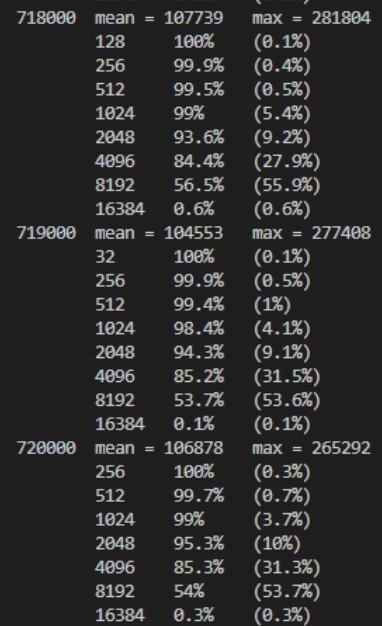
1. A plot shows episode scores of at least 100,000 training episodes.

Graphical user interface, text

Description automatically generated

訓練的時間有點超乎我的預期，原本沒有新增額外N-tuple pattern大致上一天內都可以訓練完成，但不曉得為什麼增加兩個pattern之後我訓練了三天還沒辦法到1M episode，所以只好先把最後幾輪的結果截圖下來。

1. Describe the implementation and the usage of -tuple network.

原始的盤面是一個有16個tile且每個tile大約可能會有17種可能的結果，若我們需要考慮整個盤面，我們會需要大約1716的空間來儲存一個表格，可想而知這是一件不切實際的事情。

所以有些研究表明我們可以像CNN一樣，專注在盤面上的某些特定pattern來當作評估盤面好壞的指標，此pattern可能就代表整個盤面的特徵，至於要用什麼樣的pattern能夠得到更好的效果，可能就需要做實驗以及取決於我們對此遊戲的了解。

Text

Description automatically generated

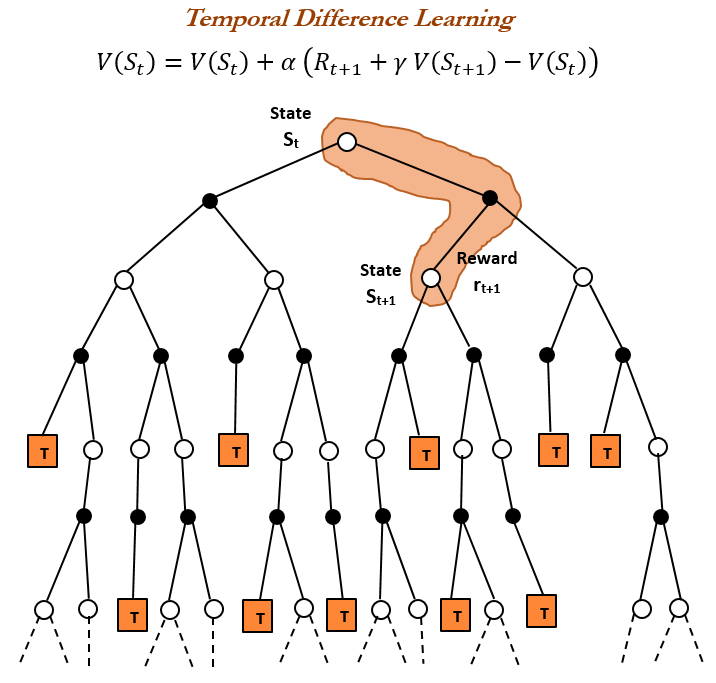
上方的程式碼可以考慮我們所給定的pattern的上下左右等翻轉一共八種不同的變化，並且把八種變化都相加成為此pattern預估的值。

Text

Description automatically generated

透過上方的程式可以取得不同pattern對於當下的盤面所代表的key。

藉由八個isomorphic pattern得到的key所對應的的weight進行加總成為此pattern在當下盤面所代表的估計值。

1. Explain the mechanism of TD(0).
2. 

比起Monte Carlo這種基於整個盤面最終值進行估計的方法，在2048這個遊戲當中我們則是會選擇使用TD Learning的方式進行參數跟新，原因是在2048的遊戲當中缺少了決定性的關鍵一步，這會讓我們最終盤面的預估值非常不穩定，有可能同樣的版面玩到最後會得到2000分，或者是200分這種差距非常大的結果，事實上會出現這種結果的原因是2048不像是圍棋或西洋棋，有佔領地盤的關鍵一步，這一部可能就會對最終結果造成相當大的影響。

所以在2048的遊戲當中我們在更新參數時，不太會相信目前盤面的最終值會是如何，而是會傾向相信下一步所出現盤面的預估值當成我們的更新目標，我們只要根據下一步的預估值就可以跟新這一步的參數。

1. Explain the TD-backup diagram of V(after-state).

Diagram, box and whisker chart

Description automatically generated

Text

Description automatically generated

儘管我們使用TD Learning的方始更新參數，在2048的遊戲當中也有兩種跟新的策略，第一種是After State的跟新方式，在此策略當中我們只會考慮滑動過後的盤面(After state)當成我們的value function去跟新權重，意即，我們假設Before state是已經確定的情況下，所以我們只需要考慮如何選擇下一個動作，根據下一個動作的Reward以及After state的期望值，來更新我們現在這一個盤面的After state的期望值，設法讓我們獲得最大的期望值。

1. Explain the action selection of V(after-state) in a diagram.

Diagram, schematic

Description automatically generated

Text

Description automatically generated

在此策略當中，因為我們會假設Before state是已經確定的，所以我們只需要去算出上下左右哪一個Action的Reward加上After state的期望值會是最大的，選擇此Action當作Best action。

1. Explain the TD-backup diagram of V(state).

Chart, diagram, box and whisker chart

Description automatically generated

Text

Description automatically generated

在第二種Before state的跟新策略當中，我們會使用滑動前的版面(Before state)當成我們的value function去跟新權重，意即，我們需要考慮目前盤面滑動之後，可能會popup出什麼樣的tile，有點類似model free的概念，我們試圖建立一個轉移矩陣，對下一個state可能出現的結果進行加權估計，並把加權過後的估計值進行加總成為更新目前state的依據。

1. Explain the action selection of V(state) in a diagram.

Diagram, schematic

Description automatically generated

Text

Description automatically generated

在此策略當中，有點類似我們想要去多看一步，暴力法地去看每一個action可能會出現的結果，並利用加權平均去計算期望值以當做此action的 分數，換句話說，如果一個遊戲的action很多，以及action後可能出現的結果也很多樣，那麼這一個步驟將會非常耗時，但可以想像的好處就是多看一步的這種估計方式，所估計出來的期望值，顯然會比較值得相信。

1. Describe your implementation in detail.

Text

Description automatically generated

上述程式會取得所有isomorphic pattern在當下盤面所得到的key，並以此key來取得對應的weight，再將全部的weight全部進行加總，得到pattern在當下棋盤所代表的值。

Text

Description automatically generated

上方程式會將pattern應該修正的error平均分配給八個isomorphic pattern，透過index\_of函式去查找各isomorphic pattern在當下盤面所代表的key，並去跟新此key對應的weight，並將更新過後的weight加總後回傳。

Text

Description automatically generated

取得給定isomorphic pattern在當下盤面所代表的key。

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

利用暴力法的方式去檢視四個不同action可能會popup出來的盤面，並且我們已經知道90%的機率會產生2，10%的機率會產生出4，我們就可以透過機率的方式去計算出popup出2或4將會獲得多少的期望值，並將全部可能的期望值進行加總再加上原本的reward當成此action所獲得的值，我們只需要選擇最大的那一個action即可。

A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

這邊基本上就是透過上述的演算法去跟新不同盤面的Before state所對應的期望值，比較不一樣的方式則是，這邊會在每一輪episode結束時，透過記錄下來的move從最後一筆去往前跟新。

1. Other discussions or improvements.
2. 新增不同的N-tuple pattern

Text

Description automatically generated

除了原本有的四個pattern以外，額外新增兩種pattern，每一場達到2048的比例從90%提升到95%。

1. 選擇action時增加Epsilon greedy

因為我們如果只靠選擇最大的期望值讓當作下一步的action，很可能不是一個最佳解，可以試想一種情況，如果一開始我們選擇期望值最大的action，但卻會讓我們接下來所獲得的reward非常小，當這種情況發生時我們的策略可能並不會是一個最好的結果，但如果我們在訓練一開始時，先讓agent隨機去決定要往哪一步走，雖然一開始可能會獲得比較小的reward，但可能之後獲得的reward都是非常大的。

不過當我實作完Epsilon greedy之後我發現一件事，也許2048這個遊戲並不太適合Epsilon greedy這種選擇action的策略，如果今天是不同的遊戲，有可能agent可以玩出超乎我們人類預期可以做到的結果，但今天在這種棋盤類遊戲之下，我們非常仰賴一開始盤面的狀況，如果一開始是隨機去選擇我們的action，有可能盤面會太糟，導致後面救不回來的窘境。