# WDA: Estimation of Travel Time for Taxi Using Wide and Deep Learning Together with Target Attention

**Yuxing Wang**

Civil Aviation University of China, Computer science and technology, Tianjing, 300300, China

yuxingw2000@outlook.com

**Abstract.** There is an increasing interest in ways to use data and machine learning methods to optimize the operation of transportation system. In this paper, the author focuses on estimation of time arrival (ETA) for taxi. As taxi is one of the popular transportation vehicles in the urban area, predicting the time of arrival for taxi would help the customer to better estimate time required for the trip, especially in case of urgencies. The author develops a comprehensive neural network with both wide and deep components, and attention component for this predictive job. The author also evaluates the network with huge historic data and compare it with industry existing solutions. This network model is proven to have a better prediction accuracy under extreme traffic and weather condition with reasonable shorter training time.

**Keywords:** estimation of time arrival (eta), wide and deep learning, target attention, gate mechanism.

## 1. Introduction

Location-based service (LBS) was developing with fast expansion and becoming much more important, ride and hail service of taxi is one of the key applications for LBS transportation system. In this paper, the author will concentrate on problem of estimation of time arrival (ETA) for taxi, where a precious ETA can help enhancing the efficiency of the transportation, and reducing the cost of travel for individual customer, as well as the air pollution. ETA has become a core component that influences decision-making at different stages of the online ride-hailing process, including route selection, vehicle dispatch, carpooling [1] . Meanwhile, the travel time is the major concern between rider and driver to reach a deal, better estimation improves customer satisfaction, especially for companies providing cabs service, like Uber, Lyft, Didi Chuxing.

Estimation of time arrival, as the name suggests, is simply defined as the forecasted time cost between the origin and destinate locations by any vehicle (bike, bus, taxi…) or even walking, in Figure 1 from Google map, there is a route in NYC with estimation of travel time by driving.

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

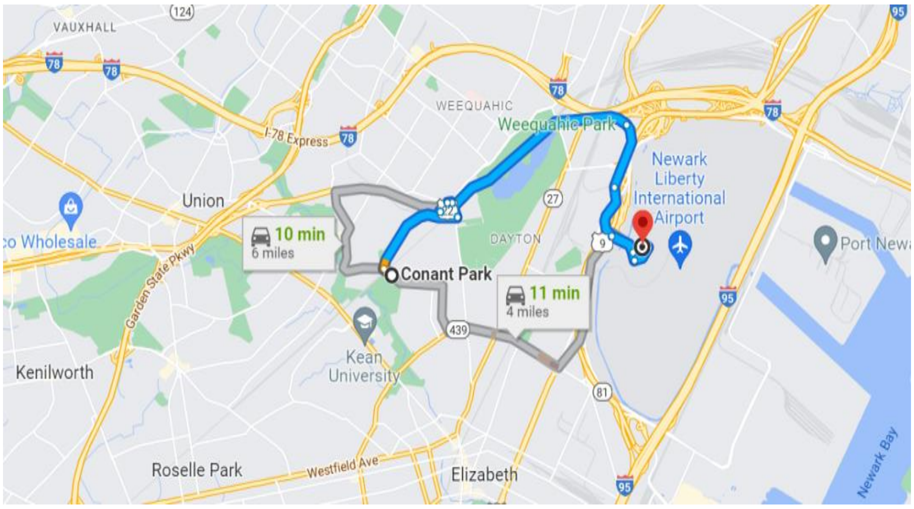

**Fig. 1.** The sample navigation in New York city from Google Map, the origin is from Conant Park, and destination is Newark Liberty International Airport. Google map provides different recommendations of route and also the estimation of time arrival in case of driving.

Travelling by taxi is one of the most vital way in our life, especially for people in modern big cities and crowd area, public research "The New York City Taxicab Fact Book" documented (Figure 2):

Combining taxis, black cars and car services, there were 6.3 taxis/liveries per 1,000 New York City residents in 2005. This figure is greater in New York City than in any other major U.S. city except Washington DC, which does not limit the number of taxicabs

Focusing on Manhattan, which accounts for over 90% of taxi trip origins, there were 8.5 yellow medallion taxis per 1,000 Manhattan residents in 2005

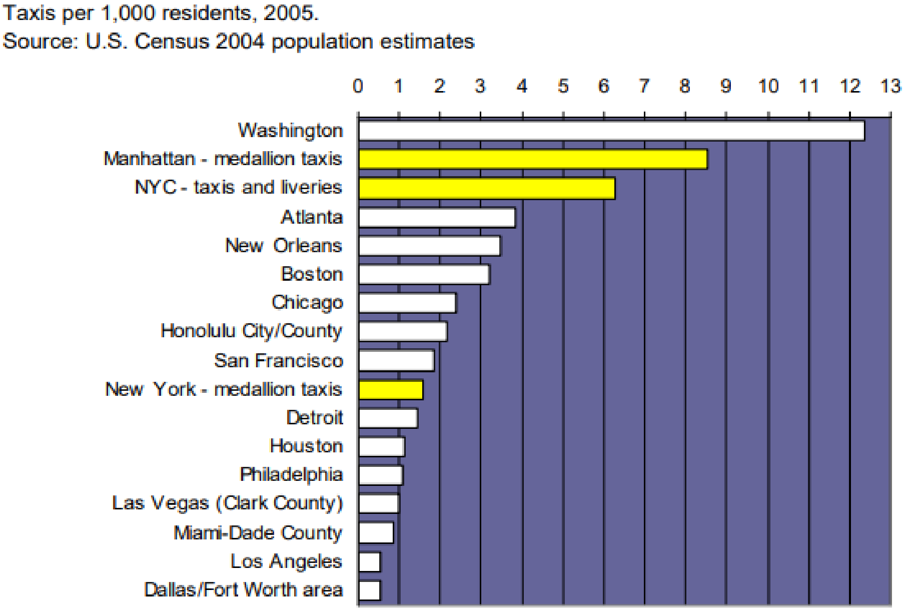

**Figure 2.** Ratio of taxicabs to population in cities with 1,300 or more cabs.

Travel time estimation problem has been widely investigated in the industry, the author can conclude the existing solutions into two main approaches, geographic based and data driven.

Geographic based approach relies on GPS data, which is the foundation of intelligent transportation system, modern GPS system collects the geographic location data with super high resolution and accuracy. This approach utilizes historic GPS data of travel individual trips, where the entire route will be separated into several pairs of <road, turn>, the travel time is the summary of the transportation time of each road and waiting time at the interconnected turn. Therefore, the forecasting of ETA can be divided into sub- problems, the simplest way is to use the historic data to extract average transportation time of road and waiting time of turn to estimate the overall travel time [2] , and Path Travel Time Estimation of sparse trajectory using tensor decomposition was proposed by [3] , which are with similar methodology. Three major drawbacks are analyzed: 1) the complexity of dynamic change for transportation cannot be easily patterned to route factor. 2) personalization of driver influences the ETA significantly. Behavior of driver, like average driver speed, route selection preference is not considered in this approach. 3) accumulated error increases due to the separation of problem, where sequential effect is neglected.

The second approach is to combine the ETA problem with the machine learning method, which became the most powerful tool to handle the prediction problem in the industry when researching with the explosion of the data warehouse. Compared with geographic based approach, more studies tend to treat the predicting as an entire route problem, the method in [4] models the travel time for given routes into a time series, using histrionic data to predict the ETA by different pickup time period during a day, and through the workday or weekend. Nevertheless, there are still following major issues of this approach: 1) the prediction is used normally with stable and uncomplex traffic and road condition, when the complexity of travel increased, the data coverage becomes a problem; 2) quite many important information is ignored in the model, such as weather and driver’s personal driving behavior, which makes the method stay low accuracy of time estimation. 3) as most of intelligent traffic system research shows, the traffic is spatial-temporal factor dependent network, simple model cannot correctly weight these factors in the model.

No matter geographic-based, or data driven achieves the good effectiveness of estimating the travel time, they have weak generalization capability and insufficient information utilization. In taxi ride-hailing system, this loss of prediction even extends, due to the fact that travel factors influence ETA are much more complex.

## 2. Factor categorization and analysis

The author realizes these four key categories of factors impacting the ETA of taxi service the most:

Spatial factors: Geospatial complexity where the trip happens highly impact the travel time of taxi, such as the information width of traffic block, traffic light, inter-section numbers, number of lanes, the length of route.

Temporal factors: ETA of taxi system has strong relationship with temporal conditions, the trip starts in rushing hour would have longer travel time than the rest of day by nature where origin and destination points are identical. And the rush and off-peak hours definition is different in workday, weekend, and holidays. In figure 3-4 below, the taxi travel time of a crowd sub-route in New York City urban area is used, to prove the clear pattern that people need longer travel time during workday than in weekend, and in rushing hours compared with off-peak periods.

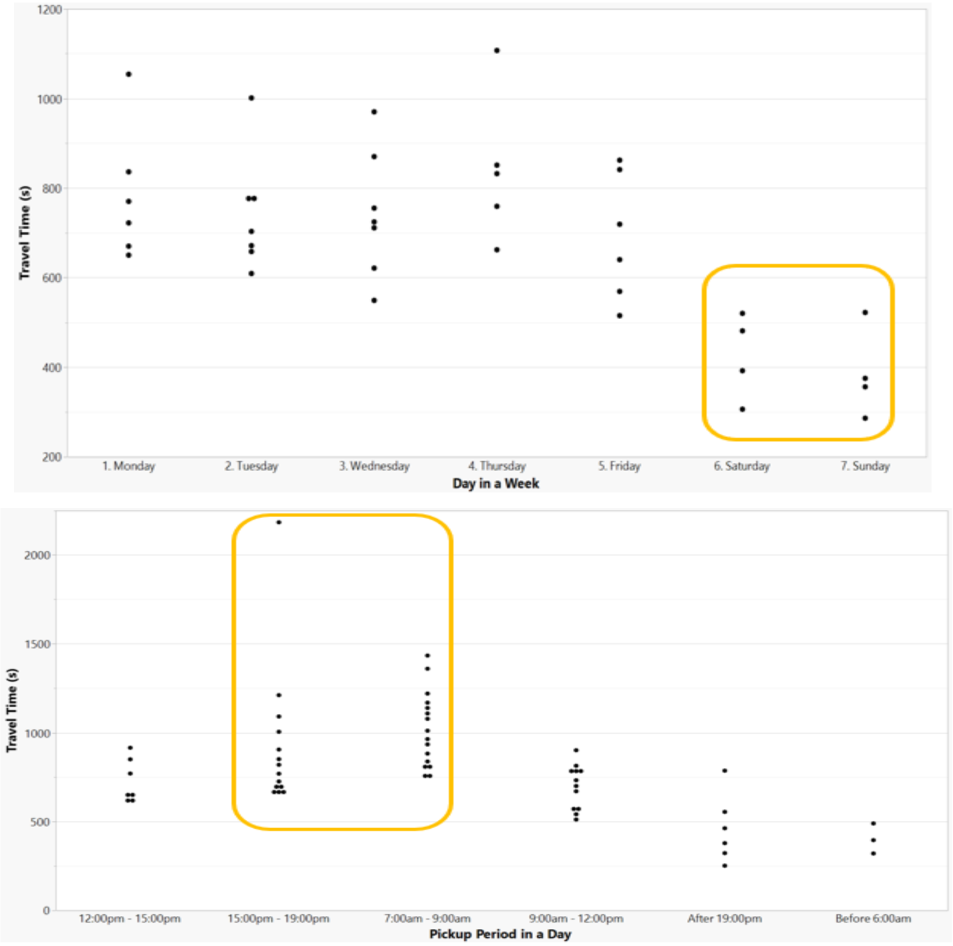

**Figure 3-4.** Travel Time Vs Workday and Pickup time pattern.

Personal factors: Driver driving habit plays the dominating role in taxi system, including driver’s average driving speed, taxi profile of driver. These kinds of factors can be studied through the driver Id, since driver Id is the unique identifier of drivers themselves.

Global factors: Travel time also has crucial dependency with weather condition of trip, research points out the taxi level of service declines remarkably especially in the region of low rain [5] , Figure 5 provides the statistics graph of the travel time cost with rough identical trip at off-peak hours in different weather conditions, without any surprise that travel time is almost doubled under snow or heavy rain compared with clear weather.

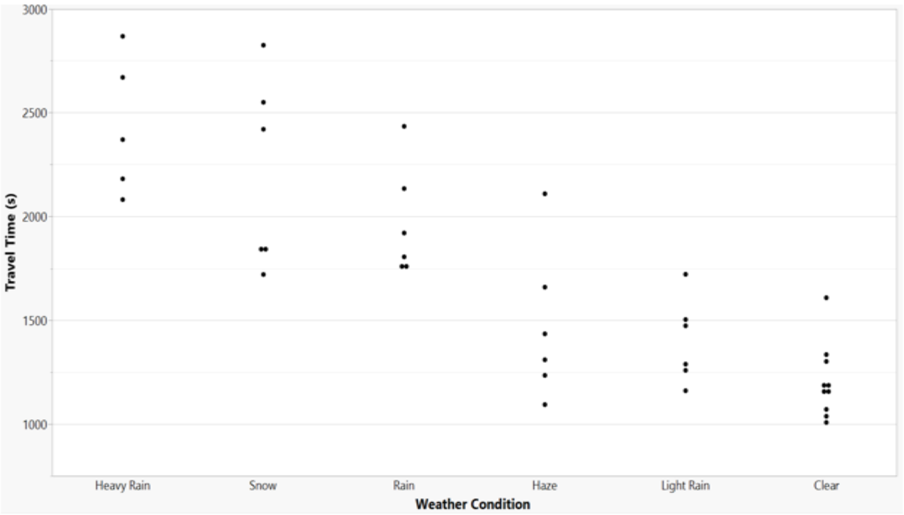

**Figure 5.** Weather impacts on travel time.

Moreover, the difficulties in travel time prediction also come from the fact that those factors are always not independent variable in the model, but inherently combined. For example, whether driver has better cognition of the spatial information for a particular route, experienced and non-experienced drivers could have contrasting performance of travel time cost [6]  The other good example could be that the bus dedicated lane is allowed to be reused by taxi in special period of time during a day in downtown area, where spatial and temporal information works together, that might slightly reduce the travel time.

## 3. Method

### *3.1. Prediction model*

Prediction refers to the procedure that applies mathematics method to pattern current and historical data in order to predict the activities of future.

Similar as prediction analytics, machine learning is the toolset that uses training and testing algorithms to look for patterns from large amount of data, even without knowing what to look for. ETA problem of taxi is a kind of supervised learning in machine learning, our prediction goal is with known ETA data $Y \in (Y_0, Y_1 \dots Y_n)$ and factor set $X \in (X_0, X_1 \dots X_n)$, where $X_n$ is the factor table for $Y$ with several dimensions, to build the model, and apply this model to predict ETA of $X_m$, where $X_m$ does not belong to known set X. The model can be described as a function of factor set $f(X_i)$.

Three key statistic values are usually used to measure the bias and how good the model matches with actual data: With the known $Y_i$, Mean Absolute Percentage of Error (MAPE) is the metric mostly applied in machine learning algorithm, together with Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), in validation of machine learning model, the target is to minimize these measurements, referring to Eqn. (1 - 3)

$$MAPE = \sum_{i=1}^{N} \frac{|Y_i - f(X_i)|}{Y_i} \tag{1}$$

$$MAE = \sum_{i=1}^{N} |Y_i - f(X_i)| \tag{2}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \left(Y_i - f(X_i)\right)^2}{N}} \tag{3}$$

### *3.2. Previous works*

Decision tree or statistics-based methodologies are not the best fit model for travel time prediction, due to the reason that precious models must come from huge historic data set, efficiency and capability of XGBoost [7] or SVM [4] are limited. In the past few years, the most dominant industry solutions focus on using deep learning network for prediction jobs, the creative ideas like DeepTTE [8] , HetETA [9] , WDR [1] have been proved to have better performance on prediction through mass routes, traffic and driving factors. Nevertheless, they either have weakness in the model to deal with sequential factor, which is high sensitivity to travel time, or using CNN, RNN (LSTM) that requires heavy training and inference resource, which is difficult to be parallelized in computing.

### *3.3. Related research*

Wide and deep network: the author noted that the travel factors can be either dense or sparse, dense factors are continuous numbers, such as number of lanes for a route, length of the traffic block. Sparse factors are discrete categorical values, for example driver Id, day in a week. In machine learning method, dense factors regression is built in linear mode, which can be described as Eqn. (4)

$$y = w^t x + b \tag{4}$$

where y is the prediction output itself, x is the factors tables $x = [x_1, x_2 \dots x_n]$, w is the weight parameter of factor, and b is the bias.

Before being feed into neural network for learning, module needs to be extended by cross product transformation, meaning adding the interaction relationship between the individual factors.

Linear regression cannot be directly applied to the sparse factors, it requires the converting operation that maps the high dimensional factors into low dimension dense like vectors, the operation is called embedding. This factor embedding layer usually converts the sparse factor into dense factor vector in the order of 20 to 256.

To combine dense and sparse factors in a single network, Google proposes (Figure 6) the wide and deep learning for recommendation system, which has similar complexity of dense and sparse factors with the ETA prediction, the author reuse it in the prediction network model, which consists of the rest of major components: 1) cross production transformer for both dense and sparse factors. 2) embedding layer for sparse factor convert. 3) concatenation in a layer between dense and sparse factors. 4) a Multi-Layer Perceptron (also known as Feed Forward Network) for deep learning. 5) back propagation mechanism to decline gradient, with compute the loss of model in validation flow. As factor analysis presents, travel time prediction is a perfect example of what the wide and deep component learning should apply: the historic data is huge in size; the job itself contains an ensemble of different dense and sparse factors; we cannot simply optimize network for a single specific task but instead for a blend of different metrics and theirs links in between.

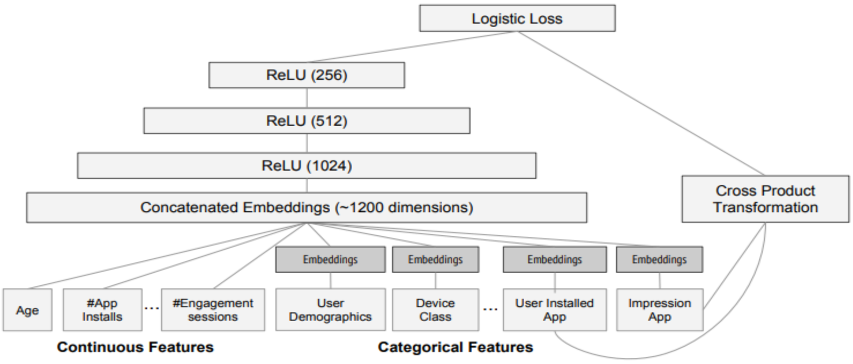

**Figure. 6.** Google wide and deep learning for recommendation system.

Sequential factor learning and attention network: travelling is not measured by route between origin and destination in theory, changes of short-term local traffic factor highly impact the result of prediction. These sequential factor in travel time problem is extracted from the road map, where the route of trip can be described as the series of traffic blocks $R = (b_1, b_2, \ldots b_n)$, where $R$ is the route, and $b_i$ is i-th traffic block the route goes through, block_id is numbered as the identifier, these traffic blocks in a route are in sequential manner. The Figure 7 below represents the route from center of Brooklyn to Maria Hernandez Park in Google Map, the route highlighted is formed with ten different traffic blocks (one marked in orange color). The length of block, the traffic condition in inter-section of blocks are all sequential factors in this case.

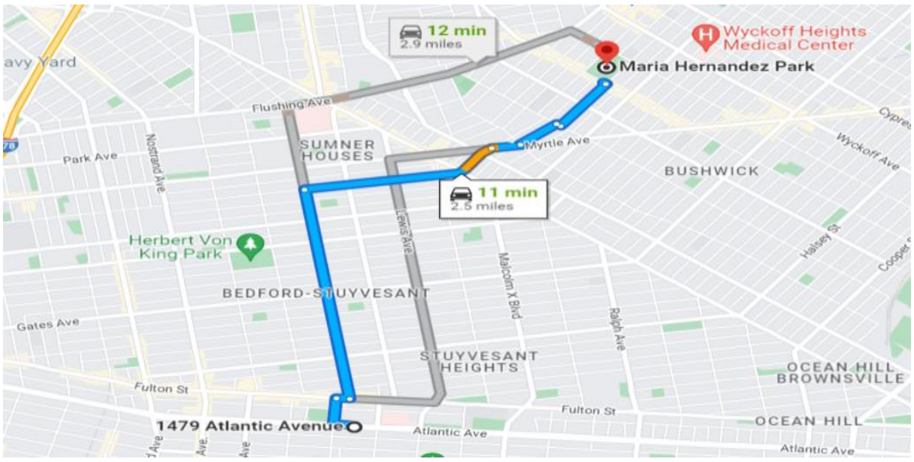

**Figure 7.** Route consists of several traffic blocks.

Wide and deep model does not fit with sequential data study, some researchers provides different types of recurrent neural network as the solution. For example LSTM [1] , which is the widely used RNN model for nature language sequential to sequential translation with forget, input and output gates, the model is described in Figure 8; GRU [10] (Gated Recurrent Unit) is the optimized version of LSTM combined forget and input gate to an update gate (refer to Figure 9 for gate micro structure), this variation is preferred for study of smaller dataset compared with LSTM, and proved to have better performance.

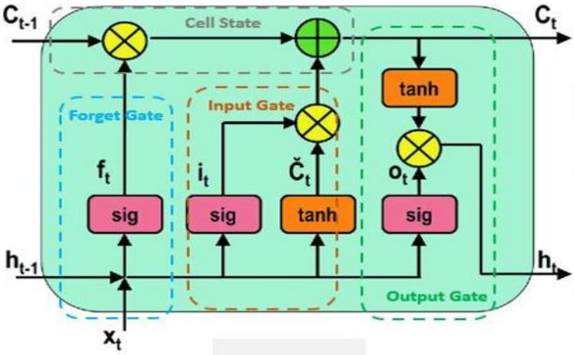

**Figure 8.** Long short-term memory.

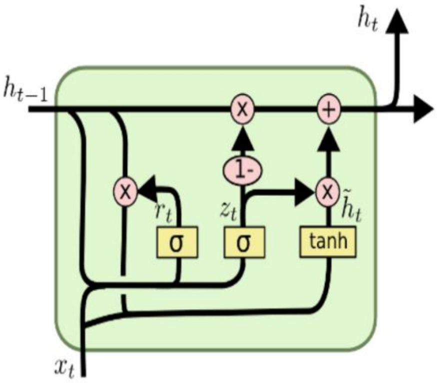

**Figure 9.** Gate structure in GRU.

RNN has its nature weakness on supporting the parallelism computing capability which modern GPU and accelerator hardware provide, since input of state in RNN hn relies on output of previous states (h1, h2…hn-1). Attention mechanism called transformer was developed to replace RNN in translation application by Google [11] in 2017, they innovatively used full attention-based network rather than using attention mechanism through encoding and decoding procedure in RNN.

Attention is built on three types of vectors <query, key, value>, where query is a set of vectors model wants to calculate attention for. Key is a set of vectors to calculate attention against. Through the dot product multiplication, a set of weights was added to present how strong attention the network should pay each query against key, then multiply it by value to get resulting set of target vector. The flow of Multi-Head Attention proposed by Google for nature language translation is inline in Figure 10.

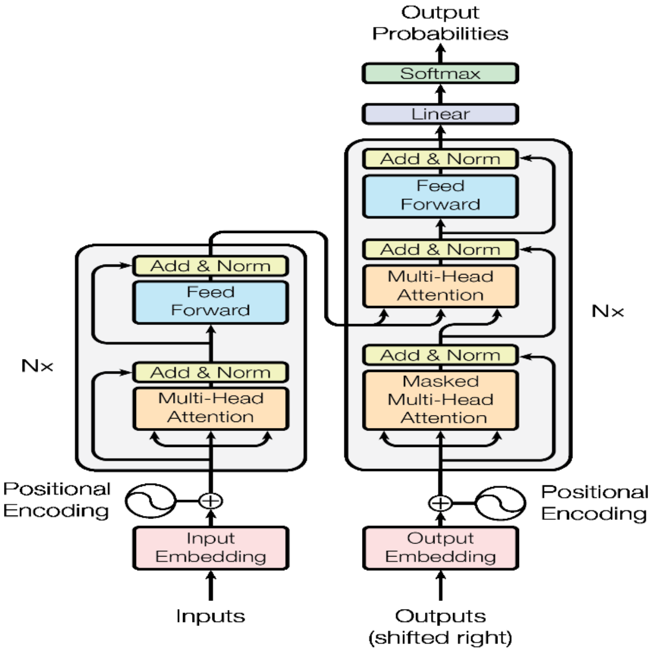

**Figure 10.** Multi-head Attention flowchart.

Personalized Cold Start Modules (POSO): In recommendation system, there is the problem that when new user comes in with cold start without enough initial data, the system might fail to balance various distributions because of data imbalance, even though personalized features are studied [12].

$$\hat{x} = C \sum_{i}^{N} [g(x^{PC})]_i f^i(x) \tag{5}$$

This is general gating network prototype equation, where $x$ and $\hat{x}$ are adjacent layers, $\left[g\left(x^{PC}\right)\right]_i$ is the gate of personalized feature representing the i_th weight. $f$ denotes the module itself.

POSO is a gating network designed to solve the critical feature flooding within the imbalance dataset where: 1) Critical feature itself is assigned with gate mechanism (refers to Eqn. 5), multi- features will be proceeded with equality same gating network, none of them would be neglected, due to the reason POSO uses another set of modules and gates to make predictions. 2) POSO highlights the weights matrix directly through raw features instead of going through the hidden activations. POSO can work with most of working network like MLP, MHA shown in Figure 11 (a,b), to bring in so-called personalization, it can be understood as the target attention when there are many of critical features impacts the prediction, and high number of bias points in the distribution.

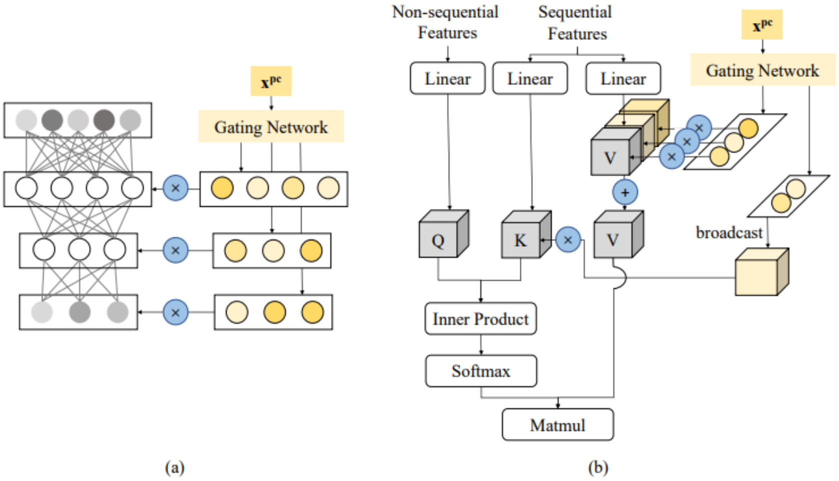

**Figure 11.** Personalized modules by POSO: (a) POSO (MLP) masks each activation in each layer respectively. (b) In POSO (MHA), $Q$ is not personalized, $K$ is lightly personalized and $V$ is totally personalized [12] .

### 3.4. Our proposal

In this paper, the author proposes an optimized network consists of both wide & deep learning together and target attention gate based sequential factor processing network, in short WDA. The contributions of this paper are: 1) using wide and deep learning network to learn mass dense and sparse factors in taxi travel time prediction. 2) employing target attention for global factors like weather, weekday under gate mechanism against sequential traffic blocks factors. 3) using the large real dataset to build and validate the proposed model, and comparing with existing solutions for deeper analyzing the pros and cons. The schematic of proposed network is shown in Figure 12:

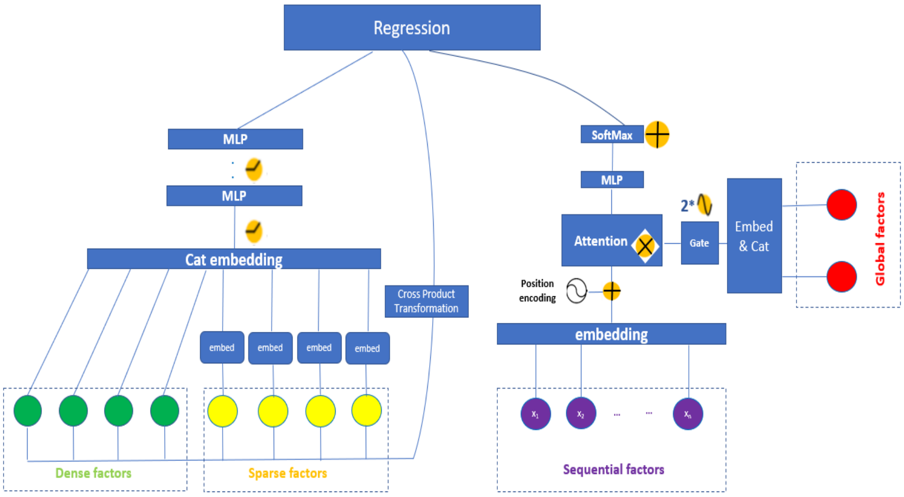

**Figure 12.** Wide and Deep Learning with Target Attention.

The detailed flow of algorithm can be described in 8 main steps:

Step 1: the wide part of algorithm converts the input dense factors which have strong correlation with travel time into embedding space vectors, such as the overall distance of trip. The vectors consist of linear components which are extracted directly from raw input factors and cross-product non-linear components, these non-linear components represent the relations between raw input factors as the Eqn. 6:

$$f(x_i, x_2 \dots x_n) = \sum_{i=1}^{n}(w^t x_i + b) + \prod_{i=1}^{d} x_i^{ck_i} \tag{6}$$

where $ck_i \in \{0,1\}$ is the Boolean value, 1 means $x_i$ which is the part of certain cross product transformation, 0 stands for not included.

Step 2: the deep neural network is in use to process factors in categorical manner. Before feeding into network, each of these high-dimensional sparse factors will go through embedding flow, in which these factors are mapped into a low-dimensional dense value-based vectors, the size of vector in our implementation is fixed to 8. Random non-identical weight values are initialized and will be updated by the training process targets to minimize the loss.

Step 3: dense input factor vectors and embedded sparse factor vectors are then combined in one layer with concatenating function.

Step 4: the combined factor vectors are feed into multi-layer network for learning. In the experiment, the author installed 16 hidden layers and used ReLU as activation function. The final prediction includes outputs from both dense and sparse factors.

These four steps are for wide and deep learning on both traffic dense and sparse factors. The attention transformer for global factors against sequential traffic block factor will run in parallel and described in Step5-7 below:

Step 5: global factors like weather, weekday… are embedded into an 8-dimensions vector and concatenated, the output of embedding layer goes through a gate which is 2*sigmoid function and then feed to attention operation, this is the global factor vector, and query in attention network. The sequential traffic block factors are embedded, each traffic block is projected into an 8-dimensions space. To take in spatial information, the author reused the position encoding from [11] with same number of dimensions, output sequential factor vector is the key of attention network, and travel time is value.

Step 6: target attention is elemental multiply of sequential factor vector and global factor vector, then feed into a 2-layers MLP with ReLU activation function.

Step 7: Softmax is applied to obtain the weight values after querying (global factor vector) all the keys (sequential factor vector) from attention network.

Finally, the regression function (1-layer MLP) is in place to combine the output from wide-deep learning and attention network for weight adjustment.

Compared with existing network models, the author believes the proposal will achieve the similar prediction accuracy as wide-deep model and beats it on training and inference speed if it combines with RNN. Moreover, our target attention network would provide better accuracy on the extreme travel condition which is the industry pain point of ETA prediction.

## 4. Experiment

### *4.1. Dataset*

The dataset used to evaluate our model in this experiment is from Kaggle, which records a part of public historic taxi pick-up and ride data in New York city 2016 published by NYC Taxi and Limousine Commission (TLC), together with extended weather and holiday data, in the evaluation:

Overall 2M data points of individual trips with the travel time are used, and key factors are listed as the Table 1 below:

80% of dataset is applied for training, 10% for validation and 10% for testing

**Table 1.** Factors in travel time prediction.

| factor | factor types |
|---|---|
| Origin zone | Sparse |
| Destination zone | Sparse |
| Length of route | Dense |
| Number of passengers | Dense |
| Driver Id | Sparse |
| Average Speed | Dense |
| Pickup time | Sparse |
| Drop time | Sparse |
| Traffic blocks | Sequential |
| Weather | Sparse |
| Weekday of ride | Sparse |
| Is holiday? | Sparse |

The trips with travel time below 60 secs or whose average speed higher than 100km/h are removed from experiment as abnormal data points.

### *4.2. Experiment setting*

The author implements the proposed model with TensorFlow.Keras and run it on a system with 1 Nvidia A100 80GB GPU card, which contains 6912 CUDA cores.

The model is compared with other baseline implementation SVM, GBDT (XGBoost) and WDR (wide and deep recurrent with LSTM) using same dataset and hardware setup. Our comparison results are listed in Table 2:

**Table 2.** Comparison result.

| | MAPE (%) | MAE (sec) | RMSE (sec) |
|---|---|---|---|
| SVM | 14.22% | 78.48 | 135.66 |
| GBDT (XGBoost) | 13.62% | 62.33 | 119.20 |
| WDR (LSTM) | 11.27% | 48.15 | 92.47 |
| WDA (our proposal) | 11.23% | 46.29 | 88.25 |

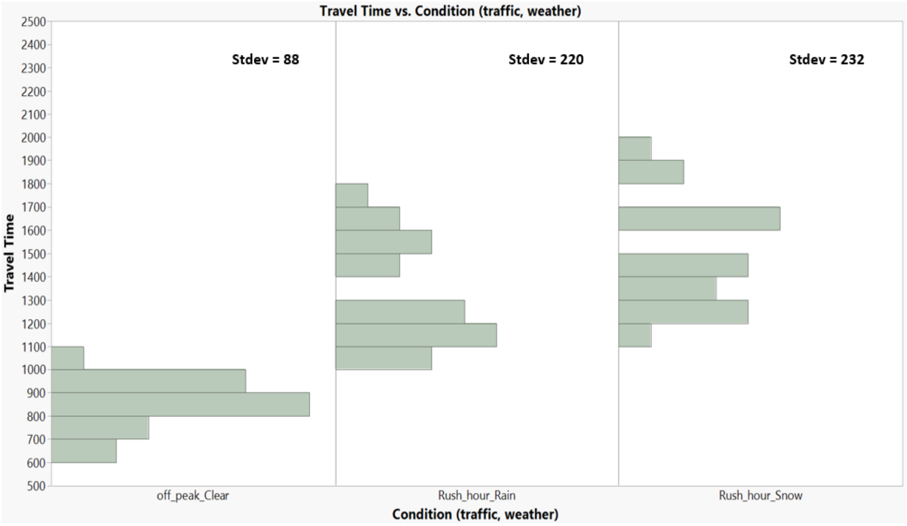

**Figure 13.** Distribution of Travel time vs different weather and traffic conditions.

From the result, our model beats SVM and GBDT on all measures, and reaches the similar MAPE with WDR network, The author also notifies that the training time used by our model is less than WDR by ballpark 8%, that is benefited from capability of parallel computing utilization in our target attention network on sequential factors learning. Furthermore, the author can see the model perform slightly better on absolute error of prediction, due to the reason attention mechanism has its strength to quantify the impact of global factors against traffic blocks, especially predicting travel time in extreme case with wider data distribution. In the example, the author realizes that the travel time under snow or rain weather at rushing hour has much wider distribution compared with it is under clear weather at off-peak time.

**Table 3.** Prediction comparison under extreme condition.

| | MAPE (%) | MAE (sec) |
|---|---|---|
| WDR (LSTM) | 16.5% | 92.62 |
| WDA (our proposal) | 13.2% | 51.25 |

In Figure 13 above, each random trip was extracted with the combination of traffic condition (off-peak, rush_hour) and weather (Clear, Rain, Snow) for an identical traffic route. The travel time distribution can be used is much wider under bad weathers in rushing hours, attention network aims to solve the problem of mis-prediction in this situation. For better apple-to-apple comparison, the author picked up ~3000 trips in testing dataset which are either under rain/snow, rushing hours, competes on the prediction accuracy between our model and WDR, as MAE has pronounced robustness when there are high outliers in the target dataset, MSE is ignored, the result is listed in Table 3.

## 5. Conclusion

In this paper, the author proposed the comprehensive neural network to estimate travel time of taxi based on wide deep learning and attention network. This idea is simple but effective compared with existing industry solutions, where wide and deep learning targets to process danse and sparse factors of travel time, and attention network focuses on exploring the influence of global factors like weather against the sequential traffic blocks. Through volume experiment on huge dataset, the proposed WDA performance on prediction reaches the equality level of existing WDR model with short training and inference time cost, and more important, WDA outperforms all existing models on the travel time predicting under “chaos” traffic and weather condition, which is the pain point where customer cares the estimation accuracy the most. As attention mechanical based network is used widely in various application, like image recognition and classification, prediction problem could be also solved by full attention network, instead of wide and deep components, this state-of-art network eases the flow of factor analysis, and author believes it is the future development of Taxi ETA problem.

## References

[1] Zheng Wang, Kun Fu, and Jieping Ye, “Learning to estimate the travel time,” in SIGKDD, 2018, pp. 858–866.

[2] Wang-Chien Lee, Weiping Si, Ling-Jyh Chen, Meng Chang Chen “A New Framework for Bus Travel Time Prediction Based on Historical Trajectories” In Proceedings of the 20th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems.

[3] Yilun Wang, Yu Zheng, and Yexiang Xue, “Travel time estimation of a path using sparse trajectories,” in SIGKDD. ACM, 2014, pp. 25–34.

[4] Chun-Hsin Wu, Jan-Ming Ho, and D. T. Lee “Travel-time prediction with support vector regression” IEEE Transactions on Intelligent Transportation Systems 5, 4 (Dec 2004), 276–281.

[5] Sun, J, Dong, H, Qin, G, Tian, Y “Quantifying the impact of rainfall on taxi hailing and operation” Journal of Advanced Transportation, 2020, 1– 14.

[6] Yoshiki Wakabayashia, Shuichi Itohb, Yota Nagamic. “The Use of Geospatial Information and Spatial Cognition of Taxi Drivers in Tokyo”. Procedia Social and Behavioral Sciences 21 (2011) 353–361

[7] Tianqi Chen and Carlos Guestrin “XGBoost: A Scalable Tree Boosting System” In Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’16).

[8] Dong Wang, Junbo Zhang, Wei Cao, Jian Li, and Yu Zheng, “When will you arrive? estimating travel time based on deep neural networks” in AAAI, 2018.

[9] Huiting Hong, Yucheng Lin, Xiaoqing Yang, Zang Li, Kun Fu, Zheng Wang, X. Qie, Jieping Ye “HetETA: Heterogeneous Information Network Embedding for Estimating Time of Arrival” KDD '20: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining August 2020 Pages 2444–2454

[10] A. A. Noman, A. Heuermann, S. A. Wiesner and K. -D. Thoben, "Towards Data-Driven GRU based ETA Prediction Approach for Vessels on both Inland Natural and Artificial Waterways," 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), 2021, pp. 2286-2291.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin, “Attention is all you need,” in NIPS, 2017, pp. 5998–6008.

[12] Dai, S., Lin, H., Zhao, Z., Lin, J., Wu, H., Wang, Z., ... & Liu, J. (2021). POSO: Personalized Cold Start Modules for Large-scale Recommender Systems. arXiv e-prints, arXiv-2108.