



**INSTITUTO FEDERAL DO CEARÁ – IFCE, CAMPUS  
MARACANAÚ  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

**RAUL GABRIEL CARVALHO DE MELO  
20142045050202**

**IMPLEMENTAÇÃO DE UMA CAMADA**

Trabalho apresentado como requisito parcial para aprovação  
na disciplina de Redes Neurais Artificiais da Instituto  
Federal do Ceará, campus Maracanaú.

Prof. Dr. Ajalmar Rego da Rocha Neto

# 1 Desenvolvimento

Foram criadas as seguintes classes:

## 1.1 Classe principal (dados artificiais/dados da iris)

Na classe principal dos dados artificiais, bem como na classe com os dados da iris, as variáveis de controle são iniciadas e as outras funções necessárias ao código são chamadas. O código está executando 20 realizações com 350 épocas e com uma taxa de aprendizagem de 0,1.

```
%% VARIÁVEIS INICIAIS
% NÚMERO DE REALIZAÇÕES, TAXA DE APRENDIZAGEM E NÚMERO DE ÉPOCAS
realizacoes = 20;
eta = 0.1;
n_epocas = 350;
```

Figura 1: Dados iniciais

## 1.2 Classe de treino

Na classe de treino temos duas funções de ativação, a função logística e a função tangente hiperbólica. Na função logística o limiar é 0.5, ou seja, valores menores que 0.5 são jogados para 0 e valor maior que o limiar é jogado para 1. Na função tangente hiperbólica o processo é parecido, sendo que o limiar é 0.

É calculado na classe treino um W diferente para cada uma das funções na mesma chamada, como mostra a figura 2.

```
for k = 1 : n_padroes
    % COMPUTA A SAIDA USANDO A FUNÇÃO LOGÍSTICA
    Y = logsig(W * X_treino(k,:)');
    % COMPUTA A SAIDA USANDO A FUNÇÃO TANGENTE HIPER.
    Y1 = tanh(W1 * X_treino(k,:)');

    % CALCULA O ERRO PARA OS 3 PERCEPTRONS
    err = Y_treino(k,:) - Y';
    err1 = Y_treino(k,:) - Y1';

    % ATUALIZA OS VETORES W's
    W = W + (taxa_ap * err' * Y'*(1-Y)*X_treino(k,:));
    W1 = W1 + (taxa_ap * err1'*0.5*[(1-Y')*(1-Y)]*X_treino(k,:));
end
```

Figura 2: Atualização dos pesos

### 1.3 Classe de teste

Na classe teste é computado o número de acertos para cada realização e é feito uma média do percentual de acertos. Para o correto cálculo do erro é necessário jogar o maior valor do vetor de erros para 1 e o resto para 0, pois em alguns casos existem mais de um valor maior do que o limiar de ativação da função.

```
for i = 1 : n_patterns
    % COMPUTA A SAIDA USANDO A FUNÇÃO LOGÍSTICA
    Y(:,i) = logsig(W * X_teste(i,:));

    % COMPUTA A SAIDA USANDO A FUNÇÃO TANGH
    Y1(:,i) = tanh(W1 * X_teste(i,:));

    % BINARIZA O VETOR, MAIOR FICA 1 E O RESTO 0
    Y(:,i) = binariza(Y(:,i));

    % BINARIZA O VETOR, MAIOR FICA 1 E O RESTO 0
    Y1(:,i) = binariza(Y1(:,i));

    % CONTA OS ACERTOS LOGÍSTICA
    count = count + isequal(Y(:,i)', Y_teste(i,:));
    % CONTA OS ACERTOS TANH
    count1 = count1 + isequal(Y1(:,i)', Y_teste(i,:));
end
```

Figura 3: Cálculo dos acertos

### 1.4 Gráfico gerado para os dados artificiais

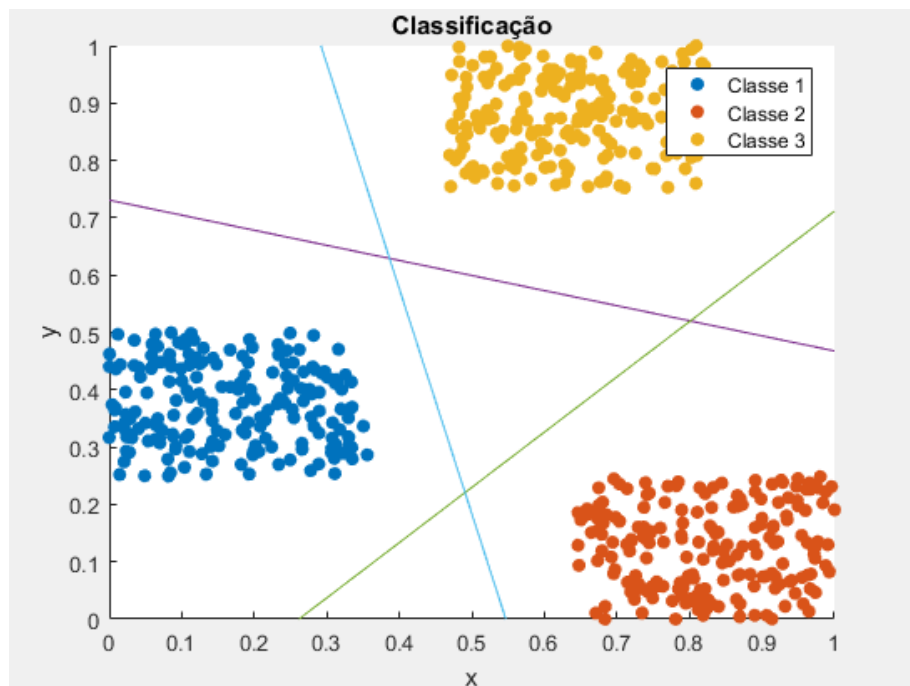


Figura 4: Classificação

## 2 Resultados

DADOS ARTIFICIAIS		
	LOGÍSTICA	TANGENTE HIP.
ACURÁCIA MÉDIA	100%	100%
MENOR ACURÁCIA	100%	100%
MAIOR ACURÁCIA	100%	100%
DESVIO PADRÃO	0	0

DADOS DA IRIS		
	LOGÍSTICA	TANGENTE HIP.
ACURÁCIA MÉDIA	97.50%	94.50%
MENOR ACURÁCIA	93.33%	80.00%
MAIOR ACURÁCIA	100%	100.00%
DESVIO PADRÃO	2.39	4.36