Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №5

за 3 семестр

По дисциплине: «МиАПР»

Тема: «Нелинейные ИНС в задачах распознавания образов»

Выполнила:

Студент 2 курса

Группы ПО-4(2)

Луд А.С.

Проверил:

Крощенко А.А.

Брест 2020

Лабораторная работа №5

Нелинейные ИНС в задачах распознавания образов

Цель работы: изучить обучение и функционирование нелинейной ИНС при решении задач распознавания образов.

Вариант 4

Задание:

Написать на любом ЯВУ программу моделирования нелинейной ИНС для распознавания образов. Рекомендуется использовать сигмоидную функцию, но это не является обязательным. Количество НЭ в скрытом слое взять согласно варианту работы №3. Его можно варьировать, если сеть не обучается или некорректно функционирует.

Провести исследование полученной модели. При этом на вход сети необходимо подавать искаженные образы, в которых инвертированы некоторые биты. Критерий эффективности процесса распознавания - максимальное кодовое расстояние (количество искаженных битов) между исходным и поданным образом.

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Вектор 1 | Вектор 2 | Вектор 3 |
| 4 | 4 | 3 | 8 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| № | Данные вектора | | | | | | | | | | | | | | | | | | | |
| 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 3 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| 5 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| 6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 8 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |

Код программы:

#include <iostream>

using namespace std;

double Sigmoid(double x) {

return 1 / (1 + pow(2.7, -x));

}

double\* Hiddens(bool\* inputs, double w12[20][40], double T\_hid[]) {

double\* hiddens = new double[40];

for (int i = 0; i < 40; i++) hiddens[i] = 0;

for (int i = 0; i < 40; i++) {

for (int j = 0; j < 20; j++) {

hiddens[i] += w12[j][i] \* inputs[j];

}

hiddens[i] -= T\_hid[i];

hiddens[i] = Sigmoid(hiddens[i]);

}

return hiddens;

}

double\* Result(bool\* Inputs, double w12[20][40], double T\_hid[], double w23[40][3], double T\_out[], double hiddens[40]) {

double\* result = new double[3];

for (int i = 0; i < 3; i++)

result[i] = 0;

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

for (int i = 0; i < 40; i++) {

result[j] += hiddens[i] \* w23[i][j];

}

result[j] -= T\_out[j];

result[j] = Sigmoid(result[j]);

}

return result;

}

int main() {

setlocale(LC\_ALL, "rus");

int era = 0;

bool Vector1[] = { 1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1 }; //3

bool Vector2[] = { 1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0 }; //4

bool Vector3[] = { 1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1 }; //8

bool\* inputs = new bool[20];

for (int i = 0; i < 20; i++) inputs[i] = 0;

bool\*\* vectors = new bool\* [12];

vectors[0] = Vector1;

vectors[1] = Vector2;

vectors[2] = Vector3;

double w12[20][40], w23[40][3], T\_hid[40], T\_out[3], E\_min = 0.001, V = 0.04, Emax = 0, outputs[3] = { 0 };

double\* currents = new double[3];

double\* hiddens = new double[40];

double errors[3] = { 0 };

double ethalons[3] = { 0 };

double error\_hid[40] = { 0 };

for (int i = 0; i < 20; i++) {

for (int j = 0; j < 40; j++) {

w12[i][j] = ((double)rand() / (RAND\_MAX)) - 0.5;

for (int k = 0; k < 3; k++) {

w23[j][k] = ((double)rand() / (RAND\_MAX)) - 0.5;

T\_out[k] = ((double)rand() / (RAND\_MAX)) - 0.5;

}

T\_hid[j] = ((double)rand() / (RAND\_MAX)) - 0.5;

}

}

do {

Emax = 0;

for (int N = 0; N < 3; N++) {

ethalons[0] = 0;

ethalons[N] = 1;

inputs = vectors[N];

hiddens = Hiddens(inputs, w12, T\_hid);

currents = Result(inputs, w12, T\_hid, w23, T\_out, hiddens);

for (int i = 0; i < 3; i++)

errors[i] = currents[i] - ethalons[i];

for (int j = 0; j < 40; j++) {

for (int m = 0; m < 3; m++) {

error\_hid[j] += errors[m] \* currents[m] \* (1 - currents[m]) \* w23[j][m];

}

}

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

for (int i = 0; i < 40; i++) {

w23[i][j] -= V \* errors[j] \* currents[j] \* (1 - currents[j]) \* hiddens[i];

}

T\_out[j] += V \* errors[j] \* currents[j] \* (1 - currents[j]);

}

for (int j = 0; j < 40; j++) {

for (int i = 0; i < 20; i++) {

w12[i][j] -= V \* error\_hid[j] \* hiddens[j] \* (1 - hiddens[j]) \* inputs[i];

}

T\_hid[j] += V \* error\_hid[j] \* hiddens[j] \* (1 - hiddens[j]);

}

Emax += pow(errors[N], 2);

}

Emax /= 2;

era++;

} while (Emax > E\_min);

cout << "Количество эпох = " << era << endl;

cout << endl;

double\* hidden\_pred;

double\* values;

bool Vector11[] = { 1,0,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1 }; //3

bool Vector12[] = { 1,1,0,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1 };

bool Vector13[] = { 1,1,1,0,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1 };

bool Vector21[] = { 1,1,1,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0 }; //4

bool Vector22[] = { 1,1,0,1,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0 };

bool Vector23[] = { 1,1,0,0,0,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0 };

bool Vector31[] = { 1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,0,0,1,1,1 }; //8

bool Vector32[] = { 1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,0,1,0,1,1 };

bool Vector33[] = { 1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,1,0,0,1,1 };

vectors[3] = Vector11;

vectors[4] = Vector12;

vectors[5] = Vector13;

vectors[6] = Vector21;

vectors[7] = Vector22;

vectors[8] = Vector23;

vectors[9] = Vector31;

vectors[10] = Vector32;

vectors[11] = Vector33;

for (int i = 0; i < 12; i++) {

inputs = vectors[i];

cout << "Вектор " << i + 1 << " = ";

for (int j = 0; j < 20; j++) {

cout << inputs[j] << ' ';

}

cout << endl << "Результат : ";

hidden\_pred = Hiddens(inputs, w12, T\_hid);

values = Result(inputs, w12, T\_hid, w23, T\_out, hidden\_pred);

cout << values[0] << ' ' << values[1] << ' ' << values[2] << endl;

cout << endl;

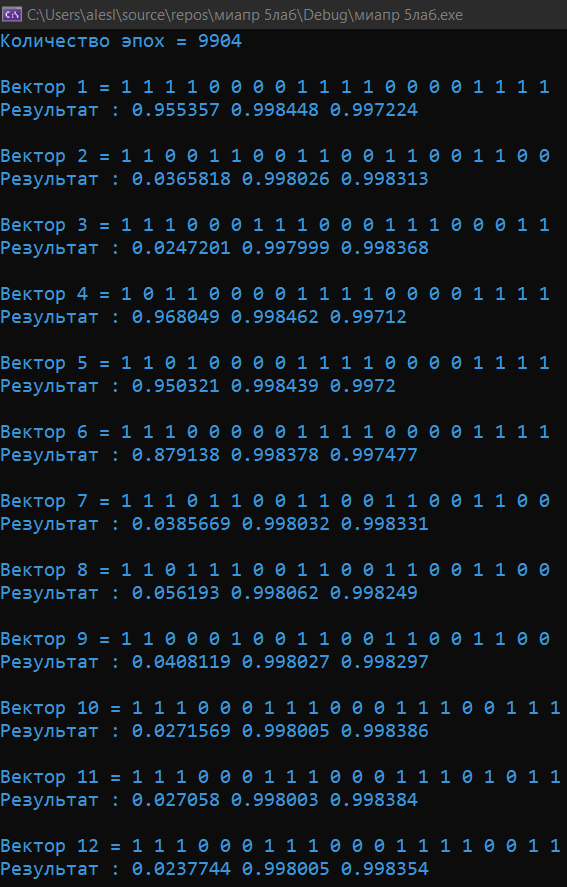
}

system("pause");

return 0;

}

Результат программы:



Вывод: изучил обучение и функционирование нелинейной ИНС при решении задач распознавания образов.