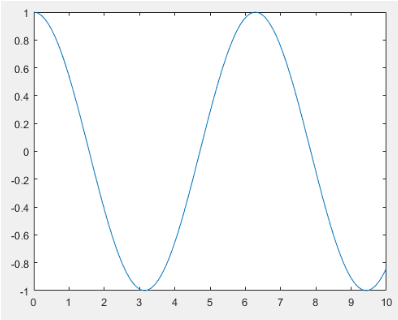
Anirudh Topiwala UID: 115192386, Homework #2, ENPM 808F- Robot Learning

1) 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.

- The 1-D function I have taken is cosine of x. The 100 evenly space function inputs are varied between (0,10). The overlap area is varied from 1 to 34 with a weight matrix of 35. The function plot is shown below:



Algorithm:

1) Main Function:
- Distribute the inputs of (0,10) over 100 evenly spaced points
- Calculate the output using the function cos (x)
- Use **initialise function** to initialize Weights and overlap area.
- Assign 70 random values from X to training data and 30 to test data.
- Run the **train** and **test** function and plot the values.

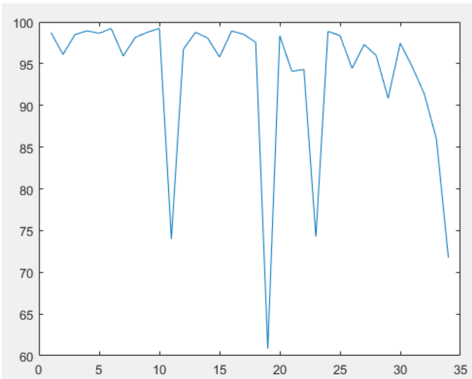
2) Initialise function
- Define input vector evenly distributed between the max and the min of inputs. This will be used to index the inputs so as associate them with the weight matrix.
- Define binary Look up table to know how many and which associate cell is linked for the inputs given.

3) Train Function
- Defining a variable indexing, it will categorize the training data so that it can then use the look up table and find out how many associative cells are linked with the inputs.
- Calculate the output by adding the values of all the associated weights.
- Calculate error and update the weights
- Calculate smape (simultaneous mean absolute error)
- Return the updated values of weights and also return the number of iterations performed.

4) Test Function
- Similar to train function, just using test data in place of train data.
- Calculate accuracy and return the number of iterations 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 for discrete function:

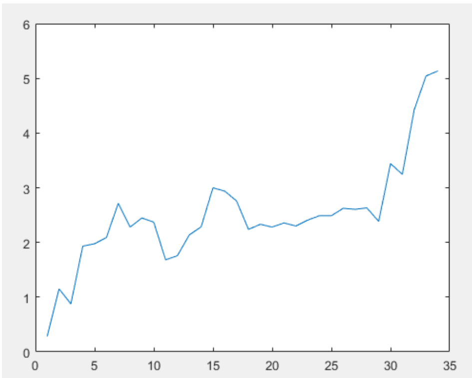


We can see that as the overlap area is increased the accuracy is somewhat decreasing. This can be explained by the concept of generalization. The more the number of overlaps or more the generalization the difficult it would be to have a higher accuracy as change in a single weight is affected across all the other weights.

The worst accuracy is 60.8522 for when the overlapping area is 19 out of 35 weights and the best accuracy is coming 99.2264 at 10 overlapping weights.

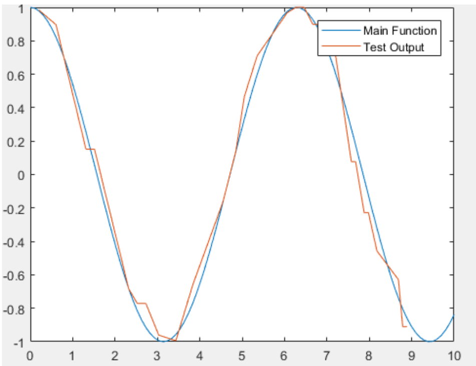
Accuracy = [98.7367 96.1160 98.4927 98.9312 98.6461 99.2160 95.9199 98.1481 98.7692 99.2264 74 96.7481 98.7573 98.0913 95.7987 98.9259 98.5203 97.5658 60.8522 98.3462 94.0890 94.3079 74.2656 98.8645 98.3771 94.4426 97.2998 95.9987 90.8668 97.4602 94.6252 91.3684 86 71.7503]

2) Time to convergence for the discrete function:

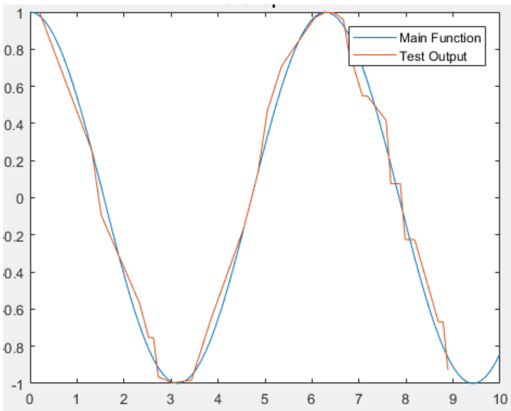


- We can infer from the graph that, as the accuracy of the function is decreasing the time for convergence is increasing.
- The maximum time taken is 5.1344 seconds which is at the overlap area of 34.
- The minimum time taken is 0.2834 s when the overlap is 1.
- The minimum time can be justified by the fact that, as the overlap is 1, it is similar to one to one mapping and therefore it takes minimum time.
- Tdiscrete: [0.2834 1.1501 0.8768 1.9326 1.9764 2.0886 2.7154 2.2816 2.4472 2.3701 1.6813 1.7589 2.1373 2.2853 2.9989 2.9381 2.7557 2.2384 2.3318 2.2805 2.3571 2.2998 2.4079 2.4887 2.4886 2.6237 2.6058 2.6317 2.3840 3.4401 3.2451 4.4243 5.0430 5.1344]

The plots for the discrete function, the training data and test data is shown:
1)for the overlapping value of 19 or the case of maximum accuracy.
2)for the overlapping value of 2 or in the case of the quickest convergence



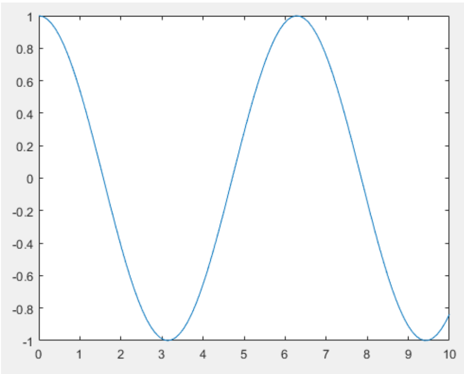
(1)



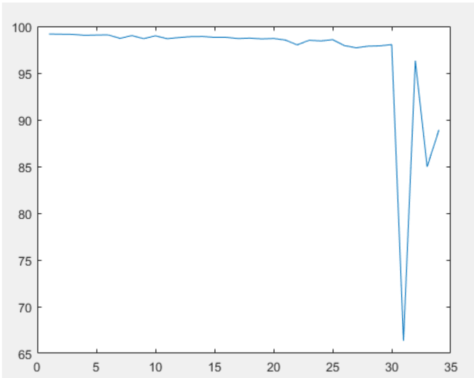
(2)

2) **Program a Continuous CMAC by allowing partial cell 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.**

- The same code is used, only the continuous values of cos (x) are taken rather than discretised values. Again, the original function is plotted below:



Plot of input function; cos(x)



Plot of accuracy against overlapping space

1) Accuracy for continuous function:

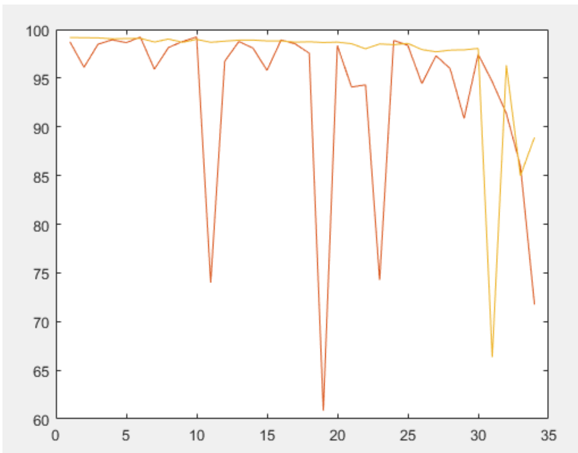
The results here are similar to the previous case. Again, the accuracy is decreasing with increase in generalization or increase in overlapping area. The maximum accuracy is 99.1740 when the overlapping area is 1 and the minimum accuracy is 66.3411 at overlapping area of 31.

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.

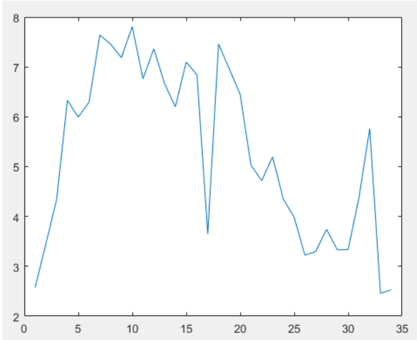
Alongside is a plot between accuracy vs overlapping area



Discrete function= red Continuous function= orange

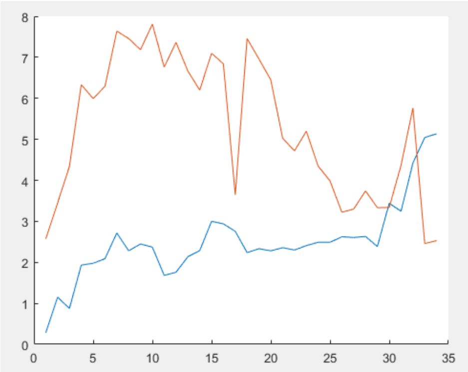
2) Time for Convergence for continuous function

- The time taken for convergence is varying between 3 to 8 seconds.
- The maximum time taken is 7.6388 seconds at overlap space of 7 and minimum is 2.5732 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]

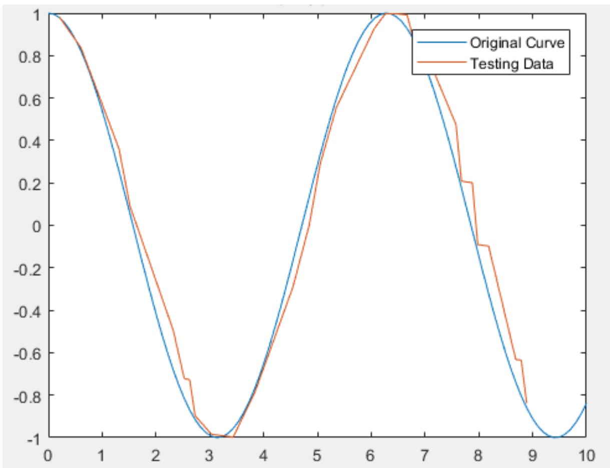
Comparing Tdiscrete and Tcontinous over weight space



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

Tdiscrete: Blue Tcontinous= RED

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



3) 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.

- Recurrent connections depend upon the idea that it receives two inputs, the current and the present past. There is information in the sequence of the past output which is analogous of having memory in humans. This sequential information is preserved in the networks hidden state and manages to span many times at it cascades forward to affect the processing of each new example. To avoid the problems of vanishing and Exploding gradients, Long Short Term Memory Units were proposed.
- In this method, the information is stored outside the normal flow of the recurrent network in gated cells. These analog gates receive, pass or block information based own their own set of weights. These are also adjusted via the recurrent networks learning process.
- To implement this concept in the original code, we would need to correct the output by error function and an additional function which depends on the output. This will make the function depend on itself or its previous outputs.

Reference: A Beginner's Guide to Recurrent Networks and LSTMs by DL4J. (https://deeplearning4j.org/lstm.html)