TDT4195 - Visual Computing Fundamentals

Image Processing - Assignment 1

Author:
Martin Skatvedt

19.10.2022

1 Spatial Filtering

1.1 Task 1

a) Sampling is the process of converting an analog signal into discrete values, a higher sampling rate results in higher image quality.

b) Quantization is the process of determining the tone for each pixel, this is done by taking the amplitude of the sampled image.

c) We can see that an iamge has a high contrast by the histogram having a broad range in the x-axis, and narrow tops which are far apart.

d)

Intensity	Sum	Normalized sum	Floored normalized sum
0	1	0.533	0
1	1	0.533	0
2	0	0	0
3	1	0.533	0
4	2	1.066	1
5	2	1.066	1
6	4	2.133	2
7	4	2.133	2

Table 1: Table for computing the histogram and normalized historgram

2	2	1	2	1
1	1	2	2	0
0	2	2	0	2

Table 2: Transformed image

e)

By using a log transformation on an image with a large variance in pixel intensities, the dynamic range gets compressed.

f)

To handle boundary conditions I padded the image with 0, to and produced the image

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

Table 3: Padded image

Example of convolution of x_{11}

$$0*1+0*0+0*(-1)+0*2+7*0+6*(-2)+0*1+5*0+4*(-1) = 6*(-2)+4*(-1) = -16$$

Performing this operation on x_{ij} for $0 < i < 6$ and $0 < j < 4$ produced the image

-16	2	-3	9	19
-21	-7	-2	15	23
-18	-12	5	8	13

Table 4: Result of convolution by hand

1.2 Task 2

a)

Figure 1: Grayscale image of lake

b)

Figure 2: Inverse image of lake

c)

Figure 3: Image convolved with the approximated gaussian kernel

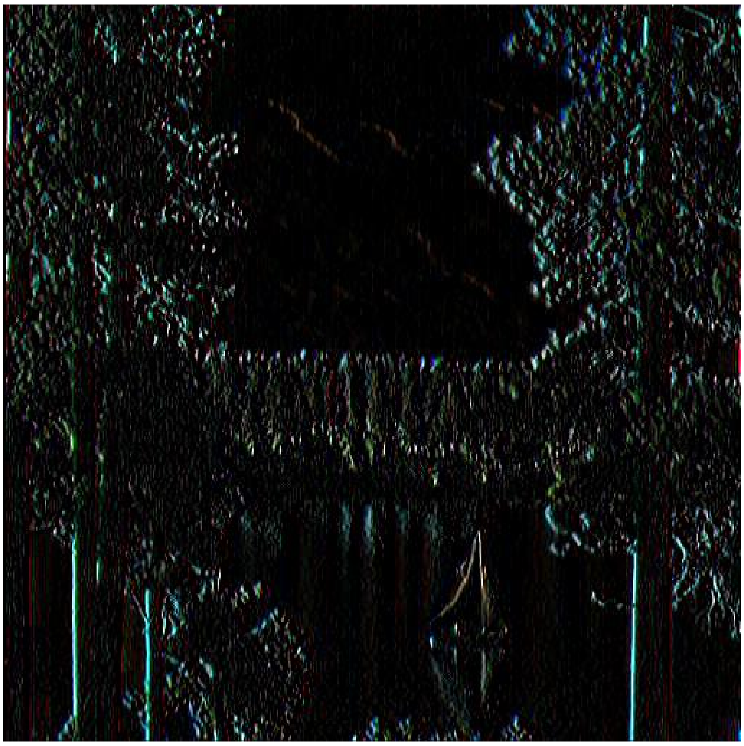

Figure 4: Image convolved with the sobel kernel

2 Neural Networks

2.1 Task 3

a) XOR can not be represented by a single-layer neural network.

b) A Hyperparameter for a neural network is a variable which determines the network structure. Such hyperparameters can be the learning rate and the activation functions.

c) The softmax function takes in a vector with size K and outputs a vector with size k, where each number is normalized in (0,1). This means that we can take the highest number, which is the highest propability to classify our input.

d)

Forward pass:

$$a_1 = w_1 * x_1 = -1 * -1 = 1$$

$$a_2 = w_2 * x_2 = 1 * 0 = 0$$

$$a_3 = w_3 * x_3 = -1 * -1 = 1$$

$$a_4 = w_4 * x_4 = -2 * 2 = 4$$

$$c_1 = a_1 + a_2 + b_1 = 1 + 0 + 1 = 2$$

$$c_2 = a_3 + a_4 + b_2 = 1 - 4 - 1 = -4$$

$$\hat{y} = max(c_1, c_2) = max(2, -4) = 2$$

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

Backward pass:

Calculating for weights

Calculating 1st term

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

Calculating 2nd term

$\frac{\delta \hat{y}}{\delta c_1} = \begin{cases} 1 & \text{if } c_1 \geq c_2 \\ 0 & \text{if } c_1 < c_2 \end{cases}$

$\frac{\delta \hat{y}}{\delta c_2} = \begin{cases} 1 & \text{if } c_2 \geq c_1 \\ 0 & \text{if } c_2 < c_1 \end{cases}$

Calculating 3th term

$\frac{\delta c_1}{\delta a_1} = \frac{\delta a_1 + a_2 + b_1}{\delta a_1} = 1$

$\frac{\delta c_1}{\delta a_2} = \frac{\delta a_1 + a_2 + b_1}{\delta a_2} = 1$

$\frac{\delta c_2}{\delta a_3} = \frac{\delta a_3 + a_4 + b_2}{\delta a_3} = 1$

$\frac{\delta c_2}{\delta a_4} = \frac{\delta a_3 + a_4 + b_2}{\delta a_4} = 1$

Calculating 4th term

$\frac{\delta a_1}{\delta w_1} = \frac{\delta w_1 * x_1}{\delta w_1} = x_1$

$\frac{\delta a_2}{\delta w_2} = \frac{\delta w_2 * x_2}{\delta w_2} = x_2$

$\frac{\delta a_3}{\delta w_3} = \frac{\delta w_3 * x_3}{\delta w_3} = x_3$

$\frac{\delta a_4}{\delta w_4} = \frac{\delta w_4 * x_4}{\delta w_4} = x_4$

By using the calculated terms, we can calculate the derivative for the cost function with respect to each weight

$$\frac{\delta C}{\delta w_1} = \frac{\delta C}{\delta \hat{y}} * \frac{\delta \hat{y}}{\delta c_1} * \frac{\delta c_1}{\delta a_1} * \frac{\delta a_1}{\delta w_1} = -(y - \hat{y}) * \begin{cases} 1 & \text{if } c_1 \geq c_2 \\ 0 & \text{if } c_1 < c_2 \end{cases} * 1 * x_1 = -(1-2) * 1 * 1 * -1 = -1$$

$$\frac{\delta C}{\delta w_2} = \frac{\delta C}{\delta \hat{y}} * \frac{\delta \hat{y}}{\delta c_1} * \frac{\delta c_1}{\delta a_2} * \frac{\delta a_2}{\delta w_2} = -(y - \hat{y}) * \begin{cases} 1 & \text{if } c_1 \geq c_2 \\ 0 & \text{if } c_1 < c_2 \end{cases} * 1 * x_2 = -(1-2) * 1 * 1 * 0 = 0$$

$$\frac{\delta C}{\delta w_3} = \frac{\delta C}{\delta \hat{y}} * \frac{\delta \hat{y}}{\delta c_2} * \frac{\delta c_2}{\delta a_3} * \frac{\delta a_3}{\delta w_3} = -(y - \hat{y}) * \begin{cases} 1 & \text{if } c_2 \geq c_1 \\ 0 & \text{if } c_2 < c_1 \end{cases} * 1 * x_3 = -(1-2) * 0 * 1 * -1 = 0$$

$$\frac{\delta C}{\delta w_4} = \frac{\delta C}{\delta \hat{y}} * \frac{\delta \hat{y}}{\delta c_2} * \frac{\delta c_2}{\delta a_4} * \frac{\delta a_4}{\delta w_4} = -(y - \hat{y}) * \begin{cases} 1 & \text{if } c_2 \geq c_1 \\ 0 & \text{if } c_2 < c_1 \end{cases} * 1 * x_4 = -(1-2) * 0 * 1 * 2 = 0$$

Calculating for bias

Calculating 1st term

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

Calculating 2nd term

$\frac{\delta \hat{y}}{\delta c_1} = \begin{cases} 1 & \text{if } c_1 \geq c_2 \\ 0 & \text{if } c_1 < c_2 \end{cases}$

$\frac{\delta \hat{y}}{\delta c_2} = \begin{cases} 1 & \text{if } c_2 \geq c_1 \\ 0 & \text{if } c_2 < c_1 \end{cases}$

Calculating 3th term

$\frac{\delta c_1}{\delta b_1} = \frac{\delta a_1 + a_2 + b_1}{\delta b_1} = 1$

$\frac{\delta c_2}{\delta b_2} = \frac{\delta a_3 + a_4 + b_2}{\delta b_2} = 1$

By using the calculated terms, we can calculate the derivative for the cost function with respect to each bias

$$\frac{\delta C}{\delta b_1} = \frac{\delta C}{\delta \hat{y}} * \frac{\delta \hat{y}}{\delta c_1} * \frac{\delta c_1}{\delta b_1} = -(y - \hat{y}) * \begin{cases} 1 & \text{if } c_1 \geq c_2 \\ 0 & \text{if } c_1 < c_2 \end{cases} * 1 = -(1-2) * 1 * 1 = -1$$

$$\frac{\delta C}{\delta b_2} = \frac{\delta C}{\delta \hat{y}} * \frac{\delta \hat{y}}{\delta c_2} * \frac{\delta c_2}{\delta b_2} = -(y - \hat{y}) * \begin{cases} 1 & \text{if } c_2 \geq c_1 \\ 0 & \text{if } c_2 < c_1 \end{cases} * 1 = -(1-2) * 0 * 1 = 0$$

e)

Formula for gradient descent for weights

$$*w_x^y = w_x^y - \alpha \frac{\delta C}{\delta w_x^y}$$

And we can calculate the new weights

$$*w_1 = -1 - (0.1 * -1) = -0.9$$

$$*w_3 = -1 - (0.1 * 0 = -1$$

Formula for gradient descent for bias

$$*b_x^y = b_x^y - \alpha \frac{\delta C}{\delta b_x^y}$$

And we can calculate the new bias

$$*b_1 = 1 - (0.1 * -1) = 1.1$$

2.2 Task 4

a)

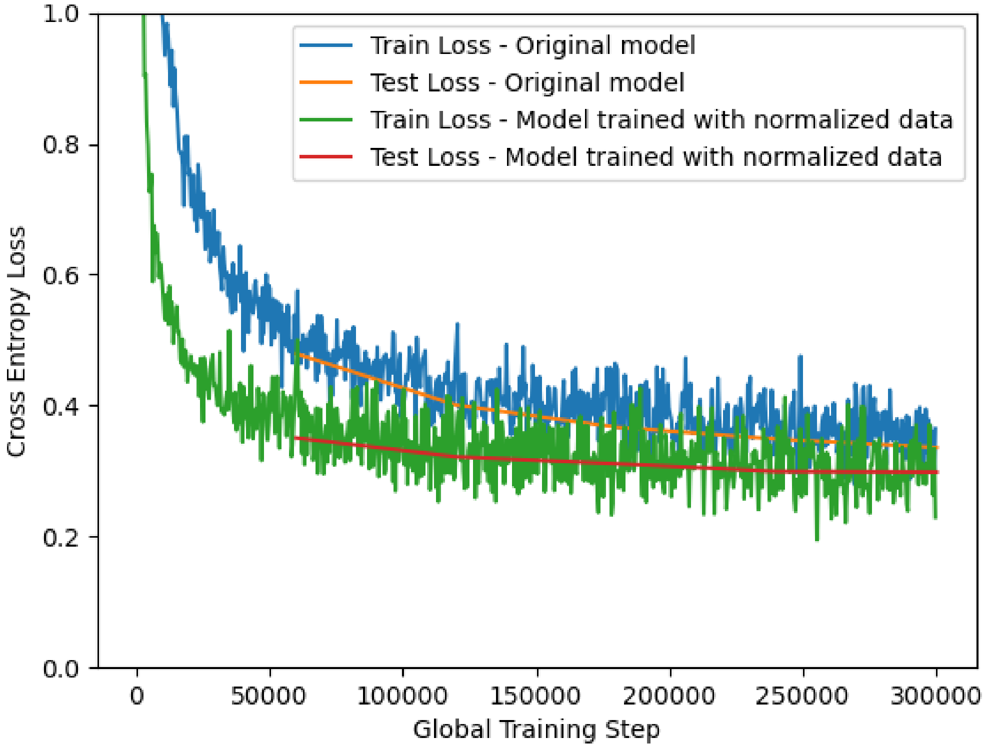

Figure 5: Plot of loss with both normalized and non-normalized data

b)

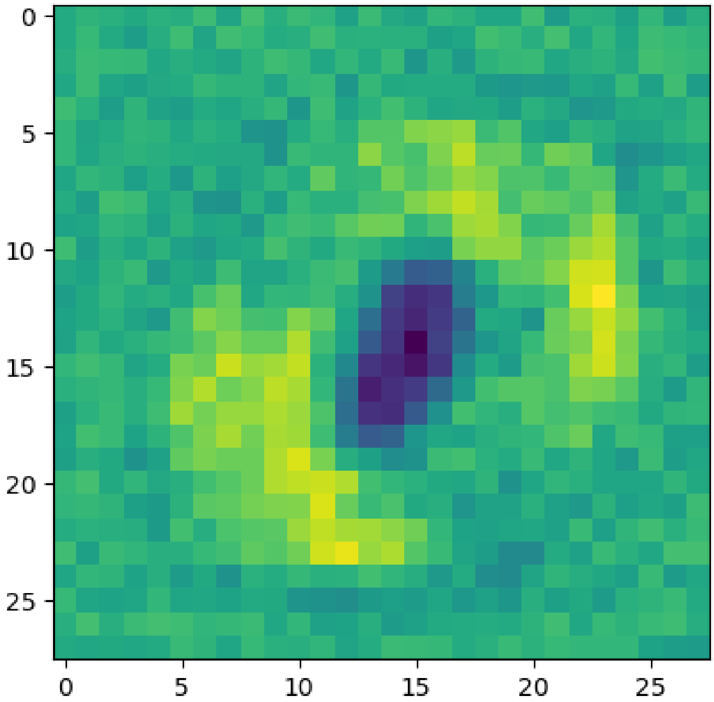

Figure 6: Weight for digit 0

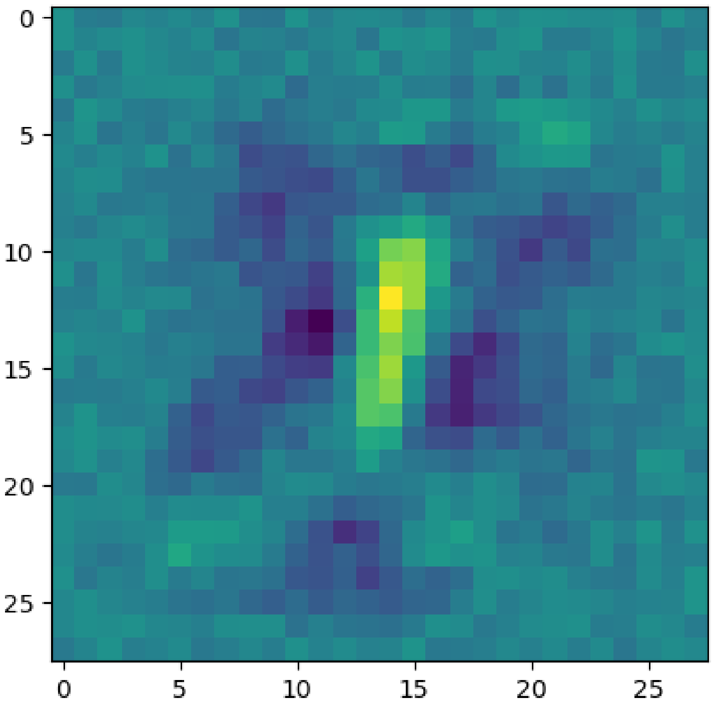

Figure 7: Weight for digit 1

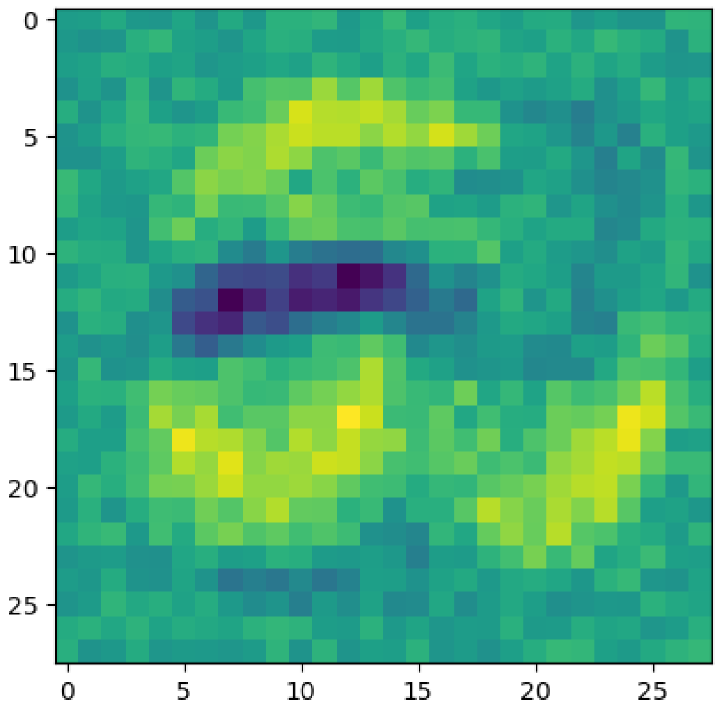

Figure 8: Weight for digit 2

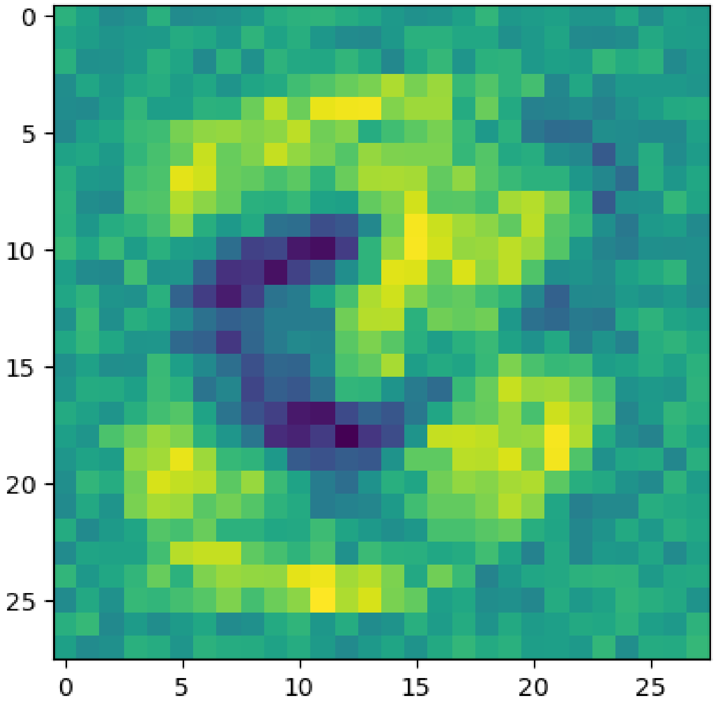

Figure 9: Weight for digit 3

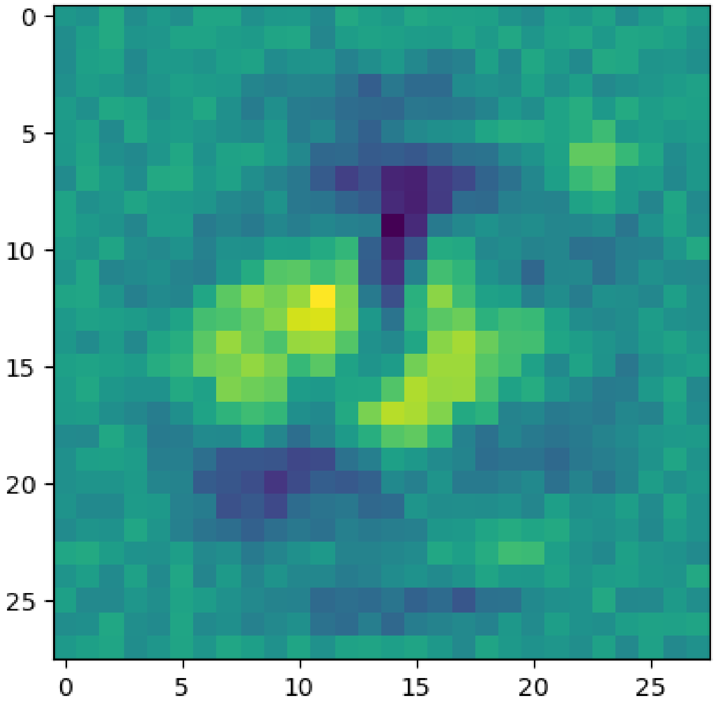

Figure 10: Weight for digit 4

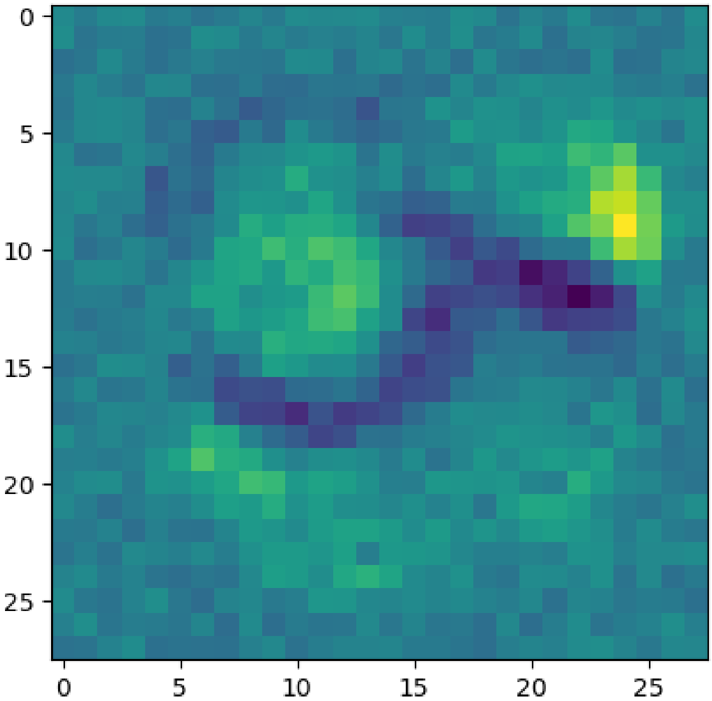

Figure 11: Weight for digit 5

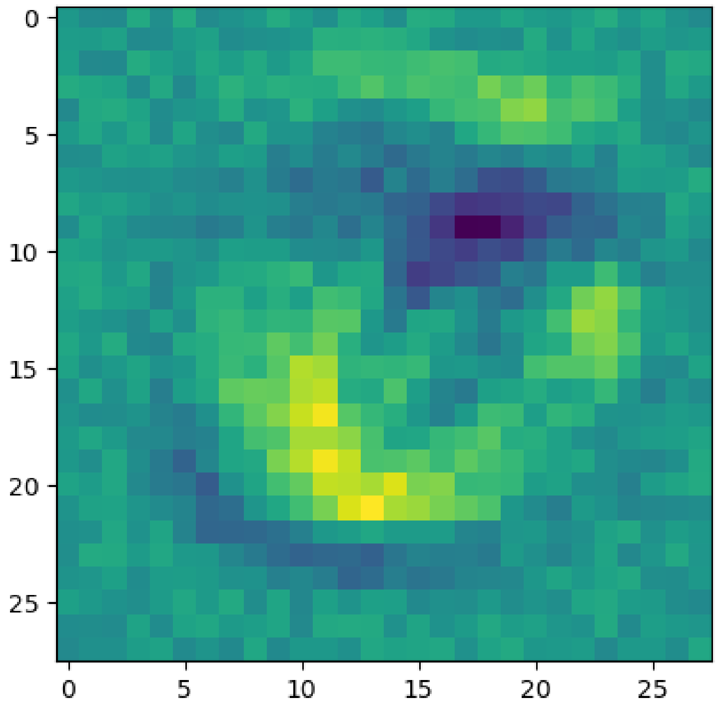

Figure 12: Weight for digit 6

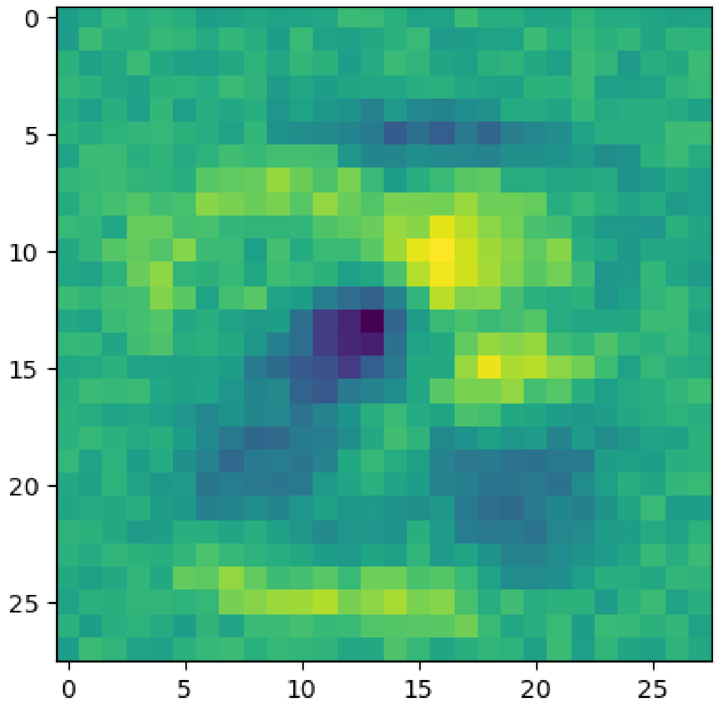

Figure 13: Weight for digit 7

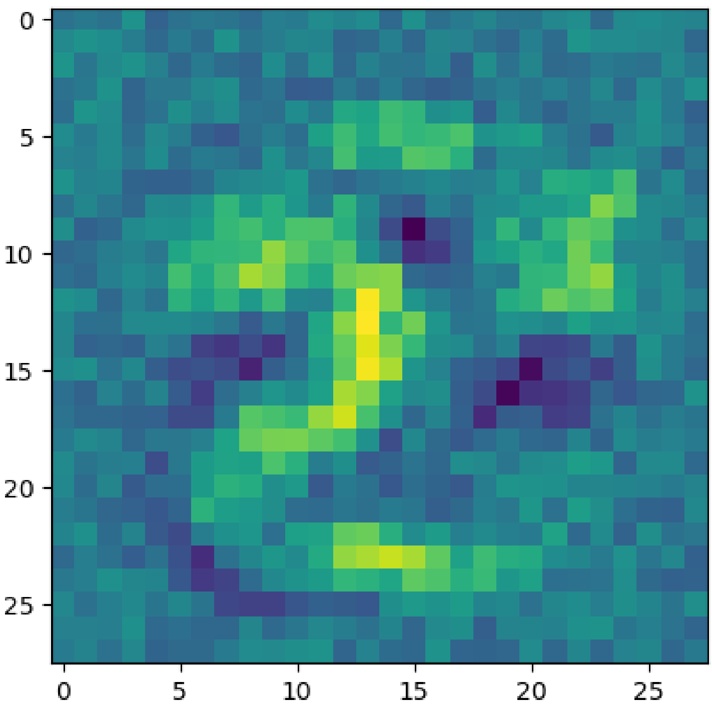

Figure 14: Weight for digit 8

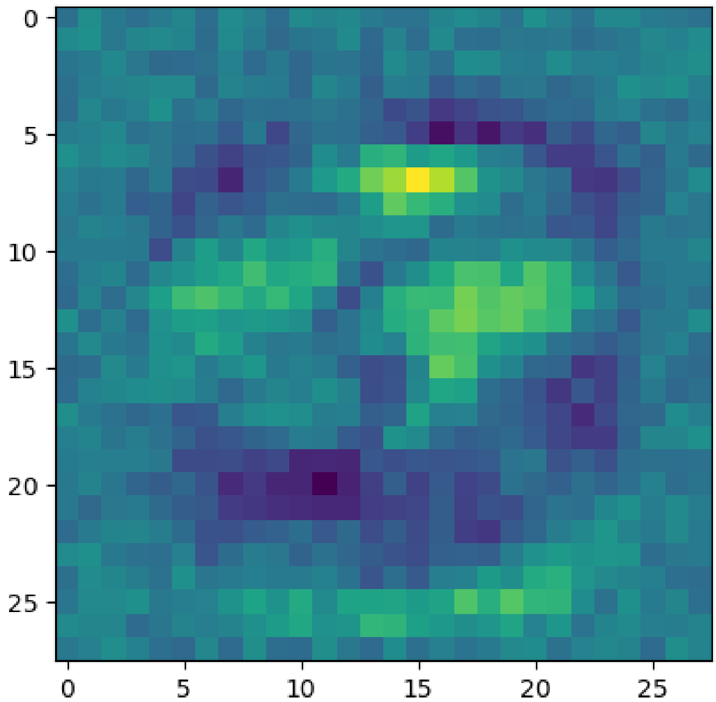

Figure 15: Weight for digit 9

By looking at the weights for each digit we can see the outline of the number we are trying to categorize. This is especially apparent for digits 0,1 and 3, where we can clearly see the number.

c)

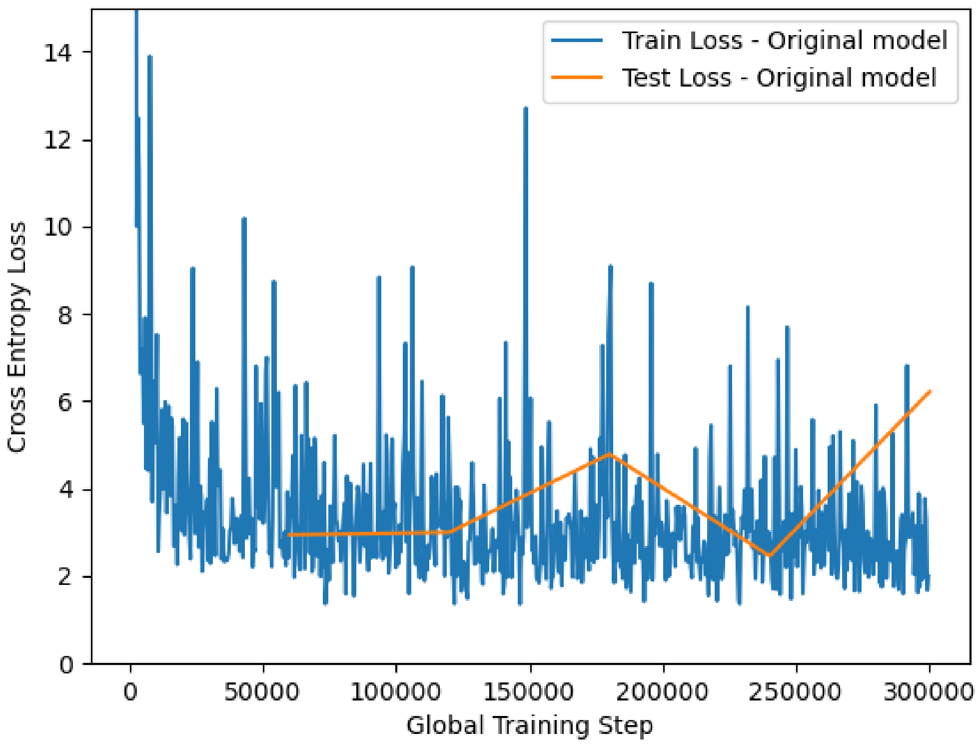

Figure 16: Network with lr=1.0: Final Test loss: 6.215373856246851. Final Test accuracy: 0.792

We can see that the network acvhieves worse accuracy than with lr=f 0.0192. This is because during the gradient descent, the weigths and biases "overshoot", and learns too quickly. This means that the network will jump over/under the desired weigths and biases, but not at the perfect spot.

d)

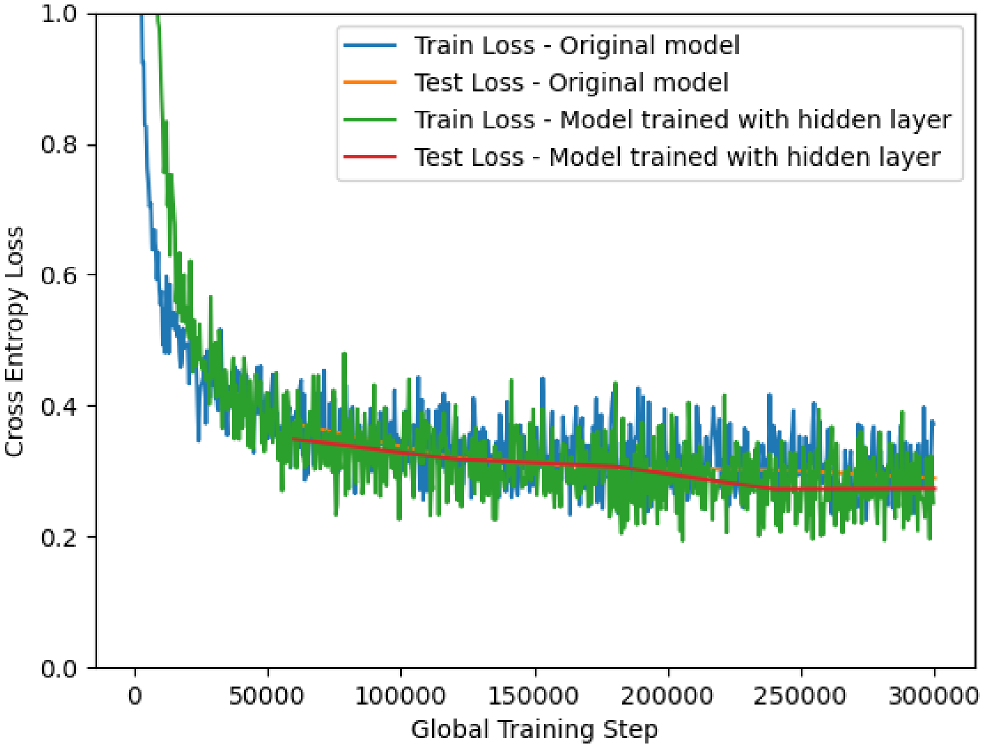

Figure 17: Plot of loss with single layer neural net, and neural neth with a hidden layer and ReLu activation

We can observe that the network with a hidden layer performs slightly better than just the single layer network.