# Team corona
# Project Summary

허찬 이정진 이나영
조민제 이지현

# Sentiment Analysis

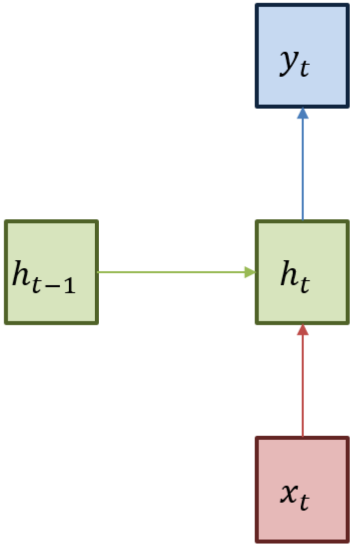
# 1. Sentiment Analysis


## 목표: 코로나 사태 이후 관련 기사 댓글들의 감정 분석

- ‘사회적 거리두기’, ‘자가격리’ 키워드

# 2.1) RNN 모델

- 기본 구조




$$y_t = W_{hy}h_t + b_y$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

**특징: Memory**

**→ '연속성'을 가지는 시퀀스 데이터 기억, 처리 가능**

**Xt:** 입력 벡터

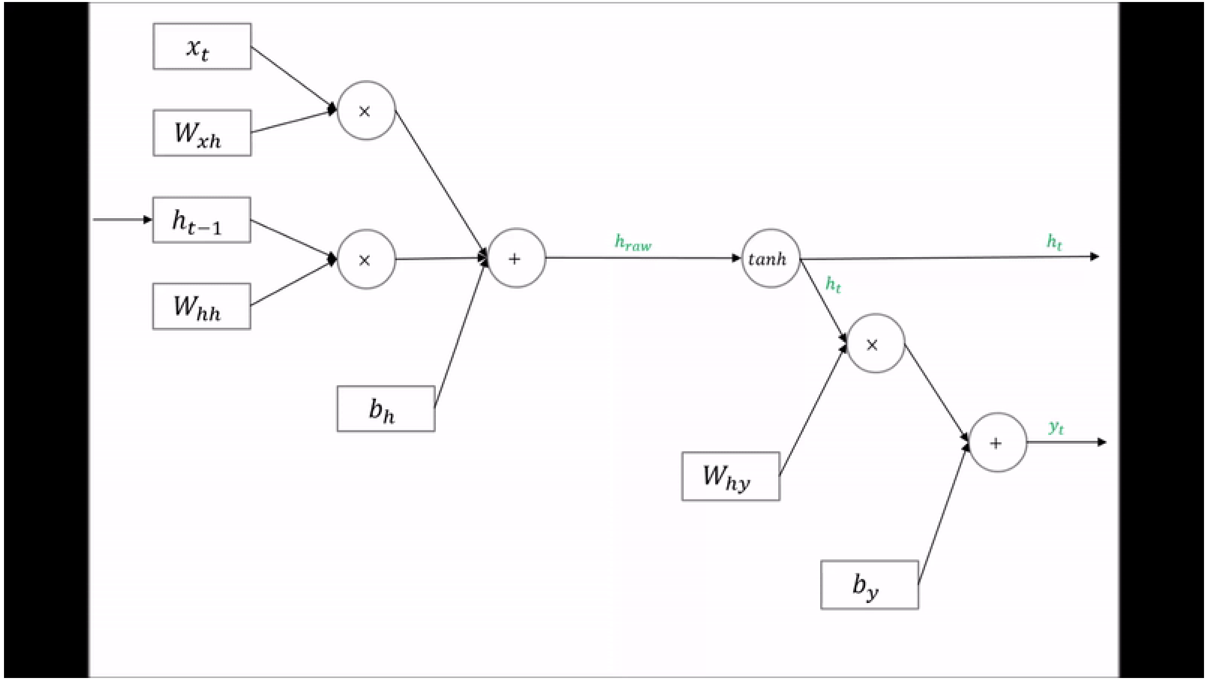
**ht:** 현재 기억

**h(t-1):** 과거 기억

**Yt:** 출력 벡터

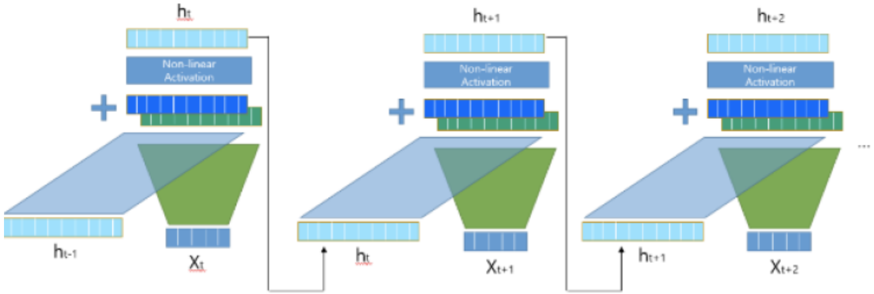
# 2.1) RNN 모델

- **Back Propagation**



# 2.1) RNN 모델

- 단점: **Vanishing Gradient**




$$h_{t-2} = tanh(W[h_{t-3}, x_{t-2}])$$

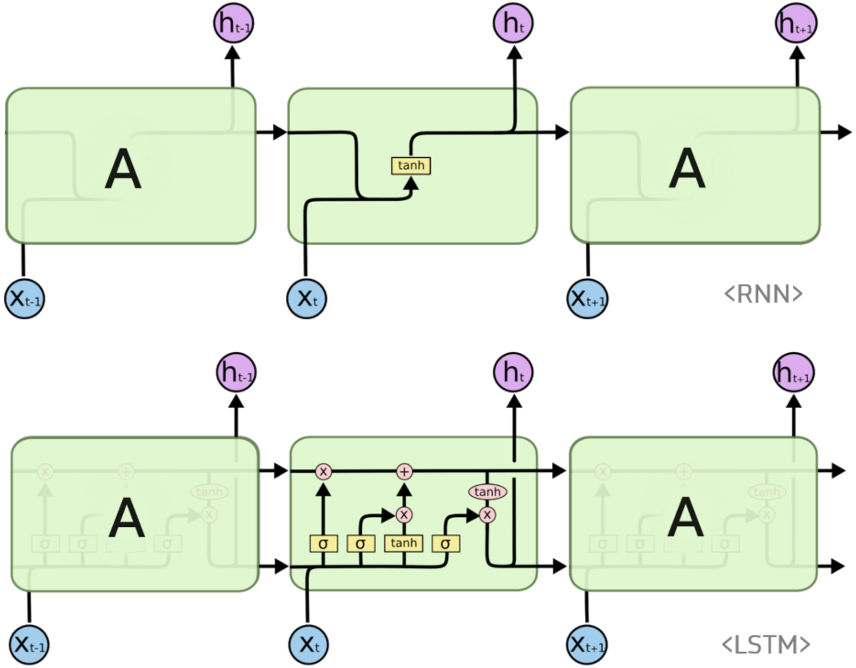
$$h_{t-1} = tanh(W[h_{t-2}, x_{t-1}])$$

$$h_t = tanh(W[h_{t-1}, x_t])$$

$$h_t = tanh(W[tanh(..tanh(..h_{t-3})), x_t])$$

So many tanh(x)!

# 2.2) LSTM 모델




Gating variables

$$\mathbf{f}_t = \sigma\left(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_t\right)$$

$$\mathbf{i}_t = \sigma\left(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i\right)$$

$$\mathbf{o}_t = \sigma\left(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o\right)$$

Candidate (memory) cell state

$$\tilde{\mathbf{c}}_t = \tanh\left(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c\right)$$

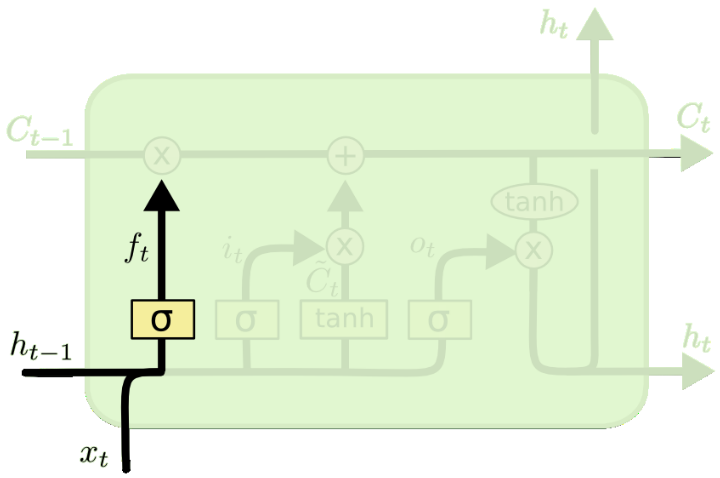
Cell & Hidden state

$$\mathbf{c}_t = \mathbf{f}_t \circ \mathbf{c}_{t-1} + \mathbf{i}_t \circ \tilde{\mathbf{c}}_t$$

$$\mathbf{h}_t = \mathbf{o}_t \circ \tanh(\mathbf{c}_t)$$

# 2.2) LSTM 모델

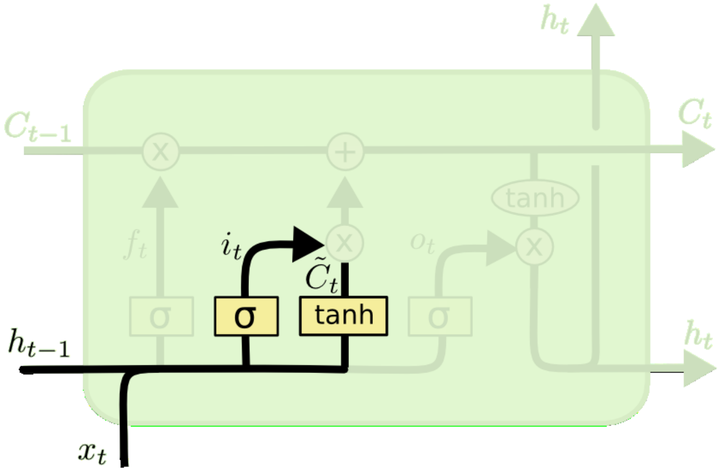
- 망각 게이트: 시간 t에 따른 정보의 중요도에 따라 얼마나 잊어버릴지 결정




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

# 2.2) LSTM 모델

- 입력 게이트: 시간 t에 따른 정보의 중요도에 따라 얼마나 기억할지 결정




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

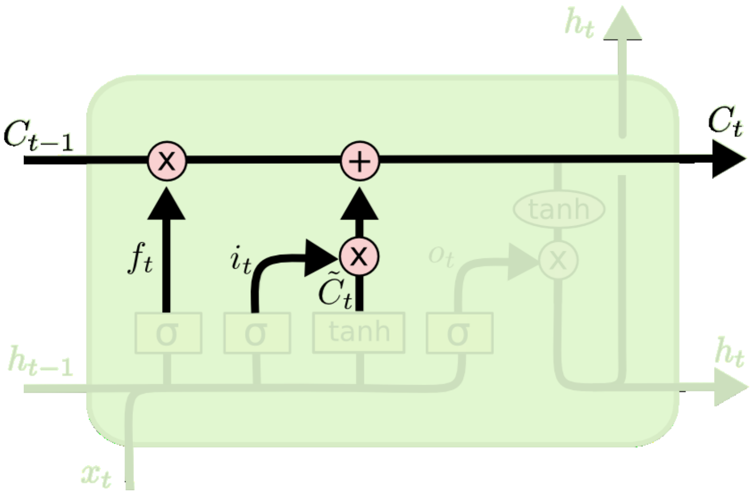
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$



KU-BIG

# 2.2) LSTM 모델

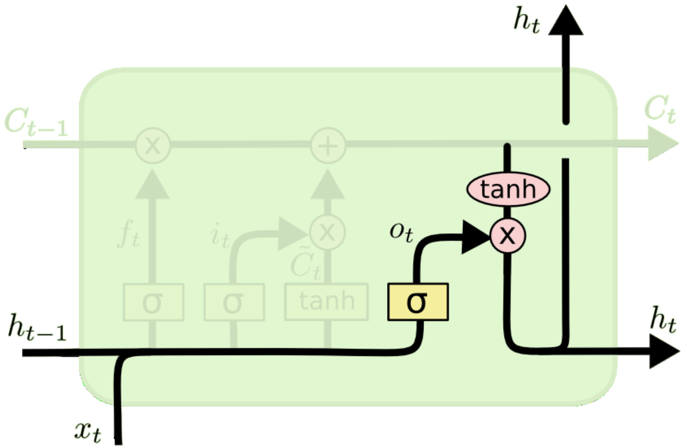
- cell-state (장기 상태): 과거의 기억과 현재의 기억을 얼마나 받아들일지 결정




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

# 2.2) LSTM 모델

- 출력 게이트: output 정보와 cell-state 정보를 모두 고려




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

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

# 2.3) BLSTM 모델

- LSTM의 단점: “이후 step이 이전 step에 영향을 준다”는 점을 고려하지 못함

“나는 늦잠을 자다가 계절학기 (신청)을 못했다.”

VS

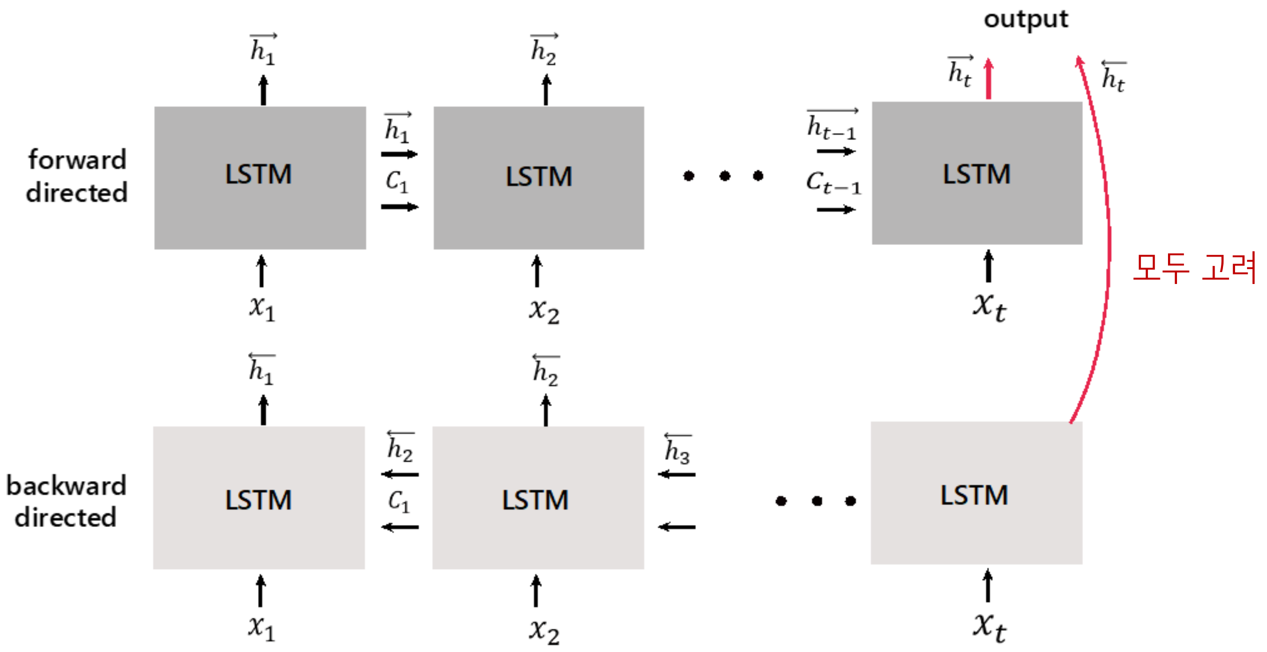
“나는 (막차)를 놓쳐 집까지 20분을 걸어왔다.”

→ 정방향 추론과 역방향 추론 모두 고려 필요



KU-BIG

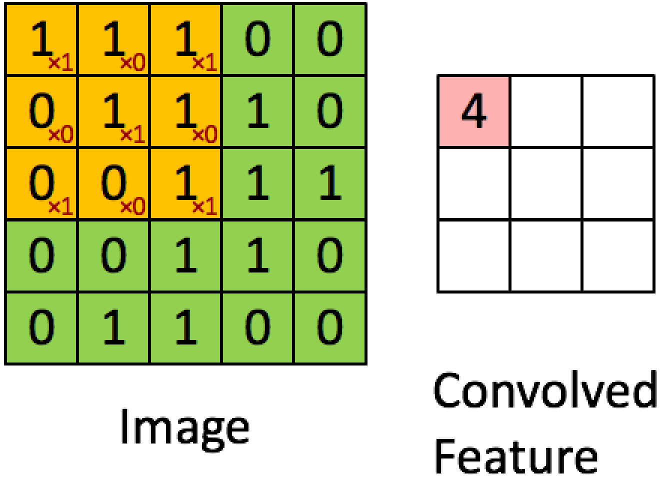
# 2.3) BLSTM 모델



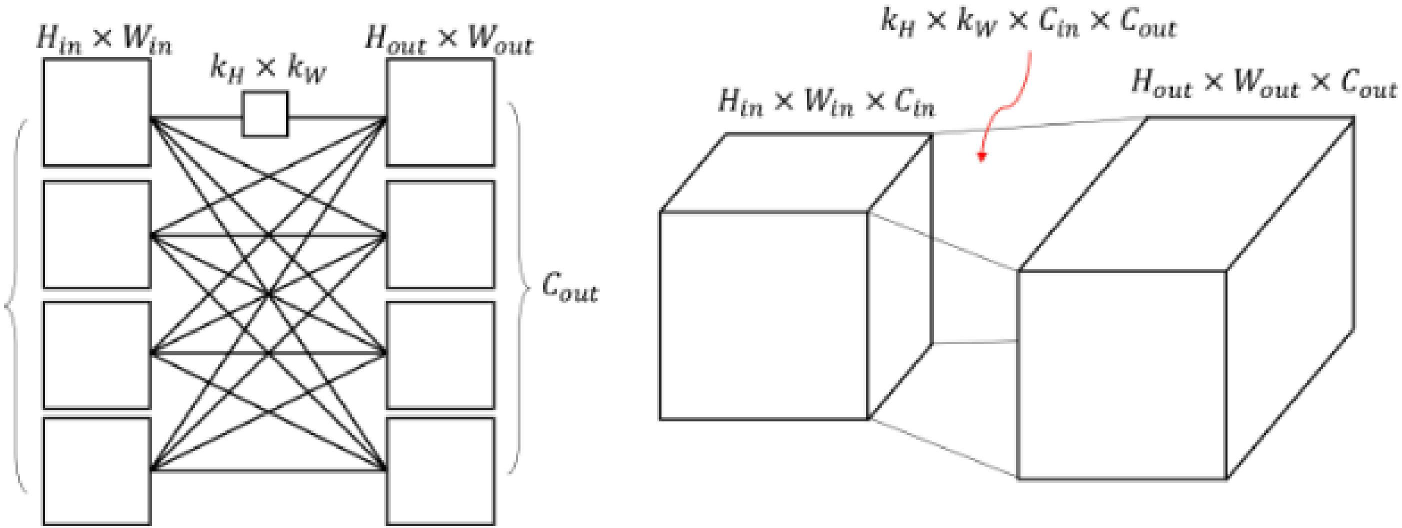

→ **Bidirectional LSTM: forward directed LSTM + backward directed LSTM**

# 3.1) CNN 모델

- 이미지 데이터와 같은 다차원 배열로 이루어진 데이터 처리 가능
- Convolution(합성곱), Pooling(풀링)과 같은 기본 연산 수행



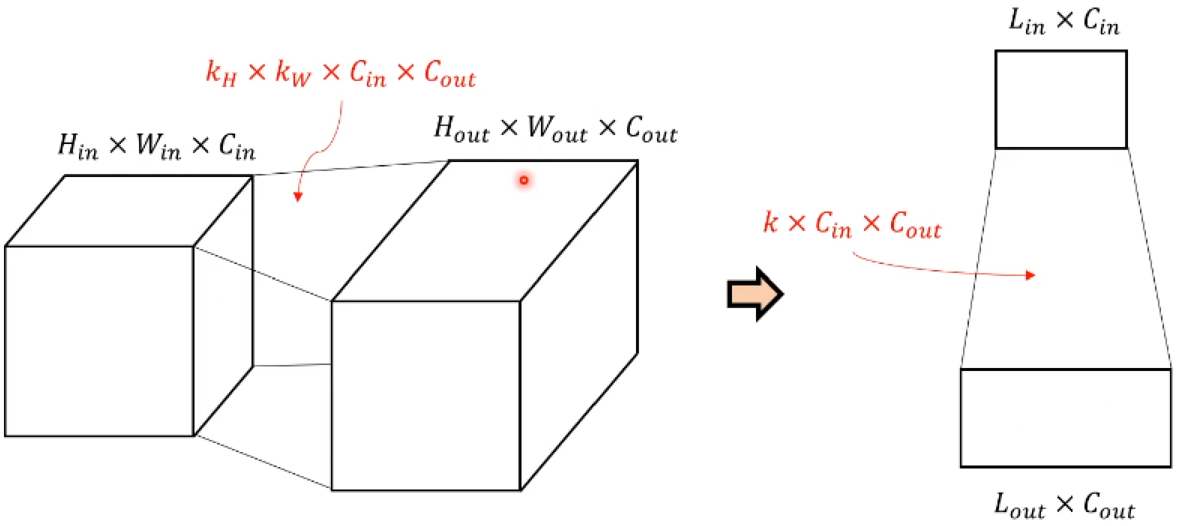
# 3.2) 2D-CNN vs. 1D-CNN






KU-BIG

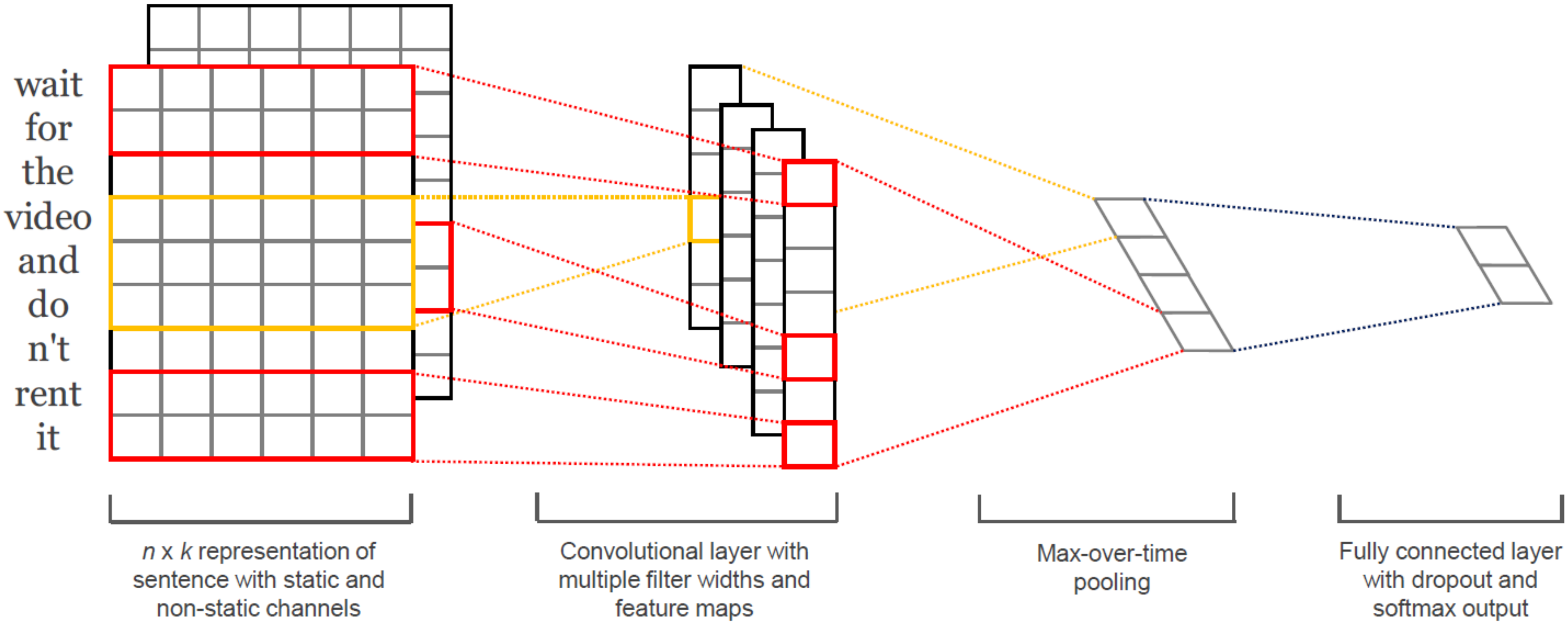
# 3.2) 2D-CNN vs. 1D-CNN



# 3.3) Text-CNN

| 단어 | V1 | V2 | V3 | V4 | … | Vp-2 | Vp-1 | Vp |
|---|---|---|---|---|---|---|---|---|
| W1 | | | | | | | | |
| W2 | | | | | | | | |
| W3 | | | | | | | | |
| … | | | | | | | | |
| Wn-2 | | | | | | | | |
| Wn-1 | | | | | | | | |
| Wn | | | | | | | | |

# 3.3) Text-CNN



wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

# 3.3) Text-CNN

1단계: $x_i \in \mathbb{R}^k$ $\mathbf{x}_{i:i+j} = \mathbf{x}_i \oplus \mathbf{x}_{i+1} \oplus \ldots \oplus \mathbf{x}_{i+j}.$

k-dimension의 단어 벡터

문장: n개의 단어를 합침

2단계: $\mathbf{w} \in \mathbb{R}^{h \times k}$ $c_i = f(\mathbf{w} \cdot \mathbf{x}_{i:i+h-1} + b)$

h*k 크기의 필터

feature map 만들기

3단계: $\hat{c} = \max\{\mathbf{c}\}$ $y = \mathbf{w} \cdot (\mathbf{z} \odot \mathbf{r}) + b$

중요한 값 추출

정규화 결과

# Visualization and Variable Selection

# Using ggplot2

- ✓ Layered Grammar of Graphics
- ✓ Hadley Wickham
- ✓ Usage

  ggplot(data) + geom_function(mapping =aes(mapping))

# Using ggplot2

✓ Data

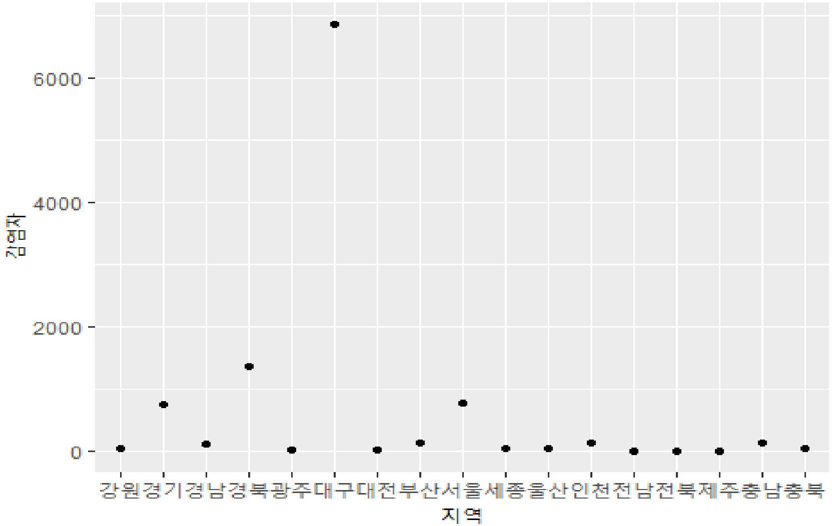
```
colnames(corona)

##  [1] "지역"              "감염자"             "사망자"              "격리병상"
##  [5] "선별진료소"         "단란주점.술."        "유흥주점.클럽."       "신천지수.추정."
##  [9] "인구"              "인구밀도"            "X1 인당.소득수준"
```

# Using ggplot2

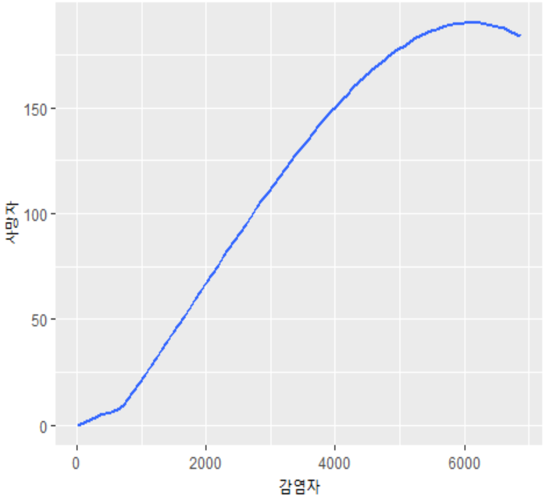
✓ Data Visualization

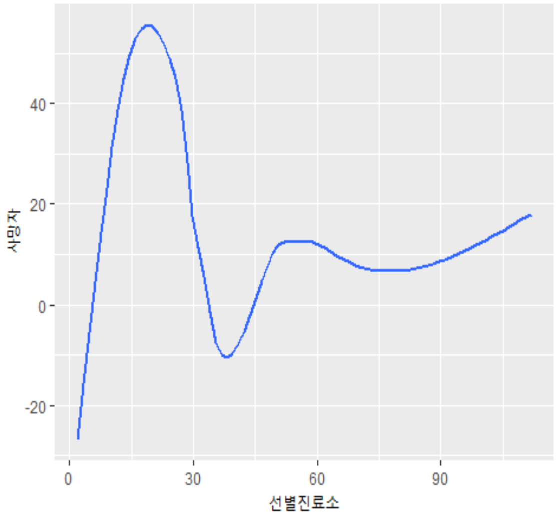
```
library(ggplot2)
library(dplyr)
ggplot(data=corona) +
  geom_point(mapping=aes(x=지역,y=감염자))
```

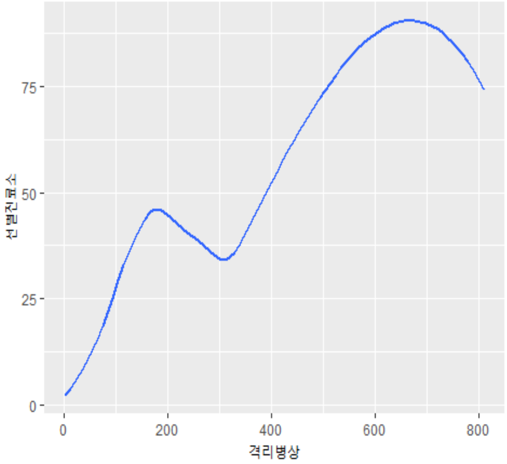


# Using ggplot2

✓ Data Visualization



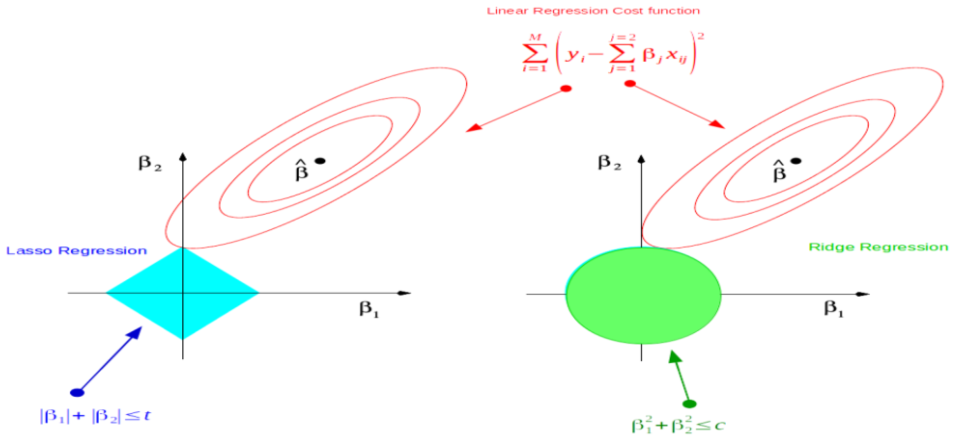



# Variable Selection – Lasso and Random Forest

✓ Lasso regression

$$\sum_{i=1}^{M}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p} w_j \times x_{ij}\right)^2 + \lambda \sum_{j=0}^{p} |w_j|$$

Cost function for Lasso regression



# Variable Selection

- ✓ Lasso regression

$$\sum_{i=1}^{M}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{M}\left(y_i - \sum_{j=0}^{p} w_j \times x_{ij}\right)^2 + \lambda \sum_{j=0}^{p} |w_j|$$

Cost function for Lasso regression

- ✓ Exist a problem of  selecting tunning parameter lambda
  Then use 5 cross validation to minimize error

# Variable Selection

✓ Lasso regression : set Y as 감염자

```
격리병상      선별진료소    단란주점.술.  유흥주점.클럽.
-8.33314     -31.19494    -0.69239         0.62251
신천지수.추정.               인구           인구밀도     1 인당.소득수준
-0.05781                  0.91458         0.66267         -0.91721
```

✓ Sorting

```
## [1] "선별진료소"      "격리병상"        "X1 인당.소득수준" "인구"
## [5] "단란주점.술."    "인구밀도"        "유흥주점.클럽."  "신천지수.추정."
```

# Variable Selection

- ✓ Random Forest feature selection – Bagging

  After Bootstrapping, make many decision trees and use mean

$$\hat{f}_{\text{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$$

→ 무작정 decision tree 생성 -> trees' high correlation

# Variable Selection

- ✓ Random Forest feature selection for solution, among predictors, choose m random samples ( sqrt(p) )
- ✓ Random Forest feature selection

- 변수를 random 변수로 바꾸었을 때
- 각 변수가 얼만큼 지니 계수를 줄이는데 기여했는지

→ out of bag error