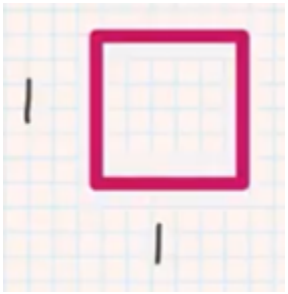
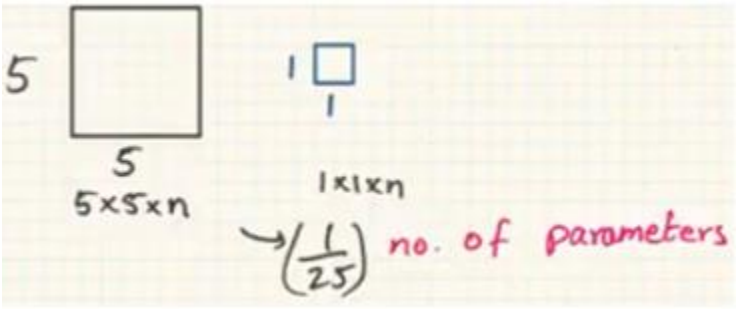
1 x 1 Convolution

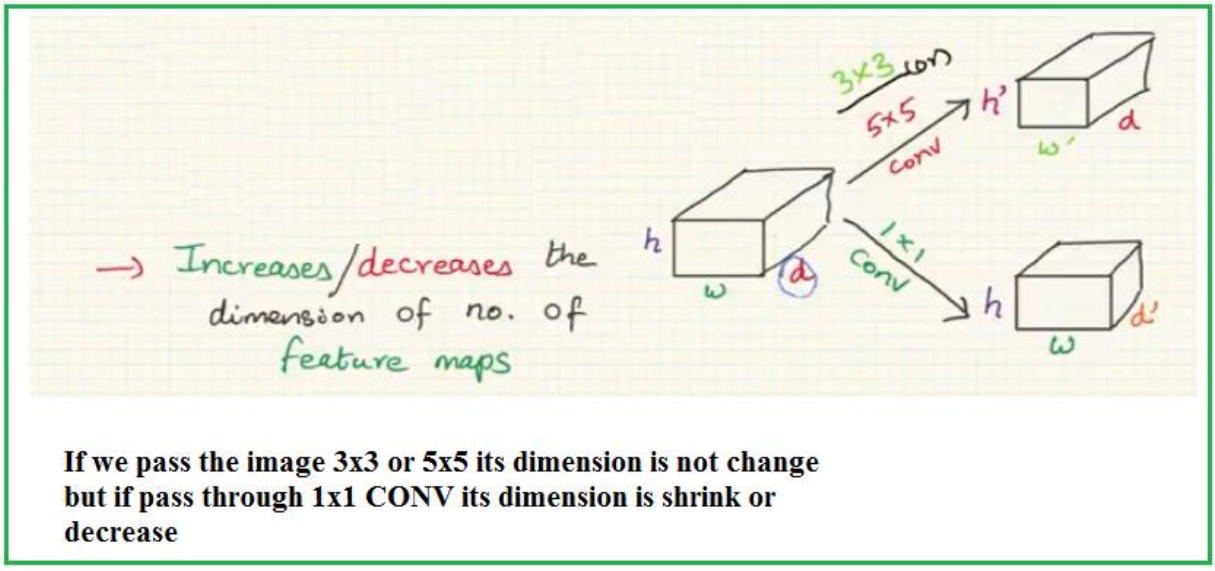
- First time introduce in the paper of Network in Network.
- Size of Filter is equal to 1.



- It uses to reduce the number of parameters.
- If more numbers of parameters chance of overfitting, so it reduces the over fitting problem.
- It also uses to decreases or increases the number of feature maps.
- Suppose we have filter 5 x 5 and other is 1 x 1, so its number of parameters reduces (1/25) times.

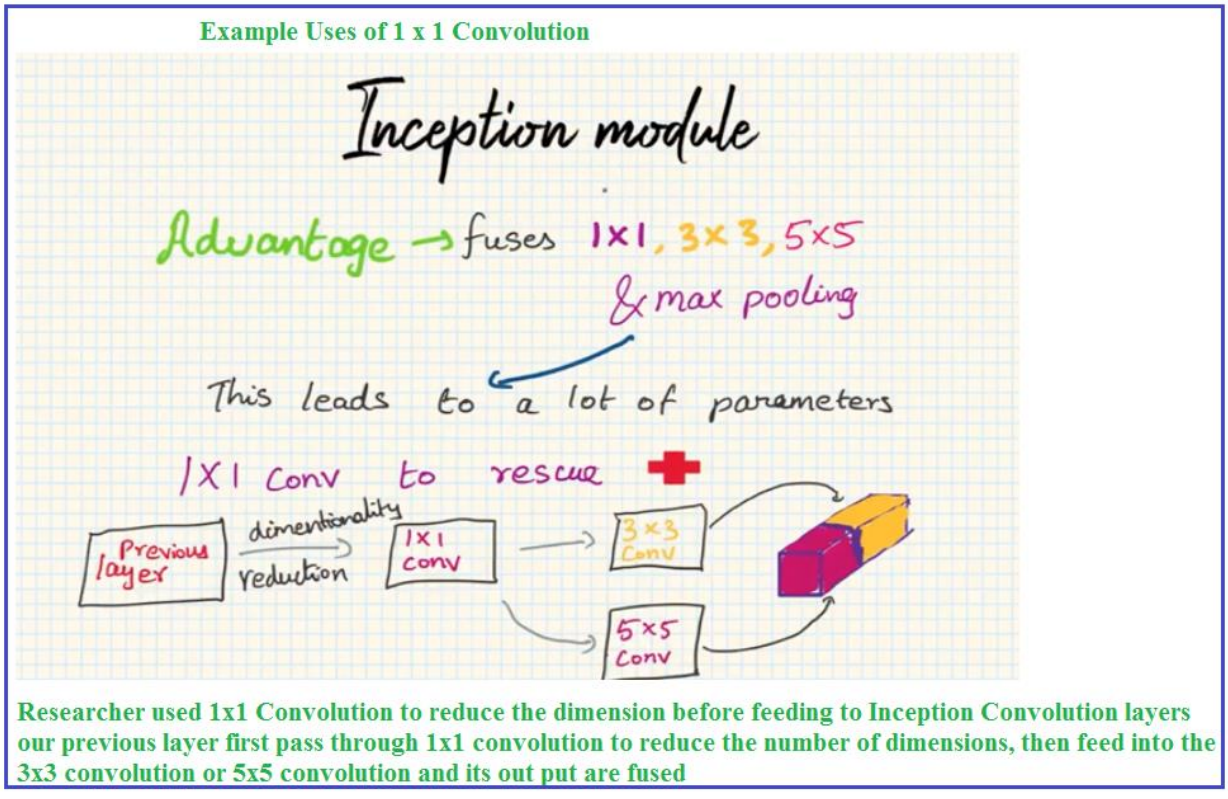


- It increases or decreases the dimension of No: of feature maps.



If we pass the image 3x3 or 5x5 its dimension is not change but if pass through 1x1 CONV its dimension is shrink or decrease

- Main application of 1 x1 Convolution is applied in Inception Network architecture

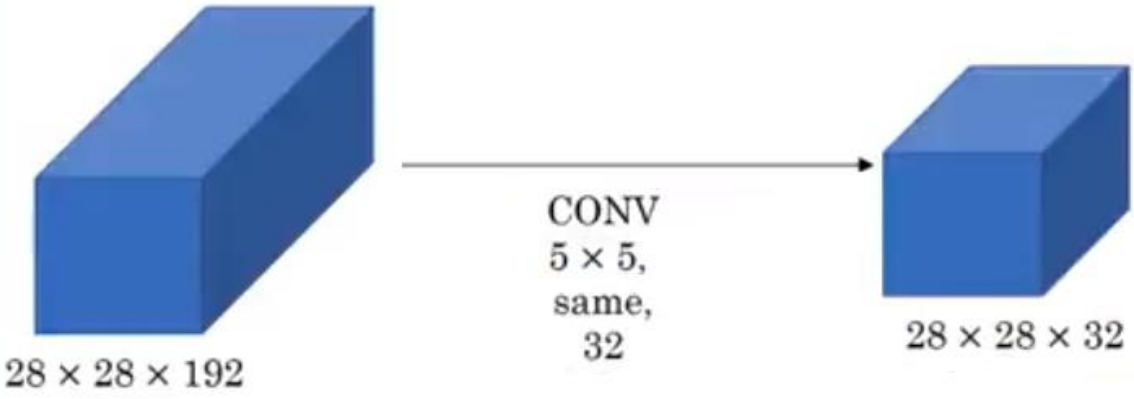


Researcher used 1x1 Convolution to reduce the dimension before feeding to Inception Convolution layers our previous layer first pass through 1x1 convolution to reduce the number of dimensions, then feed into the 3x3 convolution or 5x5 convolution and its out put are fused

- How to Reduce the number of parameters using 1x1 Convolution

The problem of computational cost

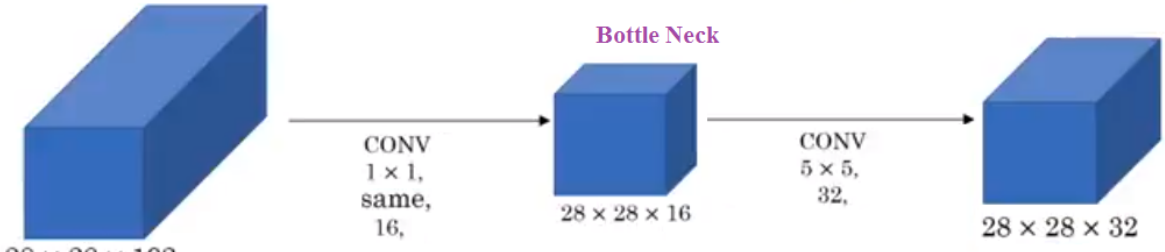
Without 1x1 Convolution



192*5*5 * 28*28*32 = 120 Million

Using 1×1 convolution

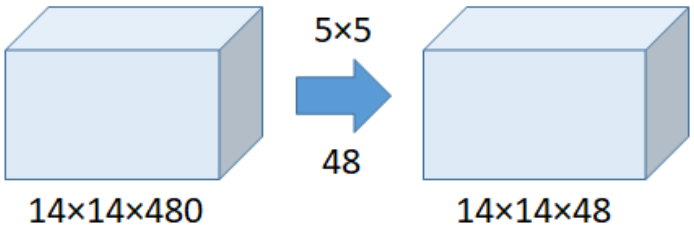
In Case of Same padding, the Height and Width of I/P and O/P Image is same



192*1*1*28*28*16	+	**16*5*5*28*28*32**
2.4 Million	+	**10.0 Million**

= 12.4 Million

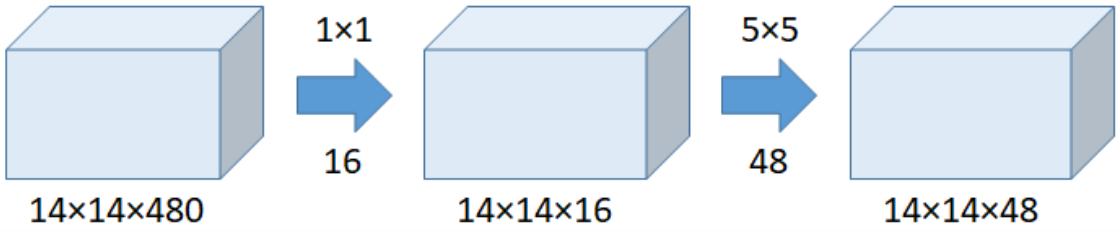
convolution **without the use of 1×1 convolution** as below:



Without the Use of 1×1 Convolution

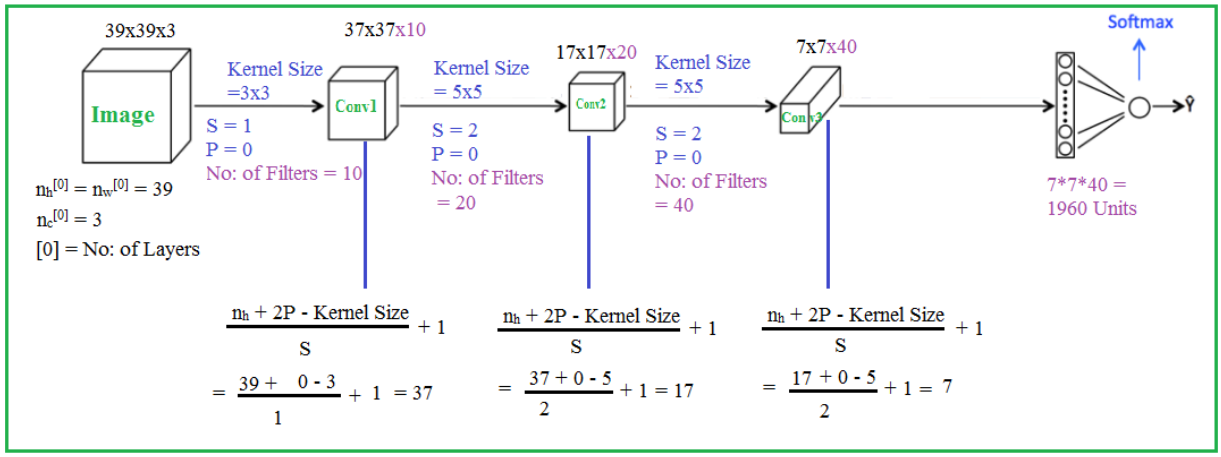
Number of operations = $(14\times14\times48)\times(5\times5\times480) = 112.9M$

With the use of 1×1 convolution:



With the Use of 1×1 Convolution

Number of operations for $1\times1 = (14\times14\times16)\times(1\times1\times480) = 1.5M$
Number of operations for $5\times5 = (14\times14\times48)\times(5\times5\times16) = 3.8M$
Total number of operations = 1.5M + 3.8M = 5.3M
which is much much smaller than 112.9M !!!!!!!!!!!!!!



1x1 Convolutions

Here apply 1x1 Convolution result is the same image

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

* 2 =

2	4	6	8
10	12	14	16
18	20	22	24
26	28	30	32

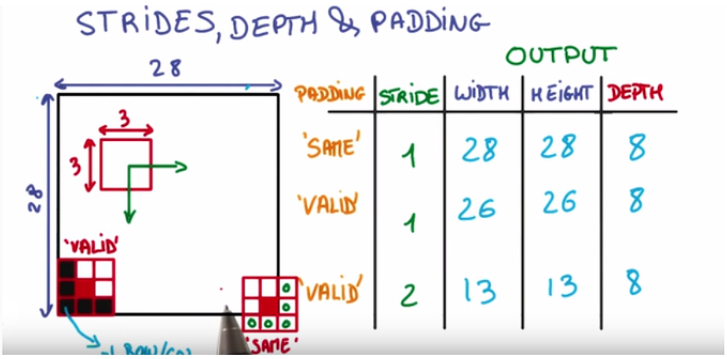
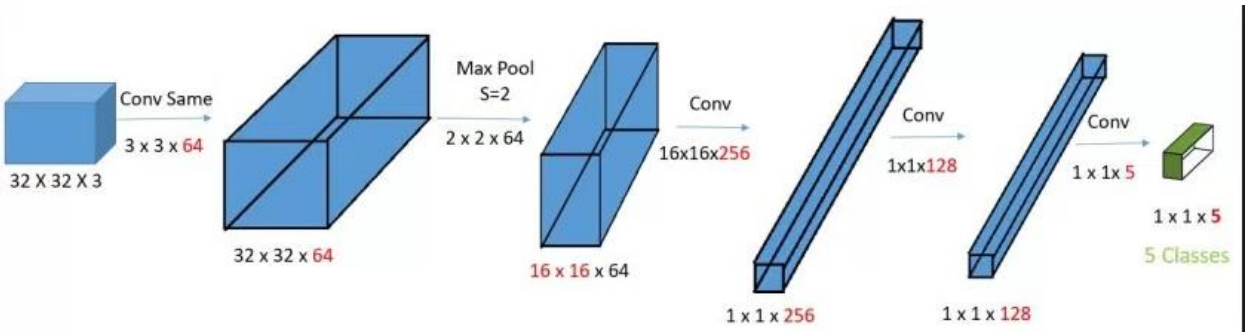


Image Size	Filter	Padding	Stride	OutPut
6x6x1	3x3	Valid	1	(6 - 3 + 1)/Stride = (28 - 3 + 1)/1 = 4x4x1
28x28x8	3x3	Valid	1	(28 - 3 + 1)/Stride = (28 - 3 + 1)/1 = 26x26x8
28x28x8	3x3	Valid	2	(28 - 3 + 1)/Stride = (28 - 3 + 1)/2 = 13x13x8
28x28x8	3x3	Same	1	28x28x8



POOLING LAYER:

If we reduce the height and width of image we can use pooling layer