# TDT4195 Visuell Databehandling

Vegard Iversen and Sebastian Skogen Raa

October 2021

## Image Processing - Assignment 1 - Visual Computing Fundamentals

## Task 1: Theory

### Task 1a

**Question**
Explain in one sentence what sampling is
**Answer**
Sampling is when we check whats the value/observation are from for example a continuous signal.

### Task 1b

**Question**
Explain in one sentence what quantization is.
**Answer**
Quantization is when you take a continuous value and make it discreet by rounding it with a certain quantity.

### Task 1c

**Question**
Looking at an image histogram, how can you see that the image has high contrast? **Answer**
If the histogram as a broad color distribution along the tonal range, or several narrow prominences set far apart.

### Task 1d

**Question**
Perform histogram equalization by hand on the 3-bit (8 intensity levels) image in Figure 1a Your report must include all the steps you did to compute the histogram, the transformation, and the transformed image. Round down any resulting pixel intesities that are not integer (use the floor operator). **Answer**
"n" goes from 0-7 since thats the highest value in the image. $f_n$ is the number of occurrences devided by the total number of pixels $F_n$ is the cumulative distribution. to get the intensities we multiply the max pixel value (7) with the cumulative distribution.

| n | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| $f_n$ | $\frac{1}{15}$ | $\frac{1}{15}$ | $\frac{0}{15}$ | $\frac{1}{15}$ | $\frac{2}{15}$ | $\frac{2}{15}$ | $\frac{4}{15}$ | $\frac{4}{15}$ |
| $F_n$ | $\frac{1}{15}$ | $\frac{2}{15}$ | $\frac{2}{15}$ | $\frac{3}{15}$ | $\frac{5}{15}$ | $\frac{7}{15}$ | $\frac{11}{15}$ | $\frac{15}{15}$ |

| Intensity | step | floored |
|---|---|---|
| $s_0$ | $7 \cdot \frac{1}{15}$ | 0 |
| $s_1$ | $7 \cdot \frac{2}{15}$ | 0 |
| $s_2$ | $7 \cdot \frac{2}{15}$ | 0 |
| $s_3$ | $7 \cdot \frac{3}{15}$ | 1 |
| $s_4$ | $7 \cdot \frac{5}{15}$ | 2 |
| $s_5$ | $7 \cdot \frac{7}{15}$ | 3 |
| $s_6$ | $7 \cdot \frac{11}{15}$ | 5 |
| $s_7$ | $7 \cdot \frac{15}{15}$ | 7 |

## Task 1e

**Question**
What happens to the dynamic range if we apply a log transform to an image with a large variance in pixel intensities? **Answer**

By utilizing a log transform when saving an image one could be able to fit a larger dynamic range. The reason is that one can be able to fit the larger intensities at the same time as the important details in the low light intensities are preserved. This is when assuming the bit depth available is limited.

## Task 1f

Table 1: Rotated cw sobel kernel by $\pi$.

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

First row

$$\begin{aligned}[0,0] =& 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1\\ & 0 \cdot -2 + 6 \cdot 0 + 7 \cdot 2\\ & 0 \cdot -1 + 4 \cdot 0 + 5 \cdot 1\\ =& 19\end{aligned}$$

$$
\begin{aligned}
[1,0] =& 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
& 6 \cdot -2 + 7 \cdot 0 + 5 \cdot 2 \\
& 4 \cdot -1 + 5 \cdot 0 + 7 \cdot 1 \\
=& 1
\end{aligned}
$$

$$
\begin{aligned}
[2,0] =& 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
& 7 \cdot -2 + 5 \cdot 0 + 4 \cdot 2 \\
& 5 \cdot -1 + 7 \cdot 0 + 0 \cdot 1 \\
=& -11
\end{aligned}
$$

$$
\begin{aligned}
[3,0] =& 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
& 5 \cdot -2 + 4 \cdot 0 + 6 \cdot 2 \\
& 7 \cdot -1 + 0 \cdot 0 + 7 \cdot 1 \\
=& 2
\end{aligned}
$$

$$
\begin{aligned}
[4,0] =& 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
& 4 \cdot -2 + 6 \cdot 0 + 0 \cdot 2 \\
& 0 \cdot -1 + 7 \cdot 0 + 0 \cdot 1 \\
=& -8
\end{aligned}
$$

Second row

$$
\begin{aligned}
[0,1] =& 0 \cdot -1 + 6 \cdot 0 + 7 \cdot 1 \\
& 0 \cdot -2 + 4 \cdot 0 + 5 \cdot 2 \\
& 0 \cdot -1 + 7 \cdot 0 + 1 \cdot 1 \\
=& 18
\end{aligned}
$$

$$
\begin{aligned}
[1,1] =& 6 \cdot -1 + 7 \cdot 0 + 5 \cdot 1 \\
& 4 \cdot -2 + 5 \cdot 0 + 7 \cdot 2 \\
& 7 \cdot -1 + 1 \cdot 0 + 6 \cdot 1 \\
=& 4
\end{aligned}
$$

$$
\begin{aligned}
[2,1] =& 7 \cdot -1 + 5 \cdot 0 + 4 \cdot 1 \\
& 5 \cdot -2 + 7 \cdot 0 + 0 \cdot 2 \\
& 1 \cdot -1 + 6 \cdot 0 + 6 \cdot 1 \\
=& -8
\end{aligned}
$$

$$
\begin{aligned}
[3,1] =& 5 \cdot -1 + 4 \cdot 0 + 6 \cdot 1 \\
& 7 \cdot -2 + 0 \cdot 0 + 7 \cdot 2 \\
& 6 \cdot -1 + 6 \cdot 0 + 3 \cdot 1 \\
=& -2
\end{aligned}
$$

$$
\begin{aligned}
[4,1] =& 4 \cdot -1 + 6 \cdot 0 + 0 \cdot 1 \\
& 0 \cdot -2 + 7 \cdot 0 + 0 \cdot 2 \\
& 6 \cdot -1 + 3 \cdot 0 + 0 \cdot 1 \\
=& -10
\end{aligned}
$$

Third row

$$
\begin{aligned}
[0,2] =& 0 \cdot -1 + 4 \cdot 0 + 5 \cdot 1 \\
& 0 \cdot -2 + 7 \cdot 0 + 1 \cdot 2 \\
& 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
=& 7
\end{aligned}
$$

$$
\begin{aligned}
[1,2] =& 4 \cdot -1 + 5 \cdot 0 + 7 \cdot 1 \\
& 7 \cdot -2 + 1 \cdot 0 + 6 \cdot 2 \\
& 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
=& 1
\end{aligned}
$$

$$
\begin{aligned}
[2,2] =& 5 \cdot -1 + 7 \cdot 0 + 0 \cdot 1 \\
& 1 \cdot -2 + 6 \cdot 0 + 6 \cdot 2 \\
& 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
=& 5
\end{aligned}
$$

$$
\begin{aligned}
[3,2] =& 7 \cdot -1 + 0 \cdot 0 + 7 \cdot 1 \\
& 6 \cdot -2 + 6 \cdot 0 + 3 \cdot 2 \\
& 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
=& -6
\end{aligned}
$$

$$
\begin{aligned}
[4,2] =& 0 \cdot -1 + 7 \cdot 0 + 0 \cdot 1 \\
& 6 \cdot -2 + 3 \cdot 0 + 0 \cdot 2 \\
& 0 \cdot -1 + 0 \cdot 0 + 0 \cdot 1 \\
=& -12
\end{aligned}
$$

Table 2: Final result after filtering with Sobel kernel

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

# Task 2: Programming

## Task 2a

**Question**
Implement a function that converts an RGB image to greyscale. Use Equation 1. Implement this in the function greyscale.

**Answer**

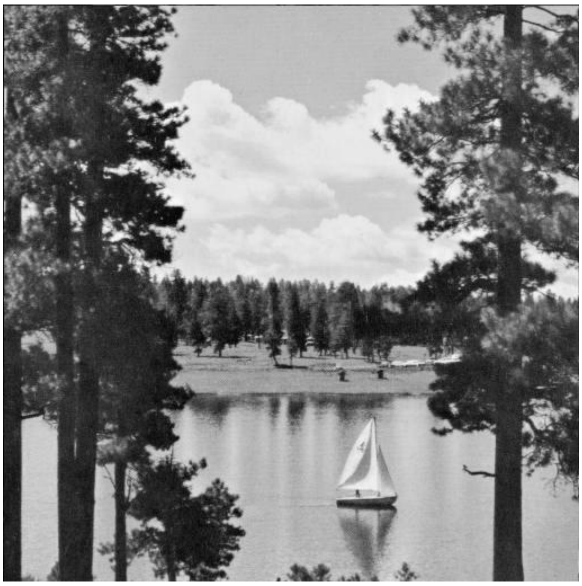

Figure 1: Greyscale

## Task 2b

**Question**
Implement a function that takes a grayscale image and applies the following intensity transformation $T(p) = 1 \; p$. Implement this in the function inverse

**Answer**

Figure 2: Greyscale inverse

## 0.1 Task 2c

**Question**
Implement a function that takes an RGB image and a convolutional kernel as input, and performs 2D spatial convolution. Assume the size of the kernel is odd numbered, e.g. $3 \times 3$, $5 \times 5$, or $7 \times 7$. You must implement the convolution operation yourself from scratch. Implement the function in convolve_im. You are not required to implement a procedure for adding or removing padding (you can return zero in cases when the convolutional kernel goes outside the original image). **Answer**

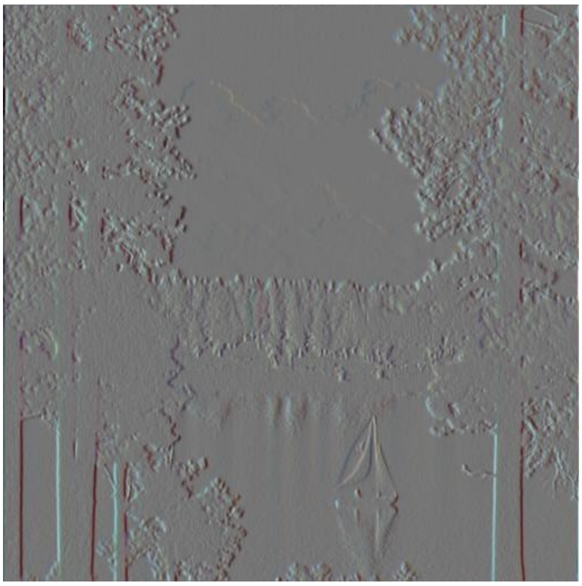

Figure 3: Lake with sobel

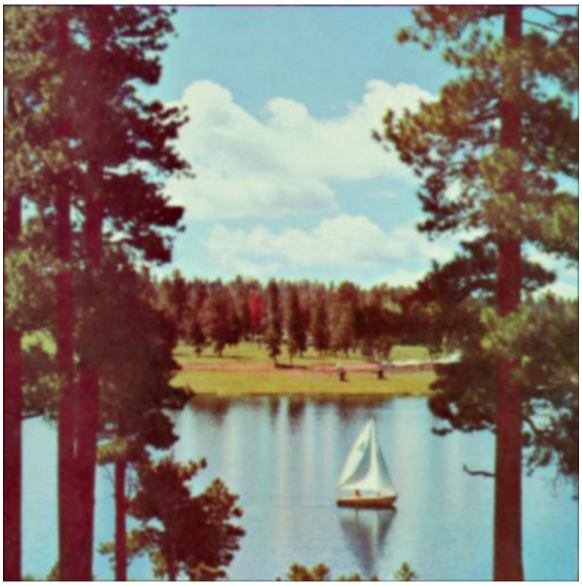

Figure 4: Lake with smoothed

# Task 23: Neural Networks

## Task 3a

**Question**
A single-layer neural network is a linear function. Give an example of a binary operation that a single-layer neural network cannot represent (either AND, OR, NOT, NOR, NAND, or XOR).

**Answer**
A single layer neural network cannot represent a XOR, because XOR is not linearly separable.

## Task 3b

**Question**
Explain in one sentence what a hyperparameter for a neural network is. Give two examples of a hyperparameter.

**Answer**
In contrast to other parameters in a network, a hyperparameter is a parameter that is predetermined by the designer of the network. Some examples of hyperparameters could be batch size as tuneability and degree of polynom to fit as trainability.

## Task 3c

**Question**
Why is the softmax activation functioned used in the last layer for neural networks trained to classify objects?

**Answer**
The function is used to get an even probabilistic distribution among the outputs such that the sum of probabilities always equals 100%. This is done by scaling.

## Task 3d

**Question**
Figure 2 shows a simple neural network. Perform a forward pass and backward pass on this network with the given input values. Use Equation 3 as the cost function and let the target value be y = 1. Explain each step in the computation, such that it is clear how you compute the derivatives.

**Answer**

We have that

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

$$\frac{\partial C}{\partial y} = y_n - \hat{y}_n$$

using $y = 1 \Rightarrow$

$$\frac{\partial C}{\partial y} = 1 - \hat{y}_n$$

$\Rightarrow$

Summarizing every contribution when doing forward pass and applying equation above where $\hat{y} = c, b, a, w$ yields:

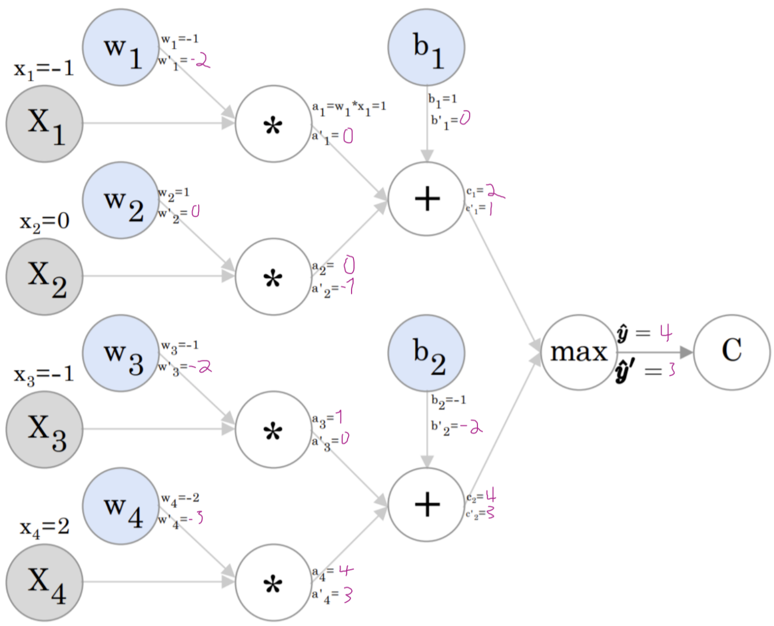


Figure 5: Neural network filled out.

## Task 3e

**Question**

Compute the updated weights w1, w3, and b1 by using gradient descent and the values you found in task d. Use $\alpha = 0.1$

**Answer**

$$\begin{aligned}\theta_{t+1} &= \theta_1 - \alpha \frac{\partial C}{\partial \theta_t} \\ &= \theta_1 - 0.1 \frac{\partial C}{\partial \theta_t}\end{aligned}$$

Applying equation above (where $\theta$ is replaced by $w_1, w_3, b_1$) in combination with results from 3d) yields:

$$\begin{aligned}w_1 &= -1 - 0.1 \cdot -2 \\ &= -0.8\end{aligned}$$

$$w_3 = -1 - 0.1 \cdot -2$$
$$= -0.8$$

$$b_1 = 1 - 0.1 \cdot 0$$
$$= 1$$

## 0.2 Task 4a

**Question**

Use the given starter code and train a single-layer neural network with batch size of 64. Then, normalize every image between a range of [-1. 1], and train the network again. Plot the training and validation loss from both of the networks in the same graph. Include the graph in your report. Do you notice any difference when training your network with/without normalization?

**Answer**

The normalized has much sharper curve and the loss will faster stabilize than the none normalized. The same result could be archived with the normalized images using fewer training steps than without normalization. Something I noticed and couldn't quite understand is that the normalized data was slower to train, do you know why?

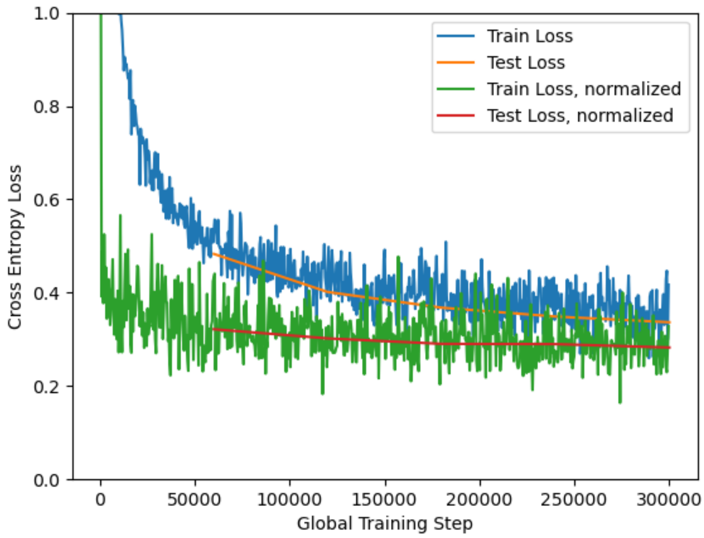


Figure 6: Difference between normalized

## 0.3 Task 4b

**Question**

The trained neural network will have one weight with shape [num classes, 28 × 28]. To visualize the learned weight, we can plot the weight as a $28 \times 28$ grayscale image. For each digit (0-9), plot the learned weight as a $28 \times 28$ image. In your report, include the image for each weight, and describe what you observe (1-2 sentences).

**Answer**

It looks like the weights looks like the number it represents. It can look like heat map of where the number can be.

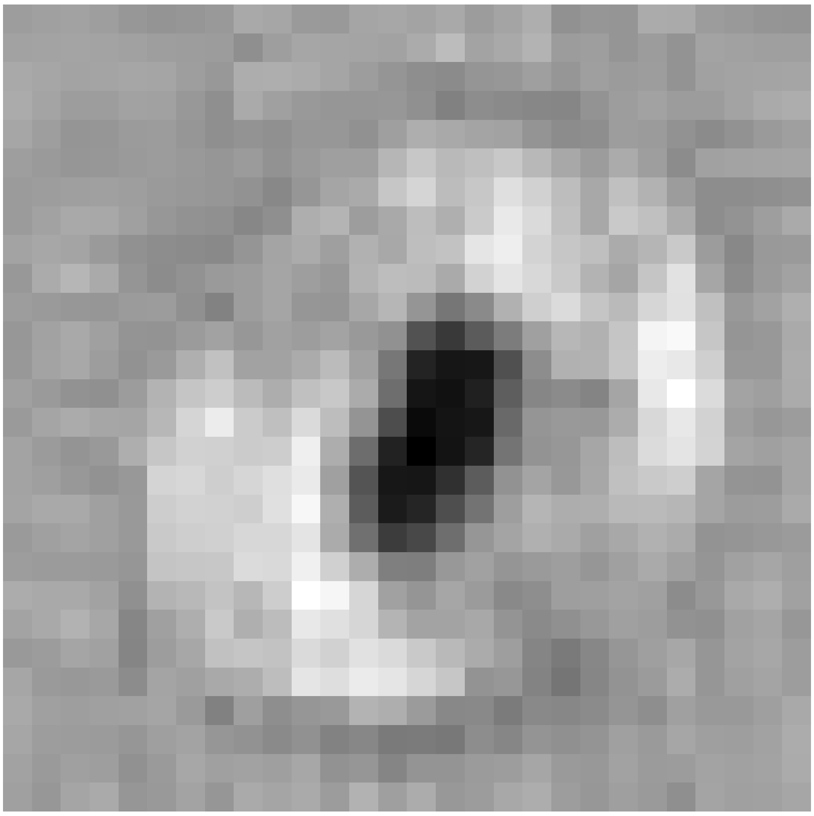

Figure 7: Visualization of number 0 weights

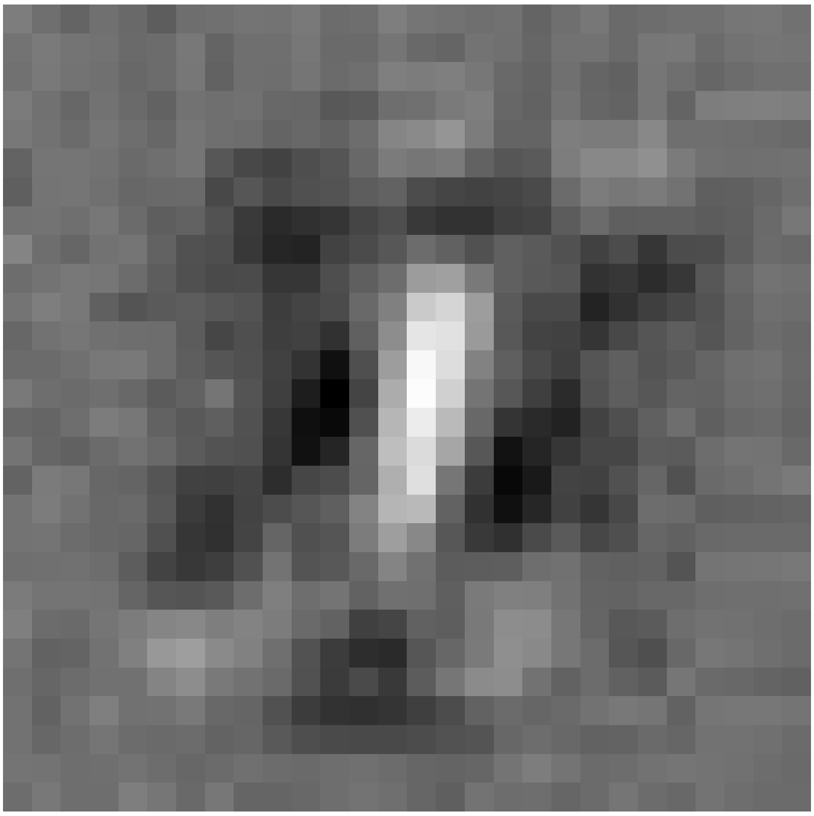

Figure 8: Visualization of number 1 weights.

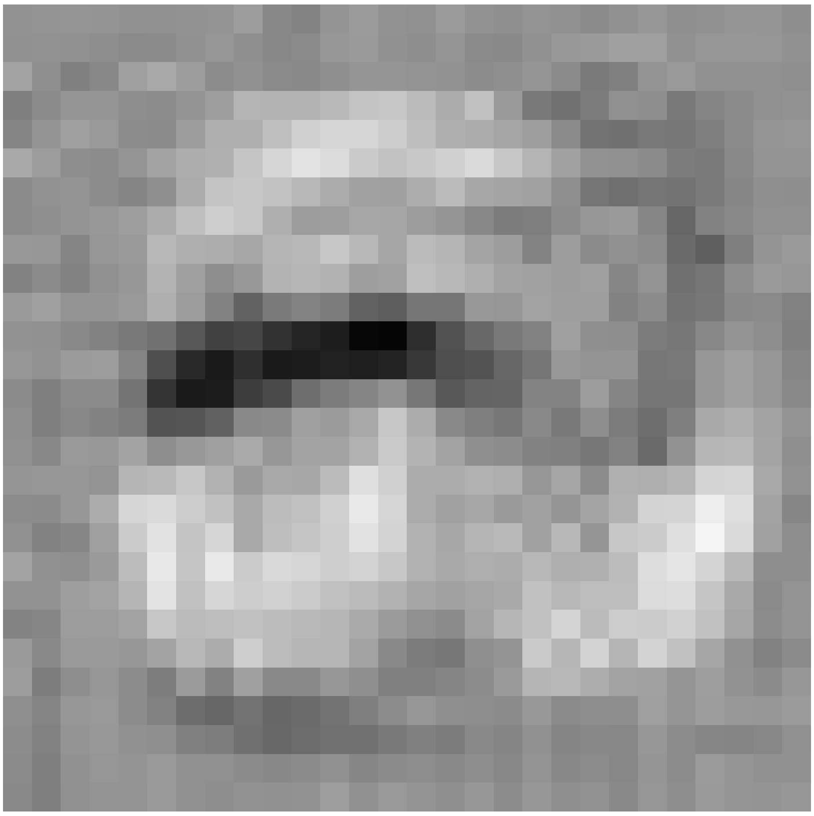

Figure 9: Visualization of number 2 weights.

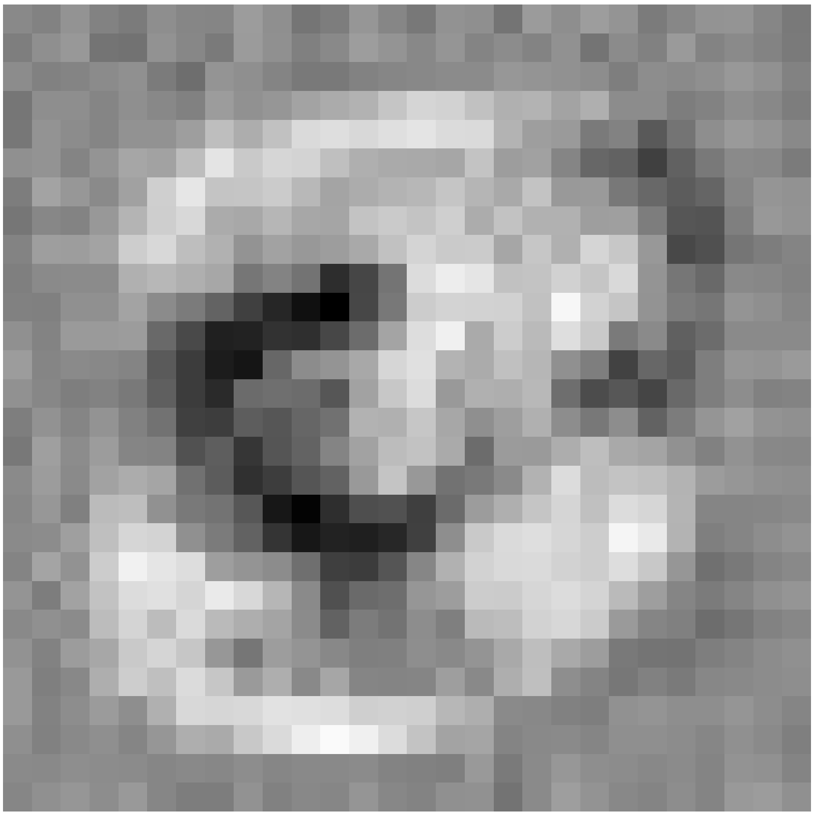

Figure 10: Visualization of number 3 weights.

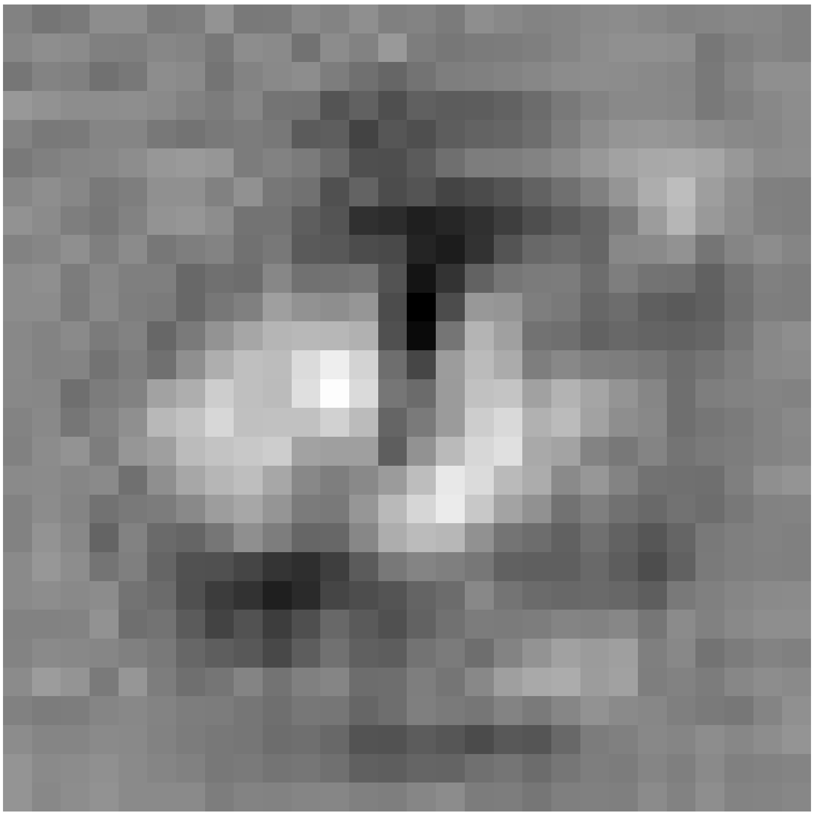

Figure 11: Visualization of number 4 weights.

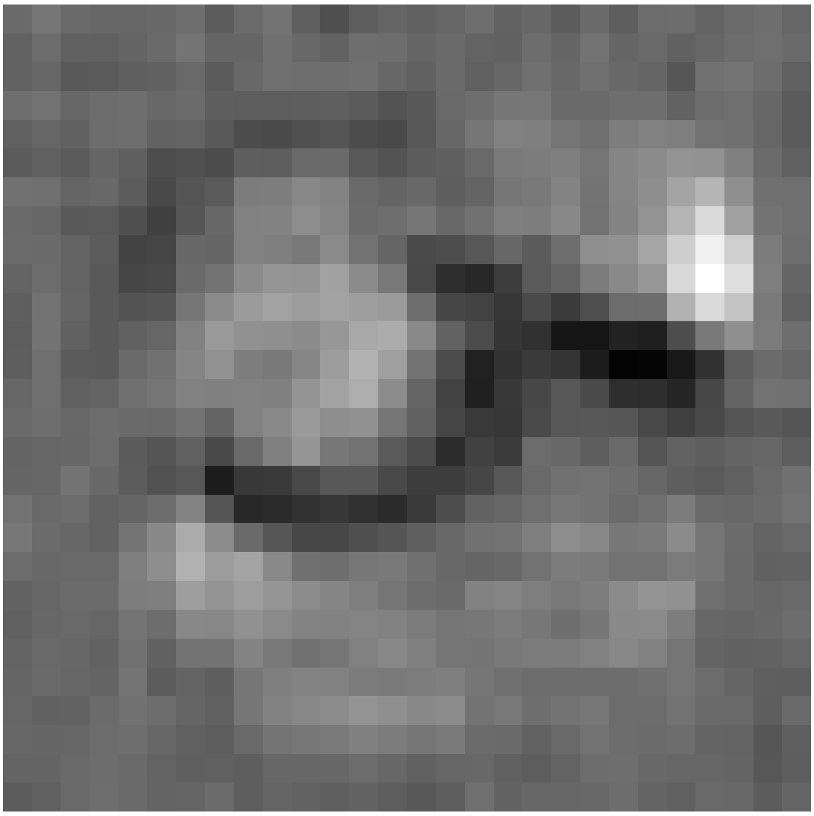

Figure 12: Visualization of number 5 weights.

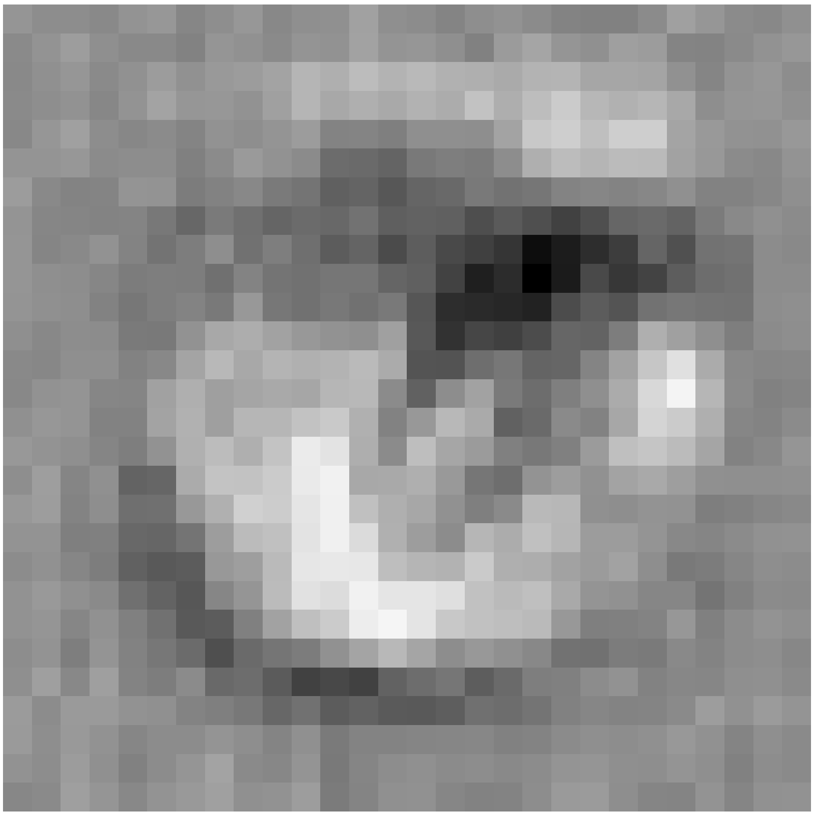

Figure 13: Visualization of number 6 weights.

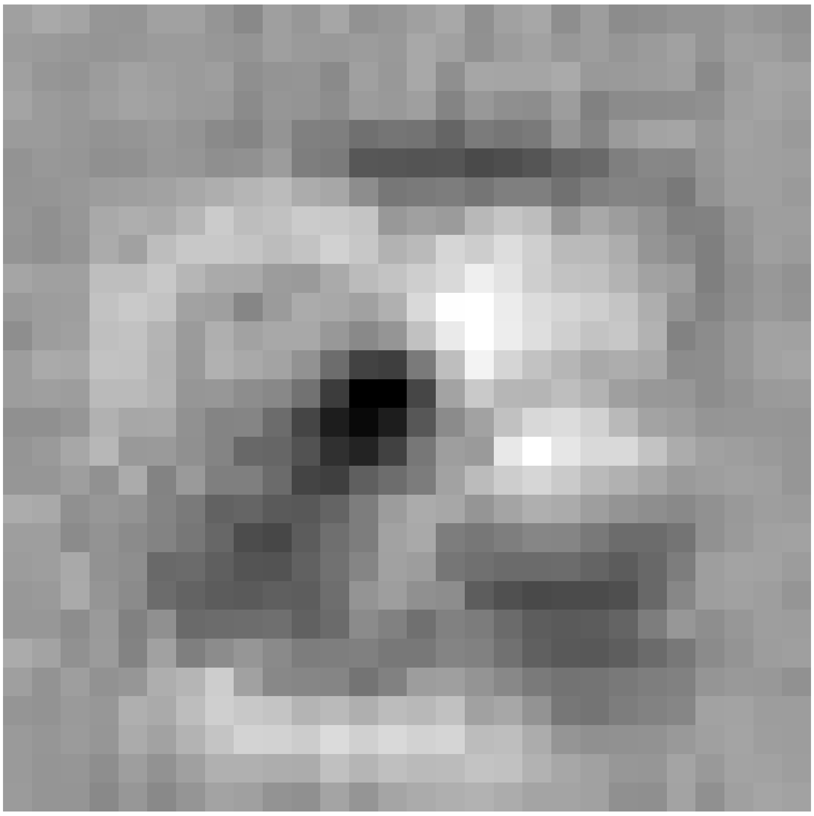

Figure 14: Visualization of number 7 weights.

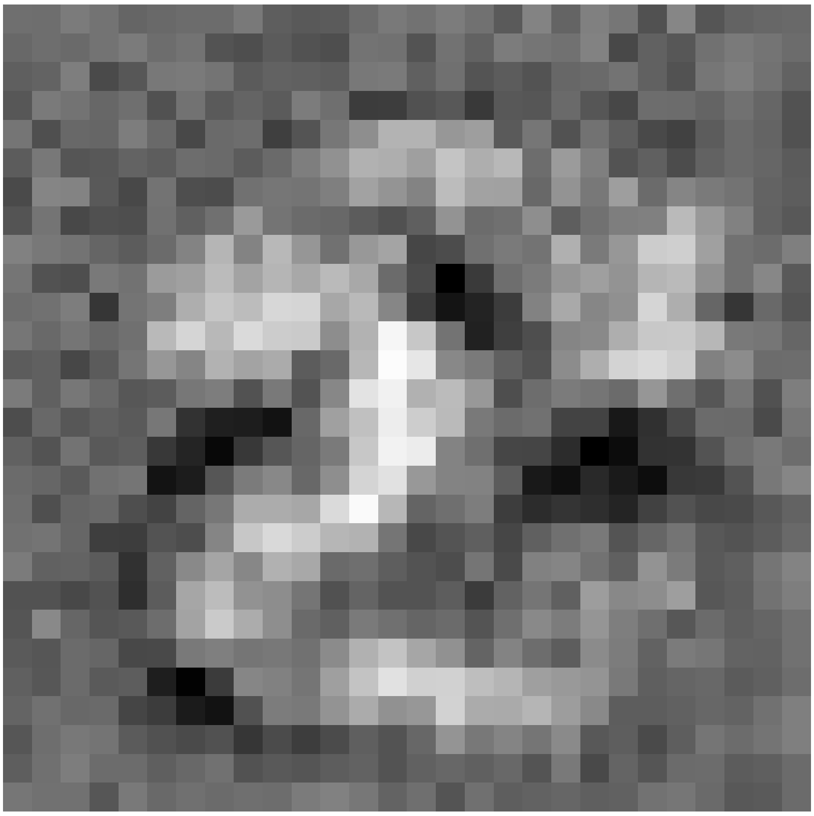

Figure 15: Visualization of number 8 weights.

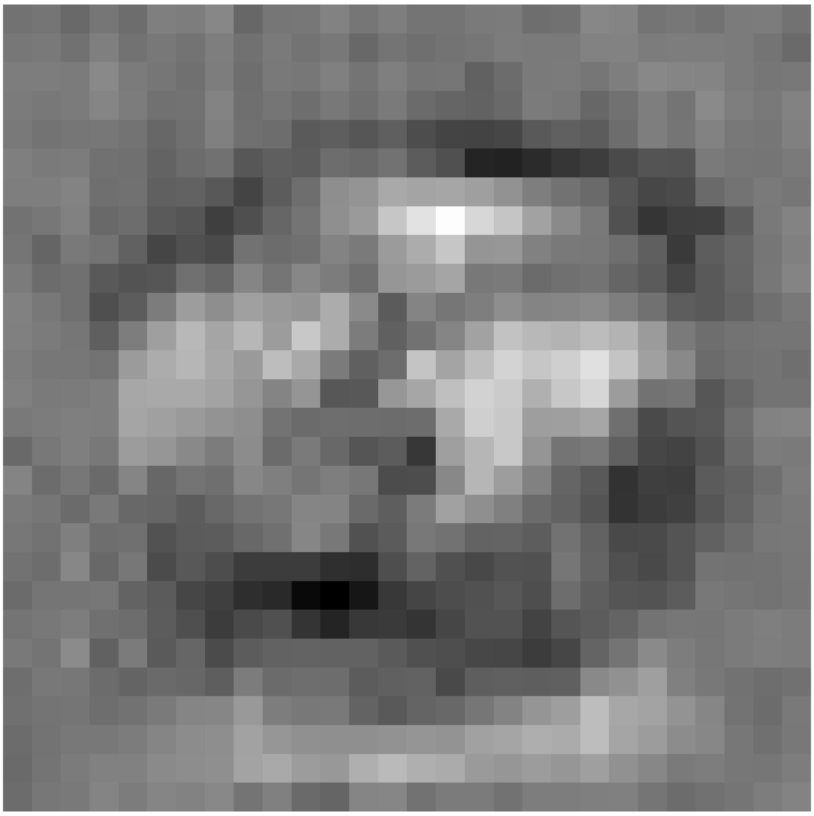

Figure 16: Visualization of number 9 weights.

## 0.4 Task 4c

**Question**
Set the learning rate to $lr = 1.0$, and train the network from scratch. Report the accuracy and average cross entropy loss on the validation set. In 1-2 sentences, explain why the network achieves worse/better accuracy than previously.

**Answer**
When the learning rate is that big it will struggle to find the best solution for the gradient. It will most likely ”step over ” the ideal solution.

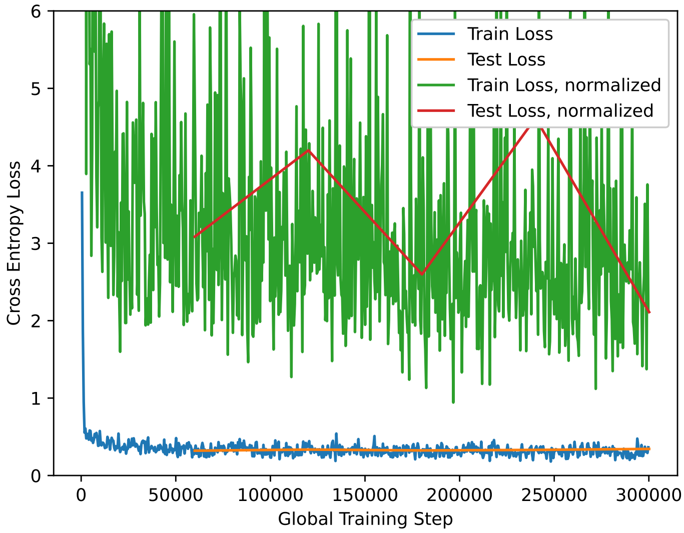


Figure 17: With learning rate = 1.

## 0.5 Task 4d

**Question**

Include an hidden layer with 64 nodes in the network, with ReLU as the activation function for the first layer. Train this network with the same hyperparameters as previously. Plot the training and validation loss from this network together with the loss from task (a). Include the plot in your report. What do you observe?

**Answer**

We can see that with another hidden layer we get a bit better results.

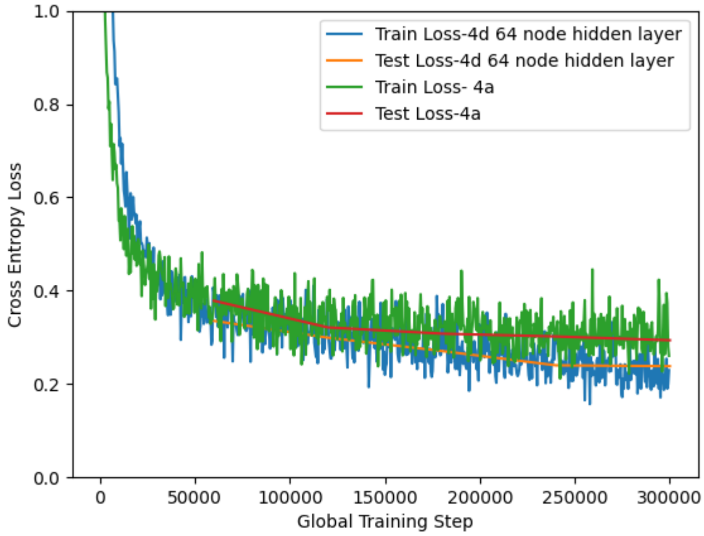


Figure 18: An extra hidden layer.

We can see that the curve in figure 18, so I tested with more epochs and as seen the results keep improving for the model with an extra layer but not much for the single layer one.

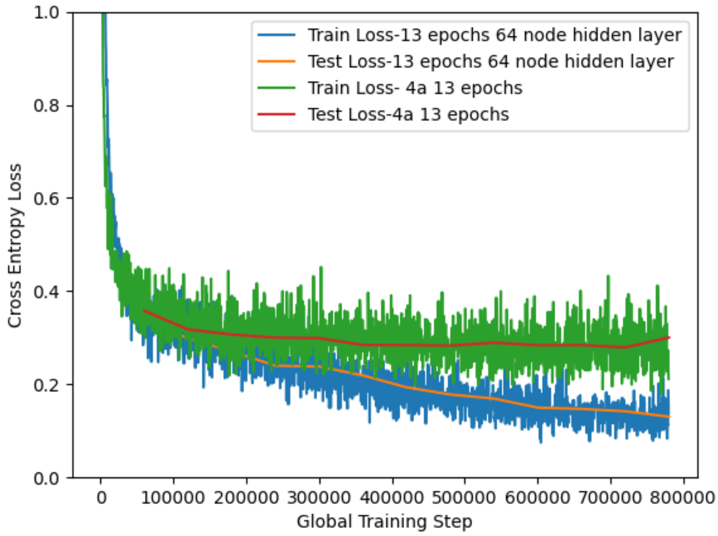


Figure 19: An extra hidden layer and 13 epochs.

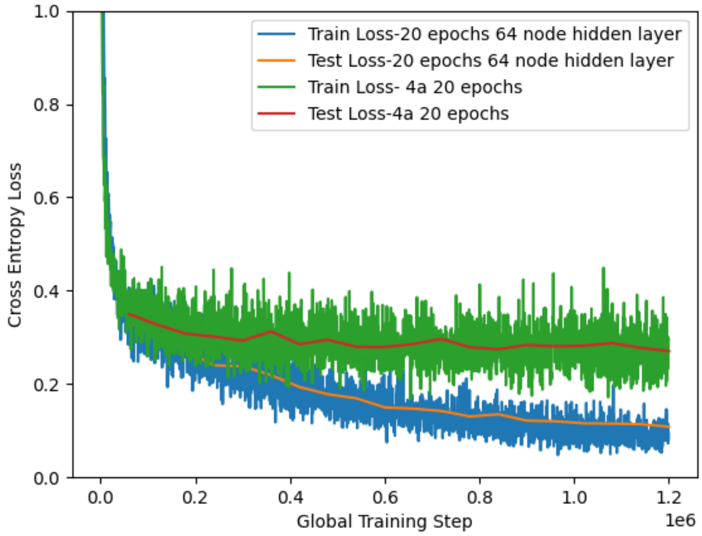


Figure 20: An extra hidden layer and 20 epochs.