ANALYSIS REPORT

Name: Kumar Raju Bandi

Roll Number: IWM2017502

In this assignment, Bidirectional Associative Memory(BAM) and Self Organizing Neural Network(SONN) were implemented from scratch and tested with the given input vectors.

Colab Link :
https://colab.research.google.com/drive/1UPCMhCI_McRHKI9tt1gXQRO0MPgVMSBI?usp=sharing

Bidirectional Associative Memory(BAM)

Bidirectional Associative Memory was implemented from scratch using the steps as discussed in the class. What I analysed is that the learning is quick as the time complexity for training is just O(n*m) where n is the dimension/size of input vector which has to be mapped and m is the dimension/size of the output vector.

It was also observed for that given test inputs, the outputs were correctly mapped and given back. Thus, resulting in 100% accuracy in mapping inputs to outputs.

I also tried the inverse that is asking the model to give the inputs for the outputs and interestingly these results were perfectly matching too. Thus, resulting in 100% accuracy in mapping outputs to inputs.

I also brought in unseen vector inputs and error correction and tried to observe the output and then feed back this output to get unseen input. This did not give the same result in the first epoch. However, by repeating this feedback loop and converging to give consistent results i.e input's output vector gives back the input vector.

The conclusion is that the BAM model showed unconditional stableness and that it is also capable of error correction.

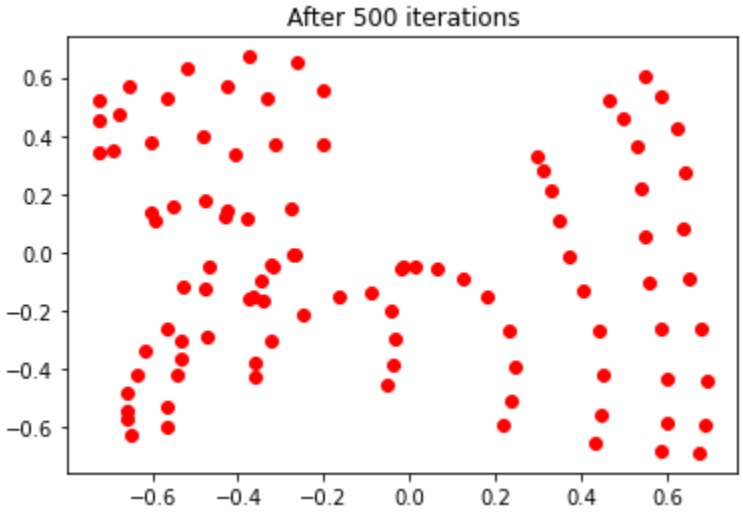
All the examples and results can be found in the implementation notebook.

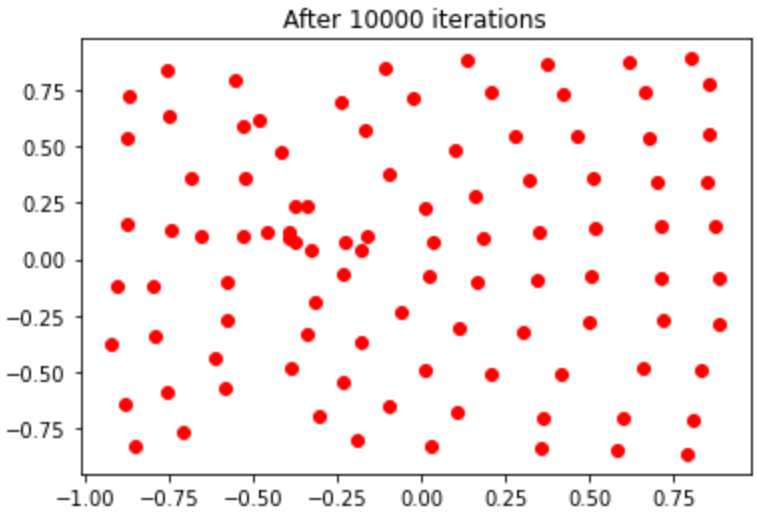
Self Organizing Neural Network(SONN)

Self Organizing Neural Network was implemented with the help of steps as discussed in the class.
The implementation was run for different epochs and different values of alpha.
Higher values of alpha made the network to become stable faster and for small values of alpha, varying the number of epochs to larger values showed that with more number of epochs, the learning got better and the network got more and more stable.

For this implementation, I had used 10x10 two-dimensional lattice of neurons for the Kohonen network.
I also created and ran several instances of the model so as to observe the interesting network shapes formed and convergence displayed over time.
Because, for input data, 1500 random vectors were used, the results were different for every run of the training

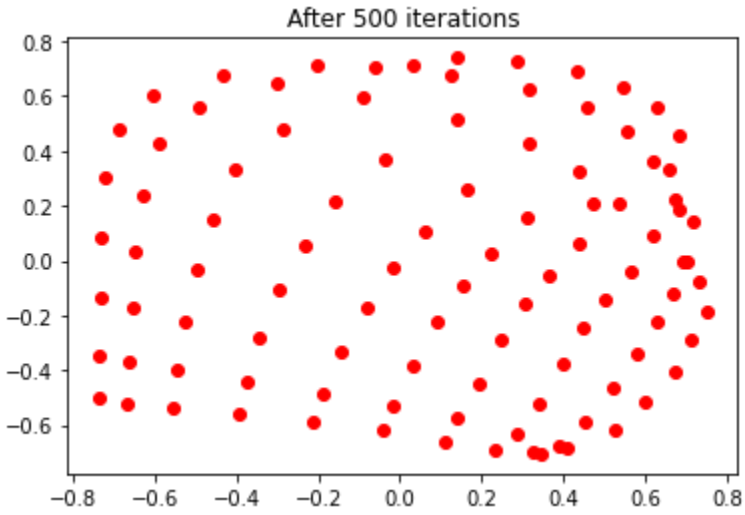
Some of the interesting results were as follows:
1.

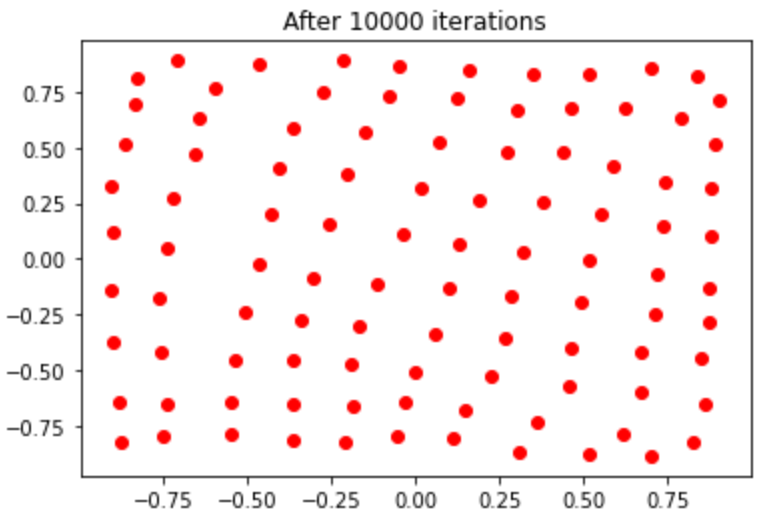




It is evident in the above plot that a lesser number of epochs did not let the network learn completely as seen in the first plot. In the second plot, with 10000 epochs, the network shaped and learnt more and showed stableness in the further epochs.

2. Here is another example





The results for the test inputs can be found in the implementation notebook.