- CNN (functional-wise)
- recall fully-connected layer (linear)

32x32x3 image

↓

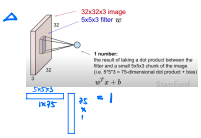
3072 x 1

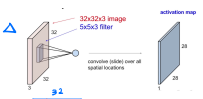
input 3072 → W 10x3072 weights → activation 10

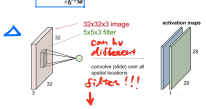
- convolutional layer 32x32x3 image (preserve spatial)

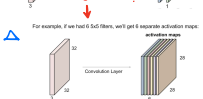
convolve the filter with the image i.e., slide over the image spatially, computing dot products

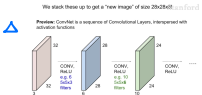
- filters always extend the full depth of the input volume



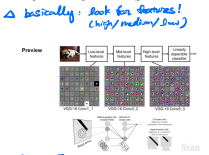




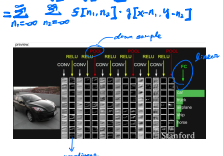


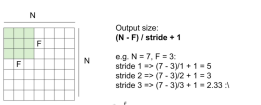


- basically: look for features! (high/medium/low)



$$[5[x,y] * 3[x,y]$$
$$= \sum_{n_1=0}^{5} \sum_{n_2=0}^{5} 5[n_1,n_2] \cdot 3[x-n_1, y-n_2]$$





Output size:
(N - F) / stride + 1

e.g. N = 7, F = 3:
stride 1 => (7 - 3)/1 + 1 = 5
stride 2 => (7 - 3)/2 + 1 = 3
stride 3 => (7 - 3)/3 + 1 = 2.33 :\

Examples time:

Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

Output volume size:
(32+2*2-5)/1+1 = 32 spatially, so
32x32x10

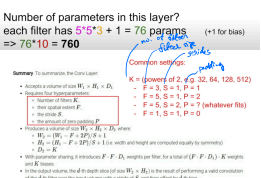
Q: no. of parameters? (the weights)

A:

Examples time:

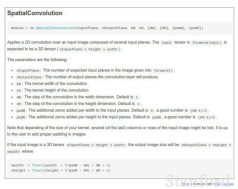
Input volume: **32x32x3**
10 5x5 filters with stride 1, pad 2

Number of parameters in this layer?
each filter has 5*5*3 + 1 = 76 params (+1 for bias)
=> 76*10 = **760**



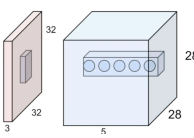
- you can do 1x1 conv

Example: CONV layer in Torch



- 5x5 filter → 5x1 receptive field for each neuron

The brain/neuron view of CONV Layer

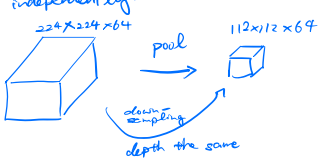


E.g. with 5 filters, CONV layer consists of neurons arranged in a 3D grid (28x28x5)

There will be 5 different neurons all looking at the same region in the input volume

- fully connected network (takes the entire input data)

- Pooling layer
  - makes the representations smaller & manageable
  - operates over each activation map independently



- max pooling (instead of dot-product we do maximization)

Single depth slice

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

x (rightward), y (downward)

max pool with 2x2 filters and stride 2 →

6	8
3	4

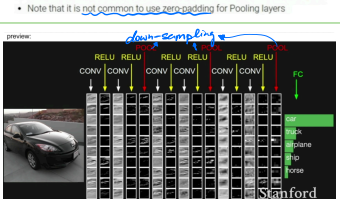
(usually not overlapping striding.)

- averaging pooling

- Accepts a volume of size $W_1 \times H_1 \times D_1$
- Requires three hyperparameters:
  - their spatial extent F, (kernel size)
  - the stride S,
- Produces a volume of size $W_2 \times H_2 \times D_2$ where:
  - $W_2 = (W_1 - F)/S + 1$
  - $H_2 = (H_1 - F)/S + 1$
  - $D_2 = D_1$

common settings:
F=2, S=2
F=3, S=2

? Introduces zero parameters since it computes a fixed function of the input
- Note that it is not common to use zero-padding for Pooling layers



- Fully connected layer

CNN output WxHxD → map → result space