

5장 합성곱 신경망1

📅 날짜	@2025년 3월 19일 → 2025년 3월 25일
📌 선택	DL세션 과제
📎 실습 파일	<u>Week3_예습과제_이수나.ipynb</u>
📌 주차	3주차
🌟 진행 상태	진행 중



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

5.1 합성곱 신경망

5.1.1 합성곱층의 필요성

5.1.2 합성곱 신경망 구조

1. 입력층 (input layer)
2. 합성곱층 (convolutional layer)
3. 풀링층 (pooling layer)
4. 완전연결층 (fully connected layer)
5. 출력층 (output layer)

5.1.3 1D, 2D, 3D 합성곱

5.2 합성곱 신경망 맛보기

Conv2d 계층에서의 출력 크기 구하는 공식

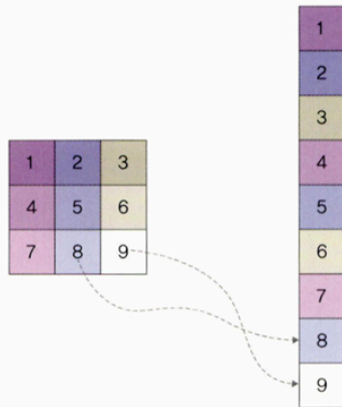
MaxPool2d 계층에서 출력 크기 구하는 공식

5.1 합성곱 신경망

5.1.1 합성곱층의 필요성

- 합성곱 신경망은 이미지나 영상 처리에 유용
- 이미지 분석은 flattening을 통해 각 픽셀에 가중치를 곱하여 은닉층으로 전달하게 됨.
이때, 데이터의 공간적 구조를 무시하지 않기 위해서 도입된 것이 합성곱층

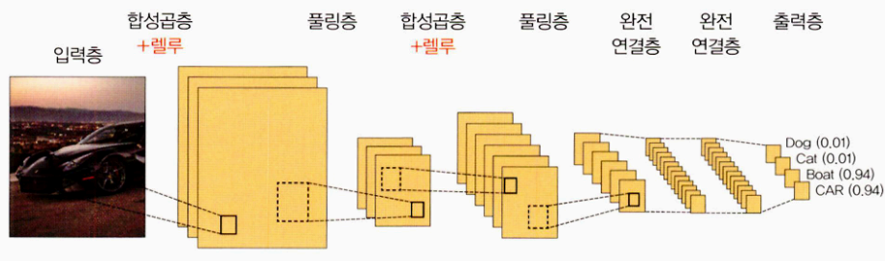
▼ 그림 5-1 합성곱층 원리



5.1.2 합성곱 신경망 구조

CNN은 음성 인식이나 이미지/영상 인식에서 주로 사용되는 신경망이다. 다차원 배열 데이터를 처리하도록 구성되어 컬러 이미지 같은 다차원 배열 처리에 특화되어 있으며, 5개의 계층으로 구성된다.

▼ 그림 5-2 합성곱 신경망 구조



1. 입력층 (input layer)

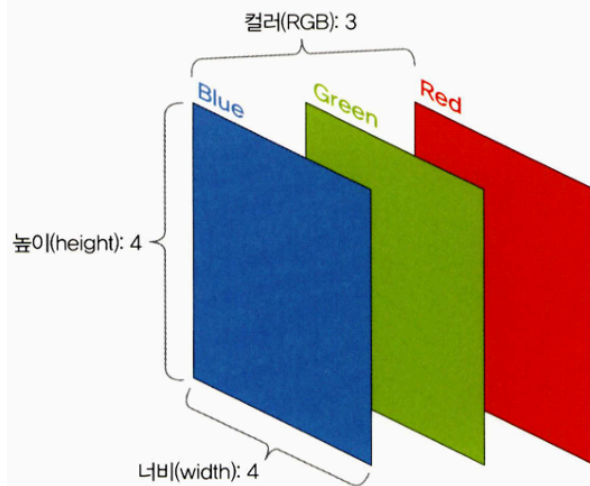
: 입력 이미지 데이터가 최초로 거치게 되는 계층

이미지는 height, width, channel의 값을 갖는 3차원 데이터

- channel : gray scale이면 1, RGB이면 3을 가짐

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

♥ 그림 5-3 채널



2. 합성곱층 (convolutional layer)

: 입력 데이터에서 특성을 추출하는 역할을 수행

이미지 입력 ⇒ 이미지에 대한 특성을 감지하기 위한 **커널**이나 **필터**를 사용 ⇒ 이미지의 모든 영역을 훑으면서 특성을 추출

⇒ 추출된 결과물 : **특성 맵(feature map)**

- 커널은 3x3, 5x5 크기로 적용되는 것이 일반적이며, **stride**라는 지정된 간격에 따라 순차적으로 이동

[그레이스케일 이미지]

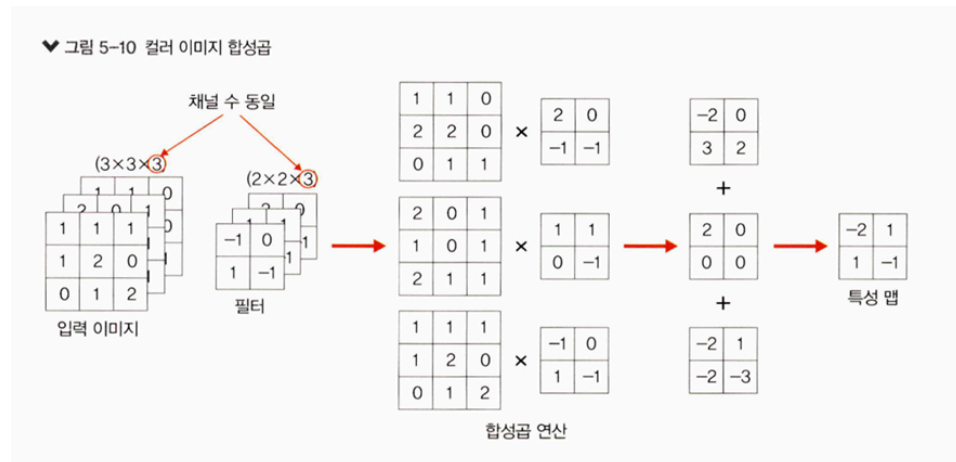
$$(0 \times 1) + (1 \times 0) + (0 \times 1) + (0 \times 0) + (1 \times 1) + (0 \times 0) + (0 \times 1) + (1 \times 0) + (0 \times 1) = 1$$

♥ 그림 5-9 입력 이미지에 필터가 1만큼 마지막으로 이동



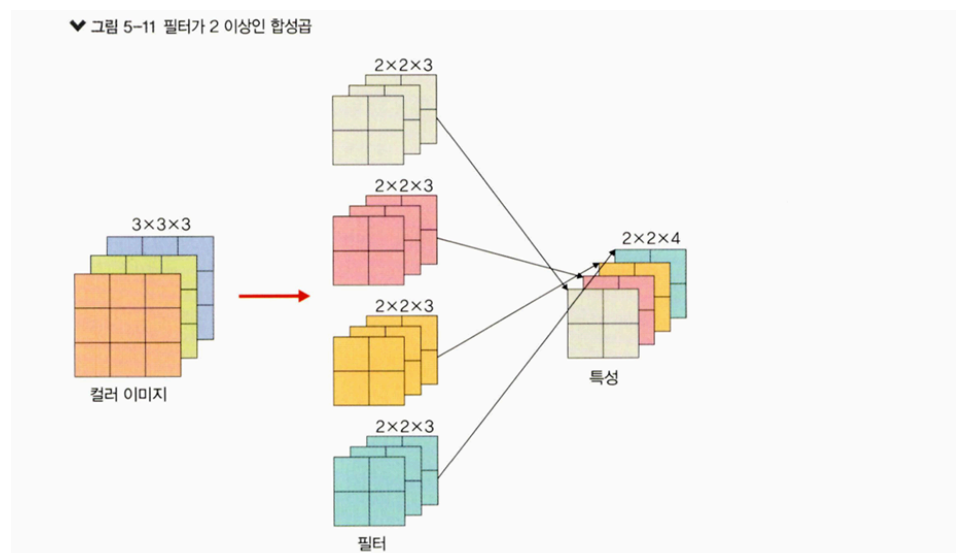
- 입력 이미지와 필터를 포개 놓고 대응되는 숫자끼리 곱한 후 모두 더함
- 스트라이드 간격으로 이동하면서 연산을 수행
- ⇒ 새로운 특성 맵

[RGB 이미지]



- 필터 채널이 3
- RGB 각각에 서로 다른 가중치로 합성곱을 적용 후 결과를 더해줌
- 필터 채널 개수 ≠ 필터 개수(1개)

[필터가 2개 이상인 합성곱]



- 필터 각각이 특성 추출 결과의 채널이 됨



합성곱층의 파라미터

<입력 데이터>

: $W1 \times H1 \times D1$ ($W1$: 가로, $H1$: 세로, $D1$: 채널 또는 깊이)

<하이퍼파라미터>

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

<출력 데이터>

$$W2 = (W1 - F + 2P) / S + 1$$

$$H2 = (H1 - F + 2P) / S + 1$$

$$D2 = K$$

3. 풀링층 (pooling layer)

: 합성곱층과 유사하게 특성 맵의 차원을 다운 샘플링하여 연산량을 감소시키고, 주요한 특성 벡터를 추출하여 학습을 효과적으로 할 수 있게 함

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

대부분의 CNN에서는 최대 풀링이 사용됨

: 평균 풀링은 각 커널 값을 평균화시켜 중요한 가중치를 갖는 값의 특성이 희미해질 수 있기 때문

▼ 그림 5-16 네 번째 최대 풀링 과정



👉 최대 풀링과 평균 풀링의 파라미터

[입력데이터] : $W1 \times H1 \times D$

[하이퍼파라미터]

- 필터 크기: F
- 스트라이드: S

[출력 데이터]

$$W2 = (W1 - F)/S + 1$$

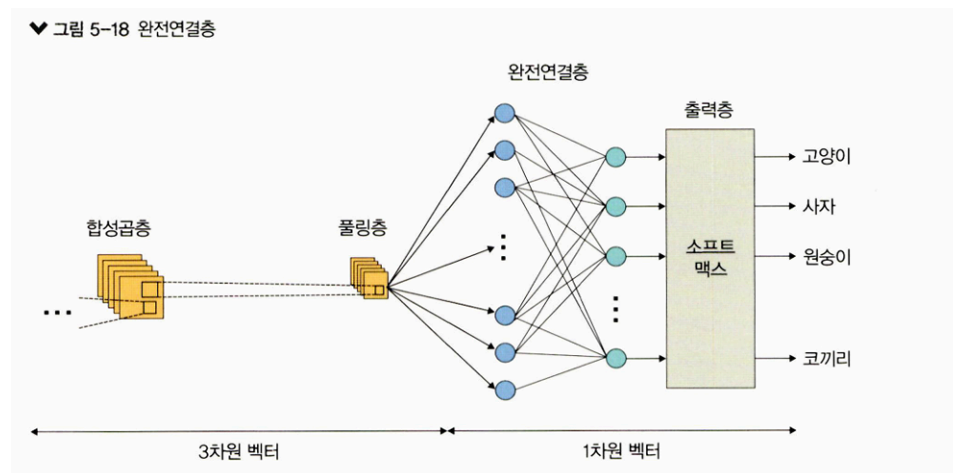
$$H2 = (H1 - F)/S + 1$$

$$D2 = Dq$$

4. 완전연결층 (fully connected layer)

: 합성곱층과 풀링층을 거치면서 차원이 축소된 특성 맵은 최종적으로 완전연결층으로 전달 됨.

이미지는 3차원 벡터 → 1차원 벡터로 펼쳐짐(flatten)



5. 출력층 (output layer)

: 소프트맥스 활성화 함수 사용 → 입력값을 0~1 사이의 값으로 출력

⇒ 이미지가 각 label에 속할 확률 값이 출력되며, 이때 가장 높은 확률을 갖는 레이블이 최종적으로 선정됨.

5.1.3 1D, 2D, 3D 합성곱

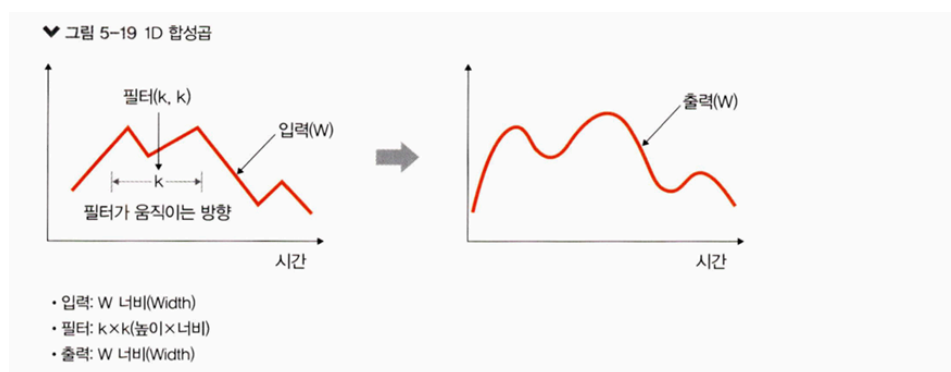
: 합성곱은 이동하는 방향의 수와 출력 형태에 따라 1D, 2D, 3D로 분류됨

<1D 합성곱>

: 필터가 시간을 축으로 좌우로만 이동할 수 있는 합성곱

- 그래프 곡선을 완화할 때 많이 사용됨

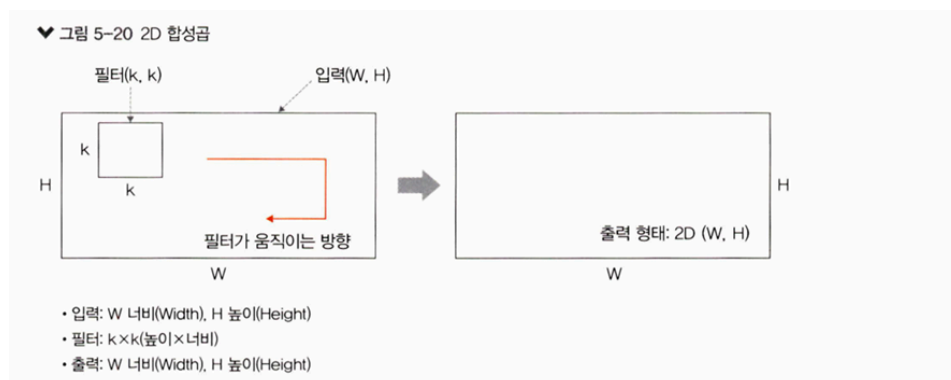
(ex) 입력 [1,1,1,1] 필터 [0.25,0.5,0.25] ⇒ 출력 [1,1,1]



<2D 합성곱>

: 필터가 방향 두 개로 움직이는 형태

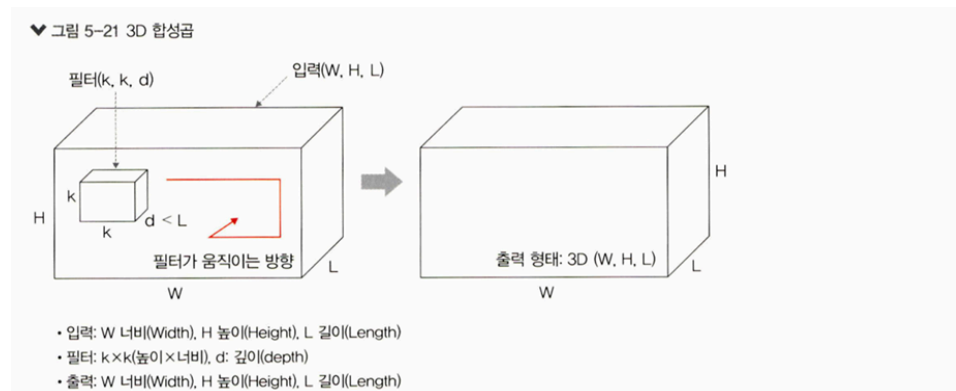
- 입력(W,H)과 필터(k,k) ⇒ 출력(W,H): 2D형태



<3D 합성곱>

: 3D 합성곱은 필터가 움직이는 방향이 3개

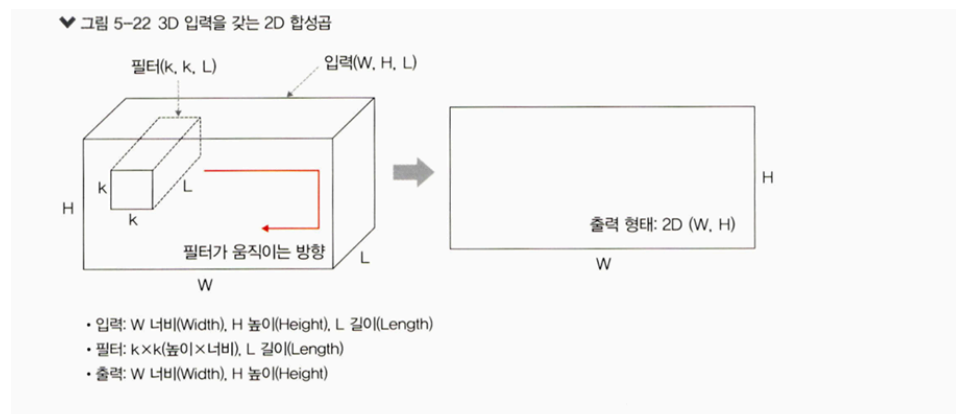
- 입력(W,H,L)과 필터(k,k,d) \Rightarrow 출력 (W,H,L): 3D
- 출력은 $d < L$ 을 유지해야 함



<3D 입력을 갖는 2D 합성곱>

3D 형태임에도 출력 형태가 2D행렬을 취하는 것

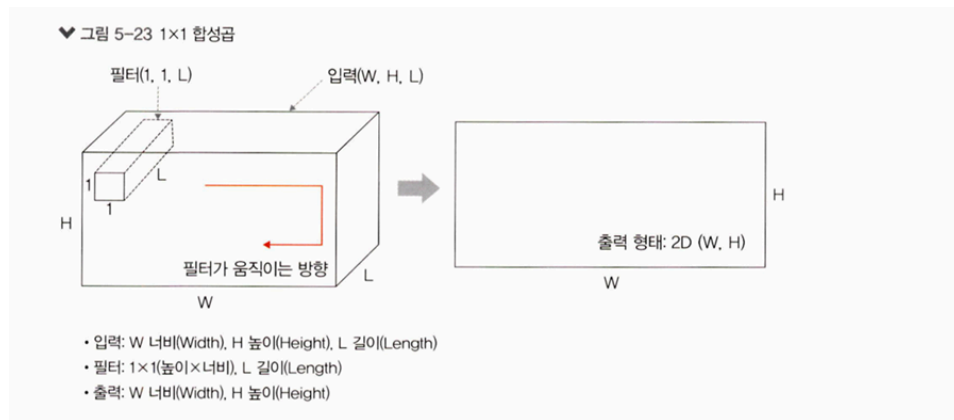
- 필터는 두 방향으로 움직임
- 입력(W,H,L) 필터(k,k,L) \Rightarrow 출력(W,H)
- LeNet-5, VGG



<1×1 합성곱>

- 1×1 합성곱에서 채널 수를 조정해서 연산량이 감소되는 효과

- 입력(W,H,L) 필터(1,1,L) ⇒ 출력(W,H)
- GooLeNet



5.2 합성곱 신경망 맛보기

fashion_mnist 데이터셋을 이용한 합성곱 신경망 실습

Conv2d 계층에서의 출력 크기 구하는 공식

- 출력 크기 = $(W - F + 2P) / S + 1$
 - W: 입력 데이터의 크기
 - F: 커널 크기
 - P: 패딩 크기
 - S: 스트라이드

(예) `nn.Conv2d(in_channels=1, out_channels=32, kernel_size=3, padding=1)`

⇒ $784 - 3 + (2 \times 1) + 1 = 784$

fashion_minist 입력크기 784, stride 기본값(1,1)

출력 형태 [32, 784, 784]

MaxPool2d 계층에서 출력 크기 구하는 공식

- 출력 크기 = IF / F
 - IF: 입력 필터의 크기(=Conv2d의 출력 크기)
 - F: 커널사이즈

(예) `nn.MaxPool2d(kernel_size=2, stride=2)`

⇒ $784(\text{Conv2d계산결과})/2 = 392$