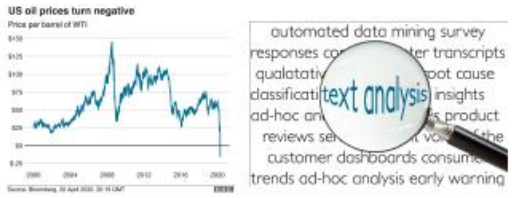
보통 순차적으로 들어오는 데이터를 시퀀스(Sequence) 데이터라고 하며 소리, 문자열, 주가 등이 있다. 시계열(time-series) 데이터 역시 시간 순서에 따라 나열된 데이터로 시퀀스 데이터에 속한다.

시퀀스 데이터는 독립동등분포(i.i.d) 가정을 잘 위배하기 때문에 순서를 바구거나 과거 정보에 손실이 발생하면 데이터의 확률분포도 바뀐다. 예를 들어, '개가 사람을 물었다'와 '사람이 개를 물었다'의 의미는 매우 다르다.



$$X_1, \ldots, X_t, \ldots$$

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

시퀀스 데이터를 다루려면 어떻게 할까? 이전 시퀀스의 정보를 가지고 앞으로 발생할 데이터의 확률분포를 다루기 위해 조건부확률을 이용할 수 있다.

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

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

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

시퀀스 데이터를 다루기 위해서는 길이가 가변적인 데이터를 다룰 수 있는 모델이 필요하다.

방법1. 우리는 과거의 모든 데이터를 가지고 예측을 할 필요는 없다. 따라서 현재 시점에서 봤을 때 고정된 t개의 데이터만 골라 사용할 수 있다. t의 경우 모델링 전에 정해줘야 하는 하이퍼 파라미터이다. 이런 매개변수를 정할 때는 사전지식이 필요하다.

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

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

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

방법2. 바로 이전의 정보와 직전 정보가 아닌 다른 과거의 정보들을 따로 모아 H_t라는 잠재변수로 인코딩해서 활용하는 것을 잠재 AR 모델이라 한다. 이때 H_t 역시 하이퍼 파라미터인데 이 문

제를 해결하기 위해 RNN이 등장한다.

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

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

가장 기본적인 RNN 모형은 MLP와 유사한 모양이다.

MLP를 먼저 상기해보자. 입력행렬에 해당하는 X로부터 가중치 행렬(W)을 곱하고 bias를 더해준다. 그리고 활성화 함수를 통과시켜 잠재변수 H를 만든다. 이 잠재변수 H에 다시 선형모델을 결합시켜 출력 행렬(O)를 출력한다.

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

$$\mathbf{O} = \mathbf{H}\mathbf{W}^{(2)} + \mathbf{b}^{(2)}$$
$$\mathbf{H} = \sigma(\mathbf{X}\mathbf{W}^{(1)} + \mathbf{b}^{(1)})$$

잠재변수 활성화함수 가중치행렬 bias

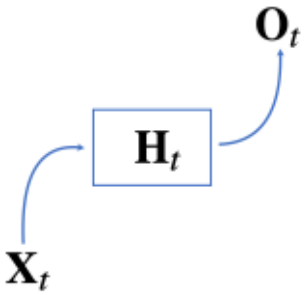
MLP 모델에 시퀀스 데이터를 넣어 모델링하면 이 모델을 가지고 과거의 정보를 다룰 수 없다. 왜냐하면 입력 행렬에 오직 t번째 행렬만 들어오기 때문이다. 즉, 과거의 정보를 잠재변수가 다룰 수 없다.

과거의 정보를 담으려면 어떻게 해야 할까?

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

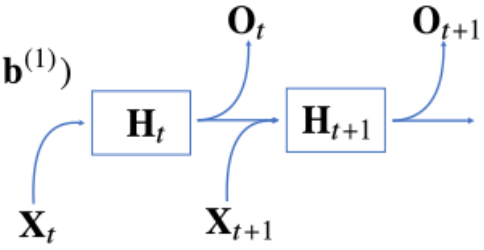


RNN(recurrent neural network)는 아래와 같은 방식으로 과거의 정보를 다룰 수 있다. 아래 식을 보면 중간에 새로운 가중치 행렬(W_H)이 생겼다. 그래서 t번째 잠재변수는 현재 들어온 X_t와 이전 시점의 잠재 변수 H_t-1을 받아서 만들어진다. 이 H_t를 이용해 출력 행렬 O_t를 만들게 된다. 기억해야 할 것은 t로 인해 변하는 것은 잠재변수 H_t, 입력 행렬 X_t이며 W_x, W_h, W는 t에 따

라 변하지 않는다.

$$\mathbf{O}_t = \mathbf{H}_t \mathbf{W}^{(2)} + \mathbf{b}^{(2)}$$
$$\mathbf{H}_t = \sigma(\mathbf{X}_t \mathbf{W}_X^{(1)} + \mathbf{H}_{t-1} \mathbf{W}_H^{(1)} + \mathbf{b}^{(1)})$$

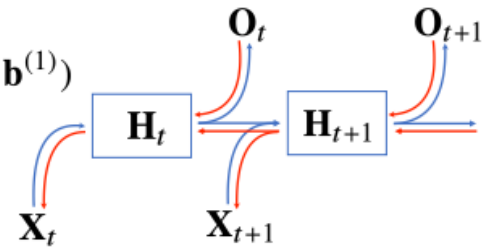
잠재변수인 $\mathbf{H}_t$ 를 복제해서 다음 순서의 잠재변수를 인코딩하는데 사용합니다



위 과정은 지금까지 RNN의 순전파 과정이였다. RNN의 역전파는 위 계산 결과의 반대 방향으로 그레디언트가 흐르게 된다. 또한 RNN의 역전파는 잠재변수의 연결 그래프에 따라 순차적으로 계산한다. 이를 BPTT라고 부른다. 잠재변수에는 바로 다음 시점의 잠재변수에서 나오는 그레디언트 벡터와 출력에서 들어오는 그레디언트 벡터 총 2개가 들어온다. 이 잠재변수에 들어오는 그레디언트 벡터를 입력과 그 이전 시점의 잠재변수로 전달하게 된다. 그리고 이 과정을 반복해 RNN의 학습이 이뤄진다.

$$\mathbf{O}_t = \mathbf{H}_t \mathbf{W}^{(2)} + \mathbf{b}^{(2)}$$
$$\mathbf{H}_t = \sigma(\mathbf{X}_t \mathbf{W}_X^{(1)} + \mathbf{H}_{t-1} \mathbf{W}_H^{(1)} + \mathbf{b}^{(1)})$$

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



BPTT를 통해 RNN의 가중치행렬의 미분을 계산해보면 아래와 같이 미분의 곱으로 이루어진 항이 계산된다.

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

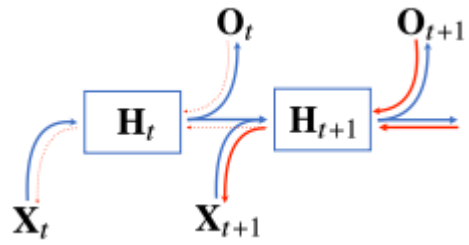
$$L(x, y, w_h, w_o) = \sum_{t=1}^{T} \ell(y_t, o_t)$$

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

시퀀스 길이가 길어질수록 이 항은 불안정해지기 쉽습니다

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

즉, 시퀀스의 길이가 길어지면 BPTT를 통한 역전파 알고리즘의 계산이 불안정해지므로 길이를 끊는 것이 필요하다. 이러한 방식을 truncated BPTT라고 부른다.



하지만 truncated BPTT를 통해 모든 문제를 해결할 수 없다. 이를 해결하기 위해 LSTM과 GRU가 등장한다.