CIS 4930/CIS 5930 Computer Vision
Homework 3, 40 points
Due: November 2nd

Part A: ConvNet Operations [40 pts]

Image

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	0	0	0
1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	1	0
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

Filter

0	-1	-1
1	0	-1
1	1	0

In this part, you will compute the output from applying a single set of convolution, non-linearity, and pooling operations, on two small examples. Below are your image (with width = height = N = 9) and your filter (with width = height = F = 3).

1. [15 pts] Write a `function [Output] = my_conv(Image, Filter, Padding, Stride)` that computes the output of applying a filter over an image, with given padding and stride. You are not allowed to use any convolution-related Matlab functions except element-wise multiplication between two matrices, followed by summation. You are allowed to implement correlation instead of convolution as a simplification.
   Inputs:
   - `Image` is a grayscale single-channel image, e.g. the one shown above (don't hard-code it in your function).
   - `Filter` is a single-channel filter, e.g. the one shown above (don't hard-code it in your function).
   - `Padding` is a scalar saying how much padding to apply above/below and to the left/right of the Image.
   - `Stride` is a scalar saying what stride to use to advance over the Image during convolution.

   Output:

   - `Output` is the single-channel matrix resulting from the convolution operation.

2. [15 pts] Write a `function [Output] = my_pool(Input, Pool_Size)` that computes the output of non-overlapping max-pooling over *Pool_Size*x*Pool_Size* regions of the input. Again, you are not allowed to use built-in Matlab functions that compute pooling.
   Inputs:
   - `Input` is a square matrix, which you should assume to be the result from applying RELU on the output from convolution.
   - `Pool_Size` is a scalar saying over what size of regions to compute max (e.g. use 2 for pooling over 2x2 regions).

   Output:

   - `Output` is the single-channel matrix resulting from the max-pooling operation.

3. A script `test_cnn_ops.m` is provided in the folder to test your two functions in two scenarios. When you run it, it will encode the example image and filter above, call your functions (conv -> relu -> pool) and save the output variables `output1`, `output2` in a file `outputs.mat`. Run this script and submit the saved output file. The tests in the script are as follows:
   - [5 pts] Test 1:
     - First, apply convolution using no padding, and a stride of 2 (in both the horizontal and vertical directions).
     - Second, apply a Rectified Linear Unit (ReLU) activation on the previous output.
     - Third, apply max pooling over 2x2 regions on the previous output.
   - [5 pts] Test 2:
     - First, apply convolution using padding 1, and a stride of 4.
     - Second, apply a Rectified Linear Unit (ReLU) activation on the previous output.
     - Third, apply max pooling over 3x3 regions on the previous output.

Submissions

Part A

- `my_conv.m`
- `my_pool.m`
- `outputs.mat`

Acknowledgment: This assignment was partially adapted from Adriana Kovashka's original assignment.