

### In Class Activity – Convolutional Neural Network (ICA 19)

Please solve the following problems and enter your responses at <https://tinyurl.com/AIF19-ICA19>

Enter your submissions for each question in row major format. Eg. the image below would be entered as: “3, 1, 4, 2”

Consider the  $2 \times 2$  input image to the left and the filter to the right

3	1
4	2

Image

-1	0
2	1

Filter

1. Use the provided template to complete the image after a single layer of zero padding.
2. Then use the provided template to compute the activation map after convolving the filter from above with stride of 1.
3. Next use the provided template to compute the activation after running the result from (2) through a ReLU activation function (i.e.,  $\text{relu}(x) = 0$  if  $x < 0$ , else  $x$ )
4. Finally, use the provided template compute the activation map after a  $2 \times 2$  max pooling feature with stride of 1.


After zero-padding


After convolution


After ReLU


After max pooling

**Solution:**

1) Add a single layer of padding all around the original image.

0	0	0	0
0	3	1	0
0	4	2	0
0	0	0	0

2) Convolve the filter using a stride of 1:

First place the filter on the image starting at the top left corner, and compute the dot product of overlapping pixels to get the first element of the convolved image.

0	0	0	0
-1	0		
0	3	1	0
2	1		
0	4	2	0
0	0	0	0

-1	0
2	1

=

3		

Dot product:  $0 * -1 + 0 * 0 + 0 * 2 + 3 * 1 = 3$

Next we move the filter by 1 pixel to the right and compute the second dot product:

0	0	0	0
	-1	0	
0	3	1	0
	2	1	
0	4	2	0
0	0	0	0

-1	0
2	1

=

3	7	

Dot product:  $0 * -1 + 0 * 0 + 3 * 2 + 1 * 1 = 7$

Similarly we compute the remaining values by moving the filter to the right until the last column and then we start from the left on the next row.

The final image after convolution is as follows:

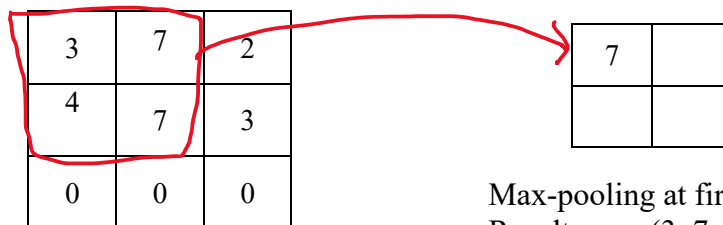
3	7	2
4	7	3
0	-4	-2

3) For each entry in the image from above, we compute the entry after Relu activation by using the original entry if it is greater than 0 and 0 otherwise. Thus the matrix after activation has 0s in the positions where -4, and -2 were in the matrix before activation.

3	7	2
4	7	3
0	0	0

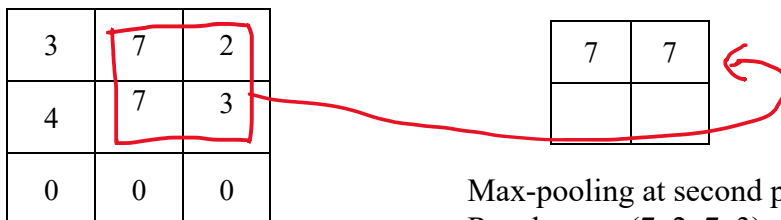
4) For max pooling we employ a similar approach as that of part 2, however here we don't take the dot product, but rather the maximum value of the pixels which overlap with the filter.

Start by placing the max-pool filter at the top left corner:



Max-pooling at first position:  
Result:  $\max(3, 7, 4, 7) = 7$

Next we move the filter by 1 pixel to the right (since the stride is 1):



Max-pooling at second position:  
Result:  $\max(7, 2, 7, 3) = 7$

Thus the final image after max pooling is obtained as:

7	7
7	7