用領域知識提升 DQN 學習效率之研究

何佩蓁[1]
jennyho0221@gmail.com

楊宗憲[2]
chyang@cyberon.com.tw

[1] 台北市私立東山高級中學
[2] 賽微科技股份有限公司

***摘要*—近年來強化學習結合神經網路優異的函數逼近能力為人工智慧領域帶來重大的突破，前幾年 Deep Q Networks (DQN) 技術在多款 Atari 2600 遊戲的表現上已超越了人類玩家。在本論文我們使用 DQN 來訓練無敵的井字遊戲人工智慧，其中使用多層感知器來近似 Q 函數，為了改善 DQN 訓練樣本運用效率不佳的問題，我們嘗試用已知的領域知識介入 DQN 的學習過程，包括在經驗回放中額外增加有價值決策的數量、以及提前給予特定種類盤面獎勵值，實驗顯示先手和後手的訓練效率可分別提升 28% 和 30%，在訓練樣本數減半的情況下，相較於對照組，加入領域知識介入 DQN 學習可讓訓練目標達成率提升近一倍。**

關鍵字—神經網路、強化學習、DQN、經驗回放

一、 緒論

強化學習適用於訓練電腦解決馬可夫決策過程問題。井字遊戲可視為一種馬可夫決策過程，每個下棋決策表示為 (s,a,r,s')，目前盤面狀態為 s，決定下棋的位置為行動 a，最多有 9 種行動，對手下子後產生的新盤面狀態 s'，獎勵值為 $r=R(s')$ 僅與新盤面狀態有關，終局贏則獎勵為 1，輸為 -1，平手為 0.1, 其他非終局皆為 0。給予平手正向獎勵可促成電腦在無法贏棋時盡可能追求合局。

本研究的目標是訓練出無敵的井字遊戲人工智慧，驗證的方法是 DQN 訓練出的人工智慧與隨機程式對奕一萬局的結果必須是 DQN 贏或雙方平手。我們使用多層感知器(MLP, Multilayer Perceptron)來近似 Q 學習法中的 Q 函數，參考[1]的做法，把 $Q(s,a)$ 改成 $Q(s)$，讓 MLP 一次輸出所有行動的 Q 值，如圖 1 所示，其中 θ 為 MLP 的參數。MLP θ 的輸入層為盤面狀態 s，盤面位置下子 X 的二進位表示為 01，O 為 10，未下子的空格為 00，圖 1 下方為範例盤面和其二進位表示。此 MLP 共有 18 個二進位輸入神經元，接著是一層隱藏層包含 36 個神經元，激發函數為 tanh(x)，最後輸出層含 9 個神經元，使用線性激發函數 $f(x)=x$，輸出棋盤上 9 個下子位置的 Q 值 $[q_0,q_1,....,q_8]$。

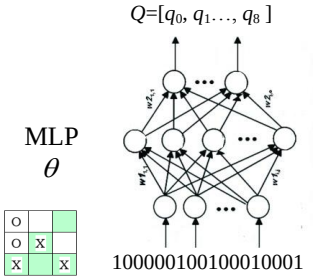

圖 1. MLP 近似 Q 函數

DQN 的訓練方法是跟隨機下子程式大量對弈，收集電腦每次下棋過程的決策 (s,a,r,s')，用 θ 算出在盤面 s 每個下子位置的 Q 值，如方程式(1)所示

$$MLP(s;\theta)\rightarrow Q=[q_0,q_1,....,q_8] \quad (1)$$

以及估算其和訓練目標 Q^t 值的誤差，並依此更新 θ。對下棋決策 (s,a,r,s')，在盤面 s 下子於位置 a 的的未來折扣獎勵目標值 q_a^t 可用 Bellman 方程式(2)估算

$$q_a^t=\begin{cases} r & \text{if } s' \text{ is terminal} \\ r+\gamma \max Q(s') & \text{else} \end{cases} \quad (2)$$

參考[2]的做法，我們將決定下子位置的 θ、跟計算訓練目標 q_a^t 的 θ^- 分離，每隔一段時間才更新一次 $\theta^-\leftarrow\theta$，由於訓練過程中 θ 會被持續更新，改用較穩定不變的 θ^- 估算訓練目標值 q_a^t，使其較不會產生劇烈變動，可讓訓練 θ 的過程更加穩定而較易收斂。方程式(2)中的 $\max Q(s')$ 可由以下方程式(3)(4)算出：

$$Q(s')=MLP(s';\theta^-)\rightarrow Q'=[q_0',q_1',....,q_8'] \quad (3)$$

$$\max Q(s')=\max_{a'}(q_{a'}',s'[a']\notin\{O,X\}) \quad (4)$$

在(4)中 $\max Q(s')$ 即為新盤面 s' 的空格位置中取出最佳者。盤面 s 的訓練目標 Q^t 即為 Q 但把位置 a 的值 q_a 更改為 q_a^t，亦即 $Q^t\leftarrow Q$、$Q^t[a]\leftarrow q_a^t$，最後將 (s,Q^t) 加入 θ 的訓練資料集，即可透過反向傳播(backpropagation)以 q_a^t 和 q_a 的誤差訓練 θ。

DQN 的缺點之一是訓練樣本運用效率不佳，常需要大量的訓練資料才能得到夠好的結果。為了改善此問題，我們嘗試用已知的領域知識(domain knowledge)介入 DQN 的學習過程，包括：

(一) 在經驗回放(Experience Replay)中額外增加有價值決策的數量，即決定輸贏的關鍵決策，以提升該決策被訓練到的機率。

(二) 提高特定種類盤面的獎勵值，刻意引導 DQN 加強學習該類盤面。

本論文接下來介紹一些提升 DQN 學習效率的相關研究，並描述我們以領域知識介入 DQN 訓練的做法。實驗的部分除了量化評估領域知識對訓練效率和穩定度的提升，更進一步用圖形檢視在訓練過程中對棋力變化的影響。最後提出我們從實驗結果中所獲得的結論。

二、相關研究

人工智慧的發展一直以超越人類的能力為重要里程碑。2013 年 Deep Mind 團隊發表了 DQN [1][2]，結合強化學習和神經網路優異的函數逼近能力，第一次在 Atari 2600 遊戲環境中表現超越了人類。其使用了經驗回放(Experience Replay)存放最近一段時間所收集到的資料，再持續從中隨機挑選來訓練神經網路。使用經驗回放的目的是讓過往經驗能多次被使用，以增加訓練資料的利用率，而隨機挑選打散了資料的相依性，可讓神經網路這類非線性的逼近函數訓練效果較佳。

然而，經驗回放中的資料對訓練 DQN 而言並非同等重要，Deep Mind 接著提出了優先化經驗回放(Prioritized Experience Replay)[3]的改良做法，以 TD (temporal-difference)誤差為基礎自動計算經驗回放中每筆資料的優先權，誤差越大優先權越高，越容易被挑選為訓練資料。實驗顯示此改良法在 49 個遊戲中有 41 個的表現超越了無優先權的經驗回放。

優先化經驗回放為一通用的方法，並未考慮其所處理的問題為何，理論上能適用於解決所有的問題。但我們相信針對個別問題，若能善用該問題領域的專家知識輔助 DQN 學習，不僅能提升其學習效率，減少對訓練資料的需求量，同時在訓練資料量有限的情況下，專家的知識應可協助訓練出更佳的結果。

三、領域知識介入 DQN 學習

強化學習常遭遇著獎勵稀疏(sparse reward)的問題，導致訓練資料的運用效率不佳，常需大量的訓練資料方能訓練出夠好的效果，但對很多現實問題而言，訓練資料的取得並非那麼容易或相對昂貴。本論文試著研究如何減少 DQN 對訓練資料的需求，不同於[3]的優先化經驗回放對訓練資料自動設定優先權，本論文用已知的領域知識為基礎介入 DQN 的學習，在訓練井字遊戲 DQN 過程中，我們嘗試增加有價值的下棋決策放入經驗回放中的數量，以提高其被選為訓練資料的機會，並量化評估此舉對 DQN 學習效率和穩定度的影響。本論文所定義的有價值下棋決策 (s,a,r,s')，為 s' 為終局盤面時的決策。

除此之外，在井字遊戲中，有一種能贏得棋局的關鍵盤面稱為「雙活路」盤面，如圖 2 範例所示，綠色標示了先手 X 雙活路在盤面上的位置，此時不論後手在哪個位置下 O，都已無法挽回輸局的頹勢。此種盤面出現在後手的機會很低，本論文僅對訓練先手時修改獎勵函數 $R(s')$ 額外提供「雙活路」獎勵，刻意引導先手加強學習該類盤面以提高學習效率及獲勝率。

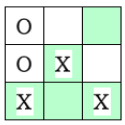

圖 2. 雙活路盤面

以下我們以虛擬碼（pseudocode）描述如何用 DQN 訓練井字遊戲先手下 X，至於訓練後手下 O 的演算法因與先手下 X 相當近似，本論文就不再贅述。

在演算法 1 中，第 3 行用 **for** 迴圈執行 T 次訓練棋局，第 5 行用 **while** 迴圈對每場棋局產生一系列的下棋決策 (s,a,r,s')，直到抵達終局為止。面對目前盤面狀態 s，第 6 行依 $\varepsilon-greedy$ 策略產生行動 a、第 7 到 12 行產生新盤面狀態 s'、第 13 行依 s' 產生獎勵值 r，並於第 14 行將以上步驟得到的 (s,a,r,s') 加入經驗回放中。注意在第 7 行先手於盤面 s 的 a 位置下 X 後所得到的盤面並非必然為 s'，若其非終局盤面，須在第 11 行等對手下 O 後，所得到新盤面方為狀態 s'。本論文的實驗重點之一在第 15 到 17 行，第 15 行檢測目前決策 (s,a,r,s') 中 s' 是否為終局盤面，以判斷是否為有價值決策，並於第 16 行增加其放入經驗回放的數量。第 18 行所呼叫的函式 $UpdateQ(\theta,\theta^-)$ 用以更新 MLP 的參數 θ，則詳述於演算法 2 中。

演算法 1：用 DQN 訓練井字遊戲先手下 X

1: 以隨機值初始化 MLP 參數 θ
2: $\theta^- \leftarrow$ 複製 θ
3: **for** *episode* $= 1, 2, \ldots, T$ **do**
4: $s \leftarrow$ 初始化空盤面
5: **while** s 非終局盤面 **do**
6: $a \leftarrow$ 以 $\varepsilon-greedy$ 方式隨機或 $\arg\max MLP(s;\theta)$ 決定下 X 位置
7: *temp* ←先手於盤面 s 的 a 位置下 X
8: **if** *temp* 為終局盤面 **then**
9: $s' \leftarrow$ *temp*
10: **else**
11: $s' \leftarrow$ 對手在盤面 *temp* 隨機下 O
12: **end if**
13: $r = R(s')$
14: 將 (s,a,r,s') 加入經驗回放
15: **if** s' 為終局盤面 **then**
16: 增加 (s,a,r,s') 加入經驗回放的數量
17: **end if**
18: $\theta \leftarrow UpdateQ(\theta,\theta^-)$
19: $s \leftarrow s'$
20: **end while**
21: 每隔若干次 *episode* 更新 $\theta^- \leftarrow$ 複製 θ
22: 每隔若干次 *episode* 衰減 $\varepsilon \leftarrow 0.9\times\varepsilon$
23: **end for**

演算法 2 實做 $UpdateQ(\theta,\theta^-)$ 函式，輸入參數 θ 用於計算盤面各下子位置的 Q 值，θ^- 則用於估算其訓練目標 Q^t 值，最後於第 17 行回傳更新後的 θ。每次呼叫 $UpdateQ(\theta,\theta^-)$ 時，在第 2 行會從經驗回放中隨機挑選若干筆資料組成 *minibatch*，其中須包含最新加入的該筆資料，以保證每筆訓練資料皆被使用到。第 4 行 **for each** 迴圈為每筆被挑選到資料計算訓練目標 Q^t 值（第 5 到 13 行），最後在第 16 行呼叫 $TrainMLP(\theta,minibatch)$ 函式，用挑選到的資料對 θ 進行 MLP 訓練並更新參數。

演算法 2：更新 MLP 的參數 θ

function $UpdateQ(\theta,\theta^{-})$
  經驗回放隨機挑選若干筆資料放入 M
  $minibatch \leftarrow \phi$
  for each (s,a,r,s') in M **do**
    $MLP(s;\theta) \rightarrow Q=[q_0,q_1,...,q_8]$
    if s' 為終局盤面 **then**
      $q_a^t = r$
    else
      $MLP(s';\theta^{-}) \rightarrow Q'=[q'_0,q'_1,...,q'_8]$
      $q_a^t = r+\gamma \max_{a'}(q'_{a'}, s'[a'] \notin \{O,X\})$
    end if
    $Q^t \leftarrow Q$
    $Q^t[a] = q_a^t$
    將 (s,Q^t) 加入 $minibatch$
  end for
  $\theta \leftarrow TrainMLP(\theta, minibatch)$
  return θ

四、實驗

本研究的目的是用 DQN 訓練出無敵的井字遊戲人工智慧。我們用隨機程式讓先手和後手互相對弈一萬局，先手的獲勝率為 58%，後手獲勝率為 29%，平手 13%，由此可知後手有天生劣勢，或許需要較多訓練。經過一些初步實驗，我們選定先手的訓練局數為兩萬局，後手為十萬局。本論文實驗的各項參數設定如下：MLP 的學習率 α 起始值為 0.01，訓練過程衰減 50 次，每次衰減為原來的 90%，$\varepsilon-greedy$ 的 ε 起始值為 1，衰減過程跟 α 一樣， DQN 未來折扣獎勵的折扣率 γ 為常數 0.9，經驗回放的容量為 2048 個，每次隨機取出 32 筆資料組成 minibatch 來訓練 MLP。由於 DQN 的訓練結果好壞常跟 MLP θ 的初始參數有很大的關係，為避免運氣成分左右實驗結果，我們對每種欲進行研究的設定值做十次相同的實驗，每次實驗皆重新隨機產生 θ 的初始參數，再將十次實驗結果取平均值。訓練過程每百次訓練棋局儲存一次 θ，並和隨機程式對弈一萬次進行檢測，看看是否已達到無敵的目標。

實驗一：增加複製有價值決策到經驗回放中對 DQN 學習效率的影響

對有價值的下棋決策，我們額外增加複製 1 至 5 筆到經驗回放中，測試此舉對 DQN 學習效率的影響，結果如表 1 所示。對於先手，增加複製有價值決策到經驗回放的效果不大，但對後手則有顯著的幫助。先手由於先天的優勢，θ 的初始參數距離無敵本來就不遠，且訓練過程贏的機率一開始就較高，能以較少的棋步贏得棋局，因此經驗回放內的決策必然有較高的比例 S' 為終局盤面，額外再增加複製數對加速 θ 到無敵的幫助不大。但後手先天處於劣勢，增加複製有價值決策到經驗回放能提高其被選為訓練資料的機率，能使後手更快熟悉如何往獲勝盤面邁進，同時避開敗北棋步，而有效地提升了學習效率。在我們的實驗中，後手在複製數增加到 4 時可讓 DQN 在最少局數達到無敵，和複製數為 0 的對照組相比，可減少高達 30% 的訓練局數。

表 1. 先手和後手訓練到無敵的平均局數(單位為千局)

增加複製數	0	1	2	3	**4**	5
先手訓練局數	9.7	8.9	11.0	10.5	11.3	9.6
後手訓練局數	82.5	68.9	72.0	62.0	**58.0**	65.6

我們從十次實驗結果中挑選出跟平均值最接近的一組來繪製訓練過程的棋力變化，如圖 3 所示，橫軸為訓練棋局數，以百局為單位，縱軸為跟隨機程式對奕一萬次輸的局數，以 50 均線(moving average)觀察其走勢。增加複製有價值決策雖然在表 1 看不出對訓練先手到無敵有顯著的幫助，但從圖 3a 可觀察出先手棋力在訓練過程到一半左右時，有增加複製數的實驗組皆超越了無複製的對照組(藍色實線)。圖 3b 也可看出後手訓練到過程的 20% 至 50% 時，實驗組的棋力也都超越了對照組，特別是增加複製數為 4 時(紫色實線)，在訓練過程的 20% 就持續地超越了對照組。由以上實驗結果可得知，增加複製有價值決策到經驗回放中，不論對先手或後手，都能有效地提升 DQN 學習效率。

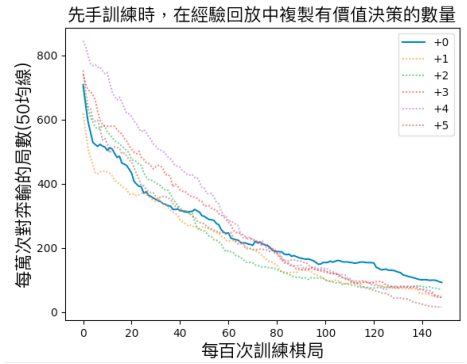

圖 3a. 先手訓練過程中棋力進步狀況

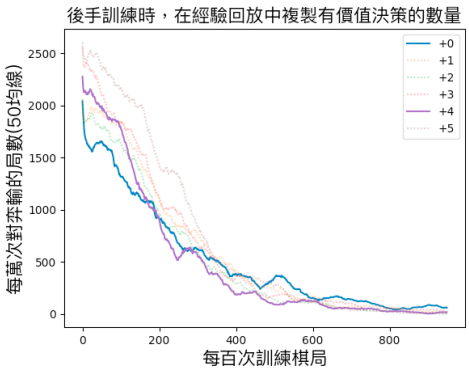

圖 3b. 後手訓練過程中棋力進步狀況

實驗二：雙活路盤面獎勵對 DQN 訓練先手學習效率的影響

我們調整獎勵函數 $R(s')$，額外提供先手雙活路盤面的獎勵值從 0.1 到 1.0，每次增加 0.1。實驗二跟實驗

一的先手採用相同的對照組，在不增加有價值決策複製數的情況下，觀察增加雙活路盤面獎勵值對先手學習效率的影響，結果如表 2 所示。由表 2 可觀察到雙活路盤面獎勵值到 0.5 以上時，開始對訓練先手到無敵能有較明顯幫助，在獎勵值 0.9 時幫助最大，與對照組相比可減少高達 28%的訓練局數。

表 2. 雙活路盤面獎勵讓先手訓練到無敵的平均局數(單位為千局)

獎勵	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
局數	9.7	10.2	8.8	9.4	9.2	8.9	7.9	7.8	8.5	**7.0**	8.5

跟實驗一相同的做法，我們也繪製雙活路盤面獎勵在訓練過程中對先手棋力變化的影響，如圖 4 所示，從中可觀察到雙活路盤面獎勵不論增加多少，相較於無獎勵的對照組，對先手棋力的進步幾乎都有迅速顯著的幫助。對照組的獎勵函數僅在終局提供獎勵，而實驗組則提前在贏得棋局的路徑中先行提供，此舉能引導電腦更有效率地找到贏棋的路徑,避免了耗時浪費的盲目探索。

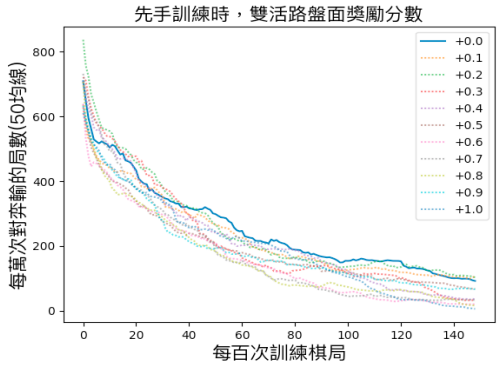

圖 4. 雙活路盤面獎勵在訓練過程中對先手棋力變化的影響

實驗三：增加複製有價值決策到經驗回放中對後手 DQN 學習穩定度的影響

井字遊戲由於後手天生的劣勢，比起先手需要更多的訓練棋局，若訓練局數不足，有時在整個訓練過程都無法出現無敵的結果，以致訓練失敗。本實驗先刻意減少訓練局數以提高失敗率，再觀察增加複製有價值決策到經驗回放後能否改善此問題。由於先手的訓練快速且結果相對穩定，本實驗只針對後手進行探討。我們運用跟實驗一相同的方法，進行十次實驗，檢視在十次中有幾次能成功訓練成無敵，次數越多代表其 DQN 學習越穩定，白費苦工的機率越低。在實驗一後手的訓練局數為十萬局時，不論是否有增加複製數，幾乎每次訓練都能達成無敵的目標，本實驗將後手的訓練局數從十萬局減少至五萬局，結果如表 3 所示。我們可觀察到複製數為 0 的對照組有六次失敗，成功率僅 40%，而增加複製數後，學習穩定度大幅提升，成功率幾乎都增加了一倍。

表 3. 複製有價值決策到經驗回放對 DQN 訓練成功率的影響

增加複製數	0	1	2	3	4	5
訓練成功率(%)	40	80	70	70	80	80

實驗四：真人對奕

在前述實驗一、二、三中，DQN 訓練出的人工智慧的驗證方法都是僅跟隨機亂下子的程式對奕。為了謹慎起見，我們請了十位年齡十五歲以上智力正常的人跟 DQN 訓練出的人工智慧對弈，每人與先手和後手分別對奕兩局，共 40 棋局，先手使用雙活路盤面獎勵 0.9，後手增加有價值決策複製數 4，結果人工智慧先手贏了 6 局，輸 0 局，平手 14 局，後手贏 4 局，輸 0 局，平手 16 局，其中先手幾乎都是利用雙活路盤面獲勝。由此實驗應可確認本研究訓練出的井字遊戲人工智慧已成功達成無敵的目標。

五、結論

本論文以 DQN 訓練出無敵的井字遊戲人工智慧為目標，透過領域知識來調整經驗回放內容和獎勵函數，探討對 DQN 學習效率和穩定度的影響。實驗結果顯示，加入了人為的領域知識介入 DQN 學習，包括增加有價值決策到經驗回放和提前給予特定盤面獎勵，都能大幅地提高 DQN 學習效率和穩定度。最後，我們讓 DQN 訓練出的人工智慧與真人對奕，確認了此法確實能訓練出無敵的井字遊戲人工智慧。

DQN 訓練要成功，常需要非常大量的資料，資料品質的優劣對學習影響甚鉅，且時常會面臨樣本運用效率不佳的問題，因此目前 DQN 的應用範圍多侷限在能容易取得大量資料的用途上，例如電腦遊戲和下棋的人工智慧。透過這次對 DQN 的研究，我們相信加入專家的領域知識介入 DQN 的學習，將是未來的重要研究方向之一，專家的知識不僅能夠提升學習品質，更能大幅降低收集資料所需耗費的金錢和時間成本，讓 DQN 能更廣泛普及地被運用於解決各種不同的問題。

致謝

本論文是作者在賽微科技實習時所進行的計畫。非常感謝楊宗憲博士的耐心指導，和提供了很多傑出想法及建議的徐志文先生，在此也向實驗四幫忙測試的親朋好友們致謝。因為有他們的幫助，計畫才得以順利完成。

參考文獻

[1] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013.

[2] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. Human-level control through deep reinforcement learning. Nature, 518(7540):529–533, 2015.

[3] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. arXiv preprint arXiv:1511.05952, 2015.