機器學習

期末專題報告

題目:Listen and Translate

組員:電信碩一 黃釋平 R06942082 (Retrival model)

電信碩一 陳建榜 R06942020 (Sequence to Sequence)

電信博一 彭正安 D06942013 (data proccessing)

1. Preprocessing/Feature Engineering

* 音訊檔(train.data)

共有45036筆資料，每筆資料的每個瞬間都是39維的向量，但由於每筆音訊長度不相同，我們必須先對其做padding到246筆，缺少的在後面補上np.zeros(39)，最後資料shape = (45036, 246, 39)。

* 翻譯檔(train.caption)

我們分別使用Seq2seq和Retrival model來進行，詳細的model在下段會介紹，而這兩種模型對中文轉向量的方式不同，但同樣都有做padding使句子長度都為13。

在Seq2seq，我們建立一個one-hot的中文字典，共有2391個字，每個中文字以長度2391的向量代表，其中只有一個值是1，即代表那個字，最後資料shape = (45036, 13, 2391)。

在Retrival model，我們使用gensim.word2vec加上以wiki資料做好的字典model，將字轉成長度300的向量，並做normalization，不在字典中的字則補np.zeros(300)。在訓練時要製造三個錯誤選項，我們以隨機方式取其他行句子作為選項，並且對四個選項shuffle，讓機器自己學該如何選，最後資料shape = (45036, 4, 13, 300)

這樣訓練出來的model，雖然在validation有相當好的表現，但實際測試上傳準確率卻非常低，剛開始相當不明白，我們將訓練資料的隨機選項都故意挑相同長度的再進行測試，結果準確率下降非常多，才發現是句子長度問題。

因此，我們又把生成三個錯誤選項的function加上「同長度」的規則，使每個選項像test.csv一樣，句子都一樣長，詳細trainning狀況會在第三段描述。

* 測試檔(test.data, test.csv)

test.data是音訊檔，同樣必須做padding至246筆，缺少的補上np.zeros(39)，最後shape = (2000, 246, 39)。

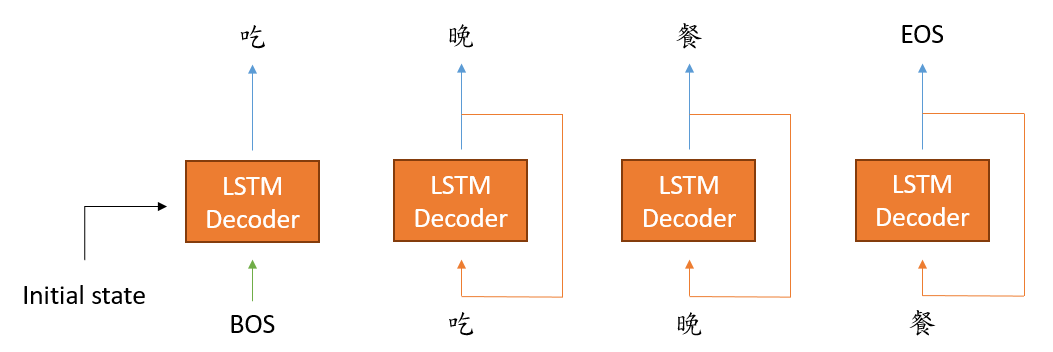
test.csv每行有四個選項，在Seq2seq中以one-hot字典轉成向量，在Retrival model則用gensim.word2vec轉，處理方式同train.caption。

1. Model Description

* Sequence to Sequence (Seq2Seq)

1. 原理

Voice data是聲音檔轉成向量表示，藉此能拿來做運算，一串聲音向量進入Encoder後，全部跑完輸出LSTM state，這個state就是代表這段聲音的意義，直接指定給Decoder當作initial state。假設sequence是這段聲音的翻譯，input\_sequence則要在sequence前面補上開頭符號(BOS)輸入給Decoder，target\_sequence要在結尾補上結束符號(EOS)作為Decoder的目標輸出。



也就是說，Decoder要學到的是，在給定的Initial state(代表情境、意義)之下，不斷預測下一個字，直到出現EOS代表句子完畢;同時，Encorder必需要能生成含有正確情報的LSTM state，提供給Decoder做預測，所以這兩者是相輔相成的。

1. 程式流程

* Retrival model

1. 原理

Information Retrieval原先應用在圖書館系統，使用各種方式將圖書資料的feature擷取出來，用來做檢索跟比對。其中一種叫做vector model，將查詢條件與書目分別encode為同長度的向量，並比對相似度(dot,cos…)，便可知道兩者的相關性。

此法亦可應用在我們的期末專題中，因為是選擇最相關的選項，可以將聲音與選項encode後做比對，相似度最高的即為答案。但在訓練時，由於給定的train.caption只有正確的翻譯，所以我們要刻意製造錯誤選項，將正確的label標為1、錯誤的標為0。

1. 程式流程
2. Experiments and Discussion

* Sequence to Sequence (Seq2Seq)
* Retrival model

剛開始嘗試生成三個錯誤選項，含正確的共有四個。音訊和選項都先經過LSTM(256)-Dense(128)-Batchnormalization()-Activation(‘linear’)轉成128維的向量，音訊向量再分別和選項向量做dot，輸出四個值，最後接Activation(‘softmax’)，代表**要輸出四個選項的為答案的機率，在預測時就選擇機率最高的做為答案**，以下是訓練的情形：

看到45.093%實在欣喜若狂，但實際測試上傳竟只有**22.8%**，與預期差異非常大。經過不斷測試，我們發現是句長問題，test.csv中的選項都是相同句長，我們在生成錯誤選項時並無這規定，將trainning set生成為相同句長選項後代回model測試，果然Accurancy下降非常多。

**我們猜想可能機器學到直接以語音長度和選項長度來做判斷，並沒學到語意，因此在之後都以相同句長的選項來做訓練**，並將validation set與trainning set完全隔離(在隨機生成選項時，不會選到彼此的)，以下是以相同架構重新訓練的結果：

Validation accurancy (val\_acc)卡在32.529%無法上去，Training accurancy (train\_acc)卻可以到將近1，很明顯overfitting，因此我們在音訊、選項的LSTM後都加上dropout(0.25)，再試一次:

val\_acc上升至37.162%，上傳結果**37.3%**，相當符合預期。但是，之後無論如何調整LSTM、Dense、Dropout、Activation，或將dot換成數層的DNN，都沒能改進。

於是我們決定改另一種方式，每次只輸入一個句子，輸出就是「對」的機率(0~1)，**在訓練時每個聲音檔只給一次正確、一次錯誤選項，預測時一樣用argmax來判斷哪個機率最高做為答案**，其餘架構都不變。**注意這種方式acc會從0.5左右開始，因為一筆聲音只對應兩筆資料**，實際訓練結果如下:

在第二圈時達到59.17%，而後不會上升了，我們大膽的以這個結果predict上傳，得到41.8%，相當意外。而一樣觀察到後面有overfitting的現象，試著加上Dropout(0.6)，但無法上升，懷疑抽取過多，而後又調成Dropout(0.25)，結果稍稍上升：

val\_acc上升至62.5%，但上傳卻只有41.5%，甚至不如上個model。**我們懷疑是initial state不太好，所以重新訓練一次，並將trainning data量增加500筆**，訓練的結果：

val\_acc能達到64.366%，上傳為**44.3%**，是目前的最佳紀錄。

**Retrival model 總整理**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| model | Best  epoch | train\_acc | val\_acc | kaggle |
| 4 options (different length) | 8 | 43.39% | 45.09% | 22.80% |
| 4 options (same length) + ~~dot~~→DNN | 8 | 34.94% | 27.99% | 不值得浪費 |
| 4 options (same length) | 10 | 59.92% | 32.53% | 32.90% |
| 4 options (same length) + dropout(0.25) | 17 | 50.93% | 37.16% | 37.30% |
| 1 option (same length) | 2 | 61.73% | 59.17% | 41.80% |
| 1 option (same length) + dropout(0.25) | 14 | 64.84% | 62.50% | 41.50% |
| 1 option (same length) + dropout(0.25)  with more 500 training data | 20 | 70.89% | 64.36% | 44.30% |

以上LSTM皆為256、Dence皆為128，有試著調整過這兩個參數，但沒有好成果，在此不附上。

* Discussion

Q :為何讓模型直接輸出四個選項各為答案的機率，表現會不如讓模型只預測單一輸入是答案的機率，手動預測四次後再比較?

* 我們剛開始認為，也許這樣更能在模型中學到好的結果，但是在四個選項的模型中，選項的神經網路都是共用的，並不是一個選項對應一組LSTM，照理說兩種模型訓練效果應該要完全一樣。

其中唯一的差別，就在於我們餵的資料，**後者只餵一正確一錯誤，前者因為LSTM共用，相當於會拿到25%要輸出「1」和75%要輸出「0」的資料**，或許因此「1」的選項較難上升。舉例來說，如果讓四筆資料都輸出0，loss也會下降很多，機器就以為學好了，實際上會學錯方向。

我們也有觀察到，四個選項的模型最高的機率的確沒有很顯著，例如[25% 26% 19% **30%**]，而一次只預測一個的則就比較明顯，例如[30% 45% 25% **61%**]。

因此，我們認為**關鍵點就在於訓練資料的平不平均。越是不平均的資料，越難fitting出好validation的結果，較容易開始overfitting。**