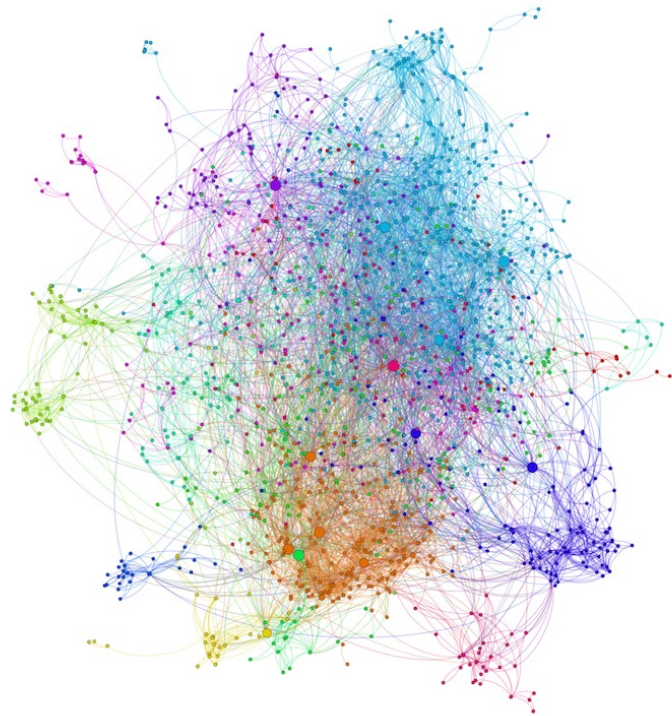


# CUAI GNN 스터디

2022.04.05

발표자 : 배병현



## 목차

1. 스터디 인증
2. 스터디 내용 공유
  - The Over-smoothing Problem in GNN

# 스터디 인증

세번째 미팅: 22.03.24

4. Link Analysis: PageRank    5. Label Propagation for Node Classification

05-message.pdf  
21/69페이지

Jure Leskovec, Stanford University  
<http://cs224w.stanford.edu>

## Probabilistic Relational Classifier (1)

Network에서 각 Node Label의 전파

- **Idea: Propagate node labels across the network**
- ✓ Class probability  $Y_v$  of node  $v$  is a weighted average of class probabilities of its neighbors.  
*Node v의 Label(분류)  $Y_v$ 은 이웃한 Node의 Label의 가중평균*
- For **labeled nodes**  $v$ , initialize label  $Y_v$  with ground-truth label  $Y_v^*$ .  
*이미 Label이 있는 Node*
- For **unlabeled nodes**, initialize  $Y_v = 0.5$ . (uniform)  
*\* Every Classification 가능*
- **Update** all nodes in a random order until convergence or until maximum number of iterations is reached.  
*순환하여 여러 번 반복  
iteration*

10/14/21 Jure Leskovec, Stanford CS224W: Machine Learning with Graphs, <http://cs224w.stanford.edu> 23

배경  
한  
권에  
진

# 스터디 인증

네번째 미팅: 22.03.31

6. Graph Neural Networks 1: GNN Model

7. Graph Neural Networks 2: Design Space

3-3. Matrix Formulation

실제 그래프 구조는 인접행렬 등의 행렬로 표현되기 때문에, 연산 역시 행렬단위로 이루어질 수밖에 없다. 이에 대해 자세히 살펴보자.

1.  $H^{(l)} = [h_1^{(l)}, \dots, h_{|V|}^{(l)}]^T$  : l번째 레이어의 모든 노드에 대한 벡터를 concat하여 행렬  $H^{(l)}$ 로 나타낼 수 있다. 이때 shape은 (노드, 차원)이 될 것이다.
2.  $\sum_{u \in N_v} h_u^{(l)} = A_{v,:} H^{(l)}$  :  $H^{(l)}$ 을 이용하면  $v$ 노드의 모든 이웃 노드의 벡터를 쉽게 더할 수 있다. 식에 보듯이, 인접행렬에서  $v$ 행은 노드  $v$ 의 이웃 노드에서만 1로 표시된 one hot vector이고, 이때의 벡터와 행렬의 연산은 결국 (1, 노드) x (노드, 차원) = (1, 차원)으로 모든 이웃 노드의 벡터의 합이 될 것이다.
3.  $D_{v,v} = \text{Deg}(v) = |N(v)|$  : D 행렬을 대각행렬이고, 각 대각 성분은  $v$ 노드의 이웃노드의 수를 가지고 있다고 하자. 이를 이용하면 D 행렬의 역행렬은 다음과 같은 대각성분을 가지게 된다.

$$D_{v,v}^{-1} = \frac{1}{|N(v)|}$$

즉,  $v$  노드의 이웃노드의 수의 역수가 된다.

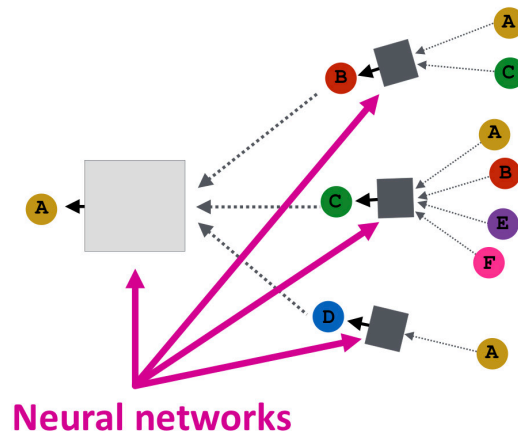
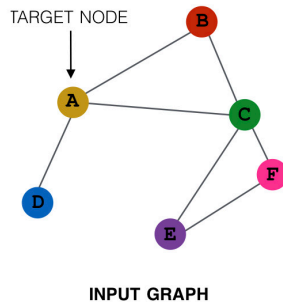
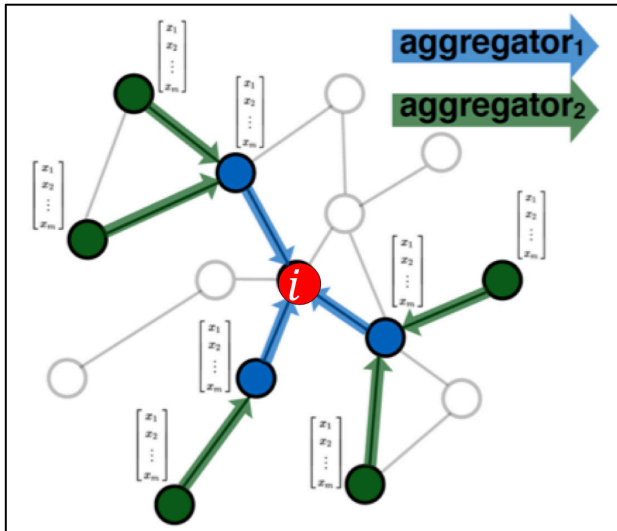
위 세가지 요소를 조합하면 이웃노드의 평균을 구하는 과정을 아래와 같이 행렬로 표현할 수 있다.

0. Recap  
1. Deep Graph Encoders  
1-1. Differences  
2. Background  
2-1. Notation  
2-2. DNN  
2-3. Convolutional Networks  
3. Graph Convolutional Networks  
3-1. Steps  
3-2. Neighborhood Aggregation  
3-3. Matrix Formulation  
3-4. Training  
3-4-1. 비지도학습  
3-4-2. 지도학습  
3-5. Model Design

## 스터디 내용 공유: The Over-smoothing Problem in GNN

### GNN: Aggregate from Neighbors

- Propagate and transform information

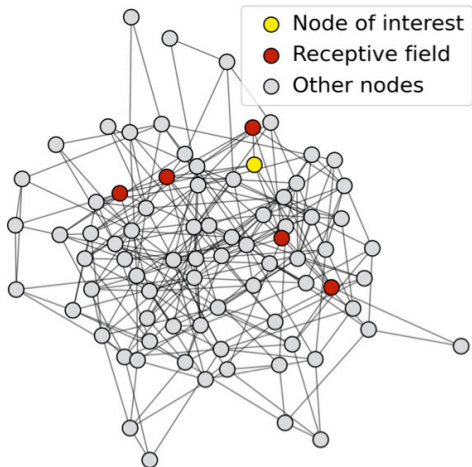


## 스터디 내용 공유: The Over-smoothing Problem in GNN

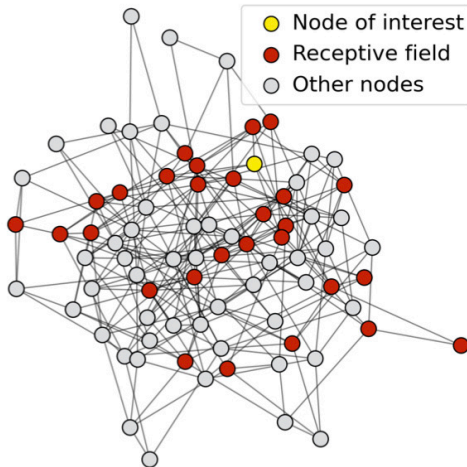
### The Issue of stacking many GNN layers

- Receptive fields: 특정한 Node의 embedding을 결정하는 Node의 set

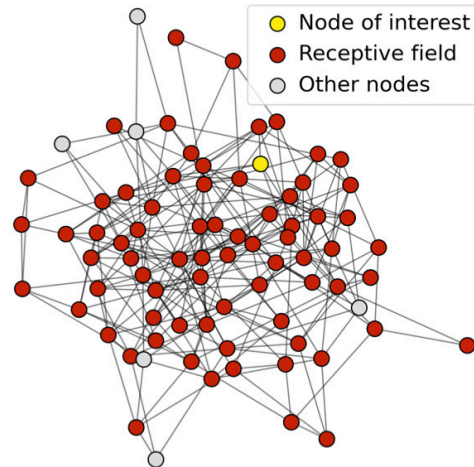
Receptive field for  
**1-layer GNN**



Receptive field for  
**2-layer GNN**



Receptive field for  
**3-layer GNN**



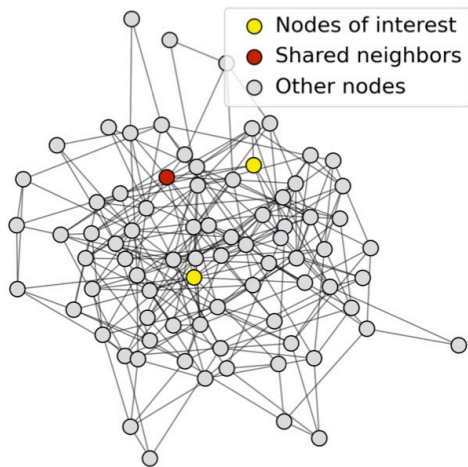
## 스터디 내용 공유: The Over-smoothing Problem in GNN

### The Issue of stacking many GNN layers

- Layer를 많이 통과할수록 Node의 receptive fields가 겹치게 됨

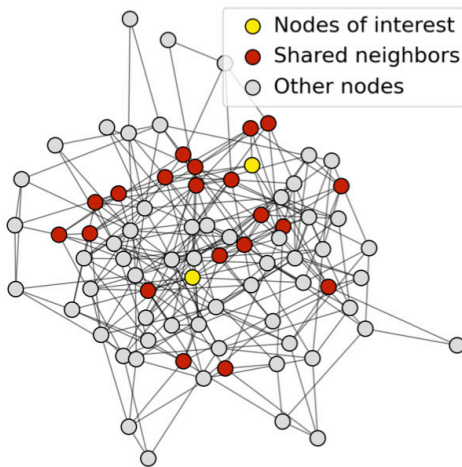
#### 1-hop neighbor overlap

Only 1 node



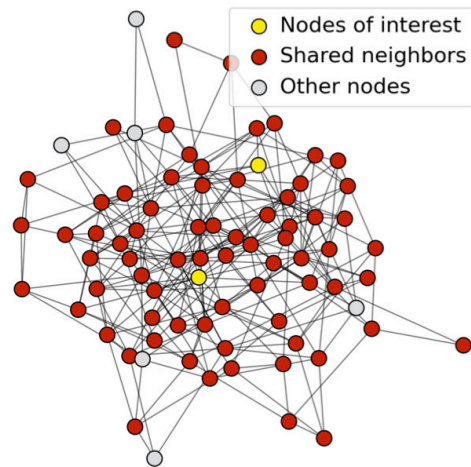
#### 2-hop neighbor overlap

About 20 nodes



#### 3-hop neighbor overlap

Almost all the nodes!

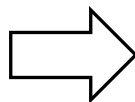
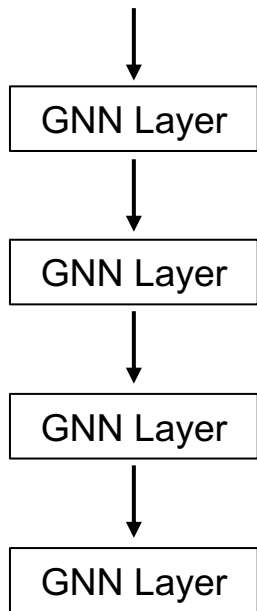




# 스터디 내용 공유: The Over-smoothing Problem in GNN

## The Issue of stacking many GNN layers

- The Over-smoothing Problem: 많은 layers를 쌓으면서 node의 embedding vector가 **같은 값**으로 수렴하는 문제



어떻게 over-smoothing problem을 해결할 수 있을까?

1. 각 layer의 expressive power 높이기
2. Message를 pass하지 않는 layers 추가
3. Skip connection 추가

그 전에 풀고자 하는 문제에 대해 receptive field가 어느 정도가 적당한지 먼저 분석하는 것도 방법!



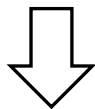
## 스터디 내용 공유: The Over-smoothing Problem in GNN

### Overcoming over-smoothing problem

1. 각 layer의 expressive power 높이기

$$\mathbf{h}_v^{(l)} = \sigma \left( \boxed{\mathbf{w}^{(l)}} \sum_{u \in N(v)} \frac{\mathbf{h}_u^{(l-1)}}{|N(v)|} \right)$$

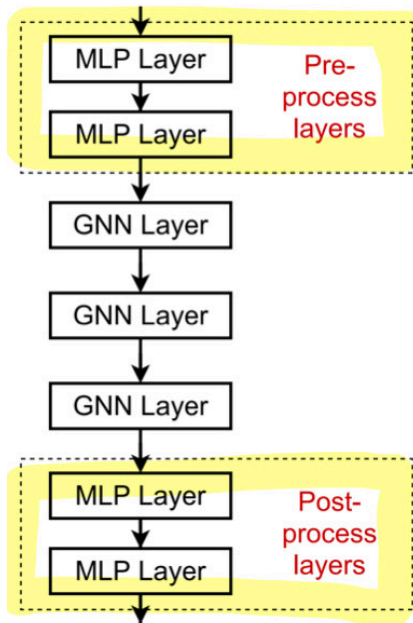
One layer perceptron



MLP로 변경

\* Universal approximation theorem

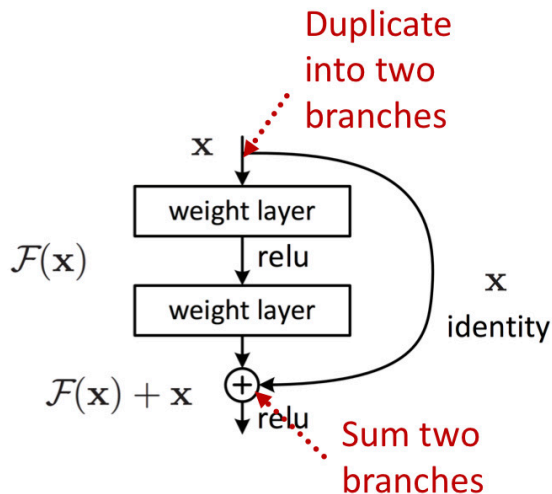
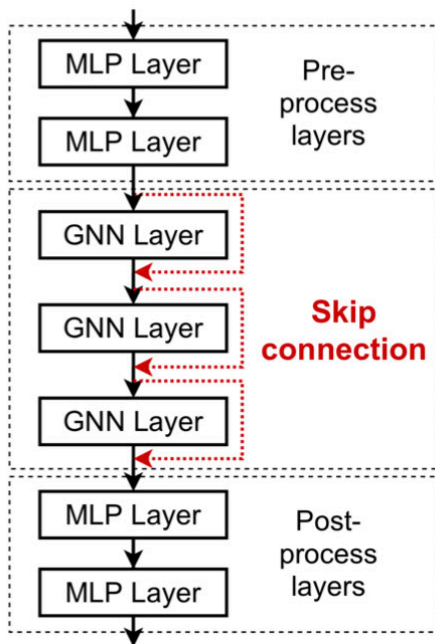
2. Message를 pass하지 않는 layers 추가



## 스터디 내용 공유: The Over-smoothing Problem in GNN

### Overcoming over-smoothing problem

#### 3. Skip connection 추가



**Idea of skip connections:**

Before adding shortcuts:

$$F(x)$$

After adding shortcuts:

$$F(x) + x$$

**THOHOI**

**THANK YOU  
FOR YOUR  
ATTENTION**