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 образов»

Выполнил:

Студент 2 курса

Группы ПО-4(1)

Боровский М.В.

Проверил:

Крощенко А.А.

Брест 2020

Лабораторная работа №5

Нелинейные ИНС в задачах распознавания образов

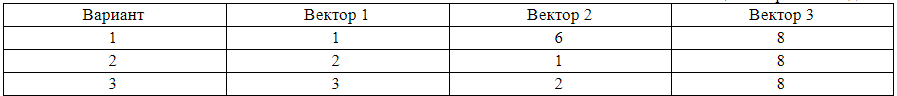
Цель работы: Изучить обучение и функционирование нелинейной ИНС при решении задач распознавания образов.

Вариант 3

Задание:

**Задание*.***

1. Изучить теоретические сведения, приведенные в данных методических указания и методических указаниях для работ №3 и №4.
2. 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.
3. 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.





Код программы:

#include <iostream>

using namespace std;

double sigmoid(double x)

{

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

}

double\* getHidden(bool\* Inputs, double W12[20][40], double THid[])

{

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] += W12[j][i] \* Inputs[j];

}

hidden[i] -= THid[i];

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

}

return hidden;

}

double\* getResult(bool\* Inputs, double W12[20][40], double THid[], double W23[40][3], double TOut[], 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] \* W23[i][j];

}

Results[j] -= TOut[j];

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

}

return Results;

}

int main()

{

setlocale(LC\_ALL, "rus");

bool Vect1[] = { 1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1 };

bool Vect2[] = { 0,0,0,0,1,1,1,1,0,0,0,0,1,1,1,1,0,0,0,0 };

bool Vect3[] = { 1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,0,0,0,1,1 };

bool\* Inputs = new bool[20];

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

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

Vectors[0] = Vect1;

Vectors[1] = Vect2;

Vectors[2] = Vect3;

double W12[20][40], W23[40][3], THid[40], TOut[3], E\_min = 0.001, alpha = 0.04, Ethalon, E = 0, Outputs[3] = { 0 };

double\* Currents = new double[3];

double\* hidden = new double[40];

double Mistakes[3] = { 0 };

double Ethalons[3] = { 0 };

double MistakesHid[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;

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

}

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

}

}

do

{

E = 0;

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

{

Ethalons[0] = Ethalons[1] = Ethalons[2] = 0;

Ethalons[N] = 1;

Inputs = Vectors[N];

hidden = getHidden(Inputs, W12, THid);

Currents = getResult(Inputs, W12, THid, W23, TOut, hidden);

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

Mistakes[i] = Currents[i] - Ethalons[i];

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

{

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

MistakesHid[j] += Mistakes[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] -= alpha \* Mistakes[j] \* Currents[j] \* (1 - Currents[j]) \* hidden[i];

}

TOut[j] += alpha \* Mistakes[j] \* Currents[j] \* (1 - Currents[j]);

}

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

{

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

{

W12[i][j] -= alpha \* MistakesHid[j] \* hidden[j] \* (1 - hidden[j]) \* Inputs[i];

}

THid[j] += alpha \* MistakesHid[j] \* hidden[j] \* (1 - hidden[j]);

}

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

}

E /= 2;

cout << E << endl;

} while (E > E\_min);

double\* HiddenPred;

double\* Values;

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

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

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

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

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

bool Vectors8[] = { 1,1,1,1,1,1,1,0,1,1,1,1,0,0,0,0,1,1,1,1 };

bool Vectors9[] = { 1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,0,0,0,1,0 };

bool Vectors10[] = { 1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,0,0,0,0,0 };

bool Vectors11[] = { 1,1,1,0,0,0,1,1,1,0,0,0,1,1,1,0,0,1,0,0 };

Vectors[3] = Vectors3;

Vectors[4] = Vectors4;

Vectors[5] = Vectors5;

Vectors[6] = Vectors6;

Vectors[7] = Vectors7;

Vectors[8] = Vectors8;

Vectors[9] = Vectors9;

Vectors[10] = Vectors10;

Vectors[11] = Vectors11;

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: ";

HiddenPred = getHidden(Inputs, W12, THid);

Values = getResult(Inputs, W12, THid, W23, TOut, HiddenPred);

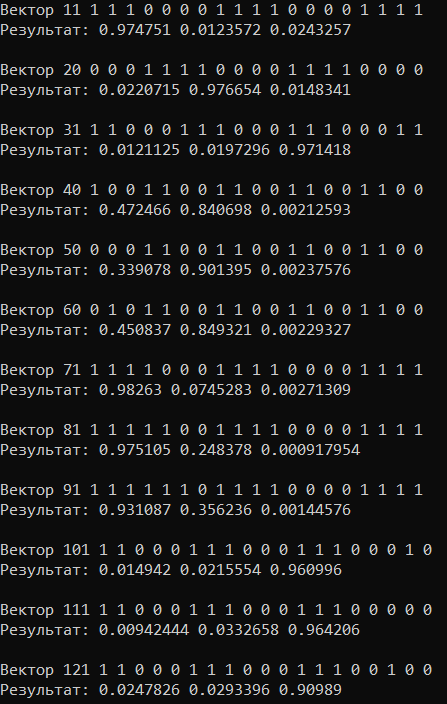
cout << Values[0] << ' ' << Values[1] << ' ' << Values[2] << endl;

cout << endl;

}

system("pause");

}



Вывод: В ходе работы разработал нелинейную ИНС для задач распознавания образов.