Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

*Кафедра програмного забезпечення автоматизованих систем*

**Лабораторна робота №1**

Тема:

**«Бінарний перцептрон»**

***Виконав*:**

ст. групи ПІт-15-3

Свирид О. Б.

***Перевірив:***

Дитко Т.В.

Івано-Франківськ

2015

**Мета:** вивчити принципи функціонування і навчання перцептрона.

**Короткі теоретичні відомості:**

Перше систематичне вивчення штучних нейронних мереж було зроблене Маккалоком і Пітсом в 1943 р. Пізніше в роботі вони досліджували мережеві парадигми для розпізнавання зображень, що піддаються зсувам і поворотам. Проста нейронна модель використовувалася в більшій частині їх роботи.

Елемент Σ множить кожний вхід *х* з вагою *w* і підсумовує зважені входи. Якщо ця сума більше заданого порогового значення, вихід рівний одиниці, в іншому випадку нулю. Ці системи (і множина до них подібних) отримали назву перцептронів. Вони складаються з одного прошарку штучних нейронів, сполучених за допомогою вагових коефіцієнтів з множиною входів.

Алгоритм навчання перцептрона:

1. Подаємо на вхід образ Х
2. Множимо кожен елемент xi на відповідний йому ваговий коефіцієнт wi

і знаходимо сумму цих добутків «а».

1. Якщо a > Θ, то Y = 1, інакше Y = 0, де Θ – порогове значення ( зазвичай = 0 )
2. Якщо значення Y вірне, то переходимо до наступного образу, якщо ж невірне, то підраховуємо δ=T-Y
3. Обчислюємо Δi = η\*δ\*xi де η – кооофіцієнт швидкості навчання , зазвичай ( 0; 0,1 )
4. Знаходимо нові вагові коефіцієнти wi(n+1) = wi(n) + Δi
5. Знаходимо нове значення Θ = Θ - η\*δ
6. Переходимо до наступного образу

Якщо при подачі кожного образу значення Y вірне, значить вагові коофіцієнти налаштовані вірно і ми припиняємо алгоритм навчання.

**Завдання:**

1. Програмно реалізувати наведений алгоритм
2. Використовуючи початкові дані та дані роботи програми побудувати таблицю 1.2



**Хід роботи**

Логічна функція:



**Текст програми на мові Java:**

***package ISAD.Lab\_1;***

***/\*\****

***\* Created by oleg on 13.09.2015.***

***\*/***

***public class Perceptron {***

***private static double THETA = 0; // Поріг спрацювання функції***

***private final static double H = 0.1; // Крок ( коофіцієнт швидкості навчання )***

***private double[] w = new double[3]; // Вагові коофіцієнти***

***private final static int[][] X = new int[8][3]; // x1, x2, x3***

***private final static int[] TRUTH = new int[8]; // Істинне значення функції***

***Perceptron() {***

***w[0] = 0.3;***

***w[1] = 0.4;***

***w[2] = 0.5;***

***boolean[] truthTable[] = {***

***{ false, false, false },***

***{ false, false, true },***

***{ false, true, false },***

***{ false, true, true },***

***{ true, false, false },***

***{ true, false, true },***

***{ true, true, false },***

***{ true, true, true }***

***};***

***boolean[] tBool = new boolean[8];***

***for (int i = 0; i < 8; i++)***

***tBool[i] = (!truthTable[i][0] && (!truthTable[i][1] || truthTable[i][2]));***

***for (int i = 0; i < 8; i++)***

***if (tBool[i]) {***

***TRUTH[i] = 1;***

***}***

***else {***

***TRUTH[i] = 0;***

***}***

***for (int i = 0; i < 8; i++) {***

***for (int j = 0; j < 3; j++)***

***if (truthTable[i][j]) {***

***X[i][j] = 1;***

***}***

***else {***

***X[i][j] = 0;***

***}***

***}***

***}***

***public void printTab() {***

***System.out.println("Truth table:\n\n x1 x2 x3 T");***

***for (int i = 0; i < 8; i++) {***

***for (int j = 0; j < 3; j++)***

***System.out.printf(" %1d", X[i][j]);***

***System.out.printf("%3d\n", TRUTH[i]);***

***}***

***System.out.println();***

***}***

***public void startStudyPerceptron() {***

***boolean work = true; // Відповідає за припинення роботи перцептрону***

***double a;***

***int Y;***

***int delta;***

***double w1, w2, w3, pTHETA, nTminusY, dTHETA; // Змінні які потрібні для подальшого виводу результатів***

***double[] dw = new double[3];***

***while (work) {***

***work = false;***

***for (int i = 0; i < 8; i++) {***

***a = 0;***

***for (int j = 0; j < 3; j++) {***

***a += X[i][j] \* w[j];***

***}***

***if (a > THETA) {***

***Y = 1;***

***}***

***else {***

***Y = 0;***

***}***

***if (Y != TRUTH[i]) {***

***work = true;***

***w1 = w[0];***

***w2 = w[1];***

***w3 = w[2];***

***pTHETA = THETA; // Для виводу результатів***

***delta = TRUTH[i] - Y;***

***nTminusY = delta \* H;***

***for (int j = 0; j < 3; j++) {***

***dw[j] = H \* delta \* X[i][j];***

***w[j] += dw[j];***

***}***

***dTHETA = H \* delta;***

***THETA -= dTHETA;***

***System.out.printf("%3.2f\t %3.2f\t %2.2f\t %2.2f\t %2d %2d %2d " +***

***"%2.2f %2d %2d %+4.2f %+4.2f %+4.2f %+4.2f %+4.2f\n",***

***w1, w2, w3, pTHETA, X[i][0], X[i][1], X[i][2],***

***a, Y, TRUTH[i], nTminusY, dw[0], dw[1], dw[2], dTHETA);***

***}***

***}***

***}***

***}***

***public static void main(String[] args) {***

***Perceptron perceptron = new Perceptron();***

***perceptron.printTab();***

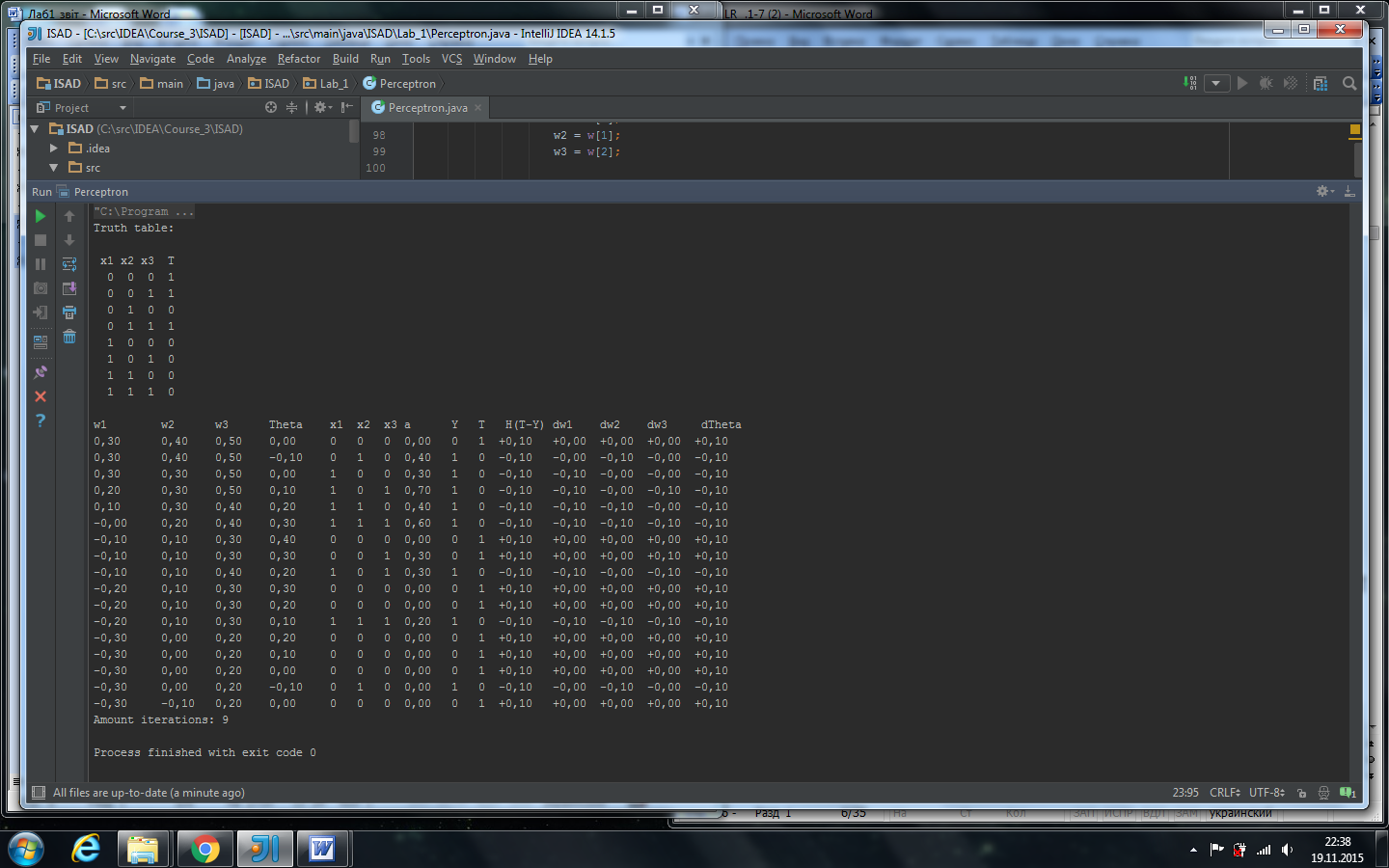
***System.out.println("w1\t\t w2\t w3\t Theta x1 x2 x3 a Y T H(T-Y) dw1 dw2 dw3 dTheta");***

***perceptron.startStudyPerceptron();***

***}***

***}***

Результат:



***Висновок:***  в ході лабораторної роботи я ознайомився з принципом роботи бінарного перцептрона і програмно реалізував алгоритм його навчання.