Convolutional Neural Network

CNN or ConvNet

Arun Kumar Anala
analaarun.k@gmail.com

Computer Vision Challenges

Image Classification

Is it Cat?
[0/1]

Object Detection

Neural Style Transfer

Content Image +
Style Image =
Generated Imgae

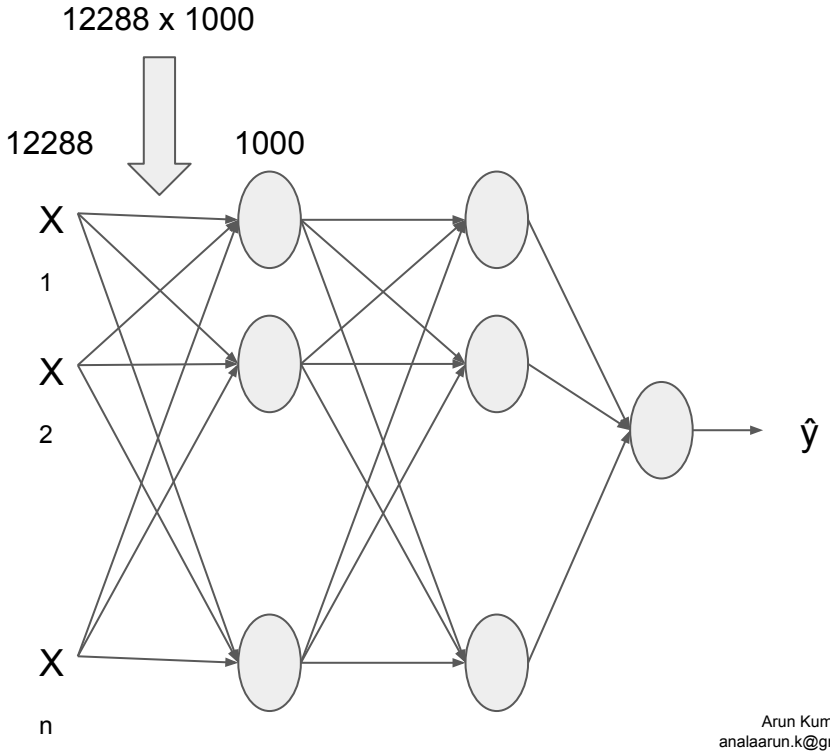
Arun Kumar Anala
analaarun.k@gmail.com

Fully Connected Network on Images

Image Classification

64 X 64 X 3
= 12288

Is it Cat?
[0/1]



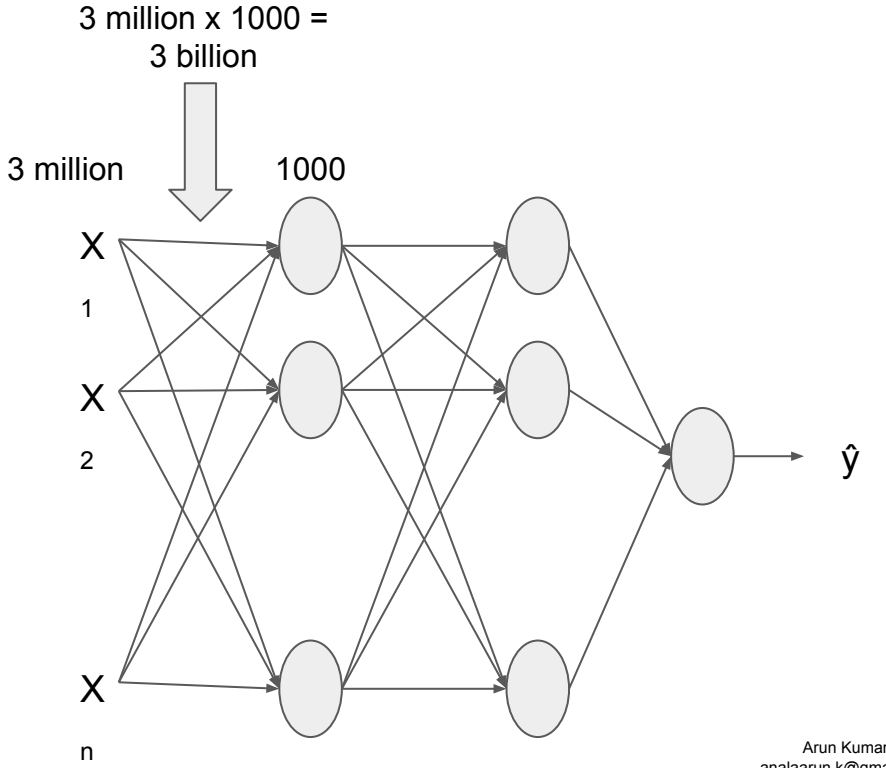
Arun Kumar Anala
analaarun.k@gmail.com

Fully Connected Network on Images

Image Classification

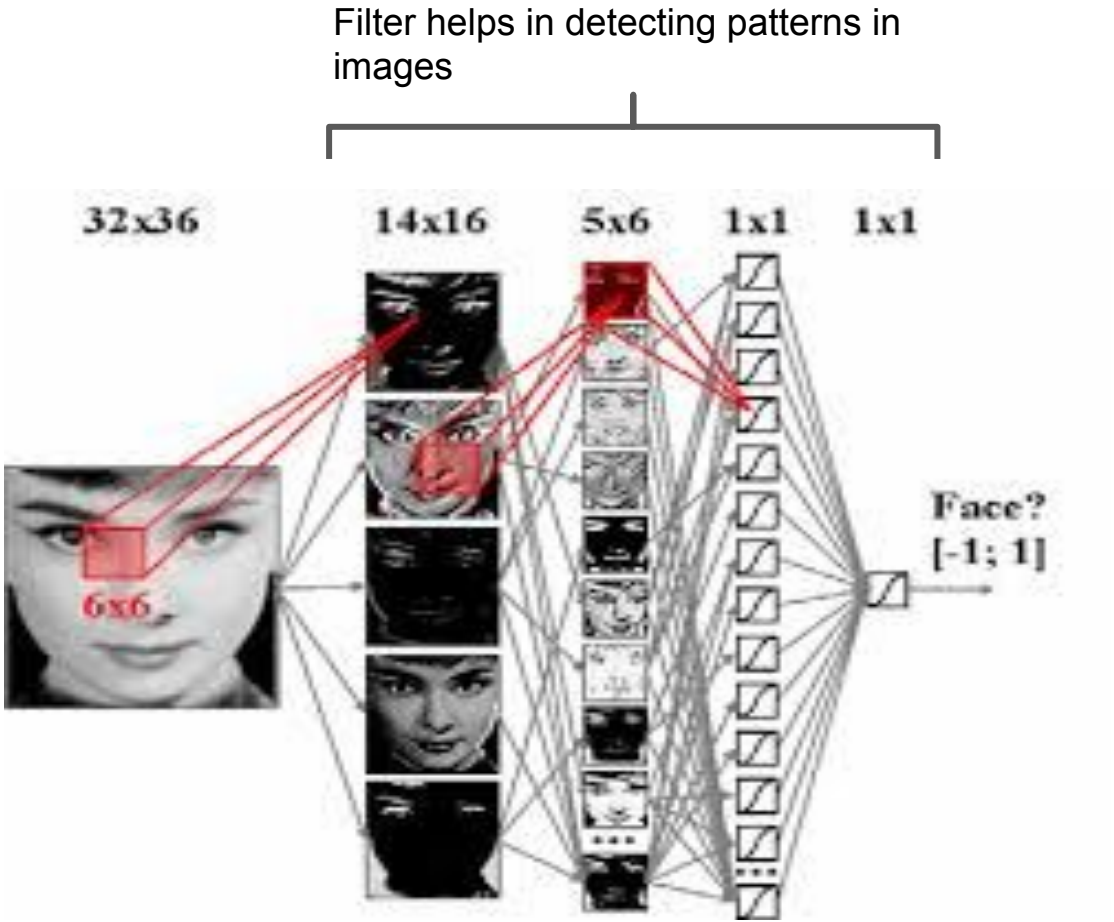
Is it Cat?
[0/1]

1000 X 1000 X 3
= 3 million



Arun Kumar Anala
analaarun.k@gmail.com

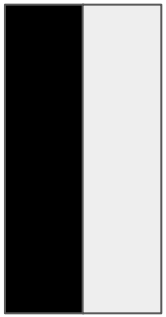
CNN - High Level



Arun Kumar Anala
analaarun.k@gmail.com

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

=

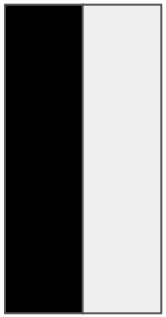
0			

[10 x 1 + 10 x 0 + 10 x (-1)] + [10 x 1 + 10 x 0 + 10 x (-1)] + [10 x 1 + 10 x 0 + 10 x (-1)] = 0

Arun Kumar Anala
analaarun.k@gmail.com

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

=

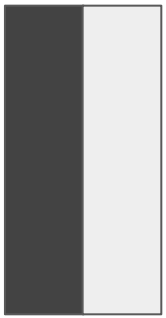
0	30		

[10 x 1 + 10 x 0 + 0 x (-1)] + [10 x 1 + 10 x 0 + 0 x (-1)] + [10 x 1 + 10 x 0 + 0 x (-1)] = 30

Arun Kumar Anala
analaarun.k@gmail.com

Filter - I know it

Detect Vertical Edge



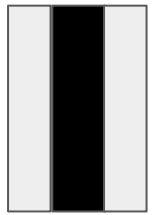
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0
10	10	10	0	0	0

*

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

=

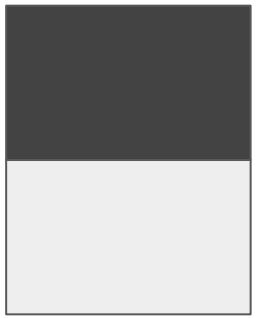
0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0



Arun Kumar Anala
analaarun.k@gmail.com

Filter - I know it

Detect Horizontal Edge



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

*

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

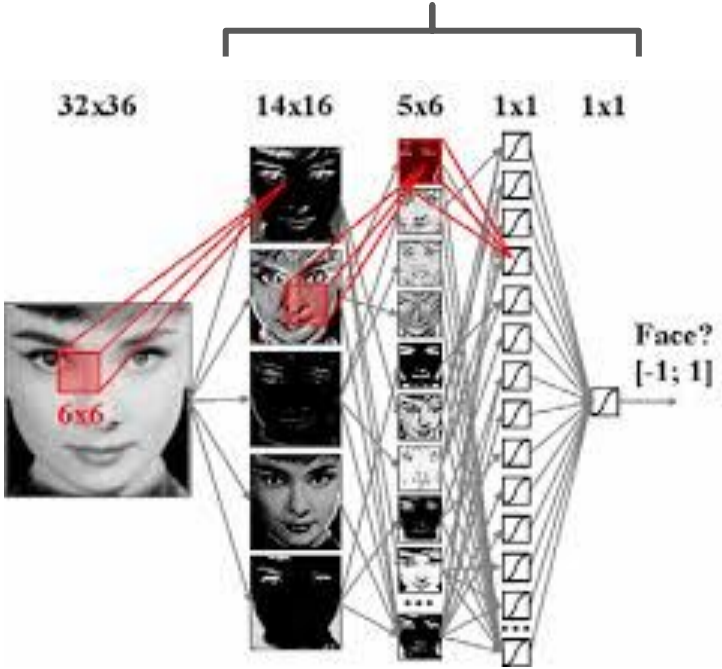
=

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

Arun Kumar Anala
analaarun.k@gmail.com

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

*

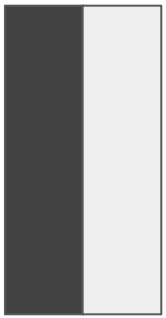
w_{11}	w_{12}	w_{13}
w_{21}	w_{22}	w_{23}
w_{31}	w_{32}	w_{33}

?

=

Arun Kumar Anala
analaarun.k@gmail.com

Valid Convolution



n x n
6 x 6

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

*

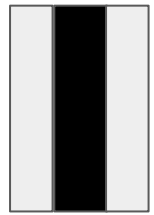
f x f
3 x 3

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

=

(n - f + 1) x (n - f + 1)
4 x 4

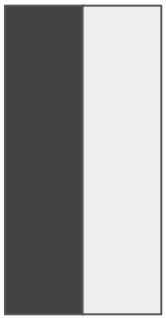
0	30	30	0
0	30	30	0
0	30	30	0
0	30	30	0



Output size is less than the input size

Arun Kumar Anala
analaarun.k@gmail.com

Same Convolution - Padding



p = 1

n x n
6 x 6

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

*

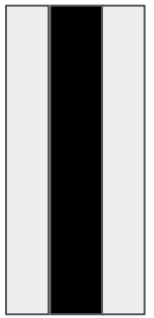
f x f
3 x 3

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

=

(n + 2p - f + 1) x (n + 2p - f + 1)
6 x 6

a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}
a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}
a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}
a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	a_{46}
a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	a_{56}
a_{61}	a_{62}	a_{63}	a_{64}	a_{65}	a_{66}



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

Arun Kumar Anala
analaarun.k@gmail.com

Strided Convolution

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

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
10	10	10	0	0	0	0

*

f x f
3 x 3

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

=

3 x 3

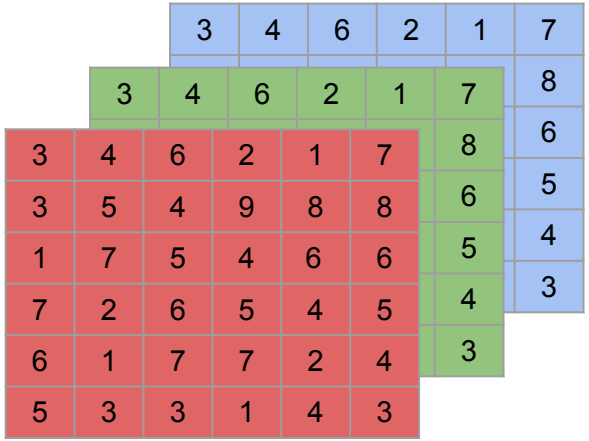
a_{11}	a_{12}	a_{13}
a_{21}	a_{22}	a_{23}
a_{31}	a_{32}	a_{33}

Arun Kumar Anala
analaarun.k@gmail.com

Image - RGB vs B&W

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

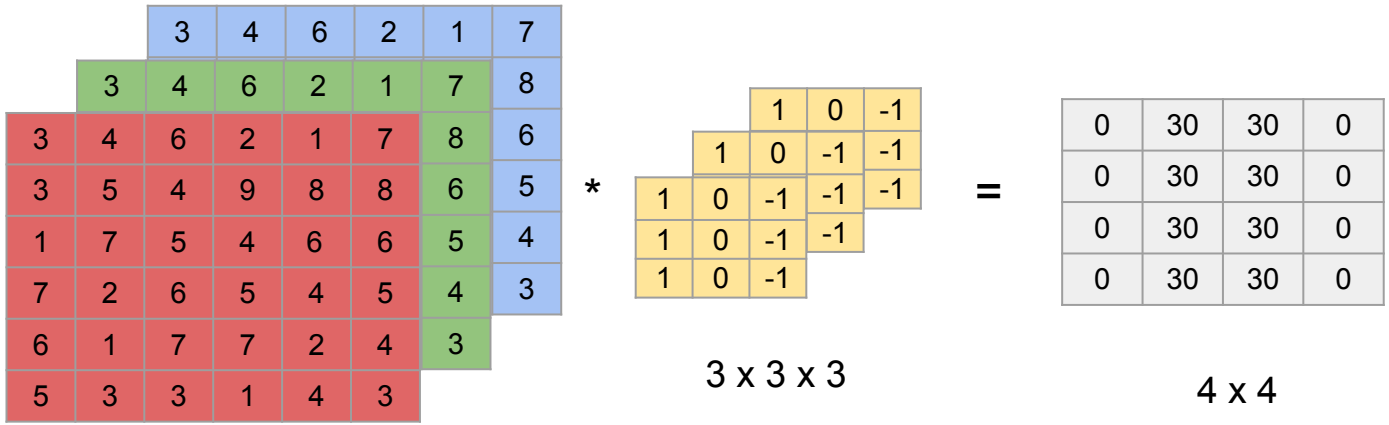
6 x 6 x 1



6 x 6 x 3

Arun Kumar Anala
analaarun.k@gmail.com

Filter for RGB



Arun Kumar Anala
analaarun.k@gmail.com

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

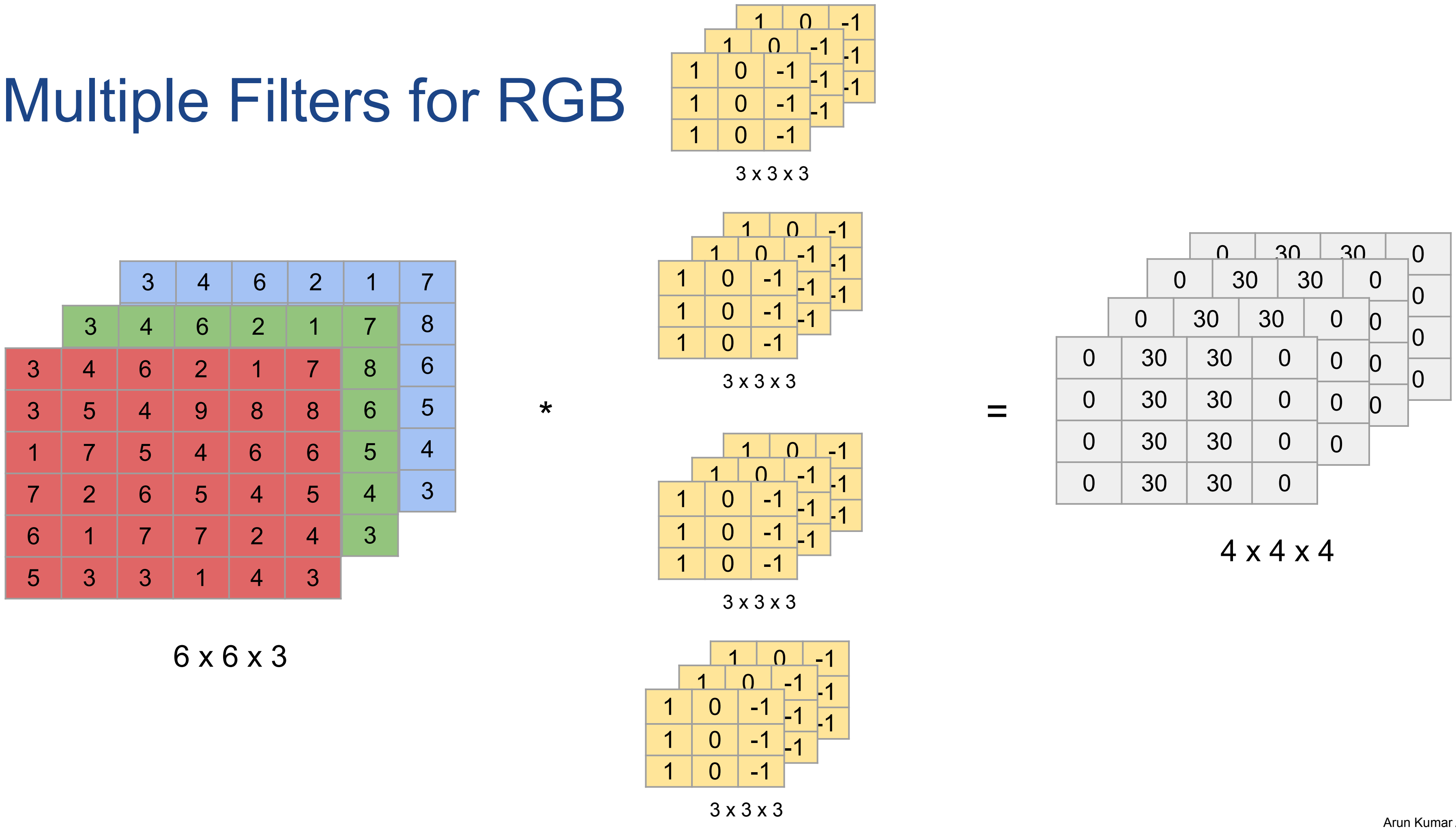
						3	4	6	2	1	7
						3	5	4	9	8	8
						1	7	5	4	6	6
			3	4	6	2	1	7	5	4	5
			3	5	4	9	8	8	7	2	4
			1	7	5	4	6	6	1	4	3
3	4	6	2	1	7	5	4	5			
3	5	4	9	8	8	7	2	4			
1	7	5	4	6	6	1	4	3			
7	2	6	5	4	5						
6	1	7	7	2	4						
5	3	3	1	4	3						

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

						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						

Arun Kumar Anala
analaarun.k@gmail.com

Multiple Filters for RGB



Arun Kumar Anala
analaarun.k@gmail.com

Question

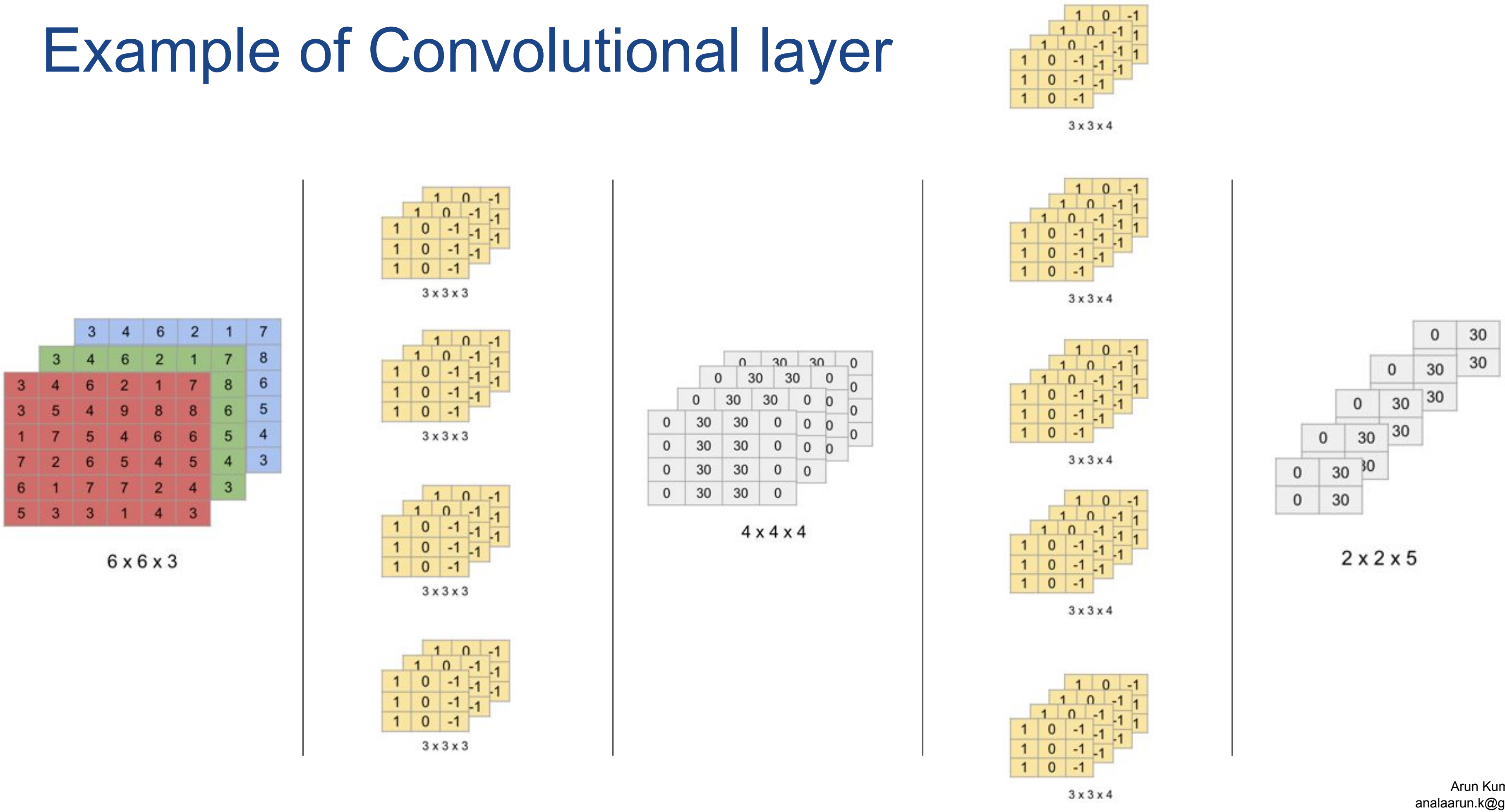
In a layer, if you have 20 filters of dimension 3x3x3 each, then how many parameter does that layer have?

For 1 filter
3 x 3 x 3 = 27 weights
1 = bias
So 28 parameter for 1 layer

For 20 filters in a layer
28 x 20 = **560 parameters**

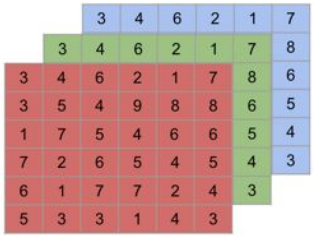
Arun Kumar Anala
analaarun.k@gmail.com

Example of Convolutional layer



Arun Kumar Anala
analaarun.k@gmail.com

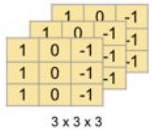
Input : (L-1)



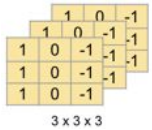
6 x 6 x 3

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

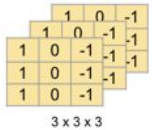
Layer : (L-1)



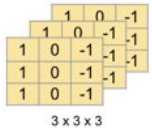
3 x 3 x 3



3 x 3 x 3



3 x 3 x 3



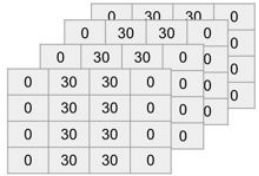
3 x 3 x 3

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

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

Output : (L-1)

Input : (L)



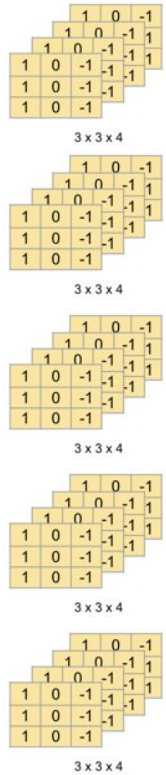
4 x 4 x 4

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

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

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

Layer : (L)



3 x 3 x 4

3 x 3 x 4

3 x 3 x 4

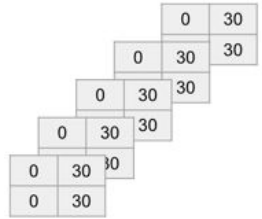
3 x 3 x 4

3 x 3 x 4

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

s^L p^L

Output : (L)



2 x 2 x 5

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

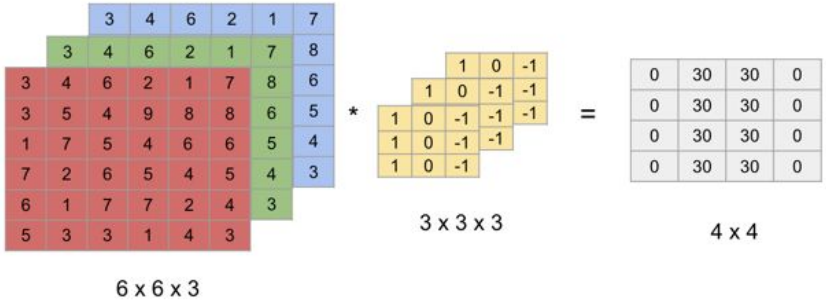
$n^L_H = n^{L-1}_H - f^L_H + 1$

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

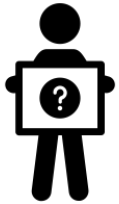
Arun Kumar Anala
analaarun.k@gmail.com

Types of layer in Convolutional Network

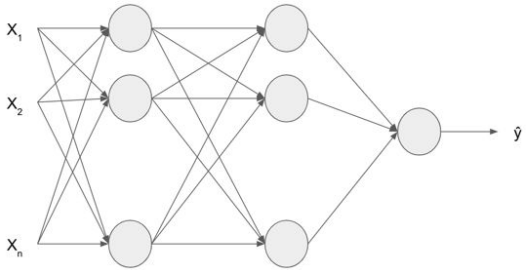
Convolution



Pool



Fully Connected



Arun Kumar Anala
analaarun.k@gmail.com

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

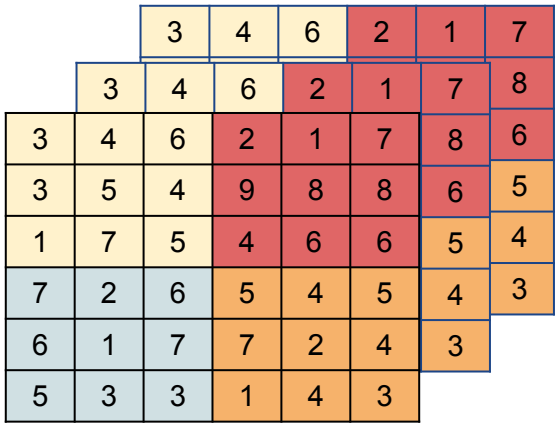
f = 3

7	9
7	7

2 x 2 x 1

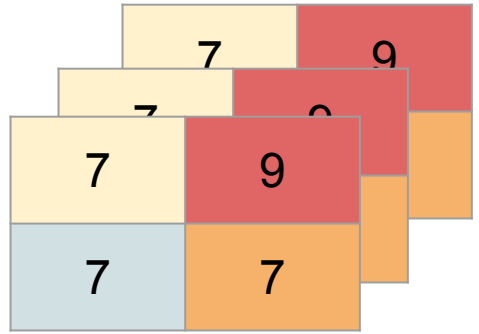
Arun Kumar Anala
analaarun.k@gmail.com

Max Pooling Layer



6 x 6 x 3

f = 3



2 x 2 x 3

Arun Kumar Anala
analaarun.k@gmail.com

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

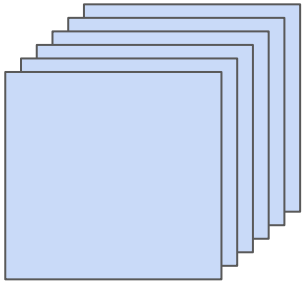
4.22	5.66
4.44	3.89

2 x 2 x 1

Arun Kumar Anala
analaarun.k@gmail.com

CONV 1

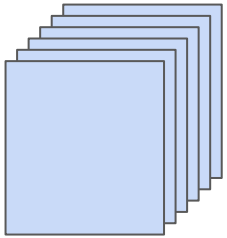
f=5
s=1



POOL 1

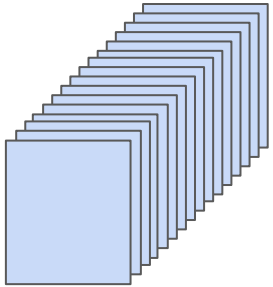
maxpool

f=2
s=2



CONV 2

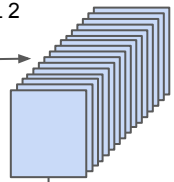
f=5
s=1



POOL 2

maxpool

f=2
s=2



32 x 32 x 3

28 x 28 x 6

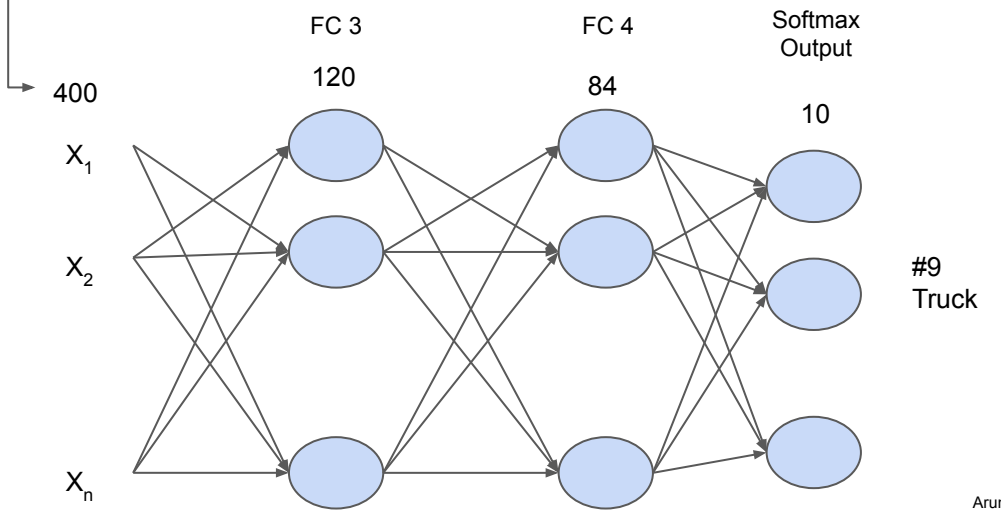
14 x 14 x 6

10 x 10 x 16

5 x 5 x 16 = 400

CNN Example

Image Classification using CIFAR10 dataset



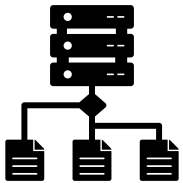
Arun Kumar Anala
analaarun.k@gmail.com

Exercise Image Classification

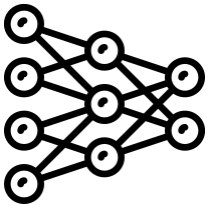
Load Dataset

Analyze Dataset

Prepare Dataset



Define Model



Train & Evaluate Model

Summarize Results

Predict New Image

Arun Kumar Anala
analaarun.k@gmail.com

Model Summary

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

Arun Kumar Anala
analaarun.k@gmail.com

CNN vs Fully Connected Network on Images

Image Classification

64 X 64 X 3
= 12288

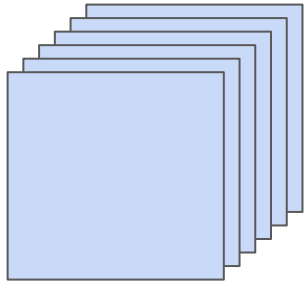
Is it Cat?
[0/1]

64 X 64 X 3 =
12288

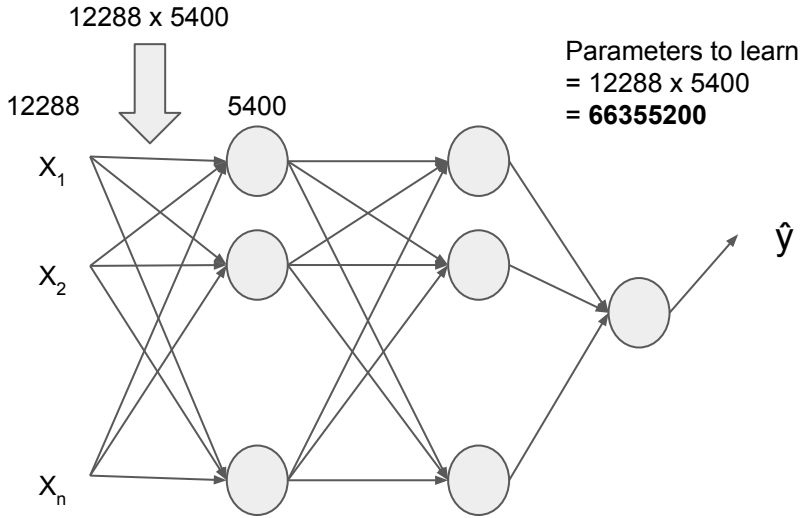
Filters# 6
5 x 5 x 3

Strides = 2

Parameters to learn
= (5 x 5 x 3 + 1) x 6
= **456**



30 X 30 X 6 =
5400



Arun Kumar Anala
analaarun.k@gmail.com