**GAI Project 2.a Arithmetic text generation**

**F74101254 資訊系 張暐俊**

* **Analysis**

1. **Model analysis**

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述

在模型的設計上，採用跟助教撰寫的程式碼相同的架構，一層的embedding層，兩層的RNN層以及一層output層，在實作上使用的是以LSTM這個模型為主。  
loss rate的部分則是會隨著超參數設定的不同而有細微的變動，但總結來說會在epoch = 30左右的時候來到最低數值，隨著epoch持續加大，他的loss rate則會上下起伏不定，再慢慢降回去。

一張含有 文字, 繪圖, 行, 圖表 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述  
上圖訓練結果之超參數，其資料數量為512,000筆一張含有 文字, 繪圖, 行, 螢幕擷取畫面 的圖片

自動產生的描述

加大到epoch = 128的loss rate結果圖，其超參數除epochs外皆相同

1. **Dataset analysis**

我總共生成了四份dataset

* 1. **filtered\_plus\_minus.csv**

取出含3個2位數字之加減運算  
一張含有 文字, 螢幕擷取畫面, 字型, 設計 的圖片

自動產生的描述

* 1. **filtered\_PlusAndMinus\_random.csv**  
     取值為集中值，為10到25以內各個數的加減運算  
     一張含有 多媒體軟體, 軟體, 文字, 繪圖軟體 的圖片

     自動產生的描述
  2. **filtered\_PlusAndMinus\_extreme.csv**  
     取極端值，10到17或41到49間各個數字的加減運算  
     一張含有 多媒體軟體, 軟體, 文字, 繪圖軟體 的圖片

     自動產生的描述
  3. **filtered\_PlusAndMinus\_out.csv**  
     取34到49間各個數字的加減運算  
     一張含有 多媒體軟體, 軟體, 繪圖軟體, 螢幕擷取畫面 的圖片

     自動產生的描述

A是我用來測試超參數的影響以及我所能訓練出的最高分數為多少，B、C、D則是有一定規則取出的dataset，資料皆根據規則抓出後的總資料量隨機提取30,000筆。另外，評估的資料特意使用了Ｂ資料的前3000筆，這麼做的原因是想測驗說，如果根據驗算測試的題目著重去測驗的話，答對率是否會比較高？結果如下附表

|  |  |  |  |
| --- | --- | --- | --- |
| **dataset** | **B (random)** | **C (extreme)** | **D (out)** |
| **Accuracy** | 75.91% | 14.77% | 0.33% |

根據上表的結果我們可以發現，B dataset有著較高的答對率，而完全沒有學習到相關驗證題目的D dataset則是幾乎都無法答對題目，而C dataset則是因為有加減學習到一些資料，所以相對提升了一些，但也因為資料量大多不服驗證題目內容，導致打對率相對慘淡。  
一張含有 文字, 功能表, 螢幕擷取畫面, 圖書 的圖片

自動產生的描述  
B dataset的驗證結果圖，可以發現運算式的數字皆介於10到25之間。  
一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述  
運算B、C、D時，使用之超參數，使用模型皆為LSTM。

1. **Discussion   
   different learning rate**:  
   一張含有 文字, 字型, 螢幕擷取畫面 的圖片

   自動產生的描述

|  |  |  |  |
| --- | --- | --- | --- |
| **Learning rate** | **0.01** | **0.001** | **0.0001** |
| **Accuracy** | 33.98% | 67.49% | 66.71% |

由上述表格可知，learning rate 越小越好，但是也有一極限值，不需要一昧地調小，當到一數值以下時，其對訓練訓能的影響則逐漸變小。

**different batch size:**  
一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述

|  |  |  |  |
| --- | --- | --- | --- |
| **Batch size** | **64** | **256** | **512** |
| **Accuracy** | 67.48% | 66.06% | 66.71% |

在未實驗前，因緣際會的查到IKMLab的github，裡面有助教們整理的上課教材，因為自身對Pytorch不是很熟悉的原因，就去拜讀了一下，其中，在教材的最下方練習，推薦可以加大batch size或許會有更好的學習成果，因此我先入為主的認為只要一直加大batch size，就能有更好的學習成果，結果實驗數據打了我的臉。在查閱一些資料後才明白，原來batch size的大小更多的是影響training的速度，但不一定會使得model有更好的學習成果，還是要配合著模型的各個超參數。

**characteristic of my model:**我使用的模型如上所述，是以LSTM為主。選擇此模型的原因是因為，相較於RNN來說，他擁有更好的效能。此外，forget gate的存在也讓這個模型有更好的學習能力與更新資料能力，而且此模型本身就是用來訓練長期記憶的模型，也符合我們希望他記取更多運算式的功能。

* **Bonus**

1. **Compare the performance of multiple models and provide a brief analysis.**測驗時使用之超參數  
   一張含有 文字, 螢幕擷取畫面 的圖片

   自動產生的描述

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **RNN** | **LSTM** | **GRU** |
| **Accuracy** | 67.25% | 66.39% | 65.75% |

各模型Losing rate 示意圖一張含有 文字, 繪圖, 行, 圖表 的圖片

自動產生的描述一張含有 文字, 繪圖, 行, 圖表 的圖片

自動產生的描述一張含有 文字, 繪圖, 行, 圖表 的圖片

自動產生的描述

在我原本的想法中，我覺得RNN模型因為其架構較為簡單，所以應該其學習成果應該較差，但測驗結果跟我想的完全相反，於是不信邪的我，決定將batch size調小後，再嘗試一遍，因為我覺得說不定是因為一次訓練的數量比較少，所以RNN會有優勢。 **一張含有 文字, 字型, 螢幕擷取畫面 的圖片

自動產生的描述**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **RNN** | **LSTM** | **GRU** |
| **Accuracy** | 67.47% | 67.49% | 67.49% |

各模型Losing rate 示意圖一張含有 文字, 行, 繪圖, 圖表 的圖片

自動產生的描述一張含有 文字, 繪圖, 行, 圖表 的圖片

自動產生的描述一張含有 文字, 繪圖, 行, 圖表 的圖片

自動產生的描述

結果竟然實測結果一模一樣，後來去查了資料才知道，其實RNN模型最大的缺陷是梯度上的問題，我更改batch size說實話應該不太能得到我想要的結果。但其實仔細去看還是有差別的，關鍵就在losing rate的圖片裡，可以發現在第40個epoch時，loss的大小排序依據是RNN > LSTM > GRU，可以側面推測出，其實GRU模型的學習效果應該是比較好的沒錯，另外，在LSTM的圖片也可以發現其下降穩定，相較於其他兩者比較不會出現忽大忽小的狀況。