ENPM690 - Robot Learning
Homework 2
Shon Cortes

1. Formulate a robot learning task using machine learning terminology. Describe what are the inputs and outputs, and how and where the supervision takes part in.

An example of supervised learning would be training a model to take in a goal position for an end effector and have it output the joint states that would achieve this position. The model would be trained on labeled data to achieve a supervised learning model. The data would need to be preprocessed before moving on to assembling the structure for a machine learning model to ensure proper organization and labeling. In a 1-D model the input would be an x position and the output would be the angle the motor needs to be set to.

2. 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.

As the generalization factor increases, time to converge decreases. This also leads to a decrease in model accuracy. With a generalization factor of 5 and a 1-D function of y=x the total epochs for training was 3474 with an accuracy of 96.69% as seen in Figure 1. When the generalization factor was increased to 15 the total epochs decreased to 2769 with an accuracy of 91.72% as shown in Figure 3. This trend was replicated when looking at the function y=sin(x). The model accuracy decreased from 99.83% to 99.55% when increasing the generalization factor from 5 to 15. These trends can be seen in Figures 5 and 7. Ideally the generalization factor should be 1/1000 of the size of the training set however our data set was too small to properly implement this.

3. 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. (You may need to provide a graph to compare)

The Continuous CMAC had worse performance than the Discrete implementation. When using a generalization factor of 5 the Continuous CMAC achieved an accuracy of 82.98% for the 1-D function y=x as seen in Figure 2. It maxed out the number of epochs at 10010. For comparison, the Discrete model achieved 96.69% accuracy in 3474 epochs. Again this trend was replicated when increasing the generalization factor to 15 as seen in figure 4.

4. Discuss how the results will change if the number of weights used in the model increases and what are the main disadvantages of CMAC.

The number of weights used affects the trained models accuracy. The generalization factor correlates to the number of active weights in the association space for a given input. This means that with a larger number of weights, a given input has greater potential for overlap with other possible inputs. A smaller number of weights allows for less overlap and therefore more unique mappings for inputs. The number of weights also will affect the speed and size of the model. CMAC requires allocating memory for all weights that the inputs will map to. Larger weights used means more memory required for storing the model.

5. 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.

Extra Credit

Appendix:

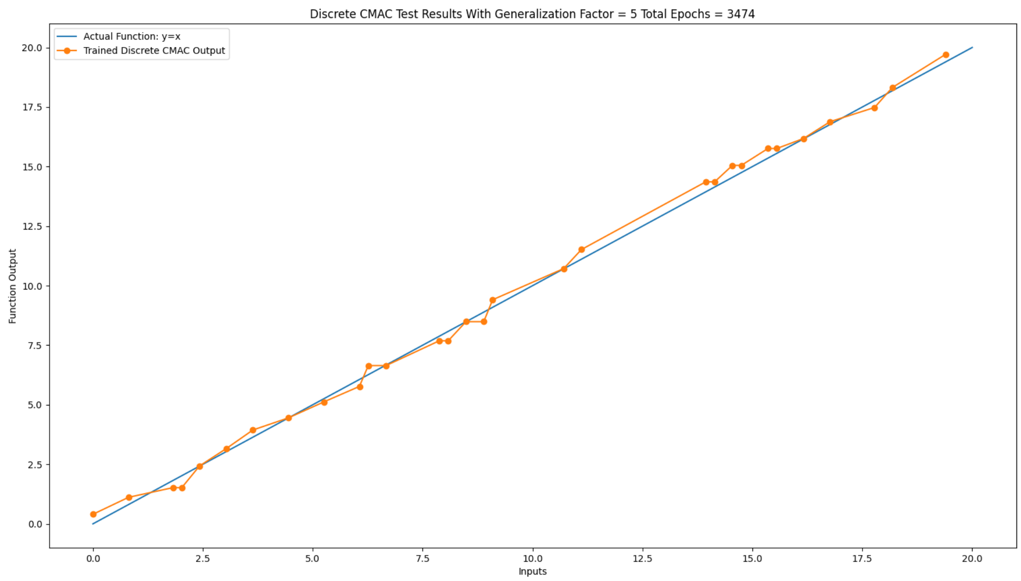

Figure 1: Discrete CMAC using a generalization factor of 5 achieved an accuracy of 96.69% after 3474 epochs. 1-D function of y=x.

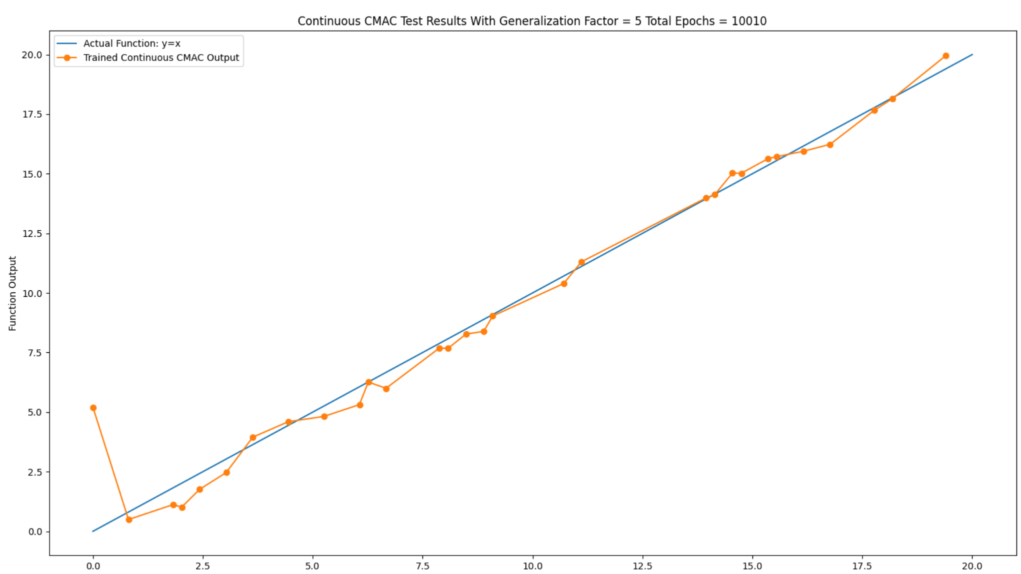

Figure 2: Continuous CMAC using a generalization factor of 5 achieved an accuracy of 82.98% after 10010 epochs. 1-D function of y=x.

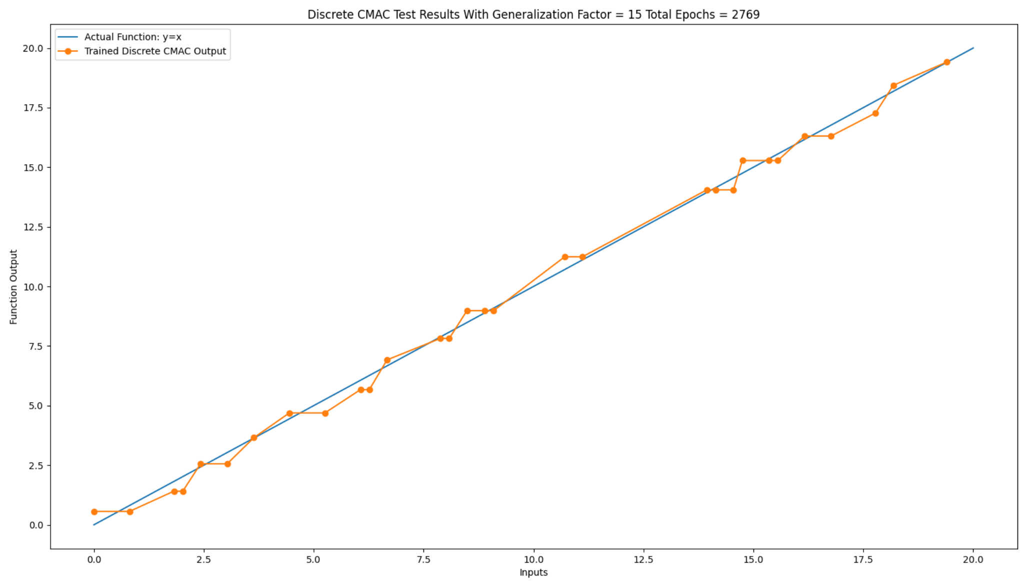

Figure 3: Discrete CMAC using a generalization factor of 15 achieved an accuracy of 91.72% after 2769 epochs. 1-D function of y=x.

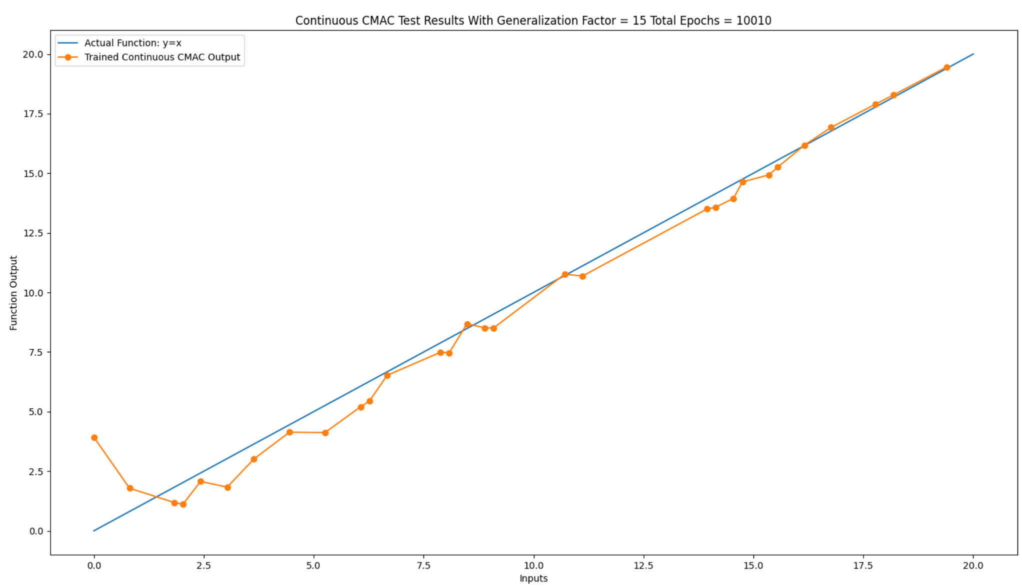

Figure 4: Continuous CMAC using a generalization factor of 15 achieved an accuracy of 73.71% after 10010 epochs. 1-D function of y=x.

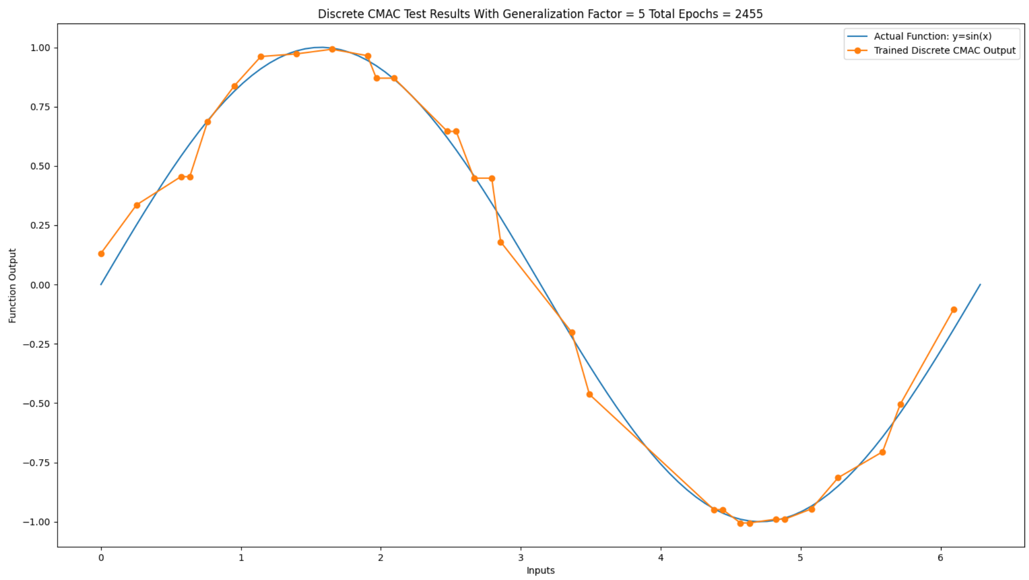

Figure 5: Discrete CMAC using a generalization factor of 5 achieved an accuracy of 99.83% after 2455 epochs. 1-D function of y=sin(x).

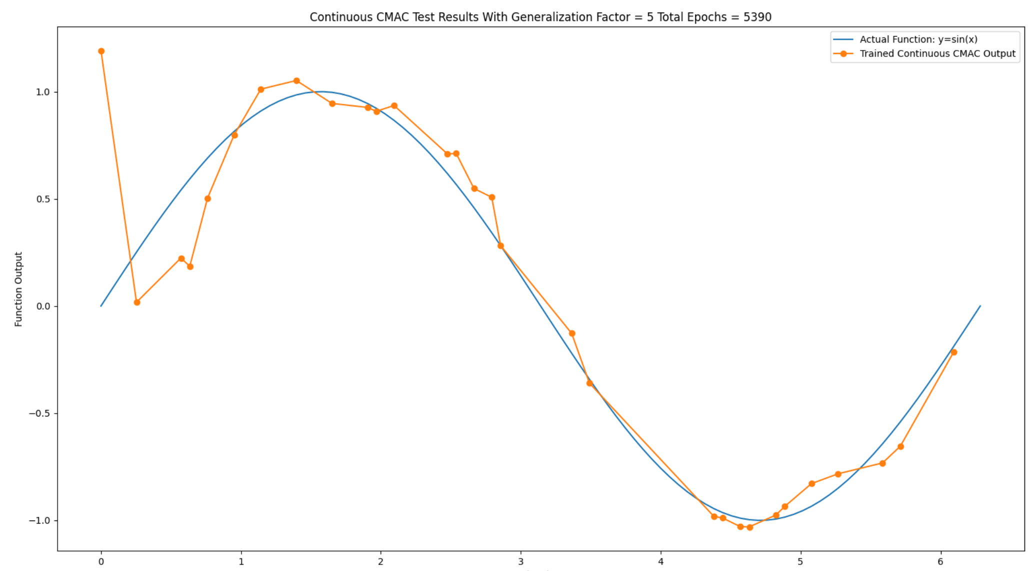

Figure 6: Continuous CMAC using a generalization factor of 5 achieved an accuracy of 98.39% after 5390 epochs. 1-D function of y=sin(x).

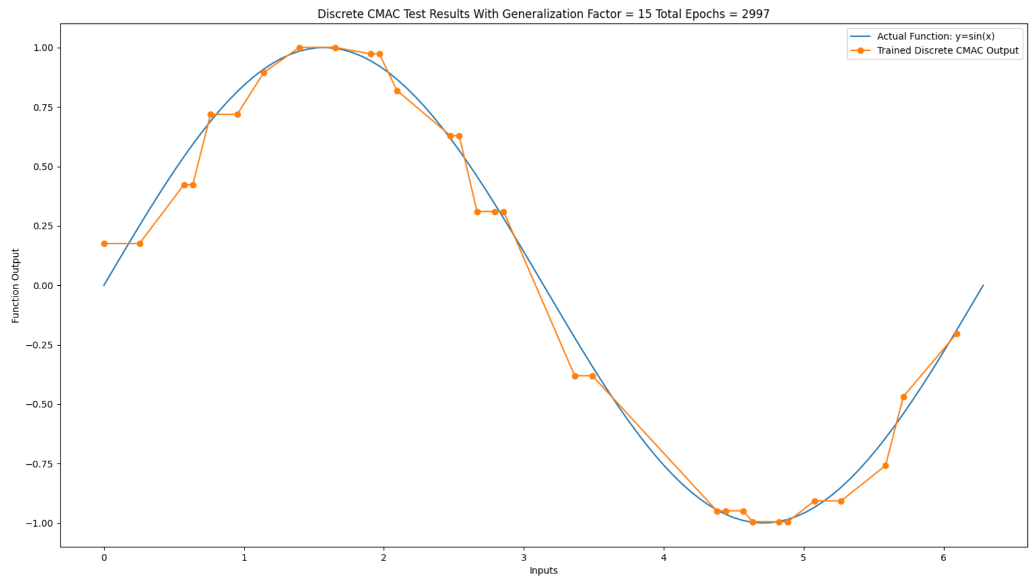

Figure 7: Discrete CMAC using a generalization factor of 15 achieved an accuracy of 99.55% after 2997 epochs. 1-D function of y=sin(x).

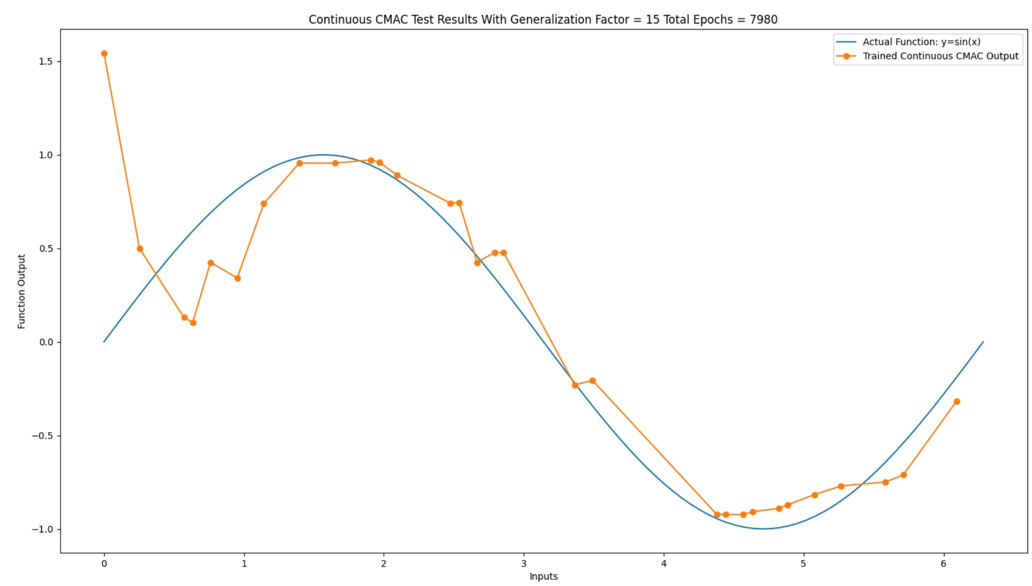

Figure 8: Continuous CMAC using a generalization factor of 15 achieved an accuracy of 97.66% after 7980 epochs. 1-D function of y=sin(x).