TDT 4195 - Lab Assignment 4

Kari Lovise Ness (MTING)
Håvard Kjellmo Arnestad (MTFYMA)

October 2019

Task 1: Theory - Spatial Filtering

a) Explain in one sentence what sampling is.

Sampling is essentialy to extract a data point from a spatial or temporal signal. [So sampling rate is how often the signal is measured, and dictates the frequency/wavenumbers that unambigously can exist, ref. Nyquist].

b) Explain in one sentence what quantization is.

Quantization happens when a signal that can take on many (infinitely/continuous range) values is converted into a smaller, discrete set. The limited number of bits available means that the signal must take on discrete values, and this process is called quantization.

c) Looking at an image histogram, how can you see that the image has high contrast?

Looking at an image histogram, one can see that the image is high contrast if the histogram make extensive use of the display range. In contrast, a low-contrast image will only make use of a narrow display range. This reflects that there is little difference between light and dark areas.

d) 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

See figure 1.

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

If one applies a log transform on an image with a large variance in pixel intensities, the dynamic range is compressed.

f) Perform spatial convolution by hand on the image in Figure 1a using the kernel in Figure 1b. The convolved image should be 3 5. You are free to choose how you handle boundary conditions, and state how you handle them in the report.

See figure 2 and 3. The boundaries has been set to zero in the calculations.

Task 2: Basic Image Processing

a)

Implement a function that converts an RGB image to grayscale. Use Equation 1. (Implement this in the grayscale function in task2ab.py)

b)

c)

Task 3: Neural networks theory

(a) A single-layer neural network is a linear function. Give an example of a binary operation that a single-layer neural network cannot represent.

As seen in figure 8, XOR is an example of a binary operation that a single-layer neural network cannot represent as it is non-linear.

(b)Explain in one sentence what a hyperparameter for a neural network is. Give two examples of a hyperparameter

A hyperparameter in a neural network is a parameter with a set value before the learning starts. In other words, a hyperparameter is a variable which the network is not able to learn by itself. Examples of such parameters are the number of layers and the number of neurons in each layer of the network.

(c) Why is the softmax activation functioned used in the last layer for neural networks trained to classify objects?

The softmax function is used in the last layer of a neural network as it outputs a probability distribution. This is useful in object recognition when there exists several classes in which a object can be classified into. The output will then be a probability distribution of the different classes. For example, a neural network which receives a picture of a cat as the input should output a higher probability for the cat-class than for the dog-class and the horse-class.

(d)Perform a forward pass and backward pass on this network with the given input values.

See figure 9, 10 and 11 for calculations.

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

See figure 12.

Task 4: Neural networks programming

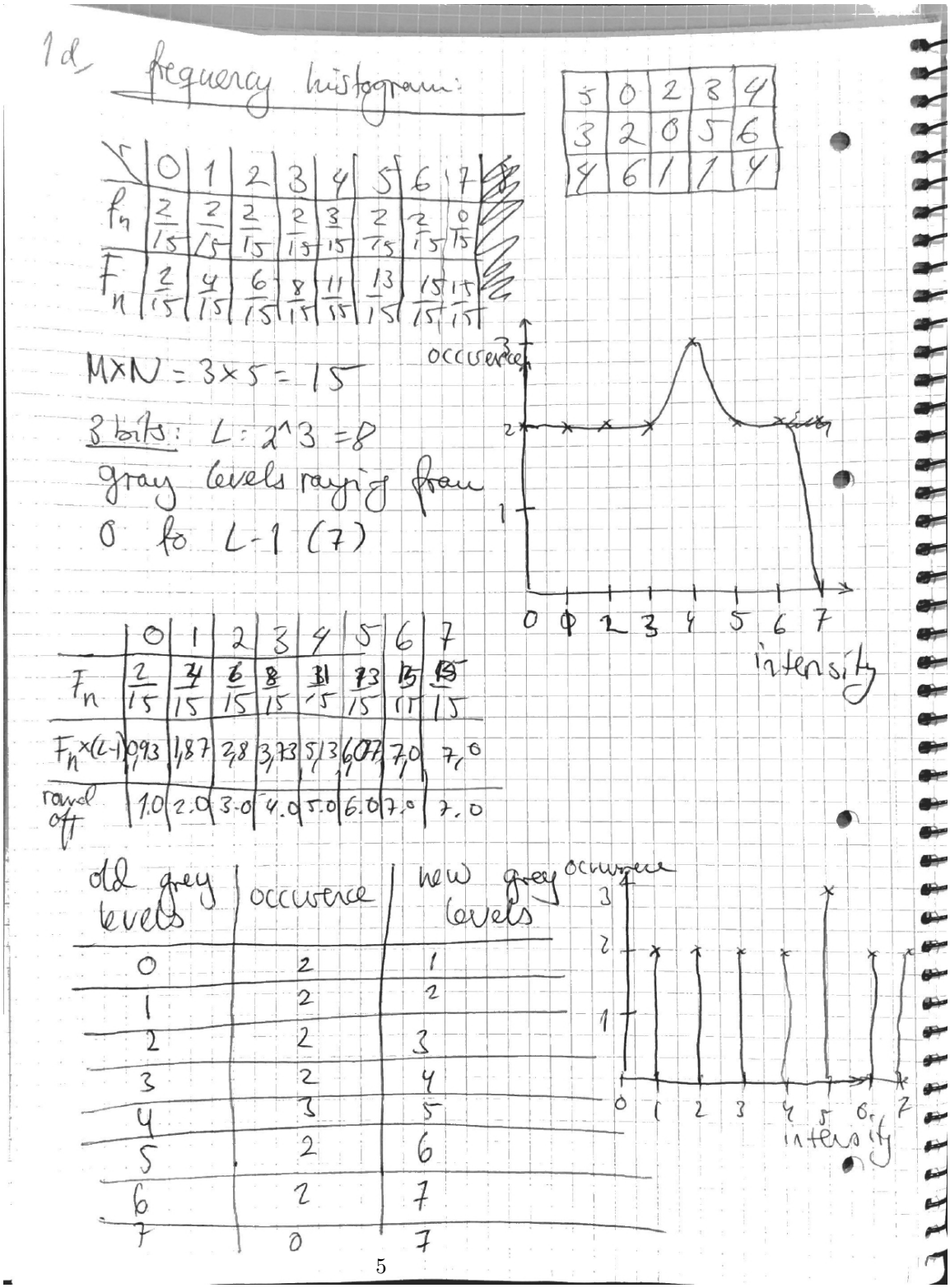
See Fig. 13 to 16.

(a) Plot the training and validation loss from both of the networks in the same graph. Include the graph in your report.

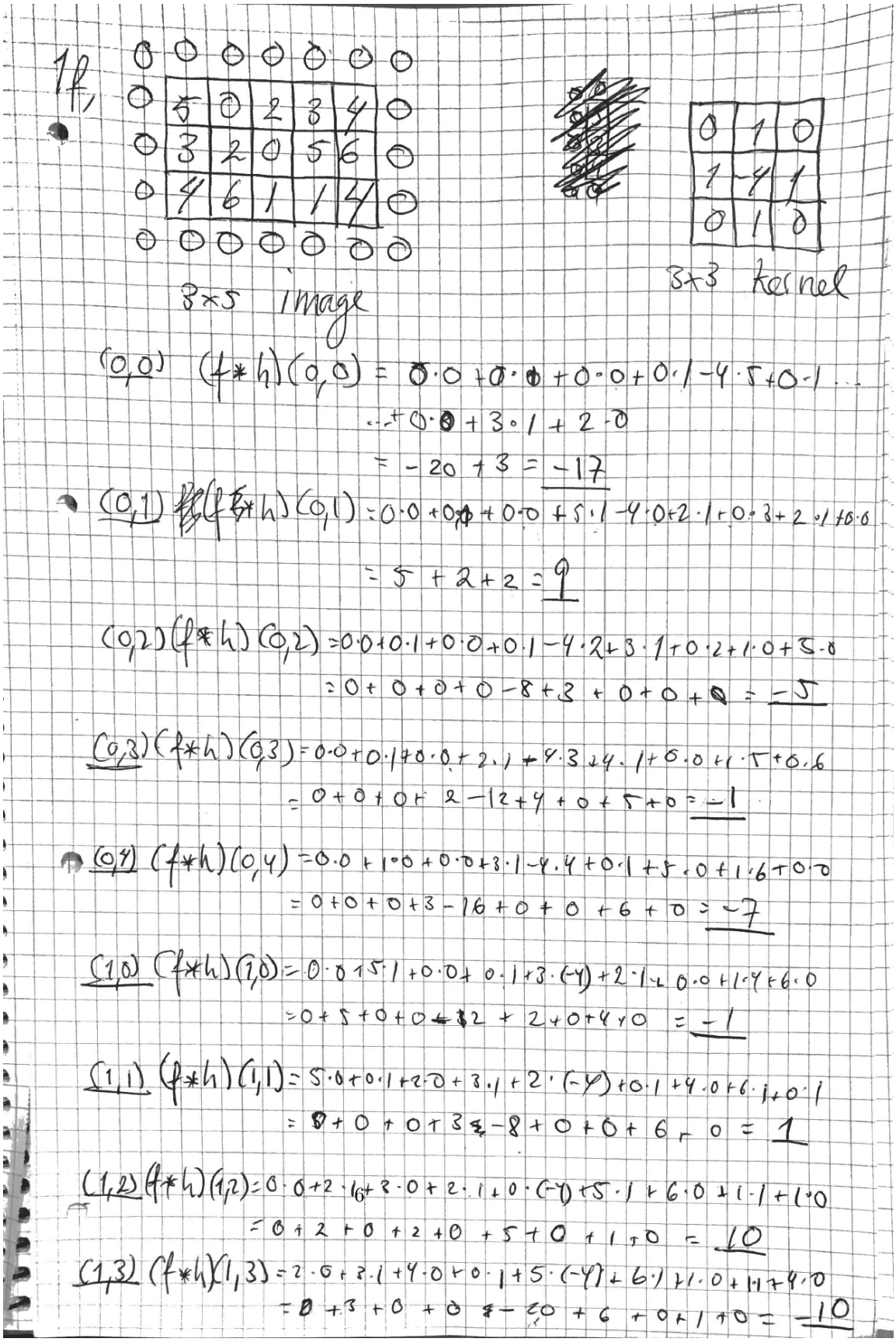
(b) For each digit (0-9), plot the learned weight as a 28 28 image. In your report.

(c) 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.

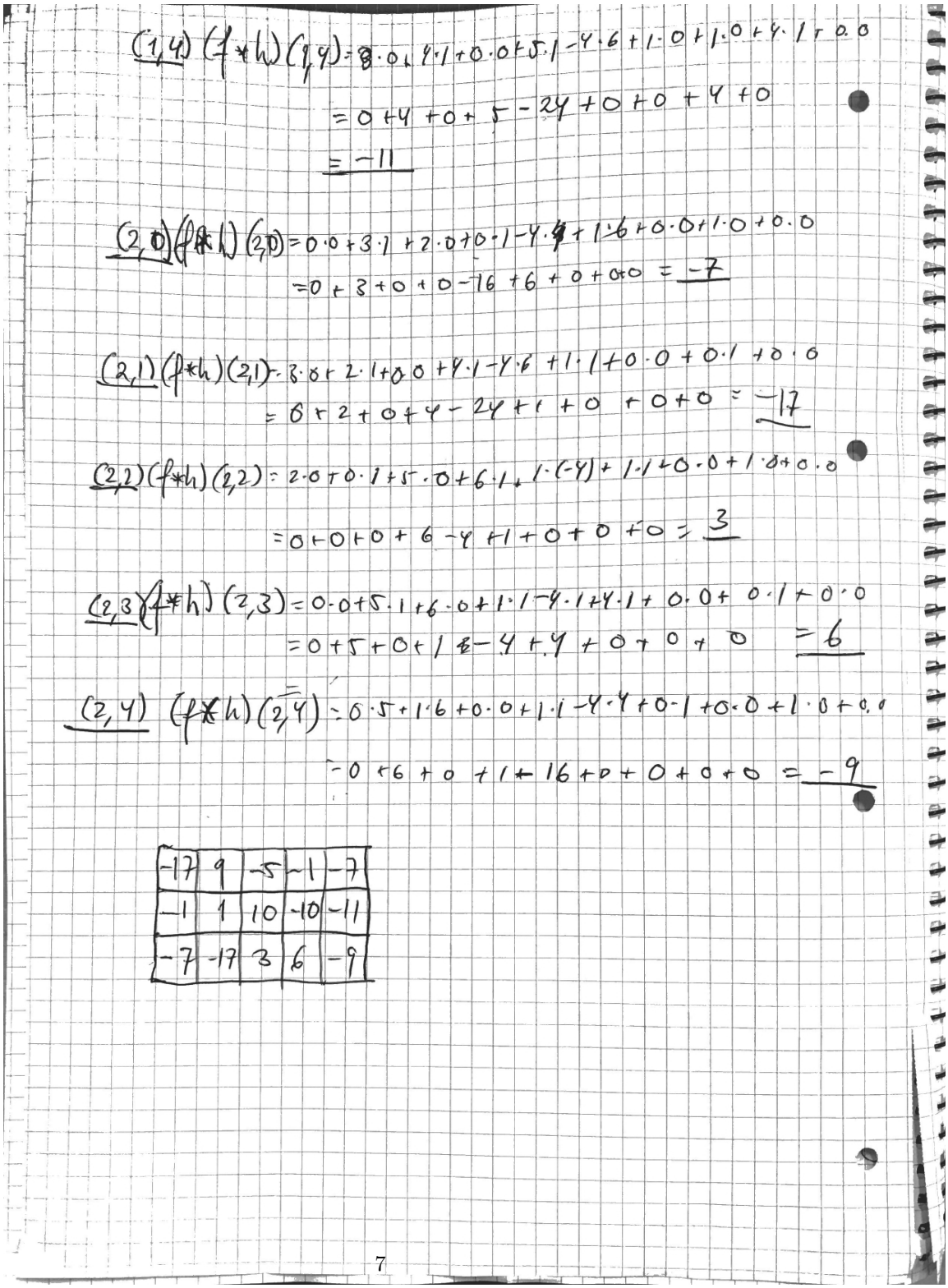
(c) Include an hidden layer with 64 nodes in the network, with ReLU as the activation function for the first layer. What do you observe?



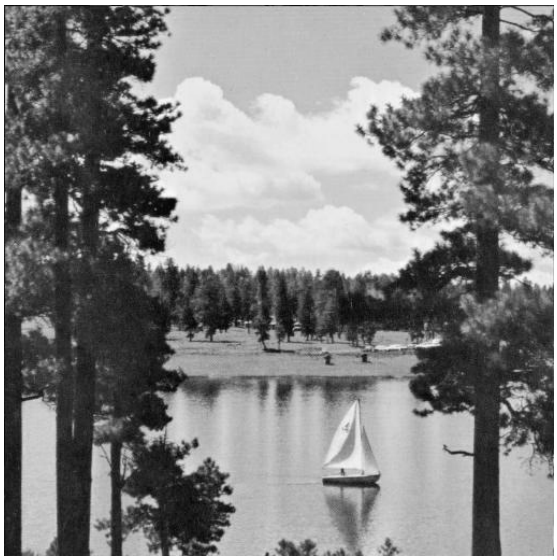
Figur 1: The manual histogram equalization performed in task 1d.



Figur 2: The manual image convolution in task 1f, part 1.



Figur 3: The manual image convolution in task 1f, part 2.

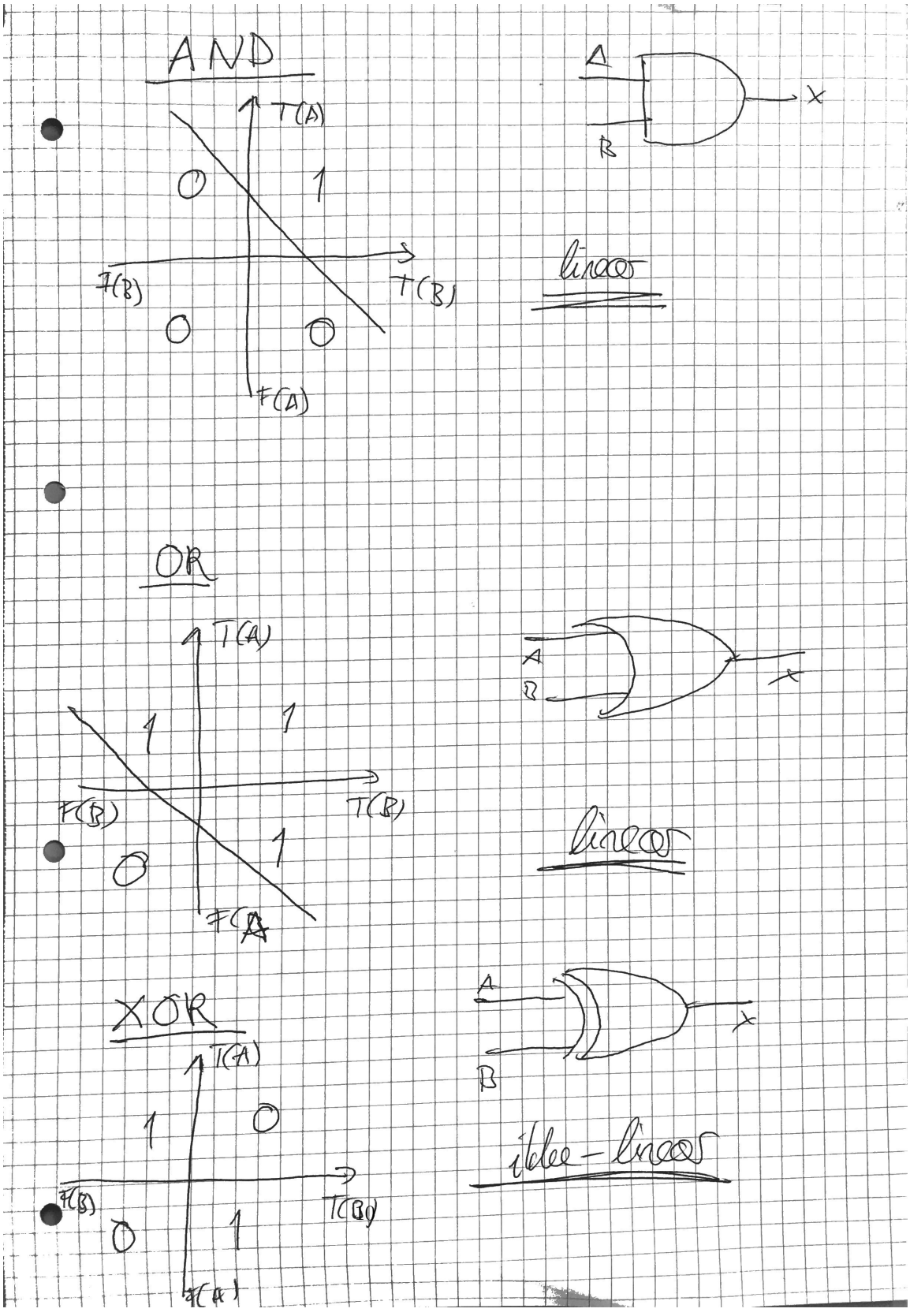


Figur 4: Grayscale image from `task2ab.py`.

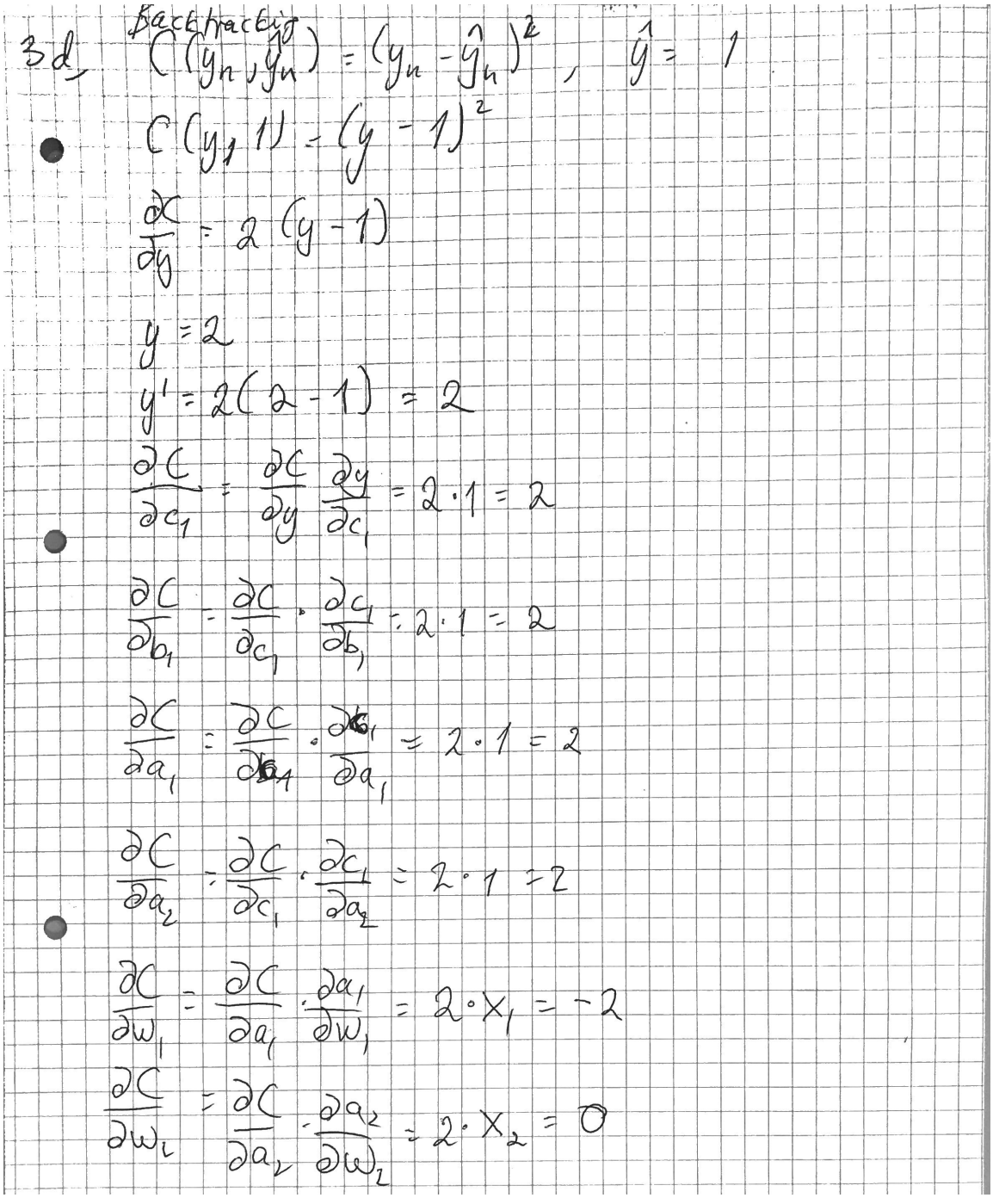
Figur 5: Inverse grayscale image from `task2ab.py`.

Figur 6: First smoothing kernel.

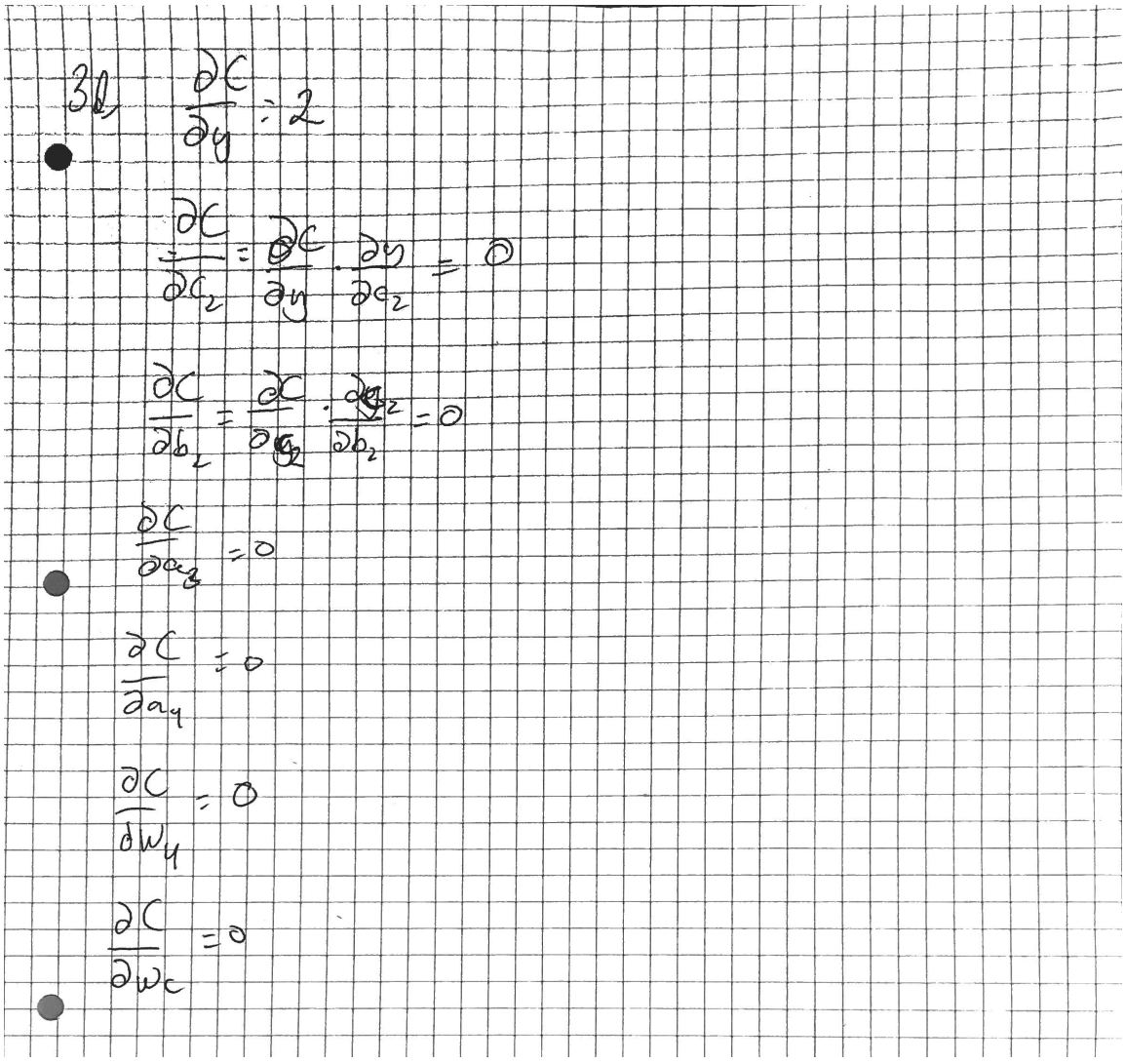
Figur 7: Second smoothing kernel.



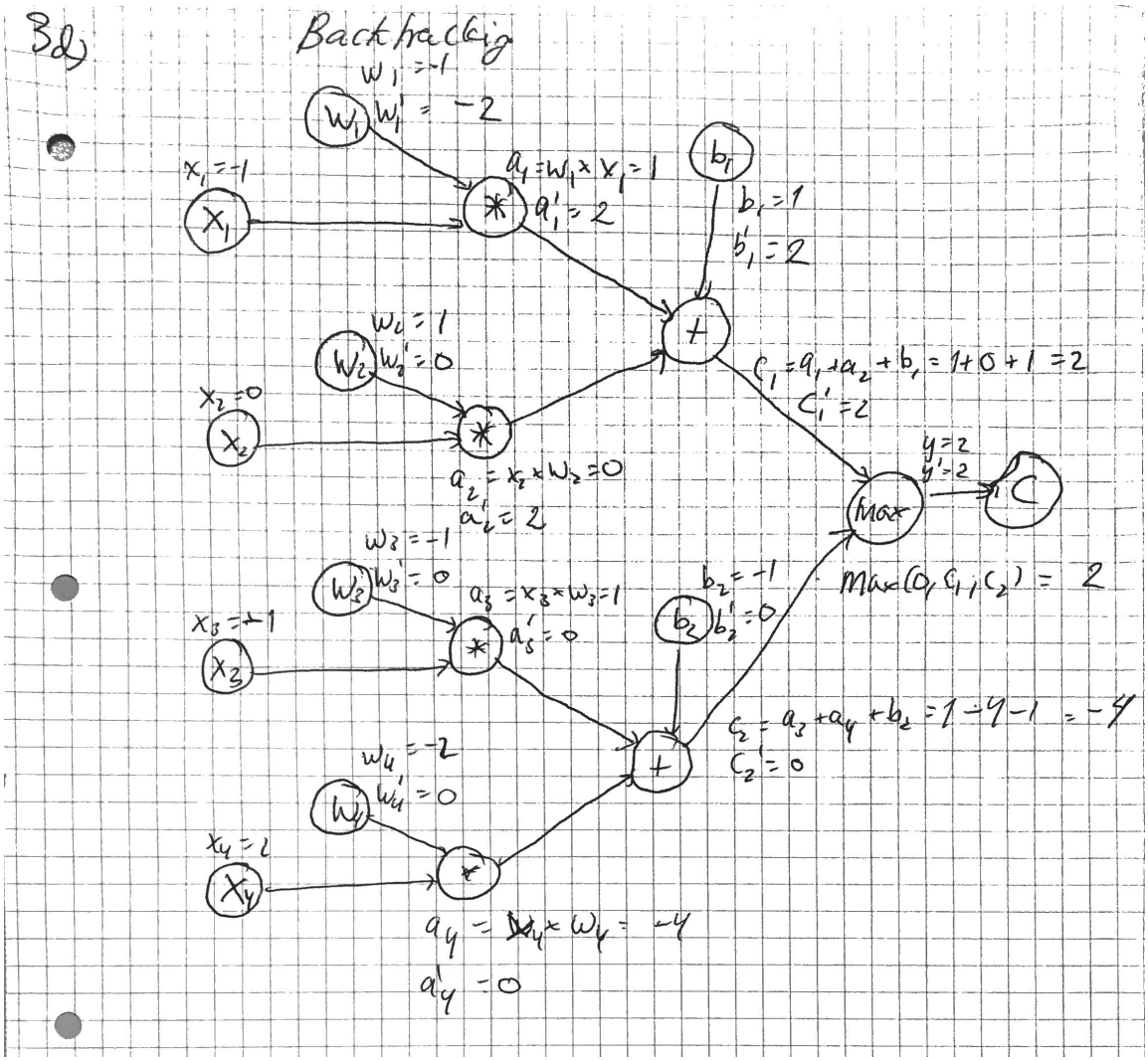
Figur 8: The manual image convolution in task 1f, part 2.



Figur 9: Backtracking on network.

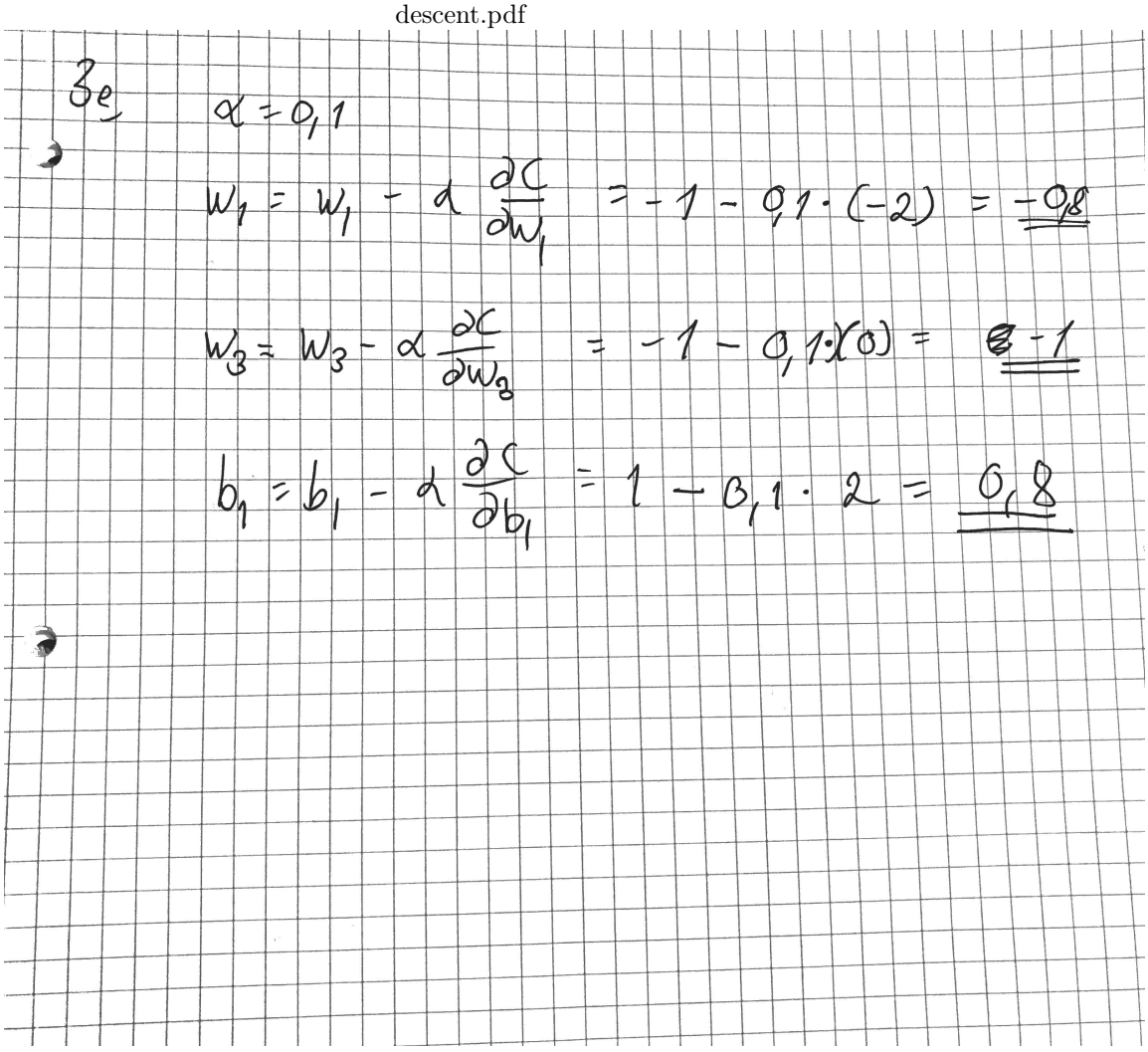


Figur 10: Backtracking on network.

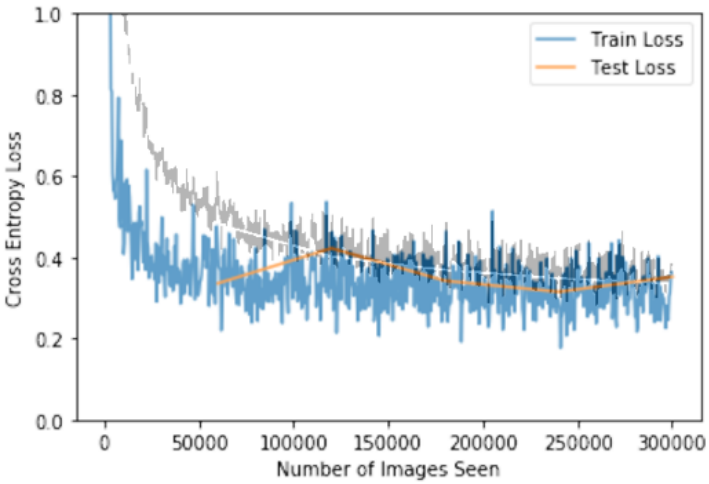


Figur 11: Backtracking on network.

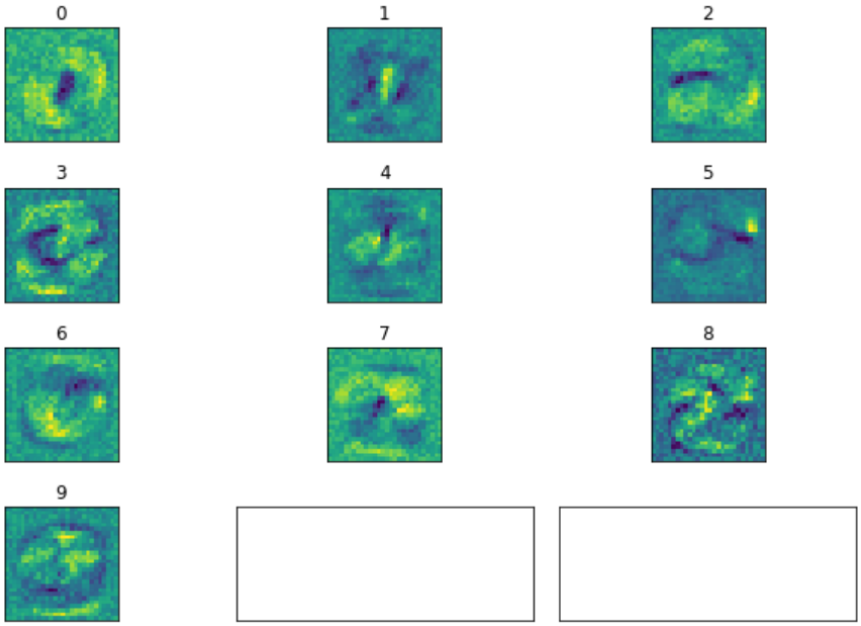
descent.pdf



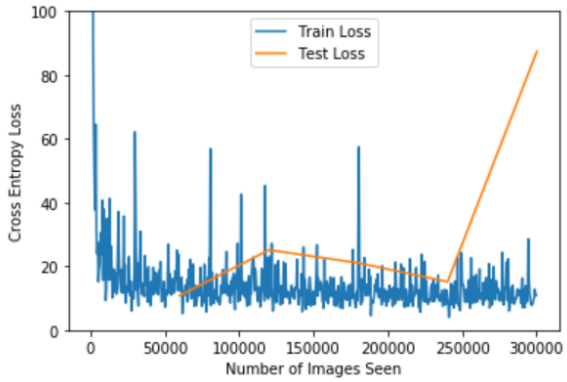
Figur 12: Gradient descent.



Figur 13: Training and validation loss, for normalized (colored) and non-normalized (black and white) training sets. With normalized images the training loss decreases more rapidly in the beginning. It jumps a bit up and down, but that is assumed to be due to randomness in the training.

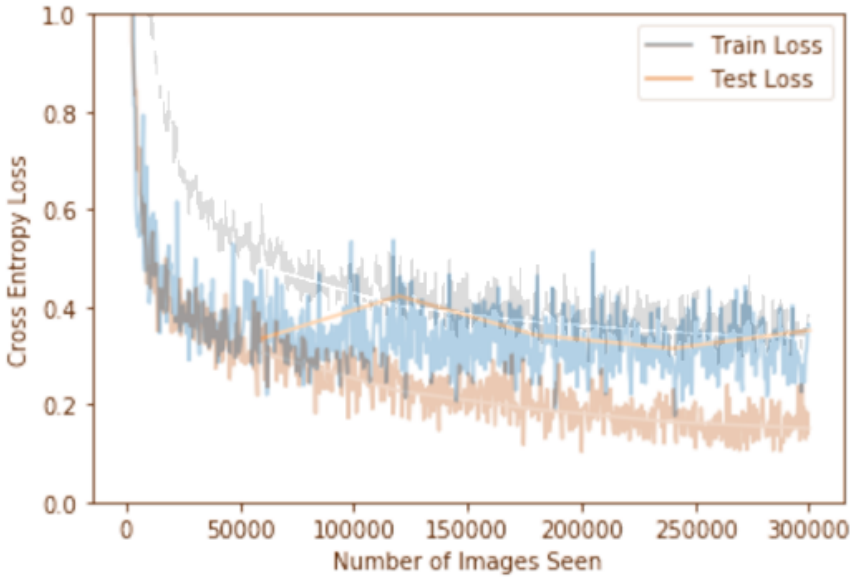


Figur 14: Weights for digits 0 to 9. We see that the weights form images that resemble the numbers they are supposed to recognize. It natural, as the network is linear and the network is similar to a cross-correlation. The weights should capture the most essential features of the a digit that are not seen in other digits.




```
Final Test Cross Entropy Loss: 87.36889603487245. Final Test accuracy: 0.523
```

Figur 15: With lr = 1, the network performs horribly. The reason is that the gradient descent is too fast, so it overshoots and never reaches a stable minimum. In other words, it does not really train well.



Figur 16: The two-layer network with ReLU activation functions have been included as the orange curve. It performs twice as good, with a final Test accuracy: 0.9541, and about half the cross entropy loss.