Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

Лабораторная работа №5

за 1 семестр

По дисциплине: «МиАПР»

Тема: «Нелинейные ИНС в задачах распознавания объектов»

Выполнил:

Студент 2 курса

Группы ПО-4(1)

Воробей Анастасия Петровна

Проверил:

Крощенко Александр Александрович

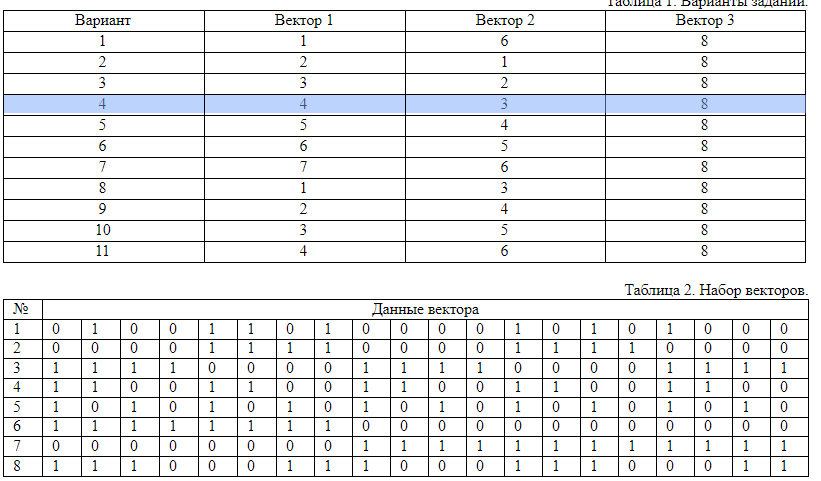
Брест 2020

Лабораторная работа №5

Нелинейные ИНС в задачах распознавания объектов

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

Вариант 4



Код программы:

#include <iostream>

using namespace std;

double Sigmoid(double x) {

return 1 / (1 + pow(2.71828, -x));

}

double\* Gethiddens(bool entrances[], double Wes1\_2[20][40], double T\_hid[]) {

double\* hidden = new double[40];

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

hidden[i] = 0;

}

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

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

hidden[i] += Wes1\_2[j][i] \* entrances[j];

}

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

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

}

return hidden;

}

double\* result(bool entrances[], double Wes1\_2[20][40], double T\_hid[], double Wes2\_3[40][3], double T\_out[], double hidden[40]) {

double\* Results = new double[3];

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

Results[i] = 0;

}

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

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

Results[j] += hidden[i] \* Wes2\_3[i][j];

}

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

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

}

return Results;

}

int main() {

setlocale(0,"");

int eras = 0;

bool Vect\_1[] = { 1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0 };

bool Vect\_2[] = { 1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1 };

bool Vect\_3[] = { 1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1 };

bool\* entrances = new bool[20];

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

entrances[i] = 0;

}

bool\*\* Vectors = new bool\* [8];

Vectors[0] = Vect\_1;

Vectors[1] = Vect\_2;

Vectors[2] = Vect\_3;

double Wes1\_2[20][40], Wes2\_3[40][3], T\_hid[40], T\_out[3], E\_min = 0.001, alpha = 0.04, reference, E = 0, Outputs[3] = { 0 };

double\* currents = new double[3];

double\* hidden = new double[40];

double errors[3] = { 0 };

double references[3] = { 0 };

double errorsHid[40] = { 0 };

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

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

Wes1\_2[i][j] = ((double)rand() / (RAND\_MAX)) - 0.5;

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

Wes2\_3[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 {

E = 0;

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

references[0] = references[1] = references[2] = 0;

references[N] = 1;

entrances = Vectors[N];

hidden = Gethiddens(entrances, Wes1\_2, T\_hid);

currents = result(entrances, Wes1\_2, T\_hid, Wes2\_3, T\_out, hidden);

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

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

}

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

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

errorsHid[j] += errors[m] \* currents[m] \* (1 - currents[m]) \* Wes2\_3[j][m];

}

}

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

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

Wes2\_3[i][j] -= alpha \* errors[j] \* currents[j] \* (1 - currents[j]) \* hidden[i];

}

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

}

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

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

Wes1\_2[i][j] -= alpha \* errorsHid[j] \* hidden[j] \* (1 - hidden[j]) \* entrances[i];

}

T\_hid[j] += alpha \* errorsHid[j] \* hidden[j] \* (1 - hidden[j]);

}

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

}

E /= 2;

cout << "\rError: " << E;

eras++;

} while (E > E\_min);

cout << endl;

cout << eras << endl;

double\* HiddenPred;

double\* Values;

bool Vectors3[] = { 0,0,0,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0 };

bool Vectors4[] = { 0,0,1,0,1,1,0,0,1,1,0,0,1,1,0,0,1,1,0,0 };

bool Vectors5[] = { 0,0,1,0,1,1,0,0,0,1,0,0,1,1,0,0,1,1,0,0 };

bool Vectors6[] = { 1,1,1,1,1,0,0,0,1,0,1,1,0,0,0,0,0,1,1,1 };

bool Vectors7[] = { 0,1,1,1,1,1,0,0,1,1,1,1,0,0,0,0,1,1,1,1 };

Vectors[3] = Vectors3;

Vectors[4] = Vectors4;

Vectors[5] = Vectors5;

Vectors[6] = Vectors6;

Vectors[7] = Vectors7;

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

entrances = Vectors[i];

cout << "Vector " << i + 1 << ": ";

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

cout << entrances[j] << " ";

}

cout << endl;

cout << "REsult: ";

HiddenPred = Gethiddens(entrances, Wes1\_2, T\_hid);

Values = result(entrances, Wes1\_2, T\_hid, Wes2\_3, T\_out, HiddenPred);

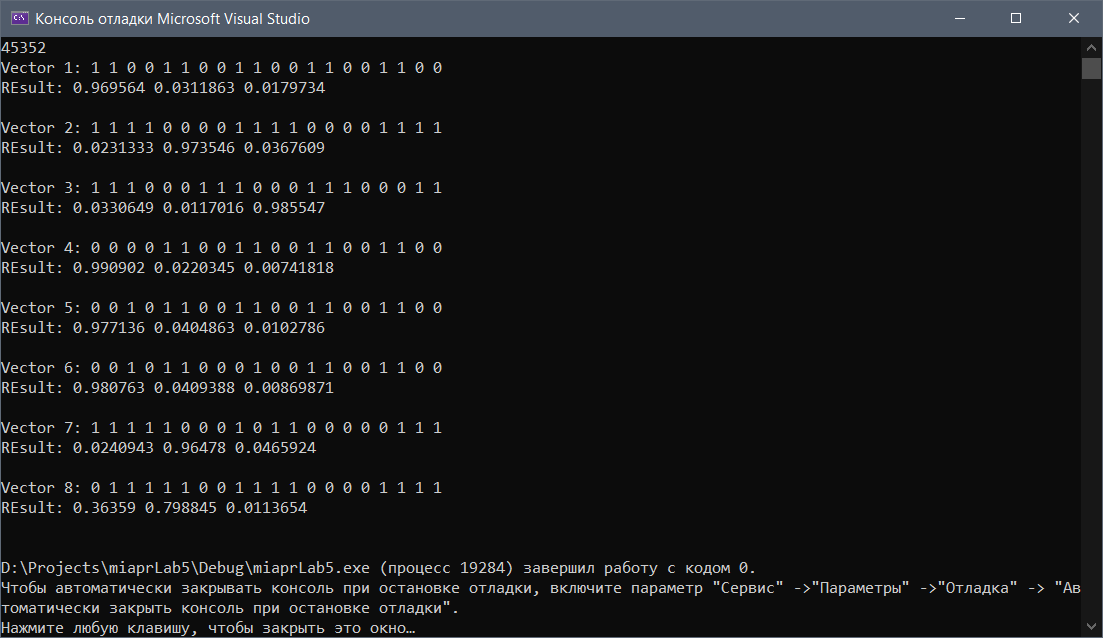
cout << Values[0] << " " << Values[1] << " " << Values[2] << endl;

cout << endl;

}

}

Какой результат меньше, на тот вектор и похожа последовательность.



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