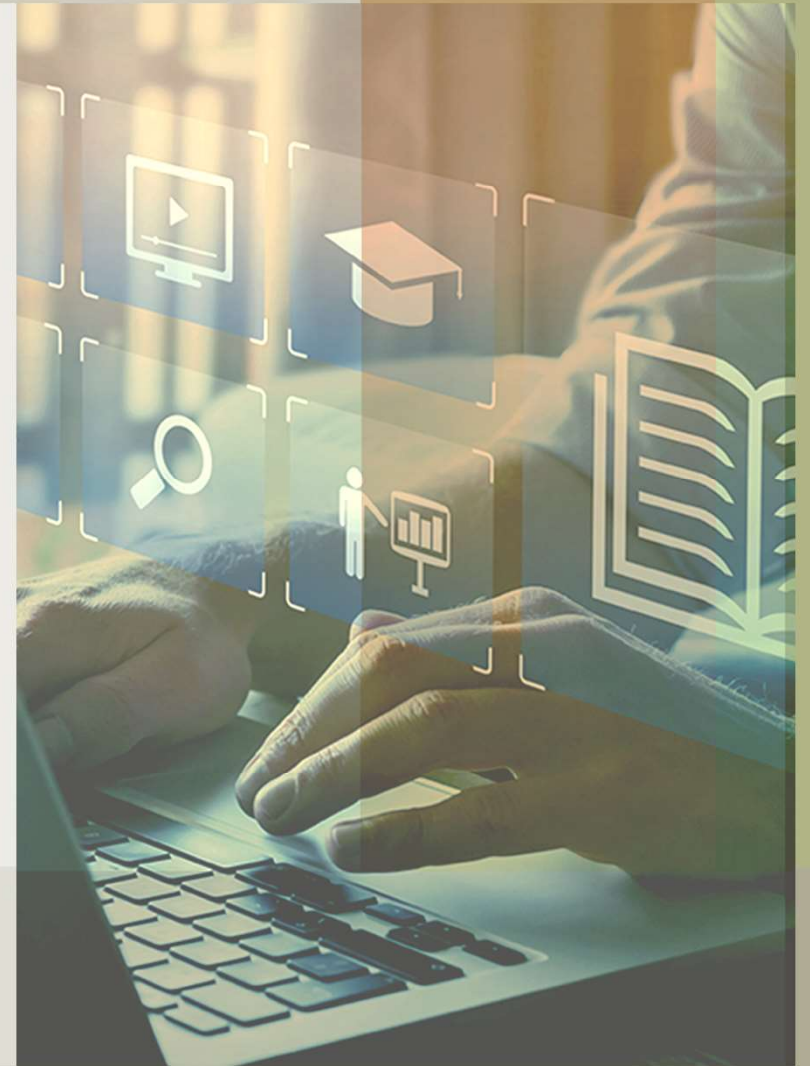


06

답러닝

합성곱 신경망(1)

방송대 우호성 교수



학습목차

- 1 합성곱의 이해
- 2 합성곱 신경망의 구조



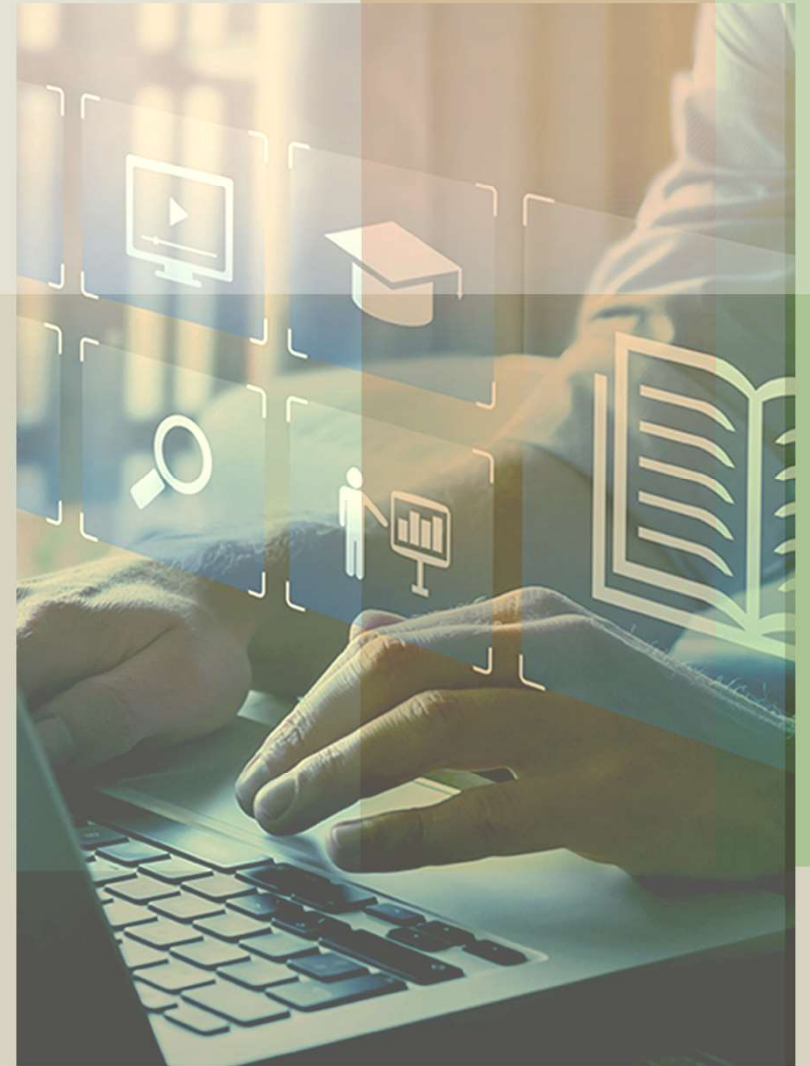
학습목표

- ① 합성곱의 개념을 설명할 수 있다.
- ② 합성곱 신경망에서 풀링의 기능을 설명할 수 있다.
- ③ 완전연결층에서 수행되는 기능을 설명할 수 있다.

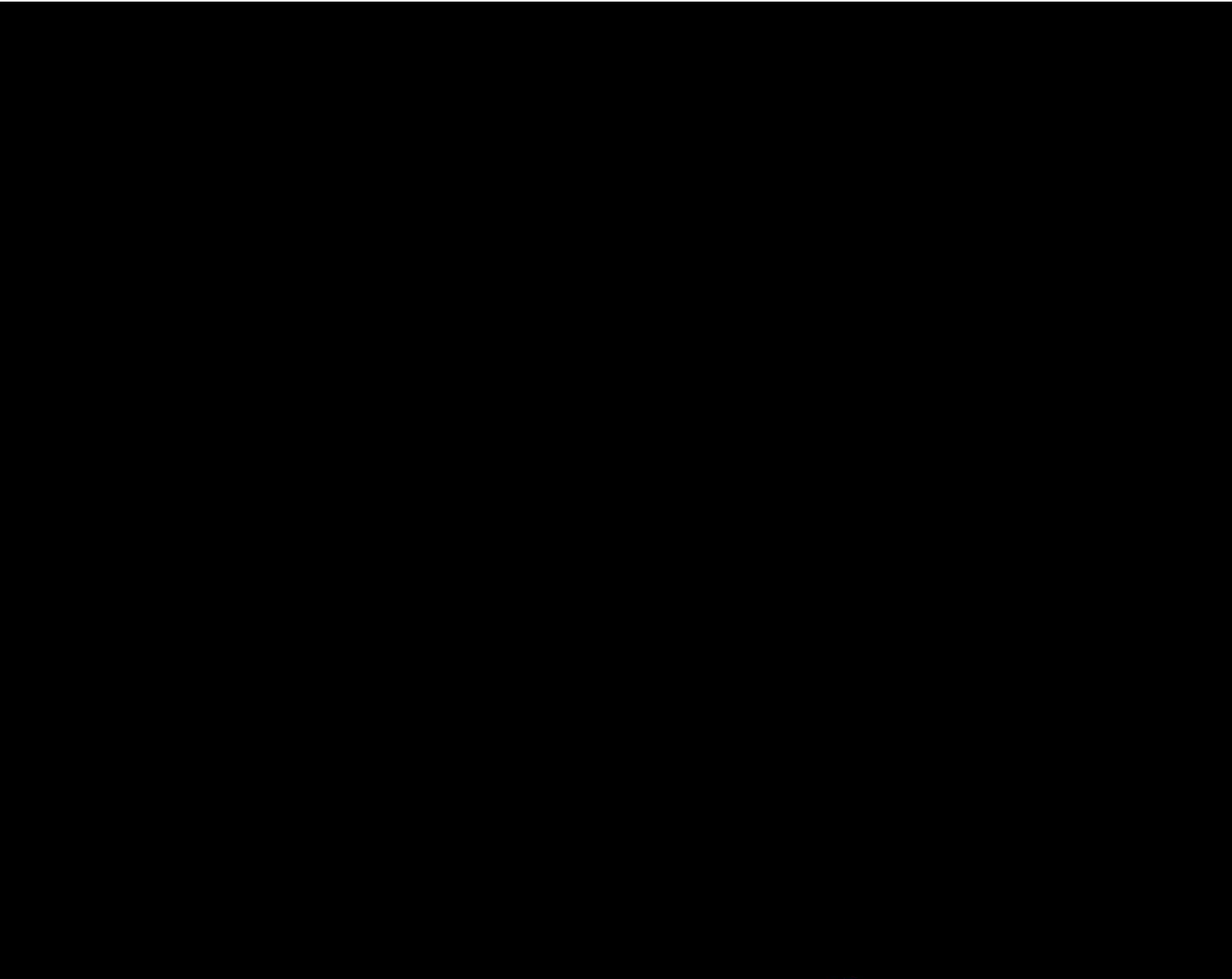


01

합성곱의 이해



Convolution Network Demo from 1993



1. 영상처리란?

● 영상처리

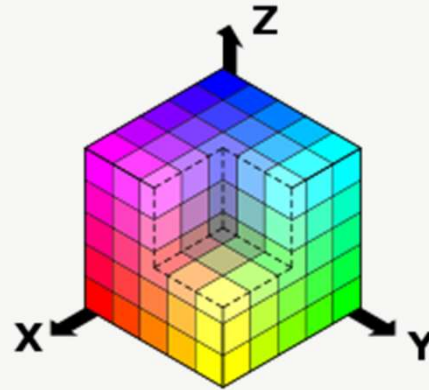
- 디지털 이미지의 조작, 분석, 인식, 생성 등의 목적을 달성하기 위한 기술
- 합성곱 신경망(Convolutional Neural Network)은 이미지를 기반으로 한 영상처리 작업에서 우수한 성능을 보이는 신경망



2. 이미지의 구조

● 컬러(RGB) 이미지 색상

- 빨강, 초록, 파랑 세 가지 색의 빛을 0~255의 다양한 강도로 섞어서 여러 가지 색을 표현



(가) 컬러(RGB) 채널

● 그레이 스케일 이미지 색상

- 색상 정보 없이 오직 밝기 정보만으로 표현



(나) 그레이스케일 채널



2. 이미지의 구조

● 컬러(RGB) 이미지 픽셀 값



메모리 1

주소: 0x000001F264C4C4B0

0x000001F264C4C4B0	e0 ee f7 ff dc ea f5 ff d5 e5 f2 ff d0 e0 f0 ff
0x000001F264C4C4C0	cb db f2 ff c8 d9 f3 ff c7 d7 f8 ff c3 d7 fa ff
0x000001F264C4C4D0	c3 d7 fc ff c1 d9 ff ff c1 d8 ff ff bf d8 ff ff
0x000001F264C4C4E0	bd d9 fe ff be da ff ff bf dc fe ff c0 dd ff ff
0x000001F264C4C4F0	c0 dc ff ff c3 dd ff ff c6 df fd ff ca e0 f8 ff
0x000001F264C4C500	cc de f2 ff d0 e2 f0 ff de ec f5 ff ea f9 ff ff
0x000001F264C4C510	ed fc ff ff ec fd ff ff e6 f8 ff ff da f0 fe ff
0x000001F264C4C520	d3 ec ff ff d1 ed ff ff c7 e6 ff ff bb dd f9 ff
0x000001F264C4C530	bd dc fb ff ba d9 f8 ff bb d9 fb ff bf dd ff ff
0x000001F264C4C540	c3 e2 ff ff c5 e4 ff ff cb e6 ff ff cf e9 ff ff

● 그레이스케일 이미지 픽셀 값



메모리 1

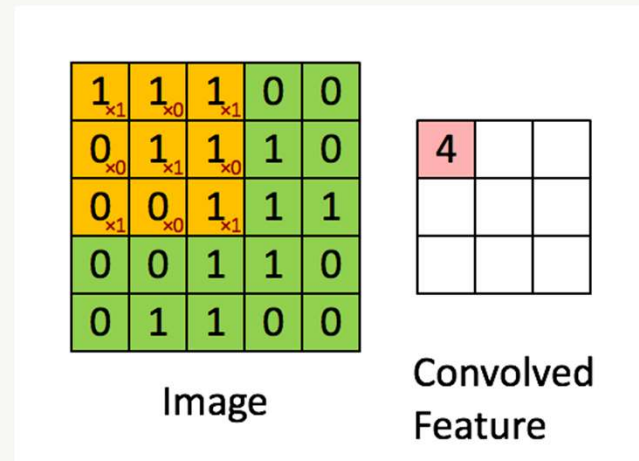
주소: 0x0000028125F6DC30

0x0000028125F6DC30	ec ec ec ff e8 e8 e8 ff e4 e4 e4 ff e0 e0 e0 ff
0x0000028125F6DC40	dd dd dd ff dc dc dc ff dc dc dc ff dc dc dc ff
0x0000028125F6DC50	dc dc dc ff dd dd dd ff dd dd dd ff dc dc dc ff
0x0000028125F6DC60	dc dc dc ff dc dc dc ff dd dd dd ff dd dd dd ff
0x0000028125F6DC70	de de de ff df df df ff e0 e0 e0 ff e1 e1 e1 ff
0x0000028125F6DC80	de de de ff e0 e0 e0 ff ea ea ea ff f5 f5 f5 ff
0x0000028125F6DC90	f8 f8 f8 ff f8 f8 f8 ff f3 f3 f3 ff ed ed ed ff
0x0000028125F6DCA0	ea ea ea ff e9 e9 e9 ff e4 e4 e4 ff db db db ff
0x0000028125F6DCB0	db db db ff d9 d9 d9 ff da da da ff de de de ff
0x0000028125F6DCC0	e1 e1 e1 ff e2 e2 e2 ff e5 e5 e5 ff e7 e7 e7 ff

3. 합성곱 연산

○ 합성곱

- 합성곱(Convolution)은 영상처리 기술에서 가장 기본적인 연산
- 합성곱 연산은 필터링 연산이라고 불리우기도 함
- 하나의 함수와 또 다른 함수를 반전 이동한 값을 곱한 다음, 구간에 대해 적분하여 새로운 함수를 구하는 수학 연산자(위키피디아)



3. 합성곱 연산의 적용 사례

- 포토샵, 사진 어플리케이션의 필터
 - 이미지에 서로 다른 고정된 가중치 적용을 통해 윤곽선 검출/블러/샤arp/엠바스 등 다양한 효과를 나타낼 수 있음



3. 합성곱 연산의 적용 사례

● 포토샵, 사진 어플리케이션의 필터



(가) 원본 이미지



-1	-1	-1
-1	8	-1
-1	-1	-1

(나) 윤곽선 검출 및 필터



0.0625	0.125	0.0625
0.125	0.25	0.125
0.0625	0.125	0.0625

(다) 블러 및 필터



0	-1	0
-1	5	-1
0	-1	0

(라) 샤프 및 필터

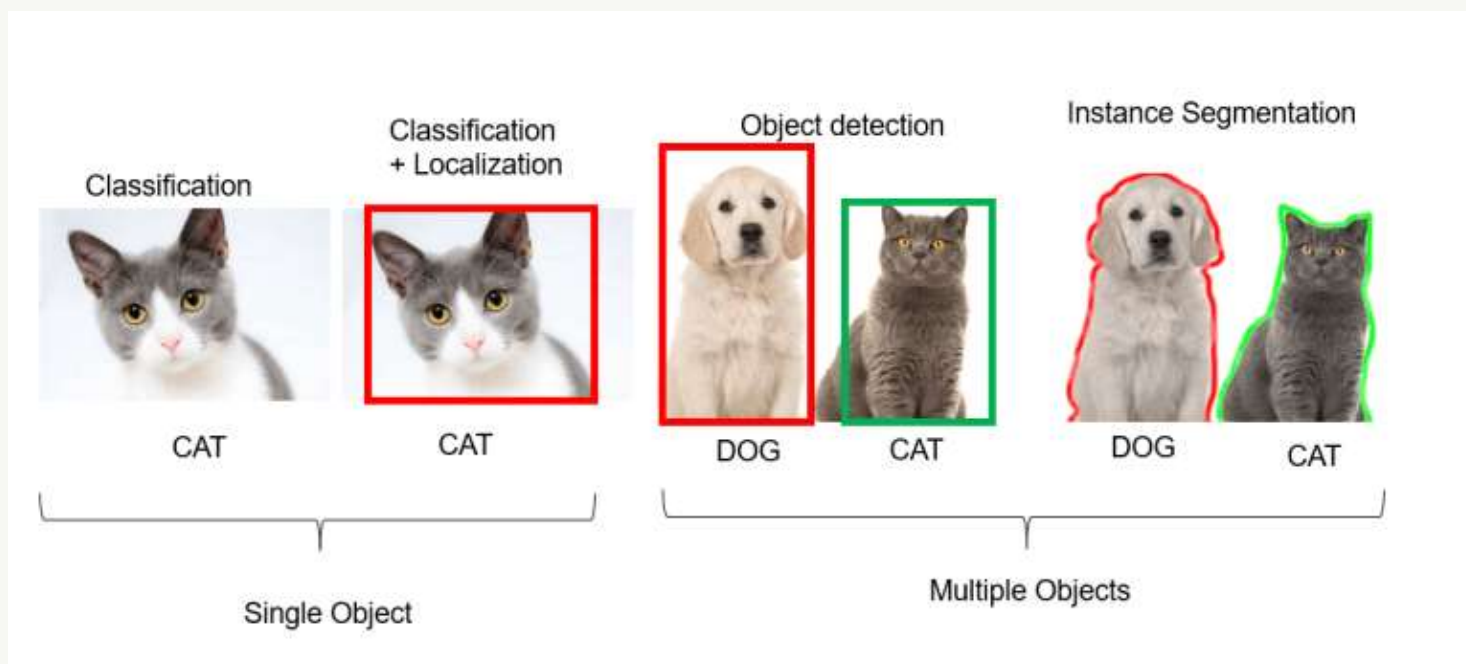


-2	-1	0
-1	1	1
0	1	2

(마) 엠바스 및 필터

3. 합성곱 연산의 적용 사례

• 다양한 객체 구분



3. 합성곱 연산의 적용 사례

● 다양한 객체 구분

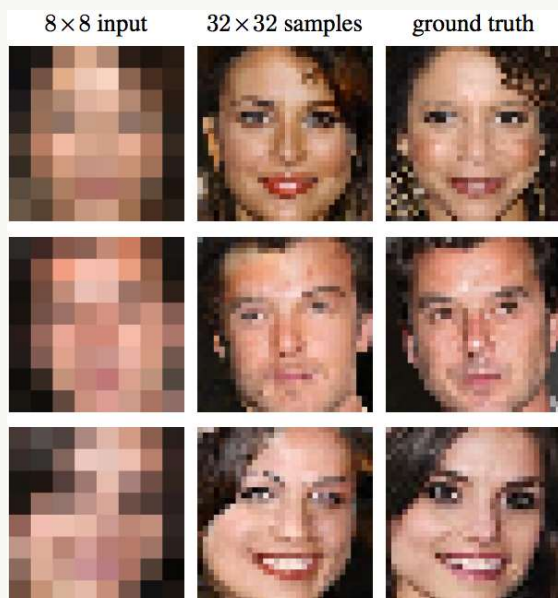
- 분류 : 이미지 안의 하나의 객체가 존재할 때, 하나의 범주로 구분하는 것
- 분류+지역화 : 이미지 안의 하나의 객체가 존재할 때, 하나의 객체가 어느 위치에 있는지 위치 정보를 출력하여 주는 것
- 객체 감지 : 이미지 안의 여러 개의 객체가 존재할 때, 각 객체의 존재 여부 파악 및 위치 정보를 출력하여 주는 것
- 인스턴스 분할 : 이미지 안의 픽셀 수준에서 객체별 형상에 따라 영역을 표시하는 것. 이때, 같은 범주의 객체여도 서로 다른 객체로 구분 (예: Dog1, Dog2, Dog3, Cat1, Cat2 등)



3. 합성곱 연산의 적용 사례

픽셀 복원

- 이미지를 저해상도로 변환시킨 후 각 이미지가 무엇과 유사한 형태를 보이는지 예측



출처 : <https://www.youtube.com/watch?v=3uoM5kfZlQ0>

3. 합성곱 연산의 적용 사례

● 색복원

- 흑백의 영상을 색이 있는 영상으로 변경

100 year old pictures...



출처 : <https://www.youtube.com/watch?v=ys5nMO4Q0iY>



4. 합성곱 신경망의 필터

- 합성곱 신경망에서 어떻게 필터가 적용될까?
 - 이미지에 효과를 주는 기존의 필터링 기법은 고정된 필터를 통해 처리
 - 합성곱 신경망에서는 고정된 필터를 사용하지 않음
 - 필터의 가중치를 모르는 상태에서 다양한 입력 데이터를 이용한 학습 과정을 통해 자동으로 가중치가 결정

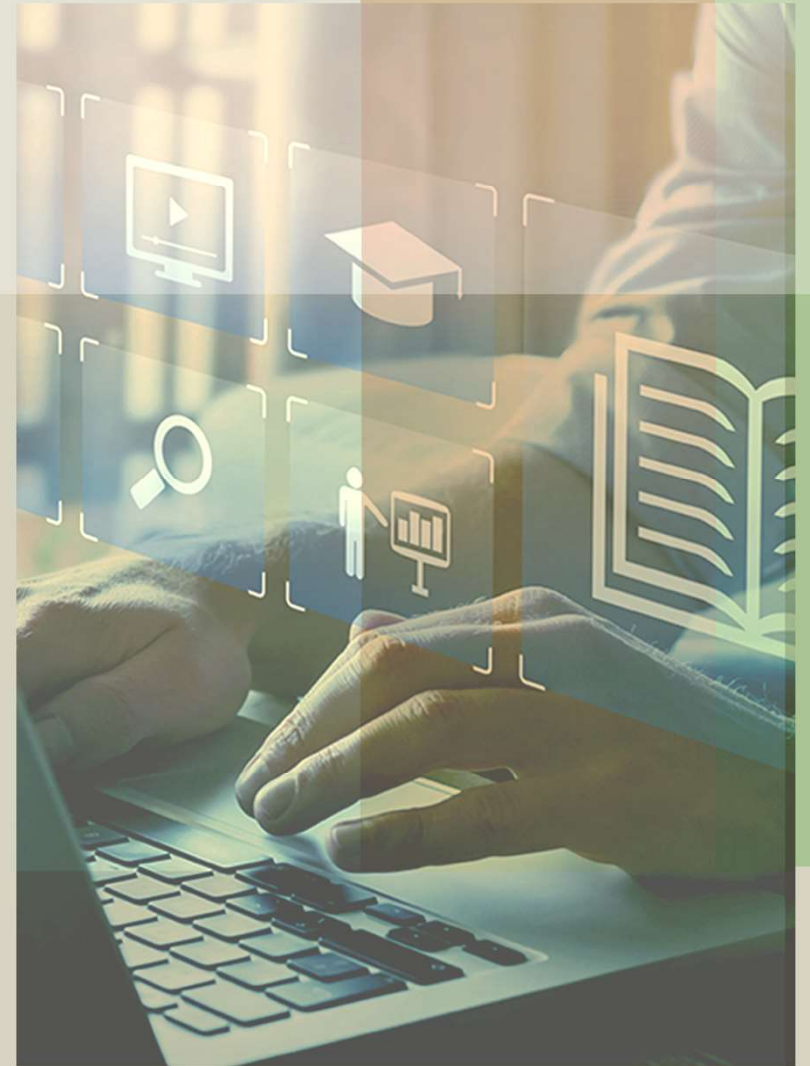


$$\begin{matrix} & + & \begin{array}{|c|c|c|} \hline ? & ? & ? \\ \hline ? & ? & ? \\ \hline ? & ? & ? \\ \hline \end{array} & = & ?
 \end{matrix}$$



02

합성곱 신경망의 구조



1. 합성곱 신경망과 MLP

○ 합성곱 신경망 VS MLP

합성곱 신경망

- 입력 데이터의 공간적인 특징을 추출하는 데 특화되었으며, 정보 손실이 적음
- 필터의 가중치 공유와 풀링층을 통한 데이터 크기 축소로 인해 연산이 효율적

MLP

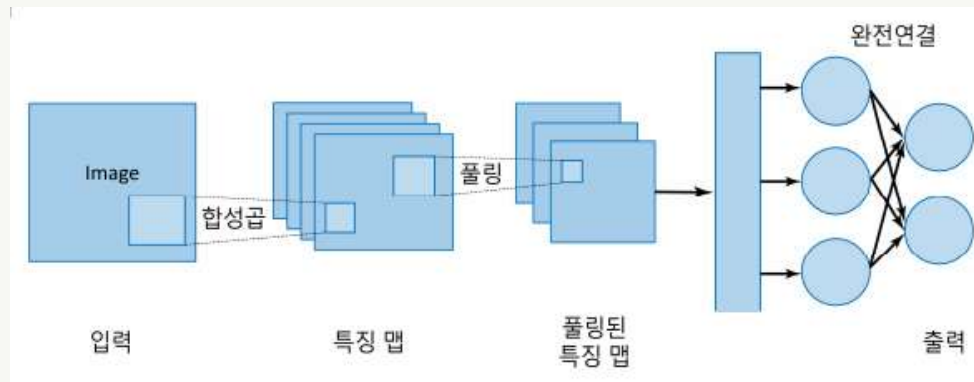
- 입력 데이터를 1차원 배열로 만든 후 신경망의 입력으로 사용하여 가중치를 계산하기 때문에 정보 손실이 큼



2. 합성곱 신경망의 주요 개념

● 합성곱 신경망 개념도

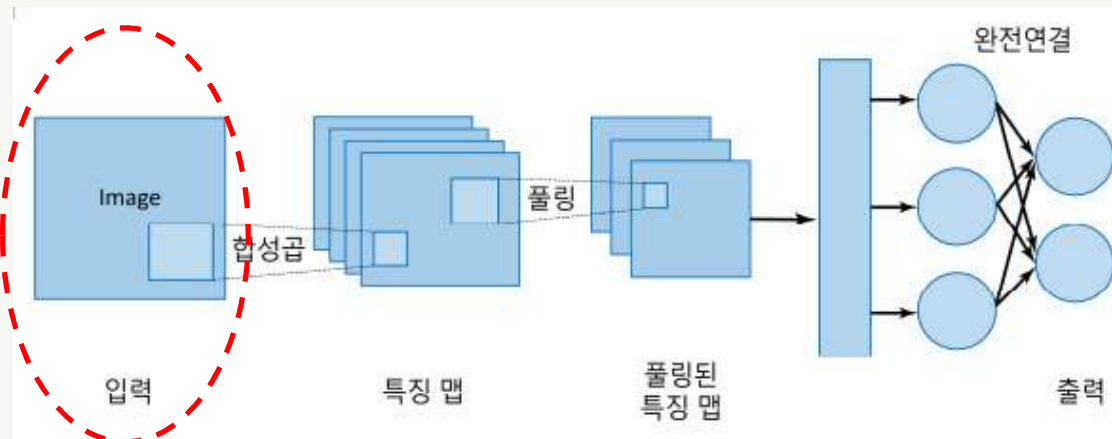
- 입력 : 입력 데이터
- 합성곱층 : 데이터에서 특징을 추출
- 풀링층 : 데이터의 차원을 축소하여, 연산량 감소 및 특징 강화
- 완전연결층 : 합성곱층과 풀링층을 거치면서 차원이 축소된 특징 맵을 1차원으로 변환. 소프트맥스를 통해 확률값으로 변환하여 출력



2. 합성곱 신경망의 주요 개념

● 입력

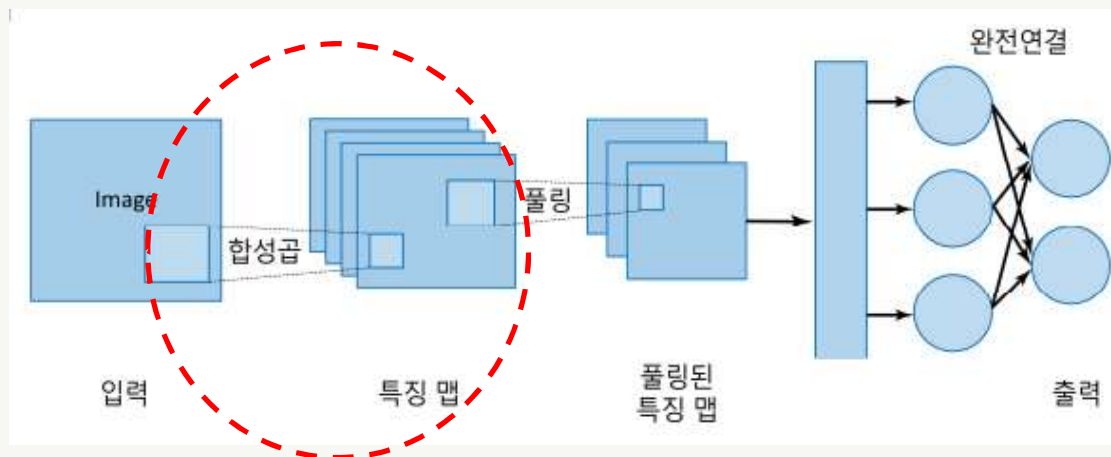
- 데이터가 합성곱 신경망에서 처리될 수 있도록 변환되는 단계
- 데이터는 높이, 너비, 채널의 값인 3차원으로 구성
- 컬러(RGB)는 3개의 채널, 그레이스케일은 1개의 채널
- (높이, 너비, 채널 수)로 표현



2. 합성곱 신경망의 주요 개념

● 합성곱층

- 합성곱 신경망에서 가장 많은 연산이 처리되는 계층
- 이전 층에서 전달받은 입력 데이터와 필터(커널)를 통해 연산
- 필터의 크기는 전통적으로 정방행렬의 3X3, 5X5, 7X7 등 홀수로 구성
- 합성곱층의 산출물을 특징 맵이라고 부름

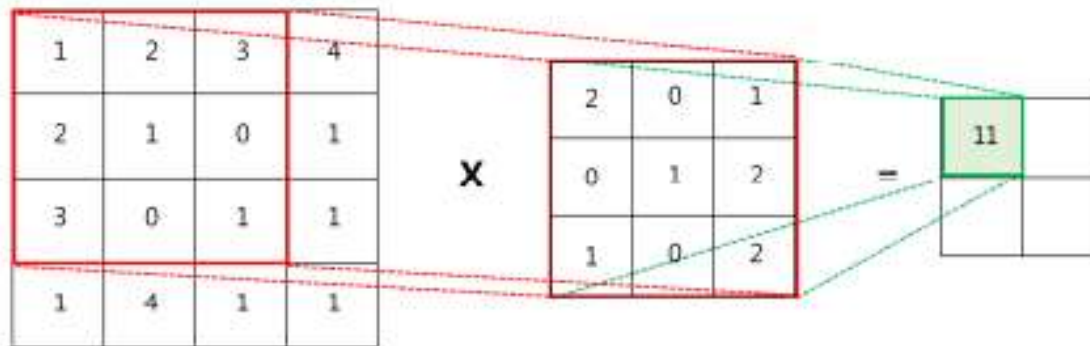


2. 합성곱 신경망의 주요 개념

● 1개 채널의 합성곱 연산의 예

- 1개 채널을 갖고 있는 4x4 크기의 데이터(4, 4, 1)에 3x3 크기의 필터를 적용한 합성곱 연산 과정

1단계



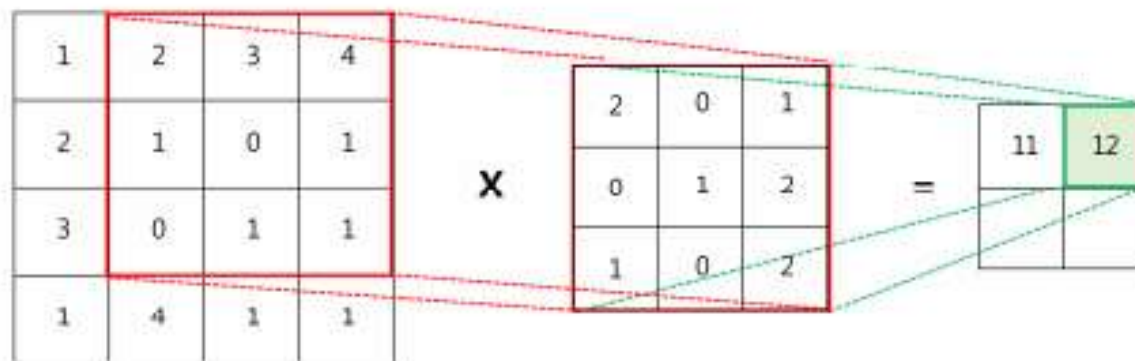
$$1 \times 2 + 2 \times 0 + 3 \times 1 + 2 \times 0 + 1 \times 1 + 0 \times 2 + 3 \times 1 + 0 \times 0 + 1 \times 2 = 11$$

2. 합성곱 신경망의 주요 개념

● 1개 채널의 합성곱 연산의 예

- 1개 채널을 갖고 있는 4x4 크기의 데이터(4, 4, 1)에 3x3 크기의 필터를 적용한 합성곱 연산 과정

2단계



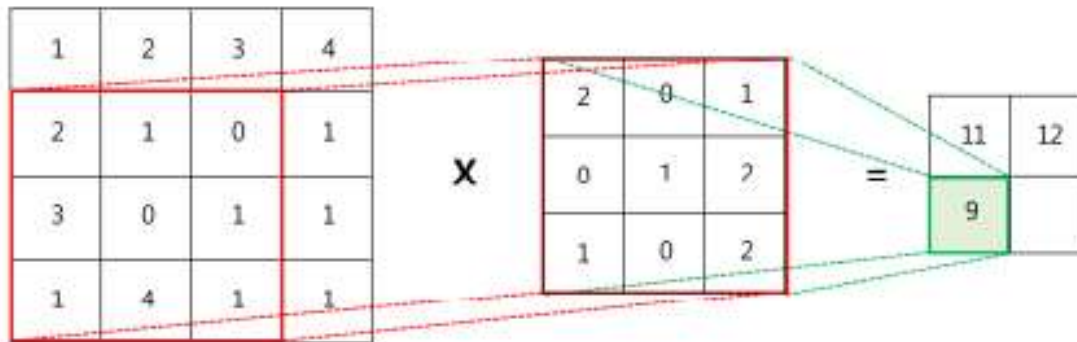
$$2 \times 2 + 3 \times 0 + 4 \times 1 + 1 \times 0 + 0 \times 1 + 1 \times 2 + 0 \times 1 + 1 \times 0 + 1 \times 2 = 12$$

2. 합성곱 신경망의 주요 개념

● 1개 채널의 합성곱 연산의 예

- 1개 채널을 갖고 있는 4x4 크기의 데이터(4, 4, 1)에 3x3 크기의 필터를 적용한 합성곱 연산 과정

3단계

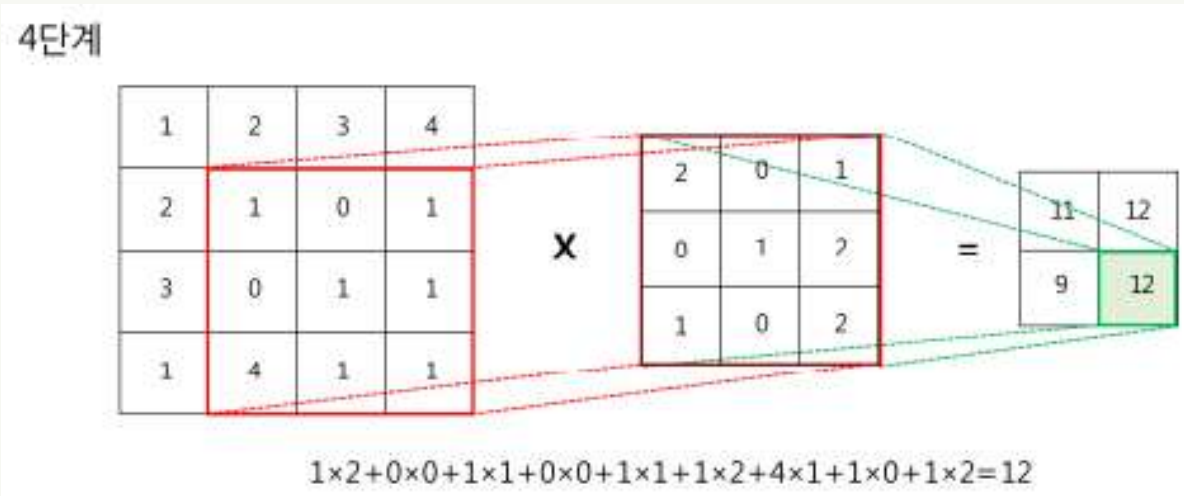


$$2 \times 2 + 1 \times 0 + 0 \times 1 + 3 \times 0 + 0 \times 1 + 1 \times 2 + 1 \times 1 + 4 \times 0 + 1 \times 2 = 9$$

2. 합성곱 신경망의 주요 개념

● 1개 채널의 합성곱 연산의 예

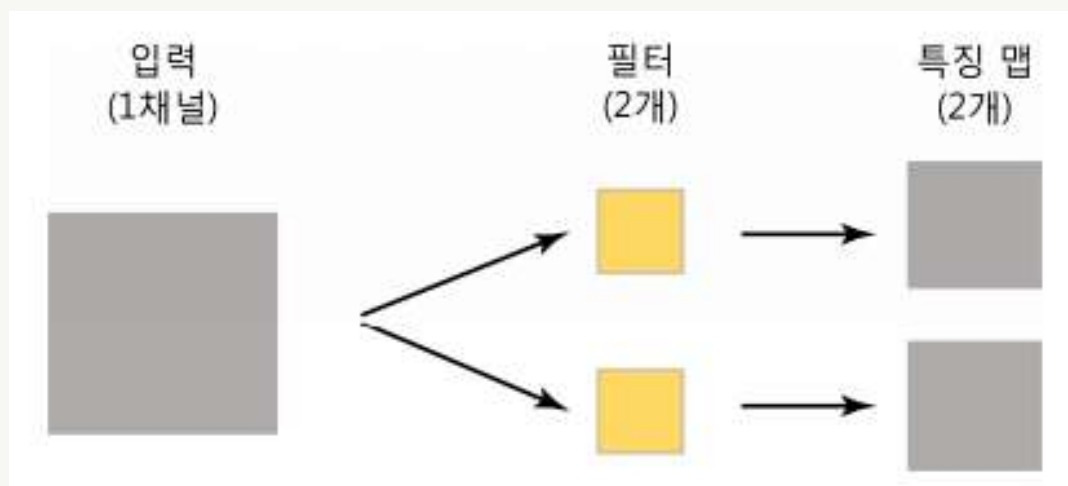
- 1개 채널을 갖고 있는 4x4 크기의 데이터(4, 4, 1)에 3x3 크기의 필터를 적용한 합성곱 연산 과정



2. 합성곱 신경망의 주요 개념

● 1개 채널의 합성곱 연산의 예

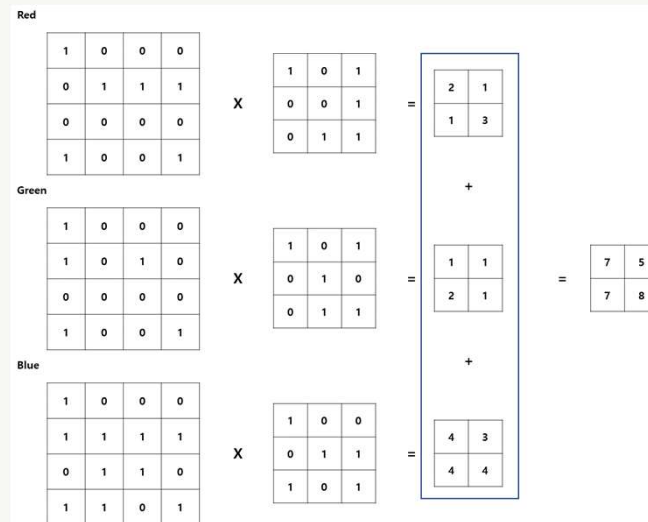
- 그레이스케일 이미지에 필터를 2개 적용하면, 하나의 채널로 구성된 특징 맵 2개



2. 합성곱 신경망의 주요 개념

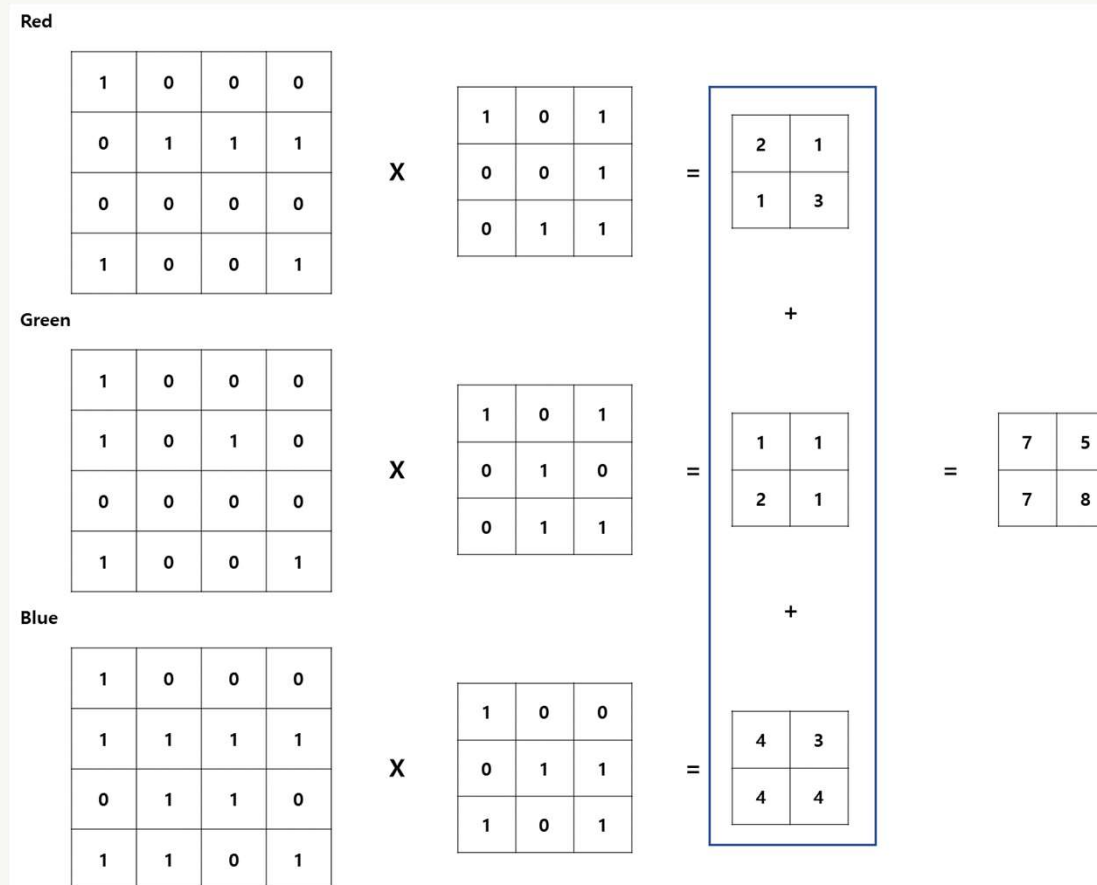
3개 채널의 합성곱 연산의 예

- 3개 채널이 있는 4x4 크기의 데이터(4, 4, 3)에 3x3 크기의 필터를 적용한 합성곱 연산
- 채널의 수가 3개인 경우 1개의 필터는 서로 다른 채널 별로 구성, 필터의 수는 1개



2. 합성곱 신경망의 주요 개념

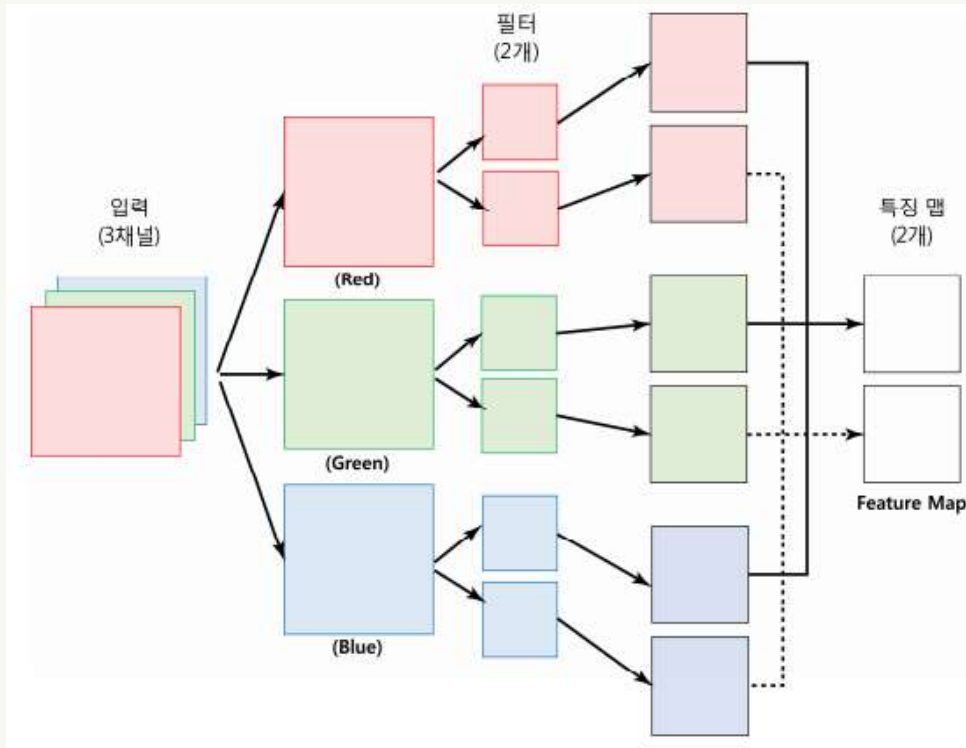
3개 채널의 합성곱 연산의 예



2. 합성곱 신경망의 주요 개념

3개 채널의 합성곱 연산의 예

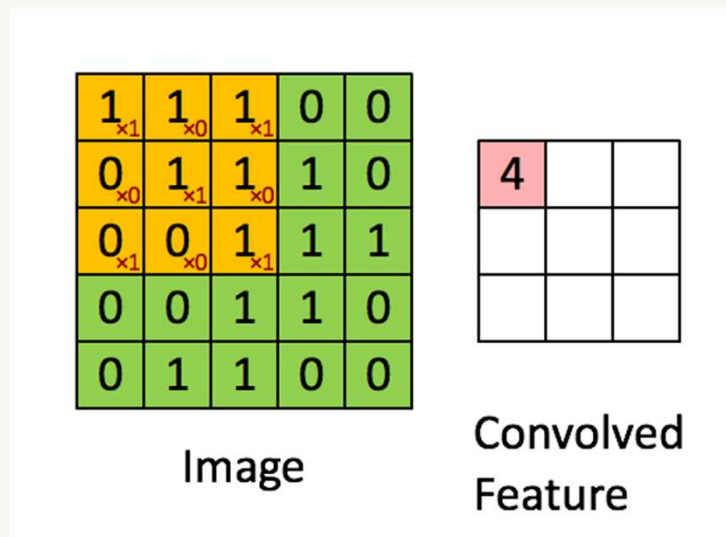
- 컬러 이미지에 필터를 2개 적용하면, 2개의 특징 맵이 만들어 짐



2. 합성곱 신경망의 주요 개념

● 스트라이드

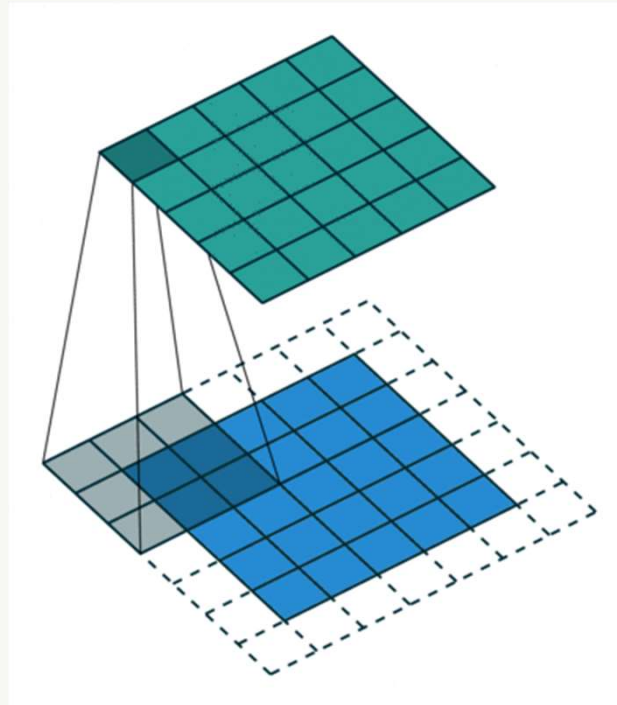
- 입력 데이터에 필터가 적용되어 계산된 후에 왼쪽에서 오른쪽으로 이동
- 이 이동 간격을 '스트라이드'(Stride)라고 부르며, 간격은 조절 가능함
- 합성곱 연산 결과인 특징 맵의 크기는 필터 크기와 스트라이드에 의해 결정



2. 합성곱 신경망의 주요 개념

● 패딩

- 임의의 데이터 주변을 가상의 값으로 채우는 것
- 패딩의 목적은 데이터의 특징을 정밀하게 추출하기 위한 것으로, 합성곱 연산을 마친 후 출력 데이터의 크기를 입력 데이터의 크기와 같게 유지할 수 있음
- 패딩은 선택사항이지만 모서리에 중요한 특징이 있다면 특징을 정밀하게 추출하기 위해 사용

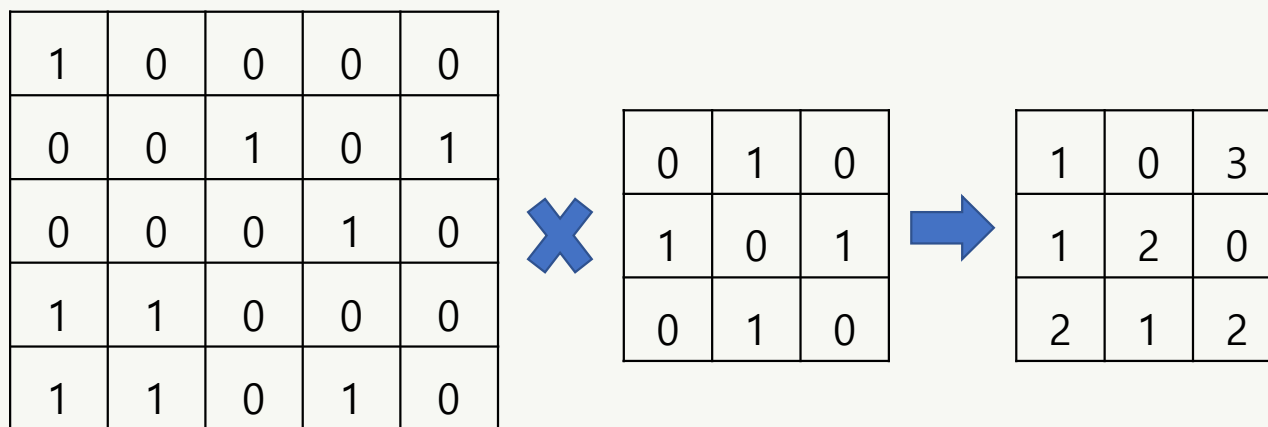


2. 합성곱 신경망의 주요 개념

● 특징 맵의 크기

- 5x5 입력 데이터에 3x3 크기의 필터를 적용
- 스트라이드 1, 패딩의 크기 0
- 입력 데이터는 5x5에서 3x3으로 축소

$$\text{특징 맵의 크기} = \frac{\text{입력된 데이터 크기} - \text{필터 크기} + 2 \times \text{패딩값}}{\text{스트라이드}} + 1$$



2. 합성곱 신경망의 주요 개념

● 합성곱층 코드

```
tf.keras.layers.Conv2D(filters, kernel_size, strides=(1, 1), padding='valid', activation=None)
```

- **filters** : 필터의 수
- **kernel_size** : 필터(커널)의 크기
- **strides** : 필터가 움직이는 간격의 크기(기본 값 = (1, 1))
- **padding** : 가상의 값으로 채우는 것을 의미하며, valid 혹은 same을 사용함.
valid는 패딩이 없음을 의미, same은 입력된 데이터 크기와 출력될 행렬
데이터 크기를 동일하게 맞춰 줌(기본 값 = 'valid')
- **activation** : 사용할 활성화함수(기본 값 = None)



2. 합성곱 신경망의 주요 개념

○ 활성화함수 종류

■ 텐서플로 API 참조 :

https://www.tensorflow.org/api_docs/python/tf/keras/activations

`deserialize(...)` : Returns activation function given a string identifier.

`elu(...)` : Exponential Linear Unit.

`exponential(...)` : Exponential activation function.

`gelu(...)` : Applies the Gaussian error linear unit (GELU) activation function.

`get(...)` : Returns function.

`hard_sigmoid(...)` : Hard sigmoid activation function.

`linear(...)` : Linear activation function (pass-through).

`mish(...)` : Mish activation function.

`relu(...)` : Applies the rectified linear unit activation function.

`selu(...)` : Scaled Exponential Linear Unit (SELU).

`serialize(...)` : Returns the string identifier of an activation function.

`sigmoid(...)` : Sigmoid activation function, $\text{sigmoid}(x) = 1 / (1 + \exp(-x))$.

`softmax(...)` : Softmax converts a vector of values to a probability distribution.

`softplus(...)` : Softplus activation function, $\text{softplus}(x) = \log(\exp(x) + 1)$.

`softsign(...)` : Softsign activation function, $\text{softsign}(x) = x / (\text{abs}(x) + 1)$.

`swish(...)` : Swish activation function, $\text{swish}(x) = x * \text{sigmoid}(x)$.

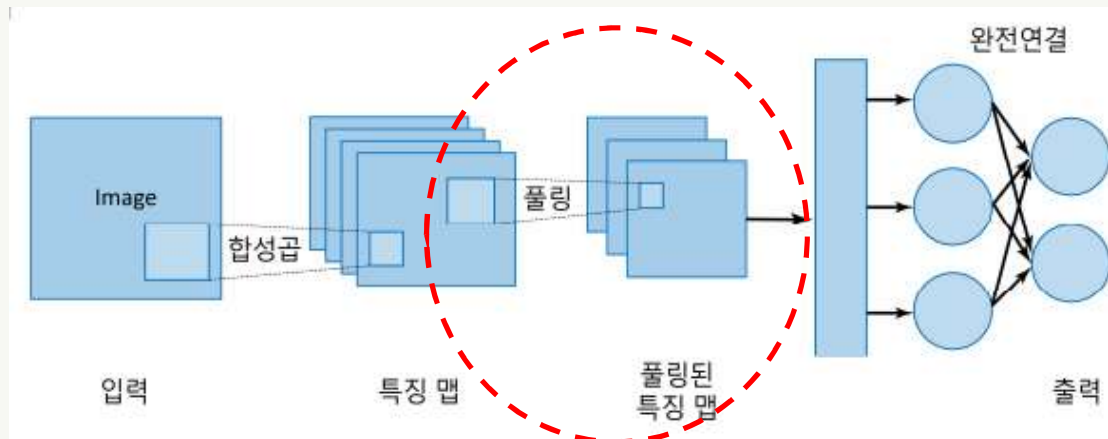
`tanh(...)` : Hyperbolic tangent activation function.



2. 합성곱 신경망의 주요 개념

● 풀링층

- 합성곱 층에서 연산된 출력 데이터를 입력으로 받아서 크기를 줄이는 역할
- 크기를 줄이기 때문에 다운샘플링(Down Sampling) 혹은 서브샘플링(Sub Sampling)이라고도 불림
- 합성곱 연산을 수행한 이후에 일반적으로 풀링층으로 구성



2. 합성곱 신경망의 주요 개념

● 최대 풀링과 평균 풀링

- 합성곱으로 연산된 모든 데이터의 특징을 활용하는 것이 아닌 대표적인 특징만 남기는 작업
- 일반적으로 풀링 기법 중 최대 풀링을 많이 사용함

입력				결과	
1	2	3	0	2	3
0	1	2	3		
3	0	1	2	3	2
2	3	0	1		

(가) 최대 풀링

입력				결과	
1	2	3	0	1	2
0	1	2	3		
3	0	1	2	2	1
2	3	0	1		

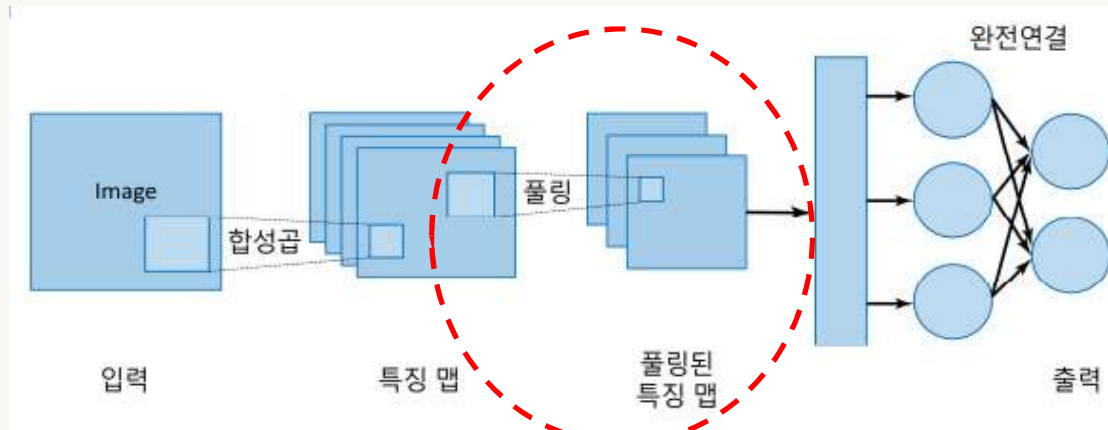
(나) 평균 풀링



2. 합성곱 신경망의 주요 개념

● 풀링의 특징

- 데이터의 크기가 작아지므로 연산량이 작아짐
- 풀링층은 합성곱층과 달리 학습을 위한 가중치가 없음
 - 풀링은 특정 영역에서 최대값이나 평균을 취하는 처리 과정
- 채널 수가 변하지 않음
 - 풀링 연산은 입력된 데이터의 채널 수대로 출력 데이터를 내보냄
- 입력된 데이터가 일부 변경되더라도 풀링의 결과는 크게 변하지 않음



2. 합성곱 신경망의 주요 개념

● 풀링 후 크기

- 4x4 데이터에 2x2 최대 풀링을 수행
- 스트라이드 2
- 풀링 연산의 결과는 4x4에서 2x2으로 축소

$$\text{특징 맵의 크기} = \frac{\text{입력된 데이터 크기} - \text{풀링 크기}}{\text{스트라이드}} + 1$$

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



2	3
4	2



2. 합성곱 신경망의 주요 개념

● 풀링층 코드

```
tf.keras.layers.MaxPooling2D(pool_size=(2, 2), strides=None)  
tf.keras.layers.AveragePooling2D(pool_size=(2, 2), strides=None)
```

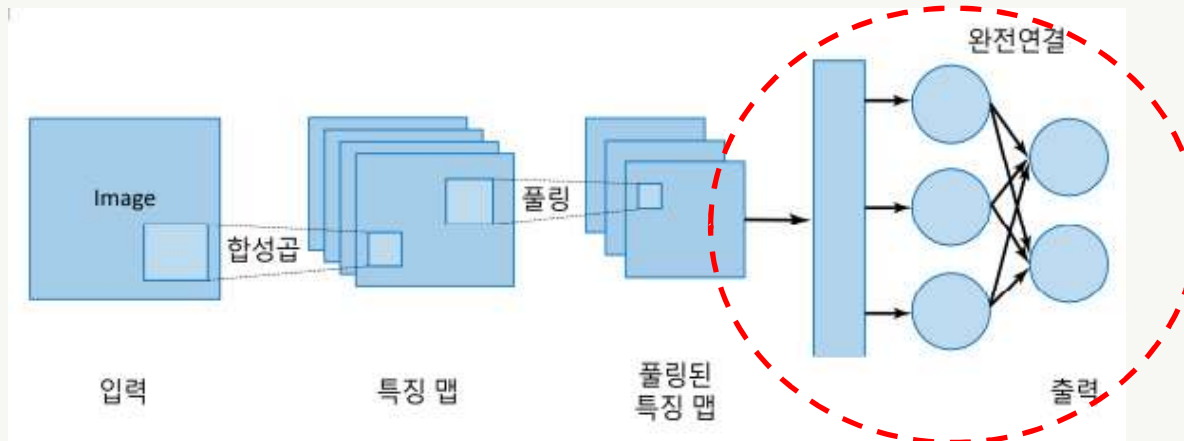
- `pool_size` : 풀링에 사용할 윈도우의 크기(기본 값 = (2, 2))
- `strides` : 풀링에 사용할 윈도우가 움직이는 간격의 크기(기본 값 = None)



2. 합성곱 신경망의 주요 개념

● 완전연결층

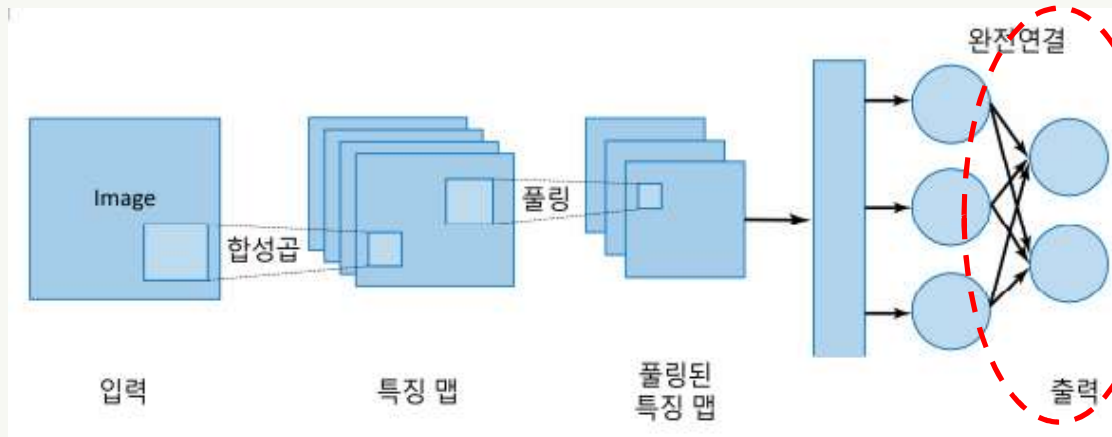
- 한 층과 그 다음 층이 모두 연결된 상태
- 완전연결층은 이전에 소개된 다층 퍼셉트론(MLP)을 지칭하는 또 다른 용어



2. 합성곱 신경망의 주요 개념

● 소프트맥스

- 출력층에서 사용되는 활성화함수로, 다중 클래스 분류 모델을 만들 때 사용
- 전체 클래스에 속할 확률을 0~1 사이의 값으로 정규화
- 출력값들의 총합은 항상 1



2. 합성곱 신경망의 주요 개념

● 완전연결층 코드

```
tf.keras.layers.Flatten(data_format=None)
tf.keras.layers.Dense(units, activation=None)
tf.keras.layers.Dense(units, activation=None)
```

- 플래튼층은 별도의 매개변수 없이 Flatten()으로 구현
- units : 출력 뉴런(노드)의 수
- activation : 사용할 활성화함수



정리하기

- 일반적으로 합성곱 신경망은 입력, 합성곱층, 풀링층, 완전연결층으로 구성된다.
- 입력은 데이터가 합성곱 신경망에서 처리될 수 있도록 변환되는 단계이다.
- 합성곱층은 합성곱 신경망에서 가장 많은 연산이 처리되는 계층으로, 이전 층에서 전달받은 입력 데이터와 필터를 통해 연산하게 된다.
- 풀링층은 합성곱층에서 연산된 출력 데이터를 입력으로 받아서 크기를 줄이는 역할을 하며, 크기를 줄이기 때문에 다운샘플링 혹은 서브샘플링이라고도 불리운다.
- 완전연결층은 한 층과 그 다음 층이 모두 연결된 상태로 1차원 배열의 형태로 변환하여 이미지를 분류하는데 사용되는 계층이다.



다음시간안내

07

합성곱 신경망(2)

