**Machine Learning and Having it Deep and Structured**

**Final Project Report**

Group Name: **hello world**

Members: 李冠穎 R03922165

張人尹 R03944007

肖可依 R03921089

陳柏豪 B99203027

**一. 流程圖**

DNN(Caffe) -> HMM -> WFST -> RNNLM

DNN(Caffe) -> Structured SVM -> WFST -> RNNLM



**二. 資料前處理、資料結構，演算法，實作**

**DNN**

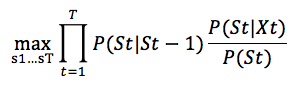
我們使用 MFCC(39維) feature ，將每個 frame 前後 7 個 frame 接起來，產生一個 39 \* 15 = 585 維的 feature，以保留 context ，也就是前後資訊，作為 DNN 的 training data 。 在使用 caffe 訓練 DNN 的過程中，我們將 DNN 的結構設定為 5 層維度為 2000 的fully-connected layer。

值得注意的是，在 7 \* 3000 - 250 - 48 的DNN結構中，我們特別取出倒數第二層的 feature 作為 structured svm 的 training data。

我們訓練了多個 DNN 模型，對每一個 phoneme ，將每個模型輸出的 label 取眾數，作為 ensemble 的結果。結果做 smooth 后上傳 HW1 ，最好可以達到 0.7759 的準確率。

**DNN-HMM Hybrid**

我們透過統計的方法計算 48 個 phoneme 各自出現的機率以及彼此間的轉換機率，並透過如下的方程式( P(St|Xt) 為 DNN 得到，Observation Prob. 及 Transition Prob. 利用 HMM 得到)，利用 Viterbi 得到 N-best 的 phoneme sequence，這也是我們最佳的結果。



**Structured SVM**

我們使用 DNN 倒數第二層 layer 的輸出結果( 200 維 feature )，並標記對應的label送進structured svm 訓練。

與 HW2 不同的是，在做 predict 時，我們選取最好的 n-best path 作為 WFST 的 input data 。預測結果。當 N 大於 48 時，每一步挑選 48-best ，只在最後一步 48 \* 48 中挑選 n-best。

**RNNLM**

我們將 3696 句 training data 送進 RNNLM Toolkit training，BPTT 5層。架構是 hidden layer 50 維，50 個 class。

**三. 實作技巧**

Acceleration:

Python 版本的 Viterbi 實作用矩陣運算平行化加速

Ensemble:

將 weight 取平均發現準確率大幅降低，因此各個 model 分別過 DNN-HMM 之後，對得到的 phoneme sequences 每個 phoneme 分別做 voting，發現比單一個 model 效果還好。

C語言中獲取n-best的方法:

計算過程中為了保持 N 個最佳結果，我們用函式 find\_min\_value 來找出目前所紀錄值中最小的，並與新算出的值比較，若新算出的值較大則將之替代掉原本 N 個紀錄中最小的值。

**四. 遇到的困難和解決方法**

在把 structured svm 的 predict 部分改成 n-best viterbi 時，我們開始是在每一個 phoneme 到下一個 phoneme 時就記錄 n-best path ，然而我們發現從第一個 phoneme 到第二個 phoneme 時，最多只能選出 48-best ，而到最后一個 phoneme 我們一共可以得到 48 x n 條 path ，在其中選 n 個最優解。因此當遇到 n > 48 的狀況，我們在 viterbi 的每一步，只記錄 48-best ，然後在最後 48 x 48 = 2305 條 path 中，選取 n-best ，由於我們的實驗 n 取值最大到 1000 ，在這種狀況下，該方法可行。

在使用 command 使 RNNLM 幫句子的 rescore 的 script 裡，中繼檔案因為開啟檔案時使用 'w' 造成每次覆寫，只選到最後一句，這個小 bug 改掉之後 edit distance 從 7.2x 上升到了 6.7x 。

**五. 實驗設定與結果**

我們試了不同的 DNN models (5層 hidden layers，每層 2000 個 units ，output layer 48 個 class 。在 HW1 的 test set 上，1個 model 通常 accuracy 74% 左右，多個 models ensemble 起來可以到 77% 左右)， RNN models (3696: hidden units = 50，class size = 50，BPTT = 5，epoch = 11; default in toolkit: hidden units = 200，class size = 100，BPTT = 5，epoch = 15)，也調了很多不同 n-best 參數和一些設定後的結果：

HMM viterbi N-best = 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DNN model | WFST N-best | RNNLM model | Smooth | Edit distance |
| 1 | 500 | trained by 3696 | O | 7.44673 |
| 1\* | 1000 | trained by 3696 | X | 6.88411 |

HMM viterbi N-best = 100

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DNN model | WFST N-best | RNNLM model | Smooth | Edit distance |
| 1 | 1000 | trained by 3696 | O | 6.82804 |
| 5 ensemble | 1000 | trained by 3696 | O | 6.77383 |
| 5 ensemble | 1000 | trained by 3696 | X | 6.66729 |
| 5 ensemble | 20 | trained by 3696 | X | 6.73645 |
| 5 ensemble | 250 | trained by 3696 | X | 6.68037 |
| 5 ensemble | 550 | trained by 3696 | X | 6.62804 |
| 5 ensemble | 500 | trained by 3696 | X | 6.63925 |
| 5 ensemble | 500 | default in toolkit | X | 6.68224 |
| 5 ensemble | 500 | n-gram | X | 7.42991 |
| 5 ensemble | 500 | trained by 3696 with english dictionary | X | 6.65421 |
| 5 ensemble | 500 | trained by 3696(BPTT 7) | X | 6.66355 |
| 5 ensemble | 500 | trained by 3696(epoch 15) | X | 6.79813 |
| 5 ensemble | 500 | trained by 3696(epoch 20) | X | 6.79065 |
| 5 ensemble\*(30 epoch) | 500 | trained by 3696 | X | 6.80935 |
| 7 ensemble | 500 | trained by 3696 | X | 6.65981 |

HMM viterbi N-best = 1000

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| DNN model | WFST N-best | RNNLM model | Smooth | Edit distance |
| 1 | 1000 | trained by 3696 | X | 6.70280 |
| 5 ensemble | 1000 | trained by 3696 | X | 6.69720 |
| 5 ensemble\* | 1000 | trained by 3696 | O | 6.82056 |
| 5 ensemble\* | 500 | trained by 3696 | X | 6.67477 |
| 7 ensemble | 500 | trained by 3696 | X | 6.67477 |
| 7 ensemble | 500 | trained by 3696 | O | 6.80561 |

(\*為快接近 deadline 時，重 train 了一個 7 層，每層都 3000 units 的 DNN )

經過測試，我們發現對 phone sequence 做 smoothing ，也就是若沒有超過 3 個連續的 phones 就會被我們去除，這會讓 edit distance 變高，所以後面的實驗大多不做。然後 rescore 時如果將句子中發現有不在 Lexicon 或不在英文字典的字，給句子一個小或大的 penalty 的話， performance 也沒有變好。

而跑 HMM viterbi 的 n-best 是會讓後面 components 選擇變多， performance 會提升。而不管是用哪種 LM ， performance 差距不大且是 trained by 3696 的最好。但最奇怪的是，不是 viterbi n-best 調越高越好，也不是 WFST 給的 N-best 調越高越好，像我們最好的結果就是 WFST-500 比 WFST-1000 還要好。

另外，我們嘗試了structured svm，但由於一開始抽取 DNN 倒數第二層的 feature 維數太多（2000維）導致 svm 沒能 train 完。最後我們重抽了 200 維的 feature，但由於時間有限，只 train 完了c=1 的 svm ,可能也是這原因導致 Structred SVM 效果比 HMM 來的差 (雖然我們認為 HMM at last 還是會贏過 structred SVM)。

**六. 貢獻**

李冠穎：Data-preprocessing，DNN(caffe)，DNN-HMM hybrid system

張人尹：Data-preprocessing，WFST，LM(RNN, n-gram)，Experiments

肖可依：svm n-best, part of report

陳柏豪：svm n-best, part of report