МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное общеобразовательное учреждение высшего образования

«Ижевский государственный технический университет имени М.Т. Калашникова»

Кафедра «Программное обеспечение»

ОТЧЕТ

по лабораторной работе №1

по дисциплине «Математические основы искусственного интеллекта»

на тему «Персептроны»

Вариант №4

Выполнил:

студент группы Б18-191-2 Р.А. Гумметов

Принял:

канд. техн. наук, доцент А.В. Коробейников

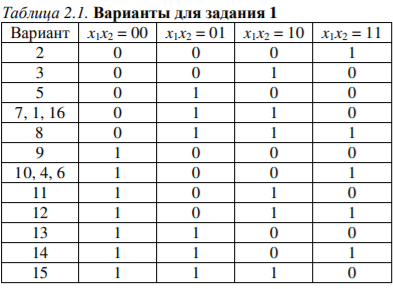
Ижевск 2021

ЦЕЛЬ РАБОТЫ

Практическое освоение основ нейронных сетей. Выполняется в виде программы на языке высокого уровня. В ходе работы использовать персептроны – пороговую функцию активации (порог θ = 0). При необходимости использовать вход смещения b = w0.

ПОСТАНОВКА ЗАДАЧИ

Реализовать с помощью персептронов логическую функцию двух переменных в соответствии с номером варианта. Вид функции приведен в табл. 2.1. Если функция для варианта линейно неразделима, то необходимо использовать число слоев сети 2 или 3. Допускается использовать ручное задание синаптических коэффициентов (без обучения сети).



ИДЕЯ РЕШЕНИЯ ЗАДАЧИ

Воспользуемся примером из методички. Разделим логическую функцию на три более простых логических функции. Первый нейрон будет реализовывать функцию "NOT OR", второй - "AND". Это будут два промежуточных нейрона. Результат работы последнего нейрона - функция "OR" нейронов скрытого слоя. Для каждого варианта подбираются свои веса. Они корректируются с учетом ошибки на этапах обучения с помощью дельта правила. С помощью пороговой функции вычисляется результат на каждой промежуточной итерации. Веса корректируются до тех пор, пока ожидаемый результат работы нейрона не будет совпадать с реальным результатом.

КОД ПРОГРАММЫ

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ConsoleApp21

{

class Program

{

static void Main(string[] args)

{

int[] x1 = { 0, 0, 1, 1 };

int[] x2 = { 0, 1, 0, 1 };

int[] r1 = { 1, 0, 0, 0 };

int[] r2 = { 0, 0, 0, 1 };

int[] R = { 1, 0, 0, 1 };

double[] w0 = { 0, 0, 0 };

double[] w1 = { 0, 0, 0 };

double[] w2 = { 0, 0, 0 };

w0 = FindW(x1, x2, r1);

w1 = FindW(x1, x2, r2);

w2 = FindW(r1, r2, R);

Console.WriteLine("w0 = {0} {1} {2}", w0[0], w0[1], w0[2]);

Console.WriteLine("w1 = {0} {1} {2}", w1[0], w1[1], w1[2]);

Console.WriteLine("w2 = {0} {1} {2}\n", w2[0], w2[1], w2[2]);

Console.WriteLine("Ожидаемый результат:");

for(int i = 0; i < R.Length; i++)

{

Console.Write(R[i] + " ");

}

Console.WriteLine("\n\nРеальный результат:");

int[] RR = new int[4];

for(int i = 0; i < 4; i++)

{

double s = w2[0] + w2[1] \* r1[i] + w2[2] \* r2[i];

if (s >= 0)

RR[i] = 1;

else

RR[i] = 0;

Console.Write(RR[i] + " ");

}

Console.ReadLine();

}

static public double[] FindW(int[] x1, int[] x2, int[] r)

{

double[] w = { 1, 0, 0 };

int[] res = { 0, 0, 0, 0 };

double b = 0;

double delta0 = 0;

double delta1 = 0;

double delta2 = 0;

double n = 1.0f;

while(!ArrayCompare(r, res))

{

for(int i = 0; i < 4; i++)

{

double s = w[0] + x1[i] \* w[1] + x2[i] \* w[2];

if (s >= 0)

res[i] = 1;

else

res[i] = 0;

b = r[i] - res[i];

delta0 = n \* b;

delta1 = n \* b \* x1[i];

delta2 = n \* b \* x2[i];

w[0] += delta0;

w[1] += delta1;

w[2] += delta2;

}

}

return w;

}

static public bool ArrayCompare(int[] a, int[] b)

{

for (int i = 0; i < a.Length; i++)

{

if (a[i] == b[i])

{

if (i == a.Length - 1)

{

return true;

}

continue;

}

else

return false;

}

return false;

}

}

}

РЕЗУЛЬТАТ РАБОТЫ ПРОГРАММЫ

Программа выводит веса, ожидаемый результат работы и реальный результат работы.



Рис.1 - Результат работы программы