

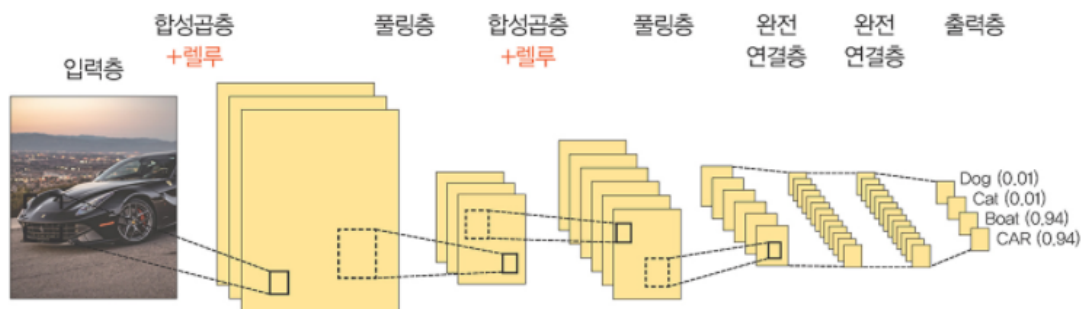


CH5

5.1 합성곱 신경망

이미지 전체를 한 번에 계산하는 것이 아닌 이미지의 국소적 부분을 계산함으로써 시간과 자원을 절약하여 이미지의 세밀한 부분까지 분석할 수 있는 신경망

- 이미지를 펼쳐서 분석하면 데이터의 공간적 구조를 무시하게 되므로 합성곱층을 사용
- 합성곱 신경망 : 음성/이미지/영상 인식에서 주로 사용되는 신경망. 다차원 배열 처리에 특화
- 다섯 개 계층으로 구성



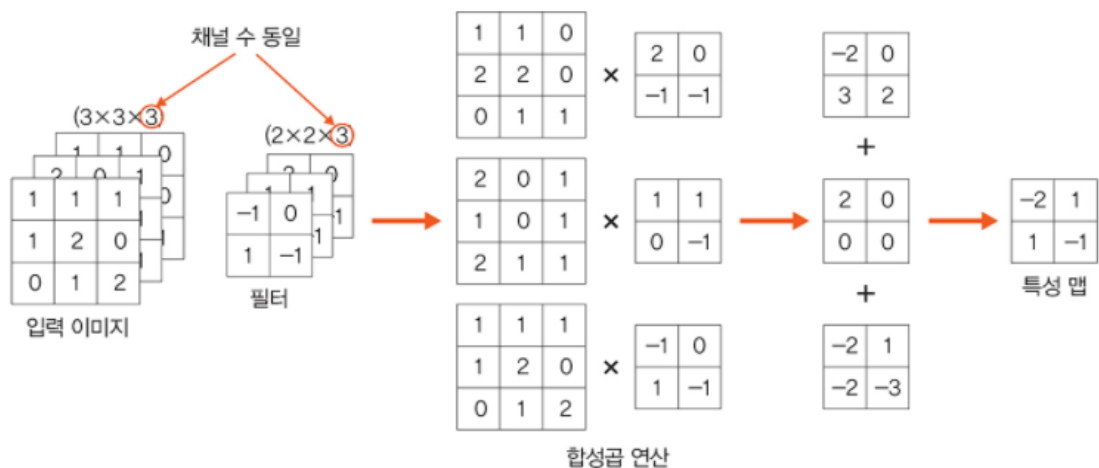
▲ 그림 5-2 합성곱 신경망 구조

1. 합성곱층과 풀링층을 거치며 입력 이미지의 주요 특성 벡터 추출
 2. 완전 연결층을 거치면서 1차원 벡터로 변환
 3. 출력층에서 활성화 함수인 소프트맥스를 사용하여 최종 결과 출력
-
1. 입력층 : 이미지는 높이,너비,채널 값을 갖는 3차원 데이터. (채널은 흑백이면 1, 컬러면 3)
 2. 합성곱층 : 커널/필터이 이미지의 모든 영역을 훑으면서 특성 추출 → 결과물을 **특성맵** 이라고 함.



▲ 그림 5-9 입력 이미지에 필터가 1만큼 마지막으로 이동

그레이스케일) 커널이 스트라이드 간격만큼 순회 → 모든 입력 값과의 합성곱 연산으로 새로운 특성맵 만들



▲ 그림 5-10 컬러 이미지 합성곱

컬러) 필터 채널이 3 & RGB 각각에 서로 다른 가중치로 합성곱을 적용하여 결과를 더 함

** 필터 개수 1개임

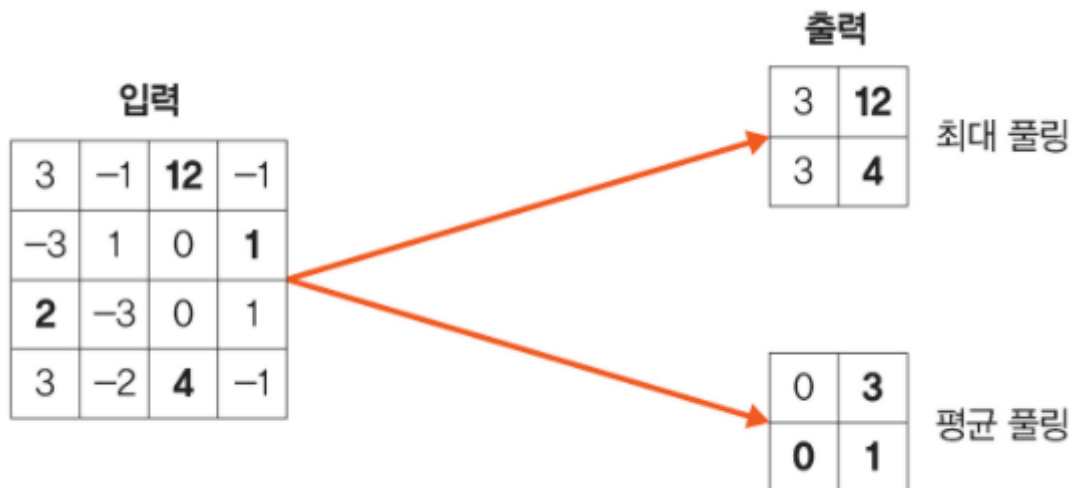
합성곱층 정리

- 입력 데이터: $W_1 \times H_1 \times D_1$
- 하이퍼파라미터
 - 필터 크기: F
 - 스트라이드: S
- 출력 데이터
 - $W_2 = (W_1 - F) / S + 1$
 - $H_2 = (H_1 - F) / S + 1$
 - $D_2 = D_1$

3. 풀링층 : 특성 맵 차원을 다운 샘플링하여 연산량 감소&주요 특성 벡터 추출을 통해 효과적 학습

4.

- 최대 풀링 : 대상 영역에서 최대값 추출
- 평균 풀링 : 대상 영역에서 평균 반환 → 중요한 가중치 갖는 값 특성이 희미해질 수 있어서 최대 풀링은 주로 사용

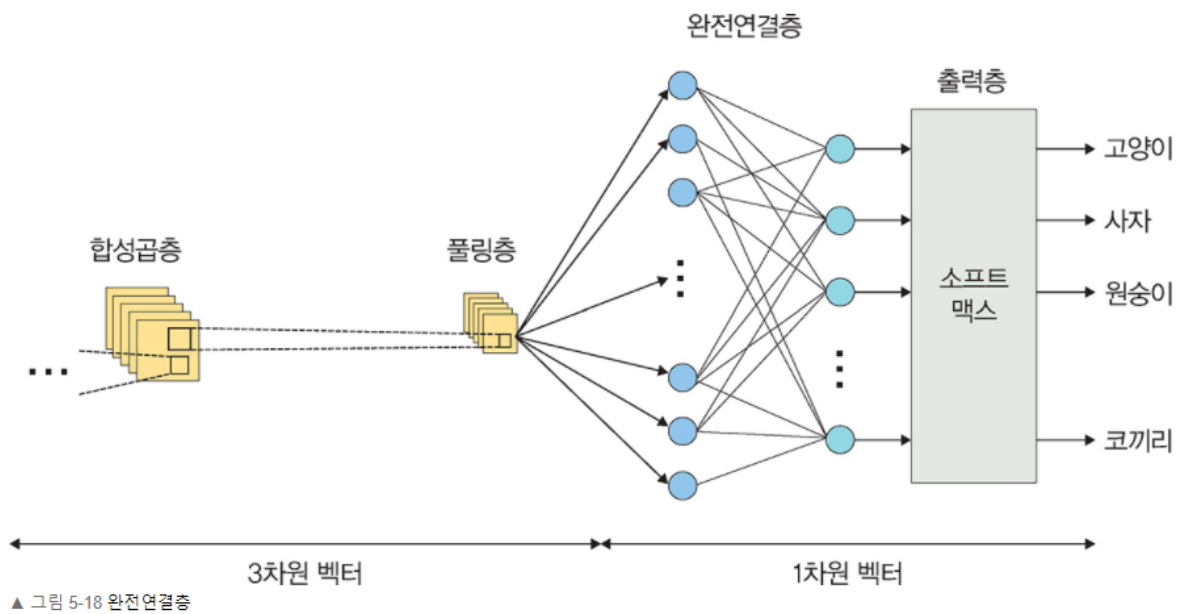


▲ 그림 5-17 최대 풀링과 평균 풀링 비교

풀링 정리

- 입력 데이터: $W_1 \times H_1 \times D_1$
- 하이퍼파라미터
 - 필터 크기: F
 - 스트라이드: S
- 출력 데이터
 - $W_2 = (W_1 - F) / S + 1$
 - $H_2 = (H_1 - F) / S + 1$
 - $D_2 = D_1$

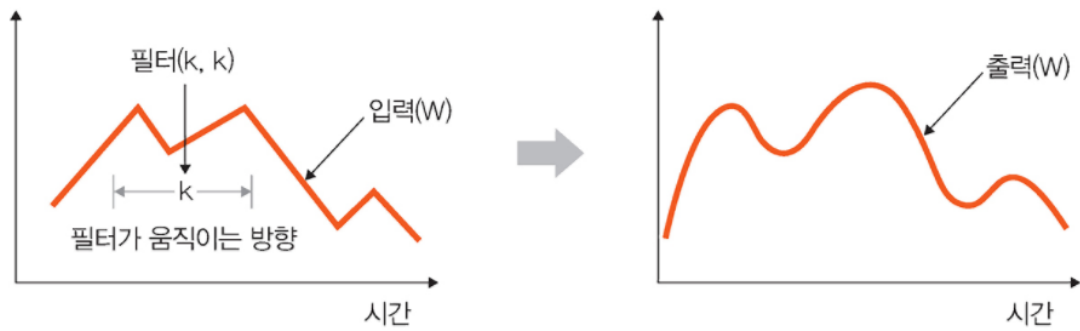
5. 완전연결층 : 1차원 벡터로 펼침(flatten)



- ## 6. 출력층 : 소프트맥스를 통해 0~1사이 값으로 출력, 레이블에 속할 확률 값을 출력하여 최대인 레이블이 최종 값이 됨

합성곱 종류

1. 1D

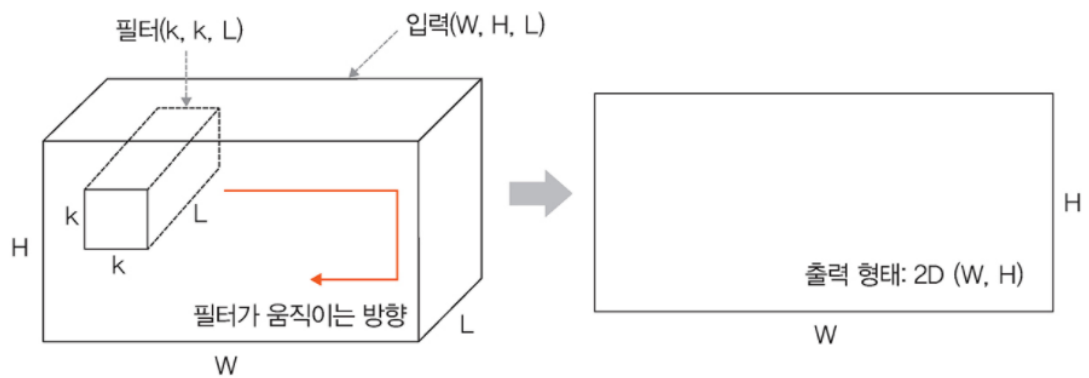


- 입력: W 너비(Width)
- 필터: $k \times k$ (높이 \times 너비)
- 출력: W 너비(Width)

▲ 그림 5-19 1D 합성곱

- 필터가 시간을 축으로 좌우만 이동

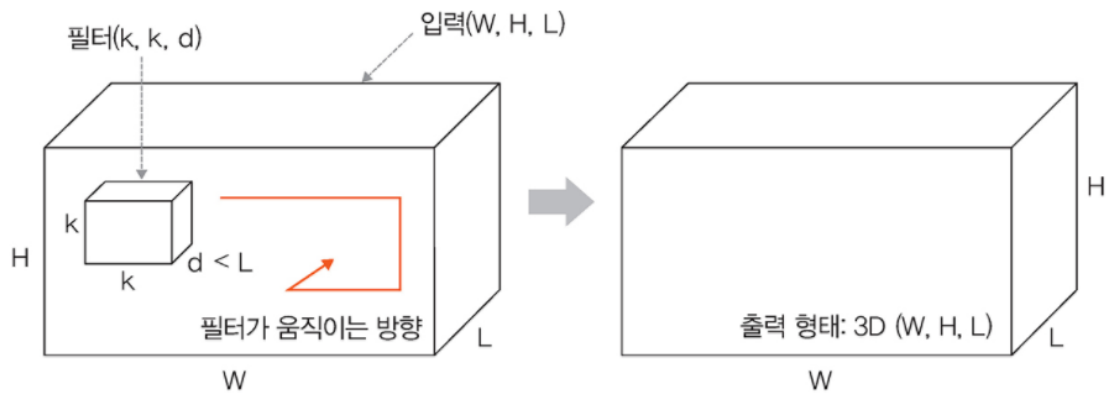
2. 2D



- 입력: W 너비(Width), H 높이(Height), L 길이(Length)
- 필터: $k \times k$ (높이 \times 너비), L 길이(Length)
- 출력: W 너비(Width), H 높이(Height)

▲ 그림 5-22 3D 입력을 갖는 2D 합성곱

3. 3D

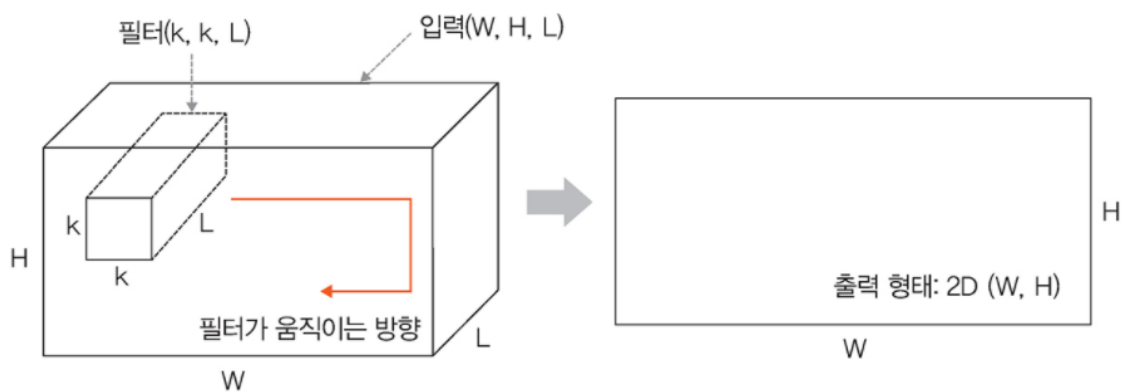


- 입력: W 너비(Width), H 높이(Height), L 길이(Length)
- 필터: $k \times k$ (높이 \times 너비), d: 깊이(depth)
- 출력: W 너비(Width), H 높이(Height), L 길이(Length)

▲ 그림 5-21 3D 합성곱

- $d < L$ 을 유지하는 것이 중요!

4. 3D 입력을 갖는 2D 합성곱

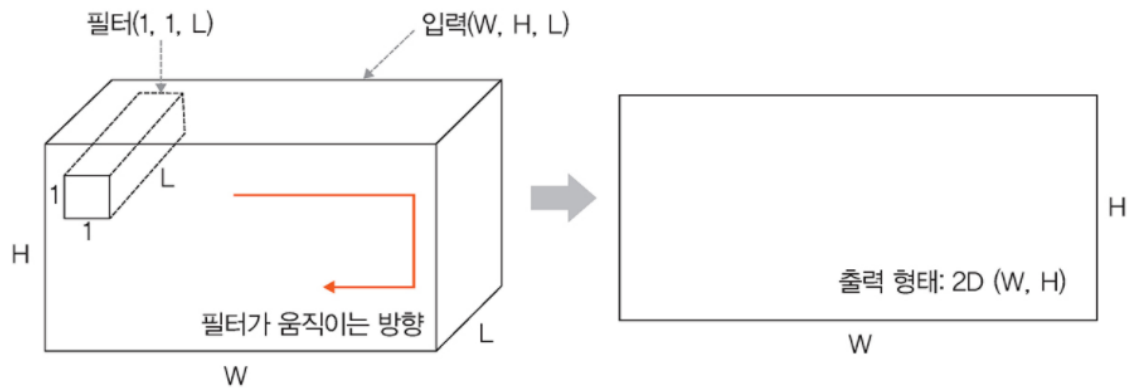


- 입력: W 너비(Width), H 높이(Height), L 길이(Length)
- 필터: $k \times k$ (높이 \times 너비), L 길이(Length)
- 출력: W 너비(Width), H 높이(Height)

▲ 그림 5-22 3D 입력을 갖는 2D 합성곱

- $(W, H, L) + (k, k, L) \rightarrow (W, H)$
- ex) LeNet-5, VGG

5. 1x1 합성곱



- 입력: W 너비(Width), H 높이(Height), L 길이(Length)
- 필터: 1×1(높이×너비), L 길이(Length)
- 출력: W 너비(Width), H 높이(Height)

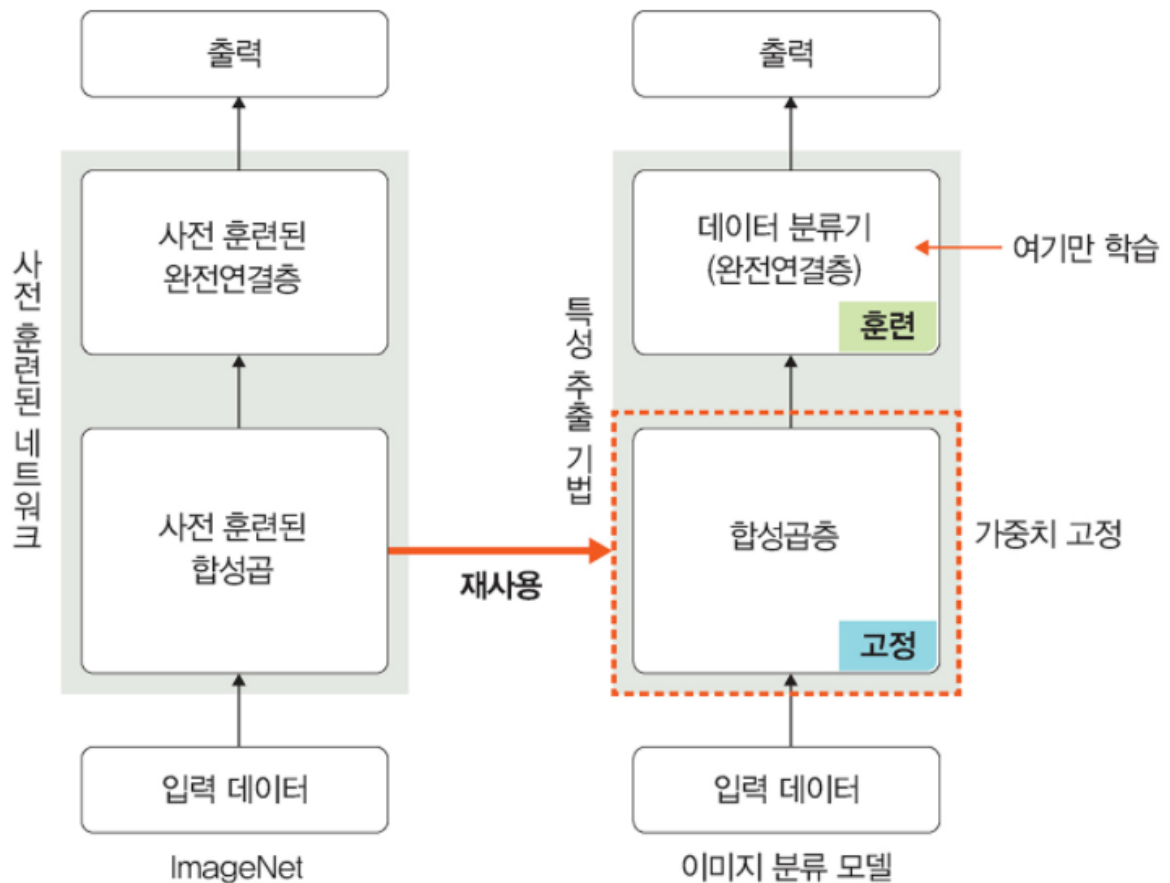
▲ 그림 5-23 1×1 합성곱

- 3D 형태 입력 (W,H,L) + (1, 1, L) → (W, H)
- 채널 수를 조정해서 연산량 감소 효과
- ex) GoogLeNet

5.3 전이 학습

이미지넷처럼 아주 큰 데이터셋을 써서 훈련된 모델(사전 훈련 모델)의 가중치를 가져와 해결하려는 과제에 맞게 보정해서 사용하는 학습

1) 특성 추출 기법



▲ 그림 5-30 특성 추출 기법

- ImageNet 데이터셋으로 사전 훈련된 모델을 가져온 후 마지막 완전연결층 부분만 새로 만들 → 마지막 완전연결층(데이터분류기= 추출 특성을 입력받아 최종 이미지 클래스분류)만 학습하고 나머지 계층들은 학습되지 않도록 함
- ex) Xception, Inception V3, ResNet50, VGG16/19, Mobile Net

2) 미세 조정 기법

- 사전 훈련 모델과 합성곱층, 데이터 분류기의 가중치를 업데이트하여 훈련
- 특성이 잘못 추출되었다면 미세 조정 기법으로 새 이미지 데이터를 이용하여 네트워크 가중치 업데이트하여 다시 추출
- 사전 훈련된 네트워크를 미세 조정하여 분석하려는 데이터셋에 맞도록 파라미터 조정

케이스	전략
데이터셋 크고 사전훈련 모델 유사성 작음	모델 전체 재학습
데이터셋 크고 사전훈련 모델 유사성 큼	합성곱층 뒷부분과 데이터 분류기 학습
데이터셋 작고 사전훈련 모델 유사성 작음	합성곱층 일부분과 데이터 분류기 학습
데이터셋 작고 사전훈련 모델 유사성 큼	데이터 분류기만 학습

5.4 설명 가능한 CNN

딥러닝 처리 결과를 사람이 이해할 수 있는 방식으로 제시

1) 특성 맵 시각화

특성 이미지에 대한 특성 맵을 시각화 = 특성 맵에서 입력 특성을 감지하게 도움

5.5 그래프 합성곱 네트워크

그래프 : 방향성이 있거나 없는 edge에서 연결된 노드 집합

노드 : 원소 / 에지 : 두 노드 연결한 선

1) 인접 행렬

- 컴퓨터가 이해하기 쉽게 그래프로 표현
- 노드 n 개를 $n \times n$ 행렬로 표현하고 ' A_{ij} 는 i 와 j 의 관련성 여부'를 만족하는 값으로 채움

2) 특성 행렬

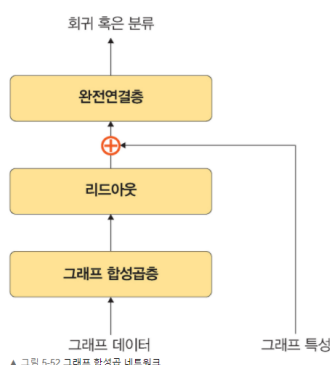
2단계. 특성 행렬(feature matrix)

- 인접 행렬만으로는 특성을 파악하기 어렵기 때문에 단위 행렬을 적용합니다.
- 각 입력 데이터에서 이용할 특성을 선택합니다.
- 특성 행렬에서 각 행은 선택된 특성에 대해 각 노드가 갖는 값을 의미합니다(예 첫 번째 행은 첫 번째 노드의 특성 값).
- 이제 노드 1-2-3-4에 대한 특성을 한눈에 파악하기 쉽게 표현되었습니다.



▲ 그림 5-51 특성 행렬⁵

그래프 합성곱 그래프(GCN)



▲ 그림 5-52 그래프 합성곱 네트워크

- 리드 아웃 : 특성 행렬을 하나의 벡터로 변환하는 함수
→ 전체 노드 특성 벡터 평균 구하고 그래프 전체 표현하는 벡터 생성
- ex) SNS 관계 네트워크, 학술 연구 인용 네트워크, 3D Mesh

