Question 4 :

1. The architecture I am using for training the Lookup Table in part 2 is depicted below.

Hidden layer

(35 neurons)

Actions

Bearing

Distance

Y

x

Output Layer:

Q(s,a)

Hidden layer

(15 neurons)

Figure 1.

The input is a 5-tuple {x, y, distance ,bearing, actions} , the x and y positions of my robot , the distance to the enemy robot , the relative bearing to the enemy robot and the action it takes in the current state. The out is the corresponding Q value. To train the Neural network offline using the Q table from part 2 , I take the non-zero Q(s,a) values, feed it through the neural network and use stochastic gradient descent to minimize the error.

b)

Error curve using learning = 0.00009 and momentum term mu = 0.95 :

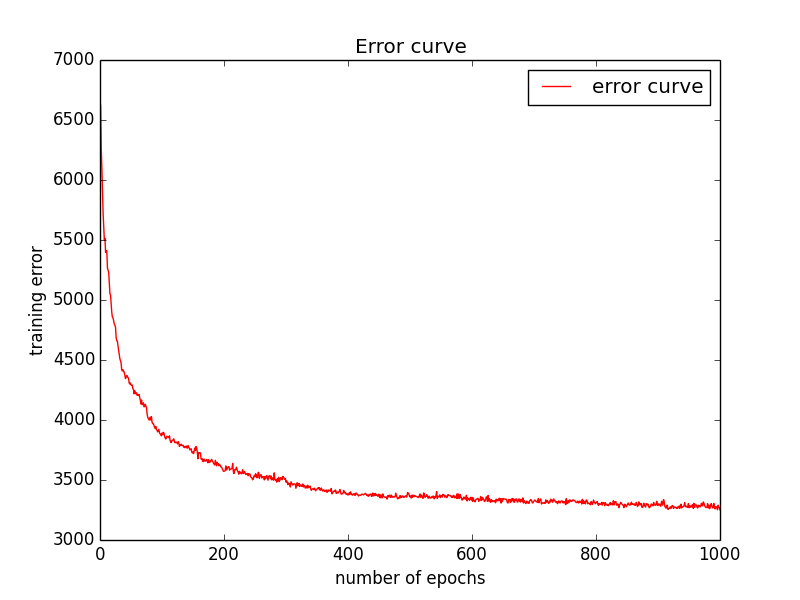


Figure 2(a).

Error curve using learning rate = 0.00009 and mu = 0.95 but using mean squared error :

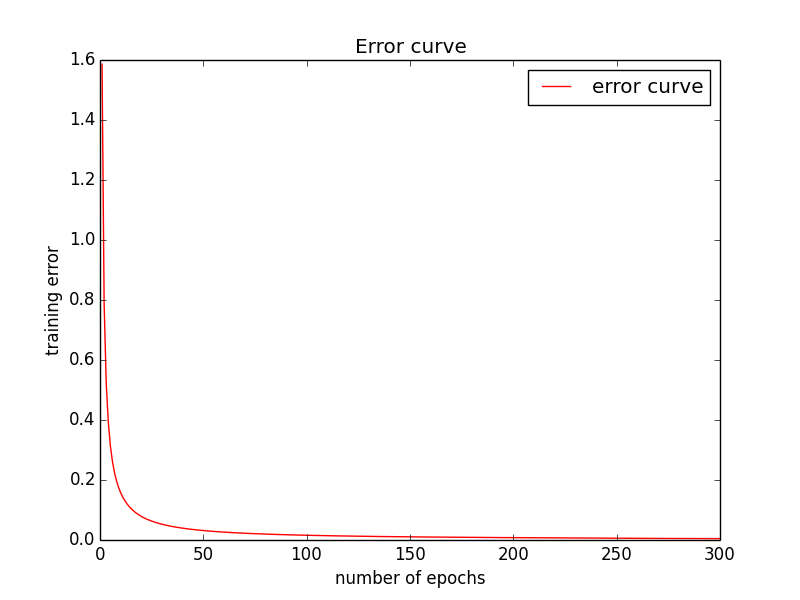


Figure 2(b).

Error curve using learning rate a = 0.00001 , and momentum mu = 0.95 :

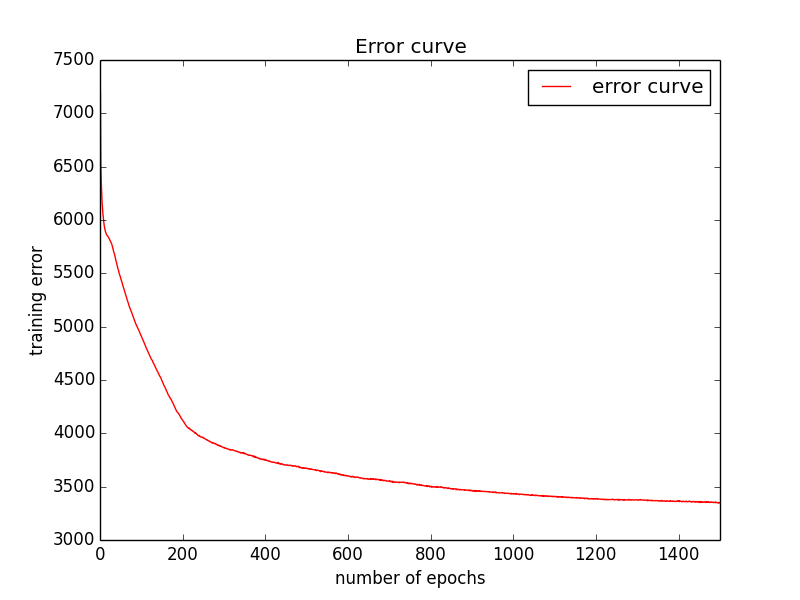


Figure 3.

As shown above , comparing using learning rate of 0.00009 and 0.00001 , larger learning rate results in faster convergence.

c)

A table comparing the input representation used by Lookup Table in part2 and the one used by neural nets is given below:

Lookup Table Neural Nets

|  |  |  |
| --- | --- | --- |
| X | 0  (integers from 0 to 7) | 67.45  (Double from 0 to 800) |
| Y | 3  (integers from 0 to 5) | 312.3  (Double from 0 to 600) |
| Distance | 5  (integers from 0 to 9) | 526.4  (Double from 0 to 1000) |
| Bearing | 1  (integers from 0 to 4) | 64.28  (Double from -180 to 180) |
| Actions | 0  (integers from 0 to 4) | 0  (integers from 0 to 4) |