- 期末專題題目：
  - Listen and Translate 台語翻譯

- 隊伍名稱：
  - NTU_r06921089_鋼彈盪單槓

- 隊伍成員：
  - R06921089 歐靖、R05921120 黃浩恩、D05921011 賴棹沅

- 工作分配：

姓名	工作項目
R06921089 歐靖	撰寫 NN 訓練模型架構 撰寫輸出檔程式
R05921120 黃浩恩	撰寫資料預處理程式 撰寫 NN 訓練模型架構 撰寫 Report
D05921011 賴棹沅	搜尋不同解題方式與方法 撰寫 NN 訓練模型架構 撰寫 Report

- Preprocessing/Feature Engineering
  - 字典之製作

    在訓練中途曾經想過要用 Facebookresearch 中的 Pre-trained word vectors，但之後發現有許多符號以及可能不會用到的語法，此方式有可能會影響 Embedding 後的 vector 與我們所需的沒有如此匹配，因此我們最後還是採用 gensim 來進行字典的預處理，並使用 training data set 自己訓練一個字典，將出現次數小於 5 的字從字典去除，並按照繁體字筆畫進行排序，使得字典內部的字為出現 5 次以上的字，並訓練該字典為每字 200 維的字典。

    除了文章的字詞外，我們亦增加了 Padding、Start of sentence、End of sentence、Unknow 四個字詞，其對應字典詞彙為<PAD>、<S>、</S>、<UNK>，使得訓練時，可以將其 Padding、Start of sentence、End of sentence、Unknow 之意義丟入 NN 模型訓練。

  - 將 training data set 整理、應用

    在 training data set 內，句子最長之字數為 13，故設計每句句子長度為 15 加上 Start of sentence、End of sentence 共 17 字，其製作 NN 輸入

之台語對應字詞時，我們先將每句透過空格分開，再將前後各插入<S>與</S>，倘若出現了字典內部沒有的詞彙，將給予<UNK>，當該句字數小於 17 時，中間再予以填入<PAD>來進行填補。如：「真厲害」一詞，將會變成「<S>真厲害</S><PAD>……< PAD >」，如此

- word embedding matrix 之製作

  訓練 word embedding 時本團隊採用的是 wor2vec 進行訓練，而非採用 keras 中的 Embedding 進行訓練。利用 wor2vec 訓練完成的 word embedding matrix 搭配 keras 的 Embedding 進行指向，可以使記憶體更有效的被利用，而製做方式則是將每個字典內部之字 200 維的 word vector 按字典順序存入新的 vector，使未來方便於 word embedding layer 上使用。

- 為了 Retrieval Model 擴增 training data set

  特別的是，因後續嘗試了 Retrieval Model，因而對 training data set 做了更新與擴增。我們先在複製一份 training data set，並將原本對應到正確答案的中文字詞進行更改，使得 voice vector 對應到的中文字詞是同長度，但錯誤的其他字句詞彙，並予以 label 為 0，原先正確的 voice vector 與中文字詞對應則予以 label 為 1，因此 training data set 被擴增了一倍，並訓練之輸入 0 與 1 之 label 比例剛好一比一，如此之方法使得機器只要學習該 voice vector 是不是如此的中文字詞就好，不必像 chatbot 一樣的能力，需要知道下句字詞是甚麼。當然 seq2seq 與 Retrieval Model 都有做，故可以也提出比較。

- 為了 Retrieval Model 對 voice vector 加入雜訊、噪音

  為了擴增 training set，也考慮 testing set 的多變，故也對 voice vector 加入雜訊，並再次對應到正確與錯誤兩種不同詞彙之組合可能。透過隨機 random 產生的正負浮點數字，直接對 voice vector 做相加，使得原先的 voice vector 稍稍些許變樣，好讓機器除了有更多的 training data set 可以學習外，亦可學習到 voice vector 變化之狀況。

- Model Description
  - Seq2Seq Model

    Seq2Seq Model 是原先使用的 NN 模型，以下是我們實際運行的架構：

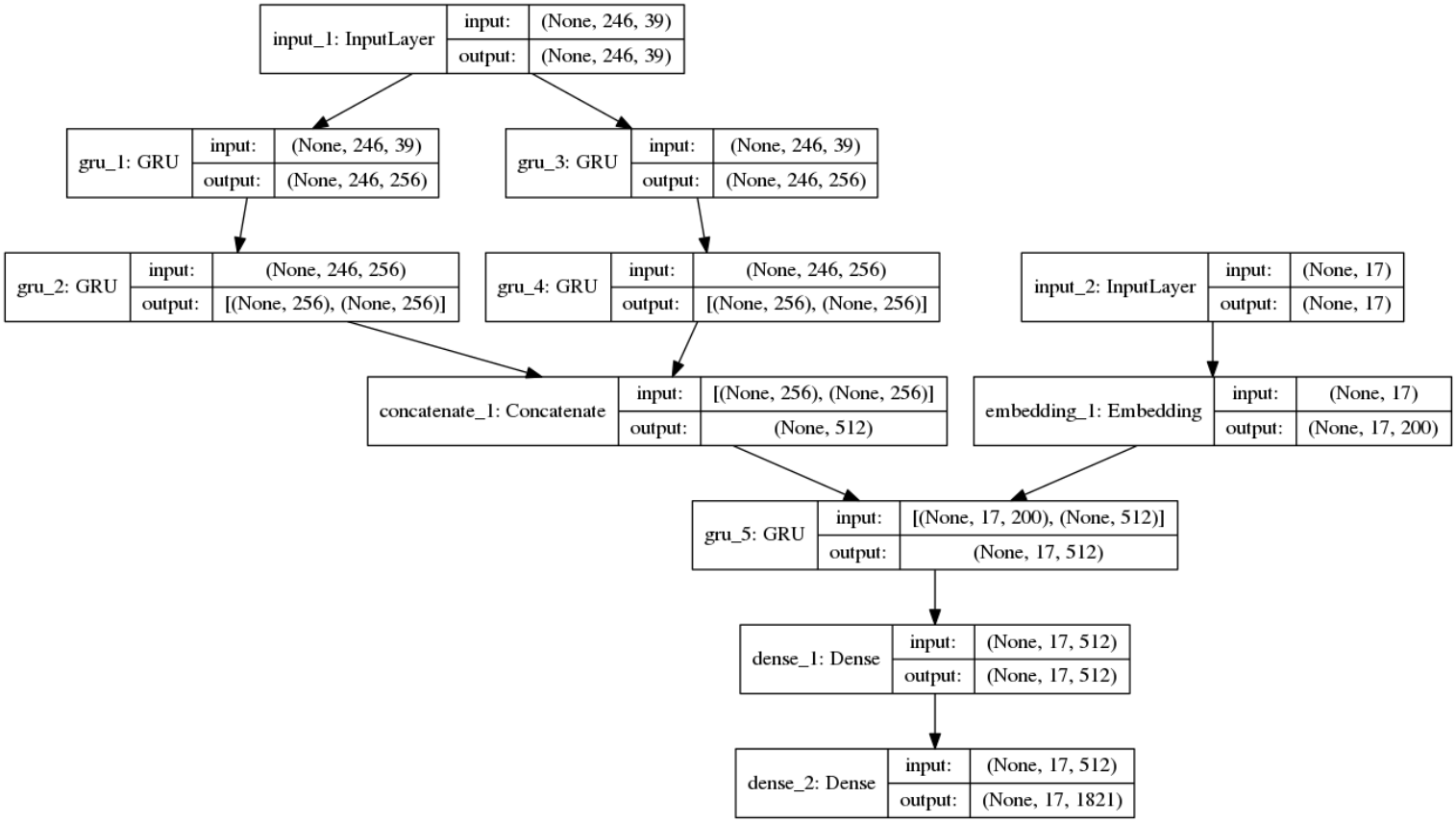

圖 1

  - Seq2Seq Encoder 設計

    Encoder 的部分則是採用 Bidirectional 的 GRU 進行訓練，先將 voice vector 分別丟入一個正向的 GRU 與反向 GRU (go_backwards=True)，使模型能在學習時能將訊息看得更廣，將訓練後最後一次更新的 State 分別由兩個模型中取，再用 concatenate 將兩個 State 接在一起(未採用平均等方式是因為希望能藉由多更多的節點進行訓練)，產生 latest merged encoder state 使得 state 於 decoder 中被接續應用，完成 encoder 設計。

    其中團隊原先希望能採用 Keras 近幾個月才放上的 Birdirectional Model 進行訓練，但是由於此套件還無法將最後一層的 State 有效的匯出，因此最終宣告失敗，回歸使用正向與反向進行訓練。而後亦嘗試疊雙層 Bidirectional 模型進行訓練，如圖 1，但是訓練結果遲遲沒有起色，因此最後選擇使用 Retrieval Model 來進行訓練。

■ Seq2Seq Decoder 設計

圖 2 中，將文字對應 index 的 vector 放入 decoder，先透過 embedding layer 索引該詞彙的 word vector 後，丟到一個使用 encoder state 當 initial state 的 GRU 內，當作 decoder。最後再由兩層 Dence 做合併，並輸出下一個可能字元的 one hot 標籤。

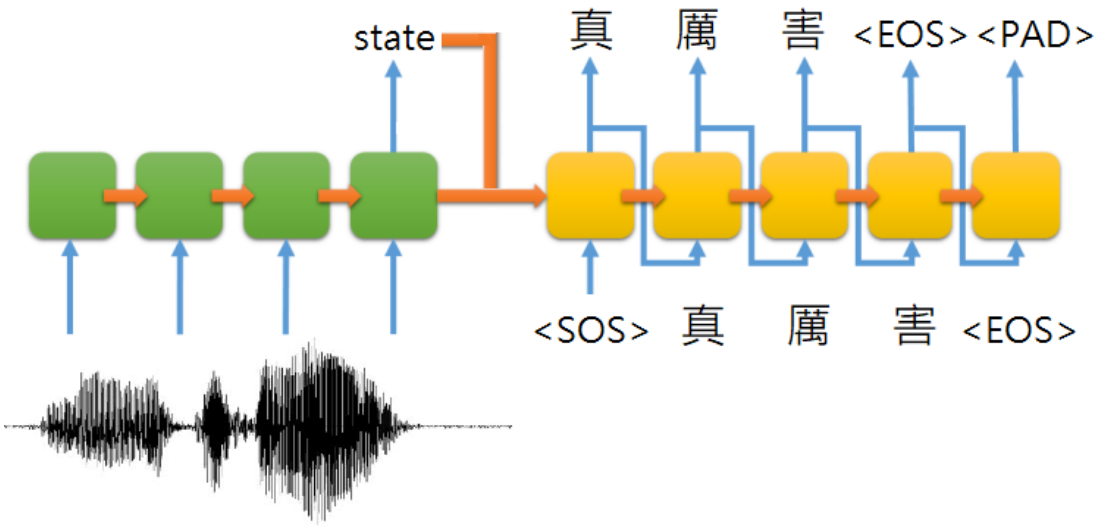

圖 2

■ Retrieval Model

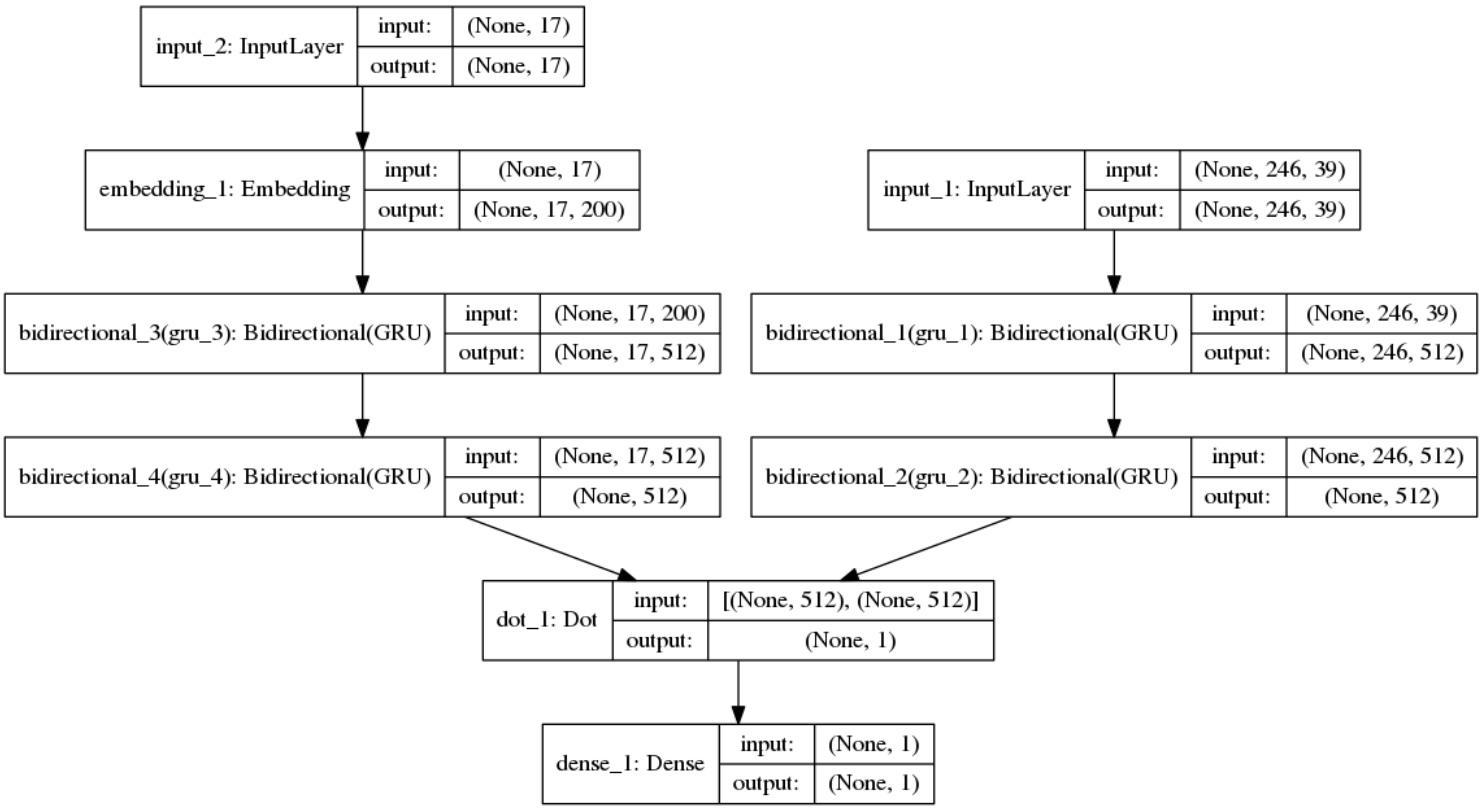

圖 3

原先 Seq2Seq Model 進步緩慢，故搜尋其他方法來實踐此題目，於此將介紹本團隊使用的 Retrieval Model，如圖 3。

圖 4 為模型架構是意圖，將 voice vector 丟入兩層的 Bidirectional GRU layer 內，再將文字對應 index 的 vector，先透過 embedding layer 索引該詞彙的 word vector 後，再丟入兩層的 Bidirectional GRU layer 內，最後 Dot 在一起，使用 Dence 產生是或否之選擇性訓練，使得機器學習快速且直覺。

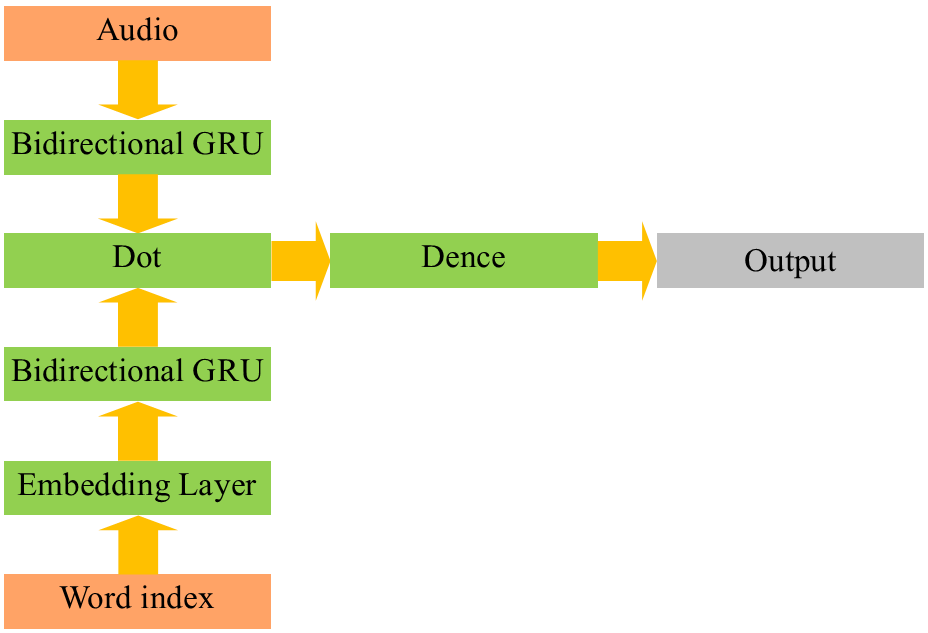

圖 4

- Experiments and Discussion
  - Seq2Seq Model

    於此模型中，其實起初因為 validation set 切太多(8.7%)而使得出來的結果距離 simple base line 差點而錯過加分機會，因為之後覺得 validation 之 loss 跟最後結果相距甚遠，故也不指定 validation 了，結果出來的分數剛好比 simple base line 高一點，才燃起繼續做下去的鬥志。下表與圖 5 為 kaggle 成績與微調項目：

Seq2Seq Model Update	Kaggle Public	Time cost
units=256、validation_split=0.087	0.31700	8 hours
units=256、validation_split=0	0.33000	8 hours
units=512、validation_split=0	0.33600	14 hours
units=1024、validation_split=0	0.34500	18 hours

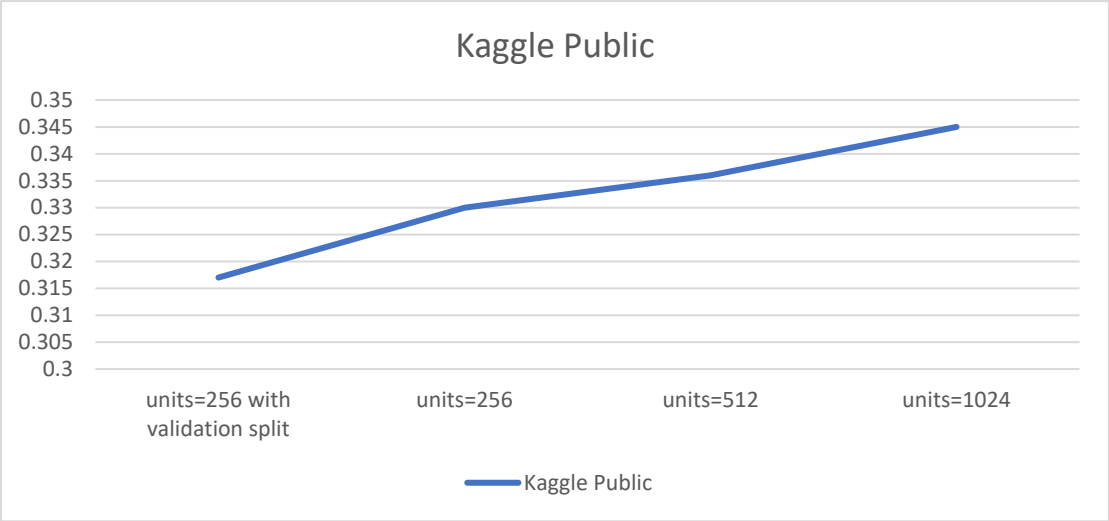

圖 5

於成績最佳之模型的 loss 方面，epochs=26 時 early stopping，如圖 6：

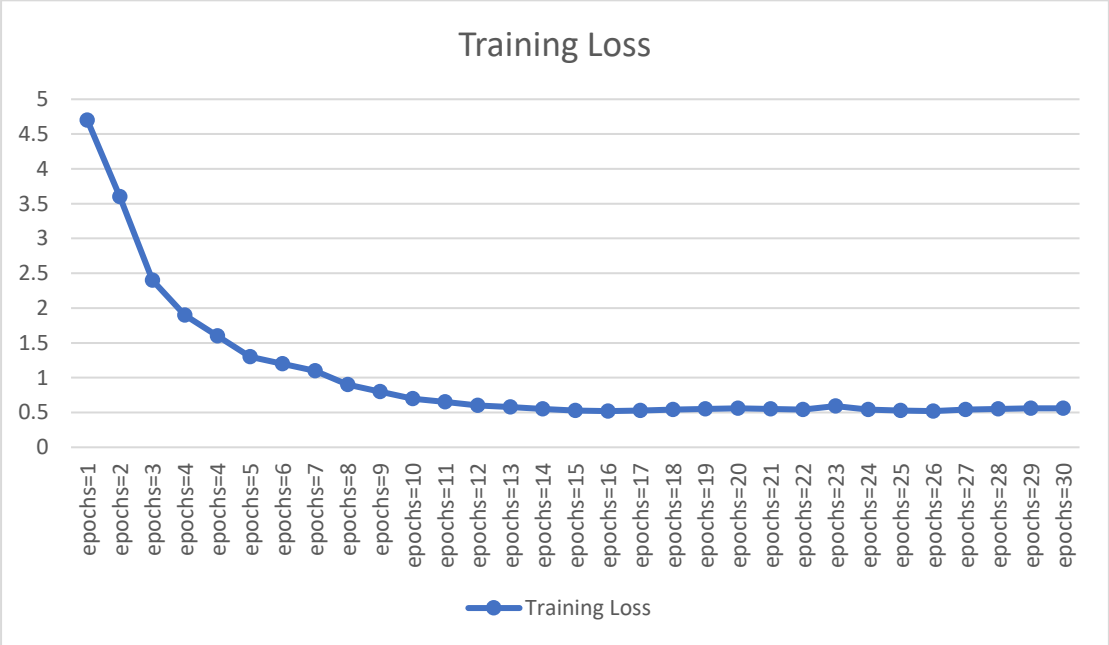

圖 6

- Retrieval Model

於 Retrieval Model 中，除了增加一倍的 training data set 外，更是將整個運作模式做更改，使得機器僅做 1 或 0 的決定，判斷該 voice vector 是不是屬於該選項之結果。其中於 Kaggle 上之結果與更改項目如下表及圖 7 所示：

Retrieval Model Update	Kaggle Public	Time cost
units=64、validation_split=0.03	0.64000	8 hours
units=128、validation_split=0.03	0.70700	10 hours
units=256、validation_split=0.1	0.61500	12 hours
units=512、validation_split=0.1	0.59600	14 hours

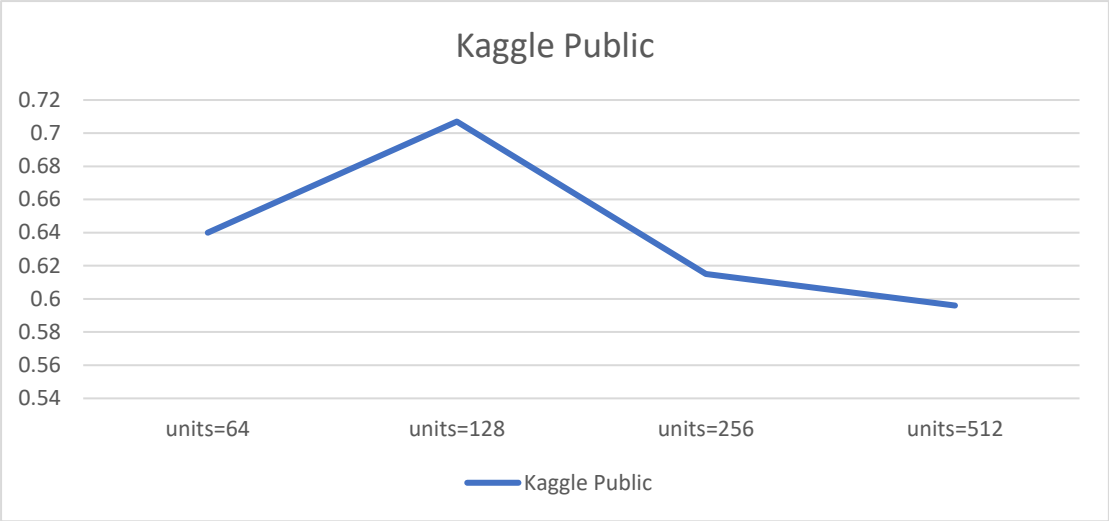

圖 7

於成績最佳之模型的 loss 方面，epochs=103 時 early stopping，如圖 8：

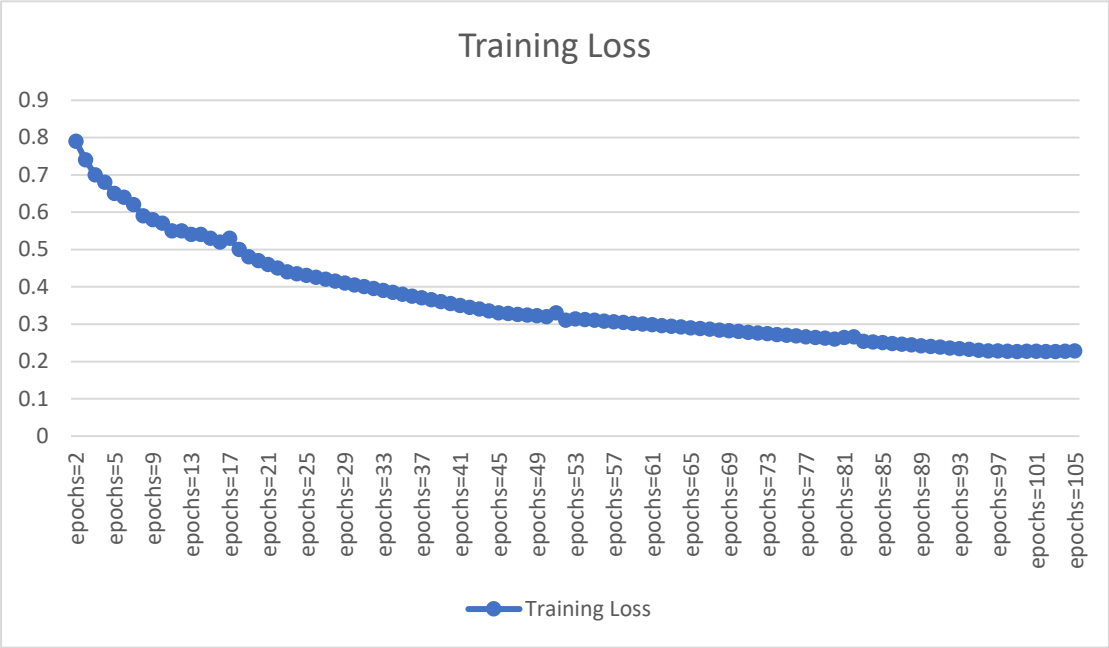

圖 8

■ 討論：

其實花最多時間的模型為 Seq2Seq Model，起初覺得該「Listen and Translate 台語翻譯」較有趣，結果是 Seq2Seq 最 train 不起來的題目，雖然能預判一些詞彙，但整體上沒辦法如同市面上的 chatbot 那麼厲害，回答可接受度那麼高。比如說於 testing data 的第一筆選項「妳說，有啦，妳會，焜肉」之詞彙，與 voice vector 預測結果為「沒不啦<PAD>，沒啦</S>，沒不不</S>，沒演大<PAD>」，其結果差異甚大。雖然如此，我們還是不斷嘗試不同的 Seq2Seq Model 來做測試，但結果都沒有想像中的好，也因此錯過了兩次提前 kaggle 之加分，加上跑一次要將近 18 小時，真的是身心疲累。

Retrieval Model 之方法也是與其他組同學討論得知可以這樣做的，才對我們的 Model 進行修改、更新，結果使用非常短的時間，得到更佳的成果，事後想想老實說這樣才有動力繼續往下做，不然連 simple base line 都過不了真的是蠻氣餒的。

但是覺得此次經驗非常寶貴，轉個彎用另一種思維改進題目並優化，覺得此為期末專題學到最寶貴的一課。