ADL HW2 Report

電信二
r04942056
余朗祺

我的程式分成下列部分：

(1) 讀入句子與對應的樹。(rvnn.py line 201~212)

樹的部分我使用 python 的 pyparsing 套件，這個套件中的 function nestedExpr 可以用來 parse 廣義的以括號標註的樹(如我們的 tree file)，把它轉成 list of lists (of lists…)的形式，以方便我們處理。
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) 初始化所有 parameters (rvnn.py line 213)

Parameters 包含所有 POS tag (NN, DT…等)對應的轉換矩陣(W)與 bias (b)。若有 K 種 POS，其中 L 種具有兩個或以上的 arguments，我們就會有 2L+(K-L) = K+L 種(W, b)。這些資訊必須從 training data 中計算。

(3) Training/Testing

每次 forward/backward 時必須先建立對應 tree 的 RvNN 網路。首先 expand_and_compile 以 DFS 的順序 traverse 這些樹，並將經過的順序記下來。build_rnn_graph 再利用這些資訊將對應的(W, b)串接起來形成 RvNN 網路。若有 POS tag 有超過三個參數，我會先結合前兩者，再將其 output 結合第三者，以此類推，故每個 POS tag 至多只需要有兩組(W, b)。

為增進學習效果，在每個 epoch 開始前會先將 training data 順序洗牌(line 220)，並在 epoch 結束前將 learning rate 降低為原本的 80% (line 263)。

Testing (line 242~262) 基本上同 Training，只是我們不會給 RvNN 網路關於 Optimizer 與 Label 的資訊。