

Янкевич Анна Артемовна, группа 2.1

**Лабораторная работа №3
Вариант №9**

Отчет

Цель работы

Изучить возможности простейших нейронных сетей для классификации гауссовских случайных векторов.

Задание

1) Провести обучение нейросетевого классификатора для разделения гауссовских случайных векторов, представленных следующим образом, серым цветом отображается класс «+», белым цветом - «о».

2) Оценить ошибки первого и второго рода, и проиллюстрировать процесс их возникновения.

3) Построить график зависимости суммарной ошибки первого и второго рода от величины смещения центров кластеров.

4) Построить график зависимости суммарной ошибки первого и второго рода от объема обучающей выборки.

Код программы:

```
clear all; close all;
V=[-15 15 -15 15]; n=100;
ds=5;%смещение относительно осей координат
P1=randn(2,n);
P1(1,:)=P1(1,:).*2.4;
P1(2,:)=P1(2,:).*1.1;
P5=randn(2,n);
P5(1,:)=P5(1,:).*2.4 ;
P5(2,:)=P5(2,:).*1.1 + 5;
a1(1,1:n)=10;
a1(2,1:n)=0;
a2(1,1:n)=0;
a2(2,1:n)=10;
P2=randn(2,n).*0.8-a1;
P3=randn(2,n).*0.8+a1;
P4=randn(2,n).*0.8+a2;
P=[P1 P2 P3 P4 P5];%входной вектор
T1(1:n)=0;
T2(1:n)=0;
```

```

T3(1:n)=0;
T4(1:n)=0;
T5(1:n)=1;
T=[T1 T2 T3 T4 T5]; %целевой вектор
figure('Name','Исходное отображение маркеров');%графическое окно
plotpv(P,T);%отображение маркеров
%создание и обучение НС
toutn=0.1+(0.9-0.1)*T;%целевой выход
bpn=feedforwardnet([5 1],'trainrp');
view(bpn);
bpn=init(bpn);
bpn.trainParam.epochs=1000;
bpn=train(bpn,P,toutn);
% Проверяем ошибки 1-го и 2-го рода
P5=randn(2,n);
P5(1,:)=P5(1,:).*1.4 ;
P5(2,:)=P5(2,:).*1.1 + 5;
P=P5;%входной вектор
a=sim(bpn,P);%моделирование нейронной сети
n1=100;
for i=1:n1
    if a(i)>=0.5 a(i)=1; else a(i)=0; end; % округление полученных результатов
end;
figure('Name','Ошибка первого рода');%графическое окно;
plotpv(P,a);%отображение маркеров
OC1=1-sum(a)/(n1);
disp(OC1);
P1=randn(2,n);
P1(1,:)=P1(1,:).*2.4;
P1(2,:)=P1(2,:).*1.1;
P2=randn(2,n).*0.8-a1;
P3=randn(2,n).*0.8+a1;
P4=randn(2,n).*0.8+a2;
P=[P1 P2 P3 P4];
b=sim(bpn,P);
for i=1:4*n1,
    if b(i)>0.5 b(i)=1; else b(i)=0; end;
end;
figure('Name','Ошибка второго рода');%графическое окно;
plotpv(P,b);
OC2=sum(b)/(4*n1);
disp(OC2);
%%% Задание 3
M=ones(1,19);
for i=2:20
    P1=randn(2,n);
    P1(1,:)=P1(1,:).*2.4;
    P1(2,:)=P1(2,:).*1.1;
    a1(1,1:n)=10;
    a1(2,1:n)=0;
    a2(1,1:n)=0;
    a2(2,1:n)=10;
    P2=randn(2,n).*0.8-a1;

```

```

P3=randn(2,n).*0.8+a1;
P4=randn(2,n).*0.8+a2;
A(1,1:n)=0;
A(2,1:n)=2.5+i/8;
P5=randn(2,n);
P5(1,:)=P5(1,:).*1.4;
P5(2,:)=P5(2,:).*1.1;
P5=P5+A;%матрица входных векторов
P=[P1 P2 P3 P4 P5];%входной вектор
toutn=0.1+(0.9-0.1)*T;%целевой выход
bpn=feedforwardnet([5 1],'trainrp');
bpn=init(bpn);
bpn.trainParam.epochs=100;
bpn=train(bpn,P,toutn);
P6 = randn(2,n);
P6(1,:)=P6(1,:).*1.4;
P6(2,:)=P6(2,:).*1.1;
P6=P6+A;
P=[P6];
a=sim(bpn,P);
for j=1:n1
    if a(j)>=0.5 a(j)=1; else a(j)=0; end; % округление полученных
результатов
end;
if(i==2 || i==10)
    figure('Name','Ошибка первого рода');
    plotpv(P,a);
end
OC1(i)=1-sum(a)/(n1)
P1=randn(2,n);
P1(1,:)=P1(1,:).*2.4;
P1(2,:)=P1(2,:).*1.1;
a1(1,1:n)=10;
a1(2,1:n)=0;
a2(1,1:n)=0;
a2(2,1:n)=10;
P2=randn(2,n).*0.8-a1;
P3=randn(2,n).*0.8+a1;
P4=randn(2,n).*0.8+a2;
A(2,1:n)=2-(i/8);
P1=P1+A;
P=[P1 P2 P3 P4];
b=sim(bpn,P);
for j=1:4*n1,
    if b(j)>0.5 b(j)=1; else b(j)=0; end;
end;
if(i==2 || i==20)
    figure('Name','Ошибка второго рода');
    plotpv(P,b);
end
OC2(i)=sum(b)/(4*n1)
M(1,i-1)=OC1(i)+OC2(i);
end

```

```

C=0.25:0.125:2.5;
figure('Name','Зависимость ошибок от смещения центров классов по Y');
title('Зависимость суммарной ошибки от смещения центров кластеров'),
plot(C,M);
xlabel('смещение'),
ylabel('OC1+OC2'),
grid on;
% Задание 4 подсчёт зависимости ошибок от величины обучающей выборки
M2=ones(1,9);
N=20:10:100;
OC11 = zeros(1,9);
OC22 = zeros(1,9);
for nn=20:10:100
    P1=randn(2,nn);
    P1(1,:)=P1(1,:).*2.4;
    P1(2,:)=P1(2,:).*1.1;
    a11(1,1:nn)=10;
    a11(2,1:nn)=0;
    a22(1,1:nn)=0;
    a22(2,1:nn)=10;
    P2=randn(2,nn);
    P2=P2.*0.8-a11;
    P3=randn(2,nn);
    P3=P3.*0.8+a11;
    P4=randn(2,nn);
    P4=P4.*0.8+a22;
    P5=randn(2,nn);
    P5(1,:)=P5(1,:).*1.4;
    P5(2,:)=P5(2,:).*1.1+2.5+(20/8);
    P=[P1 P2 P3 P4 P5];%входной вектор
    T11(1:nn)=0;
    T22(1:nn)=0;
    T33(1:nn)=0;
    T44(1:nn)=0;
    T55(1:nn)=1;
    T=[T11 T22 T33 T44 T55]; %целевой вектор
    toutn=0.1+(0.9-0.1)*T;%целевой выход
    bpn=feedforwardnet([5 1],'trainrp');
    bpn=init(bpn);
    bpn.trainParam.epochs=100;
    bpn=train(bpn,P,toutn);
    P6 = randn(2,nn);
    P6(1,:)=P6(1,:).*1.4;
    P6(2,:)=P6(2,:).*1.1+2.5+(20/8);
    P=P6;
    a=sim(bpn,P);
    for j=1:nn
        if a(j)>=0.5 a(j)=1; else a(j)=0; end; % округление полученных
результатов
    end;
    OC11(nn/10-1)=1-(sum(a)/(nn));
    P1=randn(2,nn);
    P1(1,:)=P1(1,:).*2.4;

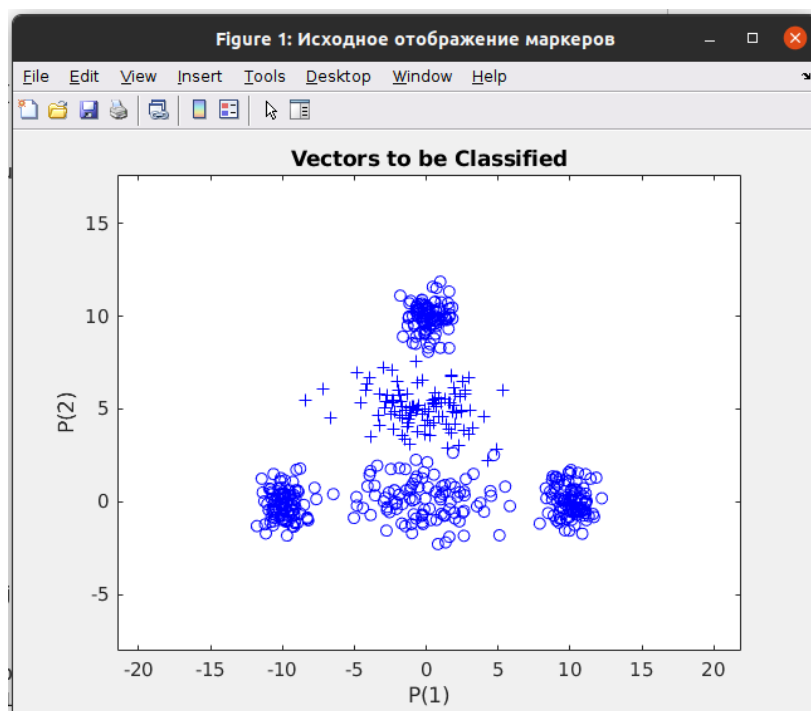
```

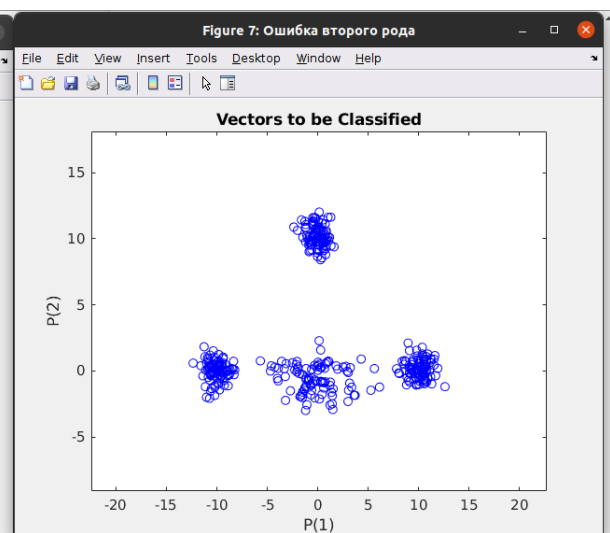
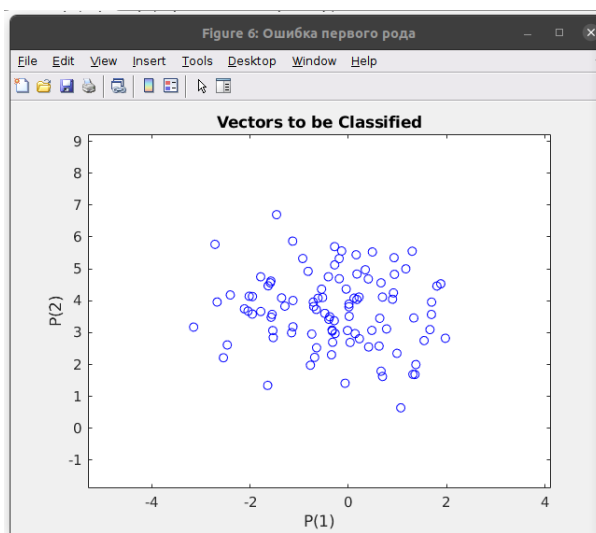
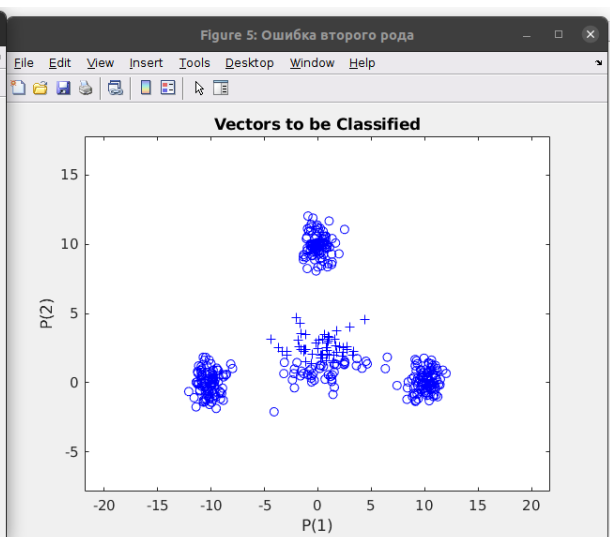
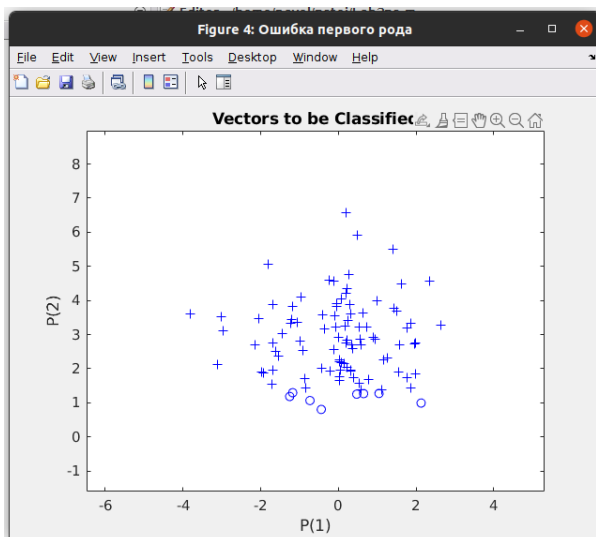
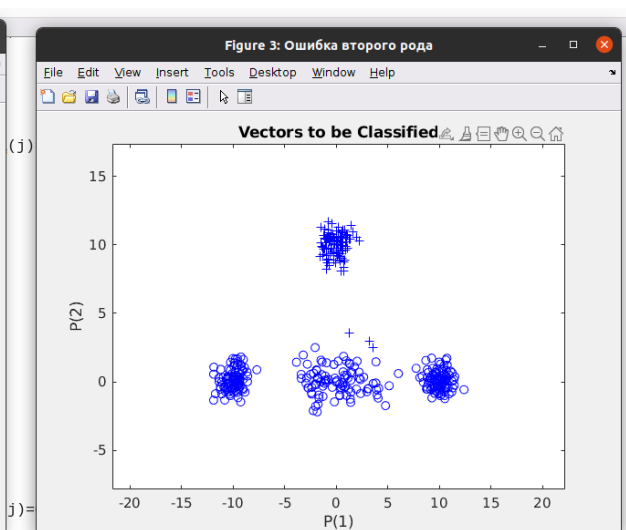
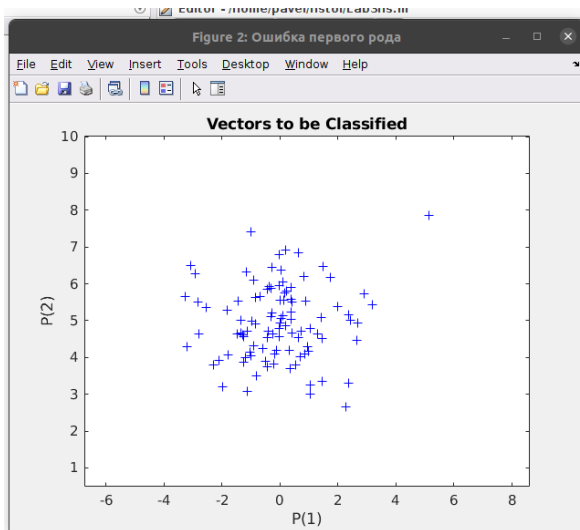
```

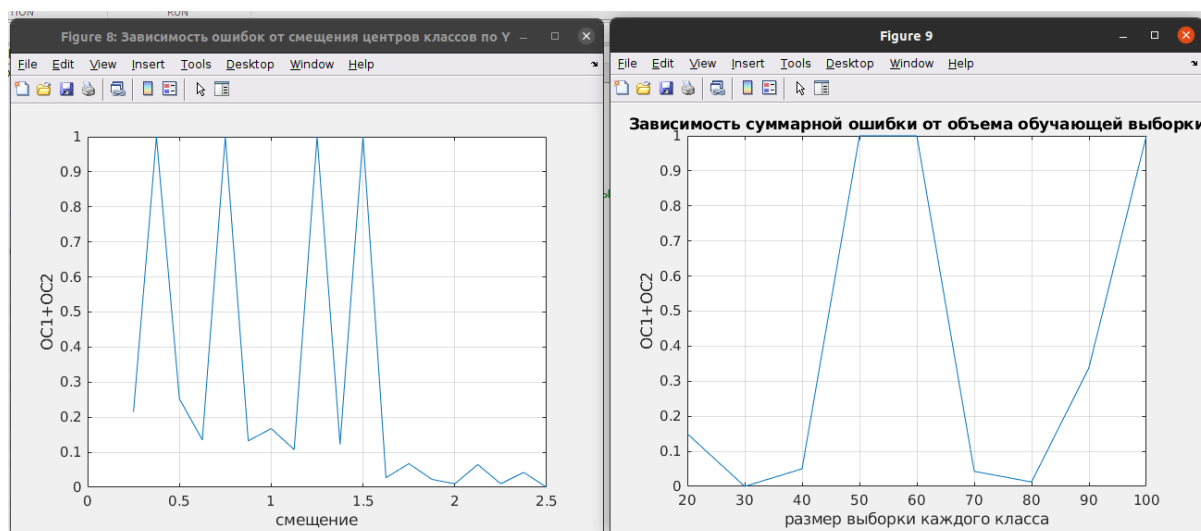
P1(2,:)=P1(2,:).*1.1+2-(20/8);
P2=randn(2,nn);
P2=P2.*0.8-a11;
P3=randn(2,nn);
P3=P3.*0.8+a11;
P4=randn(2,nn);
P4=P4.*0.8+a22;
P=[P1 P2 P3 P4];
b=sim(bpn,P);
for j=1:4*nn,
    if b(j)>0.5 b(j)=1; else b(j)=0; end;
end;
OC22(nn/10-1)=sum(b)/(4*nn);
M2(1,(nn/10)-1)=OC11(nn/10-1)+OC22(nn/10-1);
end
figure();
plot(N,M2),
title('Зависимость суммарной ошибки от объема обучающей выборки'),
xlabel('размер выборки каждого класса'),
ylabel('OC1+OC2'),
grid on;

```

Результаты выполнения задания







Вывод

1. Было проведено обучение нейросетевого классификатора для разделения гауссовских случайных векторов.
2. Были оценены ошибки первого и второго рода, а также суммарная ошибка.
3. Были построены графики зависимости суммарной ошибки от величины смещения центров кластеров и от объема обучающей выборки.