7장(1)~(4)

딥러닝 파이토치 교과서 7장 (1) ~ (4) 파트를 공부하고 assignment 레포에 제출

복습과제

Object detection에서 성능이 좋은 Yolo v2 논문을 읽어오시면 됩니다.

- 모델의 아키텍쳐와 성능 향상을 위한 추가적인 방법들에 초점을 맞춰 간단히 읽어오시면 됩니다.
- 참고 사이트
  - Yolo v2 논문 리뷰 사이트
  - Yolo v2, v3 논문 리뷰 사이트

- 궁금한 사항/공유하면 좋을 추가 자료 등 복습한 내용은 세션 발표 이후 10분동안, 랜덤으로 한 명을 뽑아 발표를 진행 할 예정입니다.

7장. 시계열 분석

7.1 시계열 문제

시계열 분석이란 시간에 따라 변하는 데이터를 사용하여 추이를 분석하는 것입니다. 예를 들어 주가/환율 변동 및 기온/습도 변화 등이 대표적인 시계열 분석입니다. 즉, 추세를 파악하거나 향후 전망 등을 예측하기 위한 용도로 시계열 분석을 사용합니다.

시계열 형태(the components of time series)는 데이터 변동 유형에 따라 불규칙 변동, 추세 변동, 순환 변동, 계절 변동으로 구분할 수 있습니다.

- **불규칙 변동**(irregular variation): 시계열 자료에서 시간에 따른 규칙적인 움직임과 달리 어떤 규칙성이 없어 예측 불가능하고 우연적으로 발생하는 변동을 의미합니다. 전쟁, 홍수, 화재, 지진, 파업 등이 대표적인 예입니다.
- **추세 변동**(trend variation): 시계열 자료가 갖는 장기적인 변화 추세를 의미합니다. 이때 추세란 장기간에 걸쳐 지속적으로 증가·감소하거나 또는 일정한 상태(stationary)를 유지하려는 성향을 의미하기 때문에 짧은 기간 동안에는 추세 변동을 찾기 어려운 단점이 있습니다. 추세 변동의 대표적인 예로는 국내총생산(GDP), 인구증가율 등이 있습니다.
- **순환 변동**(cyclical variation): 대체로 2~3년 정도의 일정한 기간을 주기로 순환적으로 나타나는 변동을 의미합니다. 즉, 1년 이내 주기로 곡선을 그리며 추세 변동에 따라 변동하는 것으로, 경기 변동이 대표적입니다.
- **계절 변동**(seasonal variation): 시계열 자료에서 보통 계절적 영향과 사회적 관습에 따라 1년 주기로 발생하는 것을 의미합니다. 보통 계절에 따라 순환하며 변동하는 특성이 있습니다.

결국 시계열 데이터는 규칙적 시계열과 불규칙적 시계열로 나눌 수 있습니다. 규칙적 시계열은 트렌드와 분산이 불변하는 데이터이며, 불규칙적 시계열은 트렌드 혹은 분산이 변화하는 시계열 데이터입니다. 시계열 데이터를 잘 분석한다는 것은 불규칙성을 갖는 시계열 데이터에 특정한 기법이나 모델을 적용하여 규칙적 패턴을 찾거나 예측하는 것을 의미합니다. 불규칙적 시계열 데이터에 규칙성을 부여하는 방법으로는 AR, MA, ARMA, ARIMA 모델을 적용하는 것이 가장 널리 알려져 있습니다. 하지만 최근에는 딥러닝을 이용하여 시계열 데이터의 연속성을 기계 스스로 찾아내도록 하는 방법이 더 좋은 성능을 내고 있습니다.

7.2 AR, MA, ARMA, ARIMA

시계열 분석은 독립 변수(independent variable)를 사용하여 종속 변수(dependent variable)를 예측하는 일반적인 머신 러닝에서 시간을 독립 변수로 사용한다는 특징이 있습니다. 독립 변수로 시간을 사용하는 특성 때문에 분석하는 데 있어 일반적인 방법론들과 차이가 있는데, 그 차이를 AR, MA, ARMA, ARIMA 모형으로 자세히 살펴보겠습니다.

7.2.1 AR 모델

AR(AutoRegressive)(자기 회귀) 모델은 이전 관측 값이 이후 관측 값에 영향을 준다는 아이디어에 대한 모형으로 자기 회귀 모델이라고도 합니다. AR에 대한 수식은 다음과 같습니다.

$$\underset{①}{Z_t} = \underset{②}{\Phi_1 Z_{t-1} + \Phi_2 Z_{t-2} + \cdots + \Phi_p Z_{t-p}} + \underset{③}{a_t}$$

①은 시계열 데이터에서 현재 시점을 의미하며, ②는 과거가 현재에 미치는 영향을 나타내는 모수(Φ)에 시계열 데이터의 과거 시점을 곱한 것입니다. 마지막으로 ③은 시계열 분석에서 오차항을 의미하며 백색 잡음이라고도 합니다. 따라서 수식은 p 시점을 기준으로 그 이전의 데이터에 의해 현재 시점의 데이터가 영향을 받는 모형이라고 할 수 있습니다.

7.2.2 MA 모델

MA(Moving Average)(이동 평균) 모델은 트렌드(평균 혹은 시계열 그래프에서 y 값)가 변화하는 상황에 적합한 회귀 모델입니다. 이동 평균 모델에서는 윈도우라는 개념을 사용하는데, 시계열을 따라 윈도우 크기만큼 슬라이딩(moving)된다고 하여 이동 평균 모델이라고 합니다. 이동 평균 모델에서 사용하는 수식은 다음과 같습니다.

$$\underset{①}{Z_t} = \underset{②}{\theta_1 a_{t-1} + \theta_2 a_{t-2} + \cdots + \theta_p a_{t-p}} + \underset{③}{a_t}$$

①은 시계열 데이터에서 현재 시점을 의미하며, ②는 매개변수(θ)에 과거 시점의 오차를 곱한 것입니다. 마지막으로 ③은 오차 항을 의미합니다. 따라서 수식은 AR 모델처럼 이전 데이터의 '상태'에서 현재 데이터의 상태를 추론하는 것이 아닌, 이전 데이터의 **오차**에서 현재 데이터의 상태를 추론하겠다는 의미입니다.

7.2.3 ARMA 모델

ARMA(AutoRegressive Moving Average)(자기 회귀 이동 평균) 모델은 AR과 MA를 섞은 모델로 연구 기관에서 주로 사용합니다. 즉, AR, MA 두 가지 관점에서 과거의 데이터를 사용하는 것이 ARMA입니다. 자동 회귀 이동 평균 모델에서 사용하는 수식은 다음과 같습니다.

$$Z_t = a + \Phi_1 Z_{t-1} + \cdots + \Phi_p Z_{t-p} + \theta_1 a_{t-1} + \cdots + \theta_q a_{t-q} + a_t$$

7.2.4 ARIMA 모델

ARIMA(AutoRegressive Integrated Moving Average)(자기 회귀 누적 이동 평균) 모델은 자기 회귀와 이동 평균을 둘 다 고려하는 모형인데, ARMA와 달리 과거 데이터의 선형 관계뿐만 아니라 **추세**(cointegration)까지 고려한 모델입니다.

ARIMA는 파이썬 코드를 이용하여 직접 살펴보겠습니다.

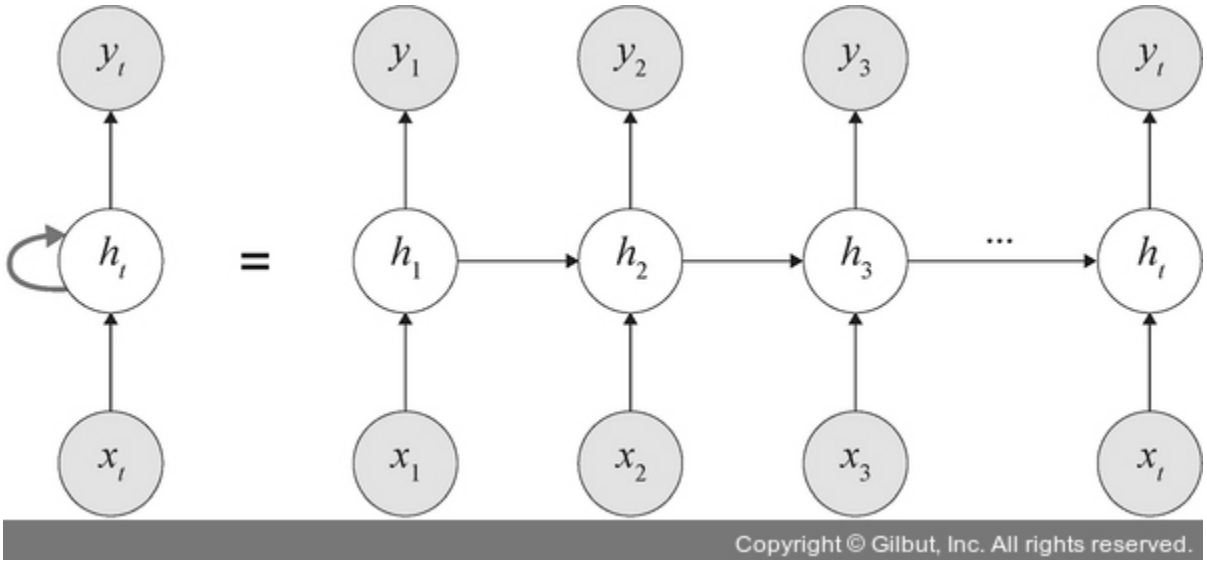
statsmodels 라이브러리를 이용하여 ARIMA 모델을 구현하는데, 절차는 다음과 같습니다.

코드 → 복습과제

여기까지는 시계열 분석을 위한 다양한 방법을 소개했습니다. 하지만 딥러닝 기반의 시계열 모델들이 소개되면서 앞에서 소개한 방법들은 잘 사용되지 않고 있으며 앞으로 다룰 순환 신경망을 많이 사용하고 있습니다.

7.3 순환 신경망(RNN)

RNN(Recurrent Neural Network)은 시간적으로 연속성이 있는 데이터를 처리하려고 고안된 인공 신경망입니다. RNN의 'Recurrent(반복되는)'는 이전 은닉층이 현재 은닉층의 입력이 되면서 '반복되는 순환 구조를 갖는다'는 의미입니다. RNN이 기존 네트워크와 다른 점은 **'기억(memory)'**을 갖는다는 것입니다. 이때 **기억**은 현재까지 입력 데이터를 요약한 정보라고 생각하면 됩니다. 따라서 새로운 입력이 네트워크로 들어올 때마다 기억은 조금씩 수정되며, 결국 최종적으로 남겨진 기억은 모든 입력 전체를 요약한 정보가 됩니다.



그림과 같이 첫 번째 입력(x1)이 들어오면 첫 번째 기억(h1)이 만들어지고, 두 번째 입력(x2)이 들어오면 기존 기억(h1)과 새로운 입력을 참고하여 새 기억(h2)을 만듭니다. 입력 길이만큼 이

과정을 얼마든지 반복할 수 있습니다. 즉, RNN은 외부 입력과 자신의 이전 상태를 입력받아 현재 상태를 갱신합니다.

RNN은 입력과 출력에 따라 유형이 다양합니다.

1. 일대일: 순환이 없기 때문에 RNN이라고 말하기 어려우며, 순방향 네트워크가 대표적 사례입니다.

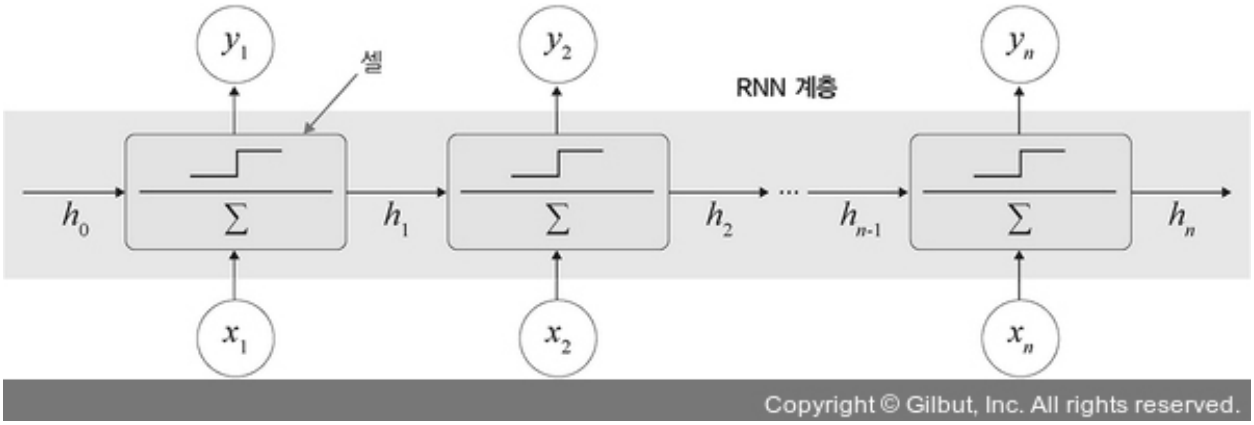
2. 일대다: 입력이 하나이고, 출력이 다수인 구조입니다. 이미지를 입력해서 이미지에 대한 설명을 문장으로 출력하는 이미지 캡션(image captioning)이 대표적 사례입니다.

3. 다대일: 입력이 다수이고 출력이 하나인 구조로, 문장을 입력해서 긍정/부정을 출력하는 감성 분석기에서 사용됩니다.

7.3.1 RNN 계층과 셀

이제 RNN을 구성하는 RNN 계층(layer)과 RNN 셀(cell)을 살펴보겠습니다.

RNN은 내장된(built - in) 계층뿐만 아니라 셀 레벨의 API도 제공합니다. RNN 계층이 입력된 배치 순서대로 모두 처리하는 것과 다르게 RNN 셀은 오직 하나의 단계(time step)만 처리합니다. 따라서 RNN 셀은 RNN 계층의 for loop 구문을 갖는 구조라고 할 수 있습니다.



▲ **그림 7-9** RNN 계층과 RNN 셀

RNN 계층은 셀을 래핑하여 동일한 셀을 여러 단계에 적용합니다. 그림 7-9에서도 X1, X2, …, Xn 등이 전체 RNN 셀에서 사용되고 있습니다. 즉, 셀은 실제 계산에 사용되는 RNN 계층의 구성 요소로, 단일 입력과 과거 상태(state)를 가져와서 출력과 새로운 상태를 생성합니다.

참고로 셀 유형은 다음과 같습니다.

- nn.RNNCell: SimpleRNN 계층에 대응되는 RNN 셀

- nn.GRUCell: GRU 계층에 대응되는 GRU 셀
- nn.LSTMCell: LSTM 계층에 대응되는 LSTM 셀

이렇게 RNN의 계층과 셀을 분리해서 설명하는 이유는 파이토치에서 이 둘을 분리해서 구현이 가능하기 때문입니다. 따라서 앞으로 진행될 RNN 예제는 이 둘을 분리해서 진행합니다.

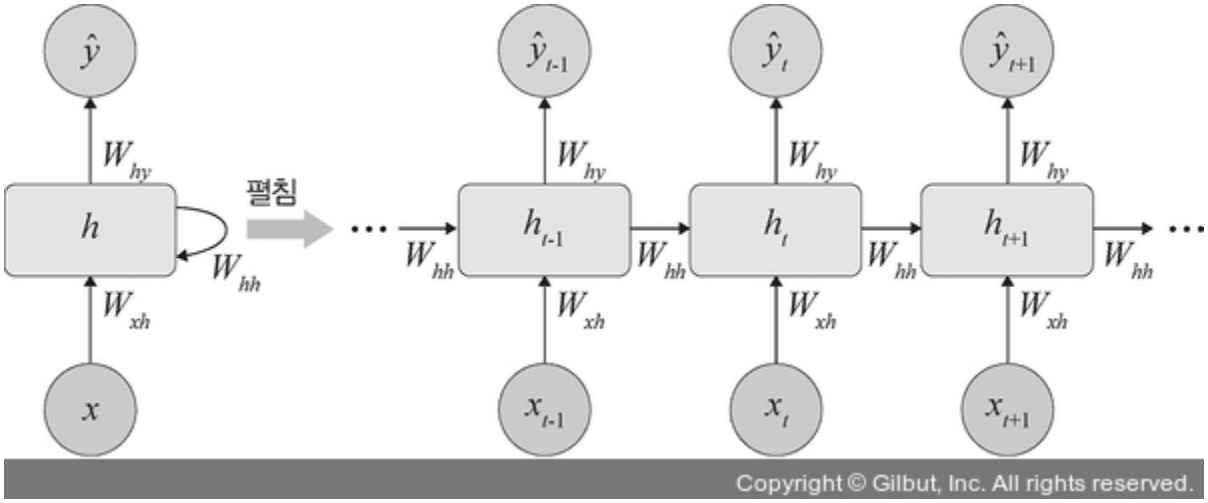
RNN의 활용 분야로는 대표적으로 '자연어 처리'를 꼽을 수 있습니다. 연속적인 단어들의 나열인 언어(자연어) 처리는 음성 인식, 단어의 의미 판단 및 대화 등에 대한 처리가 가능합니다. 이 외에도 손글씨, 센서 데이터 등 시계열 데이터 처리에 활용됩니다.

이제 구체적으로 RNN 구조를 살펴보겠습니다.

7.4 RNN 구조

RNN은 은닉층 노드들이 연결되어 이전 단계 정보를 은닉층 노드에 저장할 수 있도록 구성한 신경망입니다.

다음 그림에서 볼 수 있듯이 xt-1에서 ht-1을 얻고 다음 단계에서 ht-1과 xt를 사용하여 과거 정보와 현재 정보를 모두 반영합니다. 또한, ht와 xt+1의 정보를 이용하여 과거와 현재 정보를 반복해서 반영하는데, 이러한 구조를 요약한 것이 다음 그림의 오른쪽 부분과 같습니다.



▲ **7-10** RNN 구조

RNN에서는 입력층, 은닉층, 출력층 외에 가중치를 세 개 가집니다. RNN의 가중치는 Wxh, Whh, Why로 분류됩니다.

Wxh는 입력층에서 은닉층으로 전달되는 가중치이고, Whh는 t 시점의 은닉층에서 t+1 시점의 은닉층으로 전달되는 가중치입니다. 또한, Why는 은닉층에서 출력층으로 전달되는 가중치입니

다. 가중치 Wxh, Whh, Why는 모든 시점에 동일하다는 것에 주의할 필요가 있습니다. 즉, 가중치를 공유하는데 그림 7-10과 같이 모든 가중치가 동일한 것을 확인할 수 있습니다.

이제 t 단계에서의 RNN 계산에 대해 알아보겠습니다.

1. 은닉층 계산을 위해 xt와 ht-1이 필요합니다. 즉, (이전 은닉층×은닉층 → 은닉층 가중치 + 입력층 → 은닉층 가중치×(현재) 입력 값)으로 계산할 수 있으며, RNN에서 은닉층은 일반적으로 하이퍼볼릭 탄젠트 활성화 함수를 사용합니다. 이를 수식으로 나타내면 다음과 같습니다.

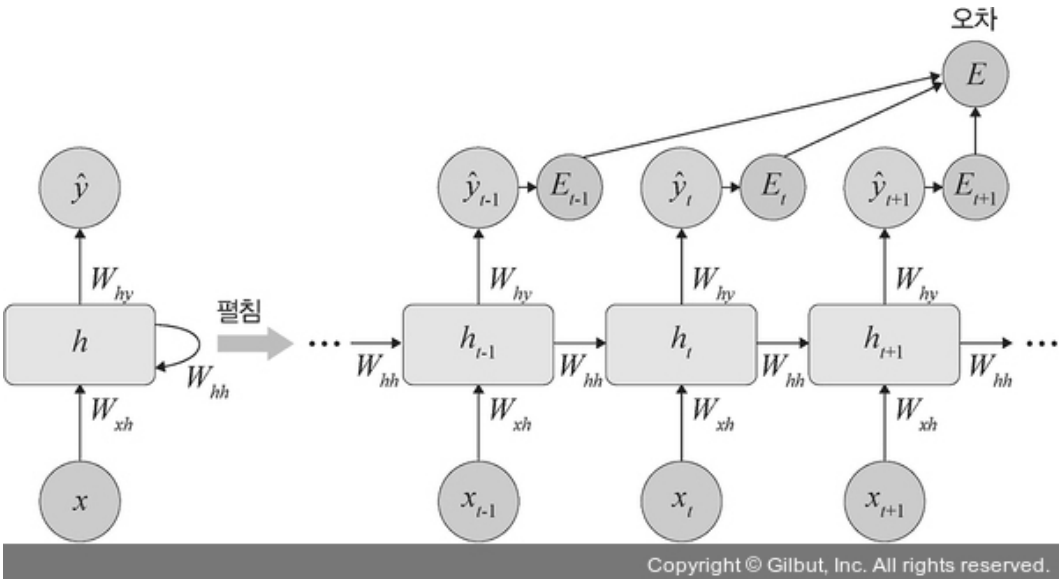
$$h_t = \tanh(\hat{y}_t)$$

$$\hat{y}_t = W_{hh} \times h_{t-1} + W_{xh} \times x_t$$

2. 출력층은 심층 신경망과 계산 방법이 동일합니다. 즉, (은닉층 → 출력층 가중치×현재 은닉층)에 소프트맥스 함수를 적용합니다. 이를 수식으로 나타내면 다음과 같습니다.

$$\hat{y}_t = \text{softmax}(W_{hy} \times h_t)$$

3. RNN의 **오차**(E)는 심층 신경망에서 전방향(feedforward) 학습과 달리 각 단계(t)마다 오차를 측정합니다. 즉, 각 단계마다 실제 값(yt)과 예측 값(yt햇)으로 오차(평균 제곱 오차(mean square error) 적용)를 이용하여 측정합니다.



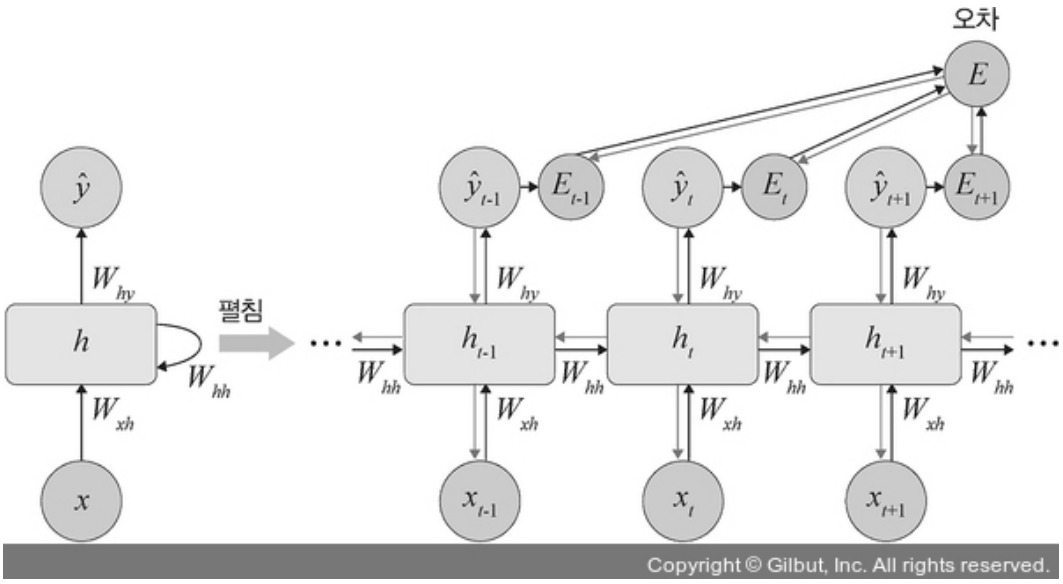
▲ **그림 7-11** RNN의 순방향 학습

4. RNN에서 **역전파**는 BPTT(BackPropagation Through Time)를 이용하여 모든 단계마다 처음부터 끝까지 역전파합니다.

오차는 각 단계(t)마다 오차를 측정하고 이전 단계로 전달되는데, 이것을 BPTT라고 합니다. 즉, **3**에서 구한 오차를 이용하여 Wxh, Whh, Why 및 바이어스(bias)를 업데이트합니다. 이때 BPTT는 오차가 멀리 전파될 때(왼쪽으로 전파) 계산량이 많아지고 전파되는 양이 점차 적어지는 문제점(기울기 소멸 문제(vanishing gradient problem))이 발생합니다. 기울기 소멸 문제를 보완하기 위해 오차를 몇 단계까지만 전파시키는 생략된-BPTT(truncated BPTT)를 사용할 수도 있고, 근본적으로는 LSTM 및 GRU를 많이 사용합니다.

Note ≡ | 생략된-BPTT

계산량을 줄이기 위해 현재 단계에서 일정 시점까지만(보통 5단계 이전까지만) 오류를 역전파하는데, 이것을 생략된-BPTT라고 합니다.



▲ **그림 7-12** RNN의 역방향 학습