Сторожук О.Д.

КН-Н921б

Лабораторна робота №1

Тема роботи: «Нейронна мережа Хебба»

Мета роботи: отримання й закріплення знань, формування практичних навичок роботи із найпростішими нейронними мережами, для навчання яких використовується алгоритм Хебба.

ЗАВДАННЯ ДО ВИКОНАННЯ

Розробити структуру мережі Хебба, яка здатна розпізнавати чотири букви А, Л, Е, К. При цьому:

- число нейронних рецепторів складає 25, для забезпечення оптимальної деталізації та можливості для розпізнавання вхідного символу;

- число вихідних нейронів 4, що дорівнює кількості аналізованих символів.

- в якості векторів вихідних сигналів буде послідовність з 1 та -1, в якій значення -1 буде знаходитися за тим номером порядку, який має даний символ.

ХІД РОБОТИ

Нижче наведений код виконуваної програми:

public class Main {

public static void main(String[] args) {

List<Image> ethalonImages = initEthalonImages();

System.out.println(ethalonImages);

Hebbian hebbian = new Hebbian();

hebbian.teachNetwork(ethalonImages);

Image testImage = new Image();

testImage.setBits(new int[]{-1, -1, 1, -1, -1,

-1, 1, -1, 1, -1,

1, 1, 1, 1, 1,

1, -1, -1, -1, 1,

1, -1, -1, 1, 1

});

testImage(ethalonImages, testImage, hebbian);

testImage.setBits(new int[]{1, -1, -1, 1, 1,

1, -1, 1, -1, -1,

1, 1, -1, -1, -1,

1, -1, 1, -1, -1,

1, 1, -1, 1, -1

});

testImage(ethalonImages, testImage, hebbian);

testImage.setBits(new int[]{

1, 1, 1, 1, 1,

1, -1, -1, 1, -1,

1, 1, 1, 1, 1,

1, 1, -1, -1, -1,

1, 1, 1, 1, 1

});

testImage(ethalonImages, testImage, hebbian);

testImage.setBits(new int[]{

-1, 1, 1, 1, -1,

-1, 1, -1, 1, 1,

-1, 1, -1, 1, -1,

-1, 1, -1, 1, -1,

1, 1, -1, 1, 1

});

testImage(ethalonImages, testImage, hebbian);

testImage.setBits(new int[]{

-1, 1, 1, 1, 1,

1, 1, -1, -1, 1,

-1, 1, -1, 1, -1,

-1, -1, -1, 1, -1,

1, 1, -1, 1, 1

});

testImage(ethalonImages, testImage, hebbian);

}

private static void testImage(List<Image> ethalons, Image unknown, Hebbian hebbian) {

int bits[] = unknown.getBits();

StringBuilder text = new StringBuilder();

for (int i = 0; i < 25; i++) {

text.append(String.format("%3s", bits[i]));

if ((i + 1) % 5 == 0) {

text.append("\n");

}

}

System.out.println("Trying to recognize image with matrix:\n" + text.toString());

int[] vector = hebbian.predict(unknown);

recognizeImageByVector(ethalons, vector);

}

private static void recognizeImageByVector(List<Image> ethalons, int[] vector) {

for (Image image : ethalons) {

if (Arrays.equals(image.getVector(), vector)) {

System.out.println("The current image is probably symbol: " + image.getSymbol());

return;

}

}

System.out.println("Can't recognize symbol");

}

//А, Л, Е, К (5х5 матрица)

private static List<Image> initEthalonImages() {

List<Image> ethalons = new ArrayList<>(4);

ethalons.add(new Image(new int[]{-1, -1, 1, -1, -1,

-1, 1, -1, 1, -1,

1, 1, 1, 1, 1,

1, -1, -1, -1, 1,

1, -1, -1, -1, 1

}, new int[]{-1, 1, 1, 1}, 'А'));

ethalons.add(new Image(new int[]{-1, 1, 1, 1, -1,

-1, 1, -1, 1, -1,

-1, 1, -1, 1, -1,

-1, 1, -1, 1, -1,

1, 1, -1, 1, -1

}, new int[]{1, -1, 1, 1}, 'Л'));

ethalons.add(new Image(new int[]{1, 1, 1, 1, 1,

1, -1, -1, -1, -1,

1, 1, 1, 1, 1,

1, -1, -1, -1, -1,

1, 1, 1, 1, 1

}, new int[]{1, 1, -1, 1}, 'Е'));

ethalons.add(new Image(new int[]{1, -1, -1, 1, -1,

1, -1, 1, -1, -1,

1, 1, -1, -1, -1,

1, -1, 1, -1, -1,

1, -1, -1, 1, -1

}, new int[]{1, 1, 1, -1}, 'К'));

return ethalons;

}

}

public class Hebbian {

private int[][] weights;

public Hebbian() {

weights = new int[4][25];

}

public void teachNetwork(List<Image> ethalonImages) {

boolean isDone = false;

int[] bits;

int[] vector;

while (!isDone) {

for (Image image : ethalonImages) {

isDone = isImageComplete(image);

if (!isDone) {

bits = image.getBits();

vector = image.getVector();

for (int j = 0; j < 4; j++) {

for (int i = 0; i < 25; i++) {

weights[j][i] += bits[i] \* vector[j];

}

}

}

}

isDone = networkTeachCompleted(ethalonImages);

}

}

private boolean networkTeachCompleted(List<Image> ethalonImages) {

for (Image image : ethalonImages) {

if (!isImageComplete(image)) {

return false;

}

}

return true;

}

public int[] predict(Image image) {

int[] vector = new int[4];

int index = 0;

int[] sum = new int[25];

int[] binaryImage = image.getBits();

for (int sym = 0; sym < 4; sym++) {

for (int i = 0; i < 25; i++) {

sum[sym] += binaryImage[i] \* weights[sym][i];

}

if (sum[sym] > 0) {

vector[index++] = 1;

} else {

vector[index++] = -1;

}

}

return vector;

}

private boolean isImageComplete(Image image) {

return Arrays.equals(image.getVector(), predict(image));

}

public void setWeights(int[][] weights) {

this.weights = weights;

}

public int[][] getWeights() {

return weights;

}

}

На рис. 1 продемонстрований етап навчання системи на еталонних зразках символів. При цьому заповнюється вагова матриця до кожного біту з матриці символів.

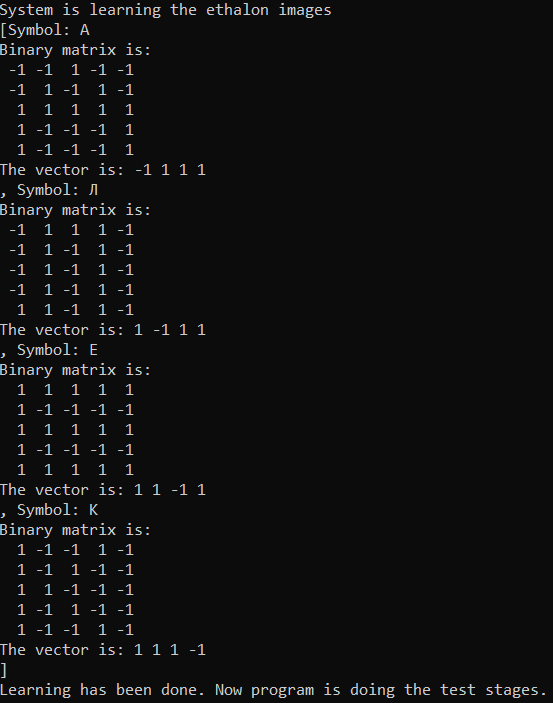


Рисунок 1 – Навчання системи на еталонних символах

Далі у програмі занесені спеціальні невизначені символи, які мають лише бітову маску, а інші її атрибути (вектор та сам символ) не записані. Задача нейронної мережі розпізнати ці символи. Процес розпізнання наведений на рис. 2.

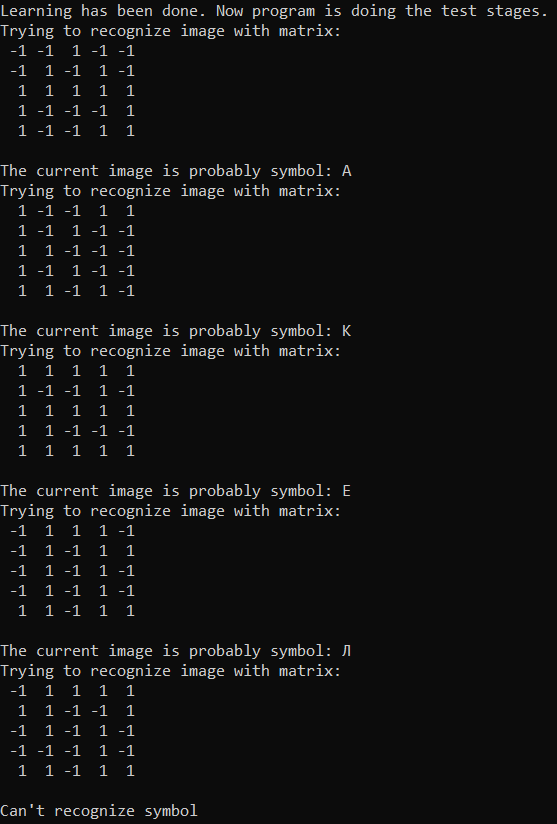


Рисунок 2 – Процес розпізнавання символів. Як видно з прикладу, програма не змогла розпізнати лише 1 останній символ, про сама програма і повідомляє користувача

Висновки: під час виконання лабораторної роботи були отримані й закріплені знання, сформовані практичні навички роботи із найпростішими нейронними мережами, для навчання яких використовується алгоритм Хебба.