

## 機器學習及其深度化與結構化 HW2-1

姓名	學號	工作
吳政軒	R06922118	2-1
黃敬庭	R06944049	2-2
馬欣婕	R06946010	Report

### ● Model description (3%)

#### Build Dictionary

將 training data 的字去掉標點符號並轉成小寫後，將有出現過的字建立 index，還有一些 special token，如<UNK>、<BOS>、<EOS>、<PAD>。

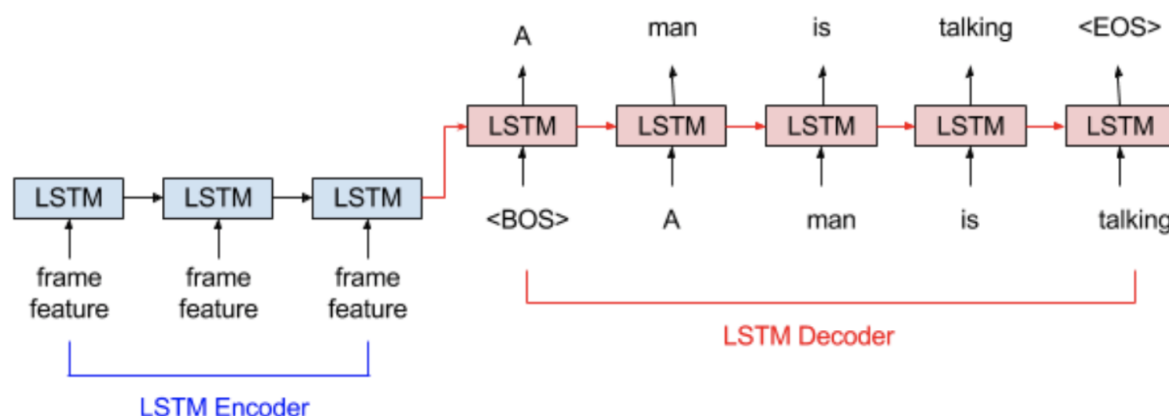
#### Build Word Vector

我們將字典裡的所有字都建立一個初始的 word embedding，我們使用 glove pretrain 好的 300 維 embedding，如果字典的字出現在 glove 裡就用 glove 的 embedding，如果沒有則 random initial 300 維的 embedding。

#### Preprocess Training Data

將訓練資料的每個句子開頭加上<BOS>，結尾加上<EOS>，不足 20 個字的就在最後面加上<PAD>，超過 20 個字則捨去後面的字，最後再將所有字轉成先前建立好的 index，就是最後的訓練資料，其中前 20000 筆是 training data，其餘為 validation data。

#### Model Structure



我們將前面建立好的 word vector 當作 embedding layer 的初始值，將 label 的 index 轉換成 embedding，然後使用一層的 LSTM 當作 encoder (加上 Dropout)，將影片裡 80 個 frame 的 feature 送進 encoder，再拿另一個一層的 LSTM 當作 decoder(加上 Dropout)，將 encoder 最後一個 hidden state 送進 decoder，並且每個 timestamp 都餵應該要產生的答案，使用 RMSPropOptimizer 來訓練，其中我們有做 clip gradient 的動作。

### Inference

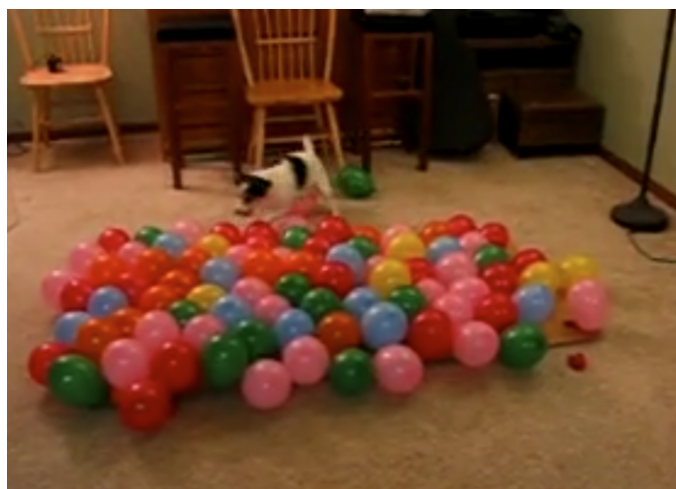
Encoder 輸入影片每個 frame 的 feature，接著餵<BOS>給 decoder，開始產生 output，將 softmax 後機率最大的字當作下一個時間點 decoder 的 input，重複此步驟直到 decoder 輸出<EOS>，再將這些 index sequence 轉回字，就是最後的 caption。

### Model parameters

Pretrained word embedding: Glove 300 維  
LSTM hidden units: 256  
Max caption length: 20  
Epoch: 100 (如果連續 3 次 validation loss 沒進步就 stop)  
Batch size: 100  
Max gradient: 5  
Initial learning rate: 1e-3  
Dropout keep\_prob: 0.9

## ● How to improve your performance (3%)

- Write down the method that makes you outstanding (1%)  
我們加上 Attention 的機制，讓 decoder 在產生字時，除了考慮該時間點的 input，也考慮 encoder 所有時間點的 hidden states。
- Why do you use it (1%)  
因為 decoder 產生每個字時，影片中每個 feature 的影響應該是不同的，使用 attention 可以讓 decoder 專注在特定的 feature。
- Analysis and compare your model without the method. (1%)



針對同一部影片 ( k06Ge9ANKM8\_5\_16.avi )

沒有加 Attention 的 model 產生的 caption: a dog is running on a man

加上 Attention 後的 model 產生的 caption: a cat is playing with a ball

可以發現加上 Attention 後產生的 caption 明顯比較符合。

### ● Experimental results and settings (1%)

- parameter tuning, schedual sampling ... etc

我們設定不同的 dropout rate 和不同的 rnn\_units 做了一些測試，BLEU Score 如下：

dropout keep\_prob: 0.7

rnn_units	64	128	256	512
basic	0.5988	0.6113	0.6442	0.6380
attention	0.6356	0.6277	0.6615	0.6346

dropout keep\_prob: 0.8

rnn_units	64	128	256	512
basic	0.5824	0.6127	0.6328	0.6033
attention	0.6221	0.6346	0.6353	0.6351

dropout keep\_prob: 0.9

rnn_units	64	128	256	512
basic	0.6343	0.6276	0.6352	0.6481
attention	0.6241	0.6446	0.6267	0.6387

- **README**

tensorflow 1.6.0

numpy 1.14.2

tqdm 4.23.0