Програма здійснює кластеризацію вхідного потоку данних, що складається з двох множин випадкових двохвимірних векторів із гаусівським законом розподілу.

procedure TForm1.Button1Click(Sender: TObject);

{eps – максимально допустиме значення похибки

alpha - коефіцієнт крутизни функції активації

fout - вихід мережі

eta - швидкість навчання

NET – значення суми добутків відповідних компонент векторів x і w

delta - нев’язка нейрона

E - функціонал похибки

х – вектор вхідних сигналів для кожної з навчаючих пар

Т – бажаний вихід для заданої навчаючої пари

w - вектор ваг нейрона

dE - вектор похідних функціонала похибки по відповідній компоненті вектора ваг w

k – кількість епох навчання

i, j, l – лічильники для циклів

n - кількість навчаючих пар

teached – логічна змінна, яка показує, чи “навчився нейрон”}

var eps, eta, delta, alpha, m1, m2, D1, D2, zs, zz, Es: Real;

i, j, l, k, n:Integer;

x: array [0..2000,0..2] of Real;

w: array [0..2] of Real;

E: array [0..2000] of Real;

fout: array [0..2000] of Real;

NET: array [0..2000] of Real;

T: array[0..2000] of Integer;

dE: array [0..2]of Real;

teached: Boolean;

begin

{блок ідентифікації змінних}

alpha := StrToFloat(Edit1.Text);

eps := StrToFloat(Edit2.Text);

eta := StrToFloat(Edit3.Text);

n:=StrToInt(Edit7.Text);

m1:=0.1; m2:=0.9;

D1:=0.1; D2:=0.2;

Randomize;

for j:=0 to (Round(n/2)-1) do

begin

x[j,2]:=1;

T[j]:=1;

for l:=0 to 1 do

begin

zs:=0;

for i:=1 to 5 do

zs:=zs+Random;

zz:=sqrt(12/5)\*(zs-5/2);

x[j,l]:=m1+D1\*(0.01\*zz\*(97+zz\*zz));

end;

end;

for j:=Round(n/2)-1 to n do

begin

x[j,2]:=1;

T[j]:=-1;

for l:=0 to 1 do

begin

zs:=0;

for i:=1 to 5 do

zs:=zs+Random;

zz:=sqrt(12/5)\*(zs-5/2);

x[j,l]:=m2+D2\*(0.01\*zz\*(97+zz\*zz));

end;

end;

for i:=0 to 2 do

begin

w[i]:=0.5;

end;

Edit4.Text:=FloatToStr(w[0]);

Edit5.Text:=FloatToStr(w[1]);

Edit6.Text:=FloatToStr(w[2]);

{блок навчання мережі}

k := 0;

teached := False;

repeat

teached := true;

Es := 0;

for l := 0 to n-1 do

begin

NET[l] := x[l, 0] \* w[0] + x[l, 1] \* w[1] + x[l, 2] \* w[2]; fout[l] := 1.0/(1.0+exp(-alpha\*NET[l]));

E[l] := 0.5 \* (fout[l] - T[l]) \* (fout[l] - T[l]);

Es := Es + E[l];

end;

if ((Es/n) > eps) then

begin

k := k + 1;

teached := false;

for j := 0 to n-1 do

begin

delta := alpha\*(fout[j]-T[j])\*fout[j]\*(1-fout[j]);

for i := 0 to 2 do

begin

dE[i] := delta \* x[j, i];

w[i] := w[i] - eta \* dE[i];

end;

end;

end;

if (k > 10000) then break;

until (teached = true);

{блок перевірки якості навчання}

Edit8.Text:=IntToStr(k);

Chart1.Series[0].Clear;

Chart1.Series[1].Clear;

Chart1.Series[2].Clear;

for j:=0 to 1000 do

begin

x[j,2]:=1;

for l:=0 to 1 do

begin

zs:=0;

for i:=1 to 5 do

zs:=zs+Random;

zz:=sqrt(12/5)\*(zs-5/2);

x[j,l]:=m1+D1\*(0.01\*zz\*(97+zz\*zz));

end;

Chart1.Series[1].AddXY(x[j,0],x[j,1],'',clGreen);

x[j,0]:=-(w[1]/w[0])\*x[j,1]-(w[2]/w[0]);

Chart1.Series[0].AddXY(x[j,0],x[j,1],'',clRed);

end;

for j:=1001 to 2000 do

begin

x[j,2]:=-1;

for l:=0 to 1 do

begin

zs:=0;

for i:=1 to 5 do

zs:=zs+Random;

zz:=sqrt(12/5)\*(zs-5/2);

x[j,l]:=m2+D2\*(0.01\*zz\*(97+zz\*zz));

end;

Chart1.Series[2].AddXY(x[j,0],x[j,1],'',clYellow);

x[j,0]:=-(w[1]/w[0])\*x[j,1]-(w[2]/w[0]);

Chart1.Series[0].AddXY(x[j,0],x[j,1],'',clRed);

end;

end;

**Висновок:** після навчання нейромережі на заданих навчаючих парах з двох множин з різними матсподіваннями та дисперсіями і з заданими бажаними виходами для кожної з них нейромережа здійснює кластеризацію вхідного потоку даних. Результати роботи програми наведені на графіках.