

Міністерство освіти і науки України
Національний технічний університет України «КПІ» імені Ігоря Сікорського
Кафедра обчислювальної техніки ФІОТ

ЗВІТ
з лабораторної роботи №2а
з навчальної дисципліни «Методи наукових досліджень»

Тема:
ДОСЛІДЖЕННЯ НЕЙРОННИХ МЕРЕЖ. МОДЕЛЬ PERCEPTRON

Виконав:
Студент 2 курсу кафедри ОТ ФІОТ,
Навчальної групи ІВ-92
Хоменко Олег
Номер у списку групи: 25

Перевірив:
Регіда П.Г.

Київ 2021

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

Завдання на лабораторну роботу:

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.

Код програми:

```
package com.example.perceptron

import androidx.appcompat.app.AppCompatActivity
import android.os.Bundle
import android.view.View
import kotlinx.android.synthetic.main.activity_main.*

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }

    fun onSolveClick(view: View) {
        if (editRate.text.isEmpty() ||
            editIterations.text.isEmpty() ||
            editThreshold.text.isEmpty()) {
            return
        }

        val data = listOf(
            listOf(0.0, 0.0) to checkBoxA.isChecked(),
            listOf(1.0, 1.5) to checkBoxC.isChecked(),
            listOf(2.0, 3.0) to checkBoxB.isChecked(),
            listOf(5.0, 4.0) to checkBoxD.isChecked()
        )
        val rate = editRate.text.toString().toDouble()
        val threshold = editThreshold.text.toString().toDouble()
        val iterations = editIterations.text.toString().toInt()
        val (results, weights) = solve(data, rate, threshold, iterations)

        resultsView.text = "Results:\nClasses: " + results.joinToString(", ") +
            "\nWeights: " + weights.joinToString(", ") + "\nIterations: " +
            iterations.toString()
    }
}
```

```

package com.example.perceptron

typealias Data = List<Pair<List<Double>, Boolean>>

fun delta(expected: Double, actual: Double) = expected - actual

fun predict(actual: Double, threshold: Double) = actual > threshold

fun output(equation: List<Pair<Double, Double>>) = equation.map{it.first *
it.second}.sum()

fun train(data: Data, weights: List<Double>, rate: Double, threshold: Double,
iterations: Int): List<Double> {
    return if (iterations == 0)
        weights
    else {
        val newWeights = trainTest(data, weights, rate, threshold)
        train(data, newWeights, rate, threshold, iterations - 1)
    }
}

fun trainTest(data: Data, weights: List<Double>, rate: Double, threshold: Double):
List<Double> {
    return if (data.isEmpty()) {
        weights
    } else {
        val xs = data[0].first
        val y = output(xs zip weights)
        val expected = data[0].second
        val prediction = predict(y, threshold)
        val err = delta(threshold, y)
        val newWeights = if (expected != prediction)
            weights.zip(xs){w, x -> w + x * rate * err}
        else
            weights
        trainTest(data.drop(1), newWeights, rate, threshold)
    }
}

fun solve(data: Data, rate: Double, threshold: Double, iterations: Int):
Pair<List<Boolean>, List<Double>> {
    val startingWeights = data.map{0.0}
    val weights = train(data, startingWeights, rate, threshold, iterations)
    val results = data.map{predict(output(it.first zip weights), threshold)}
    return results to weights
}

```

Результат виконання програми:

Perceptron

Points: A(0, 6), B(1, 5), C(3, 3), D(2, 4)

Classes:

- ☐ Point A
- ☒ Point B
- ☒ Point C
- ☐ Point D

0.1

4

500

Results:

Classes: false, false, true, false

Weights: -2.8571428553417864, 4.571428569041

Iterations: 500

SOLVE

Perceptron

Points: A(0, 6), B(1, 5), C(3, 3), D(2, 4)

Classes:

- ☒ Point A
- ☒ Point B
- ☒ Point C
- ☒ Point D

0.1

4

200

Results:

Classes: false, false, true, true

Weights: 1.2307692307692295, 1.8461538461538467

Iterations: 200

SOLVE

Висновки: ознайомилися з принципами машинного навчання за допомогою математичної моделі сприйняття інформації Перцептрон(Perceptron). Змодельовали роботу нейронної мережі та дослідили вплив параметрів на час виконання та точність результату.