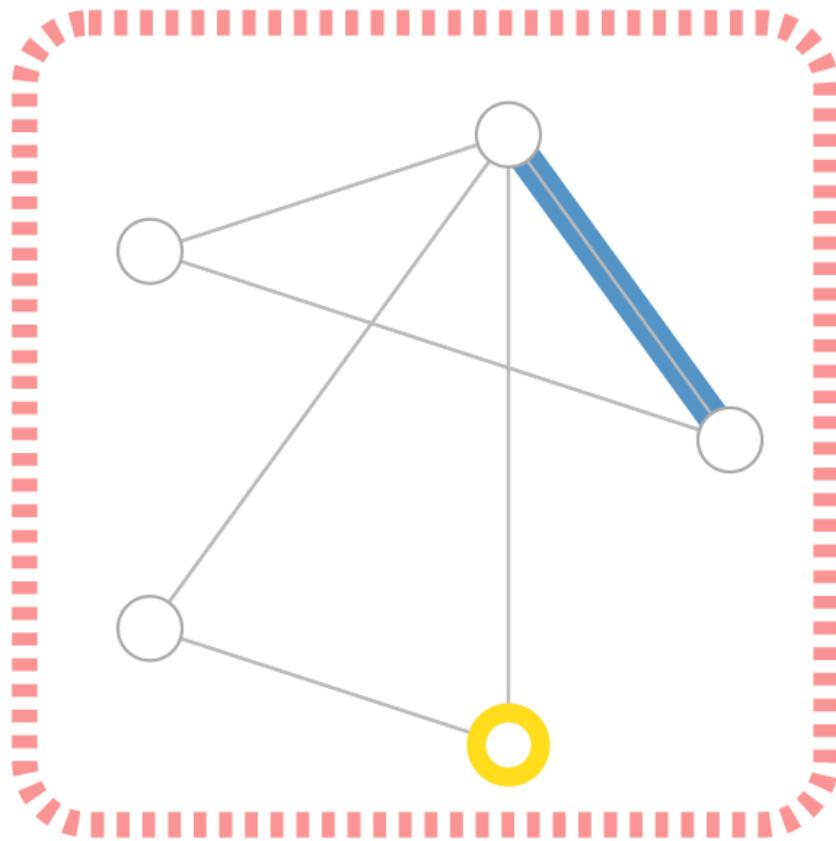


# Graph Neural Network

# Graph Network

- ▶ Distill
- ▶ 그래프에 포함되는 정보



Vertex (or node) embedding



Edge (or link) attributes and embedding



Global (or master node) embedding



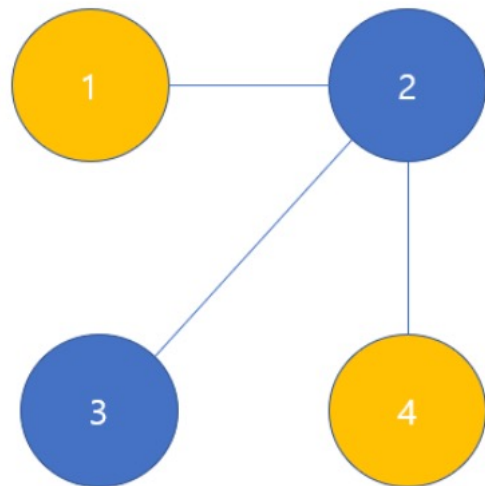
# Adjacency/Feature Matrix

- ▶ Adjacency matrix

- ▶ node 간의 관계(edge)를 표현하며 아래와 같이 방향성은 없는 경우 대칭적이다

- ▶ Feature matrix

- ▶ node에 담긴 정보를 나타내며, 아래의 경우 특성이 3개라고 가정함 (색을 구분)



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

$$\text{Feature matrix } F = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

# 그래프 예측 태스크

---

- ▶ graph-level
  - ▶ 그래프 전체의 특성 예측 (예: 분자의 냄새, 링 포함, 수용체 결합도 예측 등)
  - ▶ 이미지 분류나 문서의 감성예측과 유사한 작업
- ▶ node-level
  - ▶ 노드의 identity나 역할을 예측 (예: 각 노드가 어느 특정 노드와 가까운지 분류 예측)
  - ▶ 이미지 세그멘테이션(각 픽셀의 역할 예측), 문장에서 POS 예측과 유사한 작업
- ▶ edge-level
  - ▶ 노드간의 관계를 예측 (예: 이미지 객체간의 관계 기술)

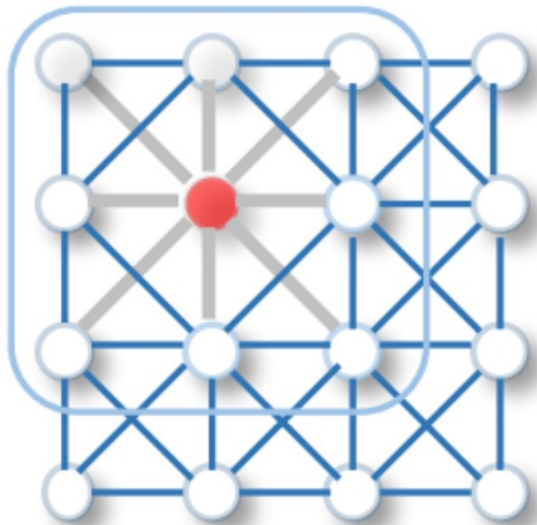
# 그래프 컨볼루션 모델(GCN)

---

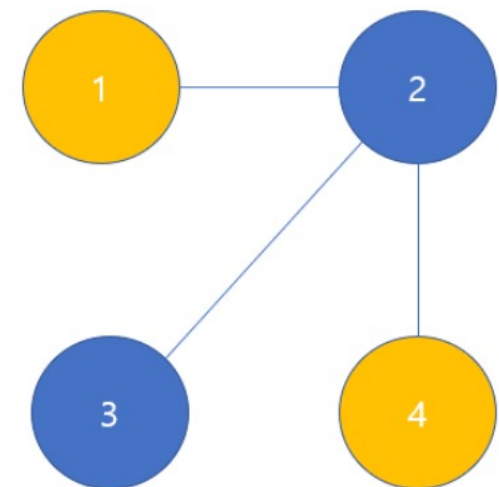
- ▶ CNN에서 커널 계수를 학습으로 찾아내듯이 분자 구조를 기술하는 계수를 학습으로 찾는다
  - ▶ 분자 그래프의 노드와 엣지를 벡터로 변환한다
- ▶ 다양한 변형
  - ▶ 그래프 컨볼루션 (GraphConvModel),
  - ▶ 위브 모델 (Weave model)
  - ▶ 메시지 전달 신경망 (MPNNModel),
  - ▶ 딥 텐서 신경망 (DTNNModel)
- ▶ 단점
  - ▶ 분자 그래프만 사용하므로 분자 구조에 대한 정보가 사라진다
    - ▶ 거대 분자에는 잘 동작하지 않는다

# GCN

- ▶ edge로 연결된 node끼리의 정보 교환을 표현하는 방법이 필요
  - ▶ 컨볼루션을 사용하는 방법을 채택
  - ▶ 주변 node의 정보를 spatial 하게 얻어 정보를 업데이트하는 방법
- ▶ 2D 컨볼루션과 그래프 컨볼루션의 차이

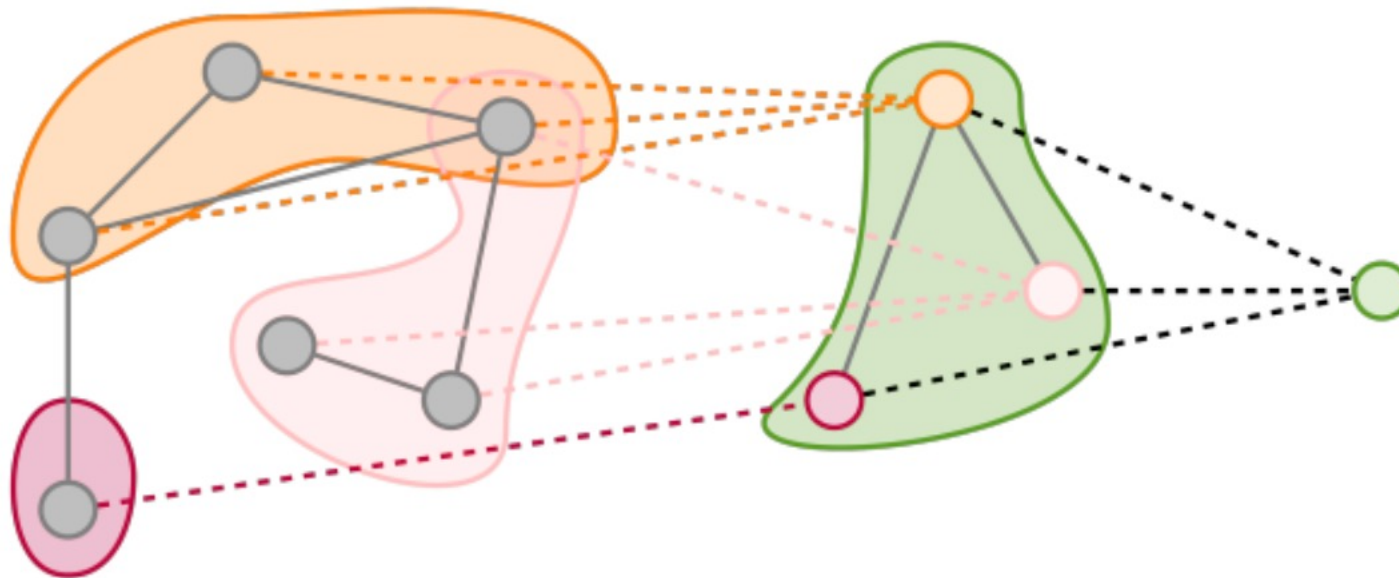


- ▶ (예) 2번 node를 업데이트할 때는 1,3,4 node를 사용하지만 다른 node는 2번 node만을 사용한다



# Graph pooling

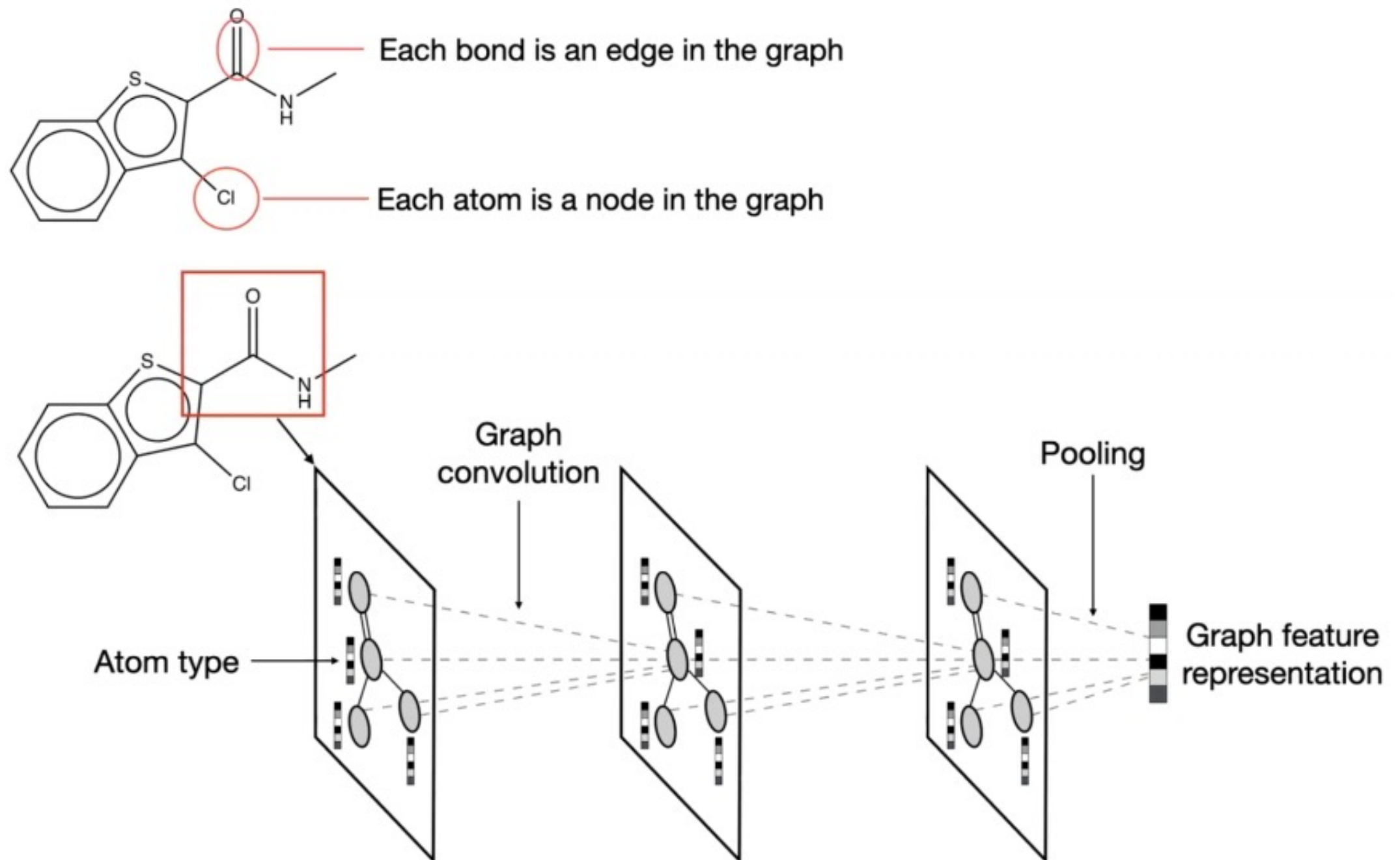
- ▶ 복잡한 구조의 graph structure를 단순하게 만드는 데 사용한다
  - ▶ feature에서 max값만 뽑거나 평균을 내어서 low dimension으로 만드는 작업을 수행



- ▶ edge나 node의 정보를 서로 전달하고자 할 때도 사용할 수 있다
  - ▶ Edge별로 연결된 node의 정보(features)를 모은다.
  - ▶ 정보(features)를 합쳐서 edge로 보낸다. (pooling)
  - ▶ Parameter를 이용해서 classification을 수행한다. (prediction)

# GCN

## ▶ 3D 분자의 구조적 정보를 반영





# Message passing

- ▶ node 혹은 edge 간의 연관성을 고려하면서 feature를 업데이트하는 방법
- ▶ (예) node를 주변 node 정보를 이용해서 업데이트하고 싶을 때 message passing은 다음과 같이 이루어진다
  - ▶ Edge로 연결되어 있는 node의 정보(features, messages)를 모은다
  - ▶ 모든 정보를 aggregate function (sum, average 등)을 이용하여 합친다
  - ▶ Update function(parameter)을 이용해서 새로운 정보로 업데이트한다
- ▶ Message passing을 여러번 반복하여 receptive field를 넓힐 수 있고 더 좋은 representation을 얻는다

