ADL HW2: Video Captioning Report

R06725035 陳廷易

Model description

A. 此次作業二的 model 主要是參考<sequence to sequence - video to text>論文的 seq2seq model 來進行實作與修改。
B. 不論是 encoding state 或是 decoding state 皆為重複使用相同的兩層 LSTM 架構，並非各自分開訓練。
C. 在 encoding state 會利用助教所提供的 80 frames* 4096 維的 vector 做為輸入，並將生成的 output 餵給第二層 LSTM。
D. 在 decoding state 則將 encoding state 第一層 LSTM 輸出的 state 做為 decoding state 第一層 LSTM 的 input；同樣地將 encoding state 第二層 LSTM 輸出的 state(i.e,<BOS>)傳遞給 decoding state 第二層 LSTM 的輸入，此外亦會將第一層 LSTM 的 output 做為第二層 LSTM 的輸入。Decoding state 也會將當前產生的字做為下一個 timestep 的輸入，不斷重複此動作直到產生<EOS>為止。

Attention mechanism

在 attention 方面則是實作 soft-alignment，會將 encoding state 第二層 LSTM 每個 timestep 的輸出記錄起來，經過權重與 bias 後傳入 decoding state 的第一層 LSTM，而整個神經網路會去學習所應賦予的參數為何。

然而因為 encoding state 與 decoding state 的架構是一樣的，意味著兩著的輸入維度要相同。在 encoding state 的第一個 LSTM input 因為沒有 attention value，所以會附加 padding vector 在前面；而在 decoding state 因為沒有影像的 vector，所以會在後面附掛 padding，以使兩者輸入的維度一致。

增加了 attention 以後，從 predict 的結果不難發現重複的字詞得以大幅下

降，不會只著重於某些 frame 當中而出現如 a man is a man 的句子。此外，若沒有 attention 則句子通常只會抓住大方向的畫面動作、較為籠統，而若增加 attention 則可以捕捉更多細節的描述。

How to improve your performance

- 資料清洗、濾除罕字、正規化
- 資料前處理、initial vector
- 增加 Hidden dimension、選擇 optimizers
- 調整 activation function、Dropout

在最一開始時，逕行將所有的 training label 當成字典建立 one-hot vector，然而除了效率差以外，效果也不好。因此改將 training label 及 testing label 一起做為字典，並將辭頻小於 3 者濾除，可大幅度提升成效。而若詞頻 threshold 過高則會使 model 預測出來的結果過於僵化；反之若太低則會使 unknown 機率增加。此外，在文字處理部分也藉助 re 套件，進行文字正規化，將原先雜亂的 label 文字清洗掉一些特殊符號或無意義的字詞。

在訓練 decoder 時若是使用 predict 的結果餵入下一個 timestep 會使 model 很難 train 起來，因此改為將 label 的答案餵入下一個 timestep 能相當程度地提升 model caption 的能力。

在模型方面，若 hidden dimension 能大一些基本上可以抓住更多的資訊，而 batch size 則不要小於 20 對結果不會有太大的影響。在 Optimizer 部分，除了 SGD 難以收斂外，Adam 與 RMSProp 的收斂速度都滿快的(約 1000 epochs 以內就會收斂完成了)，但又以 RMSProp 的結果稍微好一點。

而為了防止 overfitting，後來也增加 dropout 機制(0.5)，此舉雖讓收斂速度下降了不少，但是對 predict 出來的結果卻有不小的助益。關於 activation

function 部分，嘗試過 relu 與 elu，基本上加與不加對結果而言效果不大，但得以一定程度提升收斂的速度。

Experimental results and settings

在訓練過程中，大約每經過 10~30 個 epoch 於參數上就會有不小的調整，有時預測結果是變好的，當然也有時是會變差的，在這樣的情況下要找到最好的 model 就會變得不太容易。而我採用的方法是將助教所提供的兩種 bleu 分數進行平均，前提是皆通過兩種 baseline，才從平均分數最高的做為該 model 的代表，接著再跟其他的 model 進行比較(如有否 activation function、有否 dropout、不同的詞頻 threshold、有否 attention 等等)。而在跨 model 間的比較則是利用 predict 出來的句子，以判讀的方式來評估何種 model 的結果是比較好的。

從觀察中也發現加上 attention 後可以使句子平均的質量比較一致，而沒 attention 者則比較容易遊走灰色地帶、模糊，且質量比較不穩。但其實也很不易判定孰優孰劣，端看所預測的影片為何。

因為時間上的緣故，或許若加上 scheduled sampling 以及 beam search 的話就能夠相當程度改善最終預測的結果了。

Experimental Settings

- Sequence to Sequence Model: 2-layer LSTM with Attention Mechanism
- Hidden Dimension: 880
- Epochs:660
- Batch size: 50
- Learning rate: 0.0001
- Loss function: Softmax_Cross_Entropy_with_logits

- Optimizer: RMSProp