NLP_HW2

NTU b10705013 陳彥廷

Questions

Q1: What impact does using different learning rates have on model training?

A1:

在作業中預設的 lr 是 0.001，在我的環境下訓練 2 個 epoch 之後得到的結果是 0.6756，而我嘗試了另外兩個參數，分別是 0.005 和 0.0001，分別得到的結果是 0.64 與 0.62。

這個結果顯示了 lr 並不是越大越好或是越小越好，我們應該根據實驗的結果來選擇一個適中的參數。當 lr 過大時容易導致無法收斂或是跳過局部最佳解且較為不穩定，而 lr 過大則容易被困在局部最佳解。

Q2: If you use RNN or GRU instead of LSTM, what will happen to the quality of your answer generation? Why?

A2:

在這次作業中，我主要嘗試了 LSTM 與 RNN 兩種方式，由於 RNN 的記憶能力較差 (容易發生梯度消失的問題)，因此結果較差。

而 LSTM 在記憶的方面表現明顯優於 RNN，正確率約能達到 7 成左右。

Q3: If we construct an evaluation set using three-digit numbers while the training set is constructed from two-digit numbers, what will happen to the quality of your answer generation?

A3:

我認為如果在 eval 中出現三位數可能會導致結果明顯下降，因為對於整數的

四則運算來說，增加位數就代表模型需要記更長的位數來進行推理，而當 training set 都沒有三位數時，模型就不會做這種記憶，從而影響結果。

Q4: If some numbers never appear in your training data, what will happen to your answer generation?

A4:

我認為這個問題在這次的作業中不太需要擔心，因為我們是使用數字來 build dictionary，如 22 會被 label 成 '2'、'2' 而不是 '22'，所以不需要特別處理數字沒看過的情況，模型會自動從其他計算中推論數字與數字的關係。

Q5: Why do we need gradient clipping during training?

A5:

在神經網路中，梯度需要透過反向傳播來進行更新，如果梯度在多層的網路中反覆相乘就有可能會導致梯度變化過於劇烈並出現無法收斂的情況，因此使用 gradient clipping 可以有效幫助我們進行收斂。

Q6: Different with sample code

A6: 在這次的作業中，我採取了與作業不同的前處理方法，這是因為我發現如果使用助教所提供的前處理方法會導致模型只能學習到照抄 '=' 後面出現的字，從而導致正確率達到 1 的情況發生，因此我採取了不同的前處理方法，以 14*(43+20)=88 這個例子為例

sample code 的前處理結果

[3, 4, 5, 6, 4, 7, 8, 9, 10, 11, **2**, 12, 12, 9, 1]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, **0**, 12, 12, 9, 1]

我的前處理結果

[3, 4, 5, 6, 4, 7, 8, 9, 10, 11, 2, 12, 12, 9]

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 12, 12, 9, 1]

可以發現我將 char_id_list 的 eos 移除，並將 label_id_list 左移一格，這樣就能有效避免前面所述的問題。

Settings

All code run in win11、CPU i5-12400、GPU 4070s、Python 3.10.11

References

Chatgpt and Copilot.