Convolutional Neural Network

Can the network be simplified by considering the properties of images?

Artificial Neural Networks

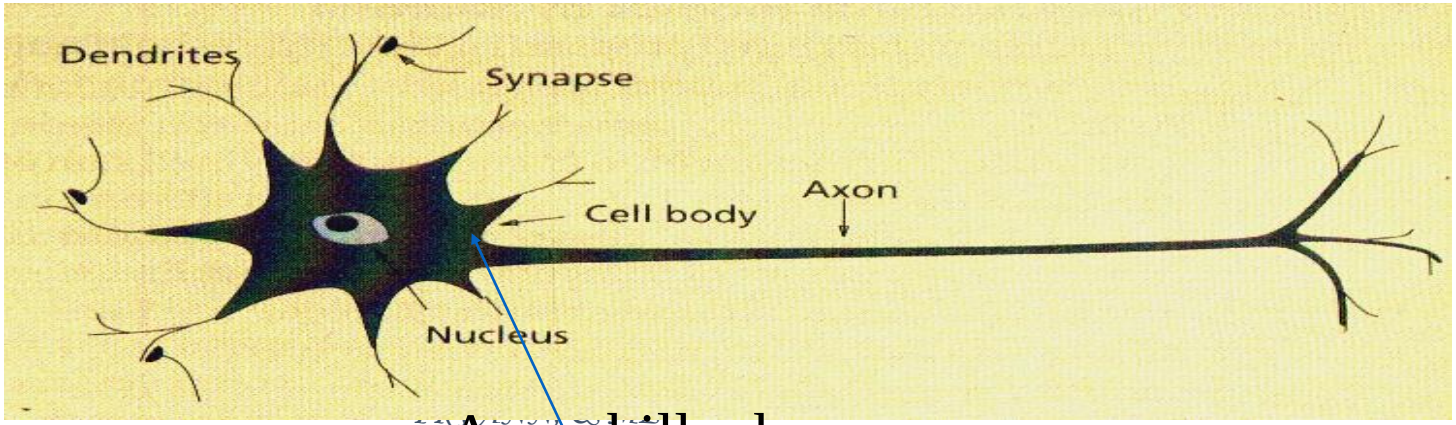
- Connectionist, PDP, etc. models
- A biologically-inspired approach for
  - intelligent computing machines
  - massive parallelism
  - distributed computing
  - learning, generalization, adaptivity
  - Tolerant of fault, uncertainty, imprecise info

Compared to Von Neumann

	Von Neumann computer	Biological neural systems
Processor	complex, high speed, few	simple, low speed, many
Memory	separate from processor,	integrated into processor, content addressable
	non-content addressable	
Computing	centralized, sequential stored programs	distributed, parallel self-learning
Reliability	vulnerable	fault tolerant

Biological Neural Networks

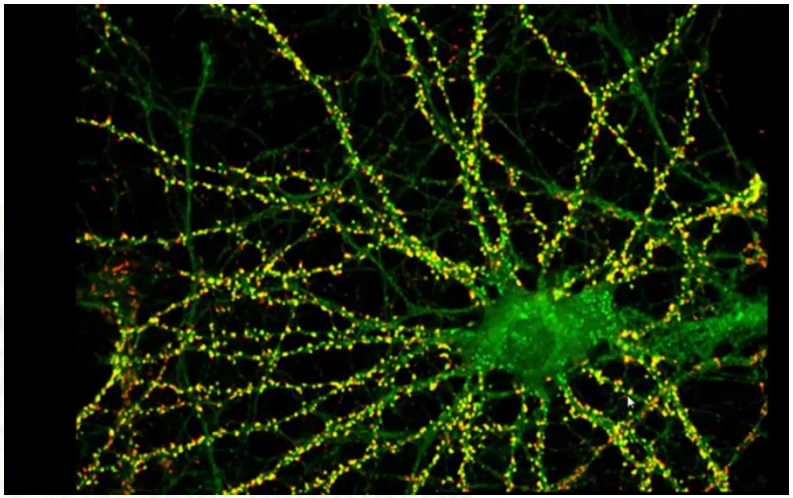
- soma (cell body)
- dendrites (receivers)
- axon (transmitters)
- synapses (connection points, axon-soma, axon-dendrite, axon-axon)
- Chemicals (neurotransmitters)
- 10^{11} neurons
- each makes about $10^3 \sim 10^4$ connections
- with an operating speed of a few milliseconds
- one-hundred-step rule



Signal Generation

- Resting potential
  - Charge difference across neuron membrane approximately –70mV
- Graded potential
  - Stimulus across synapses of post-synaptic neuron
- Action potential
  - If accumulation of graded potential across neuron membrane over a short period of time is higher than ~15mV, action potential is generated and propagated across axon
  - Same form and amplitude regardless of stimulus, signal by frequency rather than amplitude

Signal Generation



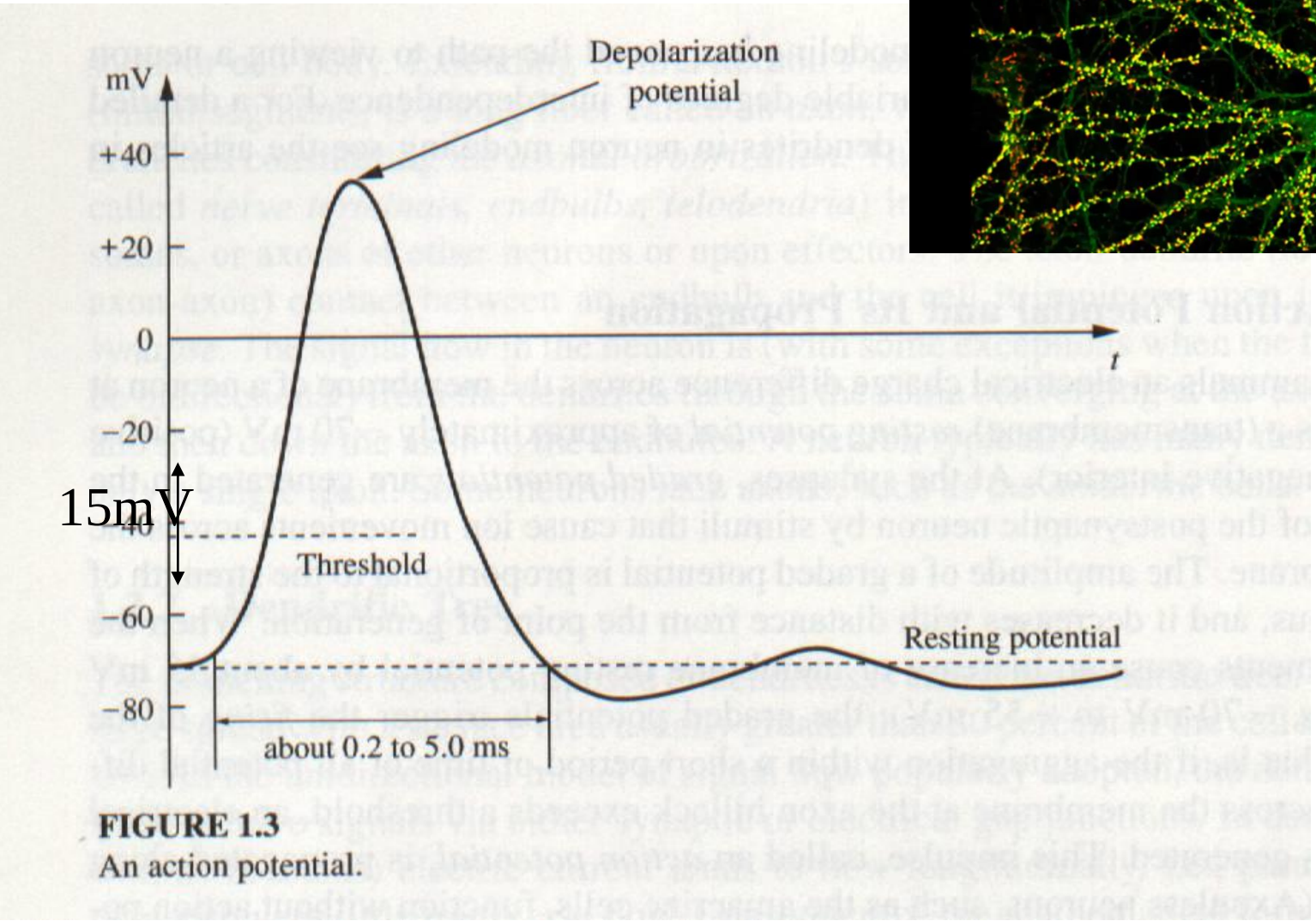
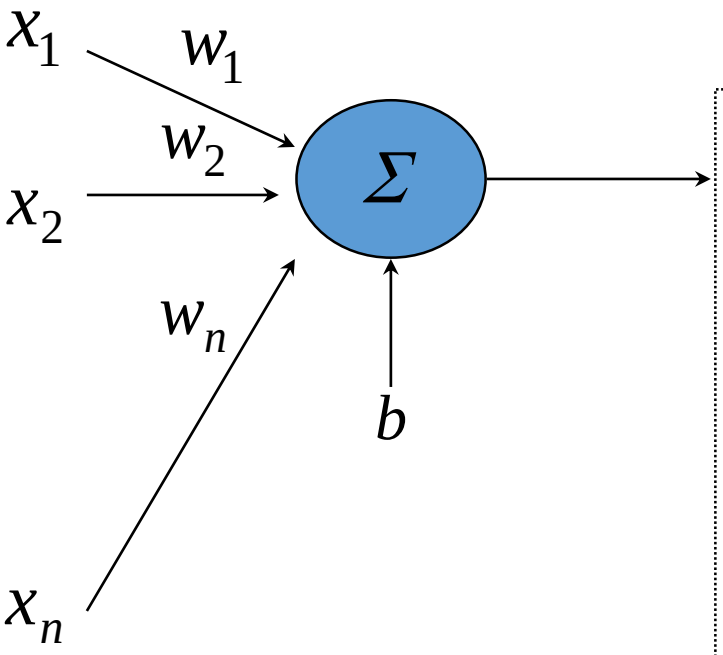
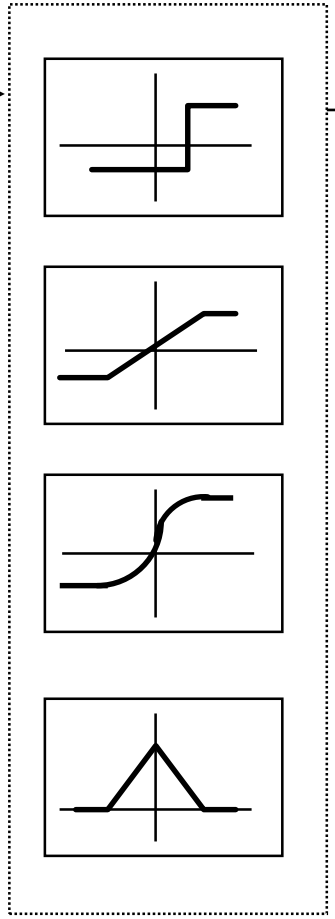


FIGURE 1.3
An action potential.

Computational Neuron Model (McCulloch and Pitts)

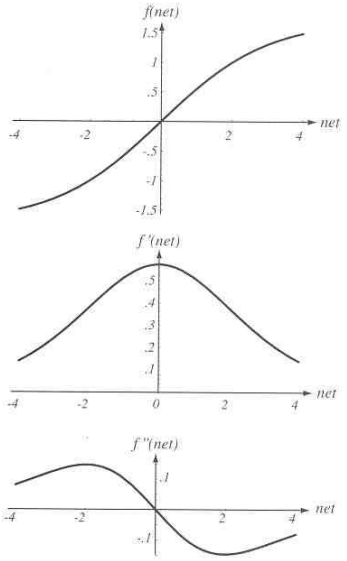




$$O = \theta(\sum_{i=1}^{n} w_i x_i - b)$$

$$\theta(x) = \frac{1}{1+e^{-x}}, 0 \le \theta \le 1$$

$$\theta(x) = \tanh(x), -1 \le \theta \le 1$$



- time dependency?
- frequency response?

What does a neuron do mathematically?

- W and b determine a "basis" function
- Unlike Fourier bases, these bases are learned to fit the data
- Pattern finder or classifier

- W and b determine a partitioning hyperplane
- They separate high-dimensional feature spaces into regions
- Pattern classifier
- Again, the partitioning hyperplanes are learned

Why CNN for Image

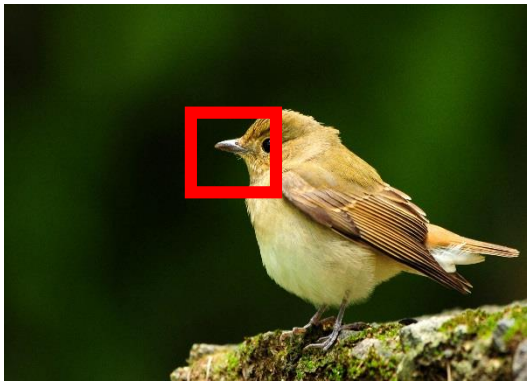
- Some patterns are much smaller than the whole image

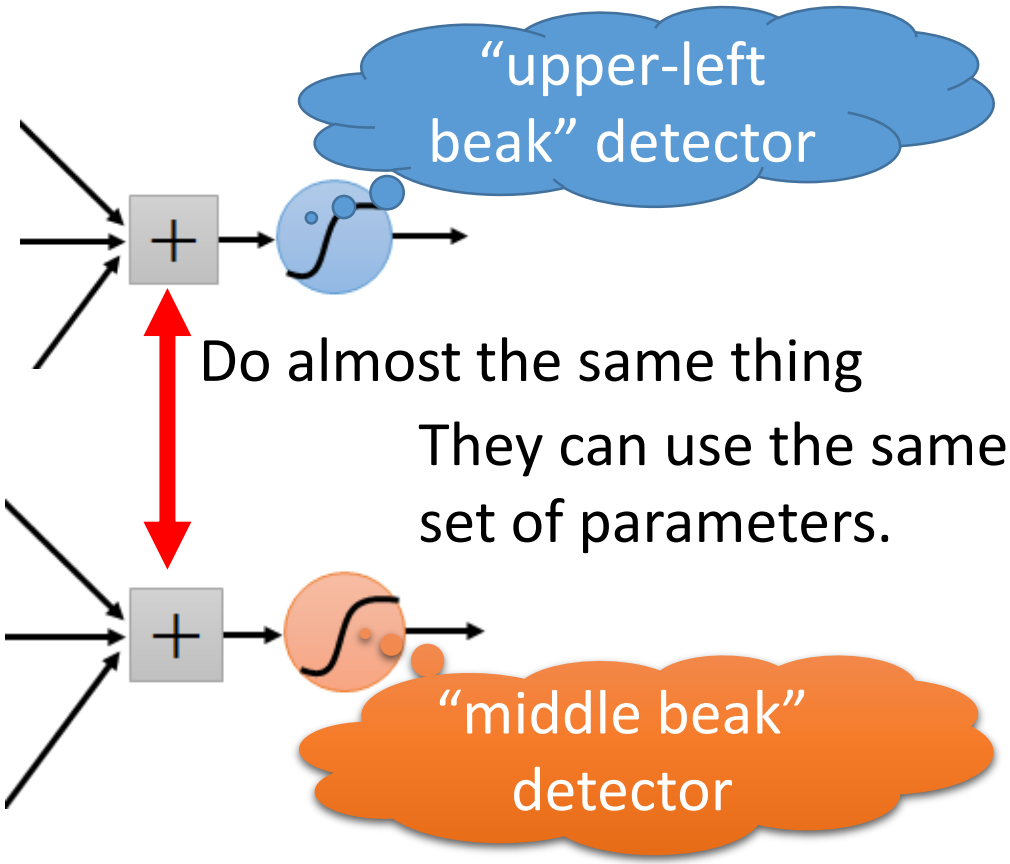
A neuron does not have to see the whole image to discover the pattern.

Connecting to small region with less parameters

Why CNN for Image

- The same patterns appear in different regions.





Why CNN for Image

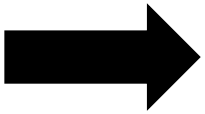
- Subsampling the pixels will not change the object

bird

subsampling

bird

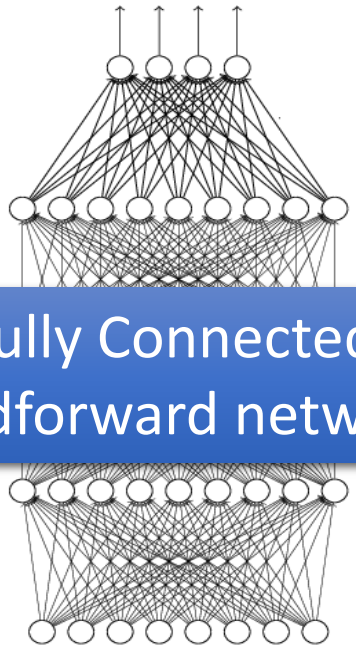
We can subsample the pixels to make image smaller



Less parameters for the network to process the image

The whole CNN

cat dog ……



Fully Connected Feedforward network

Convolution

Max Pooling

Convolution

Max Pooling

Activation

Activation

Can repeat many times

Flatten

The whole CNN

Property 1

- Some patterns are much smaller than the whole image

Property 2

- The same patterns appear in different regions.

Property 3

- Subsampling the pixels will not change the object

Convolution

Max Pooling

Convolution

Max Pooling

Can repeat many times

Flatten

The whole CNN

Convolution

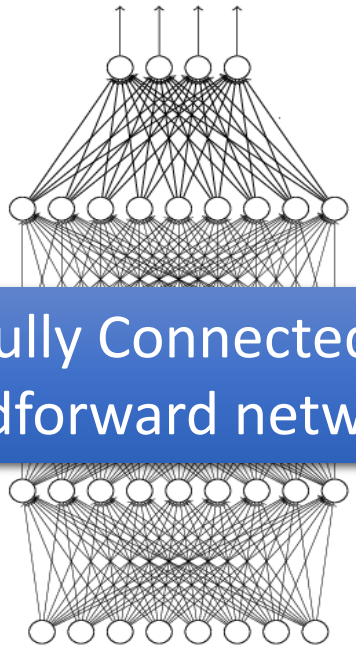
Max Pooling

Convolution

Max Pooling

Can repeat many times

Flatten



Fully Connected Feedforward network

cat dog ……

CNN – Convolution

Those are the network parameters to be learned.

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

Matrix

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

Filter 2

Matrix

⋮

Property 1

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

CNN – Convolution

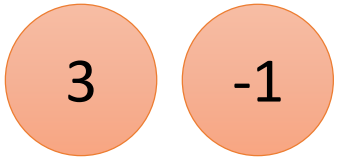
1	-1	-1
-1	1	-1
-1	-1	1

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

6 x 6 image



CNN – Convolution

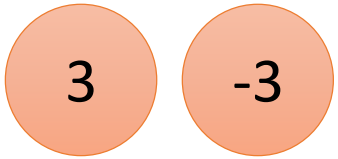
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

If stride=2

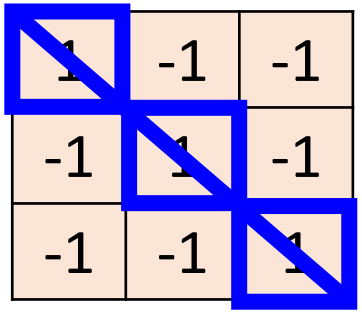
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image



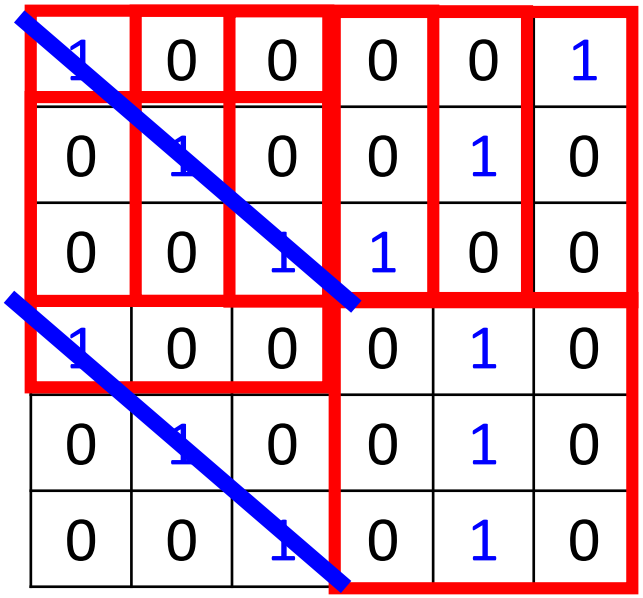
We set stride=1 below

CNN – Convolution

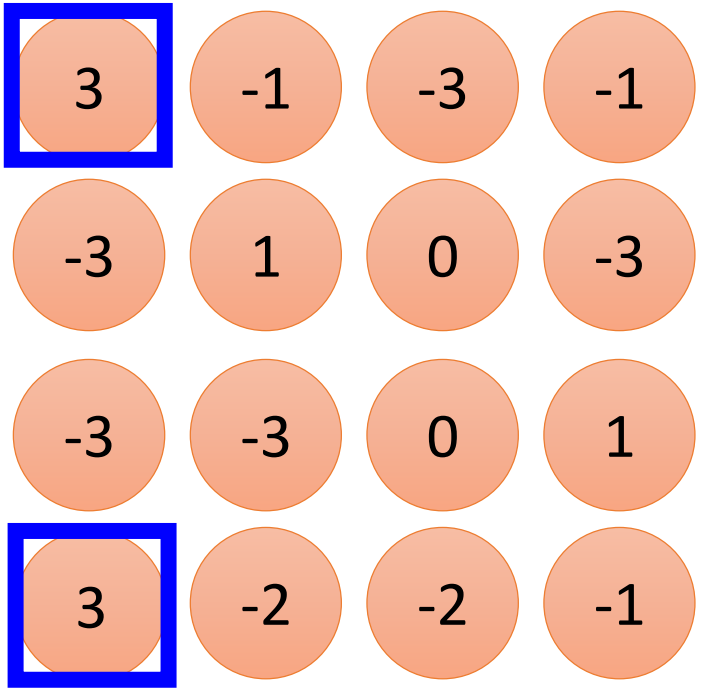


Filter 1

stride=1



6 x 6 image



Property 2

CNN – Convolution

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

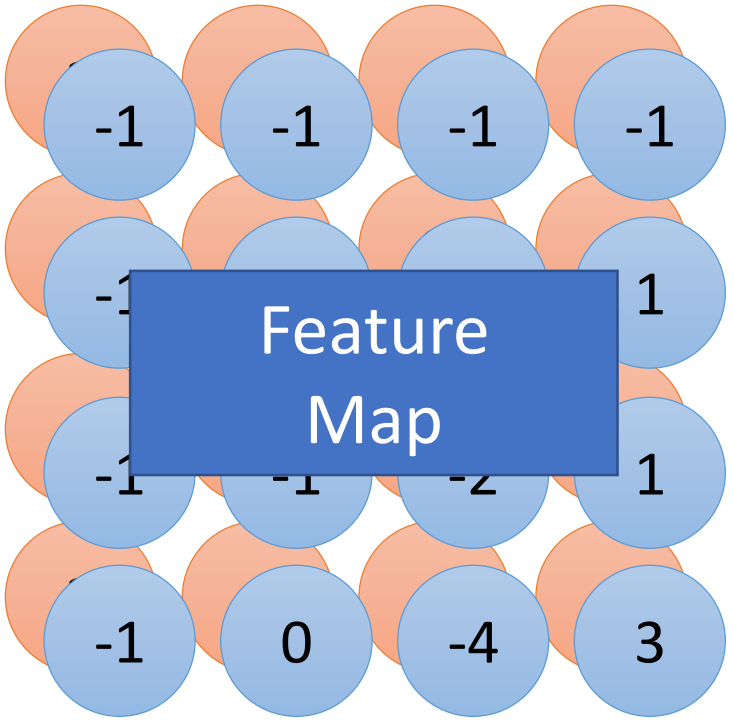
Filter 2

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

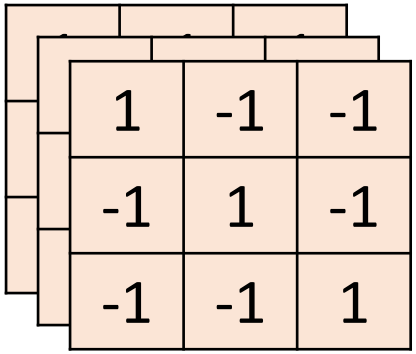
6 x 6 image

Do the same process for every filter

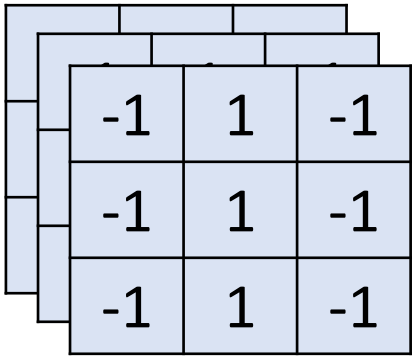


4 x 4 image

CNN – Colorful image

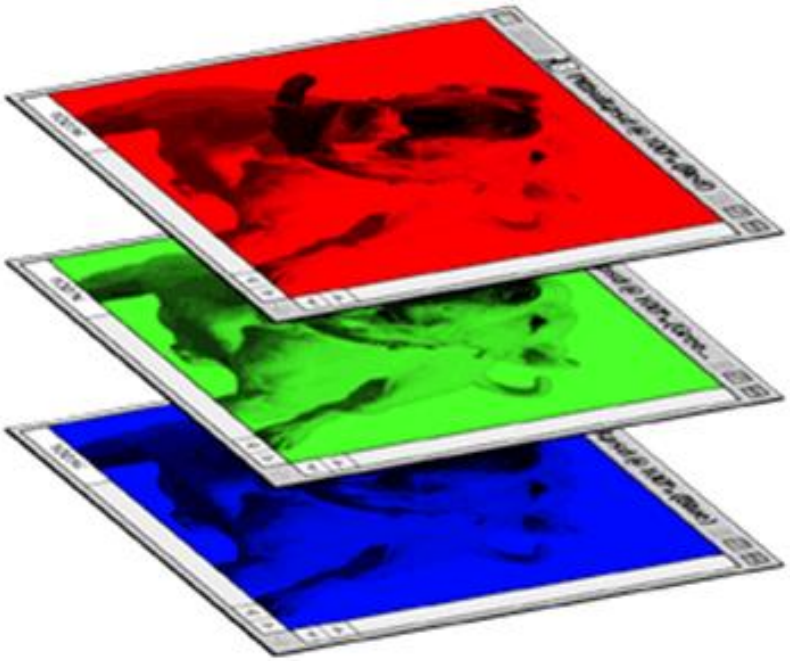


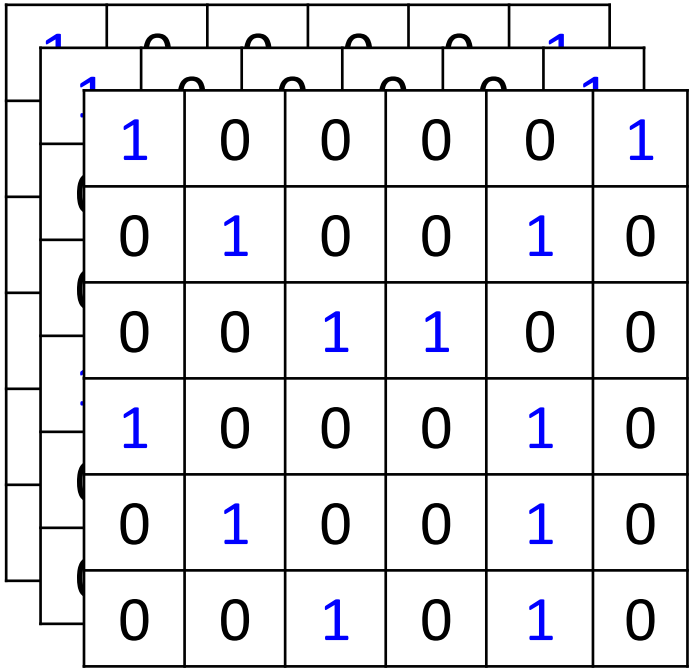
Filter 1



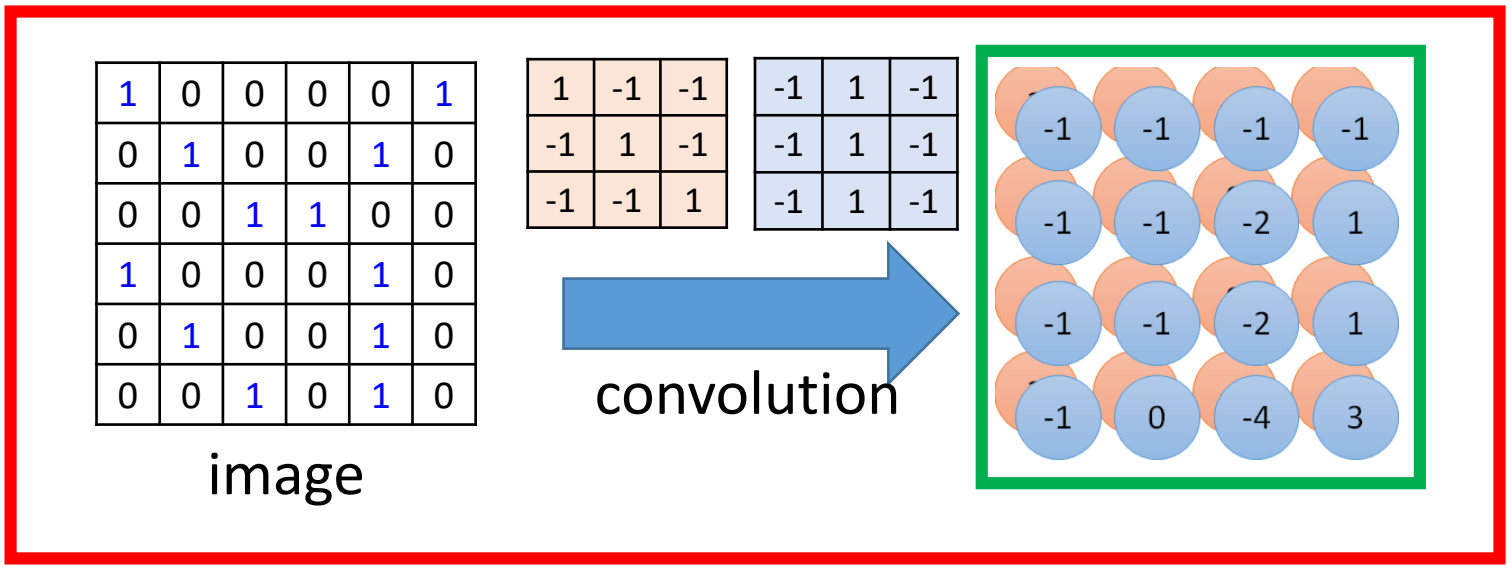
Filter 2

Colorful image



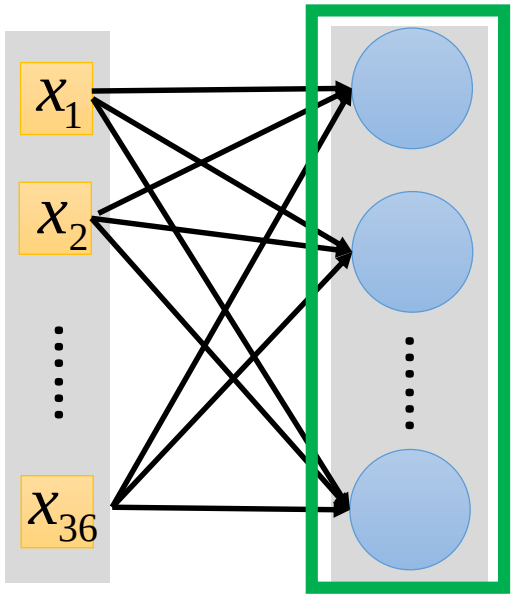


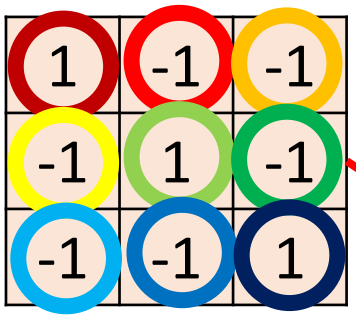
Convolution v.s. Fully Connected



Fully-connected

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





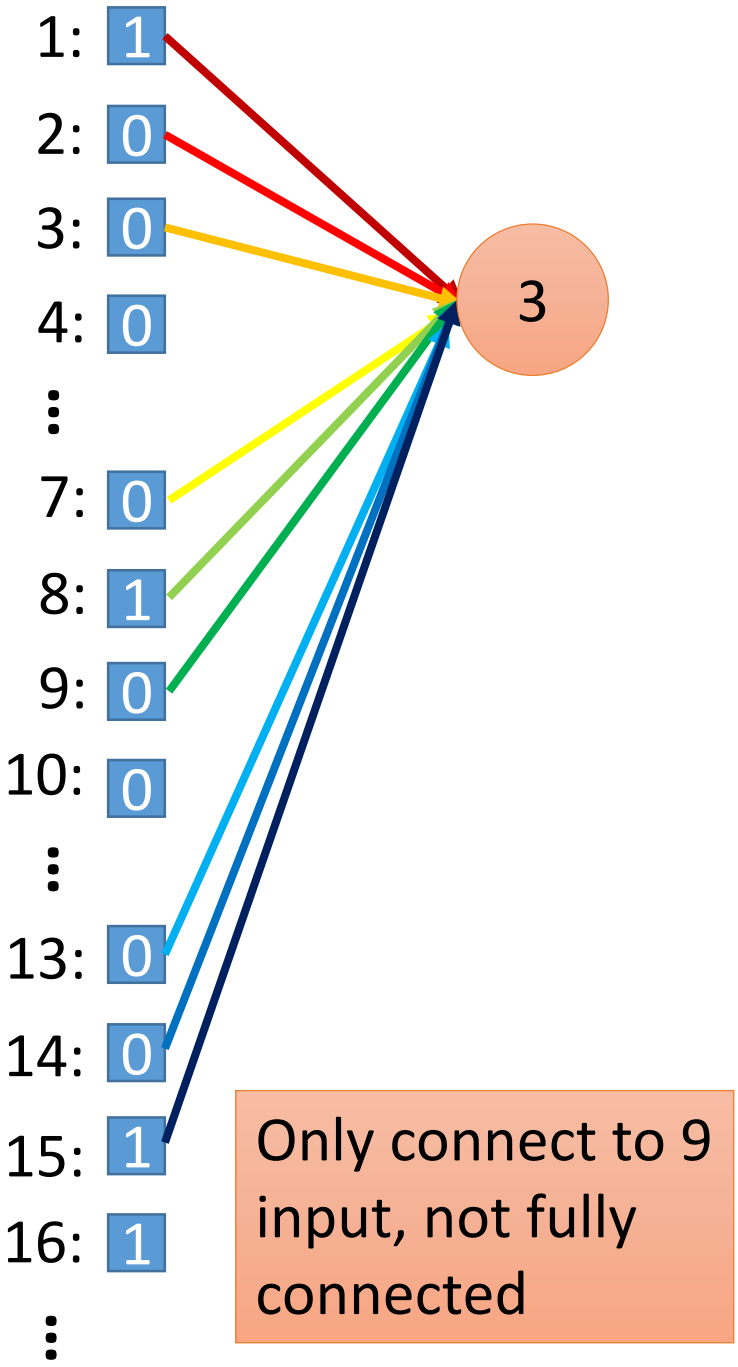
Filter 1

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

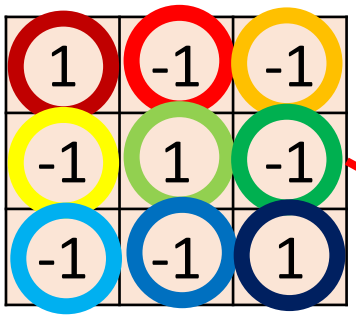
6 x 6 image

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

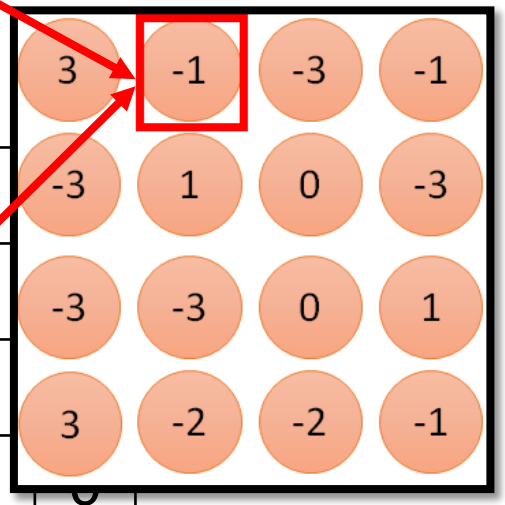
Less parameters!



Only connect to 9 input, not fully connected



Filter 1

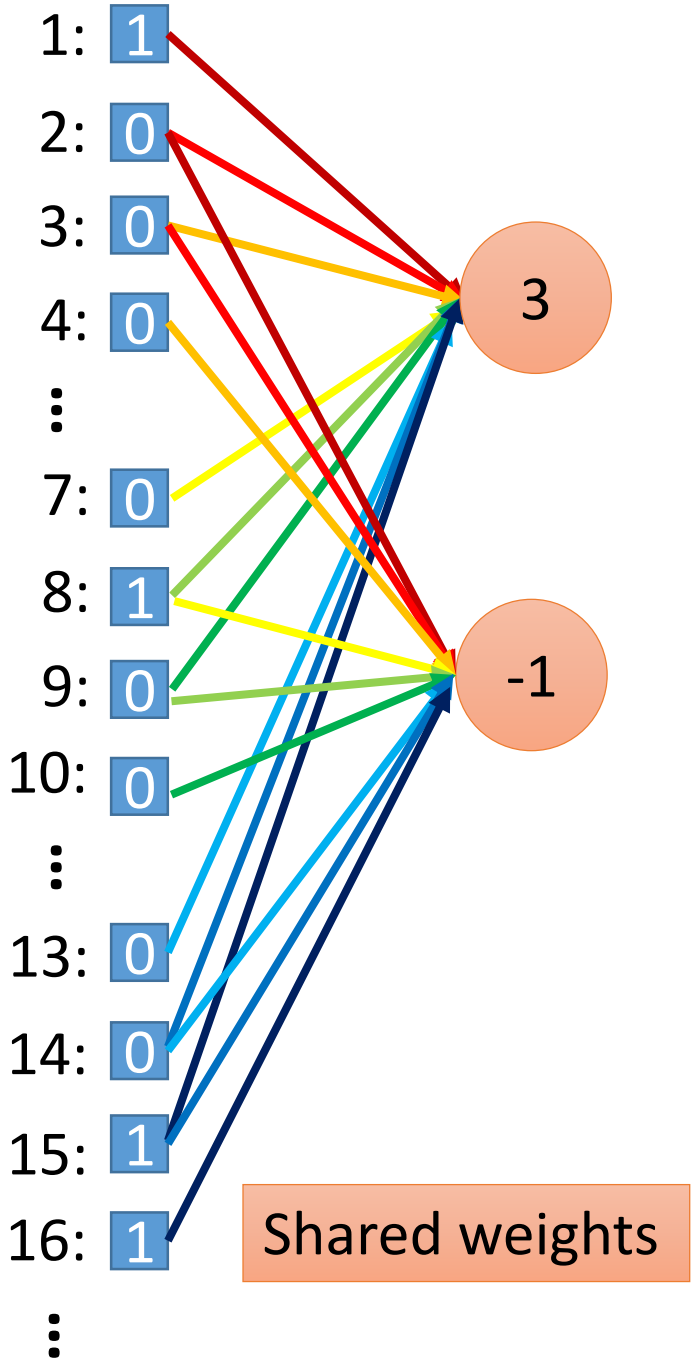


1	0	0	0	0	
0	1	0	0	1	
0	0	1	1	0	
1	0	0	0	1	
0	1	0	0	1	0
0	0	1	0	1	0

6 x 6 image

Less parameters!

Even less parameters!



Shared weights

The whole CNN

Convolution

Max Pooling

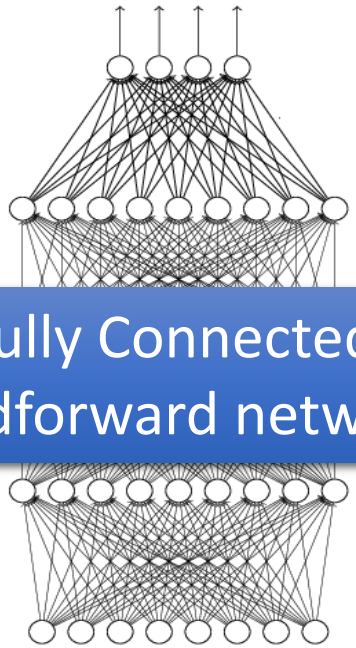
Convolution

Max Pooling

Can repeat many times

Flatten

cat dog ……



Fully Connected Feedforward network

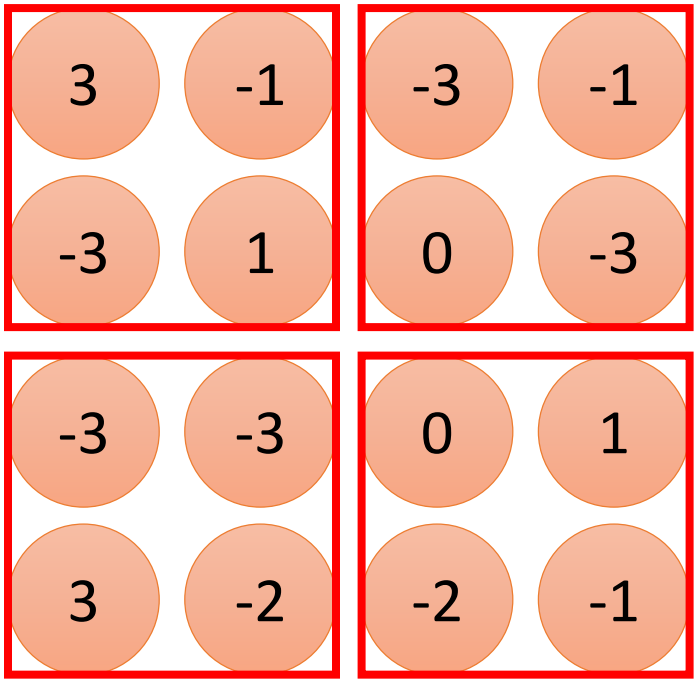
CNN – Max Pooling

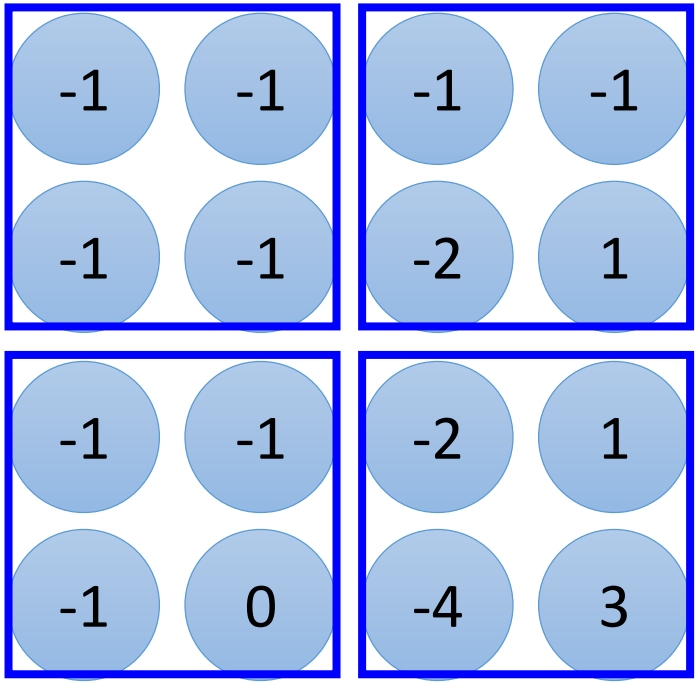
1	-1	-1
-1	1	-1
-1	-1	1

Filter 1

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

Filter 2

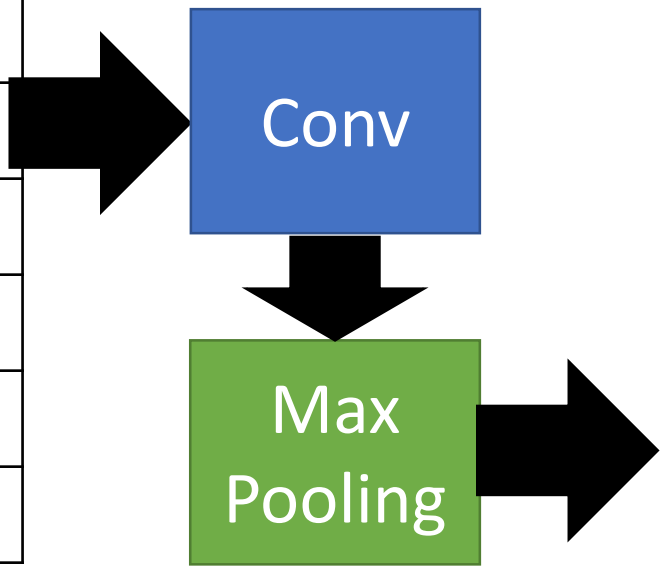




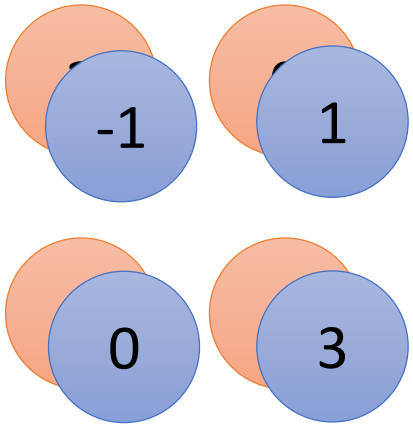
CNN – Max Pooling

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

6 x 6 image



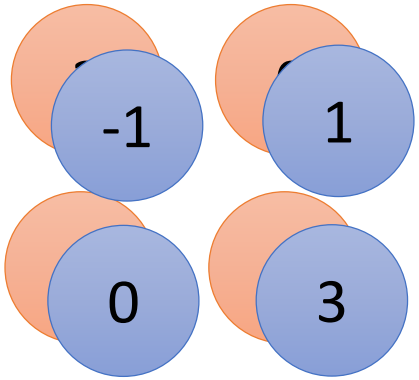
New image but smaller



2 x 2 image

Each filter is a channel

The whole CNN



A new image

Convolution

Activation

Max Pooling

Can repeat many times

Convolution

Activation

Max Pooling

Smaller than the original image

The number of the channel is the number of filters

The whole CNN

cat dog ……

Convolution

Max Pooling

A new image

Convolution

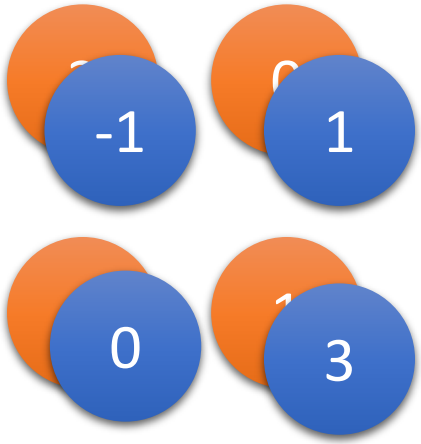
Max Pooling

A new image

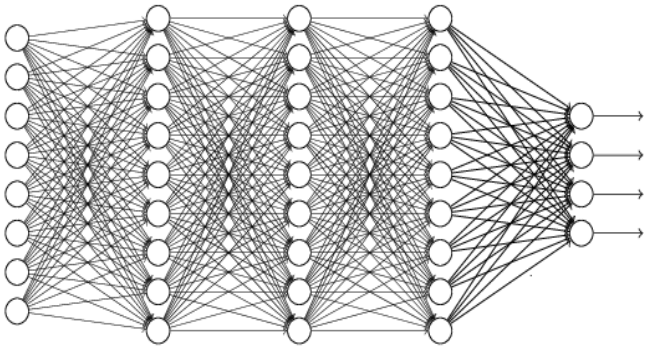
Flatten

Fully Connected Feedforward network

Flatten



Flatten



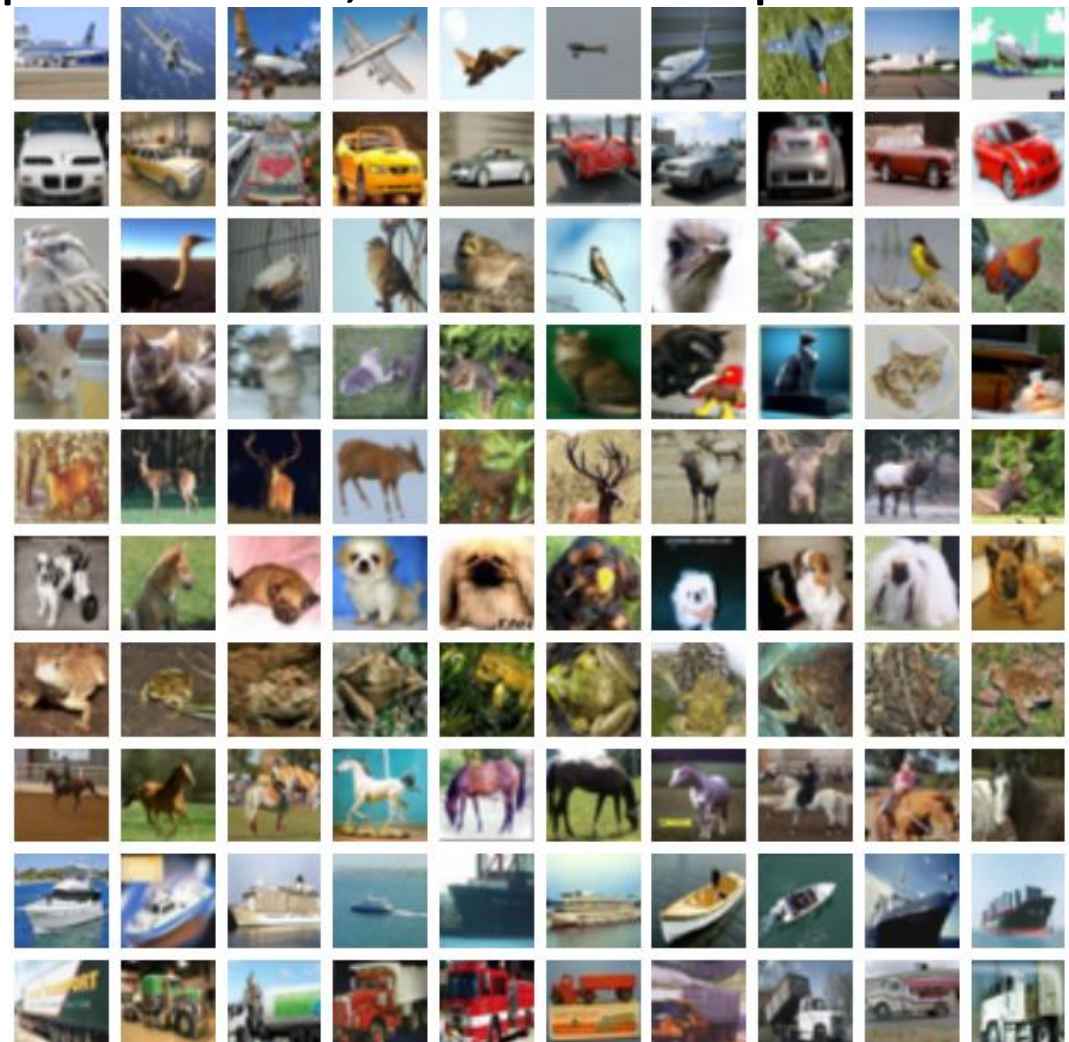
Fully Connected Feedforward network

CNN Demo - Mnist

- Hand-written digits
- 60,000 training samples and 10,000 test samples
- 28x28 binary images

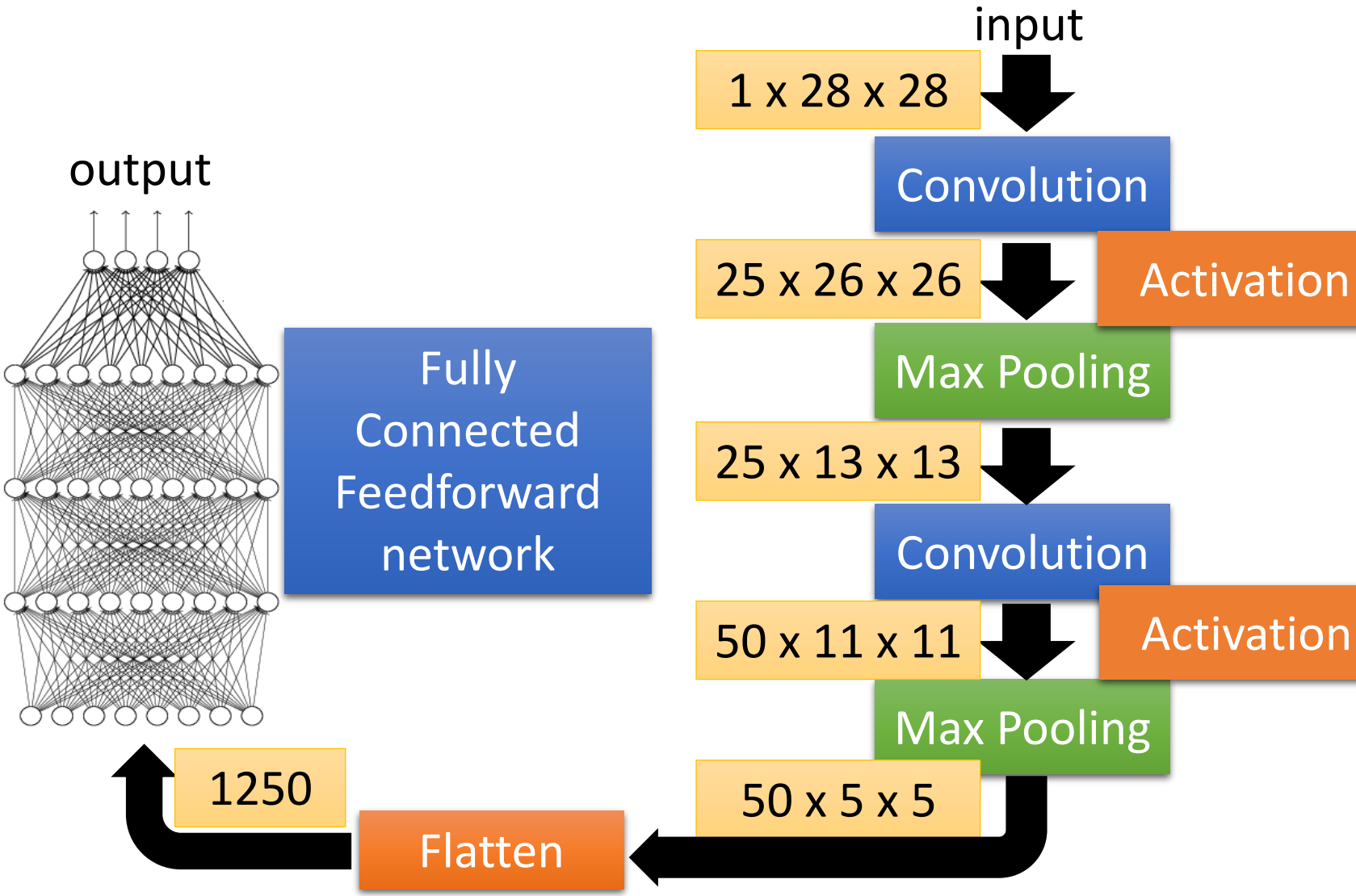
CNN Demo – CiFar10

- 10 classes (airplane, auto, bird, cat, dog, etc.)
- 50,000 training samples and 10,000 test samples
- 32x32 color images



CNN for Mnist

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



Loss Function

- Ground truth – 1-hot vector of dimension nx1
  - N: number of categories
  - 1: true category (dog, cat, etc.)
- CNN output
  - Softmax post processing to make a probability distribution
- Loss:
  - Sum over mini-batch
  - 2-norm error (doesn’t work)
  - Cross entropy

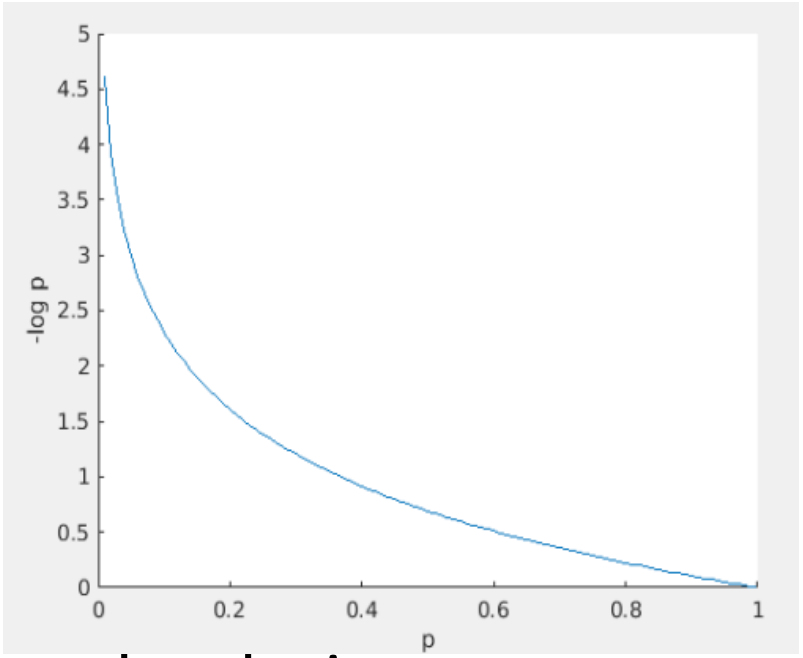
Softmax Function

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \quad \text{for } j = 1, \ldots, K \text{ and } \mathbf{z} = (z_1, \ldots, z_K) \in \mathbb{R}^K$$

- Exponent CNN output
- Normalized by sum of exponents
- Output is a probability distribution
- Accentuate positives and suppress negatives

Entropy (Information Theory)

$$S = -\sum_i P_i \log P_i.$$

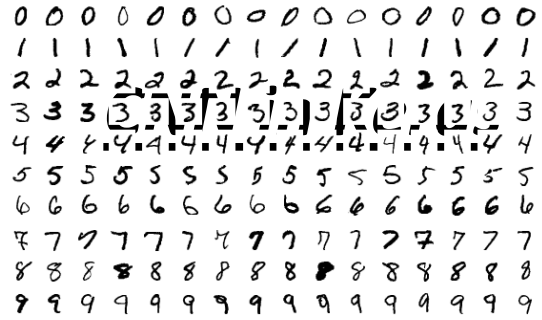


- Number of bits required to encode a lexicon
- Amount of info (surprise)
- Large Pi -> small –log(Pi) -> short codeword
- Small pi -> large –log(pi) -> large codeword
- Compared to ASCII (8-bit per char)
- Entropy is largest when all pi are the same (most uncertain)

Cross Entropy

$$H(p, q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x)$$

- Use a codebook designed for one distribution for another
- Not symmetric, hence, not a distance function
- q(x) > p(x) => x has a short codeword => with small p(x)
- q(x) > p(x) => x has a long codeword => with large p(x)
- Smallest when p==q



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

input

1 x 28 x 28

```
model2.add( Convolution2D( 25,3,3,
                input_shape=(28,28,1)) )
```

Convolution

How many parameters for each filter?

9

25 x 26 x 26

```
model2.add(MaxPooling2D((2,2)))
```

Max Pooling

25 x 13 x 13

```
model2.add(Convolution2D(50,3,3))
```

Convolution

How many parameters for each filter?

225

50 x 11 x 11

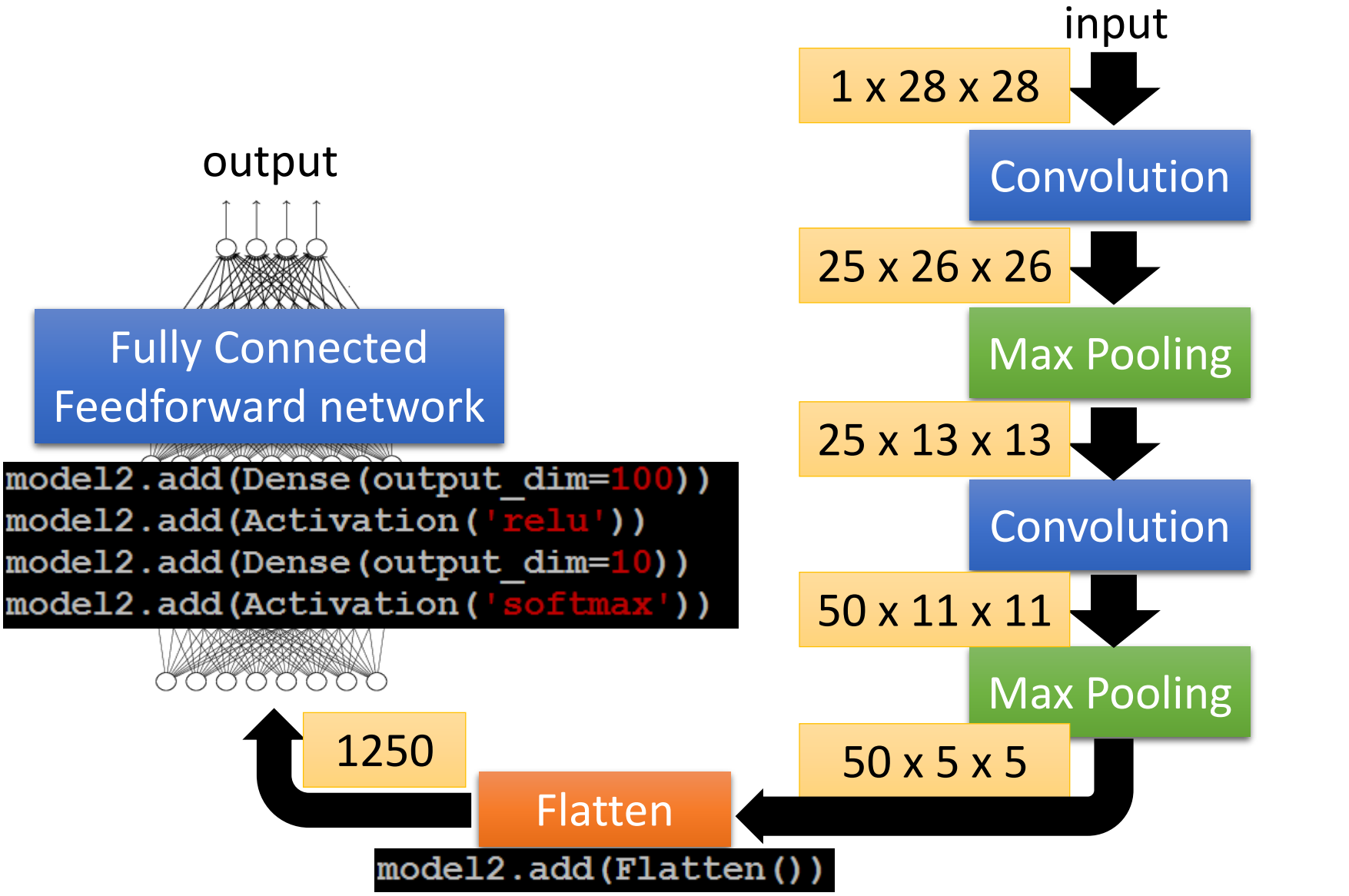
```
model2.add(MaxPooling2D((2,2)))
```

Max Pooling

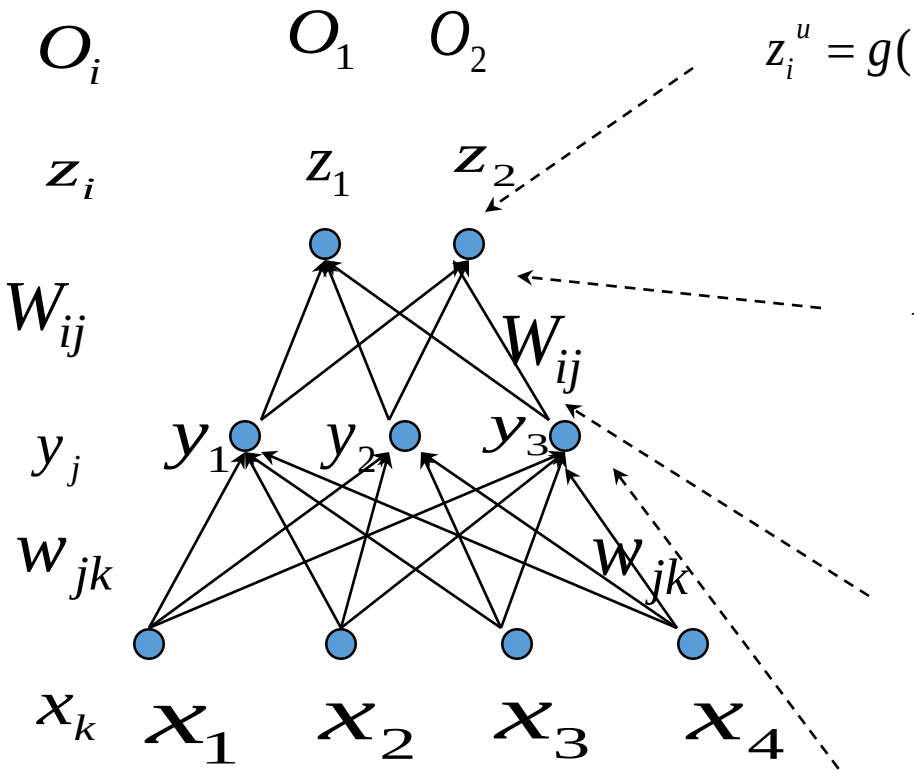
50 x 5 x 5

CNN in Keras

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



Backpropagation Learning rule



$$z_i^u = g(NET_i^u) = g(\sum_j W_{ij} y_j^u) = g(\sum_j W_{ij} g(\sum_k w_{jk} x_k^u))$$

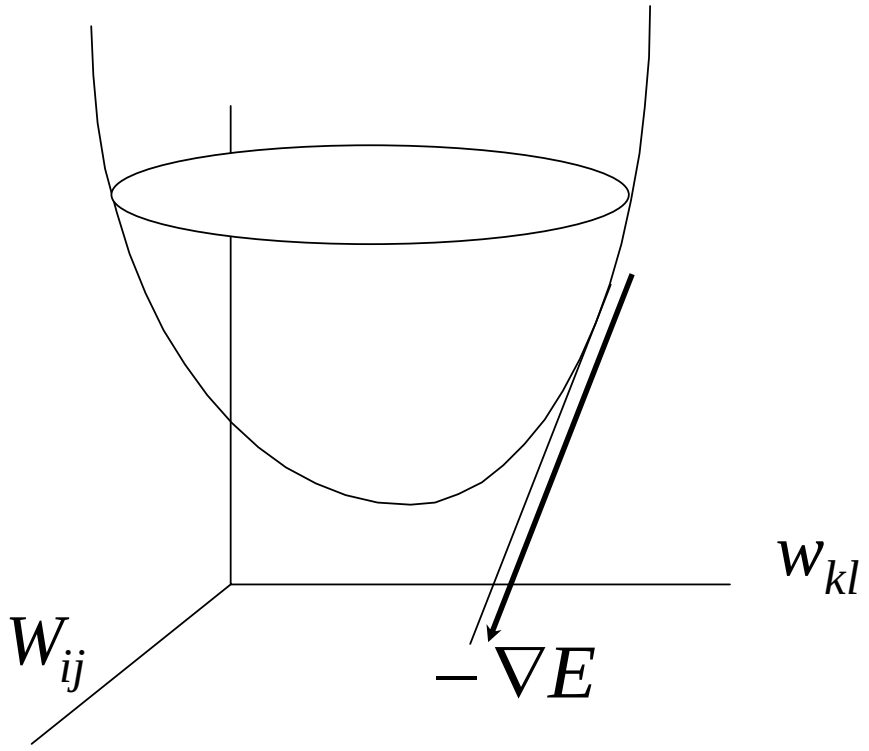
$$NET_i^u = \sum_j W_{ij} y_j^u = \sum_j W_{ij} g(\sum_k w_{jk} x_k^u)$$

$$y_j^u = g(net_j^u) = g(\sum_k w_{jk} x_k^u)$$

$$net_j^u = \sum_k w_{jk} x_k^u$$

Cost function

$$E(\mathbf{W},\mathbf{w}) = \frac{1}{2}\sum_{u,i}(O_i^u - z_i^u)^2 = \frac{1}{2}\sum_{u,i}(O_i^u - g(\sum_j W_{ij}\, g(\sum_k w_{jk} x_k^u))^2$$



What does machine learn?

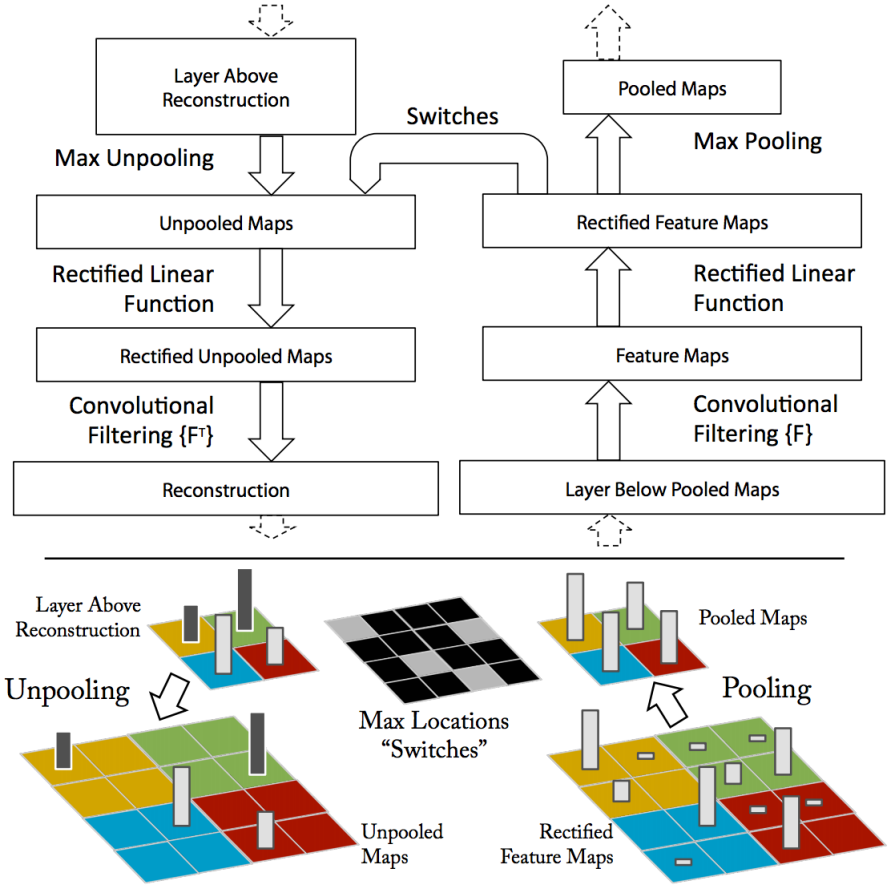
http://newsneakernews.wpengine.netdna-cdn.com/wp-content/uploads/2016/11/rihanna-puma-creeper-velvet-release-date-02.jpg

Visualizing Convolutional Networks

- Deconvolutional Network

  Map activations back to the input pixel space, show what input pattern originally caused a given activation.

- Steps
  - Compute activations at a specific layer.
  - Keep one activation and set all others to zero.
  - Unpooling and deconvolution
  - Construct input image



[1] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European Conference on Computer Vision*. Springer International Publishing, 2014.

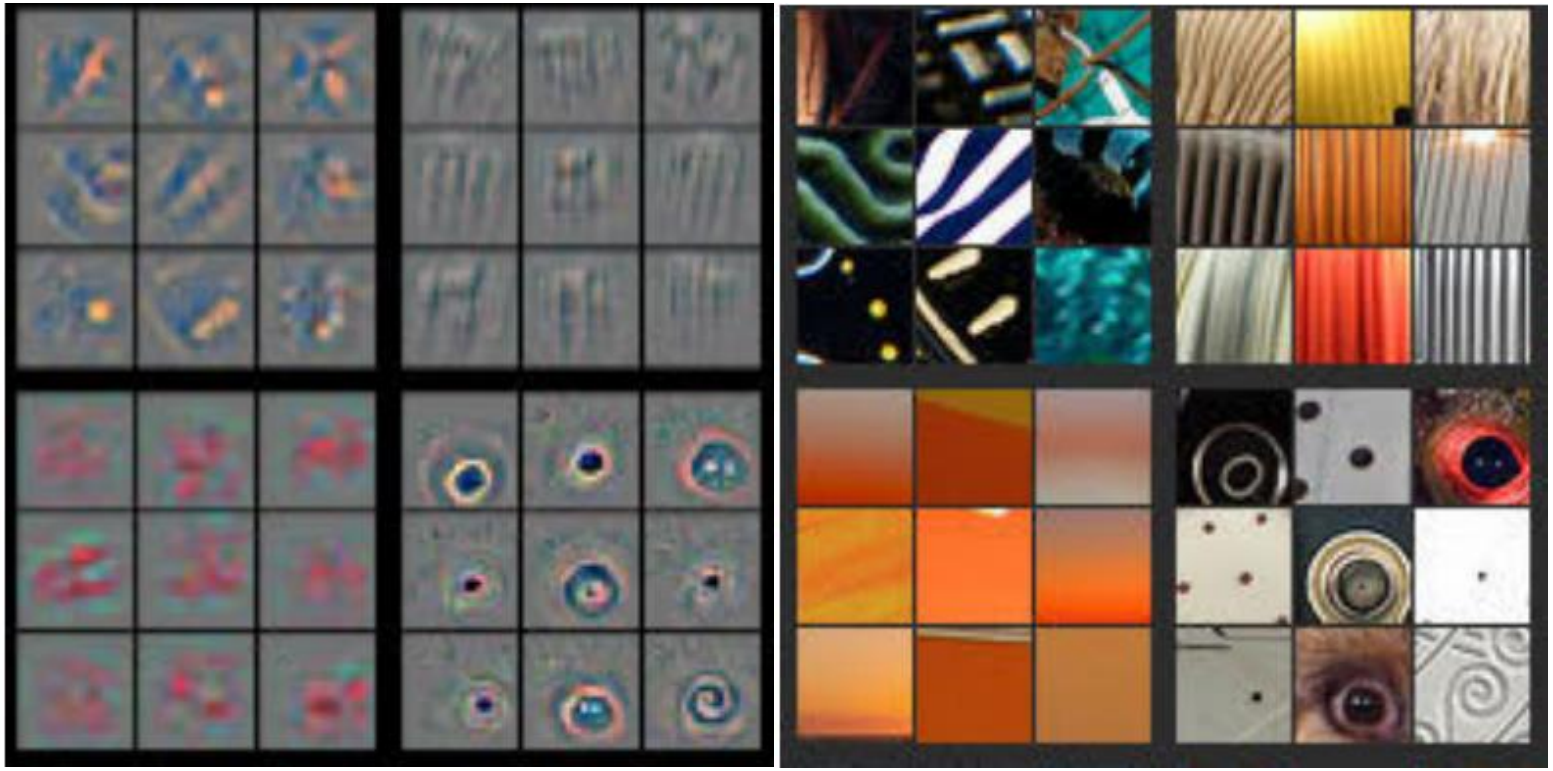
Visualizing Convolutional Networks

- Layer1

[1] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European Conference on Computer Vision*. Springer International Publishing, 2014.

Visualizing Convolutional Networks

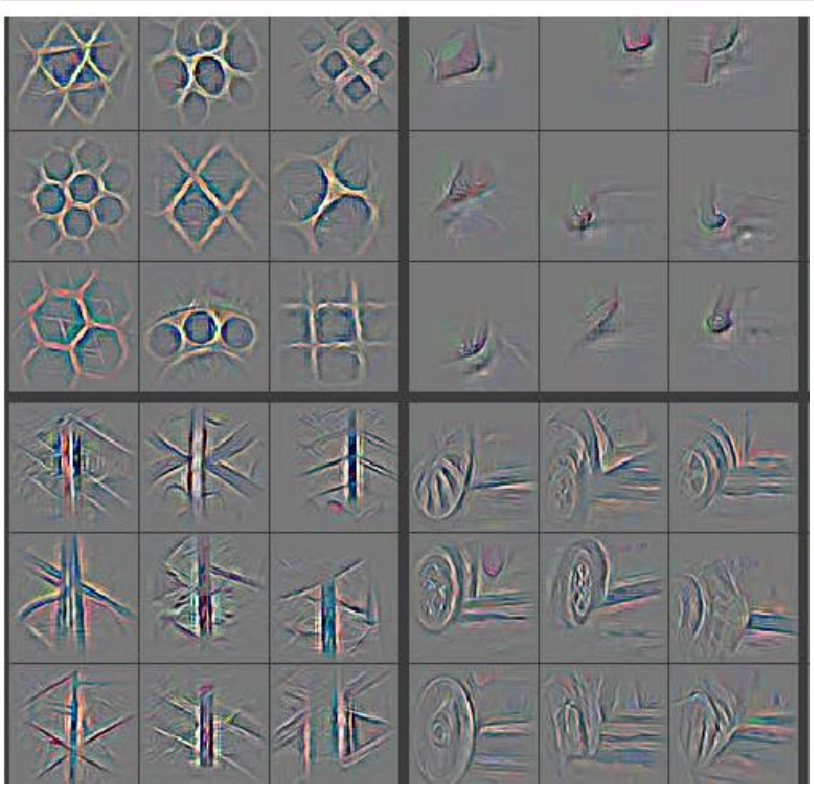
- Layer2

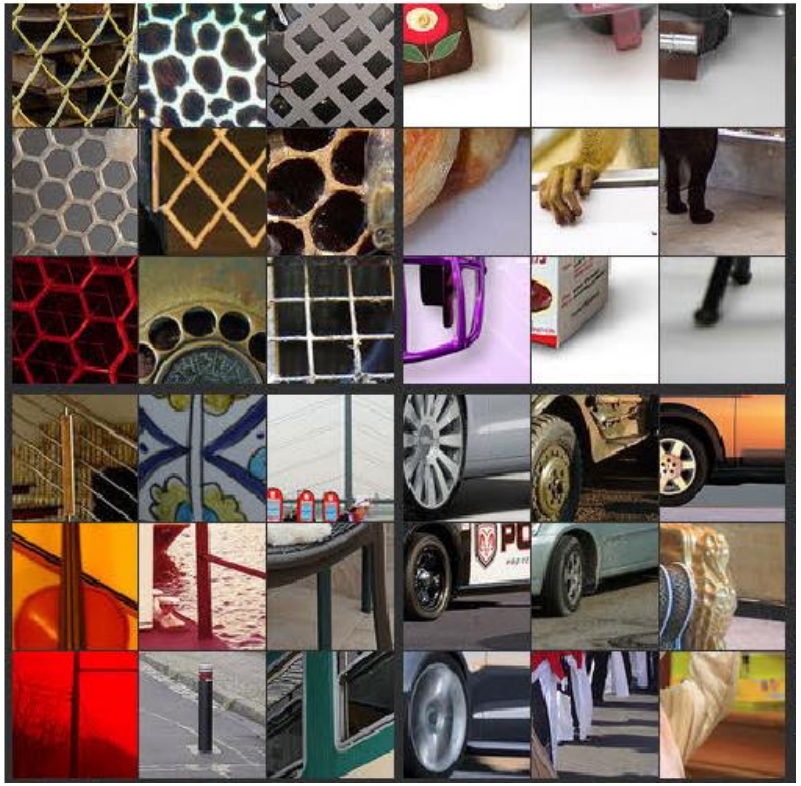


[1] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European Conference on Computer Vision*. Springer International Publishing, 2014.

Visualizing Convolutional Networks

- Layer3





[1] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European Conference on Computer Vision*. Springer International Publishing, 2014.

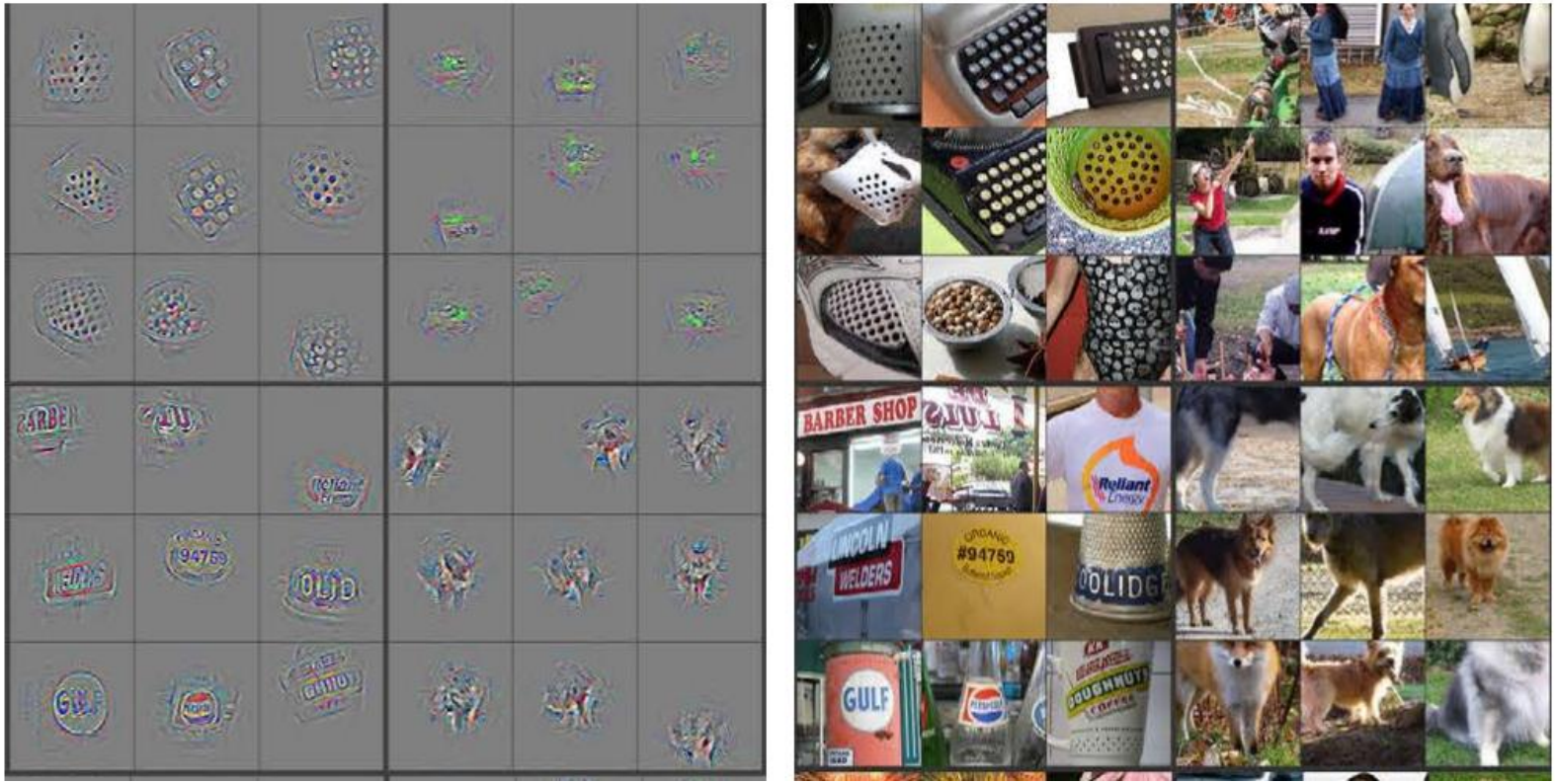
Visualizing Convolutional Networks

- Layer4

[1] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European Conference on Computer Vision*. Springer International Publishing, 2014.

Visualizing Convolutional Networks

- Layer5



[1] Zeiler, Matthew D., and Rob Fergus. "Visualizing and understanding convolutional networks." *European Conference on Computer Vision*. Springer International Publishing, 2014.

More Application: Playing Go

19 x 19 matrix (image)

Network

Next move (19 x 19 positions)

19 x 19 vector

Black: 1

white: -1

none: 0

Fully-connected feedforward network can be used

But CNN performs much better.

More Application: Playing Go

Training: record of previous plays

黑: 5之五 → 白: 天元 → 黑: 五之5 …

CNN

CNN

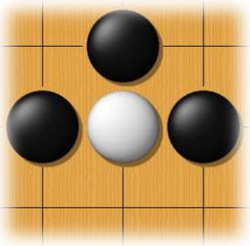
Target:
“天元” = 1
else = 0

Target:
“五之 5” = 1
else = 0

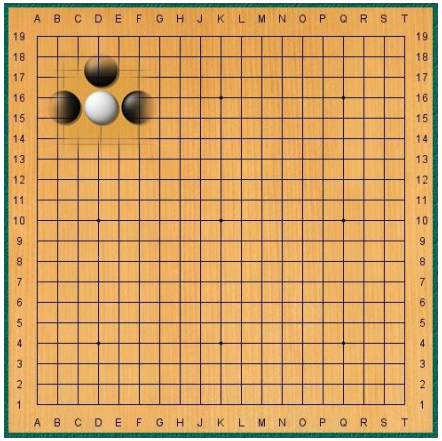
Why CNN for playing Go?

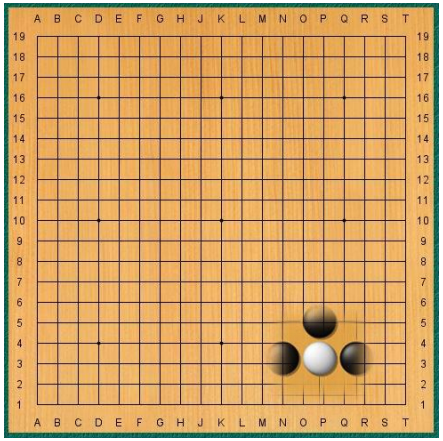
- Some patterns are much smaller than the whole image

  Alpha Go uses 5 x 5 for first layer



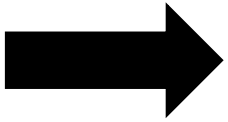
- The same patterns appear in different regions.





Why CNN for playing Go?

- Subsampling the pixels will not change the object



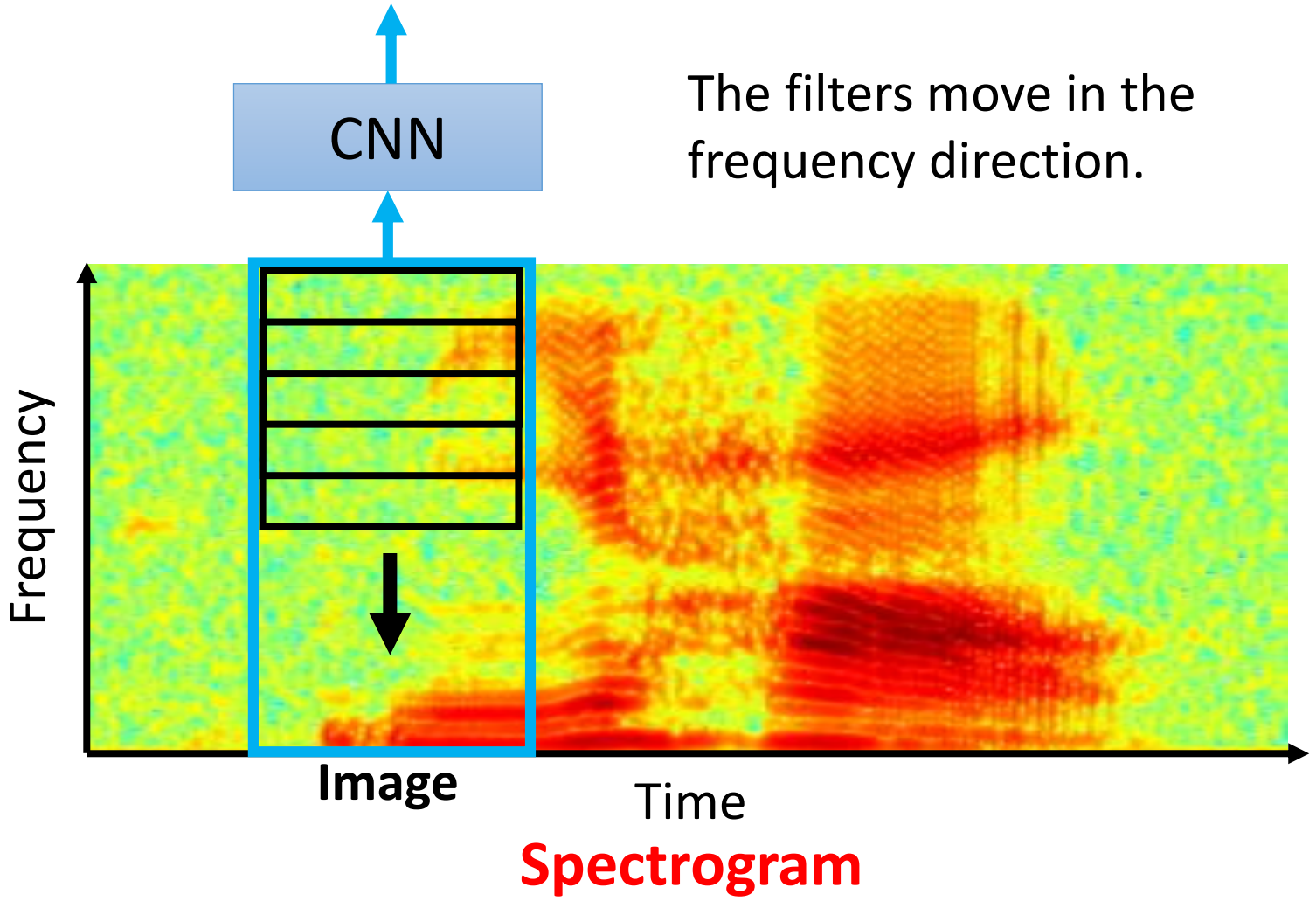
Max Pooling

How to explain this???

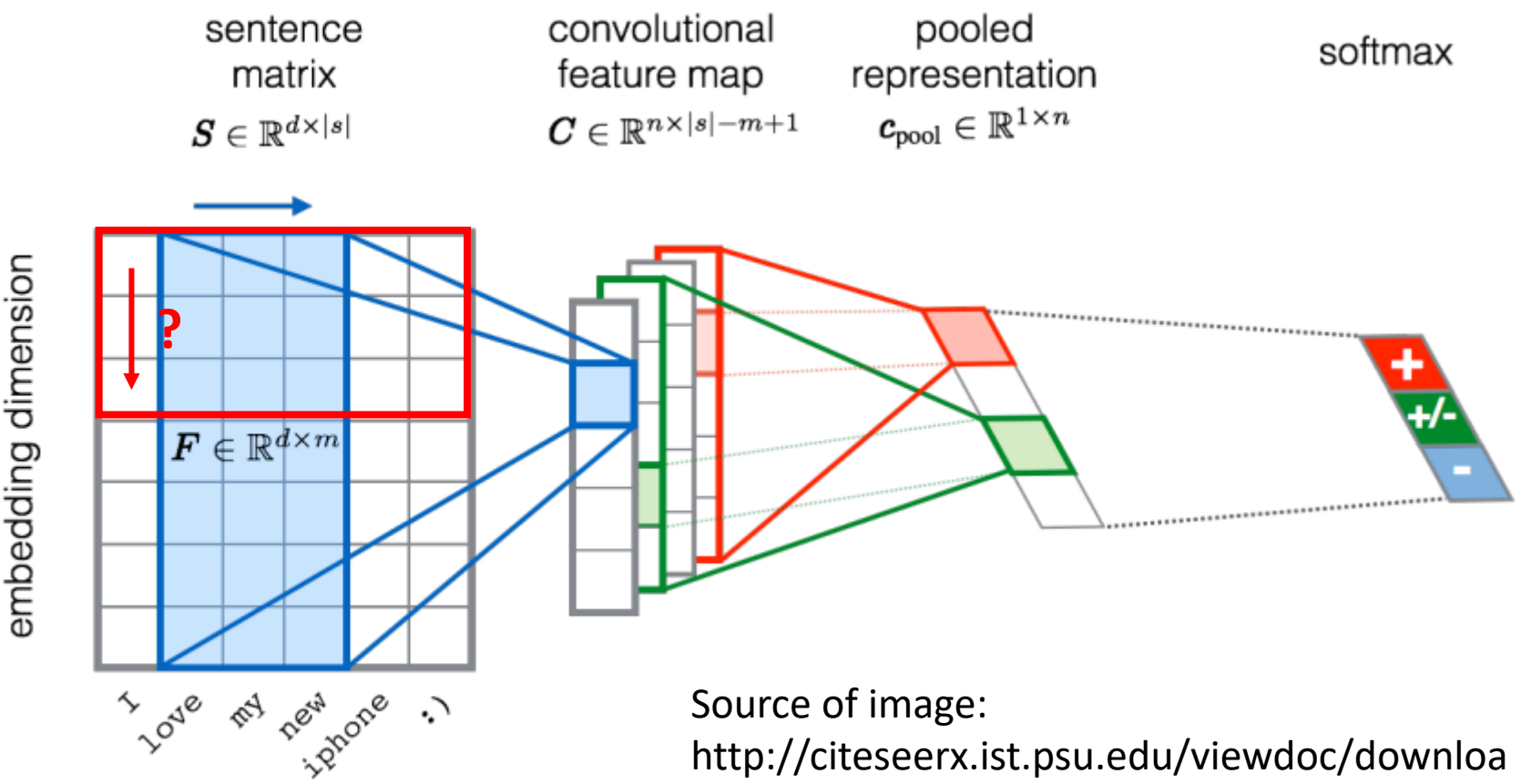
Neural network architecture. The input to the policy network is a $19 \times 19 \times 48$ image stack consisting of 48 feature planes. The first hidden layer zero pads the input into a 23×23 image, then convolves k filters of kernel size 5×5 with stride 1 with the input image and applies a rectifier 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 Extended Data Table 3 additionally show the results of training with $k = 128$, 256 and 384 filters.

Alpha Go does not use Max Pooling ……

More Application: Speech



More Application: Text



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