

ADL HW3 Report

電信二
r04942056
余朗祺

一、實作細節

我採用雙層 LSTM 網路來實作這次的兩種 tasks, 細節可以參考 rnn.py 和 rnn.int.py 中的 function *dynamic_rnn*. Cell 使用 `tf.nn.rnn_cell.LSTMCell`。為了 batch 化訓練過程, 我設定最大輸入字串長度為 28, 這是 testing data 句子的最長長度。如果 training/testing 句子超過這長度, 就直接截短, 不足的話則補零向量。此外, 我也用了 Dropout。為了做 validation, 我將 training data 分成 4,500 句的 training set 與 478 句的 development set. 每過一個 epoch, 觀察 dev set 的平均 cost 是否下降, 若有則儲存新的 model, 若無則將 learning rate 乘上 0.9。Optimizer 都是使用 Gradient Descent.

兩種 task 都總共跑了 100,000 步, batch size 是 8 句, 故總共約 177 個 epoch。Hidden layers 都是 128 維。初始 learning rate 都是 0.1。Slot filling 的 dropout rate 採用 20%, intent prediction 則是 40%。

不同的 Dropout rate/hidden layer dimension 會導致不同的結果。Slot filling 部分, 若使用 200 維 hidden size, dropout 10% 的 F-score 是 0.829, 40% 是 0.878, 50% 是 0.851; 使用 128 維 hidden size 與 20% 的 dropout, F-score 是 0.915。因為時間不足, 其他種參數組合與 intent prediction 的部分並沒有做出來。

二、程式架構

我的程式分成 slot filling (rnn.py) 與 intent prediction (rnn.int.py) 兩個任務, 兩個檔案架構大同小異, 主要差別在於訓練目標的不同。其大致架構如下:

(1) 讀入句子與對應的標籤。(rnn.py line 158~172; rnn.int.py line 152~165)

將訓練資料與自定義的 development set 讀入, 並決定 output label 的對應 id, 寫成 class_f 檔, 以便 evaluation 時參考。

Word embeddings 的資訊也在這一步讀入。我使用 Hw1 中 Word2Vec 訓練的 word embeddings, 維度為 200。

(2) 建構 LSTM 網路 (rnn.py line 174~195 ; rnn.int.py line 167~182)

細節請參考一、實作細節。

(3) Training/Testing

細節請參考一、實作細節。為方便初期調整參數，每過 0.1 epoch 會印出目前這個 epoch 的平均 cost, 以便及早停止訓練表現不佳的參數組合。

因為有用到 dropout, 在 testing 時必須將其設為 0 (keep_prob: 1.0)，以得到最佳的表現。