





DSP HW1 Report

R09944072 鍾宜樺

➤ 如何編譯

 make // compile 得 train 和 test 的 exe 檔
 make ITER=100 run // 運行作業一
 make acc // 算得 accuracy
 make clean // 清除 exe 檔

➤ 實驗結果

@ iteration = 100 (accuracy: 82.8%)

```
(base) chungyihua@chungyihuadeMBP dsp-hw1_r09944072 % cat acc.txt  
0.828868
```

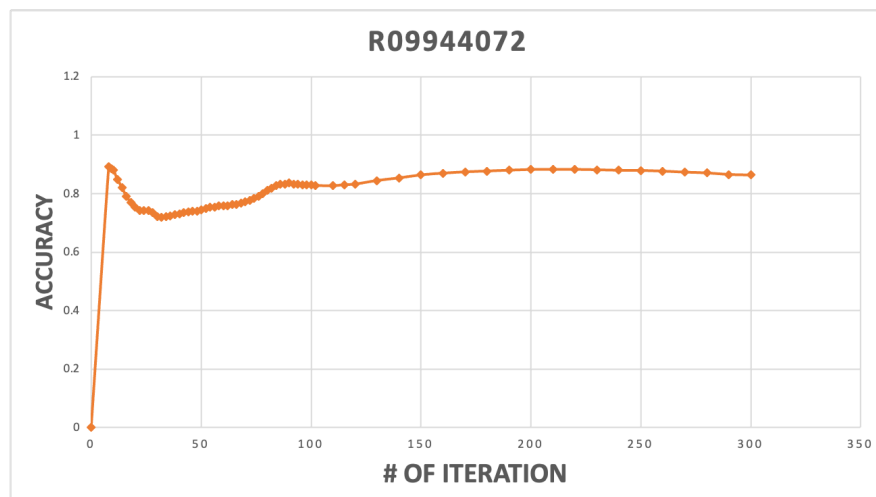
➤ 作業中遇到的困難：

1. train.c 中 gamma, epsilon 的累加，不清楚如何累積、累積運行在程式的哪較好。因為有時限，怕會寫到重複的計算。最後參考 Daniel Jurafsky 的 Speech and Language Processing 書中的 E step, M step 把這部分寫完。
2. for loop 的 index 容易寫錯。作業一裡面有很多需要做累加的參數；然而，每個參數都有自己的上下界，不注意就會寫錯 index。
3. 寫好 test.c 後，第一次跑整個作業一時正確率只有 0.01% 的當下，不確定問題是在 train.c 還是在 test.c。因為 train.c 寫完後不曉得如何驗證正確率，只有在 test.c 寫好後看正確率。最後發現是 test.c 讀檔參數寫錯導致正確率之低，看到答案輸出 82.8% 的時候，感覺是浮誇地涕泗縱橫。

➤ Test Accuracy vs # of Training Iterations

前 100 iteration 中，每間隔 2 iteration 跑一次正確率(0, 2, 4, ..., 100)。

100 - 130 次 iteration 因接近收斂了，則每間隔 10 iteration 跑一次結果：



➤ 實驗結果討論：

首先把做出來的實驗結果圖和助教的實驗結果圖相比較：

1. 一開始跑實驗時，先從 $\text{iteration} = 1, 2, \dots, 10$ 開始跑，結果 iteration 很小的時候的正確率高到讓人驚訝。雖說助教的圖一開始實驗準確率也很高，接著慢慢降低隨後再升高，但也不至於一開始的實驗就和收斂之後的準確率差不多。但把整張圖畫出來後覺得，或許結果的分佈會因為實作方式不同而有所差異。

(1) 舉例來說：在累積 gamma , epsilon 的地方，我可以想到的作法有很多種。像是可以在 M step 一次累積（我也是採用這種方法）、也可以會在更新 gamma , epsilon 的時候就另外開一個矩陣一邊更新就一邊做累積。那以 ALU operator 的立場去看在每一次乘法器作用的時候都差一點尾數，小差異慢慢累積到最後，就變成大差異了，這或許就是造成演算法一樣，但實驗結果不一樣的一個原因。

(2) 另外還可以思考到的一個原因是，在算 α_{t+1} 的時候的計算方式：

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] \cdot b_j(o_{t+1}),$$

可以把 $b_j(o_{t+1})$ 在計算前面的 $\alpha_t(i) a_{ij}$ 的時候就把 $b_j(o_{t+1})$ 一起乘進去。但在實作的時候考慮到有限時間，因此會先計算完

$\sum_{i=1}^N \alpha_t(i) a_{ij}$ 再將總和乘上 $b_j(o_{t+1})$ ，相比於前面會從 N 個乘法變成 1 次乘法，希望可以藉由少做好時的乘法得以節省時間。

2. 接著說一下，為何在 $\text{iteration}=1, 2, 4$ 的時候會高達 90% 左右，我覺得是因為這個演算法本來就和 model initialization 非常有關係，在一開始初始化很好的狀態下，就有機率跑到 90% 的準確率，這也是我想到的在 iteration 數量很小時準確率卻很高的原因。
3. 最後一個有趣的小錯誤，在跑最後一張圖時，理論上要寫一個 script 讓所有 iteration 數目都跑一次，紀錄數據再畫圖。但因為我在一開始 iteration 數量小的時候看到 accuracy 高到讓人驚訝，我想要每筆數據都邊跑邊看，所以用手動一邊跑一邊看，但手動總是麻煩的，最後想要就一次跑兩個，accuracy 就突然變得很低 ($\text{iteration} = 300$, $\text{accuracy} = 60.2\%$) 嚇得我又跑重新一次，就正常回到 88% 上下，這時候才想起來，一次跑兩個他們改的是同一份檔案會有 race condition 啊，最後把資料夾複製 8 個，分 8 個不同的第方跑就好了。