

Problem3 Assignment1 Isac Nordin

isacn

September 2021

1 code

```
1 %Main code
2 N = 200;
3 p = 45;
4 T = 2*10^5;
5 noise_beta = 2;
6
7 %for loop to average m1 to be implemented
8 mAverage = 0;
9 averageTrial = 100;
10 for i = 1:averageTrial
11     mAverage = mAverage + GetOrderParameter(N,p,T,noise_beta);
12 end
13 round(mAverage/averageTrial,3)
14
15 %gets orderparameter
16 function m1 = GetOrderParameter(N,p,T,noiseBeta)
17     m1 = 0;
18     patterns = sign(2*rand(p,N)-1);
19     weightMatrix = GetWeights(patterns);
20     x1 = patterns(1,:);
21     currentState = x1;
22     for iTrial=1:T
23         m1 = m1 + (1/N) * currentState*x1';
24         currentState=AsynchronousStochasticUpdate(currentState,weightMatrix,
25             noiseBeta);
26     end
27     m1=m1/T;
28 end
29
30 %aynchronous stochastic update of 1 bit
31 function newState = AsynchronousStochasticUpdate(currentState,weightMatrix,
32     noiseBeta)
33     nRand = floor(length(currentState)*rand+1); %update random bit
34     bn = weightMatrix(nRand,:)*currentState';
35     newState = currentState;
36     if rand <= Pb(bn,noiseBeta)
37         newState(nRand) = 1;
38     else
39         newState(nRand) = -1;
```

```

40     end
41 end
42
43 %stochastic update probability
44 function pB = Pb(bi,noise_beta)
45     pB=1/(1+exp(-2*noise_beta*bi));
46 end
47
48
49 %get WeightMatrix (P x N size)
50 function WeightMatrix = GetWeights(patterns)
51     Npatterns = size(patterns,1);
52     Nbits = size(patterns,2);
53     WeightMatrix = zeros(Nbits,Nbits);
54
55     for iPattern = 1:Npatterns
56         patternI = patterns(iPattern,:);
57         WeightMatrix = WeightMatrix+mtimes(patternI',patternI);
58     end
59     WeightMatrix = WeightMatrix/Nbits;
60
61 %modified hebb's rule
62 for iBits = 1:Nbits
63     WeightMatrix(iBits,iBits) = 0;
64 end
65 end
66
67
68 %sign(x) but if ==0 -> =1
69 function sgn = Sgn(x)
70     sgn = sign(x);
71     if sgn == 0
72         sgn = 1;
73     end
74 end

```