Homework Assignment – 2

Discrete and Continuous CMAC

ENPM690 – Robot Learning

Submitted to -

Prof. Donald Sofge

Submitted by –

Naman Gupta (116949330)

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: For 1-D function, I have used $y = \sqrt{x}$ to train CMAC. I have used the concept of human feedback system. So, our CMAC must learn to trace a curve using 70 training points. As the error increases, it will be used as a feedback to align itself to the curve.

For training the CMAC, first a variable index is defined which will categorize the training data so that it can use the look up table to find how many associative cells are linked within the inputs. Next, the output is calculated by adding the values of all the associated weights. After calculating the error, the weights are updated. Mean error is calculated and returns the updated values of weights.

For testing, the same method is employed as used for training the data. test_cmac() is used for testing the CMAC, which returns accuracy. The process is repeated for certain number of iterations, I used 30 iterations.

The problems I faced in training CMAC is the lack of documentation availability regarding CMAC, which gave me hard time to learn what is CMAC and how can it be implemented. The research paper on CMAC by Albus was too hard to understand for a beginner in machine learning like me. So, I tried and got the accuracy which is shown in figure 1.

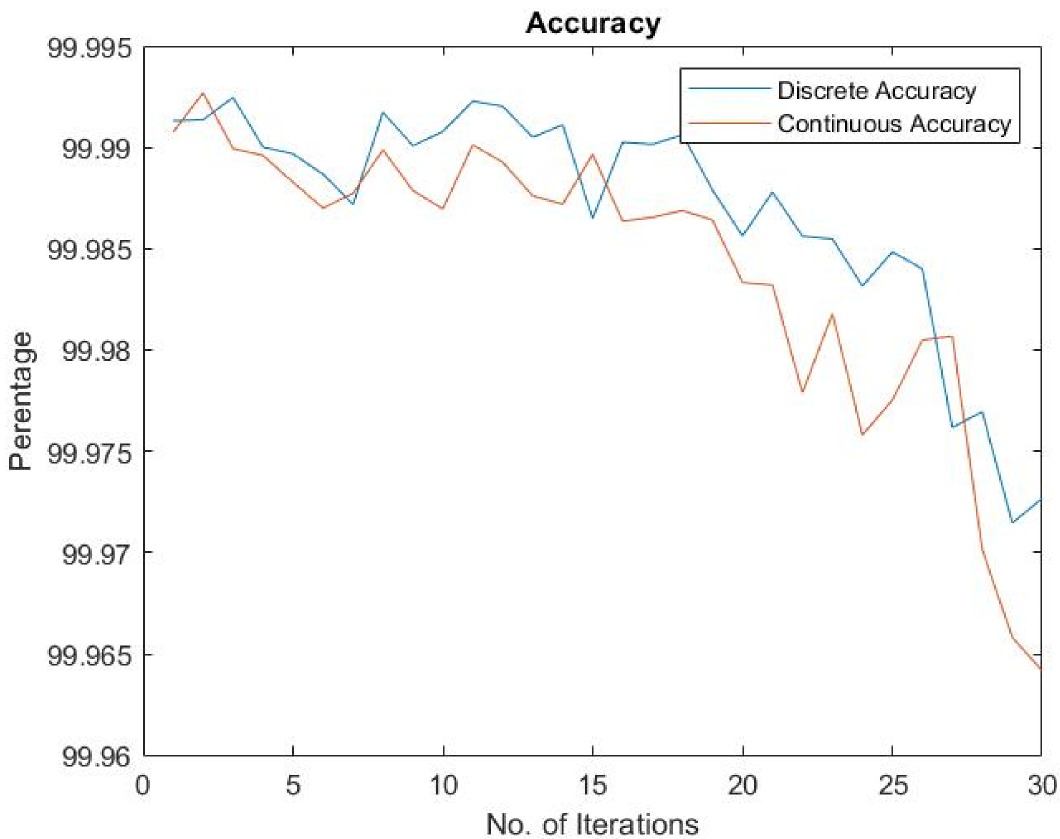


Fig-1: Accuracy of Discrete CMAC and Continuous CMAC

The plot for time of convergence is shown in figure 2.

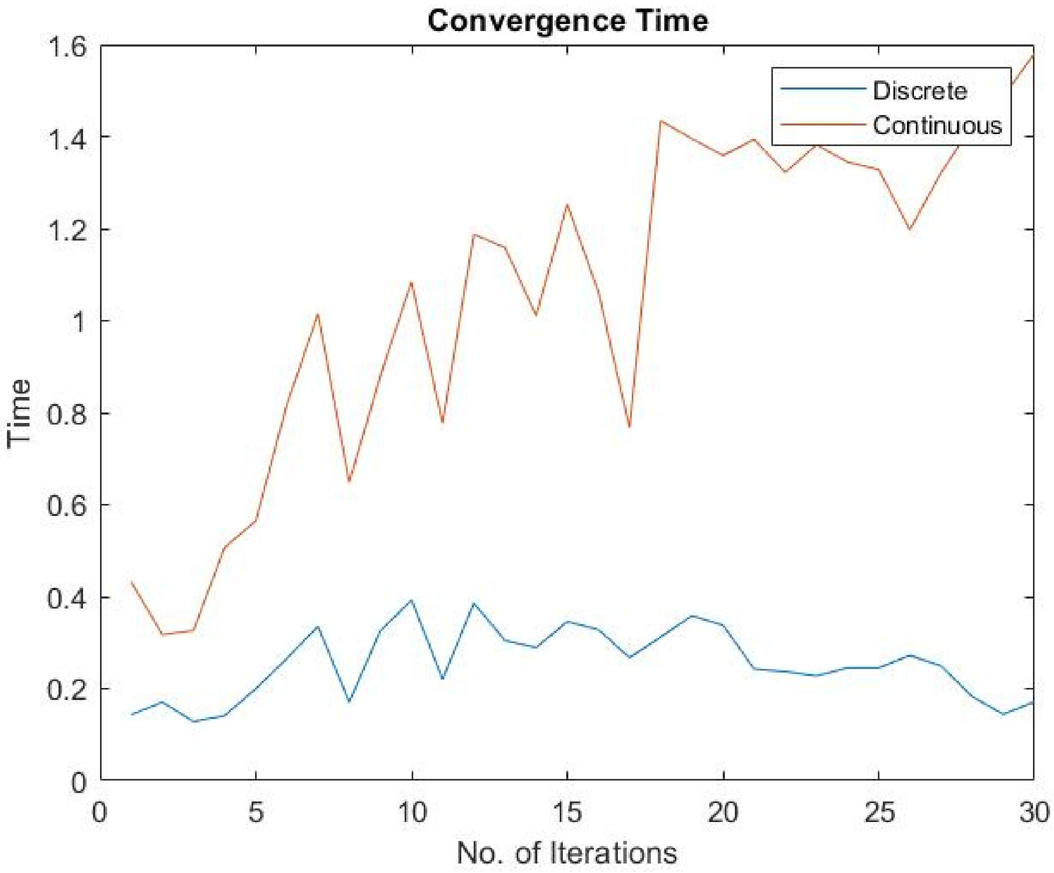


Fig – 2: Time of Convergence

We can see that, as the accuracy decreases for discrete output, the convergence time increases. The overlap area is shown in figure 3.

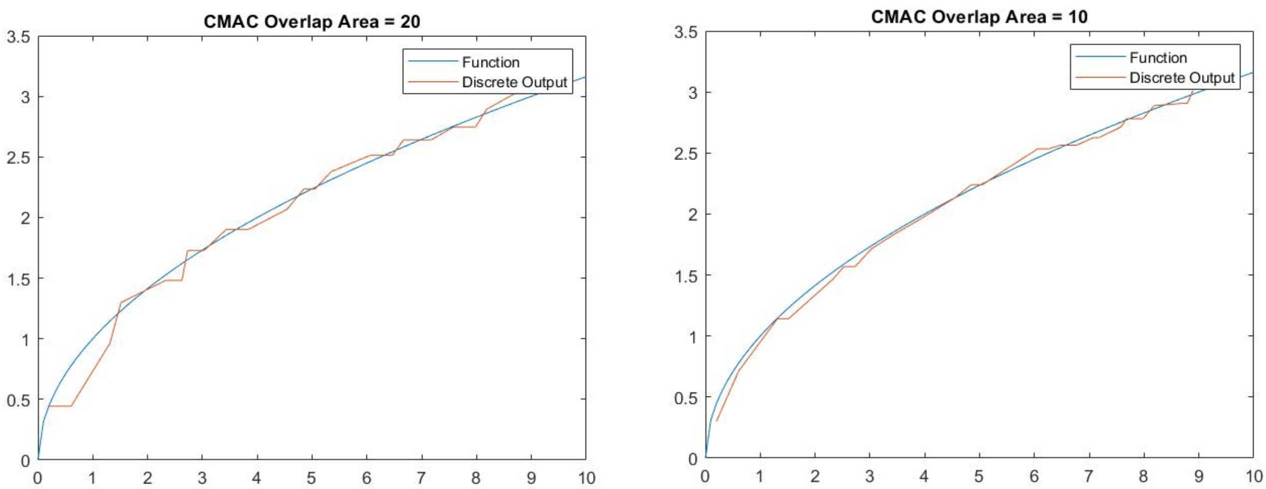
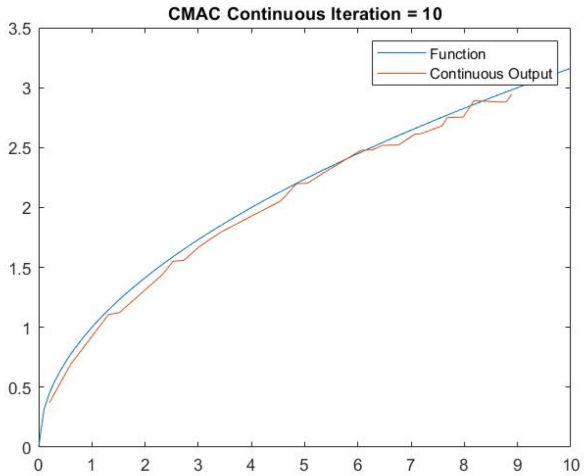


Fig - 3: Effect of Overlap Area using Discrete CMAC

The outputs show that when a greater number of overlaps and generalization are there, it is difficult to have a higher accuracy because change in a single weight affects all the other weights. This is global generalization.

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: As we can see in figure 2 and 3, the accuracy of continuous CMAC decreases as the time of convergence increases over the iterations. In figure 4, we see that as the accuracy increases the area of overlap also decreases so basically same as discrete CMAC.



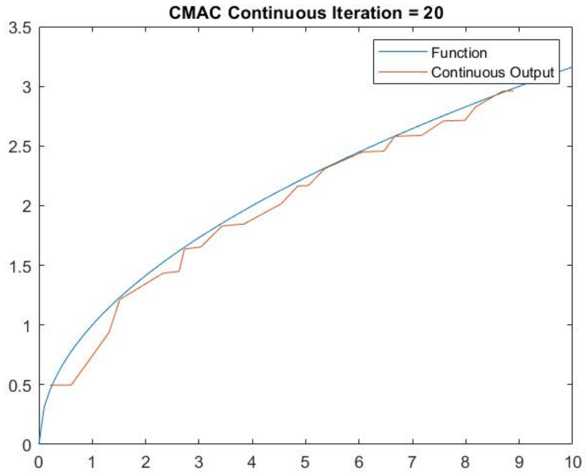


Fig – 4: Effect of Overlap Area using Discrete CMAC

Figure 5 & 6 shows the output of both discrete as well as continuous CMAC respectively.

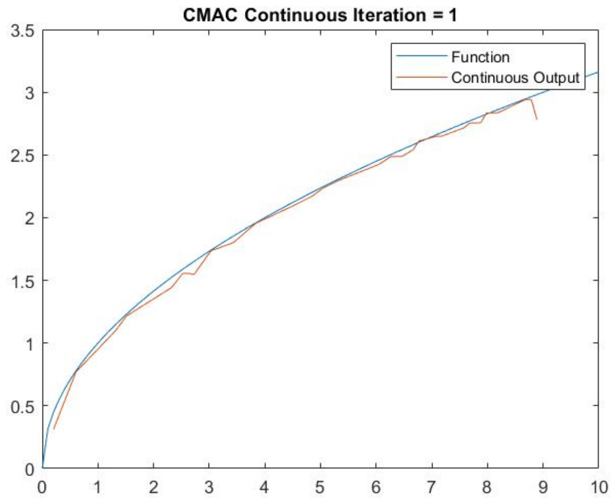


Fig – 5: Continuous CMAC

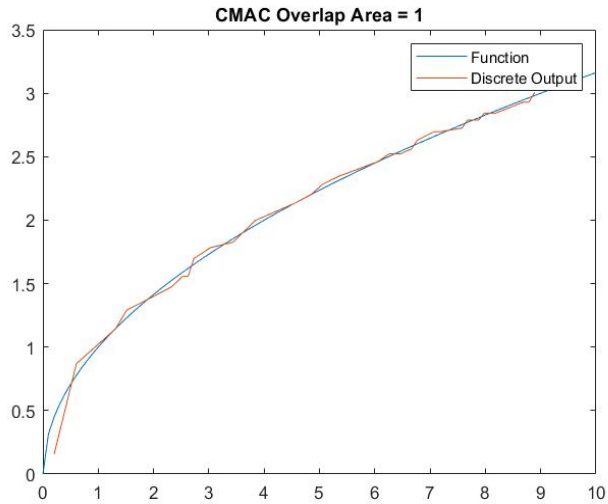


Fig – 6: Discrete CMAC

If we notice the accuracy of continuous and discrete CMAC from the figure 1, we see that discrete CMAC has somewhat similar accuracy to continuous CMAC, but eventually both are decreasing as the number of iterations increases. In figure 2, the convergence time of continuous CMAC is more than discrete CMAC.

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: For this task, a recurrent neural network can be used. A recurrent neural network (RNN) is derived from feedforward neural network, which can use its internal state for the processing of variable length sequences of input. It is like an artificial neural network where connections that are between the nodes, they form a directed graph which is along a temporal sequence, this allows it to show its temporal dynamic behavior.

We can change the code to adjust the output using error and output's feedback, like a closed loop system. So, using the error the output will adjust itself and can calculate the remaining error accordingly. Just like a PID controller which is not dependent on time but on its states.

GitHub link: https://github.com/namangupta98/Discrete-and-Continuous-CMAC