MLDS HW1 report

r04943151 梁可擎

使用模型

兩種使用的模型結構：

RNN

基本 RNN 模型：

```
RNN(units=128, dropout=0.2, recurrent_dropout=0.2)
Dense(units=nb_classes, activation=softmax)
```

這次作業使用的是一層 LSTM，結構大致如下：

```
LSTM(units=128, dropout=0.2, recurrent_dropout=0.2)
Dense(units=nb_classes, activation=softmax)
```

RNN+CNN

CNN + LSTM 的結構如下：

```
Conv1D(units=128, kernel_size=5, activation=relu)
LSTM(units=128, dropout=0.2, recurrent_dropout=0.2)
Dense(units=nb_classes, activation=softmax)
```

How to improve performance

在 data pre-processing 方面，我是以一個 instance 當作一 time step，因為 training data 最長是 777 個 frames，所以把所有 instance 都補 0 到 777。另外有對每個 feature 做 normalize，減去平均值再除以標準差。

最一開始是使用 15 個 frame 當作一個 time step，這樣的缺點除了會訓練非常慢以外，沒有看到每一個 instance 的所有 frame，loss 下降也非常慢。

另外在 model 調整方面，因為助教說從簡單的開始嘗試，就先試驗了一層 lstm，optimizer 選擇 rmsprop 和 adam。實驗結果會在後面敘述。除了 optimizer，一開始選擇 batch size 太大，例如 256 也會讓 loss 下降有限，經過實驗後發現 batch size = 32 或 64 是比較理想的狀況。

比較有趣的是選擇 padding label 的部分，padding 的 frame 要選擇 label 成 'sil' 還是新增一個 dummy label，兩個實驗也會在後面敘述。

Experimental results and settings

RNN

```
(3696, 777)
after slice:  (3696, 777, 69) (3696, 777, 39)
Building model...
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_1 (LSTM)                (None, 777, 128)          101376
_________________________________________________________________
dense_1 (Dense)              (None, 777, 39)           5031
=================================================================
Total params: 106,407
Trainable params: 106,407
Non-trainable params: 0
```

使用 rmsprop 訓練 50 epochs 的結果

```
3326/3326 [==============================] - 28s - loss: 0.5260 - val_loss: 0.4943
Epoch 49/50
3326/3326 [==============================] - 28s - loss: 0.5265 - val_loss: 0.5614
Epoch 50/50
3326/3326 [==============================] - 28s - loss: 0.5308 - val_loss: 0.4646
Test loss: 0.464022447795
```

使用 adam 訓練 300 epochs 的結果

```
Epoch 299/300
3326/3326 [==============================] - 64s - loss: 0.4728 - val_loss: 0.4335
Epoch 300/300
3326/3326 [==============================] - 64s - loss: 0.4723 - val_loss: 0.4355
Test loss: 0.427273320945
```

使用 rmsprop 訓練 300 epochs 的結果

```
Epoch 299/300
3326/3326 [==============================] - 67s - loss: 0.4634 - val_loss: 0.4098
Epoch 300/300
3326/3326 [==============================] - 67s - loss: 0.4624 - val_loss: 0.4139
Test loss: 0.401135411355
```

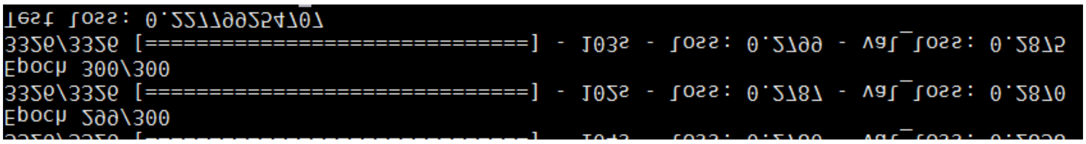
可以看出在同樣參數下，rmsprop 的效果優於 adam
另外 40 labels 會優於 39 labels，判斷可能的原因是 ‘sil’ 本身數量較多，全部 padding 成 ‘sil’ 會讓 model 誤認太多東西是 ‘sil’ 因此效果較差。

CNN+RNN best model

在 50 epochs 的情況下 loss 下降就比 RNN 快，validation loss 也較好。

optimizer=rmsprop batch_size=32 nb_classes=40

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 777, 128)          44288
_________________________________________________________________
lstm_1 (LSTM)                (None, 777, 128)          131584
_________________________________________________________________
dense_1 (Dense)              (None, 777, 40)           5160
=================================================================
Total params: 181,032
Trainable params: 181,032
Non-trainable params: 0
_________________________________________________________________
Train on 3326 samples, validate on 370 samples
Epoch 1/300
```



訓練 300 epochs 後得到 edit distance 是 10.52542，為目前最好的結果。

另外其他模型如 GRU 等等也在實驗中，GRU 的下降速度更快，但比較容易 overfitting，目前為止還沒做出比 CNN 好的結果。