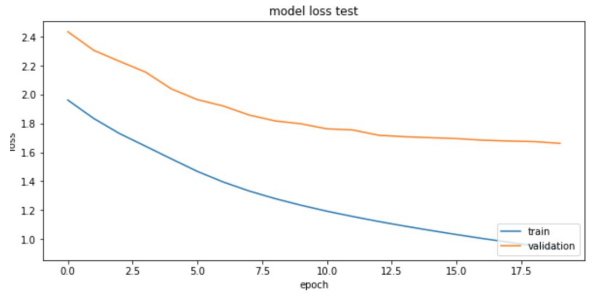
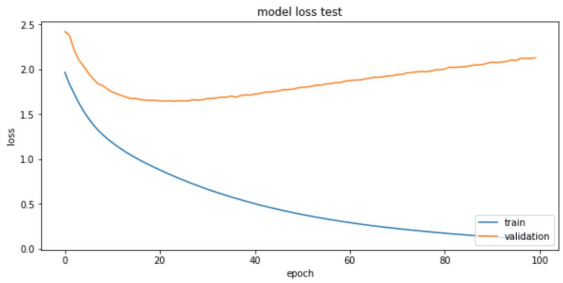
課堂活動 LSTM (2020/0609)	曾宏鈞 06160485 徐友笙 05360365 蔡毓丞 06370136 盧君彥 05360153

1. 英翻中(character-level translator)執行結果畫面

Epoch 20	Epcoh 100
Input sentence: Wait! Decoded sentence: 來！ - Input sentence: Hello! Decoded sentence: 你們都不能嗎？ - Input sentence: I try. Decoded sentence: 我會給你。 - Input sentence: I won! Decoded sentence: 我想要一個。 - Input sentence: Oh no! Decoded sentence: 你們在家嗎？ - Input sentence: Cheers! Decoded sentence: 你們的什麼？ - Input sentence: He ran. Decoded sentence: 他的話是一個好→ - Input sentence: Hop in. Decoded sentence: 別讓我們去。 - Input sentence: I lost. Decoded sentence: 我喜歡這個。	Input sentence: Let's go! Decoded sentence: 我們開始吧！ - Input sentence: Let's go! Decoded sentence: 我們開始吧！ - Input sentence: Let's go! Decoded sentence: 我們開始吧！ - Input sentence: Look out! Decoded sentence: 當心！ - Input sentence: She runs. Decoded sentence: 她跑。 - Input sentence: Stand up. Decoded sentence: 起立。 - Input sentence: They won. Decoded sentence: 他們贏了。 - Input sentence: Tom died. Decoded sentence: 湯姆去世了。 - Input sentence: Tom quit. Decoded sentence: 湯姆不幹了。

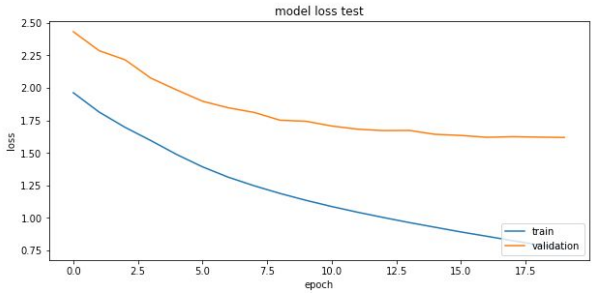
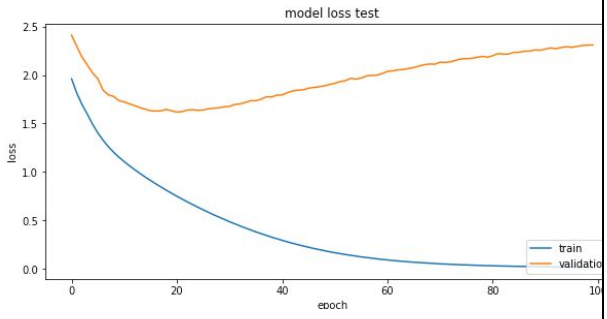
Epoch 20	Epcoh 100
	

心得

觀察由20個epoch的模型可發現，training階段的許縣loss尚還停留在1.0，因此訓練還沒有訓練完整，就結果而言也可看出翻譯的效率並未很好(幾乎全錯)，然而使用100個epoch時可發現，loss緩慢接近0(誤差很小)，雖然他的validation的曲線loss一直升，但從結果看卻比20個epoch較好。

2. 英翻中(wrod-level translator)程式碼修改後畫面與執行結果畫面

Epoch 20	Epoch 100
Input sentence: Let's go! Decoded sentence: 我們吃吧。 - Input sentence: Let's go! Decoded sentence: 我們吃吧。 - Input sentence: Look out! Decoded sentence: 再見。 - Input sentence: She runs. Decoded sentence: 她開始。 - Input sentence: Stand up. Decoded sentence: 開心。 - Input sentence: They won. Decoded sentence: 他們試了。 - Input sentence: Tom died. Decoded sentence: 湯姆走了。 - Input sentence: Tom quit. Decoded sentence: 湯姆不怕。 - Input sentence: Tom swam. Decoded sentence: 湯姆說了。 - Input sentence: Trust me. Decoded sentence: 再我一下。	- Input sentence: Let's go! Decoded sentence: 走吧。 - Input sentence: Look out! Decoded sentence: 當心！ - Input sentence: She runs. Decoded sentence: 她跑。 - Input sentence: Stand up. Decoded sentence: 起立。 - Input sentence: They won. Decoded sentence: 他們贏了。 - Input sentence: Tom died. Decoded sentence: 湯姆去世了。 - Input sentence: Tom quit. Decoded sentence: 湯姆不幹了。 - Input sentence: Tom swam. Decoded sentence: 湯姆遊泳了。 - Input sentence: Trust me. Decoded sentence: 相信我。 - Input sentence: Try hard. Decoded sentence: 努力。

Epoch 20	Epoch 100
	

心得

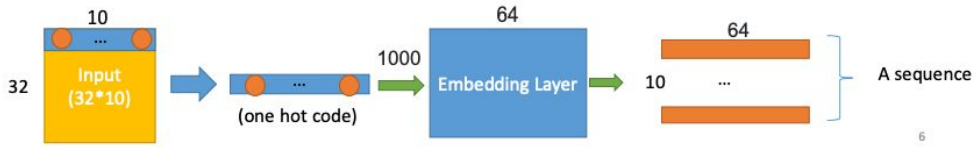
在word-level的實驗中我們可以發現 epoch20的翻譯結果比character-level好上許多，雖然大致上只有一半翻對，然而在100個epoch時，也與上個實驗一樣，增加epoch可讓train loss下降到接近0，但validation的loss一樣是隨著epoch上升而增加。至於為何在此實驗20個epoch會比前一個好，是因為當我們在做encoding時，是以“單字”為一個向量去做編碼(encoding)的，不像前一個是以“字元”做編碼，以單字來做翻譯會比字元的編碼來的準確，只是可以觀察下表發現，word-level在做編碼時，所需的input token也會相對於字元還要多，因為character-level字元編碼只需 大小寫、符號及其他格式，而單字的數量一定比字元數多上許多，雖然word-level在較少的epoch準確度較高，但相對的用到的空間(token)也會比較多，為不同的取捨方式。

character-level	word-level
Number of samples: 10000 Number of unique input tokens: 73 Number of unique output tokens: 2156 Max sequence length for inputs: 30 Max sequence length for outputs: 22 Train on 8000 samples, validate on 2000 samples	Number of samples: 10000 Number of unique input tokens: 3936 Number of unique output tokens: 2156 Max sequence length for inputs: 10 Max sequence length for outputs: 22 /usr/local/lib/python3.6/dist-packages/tensorflo "Converting sparse IndexedSlices to a dense Te Train on 8000 samples, validate on 2000 samples

For Variant Input Lengths of Data (3)

```
model = Sequential()
model.add(Embedding(1000, 64, input_length=10)) #input_length: length of input sequence
# the model will take as input an integer matrix of size (batch, input_length).
# the largest integer (i.e. word index) in the input should be 0 ~ 999 (vocabulary size: 1000).
# now model.output_shape == (None, 10, 64), where None is the batch dimension.

input_array = np.random.randint(1000, size=(32, 10)) #draw 32 of 10-long seq numbers of 0~999
model.compile('rmsprop', 'mse')
output_array = model.predict(input_array)
assert output_array.shape == (32, 10, 64)
```



在前一節實驗我們也知道了seq2seq的加法，此課堂專案是延伸seq2seq的架構，因為要使用seq2seq架構時，需要將資料轉換成詞向量使encoder可以讀入，因此先使用字串切分算出有幾個token，才能根據token的數目去計算one hot encoding的向量，因此在丟入seq2seq架構時，會先根據input計算token、最大字串長度及最大序列長度才能使其成為一個sequence丟入seq2seq架構去做訓練。