Two-level resolution neural network for freeway traffic prediction

Semin Kwak[1], Danya Li[1], and Nikolas Geroliminis[1]

[1]Urban Transport Systems Laboratory (LUTS), EPFL, Switzerland

SHORT SUMMARY

As deep learning has achieved great success in different fields, introducing proper prior domain knowledge is proven to be an efficient way of improving neural networks' performance, which can also be applied in the transportation domain. Traffic forecasting is one of the most challenging problems heavily tackled by deep learning because it requires learning highly complex Spatio-temporal correlations of traffic states. In this paper, we suggest a two-level resolution deep neural network architecture, arguing that existing methods extract the correlations in a *flat* manner. The two-level resolution blocks promote decomposition of the Spatio-temporal correlations into low and high-resolution ones, which explain general changes and details of traffic speed. Therefore, each block learns a representative regional traffic fluctuating behavior and detailed local changes separately. We successfully apply this two-level resolution structure to two existing neural network models and obtain improved performance.

Keywords: Traffic speed prediction, Freeway sensor network, Two-level resolution block, Deep neural network

1. INTRODUCTION

Traffic prediction of freeway sensor networks has been attempted by researchers in various fields, as it does not require special transportation knowledge and improved data accessibility (Ma et al., 2017; Yu, Wu, Wang, Wang, & Ma, 2017; Zhang, Wang, Chen, Cao, & Huang, 2019; He, Chow, & Zhang, 2019; Li, Yu, Shahabi, & Liu, 2018). The state-of-the-art technologies have proven that embedding proper *inductive bias* into neural networks is critical; (Battaglia et al., 2018) explains inductive bias as "Inductive bias allows a learning algorithm to prioritize one solution (or interpretation) over another, independent of the observed data."

Assuming traffic states are defined in a graph is a critical inductive bias of these state-of-the-art technologies. This assumption makes it possible to extract spatial correlations efficiently, ignoring correlations from remote sensors appropriately, and consequently, learn only the necessary correlations. For example, in a section where congestion mainly occurs, other upstream sensors are expected to see this congestion by the shock wave theory (Daganzo, 1997). This phenomenon can be easily captured with a well-defined graph structure.

Traffic states also have topologically independent components since these are a consequence of human behavior. For example, the average traffic value of a network shows strong periodic characteristics since people commute in the morning and evening. Although the average traffic seems a fairly simple variable, existing methods with the graph convolutional operation have a limitation to extract it since a high order operation is required, which is usually not desirable.

This paper claims that inducing predictors to decompose traffic states into two parts (topologically independent and dependent) improves their performance. Therefore, we suggest a two-level resolution neural network. One level predicts the average value of traffic-related to the network level, independent of the corresponding topology structure. The other predicts traffic signals on the sensor level induced by the network topology. We improve the performance of two existing models, reforming them to this two-level resolution structure.

2. METHODOLOGY

We represent the sensor network as a weighted directed graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathbf{W})$, where $\mathscr{V}$ is a set of N sensors $|\mathscr{V}| = N$, $\mathscr{E}$ is a set of edges, and $\mathbf{W} \in \mathbb{R}^{N \times N}$ is an adjacency matrix representing sensor proximity. $\mathbf{X}^{(t)} \in \mathbb{R}^{N \times F}$ is the F traffic states features (e.g. speed, time of day, day of week, etc) observed on $\mathscr{G}$ at time t, $\hat{\mathbf{X}}^{(t)} \in \mathbb{R}^{N \times 1}$ is the predicted feature for all sensors at time t. The traffic prediction problem aims to learn a function f that maps s historical traffic features to future h horizons of a target feature, given a graph $\mathscr{G}$:

$$[\mathbf{X}^{(t-s+1)}, .., \mathbf{X}^{(t)}; \mathscr{G}] \xrightarrow{f} [\hat{\mathbf{X}}^{(t+1)}, .., \hat{\mathbf{X}}^{(t+h)}]. \tag{1}$$

High-resolution block

The state-of-the-art methods typically design the function f to extract spatial correlations between sensors by using the given graph $\mathscr{G}$ and temporal correlations by the Encoder-Decoder Recurrent Neural Network (RNN) architectures (Li et al., 2018; Shang, Chen, & Bi, 2021) as follows:

$$[\mathbf{X}^{(t-s+1)}, ..., \mathbf{X}^{(t)}; \mathscr{G}] \xrightarrow{f_g} [\mathbf{H}^{(t-s+1)}, ..., \mathbf{H}^{(t)}] \xrightarrow{f_e} \mathbf{C}^{(t)} \xrightarrow{f_d} [\hat{\mathbf{X}}^{(t+1)}, ..., \hat{\mathbf{X}}^{(t+h)}]. \tag{2}$$

The function f_g converts each input $\mathbf{X}$ into $\mathbf{H}$ that absorbs topological information by $\mathscr{G}$. Then, all tensors $[\mathbf{H}^{(t-s+1)}, ..., \mathbf{H}^{(t)}]$ are encoded by the encoder f_e resulting in a context vector $\mathbf{C}^{(t)}$. At last, the decoder f_d decodes this context vector generating the final prediction outputs. In this paper, we call this standard architecture the high-resolution block since all the operations are performed in a sensor level.

Low-resolution block

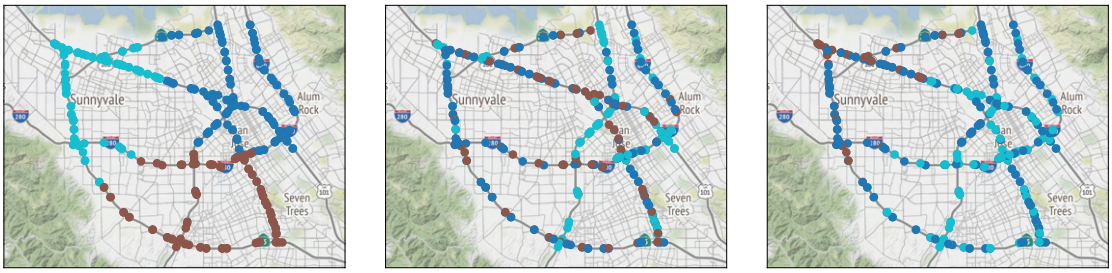
We suggest a low-resolution block that models the global traffic speed changes by predicting the average traffic status of the network. In this paper, we select the average traffic speed as the main variable of this block, but it is not limited to a single variable. The low-resolution block consists of three different functions: aggregator (g_a), encoder (g_e), and decoder (g_d):

$$[\mathbf{X}^{(t-s_l+1)}, ..., \mathbf{X}^{(t)}] \xrightarrow{g_a} [\bar{\mathbf{X}}^{(t-s_l+1)}, ..., \bar{\mathbf{X}}^{(t)}] \xrightarrow{g_e} \bar{\mathbf{C}}^{(t)} \xrightarrow{g_d} [\hat{\bar{\mathbf{X}}}^{(t+1)}, ..., \hat{\bar{\mathbf{X}}}^{(t+h)}], \tag{3}$$

where $\bar{\mathbf{X}} \in \mathbb{R}^F$ and $\hat{\bar{\mathbf{X}}} \in \mathbb{R}$ are the aggregated traffic features and predicted regional average speed of the network, respectively. The aggregator g_a aggregates each input $\mathbf{X}$ into the averaged value over N sensors ($\bar{\mathbf{X}}$). The encoder g_e and decoder g_d are learnable functions analogous to f_e and f_e respectively in Eq. (2). Note that the the input sequence length of the low-resolution block s_l can be different to s in Eq. (2).

Clustering

In the low-resolution block, the aggregator can produce multiple regional average speed of the network. A challenge is how to group sensors to produce them. Spectral clustering (Shi & Malik,



(a) Proximity based similarity. (b) Correlation based similarity. (c) Similarity with a mixture of proximity and correlation ($\alpha = 0.5$).

Figure 1: Spectral clustering with mixtures of physical proximity and signal correlation.

2000) is a K-means-based clustering algorithm taking the graph structure into account as a similarity matrix. Figure 1(a) shows a clustering result with the Spectral clustering method based on a proximity-based similarity matrix ($K = 3$). A potential problem of averaging traffic features under the clustering is that sensors which show very different traffic patterns can be grouped as the same cluster. For example, sensors on the same freeway but opposite direction usually show completely different pattern since people use one freeway during the morning but do the other during the evening for commuting. Using a signal correlation based similarity (e.g., cross-correlation, cosine similarity, etc) can alleviate this issue. With the similarity, as shown in Fig. 1(b), sensors located in opposite directions are well separated. We further mix proximity based and correlation based similarities by a convex combination with a parameter α:

$$\mathbf{W}_{\text{mix}} = (1-\alpha)\mathbf{W}_{\text{prox}} + \alpha\mathbf{W}_{\text{sim}}. \tag{4}$$

Figure 1(c) shows the clustering result with the mixed similarity with $\alpha = 0.5$. Sensors in the opposite direction are grouped in different clusters and each cluster also keeps topological locality at the same time. Therefore, the low-resolution block is capable to take representative values of each cluster and predict them. After that, these predicted value can be fed into the decoder of the high-resolution block.

Combination of two resolutions

We combine the two blocks by concatenating the prediction result of the low resolution block into the context vector of the high resolution block:

$$[\mathbf{X}^{(t-s+1)}, \ldots, \mathbf{X}^{(t)}; \mathcal{G}] \xrightarrow{f_g \circ f_e} [\mathbf{C}^{(t)}, \hat{\hat{\mathbf{X}}}^{(t+1)}, \ldots, \hat{\hat{\mathbf{X}}}^{(t+h)}] \xrightarrow{f_d} [\hat{\mathbf{X}}^{(t+1)}, \ldots, \hat{\mathbf{X}}^{(t+h)}] \tag{5}$$

Therefore, the encoder of the high-resolution block is trained to focus on residuals while low-resolution block takes responsibility for the general changes. Figure 2 shows the architecture details.

Loss

We predict one horizon each time and use the current prediction for the next prediction. We choose the Mean Absolute Error (MAE) overall horizons as our training metric. During training, we add the loss of low-resolution block to the total loss of the network by a weight γ, since such an extra loss can provide additional regularization for the model. When we evaluate the model, we discard

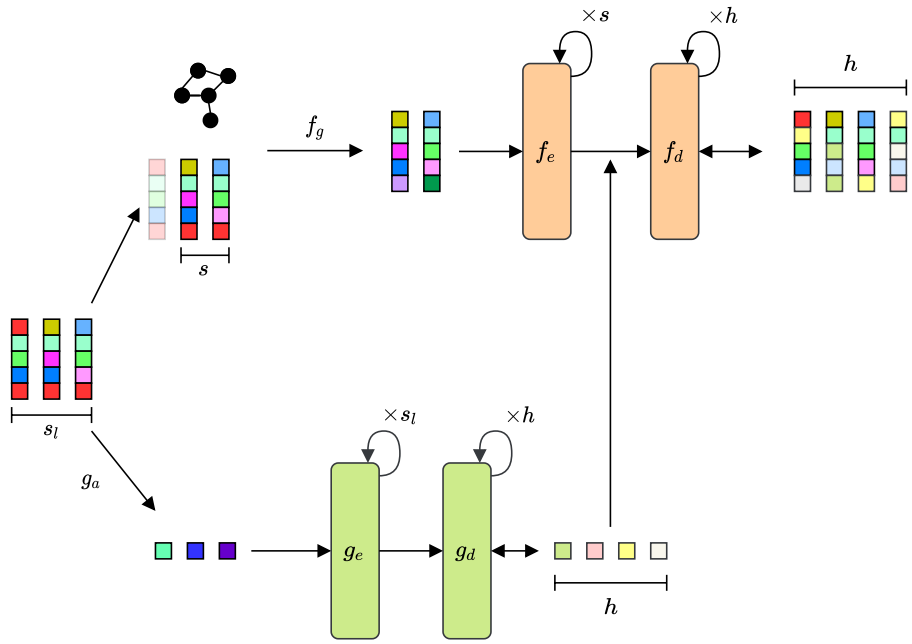


Figure 2: The architecture of a two-level resolution neural network. The upper model represent the high-level resolution and it corresponds to the sensor's speed prediction, whereas the bottom model stand for the low-level and it corresponds to the average speed prediction.

the low-resolution loss.

$$\ell_{\text{training}} = \frac{1}{hN} \sum_{\tau=t+1}^{t+h} |\hat{\mathbf{X}}^{(\tau)} - \mathbf{X}^{(\tau)}| + \frac{\gamma}{h} \sum_{\tau=t+1}^{t+h} |\hat{\bar{\mathbf{X}}}^{(\tau)} - \bar{\mathbf{X}}^{(\tau)}|, \tag{6}$$

$$\ell_{\text{inference}} = \frac{1}{hN} \sum_{\tau=t+1}^{t+h} |\hat{\mathbf{X}}^{(\tau)} - \mathbf{X}^{(\tau)}|. \tag{7}$$

3. RESULTS AND DISCUSSION

To make comparison with other state-of-the-art models, we conduct experiments on a large-scale transportation network: PEMS-BAY. This dataset consists of speed data collected from 325 sensors on the freeways of San Francisco Bay area, from January 1st to June 30th, 2017, aggregated every 5 minutes. The adjacency matrix of the nodes $\mathbf{W}$ for the high-resolution block is constructed by road network distance with a threshold Gaussian kernel. For the low-resolution block, we group all the sensor into a single cluster meaning a single average value is fed into it. Z-score normalization is applied to all the input features. The dataset is split in chronological order with 70% for training, 10% for validation and the remaining 20% for testing.

Two common metrics are used to evaluate the performance: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Table 1 summarizes the performance of the baseline models and our two-level resolution network. First, we select two classical time-series prediction methods. Historical average (HA) predicts the future traffic state of the time t as the average value of the time of five past weeks considering the day of the week. Auto-regressive integrated moving average (ARIMA) is a well-established parametric model to predict univariate time series.

Diffusion convolutional recurrent neural networks (DCRNN) (Li et al., 2018) and Graph for time-series (GTS) (Shang et al., 2021) are the two state-of-the-art methods that we apply our two-level resolution architecture. We add TLR as a prefix on the two baselines, which stands for the two-

level resolution.

Table 2 lists the performance of different time periods on the PEMS-BAY dataset to give a more specific analysis, especially to observe the improvements obtained for peak hours.

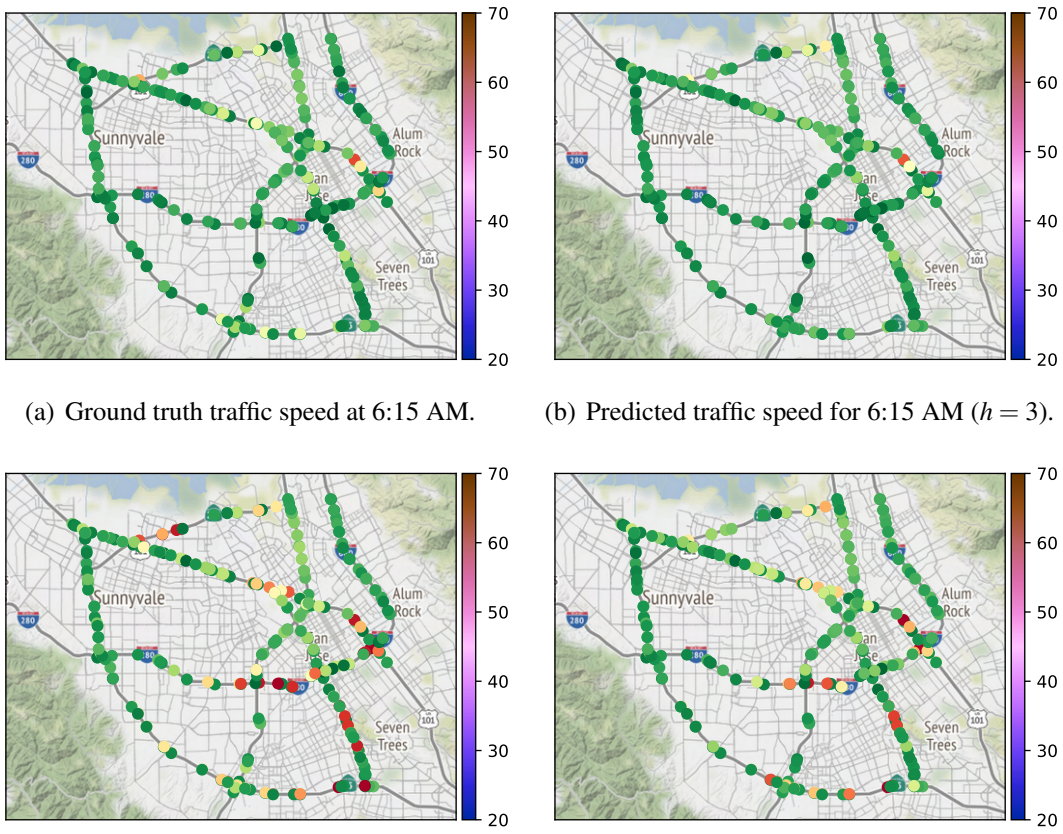
Table 1: Experiment results on PEMS-BAY dataset.

time	metric	HA	ARIMA	DCRNN	TLR-DCRNN	GTS	TLR-GTS
15 mins	MAE	2.88	1.62	1.38	**1.32**	1.34	**1.33**
	RMSE	5.59	3.30	2.95	**2.82**	2.91	**2.90**
30 mins	MAE	2.88	2.33	1.74	**1.63**	**1.66**	**1.66**
	RMSE	5.59	4.76	3.97	**3.76**	**3.87**	3.89
60 mins	MAE	2.88	3.38	2.07	**1.92**	1.96	**1.94**
	RMSE	5.59	6.50	4.74	**4.51**	**4.55**	4.57
90 mins	MAE	-	-	2.26	**2.19**	2.23	**2.17**
	RMSE	-	-	5.26	**5.05**	5.06	**4.99**
120 mins	MAE	-	-	2.66	**2.62**	2.58	**2.51**
	RMSE	-	-	6.02	**5.93**	5.71	**5.54**

Table 2: Experiment results of different time periods on the PEMS-BAY dataset. Morning peak hours are from 6 am to 10 am, evening peak hours are from 2 pm to 8 pm, the rests are off-peak hours.

time	metric	DCRNN			TLR-DCRNN		
		morning peak	evening peak	off peak	morning peak	evening peak	off peak
15 mins	MAE	1.85	1.74	1.00	**1.81**	**1.72**	**0.99**
	RMSE	3.69	3.54	2.17	**3.61**	**3.48**	**2.16**
30 mins	MAE	2.49	2.30	1.15	**2.41**	**2.24**	**1.14**
	RMSE	5.30	4.98	2.69	**5.09**	**4.78**	**2.61**
60 mins	MAE	3.08	2.73	1.34	**2.98**	**2.61**	**1.31**
	RMSE	6.57	5.99	3.21	**6.29**	**5.68**	**3.07**
90 mins	MAE	3.51	2.96	1.59	**3.38**	**2.84**	**1.55**
	RMSE	7.28	6.38	3.81	**6.97**	**6.06**	**3.73**

We focus on the results from adding the low-resolution block into the standard architecture (high-resolution block) and observe the following phenomena on the PEMS-BAY dataset. (1) Both two-level resolution (TLR) DCRNN and TLR-GTS outperforms two original works (Li et al., 2018; Shang et al., 2021) for most of horizons. This emphasizes the effectiveness of modeling the general traffic status plus a local traffic diffusion. (2) Especially, prediction in the long-term is greatly improved for both models, e.g., more than 1 hour. This demonstrates the ability of the low resolution block to really capture the long-term patterns. (3) Improvements mainly focus on the peak hours (both morning and evening peak) whereas the off-peak hours' improvements are less significant. Figure 3 shows a sample prediction result of the TLR-DCRNN method for a peak period of $h = 3$ (15 minutes after the current time) and $h = 12$ (one hour after the current time). We can see the algorithm predicts the beginning of congestion even though the input does not contain enough sign for the congestion. This result especially demonstrates the ability to model the exogenous traffic status changes of the low-resolution block.



(a) Ground truth traffic speed at 6:15 AM.

(b) Predicted traffic speed for 6:15 AM ($h = 3$).

(c) Ground truth traffic speed at 7:00 AM.

(d) Predicted traffic speed for 7:00 AM ($h = 12$).

Figure 3: Traffic speeds on 20th of June, 2017 and predicted speeds by TLR-DCRNN. The time when prediction is calculated is t =6:00 AM. The unit of speed is mph.

4. CONCLUSIONS

In this work, we propose a two-level resolution model that is decomposed into a low-resolution and high-resolution block that predicts the regional traffic status and the local traffic propagation, respectively. We introduce a graph-based clustering method to get regional representative values which are fed into the low-resolution block. We apply this two-level resolution architecture in two state-of-the-art models and obtain considerable improvements, especially on desirable peak periods.

REFERENCES

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., ... others (2018). Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*.

Daganzo, C. (1997). *Fundamentals of transportation and traffic operations* (Vol. 30). Pergamon Oxford.

He, Z., Chow, C.-Y., & Zhang, J.-D. (2019). Stcnn: A spatio-temporal convolutional neural network for long-term traffic prediction. In *2019 20th ieee international conference on mobile data management (mdm)* (pp. 226–233).

Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International conference on learning*

representations (iclr '18).

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, *17*(4), 818.

Shang, C., Chen, J., & Bi, J. (2021). Discrete graph structure learning for forecasting multiple time series. *arXiv preprint arXiv:2101.06861*.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(8), 888–905.

Yu, H., Wu, Z., Wang, S., Wang, Y., & Ma, X. (2017). Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors*, *17*(7), 1501.

Zhang, Y., Wang, S., Chen, B., Cao, J., & Huang, Z. (2019). Trafficgan: Network-scale deep traffic prediction with generative adversarial nets. *IEEE Transactions on Intelligent Transportation Systems*.