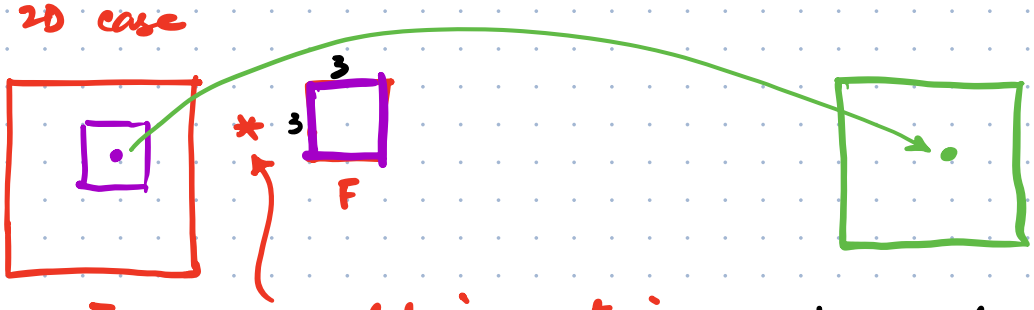
CONVOLUTIONS in Deep Networks

Simple 2D case

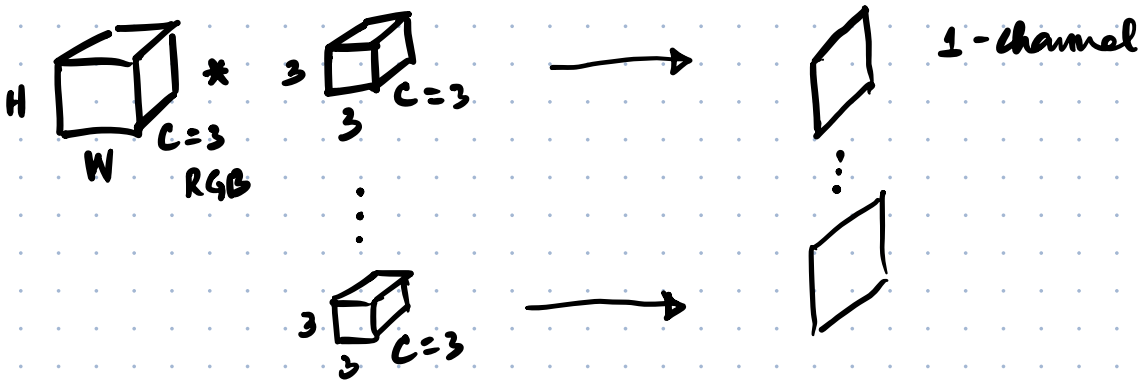


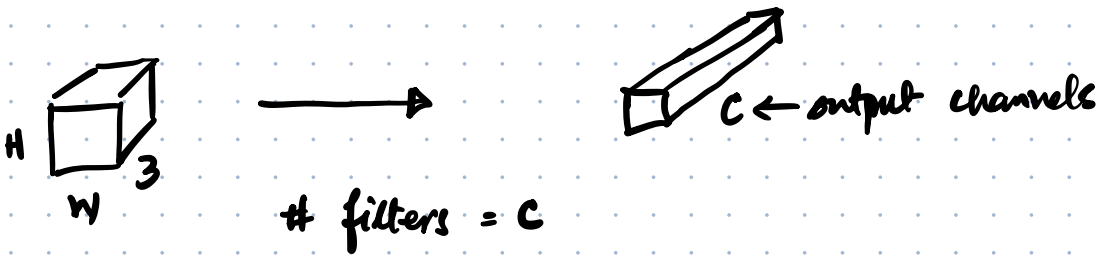
convolution operation

parameters : (3)(3)

- padding
- stride

2D convolution in Deep Networks analysing images.





filters = C

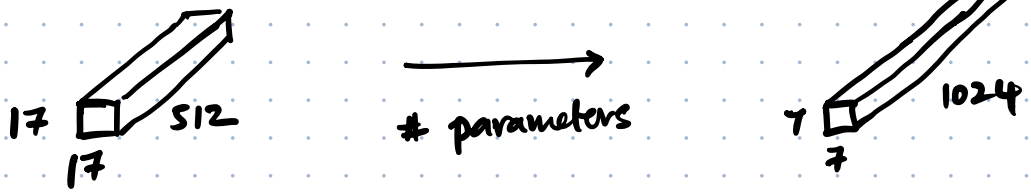
Size of fillers = $5 \times 5 \times (3)$

↑ number of input channels

num of parameters = $(75)(C)$

↑ independant of the height and width of the input and the output

depends only upon the number of channels and the size of the kernel



kernel : 3×3

parameters : (3)(3)(512)(1024) ≈ 4.5 million

Aside: How many parameters for an MLP

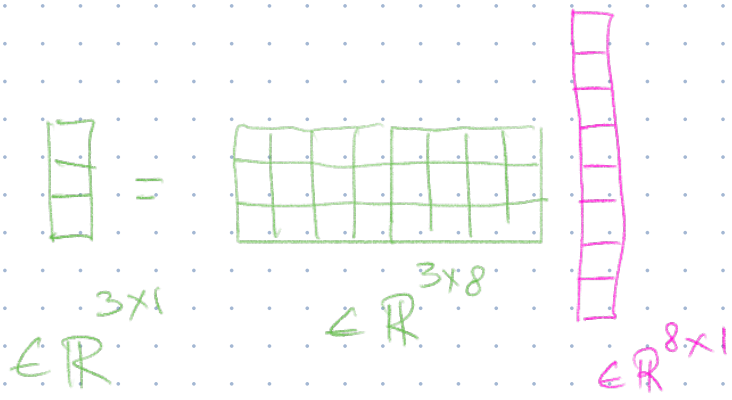
Flattened input : (17)(17)(512) = 147968

output : (7)(7)(1024) = 50176

parameters :

$$(50176)(147968) = 7.42 \times 10^{9}$$

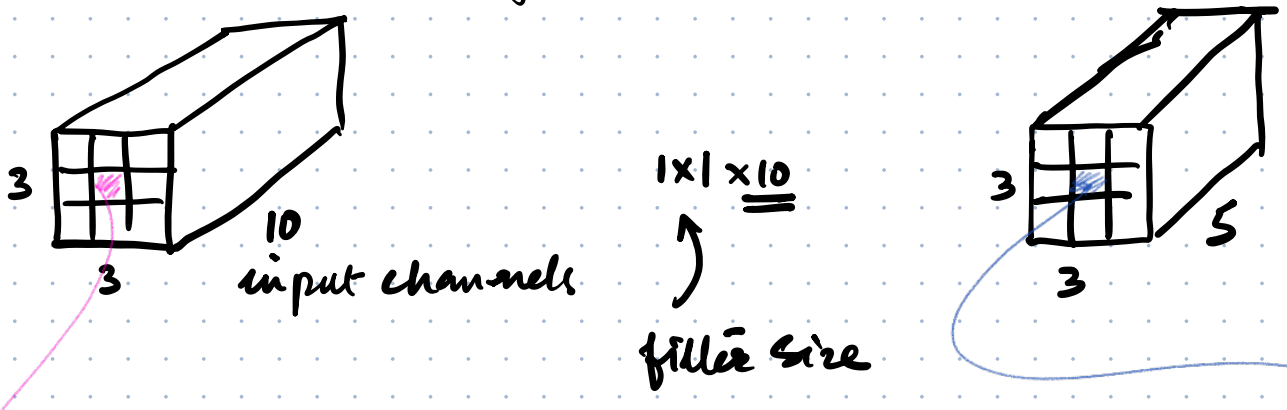
billions



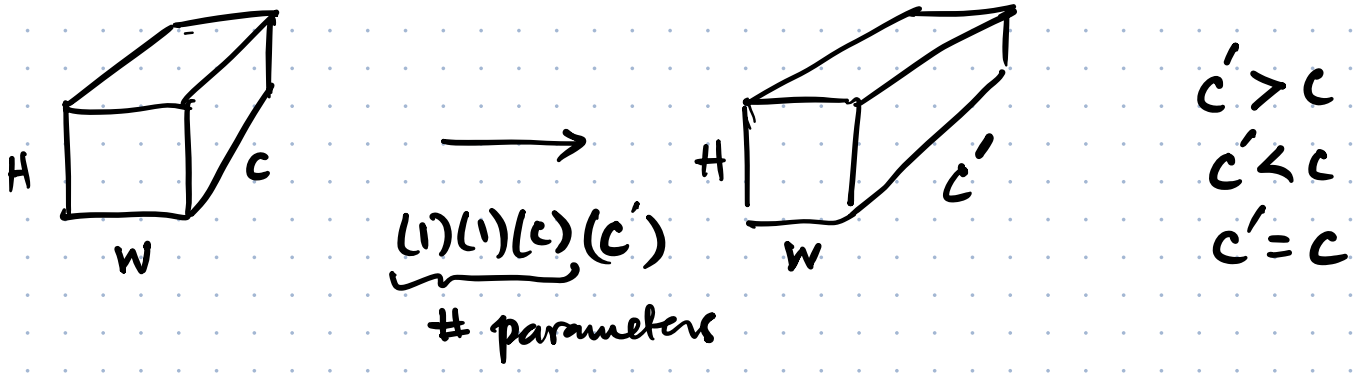
Point-wise convolution

1×1 filter

mixes channel information



parameters : (1)(1)(10)(5)



Lets go back to our problem:

Kernel : 3×3

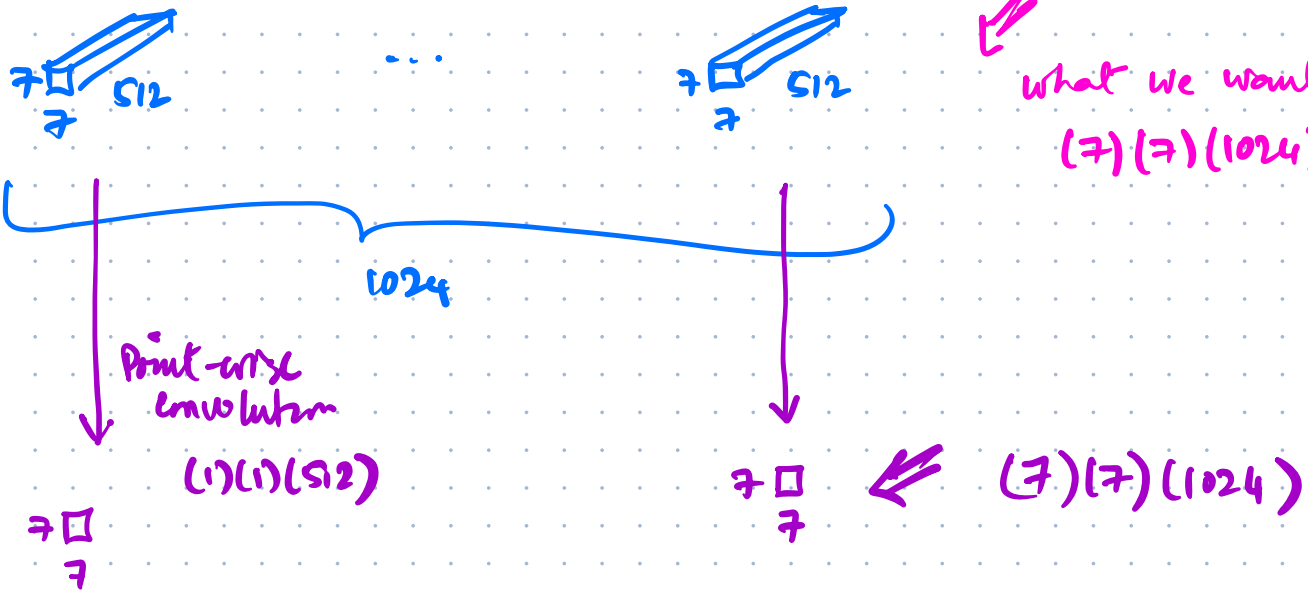
* apply 1 kernel

* applying 1024 kernels.

(7)(7)(512)(1024)

what we want

(7)(7)(1024)

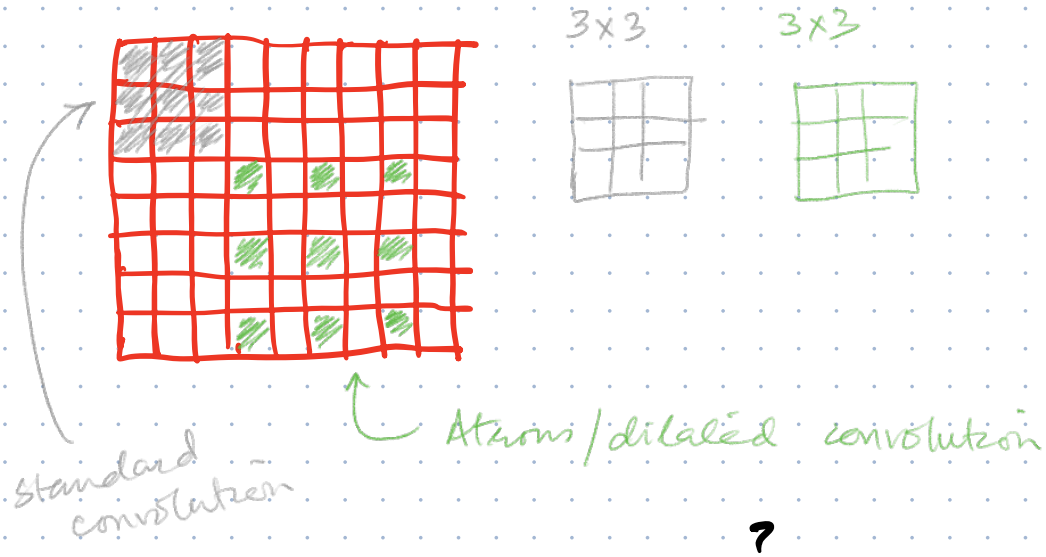


parameters ~ 50 k

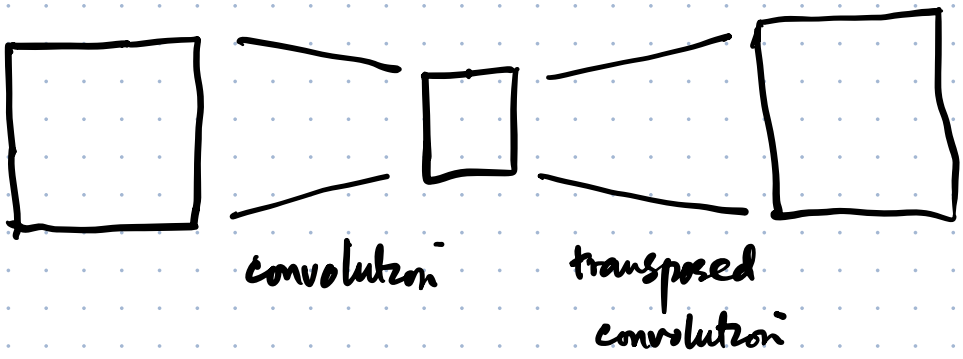
Depth-wise separable convolution:

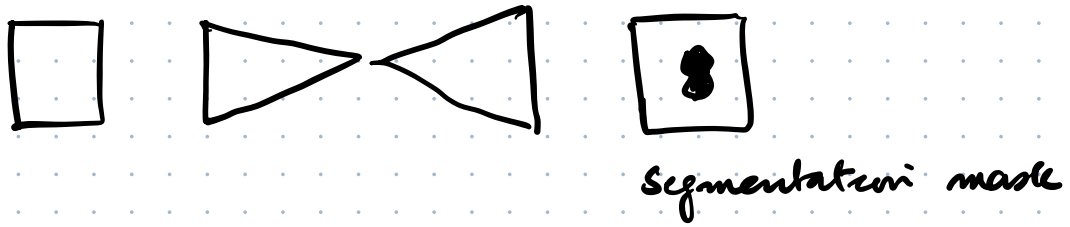
- standard
- point-wise
- depth-wise
- atrous (dilated) convolution

segmentation

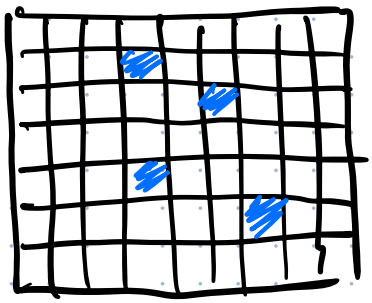


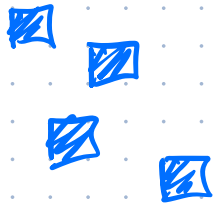
- Transposed convolution (deconvolution) ?





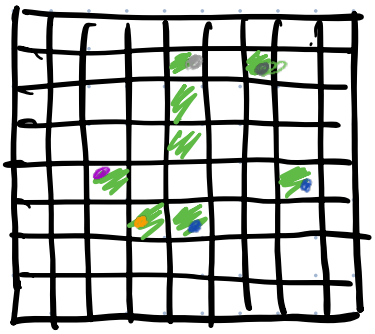
- Non - rectangular convolutions

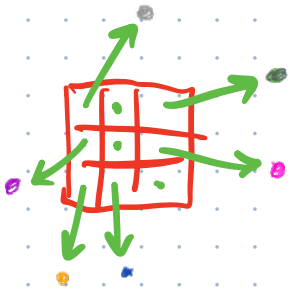




Sampling
Dot - product

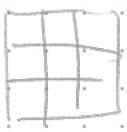
- Deformable convolution





$\in \mathbb{R}^2$

Another convolution provides offsets.



offset prediction

$\rightarrow \in \mathbb{R}^{18}$

Example -

640 x 480 x 3