Assignment #4

RNN/LSTM

한양대학교 컴퓨터소프트웨어학부 2015004911 임성준
딥러닝및응용(ITE4053, 12878) - 최용석 교수님

---

1. 개요

RNN과 LSTM 모델을 사용하여 input 값을 원하는 output 값으로 출력되도록 한다.

2. 개발환경

```
OS : macOS Big Sur 11.2.2
Language : Python 3.7.10
Source code editor : Visual Studio Code
Runtime environment : Google Colaboratory
```

3. 코드 설명

- RNN/LSTM.py

```python
import tensorflow as tf
import numpy as np

tf.set_random_seed(777) # reproducibility

sample = " if you want you"
idx2char = list(set(sample)) # index -> char
char2idx = {c: i for i, c in enumerate(idx2char)} # char -> index

# hyper parameters
dic_size = len(char2idx) # RNN iput size (one hot size)
hidden_size = len(char2idx) # RNN output size
num_classes = len(char2idx) # final output size (RNN or softmax, etc.)
batch_size = 1 # one sample data, one batch
sequence_length = len(sample) - 1 # number of LSTM rollings (unit #)
learning_rate = 0.1

# data
sample_idx = [char2idx[c] for c in sample] # char to index
x_data = [sample_idx[:-1]] # X data sample (0 ~ n-1) "if you wnant yo"
y_data = [sample_idx[1:]] # Y label sample (1 ~ n) "f you want you"
```

```
X = tf.placeholder(tf.int32, [None, sequence_length])
Y = tf.placeholder(tf.int32, [None, sequence_length])

x_one_hot = tf.one_hot(X, num_classes)
cell = tf.contrib.rnn.BasicRNNCell(num_units=hidden_size)
# cell = tf.contrib.rnn.BasicLSTMCell(num_units=hidden_size, state_is_tuple=-
True)
initial_state = cell.zero_state(batch_size, tf.float32)
outputs, _states = tf.nn.dynamic_rnn(cell, x_one_hot,
initial_state=initial_state, dtype=tf.float32)

# FC layer
X_for_fc = tf.reshape(outputs, [-1, hidden_size])
outputs = tf.contrib.layers.fully_connected(X_for_fc, num_classes,
activation_fn=None)

# reshape out for sequence_loss
outputs = tf.reshape(outputs, [batch_size, sequence_length, num_classes])

weights = tf.ones([batch_size, sequence_length])
sequence_loss = tf.contrib.seq2seq.sequence_loss(logits=outputs, targets=Y,
weights=weights)
loss = tf.reduce_mean(sequence_loss)
train = tf.train.AdamOptimizer(learning_rate=learning_rate).minimize(loss)

prediction = tf.argmax(outputs, axis=2)

with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())
    for i in range(50):
        l, _ = sess.run([loss, train], feed_dict={X: x_data, Y: y_data})
        result = sess.run(prediction, feed_dict={X: x_data})

        # print char using dic
        result_str = [idx2char[c] for c in np.squeeze(result)]

        print(i, "loss: ", l, "prediction: ", "".join(result_str))
```

- 주어진 문장의 [:-1] 부분을 input으로, [1:]부분을 예측한다.
- RNN과 LSTM을 사용해서 구현한다.
- "if you want you" 와 " if you want you" 두 가지 sample을 사용했다.
- 주어진 data를 encoding 후, cell에서 처리했다.
- Adam optimizer를 사용한다.

4. 실행 결과

RNN과 LSTM 모두 정확하게 같은 출력값을 보인다.

- sample = "if you want you"

```
0 loss:  2.274967 prediction:  uyyou oattttoo
1 loss:  2.0770009 prediction:  uyyou o  ooooo
2 loss:  1.8386259 prediction:   yyou   nttoou
3 loss:  1.5304035 prediction:   yyou   nttyou
4 loss:  1.214165 prediction:   yyou  ant you
5 loss:  0.9351304 prediction:   yyou  ant you
6 loss:  0.71001726 prediction:  f you  ant you
7 loss:  0.51931524 prediction:  f you  ant you
8 loss:  0.3605241 prediction:  f you want you
9 loss:  0.24435726 prediction:  f you want you
10 loss:  0.16603324 prediction:  f you want you
11 loss:  0.11053302 prediction:  f you want you
12 loss:  0.07431452 prediction:  f you want you
13 loss:  0.04994091 prediction:  f you want you
14 loss:  0.034032226 prediction:  f you want you
15 loss:  0.023550663 prediction:  f you want you
16 loss:  0.016359476 prediction:  f you want you
17 loss:  0.011549553 prediction:  f you want you
18 loss:  0.008419646 prediction:  f you want you
19 loss:  0.006379361 prediction:  f you want you
20 loss:  0.0050181653 prediction:  f you want you
21 loss:  0.0040789405 prediction:  f you want you
22 loss:  0.0034053712 prediction:  f you want you
23 loss:  0.0029022188 prediction:  f you want you
24 loss:  0.0025107723 prediction:  f you want you
25 loss:  0.0021947806 prediction:  f you want you
26 loss:  0.0019318929 prediction:  f you want you
27 loss:  0.0017088208 prediction:  f you want you
28 loss:  0.0015174948 prediction:  f you want you
29 loss:  0.0013526531 prediction:  f you want you
30 loss:  0.0012108095 prediction:  f you want you
31 loss:  0.0010889218 prediction:  f you want you
32 loss:  0.000984424 prediction:  f you want you
33 loss:  0.00089483405 prediction:  f you want you
34 loss:  0.00081792346 prediction:  f you want you
35 loss:  0.00075179327 prediction:  f you want you
36 loss:  0.0006947554 prediction:  f you want you
37 loss:  0.0006453162 prediction:  f you want you
38 loss:  0.00060233794 prediction:  f you want you
39 loss:  0.00056476746 prediction:  f you want you
40 loss:  0.00053184025 prediction:  f you want you
41 loss:  0.0005027231 prediction:  f you want you
42 loss:  0.0004770424 prediction:  f you want you
43 loss:  0.0004542709 prediction:  f you want you
44 loss:  0.00043397504 prediction:  f you want you
45 loss:  0.00041584866 prediction:  f you want you
46 loss:  0.00039956853 prediction:  f you want you
47 loss:  0.00038492202 prediction:  f you want you
48 loss:  0.0003717561 prediction:  f you want you
49 loss:  0.00035983248 prediction:  f you want you
```

- 실습 pdf에서 요구한 "if you want yo" 로 "f you want you" 예측하기
- 정확한 output이 출력되었다.

- sample = " if you want you"

```
0 loss:  2.3089805 prediction:  y
1 loss:  2.1492596 prediction:  y
2 loss:  2.0003135 prediction:  y   ou      yuu
3 loss:  1.8183833 prediction:  y  you    t yuu
4 loss:  1.5715853 prediction:  yy you  ttt you
5 loss:  1.3102504 prediction:  yy you wttt you
6 loss:  1.0251156 prediction:  yy you wttt you
7 loss:  0.7742169 prediction:  yf you want you
8 loss:  0.5734252 prediction:  yf you want you
9 loss:  0.41730657 prediction:  yf you want you
10 loss:  0.3060146 prediction:  yf you want you
11 loss:  0.22389343 prediction:  if you want you
12 loss:  0.1643058 prediction:  if you want you
13 loss:  0.119095206 prediction:  if you want you
14 loss:  0.085835606 prediction:  if you want you
15 loss:  0.061720874 prediction:  if you want you
16 loss:  0.044055592 prediction:  if you want you
17 loss:  0.03144555 prediction:  if you want you
18 loss:  0.022591257 prediction:  if you want you
19 loss:  0.016445149 prediction:  if you want you
20 loss:  0.012414097 prediction:  if you want you
21 loss:  0.009854467 prediction:  if you want you
22 loss:  0.008054642 prediction:  if you want you
23 loss:  0.0066304756 prediction:  if you want you
24 loss:  0.0055056047 prediction:  if you want you
25 loss:  0.0046299202 prediction:  if you want you
26 loss:  0.003938922 prediction:  if you want you
27 loss:  0.0033854397 prediction:  if you want you
28 loss:  0.0029389265 prediction:  if you want you
29 loss:  0.0025772941 prediction:  if you want you
30 loss:  0.002283073 prediction:  if you want you
31 loss:  0.0020420568 prediction:  if you want you
32 loss:  0.0018428059 prediction:  if you want you
33 loss:  0.0016763477 prediction:  if you want you
34 loss:  0.0015357841 prediction:  if you want you
35 loss:  0.0014157841 prediction:  if you want you
36 loss:  0.0013123923 prediction:  if you want you
37 loss:  0.001222509 prediction:  if you want you
38 loss:  0.0011436873 prediction:  if you want you
39 loss:  0.0010741347 prediction:  if you want you
40 loss:  0.0010123817 prediction:  if you want you
41 loss:  0.000957258 prediction:  if you want you
42 loss:  0.00090793334 prediction:  if you want you
43 loss:  0.00086360093 prediction:  if you want you
44 loss:  0.00082366704 prediction:  if you want you
45 loss:  0.0007876887 prediction:  if you want you
46 loss:  0.0007552542 prediction:  if you want you
47 loss:  0.0007260151 prediction:  if you want you
48 loss:  0.0006996705 prediction:  if you want you
49 loss:  0.0006758242 prediction:  if you want you
```

- 실습 pdf의 예시 코드에 나온 " if you want you" sample을 사용하였다.
- 정확하게 "if you want you" 문장이 output으로 출력된다.