Reinforcement Learning Algorithms used in Tic Tac Toe

June 21, 2019

1 TD(0) Learning

For episode = 1, M **do**
  Initialize a fresh game
  For t = 1, T **do**
    Play move $a_t = \begin{cases} \text{random move} & \text{with probability } \epsilon \\ \text{argmax}_a \tilde{V}(succ(s_t, a), \theta) & \text{for white} \\ \text{argmin}_a \tilde{V}(succ(s_t, a), \theta) & \text{for black} \end{cases}$
    Receive reward r_t
    Store the transition (s_t, s_{t+1}, r_t, a_t) in the replay buffer D
    Sample a random minibatch transition from D
    Set the TD-target: $y_t = \begin{cases} r_t & \text{if game terminates at step } t+1 \\ r_t + \gamma \tilde{V}(s_{t+1}, \theta) & \text{otherwise} \end{cases}$
    Perform a stochastic gradient decent step on $[y_t - V(s_t, \theta)]^2$ with respect to θ
    Every c steps set $\tilde{V} = V$
  End For
End For

γ is the discount factor and θ the neural network parameters.

2 DQN(λ)

procedure REFRESH(l)
    For transition $(s_t, s_{t+1}, r_t, R_t^\lambda, a_t) \in l$ processing back-to-front **Do**
        If terminal(s_{t+1}) **Then**
            Update $R_t^\lambda \leftarrow r_t + \gamma[\gamma R_{t+1}^\lambda + (1-\lambda)V(s_{t+1}, \theta)]$
        Else
            Get adjacent transition $(s_{t+1}, s_{t+2}, r_{t+1}, R_{t+1}^\lambda, a_{t+1})$ from l
        End If
    End For
End procedure

For episode = 1, M **do**
    Initialize a fresh game
    For t = 1, T **do**

$$\text{Play move } a_t = \begin{cases} \text{random move} & \text{with probability } \epsilon \\ \text{argmax}_a \tilde{V}(succ(s_t, a), \theta) & \text{for white} \\ \text{argmin}_a \tilde{V}(succ(s_t, a), \theta) & \text{for black} \end{cases}$$

        Receive reward r_t
        Append the transition $(s_t, s_{t+1}, r_t, R_t^\lambda, a_t)$ to L, where R_t^λ is arbitrary
        If terminal(s_{t+1}) **Then**
            REFRESH(L)
            Store L in D
        End If
        Sample a random minibatch transition from D
        Perform a stochastic gradient decent step on $[R_t^\lambda - V(s_t, \theta)]^2$ with respect to θ
        Every c steps REFRESH(D)
    End For
End For

γ is the discount factor and θ the neural network parameters.