Neural Networks and Learning Systems

Assignment 3 - Boosting

Dinuke Chrishan Jayaweera, Patrick Siegfried Hiemsch

April 4, 2022

1. Introduction

In this laboratory we explored and implemented the AdaBoost algorithm to detect faces. AdaBoost is a boosting algorithm, that is based on the training of multiple weak classifiers that together form a much stronger classifier with higher predictive power.

The dataset we used contains grey-scale pictures of faces as well as pictures of non-faces (random pictures from the internet). In total there are around 4900 face images and around 7800 non-face images. All images are represented by a 24x24 matrix. The following image shows some examples of both categories:

Figure 1: Examples of face pictures

Figure 2: Examples of non-face pictures

2. AdaBoost - Implementation and Application

2.1 Weak Classifiers - Haar-Features

The features used to train the weak classifiers for AdaBoost are Haar-Feature masks that are applied to the images. They mainly detect simple shapes like lines or edges by looking for differences in contrast (e.g. dark eyebrow and light forehead). Some examples for possible, random Haar-Filters are shown below:

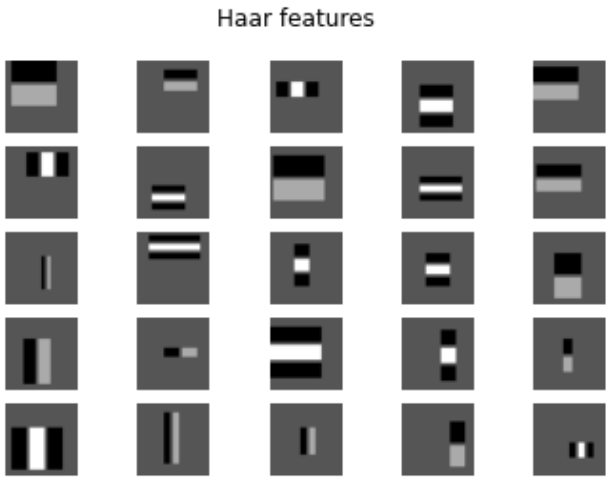

Figure 3: Three example numbers from OCR-data

Every WeakClassifier chooses the best possible threshold and polarity to obtain the smallest weighted error.

In every AdaBoost step, out of all the possible WeakClassifiers, the one is chosen that minimizes the weighted error sum.

2.2 Choosing optimal n_Haar

In the following plot we compare the test and training accuracies for different numbers of WeakClassifiers used for prediction.

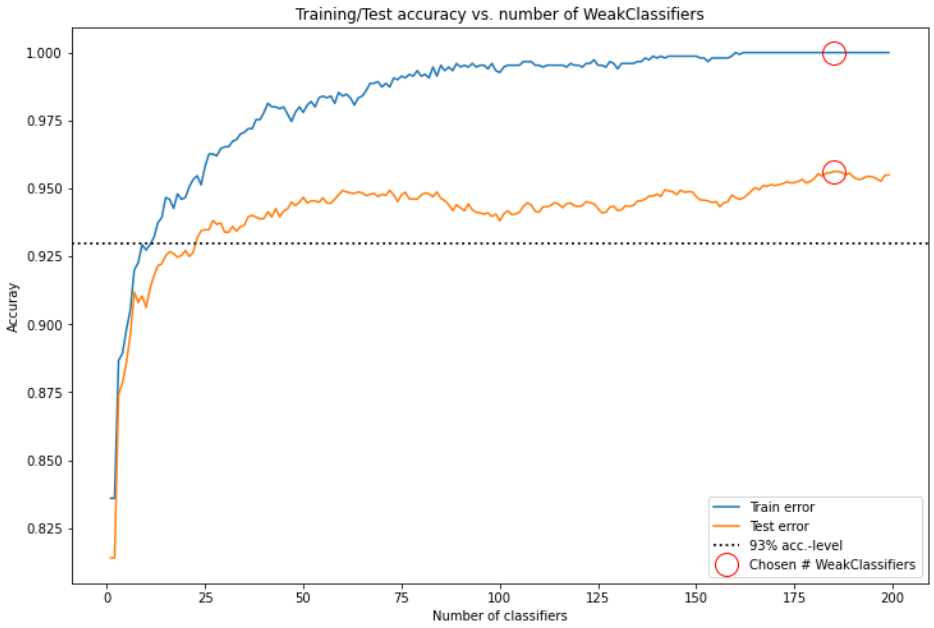

Figure 4: Training and Test accuracy depending on number of WeakClassifiers

The data was obtained by training a strong AdaBoost classifier with 200 random Haar-features on 1500 images, 50% faces and 50% non-faces, and testing on 11288 images, 37% faces and 63% non-faces.

In general we can see that the overall accuracy increases with an increasing number of the trained WeakClassifiers (which of course makes sense) and we already achieve the threshold of 93% test accuracy around 25 WeakClassifiers.

We decided to use the trained StrongClassifier with 185 single WeakClassifiers (red circle in the plot) which shows the highest accuracy score on the test data. But we could also use a simpler model with less WeakClassifiers for example around n_class = 60, which would result in only a small change in terms of accuracy (trade off between model complexity and performance). Using the 185 decision stumps, the model achieves a Training accuracy of 100% (so no misclassifications on the Training set) and a Test accuracy around 95%.

Comparison to random feature selection

In the following we compare the results obtained above with a slight modification of AdaBoost: Instead of choosing the WeakClassifier leading to the lowest weighted error for the respective iteration, we choose one of the Haar-features at random, which is decreasing the computational complexity

immensely. In the standard case, we train $W \cdot F \cdot N$ WeakClassifiers (W = Number of WeakClassifiers, F = Number of possible features, N = Number of data points) since there are F choices for every iteration W where N thresholds for the WeakClassifier need to be compared. Choosing a feature at random leads to the training of $W \cdot N$ WeakClassifiers ($O(W \cdot F \cdot N) > O(W \cdot N)$)

-The following plot shows the Training and Test accuracies depending on the number of chosen WeakClassifiers for prediciton.

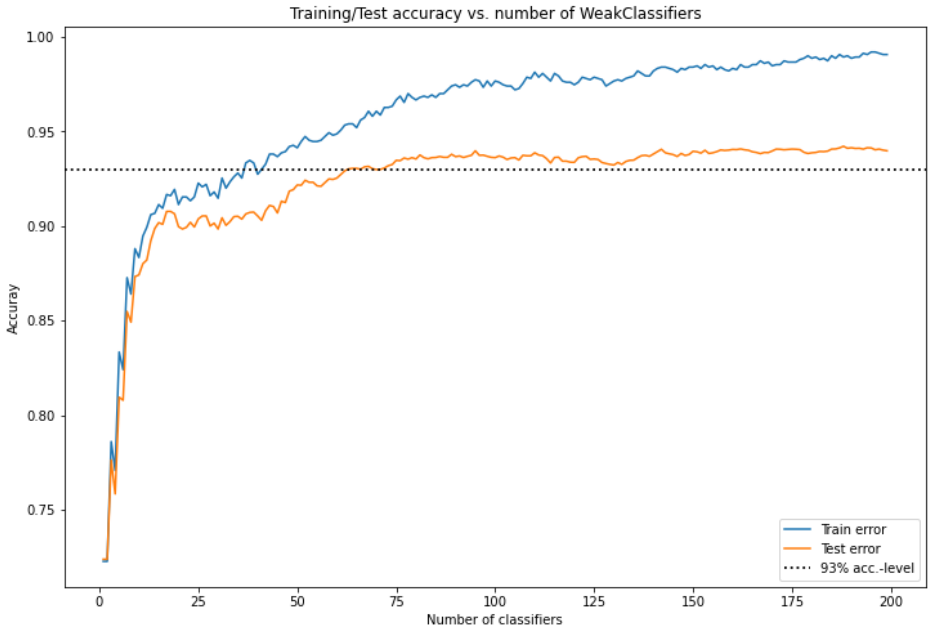

Figure 5: Training and Test accuracy depending on number of WeakClassifiers for random feature selection

We can see that Training is much faster (4 seconds vs. 772 seconds) but in general we need a lot more WeakClassifiers to reach the 93% accuracy threshold (around 60 instead of around 25) and the maximal reached accuracy is 94.53% in comparison to 95.62% for the optimal case. But still, with an increasing number of WeakClassifiers used for prediction, Training and Test accuracy increase.

2.3 Predictions

The following image shows some of the predictions produced by the StrongClassifier. RL in this context means real label (1 = face, -1 = non-face) and PL means predicted label. The first row shows a couple of right predictions, the second row shows faces that were not detected and the last row visualizes three objects that were falsly classified as a face.

We see can see that two of the faces that were not detected show men with glasses and also a man screaming. Those are images that are harder to predict especially for the Haar-feature classifiers since they rely on common shapes or patterns like two points for the eyes instead of dark sunglasses. The false positives also show some patterns that could also be found in human faces. Especially the first image with the round shape could resemble the round shape of a head.

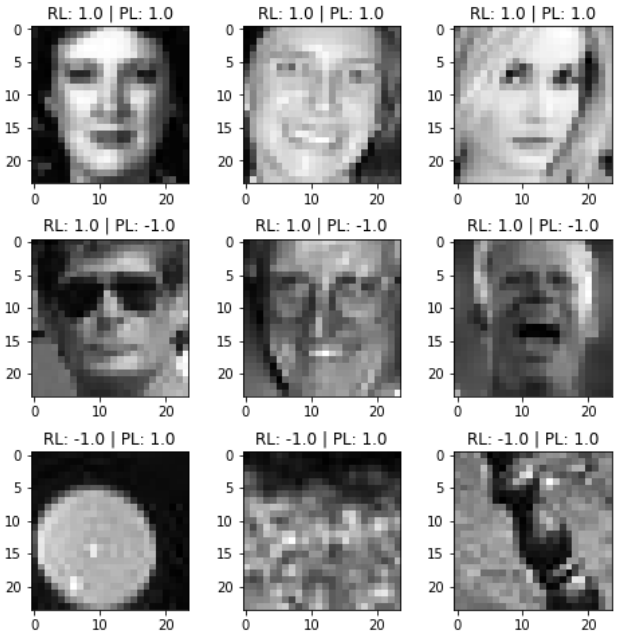

Figure 6: Sample-Predictions produced by StrongClassifier

2.4 Chosen Haar-features

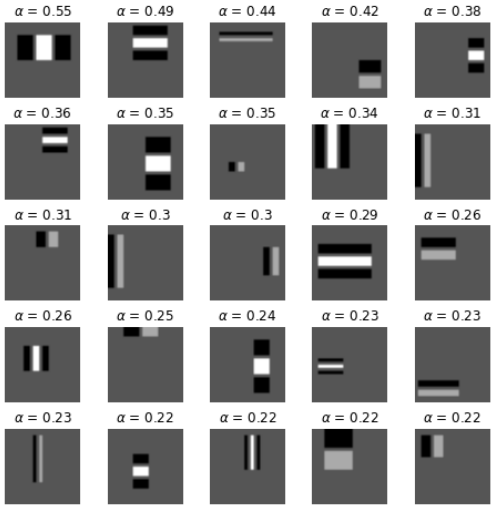

Figure 7: Most important Haar-features

The most important randomly generated Haar-features that were chosen during one run of the algorithm are shown in the plot above together with their calculated alpha value.

Also, the next plot shows the 5 most important features plotted on random faces. You can clearly see that certain areas from the faces are covered, for example the eyebrow line is covered by filter three or the difference in brightness on the face in the eye and nose/forehead region is detected by filter one.

Figure 8: Most important Haar-filters applied to faces

3. Discussion and Conclusion

The results obtained by the classifier are very good especially against the background that only WeakClassifiers were used. The predictions from the StrongClassifier are reasonable. But there could still be some potential for improvement: The filters all have a fixed position, but it could be beneficial to use the most important ones on more positions in the pictures since not all of detected features in the faces are always in the same position. Also, they are not always the same size so resizing important features and applying them could be a possible improvement-strategy too. In general, training on more data could also be beneficial.

We do not think that a perfect solution, so 100% accuracy will be possible, since there will always be some pictures of faces with sunglasses or with very bad lightning conditions. But we think that with some further work and improvement of the algorithm, an even higher accuracy could be possible.