Mar. 21, 2024

Graph Neural Network

Sang Hoon Han

Department of AISW

Dongguk University

Contents

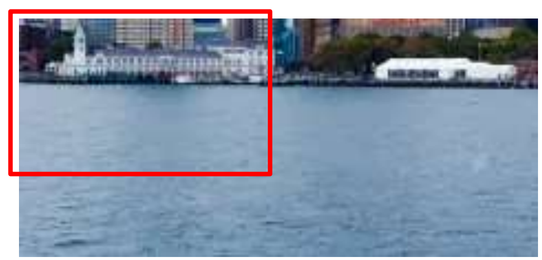
Introduction

◆ CNN(Convolution Neural Network)

* Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.

Introduction

◆ CNN(Convolution Neural Network)



* Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.

Introduction

- CNN(Convolution Neural Network)

* Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.

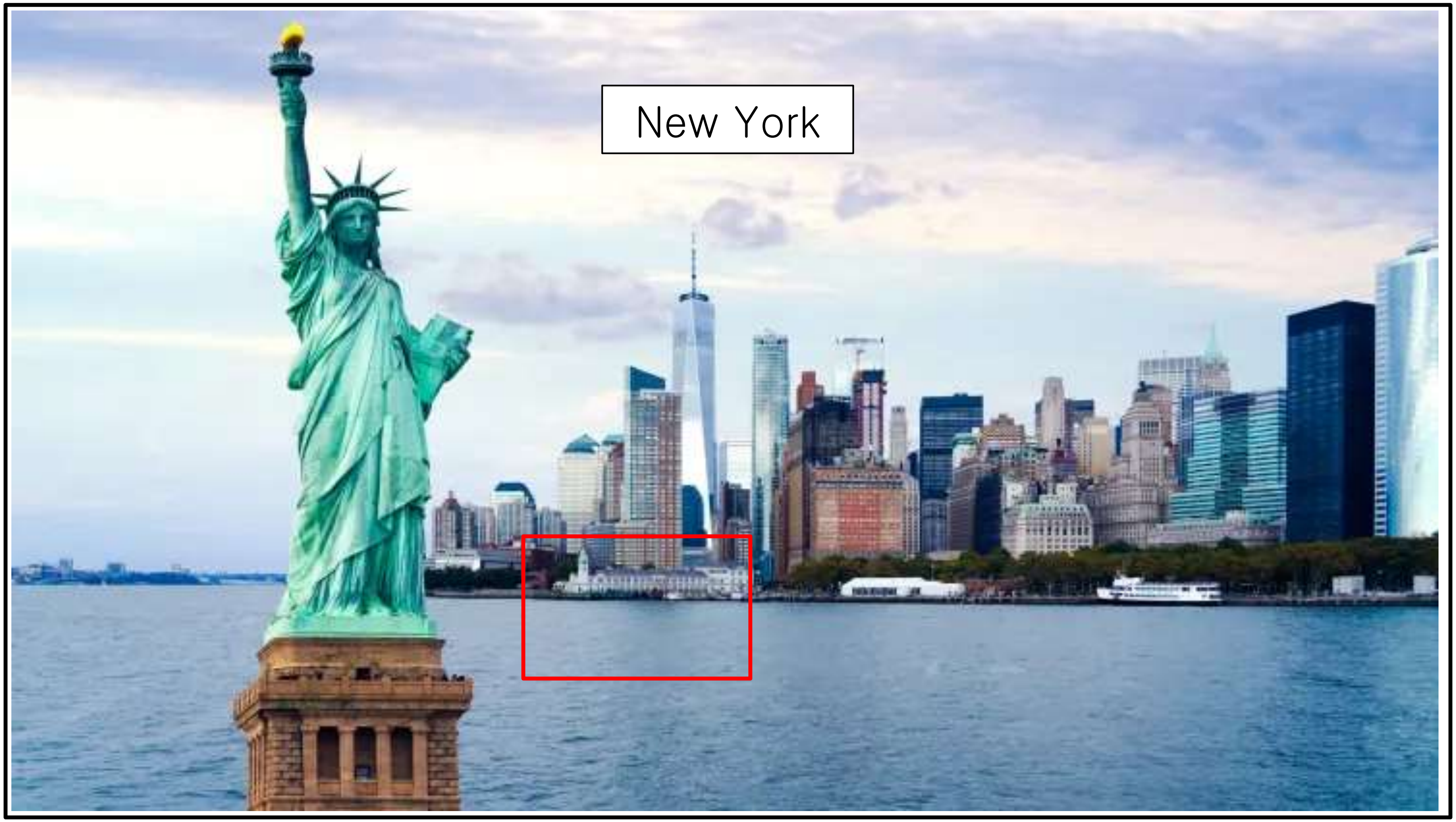
Introduction

◆ CNN(Convolution Neural Network)

* Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.

Introduction

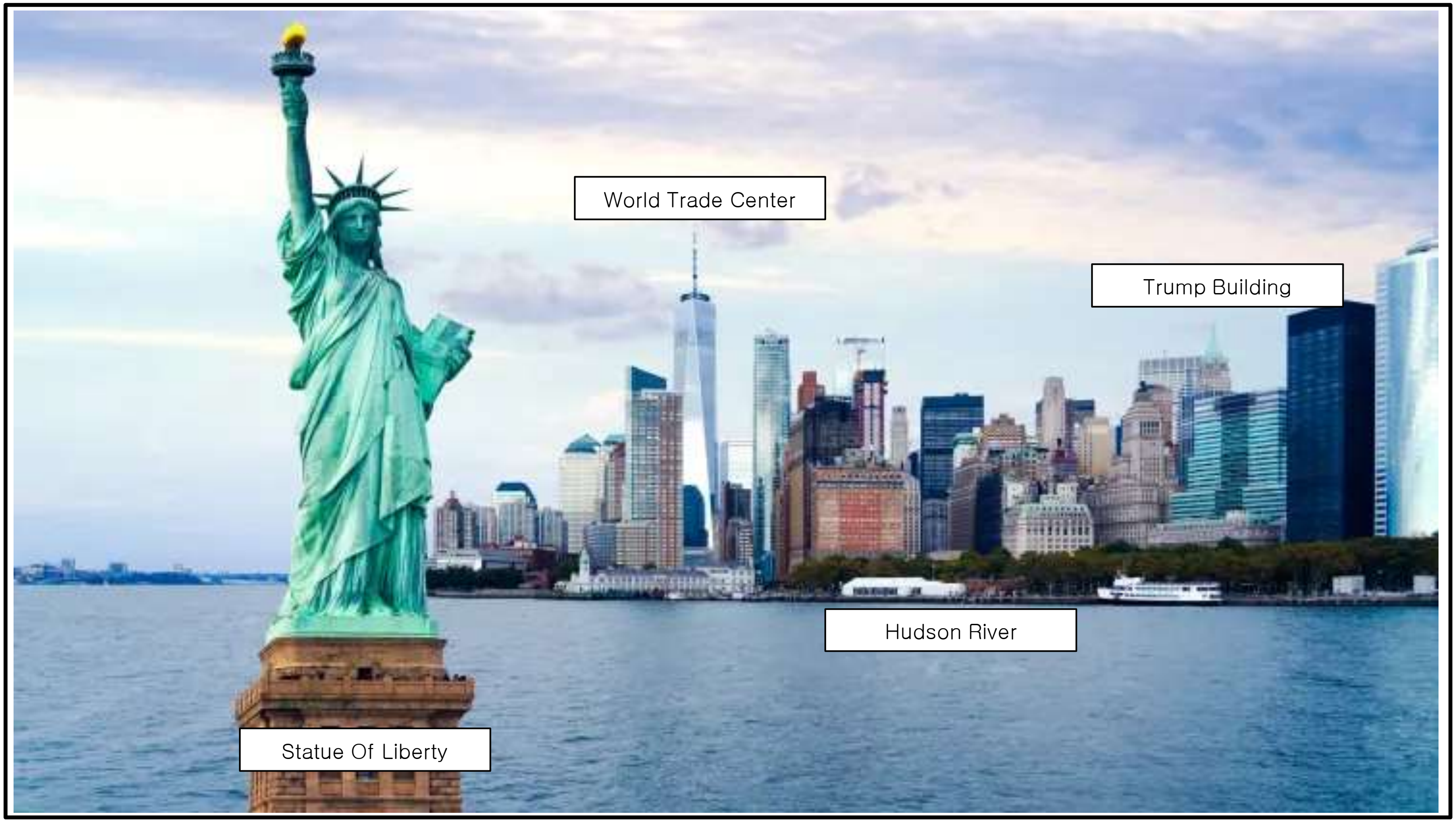
◆ CNN(Convolution Neural Network)



* Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.

Introduction

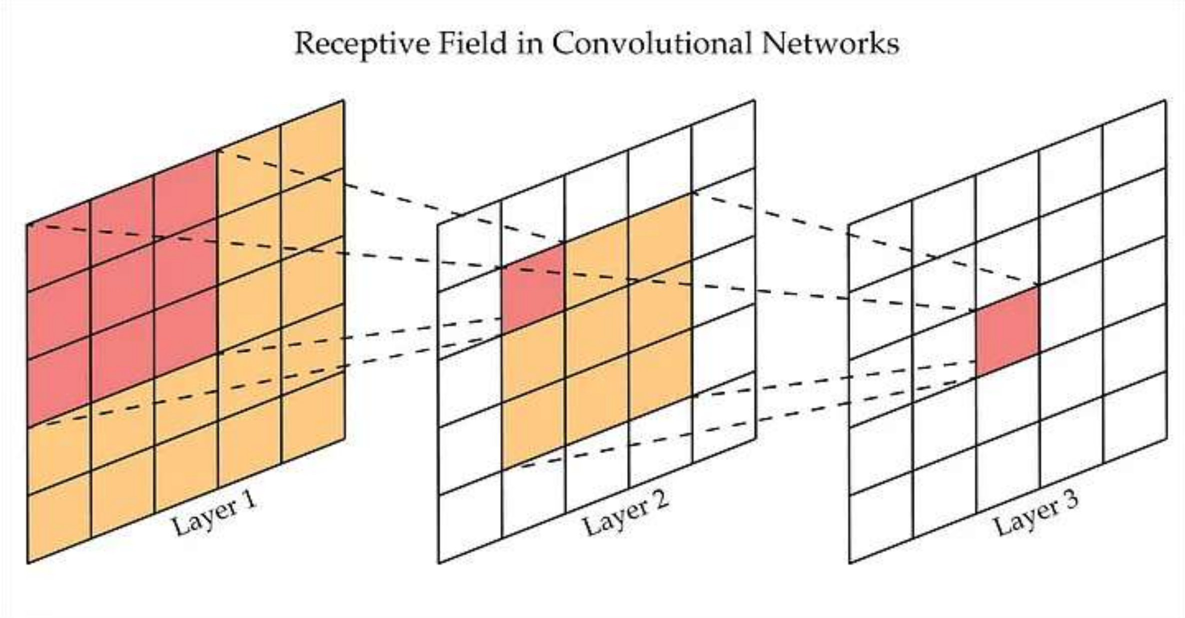
- CNN(Convolution Neural Network)



* Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.

Introduction

- CNN(Convolution Neural Network)



* Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.

Introduction

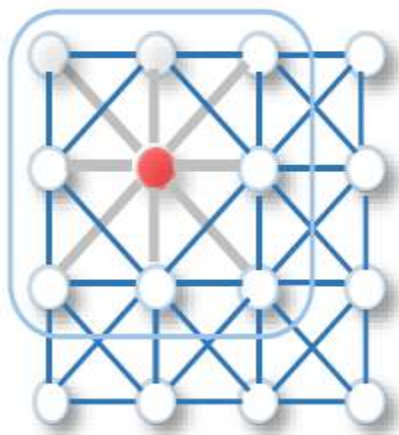
- CNN(Convolution Neural Network)
  - Want to see “Globally” -> GO DEEPER!!!

* Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.

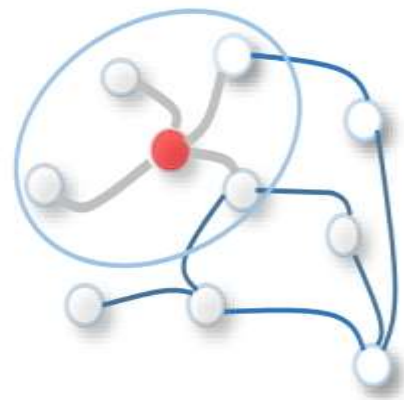
Introdution

◆ GNN(Graph Neural Network)

- CNN의 목적을 Graph Data Structure 에 적용



CNN



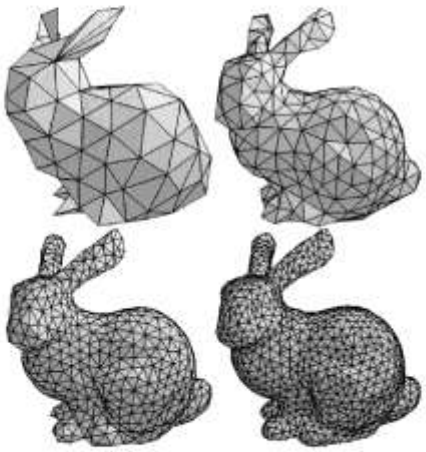
GCN

* Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." *IEEE transactions on neural networks and learning systems* 32.1 (2020): 4-24.

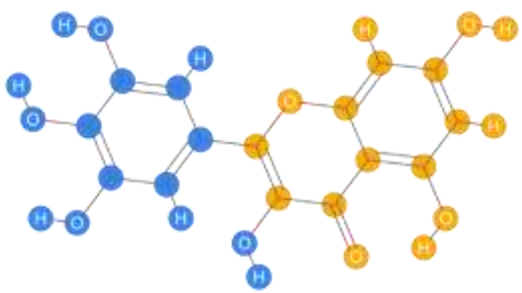
Graph Data Structure

◆ Graph Data Structure

Social Graph



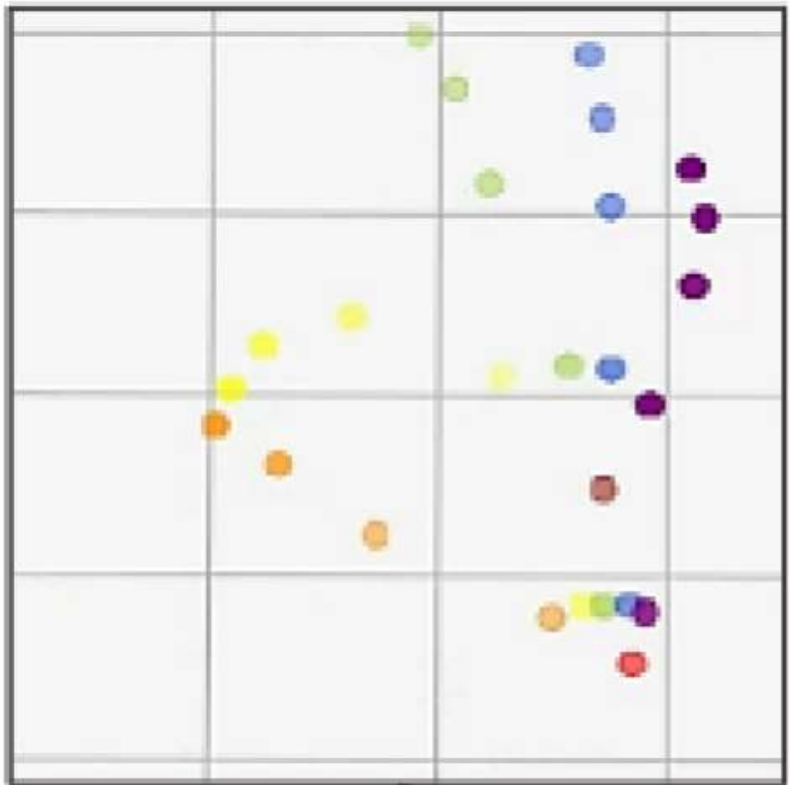
3D Mesh



Molecular Graph

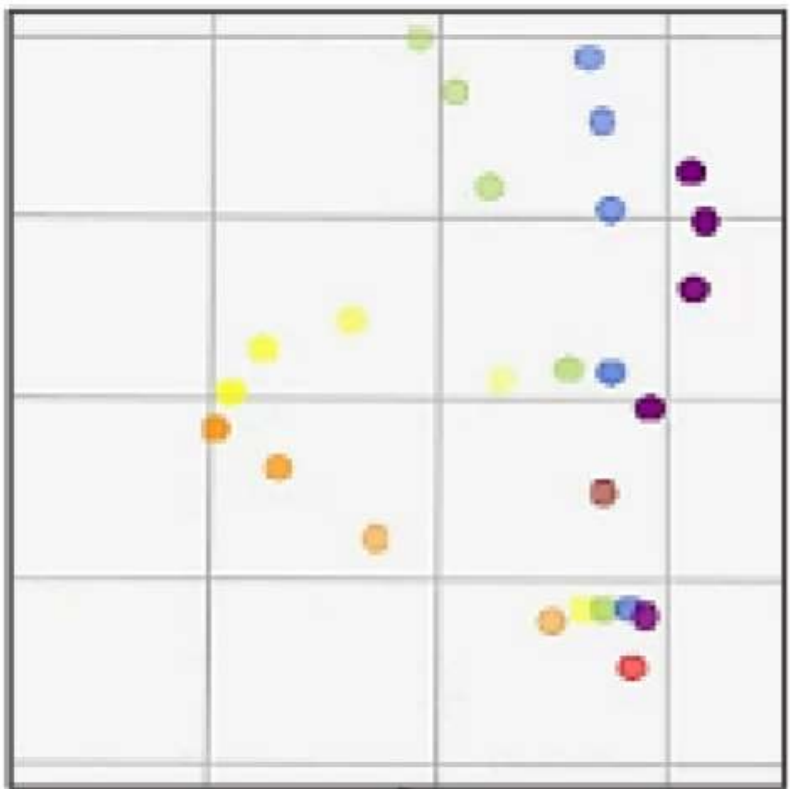
Graph Data Structure

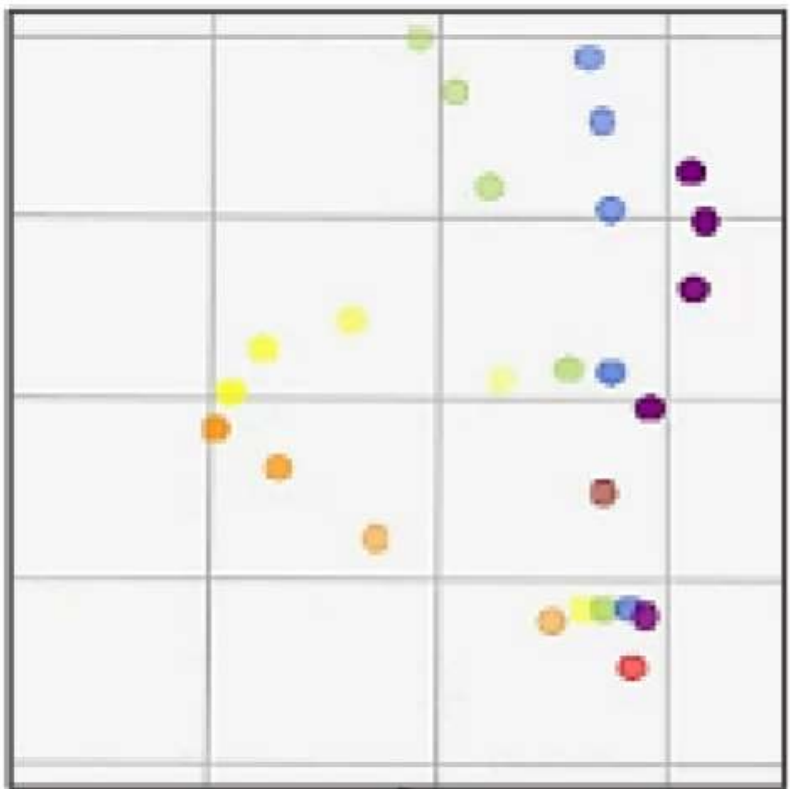
- Graph Data Structure



Graph Data Structure

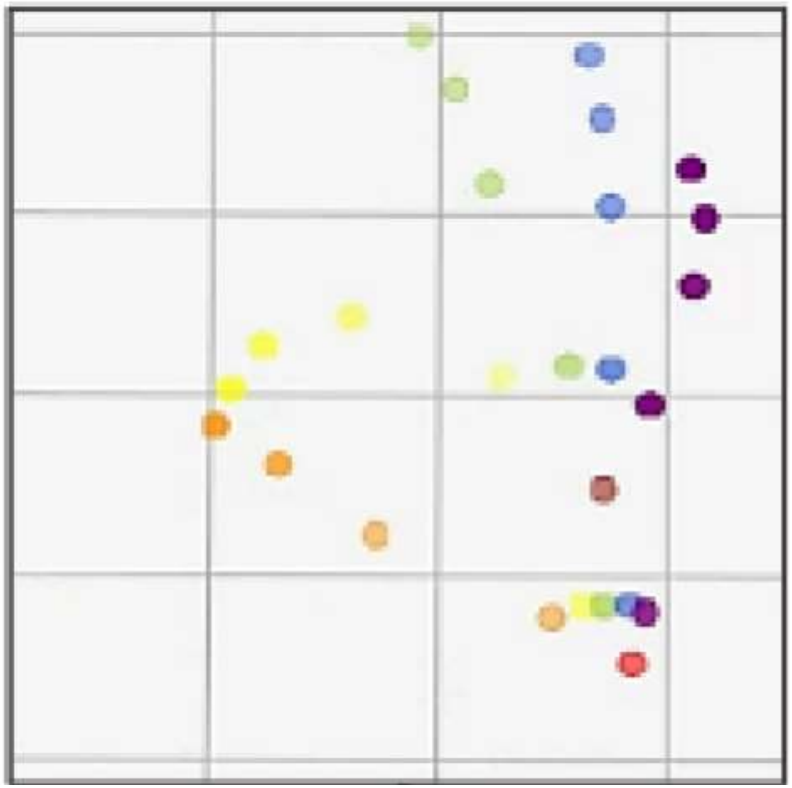
- Graph Data Structure





Graph Data Structure

- Graph Data Structure

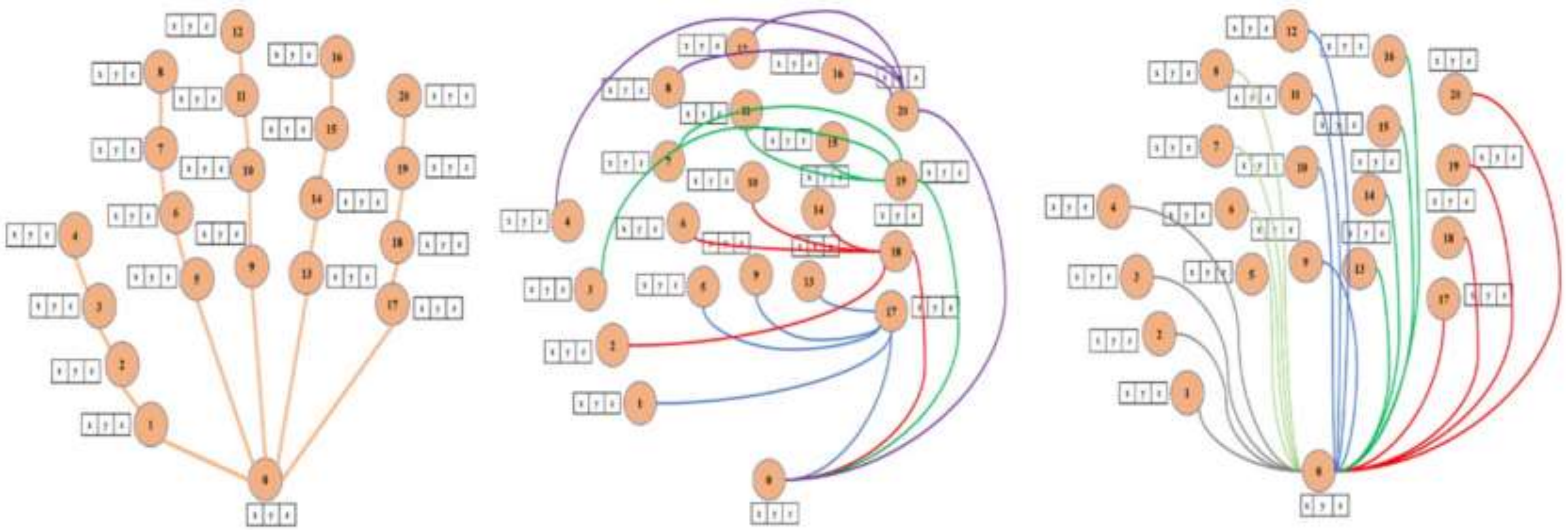


$$ {}_{26}C_2 = 325 $$

$$ 2^{325} = 683,516,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000,000 $$

Graph Data Structure

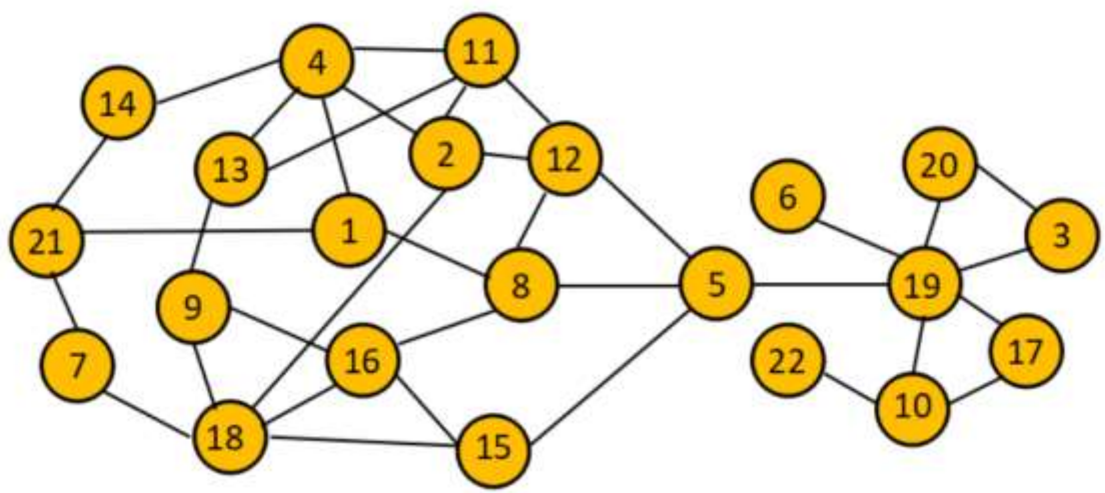
- Graph Data Structure
  - "Anything can be a graph,
    when we consider it as a graph"



Dept. ME, Dongguk Univ.

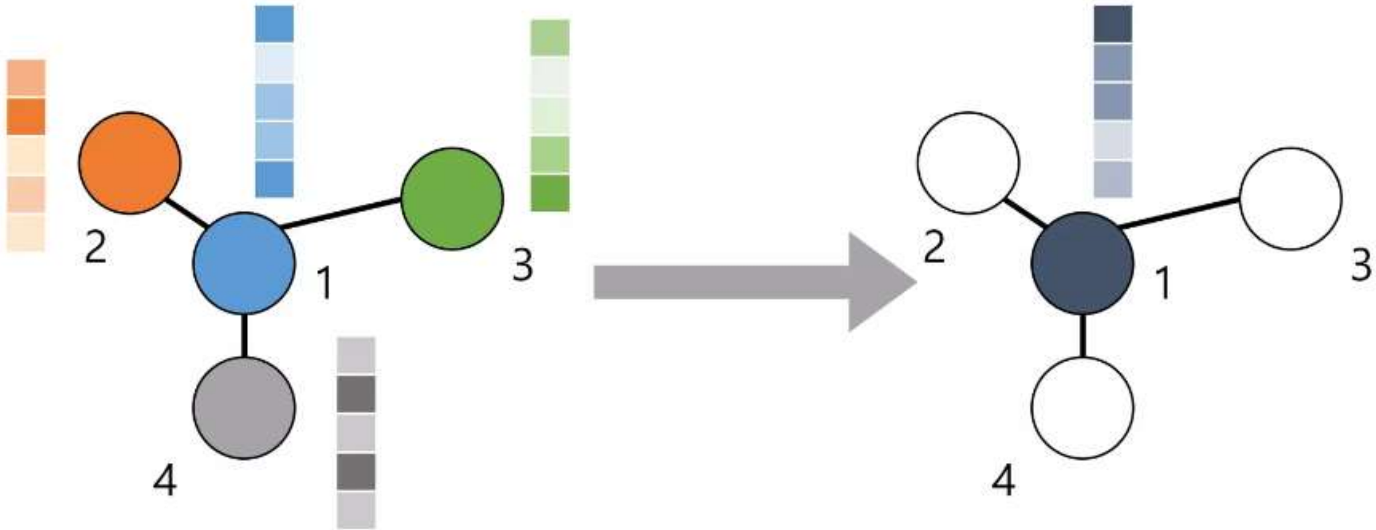
Graph Neural Network

- Tasks
  - Node focused task
    - Node classification
    - Link prediction
    - Feature Prediction
  - Graph focused task



GCN(Graph Convolution Network)

◆ Principle

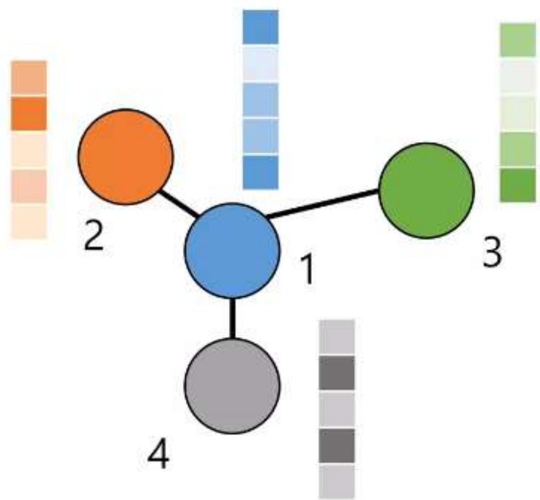


GCN(Graph Convolution Network)

◆ Principle

- Ex) Consider updating Node#1

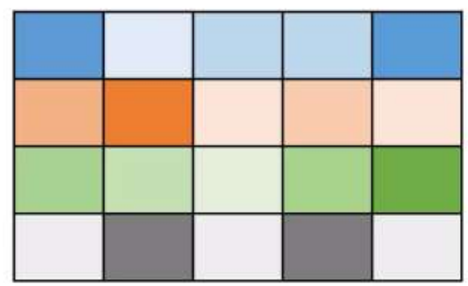
  - 4 Nodes
  - 5 Features for each node



Adjacency Matrix **A** (4 x 4)

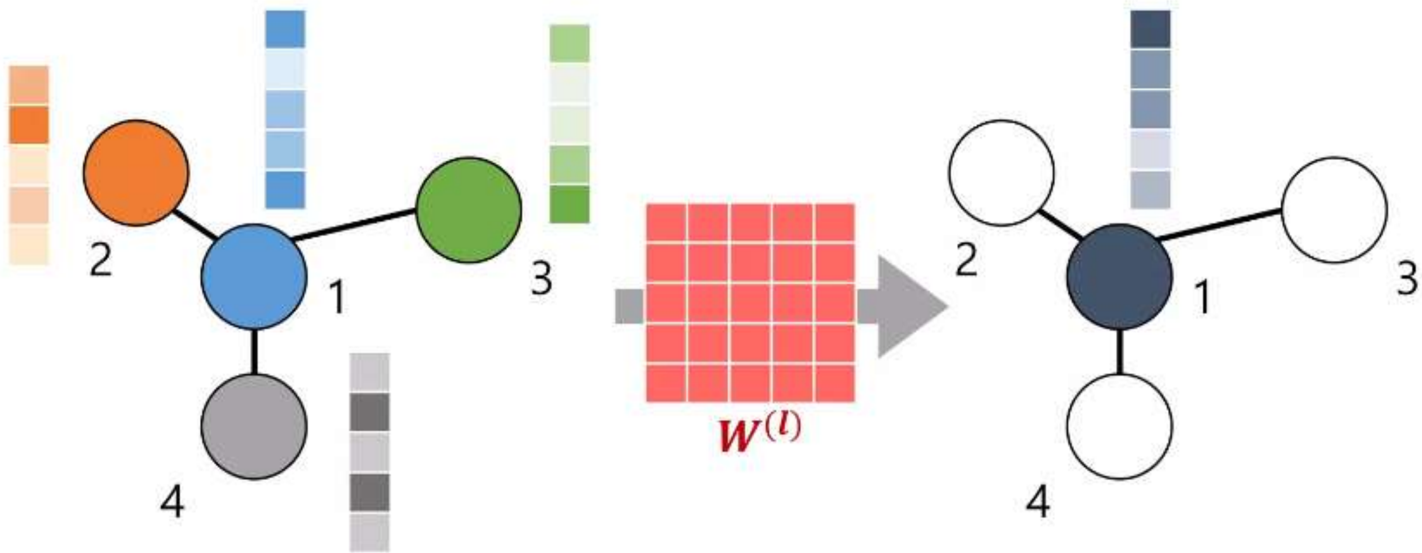
1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1

Feature Matrix **X** (4 x 5)



GCN(Graph Convolution Network)

- ◆ Principle
  - Ex) Consider updating Node#1

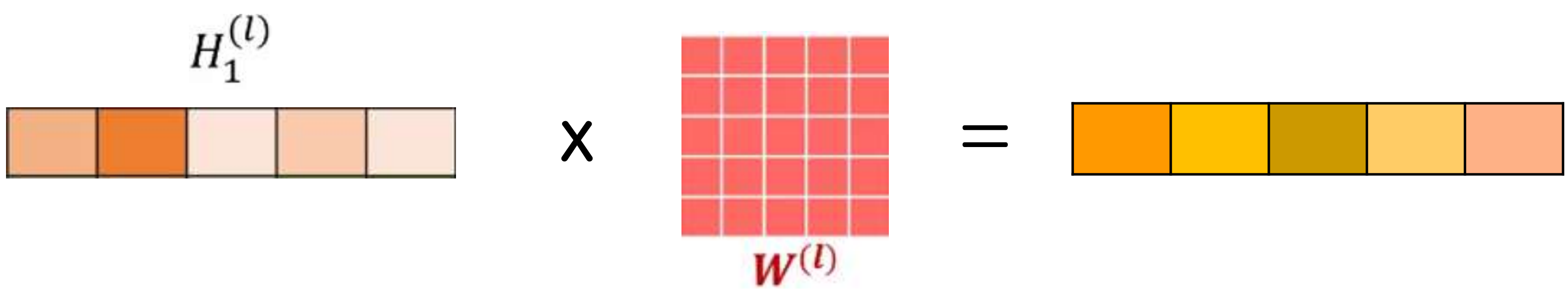


$$H_1^{(l+1)} = \sigma\left(H_1^{(l)}\boldsymbol{W}^{(l)} + H_2^{(l)}\boldsymbol{W}^{(l)} + H_3^{(l)}\boldsymbol{W}^{(l)} + H_4^{(l)}\boldsymbol{W}^{(l)} + b^{(l)}\right)$$

GCN(Graph Convolution Network)

- ◆ Principle
  - Ex) Consider updating Node#1

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

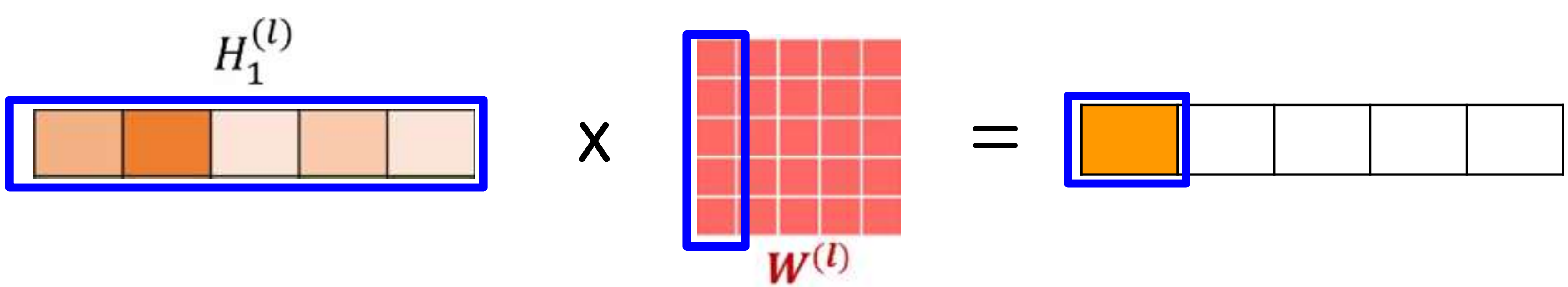


GCN(Graph Convolution Network)

◆ Principle

- Ex) Consider updating Node#1

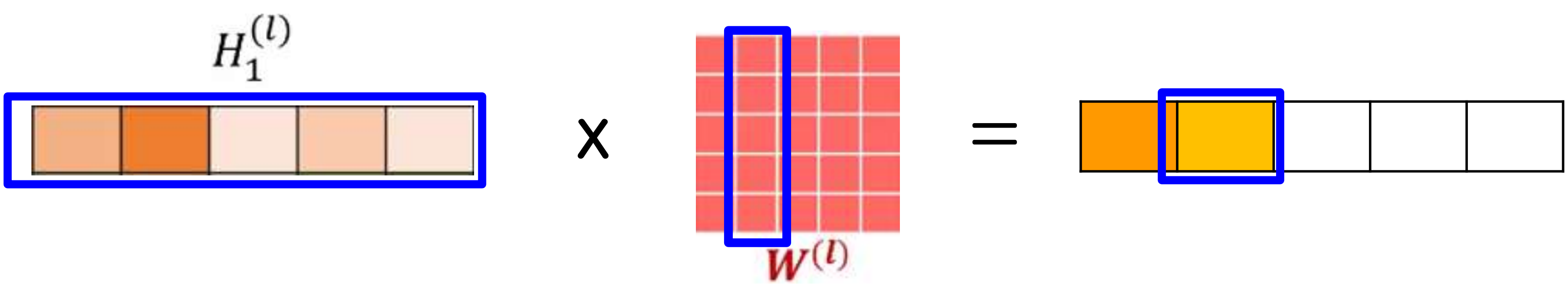
$$H_1^{(l+1)} = \sigma\left(\boxed{H_1^{(l)}\boldsymbol{W}^{(l)}} + H_2^{(l)}\boldsymbol{W}^{(l)} + H_3^{(l)}\boldsymbol{W}^{(l)} + H_4^{(l)}\boldsymbol{W}^{(l)} + b^{(l)}\right)$$



GCN(Graph Convolution Network)

- Principle
  - Ex) Consider updating Node#1

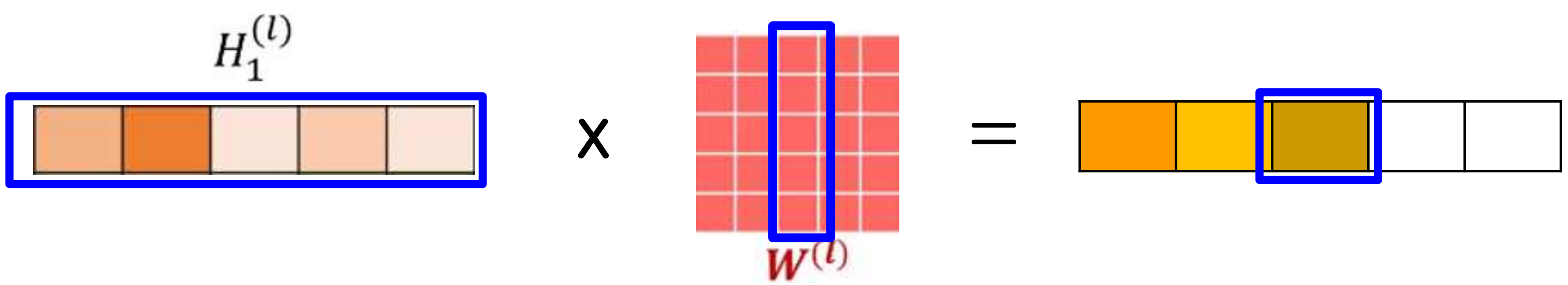
$$H_1^{(l+1)} = \sigma\left(\boxed{H_1^{(l)}\boldsymbol{W}^{(l)}} + H_2^{(l)}\boldsymbol{W}^{(l)} + H_3^{(l)}\boldsymbol{W}^{(l)} + H_4^{(l)}\boldsymbol{W}^{(l)} + b^{(l)}\right)$$



GCN(Graph Convolution Network)

- Principle
  - Ex) Consider updating Node#1

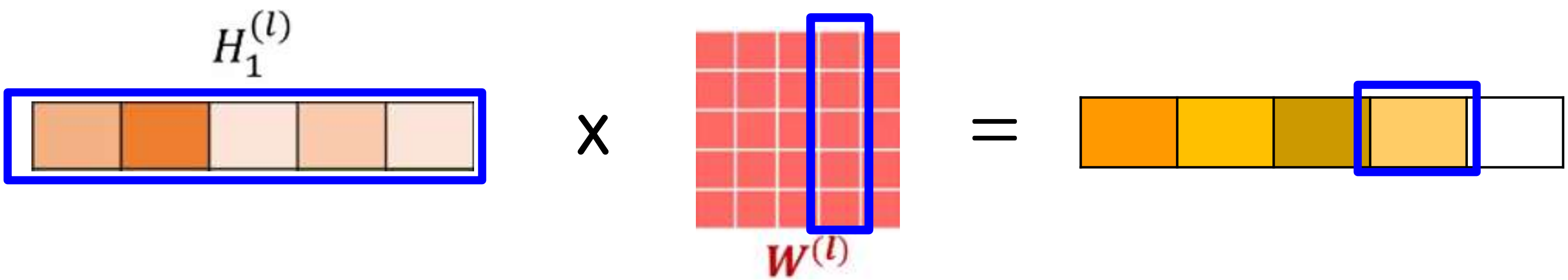
$$H_1^{(l+1)} = \sigma\left(\boxed{H_1^{(l)} \mathbf{W}^{(l)}} + H_2^{(l)} \mathbf{W}^{(l)} + H_3^{(l)} \mathbf{W}^{(l)} + H_4^{(l)} \mathbf{W}^{(l)} + b^{(l)}\right)$$



GCN(Graph Convolution Network)

- Principle
  - Ex) Consider updating Node#1

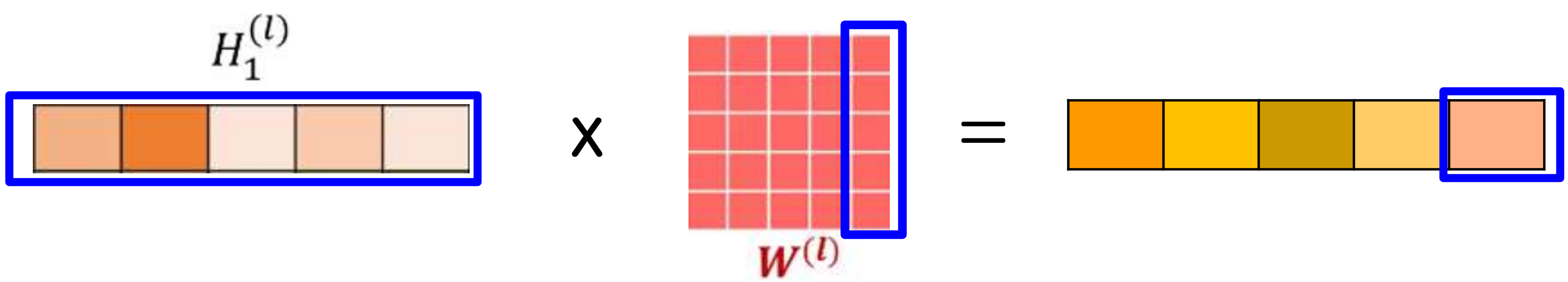
$$H_1^{(l+1)} = \sigma\left(\boxed{H_1^{(l)}\boldsymbol{W}^{(l)}} + H_2^{(l)}\boldsymbol{W}^{(l)} + H_3^{(l)}\boldsymbol{W}^{(l)} + H_4^{(l)}\boldsymbol{W}^{(l)} + b^{(l)}\right)$$



GCN(Graph Convolution Network)

- Principle
  - Ex) Consider updating Node#1

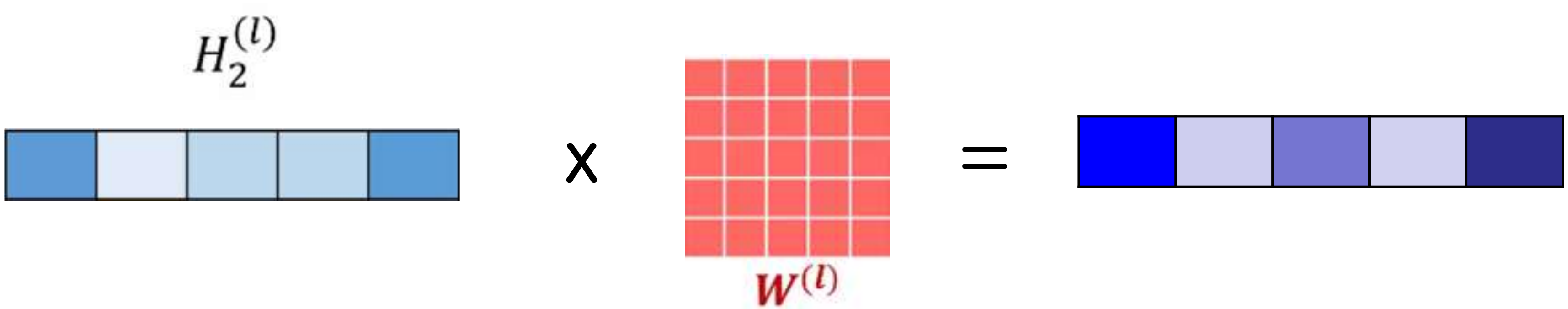
$$H_1^{(l+1)} = \sigma\left(\boxed{H_1^{(l)}\boldsymbol{W}^{(l)}} + H_2^{(l)}\boldsymbol{W}^{(l)} + H_3^{(l)}\boldsymbol{W}^{(l)} + H_4^{(l)}\boldsymbol{W}^{(l)} + b^{(l)}\right)$$



GCN(Graph Convolution Network)

- Principle
  - Ex) Consider updating Node#1

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

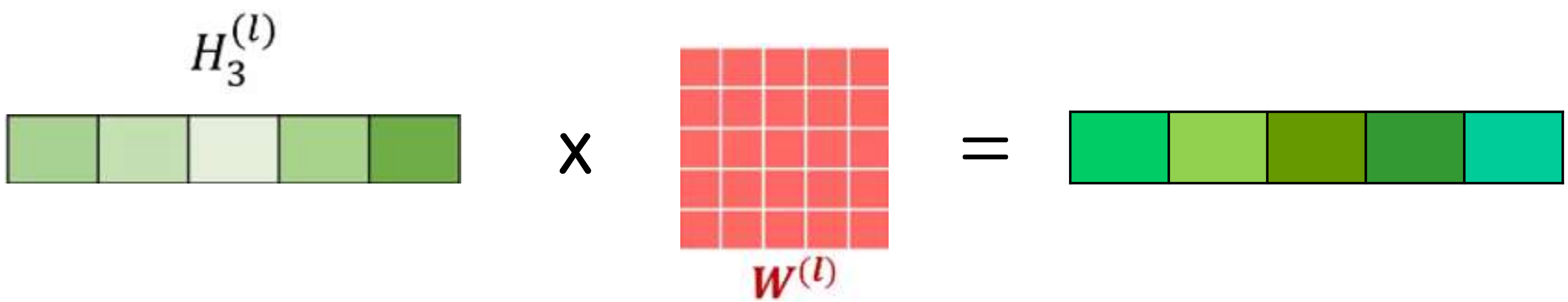


GCN(Graph Convolution Network)

Principle

- Ex) Consider updating Node#1

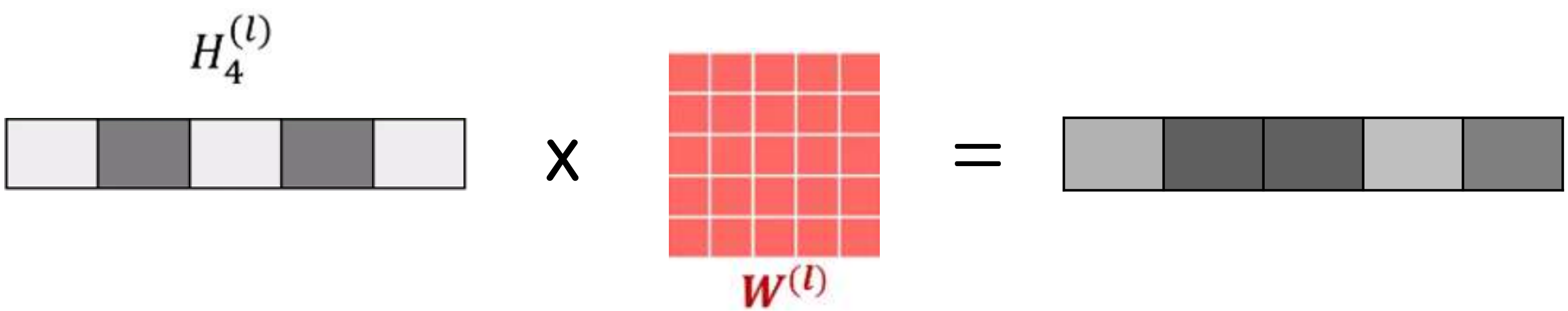
$$H_1^{(l+1)} = \sigma\left(H_1^{(l)}\mathbf{W}^{(l)} + H_2^{(l)}\mathbf{W}^{(l)} + \boxed{H_3^{(l)}\mathbf{W}^{(l)}} + H_4^{(l)}\mathbf{W}^{(l)} + b^{(l)}\right)$$



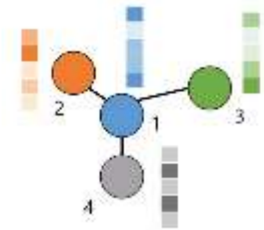
GCN(Graph Convolution Network)

- Principle
  - Ex) Consider updating Node#1

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

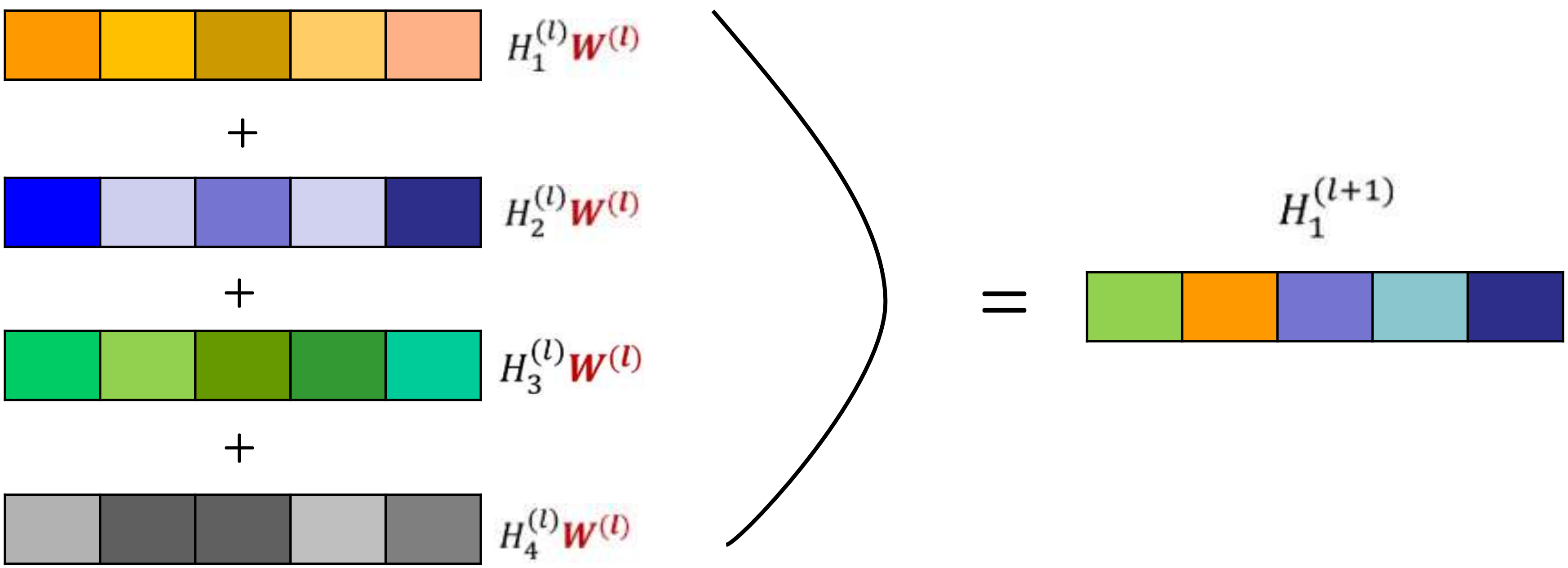


GCN(Graph Convolution Network)



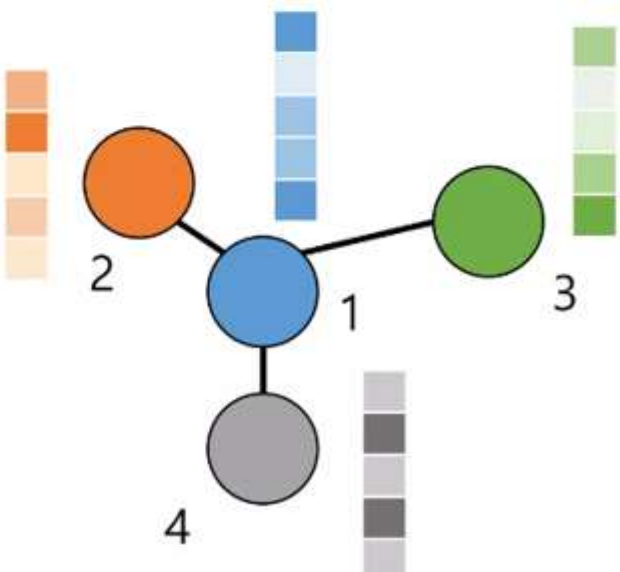
- Principle
  - Ex) Consider updating Node#1

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



GCN(Graph Convolution Network)

- Principle
  - Ex) Consider updating Node#2



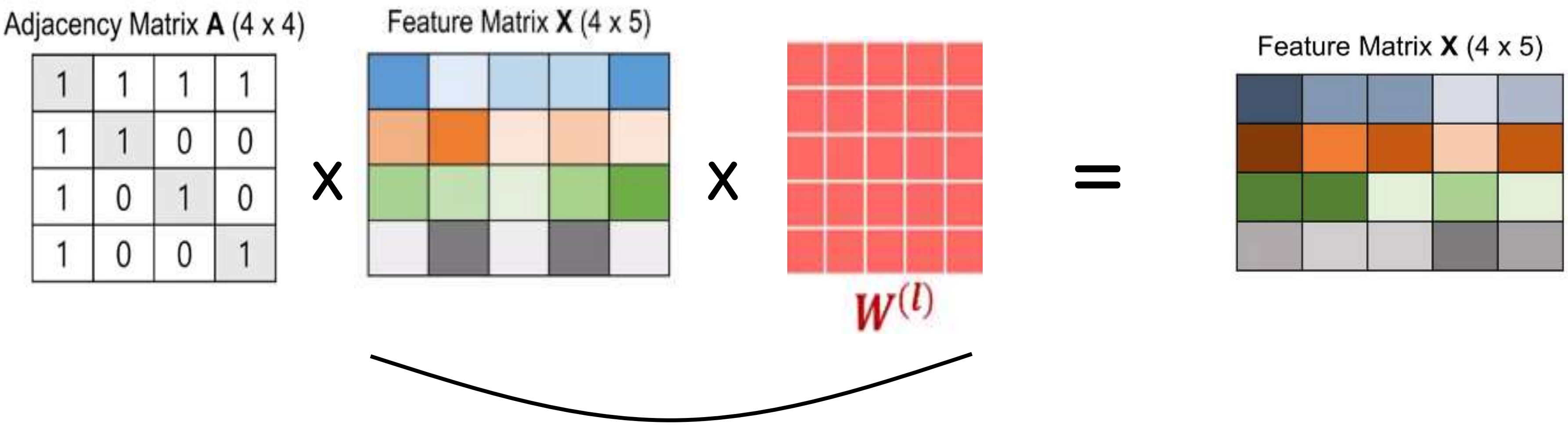
Adjacency Matrix **A** (4 x 4)

1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1

GCN(Graph Convolution Network)

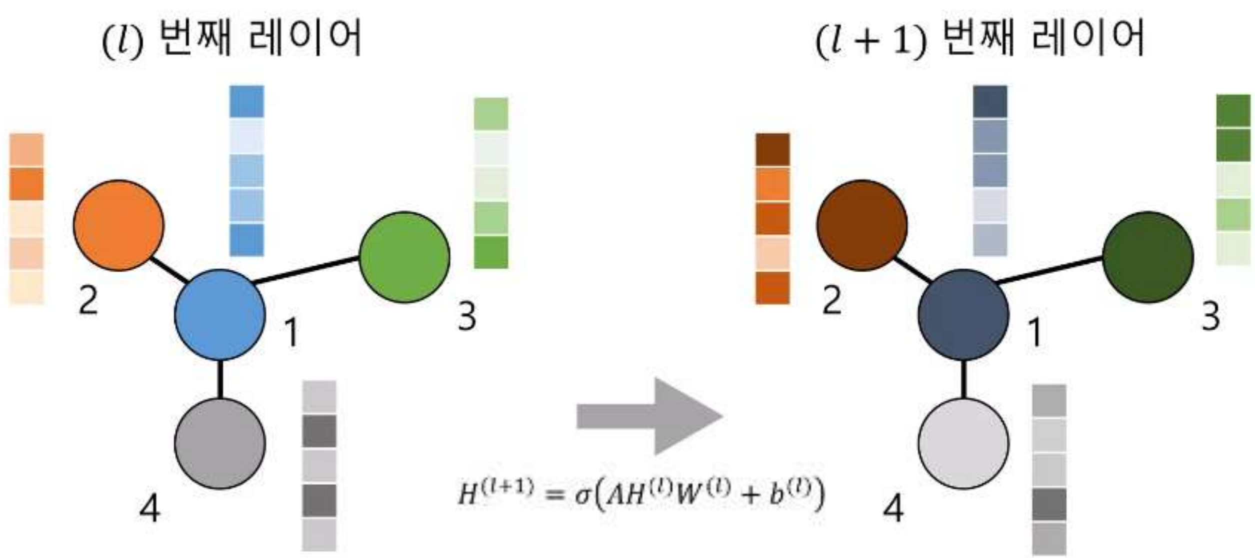
◆ Principle

- Ex) Consider updating Node#2



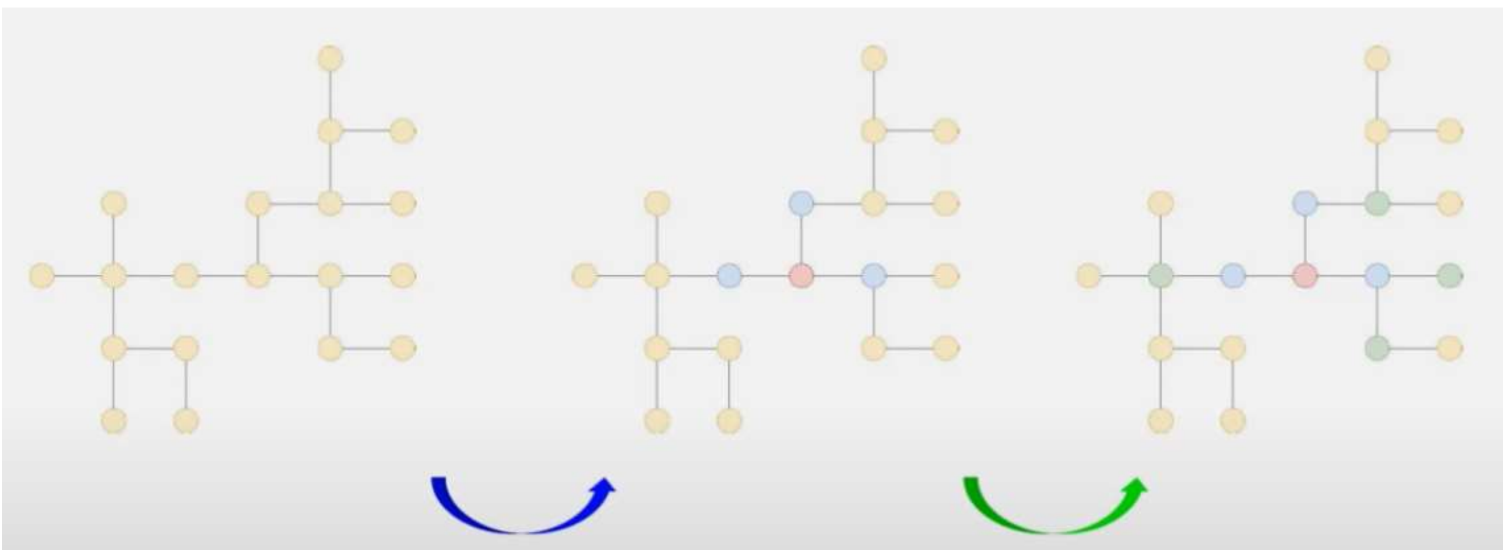
GCN(Graph Convolution Network)

- Principle
  - Ex) Consider updating Node#2



GCN(Graph Convolution Network)

- Principle
  - Ex) Consider updating Node#2



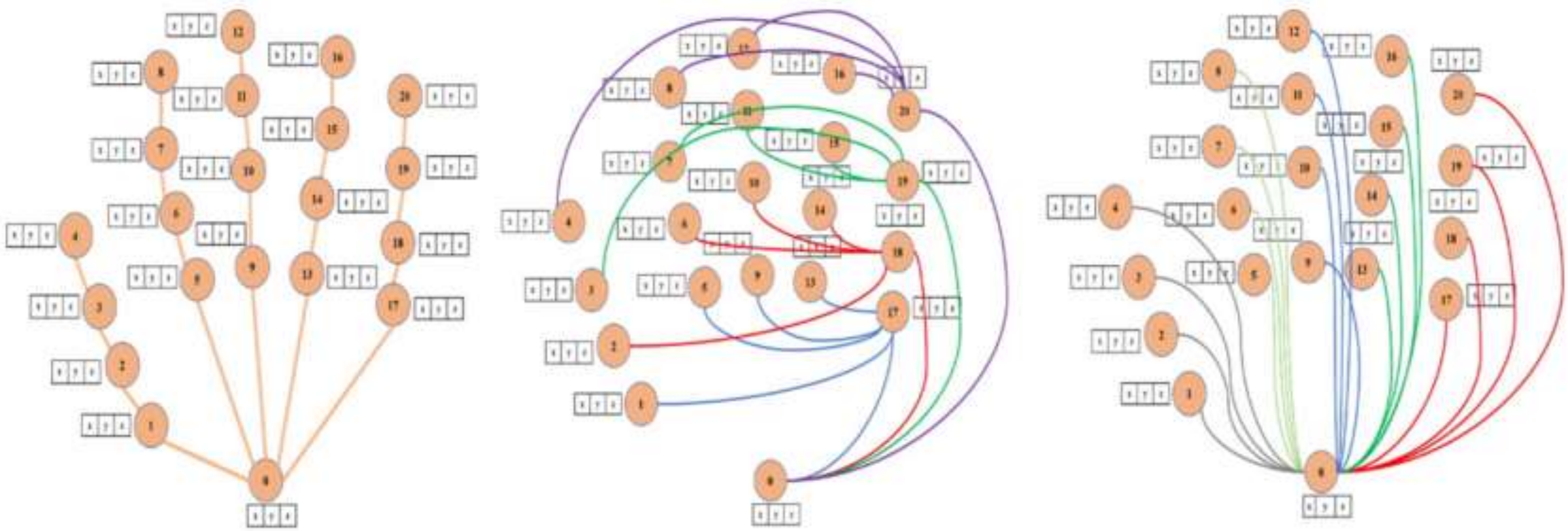
GCN
Layer

GCN
Layer

GCN(Graph Convolution Network)

◆ Limitation

- Initial Specific Graph 에 국한된다

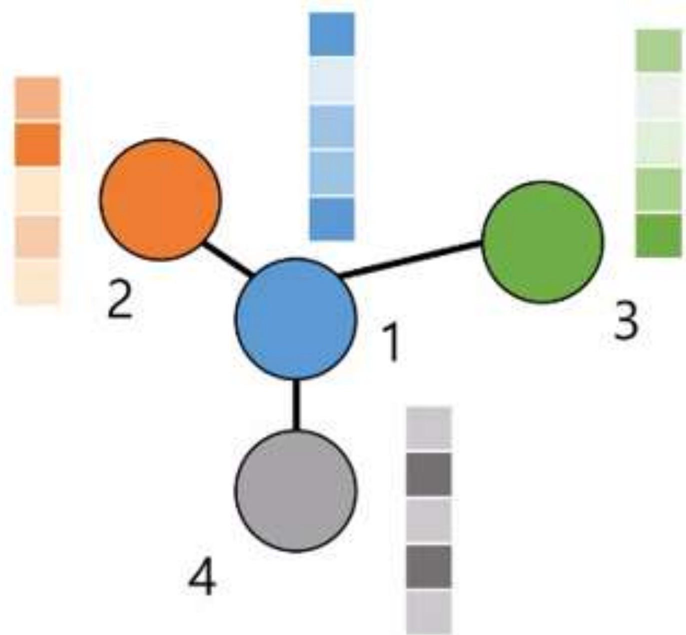


GCN(Graph Convolution Network)

- **Limitation**
  - Edge들의 가중치는 학습하지 못함
    - 다른 노드의 정보를 가지고 오냐 오지 않는냐 (이진법)

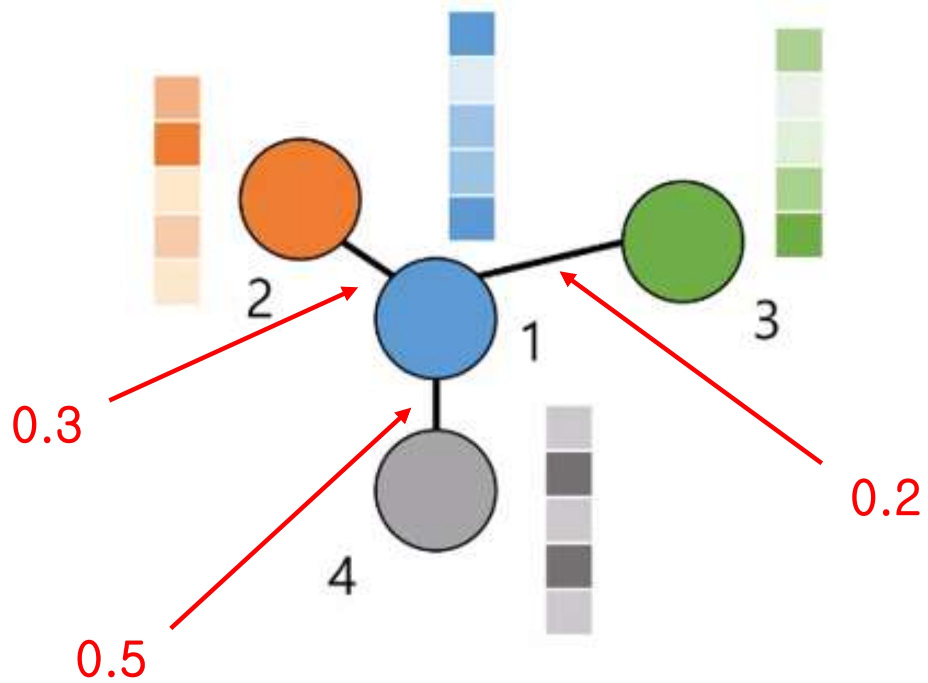
Adjacency Matrix **A** (4 x 4)

1	1	1	1
1	1	0	0
1	0	1	0
1	0	0	1



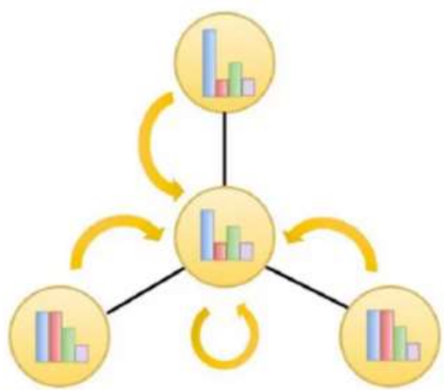
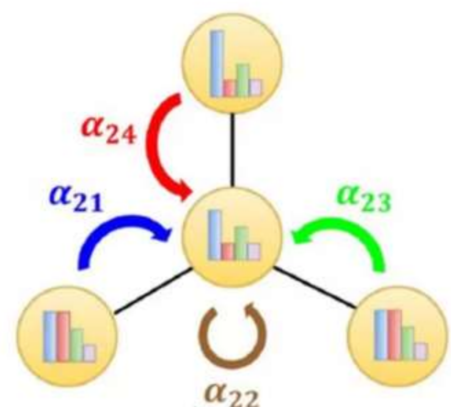
GAT(Graph Attention Network)

◆ Edge의 가중치를 학습



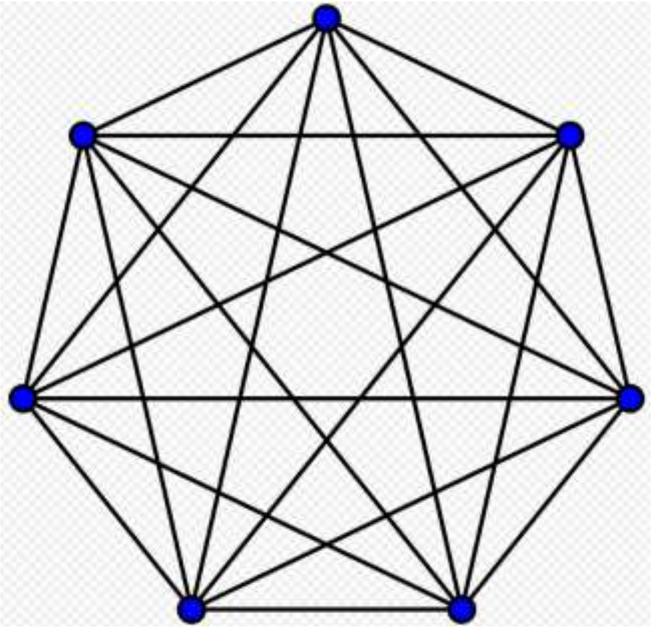
GAT(Graph Attention Network)

◆ Edge의 가중치를 학습

GCN	GAT
	
$H^{(l+1)} = \sigma\left(\sum_{j \in N(i)} H_j^{(l)} W^{(l)}\right)$	$H^{(l+1)} = \sigma\left(\sum_{j \in N(i)} \alpha_{ij} H_j^{(l)} W^{(l)}\right)$

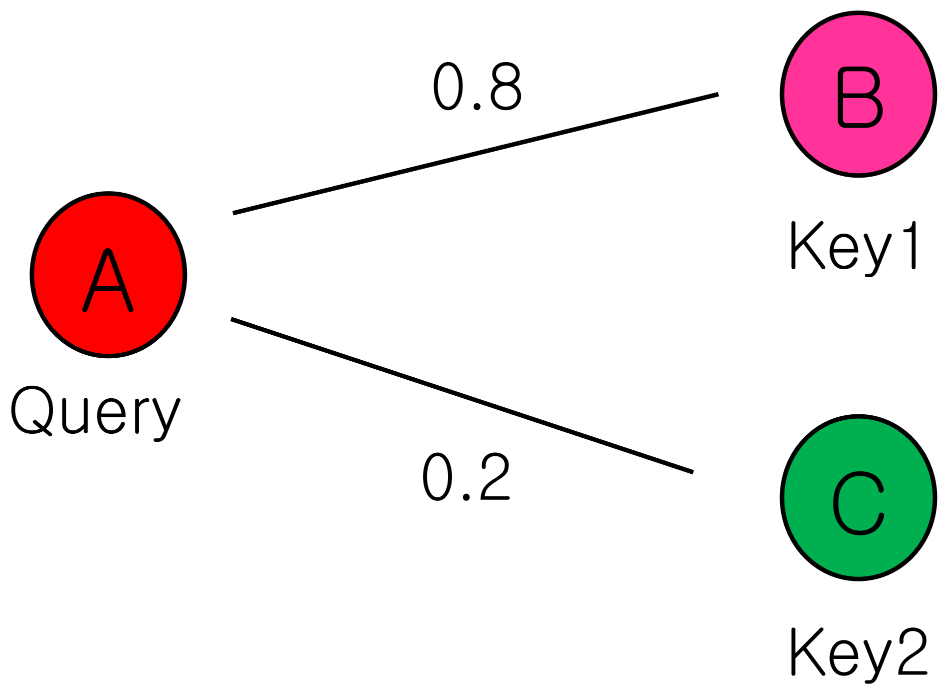
GAT(Graph Attention Network)

- General Graph를 사용가능
  - Fully Connected



GAT(Graph Attention Network)

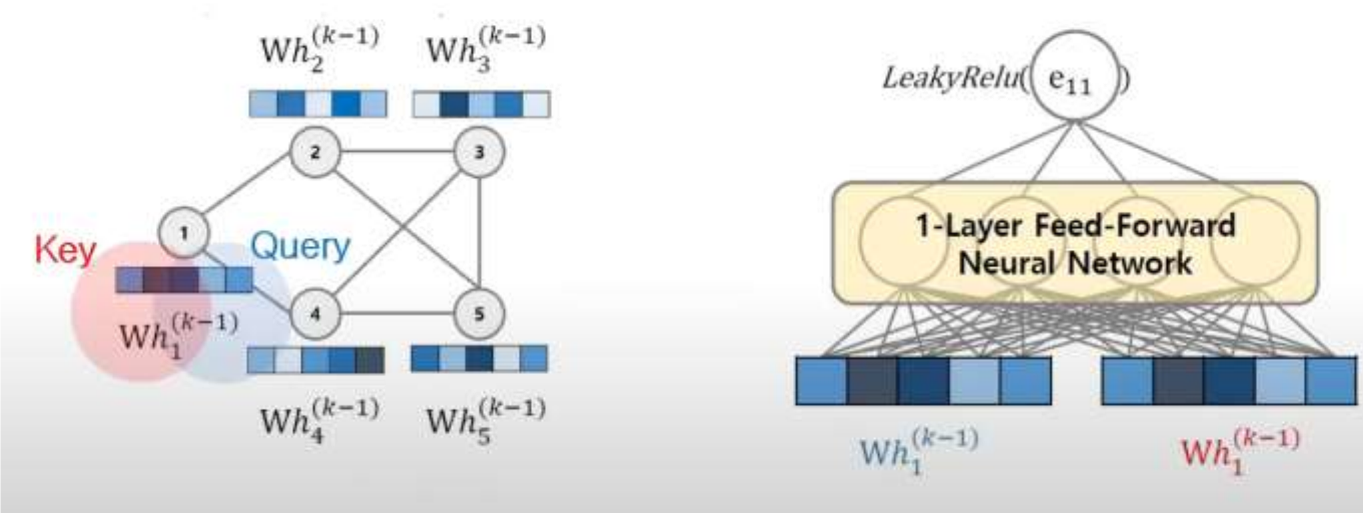
- Principle
  - Transformer의 Encoder 구조를 따름
    - Adjacency Matrix : 다른(이웃) 노드와 내(노드)가 얼마나 유사한가를 측정해서 기록



GAT(Graph Attention Network)

◆ Principle

- Ex) Intial Specific Graph가 주어졌을 때



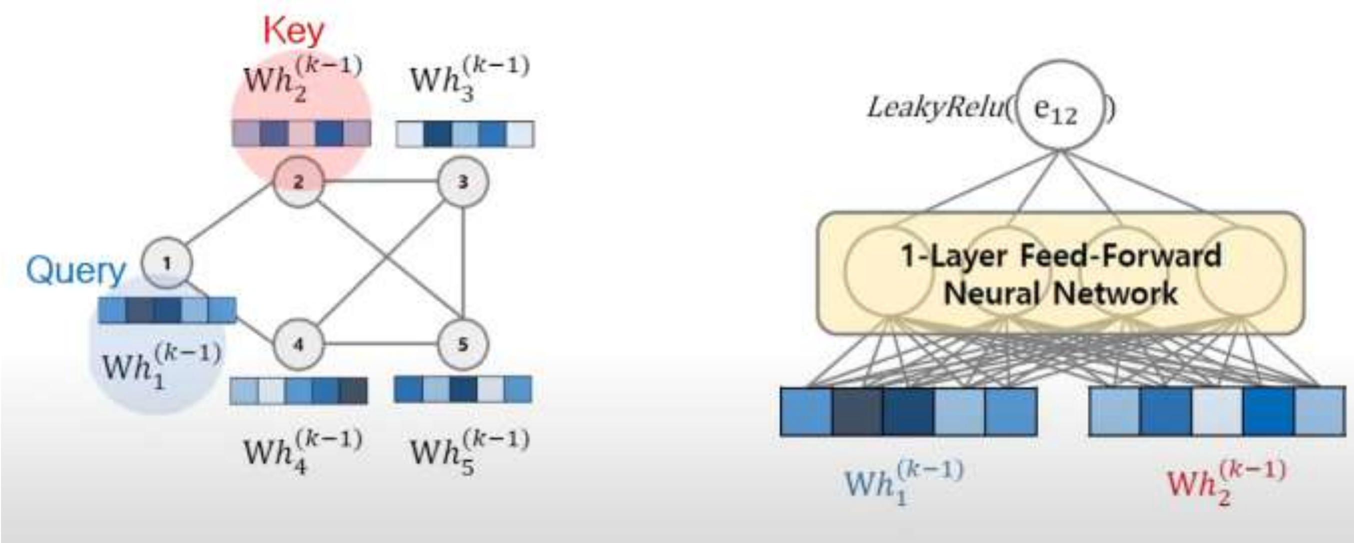
Adjacency Matrix

e_{11}				

GAT(Graph Attention Network)

◆ Principle

- Ex) Intial Specific Graph가 주어졌을 때



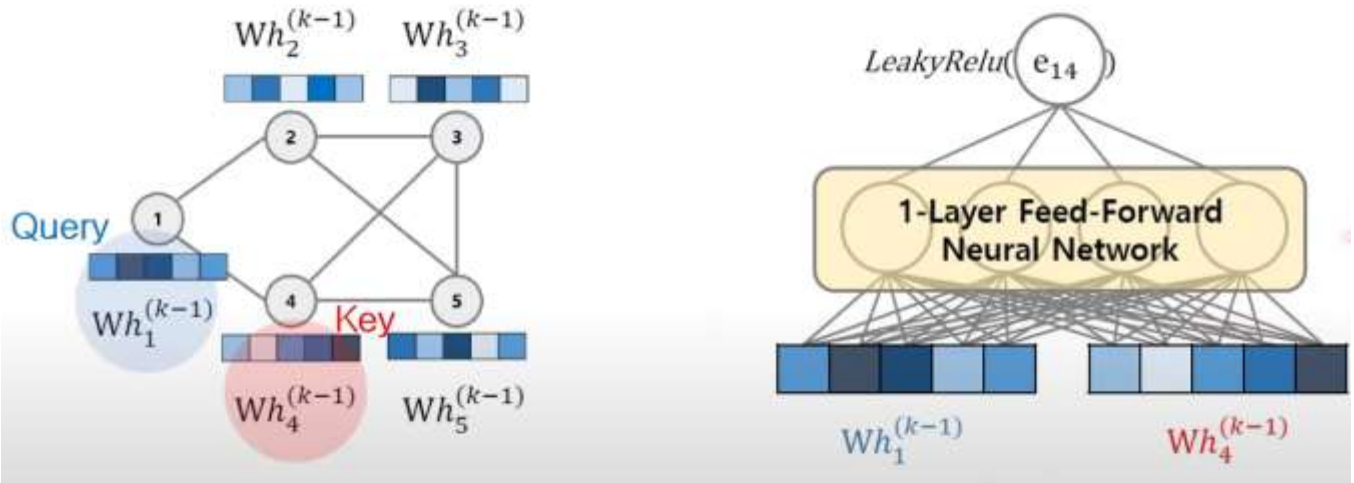
Adjacency Matrix

e_{11}	e_{12}			

GAT(Graph Attention Network)

◆ Principle

- Ex) Intial Specific Graph가 주어졌을 때



Adjacency Matrix

e_{11}	e_{12}		e_{14}	

GAT(Graph Attention Network)

- Principle
  - Ex) Intial Specific Graph가 주어졌을 때

e_{11}	e_{12}	e_{13}	e_{14}	e_{15}
e_{21}	e_{22}	e_{23}	e_{24}	e_{25}
e_{31}	e_{32}	e_{33}	e_{34}	e_{35}
e_{41}	e_{42}	e_{43}	e_{44}	e_{45}
e_{51}	e_{52}	e_{53}	e_{54}	e_{55}

Similarity

e_{11}	e_{12}	$-9e^{10}$	e_{14}	$-9e^{10}$
e_{21}	e_{22}	e_{23}	$-9e^{10}$	e_{25}
$-9e^{10}$	e_{32}	e_{33}	e_{34}	e_{35}
e_{41}	$-9e^{10}$	e_{43}	e_{44}	e_{45}
$-9e^{10}$	e_{52}	e_{53}	e_{54}	e_{55}

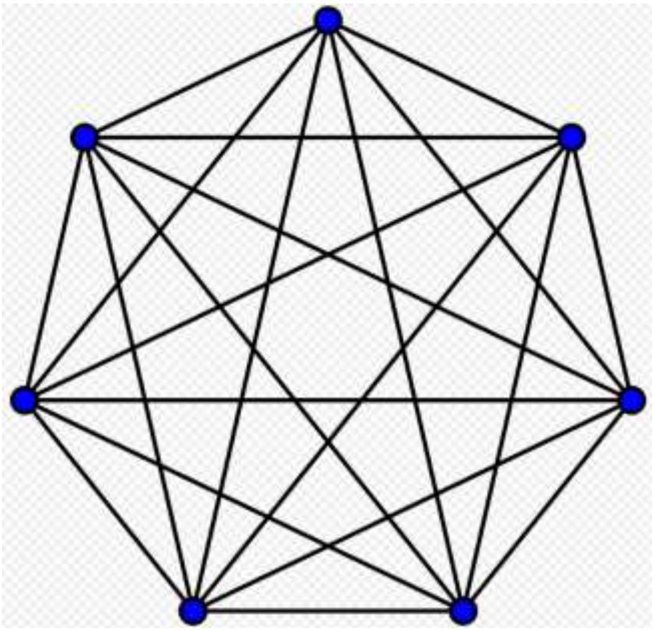
Masking

a_{11}	a_{12}	0	a_{14}	0
a_{21}	a_{22}	a_{23}	0	a_{25}
0	a_{32}	a_{33}	a_{34}	a_{35}
a_{41}	0	a_{43}	a_{44}	a_{45}
0	a_{52}	a_{53}	a_{54}	a_{55}

Softmax

GAT(Graph Attention Network)

- General Graph를 사용가능
  - Fully Connected



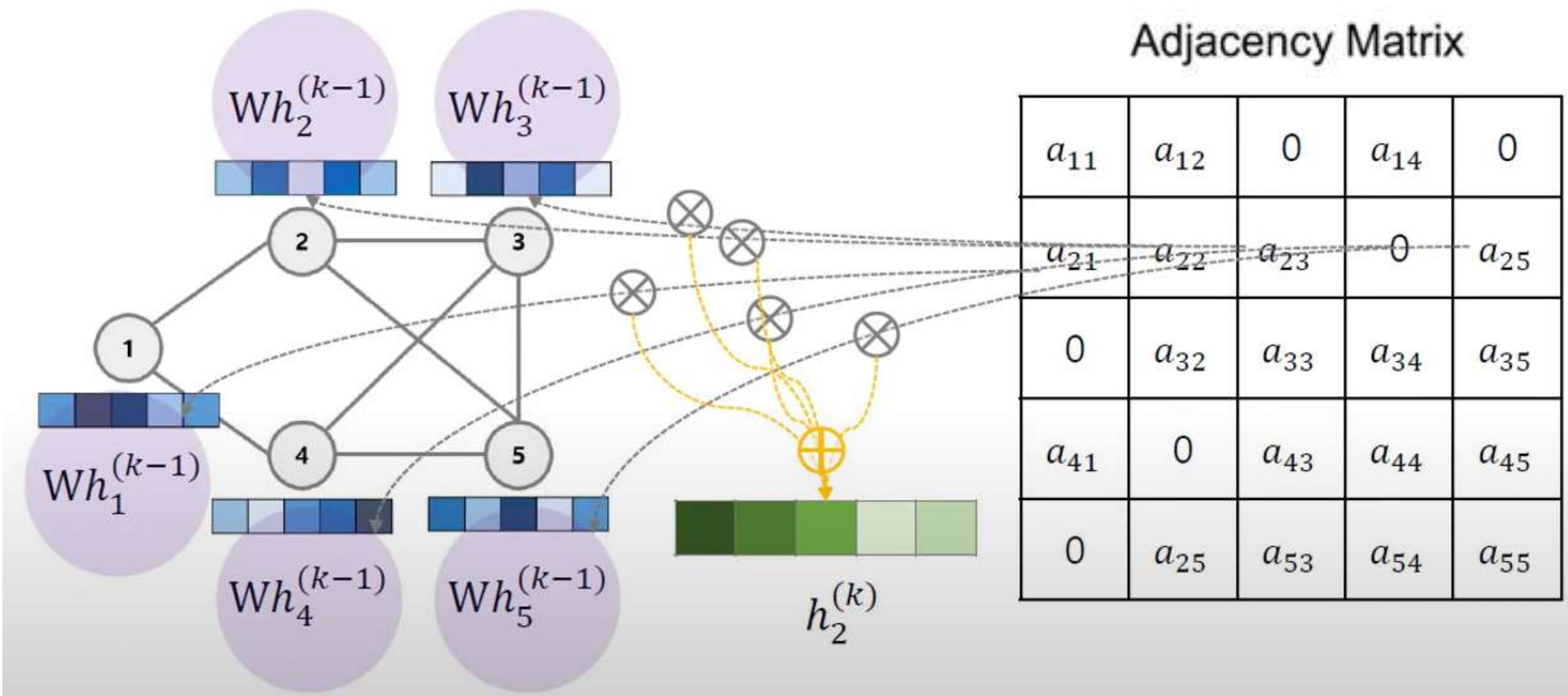
Adjacency Matrix

a_{11}	a_{12}	0	a_{14}	0
a_{21}	a_{22}	a_{23}	0	a_{25}
0	a_{32}	a_{33}	a_{34}	a_{35}
a_{41}	0	a_{43}	a_{44}	a_{45}
0	a_{52}	a_{53}	a_{54}	a_{55}

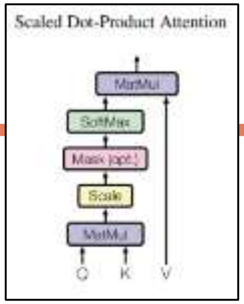
GAT(Graph Attention Network)

◆ Principle

- Ex) Intial Specific Graph가 주어졌을 때



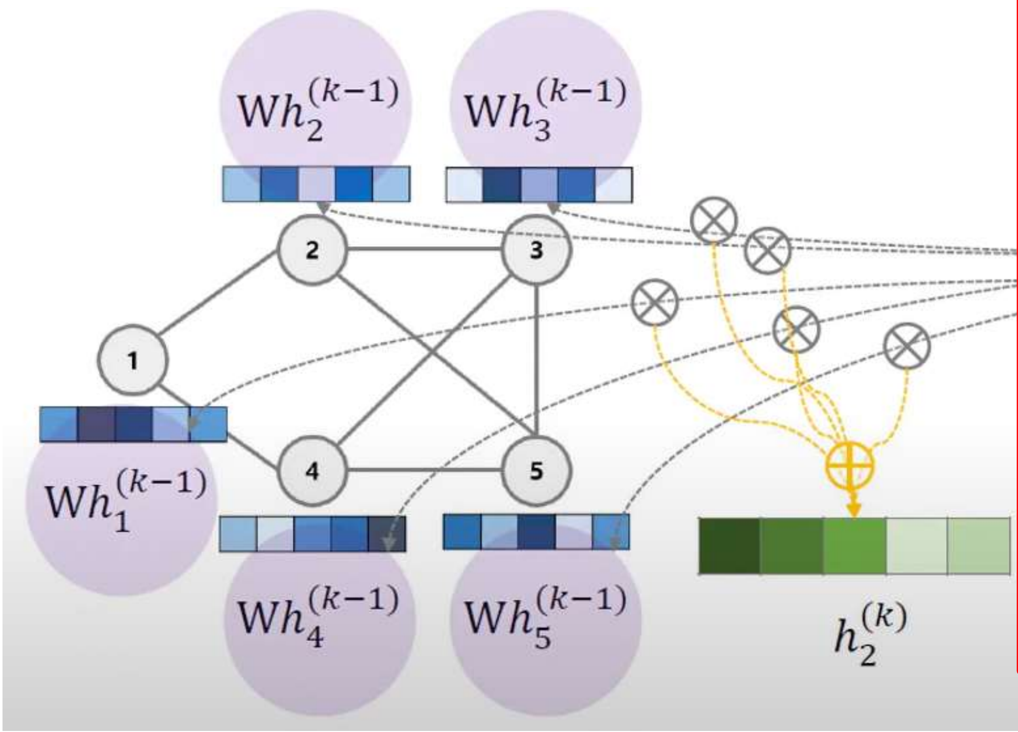
GAT(Graph Attention Network)



◆ Principle

- Transformer의 Encoder 구조를 따름

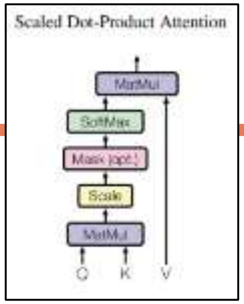
Matmul(Query, Key)



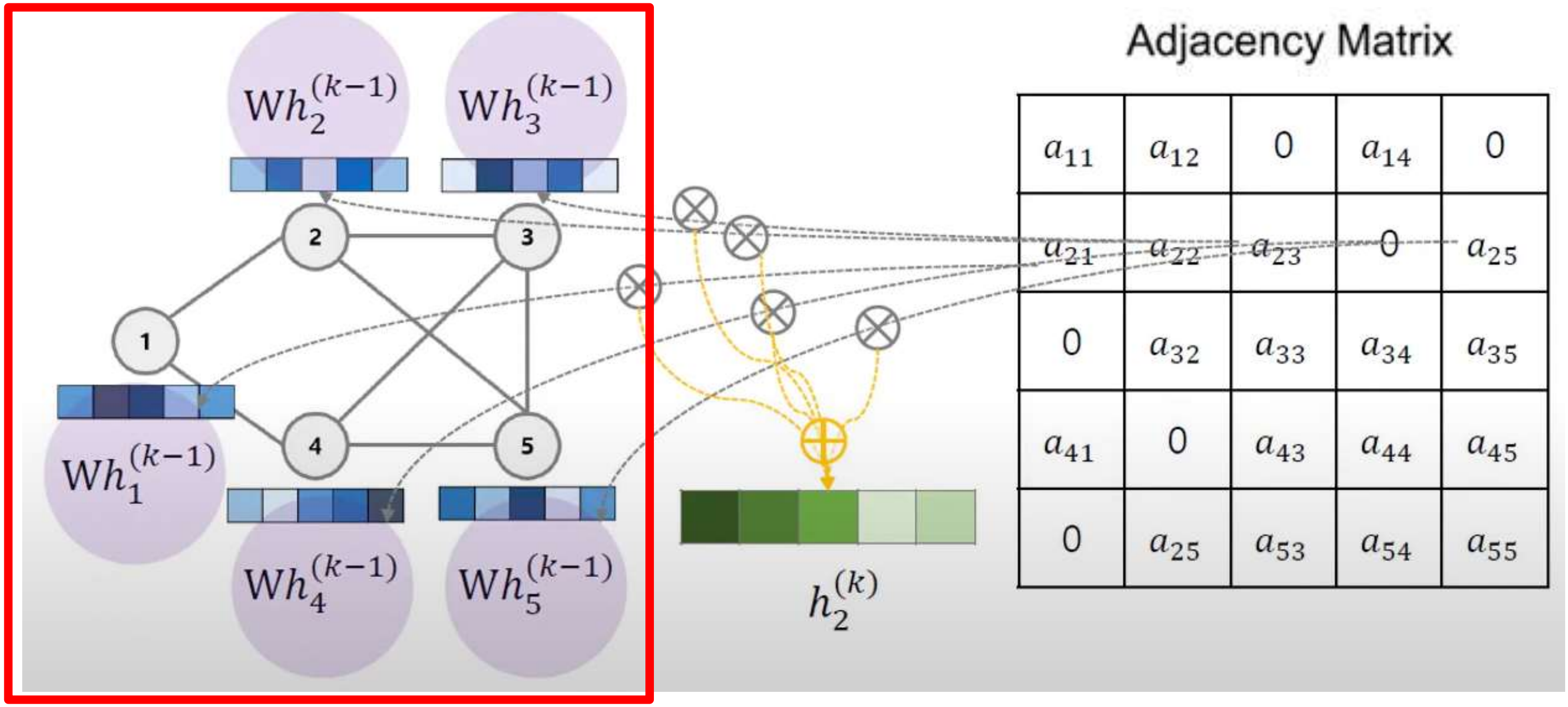
Adjacency Matrix

a_{11}	a_{12}	0	a_{14}	0
a_{21}	a_{22}	a_{23}	0	a_{25}
0	a_{32}	a_{33}	a_{34}	a_{35}
a_{41}	0	a_{43}	a_{44}	a_{45}
0	a_{25}	a_{53}	a_{54}	a_{55}

GAT(Graph Attention Network)

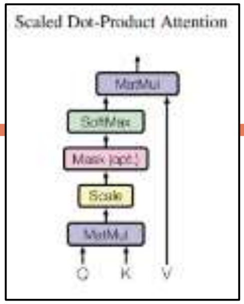


- Principle
  - Transformer의 개념을 사용

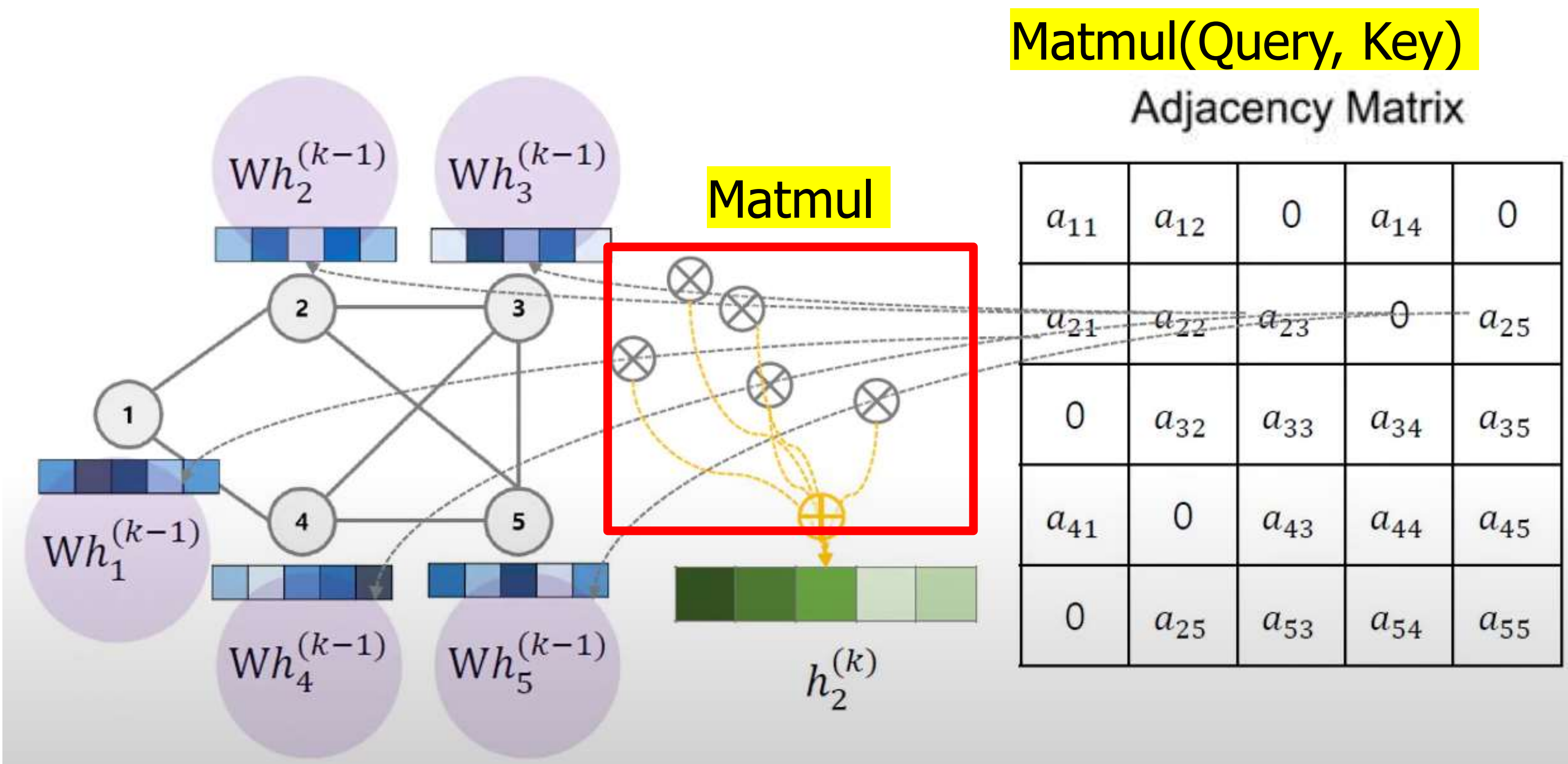


Value

GAT(Graph Attention Network)



- Principle
  - Transformer의 개념을 사용



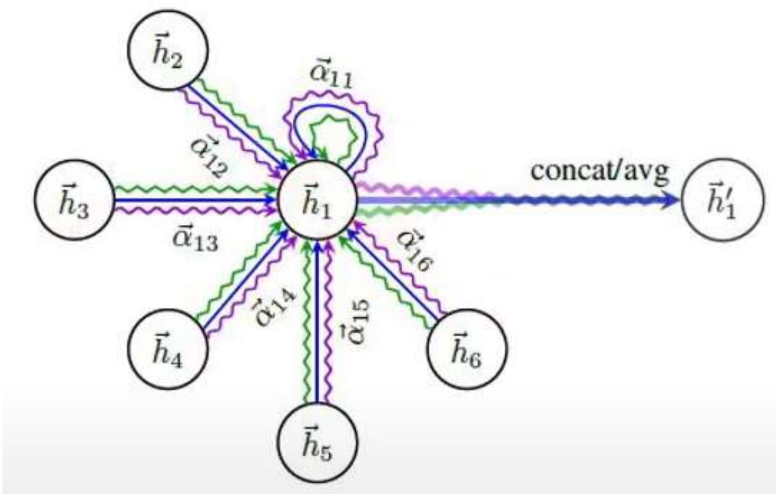
a_{11}	a_{12}	0	a_{14}	0
a_{21}	a_{22}	a_{23}	0	a_{25}
0	a_{32}	a_{33}	a_{34}	a_{35}
a_{41}	0	a_{43}	a_{44}	a_{45}
0	a_{25}	a_{53}	a_{54}	a_{55}

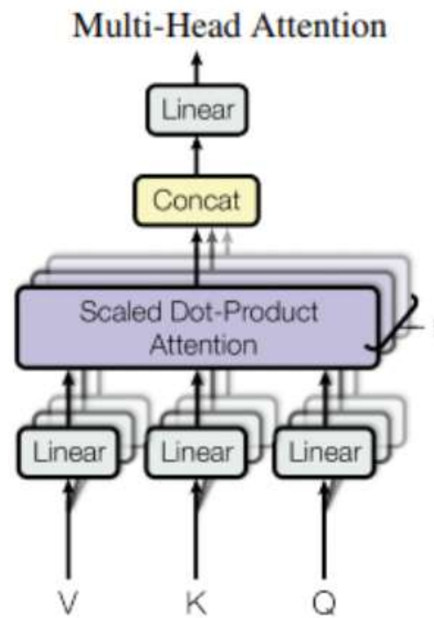
Value

GAT(Graph Attention Network)

◆ Principle

- Multi-Head 사용





GAT(Graph Attention Network)

- Principle
  - Mathematical Expression

$$z_i^{(l)} = W^{(l)} h_i^{(l)}, \quad (1)$$

$$e_{ij}^{(l)} = \text{LeakyReLU}(\vec{a}^{(l)^T}(z_i^{(l)} || z_j^{(l)})), \quad (2)$$

$$\alpha_{ij}^{(l)} = \frac{\exp(e_{ij}^{(l)})}{\sum_{k \in \mathcal{N}(i)} \exp(e_{ik}^{(l)})}, \quad (3)$$

$$h_i^{(l+1)} = \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} z_j^{(l)}\right), \quad (4)$$

$$\text{concatenation} : h_i^{(l+1)} = ||_{k=1}^{K} \sigma\left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{k} W^k h_j^{(l)}\right)$$

$$\text{average} : h_i^{(l+1)} = \sigma\left(\frac{1}{K} \sum_{k=1}^{K} \sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{k} W^k h_j^{(l)}\right)$$

Conclusion

- GCN, GAT
  - CNN과 목적은 동일
  - 주위의 중요한 노드 정보들을 종합하여, 모든 노드들이 해당 정보를 조금씩 내재하게 함으로써, 더 "Global" 하게 볼 수 있게 만들어 주는 역할을 한다.

Thank You