SEMI-SUPERVISED CLASSIFICATION WITH GRAPH CONVOLUTIONAL NETWORKS

By Thomas N. Kipf, Max Welling

Presented by Kim Han

Contents

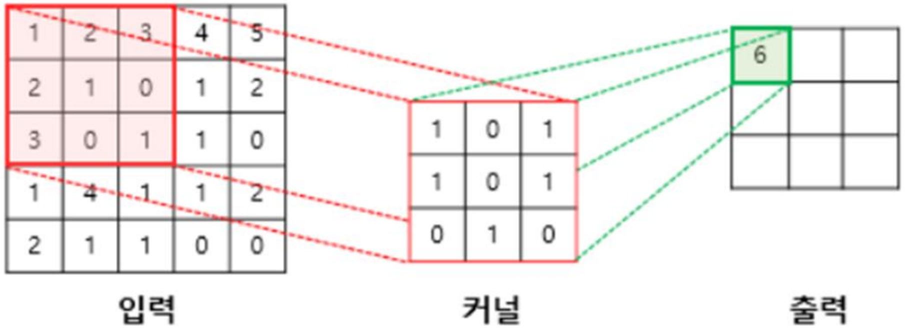
Background

Graph Neural Network(GNN)

- Recurrent Graph Neural Network
- Spatial Convolution Network
- Spectral Convolution Network

Background

Convolution

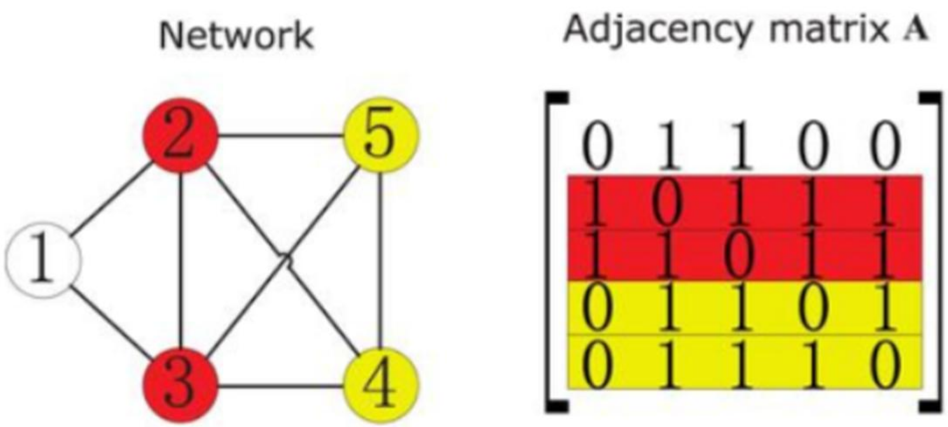


1. Weight Sharing
   - fixed size, reduce parameter

2. Learn Local Feature

Background

Convolution on Graph



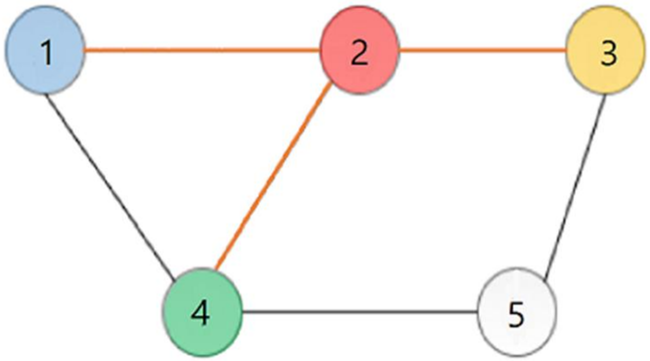
기존의 그래프는 Grid form이 아니기 때문에 일정한 size의 필터를 유지하며 정보를 처리하기 어려움

~Adjacency matrix(Graph의 Grid화)

Background

Convolution on Graph

Update hidden states



$$H_2^{(l+1)} = \sigma\left(H_1^{(l)}W^{(l)} + H_2^{(l)}W^{(l)} + H_3^{(l)}W^{(l)} + H_4^{(l)}W^{(l)} + b^{(l)}\right)$$

$$H_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} H_j^{(l)}W^{(l)} + b^{(l)}\right)$$

$$\begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

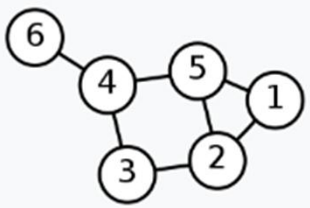
Use Adjacency matrix

Background

Convolution on Graph

$$H^{(l+1)} = \sigma\left(AH^{(l)}W^{(l)} + b^{(l)}\right)$$

Laplacian Matrix

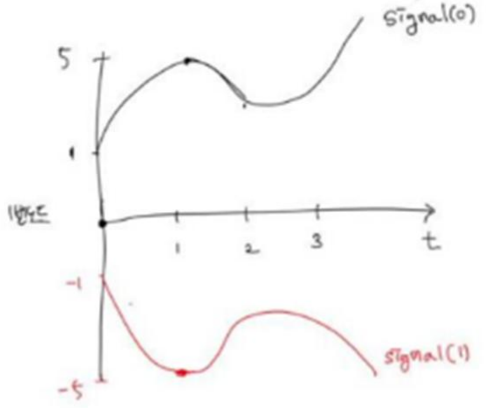
Labelled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2&0&0&0&0&0\\ 0&3&0&0&0&0\\ 0&0&2&0&0&0\\ 0&0&0&3&0&0\\ 0&0&0&0&3&0\\ 0&0&0&0&0&1 \end{pmatrix}$	$\begin{pmatrix} 0&1&0&0&1&0\\ 1&0&1&0&1&0\\ 0&1&0&1&0&0\\ 0&0&1&0&1&1\\ 1&1&0&1&0&0\\ 0&0&0&1&0&0 \end{pmatrix}$	$\begin{pmatrix}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

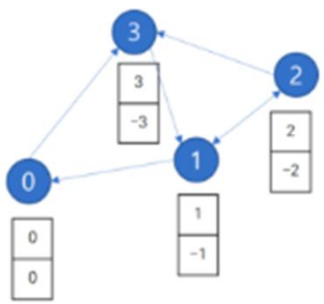
$L = D - A$ (Laplacian Matrix)

$L^{\text{sym}} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ (Symmetric normalized Laplacian)

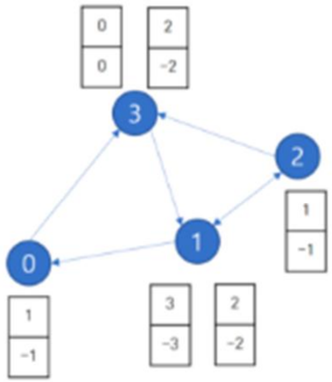
Background

Spectral Network

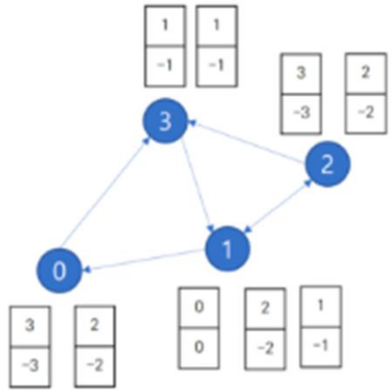




t = 0



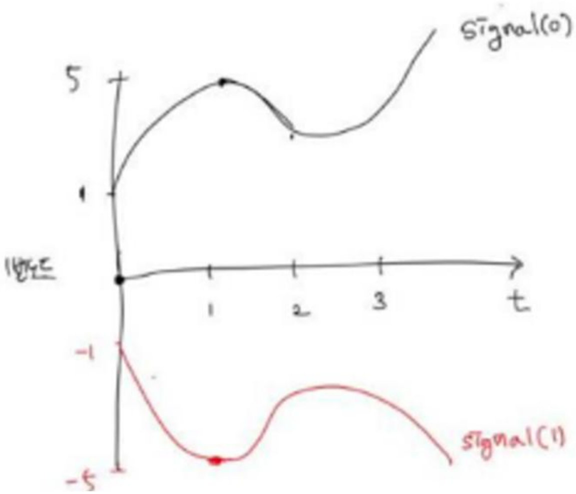
t = 1



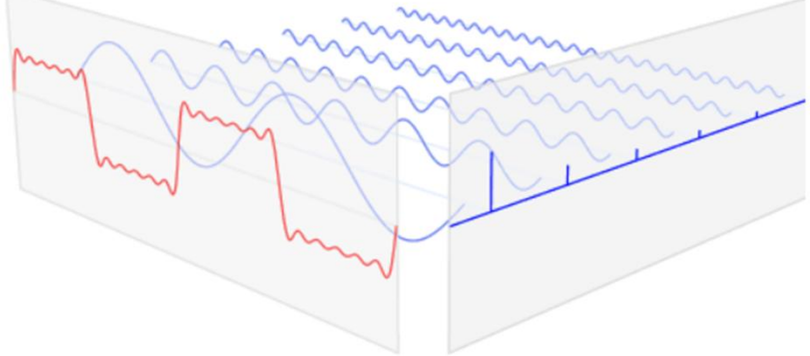
t = 2

Background

Spectral Network(Fourier Transform)



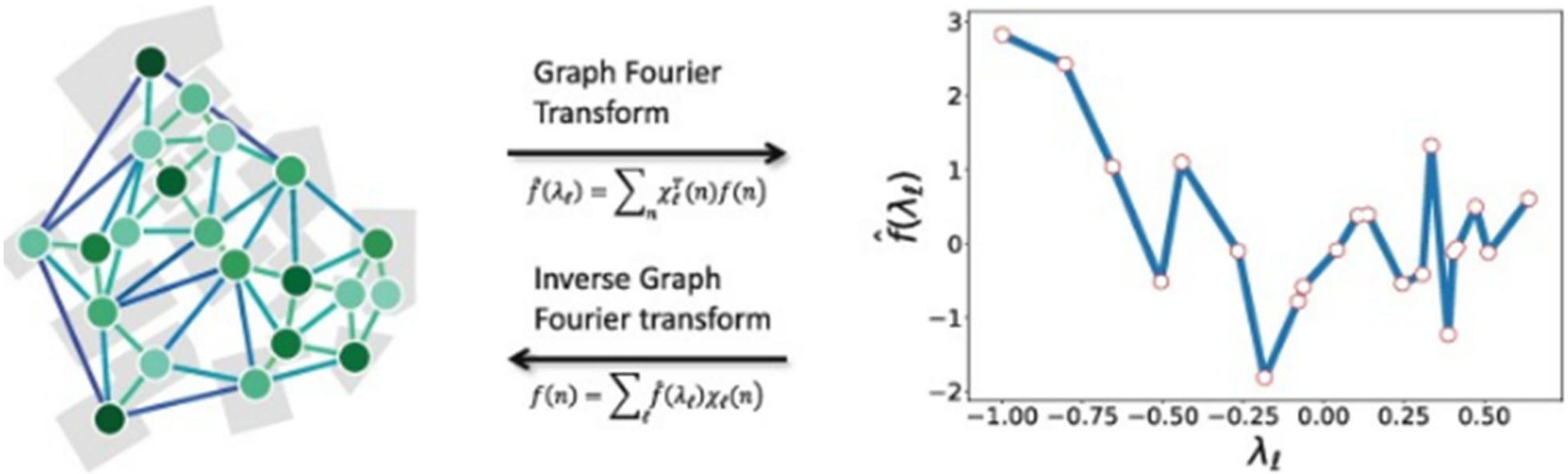
Time domain



Frequency domain

Background

Spectral Network(Fourier Transform)



Graph domain

Frequency domain

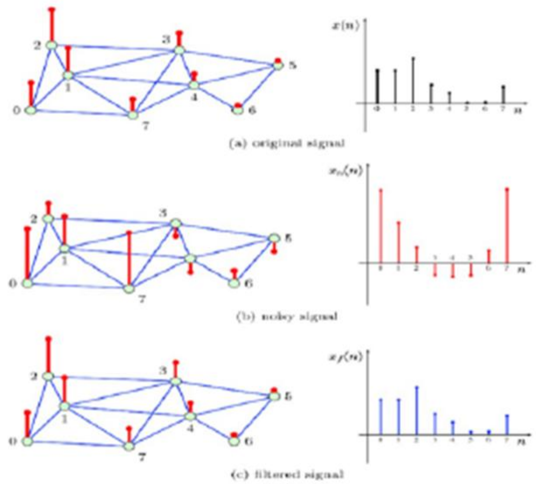
Background

Spectral Network(Fourier Transform)

Orthonormal Basis
- Orthogonal basis
- Diagonalization
- Real-symmetric matrix
- Eigenvector

-> (Normalized) Laplacian Matrix

Graph Fourier Transform



Graph Signal (Node Label) → Frequency (Difference on Central, neighbor node)

Eigenvalue of Laplacian : Frequency
Eigenvectors of Laplacian : Fourier basis

Background

요약

1. Adjacency Matrix를 통해 Graph에도 Convolution을 적용할 수 있다.
   - Laplacian Matrix

2. Laplacian Matrix의 Eigen-decomposition을 이용하면 Graph를 Frequency domain으로 변환할 수 있다.

Introduction

Goal

1. Given node feature vectors X_i,
2. Given Adjacency matrix A,
3. Only labeled for small subset of nodes.

With setting f(.), Semi-Supervised Classification With Graph Convolutional Networks
- simple and well-behaved layer-wise propagation rule
- graph-based neural network model for semi-supervised classification

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

Introduction

Spectral Convolution on Graphs

$$g_\theta \star x = U g_\theta U^\top x$$

- Eigen Decomposition for Laplacian

filter $g_\theta = \text{diag}(\theta)$ parameterized by $\theta \in \mathbb{R}^N$

$$L = I_N - D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = U \Lambda U^T$$

- U = matrix of eigenvectors of the Laplacian L

Rescaling with,,

$$\tilde{\Lambda} = \frac{2}{\lambda_{\max}} \Lambda - I_N$$

$\lambda_{\max}$ denotes the largest eigenvalue of L

θ': Chebyshev coefficients

Recursively defined as

$T_k(x) = 2xT_{k-1} - T_{k-2}(x)$

$T_0(x) = 1, T_1 = x$

$$g_{\theta'}(\Lambda) \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{\Lambda})$$

Introduction

Spectral Convolution on Graphs (Approximation)

$$g_{\theta'} \star x \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{L})x$$

K-localized (Kth-order neighborhood)

$$g_{\theta'} \star x \approx \theta'_0 x + \theta'_1 \left(L - I_N\right) x = \theta'_0 x - \theta'_1 D^{-\frac{1}{2}} A D^{-\frac{1}{2}} x$$

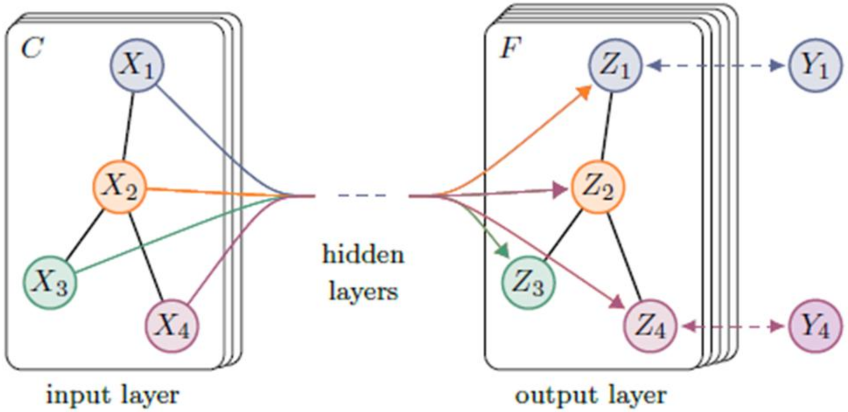
Renormalization Trick!

$$I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \rightarrow \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$$
$$\tilde{A} = A + I_N \text{ and } \tilde{D}_{ii} = \textstyle\sum_j \tilde{A}_{ij}.$$

$$Z = \tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} X \Theta$$

Introduction

Semi-Supervised Node Classification



(a) Graph Convolutional Network

(b) Hidden layer activations

$$Z = f(X, A) = \mathrm{softmax}\left(\hat{A}\ \mathrm{ReLU}\left(\hat{A} X W^{(0)}\right) W^{(1)}\right)$$

$$\mathcal{L} = \mathcal{L}_0 + \lambda \mathcal{L}_{\mathrm{reg}} \Rightarrow \mathcal{L} = -\sum_{l \in \mathcal{Y}_L} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf}$$

Experiment

Table 1: Dataset statistics, as reported in Yang et al. (2016).

Dataset	Type	Nodes	Edges	Classes	Features	Label rate
Citeseer	Citation network	3,327	4,732	6	3,703	0.036
Cora	Citation network	2,708	5,429	7	1,433	0.052
Pubmed	Citation network	19,717	44,338	3	500	0.003
NELL	Knowledge graph	65,755	266,144	210	5,414	0.001

Table 2: Summary of results in terms of classification accuracy (in percent).

Method	Citeseer	Cora	Pubmed	NELL
ManiReg [3]	60.1	59.5	70.7	21.8
SemiEmb [28]	59.6	59.0	71.1	26.7
LP [32]	45.3	68.0	63.0	26.5
DeepWalk [22]	43.2	67.2	65.3	58.1
ICA [18]	69.1	75.1	73.9	23.1
Planetoid* [29]	64.7 (26s)	75.7 (13s)	77.2 (25s)	61.9 (185s)
GCN (this paper)	**70.3** (7s)	**81.5** (4s)	**79.0** (38s)	**66.0** (48s)
GCN (rand. splits)	67.9 ± 0.5	80.1 ± 0.5	78.9 ± 0.7	58.4 ± 1.7

Experiment

Table 3: Comparison of propagation models.

Description		Propagation model	Citeseer	Cora	Pubmed
Chebyshev filter (Eq. 5)	$K=3$	$\sum_{k=0}^{K} T_k(\tilde{L})X\Theta_k$	69.8	79.5	74.4
	$K=2$		69.6	81.2	73.8
1st-order model (Eq. 6)		$X\Theta_0 + D^{-\frac{1}{2}}AD^{-\frac{1}{2}}X\Theta_1$	68.3	80.0	77.5
Single parameter (Eq. 7)		$(I_N + D^{-\frac{1}{2}}AD^{-\frac{1}{2}})X\Theta$	69.3	79.2	77.4
Renormalization trick (Eq. 8)		$\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}X\Theta$	**70.3**	**81.5**	**79.0**
1st-order term only		$D^{-\frac{1}{2}}AD^{-\frac{1}{2}}X\Theta$	68.7	80.5	77.8
Multi-layer perceptron		$X\Theta$	46.5	55.1	71.4

Implementation

```
Downloading /root/.dgl/citeseer.zip from https://data.dgl.ai/dataset/citeseer.zip...
Extracting file to /root/.dgl/citeseer
Finished data loading and preprocessing.
  NumNodes: 3327
  NumEdges: 9228
  NumFeats: 3703
  NumClasses: 6
  NumTrainingSamples: 120
  NumValidationSamples: 500
  NumTestSamples: 1000
Done saving data into cached files.
EPOCH 1 : TRAINING loss 1.871, TRAINING ACC 0.192, VALID loss 1.823, VALID ACC 0.231
EPOCH 10 : TRAINING loss 1.786, TRAINING ACC 0.192, VALID loss 1.803, VALID ACC 0.178
EPOCH 20 : TRAINING loss 1.725, TRAINING ACC 0.367, VALID loss 1.776, VALID ACC 0.300
EPOCH 30 : TRAINING loss 1.708, TRAINING ACC 0.500, VALID loss 1.752, VALID ACC 0.388
EPOCH 40 : TRAINING loss 1.643, TRAINING ACC 0.733, VALID loss 1.732, VALID ACC 0.510
EPOCH 50 : TRAINING loss 1.578, TRAINING ACC 0.783, VALID loss 1.695, VALID ACC 0.585
EPOCH 60 : TRAINING loss 1.476, TRAINING ACC 0.842, VALID loss 1.648, VALID ACC 0.637
EPOCH 70 : TRAINING loss 1.343, TRAINING ACC 0.892, VALID loss 1.585, VALID ACC 0.658
EPOCH 80 : TRAINING loss 1.237, TRAINING ACC 0.833, VALID loss 1.509, VALID ACC 0.664
EPOCH 90 : TRAINING loss 1.087, TRAINING ACC 0.867, VALID loss 1.433, VALID ACC 0.682
EPOCH 100 : TRAINING loss 0.942, TRAINING ACC 0.867, VALID loss 1.359, VALID ACC 0.693
EPOCH 110 : TRAINING loss 0.847, TRAINING ACC 0.900, VALID loss 1.301, VALID ACC 0.695
EPOCH 120 : TRAINING loss 0.738, TRAINING ACC 0.925, VALID loss 1.253, VALID ACC 0.694
EPOCH 130 : TRAINING loss 0.713, TRAINING ACC 0.942, VALID loss 1.206, VALID ACC 0.697
EPOCH 140 : TRAINING loss 0.618, TRAINING ACC 0.917, VALID loss 1.177, VALID ACC 0.691
EPOCH 150 : TRAINING loss 0.580, TRAINING ACC 0.925, VALID loss 1.142, VALID ACC 0.706
EPOCH 160 : TRAINING loss 0.548, TRAINING ACC 0.958, VALID loss 1.123, VALID ACC 0.709
EPOCH 170 : TRAINING loss 0.505, TRAINING ACC 0.958, VALID loss 1.099, VALID ACC 0.712
EPOCH 180 : TRAINING loss 0.486, TRAINING ACC 0.975, VALID loss 1.087, VALID ACC 0.704
EPOCH 190 : TRAINING loss 0.468, TRAINING ACC 0.942, VALID loss 1.079, VALID ACC 0.700
EPOCH 200 : TRAINING loss 0.441, TRAINING ACC 0.958, VALID loss 1.059, VALID ACC 0.710
0:00:46.771047
At EPOCH 185, We have Best Acc 0.7240000367164612
```

```
Finished data loading and preprocessing.
  NumNodes: 2708
  NumEdges: 10556
  NumFeats: 1433
  NumClasses: 7
  NumTrainingSamples: 140
  NumValidationSamples: 500
  NumTestSamples: 1000
Done saving data into cached files.
EPOCH 1 : TRAINING loss 2.327, TRAINING ACC 0.136, VALID loss 2.266, VALID ACC 0.091
EPOCH 10 : TRAINING loss 1.951, TRAINING ACC 0.193, VALID loss 1.948, VALID ACC 0.136
EPOCH 20 : TRAINING loss 1.906, TRAINING ACC 0.207, VALID loss 1.925, VALID ACC 0.142
EPOCH 30 : TRAINING loss 1.821, TRAINING ACC 0.507, VALID loss 1.883, VALID ACC 0.336
EPOCH 40 : TRAINING loss 1.764, TRAINING ACC 0.771, VALID loss 1.847, VALID ACC 0.680
EPOCH 50 : TRAINING loss 1.695, TRAINING ACC 0.757, VALID loss 1.799, VALID ACC 0.661
EPOCH 60 : TRAINING loss 1.574, TRAINING ACC 0.836, VALID loss 1.733, VALID ACC 0.731
EPOCH 70 : TRAINING loss 1.451, TRAINING ACC 0.814, VALID loss 1.642, VALID ACC 0.765
EPOCH 80 : TRAINING loss 1.284, TRAINING ACC 0.886, VALID loss 1.536, VALID ACC 0.789
EPOCH 90 : TRAINING loss 1.123, TRAINING ACC 0.914, VALID loss 1.423, VALID ACC 0.792
EPOCH 100 : TRAINING loss 1.015, TRAINING ACC 0.914, VALID loss 1.320, VALID ACC 0.809
EPOCH 110 : TRAINING loss 0.892, TRAINING ACC 0.900, VALID loss 1.232, VALID ACC 0.813
EPOCH 120 : TRAINING loss 0.785, TRAINING ACC 0.907, VALID loss 1.151, VALID ACC 0.817
EPOCH 130 : TRAINING loss 0.745, TRAINING ACC 0.914, VALID loss 1.081, VALID ACC 0.818
EPOCH 140 : TRAINING loss 0.607, TRAINING ACC 0.950, VALID loss 1.037, VALID ACC 0.817
EPOCH 150 : TRAINING loss 0.617, TRAINING ACC 0.943, VALID loss 0.993, VALID ACC 0.818
EPOCH 160 : TRAINING loss 0.518, TRAINING ACC 0.943, VALID loss 0.960, VALID ACC 0.819
EPOCH 170 : TRAINING loss 0.530, TRAINING ACC 0.971, VALID loss 0.927, VALID ACC 0.819
EPOCH 180 : TRAINING loss 0.507, TRAINING ACC 0.957, VALID loss 0.907, VALID ACC 0.819
EPOCH 190 : TRAINING loss 0.434, TRAINING ACC 0.964, VALID loss 0.897, VALID ACC 0.806
EPOCH 200 : TRAINING loss 0.397, TRAINING ACC 0.957, VALID loss 0.870, VALID ACC 0.808
0:00:23.634308
At EPOCH 164, We have Best Acc 0.8220000267028809
```

Limitation

1. Memory requirement (Full-batch gradient decent)

2. Directed edges and edge features (Undirected graphs)

3. Limiting assumptions(Equal importance)

Conclusion

1. Convolution on Graph

2. First-order approximation을 통해 효율적인 Local Feature

3. GCN을 통해 다른 모델보다 뛰어난 성능으로 Semi-supervised classification을 진행할 수 있음

감사합니다.