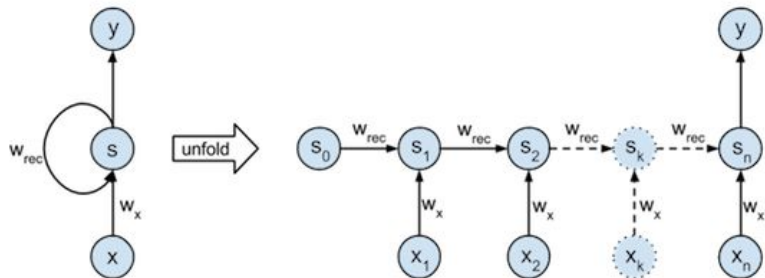


RNN_v2 concept

2022,06, April, by BeomGonYu

Current RNN



$$S_k = f(S_{k-1} \cdot W_{rec} + X_k \cdot W_x)$$

f : tanh, 활성화함수, bias는 생략

RNN의 특징

1. CNN과 비슷하게 **shared weight**를 사용함으로써 **sequence** 내에 **pattern**을 학습하는 것으로 이해할 수 있다
2. 현재 **token**의 **embedding**을 현재 **token**뿐 아니라 이전 **state**의 정보까지 포함하여 **encoding**한다.

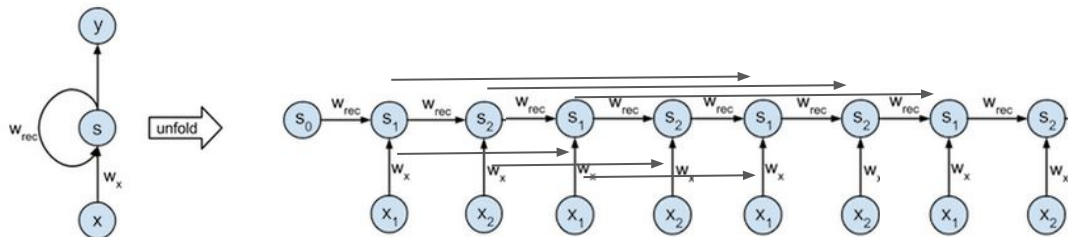
RNN의 단점

1. **sequence**가 길어질 때 앞 부분 **token**을 학습하기가 어렵다.
(difficult to learn long range dependency)

LSTM, GRU, etc

1. **vanilla RNN**의 단점을 보완하기 위해 **hidden state encoding**을 정교하게 디자인함,
2. 다양한 **sequential**한 **pattern**을 학습할 있도록 design됨

proposed RNN



Proposed RNN

1. 기존 앞뒤 token 간의 pattern 외에 추가로 2 또는 4 등 멀리 떨어진 token간 pattern을 direct로 학습
2. 현재 token의 encoding시 직전, 그리고 몇 단계 전의 token정보를 고려하여 encoding함
3. 위 그림을 예시로 보면 처음과 마지막 token간의 거리는 2단계에 지나지 않음
(long range dependency를 고려한 encoding 가능)
4. 적절한 길이의 초기값(0) token을 앞 부분에 붙여줌

$$S_k = f(S_{k-1} * W_1 + S_{k-2} * W_2 + S_{k-4} * W_4 + \dots)$$

Results

vanilla RNN 대비 1~2%의 accuracy 향상을 보임, but gru가 훨씬 더 높게 나온다.
GRU/LSTM의 경우 filter가 input에 dependent하다고 볼 수 있다.
Use dynamic filter based on input instead of fixed filter in image
dynamic filter를 갖는 customized CNN을 구현해 보자