RNN

Outline

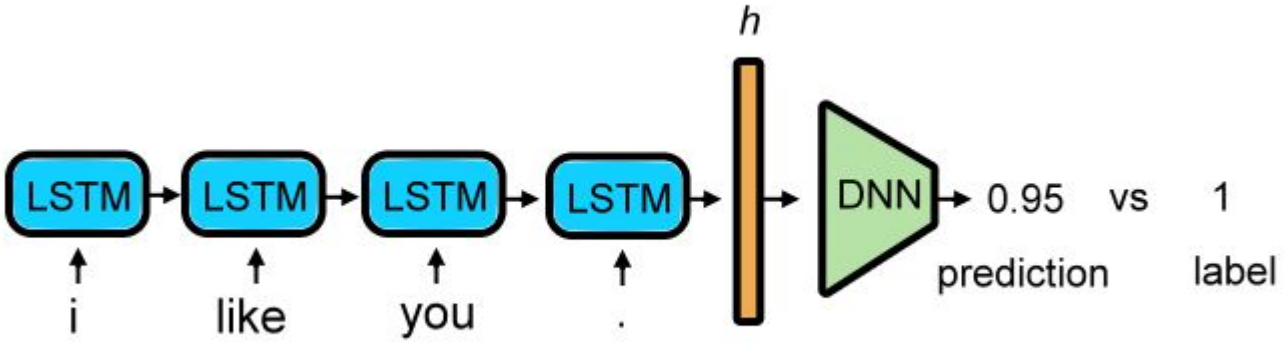
1. Task Introduction
2. Implement TIme
3. Data Preprocessing & Word Embedding
4. Demo
5. Improvement Tips

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 .
```



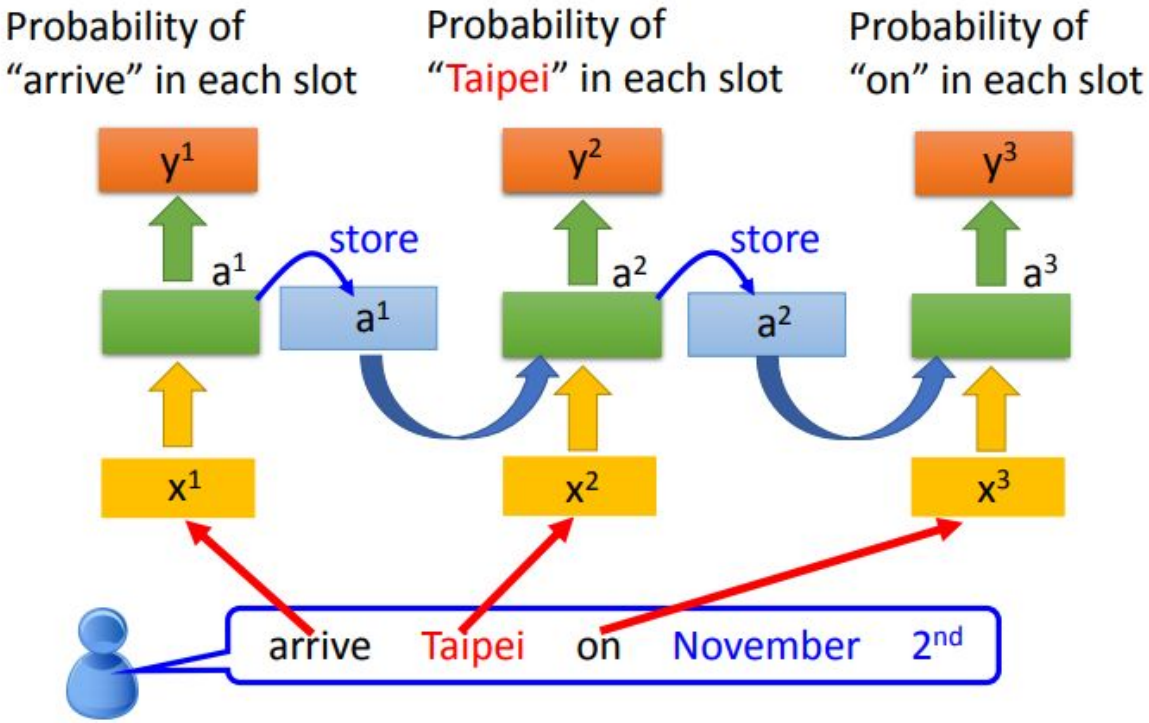
Kahoot!

https://kahoot.it/

PIN: 7120375

RNN

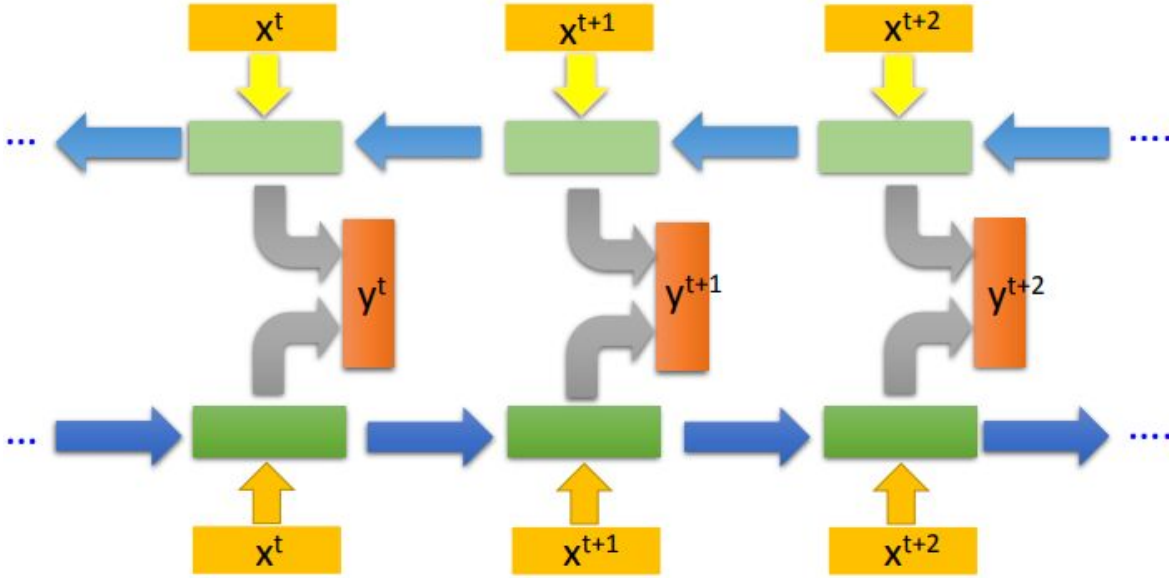
The same network is used again and again.



李宏毅教授RNN投影片

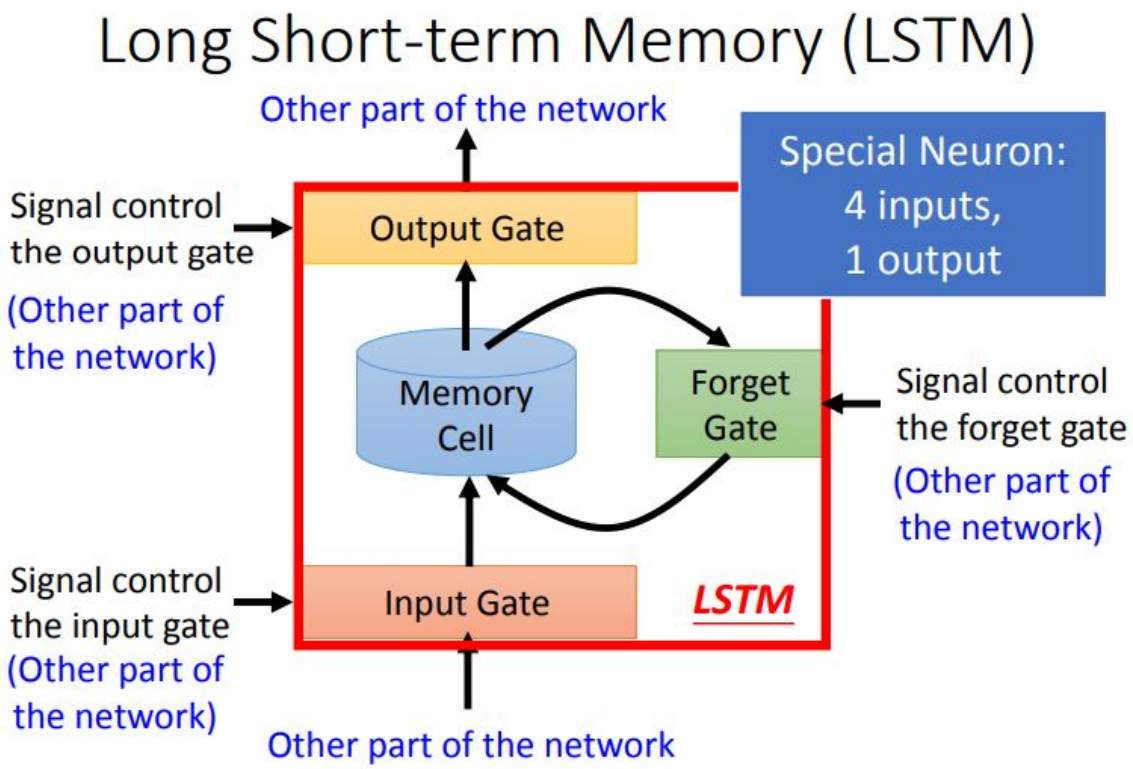
http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2017/Lecture/RNN.pdf

Bidirectional RNN



李宏毅教授RNN投影片

http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2017/Lecture/RNN.pdf



李宏毅教授RNN投影片

http://speech.ee.ntu.edu.tw/~tlkagk/courses/ML_2017/Lecture/RNN.pdf

Text Sentiment Classification

本堂實作的Dataset為twitter上收集的推文，每則都會被標注正面或負面，如：

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

1：正面

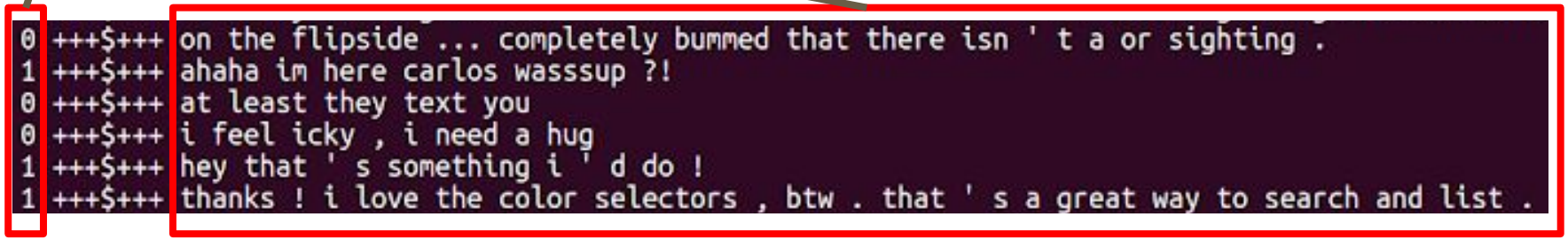
```
0 +++$+++ i feel icky , i need a hug
```

0：負面

- training data　　：20萬
- testing data　　：20萬

Data Format (labeled data)

label +++$+++ text



Data Preprocessing & Word Embedding

Text Preprocessing

```
why is it raining i need to move furniture and boxes today
writting my heart out ... because i am happy
i ' m glad everyone ' s ok . sorry 4 long day
sunshine !!!!!! woo hoo !!! have an awesome day guys off to sunbathe
sending lots of love way . losing loved ones is hard ...
another argument with my ex wife
i hate gettin woke up early if u can class 11 : 30 as early that is
why is everyone sleeping
im sleepy ... looking forward to starting my new job tomorrow hehe
triclops custom man 3 faces now online cheers n beers
```

Preprocess the sentences

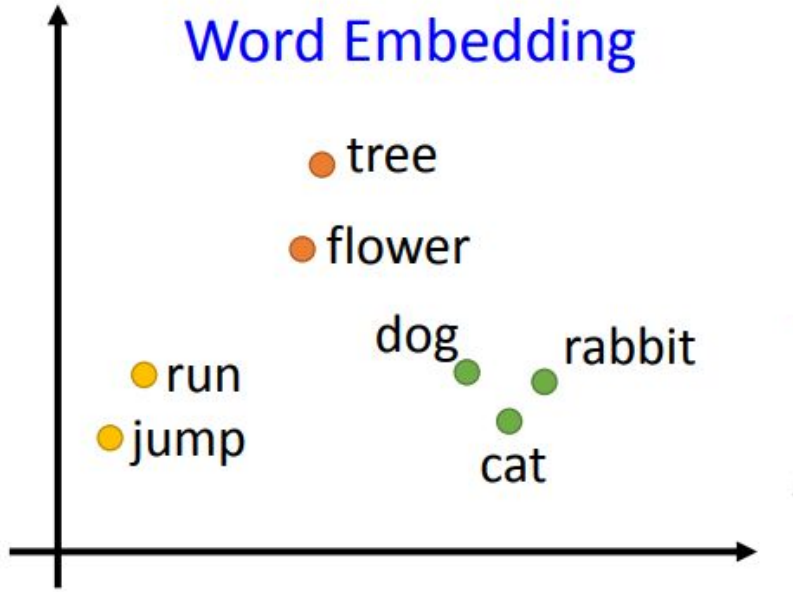
- 先建立字典, 字典內含有每一個字所對應到的index
  example:
  “I have a pen.” -> [1, 2, 3, 4]
  “I have an apple.” -> [1, 2, 5, 6]

- 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(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 [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 :
  a. 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)
  b. 很吃記憶體

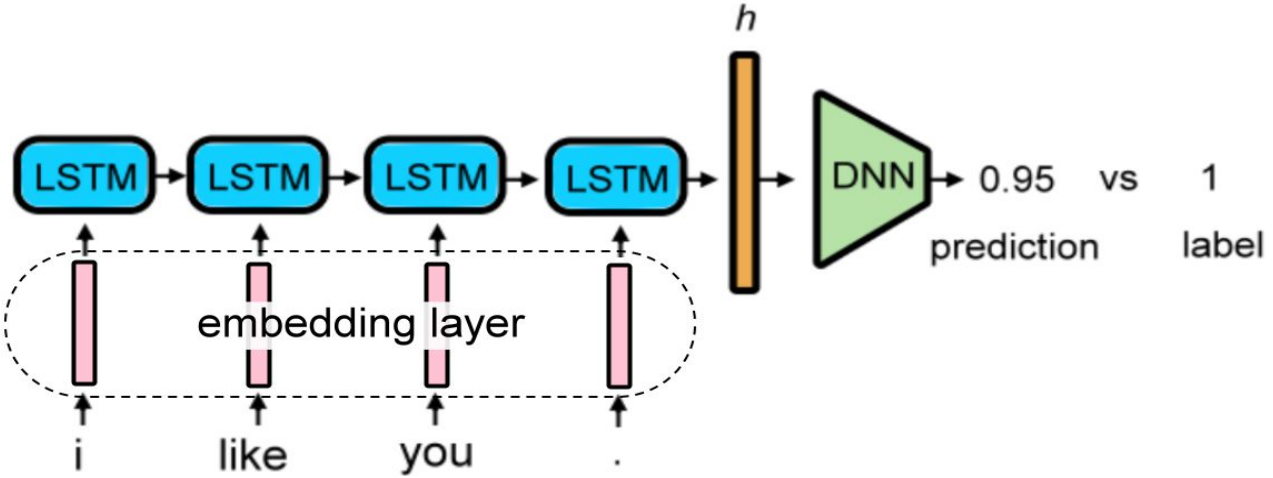
200000(data)*30(length)*20000(vocab size) *4(Byte) = 4.8*10^11 = 480 GB

Word Embedding(*)

1. 用一些方法pretrain 出word embedding (ex：skip-gram、CBOW)

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

2. 跟model的其他部分一起train (比較輕鬆)



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.

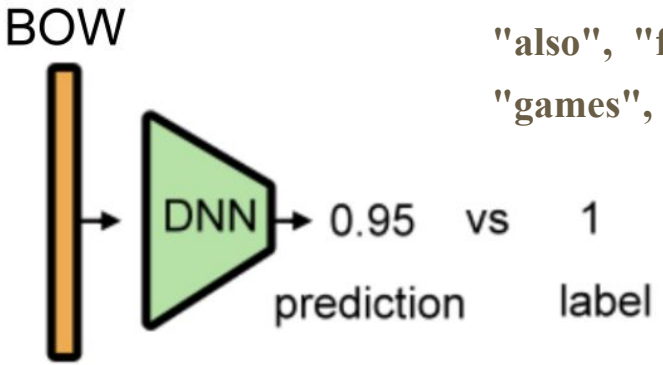
在BOW的表示方法下, 會變成:

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

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

dictionary

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



Implement Time

Code

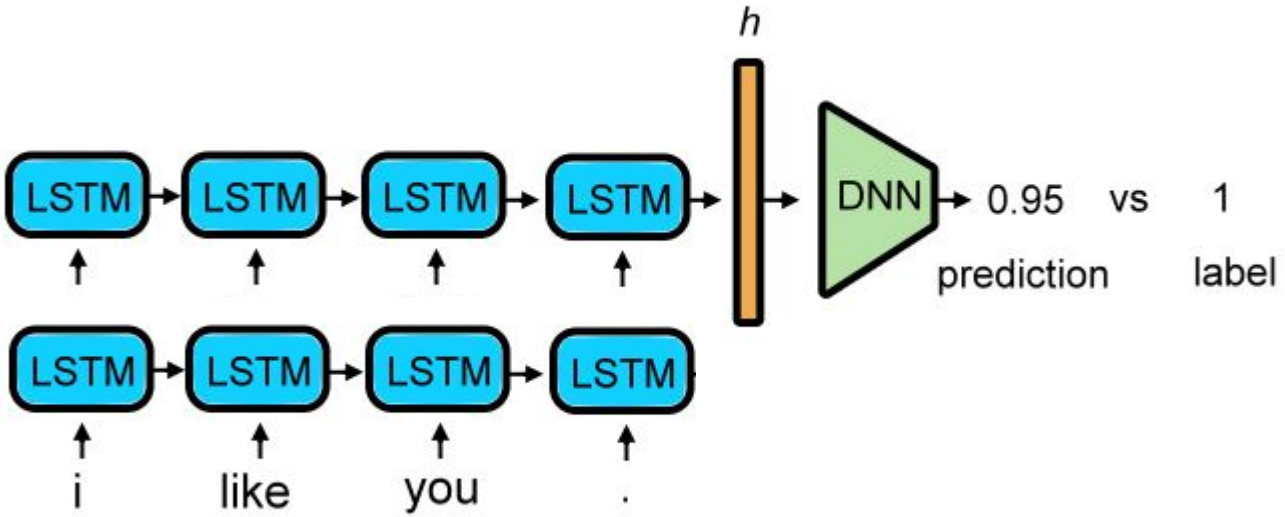
- Link
- File
  - main.py：主程式
  - answer.py：範例解答
  - data/：
    - training_data.txt
    - testing_data.csv
  - params/：存 model

Implement Time

1. Define model architecture
   a. Add embedding layer
   b. Add RNN layer
   c. Add bidirectional wrapper
   d. Add dense layer & output layer
2. Compile model
3. Start training

```
def train(tr_data, tr_label, word_indexing
    # Model
    model = Sequential()
    tr_data = np.array(tr_data)
    tr_label = np.array(tr_label)

    # Set check point to early stop and sa
    check = ModelCheckpoint('SampleSolutio

    # TODO, 1-a
    # Embed the words

    # TODO, 1-b~d
    # Define model architecture

    # TODO, 2
    # Compile model

    # TODO, 3
    # Start training
```

Define model architecture



Define model architecture - Embedding Layer

- Embedding
  - English Doc
  - 中文文檔

```
# TODO, 1-a
# Embed the words
model.add(Embedding(word_indexing,256,trainable=True))
```

- input_dim = word_indexing：總字彙有多少個
- output_dim = 256：embedding vector 的長度
- trainable：此 layer 是不是可 train 的

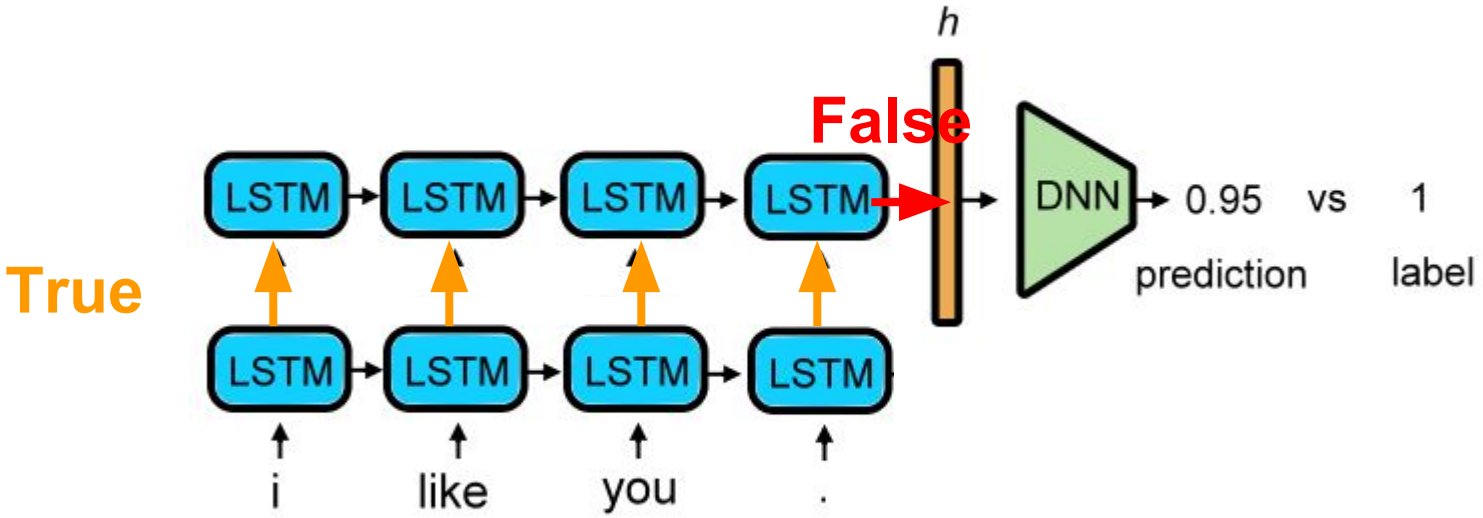
Define model architecture - Recurrent Layers

- Recurrent Layers
  - English Doc
  - 中文文檔

```
# TODO, 1-b~d
# Define model architecture
model.add(SimpleRNN(128,return_sequences=True))
model.add(SimpleRNN(128,return_sequences=False))
```

- output_dim = 128 : output vector 的長度
- return_sequences : 接給下一層的 layer，取所有 RNN unit 的 output 或是只取最後一個

Define model architecture - Recurrent Layers



Define model architecture - Bidirectional

- Bidirectional Layer Wrapper
  - English Doc
  - 中文文檔

```
model.add(Bidirectional(SimpleRNN(128,return_sequences=True)))
model.add(Bidirectional(SimpleRNN(128,return_sequences=False)))
```

- 只能使用在 recurrent layer

Define model architecture - Other

```
model.add(Dropout(0.5))
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(1,activation='sigmoid'))
```

Compile model & Training

```
# TODO, 2
# Compile model
model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])
# TODO, 3
# Start training
model.fit(tr_data,tr_label,batch_size=512,epochs=5,
          callbacks=[check],validation_split=0.1)
```

Demo

Interactive Demo

- python demo.py

```
Please input some words.
i love you !
The predicted value is [0.9068892]
The sentiment of input sentence is positive :)
```

- Ctrl + C to quit the program

Improvement Tips

Improvement Tips

- Tune the hyperparameters
  - embedding dimension
  - RNN output dimension
- Text preprocessing
  - tokenizer
  - stemmer
- Try pretrained embeddings
  - Glove
  - Fasttext
- Try different recurrent layers
  - LSTM
  - GRU
  - CuDNNGRU

DEEEEEEEEEEEEEEEPER