Homework Assignment -2

ENPM 690 Robot Learning

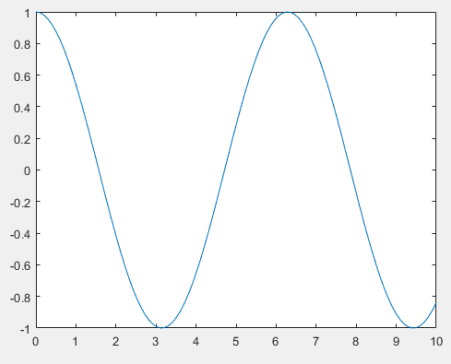
Name: - Achal Vyas UID: - 116869560

Q1) Program a Discrete CMAC and train it on a 1-D function (ref: Albus 1975, Fig. 5) Explore effect of overlap area on generalization and time to convergence. Use only 35 weights for your CMAC and sample your function at 100 evenly spaced points. Use 70 for training and 30 for testing. Report the accuracy of your CMAC network using only the 30 test points.

Ans.1: -

The function which is taken to plot the graph is a cosine of 'x'. The 100 function inputs which are evenly space are spread between (0,10). The overlap area with a weight matrix of 35 is spread from 1 to 34. The function plot for the same is shown below: -

1) Main Function: -



• Inputs of (0,10) is distributed over 100 evenly spaced points, then the output is calculated using the function cos (x).

• Initialize function is used in order to initialize Weights and overlap area. Then random 70 values are assigned from x which are used to train the data and 30 values are used for testing purpose. At last the train and test functions are runed and the values are plotted in a graph.

2) Initialize the functions: -

Here we have to define the input vector and the binary Look up table. The input vector is distributed evenly between the mac and the min of the inputs data. Whereas the table is used to know the number and the type of the associate cell which are linked for the given inputs.

3) Train Function

Now define a variable index, which will categorize the data which is used the look up table and find out how many associative cells are linked with the inputs. Then the associated weight values are added to calculate the output. Also calculate the error and then update the weights. Now, calculate the simultaneous mean absolute error and Return the updated values of weights and the count for the number of iterations performed.

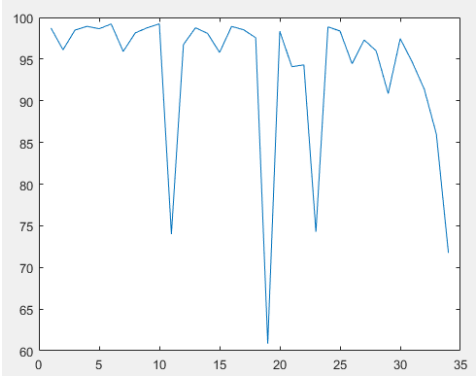
4) Test Function

This is very similar to the train function but, the only difference is that it uses the test data instead of the training data. Now find the accuracy and return the amount of iteration performed.

Now the function is made to run from an overlap area of 1 to an overlap area of 34. The following graphs are as follows: -

1) Accuracy in discrete function:

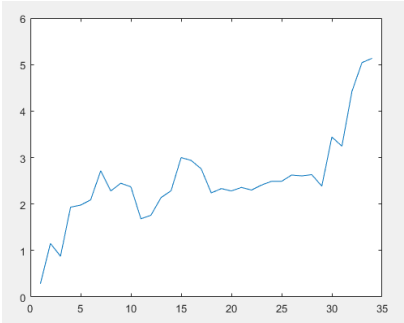
It's observed that, with the increase in the overlap area there is a decrease in the accuracy. The reason for this can be given by the generalization concept. As the number of overlaps increases, generalization becomes difficult and it would be hard to have a higher accuracy the reason being the change in a single weight is affected all across the other weights.



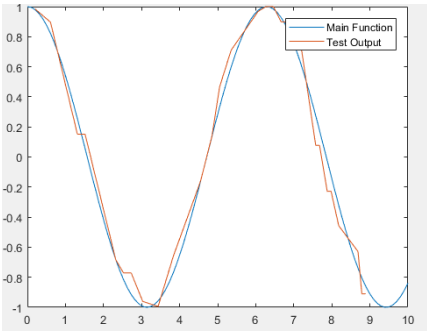
The worst accuracy is 60.94 for when the overlapping area is 19 out of 35 weights and the best accuracy is coming 98.9264 at 10 overlapping weights.

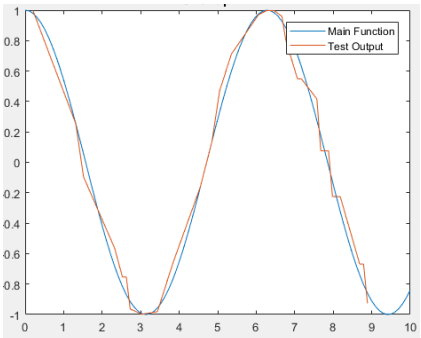
5) Time taken to converge for the discrete function:

It is very clear from the graph that, with the decrease in the accuracy of the function the time for convergence is increasing.



The graphs for the discrete function, the training data and test data is depicted below: -

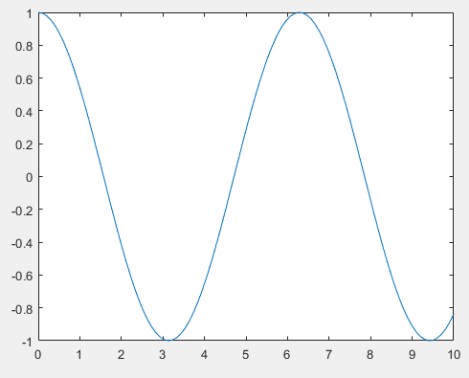




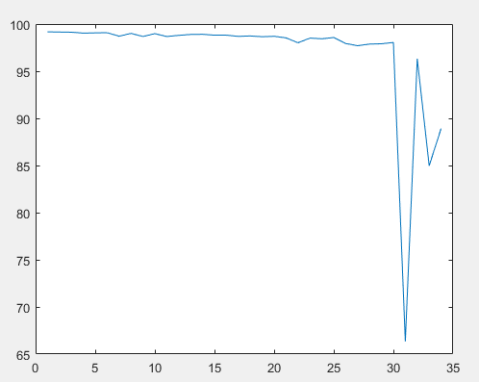
Q2) Program a Continuous CMAC by allowing partial cell to overlap and modifying the weight update rule accordingly. Use only 35 weights for your CMAC and sample your function at 100 evenly spaced points. Use 70 for training and 30 for testing. Report the accuracy of your CMAC network using only the 30 test points. Compare the output of the Discrete CMAC with that of the Continuous CMAC.

Ans 2: -

For solving the continuous CMAC, same code is used as for the question 1, the only difference is that, the continuous values of cos (x) are taken rather than discretized values. The original function for the same is plotted below: -



Graph: - Plot of input function cos(x)



Graph: - Plot of accuracy against overlapping space

1) Accuracy for continuous function:

The results here are similar to the previous case. Again, the accuracy is decreasing with increase in generalization or increase in overlapping area. The maximum accuracy is 99.1740 when the overlapping area is 1 and the minimum accuracy is 66.3411 at overlapping area of 31.

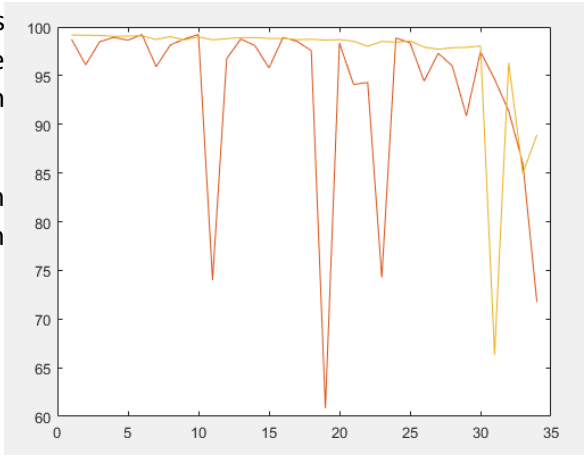
Accuracy= [99.1740 99.1535 99.1366 99.0373 99.0662 99.0837 98.7136 99.0117 98.6817 98.9848 98.6795 98.7994 98.9010 98.9112 98.8219 98.8227 98.6956 98.7386 98.6574 98.7009 98.5348 98.0162 98.5085 98.4395 98.5701 97.9353 97.7035 97.8771 97.9084 98.0485 66.3411 96.3204 84.9688 88.9222]

When we compare accuracies for both discrete and continuous functions, we can see that the accuracies for both are almost the same in the beginning, but towards the end or with increase in overlap area, the accuracies decease haphazardly.

To generalize, the accuracy of continuous function is better than that of discrete function, but the accuracy decreases in both towards the end.

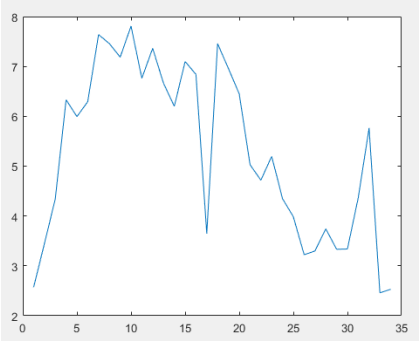
Alongside is a plot between accuracy vs overlapping area

Discrete function= red Continuous function= orange



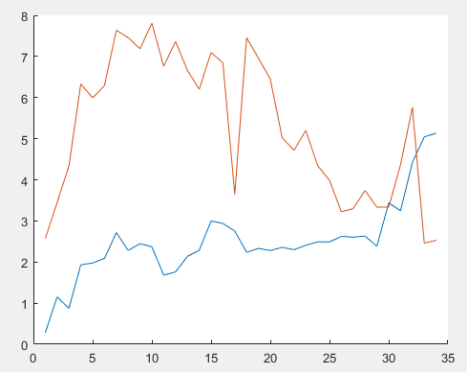
2) Time for Convergence for continuous function

The time taken for convergence differs between 2.5 to 8 seconds, where the maximum time taken is 7.745 seconds at overlap space of 7 and minimum is 2.67 seconds at overlap space of 1.



Tcontinous= [2.5732 3.4468 4.3424 6.3318 5.9932 6.2927 7.6388 7.4569 7.1859 7.8085 6.7617 7.3635 6.6637 6.1999 7.0990 6.8395 3.6472 7.4571 6.9580 6.4494 5.0279 4.7179 5.1973 4.3467 3.9857 3.2227 3.2991 3.7414 3.3322 3.3380 4.3681 5.7640 2.4583 2.5790]

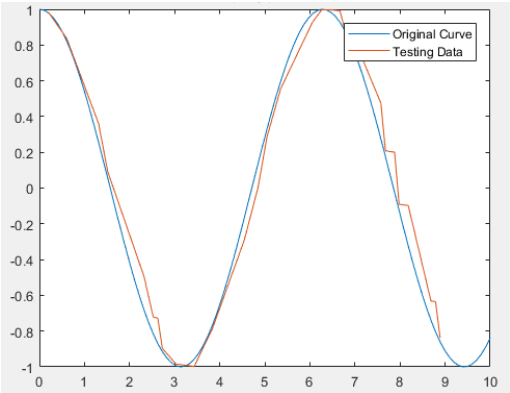
Comparing Tdiscrete and Tcontinous overweight space



On comparing, we see that the time taken to converge is generally slightly larger for continuous than discrete. This may be because of the greater number of input points in continuous function.

Tdiscrete: Blue Tcontinous= RED

• The plots for continuous function and test data are shown below when the overlap is of one. At this numcell value there is maximum accuracy and least time for convergence.



Q3) Discuss how you might use recurrent connections to train a CMAC to output a desired trajectory without using time as an input (e.g., state only). You may earn up to 5 extra homework points if you implement your idea and show that it works.

Ans 3.

The entire idea of the recurrent connection is based on the fact that, it gets two inputs. These two inputs are the current and the present past data. The idea of having memory in humans is resembled by the information which is in the sequence of the past output. This information which is in the sequential order causes spam as it cascades forward in order to affect the processing of the new examples, this occurs because of the information being preserved in the hidden network state. Thus, in order to solve the problems of missing and exploding gradients, Long Short-Term Memory Units were proposed.

In the method of recurrent connections all the information is saved in the gated cells, which is outside of the normal flow of the recurrent network. The, pass or block information is received by these analog gates based own their own set of weights. Moreover, the recurrent networks learning process is used to adjust these gates.Now, in order to implement the idea of recurrent connection into the code, We should make use of the error function and an additional function which depends on the output, to correct the output.