

Trabalho de Sistemas Inteligentes

Nome: Eros Batista Monte RA: 006208

O código usa uma rede neural para prever o valor futuro (usando regressão) de uma ação com base em dados de entrada.

```
function y = sigmoid(x)
    y = 1 ./ (1 + exp(-x));
end
inputSize = 4;           % Número de características de entrada
hiddenSize = 10;         % Número de neurônios na camada oculta
outputSize = 1;          % Número de neurônios na camada de saída
learningRate = 0.01;     % Taxa de aprendizado
numEpochs = 1000;       % Número de épocas de treinamento
% Dados de treinamento (exemplo realista)
X_train = [150, 500000, 1000000000, 0.8; % Amostra 1
           160, 600000, 1100000000, 0.7; % Amostra 2
           155, 550000, 1050000000, 0.9; % Amostra 3
           ];
y_train = [155.2; % Valor alvo correspondente à amostra 1
           161.8; % Valor alvo correspondente à amostra 2
           158.5; % Valor alvo correspondente à amostra 3
           ];
% Inicialização dos pesos
W1 = rand(hiddenSize, inputSize); % Pesos da camada oculta
W2 = rand(outputSize, hiddenSize); % Pesos da camada de saída
% Treinamento da rede neural
for epoch = 1:numEpochs
    % Propagação direta
    hiddenLayer = sigmoid(X_train * W1');
    outputLayer = hiddenLayer * W2';
    % Cálculo da função de perda
    loss = mean((outputLayer - y_train).^2);
    % Retropropagação
    deltaOutput = outputLayer - y_train;
    deltaHidden = (deltaOutput * W2) .* sigmoid(hiddenLayer) .* (1 - sigmoid(hiddenLayer));
    % Atualização dos pesos
    W2 = W2 - learningRate * hiddenLayer' * deltaOutput;
    W1 = W1 - learningRate * deltaHidden * X_train;
end
% Dados de teste (Números de exemplo)
X_test = [170, 700000, 1200000000, 0.6]; % Amostra de teste
% Teste da rede neural e exibição dos resultados
hiddenLayer_test = sigmoid(X_test * W1');
outputLayer_test = hiddenLayer_test * W2';
disp('Previsões da rede neural:');
disp(outputLayer_test);
```