

**Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки**

Лабораторна робота №3.2

з дисципліни
«Інтелектуальні вбудовані системи»

на тему
«ДОСЛІДЖЕННЯ НЕЙРОННИХ МЕРЕЖ. МОДЕЛЬ PERCEPTRON»

Виконав:

студент групи ІП-84

Валигні Андрій Олександрович
номер залікової книжки: 8503

Перевірив:

ас. Регіда П. Г.

Київ 2020

Мета роботи - ознайомлення з принципами машинного навчання за допомогою математичної моделі сприйняття інформації Перцептрон(Perceptron). Змодельювати роботу нейронної мережі та дослідити вплив параметрів на час виконання та точність результату

Основні теоретичні відомості

Важливою задачею яку система реального часу має вирішувати є отримання необхідних для обчислень параметрів, її обробка та виведення результату у встановлений дедлайн. З цього постає проблема отримання водночас точних та швидких результатів. Модель Перцептрон дозволяє покроково наближати початкові значення. Розглянемо приклад: дано дві точки A(1,5), B(2,4), поріг спрацювання $P = 4$, швидкість навчання $\delta = 0.1$. Початкові значення ваги візьмемо нульовими $W1 = 0$, $W2 = 0$. Розрахунок вихідного сигналу у виконується за наступною формулою: $x1 * W1 + x2 * W2 = y$ Для кожного кроку потрібно застосувати дельта-правило, формула для розрахунку похибки: $\Delta = P - y$ де y – значення на виході. Для розрахунку ваги, використовується наступна формули: $W1(i+1) = W1(i) + \Delta * x1$ $W2(i+1) = W2(i) + \Delta * x2$ де i – крок, або ітерація алгоритму.

Код програми

```
perceptron(int endPoint, double learningRate, int maxIterations, double
maxTime) {

    double w1 = 0;

    double w2 = 0;

    const input = [
        [0,6],
        [1,5],
        [3,3],
        [2,4],
```

```

];

double time = 0;
int iterations = 0;

while (maxIterations > time && maxTime > iterations) {
    var begin = DateTime.now().microsecondsSinceEpoch;
    for (final value in input) {
        var y = w1 * value[0].toDouble() + w2 * value[1].toDouble();
        var delta = endPoint - y;
        delta = delta > 0 ? delta : 0;

        w1 += delta * value[0].toDouble() * learningRate;
        w2 += delta * value[1].toDouble() * learningRate;
    }
    var end = DateTime.now().microsecondsSinceEpoch;
    time += (end - begin) / 1000;
    iterations++;
}
return [w1, w2, time, iterations];
}

```

Скріншоти



Висновок

Під час виконання лабораторної роботи я ознайомився з основами машинного навчання за допомогою математичної моделі. Розробив відповідну програму за допомогою Flutter Dart.