Sharan Rajendran

CU username: sharanr

ECE 8720, Takehome #3

1 Map 1: Three Regions

- Using 3 colors, how many solutions are there? You can work this out by hand. What is the total number of distinct solutions?

  There are six 'valid' solutions that satisfy the constraints of the problem. Since, the decision was made to use a hopfield net where each unit represents {Region,Color}, we have a Hopfield net with nine units, which results in 2^9 solutions.

- State the coloring problem constraints in terms of our problem and the constraints on regions and colors.

  The constraints for the problem are as follows:

  - Each region can have atmost only one color
  - Each color can be used to color atmost one region
  - All regions have to be colored
  - Adjacent regions cannot have the same color

- Determine a Hopfield unit meaning in order to map the recurrent network state, o, into a solution.

  Each hopfield unit represents a {Region,Color}. Hence, if a Region X is colored a Color Y, then the Hopfield unit {Region X, Color Y} is turned ON.

	RED	BLUE	GREEN
Region 1	1	0	0
Region 2	0	0	1
Region 3	0	1	0

Table 1: Valid Solution - Region 1=Red, Region 2=Green, Region 3=Blue

- How many units are required for your design? The set partitoning and TSP solutions we have explored should give you some insight here.

  There are nine hopfield units, one for each member of the set Region,Color

- Determine the component Ei to satisfy the coloring constraints, and thus E. Leave the influence of each component adjustable in E, i.e., E = A E1 + B E2 + ...

  The constraint energy is given by (Region represented by X and Y and Colors by i and j),

$$E_{Constraint} = E_1 + E_2 + E_2 \tag{1}$$

$$E_{Constraint} = -\frac{A}{2}\sum_{X}\sum_{i}\sum_{i!=j} O_{X,i}O_{X,j} - \frac{B}{2}\sum_{i}\sum_{X}\sum_{X!=Y} O_{X,i}O_{Y,i} - \frac{C}{2}(\sum_{X}\sum_{i} O_{X,i} - n)^2 \tag{2}$$

- Show values of your E for various hand-worked valid and invalid solutions Each 3x3 State is converted to a 9x1 vector by concatenating each column vertically

State Vector 9x1	Energy
1 0 0 1 0 0 0 0 0	-2.4
1 0 1 1 0 0 0 0 0	-2.4
0 0 1 1 0 0 0 0 0	-3.0
0 1 1 1 0 0 0 0 0	-3.0
0 0 0 1 0 0 0 0 1	-3.0

Table 2: Energy for Invalid States

State Vector 9x1	Energy
1 0 0 0 1 0 0 0 1	-3.6
0 0 1 0 1 0 1 0 0	-3.6
0 0 1 1 0 0 0 1 0	-3.6
0 1 0 1 0 0 0 0 1	-3.6
0 1 0 1 0 0 0 0 1	-3.6

Table 3: Energy for Valid States

- Determine and show the weight matrices and unit biases corresponding to each Ei and the overall W and bias values for the network.

Bias=1.8000 (for all units)

0	-0	-0	-0.6	-0	-0	-0.6	-0	-0
-0	0	-0	-0	-0.6	-0	-0	-0.6	-0
-0	-0	0	-0	-0	-0.6	-0	-0	-0.6
-0.6	-0	-0	0	-0	-0	-0.6	-0	-0
-0	-0.6	-0	-0	0	-0	-0	-0.6	-0
-0	-0	-0.6	-0	-0	0	-0	-0	-0.6
-0.6	-0	-0	-0.6	-0	-0	0	-0	-0
-0	-0.6	-0	-0	-0.6	-0	-0	0	-0
-0	-0	-0.6	-0	-0	-0.6	-0	-0	0

Table 4: Weight Contribution from Component E_1

0	-0.6	-0.6	-0	-0	-0	-0	-0	-0
-0.6	0	-0.6	-0	-0	-0	-0	-0	-0
-0.6	-0.6	0	-0	-0	-0	-0	-0	-0
-0	-0	-0	0	-0.6	-0.6	-0	-0	-0
-0	-0	-0	-0.6	0	-0.6	-0	-0	-0
-0	-0	-0	-0.6	-0.6	0	-0	-0	-0
-0	-0	-0	-0	-0	-0	0	-0.6	-0.6
-0	-0	-0	-0	-0	-0	-0.6	0	-0.6
-0	-0	-0	-0	-0	-0	-0.6	-0.6	0

Table 5: Weight Contribution from Component E_2

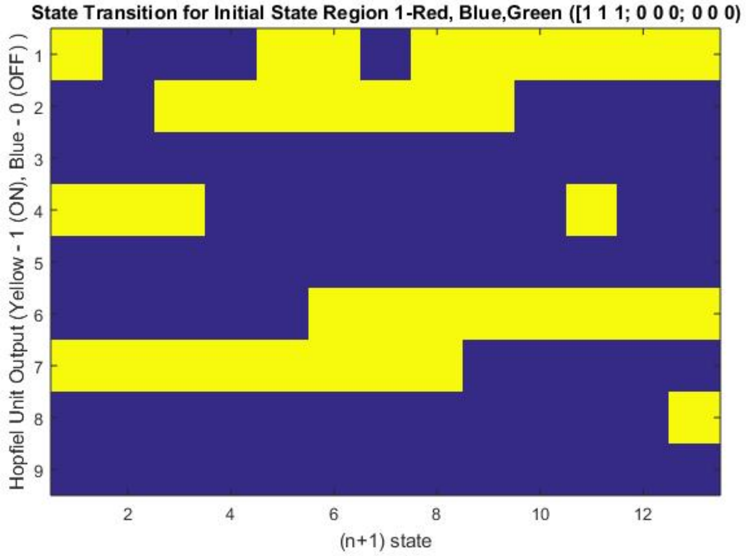
0	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6
-0.6	0	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6
-0.6	-0.6	0	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6
-0.6	-0.6	-0.6	0	-0.6	-0.6	-0.6	-0.6	-0.6
-0.6	-0.6	-0.6	-0.6	0	-0.6	-0.6	-0.6	-0.6
-0.6	-0.6	-0.6	-0.6	-0.6	0	-0.6	-0.6	-0.6
-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	0	-0.6	-0.6
-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	0	-0.6
-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	-0.6	0

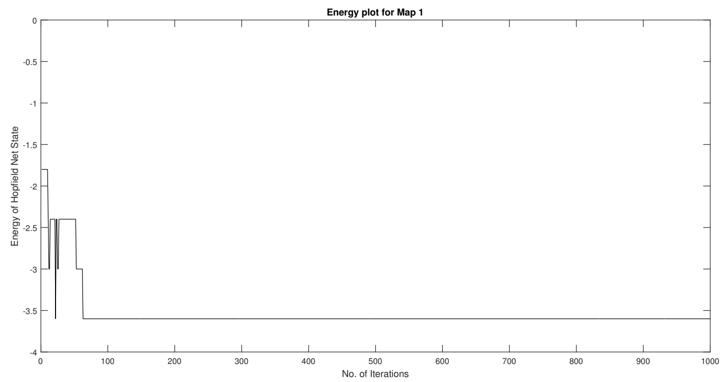
Table 6: Weight Contribution from Component E_1

0	-1.2	-1.2	-1.2	-0.6	-0.6	-1.2	-0.6	-0.6
-1.2	0	-1.2	-0.6	-1.2	-0.6	-0.6	-1.2	-0.6
-1.2	-1.2	0	-0.6	-0.6	-1.2	-0.6	-0.6	-1.2
-1.2	-0.6	-0.6	0	-1.2	-1.2	-1.2	-0.6	-0.6
-0.6	-1.2	-0.6	-1.2	0	-1.2	-0.6	-1.2	-0.6
-0.6	-0.6	-1.2	-1.2	-1.2	0	-0.6	-0.6	-1.2
-1.2	-0.6	-0.6	-1.2	-0.6	-0.6	0	-1.2	-1.2
-0.6	-1.2	-0.6	-0.6	-1.2	-0.6	-1.2	0	-1.2
-0.6	-0.6	-1.2	-0.6	-0.6	-1.2	-1.2	-1.2	0

Table 7: Weights of Overall Network

- Implement the network, start at randomly generated o(t0) and show the state evolution.
  $O(t_0)$=[1 1 1; 0 0 0; 0 0 0] - translates to Region 1 colored Red, Blue, Green.





- Show the effect of changing (tuning) the relative values of A, B, etc.
  - All the results shown have been determined using the A=B=C=0.6.
  - On changing A,B and C, it was observed that the Energy Value is scaled accordingly
  - Tuning A to a high value makes sure that each region is colored
  - Tuning B makes sure each color is used once
  - Tuning C ensures 3 units are always ON
- Does the state of your recurrent neural network converge to a valid coloring solution? Try many cases.
  The RNN converges to a valid coloring solution most of the time. Across 50 cases, the RNN converged to a valid solution atleast 20 times in 1000 iterations for the a common annealing schedule.
- Was the network able to converge to all coloring solutions for each map?
  Yes. All though it showed preference for a particular set of solutions most of the time.

2 Map 2: Four Regions

- Using 3 colors, how many solutions are there? You can work this out by hand. What is the total number of distinct solutions?

  There are twelve valid solutions and there are 2^{12} states.

- State the coloring problem constraints in terms of our problem and the constraints on regions and colors. The constraints for the problem are as follows:

  - Each region can have atmost only one color
  - Each color can be used to color atmost two regions
  - All regions have to be colored
  - Adjacent regions cannot have the same color

- Determine a Hopfield unit meaning in order to map the recurrent network state, o, into a solution.

  Each hopfield unit represents a {Region,Color}. Hence, if a Region X is colored a Color Y, then the Hopfield unit {Region X, Color Y} is turned ON. Hence, there are twelve units representing each member of the set.

	RED	BLUE	GREEN
Region 1	1	0	0
Region 2	0	1	0
Region 3	0	0	1
Region 4	1	0	0

Table 8: Valid Solution - Region 1=Red, Region 2=Green, Region 3=Blue, Region 4=Red

- How many units are required for your design? The set partitoning and TSP solutions we have explored should give you some insight here.

  There are twelve hopfield units, one for each member of the set {Region,Color}

- Determine the component Ei to satisfy the coloring constraints, and thus E. Leave the influence of each component adjustable in E, i.e., E = A E1 + B E2 + ...

  The constraint energy is given by (Region represented by X and Y and Colors by i and j,d_{XY} represents the adjaceny cost matrix),

$$E_{Constraint} = E_1 + E_2 + E_2 \tag{3}$$

$$E_{Constraint} = -\frac{A}{2}\sum_X \sum_i \sum_{i!=j} O_{X,i}O_{X,j} - \frac{B}{2}\sum_i \sum_X \sum_{X!=Y} d_{XY}O_{X,i}O_{Y,i} - \frac{C}{2}(\sum_X \sum_i O_{X,i} - n)^2 \tag{4}$$

- Show values of your E for various hand-worked valid and invalid solutions Each 4x3 State is converted to a 12x1 vector by concatenating each column vertically

State Vector 12x1	Energy
0 0 1 0 0 1 0 0 0 0 0 1	-5.4
0 0 1 0 0 1 0 0 0 0 1 1	-5.4
0 0 1 0 0 1 1 0 0 1 0 1	-3.6
0 0 1 0 0 0 1 0 0 1 0 1	-4.8
0 0 1 0 0 0 1 1 0 1 0 1	-4.2

Table 9: Energy for Invalid States

State Vector 12x1	Energy
0 0 1 1 1 0 0 0 0 1 0 0	-6.0
0 0 1 0 0 1 0 0 1 0 0 1	-6.0
0 1 0 0 0 0 1 1 1 0 0 0	-6.0
0 0 1 0 1 0 0 1 0 1 0 0	-6.0
1 0 0 1 0 1 0 0 0 0 1 0	-6.0

Table 10: Energy for Valid States

- Determine and show the weight matrices and unit biases corresponding to each Ei and the overall W and bias values for the network.

Bias=2.4000 (for all units)

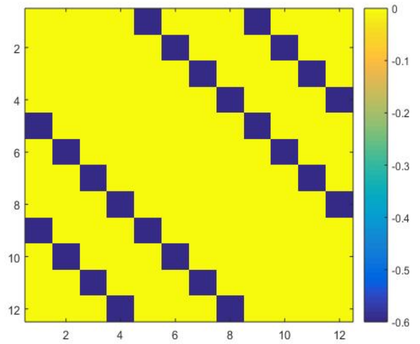


Table 11: Weight Contribution from Component E_1

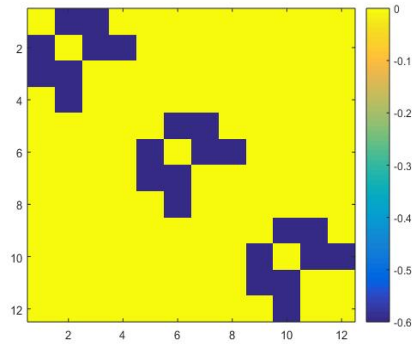


Table 12: Weight Contribution from Component E_2

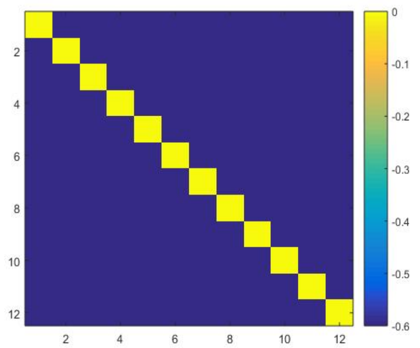


Table 13: Weight Contribution from Component E_1

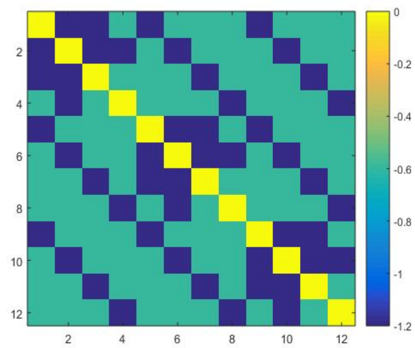
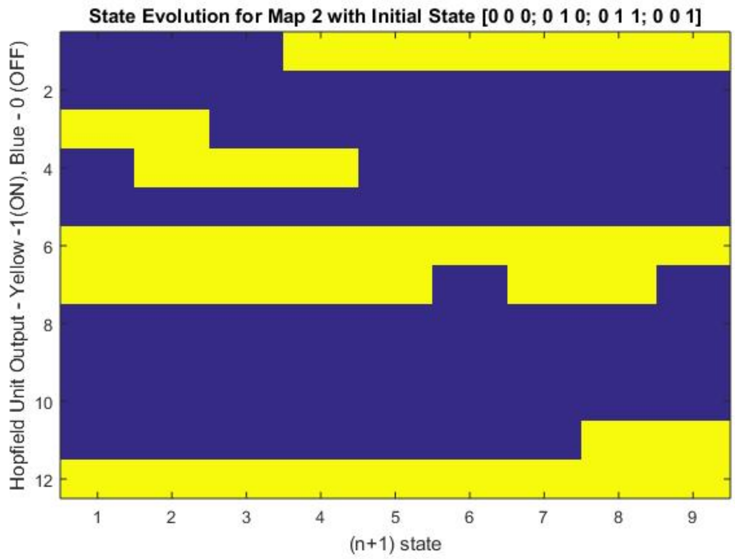
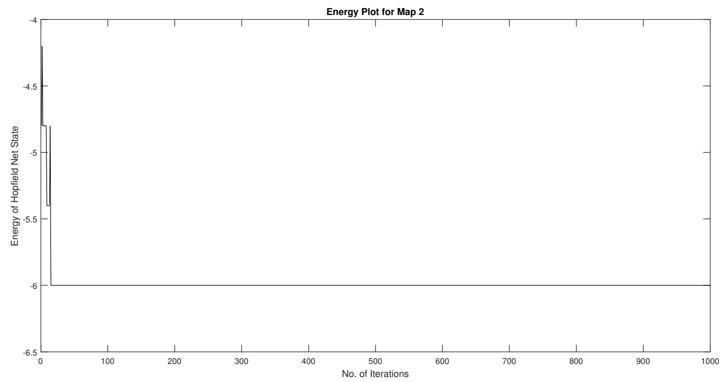


Table 14: Weights of Overall Network

- Implement the network, start at randomly generated o(t0) and show the state evolution.

  O(t0)=[0 0 0; 0 1 0; 0 1 1; 0 0 1] - translates to Region 2 colored blue, Region 3 colored blue and green, and Region 4 colored green.





- Show the effect of changing (tuning) the relative values of A, B, etc.
  - Results show using A=B=C=0.6 and Cost if regions are adjacent is 1 and 0 if not
  - Tuning A,B,C and Cost values changed the energy scale accordingly
  - Changing value of A changes the amount of importance given to coloring all region
  - Changing B ensures importance is given to using Colors atmost twice
  - Changin C varies the priority ensuring four units are always ON
- Does the state of your recurrent neural network converge to a valid coloring solution? Try many cases.

  Yes. About 80 cases managed to converge and cover all 12 valid coloring solutions.
- Was the network able to converge to all coloring solutions for each map?

  Yes.