

6. CNN(1)

AILab
Hanyang Univ.

오늘 실습내용

- **Initialize weights**
- **Check Point**
- **CNN basic**

Initialize weights

- randomly : generates tensors with a normal distribution.
 - 함수

```
# Random normal  
w1 = tf.Variable(tf.random_normal(shape=[784, 256]))
```

- Uniform distribution : generates tensors with a uniform distribution.
 - 함수

```
# Random uniform  
w2 = tf.Variable(tf.random_uniform(shape=[784, 256]))
```

- Xavier : X.Glorot and Y.Bengio, 2010
 - $w = \text{np.random.randn}(\text{fan_in}, \text{fan_out}) / \text{np.sqrt}(\text{fan_in})$ # fan_in : 입력값, fan_out : 출력값
 - 함수

```
# Xavier  
w3 = tf.get_variable("w3", shape=[784, 256], initializer=tf.contrib.layers.xavier_initializer())
```

Check Point

```
1 import tensorflow as tf
2
3
4 #data
5
6 #model
7
8 #loss function and optimizer
9
10 ckpt_path =
11 with tf.Session as sess:
12     saver = tf.train.Saver()
13     saver.restore(sess, ckpt_path)
14
15
16     #training
17
18     saver.save(sess, ckpt_path)
```

Check Point

```
1 import tensorflow as tf
2
3
4 #data
5
6 #model
7
8 #loss function and optimizer
9
10 ckpt_path =
11 with tf.Session as sess:
12     saver = tf.train.Saver()
13     saver.restore(sess, ckpt_path)
14
```

- checkpoint
- scene.ckpt.data-00000-of-00001
- scene.ckpt.index
- scene.ckpt.meta

2018-08-20 오후 7:45	파일	1KB
2018-08-20 오후 7:45	DATA-00000-OF-...	1,361,498...
2018-08-20 오후 7:45	INDEX 파일	3KB
2018-08-20 오후 7:46	META 파일	242KB

```
18     saver.save(sess, ckpt_path)
```

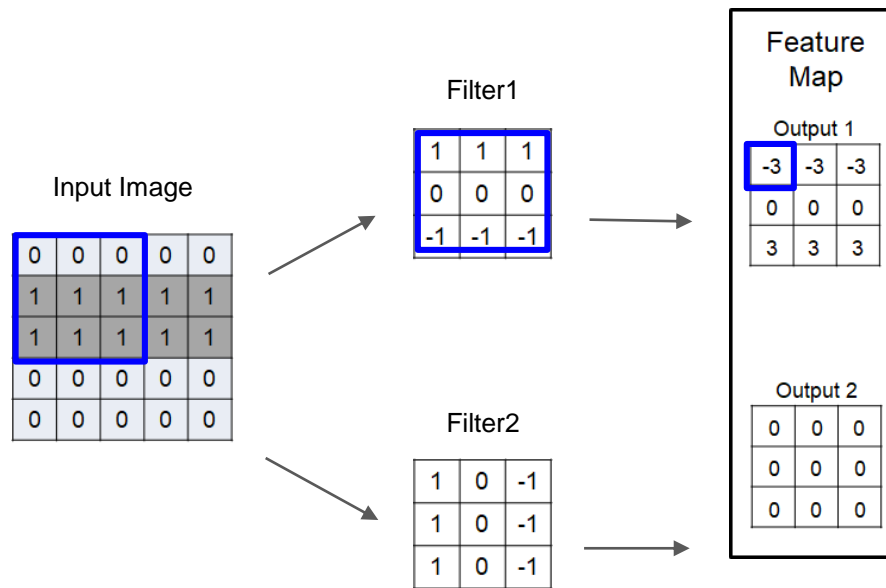
CNN

```
W = tf.get_variable(name=name + "_W", shape=[filter_size, filter_size, fin, fout],
                    initializer=tf.contrib.layers.xavier_initializer())
b = tf.get_variable(name=name + "_b", shape=[fout], initializer=tf.contrib.layers.xavier_initializer(0.0))
C = tf.nn.conv2d(din, W, strides=[1, 1, 1, 1], padding='SAME')
R = tf.nn.relu(tf.nn.bias_add(C, b))
```

CNN

Convolution?

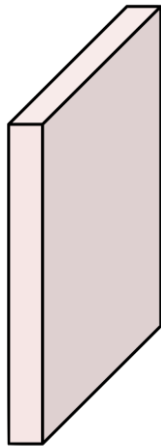
: Convolution(합성곱)은 하나의 함수와 또 다른 함수를 반전 이동한 값을 곱한 다음, 구간에 대해 적분하여 새로운 함수를 구하는 수학 연산자이다.



CNN

Channel

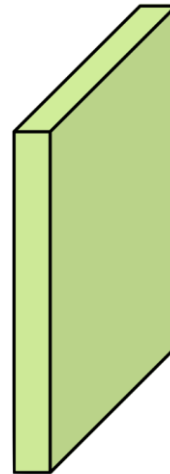
- 입력 데이터의 Channel 수와 필터의 Channel 수가 일치 해야 함
- 필터의 개수가 아웃 쪽의 Channel을 결정



Input image
(3, 3, 1)
(width, height, channel)



filter
(2, 2, 1)
(width, height, channel)

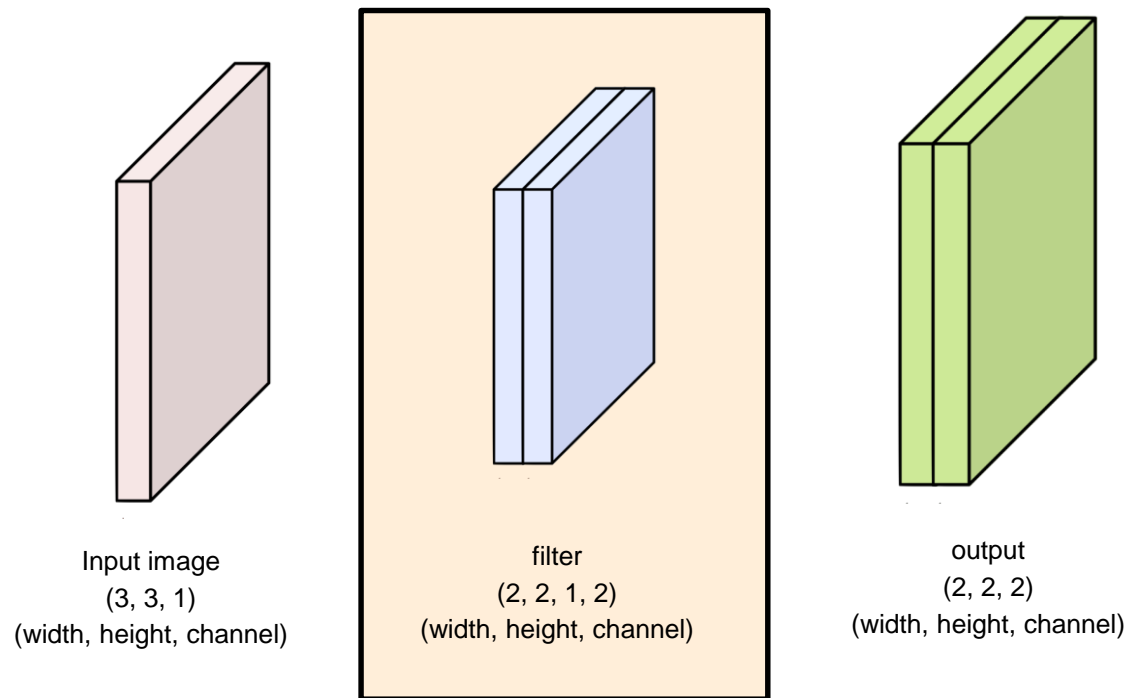


output
(2, 2, 1)
(width, height, channel)

CNN

Channel

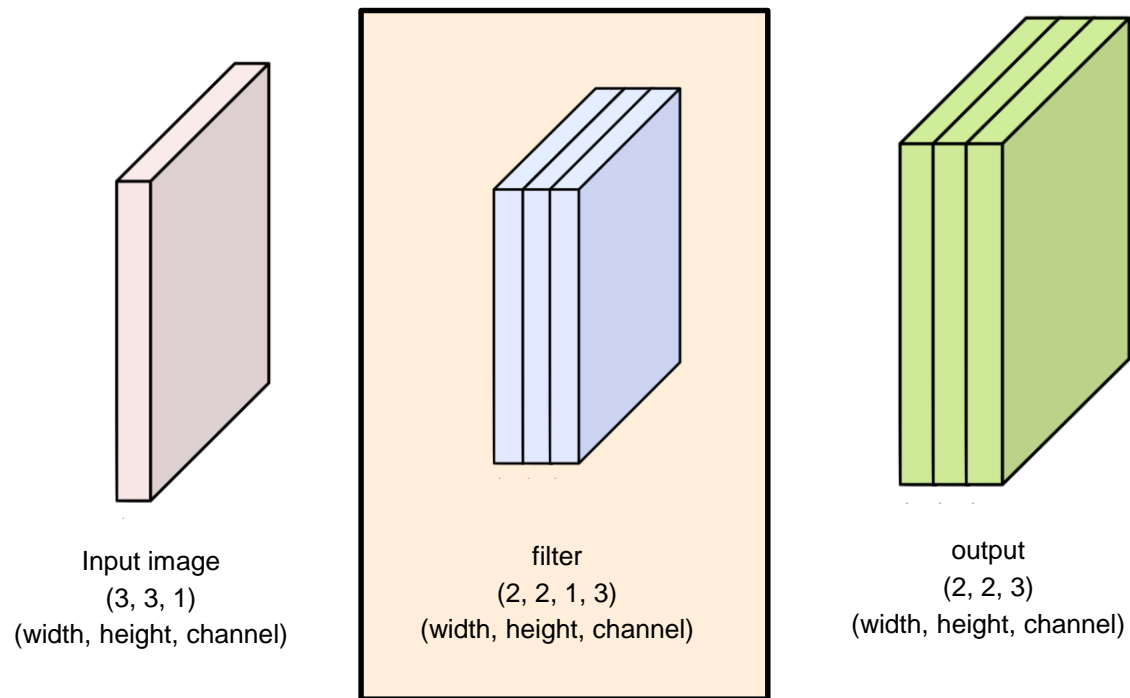
- 입력 데이터의 Channel 수와 필터의 Channel 수가 일치 해야 함
- 필터의 개수가 아웃 쪽의 Channel을 결정



CNN

Channel

- 입력 데이터의 Channel 수와 필터의 Channel 수가 일치 해야 함
- 필터의 개수가 아웃 쪽의 Channel을 결정



CNN

Stride

: 필터를 적용하는 위치의 간격

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

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

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

Stride = 1

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

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

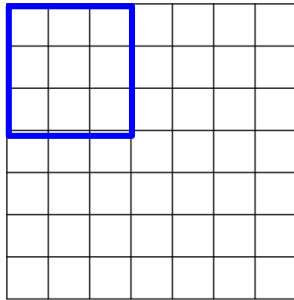
Stride = 2

```
conv2d = tf.nn.conv2d(image, weight, strides=[1, 1, 1, 1], padding='VALID')
```

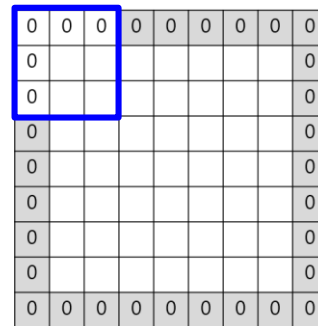
CNN

Padding

: Convolution으로 인한 image 모서리 부분 정보 손실 방지를 위해 입력데이터 주변을 특정값으로 채우는 것 (일반적으로 Zero Padding사용)



zero padding 사용X
 $7*7 \rightarrow 5*5$



zero padding 사용
 $7*7 \rightarrow 7*7$

```
conv2d = tf.nn.conv2d(image, weight, strides=[1, 1, 1, 1], padding='VALID')
```

valid
same

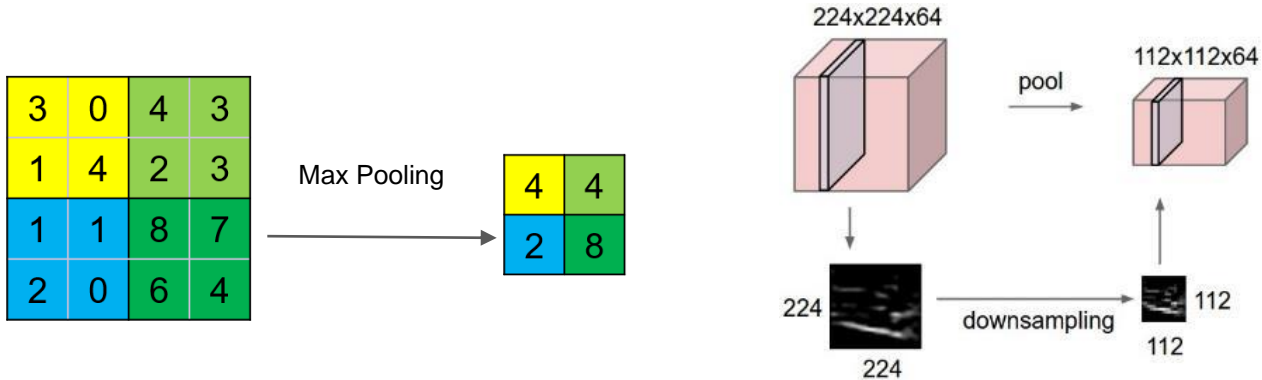
...

CNN

Pooling?

: 가로 세로 방향의 공간을 줄이는 연산

- 출력의 해상도를 낮춰 변형이나 이동에 대한 민감도를 감소
- 이미지의 크기를 줄이기 때문에 학습할 노드의 수가 줄어들어 학습속도를 높이는 효과
- 하지만, 정보의 손실이 일어남
- CNN에서는 일반적으로 Max Pooling 사용



```
pool = tf.nn.max_pool(image, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding="SAME")
```

CNN

```
W = tf.get_variable(name=name + "_W", shape=[filter_size, filter_size, fin, fout],  
                    initializer=tf.contrib.layers.xavier_initializer())  
b = tf.get_variable(name=name + "_b", shape=[fout], initializer=tf.contrib.layers.xavier_initializer(0.0))  
C = tf.nn.conv2d(din, W, strides=[1, 1, 1, 1], padding='SAME')  
R = tf.nn.relu(tf.nn.bias_add(C, b))  
pool = tf.nn.max_pool(R, ksize=[1, 2, 2, 1], strides=[1, 2, 2, 1], padding="SAME")
```