Page No.:
Date:

Week 3 (Deep learning with Deep NLP)

Understanding edge detection

So in the last 6 classes we discuss about MLP.

Now this was one Basic Structure of a neural network.

- we are going to talk about different neural network. in this whole course.

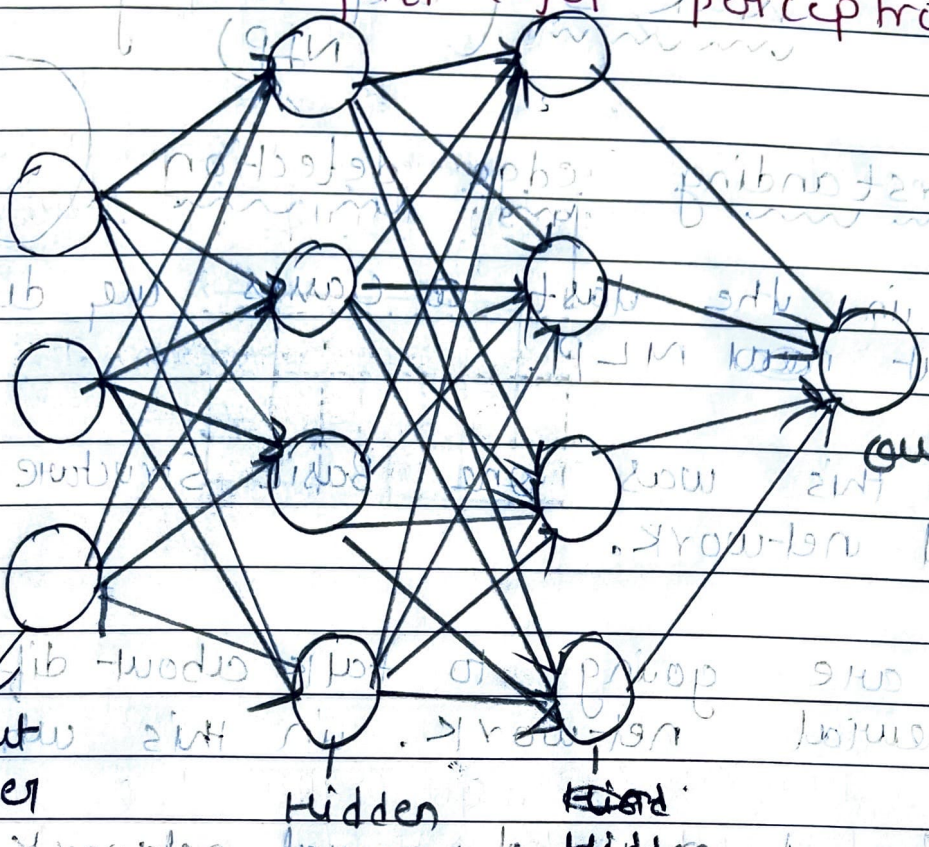
* Second part of neural network
  CNN → convulation neural networks

CNN → Basically CNN is used for any image related task.

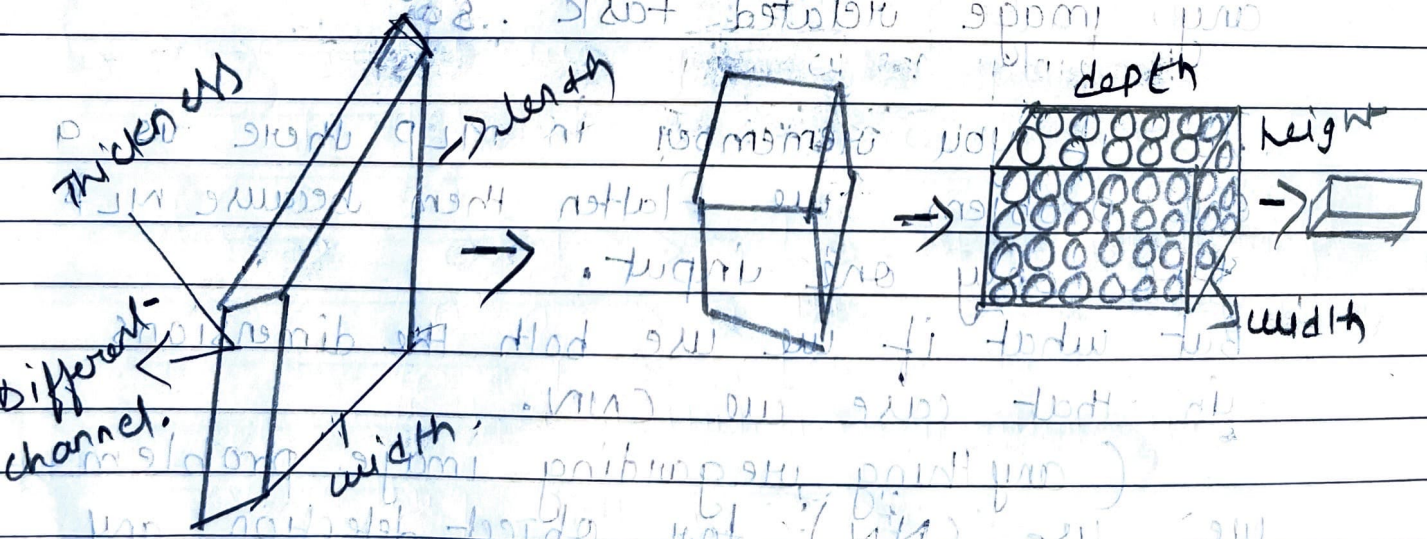
So if you remember in MLP there is a one problem we flatten them because MLP take only one input.
But what if we use both the dimensions in that case we CNN.
(anything regarding image problem we use CNN) for object detection, any classification problem we cause CNN.

M T W T F S S
Page No.:
Date:

Multi layer perceptron.



CNN -> Convlation neural network



Thickness is basically the different channels. do you know every image has RGB channel. (Red, Green, Blue)

MNIST -> Black and white images.

M T W T F S S
Page No.:
Date:

* EDGE DETECTION

0 → Black pixel
255 → White pixel

→ we are start with edge detection. why we first start edge detection i will tell you.

This is how we representing this as a number.

Edge detection basically it's you have and image and it detects the edges here.

§ → In this diagram our task it's to find edge?

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

*

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

So, what we do it's to find the edge. we convolve this with an edge detector.

* → convulation

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

→ This is our edge detector.

this is also known as a sobel edge detector. this is use to detect Horizontal lines. SO it is known Horizontal Sobel edge detector.

M T W T F S S
Page No.:
Date:

Q -> So what will happen there?

first thing -> The edge detector will be placed here

0x1	0x2	0x1			
0x0	0x0	0x0			
0x-1	0x-2	0x-1			

Now what will happen is we will multiply each of this value that was already there in the image with the value that we got from after placing this edge detector.

Q -> So, what we got after place the edge detector.

Ans -> 0 x 1 -> 0
0 x 0 -> 0
0 x -1 -> 0

After multiplying & whole thing with zeros once it has been multiplied then whole thing get added.

0			

-> Our output looks like in that shape

Page No.:
Date:

So, like we were doing in MLP also. we are multiplying the weights with the inputs. and then we are adding the bias.

In edge detector → we will multiplying all the weights. edge detector can thought as weight. So this weights with the input and we will add all thing. So this will end up with the zero

* lets go to the second value

	0x1	0x2	0x1		
	0x0	0x0	0x0		
	0x-1	0x-2	0x-1		

→

Here

Now we will place this. we will move it one step towards righ and place it

After multiplying that we will get 0. then we add the value and we get zero

0	0		

→ output.

Date:

* Third step → we will push the edge detector to one step down words.

0X1	0X2	0X1			
0X0	0X0	0X0			
255x-1	255x-2	255x-1			

So Basically what we are doing this, we are moving this detector horizontally by one step and vertically by one step.

$0 + 0 + 0 + 0 + 0 + 0 + (-255) + (-510) + (-25)$

$= 1020$

0	0	0	0
1020	-1020	-1020	-1020
1020	-1020	-1020	-1020
0	0	0	0

→ output

Same thing we are continue until we fill all the box in out-box.

* final product →

0	0	0	0
-1020	-1020	-1020	-1020
-1020	-1020	-1020	-1020
0	0	0	0

M T W T F S S
Page No.:
Date:

So, what we have done is we have convolved here an image which has one edge with an horizontal edge detection.

Q → Why we use a horizontal edge detection?
Ans - because it has one horizontal edge.

0	0	0	0
-1020	-1020	-1020	-1020
-1020	-1020	-1020	-1020
0	0	0	0

0 → Pixel
1020 → Pixel

Lets normalize this
Normalization means the values lies between 0 to 1

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

So now if you see this is the value that been convert after you have convert.

Old image →

Black
White

EDGE Detection →
we have an image. which is what? it is basically just a Matrix of numbers which represent colors.
then we convolving it with an edge detector

M T W T F S S
Page No.:
Date:

* The other way in which you can say it's this edge detector this can also be known as Kernel.

So, a Kernel it's basically use to find an edge detection.

So, I type of Kernel it's edge detector and finding out edges in a picture it's one type of convulation and it gives us one output, so this edges of a picture can also be known as feature of picture.

like I can count the number of edges and everything so it can be feature of a picture.

edge detector => Help us to find a feature of a picture. Or Kernels are used to find feature of the picture or an image.

Q -> How this Kernel used?
Ans -> with the help of convulation.

M T W T F S S
Page No.:
Date:

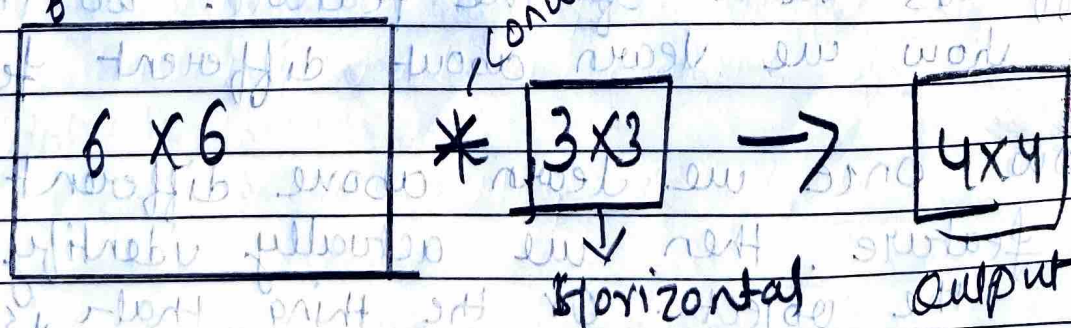
* Introduction to convolution

Edge detection → So what we see about edge detection. we said that edge detection was one of the ways in which we could extract features from an image.

Q → So, what other feature can there be?

Ans → we have edges and corners and we have many more thing that we can actually extract for images. Now there are other thing also. And I will show you all of this Obviously what the feature there extracted but I showed you very basic operation of convulation.

Q → what is was like you have image in 6×6 format. and we had a convulation of 3×3 Sobel edge detector. Horizontal Sobel edge detector. and we got a 4×4 output. and there output actually showed us where the edge was.



M T W T F S S
Page No.:
Date:

Output skewed

That was the way of identifying the edge.

3	+1	+2	+1
	0	0	0
	-1	-2	-1

3 × 3 → Sobel Edge detector also known as Kernel/filter.

3 ↑

So, this was how our edge detector looks like. So if we change this values this values then values then the new filter that will formed let say change the values to

2	3	4
1	2	6
7	4	6

so, if we change this values to some thing else this might filter might be able to identify som other feature of an image. So basically it is the features that are identified using different filter. So this values of filter are the trainable parameter this is where we learn the values, so changing the value of the filter helps us to understant differents value of the feature. so that is how we learn about different features.

Now once we learn above different feature. then we actually identify the object. Or the thing that is for

M T W T F S S
Page No.:
Date:

classification.

Q → So, let say we have a Dog here. So, what are the feature that can be learn.

Ans → Well, you can learn about eyes, tears, tougue tikking outs, two eyes etc. and all this feature can be learned once all this feature are learned and once we can extract this feature from an image. and then we can say. eyes, tears, tongue tikking, out, and two eyes. so this has to be a dog. and so the convulation neural network can then identify Okay this image that of a dog.

* By the way if you turn this Horizontal edge detector by 90° you will get the Vertical Edge Detector.

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

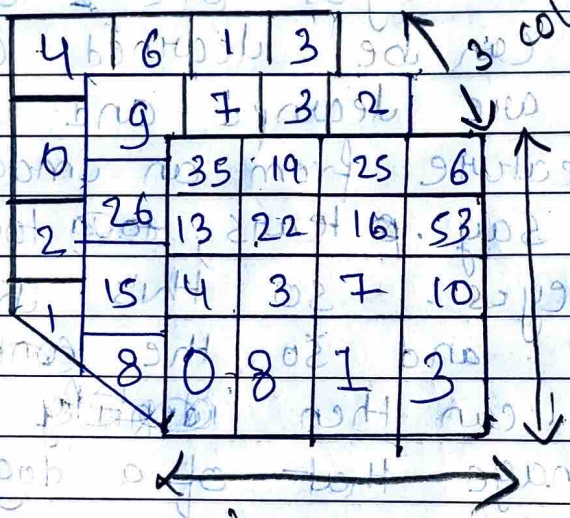
→ Vertical Edge Detector.

So this are some filter values that we know because they are used to identify the edges. Now a few changes here and there tweaks. now here and there we will be now able to identify even diagonal edge detector and different edge detector. but it is not upto us understand what numbers and which combination

M T W T F S S
Page No.:
Date:

of number will be identify a particular feature. it is for the model to learn on it own. identify the feature then say okay. So this image, image of Dog.

Q -> How actual image look like?



This image is 4 x 4 x 3

* This time we will not understand about this work specifically on edge detection. but we will learn about convution as a whole. How this work? different terms and everything.

M T W T F S S

Page No.:

Date:

Input

4	9	2	5	8	3
5	6	2	4	0	3
2	4	5	4	5	2
5	6	5	4	7	8
5	7	7	9	2	1
5	8	5	3	8	4

6

↓ 6

filter

1	0	-1
1	0	-1
1	0	-1

Result

2		

$n_H \times n_w = 6$

H → Height

w_0 → width

parameters:

Size $f = 3$

Stride $S = 1$

padding $P = 0$

$2 = 4 \times 1 + 9 \times 0 + 2 \times (-1) + 5 \times 1 + 6 \times 0 + 2 \times (-1) + 2 \times 1 + 4 \times 0 + 5 \times (-1)$

M T W T F S S
Page No.:
Date:

Q → you have an input of 6 X 6 i am using a filter of 3 X 3. I am getting an output of 4 X 4. How is this happening like is there something that control this output value? or why this specific number that happening.

* Now come back to stride and padding. So, one that you need to understand this is it is using stride and padding that we actually control the output. and all the size of result.

Stride → That whole thing that we are doing here then we moved by one pixel is known as stride. and down way we move one pixel.

Q - when your Stride : S = 2

Input

4	9	2	5	8	3
⇒		2	4	0	3
2	4	5	4	5	2
5	6	5	4	7	8
5	7	7	9	2	1
5	8	5	3	8	4

6 (height) 6 (width)

Dimension - 6 X 6

filter

1	0	-1
1	0	-1
1	0	-1

Result

2	1

Parameters:
Size f = 3
Stride S = 2
padding P = 0

1 = 2 X 1 + 5 X 0 + 3 X (-1) + 2 X 1 + 4 X 0 + 3 X (-1) + 5 X 1 + 4 X 0 + 2 X (-1)

M T W T F S S
Page No.:
Date:

Stride basically means number of Pixels you are shifting.

Now you can see in diagram automatically since this can accumulated only two time we cannot have one more time the kernel the we have only two pixel here we do not have another third pixel there so we not have another value. So Hence are result is 2 x 2

* Padding → you have the input you add one more layer outside the whole image like a border or a frame to the image. so i am adding one layer of Zeros outside. so the actual image is 6 x 6 but since i added one more layer now the actual dimension the image is 8 x 8

since i am adding one layer of Zeros outside the actual image a my paddings become 1

padding = 1

M T W T F S S
Page No.:
Date:

Input

0	0	0	0	0	0	0	0
0	4	9	2	5	8	3	0
0	5	6	2	4	0	3	0
0	2	4	5	4	5	2	0
0	5	6	5	4	7	8	0
0	5	7	7	9	2	1	0
0	5	8	5	3	8	4	0
0	0	0	0	0	0	0	0

Dimension - 6 X 6

filter

1	0	-1
1	0	-1
1	0	-1

* = Result

-15		

parameters:
Size $f = 3$
Stride $s = 2$
padding $= 1$

$-15 = 0 \times 1 + 0 \times 0 + 0 \times (-1) + 0 \times 1 + 4 \times 0 + 9 \times (-1) + 0 \times 1 + 9 \times 0 + 6 \times (-1)$

$= 15$

Now, if we have stride of a 2. that case what will happen? Next time your filter will place frome after 2 pixel.

* Types of padding

There are two types of padding.

1 - ~~valid~~ valid padding → No padding is known as valid padding.

2 - Same → it get padded as many times as require. So that the input image dimension = Same as a result image dimension. so, that means if we want result of the same dimension as the actual image, so the actual size was 6 X 6 and we need the result is 6 X 6 again. ~~so~~ again in that case we

M T W T F S S
Page No.:
Date:

we will pad How more zeros layer are required that will be padded itself so that our result is 6 X 6.

Q - why we this needed the same padding is required Some time what happen is we are actually loosing some of the values this image was 6 X 6 output is coming to 3 X 3 we have actually loss some values. Some information of the image we dont want that to happen we want to dimension remain the same.

Q - How we getting 3X3, 6X6 pixel.

Ans -> So, we have this formula, this formula is basically used to describe what will be value of the result

So, the value of the result a convolation

n = convulation

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

$n^{[l-1]}$ = input image dimension

$2p^{[l-1]}$ = 2X padding

$f^{[l]}$ = Size of filter

$s^{[l]}$ -> stride size

M T W T F S S
Page No.:
Date:

So that means the result o's

$$\text{Result} = \frac{n + 2p - f}{s} + 1$$

[ ] → This math.floor.

Q- for this equation where we want a padding to be same. in that what will be the actual number of padded require?

Sol → Resultant $\left[\frac{n + 2p - f}{s} + 1\right]$

$$6 = \left[\frac{6 + 2p - 3}{2} + 1\right]$$

Now, the result has to be because the padding is same. So the result dimension are the same as the input dimension so this is equal to same.

$$6 = \frac{3 + 2P}{2} + 1$$

$$5 = \frac{3 + 2P}{2}$$

$$10 = 3 + 2P$$

$$7 = 2P \qquad = 3.5 = P$$

M T W T F S S
Page No.:
Date:

→ Just go and learn it basically learn it. basically learn it because you are going to needed many times you will actually require it to get the value of each thing.

This is how Stride and Padding and the Size of filter actually control the dimension of the result that we are going to get.

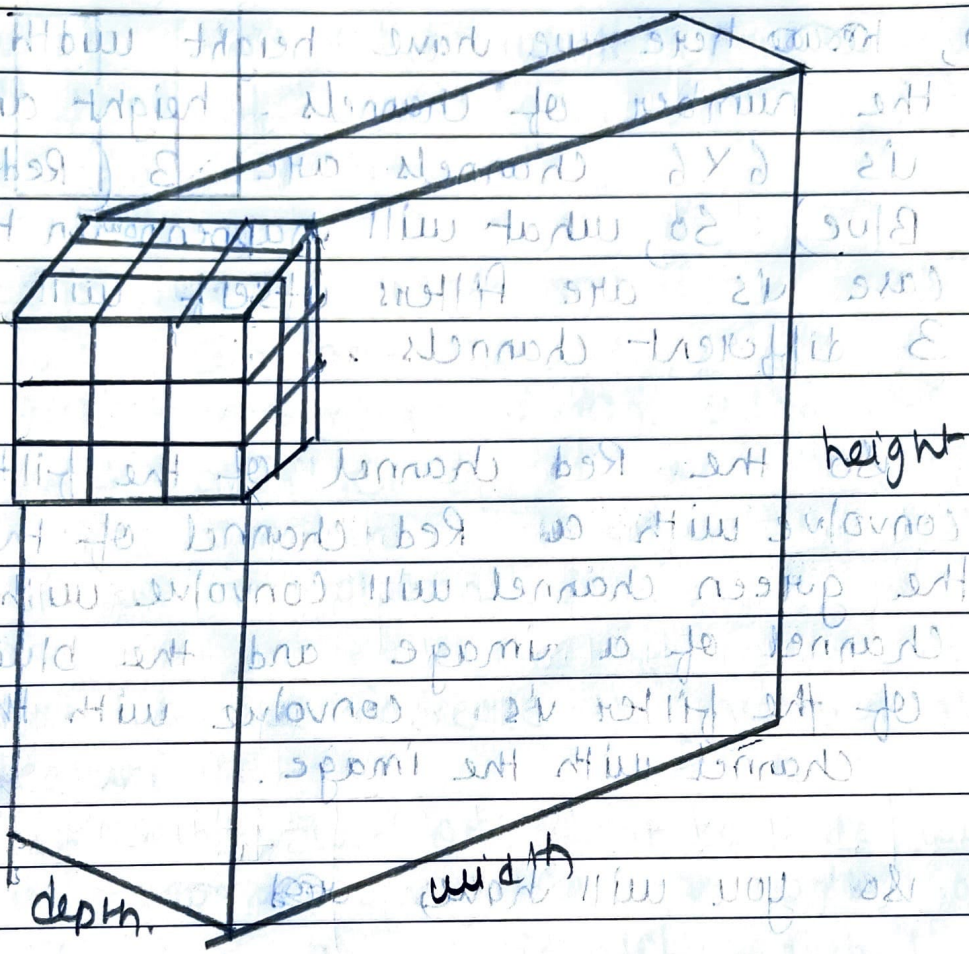
Q2- we have two dimension images. but we always talked about three dimension images. which had RGB channels.
So, How are deal with 3 dimension images

Ans → In that case what happen is our kernals are no longer flattern kernals. So our kernals themselves will have some depth to it. So in that case kind of rubics cube looking thing will be are kernal it will have three channels itself because it have three channels this is three channels is to compliment the three channel of the image. So our kernal no longer be flat we will have a 3 dimension kernal itself. So that is How we handling such kinds of problem

M T W T F S S

Page No.:

Date:



M T W T F S S
Page No.:
Date:

So, here we have height width and the number of channels - height and width is 6 X 6 channels are 3 (Red, Green, Blue) So, what will happen now in this case is are filters itself will have 3 different channels.

So the Red channel of the filter will convolve with a Red channel of the image the green channel will convolve with green channel of a image and the blue channel of the filter is convolve with the blue channel with the image.

So you will have

-> red channel of the image * red channel with filter +
-> Green channel of the image * green channel of the filter +
-> Blue channel of the image * is convolving of the blue channel of the filter

and they all get addedup adder to get the final result. So our result even though we have multiplying three different channel with three different filters will we be still getting a 2 dimensional result.

M T W T F S S
Page No.:
Date:

* So whole thing the whole result gives us one feature of the image. So we were using edge detection.

edge detection was one feature of the image. I am not sure let just say whole which numbers i am using in filter may it's a filter of corners. So we got one feature through this whole filter and convolution things.

Q -> what if we increase the number of feature

we dont want just one feature. we want one convulation network so that it is able to identify both the features and all the feature include edges, corners point, and everything in the image itself

So, we obeviously need more than one filter because one filter may be identify edge we need another filter that will identify a corner. and many more such feature. so that is why we increase the number of filter. n.

M T W T F S S
Page No.:
Date:

Q → what is pooling?

Ans → So, we knew about the edge detection. and we knew how Sobel edge detector were used to identify the edges and we know the edges can be used as one of the features

So, we had a structure like that—

it was convolved with filters. we had 3 dimensional filter. i. we have RGB images so we need 3 dimensional filters. to actually solve this RGB images. So what would happen was that for each image the Red channel converge with red channel of the filter then green channel will convolved with green channel of filter. and so on.

The output is 2 dimensional. Since we used 2 filters so the final dimension will be 2. So that means we getting 2 outputs or 2 features. They are not exactly called feature. This are known as feature map.

M T W T F S S
Page No.:
Date:

* Number of feature map = number of filters

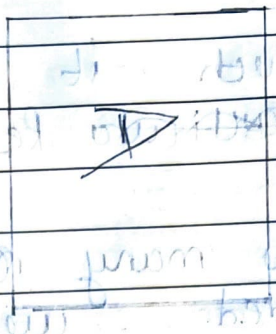
* -> usually for convulation we used Relu as Activation function. Because it is easier to understand using that. So, once you get that you have Relu over the weight + Bias.

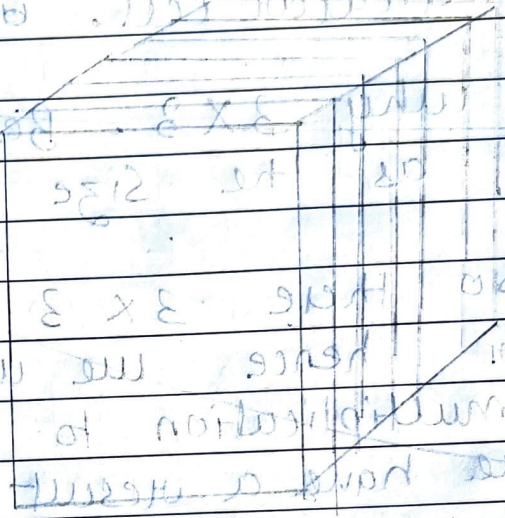
$$Y = \sum w x_i + b$$

it was Summision of whole thing so, this whole thing is there. Now you will Relu over the whole thing.

-> Relu is easier to do even CNN. because Relu is basically just maximum(0,z)

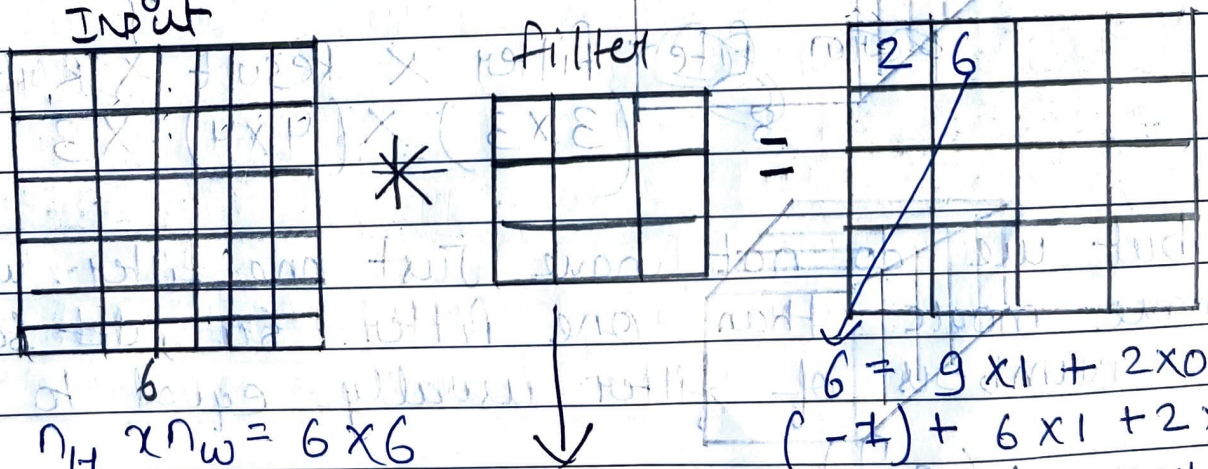
maximum (0, z) so, whichever is maximum either Zero or the number itself the value itself will be kept. else it will not be kept.





Max pooling -> Q -> why do we need max pooling? what pooling does it's basically reduce the dimension.

Q - why do we reduce the dimension?



$n_H \times n_W = 6 \times 6$

$6 = 9 \times 1 + 2 \times 0 + 5 \times (-1) + 6 \times 1 + 2 \times 0 + 4 \times (-1) + 4 \times 1 + 5 \times 0 + 4 \times (-1)$

Parameters:

size $f = 3$

stride $S = 1$

padding $p = 0$

M T W T F S S
Page No.:
Date:

→ How much it is require to calculate for this particular Result. particular cell.

So, How many multiplication are required. we require 3 x 3 multiplication to calculate a particular cell.

Q → why 3 x 3. Because we have 3 x 3 as the size of our filter.

So these 3 x 3 multiply with input layer hence we will be having 3 x 3 multiplication to find one result. but we have a result of 4 x 4.

So our input is a one dimension this is a black and white image but we also have RGB channels. So for RGB

filter x Result x RGB

$(3 \times 3) \times (4 \times 4) \times 3$

but we do not have just one filter. we have more than one filter So, let say numbers of filter usually equal to (2^n). let say we have 32 filters. but then again that is a just for a single convolution layer. we have multiple convolution layer

M T W T F S S
Page No.:
Date:

So, if if det i say i am using 3 convolution layer so you have again multiply with 3.
Now. it still not a dot. its very easy for computer.

Q - what is the problem again
The input 6 x 6. i mean you are not going to find any image that 6 x 6 even the easiest dataset mnist dataset had 28 x 28. So this

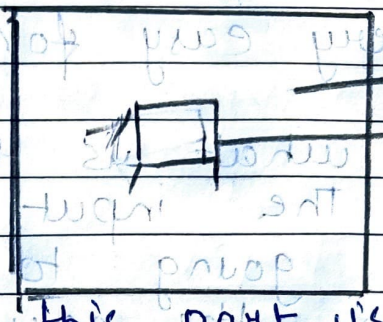
So this whole thing changes. so you are not going to get a 4 x 4 result infact.

* And even now you have camera and videos like 4K quality so 4K 1080 p. this is actually a dot to calculate. for a convolution itself. this is actually a dot to calculate for a convolution itself. and its not a one layer its a more than one layer. laptop is going to hang. the ram is going to use dot of the resources. so it is not possibly feasable. that is why we have concept of pooling.

M T W T F S S
Page No.:
Date:

pooling -> it reduces the dimensions. without actually loosing all important information in a image

So let suppose i have a image the whole this background and in image i have table. and whole thing is background. So i know for a fact this part is going to be important not a everything in image is importat. but we still have convolute the whole thing. hence we have the concept of pooling



Q - So, what is actually happen in pooling

Max Pooling

4	9	2	5
5	6	2	4
2	4	5	4
5	6	8	4

→

9	5
6	8

→ Max pooling

Avg Pooling

4	9	2	5
5	6	2	4
2	4	5	4
5	6	8	4

→

6.0	3.3
4.3	5.3

→ Avg pooling

M T W T F S S
Page No.:
Date:

* pooling is two types
  (1) → Max pooling
  (2) → Avg pooling

1- Max pooling → Basically you have to define or decide how much you want to reduce.

So, see in figure. an original image and i am going reduce the size so, in that case what i will do is. i will do max pooling.

Q- what happen in max pooling is?
it takes nearest neighbors of a particular number and we find the largest pixel value and we keep it in the same way

4	9
5	6

we have the largest value of 5 is taken and the rest is discarded. and so on.

2	5
2	4

* Avg pooling → in the say way we have avg pooling but avg pooling but avg pooling does is basically it takes the average of all 4 points and you get the value. and so, on.

Q → There is also one more know as Minimum pooling.

Ans--> it can select only minimum numbes