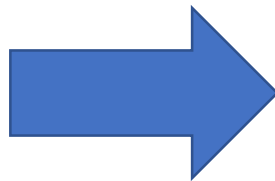


# 1. Cnn(Convolutional Neural Networks )

- Convolutional Neural Networks 합성곱 신경망이라고 함
- 인간의 시신경 구조를 모방한 기술
- 자율주행자동차, 얼굴인식과 같은 객체인식이나 computer vision이 필요한 분야에 많이 사용되고 있음
- 사람이 여러 데이터를 보고 기억한 후에 무엇인 지 맞추는 것과 유사함

# Cnn을 쓰게 된 이유

- Dnn(Deep Neural Network)의 문제점에서부터 출발
- 일반 Dnn은 1차원 형태의 데이터를 사용한다.
- 때문에 다른 차원의 이미지가 입력값이 되는 경우, 이것을 flatten시켜서 한줄 데이터로 만들어야 하는데 이과정에서 이미지의 공간적 /지역적 정보가 손실된다.
- 또한 추상화과정 없이 바론 연산과정으로 넘어가 버리기 때문에 학습시간과 능력의 효율성이 저하된다.



- 해결책
- Cnn은 이미지를 날 것 그대로 받음으로써 공간적/지역적 정보를 유지한 채 특성들의 계층을 빌드업 한다
- Cnn의 중요 포인트는 이미지 전체보다는 부분을 보는 것 , 그리고 이미지의 한 픽셀과 주변 픽셀들의 연관성을 살리는 것이다.

- 특징 추출 단계(Feature Extraction)

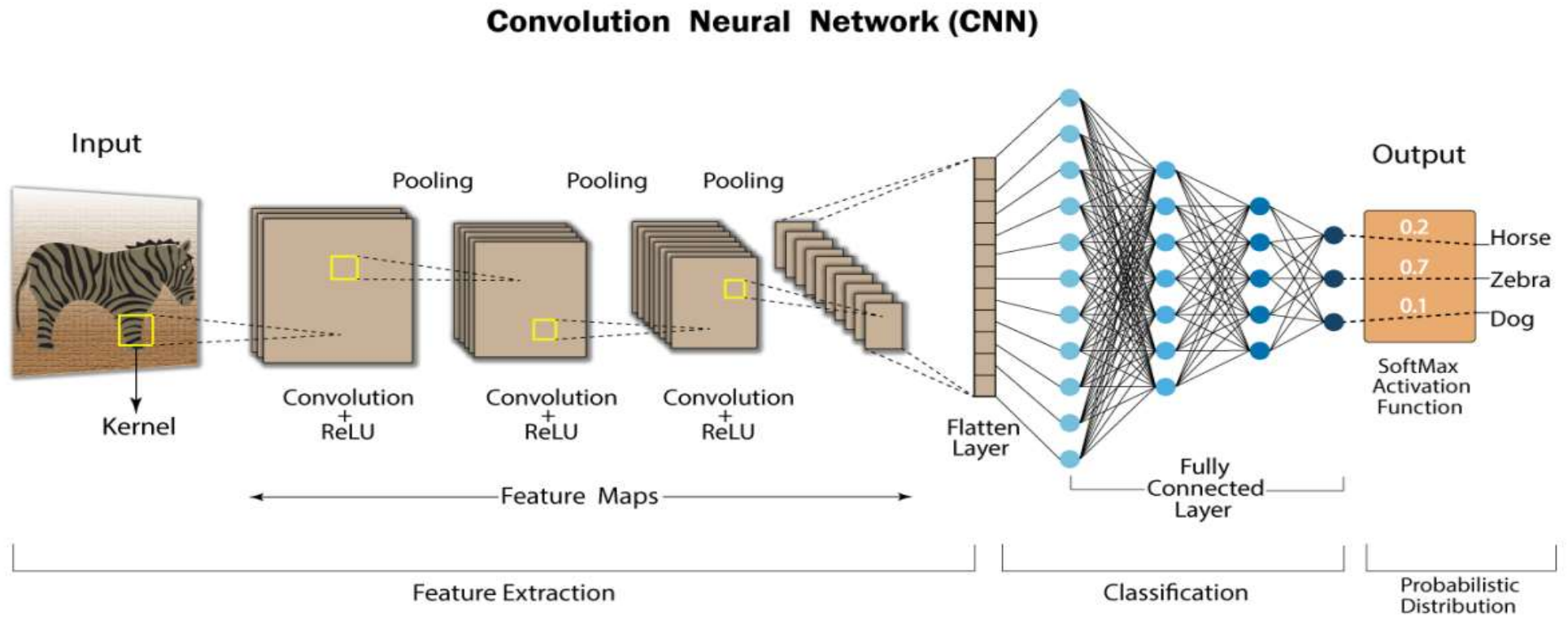
- Convolution Layer : 필터를 통해 이미지의 특징을 추출.
- Pooling Layer : 특징을 강화시키고 이미지의 크기를 줄임.
- (Convolution과 Pooling을 반복하면서 이미지의 feature를 추출)

- 이미지 분류 단계(Classification)

- Flatten Layer : 데이터 타입을 FC네트워크 형태로 변경. 입력데이터의 shape 변경만 수행.
- Softmax Layer : Classification수행.
- Output : 인식결과

- CNN의 파라미터

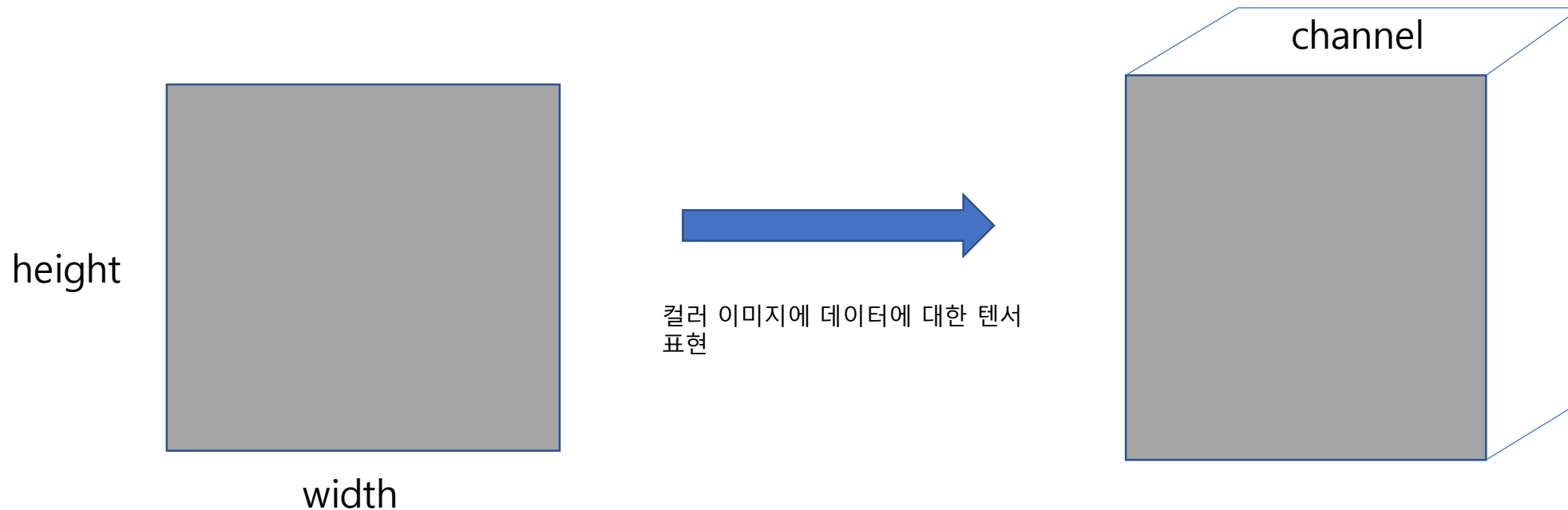
- Convolution Filter의 개수
- Filter의 사이즈
- Padding여부
- Stride



# Convolutoin(합성곱) Layer

이미지 데이터는 높이x너비x채널의 3차원 tensor로 표현될 수 있다.

만약 이미지의 색상이 RGB 코드로 표현되었다면 채널의 크기는 3이 되며 각각의 채널에는 R,G,B 값이 저장된다.



# Filter ,kernel

하나의 합성곱 계층에는 입력되는 이미지의 채널 개수만큼 필터가 존재하며, 각 채널에 할당된 필터를 적용함으로써 계층의 출력 이미지가 생성된다 이 과정에서 변환된

이미지들은 색상,선,형태,경계등의 특징이 뚜렷해진다. 필터를 통과한 이미지는 특성값을 가지고 있어 activation map이라고 한다.

0	1	7	5
5	5	6	6
5	3	3	0
1	1	1	2

 $\otimes$ 

1	0	1
1	2	0
3	0	1

 = 

40	

0	1	7	5
5	5	6	6
5	3	3	0
1	1	1	2

 $\otimes$ 

1	0	1
1	2	0
3	0	1

 = 

40	32

0	1	7	5
5	5	6	6
5	3	3	0
1	1	1	2

 $\otimes$ 

1	0	1
1	2	0
3	0	1

 = 

40	32
26	

0	1	7	5
5	5	6	6
5	3	3	0
1	1	1	2

 $\otimes$ 

1	0	1
1	2	0
3	0	1

 = 

40	32
26	25

하나의 채널에 대한 Convolution(합성곱) 계층의 동작

# padding

입력 이미지에 대해 합성곱을 수행하면 출력 이미지의 크기는 입력 이미지의 크기보다 작아지게 된다. 이러한 합성곱 계층을 거치면서 이미지의 크기는 점점 작아지게 되고, 이미지의 가장자리에 위치한 픽셀들의 정보가 사라진다.

패딩은 입력 이미지의 가장자리에 특정 값으로 설정된 픽셀들을 추가함으로써 입력 이미지와 출력 이미지의 크기를 같거나 비슷하게 만드는 역할을 수행한다.

0	0	0	0	0	0
0	0	1	7	5	0
0	5	5	6	6	0
0	5	3	3	0	0
0	1	1	1	2	0
0	0	0	0	0	0

 $\circledast$ 

1	0	0
1	2	1
1	2	3

 $=$ 

26	42	55	35
34	41	33	28
18	25	23	14
3	9	8	8

출력 이미지 크기와 입력 이미지 크기가 4x4로 유지되는 모습을 확인 할 수 있다

Zero-padding

# Pooling Layer

- 이미지의 크기를 유지한채 Fully Connected layer로 가게 된다면 연산량이 기하급수적으로 늘 것이기 때문에 적당히 크기도 줄이고, 특정 feature를 강조하는 역할을 한다.
- 처리 방법은 총 3가지로
  - Max Pooling
  - Average Pooling
  - Min Pooling
- CNN에서는 주로 Max Pooling을 사용한다.
- 이미지는 픽셀수준이 아니라, 특정 속성을 갖는 선택 영역 수준으로 표현될 수 있기에
- Max Pooling의 경우 선택 영역에서 가장 큰 값을 해당 영역의 대표 값으로 설정하는 것과 같다

# Pooling Layer

- Max Pooling을 사용하면 뉴런이 가장 큰 신호에 반응하는 것과 유사하고 이를 통해 노이즈가 감소하고 속도가 빨라지며 영상의 분별력이 좋아진다.

7	5	0	3
10	4	21	2
6	1	7	0
5	0	8	4

10	

7	5	0	3
10	4	21	2
6	1	7	0
5	0	8	4

10	21

7	5	0	3
10	4	21	2
6	1	7	0
5	0	8	4

10	21
6	

7	5	0	3
10	4	21	2
6	1	7	0
5	0	8	4

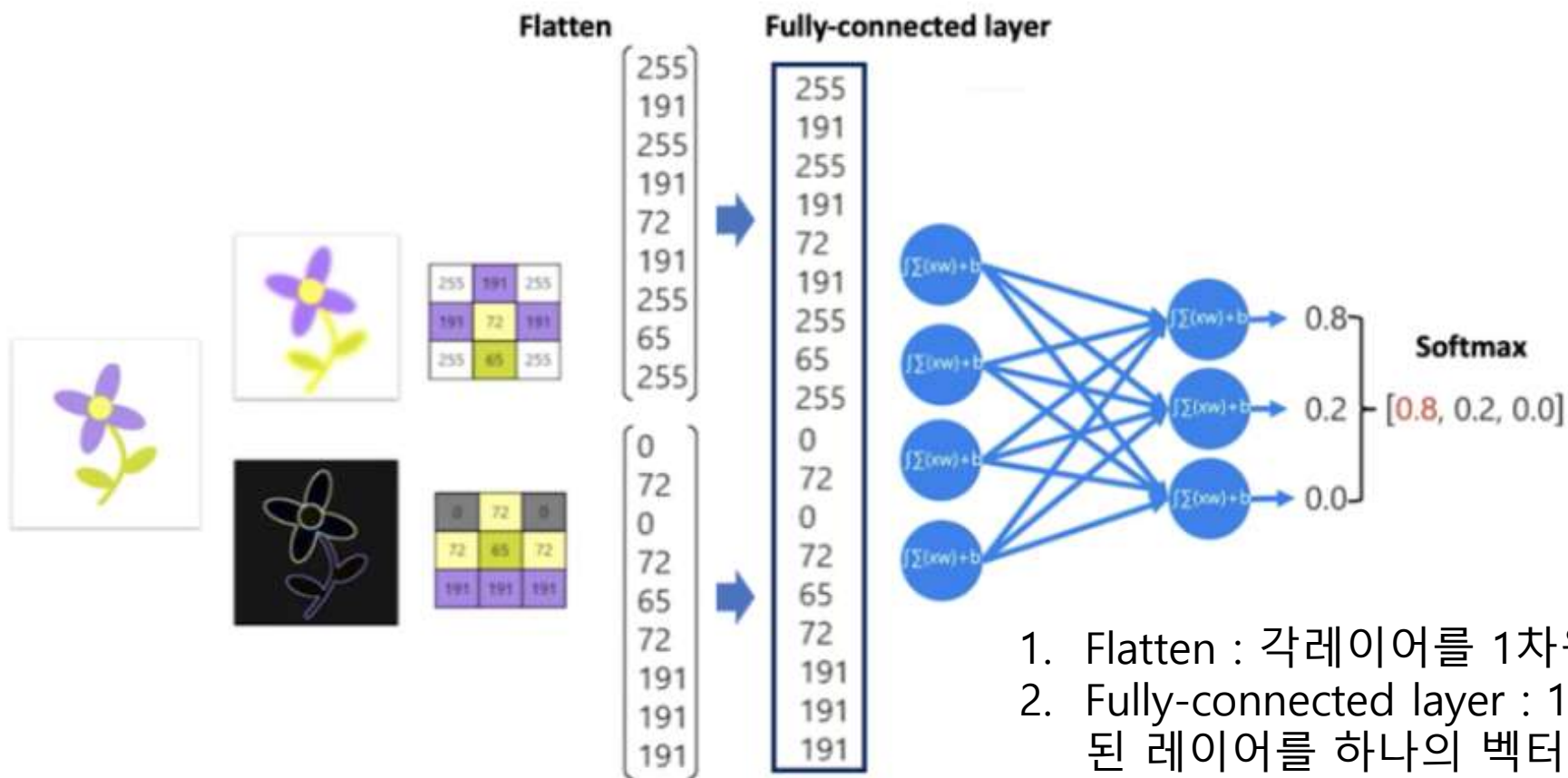
10	21
6	8

Stride가 2로 설정된 Max Pooling이 적용된 Pooling Layer



# Fully Connected Layer

이미지 특징을 추출하여 이것이 무엇을 의미하는 데이터 인지를 분류 작업 하는 것



1. Flatten : 각레이어를 1차원 벡터로 변환
2. Fully-connected layer : 1차원 벡터로 변환된 레이어를 하나의 벡터로 연결
3. Softmax 함수를 이용해 가장 확률이 높은 class를 output으로 분류한다.