Chalmers University of Technology

SSY098 - Image Analysis

Lab 2
Learning and convolutional neural networks

Bustrén Cecilia
Gertz Louise
Farhan Witjaksono Fikri

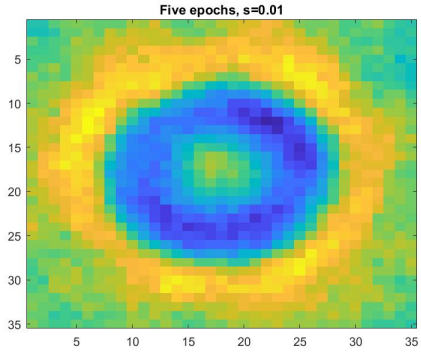
February 10, 2020

Ex. 2.3

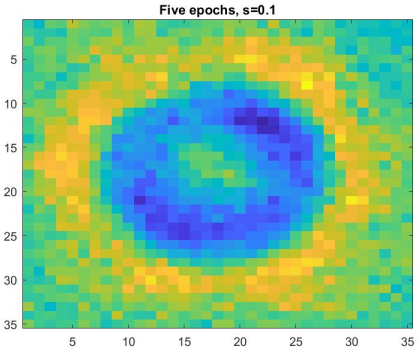
The data was split into training and validation, with 90% of the data in training, and 10% in validation. The labels of the data were taken into consideration when splitting the data, such that half of the data in each of the sets validation and training is of label 1, and half is of label 0.

Ex. 2.6

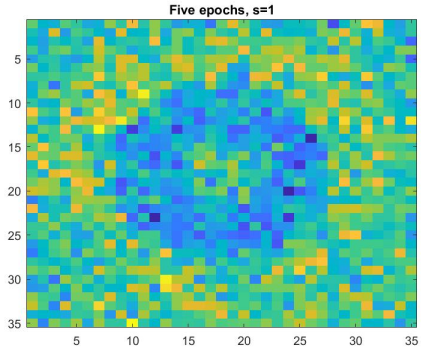
As s is increased, the initial step size is increased. A big step size makes the classifier converge slower, which is seen in figure 1 d) where the s is large. With a larger step size more epochs whould be needed for convergance.



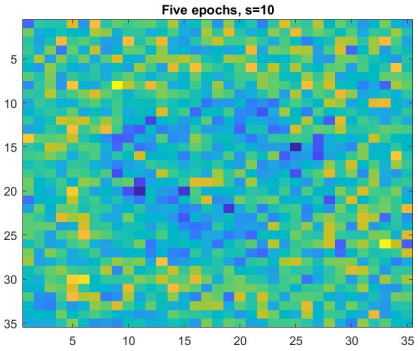
(a) s = 0.01



(b) s = 0.1



(c) s = 1



(d) s = 10

Figure 1: W visualized with different size s.

Ex. 2.7

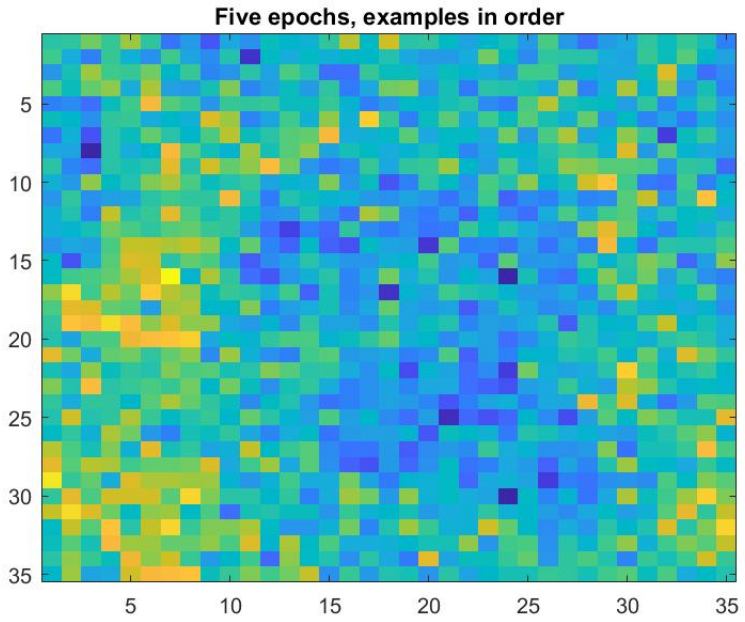


Figure 2: W visualized when the training images are chosen in order.

When the training images are chosen in order, the output of the process is significantly less accurate. This is because the images are in order of all labeled 1 first, followed by those labeled 0. The process runs only five epochs, meaning it only trains for images labeled 1. It needs to train for both labels 1 and 0 to get a good result.

Ex. 2.8

The highest accuracy values we were able to achieve are when using the validation data respective the training data:

$$\text{Accuracy training} = 97.5\%$$
$$\text{Accuracy validation} = 94.2\%$$

Ex. 2.9

The highest accuracy value we were able to achieve was:

$$\text{Accuracy training} = 97.5\%$$
$$\text{Accuracy validation} = 93.5\%$$

Ex. 2.14

Our network contains 29330 trainable parameters

$$20x5x5 + 20 + 12x12x20x10 + 10 = 29330 \tag{1}$$

Ex. 2.16

Thus, the red convolutional filter is roughly 8 times less time consuming than the blue convolutional filter. This is because the red filter has 8 times less number of operations.

Ex. 2.17

If the blue 10 5x5 convolutional filter is replaced by a sequence of two layers of 10 3x3 convolutional filter, then we get less parameters, the process is less time consuming and the output is less accurate.

Ex. 2.18

Table 1: Precision and recall values with *better_cnn_classifier* for training data.

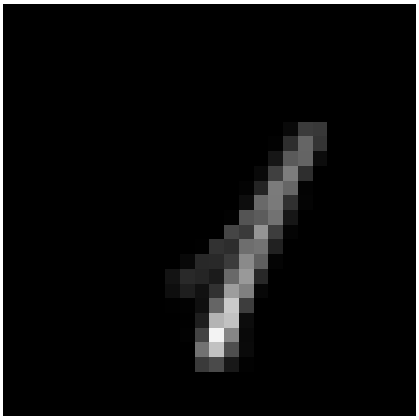
Class	Precision	Recall
0	0.8810	0.9024
1	0.7931	0.9020
2	0.9600	0.8276
3	0.5882	0.7895
4	0.9756	0.7273
5	0.5965	0.7234
6	0.8511	0.7407
7	0.9250	0.8605
8	0.7949	0.5536
9	0.7067	0.9298

Ex. 2.19

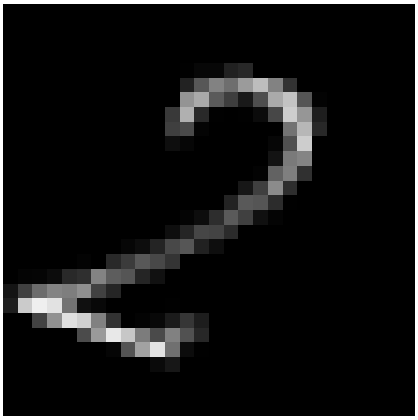
Table 2: Precision and recall values with *better_cnn_classifier* for validation data.

Class	Precision	Recall
0	0.9267	0.8340
1	0.7932	0.8900
2	0.8613	0.8200
3	0.6946	0.8460
4	0.9129	0.7760
5	0.6913	0.7660
6	0.7941	0.7560
7	0.8915	0.8380
8	0.8427	0.6320
9	0.6987	0.8440

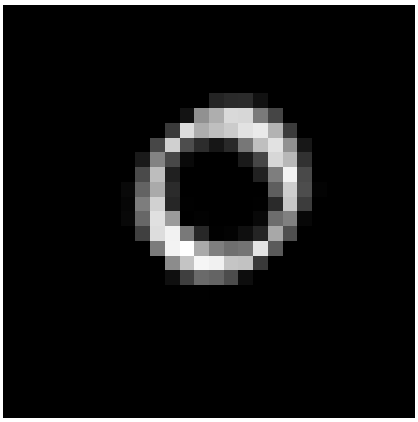
Ex. 2.20



(a) Class 1 mistaken as class 0.



(b) Class 2 mistaken as class 7



(c) Class 0 mistaken as class 9.

Figure 3: Three images of mistaken numbers.