Technická univerzita v Košiciach
Fakulta elektrotechniky a informatiky
Katedra kybernetiky a umelej inteligencie

Umelá inteligencia II

(Zadanie – Perceptron)

2. ročník, Inteligentné systémy
Zimný semester 2009

Michal Puheim

Text zadania

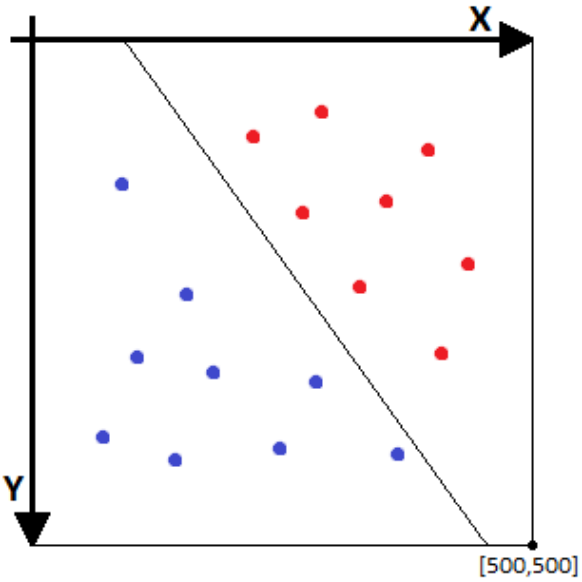
Naprogramujte jednoduchý perceptron a realizujte experimenty na lineárne seprabilných, ako aj neseparabilných dátach. Program prezentujte v programovacom jazyku C,C++,C#,java alebo Matlab. Vypracujte detailnú správu o chovaní týchto sietí a ich vplyvu ich parametrov.

Dodefinovanie zadania

Pre vypracovanie zadania som si vybral programovací jazyk C# v prostredí Visual Studio od Microsoftu. Ako vstupné dáta som určil súradnice bodov na ploche o rozmere 500x500. Perceptron mal úlohu tieto body rozdeliť (priamkou) podľa farby, resp. triedy (modrá/červená).

Analýza úlohy

Perceptrón bude má na vstupe dve čísla - súradnice bodov plochy, ktoré triedi podľa vnútorného predpisu. Tieto súradnice musia byť normované na hodnoty od 0 do 1, teda v prípade rozmeru plochy 500x500 budú delené 500. Ku každej dvojici súradníc (ku každému bodu) užívateľ priradí triedu (farbu), pomocou ktorej bude kontrolovaný výstup z perceptrónu. Kontrola bude pozostávať z porovnania výstupu z perceptrónu pre daný bod a očakávaného výstupu, tj. triedy zadanej užívateľom. Chyba určená z tejto kontroly bude určovať zmenu váh na vstupoch. Táto zmena bude prebiehať v cykloch (iteráciách), pričom jeden cyklus obsiahne spracovanie, kontrolu a zmenu váh pre všetky body.

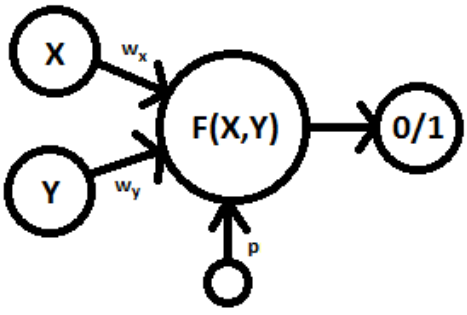


Návrh riešenia

Program po spustení zobrazí okno s vykresľovacou plochou, do ktorej bude možné kliknutím myši vložiť body (ľavým tlačidlom červené a pravým modré). Pri kliknutí na plochu sa na nej zobrazí bod v danej farbe a súčasne sa do pamäti uložia jeho súradnice a trieda (farba).

Pod plochou budú tri ovládacie tlačidlá – jedno na spustenie učenia perceptrónu, druhé na inicializáciu váh a tretie na vymazanie vykresľovacej plochy. Inicializácia váh nastaví váhy na náhodné hodnoty. A vymazanie odstráni body z plochy aj príslušné údaje v pamäti.

Samotné učenie perceptrónu bude riešené v dvoch vnorených cykloch. Vnútorný bude postupne pre všetky uložené body kontrolovať výstup z perceptrónu oproti očakávanému výstupu (tj. trieda bodu). Pre každý bod sa takto určí lokálna chyba na základe ktorej sa budú meniť váhy. Pre všetky body určíme globálnu chybu, ktorá je definovaná ako súčet absolútnych hodnôt všetkých lokálnych chýb. Vonkajší cyklus bude opakovať vnútorný, až kým program nenájde správne váhy, tj. globálna chyba bude rovná nule (resp. až kým sa nedosiahne určitý vopred daný počet iterácií – tj. zabezpečenie proti dátam, ktoré perceptrón nedokáže rozdeliť).



Vlastný perceptrón je v programe realizovaný ako predpis (funkcia), ktorý určí na základe dvoch vstupných údajov (súradníc) jeden výstupný údaj (trieda/farba). Táto funkcia je vlastne zložením dvoch funkcií – vstupnej a pracovnej. Vstupná funkcia perceptrónu má zvyčajne tvar:

$$IN(x, y) = x_{in} \cdot w_x + y_{in} \cdot w_y - w_p$$

kde x_i, y_i sú vstupy (súradnice), w_x, w_y sú váhy a w_p je prah. Keďže perceptrón pracuje s hodnotami od 0 do 1, musíme súradnice normalizovať, teda dostaneme funkciu:

$$IN(x, y) = \frac{x_i}{500} \cdot w_x + \frac{y_i}{500} \cdot w_y - w_p$$

Pracovná funkcia pomocou jednotkového skoku transformuje hodnotu vstupnej funkcie na výstup. Má tvar:

$$IN \leqslant \frac{1}{2} \quad \Rightarrow \quad OUT(IN) = 0$$
$$IN > \frac{1}{2} \quad \Rightarrow \quad OUT(IN) = 1$$

kde ½ je hranica („treshold“), ktorá rozhoduje, či bude výstup 0 alebo 1. Lokálna chyba sa potom počíta podľa rovnice $J_L = IN_e - IN$, kde IN_e je očakávaný výstup (tj. trieda bodu). Zmenu váh určíme podľa rovníc:

$$w_x = w_x + \alpha \cdot J_{Li} \cdot \frac{x_i}{500}$$
$$w_y = w_y + \alpha \cdot J_{Li} \cdot \frac{y_i}{500}$$
$$w_p = w_p + \alpha \cdot J_{Li} \cdot -1$$

kde α je parameter učenia a i index daného bodu. Globálna chyba je určená: $J = \sum_{i=1}^{n} |J_{Li}|$

Rovnicu priamky určíme spojením oboch funkcií:

$$\frac{1}{2} = X \cdot w_x + Y \cdot w_y - 500 \cdot w_p$$

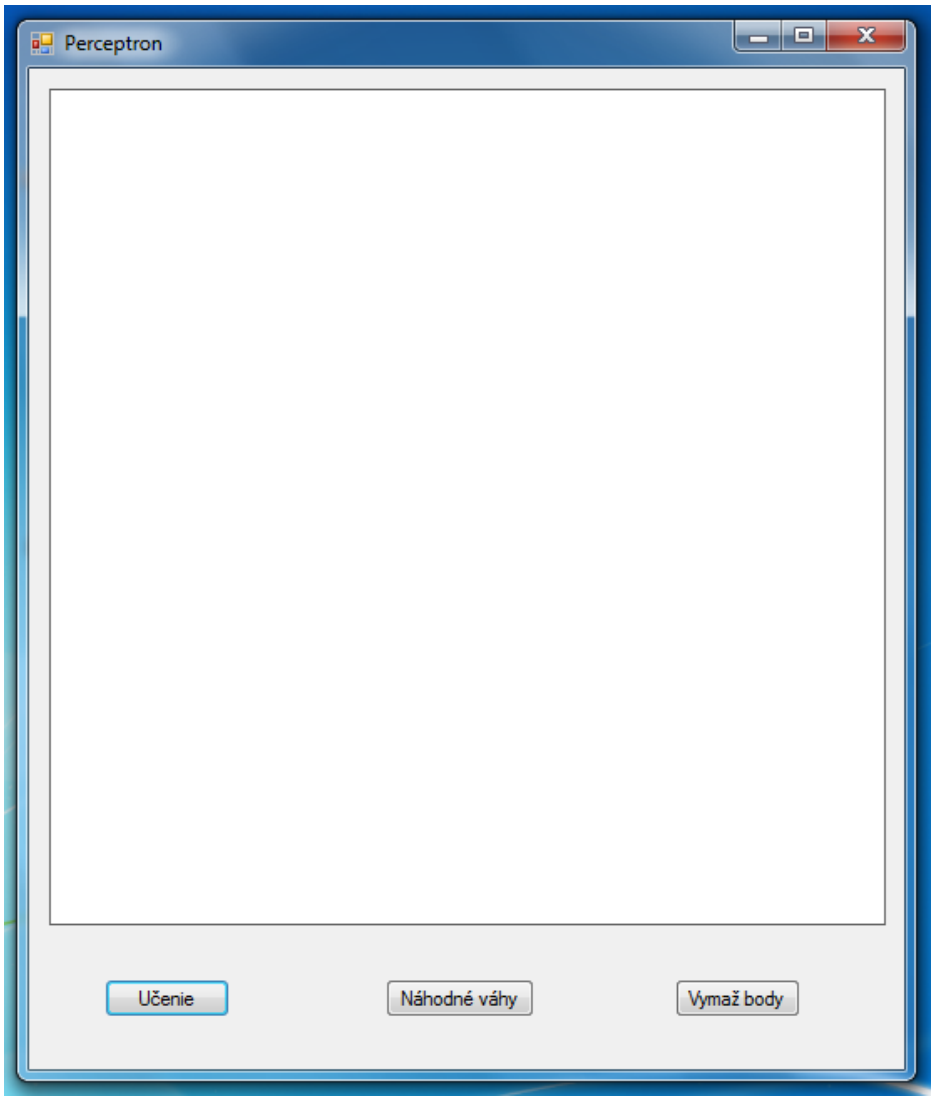
kde X,Y sú súradnice ľubovoľného bodu priamky. Odvodením dostaneme rovnice v explicitnom tvare:

$$X = -\frac{w_y}{w_x} Y + \frac{1/2 + 500 \cdot w_p}{w_x}$$
$$Y = -\frac{w_x}{w_y} X + \frac{1/2 + 500 \cdot w_p}{w_y}$$

Pomocou týchto rovníc vieme vykresliť priamku, ktorá rozdelí body na dve skupiny.

Implementácia riešenia

Program je naprogramovaný v prostredí Microsoft Visual Studio v jazyku C#. Vizuálna časť aplikácie, vstupný formulár *Form1.cs* je je navrhnutý s hlavnou kresliacou plochou o rozmere 500x500 a troma kontrolnými tlačidlami pod ňou:



Ovládacie prvky:

Kresliaca plocha -> `private void panel1_MouseClick(object sender, MouseEventArgs e).`
Pri kliknutí na plochu sa na nej zobrazí bod v danej farbe a súčasne sa do dvojrozmerného poľa `body[,]` uložia jeho súradnice a trieda (prvý rozmer je poradie bodu a druhý určuje poporadí: x,y- súradnicu a triedu). Každým zadaním bodu sa zvýsi počítadlo bodov o 1.

Učenie -> `private void button1_Click(object sender, EventArgs e)`
Spúšťa vlastný proces učenia perceptrónu. Ak nájde správne váhy do doby kratšej ako 1000 iterácií, potvrdí nájdenie riešenia a vykrelí priamku určenú aktuálnymi váhami. V prípade, že riešenie nenájde je možné pokračovať aj nad 1000 iterácii ďalším kliknutím na tlačidlo.

Náhodné váhy -> `private void button2_Click(object sender, EventArgs e)`
Nastaví náhodné váhy a zmaže priamku ak už bola vykreslená.

Vymaž body -> `private void button3_Click(object sender, EventArgs e)`
Vymaže kresliacu plochu a vynuluje počítadlo bodov.

Zdrojový kód Form1.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        int pocitadlo = 0;
        int celkove_iteracie = 0; //celkovy pocet iteracii
        double[,] body = new double[10000,3];
        double vahaX, vahaY, vahaP; // Vahy
        bool nastavene_vahy = false;

        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e) //Ucenie
        {
            if (nastavene_vahy==false)
            {
                MessageBox.Show("Váhy neboli inicializované!");
                return;
            }
            if (pocitadlo > 0)
            {
                // Prefarbi celu plochu na pociatocnu farbu
                System.Drawing.Graphics graphicsObj;
                graphicsObj = panel1.CreateGraphics();
                Pen myPen = new Pen(System.Drawing.Color.White, 600);
                Rectangle myRectangle = new Rectangle(0, 0, 500, 500);
                graphicsObj.DrawRectangle(myPen, myRectangle);
                // Opatovne vykresli zadane body
                for (int p = 0; p < pocitadlo; p++)
                {
                    graphicsObj = panel1.CreateGraphics();
                    if (body[p, 2] == 0) //cerveny
                        myPen = new Pen(System.Drawing.Color.Red, 2);
                    else //modry
                        myPen = new Pen(System.Drawing.Color.Blue, 2);
                    myRectangle = new Rectangle((int)(body[p, 0]), (int)(body[p, 1]), 2, 2);
                    graphicsObj.DrawRectangle(myPen, myRectangle);
                }
                // Nastav parameter ucenia.
                double ucenie = 0.05;
                int iteracia = 0; //Iteracie v ramci tohto cyklu ucenia
                double globalna_chyba;
                // Jadro perceptronu.
                int vystup;
                do
                {
                    globalna_chyba = 0;
                    for (int p = 0; p < pocitadlo; p++)
                    {
                        // Vypocitaj vystup.
                        double funkcia = (body[p, 0] / 500) * vahaX + (body[p, 1] / 500) * vahaY -
                        vahaP;
                        if (funkcia <= (1 / 2)) //500 je rozmer vykreslovacej plochy, delenie
                        suradnice bodu je kvoli normalizacii pre perceptron
                            vystup = 0;
                        else
                            vystup = 1;
                        // Vypocitaj lokalnu chybu (tj. chybu pre dany vstup/bod).
                        double lokalna_chyba = body[p, 2] - vystup; // chyba = ocakavany vystup -
                        vystup
                        if (lokalna_chyba != 0)
                        {
                            // Ak je lokalna chyba vacsia ako nula zmen vahy.
                            vahaX += ucenie * lokalna_chyba * (body[p, 0] / 500);
                            vahaY += ucenie * lokalna_chyba * (body[p, 1] / 500);
                            vahaP += ucenie * lokalna_chyba * (-1);
```

```
                }
                // Absolutna hodnota chyby.
                globalna_chyba += Math.Abs(lokalna_chyba);
            }
            iteracia++; //Aktualizuj iteracie v ramci tohto cyklu ucenia
        } while ((globalna_chyba != 0)&&(iteracia<1000));
        celkove_iteracie += iteracia; //Aktualizuj celkovy pocet iteracii
        if (iteracia < 1000)
        {
            MessageBox.Show(string.Format("Body boli úspešne odseparované po {0}
            iteráciách.", celkove_iteracie));
        }
        else
            MessageBox.Show(string.Format("Body neboli odseparovane ani po {0} iteráciách.
        Zrejme nie sú lineárne separovateľné.", celkove_iteracie));
        //  Vykresli priamku
        graphicsObj = panel1.CreateGraphics();
        myPen = new Pen(System.Drawing.Color.Black, 1);
        if (vahaY/vahaX<0.001)
            graphicsObj.DrawLine(myPen, -5000, (float)((5000 * vahaX + 0.5 + 500 * vahaP) /
            vahaY), 5500, (float)((-5500 *  vahaX + 0.5 + 500 * vahaP) / vahaY));
        else
            graphicsObj.DrawLine(myPen, (float)((5000 * vahaY + 0.5 + 500 * vahaP) / vahaX),
            -5000, (float)((-5500 * vahaY + 0.5 + 500 * vahaP) / vahaX), 5500);
    }
    else
    {
        MessageBox.Show("Je potrebne zadat nejake body!");
    }
}

private void button2_Click(object sender, EventArgs e) //Nahodne nastavenie vah
{
    // Randomizuj vahy.
    Random r = new Random();
    vahaX = r.NextDouble();
    vahaY = r.NextDouble();
    vahaP = r.NextDouble(); //Vaha na prahu
    // Prefarbi celu plochu na pociatocnu farbu
    System.Drawing.Graphics graphicsObj;
    graphicsObj = panel1.CreateGraphics();
    Pen myPen = new Pen(System.Drawing.Color.White, 600);
    Rectangle myRectangle = new Rectangle(0, 0, 500, 500);
    graphicsObj.DrawRectangle(myPen, myRectangle);
    // Opatovne vykresli zadane body
    for (int p = 0; p < pocitadlo; p++)
    {
        graphicsObj = panel1.CreateGraphics();
        if (body[p, 2] == 0) //cerveny
            myPen = new Pen(System.Drawing.Color.Red, 2);
        else //modry
            myPen = new Pen(System.Drawing.Color.Blue, 2);
        myRectangle = new Rectangle((int)(body[p, 0]), (int)(body[p, 1]), 2, 2);
        graphicsObj.DrawRectangle(myPen, myRectangle);
    }
    MessageBox.Show("Vahy boli nastavene na nahodne hodnoty!");
    // Vymaz celkove iteracie
    celkove_iteracie = 0;
    // Oznac vahy ako nastavene
    nastavene_vahy = true;
}

private void button3_Click(object sender, EventArgs e) //Vymazanie bodov
{
    // Prefarbi celu plochu na pociatocnu farbu
    System.Drawing.Graphics graphicsObj;
    graphicsObj = panel1.CreateGraphics();
    Pen myPen = new Pen(System.Drawing.Color.White, 600);
    Rectangle myRectangle = new Rectangle(0, 0, 500, 500);
    graphicsObj.DrawRectangle(myPen, myRectangle);
    MessageBox.Show("Vsetky body boli odstranene!");
    // Resetuj pocitadlo
    pocitadlo = 0;
    celkove_iteracie = 0;
}

private void panel1_MouseClick(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
```

```
                int x_sur = e.X;
                int y_sur = e.Y;
                System.Drawing.Graphics graphicsObj;
                graphicsObj = panel1.CreateGraphics();
                Pen myPen = new Pen(System.Drawing.Color.Red, 2);
                Rectangle myRectangle = new Rectangle(x_sur, y_sur, 2, 2);
                graphicsObj.DrawRectangle(myPen, myRectangle);
                body[pocitadlo, 0] = x_sur; //x-suradnica bodu
                body[pocitadlo, 1] = y_sur; //y-suradnica bodu
                body[pocitadlo, 2] = 0;     //trieda bodu (cerveny)
                pocitadlo++;
            }
            if (e.Button == MouseButtons.Right)
            {
                int x_sur = e.X;
                int y_sur = e.Y;
                System.Drawing.Graphics graphicsObj;
                graphicsObj = panel1.CreateGraphics();
                Pen myPen = new Pen(System.Drawing.Color.Blue, 2);
                Rectangle myRectangle = new Rectangle(x_sur, y_sur, 2, 2);
                graphicsObj.DrawRectangle(myPen, myRectangle);
                body[pocitadlo, 0] = x_sur; //x-suradnica bodu
                body[pocitadlo, 1] = y_sur; //y-suradnica bodu
                body[pocitadlo, 2] = 1;     //trieda bodu (modry)
                pocitadlo++;
            }
        }
    }
}
```

Záver

Program spoľahlivo rieši lineárne separabilné dáta a v prípade lineárne neseparabilných dát informuje, že nenašiel riešenie. Program bol vyskúšaný pre rôzne paramerte učenia perceptrónu a približne pre hodnotu 0.5 podával najrýchlejšie riešenie (vzhľadom na počet iterácií).

Ospravedlňujem sa za poslabšiu dokumentáciu implementácie v C#, ale keďže nemám teoretické základy jazyka ani sa neorientujem v objektovom programovaní, je pre mňa zložité opisovať jednotlivé objekty a vzťahy medzi nimi. Snažil som sa to vynahradiť teoretickým výkladom.