

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ  
НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ЭКОНОМИКИ И УПРАВЛЕНИЯ**

Кафедра информационных технологий

**ЛАБОРАТОРНАЯ РАБОТА № 1**

**Градиентный алгоритм машинного обучения**

Дисциплина: Информационные технологии

Номер группы:

Выполнил:

Номер варианта:

Проверила: Павлова А.И

Дата регистрации на кафедре: «\_\_» \_\_\_\_\_ 20\_\_ г.

Новосибирск 20\_\_

## Содержание работы:

1. Реализуйте алгоритм градиентного спуска на языке Python
2. Самостоятельно выполните обучение сети по трем различным вариантам с исходными данными весовые коэффициенты, смещение, коэффициенты скорости обучения, количество итераций примите равным 1000.
3. Приведите программный код, рисунки с результатами, заполните таблицу с результатами вычислений и исходными параметрами сети. Сделайте вывод.

Таблица – Исходные данные и результаты обучения сети градиентным методом.

Вариант	Начальные значения		Коэффициент скорости обучения	Функция ошибки обучения	Расчетные значения	
	W весовые коэффициенты	b смещение			W весовые коэффициенты	b смещение
Вариант 1						
Вариант 2						
Вариант 3						

### Программный код

```
import numpy as np
import matplotlib.pyplot as plt

%matplotlib inline

X = [0,0.12,0.25,0.27,0.38,0.42,0.44,0.55,0.92,1.0]
Y = [0,0.15,0.54,0.51, 0.34,0.1,0.19,0.53,1.0,0.58]

costs = []
#Step 1:
W = 0.45
b = 0.75

for i in range(1, 100):

#Step 2:
    Y_pred = np.multiply(W, X) + b
    Loss_error = 0.5 * (Y_pred - Y)**2
    cost = np.sum(Loss_error)/10

#Step 3:
    db = np.sum((Y_pred - Y))
    dw = np.dot((Y_pred - Y), X)
    costs.append(cost)

#Step 4:
    W = W - 0.01*dw
    b = b - 0.01*db

    if i%10 == 0:
        print("Cost at", i,"iteration = ", cost)

#Step 5:

#Plot the cost against no. of iterations
print("W = ", W,"& b = ", b)
plt.plot(costs)
plt.ylabel('cost')
plt.xlabel('iterations (per tens)')
plt.show()
```

Результат выполнения программы для первого варианта с исходными данными:

W = 0.45

b = 0.75

коэффициент скорости обучения равен 0,01

количество эпох: 100

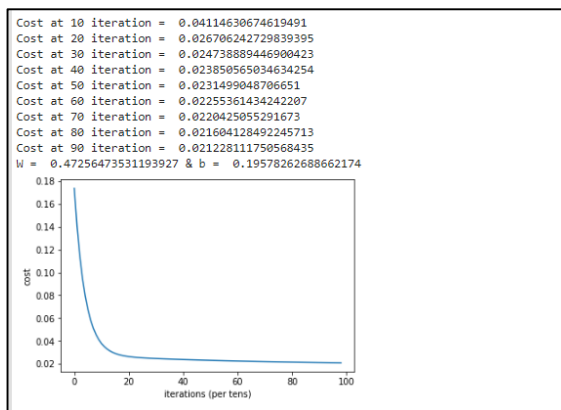


Рисунок 1 – Функция ошибки обучения сети градиентным методом (вариант 1)

## Выводы

## Список литературы