Deep Learning HW3

309512010 陳紀愷

1. Recurrent Neural Network

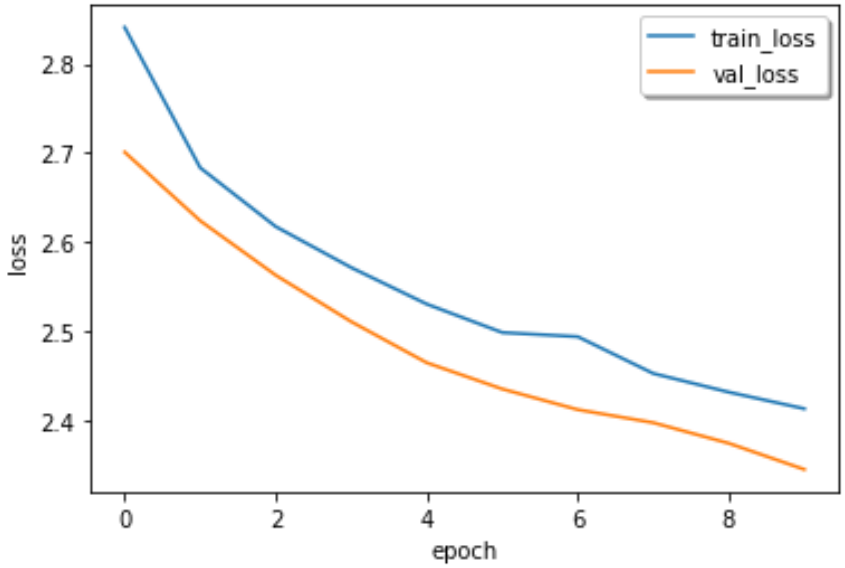
1. Construct a standard RNN then show your (1) network architecture, (2) learning curve, (3) training error rate and (4) validation error rate.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
simple_rnn_21 (SimpleRNN)    (None, 125)               15875
_________________________________________________________________
dropout_70 (Dropout)         (None, 125)               0
_________________________________________________________________
dense_95 (Dense)             (None, 100)               12600
_________________________________________________________________
dense_96 (Dense)             (None, 65)                6565
=================================================================
Total params: 35,040
Trainable params: 35,040
Non-trainable params: 0
_________________________________________________________________
```

為了預測文本，我使用了一層的 Simple RNN 和兩層的 dense 層，其中 Input 的長度為 20，裡面有 125 的 hidden node，並在 LSTM 後加入一層的 Dropout 來避免 Overfitting，整體的架構是利用前 20 個字母去預測下一個 Output，最後 dense 層後輸出 65 個不同字元的預測機率，並在最後預測時會用來加入一點隨機的因素來預測，利用像是輪盤的方式取代直接 Arg max，讓按照預測出機率分布來輪盤選擇下一個字元，也就是增加變異性的感覺。

```
def sample(preds, temperature=1.0):
    preds = np.asarray(preds).astype("float64")
    preds = np.log(preds) / temperature
    exp_preds = np.exp(preds)
    preds = exp_preds / np.sum(exp_preds)
    preds = preds.reshape(-1)
    probas = np.random.multinomial(1, preds, 1)
    return np.argmax(probas)
```

(2) learning curve



Training error rate = 2.4126
Validation error rate = 2.34442

2. Choose 5 breakpoints during your training process to show how well your network learns through more epochs. Feed some part of your training text into RNN and show the text output.

Epoch2

```
Epoch 1/1
780768/780768 [==============================] - 121s 154us/step - loss: 2.6834
334616/334616 [==============================] - 139s 415us/step

Generating text after epoch: 2
Seed:              JULIET
Generated:  :
Whirse toer thin mail taent, Fhw ii vho fottea toar tho aoereob soen bi thee  aaet a wor drne aag oeeieeer, Bnd mared tiu cathr
s aas poc fante fneu my tou,
Wot wouh whu siy oo bed aanehen hn pre has
```

Epoch4

```
Epoch 1/1
780768/780768 [==============================] - 121s 156us/step - loss: 2.5712
334616/334616 [==============================] - 141s 422us/step

Generated:  :
Whet wee tear su tha salto,
Gom taet tot la thes niecf the sase do whu oasee toale in taa wouur po mh the aasidr,
pot mi no monseplg toon tha iomd the piod oo shle bar ea thereee oo hanne tou shreee
```

Epoch6

```
Epoch 1/1
780768/780768 [==============================] - 121s 155us/step - loss: 2.4983
334616/334616 [==============================] - 141s 420us/step
```

```
Generating text after epoch: 6
Seed:              JULIET
Generated:  :
You ha hat fta tu oef uo tee nardot bno the gale to eoa mne sasd!po thf har, toon'w frt c dareend fe ankw Hs nen pe mo shin me
houh tooen fnd wid prbre wou fid thi mast as aeohdrc the kae to bie the
```

Epoch8

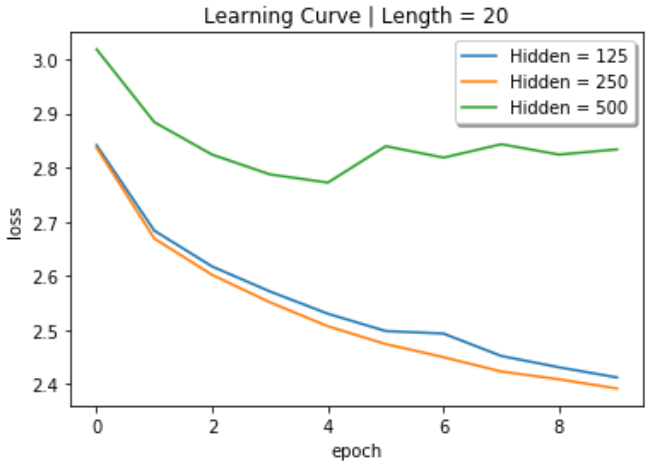
```
Epoch 1/1
780768/780768 [==============================] - 121s 155us/step - loss: 2.4522
334616/334616 [==============================] - 142s 424us/step

Generating text after epoch: 8
Seed:               JULIET
Generated:  :
Whec hov yeot thete fote toog thes the siter she rhe the thnu wha thes the brart tha nimb hy with ofti,
ahan htn she mav thee hat oa dy cemtr shet hawe the teie;
Thet mor the cldreet cnd the faue th
```

Epoch10

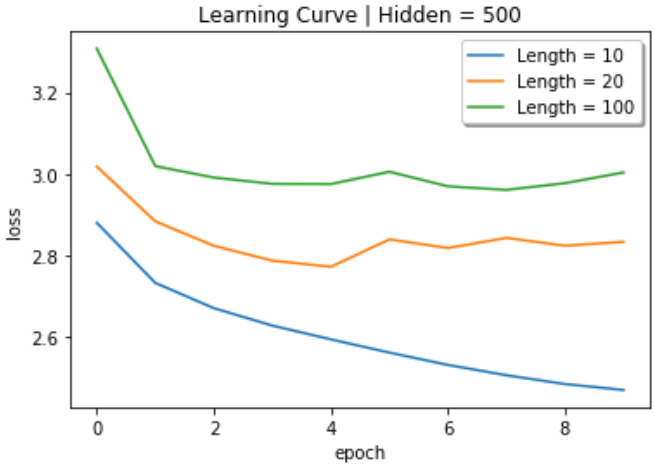
```
Epoch 1/1
780768/780768 [==============================] - 121s 155us/step - loss: 2.4127
334616/334616 [==============================] - 139s 415us/step

Generating text after epoch: 10
Seed:              JULIET
Generated:  :
What shatehs thei the boras lave mo the oate homd moa siou that io the hand, Io nake, thetee aod O wour ho totee you nart ae wo
u food of baalt,
wherk heve ie thee me hor oe thei navesed heee Teruery
```

分析上面五個 breakpoint (epoch2 4 6 8 10)後可以發現，前幾個 epoch 中比較不會出現單字，通常都是隨便把幾個同樣字母拼接起來，像是 `thereee oo hanne tou shreee` 看不太出來語意，但到 epoch8 和 10 的時候，網路已經可以順利的拼出一些句子，隨然看不太初語意，但是看起來比較有單字的概念在預測中，所以還是有進步的。

3. Compare the results of choosing different size of hidden states and sequence length by plotting the training loss vs. different parameters.



上圖是固定在長度=20 的情況下改變 hidden states 的個數，分別比較了 125,250,500，可以發現在固定長度的情況下 125 和 250 表現差不多，但都比 500 明顯好了不少，也就是在 RNN 的架構下 hidden states 的個數應該少一點比較好。



上圖是固定在 hidden states=500 的情況下改變 input 的長度，分別比較了 10,20,100，可以發現在 input 長度越短的情況下，訓練的 loss 也越少，代表在使用 RNN 預測的時候 Input 不應該過長。

4. LSTM

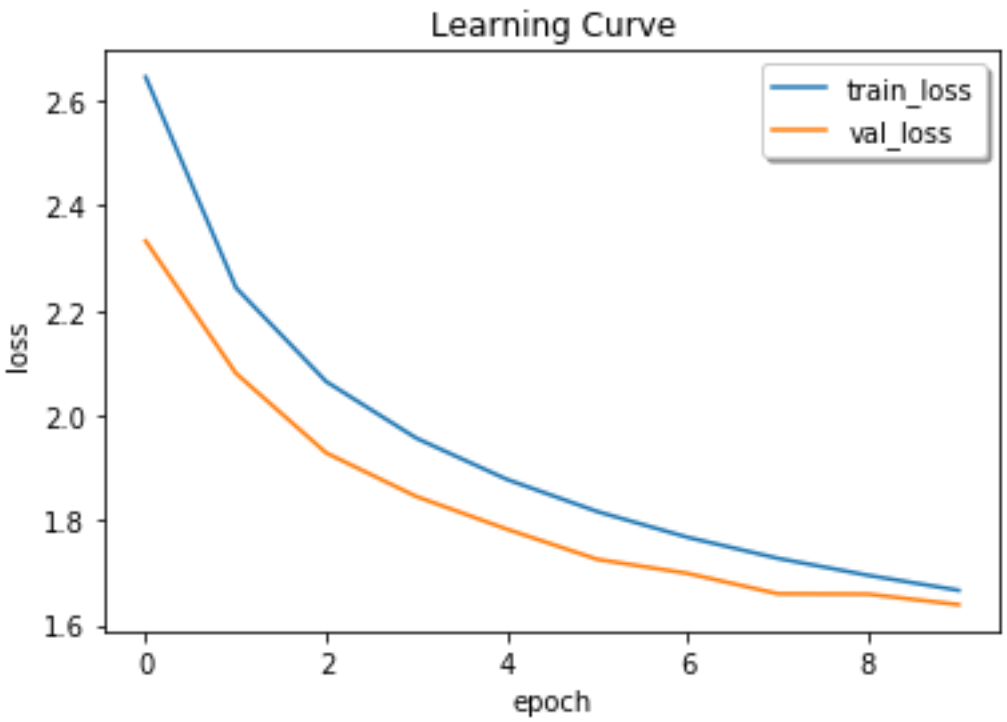
5. (1) network architecture, (2) learning curve, (3) training error rate and (4) validation error rate.

   (1) network architecture,

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_42 (LSTM)               (None, 20, 500)           1004000
_________________________________________________________________
dropout_20 (Dropout)         (None, 20, 500)           0
_________________________________________________________________
lstm_43 (LSTM)               (None, 500)               2002000
_________________________________________________________________
dropout_21 (Dropout)         (None, 500)               0
_________________________________________________________________
dense_39 (Dense)             (None, 100)               50100
_________________________________________________________________
dense_40 (Dense)             (None, 65)                6565
=================================================================
Total params: 3,062,665
Trainable params: 3,062,665
Non-trainable params: 0
_________________________________________________________________
```

為了預測文本，我總共使用了兩層的 LSTM 和雙層的 dense 層，其中 LSTM 的長度為 20，裡面有 500 的 hidden node，並在兩層 LSTM 後加入一層的 Dropout 來避免 Overfitting，最後 dense 層後輸出 65 個不同字元的預測機率，並和第一題一樣在最後預測時會用來加入一點點隨機的因素來預測。

(2) learning curve



Training error rate = 1.66649

Validation error rate = 1.63918

6. Choose 5 breakpoints during your training process to show how well your network learns through more epochs. Feed some part of your training text into LSTM show the text output.

Epoch1

```
780768/780768 [==============================] - 363s 465us/step
334616/334616 [==============================] - 229s 684us/step

  Generating text after epoch: 1
  Seed:                JULIET
  Generated:   :
  Iow, winl thnu thtu seen en the looenn pf the leut
  fe the hiut, Yhe eoru with to aooenee of the dofntan
  and bhoinren fneeet of you so thes ofelenete ho the tiess.

  PARHIUS:
  Meveere, whete ly sealeed
```

Epoch3

```
780768/780768 [==============================] - 361s 462us/step
334616/334616 [==============================] - 228s 681us/step

Generating text after epoch: 3
Seed:               JULIET
Generated:   :
The she iad had fise the iing, a goure with Lenccnrdr of As and ponr all the srnrd and the may as the dace
of she ploe,
And ho a mar of a bolmonane anl have coowente our gor cear in b fair of the pa
```

Epoch5

```
780768/780768 [==============================] - 361s 462us/step
334616/334616 [==============================] - 230s 688us/step

Generating text after epoch: 5
Seed:               JULIET
Generated:   :
Wou mow  prby,
so the tare.

RICINIUS:
Who thou ttinty, and the darg and sefntrinn wet a wartant and she pecfoss eor the fuent of hanfs,
I am a diarrer of my selevhng threat,
and there in my cods,
L
```

Epoch7

```
780768/780768 [==============================] - 364s 466us/step
334616/334616 [==============================] - 226s 677us/step
```

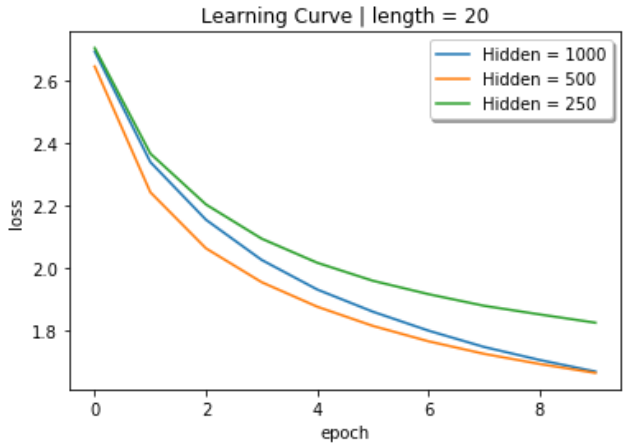
```
Generating text after epoch: 7
Seed:                JULIET
Generated:   :
And fan his letser mine Aaiillo,
I am the motth of her shan for orincesloys sout the was and we she might
And the pardon' A man hn orovd and celivmngss in the fead
And cotrt that the guert of a pfrp
```

Epoch9

```
780768/780768 [==============================] - 376s 481us/step
334616/334616 [==============================] - 333s 994us/step

Generating text after epoch: 9
Seed:                  JULIET
Generated:  RANE:
I shall he not be bach here will the peosle,
To full of arm to she fellows countery.

PETRUCHIO:
That hath the koown tiat the princely duke of my dourtesy, break,
That would I hn the reoatiri
```

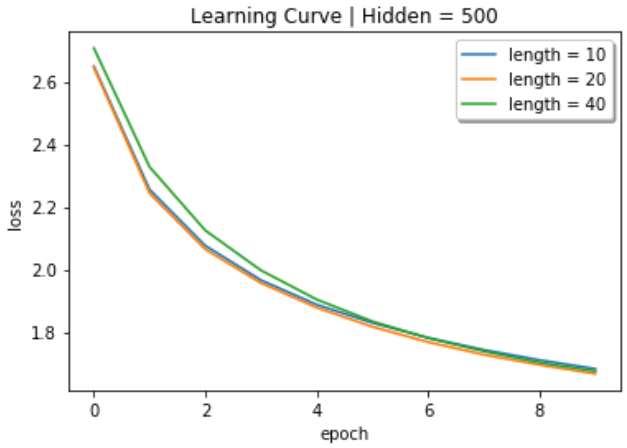
分析上面五個 breakpoint (epoch 1 3 5 7 9)後可以發現，前幾個 epoch 中比較不會出現單字，通常都是隨便把幾個字母拼接起來，但到 epoch7 和 9 的時候，網路已經可以順利的拼出一些單字、像是 might、break、shall，甚至還會有語法的存在(I shall、To she…)，可以發現確實有在成功學習文本。

7. Compare the results of choosing different size of hidden states and sequence length by plotting the training loss vs. different parameters.



上圖是固定在長度=20 的情況下改變 hidden states 的個數，分別比較了 250,500,1000，和第一題 RNN 裡看到的相反，可以發現 500 和 1000 兩

個的 loss 明顯比 250 好了一小段距離，也就是 hidden 的數量越多的時候的表現通常會比較好，而且 1000 的情況下預測的文本也明顯的比 250 還要通順許多。



上圖是固定在 hidden states=500 的情況下改變 input 的長度，分別比較了 10,20,40，可以發現長度改變似乎對 loss 的影響沒有很大，三條線都很擬合，但實際上觀察預測文本的結果可以發現，長度在 10 的時候很容易重複預測同一段字像是 the sore of the sore of the …不斷下去，所以長度月常應該還是在預測上面更加有利。

8. Use RNN or LSTM to generate some words by priming the model with a word related to your dataset.

JULIETIE:
I know you well ar this to dome,
Where is the cody of the sun,
So beaut thou wont thee hone,
Where should not do thee for the brpther's dyes,
Where he for thee and he;
Bnd therefore they are and come against their wiie!

PETRUCHIO:
Oow, sir, 'tis a tiatkng roul.

POLIXENES:
Oot she dome
To see the afversoon of his tons bnd darsiate bnd the diate that hath br rate oo srarin g so a contrary comtage:

RATCLIFF: