CNN 데이터 분석 교육

13th lecture

"Convolution neural network"

2022 - 11 - 28

수학과 최명수

지난시간?

1. Backpropagation

오늘은 무엇을?

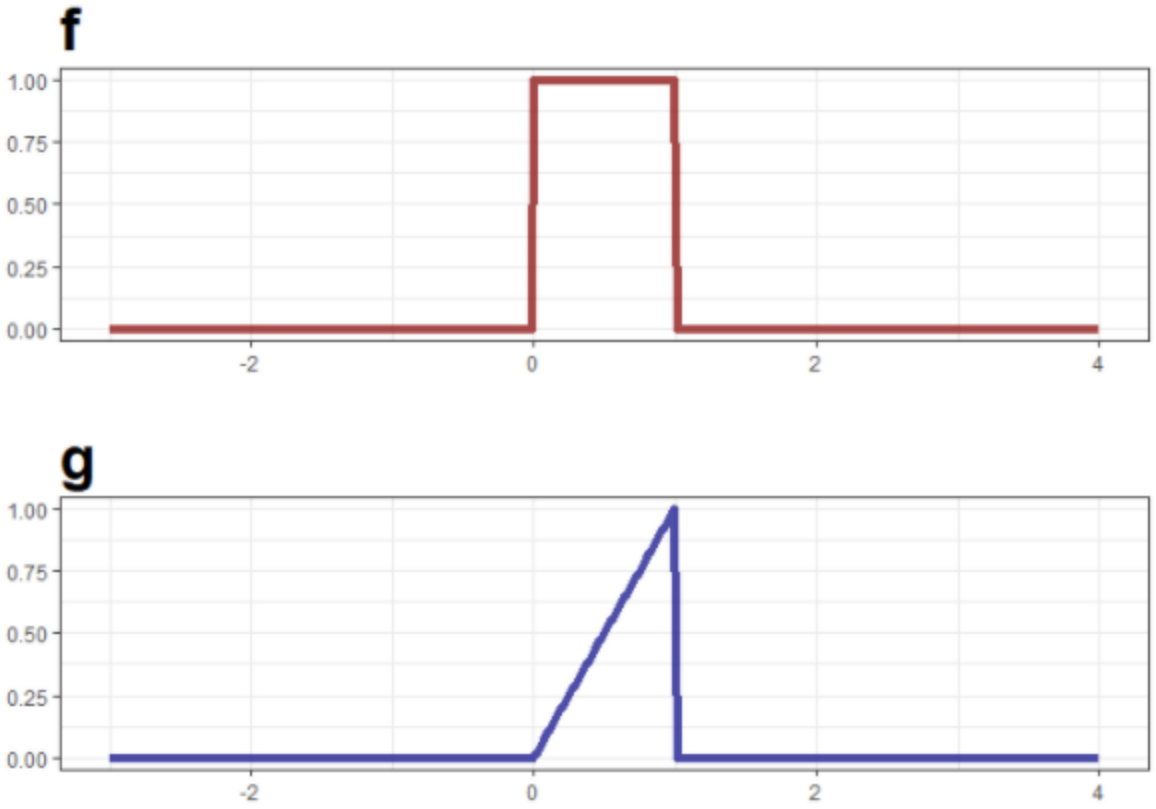
1. Convolution neural network란 무엇인가?

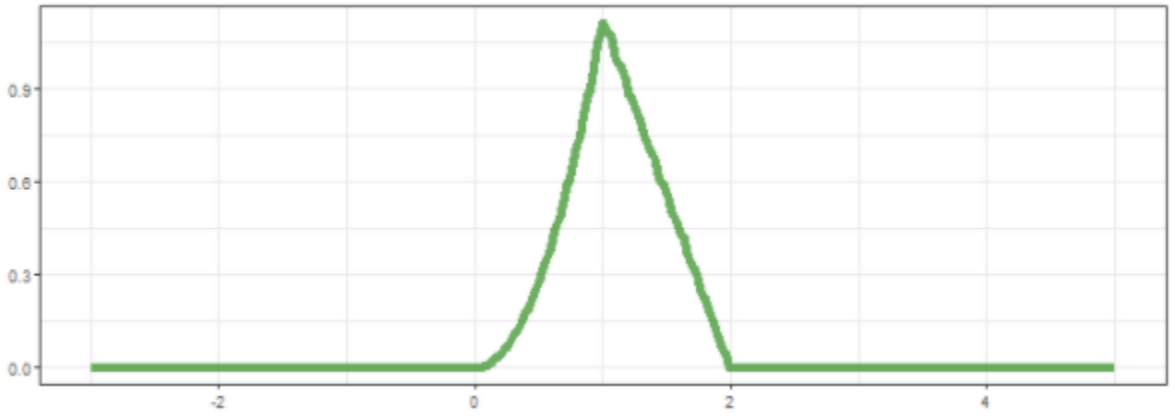
   -기존 우리가 다뤘던 MLP하고 다른 점?
   -어디에 주로 쓰이나?

Convolution

$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$

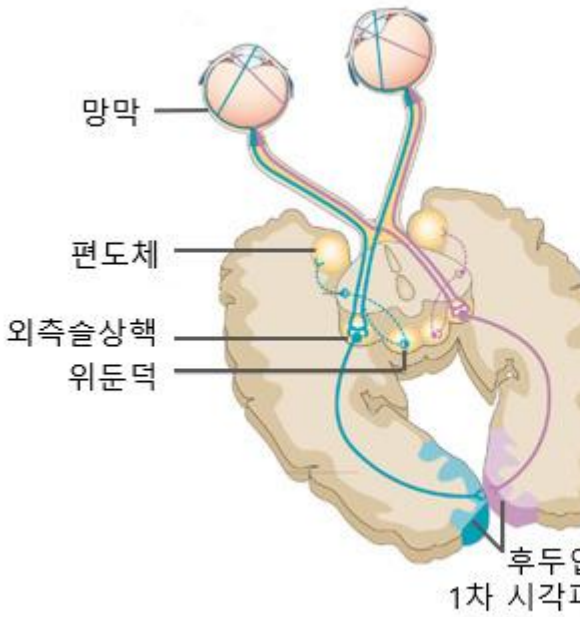
Convolution





https://yjjo.tistory.com/8

Convolution neural network



https://brunch.co.kr/@omnic

David H. Hubel과 Torsten Wiesel은 1958년과 1959년에 시각 피질의 구조에 대한 결정적인 통찰을 제공한 실험을 수행했다. 이들은 시각 피질 안의 많은 뉴런이 작은 국부 수용영역(Local Receptive Field)을 보았고, 이것은 뉴런들이 시야의 일부 범위 안에 있는 시각 자극에만 반응한다는 뜻이다[14]. 뉴런의 수용영역(Receptive Field)들은 서로 겹칠 수 있으며, 이렇게 겹쳐진 수용영역들이 전체 시야를 다루게 된다. 어떤 뉴런은 수직선에만 반응하고, 다른 뉴런은 다른 각도에서 반응하는 뉴런이 있을 뿐만 아니라, 또 다른 뉴런은 큰 수용영역을 가져 저수준의 패턴이 조합되어 복잡한 패턴에 반응하다는 것을 알게 된다[15]. 이러한 관찰을 통해 고수준의 뉴런이 이웃한 저수준 출력에 기반 한다는 아이디어를 생각해 냈으며 이러한 아이디어가 점차 합성곱 신경망으로 진화되어 왔다[16,17].

Convolution Neural Network Based Toxic Plants Discrimination system (신현석, 2020)

Convolution neural network

일반 Dense network

1	0	0	0	1
0	1	0	1	0
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1

1	0	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	1

1	1	0	0	1
0	1	0	1	0
0	0	1	0	0
0	1	0	1	0
1	0	0	1	1

Flattening

1	1	0	0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	1	1

0	0	0	0	0
0	0	0	0	0
0	0	1	0	1
0	0	0	1	0
0	0	1	0	1

0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	1	0	1

Convolution neural network

1	0	0	0	1
0	1	0	1	0
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1

1	1	0	0	1
0	1	0	1	0
0	0	1	0	0
0	1	0	1	0
1	0	0	1	1

0	0	0	0	0
0	0	0	0	0
0	0	1	0	1
0	0	0	1	0
0	0	1	0	1

Convolution neural network

1. 일반적인 DNN은 전체 데이터를 1차원 데이터로 변환한 데이터를 입력 받음
   -여기서 공간적/지역적 정보(spatial/topological information)가 손실

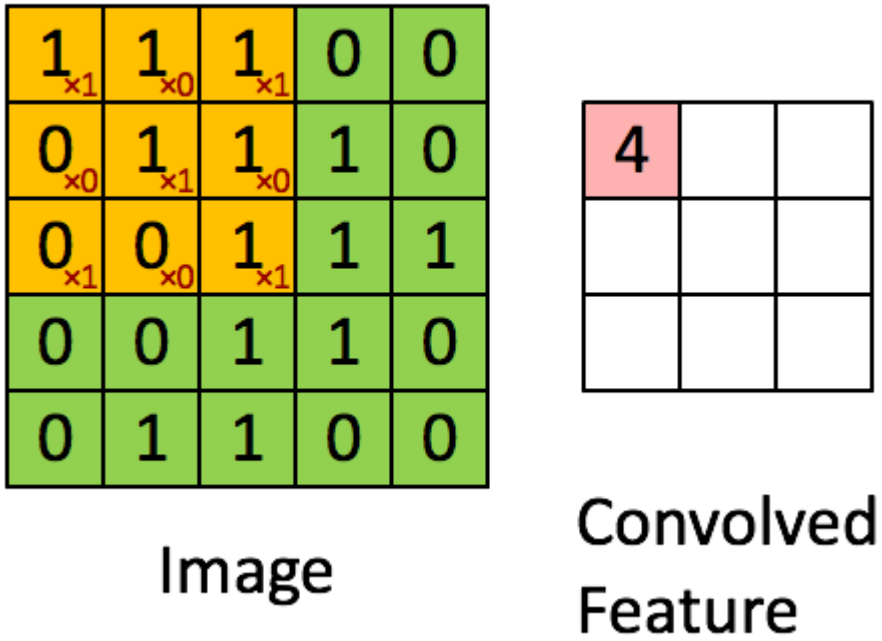
2. 이를 해결하기 위해 CNN 방법이 고안됨!
   - 이미지를 그대로 받음으로써 공간적/지역적 정보를 유지

   - 이 과정에서 이미지의 부분적인 특성(feature)들의 정보가 유지됨

   - 즉 이미지 전체보다는 부분을 보며, 한 픽셀과 주변 픽셀간의 연관성을 살리는 것이 CNN의 중요 포인트

How?

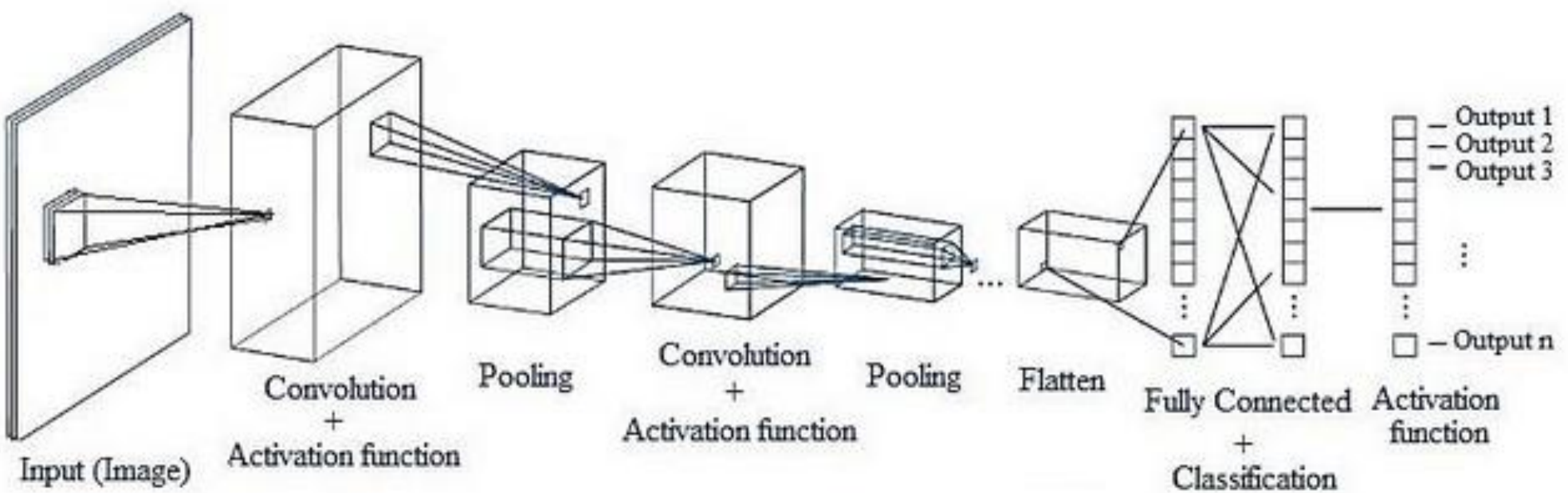
$$(f * g)(t) \stackrel{\text{def}}{=} \int_{-\infty}^{\infty} f(\tau)\, g(t-\tau)\, d\tau$$



필터를 이용하여 이미지의 특징을 추출!

http://taewan.kim/post/cnn/

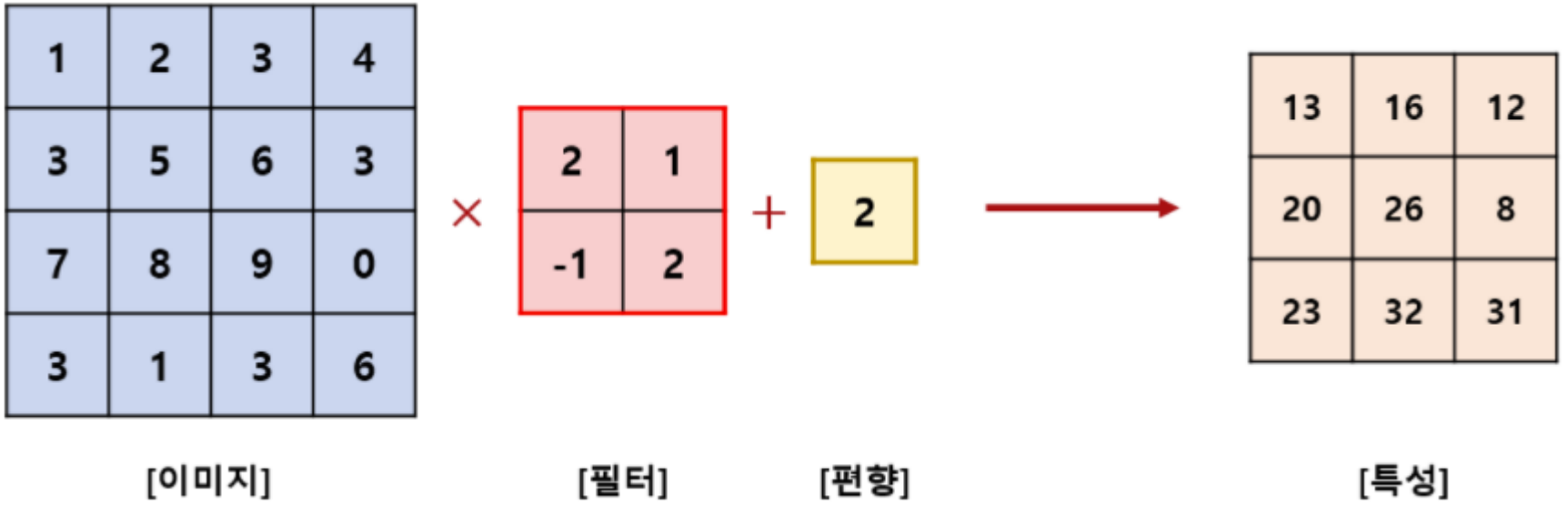
How?



https://halfundecided.medium.com/%EB%94%A5%EB%9F%AC%EB%8B%9D-%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-cnn-convolutional-neural-networks-%EC%89%BD%EA%B2%8C-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0-836869f88375

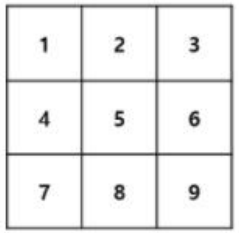
How?

1. 합성곱 연산(Convolution operation)



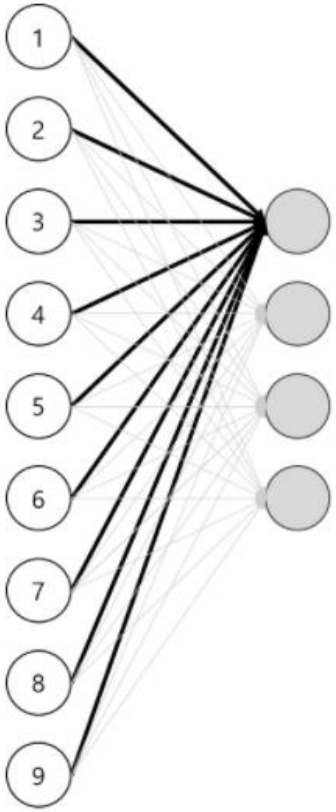
4×4 → 3×3

How?

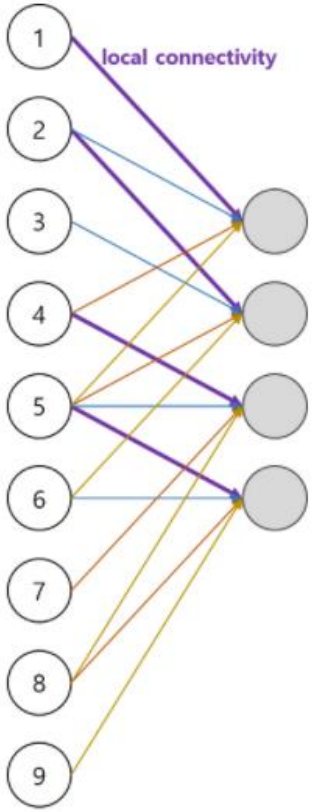


[이미지]

[필터]



[Fully Connected Layer]



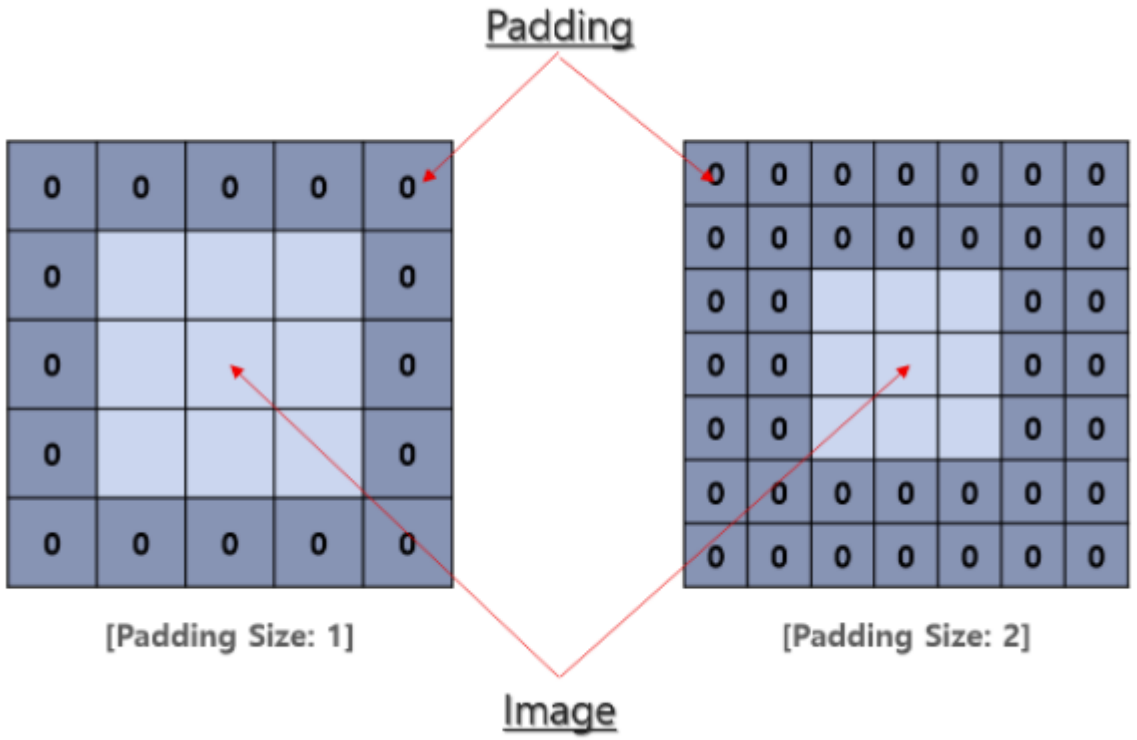
[Convolutional Layer]

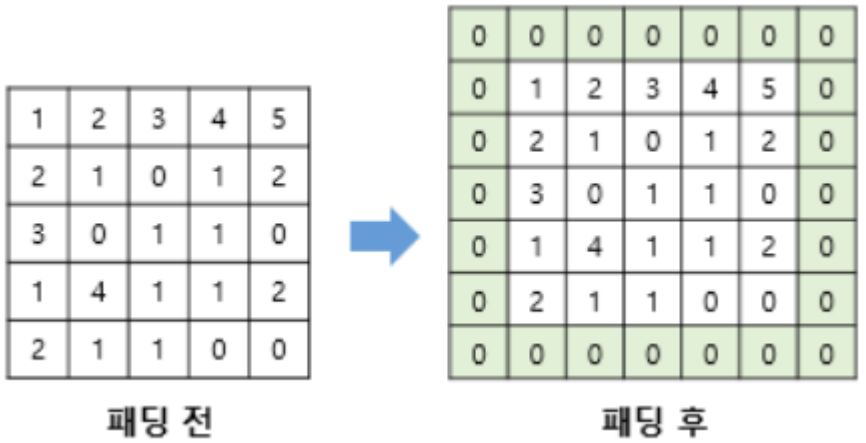
How?

How?

2. 패딩(Padding)

얼만큼씩 훑고 지나갈거냐(Stride size)에 따라 Padding size도 맞춰서 조절

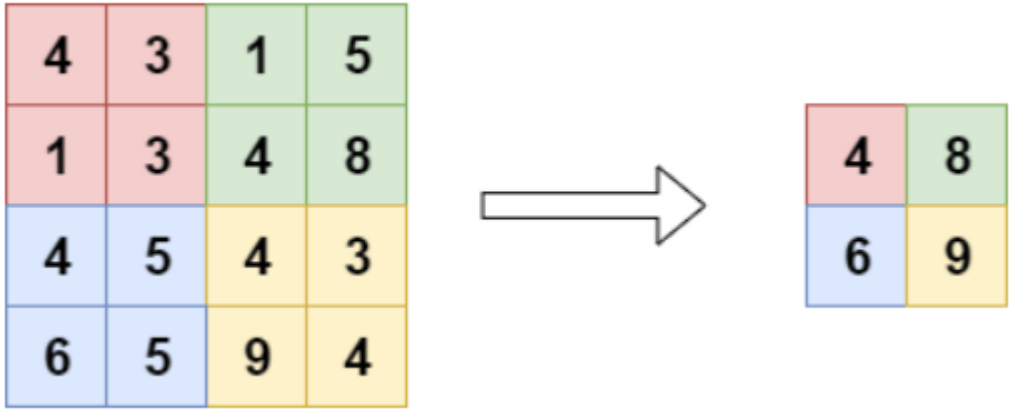




How?

3. 풀링(Padding)

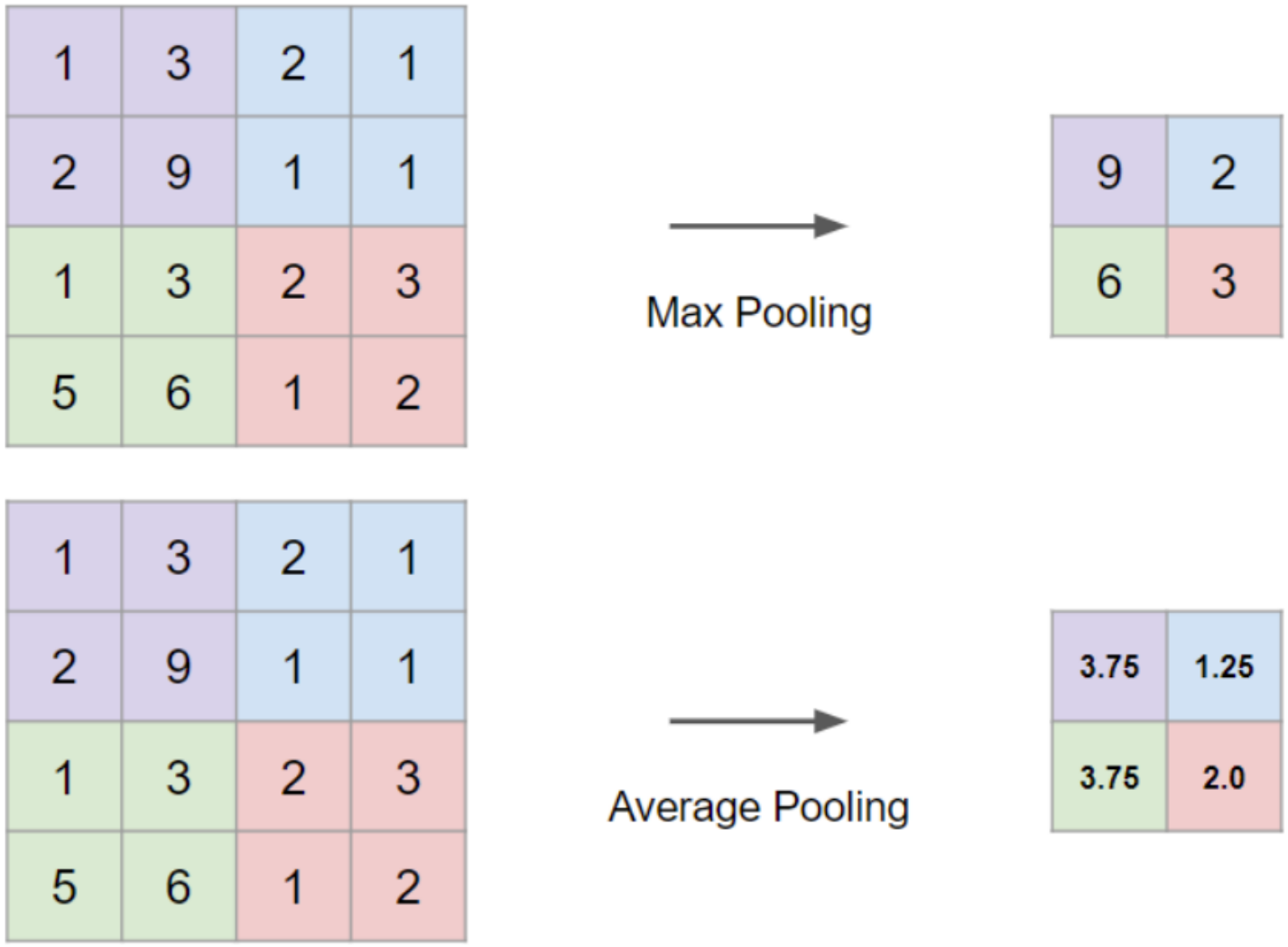
데이터의 공간적인 특성을 유지하면서 크기를 줄여줌



https://www.machinecurve.com/index.php/2020/01/30/what-are-max-pooling-average-pooling-global-max-pooling-and-global-average-pooling/

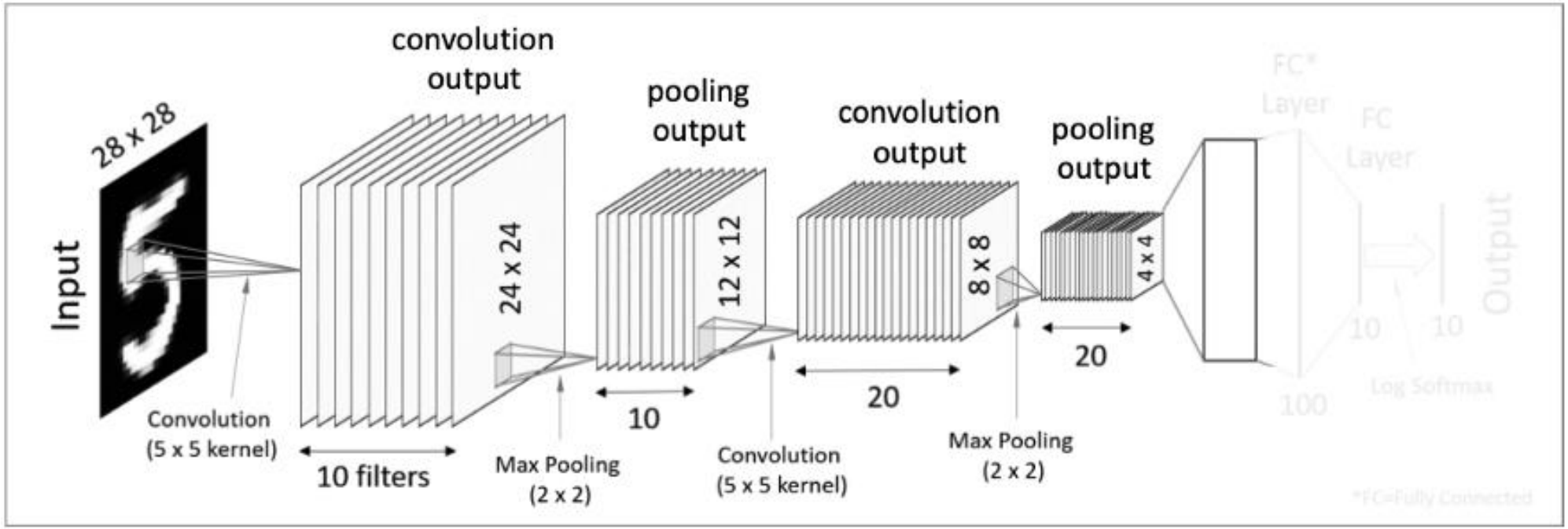
How?

3. 풀링(Padding)



https://www.edwith.org/ai218/lecture/34922

How?

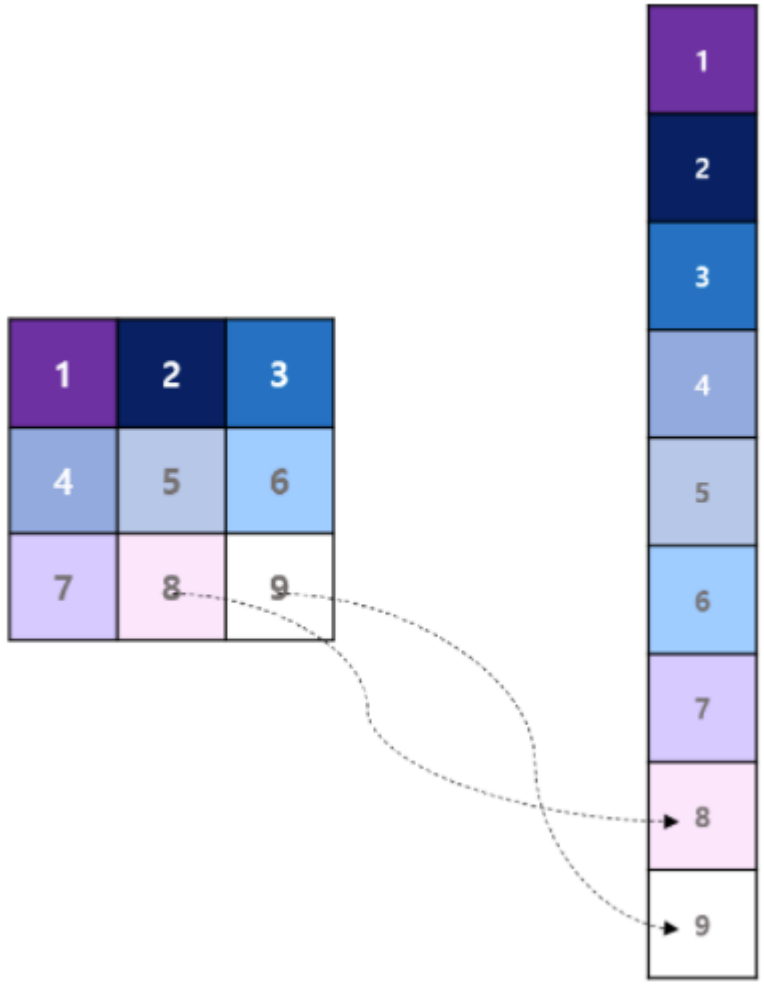


https://halfundecided.medium.com/%EB%94%A5%EB%9F%AC%EB%8B%9D-%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-cnn-convolutional-neural-networks-%EC%89%BD%EA%B2%8C-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0-836869f88375

How?

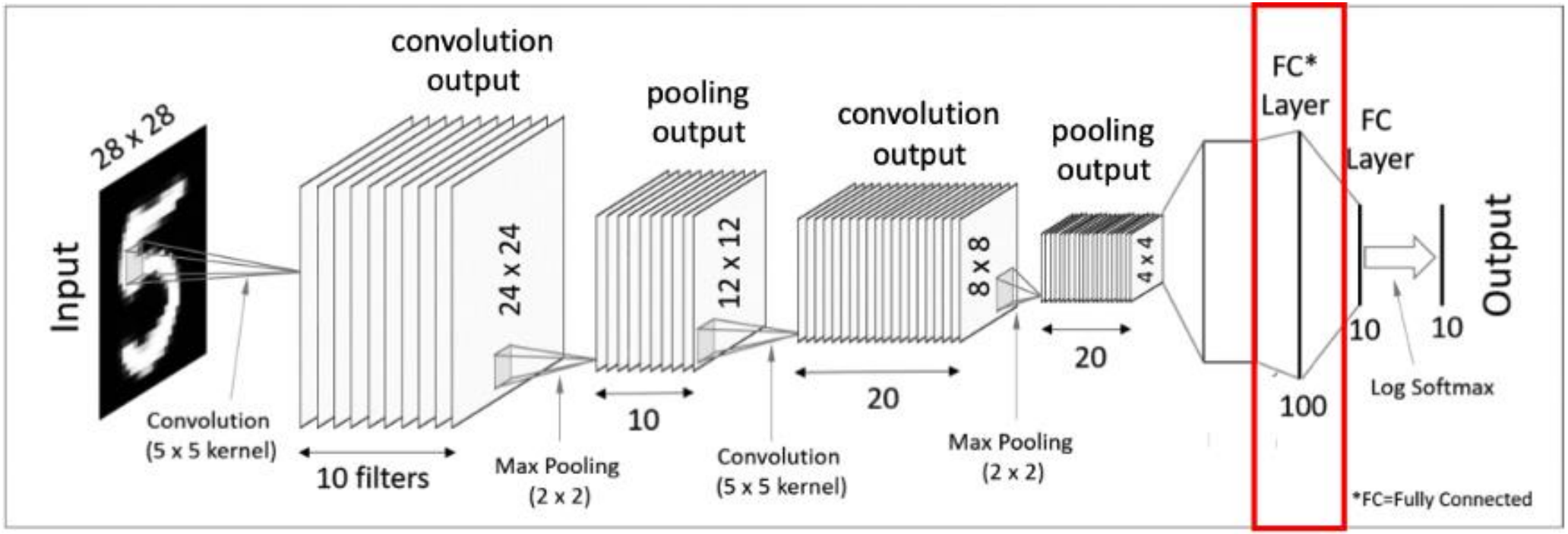
4. 플래트닝(Flattening)

CNN의 마지막 단계



https://yjjo.tistory.com/8

How?



https://halfundecided.medium.com/%EB%94%A5%EB%9F%AC%EB%8B%9D-%EB%A8%B8%EC%8B%A0%EB%9F%AC%EB%8B%9D-cnn-convolutional-neural-networks-%EC%89%BD%EA%B2%8C-%EC%9D%B4%ED%95%B4%ED%95%98%EA%B8%B0-836869f88375

Convolution neural network

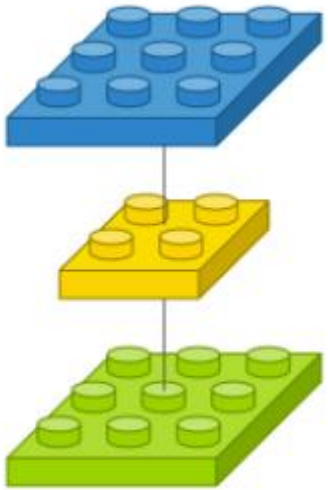
```
model.add(layers.Conv2D(32, (3, 3), padding='valid', activation='relu', input_shape=(28, 28, 1)))
```

필터 갯수

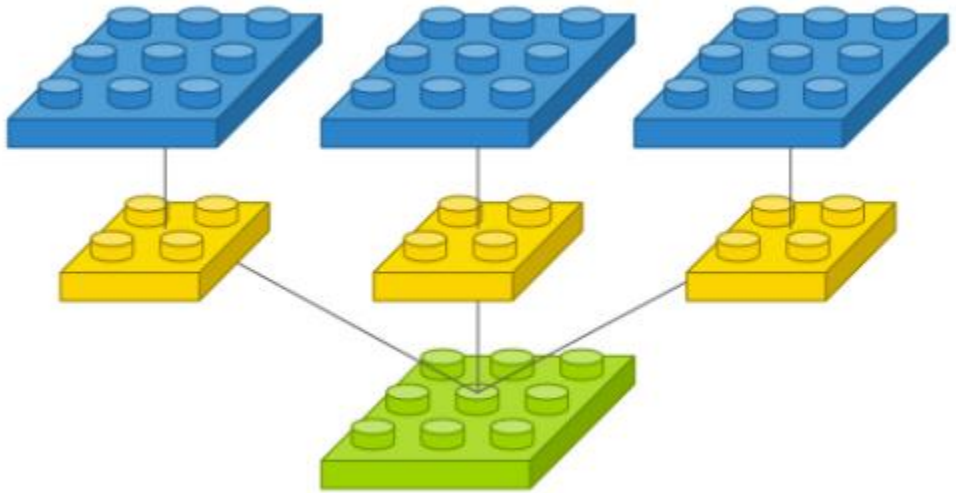
필터 크기

Input 이미지가 28x28 흑백

Con2v(1, (2,2), input_shape = (3,3,1)



Con2v(3, (2,2), input_shape = (3,3,1)

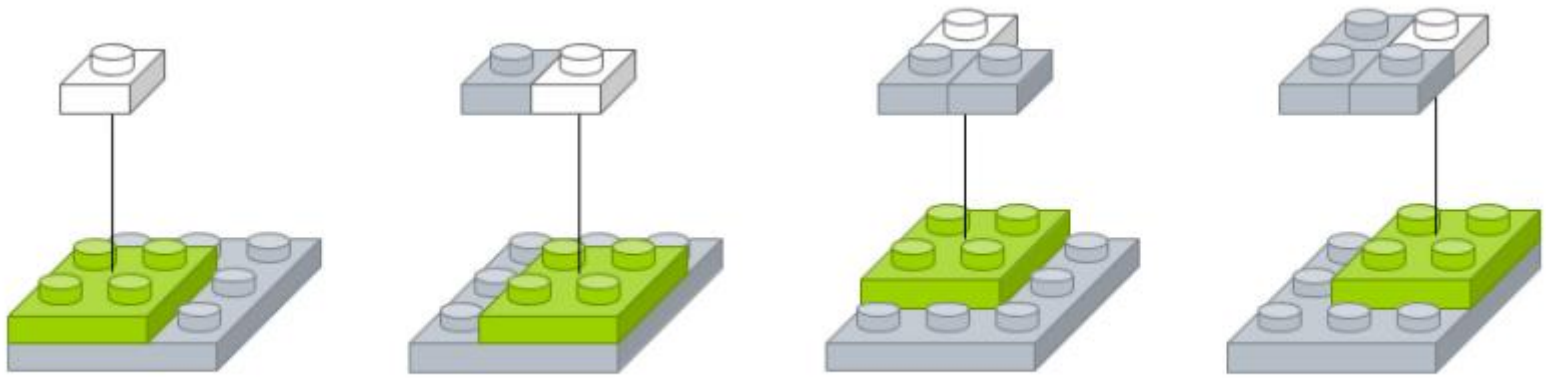


https://tykimos.github.io/2017/01/27/CNN_Layer_Talk/

Convolution neural network

```
model.add(layers.Conv2D(32, (3, 3), padding='valid', activation='relu', input_shape=(28, 28, 1)))
```

Padding = ‘valid’ : output 크기가 input 크기보다 작음

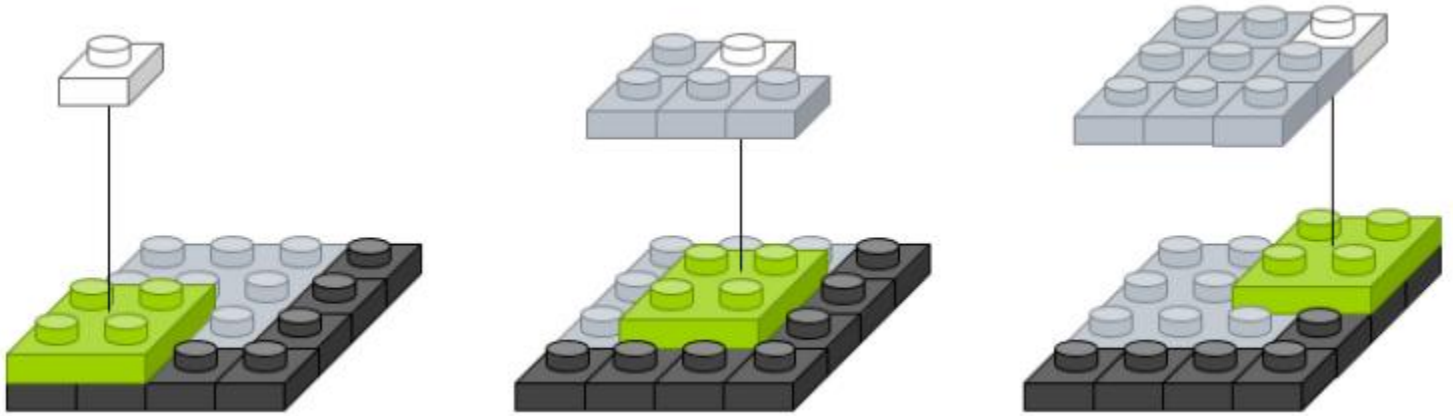


https://tykimos.github.io/2017/01/27/CNN_Layer_Talk/

Convolution neural network

```
model.add(layers.Conv2D(32, (3, 3), padding='same', activation='relu', input_shape=(28, 28, 1)))
```

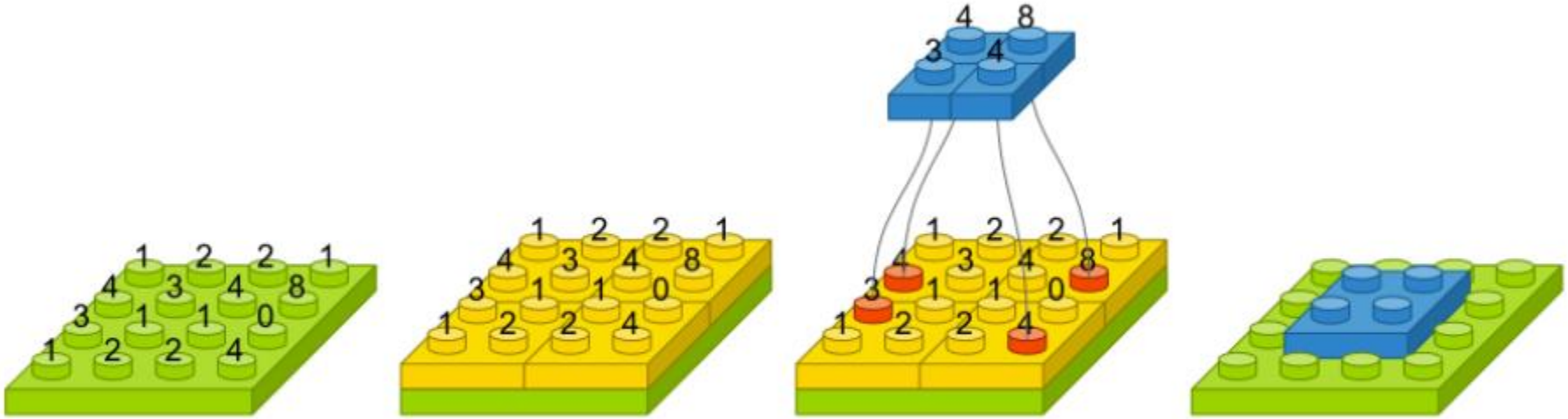
Padding = ‘same’ : output 크기와 input 크기가 같음
- 경계면 학습에 좀 더 효율적



https://tykimos.github.io/2017/01/27/CNN_Layer_Talk/

Convolution neural network

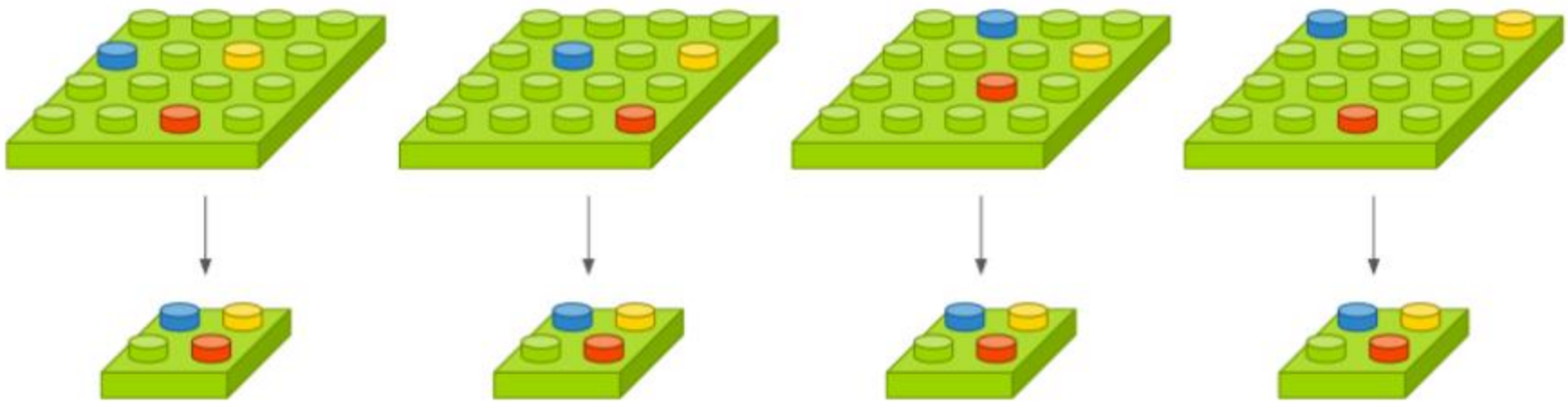
```
model.add(layers.MaxPooling2D((2, 2)))
```



https://tykimos.github.io/2017/01/27/CNN_Layer_Talk/

Convolution neural network

```
model.add(layers.MaxPooling2D((2, 2)))
```

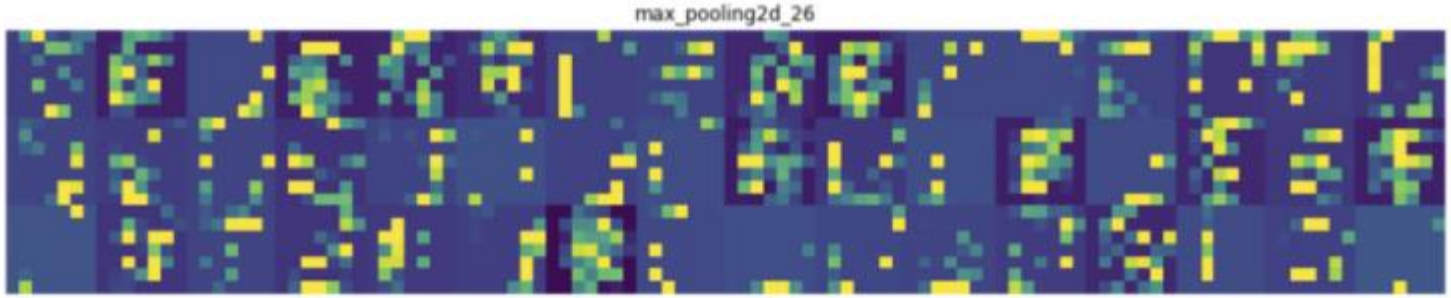


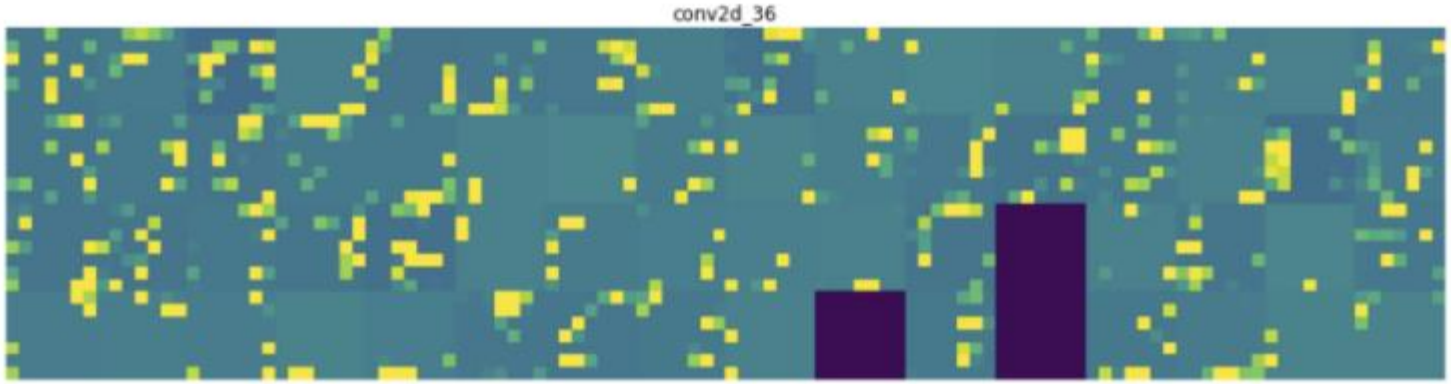
https://tykimos.github.io/2017/01/27/CNN_Layer_Talk/

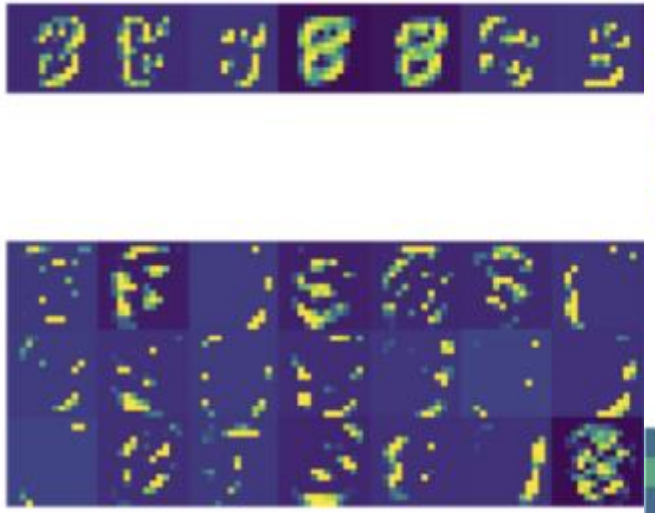
Convolution neural network

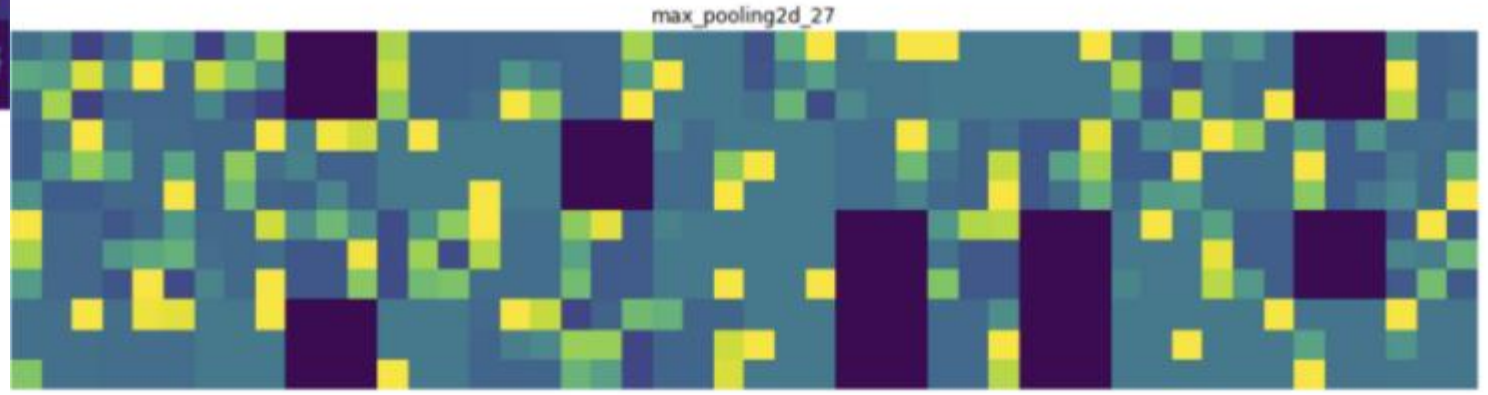
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 64)	0
conv2d_2 (Conv2D)	(None, 3, 3, 64)	36928
flatten (Flatten)	(None, 576)	0
dense (Dense)	(None, 64)	36928
dense_1 (Dense)	(None, 10)	650

Convolution neural n









Convolution neural network

DNN - Mnist

```
Epoch 1/5
1500/1500 [==============================] - 5s 3ms/step - loss: 0.7188 - accuracy: 0.8165
Epoch 2/5
1500/1500 [==============================] - 4s 3ms/step - loss: 0.3620 - accuracy: 0.9002
Epoch 3/5
1500/1500 [==============================] - 4s 3ms/step - loss: 0.3088 - accuracy: 0.9133
Epoch 4/5
1500/1500 [==============================] - 4s 3ms/step - loss: 0.2786 - accuracy: 0.9217
Epoch 5/5
1500/1500 [==============================] - 4s 3ms/step - loss: 0.2558 - accuracy: 0.9286
<keras.callbacks.History at 0x7fbfd5f8e750>
```

CNN - Mnist

```
Epoch 1/5
1875/1875 [==============================] - 27s 8ms/step - loss: 0.1502 - accuracy: 0.9540
Epoch 2/5
1875/1875 [==============================] - 14s 8ms/step - loss: 0.0476 - accuracy: 0.9848
Epoch 3/5
1875/1875 [==============================] - 14s 8ms/step - loss: 0.0331 - accuracy: 0.9893
Epoch 4/5
1875/1875 [==============================] - 14s 8ms/step - loss: 0.0254 - accuracy: 0.9919
Epoch 5/5
1875/1875 [==============================] - 14s 8ms/step - loss: 0.0205 - accuracy: 0.9938
```

공지

1. 다음 마지막 시간!
   -Neural Network을 이용하여 댓글로부터 영화 평점을 예측할 수 있는 모델 제작

끝

담에봅시당