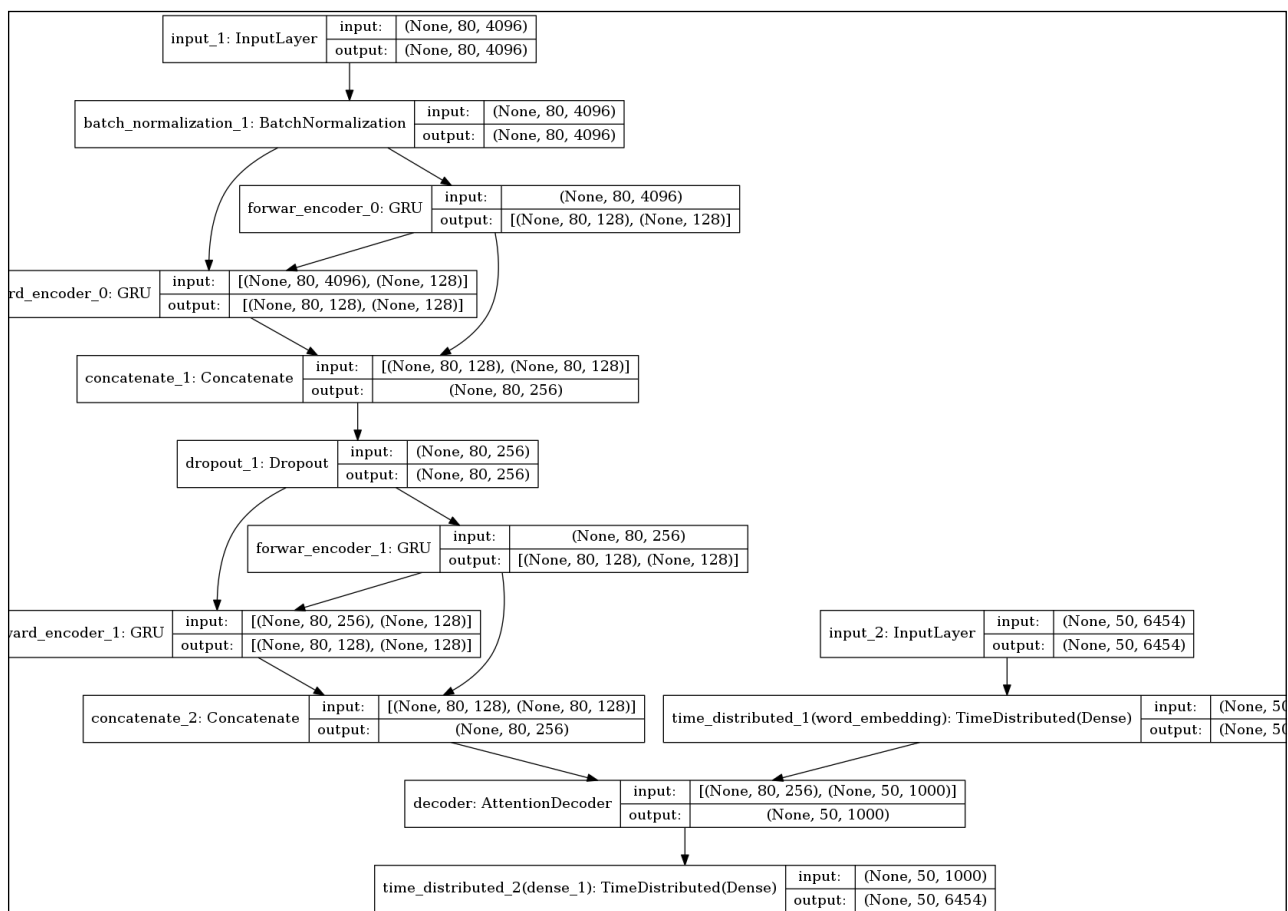


# HW2

## 模型架構

r06922115 鄭皓謙

S2VT model - 2017年11月19日



## 作業流程

這次作業使用keras寫成，一開始並沒有實作attention model。

使用兩層RNN做auto-encoder，在每個時間點使用上次輸出的label作為輸入，在訓練的時候，直接使用正確的label，而非上一個時間點的輸出。

模型的效能是在第一版belu score是0.27左右，後來加入了peekhole，決定讓decoder可以看到完整的encoder輸出，將最後一個encoder的輸出hidden state串連在decoder的每個輸入，這時效能有提高，不過有限。

觀察vocabulary set以及模型訓練初期的輸出：

[illegible]

可以觀察到，出現頻繁的字詞很容易出現，且在模型訓練完成後，句型幾乎的固定的「a man is ... | 「a woman is ....」，要輸出靈活的句型是困難的，字詞是容易的。

於是設定了sample weight，嘗試過兩種做法：

- 1.將每個字詞除上出現的頻率

想法是在每個epoch訓練完以後，每個字詞更新一樣的次數。

- ## 2. 設定weight，使其與出現頻率成反比

想到達成的效果是，希望能讓關鍵字詞先訓練出來，例如church,skateboard等等，而不是先訓練出a,man,boy等等。

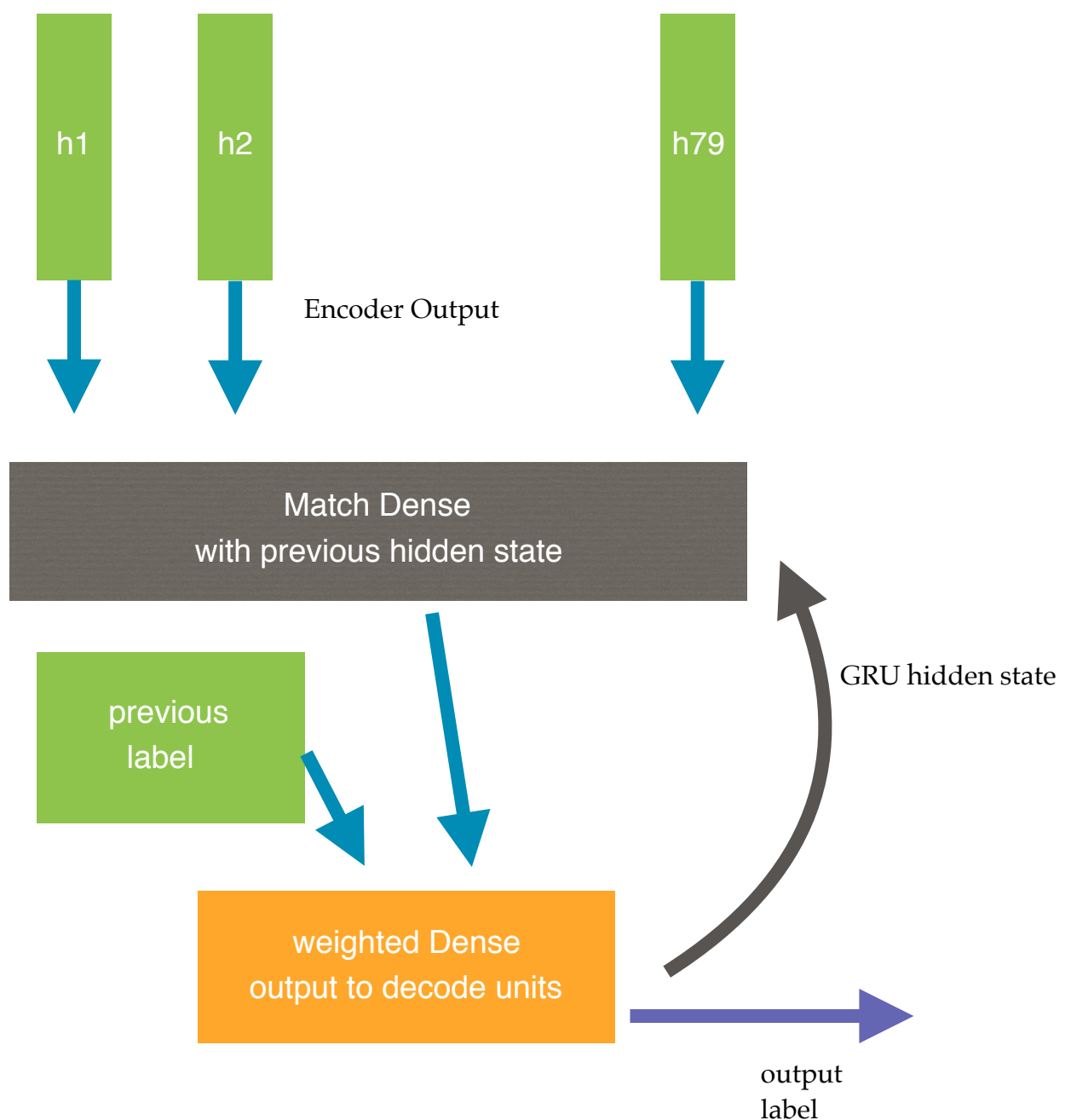
這兩種作法後來的效能都有比第一種好一些，但是並沒有提升太多。

加大了decoder encoder的hidden dim之後，模型的效能並沒有再增加，並且恐怕有overfitting的顧慮，因為這次的輸出比較難以判定好壞。

於是加入attention model，透過改寫keras的recurrent.py去改變GRU的cell operations。

實作的檔案是custom\_recurrents，另外加入了可以訓練的矩陣，有做attention match score的矩陣，也有對上一個label權重做訓練。

因為match score可以自己設計，我將架構圖寫下：



---

這個模型的效能其實並不好，belu只到了0.29。

另外也還沒有加入，在訓練時使用上一個輸出的label，因為我將上一個輸出的laebl拿來做hidden state的計算，當拿來訓練時，直接從第一個輸出就開始爆炸了。

還在嘗試用不同的match function，這個是將hidden state跟encoder的輸出做concatenate以後丟到Dense裡面，然後再跟上一個輸出concatenate，之後再丟一次Dense。

還在嘗試其他match function。

## 尚可改進的部分

首先是match function的部分，這邊是我還沒有接好各個cell的計算，應該可以讓ytm，上一個label走另外一條GRU，只有做輸出的時候才使用，就不會影響到hidden state的傳遞。

另外是vocabulary set，有聽同學討論出，人為設定部分字詞的權重，可以達到不錯的效果。

這次我來不及做好attention以及schedule sampling，我只有嘗試過直接用模型的輸出做訓練，結果爆了，上述有提到，另外以下是S2VT時就使用的模型參數，因為attention我還來不及試出比S2VT好的參數。

decoder RNN lay	: GRU depth 2 bidirectional 256 units
encoder RNN lay	: GRU depth 1 256 units
optimizer	: Adam lr = 0.001
loss	: Categorical cross entropy with mask

r06922115

鄭皓謙