Graph convolutional neural networks

T.N. Kipf & M. Welling, ICLR *conf.*, 2017.

MODNBIO

COPYRIGHT ⓒ 2020 MODNBIO, Inc. ALL RIGHTS RESERVED

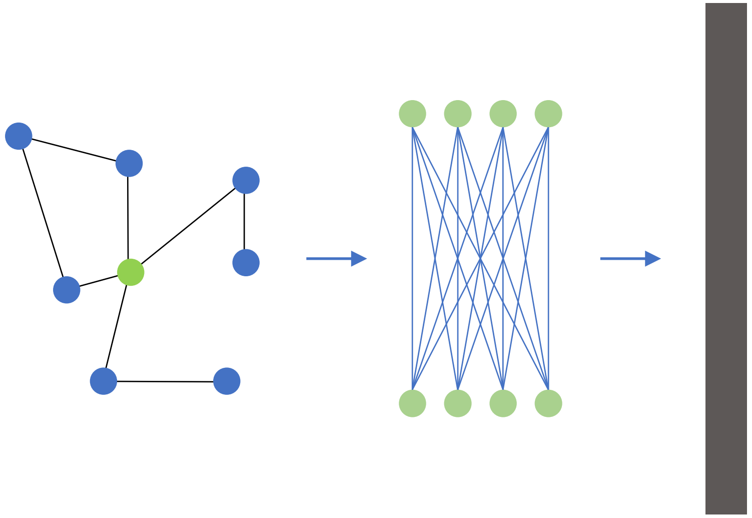
Outline

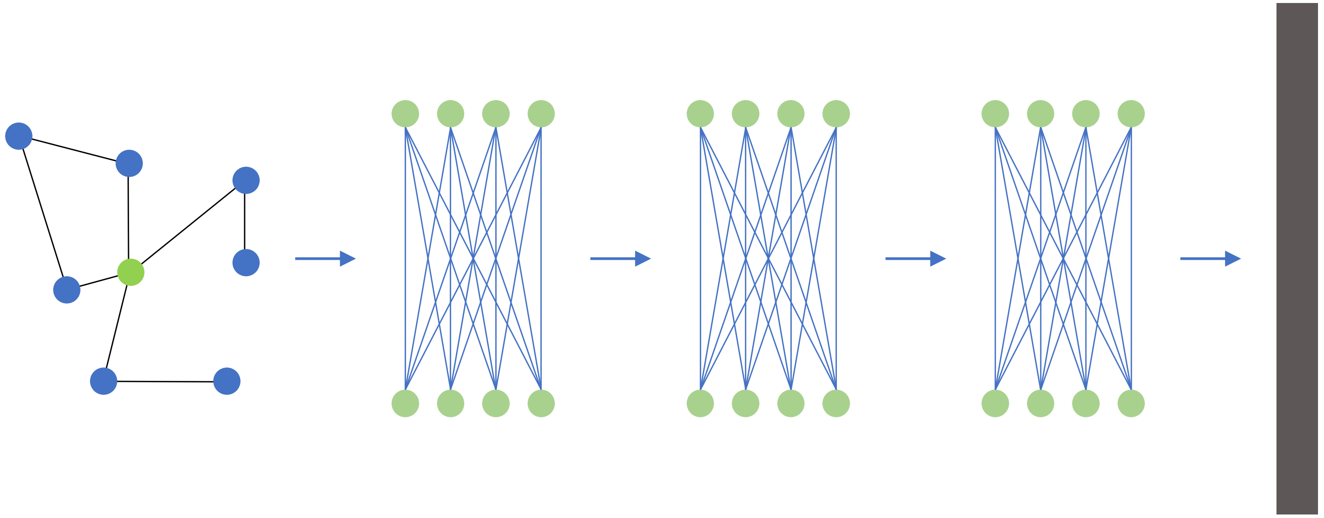
- GCFP에 대비한 Graph convolutional neural network의 개선점
- Fast approximate convolutions on graphs (algorithm)
- Semi-supervised node classification
- Graph classification
- Result and Discussion

COPYRIGHT ⓒ 2020 MODNBIO, Inc. ALL RIGHTS RESERVED

GCFP에 대비한 graph convolutional network의 개선점

- GCFP에 비해 많은 수의 parameter를 포함할 수 있기 때문에 더 크고 복잡한 graph를 표현할 수 있다.
- 네트워크 layer를 쌓을(stack) 수 있기 때문에 더 깊은 (deeper) 모델을 구현할 수 있다.
- fixed length가 아닌 variable length를 사용하기 때문에 더 유연한 모델을 만들 수 있다.
- 거대한 네트워크에서 지역적인 이웃 구조에 의한 과적합(overfitting) 문제를 완화할 수 있다.

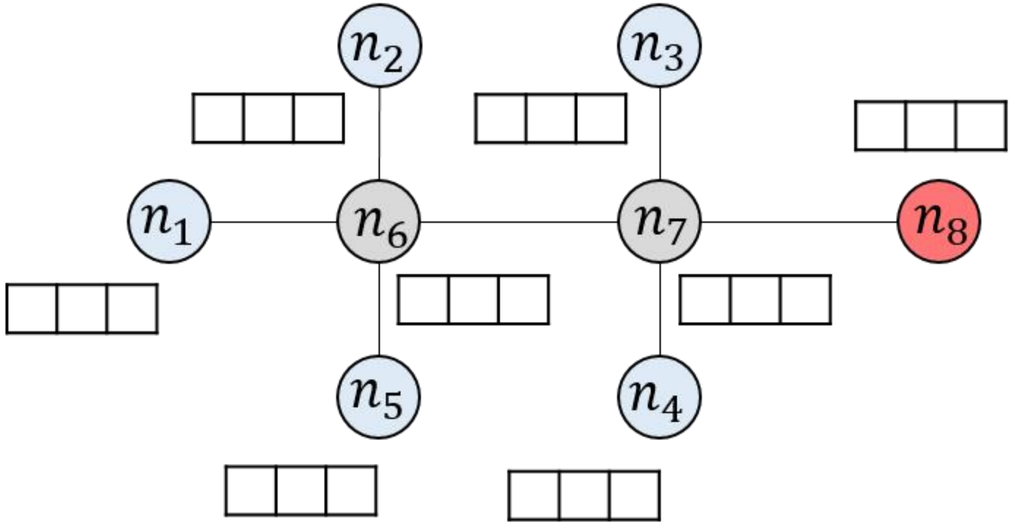




MODNBIO

COPYRIGHT ⓒ 2020 MODNBIO, Inc. ALL RIGHTS RESERVED

Fast approximate convolutions on graphs (algorithm)



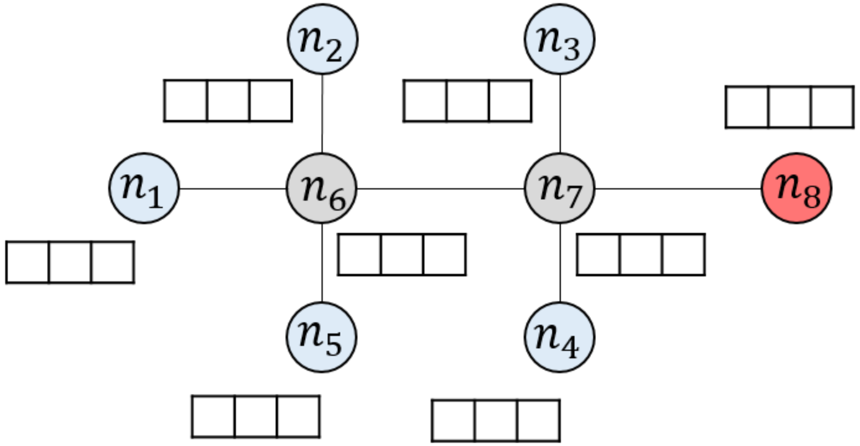
$$A = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

(8, 8)

adjacency matrix 변환

COPYRIGHT ⓒ 2020 MODNBIO, Inc. ALL RIGHTS RESERVED

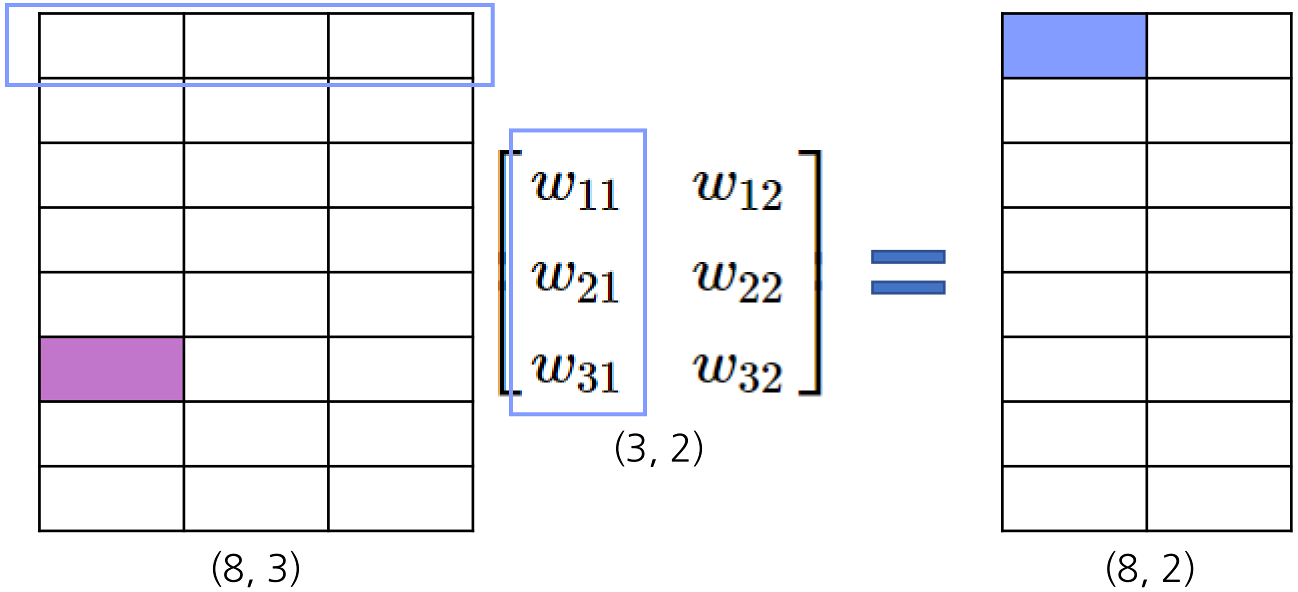
Fast approximate convolutions on graphs (algorithm)



$$S = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots \\ x_{81} & x_{82} & x_{83} \end{bmatrix}$$

$$\begin{bmatrix} 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&1&0&0 \\ 1&1&0&0&1&0&1&0 \\ 0&0&1&1&0&1&0&1 \\ 0&0&0&0&0&0&1&0 \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ \vdots & \vdots & \vdots \\ x_{81} & x_{82} & x_{83} \end{bmatrix}$$

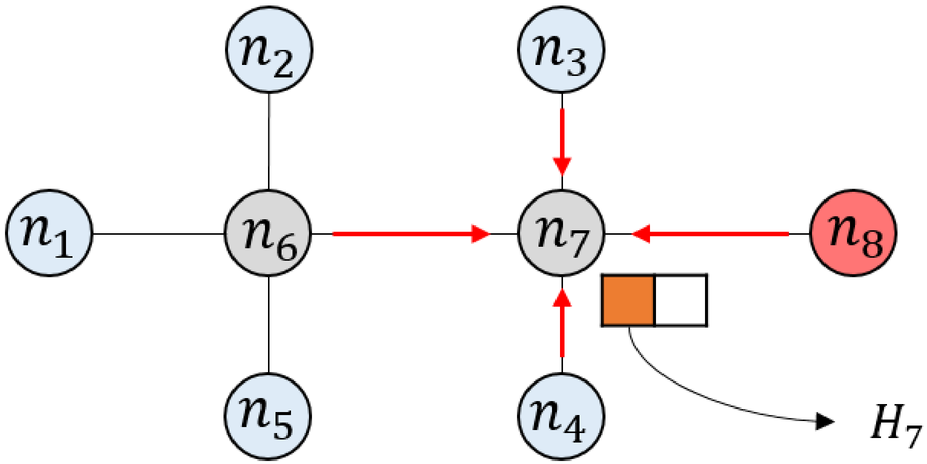
(8, 8) (8, 3)



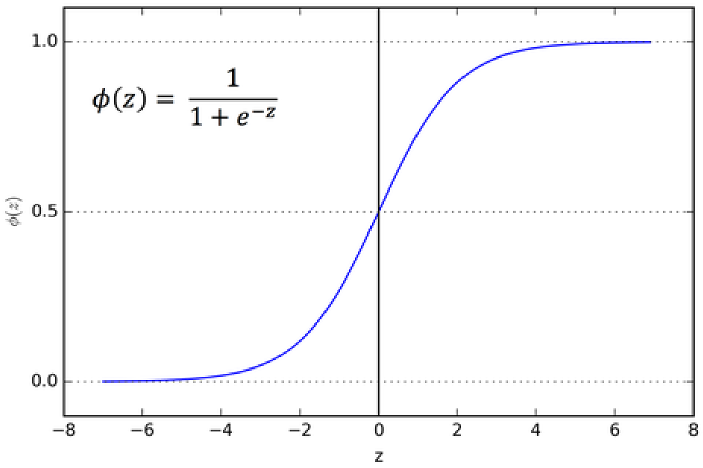
MODNBIO

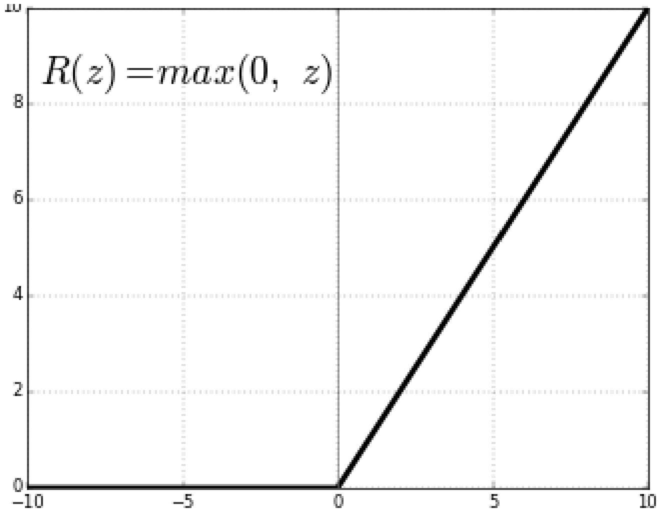
COPYRIGHT © 2020 MODNBIO, Inc. ALL RIGHTS RESERVED

Fast approximate convolutions on graphs (algorithm)



$$H_{71} = \boxed{\sigma}(s_{31} + s_{41} + s_{61} + s_{81})$$



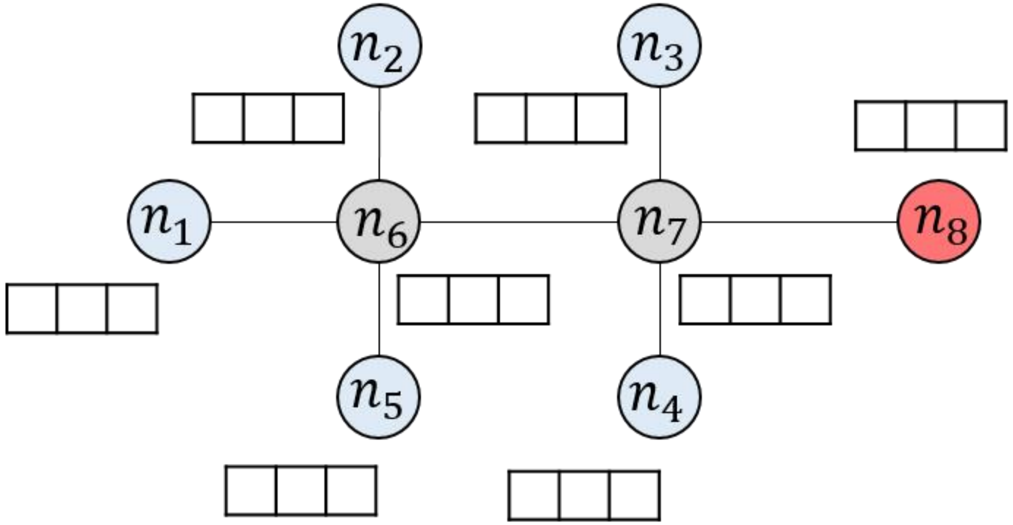


for non-linearity

MODNBIO

COPYRIGHT ⓒ 2020 MODNBIO, Inc. ALL RIGHTS RESERVED

Fast approximate convolutions on graphs (algorithm)



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

$$I = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad \tilde{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

$$\tilde{A} = A + I$$

$$\psi(A, X) = \sigma(AXW)$$

regularization term

$$\psi(\tilde{A}, X) = \sigma(\boxed{\tilde{D}^{-1/2}}\tilde{A}\boxed{\tilde{D}^{-1/2}}XW)$$

degree matrix

$$H^{(k)} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}H^{(k-1)}W^{(k)})$$

COPYRIGHT ⓒ 2020 MODNBIO, Inc. ALL RIGHTS RESERVED

Semi-supervised node classification

- Assume research citation problem (major of researcher)
- Assume some simple architecture for node classification
  - have just 1 hidden layer
  - no regularization term

prediction

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

- Loss function for semi-supervised learning

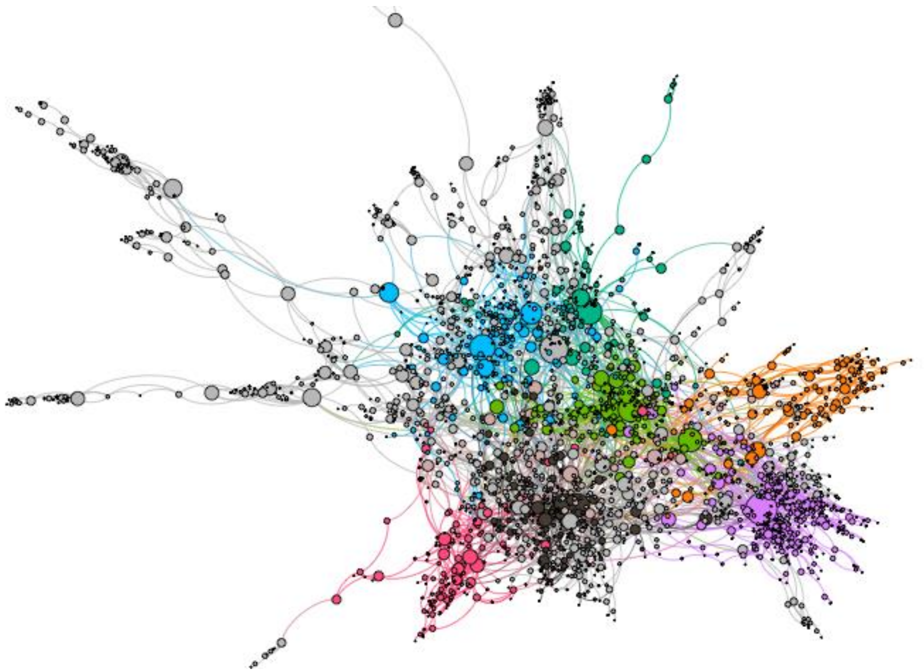
$$\mathcal{L} = -\sum_{\boxed{l \in \mathcal{Y}_L}} \sum_{f=1}^{F} Y_{lf} \ln Z_{lf} \quad \Rightarrow \quad \text{back-propagation}$$

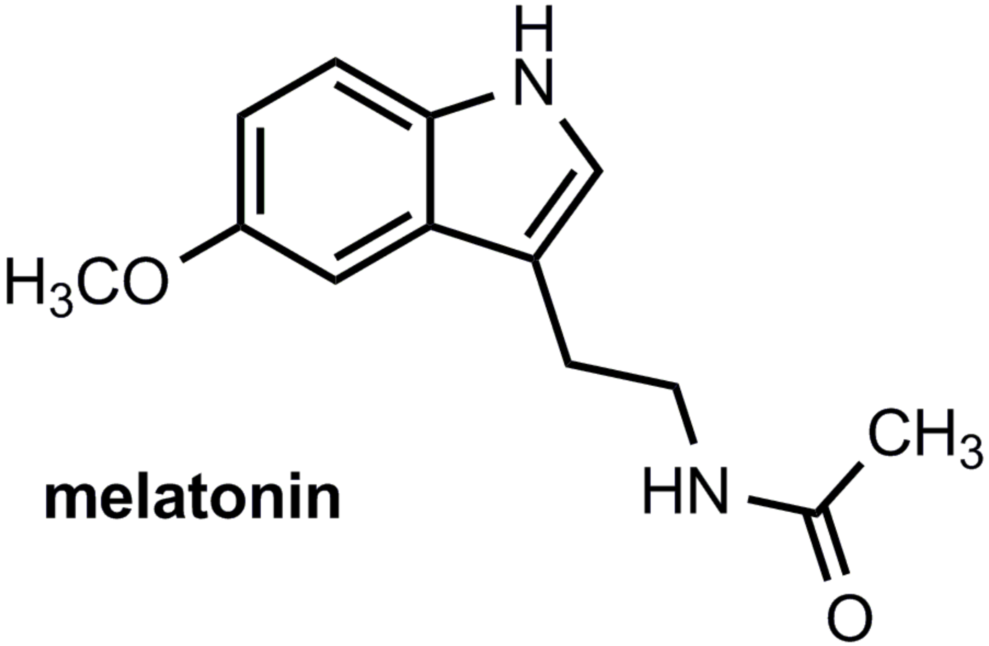
- label을 가지고 있는 이웃 노드들 모두!
- label 없는 node들 빼고!
- 이웃들은 같은 class에 속할 확률이 높음

$$\text{SUM}\left\{\begin{array}{l}
\begin{array}{ccccccc} & & f & & & & \\ Y = & [& 0 & 0 & 0 & 1 & 0 \,] \end{array} \\
Z = [\,0.2 \;\; 0.1 \;\; 0.1 \;\; 0.4 \;\; 0.2\,] \\
\\
Y = [\,0 \;\; 1 \;\; 0 \;\; 0 \;\; 0\,] \\
Z = [\,0.1 \;\; 0.5 \;\; 0.1 \;\; 0.1 \;\; 0.2\,] \\
\\
Y = [\,0 \;\; 0 \;\; 0 \;\; 1 \;\; 0\,] \\
Z = [\,0.3 \;\; 0.3 \;\; 0.1 \;\; 0.1 \;\; 0.2\,]
\end{array}\right\}$$

COPYRIGHT ⓒ 2020 MODNBIO, Inc. ALL RIGHTS RESERVED

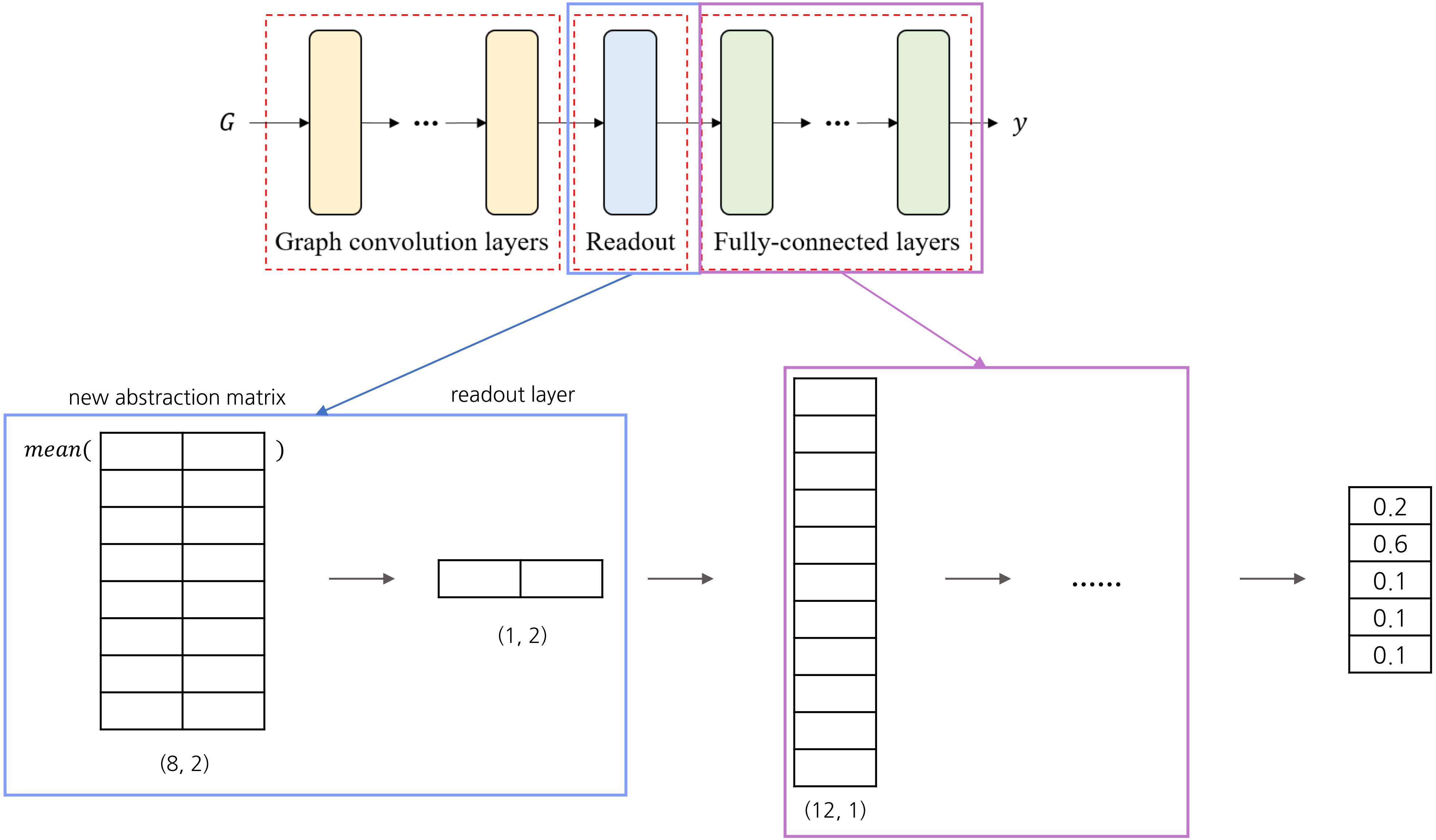
Graph classification





COPYRIGHT ⓒ 2020 MODNBIO, Inc. ALL RIGHTS RESERVED

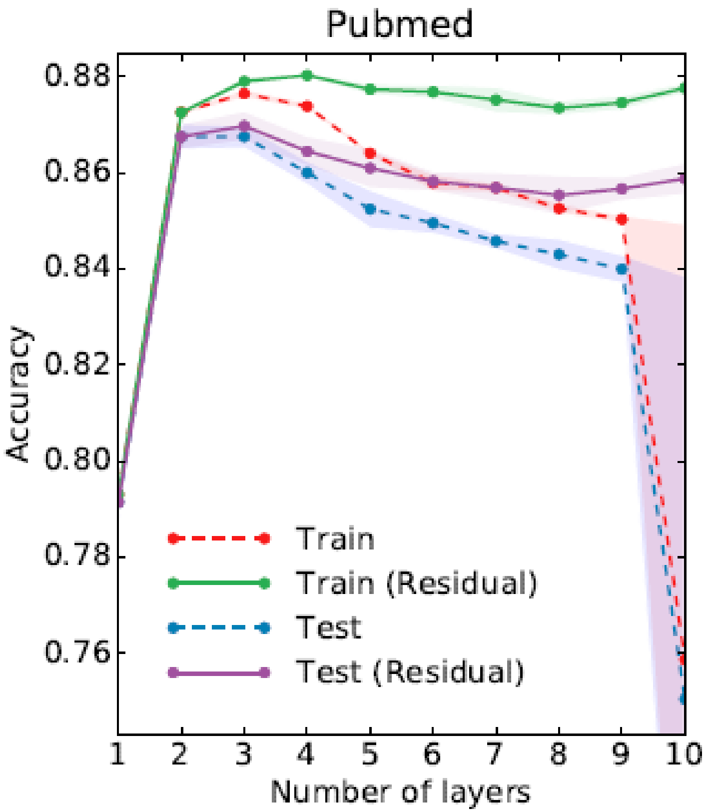
Graph classification



COPYRIGHT ⓒ 2020 MODNBIO, Inc. ALL RIGHTS RESERVED

Result and Discussion

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



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)		$X\Theta_0 + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} X \Theta_1$	68.3	80.0	77.5
Single parameter (Eq. 7)		$(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) X \Theta$	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		$D^{-\frac{1}{2}} A D^{-\frac{1}{2}} X \Theta$	68.7	80.5	77.8
Multi-layer perceptron		$X\Theta$	46.5	55.1	71.4

MODNBIO

COPYRIGHT ⓒ 2020 MODNBIO, Inc. ALL RIGHTS RESERVED