МІНІСТЕРСТВО ОСВІТИ ТА 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 гр. IП-84 Дмитренко Олександр

Перевірив: Регіда П.Г.

Основні теоретичні відомості

Важливою задачеюяку система реального часу має вирішувати є отримання необхідних для обчислень параметрів, її обробка та виведення результату у встановлений дедлайн. З цього постає проблема отримання водночає точних та швидких результатів. Модель Перцпептрон 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:

$$x_1 * W_1 + x_2 * W_2 = y$$

Для кожного кроку потрібно застосувати дельта-правило, формула для розрахунку похибки:

$$\Delta = P - y$$

де у – значення на виході.

Для розрахунку ваги, використовується наступна формули:

$$W_1(i+1) = W_1(i) + W_2 * x_{11}$$

$$W_2(i+1) = W_1(i) + W_2 * x_{12}$$

де і – крок, або ітерація алгоритму.

Завдання

Поріг спрацювання: Р = 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 = {0.5c; 1c; 2c; 5c}, 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

Лістинг програми

```
onst activations = {
default: (x) => x,
class Perceptron {
 weights = [0, 0];
  accuracy = 0;
  threshold = 1;
  learningRate = 0.1;
  activation = activations.default;
 bias = 1;
error = 1 / 1e6;
 constructor({ threshold, learningRate }) {
  Object.assign(this, { threshold, learningRate });
  guess(point) {
   return this.predict(point) > this.threshold;
  predict(point) {
   return this.activation(this.sum(point));
 train(points) {
  let success = true;
    points.forEach((point, i) => {
      success = this.adjustWeights(point, i) && success;
  learn(points, deadline = 100) {
   while (true) {
  if (deadline / points.length <= 0 || this.train(points)) break;</pre>
      deadline--;
    return [
    `W1 = ${this.weights[0]} \nW2= ${this.weights[1]} \nAccuracy = ${this.accuracy}`,
```

```
adjustWeights(point, i) {
  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;
}

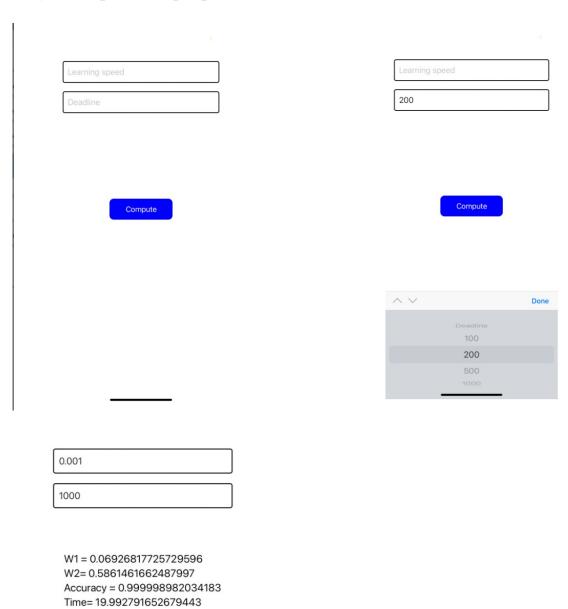
sum(point) {
  return (
    point.reduce((sum, x, i) => sum + x * this.weights[i], 0) + this.bias
    );
}

delta(y) {
  return this.threshold - y;
}
}

export default Perceptron;
```

Результат роботи програми

Compute



Висновки

Під час виконання лабораторної роботи були дослідженні принципи машинного навчання за допомогою математичної моделі сприйняття інформації Perceptron. Було змодельовано роботу нейронної мережі та досліджено вплив різних параметрів на час виконання та точність результату.