Assignment 1

Task 1

task 1.a

Sampling is the digitization of the coordinate values.

task 1.b

Quantization is the digitization of the amplitude values.

task 1.c

We can see if an image has high contrast if the histogram covers a wide range of the gray scale and if the distribution of pixels are close to the same level.

task 1.d

Initial matrix:

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

r	H(r)	p(r)	F(r)	T(r)	T(r) sum
0	1	1/15	1/15	7*1/15	0
1	1	1/15	2/15	7*2/15	0
2	0	0/15	2/15	7*2/15	0
3	1	1/15	3/15	7*3/15	1
4	2	2/15	5/15	7*5/15	2
5	2	2/15	7/15	7*7/15	3
6	4	4/15	11/15	7*11/15	5
7	4	4/15	15/15	7*15/15	7

The final matrix:

5 7 3 2 5

2 3 7 0 7

7 0 5 5 1

task 1.e

A log transform will give lower input intensities a boost and flatten high input intensities. The log transform helps see the lower ranges in high variance images.

task 1.f

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

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

Spacial Convolution				
19	1	-11	2	-8
18	4	-8	-2	-10
7	1	5	-6	-12

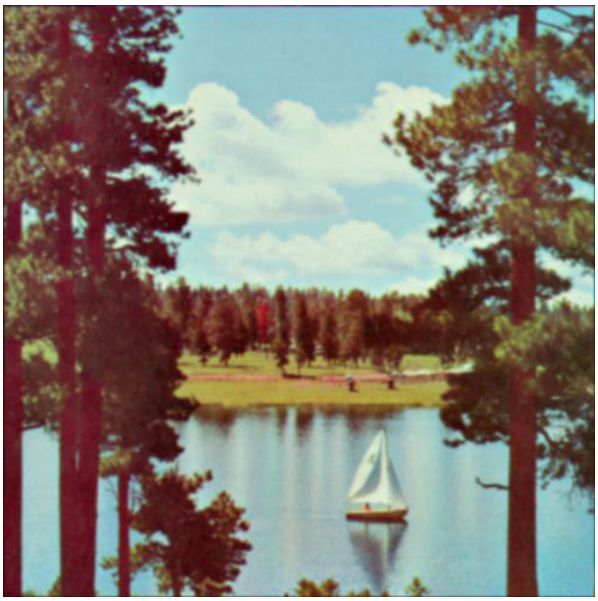
By using zero padding on the image and flipping the kernal for convolution calculation we calcualte the spacial convolution. This is done by multiplien each element of the kernel with the center of the kernel on the cell we want to calculate and sum up the total value to get the final answer.

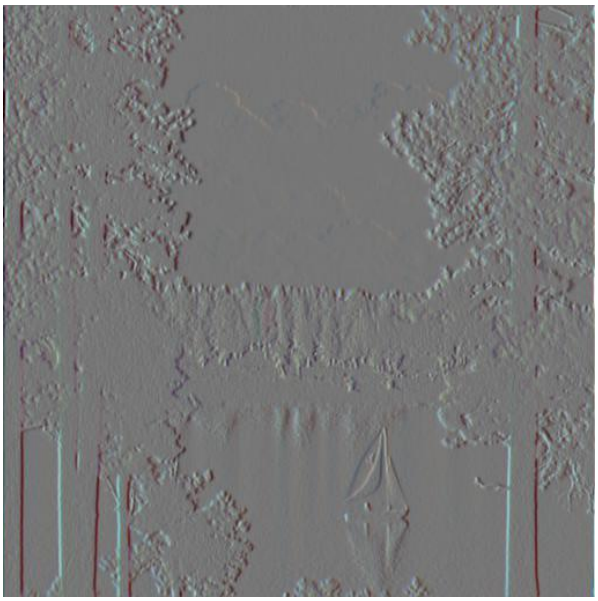
Task 2

Task 2.a

Task 2.b

Task 2.c





Task 3

Task 3.a

If we look at the binary operator XOR, we can see that this cannot be represented with a single layer neural network. When looking at the possiable states we get that we want the vector [0, 1, 1, 0] which is impossiable with a linear function. This is because this problem is lineary separable.

Task 3.b

A hyperparameter is a constant paramater used to controll the learning prosess. THis parameter is predefined and cannot be changed in the learning process. Some examples of hyperparameteres are learning rate and batch size.

Task 3.c

The softmax activation function takes a vector of numbers and returnes a vector of probabilities. The softmax function basiclly gives the end user a vector that is readable and understandable with numbers from [0,1] for each value.

Task 3.d

$C(y_n, \hat{y}_n) = \frac{1}{2}(y_n - \hat{y}_n)^2, y_n = 1$

After simple calculation we found that $c_1 = 2$ and $c_2 = -4$

Since the max activation function chooses the largest value we found that $\hat{y} = c_1 = 2$

Further we know that $\frac{dC}{dy} = (y_n - \hat{y}_n)$ which leads to $y' = \frac{dC}{dy}(1, 2) = (1 - 2) = -1$

$$\frac{dC}{dc_1} = \frac{dC}{dy}\frac{dy}{dc_1} = -1 * 1 = -1$$
$$\frac{dC}{dc_2} = \frac{dC}{dy}\frac{dy}{dc_2} = -1 * 0 = 0$$

$$\frac{dC}{db_1} = \frac{dC}{dc_1}\frac{dc_1}{db_1} = -1 * 1 = -1$$
$$\frac{dC}{db_2} = \frac{dC}{dc_2}\frac{dc_2}{db_2} = 0 * 1 = 0$$

$$\frac{dC}{da_1} = \frac{dC}{dc_1}\frac{dc_1}{da_2} = -1 * 1 = -1$$
$$\frac{dC}{da_2} = \frac{dC}{dc_1}\frac{dc_1}{da_2} = -1 * 1 = -1$$
$$\frac{dC}{da_3} = \frac{dC}{dc_2}\frac{dc_2}{da_3} = 0 * 1 = 0$$
$$\frac{dC}{da_4} = \frac{dC}{dc_2}\frac{dc_2}{da_4} = 0 * 1 = 0$$

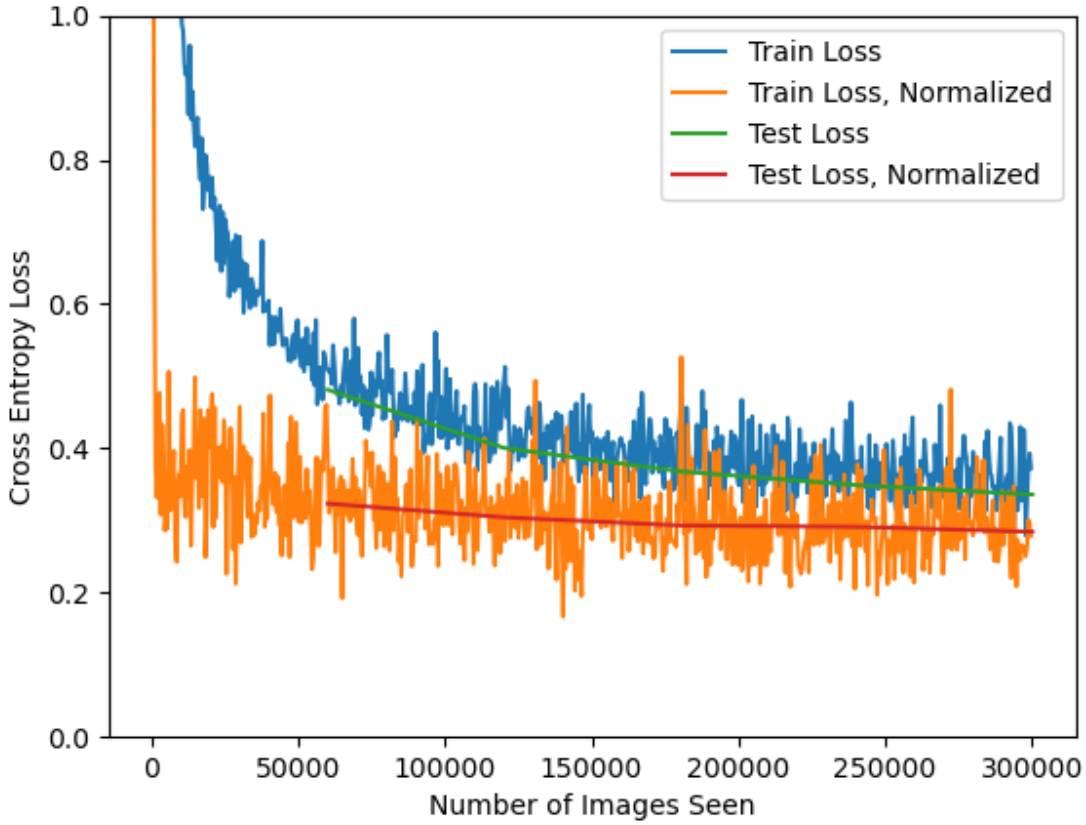
$$\frac{dC}{dw_1} = \frac{dC}{da_1}\frac{da_1}{dw_1} = -1 * x_1 = 1$$
$$\frac{dC}{dw_2} = \frac{dC}{da_2}\frac{da_2}{dw_2} = -1 * x_2 = 0$$
$$\frac{dC}{dw_3} = \frac{dC}{da_3}\frac{da_3}{dw_3} = 0 * x_3 = 0$$
$$\frac{dC}{dw_4} = \frac{dC}{da_4}\frac{da_4}{dw_4} = 0 * x_4 = 0$$

Task 3.e

$$w_1 = w_1 - \alpha\frac{dC}{dw_1} = -1 - 0.1 * 1 = 0.9$$
$$w_3 = w_3 - \alpha\frac{dC}{dw_3} = -1 - 0.1 * 0 = -1$$
$$b_1 = b_1 - \alpha\frac{dC}{db_1} = 1 - 0.1 * -1 = 1.1$$

Task 4

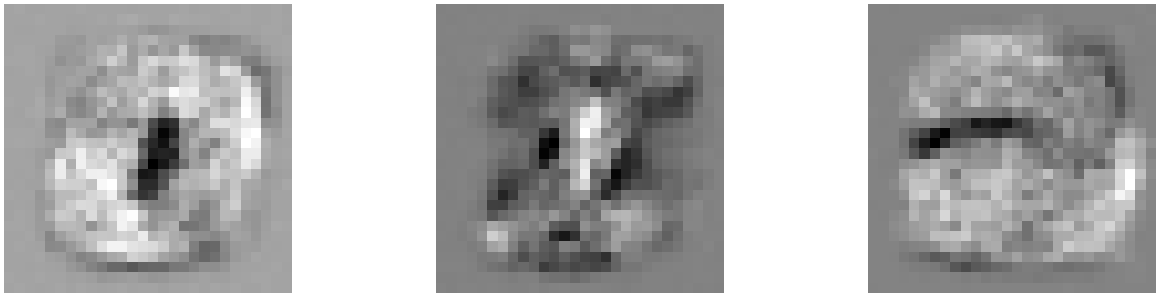
Task 4.a



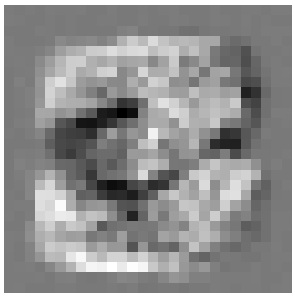
Normalized: Loss: 0.2844127357290809. Accuracy: 0.9195

As we can see the loss converges faster, to a smaller value with the normalized data.

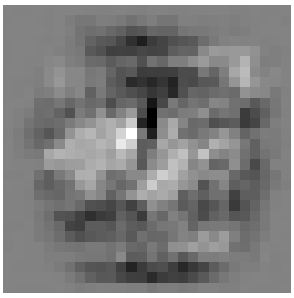
Task 4.b



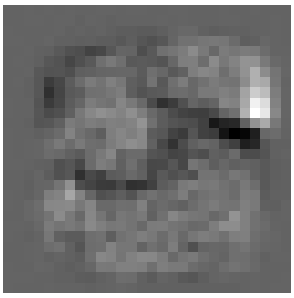
0



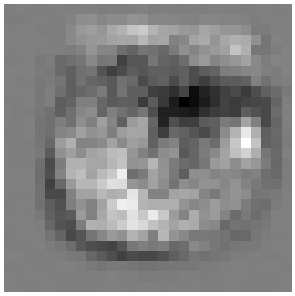
1



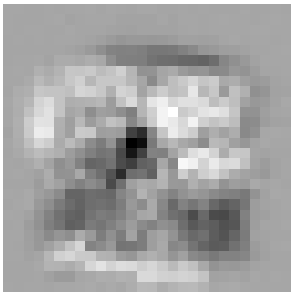
2



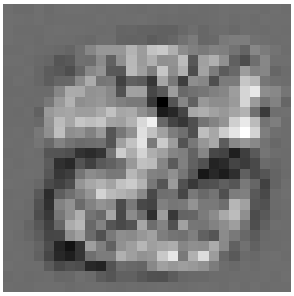
3



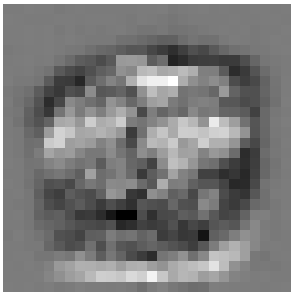
4



5



6



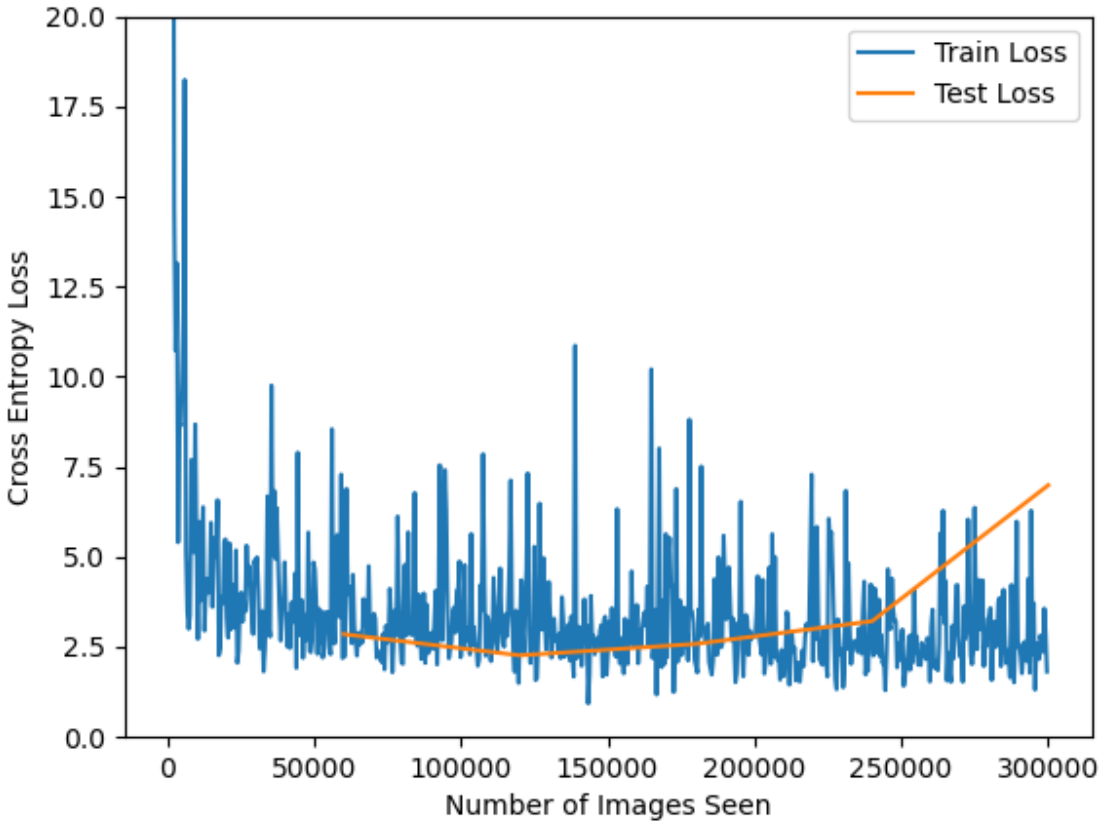
7

8

9

As ween in these images we can kinda see the different numbers written, especially for the number 0 where we can clearly see the outline of a zero with the hole in the middle. These images shows what pixel gives increses the probability of that number. Since we only have one layer, what we are seeing is a direct mapping from input to output.

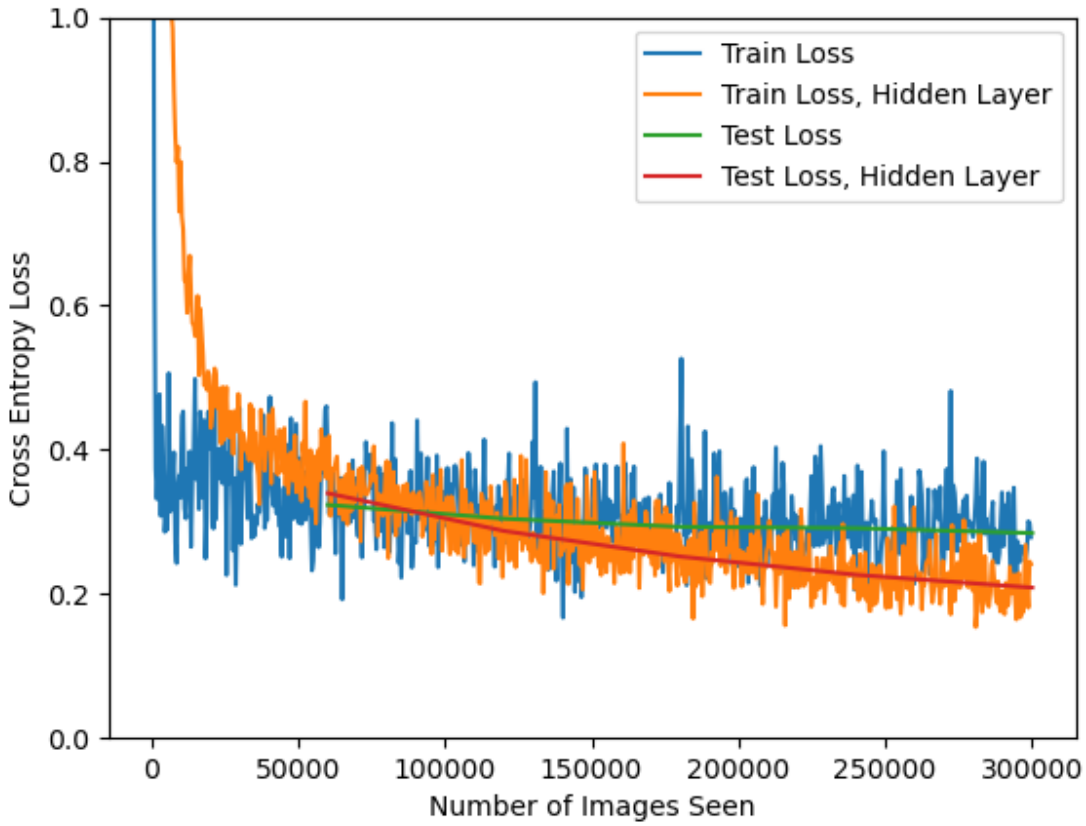
Task 4.c



Loss: 3.1971189732699297. Accuracy: 0.8519

As we can see the loss is way higher, as expected. Since we have given it a learning rate of 1, the gradient decent is much less accurate and the loss is way higher. Its worth noting that the y axis is set to a limit of 20 instead of 1. We can also see that there is a lot higher loss and lower accuracy then Task 4.a

Task 4.d



Loss: 0.20199841267435223. Accuracy: 0.937

We can see that the ReLU neural network has a lower train and test loss as well as converges to a lower value which makes the network faster and better then the original. The overall accuracy is also a lot higher for the ReLU network.