스마트 에너지를 위한
시계열 데이터 처리와
ARIAM, PROPHET, LSTM 모델

2024년 1월 22일~1월 23일

LSTM

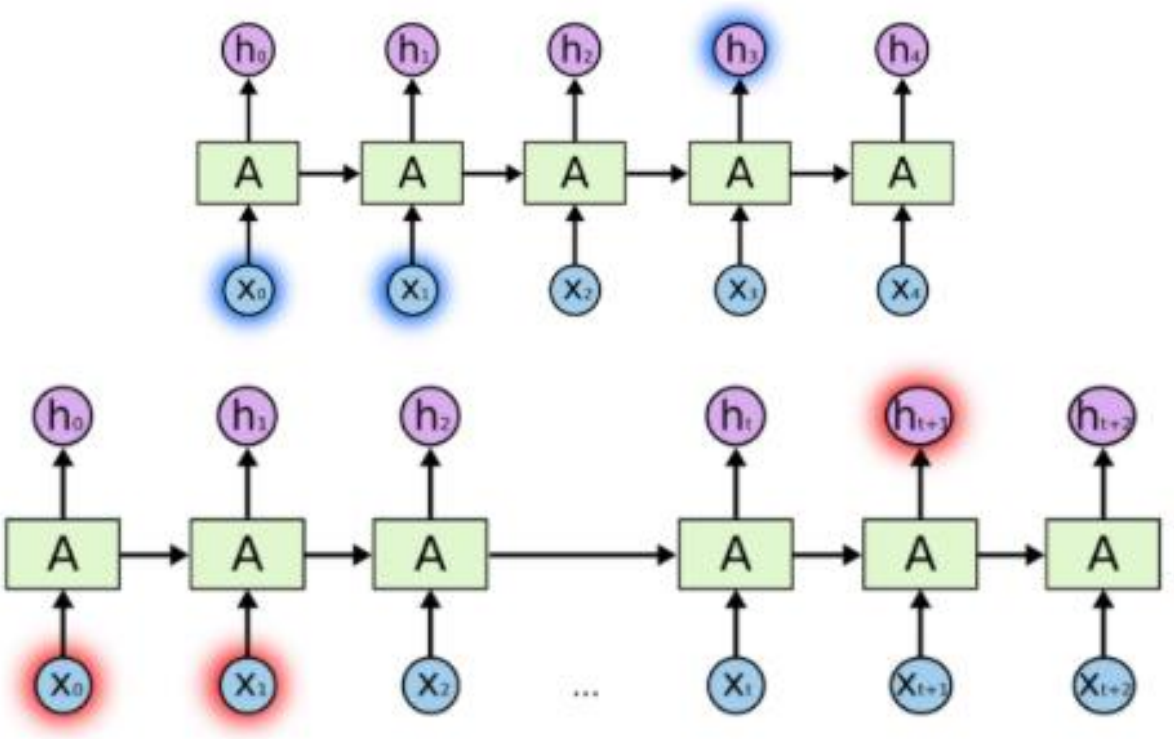
LSTM – 개념

- 단순 RNN 구조 (케라스의 SimpleRNN)
  - 실제로는 **경사 소실-발산** 등의 문제로 인해 잘 사용하지 않는다
  - 즉, 오래된 정보가 마지막 출력단에서는 매우 약해져서 학습에 사용하기 부족해진다. 따라서 step 수가 늘면 학습이 잘 되지 않는다.

- LSTM
  - RNN의 단점을 극복하기 위해서 제안
  - **여러 스텝 앞의 정보**를 놓치지 않고 **따로 뒷단으로 보내주는 채널**을 하나 더 추가(오래된 정보가 스텝을 지나면서 사라지지 않고 뒤에 영향을 미치도록 하는 것이 목적)
  - 우리가 대화를 할 때에도 바로 최근의 단어들을 듣고 뜻을 파악하지만 오래 전에 한 말을 통해서 전체적인 맥락이나 목적 등을 꾸준히 파악하는 것과 같은 의미
  - 즉, 시퀀스로 입력되는 데이터의 **단기(short) 정보**와 함께, **오래된(long) 정보**를 **병행해서 사용**하고 **학습**한다는 의미로 LSTM이라는 이름을 붙임

LSTM – Vanishing gradient problem

- RNN은 관련 정보와 그 정보를 사용하는 지점 사이의 거리가 멀 경우
  - 역전파 시, gradient 정보가 점차 줄어 들어 학습능력이 현저히 저하



LSTM – 개념

- 각 cell은 장, 단기 2개의 채널 사용
  - 단기적으로 학습한 정보가 전달되는 채널
    (현재의 입력 정보와 상태 정보에서는 필요한 부분을 선택)
  - 장기적으로 살아남아 전달되는 채널
    (과거의 전달 정보에서도 필요한 정보를 필터링하는 작업을 수행)

- LSTM에서 선택해야 할 하이퍼 파라미터
  - 임베딩 차원
  - L 출력 차원

- LSTM의 장점
  - 문서 번역
  - 질의 응답(QA)
  - 대화 서비스 (챗봇)

  등에서 좋은 성능

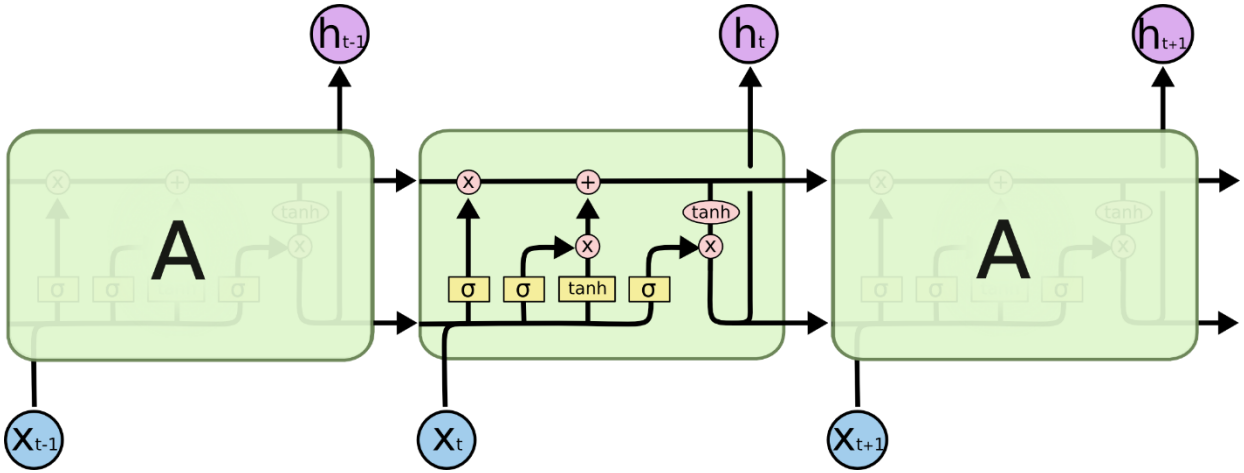
LSTM – 개념

- 내부 히든(hidden) 정보 외에 Cell state 정보를 추가로 전달
  - 선형 자기연결 정보

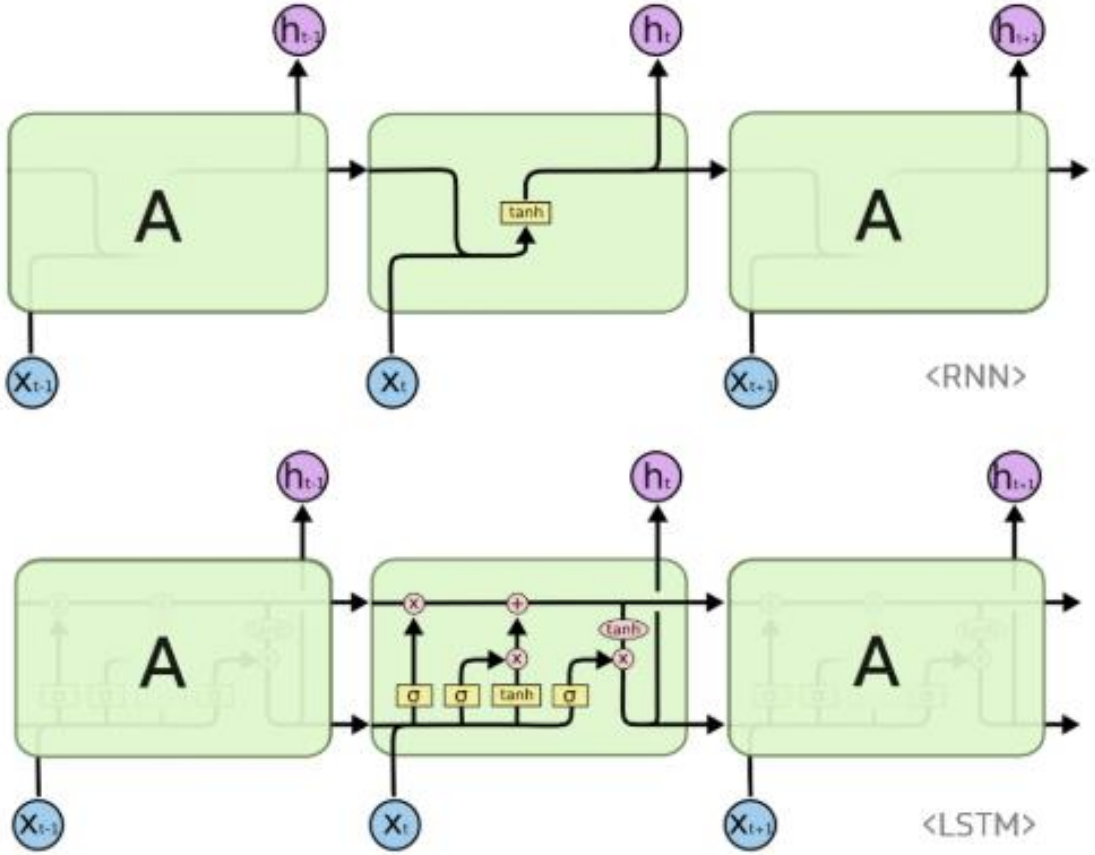
- 망각(forget), 입력(input), 출력(output) 게이트를 사용
  - 장기적인 상호작용을 학습시키는데 유용
  - 새롭고 관련성이 있는 정보를 선호하여 기억하도록 하고, 관련이 적은 정보를 잊도록 학습

LSTM – 구조

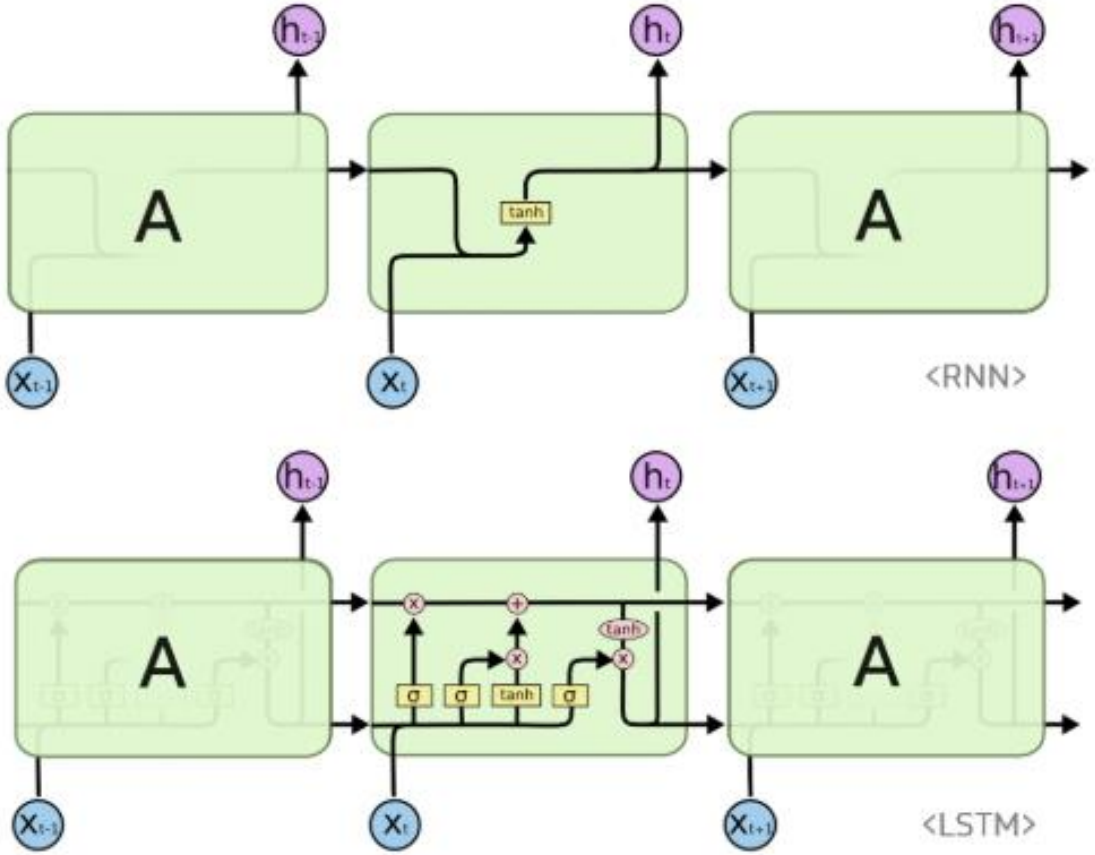
- 오래된 정보를 전달하는(carry) 별도의 채널이 있다
- 각 셀에서는
  - 입력 신호(x), 이전 단계의 상태 정보(t), 그리고 이 전달 정보(c) 세 가지 정보의 가중치 합을 구하고
  - 활성화 함수를 통과하여 출력(t)을 만든다



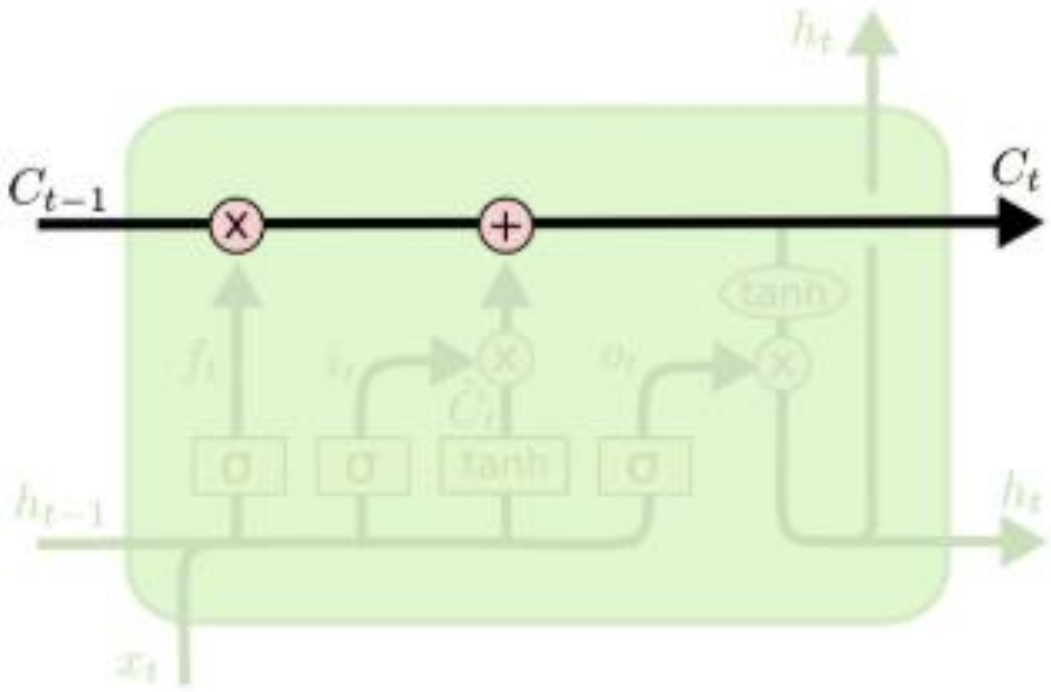
LSTM – RNN과의 비교



LSTM – RNN과의 비교

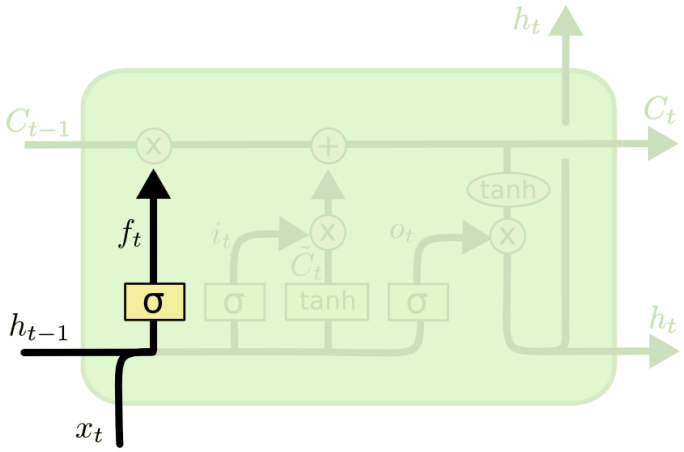


LSTM – cell state



LSTM – forget gate

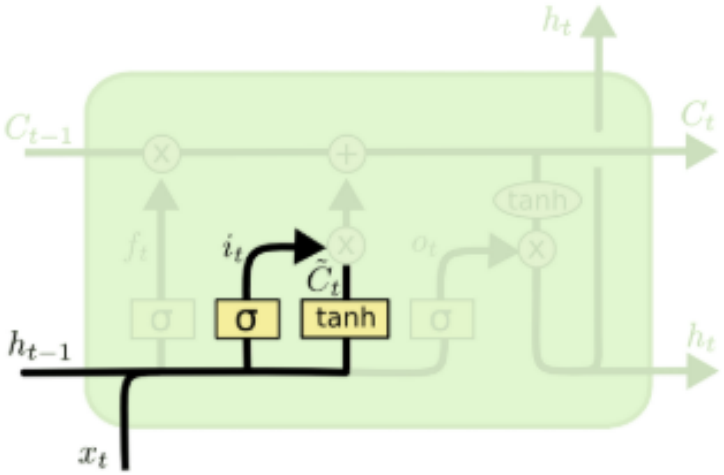
- cell state로부터 어떤 정보를 버릴 것인지를 결정
  - ➢ sigmoid layer 사용



$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

LSTM – input gate

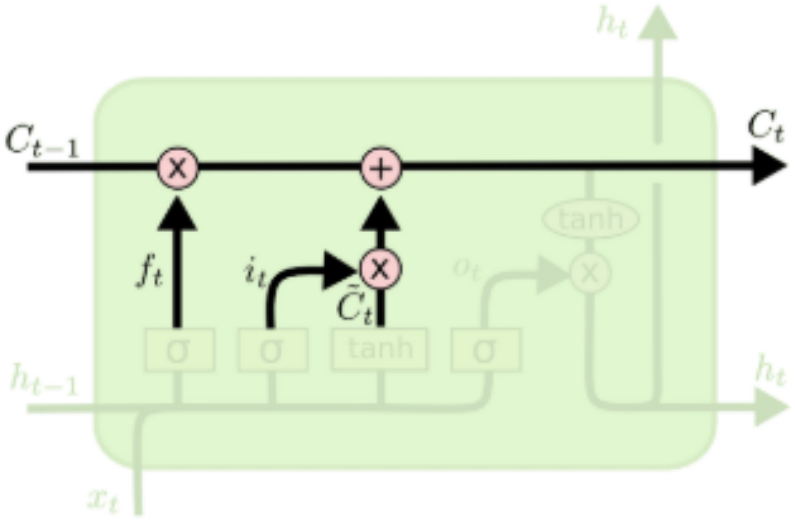
- 앞으로 들어오는 새로운 정보 중 어떤 것을 cell state에 저장할 것인지를 결정



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

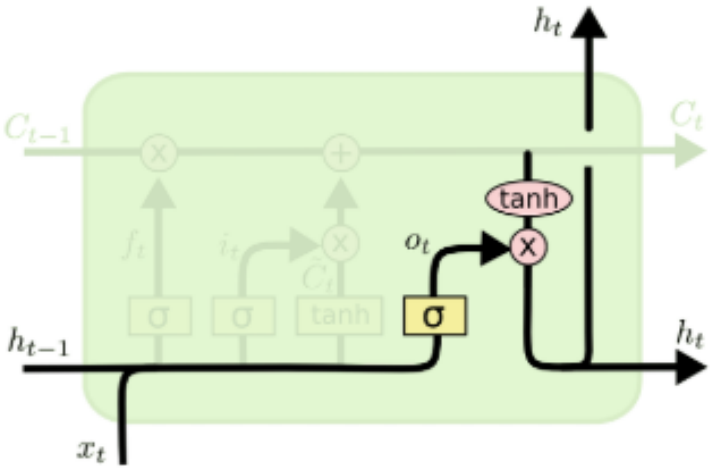
LSTM – input gate



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

LSTM – output gate

- 무엇을 output으로 내 보낼 지를 결정



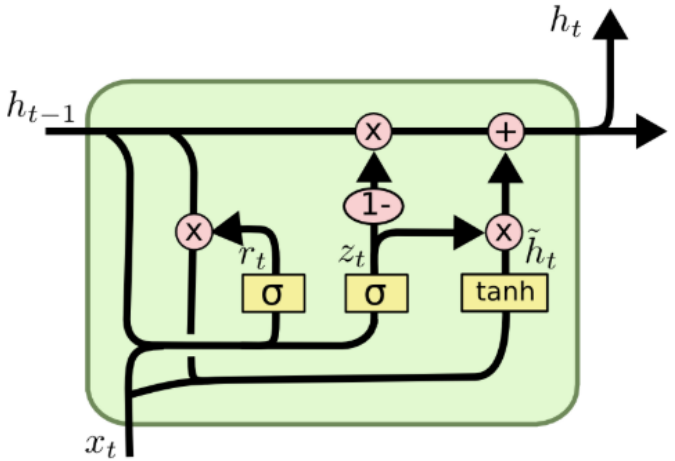
$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

GRU – 구조

- LSTM과 같은 기능을 수행, 간단한 구조 (2014)
  - 응용에 따라서 LSTM보다 성능이 우수하기도 하고 떨어지기도 함

- 2개의 게이트 사용
  - 리셋 게이트
  - 업데이트 게이트 : forget과 input 게이트를 합한 기능을 수행



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

어텐션(Attention)

어텐션 기법 – 개념

- CNN과 RNN에서 모두 **어텐션 기법**이 널리 사용
- 이미지 분석에서는 이미지의 어떤 영역이 이미지 분류에 중요한 역할을 했는지를 파악하는 것이 필요한데, 이 때 **사람이 더 집중하여 보는 영역의 개념**으로 컴퓨터가 더 중요하게 사용한 부분을 **어텐션**이라고 부름
- 자연어 처리에서는 긴 문장의 **어떤 부분이 문장의 뜻을 파악하는데 더 중요한 역할**을 하고 있는지를 파악하는 것을 어텐션 기법으로 처리

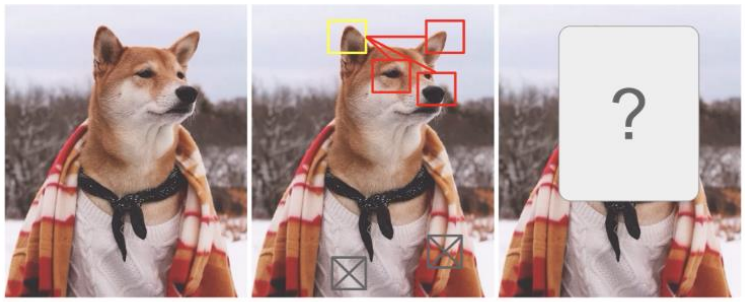


Fig. 1. A Shiba Inu in a men's outfit. The credit of the original photo goes to Instagram @menswearddog.

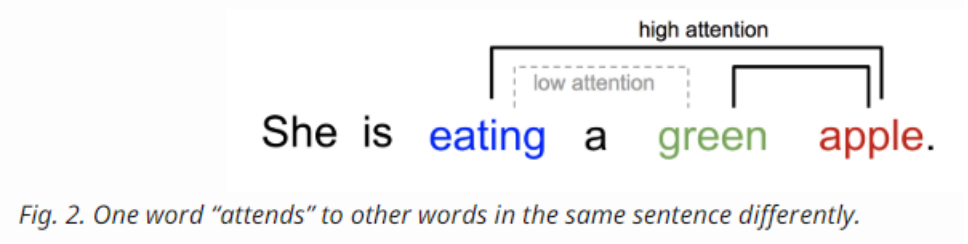
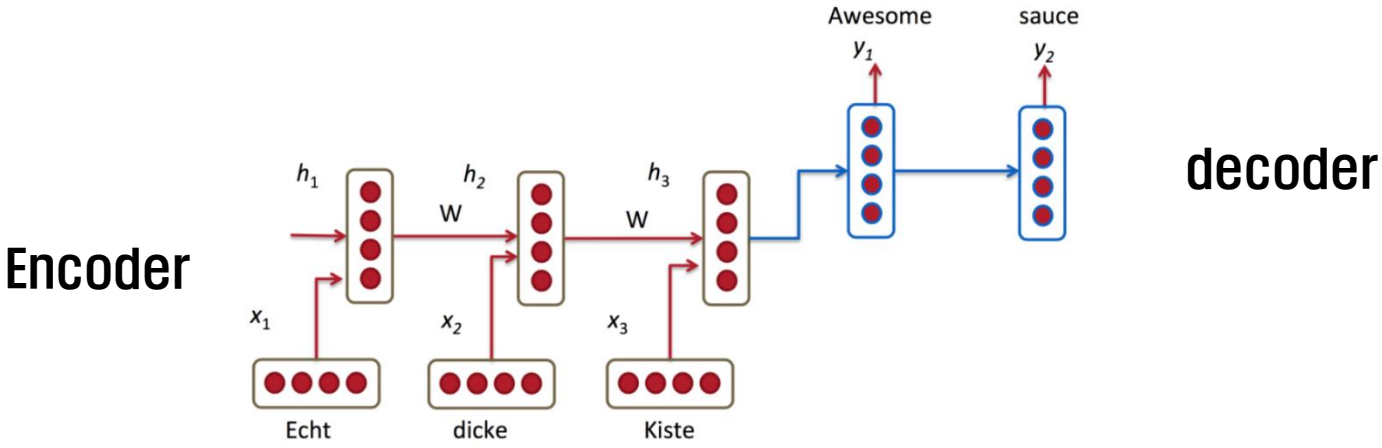


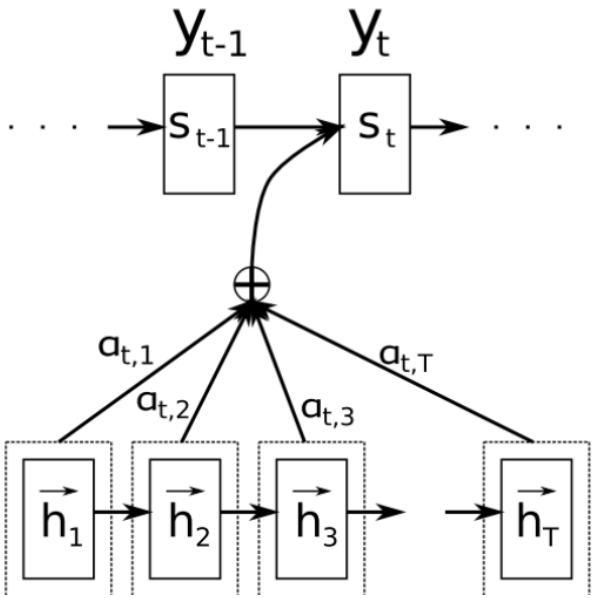
Fig. 2. One word "attends" to other words in the same sentence differently.

어텐션 기법 – 응용 예 (자동 번역기)

- RNN의 seq2seq 모델을 사용 가정
  - ➢ 디코더는 마지막 단계의 상태값인 h3에만 의존해서 번역
  - ➢ 정보가 매우 제한적
- 문장이 길어지면 오류의 문제가 커진다
  - ➢ 문장 임베딩(sentence embedding)을 수행하여 전체 문장을 하나의 벡터로 만들고 이를 디코더가 사용하는 셈
- 이를 해결하기 위해서 어텐션 기법을 도입
  - ➢ 즉, 중요한 정보에 좀 더 집중하도록 하는 방법



어텐션 기법 – 응용 예 (자동 번역기)



- 이제 디코더는 하나의 상태값에만 의존하는 것이 아니라, 출력을 얻는 각 스텝마다 입력 신호의 다른 부분에 더 주의를 집중할 수 있도록 허용

- 출력은 이제 마지막 스텝의 값만 이용하는 것이 아니라, 모든 스텝의 입력 값의 가중합(weighted sum)을 이용, 즉 shortcut connection 허용.

어텐션 기법 – 학습

- 모델은 어느 부분에 주의를 해야 하는지를, 입출력 데이터를 보고, 학습을 통해서 배운다

- 가중치 계수는 전체 합이 1이 되도록 정규화 한다.
  - 이 계수들을 보면 출력이 어떤 부분의 입력에 주의를 하고 있는지를 알 수 있다
  - 즉, 번역을 하면서 어떤 부분의 단어를 중요하게 활용하는지를 알 수 있다

- 입력과 출력의 차원이 증가하면
  - 어텐션 계수의 수가 이들의 곱에 비례하여 증가
  - 어텐션으로 계산량을 줄이고 중요한 곳에 집중해야 효과를 얻는데 잘못하면 모든 경우의 수를 다 고려하게 되어 계산량이 늘어날 수가 있다.

- 직관적으로는 인간의 주의(어텐션)와 상반되지만 이러한 기법은 현재 좋은 성능을 내고 있고 더 발전할 것으로 예상

어텐션 기법 – 응용

- 어텐션 기법은 이미지를 보고 이미지를 설명하는 캡션을 자동으로 생성하는 **이미지 캡션**에서도 사용(https://arxiv.org/abs/1502.03044)
  - 이미지를 **인코딩**할 때에는 **CNN**을 사용
  - **캡션을 생성**할 때에는 **어텐션을 도입한 RNN**을 사용

- 어텐션의 종류
  - 글로벌 어텐션 – 전체 step을 모두 사용
  - 로컬 어텐션 : 일부 step만 사용

- align 벡터를 사용하여 인코더와 디코터의 입력을 모두 고려하기도 함

- 셀프 어텐션
  - 순환 구조를 빼고 자신의 어텐션만 사용하는 방법(어텐션의 발전)
  - 성능이 우수하고 RNN을 대체하는 셈

시계열 데이터 예측

시계열 데이터의 특징

- 시계열 데이터
  - 시간 순으로 배열된 데이터
  - 주가 데이터, 센서 데이터, 클릭 스트림 데이터, 음성, 비디오 등이 포함

- 노이즈가 많고 샘플 수(차원)이 많다.
  - 이를 줄이기 위해서, 차원 축소 기술이 필요
  - 웨이블릿 분석, 디지털 필터링 등 사용

- 시계열 데이터 분석의 어려운 점
  - 분석에 효과적인 특성을 선택하는 것이 어려움(시계열 신호처리에는 도메인 전문 지식이 필요)
  - 입력과 출력의 관계가 고정적이지 않고 시간에 따라 종속(같은 입력 시퀀스라도 다른 시간대에 입력되면 다른 결과, 시간 종속성 기간이 정해져 있지도 않음)
  - 주파수 도메인으로도 표현(주파수 특성이 분석에 더 나은 경우에는 주파수 도메인에서 분석해야 한다)

시계열 데이터의 특징

- 시 불변성 (Stationality)
  - 시간 이동에 불변인 특성 추출이 중요
    (영상 처리에서는 이미지의 이동, 스케일링, 회전에 영향을 적게 받는 특성 추출이 필요)
  - 이러한 특성을 만족하는 것을 stationarity를 보장한다고 함. 즉, 시간이 이동해도 평균(mean), 분산(variance), 자기상관(autocorrelation) 값 등이 일정한 것을 말함

- 저량과 유량
  - 시계열 데이터 분석은 저량(stock)과 유량(flow)로 나누어 분석한다. 일반적으로 유량을 예측하는 경우가 많다
  - 신경망을 사용하면 계절적 요인, 트렌드를 신경망이 학습한다는 장점이 있다.
  - 데이터 양이 적을 때에는 전통적인 ARIMA 모델이 신경망보다 성능이 우수함

응용 분야

- 주식 예측
  - 주가 변동은 랜덤하다고 알려져 있음
  - 신경망을 사용하여 뉴스 키워드 등 수많은 요인을 반영하면서 주식 변화 예측도 어느 정도는 가능해질 것으로 예상
  - 관련 데이터를 더 많이 사용하면 예측률이 올라갈 수 있으나 현재는 만족스럽지 못한 수준

- 음성인식
  - 화자 인식, 성별 인식, 음성 인식(텍스트 변환), 음향 모델링 등에 사용
  - Mel-frequency cepstral coefficients 등의 신호변환 기술이 사용
  - HMM도 오랫동안 음성 인식에 사용되었음
  - LSTM RNN이 널리 사용

- 공장의 운영 감시
  - 공장의 운영에서 발생하는 센서 데이터를 보고 장비가 정상적으로 동작하는지, 고장의 징후가 있는지 등을 분석하는 요구가 증가 추세
  - 기기의 운영 정보 뿐 아니라 공장의 환경 데이터와 제품의 생산에서 생산품의 품질이 유지되는지를 분석하는 것도 필요

응용 분야

- 냄새 센서 데이터
  - 음식이 상한 것, 공기 오염, 가스 유출, 박테리아 검출, 질병 진단 등
  - 신경망을 로봇에 사용하여 냄새나는 곳을 찾는 동작 수행하기도 함
  - 레이블링 작업이 어려워서 비지도 학습 방법이 필요
  - 비지도 학습에 의한 특성 추출 기술로 상한 고기 찾기 예
    http://dx.doi.org/10.3390/s130201578

- 생물학적 데이터 분석
  - EEG, ECG, MEG 등 기타 건강 센서 데이터 분석에 필요
  - 대개 데이터 분량이 방대, 전문가의 레이블링 작업에 많은 비용 부담
  - 비지도 학습에 의한 특성 학습은 주로 비디오 신호에서 연구되었지만 일반 시계열 데이터에서 점차 중요해지고 있음
  - 장시간의 변화를 파악하려면 장시간의 데이터가 필요하고 분석이 어려움
  - 시계열 데이터는 양이 많아서 **전처리된 특성**을 **사용**하기도 함

ARIMA Model

- Stationarity (정상성, 시불변성)
  - 시계열 (time sequence)의 특징이 해당 시계열이 관측된 시간에 무관
  - 따라서, 추세나 계절성이 있는 시계열은 정상성을 나타내는 시계열이 아님.
    (추세와 계절성은 서로 다른 시간에 시계열의 값에 영향을 줄 것이기 때문)
  - 반면에, 백색잡음(white noise) 시계열은 정상성을 나타내는 시계열임.
    (언제 관찰하는지에 상관 없고, 어떤 시점에서 보더라도 똑같이 보일 것임)

ARIMA (p, d, q) Model

- **AR (AutoRegressive Model, 자기회귀모델) -> 과거 반영**
  - t 시점의 Y값이 과거의 Y값에 직접적으로 의존하는 구조
  - t 시점의 Y값이 과거 p기간 동안의 Y값의 가중 평균과 t 시점에서 발생하는 오차항(Random Shock)의 합으로 구성

$$y_t = \emptyset_0 + \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \ \ldots + \emptyset_p y_{t-p} + \varepsilon_t$$

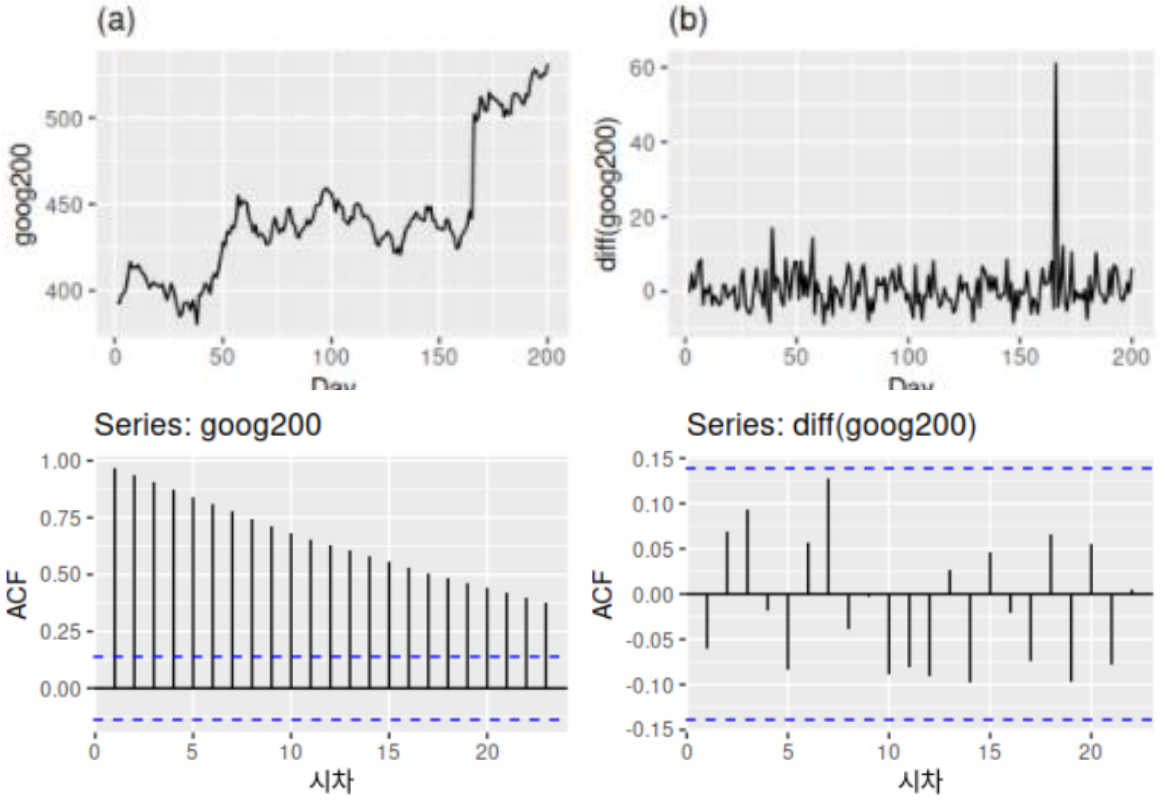
- **MA (Moving Average Model, 이동평균모델) -> 미래 예측**
  - 과거 예측 오차(forecast error)을 이용 (white noise)
  - 오차항에 의한 충격의 효과(memory)는 q 기간 지속

$$y_t = \theta_0 + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ \ldots + \theta_q \varepsilon_{t-q}$$

- **ARMA Model**

$$y_t = \emptyset_0 + \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \cdots \emptyset_p y_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \ \ldots + \theta_q \varepsilon_{t-q}$$

ARIMA Model



시계열 분석 모델

- Google stock price is non-stationary in (a), but the daily changes are stationary in (b).
- one way to make a non-stationary time series stationary:
  - compute the differences between consecutive observations.
    called "**differencing (차분)**" – parameter, d