

RNN

- **RNN: Recurrent Neural Network, 순환신경망**

반복적이고 순차적인 데이터(Sequential data)학습에 특화된 인공지능망의 한 종류

내부의 순환구조가 들어있음

순환구조를 이용하여 과거의 학습을 Weight를 통해 현재 학습 반영

기존의 지속적이고 반복적이며 순차적인 데이터학습의 한계를 해결하는 알고리즘

현재의 학습과 과거의 학습의 연결을 가능하게 하고 시간에 종속됨

음성 웨이브폼을 파악, 텍스트의 앞 뒤 성분을 파악할 때 주로 사용

- **시퀀스 데이터:** 소리, 문자열, 주가 등의 데이터를 스퀀스(sequence) 데이터로 분류

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

시퀀스 데이터는 독립동등분포(i.i.d.) 가정을 잘 위배함 => 순서를 바꾸거나 과거 정보에 손실이 발생하면 데이터의 확률분포도 바뀜

- 과거 정보 또는 앞뒤 맥락 없이 미래를 예측하거나 문장을 완성하는 건 불가능하다.

- 이전 시퀀스의 정보를 가지고 앞으로 발생할 데이터의 확률분포를 다루기 위해 조건부확률 이용 가능

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

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

- 과거의 모든 정보를 사용하지만 스퀀스 데이터를 분석할 때 모든 과거 정보들이 필요한 것 X

- 시퀀스 데이터를 다루기 위해 길이가 **가변적**인 데이터를 다룰 수 있는 모델 필요

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

- 조건부에 들어가는 데이터 길이 가변적

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

- 고정된 길이 τ 만큼의 시퀀스만 사용하는 경우 $AR(\tau)$ (Autoregressive Model) 자기회귀모델이라고 부름

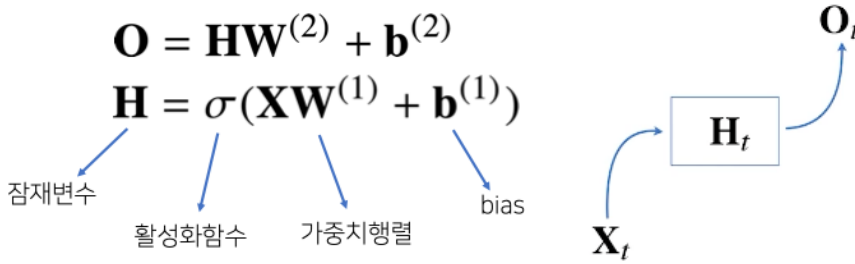
$$\begin{aligned}X_t &\sim P(X_t | X_{t-1}, \dots, X_1) \rightarrow H_t \\X_{t+1} &\sim P(X_{t+1} | X_t, X_{t-1}, \dots, X_1) \rightarrow H_{t+1}\end{aligned}$$

- 다른 방법: 바로 이전 정보를 제외한 나머지 정보들을 H_t 라는 잠재변수로 인코딩해서 활용하는 잠재 AR 모델

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

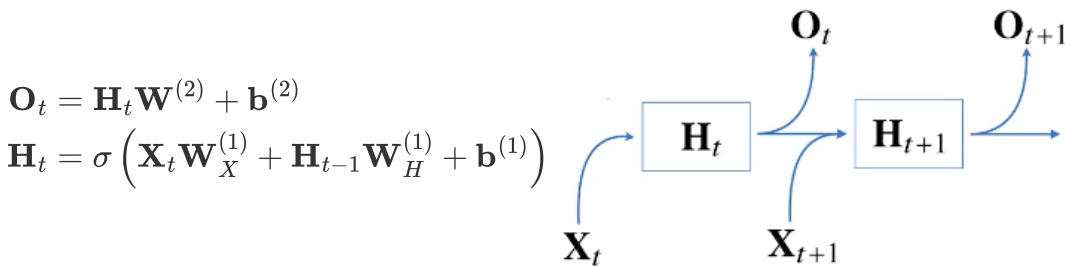
=> 어떻게 인코딩하지 => RNN(순환 신경망)

- RNN: 잠재변수 H_t 를 신경망을 통해 반복해서 사용하여 시퀀스 데이터의 패턴을 학습하는 모델
- 가장 기본적인 RNN 모형: MLP와 유사한 모양



$W^{(1)}, W^{(2)}$: 시퀀스와 상관없이 불변인 행렬

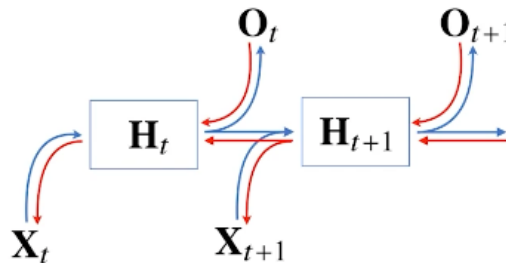
- RNN은 이전 순서의 잠재변수와 현재의 입력을 활용하여 모델링



잠재변수인 H_t 를 복제해서 다음 순서의 잠재변수를 인코딩하데 사용

- t에 따라 변하는 것은 잠재변수, 입력데이터

- RNN의 역전파



잠재변수의 연결그래프에 따라 순차적으로 계산

이를 **Backpropagation Through Time(BPTT)**이라 하고, RNN의 역전파 방법

- BPTT를 통해 RNN의 가중치행렬의 미분을 계산해보면 아래처럼 미분의 곱으로 이루어진 항 계산됨

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

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

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

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

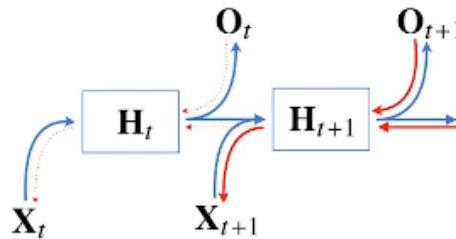


시퀀스 길이가 길어질수록
이 항은 불안정해지기 쉽습니다

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

- 기울기 소실(Gradient Vanishing)의 해결책

- **Truncated BPTT**



스퀀스 길이가 길어지는 경우, BPTT를 통한 역전파 알고리즘의 계산이 불안정해지므로 길이를 끊는 것 필요

=> Vanilla RNN은 길이가 긴 시퀀스를 처리하는데 문제가 있음

=> 이를 해결하기 위해 등장한 RNN 네트워크가 LSTM과 GUE

