Idea Factory Intensive Program #2

딥러닝 홀로서기

#30

이론강의/PyTorch실습/코드리뷰

딥러닝(Deep Learning)에 관심이 있는 학생 발굴을 통한
딥러닝의 이론적 배경 강의 및 오픈소스 딥러닝 라이브러리 PyTorch를 활용한 실습

Topics to learn today

1. Review from last lecture

Lecture: Recurrent Neural Network

2. What is Graph?

Graph Structure

3. Graph Convolutional Network (GCN)

Basic of GCN

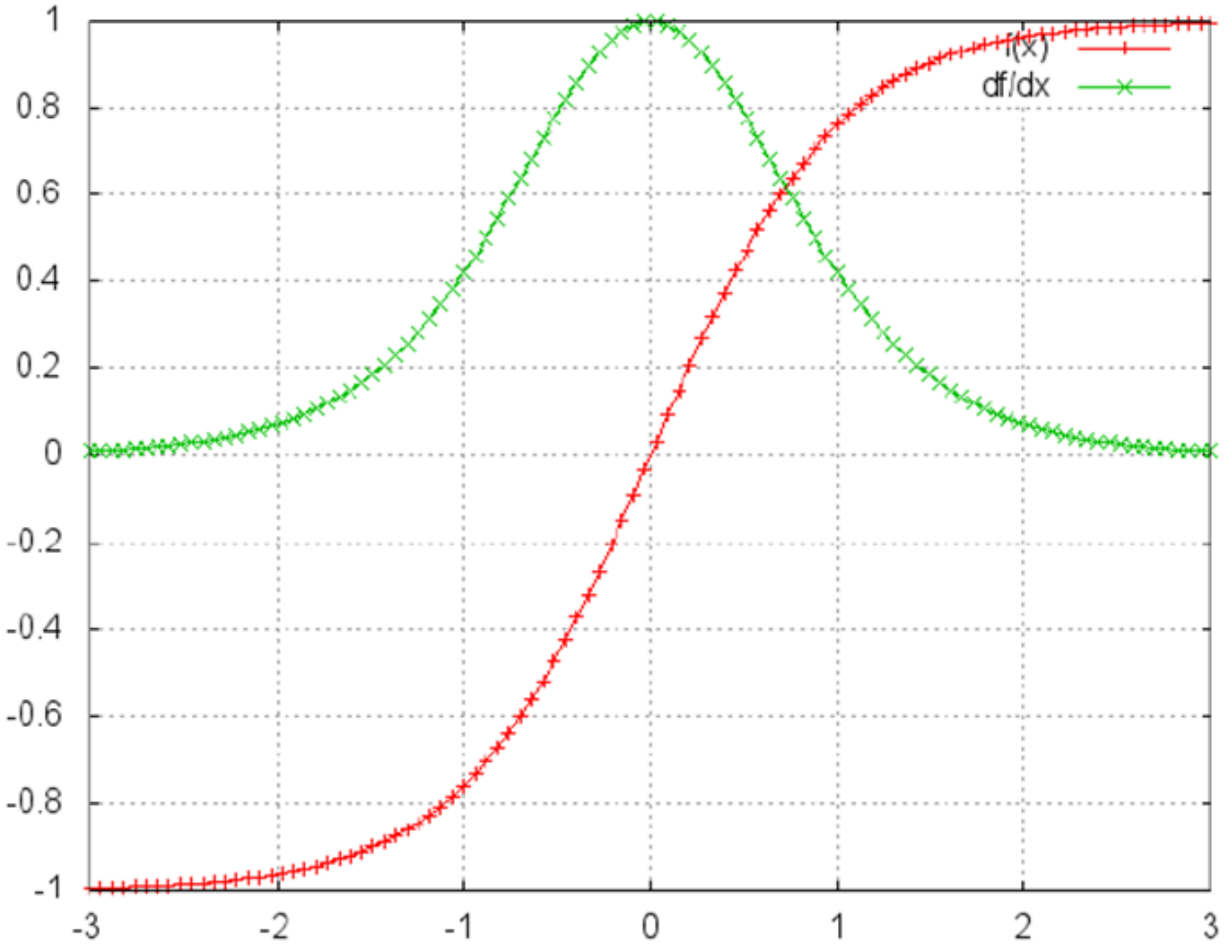
Advanced Techniques of GCN

4. Implementing GCN with pytorch

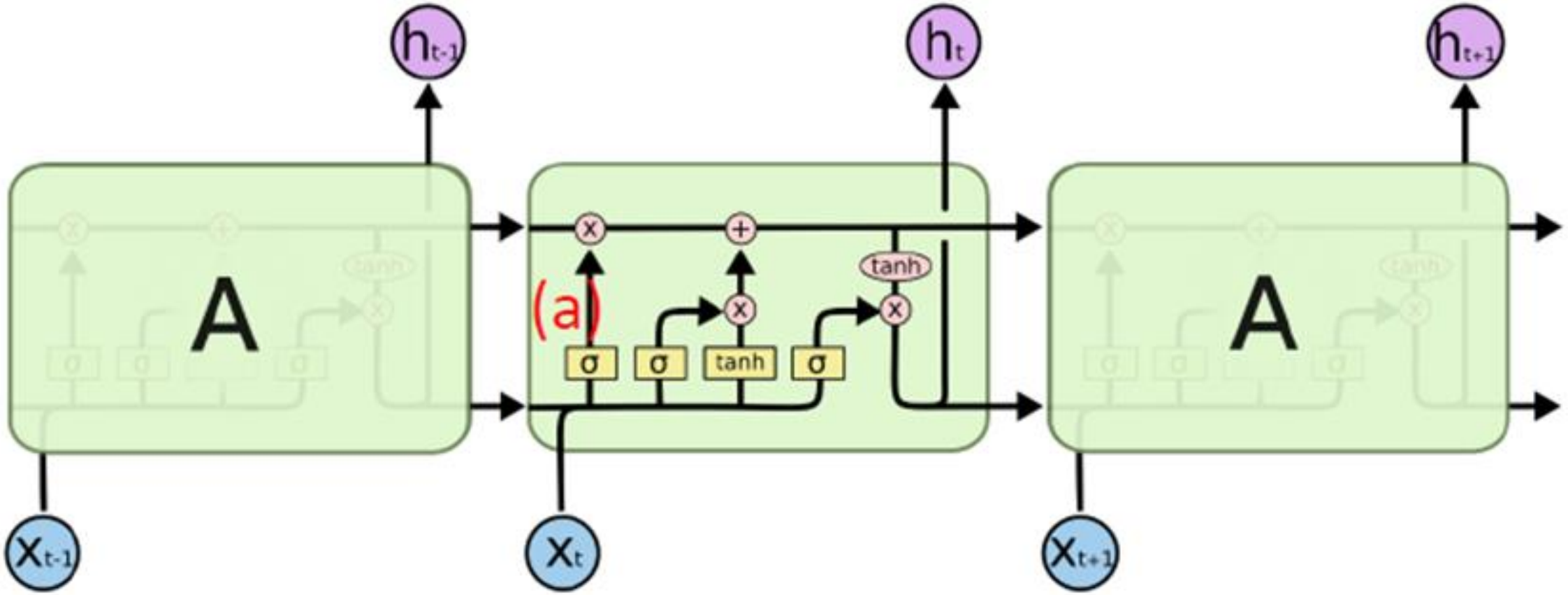
Predicting logP value of molecules

Review from Last Lecture
RNN

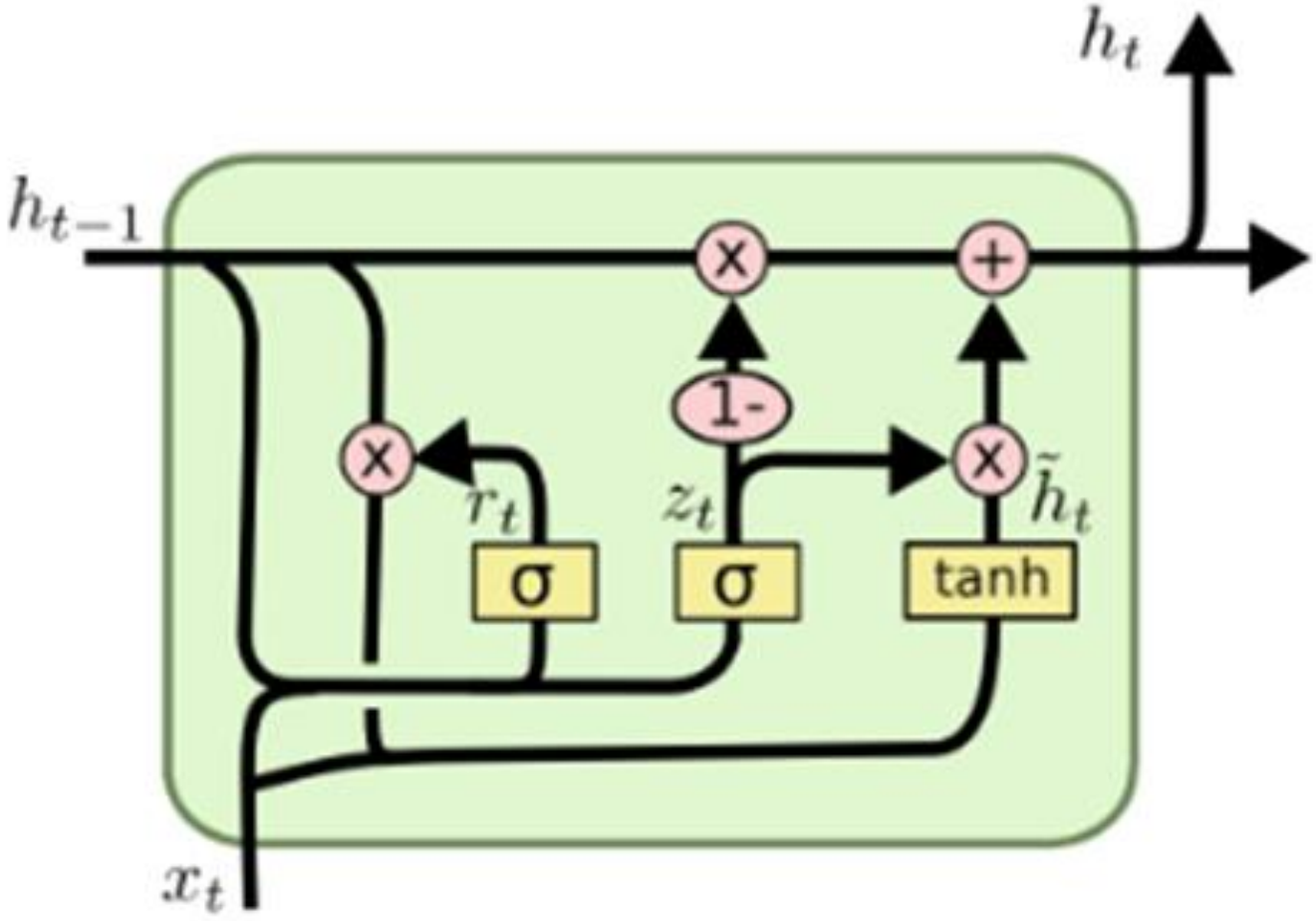
RNN – Gradient Vanishing



RNN – LSTM



RNN - GRU



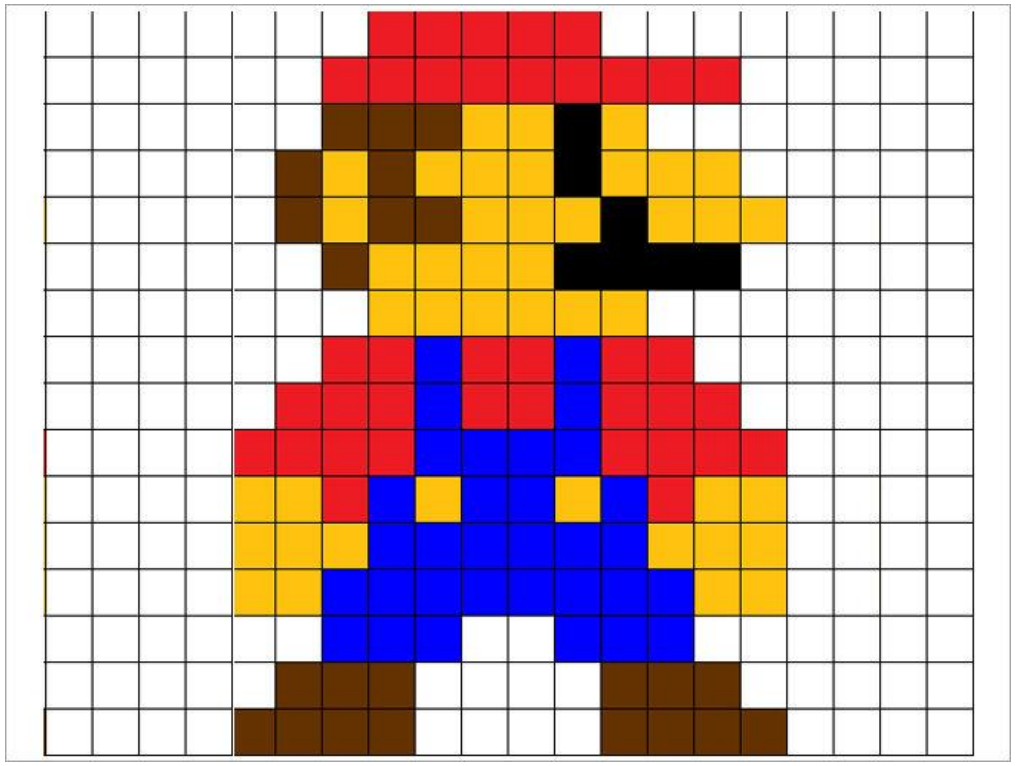
RNN – Performance

Arch.	N	N-dropout	P
Tanh	3.612	3.267	6.809
LSTM	3.492	3.403	6.866
LSTM-f	3.732	3.420	6.813
LSTM-i	3.426	**3.252**	6.856
LSTM-o	3.406	**3.253**	6.870
LSTM-b	3.419	3.345	6.820
GRU	3.410	3.427	6.876
MUT1	**3.254**	3.376	**6.792**
MUT2	3.372	3.429	6.852
MUT3	3.337	3.505	6.840

Table 2. Negative Log Likelihood on the music datasets. N stands for Nottingham, N-dropout stands for Nottingham with nonzero dropout, and P stands for Piano-Midi.

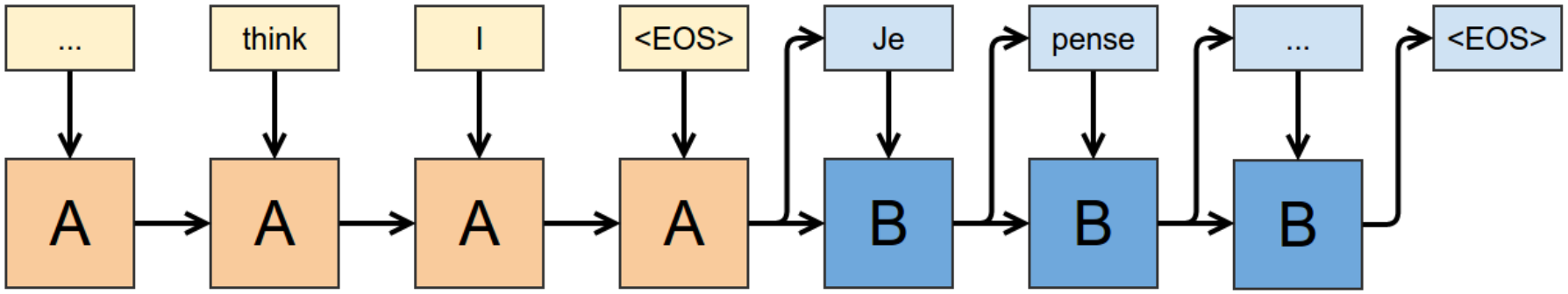
What is Graph?

Data Representation – Image and Sentence

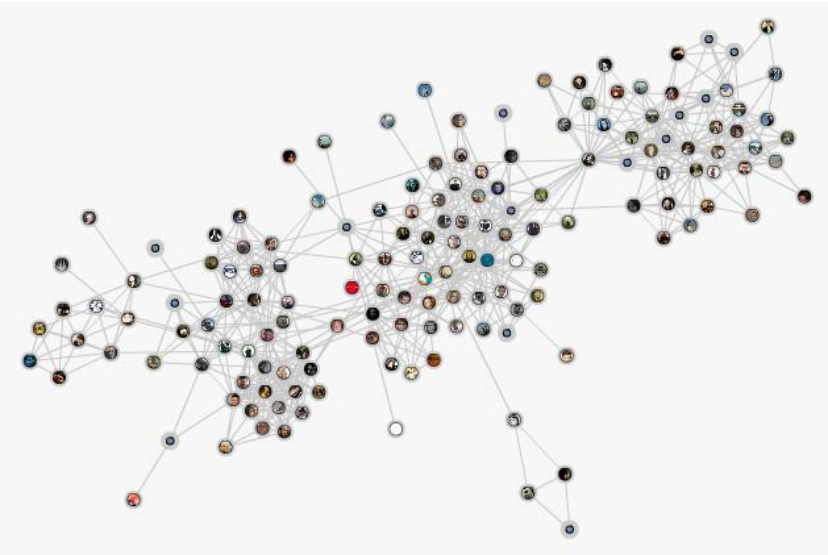


Images are represented with values on each pixel

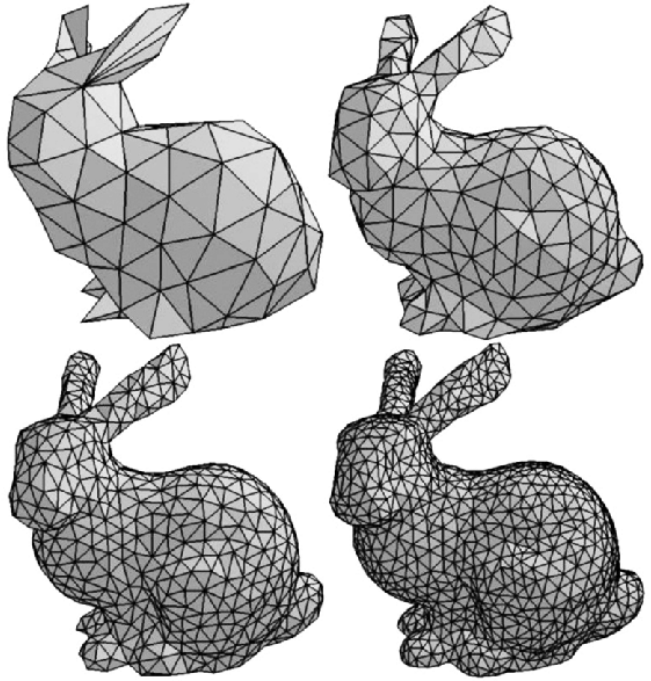
Sentences are represented with character values



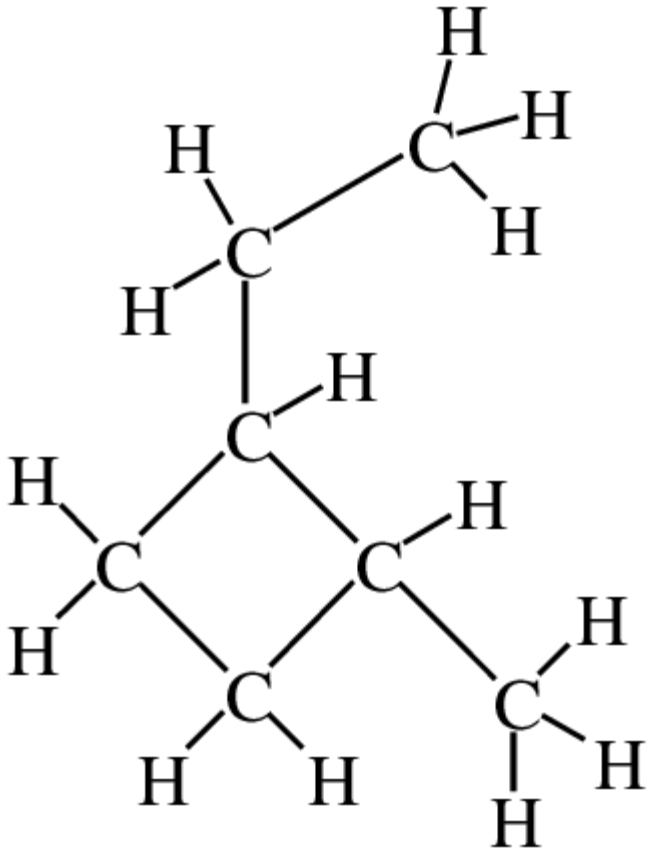
Data Representation – Graph



Social Graph

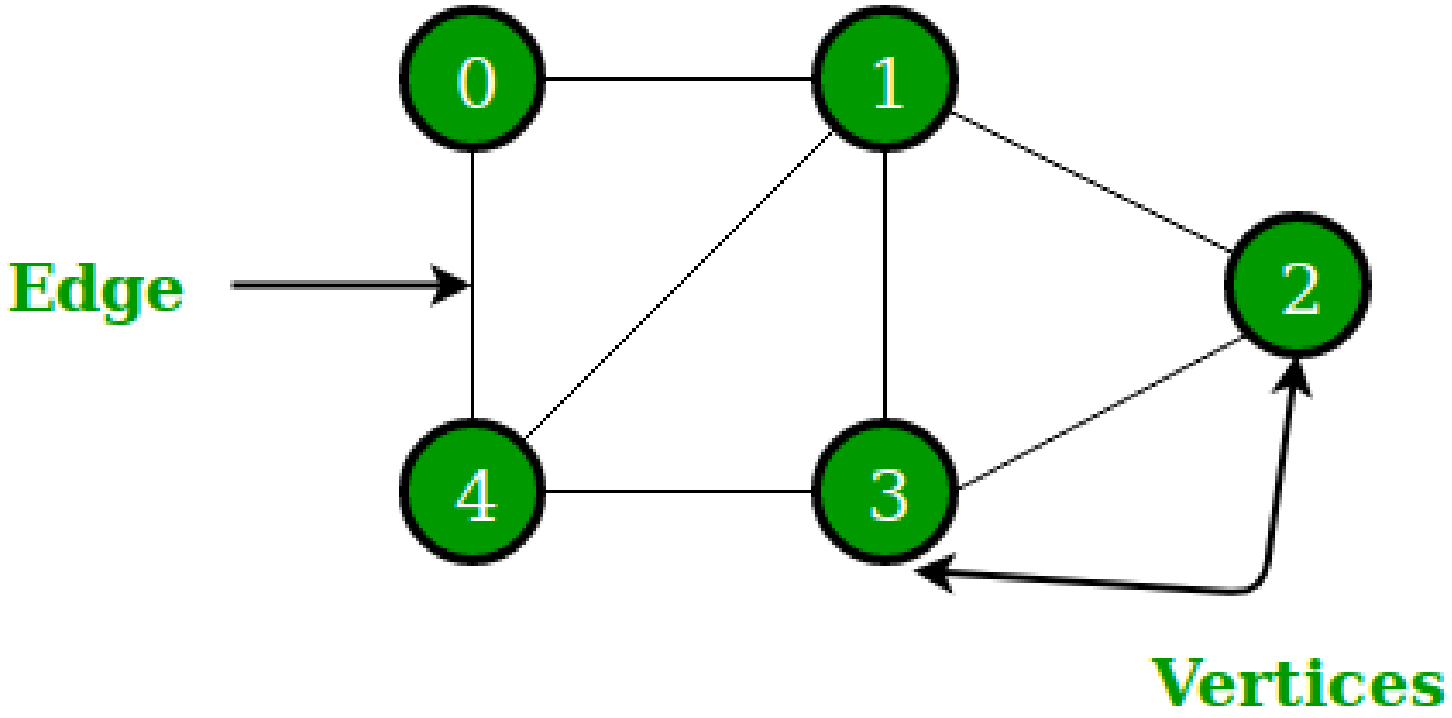


3D Mesh



Molecular Graph

Graph Structure



Vertex (Node)

Edge

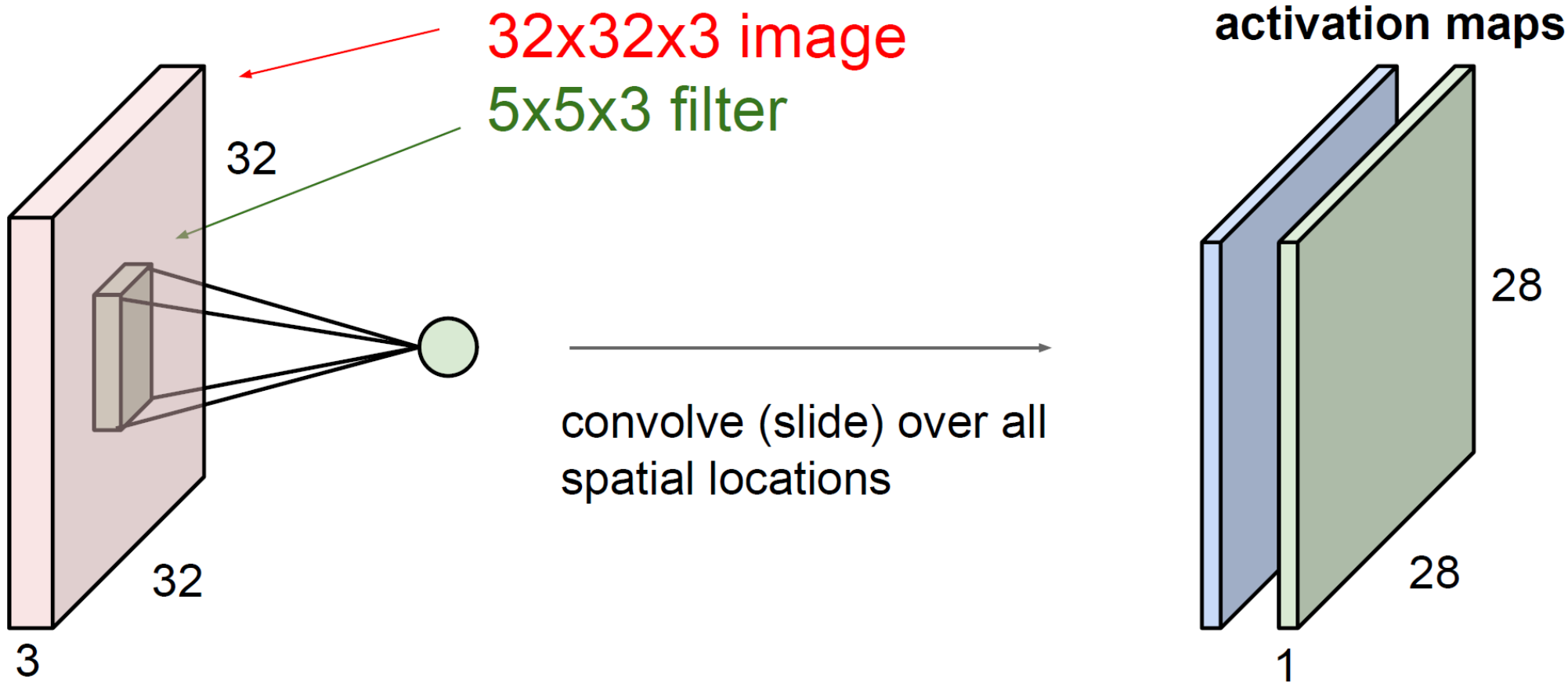
Graph Structure

Vertex (Node)
Node Feature Matrix

Edge
Adjacency Matrix

Graph Convolution Network

Convolution Layer
: Preserve the spatial structure



Weight sharing

Reduce the number of parameters
–〉 less overfitting, low computational cost

Learn local features

Translation invariance

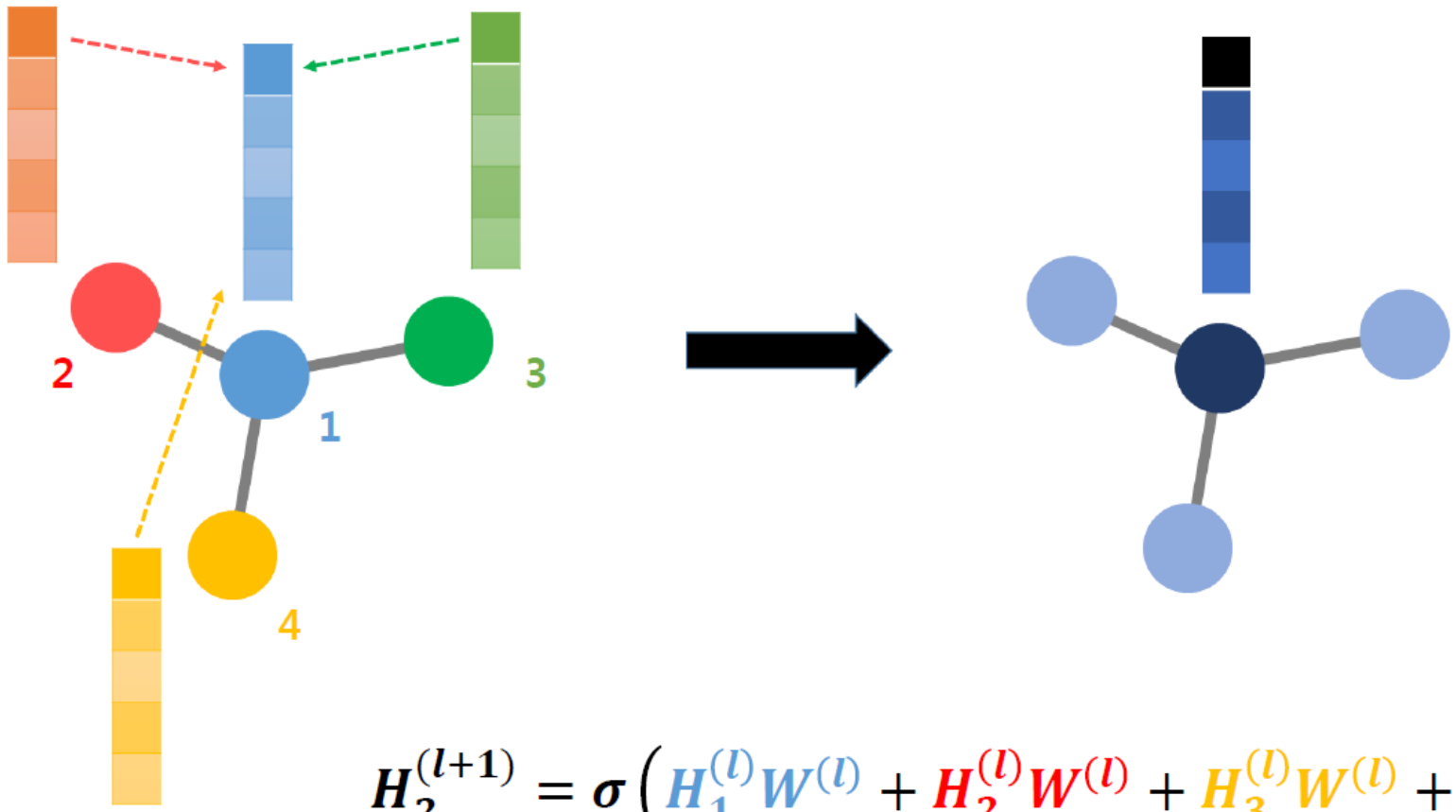
What should we update?

CNN updates values in activation map in each layer.
Values of activation map determine the state of image.

Values of each node feature determine the state of graph.

–〉 Make each layer of network update values of each node feature

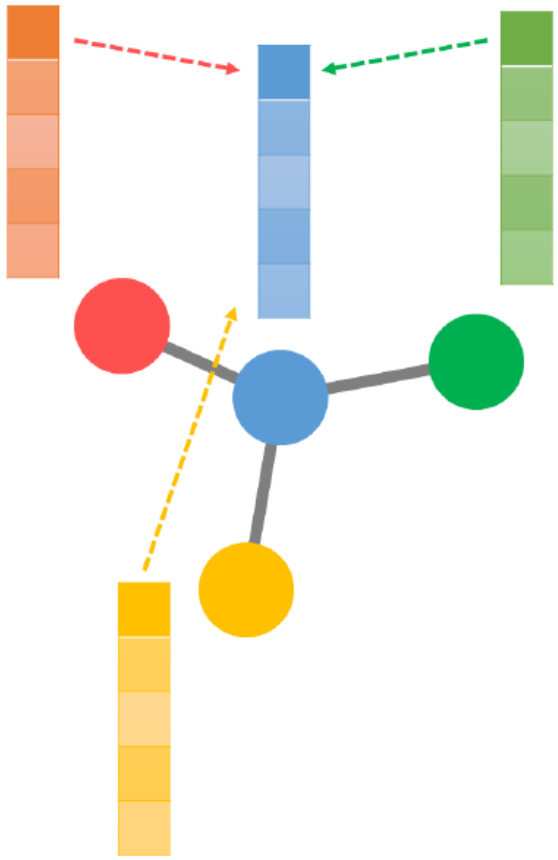
How to update hidden states in GCN

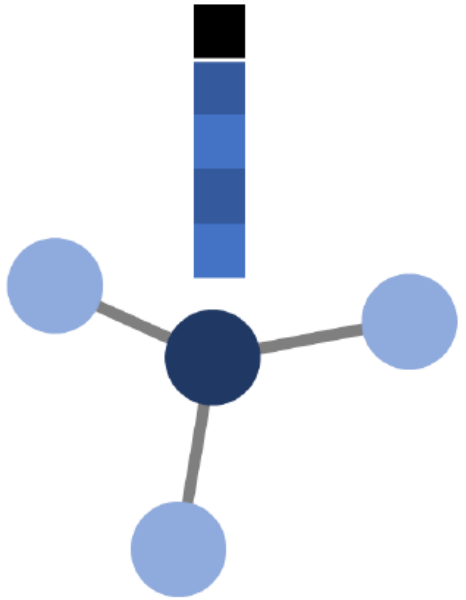


$$H_2^{(l+1)} = \sigma\left(H_1^{(l)}W^{(l)} + H_2^{(l)}W^{(l)} + H_3^{(l)}W^{(l)} + H_4^{(l)}W^{(l)} + b^{(l)}\right)$$

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

How to update hidden states in GCN

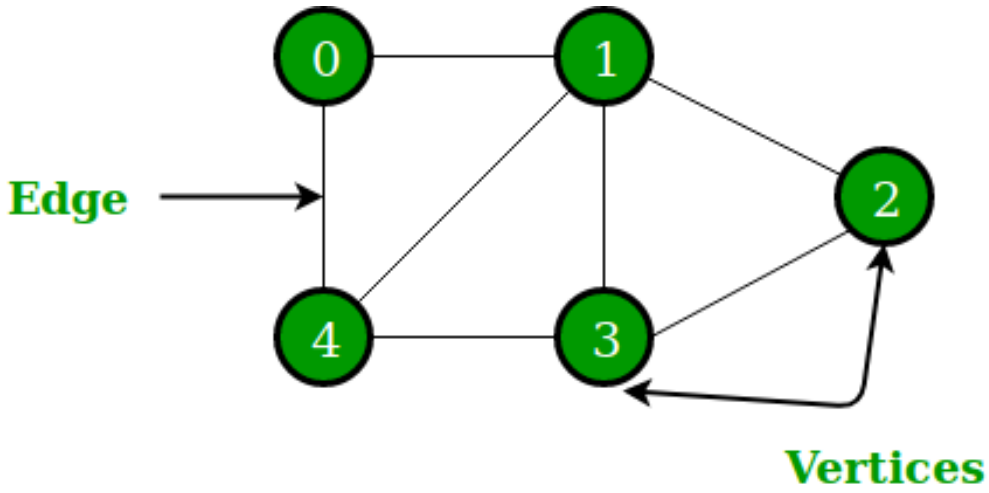




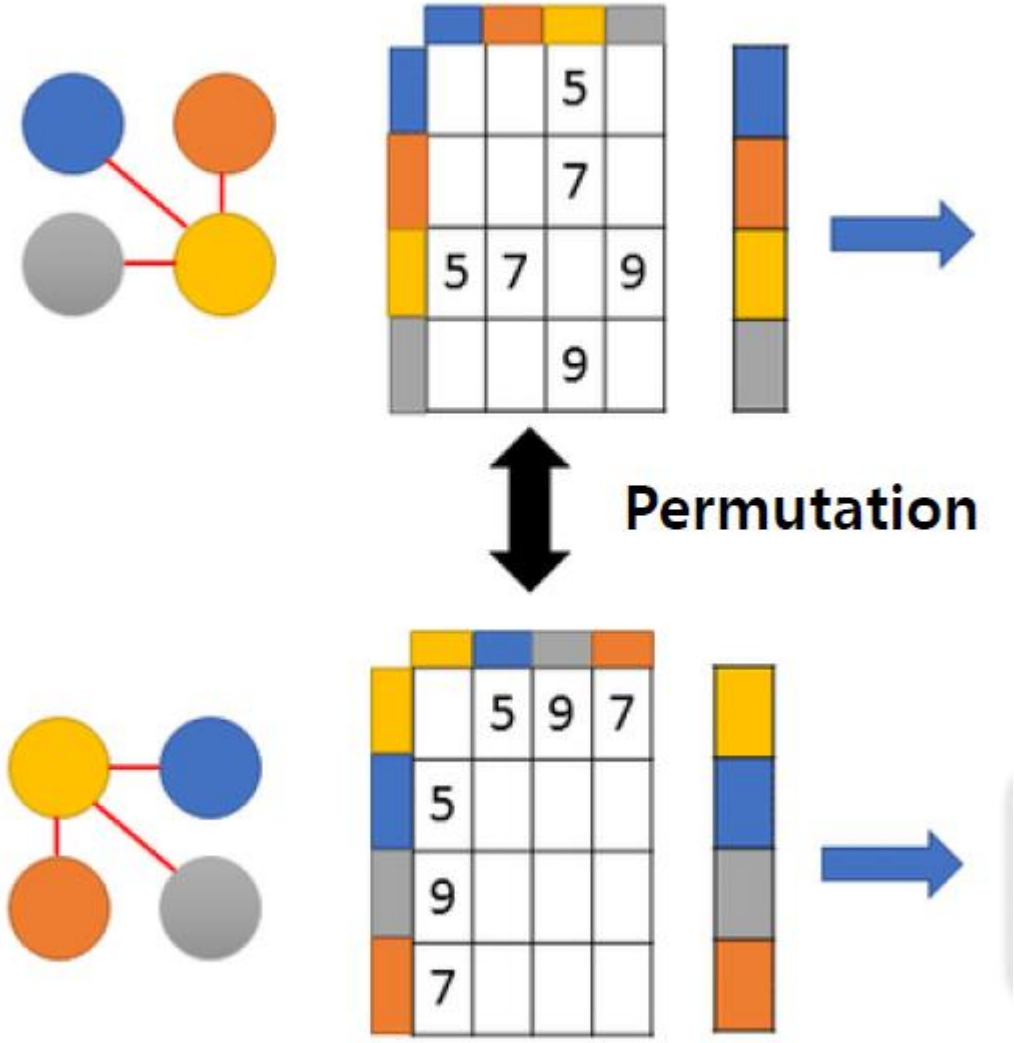
$$H^{(l+1)} = \sigma\left(AH^{(l)}W^{(l)} + b^{(l)}\right)$$

learnable parameters are shared

How to update hidden states in GCN



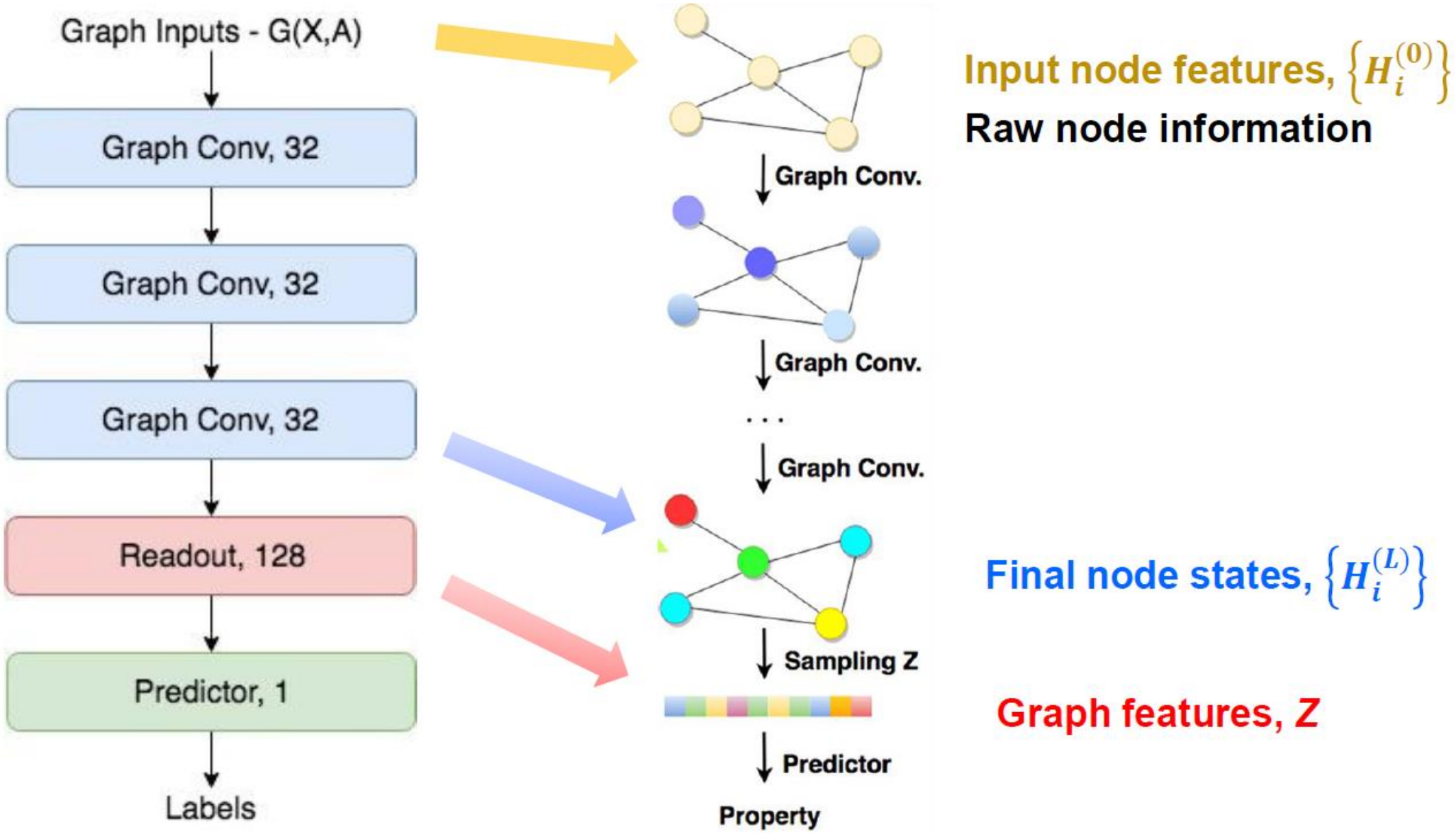
Readout – Permutation Invariance



Node-wise summation

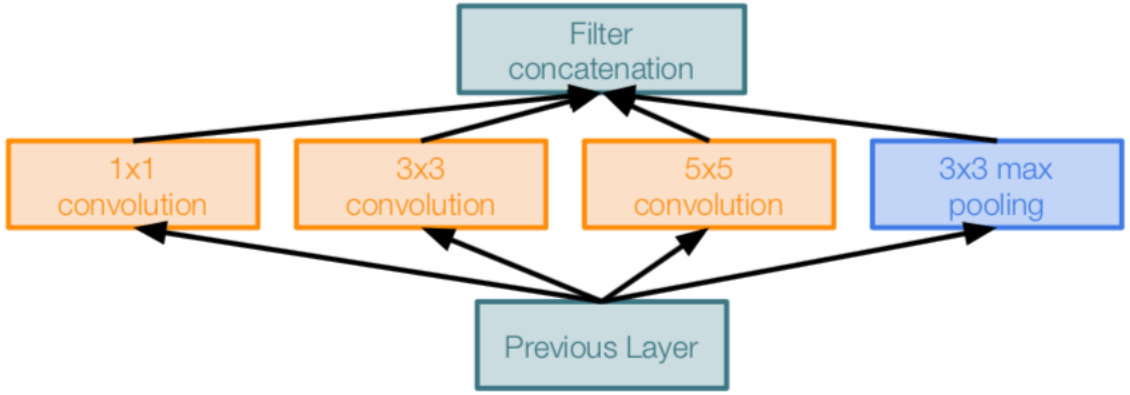
$$z_G = \tau\left(\sum_{i \in G} MLP\left(H_i^{(L)}\right)\right)$$

Overall Structure of GCN

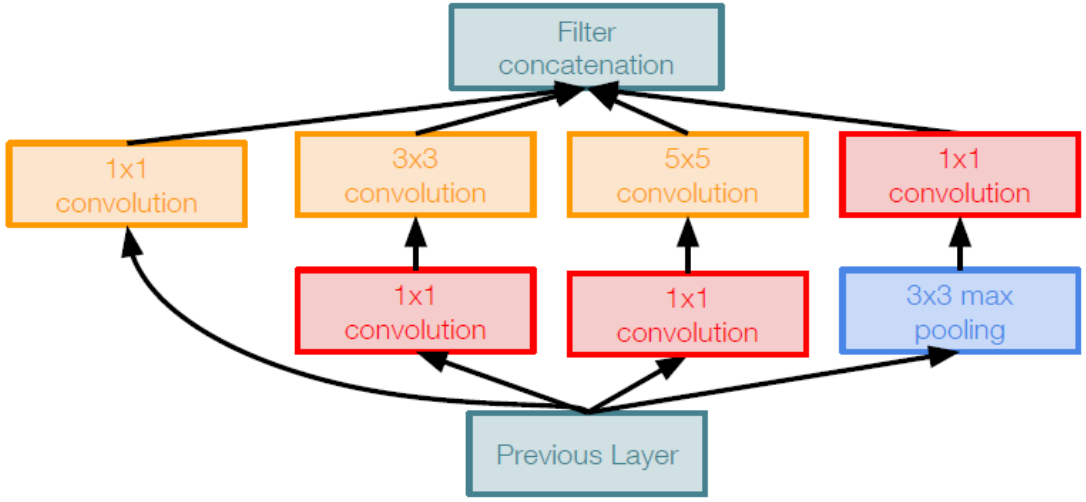


Advanced Techniques of GCN

Inception



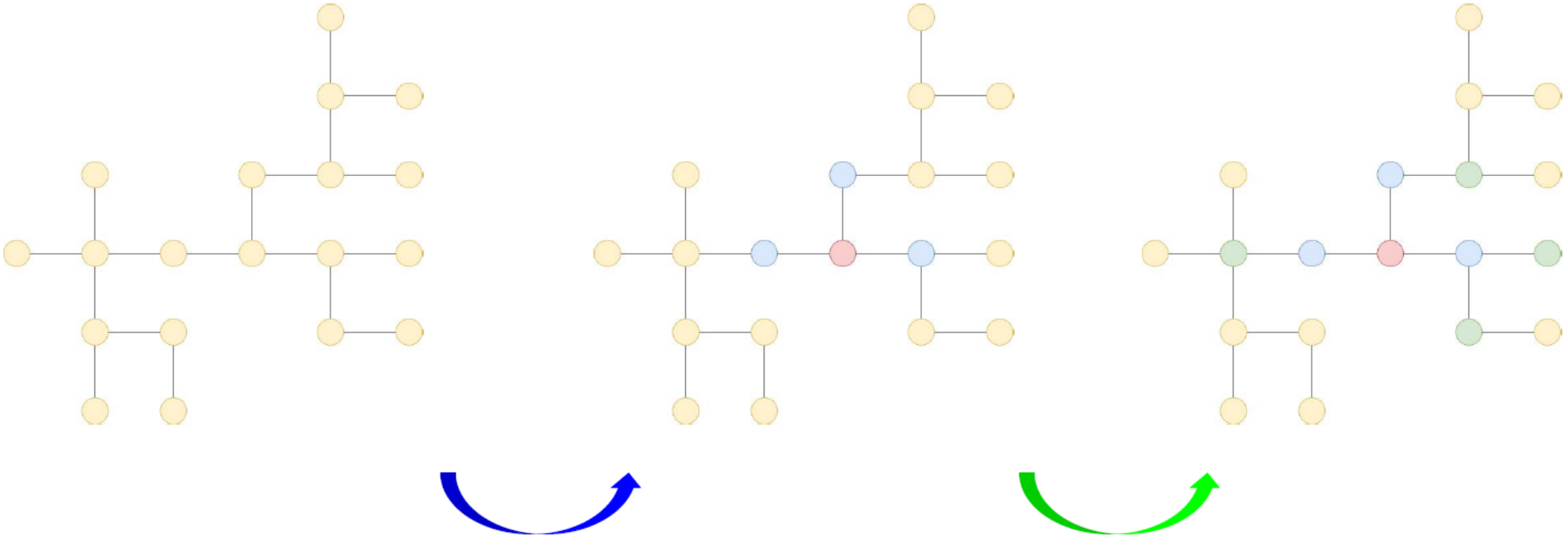
Naive Inception module



Inception module with dimension reduction

Inception

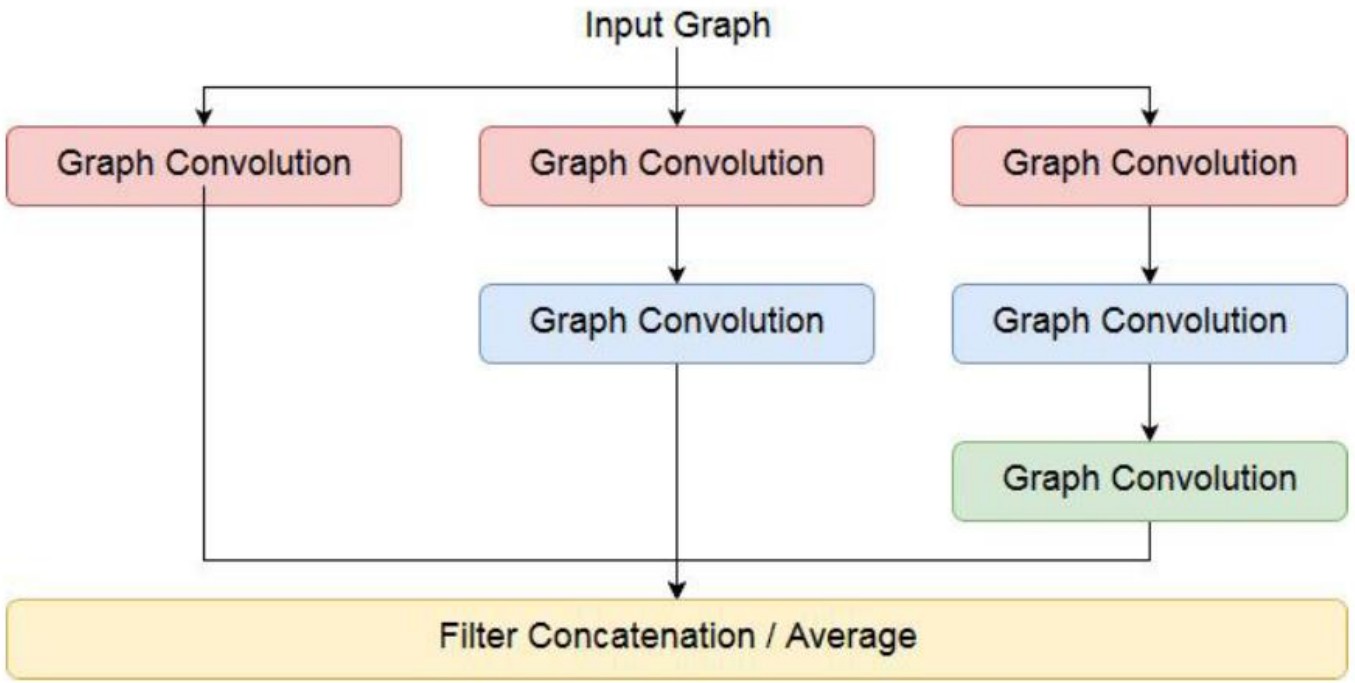
Single graph convolution reflects the first nearest neighbor information and subsequent **multiple graph convolution can deliver information of distant atoms.**



$$H^{(1)} = \sigma\left(AH^{(0)}W^{(0)}\right) \qquad H^{(2)} = \sigma\left(AH^{(1)}W^{(1)}\right)$$

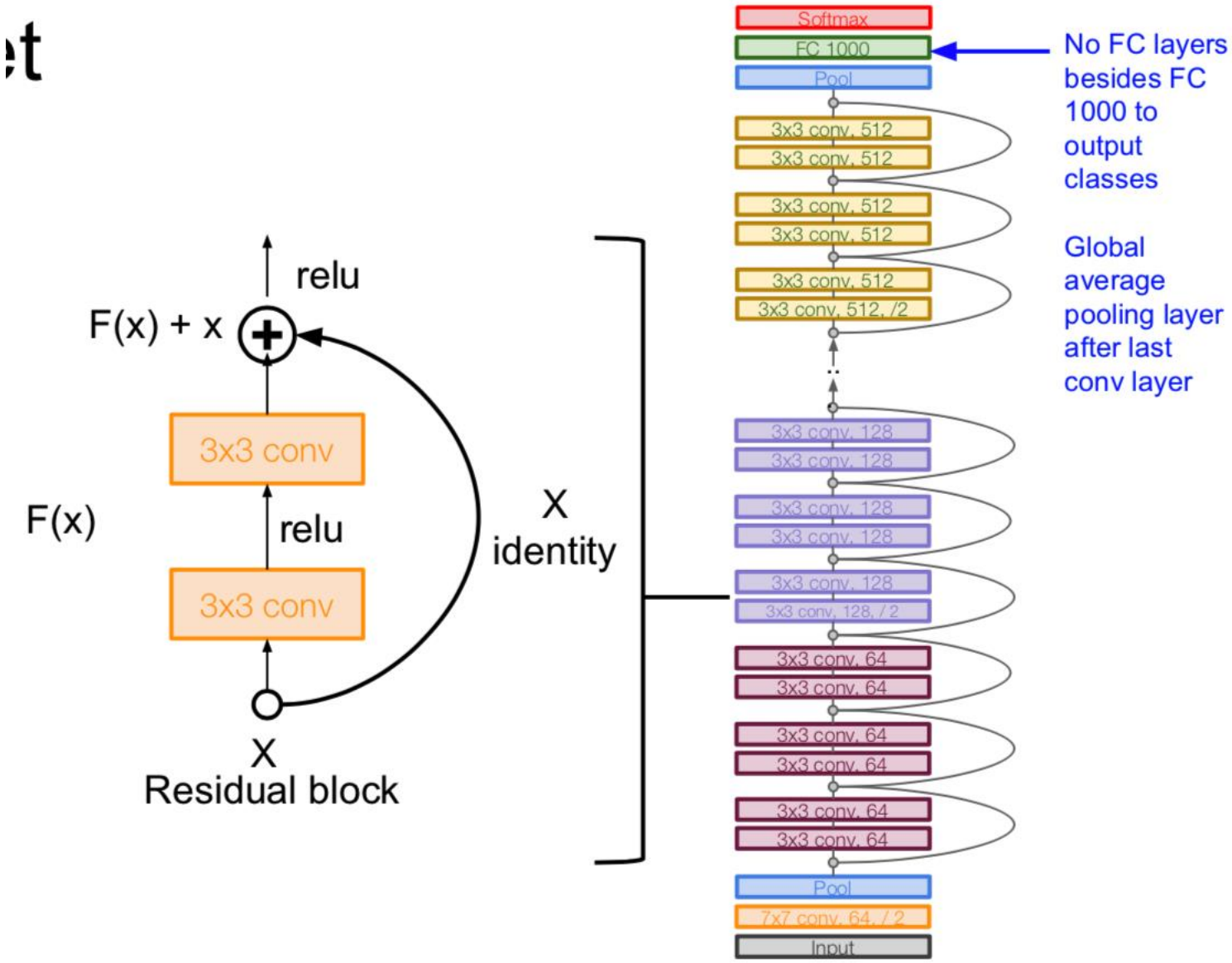
Inception

Inception module in GCN



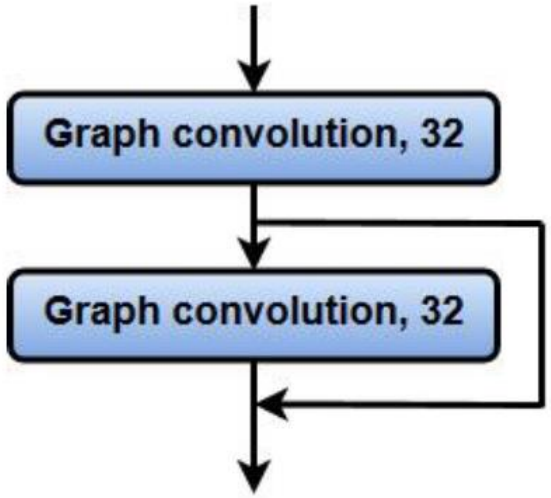
- Make network **wider**
- Avoid **vanishing gradient**

Skip Connection



Gated Skip Connection

However, naïve skip-connection unintentionally mix the information.

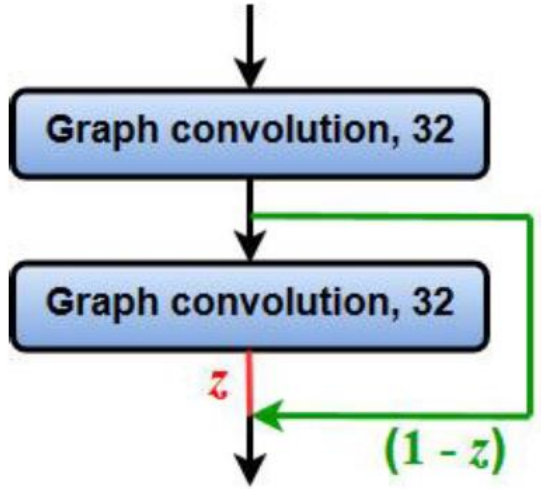


$$H_{i,sc}^{(l+1)} = H_i^{(l+1)} + H_i^{(l)}$$

Ryu, Seongok, Jaechang Lim, Seung Hwan Hong and Woo Youn Kim. "Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network." *arXiv preprint arXiv:1805.10988* (2018).

KAIST

Instead, one may use a gated-skip connection, which mixes the information with appropriate ratio, z.

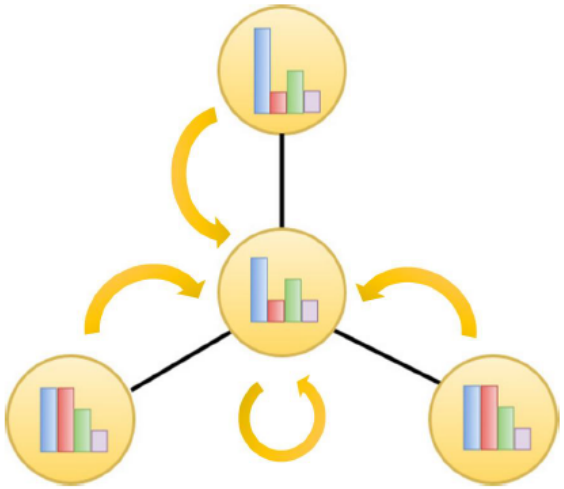


$$H_{i,gsc}^{(l+1)} = z_i \odot H_i^{(l+1)} + (1 - z_i) \odot H_i^{(l)}$$

$$z_i = \sigma\left(U_{z,1} H_i^{(l+1)} + U_{z,2} H_i^{(l)} + b_z\right)$$

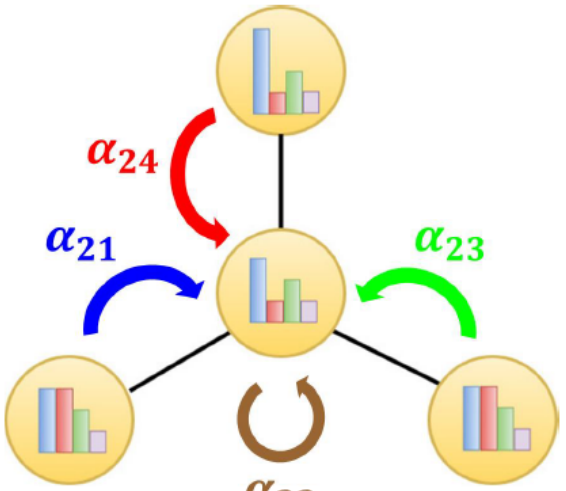
Attention

Vanilla GCN updates information of neighbor atoms **with same importance**.



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

Attention mechanism enables it to update nodes **with different importance**



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

KAIST

Ryu, Seongok, Jaechang Lim, Seung Hwan Hong and Woo Youn Kim. "Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network." *arXiv preprint arXiv:1805.10988* (2018).

Attention

Learnable parameters : Convolution weight and attention coefficient

$$H_i^{(l+1)} = \sigma\left(\sum_{j \in N(i)} \alpha_{ij}^{(l)} H_j^{(l)} W^{(l)}\right) \qquad \alpha_{ij} = f(H_i W, H_j W)$$

- Velickovic, Petar, et al. – network analysis

$$\alpha_{ij} = softmax(e_{ij}) = \frac{e_{ij}}{exp\left(\sum_{k \in N(i)} e_{ik}\right)} \qquad e_{ij} = LearkyReLU(a^T[H_i W, H_j W])$$

Velickovic, Petar, et al.
"Graph attention networks." *arXiv preprint arXiv:1710.10903* (2017).

- Seongok Ryu, et al. – molecular applications

$$\alpha_{ij} = \tanh\left((H_i W)^T C(H_j W)\right)$$

Ryu, Seongok, Jaechang Lim, Seung Hwan Hong and Woo Youn Kim.
"Deeply learning molecular structure-property relationships using attention- and gate-augmented graph convolutional network." *arXiv preprint arXiv:1805.10988* (2018).

KAIST