Graph Convolutional Neural Networks

Vedang Anand Waradpande

Department of Computer Science and Information Systems, Birla Institute of Technology and Science KK Birla Goa Campus

Contents

Definitions

Graph Theory

- $\mathcal{G} = (\mathcal{V}, \mathcal{E}, W)$, where $\mathcal{G}$ is a graph, fully defined by a finite set $\mathcal{V}$ of $|\mathcal{V}| = n$ vertices, a set of edges $\mathcal{E}$ and an adjacency matrix $W \in \mathbb{R}^{n \times n}$.
- $f : \mathcal{V} \to \mathbb{R}$ is a signal defined over the vertices of the graph.
- A Bipartite graph's vertex set can be divided into two disjoint sets, such that no two vertices in the same set are adjacent to each other.
- $D \in \mathbb{R}^{n \times n}$ is the diagonal degree matrix with $D_{i,i} = \Sigma_j W_{i,j}$.

Graph Laplacian. The Normalized Laplacian of a graph, $L = I_n - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \in \mathbb{R}^{n \times n}$ is an operator in Spectral Graph Theory. Laplacian of a graph of n vertices admits n real, orthonormal eigenvectors, $Lu_k = \lambda_k u_k$, where $k = 1, ..., n$, which are also the Fourier modes of the graph, and the eigenvalues λ_k are the frequencies.

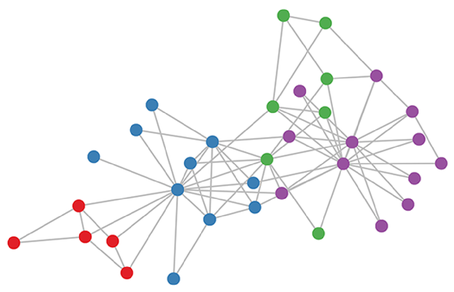

Figure 1: Graph of Zachary's karate club network from 'Semi-Supervised Learning with Graph Convolutional Networks' by Kipf and Welling, ICLR 2017

Defining Graph Convolution

- The graph Laplacian admits a spectral eigen-decomposition, $L = U \Lambda U^T$ where $U = (u_1, ..., u_n)$ is a matrix of the eigenvectors and $\Lambda = diag(\lambda_1, ..., \lambda_n)$ is a diagonal matrix with eigenvalue entries.
- Convolution Theorem: $\widehat{f * g} = \hat{f} \cdot \hat{g}$ in the spectral domain, where $\hat{f}$ is the Fourier transform of the function f.
- Fourier and Inverse Fourier transforms of f are given by $\hat{f} = U^T f$ and $f = U f^T$ respectively.
- Hence Convolution of signal f with filter g can be given by $f * g = U(U^T g \circ U^T f)$.

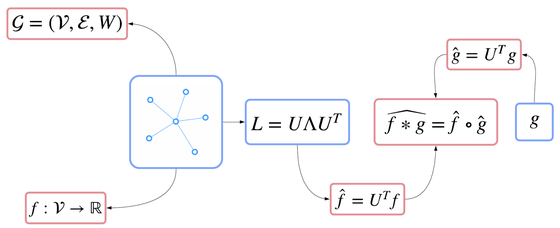

Figure 2: Definition of Graph Convolution operation

Graph Data Science Pipeline

Select graph type.

- Full graph.
- k-Nearest Neighbour graphs.

$$W_{ij} = \begin{cases} e^{-\frac{dist(x_i, x_j)^2}{\sigma^2}} & \text{if } j \in \mathcal{N}_i^k \\ 0 & \text{otherwise} \end{cases}$$

Select distance metric.

- Manhattan distance: $d_{man}(x_1, x_2) = ||x_1 - x_2||_1$
- Euclidean distance: $d_{euc}(x_1, x_2) = ||x_1 - x_2||_2$.
- Cosine distance: $d_{cos}(x_1, x_2) = |cos^{-1}(\frac{<x_1, x_2>}{||x_1||.||x_2||})|$

Preprocessing.

- Centering data, feature selection, dimentionality reduction, normalization, etc.

Machine Learning.

- Apply Machine Learning models (e.g. Graph Convolutional Neural Networks) for generating predictions for a variety of tasks (e.g. regression, classification, clustering, etc.).

Graph Convolutional Models

Table 1: Evolution of Graph Convolutional Networks

Model	Paper	Time Complexity	Space Complexity	Contributions
Spectral Graph Convnets	Bruna et al. (2014)	$O(n^2)$	$O(n)$	First formulation of Graph Convolution layer.
SplineNets	Henaff et al. (2015)	$O(n^2)$	$O(1)$	Smoothening of weights using kernels such as Splines, extension of ideas to Computer Vision and NLP applications.
ChebNets	Defferard et al. (2016)	$O(n)$	$O(1)$	Strictly localized filters, lower computational complexity, efficient algorithm for pooling.
Graph Convolutional Nets	Kipf and Welling (2016)	$O(n)$	$O(1)$	Established simple, well-behaved Graph ConvNets, adapted for semi-supervised classification.
CayleyNets	Levie et al. (2017)	$O(n)$	$O(1)$	Richer class of filters than Chebyshev, better localization in frequency.

Applications

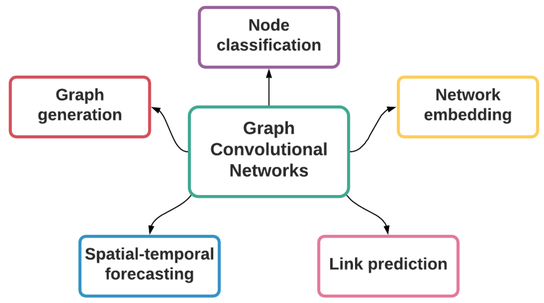

Figure 3: Various Machine Learning tasks possible with Graph Convolutional Networks

Graph Convolutional networks have been applied successfully to image data (Image classification, Visual question answering, etc.), textual data (Sentiment classification, Text classification, etc.), Biological and Chemical data (Molecule fingerprint generation, Drug-target interaction prediction) and Graph generation.

E.g.: Citation Networks

- Graph Convolutional Networks (Kipf and Welling, 2016) were used for semi-supervised node classification on Cora, Citeseer and Pubmed datasets.
- Inputs consist of a feature matrix X and an adjacency matrix A.
- The models gave higher accuracy and lesser training time on these datasets: Citeseer, 70.3 (7s); Cora, 81.5 (4s); Pubmed, 79.0 (38s).

E.g.: Drug-Target Interaction Prediction

- Variational Graph Autoencoder (VGAE) based models are used for this purpose.
- Interaction networks are modelled as Bipartite graphs.
- Higher AUROC score and lesser time were recorded: Random Forest, 76.5 (10 m); EnsemDT (Ezzat et al., 2017), 82.3 (69 m); Variational Graph Autoencoder, 85.6 (2 m).