深度學習應用——作業二報告

B03902072 江廷睿

1 Model description

1.1 參數與最佳化方式

- 批的大小：64
- 最佳化方式：Adam
- 學習率：0.001

1.2 架構描述

參考 [1] 提出的 hLSTMat 架構以及一些誤解，實作了如下的架構：

1.2.1 編碼器

使用助教提供的，對於每個訊框，用 VGG16 抽取的特徵：

$$V = \{v_1, v_2, \cdots, v_{80}\} \tag{1}$$

1.2.2 解碼器

- 底層的 LSTM 以上一個字的 embedding 作為輸入，參考原論文， LSTM 的大小是 512 。[1]

$$\begin{aligned} h_0^\vee, c_0^\vee &= \tanh([W^{ih}; W^{ic}]\text{Mean}(v_1, v_2, \cdots, v_{80})) \\ h_t^\vee, c_t^\vee &= \text{LSTM}(\text{embedding}(y_t), h_{t-1}^\vee, c_{t-1}^\vee) \end{aligned} \tag{2}$$

  其中 embedding 與 W^{ih}, W^{ic} 都是要學的參數， 參考作者的原始碼使用 embedding 維度 512 。

- 上層的 LSTM以下層的輸出 $h^\vee$ 作為輸入，參考原論文， LSTM 的大小也是 512：

$$h_t^\wedge, c_t^\wedge = \text{LSTM}(h^\vee, h_{t-1}^\wedge, c_{t-1}^\wedge) \tag{3}$$

[1]這邊的 tanh 是參考原作者的原始碼，在原論文中沒有出現。

- Attention：在每個時間點 t 使用下層 LSTM 的輸出 $h_t^\wedge$ 計算出權重

$$\alpha = \text{softmax}(w^T \tanh(W_a h_t^\wedge + U_a V + b_a)) \tag{4}$$

其中 w, W_a, U_a, b_a 都是要學的參數。原論文沒有說明這之中的維度，原作者的原始碼不太合理的使用影片特徵的維度，所以這部份我隨意的設成 128 。然後算出 attention [2]

$$a = \sum_i \alpha_i v_i \tag{5}$$

- 輸出層：將上層 LSTM 的輸出 $h^\wedge$ 與 attention a 相接後，輸入一個兩層的神經網路，並用 softmax 把輸出轉化成預測每個字的機率：

$$P(y) = \text{softmax}(U_p \tanh(W_p[h^\wedge; a] + b_p) + d) \tag{6}$$

其中中間層的維度參考原作者可怕的原始碼，同樣使用 512。

- Dropout：在上述的模型中，對 $h^\wedge, h^\vee, a$ 以及多層神經網路的中間層使用 dropout ，丟失比例參考論文使用 0.5 。

2 Attention Mechanism

2.1 How do you implement attention mechanism?

請參考 1.2.2 中 Attention 的部份。

2.2 Compare and analyze the results of models with and without attention mechanism.

這個架構在拿掉 attention （把 attention 換成 0 向量）後的 BLEU@1 分數只有 0.66 ，而若是加上 attention ，分數則有 0.70 。另外 S2VT [2] 這個沒有 attention 的架構也只有 0.66 。因此，顯然 attention 的確能增進模型的效果。

3 How to improve your performace

3.1 特徵標準化

把影片的特徵的每個維度都減去他們的平均值並除以他們的標準差，使得每個維度的平均值為 0 ，標準差為 1 。理論上這可以讓梯度下降的過程更順利，並加速訓練。實際上這的確能使模型的 bleu score 上升得更快。

3.2 兩層的 LSTM 架構

根據論文，這種架構的理念是希望上層的 LSTM 可以專住在學習語言模型，而下層的 LSTM 則是專住在處理影片的資訊。實際上而言，因為這個架構沒有使用 RNN 編碼影片，所以訓練速度會快上許多，而且 BLEU@1 分數能達到 0.70 ，因此最終決定採取這個架構。

[2] 原論文中 attention 前面多乘了一個 $\frac{1}{n}$

3.3 Beam Search

Beam Search 是一種廣度優先搜尋演算法的變形，與廣度優先演算法的差異是 Beam Search 在每個深度只會保留分數前幾高的結果，保留的數量稱為 beam size。假如 beam size 等於 1，結果則跟貪婪演算法一樣，很有可能部會得到全域最佳解；假如 beam size 設為無限，則跟暴力搜尋一樣，最後的最佳解一定是全域最佳解，但時間雜度也是指數型的。使用一個大於 1 而小於無限大的 beam size 可以在兩者之中達到平衡，有機會可以在有限的時間內獲得較好的結果。

4 Experimental results and settings

4.1 Schedule Sampling

實驗發現使用 schedule sampling 沒辦法有效的增加 BLEU 分數，此外使用 Schedule Sample 容易使模型產生不合文法的句子，像是「A man is a a」，因此最後沒有使用 schedule sampling。

4.2 改變 Embedding 大小與 Hidden Layer 的維度

這裡嘗試著改變 embedding 維度與輸出神經網路的 hidden layer 的維度，但因為原論文（程式碼）中 embedding 大小與 hidden layer 維度都相同，因此只有測試他們使用相同維度的狀況。在嘗試過 256、 512、768 的大小之後，發現這三組參數的 BLEU@1 分數都在 0.69 到 0.70 之間。

4.3 改變 Dropout Rate

雖然最後的模型是使用 dropout rate 0.5 訓練出來的，但實驗證明在維度等於 512 的情況下，沒有 dropout 或是 dropout rate 等於 0.25 的模型也可以達到 BLEU@1 0.70 的分數。

References

[1] J. Song, L. Gao, Z. Guo, W. Liu, D. Zhang, and H. T. Shen, “Hierarchical LSTM with adjusted temporal attention for video captioning,” in Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, pp. 2737–2743, 2017.

[2] S. Venugopalan, M. Rohrbach, J. Donahue, R. J. Mooney, T. Darrell, and K. Saenko, “Sequence to sequence - video to text,” in 2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015, pp. 4534–4542, IEEE Computer Society, 2015.