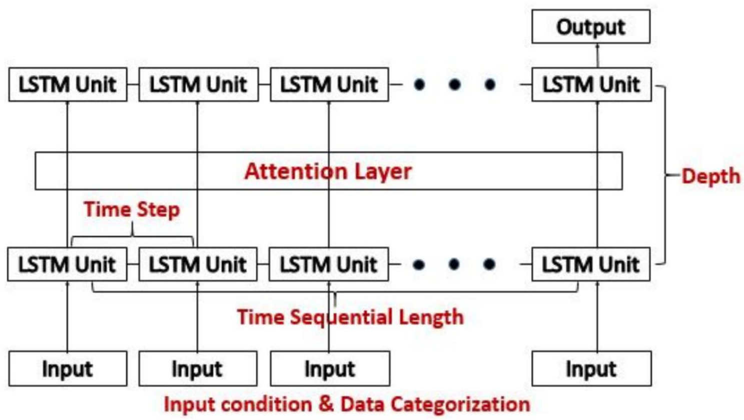

Figure 3: Architecture

The source code for the prediction is available at https://github.com/Hyoungju/CS230-Final-project-predicting-taxi-demand

5.1 Time sequential model length & Input

I experimented with different time sequential model lengths (the number of LSTM hidden layers) : 12 hours, 24 hours, 48 hours, with different input features : all weather information and only precipitation among weather information. Except that, the other parameters are set as 1 depth of LSTM, 1 hour unit time step, without categorization, 100 epochs, and 100 batch size(26280 train set, 6469 test set).

Table 2: **Error result of with different input and time sequential length**

	Result with all weather data			Result with only precipitation		
error	12 $hours$	24 $hours$	48 $hours$	12 $hours$	24 $hours$	48 $hours$
$R2(y \& \hat{y})$	0.97	0.97	0.97	0.97	0.97	0.97
$mean\ absolute\ error(MAE)$	880.87	773.33	851.58	853.98	784.15	756.61
$root\ mean\ squared\ error$	1199.40	1063.94	1198.17	1161.25	1110.63	1093.09
$MAE\ percentage\ test\ set(\%)$	9.47	7.79	8.37	8.94	7.25	6.80
$MAE\ percentage\ train\ set(\%)$	9.77	6.38	6.74	7.09	6.43	6.25

From the Table 2 results, I figured out that 48 hours time sequential length using only precipitation is the most reasonable with respect to all error analysis. These results suggests that 12 hours and 24 hours are short for LSTM hidden layer numbers causing high bias and variance. When comparing effects of including all weather information and using only precipitation, all weather information like temperature, wind speed, and weather description do not have positive effect

on predicting car demand. It shows that weather information is less correlated with future car demand, so it could cause deviation in prediction and overfit to uncorrelated data.
Thus, I will only use precipitation data among weather information and 48 hours time sequential length.

5.2 Time step

I experimented with different time step : 1 hours, 20 minutes, 10 minutes, 5 minutes. The number of dataset changes depending on the time step. Thus, I use only 26280 train set, 6469 test set to compare effect of time step excluding effect of dataset size. Except that, the other parameters are set as 1 depth of LSTM, 48 time sequential model length, without categorization, 100 epochs, and 100 batch size.

Table 3: **Error result of different time step**

error	1 $hours$	20 $minutes$	10 $minutes$	5 $minutes$
$R2(y \,\&\, \hat{y})$	0.97	0.99	0.99	0.99
$mean\ absolute\ error$	756.61	189.76	96.97	45.96
$root\ mean\ squared\ error$	1093.09	261.60	136.46	60.79
$mean\ absolute\ error\ percentage\ test\ set(\%)$	6.80	5.53	4.98	5.41
$mean\ absolute\ error\ percentage\ train\ set(\%)$	6.25	4.96	4.56	4.75

As seen in Table3, 10 minutes time step shows the most accurate prediction with respect to bias and variance. These results demonstrate that predicting car demand using previous 480 minutes is the most effective : more than 480 minutes can cause overfit to useless data and less than 480 minutes can cause missing crucial data for prediction. The prediction result using all possible data set (157680 train set, 39355 test set) with 10 minutes time step is in the Table 4.

Table 4: **Error result of 10 minutes**

	$R2(y \,\&\, \hat{y})$	MAE	$RMSE$	$MAE\ percentage\ test\ set(\%)$	$MAE\ percentage\ train\ set(\%)$
$error$	0.99	71.19	98.94	3.82	3.74

5.3 Categorized data

We can assume that depending on the day and time, future car demand might have different correlation with previous demand. I experimented with different categorized data : sorting by forecast day (from Sunday to Saturday) and forecast time (from 00:00 to 23:00). The other parameters are set as 1 depth of LSTM, 1 minute unit time step(to get enough amount of dataset for excluding effect of dataset size), and 48 time sequential model length, 100 epochs, 100 batch size.

5.3.1 Day

I use 225257 train set, 55543 test set. And here is the error result when categorized by day and without categorization.

Table 5: **Error result categorized by day and without categorization**

error	$Monday$	$Tuesday$	$Wednesday$	$Thursday$	$Friday$	$Saturday$	$Sunday$	$without$
$R2(y \,\&\, \hat{y})$	0.98	0.98	0.98	0.98	0.98	0.97	0.97	0.98
$mean\ absolute\ error(MAE)$	12.31	12.34	12.40	12.33	12.32	13.03	12.57	13.62
$root\ mean\ squared\ error$	16.10	16.13	16.24	16.21	16.19	16.98	16.45	17.69
$MAE\ percentage\ test\ set(\%)$	7.73	7.63	7.53	7.43	7.36	8.10	7.60	7.39
$MAE\ percentage\ train\ set(\%)$	6.77	6.53	6.57	6.64	6.63	6.55	6.43	6.57

5.3.2 Time

I use 65700 train set, 16393 test set. And here is the error result when categorized by time and without categorization.

From the results Table 5, I figured out that categorizing data with respect to day does not improve the prediction accuracy. However as seen in Table 6 and 7, when the data is categorized with respect to time, the prediction accuracy enhanced for all times. Thus, we can conclude that car demand has different correlation with previous historical demands depending on the time not the day. (e.g. car demand at 4:00 PM is significantly influenced by 9:00 AM car demands. However, car demand at 4:00 AM is rarely affected by 9:00 PM car demands.)

Table 6: **Error result categorized by time**

error	00 : 00	01 : 00	02 : 00	03 : 00	04 : 00	05 : 00	06 : 00	07 : 00	08 : 00	09 : 00	10 : 00	11 : 00
$R2(y \& \hat{y})$	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
$mean\ absolute\ error(MAE)$	12.20	12.17	12.16	12.13	12.22	12.17	12.12	12.17	12.20	12.12	12.19	12.19
$root\ mean\ squared\ error$	15.95	16.04	15.92	15.82	16.03	15.89	15.88	15.98	15.93	15.90	15.90	16.03
$MAE\ percentage\ test\ set(\%)$	6.88	6.83	6.81	6.83	6.93	6.83	6.83	6.87	6.93	6.82	6.79	6.86
$MAE\ percentage\ train\ set(\%)$	5.62	5.58	5.59	5.59	5.64	5.56	5.56	5.60	5.62	5.58	5.62	5.59
error	12 : 00	13 : 00	14 : 00	15 : 00	16 : 00	17 : 00	18 : 00	19 : 00	20 : 00	21 : 00	22 : 00	23 : 00
$R2(y \& \hat{y})$	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
$mean\ absolute\ error(MAE)$	12.23	12.13	12.19	12.18	12.19	12.14	12.15	12.21	12.19	12.19	12.14	12.18
$root\ mean\ squared\ error$	15.98	15.80	15.89	16.01	16.01	15.95	15.95	16.03	16.01	16.02	15.83	15.88
$MAE\ percentage\ test\ set(\%)$	6.84	6.84	6.81	6.82	6.88	6.87	6.81	6.84	6.86	6.82	6.83	6.83
$MAE\ percentage\ train\ set(\%)$	5.58	5.58	5.60	5.58	5.59	5.58	5.59	5.59	5.57	5.57	5.56	5.57

Table 7: **Error result without time categorization**

	$R2(y \& \hat{y})$	MAE	$RMSE$	$MAE\ percentage\ test\ set(\%)$	$MAE\ percentage\ train\ set(\%)$
$error$	0.98	13.31	17.28	8.24	6.86

However, when categorizing with respect to time, the data size will decrease remarkably. In the prediction using LSTM, data size is the most crucial for prediction accuracy. Even though categorization with respect to time enhance prediction, I do not have enough data of 10 minutes time step when categorized with respect to time. I would not use time categorization in future experiments.

5.4 Depth of LSTM & Attention Layer

Definitely, car demand is influenced by differently weighted historical data (e.g. car demand at 4:00 PM is affected higher by car demand at 8:00 AM and 3:00 PM than by car demand at 11:00 AM and 1:00PM). Thus, we can predict that the more complex LSTM structure can enhance the prediction of car demand. I experimented with Depth of LSTM : 1 depth, 2 depth, 3 depth. Also I experiment with an attention layer which consider relevance between arguments : architecture with an attention layer is that inputs go through a LSTM layer, a attention model and another LSTM layer sequentially. Except that, the other parameters are set as 10 minutes, 48 time sequential model length, without categorization, 100 epochs, and 100 batch size.

Table 8: **Error result of different depth of LSTM**

error	$Attention\ layer$	$1\ depth$	$2\ depth$	$3\ dept$
$R2(y \& \hat{y})$	0.99	0.99	0.99	0.99
$mean\ absolute\ error$	73.35	71.19	69.29	67.77
$root\ mean\ squared\ error$	102.75	98.94	97.57	95.81
$mean\ absolute\ error\ percentage\ test\ set(\%)$	3.82	3.82	3.61	3.49
$mean\ absolute\ error\ percentage\ train\ set(\%)$	3.70	3.74	3.66	3.61

From the results in Table 8, I found out that an attention model is not effective in predicting car demand. I could assume that predicting car demand does not require remembering early data in long sequences, which is the benefits of an attention model. However, we can easily see that stacked LSTM structure, utilizing sequences of LSTM output as input of next LSTM, can enhance the accuracy of prediction. Thus I can infer that stacked LSTM can make more effectively different weights on each time history.

6 Conclusion/Future Work

In this paper, I presented LSTM models which predict car demands in New York City. I experimented with various parameters (Time sequential model length, Input features, Time step, Categorization and Depth) of LSTM. And I tested an attention model for more accurate prediction. From these experiments, I figured out that 48 time sequential length, only precipitation weather data, 10 minutes time step, categorization with time and 3 depth of LSTM are the optimal data and architecture structure for car demand forecast.

This paper opens several new approach of research. First, it is of interest to predict car demand with other influential information (e.g. events, news). Second, I would like to develop algorithms which predict more specific information of car demand (e.g. customers requests area information and destination information).

References

[1] Ramon Iglesias, Federico Rossi, Kevin Wang, David Hallac, Jure Leskovec, and Marco Pavone. Data-driven model predictive control of autonomous mobility-on-demand systems. *arXiv preprint arXiv:1709.07032*, 2017.

[2] Nikolay Laptev, Jason Yosinski, Li Erran Li, and Slawek Smyl. Time-series extreme event forecasting with neural networks at uber. In *International Conference on Machine Learning*, 2017.

[3] Marco Pavone, Stephen L Smith, Emilio Frazzoli, and Daniela Rus. Robotic load balancing for mobility-on-demand systems. *The International Journal of Robotics Research*, 31(7):839–854, 2012.

[4] Xuan Song, Hiroshi Kanasugi, and Ryosuke Shibasaki. Deeptransport: Prediction and simulation of human mobility and transportation mode at a citywide level. In *IJCAI*, pages 2618–2624, 2016.

[5] Rick Zhang and Marco Pavone. Control of robotic mobility-on-demand systems: a queueing-theoretical perspective. *The International Journal of Robotics Research*, 35(1-3):186–203, 2016.

[6] Rick Zhang, Federico Rossi, and Marco Pavone. Routing autonomous vehicles in congested transportation networks: Structural properties and coordination algorithms. *arXiv preprint arXiv:1603.00939*, 2016.