Assignment2 Report

Name: 李威辰

Institution (school): 國立清華大學

Student ID: 112062591

Platform (Colab/Kaggle/Local): Local

Python version: 3.12.3

Operating system: Ubuntu 24.04.2 LTS

CPU:

CPU 型號：AMD Ryzen Threadripper 7960X（24 核心 48 線程）

架構：x86\_64（支援 32/64 位元）

核心數 / 線程數：24 核心 × 2 線程/核心 = 48 線程

GPU:

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

AI 產生的內容可能不正確。

Remark: For ease of grading, you are encouraged to present data in textual form rather than as images.

Present your hyper-parameters in training, including learning rate, batch size, hidden size, epochs(steps), etc. (5%)

Answer:

Base line:

batch\_size = 64

epochs = 2

embed\_dim = 256

hidden\_dim = 256

(learning rate)lr = 0.001

grad\_clip = 1

Best result:

hidden\_dim = 384

(first LSTM layer count)lstm\_layers\_1 = 1

(second LSTM layer count)lstm\_layers\_2 = 2

dropout = 0

embed\_dim = 256

batch\_size = 64

(learning rate)lr = 0.001

grad\_clip = 1.0

epochs = 3

If you use RNN or GRU instead of LSTM, what will happen to the quality of your answer generation? Why? (10%)

Answer:

LSTM :

* 特性：
  + Cell state：長期記憶，信息可以跨越很長的序列保留。
  + Forget gate：控制哪些信息要丟棄。
  + Input gate：控制哪些新信息要寫入。
  + Output gate：控制輸出哪些信息到下一層或下一時間步。
* **優點：** 能夠記住序列中很早的關鍵信息，穩定性高，適合超長序列的數據。
* **缺點：** 參數多，計算量大。

RNN :

* **特性：** 每個 hidden state 都是前一個 hidden state 和當前輸入的函數。
* **缺點：** 容易梯度消失或爆炸，較長的序列的信息難以保留。

GRU :

* **特性：** 比 LSTM 簡單:
  + update gate : 決定保留多少舊信息、加入多少新信息
  + reset gate : 決定忘掉多少舊信息
* **優點：** 能比普通 RNN 更好地記住長期依賴，且參數更少、訓練速度更快。
* **缺點 :** 沒有單獨的 cell state，較無法精細控制哪些信息要保留、哪些要丟棄，非常長序列GRU 比 LSTM 更容易忘掉早期信息。

以下使用baseline參數和印出generator的結果:

RNN :

Accuracy : 0.155688509

1 epoch training time : 2 m

1 epoch evaluating time : 10 m

12-45=-35

12\*45=594

(12-45)\*4=-144

(9-22)\*10=-120

(9-22)\*(2+8)=80

333\*333=16200

GRU :

Accuracy : 0.452774929

1 epoch training time : 2 m

1 epoch evaluating time : 10 m

12-45=-33

12\*45=540

(12-45)\*4=-132

(9-22)\*10=-130

(9-22)\*(2+8)=144

333\*333=1323

LSTM :

Accuracy : 0.665413105413105

1 epoch training time : 2 m

1 epoch evaluating time : 10 m

12-45=-33

12\*45=540

(12-45)\*4=-132

(9-22)\*10=-130

(9-22)\*(2+8)=-36

333\*333=1354

可以明顯看出準確率有符合LSTM>GRU>RNN，但訓練時常並沒有太大變化，或許跟training set裡面數學式子比較短的原因有關，如果可以生成長一點的訓練集或許可以看出更大差距。

If we construct an training set using three-digit numbers while the evaluation set is constructed from two-digit numbers, what will happen to the quality of your answer generation? (10%)

Answer:

使用chatGPT 生成的程式碼創建three-digit training set，使用baseline參數。

Evaluating accuracy:

|  |  |  |
| --- | --- | --- |
| epoch | 2-digits | 3\_digits |
| 1 | 0.580057 | 0.00033 |
| 2 | 0.665413 | 0.000403 |

Generate:

2-digit:

12-45=-33

12\*45=540

(12-45)\*4=-132

(9-22)\*10=-130

3-digit:

12-45=-232

12\*45=45100

(12-45)\*4=-11910

(9-22)\*10=49020

可以看出多了3位元的數字進入運算準確率低了非常多，由generator產出的結果也可以看出。其原因可能為:

1. Overfitting：模型會主要學會處理三位數字，所以當它遇到二位數字預測結果會有困難。
2. 難度差異 : 可能運算三位元有更多的進位借位的數字運算問題，跑到二位數字時由於便單純，模型有可能因此出錯。

If we construct a training set that includes 20% incorrect answers, how will this affect the quality of the generated responses? Present some examples. (5%)

Answer:

使用chatGPT 生成的程式碼創建20% incorrect answers training set，使用baseline參數。

|  |  |  |
| --- | --- | --- |
| epoch | baseline | 20%\_wrong |
| 1 | 0.580057 | 0.151776 |
| 2 | 0.665413 | 0.16762 |

Generate:

baseline:

12-45=-33

12\*45=540

(12-45)\*4=-132

(9-22)\*10=-130

20%\_wrong:

12-45=-33

12\*45=540

(12-45)\*4=-1274

(9-22)\*10=590

可以看出20% wrong answer數字進入運算準確率低了非常多，由generator產出的結果也可以看出。其原因為模型可能會學習到不正確的答案，導致模型將某些輸入與錯誤的輸出聯結起來，從而降低準確率。

Why do we need gradient clipping during training? (5%)

Answer:

在深度網路中，梯度需要經過很多層反向傳播，有時梯度會指數級增長，變得非常太大導致loss值NaN。

在pytorch中可以用

torch.nn.utils.clip\_grad\_value\_(model.parameters(), grad\_clip) # gradient clipping

每個梯度的絕對值不能超過grad\_clip，常設成 grad\_clip = 1.0。

e.g. grad\_clip = 1.0,

簡單容易保護參數不爆炸，但梯度的方向有可能錯誤

… Anything that can strengthen your report. (5%)

Answer:

這裡把原本的training set加入” / ”除法來去看有甚麼變化，使用best result參數。

Evaluating accuracy:

|  |  |  |
| --- | --- | --- |
| epoch | best | divide |
| 1 | 0.66 | 0.47 |
| 2 | 0.76 | 0.58 |
| 3 | 0.81 | 0.63 |

Generate:

12-45=-33

12\*45=540

(12-45)\*4=-132

(9-22)\*10=-90

(12-45)/3\*2=-24

可能跟three-digit的情況有點像，就是遇到較簡單沒有除法的情形準確率會下降(遇到簡單的反而有點轉不過來)，但其實accuracy也差不多只少了一點點。

The screenshot of your training logs and evaluation accuracy. (One Figure only) (10%)一張含有 文字, 螢幕擷取畫面, 軟體, 字型 的圖片

AI 產生的內容可能不正確。