

一、實驗結果

這次的作業主要讓我們練習語音辨識數字，雖然助教已經幫我完成大部分的 Task，不過我因為有修過李琳山老師的專題研究，所以對 Kaldi 還算熟悉。

iter	maxiterinc	numgauss	totalgauss	acc	acw
5	4	1	5	75.34%	0.87
10	4	1	5	76.17%	
20	4	10	100	86.18%	
20	10	300	1000	91.08%	
20	10	500	1000	90.56%	
20	20	300	10000	91.31%	
20	20	300	6000	92.69%	
20	20	300	6000	94.93%	0.5
20	20	300	6000	95.91%	0.1
20	20	300	6000	96.32%	0.15
20	20	300	6000	96.55%	0.25
20	20	300	6000	96.55%	0.25(testbeam=13)
20	20	300	6000	96.43%	0.25(testbeam=20)
20	20	300	6000	95.51%	0.25(testbeam=8)

一開始 baseline model 跑完大概 acc 落在 75.34%，直接說結論的話，iter 大概要在 15~25 次，起始的 gauss 在 80~300 個，total gauss 在 6000 左右、acwt 在 0.15~0.25，效果會最好。

```
Generating results for test set with acoustic weight = [ 0.15 ]  
output -> viterbi/mono/test.mlf  
log -> viterbi/mono/log/latgen.test.log  
result -> viterbi/mono/test.rec  
accuracy -> [ 96.78 ] %
```

另外我有嘗試去調過 topo，包含增減一些 sli 或是 nonsli 的 state，結果發現大部分的情況下都會導致 acc 降低，我想原因可能是因為單字的複雜度本身就已經不高，甚至做了 10 個 nonsli 的 state 結果導致 acc 大爆炸的情況。

```
accuracy -> [ 35.46 ] %
```

二、實驗遇到的困難

其實對我來說因為都很熟悉這次作業，docker 之前雖然沒用過但稍微有研究過，調參方面比較有問題的應該是 realign 的調法，然後希望下次實作部分可以更有挑戰性一點。