



3주차 모의경진대회 교통 물류 통행량 시계열 예측 과제 특강

(주)마인즈앤컴퍼니 | 김태훈 매니저

Index

1. 과제 - 교통 물류 통행량 시계열 예측 과제
2. Recurrent Neural Network
3. Gradient Descent & Gradient Vanishing
4. 모델 - LSTM
5. 평가지표 - RMSE
6. 성능 향상 방안

교통 물류 통행량 시계열 예측 과제

개요

과거의 물류 통행량 시계열 데이터를 통한 미래 물류 통행량 시계열 예측

35개 도로의 시간별 통행량 데이터를 학습, 이를 기반으로 미래의 통행량량을 예측하는 모델 개발

데이터셋

• 데이터 구조

- train.csv : 35개 도로의 2020.01.01 ~ 05.17 기간에 대한 도로 통행량 데이터
- validation.csv : 35개 도로의 2020.05.11 ~ 05.24 기간에 대한 도로 통행량 데이터
- test.csv : 35개 도로의 2020.05.18 ~ 05.31 기간에 대한 도로 통행량 데이터
- 정리하면, 35개 도로의 2020.01 ~ 2020.05.24 기간에 대한 도로 통행량 데이터가 주어진 상태에서 35개 도로의 2020.05.25 ~ 2020.05.31 기간에 대한 도로 통행량 데이터를 예측해야함 (필요에 따라 데이터를 통합해 train / validation 기간 재설정 가능)

변수

• 변수 구성

- Input : 학습 기간 동안의 35개 도로의 시간별 통행량 및 날짜와 시간 정보
- Output : 예측 기간 동안의 35개 도로의 시간별 통행량 및 날짜와 시간 정보

	날짜	시간	10	100	101	120	121	140	150	160	...	1020	1040	1100	1200	1510	2510	3000	4510	5510	6000
0	20200101	0	83247	19128	2611	5161	1588	892	32263	1636	...	1311	3482	11299	7072	1176	3810	748	3920	2133	3799
1	20200101	1	89309	19027	3337	5502	1650	1043	35609	1644	...	1162	3849	13180	8771	1283	3763	782	3483	2057	4010
2	20200101	2	66611	14710	2970	4631	1044	921	26821	1104	...	768	2299	7986	5426	1536	3229	491	2634	1526	3388
3	20200101	3	53290	13753	2270	4242	1021	790	21322	909	...	632	1716	5703	3156	1104	2882	431	2488	1268	3686
4	20200101	4	52095	17615	2406	3689	1840	922	22711	1354	...	875	2421	5816	2933	1206	2433	499	2952	1927	5608

train.csv

Index

1. 과제 - 교통 물류 통행량 시계열 예측 과제
2. Recurrent Neural Network
3. Gradient Descent & Gradient Vanishing
4. 모델 - LSTM
5. 평가지표 - RMSE
6. 성능 향상 방안

RNN(Recurrent Neural Network)

Previous methods

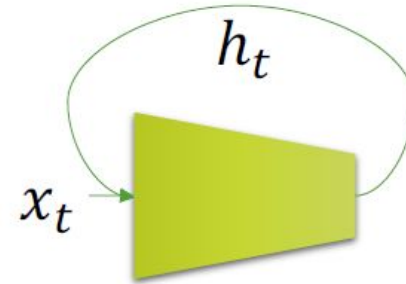
- e.g. Tabular Data, Image



$$y = f(x; \theta)$$

Recurrent Neural Networks

- e.g. Sequential Data, Time-series



$$h_t = f(x_t, h_{t-1}; \theta)$$



Sequential Data vs Time Series

- Time-stamp의 유무에 따른 차이
 - 시퀀셜 데이터는 데이터의 순서 정보가 매우 중요함
 - e.g. 텍스트 문장: 단어의 순서
 - 추가로 시계열 데이터는 해당 데이터가 발생한 시각 정보가 매우 중요함
 - e.g. 주식 데이터: 가격의 순서 및 발생 시점

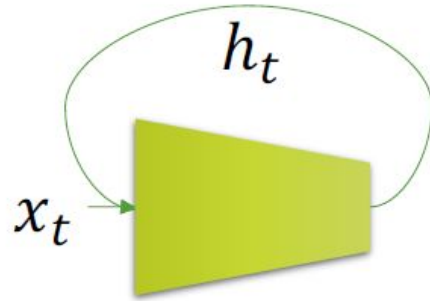
Sequential Data

- 텍스트
- (샘플링 주기가 일정한) 영상/음성

Time Series

- 주식 데이터
- 센서 데이터

How RNN Works

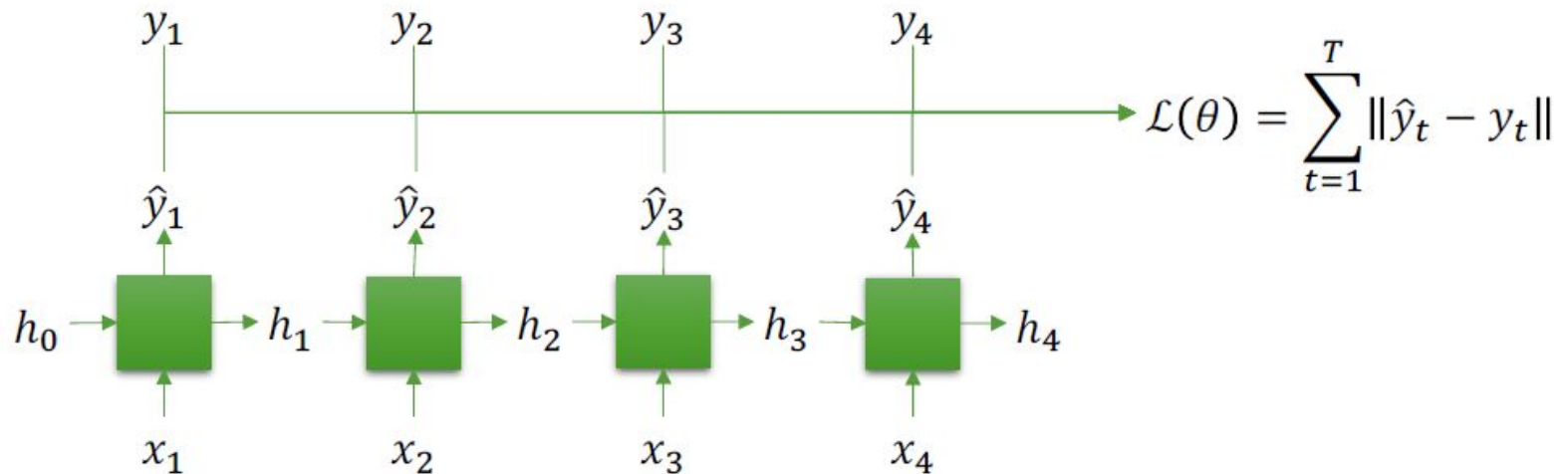


$$\hat{y}_t = h_t = f(x_t, h_{t-1}; \theta)$$

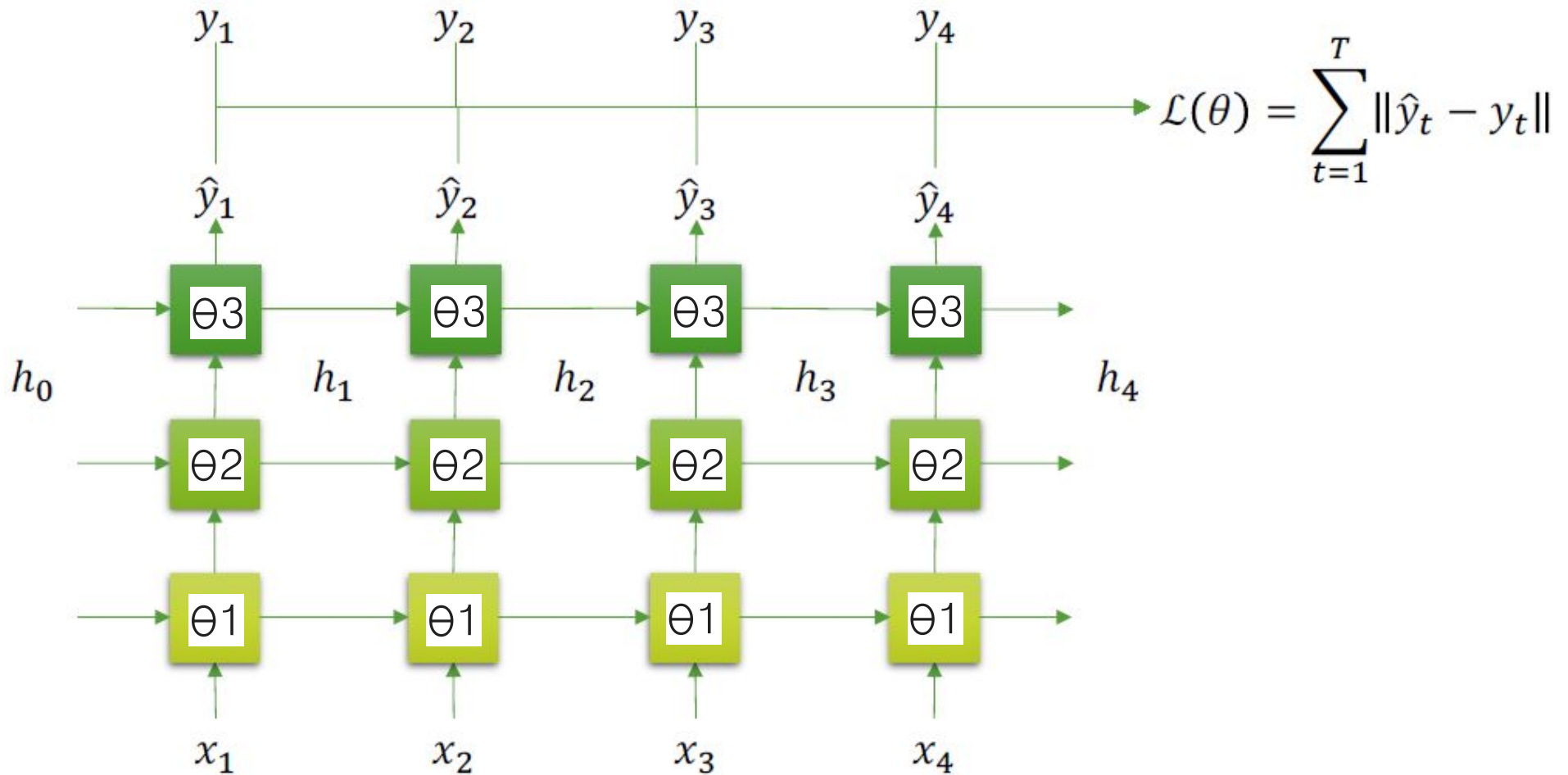
$$= \tanh(W_{ih}x_t + b_{ih} + W_{hh}h_{t-1} + b_{hh})$$

where $\theta = \{W_{ih}, b_{ih}, W_{hh}, b_{hh}\}$.

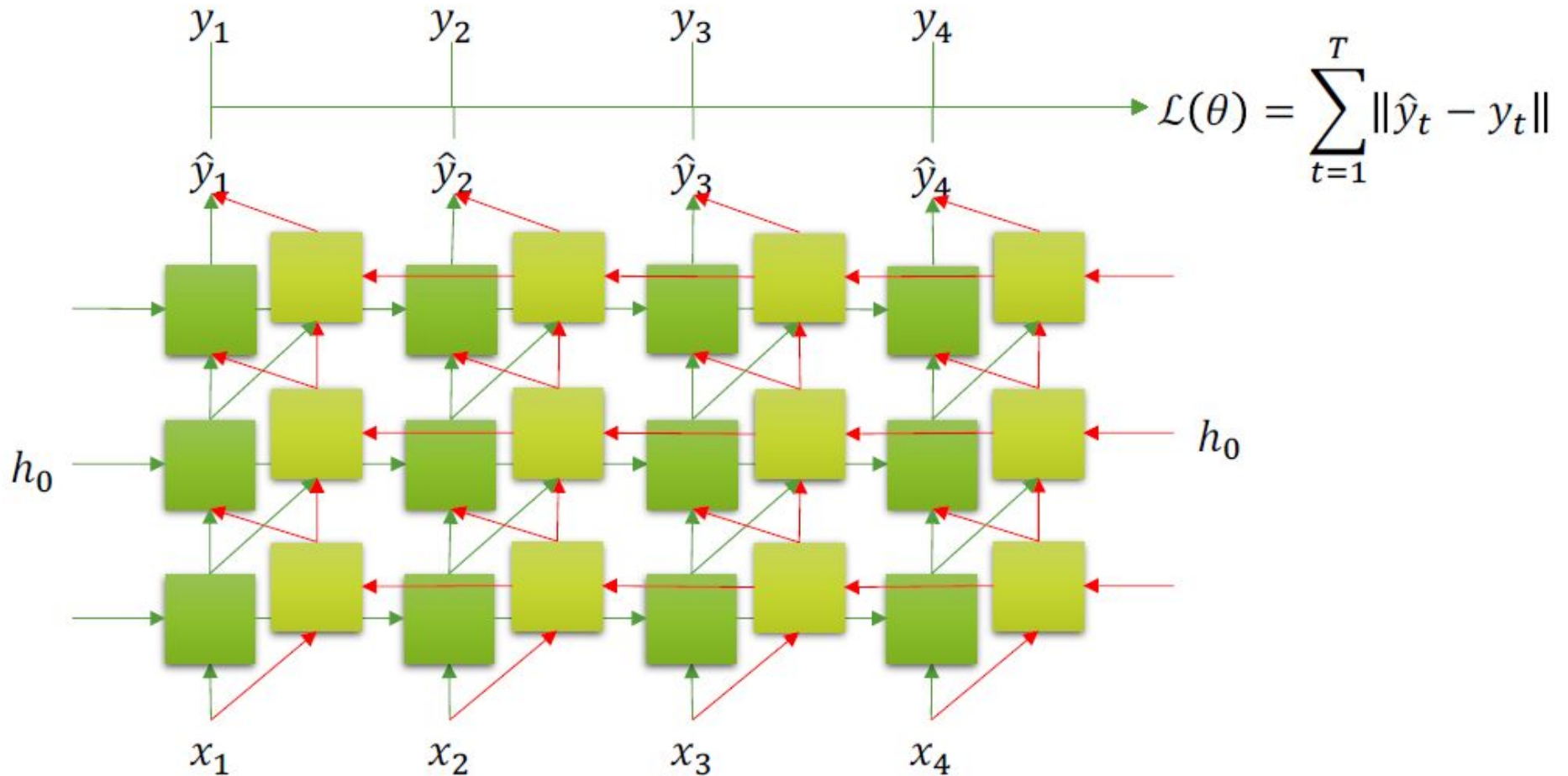
$$h_t = f(x_t, h_{t-1}; \theta)$$



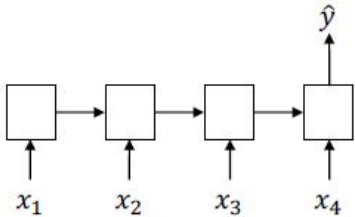
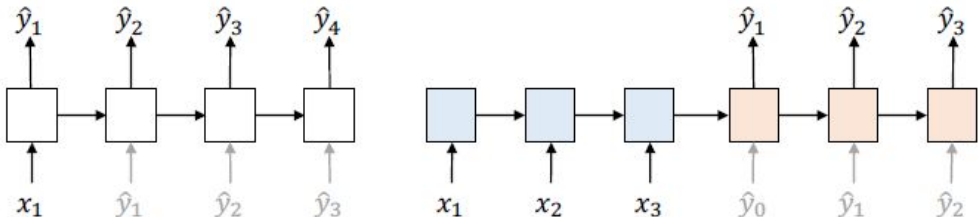
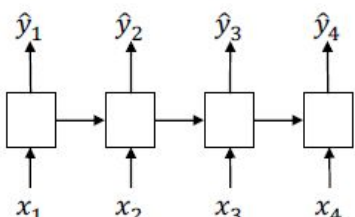
Multi-layered RNN



Bidirectional Multi-layered RNN

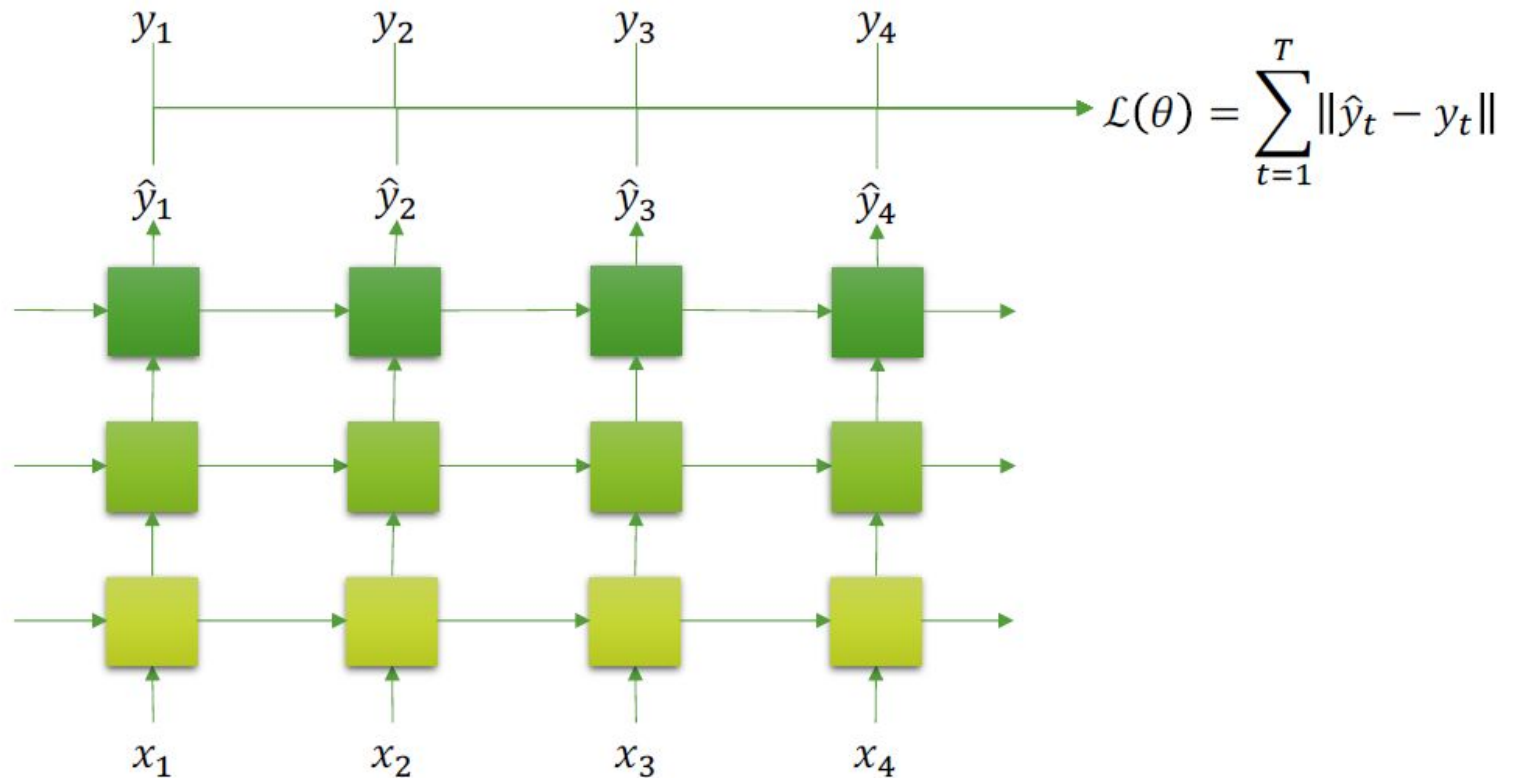


RNN 활용 사례

Type	Architecture	Applications
Many to One		Text Classification
One to Many		NLG, Machine Translation
Many to Many		POS Tagging, MRC

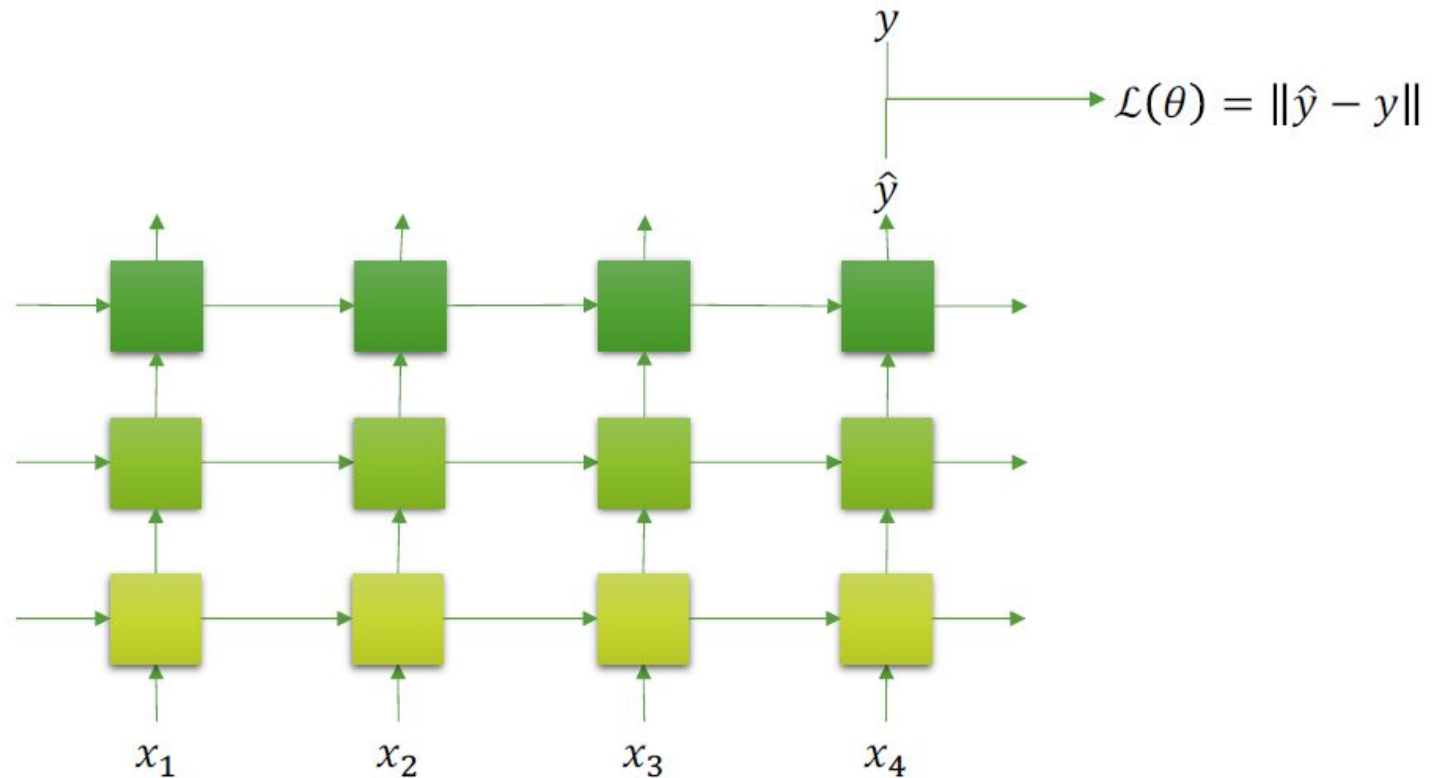
Many to Many

- e.g. POS Tagging



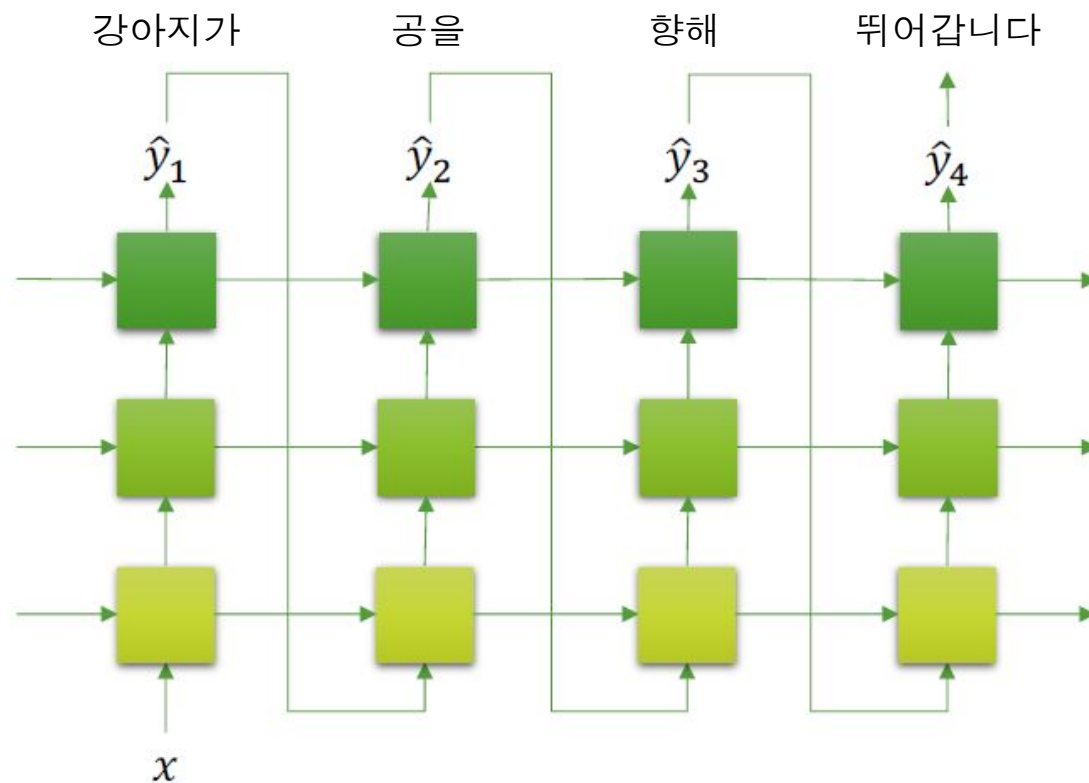
Many to One

- e.g. Text Classification



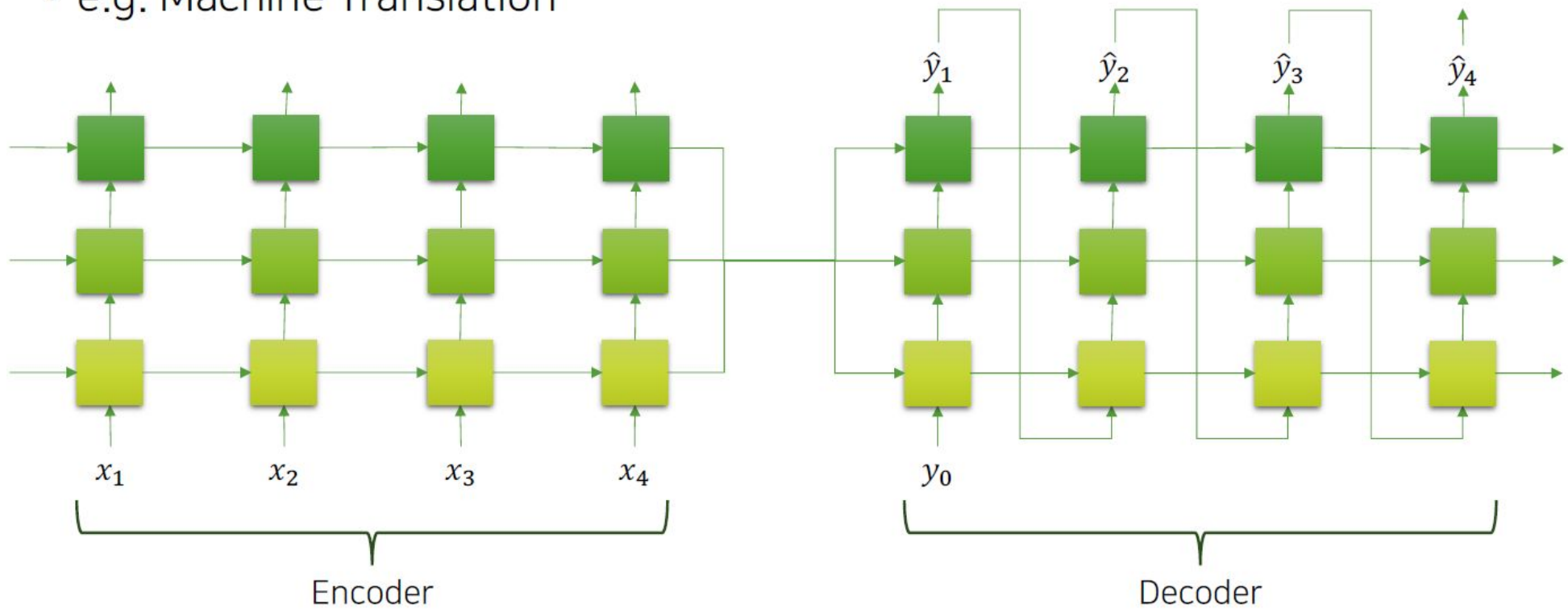
One to Many

- e.g. Natural Language Generation



One to Many (Sequence-to-Sequence)

- e.g. Machine Translation



Index

1. 과제 - 교통 물류 통행량 시계열 예측 과제
2. RNN
3. Gradient Descent & Gradient Vanishing
4. 모델 - LSTM
5. 평가지표 - RMSE
6. 성능 향상 방안

Gradient Descent

gradient

미국·영국[ˈɡreɪdɪənt]  영국식 

1 (특히 도로·철도의) 경사도

a steep **gradient** 

급경사도

descent

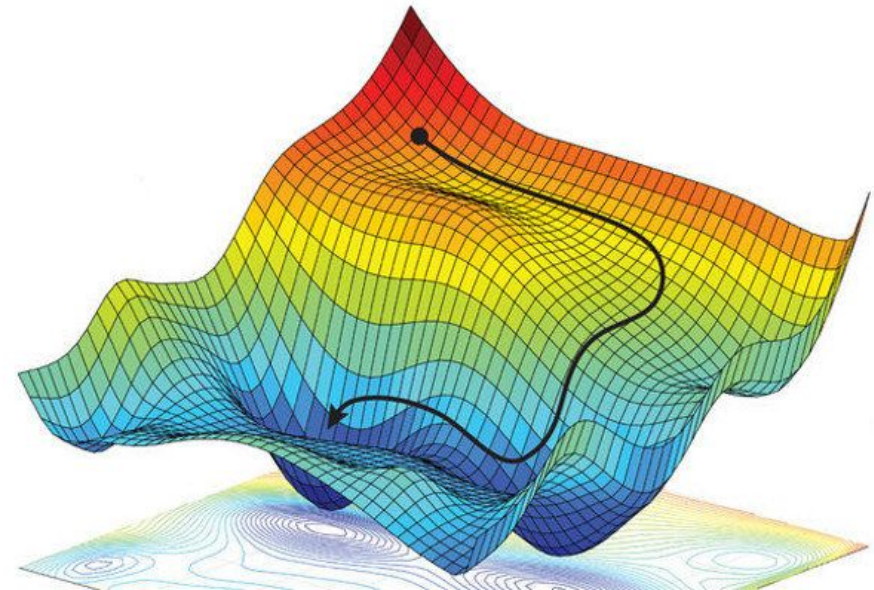
미국·영국[drɪˈsent]  영국식 

1 내려오기, 내려가기, 하강, 강하 (↔ascent)

2 내리막 (↔ascent)

Gradient Descent(경사하강법)이란?

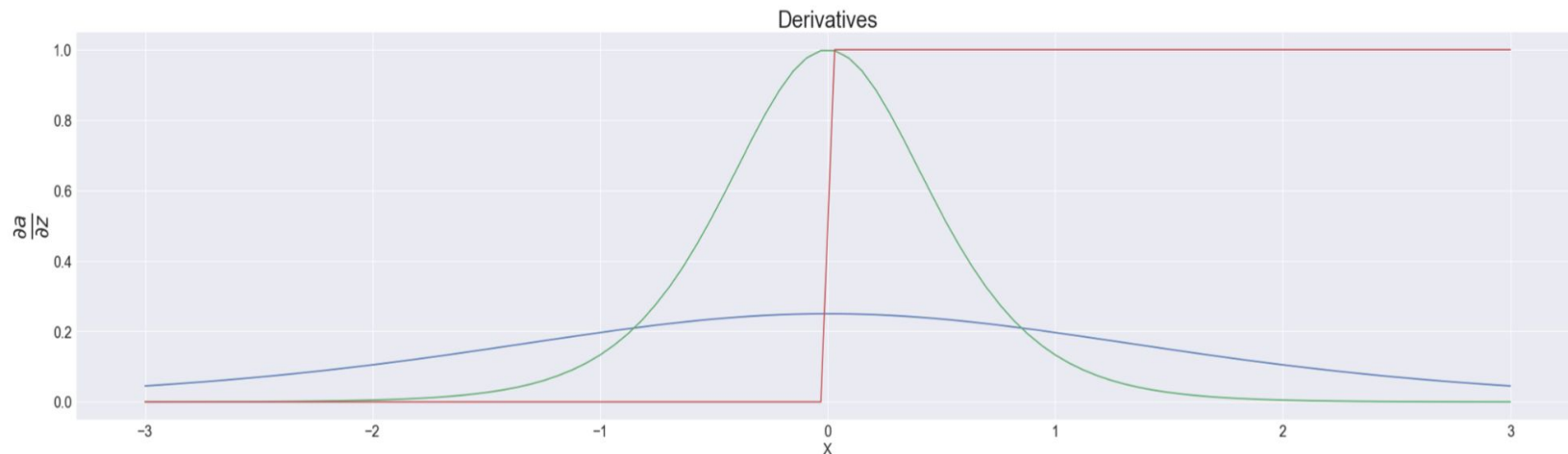
- Loss와 모델의 n 개의 파라미터들로 구성된 $(n+1)$ 차원의 고차원 평면에서 Loss가 가장 가파르게 떨어지는 방향으로 파라미터들을 업데이트해 나가는 방법
- 경사가 가파른 정도를 판단하기 위해 Loss를 파라미터로 편미분함



Gradient Vanishing

sigmoid, tanh의 도함수는 아래와 같다. 딥러닝은 back propagation 과정에서 활성화함수의 기울기만큼 웨이트가 개선된다. 그런데 기울기의 최대가 tanh의 경우 1, sigmoid는 0.3정도에 불과하다. 즉 tanh나 sigmoid를 사용하면 대부분의 경우 활성화함수의 기울기가 1보다 작다. 이때 딥러닝 모델의 layer가 많다면, back propagation 과정에서 1보다 작은값이 계속해서 곱해지게 된다. 이러면 기울기가 무한히 작아지는 현상이 발생하며 이 현상을 gradient vanishing 현상이라고 한다.

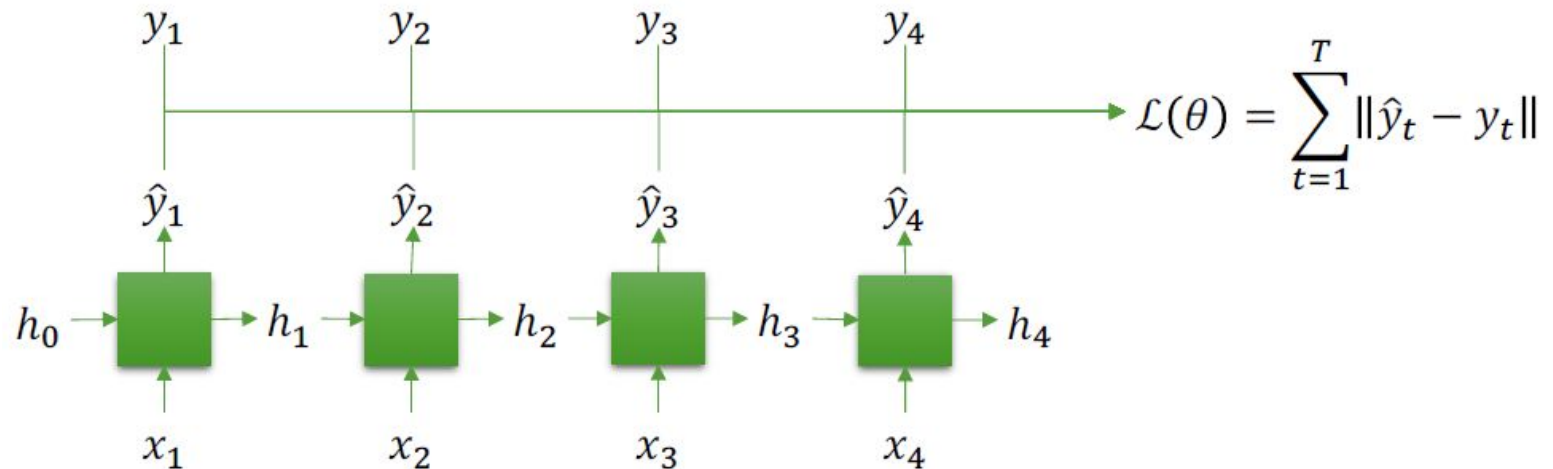
이때 gradient vanishing을 해결하기 위해 나온 활성화함수가 relu이다. relu 함수는 $x > 0$ 일 때, 기울기가 1이기 때문에 곱해서 기울기가 작아지는 현상이 발생하지 않기 때문이다.



Gradient Vanishing in Vanilla RNN

- RNN 내부에는 tanh가 있으므로, time-step이 길어짐에 따라, gradient vanishing이 발생함
 - 따라서 긴 시퀀스는 학습이 어려움

$$\begin{aligned}\hat{y}_t = h_t &= f(x_t, h_{t-1}; \theta) \\ &= \tanh(W_{ih}x_t + b_{ih} + W_{hh}h_{t-1} + b_{hh}) \\ \text{where } \theta &= \{W_{ih}, b_{ih}, W_{hh}, b_{hh}\}.\end{aligned}$$



Index

1. 과제 - 교통 물류 통행량 시계열 예측 과제
2. RNN
3. Gradient Descent & Gradient Vanishing
4. 모델 - LSTM
5. 평가지표 - RMSE
6. 성능 향상 방안

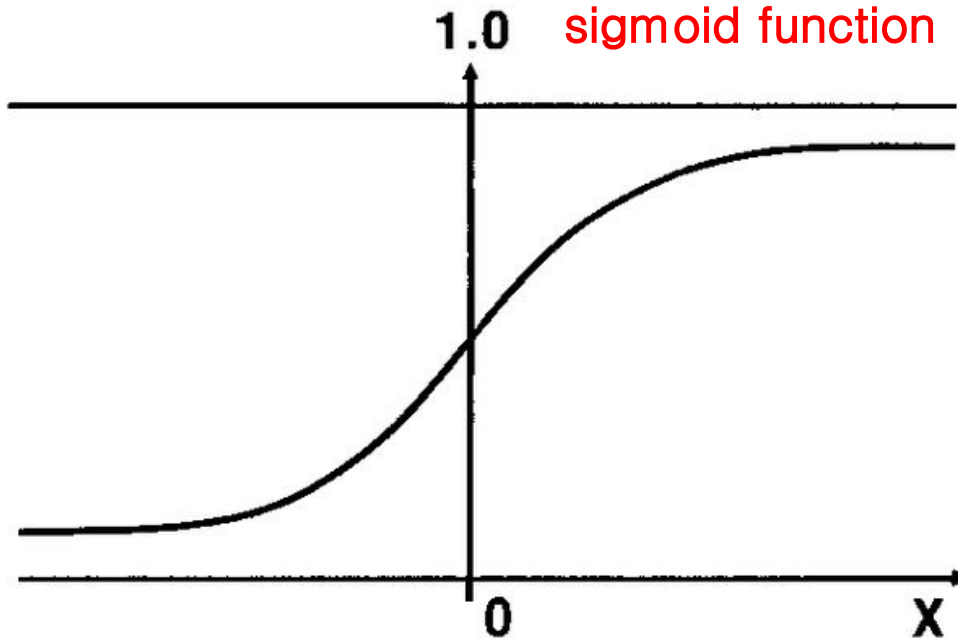
Gate Using Sigmoid

- Sigmoid는 0과 1사이의 값을 반환하므로, sigmoid를 곱하면 마치 문을 열고 닫는 듯한 효과를 낼 수 있음.

$$y = \sigma(x) \times f(x)$$



sigmoid function

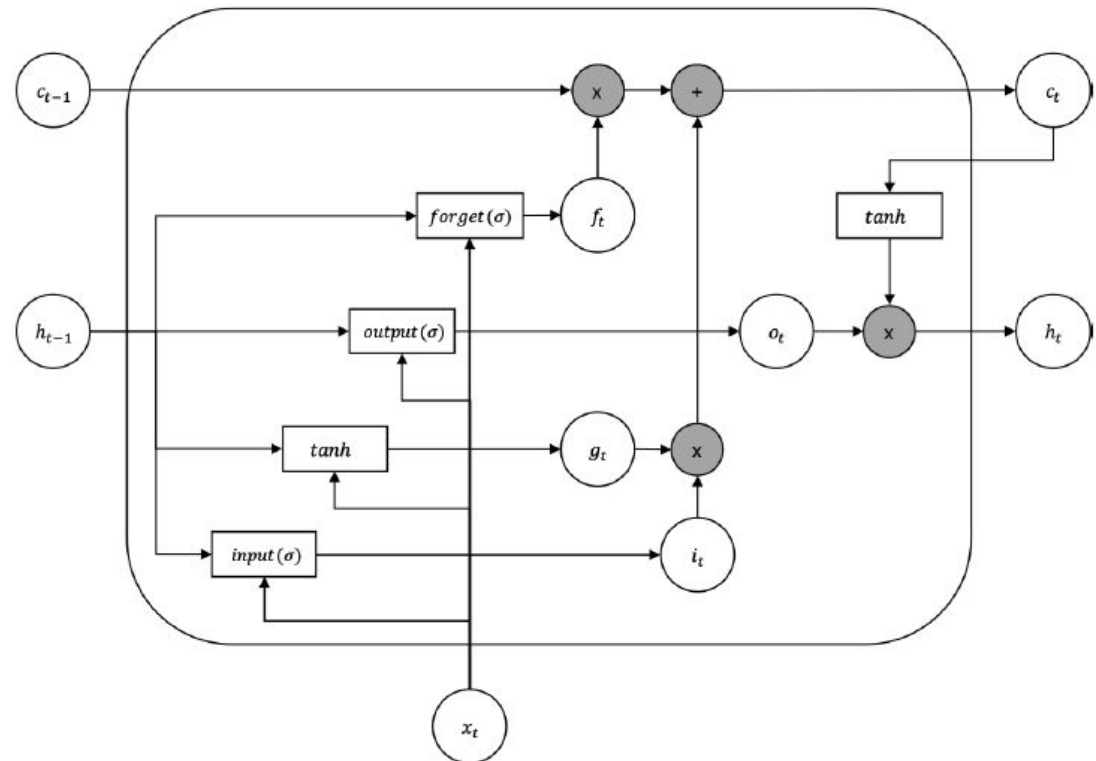


Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

LSTM(Long Short Term Memory)

$$\begin{aligned} i_t &= \sigma(W_i \cdot [x_t, h_{t-1}]) \\ f_t &= \sigma(W_f \cdot [x_t, h_{t-1}]) \\ g_t &= \tanh(W_g \cdot [x_t, h_{t-1}]) \\ o_t &= \sigma(W_o \cdot [x_t, h_{t-1}]) \\ c_t &= f_t \otimes c_{t-1} + i_t \otimes g_t \\ h_t &= o_t \otimes \tanh(c_t) \end{aligned}$$



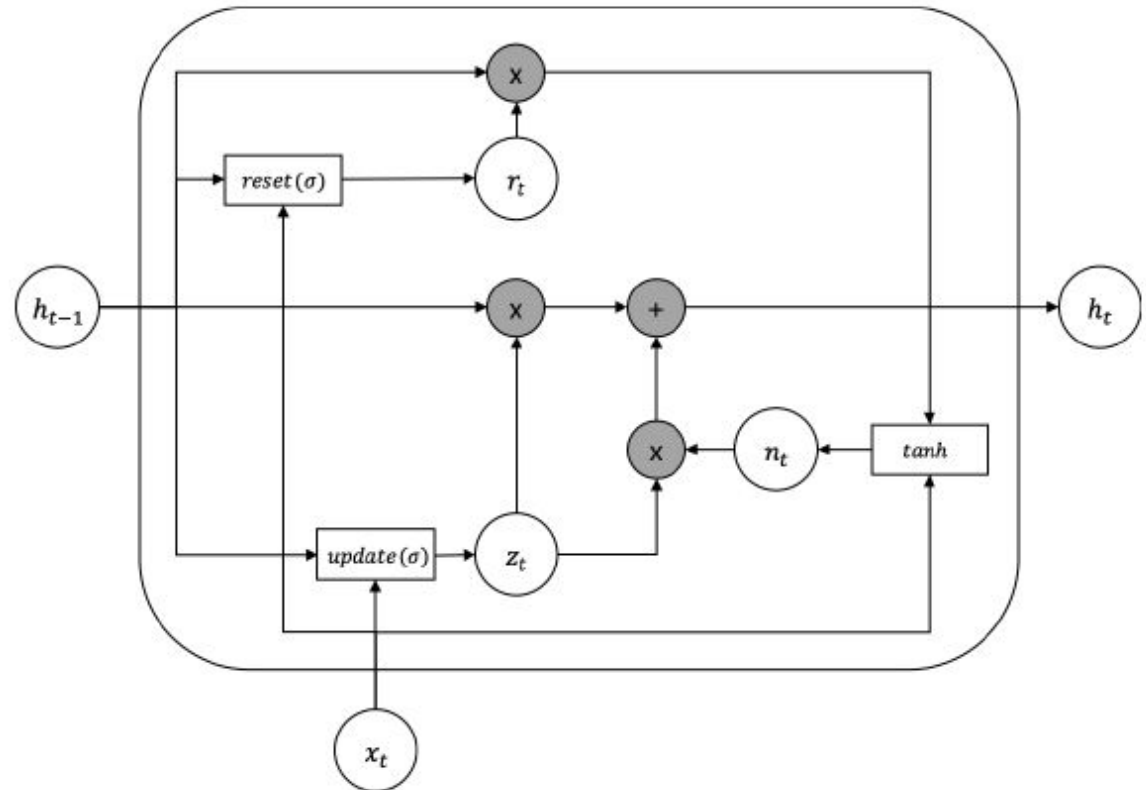
GRU(Gated Recurrent Unit)

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

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

$$n_t = \tanh(W_n \cdot [x_t, r_t \otimes h_{t-1}])$$

$$h_t = (1 - z_t) \otimes n_t + z_t \otimes h_{t-1}$$



Index

1. 과제 - 교통 물류 통행량 시계열 예측 과제
2. 횡단면 / 시계열 / 패널 데이터
3. Gradient Descent & Gradient Vanishing
4. 모델 - LSTM
5. 평가지표 - RMSE
6. 성능 향상 방안

RMSE(Root Mean Squared Error)

MAE(Mean Absolute Error)

$$MAE = \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{n}$$

n : number of errors

y_i : 실제값

\hat{y}_i : 예측값

- 오차들의 절대값의 평균
- 절대값을 취해 매우 직관적인 지표
- MSE보다 특이치에 영향을 적게 받음

MSE(Mean Squared Error)

$$MSE = \sum_{i=1}^n \frac{(y_i - \hat{y}_i)^2}{n}$$

- 오차들의 제곱의 평균
- 특이치에 민감

MAPE(Mean Absolute Percentage Error)

$$MAPE = \frac{\sum \left| \frac{y - \hat{y}}{y} \right|}{n} * 100\%$$

- 오차율들의 절대값의 평균
- MSE보다 특이치에 영향을 적게 받음

Index

1. 과제 - 교통 물류 통행량 시계열 예측 과제
2. 횡단면 / 시계열 / 패널 데이터
3. Gradient Descent & Gradient Vanishing
4. 모델 - LSTM
5. 평가지표 - RMSE
6. 성능 향상 방안

성능 향상 방안

다른 모델 사용

파라미터 조정

Data Engineering

- 개별 도로에 대한 LSTM
- GRU, RNN, Prophet, ARIMA 등
- number of layers, num_epochs 등
- (LSTM은 시간축 방향의 gradient vanishing은 막아주지만, 세로축 방향으로 층을 깊게 쌓았을 때 발생하는 gradient vanishing은 막아주지 못하여 number of layers를 지나치게 크게 하면 gradient vanishing이 발생함)
- train.csv : 35개 도로의 2020.01.01 ~ 05.17 기간에 대한 도로 통행량 데이터
- validation.csv : 35개 도로의 2020.05.11 ~ 05.24 기간에 대한 도로 통행량 데이터
- test.csv : 35개 도로의 2020.05.18 ~ 05.31 기간에 대한 도로 통행량 데이터
- 정리하면, 35개 도로의 2020.01 ~ 2020.05.24 기간에 대한 도로 통행량 데이터가 주어진 상태에서
- 35개 도로의 2020.05.25 ~ 2020.0531 기간에 대한 도로 통행량 데이터를 예측해야함
- (필요에 따라 train / validation 기간 재설정 가능)



End of document

Contact: (주)마인즈앤컴퍼니 고석태 대표 (stko@mnc.ai)