Assignment 4

(Deep learning Methods and Applications)

2016025678 이강민

1. Code Description

```
3  import tensorflow as tf
4  import numpy as np
5
6  tf.set_random_seed(777)
7
8  sample = "if you want you"
9  idx2char = list(set(sample))
10 char2idx = {c: i for i, c in enumerate(idx2char)}
11
12 dic_size = len(char2idx)
13 hidden_size = len(char2idx)
14 num_classes = len(char2idx)
15 batch_size = 1
16 sequence_length = len(sample) - 1
17 learning_rate = 0.1
18
19 sample_idx = [char2idx[c] for c in sample]
20 x_data = [sample_idx[:-1]]
21 y_data = [sample_idx[1:]]
22
23 X= tf.placeholder(tf.int32, [None, sequence_length])
24 Y= tf.placeholder(tf.int32, [None, sequence_length])
25
26 x_one_hot = tf.one_hot(X, num_classes)
27 cell = tf.contrib.rnn.BasicLSTMCell(num_units = hidden_size, state_is_tuple = True)
28 initial_state = cell.zero_state(batch_size, tf.float32)
29 outputs, _states = tf.nn.dynamic_rnn(cell, x_one_hot, initial_state = initial_state, dtype = tf.float32)
30
31 X_for_fc = tf.reshape(outputs,[-1,hidden_size])
32 outputs = tf.contrib.layers.fully_connected(X_for_fc, num_classes, activation_fn =None)
33
34 outputs = tf.reshape(outputs, [batch_size, sequence_length, num_classes])
35
36 weights = tf.ones([batch_size, sequence_length])
37 sequence_loss = tf.contrib.seq2seq.sequence_loss(logits=outputs, targets=Y, weights=weights)
38 loss = tf.reduce_mean(sequence_loss)
39 train = tf.train.AdamOptimizer(learning_rate = learning_rate).minimize(loss)
40
41 prediction = tf.argmax(outputs, axis=2)
```

- line 8~10: one hot encoding을 위해 sample을 index로 바꿉니다.
- line 12~17: hyper parameter들을 초기화시켜줍니다.
- line 19~24: x data와 y label의 기본 설정을 해줍니다.
- line 26: 실제 one hot encoding으로 만들어줍니다.
- line 27~29: rnn을 LSTM 모델의 구조로 만들어줍니다.
- line 31~41: full connected nn 형태로 바꾸어줍니다.

```
43 with tf.Session() as sess:
44     sess.run(tf.global_variables_initializer())
45     for i in range(50):
46         l, _ = sess.run([loss,train], feed_dict={X:x_data, Y:y_data})
47         result = sess.run(prediction, feed_dict={X: x_data})
48 
49         result_str = [idx2char[c] for c in np.squeeze(result)]
50 
51         print(i,"loss:", l, "Prediction:", "".join(result_str))
```

- line 43~44: 학습을 시작하기 전에 그래프를 생성하고 모든 변수를 초기화시켜줍니다.
- line 45: 총 50번의 학습을 시켜줍니다.
- line 46~51: loss 값과 예측한 값을 출력합니다.

2. Result

```
0 loss: 2.3007486 Prediction:
1 loss: 2.1012025 Prediction:
2 loss: 1.8986545 Prediction:   you   t  yoo
3 loss: 1.6297412 Prediction:   youwwttt you
4 loss: 1.3573581 Prediction:   youwwntt yyu
5 loss: 1.1427932 Prediction:   youuwant you
6 loss: 0.91711634 Prediction:   youuwant you
7 loss: 0.72969556 Prediction:   you want you
8 loss: 0.5345819 Prediction:   you want you
9 loss: 0.39815772 Prediction: f you want you
10 loss: 0.3101196 Prediction: f you want you
11 loss: 0.22676253 Prediction: f you want you
12 loss: 0.1557624 Prediction: f you want you
13 loss: 0.10827792 Prediction: f you want you
14 loss: 0.07745661 Prediction: f you want you
15 loss: 0.05601068 Prediction: f you want you
16 loss: 0.040900405 Prediction: f you want you
17 loss: 0.030539095 Prediction: f you want you
18 loss: 0.023384057 Prediction: f you want you
19 loss: 0.018283805 Prediction: f you want you
20 loss: 0.014530649 Prediction: f you want you
21 loss: 0.011699063 Prediction: f you want you
22 loss: 0.009526353 Prediction: f you want you
23 loss: 0.007840754 Prediction: f you want you
24 loss: 0.0065221544 Prediction: f you want you
25 loss: 0.0054831044 Prediction: f you want you
26 loss: 0.0046585477 Prediction: f you want you
27 loss: 0.003999713 Prediction: f you want you
28 loss: 0.0034695128 Prediction: f you want you
29 loss: 0.0030397745 Prediction: f you want you
30 loss: 0.0026887257 Prediction: f you want you
31 loss: 0.0023996055 Prediction: f you want you
32 loss: 0.0021595447 Prediction: f you want you
33 loss: 0.0019585255 Prediction: f you want you
34 loss: 0.0017887997 Prediction: f you want you
35 loss: 0.001644414 Prediction: f you want you
36 loss: 0.0015205698 Prediction: f you want you
37 loss: 0.001413728 Prediction: f you want you
38 loss: 0.001320881 Prediction: f you want you
39 loss: 0.0012396635 Prediction: f you want you
40 loss: 0.0011683038 Prediction: f you want you
41 loss: 0.0011052497 Prediction: f you want you
42 loss: 0.0010492797 Prediction: f you want you
43 loss: 0.0009994007 Prediction: f you want you
44 loss: 0.00095474697 Prediction: f you want you
45 loss: 0.0009146307 Prediction: f you want you
46 loss: 0.00087850017 Prediction: f you want you
47 loss: 0.0008458118 Prediction: f you want you
48 loss: 0.0008162173 Prediction: f you want you
49 loss: 0.0007893005 Prediction: f you want you
```

```
0 loss: 2.3240447 Prediction: u oo  y yoooo
1 loss: 2.1607463 Prediction: u   u
2 loss: 1.9850831 Prediction: u   u        u
3 loss: 1.78649 Prediction:   you wyn  oou
4 loss: 1.5079845 Prediction:   you want you
5 loss: 1.2053307 Prediction:   you want you
6 loss: 0.8995487 Prediction:   you want you
7 loss: 0.6486473 Prediction:   you want you
8 loss: 0.43540272 Prediction: f you want you
9 loss: 0.2856551 Prediction: f you want you
10 loss: 0.1813825 Prediction: f you want you
11 loss: 0.11245812 Prediction: f you want you
12 loss: 0.06930728 Prediction: f you want you
13 loss: 0.04351421 Prediction: f you want you
14 loss: 0.028416405 Prediction: f you want you
15 loss: 0.019372605 Prediction: f you want you
16 loss: 0.013700047 Prediction: f you want you
17 loss: 0.010003263 Prediction: f you want you
18 loss: 0.007525773 Prediction: f you want you
19 loss: 0.0058193337 Prediction: f you want you
20 loss: 0.0046087187 Prediction: f you want you
21 loss: 0.0037247317 Prediction: f you want you
22 loss: 0.0030621954 Prediction: f you want you
23 loss: 0.0025547026 Prediction: f you want you
24 loss: 0.0021588083 Prediction: f you want you
25 loss: 0.0018452136 Prediction: f you want you
26 loss: 0.0015935426 Prediction: f you want you
27 loss: 0.0013892065 Prediction: f you want you
28 loss: 0.0012213562 Prediction: f you want you
29 loss: 0.0010821531 Prediction: f you want you
30 loss: 0.00096561905 Prediction: f you want you
31 loss: 0.0008672412 Prediction: f you want you
32 loss: 0.000783684 Prediction: f you want you
33 loss: 0.00071229035 Prediction: f you want you
34 loss: 0.0006509623 Prediction: f you want you
35 loss: 0.0005981543 Prediction: f you want you
36 loss: 0.00055256626 Prediction: f you want you
37 loss: 0.00051302527 Prediction: f you want you
38 loss: 0.00047877507 Prediction: f you want you
39 loss: 0.00044892286 Prediction: f you want you
40 loss: 0.00042290747 Prediction: f you want you
41 loss: 0.0004001763 Prediction: f you want you
42 loss: 0.0003801934 Prediction: f you want you
43 loss: 0.00036264403 Prediction: f you want you
44 loss: 0.00034716248 Prediction: f you want you
45 loss: 0.0003334339 Prediction: f you want you
46 loss: 0.00032122014 Prediction: f you want you
47 loss: 0.00031033397 Prediction: f you want you
48 loss: 0.0003005797 Prediction: f you want you
49 loss: 0.00029181264 Prediction: f you want you
```

- LSTM 모델로 잘 학습되는 것을 확인할 수 있습니다.