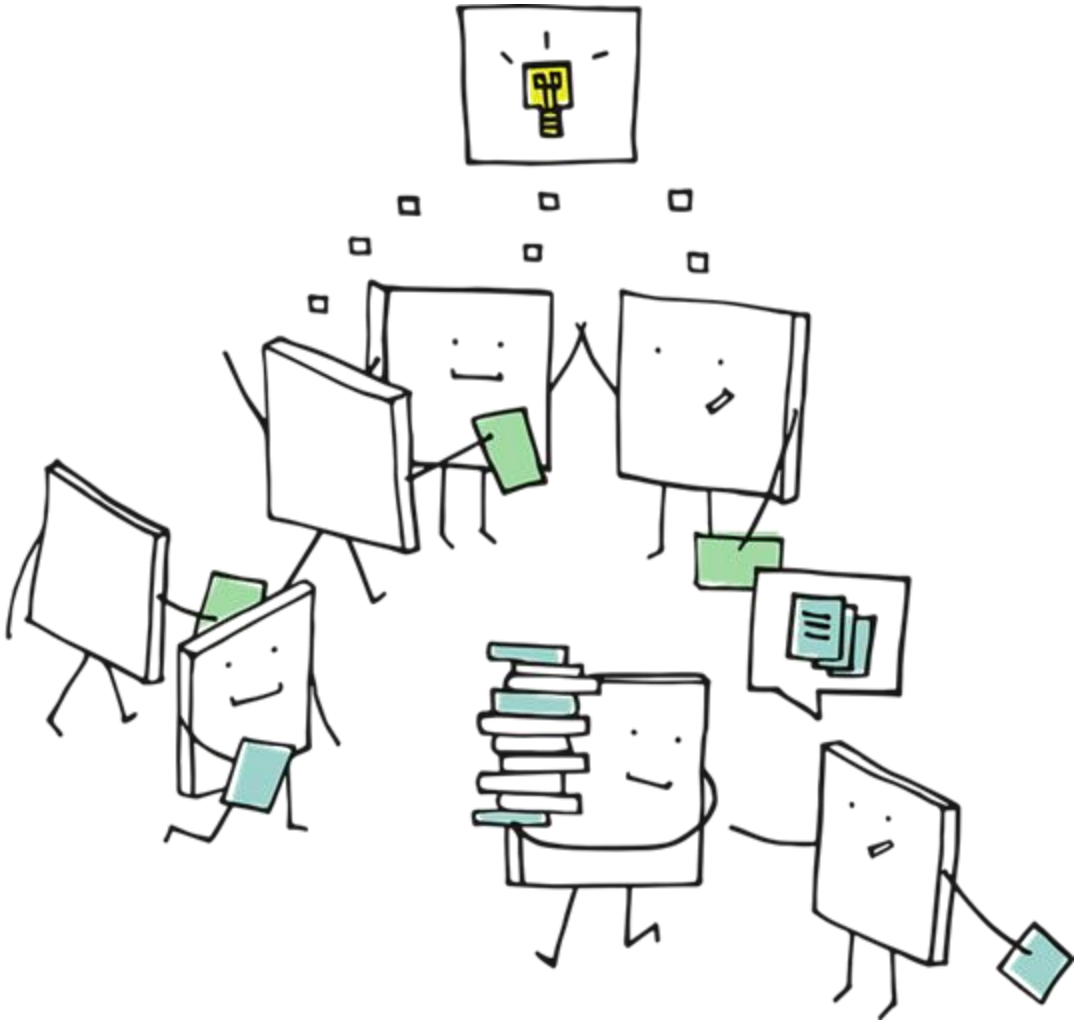


Convolutional Neural Network (CNN)

Limitations of Dense Layer

Convolutional Layer

Convolutional Neural Network (CNN)

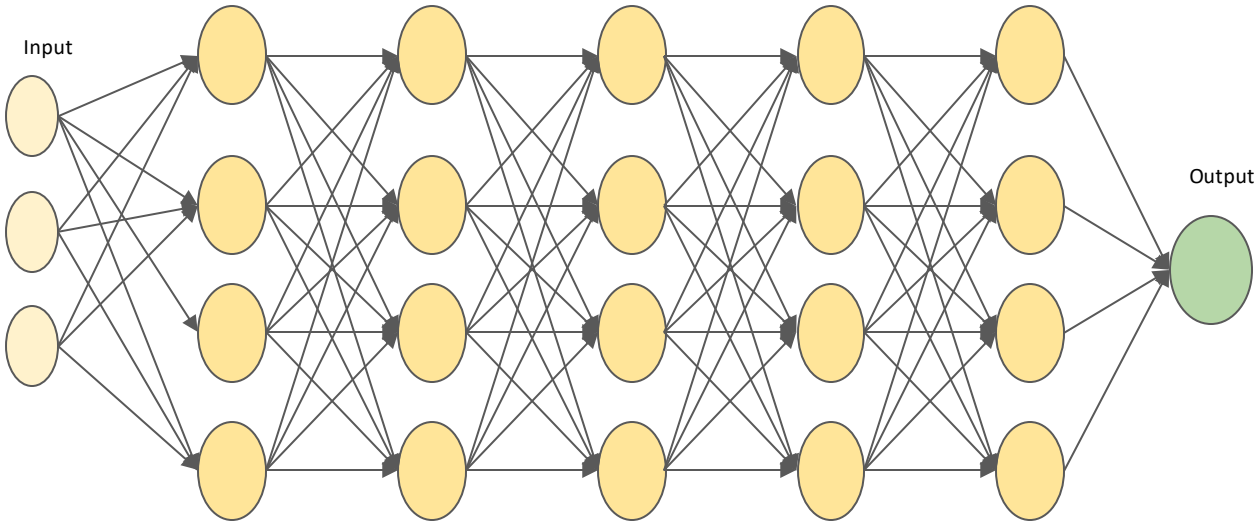
Image Processing & Augmentation

ImageNet role in Computer Vision

CNN Architectures

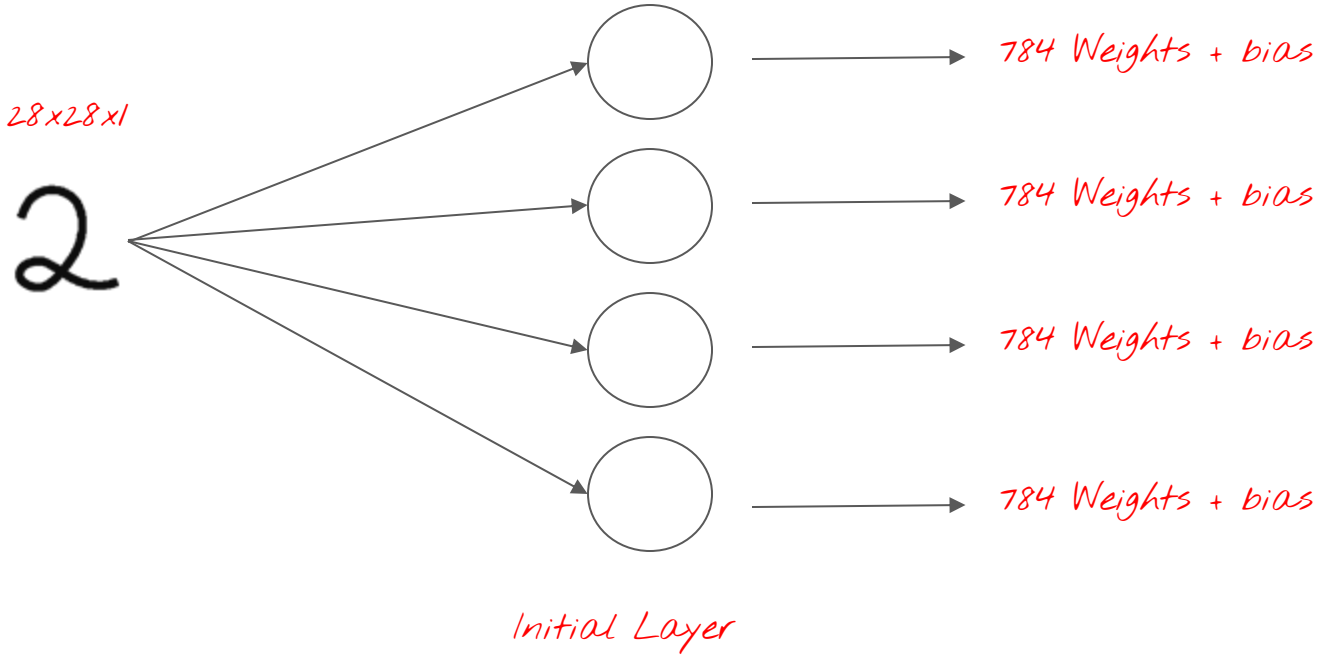
Transfer Learning

Revisiting Dense Layers



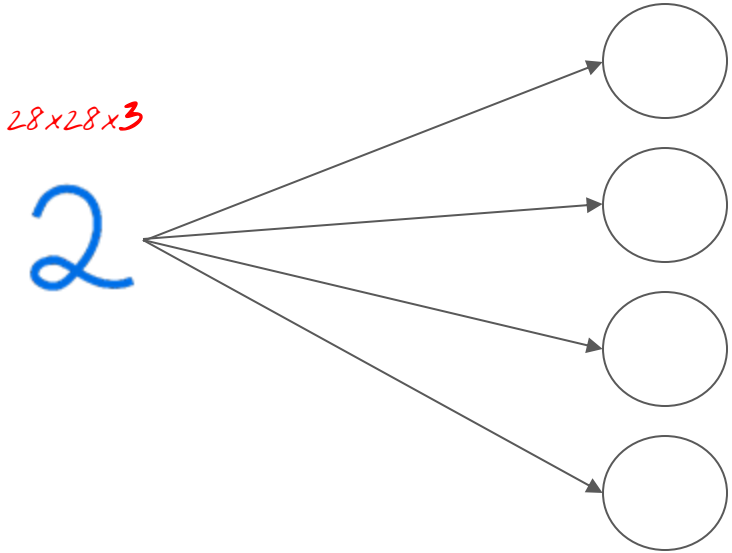
Also called Fully connected Layers

Number of Weights



200 Neurons * (784+1) = 157,000

Number of Weights



Initial Layer

How many weights for each Neuron?

2,352 + 1

200 Neurons * (2352+1) = 470,600

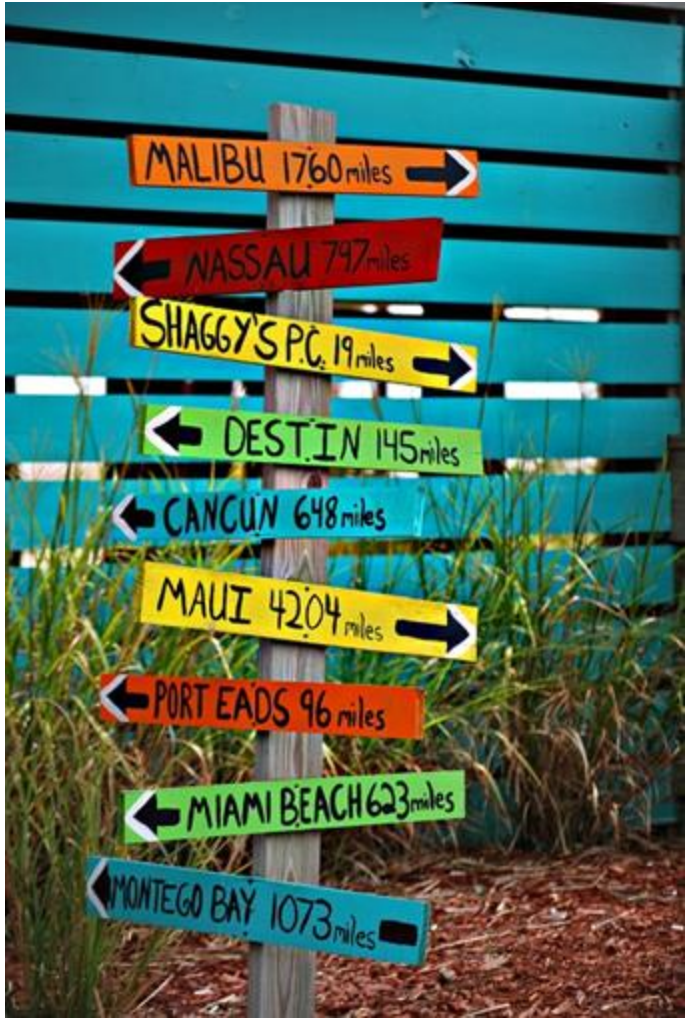
Number of Weights

300x300x3

300 * 300 * 3 + 1

200 Neurons * (270,000+1) = 54,000,200

Network becomes unmanageable :(



Spatial Information

Humans

32	27	33	12	18	29	37	27
29	18	28	16	27	18	29	30
2	33	66	155	160	23	32	28
32	27	99	180	192	86	199	100
47	90	180	190	170	30	21	39
13	100	143	195	37	29	22	26
33	142	16	28	28	26	28	30
149	27	26	17	29	30	29	27

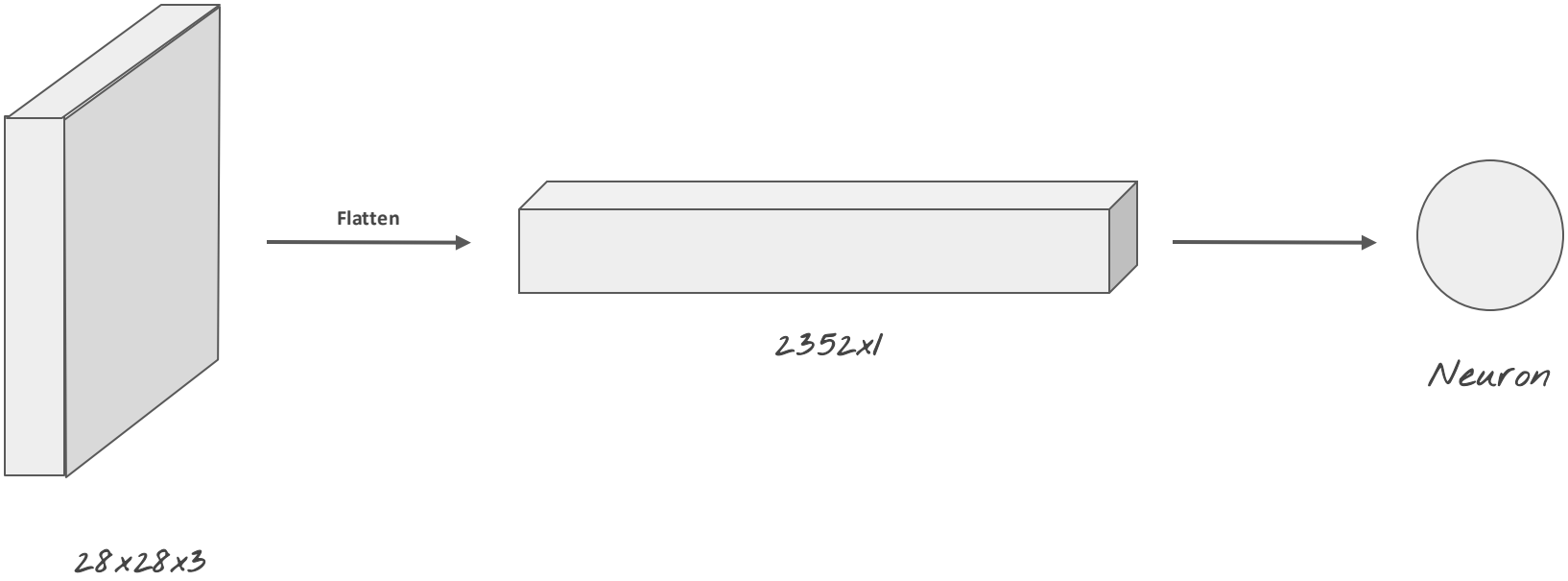
How Machine sees picture

32	27	33	12	18	29	37	27
29	18	28	16	27	18	29	30
2	33	66	155	160	23	32	28
32	27	99	180	192	86	199	100
47	90	180	190	170	30	21	39
13	100	143	195	37	29	22	26
33	142	16	28	28	26	28	30
149	27	26	17	29	30	29	27

Neighbours of '180' ?

32	27	33	12	18	29	37	27
29	18	28	16	27	18	29	30
2	33	66	155	160	23	32	28
32	27	99	180	192	86	199	100
47	90	180	190	170	30	21	39
13	100	143	195	37	29	22	26
33	142	16	28	28	26	28	30
149	27	26	17	29	30	29	27

FC Layer



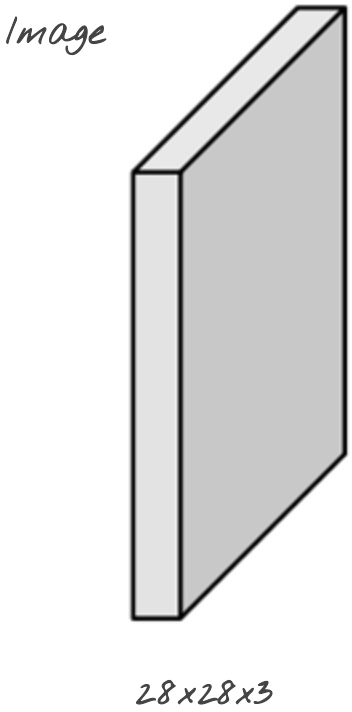
FC Layer can work only with Vectors as input

...	...	...	...	27	99	180	...	...	...	...	90	180	190	...	...	...	...	100	143	195	...	...	...

When reshaping the input for FC layer, Spatial information is lost

Convolutional Layer

Input Shape

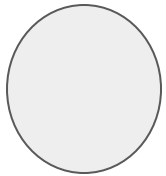


No need to flatten the image
i.e. keep Spatial info

Neuron Input Size

28x28x3

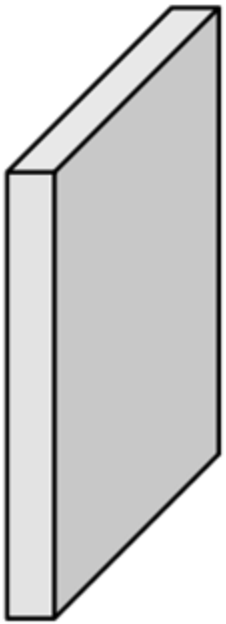
Neuron does not get whole image to look at unlike FC Layer



Neuron

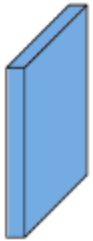
Using Filter

Image

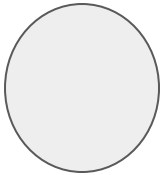


28x28x3

filter



5x5x3



Neuron

Neuron gets to look at only part of the image

Neuron Output

Image

32	27	33	12	18	29	37	27
29	18	28	16	27	18	29	30
2	33	66	155	160	23	32	28
32	27	99	180	192	86	199	100
47	90	180	190	170	30	21	39
13	100	143	195	37	29	22	26
33	142	16	28	28	26	28	30
149	27	26	17	29	30	29	27

8x8x1

filter

1	0	12
2	15	1
1	10	0

3x3x1

Neuron Output

Image

32	27	33	12	18	29	37	27
29	18	28	16	27	18	29	30
2	33	66	155	160	23	32	28
32	27	99	180	192	86	199	100
47	90	180	190	170	30	21	39
13	100	143	195	37	29	22	26
33	142	16	28	28	26	28	30
149	27	26	17	29	30	29	27

8x8x1

filter

1	0	12
2	15	1
1	10	0

3x3x1

Output is a dot product

= (32*1) + (27*0) + (33*12) + (29*2) +(18*15) + (28*1) + (2*1) + (33*10) + (66*0)

= 1116

Next Neuron Output

Image

32	27	33	12	18	29	37	27
29	18	28	16	27	18	29	30
2	33	66	155	160	23	32	28
32	27	99	180	192	86	199	100
47	90	180	190	170	30	21	39
13	100	143	195	37	29	22	26
33	142	16	28	28	26	28	30
149	27	26	17	29	30	29	27

8x8x1

filter

1	0	12
2	15	1
1	10	0

3x3x1

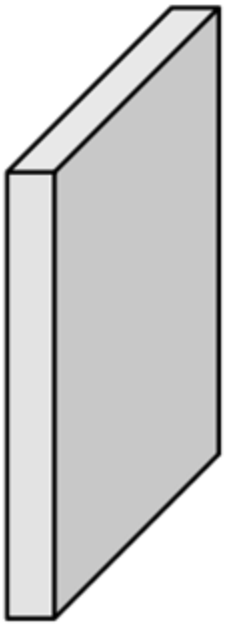
Filter weight remain same for different parts of the images

dot product

= (27*1) + (33*0) + (12*12) + (18*2) +(28*15) + (16*1) + (33*1) + (66*10) + (155*0)

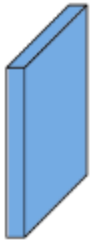
= 1336

Image

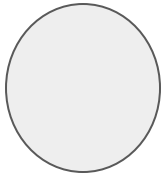


28x28x3

filter



5x5x3

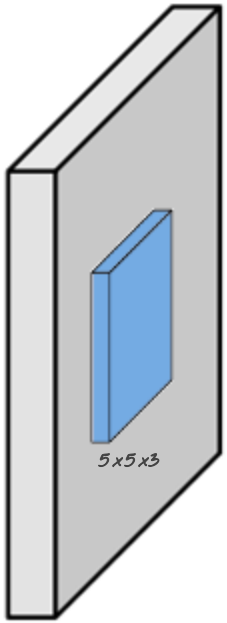


Neuron

How many Weights to learn?

75

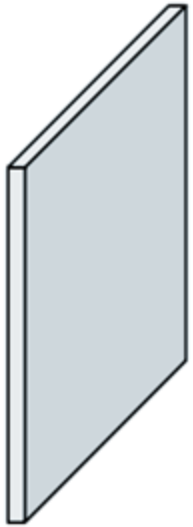
Input Image



28x28x3

'Slide' Filter over entire image

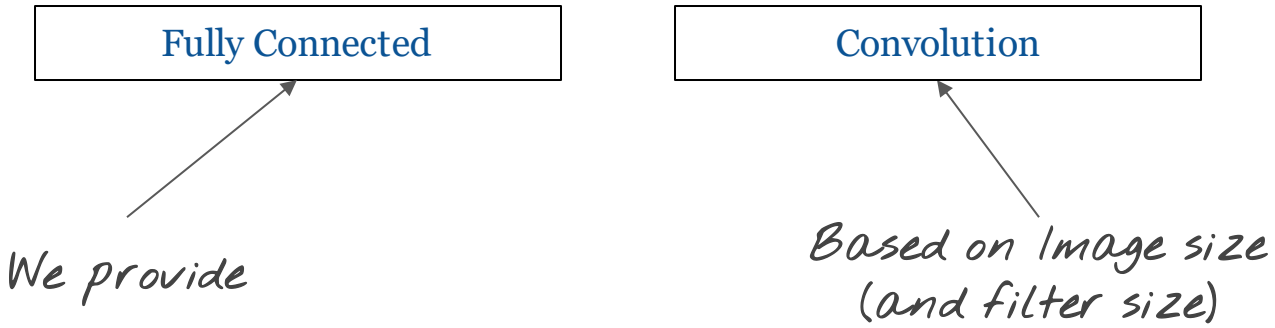
Output
(Activation Map)



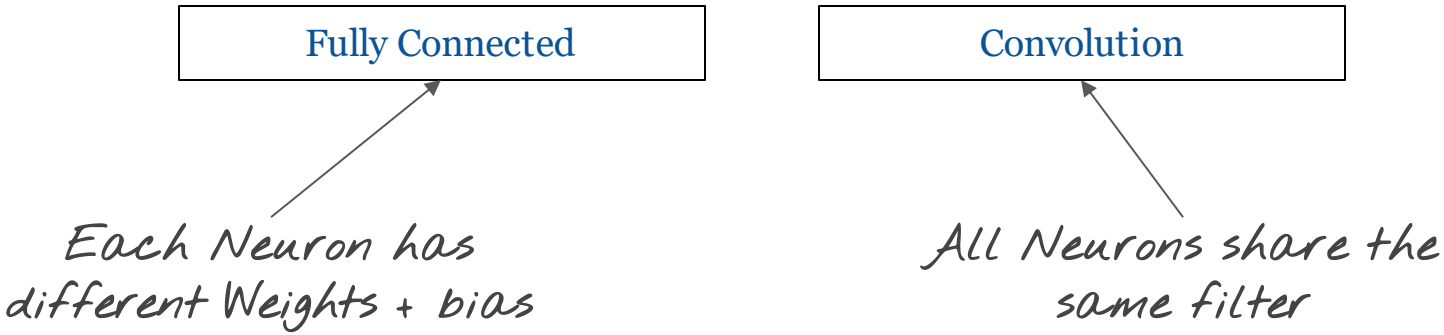
Size?

24x24

Who decides on number of Neurons?



Weights for each Neuron?



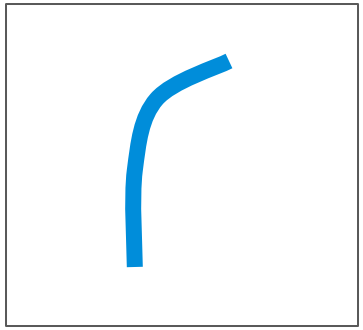
What is the goal of Convolutional Layer?

Build a good filter which identifies a feature in the input

Visualizing a filter

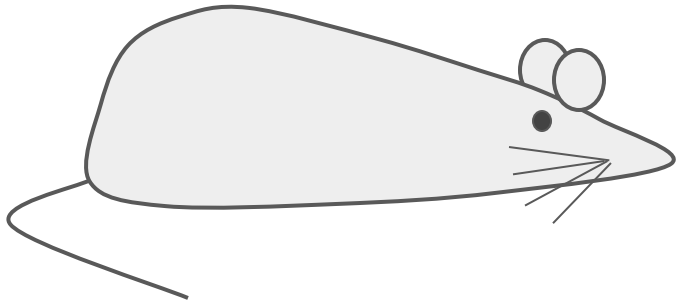
0	0	0	50	0
0	0	50	0	0
0	0	50	0	0
0	0	50	0	0
0	0	0	0	0

5x5 filter



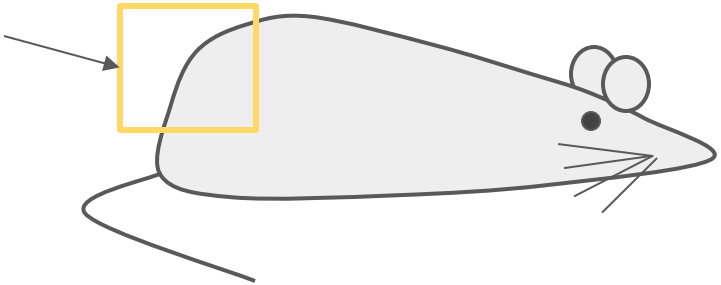
Filter visual

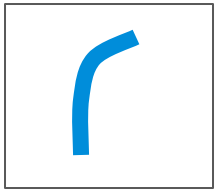
Input Image



Let's apply filter on this image

Image part where filter is being applied





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

Image Part as numbers

X

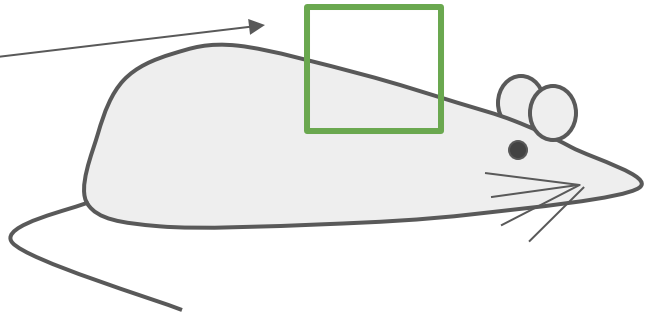
0	0	0	50	0
0	0	50	0	0
0	0	50	0	0
0	0	50	0	0
0	0	0	0	0

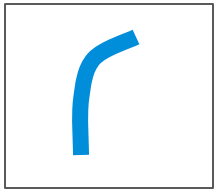
filter

= (30*50) + (30*50) + (30*50) + (30*50)

6000

Image part where filter is being applied





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

Part of Image

X

0	0	0	50	0
0	0	50	0	0
0	0	50	0	0
0	0	50	0	0
0	0	0	0	0

filter

=(30*50)

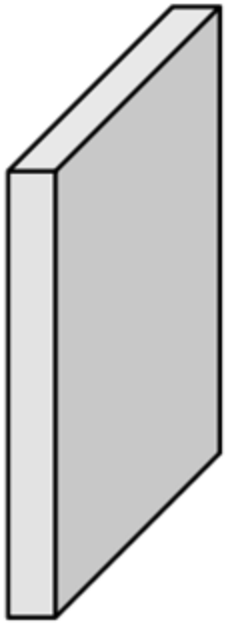
1500

What does large output signify?

Presence of the ‘feature’ represented by the filter

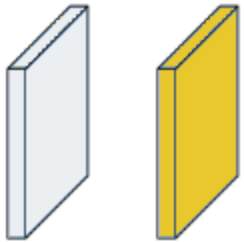
Building
Multiple filters

Input Image



28x28x3

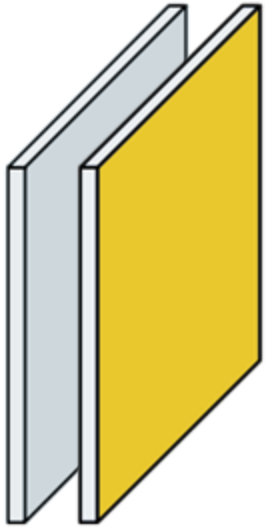
Filters



5x5x3 5x5x3

2 Filters -> 2 Activation Maps

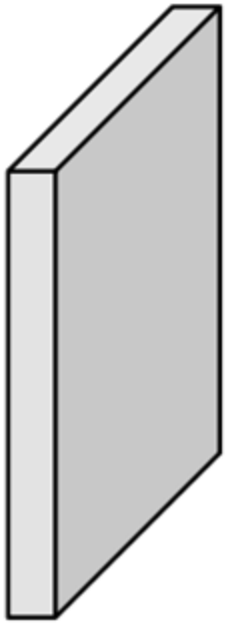
Output



Size?

24x24x2

Input Image

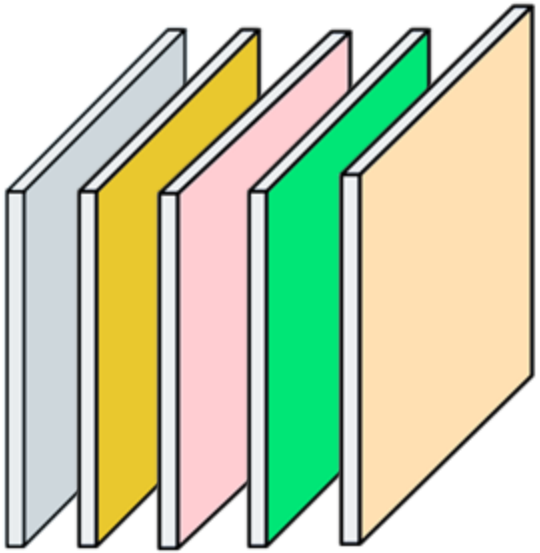


28x28x3

5 Filters

5x5x3

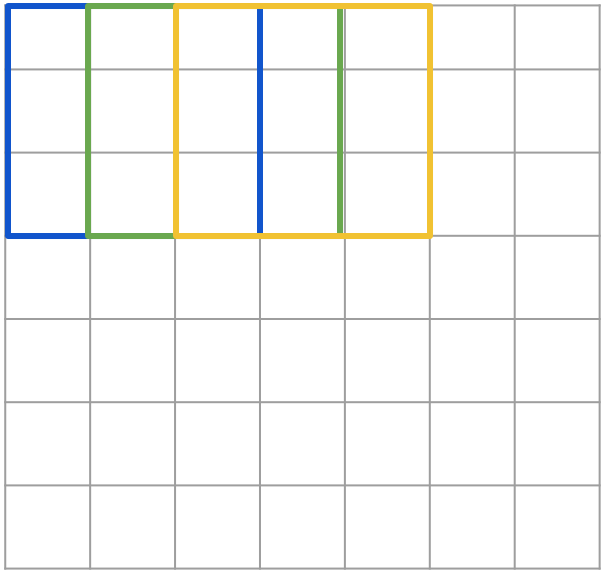
Output



Size

24x24x5

Filter Stride



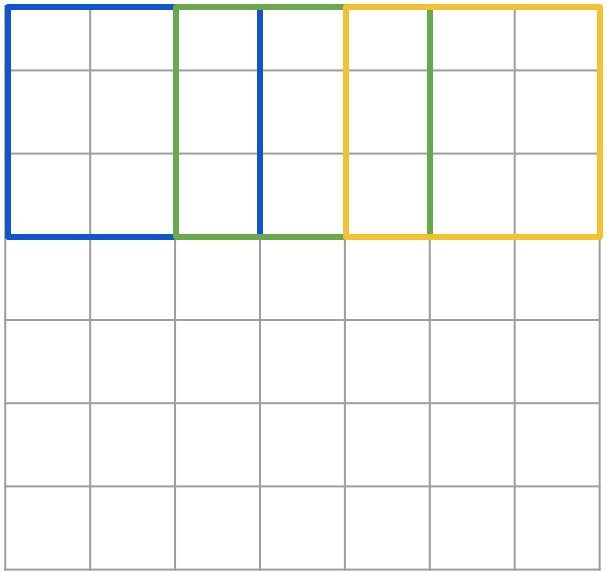
7x7 Image

Filter - 3x3

Stride = 1

Output : 5x5

Indicates how many step(s) to move when sliding filter



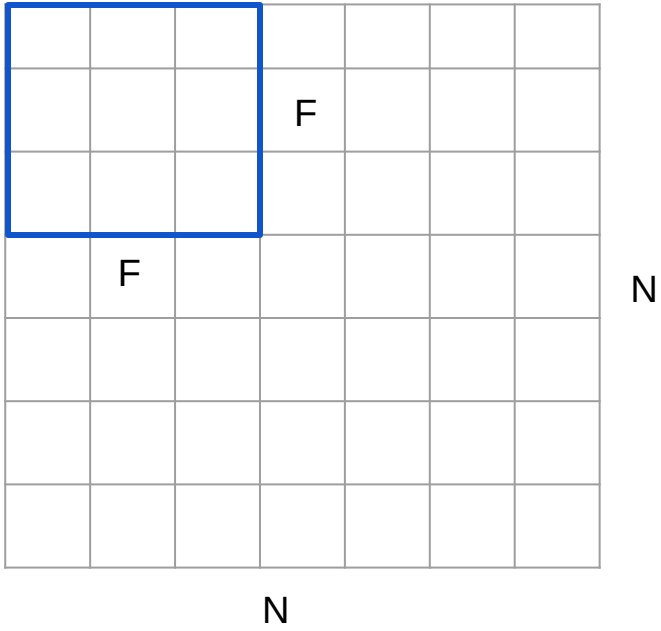
7x7 Image

Filter - 3x3

Stride = 2

Output : 3x3

Stride = S

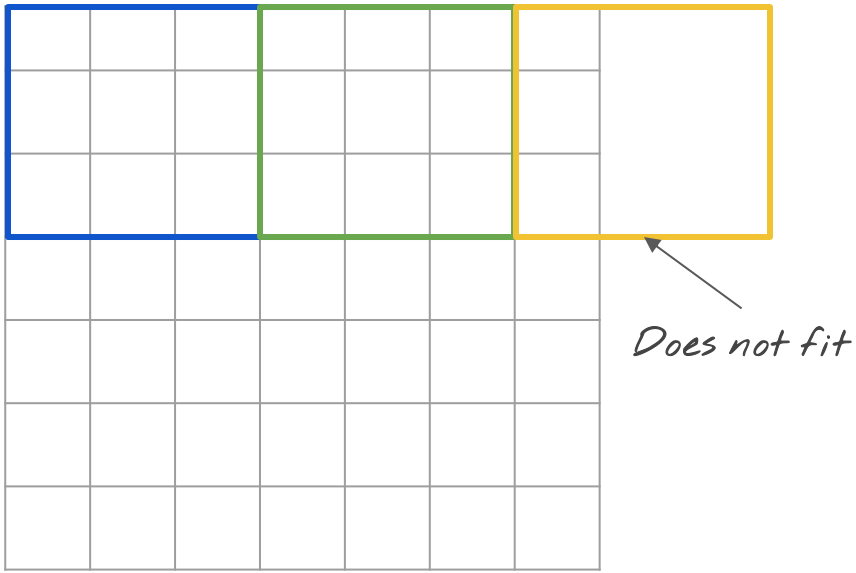


Output :
$(N - F)/S + 1$

N = 7, F = 3

S = 1 then Output is 5x5

S = 2 then Output is 3x3



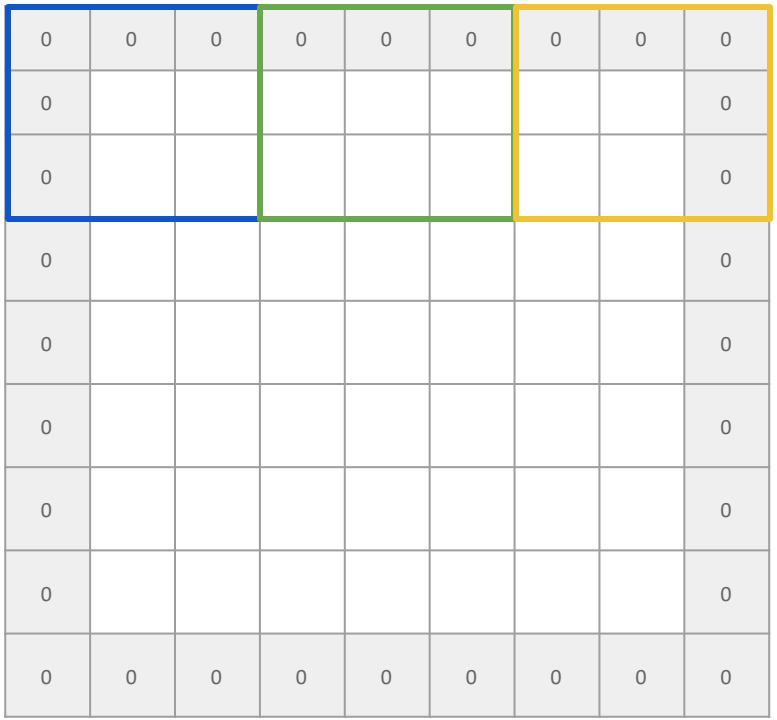
Filter - 3x3

Stride = 3

$(7-3)/3 + 1 = \mathbf{2.33}$

How to solve this problem?

Padding



7x7 Image

Filter - 3x3

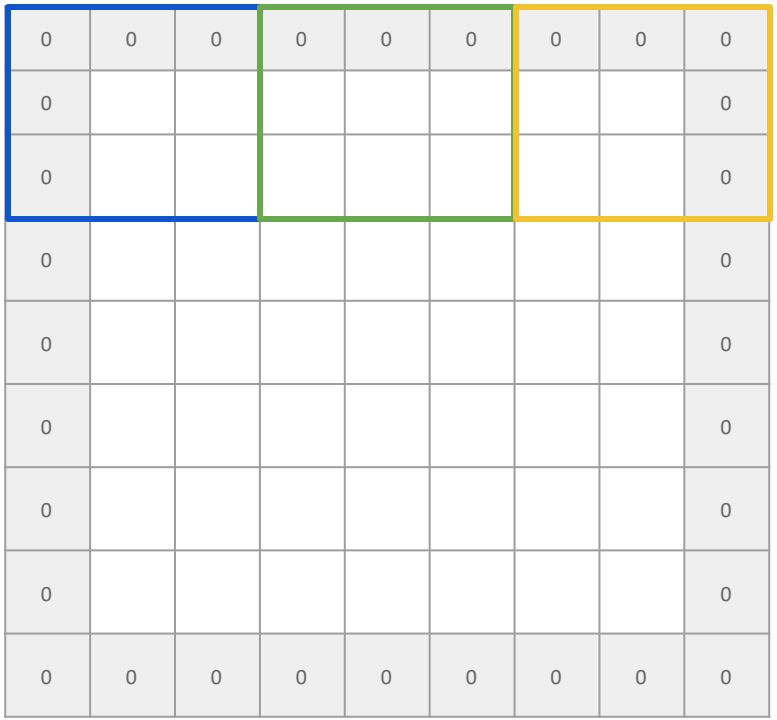
Stride = 3

Padding = 1

How much padding to add?

$(N - F + 2P)/S =$ Whole Number

Output Size with Padding



7x7 Image

Filter - 3x3

Stride = 3

Padding = 1

New Output formula

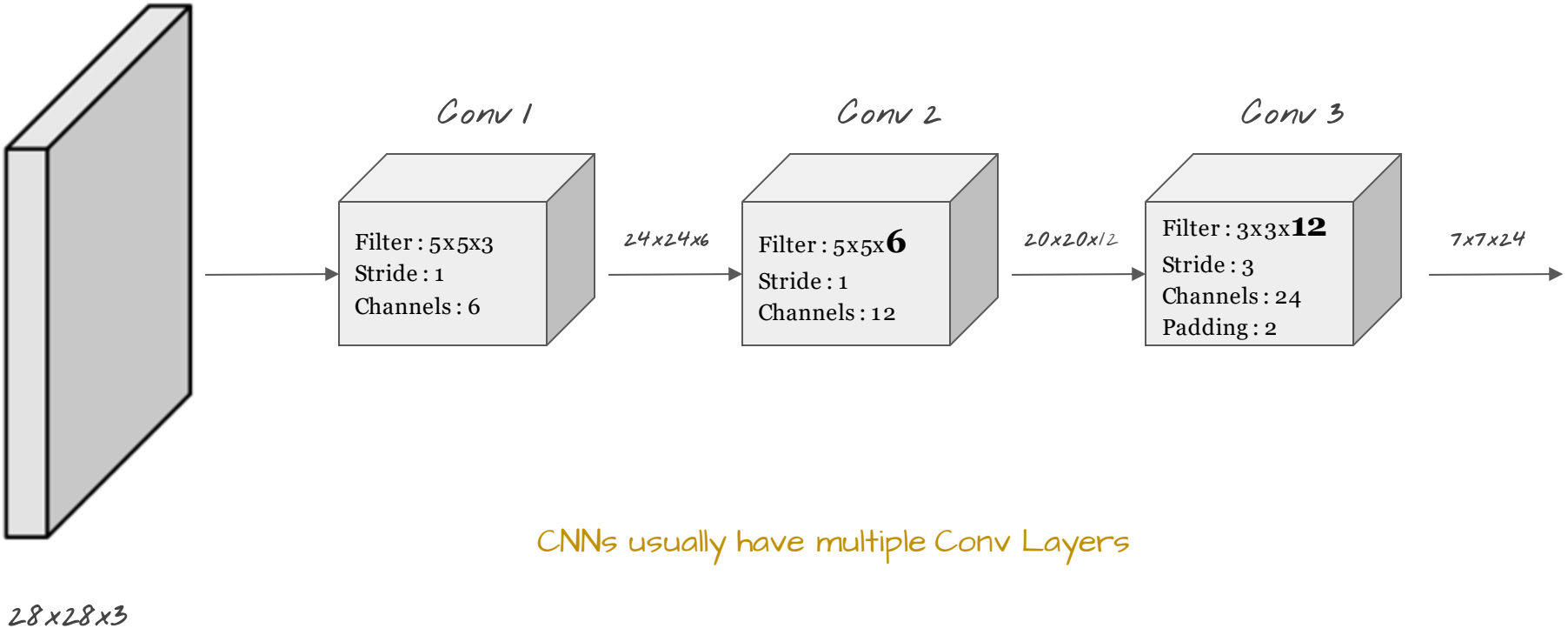
$(N + 2P - F)/S + 1$

Recap - Conv Layer

- Uses Spatial Information
- Builds filters
- Shares filters among Neurons
- Hyperparameters
  - Filter Size
  - Stride
  - Padding

Convolutional Neural Network (CNN)

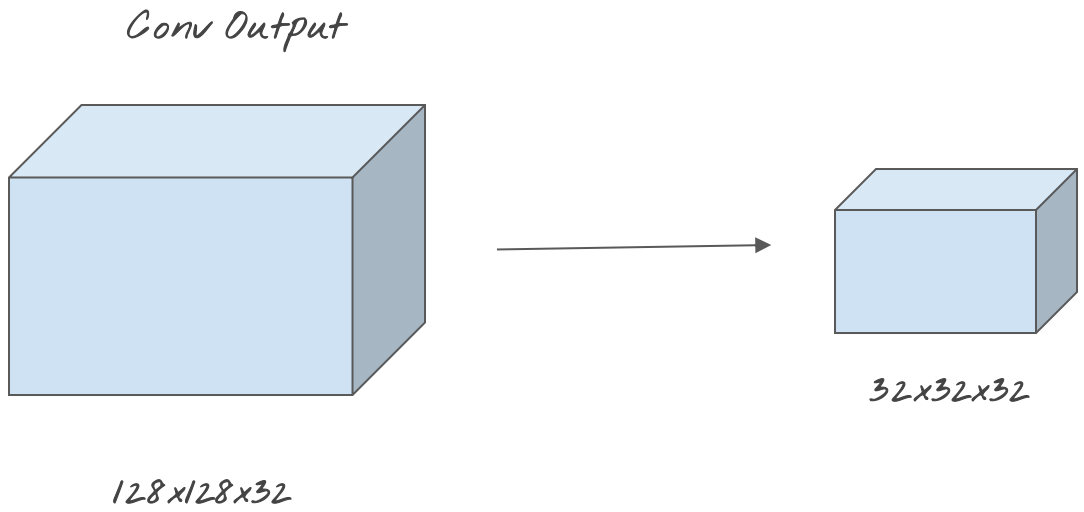
Using Convolution Layer



CNNs usually have multiple Conv Layers

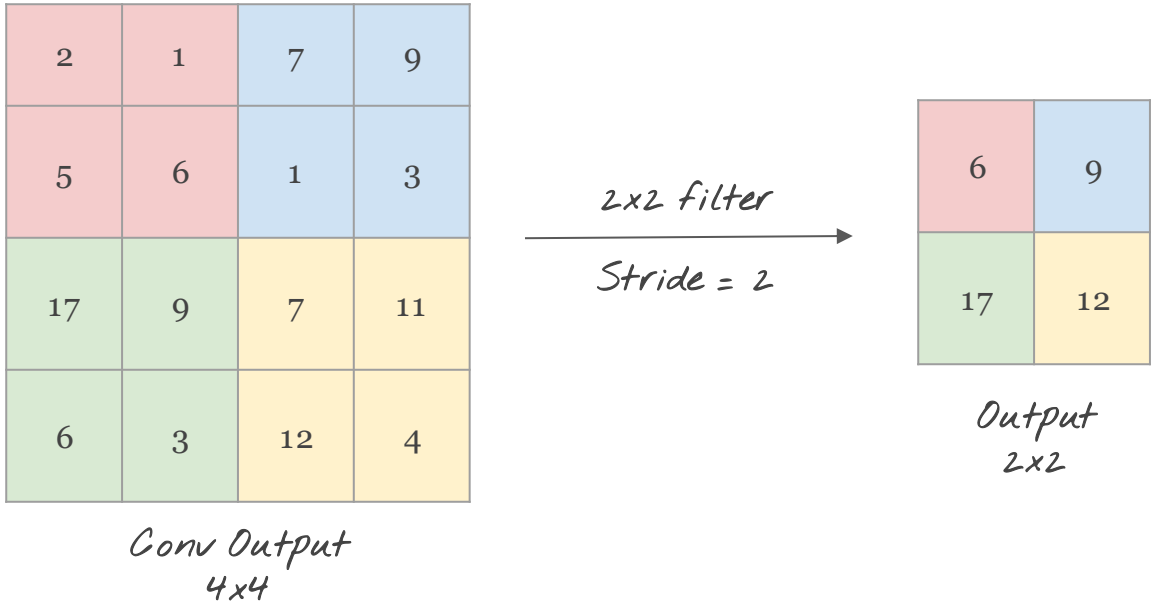
Too many parameters?

Pooling Layer (Down Sampling)

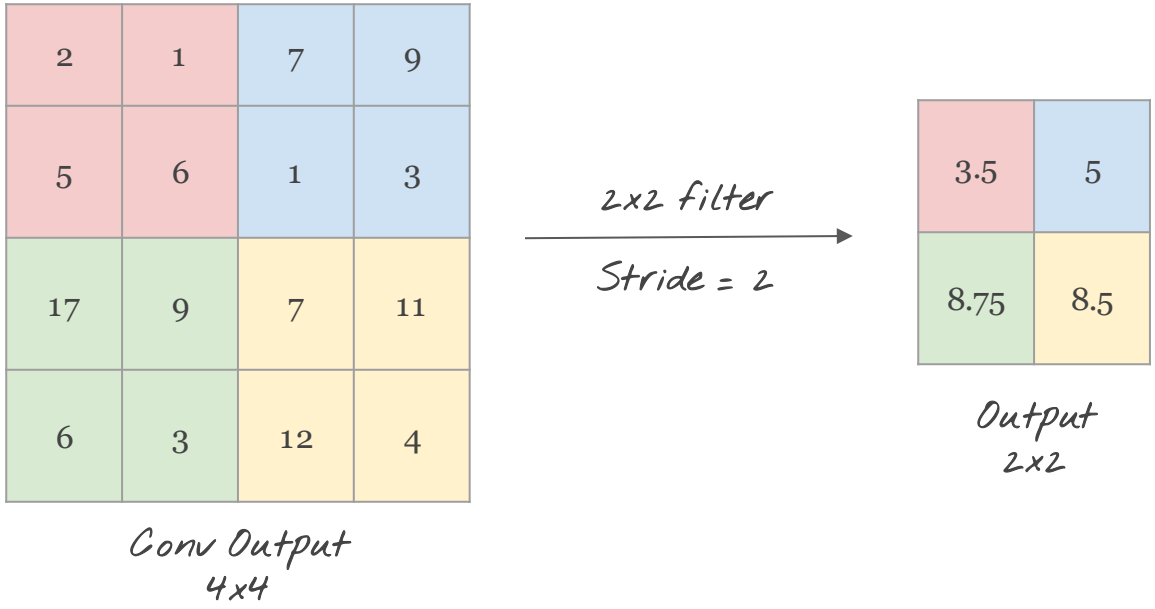


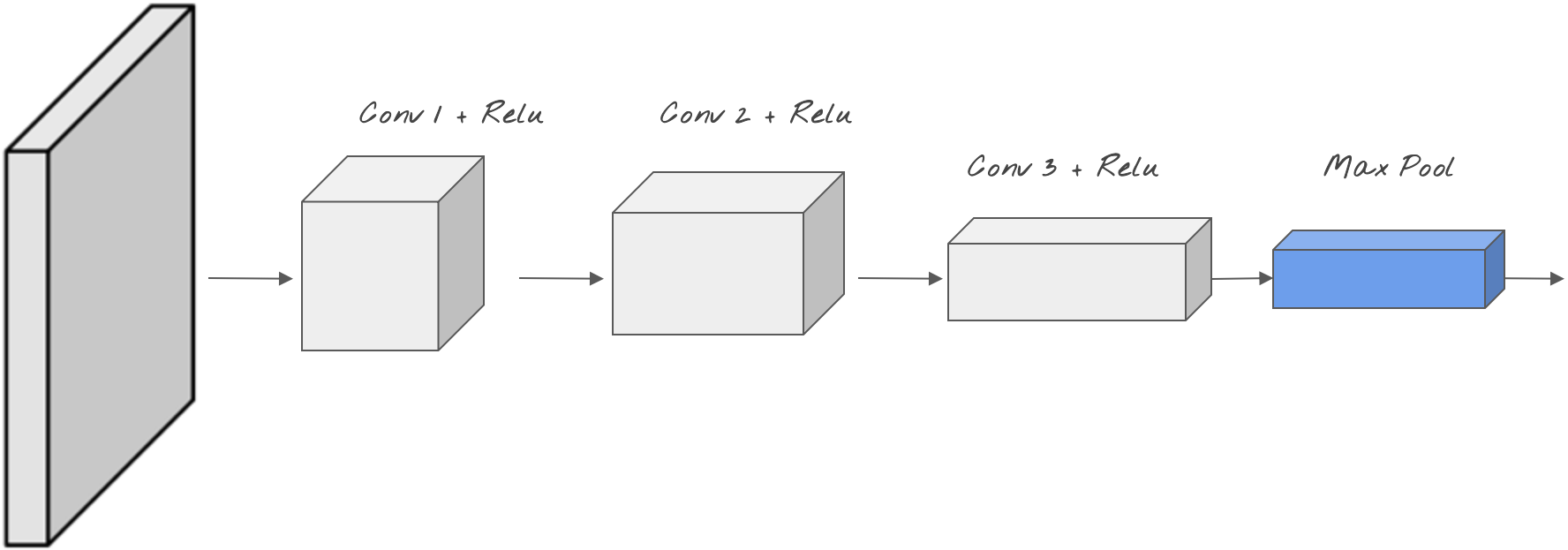
- Reduces size
- Works with each Activation Map

Max Pooling

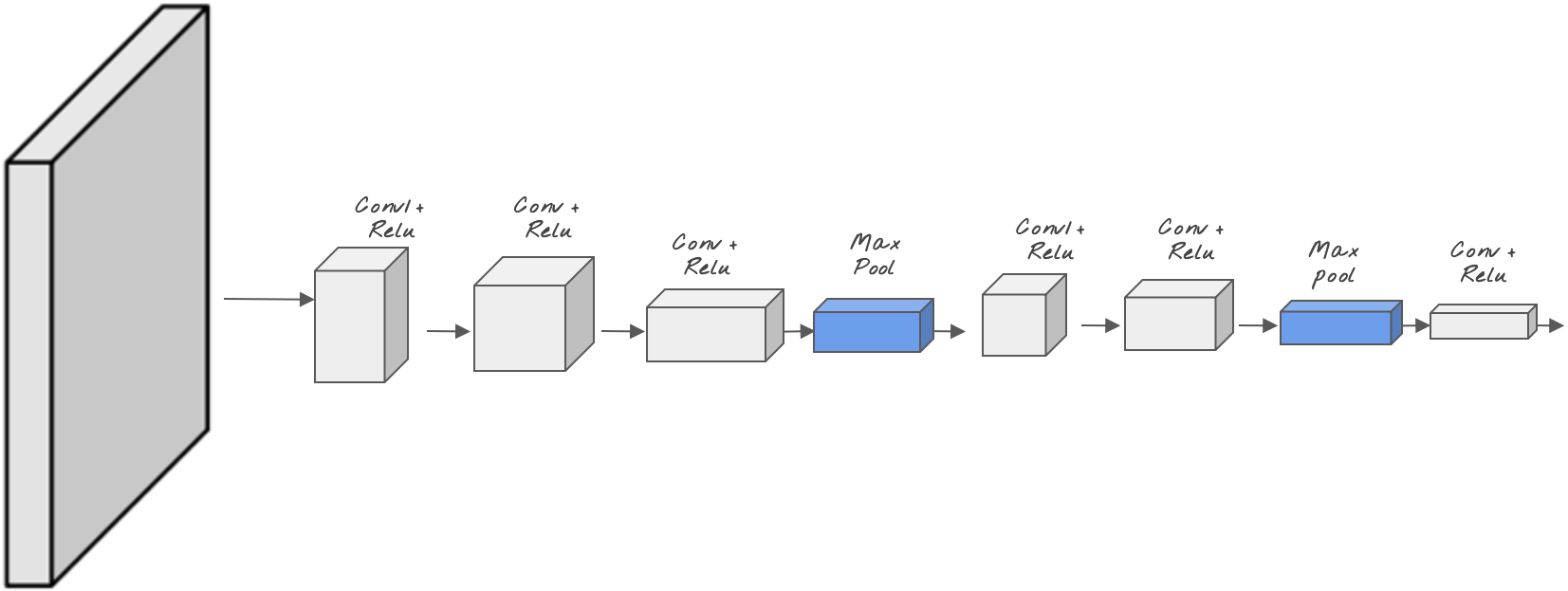

2
1
7
9
5
6
1
3
17
9
7
11
6
3
12
4
Conv Output
4x4
2x2 filter
Stride = 2
6
9
17
12
Output
2x2

Average Pooling

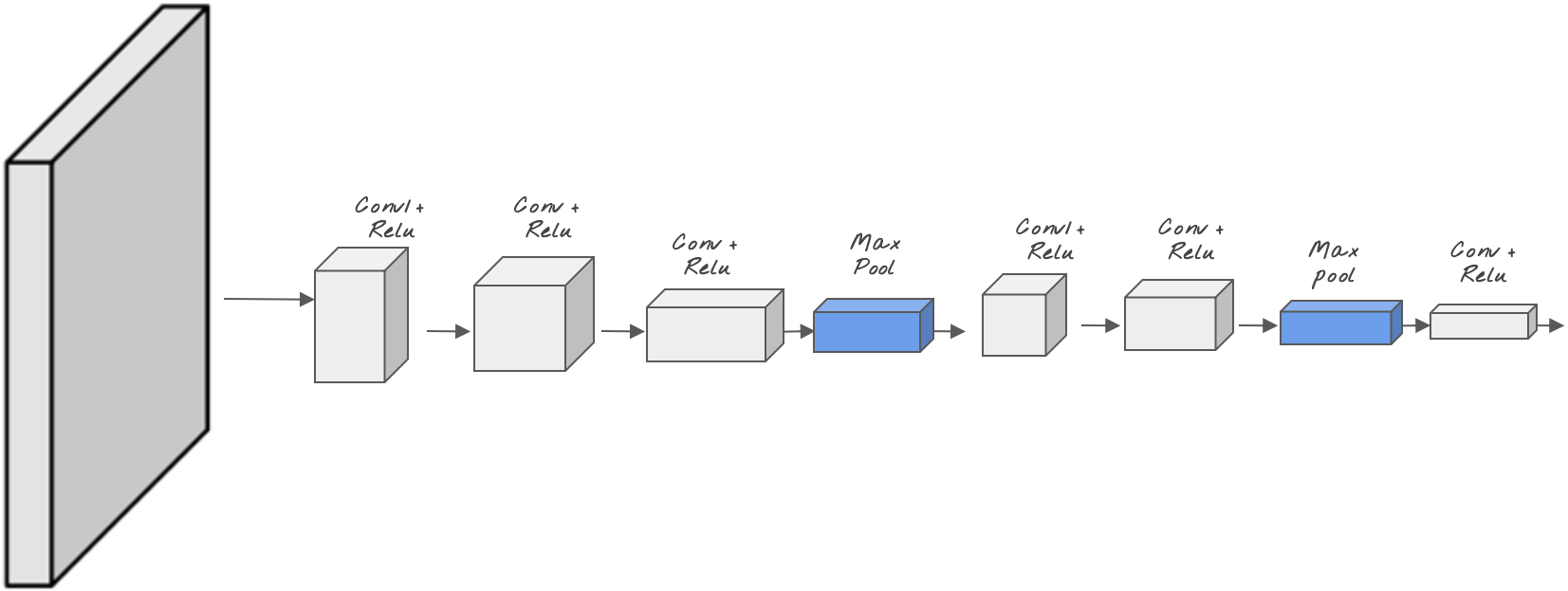




In CNNs, usually depth keeps increasing and ...



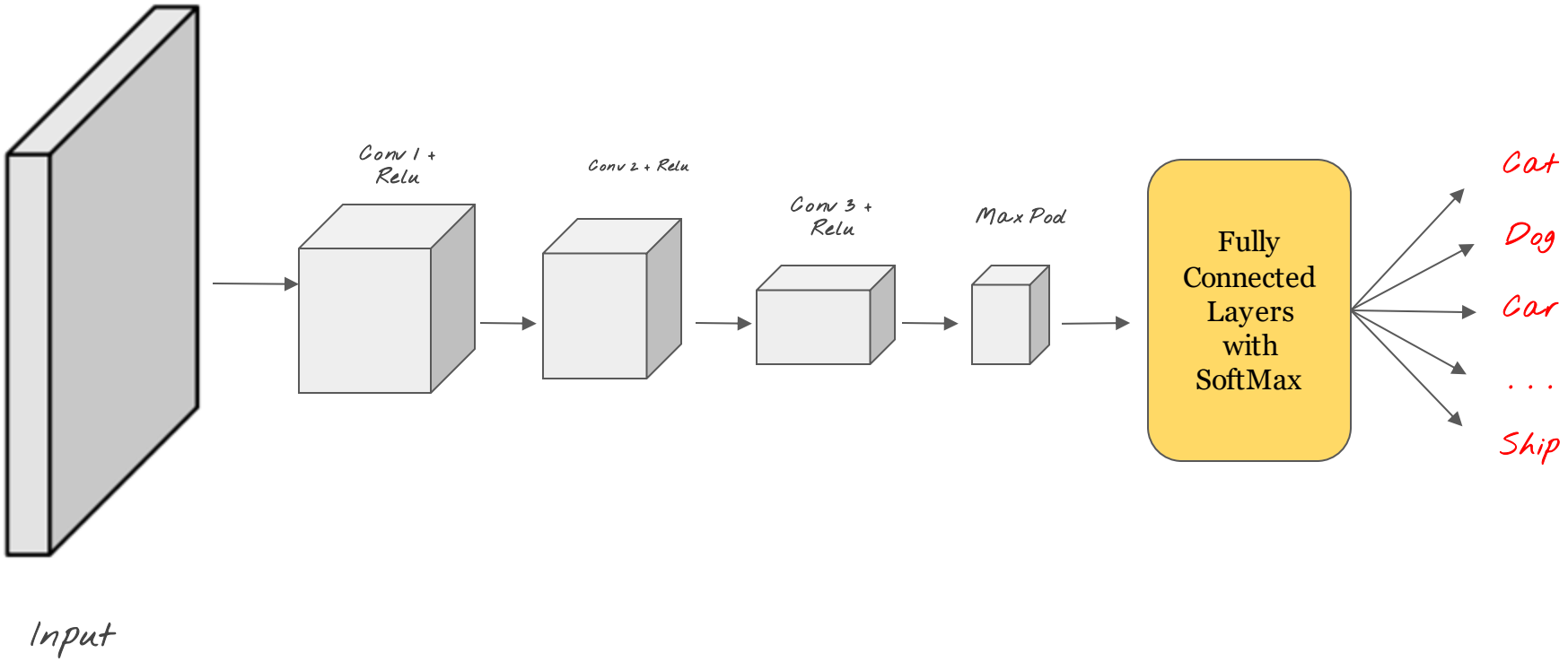
Lots of convolutions and Pooling layers



What does the output represent?

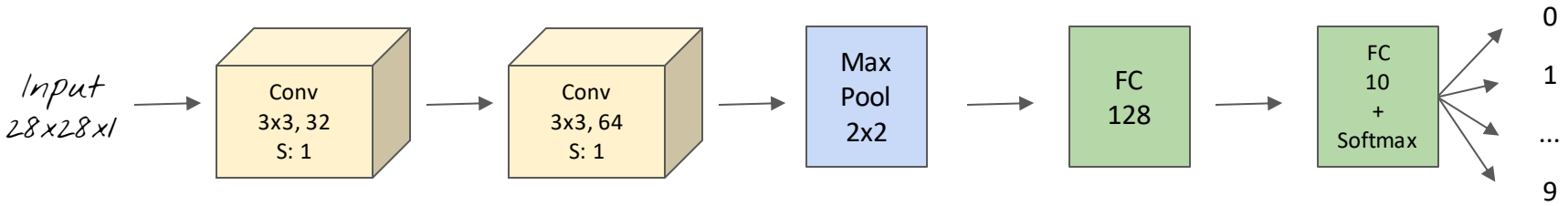
How do I classify things?

CNN


Conv 1 + Relu
Conv 2 + Relu
Conv 3 + Relu
Max Pool
Fully Connected Layers with SoftMax
Cat
Dog
Car
. . .
Ship
Input

Applying CNN

MNIST Classification



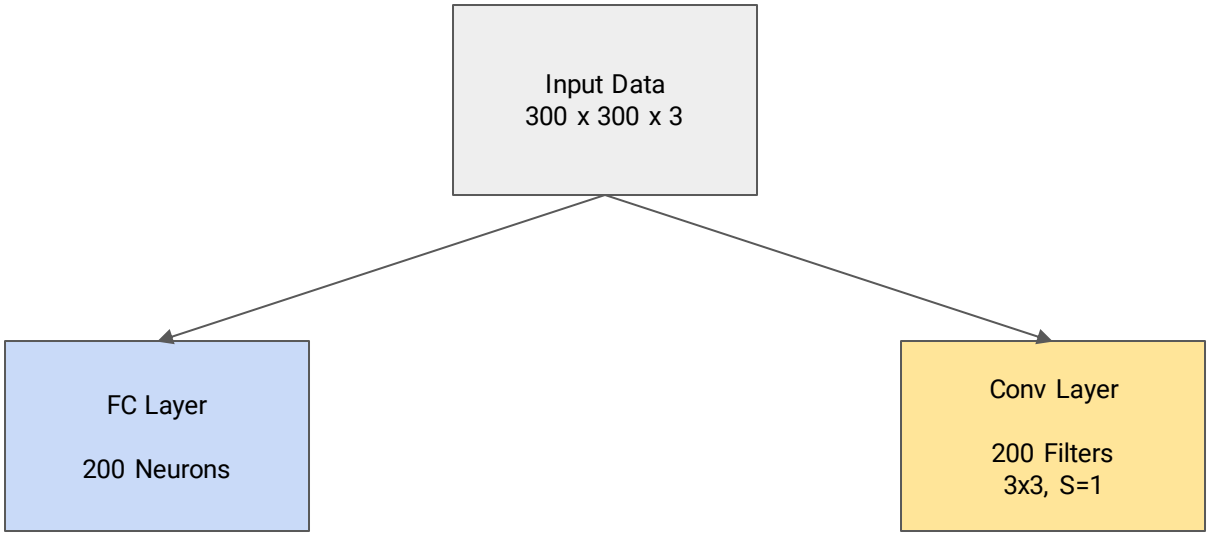
Exercise: CIFAR-10 Classification

https://www.cs.toronto.edu/~kriz/cifar.html

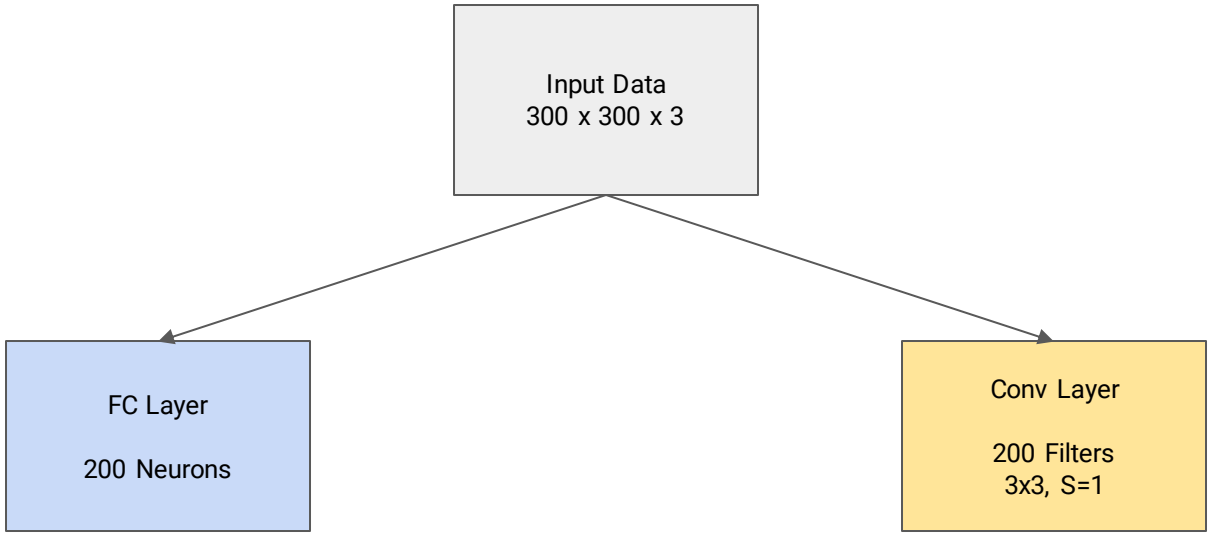
CNN and Biological Connection

- Takes inspiration from the Visual cortex.
- Individual neuronal cells in the brain responded (or fired) only in the presence of edges of a certain orientation.
- For example, some neurons fired when exposed to vertical edges and some when shown horizontal or diagonal edges.

Fully Connected	Convolutional
Works with Vector or flattened data	Can work with Multi-dimensional Data
Does not use Spatial information to learn features	Uses Spatial information using filters
Model Size is bigger	Reduced Model Size
Less Computation	More Computation



How many Weights?
How many calculations?



Weights :
200 * 300 * 300 * 3 = **54M**

Calculations:
200 * (270K Multi + 270K add) = **108M**

Weights :
200 * 3 * 3 * 3 = **5.4KB**

Calculations:
200 * (298 * 298) * (27 Multi + 27 add) = **959M**

Image Size	N x N x d
Filter Size	F x F x d
Number of Filters	M
Padding	P
Stride	S
Output size (O x O)	O = (N - F + 2P)/S + 1
Number of Weights in One Filter	F x F x d
Number of Weights for all Filters in a Conv Layer	M x F x F x d
Number of Calculations per Filter (exc. Activation)	O x O x (FxFxd Mult + FxFxd Add) = O x O x **2** x F x F x d
Number of Calculations per Conv Layer	M x O x O x 2 x F x F x d