

Bidirectional Recurrent Neural Networks (Bidirectional RNN)

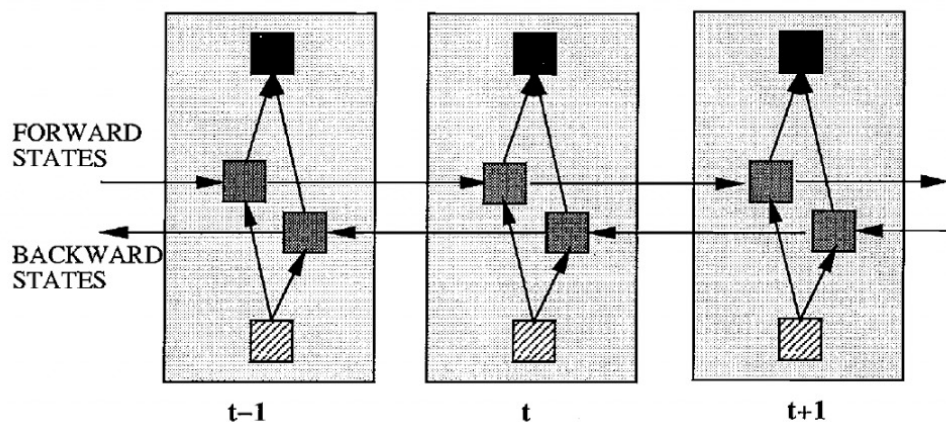
에이아이스쿨(AISchool) 대표
양진호 (솔라리스)

<http://aischool.ai>

<http://solarisailab.com>

Bidirectional Recurrent Neural Networks(RNNs)

- 이번 시간에는 이전 time-step의 상태뿐만 아니라, 다음 time-step의 상태까지 고려하는 확장된 Recurrent Neural Networks(RNNs) 형태인 Bidirectional Recurrent Neural Networks(BRNNs)에 대해 알아보자.



Unfolded Representation

Hidden Unit에서의 연산

$$\vec{h}_t = \sigma(W_{x\vec{h}}x_t + W_{\vec{h}\vec{h}}\vec{h}_{t-1} + b_{\vec{h}})$$

$$\overleftarrow{h}_t = \sigma(W_{x\overleftarrow{h}}x_t + W_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{h}_{t+1} + b_{\overleftarrow{h}})$$

Output Unit에서의 연산

$$y_t = W_{\vec{h}y}\vec{h}_t + W_{\overleftarrow{h}y}\overleftarrow{h}_t + b_y$$

Bidirectional Recurrent Neural Networks(RNNs)의 장점

- Bidirectional Recurrent Neural Networks(BRNNs)을 이용하면, 이렇게 **이전 정보**와 **이후 정보**를 모두 저장할 수 있다.
- 시계열 데이터의 현재 시간 이전 정보뿐만 아니라, **이후 정보**까지 저장해서 활용할 수 있다면 더 좋은 성능을 기대할 수 있다. 예를 들어 “푸른 하늘에 OO이 떠있다.”라는 문장에서 OO에 들어갈 단어를 예측하고자 한다면, 우리는 앞의 정보인 “푸른”, “하늘”이라는 정보를 가지고 OO에 들어갈 단어가 “구름”이라고 예측할 수도 있지만, 뒤의 정보인 “**떠있다.**”라는 단어까지 함께 고려하면 **더 높은 확률로 정답이 “구름”이라는 것을 예측할 수 있을 것이다.**

TensorFlow에서 Bidirectional RNN 구현하는 법

- tf.keras.layers.Bidirectional을 이용한 구현
- https://www.tensorflow.org/api_docs/python/tf/keras/layers/Bidirectional

```
42     model = tf.keras.Sequential([
43         tf.keras.layers.Embedding(encoder.vocab_size, 64),
44         tf.keras.layers.Bidirectional(tf.keras.layers.LSTM(64)),
45         tf.keras.layers.Dense(64, activation='relu'),
46         tf.keras.layers.Dense(1)
47     ])
```

Thank you!
