Image Processing Assignment 1

Yang Ke

[Task 1]

(a) Sampling is digitizing the x and y coordinates of an image.
(b) Quantization is digitizing the amplitude values.
(c) The gradient is significant in the cumulative frequency graph both in the low-intensity value part and the high-intensity value part.
(d)

Pixel int.	0	1	2	3	4	5	6	7
No. of pixels	1	1	0	1	2	2	4	4
pdf	1/15	1/15	0	1/15	2/15	2/15	4/15	4/15
cdf	0.0667	0.1333	0.1333	0.2	0.3333	0.4667	0.7333	1
Cum. prob. * 20	1.334	2.666	2.666	4	6.666	9.334	14.666	20
Rounded	1	2	2	4	6	9	14	20

Transformed image:

20	14	9	14	6
9	6	20	20	1
2	20	14	4	14

(e) The bright intensities is squeezed, and dark intensities is strengthened.
(f) Pre-rotate *w:*

-1	0	1
-2	0	2
-1	0	1

Padded the image with 0 to 5x7 size:

0	0	0	0	0	0	0
0	7	6	5	6	4	0
0	5	4	7	7	0	0
0	1	7	6	3	6	0
0	0	0	0	0	0	0

Multiply 1st square:

0	0	0
0	7	6
0	5	4

*

-1	0	1
-2	0	2
-1	0	1

=>2*6+ 4 =16

Multiply 2nd square:

0	0	0
7	6	5
5	4	7

*

-1	0	1
-2	0	2
-1	0	1

=> 7 *(-2) +2*5+5*(-1) + 7 =-2

……

Final result:

16	-2	3	-9	-19
21	2	2	-15	-23
18	16	-5	-7	-13

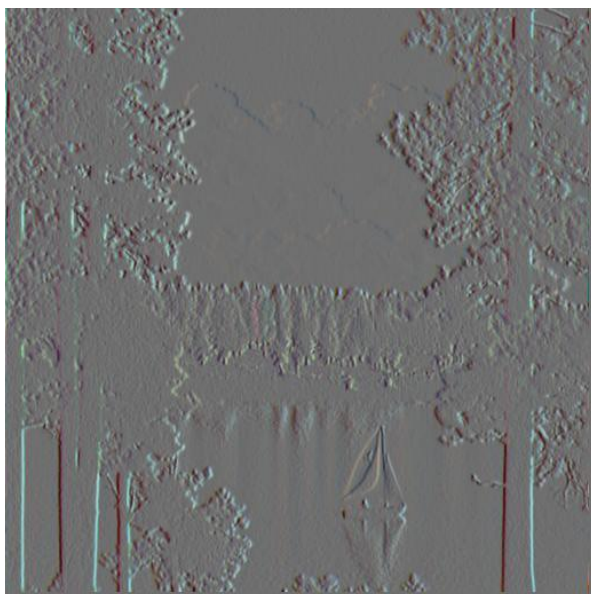
[Task 2]

(a)

(b)

(c) Smoothed:

Sobel:



[Task 3]

(a) XOR cannot be represented by a single-layer neural network.

(b) Hyperparameter is the element(s) of the neural network that is pre-determined and will not be updated during optimization.
It includes the number of hidden layers and activation function.

(c) Softmax can turn any kind of input value into a value between 0 and 1, and thus enables the interpretation of probability.

(d) (e)

Forward Propagation:

$x_1 = -1, \quad w_1 = -1, \quad a_1 = w_1 * x_1 = 1,$

$x_2 = 0, \quad w_2 = 1, \quad a_2 = w_2 * x_2 = 0$

$\Big\rangle c_1 = a_1 + a_2 + b_1 = 2$

$x_3 = -1, \quad w_3 = -1, \quad a_3 = w_3 * x_3 = 1$

$x_4 = 2, \quad w_4 = -2, \quad a_4 = w_4 * x_4 = -4$

$\Big\rangle c_2 = a_3 + a_4 + b_2 = -4$

$\Big\rangle \hat{y} = \max(c_1, c_2) = a_1 + a_2 + b_1 = 2$

$C = \frac{1}{2}(2-1)^2 = \frac{1}{2}$

Derivatives: (Backward Pass).

since $c_1 > c_2$, $\hat{y} = c_1 = a_1 + a_2 + b_1$, $\frac{\partial \hat{y}}{\partial a_1} = 1$

$\frac{\partial C}{\partial w_1} = \frac{\partial C}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_1} \cdot \frac{\partial a_1}{\partial w_1} = (y - \hat{y}) * 1 * x_1 = (1-2) * 1 * (-1) = 1$

$\frac{\partial C}{\partial w_2} = \frac{\partial C}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_2} \cdot \frac{\partial a_2}{\partial w_2} = (y - \hat{y}) * 1 * x_2 = (1-2) * 1 * 0 = 0$

$\frac{\partial C}{\partial w_3} = \frac{\partial C}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_3} \cdot \frac{\partial a_3}{\partial w_3} = (y - \hat{y}) * 0 * x_3 = 0$ $\Rightarrow$ for a_3, $\hat{y} = a_1 + a_2 + b_1$ is a constant

$\frac{\partial C}{\partial w_4} = \frac{\partial C}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial a_4} \cdot \frac{\partial a_4}{\partial w_4} = (y - \hat{y}) * 0 * x_4 = 0$

$\frac{\partial C}{\partial b_1} = \frac{\partial C}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial b_1} = (y - \hat{y}) * 1 = 1 - 2 = -1.$

$\frac{\partial C}{\partial b_2} = \frac{\partial C}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial b_2} = (y - \hat{y}) * 0 = 0.$

Update Parameters:

new $w_1 = w_1 - \alpha \frac{\partial C}{\partial w_1} = -1 - 0.1 \times 1 = -1.1$

new $w_2 = w_2 - \alpha \frac{\partial C}{\partial w_2} = 1 - 0.1 \times 0 = 1.$

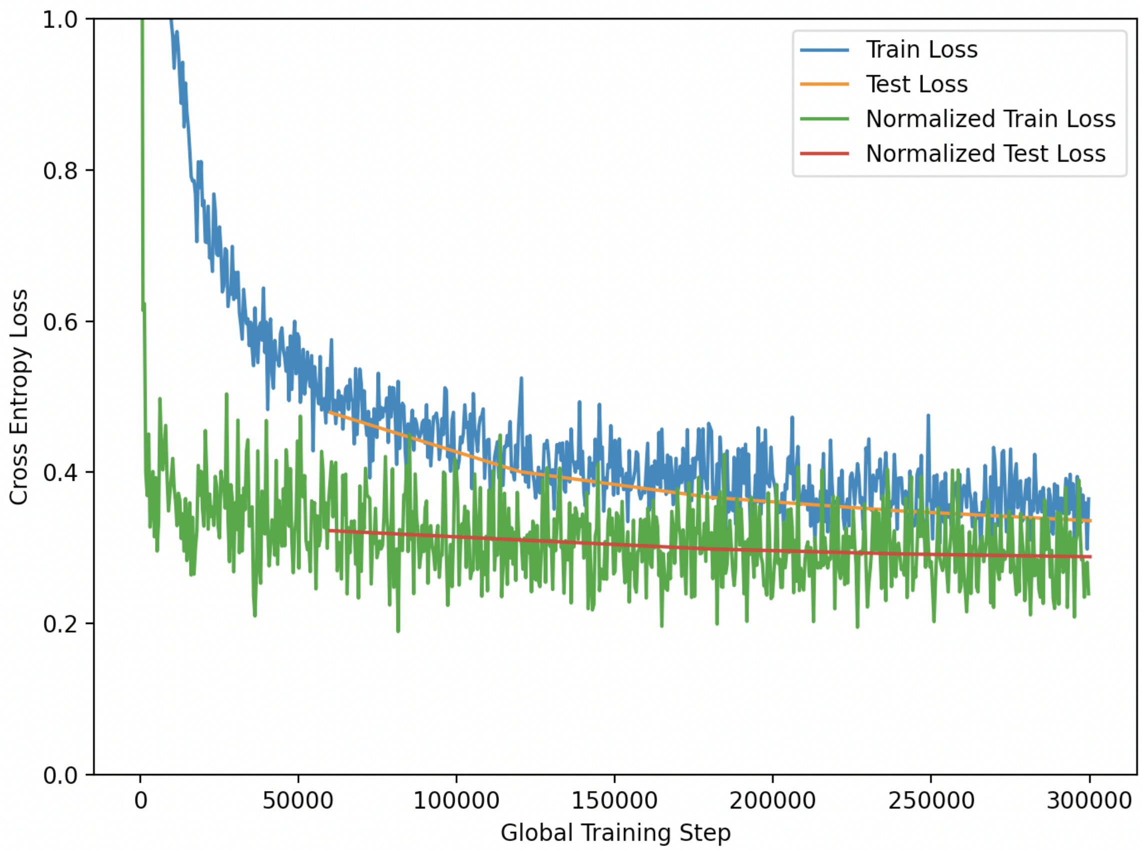
new $w_3 = w_3 - \alpha \frac{\partial C}{\partial w_3} = -1 - 0.1 \times 0 = -1$

new $b_1 = b_1 - \alpha \frac{\partial C}{\partial b_1} = 1 - 0.1 \times (-1) = 1.1$

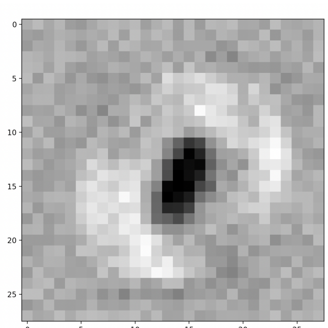
new $b_2 = b_2 - \alpha \frac{\partial C}{\partial b_2} = -1 - 0.1 \times 0 = -1.$

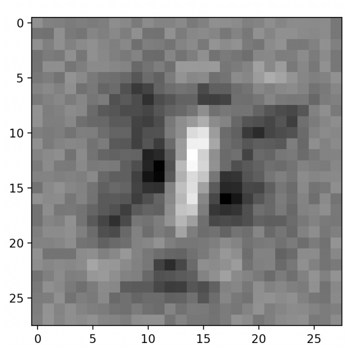
[Task 4]

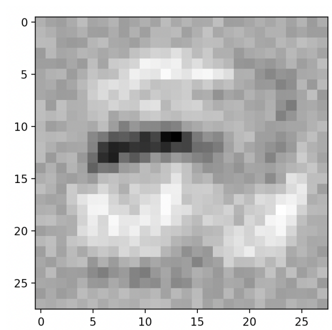
(a) With normalization, the train loss and test loss are in general smaller than the image without normalization.

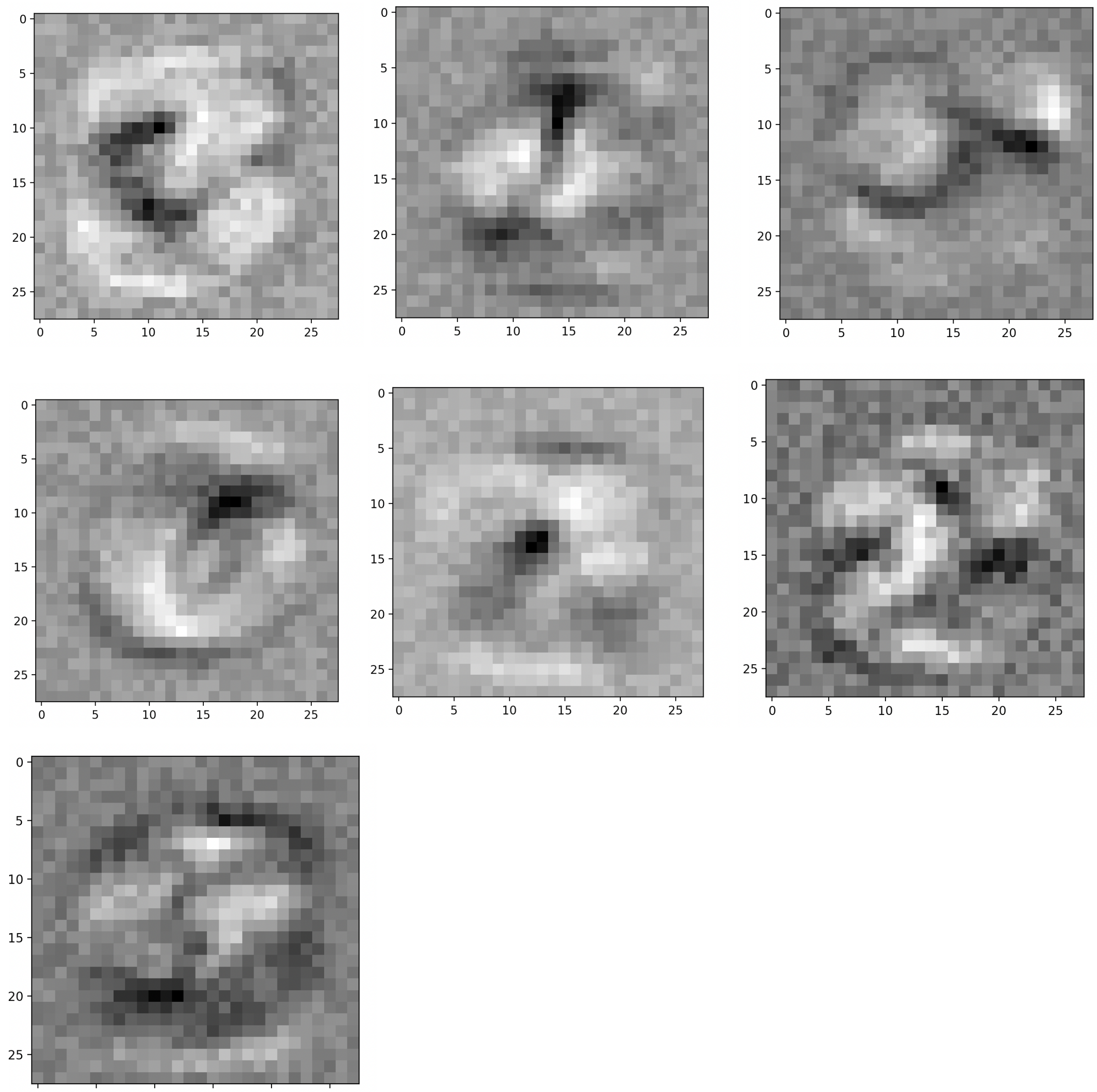


(b) Images of learned weights from 0 - 9:









Observation: Each visualized weight looks like the digit it learned.

(c) Outcomes with original learning rate of 0.0192:

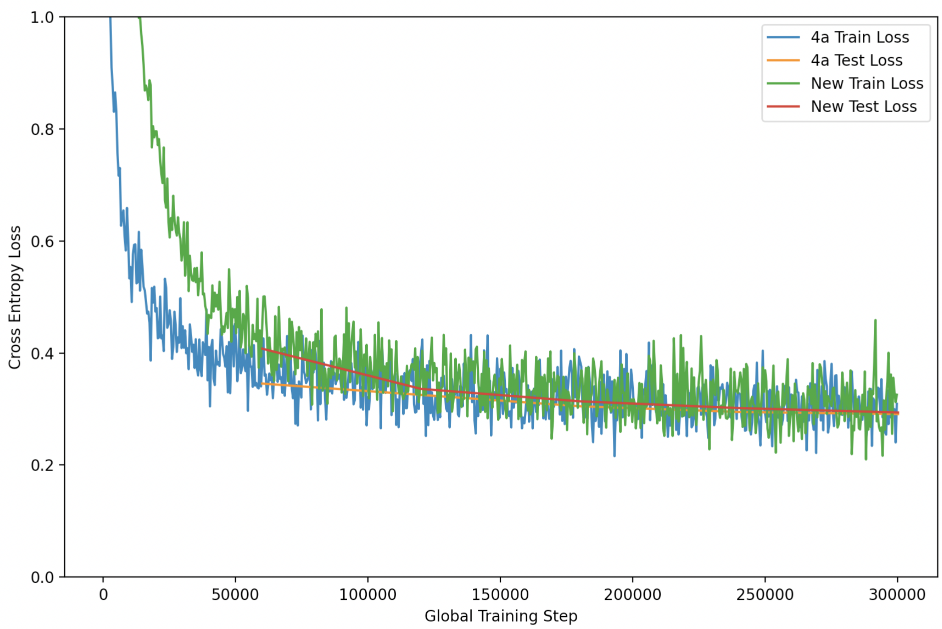
```
Average test loss of original model: 0.3122814382432373.
Final Test loss: 0.2910341706933679. Final Test accuracy: 0.9166
```

Outcome with new learning rate of 1.0:

```
Average test loss of 4c: 9.245849809376017.
Final Test loss: 4.404377571858798. Final Test accuracy: 0.8256
```

The new network has a worse accuracy because the learning rate is too high that it is not able to find the optimal weights and the point with the lowest cost.

(d) The green and the red lines are the outcomes of the new network.



Final accuracy:

```
Average test loss of original model: 0.33058028038899606.
Final Test loss: 0.2938446206319484. Final Test accuracy: 0.9181
```

Observations: The loss of the new network is slightly higher than the network of (a), however the accuracy of the new network is higher.