**Міністерство освіти і науки України**

**Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського"**

**Факультет інформатики та обчислювальної техніки**

**Кафедра обчислювальної техніки**

**ЗВІТ**

про виконання лабораторної роботи №3.2 з дисципліни

**«Інтелектуальні вбудовані системи»** на тему:

**«ДОСЛІДЖЕННЯ НЕЙРОННИХ МЕРЕЖ. МОДЕЛЬ PERCEPTRON»**

|  |  |
| --- | --- |
| Перевірив: | ВИКОНАВ: |
| асистент кафедри ОТ | студент 3 курсу |
| Регіда П. Г | групи ІП-83, ФІОТ  Васильєв Г.З.  Залікова книжка №8505  Варіант – 5 |

Київ 2021

**Завдання на лабораторну роботу**

Поріг спрацювання: P = 4

Дано точки: А(0,6), В(1,5), С(3,3), D(2,4).

Швидкості навчання: δ = {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

Програмний код:

App.js

import React, { useState } from "react";

import { StyleSheet, Text, View, SafeAreaView, Button } from "react-native";

import RNPickerSelect from "react-native-picker-select";

import Perceptron from "./Perceptron";

export default function App() {

  const [learningRate, setLearningRate] = useState(0.001);

  const [deadline, setDeadline] = useState(1000);

  const [result, setResult] = useState();

  const [time, setTime] = useState();

  return (

    <SafeAreaView style={{ flex: 1, alignItems: "center" }}>

      <RNPickerSelect

        style={pickerSelectStyles}

        onValueChange={(value) => setLearningRate(value)}

        placeholder={{ label: "Choose the learning speed", value: null }}

        items={[

          { label: "0.001", value: 0.001 },

          { label: "0.01", value: 0.01 },

          { label: "0.05", value: 0.05 },

          { label: "0.1", value: 0.1 },

          { label: "0.2", value: 0.2 },

        ]}

      />

      <RNPickerSelect

        style={pickerSelectStyles}

        onValueChange={(value) => setDeadline(value)}

        placeholder={{ label: "Choose the deadline", value: null }}

        items={[

          { label: "1000", value: 1000 },

          { label: "2000", value: 2000 },

          { label: "3000", value: 3000 },

        ]}

      />

      {<Text style={styles.result}>{[result, time]}</Text>}

      <View style={styles.btn}>

        <Button

          buttonStyle={{ backgroundColor: "black" }}

          color="white"

          title="Learn"

          onPress={() => {

            const p = new Perceptron({ threshold: 4, learningRate });

            let start = performance.now();

            const result = p.learn(

              [

                [0, 6],

                [3, 3],

                [1, 5],

                [2, 4],

              ],

              deadline

            );

            let end = performance.now();

            const diff = end - start;

            setTime(() => {

              `\nTime= ${diff}`;

            });

            diff > deadline

              ? alert("deadline exposed")

              : setResult(`${result}`);

          }}

        />

      </View>

    </SafeAreaView>

  );

}

const styles = StyleSheet.create({

  result: {

    top: 100,

    lineHeight: 30,

    alignSelf: "center",

    fontSize: 16,

  },

  btn: {

    justifyContent: "center",

    alignItems: "center",

    alignSelf: "center",

    top: 200,

    height: 50,

    width: 150,

    backgroundColor: "black",

    borderRadius: 50,

  },

});

const pickerSelectStyles = StyleSheet.create({

  inputIOS: {

    top: 20,

    width: "90%",

    alignSelf: "center",

    marginVertical: 30,

    fontSize: 20,

    paddingVertical: 12,

    paddingHorizontal: 10,

    borderWidth: 5,

    borderColor: "blue",

    borderRadius: 6,

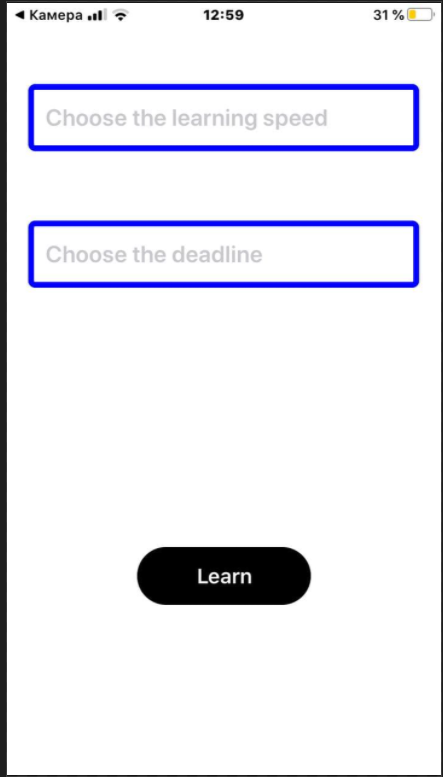
    color: "black",

  },

  inputAndroid: {},

});

Результати роботи програми:



Висновки:

Отже, в ході лабораторної роботи, ми отримали навички з мащинного навчання на прикладі математичної моделі Перцептрон та змоделювали його роботу у вигляді клієнтського додатку із зручним інтерфейсом.

Результати наведено в звіті та врепозиторії. Кінцеву мету було досягнуто.