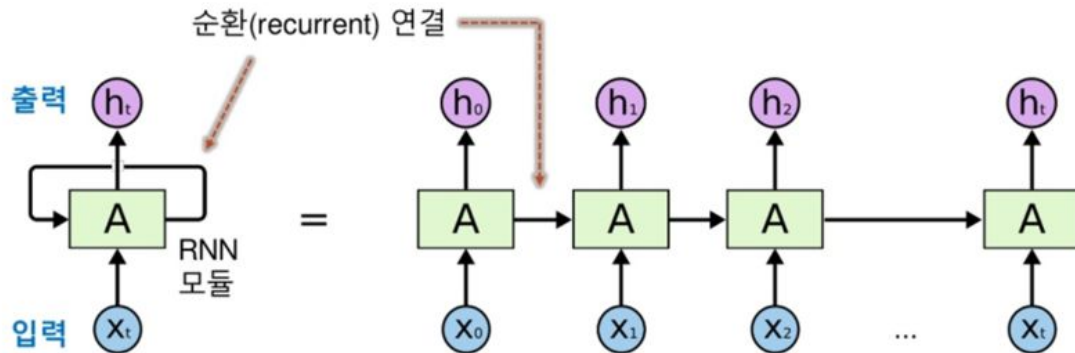


RNN(Recurrent Neural Network)
주로 시계열, 연속데이터에서 사용한다.

simpleRNN



RNN은 일반적인 인공 신경망인 FFNets(Feed-Forward Neural Networks)와 이름에서 부터 어떤 점이 다른지 드러난다. FFNets는 데이터를 입력하면 연산이 입력층에서 은닉층을 거쳐 출력층까지 차근차근 진행되고 이 과정에서 입력 데이터는 모든 노드를 딱 한번씩 지나게가 된다. 그러나 RNN은 은닉층의 결과가 다시 같은 은닉층의 입력으로 들어가도록 연결되어 있다.

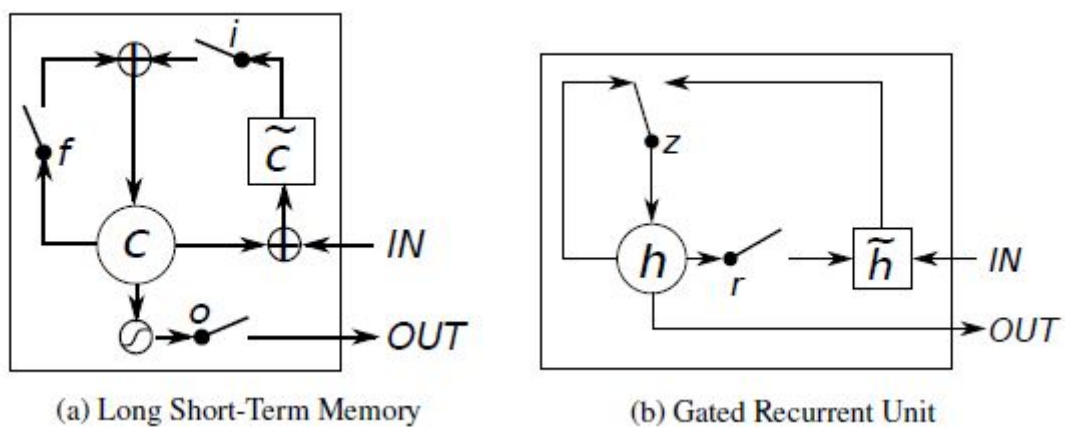
<https://ebbnflow.tistory.com/135>

파라미터 계산 공식은 $((input + 1) + output) * output$

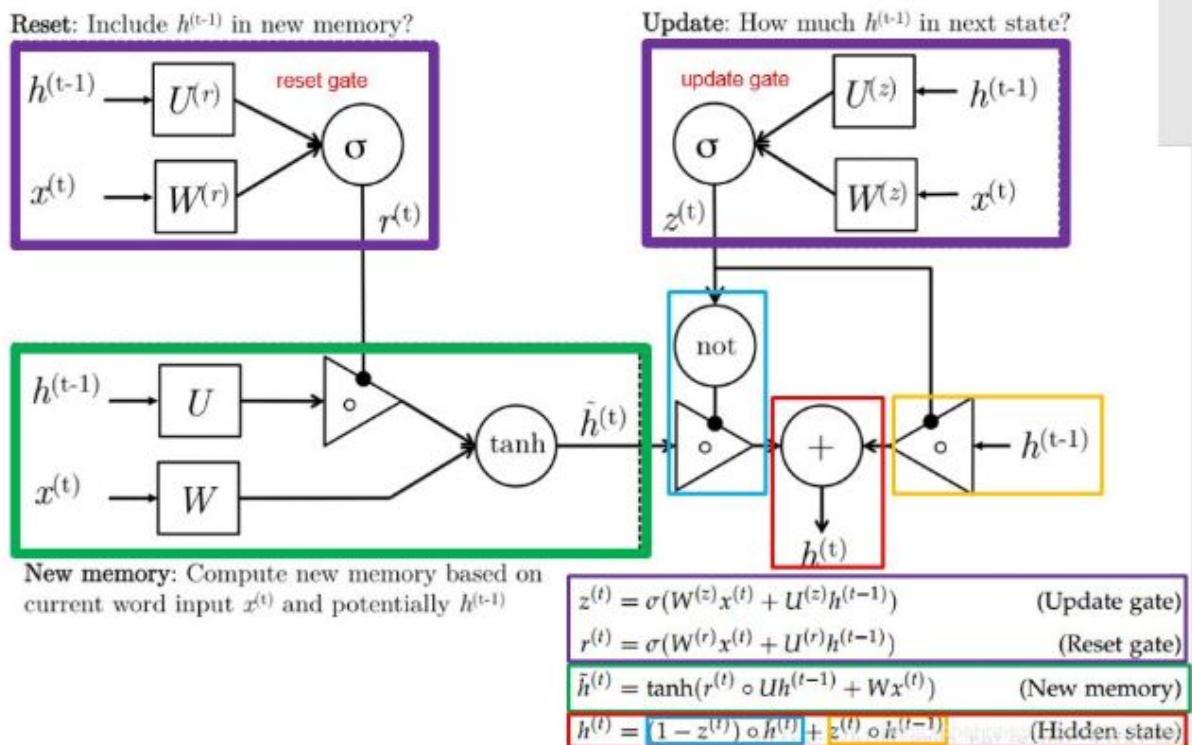
GRU

<https://www.cnblogs.com/AntonioSu/p/8798960.html>

https://www.sohu.com/a/336551522_99979179



GRU는 LSTM에서 변형한 형태이다.



1. reset gate 계산하여 임시ht를 만든다.
2. update gate를 통해 ht-1과 ht간의 비중결정
3. zt를 통해 최종ht를 계산한다.

파라미터 공식

$$3 \times (n^2 + nm + n)$$

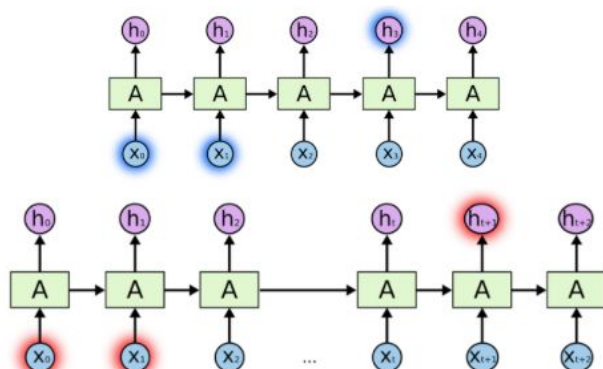
n은 아웃풋, m은 인풋 만약에 아웃풋을 10으로 인풋을 1로 설정했을때 파라미터의 수는 360개

LSTM

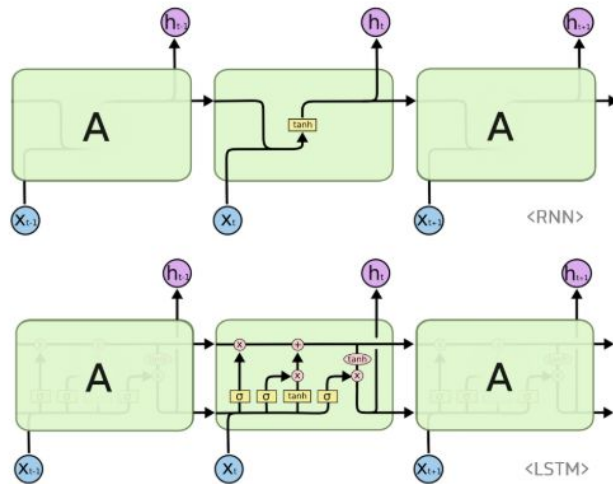
<https://dgkim5360.tistory.com/entry/understanding-long-short-term-memory-lstm-kr>

데이터 양이 방대할때 사용하기 좋음

LSTM의 기본 구조



<관련 정보와 그 정보를 사용하는 지점 사이 거리가 멀 경우 RNN 학습능력 저하>



```
params = 4 * d * ((lstm_inputs_size + 1) + d) = 4 * ((lstm_inputs_size + 1) * d + d^2)
```

```
model.add(LSTM(10, input_length = 3, input_dim = 1))
```

LSTM에서 h는 하이퍼시스 가설이라는 뜻

$h(x) = wx + b$

input_dim, 바이어스, 역전파

4 = 망각층, 인풋층, 셀층, 아웃층

<https://ratsgo.github.io/natural%20language%20processing/2017/03/09/rnnlstm/>
RNN와 LSTM(Long Short-Term Memory)에 대해 알아보기