Convolution Neural Network

CS617: Basics of Deep Learning

Instructor: Aparajita Ojha

Slide 4

Sources: Prof. Ming Li Lecture slides, University of Waterloo, Canada
Prof. Fei Fei Li's Lecture slides, Stanford University

Neural Network for Image Data

- Suppose the input vector is an image database.
- You have to classify if the picture contains a bird or not.
- You have to detect a bird in the pictures.
- Each picture is of size say 256X256.
- What will be the size of the input layer here ?
  - 66,536
- If the first hidden layer contains 1024 units, how many weight parameters will be involved ?
  - 66,536 X 1024 = 68,132,864

Image Data

High dimensionality

Has redundancies

112	112	113	115	124	137	147	151	155	155	155	155	156	155	150	146	136
112	114	116	120	130	143	151	152	154	155	156	158	157	154	151	150	146
112	116	119	124	134	147	153	152	153	156	163	168	168	162	158	157	154
116	119	120	120	130	142	149	150	154	168	175	166	165	171	168	156	164
120	125	129	130	135	144	152	155	162	171	178	166	165	173	175	170	171
122	129	136	137	140	146	154	158	167	168	166	162	163	170	177	180	171

Pixel values of a block of an image

Patterns

Smaller Network: CNN

- Main questions –
- Do we need all the edges in this fully connected model?
- Can some of the weights be shared?

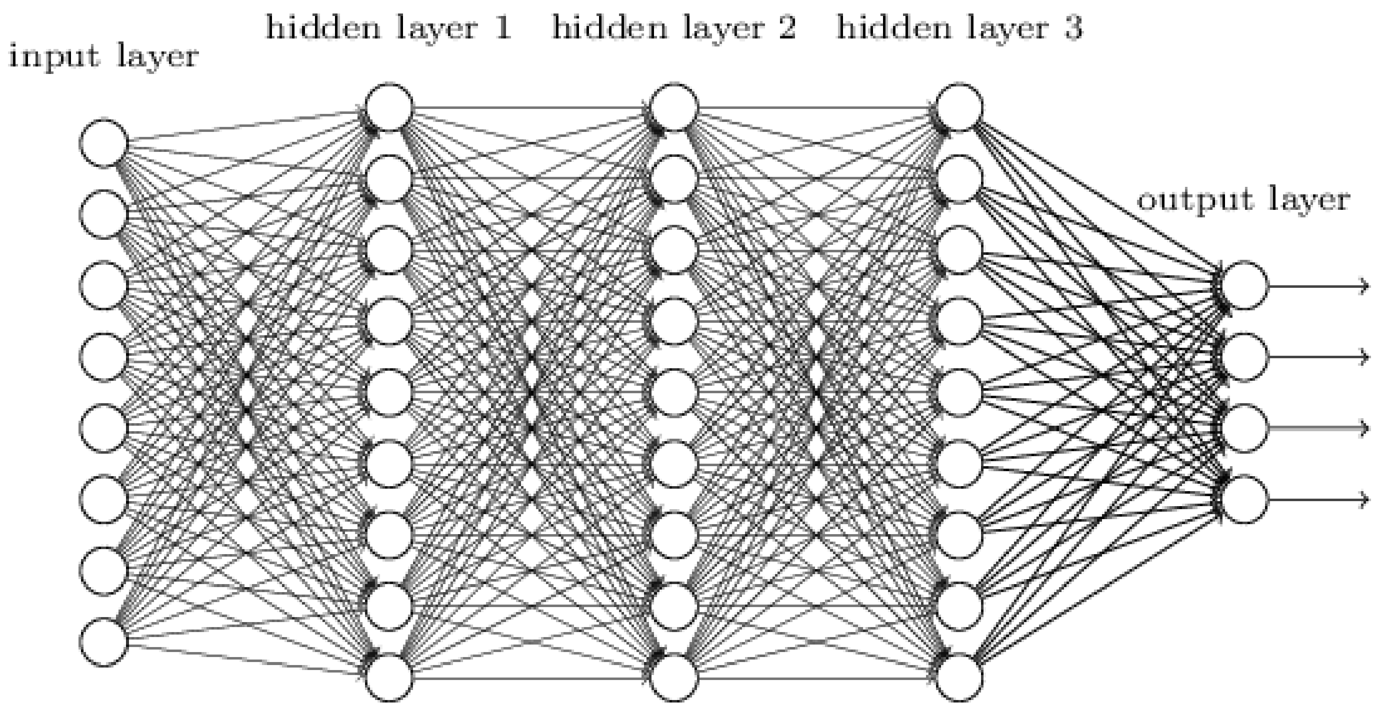
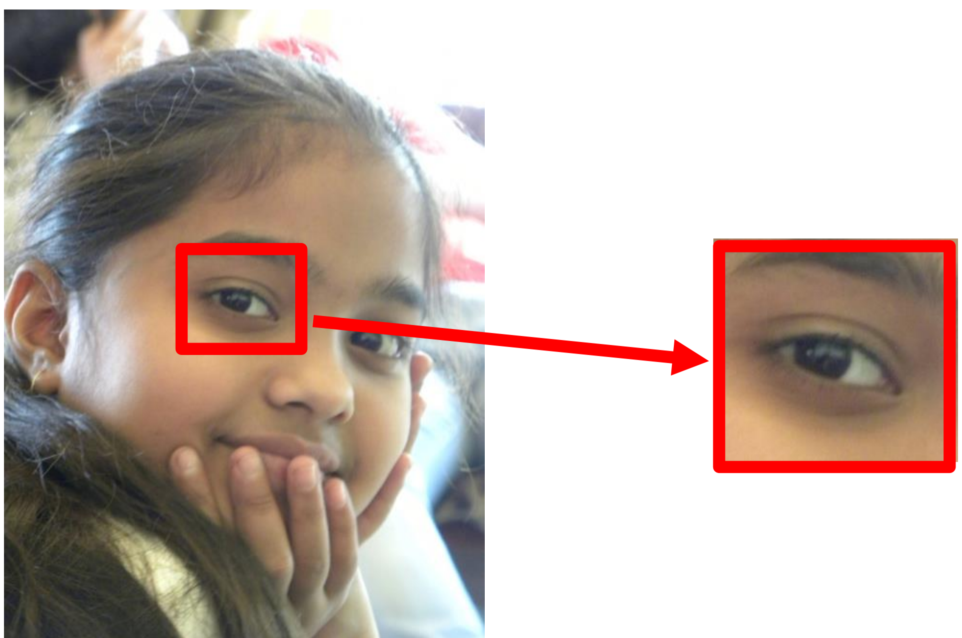


Image Understanding

- Patterns to learn can be very small compared to the image size.
- Example: Eye detector filter in images

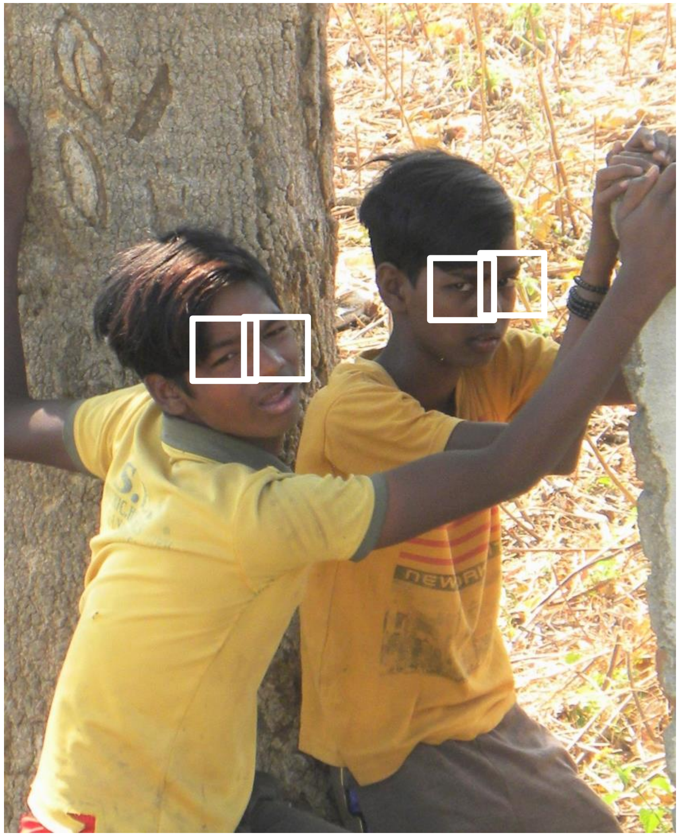


A small pattern detector can be used in place of the entire weight parameter matrix.

“eye” detector

Pattern Learning

- Same pattern can be at different positions in images.
- For examples –eyes.



Same pattern detector can move around and detect eyes at different positions !

Convolution Neural Networks

- Convolution Neural Networks (CNNs) use parameter sharing.

- Small pattern detectors called **filters** are used to convolve over the entire image.

- These filters are learned through NN training in the same way as in fully connected networks.

Convolution Neural Networks

- Just like a hidden layer in a fully connected layer, convolution layers are used in CNNs.
- To handle large size of image data, pooling layers are introduced.
- Normalization layers were used in early CNN architectures, but due to their minimal impact, they are not much used in the present CNNs.

A Convolutional Layer

A convolutional layer has a number of filters that perform convolutional operation.

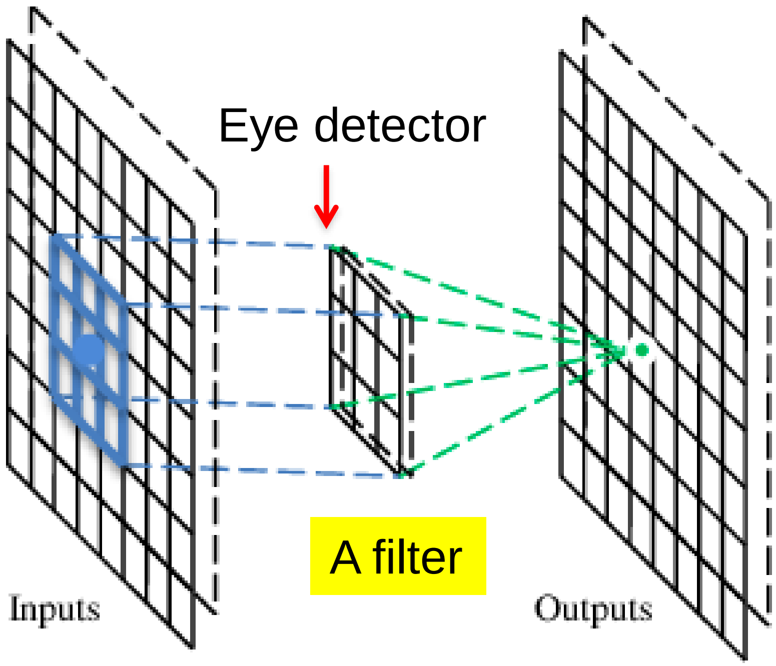


Figure source: Prof. Ming Li Lecture slides, University of Waterloo, Canada

Convolution Operation

5	1	7	7	8	1
7	2	5	7	8	1
5	3	5	4	4	2
4	4	3	3	2	1
3	3	4	7	8	1
5	7	8	2	3	1

1	0	-1
2	0	-2
1	0	-1

Convolution Operation

5	1	7	7	8	1
7	2	5	7	8	1
5	3	5	4	4	2
4	4	3	3	2	1
3	3	4	7	8	1
5	7	8	2	3	1

1	0	-1
2	0	-2
1	0	-1

Convolution Operation

5	1	7	7	8	1
7	2	5	7	8	1
5	3	5	4	4	2
4	4	3	3	2	1
3	3	4	7	8	1
5	7	8	2	3	1

1	0	-1
2	0	-2
1	0	-1

$$5 \times 1 + 1 \times 0 + 7 \times (-1) + 7 \times 2 + 2 \times 0 + 5 \times (-2) + 5 \times 1 + 3 \times 0 + 5 \times (-1) = 10$$

Convolution Operation

5	1	7	7	8	1
7	2	5	7	8	1
5	3	5	4	4	2
4	4	3	3	2	1
3	3	4	7	8	1
5	7	8	2	3	1

1	0	-1
2	0	-2
1	0	-1

$$1 \times 1 + 7 \times 0 + 7 \times (-1) + 2 \times 2 + 5 \times 0 + 7 \times (-2)$$
$$+ 3 \times 1 + 5 \times 0 + 4 \times (-1) = -17$$

Convolution Operation

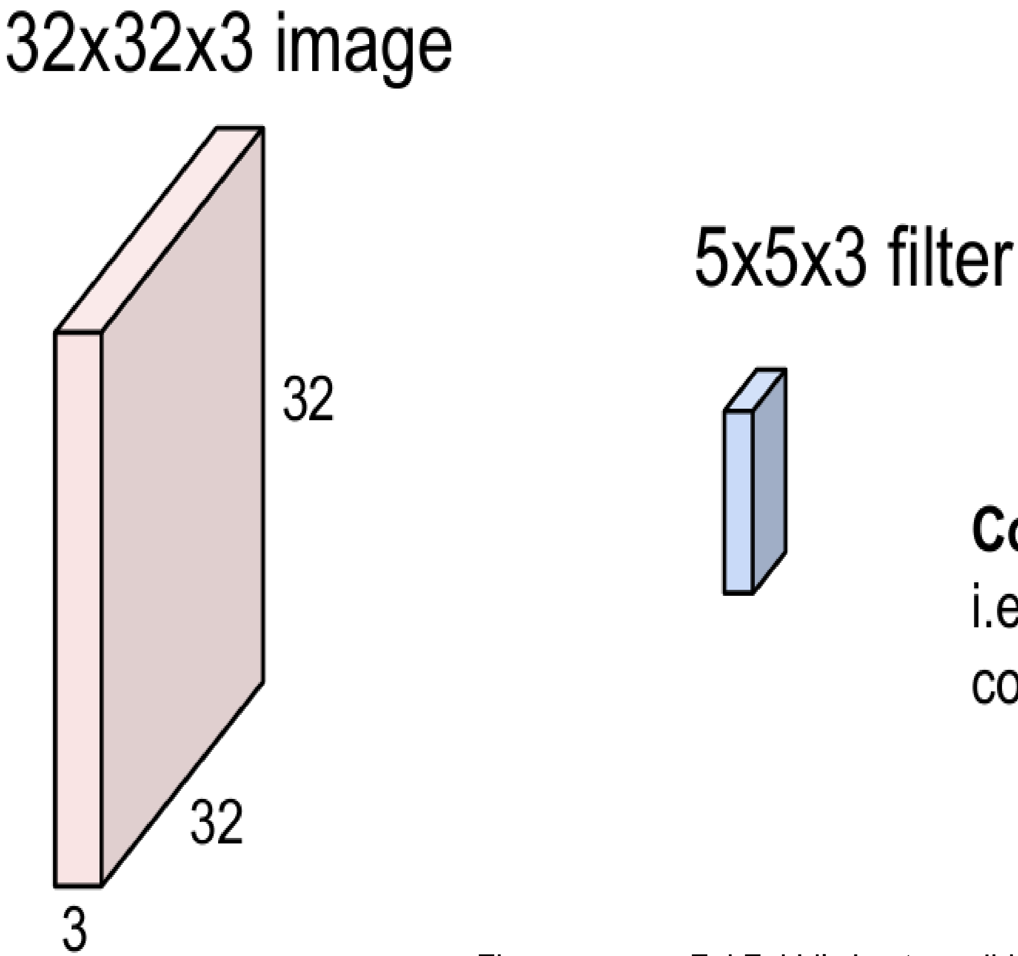
5	1	7	7	8	1
7	2	5	7	8	1
5	3	5	4	4	2
4	4	3	3	2	1
3	3	4	7	8	1
5	7	8	2	3	1

1	0	-1
2	0	-2
1	0	-1

$$7 \times 1 + 8 \times 0 + 1 \times (-1) + 7 \times 2 + 8 \times 0 + 1 \times (-2)$$
$$+ 4 \times 1 + 4 \times 0 + 2 \times (-1) = 20$$

And the process continues in this way…

A Convolution Layer

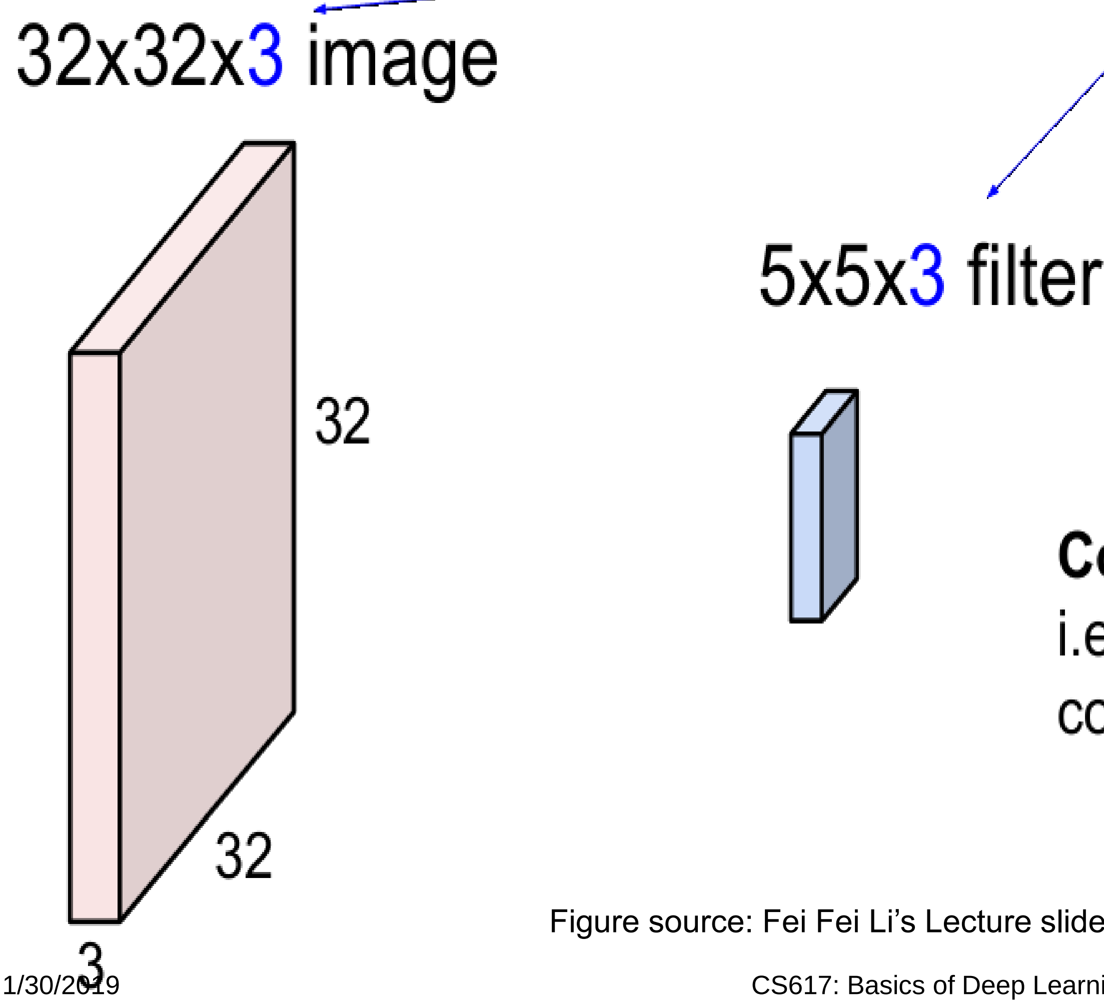


Convolve the filter with the image i.e. "slide over the image spatially, computing dot products"

Figure source: Fei Fei Li's Lecture slides, Stanford University

A Convolution Layer

Filters always extend the full depth of the input volume



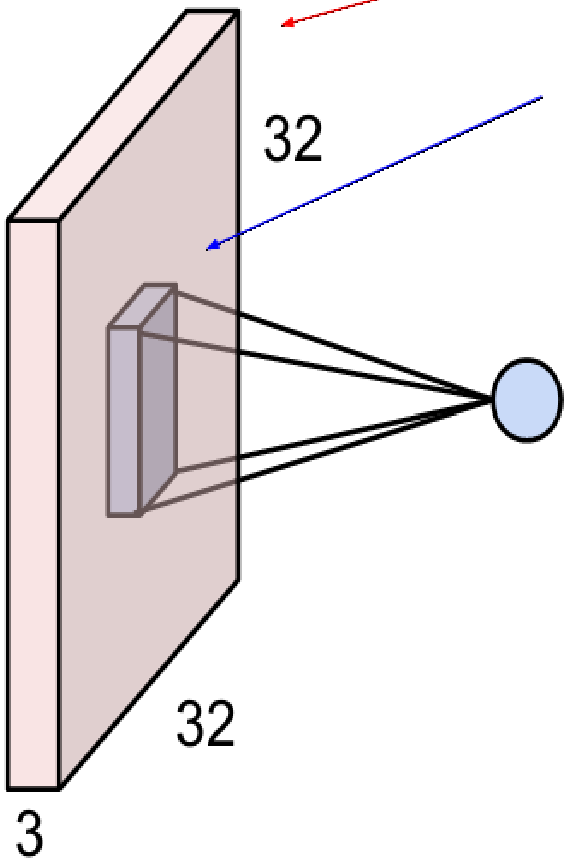
Convolve the filter with the image i.e. "slide over the image spatially, computing dot products"

Figure source: Fei Fei Li's Lecture slides, Stanford University

A Convolution Layer

32x32x3 image

5x5x3 filter w

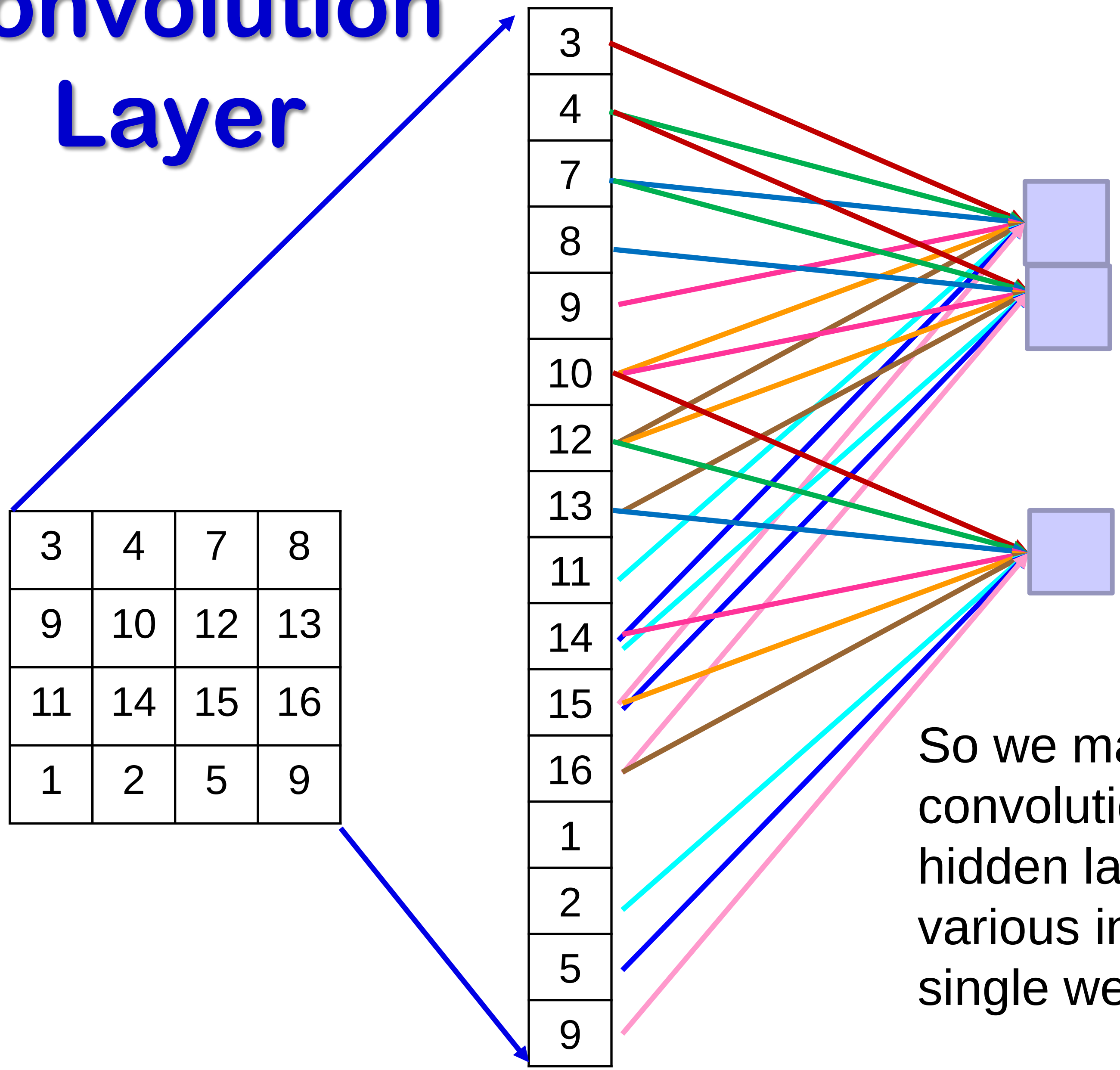


1 number:
the result of taking a dot product between the filter and a small 5x5x3 chunk of the image
(i.e. 5*5*3 = 75-dimensional dot product + bias)

$$w^T x + b$$

Figure source: Fei Fei Li's Lecture slides, Stanford University

Convolution Layer



So we may think of convolution layer as a hidden layer in which various inputs share a single weight parameter.

Convolution Layer

- Number of filters in a single Convolution layer are chosen by the user of the CNN (Depends on application and data input type).
- Typically a sizable number of filters are chosen to extract the maximum number of features.
- So what do you learn here ?
- Filters are learnt. Each filter learns to detect a small pattern.
- Filter values are initially chosen randomly.
- Gradually, values are updated to learn a pattern.

Convolution Layer

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

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

Filter 1

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

Filter 2

⋮ ⋮

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

Filter h

Color image: RGB 3 channels

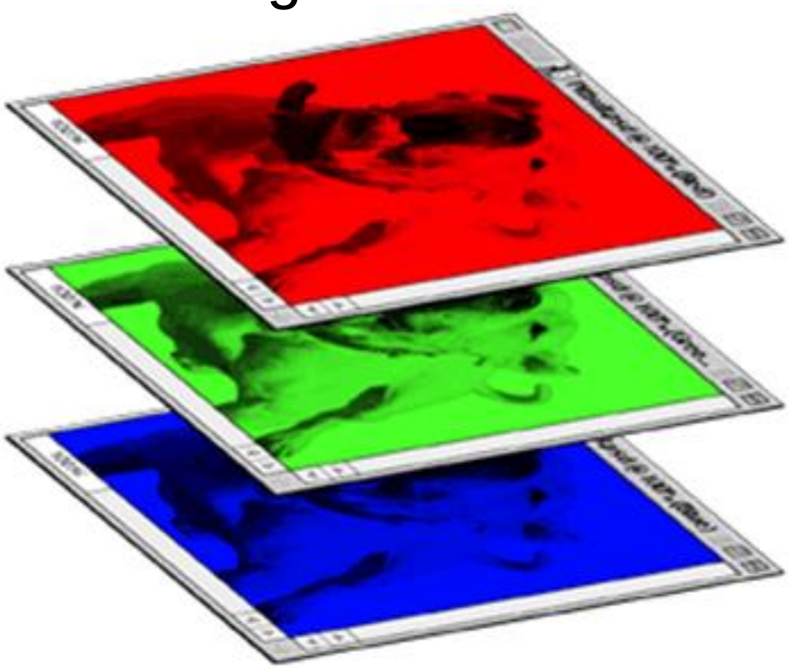
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

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

Filter 2

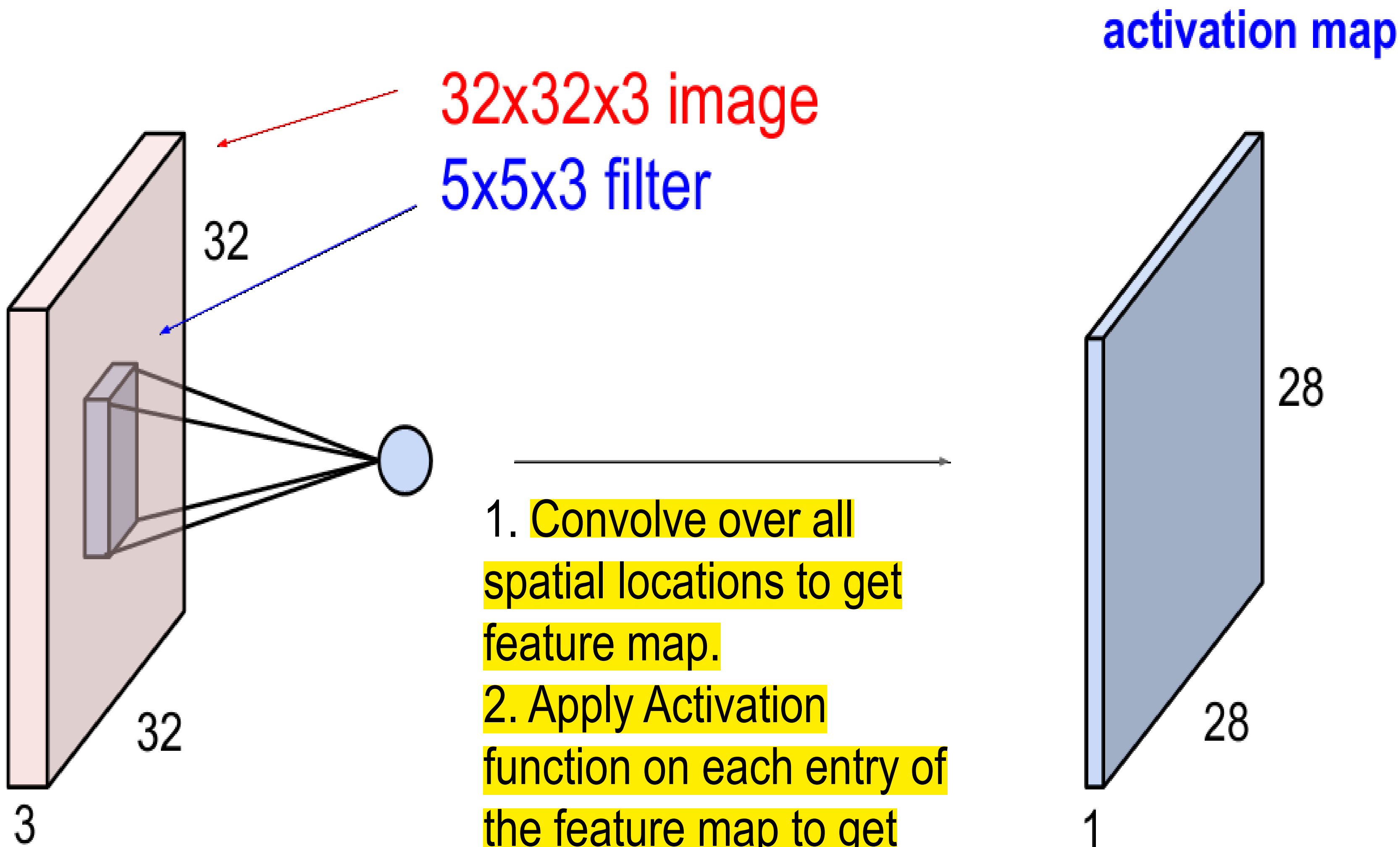
Color image



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

Figure source: Ming Li's Lecture slides, Waterloo University

Filters as activation maps



1. Convolve over all spatial locations to get feature map.
2. Apply Activation function on each entry of the feature map to get activation map

Figure source: Fei Fei Li's Lecture slides, Stanford University

Filters as activation maps

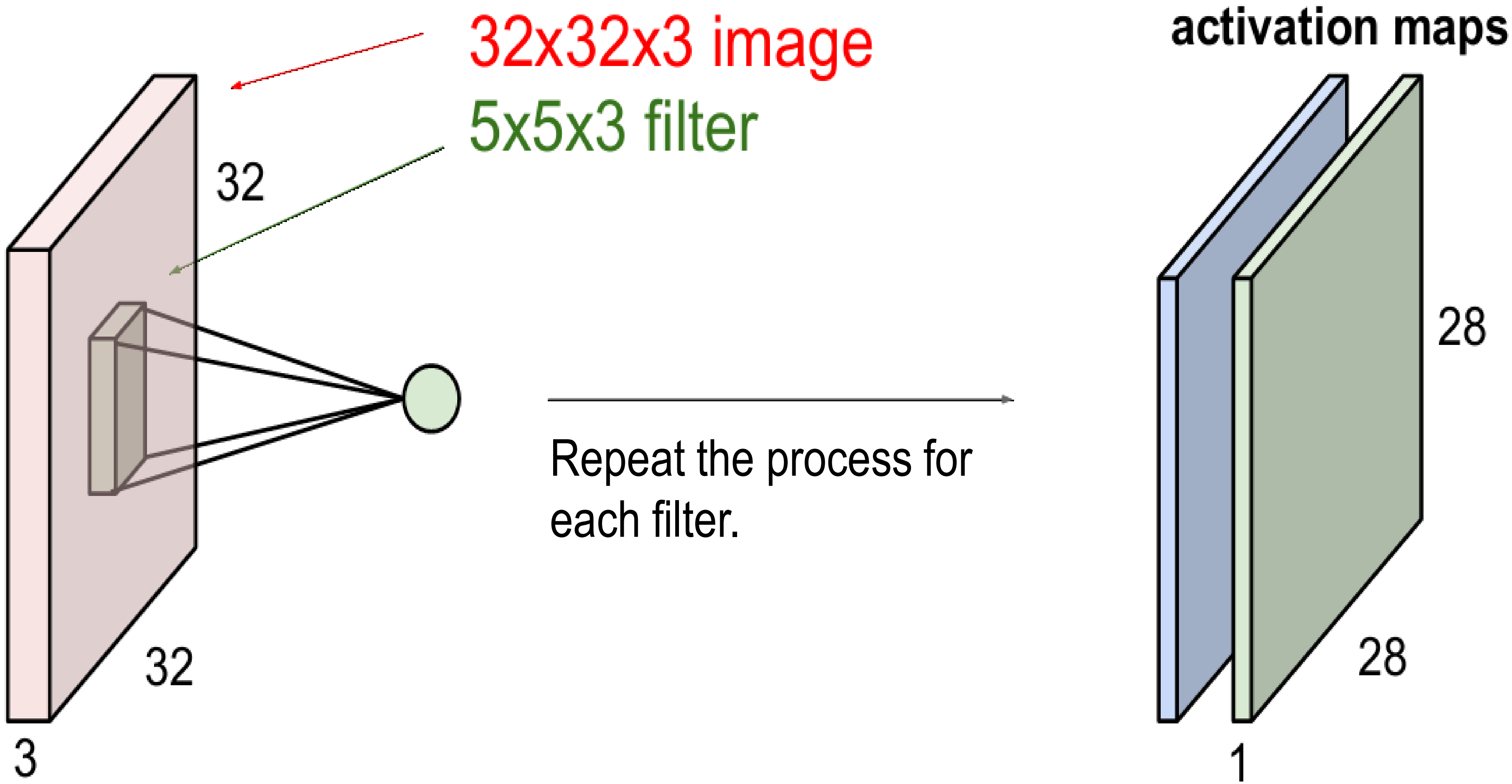
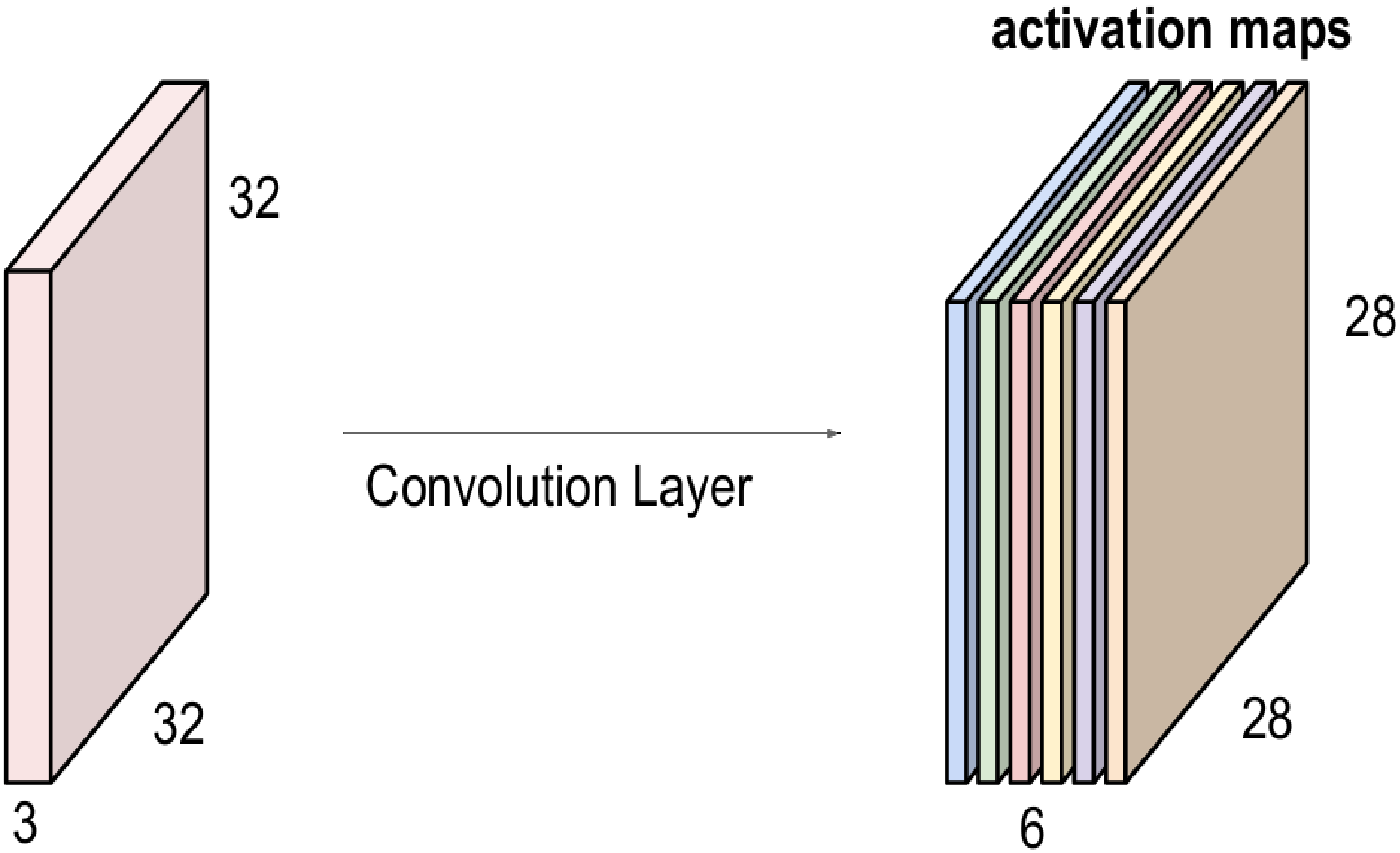


Figure source: Fei Fei Li's Lecture slides, Stanford University

Filters and Activation Maps

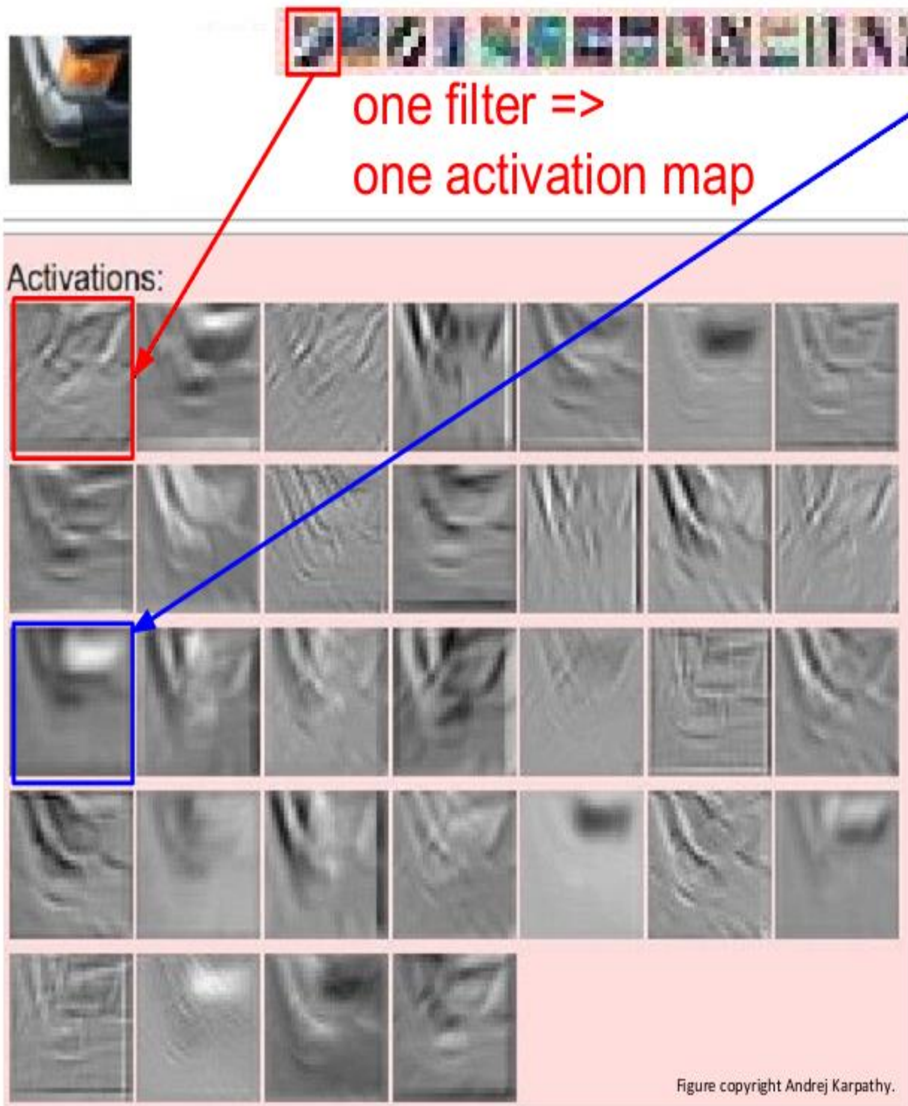
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



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

Figure source: Fei Fei Li's Lecture slides, Stanford University

Filters and Activation Maps



example 5x5 filters
(32 total)

We call the layer convolutional because it is related to convolution of two signals:

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} f[n_1,n_2] \cdot g[x-n_1, y-n_2]$$

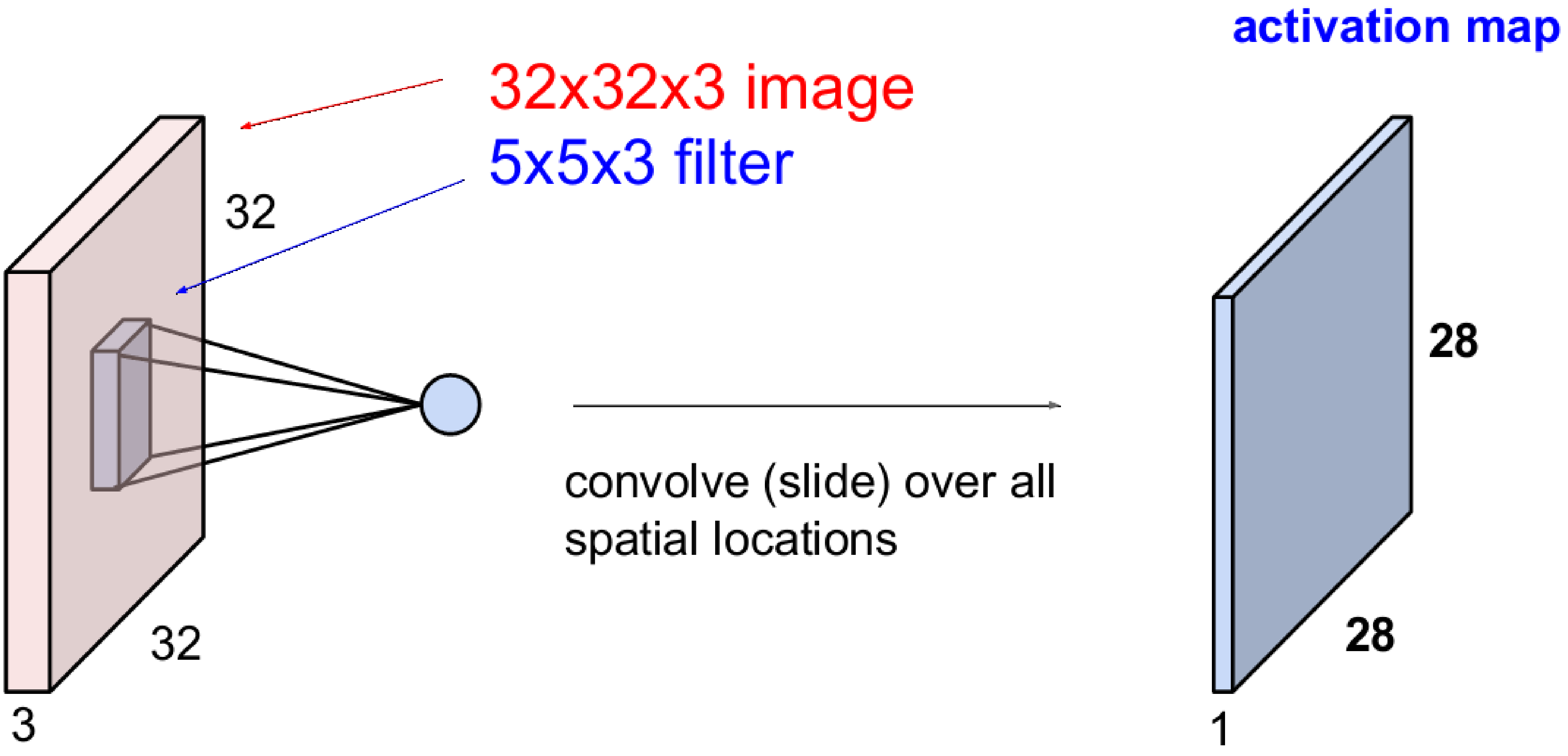
elementwise multiplication and sum of a filter and the signal (image)

Figure source: Fei Fei Li’s Lecture slides, Stanford University

Filters and Redundancies

- Observe that some filters detect almost same patterns.
- So the number of filters can be extra and can lead to redundant information.
- But a small number of filters may lead to loss of pattern information.
- So it is better to use a large number of filters in the beginning to avoid any information loss.

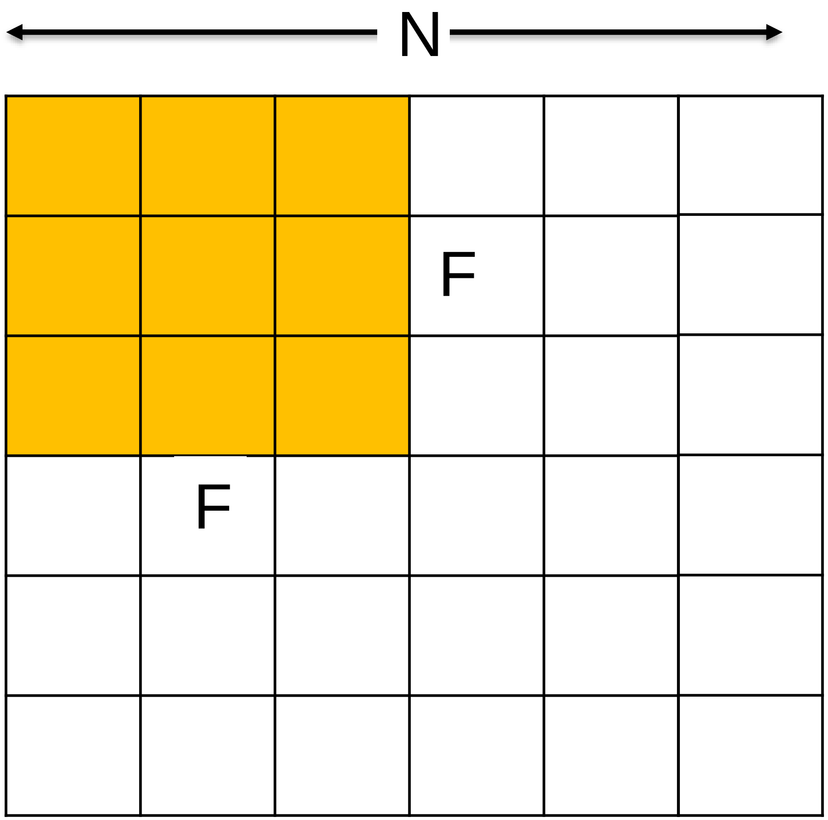
Convolution layers and image dimension



Output Volume

- Suppose the image is 256X256X3 and there are 8 filters , each of size 5X5, what will be the output volume ?
- Each filter 5x5 will be converted to a 5x5x3 filter volume.
- Then it will convolve with each block of 5x5x3 in the image to give just one output value.
- Whole convolution process with one filter will give a layer of size 252X252.
- For 8 filters, it will be a 252X252X8 volume.

Convolution Layer and Output Dimension



Output size:

$$\frac{N-F}{stride}+1$$

$$Stride = 1$$

Let N = 6 and F =3
What will happen if the stride is 3 ?

N= 10 and F = 3, Stride S = 3.
What will be the output volume?

Convolution Layers and Output Dimensions

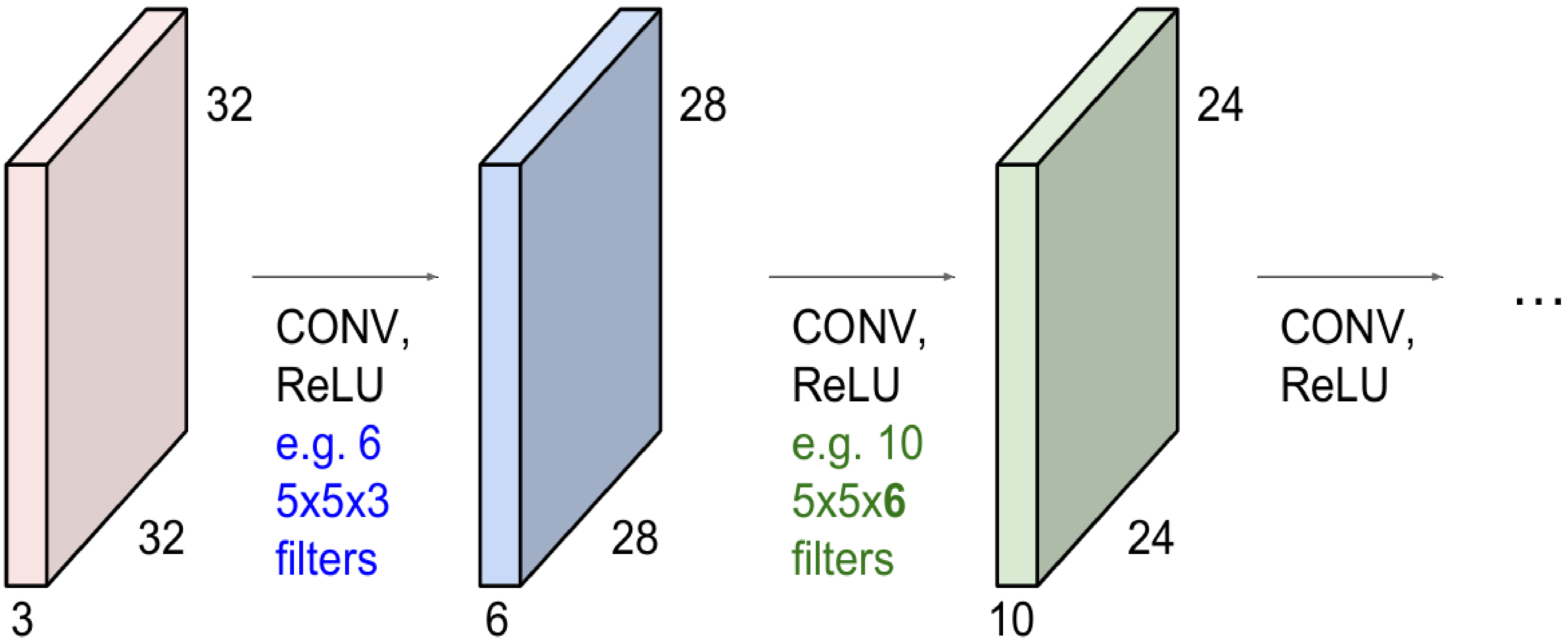


Figure source: Fei Fei Li's Lecture slides, Stanford University

Zero Padding

- Two advantages
  - Dimension reduction is controlled.
  - Corner pixels have better contribution.

0	0	0	0	0	0	0	0
0							0
0							0
0							0
0							0
0							0
0							0
0	0	0	0	0	0	0	0

Zero Padding

- Example. Input image of size 7x7
- 3x3 filter, applied with stride 1
- Pad with 1 pixel border. what will be the output?
  - 7x7
- CONV layers mostly use stride S = 1.
- What should be the size of zero padding to maintain the image's height and width ?
- (F-1)/2. (padding = 'SAME' used in Python)
- Example
  - F = 3 => zero pad with 1
  - F = 5 => zero pad with 2

Input and Output Volumes

- Suppose the input volume is 64X64X3.
- Apply 10 filters, each of size 5X5
- Padding P = 1, stride S = 1. What will be the output volume?
- Formula: $\left\lfloor \frac{N-F+2P}{S} \right\rfloor + 1.$
- After convolution apply bias b: $w^T x + b$.
- How many parameters to learn?

- Now device a formula if the input volume is of the size : $WXHX3$

- Width : $\left\lfloor\frac{W-F+2P}{S}\right\rfloor + 1$

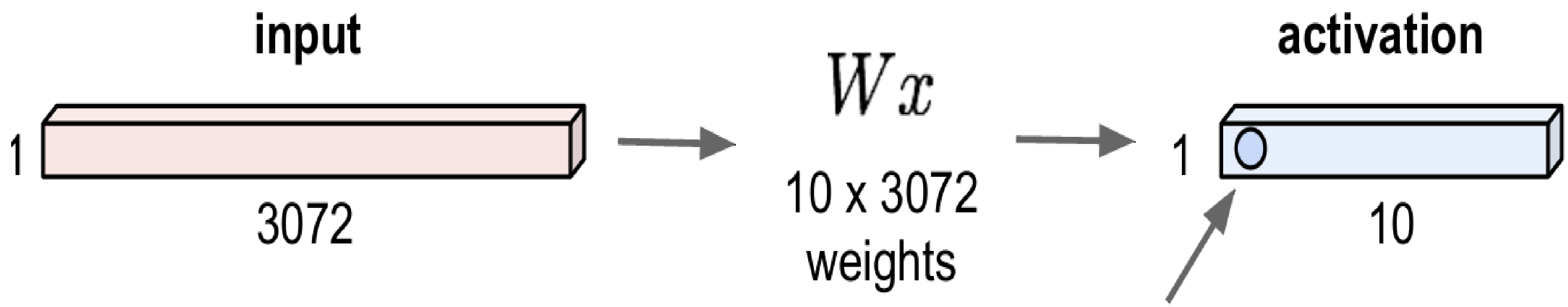
- Height : $\left\lfloor\frac{H-F+2P}{S}\right\rfloor + 1$

- Depth = number of filters.

Convolution v.s. Fully Connected

32x32x3 image -> stretch to 3072 x 1

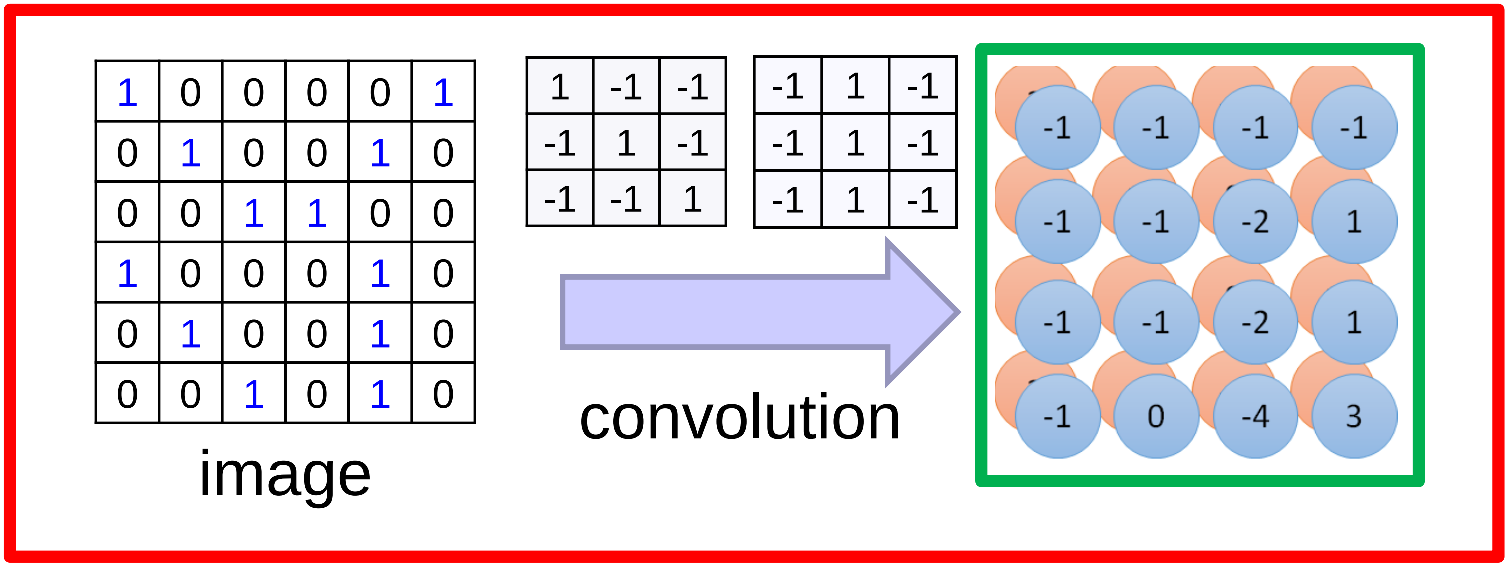
Each neuron looks at the full input volume



1 number:
the result of taking a dot product between a row of W and the input (a 3072-dimensional dot product)

Figure source: Fei Fei Li's Lecture slides, Stanford University

Convolution v.s. Fully Connected



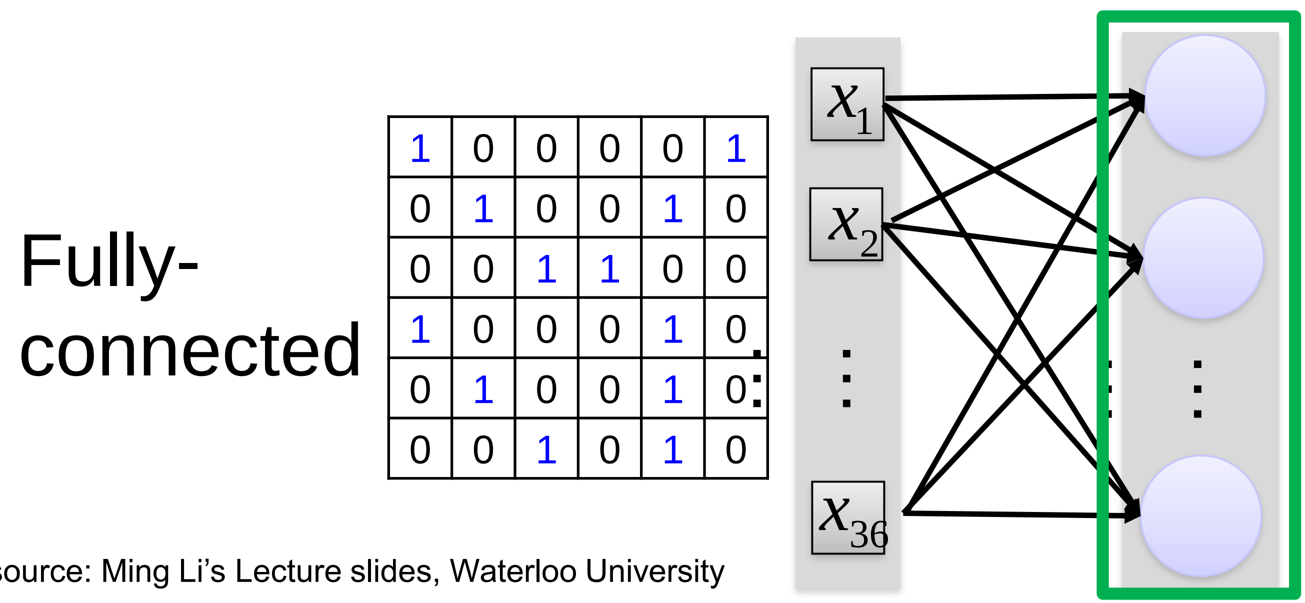


Figure source: Ming Li's Lecture slides, Waterloo University

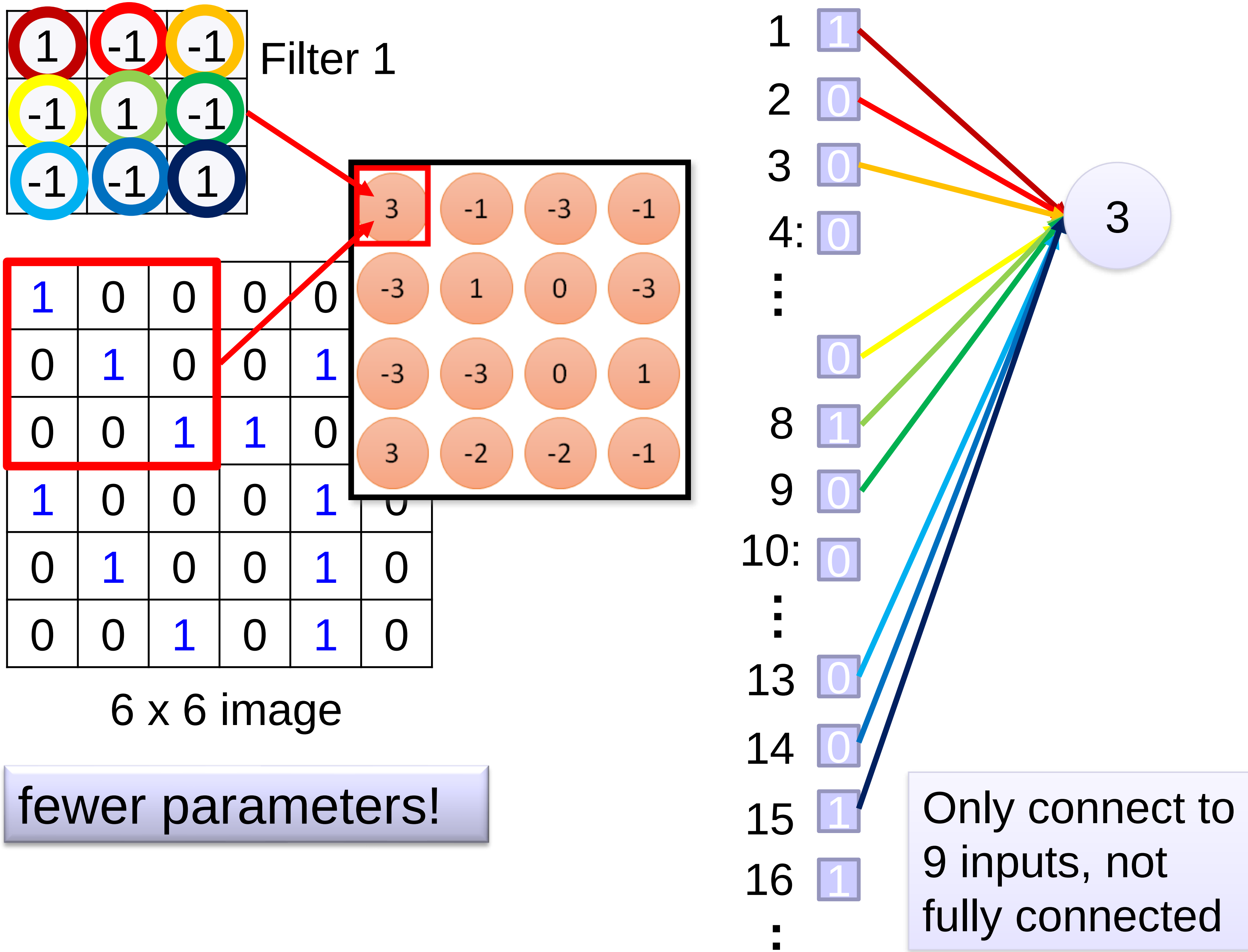
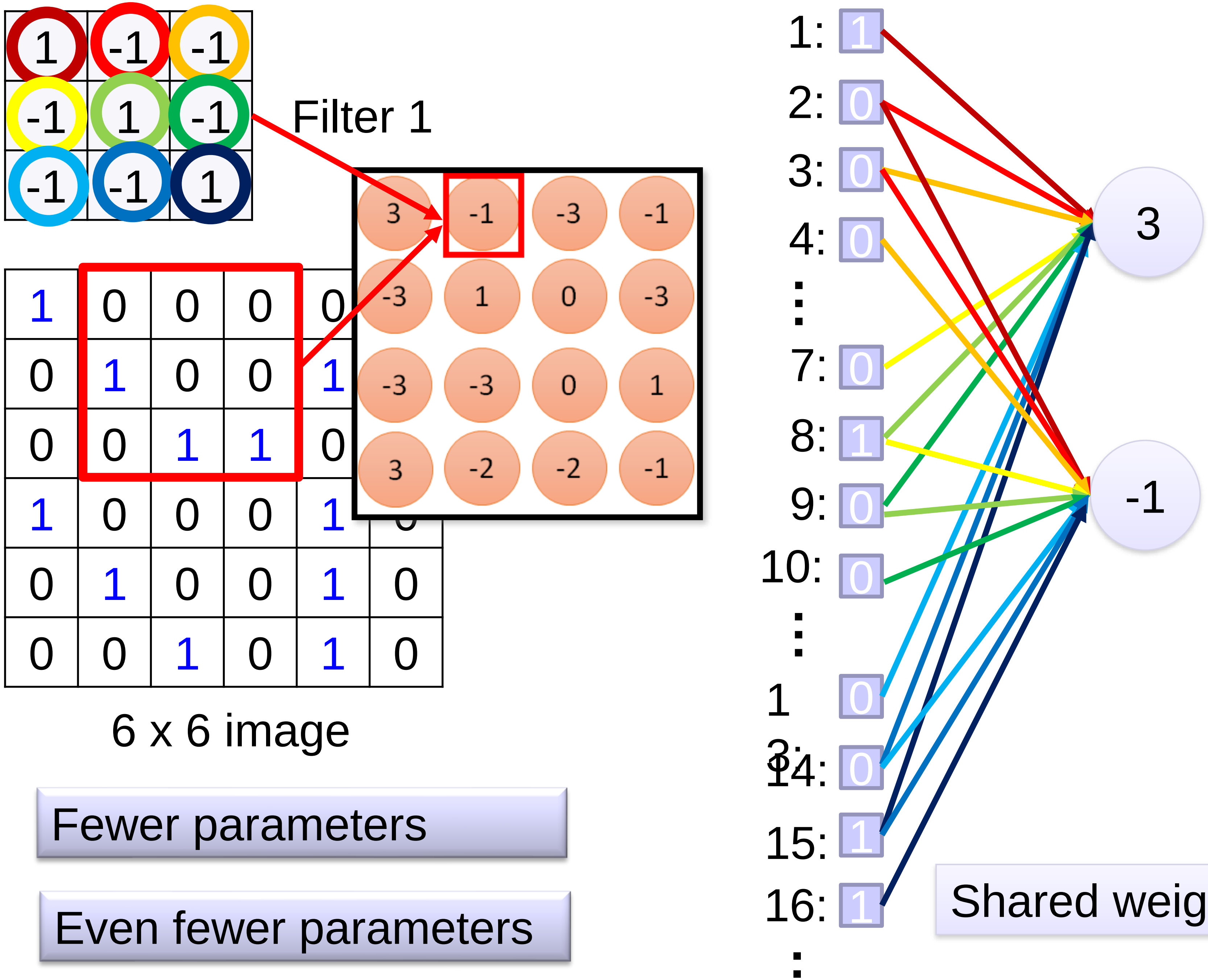


Figure source: Ming Li's Lecture slides, Waterloo University



Fewer parameters

Even fewer parameters

Shared weights

Figure source: Ming Li's Lecture slides, Waterloo University

The whole

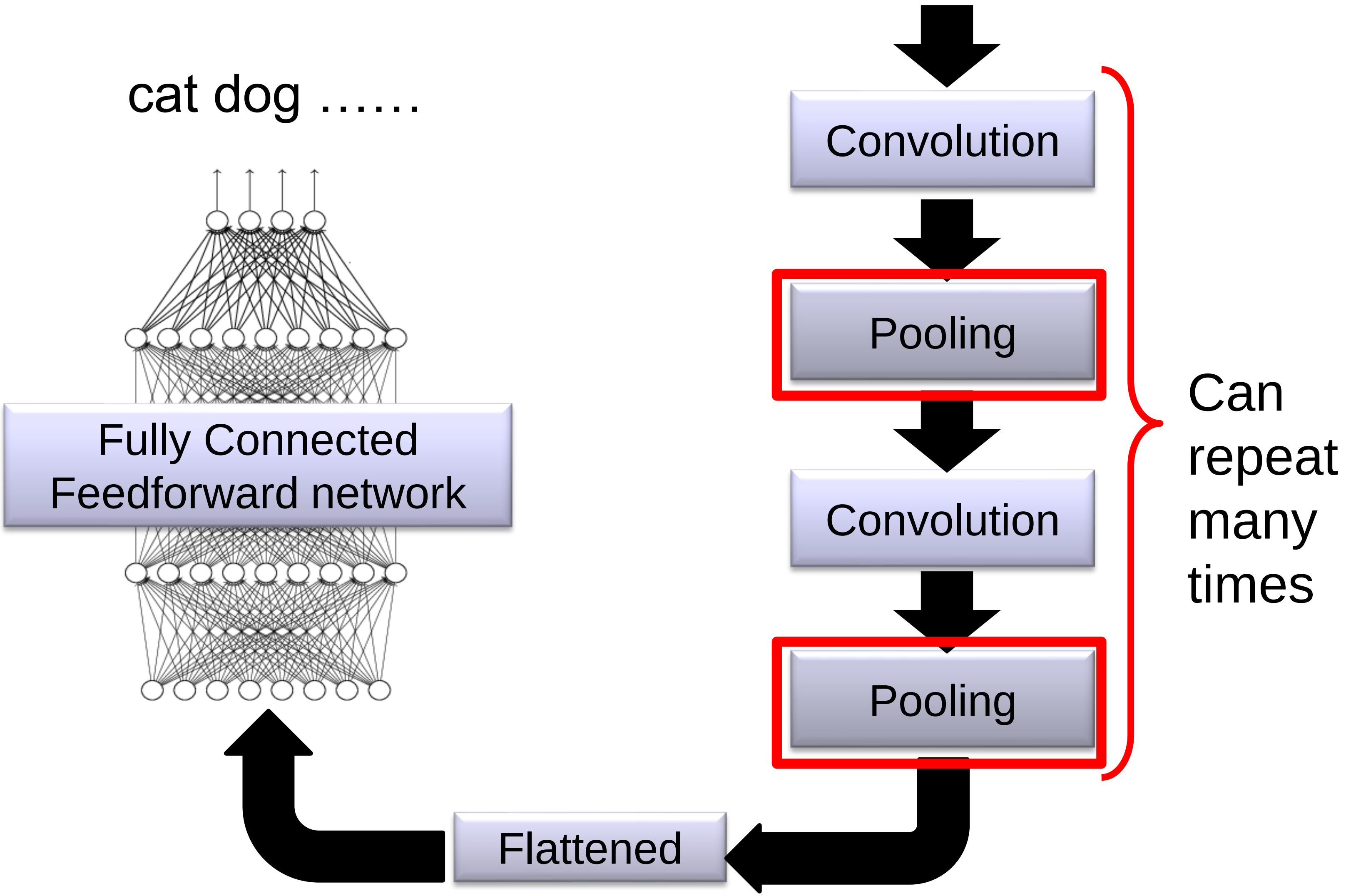


Figure source: Ming Li's Lecture slides, Waterloo University

Pooling layer

- When the image is large, one needs to compress the same for managing the input and output volumes between different convolution layers.
- Pooling layers make the representations smaller.
- Output volumes are more manageable.
- Operates over each activation map independently.

Pooling layer

- Max pooling
- Average pooling
- Others: Min pooling.

Max Pooling

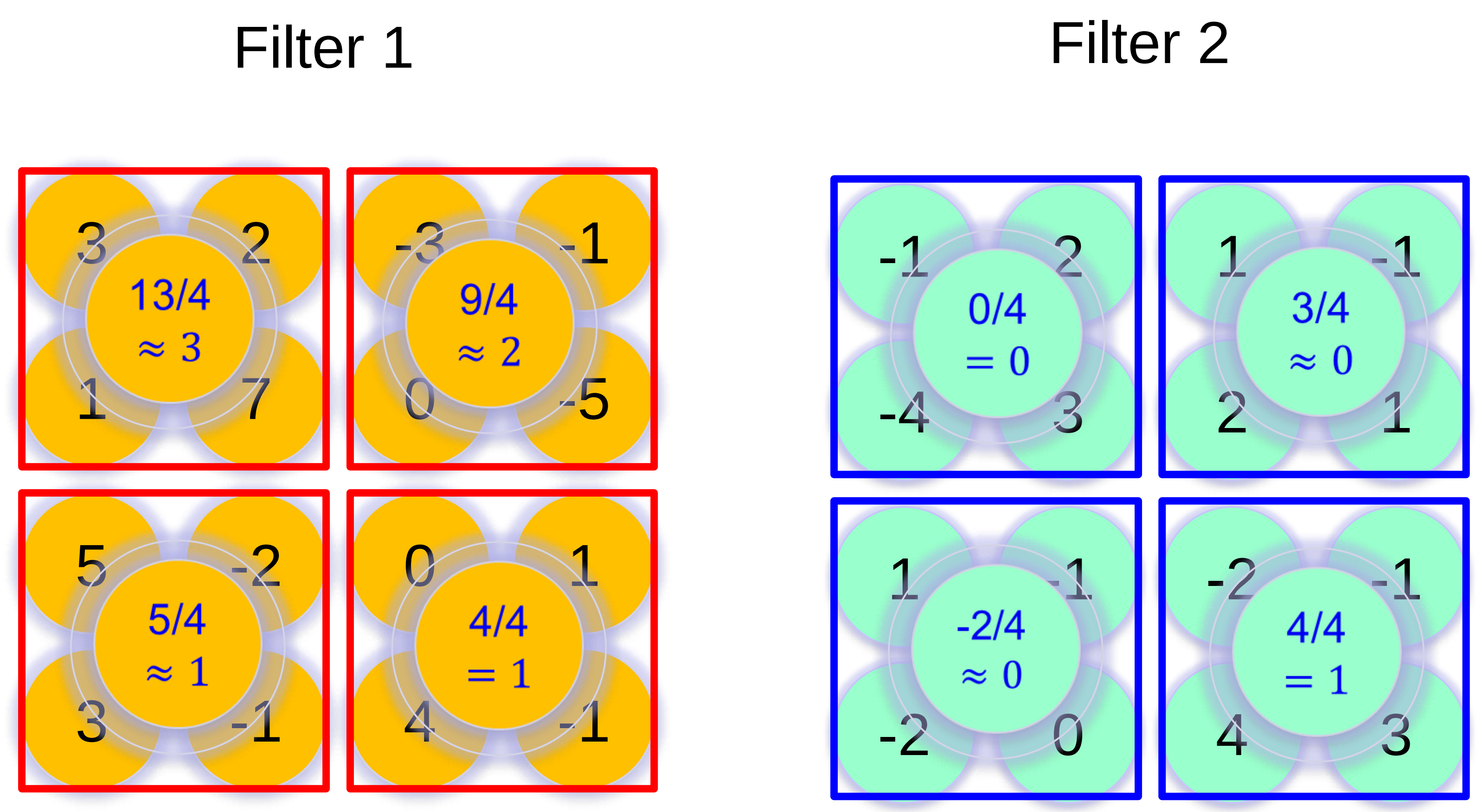
Filter 1

3	2	-3	-1
1	7	0	-5
5	-2	0	1
3	-1	4	-1

Filter 2

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

Average Pooling



Min Pooling

Filter 1

3	2	-3	-1
1	7	0	-5
5	-2	0	1
3	-1	4	-1

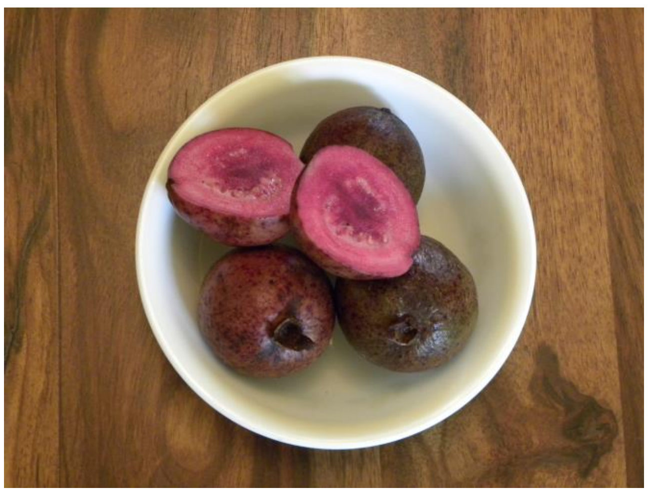
Filter 2

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

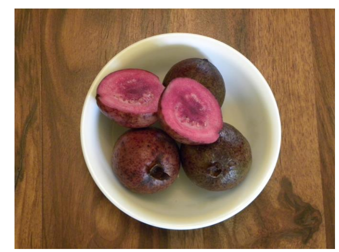
Pooling Advantage

- Pooling is a kind of subsampling.
- Subsampling the image does not loose image information.

A bowl of food on a plate



A bowl of food on a plate



Fewer parameters needed to extract features for pattern recognition and characterizing the image.

CNN Advantage in Dealing High Dimension Data

- Number of connections are reduced (Sparse interactions).
- Parameter sharing.
- Pooling layers compress the output volumes.
- Equivariant representations.
- As a result, number of parameters to learn are much less.
- Ability to work with inputs of variable size.
- But all depends on how many filters you choose for training.

A Generic CNN Architecture

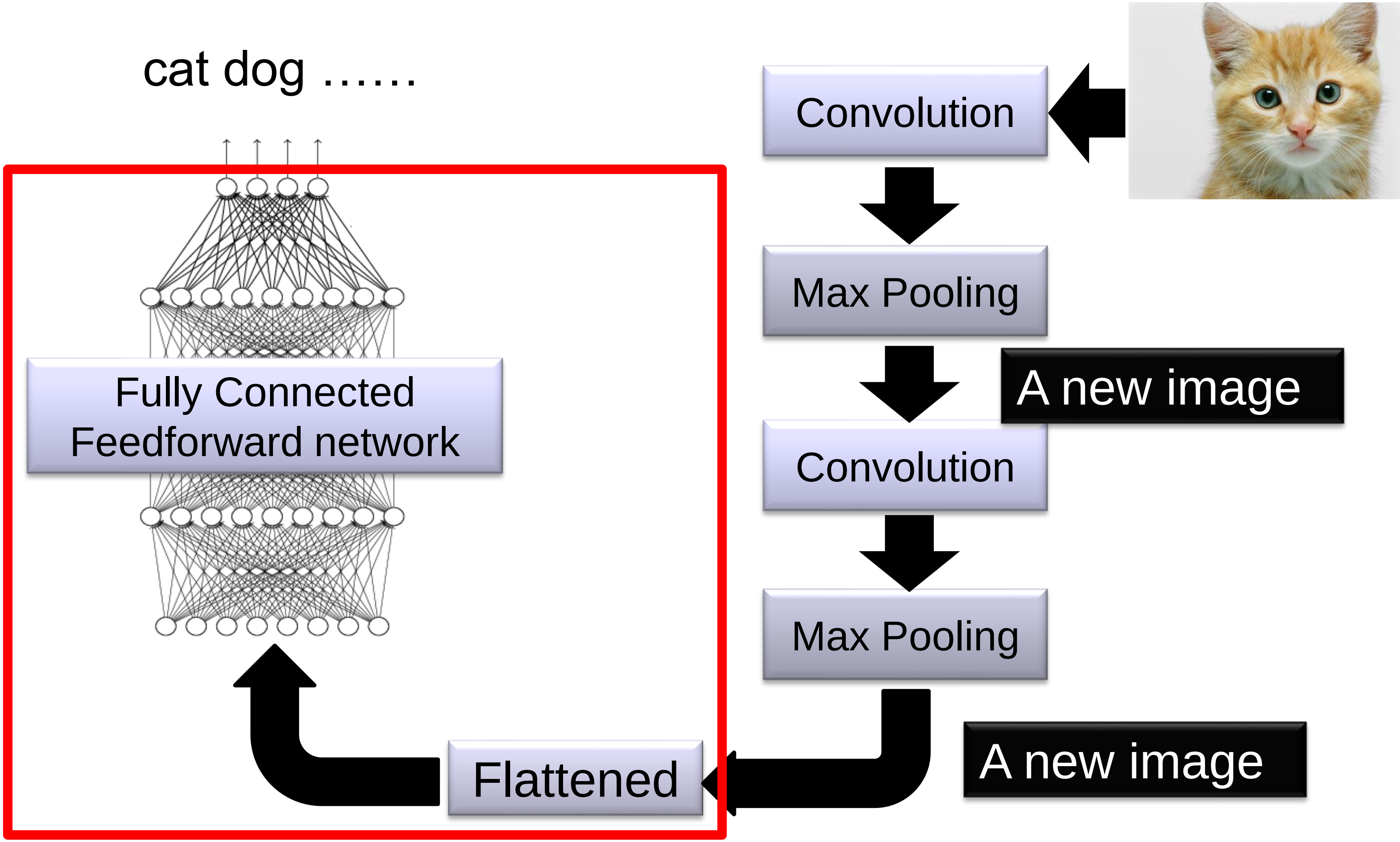
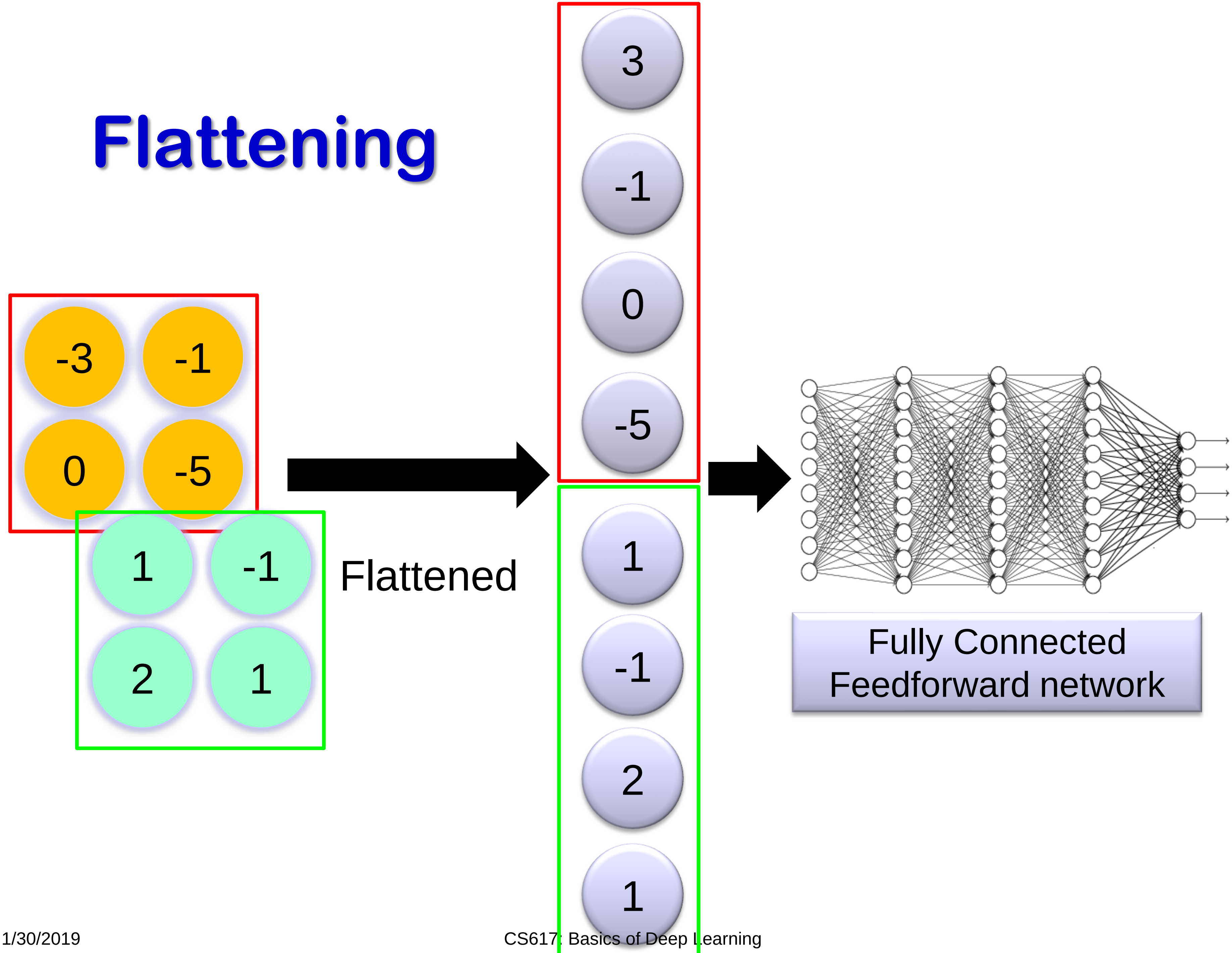


Figure source: Ming Li's Lecture slides, Waterloo University

Flattening



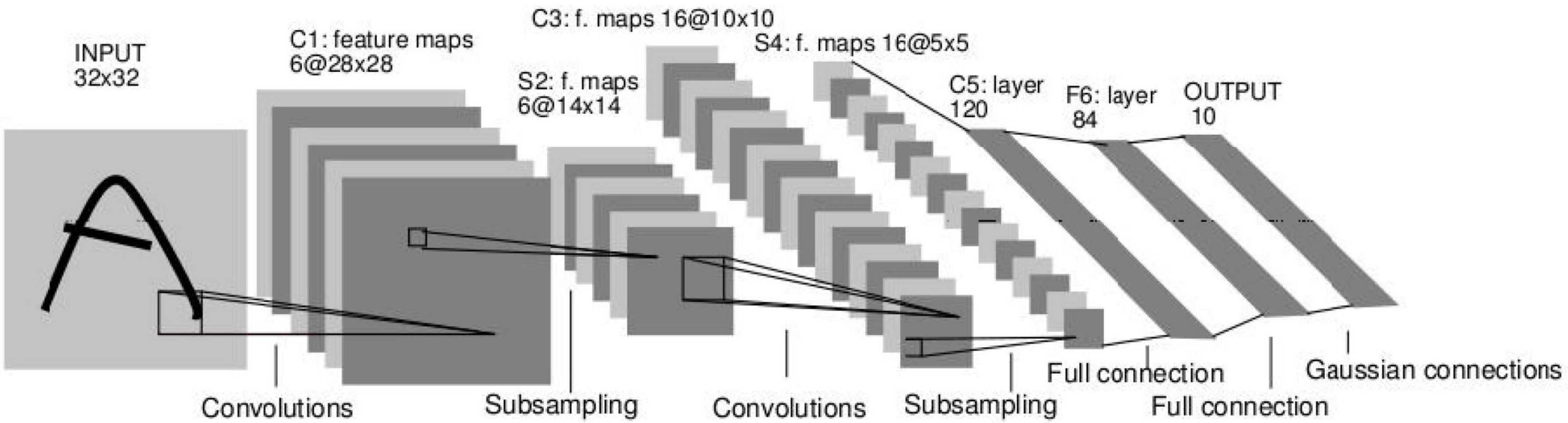
-3
-1
0
-5
1
-1
2
1
Flattened
3
-1
0
-5
1
-1
2
1
Fully Connected Feedforward network

History

- Frank Rosenblatt, ~1957: Perceptron
  - The computing machine was connected to a camera that used 20×20 cadmium sulfide photocells to produce a 400-pixel image.
  - Was able to recognize alphabets.

LeNet-5 (LeCun, 1998)

- The original CNN model introduced in1989 (LeCun)



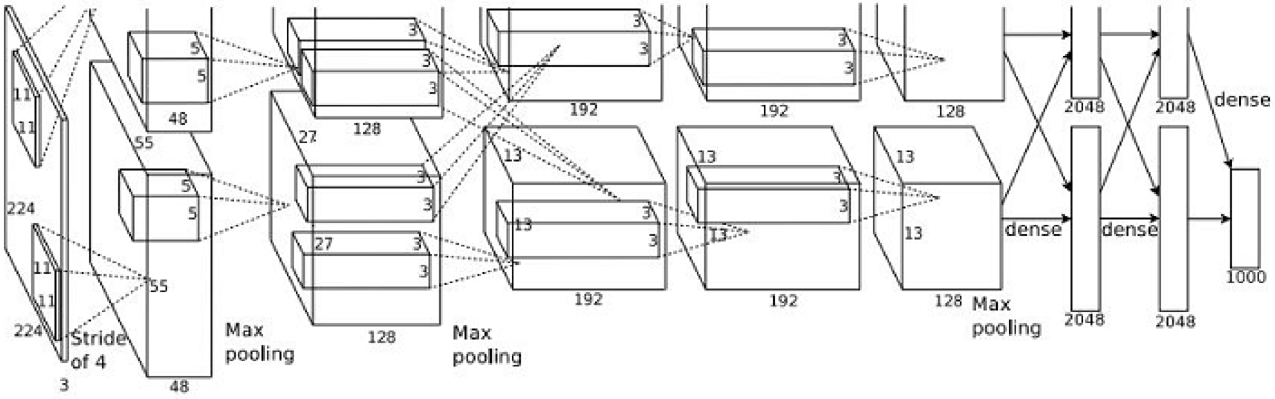
History…

- Major Breakthrough ~ 2012 –AlexNet
  - ImageNet Classification with Deep Convolutional Neural Networks
    - [Krizhevsky, Sutskever, Hinton, 2012]

Prof. Fei Fei Li's Lecture slides, Stanford University

AlexNet Architecture

- CONV1
- MAX POOL1
- NORM1
- CONV2
- MAX POOL2
- NORM2
- CONV3
- CONV4
- CONV5
- Max POOL3
- FC6
- FC7
- FC8



ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

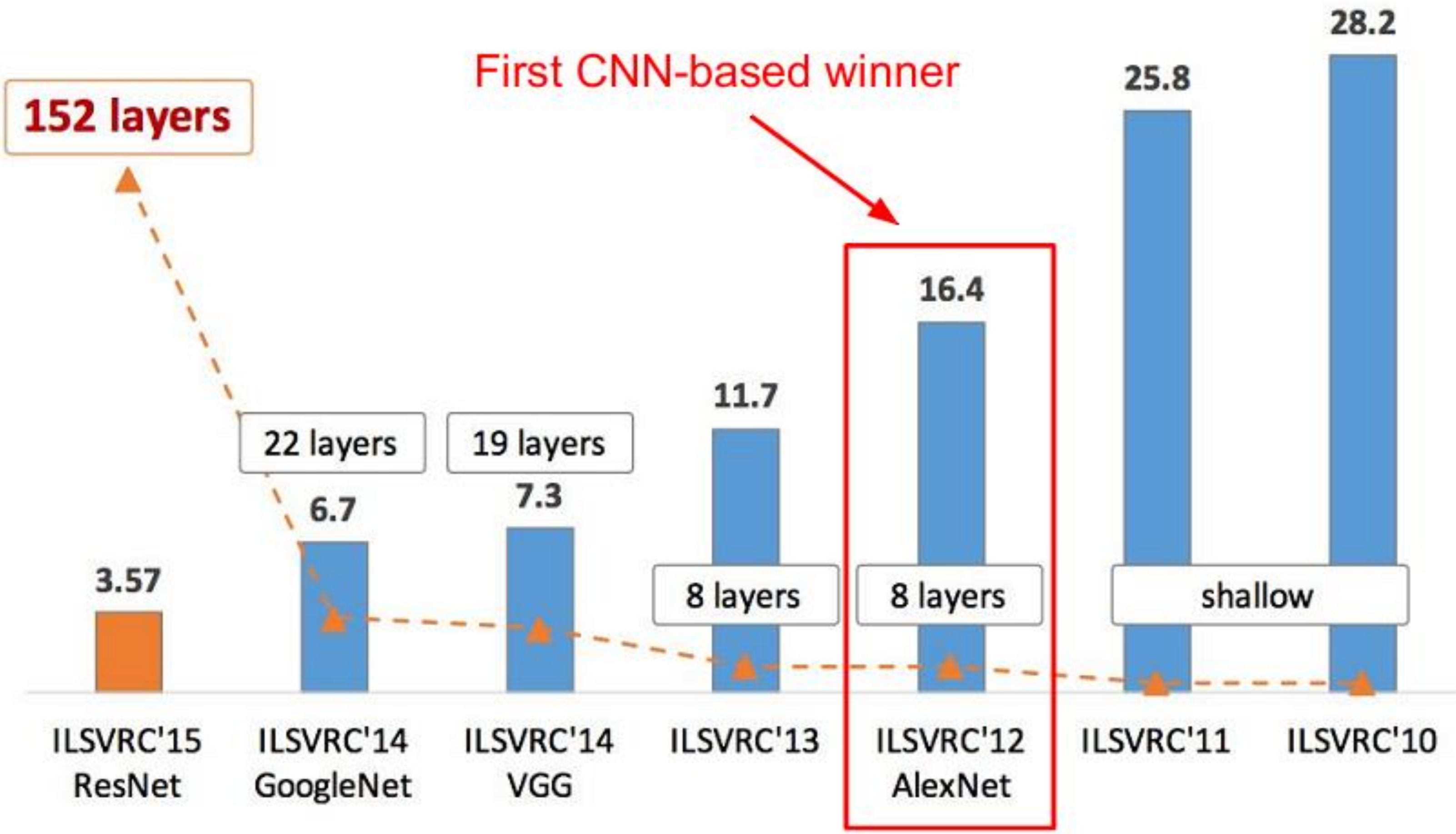


Figure copyright Kaiming He, 2016.

CNN Architectures

- Case Studies
  - AlexNet - Krizhevsky, Sutskever, Hinton, 2012
    - 8 layers
  - VGG - Simonyan and Zisserman, 2014
    - 16-19 layers
  - GoogLeNet - Szegedy et al., 2014
    - 22 layers
  - ResNet – He et al., 2015
    - 152-layers
- Also....
  - NiN (Network in Network)
  - Wide ResNet
  - ResNeXT
  - Stochastic Depth
  - -dense network
  - -fractal network
  - -sqeezeNet

Applications

- AlphaGo

19 x 19 matrix

Black: 1

white: -1

none: 0

Neural Network

Next move

(19 x 19 positions)

Fully-connected feedforward network can be used

But CNN performs much better

AlphaGo's Policy Network

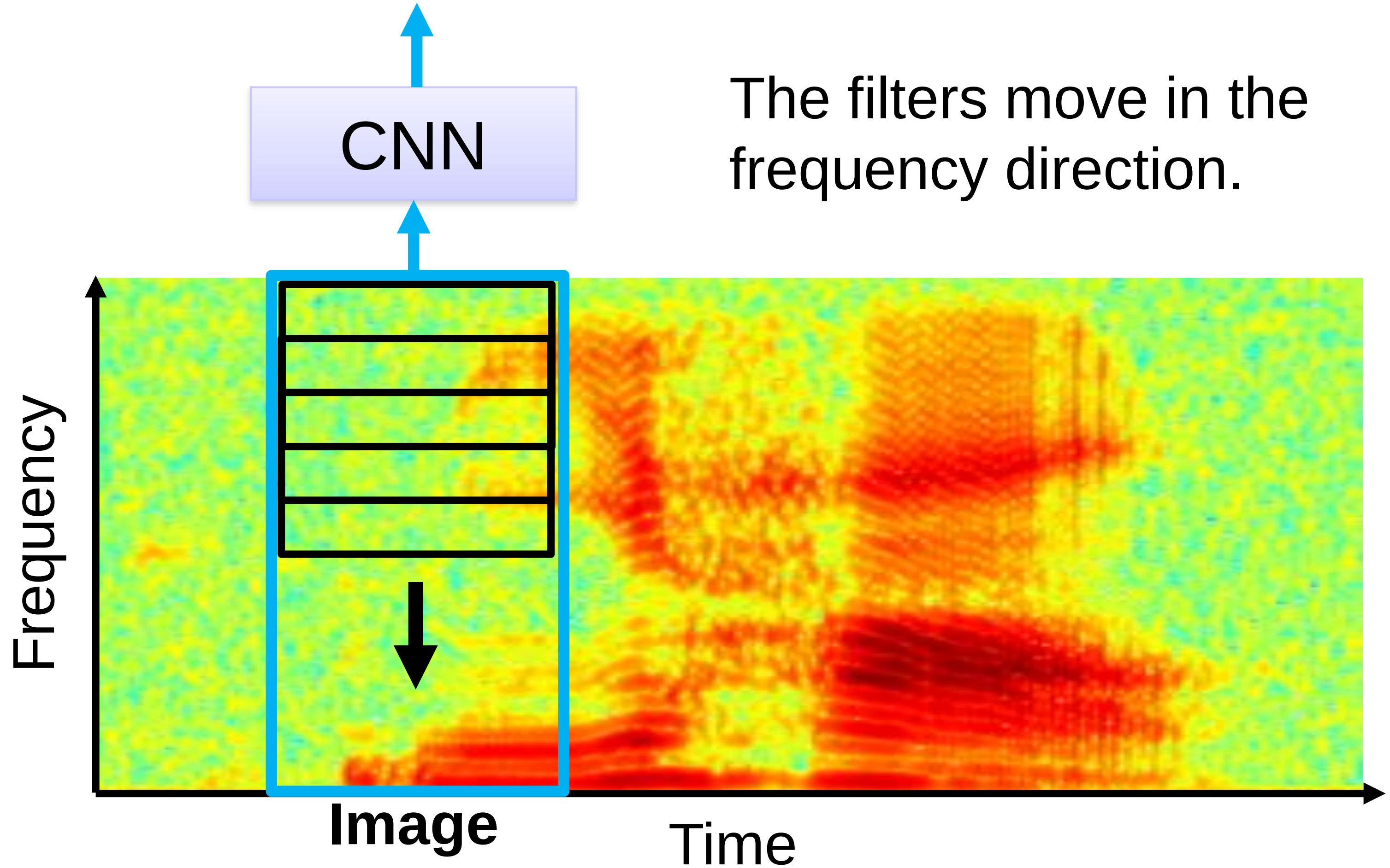
The following is quotation from their Nature article:

Note: AlphaGo does not use Max Pooling.

Neural network architecture. The input to the policy network is a 19 × 19 × 48 image stack consisting of 48 feature planes. The first hidden layer zero pads the input into a 23 × 23 image, then convolves *k* filters of kernel size 5 × 5 with stride 1 with the input image and applies a rectifier nonlinearity. Each of the subsequent hidden layers 2 to 12 zero pads the respective previous hidden layer into a 21 × 21 image, then convolves *k* filters of kernel size 3 × 3 with stride 1, again followed by a rectifier nonlinearity. The final layer convolves 1 filter of kernel size 1 × 1 with stride 1, with a different bias for each position, and applies a softmax function. The match version of AlphaGo used $k = 192$ filters; Fig. 2b and Extended Data Table 3 additionally show the results of training with $k = 128$, 256 and 384 filters.

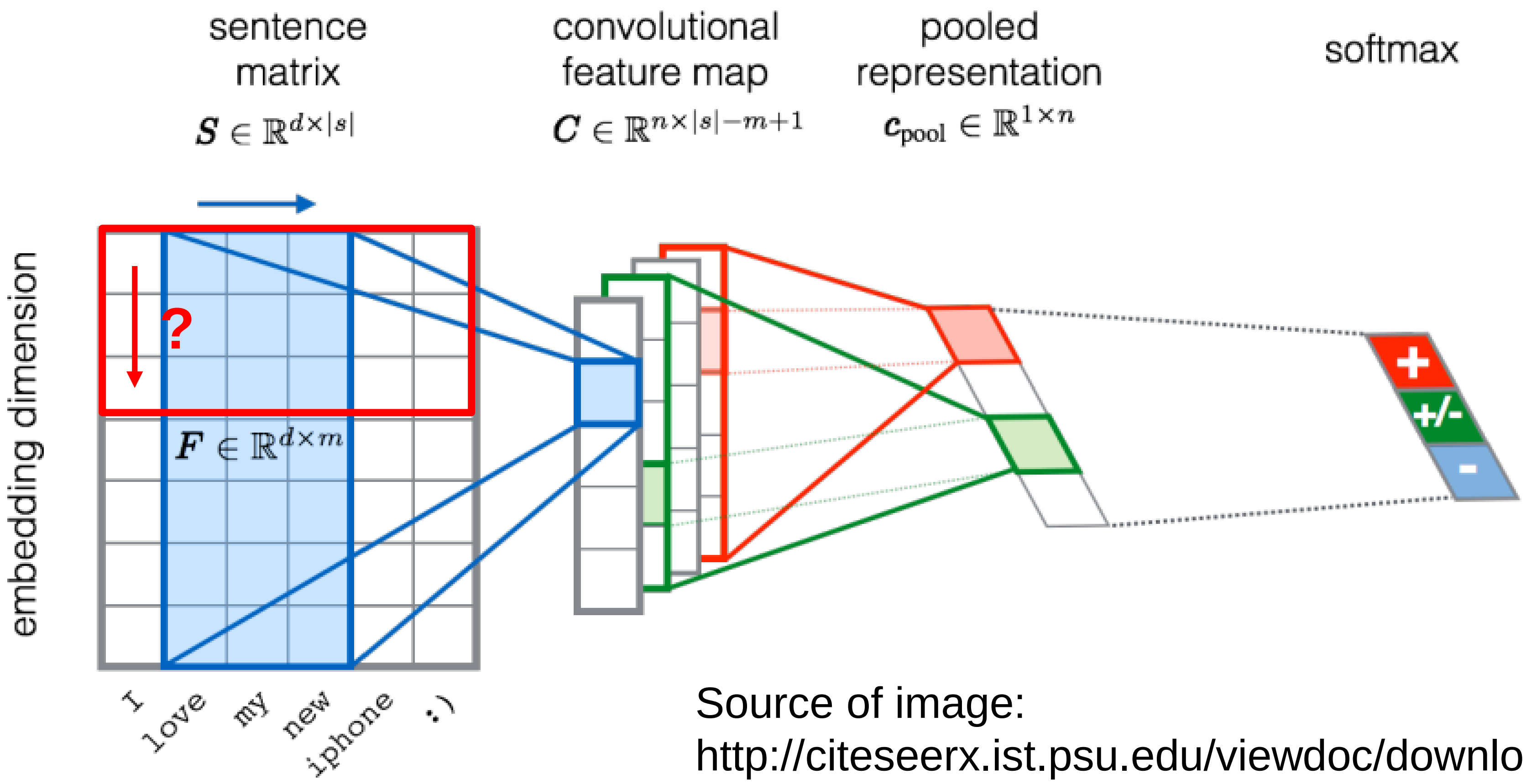
CNN in Speech Recognition

The filters move in the frequency direction.



Spectrogram

CNN in Text Classification



Source of image: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.703.6858&rep=rep1&type=pdf

Acknowledgement

Thanks to

Prof. Ming Li, University of Waterloo,Canada
Prof. Fei Fei Li's, Stanford University

For permitting to use their lecture slides.