Graph Attention Networks

School of Industrial and Management Engineering, Korea University

Eun Ji Koh

KOREA UNIVERSITY

DMQA

Contents

Research Purpose

❖ Graph Attention Networks (ICLR, 2018)

- Cambridge 대학에서 연구하였고 2022년 02월 03일 기준으로 2801회 인용됨

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

We present graph attention networks (GATs), novel neural network architectures that operate on graph-structured data, leveraging masked self-attentional layers to address the shortcomings of prior methods based on graph convolutions or their approximations. By stacking layers in which nodes are able to attend over their neighborhoods' features, we enable (implicitly) specifying different weights to different nodes in a neighborhood, without requiring any kind of costly matrix operation (such as inversion) or depending on knowing the graph structure upfront. In this way, we address several key challenges of spectral-based graph neural networks simultaneously, and make our model readily applicable to inductive as well as transductive problems. Our GAT models have achieved or matched state-of-the-art results across four established transductive and inductive graph benchmarks: the *Cora*, *Citeseer* and *Pubmed* citation network datasets, as well as a *protein-protein interaction* dataset (wherein test graphs remain unseen during training).

DMQA

Research Purpose

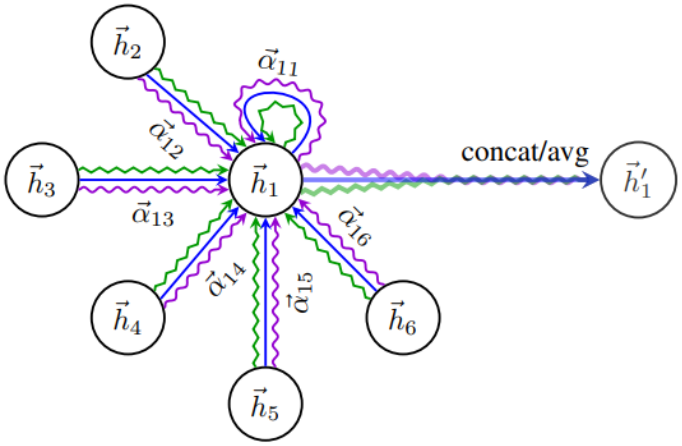
❖ Graph Attention Networks (ICLR, 2018)

- 본 논문은 node classification을 위한 attention-based architecture 제안
- Graph domain에 convolution을 접목하려는 다양한 기존 연구가 진행됨
  - ➢ Spectral approach는 graph의 laplacian eigenbasis에 의존하기 때문에, 특정 graph구조에 대해 학습된 모델을 새로운 구조의 graph에 바로 적용하는 것이 불가
  - ➢ Non-spectral approach는 weight sharing 적용과 이웃 node들과의 연산을 가능케 하는 방향으로 발전 중
  - ➢ 본 논문은 RNN, CNN based model에서 성능 향상을 이끈 attention architecture를 graph에 적용해 보고자 함
- Attention-based architecture 의 이점
  1) Node-neighbor pair에 걸쳐 parallelizable 하기 때문에 연산이 효율적
  2) 이웃 node에 각기 다른 weight를 지정함으로써 다른 degree를 갖는 graph node에도 적용 가능
  3) Inductive learning problem에 바로 적용할 수 있기 때문에 이전에 본적 없는 구조의 graph에 대해서도 일반화 가능

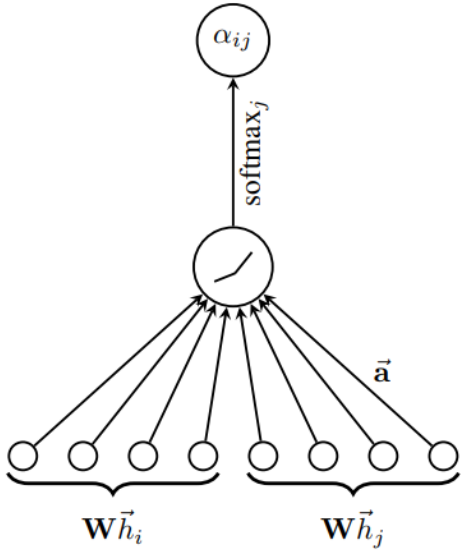
Graph Attention Networks (GATs)

❖ GAT architecture

- GAT는 attention mechanism을 사용하는 non-spectral convolution approach



An illustration of multi-head attention (with 3 heads) by node 1 on its neighborhood



Attention mechanism

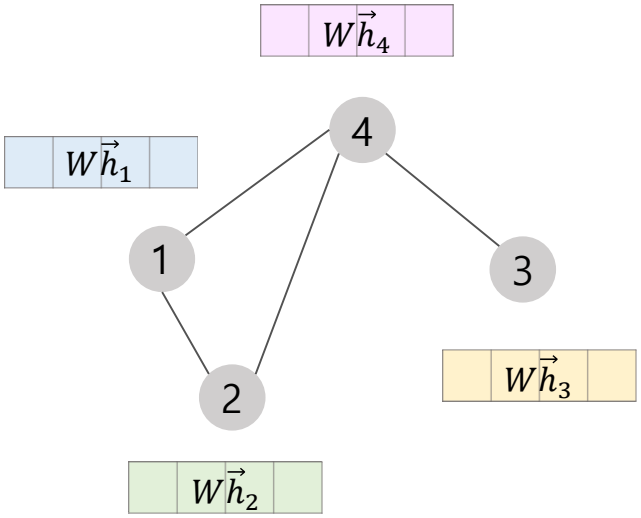
Graph Attention Networks (GATs)

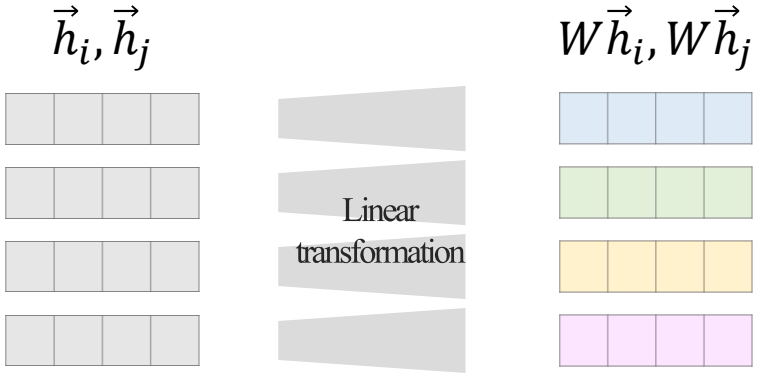
❖ Graph attentional layer

- GAT는 Bahdanau attention를 따라 single-layer feedforward neural network를 사용하여 attention score를 계산
  - ➢ Transformer가 scaled dot product attention을 사용한 것과는 차이가 있음

◆ Graph attentional layer 작동 과정

- Layer input인 node들의 feature를 higher-level feature로 표현하기 위해 초기 step에서 linear transformation 사용
  - ➢ 각 node에 대해 개별적으로 적용





- Layer input: set of node features $h = \{\vec{h}_1, \vec{h}_2, \cdots, \vec{h}_N\}$, $\vec{h}_i \in R^F$
- Layer input: set of node features $h' = \{\vec{h'}_1, \vec{h'}_2, \cdots, \vec{h'}_N\}$, $\vec{h'}_i \in R^{F'}$
- N: the number of nodes / F: the number of features in each node
- i : target node / j : neighbor node

DMQA

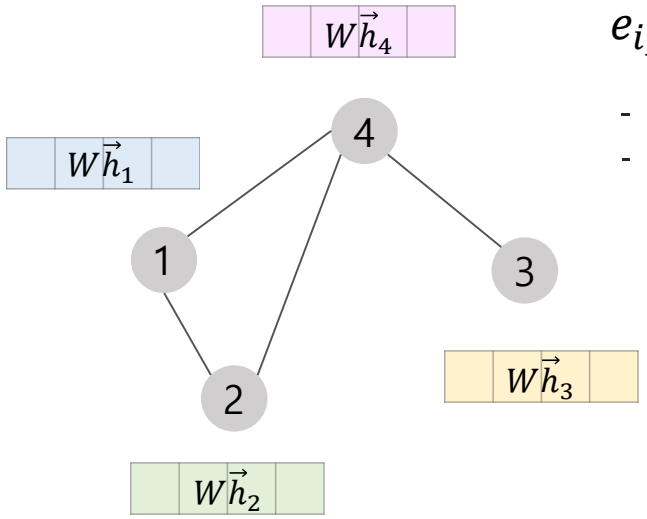
Graph Attention Networks (GATs)

❖ Graph attentional layer

- GAT는 Bahdanau attention를 따라 single-layer feedforward neural network를 사용하여 attention score를 계산
  - ➢ Transformer가 scaled dot product attention을 사용한 것과는 차이가 있음

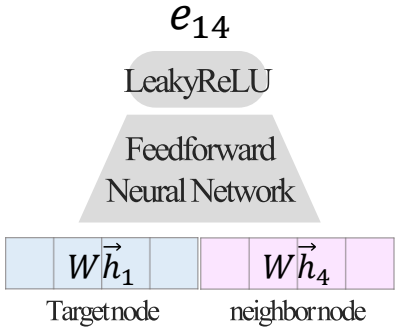
◆ Graph attentional layer 작동 과정

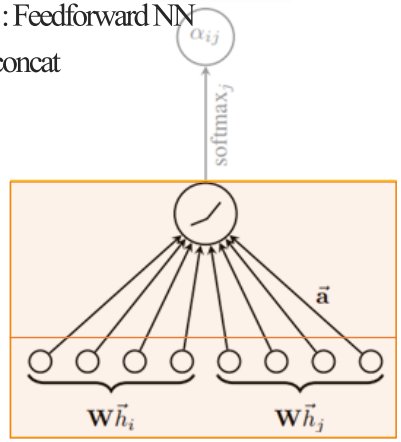
- Single-layer feedforward neural network와 LeakyReLU로 구성된 Self-attention을 target node와 이웃 node들에 대해 수행
  - ➢ Self-attention에 사용되는 key, query, value는 모두 해당 node의 hidden feature (vector)로 동일



$$e_{ij} = a(W\vec{h}_i, W\vec{h}_j) = LeakyReLU(\vec{a}^T[W\vec{h}_i || W\vec{h}_j])$$

- attention mechanism $a : R^{F'} \times R^{F'} \rightarrow R$ / a : Feedforward NN
- e_{ij} : target node와 neighbor node 간의 similarity / $\|$: concat





DMQA

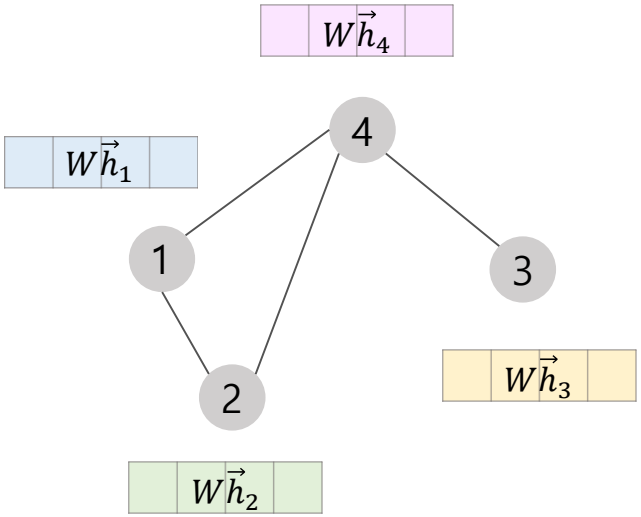
Graph Attention Networks (GATs)

❖ Graph attentional layer

- GAT는 Bahdanau attention를 따라 single-layer feedforward neural network를 사용하여 attention score를 계산
  - ➢ Transformer가 scaled dot product attention을 사용한 것과는 차이가 있음

◆ Graph attentional layer 작동 과정

- Single-layer feedforward neural network와 LeakyReLU로 구성된 Self-attention을 target node와 이웃 node들에 대해 수행
  - ➢ Graph의 구조를 반영하기 위해 adjacency matrix 사용하여 masked attention 수행



e_{11}	e_{12}	e_{13}	e_{14}
e_{21}	e_{22}	e_{23}	e_{24}
e_{31}	e_{32}	e_{33}	e_{34}
e_{41}	e_{42}	e_{43}	e_{44}

모든 node 간의 similarity
(그래프 구조 반영 X)

e_{11}	e_{12}	$-9e^{20}$	e_{14}
e_{21}	e_{22}	$-9e^{20}$	e_{24}
$-9e^{20}$	$-9e^{20}$	e_{33}	e_{34}
e_{41}	e_{42}	e_{43}	e_{44}

Masking
(그래프 구조 반영)

DMQA

Graph Attention Networks (GATs)

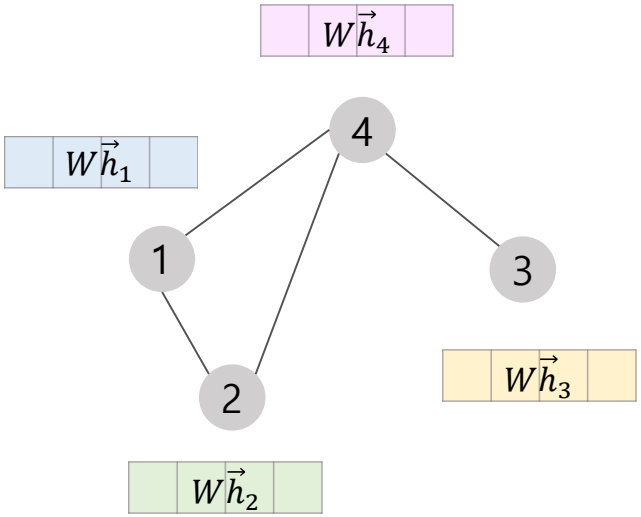
❖ Graph attentional layer

- GAT는 Bahdanau attention를 따라 single-layer feedforward neural network를 사용하여 attention score를 계산
  - ➢ Transformer가 scaled dot product attention을 사용한 것과는 차이가 있음

◆ Graph attentional layer 작동 과정

- Single-layer feedforward neural network와 LeakyReLU로 구성된 Self-attention을 target node와 이웃 node들에 대해 수행
  - ➢ Softmax 함수를 사용하여 normalized attention coefficients 산출

$$\alpha_{ij} = Softmax_i(e_{ij})$$



e_{11}	e_{12}	$-9e^{20}$	e_{14}
e_{21}	e_{22}	$-9e^{20}$	e_{24}
$-9e^{20}$	$-9e^{20}$	e_{33}	e_{34}
e_{41}	e_{42}	e_{43}	e_{44}

Masking
(그래프 구조 반영)

α_{11}	α_{12}	0	α_{14}
α_{21}	α_{22}	0	α_{24}
0	0	α_{33}	α_{34}
α_{41}	α_{42}	α_{43}	α_{44}

Normalized attention coefficients
(Softmax)

DMQA

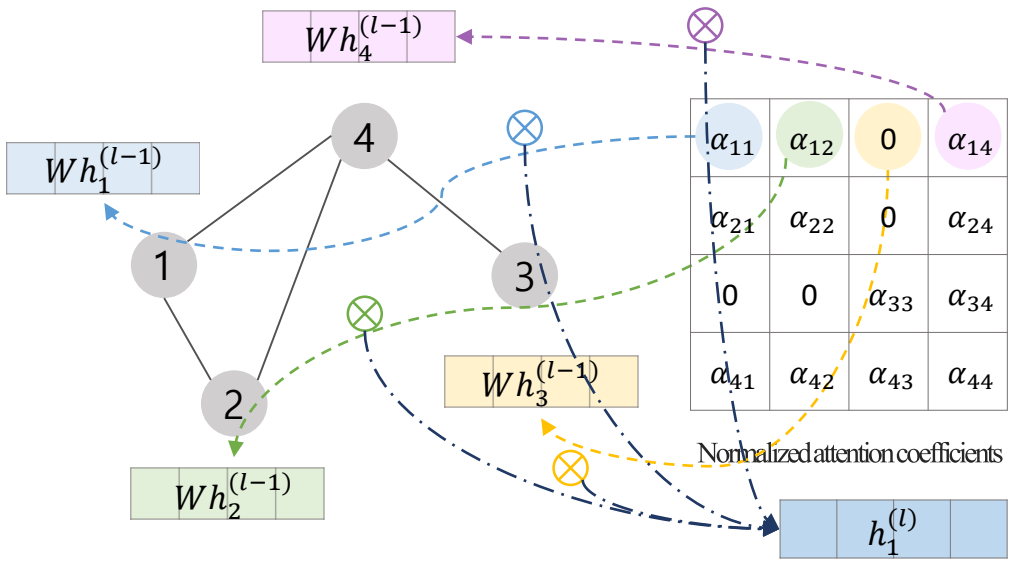
Graph Attention Networks (GATs)

❖ Graph attentional layer

- GAT는 Bahdanau attention를 따라 single-layer feedforward neural network를 사용하여 attention score를 계산
  - ➢ Transformer가 scaled dot product attention을 사용한 것과는 차이가 있음

◆ Graph attentional layer 작동 과정

- 각 node에 대해 normalized attention coefficients를 weight로 사용하여 업데이트 된 feature $\vec{h}'_i$ 출력
  - ➢ K개의 head를 갖는 multi-head attention 을 적용



$$\vec{h}'_i = \sigma\left(\sum_{j \in N_i} \alpha_{ij} W \vec{h}_j\right)$$

Multi-head attention

$$\vec{h}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j \in N_i} \alpha_{ij} W \vec{h}_j\right)$$

Experiments

❖ Evaluation

- 4가지 graph-based benchmark task에 대해 성능 평가 수행
  - ➢ Transductive learning 측면에서는 Cora, Citeseer, Pubmed dataset에 대하여 성능 평가
  - ➢ Inductive learning 측면에서는 protein-protein interaction (PPI) dataset에 대하여 성능 평가

Table 1: Summary of the datasets used in our experiments.

	Cora	**Citeseer**	**Pubmed**	**PPI**
Task	Transductive	Transductive	Transductive	Inductive
# Nodes	2708 (1 graph)	3327 (1 graph)	19717 (1 graph)	56944 (24 graphs)
# Edges	5429	4732	44338	818716
# Features/Node	1433	3703	500	50
# Classes	7	6	3	121 (multilabel)
# Training Nodes	140	120	60	44906 (20 graphs)
# Validation Nodes	500	500	500	6514 (2 graphs)
# Test Nodes	1000	1000	1000	5524 (2 graphs)

Experiments

❖ Results

- GATs는 모든 dataset에 대해 기존 모델과 유사하거나 더 뛰어난 성능을 보임
  - ➢ Cora, Citeseer dataset에서 GCN과의 성능 비교를 통해 이웃 node에 서로 다른 weight를 부여하는 GAT 구조가 효과적임을 보임
  - ➢ PPI dataset에서 GraphSAGE와의 성능 비교를 통해 GAT는 전체 graph의 구조를 반영하면 더 좋은 예측 성능을 낼 수 있음을 보임

Table 2: Summary of results in terms of classification accuracies, for Cora, Citeseer and Pubmed. GCN-64* corresponds to the best GCN result computing 64 hidden features (using ReLU or ELU).

	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 ± 0.5%	—	78.8 ± 0.3%
GCN-64*	81.4 ± 0.5%	70.9 ± 0.5%	**79.0** ± 0.3%
GAT (ours)	**83.0** ± 0.7%	**72.5** ± 0.7%	**79.0** ± 0.3%

Table 3: Summary of results in terms of micro-averaged F_1 scores, for the PPI dataset. GraphSAGE* corresponds to the best GraphSAGE result we were able to obtain by just modifying its architecture. Const-GAT corresponds to a model with the same architecture as GAT, but with a constant attention mechanism (assigning same importance to each neighbor; GCN-like inductive operator).

	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)	**0.973** ± 0.002

Experiments

❖ Results

- T-SNE를 통해 feature representation을 시각화하여 node classification 성능을 정성적으로 평가
  - ➢ 각 node classification에 영향을 주고 받은 feature 간의 관계를 표현하며 edge의 두께는 weight, 색상은 node의 class를 의미

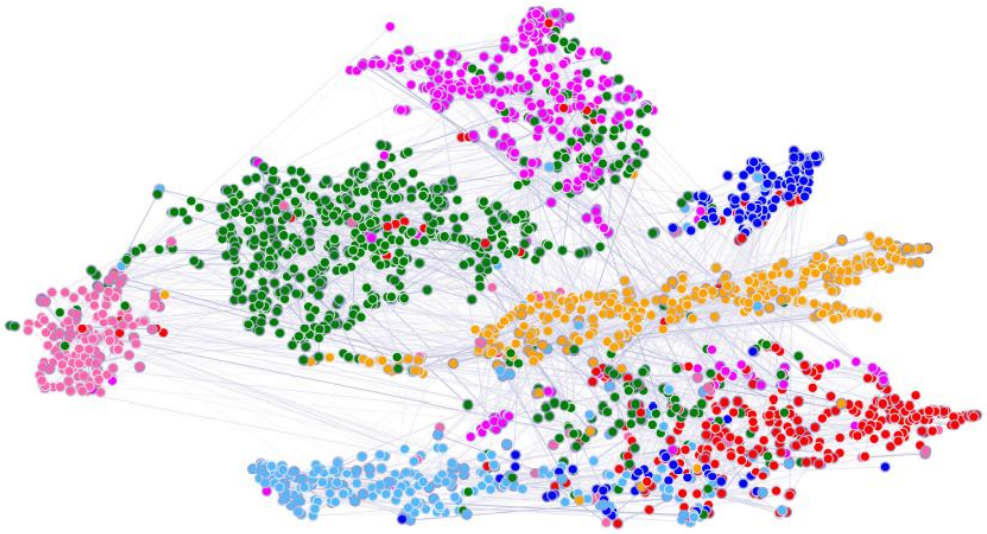


Figure 2: A t-SNE plot of the computed feature representations of a pre-trained GAT model's first hidden layer on the Cora dataset. Node colors denote classes. Edge thickness indicates aggregated normalized attention coefficients between nodes i and j, across all eight attention heads $(\sum_{k=1}^{K} \alpha_{ij}^{k} + \alpha_{ji}^{k})$.

Conclusion

❖ conclusion

- GAT는 다음과 같은 이점을 갖음
  - ➢ Computational efficiency
  - ➢ Do not depend on knowing the entire graph structure (이를 통해 spectral convolution approach의 issue들 해결 가능)
- GAT는 transductive learning과 inductive learning 모두에서 기존 모델 성능을 필적하거나 능가

Reference

1. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). Graph attention networks. arXiv preprint arXiv:1710.10903.
2. DMQA open seminar http://dmqm.korea.ac.kr/activity/seminar/296


Thank You