MLDS 2018 Assignment 2-2

- Model description (2%)
  - Describe your seq2seq model

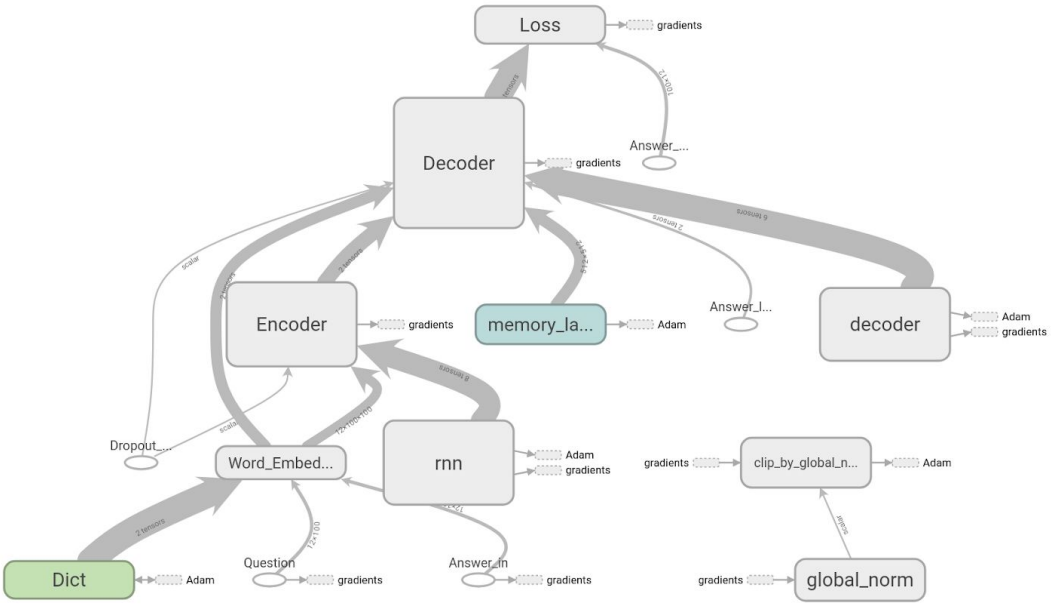
Word Embedding

vocabulary 的生成方式是直接用 traing data 送進 gensim 做 word embedding 的訓練，最小出現次數要3才算。而oov的單字最後會直接使用<UNK>的embedding，最後預測出現<UNK>的話則直接跳過。

Data Preprocessing

Data的處理方式為固定長度為12個單詞，多的砍掉，少的用padding補滿，然後decoder輸出的長度最多到12。

Model Structure



以上是 tensorboard 畫出來的圖，簡單來講 encoder 的部分使用2

層hidden layer 為 512 的 GRU，decoder用的是單層 hidden layer 為512的 GRU(就是跟助教釋出的model差不多)，使用第 encoder 第二層的 state作為 attention 的 initial hidden state。

Inference

在Encoder輸入一段話後，在decoder送入<BOS>的id，decoder就會開始依序生成答案，我們將答案最長長度設為12，但鮮少發生預測到最長長度情況，大多會是<PAD>結尾的情況。

Minor Details

最後使用的voc包含97841個單詞，每個單詞擁有100維度的word embedding。

我們在encoder和decoder的GRU layer之後都加入了dropout(rate = 0.5)，以用來防止over fitting。

- How to improve your performance (3%)
  (Please do the method different with hw2-1)
  (e.g. Attention, Schedule Sampling, Beamsearch...)
  - Write down the method that makes you outstanding (1%)

scheduled sampling，這個方法利用了在訓練時將我們預測出的結果直接放入下一階段的技巧，以增加model的容錯能力，並且希望減少訓練和預測時資料分布不一緻的問題。我們在實作的時候也使用論文中提的一種方法，在訓練的末期加入的自行生成比重才變高，以免在初期就干擾訓練，並採用線性提升的方式。

  - Why do you use it (1%)

概念較為直觀，且也有效能提昇。並且這個方法對於model的本質上並沒有太大的改面，以tensorflow實做起來也比較簡單。

  - Analysis and compare your model without the method. (1%)

我們以原本最佳的hidden layer 512 + attention 的 model在訓練時加入 scheduled sampling最為比較，最後發現加入之後可以讓 correlation 有小小的提升。而 perplexity 的分數在實驗中發現所有 model幾乎都相差無幾，因此推測使用 scheduled sampling 的 model，單就correlation的分數來講，是有提升效能的。 數據經過四捨五入。我們最終上傳的model有採用sampling。

	with sampling	without sampling
perplexity	7.81413	7.60027
correlation	0.58234	0.57311

- Experimental results and settings (1%)
  - parameter tuning, schedual sampling ... etc

Training的過程我們使用adam optimizer，將lr設定為0.001開始，並在過程中慢慢decay，並且加入gradient clipping以防止gradient爆炸。每個batch是隨機從所有training data中生成100筆data，經過90000個iteration之後的model。實驗中的model中都已經加入了attention的機制，但不包含scheduled sampling。數據經過四捨五入。

針對雙層GRU的hidden layer層數做比較：

	128	256	512
perplexity	7.86977	6.75973	7.60027
correlation	0.48439	0.54709	0.57311

可以發現單就evaluation的分數來看，增加GRU的hidden layer的size確實是對model有幫助的，由128增加到512提升在correlation的提升上非常的巨大。但就以perplexity的分數來看，其實都差不多。

範例比較(直接取testing data其中2筆為範例)：

Example 1.

輸入	你就絕不敢裝得這麼正義
128	你是個騙子
256	你是誰
512	你是個騙子

Example 2.

輸入	你是否曾經和你最愛的人說
128	我的意思是，我是說
256	我的意思是，我是說
512	我是說，我是個好人

最後我們直接以model的實際輸出結果來看，由第二個例子來看，前兩個model輸出了語意重複的話，而第三個model輸出了較讓人滿意的結果，儘管跟上下文似乎都沒什麼關係。

README:

numpy 1.14.2

tensorflow 1.6.0

gensim 3.2.0 (train word embedding用到，跑test不會用到)