CNNs

Shao Michael

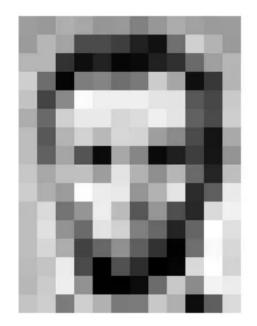
CNNs: what do they do

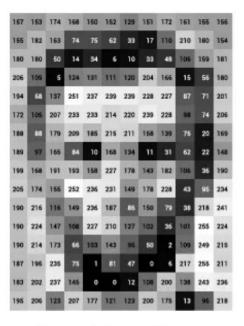
 They are a kind of neural network that does image-related tasks, such as image classification really well.

also can do other things

How are Images stored

Images are basically a bunch of numbers (typically between 0 and 255), representing the color, stored in a grid.







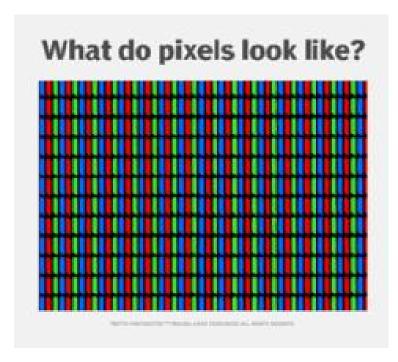
Grayscale image | Source

How are Images stored

How do we represent color with numbers?

Our computers display color with only 3 base colors:

Red, Green, and Blue (RGB)



0

How are Images stored

How do we represent color with numbers?

We can use 3 numbers, each corresponding to a shade of red, green, or blue to represent a specific color.

RGB Values for Common Colors

R	G	В	
255	0	0	
0	69	100	
255	255	0	
0	255	0	
0	0	255	
153	51	255	

Image recognition: human intuitions

- If I gave you a picture of a bird and ask you what this is,
- how can you tell it's a bird?



Image recognition: human intuitions

 We rely on small and specific patterns to classify things

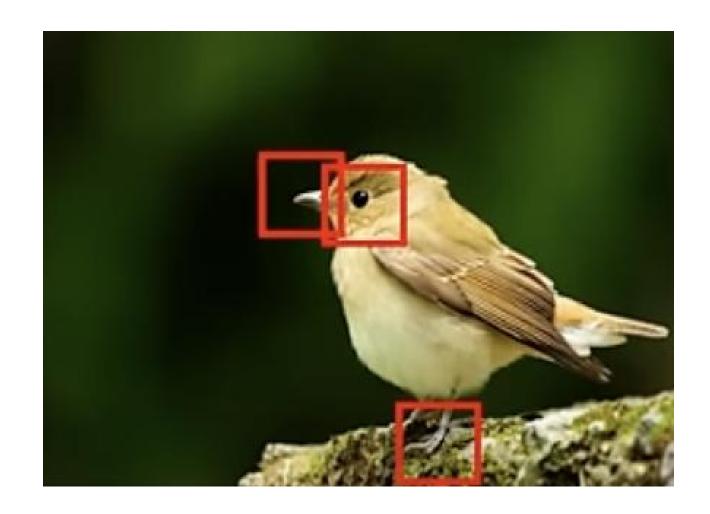


Image recognition: human intuitions

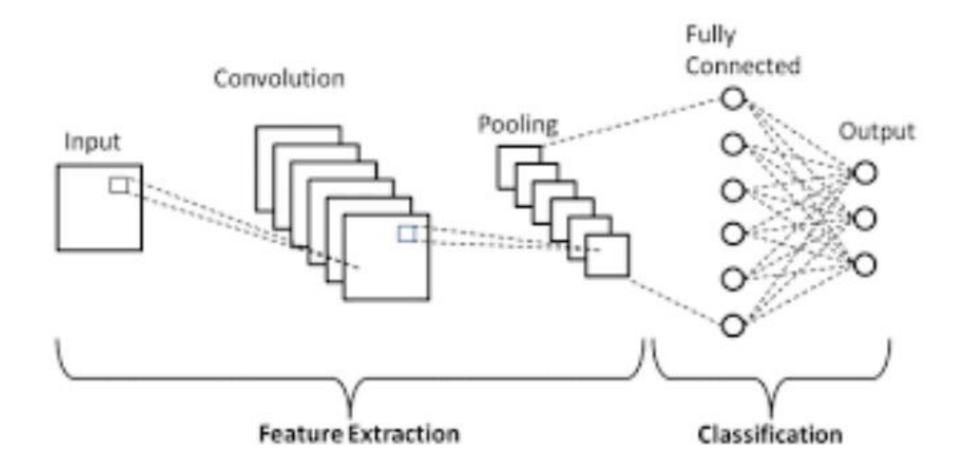
 We rely on small and specific patterns to classify things

• and sometimes, we can fail!



Back to CNNs...

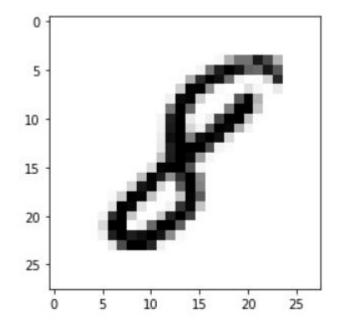
CNN Architecture



The C in CNN: What is a Convolution?

A Function that takes in two parameters

an Image,



and a Kernel

Filter / Kernel

1	0	1
1	1	1
0	0	1

What is a Convolution? (Cont.)

Input

One operation of the convolution is defined like this:

you multiply each value with it's respective kernel value, and add

them all up.

0	1	2		0	1	l	10	25
3	4	5	*	0	'	=	19	25
		Ů		2	3		37	43
6	7	8	'					ш

Kernel

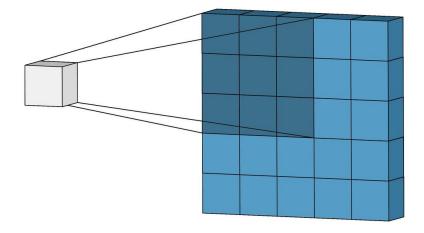
Output

•
$$(0 \times 0) + (1 \times 1) + (3 \times 2) + (4 \times 3) = 19$$

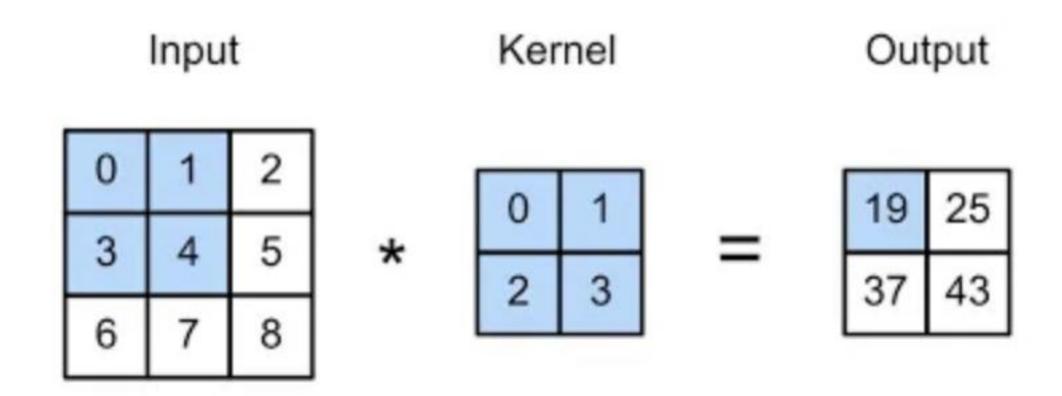
What is a Convolution? (Cont.)

it then "slides" the kernel matrix over the image, and with each position it calculates something for each position. The resulting matrix is the output, called a feature map.

https://poloclub.github.io/cnn-explainer/



Example

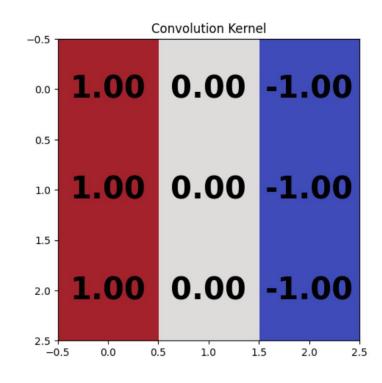


https://medium.com/thedeephub/convolutional-neural-networks-a-comprehensive-guide-5cc0b5eae175

Take this example:

Original Grayscale Image





What does the end result look like?

It looks like this!

Notice that the white parts (lager values) capture the edges where the color changes from white to black in the original image.

Convolved Image



Why is that? Let's take a closer look.

		Imag	е		Kernel					Output		
0	255	255	255	0	1	0	-1		-765			
0	255	255	255	0	1	0	-1					
0	255	255	255	0	1	0	-1					
0	255	255	255	0								
0	255	255	255	0						1		

Why is that? Let's take a closer look.

Image						ı	Kerne	el	Output		
0	255	255	255	0		1	0	-1	-765	0	
0	255	255	255	0		1	0	-1			
0	255	255	255	0		1	0	-1			
0	255	255	255	0							
0	255	255	255	0							
										1	

Why is that? Let's take a closer look.

		Imag	e			Kerne	el .	Output			
0	255	255	255	0	1	0	-1		-765	0	765
0	255	255	255	0	1	0	-1				
0	255	255	255	0	1	0	-1				
0	255	255	255	0						1	
0	255	255	255	0						1	

We see that when on a edge with black on the left and white on the right, this kernel will result in a low value, and vice versa

		Imag	e		Kernel			Output			
0	255	255	255	0	1	0	-1		-765	0	765
0	255	255	255	0	1	0	-1		-765	0	765
0	255	255	255	0	1	0	-1		-765	0	765
0	255	255	255	0						1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
0	255	255	255	0							

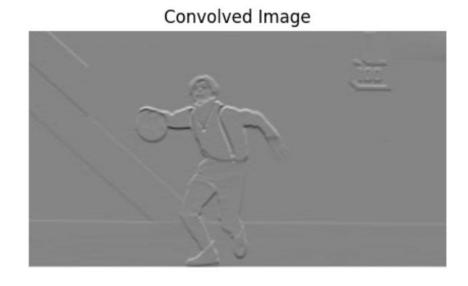
We see that when on a edge with black on the left and white on the right, this kernel will result in a low value, and vice versa That's how its able to detect out the edges!

		Imag	je		Kernel					Output		
0	255	255	255	0	1	0	-1		-765	0	765	
0	255	255	255	0	1	0	-1		-765	0	765	
0	255	255	255	0	1	0	-1		-765	0	765	
0	255	255	255	0						1		
0	255	255	255	0								

Practice

Download and run the ConvolutionFunctionVisualizer notebook.

Try to tweak the kernels to let it capture other patterns like these:





Edge Kun Blurry Kun

From that we can see that by calculating these convolutions, we can extract patterns from pictures, which is precisely what we want.

Pooling: A problem with NNs

Last class we did classification of handwritten digits, where the images are 28x28

what about bigger images?

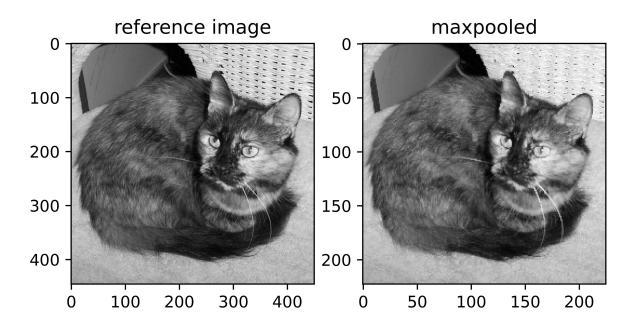
Think of how many connections and neurons it will take!

If only there was a way to make things smaller...

Pooling:

Recall how humans predict a picture

Do both of these pictures look like a cat to you? One of them is actually 4 times smaller



Pooling

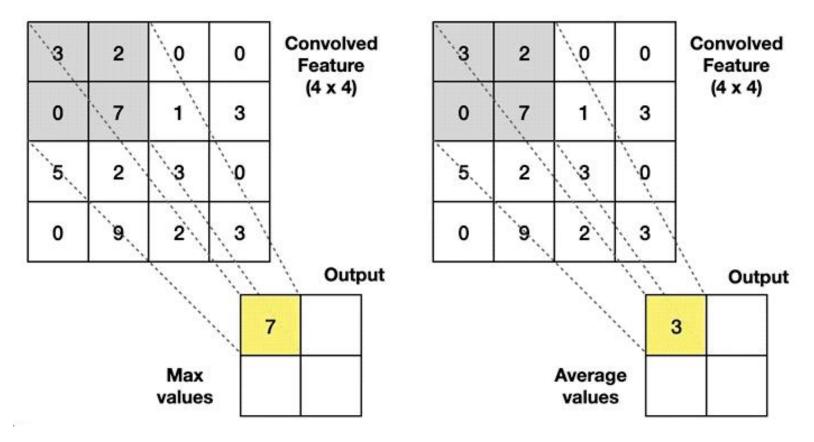
Max Pooling

Take the **highest** value from the area covered by the kernel

Average Pooling

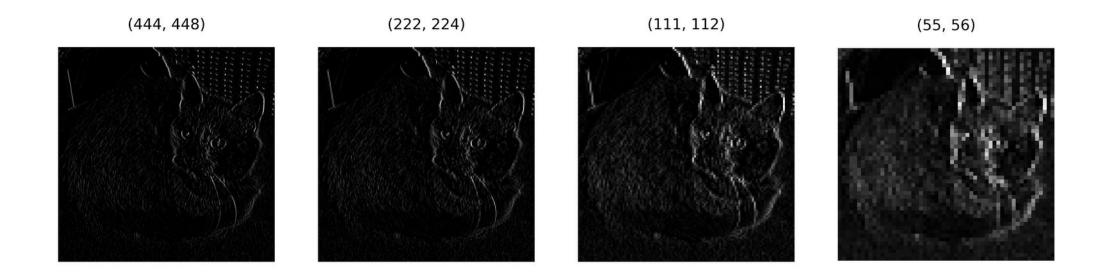
Calculate the average value from the area covered by the kernel

Example: Kernel of size 2 x 2; stride=(2,2)



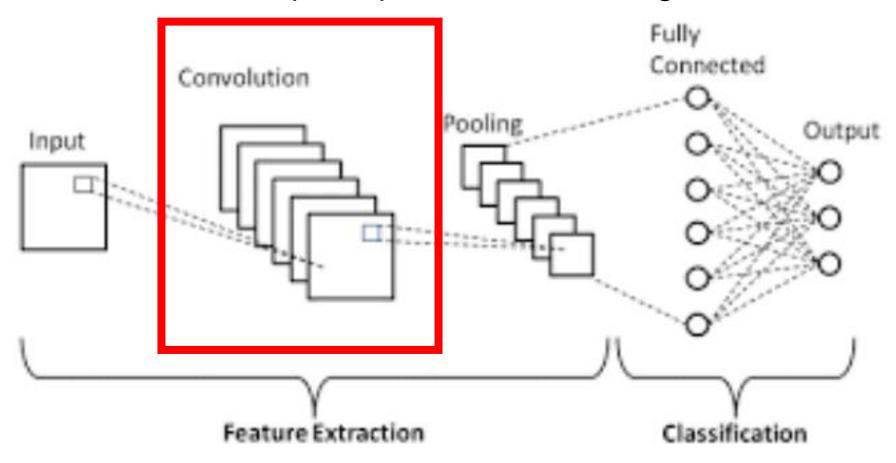
Pooling:

Pooling can actually sometimes amplify features too



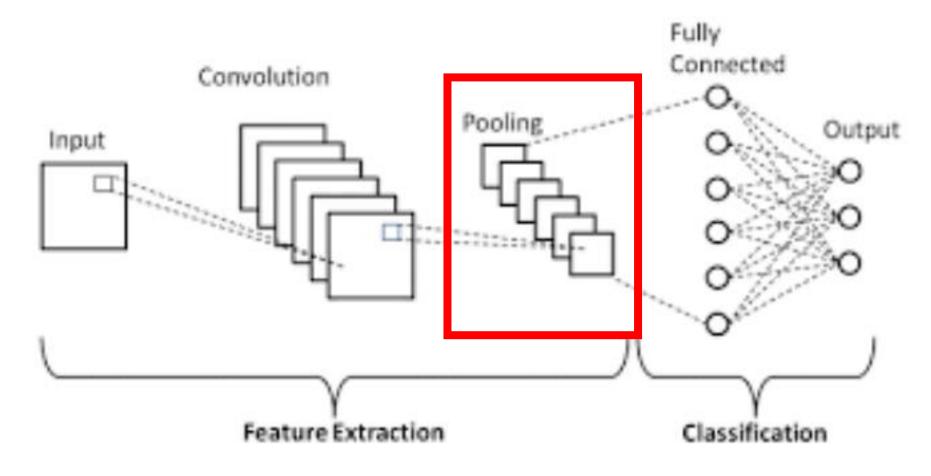
CNN Architecture, revisited

The convolutions capture patterns in the image



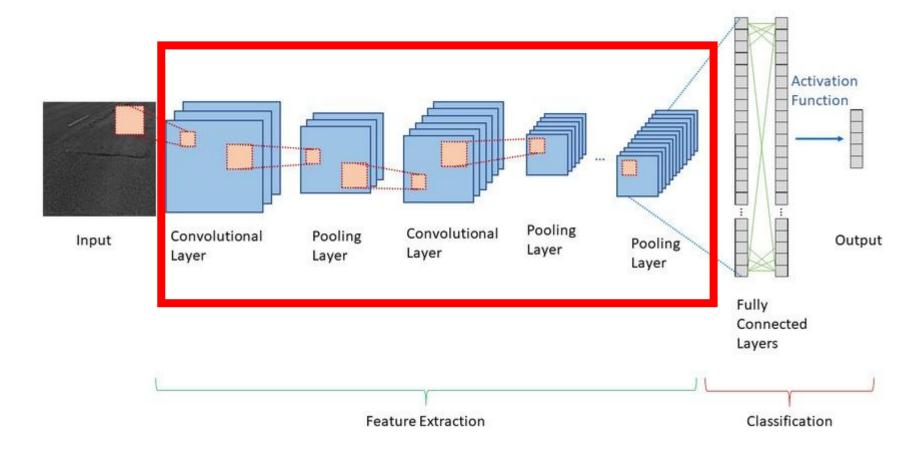
CNN Architecture, revisited

The Pooling part reduces the image size so the network can be smaller



CNN Architecture, revisited

The process of Convoluting and pooling is repeated many times to capture more complex patterns from different perspectives



References

- https://medium.com/thedeephub/convolutional-neuralnetworks-a-comprehensive-guide-5cc0b5eae175
- https://www.digitalocean.com/community/tutorials/pooling-inconvolutional-neural-networks
- https://www.youtube.com/watch?v=OP5HcXJg2Aw
- https://medium.com/@abhishekjainindore24/all-about-convolutions-kernels-features-in-cnn-c656616390a1