

Type something...

01) Convolutional Neural Networks

01_Convolution layer

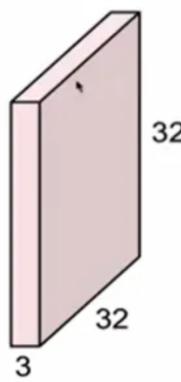
02_Pooling

01) Convolutional Neural Networks

01_Convolution layer

Convolution Layer

32x32x3 image



Filters always extend the full depth of the input volume

5x5x3 filter



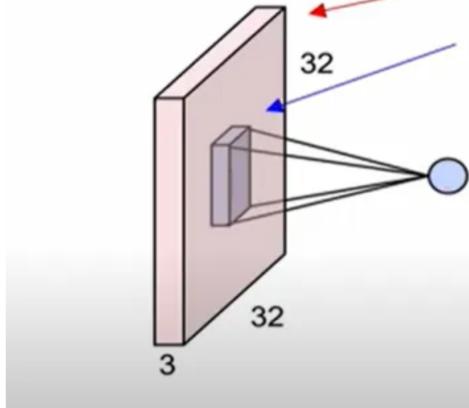
Convolve the filter with the image
i.e. "slide over the image spatially,
computing dot products"

- 항상 image와 filter의 depth는 같다는 것을 명심하자.

Convolution Layer

32x32x3 image

5x5x3 filter w

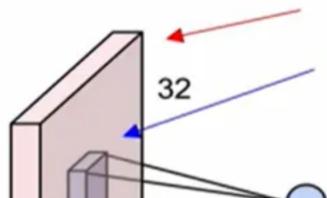


1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. $5 \times 5 \times 3 = 75$ -dimensional dot product + bias)

$$w^T x + b$$

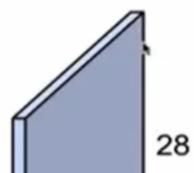
- convolution 연산을 통해서 하나의 숫자를 output으로 만든다.

Convolution Layer



32x32x3 image
5x5x3 filter

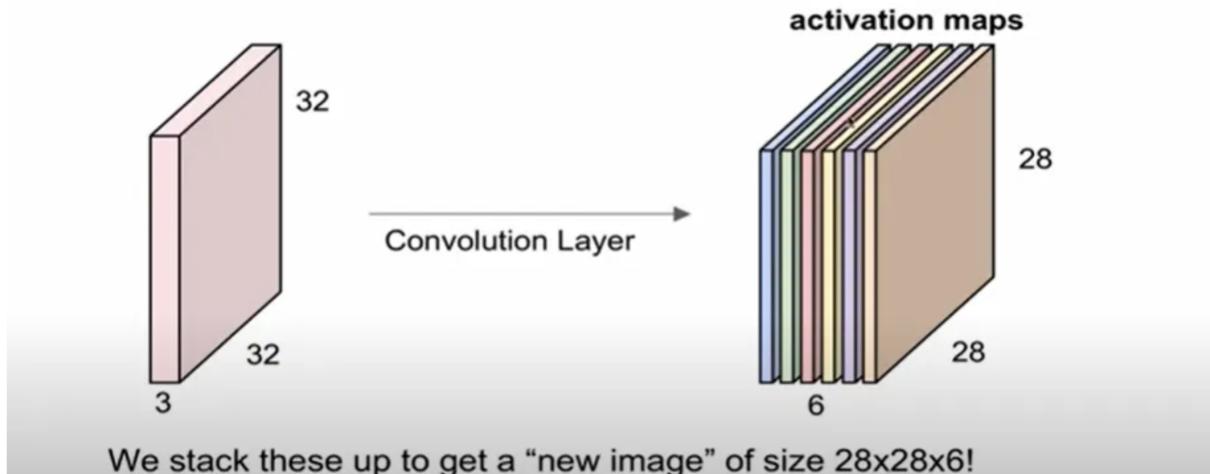
activation map





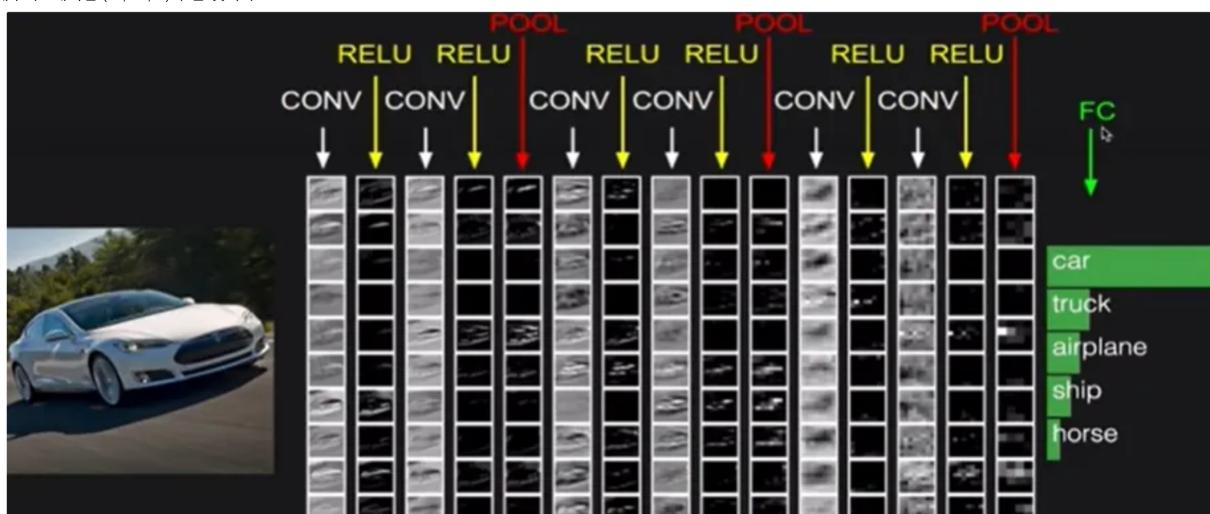
- 이 연산의 결과는 (28, 28 1)이 나오게 될 것이다.

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



We stack these up to get a “new image” of size 28x28x6!

- 만약 filter 개수가 6개이면 (28, 28, 6)가 될 것이다.



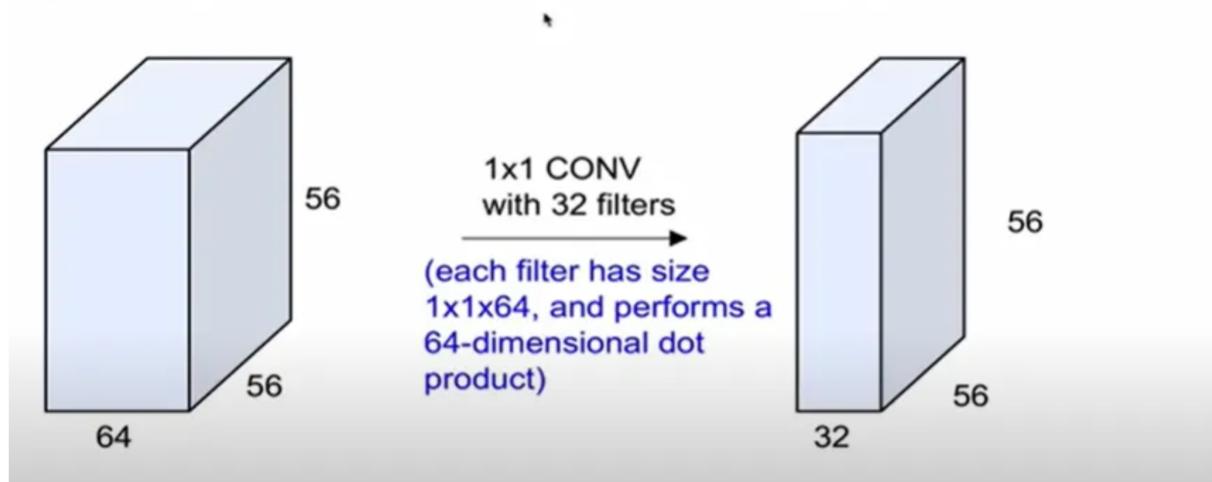
- conv → ReLU → conv → ReLU → Pool x 3 → fc

Summary. To summarize, the Conv Layer:

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires four hyperparameters:
 - Number of filters K ,
 - their spatial extent F ,
 - the stride S ,
 - the amount of zero padding P .
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
 - $W_2 = (W_1 - F + 2P)/S + 1$
 - $H_2 = (H_1 - F + 2P)/S + 1$ (i.e. width and height are computed equally by symmetry)
 - $D_2 = K$
- With parameter sharing, it introduces $F \cdot F \cdot D_1$ weights per filter, for a total of $(F \cdot F \cdot D_1) \cdot K$ weights and K biases.
- In the output volume, the d -th depth slice (of size $W_2 \times H_2$) is the result of performing a valid convolution of the d -th filter over the input volume with a stride of S , and then offset by d -th bias.

- $(w - F + 2P) / S + 1$ 로 사이즈가 계산된다.

(btw, 1x1 convolution layers make perfect sense)

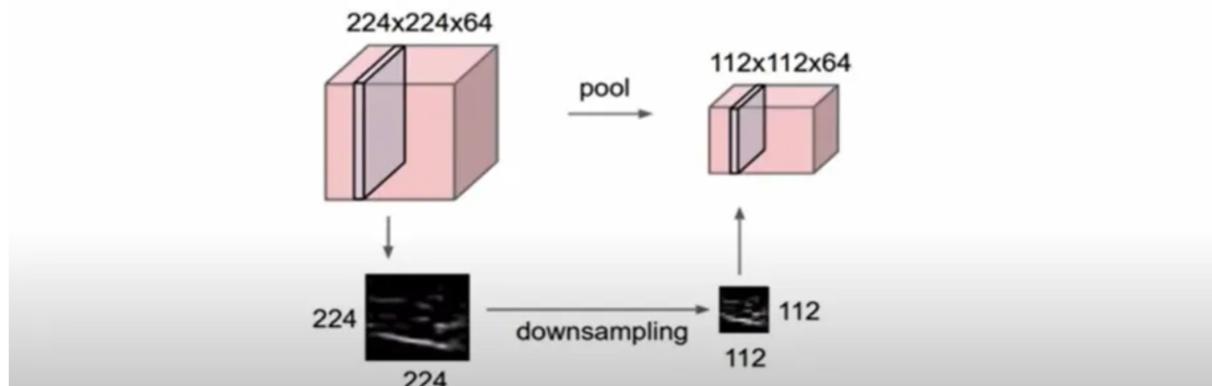


- 1x1 convolution을 통과하면 64차원 dot product를 수행하기 때문에 의미가 있다.

02_Pooling

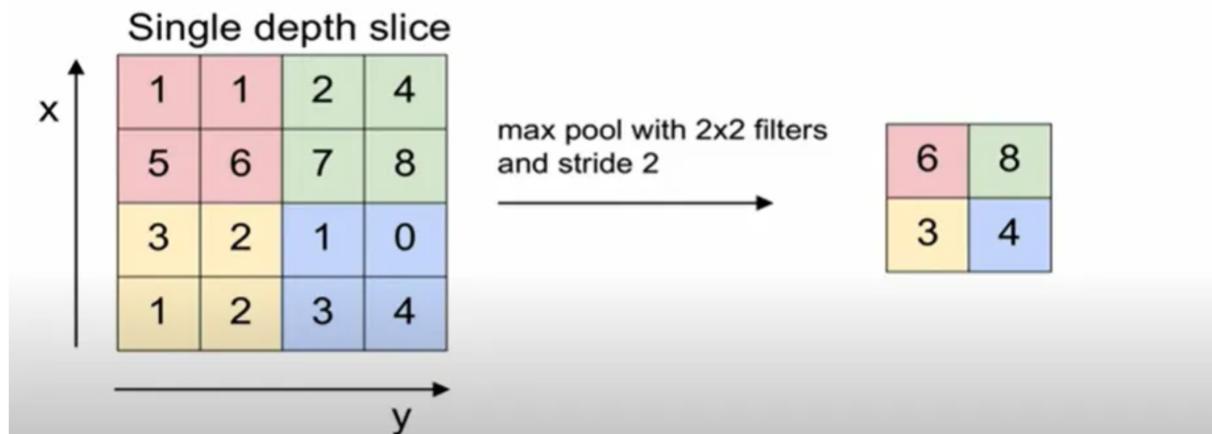
Pooling layer

- makes the representations smaller and more manageable
- operates over each activation map independently:



- 사이즈를 줄여서 관리를 더 쉽게 해주는 과정이다.

MAX POOLING



- 제일 많이 쓰는 pooling은 max pooling이다.

