TBMI26 – Computer Assignment Report  
Supervised Learning

Deadline – February 12 2018

Author/-s: Eric Herwin and Erika Anderskär

In order to pass the assignment you will need to answer the following questions and upload the document to LISAM. If you meet the deadline we correct the report within one week after the deadline. Otherwise we give no guarantees when we have time.

1. **Give an overview of the data from a machine learning perspective. Consider if you need linear or non-linear classifiers etc.**

|  |  |
| --- | --- |
|  |  |
| 1. Data set 1 | 1. Data set 2 |
|  |  |
| 1. Data set 1 | 1. OCR data set |

The first data set could be classified with a linear classifier since the green and the red classes could be separated with a line. The second data set could not since it’s impossible to draw a straight line that separates the classes. Same thing with the third data set. The forth data set is different, since it consists of 64 features the 2-dimensional plot doesn’t really show the complexity of the data, but it could probably not be classified with a linear classifier since there are so many features.

1. **Explain why the down sampling of the OCR data (done as pre-processing) result in a more robust feature representation. See** [**http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits**](http://archive.ics.uci.edu/ml/datasets/Optical+Recognition+of+Handwritten+Digits)

The pictures are divided into “subpictures” and the pixels in those areas are counted. That means that the feature dimension have been reduced to 8\*8=64 features. This also means that an average from 4\*4 pixels have been taken which means that noise or wrongly labeled pictures have been canceled out. This leads to a more robust data set.

1. **Give a short summery of how you implemented the kNN algorithm.**

The distances are calculated for every data point in the training and test set. Then the distances are sorted, and the k smallest ones are used to predict the class label.

1. **Explain how you handle draws in kNN, e.g. with two classes (k = 2)?**

In cases where 2 neighbors are used, and they give a different class label, the shortest distance is used.

1. **Explain how you selected the best k for each dataset using cross validation. Include the accuracy and images of your results for each dataset.**

K values between 1 and 10 are evaluated for three different subsets of the data. The average of the accuracy between the three subsets are calculated and used as an overall-sample-accuracy. Then the maximum accuracy is used to decide which mean is the best.

|  |  |
| --- | --- |
|  |  |
| 1. kNN result for data set 1   Accuracy = 0.95 | 1. kNN result for data set 2   Accuracy = 0.93 |
|  |  |
| 1. kNN result for data set 3   Accuracy = 0.8267 | 1. kNN result for OCR data   Accuarcy = 0.11 |

Looking at the picture a, the result looks quite good. It classifies 95 percent correct. Although, you could probably get an even better classification with a better decision boundary.

In picture b and picture c it is shown that kNN cannot find the concave shapes, but the accuracies are still quite high.

The hand-written digits can’t really be classified using kNN. The accuracy is really low and, in the picture, you can see that the LkNN does not give correct guesses.

1. **Give a short summery of your backprop network implementations (single + multi). You do not need to derive the update rules.**

Single layer:

The items necessary for performing the network was a weight matrix that has the size of (nr. Of input variables +1 \* nr. of output variables). The +1 for this is the weights of the bias, and to iclude them in the calculation then we need to add a row of ones in to the data matrix. Then the WTX can be calculated and a tanh function was applied to it and resulting in a result Z. For the backpropagation, the gradient was computed by the MSE error function that was derived by the weights.

Multi layer:

The same initialization process as the single layer, but with a new layer with neurons to keep in mind. For this, two weight matrices needs to be initialized, W and V. For this network we choose the matrix W to be the weights for the hidden layer and the matrix V to be for the output layer. The dimensions of the matrices are decided the same way as the single layer, but the number of columns for W is counted as the number of neurons in the hidden layer and the number of columns for V is counted as the number of output variables. On the output layer there is a tanh activation function applied. This forward propagation then produces a result Z. For the backpropagation, the gradient was computed by the MSE error function that was derived by the weights of each layer, i.e. the gradient of matrix V is obtained by deriving the error function of the output Z by V. The gradient of matrix W is also obtained by deriving for the error function of the output Z by W. For calculation of the gradient of W we did not include the bias of V.

1. **Present the results from the backprop training and how you reached the accuracy criteria for each dataset. Motivate your choice of network for each dataset. Explain how you selected good values for the learning rate, iterations and number of hidden neurons. Include images of your best result for each dataset, including parameters etc.**

In the multilayer neural network task, the weights were generated -0.1 and 0.1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data set nr | Number of hidden units | Number of iterations | Learning rate | Accuracy |
| 1 | 3 | 1000 | 0.001 | 99.2% |
| 2 | 3 | 4000 | 0.001 | 99% |
| 3 | 5 | 8000 | 0.05 | 99.7% |
| 4 (OCR data) | 66 | 3000 | 0.005 | 97% |

The values were selected via trail and error until the accuracy criteria for each data set was met.

|  |  |
| --- | --- |
|  |  |
| 1. Multilayer result for data set 1 | 1. Multilayer result for data set 2 |
|  |  |
| 1. Multilayer result for data set 3 | 1. Multilayer result for OCR data |

Looking at the plots a-c, almost all the observations were classified correctly. The background color shows the decision boundaries which all well-estimated. The accuracy is over 99% for these three data sets which is a good result.

The picture for the OCR data shows that the estimate class are equal to the true class for all the 16 classes. The accuracy is 97 percent which is good. Here more units in the hidden layer were needed. This is because the OCR data has more possible outputs, and more input variables, compared to the other data sets.

1. **Present the results, including images, of your example of a non-generalizable backprop solution. Explain why this example is non-generalizable.**

|  |  |  |  |
| --- | --- | --- | --- |
| Numbers of hidden units | Number of iterations | Learning rate | Accuracy |
| 5 | 8000 | 0.05 | 1 |

|  |  |
| --- | --- |
| https://scontent-arn2-1.xx.fbcdn.net/v/t34.0-12/27720694_10208452224221444_1887765991_n.png?oh=612919668d4ba456a991b279b44fca5c&oe=5A7BB2E8 |  |
| 1. Training and test error | 1. Classification result |

1. **Give a final discussion and conclusion where you explain the differences between the performances of the different classifiers. Pros and cons etc.**

The pros of kNN is that it is very simple to implement, but it gave rather poorly result, especially for the OCR data. It could possibly have given a better result with a different distance metric than Euclidean.

The single neural networks can only do linear classifications

1. **Do you think there is something that can improve the results? Pre-processing, algorithm-wise etc.**

Maybe an additional layer in the multilayer neural net could find the more complex boundaries.