# Chapter. RNN/LSTM(1) 과제

컴퓨터 소프트웨어 학부

2018008059 김은수

- 환경
- 소스설명
- 결과설명

1. 환경

   - Python 3.7.4 
   - Tensorflow 1.13.1 

2. 소스 설명

   - Assignment4_2018008059.py

```
import tensorflow as tf
import numpy as np

tf.set_random_seed(777)

sample= " if you want you"
idx2char = list(set(sample))
char2idx = {c: i for i, c in enumerate(idx2char)}
```

Sample 문구는 " if you want you"다. 주의해야할 점은 if앞에 띄어쓰기가 있다. 따라서 결과로 나오는 문구는 "f you want you"가 아니라 "if you want you"여야 한다. 해당 캡처 부분은 one hot encoding을 정의하는 부분이다.

```
dic_size = len(char2idx)
hidden_size = len(char2idx)
num_classes = len(char2idx)
batch_size = 1
sequence_length = len(sample) - 1
learning_rate = 0.1

sample_idx = [char2idx[c] for c in sample]
x_data = [sample_idx[:-1]]
y_data = [sample_idx[1:]]
```

Hyper parameter들과 one hot encoding을 구현한 부분이다.

```
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)
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)
```

RNN CELL을 구성하는 부분이다. Cell을 tf.contrib.rnn.BasicRNNCell 로 구성해준다. Cell 구동을 위해 dynamic_rnn함수에 cell과, input data로 x_one_hot 데이터를 넣어준다.

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

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)
```

'classifier를 위해 Fully connected layer를 하나 정의해준다. Fully connected layer를 거쳐서 classifier를 해준다.

Reshape을 통해서 실제 sequence data로 바꿔준다. loss함수를 정의해주고, optimizer로 Adam optimizer를 사용한다.

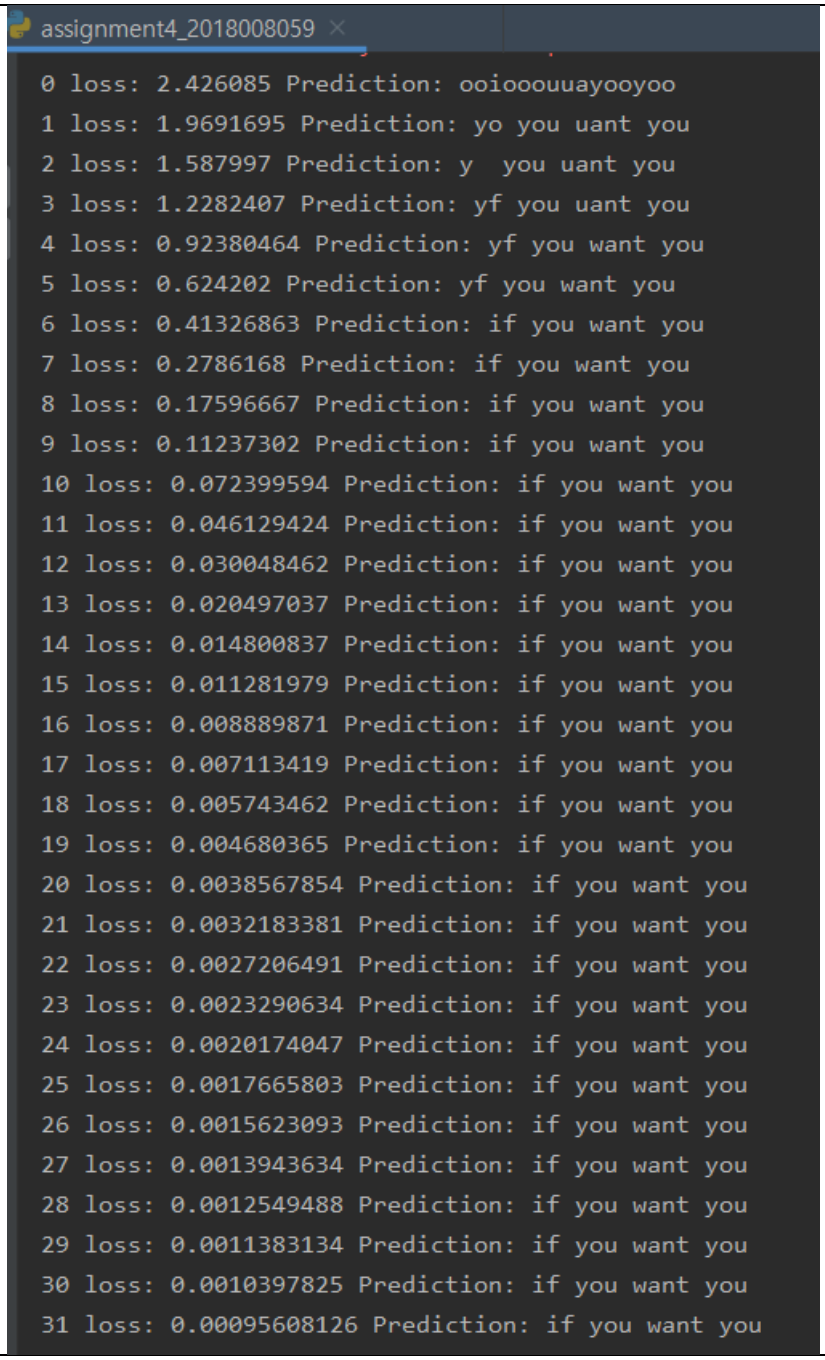
```
with tf.Session() as sess:
    sess.run(tf.global_variables_initializer())
    for i in range(50):
        l, _ = sess.run([loss, train], feed_dict={X: x_data, Y: y_data})
        result = sess.run(prediction, feed_dict={X: x_data})

        result_str = [idx2char[c] for c in np.squeeze(result)]

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

50번의 epoch을 돌려주고 몇 번째 학습인 지, loss 수치, 예측한 문구를 print한다.

3. 결과 설명

| | |
|---|---|
| 과제 참고 코드 대로 하고 (LSTM) 문구를 "if you want you"로 했을 때 | assignment4_2018008059<br>0 loss: 2.276994 Prediction: oo<br>1 loss: 2.0677304 Prediction: ou u yooooo<br>2 loss: 1.850858 Prediction: ou you<br>3 loss: 1.6005814 Prediction: you yyy yyou<br>4 loss: 1.2978673 Prediction: you want you<br>5 loss: 0.9759061 Prediction: you waat you<br>6 loss: 0.6995872 Prediction: you want you<br>7 loss: 0.4838624 Prediction: you want you<br>8 loss: 0.31832343 Prediction: you want you<br>9 loss: 0.21631865 Prediction: f you want you<br>10 loss: 0.14526173 Prediction: f you want you<br>11 loss: 0.09416981 Prediction: f you want you<br>12 loss: 0.062474493 Prediction: f you want you<br>13 loss: 0.044135045 Prediction: f you want you<br>14 loss: 0.032486256 Prediction: f you want you<br>15 loss: 0.023877332 Prediction: f you want you<br>16 loss: 0.017195357 Prediction: f you want you<br>17 loss: 0.012247814 Prediction: f you want you<br>18 loss: 0.008803914 Prediction: f you want you<br>19 loss: 0.0064855586 Prediction: f you want you<br>20 loss: 0.004927296 Prediction: f you want you<br>21 loss: 0.0038595754 Prediction: f you want you<br>22 loss: 0.0031066786 Prediction: f you want you<br>23 loss: 0.0025596167 Prediction: f you want you<br>24 loss: 0.0021510392 Prediction: f you want you<br>25 loss: 0.0018386474 Prediction: f you want you<br>26 loss: 0.0015951281 Prediction: f you want you<br>27 loss: 0.0014022173 Prediction: f you want you<br>28 loss: 0.001247219 Prediction: f you want you<br>29 loss: 0.0011211515 Prediction: f you want you<br>30 loss: 0.0010174549 Prediction: f you want you<br>31 loss: 0.00093130744 Prediction: f you want you |


➔ PPT에 참고된 실행결과처럼 9번째 epoch학습부터 prediction이 "f you want you" 가 나온다.

| | |
|---|---|
| BasicRNNCell을 사용하고 문구를 " if you want you" 로 했을 때 (띄어쓰기 有) |  |

```
32 loss: 0.00088434335 Prediction: if you want you
33 loss: 0.00082249375 Prediction: if you want you
34 loss: 0.000768861 Prediction: if you want you
35 loss: 0.00072199455 Prediction: if you want you
36 loss: 0.0006809358 Prediction: if you want you
37 loss: 0.00064467744 Prediction: if you want you
38 loss: 0.00061256933 Prediction: if you want you
39 loss: 0.0005840246 Prediction: if you want you
40 loss: 0.0005585198 Prediction: if you want you
41 loss: 0.0005356343 Prediction: if you want you
42 loss: 0.0005151065 Prediction: if you want you
43 loss: 0.0004965554 Prediction: if you want you
44 loss: 0.00047973508 Prediction: if you want you
45 loss: 0.0004644868 Prediction: if you want you
46 loss: 0.00045060416 Prediction: if you want you
47 loss: 0.0004379364 Prediction: if you want you
48 loss: 0.00042629297 Prediction: if you want you
49 loss: 0.0004156183 Prediction: if you want you

Process finished with exit code 0
```