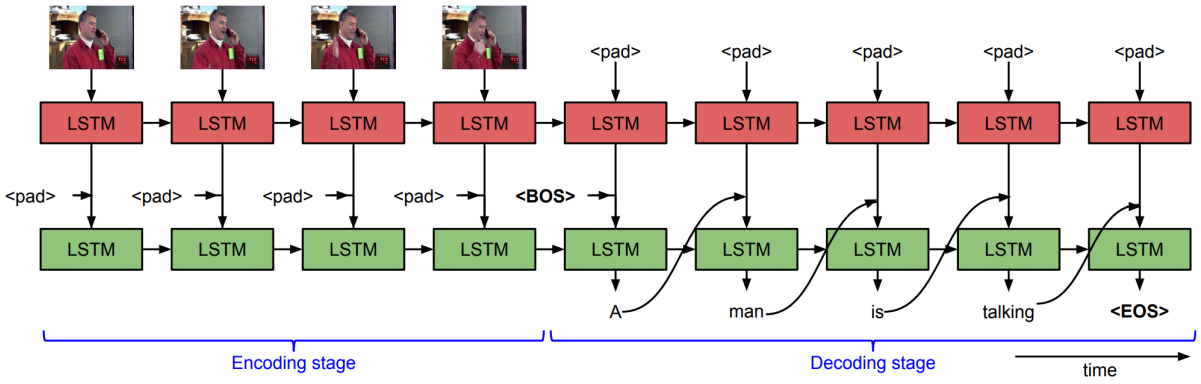
b03901057 詹晉誠

MLDS HW2 Report

A. Model Description

在本次作業中，我實作了“Sequence to Sequence — Video to Text”這篇paper提出的方法。我的network 架構如下圖所示（圖擷取自該篇paper）：



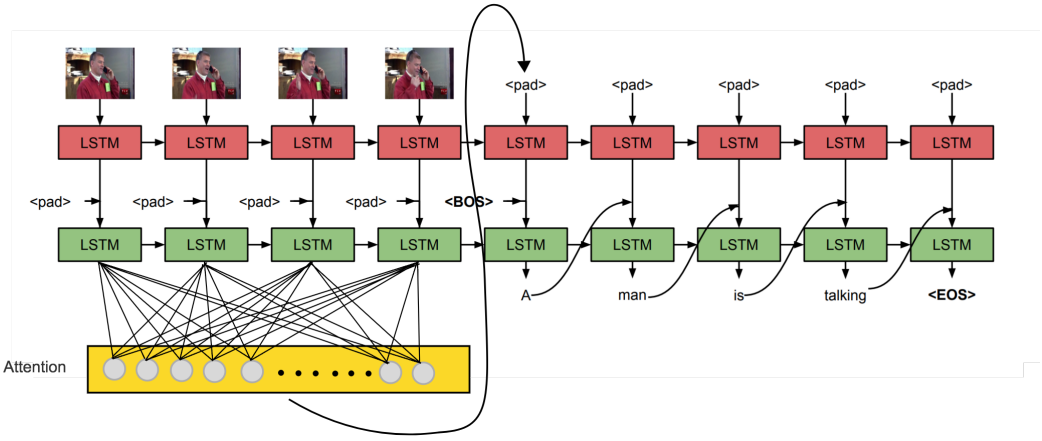
我的network依時間順序來分可以分為一個前級的encoder以及一個後級的decoder，而encoder與decoder中都是雙層LSTM的架構 — 第一層的LSTM（稱作video_LSTM) 負責接收video的input，而第二層的LSTM（稱作text_LSTM）負責產生文字的output，並且encoder中video_LSTM 與decoder中video_LSTM的參數是共享的，而encoder中text_LSTM 與decoder中text_LSTM的參數是共享的。

在training的過程中，encoder text_LSTM的input是video_LSTM在concatenate 一個zero-padded的tensor，因為在encoding stage時沒有文字輸出，而這時text_LSTM的output也不會被計算到loss中。而decoder video_LSTM的input，是zero-padded 的image，text_LSTM的input是video_LSTM的output concatenate ground-truth caption的word-embedding。此外，在我的model中，video 每個frame都會先經過一個fully-connected layer才input到video_LSTM中，而text_LSTM的每個output也會再經過一個fully-connected layer再計算loss。我計算loss的方式是把output經過softmax後再與ground-truth的one-hot encoding算cross-entropy。此外，因為label中每句的長度不一樣，於是我

固定句子長度，並將不足的補零，並在計算loss的把補零那些字mask掉，這樣就不會影響padding對loss的計算了。

B. Attention Mechanism

這次作業中我時做的attention方式如下圖所示：



Network結構基本上依照S2VT，不過我在encoder text_LSTM的輸出後端加上一個fully connected layer，然後再將fully connected layer的輸出接到decoder video_LSTM的輸入（與padding 做concatenation）。這樣設計的network原理是將整個sequence的output透過fully connected layer學習在整個video sequence中要關注在哪個部分，並且將這個資訊輸入到decoder那端。

下表列出了加上attention之前與之後model的表現以及實驗時使用的參數：

	Without attention	With attention
RNN state size	256	256
Vocab. size	6349	6349
Batch size	128	128
Learning rate	1E-03	1E-03
BLEU old	**0.250**	**0.256**
BLEU new	**0.56**	**0.56**

從實驗結果中可以發現加上attention後BLEU score有好一些，不過沒有進步很多。原因可能是因為我實作的算是比較簡單的attention，正確的實作方式應該要將decoder的output與encoder的output算similarity (cosine 或是 使用neural network)，然後再經過softmax，而我這樣實作則是假設decoder的output可以用encoder的output的線性組合表

示，雖然這樣聽起來是蠻合理的假設，但是因為時間因素，沒有辦法驗證這樣的假設是不是正確的。

C. How to improve the performance

1. 降低learning rate與batch size

在這次實驗中我認為最重要的改進是降低learning rate與batch size。在實驗過程中我發現如果learning rate太高，training時收斂的loss會比較高。這原因也是合理的，因為當learning rate太高時，參數變化的幅度太大，會使得model在minimum附近快速震盪，而無法找到真正的minimum（即使我是使用Adam optimizer），而實驗結果顯示loss越低，BLEU score會越到（有正相關的趨勢）。另一個重要的發現是降低batch size，在這次的作業中我發現batch size對learning的品質有很重要的影響（會比過去的作業還顯著），例如當我batch size太高時，收斂的loss也會比較高。雖然learning rate 與batch size對learning品質的影響是已經知道的事情，但是我認爲在video captioning中這兩個因素對learning品質有相當顯著的影響。以下列出一些實驗結果：

	High learning rate, large batch size	Low learning rate, small batch size
RNN state size	256	256
Vocab. size	6349	4000
Batch size	128	51
Learning rate	1E-03	1E-04
BLEU old	**0.250**	**0.27**
BLEU new	**0.56**	**0.61**

2. 降低vocabulary size

經過實驗發現如果將所有training 以及testing labels當做corpus的話（大約有6347個字），這樣train 出來的model效果不會很好，因為one-hot encoding 不太適用在大的vocabulary size（因為會有太多參數）。因此將vocabulary size會對training有所幫助，在我的best model中vocabulary size設為4000，其餘的用“<unk>”表示

D. Experimental results and settings

1. Best model

在我的實驗中best model是使用S2VT + attention，而參數的設定與結果如下：

	Best model
RNN state size	700
Vocab. size	4000
Batch size	52
Learning rate	1E-04
BLEU old	**0.288**
BLEU new	**0.647**

從實驗結果中可以發現將state size調大，vocabulary size縮小，batch size調小以及learning rate調低都會對learning accuracy有所幫助。

2. Scheduled sampling

此外，我也有實驗scheduled sampling的方式，但是發現效果沒有很好。在這個實驗中，我使用的decay方式是inverse sigmoid，而inverse sigmoid的參數k是設為40。以下是實驗設定與結果

	Scheduled sampling
RNN state size	700
Vocab. size	4000
Batch size	52
Learning rate	1E-04
BLEU old	**0.21**
BLEU new	**0.50**

從結果中可以發現使用scheduled sampling的反而讓BLEU score大幅下降。原因可能是因為inverse sigmoid的參數沒有調整好。根據paper所說，如果k的值太小，在training時就會太早sample model的output（相對於sample ground-truth ），因此會學不好。不過因為時間因素，沒有辦法繼續調整參數進行研究。