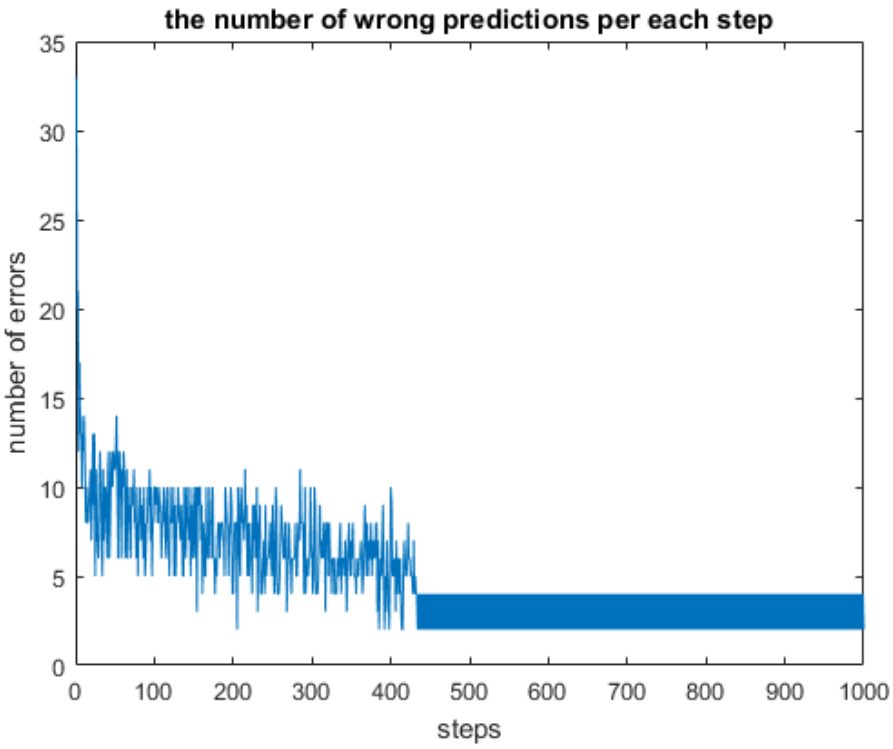
HW1 Summary PDF

Perceptron learning rule

In this homework. We attempt to create a perceptron capable of categorizing whether one viewer likes a certain movie or not depending on the movie's features. In this exercise, the feature **x** of a given movie is a 1x10 vector consisting of ones or zeros. The attached code attempts to use the perceptron learning rule to find the weight vector **w** that allows us to predict whether the viewer will like a movie or not.

This perceptron learned over all two hundred movies and the learning process was repeated a thousand times.

The results are the following:

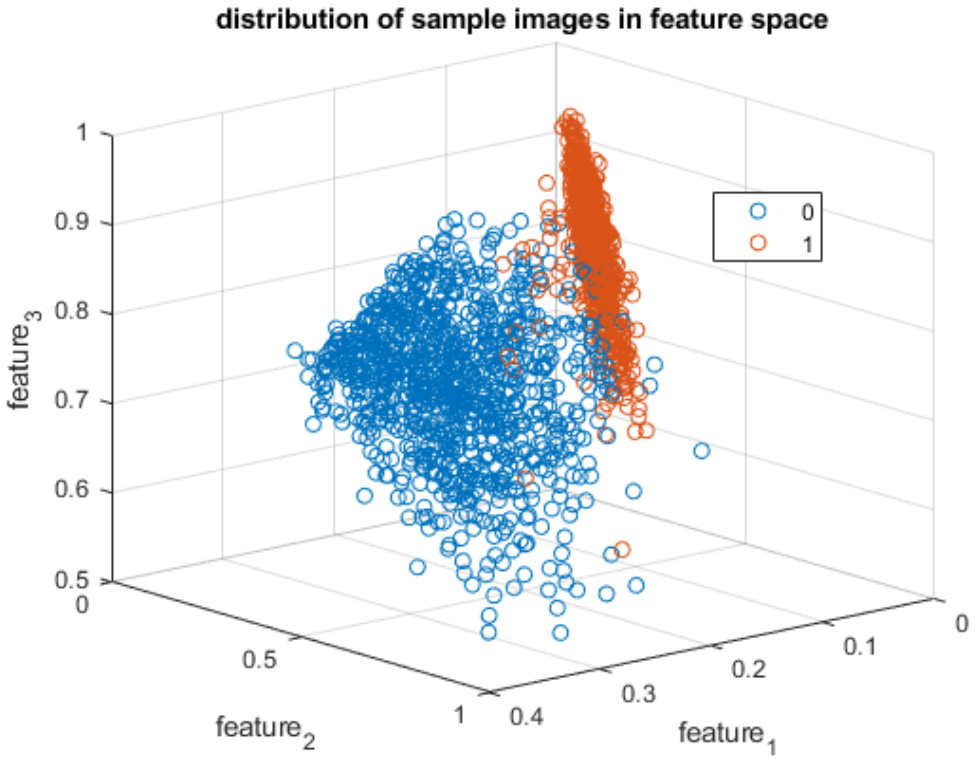


The number of wrong predictions per step decreases from 30 errors (15% of the total number of movies) on the first step to less than 5 errors (2.5%). The weight vector **w** at the end of the learning phase is the following:

w = [16.39 -16.78 21.54 18.41 -12.23 2.18 -6.33 -25.38 24.81 -35.30]

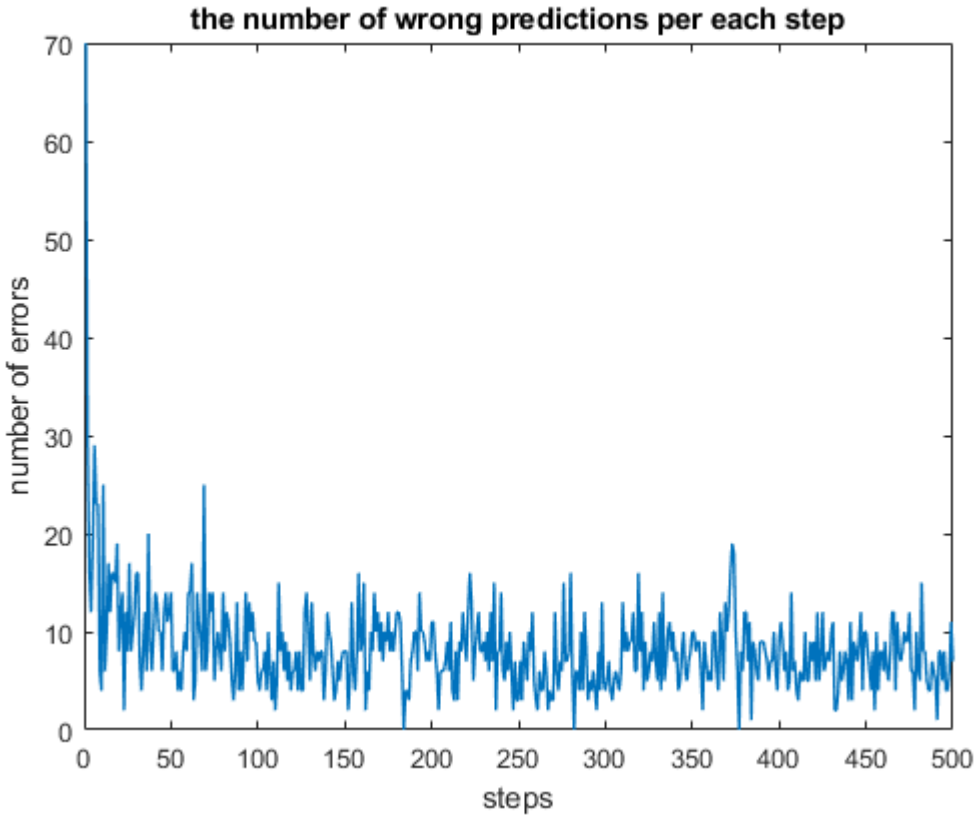
Extra credit: perceptron, handwritten digits.

In this homework, we attempt to classify handwritten digits, 0 and 1. We train a perceptron to categorize image files that show either 0 or 1. Each image is a matrix of 28x28 cells where each cell contains a value between 0 and 255. We convert each value into either 1 or 0 (1 if higher than 100, 0 if not) to convert each image into a matrix of binary values. Then we extract three features from the data: The average pixel count (**feature 1)**, the average pixel count at the centre of the image (**feature 2)**, and symmetry of the image along both x and y axis (**feature 3)**. (see code for more detail on how to extract these features). All features are characterized by values between 0 and 1. These features are saved into an input matrix, and used to train the perceptron.



The training is repeated 500 times, and the error is calculated by randomly selecting 200 images and testing whether the image category categorized by the perceptron corresponds to that of the actual image.

The results are the following:



The number of wrong predictions per step decreases from 70 errors (35% of the total number tested image samples) on the first step to around 10 errors (5%). The weight vector **w** at the end of the learning phase is the following:

w = [-22.83 4.06 2.10]