ADLxMLDS 2017 HW2 Report
B03b02014 張皓鈞

- Sequence to Sequence Model description

概述：

主要為重製S2VT的兩層LSTM-RNN作為模型架構，上層的LSTM負責encode圖像的資訊，下層則是負責decode圖像資訊並輸出caption的單字。

每個影片的圖像都事先經由VGG-16的模型抽取出80 frames x 4096維的特徵，作為LSTM-RNN的輸入。 Caption的單字則是先建立word index的表，並加入<bos>, <eos>, <pad>, <unk>符號，分別作為「開始句子」、「結束句子」、「填充長度用」、「未知單字」的意義，在輸入前先去除標點符號、加上上述符號並轉成word index，將caption用one-hot encoding表示，最後以categorical cross-entropy為loss function進行訓練。

實作細節：

使用Tensorflow實作S2VT所提出的兩層LSTM-RNN，上下層的hidden dimention皆相同。而圖像特徵輸入時先經過一層embedding layer將特徵map到hidden dimension的維度，caption的單字也同樣經過embedding layer的處理。 在Encoding階段，上層LSTM輸入圖像特徵，上層的output和embed後的<pad>符號接在一起並輸入下層LSTM，不管此時的下層output。在Decoding階段，上層LSTM輸入padding的tensor，上層的output和embed後的單字（此處為caption前一個單字，若是第一個則是用全為0的tensor）接在一起並輸入下層LSTM，此時的下層output為預測出來的單字。

以下為前處理、訓練所使用的Hyper-parameter以及後處理整理：

使用單字			訓練Labels			
使用所有training data中的captions			每一個epoch隨機選取一個caption			
Batch Size	Word Count Threshold	Hidden Dimension	Training Epochs	後處理	Old BLEU Score	New BLEU Score
50	保留所有word	512	500	切掉<pad>以後的單字	0.26476	0.57531

備註： 在模型預測結果中，我發現產生的句子皆沒有<bos>和<eos>，而有許多<pad>，因此才切掉<pad>之後的單字，根據推測，原因在於第一個放入decoder的embed單字為全0 tensor，在word index表中所代表的字為<pad>。

- Attention mechanism

實作方法： 先將Encoding階段，下層LSTM每個timestep的輸出存成一個tensor（h_prev）。在Decoding階段之前，先將h_prev通過一層dense layer，得到hidden dimension大小的attention vector。到了Decoding階段，在每個timestep中，將attention vector輸入進上層LSTM，進行decoding。

加入attetion mechanism到S2VT模型後的訓練結果：

使用單字			訓練Labels				
使用所有training data中的captions			每一個epoch隨機選取一個caption				
Batch Size	Word Count Threshold	Hidden Dimension	Attention	Training Epochs	後處理	Old BLEU Score	New BLEU Score
50	保留所有word	512	No	500	切掉<pad>以後的單字	0.26476	0.57531
			Yes			0.26894	0.57607

比較：具有Attention機制的模型輸出的句子都是"a man is playing a guitar"，雖然在BLEU score上皆比沒有Attention的模型還要高，但是就肉眼觀察，此模型的表現還是沒有原本的好。

- How to improve your performance

1. 考慮到實際應用上不會有testing data的captions，所以在建立單字表時，我去掉了testing data的caption。
2. 另外在選取訓練Label的時候，每個epoch隨機選取一個caption作為答案，在多個epoch之後可以讓模型對同一個影片學到更多種caption，而這些caption都是表達同一個影片的。
3. 使用GRU作為RNN的cell。原因在於GRU具有較少的參數量，而且在cell內部機制是整合過LSTM內的gates所設計出來的，因此我認為會有比較高效率的表現，在以下實驗結果也發現GRU的表現比LSTM好。

- Experimental results and settings

在此部份的實驗皆是以上述基本seq2seq模型(無attention機制)架構為對照組進行修改。

1. GRU-seq2seq

我另外嘗試了使用GRU作為RNN的cell，其訓練結果如下：

使用單字			訓練Labels				
使用所有training data中的captions			每一個epoch隨機選取一個caption				
Batch Size	Word Count Threshold	Hidden Dimension	RNN Cell	Training Epochs	後處理	Old BLEU Score	New BLEU Score
50	保留所有word	512	GRU	500	切掉<pad>以後的單字	0.27442	0.61457
			LSTM			0.26476	0.57531

說明：GRU cell的分數較LSTM cell的分數高，原因可能在於GRU具有較少的參數量，可以避免overfit一些影片的單字。

2. Schedule Sampling

在這裡我選擇實作exponential decay的schedule sampling。也就是一開始在第一個batch中，「選取正確訓練Label」的機率為1.0，隨著訓練次數增加，「選取正確訓練Label」的機率隨著減少，而「從decoder output分佈中sample出訓練Label」的機率隨著增加。每個batch輸進模型前，以指定機率來決定該batch是用ground truth或是從sample出來的caption。
簡要來說，P(「選取正確訓練Label」) = k ^ (batch次數)，而k < 1為一常數。

使用單字			訓練Labels				
使用所有training data中的captions			每一個epoch隨機選取一個caption				
Batch Size	Word Count Threshold	Hidden Dimension	Schedule Sampling	Training Epochs	後處理	Old BLEU Score	New BLEU Score
50	保留所有word	512	Yes (k=0.6)	500	切掉<pad>以後的單字	0.20584	0.54730
			No			0.26476	0.57531

說明：預測結果都是"a man is playing a"，顯示我所實作的這種schedule sampling方式存在著缺陷，假設沒有人為疏失的話，每個batch隨機選擇ground truth或是sample出來的caption，這樣可能會在模型訓練早期就有某個batch都是sample出來的caption，影響gradient descent的方向。