

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

**ЗВІТ**  
**Лабораторна робота №3.2**  
з дисципліни  
«Інтелектуальні вбудовані системи»  
на тему  
«Дослідження нейронних мереж. Модель perceptron»

Виконав:  
Василиненко Д.Д.  
Студент групи ІП-84  
Перевірив:  
Регіда Павло Геннадійович

Київ 2021

## Завдання

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.

## Основні теоретичні відомості

Поріг спрацювання:  $P = 4$  Дано точки:  $A(0,6)$ ,  $B(1,5)$ ,  $C(3,3)$ ,  $D(2,4)$ .  
Швидкості навчання:  $\delta = \{0,001; 0,01; 0,05; 0,1; 0,2; 0,3\}$  Дедлайн:  
часовий =  $\{0.5с; 1с; 2с; 5с\}$ , кількість ітерацій =  $\{100; 200; 500; 1000\}$   
Обрати швидкість навчання та дедлайн. Налаштувати Перцептрон для даних точок. Розробити відповідний мобільний додаток і вивести отримані значення. Провести аналіз витрати часу та точності результату за різних параметрах навчання.

## Вихідний код:

### PerceptronLogic.ts

```
type Input = number[]

const activations = {
  d: (x: number): number => x,
  RL: (x: number): number => Math.max(0, x),
  sig: (x: number): number => 1 / (1 + Math.exp(-x)),
}

export class PerceptronClass {
  public weights: number[] = [0, 0]
  public accuracy: number = 0
  protected threshold: number = 1
  protected learningRate: number = .1
  private activation = activations.d
  private b: number = 1
  private error = 1 / 1e6

  constructor(opts: {
    threshold?: number,
    learningRate?: number,
  }) {
    Object.assign(this, opts);
  }

  public train(points: Input[]): boolean {
    let success = true
    points.forEach((point, i) => {
      success = this.adjustWeights(point, i) && success
    })

    return success
  }

  public predict(point: Input): number {
    return this.activation(this.sum(point))
  }

  public learn(points: Input[], deadline: number = 100) {
    while (true) {
      if (deadline / points.length <= 0 || this.train(points)) {
        break;
      }
    }
  }
}
```

```

        }
        deadline--
    }

    return this.accuracy
}

private adjustWeights(point: Input, i: number): boolean {
    const delta = this.delta(this.predict(point));
    if (Math.abs(delta) < this.error || delta * Math.pow(-1, i) < 0) {
        return true
    }
    this.weights = this.weights.map((w, i) => {
        return w + delta * point[i] * this.learningRate
    });
    this.accuracy = 1 - delta;
    return false
}

private delta(y: number): number {
    return this.threshold - y
}

private sum(point: Input): number {
    return point.reduce((sum, x, i) => sum + x * this.weights[i], 0) +
this.b
}
}

```

## Результати роботи програми

0,001



100



COMPUTE

0,001



0,01



0,05



0,1



0,2



0,3



100



200



500



1000



COMPUTE



0,001



100



Accuracy:  
0.8982653163997707

COMPUTE

## **Висновки**

Під час виконання даної лабораторної роботи ми ознайомилися з принципами машинного навчання за допомогою математичної моделі сприйняття інформації Перцептрон(Perceptron) та змоделювали роботу нейронної мережі та дослідити вплив параметрів на час виконання та точність результату. Було створено програму у вигляді мобільного застосунку з користувацьким інтерфейсом, що дозволяє задавати швидкість навчання та різні варіанти дедлайну.