RNN練習

Prediction of Paper Acceptance

2019/7/5

Outline

- Task Description - Prediction of Paper Acceptance
- Data Format
- Data Preprocessing
- Discussion

Task Description

- 實做RNN/LSTM/GRU去判斷ICLR paper acceptance

Data format

- Dataset包含所有ICLR 2017、ICLR 2018 paper的標題，共有ICLR accepted.xlsx、ICLR rejected.xlsx兩個檔案
    - ICLR accepted.xlsx : 共582筆，以前50筆作為testing_data，之後的作為training_data
    - ICLR rejected.xlsx : 共753筆，以前50筆作為testing_data，之後的作為training_data

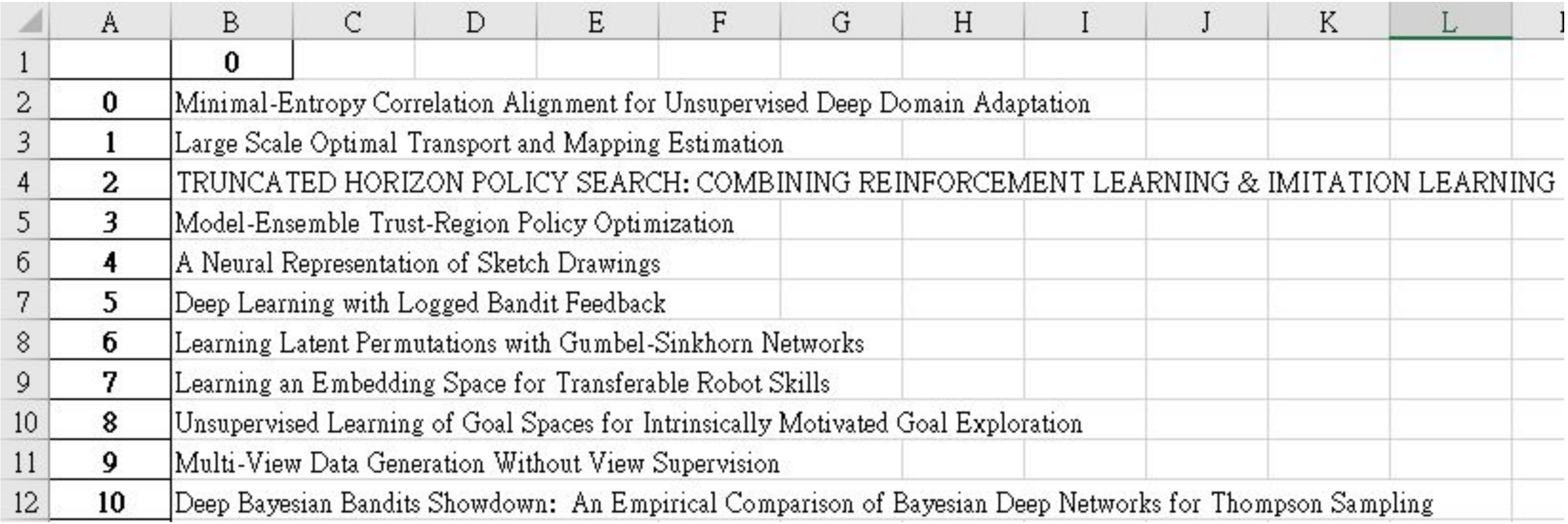
- Data下載處

  https://drive.google.com/open?id=1EB_umXWj0KARvGgM_YzSFwUcjAgfdsDs

  https://drive.google.com/open?id=1DLtzbMnvwYrjWjolgeT9LT5ij0DAJwky

- The dataset came from https://openreview.net/group?id=ICLR.cc/2017/conference https://openreview.net/group?id=ICLR.cc/2018/Conference

Data format



	A	B
1		0
2	0	Minimal-Entropy Correlation Alignment for Unsupervised Deep Domain Adaptation
3	1	Large Scale Optimal Transport and Mapping Estimation
4	2	TRUNCATED HORIZON POLICY SEARCH: COMBINING REINFORCEMENT LEARNING & IMITATION LEARNING
5	3	Model-Ensemble Trust-Region Policy Optimization
6	4	A Neural Representation of Sketch Drawings
7	5	Deep Learning with Logged Bandit Feedback
8	6	Learning Latent Permutations with Gumbel-Sinkhorn Networks
9	7	Learning an Embedding Space for Transferable Robot Skills
10	8	Unsupervised Learning of Goal Spaces for Intrinsically Motivated Goal Exploration
11	9	Multi-View Data Generation Without View Supervision
12	10	Deep Bayesian Bandits Showdown: An Empirical Comparison of Bayesian Deep Networks for Thompson Sampling

Word Vectors

- 將每個字/詞轉換為 vector 以利後續 model training。

- 如何將字/詞轉換為 vector？
  - One-hot Encoding
  - Word Embedding

One-hot Encoding

- 假設有一個五個字的字典 [1,2,3,4,5]
  我們可以用不同的one-hot vector來代表這個字

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

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

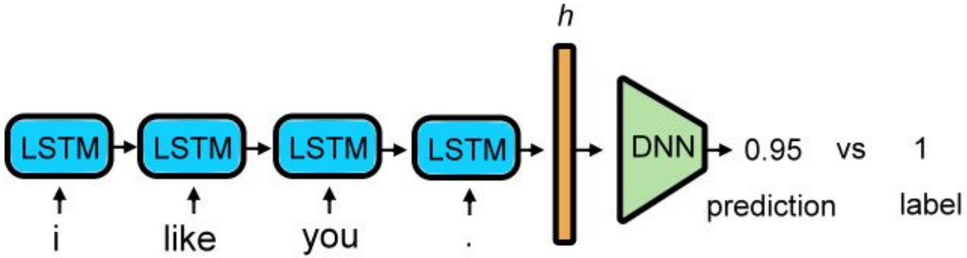
  3 -> [0,0,1,0,0]

  4 -> [0,0,0,1,0]
- 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

200000(data)*30(length)*20000(vocab size) *4(Byte) = 4.8*10^11 = 480 GB

Word Embedding

- 用一個向量(vector)表示字(詞)的意思
- 用一些方法 pretrain 出 word embedding (ex : skip-gram、CBOW)
  可使用 Word2Vec 實做（套件 : gensim/GloVe）
- 或是跟 model 的其他部分一起 train



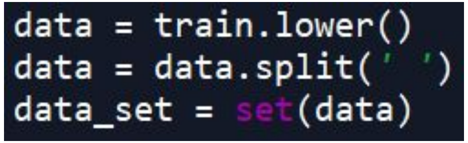
Data Preprocessing

1. **切training_data & testing_data**

   ICLR accepted及ICLR rejected 皆以前50筆作為testing_data，之後的作為training_data

2. **把所有字母變小寫再依空格切字並轉成set**

ex:



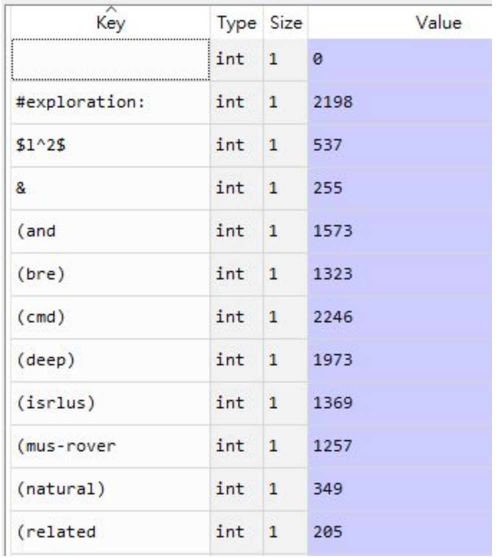
```
data = train.lower()
data = data.split(' ')
data_set = set(data)
```

Data Preprocessing

3. **建立自己的字典**

Ex : given the sequence data {”NCTU is good”}

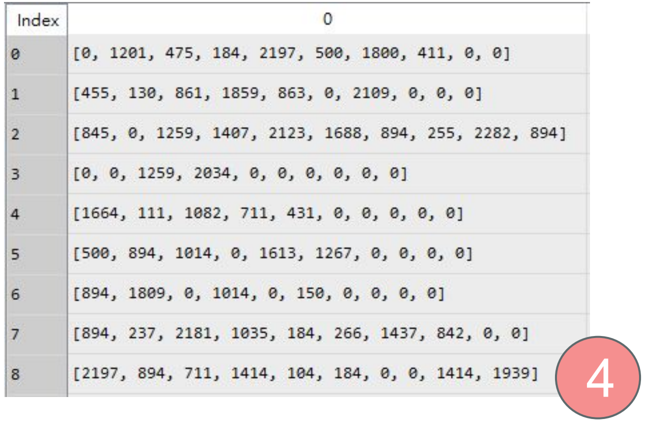
⇨ build a dictionary { 0 : ”NCTU”, 1 : ”is”, 2 : ”good” }
then convert the sequence to [0, 1, 2]



Key	Type	Size	Value
	int	1	0
#exploration:	int	1	2198
1^2	int	1	537
&	int	1	255
(and	int	1	1573
(bre)	int	1	1323
(cmd)	int	1	2246
(deep)	int	1	1973
(isrlus)	int	1	1369
(mus-rover	int	1	1257
(natural)	int	1	349
(related	int	1	205

Data Preprocessing

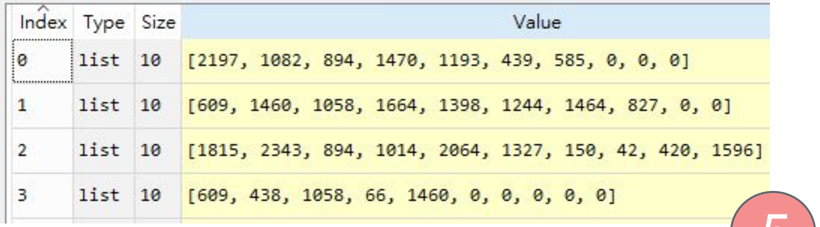
4. **把**ICLR accepted及ICLR rejected
   對照字典轉換成序列



Index	0
0	[0, 1201, 475, 184, 2197, 500, 1800, 411, 0, 0]
1	[455, 130, 861, 1859, 863, 0, 2109, 0, 0, 0]
2	[845, 0, 1259, 1407, 2123, 1688, 894, 255, 2282, 894]
3	[0, 0, 1259, 2034, 0, 0, 0, 0, 0, 0]
4	[1664, 111, 1082, 711, 431, 0, 0, 0, 0, 0]
5	[500, 894, 1014, 0, 1613, 1267, 0, 0, 0, 0]
6	[894, 1809, 0, 1014, 0, 150, 0, 0, 0, 0]
7	[894, 237, 2181, 1035, 184, 266, 1437, 842, 0, 0]
8	[2197, 894, 711, 1414, 104, 184, 0, 0, 1414, 1939]

4

5. **把每個句子padding到同樣長度**



Index	Type	Size	Value
0	list	10	[2197, 1082, 894, 1470, 1193, 439, 585, 0, 0, 0]
1	list	10	[609, 1460, 1058, 1664, 1398, 1244, 1464, 827, 0, 0]
2	list	10	[1815, 2343, 894, 1014, 2064, 1327, 150, 42, 420, 1596]
3	list	10	[609, 438, 1058, 66, 1460, 0, 0, 0, 0, 0]

5

Word Embedding-Keras

```
keras.layers.Embedding(input_dim, output_dim, input_length)
```

input_dim = 總詞彙編碼數量

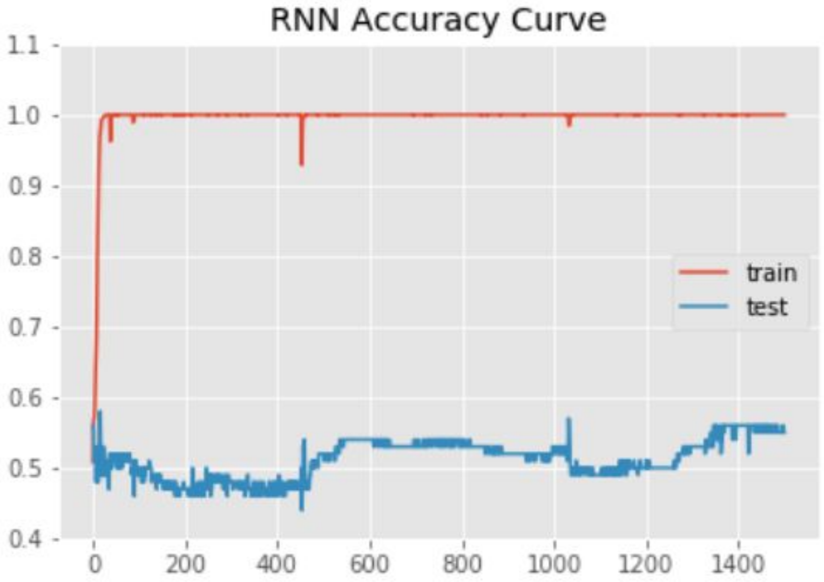
output_dim = 詞彙向量大小

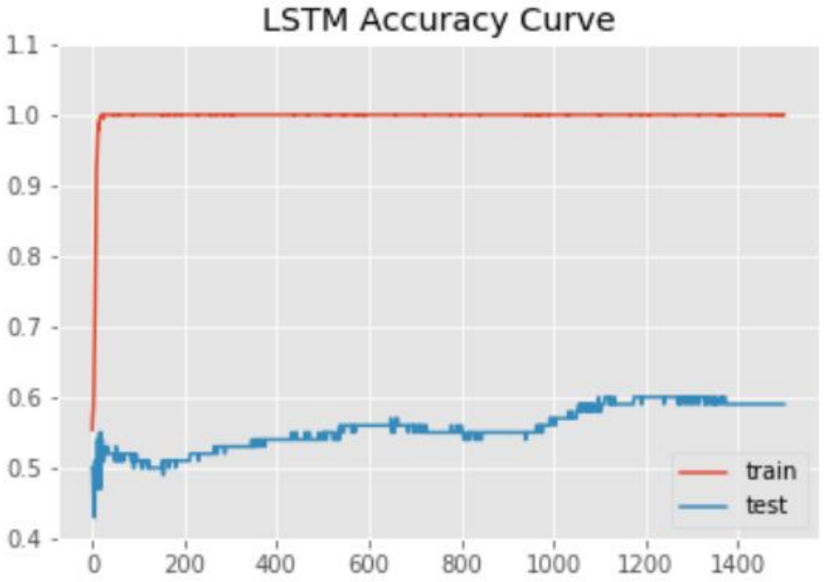
input_length = 序列長度(padding的數量)

Discussion

- optimizer?
- loss function?
- activation function?

Results


RNN Accuracy Curve
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
200
400
600
800
1000
1200
1400
train
test


LSTM Accuracy Curve
1.1
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0
200
400
600
800
1000
1200
1400
train
test