1. Model analysis  
   本次2a作業中我所使用的模型，架構為一層embedding layer，他將經過tokenization過的每一個word，轉換為輸入向量，並輸入RNN layer 。接下來的一層RNN layer，因為是簡單的三個數字加減乘除問題，使用一層RNN layer訓練資料即可。最後接上一層fully connection layer，將輸出的結果壓縮至一類。在loss function的選擇上，我使用了mean square error 作為loss function，目的是為了解決本次2a作業的回歸問題。優化器使用的是Adam。
2. Dataset analysis

我設計的資料集有兩種各100000筆資料，一是90%是三個數字的加減運算，10%是三個數字的加減乘除運算，二是全部都是加減運算，但是50%是三個二位數運算，而50%是三個個位數運算。

我認為第一點，數據量的多寡會影響訓練結果，像是第一個資料集的加減運算會比加減乘除的預測結果好。理由是模型對於\*/的運算是比較陌生的，預測出來的結果理所當然會比常見的+-運算還要來的差。

圖左10000筆三個數乘除測試結果 圖右10000筆三個數加減的資料測試結果

一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

自動產生的描述

而第二點，數據的結果範圍會影響訓練結果，像是第二個資料集的個位數運算會比二位數的預測結果好。理由是對結果範圍較大的資料來說，應當需要更大量的數據，來訓練模型更多樣的組合，因此在二位數的預測上，預測出來的結果會比個位數運算還要來的差。

圖左10000筆三個二位數測試結果 圖右10000筆三個個位數的資料測試結果

一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

自動產生的描述

1. Discussion

在模型訓練中，learning rate越大，會讓gradient 較難收斂到最小值，並且會容易震盪，不過訓練的過程會比較快，learning rate越小，會讓比較容易將gradient收斂到最小值，不過訓練的過程會比較慢。

左圖 learning rate =0.001 右圖learning rate = 1

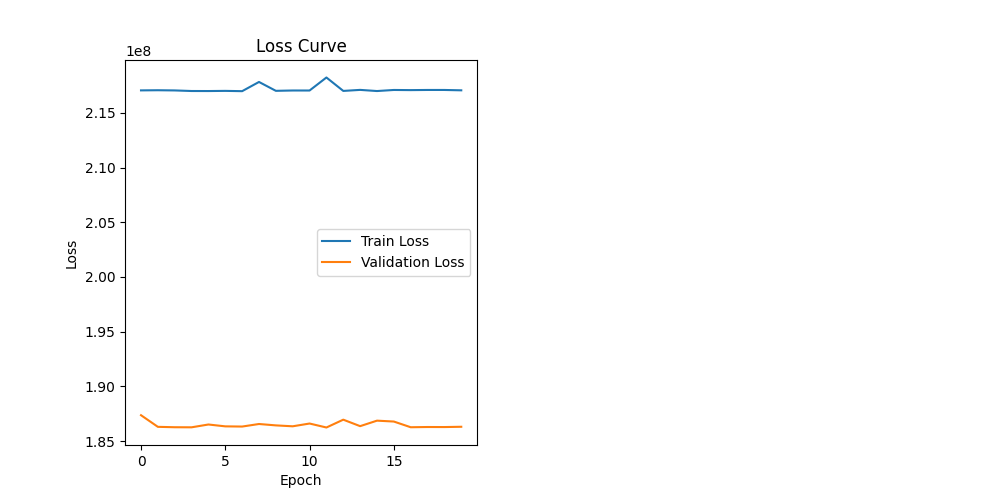
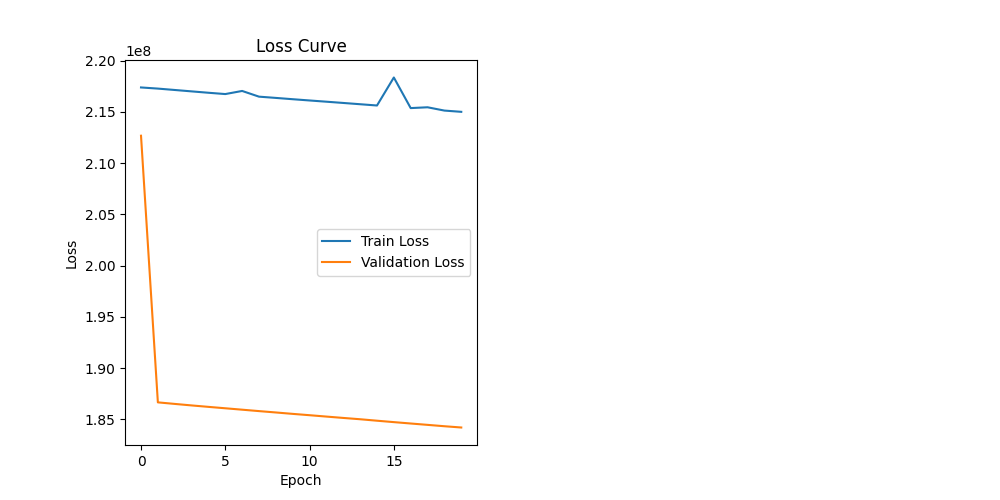
Dataset : 50%是三個二位數運算 50%三個個位數運算

一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

自動產生的描述一張含有 文字, 圖表, 行, 螢幕擷取畫面 的圖片

自動產生的描述

Dataset: 90%三個數字的加減運算10%三個數字的加減乘除運算



較大的 batch size，能收斂的 learning rate 區間通常較小，且loss也較大。而較小的 batch size 能在較多的 base learning rate 區間上成功收斂，代**表 batch size 較小時，能讓模型收斂的學習率範圍很大，因此更容易收斂。**

左圖 batch size=1024 右圖 batch size=16

Dataset : 50%三個二位數運算 50%三個個位數運算

一張含有 文字, 圖表, 螢幕擷取畫面, 繪圖 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 圖表, 行 的圖片

自動產生的描述

Dataset: 90%三個數字的加減運算10%三個數字的加減乘除運算

一張含有 文字, 圖表, 繪圖, 行 的圖片

自動產生的描述一張含有 文字, 行, 圖表, 螢幕擷取畫面 的圖片

自動產生的描述

我所使用的模型是一層RNN的模型，由於RNN模型特別適合處理具有序列性質的任務，而且本次的任務只需要預測三個二位數的加減結果。因此，一層RNN模型便可以很好的勝任這個任務。

1. 我比較了LSTM、RNN、GRU在相同batch\_size、learning\_rate、資料集的狀況下，三者的模型loss差異。

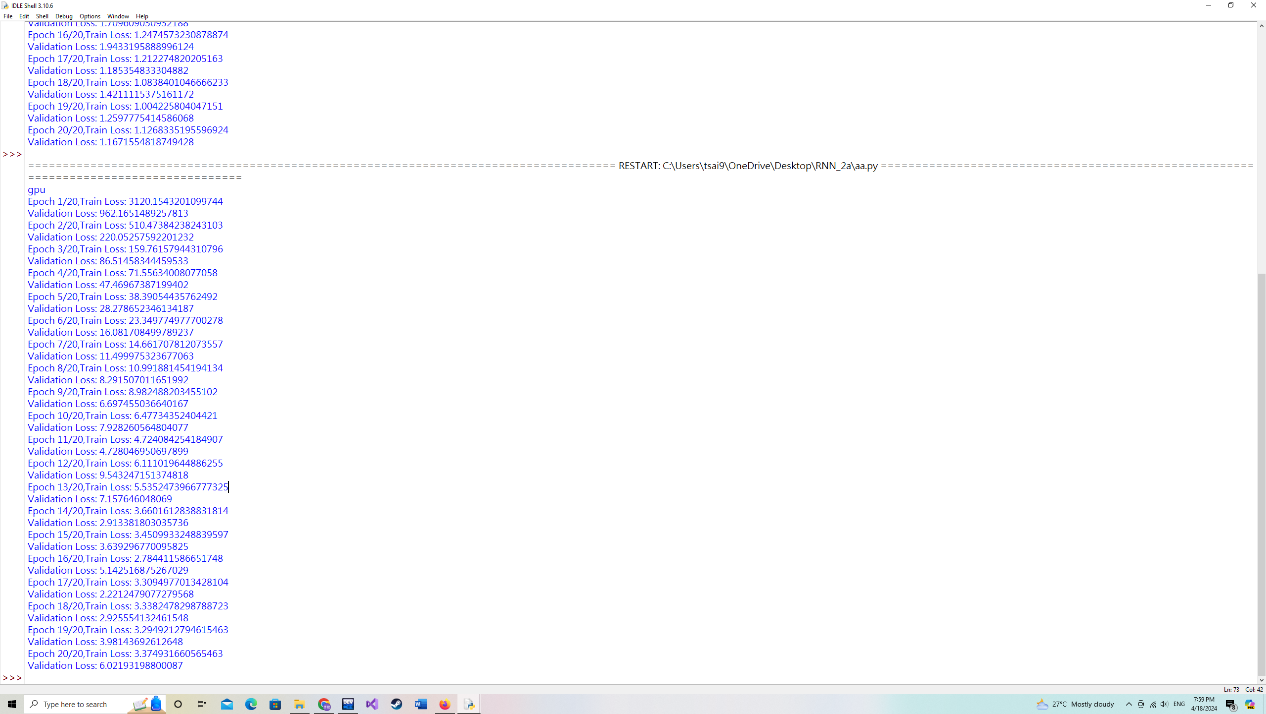
而由於LSTM與GRU具有閘門來控制以往資料的篩選，因此對RNN這種沒有閘門的模型來說，更能夠保留資料的長期記憶，其中LSTM又比GRU多了長期記憶單元，所以在大量的資料長度下，LSTM能夠比GRU有更好的預測結果。

以下是訓練的LOSS與TEST結果

左圖 LSTM\_train\_loss 中圖GRU\_train\_loss 右圖RNN\_train\_loss

一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

自動產生的描述 

我們能夠發現LSTM與GRU的loss結果差不多，都為1左右，這是因為資料長度沒有太長的原因，而RNN的效果較差，並且在training時會有loss抖動的問題

左圖 LSTM\_test\_loss 中圖GRU\_ test \_loss 右圖RNN\_ test \_loss

一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

自動產生的描述 一張含有 文字, 螢幕擷取畫面, 陳列, 軟體 的圖片

自動產生的描述

而在測試時，我們能夠發現LSTM與GRU的結果差不多，而RNN的效果較差