
Deep Networks on Graph-Structured Data

Anonymous Author(s)

Affiliation

Address

email

Abstract

1 Introduction

Generalizing convolutional networks to graph-structured data is an important open problem. Thus far ConvNets have proved extremely successful for a large variety of tasks in computer vision and acoustic modeling [3, 2]. This is largely due to their ability to efficiently exploit stationarity and local statistics to greatly reduce the number of parameters in the network without sacrificing the capacity to accurately represent the data. As a result, one is able to train very large networks while limiting the overfitting problem. The implicit assumption behind a ConvNet is that the data lives on a lattice, a specific type of graph. Images can be thought of as signals defined on a 2-D lattice, where each pixel is a node which is connected to its immediate neighbors. Similarly, audio waveforms can be viewed as signals defined on a 1-D lattice, where each node is a time point.

However, many types of data are defined on more complex graphs. For example, text documents represented as bags of words can be thought of as signals defined on a graph whose nodes are vocabulary terms and whose weights represent some similarity measure between terms, such as co-occurrence statistics. In medicine, a patient's gene expression data can be viewed as a signal defined on the graph imposed by the regulatory network. In fact, computer vision and audio, which are the main focus of research efforts in deep learning, only represent a special case of data defined on an extremely simple graph.

In this work we propose two different approaches to generalizing ConvNets to data defined arbitrary graphs. We explore these approaches in two areas of application for which it has not been possible to apply convolutional networks: text categorization and bioinformatics. Our results show that our method is capable of matching or outperforming large, fully-connected networks trained with dropout using fewer parameters.

2 Spectral Networks

Our work builds upon [1] which defines spectral networks. We recall the definition here. A spectral network generalizes a convolutional network through the Graph Fourier Transform, which is in turn defined through a generalization of the Laplacian operator on the grid to the graph Laplacian.

Definition 1. Let W be a similarity matrix representing a graph G , and let L be its graph Laplacian with eigenvectors U . Then a graph convolution of input signals x and y defined on G is defined by $f \star g = U^T (Uf \odot Ug)$, where \odot represents a pointwise product.

There are several ways of computing L . In this work, we choose $L = I - D^{-1/2}WD^{-1/2}$, where D is a diagonal matrix with entries $D_{ii} = \sum_j W_{ij}$. Note that in the case where W represents the lattice, from the definition of L we recover the discrete Laplacian operator Δ . Also note that the Laplacian commutes with the translation operator, which is diagonalized in the Fourier basis. It

follows that the eigenvectors of Δ are given by the Discrete Fourier Transform (DFT) matrix. We then recover a classical convolution through the Convolution Theorem.

We can then generalize a convolutional network by replacing the convolutions in the three main operations by graph convolutions defined above. Specifically, we have:

Algorithm 1 Train Graph Convolution Layer

- 1: Given GFT matrix U , interpolation kernel K , weights w .
 - 2: **Forward Pass:**
 - 3: Fetch input batch x and gradients w.r.t outputs ∇y .
 - 4: Compute interpolated weights: $w_{f'f} = K w_{\tilde{f}f}$.
 - 5: Compute output: $y_{sf'} = U^T \left(\sum_f U x_{sf} \odot w_{f'f} \right)$.
 - 6: **Backward Pass:**
 - 7: Compute gradient w.r.t input: $\nabla x_{sf} = U^T \left(\sum_{f'} \nabla y_{sf'} \odot w_{f'f} \right)$
 - 8: Compute gradient w.r.t interpolated weights: $\nabla w_{\tilde{f}f} = U^T \left(\sum_s \nabla y_{sf'} \odot x_{sf} \right)$
 - 9: Compute gradient w.r.t weights $\nabla w_{\tilde{f}f} = K^T \nabla w_{f'f}$.
-

In the case of a classical ConvNet, the weights w are small filters localized in space. In the case of the grid, localization in space is inversely related to localization in frequency.

3 Graph Estimation

4 Experiments

References

- [1] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and deep locally connected networks on graphs. In *Proceedings of the 2nd International Conference on Learning Representations*, 2013.
- [2] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara Sainath, and Brian Kingsbury. Deep neural networks for acoustic modeling in speech recognition. *Signal Processing Magazine*, 2012.
- [3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.