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 часть3**

**Практическая часть3**

**Задание 13**

**Список литературы4**

**Текст программы5**

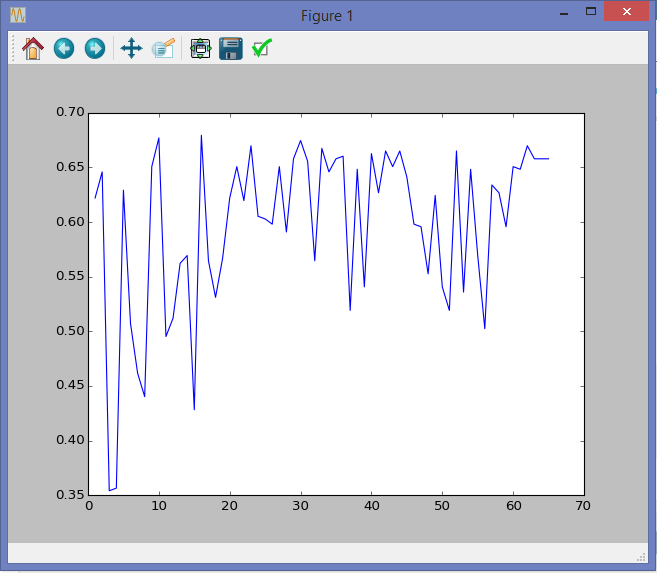
**ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

**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()