ROBOT LEARNING (ENMP-690)
HOMEWORK ASSIGNMENT – 2

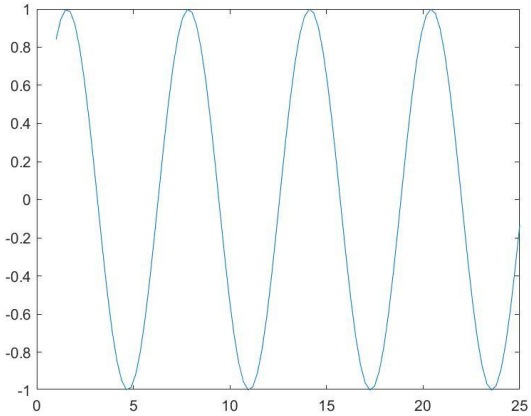
By
- Santosh Ajay Teja Kesani
- UID – 117035605

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 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.

2) Program a Continuous CMAC by allowing partial cell overlap, and modifying the weight update rule accordingly. Use only 35 weights 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 (1&2):

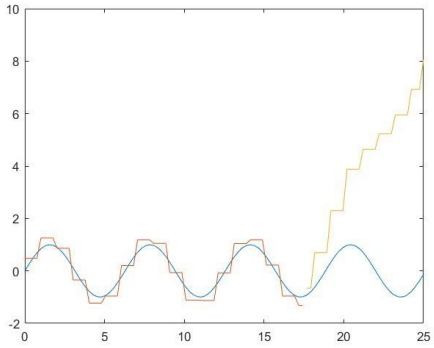
1-D function chosen is sine of x. Inputs are varied between (1,25) [Evenly spaced].
Function Plot:



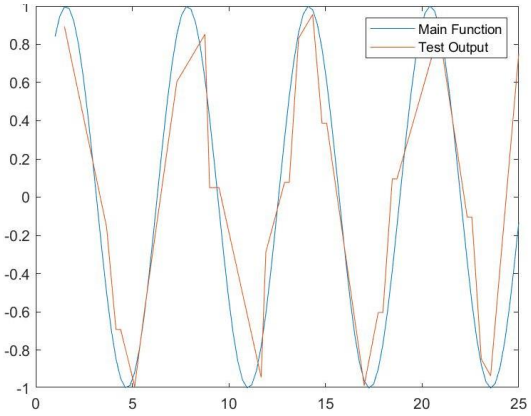
CODE : The Prog_main.m is used to - sample the inputs over 100 evenly spaced point, intialize the weights and overlaps, assigning the random values to variables and also to run the training and testing functions. The Input_module.m defines the input vectors and how the weights are to be assigned. The train_module.m is to train the network like calculating error, updating the weights.Finally, the test_module.m is to test the accuracy and return the iterations performed.

Discrete CMAC assigns equal weightage for all the weight vectors by allowing a partial cell to overlap and modifying the weight update rule accordingly whereas continuous CMAC assigns weightage based on the sliding window concept where the weight vectors at the start and the end of the window will have unequal weights.

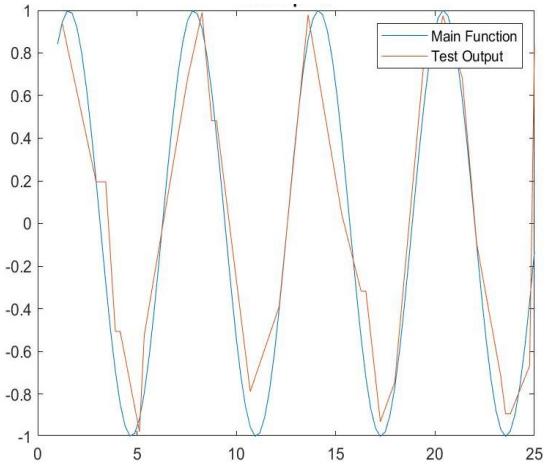
As there is an increase in the overlap area there is a simultaneous decrease in the accuracy which is portrayed through the graph which can be explained through the concept of generalization.
The graph also represents the sudden growth in the inaccuracy of the discrete CMAC as compared to the continuous CMAC as the overlap areas begins to increase drastically.
Similarly, there is corresponding variance in the converge time of the CMAC because the continuous CMAC has greater inputs.



Accuracy Variation



Discrete



Continuous

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.

Ans:

- **Recurrent Connections** are based on the idea that they are fed information not just from the previous layer but also from themselves from the previous pass. This means that the order in which you feed the input and train the network matters.
- Major problem with this, is the vanishing (or exploding) gradient problem where, information rapidly gets lost over time. Intuitively this wouldn't be much of a problem because these are just weights and not neuron states, but the weights through time is where the information from the past is stored. The weights are also adjusted by the recurrent network model.
- To use this model in our code, the output should be rectified using an error function and we also require another function which will depend on the output generated. Following this would make the function depend on itself and also on the output.
- This method can be used in many fields as most forms of data that don't have a timeline can be represented as a sequence. In general, recurrent networks are a good choice for advancing or completing information, such as autocompletion.