ENPM690 HW2

Aman Virmani

Introduction

This assignment describes the operation of the CMAC neural network. Both discrete and continuous CMAC has been trained and tested on a one-dimensional function. Performance comparisons have also been drawn in both versions.

CMAC is an acronym for Cerebellar Model Articulation Controller. The CMAC was first described by Albus in 1975 as a simple model of the cortex of the cerebellum. Despite its biological relevance, the main reason for using the CMAC is that it operates very fast, which makes it suitable for real-time adaptive control.

Algorithm

Please note that the overlap area factor is mentioned as the Generalization Factor in my code and that I have used cosine function as the 1-D function.

- The steps mentioned below are repeated for 35 different values of the Generalization Factor.
- Distribute the input and true output of a function over 100 evenly-spaced points.
- Split data randomly taking 70 points for training and 30 points for testing.
- The weight matrix is initialized with zero values.
- The CMAC is then trained to updated the values of the weight matrix to learn the input function. This involves calculating output using training values. This output is then subtracted from actual output to get the absolute error. This absolute error is scaled by the learning rate to update the weights.
- The trained weights are then used to predict the output for test data and testing error is then calculated to analyze the performance.

Performance

Here, we will discuss the performance of both discrete and continuous CMAC, on the following metrics:

1. Overlap Area

From Figure 5., we observe that the training accuracy first decreases significantly (around 30%) and gradually starts increasing as the overlap increases. This dip in the training accuracy is observed because as the generalization factor is increased, the mapping from input to weight space is changed from one-to-one to one-to-many. However, when the generalization (or overlap) is increased further, the training accuracy increases providing better learning on the training data.

We also observe that the CMAC network shows better generalization to new test data as the overlap is increased.

2. **Time of Convergence**

From Figure 5, it can be seen that accuracy increases with an increase in time of convergence. A more accurate sense of convergence time can be gained from subplot 3 of Figure 5. It can be seen that the convergence time also increases with an increase in the generalization factor. As explained in the previous section, the generalization factor of n denotes 1-to-n mapping. Therefore, more complex mapping indicates more time will be taken for global convergence.

Learning Trajectory using Recurrent Connections (without Time)

Recurrent connections depend upon the idea that it receives two inputs, the present, and the past. There is information in the sequence of the past output which is analogous of having a memory in humans. This sequential information is preserved in the network's 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 on 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.

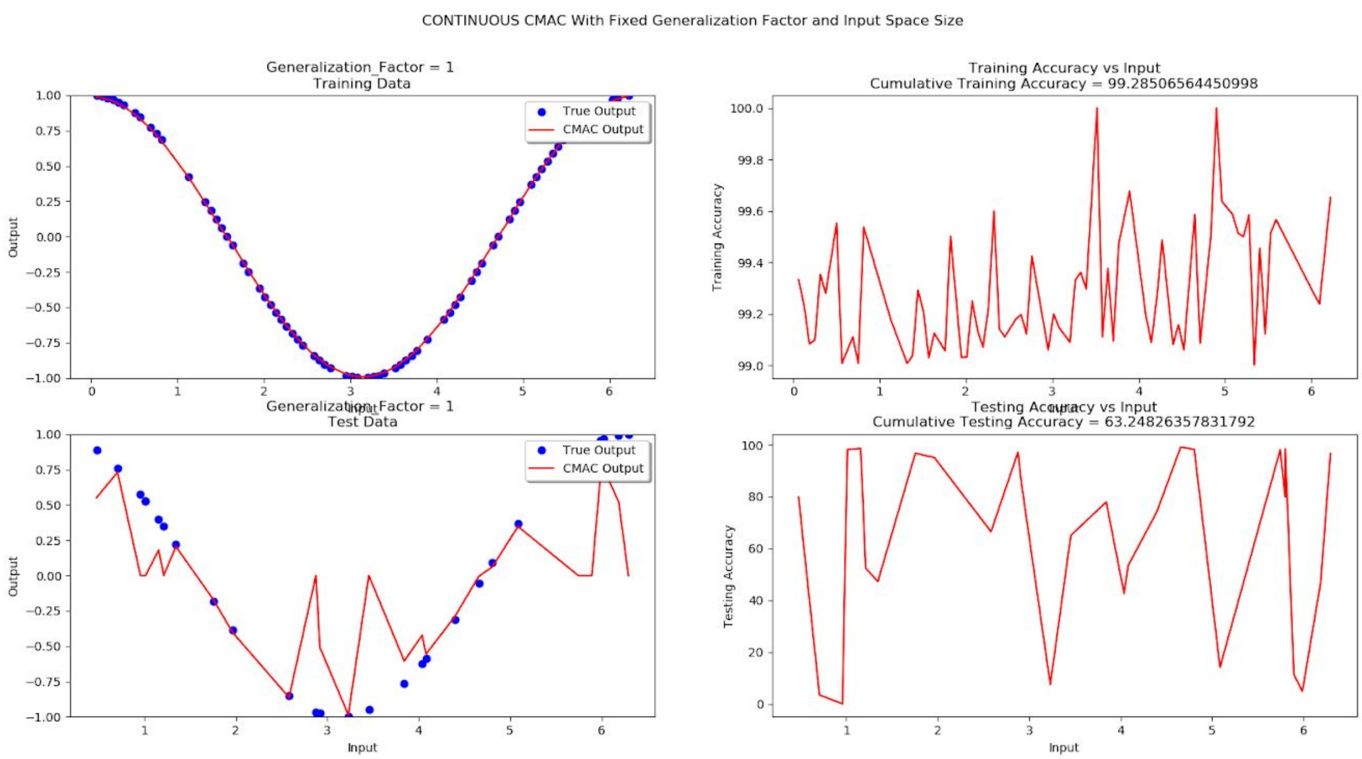

Figure 1: Continuous CMAC with a fixed generalization factor 1

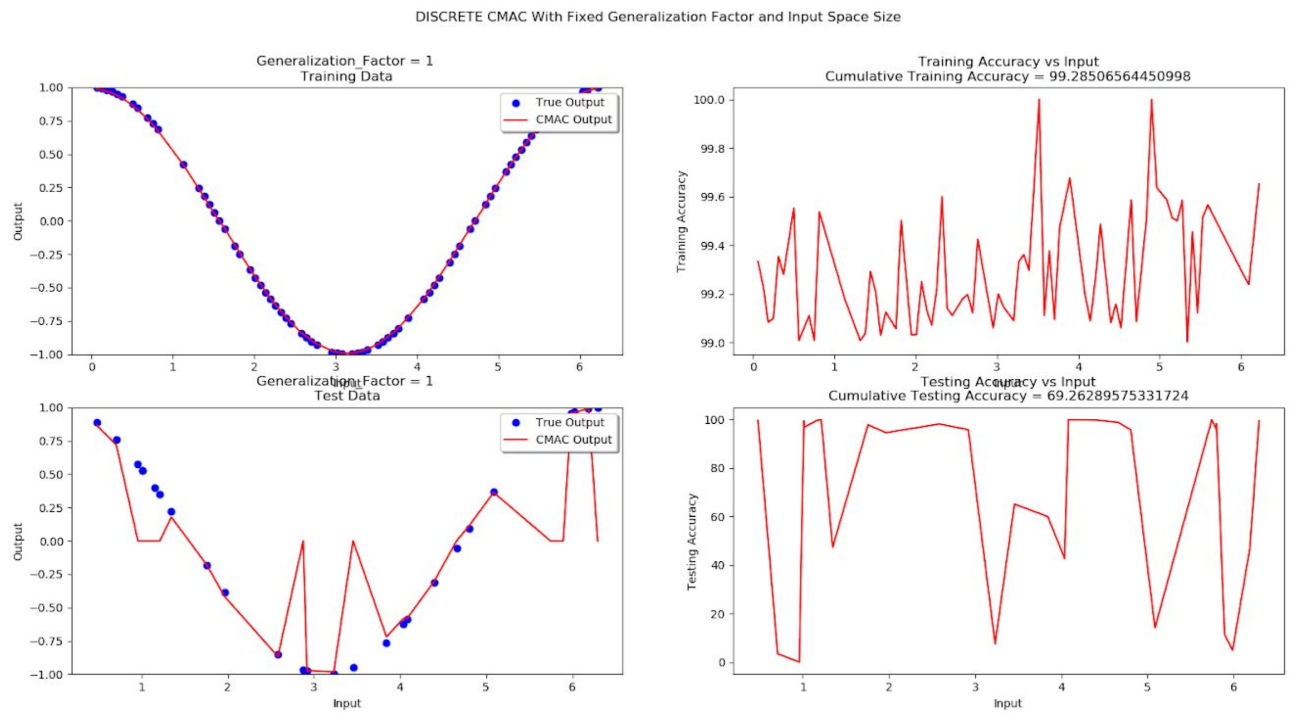

Figure 2: Discrete CMAC with a fixed generalization factor 1

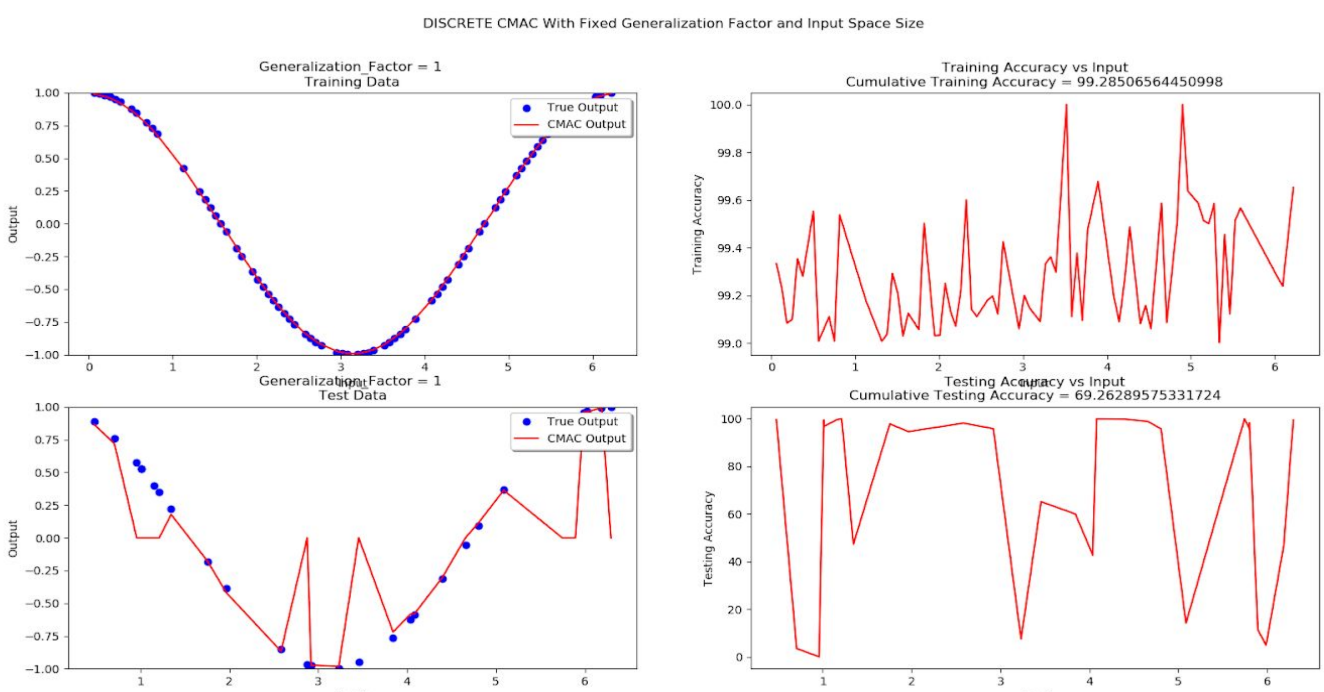

Figure 3: Discrete CMAC with an increase in Overlap Area (Best Case)

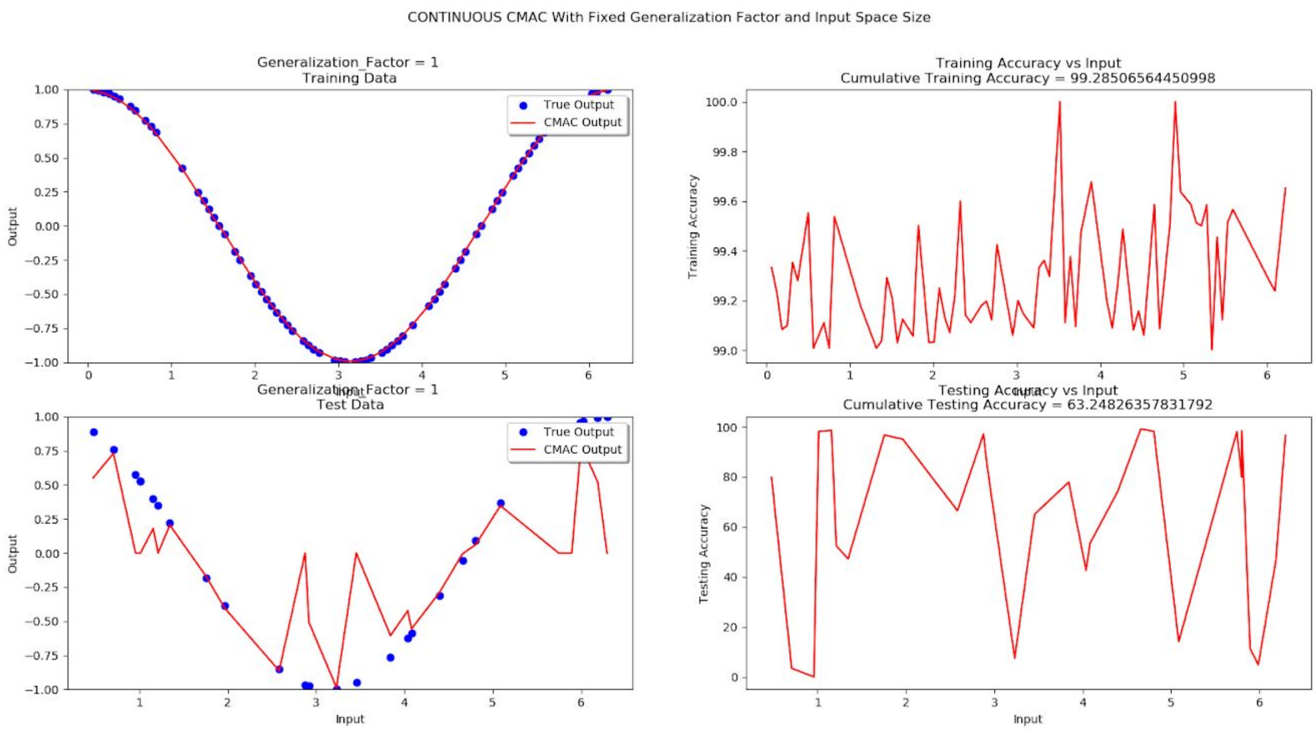

Figure 4: Continuous CMAC with an increase in Overlap Area (Best Case)

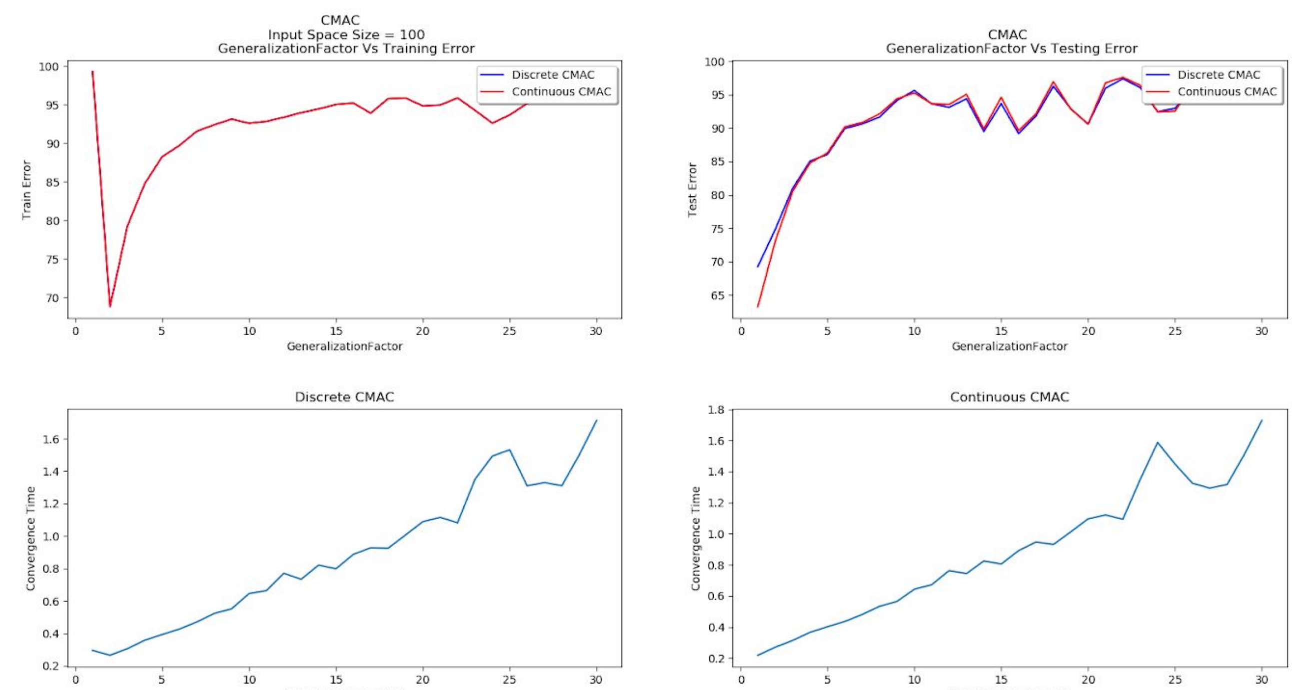

Figure 5: Performance Comparison of Discrete and Continuous CMAC