Assignment - 2
ENPM 690: Robot Learning

- Rishabh Mukund (UID: 117556124)

Answers:

1) Let’s say we want to make a wake word engine (A network that listens and is triggered when a specific word is uttered, example ‘ok google’ oe ‘alexa’)

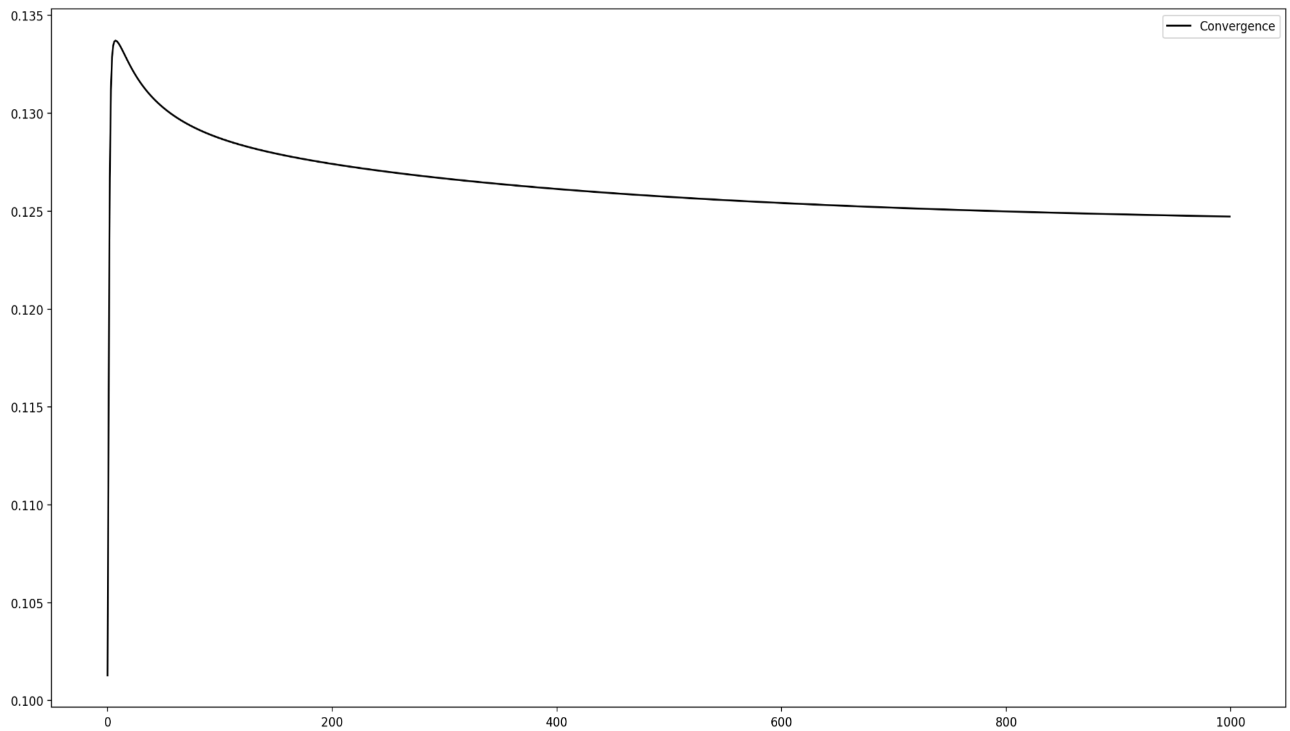
   To make a wake word engine, our data will be,
   input: mfcc coefficients of sampled speech for n second, and
   Output: 1 if wake word is present, 0 if no wake word is present.

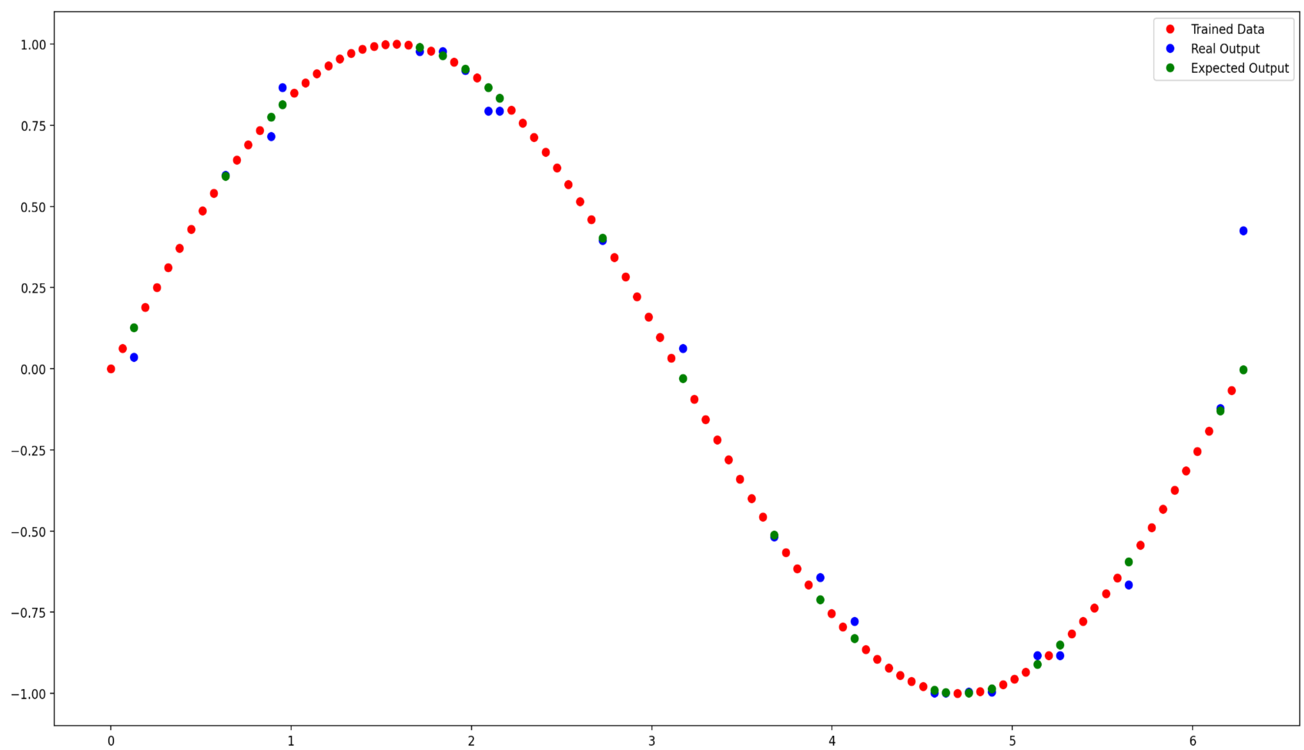
   We will organize the given data into test and train sets, we will train the network for multiple iterations for each pair of data in the training set. This is where the supervision takes place. We will pass the mfcc coefficients to the network and we will compare the results of the network with the desired output. If the output doesn't match the result of the network, we will update the weights in backpropagation.

   After updating the weights based on the training data set and supervision, we will check the efficiency of the network on the test data set. We will run the network on the test data set and check if we got the desired output.

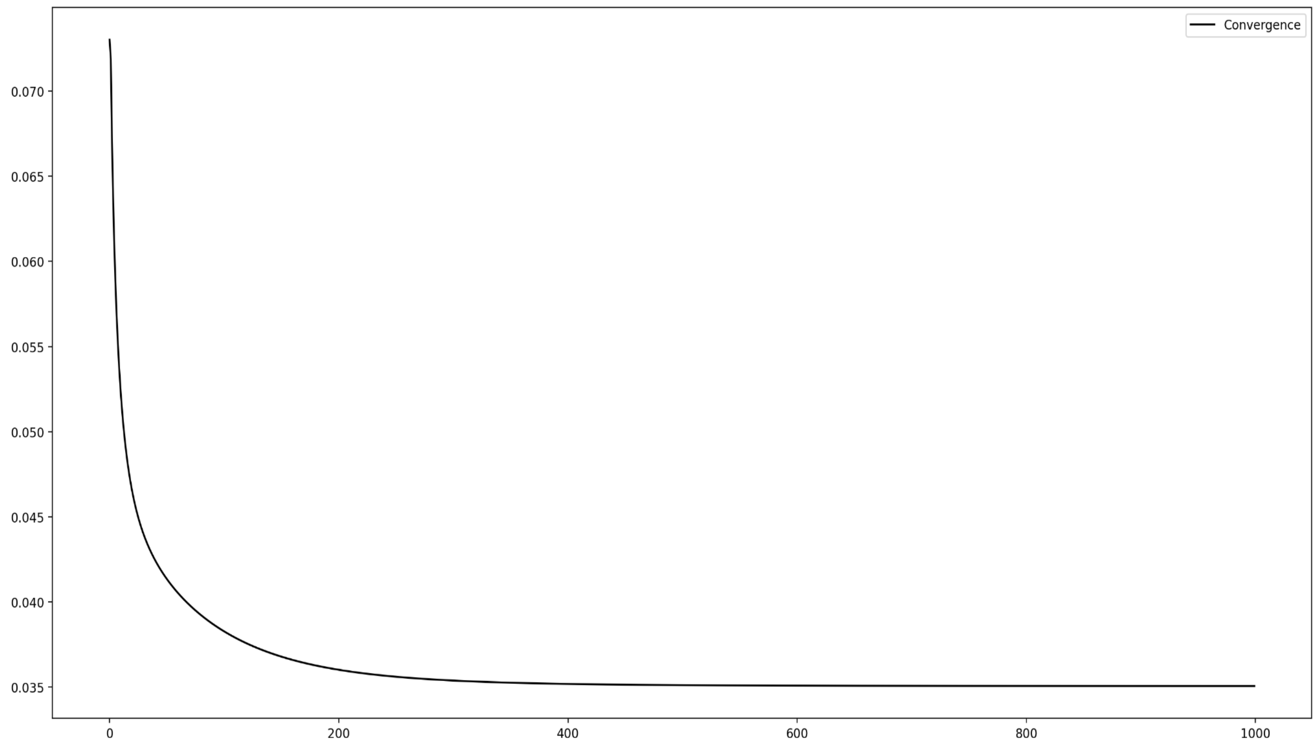
2) Discrete CMAC
I am using a 1D sin wave.
35 weights
70 : 30 - training to test data split

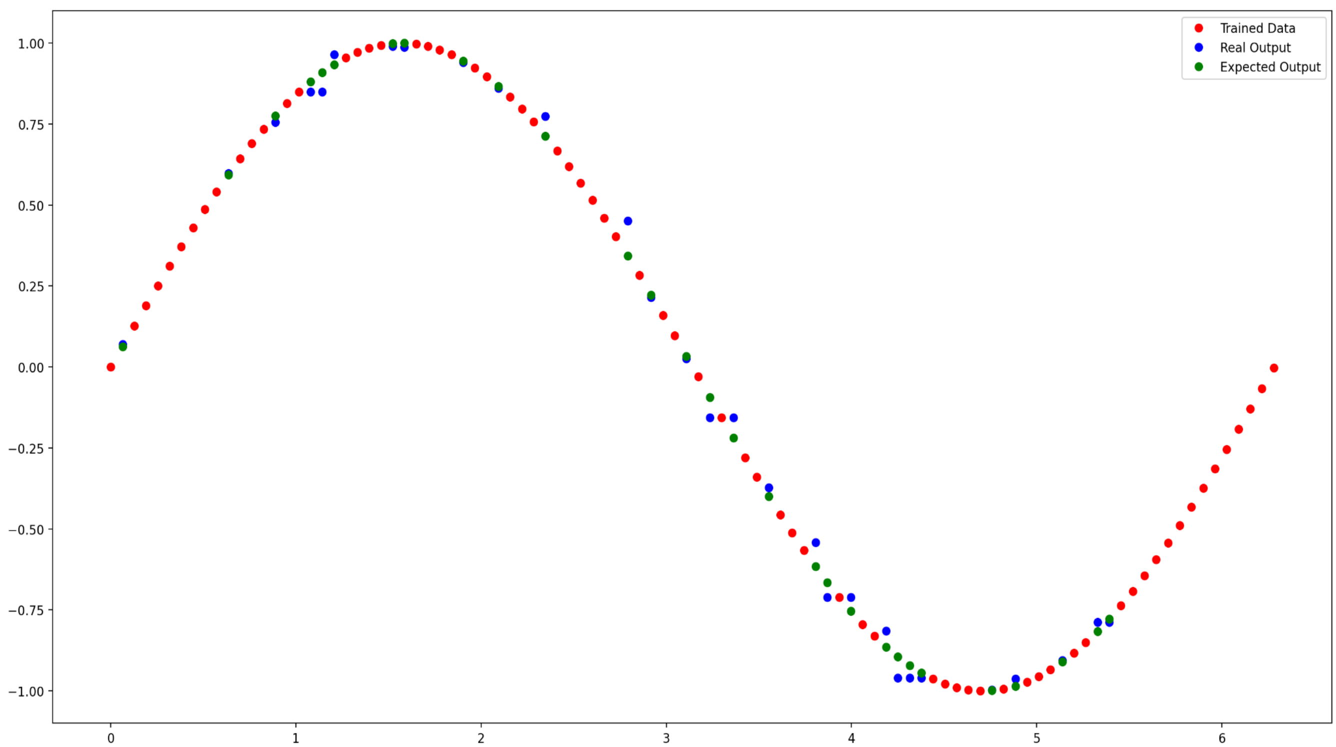
For, overlapping = 2 -> Accuracy = 0.68



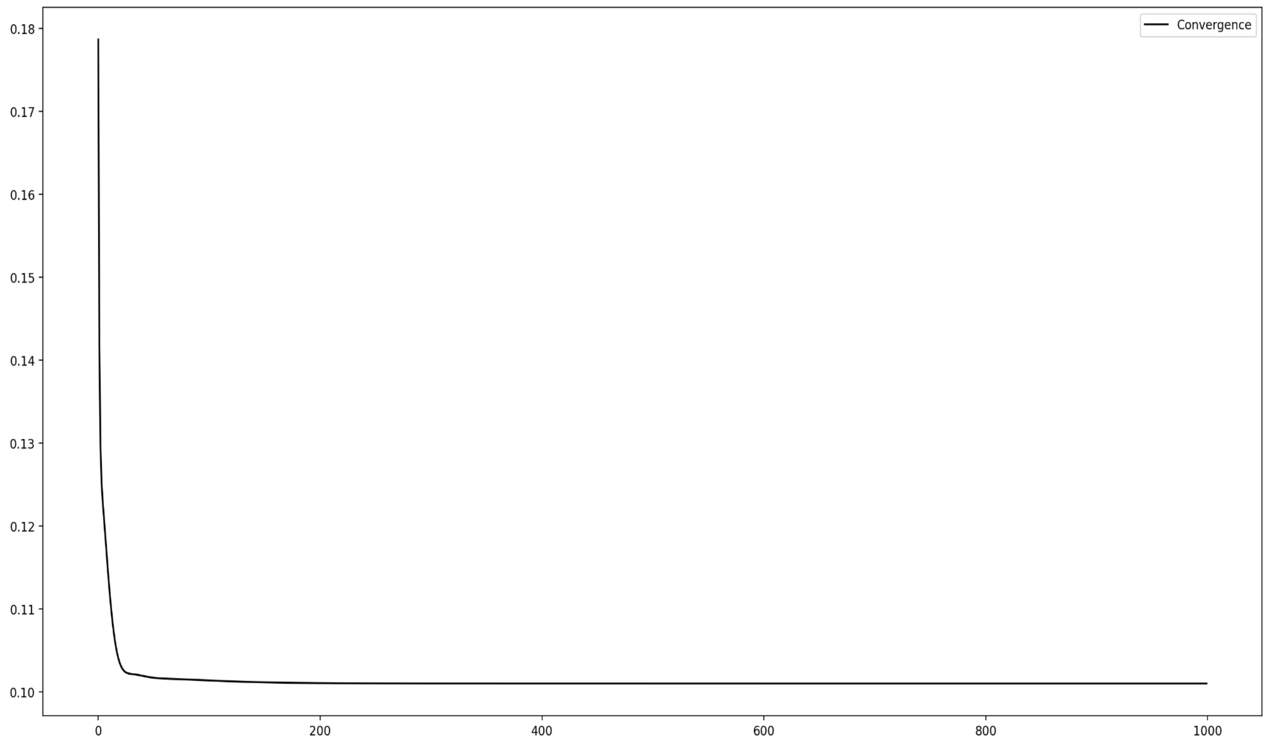


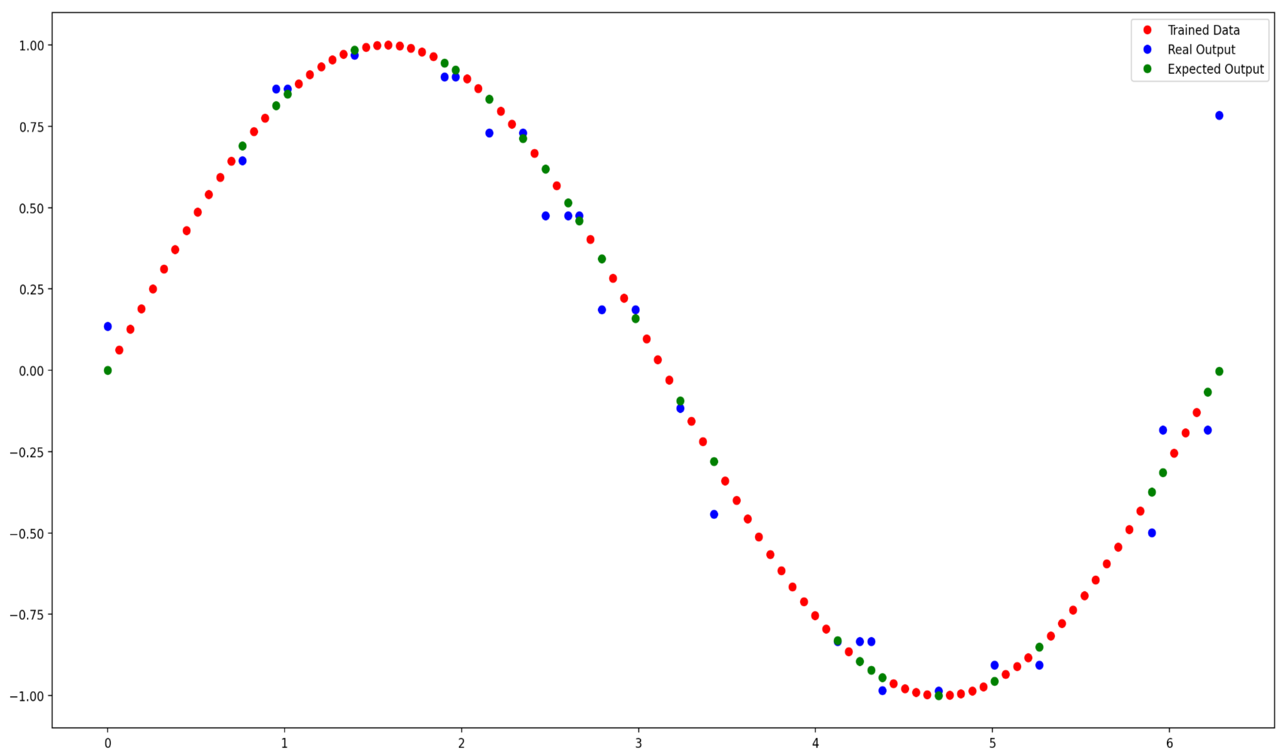
For, overlapping = 4 -> Accuracy = 0.87





For overlapping = 15 -> Accuracy = 0.48

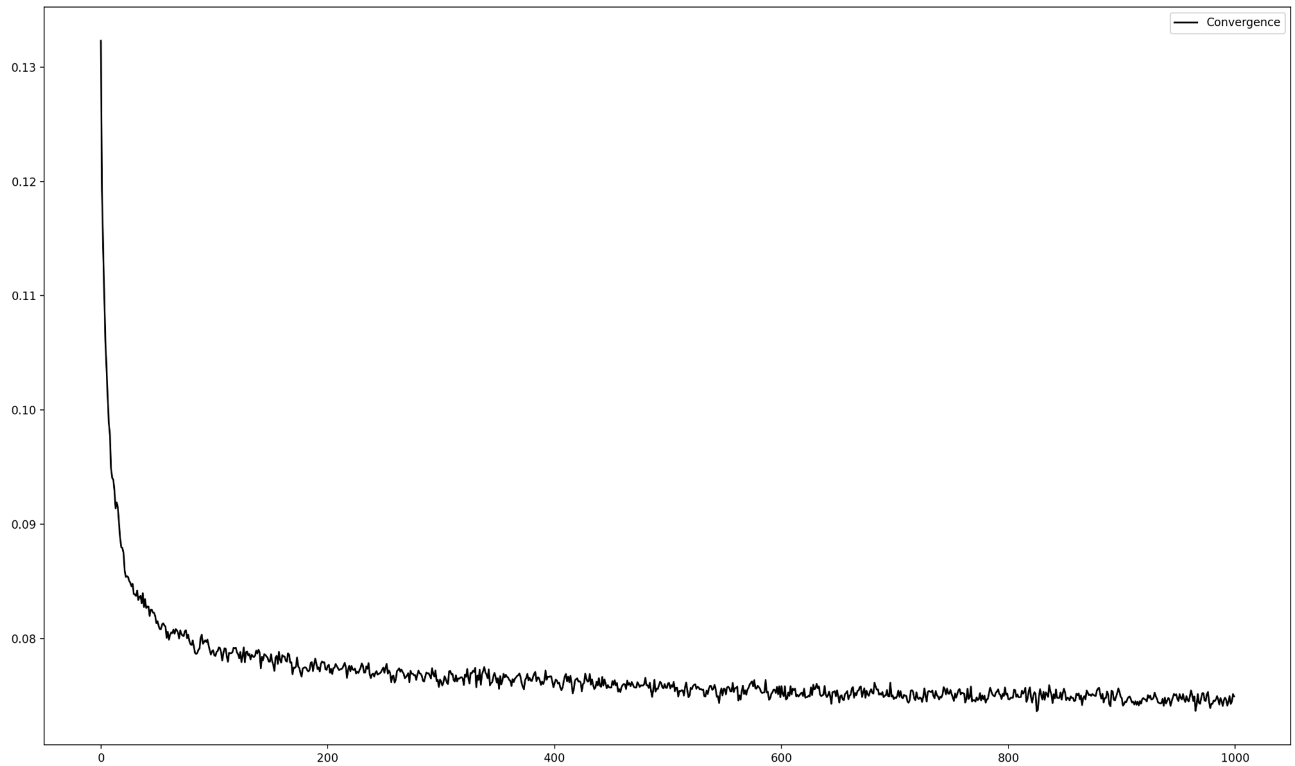


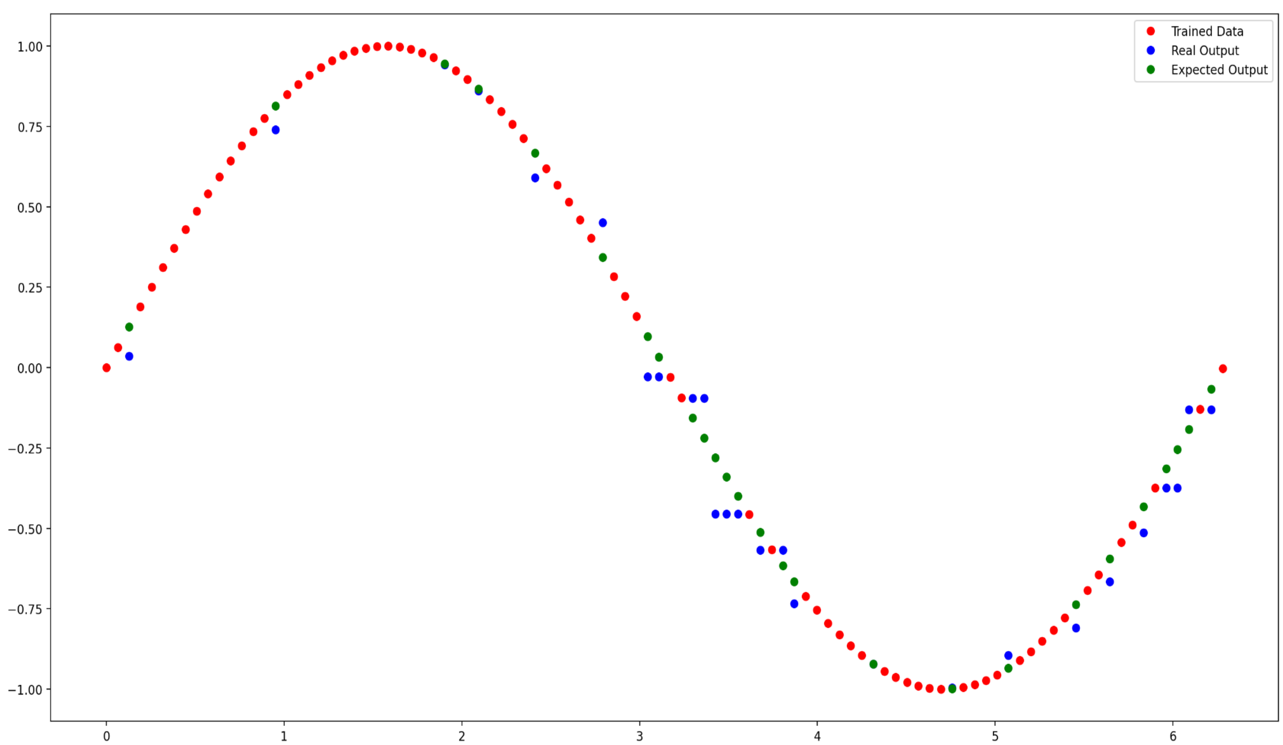


As we can see, with an **increase in overlapping area the testing accuracy first increases and then decreases**, but **with an increase in overlapping the convergence is slower**. This is because, overlapping of 1 is essentially 1 to 1 mopping and as we increase the overlapping it becomes many to 1 mapping. SO more complex the mapping is the convergence is that much slower.

3) Continuous CMAC
I am using a 1D sin wave.
35 weights
70 : 30 - training to test data split

For, overlapping = 4 -> Accuracy = 0.56





Continuous CMAC vs Discrete CMAC

The accuracy and convergence of a continuous CMAC are slightly higher than that of Discrete CMAC. If we want to choose one over the other, it depends on our application. The application will justify the trade-offs we can make. If the application can afford a delay of a few milliseconds in convergence time for a 1-2% accuracy gain, then we can employ continuous CMAC for that application.

---

4) If the number of **weights used in the model increases, we will have less overlapping**. Because of this, the network **will have better accuracy till a point and then starts to decrease rapidly**, the **convergence speed on the other hand will keep decreasing with increase in number of weights** but will have low accuracy.

   The Disadvantages of CMAC are: It is very memory intensive, the resources allocation is not automatic, Continuity and Differentiability of mapping are more difficult to generate.

---

5) Recurrent connections are mainly used in RNN (Recurrent Neural Network). In order to avoid the input as time, we can employ the logic of Finite State Machines to output the desired trajectory.

   In an FSM, the next output of the system is decided based on the current state and condition of the system. This is really helpful in scenarios where the network has to follow a trajectory. A simple flow diagram of FSM is given below. S_i represents each state and the fraction along each arrow represents the condition for transitioning from one state to another. We will have to decide upon some general conditions that acknowledge the change of state and alter the weights functions to take conditions into consideration while updating the weights.