Pattern Recognition
Exercise 2

Mario Kaufmann
Ramona Imhof
Jakob Schärer
Adrian Wälchli
Jan Luca Liechti

April 10, 2017

1 SVM

Using the *libsvm*[1] SVM library, we used the following parameters to obtain the best possible results:

- $k = 10$ (for k-fold cross-validation)
- $C = 2^{-5}, \dots, 2^{15}$ (10 different values in total)
- $\gamma = 2^{-15}, \dots, 2^{3}$ (10 different values in total)

With these values, we are following the *libsvm* guide to support vector machines[2]. We also normalize all values to the interval $[0, 1]$. In order to obtain results faster, we only ran 10% of the training and test set.

The grid search with above listed parameters produced the following results:

[1] https://www.csie.ntu.edu.tw/~cjlin/libsvm/

[2] https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf, p. 5-8.

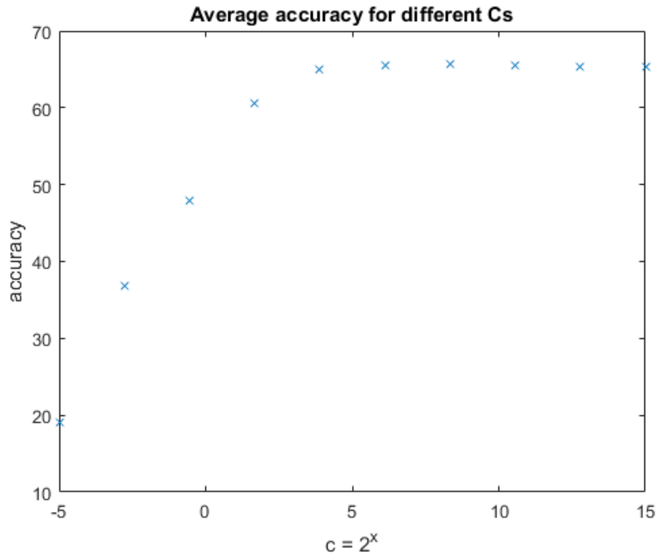

Figure 1: RBF kernel: C vs. accuracy

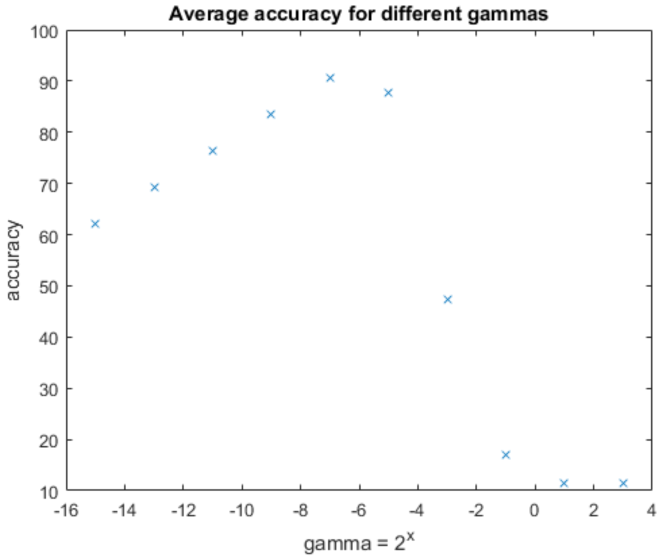

Figure 2: RBF kernel: gamma vs. accuracy

These are the results for the RBF kernel. The best combination of C and γ turned out to be $C = 2^{1.\overline{6}}$ and $\gamma = 2^{-5}$. With this combination, we achieve an overall accuracy of **95.136576** percent.

As for the linear kernel, the different Cs all gave quite similar accuracies: Here, we had an overall accuracy of **90.037** percent with $C = 2^{-5}$.

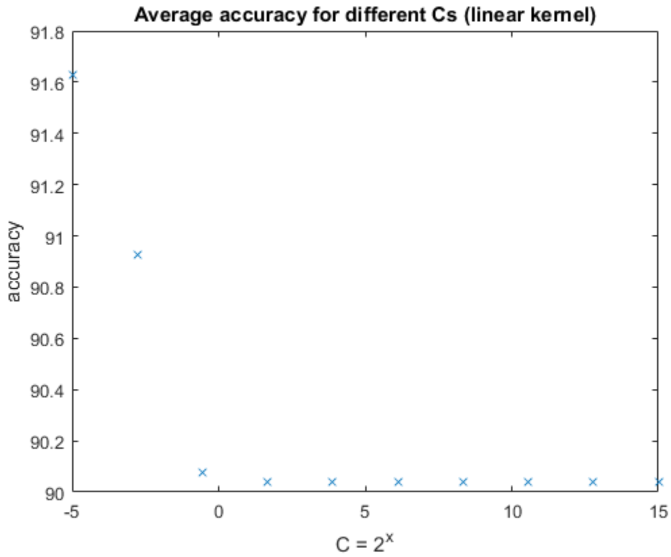

Figure 3: Linear kernel: C vs. accuracy

2 MLP

2.1 Implementation with NN Toolbox

The Neural Network toolbox from MATLAB was used to implement the MLP. We used cross-validation to find the optimal number of neurons in the hidden layer as well as the optimal learning rate. We adapted a grid search on the following search space:

- Layer size: 50, 60, 70, ..., 150
- Eight learning rates on the interval $[0.001, 0.05]$

We found the optimal parameters are 140 neurons and 0.001 for the learning rate. On the full training set, the runtime for the parameter search is about 25 minutes. We save the performance for the best network and show it in figure 4. The classification accuracy on the test set with these parameters is **96.2736** percent.

2.2 Implementation with standard MATLAB tools

Additionally to the solution with the neural network toolbox we implemented our own approach for training a MLP. It can be found in the folder *mlp-regularization*. The implementation uses regularization controlled by the parameter $\lambda \in [0, 0.3]$. The other parameters were set as follows after optimization:

- learning rate $c = 1$
- number of hidden nodes: 100

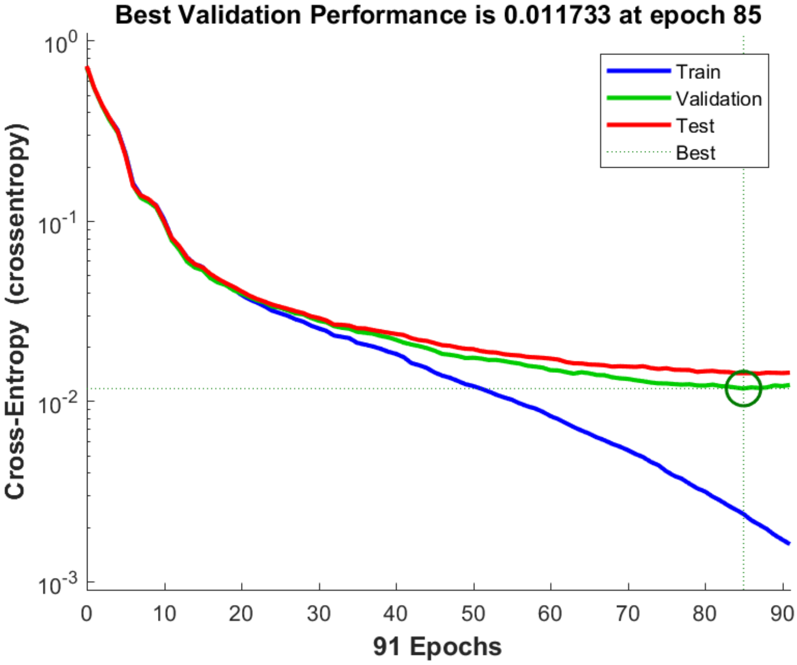

Figure 4: Cross-entropy error vs. epochs. The test set in this figure is not the same as the test set we use for the final evaluation. These are the three splits to verify the performance during training. We abort after 6 validation checks (when the performance does not improve anymore).

- number of epochs: 100

In addition, the feature descriptor *HOG*[3] was used to transform the samples into another feature space.
The parameter λ heavily influences the classification accuracy. As can be seen in figure 5, the optimal value for λ lies around 0.1. Furthermore, it is visible that as expected, the cost is lower on the training set than on the test and validation set, just as the accuracy is lower on the test and validation set than on the training set. Under this configuration the classification accuracy was found to be **96.5936** percent on the test set.

[3] http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf

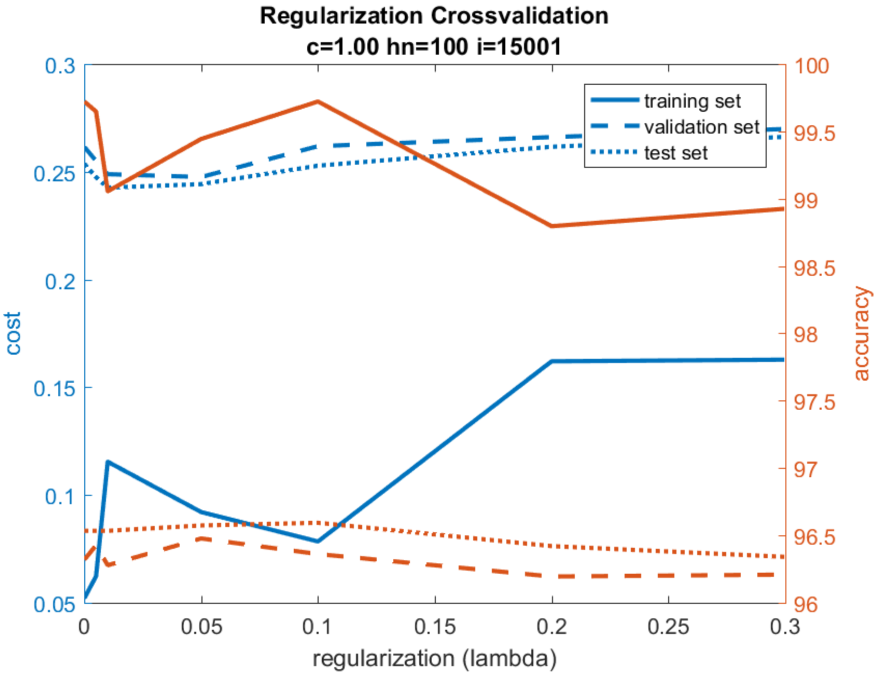

Figure 5: Accuracy and cost on the training, validation and test set depending on the parameter λ. The optimal value for λ lies around 0.1.