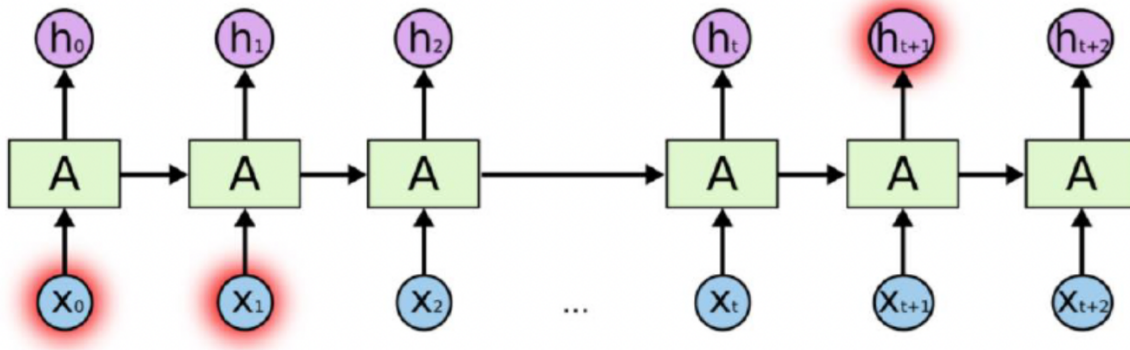


## RNN

: 순차적인 데이터를 처리하는 데 강점을 가지고 있지만, 한계점이 존재



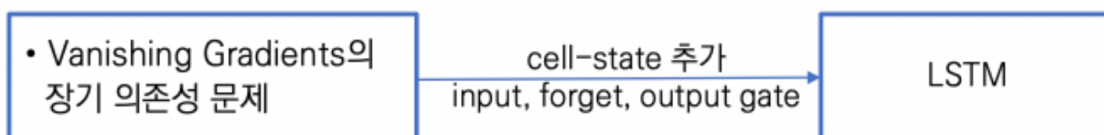
하나의 네트워크가 여러개의 복사된 형태를 띄고 있고, 각각의 네트워크는 다음 단계로 정보를 넘겨줌. 시간에 따라, 학습이 진행됨에 따라 앞에서 input으로 받은 정보가 학습에 미치는 영향이 점점 감소하다가 결국 사라져버림.

ex) “나 지금 이탈리아 여행 왔는데 건물도 화려하고 음식들도 너무 맛있어. 확실히 한국 이랑 건물들이 다르게 생겼네. 여기서 하루종일 구경하고 맛있는 거 먹기만 하면 하루가 끝나서 너무 아쉬운 것 같아. 너 \_\_\_\_\_ 가본 적 있어?”

이 문장에서 \_\_\_\_\_ 에 들어갈 단어를 예측하기 위해서는 장소 정보가 필요함. 그러나 RNN이 충분한 기억력을 가지고 있지 못한다면 다음 단어를 엉뚱하게 예측할 수 있음.

이렇게 아래와 같은 문제가 발생함

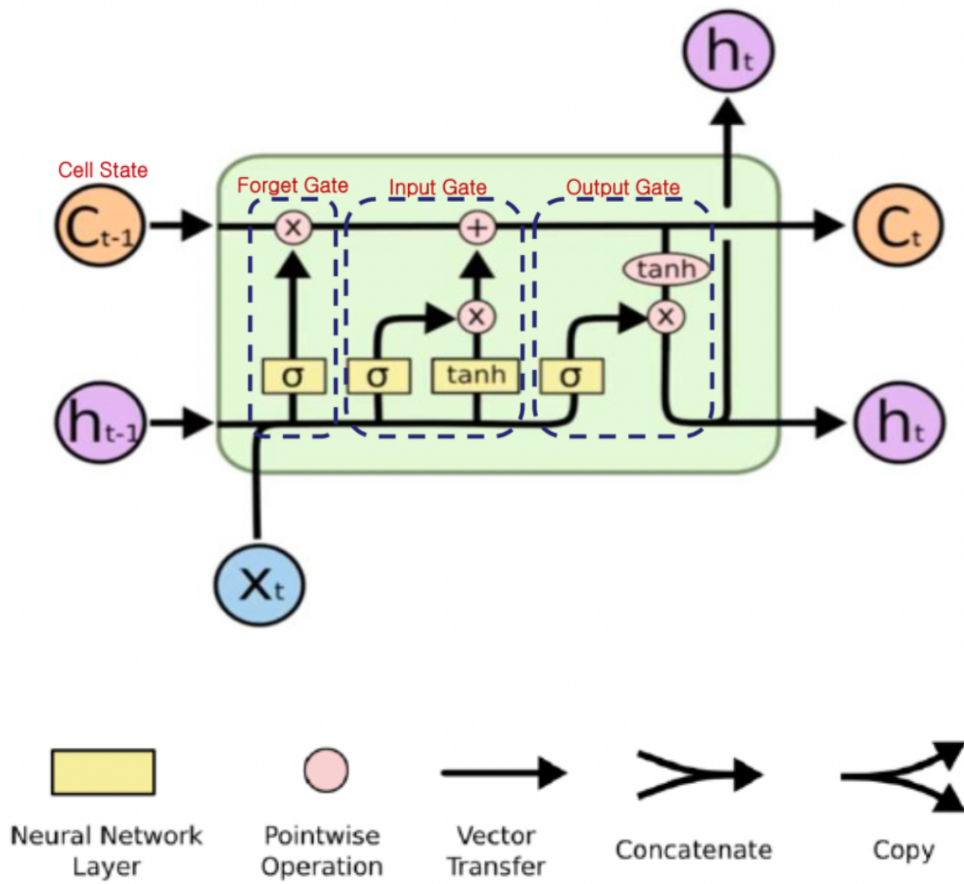
1. 장기 의존성 처리가 어려움
2. 그라디언트의 소실 및 폭발 문제
3. 계산 효율성 - 학습속도 저하



이를 극복하기 위해 나온 모델 중 하나

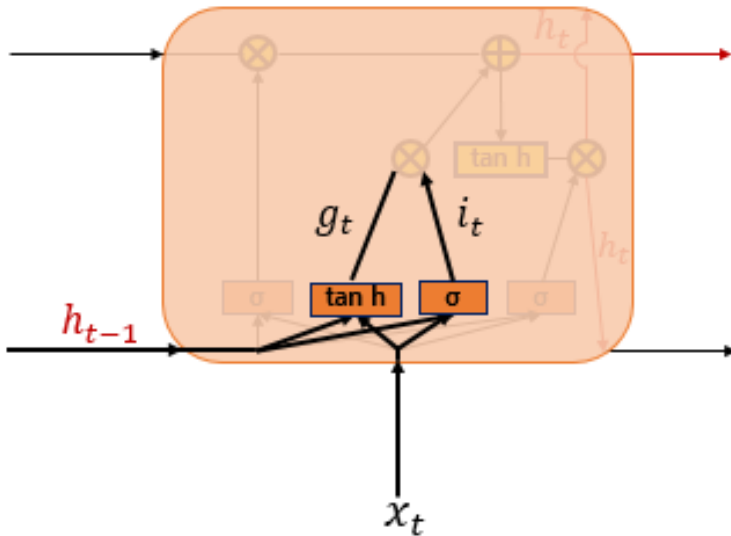
-> LSTM (Long Short-Term Memory)

## LSTM의 핵심 구성 요소



## LSTM 동작방식

1) input 입력 게이트 -  $i_t$   $g_t$



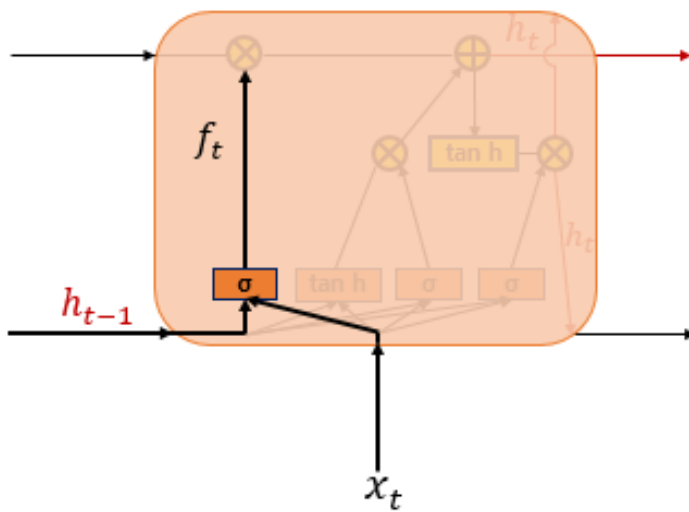
입력 게이트 : 현재 정보를 기억하기 위한 게이트

현재 시점( $t$ )의  $x$ 값과 이전 시점( $t-1$ )의 은닉상태( $h_{t-1}$ )에 각각의 가중치를 곱하여 더한 값을 시그모이드 함수에 넣은 값이  $i_t$ , 탄젠트 함수에 넣은 값이  $g_t$ 임

시그모이드 함수를 지나 0과 1 사이의 값과 하이퍼볼릭탄젠트 함수를 지나 -1과 1사이의 값 두 개가 나오게 됨

⇒ 이 두 개의 값을 가지고 이번에 선택된 기억할 정보의 양을 정하게 됨

## 2) forget 삭제 게이트 - $f_t$



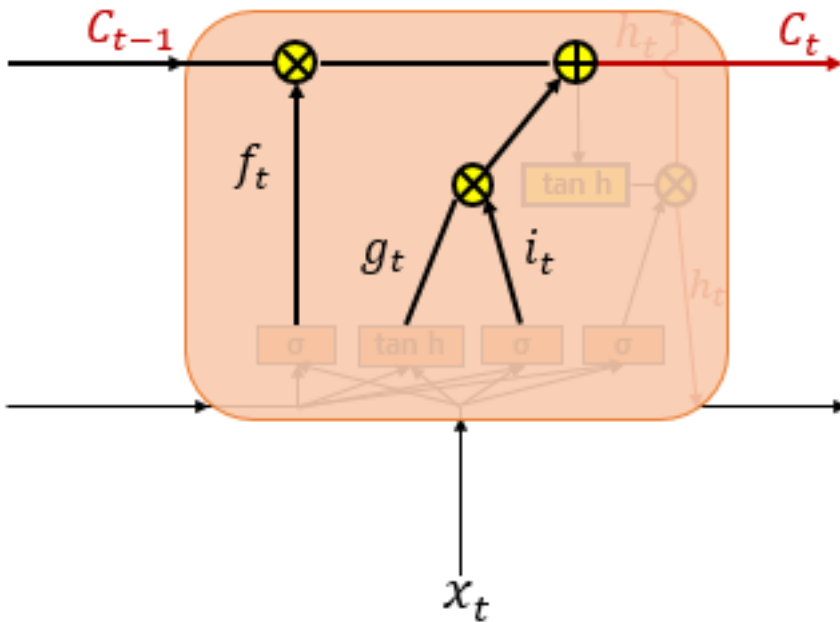
삭제 게이트 : 기억을 삭제하기 위한 게이트

현재 시점( $t$ )의  $x$ 값과 이전 시점( $t-1$ )의 은닉상태가 시그모이드 함수를 지남

이 값이 0에 가까울수록 정보가 많이 삭제된 것이고 1에 가까울수록 정보를 온전히 기억한 것

⇒ 이를 가지고 셀 상태를 구하게 됨

### 3) 셀 스테이트 (cell state)



셀 상태  $C_t$ 를 LSTM에서는 장기 상태라고도 함.

입력 게이트에서 구한  $i_t$ ,  $g_t$  이 두 개의 값에 대해서 원소별 곱을 진행함

⇒ 이것이 이번에 선택된 기억할 값

입력 게이트에서 선택된 기억을 삭제 게이트의 결과값과 더함

⇒ 이 값을 현재 시점  $t$ 의 셀 스테이트라고 하며, 이 값은 다음  $t+1$  시점의 LSTM 셀로 넘겨짐

## 삭제 게이트와 입력 게이트의 영향력 !!

만약 삭제 게이트의 출력값인  $f_t$ 가 0이 된다면, 이전 시점의 셀 상태값인  $C_{t-1}$ 은 현재 시점의 셀 상태값을 결정하기 위한 영향력이 0이 되면서, 오직 입력 게이트의 결과만이 현재 시점의 셀 상태값  $C_t$ 을 결정할 수 있음.

반대로 입력 게이트의  $i_t$ 값을 0이라고 한다면, 현재 시점의 셀 상태값  $C_t$ 는 오직 이전 시점의 셀 상태값  $C_{t-1}$ 의 값에만 의존함.

결과적으로 삭제 게이트는 이전 시점의 입력을 얼마나 반영할지를 의미하고, 입력 게이트는 현재 시점의 입력을 얼마나 반영할지를 결정함

# GRU

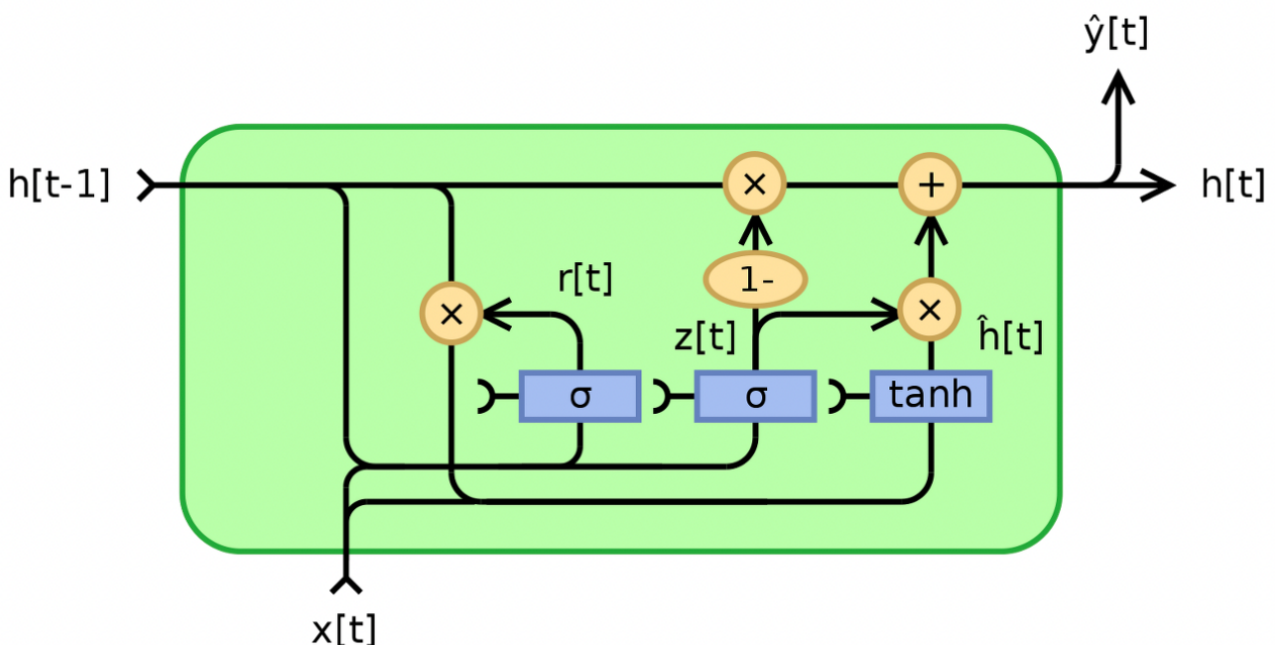
: LSTM 구조를 보다 경량화해서 적은 메모리으로도 빠른 계산이 가능하도록 만든 모델

전체적인 동작은 LSTM과 비슷!

GRU에는 LSTM에서 입력데이터로 사용되던 셀 스테이트와 hidden state (은닉 상태) 두개가 아닌 이를 일원화한 hidden state (은닉 상태) 사용한다는 것이 특징

즉, GRU에서의 hidden state는 LSTM에서의 셀 스테이트와 유사한 역할을 한다고 볼 수 있음

## GRU의 구조



LSTM에서는 input gate와 forget gate의 독립적인 두 개의 gate 결과를 가지고 cell state를 업데이트했다면, GRU에서는 하나의 gate에서 hidden state를 연산하는 것을 볼 수 있음!

이로 인해 구조적으로 GRU는 LSTM에 비해 경량화된 모델로 볼 수 있음

정보를 담는 주된 벡터인 LSTM의 셀스테이트 또는 GRU에서의 hidden state를 업데이트 되는 과정이 기존 RNN처럼 동일한 것을 계속 곱하는 연산이 아니라 매 time step마다 값이 다른 forget gate를 곱하고, 필요로 하는 정보를 곱셈 뿐만이 아니라 덧셈을 통해서 만들어 낼 수 있다는 특징으로 인해 그라디언트 소실 또는 폭발 문제가 많이 사라지는 것으로 알려져있음