

Machine Learning

8장 RNN의 적용과 응용

고려대학교 통계학과
박유성



Contents

01 RNN 모형의 비교

02 다른 형태의 RNN 모형

03 딥러닝을 이용한 시계열 자료분석

RNN의 적용과 응용

- RNN (recurrent neural networks)은 입력자료의 순서가 중요한 정보를 가지고 있을 때 유용한 딥러닝 모형이다.
- 언어의 이해 또는 기계번역에서 단어의 순서는 매우 중요한 정보전달 수단이며,
- 주식가격과 같은 시계열 자료에서 주식가격예측을 위해 최근의 주식가격 또는 주식정보는 매우 중요한 특성변수가 된다.
- 이러한 자료는 한 시점 또는 일정시점 앞의 관측치가 목적변수가 되고 현 시점을 포함하여 과거시점의 관측치가 특성변수의 일부가 된다.
- 그러므로 RNN의 기본구조는 현시점의 자료와 과거시점의 자료가 은닉층에 입력되고 은닉층의 출력도 현시점과 과거시점의 노드값이며 다음 은닉층의 입력으로 전달하는 구조를 가지게 된다.

RNN의 적용과 응용

- 시점 t 에서 은닉층은 두 개의 입력 데이터를 가지고 있다. 직전 은닉층에서의 t 시점 출력값과 동일 은닉층에서의 $(t-1)$ 시점 출력값이다.
- 동일 은닉층에서의 $(t-1)$ 시점 출력값을 입력받음으로써, 과거시점의 정보가 특성 변수의 역할을 할 수 있게 된다.
- 그러나 이러한 반복적인 은닉층의 입력과 출력은 backpropagation 과정에서 모수의 사라짐(weight vanishing)현상이 발생하게 된다.
- 이에 대한 대안이 LSTM (long short term memory)와 GRU (gated recurrent unit)이다.
- 기본적으로 과거 시점 은닉층의 정보가 현시점 은닉층에 간접적으로 입력되는 구조를 가지고 있다. LSTM은 4개의 gate를 이용하여 상태 벡터 c_t 와 은닉층 벡터 h_t 를 출력하고 GRU는 3개의 gate를 이용하여 은닉층 벡터 h_t 만을 출력한다.

01 RNN 모형의 비교

- IMDB 데이터를 이용하여 simple RNN, LSTM, GRU, Bidirectional RNN 등을 비교하고자 한다.

02 다른 형태의 RNN 모형

- 지금까지 논의한 RNN은 자료의 순서가 시간차례대로(chronological order) 입력된다. 시간차례가 주요한 정보를 가지고 있기 때문이다.
- 그러나 시간이 과거에서 현재로 전달해야만 RNN 모형이 좋은 성능을 보이는 것은 아니다.
- 예를 들어, IMDB 자료와 같이 문장을 구성하는 단어의 순서보다 단어의 구성이 좀 더 중요할 경우, 시간의 반대방향(anti-chronological order)으로 제공된 자료도 중요한 정보를 가질 수 있다.
- 시간의 양방향 정보를 모두 이용하는 모형을 양방향 RNN (bidirectional RNN) 이라고 한다.
- 양방향 RNN은 simple RNN, LSTM, GRU 모두 사용 가능하다.

02 다른 형태의 RNN 모형

- IMDB자료와 같이 단어의 순서가 크게 중요하지 않거나 RNN 자료구조에서 시간순서(time steps)가 지나치게 클 경우, CNN의 1D convolution을 이용하여 RNN 분석을 대체할 수 있다.

| | | |
|-------------|-------------|-------------|
| $x_1^{(1)}$ | $x_2^{(1)}$ | $x_3^{(1)}$ |
| $x_1^{(2)}$ | $x_2^{(2)}$ | $x_3^{(2)}$ |
| $x_1^{(3)}$ | $x_2^{(3)}$ | $x_3^{(3)}$ |
| $x_1^{(4)}$ | $x_2^{(4)}$ | $x_3^{(4)}$ |
| $x_1^{(5)}$ | $x_2^{(5)}$ | $x_3^{(5)}$ |
| $x_1^{(6)}$ | $x_2^{(6)}$ | $x_3^{(6)}$ |

 \otimes

| |
|-------|
| w_1 |
| w_2 |
| w_3 |

| |
|-------|
| w_4 |
| w_5 |
| w_6 |

| |
|-------|
| w_7 |
| w_8 |
| w_9 |

 $=$
$$\begin{aligned} &w_1x_1^{(1)} + w_2x_1^{(2)} + w_3x_1^{(3)} \\ &+ w_4x_2^{(1)} + w_5x_2^{(2)} + w_6x_2^{(3)} \\ &+ w_7x_3^{(1)} + w_8x_3^{(2)} + w_9x_3^{(3)} \end{aligned}$$

03 딥러닝을 이용한 시계열 자료분석

- 시계열자료는 하나의 관측대상에 대해 시간의 순서에 따라 반복적으로 자료가 수집된다.
- 그러므로 관측치들이 서로 간에 상호의존적이며 이는 지금까지 딥러닝에서 다룬 데이터와 가장 큰 차이점이다.
- MLP에서 각 표본은 1D 텐서의 특성변수로 구성되어 있으며 이 특성변수가 주어졌다는 가정에서 목적변수가 서로 간에 독립일 때 MLP 모형의 성능이 가장 좋게 된다.
- 동일하게, CNN모형에서는 3D텐서로 구성된 특성변수가 주어져 있을 때, RNN 모형에서는 2D텐서로 구성된 특성변수가 주어져 있을 때, 목적변수 y 가 독립이면 좋은 성능을 구현할 수 있다.
- 이러한 특성을 가진 자료를 시계열자료(time series)와 대비되는 횡단면(cross-sectional)자료라고 한다.

03 딥러닝을 이용한 시계열 자료분석

- 시계열자료는 자료의 순서가 매우 중요한 정보이므로 딥러닝의 아키텍처 중 RNN이 좋은 후보모형이다.
- RNN의 자료구조는 (sample, time-steps, features)로 3D텐서이며 (time-steps, features) 2D텐서가 입력되는 구조이다.
- 시점 t 에서의 주식가격을 y_t 라고 정의하고 y_t 를 목적변수로 $y_{t-1}, y_{t-2}, \dots, y_{t-p}$ 로 하여
- 즉, p 개의 과거시점의 주식가격으로 현시점의 주식가격을 예측하고자 할 때(물론 y_t 를 설명하는 다른 특성변수, 예를 들어 거래량, 시작가, 최저가 등이 있을 수 있다) RNN모형을 적용하게 된다.
- 이 경우, $\text{time-steps}=p, \text{features}=1$ 이 된다. 거의 모든 딥러닝 교재에서는 이러한 자료구조를 가진 RNN모형으로 시계열자료 분석을 하고 있다

03 딥러닝을 이용한 시계열 자료분석

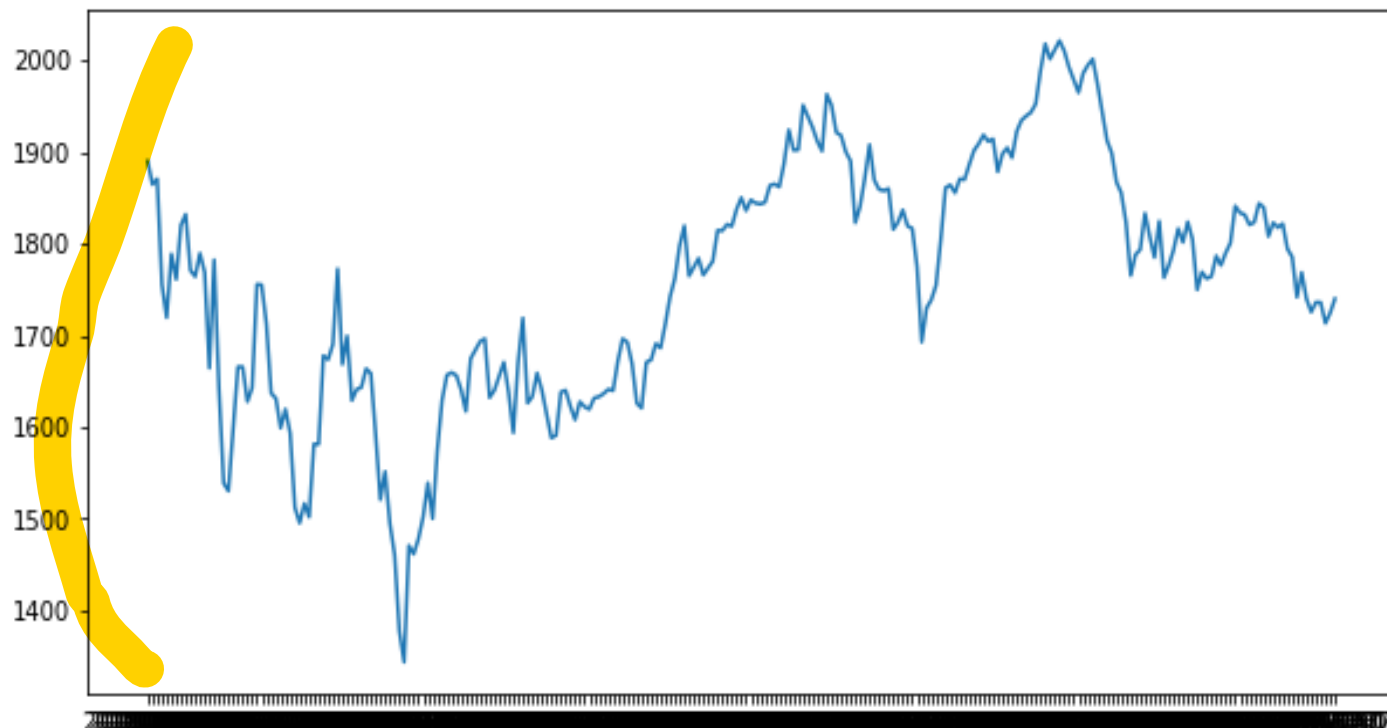
- 그러나 이러한 접근은 두 가지의 근본적인 문제점을 가지고 있다.
- 첫 번째는 임의의 시점 t 를 n 개 뽑았을 때 n 개의 표본을 만들었을 때(Chollet, 2018), 서로 다른 n 개 시점의 y_t 는 사실 조건부 $y_t | y_{t-1}, y_{t-2}, \dots, y_{t-p}$ 로 표현된다.
- 그 이유는 $y_{t-1}, y_{t-2}, \dots, y_{t-p}$ 는 특성변수이고 y_t 는 목적변수이기 때문이다.
- 그러나 이러한 조건부 y_t 가 독립이라는 가정은 일반적으로 성립하지 않는다.
- 이러한 조건부 y_t 는 상관관계가 매우 높은 경우가 많기 때문에 RNN모형의 성능은 떨어질 수 밖에 없으며 RNN모형을 사용할 수도 없다.

03 딥러닝을 이용한 시계열 자료분석

- $y_{t-1}, y_{t-2}, \dots, y_{t-p}$ 는 특성변수를 목적변수 y_t 에 직접 연결하지 못하고
- RNN모형의 특성상, 간접적으로 연결하고 있다.
- 즉, y_{t-p} 를 입력하여 은닉노드 h_{t-p} 를 만든 후, y_{t-p+1} 과 h_{t-p} 를 결합하여 h_{t-p+1} 를 산출한다.
- 이를 반복하여 y_{t-1} 과 h_{t-2} 를 결합하여 최종적으로 y_t 를 예측하는 모형이 RNN이기 때문이다.
- 첫 번째 문제의 발생원인은 시계열 자료의 정상성(stationary) 결여 때문이다. 비정상시계열(non-stationary time series)의 전형적인 예는 시계열의 전반적인 추세(general trend)에서 어떤 시점의 값이 이 추세를 벗어나면 이 시점 이후 일정시간 동안 추세를 벗어나 있다가 느리게 전반적인 추세로 돌아오거나 아예 돌아오지 않은 시계열 플롯을 보일 때이다.

03 딥러닝을 이용한 시계열 자료분석

- 다음 그림은 NASDAQ GS stock (2018년 10월 5일~2019년 10월 14일)에 대한 총 251일 동안의 일일종가 플롯이다.



03 딥러닝을 이용한 시계열 자료분석

- 실제로 시계열의 정상성여부는 Dickey-Fuller의 단위근 검정(unit root test)을 통해 점검한다.
- 단위근 검정의 귀무가설은 H_0 : "단위근이 있다"(비정상시계열)이고 대립가설 H_a : "단위근이 없다".
- Dickey-Fuller의 단위근 검정결과 귀무가설을 기각하지 못하면 먼저 시계열 자료를 차분하여 자료를 정상시계열로 전환하여야 한다
- $y_t - y_{t-d}$ 를 차분이라고 하며 일반적으로 $d=1$ 이다.
- y_t 를 주식가격이라고 할 때 $\log y_t - \log y_{t-1} = \log \frac{y_t}{y_{t-1}}$ 을 로그수익률이라고 함.
- 이 수익률을 분석자료로 이용한다. 근본적인 이유는 주식가격 y_t 는 비정상시계열로 통계적 분석대상이 될 수 없지만, 차분을 취해 구한 로그수익률은 정상시계열로 통계적 모형(동일하게 머신러닝 또는 딥러닝 모형)의 대상이 되기 때문이다.

03 딥러닝을 이용한 시계열 자료분석

- 정상시계열을 만든 후, 예를 들어 $z_t = \log \frac{y_t}{y_{t-1}}$ 으로 정의하면 $z_t | z_{t-1}, z_{t-2}, \dots, z_{t-p}$ 가 독립이 되는 p 를 결정해야 한다.
- p 는 자기상관함수(autocorrelation function, acf)와 편자기상관함수(partial autocorrelation function, pacf)를 이용하여 구할 수 있다.
- acf와 pacf는 자료가 정상성이면 p 가 증가하면서 지수적 감소 또는 절단(cut-off)형태이어야 한다.
- 만약 둘 중 하나라도 느리게 감소하거나 증가하면 시계열자료가 비정상이므로 적절한 차분을 통해 자료를 정상성으로 만들어야 한다.
- acf와 pacf를 통해 p 가 결정되면 z_t 는 목적변수가 되고 $z_{t-1}, z_{t-2}, \dots, z_{t-p}$ 는 특성변수가 된다.

Q & A