RNN 보고서

컴퓨터소프트웨어학부
2016025996
조성진

Assignment4는 rnn을 한번 사용해보는 과제로, if you want yo라는 입력값을 넣어 f you want you를 얻어내는것이 목적이다

코드를 보면

```
tf.set_random_seed(777)

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

print(idx2char)
print(char2idx)
```

```
['o', ' ', 'i', 'f', 't', 'w', 'n', 'a', 'u', 'y']
{'o': 0, ' ': 1, 'i': 2, 'f': 3, 't': 4, 'w': 5, 'n': 6, 'a': 7, 'u': 8, 'y': 9}
```

다음 코드들은 unique한 character 만 뽑은 후 각각의 character에 index를 부여해 주는 코드이다.

```
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 parameter & data를 설정해주는 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)
outputs, _states = tf.nn.dynamic_rnn(cell, x_one_hot, dtype=tf.float32)
```

이 부분을 채워넣어야 했는데, X 와 Y는 아래의 test과정에서 변수로 대입이 되니

까 placeholder로 선언해주고, X는 one hot형식으로 임베딩해준다. 그리고 RNN의 cell을 만들고 rnn을 구동시킬 dynamic rnn을 생성한다.

```
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)
```

```
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))
```

다음은 RNN이 끝난 후 Fully connected layer을 만들고 session을 만들어 학습시키는 코드이다.

결과로

```
0 loss: 2.2016954 Prediction: f uoo you oouy
1 loss: 5.18889 Prediction: fyuf yuufyaun
2 loss: 5.78971 Prediction: f wauu woan  w
3 loss: 3.2497675 Prediction: f youu yfotuui
4 loss: 3.0601354 Prediction: f youufnntuuwn
5 loss: 2.0037267 Prediction: f yo wwant wwa
6 loss: 1.8349432 Prediction: f yt  want  wa
7 loss: 1.4904538 Prediction: f yt  onnt  on
8 loss: 1.1968056 Prediction: f yo yonn  yon
9 loss: 1.3591124 Prediction: f yooywunt you
10 loss: 0.96862257 Prediction: f yooywant ywu
```

처음에는 학습이 충분히 되지 않아 이상한 값을 얻어내다가,

```
30 loss: 0.008400154 Prediction: f you want you
31 loss: 0.0072552883 Prediction: f you want you
32 loss: 0.006492284 Prediction: f you want you
33 loss: 0.0056209317 Prediction: f you want you
34 loss: 0.0045674965 Prediction: f you want you
35 loss: 0.0037830756 Prediction: f you want you
36 loss: 0.002810052 Prediction: f you want you
37 loss: 0.0022307723 Prediction: f you want you
38 loss: 0.0019337559 Prediction: f you want you
39 loss: 0.0016178677 Prediction: f you want you
40 loss: 0.0013797384 Prediction: f you want you
41 loss: 0.0012247077 Prediction: f you want you
42 loss: 0.0011117978 Prediction: f you want you
43 loss: 0.0010168138 Prediction: f you want you
44 loss: 0.00093217066 Prediction: f you want you
45 loss: 0.0008592282 Prediction: f you want you
46 loss: 0.00079781725 Prediction: f you want you
47 loss: 0.0007446163 Prediction: f you want you
48 loss: 0.00069663994 Prediction: f you want you
49 loss: 0.000652497 Prediction: f you want you
```

일정 부분이 지나면 의도한 대로 예측하는 것을 확인할 수 있다.