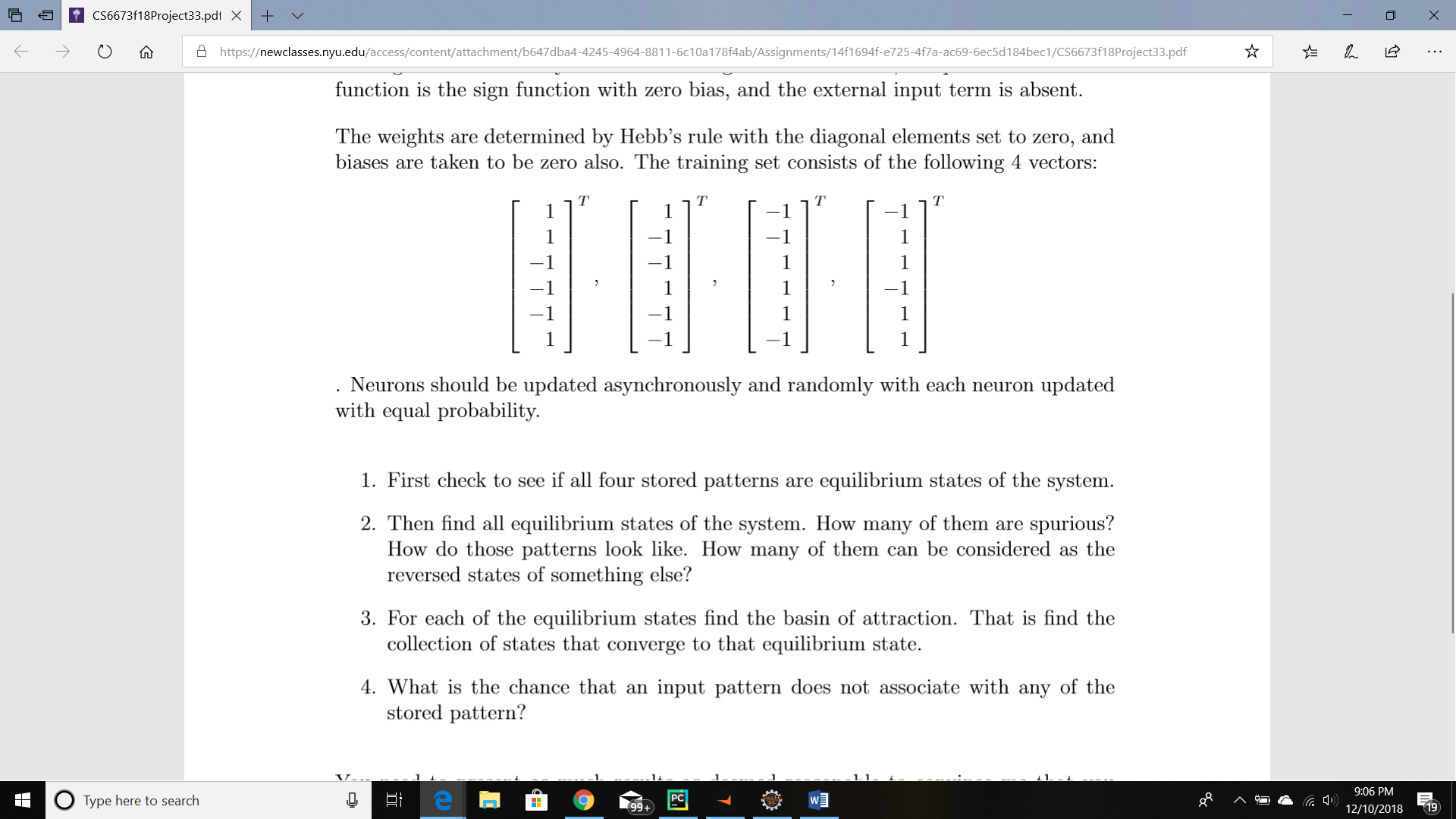
**Neural Networks**

Shrey Gupta

The given stored vectors are



These stored vectors are used to calculate the weights using the Hebb’s learning rule. The outputs are considered to be the stored patterns themselves. As a result, the stored vectors give the following weight matrix:

0 0 -4 0 -4 0

0 0 0 -4 0 4

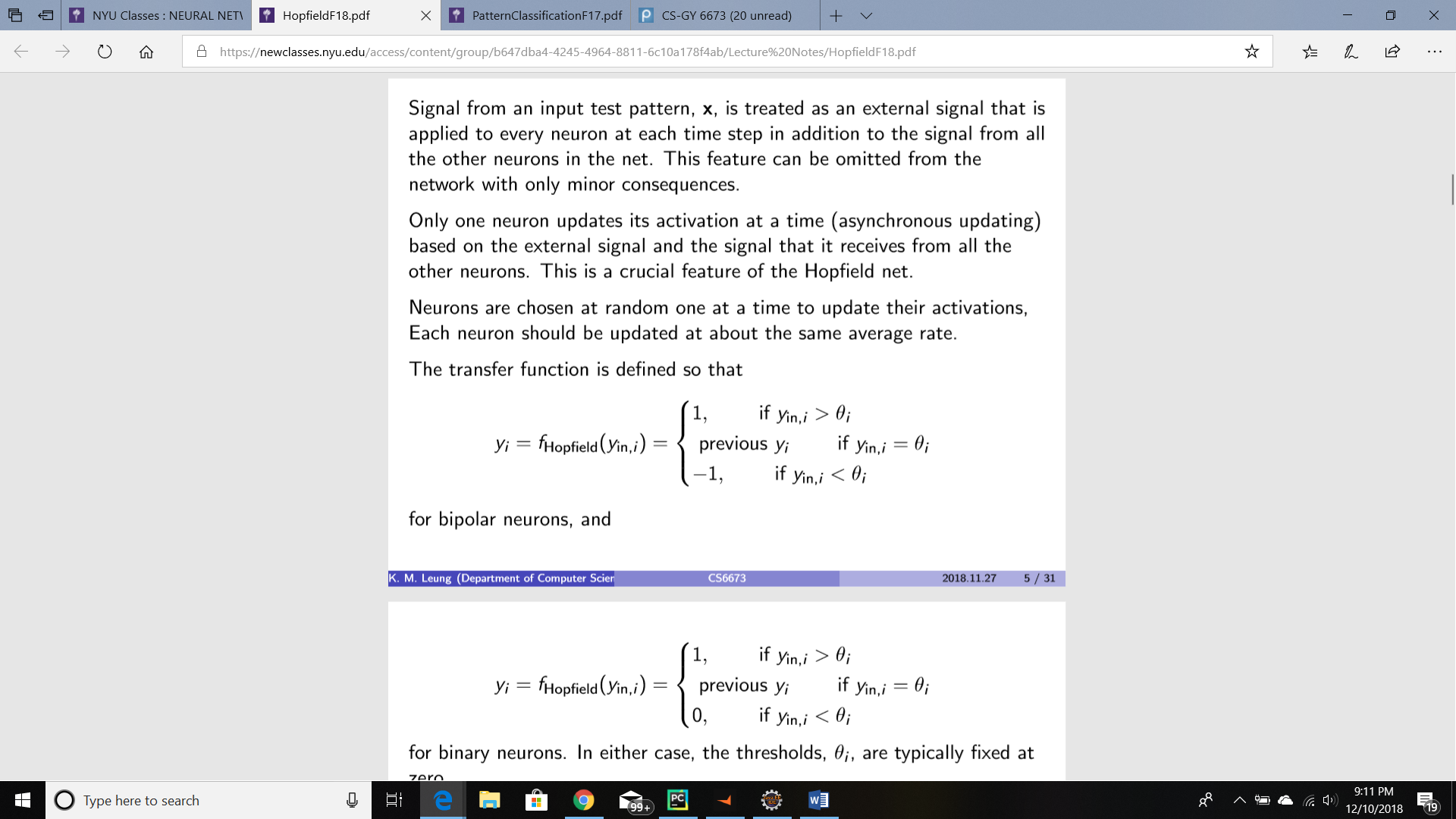
-4 0 0 0 4 0

0 -4 0 0 0 -4

-4 0 4 0 0 0

0 4 0 -4 0 0

The diagonals have been set to 0 according to the given Discrete Hopfield Net algorithm. The activation function is used for the bipolar neurons as the follows:



Where theta is kept as 0 in this case.

***NOTE:*** During the training of the model , the neuron ‘i’ is chosen at a random number when the changes in activation is performed. In this way a random neuron I is chosen and each neuron is given equal probability. The loop for updation of y goes on until it is made sure that all the neurons remain unchanged

**---Check if the four stored patterns are in equilibrium state or not**

A given test pattern specifies the state of the network. The activations are updated and after some time , the activations cease to change. This is the state of equilibrium. It is referred to as the net work output for that particular test pattern.   
The given 4 stored vectors will be in the equilibrium state, when we input each of the stored vectors into the model. The output state achieved after the activation cease to stop is the equilibrium state. If these outputs are exactly the same as the input stored patterns, then the stored patterns are called to be in equilibrium state. It is seen that the outputs from the model when the stored vectors are fed is same as the stored vector. Hence , they are in equilibrium state.

Hence according to the code output, it is seen that these 4 stored patterns are in equilibrium state.

**---Find all the equilibrium states of the system**

To find all the equilibrium states of the system, we need to find all the possible input test vectors that can be given to the system. Since the stored vector is of the size (1X6), the total possibility of unique test vectors are 26 which is 64. The input test vectors are given in the output of the code. Each of these vectors are fed into the model and the activations are updated. As soon as the activations cease to change, the output is stored into a separate array of equilibrium state. These equilibrium states are also presented in the output of the code. A total of 64 equilibrium states are produced.

The test input vectors used are total 64 :

[[1 1 1 1 1 1]]

[[-1 1 1 1 1 1]]

[[ 1 -1 1 1 1 1]]

[[ 1 1 -1 1 1 1]]

[[ 1 1 1 -1 1 1]]

[[ 1 1 1 1 -1 1]]

[[ 1 1 1 1 1 -1]]

[[-1 -1 1 1 1 1]]

[[-1 1 -1 1 1 1]]

[[-1 1 1 -1 1 1]]

[[-1 1 1 1 -1 1]]

[[-1 1 1 1 1 -1]]

[[ 1 -1 -1 1 1 1]]

[[ 1 -1 1 -1 1 1]]

[[ 1 -1 1 1 -1 1]]

[[ 1 -1 1 1 1 -1]]

[[ 1 1 -1 -1 1 1]]

[[ 1 1 -1 1 -1 1]]

[[ 1 1 -1 1 1 -1]]

[[ 1 1 1 -1 -1 1]]

[[ 1 1 1 -1 1 -1]]

[[ 1 1 1 1 -1 -1]]

[[-1 -1 -1 1 1 1]]

[[-1 -1 1 -1 1 1]]

[[-1 -1 1 1 -1 1]]

[[-1 -1 1 1 1 -1]]

[[-1 1 -1 -1 1 1]]

[[-1 1 -1 1 -1 1]]

[[-1 1 -1 1 1 -1]]

[[-1 1 1 -1 -1 1]]

[[-1 1 1 -1 1 -1]]

[[-1 1 1 1 -1 -1]]

[[ 1 -1 -1 -1 1 1]]

[[ 1 -1 -1 1 -1 1]]

[[ 1 -1 -1 1 1 -1]]

[[ 1 -1 1 -1 -1 1]]

[[ 1 -1 1 -1 1 -1]]

[[ 1 -1 1 1 -1 -1]]

[[ 1 1 -1 -1 -1 1]]

[[ 1 1 -1 -1 1 -1]]

[[ 1 1 -1 1 -1 -1]]

[[ 1 1 1 -1 -1 -1]]

[[-1 -1 -1 -1 1 1]]

[[-1 -1 -1 1 -1 1]]

[[-1 -1 -1 1 1 -1]]

[[-1 -1 1 -1 -1 1]]

[[-1 -1 1 -1 1 -1]]

[[-1 -1 1 1 -1 -1]]

[[-1 1 -1 -1 -1 1]]

[[-1 1 -1 -1 1 -1]]

[[-1 1 -1 1 -1 -1]]

[[-1 1 1 -1 -1 -1]]

[[ 1 -1 -1 -1 -1 1]]

[[ 1 -1 -1 -1 1 -1]]

[[ 1 -1 -1 1 -1 -1]]

[[ 1 -1 1 -1 -1 -1]]

[[ 1 1 -1 -1 -1 -1]]

[[-1 -1 -1 -1 -1 1]]

[[-1 -1 -1 -1 1 -1]]

[[-1 -1 -1 1 -1 -1]]

[[-1 -1 1 -1 -1 -1]]

[[-1 1 -1 -1 -1 -1]]

[[ 1 -1 -1 -1 -1 -1]]

[[-1 -1 -1 -1 -1 -1]]

The equilibrium states that were produced are:

[[-1 1 1 -1 1 1]]

[[-1 1 1 -1 1 1]]

[[-1 -1 1 1 1 -1]]

[[ 1 1 -1 -1 -1 1]]

[[-1 1 1 -1 1 1]]

[[ 1 1 -1 -1 -1 1]]

[[-1 -1 1 1 1 -1]]

[[-1 -1 1 1 1 -1]]

[[-1 1 1 -1 1 1]]

[[-1 1 1 -1 1 1]]

[[-1 1 1 -1 1 1]]

[[-1 -1 1 1 1 -1]]

[[ 1 -1 -1 1 -1 -1]]

[[-1 1 1 -1 1 1]]

[[ 1 -1 -1 1 -1 -1]]

[[-1 -1 1 1 1 -1]]

[[ 1 1 -1 -1 -1 1]]

[[ 1 1 -1 -1 -1 1]]

[[ 1 -1 -1 1 -1 -1]]

[[ 1 1 -1 -1 -1 1]]

[[-1 1 1 -1 1 1]]

[[ 1 -1 -1 1 -1 -1]]

[[-1 -1 1 1 1 -1]]

[[-1 1 1 -1 1 1]]

[[-1 -1 1 1 1 -1]]

[[-1 -1 1 1 1 -1]]

[[-1 1 1 -1 1 1]]

[[ 1 1 -1 -1 -1 1]]

[[-1 -1 1 1 1 -1]]

[[-1 1 1 -1 1 1]]

[[-1 1 1 -1 1 1]]

[[-1 -1 1 1 1 -1]]

[[ 1 1 -1 -1 -1 1]]

[[ 1 -1 -1 1 -1 -1]]

[[ 1 -1 -1 1 -1 -1]]

[[ 1 1 -1 -1 -1 1]]

[[-1 -1 1 1 1 -1]]

[[ 1 -1 -1 1 -1 -1]]

[[ 1 1 -1 -1 -1 1]]

[[ 1 1 -1 -1 -1 1]]

[[ 1 -1 -1 1 -1 -1]]

[[ 1 1 -1 -1 -1 1]]

[[-1 1 1 -1 1 1]]

[[ 1 -1 -1 1 -1 -1]]

[[-1 -1 1 1 1 -1]]

[[-1 1 1 -1 1 1]]

[[-1 -1 1 1 1 -1]]

[[-1 -1 1 1 1 -1]]

[[ 1 1 -1 -1 -1 1]]

[[-1 1 1 -1 1 1]]

[[ 1 -1 -1 1 -1 -1]]

[[-1 1 1 -1 1 1]]

[[ 1 1 -1 -1 -1 1]]

[[ 1 -1 -1 1 -1 -1]]

[[ 1 -1 -1 1 -1 -1]]

[[ 1 -1 -1 1 -1 -1]]

[[ 1 1 -1 -1 -1 1]]

[[ 1 1 -1 -1 -1 1]]

[[-1 -1 1 1 1 -1]]

[[ 1 -1 -1 1 -1 -1]]

[[-1 -1 1 1 1 -1]]

[[ 1 1 -1 -1 -1 1]]

[[ 1 -1 -1 1 -1 -1]]

[[ 1 -1 -1 1 -1 -1]]

The unique equilibrium states out of these 64 are found to be as follows:

[[-1 -1 1 1 1 -1]]

[[-1 1 1 -1 1 1]]

[[ 1 -1 -1 1 -1 -1]]

[[ 1 1 -1 -1 -1 1]]

**---How many of them are spurios?**

Spurios vectors are those vectors which do not converge to the stored vectors. We are given 4 stored vectors. On comparing each of these 64 equilibrium states to the stored vectors, it is found that none of these equilibrium states vector is different from the stored vectors.The comparison of equilibrium vector with the stored vectors is done on line 155. Each of the elements of the unique equilibrium state is matched with the stored vectors and output is generated accordingly. Each of these vectors is equal to either of the stored vectors , hence spurios vectors are found to be 0. If any spurios vector was found, the code has been given to print out those vectors also.

**---How do those patterns look like?**

If patterns are referred to the equilibrium states of the network, then it can be said that these equilibrium states look exactly like any one of the stored vectors. If patterns are referred to the spurios vectors, then it is seen that there are 0 spurios vectors.

**---How many of them can be considered to be reversed states of something?**

A state will be reverse of something if it is at a maximum Hamming distance from each other. In that case, each of the neuron is the negative of the other vector. Since there are no spurios states, there are 0 reversed states.Still, the algorithm and code for the same has been provided in the code, in case spurios vectors were present . The code for the same can be found on line 166 .

**---For each of the equilibrium, find the basin of attraction**

The basin of attraction is the collection of the states that will converge to that equilibrium state. To find out the basin of attraction we find out which of the input test vectors will converge to each equilibrium state seperately. The output of the input test vector is calculated by feeding the input into the model. The output is then checked with the equilibrium state to find whether it lies in it’s basin of attraction or not. The result for the basin of attraction is present in the output of the code:

Finding the Basin of attraction for all the equilibrium states

The size of basin of attraction of [[-1 -1 1 1 1 -1]] is 16 . Basin is =

[[ 1 -1 1 1 1 1]]

[[ 1 1 1 1 1 -1]]

[[-1 -1 1 1 1 1]]

[[-1 1 1 1 1 -1]]

[[ 1 -1 1 1 1 -1]]

[[-1 -1 -1 1 1 1]]

[[-1 -1 1 1 -1 1]]

[[-1 -1 1 1 1 -1]]

[[-1 1 -1 1 1 -1]]

[[-1 1 1 1 -1 -1]]

[[ 1 -1 1 -1 1 -1]]

[[-1 -1 -1 1 1 -1]]

[[-1 -1 1 -1 1 -1]]

[[-1 -1 1 1 -1 -1]]

[[-1 -1 -1 -1 1 -1]]

[[-1 -1 1 -1 -1 -1]]

The size of basin of attraction of [[-1 1 1 -1 1 1]] is 16 . Basin is =

[[1 1 1 1 1 1]]

[[-1 1 1 1 1 1]]

[[ 1 1 1 -1 1 1]]

[[-1 1 -1 1 1 1]]

[[-1 1 1 -1 1 1]]

[[-1 1 1 1 -1 1]]

[[ 1 -1 1 -1 1 1]]

[[ 1 1 1 -1 1 -1]]

[[-1 -1 1 -1 1 1]]

[[-1 1 -1 -1 1 1]]

[[-1 1 1 -1 -1 1]]

[[-1 1 1 -1 1 -1]]

[[-1 -1 -1 -1 1 1]]

[[-1 -1 1 -1 -1 1]]

[[-1 1 -1 -1 1 -1]]

[[-1 1 1 -1 -1 -1]]

The size of basin of attraction of [[ 1 -1 -1 1 -1 -1]] is 16 . Basin is =

[[ 1 -1 -1 1 1 1]]

[[ 1 -1 1 1 -1 1]]

[[ 1 1 -1 1 1 -1]]

[[ 1 1 1 1 -1 -1]]

[[ 1 -1 -1 1 -1 1]]

[[ 1 -1 -1 1 1 -1]]

[[ 1 -1 1 1 -1 -1]]

[[ 1 1 -1 1 -1 -1]]

[[-1 -1 -1 1 -1 1]]

[[-1 1 -1 1 -1 -1]]

[[ 1 -1 -1 -1 1 -1]]

[[ 1 -1 -1 1 -1 -1]]

[[ 1 -1 1 -1 -1 -1]]

[[-1 -1 -1 1 -1 -1]]

[[ 1 -1 -1 -1 -1 -1]]

[[-1 -1 -1 -1 -1 -1]]

The size of basin of attraction of [[ 1 1 -1 -1 -1 1]] is 16 . Basin is =

[[ 1 1 -1 1 1 1]]

[[ 1 1 1 1 -1 1]]

[[ 1 1 -1 -1 1 1]]

[[ 1 1 -1 1 -1 1]]

[[ 1 1 1 -1 -1 1]]

[[-1 1 -1 1 -1 1]]

[[ 1 -1 -1 -1 1 1]]

[[ 1 -1 1 -1 -1 1]]

[[ 1 1 -1 -1 -1 1]]

[[ 1 1 -1 -1 1 -1]]

[[ 1 1 1 -1 -1 -1]]

[[-1 1 -1 -1 -1 1]]

[[ 1 -1 -1 -1 -1 1]]

[[ 1 1 -1 -1 -1 -1]]

[[-1 -1 -1 -1 -1 1]]

[[-1 1 -1 -1 -1 -1]]

**---Chance that an input pattern does not associate with any of the stored pattern**

It is assumed that the chance is referred to as probability. Hence, the probability that a pattern does not associate with a stored pattern can be calculated by first finding out the total number of patterns that are not associated to the stored vectors. These are the input test vectors that do not converge to the stored vectors. Since there are total possible 64 input test vectors, the chance can be given as:

Chance = Number of test vectors not converging to stored vectors/Total input test vectors

0 is the number of vectors calculated which do not converge to the stored vector.

Hence chance = 0/64 = 0