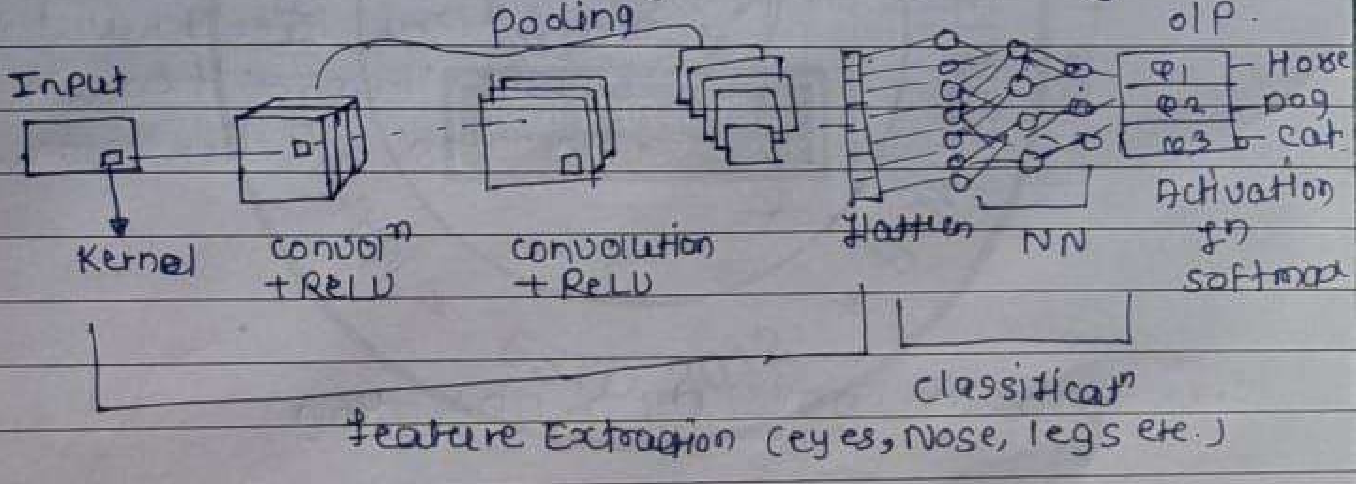
CNN (convolutional Neural N/w.

→ Images , vedio frame
→ used in Image classification , object detection.
→ CNN inspired by human visual cortex

* Convolution Operation

In this our i/p image or video will be passing through diff layers

1. Convolution layer
2. Padding
3. Stride
4. filters (Kernels)
5. max pooling → Importance
6. flaten layer → flatenening



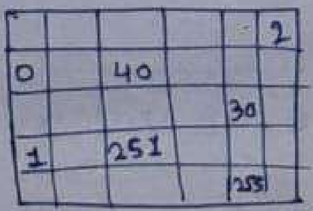
To understand this operation we need to know imgs . what are imgs

0 → black
255 → white

→ It has some pixels.
→ There are 2 types of imgs
1. blank & white
2. RGB Color img

① black & white

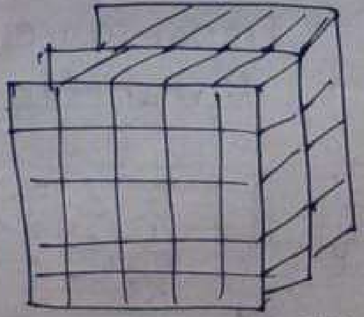
we Just have single channel
e.g. If we have 6X6 pixel img
then each & every pixel will ranging from 0-255
only.



6X6

② RGB

we will be having 3 channels, one for Red, 2nd for Green and 3rd for Blue.



5X5X3

Here also the value ranging betn 0-255 with respect to each & every channel, and when we combine them ~~tt~~ then we get combination of diff color.

e.g. steps in convolution operation

e.g ① we have i/p img.
lets say 6X6 img.

② we pass this img through filter also called as kernel, this filters are predefined in CNN. with this we just trying to extract some info from the img.
suppose, here we applied 3X3 filter.

$$\underset{6X6}{\begin{array}{|c|c|c|c|c|c|}\hline 0&0&0&255&255&255\\\hline 0&0&0&255&255&255\\\hline 0&0&0&255&255&255\\\hline 0&0&0&255&255&255\\\hline 0&0&0&255&255&255\\\hline 0&0&0&255&255&255\\\hline\end{array}} * \underset{\text{filter}}{\underset{3X3}{\begin{array}{|c|c|c|}\hline 1&2&1\\\hline 0&0&0\\\hline -1&-2&-1\\\hline\end{array}}} = \underset{\text{o|p}}{\underset{4X4}{\begin{array}{|c|c|c|c|}\hline \ & \ & \ & \ \\\hline \ & \ & \ & \ \\\hline \ & \ & \ & \ \\\hline \ & \ & \ & \ \\\hline\end{array}}}$$

whenever we pass 6X6 img in a 3X3 filter we get o|p of 4X4 how?
for that in CNN we use mathematical formula.
i.e. o|p = n - F + 1
n is size of i|p, F is size of filter
so o|p = 6 - 3 + 1

$\boxed{\text{o|p} = 4}$

Now we have i|p, in that we values 0 & 255 will try to convert this value in 0 & 1. how? by deviding each pixel by 255. basically we are doing here minmax scaling. CNN will do this operation on every i|p img.

Now we img.

0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1
0	0	0	1	1	1

for filter we value

1	2	1
0	0	0
-1	-2	1

This filter is actually called as horizontal ~~filt~~ edge filter.

$$\left.\begin{matrix}1 & 0 & -1\\ 2 & 0 & -2\\ 1 & 0 & -1\end{matrix}\right\}$$ vertical edge filter.

In convolution operation we will take that filter and just place it in top of the specific i/p img. from left.

1 step = stride

$$\begin{bmatrix}0&0&0&1&1&1\\0&0&0&1&1&1\\0&0&0&1&1&1\\0&0&0&1&1&1\\0&0&0&1&1&1\\0&0&0&1&1&1\end{bmatrix} * \begin{bmatrix}1&2&1\\0&0&0\\-1&-2&-1\end{bmatrix} = \begin{bmatrix}0&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{bmatrix}$$

① 1st operation is multiplication.
1x0, 2x0, 1x0 like this all the values became 0 in the first instance, whenever we add up this value whatever is the final o/p will place it in the op img

② 2nd operation, we move 1 step ahead to the right that is called as "stride"

if stride = 1 → jumping 1 step right.
stride = 2 → Two step right

In 2nd stride the calculation is

$$0+0+1+0+0+0+0+0-1=0$$

3rd operation

0+2+1+0+0+0+0-2-1=0

4th operation

1+2+1+0+0+0-1-2-1=0

5th = 0 8th = 0

6th = 0 9th = 0

7th = 0 10th = 0 . . . . .

This operation of multiplication & adding is called as convolⁿ

Now when we apply vertical edge filter i.e on that 6X6 img

1	0	-1
2	0	-2
1	0	-1

we will getting o/p like

0	-4	-4	0
0	-4	-4	0
0	-4	-4	0
0	-4	4	0

In CNN our aim is to bring all the pixel values betⁿ 0 & 1

here -4 will become 0 becoz lowest values will get converted into 0 & highest values will get converted into 255 i.e. or 1

again we did feature scaling (min max scaling.

so the o/p is.

255	0	0	255
255	0	0	255
255	0	0	255
255	0	0	255

Black

white

Now, whenever we pass 6X6 img in a 3X3 filter we are getting an o/p of 4X4.
as we know the formula

$$o/p = n-f+1$$
$$= 6-3+1$$
$$= 4$$

here we can say that our img size is decreasing and this should not happen, that basically means we are loosing some kind of info, in order to prevent this loss we came up with something called as $\boxed{\text{padding.}}$

③ Padding

If we want our o/p img is 6X6 itself then will do padding. its just like building a compound around img.

why use stride
1) when we want higher level features we increase stride

	0	0	0	1	1	1	
	0	0	0	1	1	1	
	0	0	0	1	1	1	
	0	0	0	1	1	1	
	0	0	0	1	1	1	
	0	0	0	1	1	1	

6X6.

It means that we are protecting the img by adding another layer on top of it.
Now what values in this specific cell we have?

for that there are diff kind of padding.

① zero padding

② whatever is the nearest value you try to put that.

o/~~p~~ then o/p = ~~8~~ $n-f+1$

$= 8-3+1$

$= 6$ → After padding.

after padding the formula is

$n+2p-f+1$ ... $2p$ = how many padding

here $8+2\times1-3+1=6$

Q. what is the importance of padding?

→ To prevent the info loss of the img

———x———x———x———x———x

In ANN we create Neural N/W, we have hidden layer in hidden layer we assign weights and in the back propagatn we ~~to~~ update this wts.

So in CNN also you have to make sure that you will update the filters. based on the i/p imgs.

here every img will be diff, black & white or RGB -

then how back propagation happens in CNN?

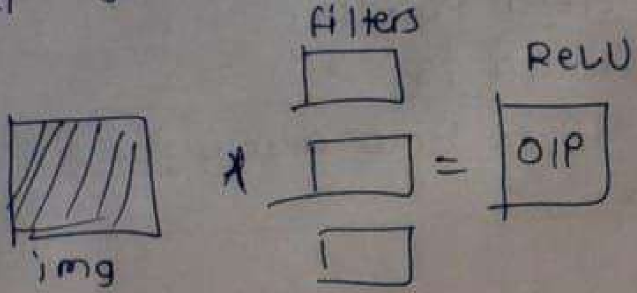
after the convolutn operatn whenever we get the o/p, on top of this for each & every value we apply RELU activatn fn.

ReLU activatn fn is nothing but $max(0,x)$, so on ~~each and every val~~ but why only ~~a~~ ReLU?

coz in ReLU during the back propagatn the derivatio can be found out. this is required in back propagat to update all kinds of filter values.

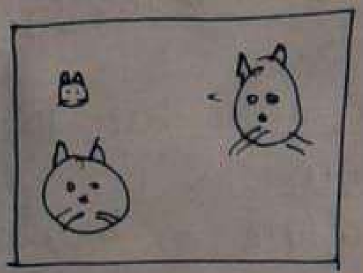
④ max -pooling (mp)

our main aim In convolution operatn is that we need to learn from ~~this~~ filters and update based on the i/p imgs.



After convolutn operatn there is something called as (mp).

e.g. let's say we have 3 cats in one img.



steps

① we start passing this img through a filter

② we get o/p

③ we apply ReLU act fn

from this steps will able to extract some info.

now there is concept called "location variant"

we used this concept whenever we ~~have~~ will be having multiple objects in the img.

our CNN till it goes ahead towards the next neural n/w. It should be able to extract more & more info.

- So locatn variant says that our objects may present anywhere, but as we pass img from many convoluth NN, It should be able to extract info more in a better or in a clear way for that we specifically used Something called as max pooling. It reduces the sice of img we apply max pooling on top of img.

types of mp

① average mp
② min pooling
③ max pooling

① max pooling :- whenever we apply mp on the o/p
(down sampling) this o/p will only be focusing on the biggest no./ the most clear no. over here.

e.g. we have img.

stride

1	2	3
4	5	7
0	3	5

① In 1st stride 5 is the highest no so mp is 5
② In 2nd = 7 (stride = 2)
③ 3rd = 3

5	7
3	7

→ feature map

low level details will elimate (down sampling)