

2022학년도 1학기 컴퓨터언어학

제11강 합성곱 신경망 (1)

박수지

서울대학교 인문대학 언어학과

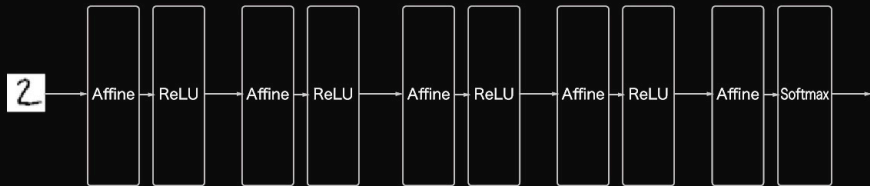
2022년 4월 11일 월요일

오늘의 목표

- 1 합성곱 신경망(Convolutional Neural Network, CNN)의 구조를 설명할 수 있다.
- 2 CNN에서 합성곱 계층과 풀링 계층이 어떻게 작동하는지 설명할 수 있다.

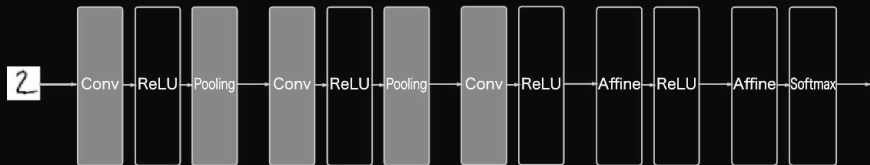
가중치 매개변수 행렬과 곱해 주는 Affine 계층 대신...

그림 7-1 완전연결 계층(Affine 계층)으로 이뤄진 네트워크의 예



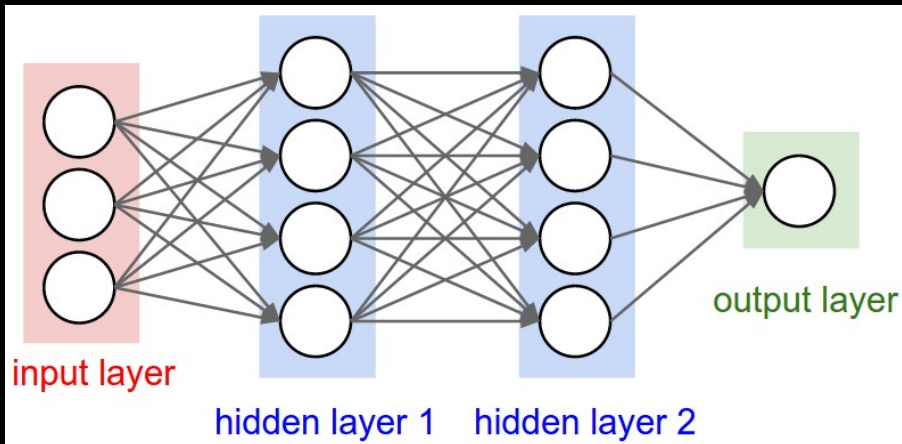
합성곱 계층(Conv)과 풀링 계층(Pooling)을 추가한다.

그림 7-2 CNN으로 이뤄진 네트워크의 예 : 합성곱 계층과 풀링 계층이 새로 추가(회색)



지금까지 했던 신경망

완전연결 신경망: 인접하는 계층의 모든 뉴런과 결합한다.



완전연결 신경망의 한계점

모든 입력 데이터를 동등하고 독립적인 뉴런으로 취급하므로 형상에 담긴 정보를 살릴 수 없다.

- 이미지는 세로·가로·채널(색상)로 구성된 3차원 데이터인데...
- 완전연결 계층에서는 1차원 데이터(벡터)로 표상해야 했다.

3차원 형상에 담길 수 있는 정보

- 공간적으로 가까운 픽셀 사이의 연관성
- RGB 색상 사이의 연관성

그러나 CNN에서는 3차원 데이터를 사용할 수 있다!

CNN의 희망

이미지를 3차원 데이터로 입력받고 전달하므로 이미지를 제대로 이해할 수 있다.

합성곱 연산

필터의 윈도우를 일정 간격으로 이동해 가며 입력 데이터에 적용한다.

- 필터(filter)는 커널(kernel)이라고도 한다.
- 필터의 매개변수가 가중치에 해당한다.

적용 방법

- 1 입력과 필터에서 대응하는 원소끼리 곱한 후 총합을 구한다.
- 2 (2)의 계산 결과를 출력의 해당 장소에 저장한다.
- 3 (1-2)를 모든 장소에서 수행한다.

형상

출력 (2, 2)

1	2	3	0
0	1	2	3
3	0	1	2
2	3	0	1

⑥

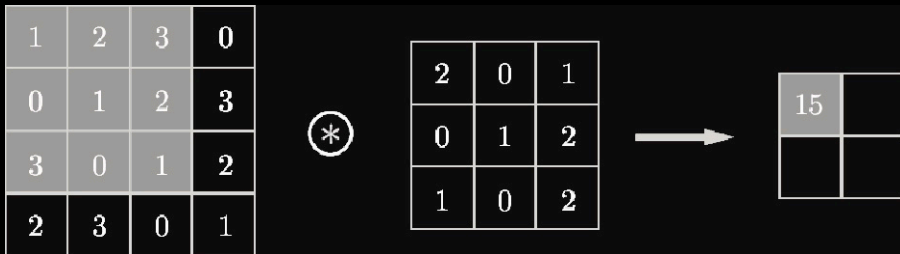
2	0	1
0	1	2
1	0	2

필터



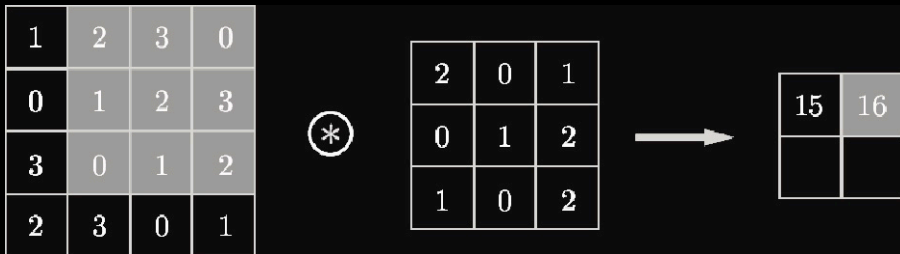
15	16
6	15

합성곱 연산 예시

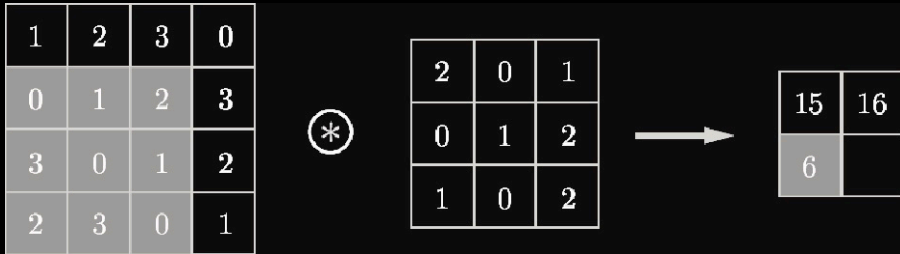


$$1 \times 2 + 2 \times 0 + 3 \times 1 + 0 \times 0 + 1 \times 1 + 2 \times 2 + 3 \times 1 + 0 \times 0 + 1 \times 2 = 15$$

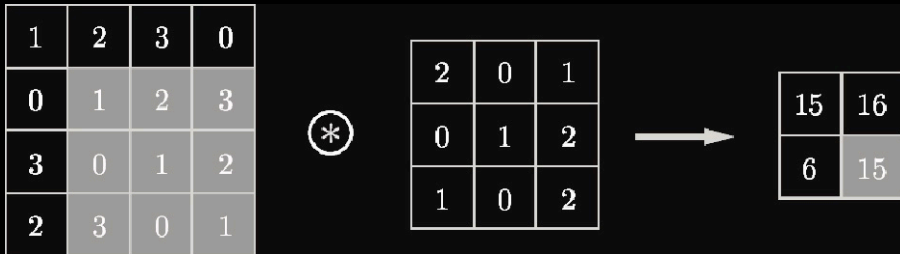
합성곱 연산 예시



합성곱 연산 예시



합성곱 연산 예시



합성곱 연산 예시 (편향 추가)

형상

(스트라이드 1)

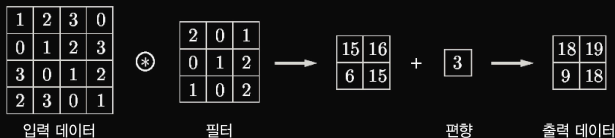
입력 데이터 (4, 4)

필터 (3, 3)

편향 (1, 1)

출력 (2, 2)

그림 7-5 합성곱 연산의 편향 : 필터를 적용한 원소에 고정값(편향)을 더한다.



패딩(padding)

입력 데이터 주변을 특정 값(주로 0)으로 채우는 것

관찰

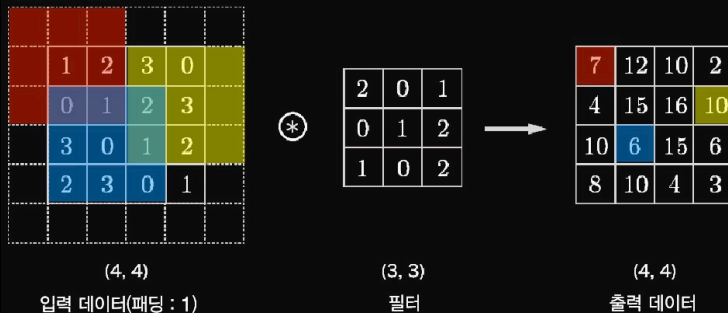
합성곱 연산을 거칠 때마다 출력 크기가 줄어든다.

패딩의 효과

입력 데이터의 크기를 줄이지 않고 다음 계층에 전달할 수 있다.

합성곱 연산의 패딩 처리 예시

그림 7-6 합성곱 연산의 패딩 처리 : 입력 데이터 주위에 0을 채운다(패딩은 점선으로 표시했으며 그 안의 값 '0'은 생략했다).



스트라이드(stride)

필터를 적용하는 위치의 간격

그림 7-7 스트라이드가 2인 합성곱 연산

1	2	3	0	1	2	3
0	1	2	3	0	1	2
3	0	1	2	3	0	1
2	3	0	1	2	3	0
1	2	3	0	1	2	3
0	1	2	3	0	1	2
3	0	1	2	3	0	1

⊗

2	0	1
0	1	2
1	0	2



15		

스트라이드 : 2



1	2	3	0	1	2	3
0	1	2	3	0	1	2
3	0	1	2	3	0	1

2	0	1
---	---	---

15	17	
----	----	--

출력 크기

- 패딩이 커지면 출력 크기가 커진다.
- 스트라이드가 커지면 출력 크기가 작아진다.

입력 크기 (H, W), 필터 크기 (FH, FW), 패딩 P, 스트라이드 S
⇒ 출력 크기 (OH, OW)

$$OH = \frac{H + 2P - FH}{S} + 1, \quad OW = \frac{W + 2P - FW}{S} + 1$$

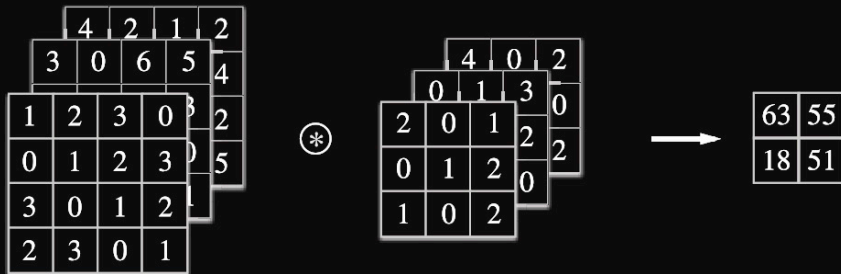
지금까지 본 것

세로, 가로 (2차원)

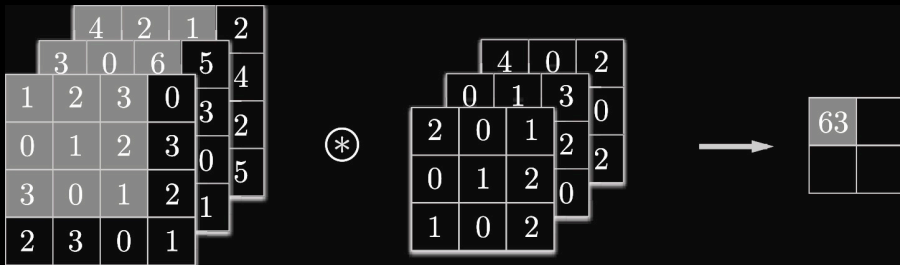
앞으로 볼 것

채널, 세로, 가로 (3차원)

그림 7-8 3차원 데이터 합성곱 연산의 예



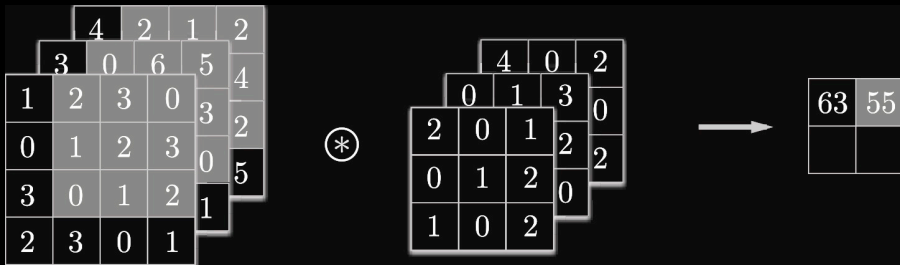
3차원 데이터의 합성곱 연산 예시



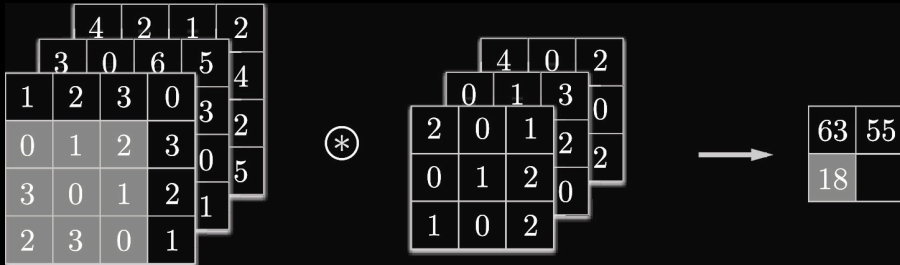
주의

- 필터의 크기는 모든 채널에서 똑같아야 한다.
- 필터의 채널 수는 입력 데이터의 채널 수와 같아야 한다.

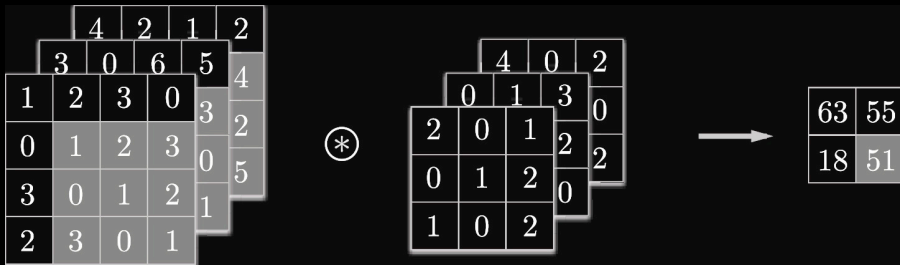
3차원 데이터의 합성곱 연산 예시



3차원 데이터의 합성곱 연산 예시

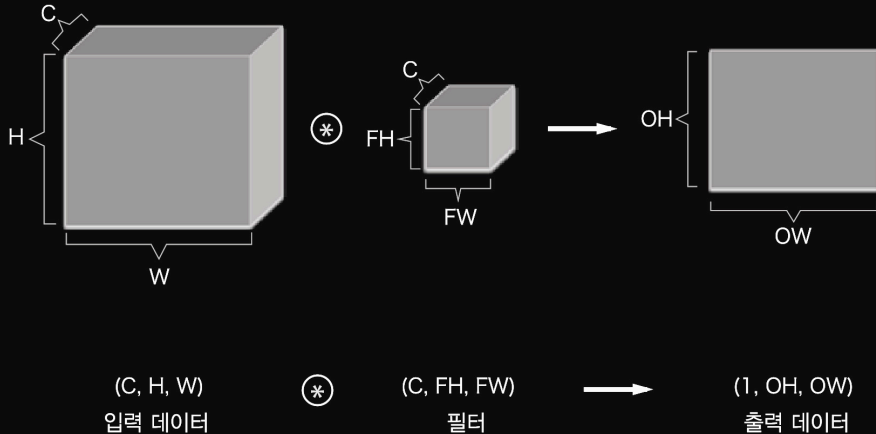


3차원 데이터의 합성곱 연산 예시



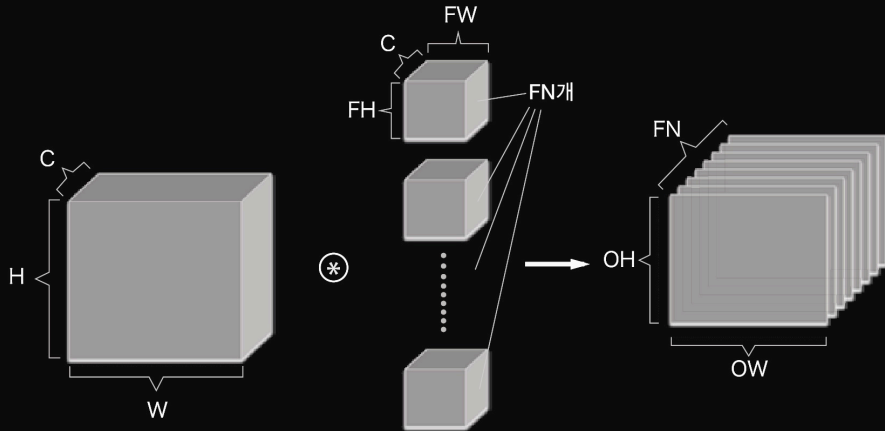
필터 1개 → 출력 데이터의 채널이 1개

그림 7-10 합성곱 연산을 직육면체 블록으로 생각한다. 블록의 형상에 주의할 것!



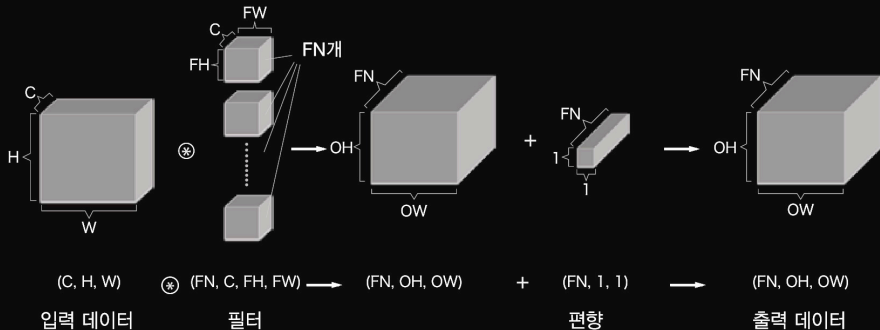
필터 FN개 \rightarrow 출력 데이터의 채널 FN개

그림 7-11 여러 필터를 사용한 합성곱 연산의 예



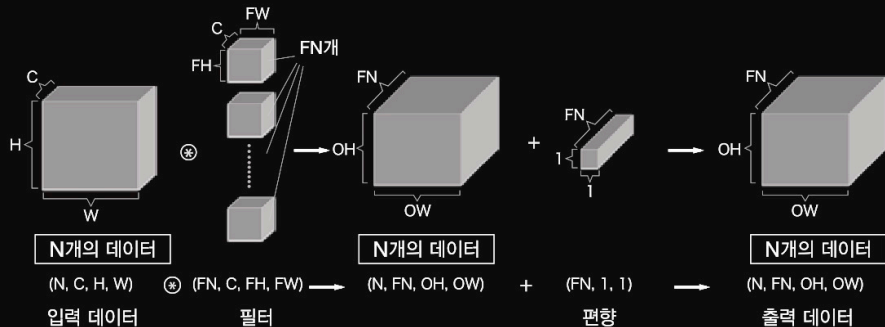
합성곱 연산의 처리 흐름(편향 추가)

그림 7-12 합성곱 연산의 처리 흐름(편향 추가)

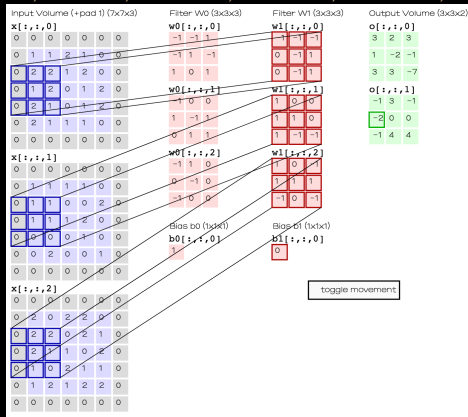


합성곱 연산의 처리 흐름(배치 처리)

그림 7-13 합성곱 연산의 처리 흐름(배치 처리)



합성곱 연산 데모 (배치 처리 생략)

<http://cs231n.github.io/convolutional-networks/>
 $C = 3, H = 5, W = 5, FN = 2, FH = 3, FW = 3, P = 1, S = 2$


풀링(pooling)

영역을 원소 하나로 집약하여 공간 크기를 줄이는 연산

- 최댓값이나 평균을 주로 사용한다.

풀링의 원도 크기가 2×2 인 경우 스트라이드를 2로 설정한다.

그림 7-14 최대 풀링의 처리 순서

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	3

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	3
4	

1	2	1	0
0	1	2	3
3	0	1	2
2	4	0	1



2	3
4	2

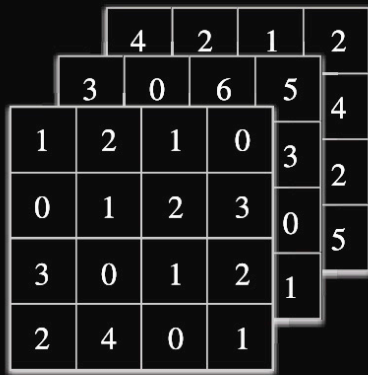
풀링 계층의 특징

- 1 학습해야 할 매개변수가 없다.
 - 최댓값이나 평균을 취하면 되므로 학습할 것이 없다.
- 2 채널 수가 변하지 않는다.
 - 채널마다 독립적으로 최댓값이나 평균을 계산한다.
- 3 입력의 변화에 영향을 적게 받는다(강건하다).
 - 입력 데이터의 조금 변해도 (특히 최댓값의 경우) 풀링의 결과는 잘 변하지 않는다.

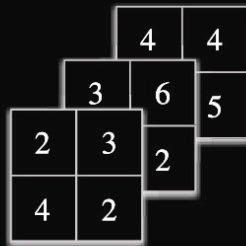
풀링 계층의 특징 (2)

채널 수가 변하지 않는다.

그림 7-15 풀링은 채널 수를 바꾸지 않는다.



입력 데이터



출력 데이터

강건하다.

그림 7-16 입력 데이터가 가로로 1원소만큼 어긋나도 출력은 같다(데이터에 따라서는 다를 수도 있다).



9	7
6	8

다음 시간에 할 일

- CNN으로 텍스트를 분류하는 방법을 알아본다.
- Kim (2014) 논문을 읽어 온다.