MLV Lab GNN Study

Graph Attention Networks

Presenter: Jiwon Jeong (Data Science 21)
jjwon4086@korea.ac.kr

Paper

- Published at ICLR 2018 (link)

GRAPH ATTENTION NETWORKS

Petar Veličković*
Department of Computer Science and Technology
University of Cambridge
petar.velickovic@cst.cam.ac.uk

Guillem Cucurull*
Centre de Visió per Computador, UAB
gcucurull@gmail.com

Arantxa Casanova*
Centre de Visió per Computador, UAB
ar.casanova.8@gmail.com

Adriana Romero
Montréal Institute for Learning Algorithms
adriana.romero.soriano@umontreal.ca

Pietro Liò
Department of Computer Science and Technology
University of Cambridge
pietro.lio@cst.cam.ac.uk

Yoshua Bengio
Montréal Institute for Learning Algorithms
yoshua.umontreal@gmail.com

Abstract

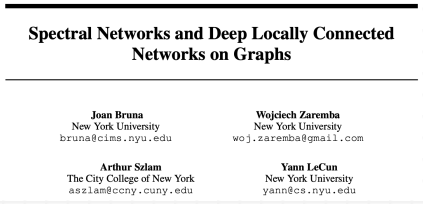
- Graph Attentional Networks (GATs)
  - Graph-structed data
  - Masked self-attentional networks
  - Address the shortcomings of prior methods
    - Computationally efficient
    - Not depend on knowing graph structure
    - Readily applicable to inductive and transductive problems
    - SOTA performance

Introduction

- CNN
  - Grid-like structure
  - But, in many tasks, graph-structed data
- GNN
  - RNN: process data represented in directed acyclic graph
  - GNN: generalization of RNN (cyclic/acyclic, directed/undirected)

Introduction

- Generalizing convolution
  - Spectral approaches
    - Fourier domain by computing the eigendecomposition of the graph Laplacian
    - Graph Laplacian: matrix representation of a graph
      ex. Laplacian matrix = Degree matrix - Adjacency matrix



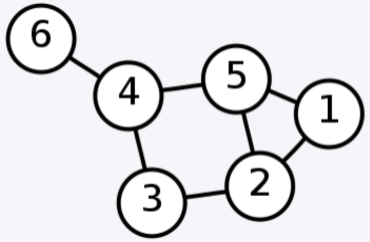
Spectral Networks and Deep Locally Connected Networks on Graphs

Joan Bruna
New York University
bruna@cims.nyu.edu

Wojciech Zaremba
New York University
woj.zaremba@gmail.com

Arthur Szlam
The City College of New York
aszlam@ccny.cuny.edu

Yann LeCun
New York University
yann@cs.nyu.edu

Labelled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2&0&0&0&0&0\\ 0&3&0&0&0&0\\ 0&0&2&0&0&0\\ 0&0&0&3&0&0\\ 0&0&0&0&3&0\\ 0&0&0&0&0&1 \end{pmatrix}$	$\begin{pmatrix} 0&1&0&0&1&0\\ 1&0&1&0&1&0\\ 0&1&0&1&0&0\\ 0&0&1&0&1&1\\ 1&1&0&1&0&0\\ 0&0&0&1&0&0 \end{pmatrix}$	$\begin{pmatrix} 2&-1&0&0&-1&0\\ -1&3&-1&0&-1&0\\ 0&-1&2&-1&0&0\\ 0&0&-1&3&-1&-1\\ -1&-1&0&-1&3&0\\ 0&0&0&-1&0&1 \end{pmatrix}$

Introduction

- Generalizing convolution
  - Spectral approaches
    - Fourier domain by computing the eigendecomposition of the graph Laplacian
    - Intense computations, non-spatially localized filters
    - Learned filters depend on the Laplacian eigenbasis
      : not applied to graph with a different structure
  - Non-spectral approaches
    - Operating on groups of spatially close neighbors
    - How to define operators?

Introduction

- Generalizing convolution
  - Non-spectral approaches
    - Works with different sized neighborhoods
    - Maintains the weight sharing property of CNNs
  - MoNet
  - GraphSAGE
    - Impressive performance across large-scale inductive benchmarks

Introduction

- Attention
  - Benefits
    - Dealing with variable sized inputs
    - Focusing on the most relevant parts of the input to make decisions
  - Self-attention

Introduction

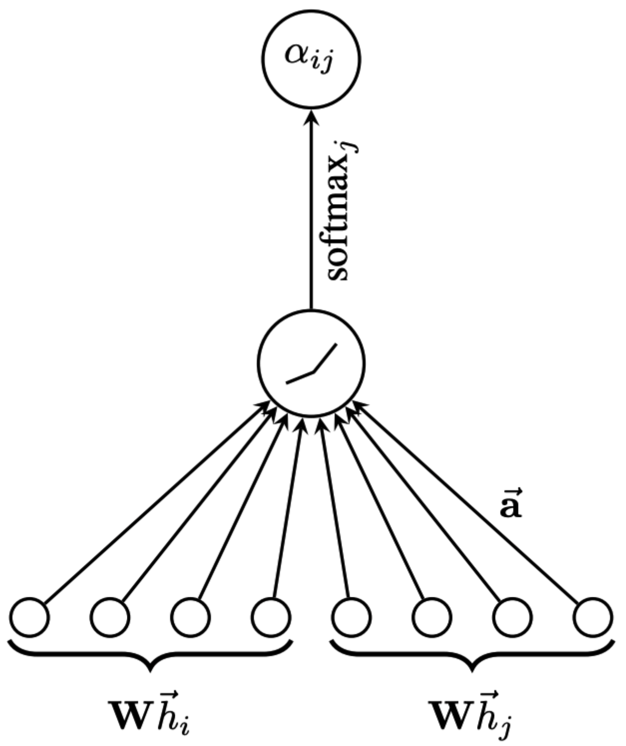
- Attention-based architecture
  - Perform node classification of graph-structed data
  - Self-attention strategy
    - Efficient operations
    - Specifying arbitrary weights to the neighbors
    - Directly applicable to inductive learning problems

GAT Architecture

- Graph Attentional Layer
  - Layer $\mathbf{h} \rightarrow \mathbf{h}'$ $(\mathbb{R}^F \rightarrow \mathbb{R}^{F'})$
    - Node features $\mathbf{h} = \{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\}, \vec{h}_i \in \mathbb{R}^F$ (N: # of nodes, F: # of features)
    - Layer outputs $\mathbf{h}' = \{\vec{h}'_1, \vec{h}'_2, \ldots, \vec{h}'_N\}, \vec{h}'_i \in \mathbb{R}^{F'}$ (F': Dim of Hidden embeddings)
  - Masked self-attention
  - Multi-head attention

GAT Architecture

- Masked self-attention
  - Attention coefficients $e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$ $(j \rightarrow i)$
    - $j \in \mathcal{N}_i$ (Neighborhood of node i)
    - $\mathbf{W} \in \mathbb{R}^{F' \times F}$: Learnable parameters
    - $a(\,\cdot\,)$: shared attentional mechanism $\mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R}$
    - Inject graph structure by performing masked attention



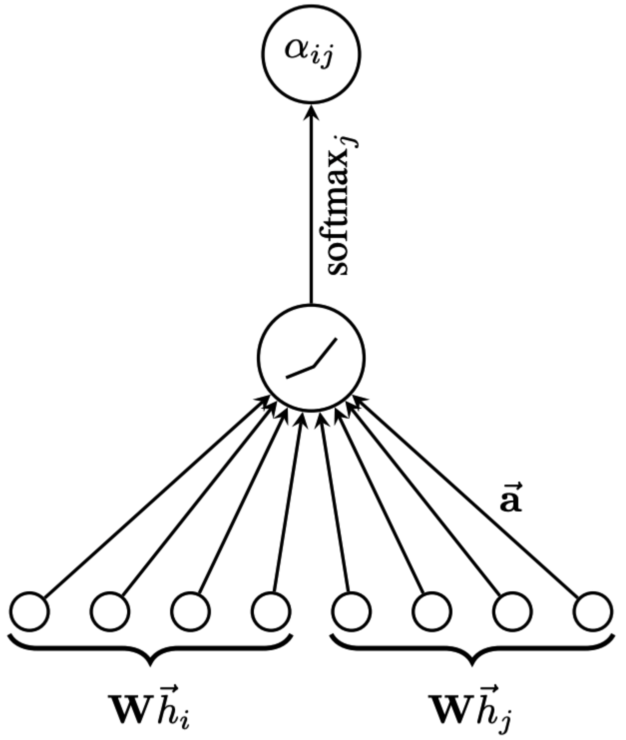
GAT Architecture

- Masked self-attention
  - Normalize:

$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(e_{ik})}$$

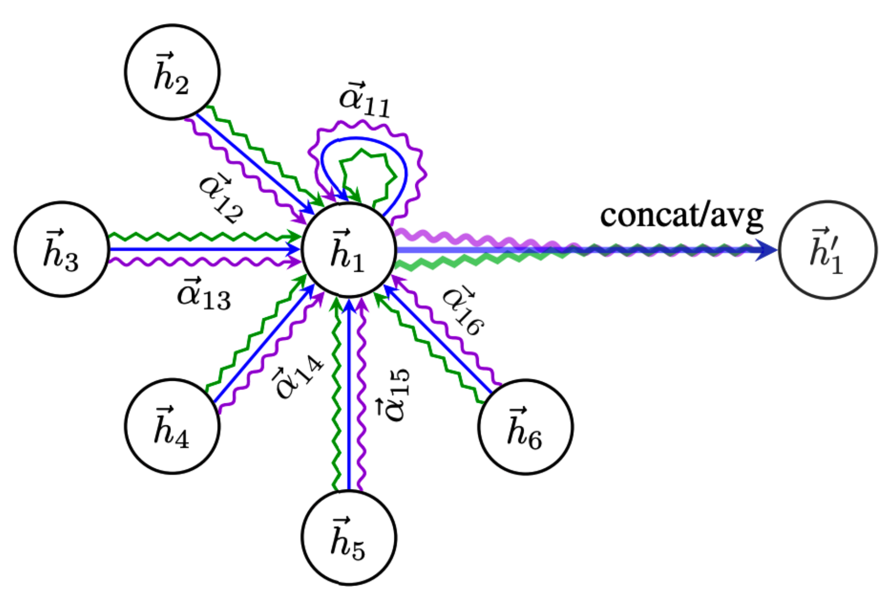
  - Expanded:

$$\alpha_{ij} = \frac{\exp\left(\mathrm{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\mathrm{LeakyReLU}\left(\vec{\mathbf{a}}^T[\mathbf{W}\vec{h}_i \| \mathbf{W}\vec{h}_k]\right)\right)}$$



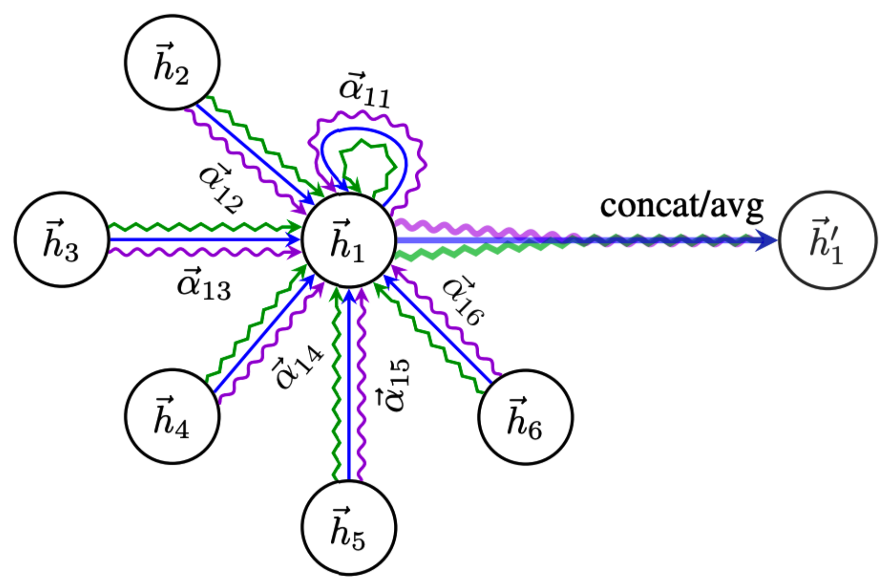
GAT Architecture

- Single-head attention
  - $$\vec{h}'_i = \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{h}_j\right)$$
- Multi-head attention (concatenation)
  - $$\vec{h}'_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W} \vec{h}_j\right) \rightarrow \mathbf{h}' \in \mathbb{R}^{KF'}$$



GAT Architecture

- Multi-head attention
  - Employ averaging
  - Delay applying final nonlinearity
  - $$\vec{h}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i}\alpha_{ij}^k\mathbf{W}^k\vec{h}_j\right)$$
    - $K = 3$



Comparison to Related Work

- Highly efficient computation
  - The operation of the self-attentional layer can be parallelized
    - Single: $O(|V|FF' + |E|F')$
    - Multi: storage and parameter requirements $\times K$, independent and parallelized
- As opposed to GCN,
  - Assigning different importances to nodes of a same neighborhood
    - Leap in model capacity, Benefits in interpretability

Comparison to Related Work

- Attention mechanism
  - Applied in a shared manner to all edges in the graph
  - Not depend on upfront access to the global graph structure
    - Not required to be undirected ($\alpha_{ij} : j \rightarrow i$)
    - Applicable to inductive learning
      : evaluated on graphs that completely unseen during training

Comparison to Related Work

- Vs. prior inductive method
  - Works with the entirety of the neighborhood
  - Does not assume any ordering within nodes
- Vs. MoNet
  - Use node features for similarity computations,
    rather than the node's structural properties (knowing the graph structure)

Evaluation

- Results

Transductive

Method	Cora	Citeseer	Pubmed
MLP	55.1%	46.5%	71.4%
ManiReg (Belkin et al., 2006)	59.5%	60.1%	70.7%
SemiEmb (Weston et al., 2012)	59.0%	59.6%	71.7%
LP (Zhu et al., 2003)	68.0%	45.3%	63.0%
DeepWalk (Perozzi et al., 2014)	67.2%	43.2%	65.3%
ICA (Lu & Getoor, 2003)	75.1%	69.1%	73.9%
Planetoid (Yang et al., 2016)	75.7%	64.7%	77.2%
Chebyshev (Defferrard et al., 2016)	81.2%	69.8%	74.4%
GCN (Kipf & Welling, 2017)	81.5%	70.3%	**79.0%**
MoNet (Monti et al., 2016)	$81.7 \pm 0.5\%$	—	$78.8 \pm 0.3\%$
GCN-64*	$81.4 \pm 0.5\%$	$70.9 \pm 0.5\%$	$\mathbf{79.0} \pm 0.3\%$
GAT (ours)	$\mathbf{83.0} \pm 0.7\%$	$\mathbf{72.5} \pm 0.7\%$	$\mathbf{79.0} \pm 0.3\%$

Inductive

Method	PPI
Random	0.396
MLP	0.422
GraphSAGE-GCN (Hamilton et al., 2017)	0.500
GraphSAGE-mean (Hamilton et al., 2017)	0.598
GraphSAGE-LSTM (Hamilton et al., 2017)	0.612
GraphSAGE-pool (Hamilton et al., 2017)	0.600
GraphSAGE*	0.768
Const-GAT (ours)	0.934 ± 0.006
GAT (ours)	$\mathbf{0.973} \pm 0.002$

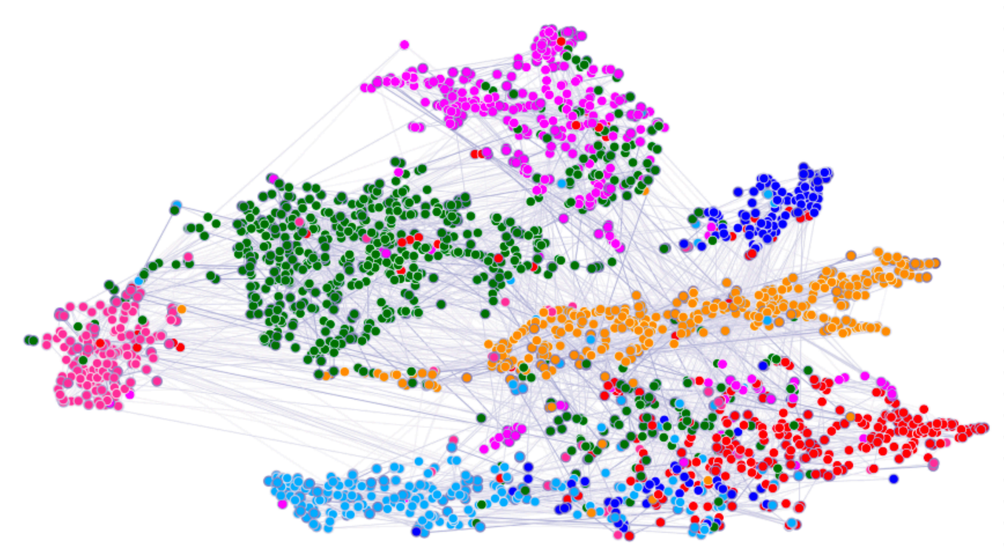
  - Transductive: mean classification accuracy

    Inductive: micro-averaged F_1 score on the nodes of the two unseen test graphs

  - Const-GAT: significance of being able to assign different neighbors

Evaluation

- Results



  - Visualization of the t-SNE-transformed feature representations
  - The representation exhibits discernible clustering in the projected 2D space

Conclusions

- Graph attention networks (GATs)
  - Novel convolution-style neural networks that operate on graph-structured data, leveraging masked self-attentional layers
  - Computationally efficient
  - Assigning different importances to different nodes within a neighborhood
  - Not depend on knowing the entire graph structure upfront
  - Successfully achieved or matched state-of-the-art performance

Questions?