

# 5장 합성곱 신경망 1

⌚ 생성일	@2024년 9월 22일 오후 4:06
≡ 주차	Week 3
☑ 완료여부	<input type="checkbox"/>

## 5.1 합성곱 신경망

딥러닝 순전파 과정에서 계산된 오차 정보가 신경망의 모든 노드(출력층→은닉층→입력층)으로 전송되는데 계산과정이 많고 시간문제를 해결하고자 함

이미지 전체를 한번에 계산하지 않고 국소적 부분을 계산함으로써 시간과 자원 절약

### 5.1.1 합성곱층의 필요성

합성곱 신경망은 이미지, 영상 처리하는데 유용함

이미지를 펼쳐서 분석하면 공간적인 정보 소실 → 합성곱층 도입

### 5.1.2 합성곱 신경망 구조 Convolutional Neural Network

#### 1. 입력층

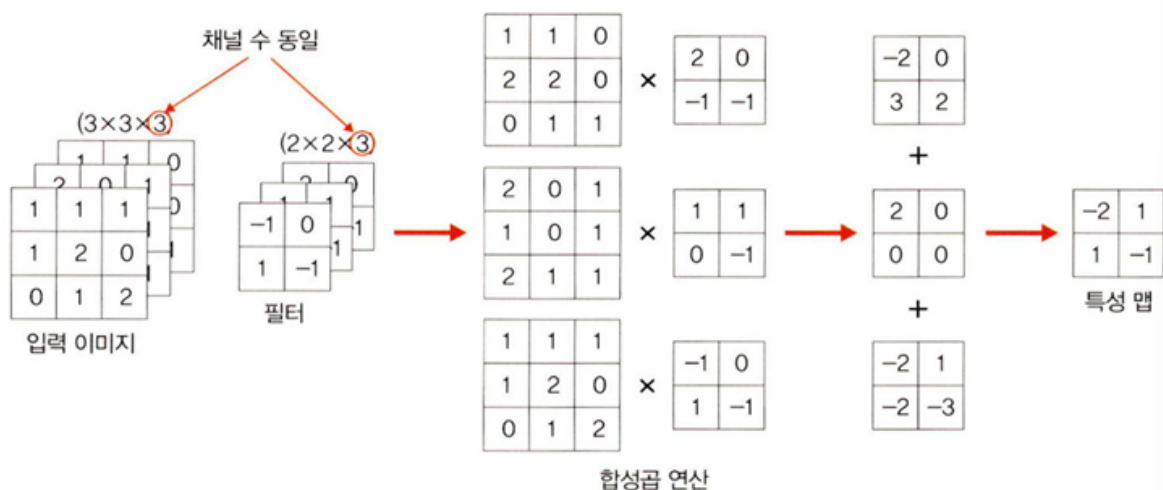
이미지: height, weight, channel의 3차원 데이터로 channel은 흑백이면 1, 컬러면 3임

#### 2. 합성곱층

kernel이나 filter를 사용하여 입력데이터의 특성을 추출하는 역할 → feature map 생성

이때 커널의 일반적인 크기는 3x3, 5x5, 지정된 간격 stride에 따라 이동함

♥ 그림 5-10 컬러 이미지 합성곱



입력데이터:  $W \times H \times D$  이때  $D$ 는 채널의 수

하이퍼파라미터: 필터개수  $K$ , 필터크기  $F$ , 스트라이드  $S$ , 패딩  $P$

출력 데이터

$$\bullet W_2 = (W_1 - F + 2P) / S + 1$$

$$\bullet H_2 = (H_1 - F + 2P) / S + 1$$

$$\bullet D_2 = K$$

출력데이터 계산식

### 3. 풀링층 Pooling Layer

feature map의 차원을 다운 샘플링하여 연산량을 감소시키고 주요한 특성 벡터를 추출함

최대풀링 → 대상 영역에서 최댓값 추출, 주로 사용됨

평균풀링 → 대상 영역에서 평균값 추출

입력데이터:  $W \times H \times D$  이때  $D$ 는 채널의 수

하이퍼파라미터: 필터크기  $F$ , 스트라이드  $S$

출력 데이터

$$\bullet W_2 = (W_1 - F) / S + 1$$

$$\bullet H_2 = (H_1 - F) / S + 1$$

$$\bullet D_2 = D_1$$

출력 데이터 계산식

### 4. 완전연결층 Fully Connected Layer

이미지(3차원 벡터)가 1차원 벡터로 됨 `flatten()`

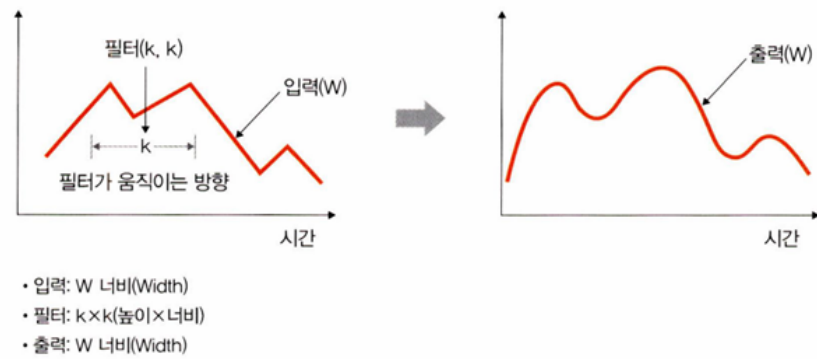
5.1.3 1D, 2D, 3D 합성곱 ← 출력 형태에 따라 분류

#### 1. 1D 합성곱

필터가 시간을 기준으로 좌우로만 이동할 수 있는 합성곱

따라서 출력 형태는 1D 배열이 됨

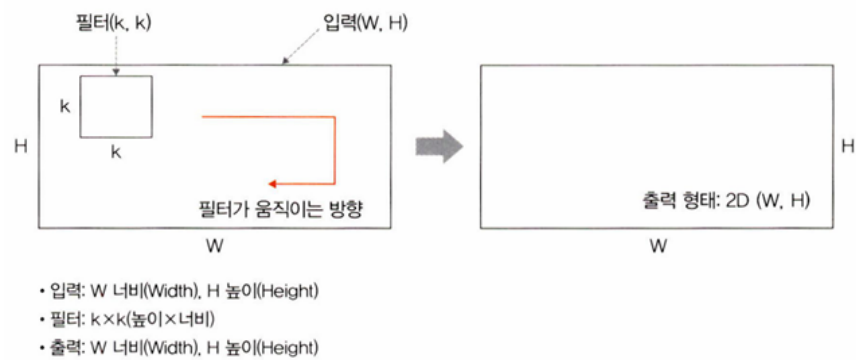
♥ 그림 5-19 1D 합성곱



## 2. 2D 합성곱

필터가 2개의 방향으로 움직임. 따라서 출력 형태는 2D 형태

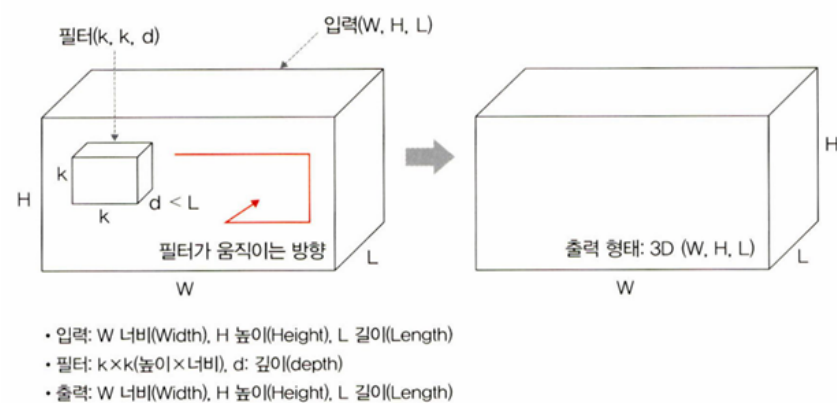
♥ 그림 5-20 2D 합성곱



## 3. 3D 합성곱

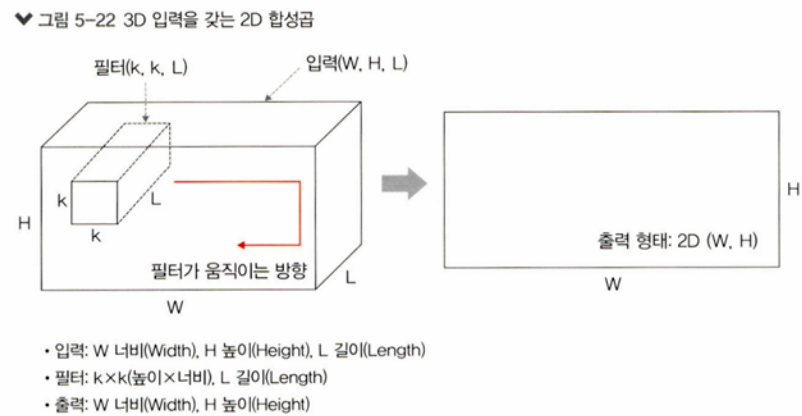
필터가 3개의 방향으로 움직임. 이때 필터의 깊이 < 입력의 길이가 되어야 함

♥ 그림 5-21 3D 합성곱



## 4. 3D 입력을 갖는 2D

입력( $W, H, L$ )에 필터( $k, k, L$ )를 적용하면 출력은 ( $W, H$ )가 됨. 이때 필터는 두 방향으로 움직이고 출력형태는 2D 행렬이 됨



## 5. $1 \times 1$ 합성곱

입력( $W, H, L$ )에 필터 ( $1, 1, L$ )를 적용하면 출력은 ( $W, H$ )로 3D 형태로 출력됨. 대표적인 사례는 Google Net으로 채널수를 조정해서 연산량이 감소되는 효과가 있음