ETH zürich

Traffic Map Forecasting using Graph Convolutional Neural Networks

Konstantin Arbogast
Master Thesis, FS 2020
Supervisor(s): Prof. Dr. Martin Raubal, Henry Martin, Dominik Bucher

IKG
Institut für Kartografie und Geoinformation

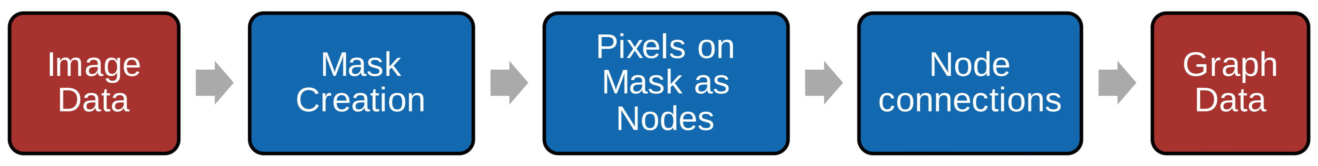
geoinformation engineering ikg eth zurich

MIE lab
Mobility Information Engineering Lab at ETH Zurich

1 Introduction

In 2019, the IARAI launched the *Traffic4cast* competition, where traffic data for 15 minutes was predicted, based on images of the previous hour. This was done using Convolutional Neural Networks (CNN). During this thesis, a Graph Convolutional Neural Network (GCN) was developed to shift the problem from image to graph representation, in order to work with the additional spatial information given by the street network. The goal of this thesis was to compare this GCN with a CNN developed for the competition [1] in terms of prediction error, generalization ability and efficiency.

2 Preprocessing



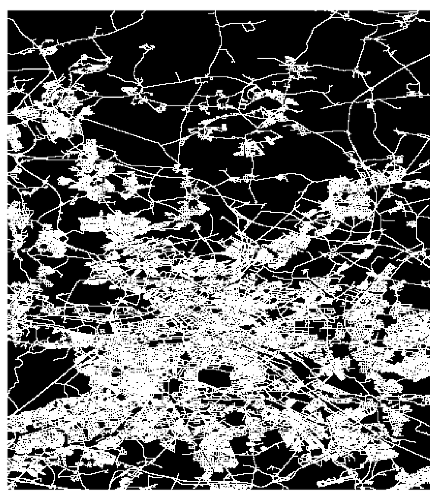


Fig. 1. Mask for Berlin. White pixels represent pixels on the road network.

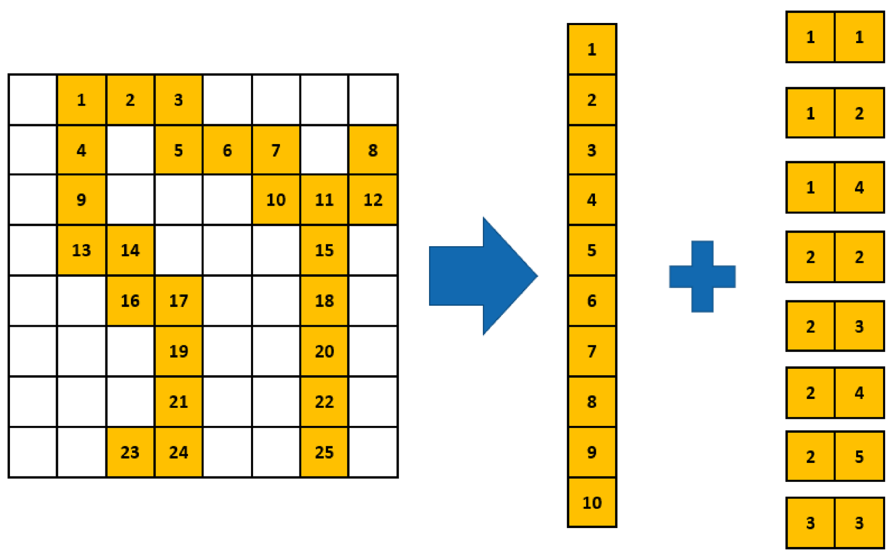


Fig. 2. Getting from image to graph data: Representing pixels as nodes, storing neighboring pixels as connections in a tensor.

3 Experimental Setup

- Chebyshev convolution [2] used as graph convolution technique.
- Convolution block consisting of graph convolution, batch normalization and ReLU activation.
- Max Pooling done with street based clustering approach: Clustering according to position in the image by overlaying grid. Additionally getting the street types from open street maps and splitting the clusters accordingly.
- KNN interpolation for unpooling.
- Dropout layer to prevent over-fitting.

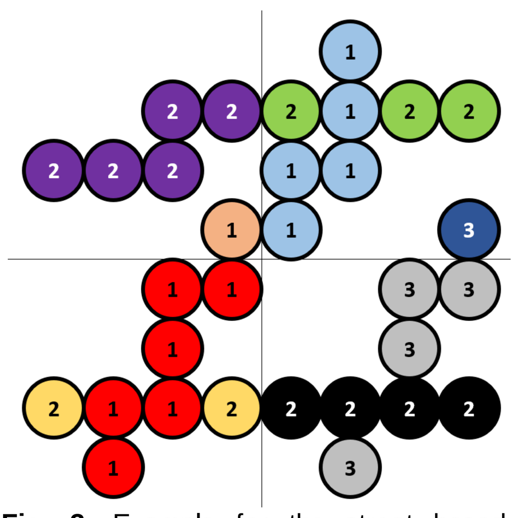


Fig. 3. Example for the street based clustering used for pooling. Numbers represent street types. Nodes are clustered by location and street type.

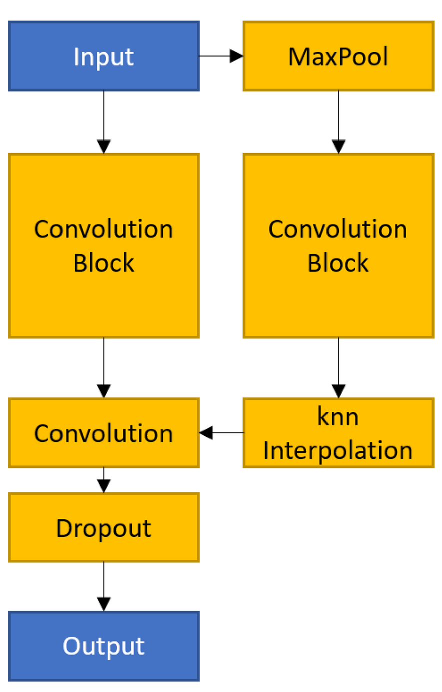


Fig. 4. Simplified representation of the model architecture – Full model has additional width and depth.

4 Results and discussion

GCN	Berlin	Moscow	Istanbul
Berlin	193.3	***205.8***	***203.0***
Moscow	***219.4***	187.6	***193.0***
Istanbul	278.5	***255.3***	236.7

CNN	Berlin	Moscow	Istanbul
Berlin	***182.2***	312.0	287.4
Moscow	283.6	***181.1***	310.9
Istanbul	***276.3***	301.4	***232.1***

Tab. 1. Mean squared errors of the GCN (left) and the CNN (right) calculated on the whole images. Model trained on the city at the top applied to the city on the left. Lower losses compared to the other model are highlighted.

	t/epoch	Memory usage
GCN	05:58:06	***4335 MiB***

	t/epoch	Memory usage
CNN	***03:28:38***	9273 MiB

Tab. 2. Efficiency evaluation for GCN (left) and CNN (right). Training time per epoch and GPU memory usage.

- Lower validation losses for CNN when the model is applied to the city it was trained on.
- Both models beat the baseline when applied to the same city.
- Lower validation losses for GCN when the model is applied to other cities.
- GCN model trained on Istanbul even beats the baseline when applied to other cities.
- GCN training is slower, but uses significantly less GPU memory.

Baseline	
Berlin	204.5
Moscow	195.9
Istanbul	240.2

Tab. 3. Baseline for MSE values for all three cities. Predictions done by calculating the average traffic within the last hour and applying it to the next 15 minutes.

5 Conclusion

- **Prediction Error:** The CNN outperforms the GCN when applied to the city the model was trained on.
- **Generalization ability:** The GCN is more able to generalize, as it outperforms the CNN significantly in most cases when applied to another city, sometimes even beating the baseline in other cities.
- **Efficiency:** There's a trade-off between the two models in terms of efficiency. While the CNN training needs less time per epoch, the GCN training uses significantly less GPU memory.

Generally, a GCN seems to be a promising method for traffic predictions. Especially when the goal is to train a single model for multiple cities.

6 References

1. Martin, H., Hong, Y., Bucher, D., Rupprecht, C., and Buffat, R. (2019). Traffic4cast – traffic map movie forecasting team mie-lab. *arXiv: 1910.13824v2 [cs.CV].*
2. Defferrard, M., Bresson, X., and Vandergheynst, P. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in Neural Information Processing Systems*, 29:3844–3852.