Министерство образования и науки Российской Федерации

Калужский филиал

федерального государственного бюджетного образовательного

учреждения высшего образования

**«Московский государственный технический университет**

**имени Н.Э. Баумана**

**(национальный исследовательский университет)»**

(КФ МГТУ им. Н.Э. Баумана)

М.Г. Семененко

(к.ф.-м.н. доцент)

**ЛАБОРАТОРНАЯ РАБОТА № 3**

**по курсу «Методы машинного обучения»**

## Методы минимизации

Калуга

2025

**Поиск экстремума функции нескольких переменных**

Поиск экстремумов, в частности, минимума функции потерь, является центральной задачей процесса обучения любой нейросети. Весь процесс обучения представляет собой целенаправленную оптимизацию, где алгоритмы шаг за шагом корректируют параметры, двигаясь в направлении антиградиента, чтобы попасть в низшую точку сложной многомерной поверхности.

Если алгоритм оптимизации «застревает» в локальном минимуме, результаты будут некорректными. Сложные функции потерь делают задачу поиска экстремума нетривиальной. Задача решается выбором правильного метода оптимизации, скорости обучения и методов регуляризации.

**Нахождение минимума функции одной переменной методом половинного деления**

//Метод половинного деления для нахождения минимума в градиентном спуске

double Dihotomia(double a0, double b0, double epsilon, double x, double y)

{

//Номер шага

int k;

//Отклонение от середины отрезка влево, вправо

double lk, mk;

//Величина на которую мы отклонимся от середины отрезка

double delta=0.5\*epsilon;

//Точка минимума

double x\_;

//Отрезок локализации минимума

double ak=a0, bk=b0;

k=1;

//Пока длина отрезка больше заданной точности

do

{

//Берем середину (ну почти середину - +\- некоторое дельта в частности у нас delta=0.5\*epsilon)

lk=(ak+bk-delta)/2;

mk=(ak+bk+delta)/2;

k++;

//Проверяем в какую часть попадает точка минимума слева от разбиения или справа и выбираем соответствующую точку

if(g(x,y,lk)<=g(x,y,mk))

{

//Теперь правая граница отрезка локализации равна mk

bk=mk;

}

else

{

//Теперь левая граница отрезка локализации равна mk

ak=lk;

}

} while ((bk-ak)>=epsilon);

x\_=(ak+bk)/2; //minimum point

return x\_;

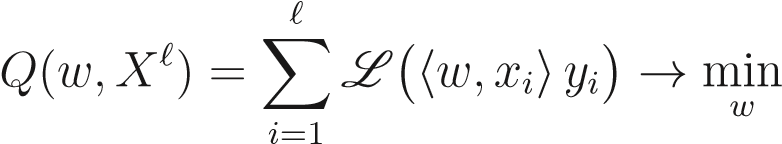
}

**Метод стохастического градиента**

Пусть задана обучающая выборка



Требуется найти вектор весов w∈ Rn, при котором достигается минимум функции



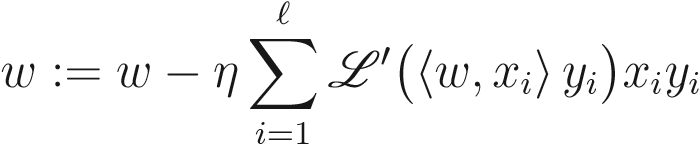
.

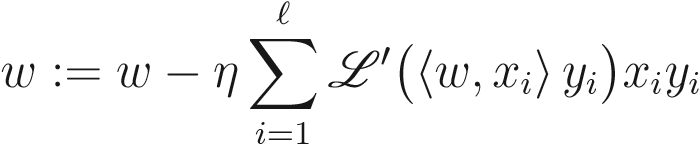
Применим для минимизации Q(w) метод градиентного спуска. В этом методе выбирается некоторое начальное приближение для вектора весов w, затем запускается итерационный процесс, на каждом шаге которого вектор w изменяется в направлении наиболее быстрого убывания функционала Q. Это направление противоположно вектору градиента :



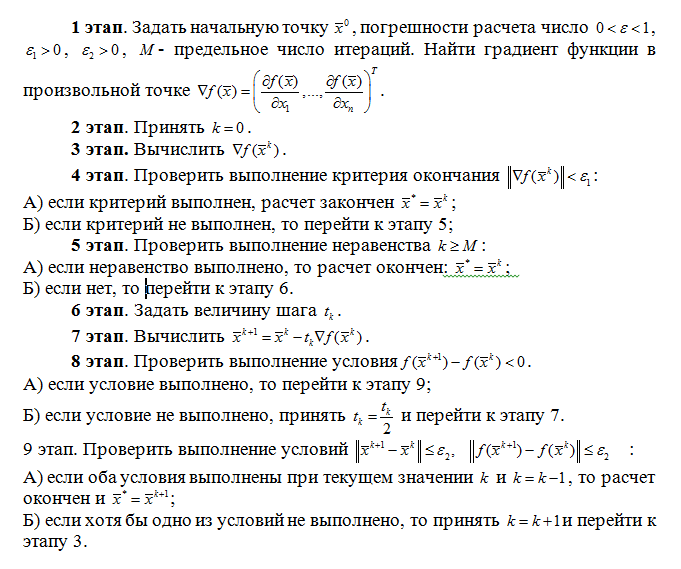
w:= w− ηQ′(w),

где η>0 — величина шага в направлении антиградиента, называемая также темпом обучения (learning rate). Предполагая, что функция потерь L дифференцируема, распишем градиент:





Алгоритм метода градиентного спуска состоит из следующих этапов:



**Задание**

1. Написать блок-схему алгоритма нахождения минимума функции методом половинного деления.
2. Методом половинного деления найти минимум функции J(u) на отрезке [-10; 10] (по вариантам). Построить график функции и показать на нем точку минимума.

**Изображение выглядит как текст, снимок экрана, число, чек

Содержимое, созданное искусственным интеллектом, может быть неверным.**

1. Сравнить результат с результатом использования встроенной функции.
2. Написать блок-схему алгоритма нахождения минимума функции двух переменных методом градиентного спуска.
3. Допустим, что задана решающая функция линейного классификатора в упрощенном виде (по вариантам).

Найти координаты и значение функции в точке минимума методом градиентного спуска.

1. Сравнить результат с результатом использования встроенной функции (показать график).

|  |  |
| --- | --- |
| **№ варианта** | **Функция** |
| **1** |  |
| **2** |  |
| **3** |  |
| **4** |  |
| **5** |  |
| **6** |  |
| **7** |  |
| **8** |  |
| **9** |  |
| **10** |  |
| **11** |  |
| **12** |  |
| **13** |  |
| **14** |  |
| **15** |  |

**Пример кода метода градиентного спуска**

**Изображение выглядит как текст, снимок экрана, Шрифт, алгебра

Содержимое, созданное искусственным интеллектом, может быть неверным.**

**Изображение выглядит как текст, снимок экрана, Шрифт, документ

Содержимое, созданное искусственным интеллектом, может быть неверным.**