Pattern Recognition Final Project

Writer Recognition

卢巧渝

项芳琪

蔡宗羲

目錄

1 任務摘要

1.1 任務簡介

手寫漢字識別（HCCR）是模式識別的重要領域，在手寫文字輸入裝置、手稿文書光學字元識別等任務中有著廣泛的應用前景。

在本次任務中，我們聚焦於用深度神經網路處理 HCCR 中書寫者鑒別的任務，根據手寫的單個漢字識別出它們的書寫者。我們將完整地完成資料獲取、處理、分類識別、分析報告整 個過程。

1.2 具體任務

每位同學通過線上錄入工具採集 500 個常用漢字做為為資料，其中，將 300 個漢字作為訓練集，100 漢字作為驗證集，這部分資料由助教在網路學堂上公佈，而最後 100 個漢字為不公開的測試集，用於最後測評最終代碼所使用。要求使用深度神經網路作為分類模型，可 以採用多種模型，其中必須有一個基於 RNN 的模型，用 LSTM 或 GRU 均可，並利用所搭 建的神經網路模型來識別出書寫者

1.2.1 任務 1

由助教挑選出字跡差異明顯的 10 位元同學的資料，完成 10 類別的書寫者分類任務.

1.2.2 任務 2

使用大部分同學的資料，完成 107 類別的書寫者分類任務.

1.3 開發工具

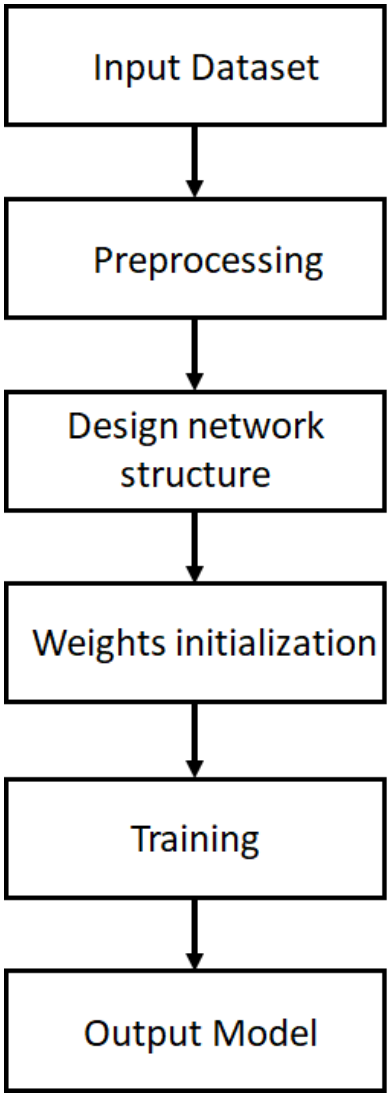
應用平台：linux 系統

開發工具： jupyter notebook、pycharm

開發語言：python3

2.具體設計

2.1 設計結構



2.2 數據預處理

一般來說，對於原始資料，可能因為每個人的採集狀況不同，或是在採集過程中紀錄了一些躁聲點、異常值等，又或是有缺失值的狀況，而這些情況都將不利於模型的訓練，一般原始資料存在的問題如下所示：

- 含噪聲：數據中有明顯錯誤、異常偏離期望值的數據
- 不完整：數據少部分的缺失

因此，如行對原始資料進行整理，提供給模型訓練，是在學習領域中很重要的一環，以下介紹在這次的書寫者識別任務中，如何對資料進行預處理

在書寫者識別任務當中，資料的格式如下：

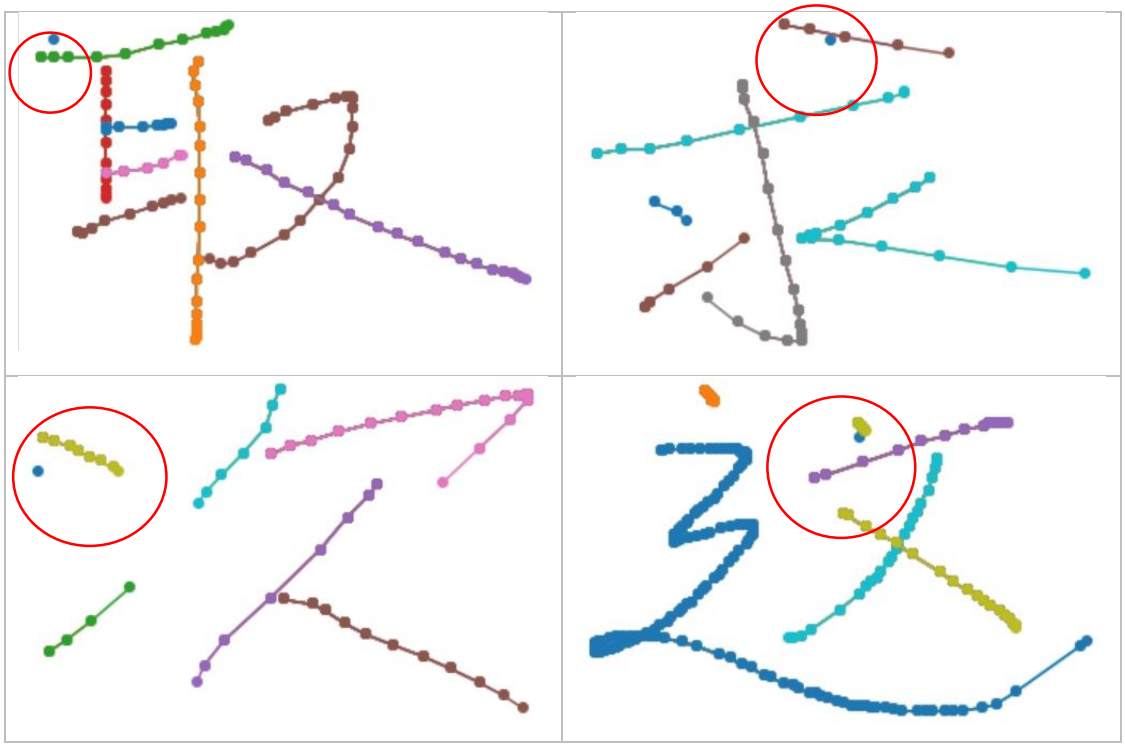
- 以每一位同學為一個書寫者類別
- 每一類別(每一位同學)，有 500 個樣本(500 個漢字)，其中 300 個樣本作為訓練 集，100 個樣本作為驗證集，100 個樣本作為最終的測試集
- 對於每一個樣本(每一個漢字)，紀錄時每次筆面接觸紙面到抬起為一個筆劃，每 一個筆劃用一個書寫過程中經歷的一系列二為座標點數表示：

$$stroke = \left[[x_1 , y_1] , [x_2 , y_2] , [x_3 , y_3] \dots\dots [x_n , y_n] \right]$$

- 每個漢字的數據以筆畫組成的組數表示：

$$character = [stroke_1 , stroke_2 , stroke_3 \dots\dots stroke_k]$$

2.2.1 去除只含有一的點的筆劃：



如上表格內圈起来的方所示，如果該筆畫只有一個點($stroke_n = [[x_1, y_1]]$)，可以將其視為採樣時不小心接觸到紙面而被記錄下來的異常值，予以去除，即，將 $len(stroke_n) = 1$ 的筆畫去除

2.2.2　去除筆畫中的異常值

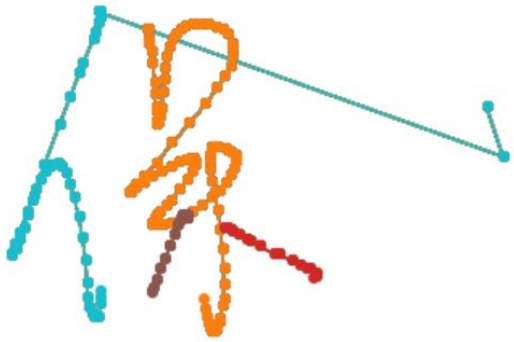
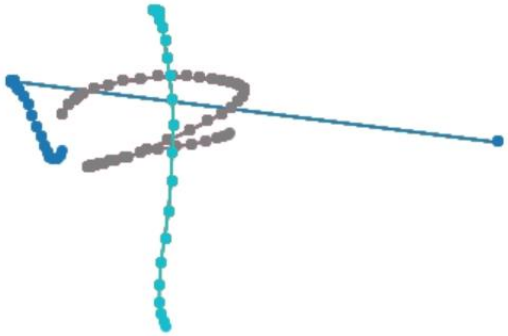
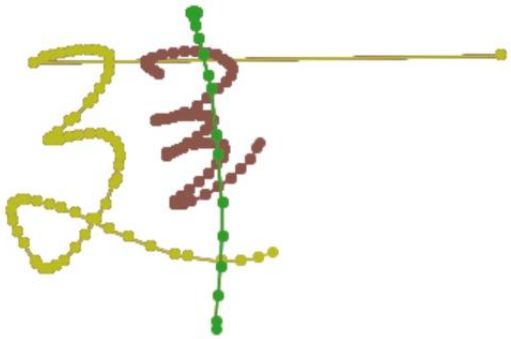
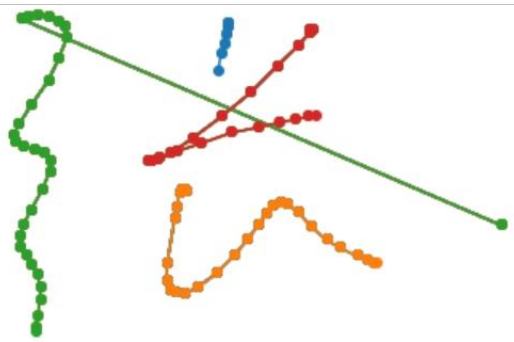
如果在該筆劃內，有一點與其他點的距離異常的遠，則將該點視為異常值，需要將此筆畫內的該點去除

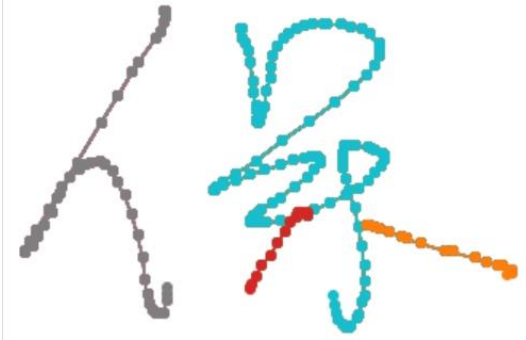
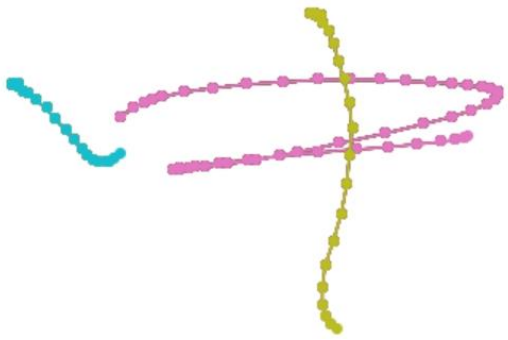
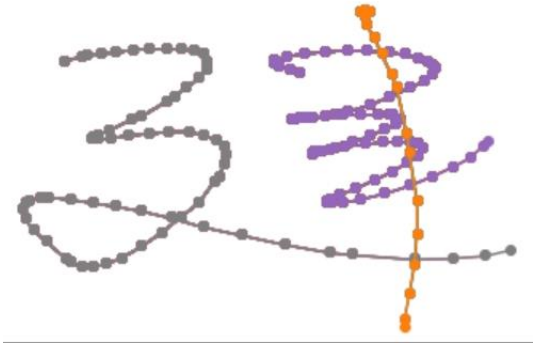
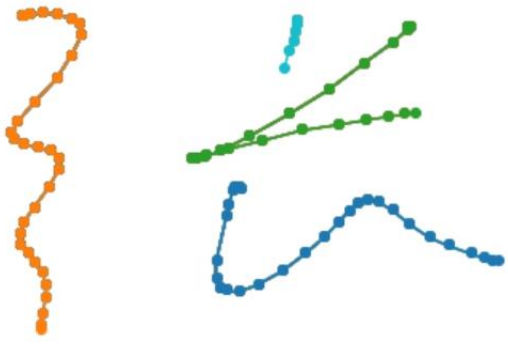
判斷方式：

對於一般常態分佈，在 3 倍標準偏差的原則下，異常值為與平均值偏差超過三倍標準差的值，機率為:

$$P(|x - \mu| > 3\sigma) \leq 0.003$$

考慮到該筆畫下所有點的分布不是常態分佈，且漢字中有些筆畫的方差較大，如”辶”字旁，因此這裡取$(\mu \pm 4\sigma)$作為判斷標準，處理前後如下表格所示

刪除前	
	
	

刪除後	
	
	

2.2.3　數據預處理

與一般圖像資料不同的是，線上漢字採集紀錄了每一筆化的先後順序，若妥善的運用每　一次的起筆、落筆、筆化的先後等資料，將更有利於模型的訓練，這裡參考文獻裡的作法， 將筆與紙面接觸表示為(1)，筆與紙面離開時表示為(-1)，因此每個 stroke 可以表示為：

$$\mathrm{s} = \left[[x_1\,, y_1\,, 1\,]\,, [x_2\,, y_2\,, 1] \ldots\ldots [x_n\,, y_n\,, -1]\right]$$

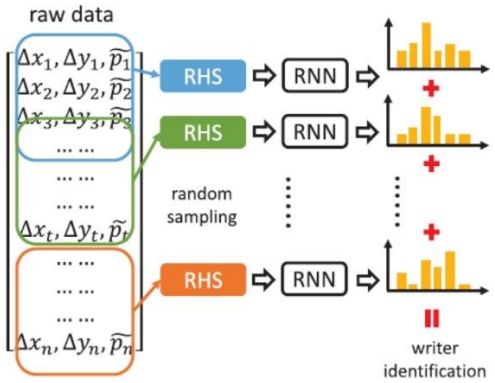
再進一步利用 s 將書寫時的軌跡表示出来，筆跡點的移動距離為

$$(x_{i+1} - x_i)\,, (y_{i+1} - y_i)\,, (p_{i+1} \times p_i)$$

其中$(x_{i+1} - x_i)\,, (y_{i+1} - y_i)$表示了筆跡在二維座標上的位移，而$(p_{i+1} \times p_i)$則可以區分不同的筆畫，每一段筆畫完整位移的前後分別是(-1)，中間位移部分為(1)，因此可以將一個字表示為：

$$\Delta s = [\,[\Delta x_1\,, \Delta y_1\,, \Delta p_1]\,, [\Delta x_2\,, \Delta y_2\,, \Delta p_2]\,, \ldots\ldots\ldots]$$

而為了有足夠多的訓練樣本，將每一位元同學的所有筆跡表示成一個Δs序列，從中擷取一段一段的 RHS，藉以獲得足夠的訓練樣本，如下圖所示：

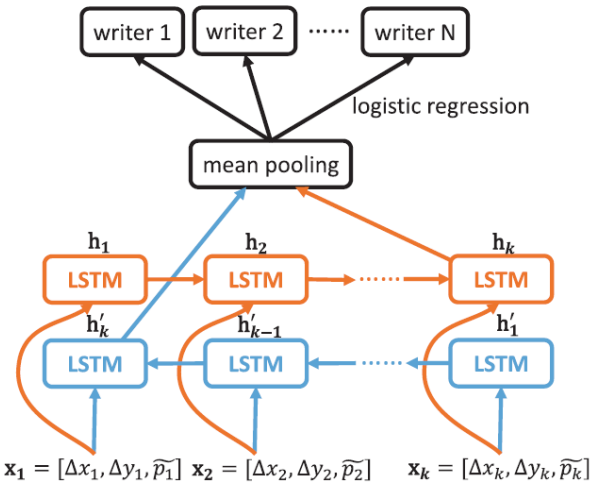


2.3 神經網路模型

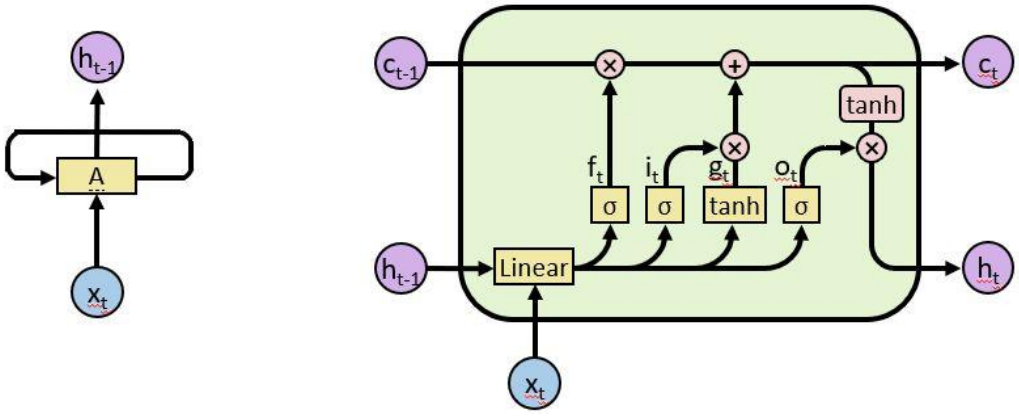
本次採用兩種神經網路模型 lstm 與 gru 模型，分別介紹如下

2.3.1 LSTM

LSTM（Long Short-Term Memory）是長短期記憶網路，是一種時間迴圈神經網路，適 合於處理和預測時間序列中間隔和延遲相對較長的重要事件。本次任務按照文獻方法，採用 雙向 lstm 模型，將預處理得到的 RHS 分別以順向與反向的方式作為輸入資料，訓練得到隱 藏層節點 hk 與 hk，將兩者相加平均後再輸入到全連接層得到輸出值，並利用交叉熵作為損 失函數得到 loss，再用 adam 演算法優化反向傳播更新參數，如下圖所示:



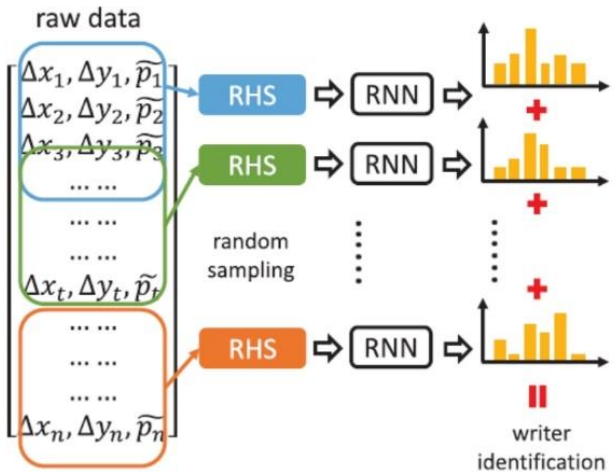
單個 LSTM 結構如下所示:



在第 t 次輸入時，輸入門、遺忘門、輸出門如下：

$$
\begin{aligned}
\mathbf{i}_t &= \text{sigm}\left(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + b_i\right) \\
\mathbf{f}_t &= \text{sigm}\left(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + b_f\right) \\
\mathbf{o}_t &= \text{sigm}\left(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + b_o\right) \\
\widetilde{\mathbf{c}_t} &= \tanh\left(W_c \mathbf{x}_t + U_c \mathbf{h}_{t-1} + b_c\right) \\
\mathbf{c}_t &= \mathbf{i}_t \odot \widetilde{\mathbf{c}_t} + \mathbf{f}_t \odot \mathbf{c}_{t-1} \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}
$$

將訓練集按上述方式預處理後經由網路預測得到單個 RHS 的分類結果，並將分類結 果做投票，投票機制為將數目最多的類別作為預測結果，如下圖所示：



Lstm 包含參數模型如下:

Layer:雙向 LSTM 的網路層數

Input_size:輸入數據的特徵維度

Hidden_size：隱藏節點的特徵維度

Batch_size：每次訓練的 batch 的個數

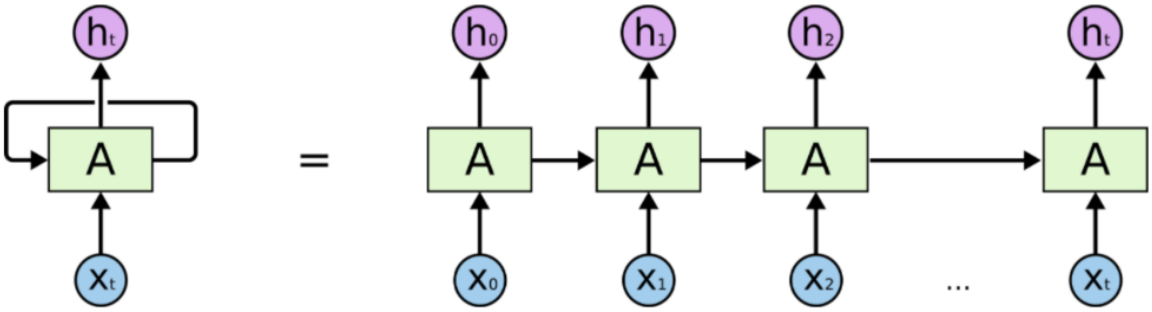
H0 及 c0 初始值為服從標準正態分布的隨機數

損失函數為交叉熵；優化方法為 Adam 方法，學習率為 0.01

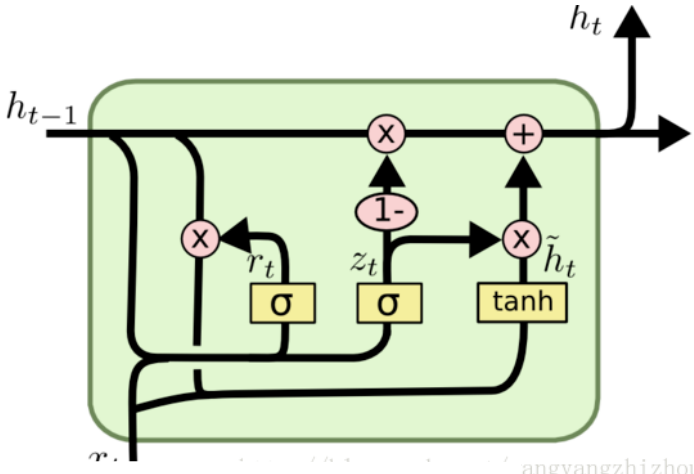
2.3.2 GRU

GRU 是 LSTM 網路的一種效果很好的變體，它較 LSTM 網路的結構更加簡單，而且效果也很好，因此也是當前非常流形的一種網路。GRU 也是可以解決 RNN 網路中的長依賴 問題，因此本次任務也採用了 GRU 模型來進行訓練並於 LSTM 做比較分析。

在此採用的 GRU 模型為單向，因此與上述雙向 LSTM 不同的是，網路並沒有反向信息，因此不需要做 mean pooling 的操作，直接輸入到全連接層得到輸出值，接下來同 LSTM 利用交叉熵作為損失函數得到 loss，再用 adam 演算法優化反向傳播更新參數，如下圖所示：



相較於在 LSTM 中引入了三個門函數：輸入門、遺忘門和輸出門來控制輸入值、記憶值和輸出值，在 GRU 模型中只有兩個門：分別是更新門和重置門。每個 GRU 單元 A 的具 體結構如下圖所示：



網路的前向傳播公式如下:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W_{\tilde{h}} \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

$$y_t = \sigma(W_o \cdot h_t)$$

LSTM 和 CRU 都是通過各種門函數來將重要特徵保留下來，這樣就保證了在 long-term 傳播的時候也不會丟失。此外 GRU 相對於 LSTM 少了一個門函數，因此在參數的數量 上也是要少於 LSTM 的，所以整體上 GRU 的訓練速度要快於 LSTM 的。

預測方式同 LSTM，將訓練集按上述方式預處理後經由網路預測得到單個 RHS 的分類結果，並將分類結果做投票，投票機制為將數目最多的類別作為預測結果。

GRU 包含參數模型如下:

Layer:GRU 的網路層數

Input_size:輸入數據的特徵維度

Hidden_size：隱藏節點的特徵維度

Batch_size：每次訓練的 batch 的個數

損失函數為交叉熵；優化方法為 Adam 方法，學習率為 0.01

3.實驗結果

3.1 基於雙向 LSTM 的 RNN 模型實驗

3.1.1 十分類實驗

首先，對提供的十分類資料進行了實驗，Train 檔作為訓練資料，Validation 檔作為測 試資料。因為十分類只是測試模型的可用性，所以這裡簡單對兩種 RHS 的序列長度 50 和 100 進行了對比。關鍵參數設置如表 3.1-1。

表 3.1-1 十分類關鍵參數

數據： 訓練數據每人所取 RHS 數目：3000 測試數據每人所取 RHS 數目：1000
輸入： 單個 RHS 序列長度：sequence 特徵維度：3
模型： LSTM 層數：2 隱藏節點數：300 雙向：True batch 大小：200 batch_first：True
其它： 訓練 epoch 數：20

訓練時的 loss 變化如圖 3.1-1 所示。兩種 sequence 長度下，loss 的下降趨勢幾乎重合，可 能類別數較少，sequence 達到 50 時已經能很好地學習到主要特徵，sequence 繼續增加，學習 的特徵數已經趨於飽和而不會給 loss 帶來明顯的變化。

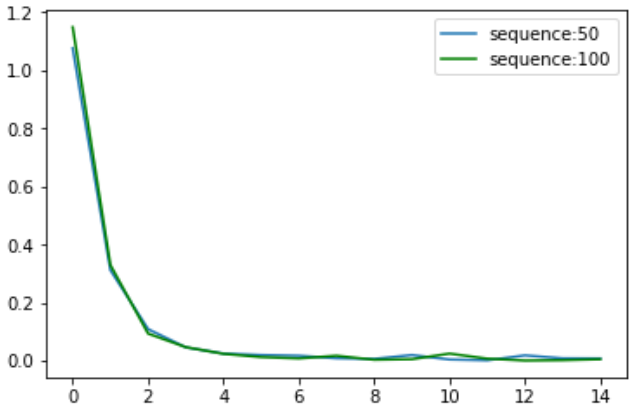


图 3.1-1 十分類 loss 歷史數據

某次實驗時，兩種序列長度下的訓練記錄以及測試情況如表 3.1-2 所示。可以發現，序列長度增加一倍後，訓練時間幾乎增長到八倍。雖然 sequence=50 時的單個 RHS 測試準確率略低於 sequence=100 的結果，但也超過了 95%，並且兩種情況下的投票準確率都達到了 100%。此外，兩種情況的平均 loss 幾乎一致，約為 0.11。因此在十分類情況下，綜合考慮資 源消耗和準確率，sequence 取為 50 更為合適。我們也比較了 sequence=100 時，hidden size（隱藏節點數）增加到 500 的情況，訓練時間會相應增加到 5830.22s，但是單個 RHS 的測試準確率並沒有進一步上升，仍為 98%。可見，對於十分類，表 3.1-1 中模型的參數已經能很好完成任務。

表 3.1-2 十分類實驗結果

序列長度	訓練時間（s）	訓練 loss	單個 RHS 測試準確率（100%）	投票測試準確率（100%）
50	722.69	0.1136	96.00	100.00
100	5647.02	0.1170	98.00	100.00

3.1.2 百分類實驗

在十分類任務中確定模型的可用性後，我們使用提供的 107 類書寫者的資料進行實驗。在保證模型在一定時間內能收斂的前提下，這裡分別對 sequence 和 hidden size 進行了比較，其中 sequence=100、hidden size=256 作為 base line，hidden size 增加時 LSTM 層數選擇 1 否則為 2。關鍵參數設置如表 3.1-3 所示。

表 3.1-3 百分類關鍵參數

數據： 訓練資料每人所取 RHS 數目：3000 測試資料每人所取 RHS 數目：1000
輸入： 單個 RHS 序列長度：sequence 特徵維度：3
模型： LSTM 層數：layer 隱藏節點數：hidden size 雙向：True batch 大小：200 batch_first：True
其它： 訓練 epoch 數：5

（1）改變單個 RHS 的 sequence 長度

Hidden size 固定為 256，對應 layer=2，sequence 分別取 50、100、150 時，訓練中 loss 的歷史資料如圖 3.1-2 所示，整個實驗結果如表 3.1-4 所示。

由圖 3.1-2 可知，sequence=100 時，起始的 loss 最低，sequence 為 150 時排第二，但下降速度最快。sequence=50 時，loss 始終比另外兩種情況大；而另外兩種情況的 loss 曲線在後期相差不大。此外，三種 sequence 情況下，loss 的收斂還不夠理想，在尾端仍有震盪，受資源時間所限，沒有增加 epoch 的數目進一步觀察後續結果。也許調整參數可以使得 loss 曲線在 5 個 epoch 下也能很光滑。

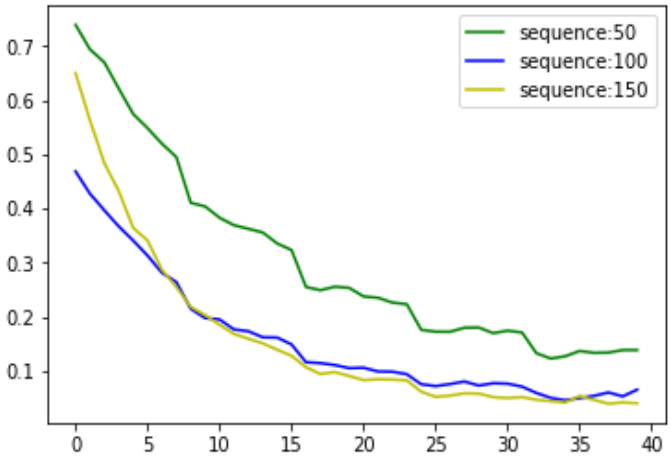


图 3.1-2 百分類不同 sequence 時的 loss 歷史數據

由表 3.1-4 所知，儘管 epoch 只取到 5，sequence 為 50 時，訓練時間已經達到了 7912s，同時跟 sequence 不是正比例的倍數關係，這跟網路的結構有關係。收斂時 sequence=100 時的 loss 最低為 0.1542，sequence=150 緊隨其後，這與圖 3.1-2 的直觀感覺是一致的；而 sequence=50 時的 loss 增大了一倍。測試單個 RHS 時，在本實驗三種情況下，sequence 越長， 準確率越高。無論 sequence 長度為多少，最後的投票測試率都能達到 100%。綜合考慮下， sequence 選為 100 更為合適。

表 3.1-4 百分類改變 sequence 的實驗結果

序列長度	訓練時間（s）	訓練 loss	單個 RHS 測試準確率（100%）	投票測試準確率（100%）
50	7912.26	0.3047	86.00	100.00
100	8378.23	0.1542	91.00	100.00
150	8686.45	0.1560	93.00	100.00

（2）改變 hidden size

sequence 固定為 100，hidden size 分別取 256、500 (layer=1)、800 (layer=1)時，訓練中 loss 的歷史資料如圖 3.1-3 所示，整個實驗結果如表 3.1-5 所示。

由圖 3.1-3 所示，hidden size 增加到 500 和 800 時，loss 曲線比圖 3.1-2 中曲線更平滑，從這個角度來看，參數仍然有調整的空間。Hidden size 增加後，起始 loss 是 base line 的 8 倍， 但是下降趨勢也很明顯。三種情況收斂時的 loss 基本一致。說明 base line 的網路參數已經能 使 loss 達到很低的水準，再變換其他組合，也難以達到更低的 loss。事實上，收斂時很多次 batch 的 loss 已經接近 0；而考慮到過擬合情況，訓練 loss 也不是越小越好，而是有一個合適 的取值。

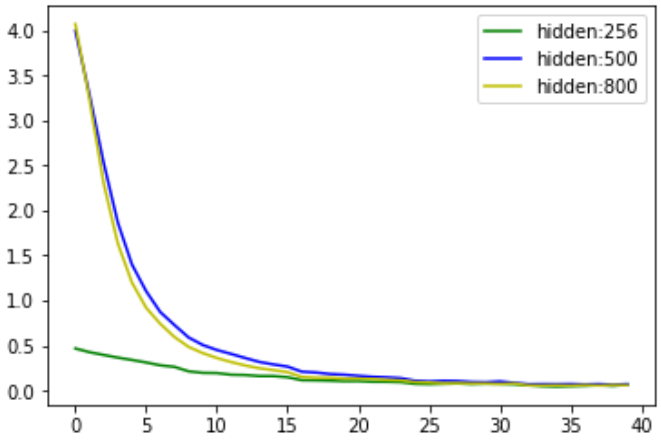


图 3.1-3 百分類不同 hidden size 時的 loss 歷史數據

由表 3.1-5 可知，hidden size 增加到 500 和 800，同時網路 layer 從 2 降到 1 時，訓練收斂時的 loss 是 baseline 的三倍多，但是就單個 RHS 測試而言，三種情況結果幾乎一致，hidden size=800 時甚至高出 1%。這說明 baseline 模型可能會有輕微過擬合。經投票統計，準確率都能達到 100%。

表 3.1-5 百分類改變 hidden size 的實驗結果

隱藏節點	訓練時間（s）	訓練 loss	單個 RHS 測試準確率（100%）	投票測試準確率（100%）
256	8378.23	0.1542	91.00	100.00
500	12088.57	0.5424	91.00	100.00
800	13140.54	0.4854	92.00	100.00

3.2 基於 GRU 的 RNN 模型

基於 GRU 的模型直接對 107 分類進行實驗，我們對比了單個 RHS 的 sequence 取三種長度的情況。模型的關鍵參數如表 3.2-1 所示。

表 3.2-1 基於GRU 的 RNN 模型參數

數據： 訓練數據每人所取 RHS 數目：3000 測試數據每人所取 RHS 數目：1000
輸入： 單個 RHS 序列長度：sequence 特徵維度：3

模型： GRU 層數：1 隱藏節點數：256 batch 大小：200 batch_first：True
其它： 訓練 epoch 數：10

單個 RHS 的 sequence 長度分別取 50、100、150，實驗結果如表 3.2-2 所示。首先，GRU 的訓練時間相對比較短。訓練收斂時的 loss 都大於 1，sequence 長度取 150 時，達到最低值 1.280，這可能是因為 epoch 比較小，收斂還不夠。可以觀察到，單個 RHS 的測試準確率最高只有 57%，但引入投票機制後，準確率最低也能到 90%，可見投票機制的有效性。

表 3.2-2 基於 GRU 的 RNN 實驗結果

序列長度	訓練時間（s）	訓練 loss	單個 RHS 測試準確率	投票測試（正確數/總數）
50	163.380	1.995	0.39	102/107
100	353.14	2.449	0.33	98/107
150	502.114	1.280	0.57	105/107

3.3 總結與分析

3.3.1 GRU 與 LSTM

由於 GRU 比 LSTM 少了一個門的設定，並且沒有雙向機制，在其它參數一致時，即使前者 epoch 數目多出一倍，GRU 的訓練時間仍然比 LSTM 減少很多。在我們的實驗環境中，GRU 在本實驗資料集上的表現不如 LSTM，這可能跟模型複雜度有關：LSTM 的複雜度較高，因此能學到更多的區別性特徵，從而分類性能更好。不管是 GRU 還是 LSTM，單個 RHS 測試準確率都有一定的上限，而引入投票機制後，準確率幾乎可以接近 100%，說明一系列隨機取出的 RHS 中能涵蓋盡可能多的書寫者書寫模式，並且具有一定效力的模型都能抓取幾乎全部的特徵，從而當測試樣例量大時，大部分樣例所包含的書寫特徵總能是模型已經明確學到的，通過投票後，就能成功鎖定真正的書寫者。

4.任務分工

卢巧渝：神經網路架構的搭建及實驗

项芳琪：神經網路架構的搭建及實驗

蔡宗羲：數據預處理及實驗神經網路

參考文獻

[1] 金莲文，钟卓耀，杨钊，等. 深度学习在手写汉字识别中的应用综述[J]. 自动化学报，2016, 42(8): 1125-1141

[2] Zhang X Y, Xie G S, Liu C L, et al End-to-end online writer identification with recurrent neural network[J]. IEEE Transactions on Human-Machine Systems, 2017, 47(2): 285-292