```
clear all; close all;
p0 = [1,1,1,1,1;1,0,0,0,0;1,1,1,1,0;1,0,0,0,0;1,0,0,0,0;]
p0 = p0(:);
p1 = [1,1,1,1,1;1,0,0,0,0;1,0,0,1,1;1,0,0,0,1;1,1,1,1,1;]
p1 = p1(:);
p2 = [1,0,0,0,1;1,0,0,0,1;1,1,1,1,1;1,0,0,0,1;1,0,0,0,1;]
p2 = p2(:);
p3 = [0,0,1,0,0;0,0,1,0,0;0,0,1,0,0;0,0,1,0,0;0,0,1,0,0;]
p3 = p3(:);
p4 = [1,0,0,0,0;1,0,0,0,0;1,0,0,0,0;1,0,0,0,0;1,1,1,1,1;]
p4 = p4(:);
p5 = [1,1,1,1,1;1,0,0,0,1;1,0,0,0,1;1,0,0,0,1;1,1,1,1,1;]
p5 = p5(:);
p6 = [1,1,1,1,1;1,0,0,0,1;1,1,1,1,1;1,0,0,0,0;1,0,0,0,0;]
p6 = p6(:);
p7 = [1,1,1,1,1;1,0,0,0,0;1,1,1,1,1;0,0,0,0,1;1,1,1,1,1;]
p7 = p7(:);
p8 = [1,1,1,1,1;0,0,1,0,0;0,0,1,0,0;0,0,1,0,0;0,0,1,0,0;]
p8 = p8(:);
p9 = [1,1,1,1,1;0,0,1,0,0;0,0,1,0,0;0,0,1,0,0;1,1,1,0,0;]
p9 = p9(:);

p = [p1, p2, p3, p4, p5, p6, p7, p8, p9, p0;]

[M, N] = size(p);
t = eye(10);

net = newff(p,t,9);

%training parameters%
net.trainParam.epochs = 2000;
net.divideParam.trainRatio = 1;
net.divideParam.valRatio = 0;
net.divideParam.testRatio = 0;

for aux = 1: net.numLayers
    net.layers{aux}.transferFcn = 'tansig';
end

net.trainFcn = 'trainscg'; %'traingda';

net = train(net,p,t);

inputs = p;
saida = sim(net,inputs)

saida_final = zeros(10,10);
for i = 1:10
    saida_final(saida(:,i) == max(saida(:,i)),i) = 1;
end

plotconfusion(t , saida_final)
[c, cm] = confusion(t, saida_final)

fprintf('Percentagem de classificação correta : %f%%\n', 100*(1-c));
```

```
fprintf('Percentagem de classificação incorreta : %f%%\n', 100*c);

input_final = [];

for j=1:10
    for i=1:M
        input_error(:,1) = p(:,j);
        input_error(i,1) = ~p(i,j);
        input_final = [input_final input_error];
    end
end

[lin, col] = size(p);

%sem erros%
inputs = p;
mat_erro = [];
mat_target = [];

%com 1 erro%
for j=1:col
    for i = 1 : lin
        erro = p(:,j);
        erro(i) = ~erro(i);
        mat_erro = [mat_erro erro];
        mat_target = [mat_target t(:,j)];
    end
end

output = sim(net, mat_erro);

%converter a saida para 0's e 1's%

final_output = round(output);
figure();
plotconfusion(mat_target,final_output);

[c, cm] = confusion(mat_target,final_output);

fprintf('Percentagem de classificação correta com 1 erro: %f%%\n',
100*(1-c));
fprintf('Percentagem de classificação incorreta com 1 erro: %f%%\n',
100*c);


%Gerar todas com 2 erros%
mat_erro2 = [];
mat_target2 = [];
for j = 1 : col
    for i = 1 : lin-1
        for k = i+1 :lin
            erro2 = p(:,j);
            erro2(i) = ~erro2(i);
            erro2(k) = ~erro2(k);
            mat_erro2 = [mat_erro2 erro2];
```

```
            mat_target2 = [mat_target2 t(:,j)];
        end
    end
end


%simular para todas as saidas geradas%
output2 = sim (net, mat_erro2);

%formar as saidas a 0 ou 1%
final_output2 = round(output2);
%aplicar o confusion%
figure();
plotconfusion(mat_target2,final_output2);
[c, cm] = confusion(mat_target2,final_output2);

fprintf('Percentagem de classificação correta com 2 erros: %f%%\n',
100*(1-c));
fprintf('Percentagem de classificação incorreta com 2 erros: %f%%\n',
100*c);
```