Fundamentals of Computer Vision

Dr.Venkataramana Veeramsetty
NVIDIA DLI Instructor & Head of CAIDL
Assistant Professor
Dept. of EEE
SR Engineering College

December 17, 2020

Computer Vision Problems

- Image Classification
  - You might take as input say a 64 by 64 image and try to figure out, is that a cat?
- Object Detection
  - If you're building a self-driving car, maybe you don't just need to figure out that there are other cars in this image. But instead, you need to figure out the position of the other cars in this picture, so that your car can avoid them.
- Neural style transfer
  - Paint one image (Content Image) In another image style

Why Convolution Operation Required

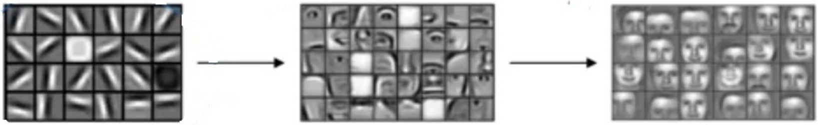
- Consider 64 X 64 gray scale image, number of features in input matrix are 4096
- Consider 64 X 64 RGB image, number of features in input matrix are 12288
- Consider 1000 X 1000 RGB image, number of features in input matrix are 30,00,000
- Consier FCN, with first hidden layer has 1000 hidden units. then size of the weight matrix is 30,00,000 x 1000 size.

Problems and Approach

- It is difficult to get enough data to prevent neural network form over fitting
- Need more computation time
- Need more memory

To avoid these problem, use convolution operation which is fundamental building block for convolutional Neural Network (CNN)

Vertical Edge Detection



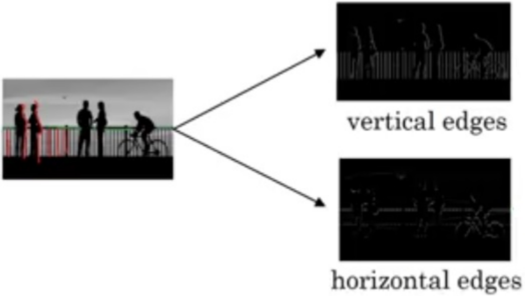


Table 1 : Gray Scale Image

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

Email : dr.vvr.research@gmail.com
(Dr. Venkataramana Veeramsetty)

Table 2 : Filter for Vertical Edge Detection

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

Table 3 : Convolution Operation

3(1)	0(0)	1(-1)	2	7	4
1(1)	5(0)	8(-1)	9	3	1
2(1)	7(0)	2(-1)	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

Table 4 : Output

-5			

Email : dr.vvr.research@gmail.com
(Dr. Venkataramana Veeramsetty)

Email : dr.vvr.research@gmail.com
(Dr. Venkataramana Veeramsetty)

Table 5 : Output Image after Convolution

-5	-4	0	8
-10	-2	2	3
0	-2	-4	-7
-3	-2	-3	-16

- Python: conv_forward
- Tensorflow: tf.nn.conv2d
- keras: Conv2D

Email :
(Dr. Ve

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

om
ty)

$$\begin{bmatrix} 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 0 & -30 & -30 & 0 \\ 0 & -30 & -30 & 0 \\ 0 & -30 & -30 & 0 \\ 0 & -30 & -30 & 0 \end{bmatrix}$$

Email : dr.vvr.research@gmail.com
(Dr. Ven tty)

Multiple Edge Detector

Table 6 : Vertical Edge Filter

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

Table 7 : Horizontal Edge Filter

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

$$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \\ 0 & 0 & 0 & 10 & 10 & 10 \end{bmatrix} * \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 30 & 10 & -10 & -30 \\ 30 & 10 & -10 & -30 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Sobel filter

Table 8 : Sobel filter for Vertical Edge Detection

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

Table 9 : Sobel filter for Horizontal Edge Detection

1	2	1
0	0	0
-1	2	-1

Scharr filter

Table 10 : Scharr filter for Vertical Edge Detection

3	0	-3
10	0	-10
3	0	-3

Table 11 : Scharr filter for Horizontal Edge Detection

3	10	3
0	0	0
-3	-10	-3

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

=

n=number of pixels in input image
f=number of pixels in filter
o=number of pixels in output image
o=n-f+1

For example

f input image size is 6 X 6 and filter size is 3 X 3
means n=6, f=3

Number of parameters in one layer

If you have 10 filters that are 3 x 3 x 3 in one layer of a neural network, how many parameters does that layer have?

Email : dr.vvr.research@gmail.com
(Dr. Venkataramana Veeramsetty)

Padding

- In order to build deep neural network one modification to the basic convolutional operation that needs to be use in **padding**

Limitations of standard convolution operation

- Can not detect edges or other feature without shrinking input image
- Throwing away a lot of the information near the edge of the image

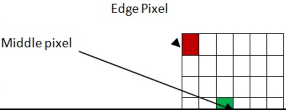


Table 12 : Padding one layer ($\mathbf{p=1}$)

0	0	0	0	0	0	0	0
0	3	0	1	2	7	4	0
0	1	5	8	9	3	1	0
0	2	7	2	5	1	3	0
0	0	1	3	1	7	8	0
0	4	2	1	6	2	8	0
0	2	4	5	2	3	9	0
0	0	0	0	0	0	0	0

Email : dr.vvr.research@gmail.com
(Dr. Venkataramana Veeramsetty)

Finding output image size

n=number of pixels in input image
f=number of pixels in filter
o=number of pixels in output image
p=number of layers in padding $o=n+2p-f+1$
For example
if input image size is 6 X 6, filter size is 3 X 3 and padding with one layer
means $n=6$, $f=3$, $p=1$
Then output image size is $(n+2p-f+1)$ X $(n+2p-f+1) = (6+2*1-3+1)$ X $(6+2*1-3+1) = 6$ X 6

Valid and Same Convolutions

- **Valid Convolutions**: No padding

n X n * f X f = n-f+1 X n-f+1

- **Same Convolutions** : Pad such that size of input and output size must be same

n X n * f X f = n+2*p-f+1 X n+2*p-f+1

In order to keep input image size same as output image size, padding parameter p is designed based filter size (f)

n+2*p-f+1 = n $\Rightarrow$ **p=** $\frac{f-1}{2}$

f value usually odd

Striding Convolutions

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

=

91		

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

=

91	100	

Striding parameter s=2

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

=

91	100	83

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

=

91	100	83
69		

Striding parameter s=2

Email : dr.vvr.research@gmail.com
(Dr. Venkataramana Veeramsetty)

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

=

91	100	83
69	91	

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

=

91	100	83
69	91	127

Striding parameter s=2

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

=

91	100	83
69	91	127
44		

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

=

91	100	83
69	91	127
44	72	

Striding parameter s=2

Email : dr.vvr.research@gmail.com
(Dr. Venkataramana Veeramsetty)

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

3	4	4
1	0	2
-1	0	3

=

91	100	83
69	91	127
44	72	74

Striding parameter s=2 and output size 3 X 3
Perform convolution only if all pixels in filter must overlap on image

Email : dr.vvr.research@gmail.com
(Dr. Venkataramana Veeramsetty)

Finding output image size

n=number of pixels in input image

f=number of pixels in filter

o=number of pixels in output image

p=number of layers in padding

s=striding parameter

$o = \frac{n+2p-f}{s} + 1$ if that fraction is integer

$o = \text{floor}(\frac{n+2p-f}{s} + 1)$ if that fraction is not integer

For example
if input image size is 7 X 7, filter size is 3 X 3,
padding with no layer and striding parameter $s=2$
means $n=6$, $f=3$, $p=0$, $s=2$
Then output image size is $(n+2p-f+1)$ X
$(n+2p-f+1) = (7+2*0-3)/2+1)$ X
$(7+2*0-3)/2+1) = 3$ X 3

Email : dr.vvr.research@gmail.com
(Dr. Venkataramana Veeramsetty)

Cross - Correlation

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3	4	8	3	8	9	7
7	8	3	6	6	3	4
4	2	1	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

*

Filter

3	4	4
1	0	2
-1	0	3

3	2	4
0	0	4
-1	1	3

=

101		

Flip filter and DO convolution operation

Convolution Over Volume

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

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

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

=

-15	-12	0	24
-30	-6	6	9
0	-6	-12	-21
-9	-6	-9	-48

$$O(1,1)=R(1,1)*fR(1,1)+G(1,1)*fG(1,1)+B(1,1)*fB(1,1)$$

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

3	0	1	2	7	4
1	5	8	9	3	1
2	7	2	5	1	3
0	1	3	1	7	8
4	2	1	6	2	8
2	4	5	2	3	9

*

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

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

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

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

1	1	1
0	0	0
-1	0	-1

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

=

-15	-12	0	24
-30	-6	6	9
0	-6	-12	-21
-9	-6	-9	-48

-21	-33	6	12
30	51	27	-9
12	15	-3	-21
-21	-18	3	6

Email : dr.vvr.research@gmail.com
(Dr. Venkataramana Veeramsetty)

n X n X n_c * f X f X n_c X n_f $= \frac{n+2p-f}{s} + 1$ X $\frac{n+2p-f}{s} + 1$ X n_f

$6 \text{ X } 6 \text{ X } 3 * 3 \text{ X } 3 \text{ X } 3 \text{ X } 2 = \frac{6+2*0-3}{1} + 1 \text{ X } \frac{6+2*0-3}{1} + 1 \text{ X } 2$

One Layer of Convolutional Neural Network

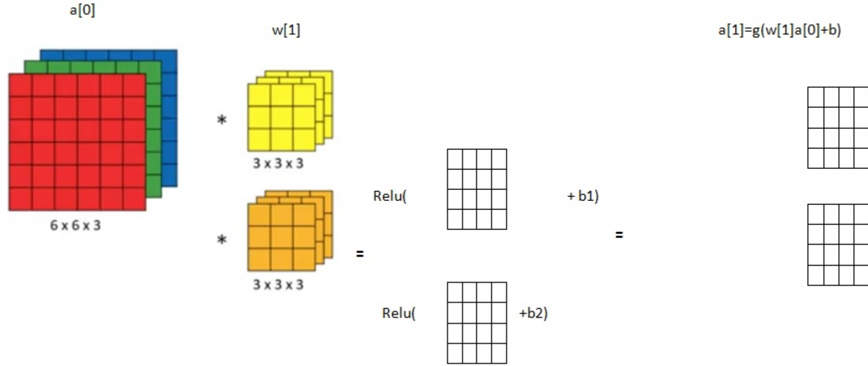
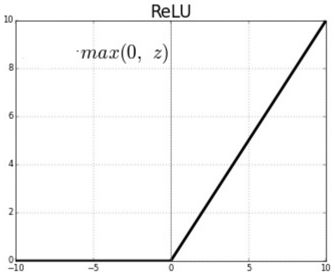


Image * convolution operation + nonlinear operation = output of layer 1

Relu Activation Function



output= max (0,z)

Whereas for Leaky Relu

output= max(0.1z,z)

f^l= Number of rows and columns in filter in layer 1

p^l= Padding size in layer 1

s^l= Striding size in layer 1

n_c=Number of channels of image

n_f^l=Number of Filters

Image size=$n_H^{l-1} X n_W^{l-1} X n_c$

Filter size= $f^l X f^l X n_c$

Output size= $n_H^l X n_W^l X n_c$

Where $n_H^l = n_W^l = \frac{n_H^{l-1} + 2p^l - f^l}{s^l} + 1$

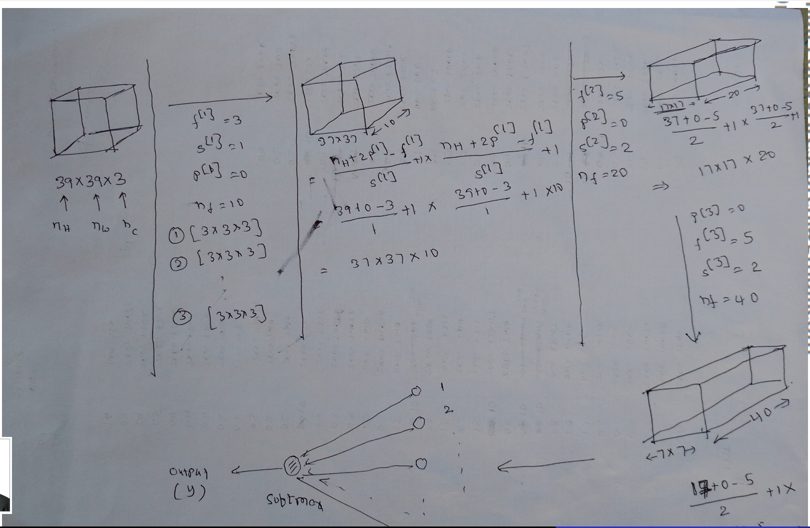
Number of activations $= n_H^l * n_W^l * n_c$

Weights= $f^l * f^l * n_c * n_f^l$

bias parameters $= n_f^l$ For one filter one bias parameter

Email : dr.vvr.research@gmail.com
(Dr. Venkataramana Veeramsetty)

Simple Convolutional Neural Network



Layers In CNN

- Convolution Layer
- Pooling Layer
- Fully Connected Layer

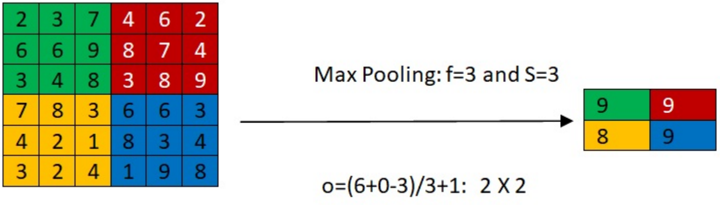
mail : dr.vvr.research@gmail.com
r. Venkataramana Veeramsetty)

Pooling

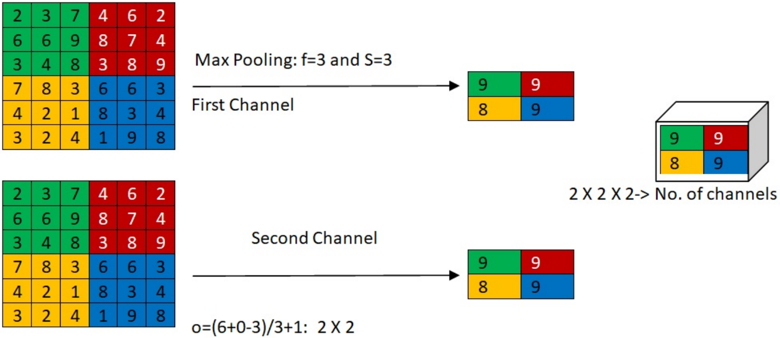
Other than convolution layers, ConvNet also uses Pooling layers. The main purposes of using this pooling layer in ConvNet are

- To reduce the size of representation
- To speedup the computation
- It detects some of the features more robust
- No parameters to learn

Max Pooling

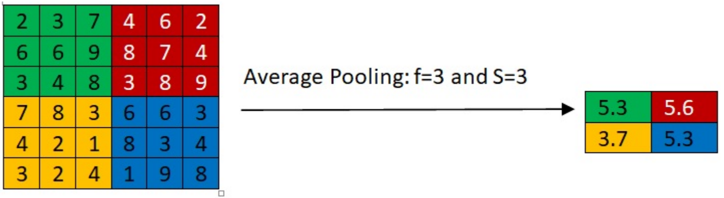


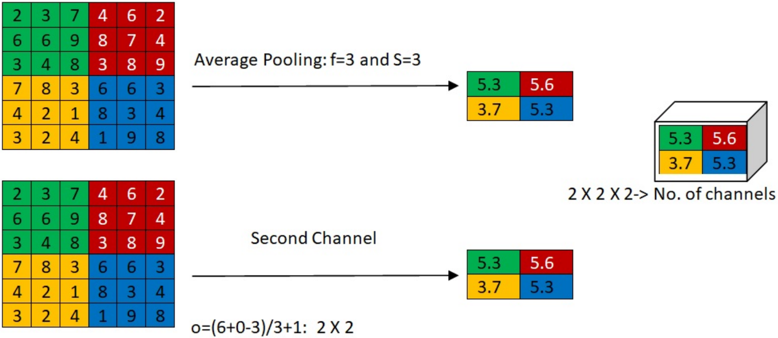
nail : dr.vvr.research@gmail.com
r. Venkataramana Veeramsetty)



Email : dr.vvr.research@gmail.com

Average Pooling





Email : dr.vvr.research@gmail.com

Hyper Parameters

f: Filter Size

s: Striding

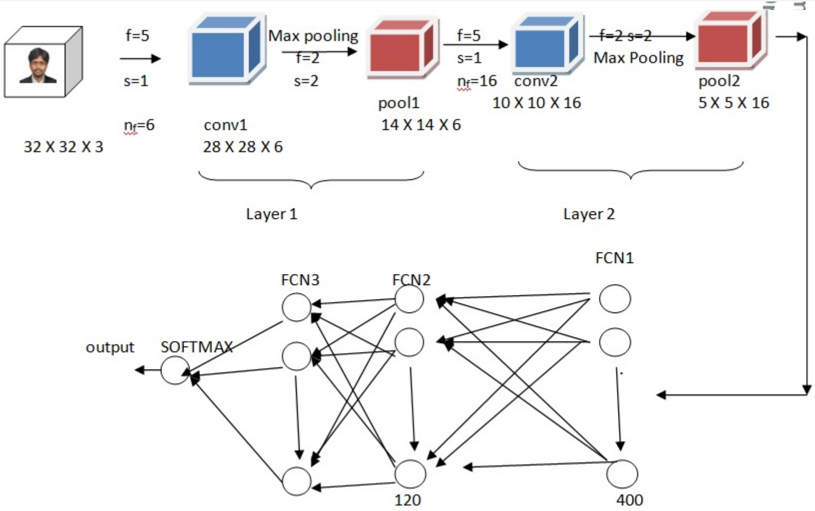
n: Size of image in previous layer

n_c: Number of channels

Size of Pooling Output: $\frac{n+2p-f}{s} + 1$ X $\frac{n+2p-f}{s} + 1$ X n_c

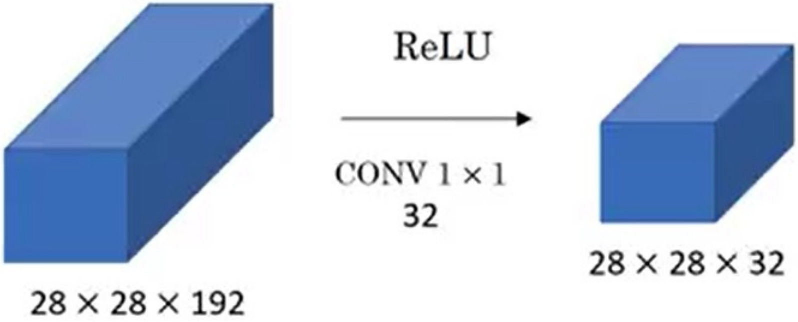
In pooling padding is usually zero. $(\mathbf{p=0})$

LENET

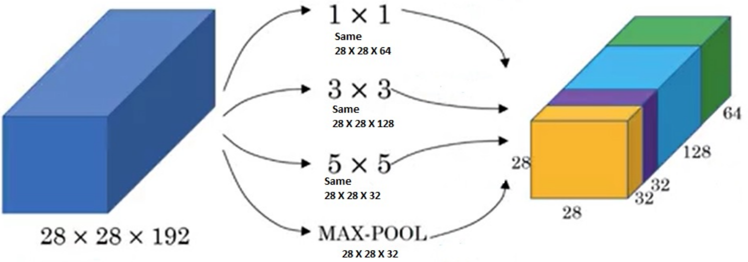

32 X 32 X 3
f=5
s=1
n_f=6
conv1
28 X 28 X 6
Max pooling
f=2
s=2
pool1
14 X 14 X 6
f=5
s=1
n_f=16
conv2
10 X 10 X 16
f=2 s=2
Max Pooling
pool2
5 X 5 X 16
Layer 1
Layer 2
FCN1
FCN2
FCN3
output
SOFTMAX
120
400

Number of parameters in convolution layer having filter size f, number of filters n_f and number of n_c is $(f*f*n_c+1)*n_f$

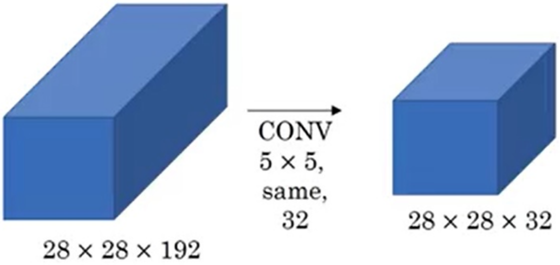
1X1 convolutions



Inception Network

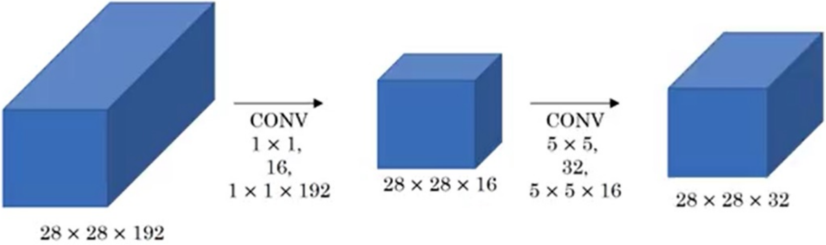


Computation Cost



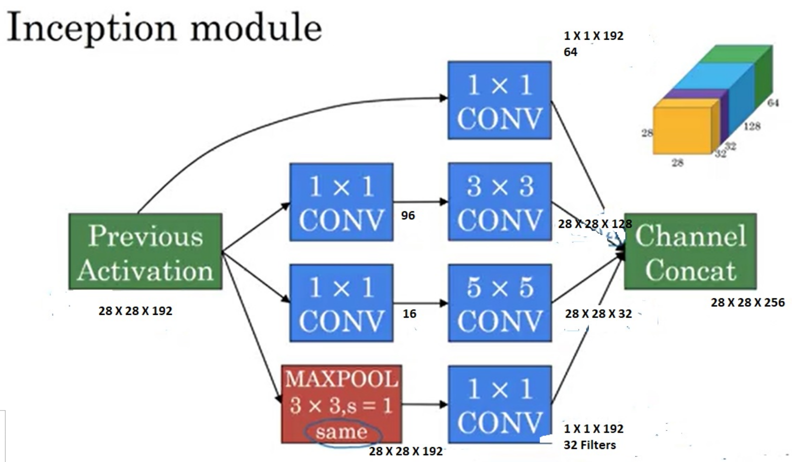
cost=5*5*192*28*28*32=12,04,22,400

Computation Cost

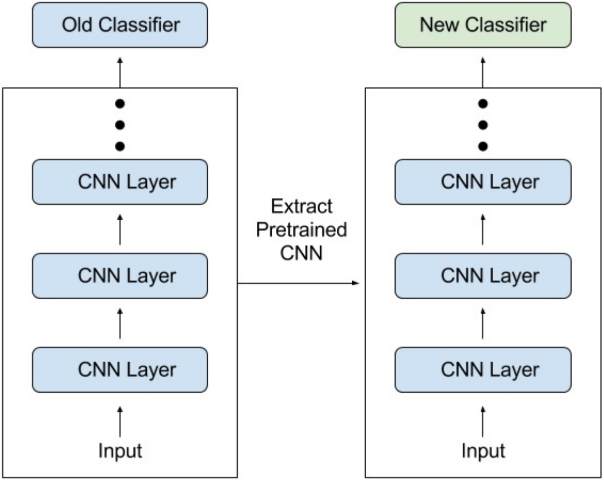


(1 X 1 X 192 X 28 X 28 X 16) + (5 X 5 X 16 X 28 X 28 X 32) = 2408448 + 10035200 = 12443648

Computation Cost



Transfer Learning



Object Detection

Data Augmentation

- Flip
- Rotation
- Scale
- Crop
- Translate
- Noise

Email : dr.vvr.research@gmail.com
(Dr. Venkataramana Veeramsetty)