

Deep Learning Seminar

4. Convolutional Neural Network

Contents

1. CNN overview
2. Convolution Filter & Feature-map
3. Activation Function & Pooling

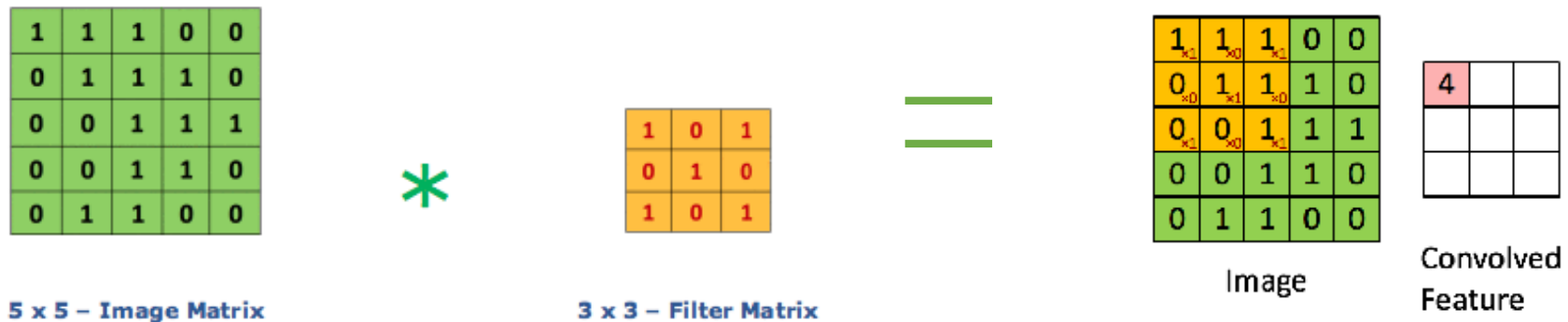
What is CNN ?

- Convolution Neural Network (CNN)

영상처리 (Computer Vision)에서 널리 사용하는 기계학습 방법론 중 하나.

이미지의 전체가 아닌 여러 개의 **작은 부분들을 보고 (by Convolution Filter)**

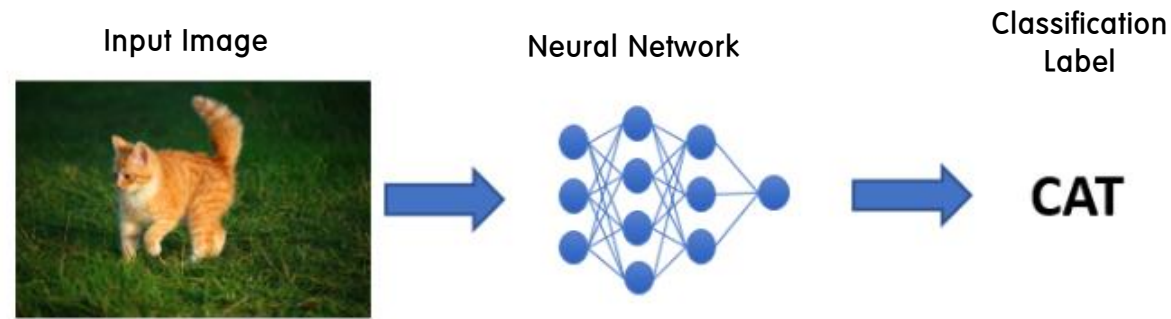
핵심적인 정보 (Feature-map)를 추출하여 해당 정보를 이용하여 원하는 결과를 예측하는 기계학습 방법론.



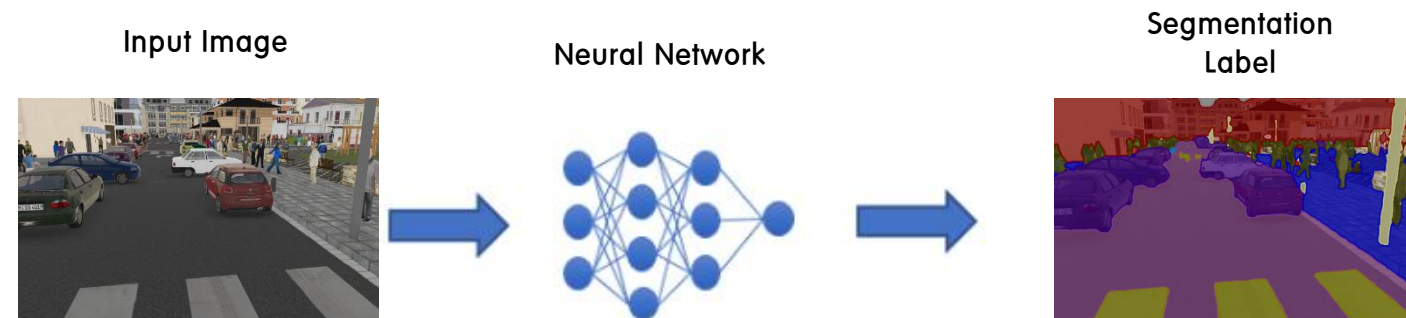
< Convolution Filter >

What can we do using CNN?

- Classification

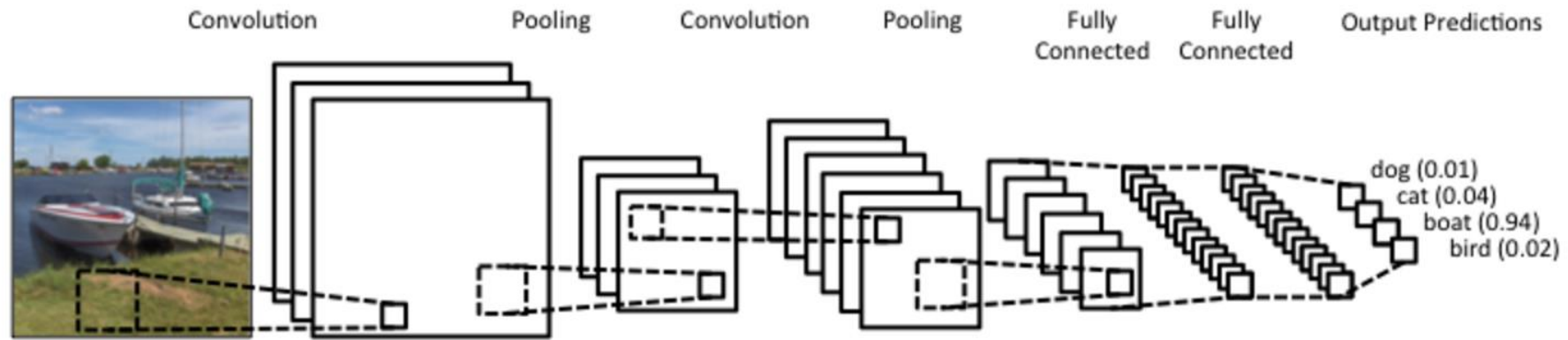


- Segmentation



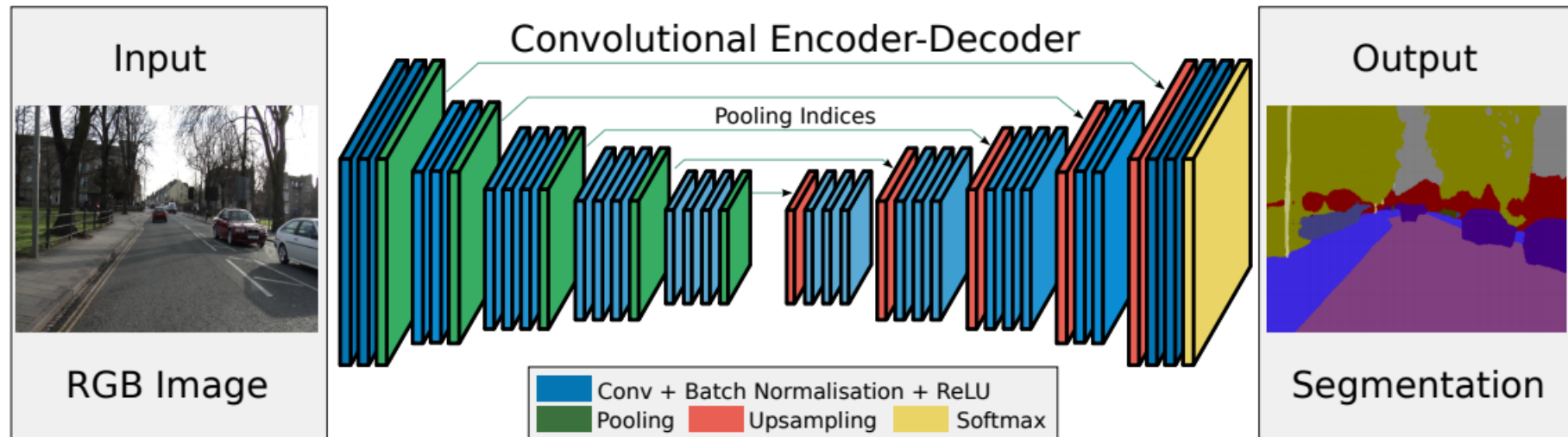
How does CNN work ?

- Classification Model



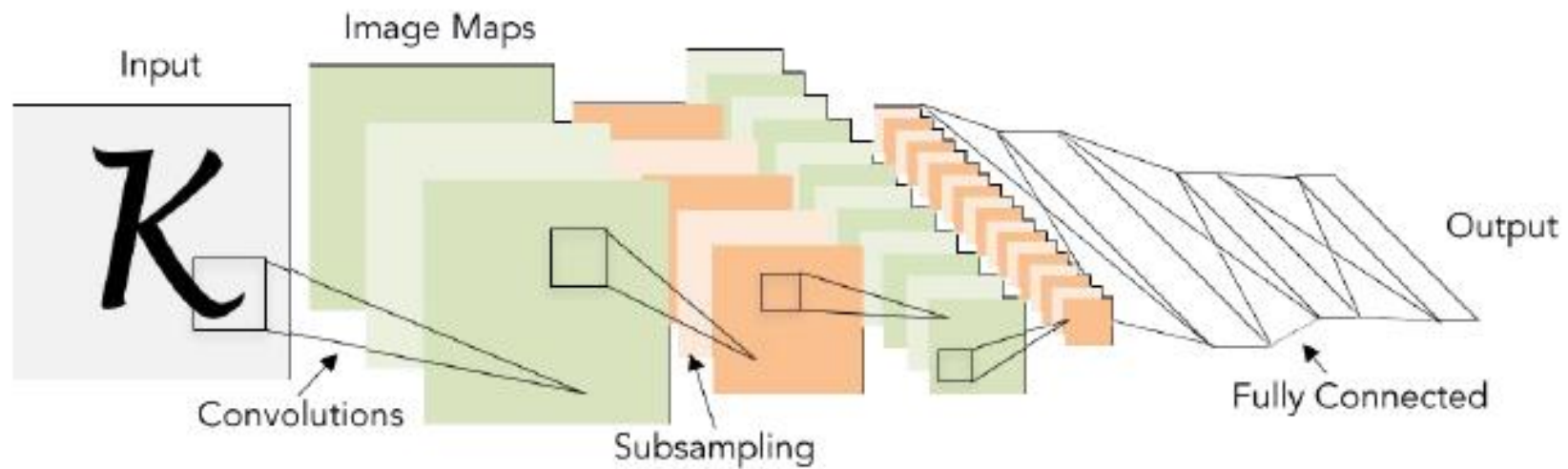
How does CNN work ?

- Segmentation Model



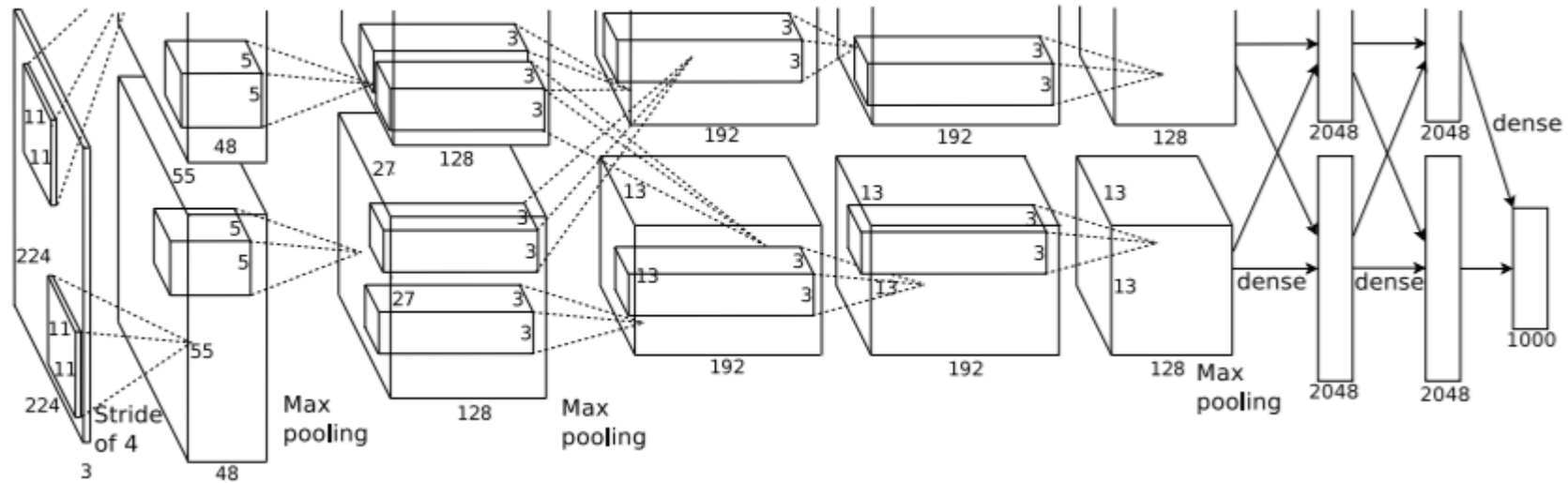
CNN History

- 1) LeNet (1998)



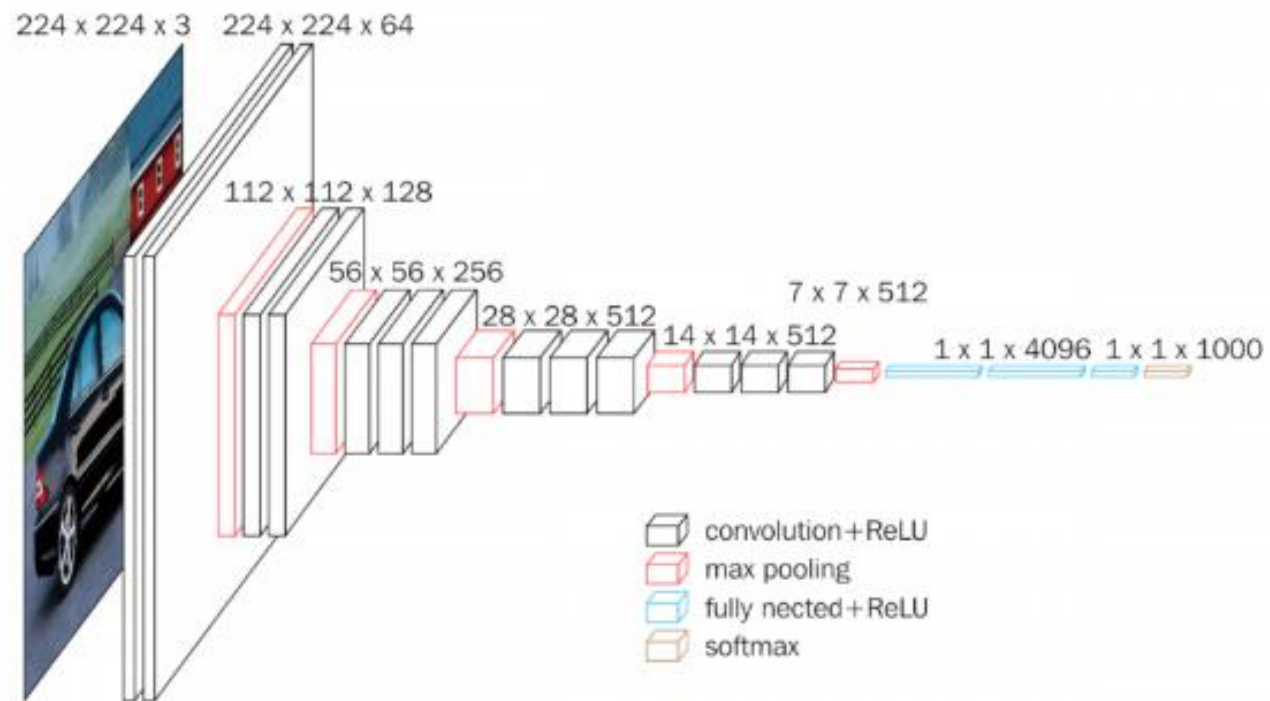
CNN History

- 2) AlexNet (2012)



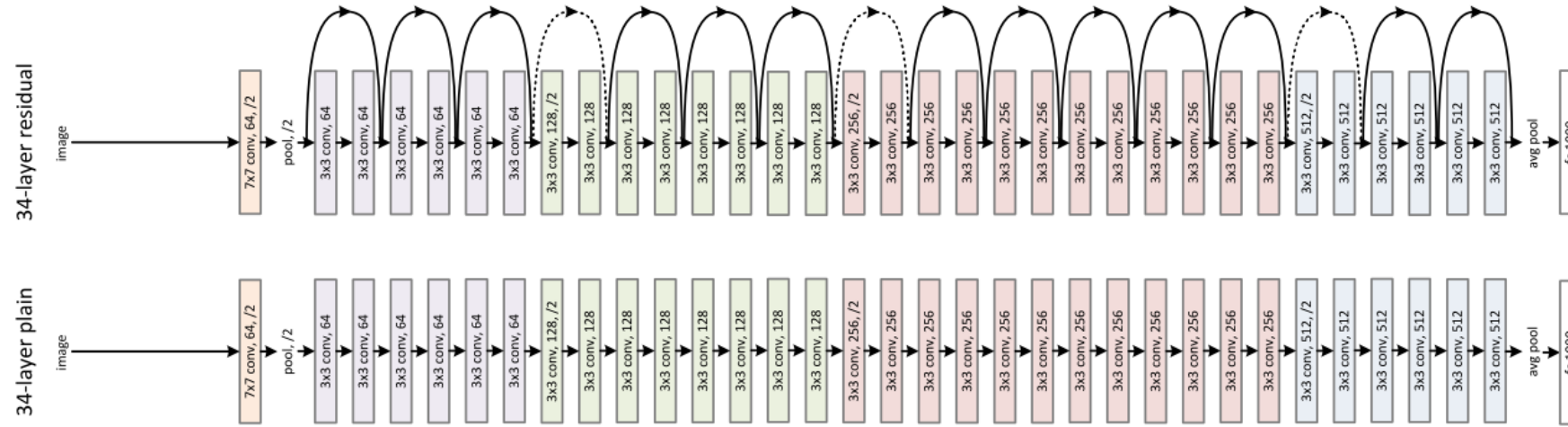
CNN History

- 3) VGG (2014)



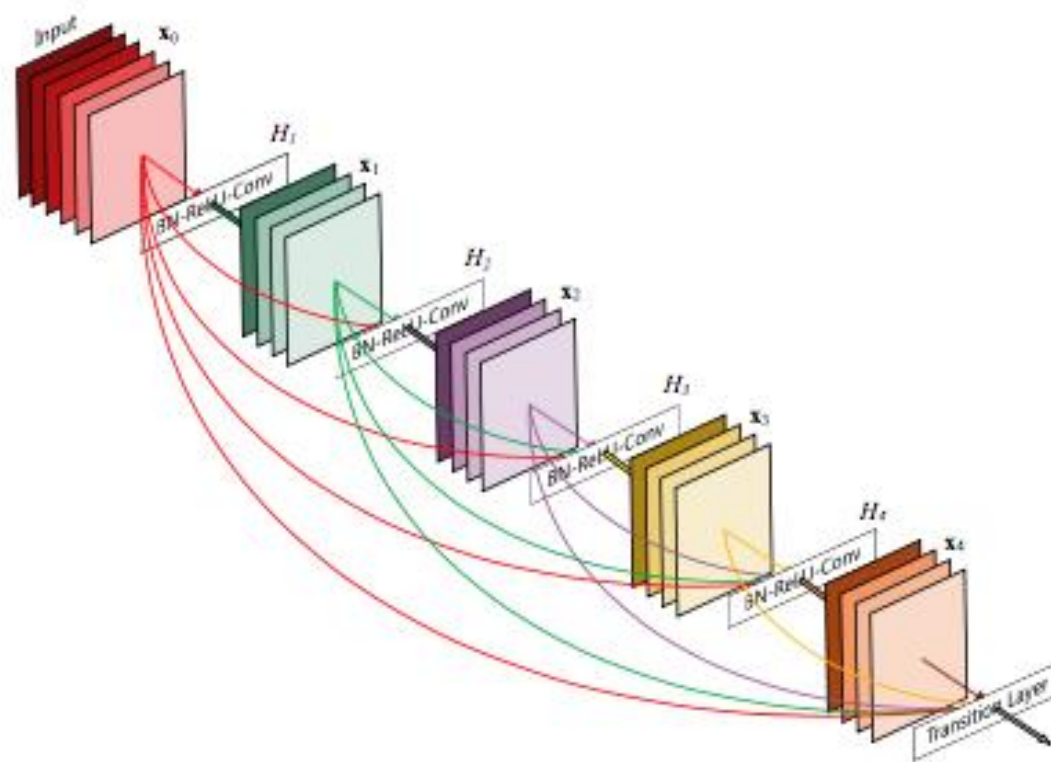
CNN History

- 4) ResNet (2015)



CNN History

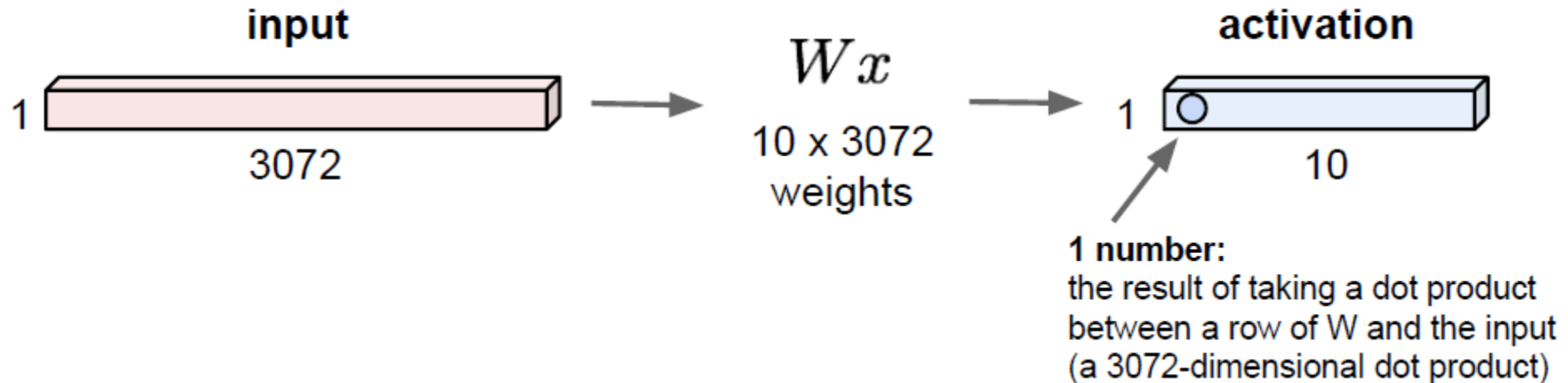
- 5) DenseNet (2016)



Convolutional Neural Network

- Fully Connection Layer

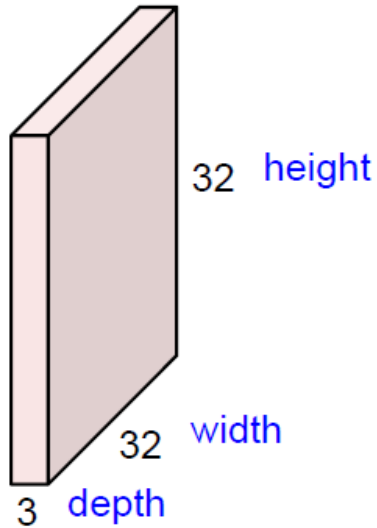
32x32x3 image -> stretch to 3072 x 1



Convolutional Neural Network

- Convolution Layer

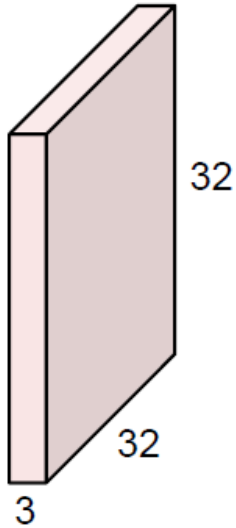
32x32x3 image -> preserve spatial structure



Convolutional Neural Network

- Convolution Layer

32x32x3 image



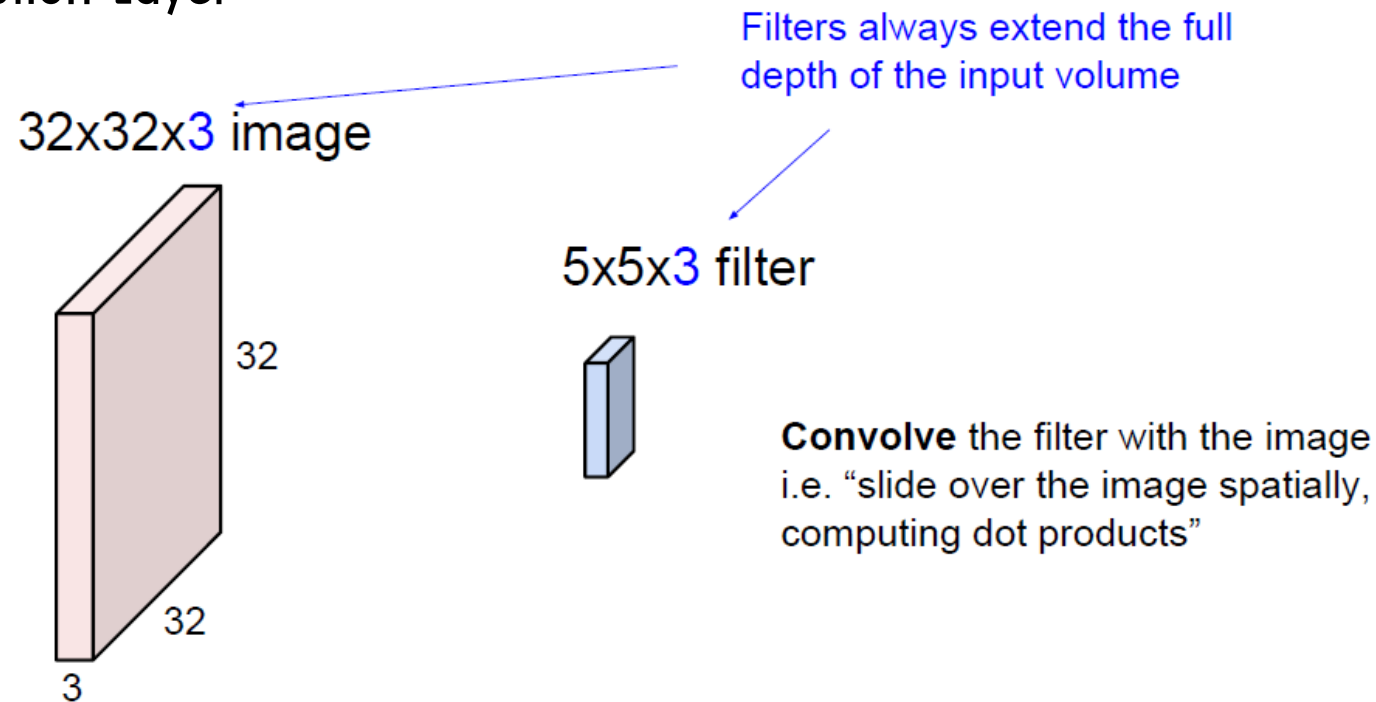
5x5x3 filter



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

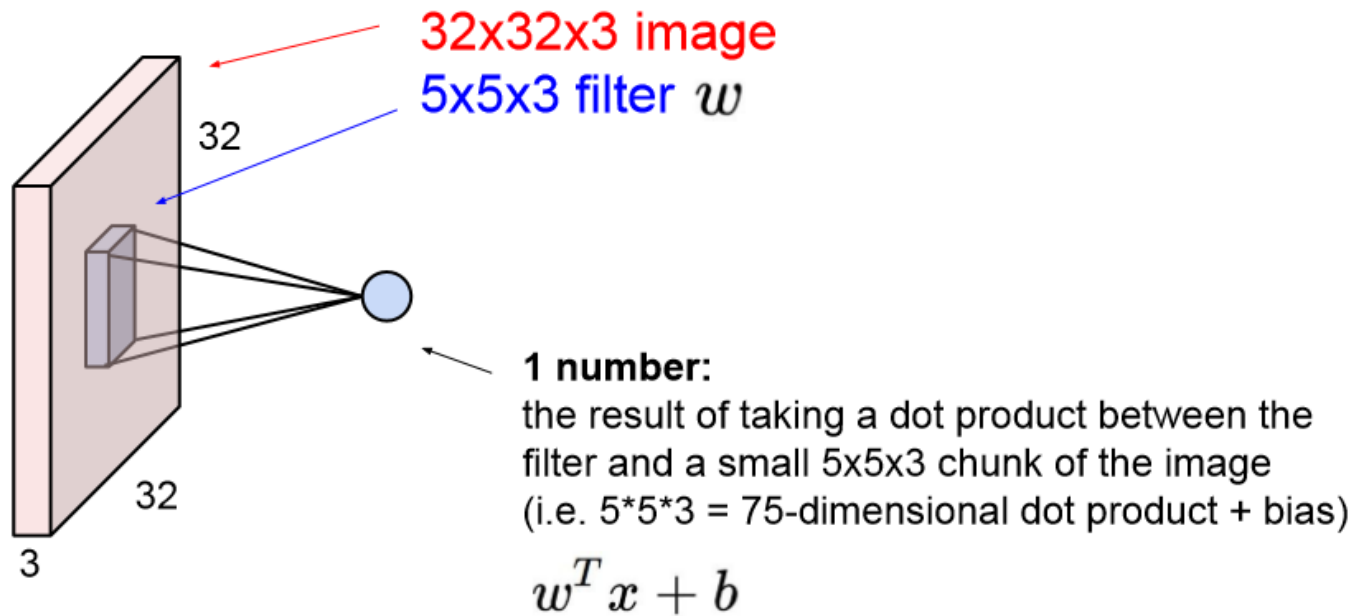
Convolutional Neural Network

- Convolution Layer



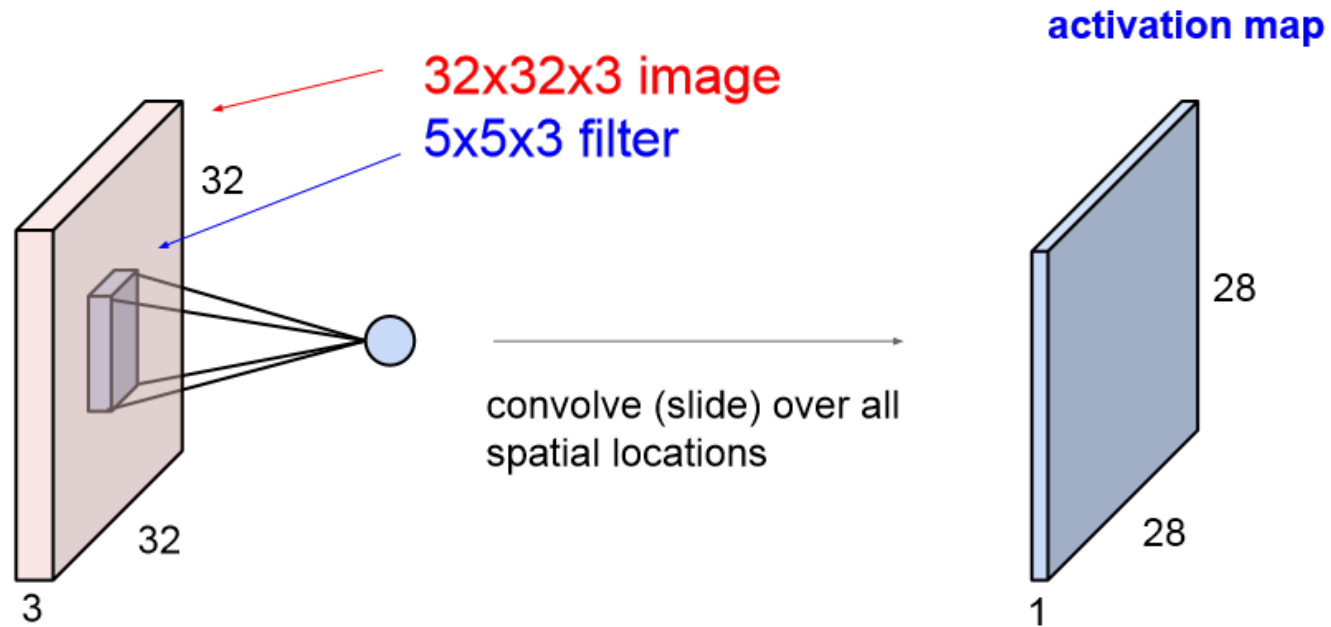
Convolutional Neural Network

- Convolution Layer



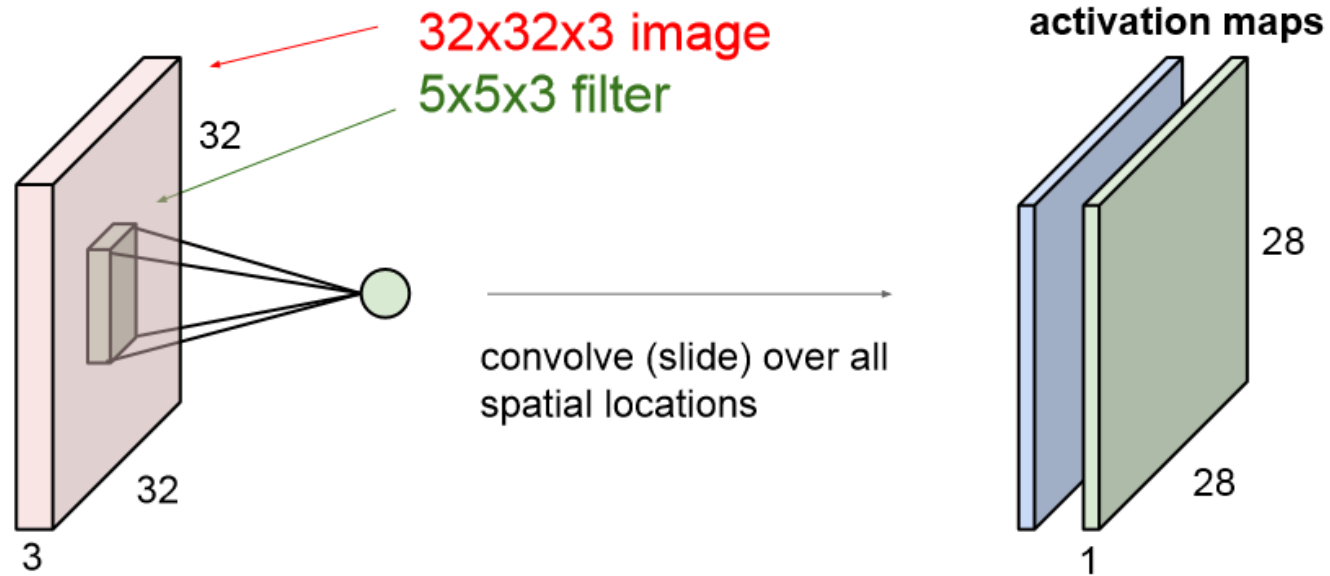
Convolutional Neural Network

- Convolution Layer



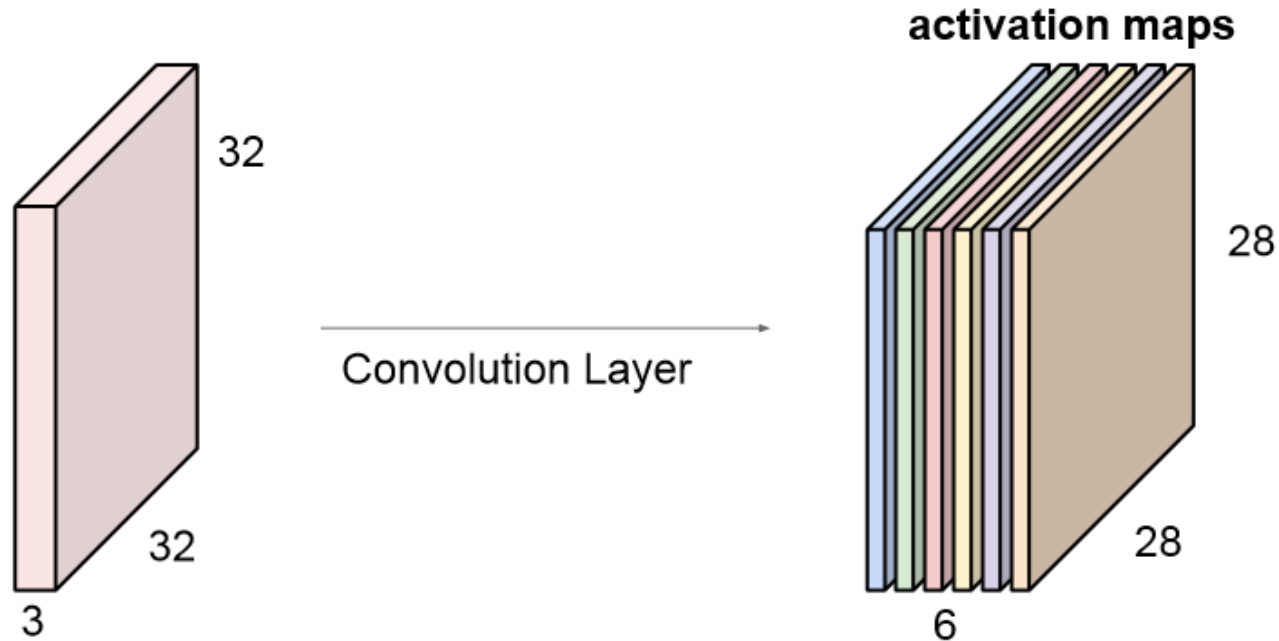
Convolutional Neural Network

- Convolution Layer



Convolutional Neural Network

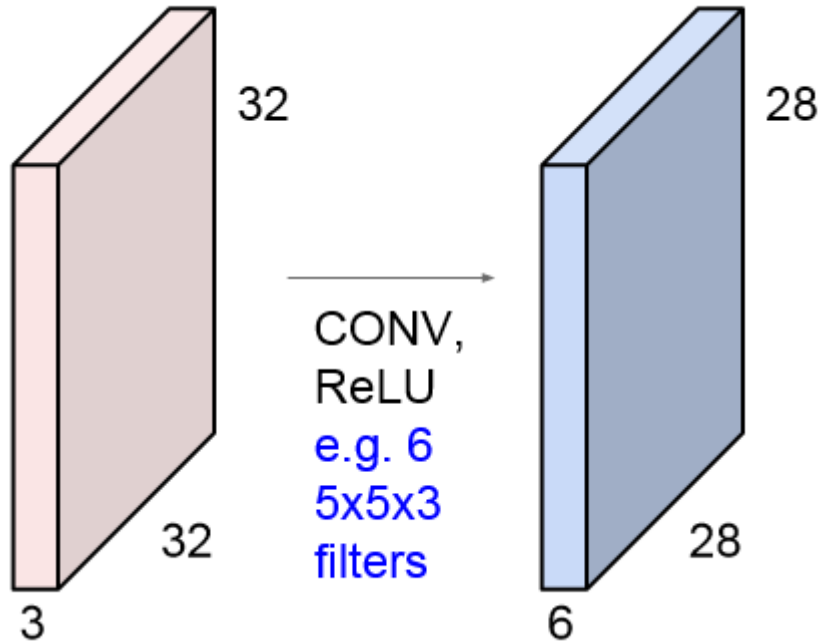
- Convolution Layer



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

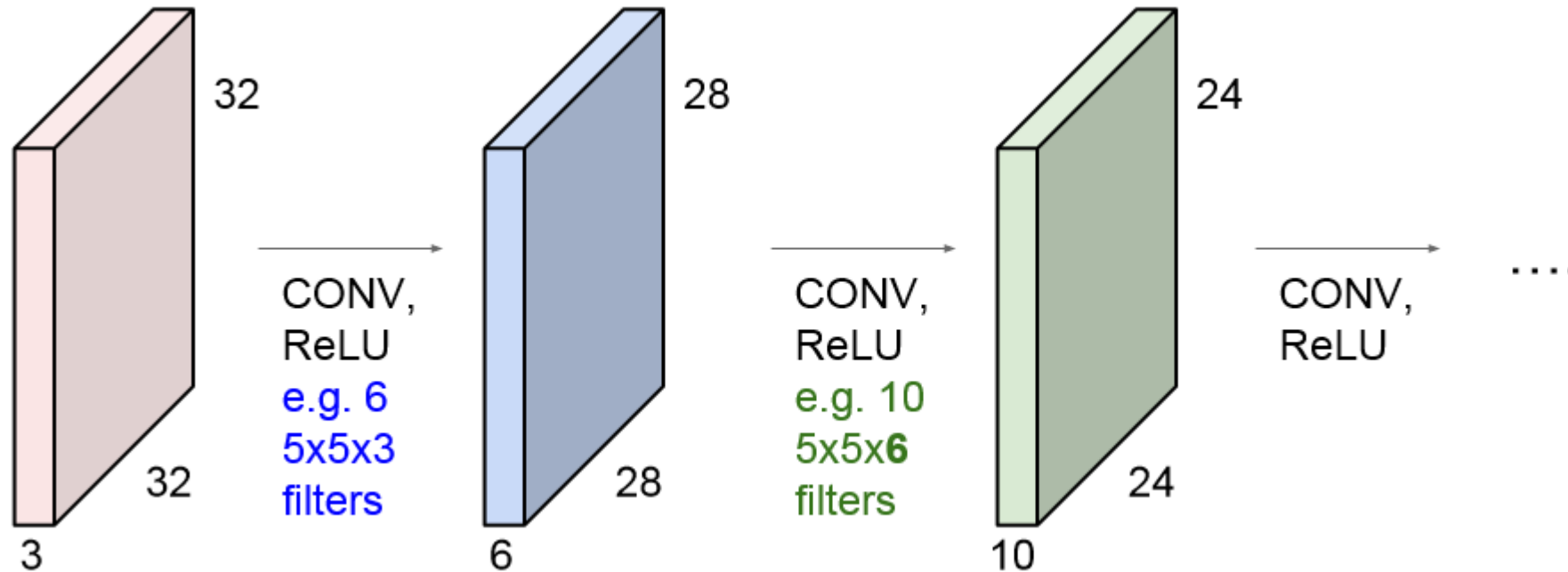
Convolutional Neural Network

- Convolution Layer



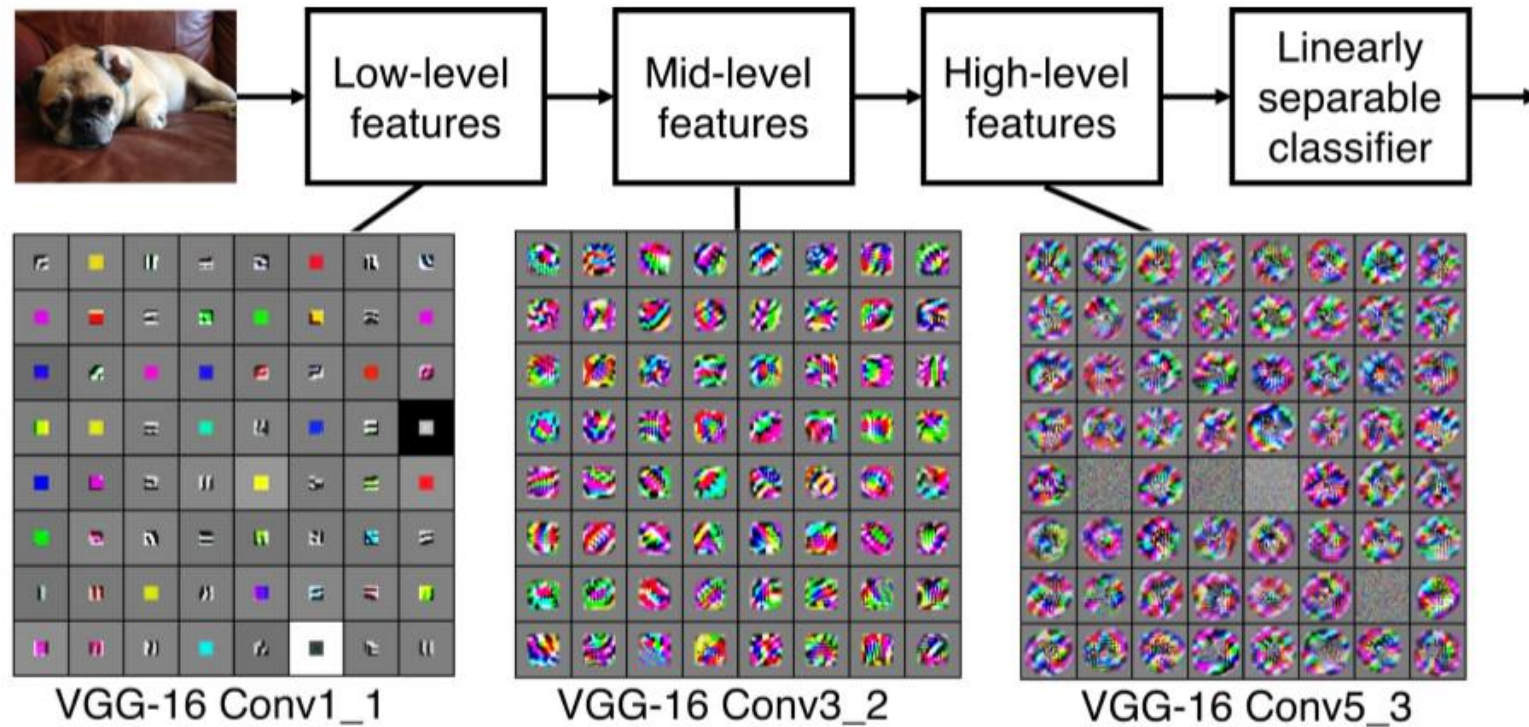
Convolutional Neural Network

- Convolution Layer



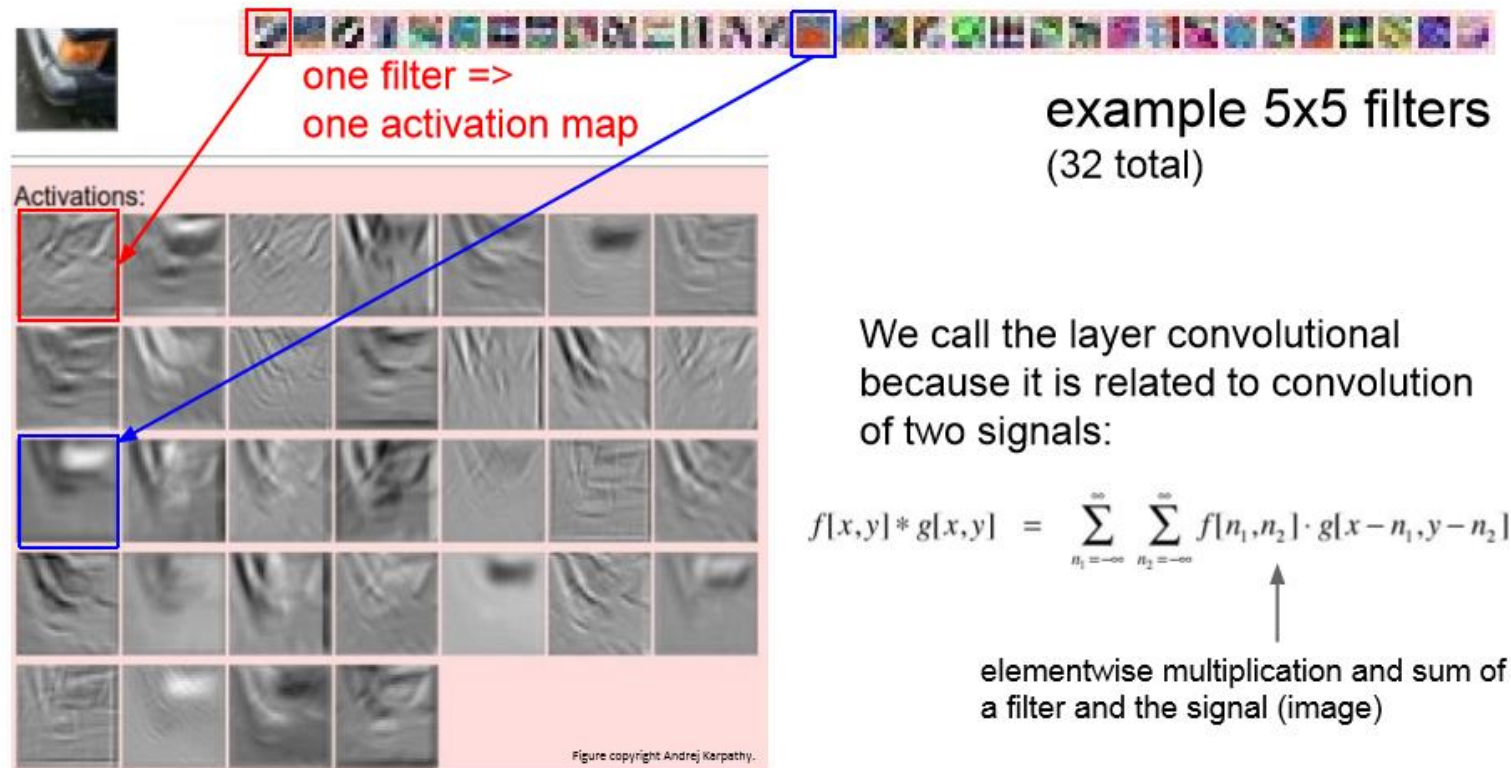
Convolutional Neural Network

- Feature-maps (Layers)



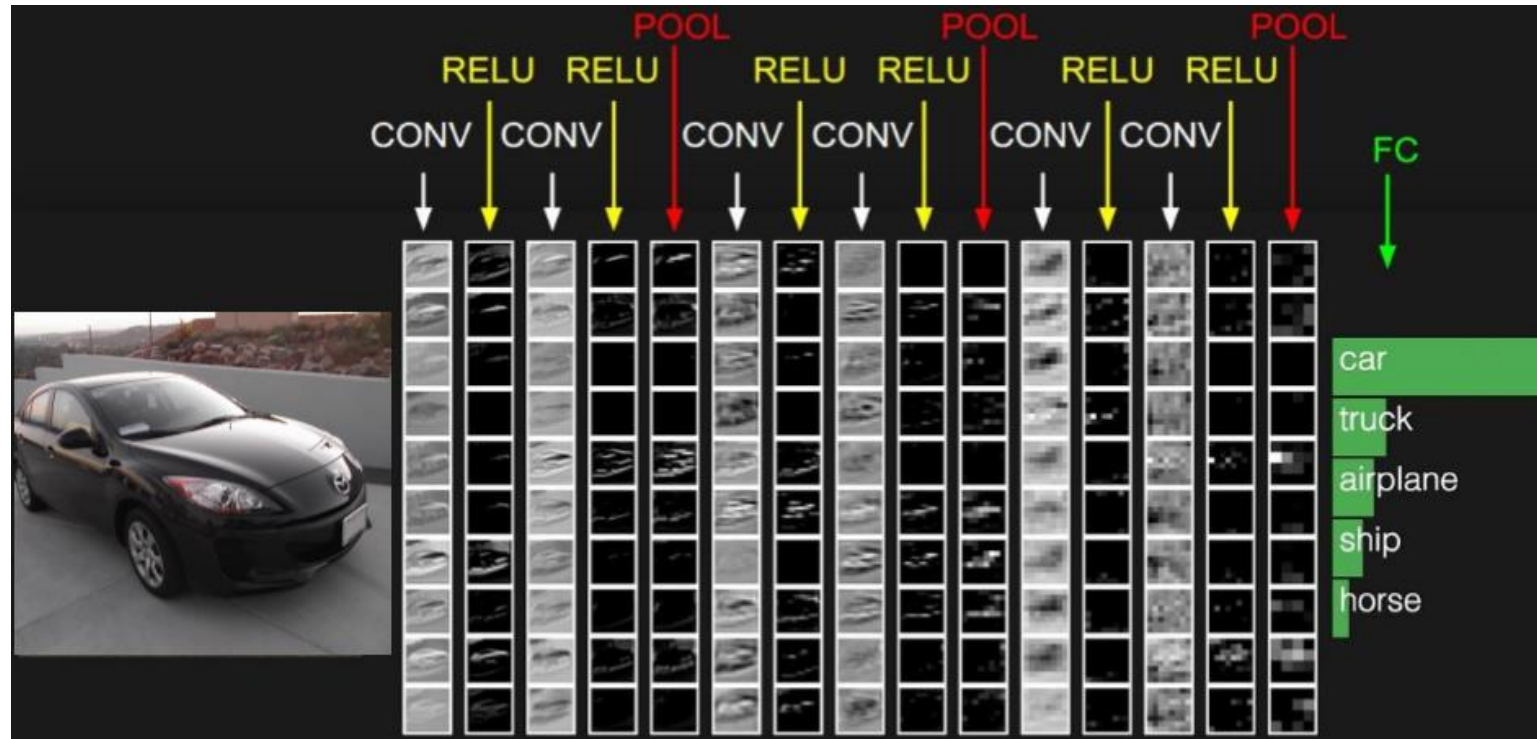
Convolutional Neural Network

- Feature-maps (Filters)



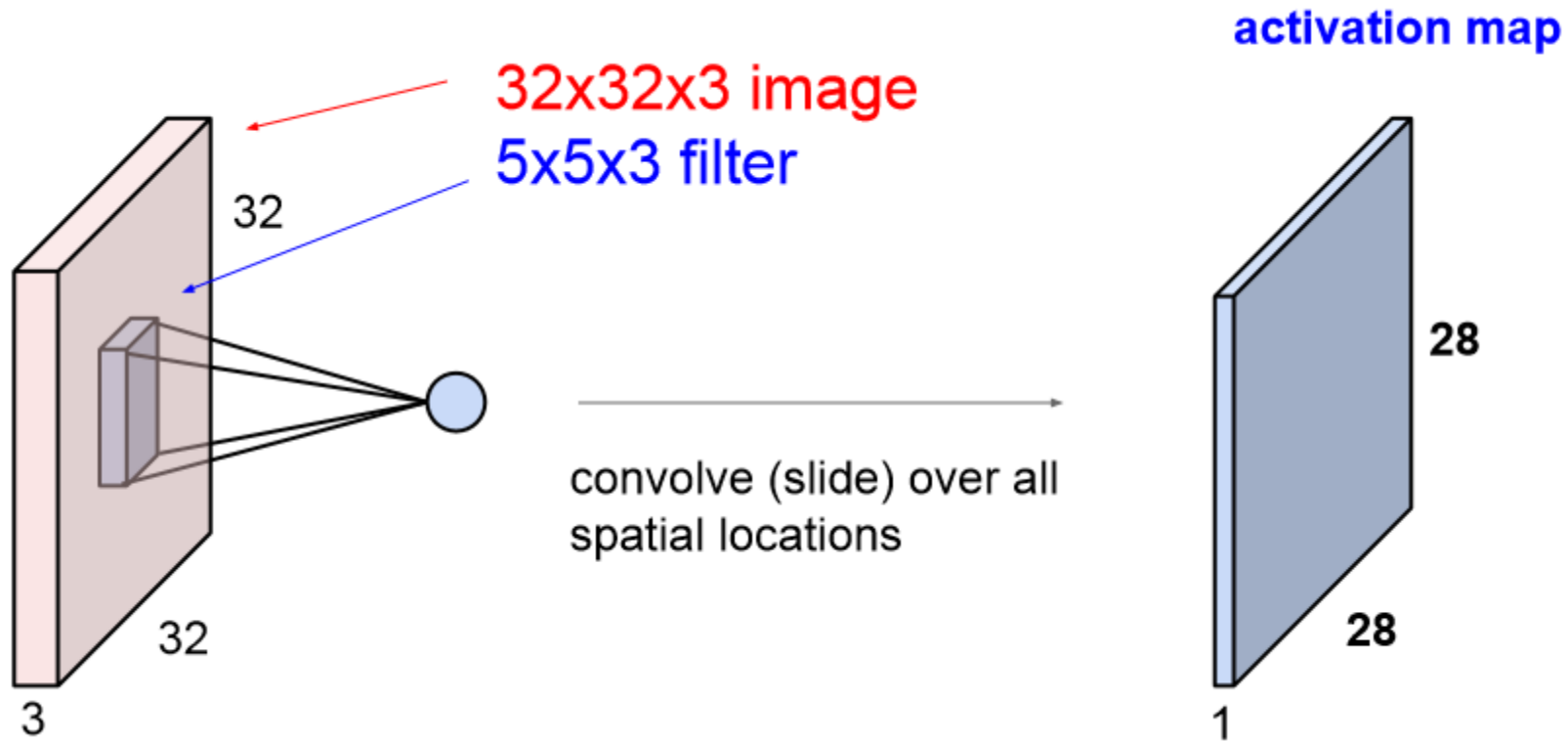
Convolutional Neural Network

- Feature-maps (All)



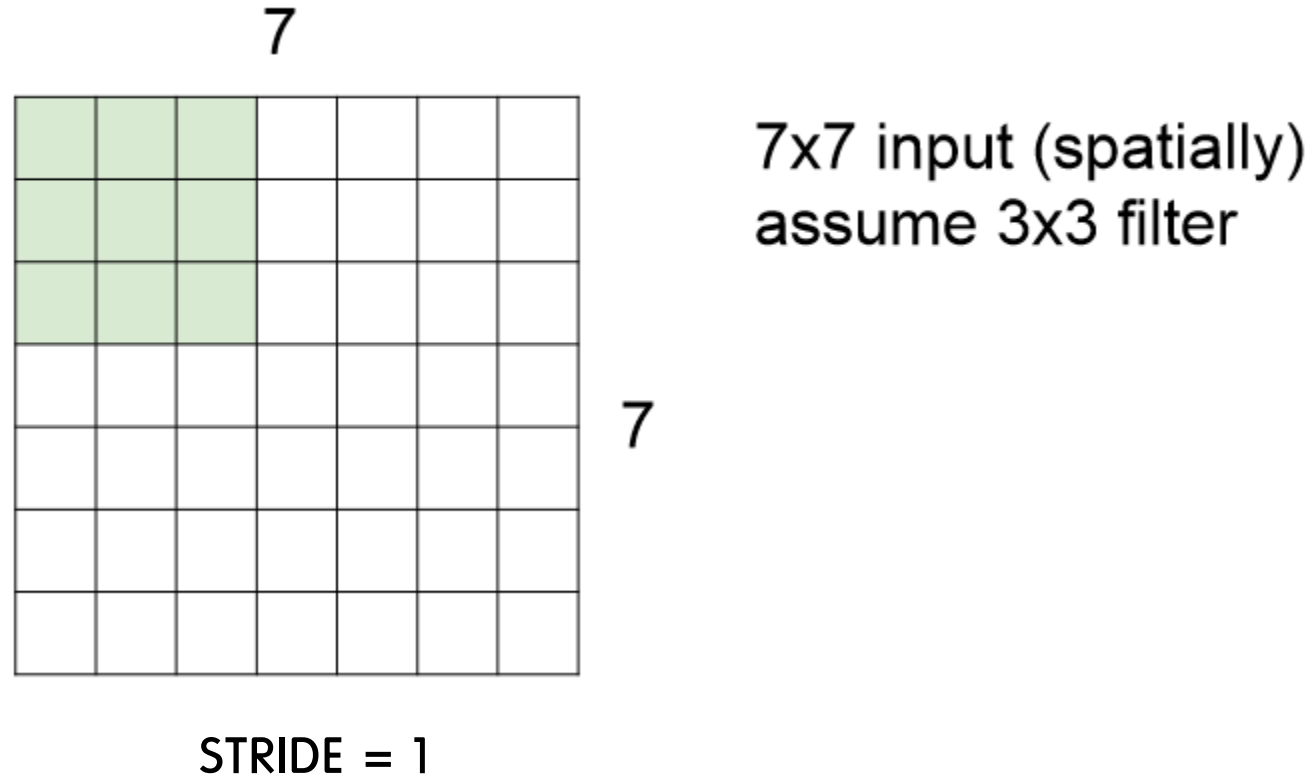
Convolutional Neural Network

- Convolution Layer



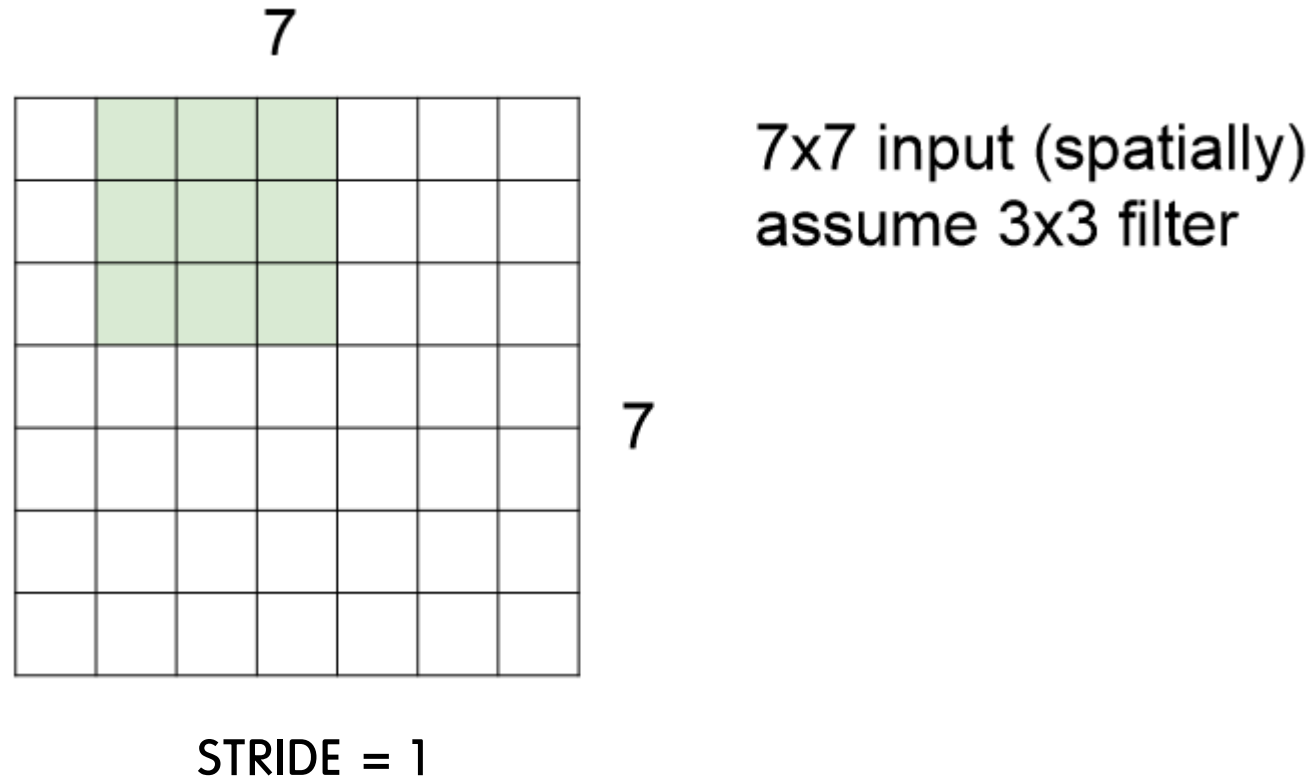
Convolutional Neural Network

- Convolution Layer



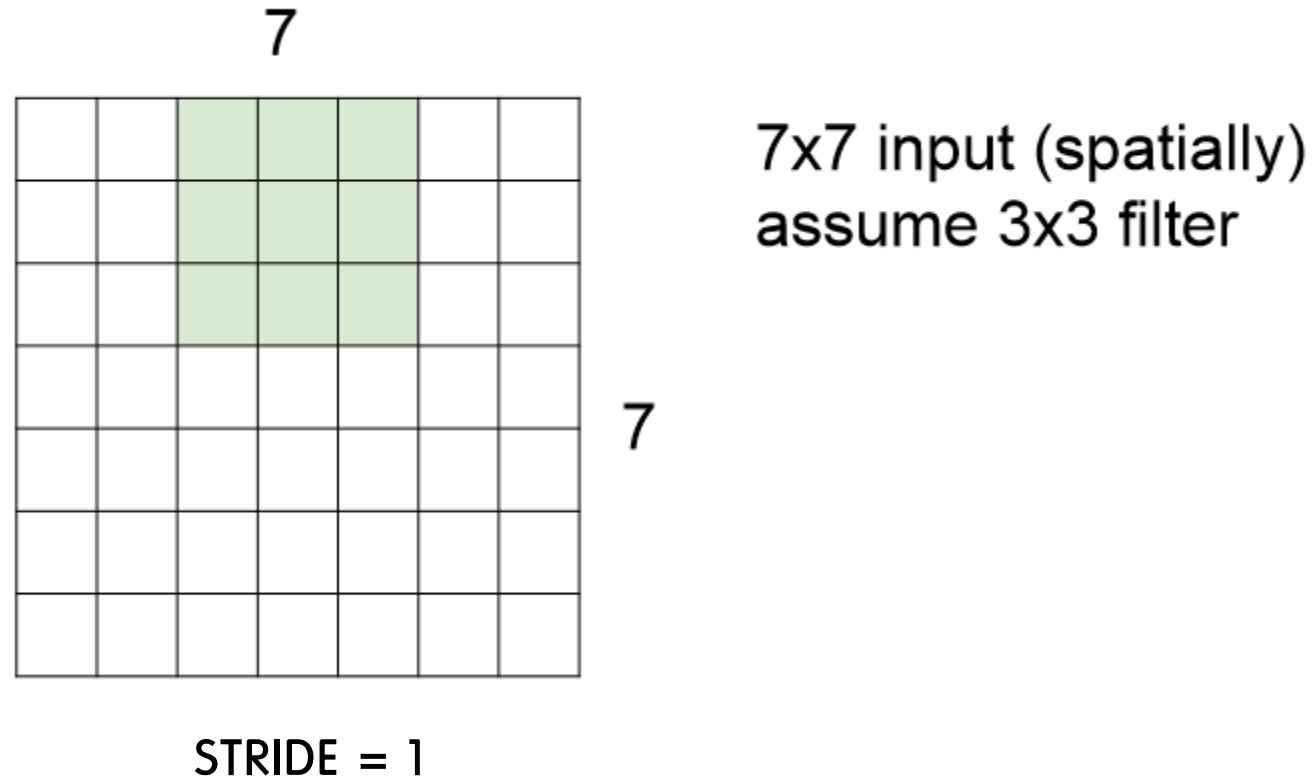
Convolutional Neural Network

- Convolution Layer



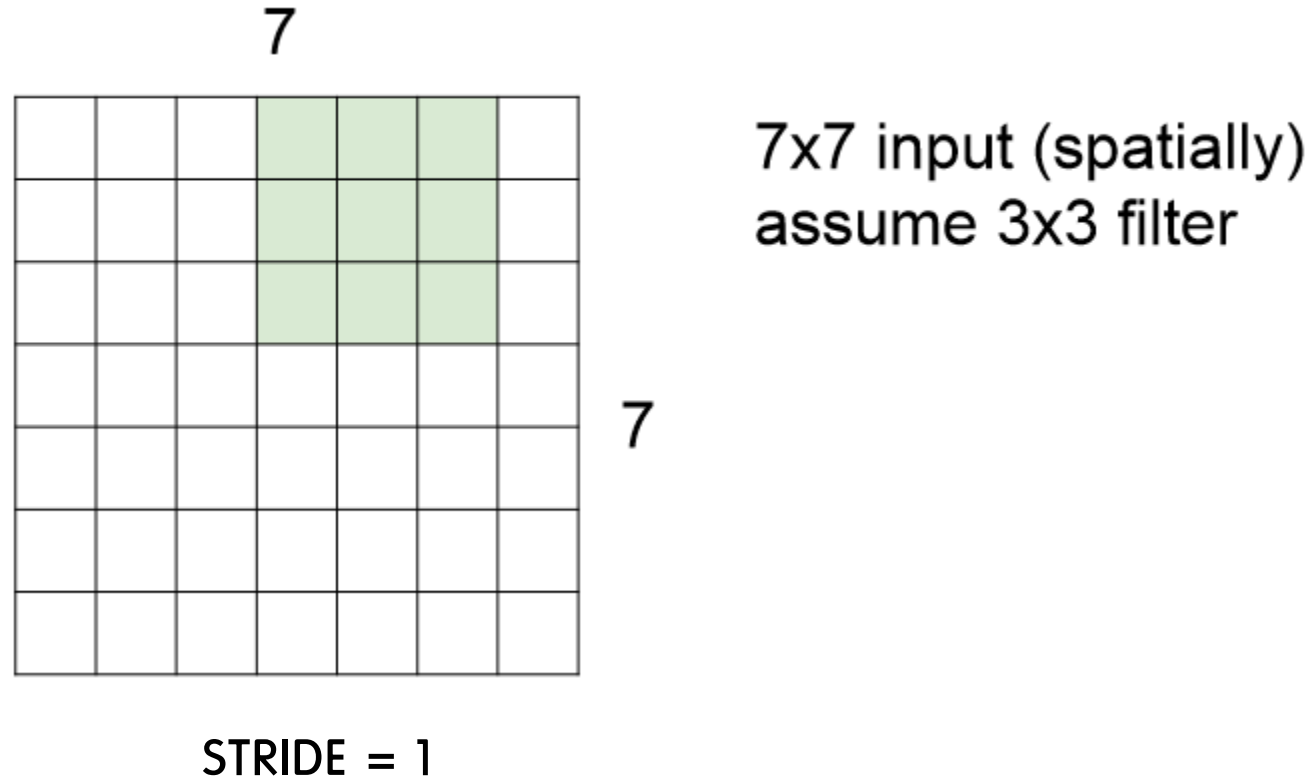
Convolutional Neural Network

- Convolution Layer



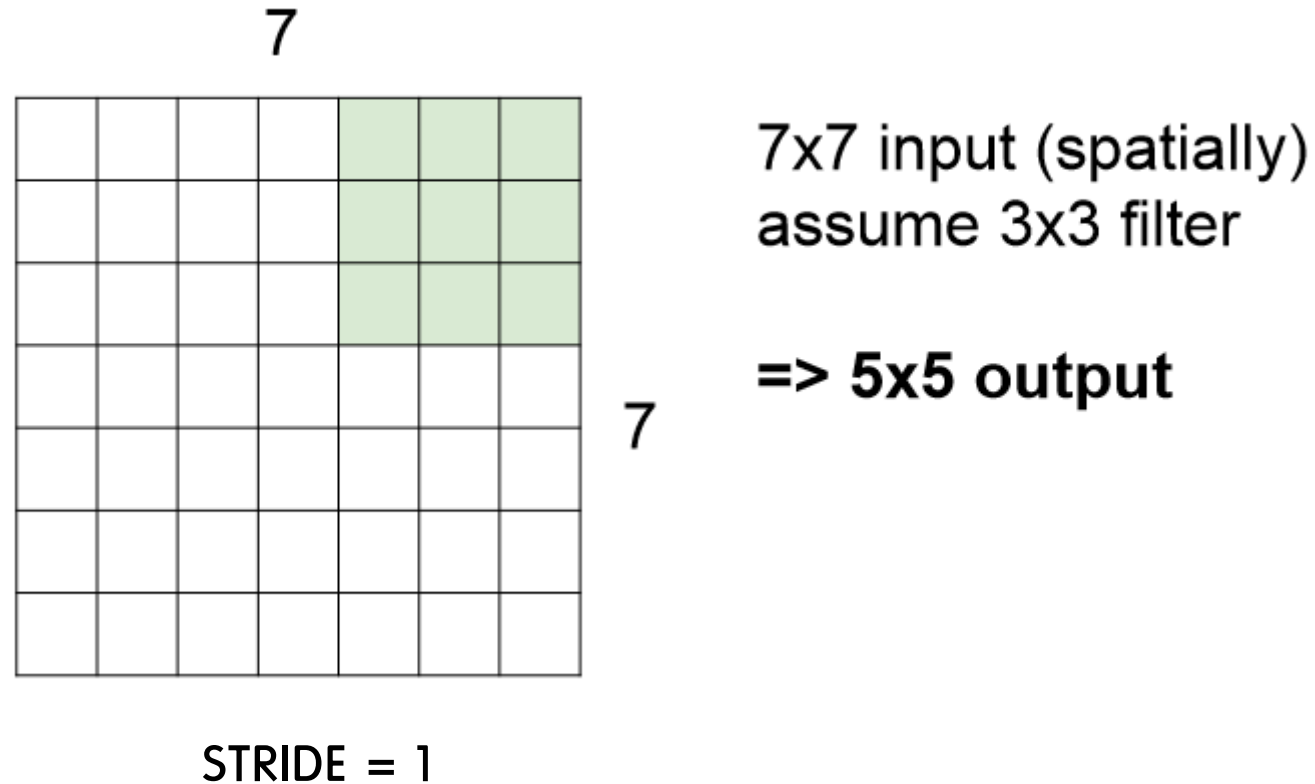
Convolutional Neural Network

- Convolution Layer



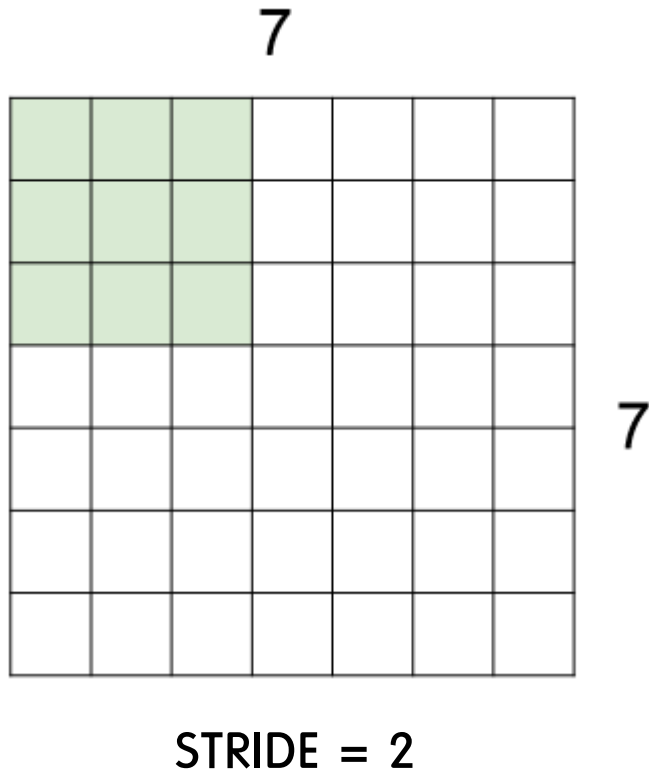
Convolutional Neural Network

- Convolution Layer



Convolutional Neural Network

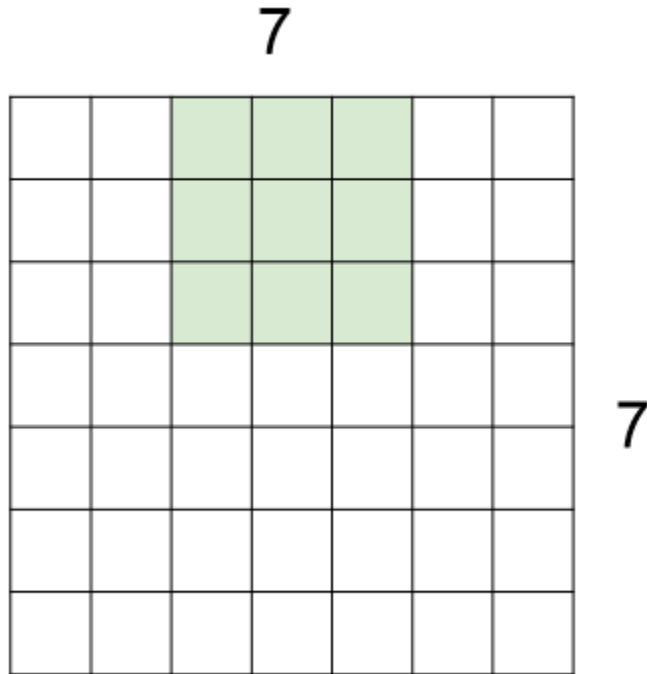
- Convolution Layer



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

Convolutional Neural Network

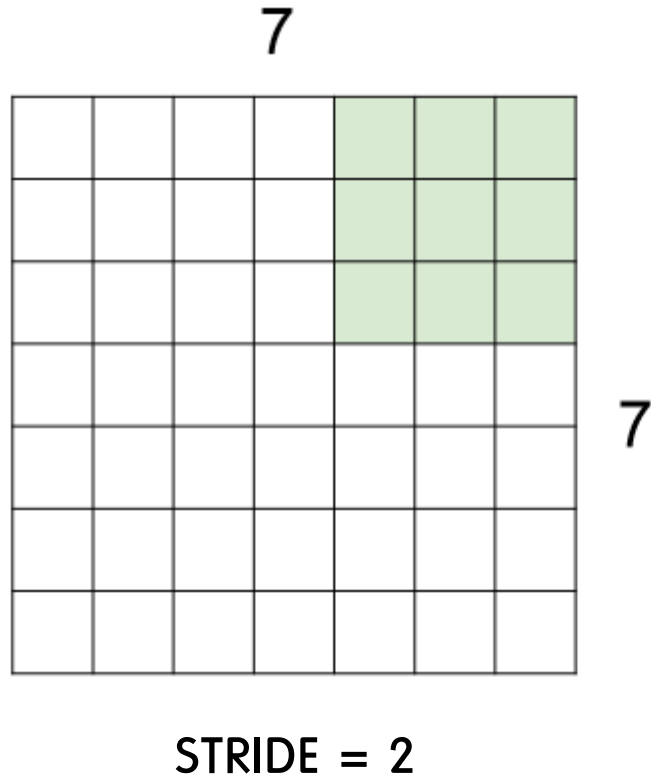
- Convolution Layer



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

Convolutional Neural Network

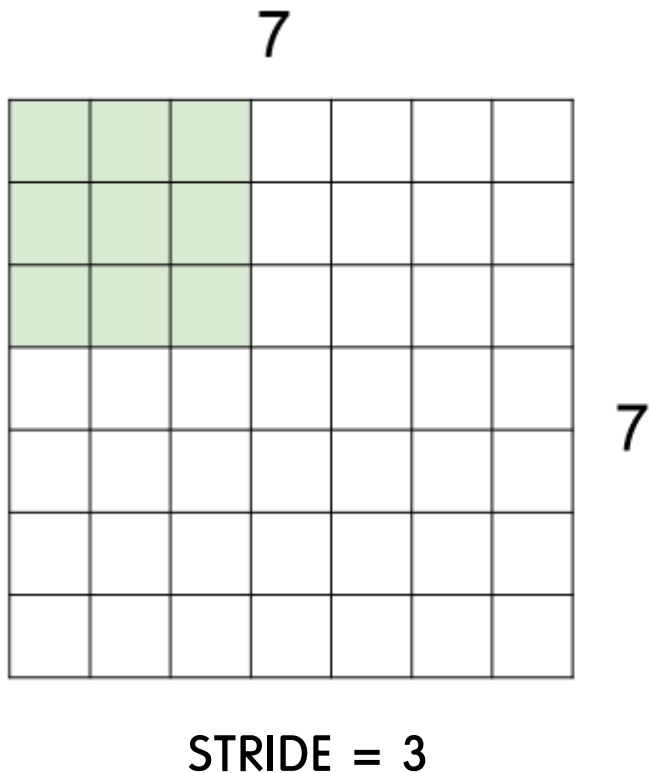
- Convolution Layer



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

Convolutional Neural Network

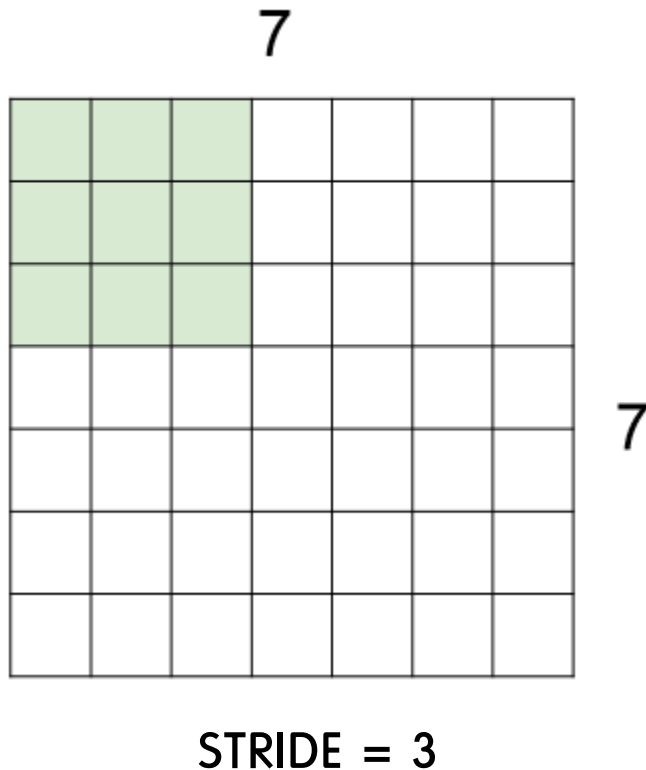
- Convolution Layer



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

Convolutional Neural Network

- Convolution Layer

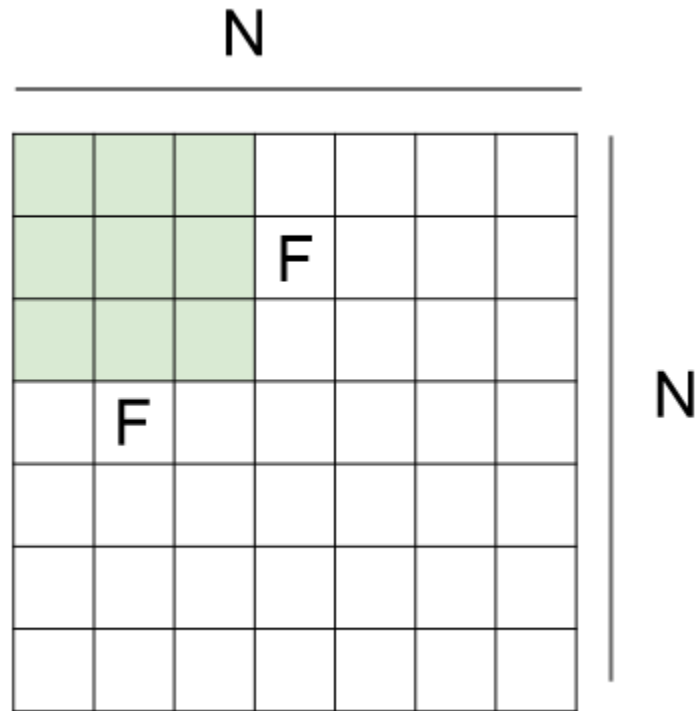


7x7 input (spatially)
assume 3x3 filter
applied **with stride 3?**

doesn't fit!
cannot apply 3x3 filter on
7x7 input with stride 3.

Convolutional Neural Network

- Convolution Layer



Output size:
 $(N - F) / \text{stride} + 1$

e.g. $N = 7, F = 3$:

stride 1 $\Rightarrow (7 - 3) / 1 + 1 = 5$

stride 2 $\Rightarrow (7 - 3) / 2 + 1 = 3$

stride 3 $\Rightarrow (7 - 3) / 3 + 1 = 2.33 \therefore \backslash$

Convolutional Neural Network

- Convolution Layer

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?

7x7 output!

in general, common to see CONV layers with stride 1, filters of size $F \times F$, and zero-padding with $(F-1)/2$. (will preserve size spatially)

e.g. $F = 3 \Rightarrow$ zero pad with 1

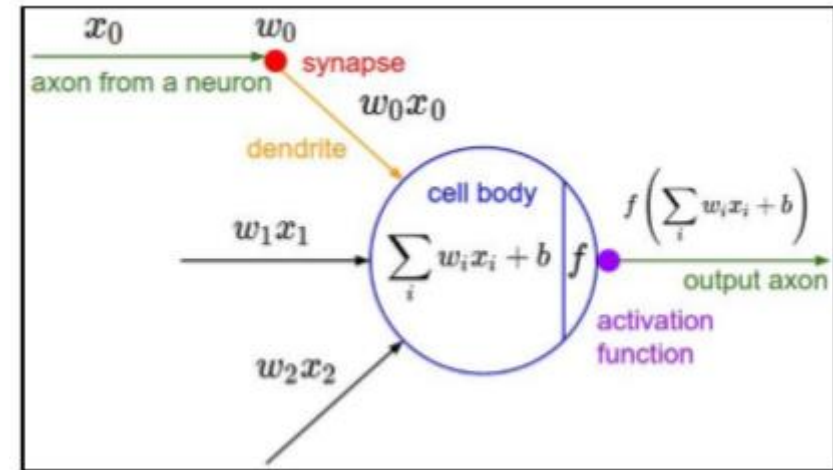
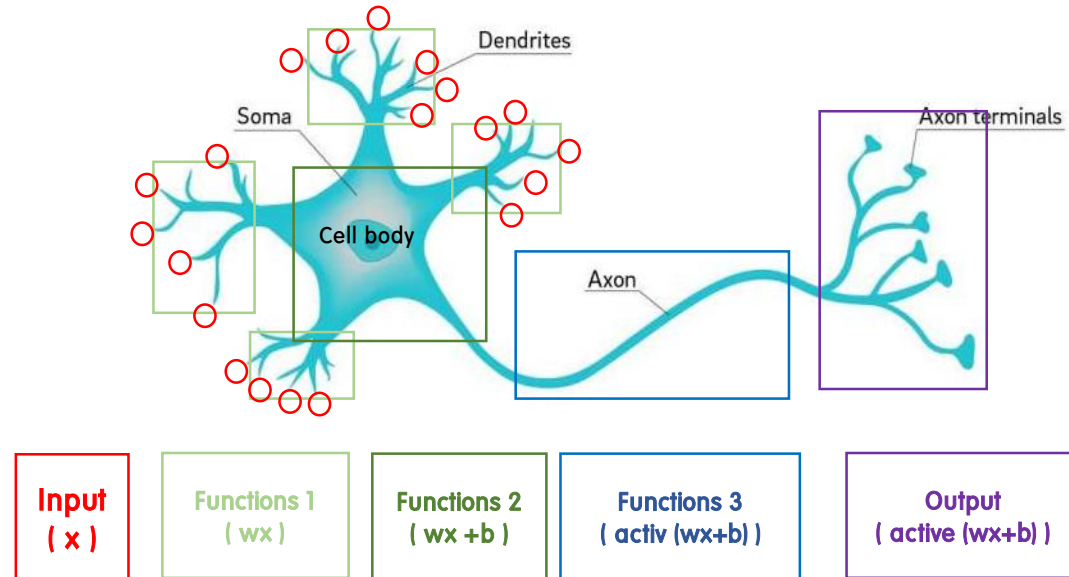
$F = 5 \Rightarrow$ zero pad with 2

$F = 7 \Rightarrow$ zero pad with 3

Output size:
 $(N - F) / \text{stride} + 1$

Convolutional Neural Network

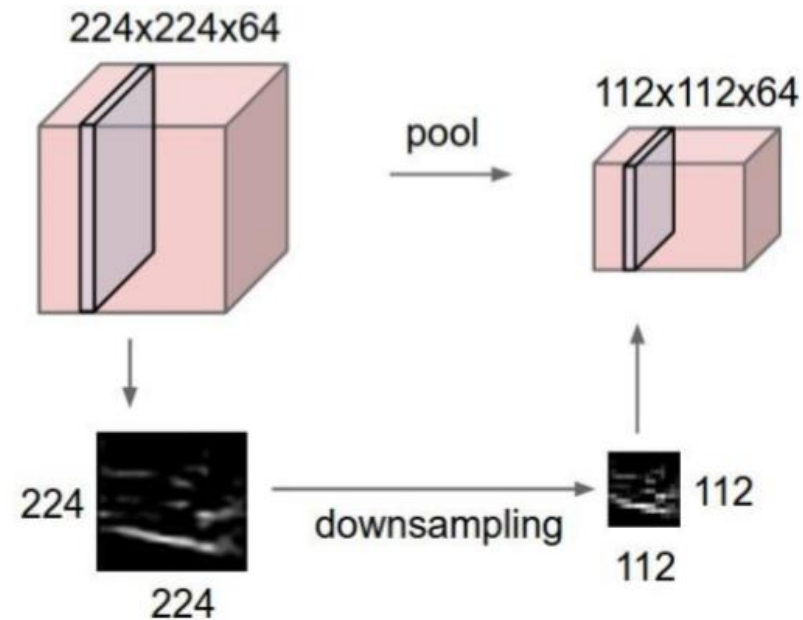
- Structure of a neuron



Convolutional Neural Network

- Pooling Layer

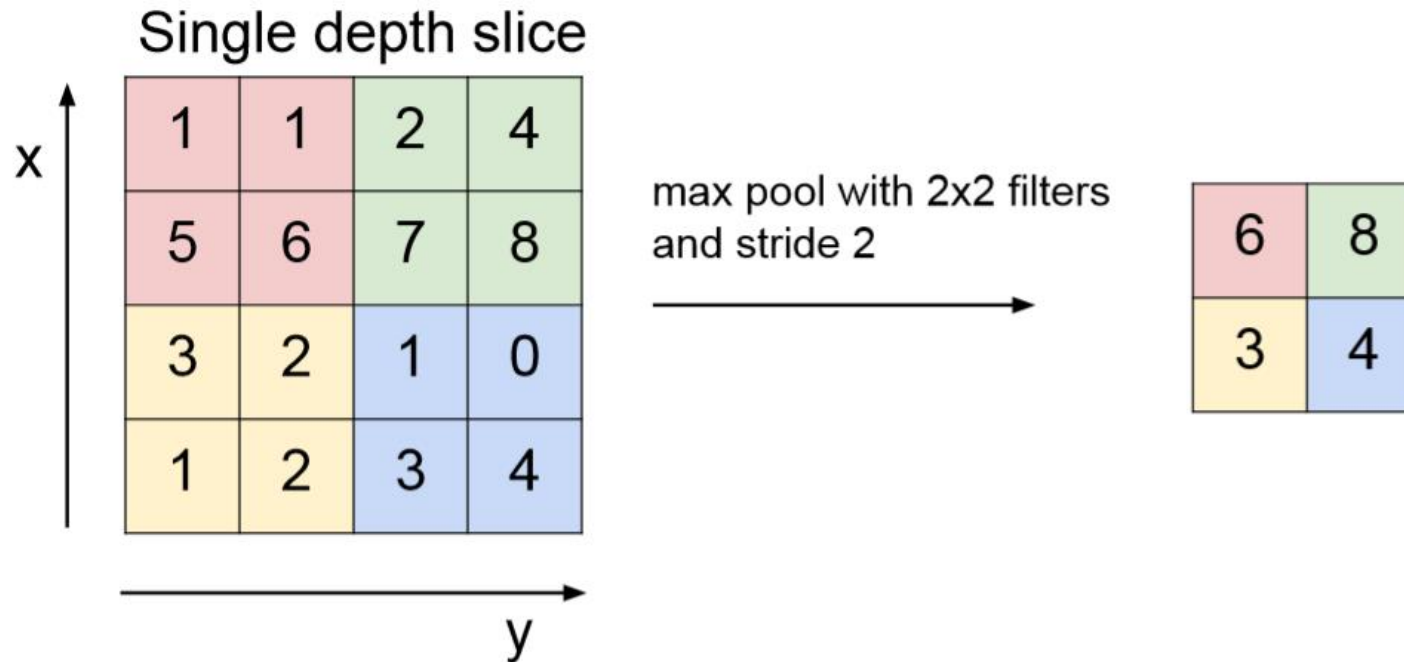
- makes the representations smaller and more manageable
- operates over each activation map independently:



Convolutional Neural Network

- Pooling Layer

MAX POOLING



Convolutional Neural Network

- Pooling Layer
 - Max Pooling
 - Average Pooling
 - ~~- Min Pooling~~

Convolutional Neural Network

- Summary

$[(\text{Conv-ReLU}) * N - \text{Pool}] * M - (\text{FC-ReLU}) * K - \text{Softmax}$

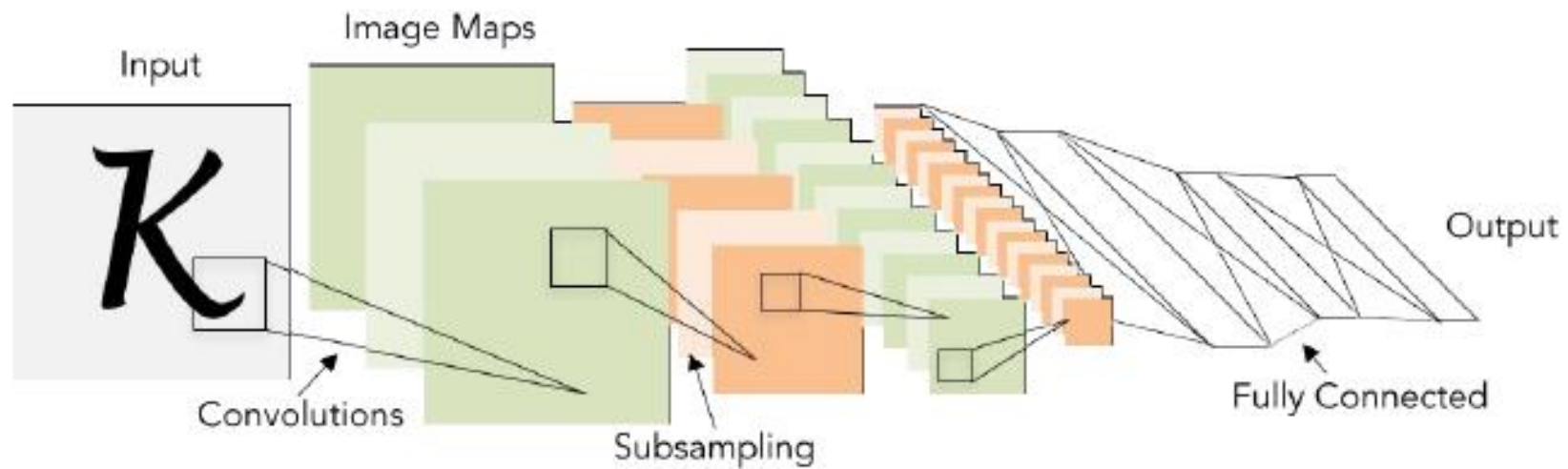
Convolutional Neural Network

- Summary

$[(\text{Conv-ReLU}) * 2 - \text{Pool}] * 3 - (\text{FC-ReLU}) * 2 - \text{Softmax}$

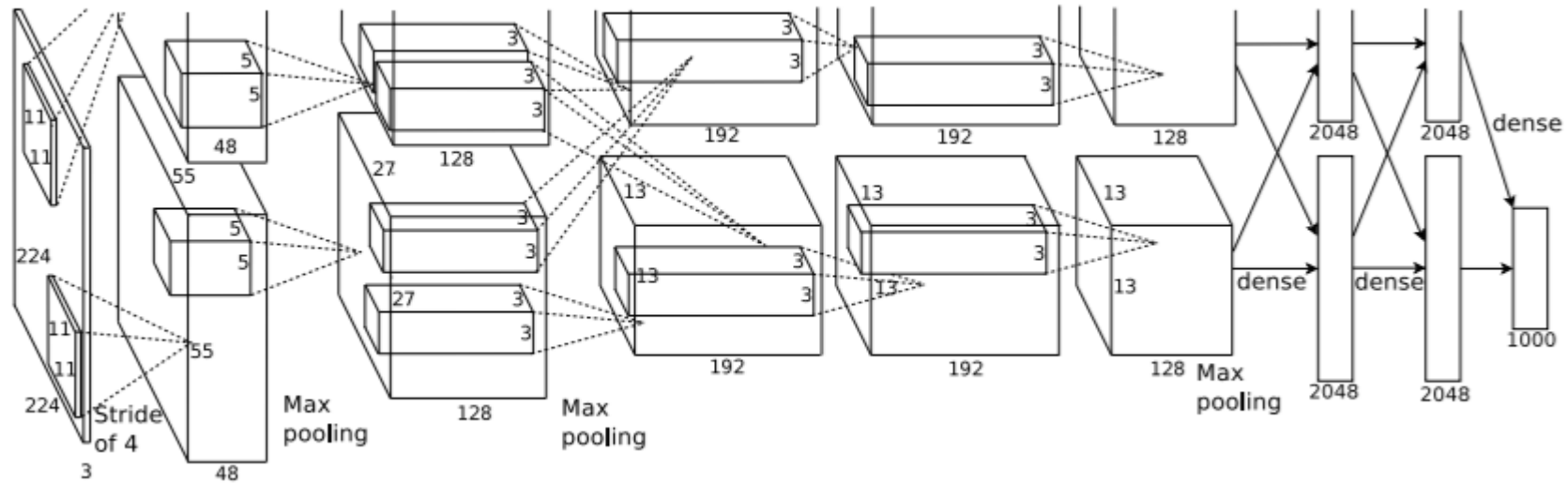
CNN History

- 1) LeNet (1998)



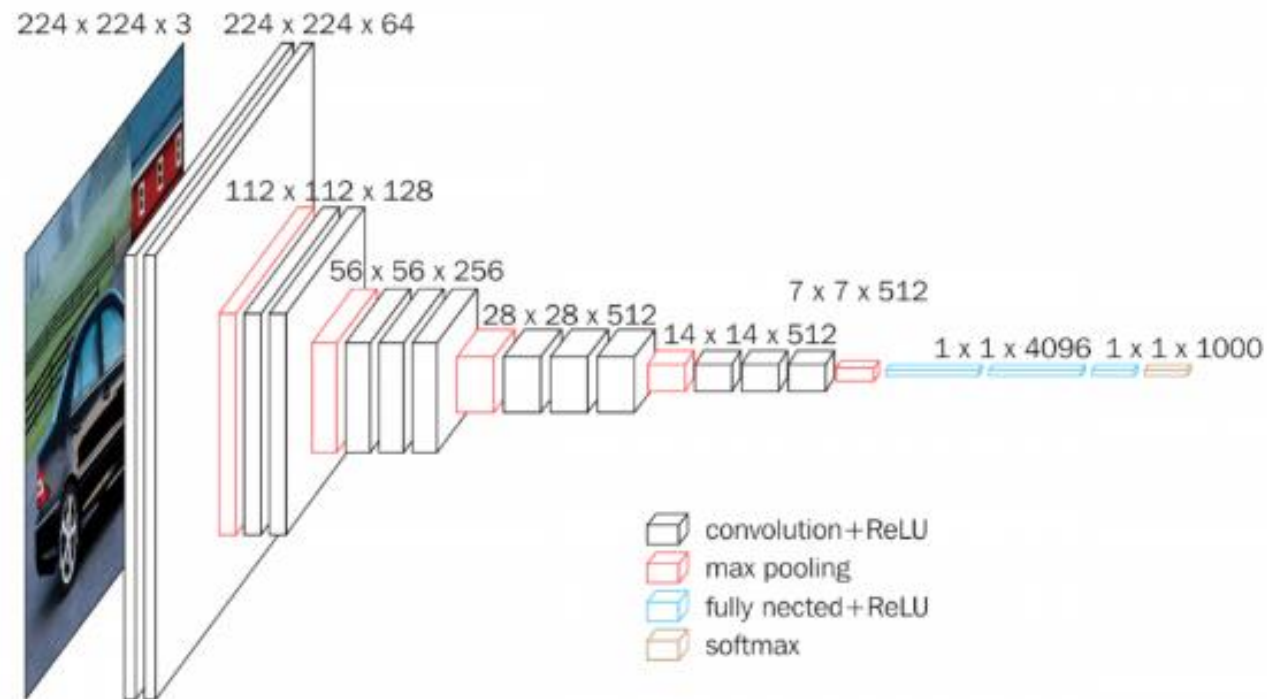
CNN History

- 2) AlexNet (2012)



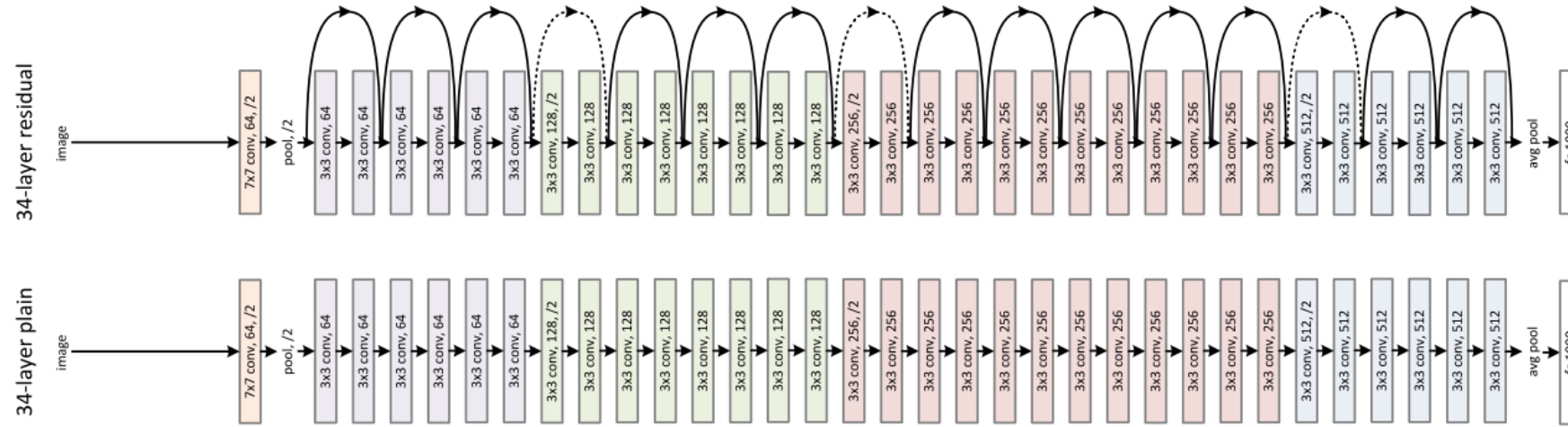
CNN History

- 3) VGG (2014)



CNN History

- 4) ResNet (2015)



CNN History

- 5) DenseNet (2016)

