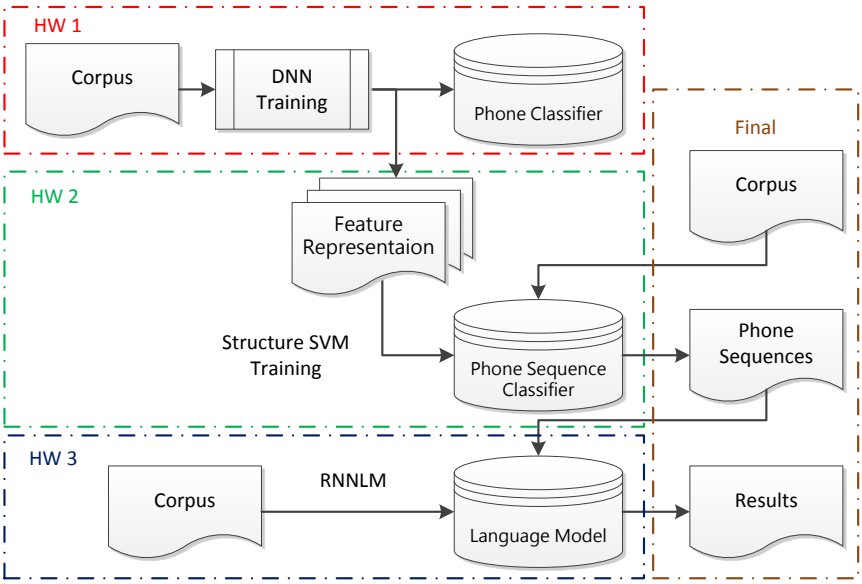
MLDS Final

Team: 深度學習的憂鬱
R03922152 王浩丞、R03922151 賴彥麟、D03944011 賴威昇、D02922002 顏明祺

1 流程圖



2 演算法

2.1 DNN

i. 延伸 HW1 的 DNN 架構，我們增加了 momentum、ReLU、weight decay。

ii. 另外我們也嘗試了 RMSProp 和 dropout，但效果並不佳，推測 dropout 應該要先做 pre-training，於是我們嘗試了 theano 的 RBM 和 auto-encoder，但遇到一些 memory issue 因此並沒有完整的實驗結果。

2.2 N-best Viterbi

i. 由於每個語者發音方式會有所差異，導致計算出來的 phone sequence 可能難以轉換為正確的 words，所以在這個步驟中需要輸出更多的可能，並利用後續語言模型重新排名才能得到較佳的結果。

ii. 這個部份我們使用 heap 來 implement，相比原本的 viterbi 演算法，時間複雜度約會增加$n \times log(label_num)$ 倍。

2.3 Language Model

i. **RNNLM**

1. 使用HW3的小說為training，Final的1734句(原本3000句，去除重複)為validation。處理方式皆為移除 stopwords，將 timit 字典中沒有的字替換成<unk>，再做 stemming。
2. 由於移除 stopwords 會使文法結構改變，可能會造成 noise，所以建立一個沒有移除 stopwords 的語言模型。
3. 由於 HW3 小說的年代、domain 皆與 timit 不同，所以建立一個只使用 Final 1734 句的以其中 1534 句為 training，200 句為 validation。其他設定同(2)。

ii. **Bigram**

1. 由 RNNLM 結果可以知道 Final corpus 效果大於 HW3，但是 Final 資料太少，建立 Bigram 很容易有遺漏，因此需參考 HW3。為此，建立$Bigram_{Final}$和$Bigram_{HW3}$，再用 weight 組合起來：$\alpha \times Bigram_{Final} + (1-\alpha)Bigram_{HW3}$。

3 實驗結果

3.1 HW1 實驗數據

我們重新 train 許多不同的 DNN feature，使用舊的 testing data 上傳到 HW1 kaggle 比較結果。fbankN 表示合併前後 N 個 frame 的 fbank feature(共 2N+1 個 frame)，feature 皆有先做過 standard normalization，learning rate=0.01，momentum=0.9，weight decay=0.0005，batch size=256，epoch=1000，activation=ReLU，output label 使用 48-phone。我們將 training set 切成 1000000 筆的 training data 和 124823 筆的 validation data。做完 prediction 之後，我們也將 output label 做 smoothing，將每個 label 附近五個 frame 出現最多次的 label

表 1. 不同 feature 的比較

feature	hidden layer	validation accuracy	testing accuracy	testing accuracy (smoothing)
fbank	2048 x 2048	0.663321	0.68451	0.69784
fbank1	2048 x 2048	0.730985	0.72204	0.72979
fbank2	2048 x 2048	0.754619	0.73163	0.73831
fbank3	2048 x 2048	0.772164	0.73312	0.74024
fbank4	2048 x 2048	0.784757	0.73414	0.74133
fbank5	2048 x 2048	0.796855	0.73866	0.74529
fbank7	2048 x 2048	0.818205	0.74269	0.75006
fbank8	2048 x 2048	0.819463	0.73881	0.74621
fbank10	2048 x 2048	0.825007	0.73617	0.74238

由上表可以發現，合併越多 frame 通常可以得到更好的結果(fbank8 和 fbank10 可能需要更大的 network)，另外 smoothing 可以提升約 0.006 到 0.01 左右的正確率。從我們的實驗數據可知，通常還是兩層的架構可以得到比較好的結果，參數數量較多的 network 反而會 overfit 導致 testing accuracy 下降。在這個部分我們最好的結果是 0.75146 (fbank7，2048 x 2048，使用全部的 training data + smoothing)，比起之前 HW1 我們這組最好的結果(0.71086)，我們的 DNN 提升了大約 4%左右的正確率。

表 2. 不同 network 大小之比較:

feature	hidden layer	# parameter	validation accuracy	testing accuracy
fbank1	2048 x 2048	4.4M	0.730985	0.72204
fbank1	3072 x 3072	9.7M	0.736232	0.73029
fbank1	1024 x 1024 x 1024	2.2M	0.750052	0.71080
fbank1	1024 x 512 x 256 x 128	0.7M	0.746663	0.68163
fbank2	2048 x 2048	4.7M	0.754619	0.73163
fbank2	3072 x 3072	10.2M	0.761629	0.73029
fbank3	2048 x 2048	5.2M	0.772164	0.73312
fbank3	2048 x 2048 x 2048	9.5M	0.81182	0.72422
fbank3	512 x 512 x 512 x 512	1.0M	0.773413	0.70872
fbank4	2048 x 2048	5.5M	0.784757	0.73414
fbank4	2048 x 2048 x 2048	9.7M	0.814255	0.72302
fbank4	1024 x 1024	1.7M	0.777219	0.73942
fbank4	1024 x 1024 x 1024	2.8M	0.809617	0.72838
fbank4	1024 x 1024 x 1024 x 1024	3.8M	0.820256	0.70985
fbank7	2048 x 2048	6.4M	0.818205	0.74269
fbank7	2048 x 2048 x 2048	10.6M	0.835614	0.72829

3.2 **HW2 實驗數據**

我們從 HW1 結果較好的幾個 network 取出最後一層 softmax 之前的 48 維 output 當作 feature，經過 standard normalization 並加上一維的 bias 之後拿給 svm-struct (c=1000)做 training，用舊的 testing data 上傳到 HW2 的 kaggle 做比較。另外我們也比較了直接把 HW1 output label 做 trimming 的結果，以及使用 Final 的 testing data、1-best Viterbi + 1-WFST(沒使用 language model)的結果。由實驗結果可以觀察到: HW1 正確率較高的 feature 在 HW2 的表現不一定較好。由於 svm-struct 已經有考慮 sequence 的前後關係，因此只 merge 前後 1~3 個 frame 的 feature 反而表現比較好。

表 3. 不同方法的綜合比較

feature	hidden layer	HW1 testing accuracy	HW1 (trimming) edit distance	HW2 (svm-struct) edit distance	Final (w/o LM) edit distance
fbank	2048 x 2048	0.68451	39.45946	9.28716	7.08037
fbank1	2048 x 2048	0.72204	28.32432	8.94595	7.05794
fbank2	2048 x 2048	0.73163	27.83446	8.89527	7.10093
fbank3	2048 x 2048	0.73312	27.26351	9.0777	7.06355
fbank4	2048 x 2048	0.73414	27.46284	9.46284	7.24486
fbank5	2048 x 2048	0.73866	27.43919	9.625	7.41199
fbank7	2048 x 2048	0.74269	27.0473	10.13176	7.35581
fbank8	2048 x 2048	0.73881	27.69595	10.05068	7.50187
fbank10	2048 x 2048	0.73617	27.83784	10.00676	7.5206

將 HW2 實驗數據經過 WFST 取出 1000 筆後再加上 Language Model 後，我們發現表現最好的是 fbank3 這組，其在不同 Language Model 的結果比較如表 4 和表 5。可以看出 HW3(Holmes)和 Final(1734 句)兩個 corpus 在不同語言模型中有著截然不同的表現。

表 4. RNNLM based Language Model

	HW3 data w/o stopwords	HW3 data w/ stopwords	Final data w/ stopwords
fbank3	6.76075	6.57196	6.5514

表 5. Bigram based Language Model with different α values

α	0	0.25	0.5	0.75	1
fbank3	**6.46442**	6.47940	6.48315	6.49232	6.50187

4 結論

在本次期末中我們實作了以 fbank 音訊特徵為輸入的語音辨識系統，我們利用 dnn 中間層的輸出作為新的特徵加入 structural svm 來取得可能的 phone sequence，再利用 OpenFST 與 RNNLM 來找出最有可能的語句。我們的系統在給定的 TIMIT dataset 上能得到 **6.46442** 的分數，然而我們的系統與實際的語音辨識系統(google baseline)仍有相當大的差距，距離實用還有很多精進的空間。

5 分工

DNN：賴威昇
N-best：賴彥麟
RNNLM：王浩丞、賴彥麟
Bigram：王浩丞
Kaldi(未使用方法)：顏明祺
Report：王浩丞、賴彥麟、賴威昇、顏明祺