Politechnika Warszawska - Wydział Elektroniki i Technik Informacyjnych

Pattern recognition - Laboratorium 3 - Linear Classification

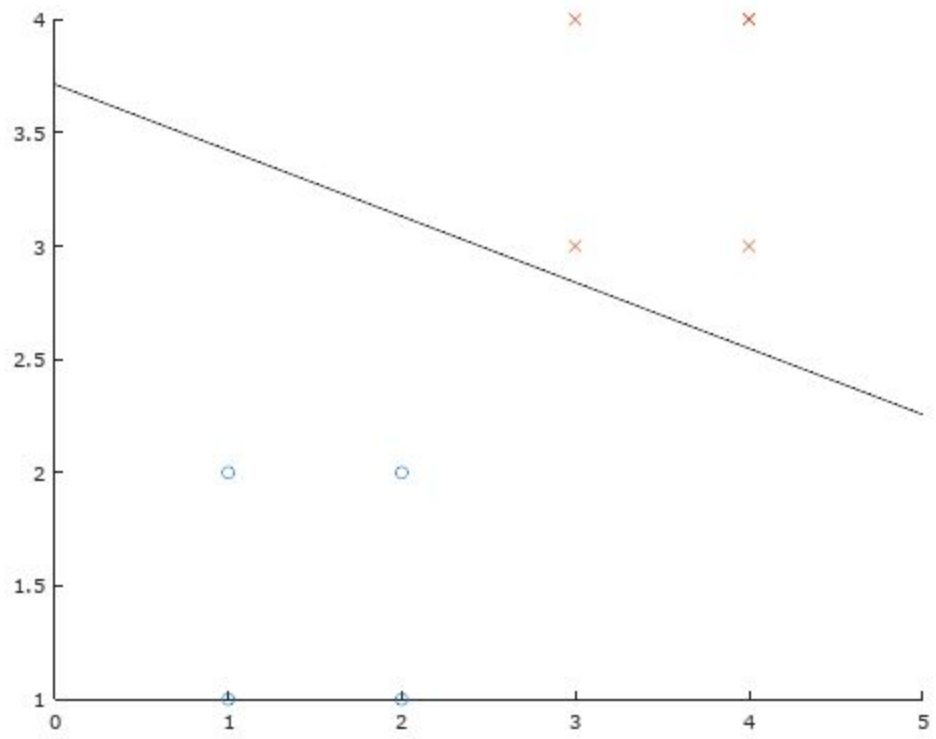
Wojciech Korzeniowski

Introduction

I will check how well perceptron classifier works in ensemble to classify digits from MNIST data. Ensemble is build from perceptron classifiers, each per pair of classes. In MNIST data there will be numbers from 0 to 9, total number of classes 10, the number of all possible combinations is equal to $\binom{10}{2} = 45$. Then for each sample, all of the 45 perceptions take a vote and decide on which of 2 given classes the sample fits the most. If there is a class, that for given sample was selected by all perceptrons that involves this class then the sample is classified as the class.

Task 1

I prepared perceptron classifier and checked it on simple test set. Following figure presents training set and the separation plane that was found by perceptron. Blue circles are the first class and red crosses are the other.



Task 2

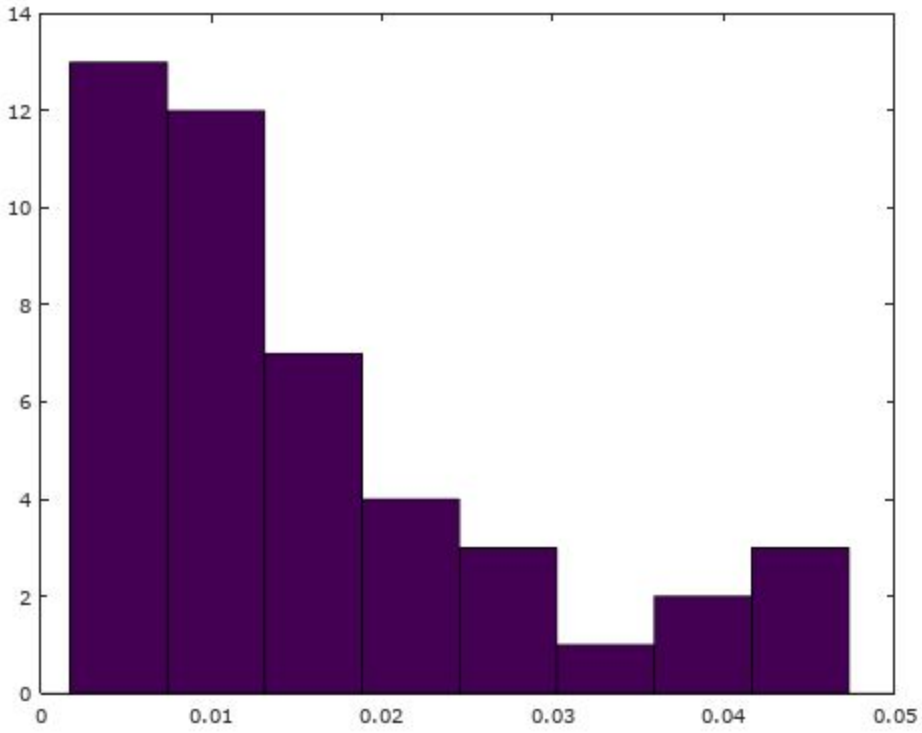
I tested perceptron algorithm against MNIST data set. Before classification I want to reduce number of features. Raw data consists of 784 features (28 x 28 pixels) so I used PCA algorithm extract 50 primary components. I created multiclass classifier using one versus one ensemble.

Following table shows error value for each classifier that distinguishes between two different classes from dataset.

Class 1	Class 2	Error
4	6	4.74%
8	10	4.59%
6	9	4.22%
4	9	3.71%
5	10	3.70%
3	9	3.11%
3	4	2.83%
3	6	2.61%
6	7	2.53%
9	10	2.01%
3	5	1.98%
3	7	1.95%
2	9	1.91%
6	10	1.76%
4	10	1.74%
3	8	1.68%
1	6	1.54%
4	8	1.48%
3	10	1.44%
5	8	1.35%
5	6	1.28%
1	3	1.20%
5	7	1.10%
7	9	1.05%
8	9	1.03%
2	4	1.01%
1	7	0.96%
2	3	0.94%
5	9	0.92%
1	9	0.89%
6	8	0.82%
1	4	0.75%

2	6	0.69%
1	10	0.65%
4	5	0.63%
2	8	0.62%
4	7	0.55%
2	10	0.50%
1	5	0.44%
2	5	0.44%
1	8	0.38%
7	10	0.35%
7	8	0.28%
2	7	0.23%
1	2	0.17%

Following figure shows histogram of previous errors.



Task 3

Currently, classification results are following:

Ok	Error	Rejection
90.88%	5.29%	3.83%

	Guess											
Truth	943	0	3	2	0	6	5	1	0	2		18
	0	1102	3	2	0	2	1	3	6	1		15
	0	6	937	6	5	1	6	6	11	1		53
	2	0	8	909	2	20	1	8	10	5		45
	1	0	2	1	893	1	4	8	4	18		50
	8	2	3	41	3	759	7	0	24	6		39
	8	1	9	2	7	19	878	2	2	0		30
	0	2	16	7	8	0	0	935	2	21		37
	3	1	1	24	4	28	5	6	846	5		51
	3	4	1	7	23	6	0	30	4	886		45

We can manipulate the threshold at which the model decides whether classify given sample or mark it as rejected to classify. The impact should be following:

- Increase of correctly classified samples
- Increase of wrongly classified samples
- Decrease the number of rejection

I changed the way of deciding what class should be picked based on the results of the OVO-voting. So far I was picking a class that collected all 9 votes. I changed this to pick a class that have the highest number of votes unless there is other class that contains the same number of votes, then mark sample as reject. Updated classification results:

Ok	Error	Rejection
91.41%	5.81%	2.78%

	Guess											
Truth	945	0	4	2	0	8	5	1	1	2		12
	0	1103	3	2	0	3	2	3	6	1		12
	1	6	943	7	6	2	8	8	14	2		35
	2	0	9	915	2	21	1	8	11	6		35
	1	0	3	2	909	2	6	8	4	18		29
	8	2	3	43	3	763	8	0	24	7		31
	8	2	11	2	7	20	882	4	2	0		20
	0	4	17	9	8	0	0	939	2	22		27
	3	1	1	24	5	32	5	6	854	5		38
	4	4	2	8	23	6	0	31	4	888		39