시즌 1 - 딥러닝의 기본 - ML lab 12

노트북: 모두를 위한 머신러닝
만든 날짜: 2019-01-11 오후 4:53 **수정한 날짜:** 2019-01-14 오후 3:25
작성자: rr
태그: #모두를 위한, .ML lab

ML lab 12

= RNN

1. cell 생성

```
cell = tf.contrib.rnn.BasicRNNCell(num_units=hidden_size)
```

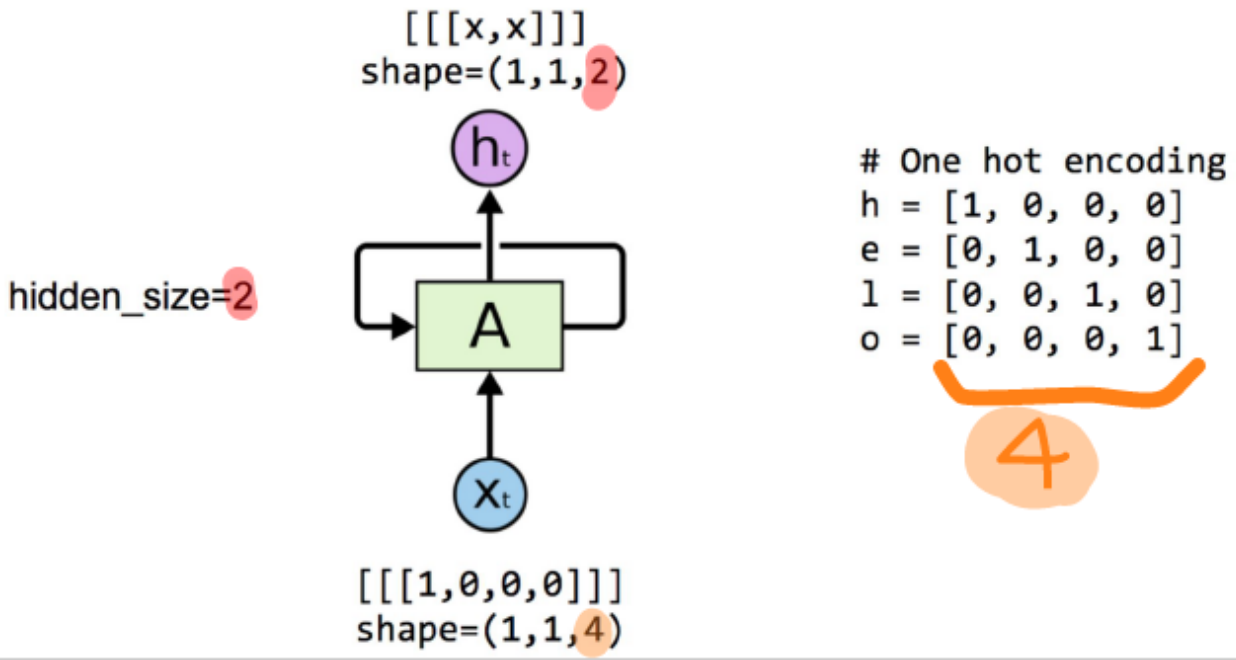
어떤 형태의 cell을 만들 것인가? BasicRNNCell, BasicLSTMCell
num_units, hidden_size: output 출력 크기

2. cell 구동

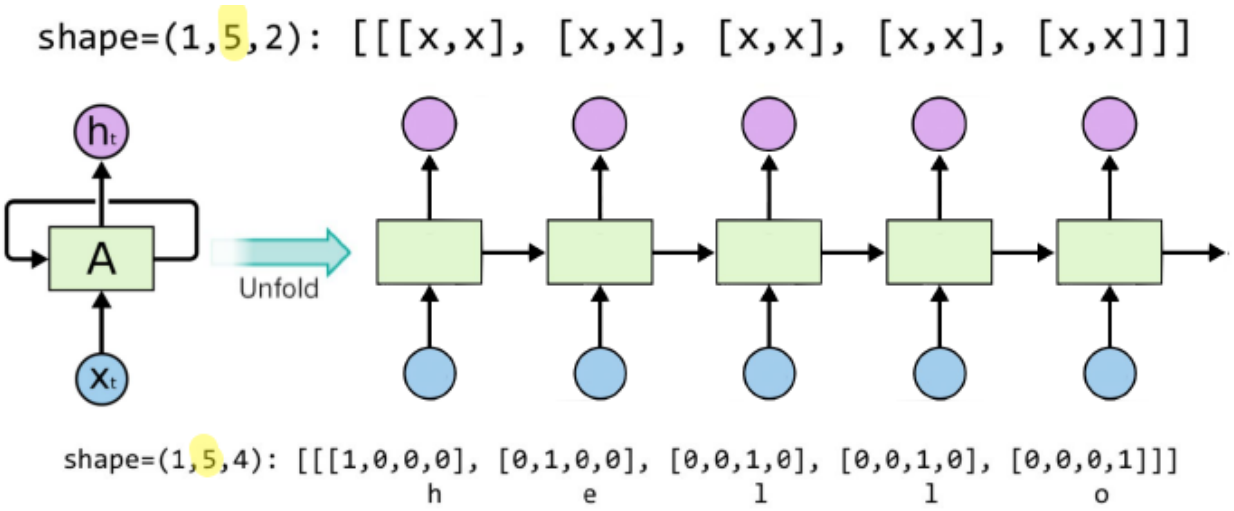
```
outputs, _states = tf.nn.dynamic_rnn(cell, x_data, dtype=tf.float32)
```

만든 cell, 원하는 입력 데이터를 넘겨줘서 두 가지 출력(outputs, _states)을 만들어냄

= One node: 4 (input-dim) in 2 (hidden_size)



= unfolding to n sequences



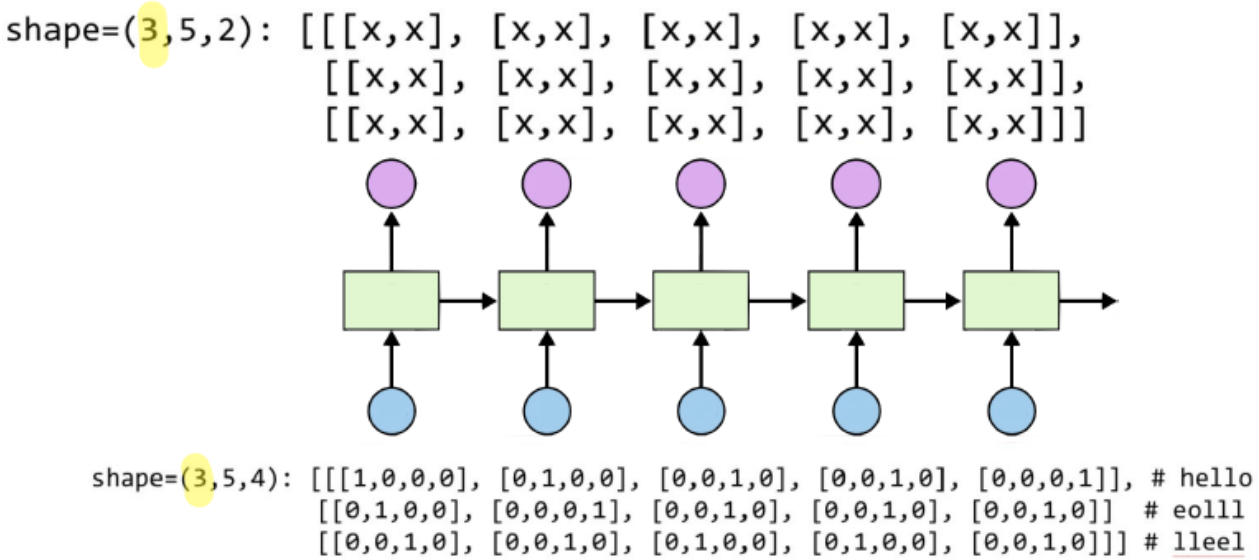
- shape=(1, 5, 4)

5: sequence_length

- shape=(1, 5, 2)

2: hidden_size

= Batching input



- shape=(3, 5, 4)

3: batch_size

= hihello RNN

- RNN parameters

```
input_dim = 5  # one-hot size
```

```
hidden_size = 5  # output from the LSTM. 5 to directly predict one-hot
batch_size = 1   # one sentence
sequence_length = 6  # |ihello| == 6
```

- Feed to RNN

```
x = tf.placeholder(tf.float32, [None, sequence_length, input_dim])
```

= RNN with long sequences

- RNN parameters

```
char_set = list(set(sentence))
char_dic = {w: i for i, w in enumerate(char_set)}

data_dim = len(char_set)
hidden_size = len(char_set)
num_classes = len(char_set)
sequence_length = 10  # Any arbitrary number

batch_size = len(dataX)
```

이전에는 데이터가 한 줄 밖에 없어서 batch_size가 1이었지만, 이제는 데이터가 많기 때문에 batch_size가 달라짐

= Stacked RNN + Softmax layer

- Stacked RNN

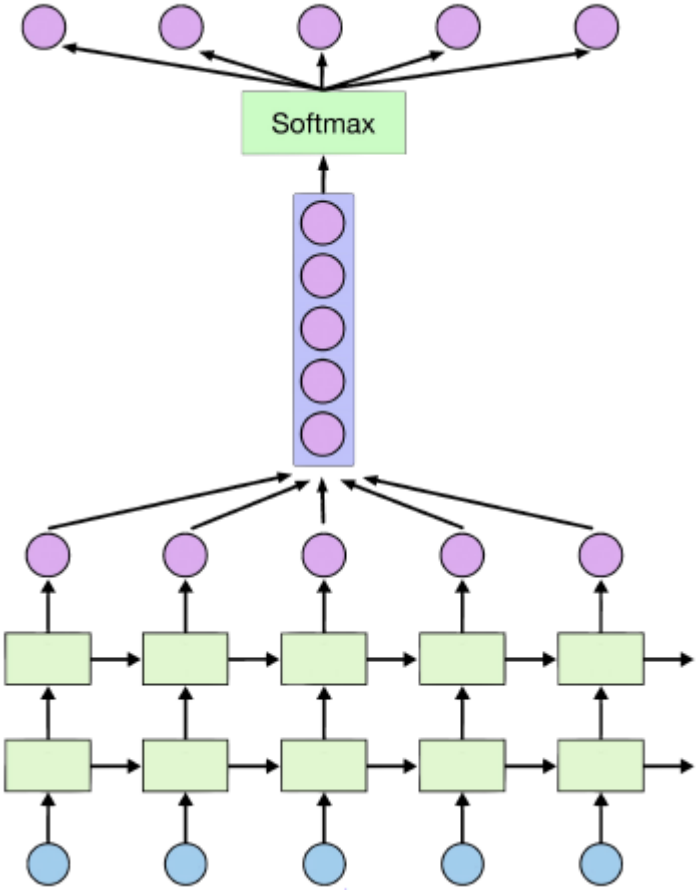
2층, 3층 ... 이상으로 쌓는 것

```
rnn.MultiRNNCell([cell] * n, state_is_tuple=True)
```

2층으로 쌓고 싶다 -> n = 2
100층으로 쌓고 싶다 -> n = 100

- Softmax layer

1. 시퀀스 개수에 맞게 softmax를 만들 필요 없이, reshape로 하나로 만들어주면 됨
2. softmax 만든 결과를 reshape로 펼쳐주기만 하면 됨




```
# 1. softmax layer
X_for_softmax = tf.reshape(outputs, [-1, hidden_size])

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

= Dynamic RNN

- 가변하는 시퀀스 길이를 가져야 함

<pad>가 있어도 weight가 있어서 값이 나옴 예측에 방해가 될 수 있음

```
sequence_length = [5, 3, 4]
```

없는 데이터는 0으로 만들어줌

= RNN with Time Series Data

- Time Series Data: 시간에 따라서 변하는 데이터

- many to one

이전 데이터들을 다 연결해서 이용함

- LSTM and Loss

```
Y_pred = tf.contrib.layers.fully_connected(outputs[:, -1], output_dim, activation_fn=None)  # We use the
last cell's output
```

LSTM -> FC -> Y-hat