

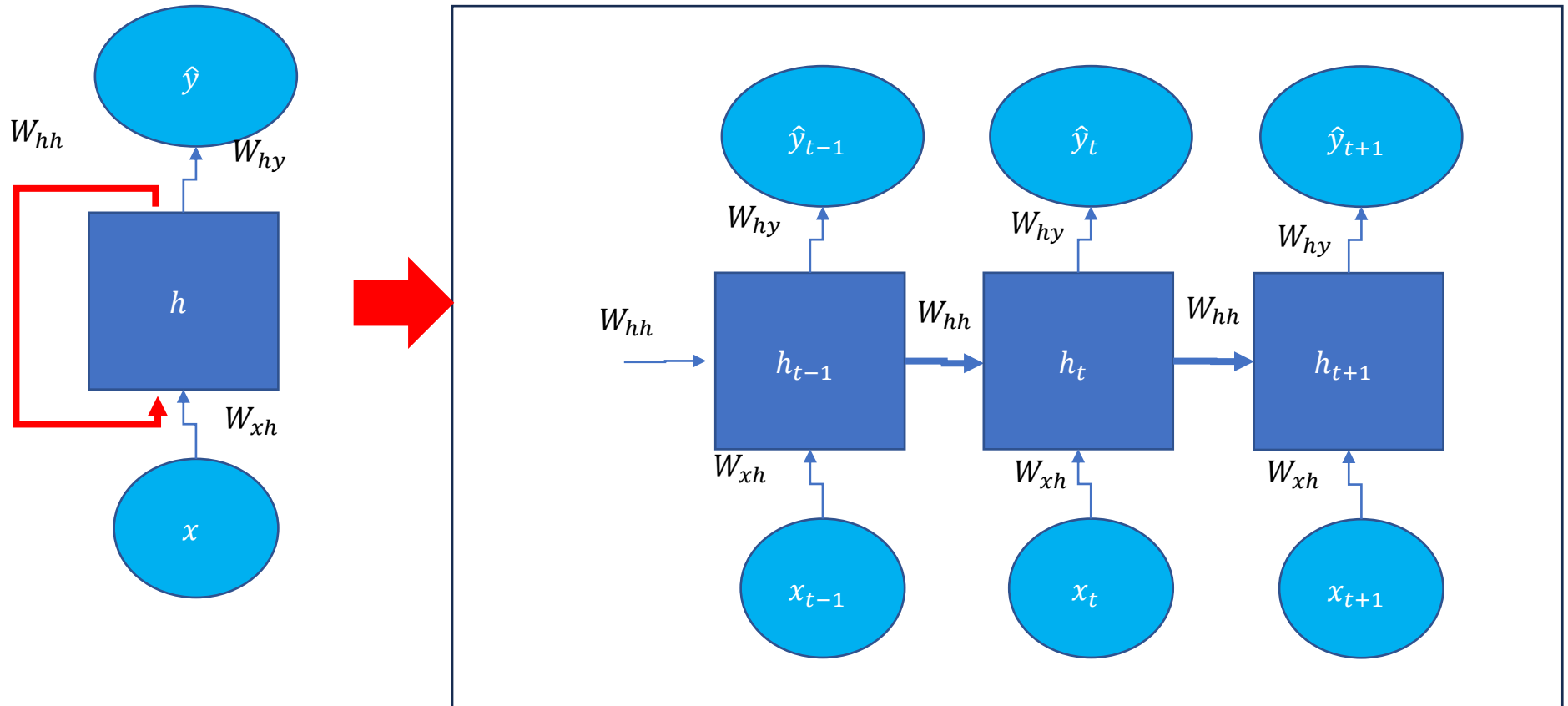


# 딥러닝

2024

# I. RNN 소개

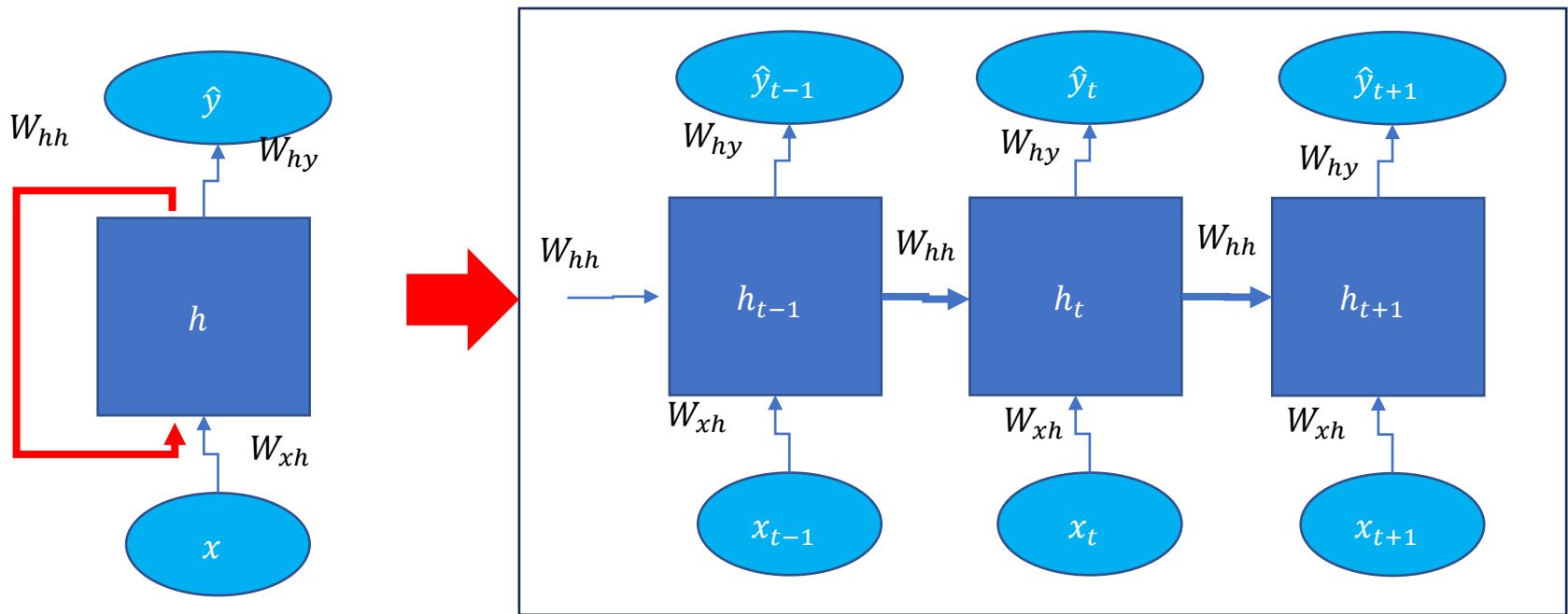
- 순환신경망(Recurrent Neural Networks) 구조



# I. RNN 소개

## • 순환신경망(Recurrent Neural Networks) 구조

- 은닉층 뉴런의 수 = Unit
- $W_{hy}$  : 은닉층에서 출력층으로 전달되는 정보에 대한 가중치
- $W_{xh}$  : 입력층에서 은닉층으로 전달되는 정보에 대한 가중치
- $W_{hh}$  : t시점의 은닉층에서 t+1 시점의 은닉층으로 전달되는 가중치
- 가중치는 모든 시점에 동일(우측 그림 참고)



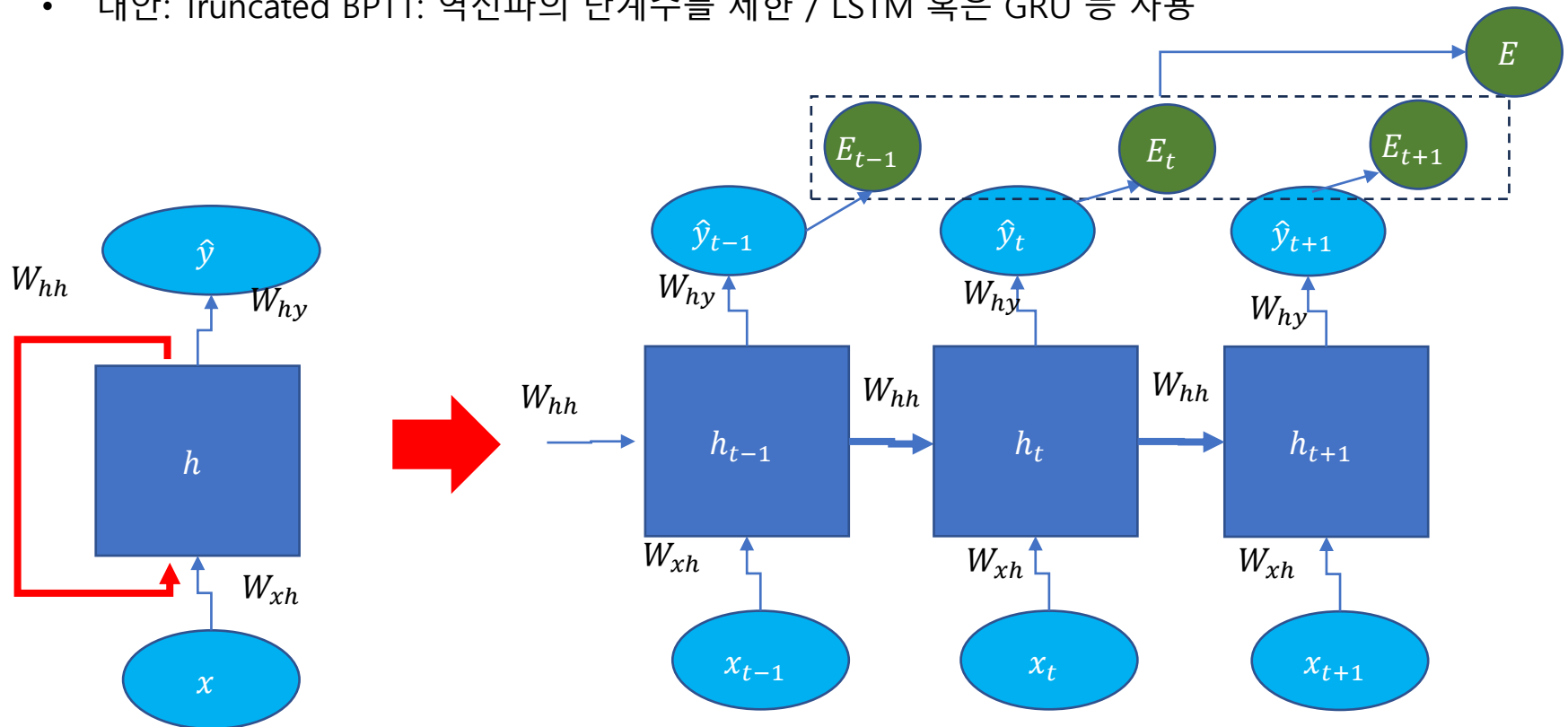
- 순환신경망(Recurrent Neural Networks) 구조

- $t$ 단계 계산의 예
- 은닉층
  - $h_{t-1}$ 와  $x_t$ 가 필요
  - 활성화함수: Hyperbolic Tangent 함수
  - $h_t = \tanh(\hat{y}_t)$
  - $\hat{y}_t = W_{hh} \times h_{t-1} + W_{xh} \times x_t$
- 출력층
  - $\hat{y}_t = \text{softmax}(W_{hy} \times h_t)$
- 오차(E)
  - 각 단계(t)마다 오차 측정: 각 단계의 실제값( $y_t$ )과 예측값( $\hat{y}_t$ )로 MSE 측정
- 역전파(BPTT)
  - Back propagation through time
  - 모든 단계마다 처음부터 끝까지 역전파

# I. RNN 소개

## • 순환신경망(Recurrent Neural Networks) 구조

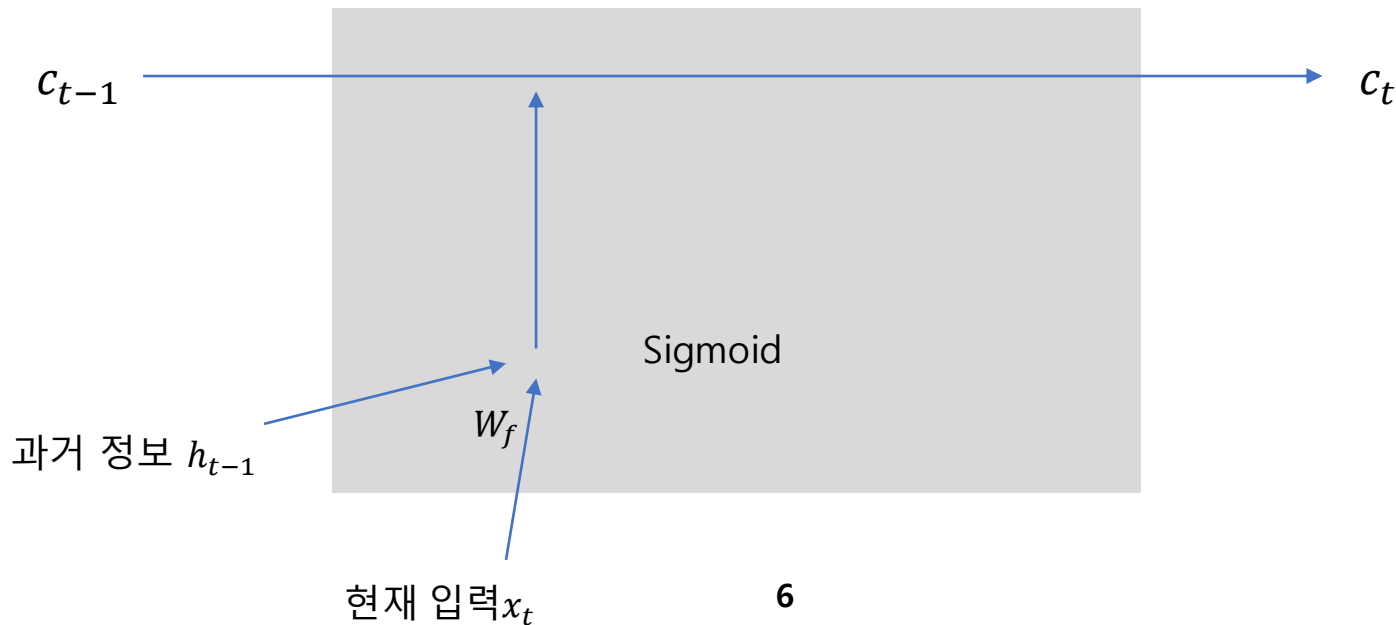
- RNN의 역전파
  - 앞에서 구한 오차를 사용하여 3 종류의 가중치와 Bias 업데이트
  - Vanishing Gradient Problem: BPTT는 오차가 멀리 전파될 수록 전파되는 오차의 양이 감소
  - 대안: Truncated BPTT: 역전파의 단계수를 제한 / LSTM 혹은 GRU 등 사용



# I. RNN 소개

## • LSTM 구조

- 메모리셀인 망각/입력/출력 게이트를 은닉층 각 뉴런에 추가하여 Vanishing Gradient를 해결
  - 망각 게이트: 과거 정보의 기억 범위를 결정
  - 과거 정보  $h_{t-1}$ , 현재 데이터  $x_t$ 를 입력으로 하여 Sigmoid함수를 취한 후 그 값을 과거 정보와 곱
  - Sigmoid 출력이 0이면 과거 정보 망각
  - $f_t = \sigma(W_f[h_{t-1}, x_t])$
  - 셀  $c_t = f_t \cdot c_{t-1}$



# I. RNN 소개

---

- LSTM 구조

- 메모리셀인 망각/입력/출력 게이트를 은닉층 각 뉴런에 추가하여 Vanishing Gradient를 해결
  - 입력 게이트: 현재 시점의 정보를 기억
  - 과거 정보와 현재 데이터를 입력받아 시그모이드, Hyperbolic Tangent를 적용
  - 계산된 값이 1이면  $x_t$ 가 들어오도록 허용, 0이면 계속 차단
  - 출력 게이트: 과거 정보와 현재 데이터를 사용하여 뉴런의 출력 결정
    - 갱신된 메모리 출력을 제어할 수 있음.

---

**QnA**