

# ADLxMLDS2017 - HW2

電信碩一 宋易霖 r06942076

## 1 Problem Define

從透過CNN取出的80個video frame的特徵和該影片對應的標題讓機器學習看到影片特徵預測出一個合理的標題。

## 2 Dataset

MSVD, 助教已經利用pretrain好的VGG19將每個影片取出 $80 \times 4096$ 個特徵。

## 3 Model Description

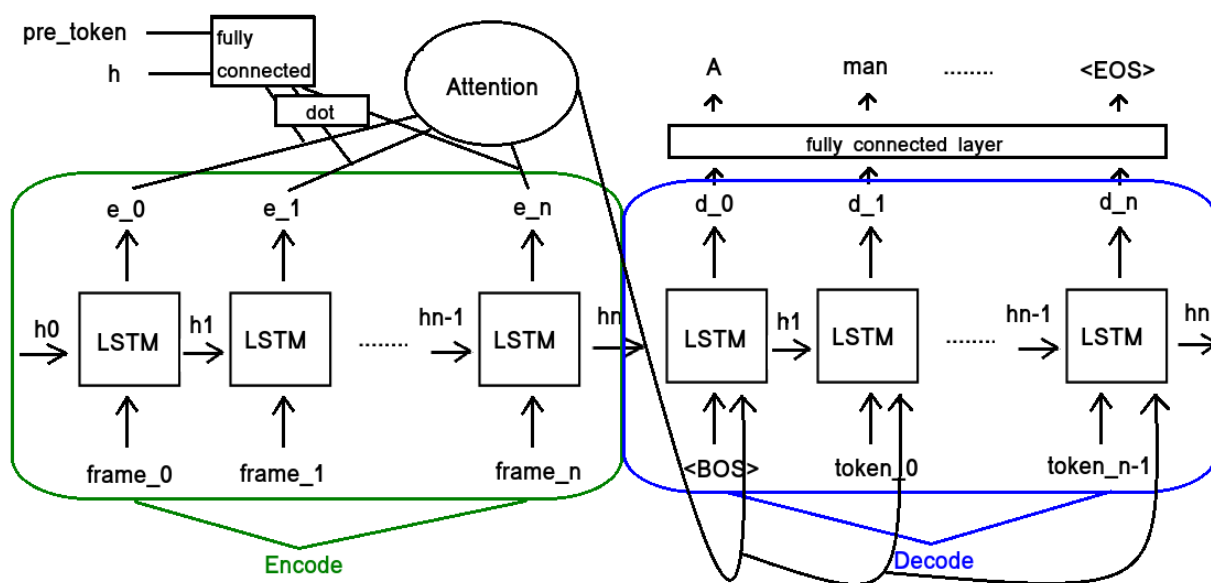


Figure 1: sequence2sequence model structure with attention

我的模型就是傳統的sequence2sequence model，前半部份的LSTM是encoder，輸入是80個frame的特徵，輸出有時間特性的影片特徵及第80個frame的hidden state當作decoder的輸入。decoder的作用是輸入encoder的輸出，並輸出一串可能的標題。而標題中每個字可能和前一個字相關，因此decoder的輸入除了encoder的輸出外，也會輸入前一時間decoder輸出的字(或是正確答案)。

## 4 Attention mechanism

Attention的部份參照老師的上課投影片及pytorch網站的教學。

假設現在正在output第 $n$ 個字，將前一時間輸出的hidden state和輸出的字(或是正確答案)接起來，然後經過一個fully connected layer及softmax算出每個encoder output對應的權重。此權重也代表decoder要focus在每個encoder輸出的比例。

$$a_t = \text{softmax}(\text{linear}([\text{token}_{t-1}, h_{t-1}]))$$

將權重和encoder的輸出做weighted sum，算出來的向量和前一時間輸出的字接起來，即為decoder新的輸入。

$$d_{input,t} = [\text{token}_{t-1}, \sum_i a_{t,i} h_i]$$

詳細的實驗結果請見Experimental Results and Settings。

## 5 How to improve your performance

### 5.1 beam search

在decoder輸出的每個字的機率中，若每次都選擇最大的這種作法不一定真的能選到最有可能性的句子，因此利用beam search每個time step都保留前 $n$ 高的機率的句子，直到有 $n$ 句子都選到 $\langle EOS \rangle$ 為止，再挑出最高機率的那個句子。

### 5.2 train with multiple caption

訓練集中的標題每個影片都對應很多標題，因此我在訓練的過程中會隨機從可能的標題中挑選出一個，如此一來可以讓機器看過大部份的標題，如此也可以避免overfitting。

### 5.3 schedule sampling while training with one caption

使用schedule sampling可以使得訓練時取前一個字的方法和測試時儘量相似，但是使用schedule sampling如果像5.2一樣又隨機取標題的話，訓練出來的機器會很差。因此當我使用schedule sampling時，會固定用某一個標題來訓練，如此一來便不會壞掉。我的schedule decay為 $0.97^{epoch}$

## 6 Experimental Results and Settings

RNN type	number of rnn layers	nodes of a rnn layer	dropout
lstm	1	200	0.5

Table 1: 實驗設定

以下我列出了我使用的幾個不同的訓練方式，但是都是使用如Table 1的結構。

A: 沒有attention，每回都隨機取標題，沒有schedule sampling

B: 使用attention，每回都隨機取標題，沒有schedule sampling

C: 使用attention，每回都固定取第一個標題，使用schedule sampling

D: 前50%的epoch使用同B的訓練方式，後面50%使用C的訓練方式

experiment type	BLEU old	BLEU new
a: A, beam size 1	0.2528	0.5315
b: A, beam size 3	0.2584	0.5292
c: B, beam size 1	0.2818	0.6102
d: B, beam size 3	<b>0.2821</b>	<b>0.6194</b>
e: C, beam size 1	0.2792	0.5951
f: C, beam size 3	0.2766	0.5966
g: D, beam size 1	0.2626	0.5730
h: D, beam size 3	0.2641	0.5743

Table 2: 不同實驗結果

從Table 2中可以發現，沒有attention的BLEU score都比有attention的還要差了不少。而使用了beam search能些微提高一些分數，但是大多數情況用greedy的方式就已經足夠好了。而在我的model上，沒有使用schedule sampling的結果是好一點的，主要因為要使用schedule sampling要固定用同一個標題，這樣的話就很容易overfitting。也因為這個狀況我嘗試了使用D這個作法，先使用全部的caption當作pretrain，後面再用固定的一個caption搭配schedule sampling，從BLEU score看來還是不好，但是這樣的訓練方式有時候會出現驚人的成果，如Table 3所示。還有一些是從special mission選的影片所預測出來的，如Table 4，可以看出這個model會加油添醋很多東西，有時後好有時候壞的...



video	B, beam size 3	D, beam size 3
	A cat is playing the piano.	A cat sitting on a piano bench is laying his head on the piano keys and playing the piano with one paw.
	A man is lifting a truck.	A bearded man lifts the back of a blue truck up and down.

Table 3: 某些影片結果




video	B, beam size 3	D, beam size 3
	A man is riding a bicycle.	A man is riding a dirt on his motorcycle.
	A person is cutting a shrimp.	A woman is cutting a piece of raw shrimp with a knife.
	A man is taking a cup of milk.	A man puts a large plastic container.

Table 4: 其他影片及其他model結果