Colton Cunov
Artificial Neural Networks
B. Mehlig

# Linear separability of 4-dimensional Boolean functions (2020)

```
X = readmatrix('input_data_numeric.csv');
X(:,1)=[];
boolean_functions = readmatrix('boolean_functions.txt');
t = boolean_functions(6,:);
n = .02;
W = -.2 + .4.*rand(1,4);
T = -1 + 2*rand;

converged = 0;
H = zeros(1,10^5);
for iRun = 1:10
    iLearned = 1;
    O = zeros(1,16);
    sigO = zeros(1,16);
    while converged == 0 && iLearned < 10^5
        % calc output for all patterns
        for mu=1:16
            dotProduct = dot(X(mu,:),W);
            O(mu) = tanh(dotProduct - T);
            if O(mu) >= 0
                sigO(mu) = 1;
            else
                sigO(mu) = -1;
            end
            H(iLearned) = H(iLearned) + (t(mu)-O(mu))^2;
        end
        H(iLearned) = H(iLearned)/2;
        % check for convergence
        if isequal(sigO,t)
            converged = 1;
            break
        end
        % pick random pattern and update weights, thresh
        muRand = randi(16,1);
        dotProduct = dot(X(muRand,:),W);
        gPrime = 1 - tanh(dotProduct-T)^2;
        gradient = gPrime * (t(muRand)-O(muRand));
        for i=1:4
            W(i) = W(i) + n*gradient*X(muRand,i);
        end
        T = T + -n*gradient;
        iLearned = iLearned + 1;
    end
end
H(H==0) = [];
converged
plot(H(1:end))
```