**Report：NLP\_HW2 成功大學\_N96134645\_許宸華**

**Running environment：Colab Python version：Colab**

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述首先是結果呈現：一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述

在使用learning rate = 0.001, epochs = 5的情況下，準確率約為0. 88，我認為是可以接受的準確率，但如果epochs再增加表現會更好。

一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述在測試中，對於一位數的加減乘是可以做到完全正確的。然而在二位數的加減乘中，可以看出隨著生成的長度越長，正確率越低，通常10位數還會生成正確，但個位數就不正確了。

準確率：  
在一開始準確率的評估一直是0，後來在chatgpt的幫助下發現是因為模型在生成padding時，id不會與true table的padding一致 (因為在訓練時我們不在乎padding的位置生成甚麼，只關注在=後面的答案是否生成正確) 所以在評估時應該把padding與eos後面補的padding都給刪掉再來比對。

**一、What impact does using different learning rates have on model training?**

在這次實驗中，我總共使用了4種不同的learning rate ，分別是0.001、0.0001、0.005、0.0005，在其中0.001的表現最好。以下是結果比對

0.001：一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

0.0001：一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

0.005：一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

0.0005：一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

在0.001~0.0001之間可以learning rate越大準確率越高，學習率一旦過小會導致模型學習速度過慢，可能會遇到loss的local minimum而找不到global minimum。然而一旦超過一個域值 (在測試中0.001最接近這個域值) 梯度就會因為學習率過高而無法達到最低點，像使用0.005作為學習率時準確率就非常差，就是因為過大的學習率導致的模型表現很差。

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

以下是同結果用learning rate = 0.001, epochs = 5來進行訓練的RNN模型。

一張含有 文字, 字型, 代數, 螢幕擷取畫面 的圖片

自動產生的描述**一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述**

改用RNN後的訓練結果出乎我意料的差，準確率甚至不到1%，我預期只會比LSTM差一點沒想到差那麼多，我認為的原因有以下：

1. 梯度消失和梯度爆炸問題：  
   RNN 在反向傳播過程中，當（sequence length）增長時，誤差會逐步傳回每個sequence。如果序列較長，梯度容易消失或爆炸，這讓 RNN 無法有效學習序列中遠距離的依賴關係。這是因為 RNN 只具有單一的隱藏狀態，無法儲存長時間的資訊。
2. 缺乏記憶能力：  
   LSTM引入了一些Gate來控制記憶與否，可以根據資料學習選擇性地保留或遺忘過去資訊。這讓 LSTM 能夠有效捕捉長期依賴的特徵。而標準 RNN 無此機制，只能依賴當前的隱藏狀態，因此容易忽略遠距離的資訊。

**三、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?**

一張含有 文字, 字型, 行, 白色 的圖片

自動產生的描述

由於資料的缺乏在生成三位數算式的答案時已經看不出生成的邏輯跟規律了，由此可見在使用AI時data的重要性。

**四、If some numbers never appear in your training data, what will happen to  
your answer generation?**

因為在轉換成id的時候會因為沒出現過造成error，所以我沒有成功產生結果，但有嘗試一些”不存在的算式”，而結果如下：一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述

有趣的是在生成1+\*1=時他似乎判定跟1+1=是類似的狀況，我認為跟LSTM有記憶能力是相關的。而模型就算在輸入奇怪的時候還是可以判斷大概要輸出多長的答案並生成<eos>是超乎我預料的結果。

**五、Why do we need gradient clipping during training?**

在深層網絡或處理長序列的循環網絡中，梯度會隨著時間步長的增多而指數級增大，導致梯度值非常大。這就是梯度爆炸，梯度爆炸會使模型變得不穩定或發散，無法有效學習。gradient clipping就是為了防止梯度爆炸，它會使梯度不會超過一個值，控制梯度的範圍，有助於模型的泛化能力和收斂效果，才不會讓loss爆走。

補充：在後來有發現一開始先用0.001的learning rate進行訓練後，再反覆使用更小的learning rate進行訓練，會讓準確率越來越好，雖然Adam的更新學習率方式已經很通用方便，但如果能依照模型需求手動去提升learning rate會有更好的效果。

一張含有 文字, 螢幕擷取畫面, 字型, 行 的圖片

自動產生的描述