

# Mathematics for Artificial Intelligence

10강: RNN 첫걸음

---

임성빈

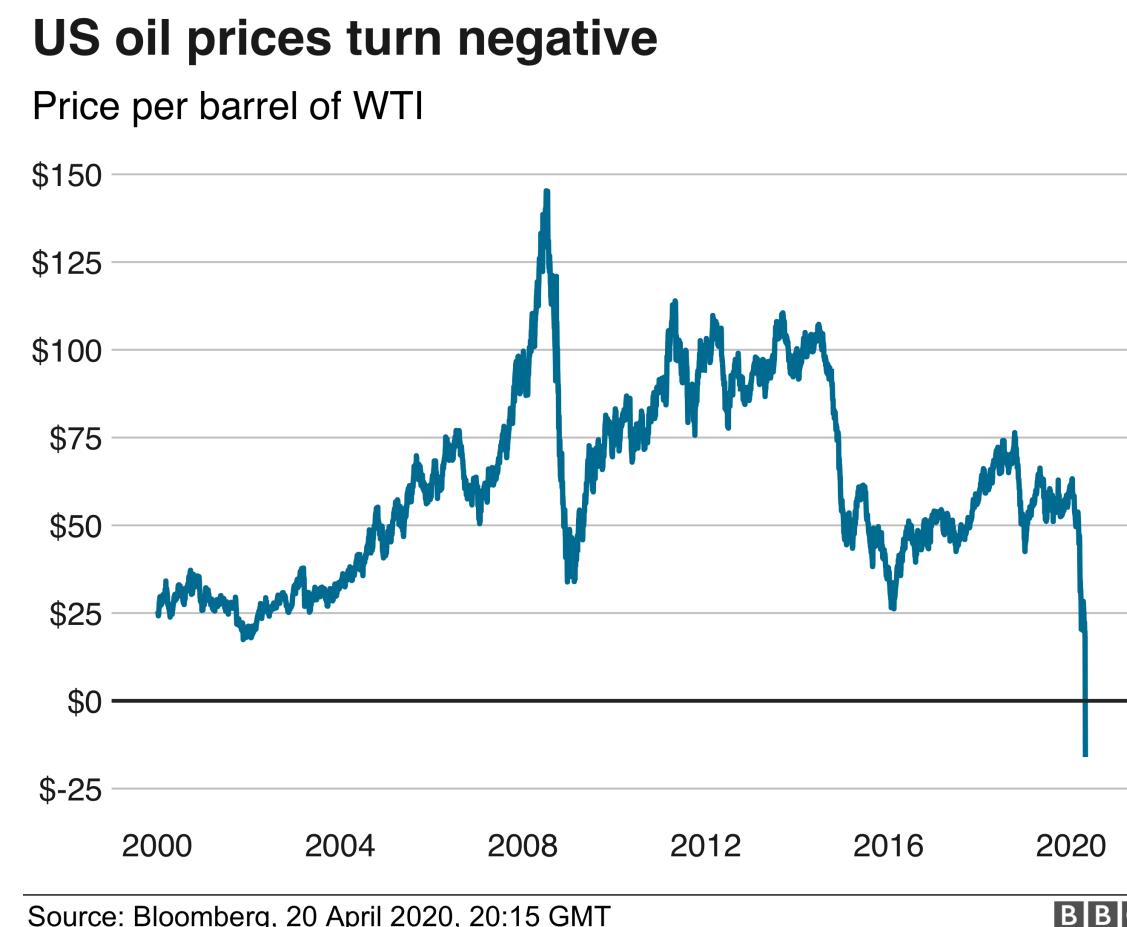


인공지능대학원 & 산업공학과  
Learning Intelligent Machine Lab



# 시퀀스 데이터 이해하기

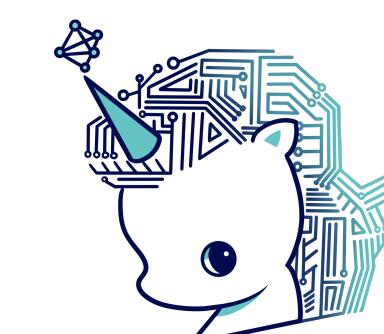
- 소리, 문자열, 주가 등의 데이터를 시퀀스(sequence) 데이터로 분류합니다



automated data mining survey  
responses computer transcripts  
qualitative root cause  
classification insights  
ad-hoc analysis product  
reviews sentiment analysis  
customer dashboards consumer  
trends ad-hoc analysis early warning



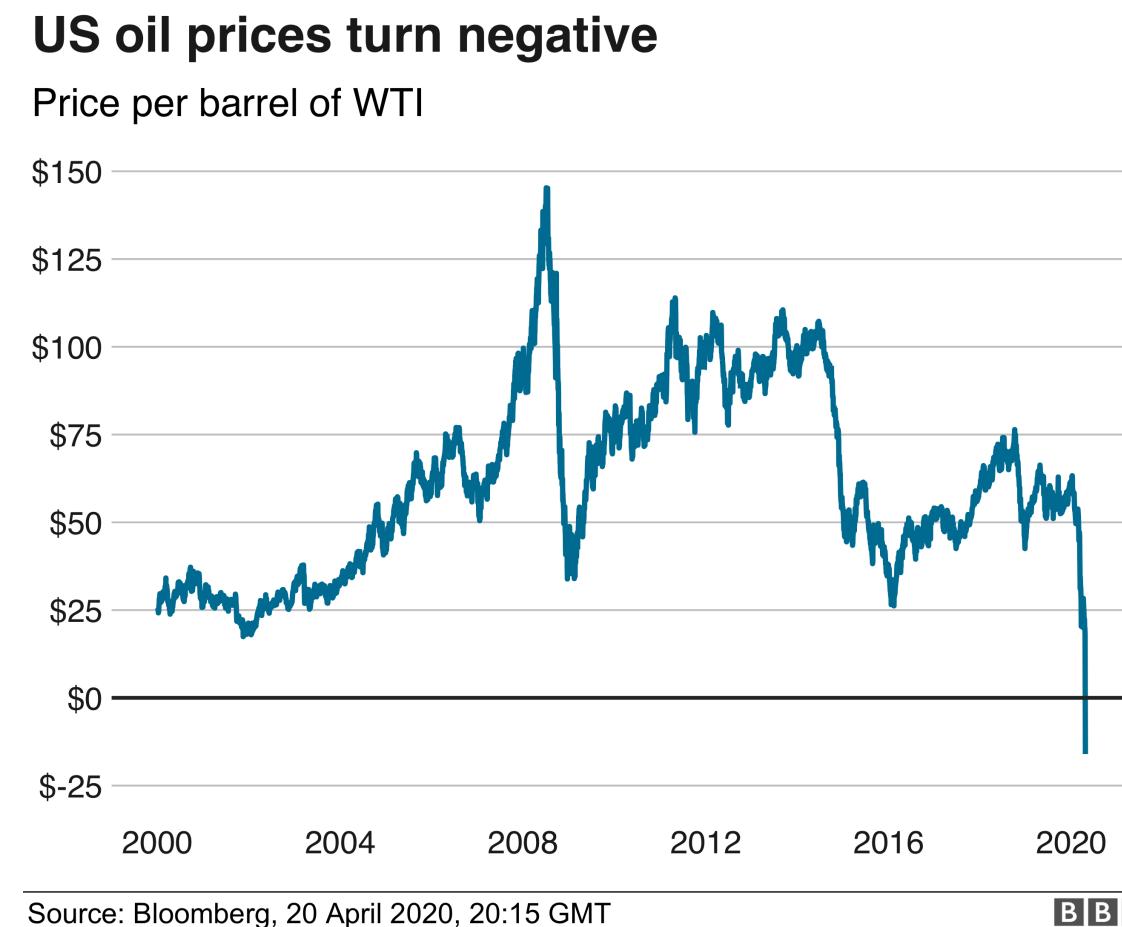
$$X_1, \dots, X_t, \dots$$



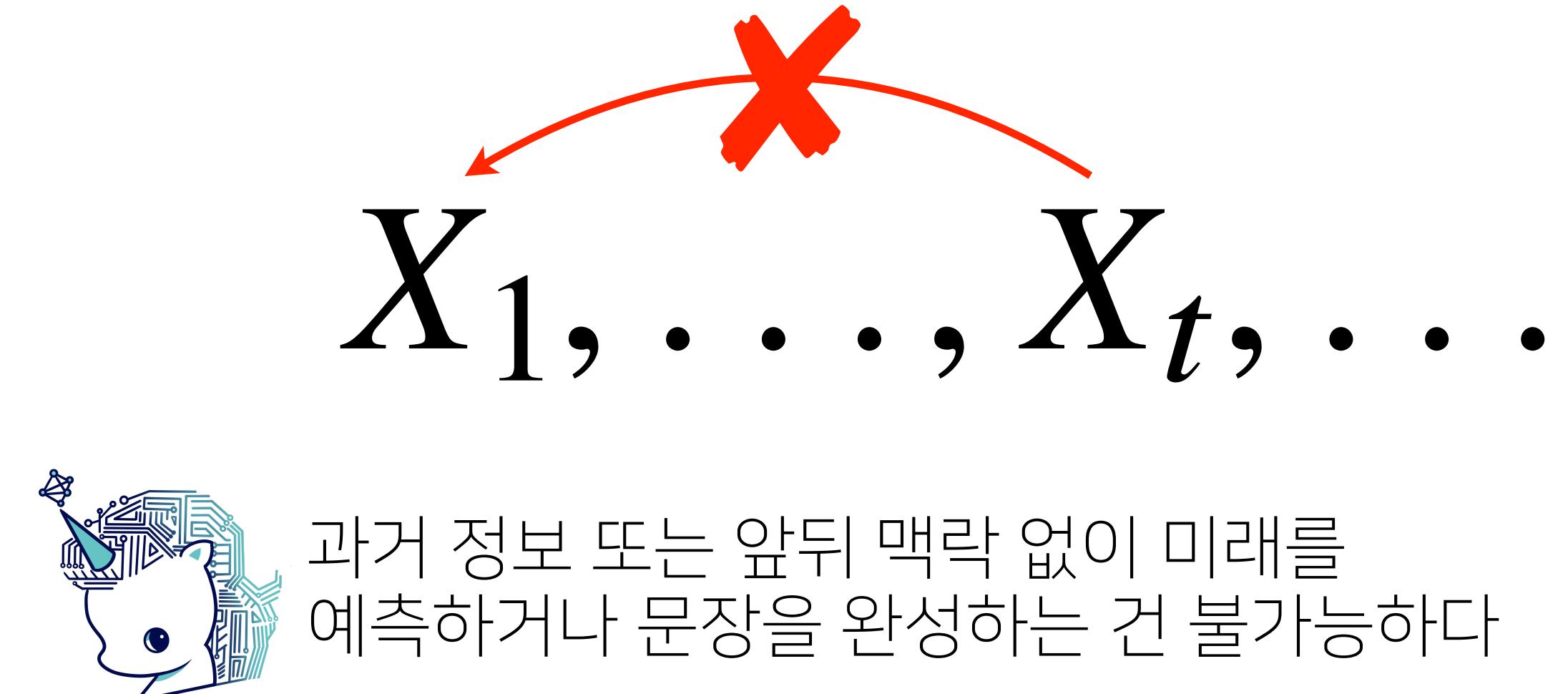
시계열(time-series) 데이터는 시간 순서에 따라 나열된 데이터로 시퀀스 데이터에 속한다

# 시퀀스 데이터 이해하기

- 소리, 문자열, 주가 등의 데이터를 시퀀스(sequence) 데이터로 분류합니다
- 시퀀스 데이터는 독립동등분포(i.i.d.) 가정을 잘 위배하기 때문에 순서를 바꾸거나 과거 정보에 손실이 발생하면 데이터의 확률분포도 바뀌게 됩니다



automated data mining survey  
responses computer transcripts  
qualitative root cause  
classification insights  
ad-hoc analysis product  
reviews sentiment analysis  
customer dashboards consumer  
trends ad-hoc analysis early warning



# 시퀀스 데이터를 어떻게 다루나요?

---

- 이전 시퀀스의 정보를 가지고 앞으로 발생할 데이터의 확률분포를 다루기 위해 조건부확률을 이용할 수 있습니다

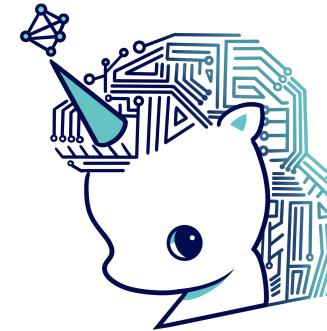
$$P(X_1, \dots, X_t) =$$

# 시퀀스 데이터를 어떻게 다루나요?

---

- 이전 시퀀스의 정보를 가지고 앞으로 발생할 데이터의 확률분포를 다루기 위해 조건부확률을 이용할 수 있습니다

$$P(X_1, \dots, X_t) = P(X_t | X_1, \dots, X_{t-1})P(X_1, \dots, X_{t-1})$$



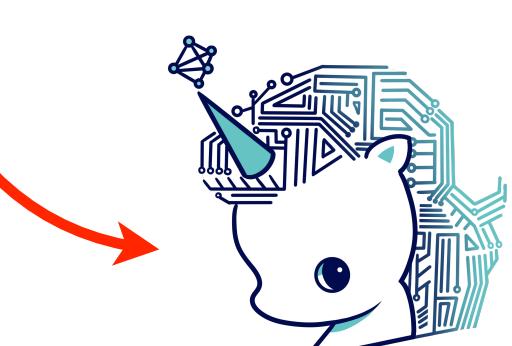
이전에 배운 베이즈 법칙을 사용합니다

# 시퀀스 데이터를 어떻게 다루나요?

---

- 이전 시퀀스의 정보를 가지고 앞으로 발생할 데이터의 확률분포를 다루기 위해 조건부확률을 이용할 수 있습니다

$$\begin{aligned} P(X_1, \dots, X_t) &= P(X_t | X_1, \dots, X_{t-1}) P(X_1, \dots, X_{t-1}) \\ &= P(X_t | X_1, \dots, X_{t-1}) P(X_{t-1} | X_1, \dots, X_{t-2}) \times \\ &\quad \times P(X_1, \dots, X_{t-2}) \\ &= \prod_{s=1}^t P(X_s | X_{s-1}, \dots, X_1) \end{aligned}$$



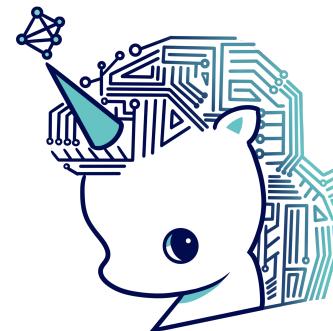
요 기호는  $s = 1, \dots, t$  까지 모두 곱하라는 기호입니다

# 시퀀스 데이터를 어떻게 다루나요?

---

- 이전 시퀀스의 정보를 가지고 앞으로 발생할 데이터의 확률분포를 다루기 위해 조건부확률을 이용할 수 있습니다

$$X_t \sim P(X_t | X_{t-1}, \dots, X_1)$$



위 조건부확률은 과거의 모든 정보를 사용하지만 시퀀스 데이터를 분석할 때 모든 과거 정보들이 필요한 것은 아닙니다

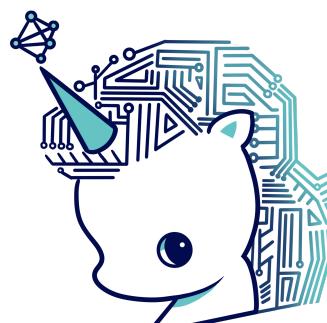
# 시퀀스 데이터를 어떻게 다루나요?

---

- 이전 시퀀스의 정보를 가지고 앞으로 발생할 데이터의 확률분포를 다루기 위해 조건부확률을 이용할 수 있습니다
- 시퀀스 데이터를 다루기 위해선 길이가 가변적인 데이터를 다룰 수 있는 모델이 필요합니다

$$X_t \sim P(X_t | X_{t-1}, \dots, X_1)$$

$$X_{t+1} \sim P(X_{t+1} | X_t, X_{t-1}, \dots, X_1)$$



조건부에 들어가는 데이터 길이는 가변적입니다

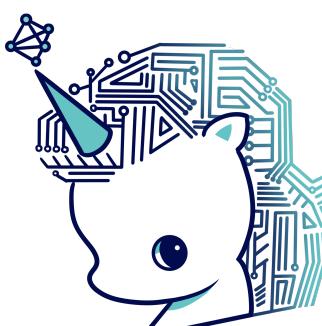
# 시퀀스 데이터를 어떻게 다루나요?

---

- 이전 시퀀스의 정보를 가지고 앞으로 발생할 데이터의 확률분포를 다루기 위해 조건부확률을 이용할 수 있습니다
- 시퀀스 데이터를 다루기 위해선 길이가 가변적인 데이터를 다룰 수 있는 모델이 필요합니다

$$X_t \sim P(X_t | X_{t-1}, \dots, X_1)$$

$$X_{t+1} \sim P(X_{t+1} | X_t, X_{t-1}, \dots, X_1)$$



고정된 길이  $\tau$  만큼의 시퀀스만 사용하는 경우 AR( $\tau$ )  
(Autoregressive Model) 자기회귀모델이라고 부릅니다

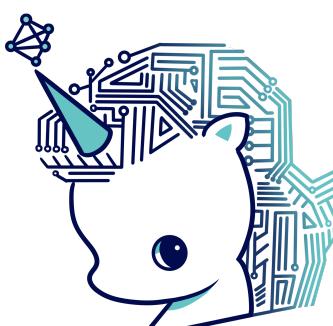
X

# 시퀀스 데이터를 어떻게 다루나요?

- 이전 시퀀스의 정보를 가지고 앞으로 발생할 데이터의 확률분포를 다루기 위해 조건부확률을 이용할 수 있습니다
- 시퀀스 데이터를 다루기 위해선 길이가 가변적인 데이터를 다룰 수 있는 모델이 필요합니다

$$X_t \sim P(X_t | X_{t-1}, \dots, X_1) \rightarrow H_t$$

$$X_{t+1} \sim P(X_{t+1} | X_t, X_{t-1}, \dots, X_1) \rightarrow H_{t+1}$$

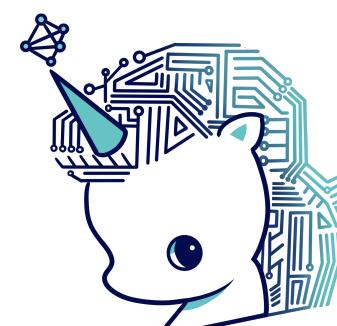


또 다른 방법은 바로 이전 정보를 제외한 나머지 정보들을  $H_t$ 라는 잠재변수로 인코딩해서 활용하는 잠재 AR 모델입니다

# 시퀀스 데이터를 어떻게 다루나요?

- 이전 시퀀스의 정보를 가지고 앞으로 발생할 데이터의 확률분포를 다루기 위해 조건부확률을 이용할 수 있습니다
- 시퀀스 데이터를 다루기 위해선 길이가 가변적인 데이터를 다룰 수 있는 모델이 필요합니다

$$X_t \sim P(X_t | X_{t-1}, H_t)$$
$$H_t = \text{Net}_\theta(H_{t-1}, X_{t-1})$$
$$X_{t+1} \sim P(X_{t+1} | X_t, H_{t+1})$$

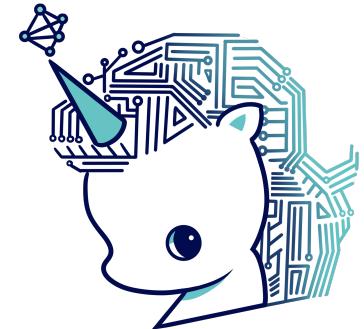


잠재변수  $H_t$  를 신경망을 통해 반복해서 사용하여 시퀀스 데이터의 패턴을 학습하는 모델이 RNN 입니다

X

# Recurrent Neural Network 을 이해하기

- 가장 기본적인 RNN 모형은 MLP 와 유사한 모양입니다



$\mathbf{W}^{(1)}, \mathbf{W}^{(2)}$  은 시퀀스와 상관없이 불변인 행렬입니다

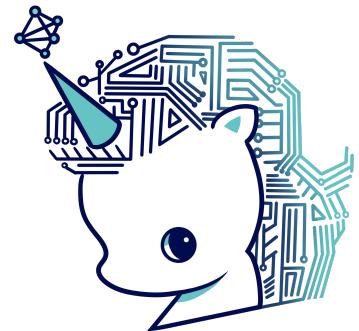
$$\mathbf{O} = \mathbf{H}\mathbf{W}^{(2)} + \mathbf{b}^{(2)}$$

$$\mathbf{H} = \sigma(\mathbf{X}\mathbf{W}^{(1)} + \mathbf{b}^{(1)})$$



# Recurrent Neural Network 을 이해하기

- 가장 기본적인 RNN 모형은 MLP 와 유사한 모양입니다



이 모델은 과거의 정보를 다룰 수 없습니다

$$\mathbf{O}_t = \mathbf{H}_t \mathbf{W}^{(2)} + \mathbf{b}^{(2)}$$

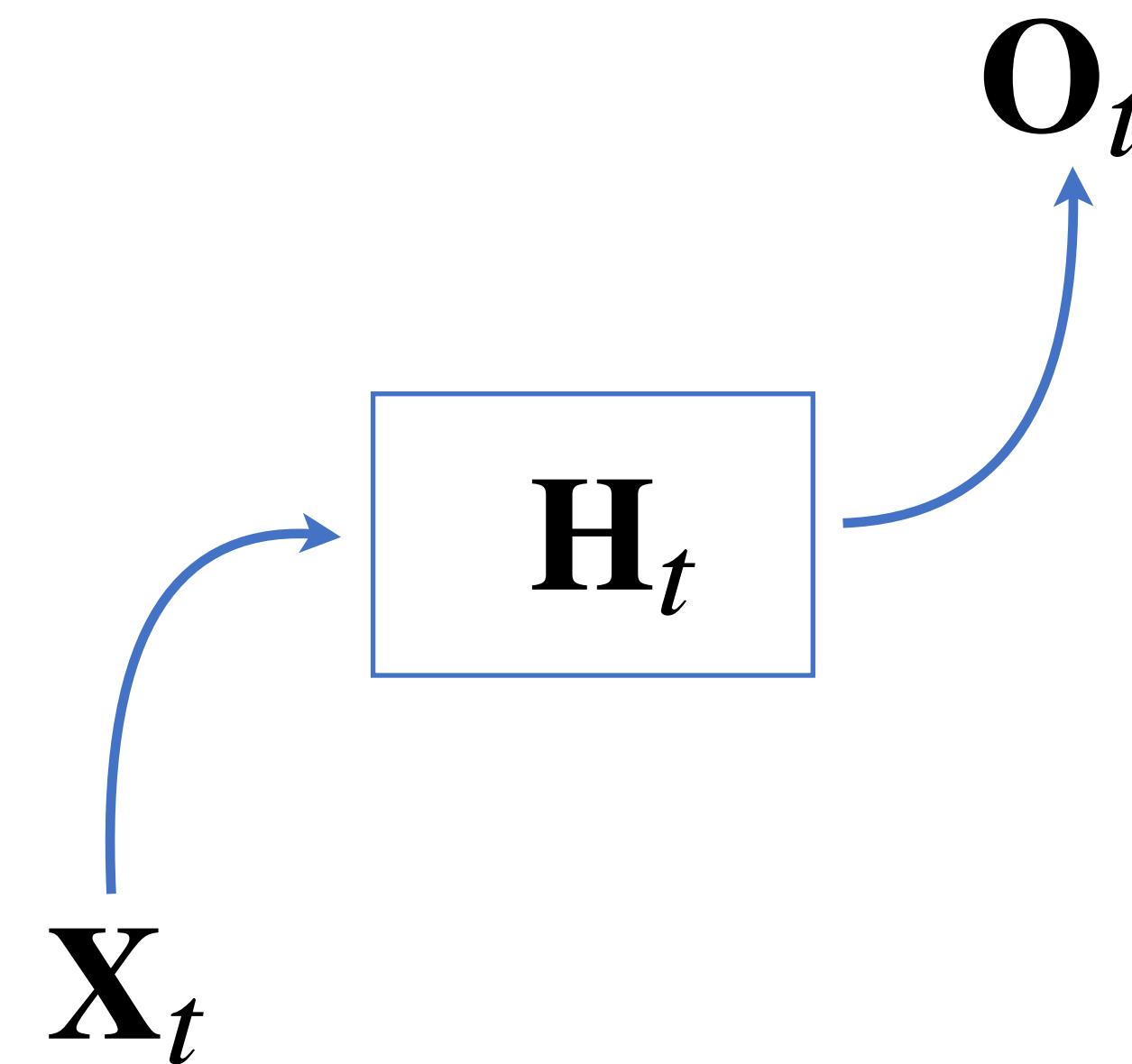
$$\mathbf{H}_t = \sigma(\mathbf{X}_t \mathbf{W}^{(1)} + \mathbf{b}^{(1)})$$

잠재변수

활성화함수

가중치행렬

bias



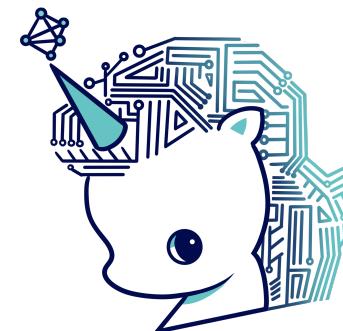
X

# Recurrent Neural Network 을 이해하기

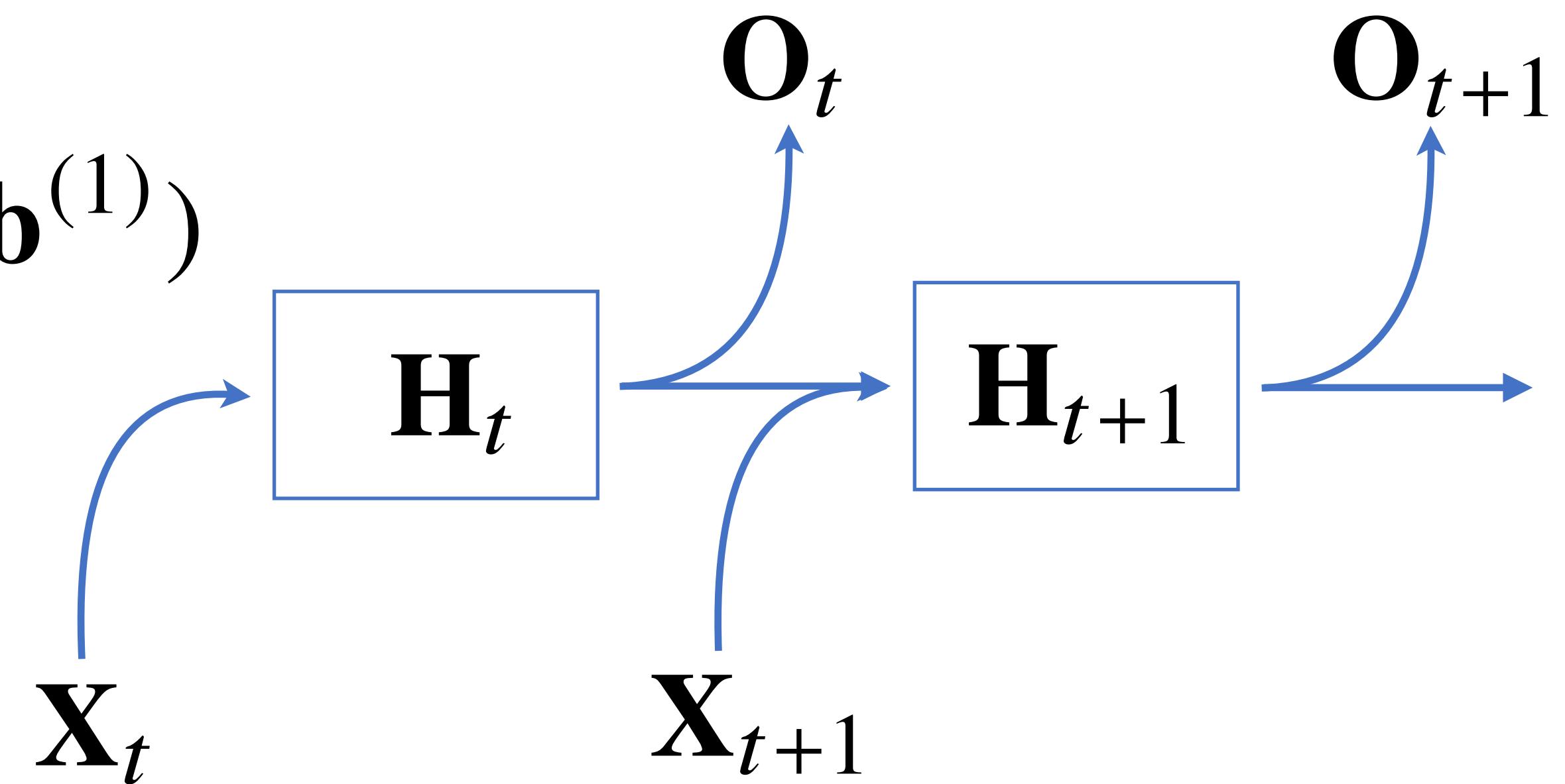
- 가장 기본적인 RNN 모형은 MLP 와 유사한 모양입니다
- RNN 은 이전 순서의 잠재변수와 현재의 입력을 활용하여 모델링합니다

$$\mathbf{O}_t = \mathbf{H}_t \mathbf{W}^{(2)} + \mathbf{b}^{(2)}$$

$$\mathbf{H}_t = \sigma(\mathbf{X}_t \mathbf{W}_X^{(1)} + \mathbf{H}_{t-1} \mathbf{W}_H^{(1)} + \mathbf{b}^{(1)})$$



잠재변수인  $\mathbf{H}_t$  를 복제해서 다음 순서의  
잠재변수를 인코딩하는데 사용합니다

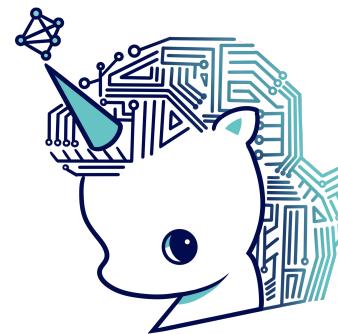


# Recurrent Neural Network 을 이해하기

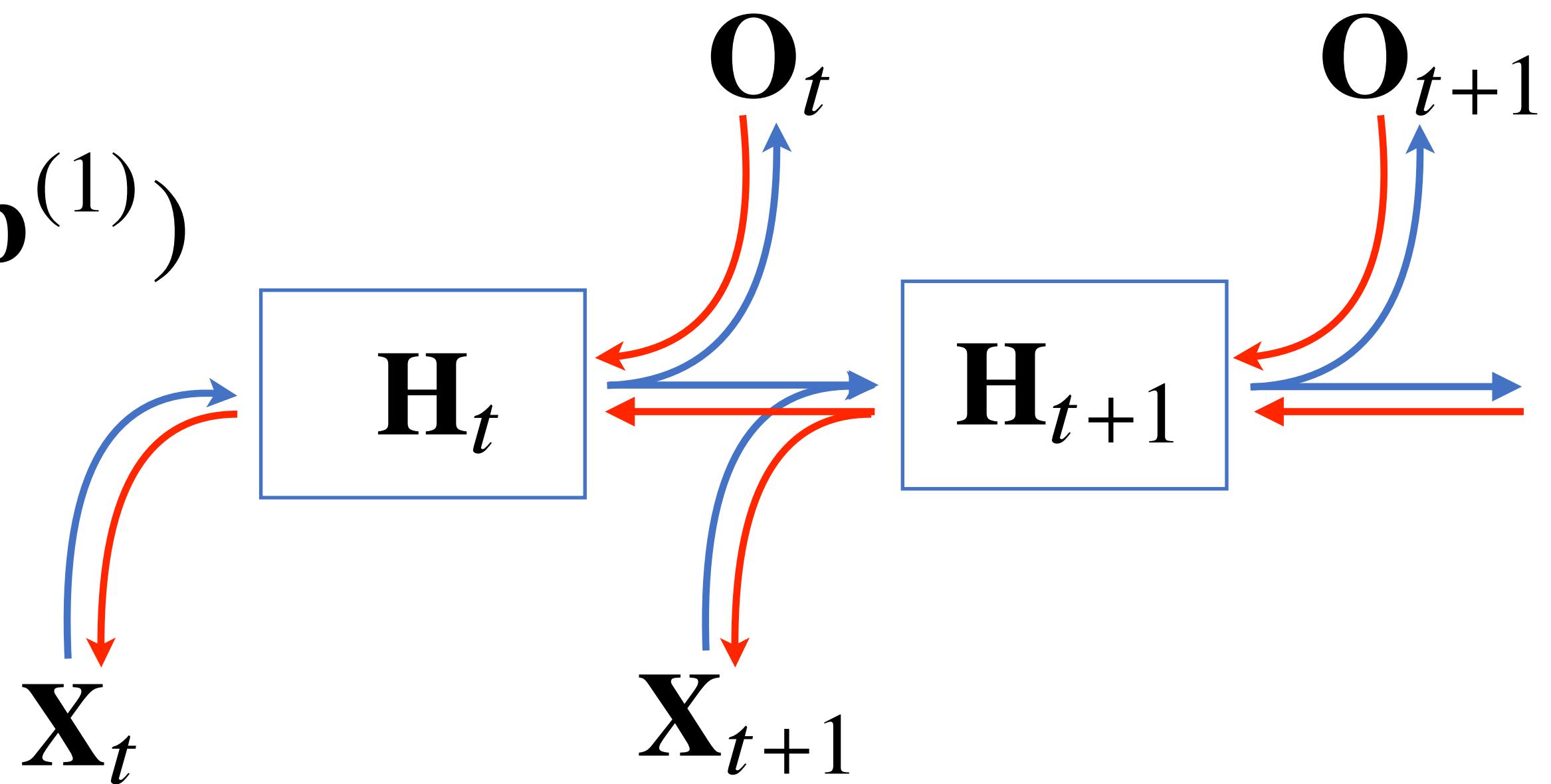
- 가장 기본적인 RNN 모형은 MLP 와 유사한 모양입니다
- RNN 은 이전 순서의 잠재변수와 현재의 입력을 활용하여 모델링합니다
- RNN 의 역전파는 잠재변수의 연결그래프에 따라 순차적으로 계산합니다

$$\mathbf{O}_t = \mathbf{H}_t \mathbf{W}^{(2)} + \mathbf{b}^{(2)}$$

$$\mathbf{H}_t = \sigma(\mathbf{X}_t \mathbf{W}_X^{(1)} + \mathbf{H}_{t-1} \mathbf{W}_H^{(1)} + \mathbf{b}^{(1)})$$



이를 **Backpropagation Through Time** (BPTT) 이라 하며 RNN 의 역전파 방법이다



# BPTT 를 좀 더 살펴봅시다

---

- BPTT 를 통해 RNN 의 가중치행렬의 미분을 계산해보면 아래와 같이 미분의 곱으로 이루어진 항이 계산됩니다

$$L(x, y, w_h, w_o) = \sum_{t=1}^T \ell(y_t, o_t)$$

$$h_t = f(x_t, h_{t-1}, w_h) \text{ and } o_t = g(h_t, w_o).$$

# BPTT 를 좀 더 살펴봅시다

- BPTT 를 통해 RNN 의 가중치행렬의 미분을 계산해보면 아래와 같이 미분의 곱으로 이루어진 항이 계산됩니다

$$L(x, y, w_h, w_o) = \sum_{t=1}^T \ell(y_t, o_t)$$

$$\partial_{w_h} L(x, y, w_h, w_o) = \sum_{t=1}^T \partial_{w_h} \ell(y_t, o_t) = \sum_{t=1}^T \partial_{o_t} \ell(y_t, o_t) \partial_{h_t} g(h_t, w_o) [\partial_{w_h} h_t]$$

$$\partial_{w_h} h_t = \partial_{w_h} f(x_t, h_{t-1}, w_h) + \sum_{i=1}^{t-1} \left( \prod_{j=i+1}^t \partial_{h_{j-1}} f(x_j, h_{j-1}, w_h) \right) \partial_{w_h} f(x_i, h_{i-1}, w_h)$$

×

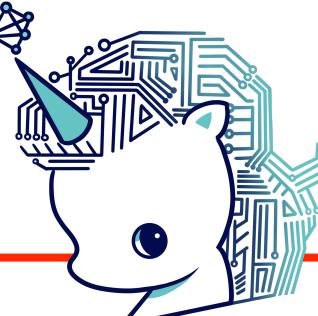
# BPTT 를 좀 더 살펴봅시다

- BPTT 를 통해 RNN 의 가중치행렬의 미분을 계산해보면 아래와 같이 미분의 곱으로 이루어진 항이 계산됩니다

$$L(x, y, w_h, w_o) = \sum_{t=1}^T \ell(y_t, o_t)$$

$$h_t = f(x_t, h_{t-1}, w_h) \text{ and } o_t = g(h_t, w_o).$$

$$\partial_{w_h} L(x, y, w_h, w_o) = \sum_{t=1}^T \partial_{w_h} \ell(y_t, o_t) = \sum_{t=1}^T \partial_{o_t} \ell(y_t, o_t) \partial_{h_t} g(h_t, w_h) [\partial_{w_h} h_t]$$



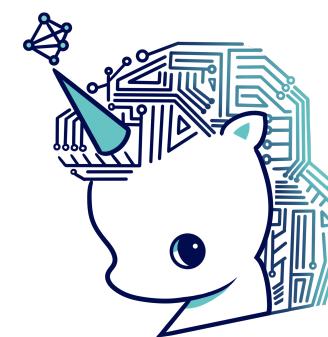
시퀀스 길이가 길어질수록  
이 항은 불안정해지기 쉽습니다

$$\partial_{w_h} h_t = \partial_{w_h} f(x_t, h_{t-1}, w_h) + \sum_{i=1}^{t-1} \left( \prod_{j=i+1}^t \partial_{h_{j-1}} f(x_j, h_{j-1}, w_h) \right) \partial_{w_h} f(x_i, h_{i-1}, w_h)$$

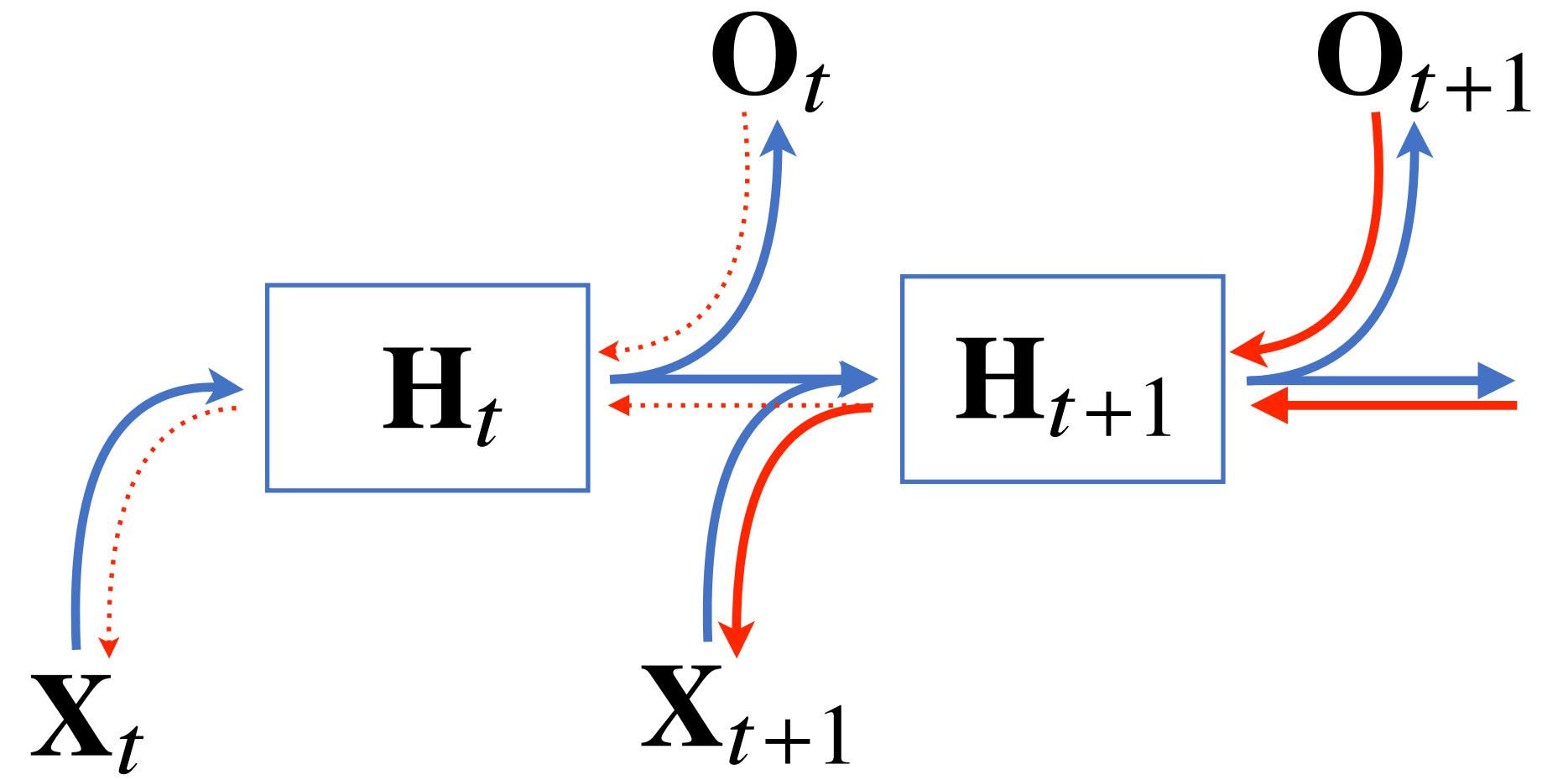
×

# 기울기 소실의 해결책?

- 시퀀스 길이가 길어지는 경우 BPTT 를 통한 역전파 알고리즘의 계산이 불안정 해지므로 길이를 끊는 것이 필요합니다

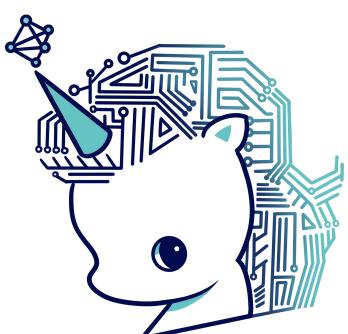


이를 truncated BPTT 라 부릅니다

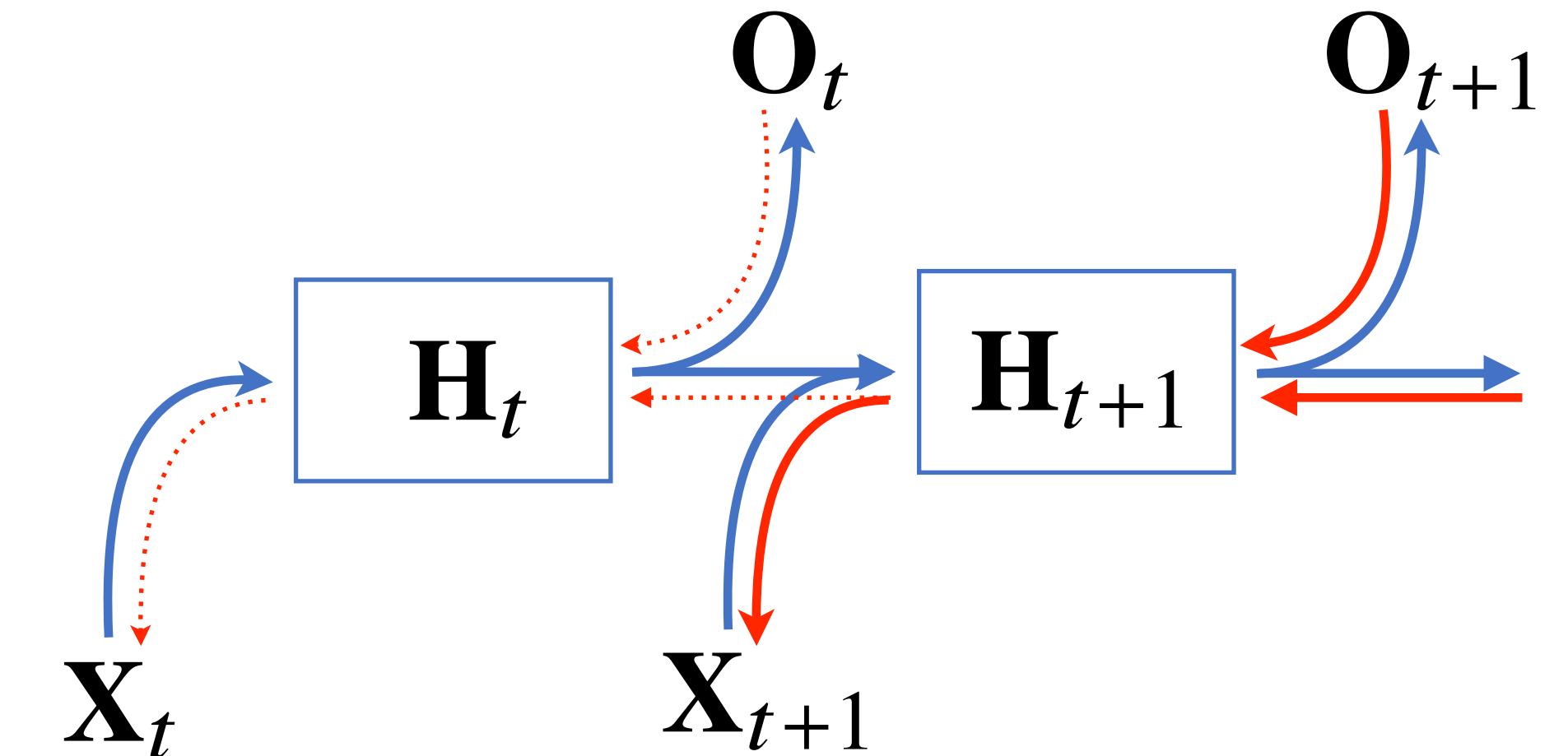
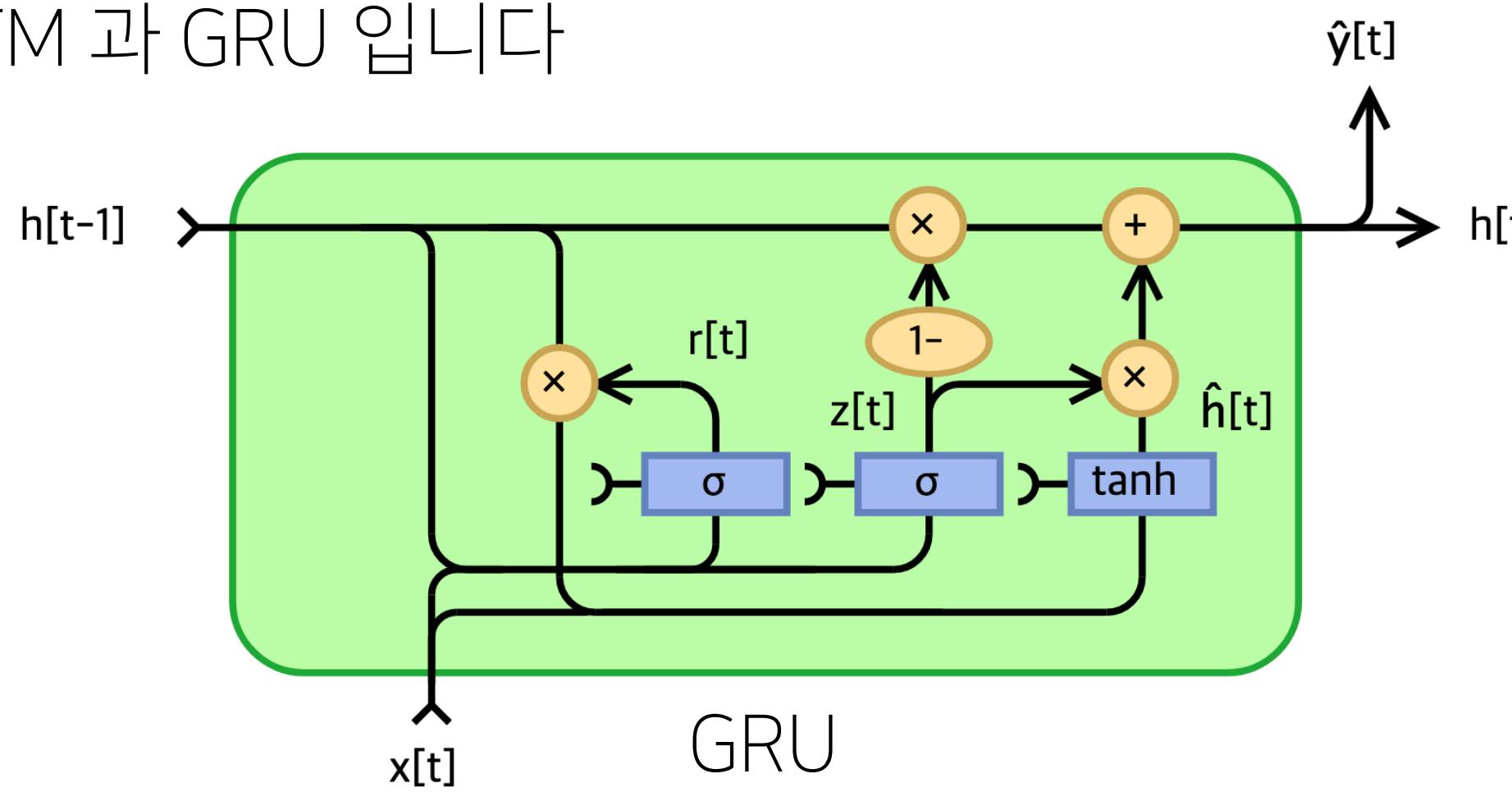


# 기울기 소실의 해결책?

- 시퀀스 길이가 길어지는 경우 BPTT를 통한 역전파 알고리즘의 계산이 불안정 해지므로 길이를 끊는 것이 필요합니다
- 이런 문제들 때문에 Vanilla RNN은 길이가 긴 시퀀스를 처리하는데 문제가 있습니다



이를 해결하기 위해 등장한 RNN  
네트워크가 LSTM과 GRU입니다

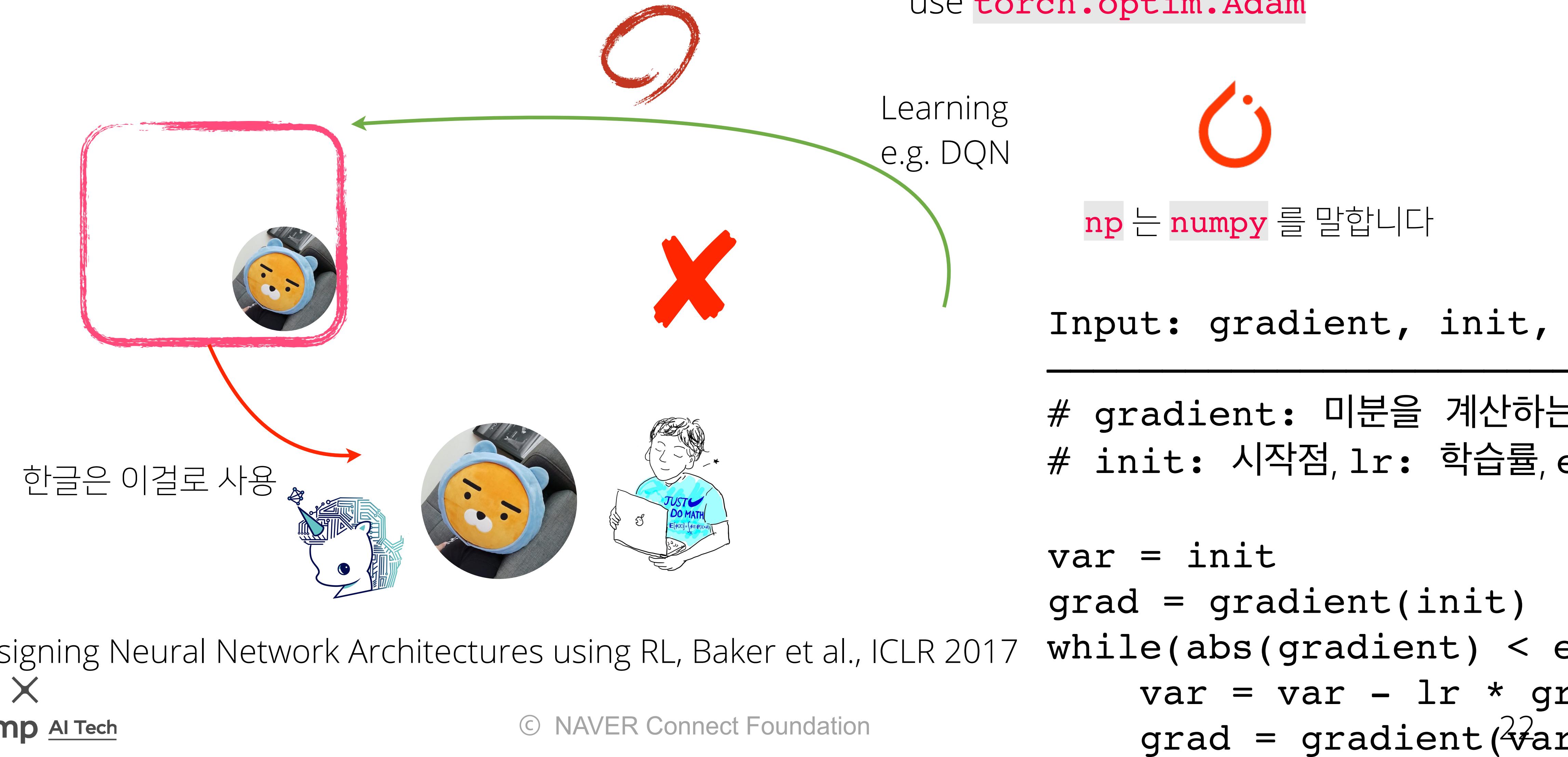


# THE END

---

수고하셨습니다

# Layer Selection ↵ MDP



QNAS: Designing Neural Network Architectures using RL, Baker et al., ICLR 2017