**Министерство образования Российской Федерации**

**МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**им. Н.Э. БАУМАНА**

Факультет: Информатика и системы управления Кафедра: Информационная безопасность (ИУ8)

МЕТОДЫ ОПТИМИЗАЦИИ

Лабораторная работа №5 на тему:

«***Двумерный поиск для подбора коэффициентов простейшей нейронной сети на примере решения задачи линейной регрессии экспериментальных данных***»

Вариант 5

**Преподаватель:**

Коннова Н. С.

**Студент**:

Григорьев А. С.

**Группа:**

ИУ8-32

Москва 2020

# Цель работы

Знакомство с простейшей нейронной сетью и реализация алгоритма поиска ее весовых коэффициентов на примере решения задачи регрессии экспериментальных данных.

# Постановка задачи

Найти линейную регрессию функции (коэффициенты наиболее подходящей прямой ) по набору ее дискретных значений, заданных равномерно на интервале со случайными ошибками . Выполнить расчет параметров градиентным методом. Провести двумерный пассивный поиск оптимальных весовых коэффициентов нейронной сети (НС) регрессии.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № пп |  |  |  |  |  |  | Алг. поиска | Алг. поиска |
| 5 | –500 | 200 | 0 | 10 | 24 | 1000 | МНК | МНК |

# Ход работы

Коэффициенты будем искать по методу наименьших квадратов:

**Рис. 1** График функции f(x) = -500x + 200 с шумом и без него

**Табл. 1** Точки незашумленной функции.



Параметры функции, найденные по методу наименьших квадратов:

с =-505,217, d = 235,507

Аналогичные вычисления требуется проделать для зашумленной функции.

# Табл. 2 Точки зашумленной функции.

# 

Параметры функции, найденные по методу наименьших квадратов:

с = -496,654, d = 237,759

***Рис. 2*** *Графики функции по вычисленным параметрам c и d*

# Выводы

В ходе проделанной работы был проведено знакомство с простейшей нейронной сетью и был проведен поиск ее весовых коэффициентов по методу наименьших квадратов на примере решения задачи регрессии экспериментальных данных.

# Вопросы

1. Поясните суть метода наименьших квадратов.

Метод наименьших квадратов (МНК) — математический метод, применяемый для решения различных задач, основанный на минимизации суммы квадратов отклонений некоторых функций от искомых переменных. {

Пусть  - набор неизвестных переменных (параметров),  - совокупность функций от этого набора переменных. Задача заключается в подборе таких значений , чтобы значения этих функций были максимально близки к некоторым значениям . Суть МНК заключается в выборе в качестве «меры близости» суммы квадратов отклонений левых и правых частей .

\displaystyle y\_{i}}

# Приложение А.

*Файл ‘lab05.cpp’.*

#include <iostream>

#include <ctime>

#include <vector>

double f(double c, double d, double x) {

return (c \* x + d);

}

double x\_rand(double xMin, double xMax) {

double x = (double)rand() / RAND\_MAX;

return xMin + x \* (xMax - xMin);

}

double calc\_c\_mnk(int N, double sum\_x, double sum\_y, double sum\_xy, double sum\_xx) {

return (N \* sum\_xy - sum\_x \* sum\_y) / (N \* sum\_xx - sum\_x \* sum\_x);

}

double calc\_d\_mnk(int N, double sum\_x, double sum\_y, double c\_) {

return (sum\_y - c\_ \* sum\_x) / N;

}

int main() {

srand(time(0));

double a = 0;

double b = 10;

double c = -500; //mnk

double d = 200; //mnk

int N = 24;

int A = 1000;

double error = 0;

double x\_step = (b - a) / N;

double x\_point = 0;

std::vector<std::pair<double, double>> y\_vec;

std::vector<std::pair<double, double>> y\_vec\_error;

for (int i = 0; i <= N; i++) {

x\_point = a + i \* x\_step;

y\_vec.push\_back(std::make\_pair(x\_point, f(c, d, x\_point)));

error = A \* x\_rand(-0.5, 0.5);

y\_vec\_error.push\_back(std::make\_pair(x\_point, f(c, d, x\_point) + error));

}

double sum\_x = 0, sum\_y = 0, sum\_xy = 0, sum\_xx = 0;

for (auto& el : y\_vec) {

sum\_x += el.first;

sum\_y += el.second;

sum\_xy += el.first \* el.second;

sum\_xx += el.first \* el.first;

std::cout << "X = " << el.first << " " << "Y = " << el.second << std::endl;

}

double c\_ = calc\_c\_mnk(N, sum\_x, sum\_y, sum\_xy, sum\_xx);

double d\_ = calc\_d\_mnk(N, sum\_x, sum\_y, c\_);

std::cout << "Without noise: " << c\_ << " " << d\_ << std::endl;

sum\_x = 0, sum\_y = 0, sum\_xy = 0, sum\_xx = 0;

for (auto& el : y\_vec\_error) {

sum\_x += el.first;

sum\_y += el.second;

sum\_xy += el.first \* el.second;

sum\_xx += el.first \* el.first;

std::cout << "X = " << el.first << " " << "Y = " << el.second << std::endl;

}

c\_ = calc\_c\_mnk(N, sum\_x, sum\_y, sum\_xy, sum\_xx);

d\_ = calc\_d\_mnk(N, sum\_x, sum\_y, c\_);

std::cout << "With noise: " << c\_ << " " << d\_ << std::endl;

}