**Assignment 5**

**Experiment 11**

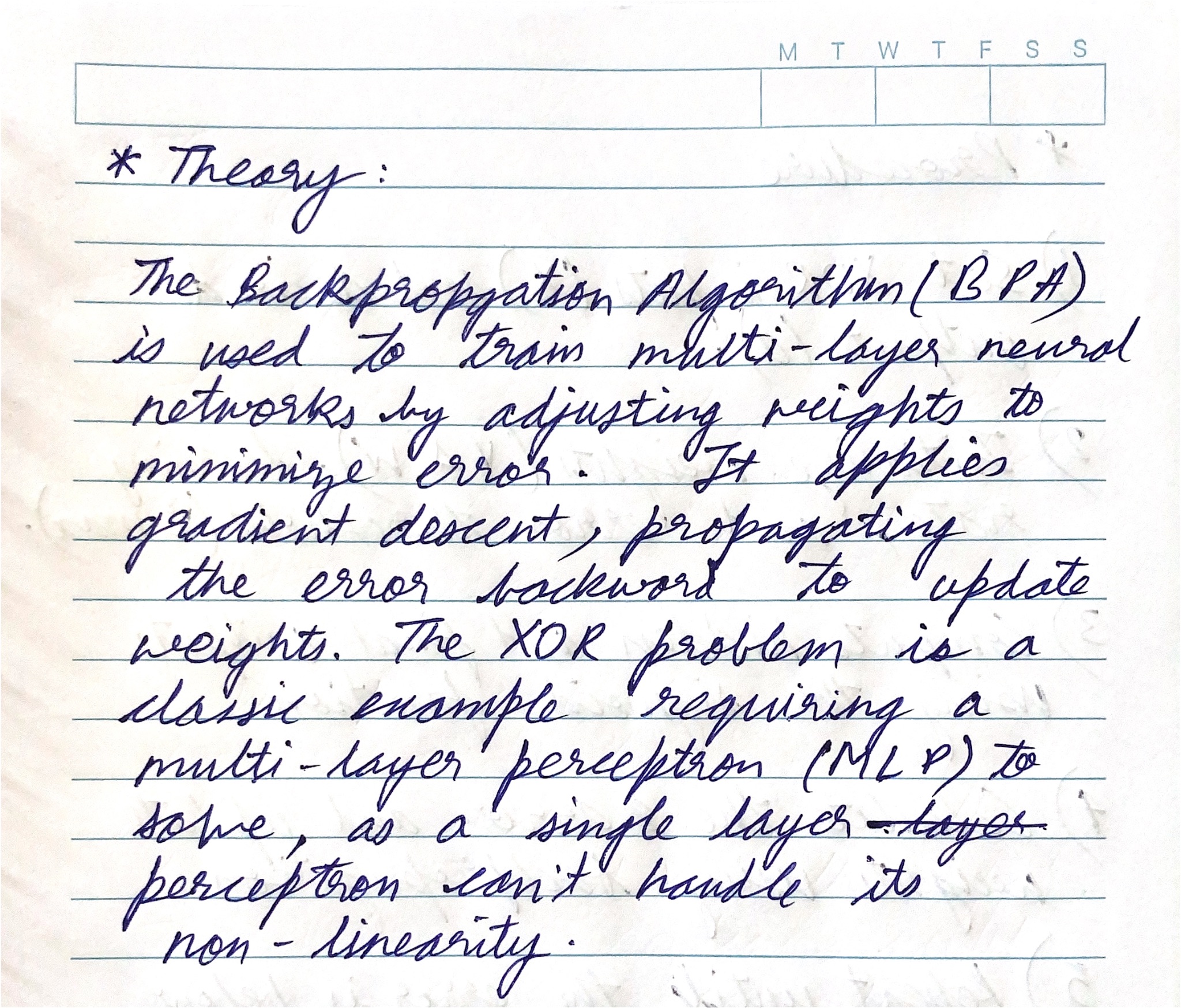
Title: Implement BPA to train XOR problem

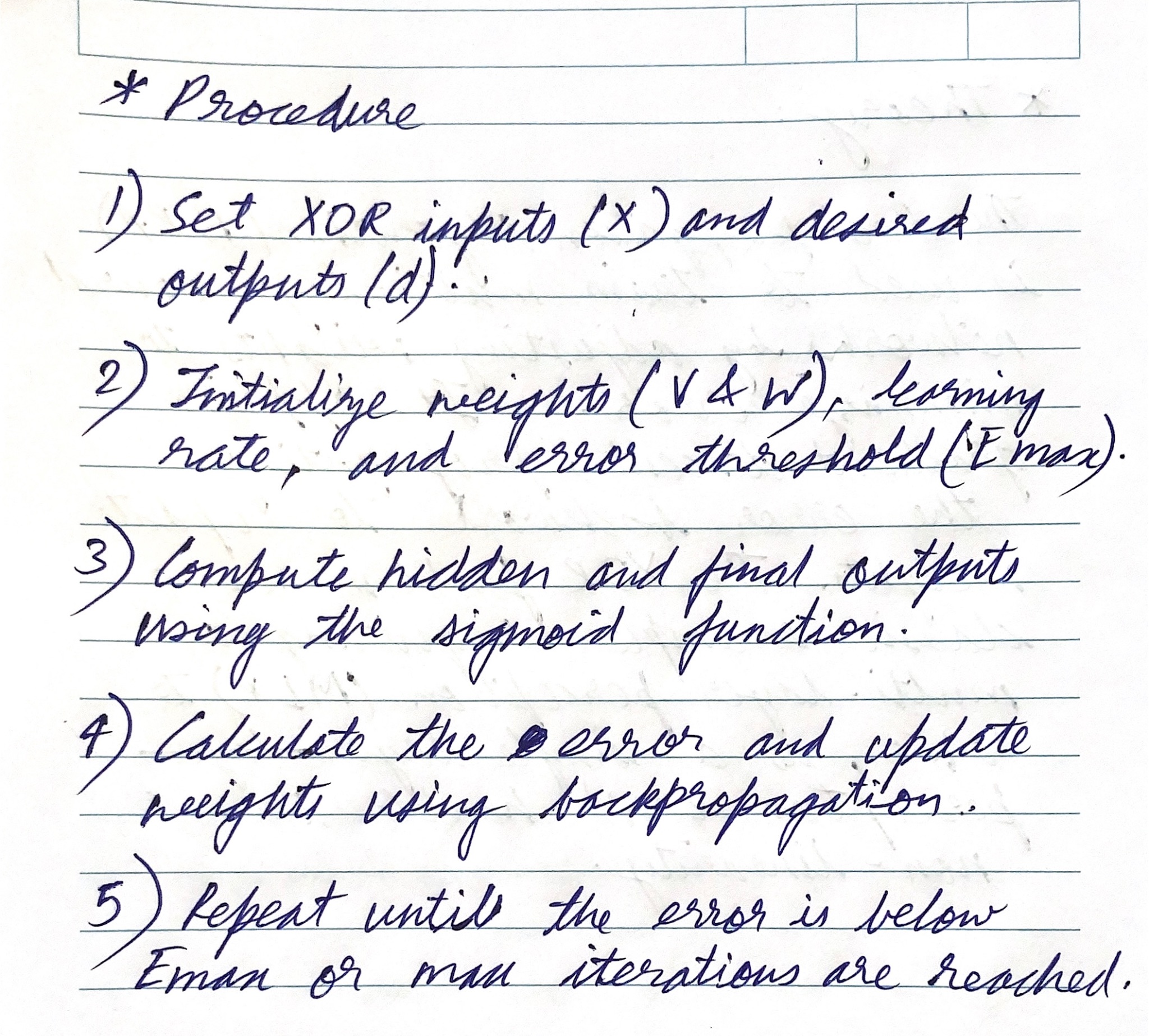
Name of Student: Sangeet Agrawal PRN No. 21070122140

DoP: 7 Oct DoS: 9 Oct

**Aim:**To design and simulate a Backpropagation training algorithm for multi-layer continuous perception.

**Problem Statement:** Implement BPA to train XOR problem





**Inputs:**

X = [0 0; 0 1; 1 0; 1 1];

d = [0; 1; 1; 0];

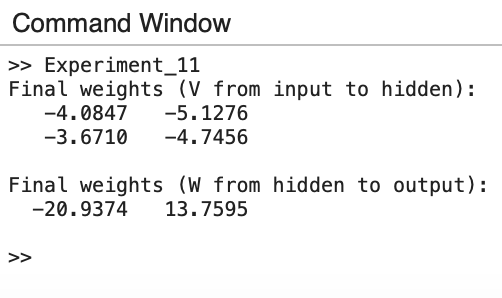
V = [0.5 0.5; 1 0.3];

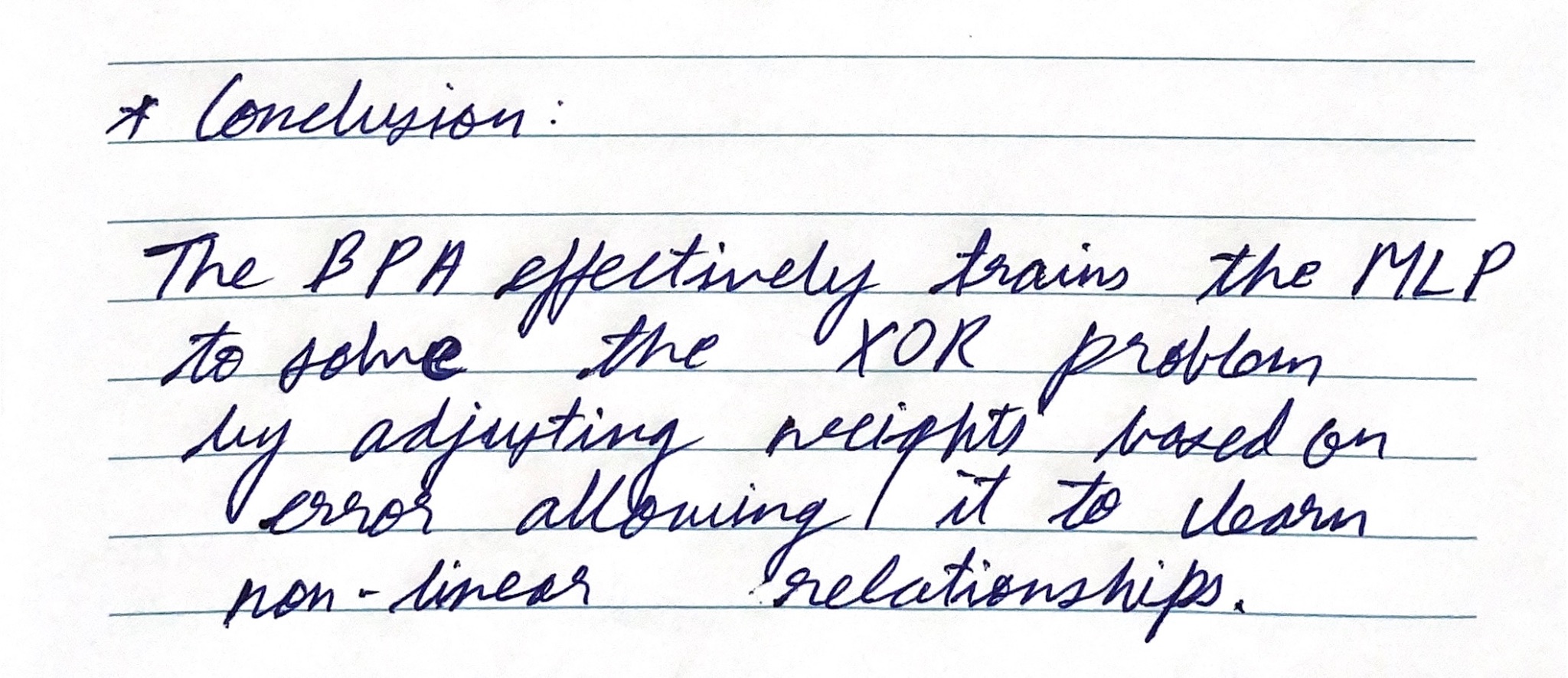
W = [0.5 0.5];

**Code:**

| X = [0 0; 0 1; 1 0; 1 1];  d = [0; 1; 1; 0];  V = [0.5 0.5; 1 0.3];  W = [0.5 0.5];  Emax = 0.1;  learning\_rate = 1;  max\_iters = 10000;  % Activation function and its derivative  f = @(x) 1 ./ (1 + exp(-x));  f\_prime = @(x) f(x) .\* (1 - f(x));    for iter = 1:max\_iters  total\_error = 0;    for p = 1:size(X, 1)  x = X(p, :)';  y = f(V \* x);  o = f(W \* y);    dk = d(p);  ok = o;  e = 0.5 \* (dk - ok)^2;  total\_error = total\_error + e;      dO = (dk - ok) \* f\_prime(ok);  dY = (W' \* dO) .\* f\_prime(y);    % Update weights  W = W + learning\_rate \* dO \* y'; % Update weights from hidden to output layer  V = V + learning\_rate \* dY \* x';  end    if total\_error < Emax  fprintf('Training converged at iteration %d with total error %f\n', iter, total\_error);  break;  end  end  disp('Final weights (V from input to hidden):');  disp(V);  disp('Final weights (W from hidden to output):');  disp(W); |
| --- |

**Output:**

****

****