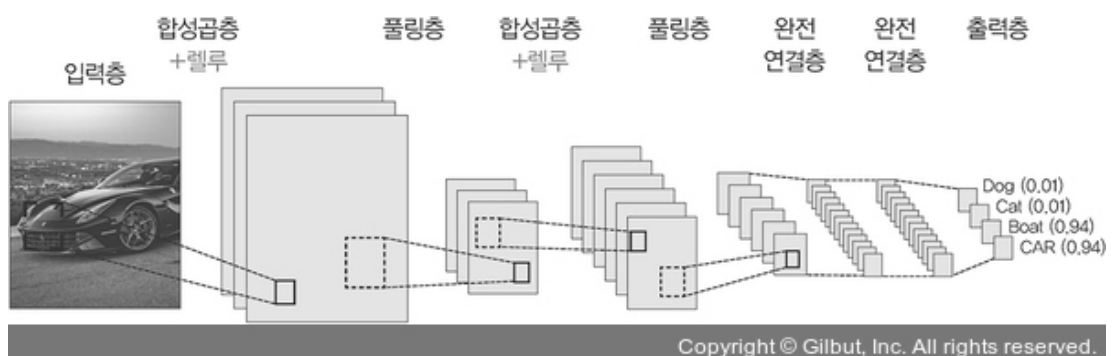


5장 예습과제 : 합성곱 신경망

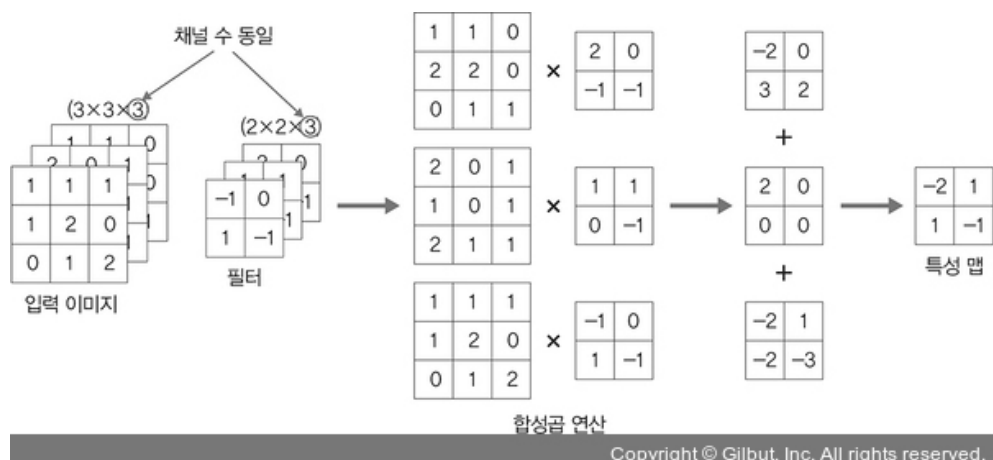
▼ 5.1 합성곱 신경망

• 합성곱 신경망의 구조

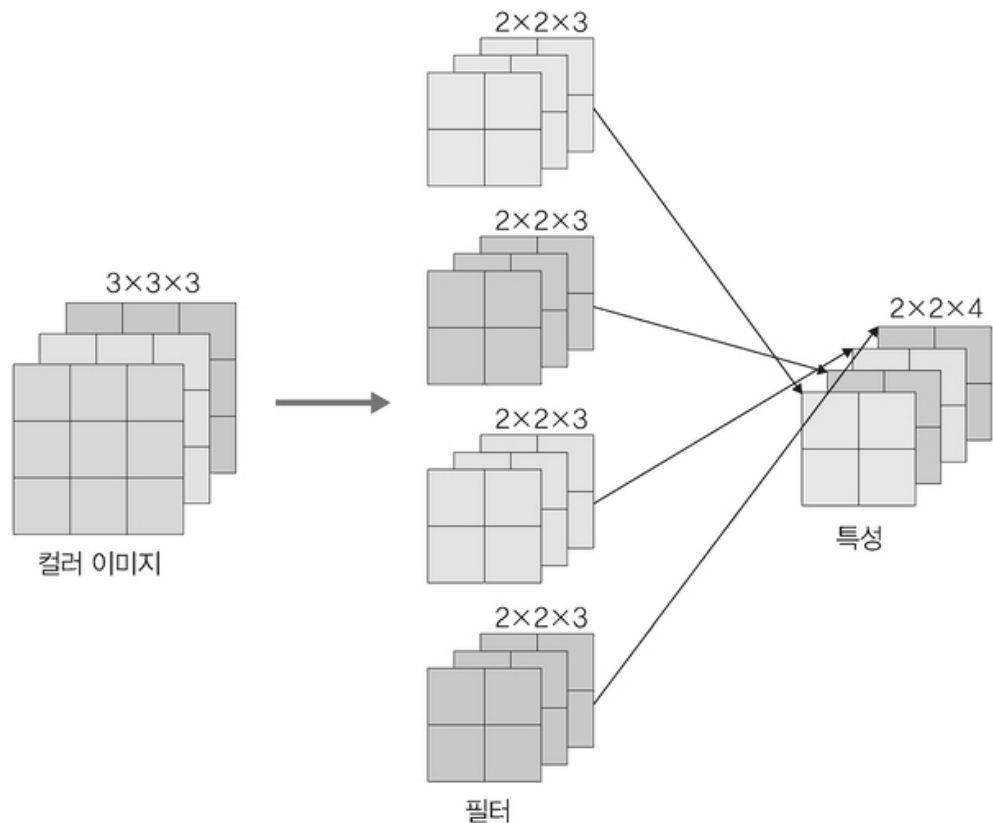
합성곱 신경망은 합성곱층과 풀링층을 거치면서 입력 이미지의 주요 특성 벡터를 추출. 추출된 주요 특성 벡터들은 완전 연결층을 거치면서 **1차원 벡터**로 변화되며, 마지막으로 출력층에서 활성화 함수인 소프트맥스 함수를 이용하여 최종 결과 출력



- 입력층 : 이미지(높이, 너비, 채널의 값을 갖는 3차원데이터)가 최초로 거치게 되는 계층. 채널은 그레이스케일(흑백)이면 1, 컬러면 3의 값.
- 합성곱층 : 입력데이터에서 특성 추출. 특성을 감지하기 위해 커널이나 필터 사용하여 특성 맵이라는 결과물 추출. 보통 **커널은 3*3, 5*5** 사용, **stride는 1**
 - 컬러이미지&필터 1개일 때



- 컬러이미지&필터 3개일 때



• 입력 데이터: $W_1 \times H_1 \times D_1$ (W_1 : 가로, H_1 : 세로, D_1 : 채널 또는 깊이)

• 하이퍼파라미터

- 필터 개수: K
- 필터 크기: F
- 스트라이드: S
- 패딩: P

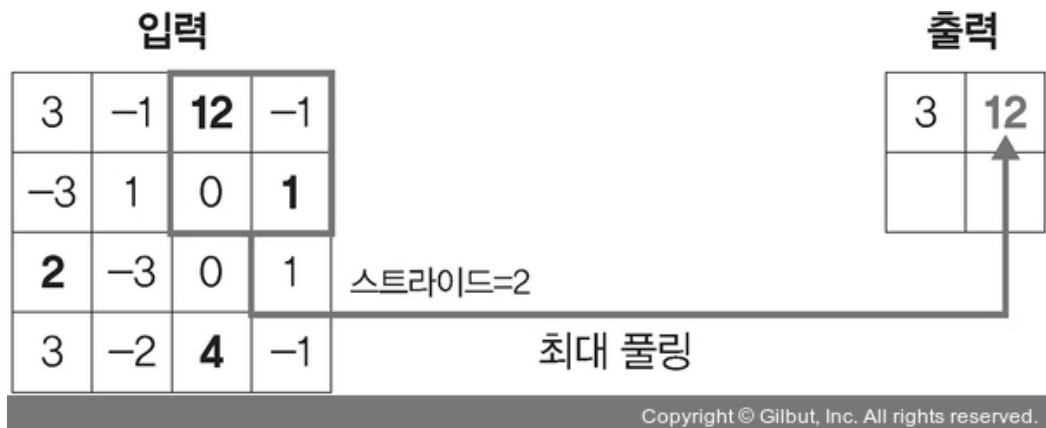
• 출력 데이터

- $W_2 = (W_1 - F + 2P) / S + 1$
- $H_2 = (H_1 - F + 2P) / S + 1$
- $D_2 = K$

■

- 풀링층 : 합성곱층과 유사하게 특정 맵의 차원을 다운 샘플링하여 연산량을 감소, 주요한 특성 벡터를 추출하여 학습을 효과적으로

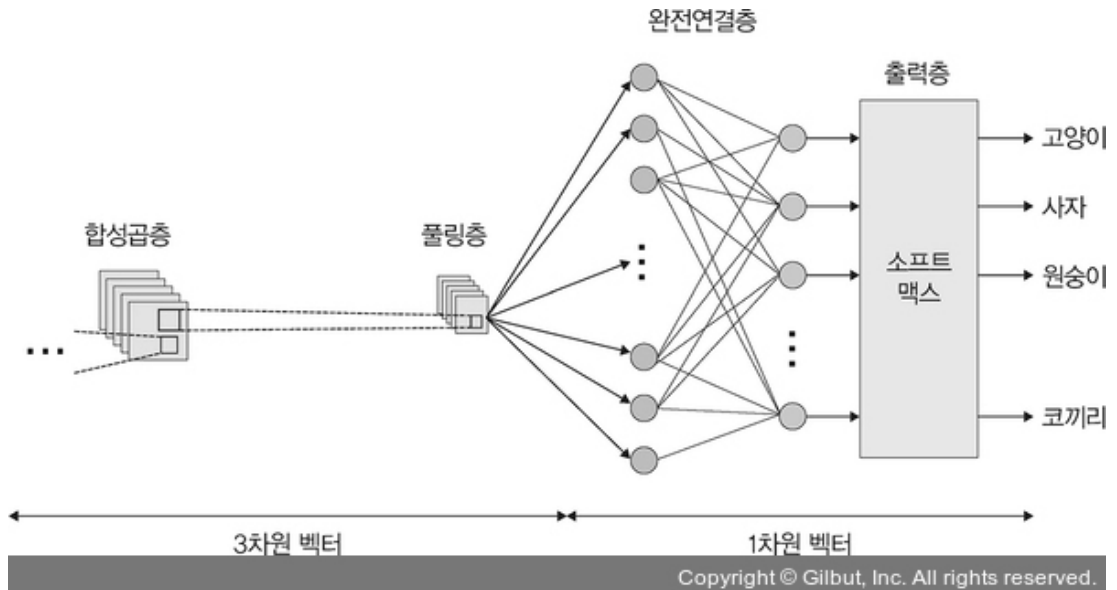
- max pooling : 대상 영역에서 최댓값을 추출 → 대부분이거 사용. 평균화하면 특성이 희미해져서



- average pooling : 대상 영역에서 평균을 반환

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

- 완전연결층 : 3차원벡터에서 1차원 벡터로 펼쳐짐

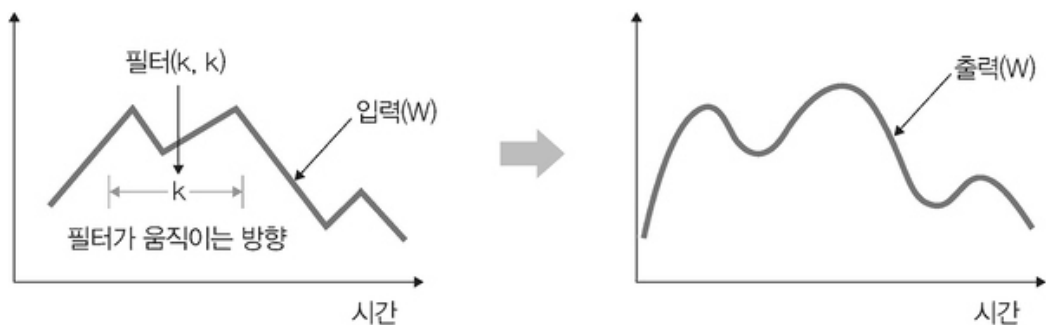


- 출력층 : 소프트맥스 활성화 함수가 사용. 입력받은 값을 0-1사이의 값으로 출력. 이미지가 각 레이블에 속할 확률 출력, 가장 높은 확률의 값이 최종값으로 선정

• 1D, 2D, 3D 합성곱

○ 1D 합성곱

- 필터가 시간을 축으로 좌우로만 이동 가능. 그래프 곡선을 완화할 때 많이 사용.

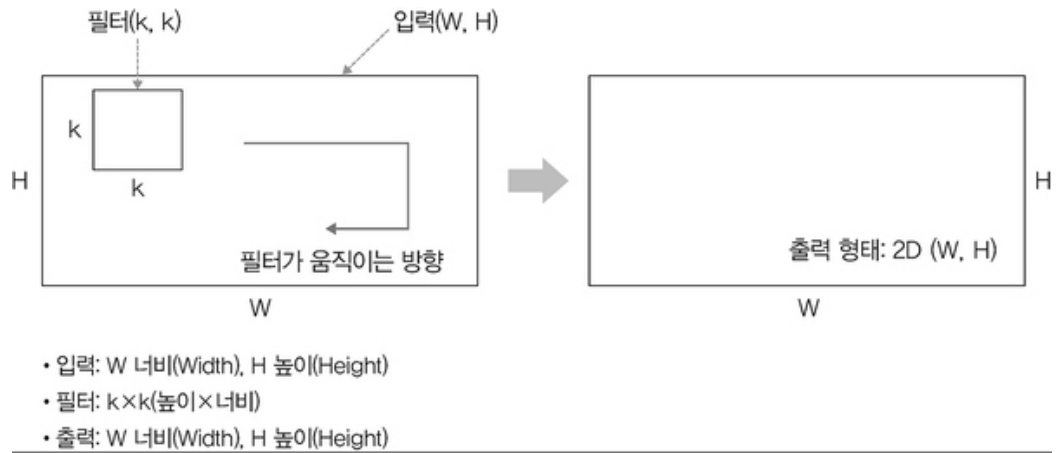


- 입력: W 너비(Width)
- 필터: $k \times k$ (높이 \times 너비)
- 출력: W 너비(Width)

Copyright © Gilbut, Inc. All rights reserved.

○ 2D 합성곱

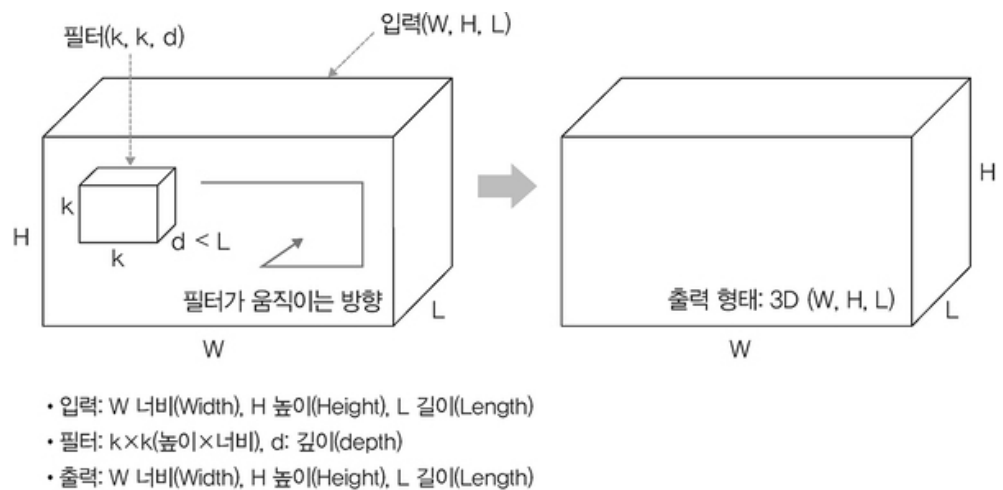
- 필터가 방향 2개로 이동. 출력 형태는 2D 행렬



Copyright © Gilbut, Inc. All rights reserved.

◦ 3D 합성곱

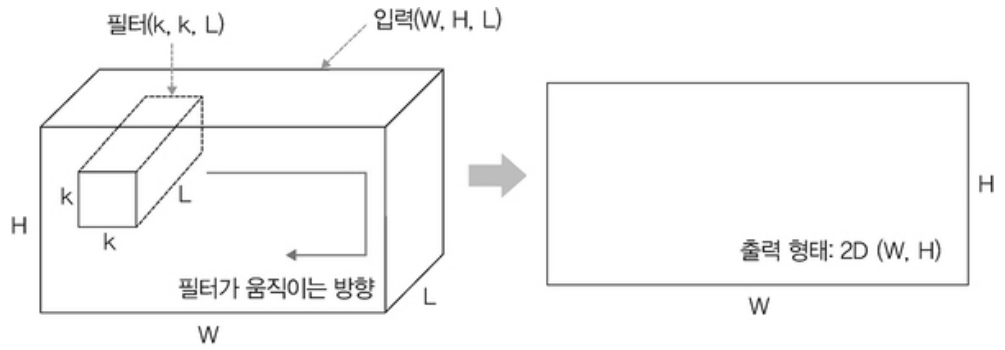
- 움직이는 방향이 3가지. 출력은 3D형태이며 이때 $d < L$ 을 유지하는 것이 중요



Copyright © Gilbut, Inc. All rights reserved.

◦ 3D 입력을 갖는 2D 합성곱

- 입력(W, H, L)에 필터(k, k, L)를 적용하면 출력은 (W, H)
- 필터는 다음 그림과 같이 두 방향으로 움직이며 출력 형태는 2D 행렬
- 대표적 사례로는 LeNet-5와 VGG



- 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)

Copyright © Gilbut, Inc. All rights reserved.

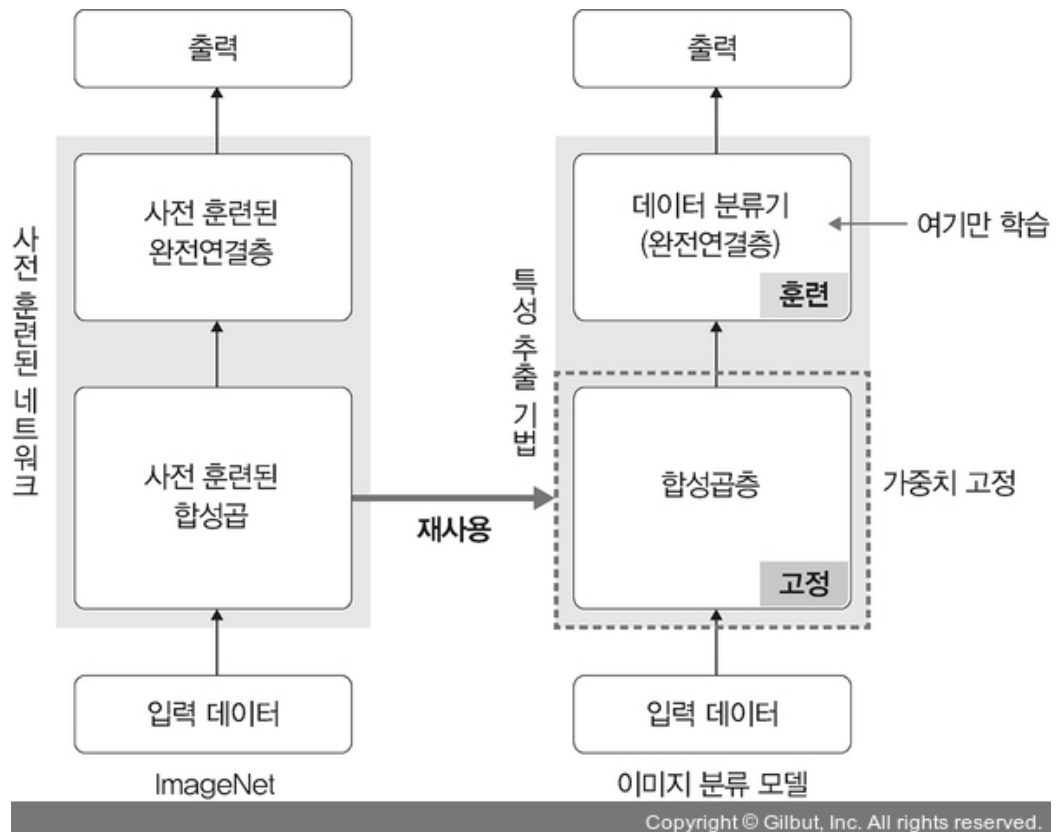
- 1×1 합성곱
 - 3D 형태로 입력
 - 입력(W, H, L)에 필터(1, 1, L)를 적용하면 출력은 (W, H)
 - 연산량이 감소되는 효과가 있으며, 대표적 사례로는 GoogLeNet

▼ 5.2 합성곱 신경망 맛보기

- fashion_mnist 데이터셋

▼ 5.3 전이학습

- 이미지넷(ImageNet)처럼 아주 큰 데이터셋을 써서 훈련된 모델의 가중치를 가져와 우리가 해결하려는 과제에 맞게 보정해서 사용하는 것
 - 특성 추출
 - 마지막에 완전 연결층 부분만 새로 만들.
 - 학습할 때는 마지막 완전연결층만 학습하고, 나머지 계층들은 학습되지 않도록

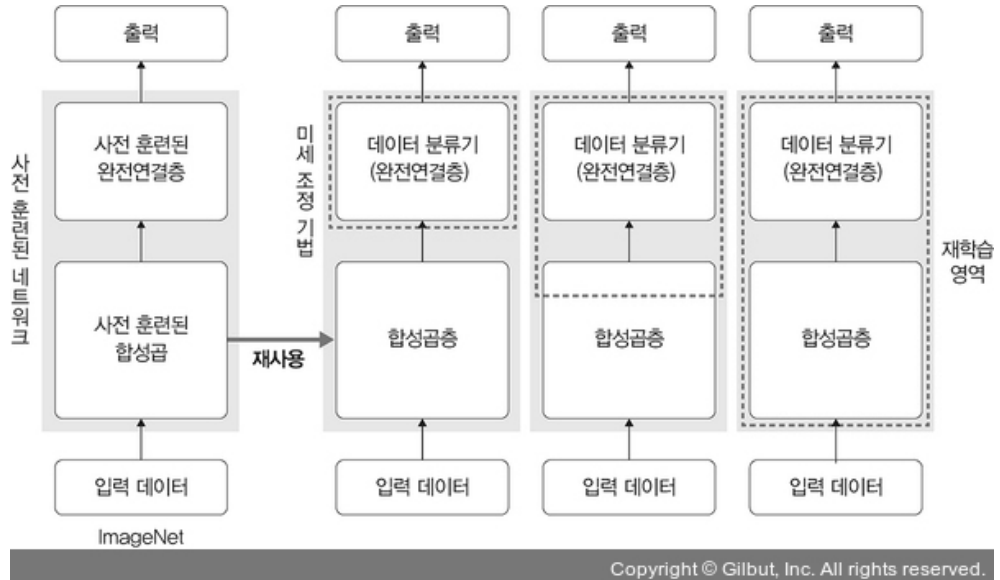


○ 미세 조정 기법

- 특성 추출 기법에서 더 나아가, 사전 훈련된 모델과 합성곱층, 데이터 분류기의 가중치를 업데이트하여 훈련시키는 방식
- 특성 추출에서의 특성이 잘못 추출되었다면, 미세조정 기법으로 새로운 이미지 데이터를 사용하여 네트워크의 가중치를 업데이트하여 **특성을 다시 추출할 수 있음**
- 데이터셋 크기와 사전 훈련된 모델에 따른 전략
 - 1) 데이터셋 크고, 사전훈련 모델과 유사성 적음
모델 전체를 재학습.
 - 2) 데이터셋 크고, 사전훈련 모델과 유사성 큼
합성곱층의 뒷부분(완전연결층과 가까운 부분)과 데이터 분류기를 학습.
 - 3) 데이터셋 작고, 사전훈련 모델과 유사성 적음
합성곱층의 일부분과 데이터 분류기를 학습.
데이터가 적어서 일부 계층에 미세조정기법을 적용한다 하더라도 효과가 없을 수 있음

4) 데이터셋 작고, 사전훈련 모델과 유사성 큼

데이터 분류기(완전연결층)만 학습. 데이터셋이 작아서 미세조정기법 사용하면 과적합 발생할 수 있음



▼ 5.4 설명가능한 CNN

- 필터에 대한 시각화와 특성 맵에 대한 시각화가 있지만, 책에서는 특성 캡에 대한 시각화만 다룸
- 특성 맵에 대한 시각화
 - 결론 : 출력층에 가까울수록 원래 형태를 찾아볼수 없음. 이미지 특징만 전달
 - 결과에 대한 신뢰성이 문제가 되면서, '설명가능한 인공지능'이 이슈가 되는데, CNN은 필터와 특성 맵을 시각화하여 신뢰성 확보 가능

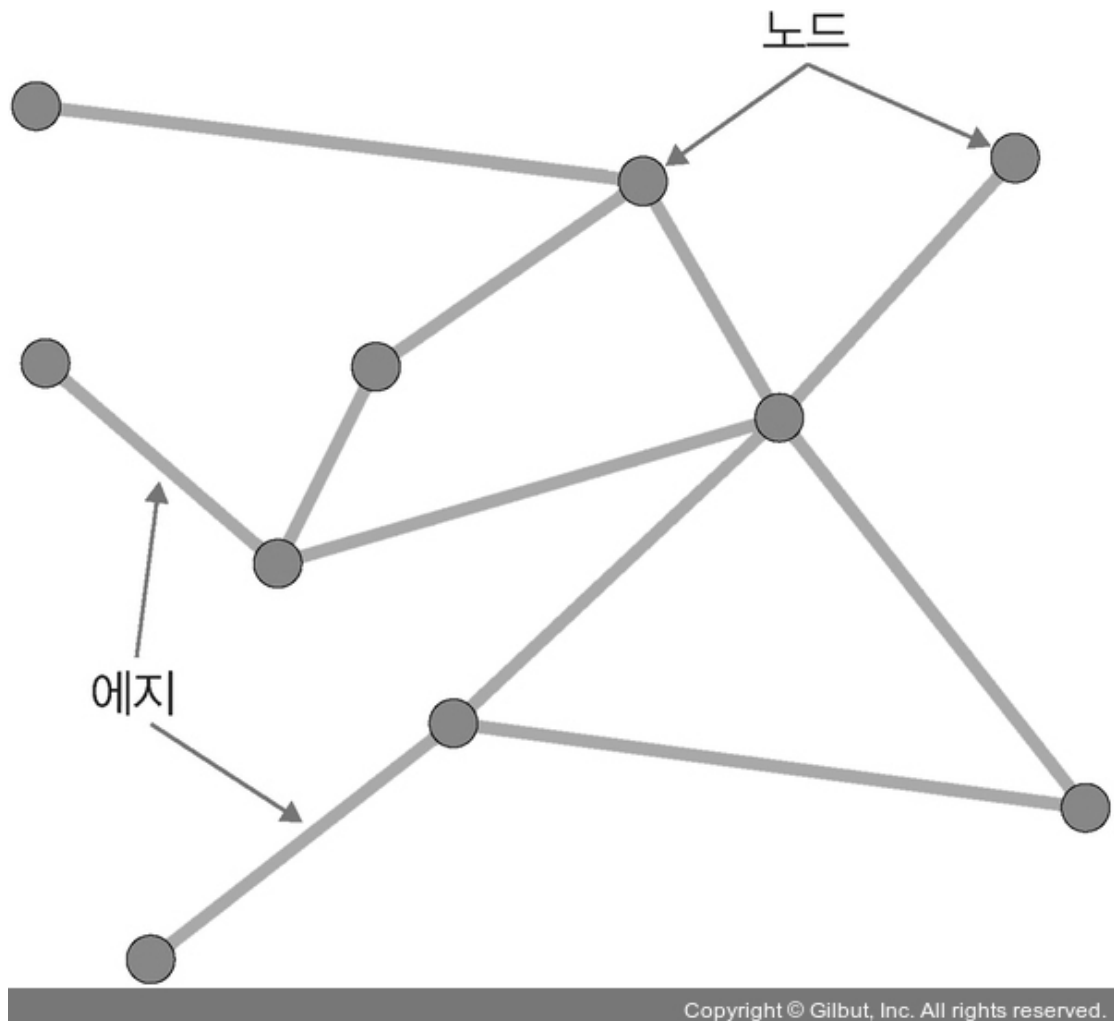
▼ 5.5 그래프 합성곱 네트워크

그래프 데이터를 위한 신경망

5.5.1 그래프란

그래프는 방향성이 있거나(directed) 없는(indirected) 에지로 연결된 노드의 집합.

여기에서 노드와 에지는 일반적으로 풀고자 하는 문제에 대한 전문가 지식이나 직관 등으로 구성



- **노드**(node, vertex): 그림 5-50에서 파란색 원. 원소
- **에지**(edge): 두 노드를 연결한 선. 결합방법(single, double, triple, aromatic)

5.5.2 그래프 신경망

그래프 신경망(Graph Neural Network, GNN)은 그래프 구조에서 사용하는 신경망



1단계. 인접 행렬(adjacency matrix)

왼쪽과 같은 네트워크가 있을 때 노드 n 개를 $n \times n$ 행렬(matrix)로 표현
인접 행렬 내의 값은 'A_{ij}는 i와 j의 관련성 여부'를 만족하는 값으로 채워 줌
컴퓨터가 이해하기 쉽게 그래프로 표현하는 과정

2단계. 특성행렬(feature matrix)

인접 행렬만으로는 특성을 파악하기 어렵기 때문에 **단위 행렬**을 적용
각 입력 데이터에서 이용할 특성을 선택
여기서 각 행은 선택된 특성에 대해 각 노드가 갖는 값을 의미
→ 그래프 특성 추출

5.5.3 그래프 합성곱 네트워크

