**COMP 642 Assignment --- MODULE 9**

1. Given a one channel gray scale image of size 6 \* 6 shown below, if five 2 \* 2 filters are applied with stride size equal to 2 and no padding to the image, what will be the size of the result? How many trainable parameters will there be in this layer? (each filter has its own bias).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 1 | 2 | 3 | 4 | 2 |
| 2 | 5 | 9 | 7 | 6 | 5 |
| 1 | 2 | 0 | 1 | 5 | 4 |
| 1 | 2 | 2 | 1 | 7 | 0 |
| 6 | 4 | 0 | 1 | 5 | 2 |
| 4 | 3 | 3 | 2 | 5 | 1 |

Image

**The size of the output is equal to the difference between the input size and filter size divided by the stride size plus 1. (6 – 2)/2 + 1 = 3. Which returns a 3x3 filter map of each filter. There are 5 2x2 filters and a bias for each. Which means that there are 5 parameters per filter and a total of 5 filters puts us at 25 total parameters.**

1. If there is padding of one pixel around the image above, what will be the result for the first question?

**A padding of one pixel will increase size by 2 pixels. So, rather than a 6x6 image, we will be working with an 8x8 image. Thus, we would calculate the output size as 8-2/2 + 1 = 4. However, the total number of parameters have not changed because the size and number of filters remain the same.**

1. Apply the following filters with stride equal to 1 to the original image above. What kind of possible feature in the image do you think can be captured by these filters? *(calculations completed in python notebook)*

**With a stride of 1, we will be applying the filters one pixel at a time throughout the entire image and will not skip any pixel. Resulting in a complete mapping of the image.**



|  |  |  |
| --- | --- | --- |
| -1 | -1 | -1 |
| 1 | 1 | 1 |
| -1 | -1 | -1 |

**This is a horizontal edge detector with positive values across the middle row and negative across the top and bottom. This filter responds strongly to edges that have an intensity which leans towards the vertical direction. This will, essentially, call attention to horizontal edges that contain a transition from light to dark.**

array([[ 9., 12., 7., -1.],

[-18., -23., -26., -16.],

[ -8., -3., -2., -10.],

[ -5., -8., -14., -8.]])



|  |  |  |
| --- | --- | --- |
| -1 | 1 | -1 |
| -1 | 1 | -1 |
| -1 | 1 | -1 |

**This is a vertical edge detector containing positive values down the middle column and negative values down the left and right columns. This filter will call attention to vertical edges by highlighting the changes for intensity in the horizontal direction.**

array([[ -7., -8., -15., -7.],

[ -6., -7., -20., 0.],

[ -2., -9., -16., 8.],

[ -7., -8., -18., 10.]])



|  |  |  |
| --- | --- | --- |
| 1 | -1 | -1 |
| -1 | 1 | -1 |
| -1 | -1 | 1 |

**This is a diagonal edge detector which has positive values diagonally, across the matrix, and negative on both sides of the diagonal and will result in an increased attention towards images that have a diagonal contrast transition.**

array([[-11., -8., -9., -11.],

[-12., -17., -4., -12.],

[-12., -3., -10., -6.],

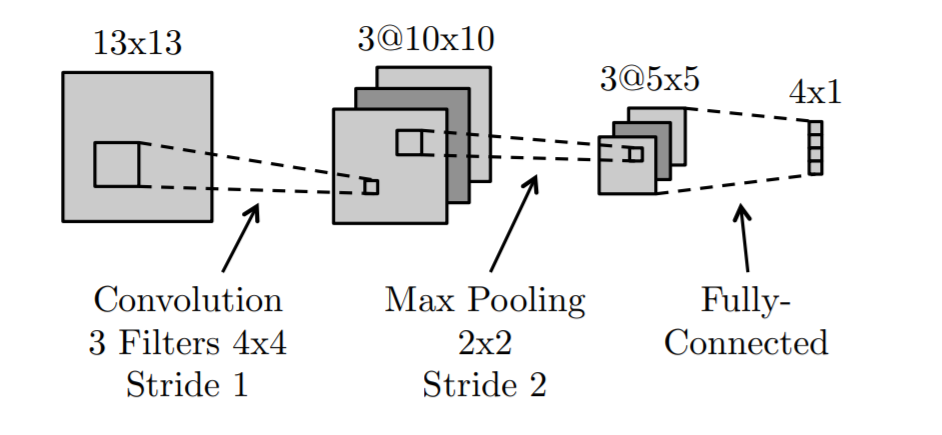
[ -9., -10., -10., -10.]])

1. What will be the result if applying a 3 \* 3 max pooling filter to the original image above (Assume stride of 1)?

**Max pooling will select the max value from within the specified max pooling matrix, here a 3x3 max pooling, then will only shift over by one pixel from that original point due to the stride only being 1. This process is repeated until the entire image has been covered. The result of max pooling the 6x6 image will be the (input size – pooling size / stride) + 1 which in this situation is (6-3)/1 + 1 = 4. Below is the resulting matrix**

|  |  |  |  |
| --- | --- | --- | --- |
| 9 | 9 | 9 | 7 |
| 9 | 9 | 9 | 7 |
| 6 | 4 | 7 | 7 |
| 6 | 4 | 7 | 7 |

1. Below is a diagram of a small convolutional neural network that converts a 13x13 image into 4 output values.



The network has the following layers/operations from input to output: convolution with 3 filters, max pooling, ReLu, and finally a fully-connected layer. There are no bias/offset parameters

1. How many learnable weights are in the convolutional layer?

**With 3 4x4 filters, we have (4x4) x 3 + 3(bias) total learnable weights which gives us 51 total learnable weights.**

1. How many ReLu operations are performed on the forward pass?

**ReLu are applied after convolution to every pixel of the output. With 3 10x10 matrices, we have 3 x 10 x 10 = 300 ReLu operations being performed.**

1. How many learnable weights are there for the entire network?

**From part a. we have 51 learnable weights and also the learnable weights from the max pooling to a 3 x 5 x 5 matrix which is 75 while also connecting to the 4 output neurons; 75 \* 4 = 300. Also, we should include and assume a bias of 1 for each of the outputs which gives us 51 + 300 + 4 = 355 total learnable weights for the entire network.**

1. Would a fully-connected neural network with the same size layers as the above network (13x13 → 3x10x10 → 3x5x5 → 4x1) be able to represent any classifier that the above convolutional network can represent? Explain.

**Yes, the network can represent any function that the convolutional network can represent because of the approximation calculations that it is able to perform on the original matrices. However, spatial hierarchies and shared weights would not be utilized and may require more data and training to gain similar performance.**

1. What is the disadvantage of a fully-connected neural network compared to a convolutional neural network with the same size layers?

**A fully connected neural network requires more parameters. All inputs in a fully connected network has a connection with every unit in the prior layer which increases the number of weights and calculations significantly and increases over fitting. Convolutional neural networks reduce the number of parameters/calculations by sharing weights and local connectivity and result in a more efficient process and greater performance.**

1. Execute hw\_9.ipynb. Apply the sample code to a data set of your choosing and document the results.

Submit a .doc or .pdf with your written answers. Submit your Python notebook. Submit a PDF of your Python notebook.