CIS 419/519: Applied Machine Learning	Spring 2023
Homework 5	
Handed Out: March 15	*Due: March 29, 7:59 p.m.*

- **Your Name:** *Qihang Dai*
- **Your PennKey:** *ahgdyycc*
- **Your PennID:** *78803164*

1 Multiple Choice & Written Questions

1. (a) choose B. this filter gonna average one pixel with the rest 8 pixels around it.
   (b) this filter will double the value of each pixel, thus the image will be brighter.
   (c) this filter tends to detect vertical edges. If in the receptive field of this filter, left side and right side has a large difference, it will be amplified by this filter and thus detected as an edge.

2.
$$p1 = 0 * 0.9 - 0.1 * 1 = -0.1$$
$$w1 = w1 + p = 0.9$$

$$p2 = -0.1 * 0.9 - 0.1 * 0 = -0.09$$
$$w2 = w1 + p = 0.81$$

$$p3 = -0.09 * 0.9 - 0.1 * 1 = -0.189$$
$$w3 = w2 + p = 0.621$$

3.
$$out = (in + 2 * padding - kernelSize)/stride + 1$$

step a: Conv. out = (232 - 5)/1 + 1 = 228. dimension: (228, 228, 5)

step b: Relu. no change

step c: MaxPool. out = (228 - 2)/2 + 1 = 114. dimension: (114, 114, 5)

step d: Conv. out = (114 - 3)/1 + 1 = 112. dimension: (112, 112, 10)

step e: Relu. no change

step f: MaxPool. out = (112 - 2)/2 + 1 = 56. dimension: (56, 56, 10)

step g: Conv. out = (56 - 3)/1 + 1 = 54. dimension: (54, 54, 20)

step h: Relu. no change

step i: MaxPool. out = (54 - 2)/2 + 1 = 27. dimension: (27, 27, 20)

number of parameters: there is no fully connected layer. pool and input layer have zero parmeter. Thus calculate the total number of parameters in each convolutional layer.

the equation is: ((shape of width of the filter * shape of height of the filter * number of filters in the previous layer+1)*number of filters).

$$conv_a = (5 * 5 * 3 + 1) * 5 = 380 \quad (1)$$
$$conv_d = (3 * 3 * 5 + 1) * 10 = 460 \quad (2)$$
$$conv_g = (3 * 3 * 10 + 1) * 20 = 1820 \quad (3)$$
$$totalParameters = 380 + 460 + 1820 = 2650 \quad (4)$$

4.

$$W_1 = \begin{bmatrix} w_{11} & w_{21} \\ w_{12} & w_{22} \end{bmatrix} \quad (5)$$
$$= \begin{bmatrix} 0.1 & 0.2 \\ -0.4 & 0.3 \end{bmatrix} \quad (6)$$
$$W_2 = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} \quad (7)$$
$$y_1 = w_{11} * x_1 + w_{21} * x_2 = 0.1 * 10 + 0.2 * 8 = 2.6 \quad (8)$$
$$y_2 = w_{12} * x_1 + w_{22} * x_2 = -0.4 * 10 + 0.3 * 8 = -1.6 \quad (9)$$
$$Y = [y_1, y_2] = [2.6, -1.6] \quad (10)$$
$$output = W2_1 * y_1 + W2_2 * y_2 = 0.1 * 2.6 + 0.2 * -1.6 = 0.06 \quad (11)$$
$$L(output) = \sigma(output) = \frac{1}{1 + e^{-0.06}} = 0.5149 \quad (12)$$
$$\frac{\partial L}{\partial output} = \frac{1}{1 + e^{-0.06}} * (1 - \frac{1}{1 + e^{-0.06}}) = 0.249 \quad (13)$$
$$\frac{\partial output}{\partial W_2} = relu([y_1, y_2]) = [2.6, 0] \quad (14)$$
$$(15)$$

by chain rule:

$$\frac{\partial L}{\partial W_2} = \frac{\partial L}{\partial output} * \frac{\partial output}{\partial W_2} = 0.249 * [2.6, 0] = [0.64, 0] \quad (16)$$

$$\frac{\partial output}{\partial relu([y_1, y_2])} = W_2 = [0.1, 0.2] \tag{17}$$

$$\frac{\partial L}{\partial y1} = \frac{\partial L}{\partial output} * \frac{\partial output}{\partial relu(y1)} * \frac{\partial relu(y1)}{\partial y1} = 0.249 * 0.1 * 1 = 0.0249 \tag{18}$$

$$\frac{\partial L}{\partial y2} = \frac{\partial L}{\partial output} * \frac{\partial output}{\partial relu(y2)} * \frac{\partial relu(y2)}{\partial y2} = 0.249 * 0.2 * 0 = 0 \tag{19}$$

$$\frac{\partial L}{\partial Y} = [\frac{\partial L}{\partial y1}, \frac{\partial L}{\partial y2}] = [0.0249, 0] \tag{20}$$

then we can calcualte the gradient of loss with respect to the weight of the first layer W1:

$$\frac{\partial L}{\partial W_1} = \frac{\partial L}{\partial Y} * \frac{\partial Y}{\partial W_1} = [0.0249, 0] * [x1, x2]^T = \begin{bmatrix} 0.0249 * 10 & 0.0249 * 8 \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.249 & 0.1992 \\ 0 & 0 \end{bmatrix} \tag{21}$$

5. (a) constraint(i):

$$C_h \wedge C_c = 0$$

$$max(0, C_h + C_c - 1) = 0$$

constraint(ii):

$$C_h \wedge \neg C_m = 0$$

$$max(0, C_h + \neg C_m - 1) = max(0, C_h + 1 - C_m - 1) = max(0, C_h - C_m) = 0$$

constraint(iii):

$$C_c \wedge \neg C_m = 0$$

$$max(0, C_c - C_m) = 0$$

(b) not really need an is function. t-norm is already within [0, 1] range.

(c) to satisfy the inital three constraints, we set all target value to zero. Then we use the constraint value and the zero to compute MSE. Thus

$$L_i = max(0, C_h + C_c - 1)^2$$

$$L_i i = max(0, C_h - C_m)^2$$

$$L_i ii = max(0, C_c - C_m)^2$$