Problem definition:

Given two images we want to assert whether the images were produced by the same user or not. Instead of tackling the problem as a regression model, we modeled it as a classification problem. The classification problem is posed as following:

1. The program in presented with two images.
2. The output from the program is ‘1’ if the images are from the user (similar images) and ‘0’ if the images are from different users (dissimilar images).

Data:

The data consists of 15500 images of different sizes most of them displaying the word “and” written by a user. For every user we have from 10 to 15 samples. There are 1500 users in total. We also had a few blank images which were discarded.

Method:

Since, as we mentioned before, the images were not all of the same size we rescaled them all to 40x40 pixels. We chose these dimensions basing on the dimensions that were used in “The role of Size Normalization on the Recognition Rate of Handwritten Numerals” by Chun Lei He, Ping Zhang, Jianxiong Dong, Ching G. Suenang Tien D. Bui.

In order to create training set we split the data in to two categories.

The first one which we call “positive dataset” contains pairs of images coming from the same users. In order to populate this category we created all the combination of images from the same users, and tagged with 1.

Let n(u) be the size of the positive dataset for a user u.

The second category called “negative dataset” contains pairs of images coming from different users. In order to populate this category: for every user u we take n(u) images from the other users and we pair them with n(u) images randomly selected from the positive dataset of the user u.

FEATURE EXTRACTION:

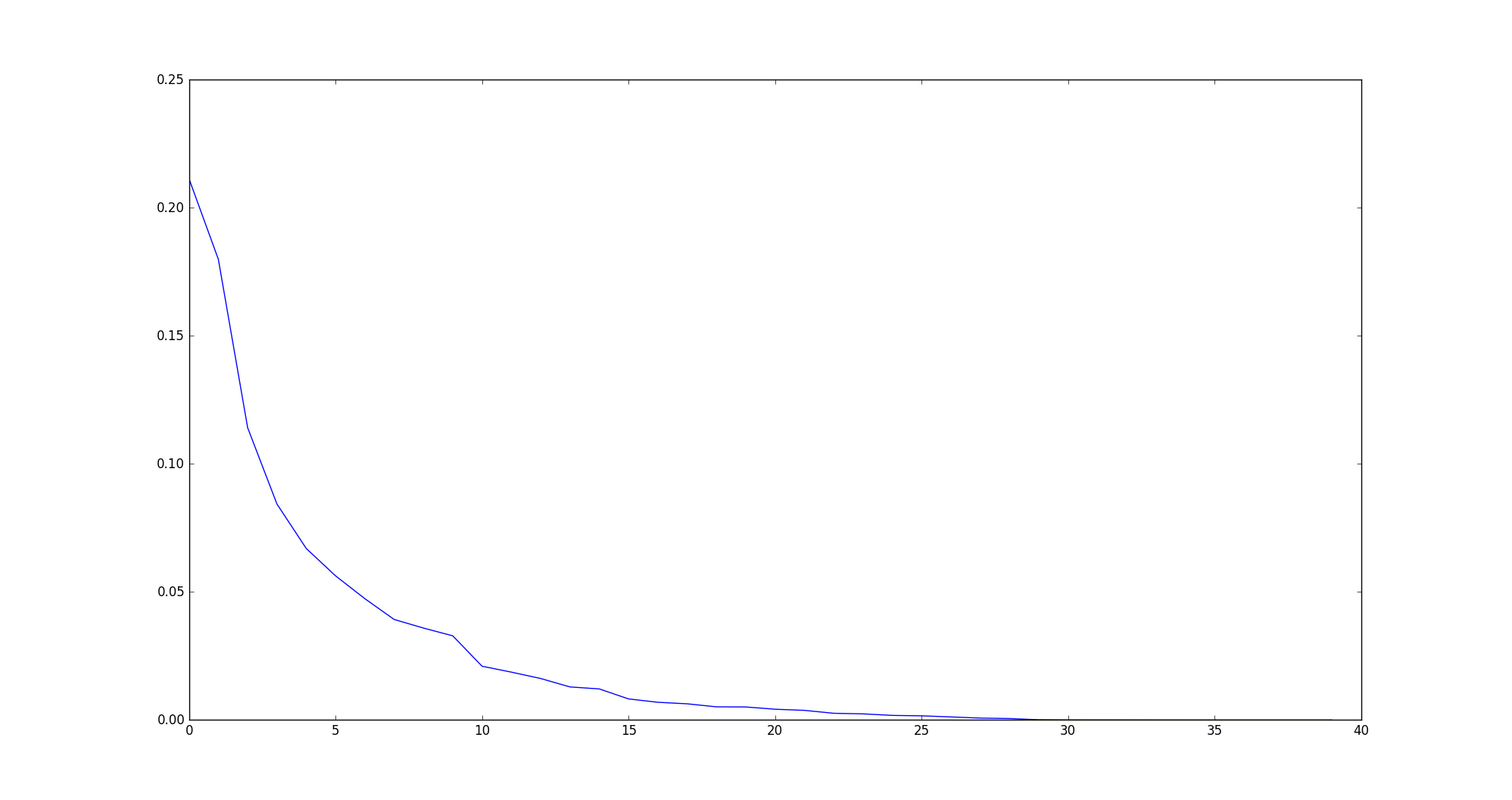
Autoencoders:

The architecture of autoencoders is as follows: 3 hidden layers with decreasing layer size. The outermost layers were of size 600 (number of neurons). The middle layer is of size 200. We used hyperbolic tangent as transfer function and min squared error as the cost function. The the network is trained using adaptive gradient descent with learning rate of 0.05.

Since the number of training examples was too low batch gradient was not used.

The middle layer activations were used as features for classification.

PCA: We applied PCA transformation to differences of the elements of pairs coming from the positive dataset and negative dataset. We decided to remove 15 dimensions. Because on average the plot of the pca variances would look like the following.

Then we used the first 25 eigen vectors as the features for the classification.

Hand Engineered Features:

We extracted 10 features from the image pairs which use different metrics to quantify the dissimilarity/similarity between pairs of images. The metrics are:

1) Pearson’s coefficient

2) Housdroff’s distance

3) Spearman Rho coefficient

4) Tanimoto index

5) Kendall index

6) Mean squared deviation

7) Median of mean squared deviation

8) Structural similarity index

9) Norm of image 1

10) Norm of image 2

RESULTS:

Both the methods based on autoencoders and PCA performed pretty badly, with an accuracy on test which is just slightly better than chance. On average we had 51% where the chance was 50%. We instead performed well on training set achieving 85%, so this made us think we overfitted the data. We tried to regularize (on the PCA method) the loss function (cross entropy) using the l2 norm regularization over the weights. However the performances did not improve. The method based on hand engineered features extraction performed relatively better.

We experienced an average of 70% on test and 85% on training data.