TLab: Second Place Solution Towards Traffic4cast 2020 Competition

Fanyou Wu[1,a,*], Yang Liu[2,b,*], Zhiyuan Liu[2], Xiaobo Qu[3], Rado Gazo[1], and Eva Haviarova[1]

[1]Purdue University, Department of Forestry and Natural Resource, West Lafayette, USA
[2]Southeast Univeristy, School of Transportation, Nanjing, China
[3]Chalmers University of Technology, Department of Architecture and Civil Engineering, Gothenburg, Sweden
[a]Email: wu1297@purdue.edu
[b]Email: 230179629@seu.edu.cn
[*]Equal Contribution and Communication Authors

ABSTRACT

The problem of the effective prediction for large-scale spatio-temporal traffic data has long haunted researchers in the field of intelligent transportation. Limited by the quantity of data, citywide traffic state prediction was seldom achieved. Hence the complex urban transportation system of an entire city cannot be truly understood. Thanks to the efforts of organizations like IARAI, the massive open data provided by them has made the research possible. In our 2020 Competition solution, we further design multiple variants based on HR-NET and UNet. Through feature engineering, the hand-crafted features are input into the model in a form of channels. It is worth noting that, to learn the inherent attributes of geographical locations, we proposed a novel method called geo-embedding, which contributes to significant improvement in the accuracy of the model. In addition, we explored the influence of the selection of activation functions and optimizers, as well as tricks during model training on the model performance. In terms of prediction accuracy, our solution has won 2nd place in NeurIPS 2020, Traffic4cast Challenge. The source codes are available in https://github.com/wufanyou/Traffic4Cast-2020-TLab.

1 Introduction

Real-time traffic state prediction is an essential component for traffic control and management in an urban road network. The ability to predict future traffic state (e.g., flow, speed) can help improve traffic conditions, fleet organization, utilization rate, and social welfare[1,2]. Essentially, the traffic prediction is a time series problem, which is performed based on the changes in historical demand. A representative time-series prediction tool is the recurrent neural network (RNN), along with its diverse variants[3,4]. Apart from the temporal dimension, the correlation in the spatial dimension is also extensively incorporated by many works. Regions that are close to each other or share similar land-use structures may exhibit a homogeneous demand pattern. Techniques widely applied in computer vision like convolutional neural network (CNN)[2,5] and the emerging graph-based networks[6–9] are often adopted. Furthermore, multi-source data are also introduced in some literature to allow for the external influencing factors, such as weather conditions and neighboring points-of-interest[10,11].

2 Data Description and Problem Definition

The organizer provides industrial-scale, real-world data for 3 entire cities (Berlin, Istanbul and Moscow) over a year period. The organizer divides each city into a 436×495 grid; each pixel of this grid represents a region of $100m \times 100m$. In this competition, the dataset is comprised of dynamic data (*e.g.*, traffic speed), and static data (*e.g.*, the number of entertainment amenities). In the dynamic dataset, there are a total of 288 frames each day, each frame representing the aggregated information with nine channels over five minutes, including the traffic volume,

speed, and an aggregated incident level channel (a higher value indicates a more severe incident) in each ordinal direction (*e.g.*, NW, NE, SW, SE). The static dataset describes the locations of road junctions and points of interest, such as food and drink, shopping, parking, transit, etc. The sample images are shown in Figure **??**. The data of volume,speed, and incident level are scaled to $[0, 255]$ through a min-max scaler. Missing values are represented by 0.

Problem: This challenge is a multi-task learning problem, *i.e.*, use the given historical data to predict traffic volume, speed, and incident level in each direction. Pixel-wise mean squared error (MSE) is used to evaluate the performance for ranking the submitted prediction results.

3 Solution

In this section, we will introduce most of the technical details for our solution along with some experimental results.

3.1 Models

3.1.1 Choice of Model Architecture

In the 2019 Traffic4cast Challenge, was adopted U-NET[12] as the basic model. This year, we introduce HR-NET[13] (see Figure 1) in the competition, where HR-NET is an advanced network architecture for image segmentation that has demonstrated extraordinary performance in many tasks.

Table 1 shows a comparison of two basic models. We can conclude that, in general, HR-NET performs better than U-NET. Therefore, for the final solution, HR-NET is adopted as our backbone architecture. In the following section, most experiment results are based on HR-NET.

Model	Berlin	Istanbul	Moscow
UNET-EfficientNetB3	1.3175e-3	0.9214e-3	**1.3701e-3**
HRNET-W18	1.2937e-3	**0.9100e-3**	1.3705e-3
HRNET-W48	**1.2919e-3**	-	-

Table 1. Comparison of models in terms of the MSE on the validation set of each city using same features.

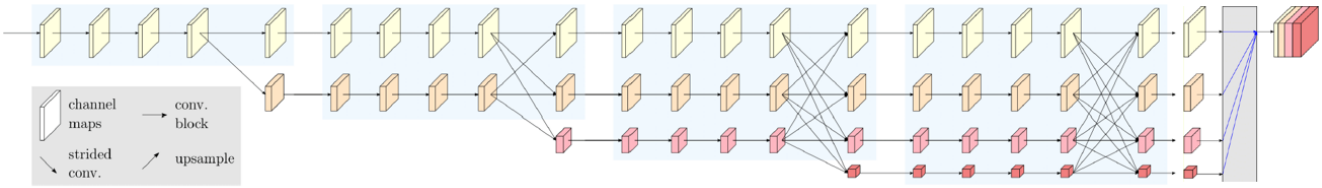

Figure 1. Visualization of HR-NET.

3.1.2 Choice of Hidden Layer Activation Function

The commonly used ReLU activation function is not the optimal activation function for many tasks, hence we experimented with some alternatives. Given limited RAM, we focused on in-place activation functions only. Table 2 lists the hidden activation functions we tested. ELU performed surprisingly better than any other activation function. Therefore, we used ELU as the hidden layer activation function in most places of the final solution.

3.2 Features

Essentially, this task is a time-series prediction problem. For one-dimensional time-series prediction, a typical solution is to transform the problem into a supervised learning problem through feature engineering. In addition to

Type	Berlin	Istanbul	Moscow
ReLU	1.2980e-3	-	-
ELU	**1.2951e-3**	-	-
ReLU6	1.2980e-3	-	-
LeakyReLU	1.2982e-3	-	-

Table 2. Comparison of the hidden layer activation functions in terms of the MSE on the validation set of each city using the same inputs.

the given 12×9 spatiotemporal channels and nine fixed spatial channels as the inputs, extra features (*e.g.*, periodic features, and holiday features) were also incorporated and valued. In summary, considering the decrease in MSE loss, we introduce three different types of features in this section.

3.2.1 Periodic features

The similarity between traffic states of two different days can be attributed to the periodic characteristics of traffic states, which typically repeat every 24 hours. In total, we use 10×8 daily average statistics from $\{D-7\} \bigcup [D-3, D-1] \bigcup [D+1, D+3] \bigcup \{D+7\}$, where D is the predicted day. Since the training set and test set are spitted based on time, we cannot obtain full observation of data in one day during testing. To relieve the gap between training and testing, we randomly sampled 1-4 periods (12-48 time steps) in one day and used it to estimate the average daily statistics.

Type	Berlin	Istanbul	Moscow
Without Periodic Features	1.3040e-3	0.9173e-3	1.3807e-3
With Periodic Features	**1.2980e-3**	**0.9165e-3**	**1.3764e-3**

Table 3. Validity of periodic features based on the MSE on the validation set of each city using the same inputs.

3.2.2 Time, Weekday and Holiday Features

Time, weekday, and holiday features are definitely useful in traffic flow prediction tasks. We used a two-dimensional vector to represent time in one day by projecting $[0, 287]$ to a unit circle. We used a one-hot vector to represent weekday and a Boolean value to represent the holiday. Holiday information was obtained from www.officeholidays.com. Table 4 shows strong validity of holiday features, which is in line with our expectations.

Type	Berlin	Istanbul	Moscow
Without Holiday Features	1.2980e-3	0.9165e-3	1.3764e-3
With Holiday Features	**1.2937e-3**	**0.9100e-3**	**1.3705e-3**

Table 4. Validity of holiday features based on the MSE on the validation set of each city using same inputs. Note that features and dataset used here might not be the same as other tables in this paper.

3.2.3 Geo-Embedding features

The locality is a common assumption in image segmentation and classification task that the object should be identical irrespective of its position in the image. This assumption is not valid for spatiotemporal data, as each pixel in the spatiotemporal data represents a region in the physical world, which has its inherent attributes. To adapt semantic segmentation models to spatiotemporal data, it is necessary to develop a technique to learn the inherent attributes of each pixel. In our previous studies[14], we used embedding technique to generate regional 'personalized' temporal

information and feed it into the convolutional neural network. In this competition, we further propose the method of geo-embedding to learn the inherent attributes of each location (*i.e.*,pixel). We concatenate a learnable tensor $C \times N \times M$ to each input and optimize it using the model. For the implementation, we use the NN.EMBEDDING in Torch by assigning a different ID to each pixel since NN.EMBEDDING has options of norms to parameters. Table 5 lists the effects of geo-embedding features, and the contribution of geo-embedding features can be observed.

Type	Berlin	Istanbul	Moscow
Without Embedding Features	1.2919e-3	-	-
With Embedding Features	**1.2913e-3**	-	-

Table 5. Validity of embedding features based on the MSE on the validation set of each city using the same inputs.

3.3 Training

Due to device issues, most of our models were trained on a mini-batch size of 3×4 with $3\times$ 2080Ti GPU. We typically trained each model with 15 epochs with an initial learning rate of 0.01 and a linear learning rate decay. It typically took 16-20 hours to train a model. We also included SYNCBATCHNORM to stabilize and speed up the training. In this section, we will introduce some options during training that potentially contribute to model performance.

3.3.1 Choice of Optimizer

SGD and ADAM[15] are commonly used to optimize the model. In most conditions, SGD is slower but theoretically guarantees to convert, while ADAM is slightly faster, but may not guarantee to convert. Recently, other self-adaptive optimizers have also become popular, e.g., LAMB[16]. It should be noted that, we did not optimize the learning rate for SGD; hence it might be possible, but less likely, that after a careful design of the initial learning rate for SGD, the result will be comparable to the self-adaptive optimizer in the same training time. We compare several optimizers in Table 6, and LAMB appears to be the best optimizer for this task.

Type	Berlin	Istanbul	Moscow
SGD	1.8271e-3	-	-
ADAM	1.3067e-3	-	-
ADAMW	1.3042e-3	-	-
LAMB	**1.3040e-3**	-	-

Table 6. Comparison of optimizers based on the MSE score on the validation set of each city using the same inputs. The number of epochs is set to 15 and initial learning rate is set to 0.01.

3.3.2 Warm-up

Learning rate warm-up has been used for many NLP tasks. Table 7 lists results of our examination of the effects of warm-up learning rate. By fixing the training epochs and learning rate, using warm-up can generate far better results.

Type	Berlin	Istanbul	Moscow
Without warm-up	1.6280e-3	-	-
With warm-up	**1.3067e-3**	-	-

Table 7. Validity of warm up based on the MSE on the validation set of each city using the same inputs. Note that features and dataset used here might not be the same asin other tables in this paper.

3.3.3 Inclusion of Validation Set into Training Process

After two weeks of experiments and submissions, we found that the offline validation set MSE score and the online MSE score were consistent, even though we greedily selected the best model parameter based on validation MSE. This phenomenon gave us hint to include the validation data in the training process. In the final solution, half of the models are trained by both the training set and the validation set.

4 Conclusion

In this paper, we conducted systematic research on large-scale spatiotemporal traffic prediction. The model structure is designed to accommodate spatiotemporal prediction based on the image segmentation model. Targeting spatiotemporal properties of traffic data, we proposed several new methods, including geo-embedding, and explored the details and tricks in model training. It should be noted that one assumption of the current dynamic traffic state prediction model is that no significant external influences exist.However, in reality, such as when large events are held, the model's predictive performance is drastically reduced. For future research, we will investigate effective traffic prediction under strong external influences.

About Us

Fanyou Wu is pursuing a Ph.D. degree in Forestry and Natural Resources Department, Purdue University, West Lafayette, USA. He has gained a wealth of experience in the theory of interdisciplinary applications of machine learning techniques and has won many championships in AI competitions organized by leading international AI conferences or research institutes, including the championship of JDD (2019), championship of IJCAI-Adversarial AI Challenge (2019), and championship of KDD Cup (2020).

Yang Liu is pursuing a Ph.D. degree in Transportation Engineering in the School of Transportation at Southeast University, Nanjing, China. He has rich experience in artificial intelligence applications, covering diverse fields from spectral classification for LAMOST to local climate zone classification for Sentinel-1, and from short video recommendation for TikTok users to travel mode recommendation for travelers. He has won three championships of Alibaba's Tianchi Algorithm Competition, championship of IJCAI-Adversarial AI Challenge (2019), and championship of KDD Cup (2020).

References

1. Ke, J., Zheng, H., Yang, H. & Chen, X. M. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transp. Res. Part C: Emerg. Technol.* **85**, 591–608 (2017).
2. Yao, H. *et al.* Deep multi-view spatial-temporal network for taxi demand prediction. In *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).
3. Liu, Y., Liu, Z. & Jia, R. Deeppf: A deep learning based architecture for metro passenger flow prediction. *Transp. Res. Part C: Emerg. Technol.* **101**, 18–34 (2019).
4. Fu, R., Zhang, Z. & Li, L. Using lstm and gru neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 324–328 (IEEE, 2016).
5. Zhang, J., Zheng, Y. & Qi, D. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *Thirty-First AAAI Conference on Artificial Intelligence* (2017).
6. Geng, X. *et al.* Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting. In *2019 AAAI Conference on Artificial Intelligence (AAAI'19)* (2019).
7. Li, Y., Yu, R., Shahabi, C. & Liu, Y. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations* (2018).
8. Pan, Z. *et al.* Urban traffic prediction from spatio-temporal data using deep meta learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD'19)*, 1720–1730 (ACM, New York, NY, USA, 2019).
9. Yu, B., Yin, H. & Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)* (2018).
10. Koesdwiady, A., Soua, R. & Karray, F. Improving traffic flow prediction with weather information in connected cars: A deep learning approach. *IEEE Transactions on Veh. Technol.* **65**, 9508–9517 (2016).
11. Liao, B. *et al.* Deep sequence learning with auxiliary information for traffic prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 537–546 (ACM, 2018).
12. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241 (Springer, 2015).
13. Wang, J. *et al.* Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis Mach. Intell.* 1–1, DOI: 10.1109/TPAMI.2020.2983686 (2020).
14. Liu, Z., Liu, Y., Lyu, C. & Ye, J. Building personalized transportation model for online taxi-hailing demand prediction. *IEEE Transactions on Cybern.* (2020).
15. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. In Bengio, Y. & LeCun, Y. (eds.) *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (2015).
16. You, Y. *et al.* Large batch optimization for deep learning: Training bert in 76 minutes. In *International Conference on Learning Representations* (2020).