
Graph Convolutional Network



Pattern Recognition & Machine Learning Laboratory

Ha-na Jo

July 27, 2021



그래프

■ 그래프가 필요한 이유

➢ 유클리드 공간에 표현할 수 없는 정보가 존재

- 유클리드 공간 : 벡터 (좌표계) 의 형태로 표현 가능 (예 : 이미지, 문장)
- 그래프 사용 예

– 소셜 그래프, 3-D Mesh, 분자 구조, 관계, 상호 작용

■ 그래프의 구성

➢ 점들과 그 점을 잇는 선으로 이루어진 데이터 구조

➢ 노드

- 일반적으로 입력 데이터를 포함

➢ 엣지

- 일반적으로 데이터 간 관계 정보를 포함

– 방향 / 무방향 그래프 : 방향성 포함 유 / 무

– 가중 / 무가중 그래프 : 거리, 중요도 포함 유 / 무

➢ 인접 행렬

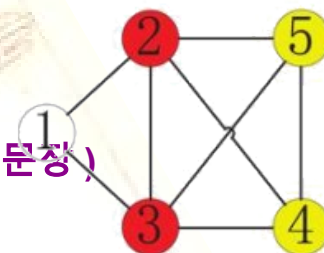
- 엣지의 연결을 표현하는 행렬 (), 노드 개수

– 0: 엣지 미연결, 1: 엣지 연결

➢ 특징 행렬

- 노드가 가진 특징을 표현하는 행렬 (), 특징 개수 (사용자 선택)

– 선정한 특징에 대한 각 노드가 갖는 값으로 구성



그래프 네트워크의 예
(같은 특징이면 같은 색으로 표현)

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

그래프 네트워크 예의 인접 행렬

	feature 1	feature 2	feature 3	feature 4	feature 5
node 1	1	1	1	0	0
node 2	1	1	1	1	1
node 3	1	1	1	1	1
node 4	0	1	1	1	1
node 5	0	1	1	1	1

그래프 네트워크 예의 특징 행렬
(특징이 5개인 경우)



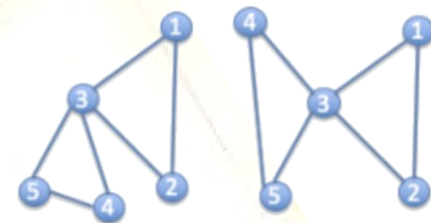
그래프 분석 방법

■ 그래프 분석이 어려운 이유

- 유클리드 공간으로 표현 불가능
- 고정되지 않은 형태
 - 노드와 엣지의 관계는 같고 형태가 다를 수 있다는 가능성 존재

■ 기존의 그래프 분석 방법

- 검색 알고리즘, 최단 경로 알고리즘, 클러스터링 방법 등
 - 입력 그래프에 대한 사전 지식 필요
 - 그래프 자체 연구 불가능, 그래프 레벨에서의 예측 불가능
 - » 그래프 레벨: 그래프에 속하는 점, 선의 정보가 아닌 여러 그래프 속 그래프의 정보 예측



그래프 형태 예시

■ 그래프 인공신경망 (Graph neural network)

- 그래프 레벨에서의 예측 작업에 활용
- 연결관계와 이웃들의 상태를 이용, 노드의 상태 업데이트, 노드 임베딩 출력
 - 노드 분류, 연관성 예측, 그래프 분류에서 활용
- 이웃과 자기 자신의 이전 임베딩에 곱하는 가중치의 값이 상이
- 이웃들의 임베딩 평균을 구하여 사용

■ 그래프 합성곱 신경망 (Graph convolutional network, GCN)

- 이웃과 자기 자신에 대해서 동일한 가중치 사용
- 이웃끼리 인접할 때 정규화를 적용하여 차등적으로 반영



GCN (1/2)

합성곱 인공신경망 (Convolutional neural network, CNN) 의 장점

➤ 가중치 공유 (Weight sharing)

- 파라미터 감소, 오버피팅 감소라는 장점

➤ 로컬 특징 학습 (Learn local feature)

- 하나의 노드가 담는 정보는 인접한 노드들의 지역적 정보 합축

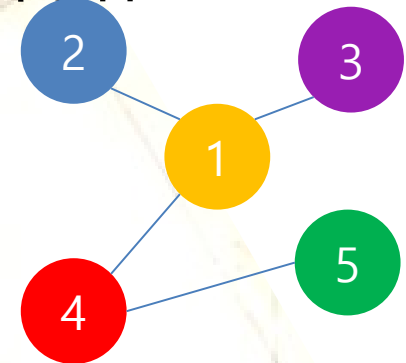
➤ 트랜스레이션 불변 (Translation invariance)

- 이미지의 변화가 생기더라도 결과에 미치는 영향 감소

GCN 으로 의 활용

➤ 레이어를 통과하면서 노드의 특징을 업데이트하며 추출하는 것이 목표

- : 활성화 함수, : 연결된 노드, : 은닉층, : 가중치, : 바이어스 (bias)
- CNN 의 장점 활용 : 같은 가중치 사용, 인접한 노드들의 정보 합축 사용
 - 모든 노드마다 연결된 노드를 확인하는 것은 비효율
- : 노드 개에 대한 인접 행렬 ()
- : 사용자가 지정한 특징의 개수 () 를 가진 특징 행렬 ()
- : 사용자가 지정한 개수 () 의 CNN 에서의 필터와 같은 역할 ()
 - : 필터를 거친 고차원 성분으로 이루어진 행렬
 - : 연결되어 있는 노드의 특징 값들의 합



그래프 네트워크 예시



GCN (2/2)

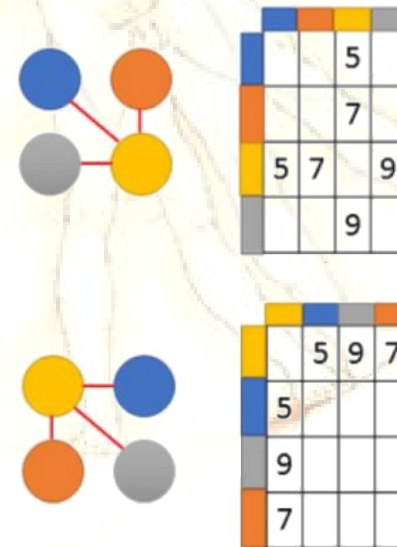
리드아웃 (Readout)

- 순서가 바뀐 그래프의 특징 행렬이 달라지는 현상 (치환) 을 방지
- 노드별 합계 (Node-wise summation)
 - 리드아웃 (Readout) 의 방법 중 하나

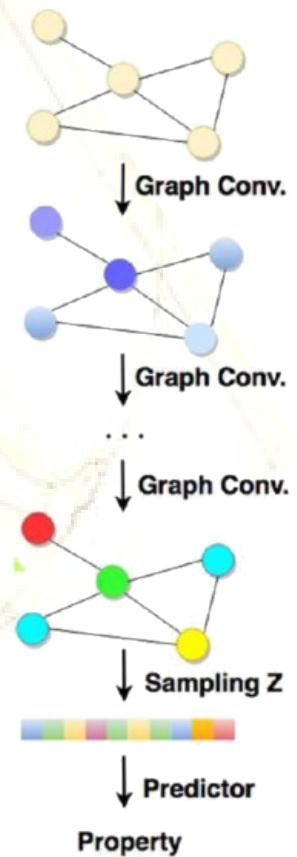
- : 활성화 함수, : 그래프 공간
- 은닉층에 다층 퍼셉트론 (Multi layer perceptron) 를 씌우고 더하는 방식
- 하나의 벡터 표현 가능
 - » 벡터를 더하면 순서가 바뀌어도 무관

GCN 의 구조

- 입력
 - 입력 그래프의 인접 행렬, 특징 행렬
- 그래프 합성곱
 - 전체에 대한 정보 획득
 - 고차원 정보 획득
- 리드아웃 (Readout)
 - 하나의 벡터로 표현
- 추정
 - 하나의 값 추정 혹은 여러 라벨의 확률 값 추정



치환의 예



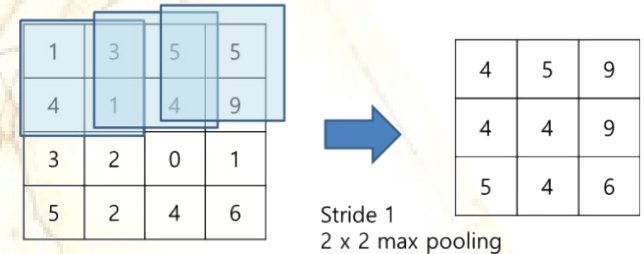
GCN 의 구조



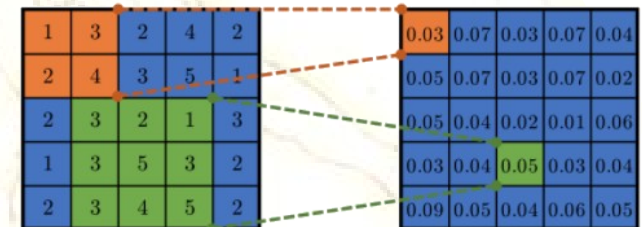
CNN 주요 모델 (1/2)

■ AlexNet

- 중첩 풀링 레이어 (Overlapping pooling layer)
 - 일반적으로는 보폭 (Stride) 을 여유롭게 하여 연산
 - 이 경우, 보폭을 좁혀 겹쳐서 풀링 (Pooling) 을 진행
 - 정확도 0.4 % 향상, 연산 속도는 감소
- 렐루 (ReLU*) 사용하여 속도 6 배 이상의 향상
- 로컬 응답 정규화 (Local response normalization) 사용
 - 주변 부분을 탐색 후 정규화하는 과정
 - 정확도 1.4 % 향상
 - 밝은 빛을 본 후 주변이 어두워 보이는 현상 원리 사용
- 드롭아웃 (Dropout) 사용
- 데이터 증가 (Data augmentation) 사용
 - 이미지를 변형하여 학습하여 오버피팅 방지



중첩 풀링 레이어의 예



로컬 응답 정규화 적용 전 (좌) 과 후 (우)

■ GoogLeNet

- 네트워크가 깊어지면 성능은 향상하지만 파라미터는 증가하는 문제 발생
- 적게 (Sparse) 연산하고, 조밀하게 (Dense) 출력하는 것이 목표
 - 일반적으로 적은 연산은 합성곱, 조밀한 연산은 완전 연결 레이어 (Fully connected layer)
 - 차원은 줄이면서 네트워크는 더 깊게 만드는 방향

*ReLU: rectified linear unit



CNN 주요 모델 (2/2)

➤ 인셉션 (Inception)

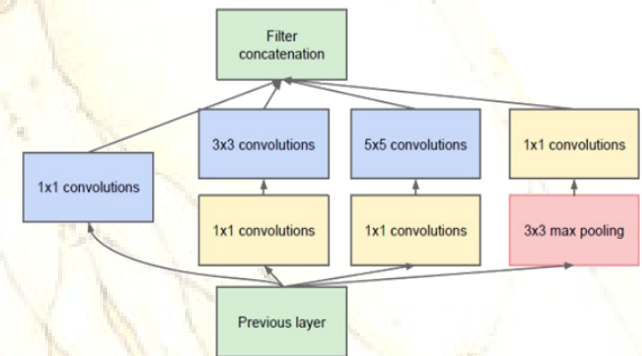
- 크기의 커널을 가지는 합성곱, 크기의 커널을 가진 최댓값 풀링 (Max pooling) 을 진행 후 합치는 과정
 - 넓은 네트워크, 다양한 특징 적용 가능, 연산량이 많아진다는 단점 발생
- 커널 크기를 가진 합성곱을 이용
 - 채널 수 조절 가능, 연산량 감소, 비선형성 증가

■ VGGNet

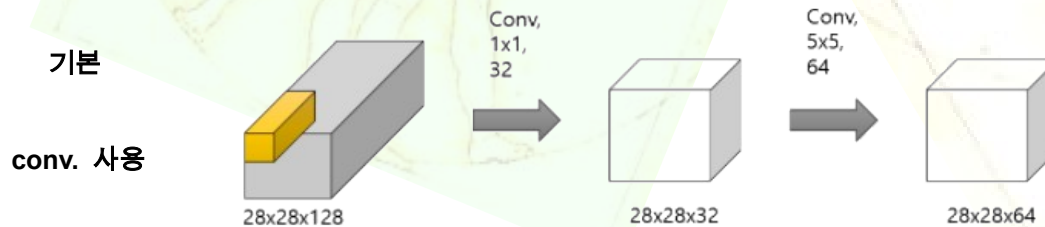
- 커널 크기의 필터 사용
 - 비선형성 증가, 파라미터 감소, 깊은 모델
- 커널 크기를 가진 합성곱을 이용

■ ResNet

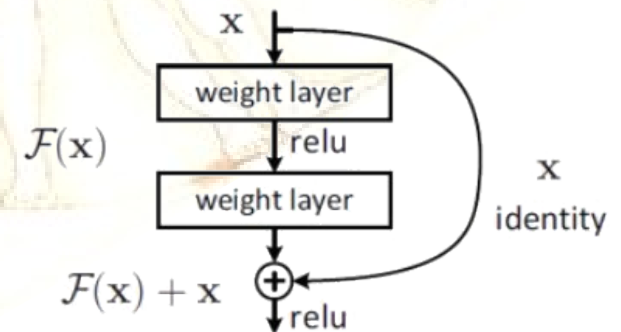
- VGGNet 의 깊은 모델에서 발생하는 오류 현상의 해결
- 스킵 커넥션 (Skip connection)
 - 레이어의 입력을 레이어의 출력에 바로 연결



인셉션 레이어의 형태



커널 크기를 가진 합성곱을 사용한 예



스킵 커넥션의 모습



GCN의 발전된 구조

■ 인셉션 (Inception)

➢ CNN에서의 핵심

- 다양한 커널 크기의 필터를 사용

➢ GCN에서의 적용

- 적용할 노드의 거리를 확장
- 인접 행렬의 곱셈으로 적용 가능

■ 스킵 커넥션 (Skip connection)

- : 그래프의 은닉층, : 스킵 커넥션

■ 게이트드 스킵 커넥션 (Gated-skip connection)

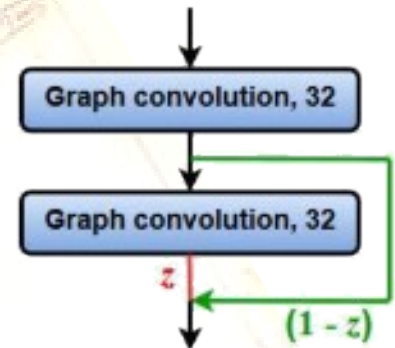
- : 그래프의 은닉층, : 게이트드 스킵 커넥션, : 가중치

➢ 스킵 커넥션을 적용할 두 정보의 적절한 가중치를 학습

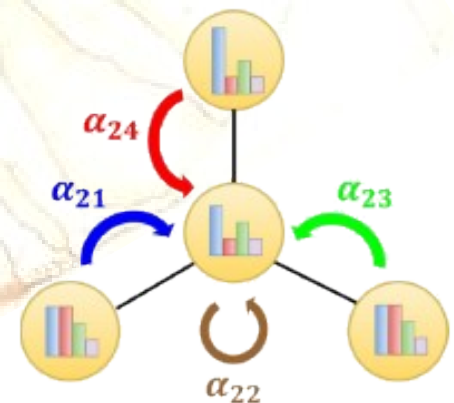
■ 어텐션 (Attention)

- : 활성화 함수, : 연결된 노드, : 가중치

➢ 하나의 노드를 업데이트할 때 각각의 노드의 가중치 () 를 학습



게이트드 스킵 커넥션의 모습



어텐션의 모습