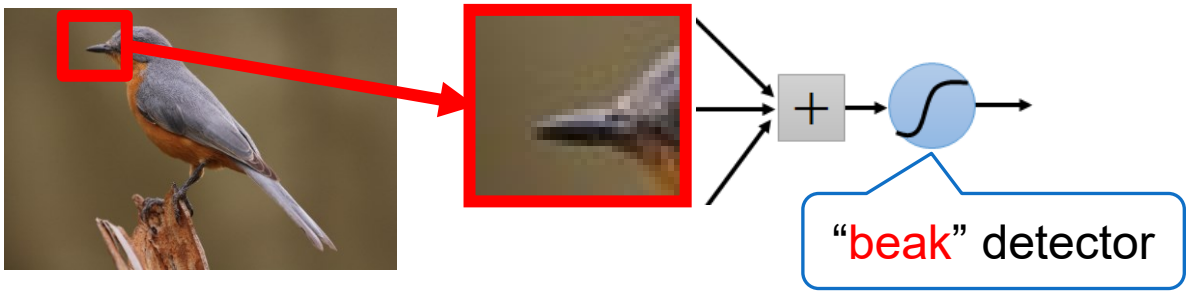
Color Images, Multiple Filters, and Parameters Sharing

Consider Learning an Image

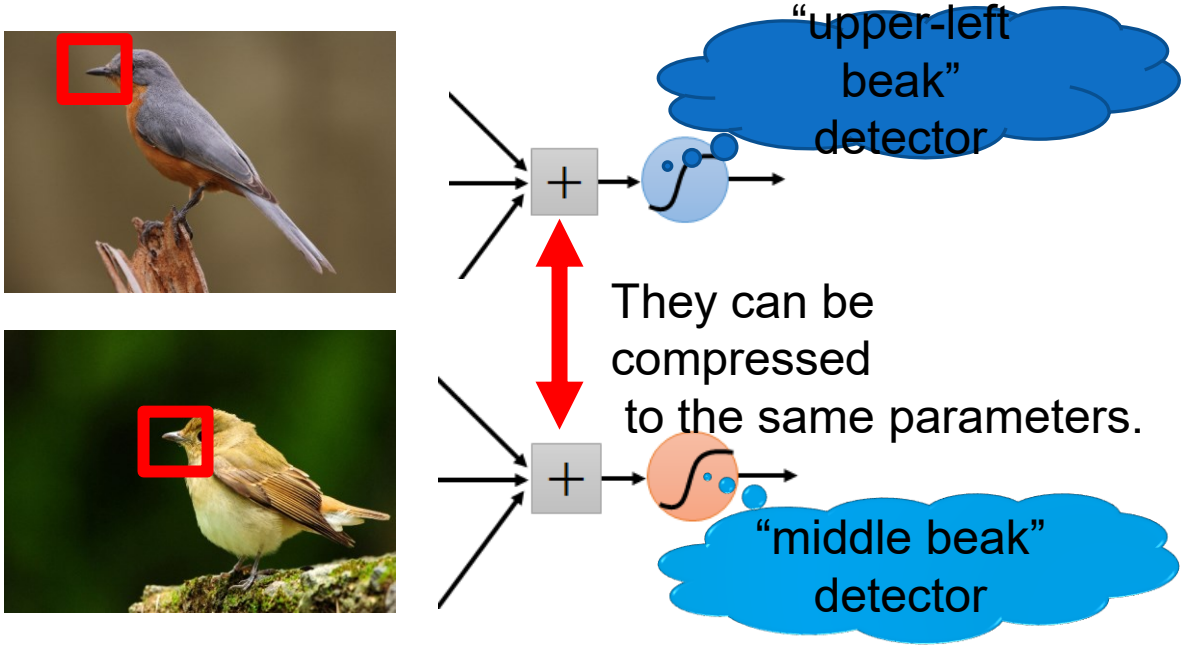
- Some patterns are much smaller than the whole image

Can represent a small region with fewer parameters



What if Pattern Location Changes?

- What about training a lot of such “small” detectors, and each detector must “move around”?



Convolution

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

These are the network parameters to be learned.

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

-1	1	-1
-1	1	-1
-1	1	-1

Filter 2

⋮ ⋮

Each filter detects a small pattern (3 x 3).

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

stride= 1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Dot product → 3 -1

6 x 6 image

Convolution

1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

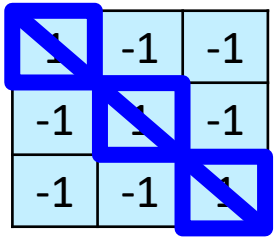
If stride=2

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

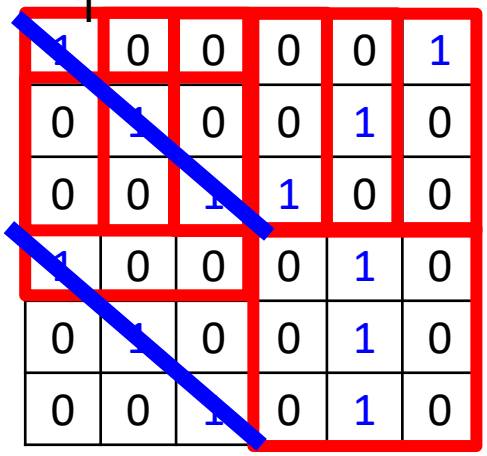
3 -3

Convolution

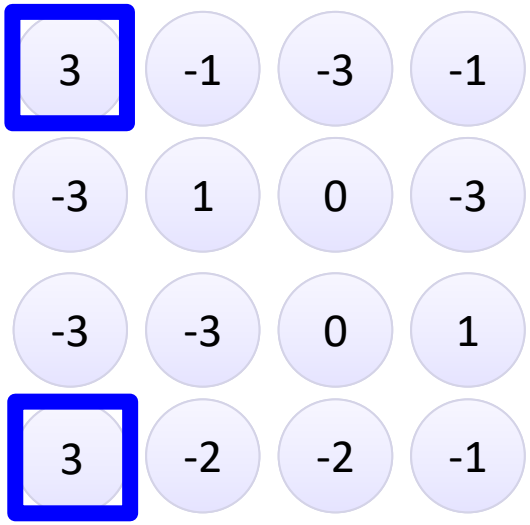


Filter 1

stride= 1



6 x 6 image



Convolution

-1	1	-1
-1	1	-1
-1	1	-1

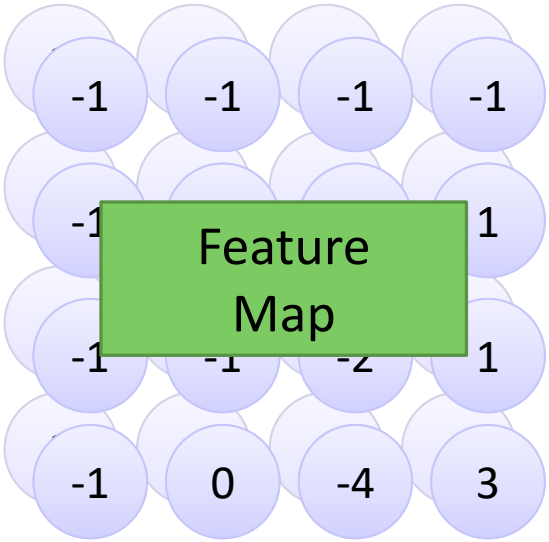
Filter 2

stride= 1

1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

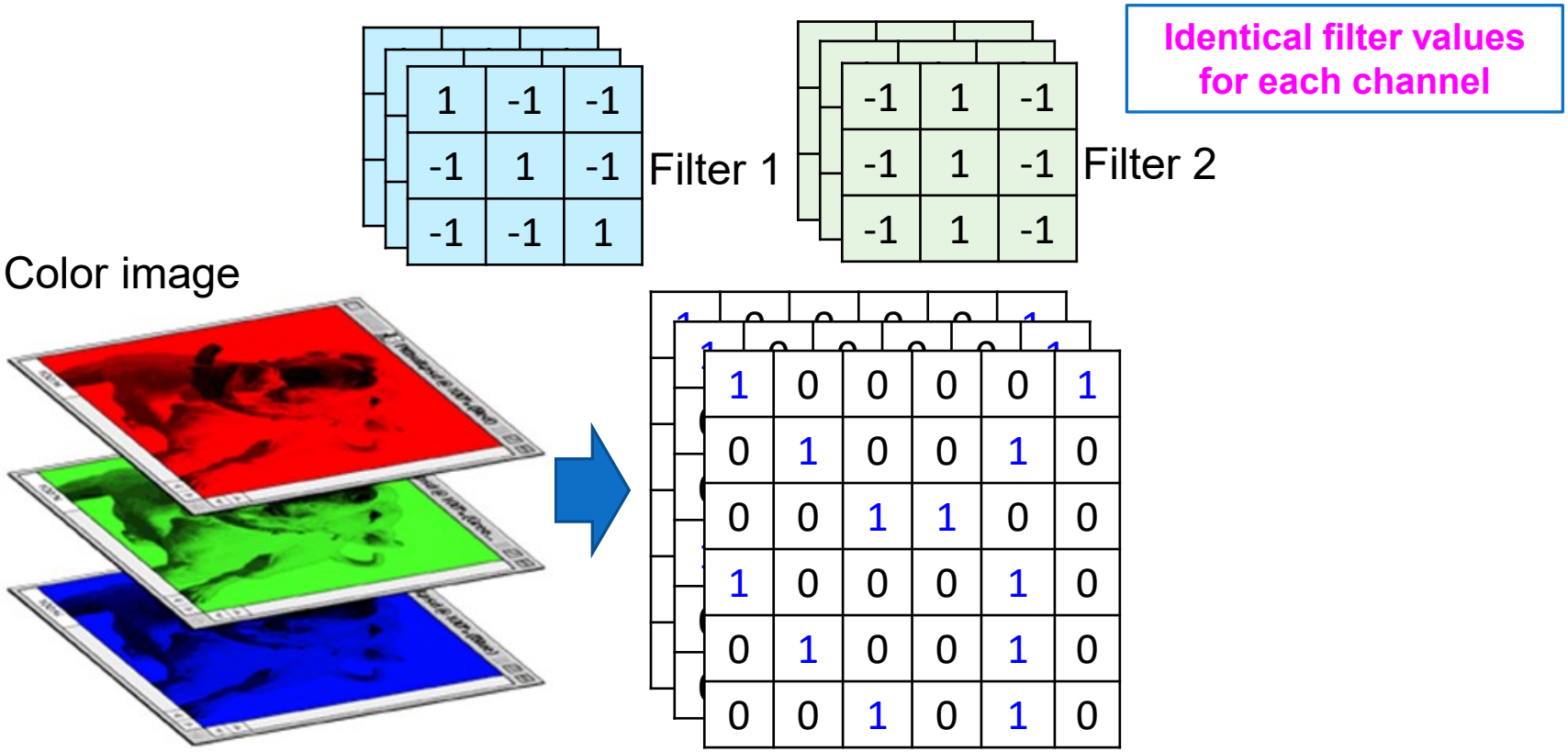
6 x 6 image

Repeat this for each filter

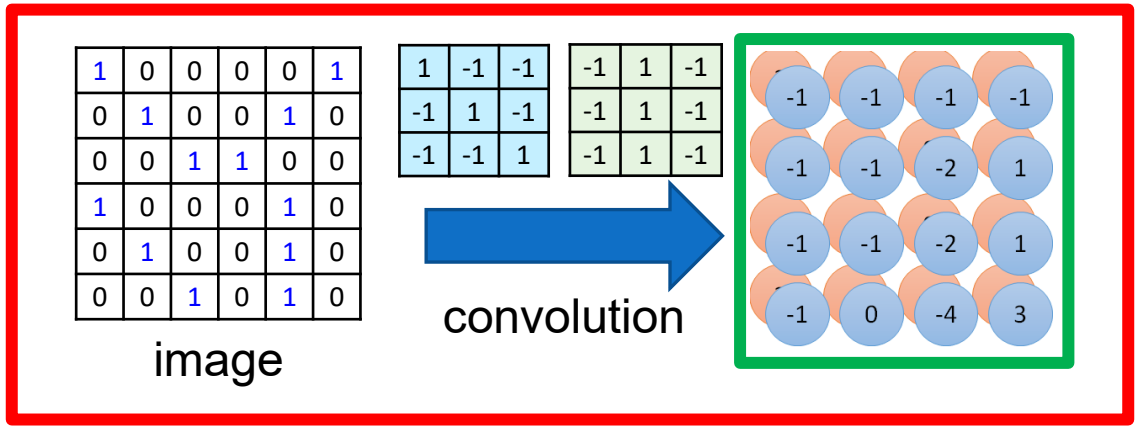


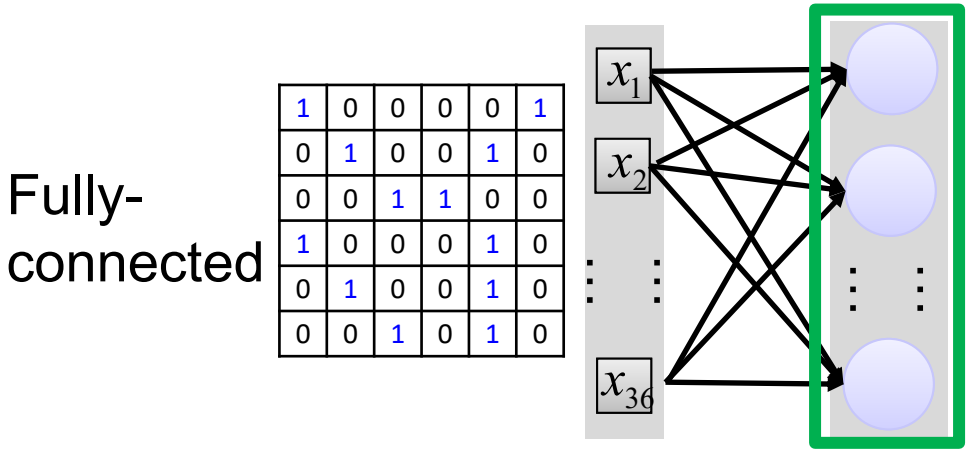
Two 4 x 4 images
Forming 4 x 4 x 2 matrix

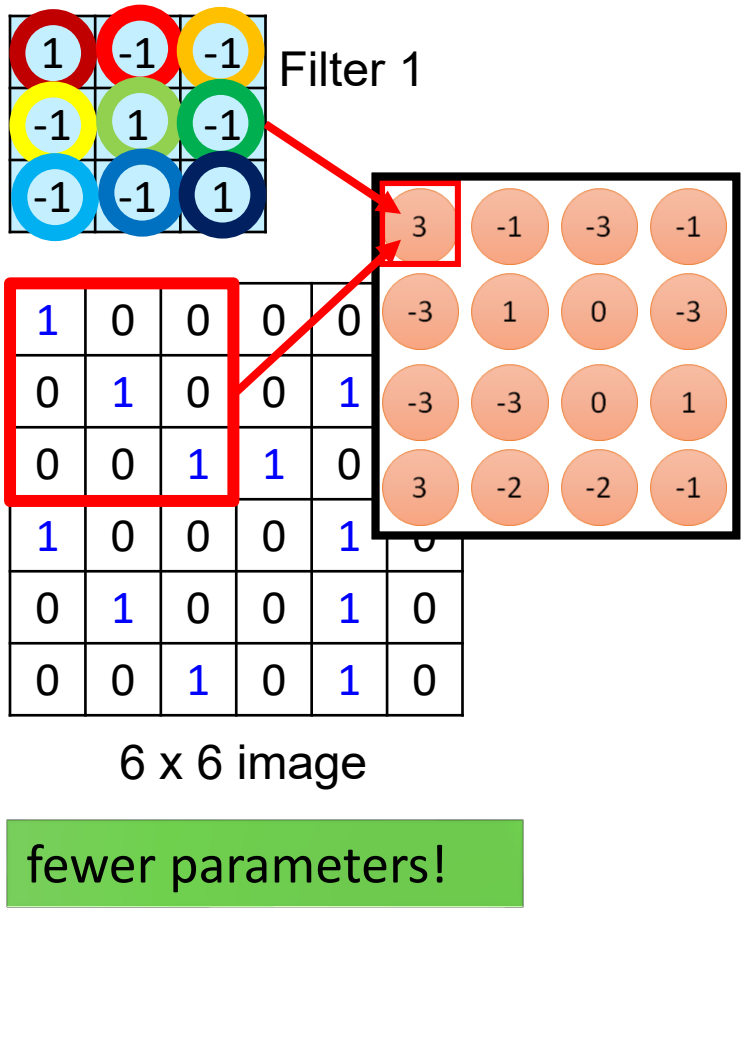
Convolutions: Color Image



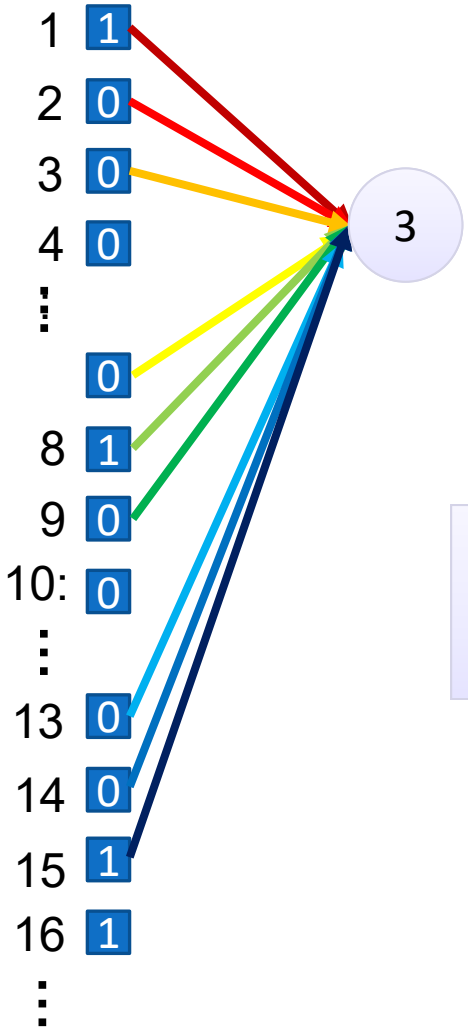
Convolutions VS Fully Connected



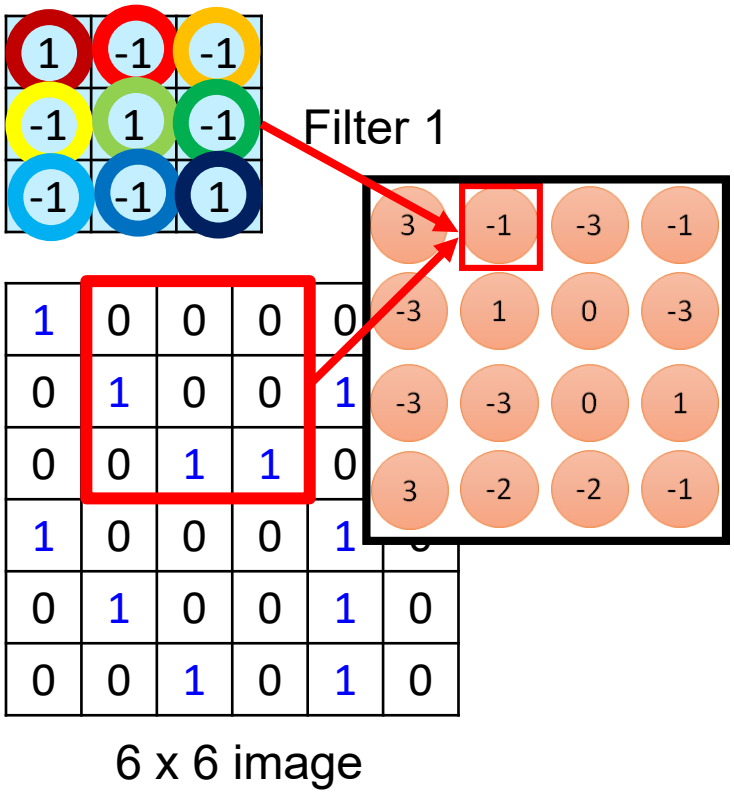




fewer parameters!

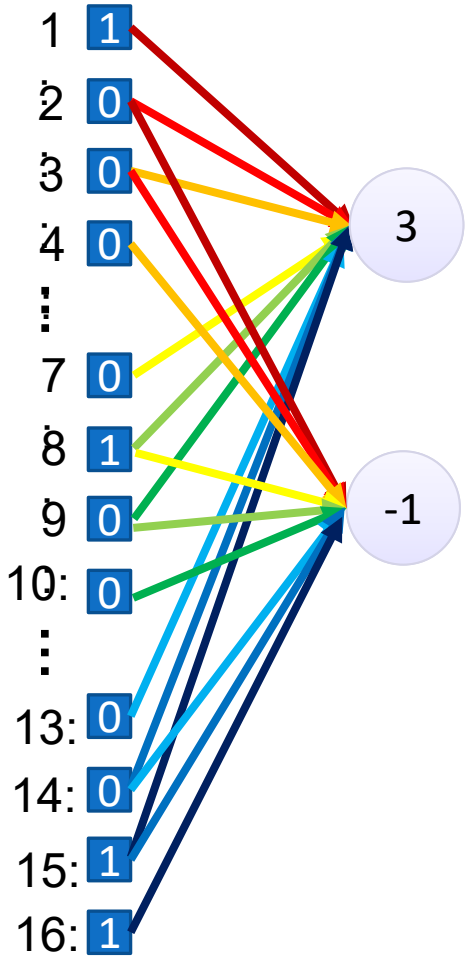


Only connect to 9 inputs, not fully connected



Fewer parameters

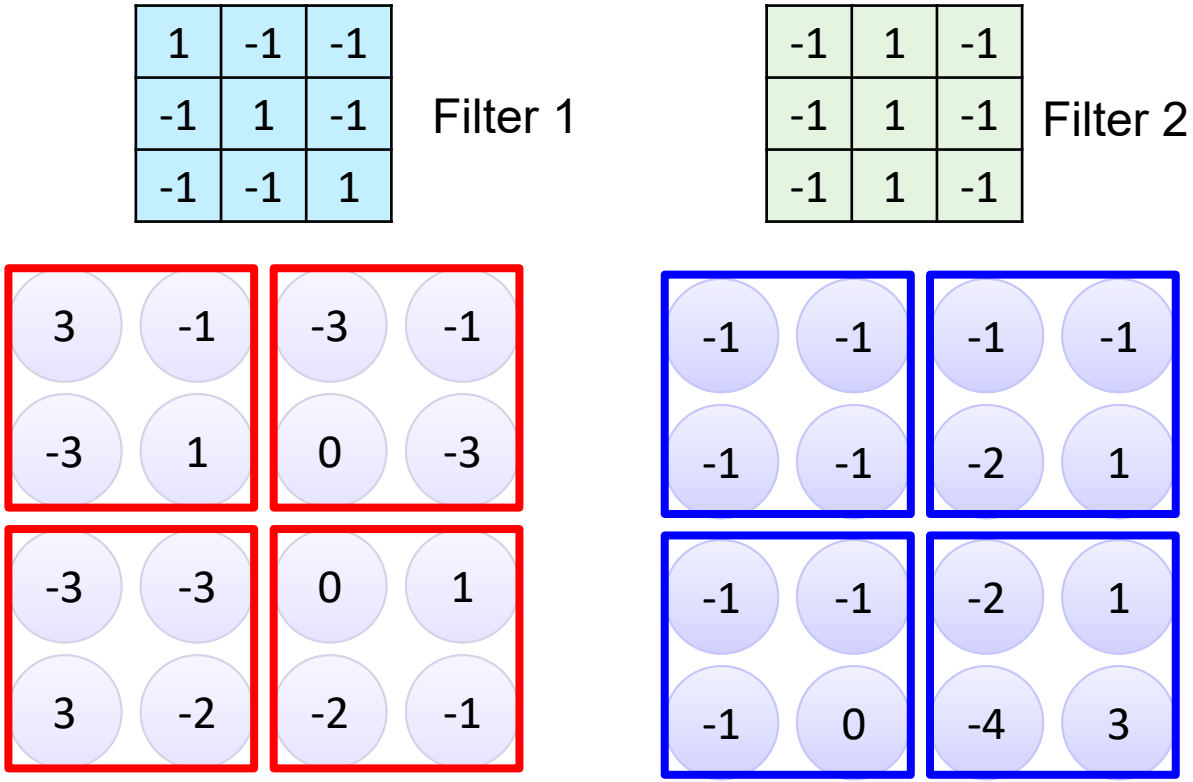
Even fewer parameters



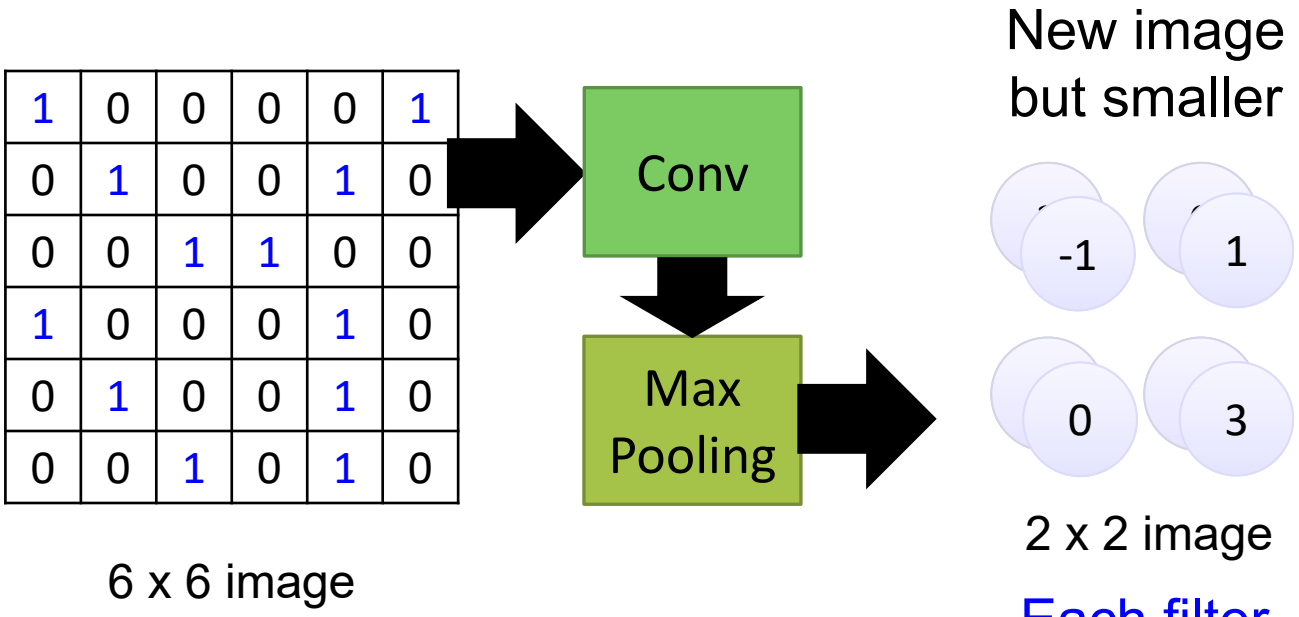
Shared weights

Filter/Kernel values are equivalent to weights in Feedforward network…

Max Pooling

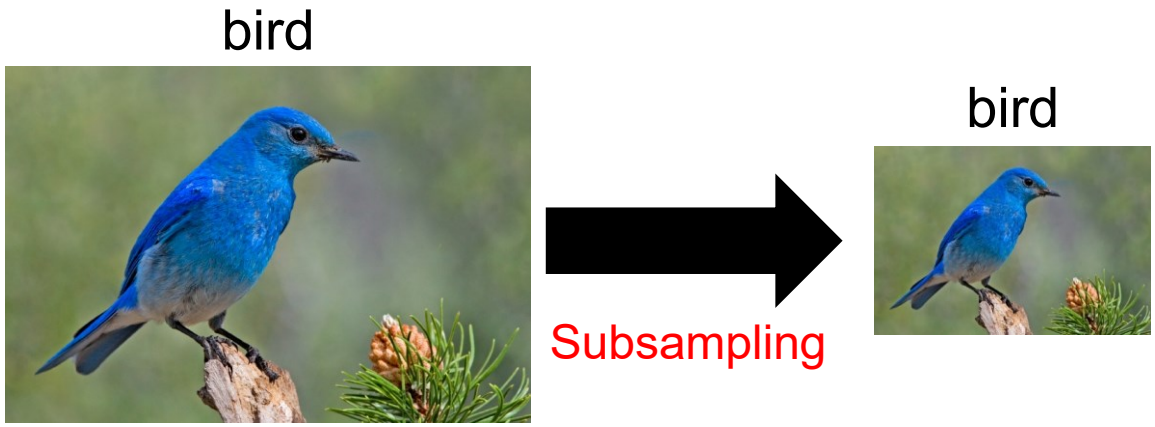

1 -1 -1
-1 1 -1
-1 -1 1
Filter 1
-1 1 -1
-1 1 -1
-1 1 -1
Filter 2
3 -1
-3 1
-3 -1
0 -3
-3 -3
3 -2
0 1
-2 -1
-1 -1
-1 -1
-1 -1
-2 1
-1 -1
-1 0
-2 1
-4 3

Max Pooling


1 0 0 0 0 1
0 1 0 0 1 0
0 0 1 1 0 0
1 0 0 0 1 0
0 1 0 0 1 0
0 0 1 0 1 0
Conv
Max Pooling
New image but smaller
-1
1
0
3
6 x 6 image
2 x 2 image
Each filter is a channel

Why Pooling?

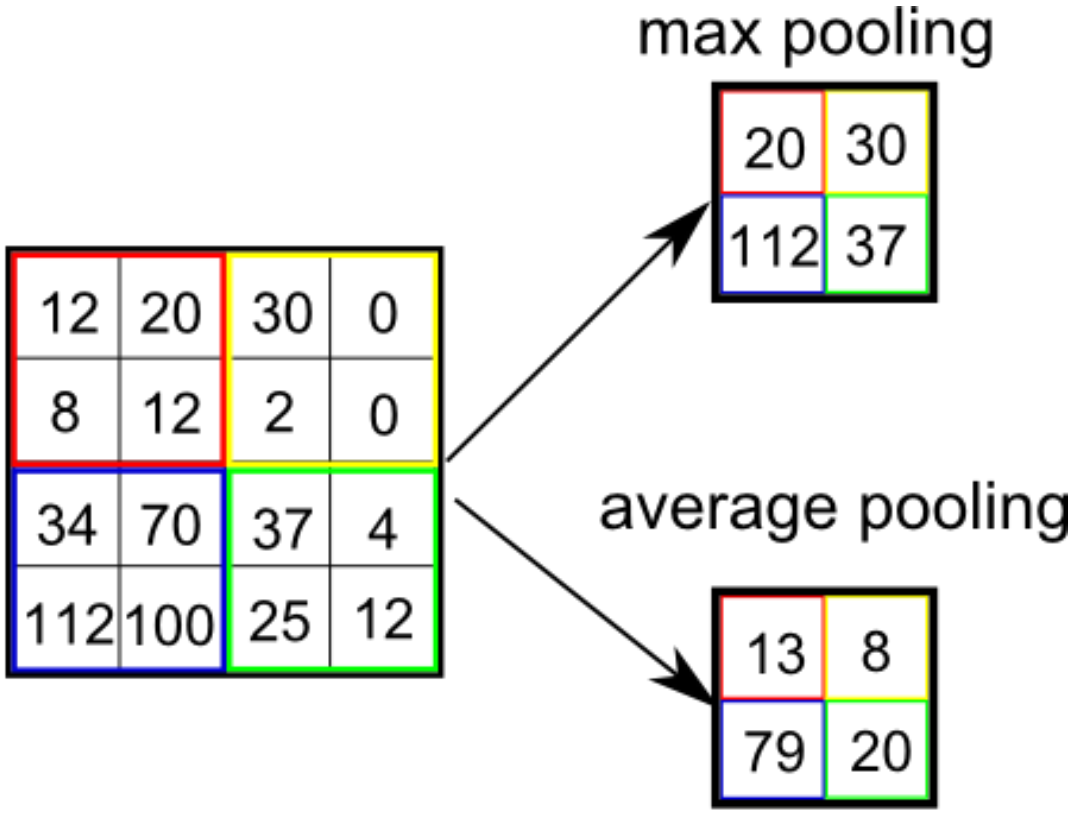
- Subsampling the pixels will **not change the object!!**



We can subsample the pixels to make image smaller

→ fewer parameters to characterize the image

Pooling Strategies

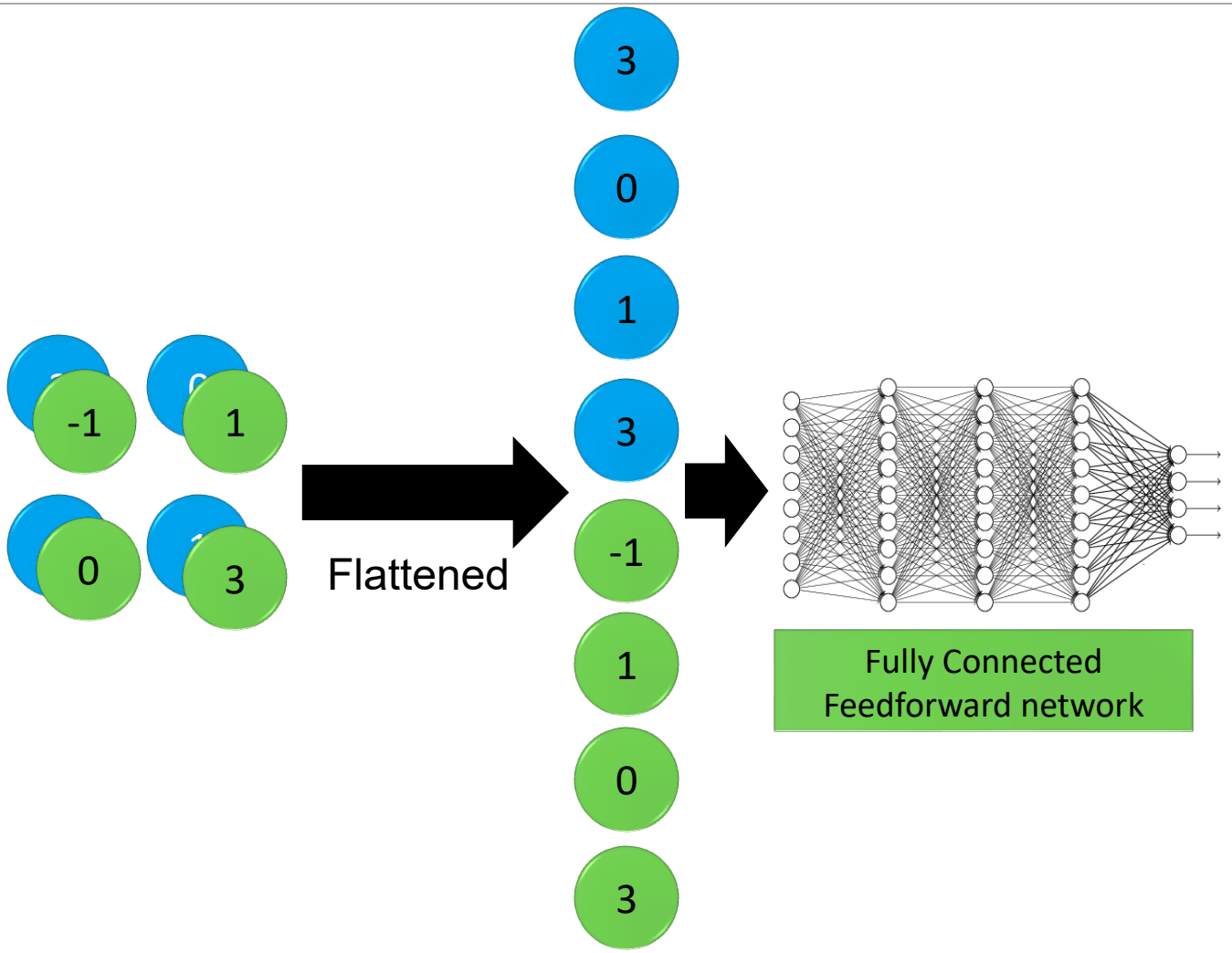


A CNN Compresses A Fully Connected Network…

- CNN Compresses a fully connected network in two ways:
  - Reducing number of connections
  - Shared weights

- Max/Average Pooling further reduces the complexity

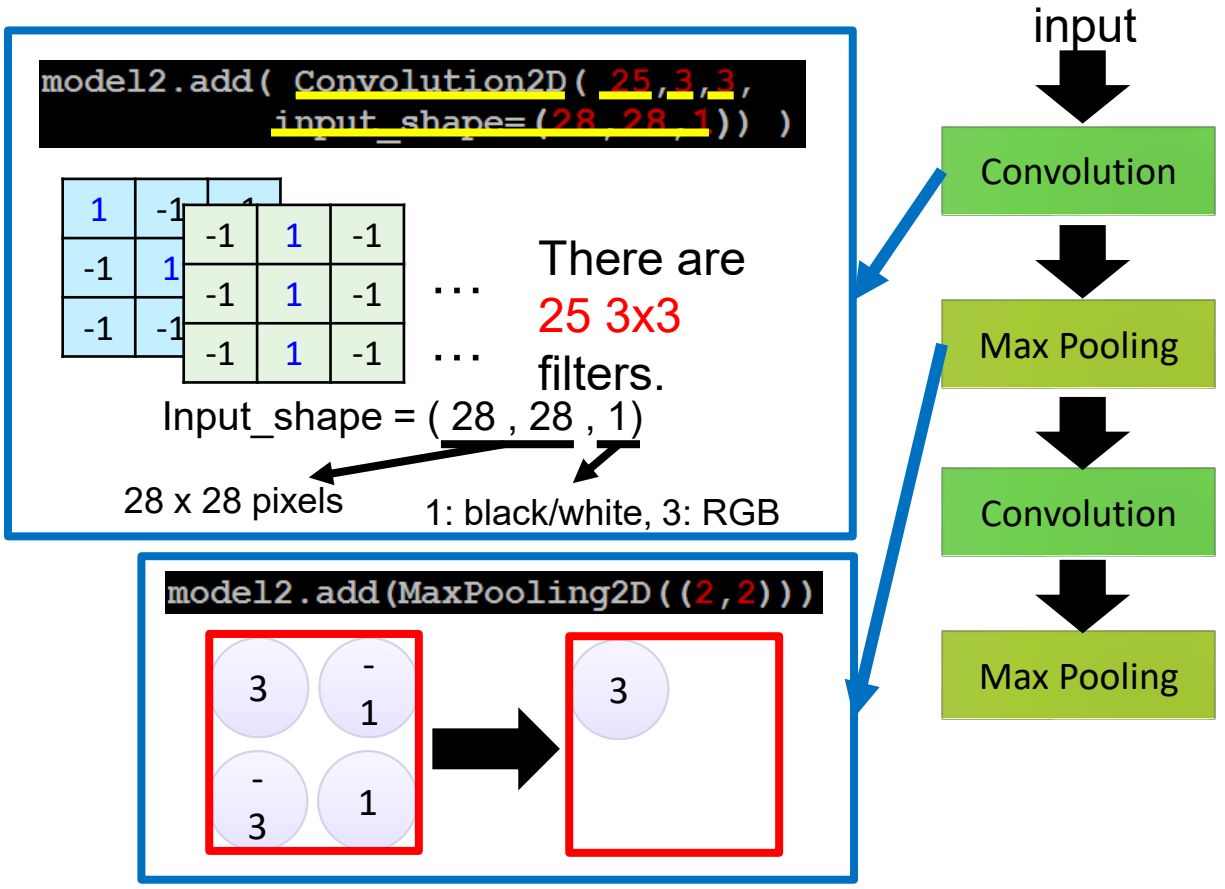
Flattening


-1
1
0
3
Flattened
3
0
1
3
-1
1
0
3
Fully Connected
Feedforward network

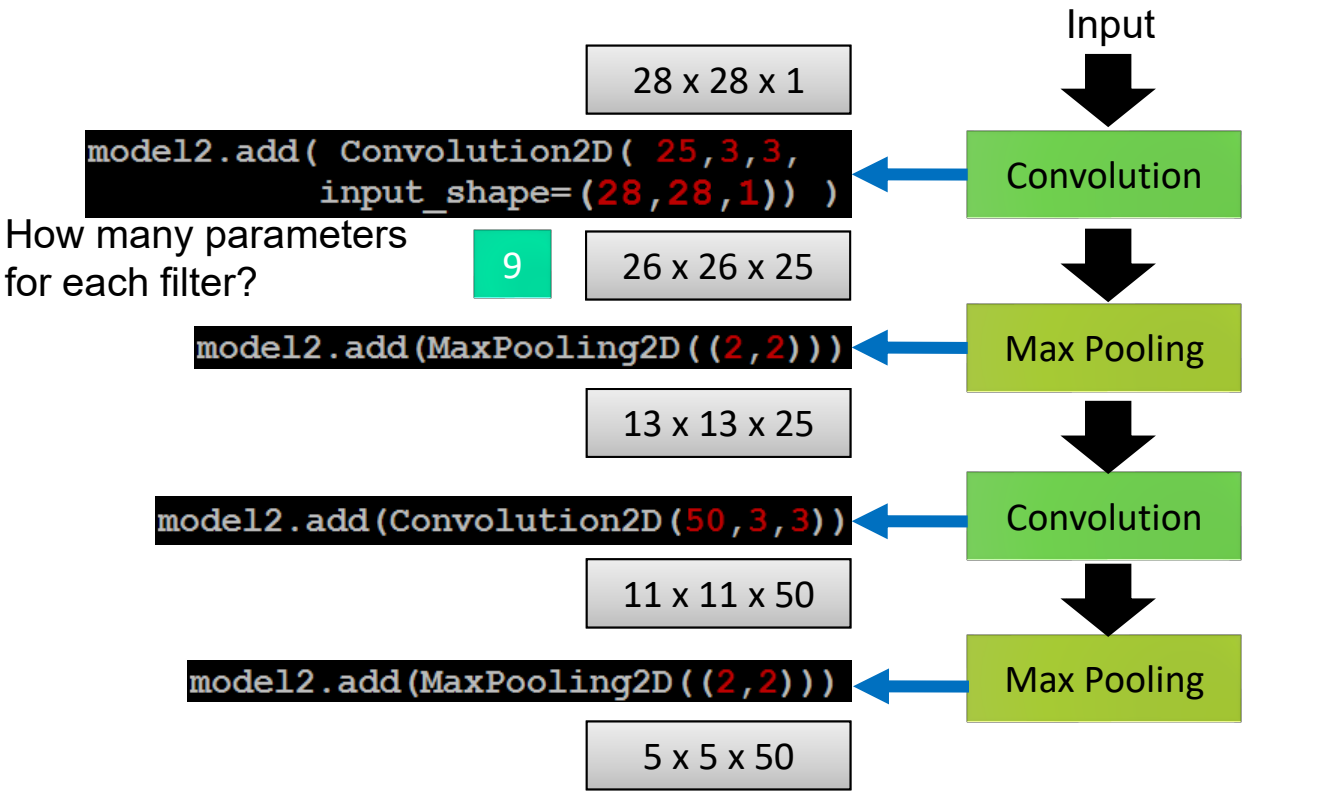
Implementation in TensorFlow Keras

USING SEQUENTIAL MODEL (CAN ALSO BE ADAPTED TO FUNCTIONAL API)

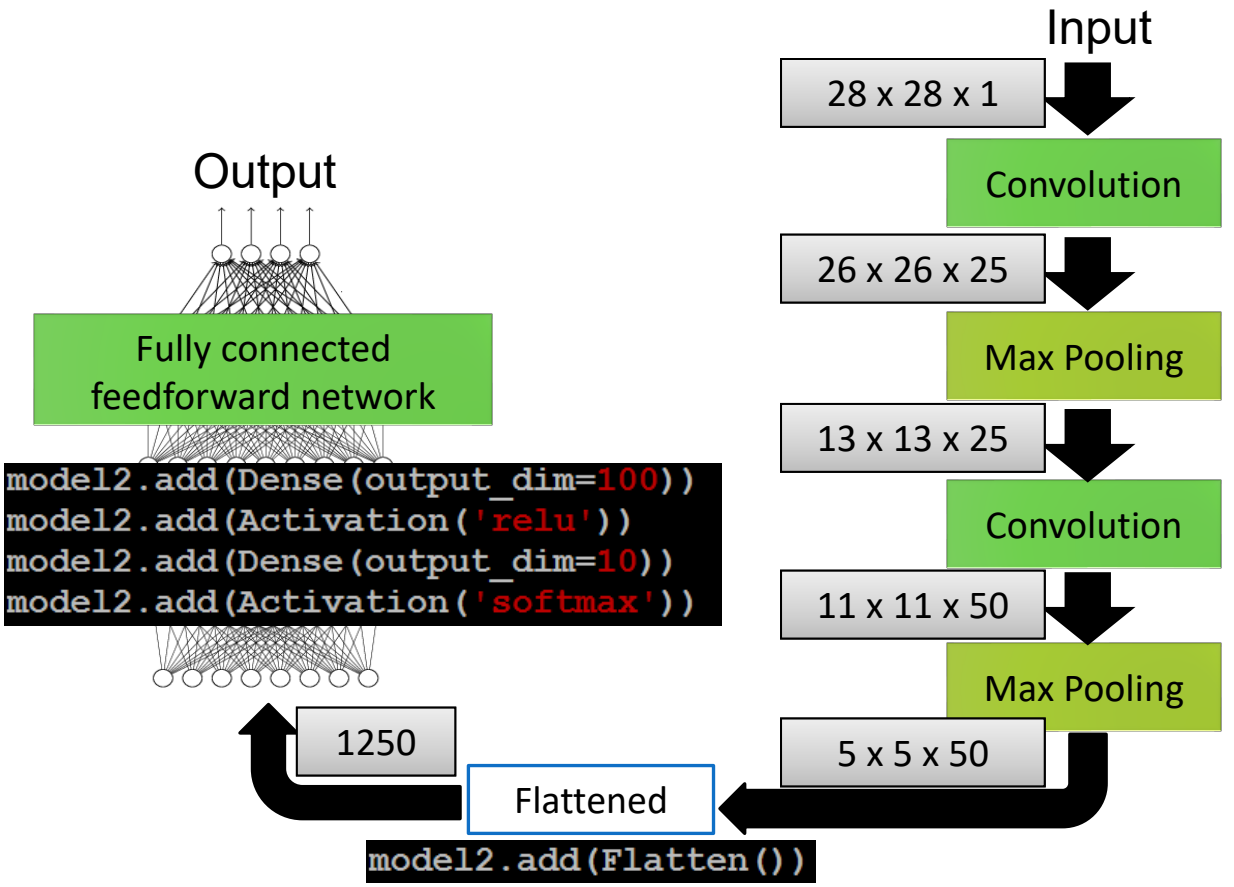
Only modified the ***network structure*** and ***input format (vector -> 3-D tensor)***



Only modified the ***network structure*** and ***input format (vector -> 3-D array)***

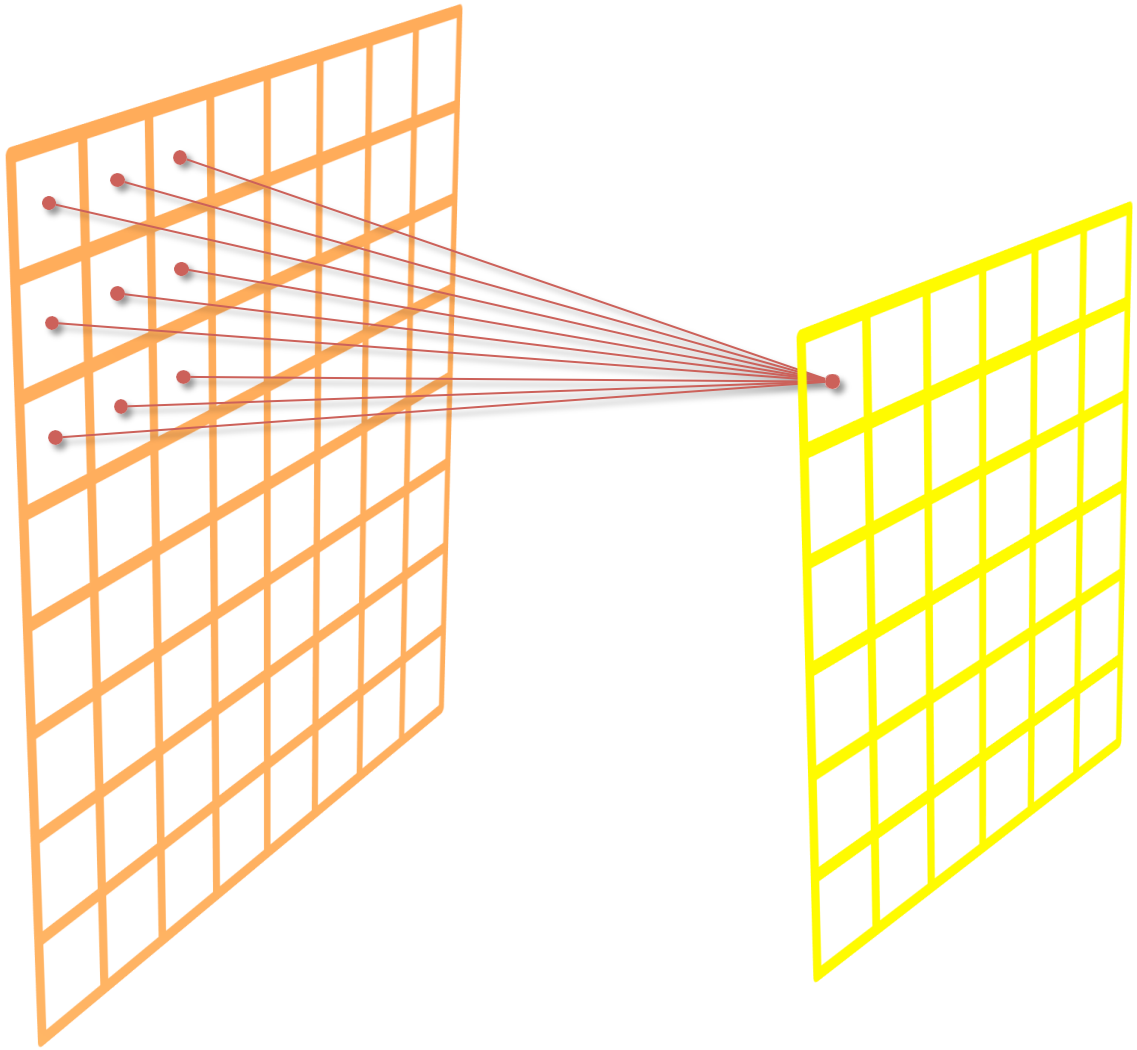


Only modified the ***network structure*** and ***input format (vector -> 3-D array)***

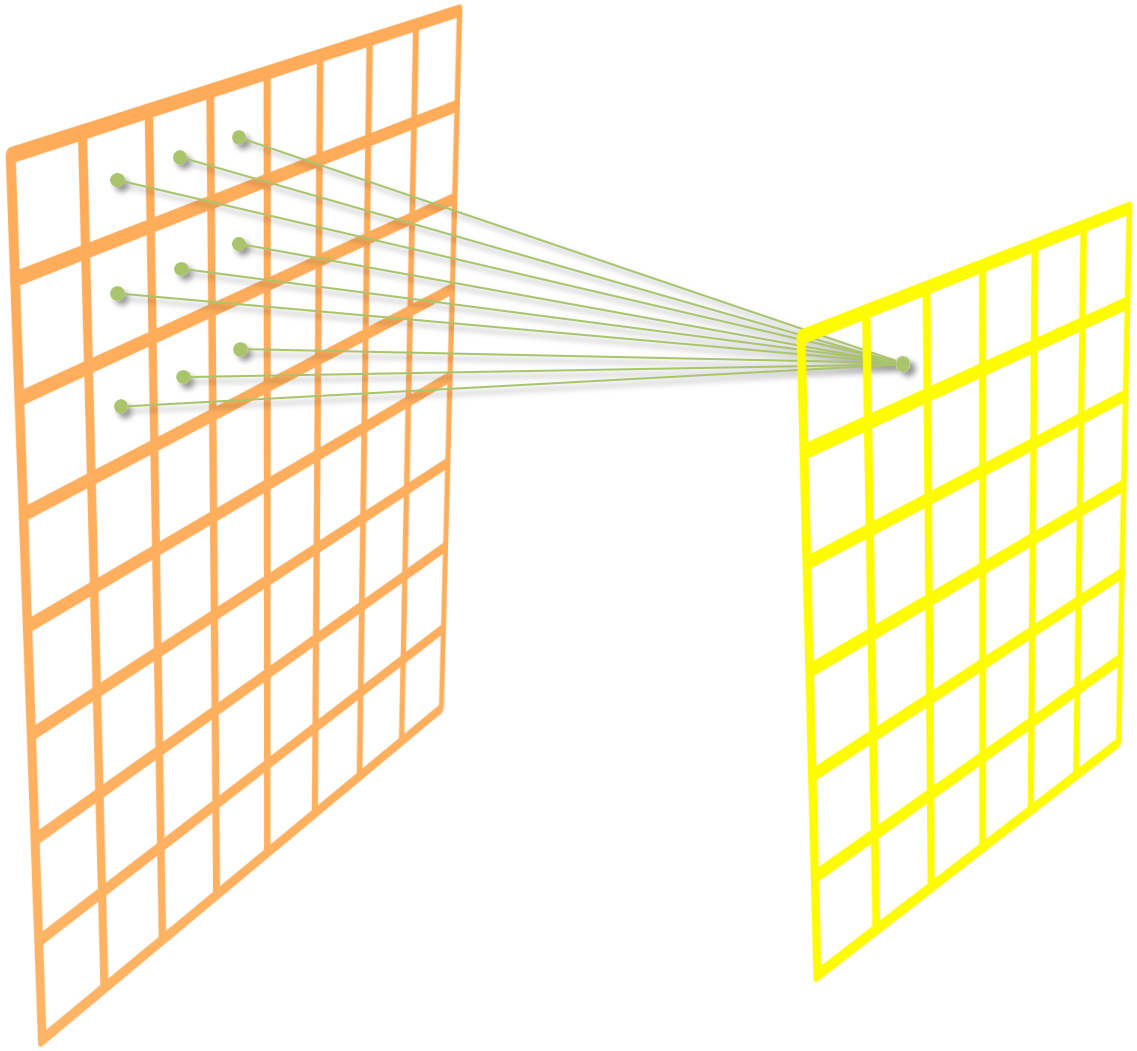


Weights Sharing in CNNs

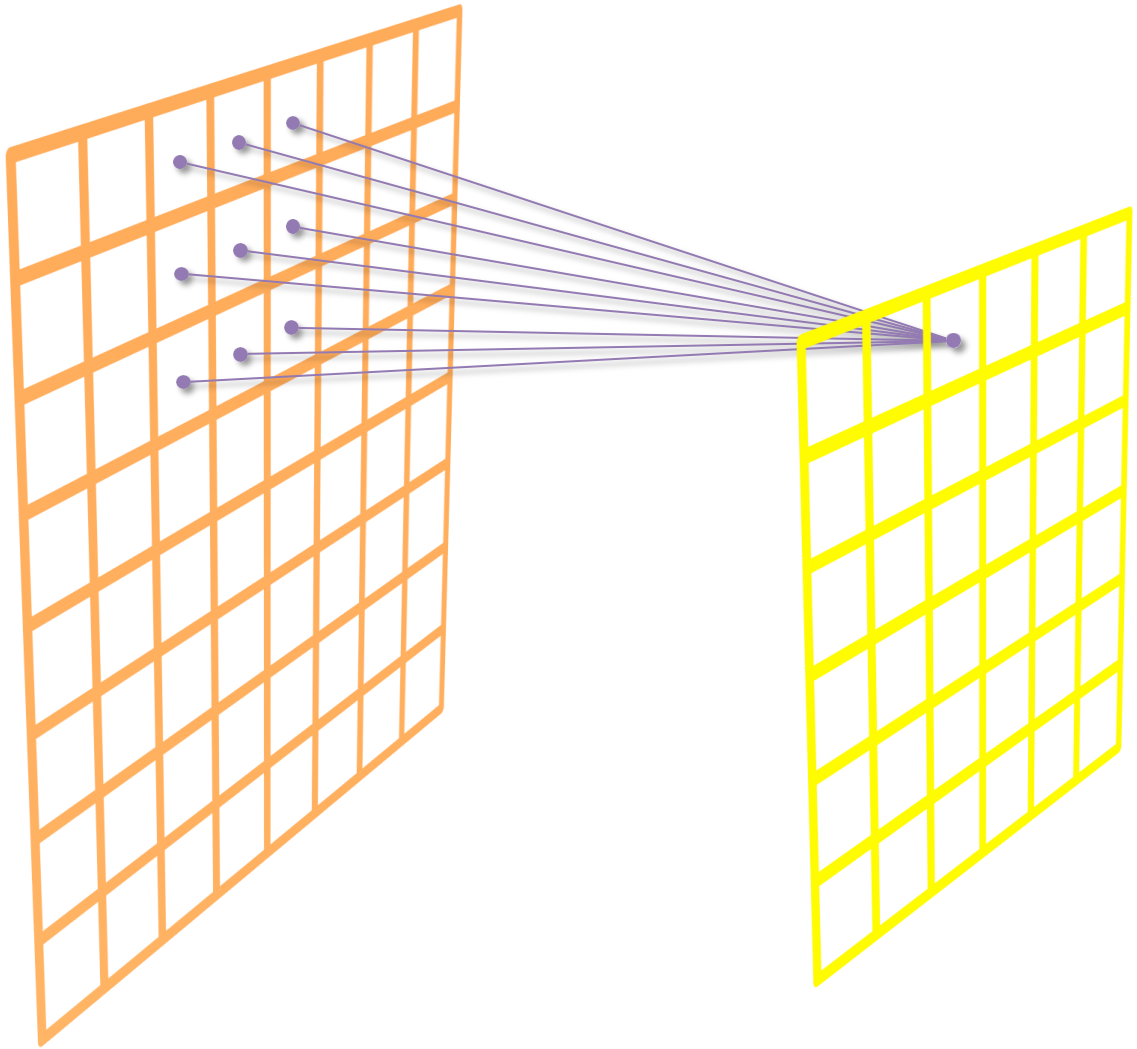
8x8 image, 3x3 filter, Stride 1



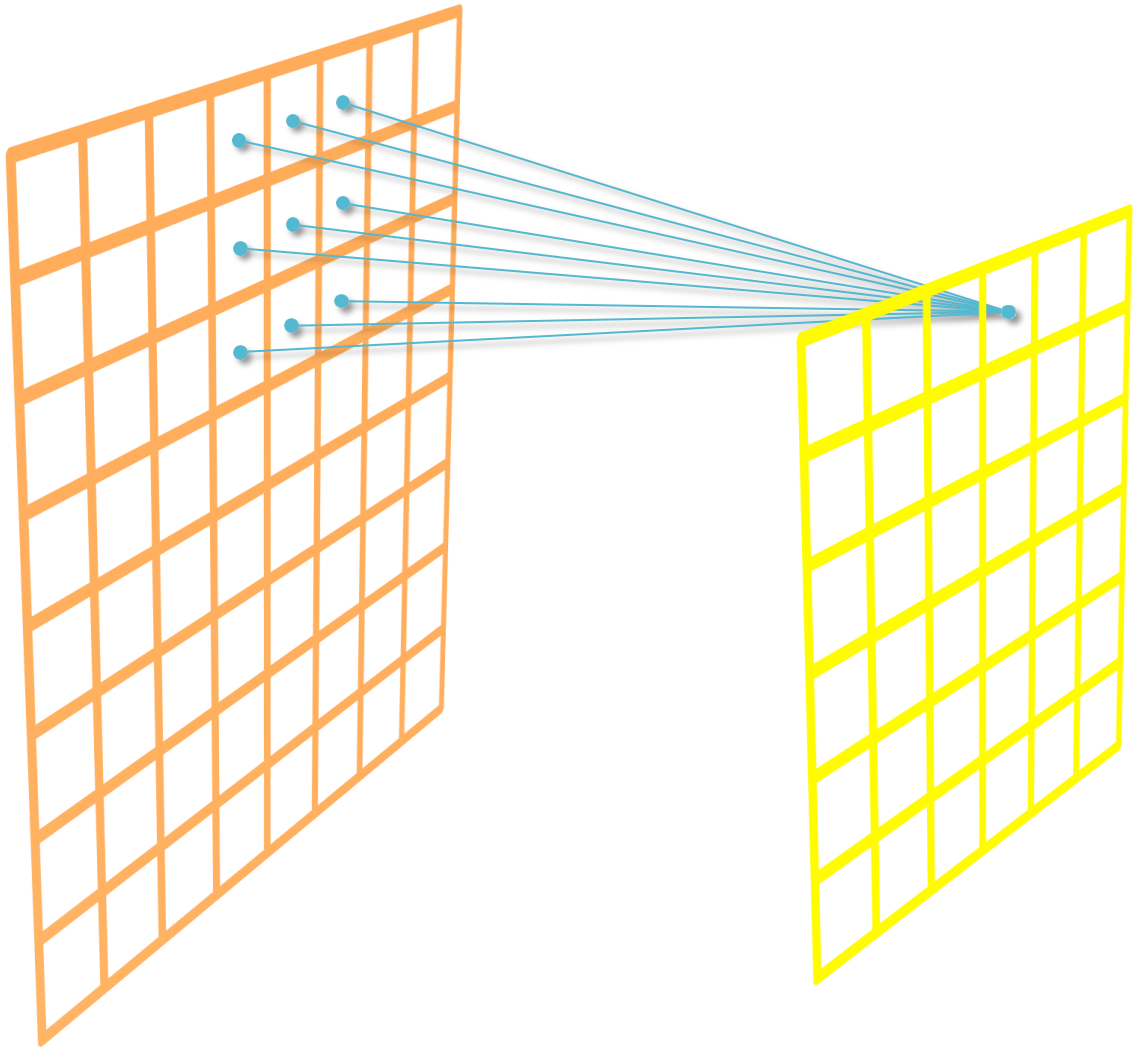
8x8 image, 3x3 filter, Stride 1



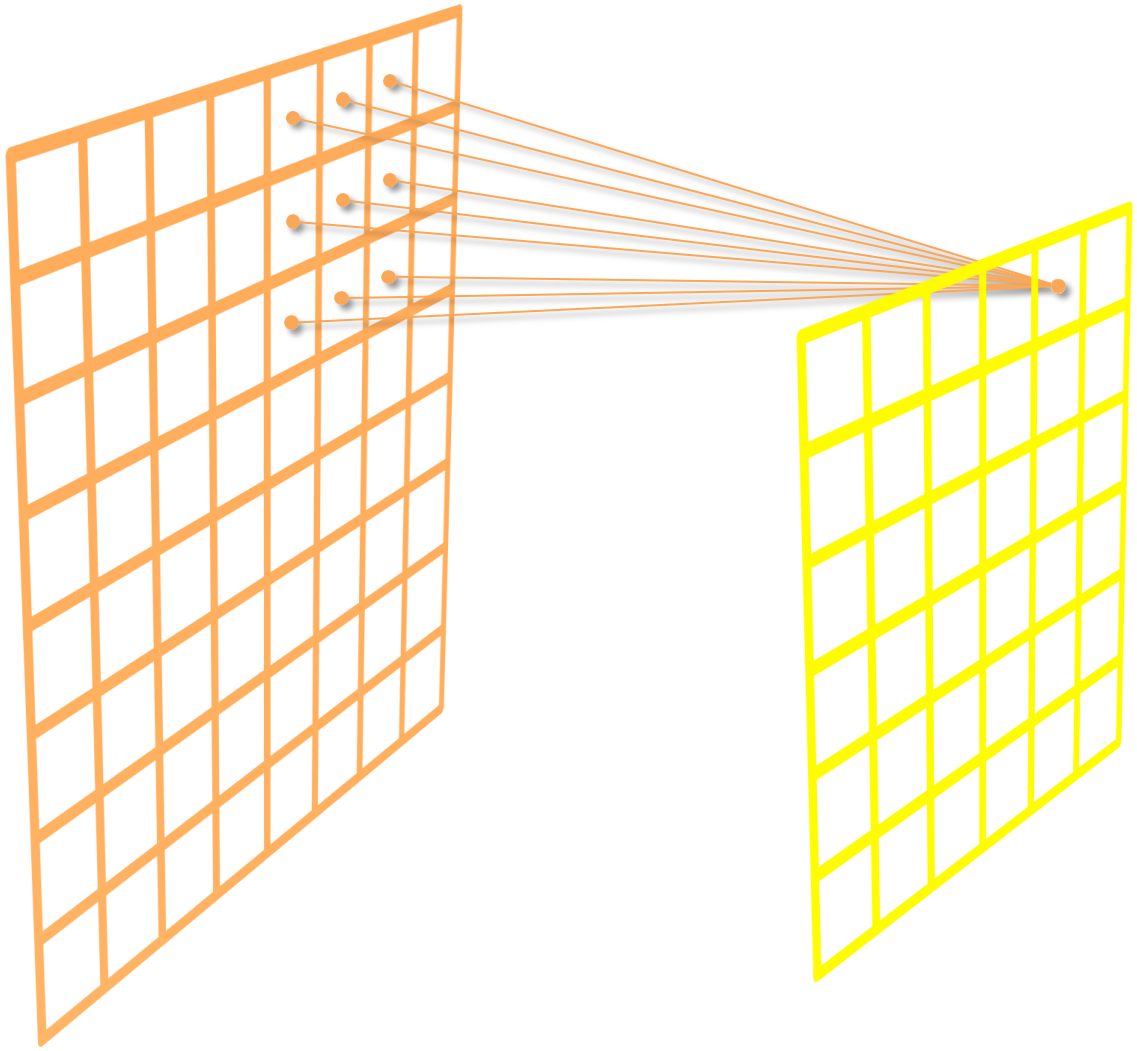
8x8 image, 3x3 filter, Stride 1



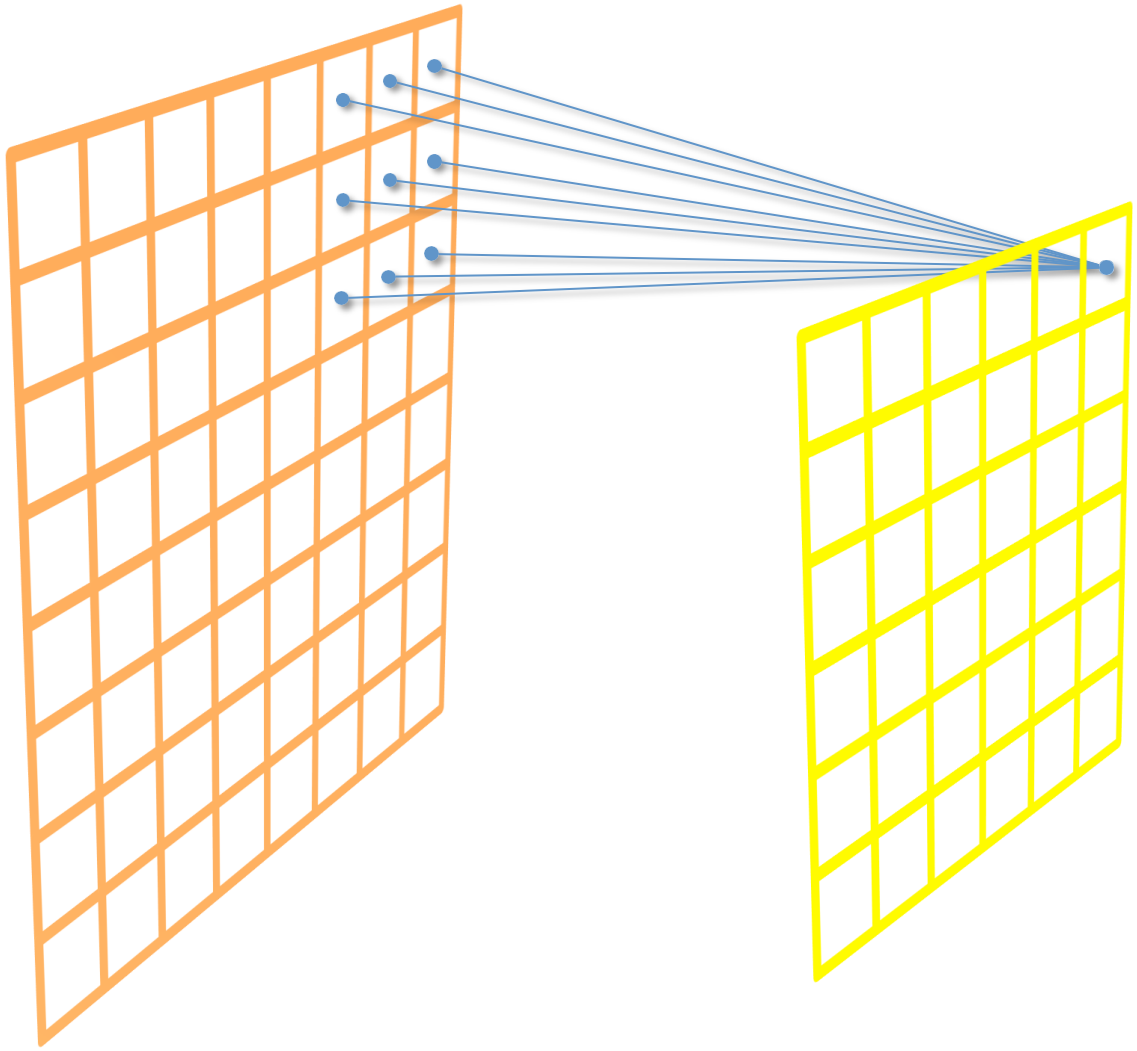
8x8 image, 3x3 filter, Stride 1



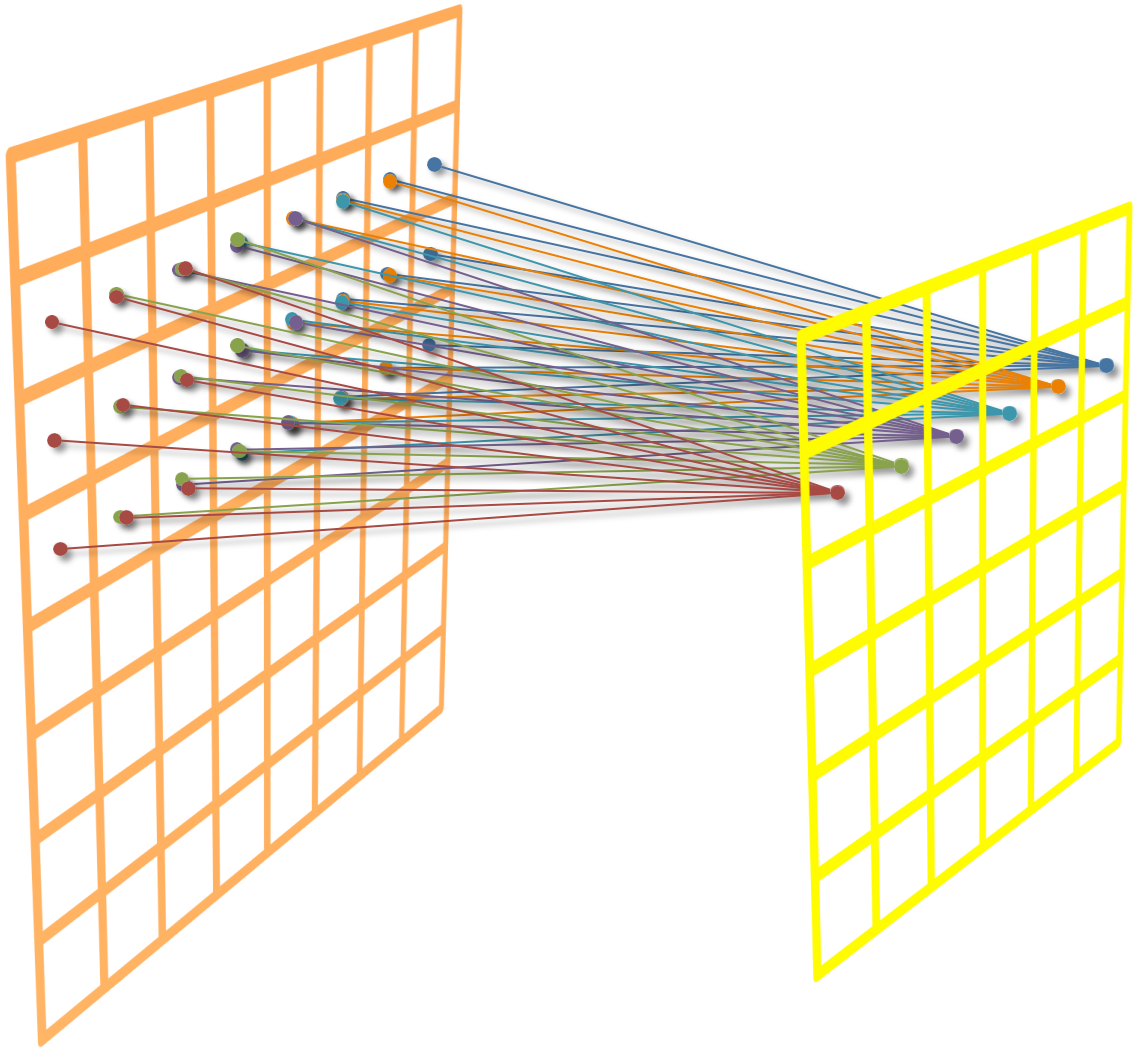
8x8 image, 3x3 filter, Stride 1



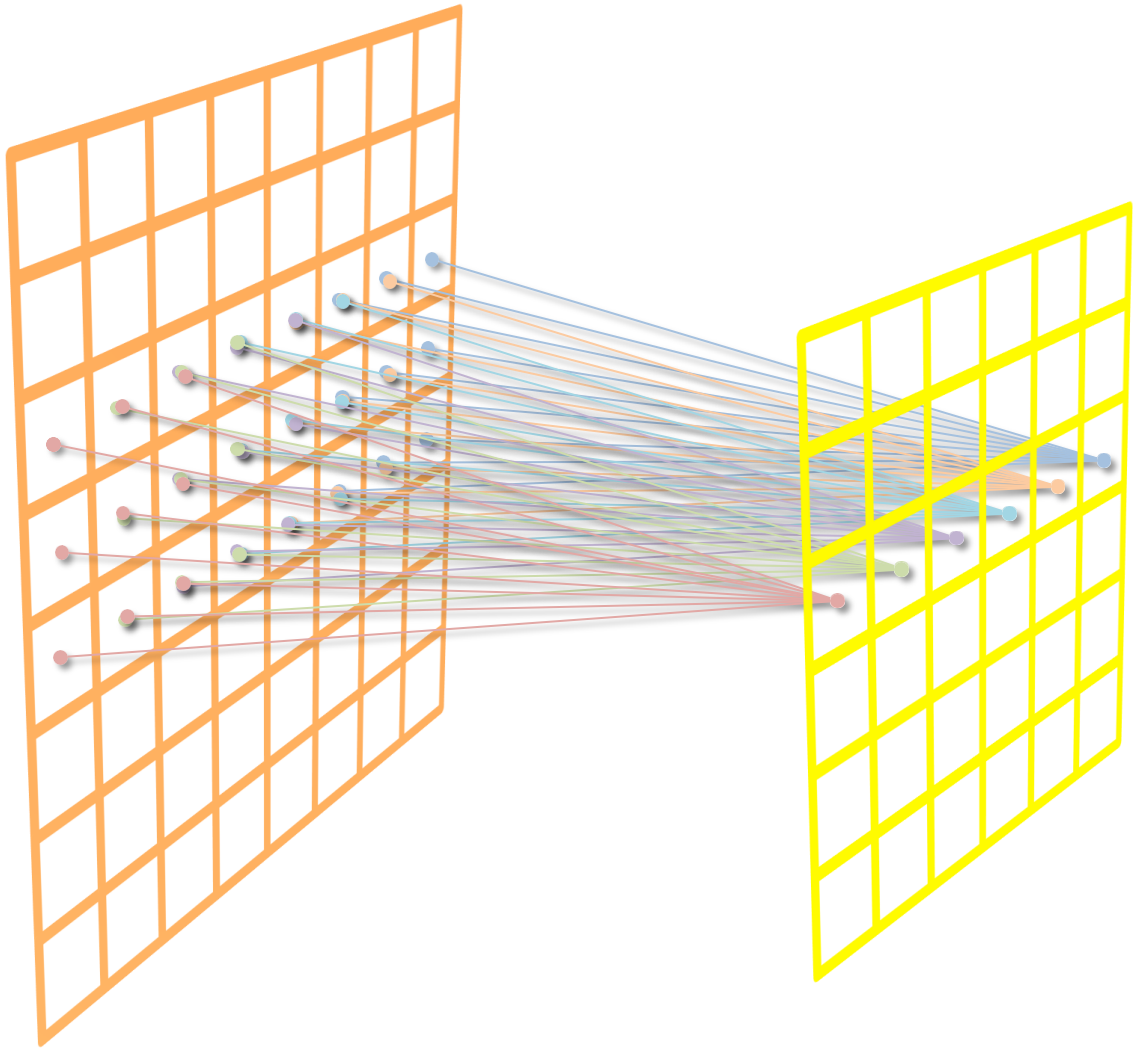
8x8 image, 3x3 filter, Stride 1

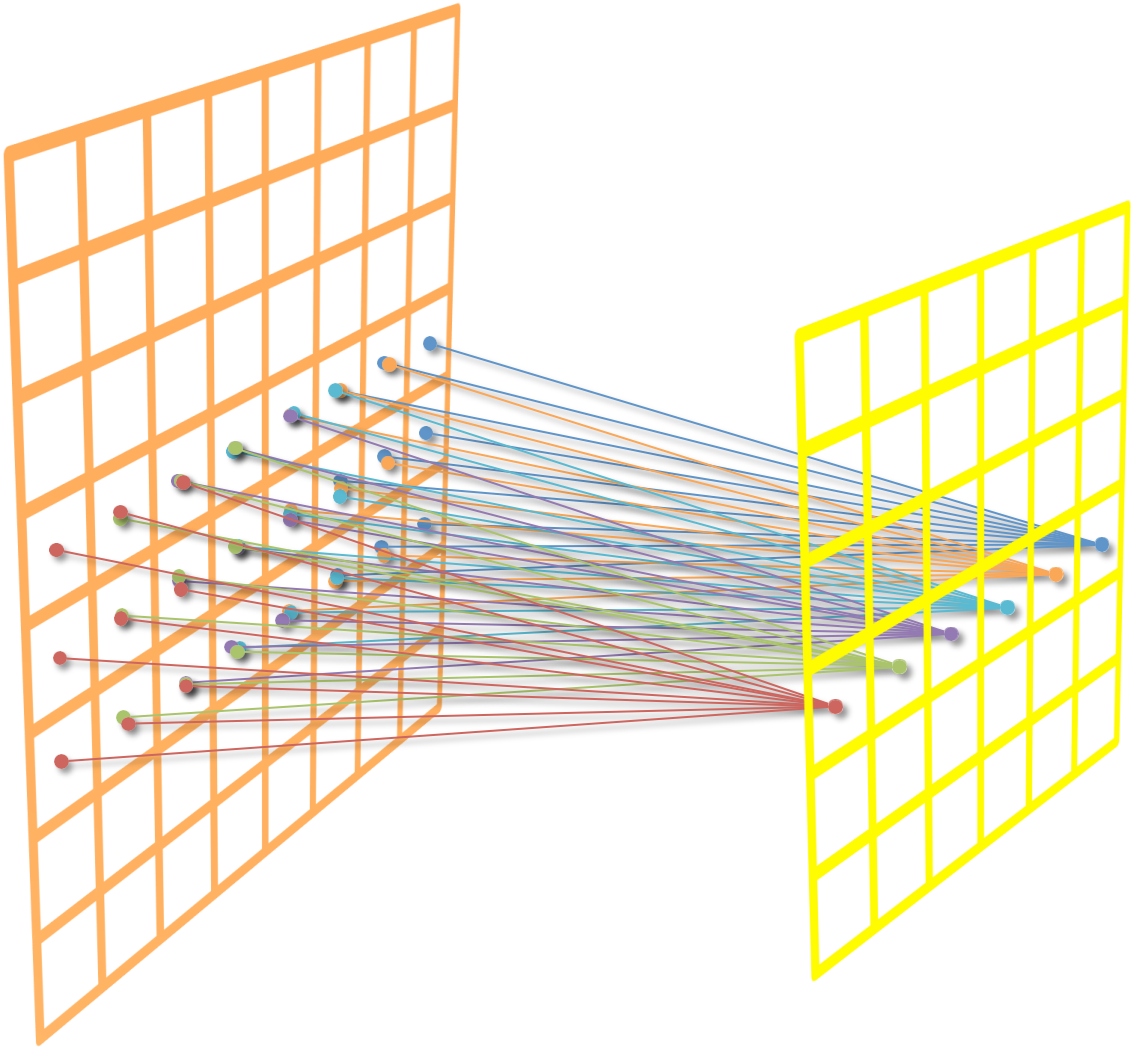


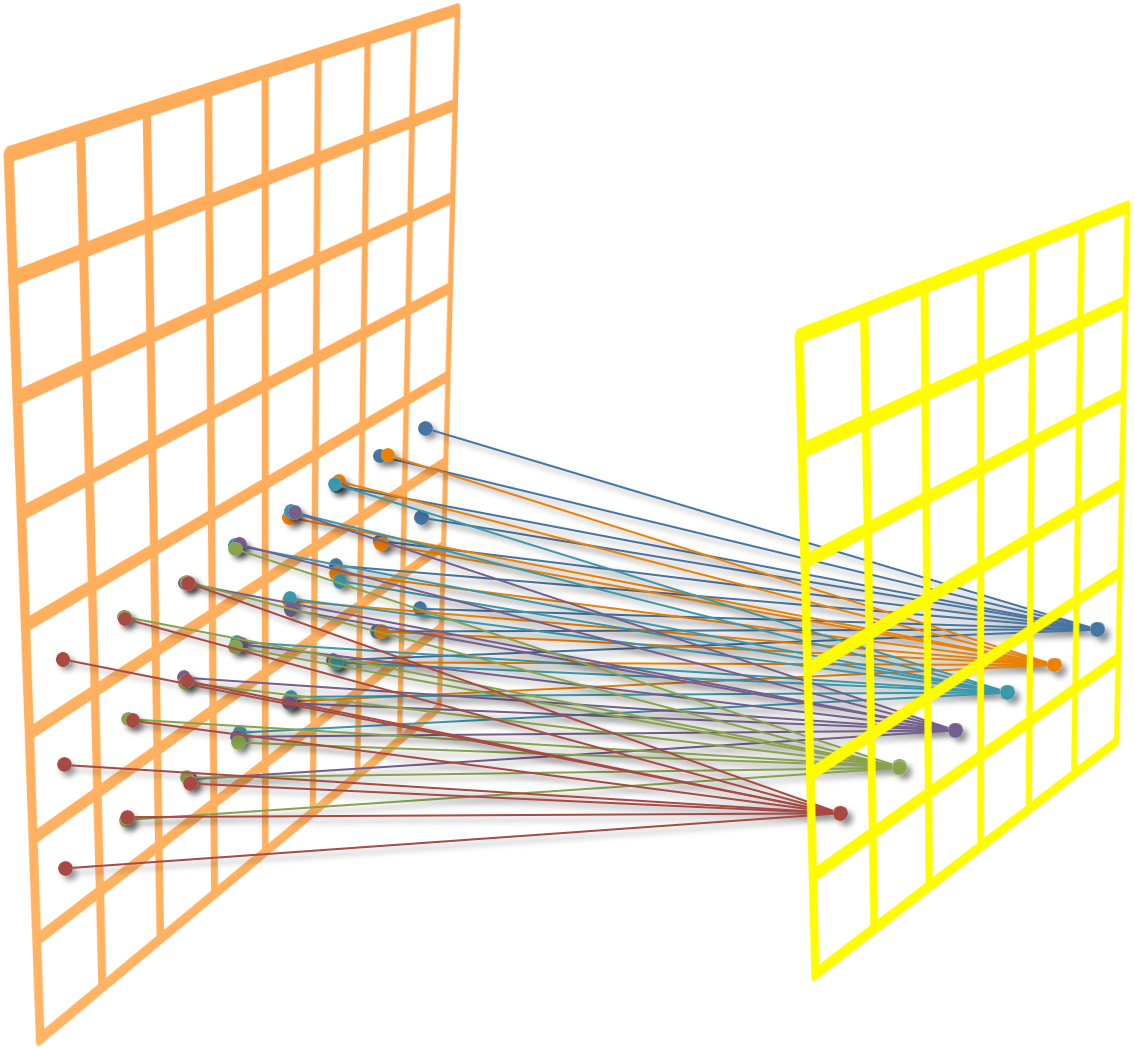
8x8 image, 3x3 filter, Stride 1



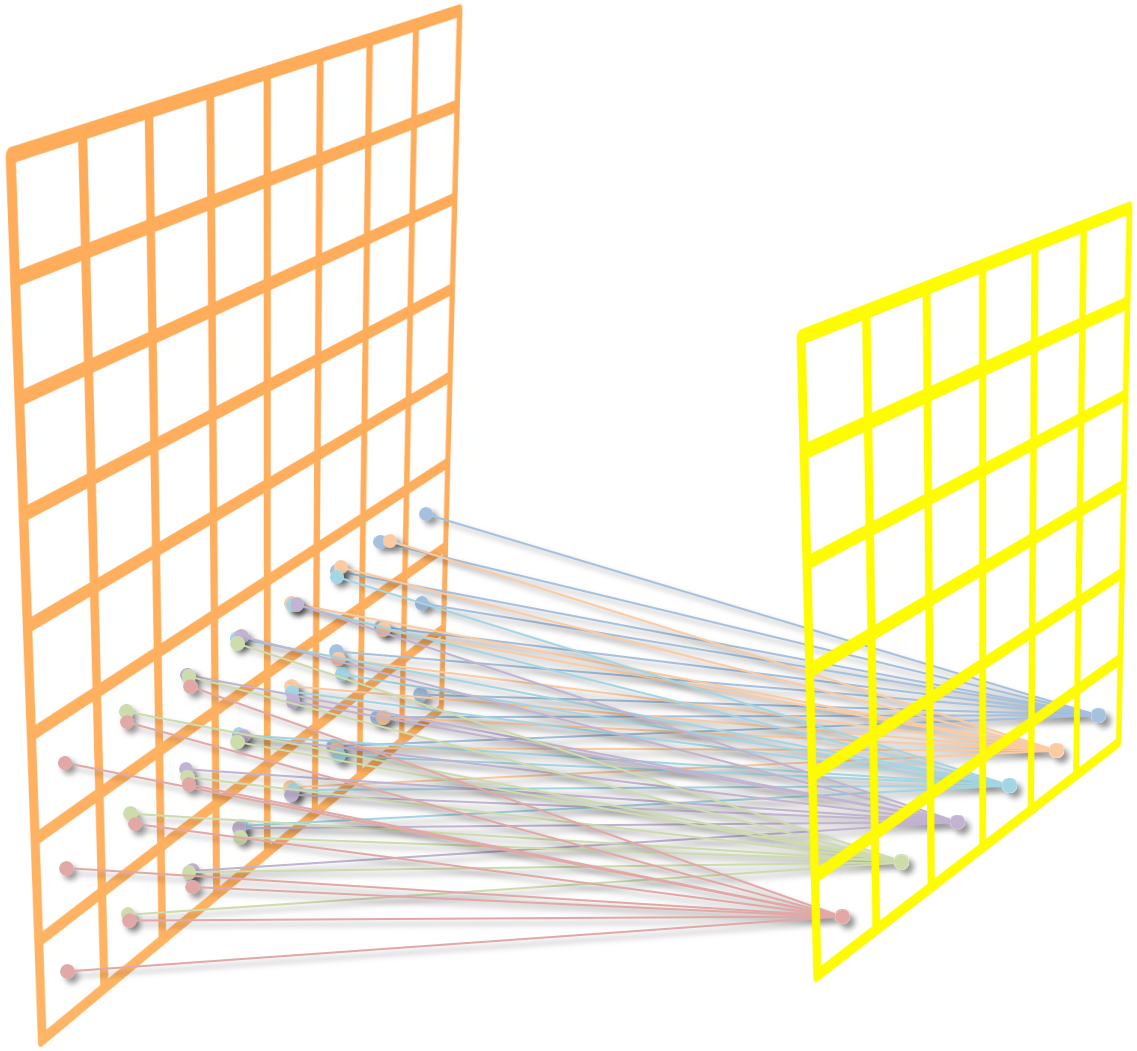
8x8 image, 3x3 filter, Stride 1



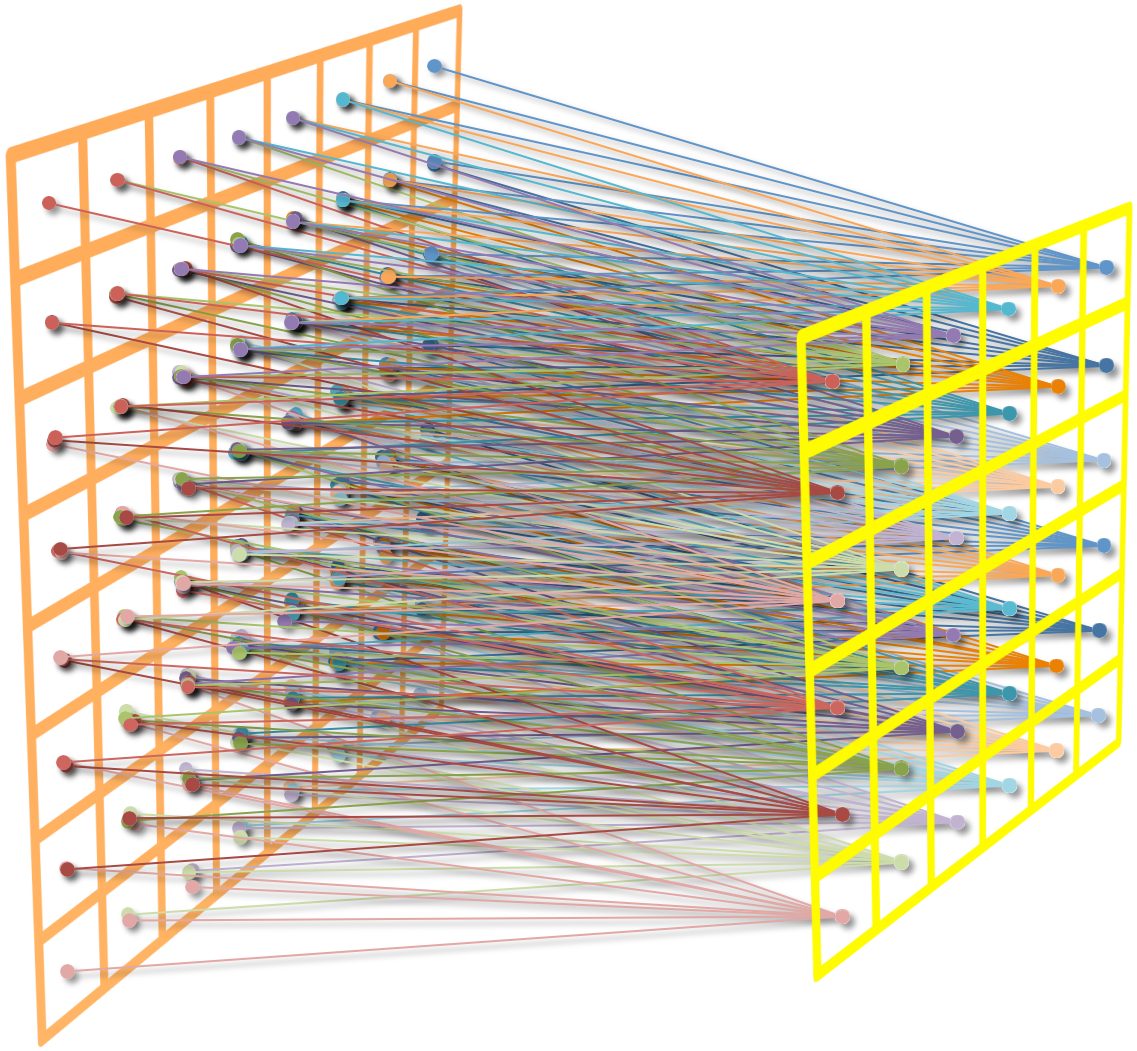




8x8 image, 3x3 filter, Stride 1



8x8 image, 3x3 filter, Stride 1



Total Connections?

6x6x1 x 3x3x1

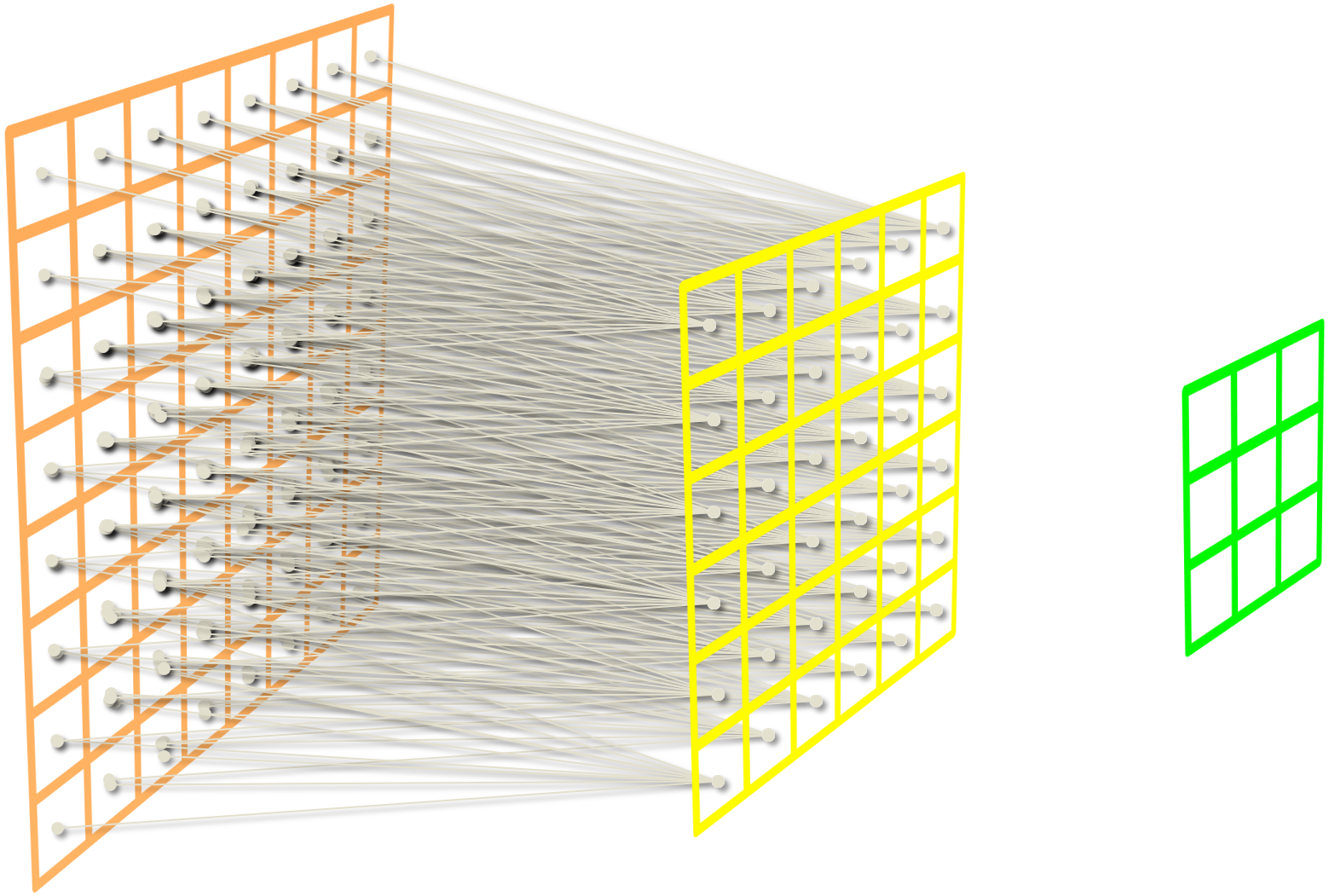
What would have been the number of connections if this had been a fully-connected layer?

Total Unique Parameters?

3x3x1
+ bias

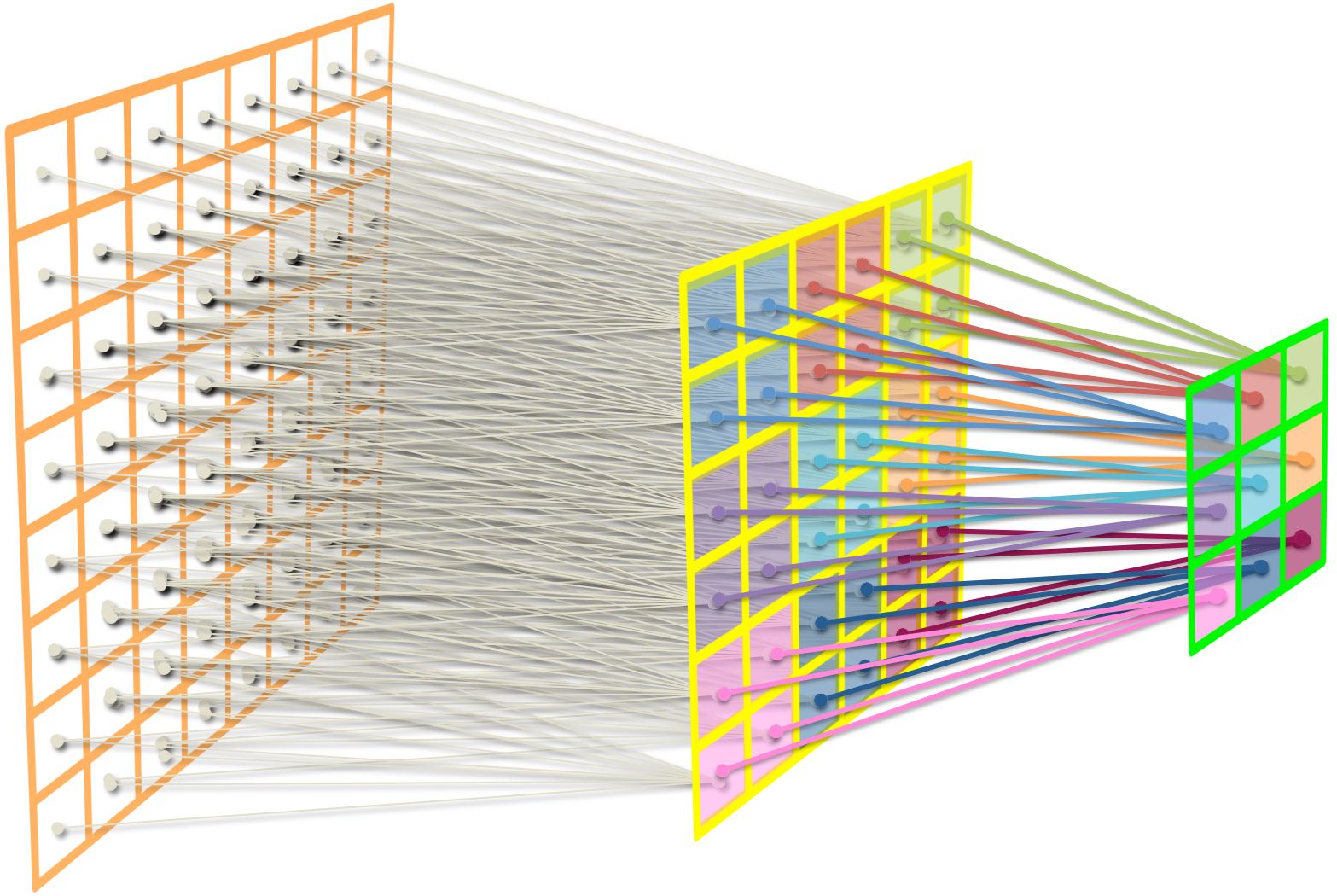
8x8 image, 3x3 filter, Stride 1

2x2 pooling, Stride 2



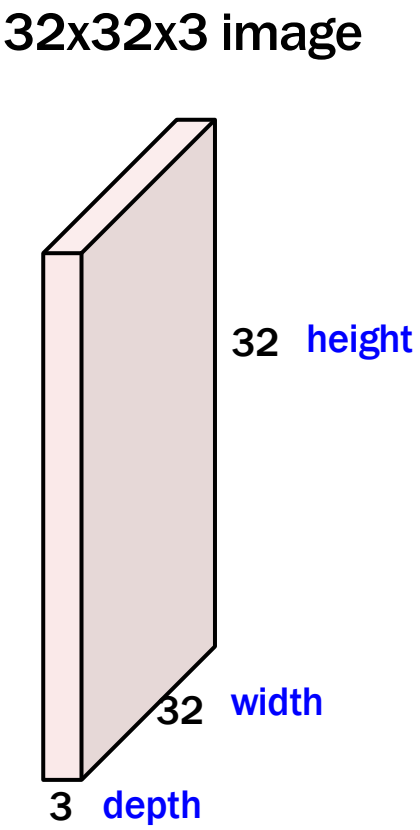
8x8 image, 3x3 filter, Stride 1

2x2 pooling, Stride 2



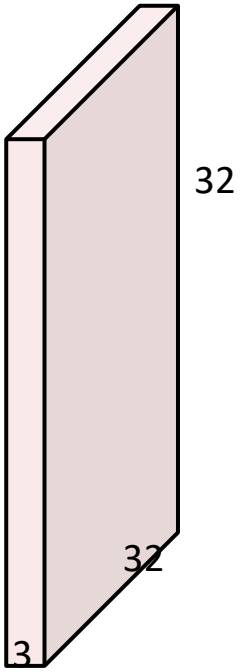
How Color Channels are Handled?

Convolution Layer

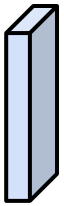


Convolution Layer

32x32x3 image

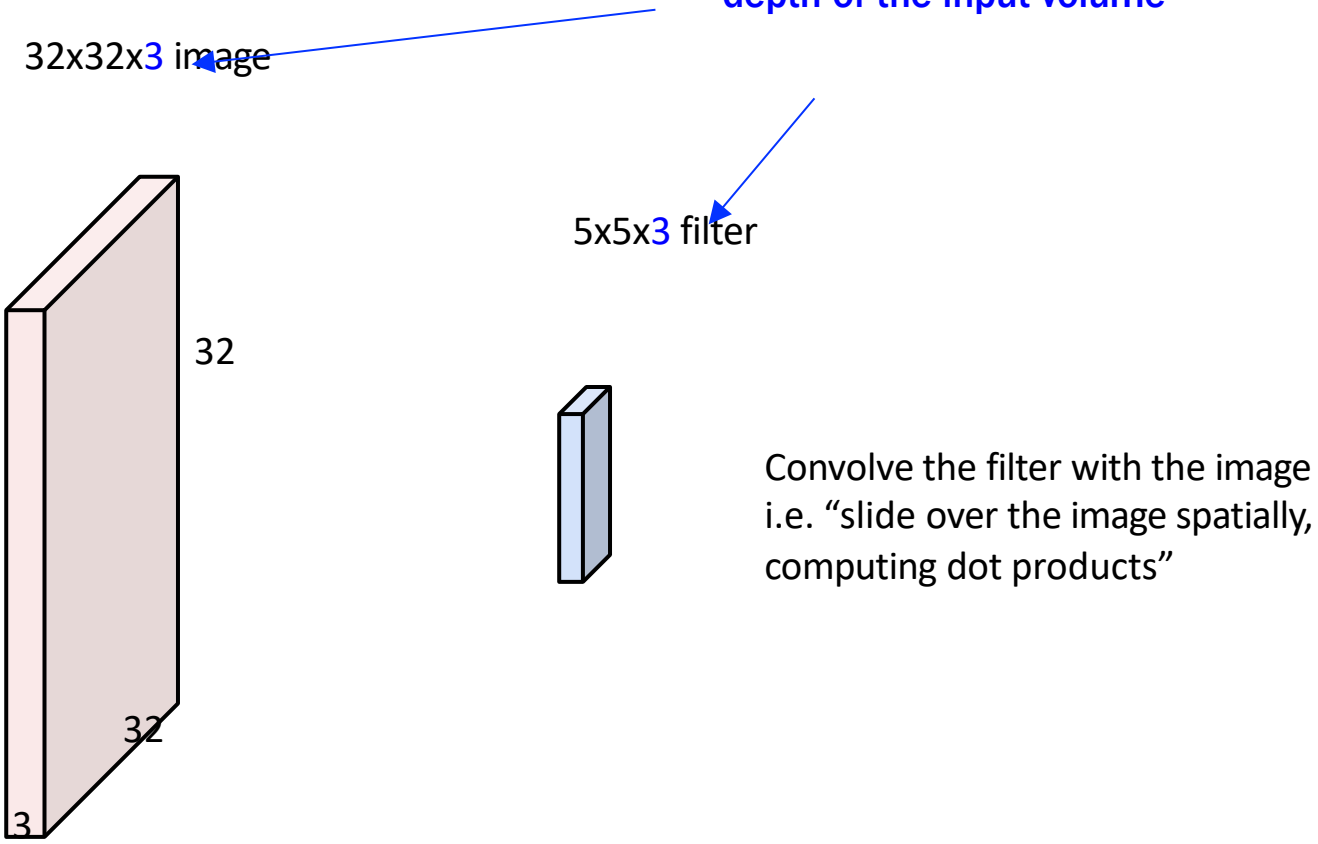


5x5x3 filter

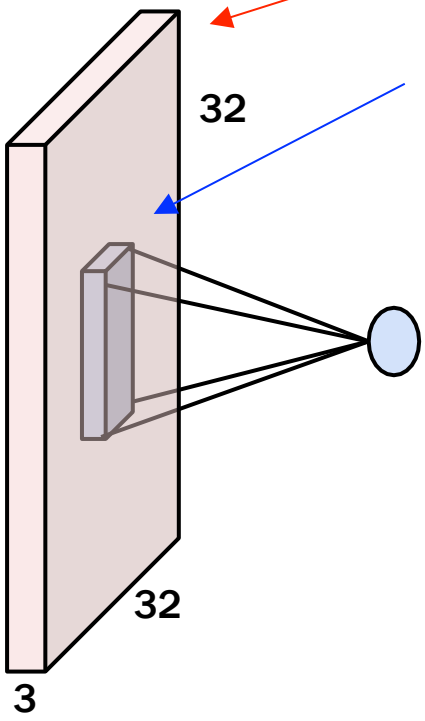


Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

Convolution Layer



Convolution Layer



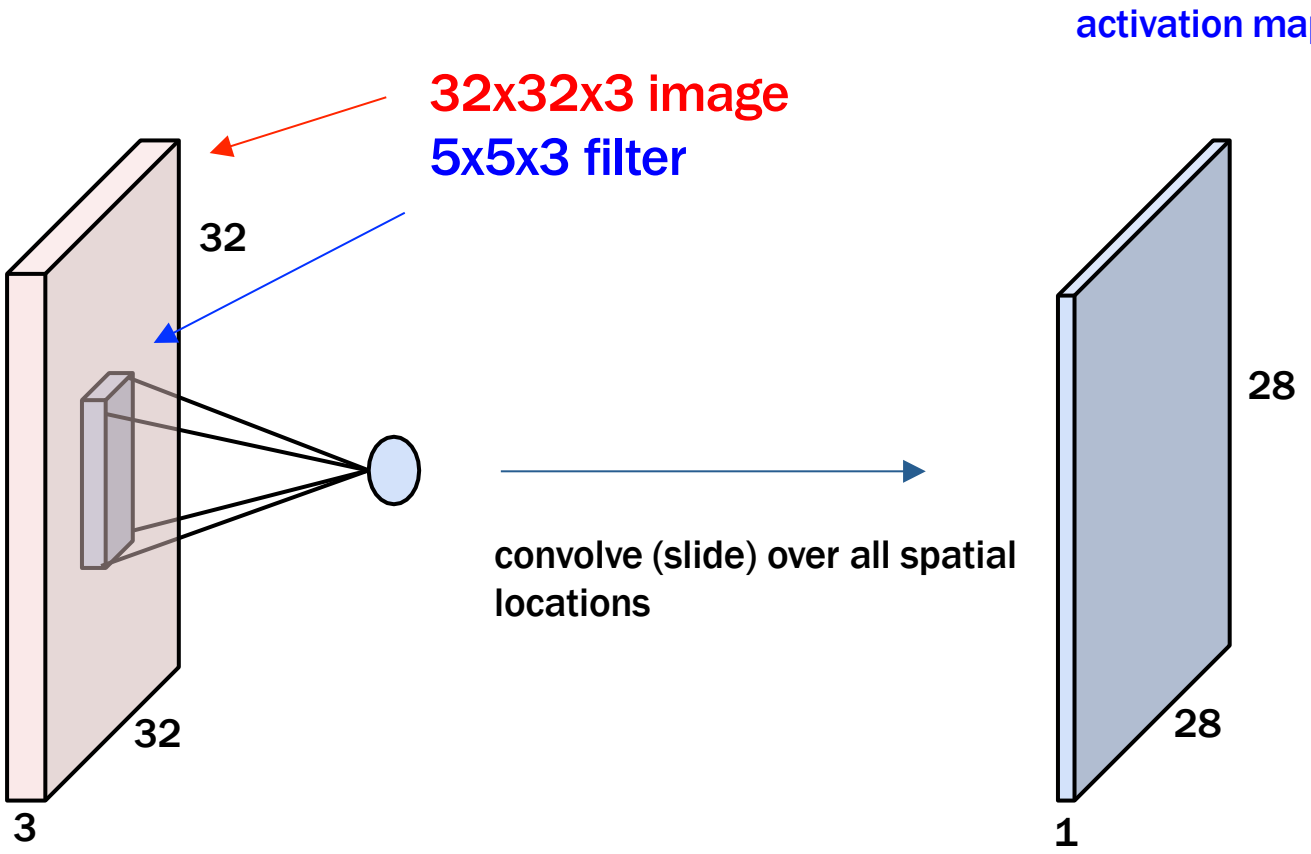
32x32x3 image

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*5*3 = 75-dimensional dot product + bias)

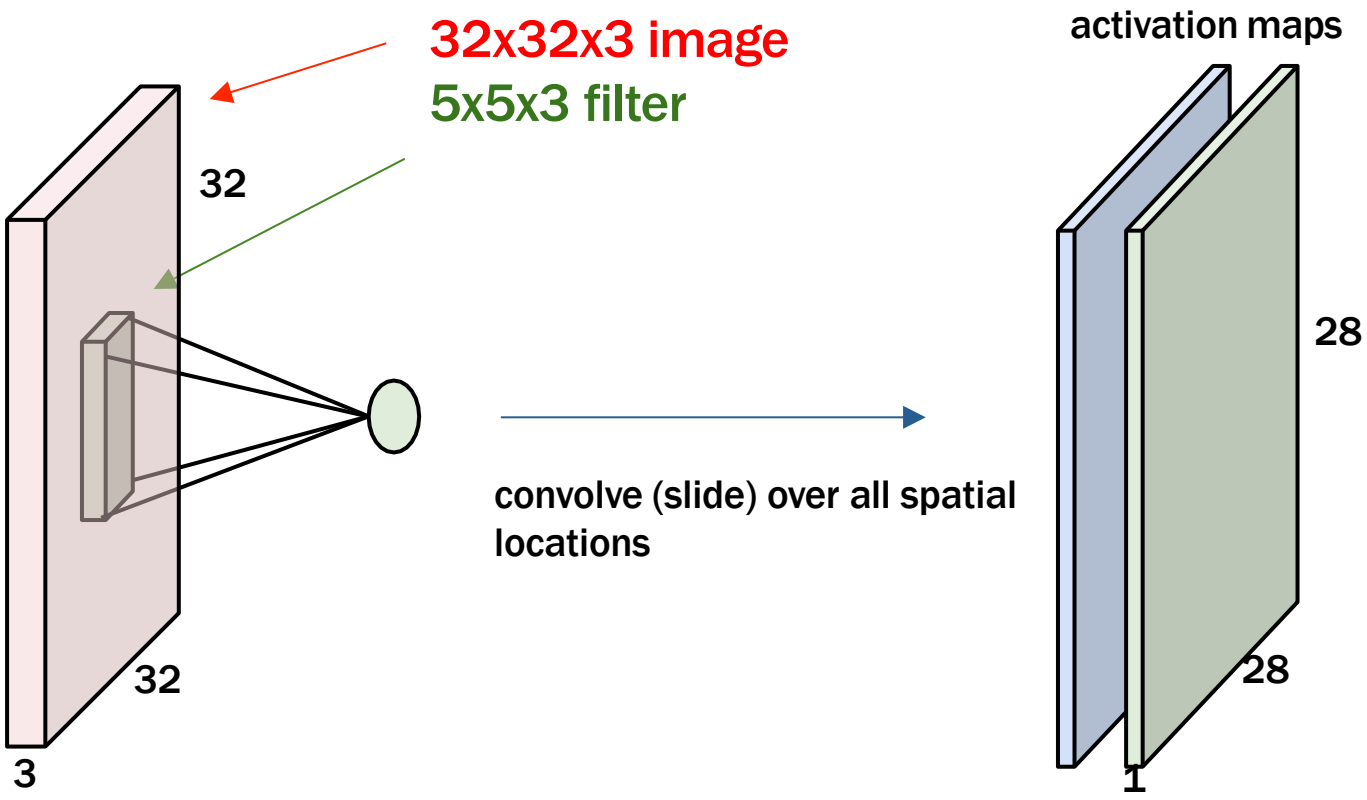
$$w^T x + b$$

Convolution Layer

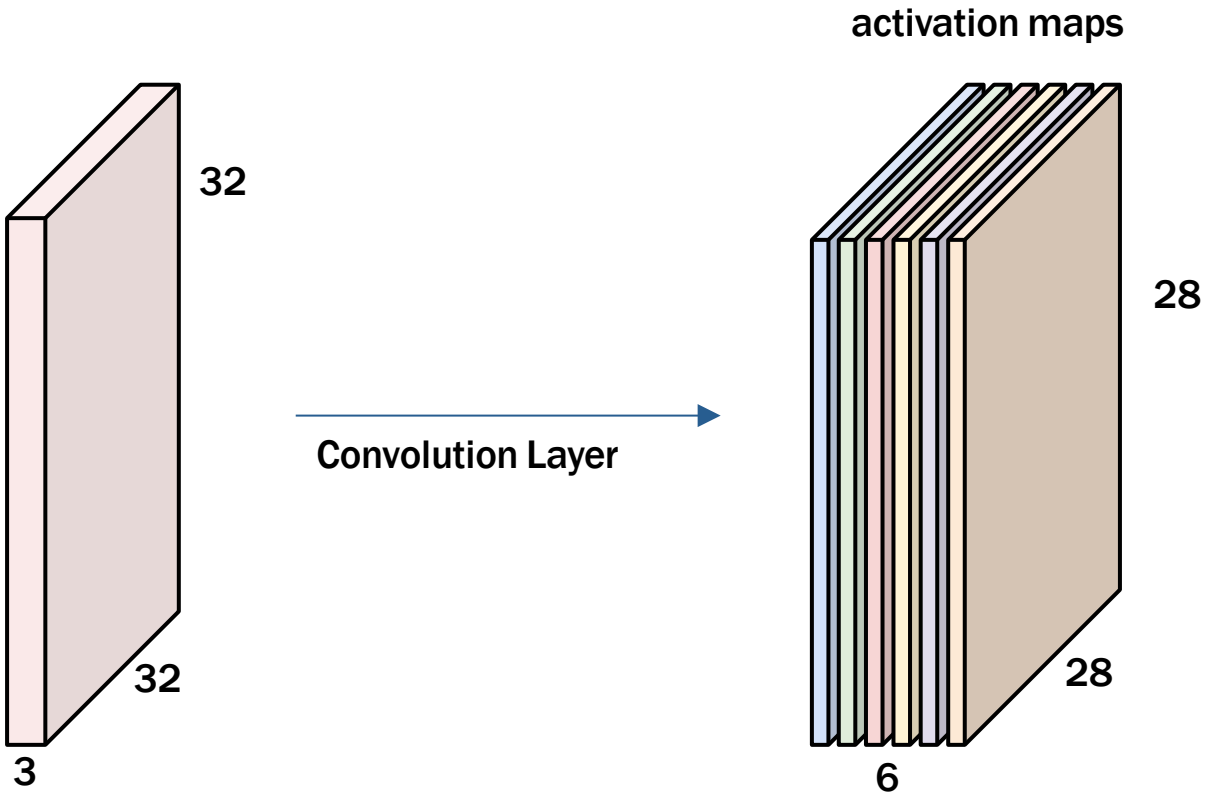


Convolution Layer

consider a second, green filter

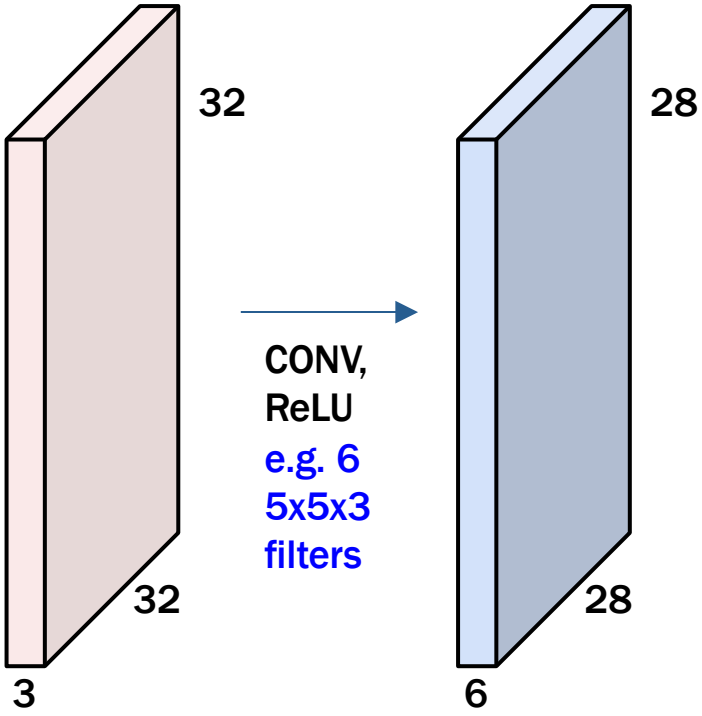


For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:

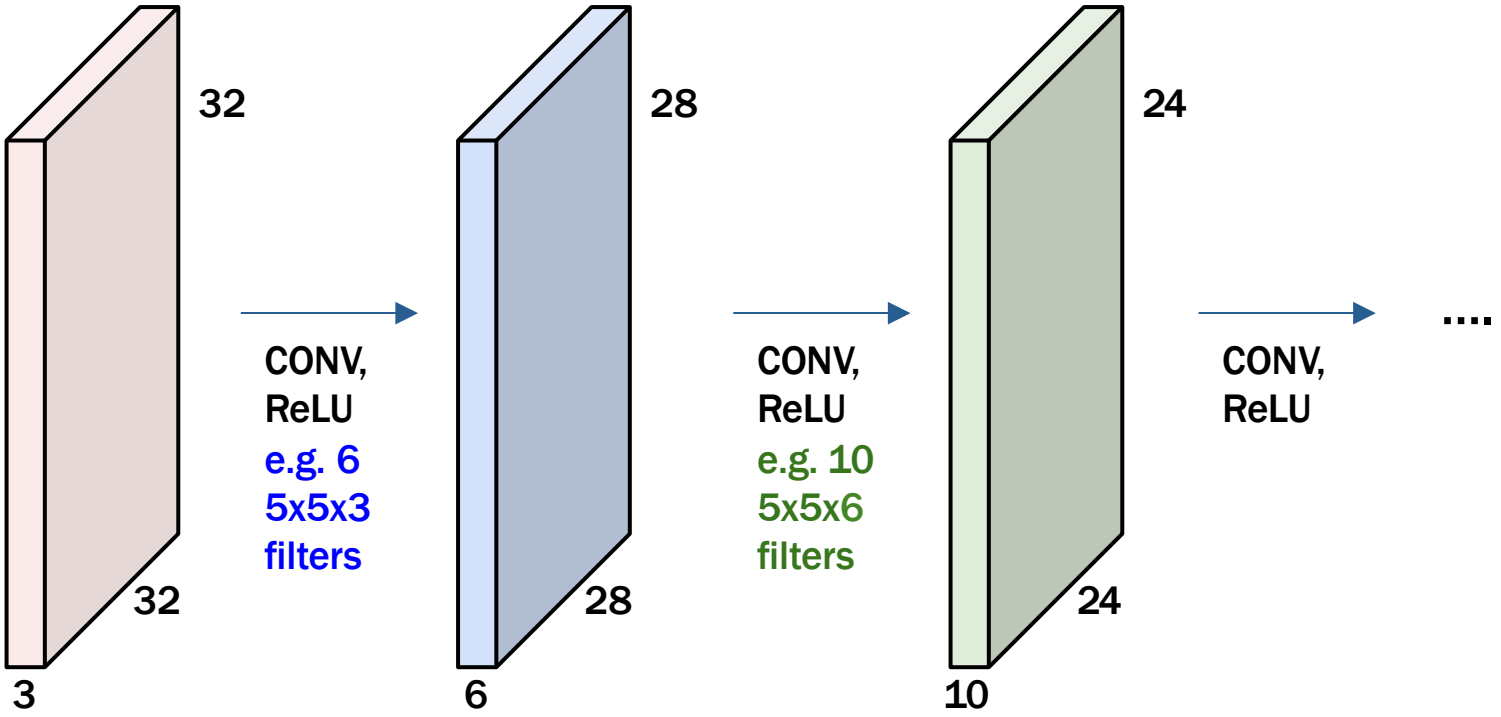


We stack these up to get a "new image" of size 28x28x6!

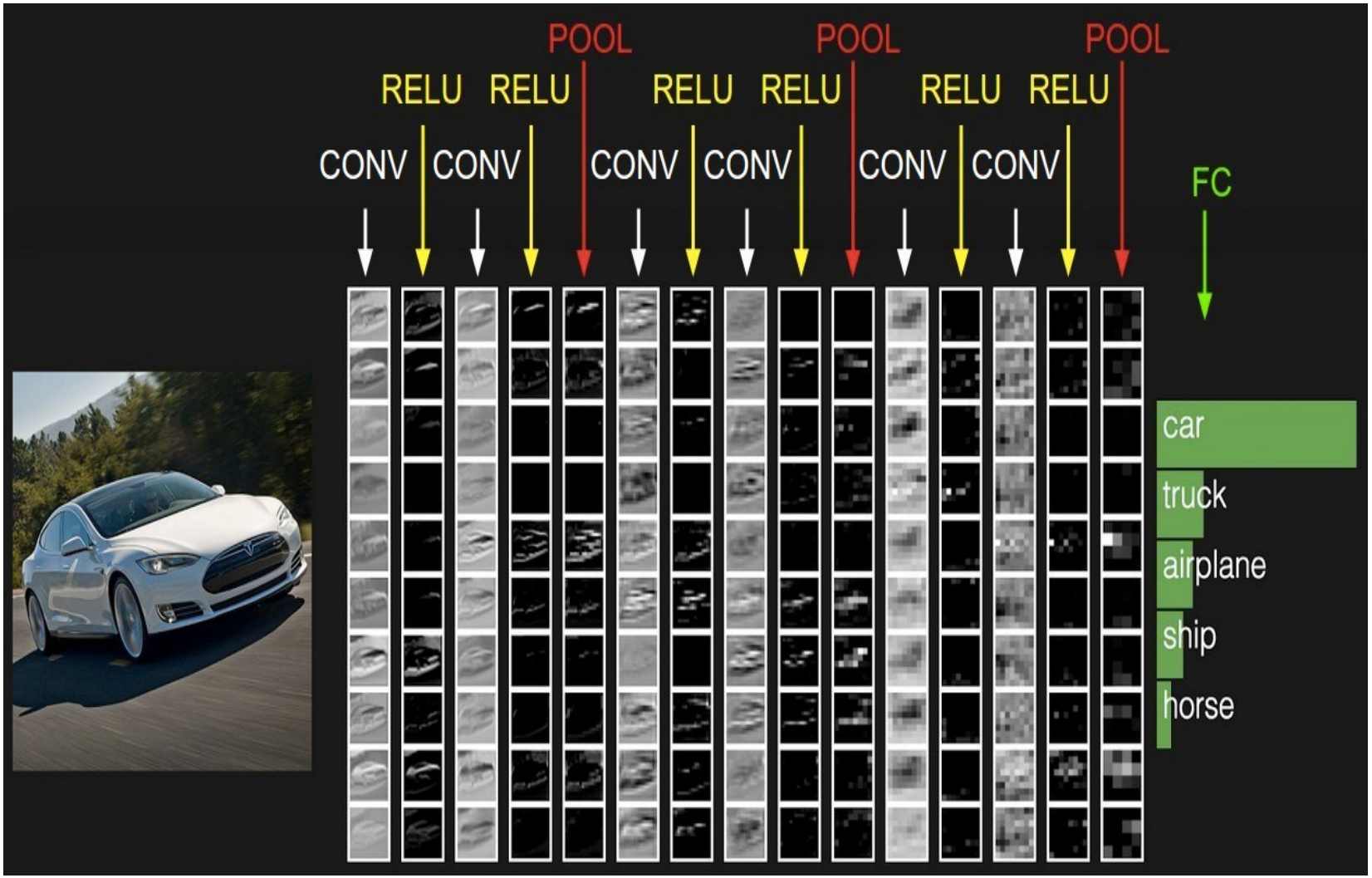
Preview: ConvNet is a sequence of Convolution Layers, interspersed with activation functions



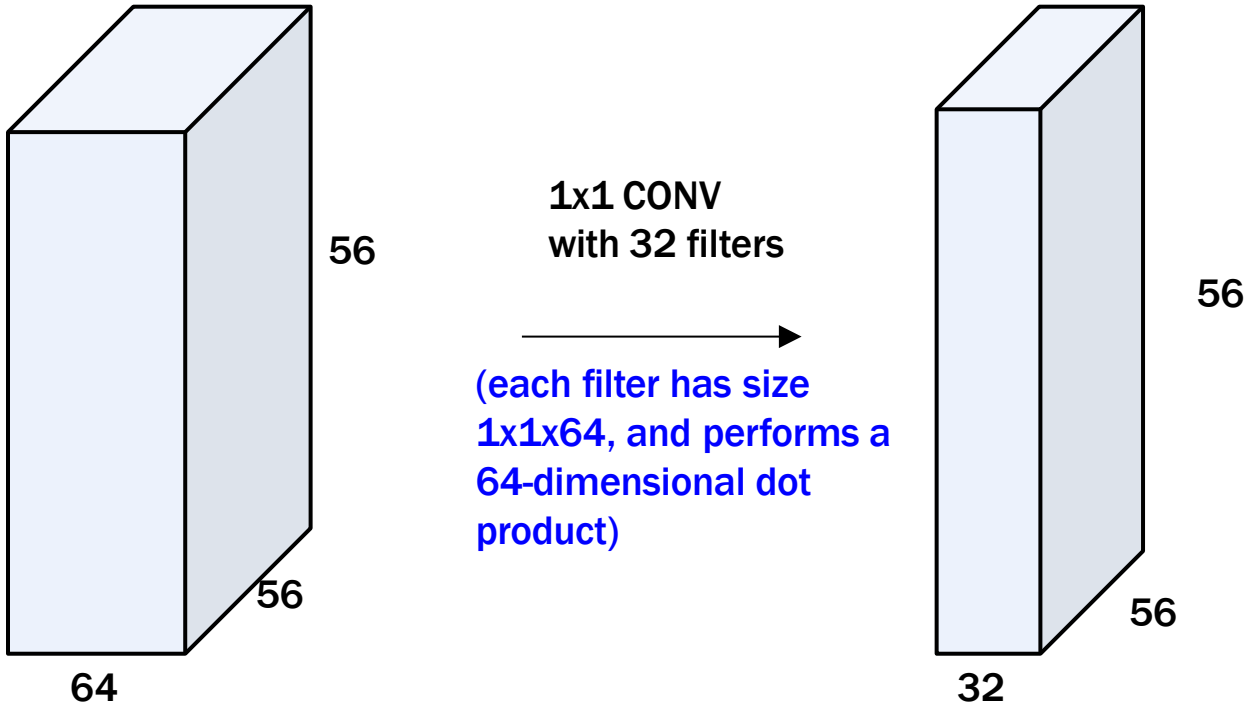
Preview: ConvNet is a sequence of Convolutional Layers, interspersed with activation functions



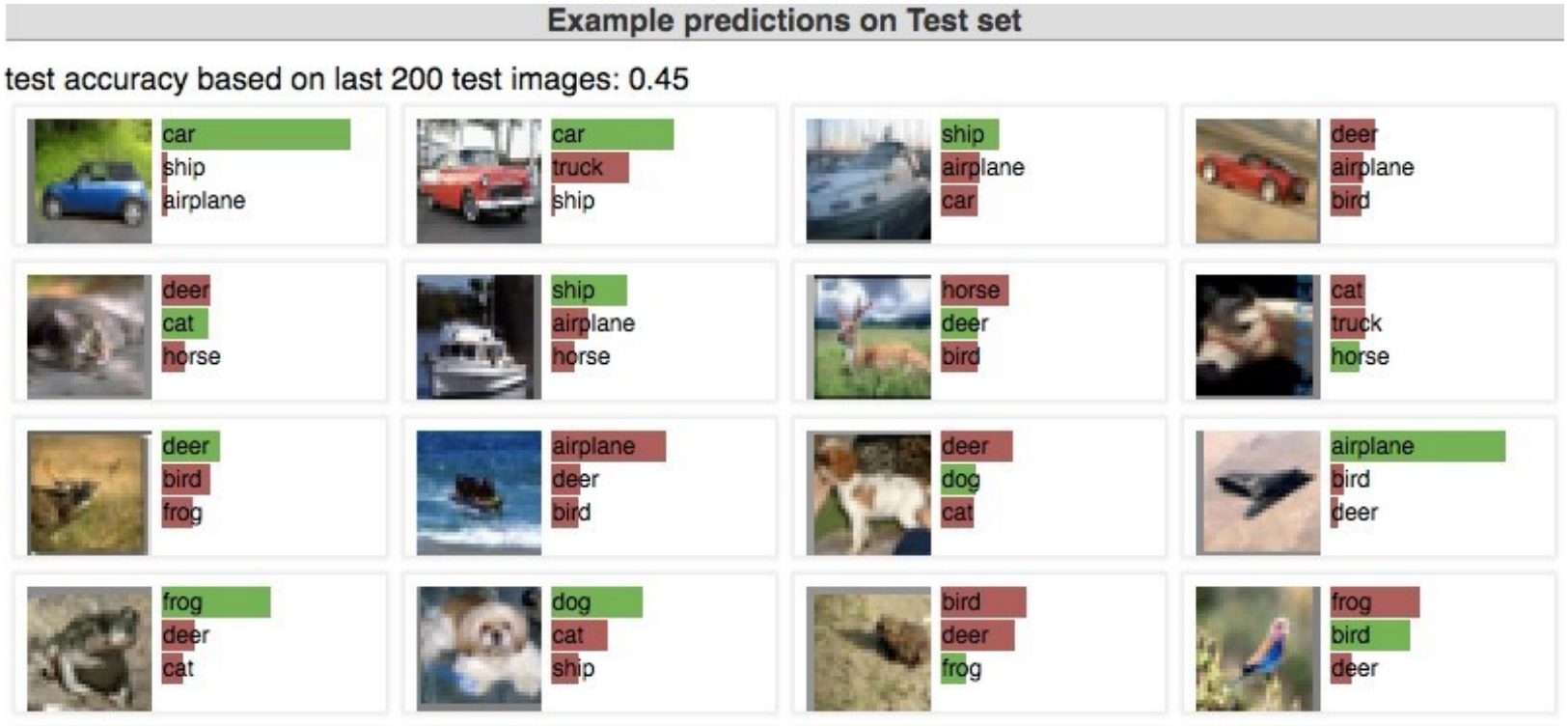
preview:


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

(btw, 1x1 convolution layers make perfect sense)



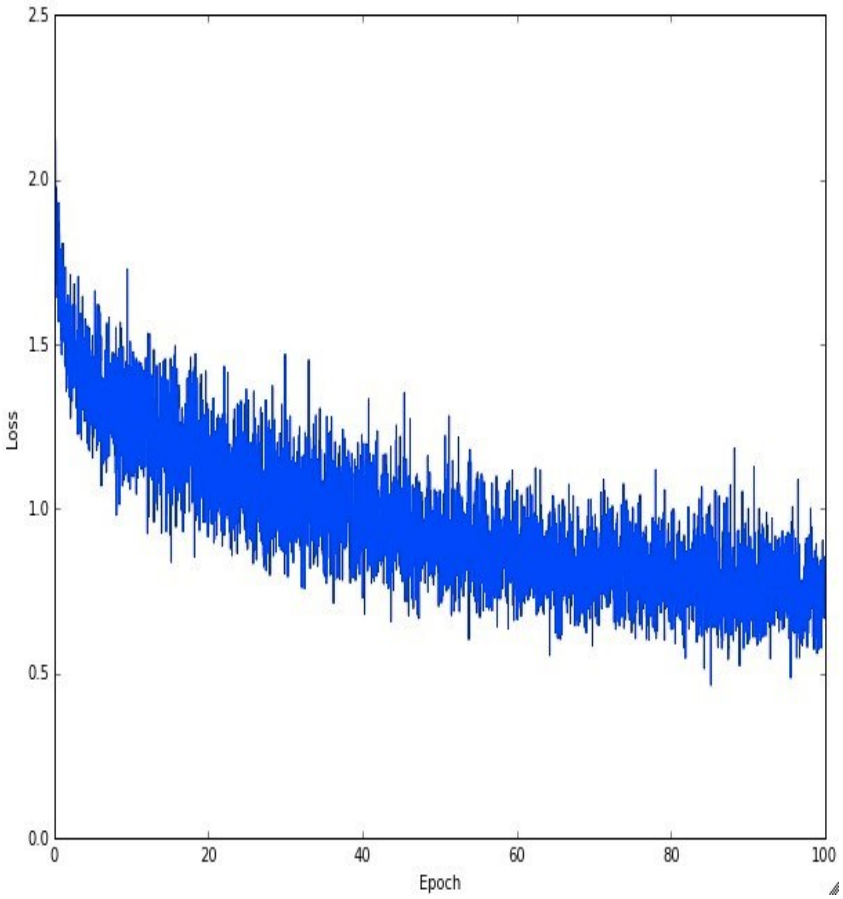
Demo

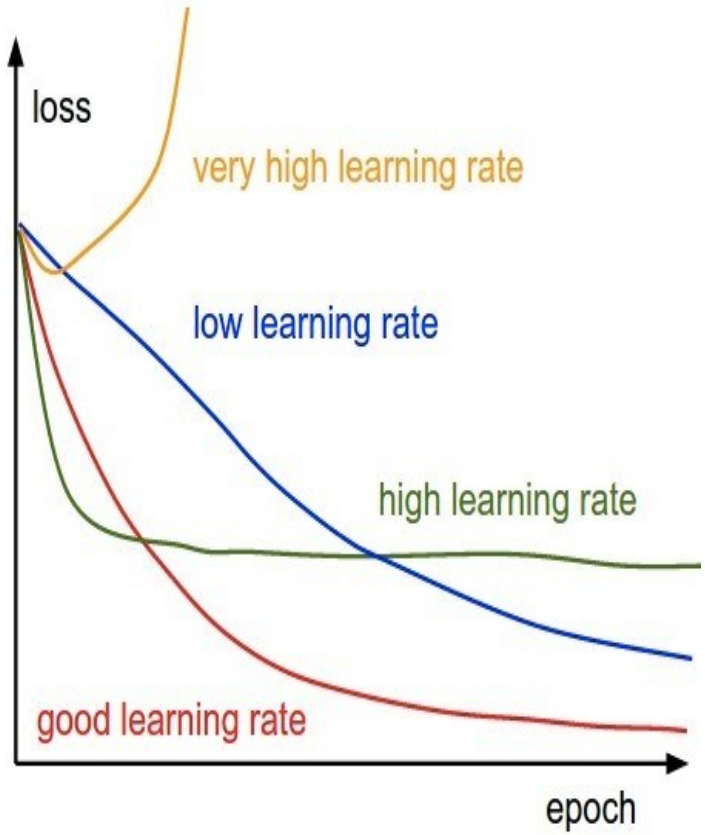


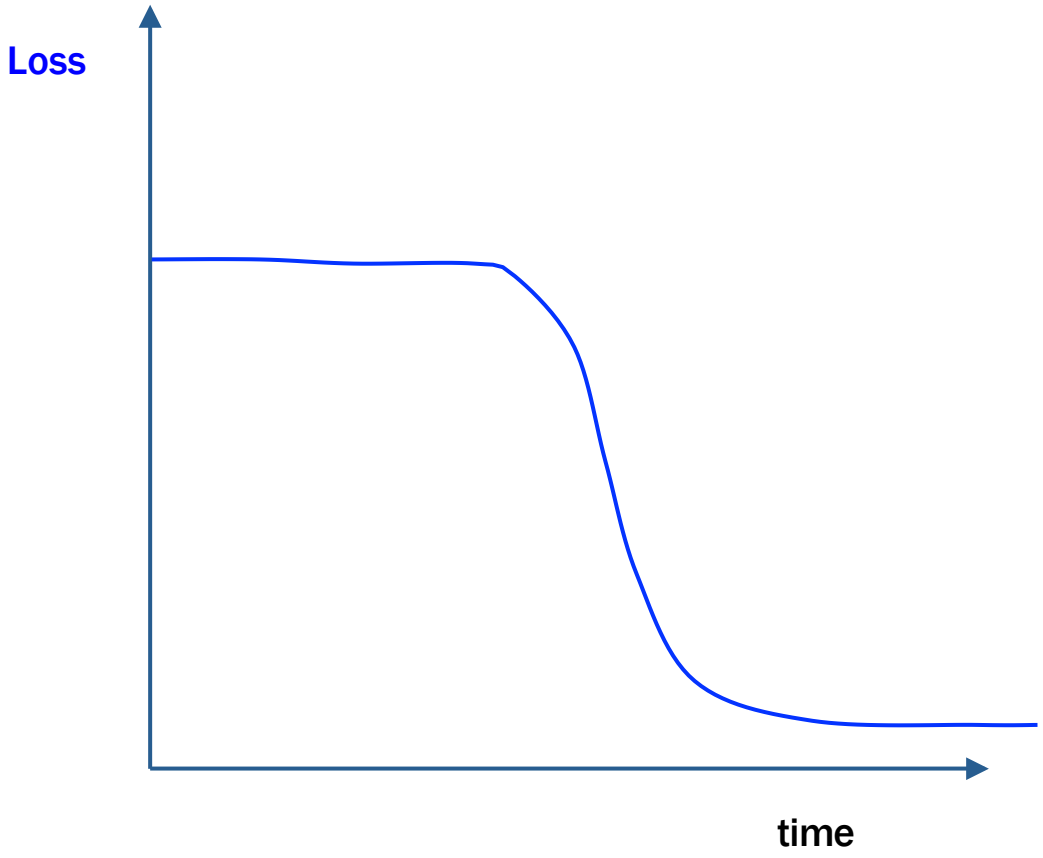
http://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

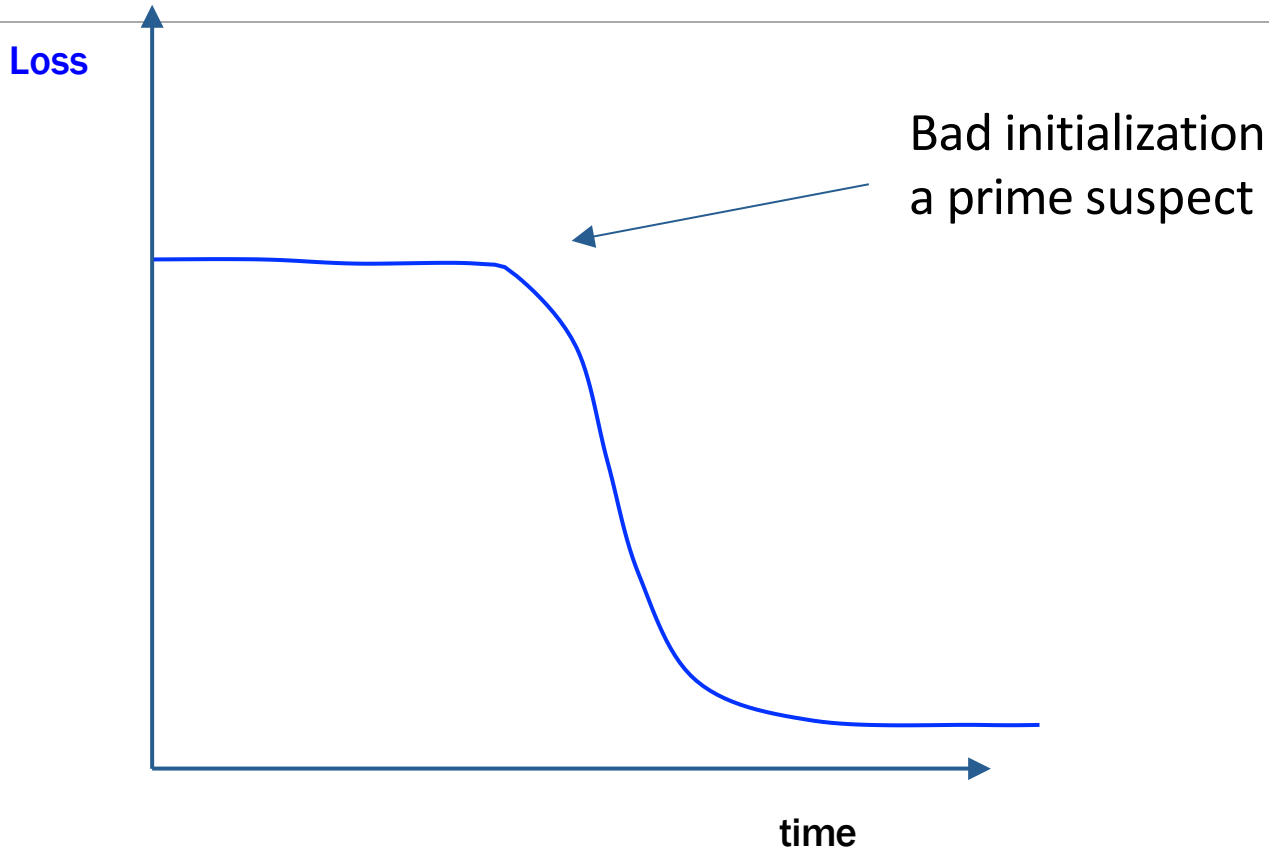
How to Tune Hyperparameters?

Monitor and visualize the loss curve

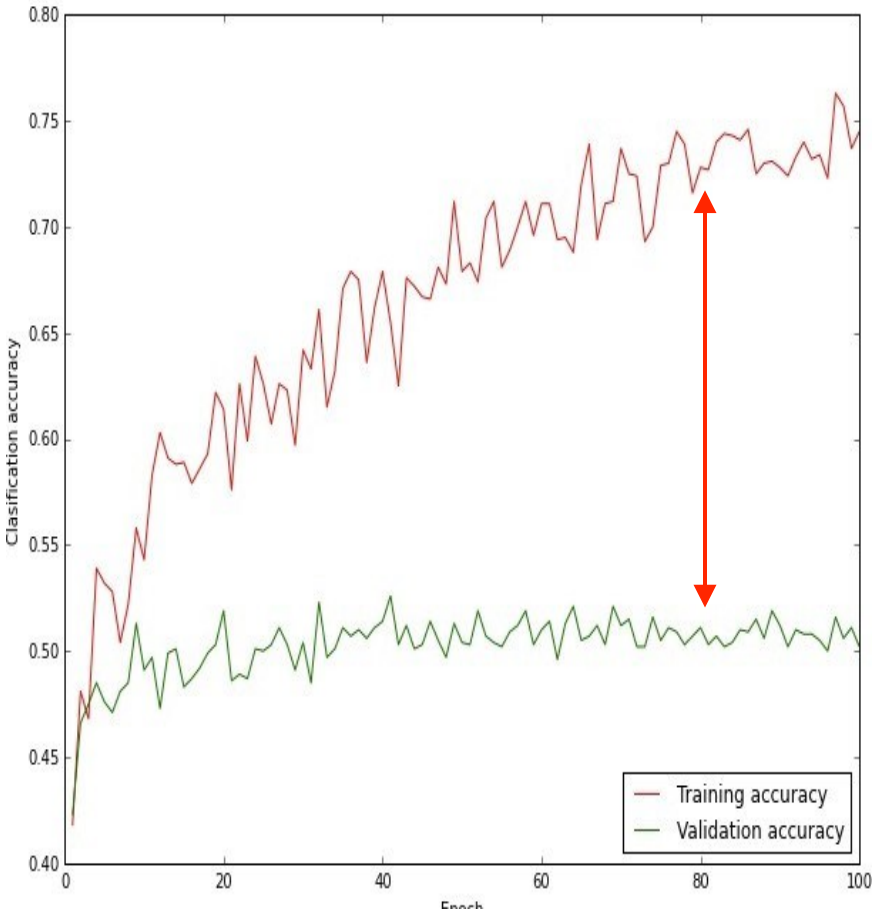





Loss
time


Loss
Bad initialization
a prime suspect
time

Monitor and visualize the accuracy:



big gap = overfitting

=> increase regularization strength?

no gap

=> increase model capacity?

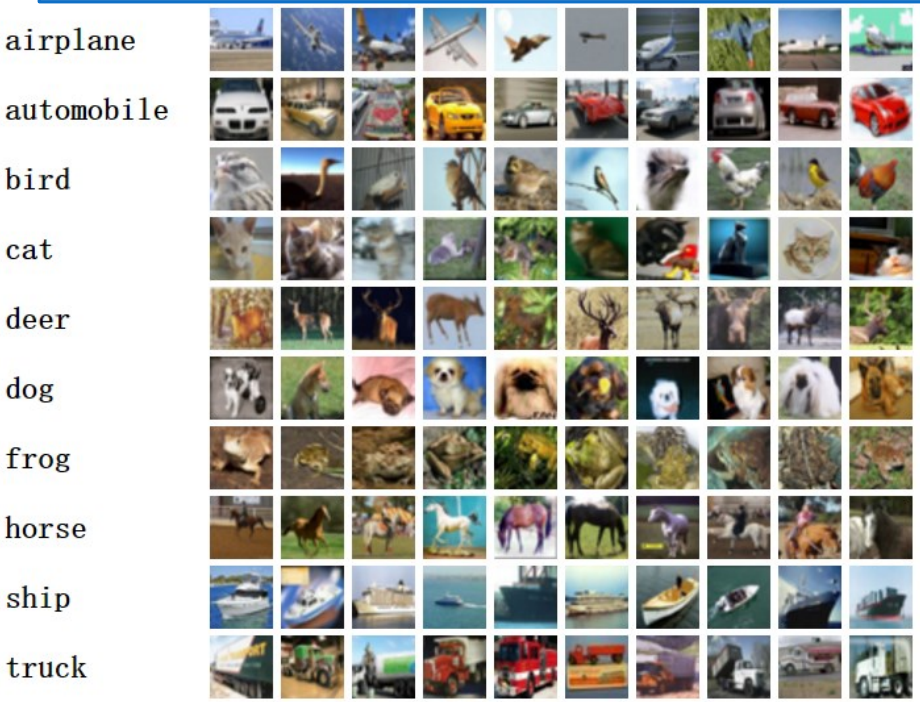
Pretrained Models

ImageNet Dataset

- The ImageNet project is a large visual database designed for use in visual object recognition software research.
- As of 2016, **over ten million URLs of images have been hand-annotated** by ImageNet to indicate what objects are pictured; in at least one million of the images, **bounding boxes** are also provided.
- The database of annotations of third-party image URL's is freely available directly from ImageNet; however, the actual images are not owned by ImageNet.
- Since 2010, the ImageNet project runs an annual software contest, the **ImageNet Large Scale Visual Recognition Challenge (ILSVRC),** where software programs compete to correctly classify and detect objects and scenes.

CIFAR10 dataset and Results of Different Models

The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes,
with 6000 images per class. There are 50000 training images and 10,000 test images.



Accuracy

Model	Acc.
VGG16	92.64%
ResNet18	93.02%
ResNet50	93.62%
ResNet101	93.75%
ResNeXt29(32x4d)	94.73%
ResNeXt29(2x64d)	94.82%
DenseNet121	95.04%
PreActResNet18	95.11%
DPN92	95.16%

Pretrained Models

- Many pretrained models exist on ImageNet dataset
  - https://www.tensorflow.org/api_docs/python/tf/keras/applications

- **How those are useful for us?**

Book Reading

- Murphy – Chapter 8
- Jurafsky – Chapter 5, Chapter 4, Chapter 7
- Tom Mitchel – Chapter 4