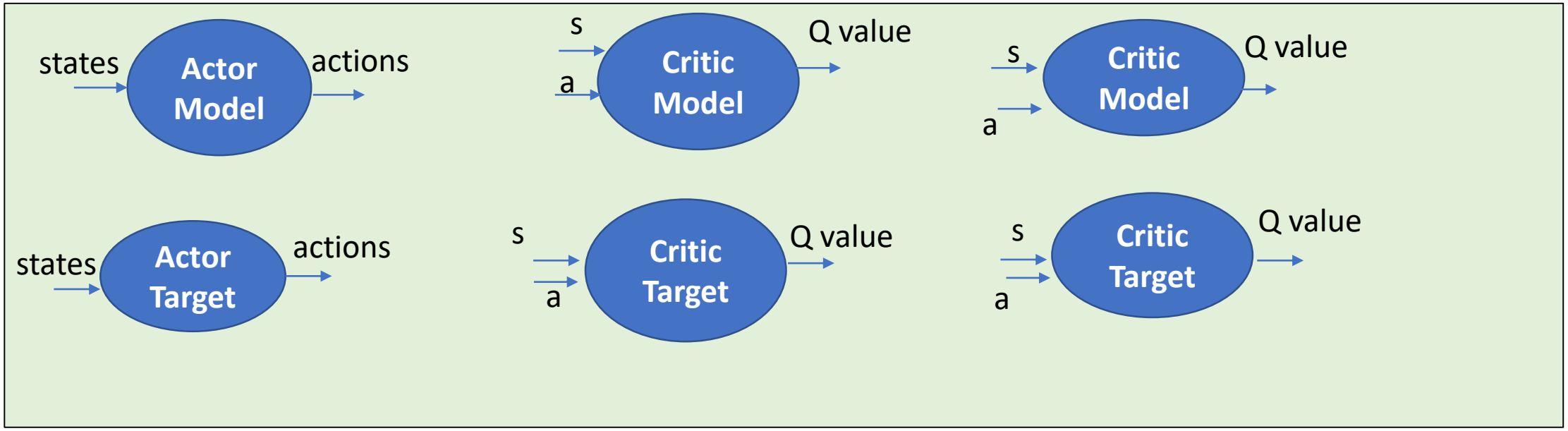
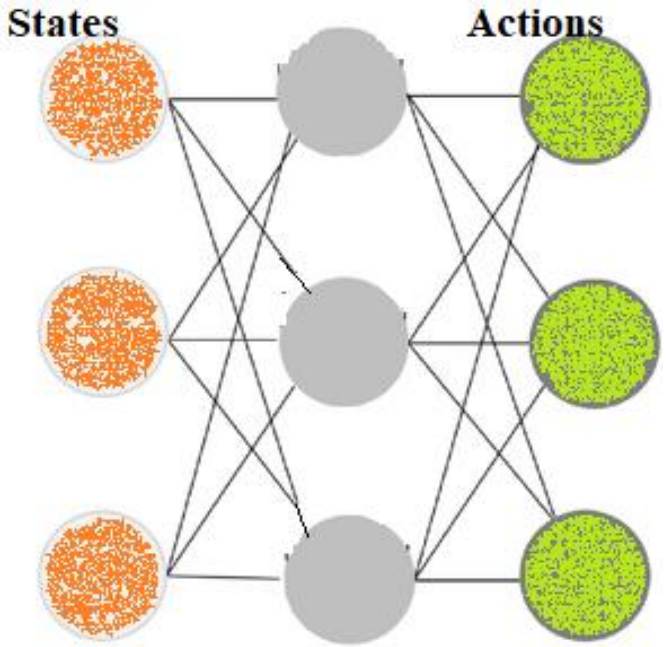
Twin delayed DDPG (T3D model) Architecture and implementation

- Actors learn policies
- Critics learn Q values
- 6 DNN models
- Why Delayed? We update our models at every step, but our target once every two steps.



Actor

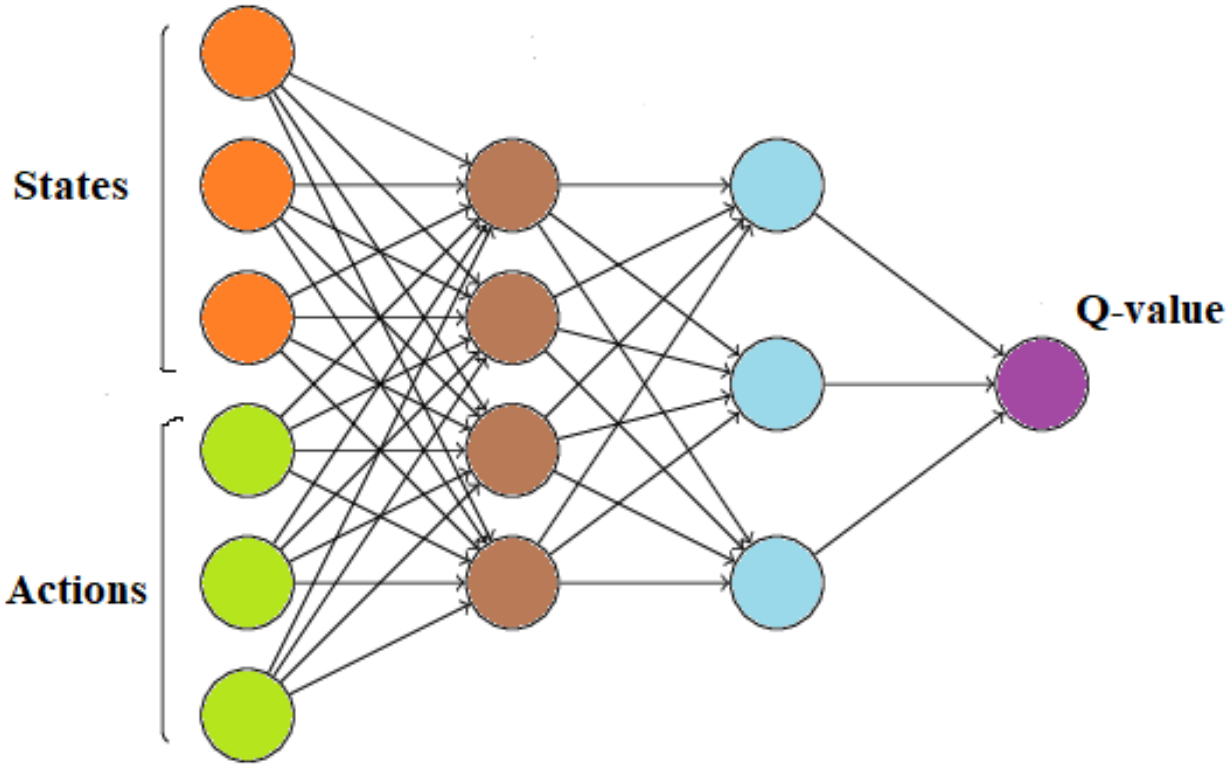


Actor Model

Actor Target

same DNN definitions

Critic

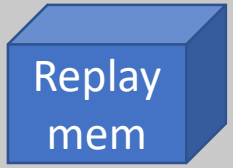


Critic Model 1
Critic Target 1
Critic Model 2
Critic Target 2.

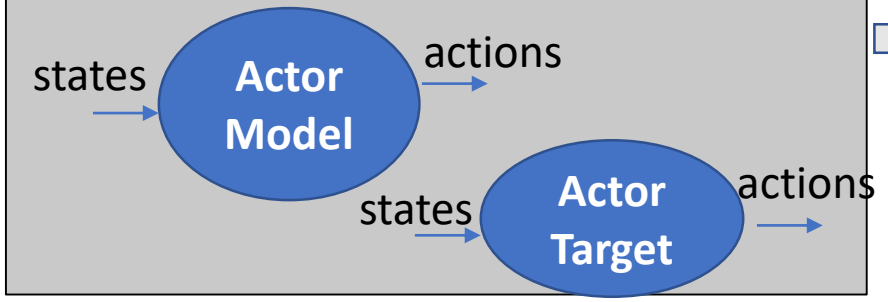
All 4 same DNN definition

Step 1 : Initialize replay memory, with each new transition of state, nxtState, action, reward

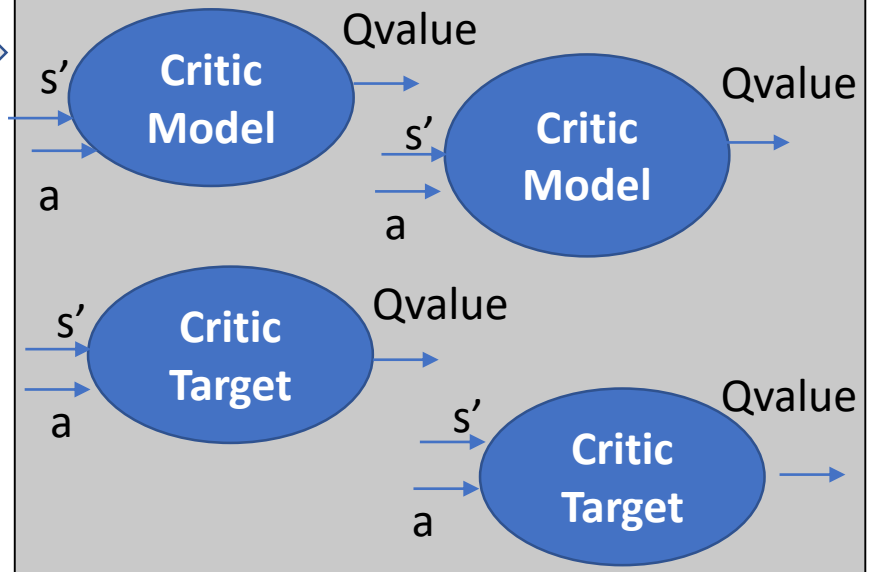
(s, s`, a, r)



Step 2 :
Build 2 kinds of Actor models

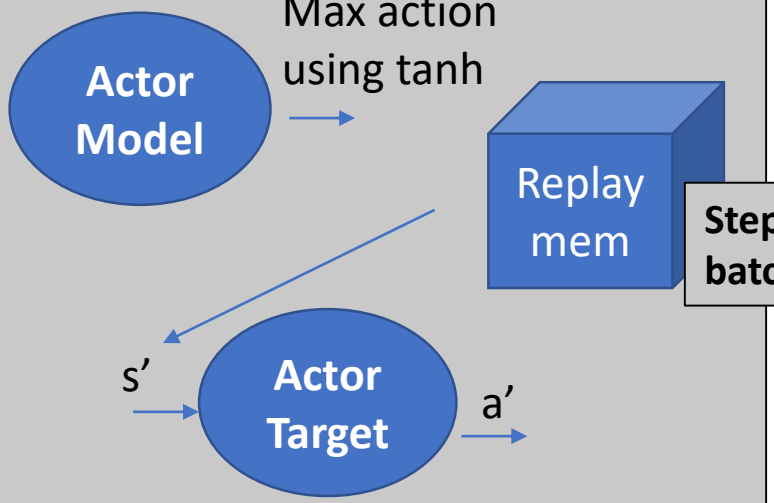


Step 3 : Build 2 sets of Critic models and Critic target



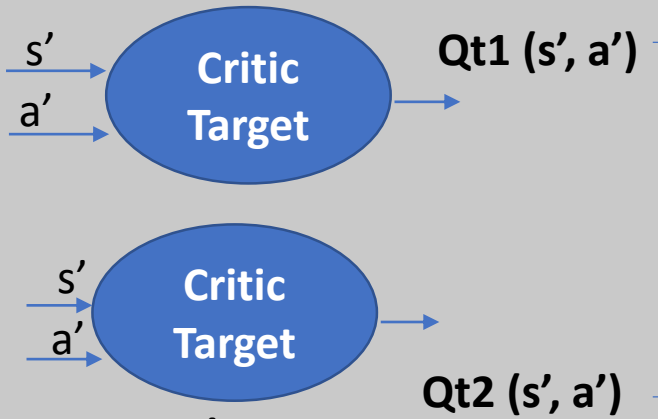
Step 4 : Sample a batch (s, s`, a, r)

Step 5 :



Step 6 : Add Gaussian noise to next action a`, clamp it

Step 7 :
2 Critic targets each take s', a' as input and return two Q values



Rest 4 remains as is from Step 5

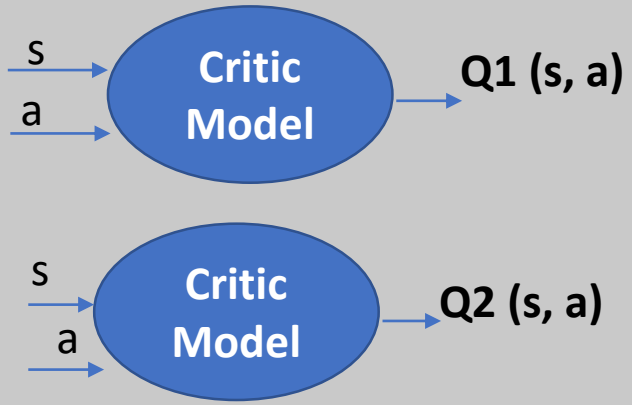
Step 8 :
Find Min (Qt1, Qt2)

Step 9 :
Calculate Q target of 2 Critic models
Qt = R + gamma* min(Qt1, Qt2)

To Step 10

From Step 9

Step 10 : 2 Critic Model each take (s,a) as input and return two Q values

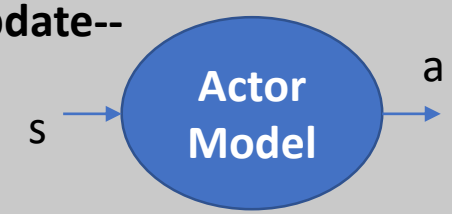


Rest 4 remains as is from Step 9

Step 11 :
Compute Critic Loss = MSELoss (Q1 (s,a), Qt) + MSELoss (Q2 (s,a), Qt) (use Qt from Step9)

Step 12 :
Backpropagate Critic Loss Update param of 2 Critic Models

Step 13 :
Update actor model Perform gradient ascent on O/P Update--



Step 14 :
Update weight of Actor Target by Polyak Averaging

Update

Step 15 :
Update weight of Critic Target by Polyak Averaging