

5장

합성곱 신경망 I

5.1 합성곱 신경망

용어정리

시그모이드 함수

- 시그모이드 함수는 선형 함수의 결과를 0~1 사이에서 비선형 형태로 변형

렐루 함수

- 최근 활발히 사용되는 렐루(ReLU) 함수는 입력(x)이 음수일 때는 0을 출력하고, 양수일 때는 x 를 출력
- 경사 하강법(gradient descent)에 영향을 주지 않아 학습 속도가 빠르고, 기울기 소멸 문제가 발생하지 않는 장점이 있음

소프트맥스 함수

- 소프트맥스(softmax) 함수는 입력 값을 0~1 사이에 출력되도록 정규화하여 출력 값들의 총합이 항상 1이 되도록 함
- 소프트맥스 함수는 보통 딥러닝에서 출력 노드의 활성화 함수로 많이 사용
- **손실 함수**는 학습을 통해 얻은 데이터의 추정치가 실제 데이터와 얼마나 차이가 나는지 평가하는 지표라고 할 수 있음
- 이 값이 클수록 많이 틀렸다는 의미이고, 이 값이 '0'에 가까우면 완벽하게 추정할 수 있다는 의미

5.1 합성곱 신경망

- 합성곱 신경망

- 합성곱 신경망(Convolutional Neural Network, CNN)은 합성곱층(convolutional layer)과 풀링층(pooling layer)을 포함하는 이미지 처리 성능이 좋은 인공 신경망 알고리즘
- 영상 및 사진이 포함된 이미지 데이터에서 객체를 탐색하거나 객체 위치를 찾아내는 데 유용한 신경망
- 합성곱 신경망은 이미지에서 객체, 얼굴, 장면을 인식하기 위해 패턴을 찾는 데 특히 유용함

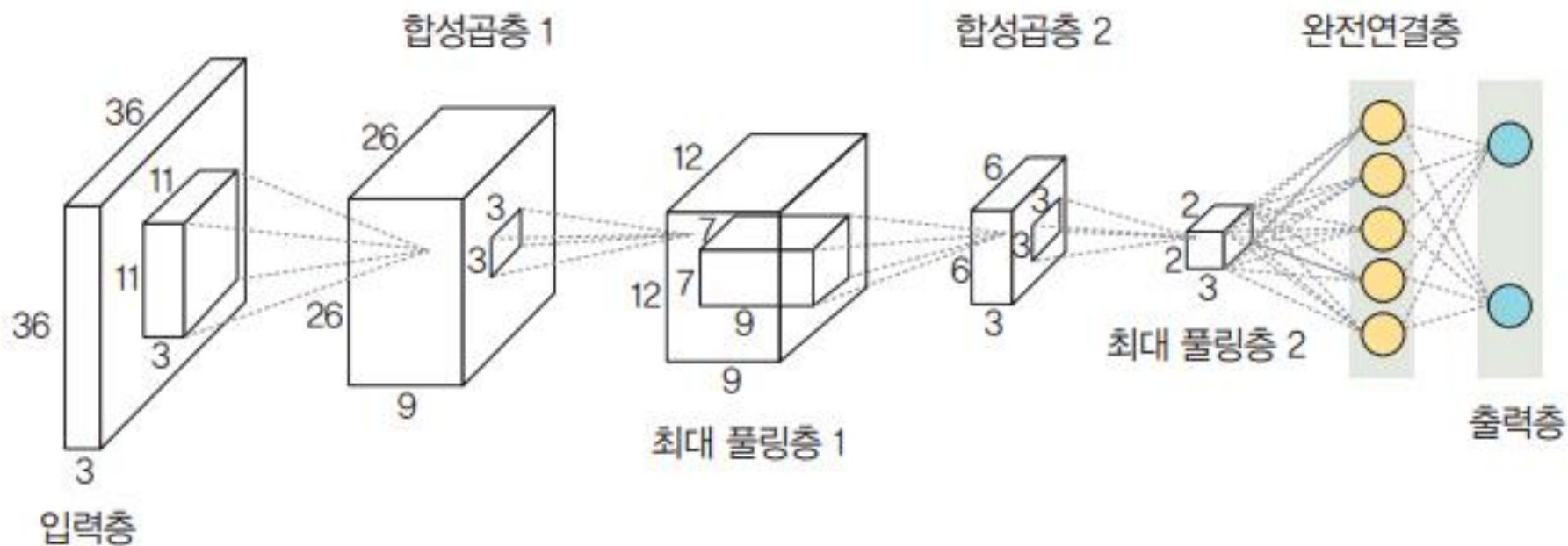
5.1 합성곱 신경망

● 합성곱 신경망

- 합성곱 신경망은 이미지 전체를 한 번에 계산하는 것이 아닌 이미지의 국소적 부분을 계산함으로써 시간과 자원을 절약하여 이미지의 세밀한 부분까지 분석할 수 있는 신경망.
- 순전파 과정에 따라 계산된 오차 정보가 신경망의 모든 노드(출력층→은닉층→입력층)로 전송
- 이러한 계산 과정은 복잡하고 많은 자원(CPU 혹은 GPU, 메모리)을 요구
- 또한, 계산하는 데도 오래 걸림
- 이 문제를 해결하고자 하는 것이 합성곱 신경망

5.1 합성곱 신경망

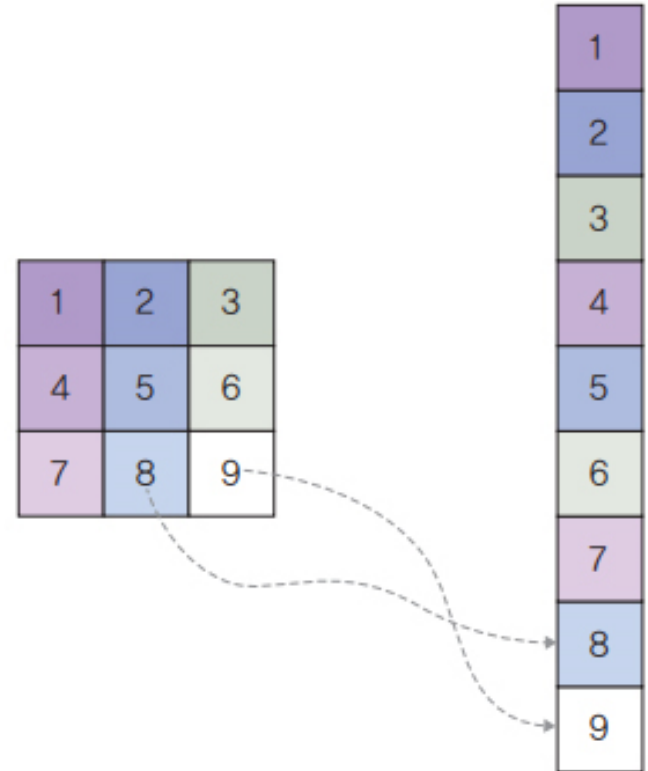
▼ 그림 4-25 합성곱 신경망



5.1 합성곱 신경망

● 합성곱층의 필요성

- 합성곱 신경망은 이미지나 영상을 처리하는 데 유용함
- 예를 들어 다음과 같이 3×3 흑백(그레이스케일) 이미지가 있다고 가정해 보자
- 이미지 분석은 다음 그림의 왼쪽과 같은 3×3 배열을 오른쪽과 같이 펼쳐서(flattening) 각 픽셀에 가중치를 곱하여 은닉층으로 전달하게 됨
- 이미지를 펼쳐서 분석하면 데이터의 공간적 구조를 무시하게 되는데, 이것을 방지하려고 도입된 것이 합성곱층



5.1 합성곱 신경망

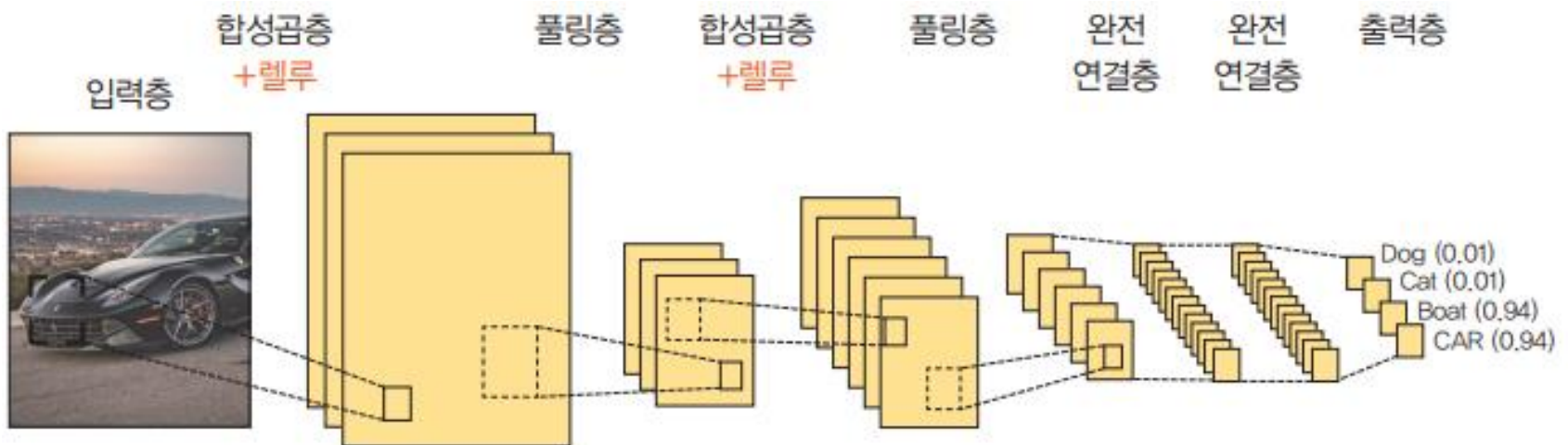
● 합성곱 신경망 구조

- 합성곱 신경망(Convolutional Neural Network, CNN 또는 ConvNet)은 음성 인식이나 이미지/영상 인식에서 주로 사용되는 신경망
- 다차원 배열 데이터를 처리하도록 구성되어 컬러 이미지 같은 다차원 배열 처리에 특화되어 있으며, 다음과 같이 계층 다섯 개로 구성

1. 입력층
2. 합성곱층
3. 풀링층
4. 완전연결층
5. 출력층

5.1 합성곱 신경망

▼ 그림 5-2 합성곱 신경망 구조



5.1 합성곱 신경망

- 합성곱 신경망 구조

- 합성곱 신경망은 합성곱층과 풀링층을 거치면서 입력 이미지의 주요 특성 벡터(feature vector)를 추출
- 그 후 추출된 주요 특성 벡터들은 완전연결층을 거치면서 1차원 벡터로 변환
- 마지막으로 출력층에서 활성화 함수인 소프트맥스(softmax) 함수를 사용하여 최종 결과가 출력

5.1 합성곱 신경망

- 합성곱 신경망 구조

입력층

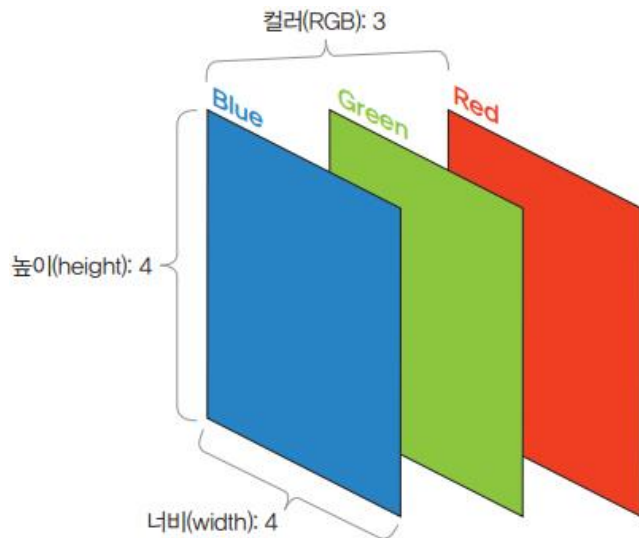
- 입력층(input layer)은 입력 이미지 데이터가 최초로 거치게 되는 계층
- 이미지는 단순 1차원의 데이터가 아닌 높이(height), 너비(width), 채널(channel)의 값을 갖는 3차원 데이터
- 이때 채널은 이미지가 그레이스케일(gray scale)이면 1 값을 가지며, 컬러(RGB)이면 3 값을 가짐

5.1 합성곱 신경망

● 합성곱 신경망 구조

- 예를 들어 다음 그림과 같은 형태는 높이 4, 너비 4, 채널은 RGB를 갖고 있으므로, 이미지 형태(shape)는 (4, 4, 3)으로 표현할 수 있음

▼ 그림 5-3 채널

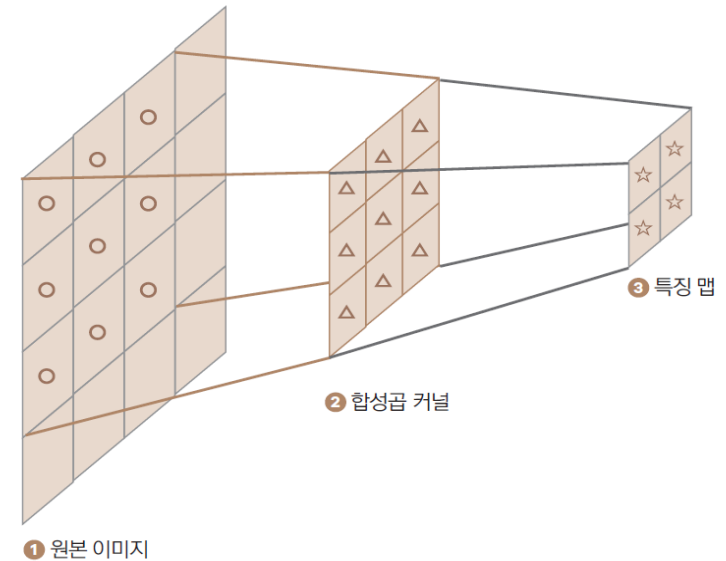


5.1 합성곱 신경망

● 합성곱 신경망 구조

- 합성곱층(convolutional layer)은 입력 데이터에서 특성을 추출하는 역할을 수행
 - 입력 이미지가 들어왔을 때 이미지에 대한 특성을 감지하기 위해 커널(kernel)이나 필터를 사용
 - 커널/필터는 이미지의 모든 영역을 훑으면서 특성을 추출하게 되는데, 이렇게 추출된 결과물이 특성 맵(feature map)
 - 이때 커널은 3×3 , 5×5 크기로 적용되는 것이 일반적이며, 스트라이드(stride)라는 지정된 간격에 따라 순차적으로 이동
- 스트라이드 : 커널의 이동 거리

▼ 합성곱의 계산



5.1 합성곱 신경망

● 합성곱 신경망 구조

- 다음은 스트라이드가 1일 때 이동하는 과정

1단계. 입력 이미지에 3×3 필터 적용

- 입력 이미지와 필터를 포개 놓고 대응되는 숫자끼리 곱한 후 모두 더함

$$(1 \times 1) + (0 \times 0) + (0 \times 1) + (0 \times 0) + (1 \times 1) + (0 \times 0) + (0 \times 1) + (0 \times 0) + (1 \times 1) = 3$$

▼ 그림 5-4 입력 이미지에 3×3 필터 적용



5.1 합성곱 신경망

- 합성곱 신경망 구조

2단계. 필터가 1만큼 이동

$$(0 \times 1) + (0 \times 0) + (0 \times 1) + (1 \times 0) + (0 \times 1) + (0 \times 0) + (0 \times 1) + (1 \times 0) + (1 \times 1) = 1$$

▼ 그림 5-5 입력 이미지에 필터가 1만큼 이동



5.1 합성곱 신경망

● 합성곱 신경망 구조

6단계. 필터가 1만큼 마지막으로 이동

$$(0 \times 1) + (1 \times 0) + (0 \times 1) + (0 \times 0) + (1 \times 1) + (0 \times 0) + (0 \times 1) + (1 \times 0) + (0 \times 1) = 1$$

▼ 그림 5-9 입력 이미지에 필터가 1만큼 마지막으로 이동



- 이미지 크기가 (6, 6, 1)이며, 3×3 크기의 커널/필터가 스트라이드 1 간격으로 이동하면서 합성곱 연산을 수행
- 커널은 스트라이드 간격만큼 순회하면서 모든 입력 값과의 합성곱 연산으로 새로운 특성 맵을 만들게 됨
- 커널과 스트라이드의 상호 작용으로 원본 (6, 6, 1) 크기가 (4, 4, 1) 크기의 특성 맵으로 줄어 들었음

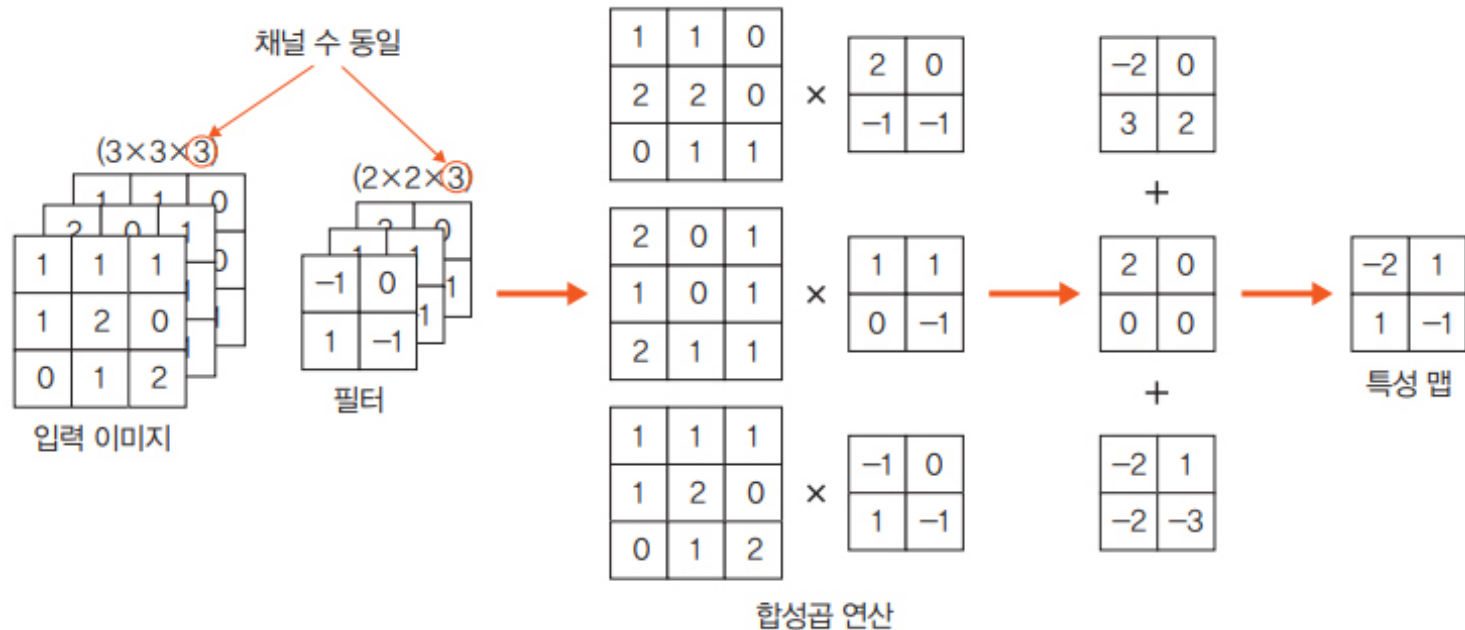
5.1 합성곱 신경망

● 합성곱 신경망 구조

- 지금까지 그레이스케일에 대한 이미지를 확인했는데, 이제 컬러 이미지의 합성곱을 알아보자
- 앞서 다룬 그레이스케일 이미지와 구분되는 특징은 첫째, 필터 채널이 3이라는 것과 둘째, RGB 각각에 서로 다른 가중치로 합성곱을 적용한 후 결과를 더해 준다는 것
- 그 외 스트라이드 및 연산하는 방법은 동일
- 이때 필터 채널이 3이라고 해서 필터 개수도 세 개라고 오해하기 쉬운데, 실제로는 필터 개수가 한 개라는 점에 주의

5.1 합성곱 신경망

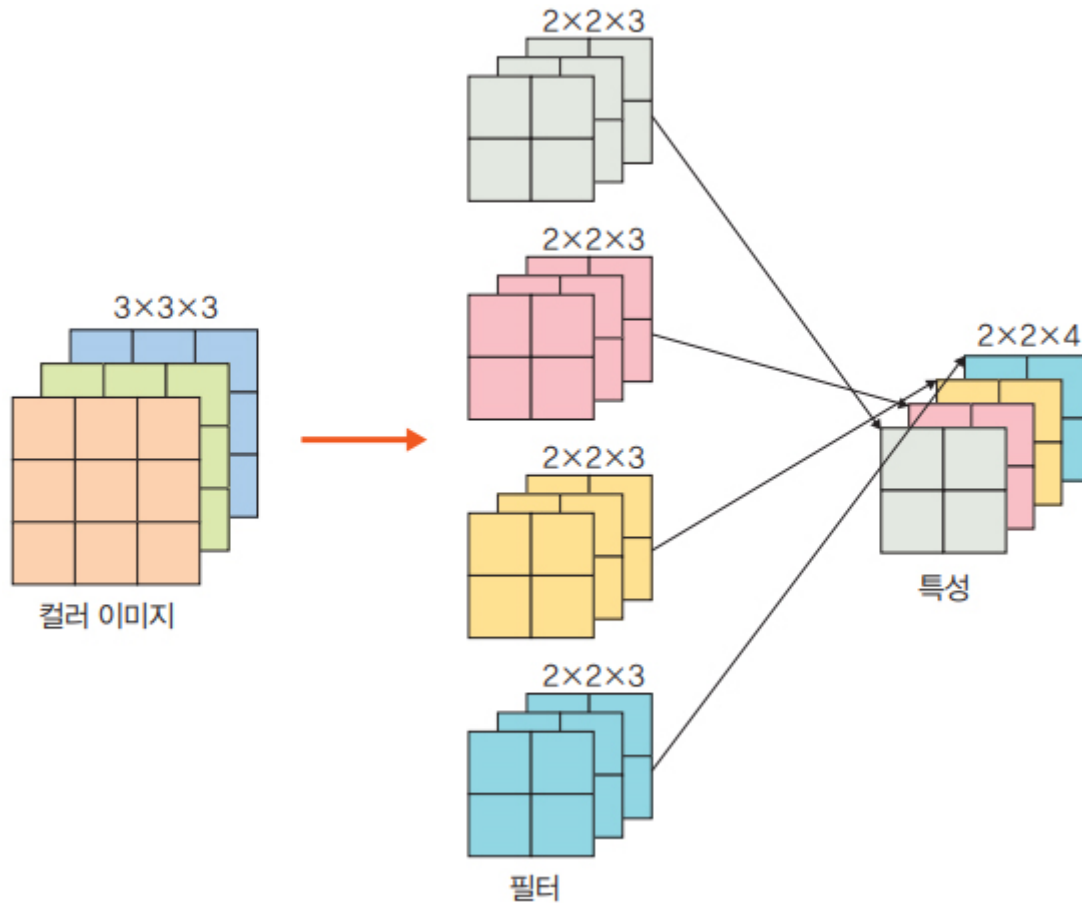
▼ 그림 5-10 컬러 이미지 합성곱



- , 필터 채널이 3이라는 것과 둘째, RGB 각각에 서로 다른 가중치로 합성곱을 적용한 후 결과를 더해 준다는 것
- 그 외 스트라이드 및 연산하는 방법은 동일
- 이때 필터 채널이 3이라고 해서 필터 개수도 세 개라고 오해하기 쉬운데, 실제로는 필터 개수가 한 개라는 점에 주의

5.1 합성곱 신경망

▼ 그림 5-11 필터가 2 이상인 합성곱



5.1 합성곱 신경망

● 합성곱 신경망 구조

- 즉, 합성곱층을 요약하면 다음과 같음

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

5.1 합성곱 신경망

- 합성곱 신경망 구조

- 다운 샘플링

- 다운 샘플링(sub-sampling)은 다음 그림과 같이 이미지를 축소하는 것

- ▼ 그림 5-12 다운 샘플링



- 풀링층(pooling layer)은 합성곱층과 유사하게 특성 맵의 차원을 다운 샘플링하여 연산량을 감소시키고, 주요한 특성 벡터를 추출하여 학습을 효과적으로 할 수 있게 함

5.1 합성곱 신경망

- 합성곱 신경망 구조

- 풀링 연산에는 두 가지가 사용

- 최대 풀링(max pooling): 대상 영역에서 최댓값을 추출
- 평균 풀링(average pooling): 대상 영역에서 평균을 반환

- 대부분의 합성곱 신경망에서는 최대 풀링이 사용되는데, 평균 풀링은 각 커널 값을 평균화시켜 중요한 가중치를 갖는 값의 특성이 희미해질 수 있기 때문임

5.1 합성곱 신경망

- 합성곱 신경망 구조

- 다음은 최대 풀링의 연산 과정

첫 번째 최대 풀링 과정

- 3, -1, -3, 1 값 중에서 최댓값(3)을 선택

▼ 그림 5-13 첫 번째 최대 풀링 과정



5.1 합성곱 신경망

● 합성곱 신경망 구조

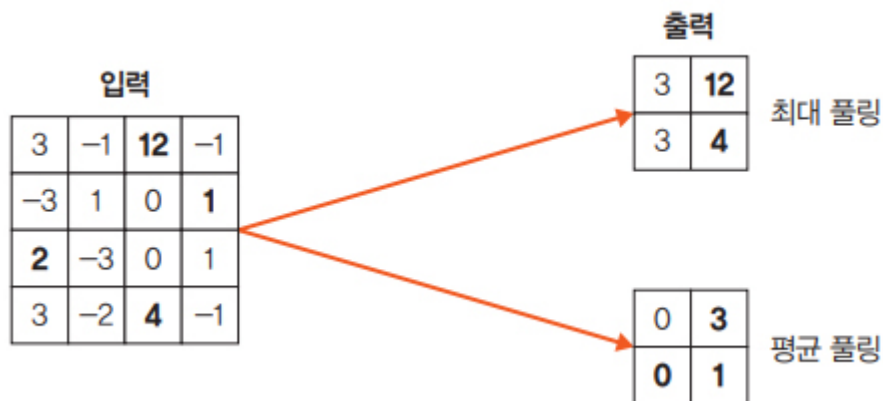
- 평균 풀링의 계산 과정은 최대 풀링과 유사한 방식으로 진행하되 다음과 같이 각 필터의 평균으로 계산

$$0 = (3 + (-1) + (-3) + 1) / 4$$

$$3 = (12 + (-1) + 0 + 1) / 4$$

$$0 = (2 + (-3) + 3 + (-2)) / 4$$

$$1 = (3 + (-2) + 4 + (-1)) / 4$$



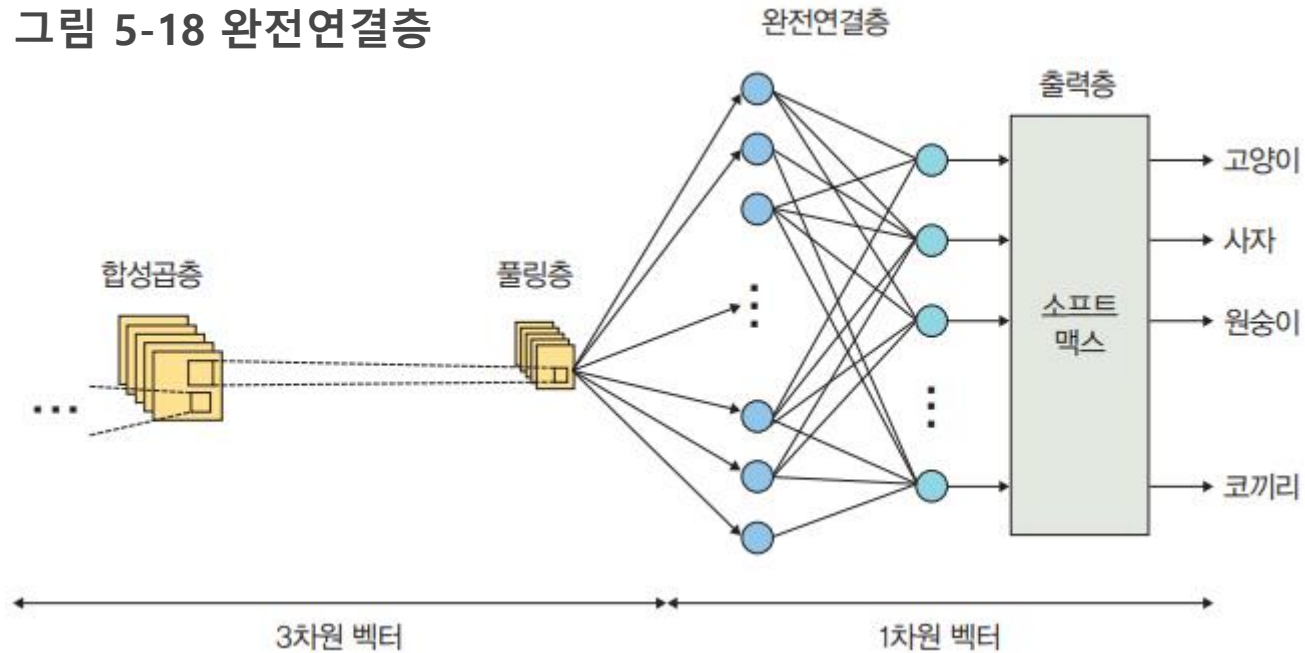
5.1 합성곱 신경망

● 합성곱 신경망 구조

- 최대 풀링과 평균 풀링을 요약하면 다음과 같음(최대 풀링과 평균 풀링의 계산 과정은 다르지만 사용하는 파라미터는 동일)
 - 입력 데이터: $W_1 \times H_1 \times D_1$
 - 하이퍼파라미터
 - 필터 크기: F
 - 스트라이드: S
 - 출력 데이터
 - $W_2 = (W_1 - F) / S + 1$
 - $H_2 = (H_1 - F) / S + 1$
 - $D_2 = D_1$

5.1 합성곱 신경망

▼ 그림 5-18 완전연결층



- 출력층(output layer)에서는 소프트맥스 활성화 함수가 사용되는데, 입력받은 값을 0~1 사이의 값으로 출력
- 마지막 출력층의 소프트맥스 함수를 사용하여 이미지가 각 레이블(label)에 속할 확률 값이 출력되며, 이때 가장 높은 확률 값을 갖는 레이블이 최종 값으로 선정

감사합니다

용어정리

용어정리

시그모이드 함수

- 시그모이드 함수는 선형 함수의 결과를 0~1 사이에서 비선형 형태로 변형

렐루 함수

- 최근 활발히 사용되는 렐루(ReLU) 함수는 입력(x)이 음수일 때는 0을 출력하고, 양수일 때는 x 를 출력
- 경사 하강법(gradient descent)에 영향을 주지 않아 학습 속도가 빠르고, 기울기 소멸 문제가 발생하지 않는 장점이 있음

소프트맥스 함수

- 소프트맥스(softmax) 함수는 입력 값을 0~1 사이에 출력되도록 정규화하여 출력 값들의 총합이 항상 1이 되도록 함
- 소프트맥스 함수는 보통 딥러닝에서 출력 노드의 활성화 함수로 많이 사용
- **손실 함수**는 학습을 통해 얻은 데이터의 추정치가 실제 데이터와 얼마나 차이가 나는지 평가하는 지표라고 할 수 있음
- 이 값이 클수록 많이 틀렸다는 의미이고, 이 값이 '0'에 가까우면 완벽하게 추정할 수 있다는 의미

용어정리

구분	구성 요소	설명
층	입력층(input layer)	데이터를 받아들이는 층
	은닉층(hidden layer)	모든 입력 노드부터 입력 값을 받아 가중합을 계산하고, 이 값을 활성화 함수에 적용하여 출력층에 전달하는 층
	출력층(output layer)	신경망의 최종 결과값이 포함된 층
가중치(weight)		노드와 노드 간 연결 강도
바이어스(bias)		가중합에 더해 주는 상수로, 하나의 뉴런에서 활성화 함수를 거쳐 최종적으로 출력되는 값을 조절하는 역할을 함
가중합(weighted sum), 전달 함수		가중치와 신호의 곱을 합한 것
함수	활성화 함수(activation function)	신호를 입력받아 이를 적절히 처리하여 출력해 주는 함수
	손실 함수(loss function)	가중치 학습을 위해 출력 함수의 결과와 실제 값 간의 오차를 측정하는 함수