

Convolutional Neural Network

CNN or ConvNet

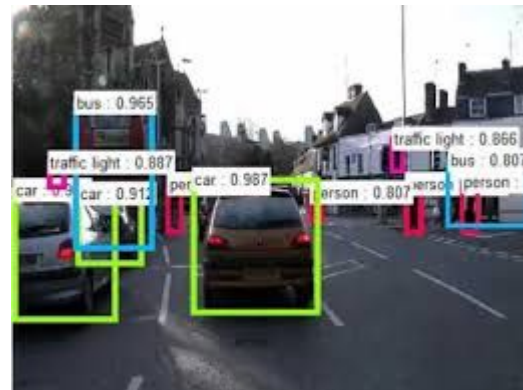
Computer Vision Challenges

Image Classification



Is it Cat?
[0/1]

Object Detection



Neural Style Transfer

Content Image +
Style Image =
Generated Image



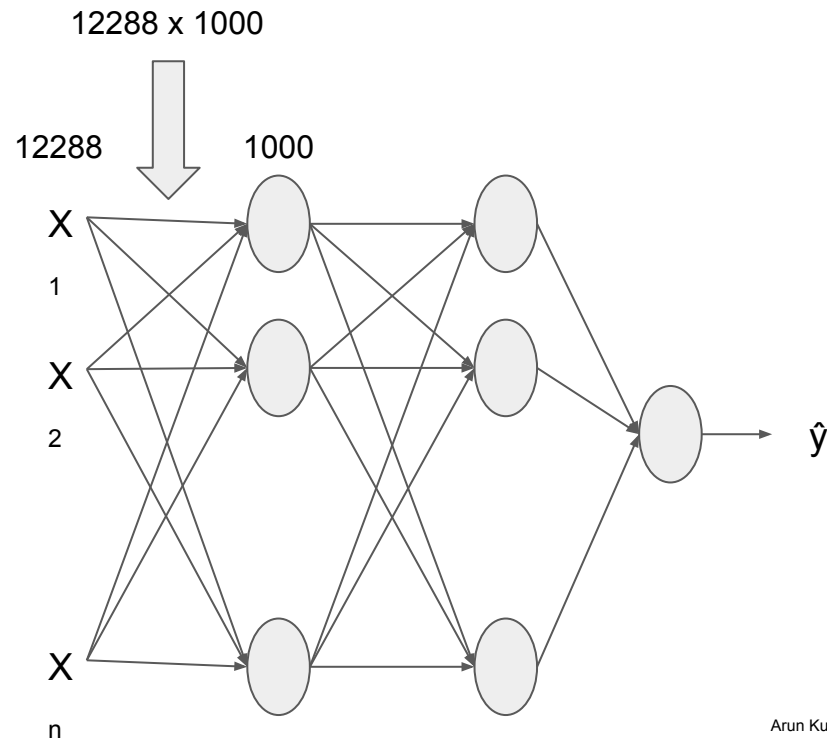
Fully Connected Network on Images

Image Classification



Is it Cat?
[0/1]

$$64 \times 64 \times 3 \\ = 12288$$



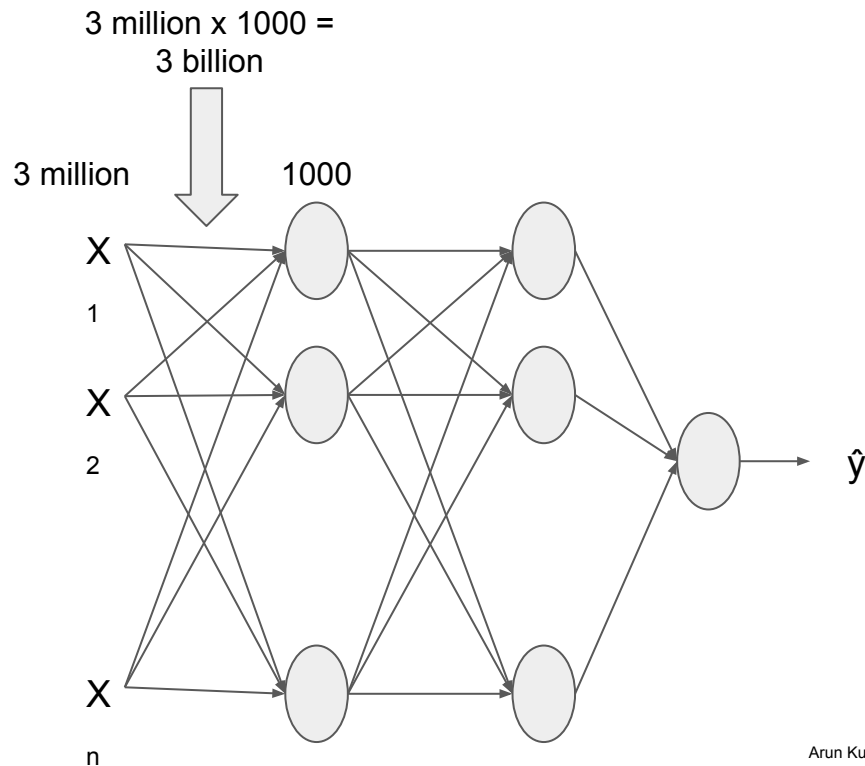
Fully Connected Network on Images

Image Classification

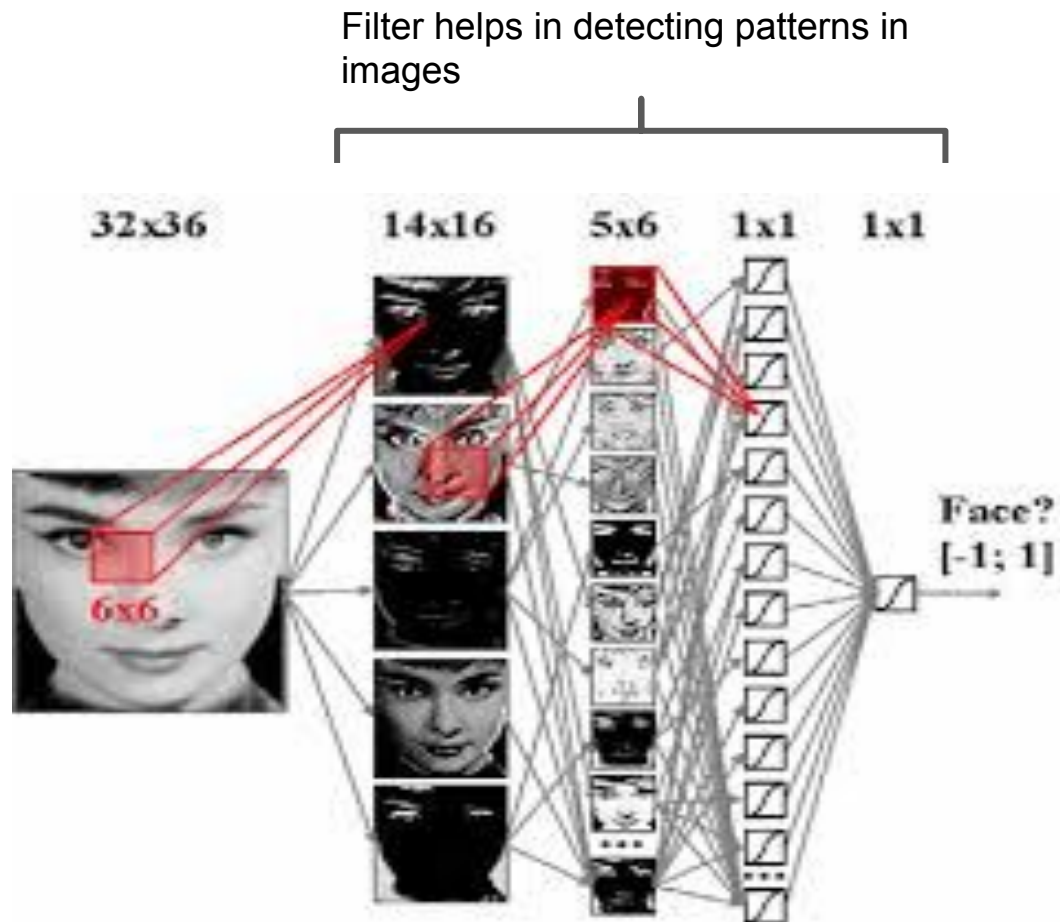


Is it Cat?
[0/1]

$$1000 \times 1000 \times 3 \\ = 3 \text{ million}$$

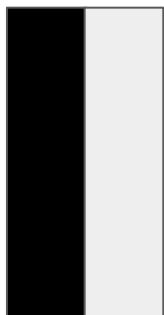


CNN - High Level



Filter - I know it

Detect Vertical Edge



10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

1	0	-1
1	0	-1
1	0	-1

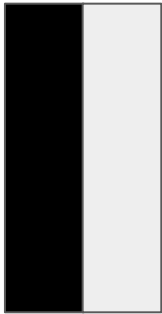
=

0			

$$[10 \times 1 + 10 \times 0 + 10 \times (-1)] + [10 \times 1 + 10 \times 0 + 10 \times (-1)] + [10 \times 1 + 10 \times 0 + 10 \times (-1)] = 0$$

Filter - I know it

Detect Vertical Edge



10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

1	0	-1
1	0	-1
1	0	-1

=

0	30		

$$[10 \times 1 + 10 \times 0 + 0 \times (-1)] + [10 \times 1 + 10 \times 0 + 0 \times (-1)] + [10 \times 1 + 10 \times 0 + 0 \times (-1)] = 30$$

Filter - I know it

Detect Vertical Edge



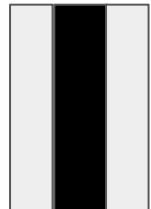
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

1	0	-1
1	0	-1
1	0	-1

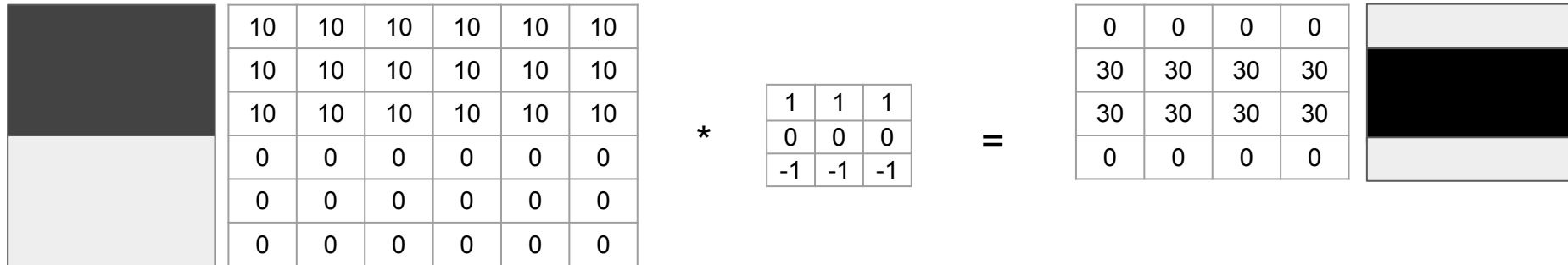
=

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0



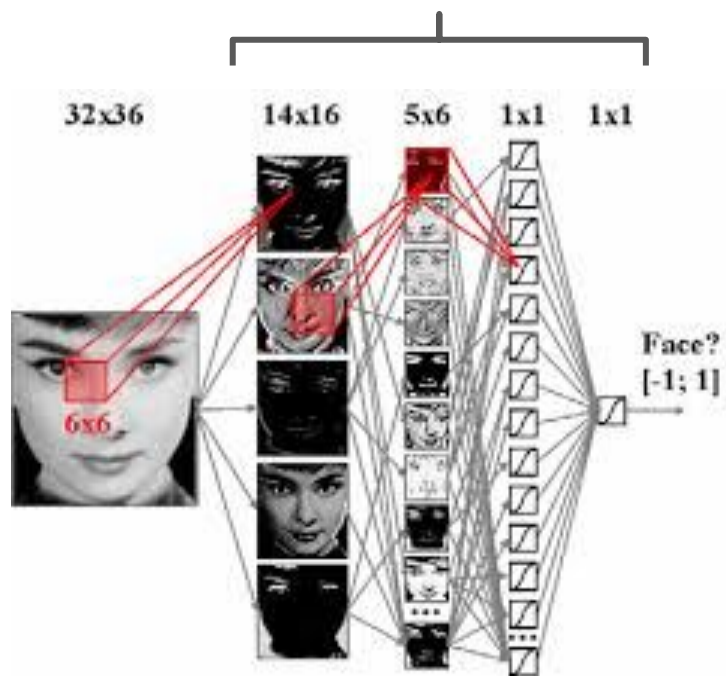
Filter - I know it

Detect Horizontal Edge



Filter - I don't know, but let CNN learn it

Filter helps in detecting patterns in images



Detect Pattern in Images

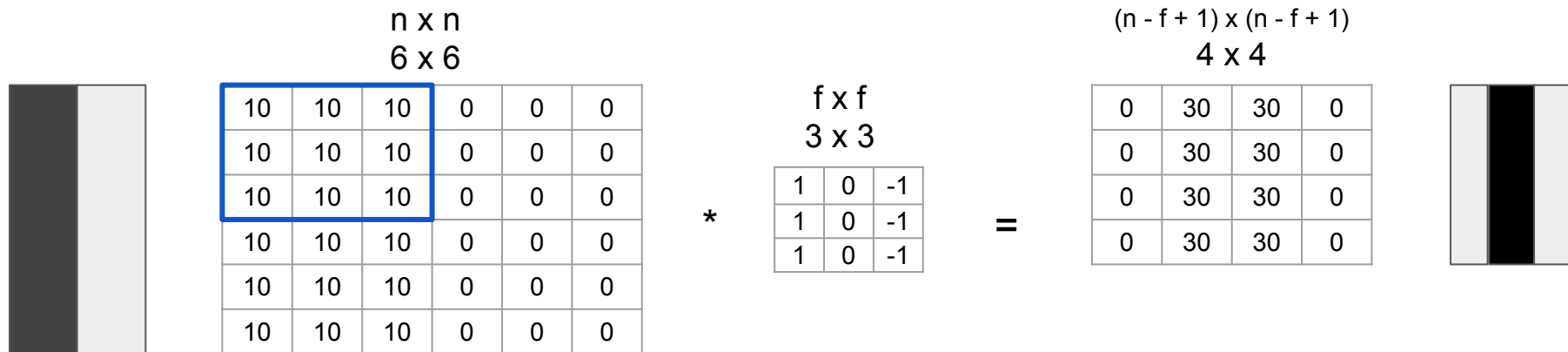
112	76	144	55	43	23
164	43	43	34	25	13
12	13	30	67	44	14
6	45	24	77	87	89
64	64	60	67	65	67
34	35	54	60	44	56



$$\begin{matrix}
 * & \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} & = & \text{[Filtered Image]}
 \end{matrix}$$

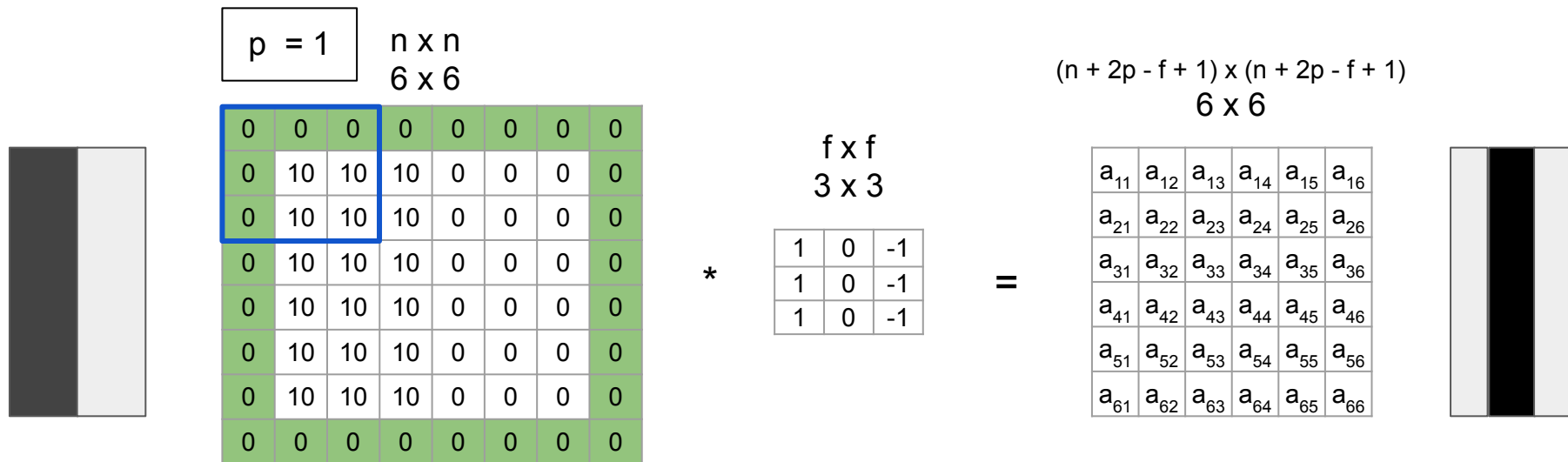
?

Valid Convolution



Output size is less than the input size

Same Convolution - Padding



Pad so that the output size is same as the input size

Strided Convolution

s = 2		n x n 7 x 7				
10	10	10	0	0	0	0
10	10	10	0	0	0	0
10	10	10	0	0	0	0
10	10	10	0	0	0	0
10	10	10	0	0	0	0
10	10	10	0	0	0	0

$$\left[\frac{(n + 2p - f)}{s} + 1 \right] \times \left[\frac{(n + 2p - f)}{s} + 1 \right]$$

3 x 3

1	0	-1
1	0	-1
1	0	-1

*

a ₁₁	a ₁₂	a ₁₃
a ₂₁	a ₂₂	a ₂₃
a ₃₁	a ₃₂	a ₃₃

=



3	4	6	2	1	7
3	5	4	9	8	8
1	7	5	4	6	6
7	2	6	5	4	5
6	1	7	7	2	4
5	3	3	1	4	3

6 x 6 x 1



		3	4	6	2	1	7	
	3	4	6	2	1	7		8
3	4	6	2	1	7		8	6
3	5	4	9	8	8		6	5
1	7	5	4	6	6		5	4
7	2	6	5	4	5		4	3
6	1	7	7	2	4		3	
5	3	3	1	4	3			

6 x 6 x 3

Filter for RGB

		1	0	-1
	1	0	-1	-1
1	0	-1	-1	-1
1	0	-1	-1	
1	0	-1		

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0

-1											
						3	4	6	2	1	7
						3	5	4	9	8	8
						1	7	5	4	6	6
			3	4	6	2	1	7	5	4	5
			3	5	4	9	8	8	7	2	4
			1	7	5	4	6	6	1	4	3
3	4	6	2	1	7	5	4	5			
3	5	4	9	8	8	7	2	4			
1	7	5	4	6	6	1	4	3			
7	2	6	5	4	5						
6	1	7	7	2	4						
5	3	3	1	4	3						

						3	4	6	2	1	7
						3	5	4	9	8	8
						1	7	5	4	6	6
			3	4	6	2	1	7	5	4	5
			3	5	4	9	8	8	7	2	4
			1	7	5	4	6	6	1	4	3
3	4	6	2	1	7	5	4	5			
3	5	4	9	8	8	7	2	4			
1	7	5	4	6	6	1	4	3			
7	2	6	5	4	5						
6	1	7	7	2	4						
5	3	3	1	4	3						

Question

In a layer, if you have 20 filters of dimension 3x3x3 each, then how many parameter does that layer have?

For 1 filter

$$3 \times 3 \times 3 = 27 \text{ weights}$$

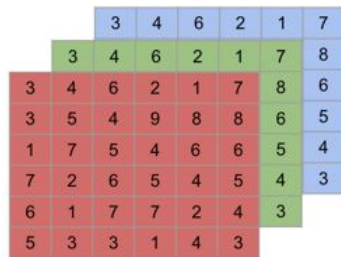
1 = bias

So 28 parameter for 1 layer

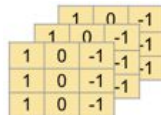
For 20 filters in a layer

$$28 \times 20 = \mathbf{560 \text{ parameters}}$$

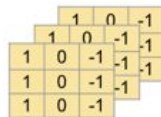
Example of Convolutional layer



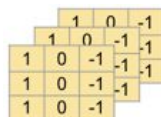
6x6x3



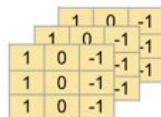
3x3x3



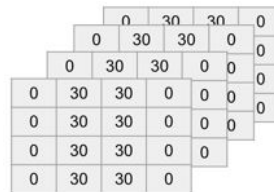
3x3x3



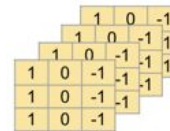
3x3x3



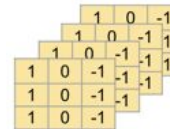
3x3x3



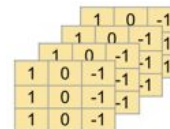
4x4x4



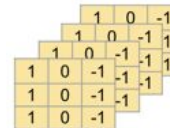
3x3x4



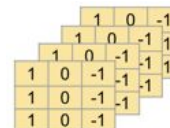
3x3x4



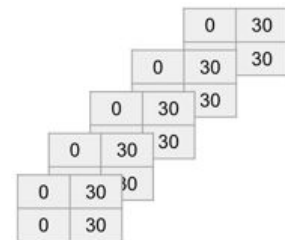
3x3x4



3x3x4

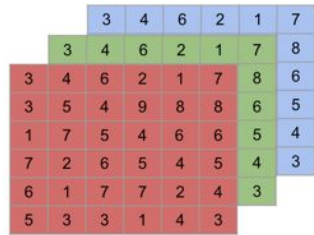


3x3x4



2x2x5

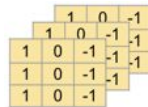
Input : (L-1)



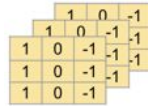
6 x 6 x 3

$$n^{L-2}_H \times n^{L-2}_W \times n^{L-2}_C$$

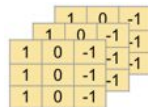
Layer : (L-1)



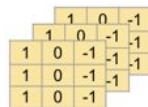
3 x 3 x 3



3 x 3 x 3



3 x 3 x 3



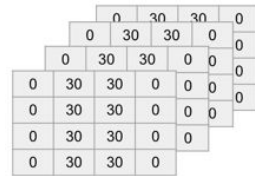
3 x 3 x 3

$$f^{L-1}_H \times f^{L-1}_W \times n^{L-2}_C \times n^{L-1}_C$$

s^{L-1}p^{L-1}

Output : (L-1)

Input : (L)



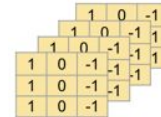
4 x 4 x 4

$$n^{L-1}_H \times n^{L-1}_W \times n^{L-1}_C$$

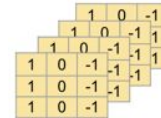
$$n^{L-1}_H = \frac{n^{L-2}_H - f^{L-1}_H + 1}{1}$$

$$n^{L-1}_H = (n^{L-2}_H + 2p^{L-1} - f^{L-1}_H) / s^{L-1} + 1$$

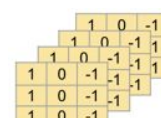
Layer : (L)



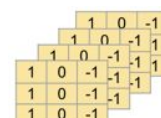
3 x 3 x 4



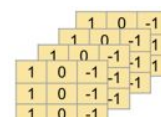
3 x 3 x 4



3 x 3 x 4



3 x 3 x 4

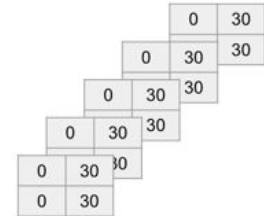


3 x 3 x 4

$$f^L_H \times f^L_W \times n^{L-1}_C \times n^L_C$$

s^Lp^L

Output : (L)



2 x 2 x 5

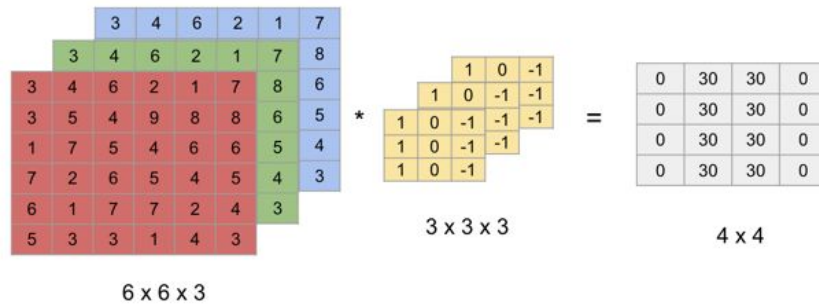
$$n^L_H \times n^L_W \times n^L_C$$

$$n^L_H = n^{L-1}_H - f^L_H + 1$$

$$n^L_H = (n^{L-1}_H + 2p^L - f^L_H) / s^L + 1$$

Types of layer in Convolutional Network

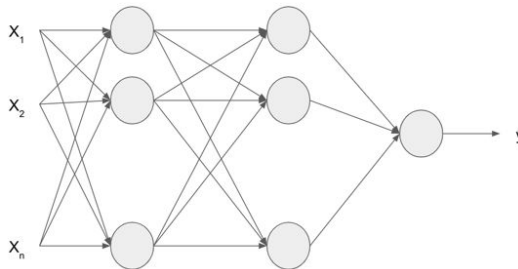
Convolution



Pool



Fully
Connected



Max Pooling Layer

3	4	6	2	1	7
3	5	4	9	8	8
1	7	5	4	6	6
7	2	6	5	4	5
6	1	7	7	2	4
5	3	3	1	4	3

6 x 6 x 1

f = 3



7	9
7	7

2 x 2 x 1

Max Pooling Layer

			3	4	6	2	1	7	
		3	4	6	2	1	7	8	
	3	4	6	2	1	7	8	6	
3	5	4	9	8	8	6	5		
1	7	5	4	6	6	5	4		
7	2	6	5	4	5	4	3		
6	1	7	7	2	4	3			
5	3	3	1	4	3				

$6 \times 6 \times 3$

$f = 3$



			7		9	
		7		9		
	7		9			
7		9				
7		7				

$2 \times 2 \times 3$

Average Pooling Layer

3	4	6	2	1	7
3	5	4	9	8	8
1	7	5	4	6	6
7	2	6	5	4	5
6	1	7	7	2	4
5	3	3	1	4	3

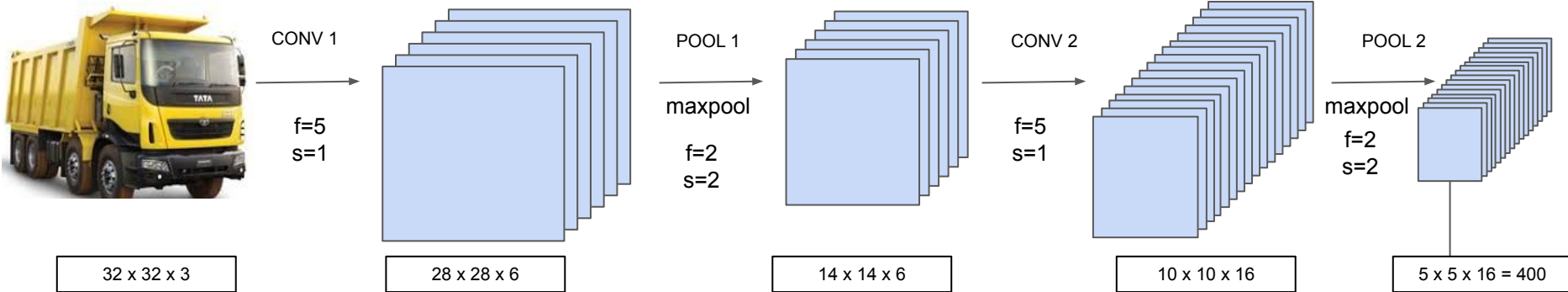
6 x 6 x 1

f = 3



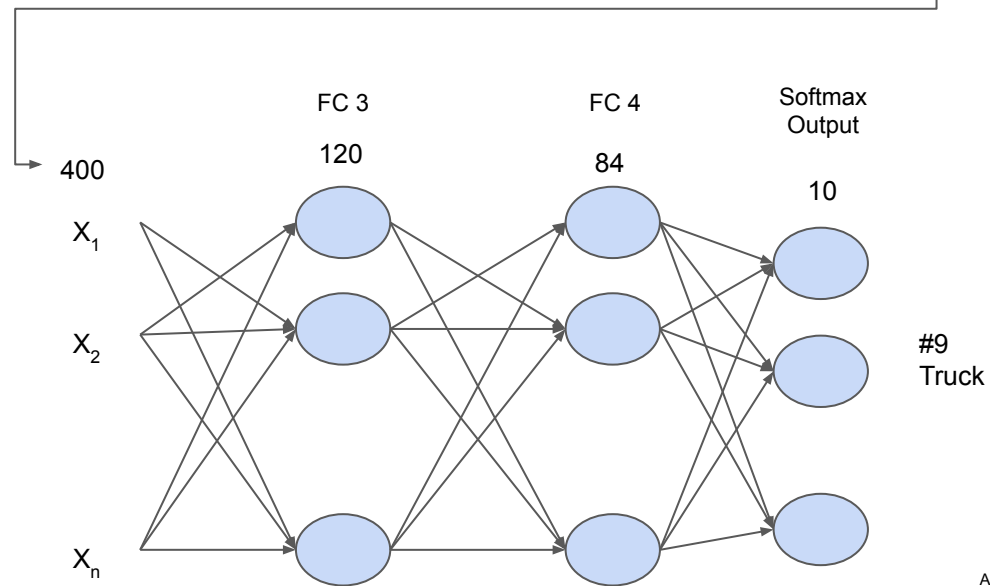
4.22	5.66
4.44	3.89

2 x 2 x 1



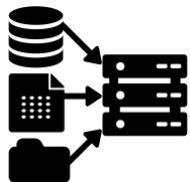
CNN Example

Image Classification using
CIFAR10 dataset



Exercise Image Classification

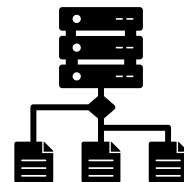
Load Dataset



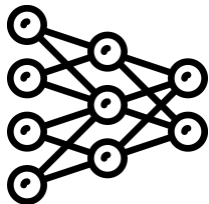
Analyze Dataset



Prepare Dataset



Define Model



Train & Evaluate Model



Summarize Results



Predict New Image



Model Summary

Layer (type)	Input layer dimensions	Output Shape	Params #	Comments
Input		(32, 32, 3)		
CONV 1	$5 \times 5 \times 3 \times 6$	(28, 28, 6)	456	$(5 \times 5 \times 3 + 1) \times 6$
POOL 1	2×2	(14, 14, 6)	0	
CONV 2	$5 \times 5 \times 6 \times 16$	(10, 10, 16)	2416	$(5 \times 5 \times 6 + 1) \times 16$
POOL 2	2×2	(5, 5, 16)	0	
FC 1	$5 \times 5 \times 16 = 400$	120	48120	$(400 + 1) \times 120$
FC 2	120	84	10164	$(120 + 1) \times 84$
Softmax Output	84	10	850	$(84 + 1) \times 10$
Total			62,006	

CNN vs Fully Connected Network on Images

Image Classification



$64 \times 64 \times 3$
 $= 12288$

Is it Cat?
[0/1]

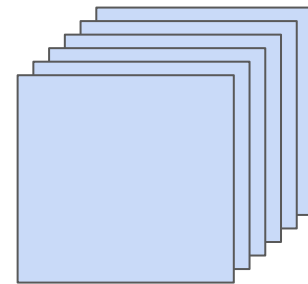


$64 \times 64 \times 3 =$
 12288

Filters# 6
 $5 \times 5 \times 3$

Strides = 2

Parameters to
learn
 $= (5 \times 5 \times 3 + 1) \times 6$
 $= 456$



$30 \times 30 \times 6 =$
 5400

