Keras와 함께 하는 딥러닝 기초

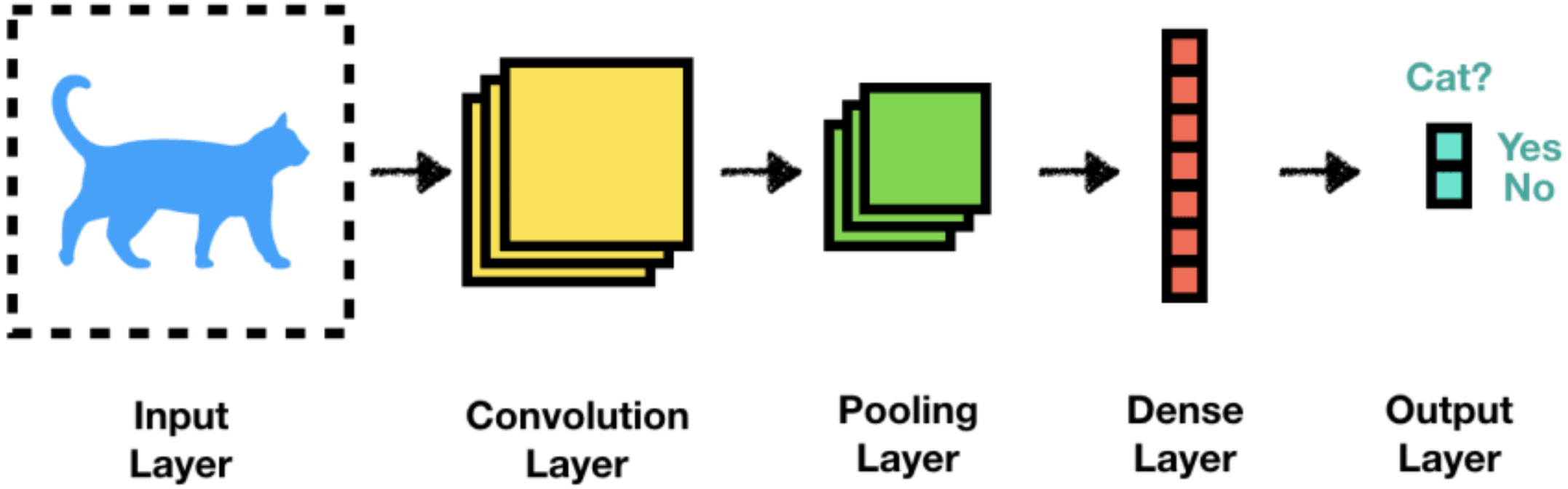
이정근

소프트웨어 융합 대학 (School of Software)

JeongGun.Lee@hallym.ac.kr / www.onchip.net

Convolutional Neural Networks (CNN)

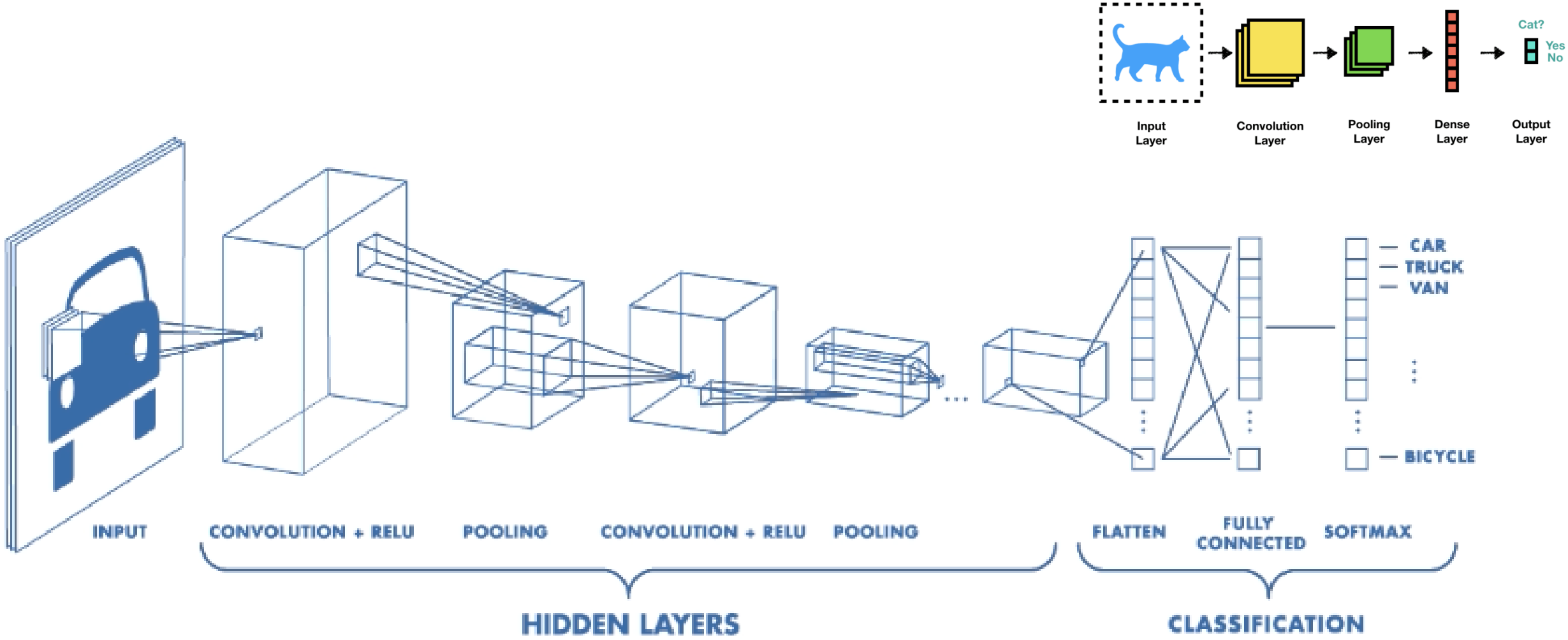
Keras 딥러닝: CNN 동작 원리



https://github.com/jeonggunlee/DeepLearningBasics/blob/master/Lab07_introduction_to_convnets.ipynb

* https://github.com/rickiepark/deep-learning-with-python-notebooks

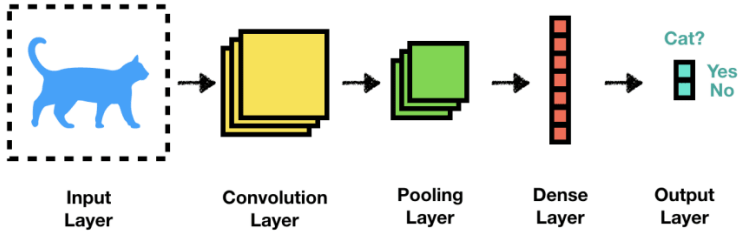
Keras 딥러닝: CNN 동작 원리



https://github.com/jeonggunlee/DeepLearningBasics/blob/master/Lab07_introduction_to_convnets.ipynb

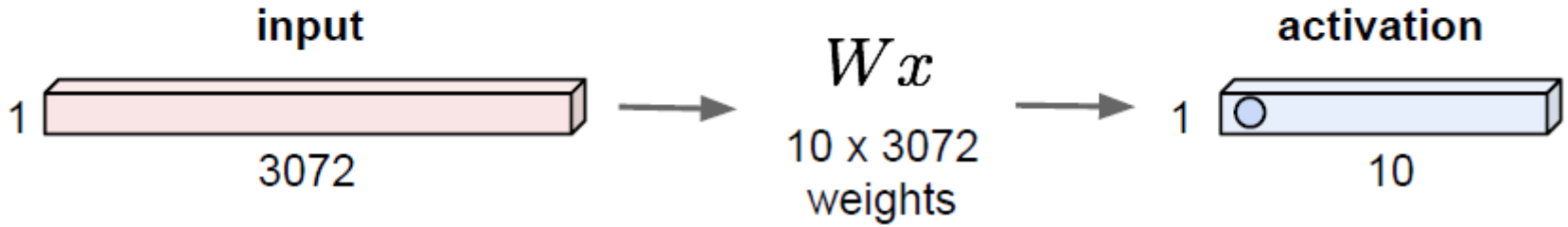
* https://github.com/rickiepark/deep-learning-with-python-notebooks

Keras 딥러닝: CNN 동작 원리

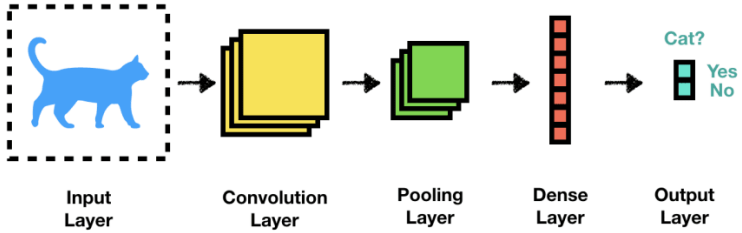


Fully Connected Layer

32x32x3 image -> stretch to 3072 x 1

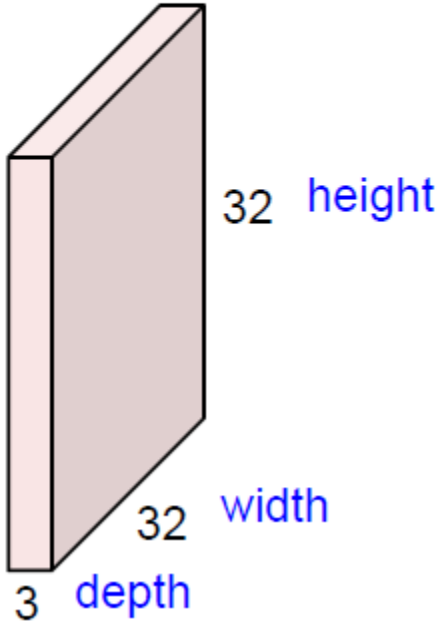


Keras 딥러닝: CNN 동작 원리

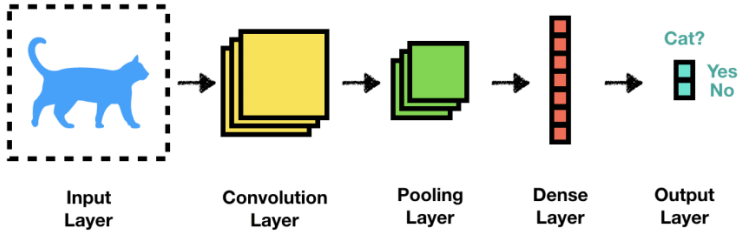


Convolution Layer

32x32x3 image -> preserve spatial structure

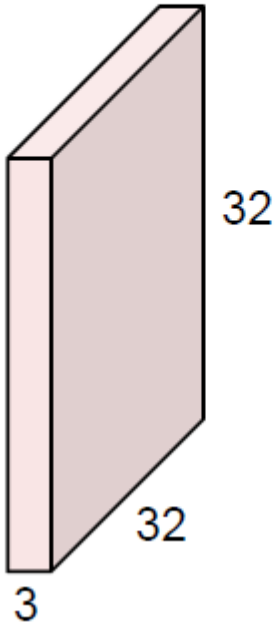


Keras 딥러닝: CNN 동작 원리

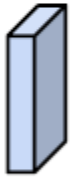


Convolution Layer

32x32x3 image

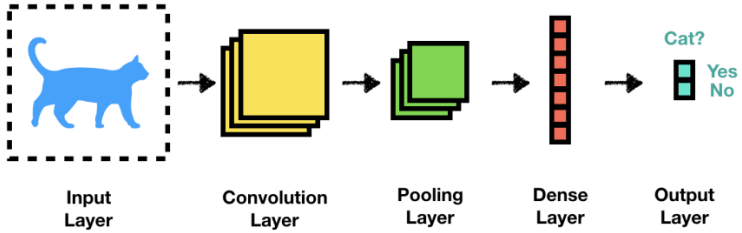


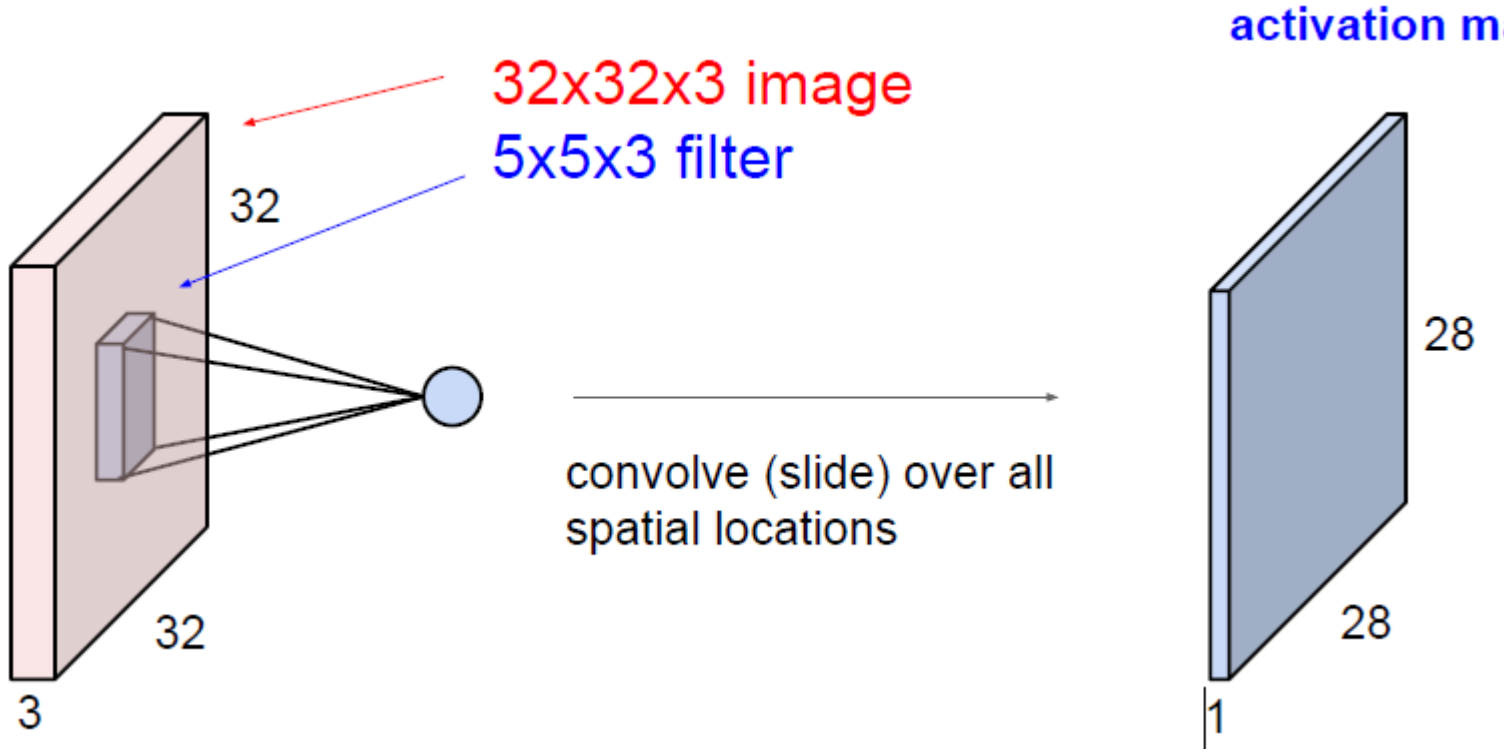
5x5x3 filter



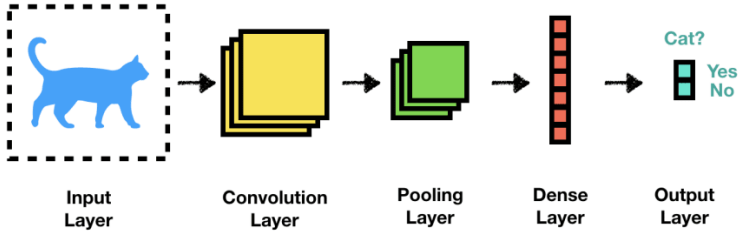
Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

Keras 딥러닝: CNN 동작 원리

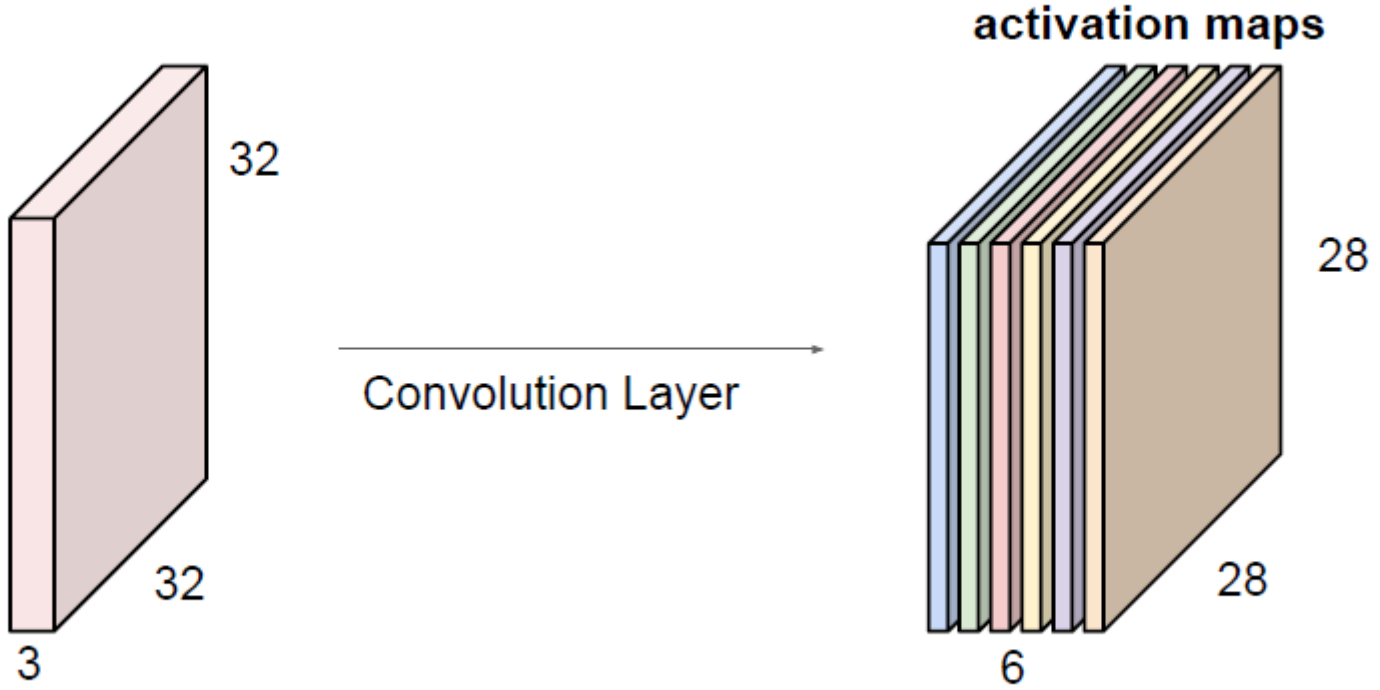




Keras 딥러닝: CNN 동작 원리

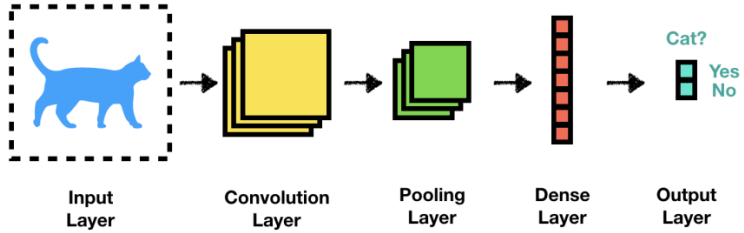


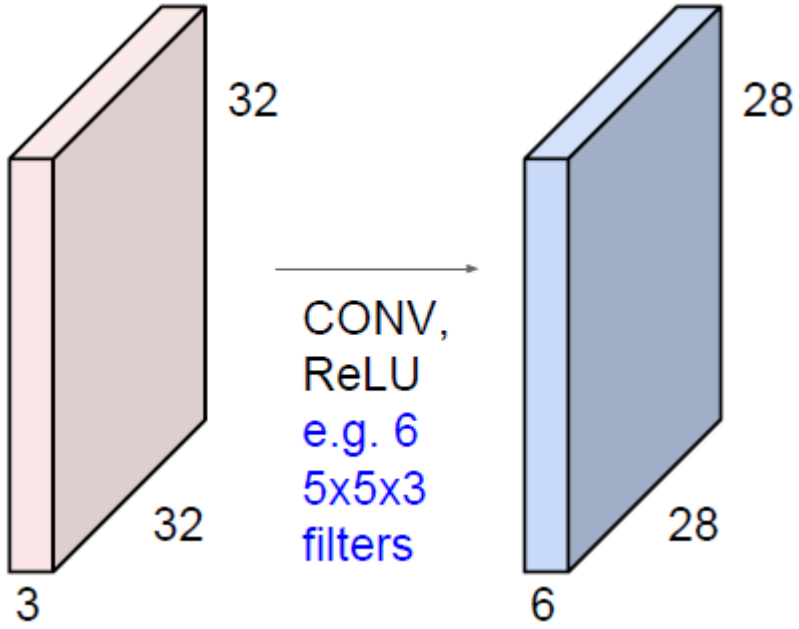
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



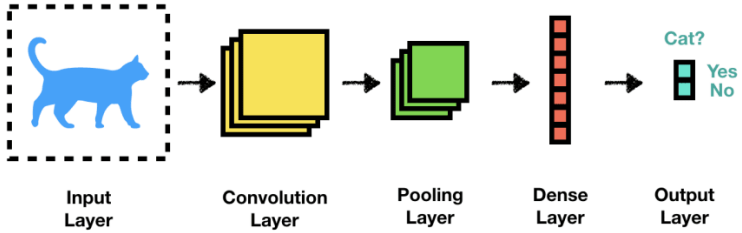
We stack these up to get a "new image" of size 28x28x6!

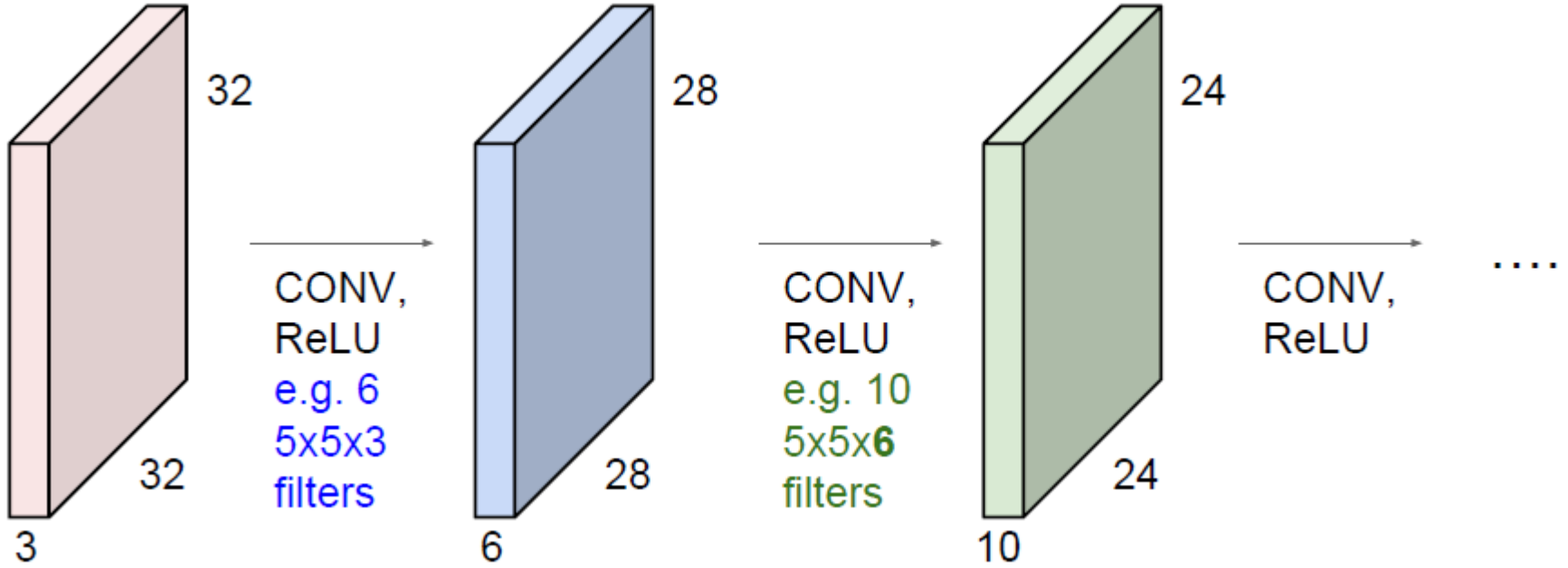
Keras 딥러닝: CNN 동작 원리



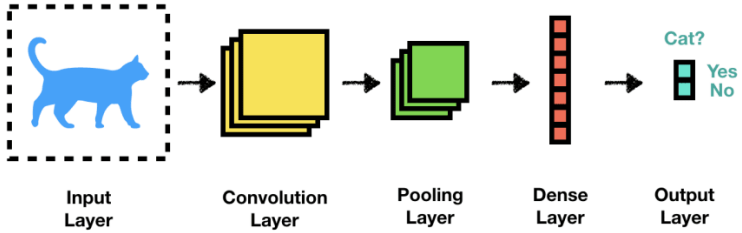


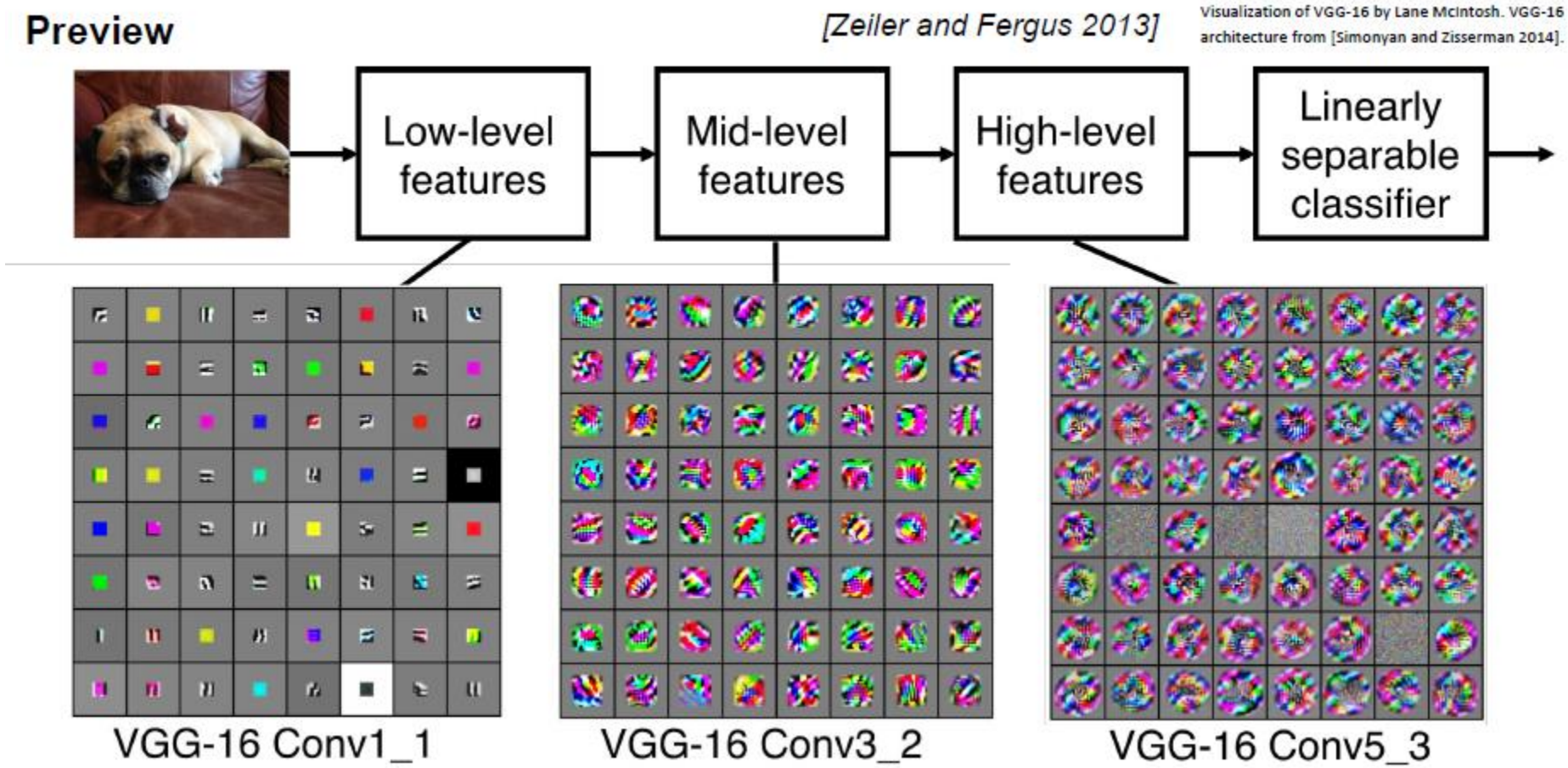
Keras 딥러닝: CNN 동작 원리



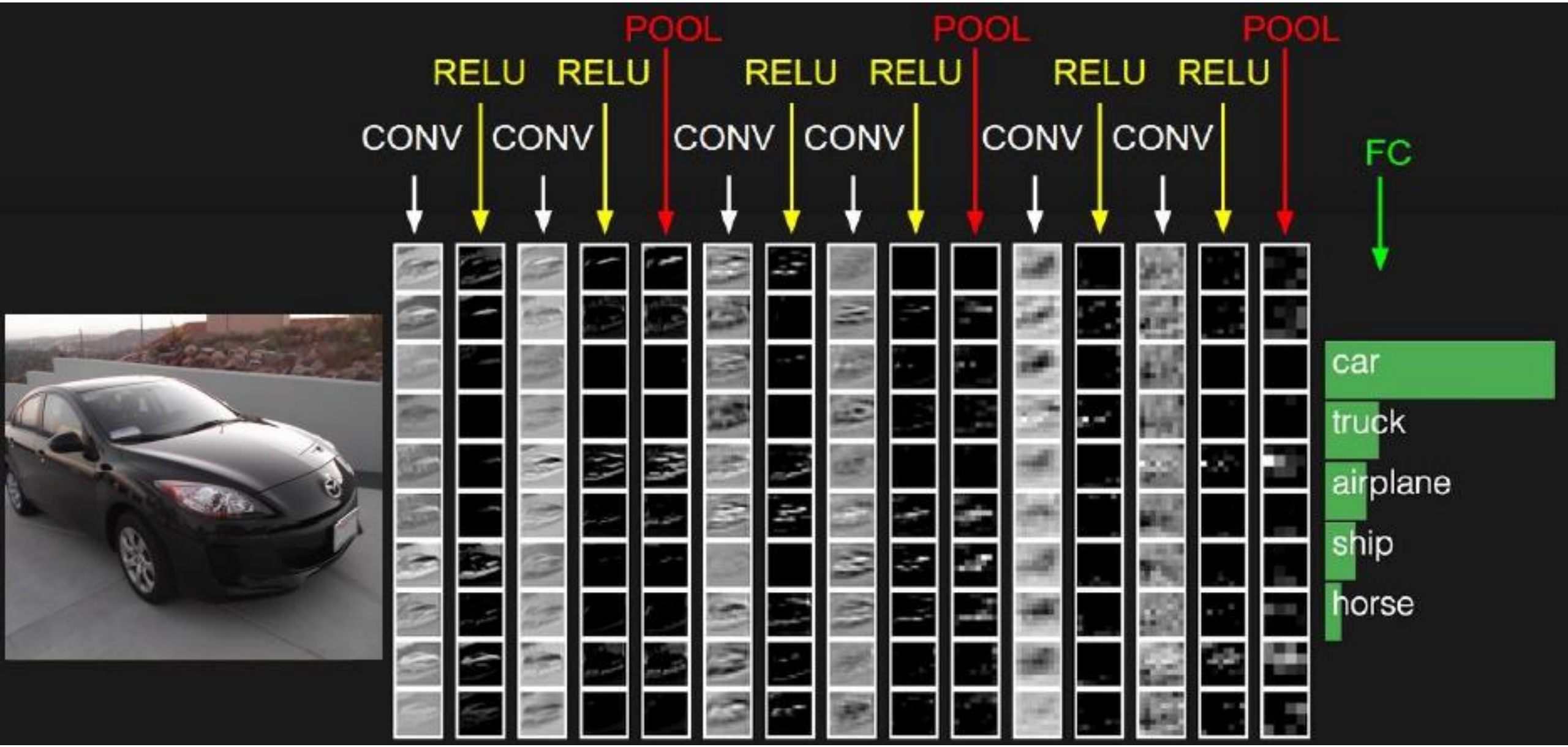


Keras 딥러닝: CNN 동작 원리

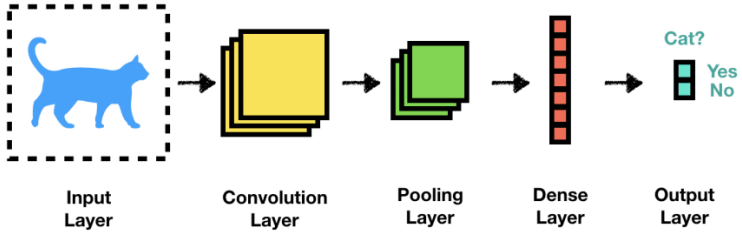




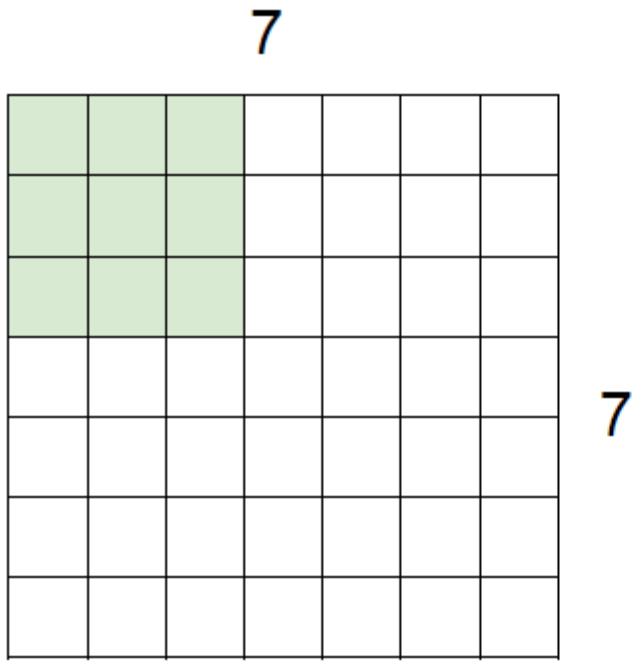
Keras 딥러닝: CNN 동작 원리



Keras 딥러닝: CNN 동작 원리

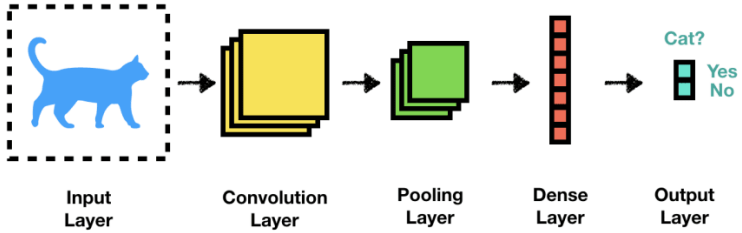


A closer look at spatial dimensions:

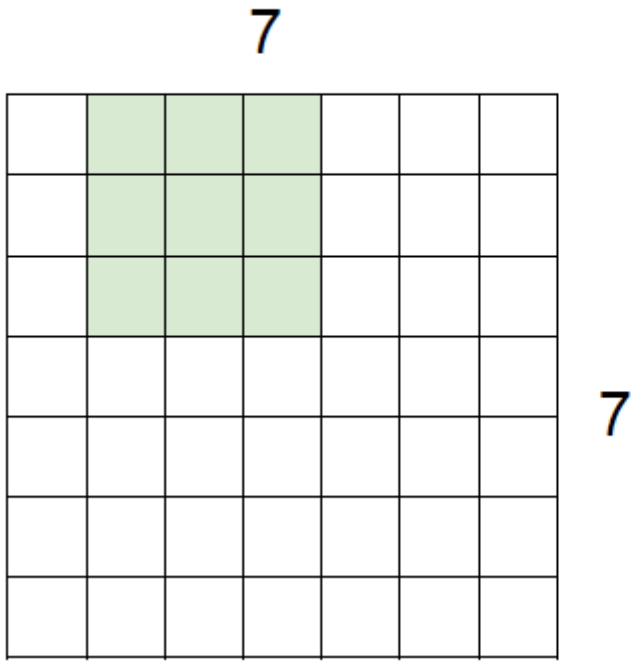


7x7 input (spatially)
assume 3x3 filter

Keras 딥러닝: CNN 동작 원리

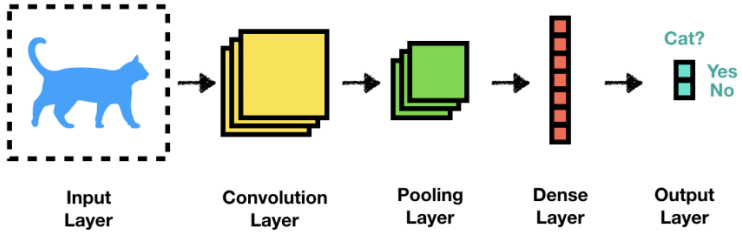


A closer look at spatial dimensions:

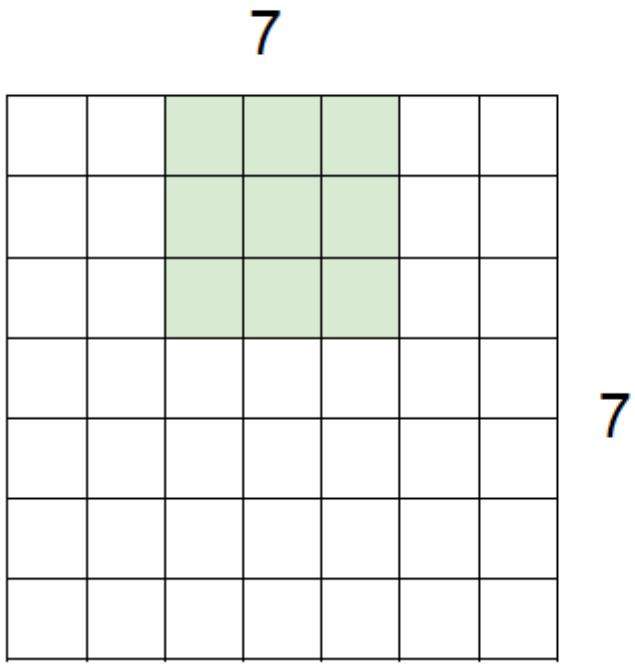


7x7 input (spatially)
assume 3x3 filter

Keras 딥러닝: CNN 동작 원리

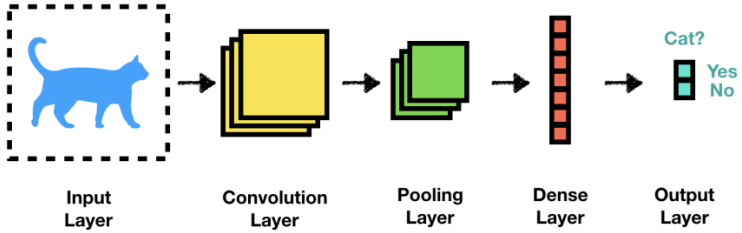


A closer look at spatial dimensions:

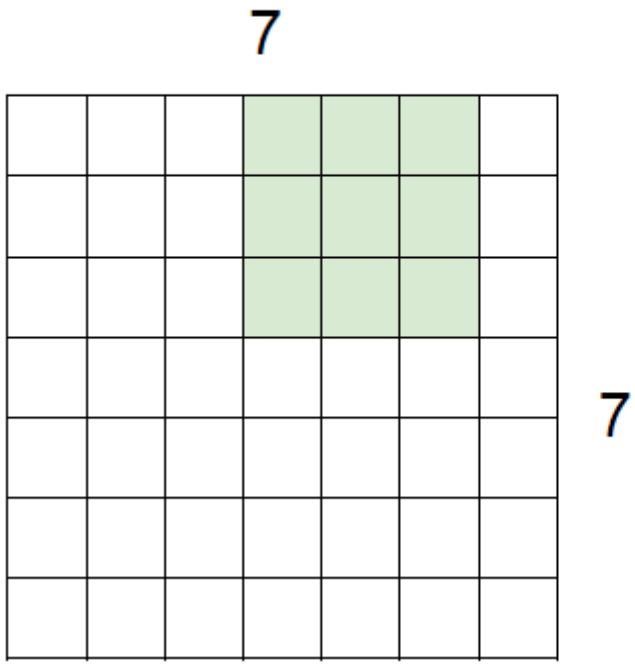


7x7 input (spatially)
assume 3x3 filter

Keras 딥러닝: CNN 동작 원리

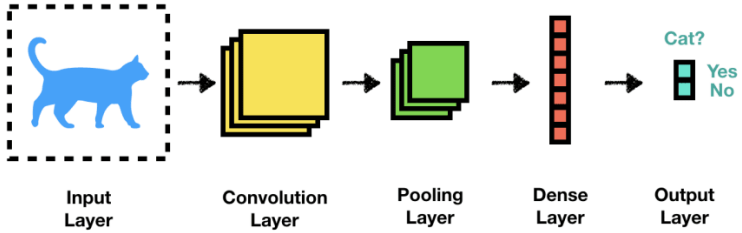


A closer look at spatial dimensions:

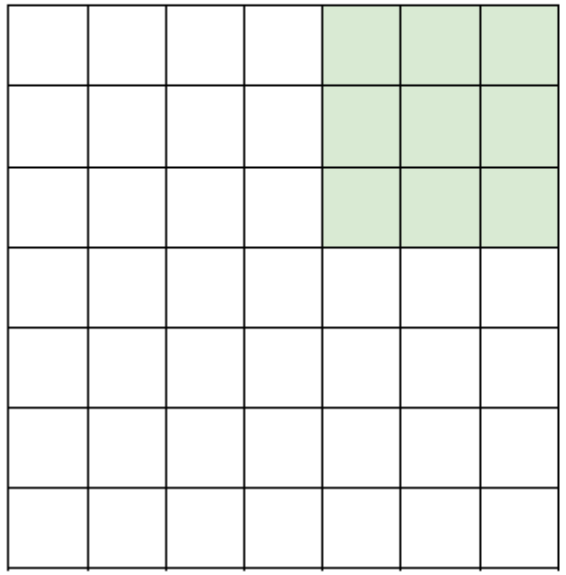


7x7 input (spatially)
assume 3x3 filter

Keras 딥러닝: CNN 동작 원리



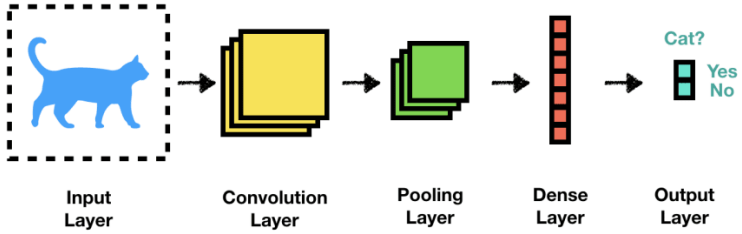
A closer look at spatial dimensions:



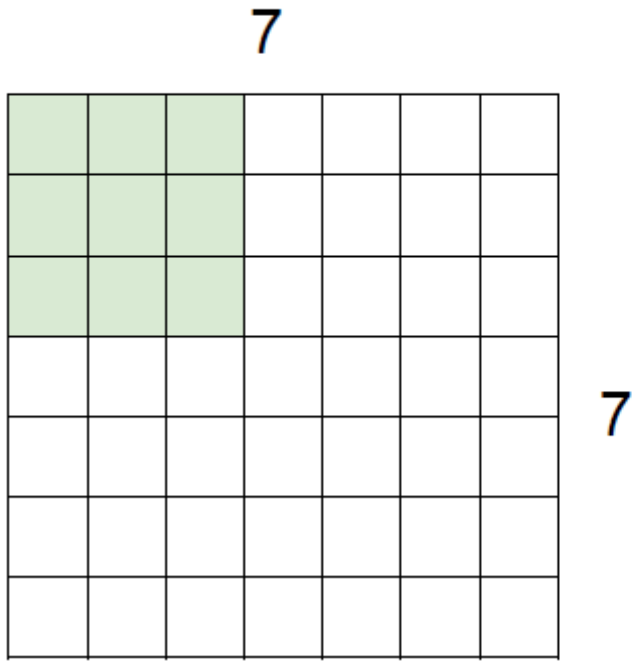
7x7 input (spatially)
assume 3x3 filter

=> 5x5 output

Keras 딥러닝: CNN 동작 원리

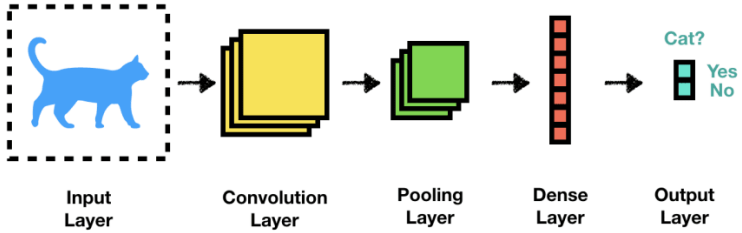


A closer look at spatial dimensions:

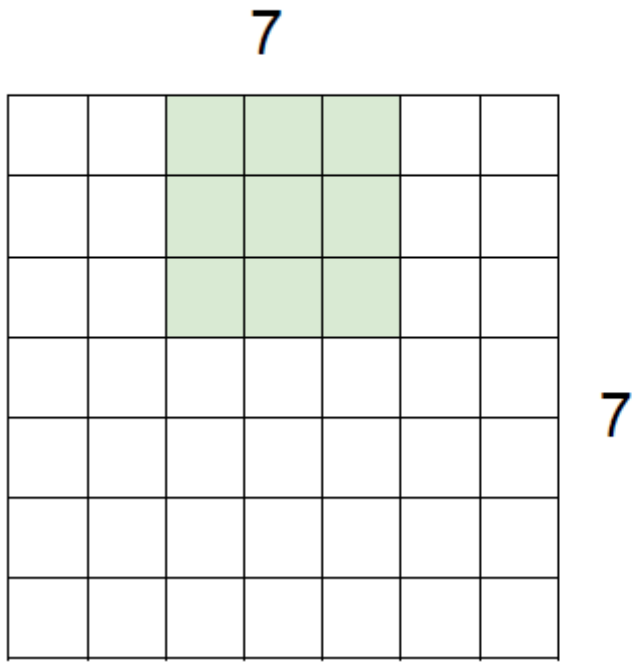


7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**

Keras 딥러닝: CNN 동작 원리

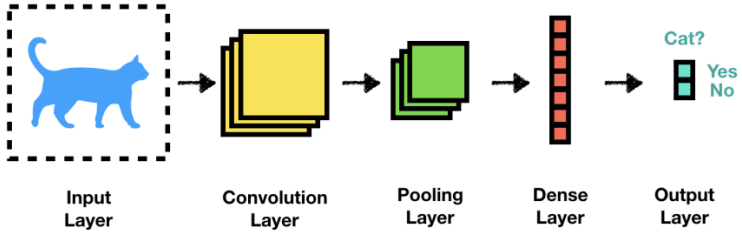


A closer look at spatial dimensions:

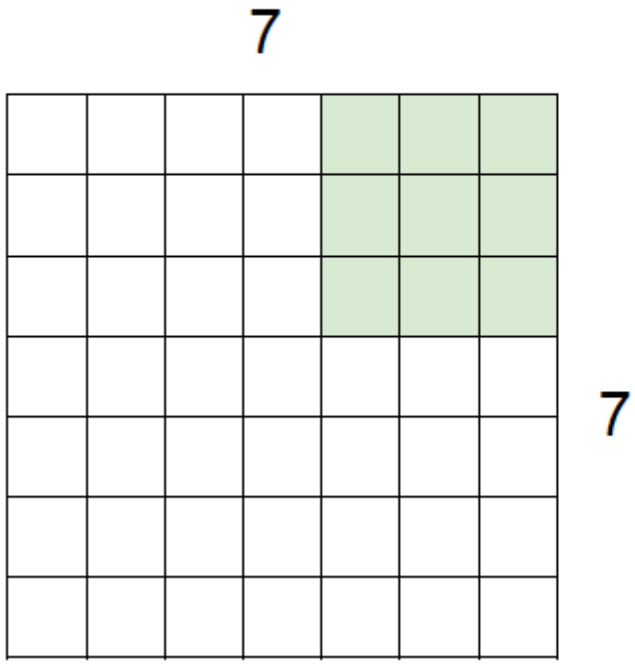


7x7 input (spatially)
assume 3x3 filter
applied **with stride 2**

Keras 딥러닝: CNN 동작 원리

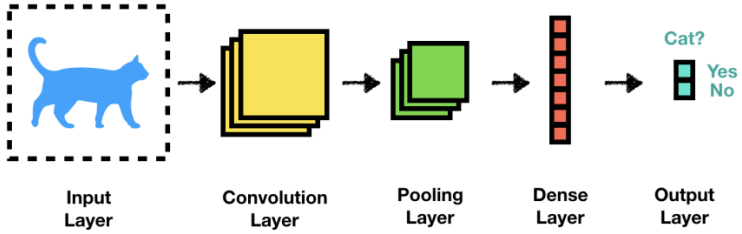


A closer look at spatial dimensions:



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

Keras 딥러닝: CNN 동작 원리



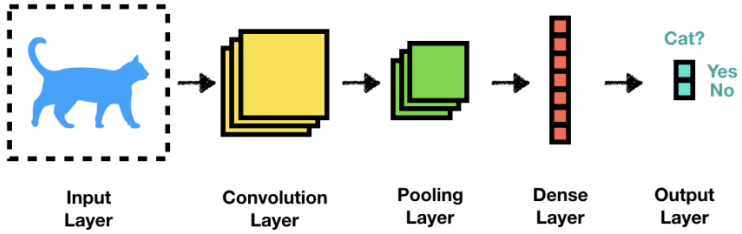
In practice: Common to zero pad the border

0	0	0	0	0	0			
0								
0								
0								
0								

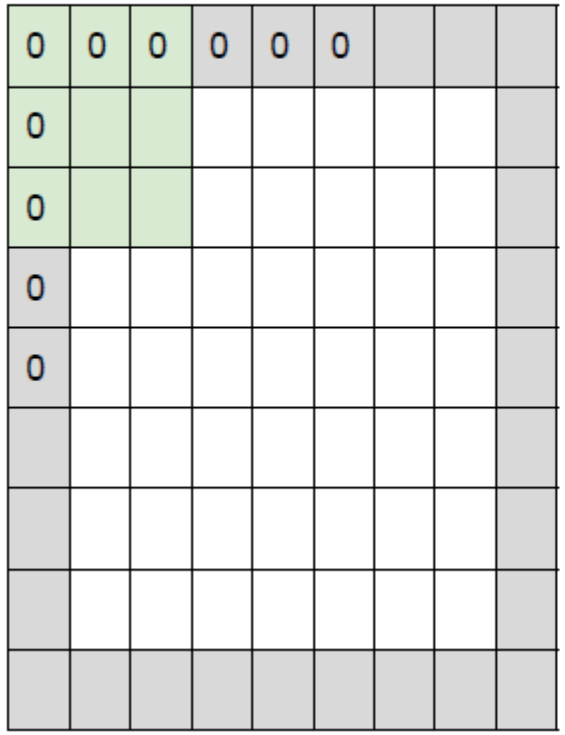
e.g. input 7x7
3x3 filter, applied with **stride 1**
pad with 1 pixel border => what is the output?

(recall:)
(N - F) / stride + 1

Keras 딥러닝: CNN 동작 원리



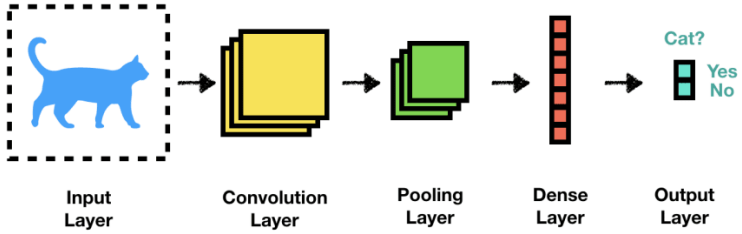
In practice: Common to zero pad the border



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

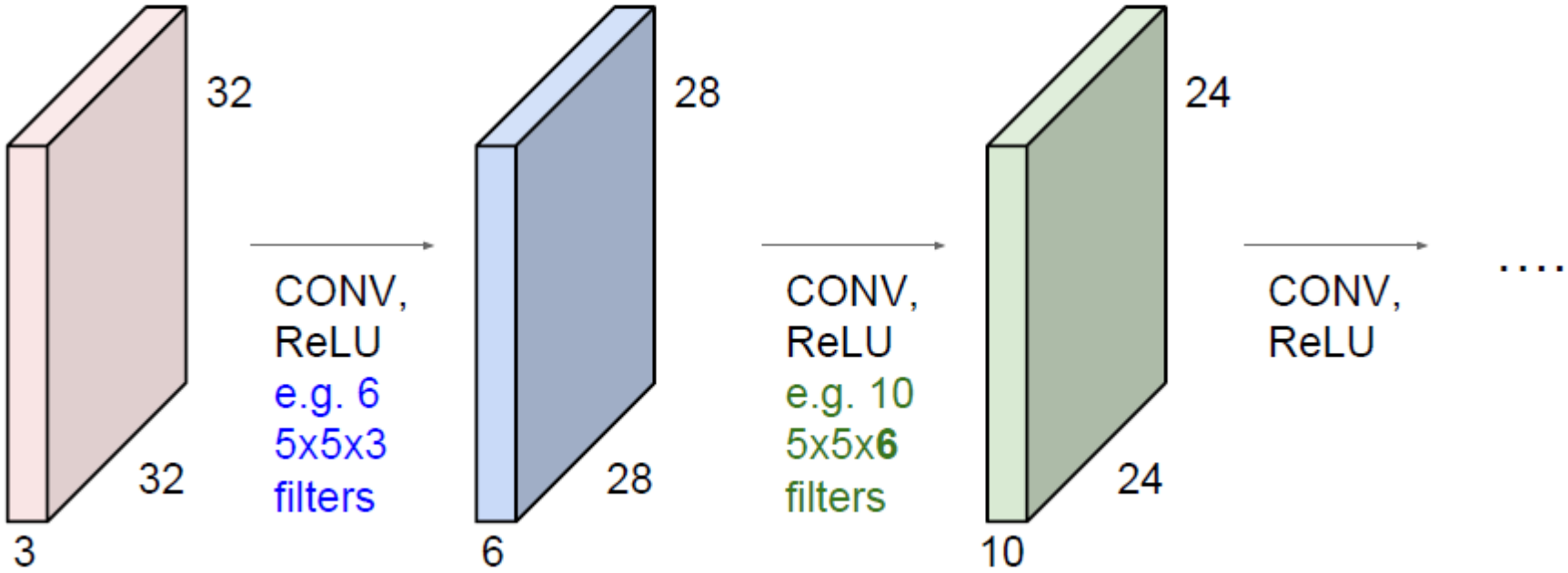
7x7 output!
in general, common to see CONV layers with stride 1, filters of size FxF, and zero-padding with (F-1)/2. (will preserve size spatially)
e.g. F = 3 => zero pad with 1
F = 5 => zero pad with 2
F = 7 => zero pad with 3

Keras 딥러닝: CNN 동작 원리

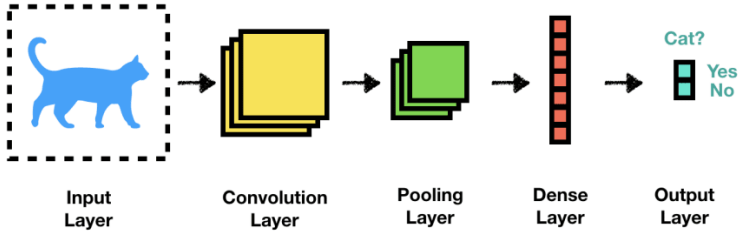


Remember back to…

E.g. 32x32 input convolved repeatedly with 5x5 filters shrinks volumes spatially! (32 -> 28 -> 24 ...). Shrinking too fast is not good, doesn't work well.



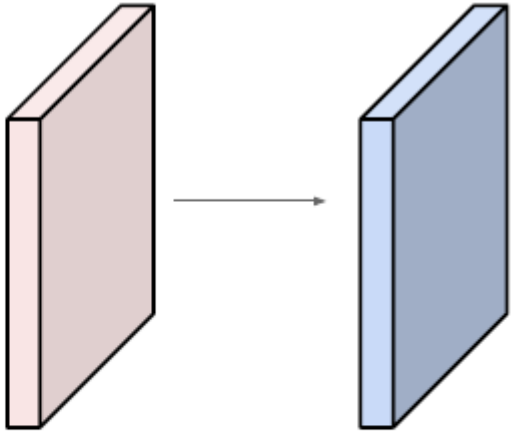
Keras 딥러닝: CNN 동작 원리



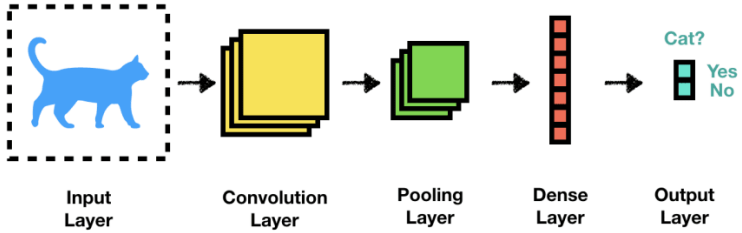
Examples time:

Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

Output volume size: ?

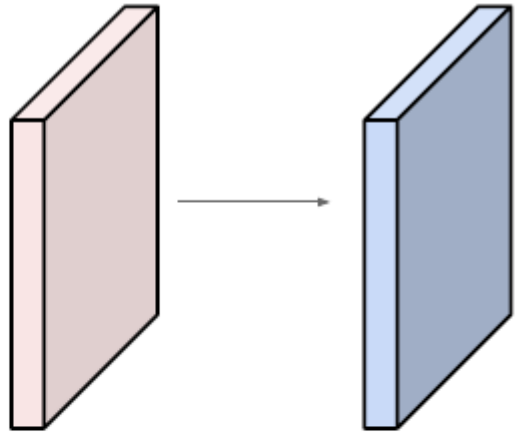


Keras 딥러닝: CNN 동작 원리



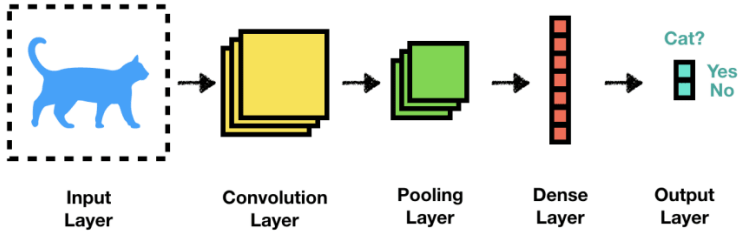
Examples time:

Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2



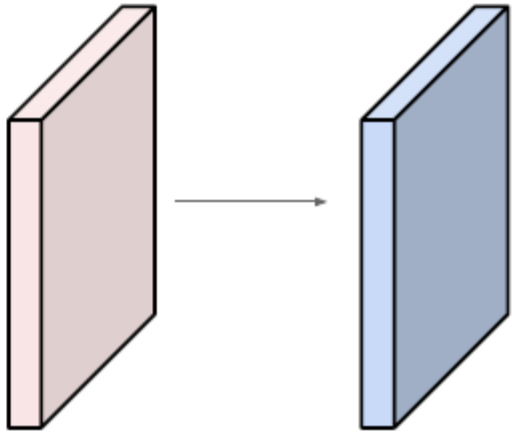
Output volume size:
(32+2*2-5)/1+1 = 32 spatially, so
32x32x10

Keras 딥러닝: CNN 동작 원리



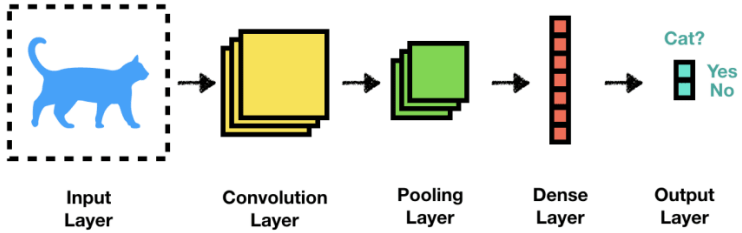
Examples time:

Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

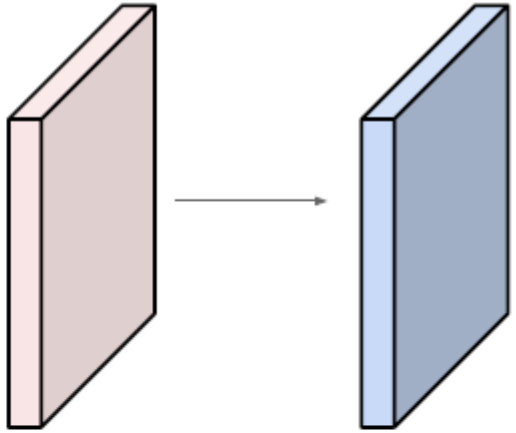


Number of parameters in this layer?

Keras 딥러닝: CNN 동작 원리



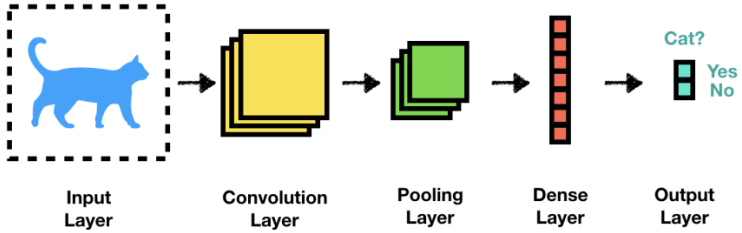
Examples time:



Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

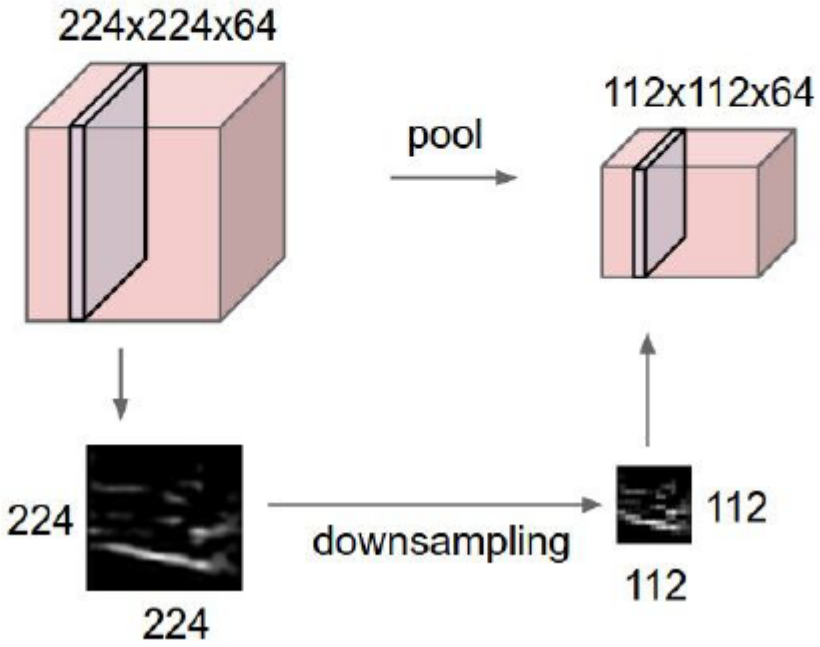
Number of parameters in this layer?
each filter has 5*5*3 + 1 = 76 params (+1 for bias)
=> 76*10 = **760**

Keras 딥러닝: CNN 동작 원리

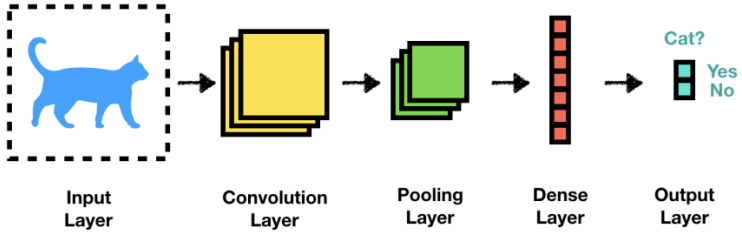


Pooling layer

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



Keras 딥러닝: CNN 동작 원리



MAX POOLING

Single depth slice

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

x

y

max pool with 2x2 filters and stride 2

6	8
3	4