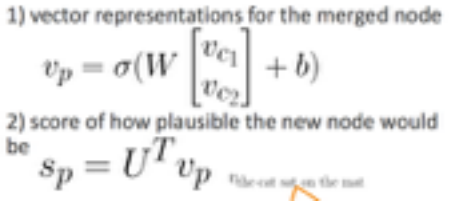
Assignment 2 Report

B02902034 資工四 邱筱晴

I. Code Structure

1. Data preprocess：將raw data以句字為單位讀進來，轉換成nltk tree，使用nltk function collapse unary (只有一個小孩的node) 及 binarize tree (多於兩個小孩的node)，再將pos tag的部分parse出來換成positive/negative label，輸出成tran_tree_[type].txt 檔案
2. tree.py：將preprocess過的data讀進來，建成tree
3. utils.py：統計training data裡的句子出現過的字詞，建成Vocab。
4. approach 1 — glove：讀進pretrain好的glove word embedding，根據3.建好的vocab，根據vocab index依序在embedding matrix裡放入glove embedding，若word不在glove的model裡則初始化為uniform。使用embedding matrix 作為embeddings的initializer。
5. approach 2 — init W matrix：將W使用numpy array初始化為對角線為0.5的陣列，意即將左右兩個小孩各取權重一半相加。
6. rvnn.py：
   A. __init__：實作4.，並根據公式宣告所需的variables、placeholders（如下圖）
   B. train(), run_epoch()：
      with tf.Session() as sess:
      sess.run([self.root_loss, self.train_op], feed_dict=feed_dict)
      在這裡採用root的label計算loss，以及Adam optimizer做優化，其中feed_dict的參數根據右下圖分別有vocab, node_words, is lead, left/right child，可以traverse tree得到。
   C. make_conf所輸出的matrix，對角線上為預測正確的data數量。依序印出training data及validation data的make_conf matrix作為參考。




```
vocab = {'the': 0, 'old': 1, 'cat': 2}
node_words = ['the', 'old', 'cat', '', '']
is_leaf = [True, True, True, False, False]
left_children = [-1, -1, -1, 1, 0]   # indices of left children nodes in this list
right_children = [-1, -1, -1, 2, 3]  # indices of right children nodes in this list

node_word_indices = [vocab[word] if word else -1 for word in node_words]
```

7. Testing: Load train好的model，使用predict()，得到prediction並輸出檔案。

II. How to execute the code

1. The model size is too large to put onto Github, so please download it at https://drive.google.com/file/d/0B6z4WeJetpHgeTBaTGF1cmkzaDg/view?usp=sharing
2. Name the folder “weights” and put it in to the same directory with rvnn.py
3. Run the script

III. Difficulties/Observation

1. approach 1：一開始只使用glove embedding時，用 300維 learning rate=0.01時，loss極大，不可train QQ。因此降低learning rate 為 0.001，train 10 epoch 時，training data的accuracy可提高至0.8~0.9，但validation accuracy 依舊維持在0.5~0.55，無法超過0.6，懷疑 overfitting(但應該沒這麼容易?)。整體來說最好的accuracy落在embed_size=300, epoch=15, lr=0.001為0.51~0.55，但performance仍不夠超越baseline 0.55。
2. 因此採用approach 2，初始化W陣列。此方法很有用！embedding 與learning rate 仍維持在300與0.001，3個epoch就可以達到0.578的正確率，相較之下快速許多。

IV. Conclusion

我覺得使用pretrain的embedding效益不大，可能原因是data size較小，無法將embedding train完整。初始化W反而出奇的有效，因為random給uniform可能要訓練較久才能達到這樣的效果，然而我們已知rvnn為二元樹結構，因此可以用初始化matrix來優化結果與效率。