TDT4195 – Assignment 1 IP

Task 1

a) Sampling is the process of pixelating an input (i.e. input: a continues image, output: an image with a finite number of pixels).
b) Quantization is the process of transforming a continuous signal into a discrete signal (i.e. a finite number of greyscale values).
c) The image has high contrast if the entire range (black to white) is used.

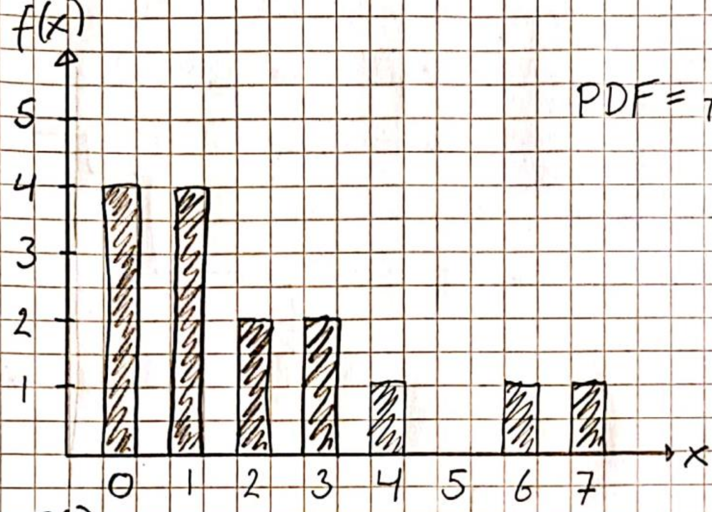
d)

1) d)

1	0	2	3	1
3	2	0	7	0
0	6	1	1	4

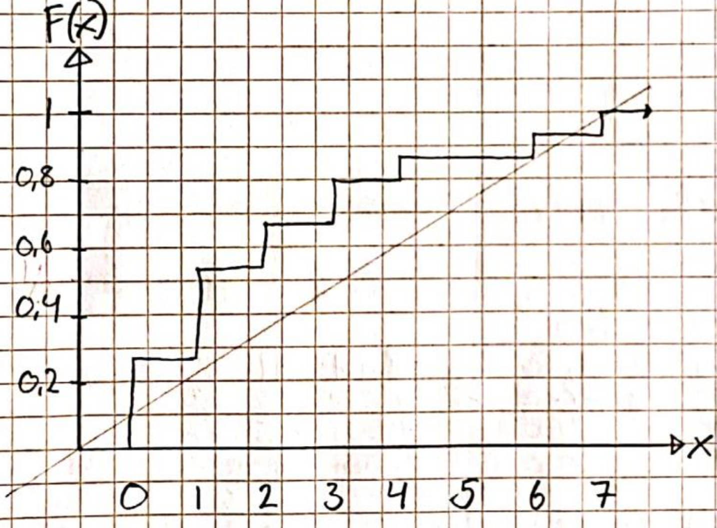
→ 0 0 0 0 1, 1 1 1 2 2, 3 3 4 6 7

15 values in total



PDF = f(x)

x	f(x)
0	4
1	4
2	2
3	2
4	1
5	0
6	1
7	1



CDF = F(x)

$F(a) = P(x \leq a)$

x	F(x)
0	4/15
1	8/15
2	10/15
3	12/15
4	13/15
5	13/15
6	14/15
7	1

$T(x) = \lfloor 7F(x) \rfloor$

x	T(x)
0	1
1	3
2	4
3	5
4	6
5	6
6	6
7	7

1	0	2	3	1
3	2	0	7	0
0	6	1	1	4

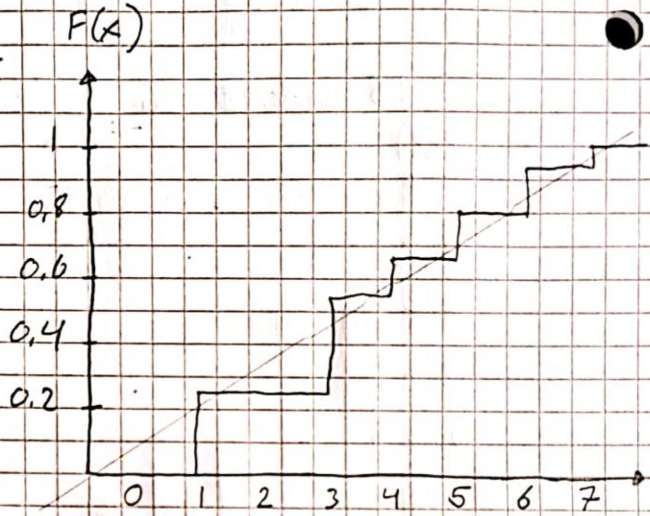
=>

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

3 1 4 5 3
5 4 1 7 1 => 1 1 1 1 3, 3 3 3 4 4, 5 5 6 6 7
1 6 3 3 6

x	f(x)
0	0
1	4
2	0
3	4
4	2
5	2
6	2
7	1

x	F(x)
0	0
1	4/15
2	4/15
3	8/15
4	10/15
5	12/15
6	14/15
7	1



1) f) Since we apply a convolution, we flip the kernel.

$$\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 0&0&0&0&0&0&0 \\ 0&1&0&2&3&1&0 \\ 0&3&2&0&7&0&0 \\ 0&0&6&1&1&4&0 \\ 0&0&0&0&0&0&0 \end{bmatrix} = \begin{bmatrix} -2 & 1 & -11 & 2 & 13 \\ -10 & 4 & -8 & -2 & 18 \\ -14 & 1 & 5 & -6 & 9 \end{bmatrix}$$

For example (2,1): $\begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * \begin{bmatrix} 0 & 2 & 3 \\ 2 & 0 & 7 \\ 6 & 1 & 1 \end{bmatrix} = 1\cdot0 + 2\cdot0 - 3\cdot1$

$+2\cdot2 + 0\cdot0 - 2\cdot7 + 6\cdot1 + 0\cdot0 - 1\cdot1 = 0 + 0 - 3 + 4 + 0$

$-14 + 6 + 0 - 1 = -3 + 4 - 14 + 6 - 1 = -8$

e) The dynamic range will be compressed to the right (more white) for a log, and compressed to the left (more black) for an inverse log.

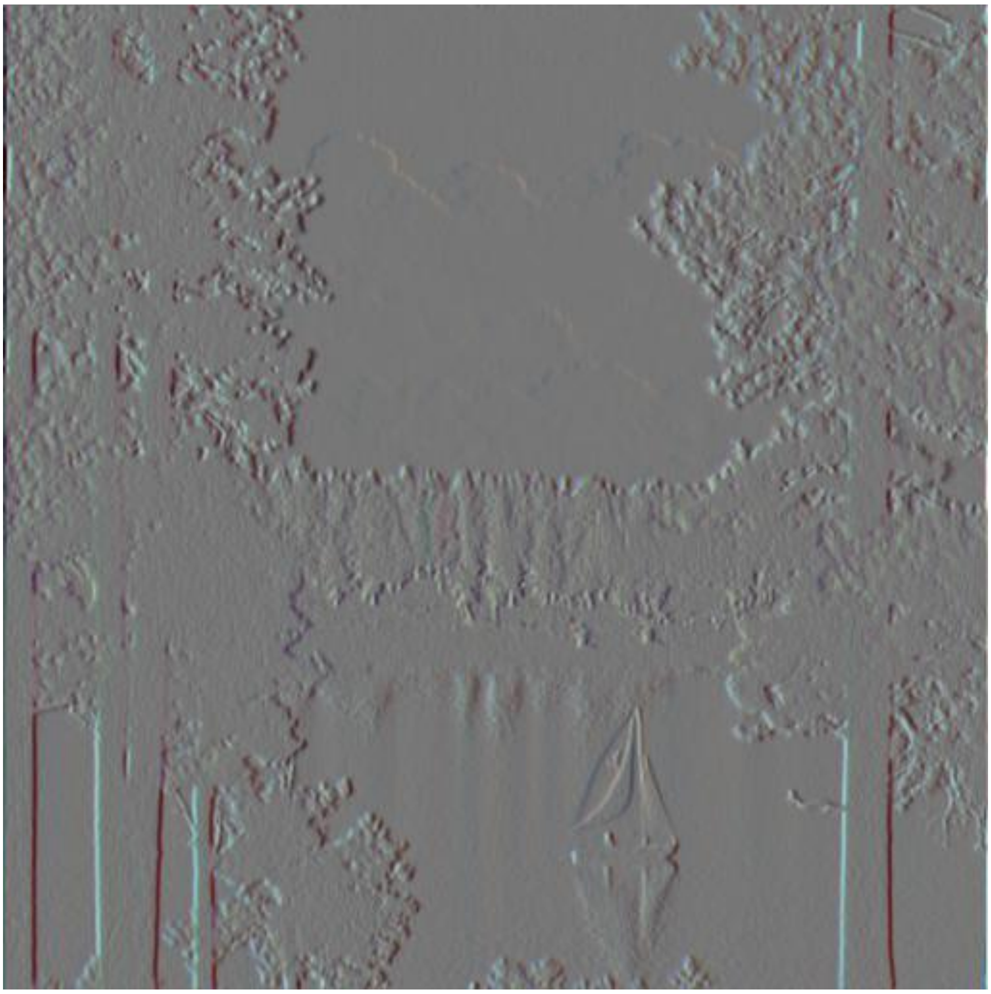
f) See the image above.

Task 2

Task A

Task B

Task C



Sobel.

Smoothed.

Task 3

a) XOR
b) Hyperparameters are the variables that are set before training. For example, batch size and learning rate.
c) The SoftMax activation function is used because it maps all input values to the [0,1]. In addition, all these [0, 1] values sum to 1, so the output values can be seen as a probability distribution (used in classifiers).

d) Short answer: $-2, 0, 0, 0, 2, 0$ and $-0.8, -1, 0.8$.

3) d) $\hat{y} = 1$

$$C(y_n, \hat{y}_n) = (y_n - \hat{y}_n)^2 \qquad \theta_{t+1} = \theta_t - \alpha \frac{dC}{d\theta_t} = \theta_t - \alpha \frac{\partial}{\partial \theta_t} C$$

$t = 0$:

$$\left.\begin{array}{l} b_1 = 1 \\ x_1 = -1,\ w_1 = -1 \Rightarrow a_1 = 1 \\ x_2 = 0,\ w_2 = 1 \Rightarrow a_2 = 0 \end{array}\right\} \Rightarrow c_1 = 2$$

$$\left.\begin{array}{l} x_3 = -1,\ w_3 = -1 \Rightarrow a_3 = 1 \\ x_4 = 2,\ w_4 = -2 \Rightarrow a_4 = -4 \\ b_2 = -1 \end{array}\right\} \Rightarrow c_2 = -4$$

$$\Rightarrow y = 2$$

$$\frac{dC}{dy} = 2(y_n - \hat{y}) = 2(2-1) = 2$$

$$\frac{dC}{dc_1} = \frac{dC}{dy} \cdot \frac{dy}{dc_1} = 2 \cdot 1 = 2$$

$$\frac{dC}{db_1} = \frac{dC}{dc_1} \cdot \frac{dc_1}{db_1} = 2 \cdot 1 = 2$$

$$\frac{dC}{da_1} = \frac{dC}{dc_1} \cdot \frac{dc_1}{da_1} = 2 \cdot 1 = 2$$

$$\frac{dC}{dw_1} = \frac{dC}{da_1} \cdot \frac{da_1}{dw_1} = 2 \cdot x_1 = -2$$

$$\frac{dC}{da_2} = \frac{dC}{dc_1} \cdot \frac{dc_1}{da_2} = 2 \cdot 1 = 2$$

$$\frac{dC}{dw_2} = \frac{dC}{da_2} \cdot \frac{da_2}{dw_2} = 2 \cdot x_2 = 0$$

Since $c_1 > c_2$ and $y = \max(c_1, c_2)$ $\Rightarrow y = c_1$. This means that y does not depend on c_2. Thus, $\frac{dC}{dc_2} = 0$

Now, the lower part of the network:
Since $\frac{dC}{dc_2} = 0$, all gradients of the lower part of the network is also 0. So:

$$\frac{dC}{dw_1} = -2 \quad \frac{dC}{dw_2} = 0 \quad \frac{dC}{db_1} = 2 \quad \frac{dC}{dw_3} = \frac{dC}{dw_4} = \frac{dC}{db_2} = 0$$

e)

3) e) $\alpha = 0{,}1$

$$w_1 \Leftarrow w_1 - \alpha \frac{dC}{dw_1} = -1 - 0{,}1(-2) = \underline{\underline{-0{,}8}}$$

$$w_3 \Leftarrow w_3 - \alpha \frac{dC}{dw_3} = -1 - 0{,}1 \cdot 0 = \underline{\underline{-1}} \} \rightarrow \text{Unchanged}$$

$$b_1 \Leftarrow b_1 - \alpha \frac{dC}{db_1} = 1 - 0{,}1 \cdot 2 = \underline{\underline{0{,}8}}$$

As we can see below, the new calculated value for C is 1.6 at $t = 1$. This value is closer to the target value, than C at $t = 1$.

$$w_2 \Leftarrow w_2 - \alpha \frac{dC}{dw_2} = 1 - 0{,}1 \cdot 0 = 1$$

$$\left.\begin{array}{l} w_4 \Leftarrow w_4 - \alpha \frac{dC}{dw_4} = -2 - 0{,}1 \cdot 0 = -2 \\ b_2 \Leftarrow b_2 - \alpha \frac{dC}{db_2} = -1 - 0{,}1 \cdot 0 = -1 \end{array}\right\} \text{Unchanged}$$

Forward pass $t=1$:

$$\left.\begin{array}{l} \left.\begin{array}{lll} x_1 = -1, & w_1 = -0{,}8 & \Rightarrow a_1 = 0{,}8 \\ x_2 = 0, & w_2 = 1 & \Rightarrow a_2 = 0 \\ & b_1 = 0{,}8 & \end{array}\right\} C_1 = 1{,}6 \\ \left.\begin{array}{lll} x_3 = -1, & w_3 = -1 & \Rightarrow a_3 = 1 \\ x_4 = 2, & w_4 = -2 & \Rightarrow a_4 = -4 \\ & b_2 = -1 & \end{array}\right\} C_2 = -4 \end{array}\right\} Y_1 = 1{,}6 \quad \square$$

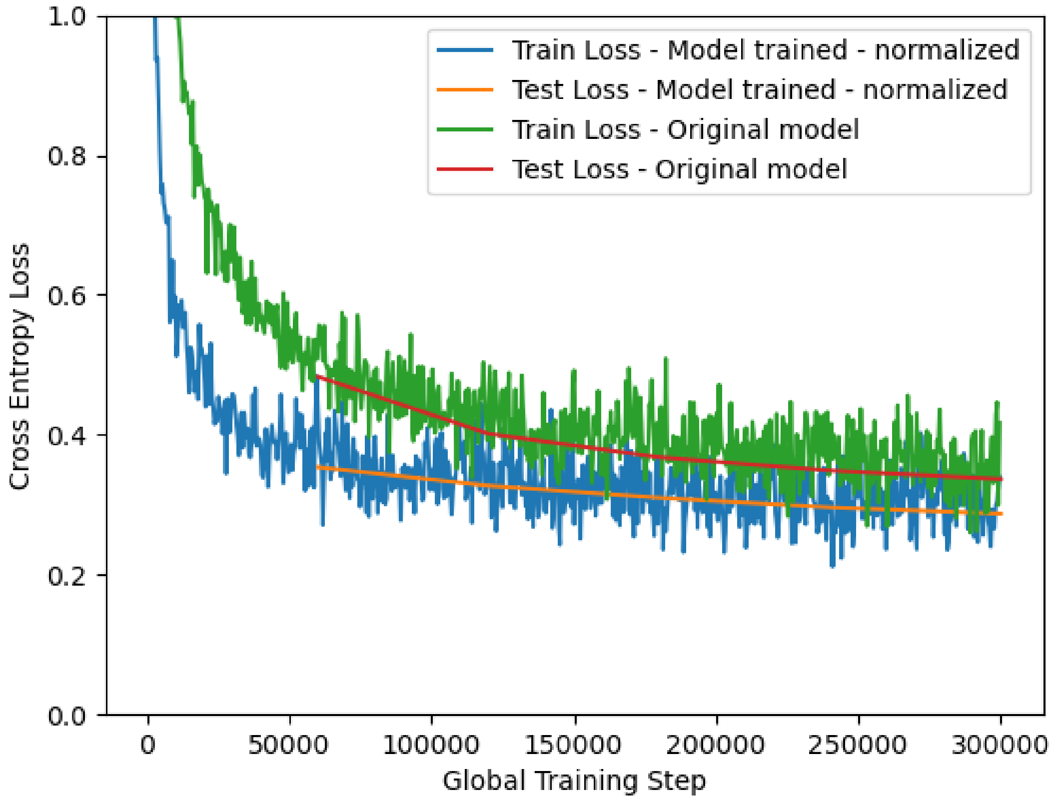
Unchanged

We only need to calculate new values for the upper part of the network, because it won the max() function.

Task 4

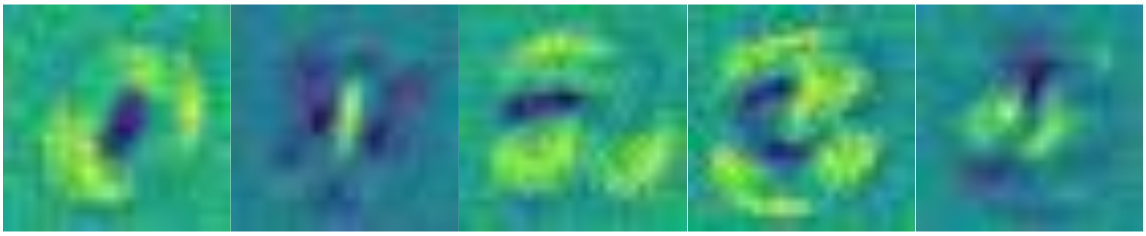
Task A

As we can see, the model learns a lot faster in the beginning with the normalized image.

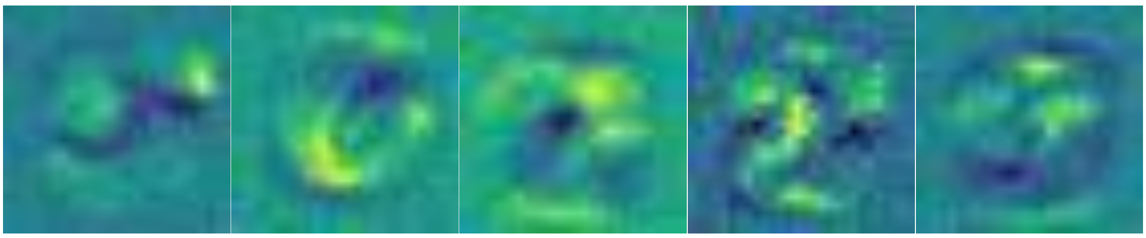


Task B

The images below represent the inputs that are important for a certain class (i.e. the weights). As an example, let us inspect 0. The 0 class cares most about whether the pixels around the center is lit up and not the center itself.



0-4

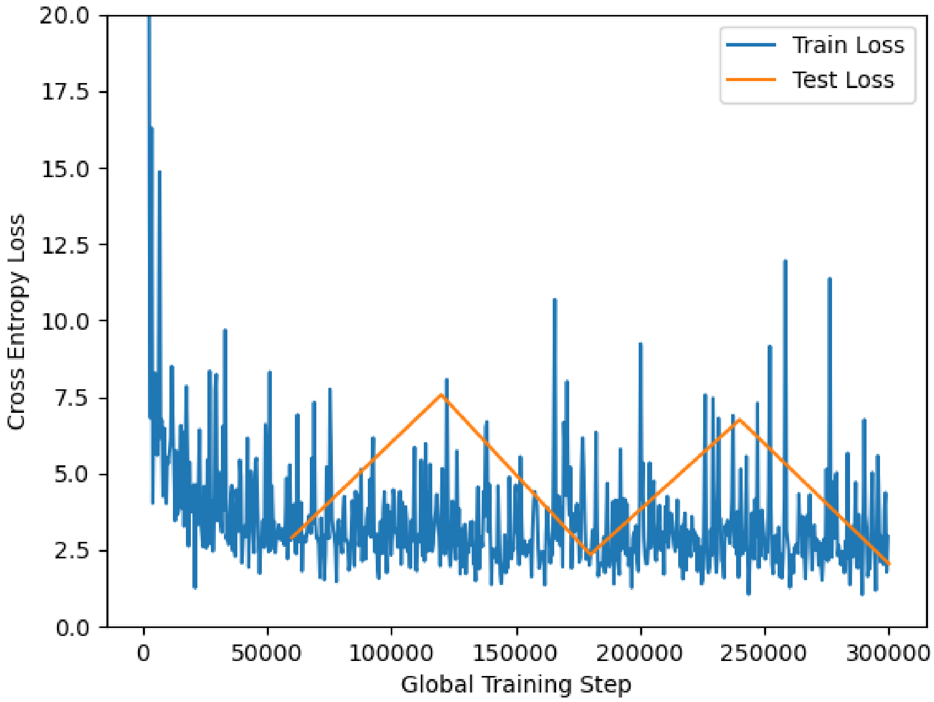


5-9

Task C

With a learning rate of 1.0, the model performs worse than with a learning rate of 0.192.

This is because the learning rate is too high, so the gradient descent algorithm fails to converge.



Task D

With one hidden layer, we can see that the model learns better. The loss does not stop at 0.3 but keeps sinking down to 0.2 as the step increases.