Spatio-Temporal-Graph-Convolutional-Networks-A-Deep-Learning-Framework-for-Traffic-Forecasting

reference

- https://www.zhihu.com/question/54504471
- https://en.wikipedia.org/wiki/Laplacian_matrix
- https://tkipf.github.io/graph-convolutional-networks/
- https://www.inference.vc/how-powerful-are-graph-convolutions-review-of-kipf-welling-2016-2/
- http://cs229.stanford.edu/section/cs229-moregaussians.pdf

abstract

- Spatio-Temporal Graph Convolutional Network
- tackle the time series prediction problem in traffic domain
- complete convolutional structures.

introduction

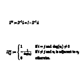
- linear regression perform well on short interval forecast instead of long terms
- this is a data-driven and using spotio-temporal information method.
- fully utilize spatio-information instead of treating it as discrete units
- $$\hat{v}_{t+1}, \ldots, \hat{v}_{t+H} = argmaxlog_{10} P(v_{t+1}, \ldots, v_{t+H} | v_{t-M}, \ldots, v_t)$$
-
- where

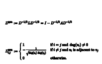
$$v_t \in R^n$$

, n is an observation vector of n road segments at time step t

Convolutions on Graphs

- normalized Laplacian
  - Random walk normalized Laplacian

    
  - analogy to The Multivariate Gaussian Distribution
  - Symmetric normalized Laplacian L:

    
- first generation of GNC

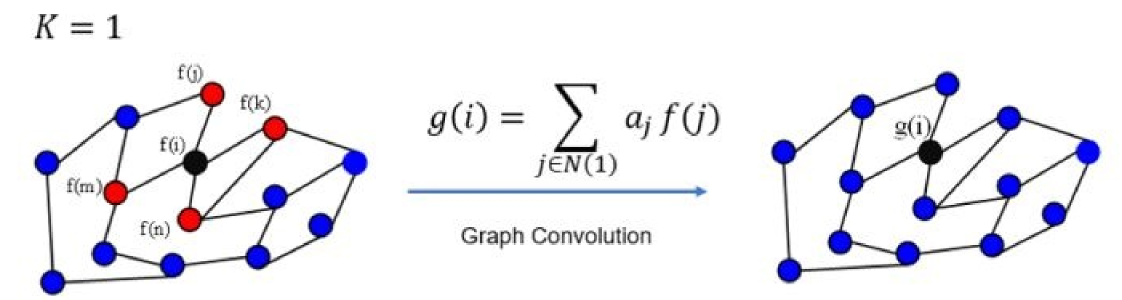
$$y_{output} = \sigma\left(U \begin{pmatrix} \theta_1 & & \\ & \ddots & \\ & & \theta_n \end{pmatrix} U^T x\right) \qquad (3)$$

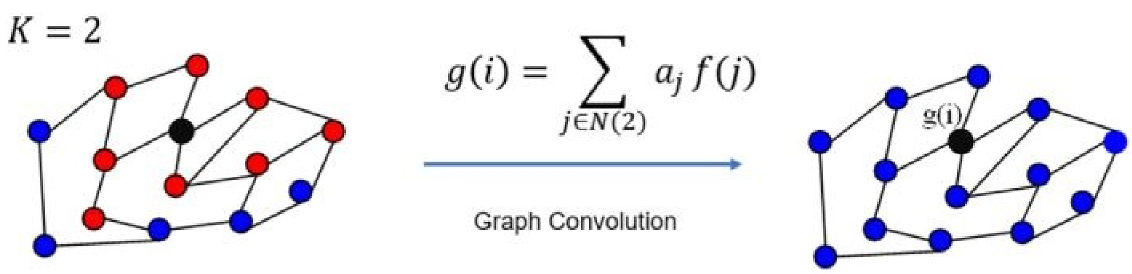
- second generation of GNC

$$y_{output} = \sigma\left(U \begin{pmatrix} \sum_{j=0}^{K} \alpha_j \lambda_1^j & & \\ & \ddots & \\ & & \sum_{j=0}^{K} \alpha_j \lambda_n^j \end{pmatrix} U^T x\right) \qquad (4)$$

$$U \sum_{j=0}^{K} \alpha_j \Lambda^j U^T = \sum_{j=0}^{K} \alpha_j U \Lambda^j U^T = \sum_{j=0}^{K} \alpha_j L^j$$

$$y_{output} = \sigma\left(\sum_{j=0}^{K} \alpha_j L^j x\right) \qquad (5)$$





  - if k == n, receptive field is n hop
- third generation of GNC
  - where c_1 , c_2 and c_3 are fixed

- The only trainable parameters are

$$\theta_0$$

and

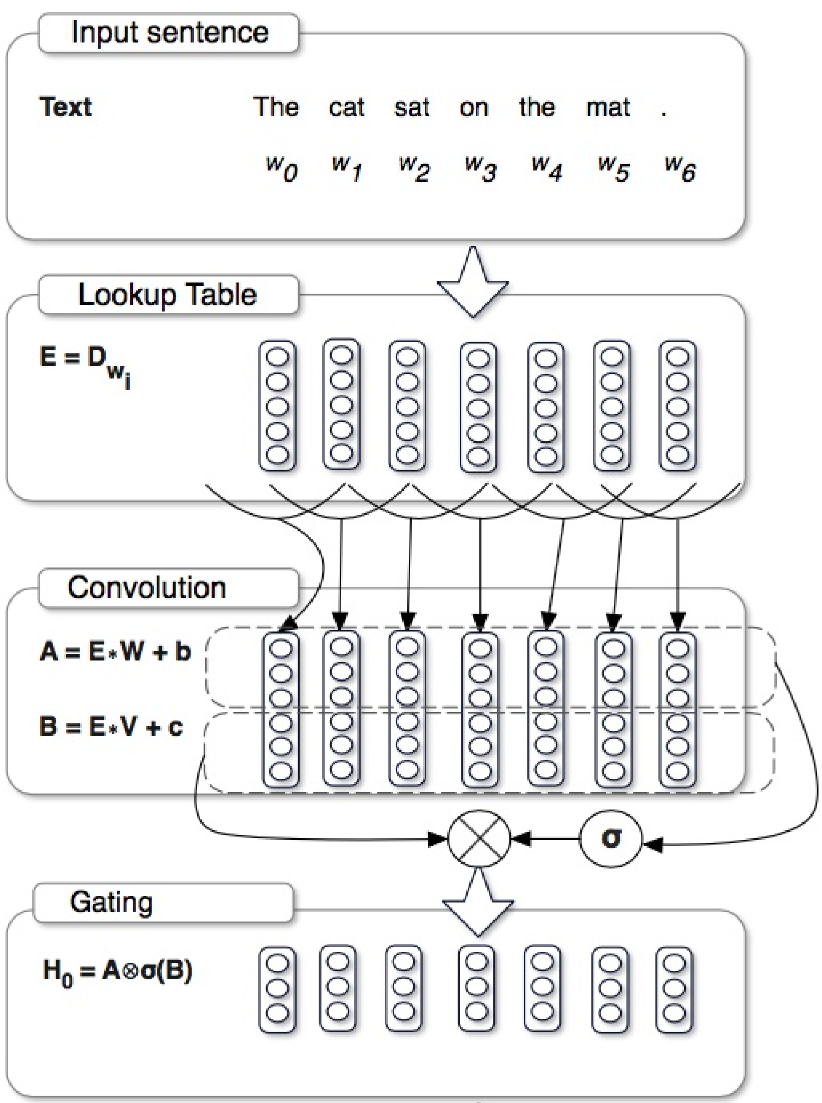
$$\theta_1$$

- in the final version the authors even further fix

$$\theta_0 = -\theta_1$$

Network Architecture

- main architecture
- GLU architecture



- main equation

- final equation

Experiments

- linear interpolation method for missing values
- normalized by standard score method((x-mean)/std)
- adjacency matrix

  10,0.5

result