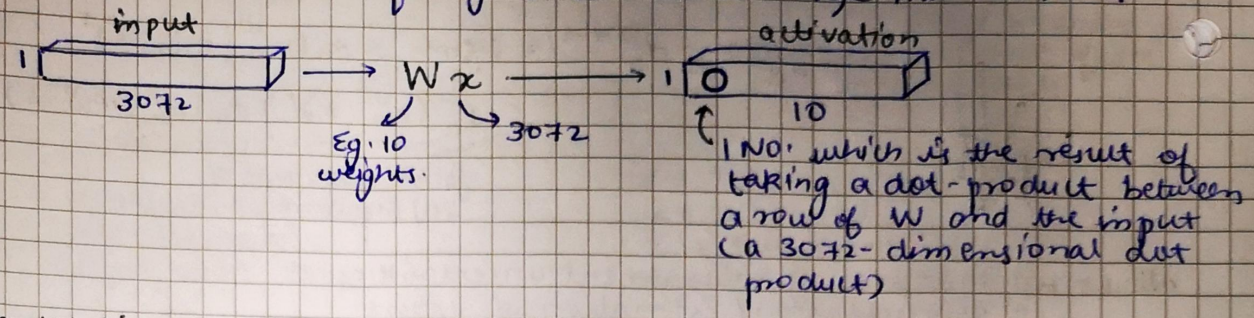
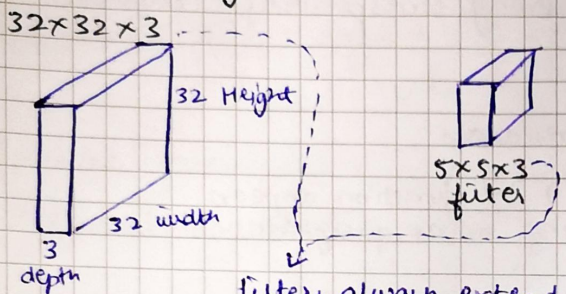
Lec-5 – Convolutional neural networks.

① Fully connected layer → In a fully connected layer, we stretch an image for eg. a 32×32×3 image into 3072×1

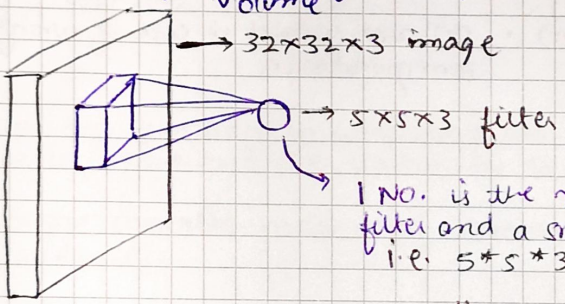


② Convolution layer → Preserves the spatial structure

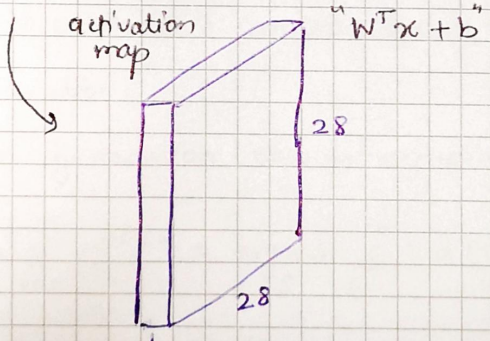


convolve the filter with the image. i.e. slide over the image spatially, computing the dot products.

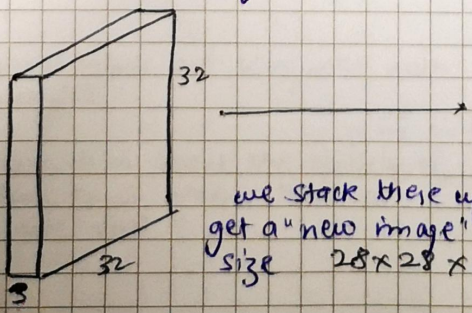
filters always extend the full depth of the input volume



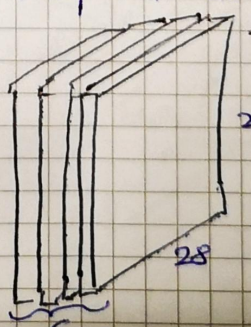
1 NO. is the result of taking a dot product b/w the filter and a small 5×5×3 chunk of the image.
i.e. 5*5*3 → 75-dimensional dot product + bias



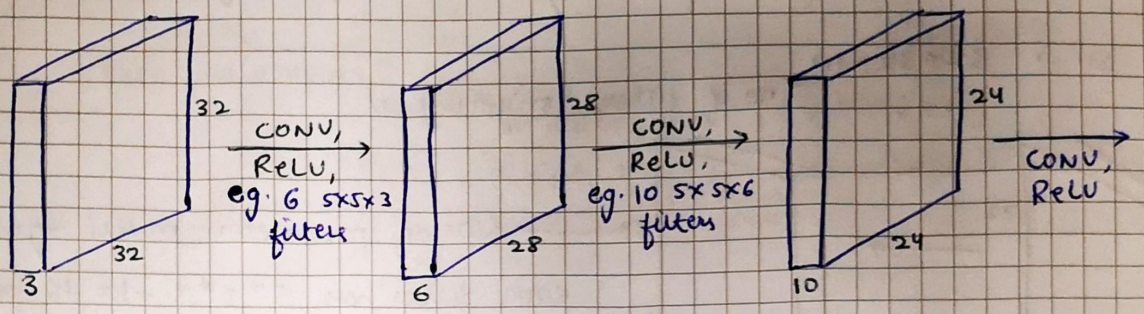
if we had 6, 5×5 filters, we will get 6 separate activation maps.



we stack these up to get a "new image" of size 28×28×6



→ ConvNet → is a sequence of convolution layers, interspersed with activation functions.

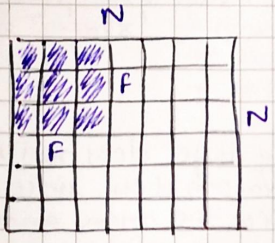


★ We call the layer convolutional because it is related to convolution of 2 signals:

$$f[x,y] * g[x,y] = \sum_{n_1=-\infty}^{+\infty} \sum_{n_2=-\infty}^{+\infty} f[n_1,n_2] \cdot g[x-n_1, y-n_2]$$

(elementwise multiplication and sum of a filter and the signal (image)).

→ Output size of a convolution layer →



Output Size: $\frac{N-F}{\text{stride}} + 1$

Eg. $N=7$, $F=3$

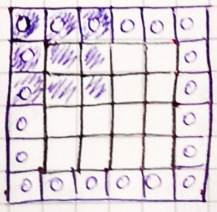
stride 1 → $\frac{7-3}{1} + 1 \rightarrow 5$

stride 2 → $\frac{7-3}{2} + 1 \rightarrow 3$

stride 3 → $\frac{7-3}{3} + 1 \rightarrow 2.33$:C

✗

→ In practise, it is common to put (zero pad) the border.



eg. input 7x7, after padding it becomes 9x9 applying the 3x3 filter again.

$N=9$, $F \rightarrow 3$, stride 3 → $\frac{9-3}{3} + 1 \rightarrow$ 3

→ In general, common to see CONV layers with stride 1, filters of size F×F, and zero-padding with (F-1)/2 (will preserve the size spatially)

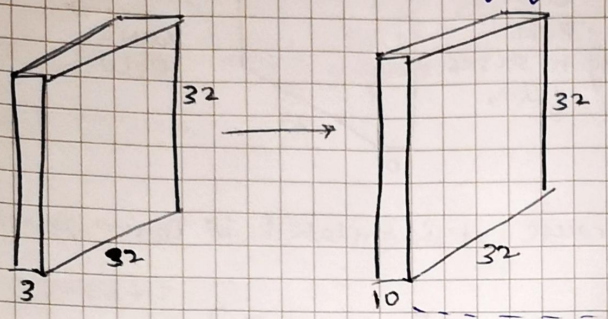
eg. $F=3$ → zero pad with 1
$F=5$ → " " 2
$F=7$ → " " 3

→ As we saw earlier, a 32x32 input convolved repeatedly with 5x5 filters shrinks volume spatially (32 → 28 → 24). Shrinking too fast is not good. It doesn't work well.

→ Eg. input Volume → 32×32×3, 10 filters with stride 1, pad 2

output Volume → $\left(\frac{32 + 2*2 - 5}{1}\right) + 1 = 32$ spatially

So → 32×32×10 → no. of filters (originally 10)

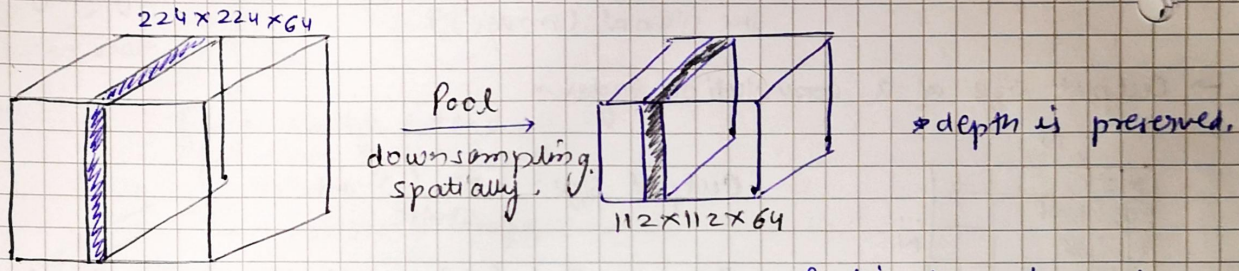


→ No. of parameters in this layer?

each filter has 5*5*3 + 1 = 76 param. (↳ bias) ($w^T x + b$)

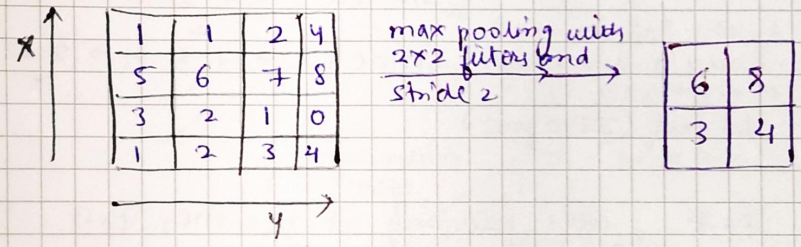
= 76 * 10 → 760 parameters

→ Pooling layer → makes the representations smaller & more manageable.
→ operates over each activation map individually.



* depth is preserved.

* generally observed is "max pooling".



* Pooling layer does not have any parameters since we only select the max among them, whereas, we associate parameters with weights such as in convolutional layers.

→ Fully connected layer → contains neurons that connect to the entire input volume as in ordinary Neural Networks.

→ Typical Network architectures look like

[(CONV - RELU)*N - POOL?]*m - (FC - RELU)*K, SOFTMAX

N → usually upto 5
m → large

$0 \leq K \leq 2$