Machine Learning 2018 Fall
Final Project Report

組別：NTU_b05901011_喔喔喔喔歐翰墨

組員：許秉倫、楊晟甫、歐瀚墨

題目選定：Video Caption

壹、 題目敘述和選題動機

一、題目介紹

給定一個影片，欲從五個選項中選取最符合影片內容的敘述。影片已經由 Feature Extractor 得到 feature，輸入的內容為 80 個 time slice，每個 slice 由 4096 維的向量表示。訓練資料有 1400 多部影片的 features，每一部影片都有數十句對應的敘述。除了 feature 之外，測試資料還包含五句敘述，程式必須能選出最接近影片內容的選項。

二、選題動機

1. 想要挑戰有關影片的機器學習
2. 這個題目是三題中唯一可能用到 Seq2seq 生成序列的技術
3. 有時間的話想利用 GAN 生成更多的 training data
4. 想要挑戰怎麼把不同 domain 的 feature 結合一起運用

貳、資料處理和現成模型的運用

一、預處理

1. W2V

我們在使用 S2VT 模型時，我們需要將每個詞編號，以作為模型預測敘述的輸入判斷依據，我們在此使用 gensim 的 W2V 套件作為實現的方法。另外，在我們的 Two-Way 和 MOSfET 模型中，Caption 端是以 W2V 產生的向量輸入 LSTM 來進行，因此夠好的 W2V 會影響我們的模型之表現。

2. Testing Data 清理

我們發現 testing data 的有些敘述句有含逗號，這樣會讓我們的 csv 讀檔套件讀入不完整的句子，而且造成句子的選項號碼錯誤，因此我們以程式自動修正了這個問題，並以新的 testing options 檔案來作為判斷時的參考。

二、現成的模型 - Sent2Vec[5]

原理：基於 CBOW 和已有的 word2vec 及 network weight 上去實作

優點：可以避免掉有些單字在 word2vec 裡面 mapping 不到的問題，也可以讓相同語意的句子在向量空間上有較大的 cosine similarity

缺點：由於測資只有 2500(500*5)筆，分布在我們所設定的 700dim 上可能會有太分散的情形，不能很好的區分其關聯性，可能會讓某些重要的字(ex. 主詞、動詞)其重要性降低，進而減少準確率

使用：為避免 training data 裡的字未能涵蓋 testing data 裡的關鍵字，我們使用了網路上 pretrained 好的 model：

sent2vec_toronto books_unigrams 2GB
(700dim, trained on the BookCorpus dataset)
sent2vec_toronto books_bigrams 7GB
(700dim, trained on the BookCorpus dataset)

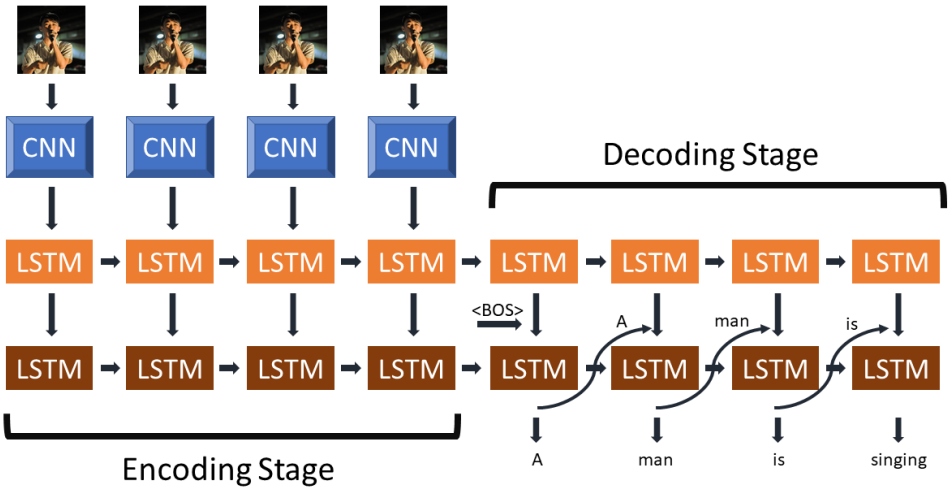
參、模型設計和訓練

一、S2VT

1. 簡介和原理

我們初始的計畫是採用 S2VT[1](Sequence to Sequence Video to Text)模型，由一個 LSTM encoder 和一個 LSTM decoder 組成，其架構參考自 Translating Videos to Natural Language Using Deep Recurrent Neural Networks[1]

2. 結構圖



圖一、S2VT 模型

3. 訓練和測試方法

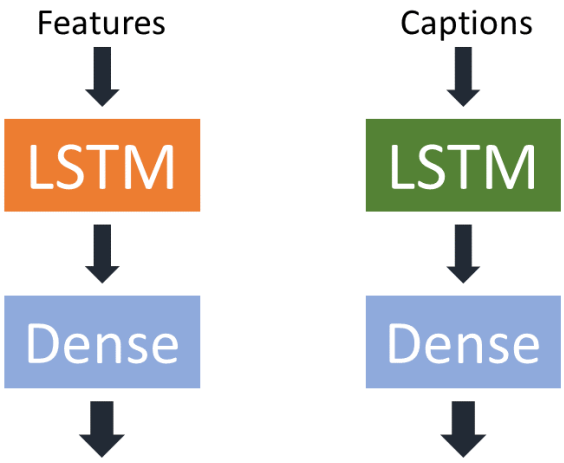
我們將答案的敘述句的每個字編號，模型的輸入是 Feature，輸出的是一個一維陣列，其中每個數字分別代表其對應詞彙的機率。訓練時 decoder input 是 ground truth，測試時則是用已經產生的前一字。

二、Two-Way

1. 簡介和原理

這個做法是延伸之前的方法，只是我們把 captions 也輸入 LSTM 層，基本想法是把 features 和 captions 都投映到一個向量上，並選出 features 的投映和 captions 的投映最接近的選項。

2. 結構圖



圖二、Two-Way 模型

3. 訓練和測試方法

這個模型因為有兩個部份要訓練，因此必須輪流訓練。我們一開始選擇先訓練 captions 的那一邊，然後固定 captions，改成訓練 features，依此輪流訓練 5-10 輪。測試時，我們將選項與影片分別輸入，並比較兩個模型的輸出 MSE 值，選取 MSE 最低的選項。

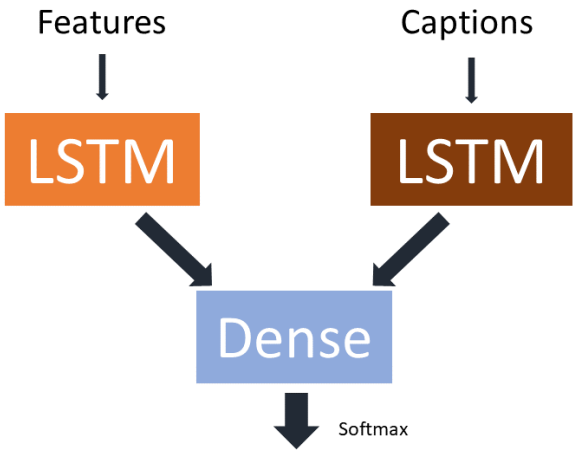
三、Multiple Option Selection for Evaluation Tasks (MOSfET)

1. 簡介和原理

前述模型的缺點在於無法準確分辨差距很大的選項。因此我們必須要有方法可以把錯誤的選項的輸出拉開。在想到這次的題目是要做選擇後，我們就決定以回答選擇題的方式來訓練我們的模型。我們設計的模型會把 features 和 captions 同時讀入，並判斷兩者是否是對應的，輸出是一個 0-1 的數，越接近 1 表示兩者相似度越高。兩個

LSTM 的輸出被合併起來，再以 linear 層降為一個 0-1 的數字。

2. 結構圖



圖三、MOSfET 模型

3. 訓練和測試方法

訓練時，我們有 50%的機率會選中正確的敘述(Ground Truth 為 1)，剩下的隨機從別的影片敘述中選擇，並將 ground truth 設為 0， 作為訓練資料。測試時則分別將五個選項與 feature 一起輸入，並選取值最高的選項。

肆、測試、觀察和調整

一、前面兩個不成功的想法

一開始自然想到的就是 S2VT 的作法，先用 S2VT 產生一段句子，再以 BOW 的相似程度去比較，但是經過多次的嘗試和調整我們仍然無法有效的訓練出任何成果。S2VT 有一個困難是，我們訓練時紀錄的各項數據無法作為測試準確率的任何參考，因此很難判斷訓練是否有進步。

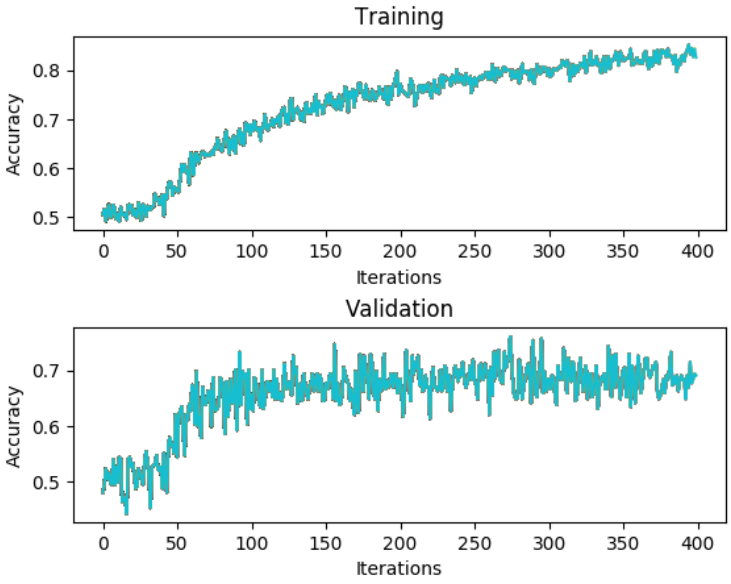
之後我們想到 two-way 的方式，也就是在 feature 的 LSTM 之外，將 captions 送入另一個 LSTM 並使兩者輸出盡量相等，我們一開始認為可能會有某種程度的效果，但是結果使我們非常的失望，經過對於測試時輸出的各選項 MSE 值的觀察，我們發現 two-way 模型有一個很大的瑕疵，那就是兩個模型都會把所有的輸入投映到一樣的值以縮小 MSE，導致無法分離正確和錯誤的選項。

二、MOSfET 的嘗試

接著我們想到，可以訓練判斷敘述句是否符合影片的模型，這樣有一個很直接的優點，就是我們可以同時訓練正確的組合和錯誤的組合，避免前述的問題，同時訓練時的準度可以讓我們看出模型是否有進步。此外，我們認為自己訓練的 W2V 模型因為資料量不夠所以不太

好，於是我們用了網路上現成的 S2V 模型將每個句子降至 dim 是 700 的向量作為 caption 端的輸入。出乎預料的，第一次的嘗試就讓我們通過 strong baseline。下圖是其中一次訓練的曲線:

由圖可見，驗證時的準確率在大約 70-75%達到飽和，我們的做法是挑選最高的那次模型，然後 ensemble 多次訓練的結果。



圖四、訓練曲線

三、優化

我們在剛剛接觸到訓練的影片後就發現一個很有趣的現象，影片都很短而且畫面的變化不大，因此 LSTM 層就好像不重要，可以用一個畫面來取代，於是在修正後，我們每一個影片取一個 time slice 的影像 features 來訓練，結果準確度居然又提高了 10%！於是很自然的，我們訓練時改用隨機擷取畫面，而測試時使用 80 個畫面的平均值來比較選項，又有些取的進步。

我們嘗試過的方法還包含調整正確和錯誤資料的比率，讓我們的 training data 應判斷是 1 的機率是 20%(testing set ground truth 為 1 的機率)，但是這樣做反而發現訓練時的準確率更容易飽和，而且 Kaggle 成績也不太理想，我們猜測是因為這樣調整會過度得讓模型偏好猜 0，造成正向的調整不太容易訓練。

伍、結論

一、題目的困難

在這次的挑戰中，最大的難題便是訓練資料不足，影片的內容千奇百怪但只有一千餘部，同時敘述有很多拼錯的地方和語意、文法上的錯誤，因此，如果只用官方提供的句子來訓練語言模型，勢必得不

到好結果，因此我們使用 S2V 的 pretrain 模組來解決這個問題。

另外，訓練可以自動生成影片敘述的工作十分困難，因為這不但需要很仔細設計的 seq2seq 模型和夠大的資料量，還涉及訓練時的各類監督，以避免模型一直產生類似亂碼的句子，甚至需要 attention 等高難度技術。

二、面對選擇題的訓練

進行這個 project 的過程中，我們發現對直接的方式無法得到成功，甚至完全沒有在正確率上有實質的進展，事實上，產生一個部分正確的句子，仍然無法指向正確的選項。

我們用一個專門為選擇題設計的模型得到成功，利用隨機選取錯誤的選項來擴增訓練資料，並將模型輸出設計成只有一個數字，在選擇題的題目設計下，我們得以讓我們僅有的資料發揮到最大的功能，模型可以在只學到一小部分的辨識能力下做出正確的判斷。

三、總結

Simple is better than complex.

Complex is better than complicated. – The Zen of Python[6]

我們在這次的 project 中，學到一個很寶貴的經驗，那就是：不應該用複雜的 seq2seq 方法來解決選擇題這樣簡單直觀的問題，面對選擇題，就應該用選擇題的作法。能夠用一張圖片辨識處理的問題，就不該用 LSTM 將問題複雜化。

陸、如何執行

1. Requirement

```
sent2vec==0.0.0
numpy==1.15.4
torch==0.4.1
```

2. Installation

```
Run install.sh and it will install sent2vec package for you.
./install.sh
```

3. Reproduce

```
./test.sh <path to data folder> <path to output file>
 e.g ./test.sh /home/final/data ./ans.csv
```

柒、工作分配

許秉倫：S2VT 訓練、W2V 訓練、MOSfET 訓練。

楊晟甫：文獻探討、現成模型研究、Pretrained 模型接口程式。

歐瀚墨：Testing、Two-Way 訓練，訓練程式框架、報告撰寫。

捌、參考資料

1. Saenko, Kate(2017) Translating Videos to Natural Language Using Deep Recurrent Neural Networks Retrieved from : https://berkeley-deep-learning.github.io/cs294-131-s17/slides/saenko-talk.pdf?fbclid=IwAR1gDyidPZ-CPxipsG9sQw62qoWhfbVTP0yJP6YbRN_DiFiA-EQshASyz2c
2. Venugopalan et al. Sequence to Sequence－Video to Text Retrieved from : http://www.cs.utexas.edu/users/ml/papers/venugopalan.iccv15.pdf?fbclid=IwAR0auU-S0yS5vrTIyf0Ynftr1AusoSc6r3-ULU8gqo7YQ-bfgaz4aPzIV8A
3. Ramakanth Pasunuru and Mohit Bansal(2017) Multi-Task Video Captioning with Video and Entailment Generation Retrieved from : https://arxiv.org/pdf/1704.07489.pdf
4. Shen et al. (2017) Weakly Supervised Dense Video Captioning Retrieved from : https://arxiv.org/pdf/1704.01502.pdf
5. Epfml sent2vec Retrived from: https://github.com/epfml/sent2vec?fbclid=IwAR1uvIUq_Ef3kFfKu6E1mA8TRYyhRVGRWPHpX3JnsHB_NvAkim-SHvfTUIs
6. https://www.python.org/dev/peps/pep-0020/