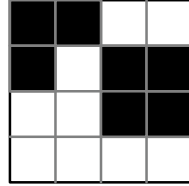


## 6.390 Introduction to Machine Learning

Recitation Week #7

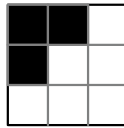
Issued April 3, 2023

1. Consider the following  $4 \times 4$  image:



When referring to a pixel  $(i, j)$  in the image above,  $i$  is the row and  $j$  is the column. Assume that 1-indexing is used, i.e., the upper-left pixel is indexed by  $(1, 1)$  and the lower-left pixel is indexed by  $(4, 1)$ . **Assume dark pixels have a value of 1 and light pixels have a value of -1.**

- (a) Consider what results from filtering this image with this  $3 \times 3$  filter using a stride of 1. Assume that the input image is padded with one layer of zeros, i.e., so that the zero-padded input image is of size  $6 \times 6$  with rows and columns 1 and 6 containing all zeros.



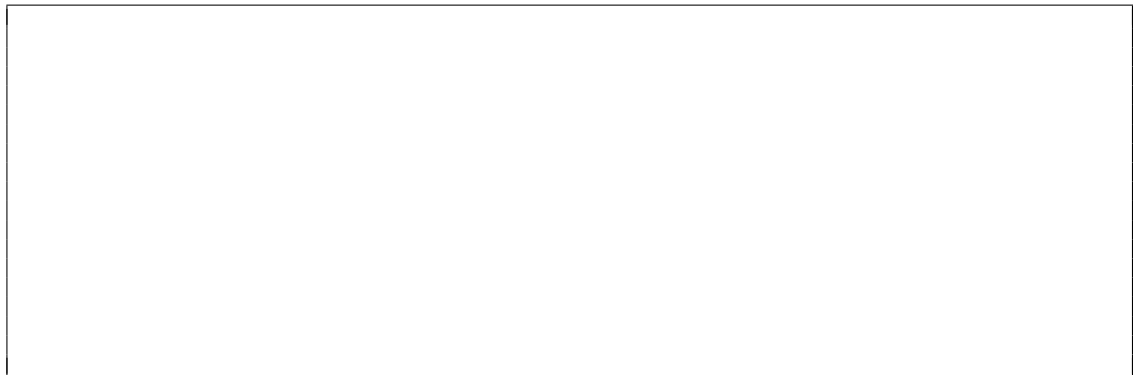
To compute the output value of a particular pixel  $(i, j)$ , apply the filter with its center on pixel  $(i, j)$  of the input image.

- i. What is the output value for the top-left image pixel (that is, the pixel with indices  $(1, 1)$  in one-based indexing)?

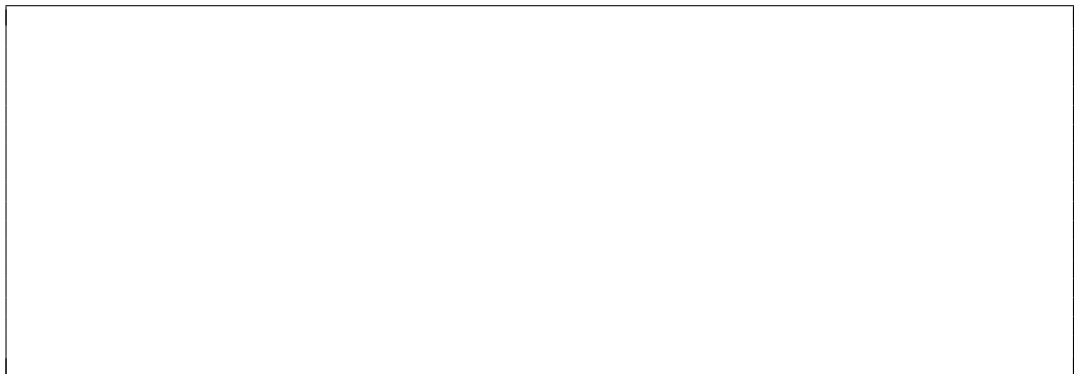
- ii. Which pixel in the output image will have the highest value? Identify this pixel's indices.



- (b) What would the dimensions of the resulting output be with the one  $3 \times 3$  filter? If, instead of that one filter, we used five different filters, each with size  $3 \times 3$  and stride 1 on this image, what would the dimensions of the resulting output be?



- (c) What would be the result of applying max-pooling with size  $2 \times 2$  and stride 2 on the original, unfiltered image above?
- i. What are the dimensions of the resulting image?
- \_\_\_\_\_
- ii. Draw the resulting image with numerical values for each pixel in the space below.



2. Conne von Lucien has many pictures from her trip to Flatland and wants to determine which ones have her or her friends in the image. All of the pictures are matrices of size  $6 \times 6$ , with array values of either 0 or 1. She and her friends look like the following:

$$\text{Conne: } \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \text{ Doni: } \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \text{ Johnny: } \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

If a picture contains any of the patterns above anywhere inside the input image, it should be classified as a positive example; otherwise, the image should be classified as a negative example. Fortunately, you learned about CNNs and have helped Conne by designing the following network architecture with three layers (in this order): a convolutional layer, a max-pooling layer, and a fully connected layer.

- (a) First, you look to specify the convolution layer. Assume that the input image is padded with one layer of zeros, and using a stride of 1. Help us specify each of the following architecture components:

i. Number of filters and the size of each filter:

iii. Output size (assuming the input is a single picture of size  $6 \times 6$ ):

- (b) For each filter, you learn a set of weights and an offset. What are the total number of weight parameters associated with a single filter? What is the total number of weight parameters associated with the entire convolutional layer?

- (c) Suppose that the max pooling layer has size  $3 \times 3$  and stride 3. What are the dimensions of the input and output of the max pooling layer?

- (d) Finally, the goal of the output layer is to classify whether an image contains any of Conne or her friends. What would be a sensible activation to use for the output layer? What is the size of the output?

- (e) What loss function  $\mathcal{L}(h_{CNN}(x^{(i)}), y^{(i)})$  is most appropriate here, for detecting whether or not Conne or any of her friends are present in an image?
- ☐ A. NLL loss
  - ☐ B. 0-1 loss
  - ☐ C. Squared error loss
- (f) What would you need to modify to determine whether Conne, Doni, or Johnny are present in each image *individually*? For example, Conne wants to determine whether she is in an image, but Doni and Johnny are not.