

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ ТА ПРОГРАМУВАННЯ»

Розрахункове завдання по курсу:
«ОСНОВИ НЕЙРОКОМП'ЮТИНГУ»

Виконав:
ст. гр. КІТ–М120Д
Надірян Г. О.

Перевірив:
Бречко В.О.

Харків – 2021

ЗМІСТ

Розрахункове завдання	3
Завдання №1	4
КОД ПРОГРАМИ ДЛЯ ЗАВДАННЯ №1	5
Завдання № 2 - 3.....	9
КОД ПРОГРАМИ ДЛЯ ЗАВДАННЯ № 2-3	9
Завдання №4.....	13
КОД ПРОГРАМИ ДЛЯ ЗАВДАННЯ №4	13
Завдання №5.....	15
Код програми завдання №5.....	15
Висновок	20

Розрахункове завдання

№ розділу	Розділи роботи	Обсяг самотійної роботи, годин	Строки (на якому тижні)
1	2	3	4
1	Розробити мережу Хебба, здатну розпізнавати п'ять букв Вашого прізвища або імені. При цьому обґрунтуйте вибір: числа рецепторних та вихідних нейронів, вибір векторів вихідних сигналів.	2	3
2	Розробити мережу Хеммінга, здатну розпізнавати п'ять букв Вашого прізвища або імені. При цьому обґрунтуйте вибір: числа рецепторних та вихідних нейронів, види функцій активації нейронів кожного шару, величини ваг зв'язків та зсувів у підмережі Хеммінга.	1	4
3	Навчити нейронну мережу Хеммінга еталонним зображенням букв. Підберіть вхідне зображення, що рівно віддалене по відстані Хеммінга від двох еталонних зображень. Яка реакція мережі при пред'явленні цього зображення? Запропонуйте спосіб більш інформативного поводження мережі при пред'явленні подібних зображень.	2	5
4	Розробити дискретну мережу Хопфілда, здатну розпізнавати п'ять букв Вашого прізвища або імені. При цьому обґрунтуйте вибір: числа нейронів мережі та спосіб кодування елементів чорно-білих зображень.	2	6
5	Розробити дискретну мережу Кохонена, здатну розпізнавати п'ять букв Вашого прізвища або імені. При цьому обґрунтуйте вибір: числа нейронів мережі, попередній вибір параметрів α , R та ваг зв'язків нейронної мережі для Вашого випадку.	2,5	11
6	Порівняти працездатність нейронних мереж по завадостійкості, складності алгоритму навчання.	3	12
9	Оформити звіт по розрахунковому завданню.	3,5	14

ЗАВДАННЯ №1

Розробити мережу Хебба, здатну розпізнавати п'ять букв Вашого прізвища або імені. При цьому обґрунтуйте вибір: числа рецепторних та вихідних нейронів, вибір векторів вихідних сигналів.

1	2	3
4	5	6
7	8	9

1	2	3
4	5	6
7	8	9

1	2	3
4	5	6
7	8	9

1	2	3
4	5	6
7	8	9

1	2	3
4	5	6
7	8	9

$M = \{ X^1 = (1, 1, 1, 1, -1, -1, 1, -1, -1, -1, -1),$
 $X^2 = (-1, 1, -1, 1, 1, 1, 1, -1, 1), -1, 1, -1, -1, -1),$
 $X^3 = (1, 1, 1, -1, 1, -1, -1, 1, -1), -1, -1, 1, -1, -1)$
 $X^4 = (1, -1, 1, 1, 1, 1, 1, -1, 1), -1, -1, -1, 1, -1),$
 $X^5 = (1, -1, 1, 1, 1, 1, 1, 1, 1), -1, -1, -1, -1, 1),$

};

КОД ПРОГРАМИ ДЛЯ ЗАВДАННЯ №1

```
public class Main {

    static int F=0;
    static int F2=0;
    static int W[][]=new int[25][9];
    static int s1[]=new int[5];
    static int s2[]=new int[5];
    static int s3[]=new int[5];
    static int s4[]=new int[5];
    static int s5[]=new int[5];

    static int search(int x[],int y){
        int S=0;
        if(F>4){
            for (int i = 0; i <x.length; i++) {
                W[F][i]=W[F-5][i]+x[i]*y;
                System.out.print(W[F][i]+" ");
            }
        }
        else {
            for (int i = 0; i <x.length; i++) {
                W[F][i]=+x[i]*y;
                System.out.print(W[F][i]+" ");
            }
        }
        for (int i = 0; i <x.length ; i++) {
            S=S+((W[F][i])*(x[i]));
        }
        System.out.println("S: "+S);
        F++;
        return S;
    }
    static int inputPattern(int x[],int y){
        int S=0;
        for (int i = 0; i <x.length; i++) {
            W[F2][i]=+x[i]*y;
            System.out.print(W[F2][i]+" ");
        }
        for (int i = 0; i <x.length ; i++) {
            S=S+W[F2][i]*x[i];
        }
        System.out.println("S: "+S);
        F2++;
        return S;
    }

    static void test(int input[]){
        int mark1=0;
        int mark2=0;
        int mark3=0;
        int mark4=0;
        int mark5=0;
        for (int i = 0; i < 5; i++) {
            if(s1[i]>0)s1[i]=1;
            else if(s1[i]<0)s1[i]=-1;
            if(s2[i]>0)s2[i]=1;
            else if(s2[i]<0)s2[i]=-1;
            if(s2[i]>0)s2[i]=1;
            else if(s2[i]<0)s2[i]=-1;
        }
    }
}
```

```

        if(s3[i]>0)s3[i]=1;
        else if(s3[i]<0)s3[i]=-1;
        if(s4[i]>0)s4[i]=1;
        else if(s4[i]<0)s4[i]=-1;
        if(s5[i]>0)s5[i]=1;
        else if(s5[i]<0)s5[i]=-1;
        if(input[i]>0)input[i]=1;
        else if(input[i]<0)input[i]=-1;
    }
    System.out.print("S1:");
    for (int i = 0; i < 5; i++) {
        System.out.print(s1[i]+" ");
    }
    System.out.print("\nS2:");
    for (int i = 0; i < 5; i++) {
        System.out.print(s2[i]+" ");
    }
    System.out.print("\nS3:");
    for (int i = 0; i < 5; i++) {
        System.out.print(s3[i]+" ");
    }
    System.out.print("\nS4:");
    for (int i = 0; i < 5; i++) {
        System.out.print(s4[i]+" ");
    }
    System.out.print("\nS5:");
    for (int i = 0; i < 5; i++) {
        System.out.print(s5[i]+" ");
    }
    System.out.print("\ninput:");
    for (int i = 0; i < 5; i++) {
        System.out.print(input[i]+" ");
    }

    for (int i = 0; i < 5 ; i++) {
        if (input[i]==s1[i])mark1++;
        if (input[i]==s2[i])mark2++;
        if (input[i]==s3[i])mark3++;
        if (input[i]==s4[i])mark4++;
        if (input[i]==s5[i])mark5++;
    }
    if (mark1>mark2&&mark1>mark3&&mark1>mark4&&mark1>mark5)
System.out.println("\nопределена буква Г");
    if
(mark2>mark1&&mark2>mark3&&mark2>mark4&&mark2>mark5)System.out.println("определ
ена буква А");
    if
(mark3>mark1&&mark3>mark2&&mark3>mark4&&mark3>mark5)System.out.println("определ
ена буква Т");
    if
(mark4>mark1&&mark4>mark2&&mark4>mark3&&mark4>mark5)System.out.println("определ
ена буква Н");
    if
(mark5>mark1&&mark5>mark2&&mark5>mark3&&mark5>mark4)System.out.println("определ
ена буква М");
}
public static void main(String[] args) {
    int x1[]={ 1, 1, 1,1, -1, -1,1,-1,-1};
    int x2[]={-1, 1, -1,1, 1, 1,1,-1,1};
    int x3[]={ 1, 1, 1,-1, 1,-1,-1, 1,-1};
    int x4[]={ 1,-1, 1,1, 1, 1,1,-1,1};
    int x5[]={ 1,-1, 1,1, 1, 1,1,1,1};
    int y1=1;
    int y2=-1;

```

```

int y3=-1;
int y4=-1;
int y5=-1;
    //T
    System.out.println("T");
    s1[0]= search(x1,y1);
    s1[1]=search(x1,y2);
    s1[2]= search(x1,y3);
    s1[3]= search(x1,y4);
    s1[4]= search(x1,y5);
    //A
    System.out.println("A");
    s2[0]=search(x2,y2);
    s2[1]=search(x2,y1);
    s2[2]=search(x2,y3);
    s2[3]=search(x2,y4);
    s2[4]=search(x2,y5);
    //T
    System.out.println("T");
    s3[0]=search(x3,y2);
    s3[1]=search(x3,y3);
    s3[2]=search(x3,y1);
    s3[3]=search(x3,y4);
    s3[4]=search(x3,y5);
    //H
    System.out.println("H");
    s4[0]=search(x4,y2);
    s4[1]=search(x4,y3);
    s4[2]=search(x4,y4);
    s4[3]=search(x4,y1);
    s4[4]=search(x4,y5);
    System.out.println("M");
    s5[0]=search(x5,y2);
    s5[1]=search(x5,y3);
    s5[2]=search(x5,y4);
    s5[3]=search(x5,y5);
    s5[4]=search(x5,y1);
    //T
    System.out.println("Rec");
    int test[]={1,-1, 1,1, 1, 1,1,1,1};
    int input[]=new int[5];
    input[0]= inputPattern(test,y2);
    input[1]=inputPattern(test,y3);
    input[2]= inputPattern(test,y4);
    input[3]= inputPattern(test,y5);
    input[4]= inputPattern(test,y1);
    test(input);
}
}

```

```

Г
1 1 1 1 -1 -1 1 -1 -1 S: 9
-1 -1 -1 -1 1 1 -1 1 1 S: -9
-1 -1 -1 -1 1 1 -1 1 1 S: -9
-1 -1 -1 -1 1 1 -1 1 1 S: -9
-1 -1 -1 -1 1 1 -1 1 1 S: -9
A
2 0 2 0 -2 -2 0 0 -2 S: -10
-2 0 -2 0 2 2 0 0 2 S: 10
0 -2 0 -2 0 0 -2 2 0 S: -8
0 -2 0 -2 0 0 -2 2 0 S: -8
0 -2 0 -2 0 0 -2 2 0 S: -8
T
1 -1 1 1 -3 -1 1 -1 -1 S: -3
-3 -1 -3 1 1 3 1 -1 3 S: -15
1 -1 1 -3 1 -1 -3 3 -1 S: 13
-1 -3 -1 -1 -1 1 -1 1 1 S: -5
-1 -3 -1 -1 -1 1 -1 1 1 S: -5
H
0 0 0 0 -4 -2 0 0 -2 S: -8
-4 0 -4 0 0 2 0 0 2 S: -4
0 0 0 -4 0 -2 -4 4 -2 S: -16
0 -4 0 0 0 2 0 0 2 S: 8
-2 -2 -2 -2 -2 0 -2 2 0 S: -10
M
-1 1 -1 -1 -5 -3 -1 -1 -3 S: -17
-5 1 -5 -1 -1 1 -1 -1 1 S: -13
-1 1 -1 -5 -1 -3 -5 3 -3 S: -17
-1 -3 -1 -1 -1 1 -1 -1 1 S: -1
-1 -3 -1 -1 -1 1 -1 3 1 S: 3
Rec
-1 1 -1 -1 -1 -1 -1 -1 -1 S: -9
-1 1 -1 -1 -1 -1 -1 -1 -1 S: -9
-1 1 -1 -1 -1 -1 -1 -1 -1 S: -9
-1 1 -1 -1 -1 -1 -1 -1 -1 S: -9
1 -1 1 1 1 1 1 1 1 S: 9
S1:1 -1 -1 -1 -1
S2:-1 1 -1 -1 -1
S3:-1 -1 1 -1 -1
S4:-1 -1 -1 1 -1
S5:-1 -1 -1 -1 1
input:-1 -1 -1 -1 1 определена буква M

```

Рисунок 1 – результат роботи завдання 1

Завдання № 2 - 3

Розробити мережу Хеммінга, здатну розпізнавати п'ять букв Вашого прізвища або імені. При цьому обґрунтуйте вибір: числа рецепторних та вихідних нейронів, види функцій активації нейронів кожного шару, величини ваг зв'язків та зсувів у підмережі Хеммінга. Навчити нейронну мережу Хеммінга еталонним зображенням букв Підберіть вхідне зображення, що рівно віддалене по відстані Хеммінга від двох еталонних зображень. Яка реакція мережі при пред'явленні цього зображення? Запропонуйте спосіб більш інформативного поведження мережі при пред'явленні подібних зображень.

КОД ПРОГРАМИ ДЛЯ ЗАВДАННЯ № 2-3

```
package lab3;

public class Main {
    static String recLetter="";
    static double m = 9;
    static double bi=m/2;
    static double n=5;
    static double exelant=0.2;
    static double v1[]={
        1,-1,1,
        1,1,1,
        1,-1,1};
    static double v2[]={
        -1,1,-1,
        1,1,1,
        1,-1,1};
    static double v3[]={
        1,-1,1,
        -1,1,-1,
        -1,1,-1};
    static double v4[]={
        -1,1,-1,
        -1,1,-1,
        -1,1,-1};
    static double v5[]={
        1,1,1,
        1,-1,1,
        1,1,1};
    static double v6[]={
        1,1,1,
        -1,1,-1,
        -1,1,-1};
    static double s1[]={
        -1,-1,1,
        1,1,1,
        1,-1,1};
    static double s2[]={
        -1,-1,1,
        -1,1,-1,
        -1,-1,-1};
    static double s3[]={
        -1,-1,-1,
```

```

        -1,1,-1,
        -1,1,-1};

static double wik[][]=new double[9][6];
static double wik2[][]=new double[9][6];

static void half(){
    for (int i = 0; i <9; i++) {
        wik[i][0]=v1[i]/2;
        wik[i][1]=v2[i]/2;
        wik[i][2]=v3[i]/2;
        wik[i][3]=v4[i]/2;
        wik[i][4]=v5[i]/2;
        wik[i][5]=v6[i]/2;
        wik2[i][0]=s1[i]/2;
        wik2[i][1]=s2[i]/2;
        wik2[i][2]=s3[i]/2;
    }
}

static double uIn[]=new double[6];
static double uZ[]=new double[6];
static double uA[]=new double[6];

static void calcInputSignal(){
    for (int i = 0; i <9 ; i++) {
        uIn[0]=uIn[0]+(wik[i][0]*s1[i]);
        uIn[1]=uIn[1]+(wik[i][1]*s1[i]);
        uIn[2]=uIn[2]+(wik[i][2]*s1[i]);
        uIn[3]=uIn[3]+(wik[i][3]*s1[i]);
        uIn[4]=uIn[4]+(wik[i][4]*s1[i]);
        uIn[5]=uIn[5]+(wik[i][5]*s1[i]);
    }
    uIn[0]+=bi;
    uIn[1]+=bi;
    uIn[2]+=bi;
    uIn[3]+=bi;
    uIn[4]+=bi;
    uIn[5]+=bi;
    System.out.println("~Uin~");
    System.out.println(+uIn[0]);
    System.out.println(+uIn[1]);
    System.out.println(+uIn[2]);
    System.out.println(+uIn[3]);
    System.out.println(+uIn[4]);
    System.out.println(+uIn[5]);
    System.out.println("~~~~~Rusult~~~~~");

    double k = 0.1;
    int mark=0;
    for (int j = 0; j < 6; j++)
        uZ[j] = uIn[j] * k;
    while (true) {

        uA[0] = uZ[0] - execelant * (uZ[1] + uZ[2] + uZ[3] + uZ[4]+uZ[5]);
        uA[1] = uZ[1] - execelant * (uZ[0] + uZ[2] + uZ[3] + uZ[4]+uZ[5]);
        uA[2] = uZ[2] - execelant * (uZ[0] + uZ[1] + uZ[3] + uZ[4]+uZ[5]);
        uA[3] = uZ[3] - execelant * (uZ[0] + uZ[1] + uZ[2] + uZ[4]+uZ[5]);
        uA[4] = uZ[4] - execelant * (uZ[0] + uZ[1] + uZ[2] + uZ[3]+uZ[5]);
        uA[5] = uZ[5] - execelant * (uZ[0] + uZ[1] + uZ[2] + uZ[3]+uZ[4]);

        if (uA[0] < 0) uA[0] = 0;
        if (uA[1] < 0) uA[1] = 0;
        if (uA[2] < 0) uA[2] = 0;

```

```

        if (uA[3] < 0) uA[3] = 0;
        if (uA[4] < 0) uA[4] = 0;
        if (uA[5] < 0) uA[5] = 0;
        if (uA[0] > 0) mark++;
        if (uA[1] > 0) mark++;
        if (uA[2] > 0) mark++;
        if (uA[3] > 0) mark++;
        if (uA[4] > 0) mark++;
        if (uA[5] > 0) mark++;
        for (int i = 0; i < 6; i++) {
            System.out.printf("%.4f ", uA[i]);
        }
        if (mark == 1) {
            System.out.println();
            for (int i = 0; i < 6; i++) {
                if (uA[i] > 0) {
                    if (i == 0) recLetter += " H ";
                    if (i == 1) recLetter += " A ";
                    if (i == 2) recLetter += " Y ";
                    if (i == 3) recLetter += " I ";
                    if (i == 4) recLetter += " R ";
                    if (i == 5) recLetter += " T ";
                    System.out.println("Распознано буквы: "+recLetter);
                }
            }
            break;
        }
        for (int i = 0; i < 6; i++) {
            uZ[i] = uA[i];
        }
        System.out.println();
        mark = 0;
    }

}

public static void main(String[] args) {
    half();
    calcInputSignal();
    System.out.println();
}
}

```

```
~Uin~
8.0
7.0
3.0
2.0
5.0
2.0
~~~~~Result~~~~~
0,4200 0,3000 0,0000 0,0000 0,0600 0,0000
0,3480 0,2040 0,0000 0,0000 0,0000 0,0000
0,3072 0,1344 0,0000 0,0000 0,0000 0,0000
0,2803 0,0730 0,0000 0,0000 0,0000 0,0000
0,2657 0,0169 0,0000 0,0000 0,0000 0,0000
0,2623 0,0000 0,0000 0,0000 0,0000 0,0000
Распознано букву:  Н
```

Рисунок 2 – результат роботи програми завдання №2-3

Завдання №4

Розробити дискретну мережу Хопфілда, здатну розпізнавати п'ять букв Вашого прізвища або імені. При цьому обґрунтуйте вибір: числа нейронів мережі та спосіб кодування елементів чорно-білих зображень.

КОД ПРОГРАМИ ДЛЯ ЗАВДАННЯ №4

```
package lab4;
import java.util.Arrays;
public class RGZ {
    static int sign(int x){return x>0?1:-1;}
    static void rec(int x1[],int x2[],int x3[],int x4[],int x5[],int rec[]){
        int mark[]=new int[5];
        for (int i = 0; i <9 ; i++) {
            if (x1[i]==rec[i])mark[0]++;
            if (x2[i]==rec[i])mark[1]++;
            if (x3[i]==rec[i])mark[2]++;
            if (x4[i]==rec[i])mark[3]++;
            if (x5[i]==rec[i])mark[4]++;
        }
        System.out.print("Распознана буква: ");
        if (mark[0]>mark[1]&&mark[0]>mark[2]&&mark[0]>mark[3]&&mark[0]>mark[4])
System.out.println("H");
        else if
(mark[1]>mark[0]&&mark[1]>mark[2]&&mark[1]>mark[3]&&mark[1]>mark[4])
System.out.println("T");
        else if
(mark[2]>mark[1]&&mark[2]>mark[0]&&mark[2]>mark[3]&&mark[2]>mark[4])
System.out.println("I");
        else if
(mark[3]>mark[1]&&mark[3]>mark[0]&&mark[3]>mark[2]&&mark[3]>mark[4])
System.out.println("A");
        else if
(mark[4]>mark[1]&&mark[4]>mark[0]&&mark[4]>mark[2]&&mark[4]>mark[3])
System.out.println("J");

        else System.out.println("Не распознано");
    }
    public static void main(String[] args) {
        int w[][]=new int[9][9];
        int x1[]={1,-1,1, 1,1,1, 1,-1,1};
        int x5[]={1,1,1, 1,1,1, 1,-1,1};
        int x6[]={1,-1,1, 1,-1,1, 1,-1,1};
        int x2[]={1,1,1, -1,1,-1, -1,1,-1};
        int x4[]={-1,1,1,
                -1,1,-1,
                -1,1,-1};
        int x3[]={-1,-1,-1,
                1, 1,-1,
                -1, 1,-1};

        System.out.println("Буква H: "+Arrays.toString(x1));
        System.out.println("Буква T: "+Arrays.toString(x2));
        System.out.println("Буква I: "+Arrays.toString(x4));
        System.out.println("Буква A: "+Arrays.toString(x5));
        System.out.println("Буква J: "+Arrays.toString(x6));
        System.out.println("Буква для распознавание: "+Arrays.toString(x3));
```

```

        for (int i = 0; i < 9; i++) {
            for (int j = 0; j < 9; j++) {
                w[i][j] = x1[i] * x1[j] + x2[i] * x2[j] + x4[i] * x4[j] + x5[i] * x5[j] + x6[i] * x6[j];
            }
        }
        for (int i = 0; i < 9; i++) {
            w[i][i] = 0;
        }
        System.out.println("\t\t\t\t\tМатриця ваг зв'язків");

        for (int i = 0; i < 9; i++) {
            for (int j = 0; j < 9; j++) {
                System.out.print(w[i][j] + "\t\t");
            }
            System.out.println("");
        }
        int y[] = new int[9];
        for (int j = 1; j < 3; j++) // итерации
        {
            for (int i = 0; i < 9; i++)
            {
                y[i] = 0;
                for (int k = 0; k < 9; k++) y[i] += w[i][k] * x3[k];
            }
            for (int i = 0; i < 9; i++) x3[i] = sign(y[i]); // обратная связь
        }
        System.out.println("Проверка. Обратная связь для [x3]: " +
Arrays.toString(x3));
        for (int i = 0; i < 9; i++)
        {
            System.out.print(y[i] + " ");
        }
        System.out.println("\nПроверка. Обратная связь для [x3]: " +
Arrays.toString(x3));

        rec(x1, x2, x4, x5, x6, x3);
    }

}

```

```

C:\Users\hamlet\.jdk\openjdk-15.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
Буква Н: [1, -1, 1, 1, 1, 1, 1, -1, 1]
Буква Т: [1, 1, 1, -1, 1, -1, -1, 1, -1]
Буква Г: [-1, 1, 1, -1, 1, -1, -1, 1, -1]
Буква А: [1, 1, 1, 1, 1, 1, 1, -1, 1]
Буква Л: [1, -1, 1, 1, -1, 1, 1, -1, 1]
Буква для распознавание: [-1, -1, -1, 1, 1, -1, -1, 1, -1]
Матриця ваг зв'язків
0      -1      3      3      1      3      3      -3      3
-1      0      1     -3      3     -3     -3      3     -3
3       1      0      1      3      1      1     -1      1
3     -3      1      0     -1      5      5     -5      5
1       3      3     -1      0     -1     -1      1     -1
3     -3      1      5     -1      0      5     -5      5
3     -3      1      5     -1      5      0     -5      5
-3      3     -1     -5      1     -5     -5      0     -5
3     -3      1      5     -1      5      5     -5      0
Проверка. Обратная связь для [x3]: [-1, 1, -1, -1, 1, -1, -1, 1, -1]
-20 12 -10 -26 4 -26 -26 26 -26
Проверка. Обратная связь для [x3]: [-1, 1, -1, -1, 1, -1, -1, 1, -1]
Распознана буква: Г
Process finished with exit code 0

```

Рисунок 3 – работа программы завдання №4

Завдання №5

Розробити дискретну мережу Кохонена, здатну розпізнавати п'ять букв Вашого прізвища або імені. При цьому обґрунтуйте вибір: числа нейронів мережі, попередній вибір параметрів α , R та ваг зв'язків нейронної мережі для Вашого випадку

Код програми завдання №5

```
package example1;

import java.util.ArrayList;
import java.util.Arrays;

public class Main {

    //static int vector[][]={{0,0,0,1},{0,0,1,1},{1,0,0,0},{1,1,0,0}};
    static String[] letter={"Г","А","Н","Р","Т"};
    static int vector[][]={

        {1,1,1,1,1,
         1,0,0,0,0,
         1,0,0,0,0,
         1,0,0,0,0,
         1,0,0,0,0,
         1,0,0,0,0,
         1,0,0,0,0},
        },
        {0,1,1,1,0,
         1,0,0,0,1,
         1,1,1,1,1,
         1,0,0,0,1,
         1,0,0,0,1,
         1,0,0,0,1,
         1,0,0,0,1},
        },
        {1,0,0,0,1,
         1,0,0,0,1,
         1,1,1,1,1,
         1,0,0,0,1,
         1,0,0,0,1,
         1,0,0,0,1},
        },
        {1,1,1,1,1,
         1,0,0,0,1,
         1,1,1,1,1,
         1,0,0,0,0,
         1,0,0,0,0,
         1,0,0,0,1},
        },
        {1,1,1,1,1,
         0,0,1,0,0,
         0,0,1,0,0,
         0,0,1,0,0,
         0,0,1,0,0,
         0,0,1,0,0,
         0,0,1,0,0},
        },
        {
            1,1,1,1,1,
            0,0,1,0,0,
            0,0,1,0,0,
            0,0,1,0,0,
```

```

        0,0,1,0,0,
        0,1,1,1,0
    },

};

static int neuron = 4;
static int m = neuron/2;
static int A = 4;
static double k = 0.25;
static double alfa0 = 0.95;
static double mq[][]={
    {0.7,0.6,0.5,0.4,0.3,0.3},
    {0.4,0.3,0.2,0.15,0.1,0.1},
    {0.5,0.5,0.5,0.5,0.5,0.5},
    {0.2,0.3,0.4,0.5,0.9,0.9},
    {0.4,0.32,0.31,0.3,0.2,0.2},
    {0,3,0.2,0.1,0.1,0.5,0.5},
};

static double destination[][]=new double[10000][6];
static int index_i=0;
static int index_j=0;

static ArrayList<double[][]> arrayList = new ArrayList<double[][]>();

static double R = 0;
static int res=0;
static int res2=0;

static double[][] run(){
    double temp1[][]=new double[36][6];
    double temp2[][]=new double[36][6];
    int n1=0;
    int n2=1;
    int n3=0;

    double alf = 0.40;
    double alf2 = 0.6;

    int one=1;
    double res1=0;
    int counter = 0;
    boolean array=false;
    boolean array2=false;
    alf2=1;
    for (int i = 1; i <1000 ; i++) {
        alf2=alf2*0.6;
        alf=one-alf2;
        n3=0;
        extracted(n1, n2,n3,alf,alf2);
        n3=1;
        extracted(n1, n2,n3,alf,alf2);
        n3=2;
        extracted(n1, n2,n3,alf,alf2);
        n3=3;
        extracted(n1, n2,n3,alf,alf2);
        for ( index_i= 0; index_i < 6; index_i++) {
            for ( index_j = 0; index_j <6 ; index_j++) {
                destination[res][index_j]=mq[index_i][index_j];
            }
            res++;
        }

        if(i%2==0){

```



```

        for ( index_i= 0; index_i < 6; index_i++) {
            for ( index_j = 0; index_j <6 ; index_j++) {
                temp1[index_i][index_j]=mq[index_i][index_j];
            }
            // res++;
        }
        array=true;
    }
    else {
        for ( index_i= 0; index_i < 6; index_i++) {
            for ( index_j = 0; index_j <6 ; index_j++) {
                temp2[index_i][index_j]=mq[index_i][index_j];
            }
            // res2++;
        }
        array2=true;
    }
    if (array&&array2){
        for ( index_i= 0; index_i < 6; index_i++) {
            for ( index_j = 0; index_j <6 ; index_j++) {
                res1 = temp2[index_i][index_j]-temp1[index_i][index_j];
                if (res1<=0.0005) counter++;
            }
        }

    }
    if (counter==36) {
        System.out.println("Количество итерации:"+i);

        /*
        for ( index_i= 0; index_i < 5; index_i++) {
            for ( index_j = 0; index_j <5 ; index_j++) {
                System.out.printf("%5f ",temp1[index_i][index_j]);
            }
            System.out.println();
        }*/
        System.out.println();
        for ( index_i= 0; index_i < 6; index_i++) {
            for ( index_j = 0; index_j <6 ; index_j++) {
                if (temp2[index_i][index_j]>=0.5)
                    //
temp2[index_i][index_j]=Math.ceil(temp2[index_i][index_j]);
                temp2[index_i][index_j]=1.0;
                else if(temp2[index_i][index_j]<0.5)
temp2[index_i][index_j]=Math.floor(temp2[index_i][index_j]);

                System.out.printf("%5f ",temp2[index_i][index_j]);
            }
            System.out.println();
        }
        break;
    }
    counter=0;
}

return temp2;

}

private static void extracted(int n1, int n2,int n3,double alf1,double
alf2) {
    double da1=0;
    double da2=0;

```

```

        for (int i = 0; i < 6 ; i++) {
            da1=da1+Math.pow((mq[i][n1]-vector[n3][i]),2);
            da2=da2+Math.pow((mq[i][n2]-vector[n3][i]),2);
        }
        if(da1>da2){

            for (int i = 0; i < 6 ; i++) {
                mq[i][n2]=alf1*mq[i][n2]+alf2*vector[n3][i];

            }
        }
        if(da1<da2){
            for (int i = 0; i < 6 ; i++) {
                mq[i][n1]=alf1*mq[i][n1]+alf2*vector[n3][i];
            }
        }
    }
    static void copyTo(){
        double output [] [] = run();
        double result[][]=new double[6][5];
        double input[]=new double[6];
        for (int i = 0; i < 6; i++) {
            for (int j = 0; j < 5 ; j++) {
                result[i][j]=output[i][j];
            }
        }
        for (int i = 0; i < 6 ; i++) {
            input[i]=output[i][5];
        }
        System.out.println();
        check(result,input);
    }

    private static void check(double[][] result, double[] input) {
        int cluster[] = new int[5];
        int clusterSum[] = new int[5];
        /*
            for (int i = 0; i < 6; i++) {
                for (int j = 0; j < 5 ; j++) {
                    if (result[i][j]==1){
                        cluster[j]+=1+i/2;
                    }
                }
            }
        */
        System.out.println();

        for (int i = 0; i < 5; i++) {
            cluster[i] = i;
            System.out.print("Cluster №" + cluster[i] + "-" + letter[i]);
            System.out.println();
        }
        int max[] = new int[5];
        /*
            for (int i = 0; i < 6; i++) {
                if(result[i][0]==input[i]){
                    max[0]+=1;
                }
            }
        */

        for (int i = 0; i < 6; i++) {
            for (int j = 0; j < 5 ; j++) {
                if (result[i][j] == input[i]) {
                    max[j] += 1;
                }
            }
        }
    }

```

```

    }
}
System.out.println("Номер кластер входной буквы = " +
maxArrayIndex(max, 0, max.length-1));

System.out.println();
}
static int maxArrayIndex(int[] X, int p, int r) {
    int currentMaxIndex = 0;
    for (int i = 0; i < X.length; i++) {
        if(X[i] > X[currentMaxIndex]){
            currentMaxIndex = i;
        }
    }
    return currentMaxIndex;
}
public static void main(String[] args) {
    copyTo();
}
}

```

```

Количество итерации:9

1,000000 1,000000 1,000000 1,000000 0,000000 0,000000
1,000000 0,000000 0,000000 0,000000 0,000000 0,000000
1,000000 1,000000 1,000000 1,000000 1,000000 1,000000
1,000000 0,000000 0,000000 1,000000 1,000000 1,000000
1,000000 0,000000 0,000000 0,000000 0,000000 0,000000
1,000000 1,000000 0,000000 0,000000 0,000000 1,000000

Cluster №0-Г
Cluster №1-A
Cluster №2-Н
Cluster №3-Р
Cluster №4-Т
Номер кластер входной буквы = 4

```

Рисунок 4 – результат работы программы №5

Висновок

Розроблено мережу Хебба, здатну розпізнавати п'ять букв прізвища. При цьому обґрунтовано вибір: числа рецепторних та вихідних нейронів, вибір векторів вихідних сигналів.

Розроблено мережу Хеммінга, здатну розпізнавати п'ять букв Вашого прізвища. При цьому обґрунтовано вибір: числа рецепторних та вихідних нейронів, види функцій активації нейронів кожного шару, величини ваг зв'язків та зсувів у подмережі Хеммінга.

Розроблено дискретну мережу Хопфілда, здатну розпізнавати п'ять букв імені. При цьому обґрунтовано вибір: числа нейронів мережі та спосіб кодування елементів чорно-білих зображень.

Розроблено дискретну мережу Кохонена, здатну розпізнавати п'ять букв імені. При цьому обґрунтовано вибір: числа нейронів мережі, попередній вибір параметрів R та ваг зв'язків нейронної мережі.