Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

*Кафедра програмного забезпечення автоматизованих систем*

**Лабораторна робота №5**

Тема:

**«**Мережа RBF**»**

***Виконав*:**

ст. групи ПІт-15-3

Свирид О. Б.

***Перевірив:***

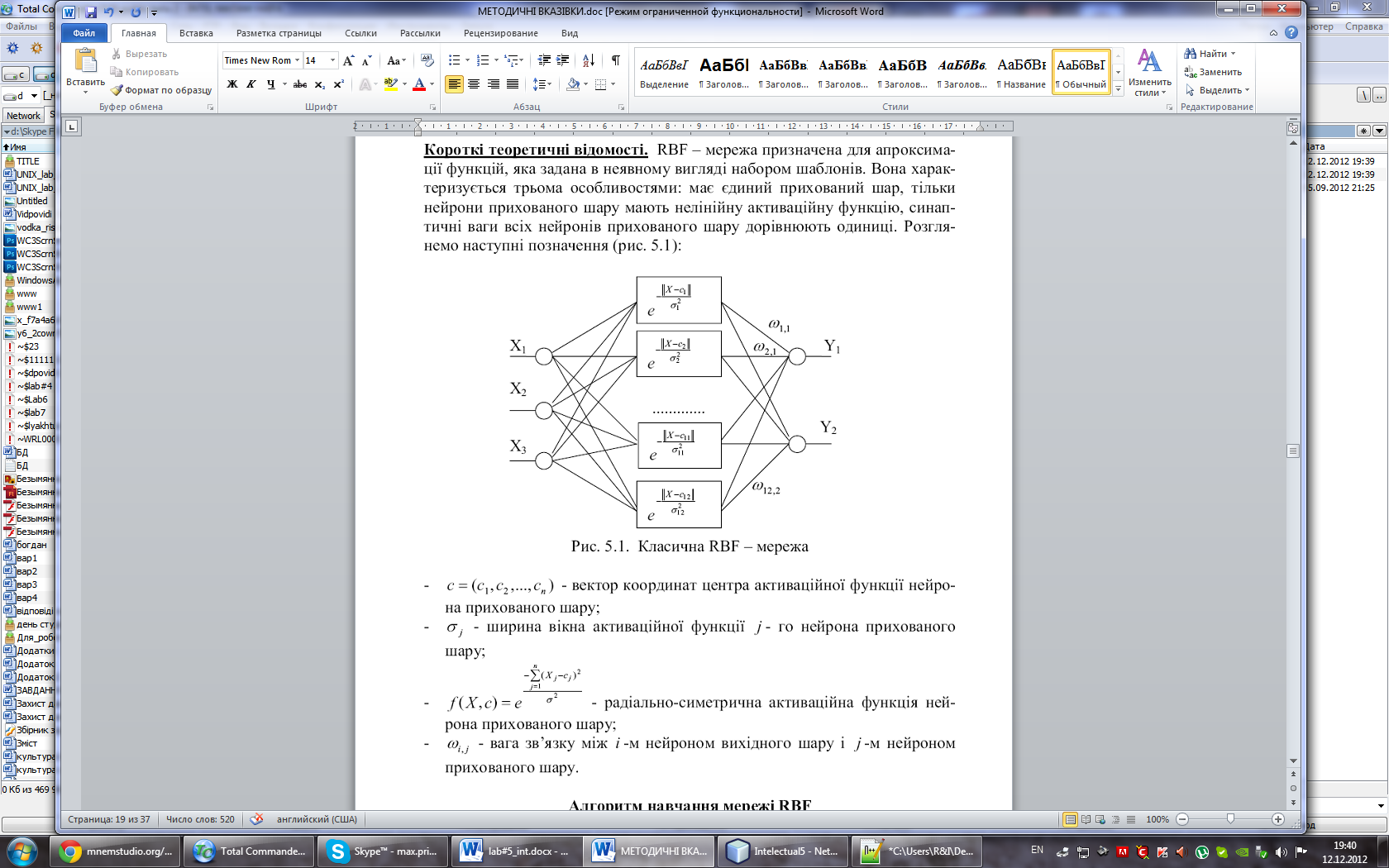
Дитко Т.В.

Івано-Франківськ

2015

**Мета:** вивчити принципи функціонування і навчання мережі типу RBF та програмно реалізувати ї навчання.

**Короткі теоретичні відомості**



**Завдання:** реалізувати алгоритм роботи мережі RBF.

**Хід роботи**

package com.ISAD;

import java.util.Arrays;

public class RBF extends javax.swing.JFrame {

private void initialize() {

pattern = new int[][]{{1, 1, 0, 0},

{0, 0, 0, 1},

{1, 0, 0, 0},

{0, 0, 1, 1},

{0, 1, 0, 0},

{0, 0, 1, 0},

{1, 0, 1, 0}};

private static final int n = 4;

private static final int m = 3;

private static final double MAX\_ERROR = 0.4;

private static final int PATTERNS = 7;

private static final int TRAIN\_COUNT = 4;

private int pattern[][] = null;

private double bw[][] = null;

private double tw[][] = null;

private int f1a[] = null;

private int f1b[] = null;

private double f2[] = null;

jTextArea1.append("Ваги:\t");

bw = new double[m][n];

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

bw[i][j] = 1.0 / (1.0 + n);

jTextArea1.append(bw[i][j] + ", ");

}

jTextArea1.append("\n");

}

jTextArea1.append("\n");

tw = new double[m][n];

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

tw[i][j] = 1.0;

jTextArea1.append(tw[i][j] + ", ");

}

jTextArea1.append("\n");

}

jTextArea1.append("\n");

f1a = new int[n];

f1b = new int[n];

f2 = new double[m];

}

private void RBF1() {

int inputSum = 0;

int activationSum = 0;

int f2Max = 0;

boolean reset = true;

jTextArea1.append("\nПочаток\n");

for (int vecNum = 0; vecNum < PATTERNS; vecNum++) {

jTextArea1.append("Вхід: " + vecNum + "\n");

for (int i = 0; i < m; i++)

f2[i] = 0.0;

for (int i = 0; i < n; i++)

f1a[i] = pattern[vecNum][i];

inputSum = vectorSum(f1a);

for (int i = 0; i < n; i++)

f1b[i] = f1a[i];

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++)

f2[i] += bw[i][j] \* (double) f1a[j];

jTextArea1.append("\n");

}

jTextArea1.append("\n");

reset = true;

while (reset) {

f2Max = maximum(f2);

for (int i = 0; i < n; i++)

f1b[i] = f1a[i] \* (int) Math.floor(tw[f2Max][i]);

activationSum = vectorSum(f1b);

jTextArea1.append("\nАктивація (x(i)) = " + activationSum + "\n");

reset = testForReset(activationSum, inputSum, f2Max);

}

if (vecNum < TRAIN\_COUNT)

updateWeights(activationSum, f2Max);

jTextArea1.append("\nВектор #" + vecNum + " належить до кластера #" + f2Max + "\n");

}

return;

}

private int vectorSum(int[] nodeArray) {

int sum = 0;

for (int i = 0; i < n; i++)

sum += nodeArray[i];

return sum;

}

private void updateWeights(int activationSum, int f2Max) {

for (int i = 0; i < n; i++)

bw[f2Max][i] = (2.0 \* (double) f1b[i]) / (1.0 + (double) activationSum);

for (int i = 0; i < m; i++)

for (int j = 0; j < n; j++)

bw[i][j] += Math.random() / 10;

for (int i = 0; i < n; i++)

tw[f2Max][i] = f1b[i];

for (int i = 0; i < m; i++)

for (int j = 0; j < n; j++)

tw[i][j] += Math.random() / 4;

return;

}

private boolean testForReset(int activationSum, int inputSum, int f2Max) {

if ((double) activationSum / (double) inputSum >= MAX\_ERROR) {

return false;

} else {

f2[f2Max] = -1.0;

return true;

}

}

private int maximum(double[] nodeArray) {

double max = nodeArray[0];

int max\_i = 0,i = 0;

for(; i < nodeArray.length; i++)

if(nodeArray[i] > max)

{

max = nodeArray[i];

max\_i = i;

}

return max\_i;

}

private void printResults() {

jTextArea1.append("\nФінальні ваги:\n");

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

jTextArea1.append(bw[i][j] + ", ");

}

jTextArea1.append("\n");

}

for (int i = 0; i < m; i++) {

for (int j = 0; j < n; j++) {

jTextArea1.append(tw[i][j] + ", ");

}

jTextArea1.append("\n");

}

}

public void go() {

initialize();

RBF1();

printResults();

}

public RBF() {

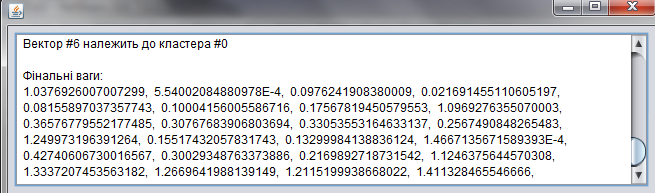
initComponents();

go();

}

public static void main(String args[]) { new RBF().setVisible(true);}

**Результати виконання програми**



***Висновок:***  в ході лабораторної роботи я ознайомився з принципом роботи мережі RBF і програмно реалізував алгоритм її навчання.