

Refer NB

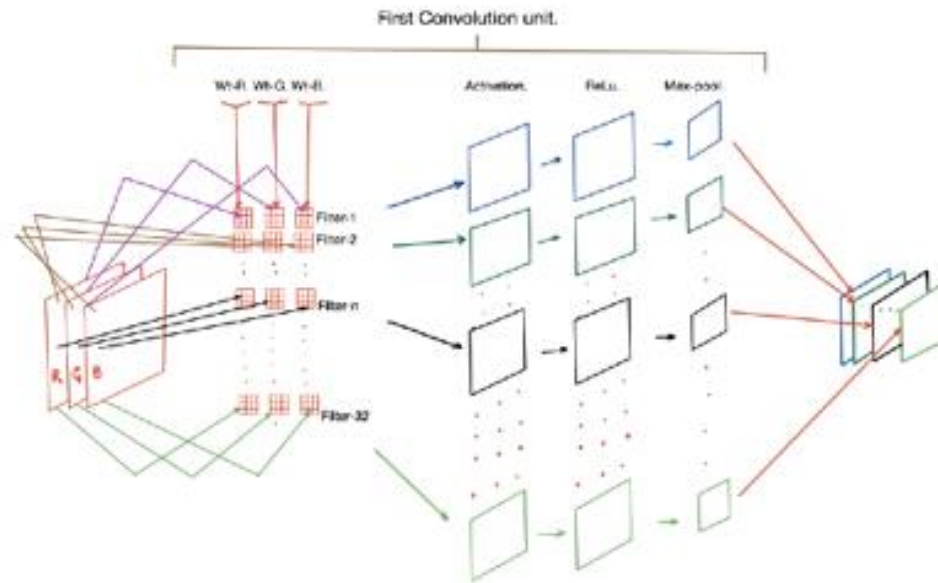
“4 ANN_Regression.ipynb”

DEEP LEARNING – Convolutional Neural Network

Trainer: Dr. Darshan Ingle.

What is Convolution?

- A Convolutional Neural Network is a “Neural Network with Convolution”
- So in order to understand CNN, we must understand Convolution.



Is Convolution Mysterious?

Not at all.

Convolution: EXTC / ELEX engineers.
used for signal processing
& computer vision.

Conv. can be COMPLEX too.

① ADD } You understand
② MULT } CONVOLUTION

Addition	+
Subtraction	-
Multiplication	x
Division	÷



Understanding Convolution without Math

- Lets try to understand Convolution qualitatively by examples.

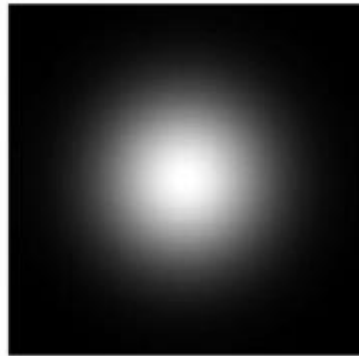
Input Image



*

Filter (Kernel)

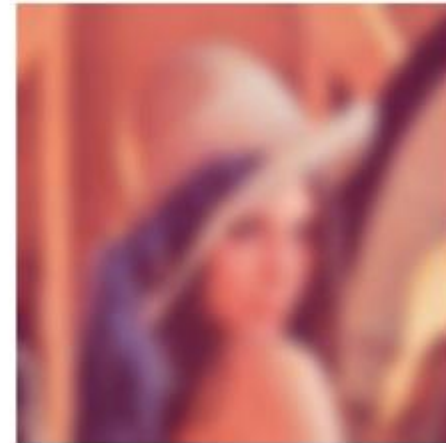
BLURRING



*

=

Output Image



- ① glares
- ① Brown shade
- ② Black shade
- ③ Blue shade
- ④ yellow shade

FILTERS

- ② Instagram filters

Understanding Convolution without Math

EDGE Detection

Input Image

*

Filter (Kernel)

=

Output Image



*

$G_x =$

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

$G_y =$

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



- <https://www.codingame.com/playgrounds/2524/basic-image-manipulation/filtering>
- <https://www.javatpoint.com/dip-high-pass-vs-low-pass-filters>

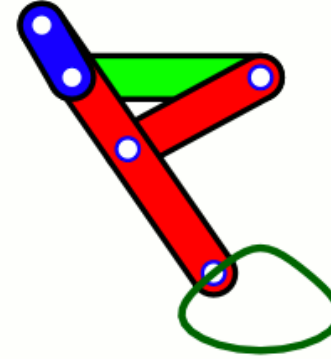
Basically Convolution = Image Modifier

- This is our first perspective on Convolution.
- Eg: it is merely a feature transformation on an image.
- Now the question is “ What make one convolution (blur) different from another (edge detection) ?
- Answer is “The Filter”.
- Next question: How do we design Filter?

How?
A filter = Edge (Blur)
Sharpen,



The Mechanics of Convolution



Image

0	10	10	0
20	30	30	20
10	20	20	10
0	5	5	0

*

Filter

1	0
0	2

The Mechanics of Convolution

Image	*	Filter	=	Output																													
<table><tr><td>0</td><td>10</td><td>10</td><td>0</td></tr><tr><td>20</td><td>30</td><td>30</td><td>20</td></tr><tr><td>10</td><td>20</td><td>20</td><td>10</td></tr><tr><td>0</td><td>5</td><td>5</td><td>0</td></tr></table>	0	10	10	0	20	30	30	20	10	20	20	10	0	5	5	0		<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>2</td></tr></table>	1	0	0	2		<table><tr><td><u>60</u></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	<u>60</u>								
0	10	10	0																														
20	30	30	20																														
10	20	20	10																														
0	5	5	0																														
1	0																																
0	2																																
<u>60</u>																																	

$$\begin{array}{rcl} & 1 \times 0 & 0 \\ + & 0 \times 10 & + 0 \\ + & 0 \times 20 & + 0 \\ + & 2 \times 30 & + 60 \\ & & \hline & & = 60 \end{array}$$

The Mechanics of Convolution

Image				*	Filter		=	Output		
0	10	10	0		1	0		60	70	
20	30	30	20		0	2				
10	20	20	10							
0	5	5	0							

$$\begin{array}{rcl} & 1 \times 10 & 10 \\ + & 0 \times 10 & + 0 \\ + & 0 \times 30 & + 0 \\ + & 2 \times 30 & + 60 \\ & & \hline & & = 70 \end{array}$$

The Mechanics of Convolution

Image	*	Filter	=	Output																													
<table><tr><td>0</td><td>10</td><td>10</td><td>0</td></tr><tr><td>20</td><td>30</td><td>30</td><td>20</td></tr><tr><td>10</td><td>20</td><td>20</td><td>10</td></tr><tr><td>0</td><td>5</td><td>5</td><td>0</td></tr></table>	0	10	10	0	20	30	30	20	10	20	20	10	0	5	5	0		<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>2</td></tr></table>	1	0	0	2		<table><tr><td>60</td><td>70</td><td><u>50</u></td></tr><tr><td></td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	60	70	<u>50</u>						
0	10	10	0																														
20	30	30	20																														
10	20	20	10																														
0	5	5	0																														
1	0																																
0	2																																
60	70	<u>50</u>																															

Filter Movement

L → R
T → B

$$\begin{array}{rcl} & 1 \times 10 & 10 \\ + & 0 \times 0 & 0 \\ + & 0 \times 30 & 0 \\ + & 2 \times 20 & 40 \\ \hline & & = 50 \end{array}$$

The Mechanics of Convolution

Image	*	Filter	=	Output																													
<table><tr><td>0</td><td>10</td><td>10</td><td>0</td></tr><tr><td>20</td><td>30</td><td>30</td><td>20</td></tr><tr><td>10</td><td>20</td><td>20</td><td>10</td></tr><tr><td>0</td><td>5</td><td>5</td><td>0</td></tr></table>	0	10	10	0	20	30	30	20	10	20	20	10	0	5	5	0		<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>2</td></tr></table>	1	0	0	2		<table><tr><td>60</td><td>70</td><td>50</td></tr><tr><td>60</td><td></td><td></td></tr><tr><td></td><td></td><td></td></tr></table>	60	70	50	60					
0	10	10	0																														
20	30	30	20																														
10	20	20	10																														
0	5	5	0																														
1	0																																
0	2																																
60	70	50																															
60																																	

$$\begin{array}{rcl} 1 \times 20 & & 20 \\ + 0 \times 30 & = & + 0 \\ + 0 \times 10 & & + 0 \\ + 2 \times 20 & & + 40 \\ \hline & & = 60 \end{array}$$

The Mechanics of Convolution

Try the rest yourself!

$$\begin{array}{c} \text{Image} \\ \begin{array}{|c|c|c|c|} \hline 0 & 10 & 10 & 0 \\ \hline 20 & 30 & 30 & 20 \\ \hline 10 & 20 & 20 & 10 \\ \hline 0 & 5 & 5 & 0 \\ \hline \end{array} \end{array} * \begin{array}{c} \text{Filter} \\ \begin{array}{|c|c|} \hline 1 & 0 \\ \hline 0 & 2 \\ \hline \end{array} \end{array} = \begin{array}{c} \text{Output} \\ \begin{array}{|c|c|c|} \hline 60 & 70 & 50 \\ \hline 60 & 70 & 50 \\ \hline 20 & 30 & 20 \\ \hline \end{array} \end{array}$$

Note



SQUARE

$$N = 4$$
$$K = 2$$

Image

* Filter

=

Output

0	10	10	0
20	30	30	20
10	20	20	10
0	5	5	0

4x4

1	0
0	2

2x2

60	70	50
60	70	50
20	30	20

3x3

$$\therefore N - K + 1 = 4 - 2 + 1$$
$$= 2 + 1$$
$$= 3$$

Length of Input = 4
Length of filter/kernel = 2
Length of Output = 3

CLAIM:

if $i/p = N$
kernel = K
 $\therefore o/p = N - K + 1$

Note

Given: input_image, kernel

$\text{output_height} = \text{input_height} - \text{kernel_height} + 1$

$\text{output_width} = \text{input_width} - \text{kernel_width} + 1$



Are Images always SQUARE?

No (eg: TV screen, Mobile screen, Laptop screen)

But, Kernels / Filters are always almost SQUARE.

Pseudocode for Convolution



```
Given: input_image, kernel
output_height = input_height - kernel_height + 1
output_width = input_width - kernel_width + 1
output_image = np.zeros((output_height, output_width))
for i in range(0, output_height):
    for j in range(0, output_width):
        for ii in range(0, kernel_height):
            for jj in range(0, kernel_width):
                output_image[i,j] += input_image[i+ii,j+jj] * kernel[ii,jj]
```


Convolution Equation

- This explains how to calculate (i,j) th entry of the output.
- Well TensorFlow does all these things for us automatically. Then why are we studying this?
- Answer: And the answer is that this will help you immensely and understanding the different perspectives on convolution that we are going to discuss later.

Convolution on Wikipedia



- Moreover if you're a curious person and you go on Wikipedia to read about convolution you'll see something very similar to this.

$$Z(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \underbrace{X(m,n)}_{\text{filter}} \underbrace{Y(i-m, j-n)}_{\text{input image}}$$

Handwritten red annotations:

- A 4x4 grid with a small 2x2 grid to its right.
- Red text: $(x+x+y+y)$
- Red text: $+(x+x+y+y)$
- Red text: $+ \dots$
- Red text: $+ \dots$

- In this example you can think of X as the filter y as the input image and Z as the output image.

Convolution on Wikipedia

$$Z(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X(m,n) Y(i \boxed{-} m, j \boxed{-} n)$$

+ or - both are fine as the model ultimately learns the weights using A.D. therefore,

- Now you might notice something weird about this equation from Wikipedia which is that instead of plus signs we have minus signs.
- Why is that?
- Am I wrong?

No. It doesn't matter if the filter is reversed or Not.