НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Дисциплина: «Анализ данных»

Домашнее задание на тему:
«Лабораторная работа №4»

Выполнил: Осипов Лев,
студент группы 301ПИ (1).

Москва, 2015 г.

СОДЕРЖАНИЕ

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

Сдана на семинаре.

ПРАКТИЧЕСКАЯ ЧАСТЬ

ЗАДАНИЕ 1

Для решения задания была написана программа, обучающая персептрон с фиксированной скоростью обучения.

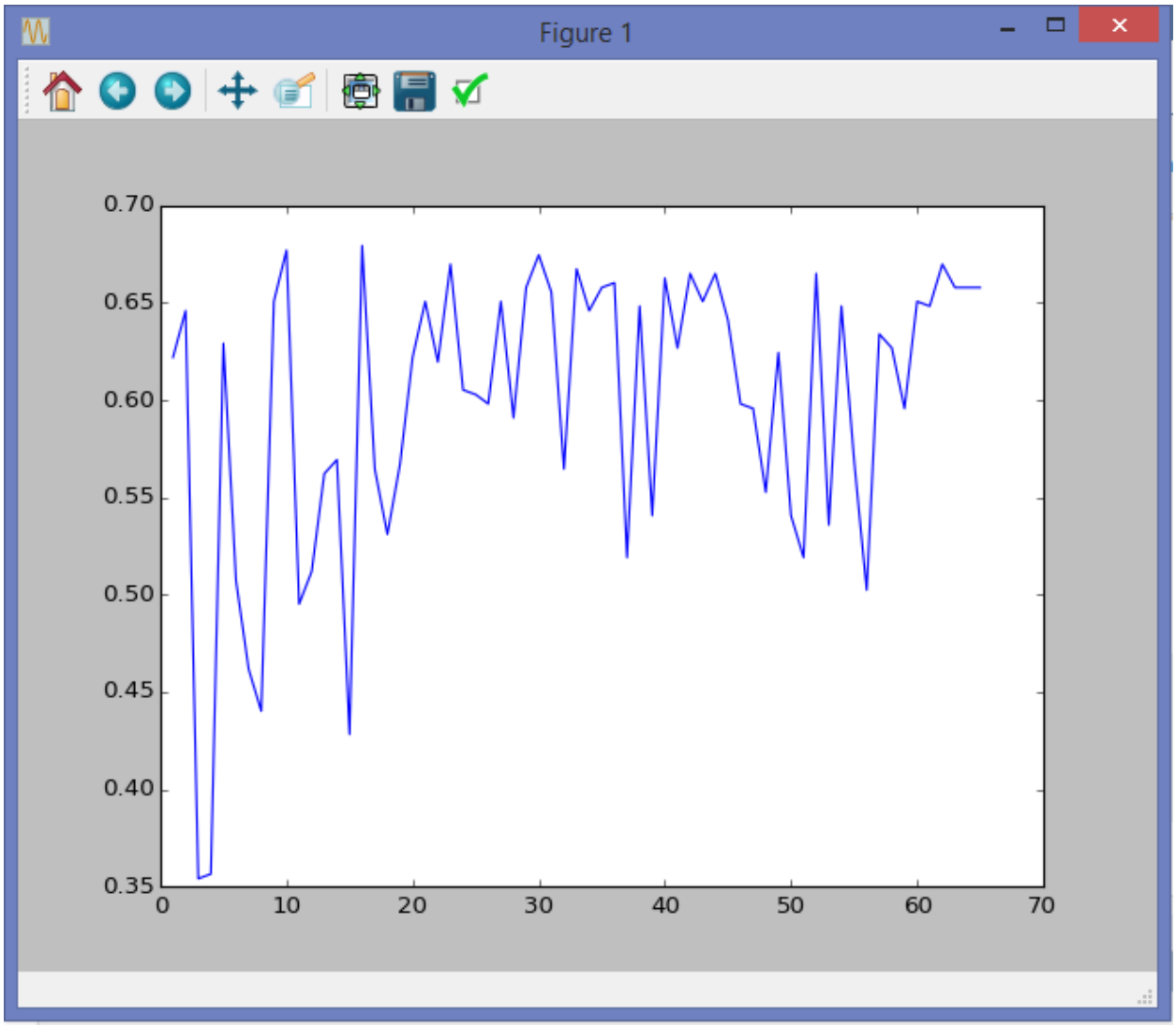

Рис. 1. Персептрон с фиксированной скоростью обучения. Зависимость величины ошибки от итераций

По результатам программы (Рис. 1) видно, что с фиксированной скоростью обучения после некоторого количества итераций алгоритм начинается переобучаться и выдавать хаотичные данные.

СПИСОК ЛИТЕРАТУРЫ

1) **Анализ данных (Программная инженерия)** – http://wiki.cs.hse.ru/%D0%90%D0%BD%D0%B0%D0%BB%D0%B8%D0%B7_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85_%28%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%B0%D1%8F_%D0%B8%D0%BD%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%B8%D1%8F%29#.D0.9E.D1.84.D0.BE.D1.80.D0.BC.D0.BB.D0.B5.D0.BD.D0.B8.D0.B5_.D0.BF.D0.B8.D1.81.D0.B5.D0.BC

ТЕКСТ ПРОГРАММЫ

```
__author__ = 'Lev Osipov'

import numpy as np
import pandas as pd
import pylab as pl


def f(wghts, vec):
    result = np.dot(wghts, vec)
    if result > 0:
        return 1
    elif result < 0:
        return -1
    else:
        return 0


def iteration(wghts, lrn_rt, tr_classes, tr_data):
        for j in range(0, 10):
            i = np.random.randint(tr_data.shape[0])
            wghts += lrn_rt * (tr_classes[i] - f(wghts,
tr_data.iloc[i])) * tr_data.iloc[i]


def test(wghts, data, classes):
    result = 0
    for i in range(0, data.shape[0]):
        if f(wghts, data.iloc[i]) == classes[i]:
            result += 1

    return result / float(len(data))


train_data = pd.read_csv('train.csv')
test_data = pd.read_csv('test.csv')

train_classes = train_data['class']
train_data = train_data.drop('class', axis=1)

test_classes = test_data['class']
test_data = test_data.drop('class', axis=1)

learning_rate = 1
max_iterations = 100

weights = np.zeros(train_data.shape[1])
results = []
local_max = 0
local_max_index = -1
for i in range(0, max_iterations):
    if local_max_index + 50 <= i:
```

```
        break
    iteration(weights, learning_rate, train_classes, train_data)
    learning_rate *= 0.95
    success_rate = test(weights, test_data, test_classes)
    if local_max < success_rate:
        local_max = success_rate
        local_max_index = i
    results.append([int(i + 1), success_rate])
results = np.array(results)
pl.plot(results[:, 0], results[:, 1])
pl.show()
```