Idea Factory Intensive Program #2

딥러닝 홀로서기

#20

이론강의/PyTorch실습/코드리뷰

딥러닝(Deep Learning)에 관심이 있는 학생 발굴을 통한
딥러닝의 이론적 배경 강의 및 오픈소스 딥러닝 라이브러리 PyTorch를 활용한 실습

Topics to learn today

1. **Review from last lecture**
   - Assignment: CIFAR-10 classification with MLP
   - Lecture: Advanced Gradient Descent Algorithms
2. **Problems of MLP**
3. **What is Convolutional Neural Network?**
   - Convolutional Layer, Pooling Layer
4. **Implementing CNN with Pytorch**

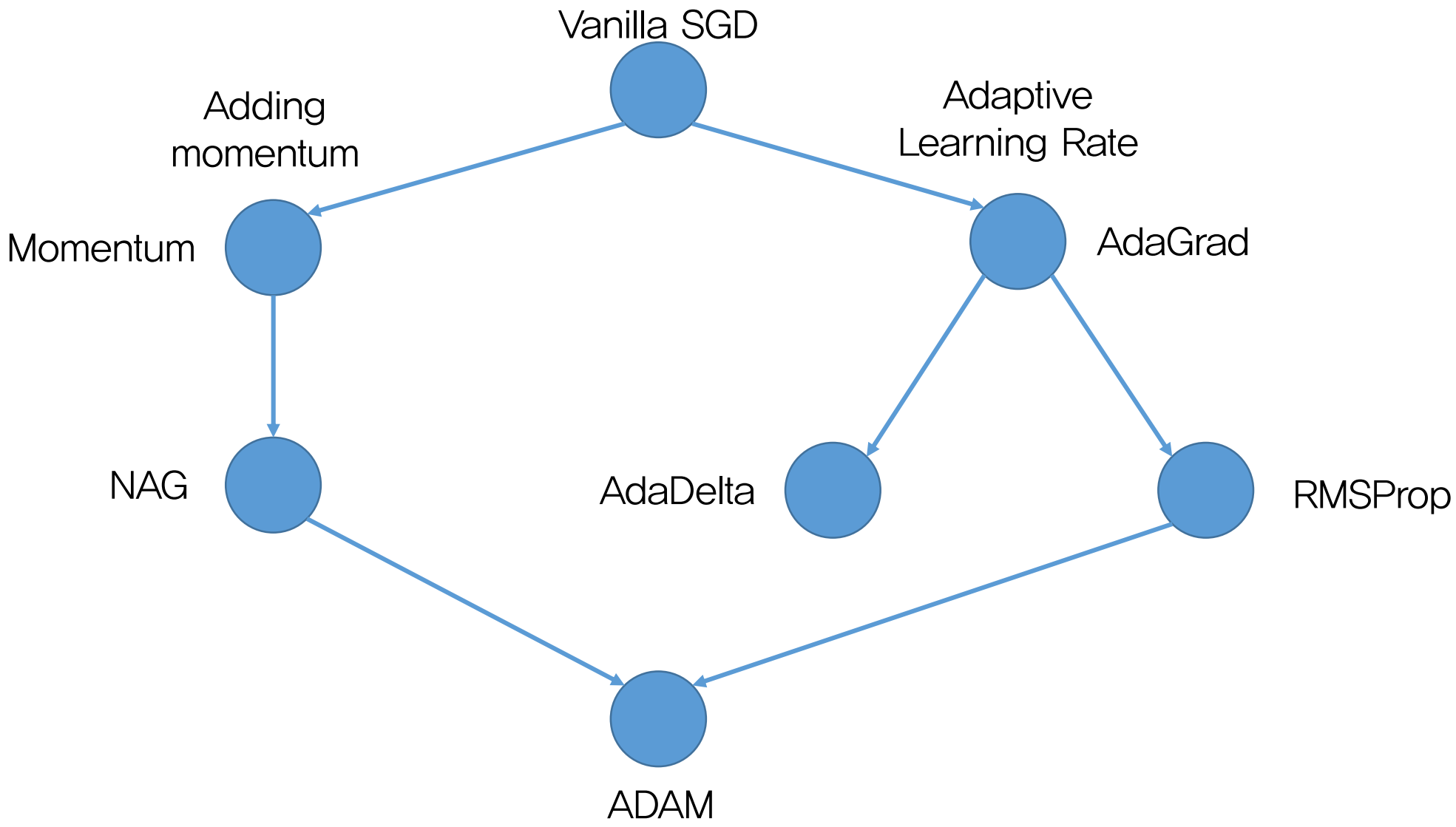
Review from Last Lecture

Batch / Stochastic Gradient Descent

$$\theta = \theta - \eta \nabla J(\theta)$$

θ : Parameter set of the model η : Learning rate $J(\theta)$: Loss function

Review from Last Lecture



Problems of MLP

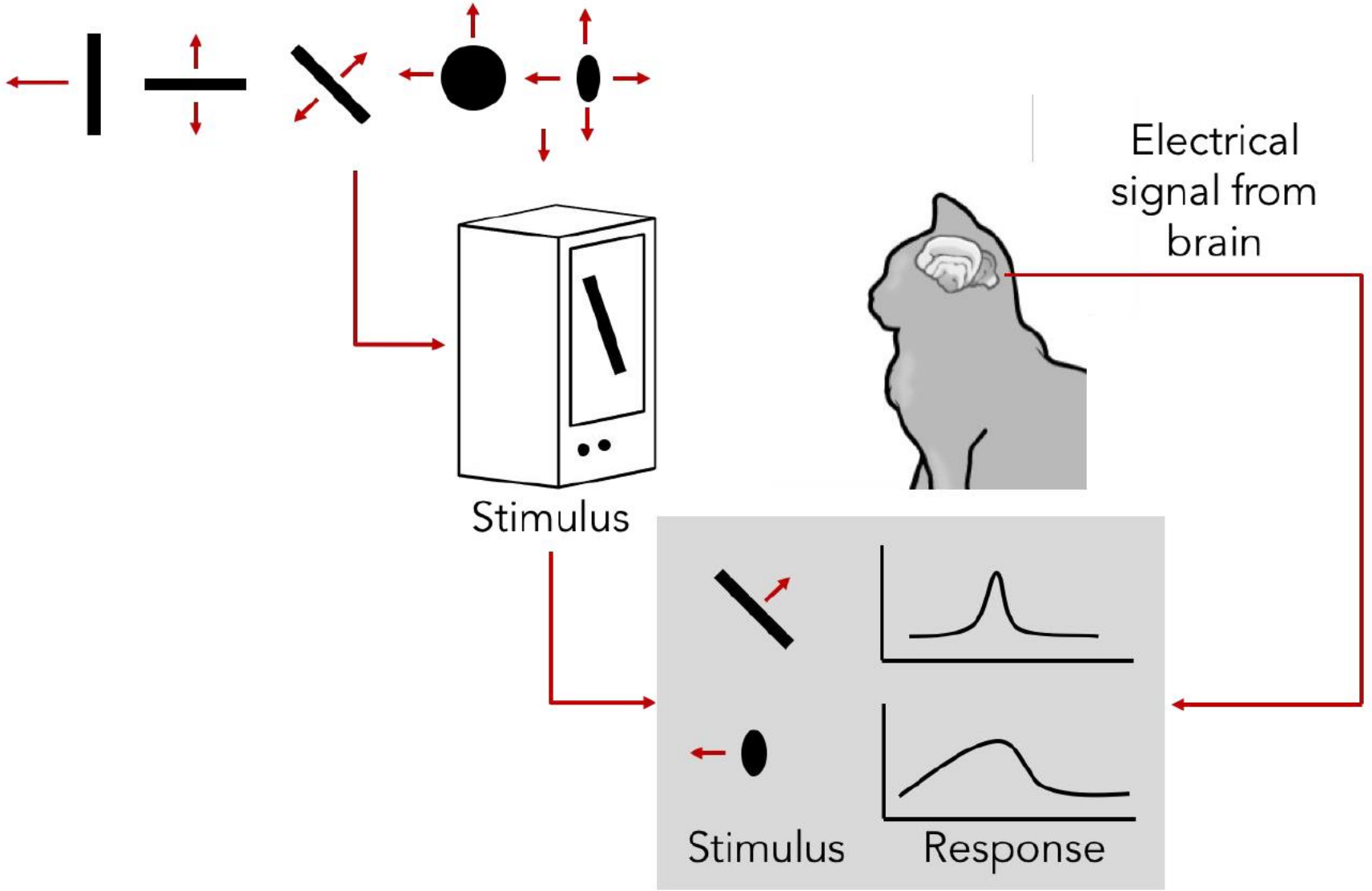
Number of parameters

Since a neuron is connected with every neurons in preceding layer, number of parameters explodes as model gets deeper.

Some of the parameters are meaningless.

What is Convolutional Neural Network?

How human recognize an image?



Hierarchical organization

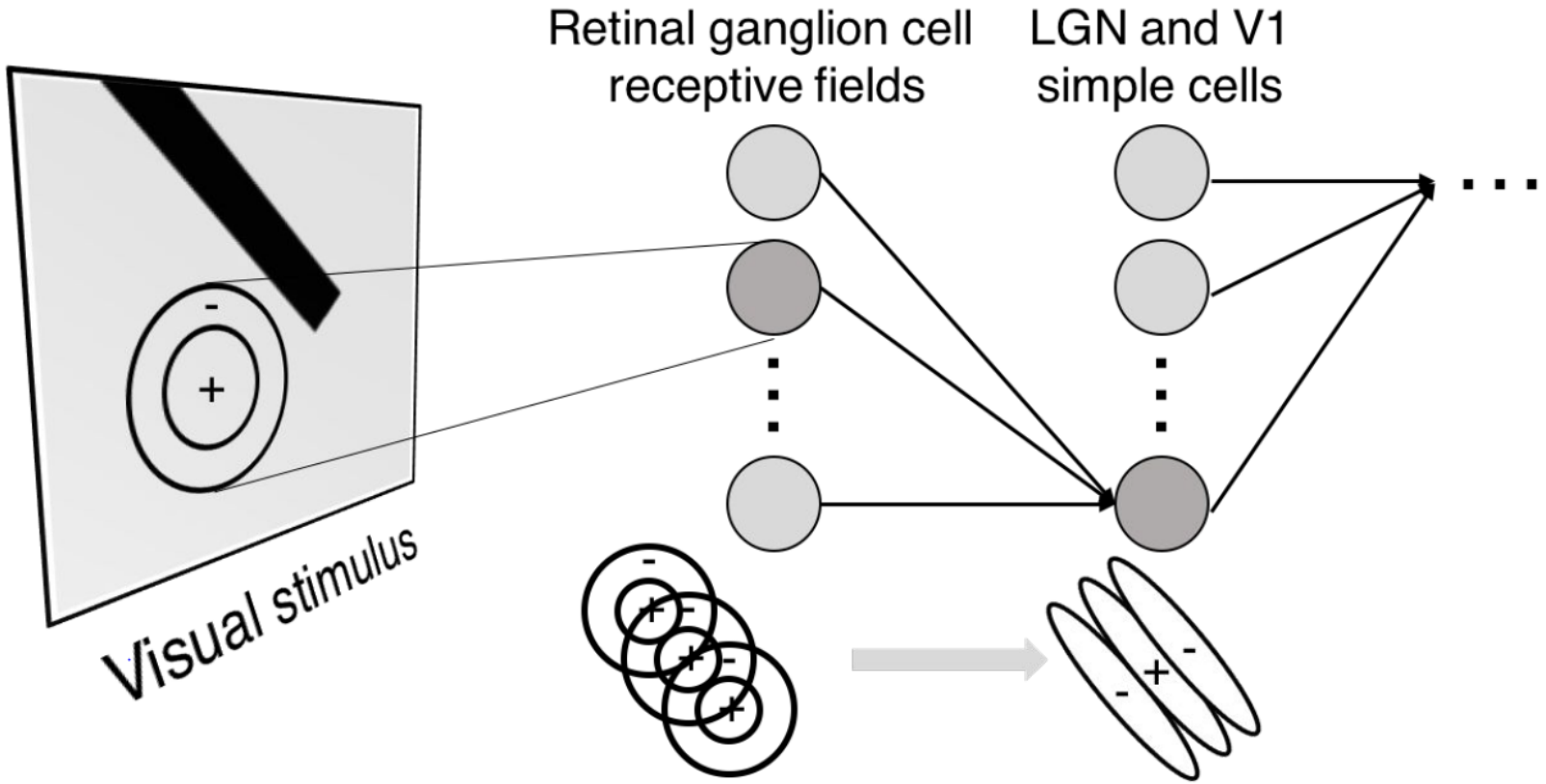
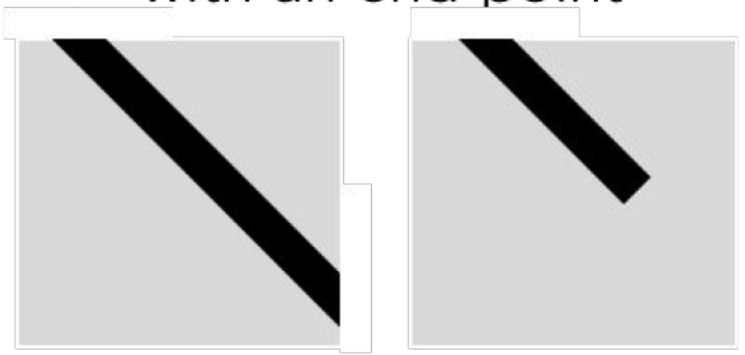


Illustration of hierarchical organization in early visual pathways by Lane McIntosh, copyright CS231n 2017

Simple cells:
Response to light orientation

Complex cells:
Response to light orientation and movement

Hypercomplex cells:
response to movement with an end point

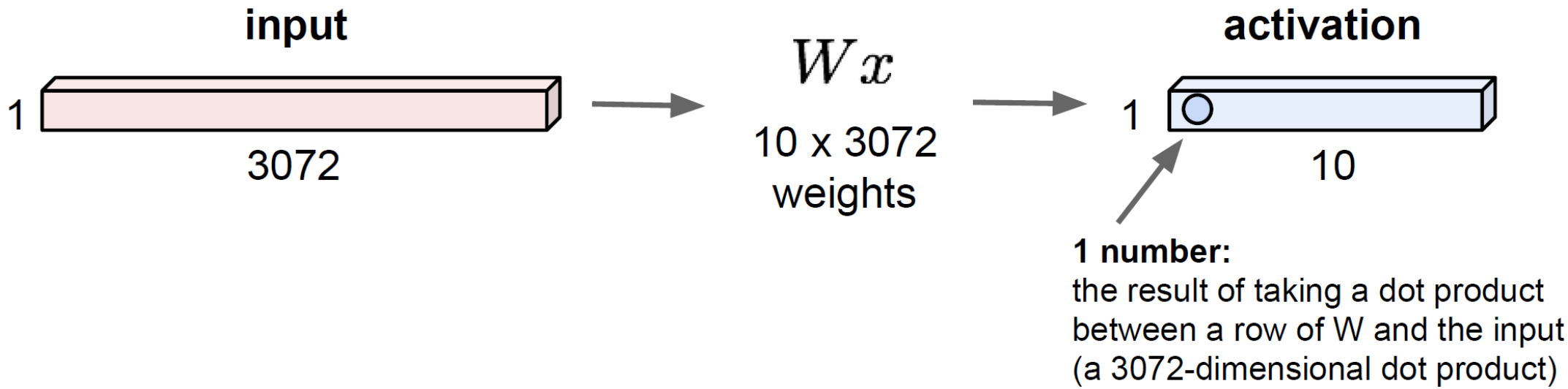


No response

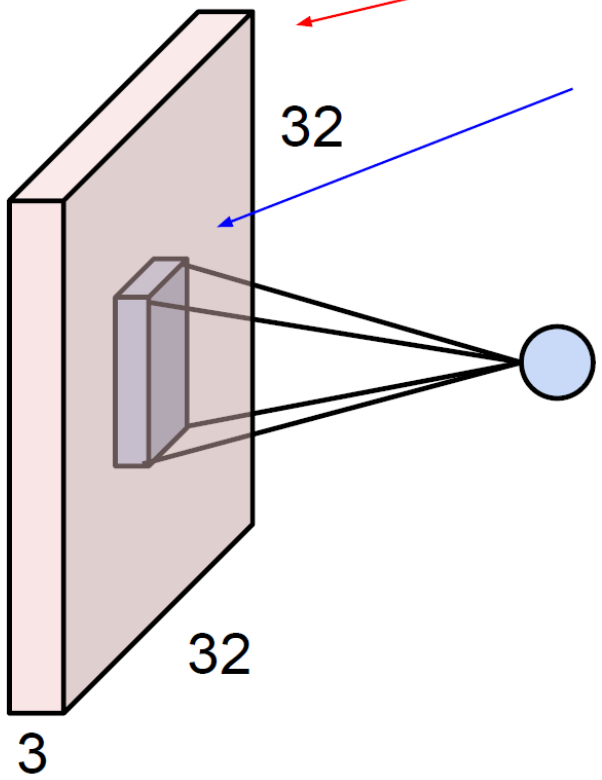
Response (end point)

MLP / Fully Connected Layer

32x32x3 image -> stretch to 3072 x 1



Convolution Layer
: Preserve the spatial structure



32x32x3 image

5x5x3 filter w

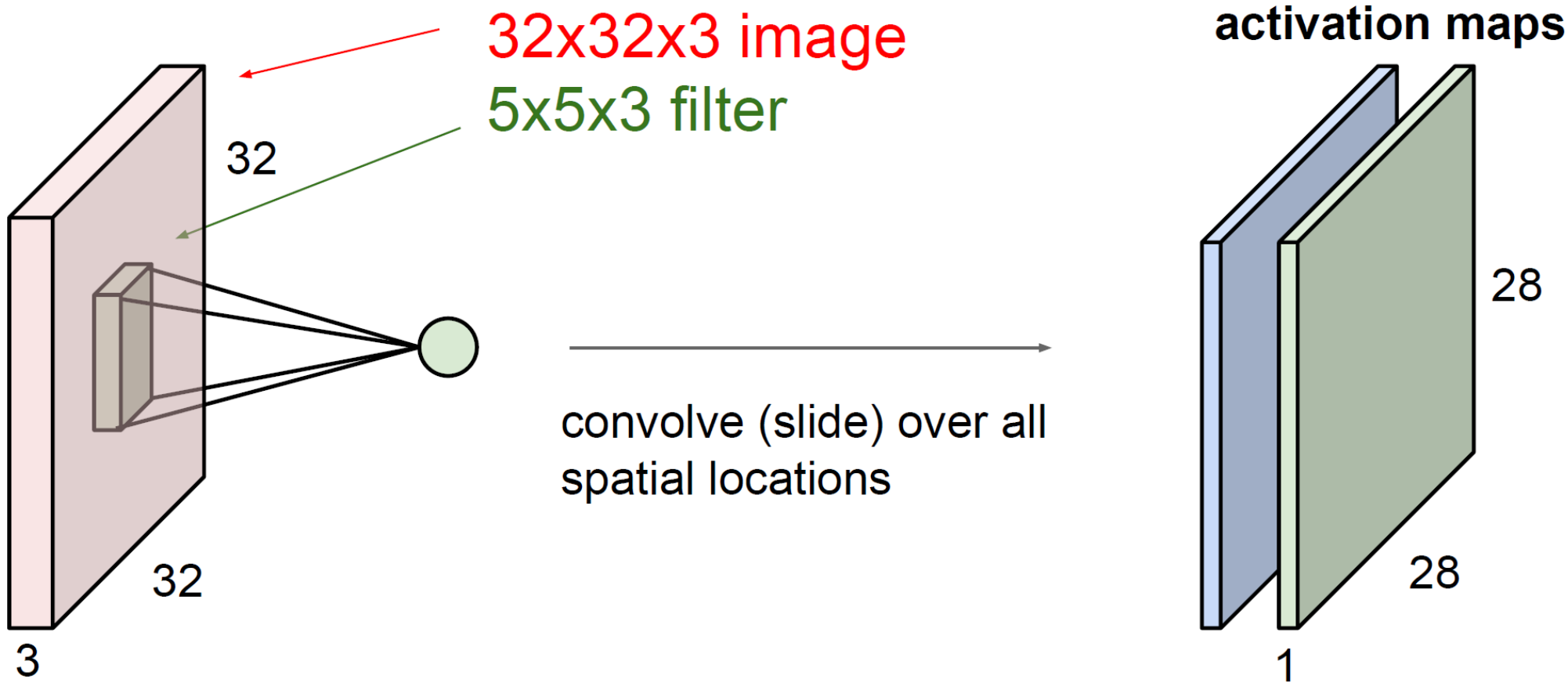
1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

Convolution Layer

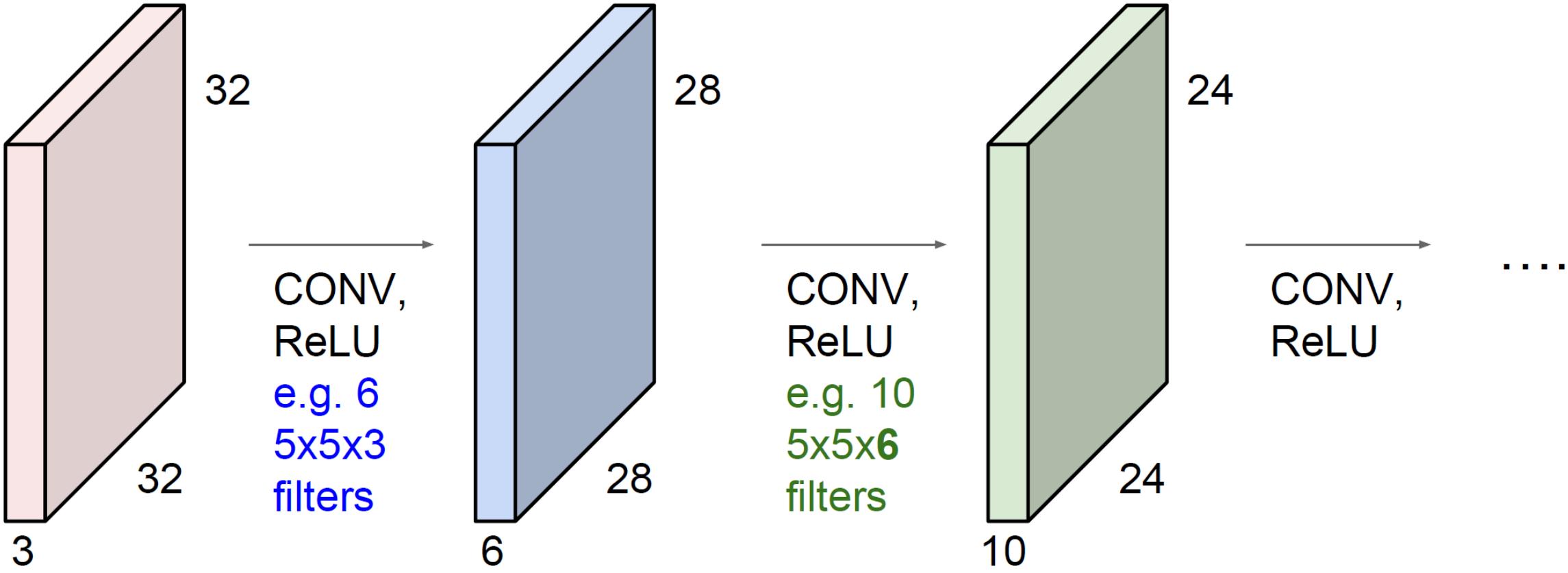
: Preserve the spatial structure

Convolution Layer
: Preserve the spatial structure



Convolutional Net

: Sequence of Convolutional Layers, interspersed with activation functions

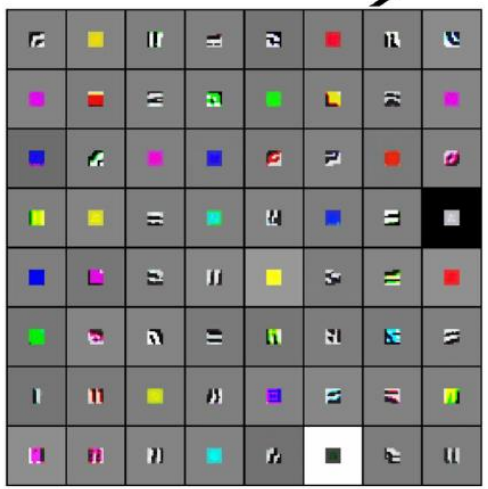


Low-level features

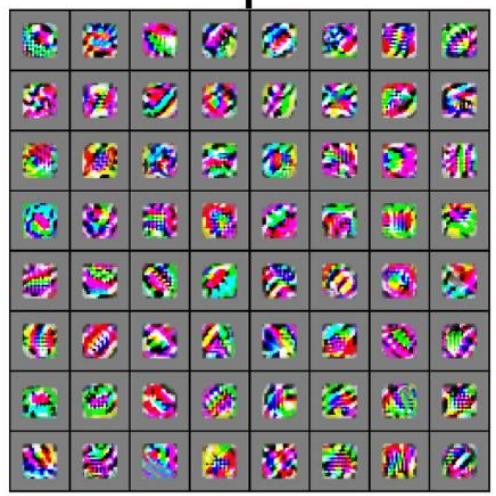
Mid-level features

High-level features

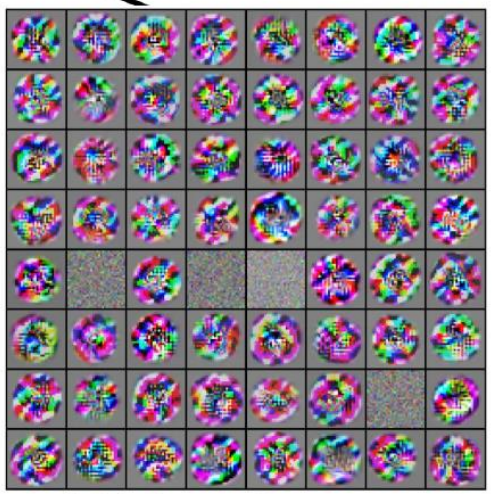
Linearly separable classifier



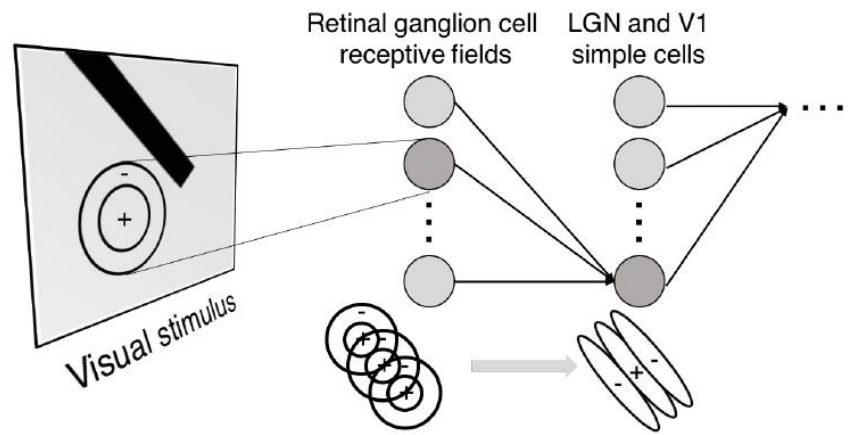
VGG-16 Conv1_1



VGG-16 Conv3_2



VGG-16 Conv5_3



Complex cells:
Response to light orientation and movement

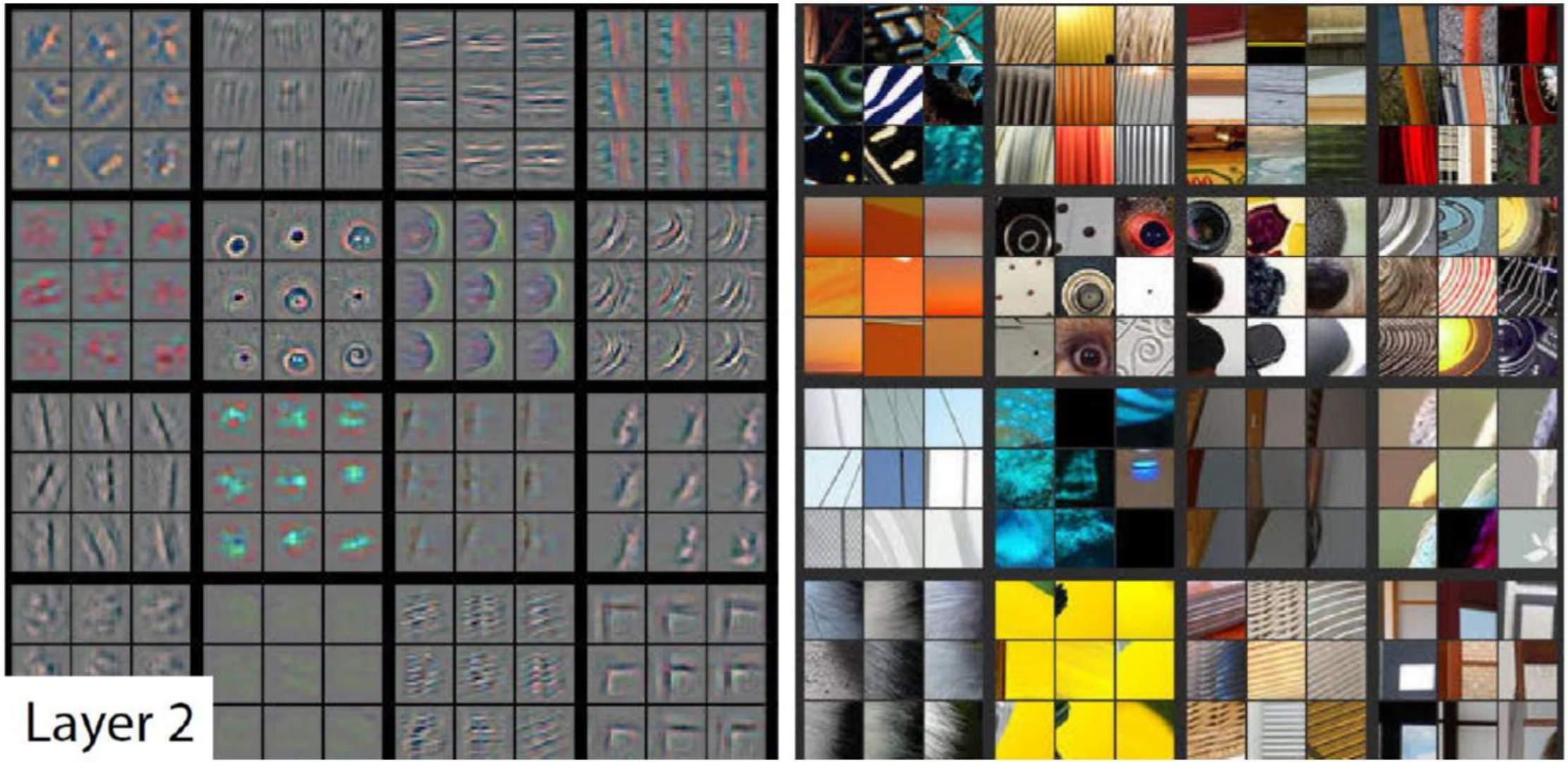
Hypercomplex cells:
response to movement with an end point

Stanford, cs231n 2017

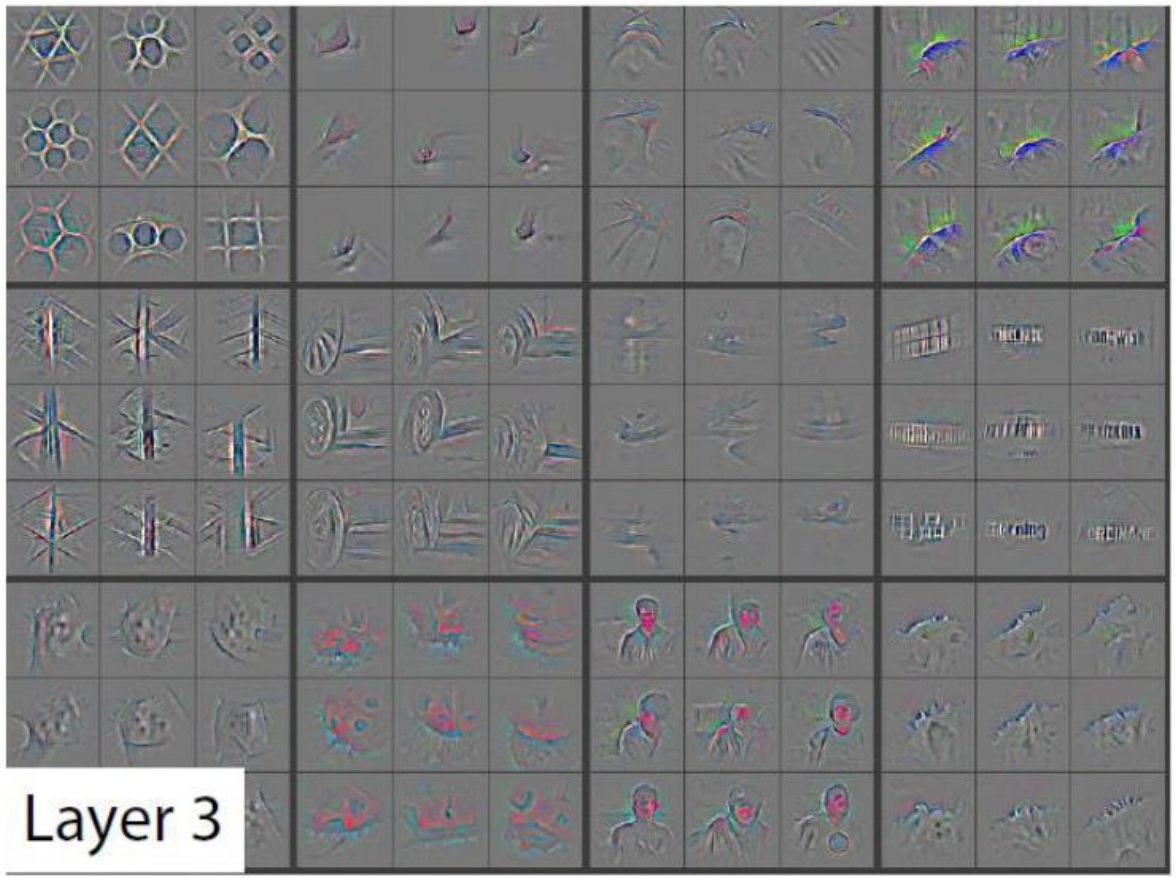
Layer 1

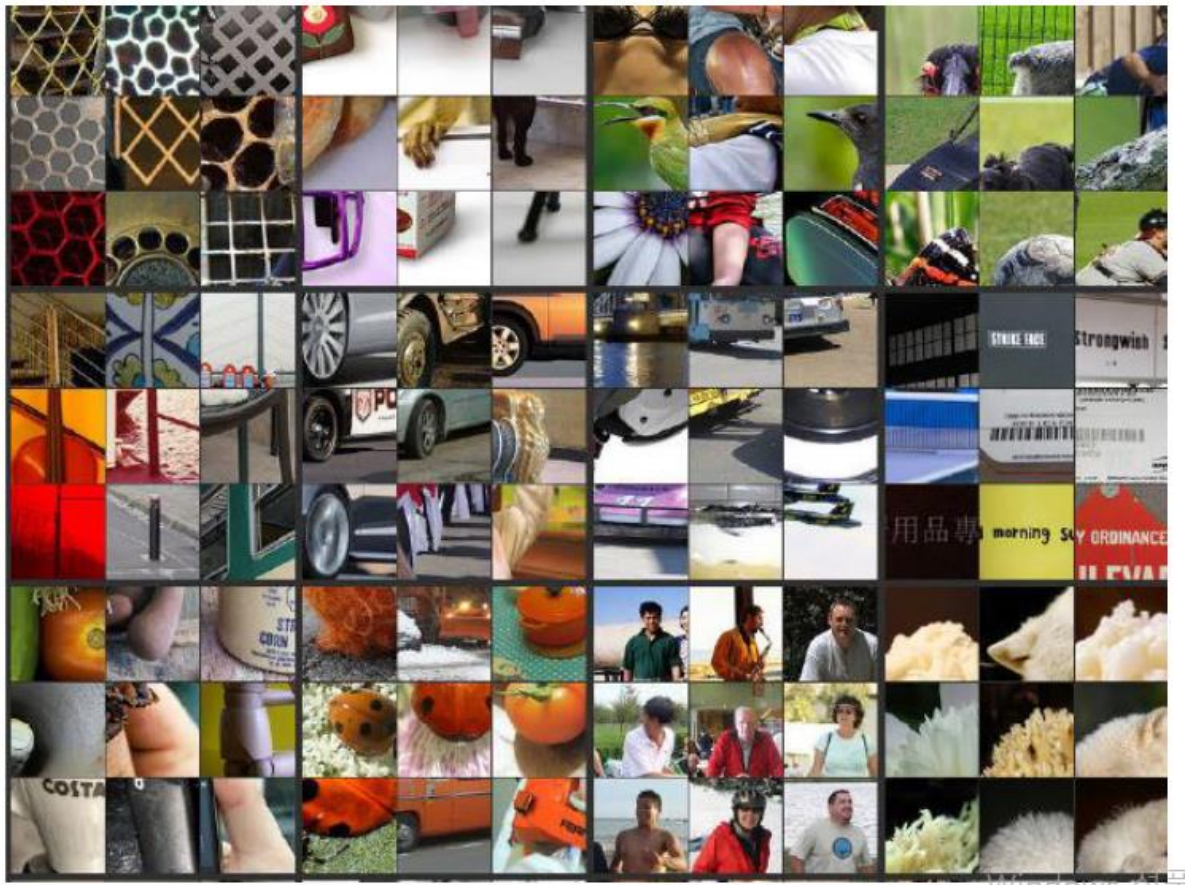
Layer 1

Layer 2



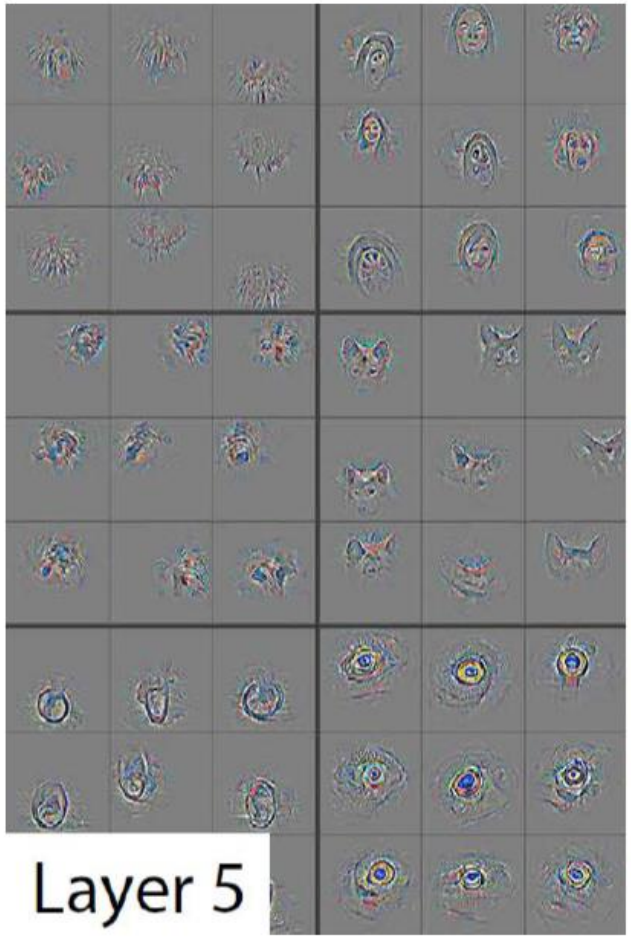
Layer 3

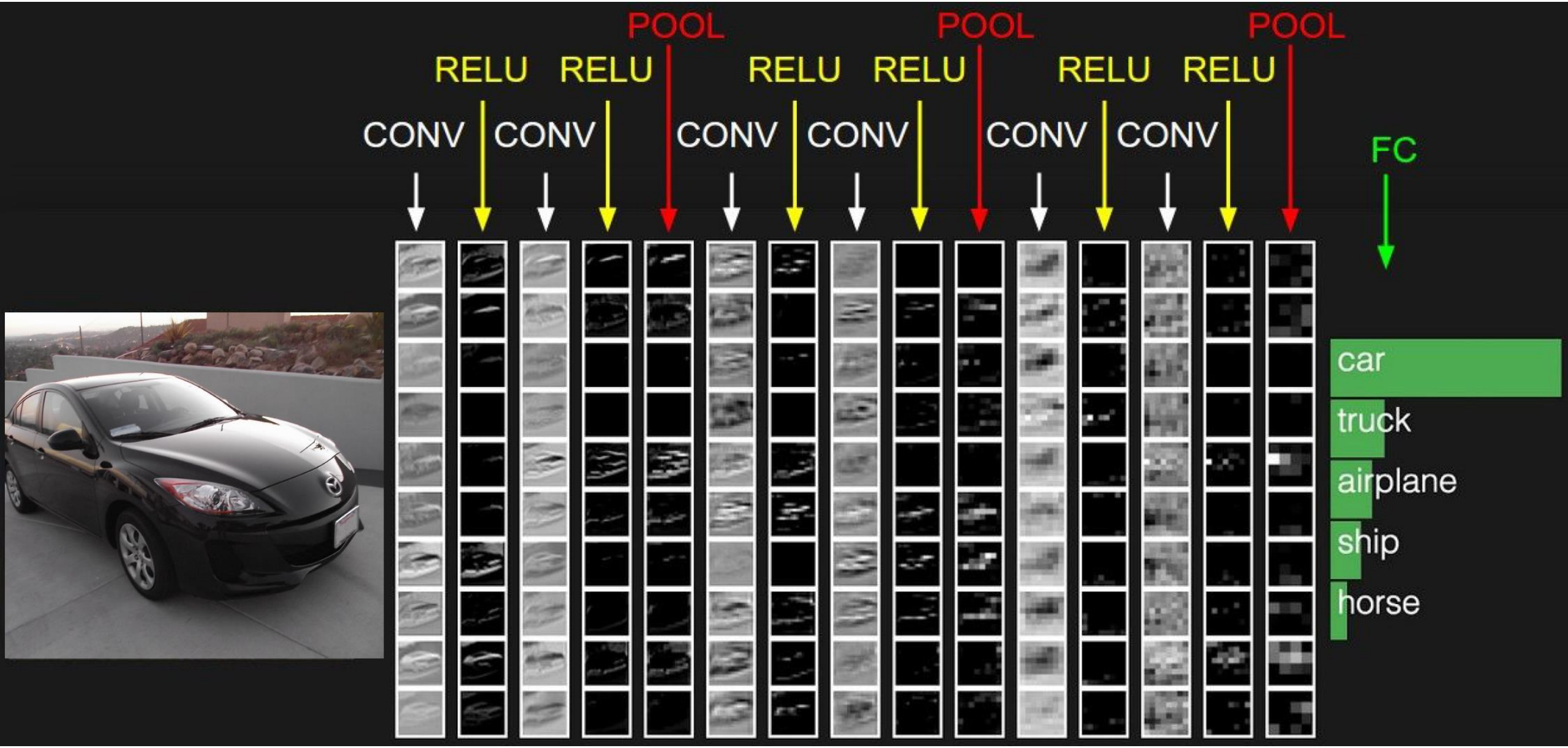




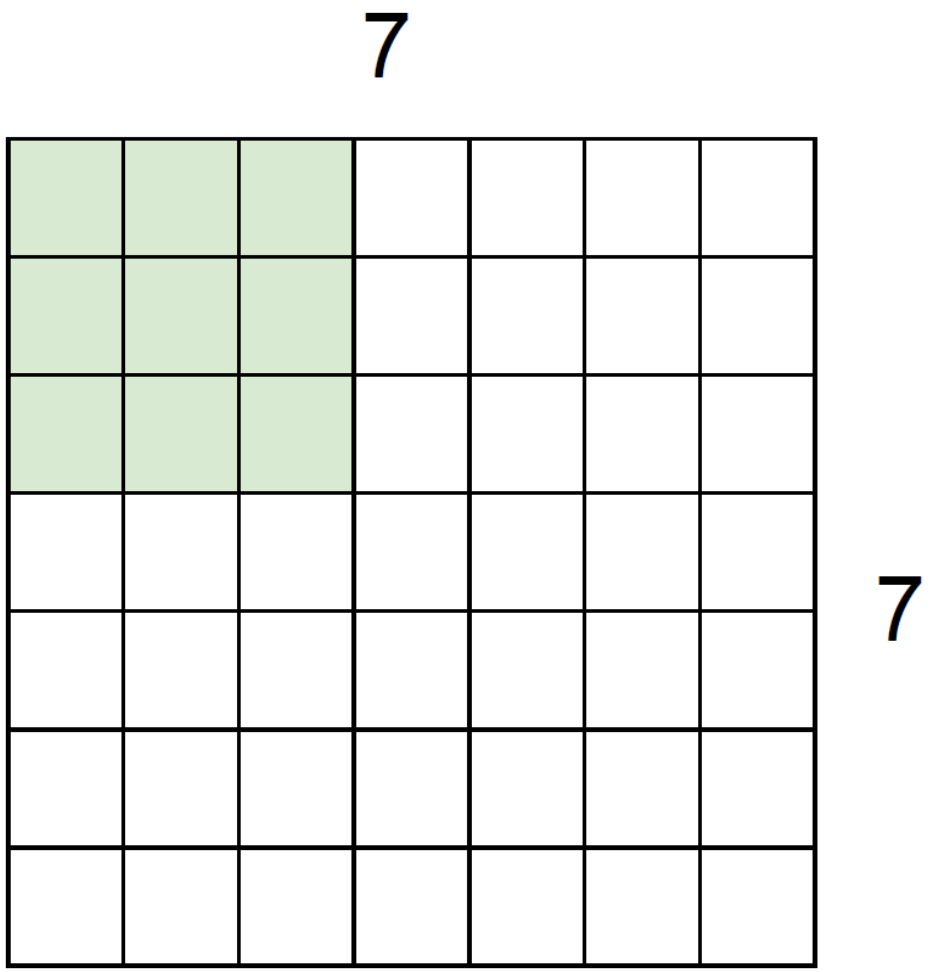
Neuron view of Convolutional Layer

Layer 5



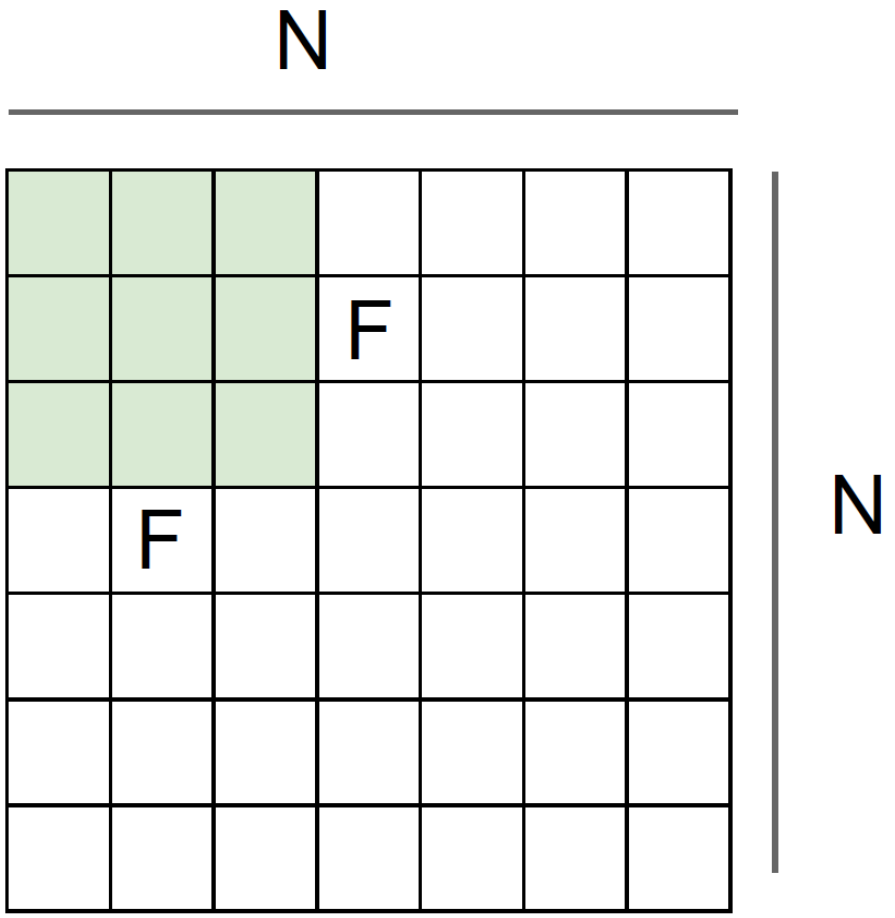

POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

Calculating spatial dimension



7x7 input (spatially)
assume 3x3 filter

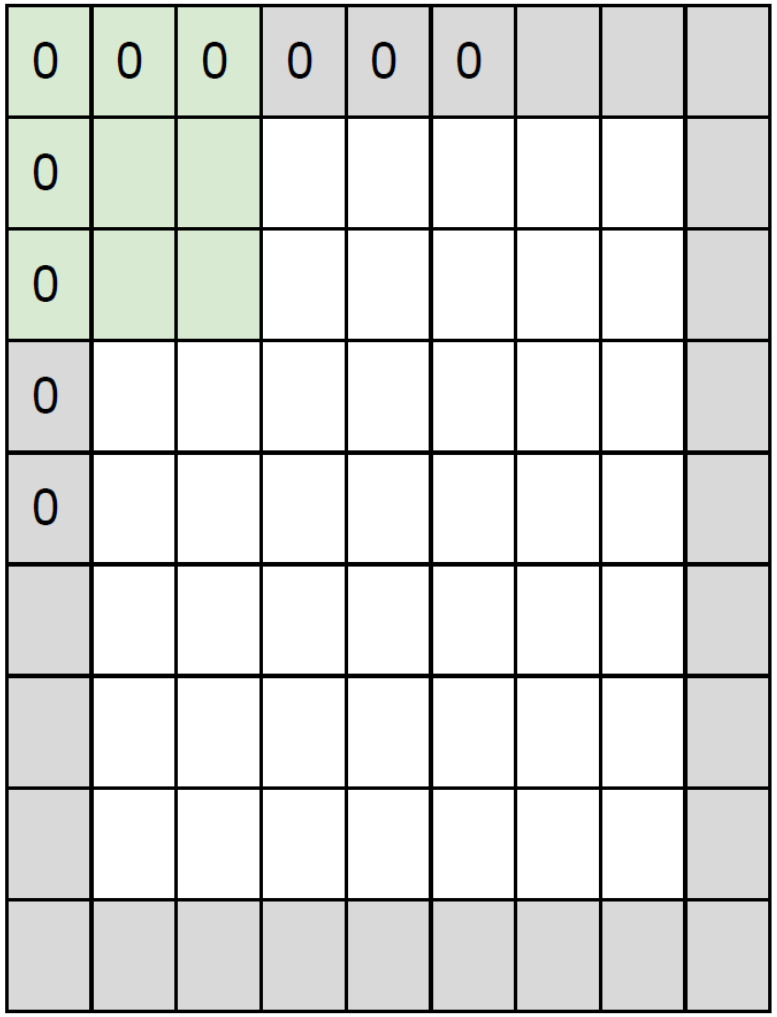
Calculating spatial dimension



Output size:
(N - F) / stride + 1

e.g. N = 7, F = 3:
stride 1 => (7 - 3)/1 + 1 = 5
stride 2 => (7 - 3)/2 + 1 = 3
stride 3 => (7 - 3)/3 + 1 = 2.33

Zero padding



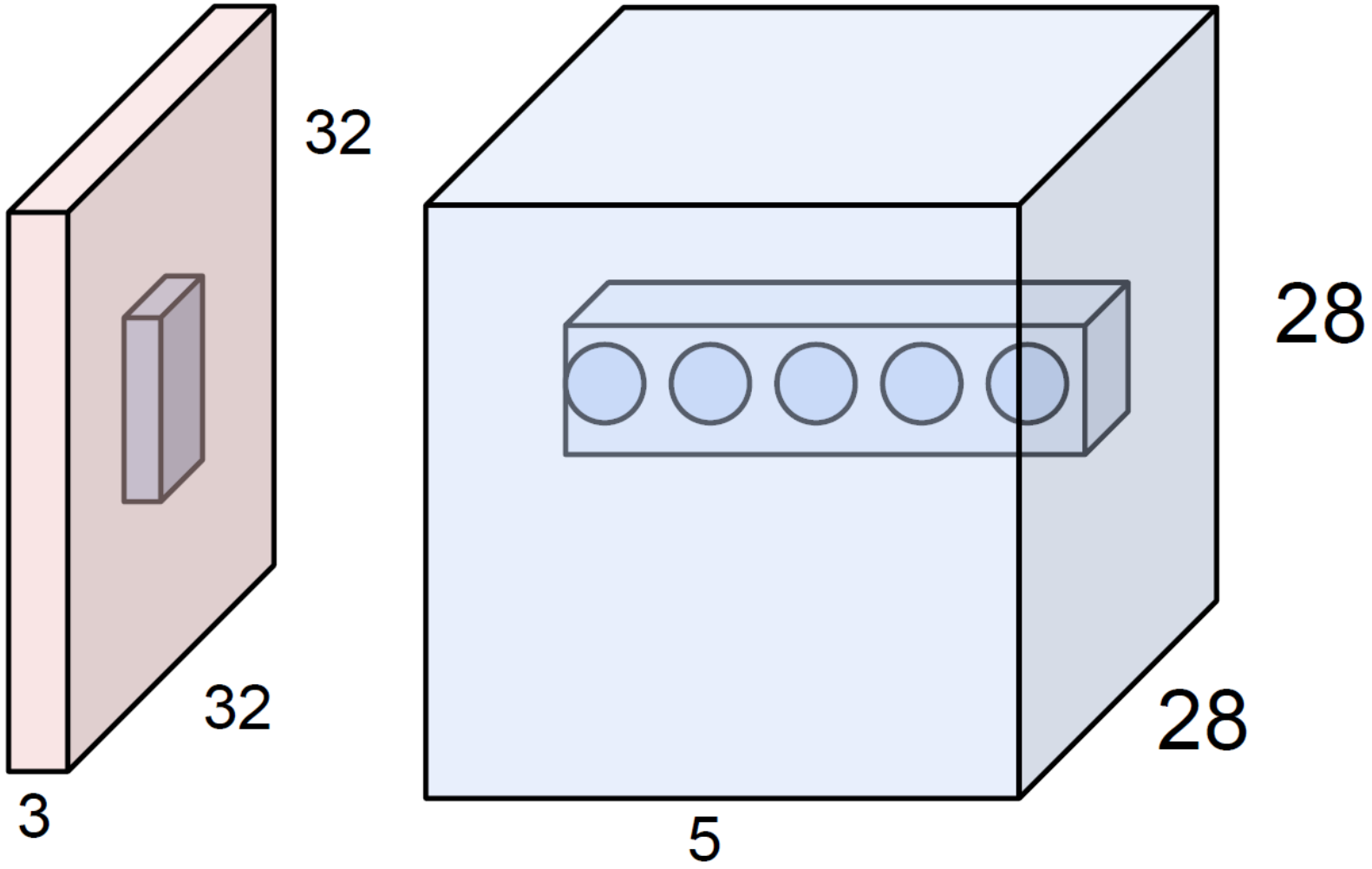
e.g. input 7x7
3x3 filter, applied with **stride 1**
pad with 1 pixel border => what is the output?

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

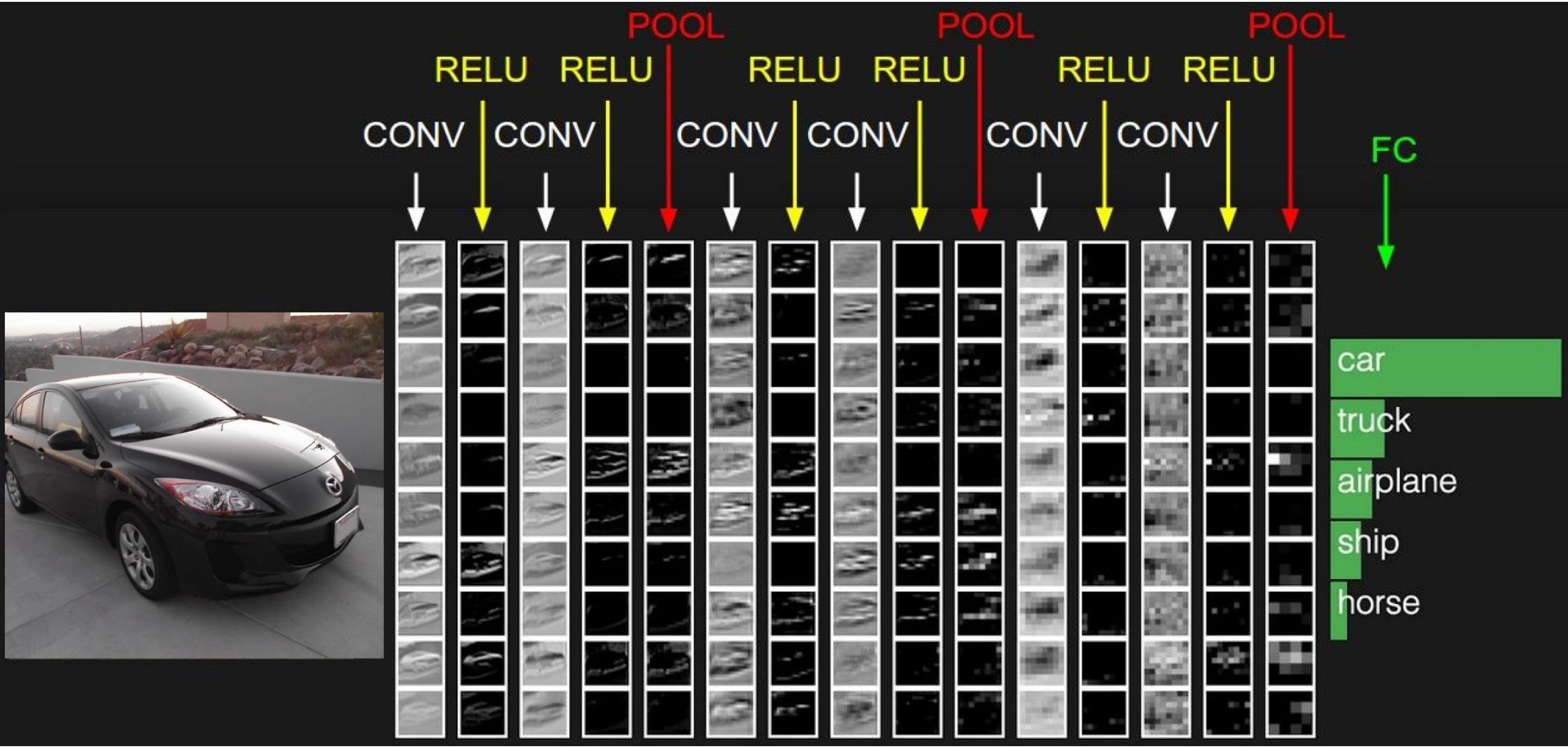
Dimension Formula

- 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)
  - $D_2 = K$

Neuron view of Convolutional Layer

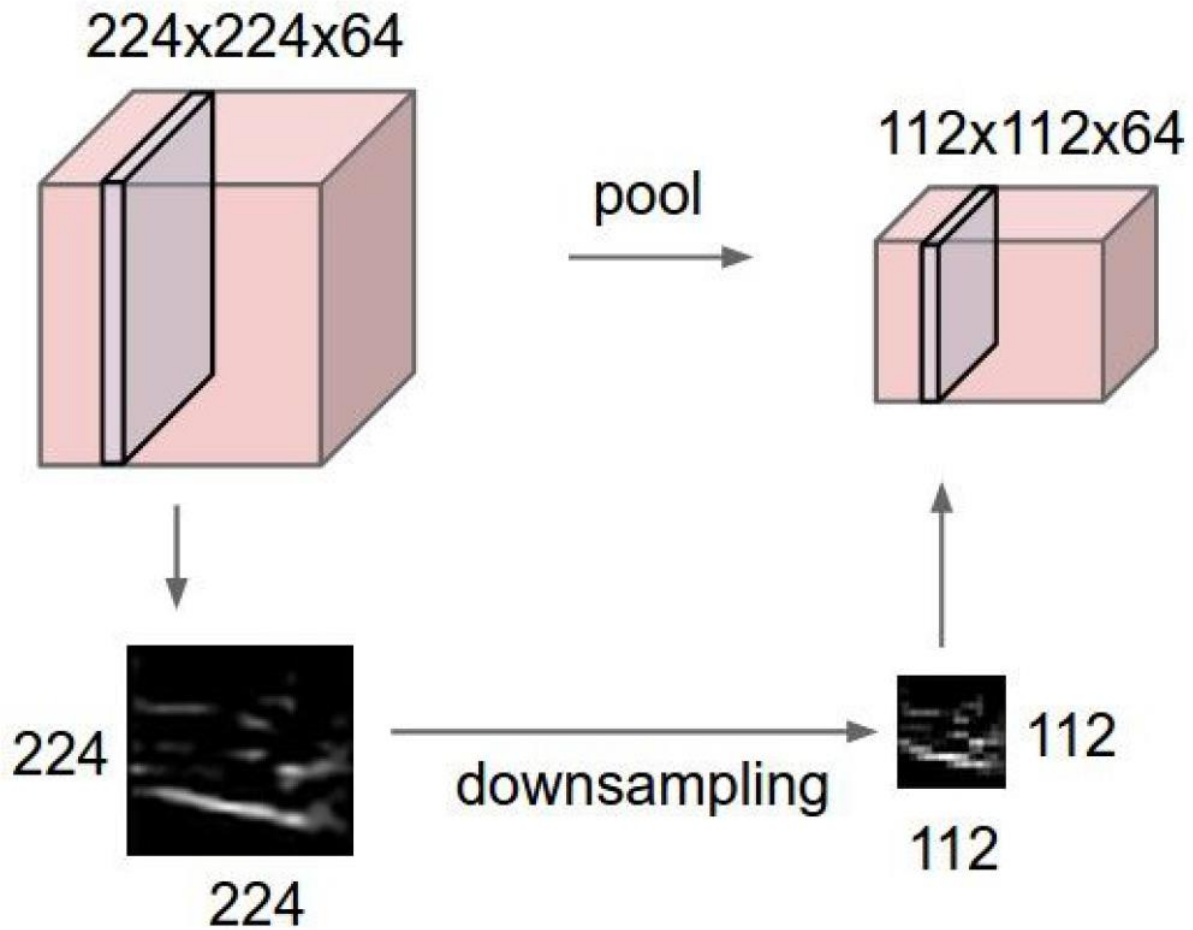


Pooling and FC Layer

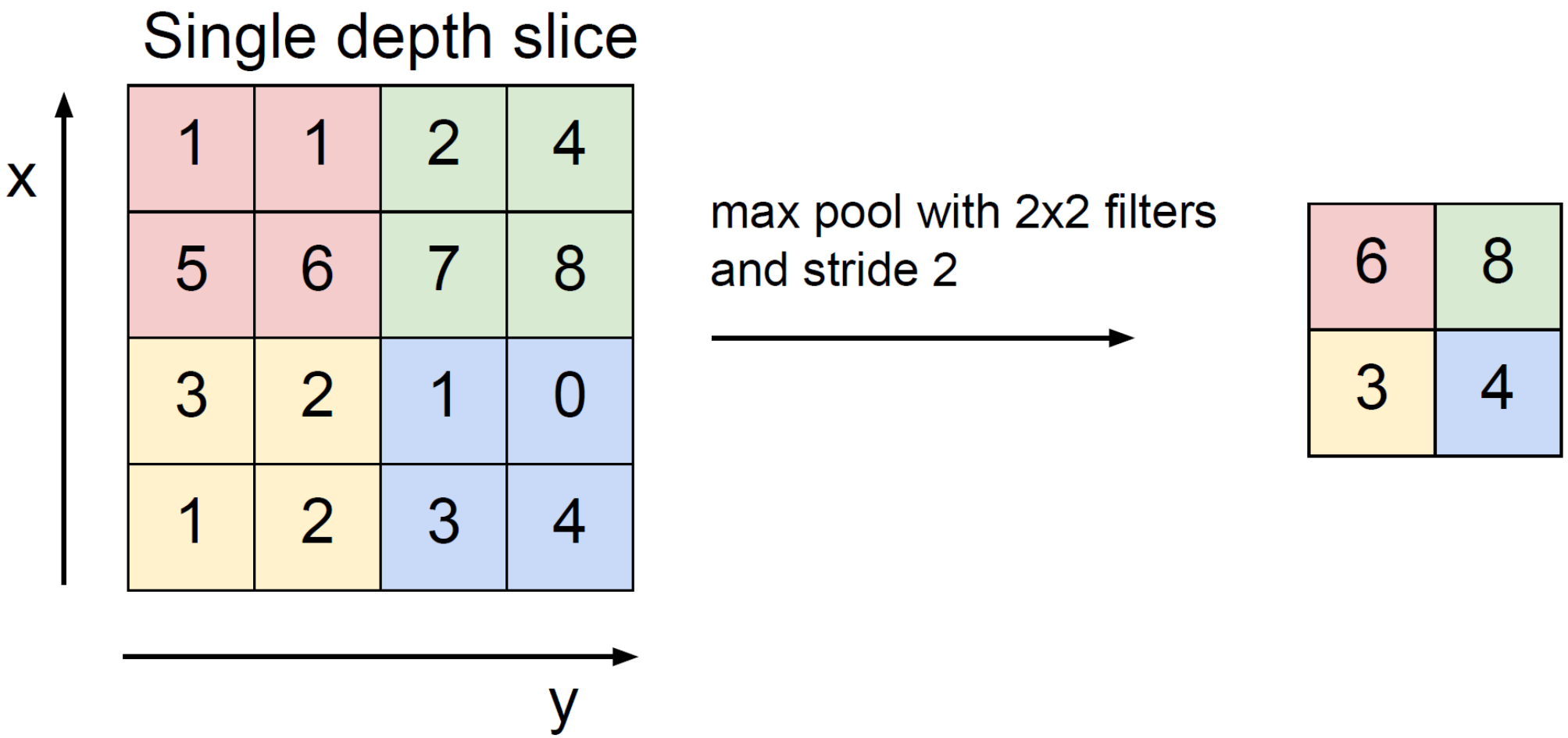

POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

Pooling layer

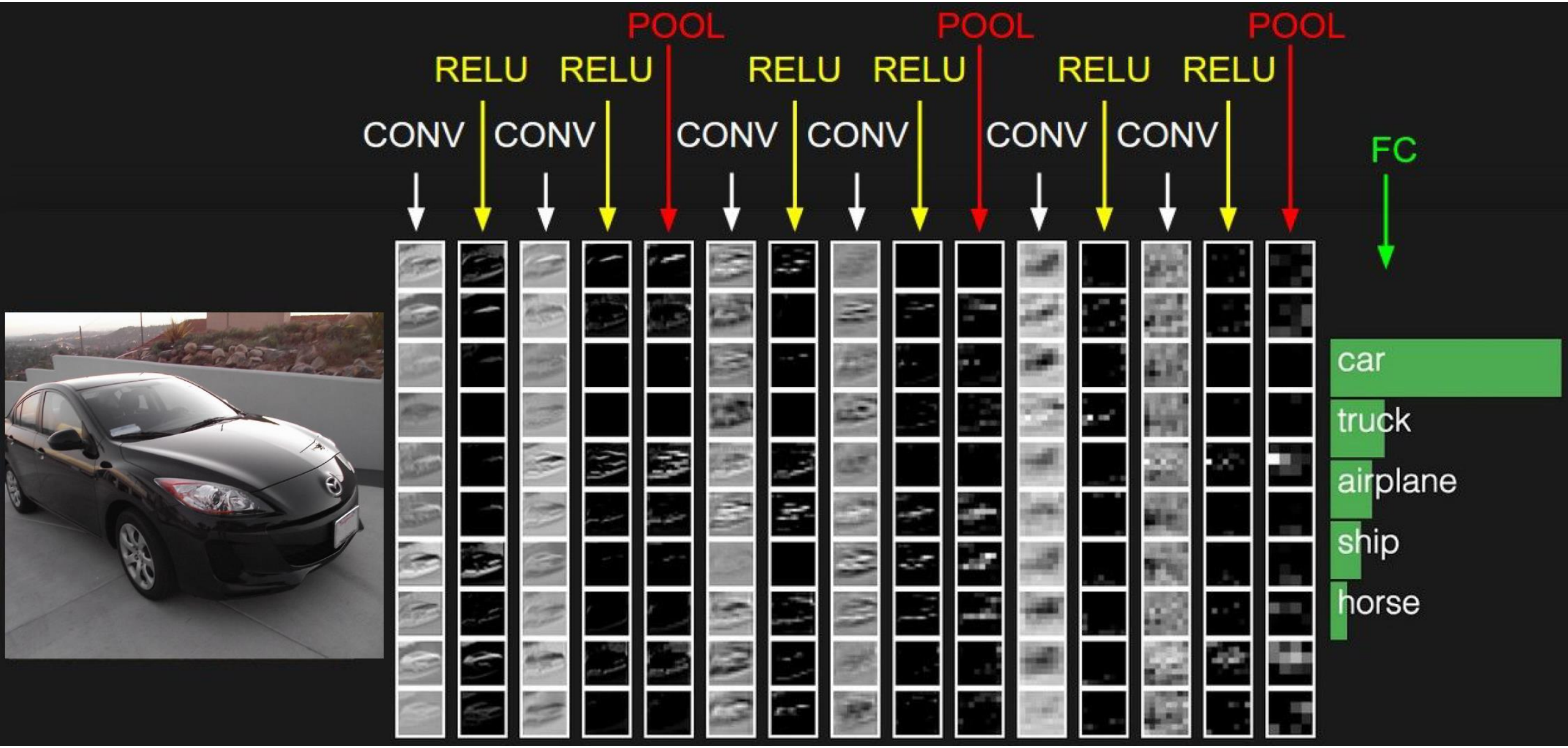
- makes the representations smaller and more manageable
- operates over each activation map independently:



MAX POOLING


Single depth slice
x
y
1 1 2 4
5 6 7 8
3 2 1 0
1 2 3 4
max pool with 2x2 filters
and stride 2
6 8
3 4

Fully Connected Layer


POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse