

```

                                offlineBackpropagation.m
function back = offlineBackpropagation()
    numofInputs = 2;
    numofNeurons = 2;
    maxError = 0.00001;

    myInput = [0 0; 1 0; 0 1; 1 1];
    myOutput = [0 1 1 0];

    w1 = rand(numofInputs+1,numofNeurons);
    w2 = rand(numofNeurons+1,1);

    o = zeros(1,numofInputs+1);
    o(1,numofInputs+1) = 1;
    o1 = zeros(1,numofNeurons+1);
    o1(1,numofNeurons+1) = 1;

    gamma = 0.5;
    time = 0;
    page_output_immediately(1);

    do
        error = 0;
        deltaw1 = zeros(numofInputs+1, numofNeurons);
        deltaw2 = zeros(numofNeurons+1,1);
        for i=1:size(myInput,1)
            # define some strange thing written in the pdf ;)
            o(1,1:numofInputs) = myInput(i,1:2);
            o1(1,1:2) = sigmoid(o*w1);
            o2 = sigmoid(o1*w2);
            d2 = o2 * (1-o2);
            d1(1,1) = o1(1,1) * (1-o1(1,1));
            d1(2,2) = o1(1,2) * (1-o1(1,2));

            # correct weight
            e = o2 - myOutput(1,i);
            error = max(error,abs(myOutput(1,i)-o2));
            delta2 = d2*e;
            delta1 = d1 * w2(1:2, 1) * delta2;
            deltaw2 += (-gamma*delta2*o1)';
            deltaw1 += (-gamma*delta1*o)';
        endfor

        w1 += deltaw1/size(myInput,1);
        w2 += deltaw2/size(myInput,1);

        time++;
        if (mod(time,1000) == 0)
            error
        endif
        if (mod(time,2000) == 0 && error < 0.3)
            gamma*=2
        endif

    until (error< maxError)

    w1
    w2

endfunction

function sig = sigmoid(x)
    c = 1;
    for i=1:size(x,1)
        for j=1:size(x,2)
            sig(i,j) = 1 / ( 1 + exp(-c*x(i,j)));
        endfor
    endfor
endfunction

```