PAGE
DATE

(Course4) Convolution neural network.

(week1)

→ Computer vision

- image classification (or) recognition → we take ex 64X64 picture & try figure out is there a cat or not
- Object detection → not to figure out what is in image, it is about drawing square box around the object
- neural style transfer → if we have pic [X] & we want to repaint it in diff style with style image [Y] → [X] we use neural network to put them together (or) apply style of one image to content of another img.

→ edge detection example

- The convolution operations are key element of CNN
- The earlier layer of NN detect edges then later layers detect parts of objects. later layers detect part of complete object.

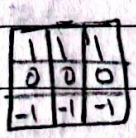
* • given a picture for computer ① we can detect vertical edges
② detect horizontal lines

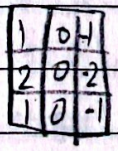
- if we have for ex 6X6 image to detect edges we construct 3X3 filter  then we make convolution operation between image & filter

→ more edge detection.

- in this video we try to know the diff between +ve & -ve edges (contors) which is transition of diff. brightness

 vertical

 horizontal

 Sobel → it focus more on pixels in the middle
filter

PAGE
DATE

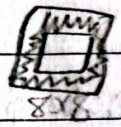
→ Padding

• in Deep NN one modification to CNN to use is padding

image 6x6 nxn * filter 3x3 fxf = 4x4 n-f+1 * n-f+1

• This cause shrinking output from 6X6 to 4X4

& the info at the edges are discarded So we pad the image by adding additional edge so from 8x8 to 6x6 by that we presrve the original image.



• Valid & Same Convolutions : how nus to pad

in valid : means no padding → nxn * fxf → n-f+1 * n-f+1

in same : pad so the output is the same as the input of original image

• f is usally odd. so it has central pixel

→ Strided convolutions

• is basic block in CNN if stride = 2 basskip two columns & 2 rows while cal. 7X7 * 3X3 = 3X3

• if nxn image & fxf filter & we use padding p & stride s so output = $\frac{n+2p-f}{s}+1 \times \frac{n+2p-f}{s}+1$

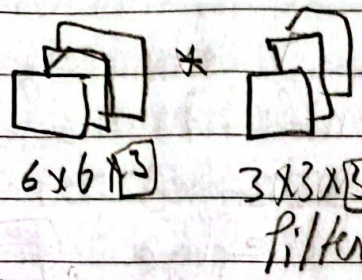
if the answer has fraction we use floor()

Cross-Correlation (VS) Convolution (*)

PAGE
DATE

→ Convolutions over volumes

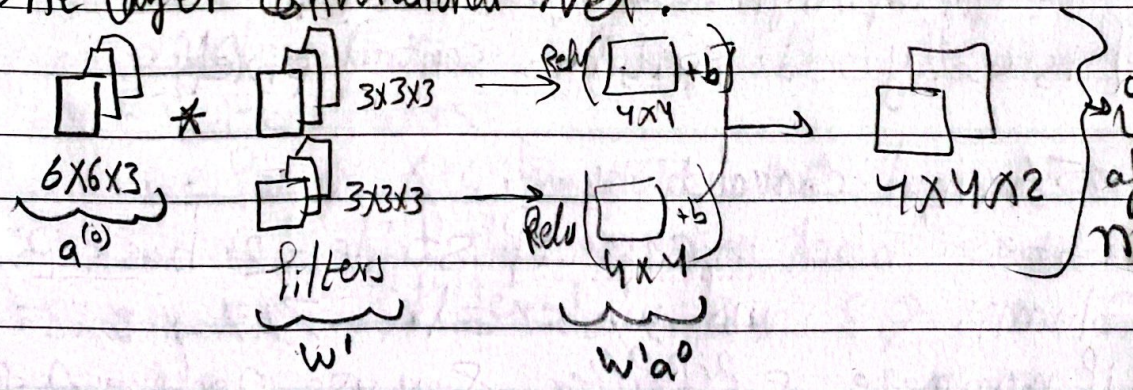
• we have learned the Conv. over 2D images, now we will see over 3D ex RGB images



- number of channels must be equal
= 4x4x1

• First we take 3x3x3 filter & replace it in upper left Corner. So we take (27) the 27 numbers & multiply them with the numbers corresponding to RGB images & then we slide the filter by one & etc.

→ One layer Convolutional Net.



• $z^1 = w^1 a^0 + b^1 \rightarrow a^1 = g(z^1)$

• if you have 10 filters 3x3x3 in one layer of NN how many param. does that layer have?

→ 3x3x3 = 27 param. +b = 28 so 28x10 = 280 params

• $f^{[l]}$ = filter size • p^l = padding • s^l = stride

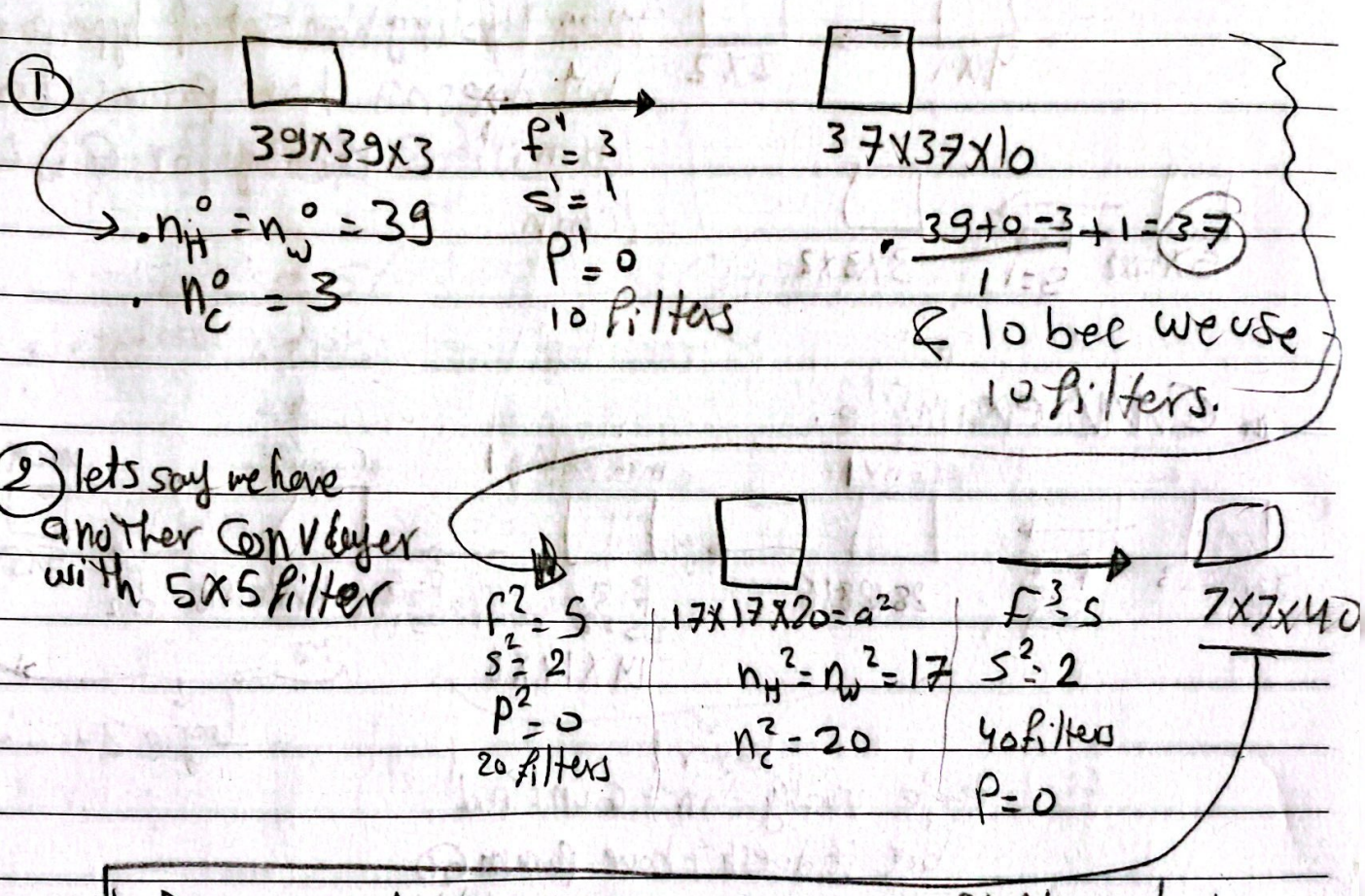
• input = $n^{l-1} \times n^{l-1} \times n_c^{l-1}$ → output = $n^l \times n_w^l \times n_c^l$

$$n^l = \frac{n^{l-1} + 2p^l - f^l}{s^l} + 1$$

PAGE
DATE

→ Simple convolution network example.

.ex: we have image & want to classify if it is my cat or not.



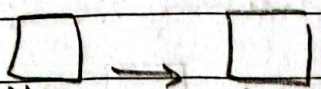
. Then we take 7x7x4 = 1960 & flatten it to vector & feed this vector to a softmax for classification

. hyperparameters in neural net. is: . filter size . stride . padding . how many filters to use .

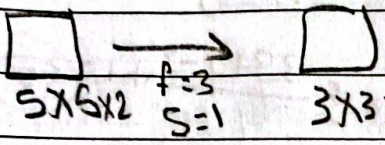
PAGE
DATE

→ Pooling layers.

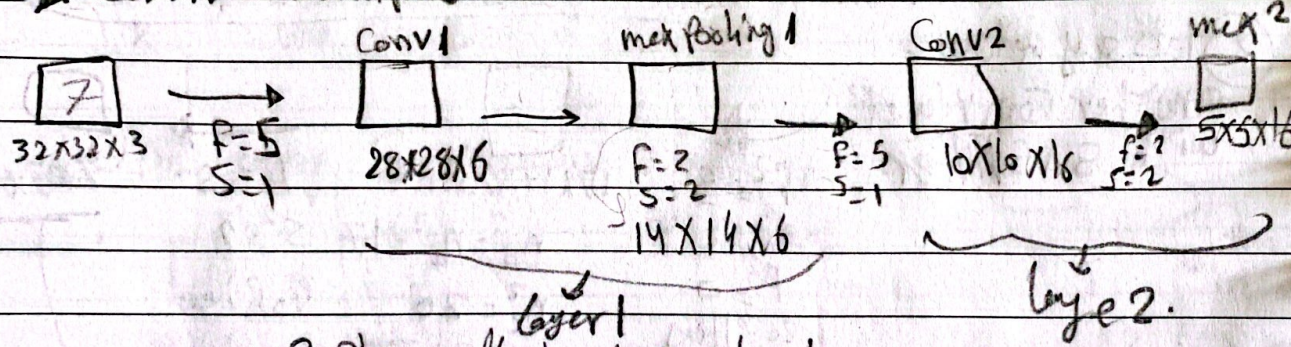
- pool layers reduce the size of representation, which speed up computation & detect features much better



max Pooling have set of hyperparam but doesn't have params. to learn, there's nothing for GD to learn

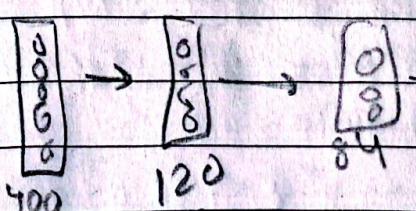


→ CNN example.



People usually don't count pool bec it doesn't have param or weights

→ 5x5x16 = 400 So then we flatten it in vector & take this 400 neurons & build a next layer having 120 units or neurons



↳ is called fully connected layer as each of 400 is connected to 120 neurons.