ML_conv_hw3

April 11, 2019

```
In [0]: # Question 1: Convolution Operations
        import numpy as np


        # Question 1.a
        def ConvID(input_image, H):
            final_size = input_image.shape[0] - H.shape[0] + 1
            H = np.flip(H, axis=0)
            output = np.zeros(final_size)
            for i in range(1, input_image.shape[0] - 1):
                output[i-1] = np.dot(H, input_image[i-1:i+2])

            return output
```

```
In [0]: F = np.array([1, 2, 1, 3, 2, 3, 1, 2, 3, 8, 7, 8, 9, 9, 7, 8])
        W1 = np.array([1, 1, 1])
        W2 = np.array([1, 0, -1])
```

```
In [0]: # Question 1. a.i
        conv1 = ConvID(F, W1)
        print(conv1)
        print(conv1.shape)
```

```
[ 4.  6.  6.  8.  6.  6.  6. 13. 18. 23. 24. 26. 25. 24.]
(14,)
```

```
In [0]: # Question 1. a.ii
        conv2 = ConvID(F, W2)
        print(conv2)
        print(conv2.shape)
```

```
[ 0.  1.  1.  0. -1. -1.  2.  6.  4.  0.  2.  1. -2. -1.]
(14,)
```

```
In [0]: """
        http://www.songho.ca/dsp/convolution/convolution.html
        http://www.songho.ca/dsp/convolution/convolution2d_example.html
        https://github.com/aditya30394/PredictForestFires/blob/master/ML_HW1-KNN.ipynb
        https://docs.scipy.org/doc/numpy/reference/generated/numpy.flip.html
        http://setosa.io/ev/image-kernels/
        """
```

```
Out[0]: '\nhttp://www.songho.ca/dsp/convolution/convolution.html\nhttp://www.songho.ca/dsp/conv
```

```
In [0]: # Question 1. b.i

        def Conv2D(input_image, H):
            final_size = input_image.shape[0] - H.shape[0] + 1
            H = np.rot90(H,2)
            output = np.zeros((final_size, final_size))
            for i in range(1, input_image.shape[0] - 1):
                for j in range(1, input_image.shape[1] - 1):
                    output[i-1][j-1] = np.sum(np.multiply(input_image[i-1:i+2, j-1:j+2],H))

            return output
```

```
In [0]: I = np.array([[164, 188, 164, 161, 195], [178, 201, 197, 150, 137], [174, 168, 181, 190,
        184], [131, 179, 176, 185, 198], [92, 185, 179, 133, 167]])
        F1 = np.array([[1, 1, 1], [1, 1, 1], [1, 1, 1]])
        F2 = np.array([[-1, -2, -1], [0, 0, 0], [1, 2, 1]])
        F3 = np.array([[-1, -1, -1], [-1, 9, -1], [-1, -1, -1]])
```

```
In [0]: # Question 1. b.ii
        print(I)
        print(F1)
        print(Conv2D(I, F1))
```

```
[[164 188 164 161 195]
 [178 201 197 150 137]
 [174 168 181 190 184]
 [131 179 176 185 198]
 [ 92 185 179 133 167]]
[[1 1 1]
 [1 1 1]
 [1 1 1]]
[[1615. 1600. 1559.]
 [1585. 1627. 1598.]
 [1465. 1576. 1593.]]
```

```
In [0]: # Question 1. b.ii
        print(I)
        print(F2)
        print(Conv2D(I, F2))
```

```
[[164 188 164 161 195]
 [178 201 197 150 137]
 [174 168 181 190 184]
 [131 179 176 185 198]
 [ 92 185 179 133 167]]
[[-1 -2 -1]
 [ 0  0  0]
 [ 1  2  1]]
[[  13.  -43.  -64.]
 [ 112.   29. -110.]
 [  50.   44.  133.]]
```

```
In [0]: # Question 1. b.ii
        print(I)
        print(F3)
        print(Conv2D(I, F3))
```

```
[[164 188 164 161 195]
 [178 201 197 150 137]
 [174 168 181 190 184]
 [131 179 176 185 198]
 [ 92 185 179 133 167]]
[[-1 -1 -1]
 [-1  9 -1]
 [-1 -1 -1]]
```

```
[[395. 370. -59.]
 [ 95. 183. 302.]
 [325. 184. 257.]]
```