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

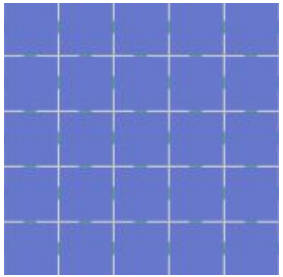
Arantxa Casanova*
Centre de Visió per Computador, UAB
ar.casanova.8@gmail.com

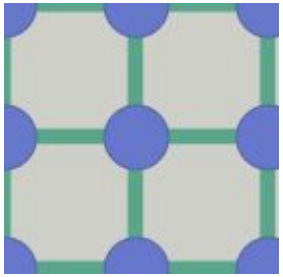
Adriana Romero
Montréal Institute for Learning Algorithms
adriana.romero.soriano@umontreal.ca

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

Introduction





Grid vs Graph 구조 데이터

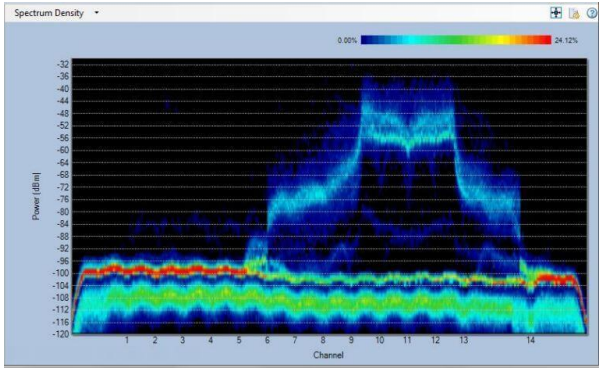
- **CNN**은 데이터가 **grid** 형태로 표현되는 분야에 성공적이었음
  (image classification, semantic segmentation, machine translation 등)
- 하지만 **graph** 구조로 표현되어야 하는 분야들도 존재함
  (3D Meshes, social networks, telecommunication networks, biological networks, brain connections, 등)

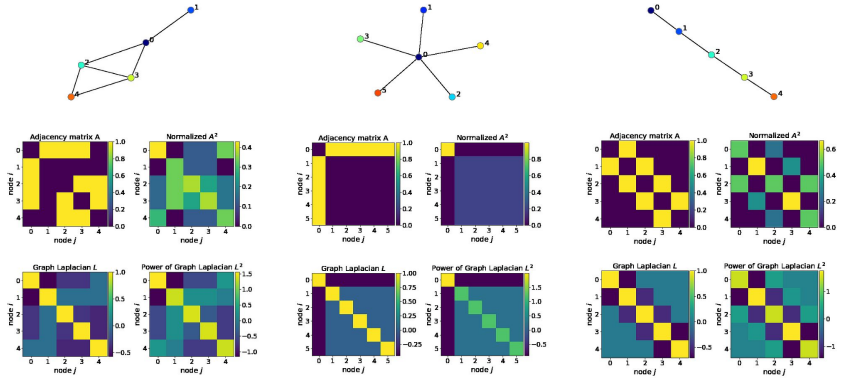
Graph에 대한 Neural Network 기반 접근법

- **Spectral Representation**
  - Graph Laplacian의 고유값 분해를 계산 후 Fourier domain 내 합성곱 연산
  - intense한 연산과 non-spatially localized한 필터 생성
  - 세부적인 구조와 피처의 특성을 반영하는데 특화
  - 새로운 구조 그래프나 새로운 노드에 대한 정보 반영 어려움
- **Non-spectral & Spatial Representation**
  - 합성곱을 Graph에 직접적으로 적용하고, 인접(spatially close)한 이웃 그룹에 대해 연산을 수행(e.g. GraphSAGE)
  - GraphSAGE는 스케일이 큰 Large Graph에 대해 효과적인 접근법으로 평가됨

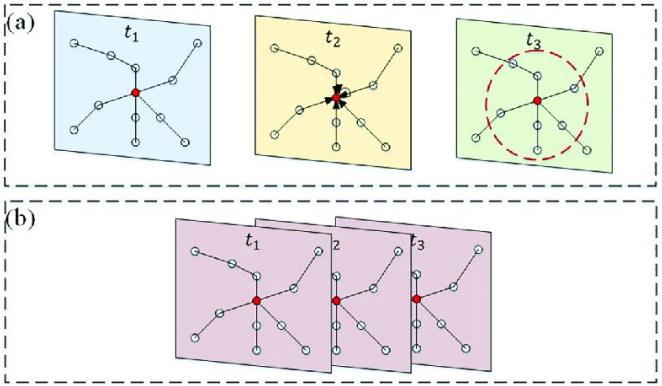
Introduction

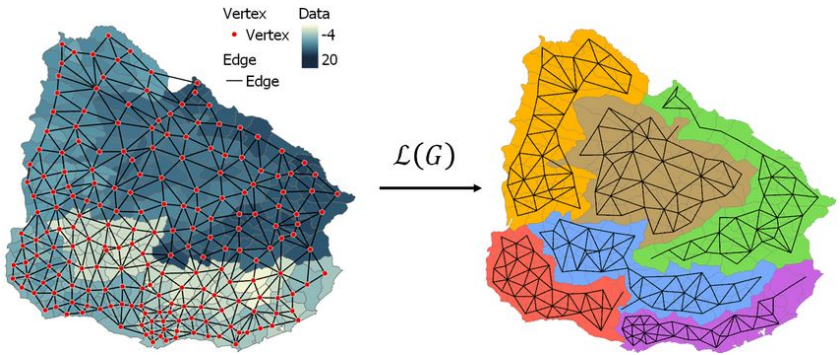
Spectral Representation





Spatial Representation

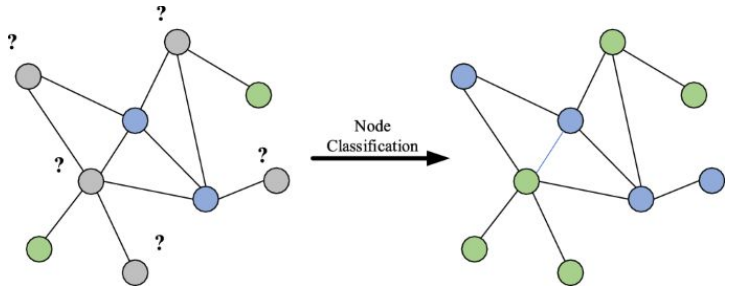




Introduction

Node Classification

- 그래프 내의 노드의 라벨(Node Type, Feature)을 분류
  - Node Type은 노드가 속한 범주나 클래스
    (ex. 소셜 네트워크 그래프: 사람, 게시물, 댓글 등)
  - Node Feature는 노드를 설명하는 추가적인 정보나 속성 의미
    (ex. 소셜 네트워크 그래프 "사람" 노드의 특성: 나이, 성별, 관심사 등)



Attention vs. Self-Attention Mechanism

- **Attention**
  - **모델이 입력 데이터의 특정 부분에 집중**할 수 있도록
    각 부분에 다른 가중치를 할당하는 방법
  - ex. 기계 번역에서 타겟 문장의 각 단어를 번역할 때
    소스 문장의 관련 단어에 집중할 수 있도록 함
- **Self-Attention**
  - **입력 데이터가 자기 자신에게 집중하는 Attention의 특수한 경우**로,
    동일한 시퀀스로 서로 다른 위치 간의 Attention 점수를 계산
  - ex. 자연어처리에서는 Self-Attention이 문장 내의 각 단어가
    다른 단어에 집중할 수 있도록 하여 단어 간의 관계를 포착

Introduction

Graph Attention Networks(GAT) 모델

- 그래프 구조의 데이터에 대해 **Node Classification** 위해 **Attention** 메커니즘 사용

GAT의 장점

- **연산의 효율성**
  - Attention 가중치 계산 시 모든 이웃을 고려하지 않고 특정 벡터를 신경망에 통과시켜 계산
  - 그래프 크기가 커질수록 계산 복잡도 크게 감소
  - 병렬 처리 가능
- **노드의 차수(degree)에 무관한 적용 가능성**
  - Attention 메커니즘을 통해 이웃 노드들의 중요도를 학습
  - 노드의 차수와 관계없이 유연하게 적용 가능
  - 각 노드는 이웃 노드들에 대해 서로 다른 중요도(Weight) 할당
  - 그래프의 구조적 특성 반영 가능
- **귀납적 학습 가능성**
  - 귀납적 학습(Inductive Learning)이 가능한 모델
  - 학습 단계에서 보지 못한 데이터에 대해 일반화 가능
  - 새로운 그래프에 대해서도 예측 수행 가능
  - 새로운 그래프 데이터에 대해 높은 성능을 보임

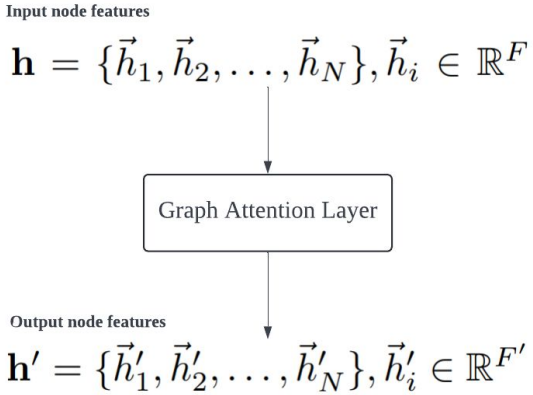
GAT Architecture

GAT 모델의 주요 구성 요소 및 수식

$$a : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \rightarrow \mathbb{R} \qquad e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$

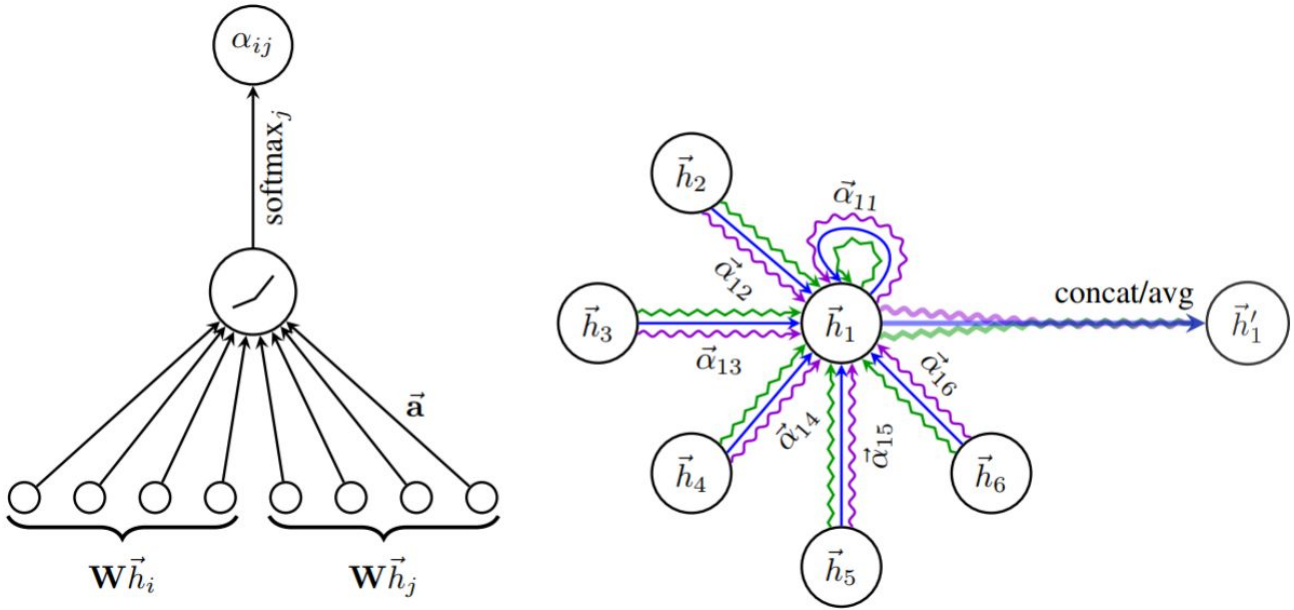
$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}.$$

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



GAT Architecture

Attention 메커니즘과 그래프 상 Multi-head Attention 예시



GAT Architecture

GAT (Graph Attention Network) 모델의 동작 프로세스 수식

1. 단일 헤드 그래프 어텐션 레이어에서 **노드i 의 새로운 특징 벡터 hi'를 계산**하는 과정

2. **다중 헤드** 그래프 어텐션 레이어에서 노드 i의 새로운 특징 벡터 hi'를 계산하는 과정

   K개의 독립적인 어텐션 헤드 각각에 대해 동일한 과정을 수행한 후, 각 헤드의 출력을 연결(Concatenation)

3. **다중 헤드** 그래프 어텐션 레이어의 **출력을 평균 내어 노드 i의 새로운 특징 벡터 hi'를 계산**하는 과정

   각 어텐션 헤드의 출력을 평균낸 후, 활성화 함수

$$\vec{h}_i' = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\mathbf{W}\vec{h}_j\right). \qquad \vec{h}_i' = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k\mathbf{W}^k\vec{h}_j\right) \qquad \vec{h}_i' = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k\mathbf{W}^k\vec{h}_j\right)$$

Datasets

실험에 사용한 데이터셋 : **Cora, Citeseer, Pubmed**

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)

Datasets

Transductive Learning

- 표준 인용 네트워크 벤치마크 데이터셋 활용 (Cora, Citeseer, Pubmed)
- 각 노드는 문서를 나타내고, 에지는 인용 관계 표현
- 학습에는 클래스별 20개 노드만 사용하지만, 모든 노드의 특징 벡터에 접근 가능
- 1000개 테스트 노드와 500개 검증 노드로 성능 평가

Inductive Learning

- 단백질 상호작용 (Protein-Protein Interaction, PPI) 데이터셋 활용
- 서로 다른 인간 조직에 해당하는 24개 그래프로 구성 (20개 학습, 2개 검증, 2개 테스트)
- 테스트 그래프는 학습 중 관찰되지 않음을 전제
- 평균 노드 수는 2372개이며, 각 노드는 50개의 특징과 121개의 레이블 보유

Experimental Setup

Transductive Learning

- 2-layer GAT 모델 적용
  - 첫 번째 레이어: 8개 어텐션 헤드, 각 헤드는 8개 특징 계산
  - 두 번째 레이어: 1개 어텐션 헤드, 클래스 수에 해당하는 특징 계산
- L2 정규화 ($\lambda = 0.0005$), Dropout ($p = 0.6$) 적용
- Pubmed 데이터셋은 학습 예제 수가 적어 약간의 아키텍처 변경 필요

Inductive Learning

- 3-layer GAT 모델 적용
  - 처음 2개 레이어: 4개 어텐션 헤드, 각 헤드는 256개 특징 계산
  - 마지막 레이어: 6개 어텐션 헤드, 각각 121개 특징 계산 후 평균
- Skip connection 적용, 배치 크기는 2개 그래프
- 어텐션 메커니즘의 효과를 평가하기 위해 상수 어텐션 메커니즘 (모든 이웃에 동일한 중요도 할당) 적용 모델도 비교

Results - Transductive

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)	$81.7 \pm 0.5\%$	—	$78.8 \pm 0.3\%$
GCN-64*	$81.4 \pm 0.5\%$	$70.9 \pm 0.5\%$	$\mathbf{79.0} \pm 0.3\%$
GAT (ours)	$\mathbf{83.0} \pm 0.7\%$	$\mathbf{72.5} \pm 0.7\%$	$\mathbf{79.0} \pm 0.3\%$

Results - Inductive

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)	$\mathbf{0.973} \pm 0.002$

Results - Visualization

GAT 모델의 첫 번째 히든 레이어에서 계산된 피처 표현의 **t-SNE** 플롯

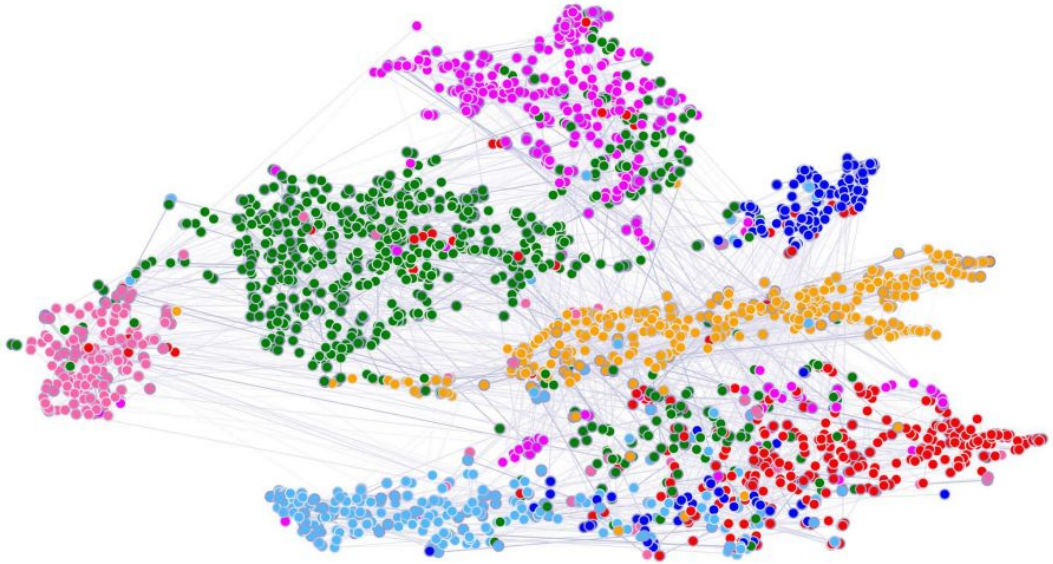


Figure 2: A t-SNE plot of the computed feature representations of a pre-trained GAT model's first hidden layer on the Cora dataset. Node colors denote classes. Edge thickness indicates aggregated normalized attention coefficients between nodes i and j, across all eight attention heads $(\sum_{k=1}^{K}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

Conclusion

- Graph Attention Networks (GATs) 제안
  - Masked Self-Attentional 레이어를 활용한 새로운 Convolution 기반 신경망
  - 계산 효율성, 서로 다른 크기의 이웃에 대한 dynamic attention, 귀납적 학습 등 이점
- Transductive 및 Inductive 노드 분류 벤치마크에서 최고 성능 달성 또는 동등한 성능
- 향후 연구 방향
  - 대용량 배치 처리를 위한 개선
  - 모델 해석 가능성 향상을 위한 어텐션 메커니즘 활용
  - 그래프 분류 작업으로 확장
  - 에지 특징 통합을 통한 다양한 문제 해결