

Lab3

CNN & RNN



Seoul National University



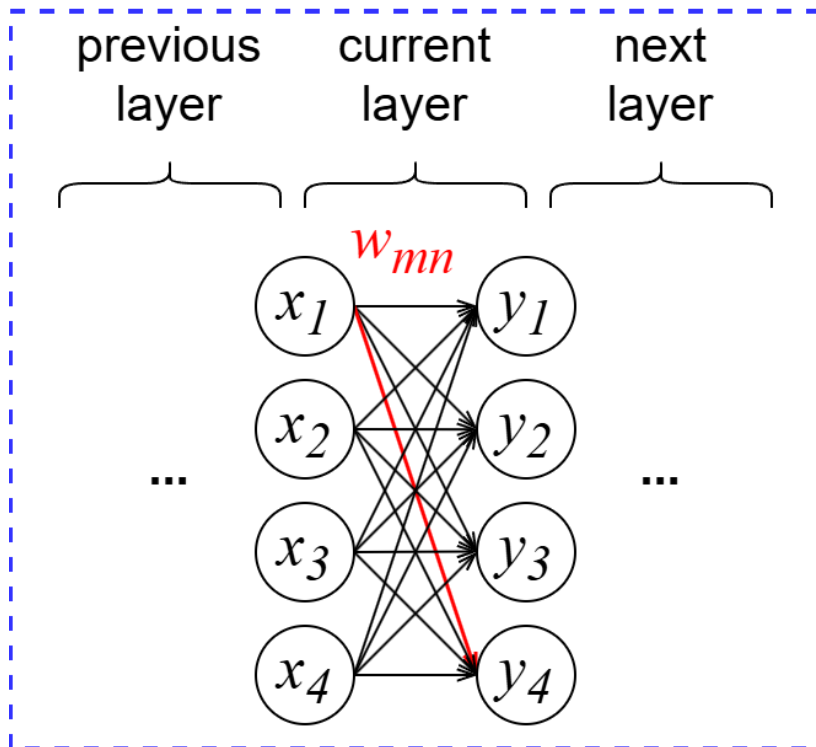
Human Interface Laboratory

Contents

- **Image Classification**
 - Image Classification Using Fully Connected (FC) Layers
 - Softmax
 - Cross Entropy Loss
 - Mini-batch Training
- **Convolutional Neural Network (CNN)**
 - Motivation
 - Convolutional Layer
 - Pooling
- **Recurrent Neural Network (RNN)**
 - Sequence Data
 - Plain RNN
 - Long Short-Term Memory (LSTM)
 - Gated Recurrent Unit (GRU)

Fully Connected Layer 복습

Deep Neural Network




- Fully Connected Layer

$$y_m = \sum_n w_{mn} \cdot x_n + b_m$$

$$Y = W \cdot X + B$$

- 학습할 parameter $\theta = [W, B]$

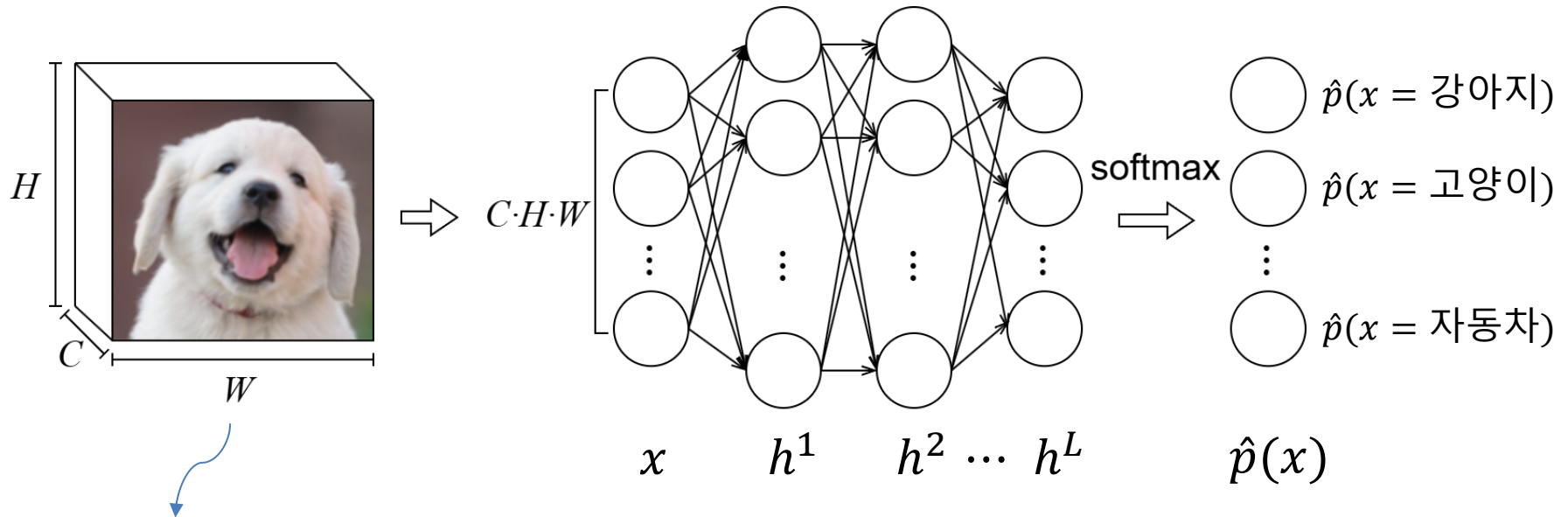
Image Classification

$$N.N. \text{의 출력 } \hat{y} = f_{\theta}(\text{) = \begin{bmatrix} \hat{p}(x = \text{강아지}) \\ \hat{p}(x = \text{고양이}) \\ \vdots \\ \hat{p}(x = \text{자동차}) \end{bmatrix}$$

$$\text{원하는 출력 } y = \begin{bmatrix} p(x = \text{강아지}) \\ p(x = \text{고양이}) \\ \vdots \\ p(x = \text{자동차}) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

- $\hat{y} \approx y$ 되도록 (Loss 작아지도록) Neural Network의 parameter θ 학습

Image Classification Using FC Layers



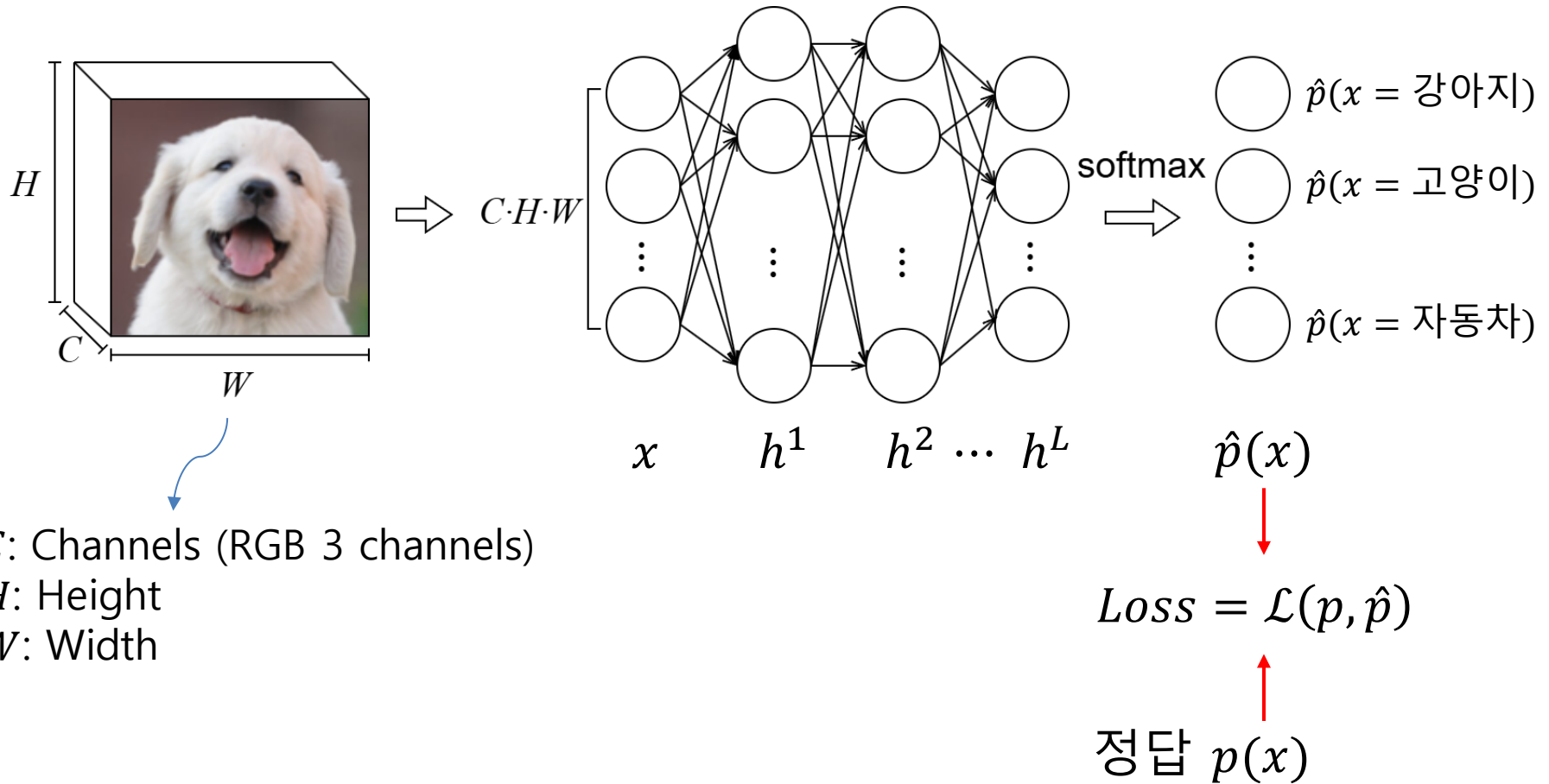
C : Channels (RGB 3 channels)
 H : Height
 W : Width

Softmax

- 모델의 출력이 $\hat{y}_n = \hat{p}(x = n)$ 확률을 모델링 하기 위한 조건
 - $\sum_n \hat{y}_n = 1$
 - $\hat{y}_n \geq 0$
- 두 조건 모두 만족하는 activation function | Softmax

N.N. Output		Softmax		Probability
$h = \begin{bmatrix} 1.3 \\ 2.1 \\ -1.2 \\ 0.7 \end{bmatrix}$	\Rightarrow	<div style="border: 1px solid black; padding: 10px; display: inline-block;">$\hat{y}_n = \frac{e^{h_n}}{\sum_m e^{h_m}}$</div>	\Rightarrow	$\hat{y} = \begin{bmatrix} 0.26 \\ 0.58 \\ 0.02 \\ 0.14 \end{bmatrix}$

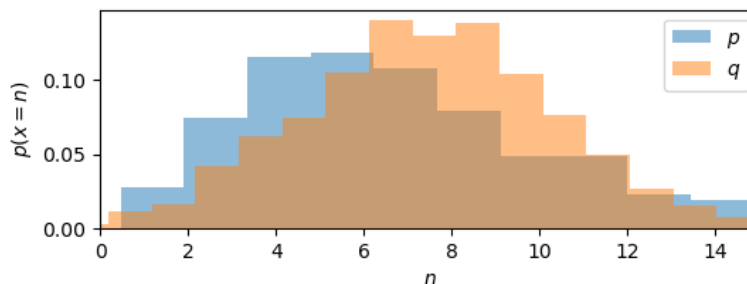
Image Classification Using FC Layers



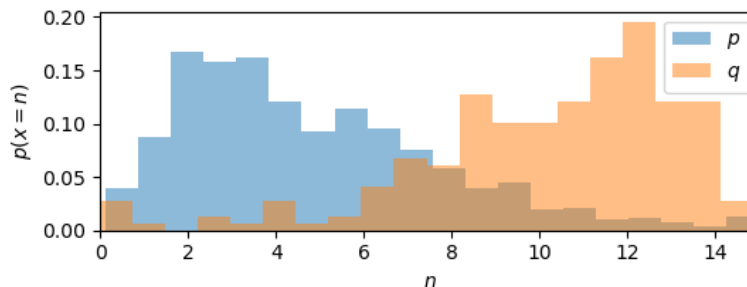
Cross Entropy Loss

$$\mathcal{L}(p, q) = -\frac{1}{N} \sum_{n=1}^N p(x = n) \cdot \log q(x = n)$$

- 두 확률분포 간의 차이를 측정 (정보이론)
- 작을수록 두 확률분포가 비슷함을 의미
- 클수록 두 확률분포가 많이 다름을 의미



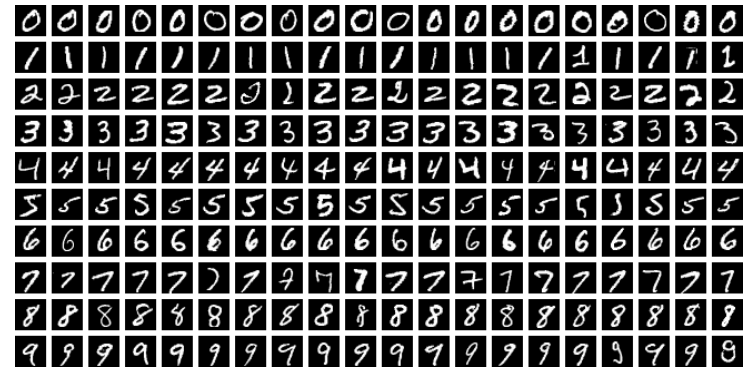
$$\Rightarrow \mathcal{L}(p, q) = 2.61$$



$$\Rightarrow \mathcal{L}(p, q) = 3.82$$

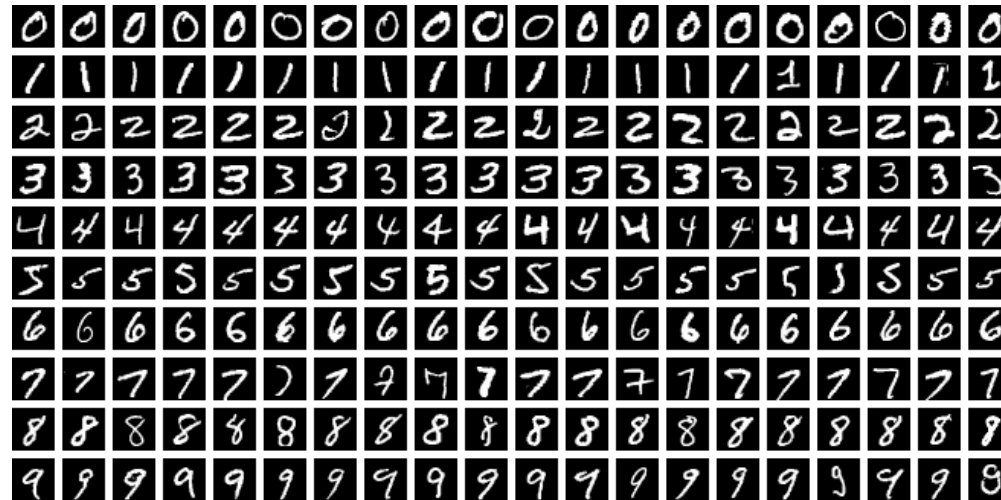
Mini-batch Training

- 오른쪽과 같은 손글씨 인식 Task에서
- 전체 학습 데이터 개수: 20000개 라면
- 전체 데이터(20000개)를 한 번 학습하는 것을 one epoch.



- 이미지 하나하나마다 \mathcal{L} 를 구해서 θ 를 학습하는 것이 아니라 (20000 iteration / 1 epoch)
- 200개의 이미지에 대해 \mathcal{L} 를 구해서 평균값으로 θ 를 학습 (100 iteration / 1 epoch)

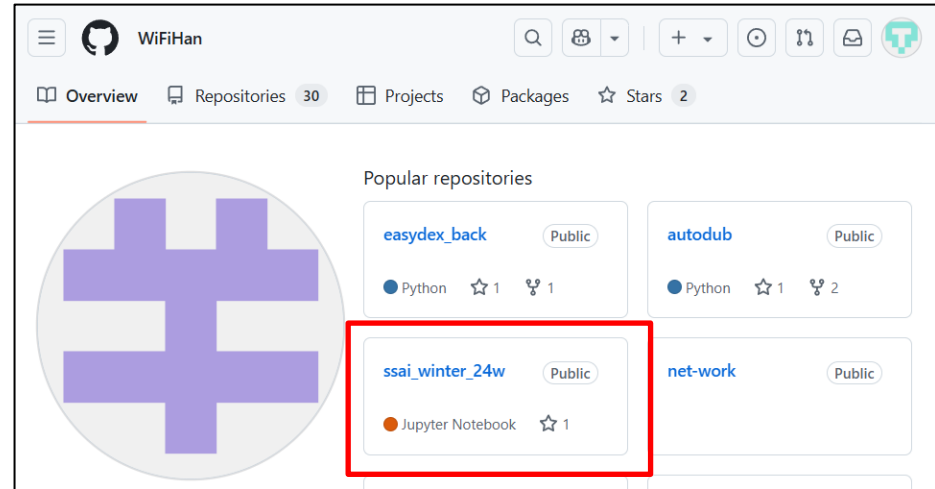
실습 (Image Classification Using FC Layers)



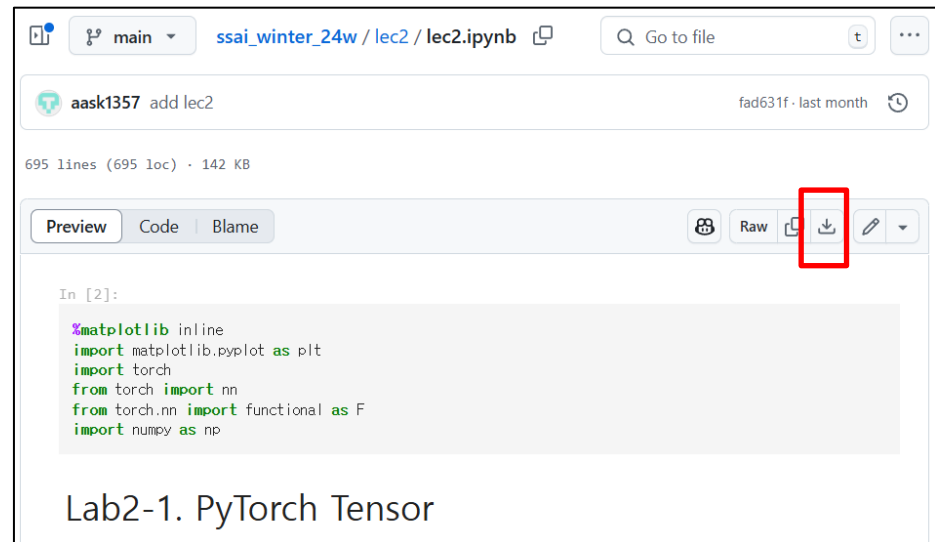
- Lab3-1. Image Classification Using FC Layers
- Image에서 Fully Connected Layer가 최선일까?

실습 (Image Classification Using FC Layers)

1. github.com/wifihan 접속
2. [ssai_winter_24w](#) 클릭

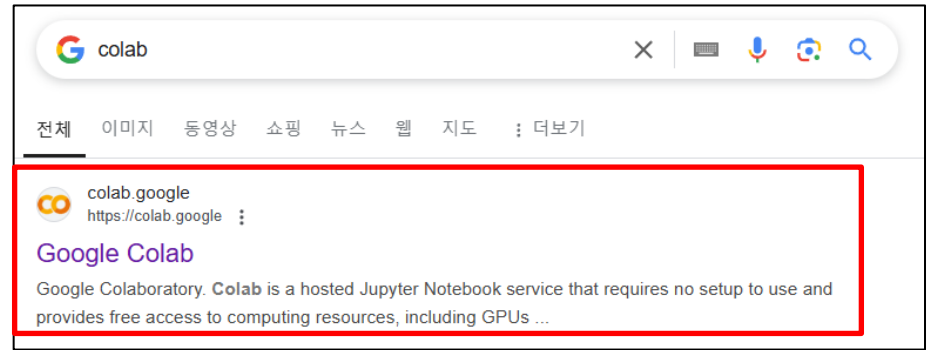


3. [lec3/lec3.ipynb](#) 다운로드



실습 (Image Classification Using FC Layers)

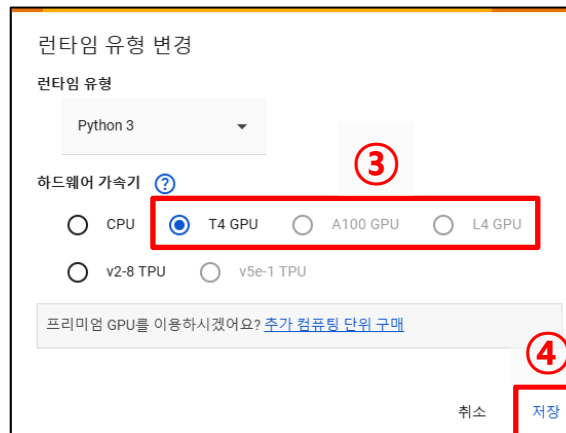
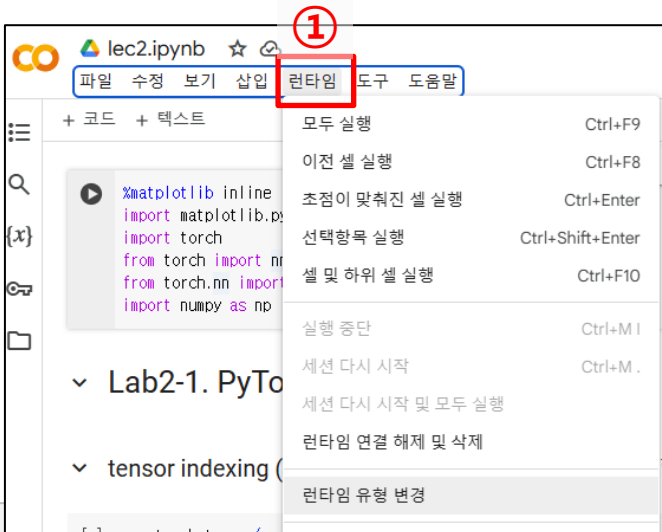
4. 구글에 colab 검색 후 접속



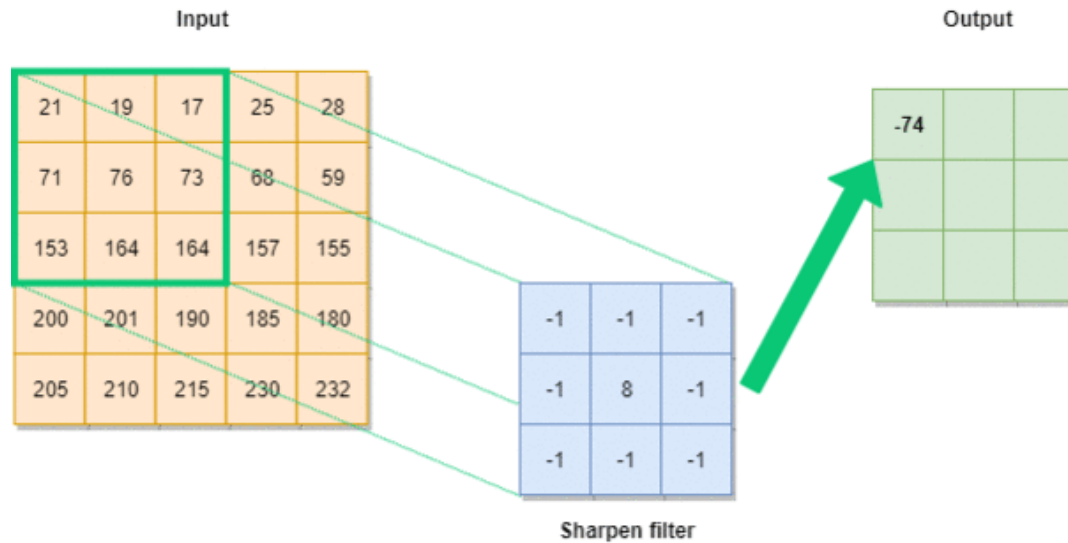
4. Open Colab -> 파일 -> 업로드 (방금 다운받았던 lec3.ipynb)

5. 런타임 유형 GPU로 변경

6. 연결 확인



Convolutional Layer

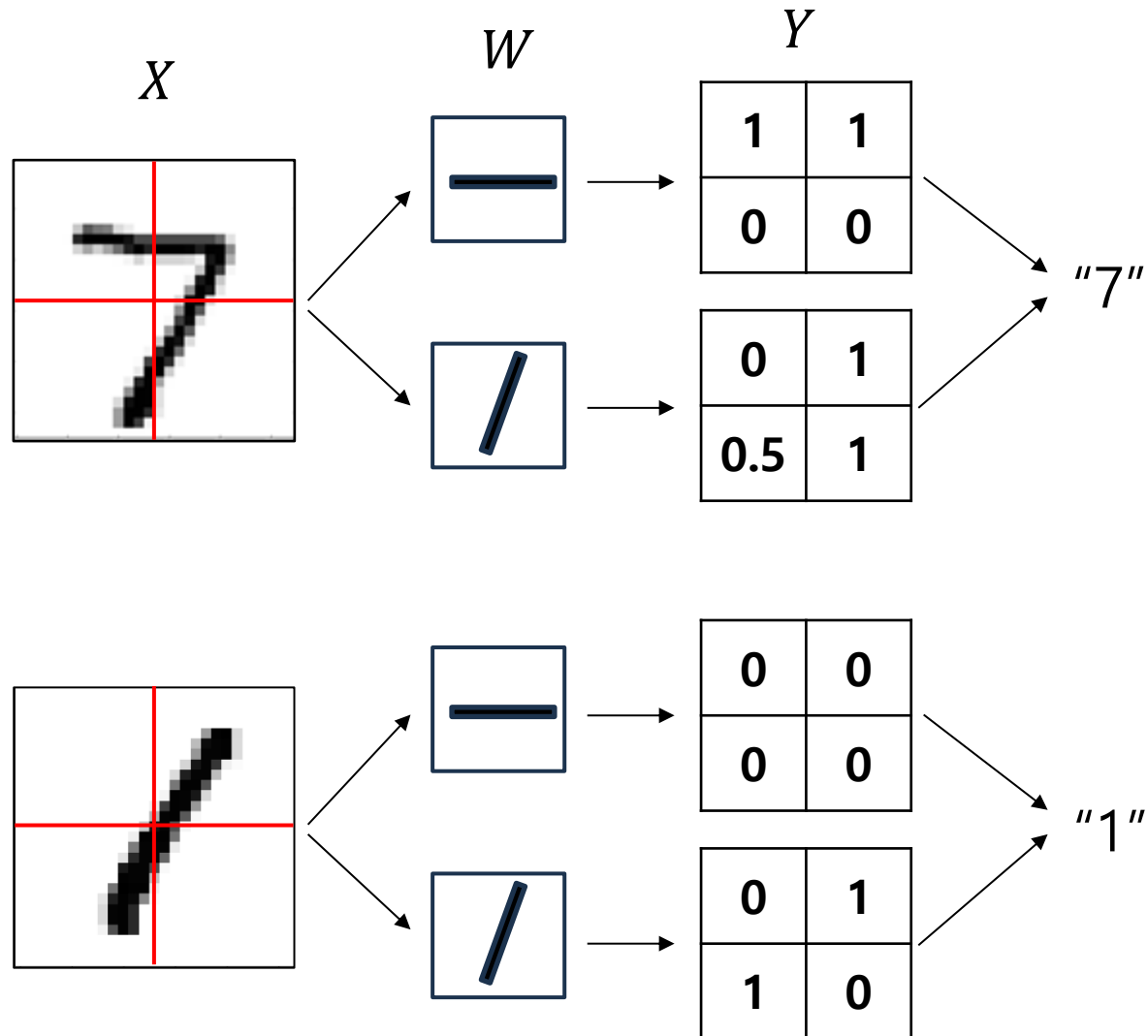


$$y[m, n] = \sum_{s, t} w[s, t] \cdot x[m + s, n + t] + b$$

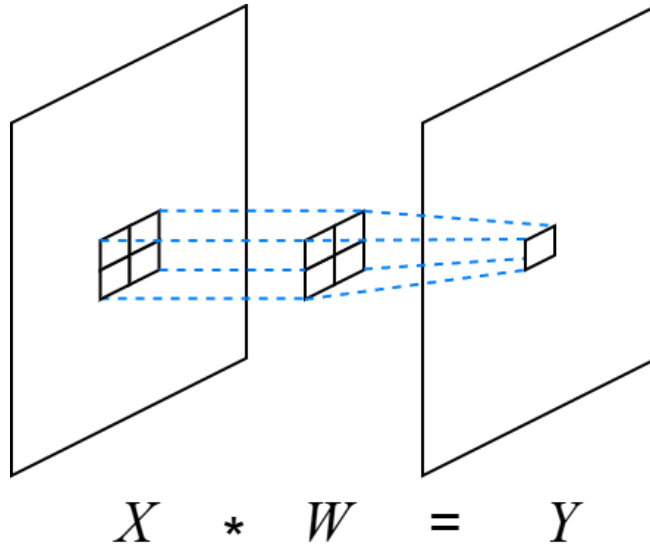
$$Y = W * X + b$$

➤ 학습할 parameter $\theta = [W, b]$

Motivation



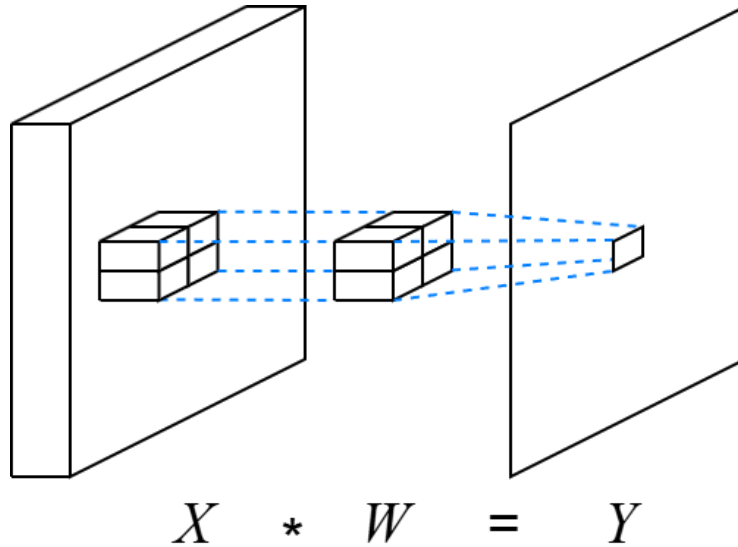
Convolutional Layer



X 의 shape: $H_i \times W_i$
 W 의 shape: $K_h \times K_w$
 B 의 shape: 1
 Y 의 shape: $H_o \times W_o$

$$y[m, n] = \sum_{s, t} w[s, t] \cdot x[m + s, n + t] + b$$

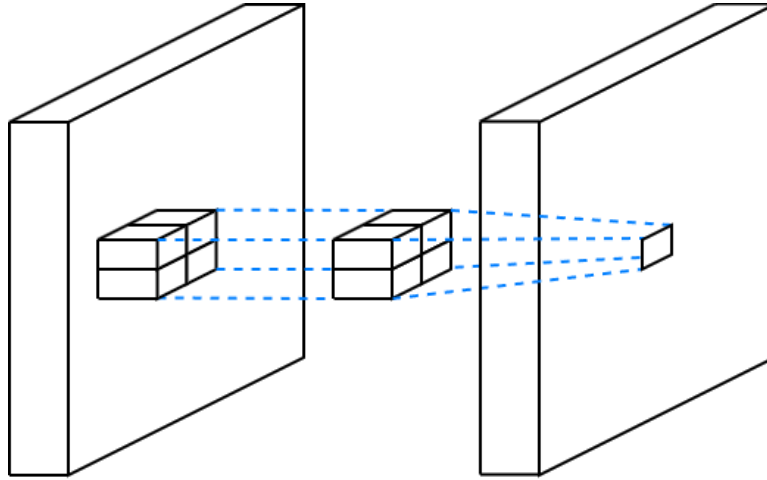
Convolutional Layer ($C_{in} > 1$)



X 의 shape: $C_i \times H_i \times W_i$
 W 의 shape: $C_i \times K_h \times K_w$
 B 의 shape: 1
 Y 의 shape: $H_o \times W_o$

$$y[m, n] = \sum_{c_i, s, t} w[c_i, s, t] \cdot x[c_i, m + s, n + t] + b$$

Convolutional Layer ($C_{out} > 1$)

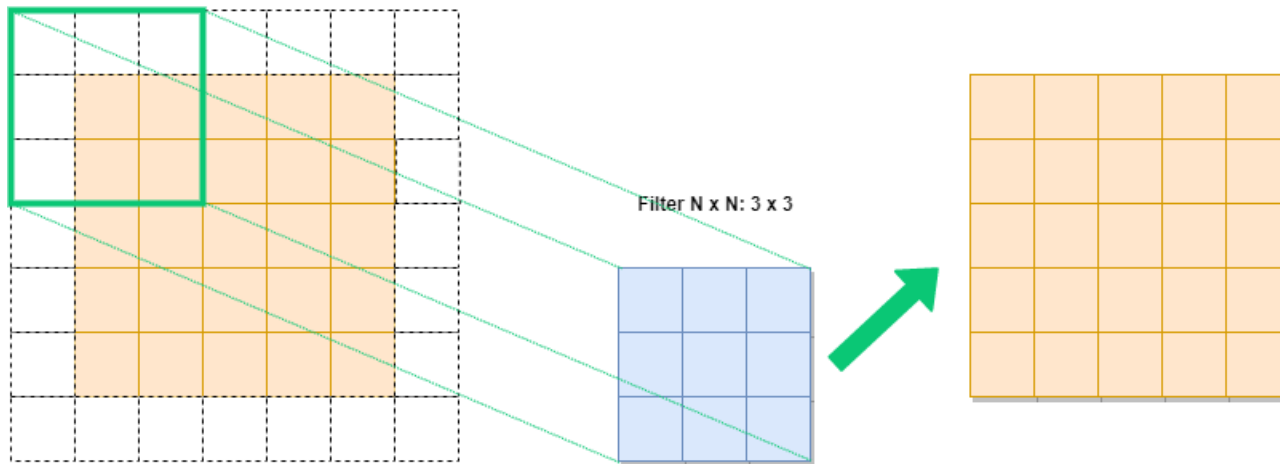
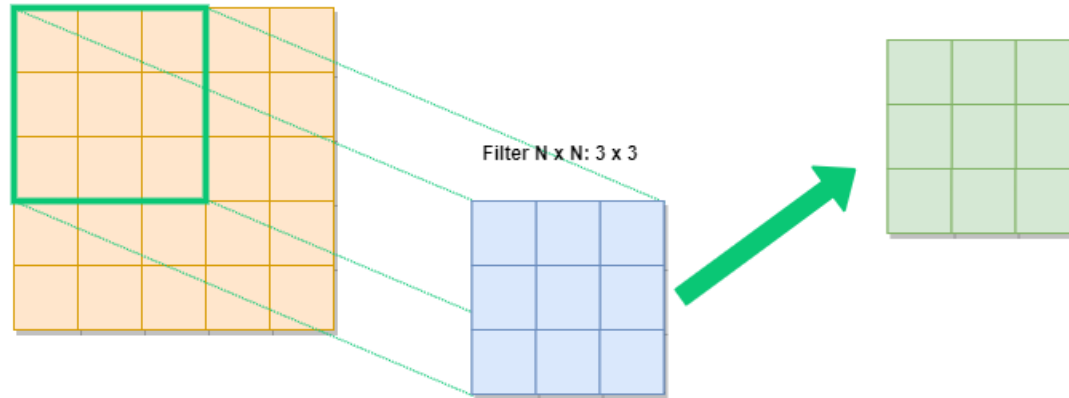


X 의 shape: $C_i \times H_i \times W_i$
 W 의 shape: $C_o \times C_i \times K_h \times K_w$
 B 의 shape: C_o
 Y 의 shape: $C_o \times H_o \times W_o$

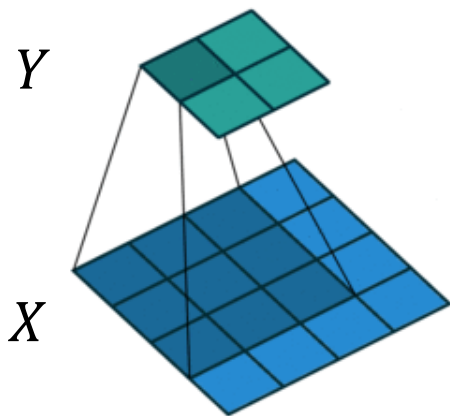
$$X * W[c_o] = Y[c_o]$$

$$y[c_o, m, n] = \sum_{c_i, s, t} w[c_o, c_i, s, t] \cdot x[c_i, m + s, n + t] + b[c_o]$$

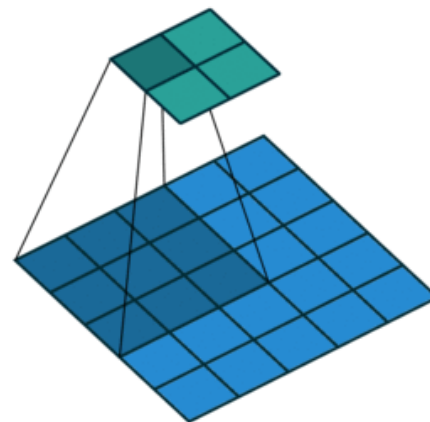
Padding



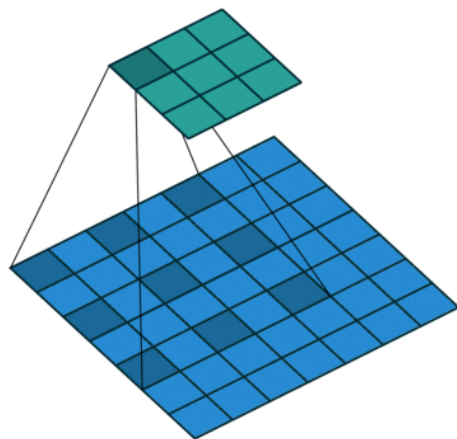
Stride, Dilation, Transposed



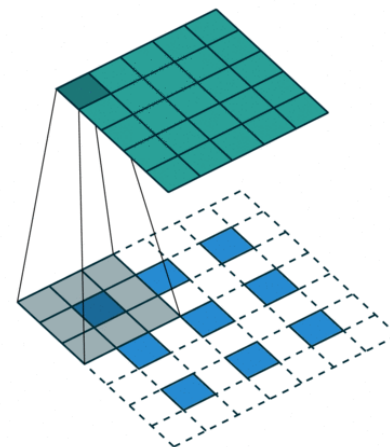
Default Convolution



Convolution (Stride=2)



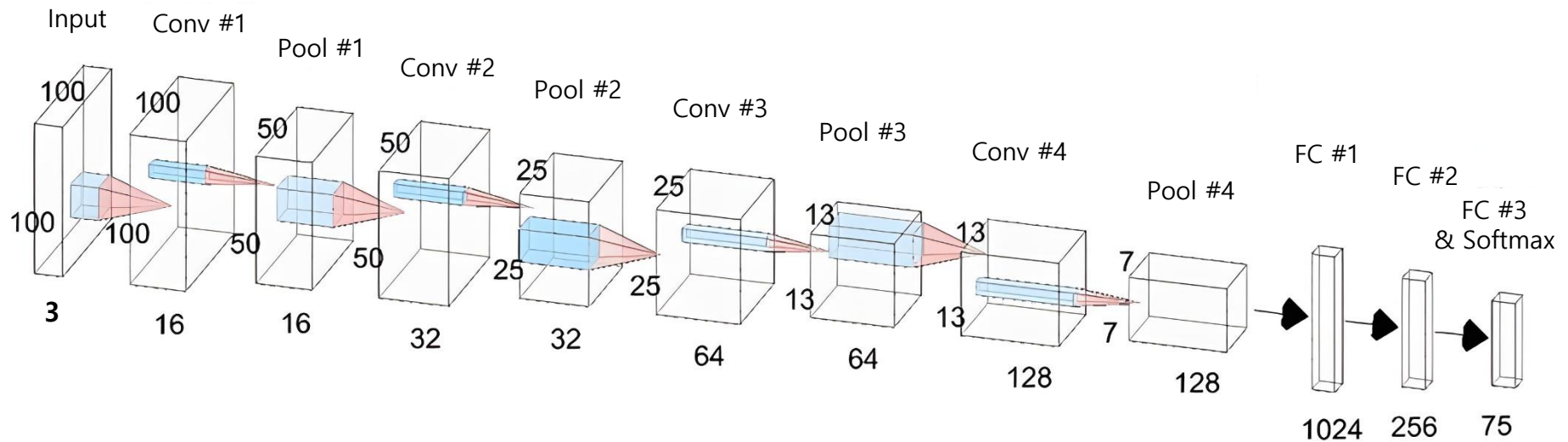
Convolution (Dilation=2)



Transposed Convolution (Stride=2)

Convolutional Neural Network (CNN)

- Convolutional Layers로 구성된 Neural Network



Pooling

Max Pooling

29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2
pool size

100	184
12	45

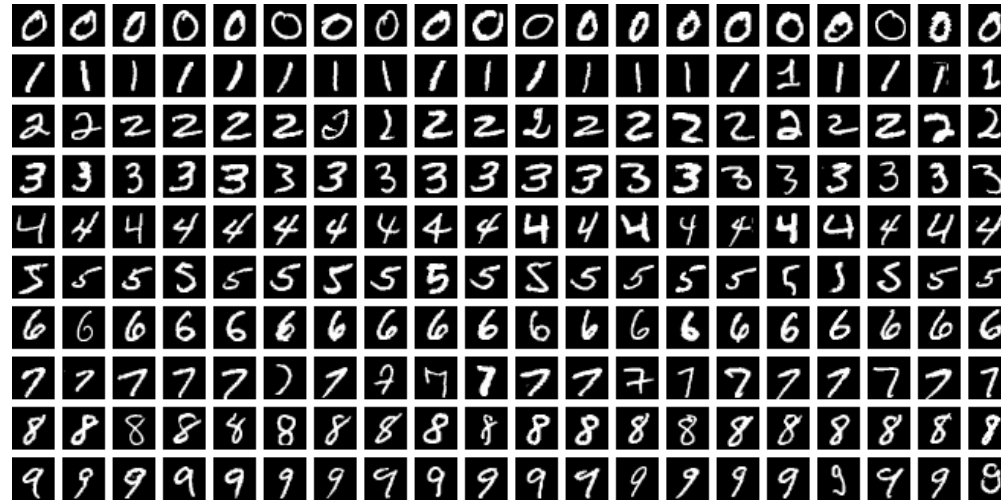
Average Pooling

31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2
pool size

36	80
12	15

실습 (Image Classification Using CNN)

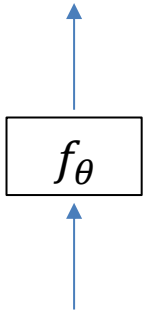


- Lab3-2. Image Classification Using CNN

Sequence Data

Text Classification

$$p(x = \text{긍정}) = 0.9$$
$$p(x = \text{부정}) = 0.1$$



"꿀잼. 넘 재밌다."

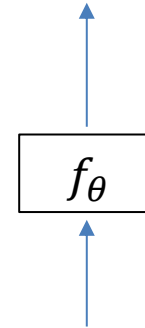
Text-to-Speech



"안녕하세요?"

ChatGPT

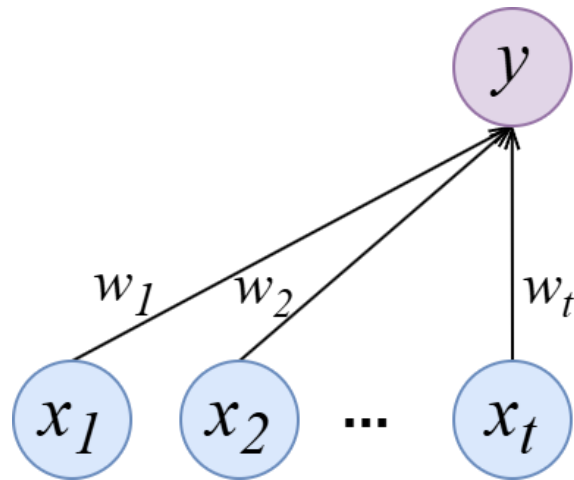
"Answer"



"Question"

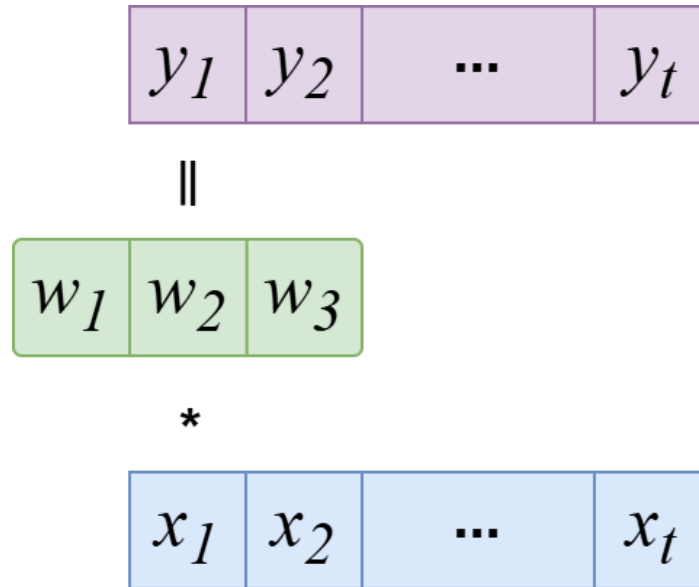
- $y = f_\theta(x_1, x_2, \dots)$
- $y_1, y_2, \dots = f_\theta(x)$
- $y_1, y_2, \dots = f_\theta(x_1, x_2, \dots)$

FC for Sequence Data?



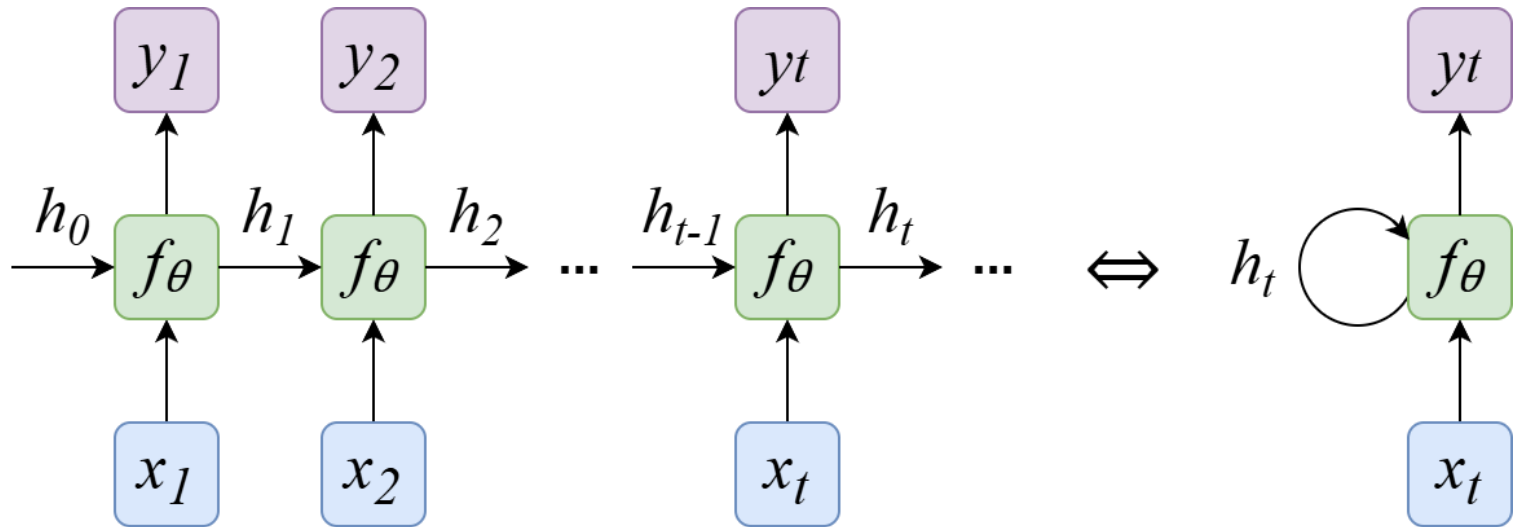
➤ 입력 길이가 달라지면?

CNN for Sequence Data?



➤ y_t 생성할 때 x_1 은 못보는데?

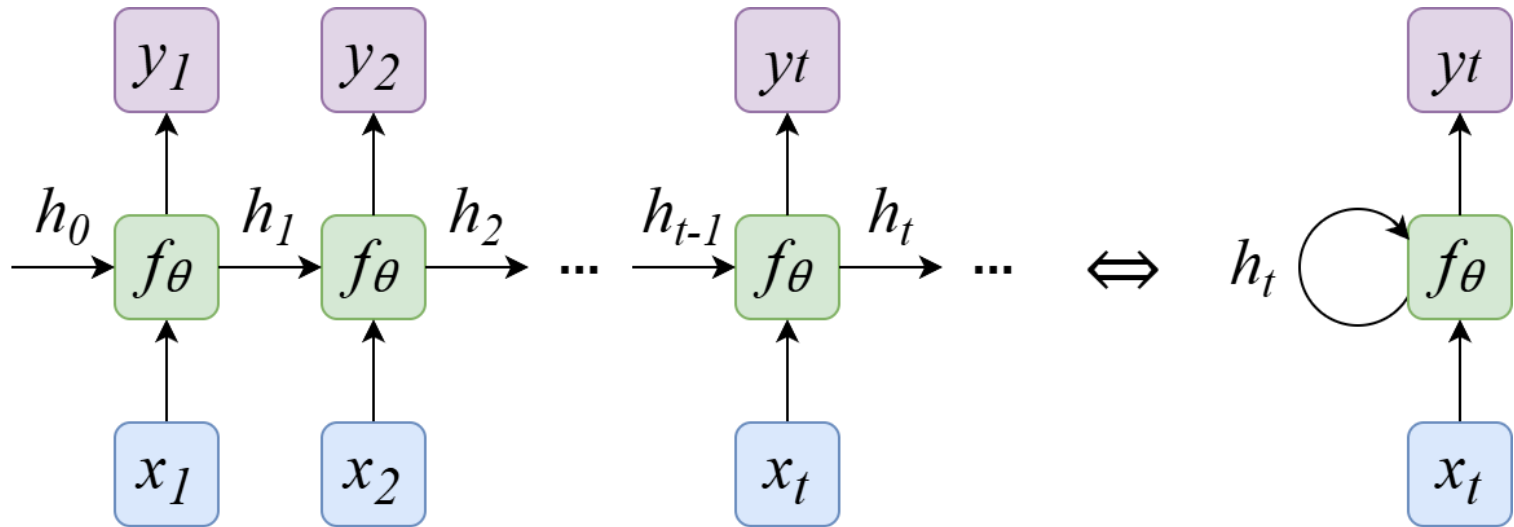
Recurrent Neural Network (RNN)



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

- 입력 길이가 일정하지 않아도 OK
- y_t 생성할 때 $x_1 \sim x_t$ 정보 모두 담고 있음

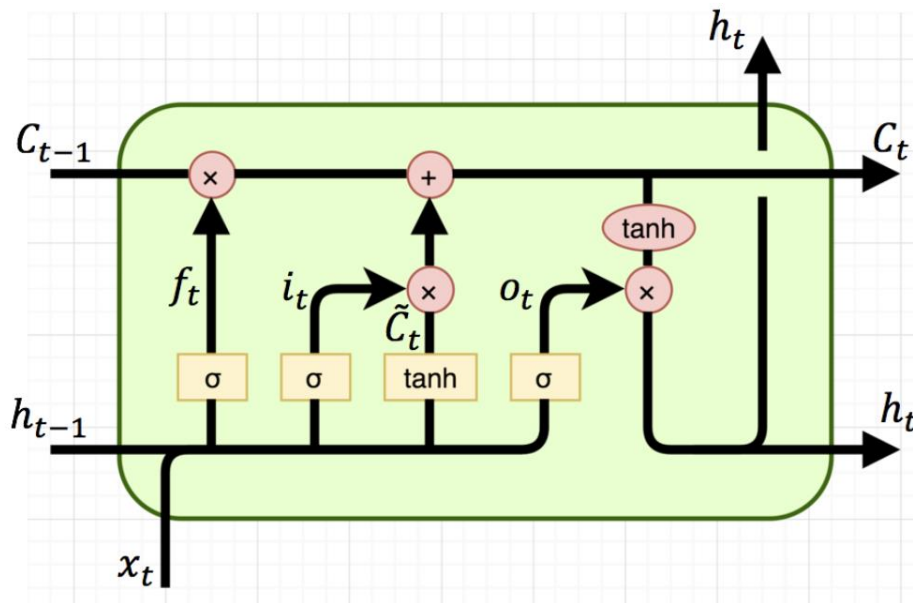
Plain RNN



$$h_t = \sigma(w_x x_t + w_h h_{t-1} + b)$$

- 학습할 parameter $\theta = [w_x, w_h, b]$
- 성능이 별로 좋지 못함

LSTM



학습할 parameter θ

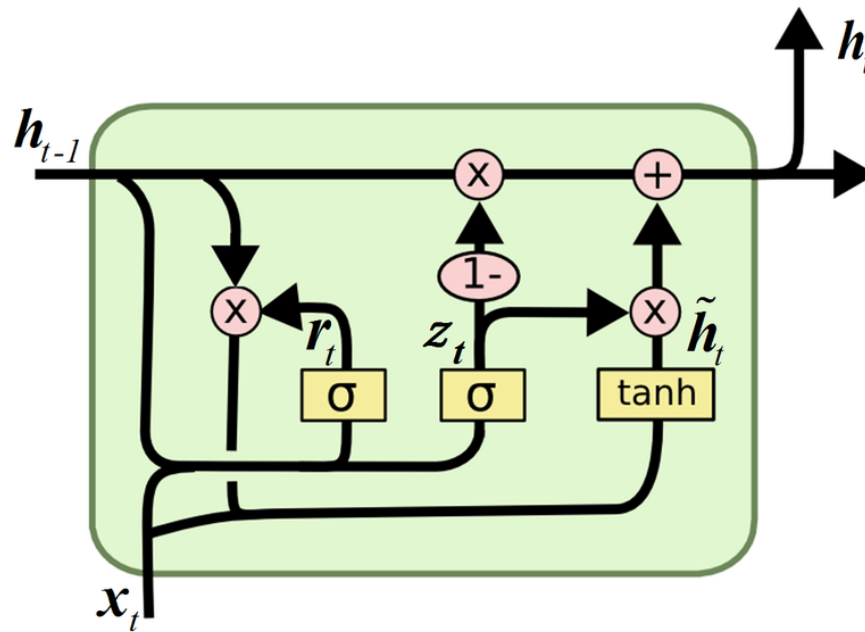
$$\begin{aligned}
 f_t &= \sigma(w_{xf}x_t + w_{hf}h_{t-1} + b_f) \\
 i_t &= \sigma(w_{xi}x_t + w_{hi}h_{t-1} + b_i) \\
 \tilde{c}_t &= \tanh(w_{xc}x_t + w_{hc}h_{t-1} + b_c) \\
 o_t &= \sigma(w_{xo}x_t + w_{ho}h_{t-1} + b_o) \\
 c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \\
 h_t &= o_t \odot \tanh(c_t)
 \end{aligned}$$

σ : sigmoid

\tanh : hyperbolic tangent

\odot : element-wise multiplication

GRU



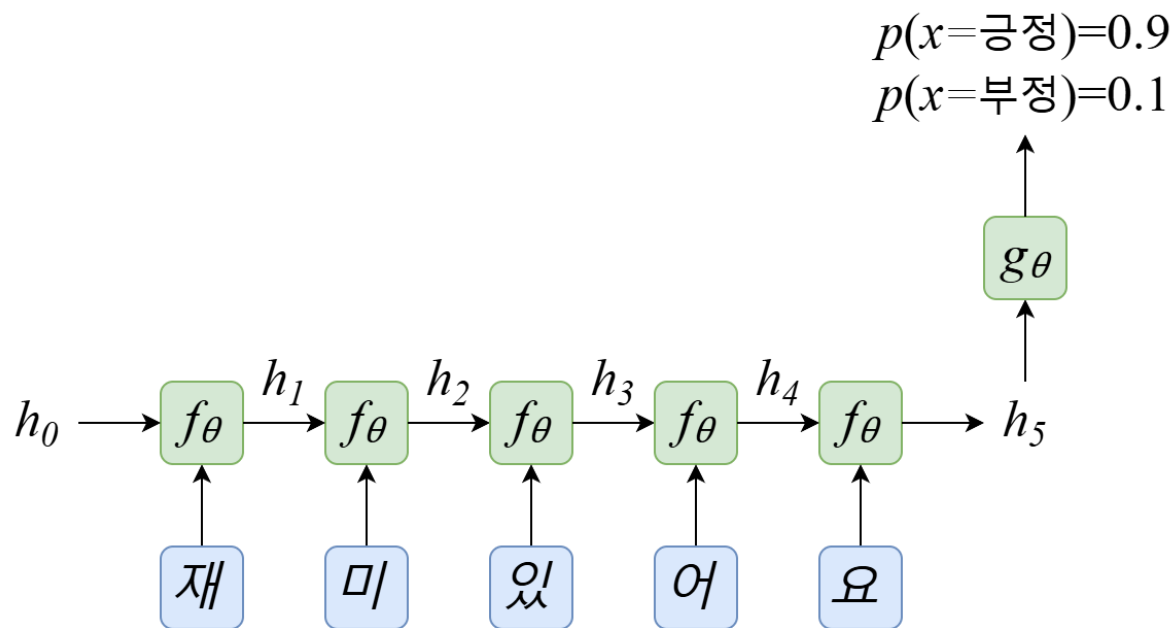
$$r_t = \sigma(\mathbf{w}_{xr}x_t + \mathbf{w}_{hr}h_{t-1} + b_r)$$

$$z_t = \sigma(w_{xz}x_t + w_{hz}h_{t-1} + b_z)$$

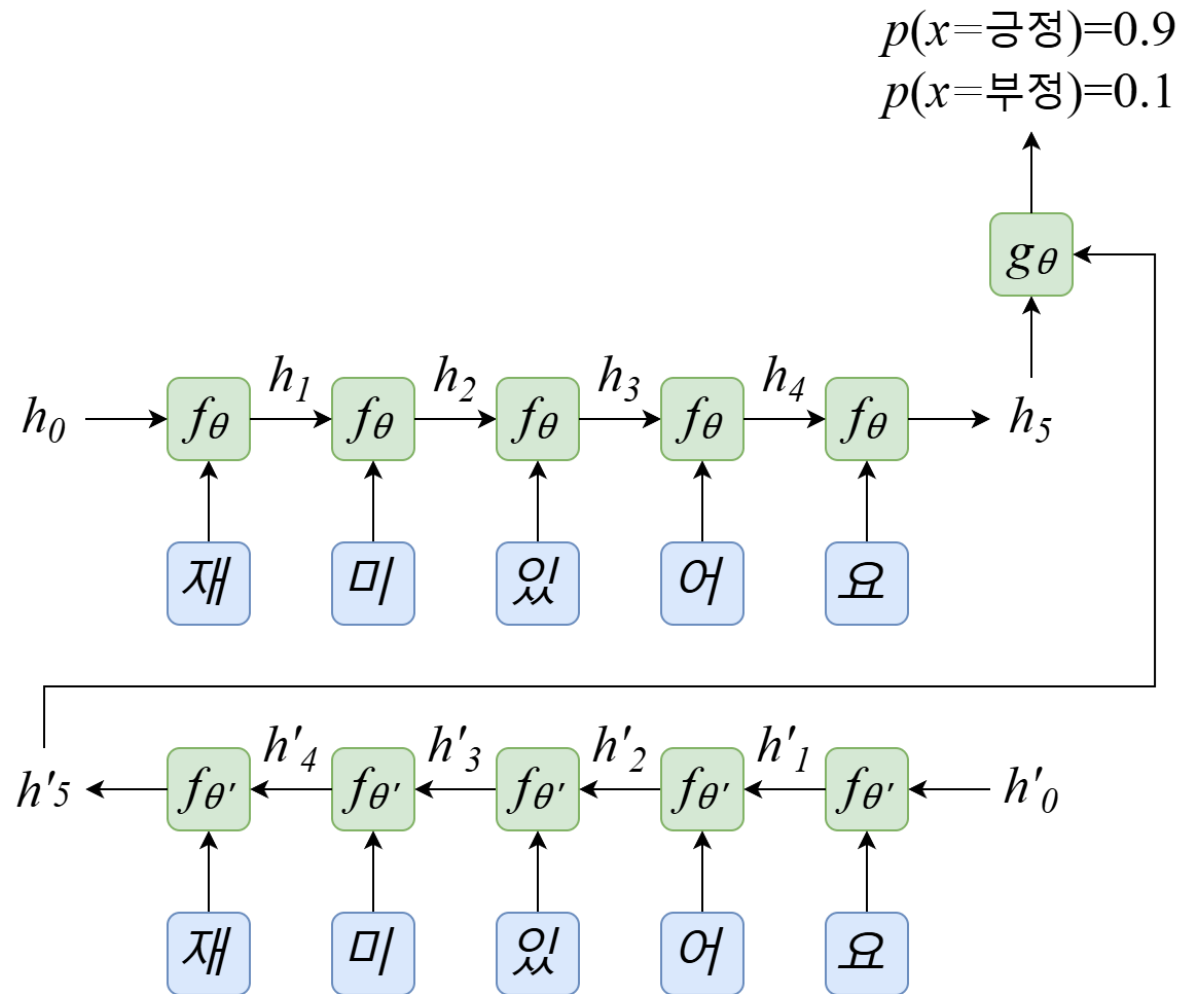
$$\tilde{h}_t = \tanh\left(w_{xh}x_t + b_{h_1} + r_t \odot (w_{hh}h_{t-1} + b_{h_2})\right)$$

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

Text Classification Using RNN



Bi-directional RNN



실습 Speech Command Classification)

- Lab3-3. Speech Command Classification Using RNN