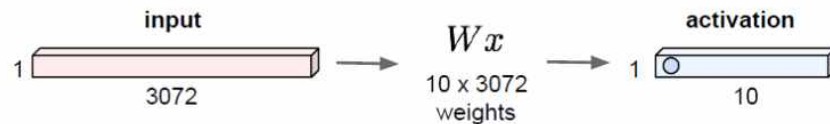


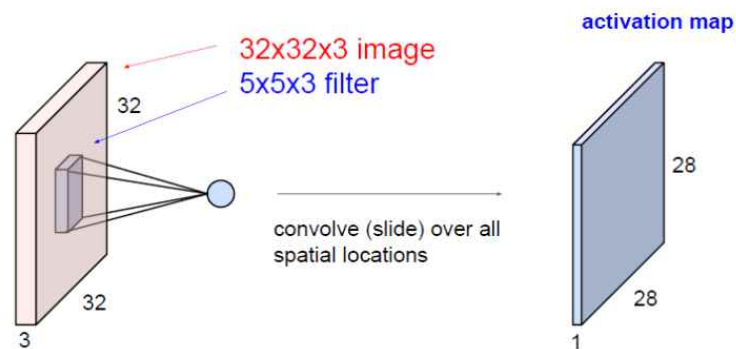
1. Fully connected layer

- (32, 32, 3)의 이미지가 input 으로 들어오면 (3072,1) 의 형태로 만들어 계산 수행



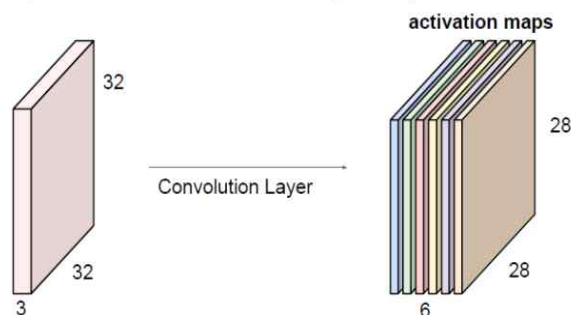
2. Convolution layer

- Filter를 하나 만들고 filter 가 움직이면서 output을 만듦
- filter 가 w 라 한다면, output의 한 점은 $w^T x + b$

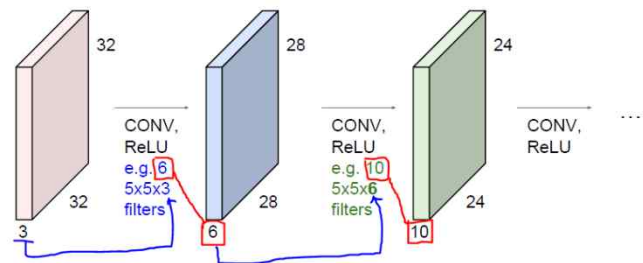


- 하나의 filter 는 항상 입력 채널 수만큼 확장
- filter의 수로 output의 activation map 수를 결정

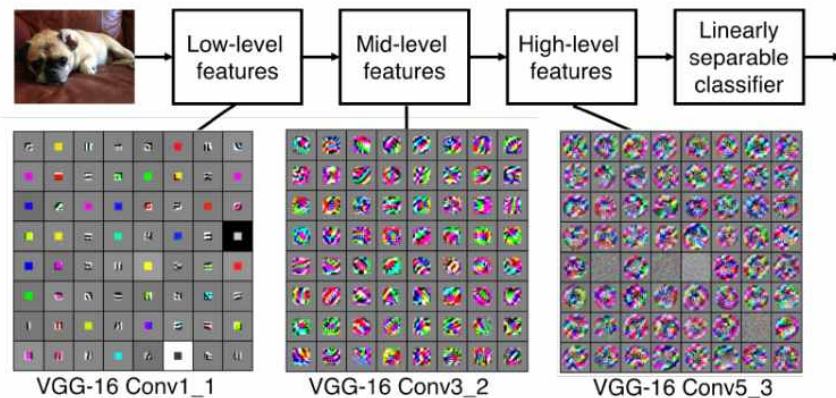
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



- 다른 filter들로 같은 위치에서 다른 특징들을 뽑아낼 수 있음
- convolution layer를 여러 번 수행하는 경우

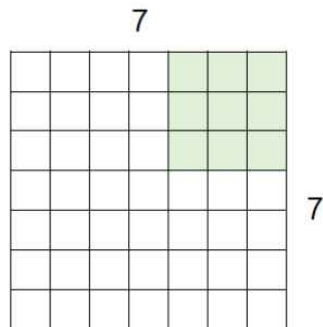


- input이 이미지일 때 각 layer마다 어떤 feature들이 뽑아질 수 있는지 확인 가능
- low-level부터 high-level 특징점까지 다양한 특징 추출



- stride : filter 가 입력 image를 sliding 할 때, 몇 칸씩 띄어가면서 계산을 할 것인지

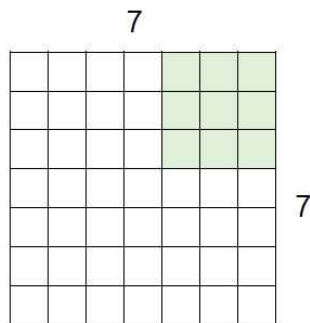
* stride = 1



7x7 input (spatially)
assume 3x3 filter

=> 5x5 output

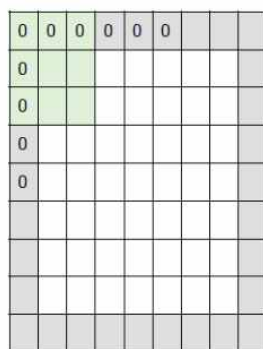
* stride = 2



7x7 input (spatially)
assume 3x3 filter
applied with **stride 2**
=> **3x3 output!**

- padding : convolution 과정 후 activation map 가로, 세로가 줄어들게 되는데 input size를 유지하기 위해 image 가장자리에 값을 채워주는 것.

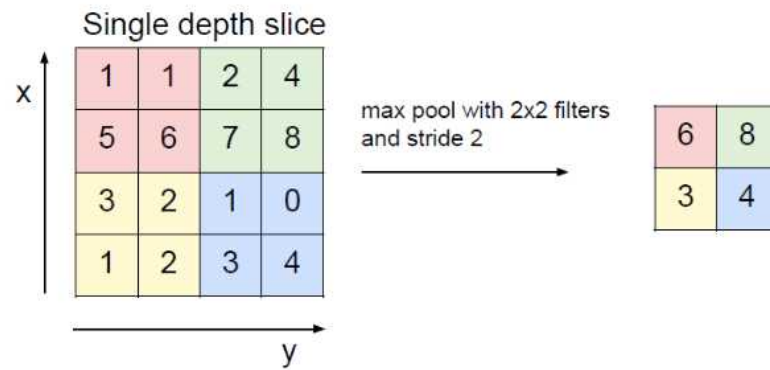
ex) zero padding



e.g. input 7x7
3x3 filter, applied with **stride 1**
pad with 1 pixel border => what is the output?

7x7 output!
in general, common to see CONV layers with
stride 1, filters of size FxF, and zero-padding with
(F-1)/2. (will preserve size spatially)
e.g. F = 3 => zero pad with 1
F = 5 => zero pad with 2
F = 7 => zero pad with 3

- pooling : 강제로 downsampling 하고 싶을 때 사용하는 기법.
 : depth에는 영향 x. 파라미터의 수를 줄게 하는 효과.
 : ex) max pooling



- convolution layer를 여러 개 사용하고 fully connected layer을 마지막에 사용하면 CNN 딥러닝 모델

