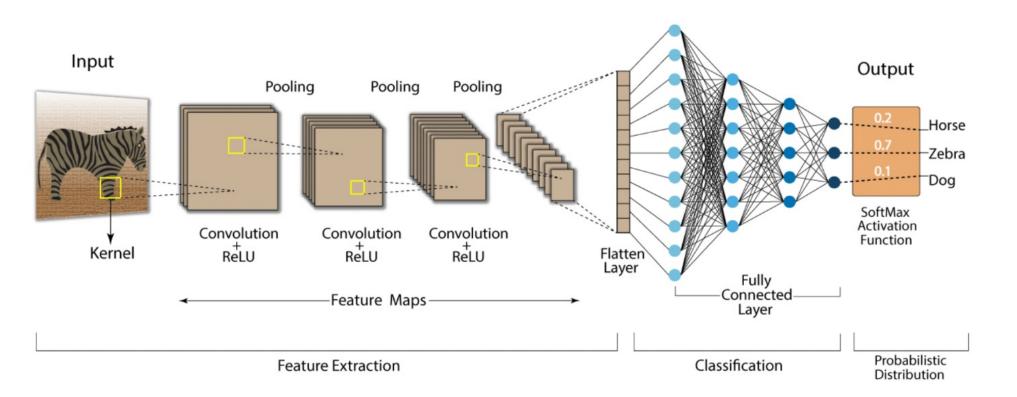
Convolutional Neural Networks

CNN are well suited for images as inputs

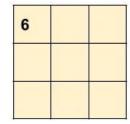


Convolutional Layer

- Image representation by learning the weights of the Conv Layer
- The filter moves around and generates a new representation of the image.
- Edge and shape detection.
- Each filter captures different features of the image.

7	2	3	3	8
4	5	3	8	4
3	3	2	8	4
2	8	7	2	7
5	4	4	5	4

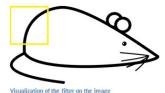
1	0	-1
1	0	-1
1	0	-1



7x1+4x1+3x1+ 2x0+5x0+3x0+ 3x-1+3x-1+2x-1 = 6

Convolutional Layer







Visualization of the receptive field

0	0	0	0	0	0	30
0	0	0	0	50	50	50
0	0	0	20	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0
0	0	0	50	50	0	0

Pixel representation of the receptive field



0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = (50*30)+(50*30)+(50*30)+(50*30)+(50*30)=6600 (A large number!)



Visualization of the filter on the image

 0
 0
 0
 0
 0
 0
 0

 0
 40
 0
 0
 0
 0
 0

 40
 0
 40
 0
 0
 0
 0

 40
 20
 0
 0
 0
 0
 0

 0
 50
 0
 0
 0
 0
 0

 0
 0
 50
 0
 0
 0
 0

 25
 25
 0
 50
 0
 0
 0
 0

Pixel representation of receptive field



0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

Pixel representation of filter

Multiplication and Summation = 0

Convolutional Layer

Padding: 1

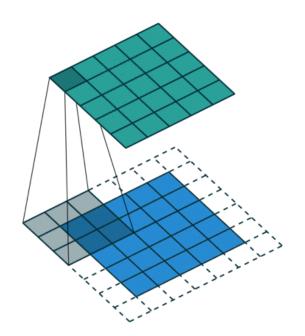
Stride: 1

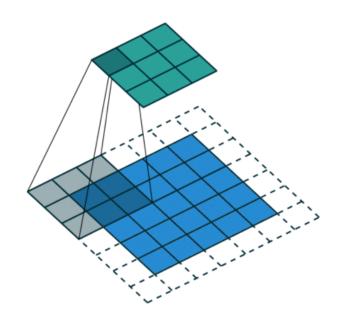
Padding: 1

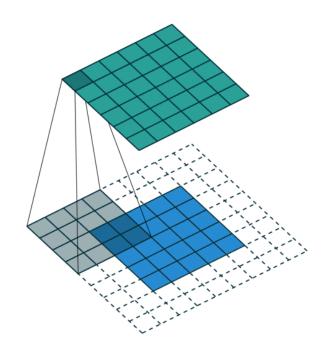
Stride: 2

Padding: 2

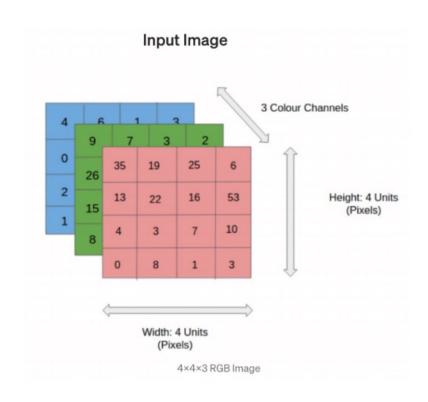
Stride: 1





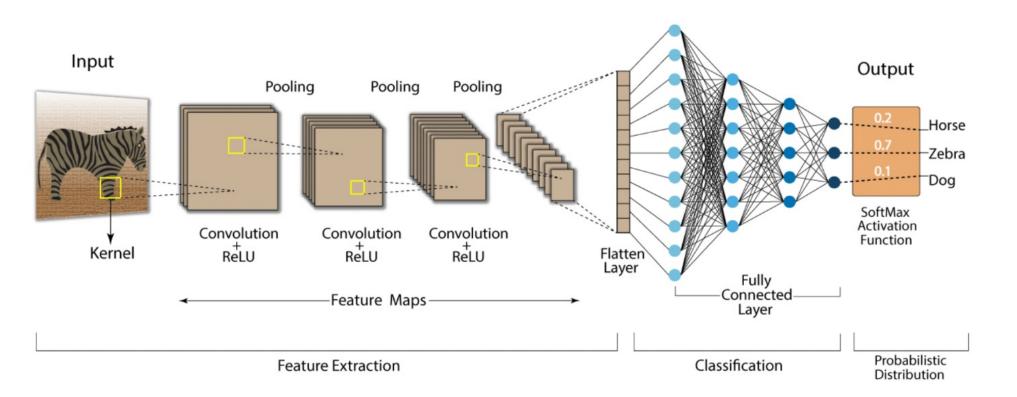


Color images (3 channels)



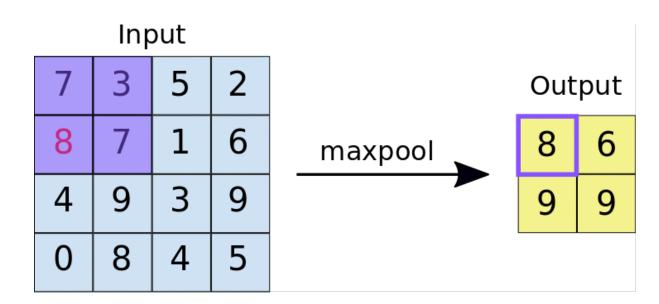
A	Α	В	С	D	E	F	G	Н	1	J	K	L	М	N	0	Р	Q	R	S	Т	U	٧	W	X	Y
1																									
2			1	Inpu	<u>t</u>					<u>k</u>	Cern	<u>el</u>			Inte	rme	diate	Ou	tput						
3																									ļ
4		1	0	1	0	2																			
5		1	1	3	2	1				0	1	0				7	5	3							
6		1	1	0	1	1				0	0	2				4	7	5							
7		2	3	2	1	3				0	1	0				7	2	8							
8		0	2	0	1	0																			
9																						0	utpi	<u>ut</u>	
10		1	0	0	1	0																			
11		2	0	1	2	0				2	1	0				5	3	10				19	13	15	
12		3	1	1	3	0				0	0	0				13	1	13				28	16	20	
13		0	3	0	3	2				0	3	0				7	12	11				23	18	25	
14		1	0	3	2	1																			
15																									
16		2	0	1	2	1																			
17		3	3	1	3	2				1	0	0				7	5	2							
18		2	1	1	1	0				1	0	0				11	8	2							
19		3	1	3	2	0				0	0	2				9	4	6							
20		1	1	2	1	1																			
21																									

CNN are well suited for images as inputs

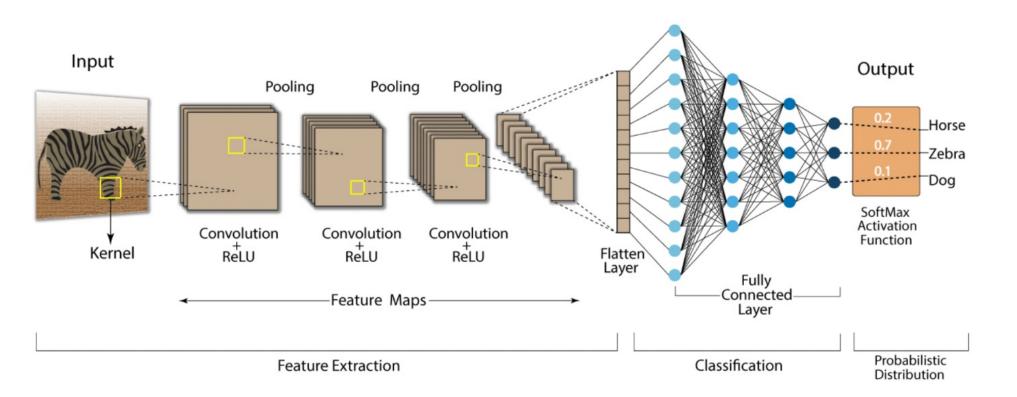


Pooling Layer

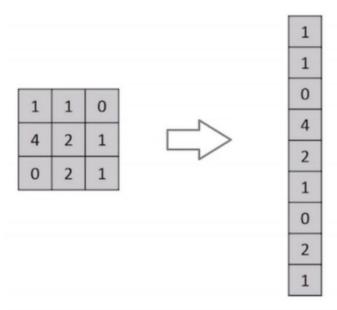
- Decreases dimensionality.
- Highlights relevant features.
- Some sort of 'regularization'



CNN are well suited for images as inputs



Flattening



Flattening of a 3×3 image matrix into a 9×1 vector