R05922096 資工所 碩 1 李哲安

Neural Network Hw3 Report

Goal:

Modify the two functions get action.m and failed_update.m within demo codes for inserting ACE to solve the same problem as original demo codes, comparing the performance and briefly states your findings.

Q-Learning System:

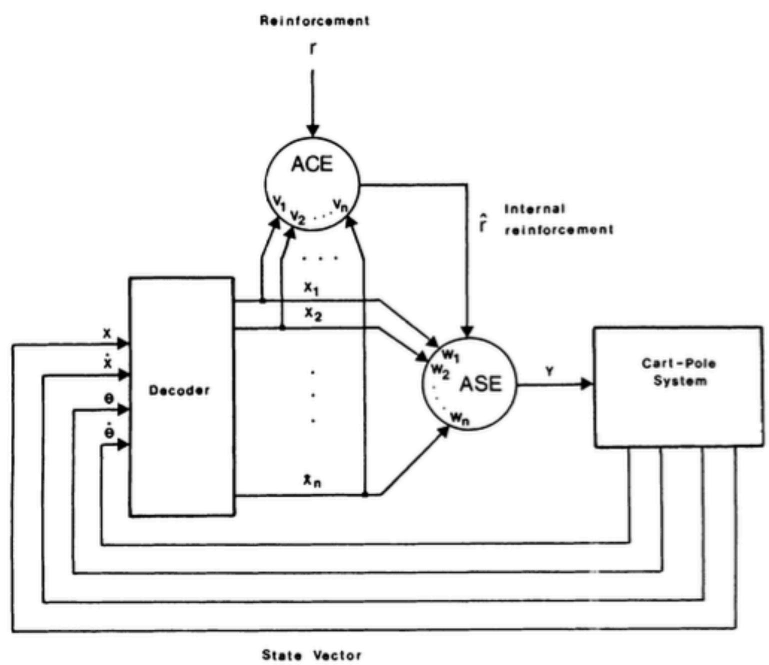


Fig. 3. ASE and ACE configured for pole-balancing task. ACE receives same nonreinforcing input as ASE and uses it to compute an improved or internal reinforcement signal to be used by ASE.

ACE:

評價系統 (adaptive critic element, ACE) 實作細節

$$
\begin{aligned}
p(t) &= \sum_{i=1}^{n} v_i(t)x_i(t) \\
v_i(t+1) &= v_i(t) + \beta\left[r(t) + \gamma p(t) - p(t-1)\right]\bar{x}_i(t) \\
\bar{x}_i(t+1) &= \lambda\bar{x}_i(t) + (1-\lambda)x_i(t) \\
\hat{r}(t) &= r(t) + \gamma p(t) - p(t-1)
\end{aligned}
$$

ASE:

動作系統 (associative search element, ASE) 實作細節

$$
\begin{aligned}
y(t) &= f\left[\sum_{i=1}^{n} w_i(t)x(t) + \mathrm{noise}(t)\right] \\
w_i(t+1) &= w_i(t) + \alpha r(t)e_i(t) \\
e_i(t+1) &= \delta e_i(t) + (1-\delta)y(t)x_i(t)
\end{aligned}
$$

Implementation:

在matlab中實作 ACE.m 和 ASE.m 兩個function並且加入到原本的get_action.m 和 failed_update.m 中，並觀察修改前後的學習效果。

Code: (ACE.m and ASE.m)

ACE.m

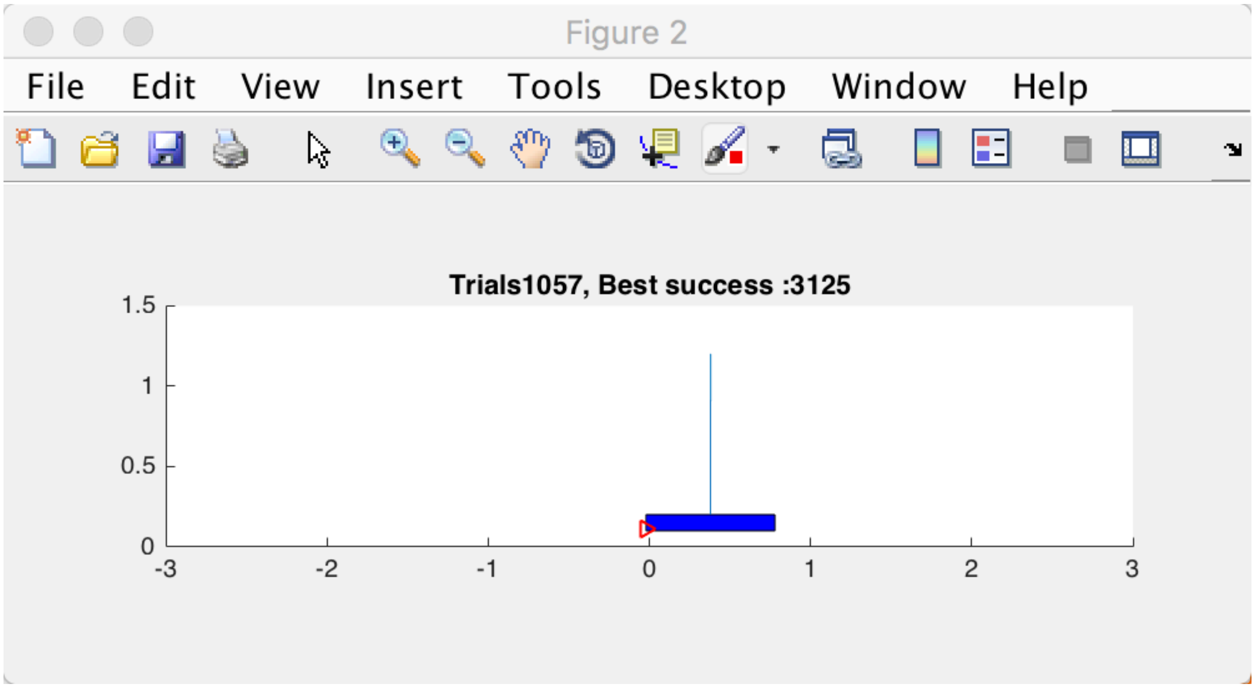
```
function [reward_hat, p ,v_val] = ACE(learn, decay, reward, gamma, p_before, v_val, cur_state)
    global NUM_BOX
    if (reward == -1)
        p = 0;
    else
        p = v_val(cur_state, 1);
    end


    reward_hat = reward + gamma*p - p_before;
    for i = 1:NUM_BOX
        v_val(i, 1) = v_val(i, 1) + learn * reward_hat * v_val(i, 2);
    end

    for i = 1:NUM_BOX
        v_val(i, 2) = decay * v_val(i, 2);
    end
    v_val(cur_state, 2) = v_val(cur_state, 2) + (1-decay);
end
```

ASE.m

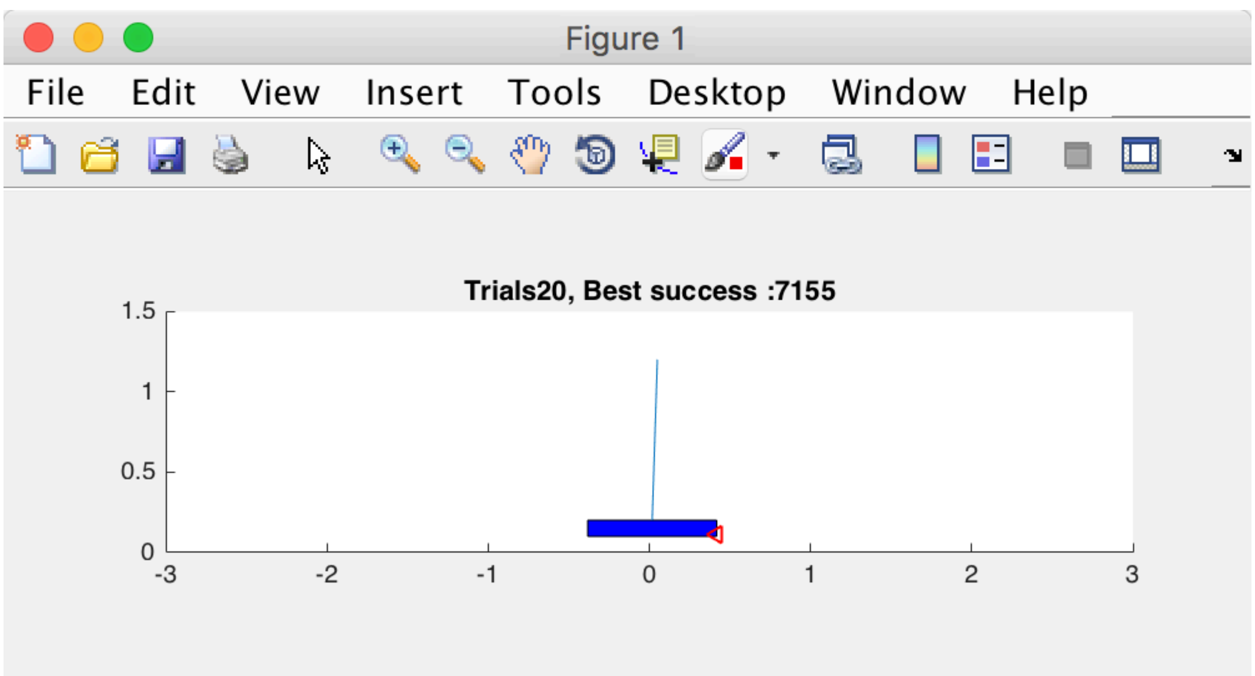
```
function [y, q_val] = ASE(learn, decay, reward, q_val, cur_state)
    global BETA NUM_BOX

    noise = rand*BETA;

    x = q_val(cur_state, 1) + noise;
    if (x + noise >= 0)
        y = 2;
    else
        y = 1;
    end

    for i = 1:NUM_BOX
        q_val(i, 1) = q_val(i, 1) + learn * reward * q_val(i, 2);
    end

    for i = 1:NUM_BOX
        q_val(i, 2) = decay * q_val(i, 2);
    end
    q_val(cur_state, 2) = q_val(cur_state, 2) + (1-decay) * ((y-1)*2-1);
end
```

Compare Result:

Old (without ACE):



New (added ACE):



Conclusion:

由以上兩張圖片結果可以看出，在原本的Q-Lerning版本中(第一張圖)第1057次Trials時Best Success為3125分，而加入ACE後的Q-Learning版本中(第二張圖)第20次Trials時Best Success為7155分，學習效果有明顯的進步。從實驗結果可以得出，我們可以有效的透過ACE得到一個較好的reward，並且利用這個新reward來提升Q-Learning的學習效果。