ECE 661: Homework #1
Linear Model, Back Propagation and Building a CNN

Student NetID: wh162

1. True/False Questions
   i Problem 1.1: True, the set of the weight was trained under different guesses and thus there's no guarantee that the exact same set of weight can be found even if we control all the conditions.
   ii Problem 1.2: False, Latency is the delay time that takes for the data to go through the mode, so it's positively relative to our processor.
   iii Problem 1.3: True, the gradient will no vanish even when $x \to \infty$, making the model to be able to know what's the gradient.
   iv Problem 1.4: True, convolution layer has less parameters compare to FC.
   v Problem 1.5: True.
2. Adalines
   i Problem 2.1: Logic AND function

x1	x2	s	y
-1	-1	-5	-1
-1	+1	-3	-1
+1	-1	-1	-1
+1	+1	+1	+1

   ii Problem 2.2: $w_0 = -1$, $w_1 = -1$, $w_2 = -1$

x1	x2	s	y
-1	-1	-1	+1
-1	+1	-1	-1
+1	-1	-1	-1
+1	+1	-3	-1

   iii Problem 2.3: $w_0 = 0$, $w_1 = 1$, $w_2 = 1$, $w_3 = 1$

X1	X2	X3	s	y
-1	-1	-1	-3	-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	+1	-1	+1	+1
+1	+1	+1	+3	+1

iv Problem 2.4: $w_{20} = -2$, $w_{21} = 2$, $w_{22} = -1$

x1	x2	s	y
-1	-1	-3	-1
-1	+1	+1	+1
+1	-1	+1	+1
+1	+1	-1	-1

3. Back Propagation

i Problem 3.1:

$$X_2 = \mathrm{ReLU}(W_1 X_1 + b_1)$$
$$X_3 = W_2 X_2 + b_2$$
$$L = \tfrac{1}{2}(t - X_3)^T (t - X_3)$$

$t = k\times 1$, $X_2 = m\times 1$, $W_2 = k\times m$, $W_1 = m\times n$, $X_3 = k\times 1$, $b_1 = m\times 1$, $X_1 = n\times 1$, $b_2 = k\times 1$

$$\boxed{\frac{\partial L}{\partial W_1}} = \frac{\partial L}{\partial X_3} \cdot \frac{\partial X_3}{\partial X_2} \cdot \frac{\partial X_2}{\partial W_1}$$

$$= \frac{1}{2}\Big[\underbrace{-(t-X_3)}_{k\times 1} \; \underbrace{-(t-X_3)^T}_{1\times k}\Big] \cdot \underbrace{W_2}_{k\times m} \cdot \mathrm{ReLU} \cdot \underbrace{X_1}_{n\times 1}$$

$m\times n$

$$= -\frac{1}{2} W_2 X_1 [\mathrm{ReLU}] \cdot (t - X_3) - \frac{1}{2} W_2 X_1 [\mathrm{ReLU}] \cdot (t - X_3)^T$$

need to match output

$$\boxed{= \begin{cases} -W_2^T (t - X_3) X_1^T, & W_1 X_1 + b > 0 \\ 0 & W_1 X_1 + b < 0 \end{cases}}$$ #

$$X_2 = \mathrm{ReLU}(W_1 X_1 + b_1)$$
$$X_3 = W_2 X_2 + b_2$$
$$L = \tfrac{1}{2}(t - X_3)^T (t - X_3)$$

$t = k\times 1$, $X_2 = m\times 1$, $W_2 = k\times m$, $W_1 = m\times n$, $X_3 = k\times 1$, $b_1 = m\times 1$, $X_1 = n\times 1$, $b_2 = k\times 1$

$$\boxed{\frac{\partial L}{\partial W_2}} = \frac{\partial L}{\partial X_3} \cdot \frac{\partial X_3}{\partial W_2}$$

$k\times m$

$$= \frac{1}{2}\Big[\underbrace{-(t-X_3)}_{k\times 1} \; \underbrace{-(t-X_3)^T}_{1\times k}\Big] \cdot \underbrace{X_2}_{m\times 1}$$

$$\boxed{= \begin{cases} -(t - X_3) \cdot X_2^T, & (W_1 X_1 + b_1) > 0 \\ 0, & (W_1 X_1 + b_1) < 0 \end{cases}}$$

$$X_2 = \mathrm{ReLU}(W_1 X_1 + b_1)$$
$$X_3 = W_2 X_2 + b_2$$
$$L = \tfrac{1}{2}(t - X_3)^T (t - X_3)$$

$t = k\times 1$, $X_2 = m\times 1$, $W_2 = k\times m$, $W_1 = m\times n$, $X_3 = k\times 1$, $b_1 = m\times 1$, $X_1 = n\times 1$, $b_2 = k\times 1$

$$\boxed{\frac{\partial L}{\partial b_1}} = \frac{\partial L}{\partial X_3} \cdot \frac{\partial X_3}{\partial X_2} \cdot \frac{\partial X_2}{\partial b_1}$$

$m\times 1$

$$= \frac{1}{2}\Big[\underbrace{-(t-X_3)}_{k\times 1} \; \underbrace{-(t-X_3)^T}_{1\times k}\Big] \cdot \underbrace{W_2}_{k\times m} \cdot$$

$$\boxed{= \begin{cases} -W_2^T \cdot (t - X_3), & W_1 X_1 + b_1 > 0 \\ 0, & W_1 X_1 + b_1 < 0 \end{cases}}$$

$$X_2 = \mathrm{ReLU}(W_1 X_1 + b_1)$$
$$X_3 = W_2 X_2 + b_2$$
$$L = \tfrac{1}{2}(t - X_3)^T (t - X_3)$$

$t = k\times 1$, $X_2 = m\times 1$, $W_2 = k\times m$, $W_1 = m\times n$, $X_3 = k\times 1$, $b_1 = m\times 1$, $X_1 = n\times 1$, $b_2 = k\times 1$

$$\boxed{\frac{\partial L}{\partial b_2}} = \frac{\partial L}{\partial X_3} \cdot \frac{\partial X_3}{\partial b_2}$$

$k\times 1$

$$= \frac{1}{2}\Big[\underbrace{-(t-X_3)}_{k\times 1} \; \underbrace{-(t-X_3)^T}_{1\times k}\Big] \cdot 1$$

$$\boxed{= \begin{cases} -(t - X_3), & W_1 X_1 + b_1 > 0 \\ 0, & W_1 X_1 + b_1 < 0 \end{cases}}$$

ii Problem 3.2:

$$W_1 = \begin{bmatrix} 2 & -1 & 1 \\ -3 & 2 & -1 \end{bmatrix},\ W_2 = \begin{bmatrix} 1 & -2 \\ -3 & 1 \end{bmatrix},\ b_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix},\ b_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}\quad x_1 = [0\ 1\ 1]^T,\ t = [1, 1]^T$$

$$W_1 \cdot X_1 + b_1 = \begin{bmatrix} 2 & -1 & 1 \\ -3 & 2 & -1 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \end{bmatrix} = X_2$$

$$W_2 \cdot X_2 + b_2 = \begin{bmatrix} 1 & -2 \\ -3 & 1 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} -4 \\ 1 \end{bmatrix} = X_3$$

$$\partial L / \partial W_1 = -W_2^T (t - X_3) X_1^T$$
$$= \begin{bmatrix} -1 & 3 \\ 2 & -1 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 0 \end{bmatrix} \cdot [0, 1, 1] = \begin{bmatrix} 0 & -5 & -5 \\ 0 & 10 & 10 \end{bmatrix}$$

$$\partial L / \partial W_2 = -(t - X_3) \cdot X_2^T = \begin{bmatrix} -5 \\ 0 \end{bmatrix}[1\ 3] = \begin{bmatrix} -5 & -15 \\ 0 & 0 \end{bmatrix}$$

$$\partial L / \partial b_1 = -W_2^T (t - X_3) = \begin{bmatrix} -1 & 3 \\ 2 & -1 \end{bmatrix}\begin{bmatrix} 5 \\ 0 \end{bmatrix} = \begin{bmatrix} -5 \\ 10 \end{bmatrix}$$

$$\partial L / \partial b_2 = -(t - X_3) = \begin{bmatrix} -5 \\ 0 \end{bmatrix}$$

$$L = \frac{1}{2}(t - X_3)^T (t - X_3) = \frac{1}{2}[5\ 0]\begin{bmatrix} 5 \\ 0 \end{bmatrix} = 12.5$$

4. 2D Convolution:

i Problem 4.1:

$$\text{input map} \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix} \text{padding}$$

$$\text{Kernel} \begin{bmatrix} 1/32 & 1/8 & 1/32 \\ 1/8 & 1/2 & 1/8 \\ 1/32 & 1/8 & 1/32 \end{bmatrix} \qquad 16+16+2 = 34$$

$$\text{output map} = \frac{1}{32}\begin{bmatrix} 0 & 1 & 5 & 10 & 22 & 10 & 5 & 1 & 0 \\ 5 & 10 & 26 & 31 & 34 & 31 & 26 & 10 & 5 \\ 20 & 26 & 34 & 36 & 36 & 36 & 34 & 26 & 20 \\ 5 & 10 & 26 & 31 & 34 & 31 & 26 & 10 & 5 \\ 0 & 1 & 5 & 10 & 22 & 10 & 5 & 1 & 0 \end{bmatrix}$$

ii Problem 4.2: After the shifting of the 3x3 kernel, we can see that not only did the “1” numbers decreased, but also its neighbor was added to some degree of noise. If this kernel were applied to an image, the result would be that the edge will be dilate and the image would turn vague.

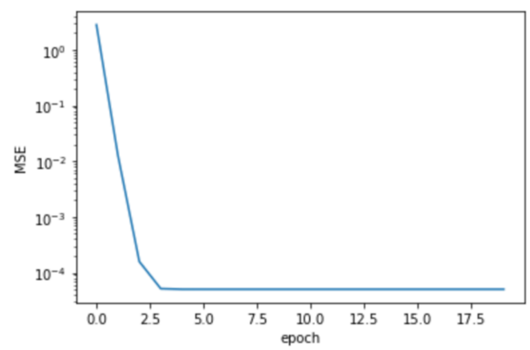
5. Lab: LMS Algorithms:

   i Problem 5.1: W^* = [[1.0006781][1.00061145][-2.00031968]] , MSE = 5.03995157e-05

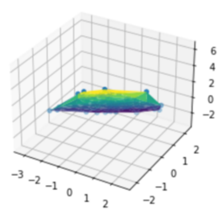
   ii Problem 5.2:

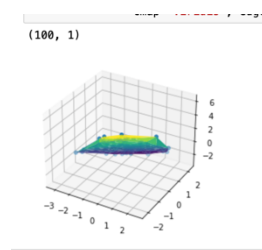
```
1 epoch,  MSE: 2.812639199865521
2 epoch,  MSE: 0.012910585982472512
3 epoch,  MSE: 0.00015834190092198875
4 epoch,  MSE: 5.165077987103672e-05
5 epoch,  MSE: 5.041534974744646e-05
6 epoch,  MSE: 5.0399719872042845e-05
7 epoch,  MSE: 5.0399518304804874e-05
8 epoch,  MSE: 5.039951569304508e-05
9 epoch,  MSE: 5.03995156591307e-05
10 epoch,  MSE: 5.039951565868916e-05
11 epoch,  MSE: 5.0399515658683576e-05
12 epoch,  MSE: 5.039951565868373e-05
13 epoch,  MSE: 5.039951565868364e-05
14 epoch,  MSE: 5.0399515658683454e-05
15 epoch,  MSE: 5.0399515658683705e-05
16 epoch,  MSE: 5.039951565868355e-05
17 epoch,  MSE: 5.0399515658683576e-05
18 epoch,  MSE: 5.039951565868371e-05
19 epoch,  MSE: 5.039951565868371e-05
20 epoch,  MSE: 5.039951565868371e-05
```

```
Out[126]: [<matplotlib.lines.Line2D at 0x7fb3919e6a10>]
```



   iii Problem 5.3: (a) (b)





   iv Problem 5.4:

6. Lab: Simple NN

   i (a)

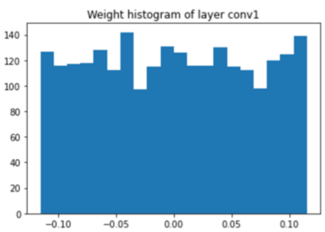
```python
class SimpleNN(nn.Module):
    def __init__(self):
        super(SimpleNN, self).__init__()
        # Layer definition
        self.conv1 = CONV(3, 32, 5, stride = 1, padding = 2) #Your code here
        self.conv2 = CONV(32, 32, 5, stride = 1, padding = 2) #Your code here
        self.conv3 = CONV(32, 64, 5, stride = 1, padding = 2)#Your code here
        self.fc1   = FC(576, 64) #Your code here
        self.fc2   = FC(64, 10) #Your code here

    def forward(self, x):
        out = self.conv1(x)
        out = F.relu(out)
        out = F.max_pool2d(out, 3, stride = 2)
        out = self.conv2(out)
        out = F.relu(out)
        out = F.max_pool2d(out, 3, stride = 2)
        out = self.conv3(out)
        out = F.relu(out)
        out = F.max_pool2d(out, 3, stride = 2)
        out = torch.flatten(out, 1)
        out = self.fc1(out)
        out = F.relu(out)
        out = self.fc2(out)
        out = F.relu(out)
        return out
```

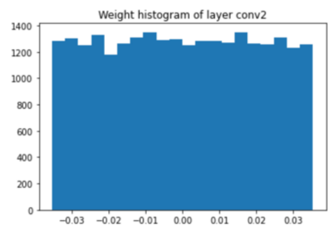
ii (b)

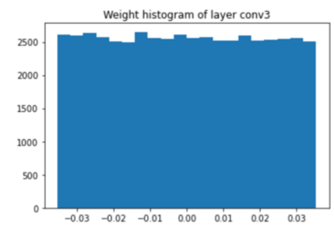
Layer	Input shape	Output shape	Weight shape	# Param	# MAC
Conv 1	(1,3,32,32)	(1,32,32,32)	(32,3,5,5)	2400	2457600
Conv 2	(1,32,15,15)	(1,32,15,15)	(32,32,5,5)	25600	54000
Conv 3	(1,32,7,7)	(1,64,7,7)	(64,32,5,5)	51200	2508800
FC1	(1,576)	(1,64)	(64,576)	36928	36864
FC2	(1,64)	(1,10)	(10,64)	650	640

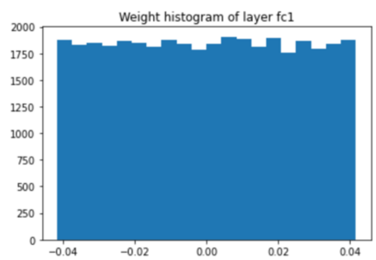
7. Lab3

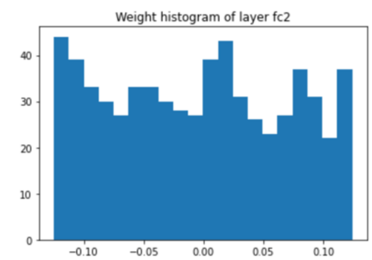
i Bonus



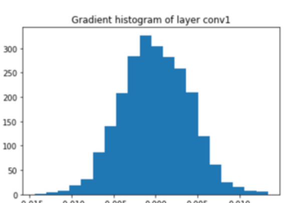


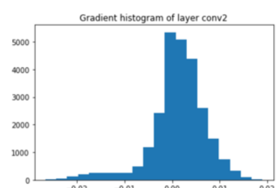


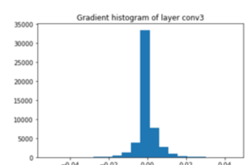


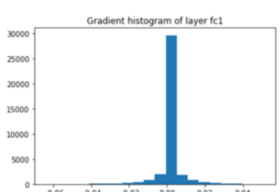


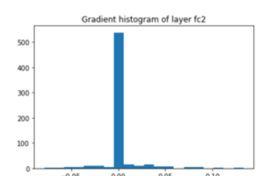
ii Bonus



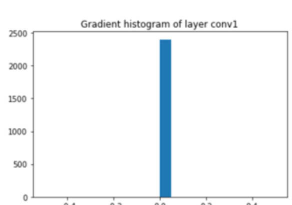


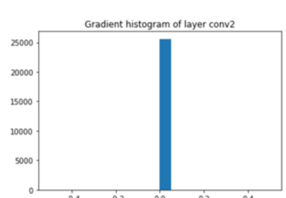


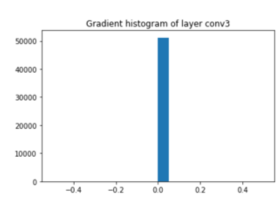


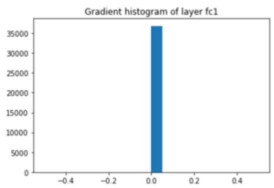


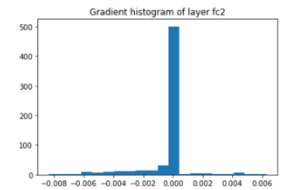
iii Bonus











by initializing the weights to zero, we can see that the gradient is in a stable 0 for all the layers in the CNN model.