Machine Learning

CS 539

Worcester Polytechnic Institute

Department of Computer Science

Instructor: Prof. Kyumin Lee

Upcoming Schedule

- Project Proposal
  - https://canvas.wpi.edu/courses/58900/assignments/355606
  - Due date is June 25

- HW 3
  - https://canvas.wpi.edu/courses/58900/assignments/356655
  - Due date is July 2

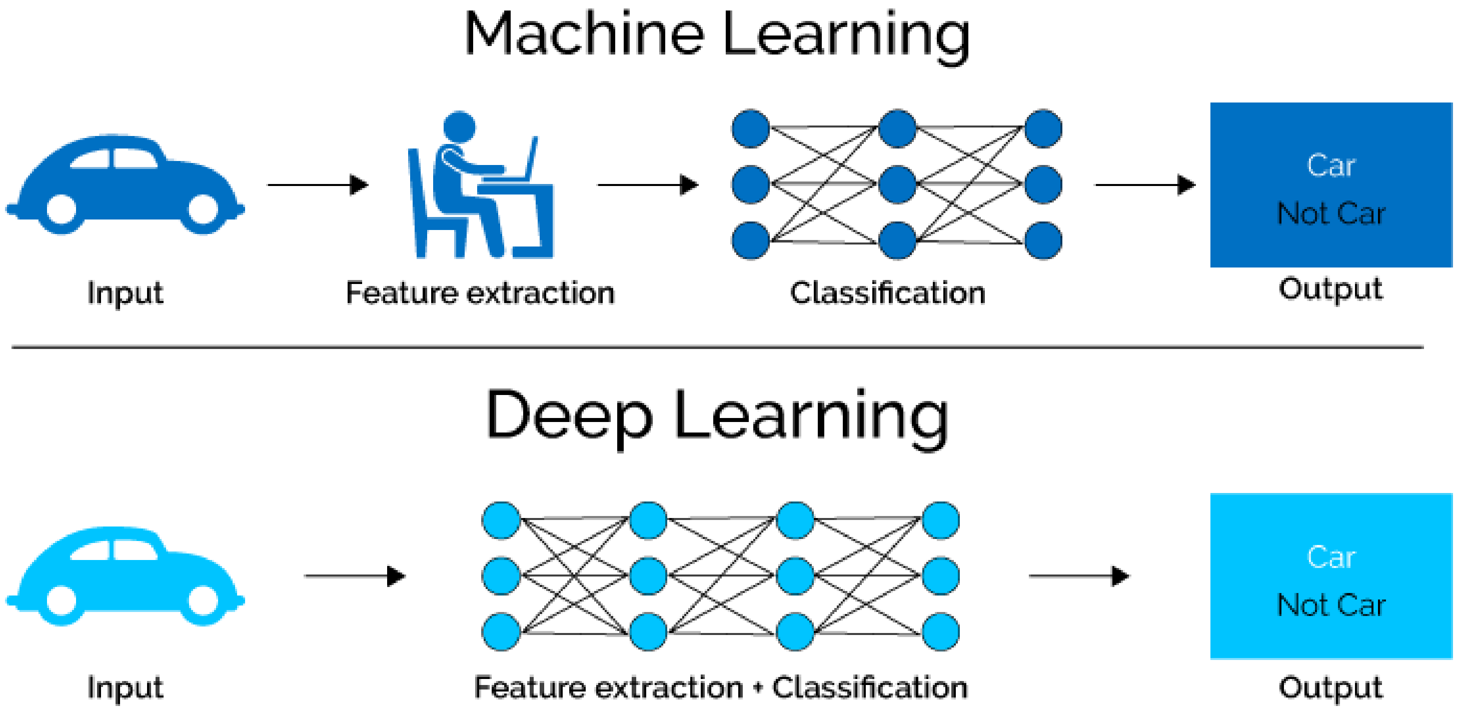
Deep Learning

What is Deep Learning (DL) ?

A machine learning subfield of learning **representations** of data. Exceptional effective at **learning patterns**.

Deep learning algorithms attempt to learn (multiple levels of) representation by using a **hierarchy of multiple layers**

If you provide the system **tons of information**, it begins to understand it and respond in useful ways.



Machine Learning

Input → Feature extraction → Classification → Output

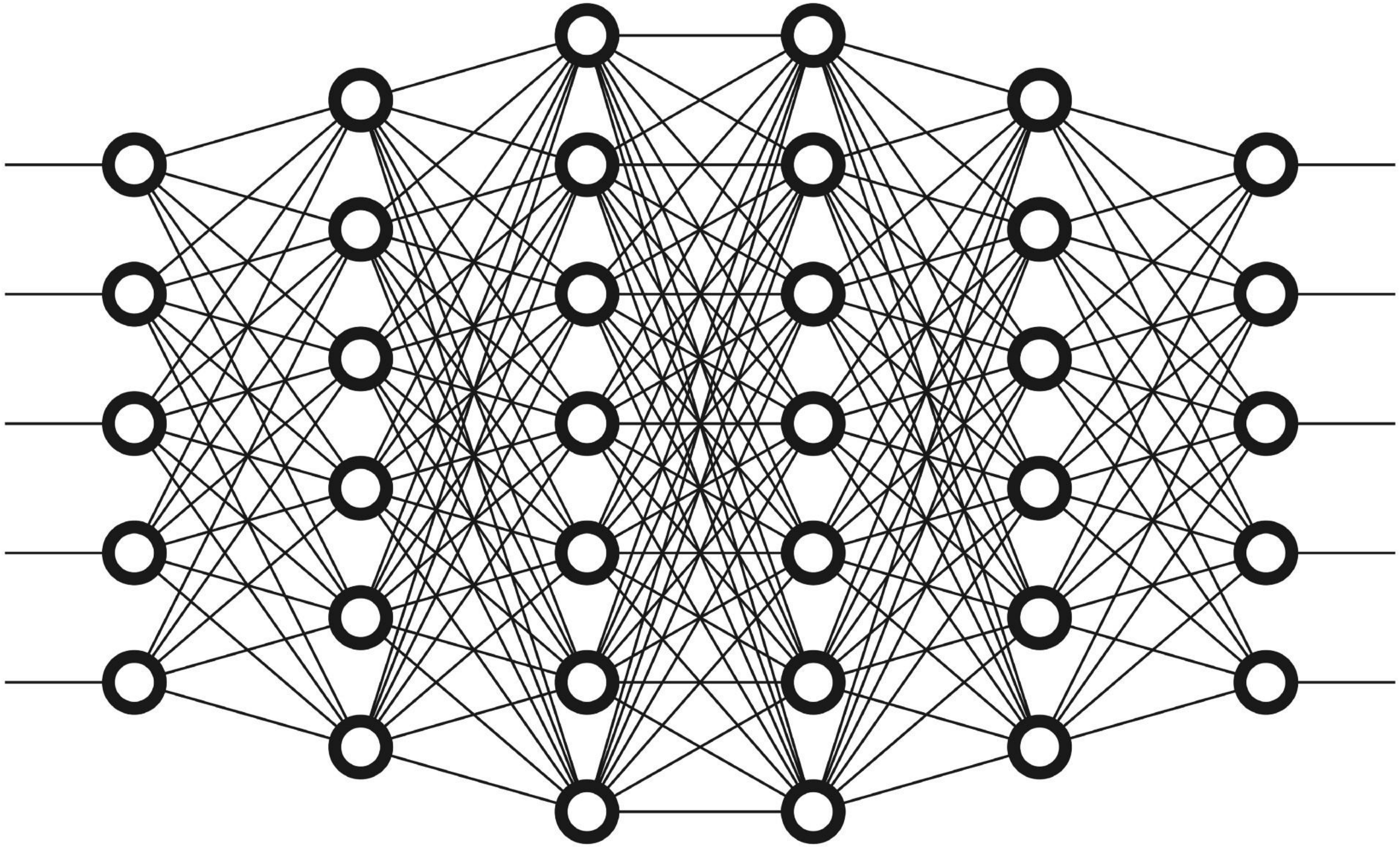
Car
Not Car

Deep Learning

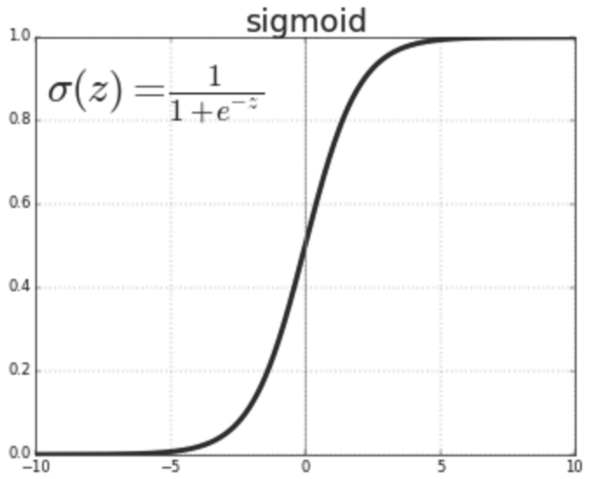
Input → Feature extraction + Classification → Output

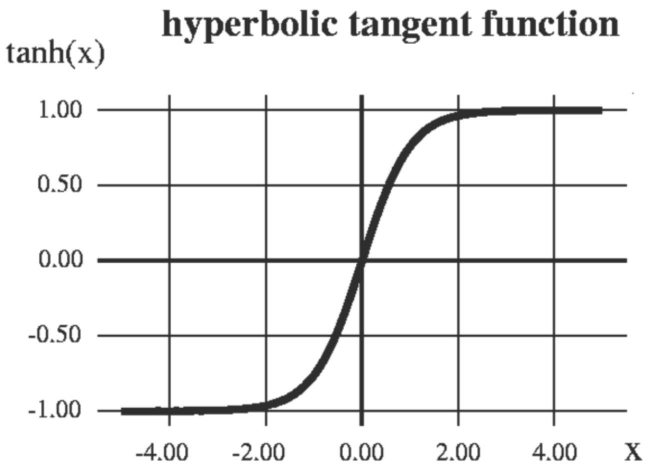
Car
Not Car

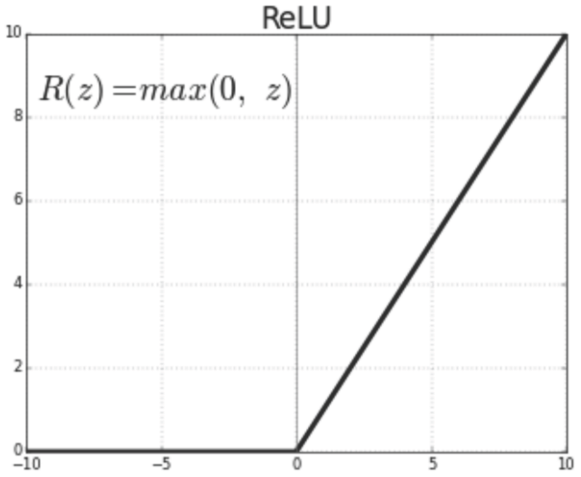
Deep Neural Network



Deep Net Activation Functions

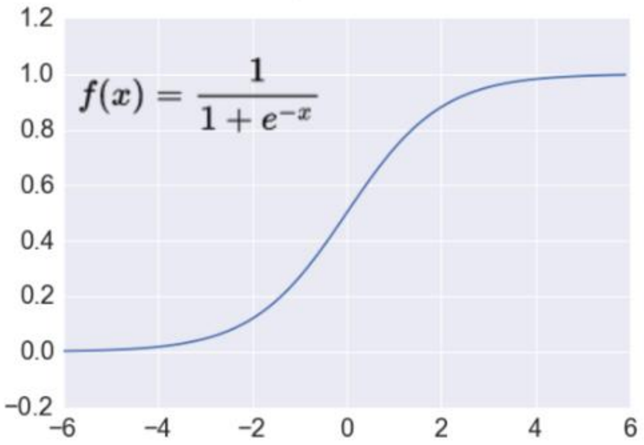






Rectified Linear Unit (ReLU)

Activation: Sigmoid



http://adilmoujahid.com/images/activation.png

Takes a real-valued number and "squashes" it into range between 0 and 1.

$$R^n \rightarrow [0,1]$$

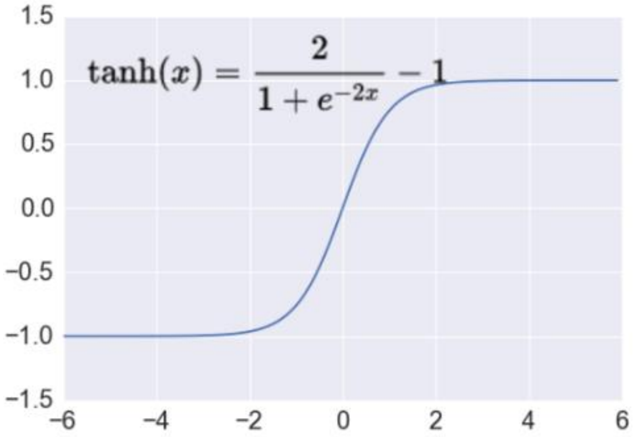
+ Nice interpretation as the **firing rate** of a neuron

- 0 = not firing at all
- 1 = fully firing

- Sigmoid neurons **saturate** and **kill gradients**, thus NN will barely learn

- when the neuron's activation are 0 or 1 (saturate)
  - 🙁 gradient at these regions almost zero
  - 🙁 almost no signal will flow to its weights
  - 🙁 if initial weights are too large then most neurons would saturate

Activation: Tanh



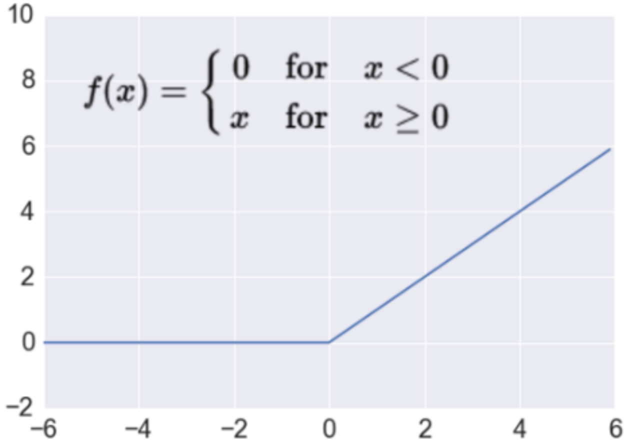
http://adilmoujahid.com/images/activation.png

Takes a real-valued number and "squashes" it into range between -1 and 1.

$$R^n \rightarrow [-1,1]$$

- Like sigmoid, tanh neurons **saturate**
- Unlike sigmoid, output is **zero-centered**
- Tanh is a **scaled sigmoid**: $\tanh(x) = 2sigm(2x) - 1$

Activation: ReLU



http://adilmoujahid.com/images/activation.png

Takes a real-valued number and thresholds it at zero $\mathrm{f}(x) = \max(0, x)$

$$R^n \rightarrow R_+^n$$

Most Deep Networks use ReLU nowadays

🙂 Trains much **faster**

- accelerates the convergence of SGD
- due to linear, non-saturating form

🙂 Less expensive operations

- compared to sigmoid/tanh (exponentials etc.)
- implemented by simply thresholding a matrix at zero

🙂 Prevents the **gradient vanishing problem**

Hardware Power

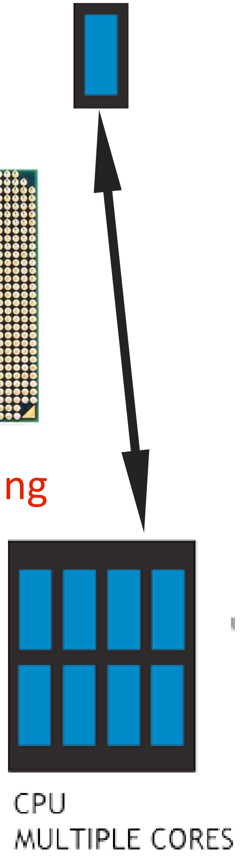
a single-core 2.5 GHz CPU has a theoretical performance of 10 billion FLOPS = 10 GFLOPS

1 CPU core = 10 GFlops

1 multi-core Hyper-threading CPU

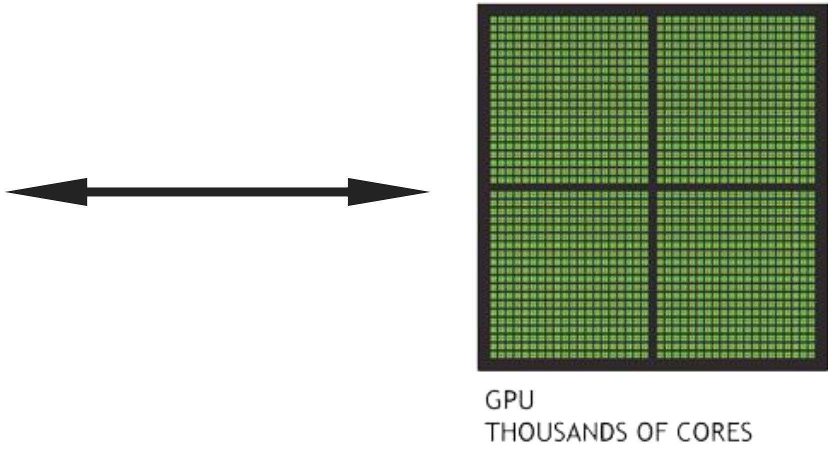
= 8 Cores X 2 thread

= 16 CPUs



Graphics processing unit

1 GPU



What is Flops?

FLoating-point Operations Per Second

measure of computer performance, useful in fields of scientific computations that require floating-point calculations

Loan: 325,000

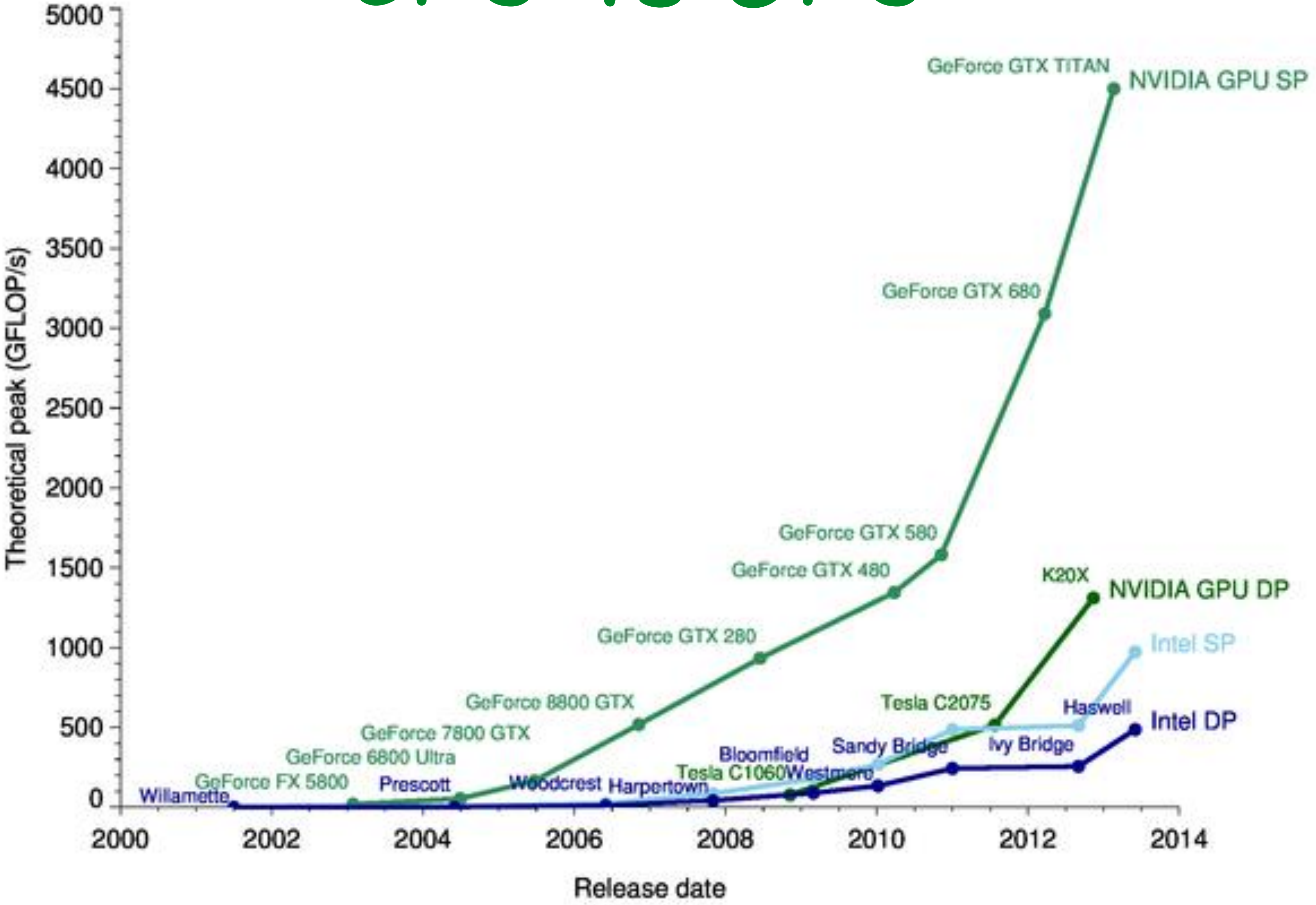
Interest Rate(APR): 2.225%

Term: 1 year

Interest Paid: ? 7,312.5

https://en.wikipedia.org/wiki/FLOPS

CPU vs GPU


Theoretical peak (GFLOP/s)
5000
4500
4000
3500
3000
2500
2000
1500
1000
500
0
2000
2002
2004
2006
2008
2010
2012
2014
Release date
GeForce GTX TITAN
NVIDIA GPU SP
GeForce GTX 680
GeForce GTX 580
GeForce GTX 480
K20X
NVIDIA GPU DP
Intel SP
GeForce GTX 280
GeForce 8800 GTX
Tesla C2075
Haswell
Intel DP
GeForce 7800 GTX
Sandy Bridge
Ivy Bridge
GeForce 6800 Ultra
Bloomfield
Tesla C1060
Westmere
GeForce FX 5800
Prescott
Woodcrest
Harpertown
Willamette

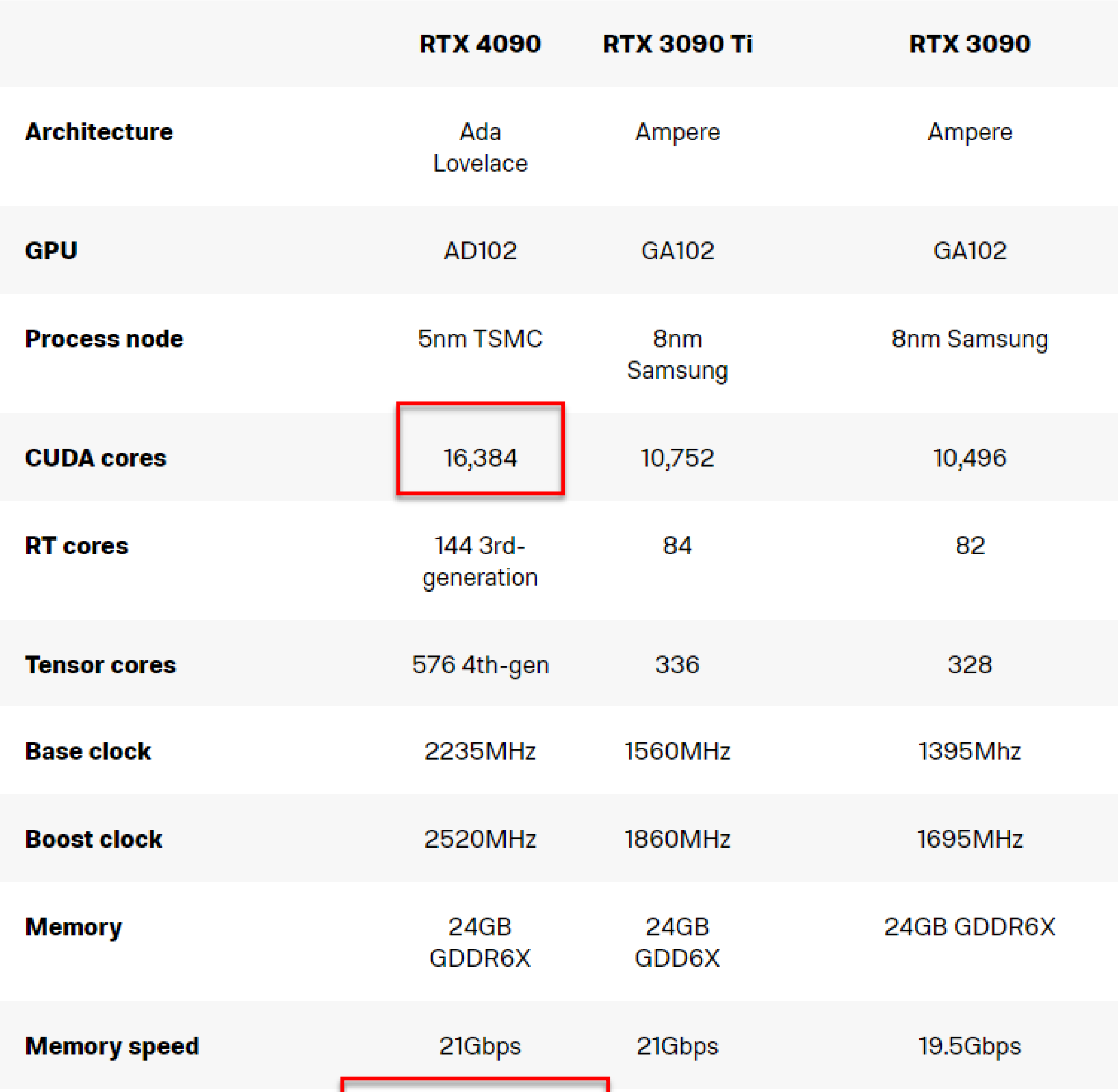
GeForce RTX 4090

BEYOND FAST

The NVIDIA® GeForce RTX™ 4090 is the ultimate GeForce GPU. It brings an enormous leap in performance, efficiency, and AI-powered graphics. Experience ultra-high performance gaming, incredibly detailed virtual worlds, unprecedented productivity, and new ways to create. It's powered by the NVIDIA Ada Lovelace architecture and comes with 24 GB of G6X memory to deliver the ultimate experience for gamers and creators.

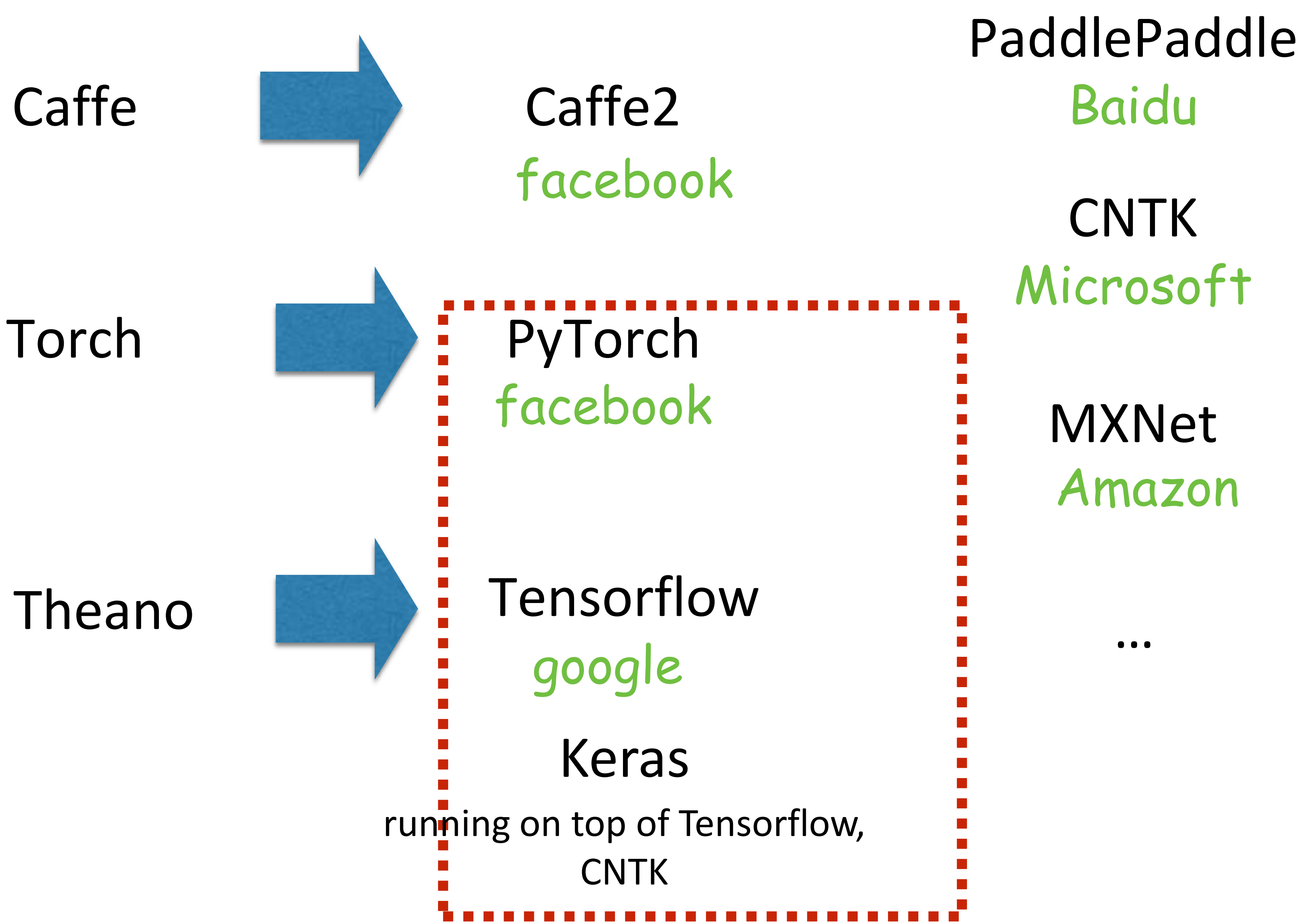
Starting at $1599.00

See All Buying Options
RTX 4090



	RTX 4090	RTX 3090 Ti	RTX 3090
Architecture	Ada Lovelace	Ampere	Ampere
GPU	AD102	GA102	GA102
Process node	5nm TSMC	8nm Samsung	8nm Samsung
CUDA cores	16,384	10,752	10,496
RT cores	144 3rd-generation	84	82
Tensor cores	576 4th-gen	336	328
Base clock	2235MHz	1560MHz	1395Mhz
Boost clock	2520MHz	1860MHz	1695MHz
Memory	24GB GDDR6X	24GB GDD6X	24GB GDDR6X
Memory speed	21Gbps	21Gbps	19.5Gbps
	83 TFLOPS	40 TFLOPS	36 TFLOPS

Deep Learning Software



Caffe → Caffe2
facebook

Torch → PyTorch
facebook

Theano → Tensorflow
google

Keras
running on top of Tensorflow,
CNTK

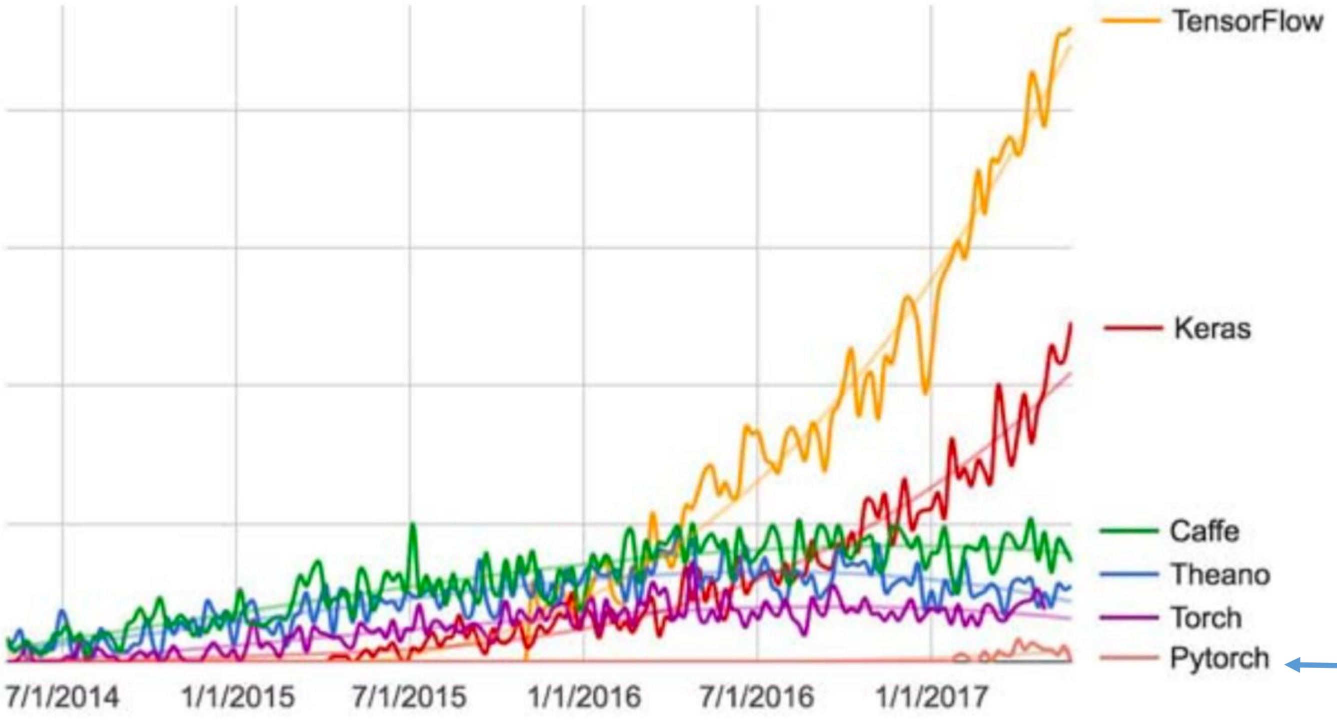
PaddlePaddle
Baidu

CNTK
Microsoft

MXNet
Amazon

...

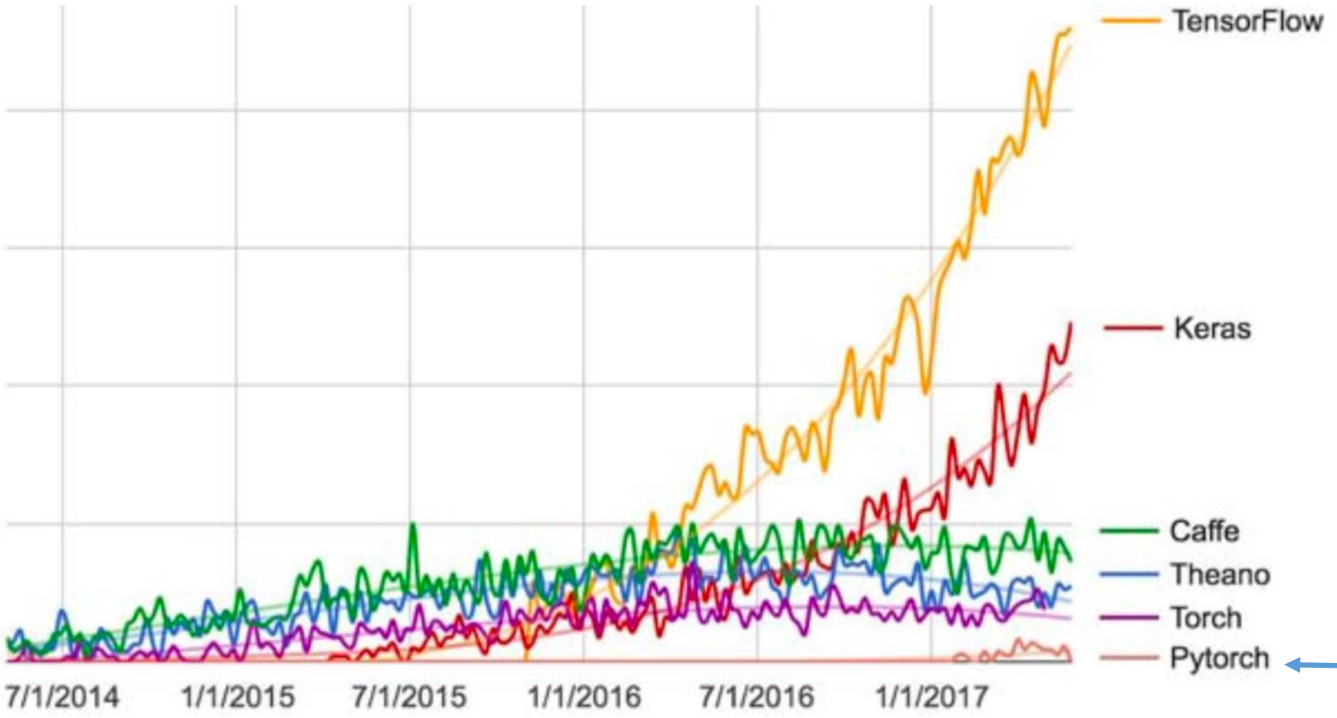
Deep Learning Software



PyTorch is getting popular today

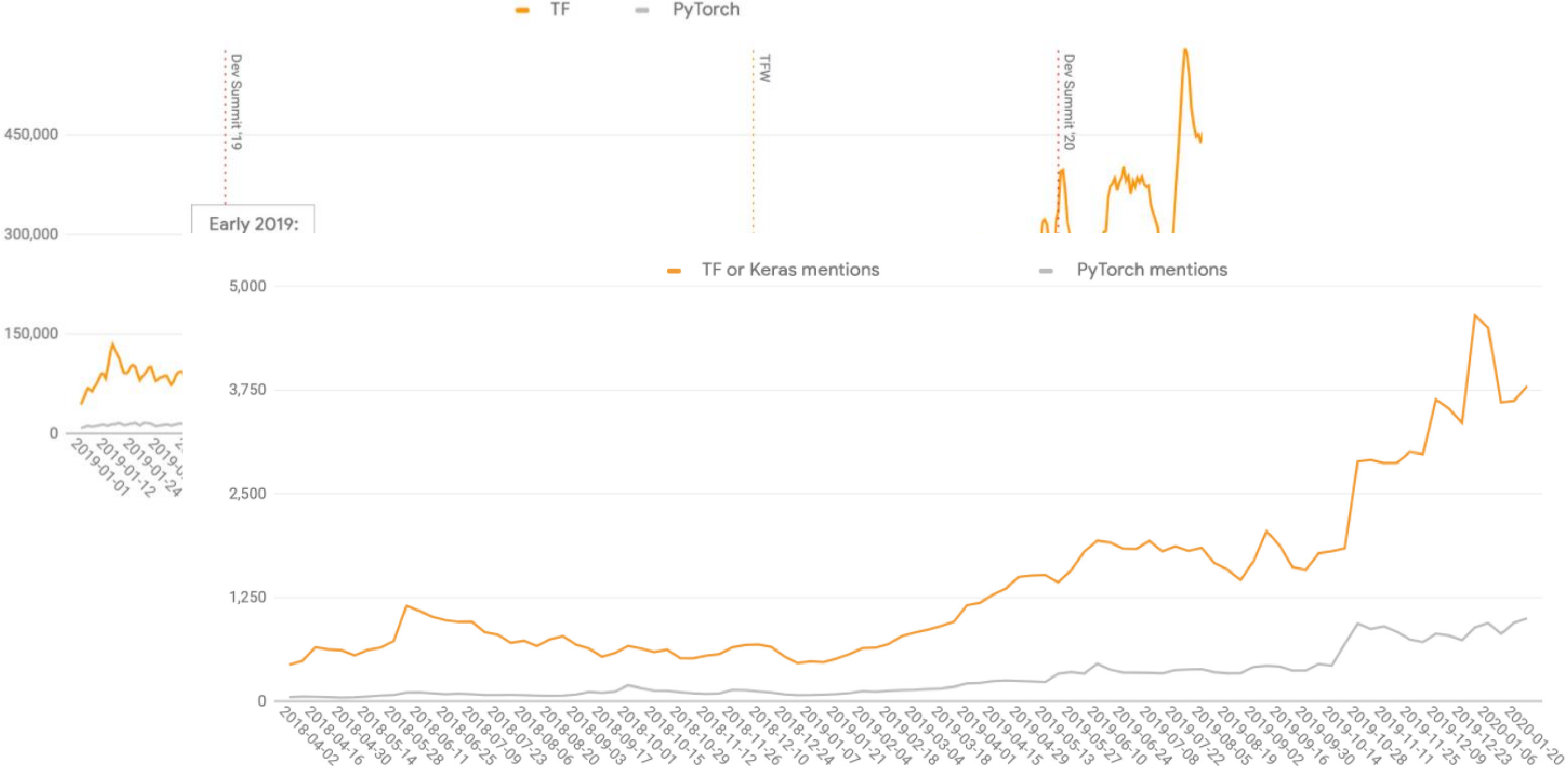
Figure 3.2 **Google web search interest for different deep-learning frameworks over time**

Deep Learning Software



PyTorch
is getting
popular
today

Figure 3.2 **Google web search interest for different deep-learning frameworks over time**



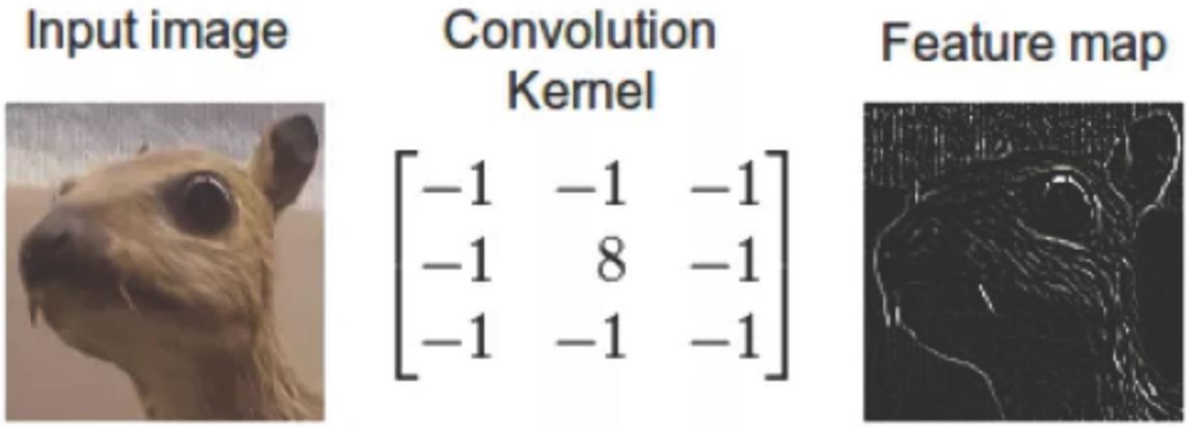
Number of articles mentioning deep learning frameworks indexed on Google Scholar by week
(exponential moving average)

Convolutional Neural Network (CNN)

Convolution:

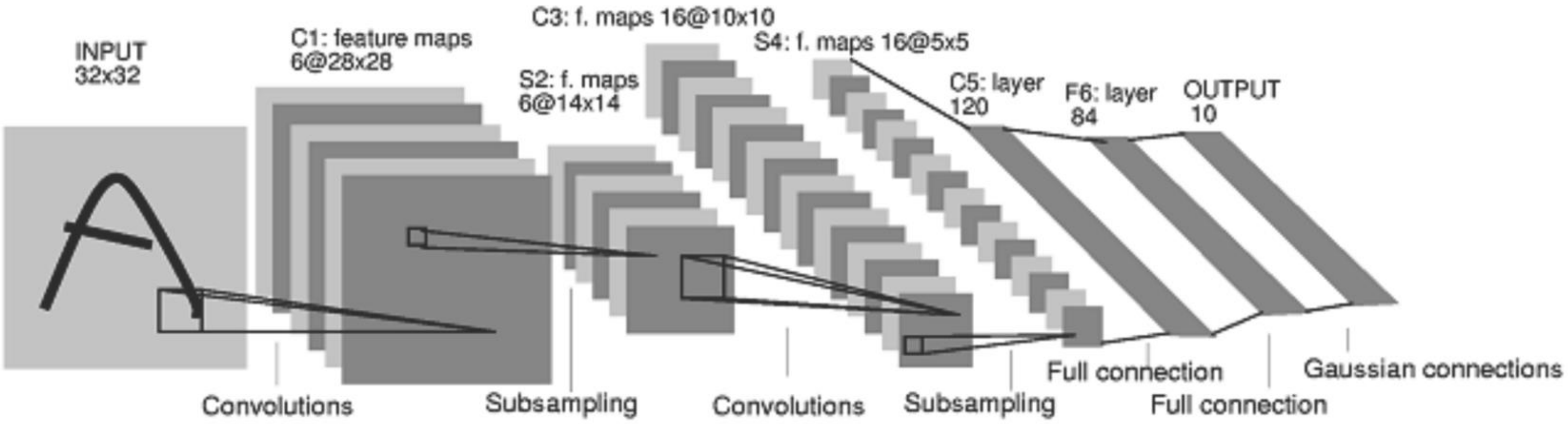
Convolution is a mathematical operation on two functions to produce a third function that expresses how the shape of one is modified by the other. It is a term that the neural network community (Hinton, specifically) adopted from the signal and image processing communities to architect the “feature detection” layers of “deep” neural networks.

Convolutional Neural Network



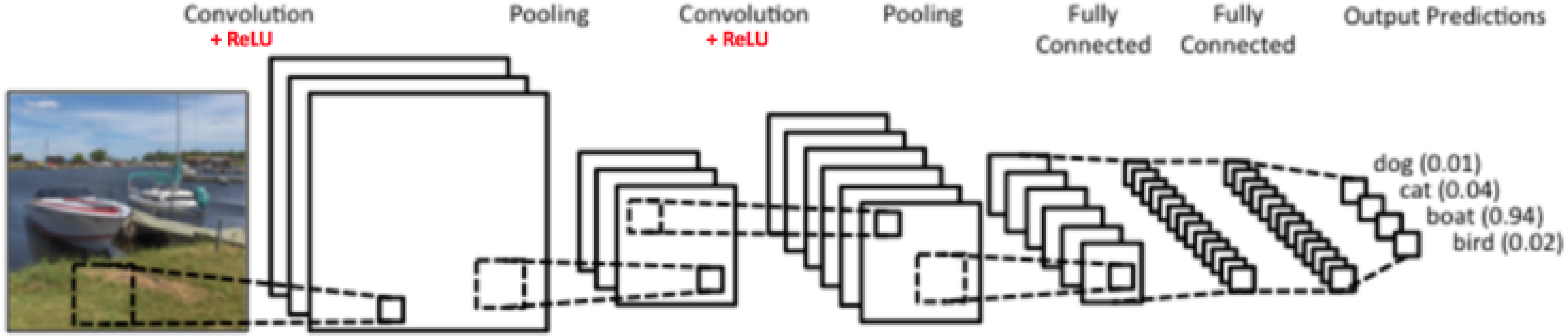
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

[Image credit: http://timdettmers.com/2015/03/26/convolution-deep-learning/]



Reading: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

Convolutional Neural Network



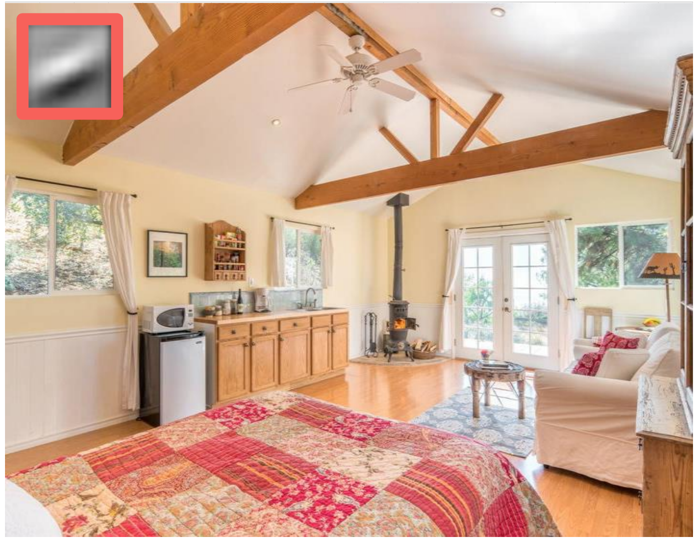
A simple ConvNet

http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

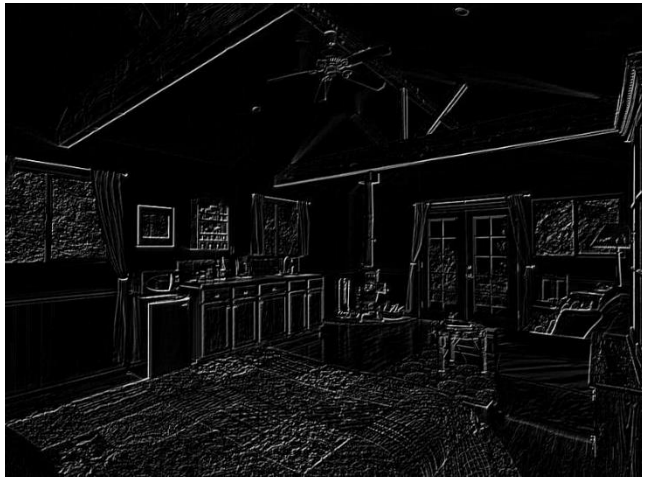
Convolutional Layer

- Filter

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{pmatrix}$$



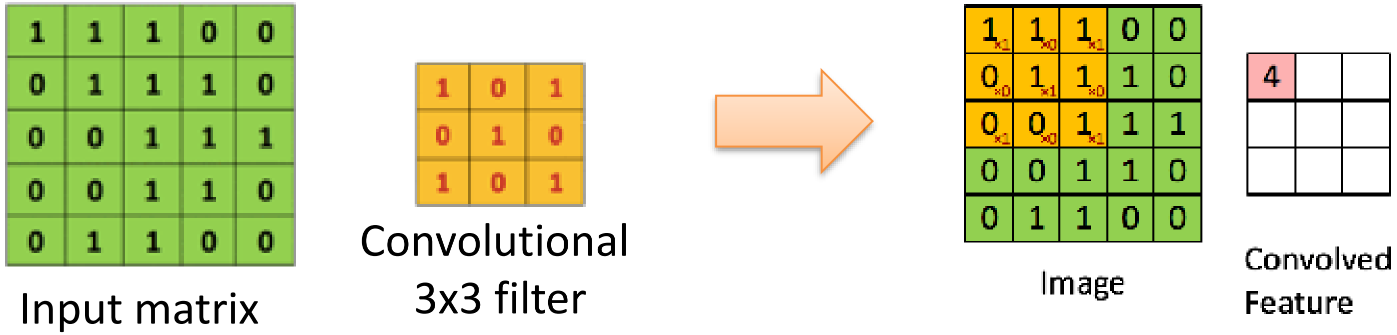
Input Image



Convoluted Image

- Inspired by the neurophysiological experiments conducted by Hubel and Wiesel 1962.

Convolutional Layer



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

Input matrix

1	0	1
0	1	0
1	0	1

Convolutional 3x3 filter

$1_{\times 1}$	$1_{\times 0}$	$1_{\times 1}$	0	0
$0_{\times 0}$	$1_{\times 1}$	$1_{\times 0}$	1	0
$0_{\times 1}$	$0_{\times 0}$	$1_{\times 1}$	1	1
0	0	1	1	0
0	1	1	0	0

Image

4		

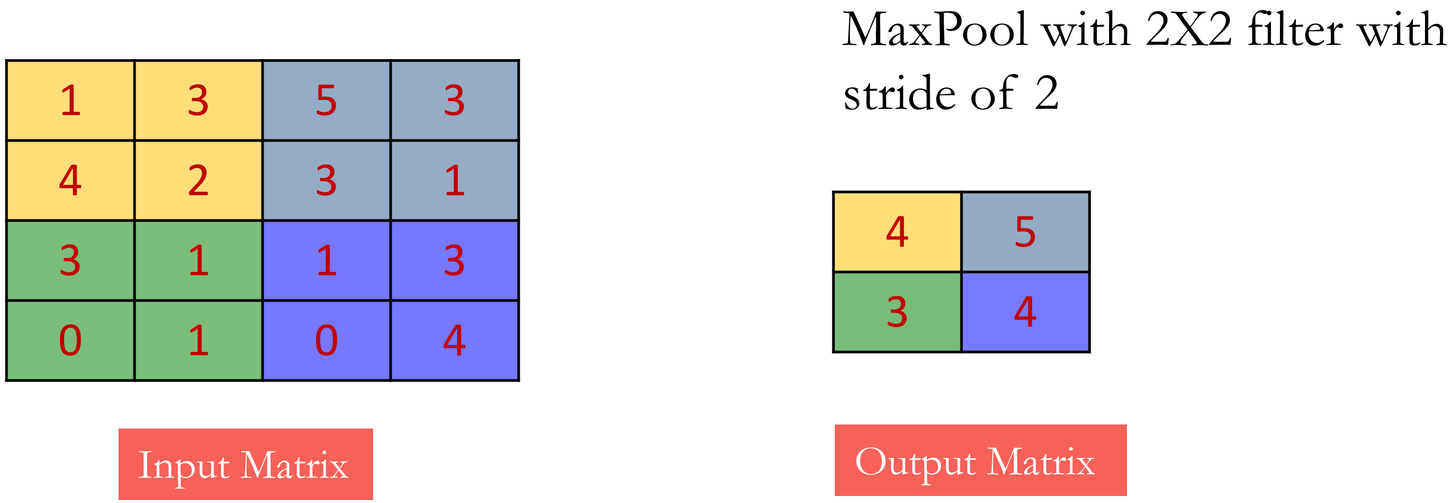
Convolved Feature

http://deeplearning.stanford.edu/wiki/index.php/Feature_extraction_using_convolution

Convolution operation captures the local dependencies in the ordinal image

Pooling Layer

- Pooling reduces the dimensionality of each feature map
  - Max pooling: reports the maximum output within a rectangular neighborhood.
  - Average pooling: reports the average output of a rectangular neighborhood.



1	3	5	3
4	2	3	1
3	1	1	3
0	1	0	4

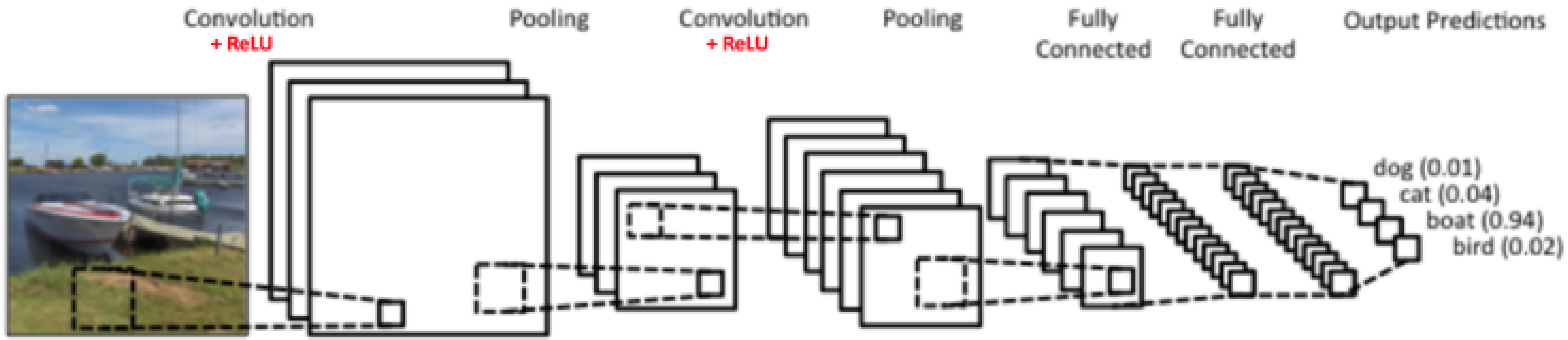
Input Matrix

MaxPool with 2X2 filter with stride of 2

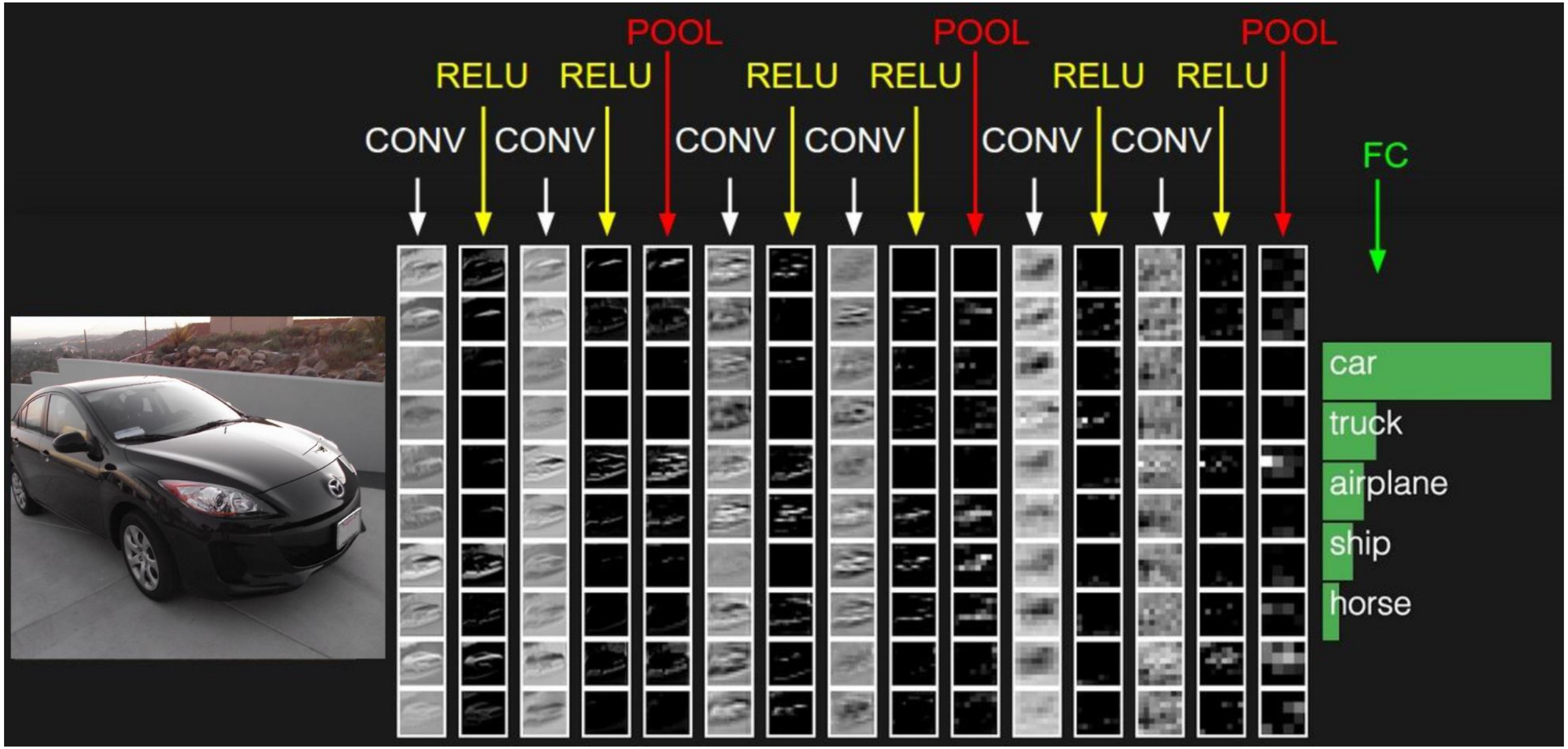
4	5
3	4

Output Matrix

Fully Connected Layers

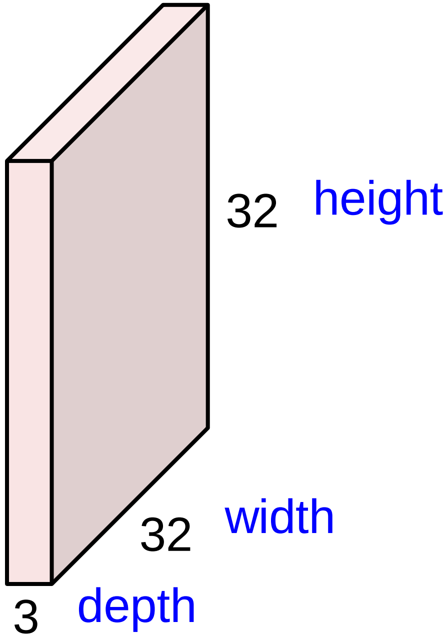


A simple ConvNet
http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/


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

Convolution Layer

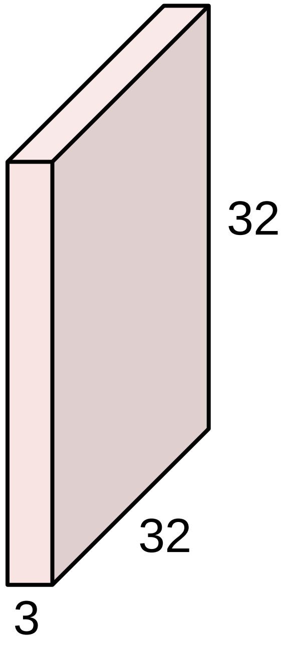
32x32x3 image -> preserve spatial structure



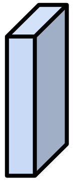
Following slides are adapted from Fei-Fei Li’s slides

Convolution Layer

32x32x3 image



5x5x3 filter



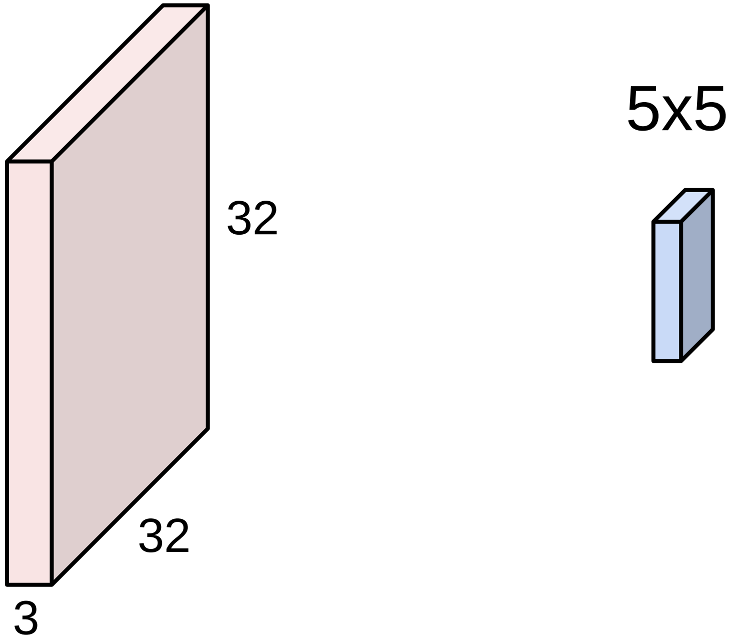
Convolve the filter with the image
i.e. “slide over the image spatially,
computing dot products”

Convolution Layer

Filters always extend the full depth of the input volume

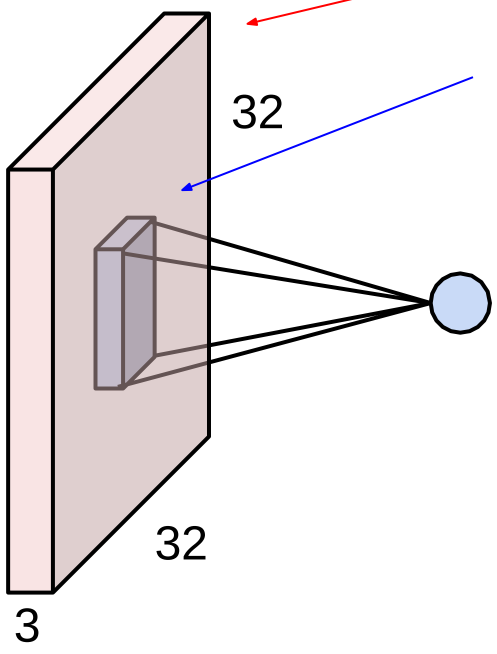
32x32x3 image

5x5x3 filter



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

Convolution Layer



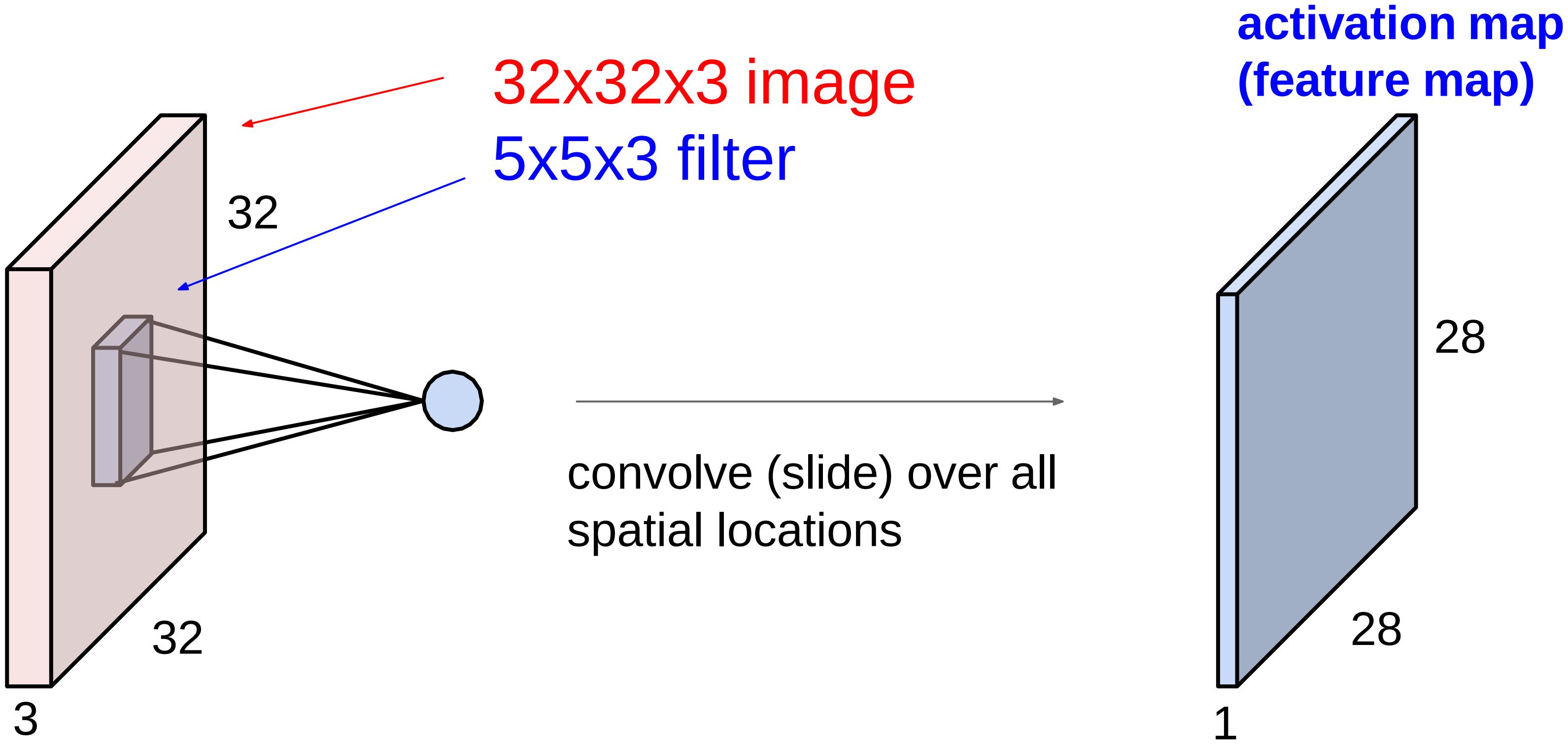
32x32x3 image

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



32x32x3 image

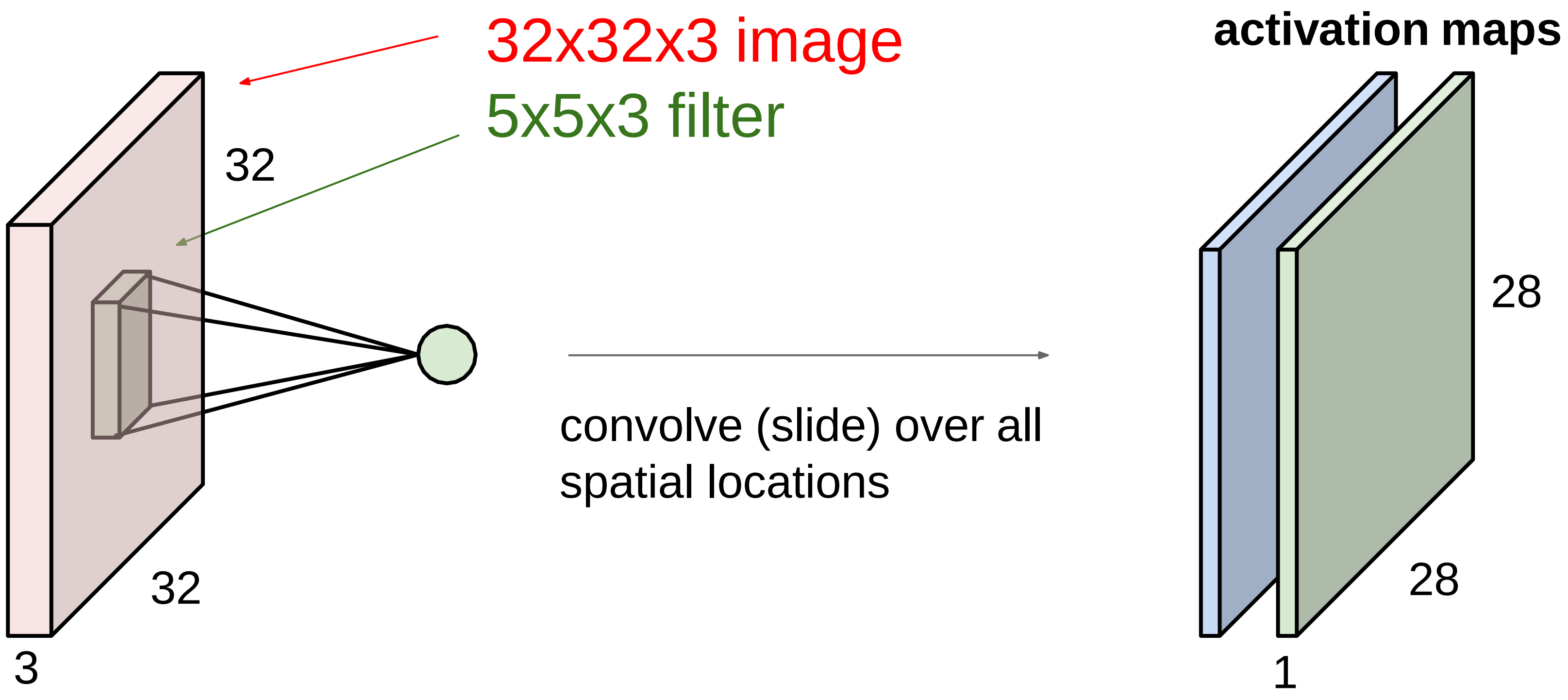
5x5x3 filter

convolve (slide) over all spatial locations

activation map (feature map)

Convolution Layer

consider a second, green filter



32x32x3 image

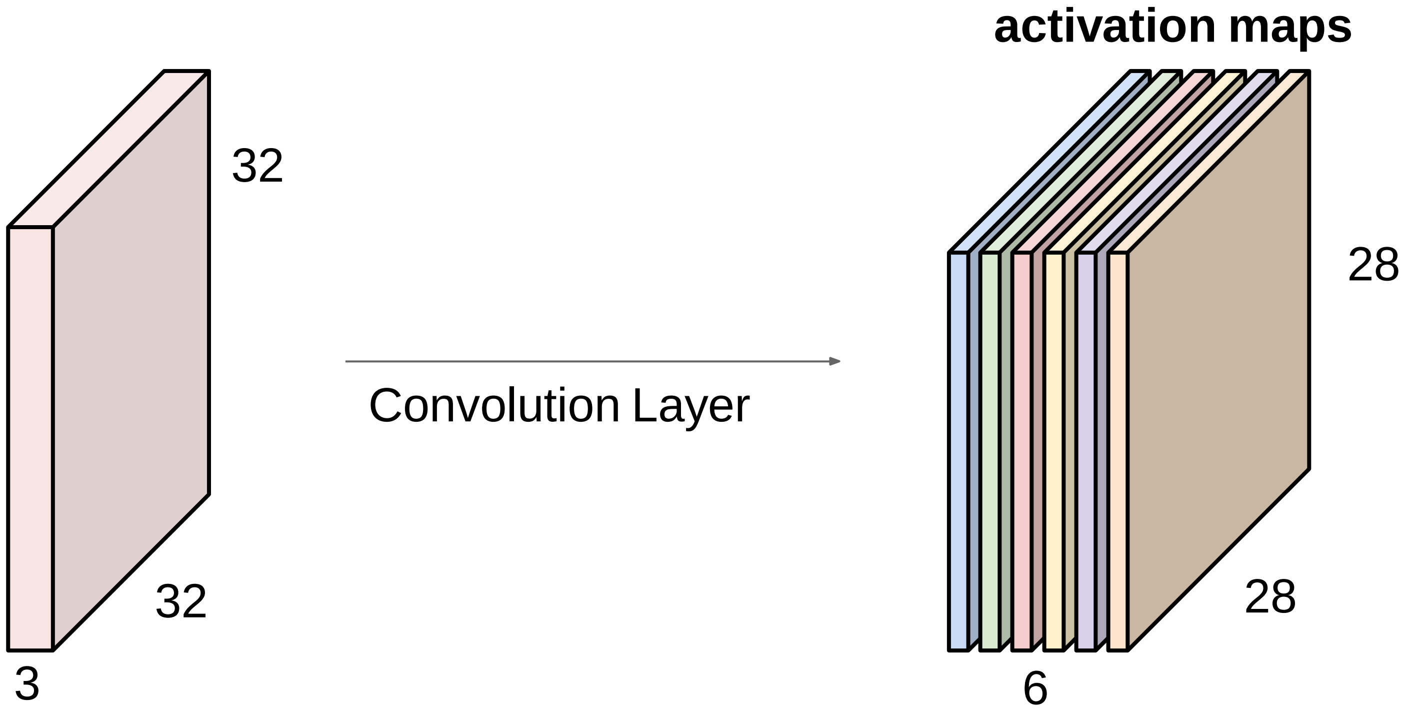
5x5x3 filter

convolve (slide) over all spatial locations

activation maps

Convolution Layer

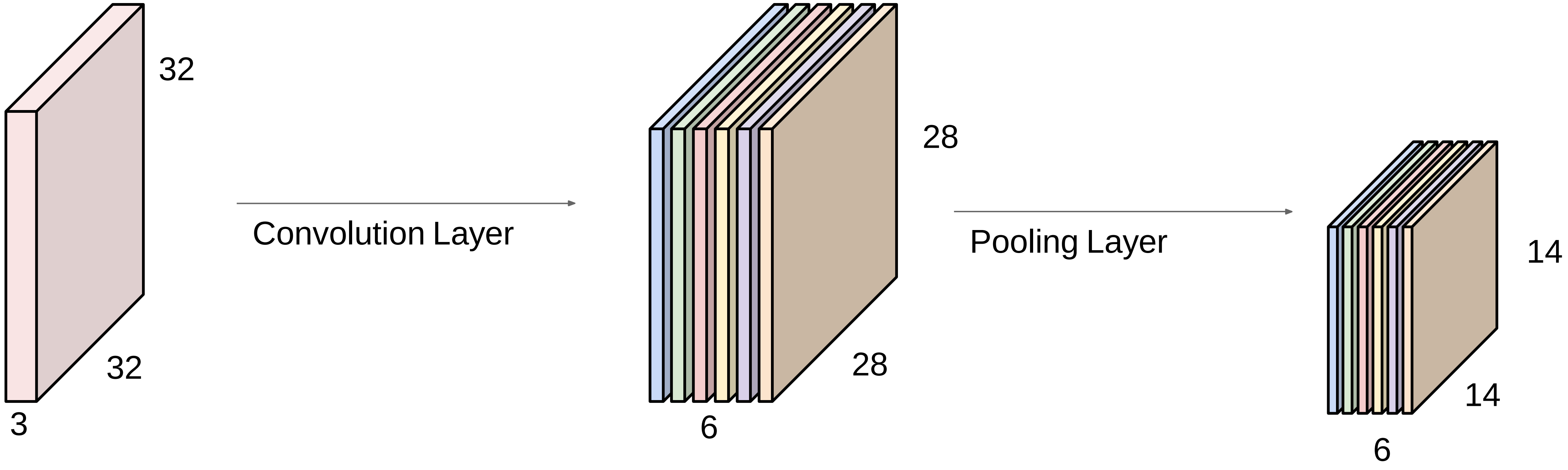
For example, if we had 6 5x5 filters, we'll get 6 separate activation maps:



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

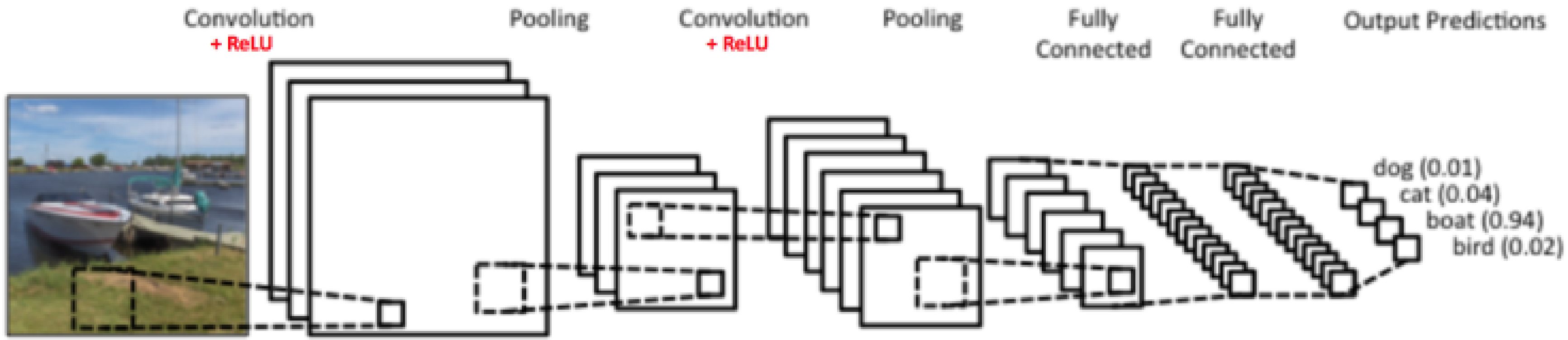
Pooling Layer

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



Assume pool with 2x2 filters and stride 2

Convolutional Neural Network



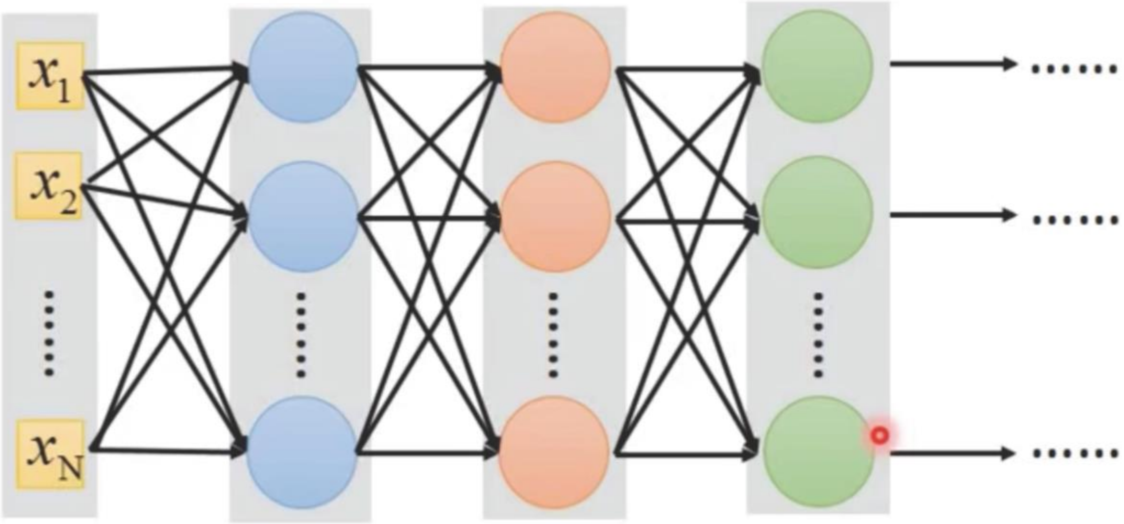
A simple ConvNet

http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

Why CNN for Image?

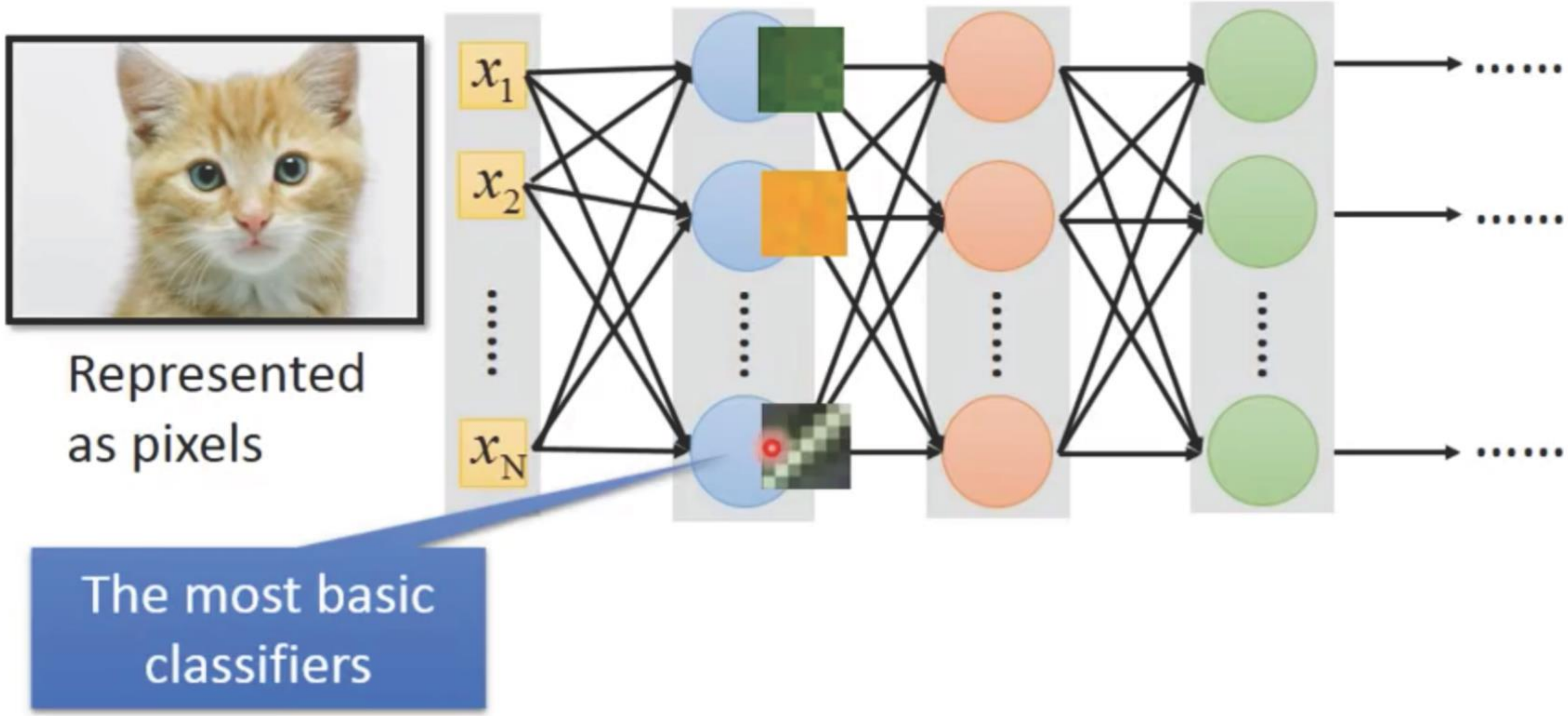
[Zeiler, M. D., *ECCV 2014*]

Represented as pixels



Why CNN for Image?

[Zeiler, M. D., *ECCV 2014*]

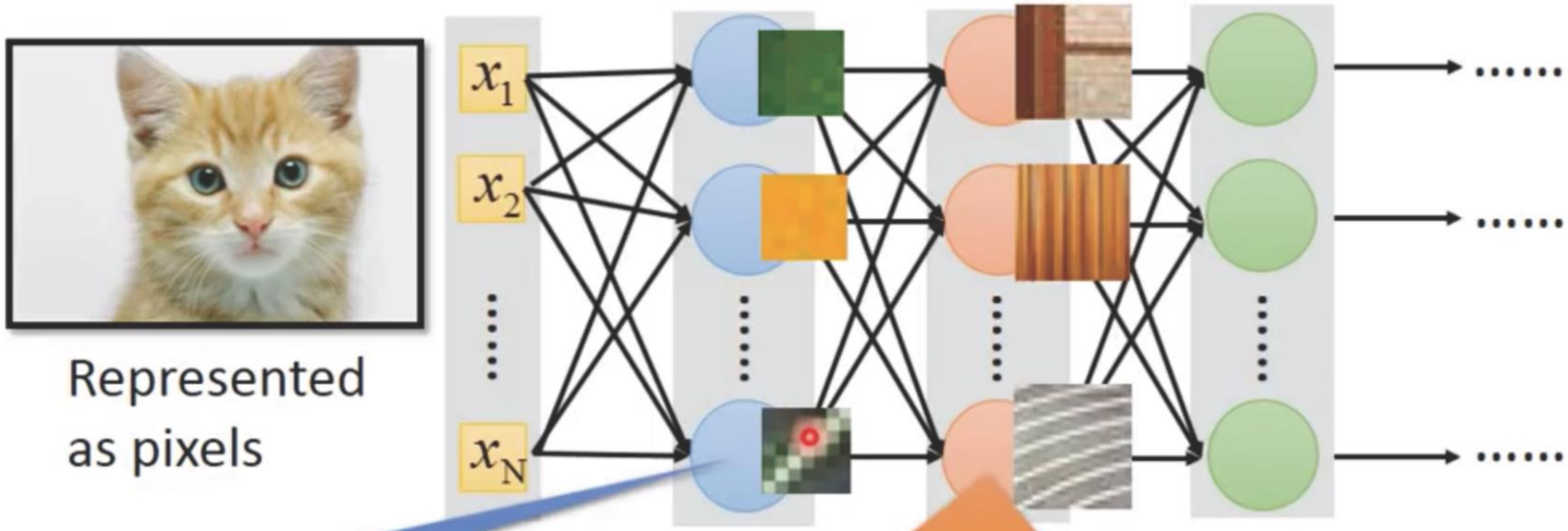


Represented as pixels

The most basic classifiers

Why CNN for Image?

[Zeiler, M. D., *ECCV 2014*]

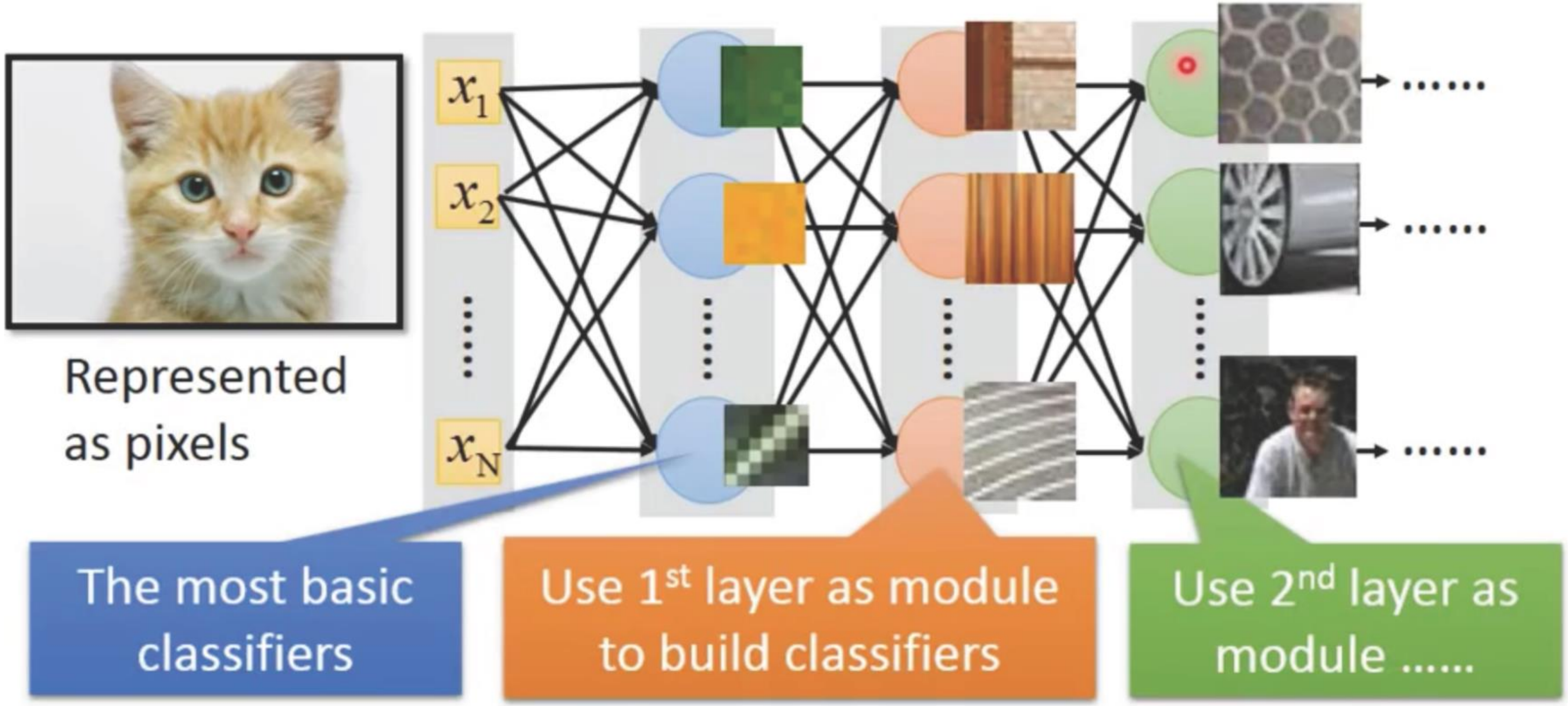


The most basic classifiers

Use 1st layer as module to build classifiers

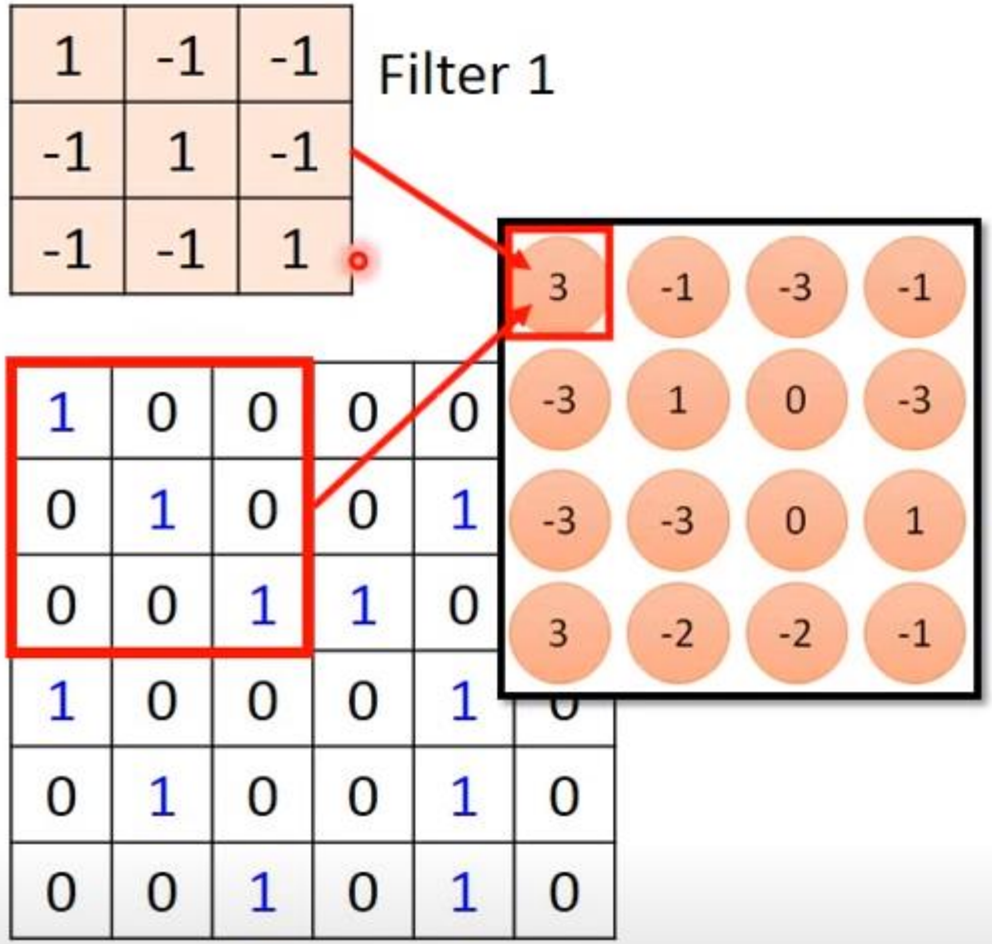
Why CNN for Image?

[Zeiler, M. D., *ECCV 2014*]



Represented as pixels
x_1
x_2
x_N
……
……
……
The most basic classifiers
Use 1st layer as module to build classifiers
Use 2nd layer as module ……

Too many weights in a dense network!



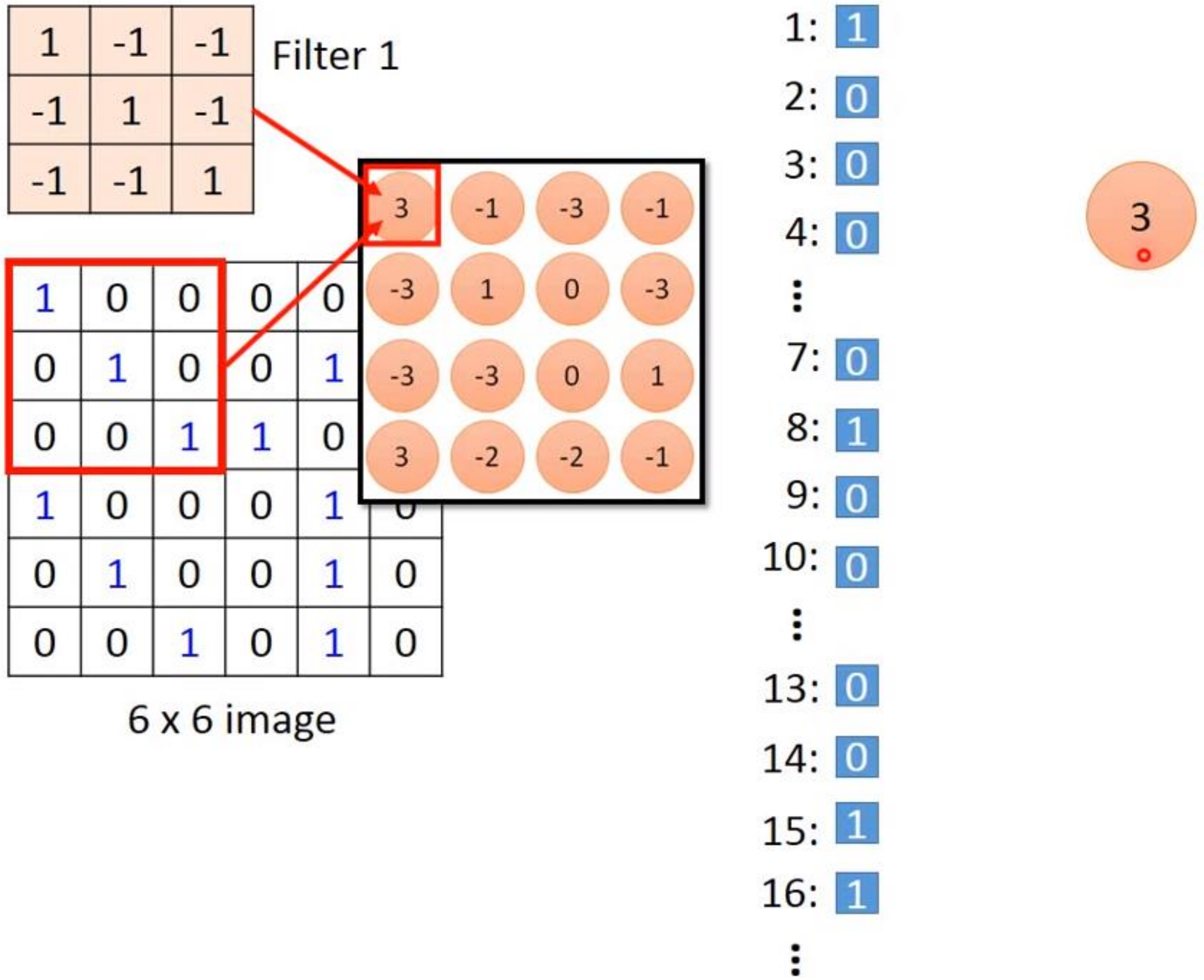
Filter 1

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

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

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

6 x 6 image



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

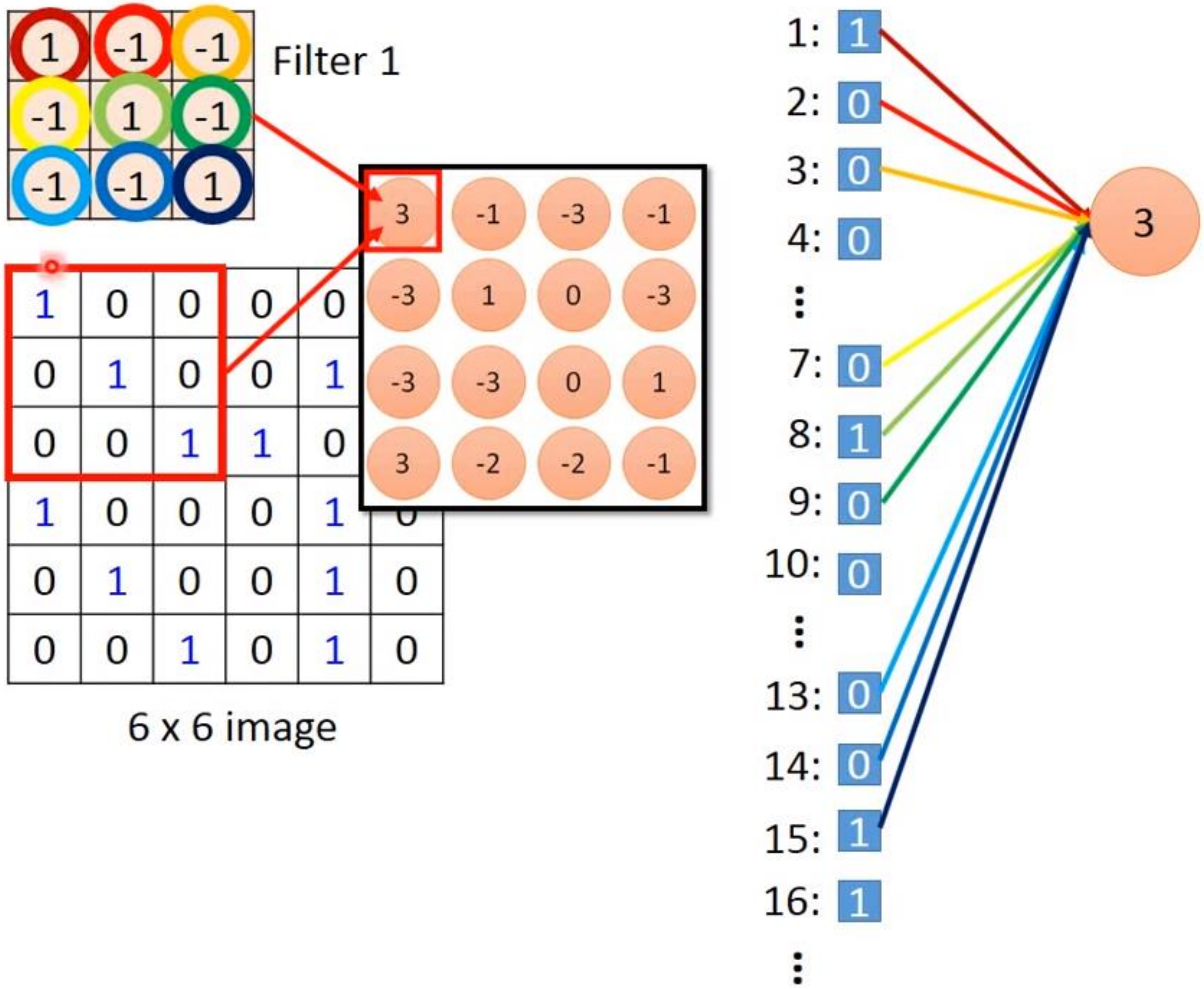
Filter 1

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

6 x 6 image

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

1:	1
2:	0
3:	0
4:	0
⋮	
7:	0
8:	1
9:	0
10:	0
⋮	
13:	0
14:	0
15:	1
16:	1
⋮	



Filter 1

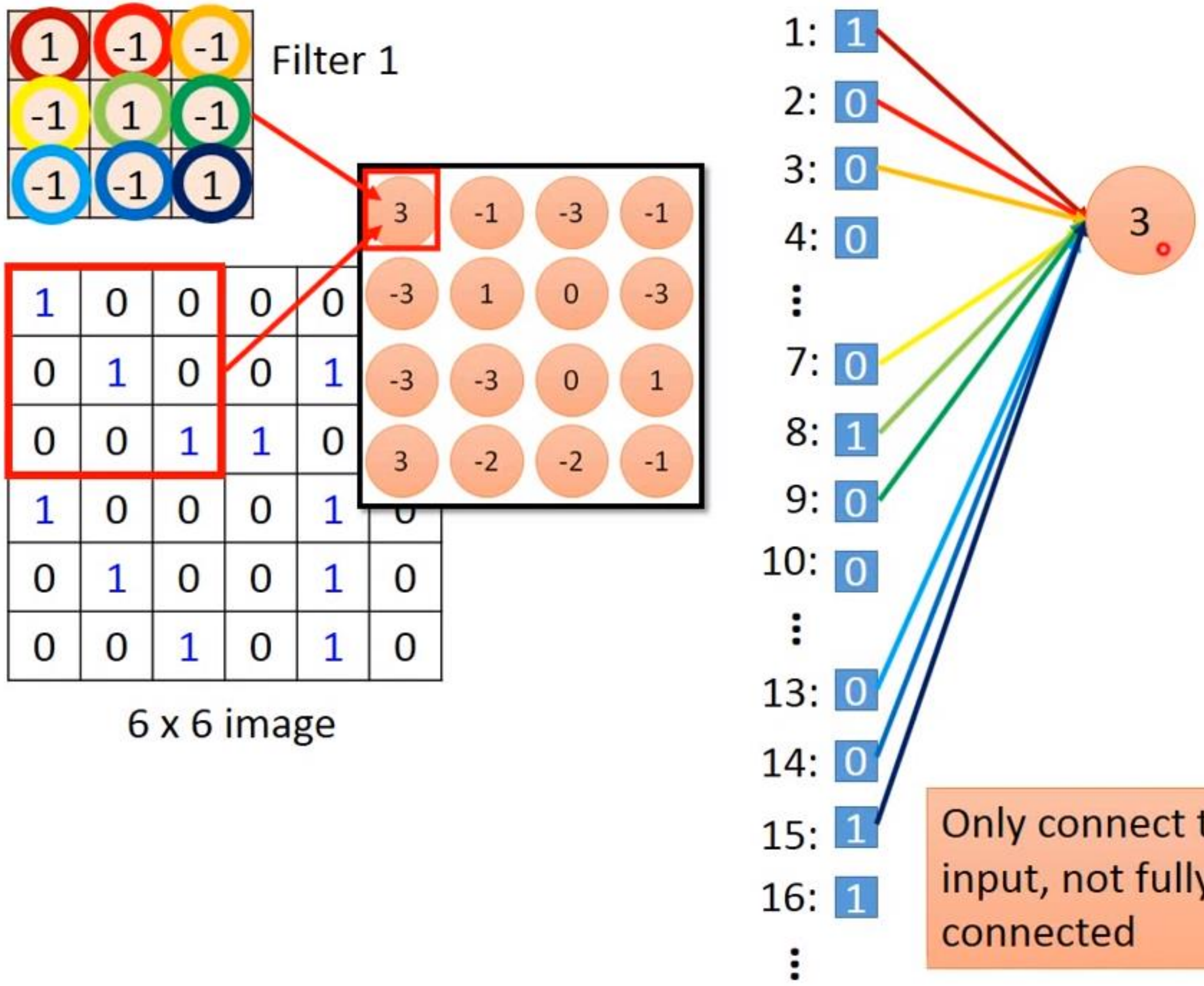
1	-1	-1
-1	1	-1
-1	-1	1

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

1	0	0	0	0	
0	1	0	0	1	
0	0	1	1	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
2: 0
3: 0
4: 0
⋮
7: 0
8: 1
9: 0
10: 0
⋮
13: 0
14: 0
15: 1
16: 1
⋮



Filter 1

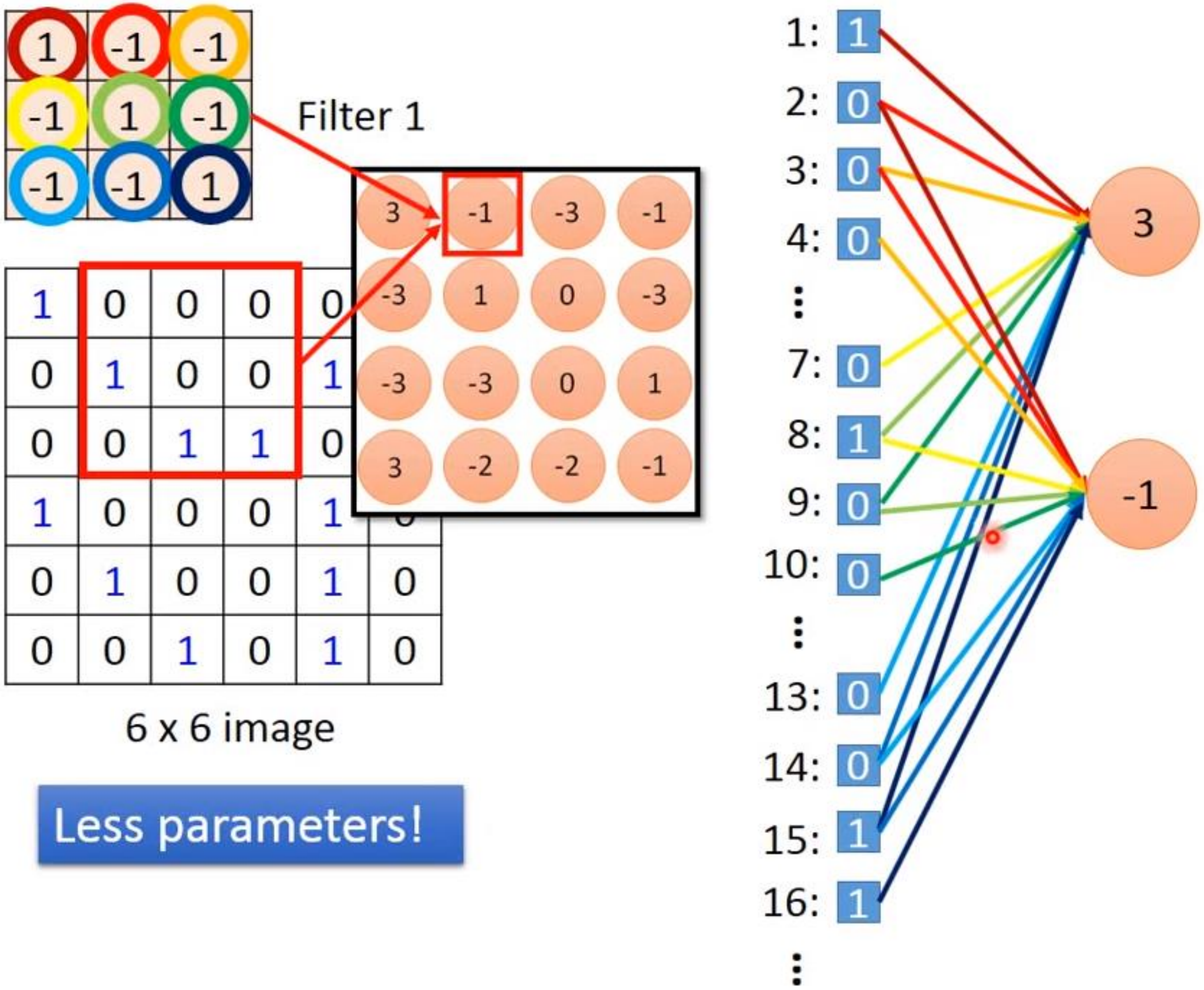
1	-1	-1
-1	1	-1
-1	-1	1

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

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

6 x 6 image

Only connect to 9 input, not fully connected



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

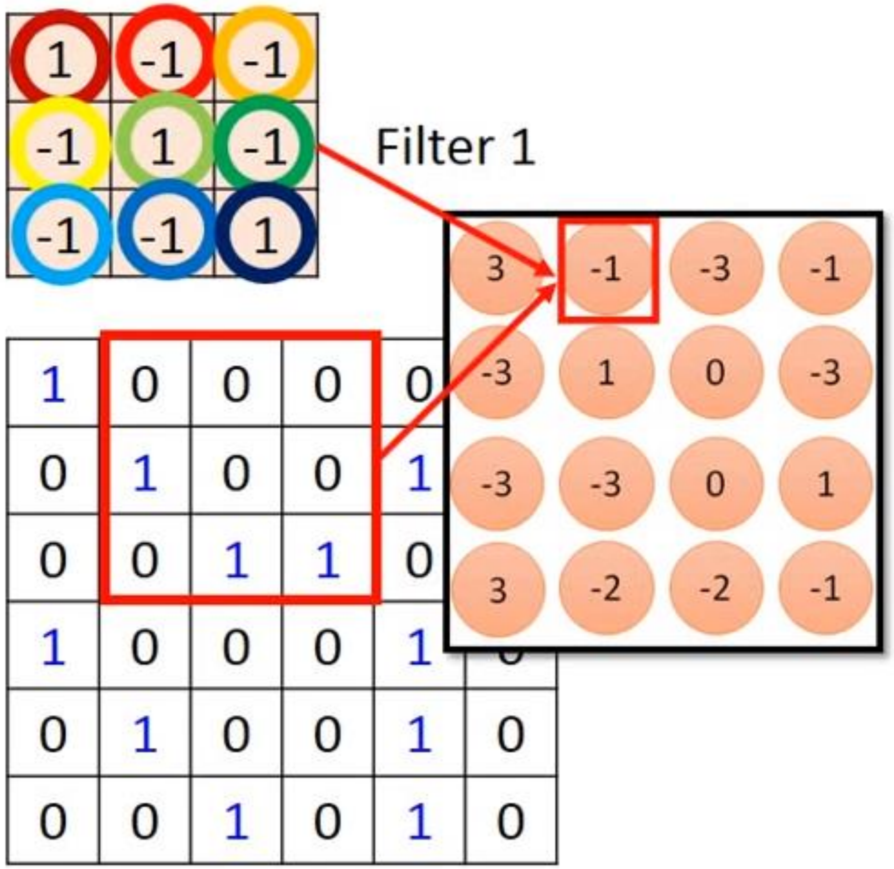
Filter 1

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

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

6 x 6 image

Less parameters!



Filter 1

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

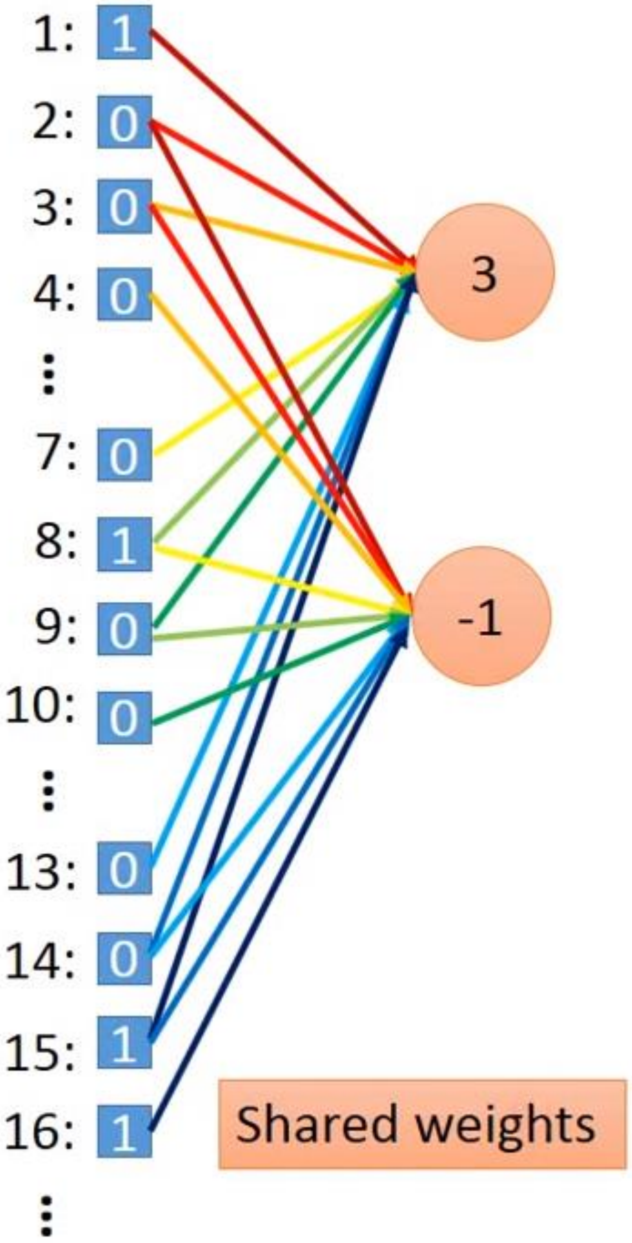
3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1

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

6 x 6 image

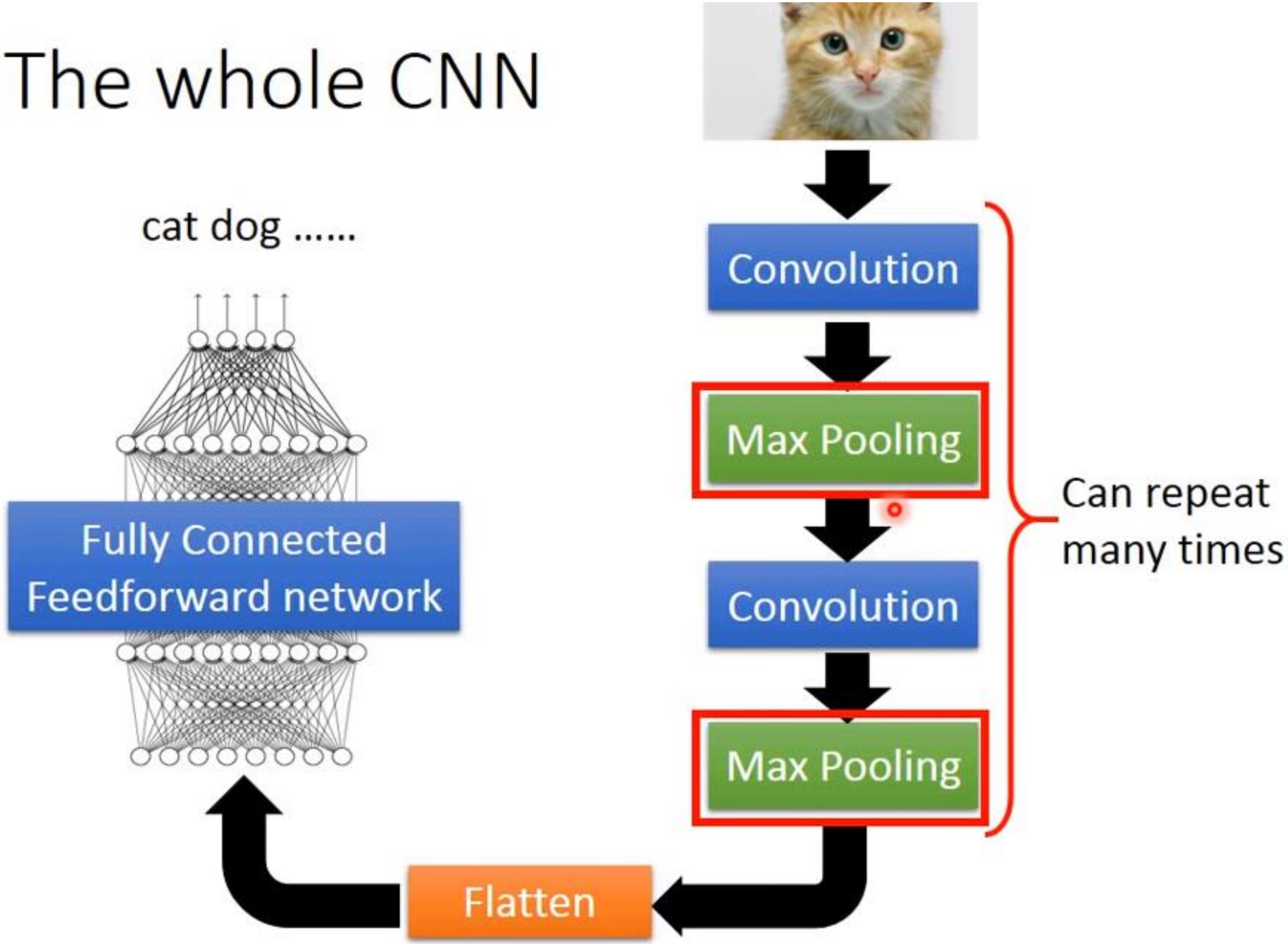
Less parameters!

Even less parameters!



Shared weights

The whole CNN


cat dog ……
Convolution
Max Pooling
Convolution
Max Pooling
Can repeat many times
Flatten
Fully Connected Feedforward network