Assignment 1 - Supervised learning: kNN and Backpropagation

Overview of the data

Looking at the data from a machine learning perspective we can observe how the data is compressed from 32x32 bitmap images to 64 features with the integer range 0..16. This means that we can create a hypercube feature space with 64 dimensions. This is a significantly dimensionally reduced feature space compared to 32x32 binary dimensions.

The pre processing of the data doesn't only reduce the number of dimensions, it also reduces the variance in small distortions.

Implementation of the kNN algorithm

kNN without cross-validation

The implementation of kNN is fairly straight forward.

We iterate though all the cases in the test set and calculate the distance to each point in the training set. The result is then sorted in ascending order and the `k` first elements are counted in regards to what label they are assigned. The label with the most values are then picked to be assigned for the testing case we are currently processing. When no counted label is higher than another we pick the label with the data point which have the closest euclidean distance to the test case.

The result of our implementation of the kNN algorithm with k arbitrarily choose as 4 can be seen below.

Parameters: k = 4

```
cM =

    99     0     0     0     0     0     0     0     0     0
     0    97     1     0     0     0     0     0     2     3
     0     0    97     0     0     0     0     0     0     1
     0     1     0    98     0     0     0     0     0     3
     0     1     0     0    96     0     0     0     0     2
     0     0     0     1     0   100     0     0     0     1
     1     0     0     0     0     0   100     0     0     0
     0     0     2     0     0     0     0   100     0     0
     0     1     0     0     1     0     0     0    96     2
     0     0     0     1     3     0     0     0     2    88
```

```
acc =

    0.9710
```

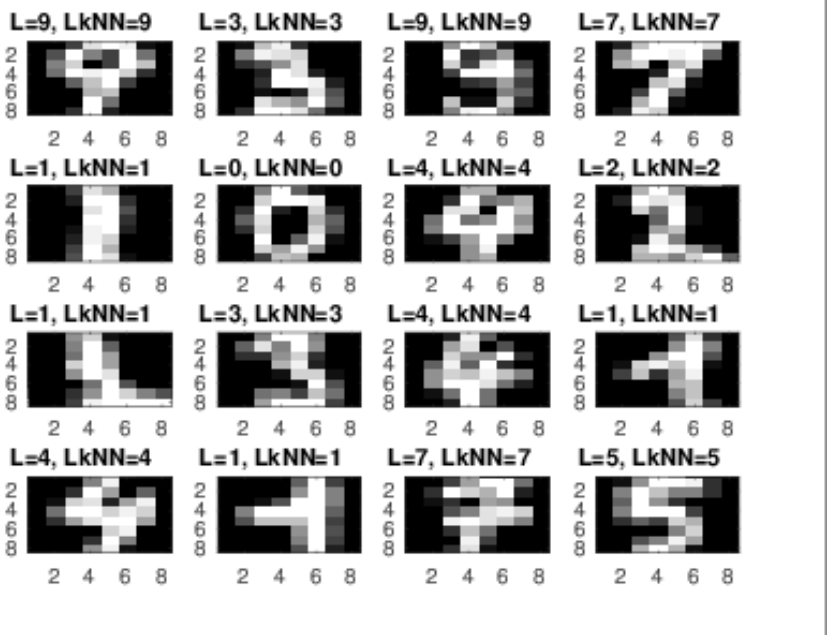

Figure 1: Result from kNN

kNN with n-fold cross validation

For the cross validation version the n-fold cross validation algorithm was used to determine the best value for k. For all of the following results a n of 2 was used but he algorithm implementation allow for any value of n as long as it can evenly distribute the data set. Accuracy is used as the validation score in order to find the best value for k.

Data set 1

Best parameters: k = 1 Accuracy = 0.9900

Data set 2

Best parameters: k = 1 Accuracy = 1

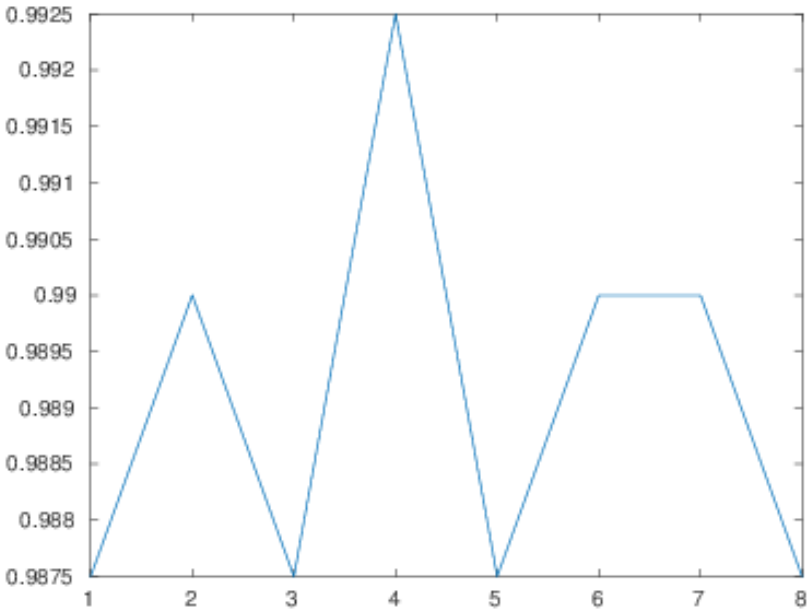

Figure 2: CV scores

Figure 3: OCR result

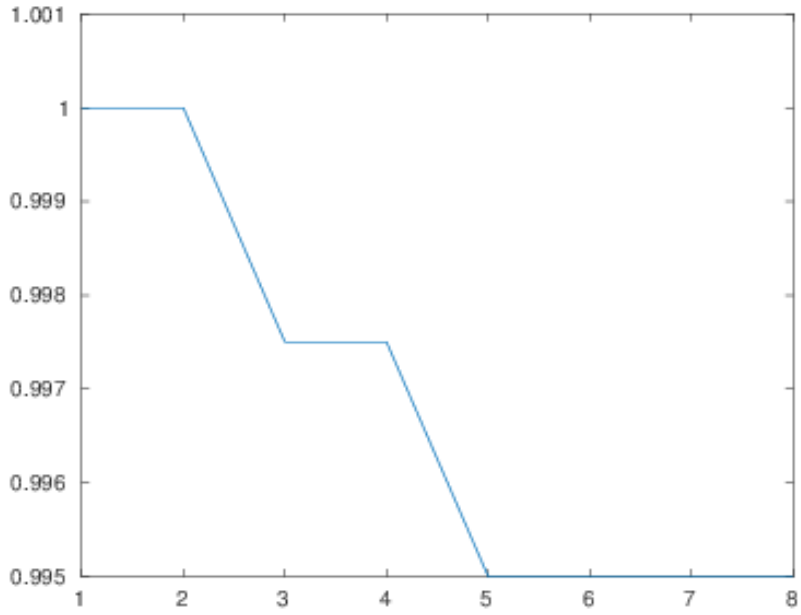

Figure 4: CV scores

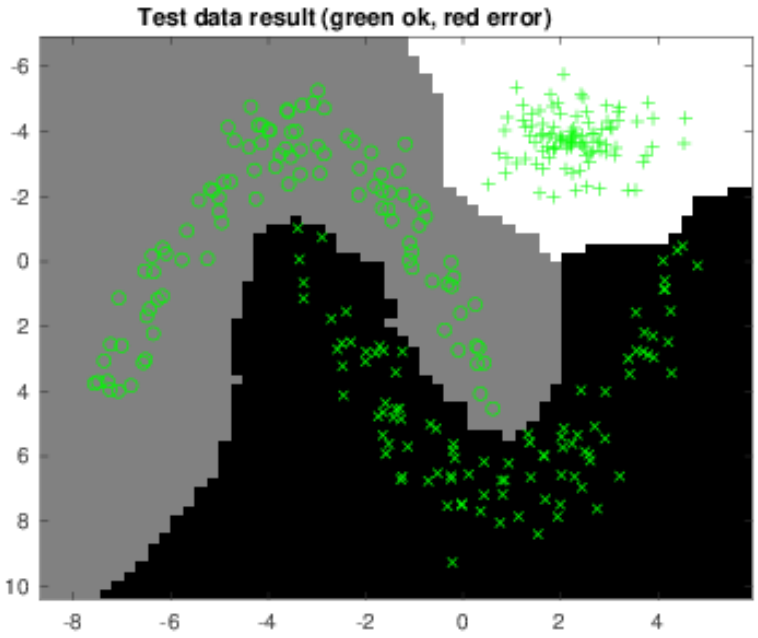

Figure 5: OCR result

Data set 3

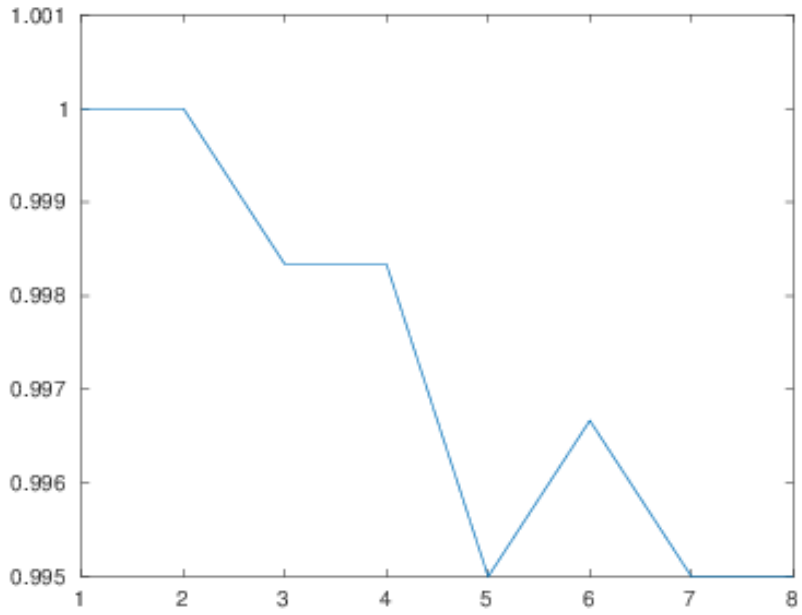

Figure 6: CV scores

Best parameters: k = 1 Accuracy = 1

Data set 4

Best parameters: k = 1 Accuracy = 0.9840

- Explain how you selected the best k for each data set using CV and the result, include the accuracy and images of your results for each data set
- A short summary of your backprop network implementation (single + multi)
- Present the result from the backprop training and how you reached the accuracy criteria for each dataset. Motivate your choice of network for each dataset, then explain how you selected good values for the learning rate
- Present the result, including images, of your example for a non-generalisable backprop solution. Explain why this example is non-generalisable
- A final discussion and conclusion where you explain the difference between the performance of the different classifiers, Pros and cons etc.

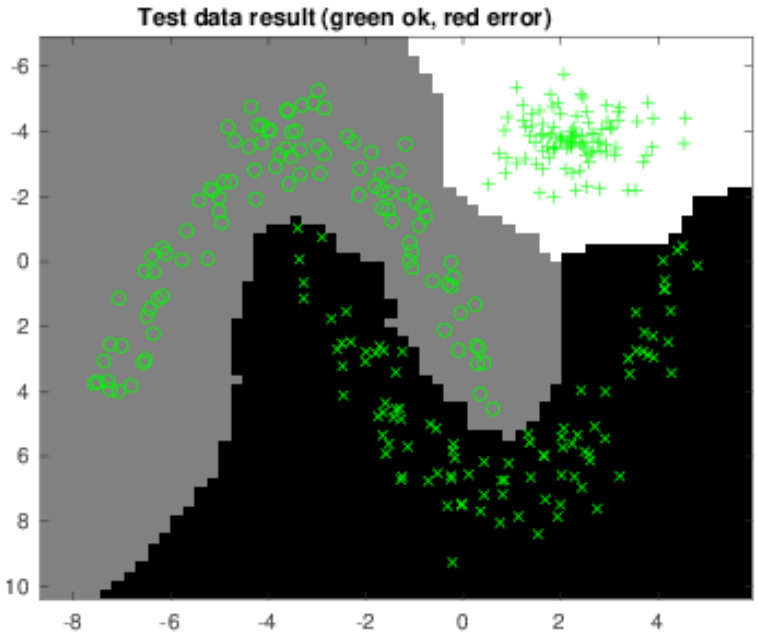

Figure 7: OCR result

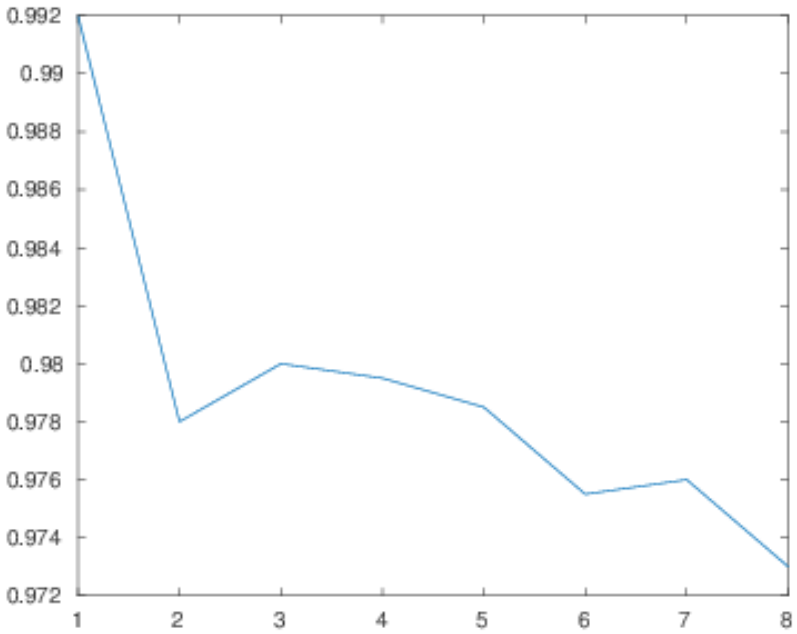

Figure 8: CV scores

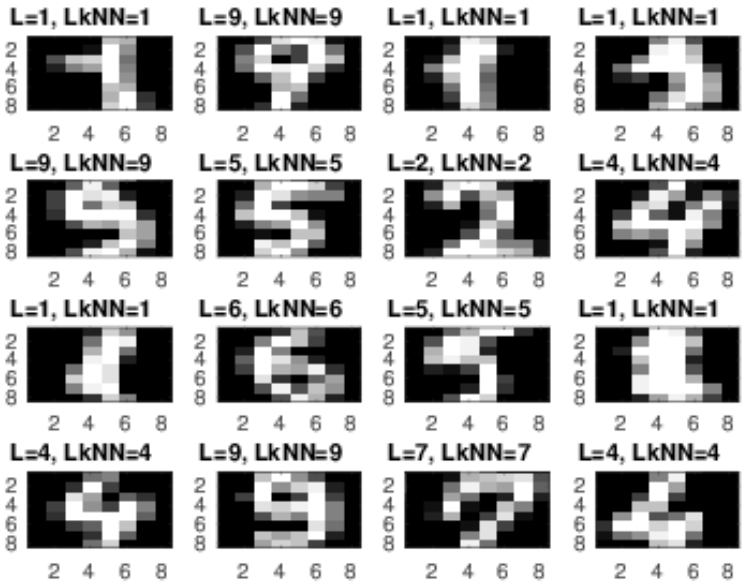

Figure 9: OCR result