
Machine Learning HW4

Recurrent Neural Networks

ML TAs

— ntu-ml-2020spring-ta@googlegroups.com —

Outline

1. Requirements
2. Task Introduction
3. Data Format
4. Kaggle
5. Rules, Deadline, Policy, Score
6. FAQ

Requirements

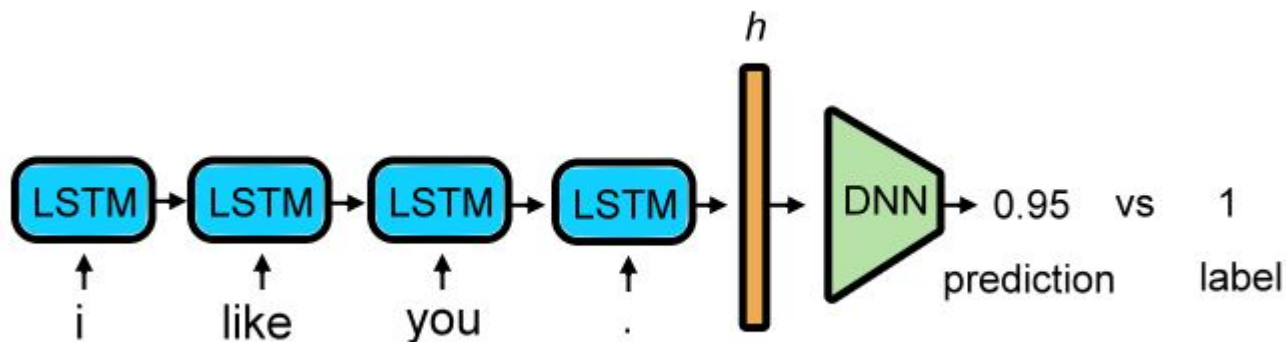
- 請使用 RNN 實作
- 不能使用額外 data (禁止使用其他 corpus 或 pretrained model)
- 請附上訓練好的 best model (及其參數) 至 GitHub release 或 Dropbox, 並於 hw4_test.sh 中寫下載的 command (可參照[這裡](#)的方法)
 - model 大小在 100MB 以內的可以直接上傳到 GitHub
- hw4_test.sh 要在 10 分鐘內跑完 (model 下載時間不包含在此)
- 套件的部份請參考[期初公告](#)

Task introduction

(Text Sentiment Classification)

Task - Text Sentiment Classification

```
0 +++$+++ on the flipside ... completely bummed that there isn ' t a or sighting .  
1 +++$+++ahaha im here carlos wasssup ?!  
0 +++$+++ at least they text you  
0 +++$+++ i feel icky , i need a hug  
1 +++$+++ hey that ' s something i ' d do !  
1 +++$+++ thanks ! i love the color selectors , btw . that ' s a great way to search and list .
```



Text Sentiment Classification

本次作業為 Twitter 上收集到的推文，每則推文都會被標注為正面或負面，如：

```
1 +++$+++ thanks ! i love the color selectors , btw . that ' s a great way to search and list .
```

1:正面

```
0 +++$+++ i feel icky , i need a hug
```

0:負面

除了 labeled data 以外，我們還額外提供了 120 萬筆左右的 unlabeled data

- labeled training data :20萬
- unlabeled training data :120萬
- testing data :20萬(10 萬 public, 10 萬 private)

Preprocessing the sentences

- 先建立字典, 字典內含有每一個字所對應到的 index
example:

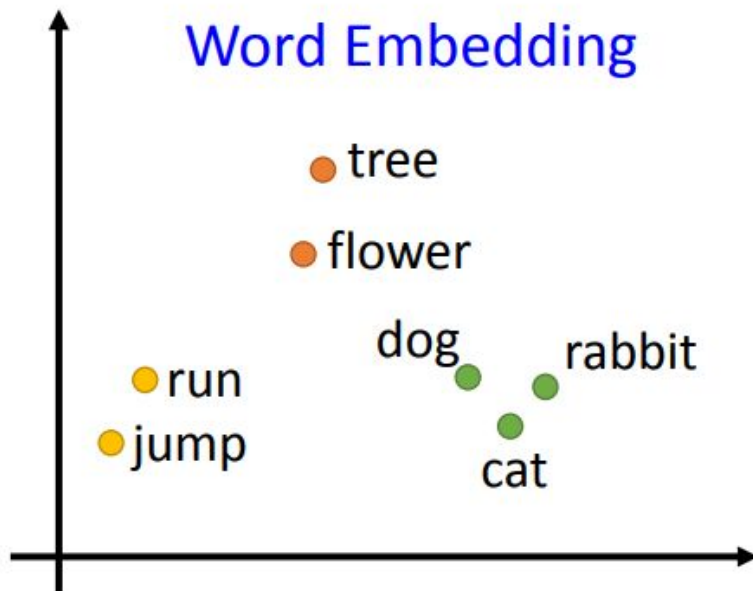
“I have a pen.” -> [1, 2, 3, 4]

“I have an apple.” -> [1, 2, 5, 6]

- 利用 Word Embedding 來代表每一個單字,
並藉由 RNN model 得到一個代表該句的 vector (這份投影片 p.5 的 h)
- 或可直接用 bag of words (BOW) 的方式獲得代表該句的 vector

What is Word Embedding

- 用一個向量 (vector) 表示字 (詞) 的意思



1-of-N encoding

- 假設有一個五個字的字典 [apple, bag, cat, dog, elephant]
我們可以用不同的 one-hot vector 來代表這個字

apple -> [1,0,0,0,0]

bag -> [0,1,0,0,0]

cat -> [0,0,1,0,0]

dog -> [0,0,0,1,0]

elephant -> [0,0,0,0,1]

ref: RNN投影片p4

[http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2016/Lecture/RN%20\(v2\).pdf](http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2016/Lecture/RN%20(v2).pdf)

- Issue :

- 缺少字與字之間的關聯性 (當然你可以相信 NN 很強大他會自己想辦法)
- 很吃記憶體 one hot encoding太吃記憶體 而且label之間的關聯性被弄掉了

$$200000(\text{data}) * 30(\text{length}) * 20000(\text{vocab size}) * 4(\text{Byte}) = 4.8 * 10^{11} = \mathbf{480 \text{ GB}}$$

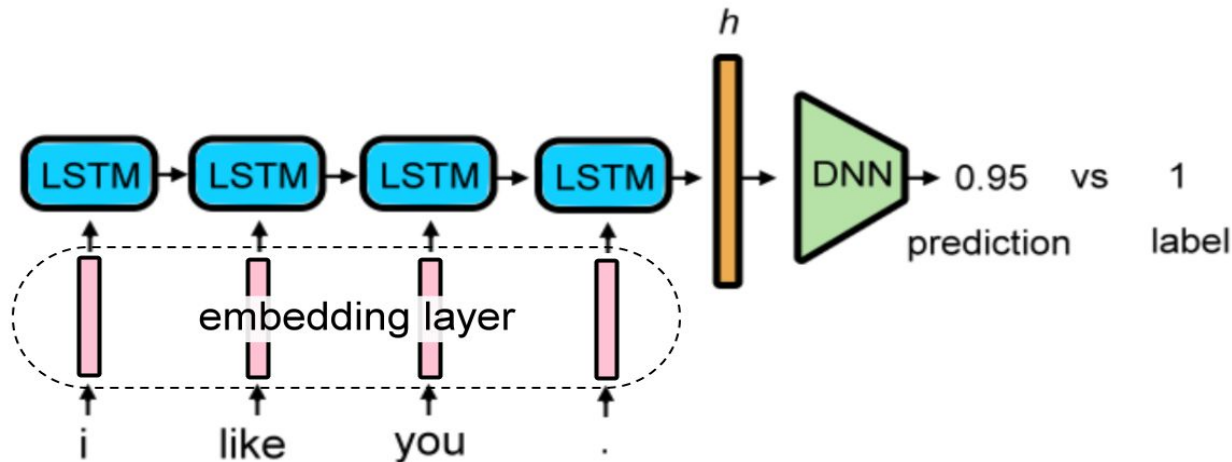
Word Embedding

- 用一些方法 pretrain 出 word embedding (e.g., skip-gram, CBOW.)

reference : [http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2017/Lecture/word2vec%20\(v2\).pdf](http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2017/Lecture/word2vec%20(v2).pdf)

小提醒:如果要實作這個方法, pretrain 的 data 也要是作業提供的 !

- 然後跟 model 的其他部分一起 train ([colab Model line 12](#))



Bag of Words (BOW)

- BOW 的概念就是將**句子**裡的文字變成一個袋子裝著這些詞的方式表現，這種表現方式不考慮文法以及詞的順序。

例如：

(1) John likes to watch movies. Mary likes movies too.

(2) John also likes to watch football games.

在 BOW 的表示方法下，會變成：

(1) -> [1, 2, 1, 1, 2, 0, 0, 0, 1, 1]

(2) -> [1, 1, 1, 1, 0, 1, 1, 1, 0, 0]

有重複的字詞數量變大

dictionary

["John", "likes", "to",
"watch", "movies",
"also", "football",
"games", "Mary", "too"]

BOW



0.95
prediction

vs

1
label

Semi-supervised Learning

- semi-supervised 簡單來說就是讓機器利用 unlabeled data , 而方法有很多種, 這邊簡單介紹其中一種比較好實作的方法 Self-Training
- Self-Training:

把 train 好的 model 對 unlabeled data 做預測, 並將這些預測後的值轉成該筆 unlabeled data 的 label, 並加入這些新的 data 做 training。


你可以調整不同的 threshold, 或是多次取樣來得到比較有信心的 data。

e.g., 設定 $\text{pos_threshold} = 0.8$, 只有 $\text{prediction} > 0.8$ 的 data 會被標上 1

Data Format

Data Format (labeled data)

label +++\$+++ text



```
0 +++$+++ on the flipside ... completely bummed that there isn ' t a or sighting .  
1 +++$+++ahaha im here carlos wasssup ?!  
0 +++$+++at least they text you  
0 +++$+++i feel icky , i need a hug  
1 +++$+++hey that ' s something i ' d do !  
1 +++$+++thanks ! i love the color selectors , btw . that ' s a great way to search and list .
```

Data Format (unlabeled data)

text

```
7 1 more day !  
8 nursing celeste with a tummy ache .  
9 hates being this burnt !! ouch  
10 just couldn ' t sleep last night . working 7a 3p , than dinner with megan . happy bday jl !  
11 i love slaves ! by david raccah , linkedin , rotfl  
12 is being super organised and making up orders to post first thing tomorrow !  
13 laying in the bed . it feels soooooo good . what a long day  
14 finally , at the airport . currently chilling out at the citibank lounge . maaaaan , the wi fi here doesn ' t work ! lameeee !  
15 back and still feeling shattered . still no cockney ... i ' m ashamed to say .  
16 so do i
```

Training Time

- colab code 跑 20個epoch
 - real 3m33.317s
 - user 3m29.813s
 - sys 1m9.469s

Kaggle

Kaggle submission format

Kaggle link: <https://www.kaggle.com/c/ml2020spring-hw4>

請預測 testing set 中二十萬筆資料並將結果上傳 Kaggle

1. 上傳格式為 csv 檔。
2. 第一行必須為 id, label, 第二行開始為預測結果。
3. 每行分別為 id 以及預測的 label, 請以逗號分隔。
4. Evaluation: accuracy

1	id, label
2	0,0
3	1,0
4	2,0
5	3,0
6	4,0
7	5,0
8	6,0
9	7,0
10	8,0
11	9,0
12	10,0
13	11,0
14	12,0
15	13,0
16	14,0
17	15,0
18	16,0
19	17,0
20	18,0
21	19,0

Rules, Deadline, Policy, Score

Policy

GitHub 上 hw4-<account> 裡面請至少包含:(1, 2, 3的檔名請務必一模一樣)

1. report.pdf
2. hw4_train.sh
3. hw4_test.sh
4. train/test Python files
5. model 參數 (Make sure it can be downloaded by your script.)
 - 請將 model 下載到與 script 相同的位置
 - 上傳的 model 總和大小建議在 600 MB 以內

請不要上傳 dataset, 請不要上傳 dataset, 請不要上傳 dataset

Policy

1. 以下的路徑, 助教在跑的時候會另外指定, 請保留可更改的彈性, 不要寫死
2. Script usage:

```
bash hw4_train.sh <training label data> <training unlabel data>
```

training label data: training_label.txt 的路徑

training unlabel data: training_nolabel.txt 的路徑

```
bash hw4_test.sh <testing data> <prediction file>
```

testing data: testing_data.txt 的路徑

prediction file: 輸出結果的 csv 檔路徑

(除非有狀況, 不然原則上助教只會跑 testing, 不會跑 training, 因此請用讀取 model 參數的方式進行預測。)

Score - Report.pdf

Report: <https://reurl.cc/K6yybR>
Collaborators 請附上學號與姓名

- (1%) 請說明你實作的 RNN 的模型架構、word embedding 方法、訓練過程 (learning curve) 和準確率為何？(盡量是過 public strong baseline 的 model)
- (2%) 請比較 BOW + DNN 與 RNN 兩種不同 model 對於 "today is a good day, but it is hot" 與 "today is hot, but it is a good day" 這兩句的分數 (過 softmax 後的數值), 並討論造成差異的原因。
- (1%) 請敘述你如何 improve performance (preprocess、embedding、架構等等), 並解釋為何這些做法可以使模型進步, 並列出準確率與 improve 前的差異。(semi-supervised 的部分請在下題回答)
- (2%) 請描述你的 semi-supervised 方法是如何標記 label, 並比較有無 semi-supervised training 對準確率的影響並試著探討原因 (因為 semi-supervised learning 在 labeled training data 數量較少時, 比較能夠發揮作用, 所以在實作本題時, 建議把有 label 的 training data 從 20 萬筆減少到 2 萬筆以下, 在這樣的實驗設定下, 比較容易觀察到 semi-supervised learning 所帶來的幫助)

FAQ

- 若有其他問題, 請貼在 FB 社團裡或寄信至助教信箱, **請勿直接私訊助教。**
- 助教信箱: ntu-ml-2020spring-ta@googlegroups.com

Link

- 雲端使用方法: <http://slides.com/sunprinces/deck-16#/>
- Kaggle: <https://www.kaggle.com/c/ml2020spring-hw4>
- Report template: <https://reurl.cc/7X9yby>
- 遲交表單: <https://bit.ly/39d2x2m>