Traffic Speed Prediction and Congestion Source Exploration: A Deep Learning Method

Jingyuan Wang, Qiang Gu, Junjie Wu, Guannan Liu, Zhang Xiaoing

Beihang University, Beijing, China
Targets of our work

• Traffic speed prediction
  — Problem: Predicting future traffic speed of a road segment using history speed data
  — Users: car drivers

• Congestion source exploration
  — Problem: Discovering segments that may cause traffic congestions
  — Users: urban planners
Related works

• Traffic speed prediction
  — Before 2000: ARIMA and its variations
    • ARIMA (1979), Kohonen-ARIMA (1996), ARIMAX (1999), etc.
  — After 2000: ANN and SVR
  — After 2014: Deep Learning
    • RNN-RBM (PlosOne2015), SAE (T-ITS2014, T-ITS2015)

• Shortcomings
  — Consider traffic prediction as a regular time series prediction problem.
  — Fail to model some unique features of the traffic scenarios.
Unique features

• Spatio-temporal correlations
  — Congestion may be caused by a segment in the downstream of a road.
  — Traffic congestion on a road always last a very long time.

• Spatio-temporal locality

No-congestion  Congestion
Unique features

- Unpredictable events
• Error Feedback Recurrent CNN (eRCNN)

1) The Spatio-Temporal Input Matrix
Function: Modeling ST relationship
The Spatio-Temporal Input Matrix

Goal: Predicting traffic speed of a road segment at the time $t+1$

Speed of up/down stream segments at the time $t$

at the time $t-1$

at the time $t-n$
Error Feedback Recurrent CNN

• The Spatio-Temporal Input Matrix

```
\textbf{v_{s-m,:}} \quad \ldots \quad \textbf{v_{s,:}} \quad \ldots \quad \textbf{v_{s+m,:}}
```

Traffic direction

Spatial trends

Temporal trends
Framework

• Error Feedback Recurrent CNN

2) The CNN-based Feature Extracting Function: Extracting ST correlation and locality Feature
Error Feedback Recurrent CNN

• The CNN-based Feature Extracting

Insight: There exists a number of local patterns (spatio-temporal trends) in the spatio-temporal input matrix.

Idea: We adopt a Convolutional Neural Network based structure to extract spatio-temporal trend features from the spatio-temporal input matrix.
The CNN-based Feature Extracting

**Convolution layer:**
\[ c_{k}^{p,q} = \sigma \left( b_{k} + \sum_{x=0}^{i} \sum_{y=0}^{i} w_{k}^{x,y} m_{p+x,q+y}^{x+y} \right) \]

Using several filters to convolute the input matrix.

**Pooling layer:**
\[ p_{c} = \frac{1}{N} \sum_{p} \sum_{q} c_{k}^{p,q} \]

Using average pooling to down sampling the convolution neuron matrix.
Framework

- Error Feedback Recurrent CNN

3) The Error-Feedback Recurrent Layer
   Function: Handle effect of sudden events
The Error-Feedback Recurrent Recurrent Layer

A speed drop off caused by an unpredictable event.

The prediction model input does not contain enough information about the event.
The Error-Feedback Recurrent Layer

Idea: feed the prediction error back to the network
• Error Feedback Recurrent CNN

4) The Regression Output Layer
Function: speed regression
Error Feedback Recurrent CNN

• The Output Layer

Modified ReLU

\[ \sigma(x) = \begin{cases} 
0 & \text{if } x \leq 0 \\
 x & \text{if } 0 < x < 1 \\
1 & \text{if } x \geq 1 
\end{cases} \]

Speed Normalization

\[ \psi(x) = \begin{cases} 
1 & \text{if } x \geq 80 \text{ km/h} \\
1 - \frac{80-x}{70} & \text{if } x \in [10, 80] \text{ km/h} \\
0 & \text{if } x \leq 10 \text{ km/h} 
\end{cases} \]
Network Training

- Parameters Training: **Back Propagation**

\[ L = \frac{1}{2} \sum_k (y_k - o_k)^2 \]

\[
\frac{\partial L}{\partial w^{(E)}_k} = \frac{1}{m} \sum_m d^{(E)}_k(t) e(t - 1), \\
\frac{\partial L}{\partial w^{(O)}_k} = \frac{1}{m} \sum_m d^{(O)}_k(t) [r^{(R)}; r^{(E)}], \\
\frac{\partial L}{\partial w^{(R)}_k} = \frac{1}{m} \sum_m d^{(R)}_k(t) p, \\
\frac{\partial L}{\partial b^{(O)}_k} = \frac{1}{m} \sum_m d^{(O)}_k(t),
\]
Network Training

- Pre-Training and Fine-Tuning eRCNN

Segments in the same cluster

The segment to be predicted

Pre-training

Fine-Tuning

The finial model
Experiments: Data Description

• The 2nd ring road and the 3rd ring road
  — About 10% of the total traffic flow in Beijing downtown area.
  — The average length of each road segment is **400 meters**.
  — The traffic speed of a segment is updated every **5 minutes**.

• The data set was collected from the **25 weekdays** in Nov. 2013.
  — The data of the first **20 weekdays** were used as the training set.
  — The remaining **five days** is the test set.
Experiments

• Benchmarks
  — Auto Regression Integrated Moving Average (ARIMA) [1]
  — Support Vector Regression (SVR) [11]
  — Stacked Auto Encoders (SAE) [5]
  — 1D Convolutional Neural Network (1D-CNN)
    • the same as the CNN part of eRCNN, but the input matrix reduces to the time series of the traffic speeds of the segment to be predicted.
    • benchmark to test the spatio-temporal input matrix.
  — Convolutional Neural Network (CNN)
    • the same as eRCNN, except the error feedback procedure is removed.
    • benchmark to test the performance of the error feedback scheme.
Experiments

- Overall performance: scenario I

The 2\textsuperscript{nd} ring road

The 3\textsuperscript{rd} ring road
Experiments

- Overall performance: scenario II

The 2nd ring road

The 3rd ring road
Experiments

• Performance with Time Variation

**Prediction delay**
- 19:00 to 19:30, the traffic recovers from the last traffic jam of the night peak
- around 20:20, the traffic speed decreases again due to a small accident

**Performance**
- eRCNN captures the abrupt changes in speeds
- the prediction curve exactly matches the real values
- CNN model does not follow the abrupt changes of traffic speeds
Targets of our work

• Traffic speed prediction
  — Problem: Predicting future traffic speed of a road segment using history speed data
  — Applications: navigation
  — Users: car drivers

• Congestion source exploration
  — Discovering segments that cause traffic congestions
  — Users: urban planners
Importance analysis for road segments

We use the traffic speed of the segment \( i-m \) to predict the speed of the segment \( i \).

We define the \textit{influence} of segment \( i \) to segment \( j \) as the derivative of \( v_j \) to \( v_i \), i.e.

\[
I_i(j) = \frac{df(v_i)}{dv_i} = \lim_{\varepsilon \to 0} \frac{f(v_i) - f(v_i - \varepsilon)}{\varepsilon}
\]

We approximately calculate the \textit{influence} of the segment \( s-m \) to \( s \) at time \( t \) as

\[
I_{s-m,t}(s) = \sum_{k=t}^{t-n} \left| \frac{\partial o_{s,t+1}}{\partial v_{s-m,k}} \right|
\]

We define the \textit{importance} of the segment \( k \) as its influence to all segments in the same road with it, i.e.

\[
\text{Importance}_k = \sum_{t} \sum_{s \neq k} I_{k,t}(s).
\]
Importance analysis for road segments

• The importance of segments in the 2nd and 3rd ring roads.
Conclusion

• In this paper, we proposed a novel deep learning method called eRCNN for traffic speed prediction of high accuracy.

• Experiments on real-world traffic speed data of the ring roads of Beijing city demonstrated the advantages of eRCNN to the excellent competitors.

• In particular, we illustrated how to explore the congestion sources from eRCNN.
THANK YOU!
Experiments

• Performance with Pre-Training
  – eRCNN is greatly enhanced by the pre-training scheme even facing the drastic speed changes during the morning and evening peaks.
Importance analysis for road segments

**Input Matrix**  
Convolution  
Pooling  
Error feedback  
Output

$$\frac{\partial c_{p,q}^k}{\partial V} = c_{p,q}^k (1 - c_{p,q}^k) W_k^{(C)}$$

$$\frac{\partial p_k}{\partial V} = \sum_i \sum_j w_{i,j,k}^{(R)} \frac{\partial p_{i,j}^k}{\partial V}$$

$$\frac{\partial o}{\partial V} = \delta(o) W_{(OR)}^{(R)} \frac{\partial r^{(R)}}{\partial V}$$

$$\frac{\partial p_{i,j}^k}{\partial V} = \frac{1}{4} \sum_{m=2i-1}^{2i} \sum_{n=2j-1}^{2j} \frac{\partial c_{m,n}^k}{\partial V}$$

$$\frac{\partial r^{(R)}}{\partial V} = r^{(R)} (1 - r^{(R)}) \sum_k \frac{\partial p_k}{\partial V}$$
Error Feedback Recurrent CNN

• The Error-Feedback Recurrent Layer

Prediction Delay

before 5 > after 1
before 4 > after 2
before 3 = after 3
before 2 < after 4