

HW3 Report

R06922048 陳柏堯

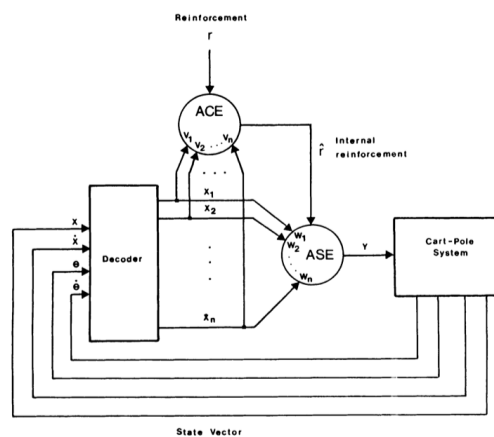
程式執行環境：Matlab R2017b

實作方式：

根據助教所提供的 Q_demo 作修改。

對於 ACE 和 ASE 的概念一部分參考 Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems，另一部分參考 <http://mropengate.blogspot.tw/2016/12/q-learning-openai-gym-cart-pole-system.html> 的“Q-learning 與 類神經網路：用 OpenAI gym 模擬木棒台車平衡問題 (cart-pole system)”的資料。

其中主要說明了一下 Reinforcement Learning 的架構，如下：



在 Cart-Pole System 的 162 維的 x 放入到 ACE 得到新的 Reward 的值，用於訓練新的 Reward Function，然後再用 ASE 吃入 Reward 值訓練出新的 Action y ，從而實現整個 Reinforcement learning 系統。

ACE 的公式：

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

ASE 的公式：

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

在 Matlab 中，我分別實作了 ACE.m 和 ASE.m，然後在 get_action.m 中分別

接入 ACE.m 和 ASE.m, 由於助教提供的 code 沒有看到 failed_update.m, 於是我直接在 Q_demo.m 當中的 if (box== -1) 下面直接接入 ACE.m 和 ASE.m。

使用的參數用默認值, 如下:

success update 中:

ACE learn_rate = 0.5

ACE Decay = 0.8

ACE Gamma = 0.95

ASE Learn_rate = 1000

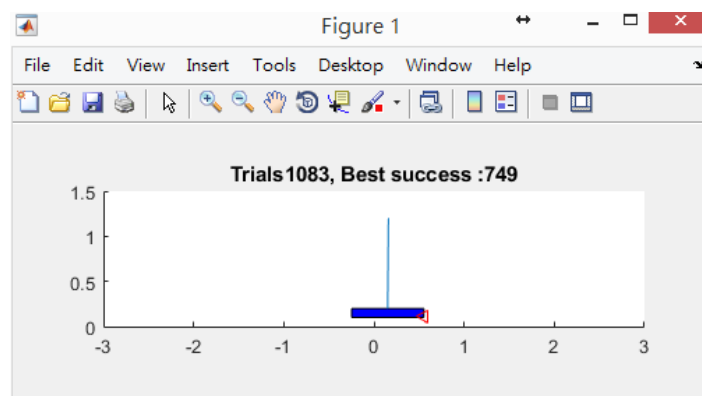
ASE Decay = 0.9

Failed update 中:

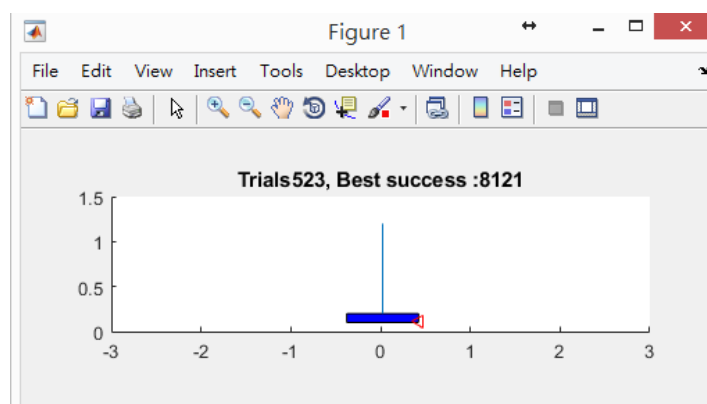
ACE Reward = -1

實驗結果:

原始 Q-learning:



實作 ACE 和 ASE 之後:



比較可以發現, 實作 ACE 和 ASE 之後, 523 Trials 之後最高分數可以到 8121, 而實作前經過 1083 次 trail 最高分只有到 749 分。