

Section 3. 손실 함수 (Loss Function)

목차

- 섹션 0. 강의 소개
- 섹션 1. PyTorch 환경 설정
- 섹션 2. 딥러닝이란?
- **섹션 3. 손실 함수 (Loss Function)**
- 섹션 4. 손실 함수에 대한 심화 이론 (Advanced Topics on Loss Function)
- 섹션 5. 경사 하강 (Gradient Descent)
- 섹션 6. 경사 하강에 대한 심화 이론 (Advanced Topics on Gradient Descent)

“Section 2. 딥러닝이란?”에서 배웠던 내용

Recap

Section 2. 딥러닝이란?

- Neural Network이 학습되는 과정 = weight값이 최적화되는 과정

Gradient Descent (경사 하강)을 통한 **Loss function (손실 함수)** 값을 최소화하도록

weight 값을 최적화하여 점진적으로 모델의 예측 정확도를 높인다.

- 그렇다면 Loss Function을 뭘까?
 - Loss function은 무엇이고 왜 최소화되어야 할까?

Objective

학습 목표

- Loss function의 정의
- Regression task와 Classification task의 정의
- Regression task의 Loss
- Classification task의 Loss

3-1. 손실 함수의 정의

Loss Function

Loss Function = 정답값과 예측값의 오차

Loss Function L (손실 함수) 의 정의

Neural Network 모델이 예측한 값 \hat{Y} 과 원래 정답 Y 간의 차이 (오차)의 지표

Loss Function

Loss Function = 정답값과 예측값의 오차

Loss Function L (손실 함수) 의 정의

Neural Network 모델이 예측한 값 \hat{Y} 과 원래 정답 Y 간의 차이 (오차)의 지표

대표적으로 Mean Square Error Loss 가 있다:

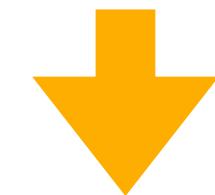
$$L(Y, \hat{Y}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

Loss Function

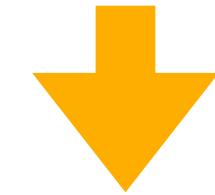
Loss Function = 정답값과 예측값의 오차

(다양한 손실 함수가 있지만) 모두 공통적으로:

정답값 Y 과 예측값 \hat{Y} 간의 **오차가 크다**

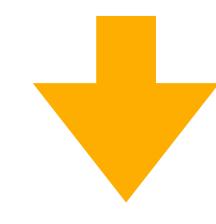


부정확한 예측



손실 함수의 값 $L(Y, \hat{Y})$ 도 크다.

정답값 Y 과 예측값 \hat{Y} 간의 **오차가 작다**



정확한 예측



손실 함수의 값 $L(Y, \hat{Y})$ 도 작다.

즉, 손실 함수의 값이 **최소화**하도록 모델의 **weight**을 **최적화**하면 모델의 **정확도가 높아진다!**

Loss Function

Loss Function = 정답값과 예측값의 오차

Loss function은 풀고자 하는 Task (문제)에 따라 다르다!

3-2. Task의 종류

Loss function

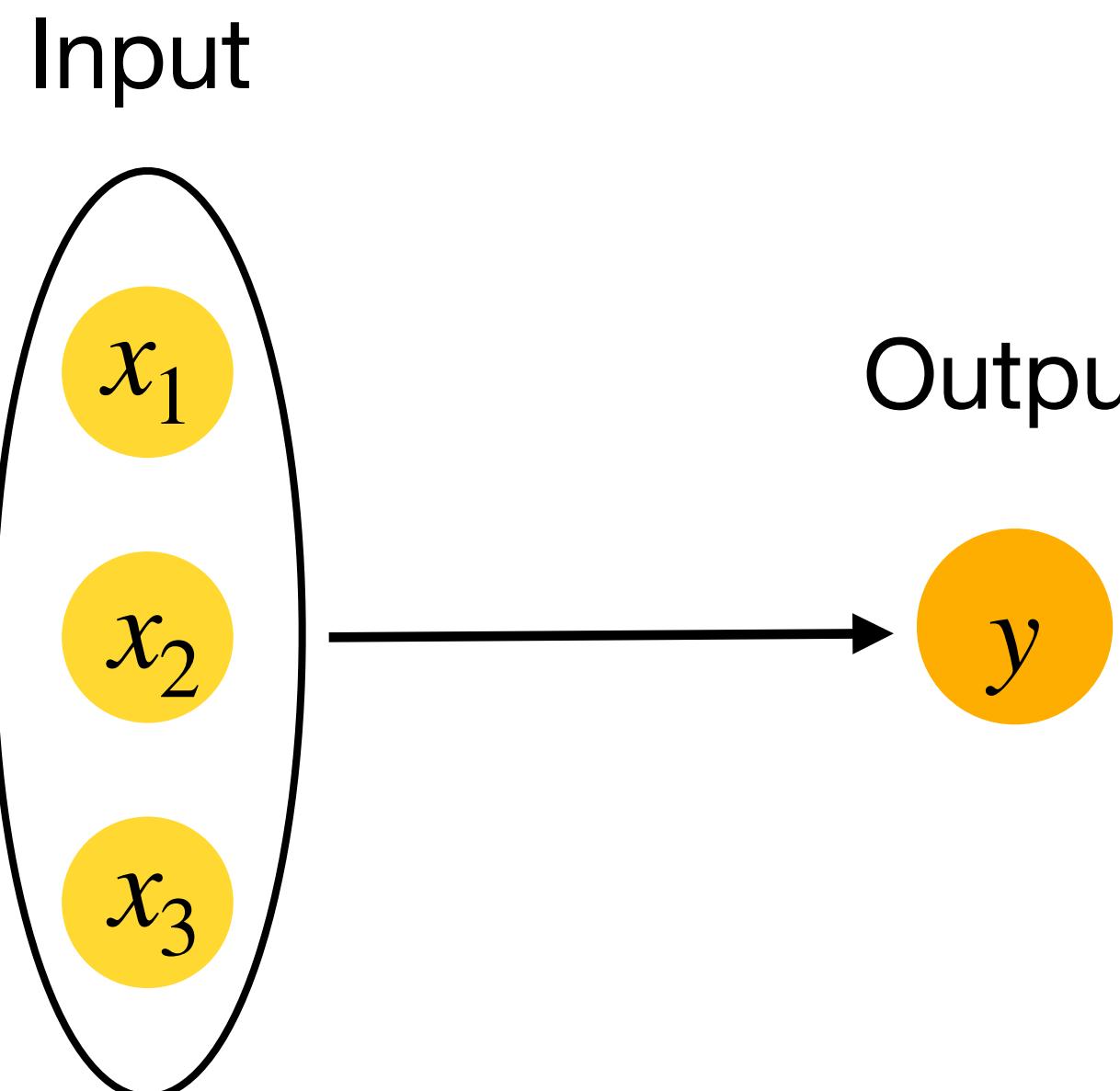
대표적인 Task의 종류

- Regression Task (회귀)
- Classification Task (분류)

Regression Task vs. Classification Task

- “**Regression (회귀)**” = 어떤 연속적인 값을 예측하는 task
 - 예시: 주가의 가격을 예측하는 task 등등.
- “**Classification (분류)**” = “discrete”, “categorical” 클래스 라벨을 예측하는 task
 - 예시: 책의 장르를 예측하는 task 등등.

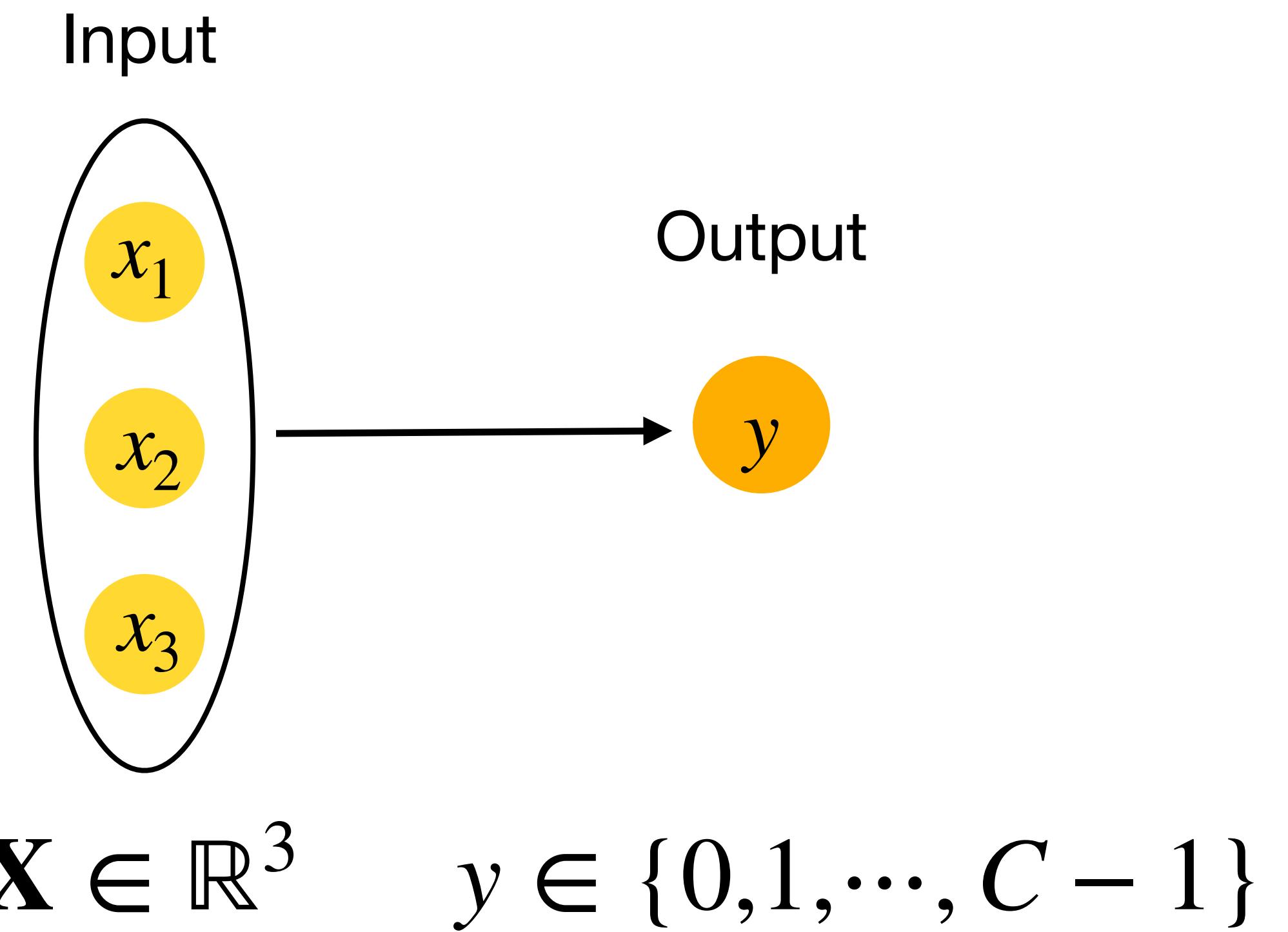
Regression Task vs. Classification Task



$$\mathbf{X} \in \mathbb{R}^3$$

$$y \in [a, b]$$

Regression Task



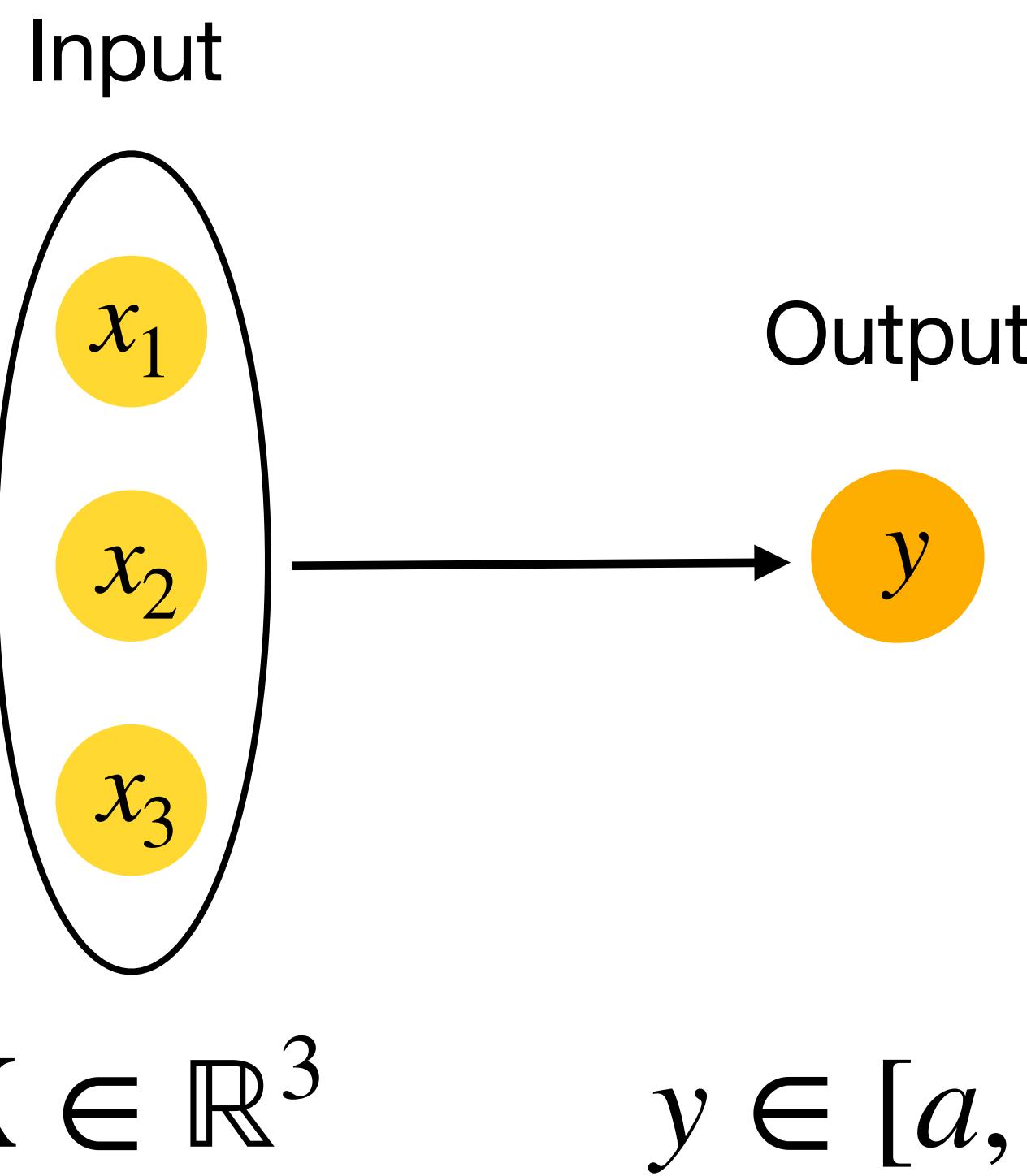
$$\mathbf{X} \in \mathbb{R}^3$$

$$y \in \{0, 1, \dots, C - 1\}$$

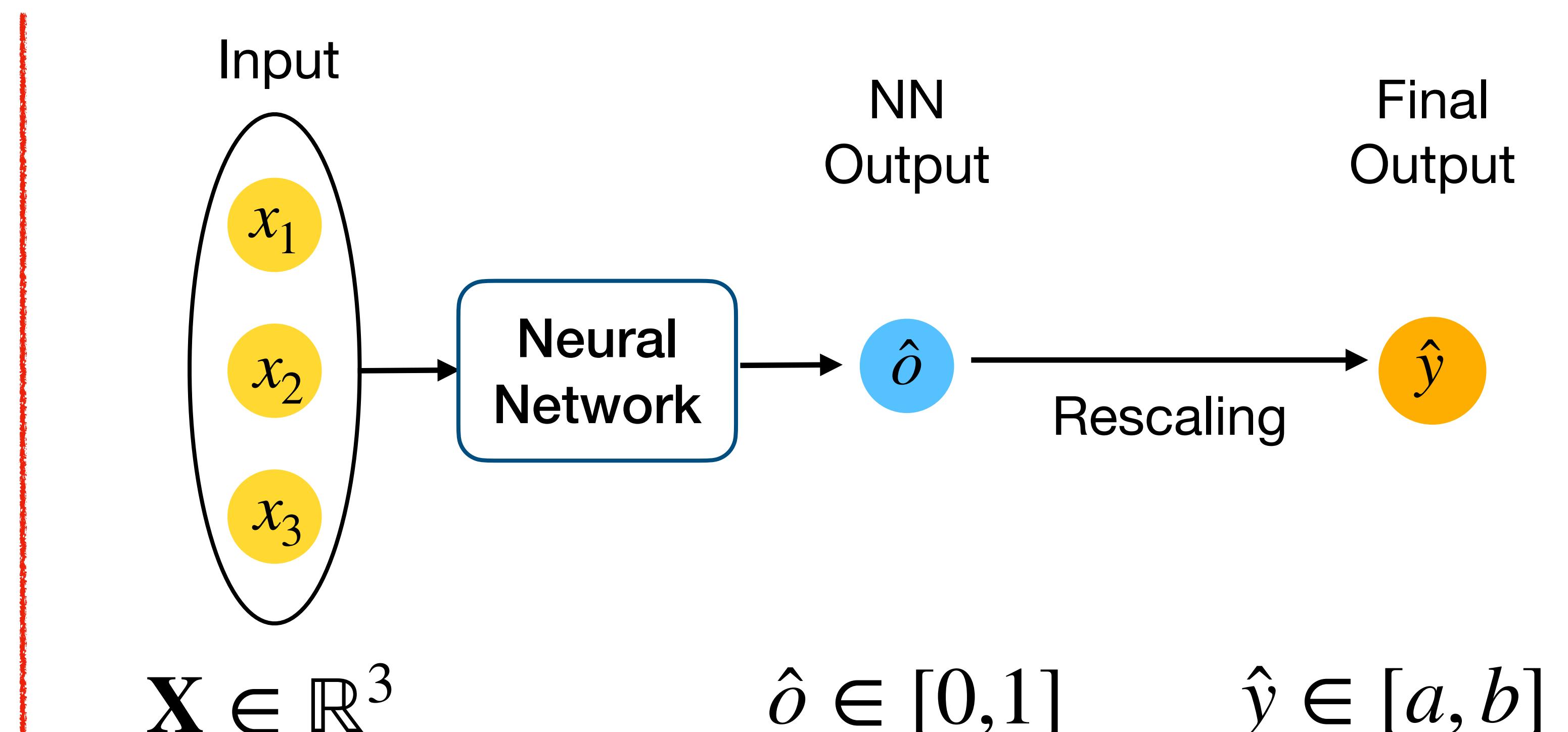
Classification Task

ACADENTIAL

Regression Task에 대한 접근 방법



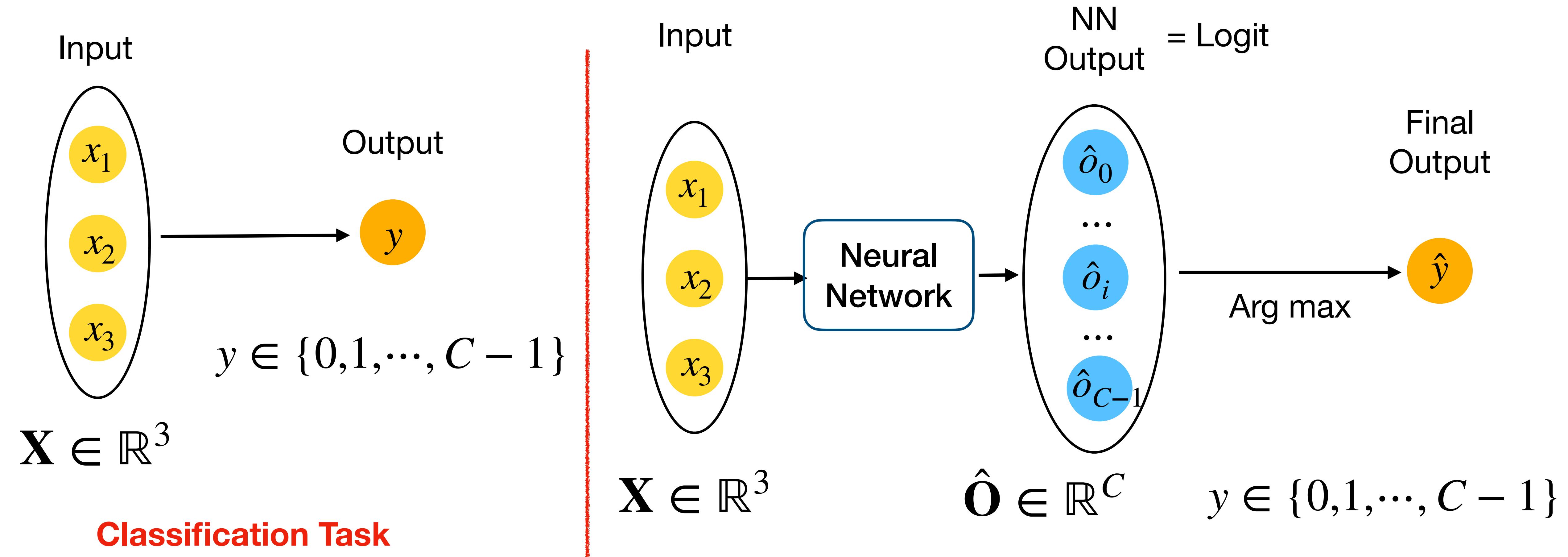
Regression Task



Rescaling:

$$\hat{y} = (b - a) \cdot \hat{o} + a$$

Classification Task에 대한 접근 방법



Loss function

살펴본 내용

- Regression Task (회귀)
- Classification Task (분류)

3-3. Regression Task의 Loss Function: L1, L2, Huber, Log Cosh

Regression task의 Loss 종류

Loss for Regression Task:

- L2 Loss (MSE Loss)
- L1 Loss (MAE Loss)
- Huber Loss
- Log-Cosh Loss

Regression task의 Loss 종류

Loss for Regression Task:

- L2 Loss (MSE Loss) → Mean Squared Error
- L1 Loss (MAE Loss) → Mean Absolute Error
- Huber Loss
- Log-Cosh Loss

Regression task의 Loss 종류

Loss for Regression Task:

L2 Loss (MSE Loss)

\hat{Y}_i predicted value

Y_i ground truth value

Regression task의 Loss 종류

Loss for Regression Task:

$$\text{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

\hat{Y}_i predicted value

Y_i ground truth value

Regression task의 Loss 종류

Loss for Regression Task:

$$\text{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

\hat{Y}_i predicted value

L1 Loss (MAE Loss)

Y_i ground truth value

Regression task의 Loss 종류

Loss for Regression Task:

$$\text{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

\hat{Y}_i predicted value

$$\text{L1 Loss (MAE Loss)} = | Y_i - \hat{Y}_i |$$

Y_i ground truth value

Regression task의 Loss 종류

Loss for Regression Task:

$$\text{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

$$\text{L1 Loss (MAE Loss)} = | Y_i - \hat{Y}_i |$$

\hat{Y}_i predicted value

Y_i ground truth value

Huber Loss

Regression task의 Loss 종류

Loss for Regression Task:

$$\text{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

$$\text{L1 Loss (MAE Loss)} = |Y_i - \hat{Y}_i|$$

$$\begin{aligned} \text{Huber Loss} &= \frac{1}{2}(\hat{y}_i - y_i)^2 && \text{if } |\hat{y}_i - y_i| < \delta \\ &= \delta \cdot \left(|\hat{y}_i - y_i| - \frac{\delta}{2} \right) && \text{otherwise} \end{aligned}$$

\hat{Y}_i predicted value

Y_i ground truth value

Regression task의 Loss 종류

Loss for Regression Task:

$$\text{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

\hat{Y}_i predicted value

$$\text{L1 Loss (MAE Loss)} = |Y_i - \hat{Y}_i|$$

Y_i ground truth value

$$\text{Huber Loss} = \frac{1}{2}(\hat{y}_i - y_i)^2$$

if $|\hat{y}_i - y_i| < \delta$

$$\delta \cdot \left(|\hat{y}_i - y_i| - \frac{\delta}{2} \right)$$

otherwise

Log-Cosh Loss

Regression task의 Loss 종류

Loss for Regression Task:

$$\text{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

$$\text{L1 Loss (MAE Loss)} = |Y_i - \hat{Y}_i|$$

$$\text{Huber Loss} = \begin{cases} \frac{1}{2}(\hat{y}_i - y_i)^2 & \text{if } |\hat{y}_i - y_i| < \delta \\ \delta \cdot \left(|\hat{y}_i - y_i| - \frac{\delta}{2} \right) & \text{otherwise} \end{cases}$$

$$\text{Log-Cosh Loss} = \log (\cosh(\hat{y}_i - y_i))$$

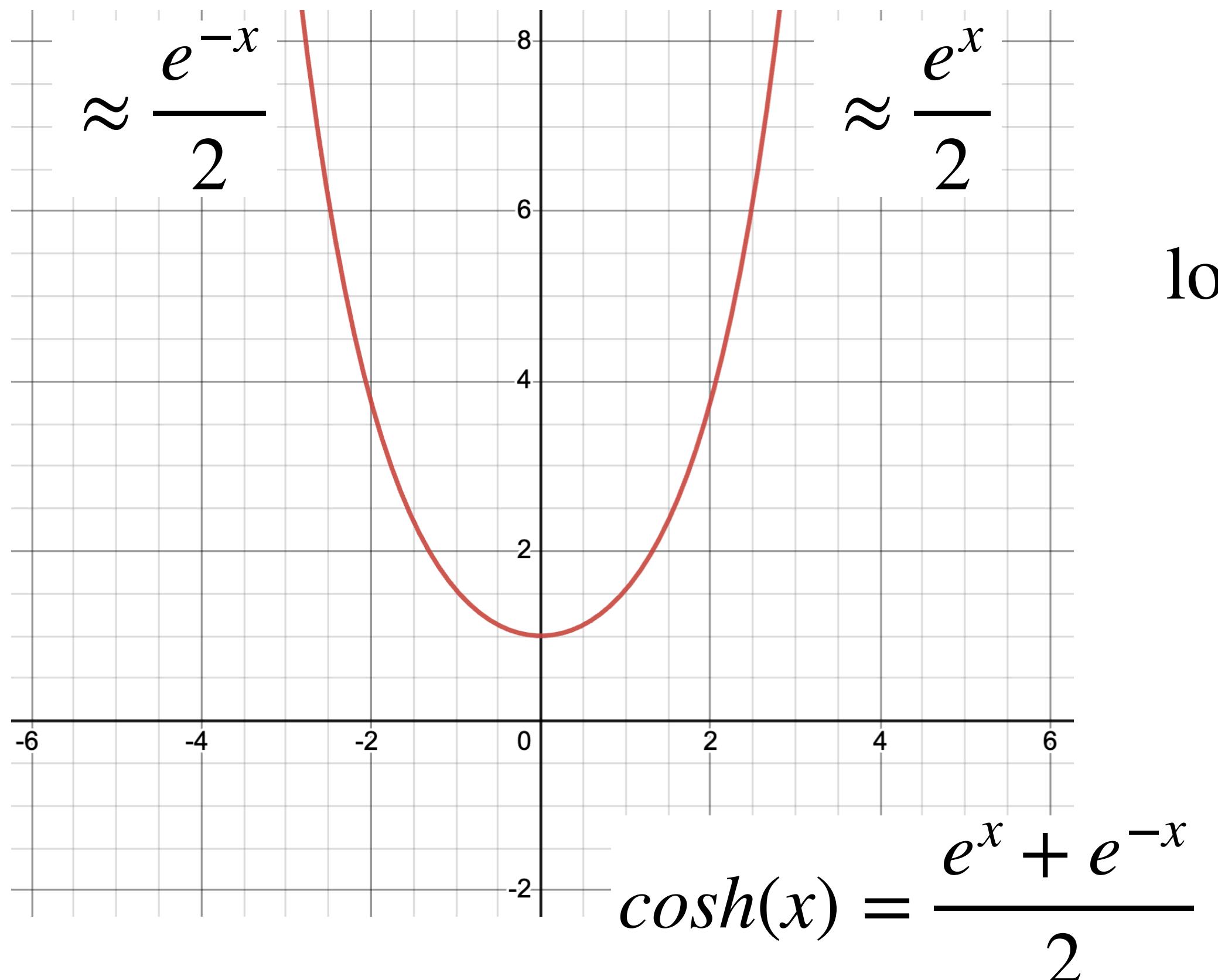
$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$

Regression task의 Loss 종류

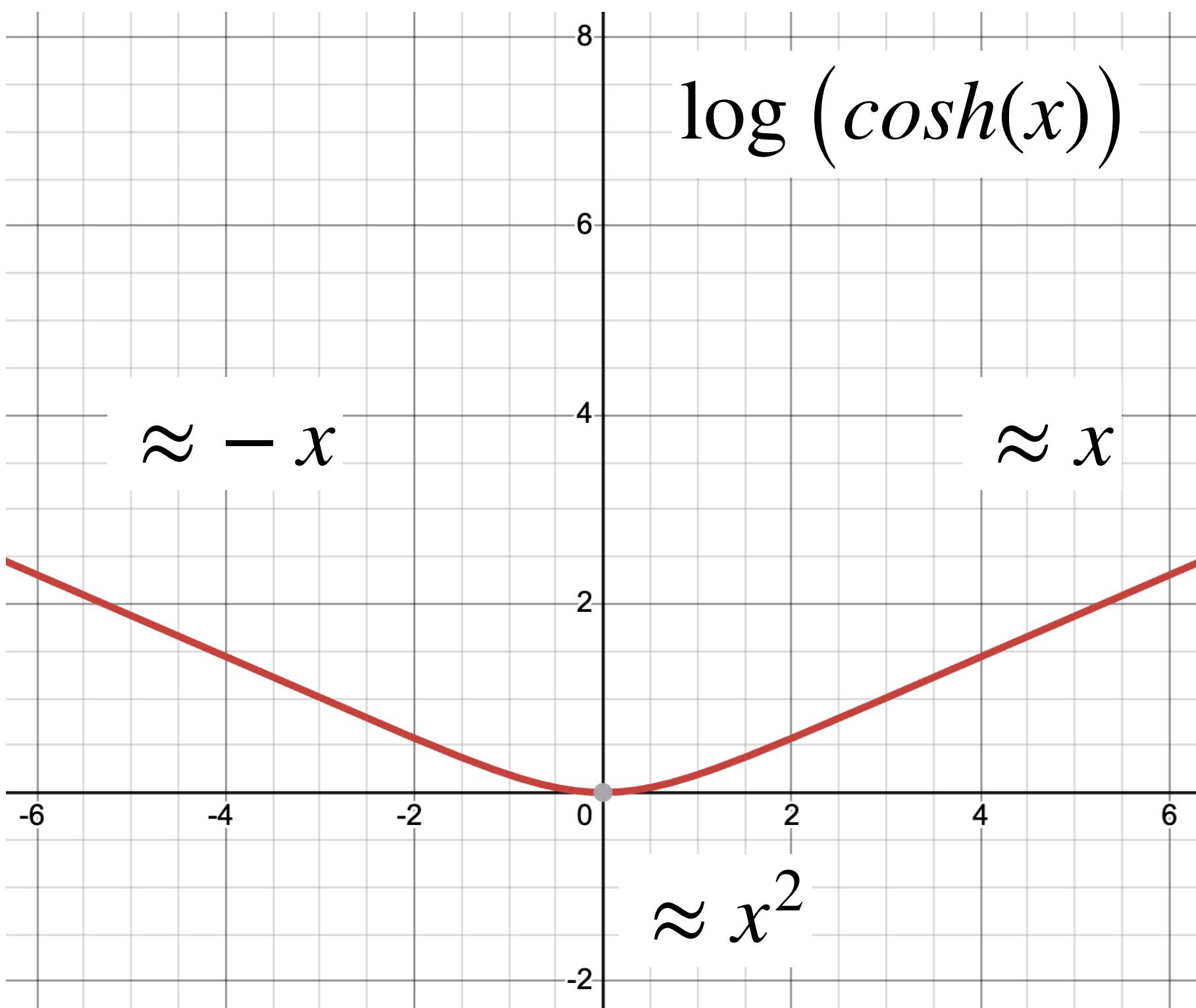
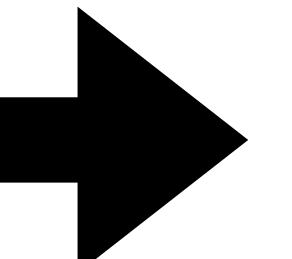
Log-Cosh Loss

$$= \log (\cosh(\hat{y}_i - y_i))$$

$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$



log 을 취하면



Regression task의 Loss 종류

Loss for Regression Task:

$$\text{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

\hat{Y}_i predicted value

$$\text{L1 Loss (MAE Loss)} = |Y_i - \hat{Y}_i|$$

Y_i ground truth value

$$\text{Huber Loss} = \frac{1}{2}(\hat{y}_i - y_i)^2$$

if $|\hat{y}_i - y_i| < \delta$

$$\delta \cdot \left(|\hat{y}_i - y_i| - \frac{\delta}{2} \right)$$

otherwise

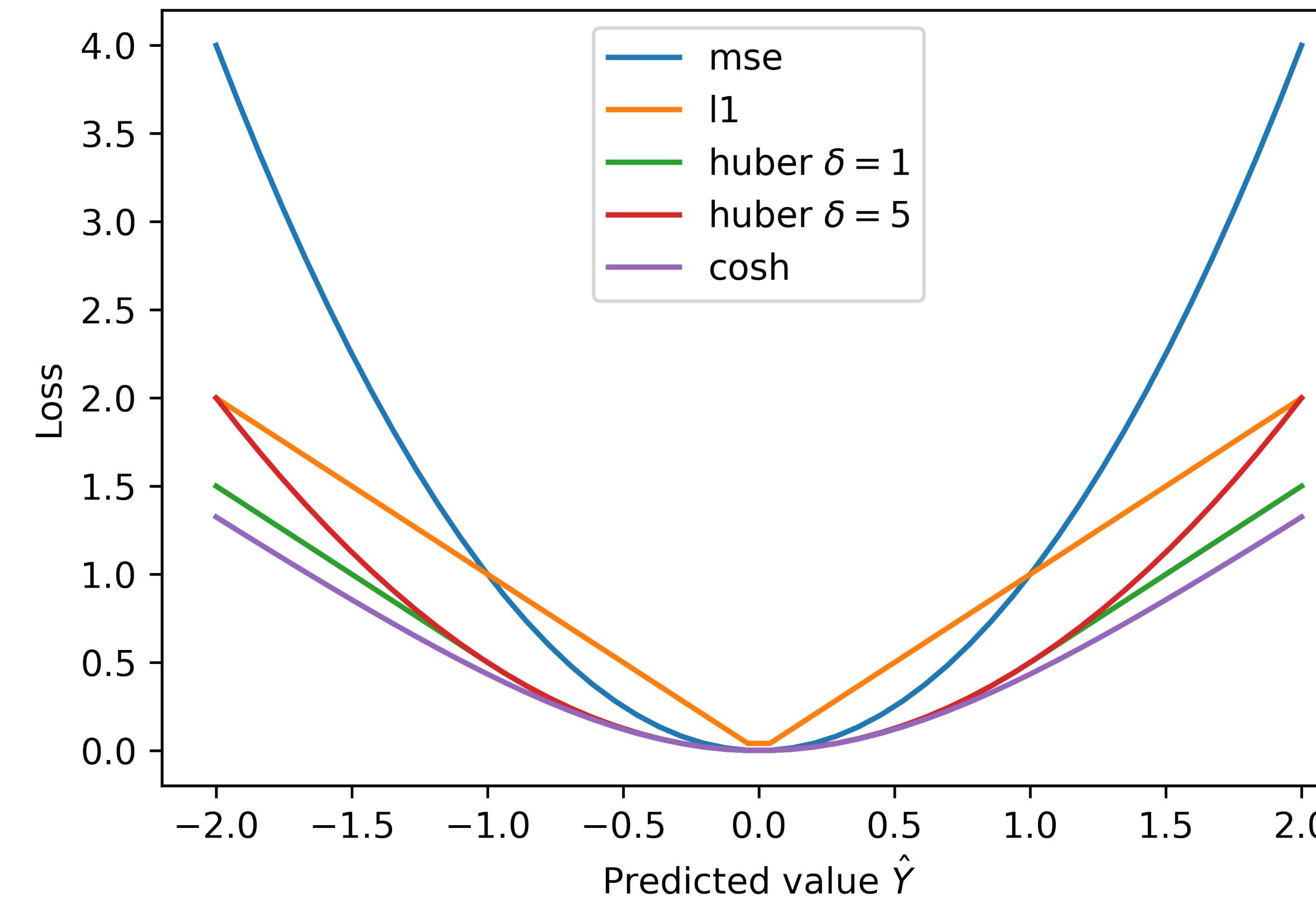
$$\text{Log-Cosh Loss} = \log (\cosh(\hat{y}_i) - y_i))$$

$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$

Loss for Regression task

MSE, L1, Huber Loss, Log-cosh Loss Visualization

Copyright©2023. Acadential. All rights reserved.



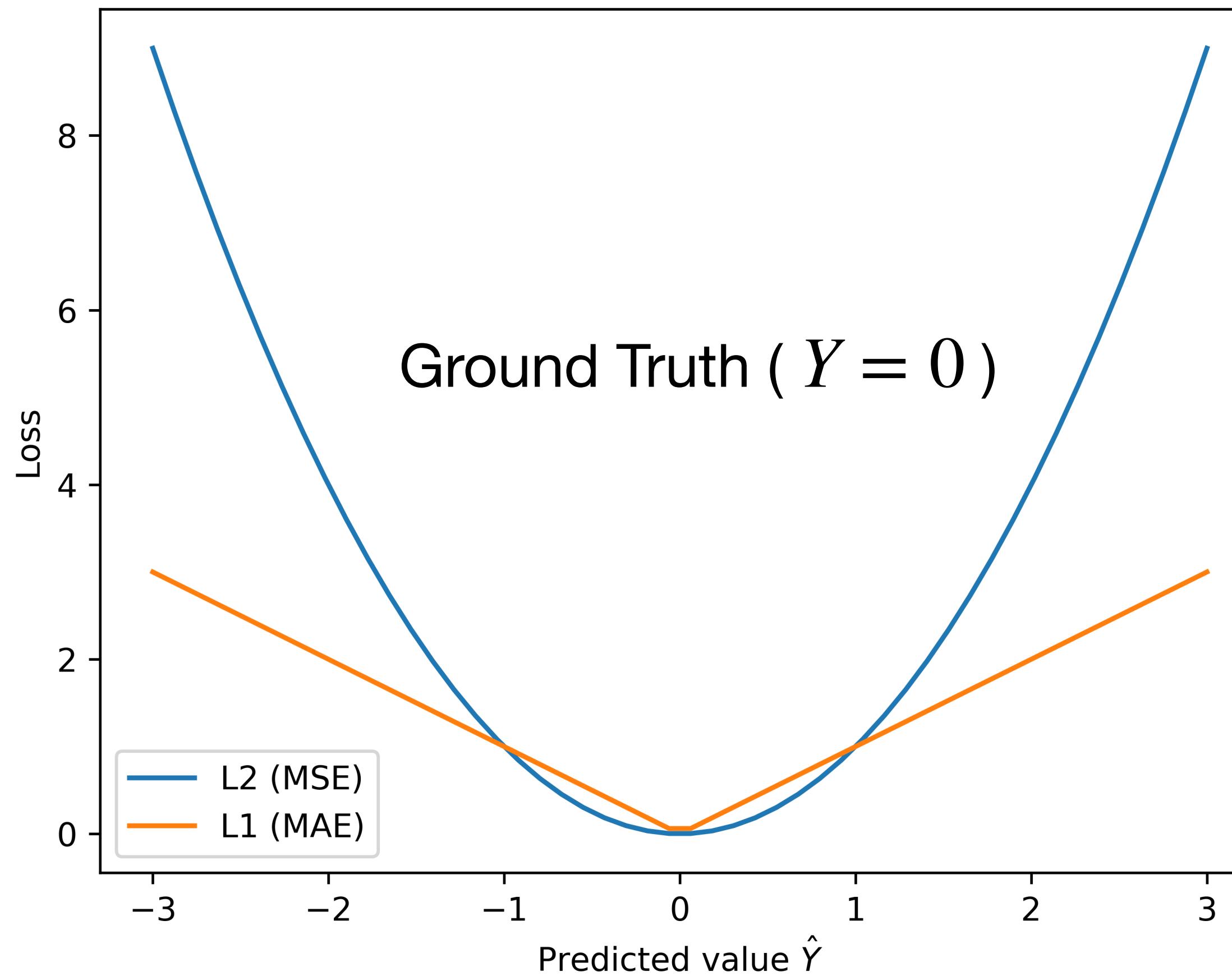
Ground Truth ($Y = 0$)

Diagram Drawn using matplotlib

Loss for Regression task

L2 Loss (MSE) vs. L1 Loss (MAE)

Copyright©2023. Acadential. All rights reserved.



$$\textbf{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

$$\textbf{L1 Loss (MAE Loss)} = |Y_i - \hat{Y}_i|$$

Diagram Drawn using matplotlib

여기서 잠깐... Anomaly란???

Anomaly란...?

Anomaly = 비정상적인 데이터



Images labelled as cat



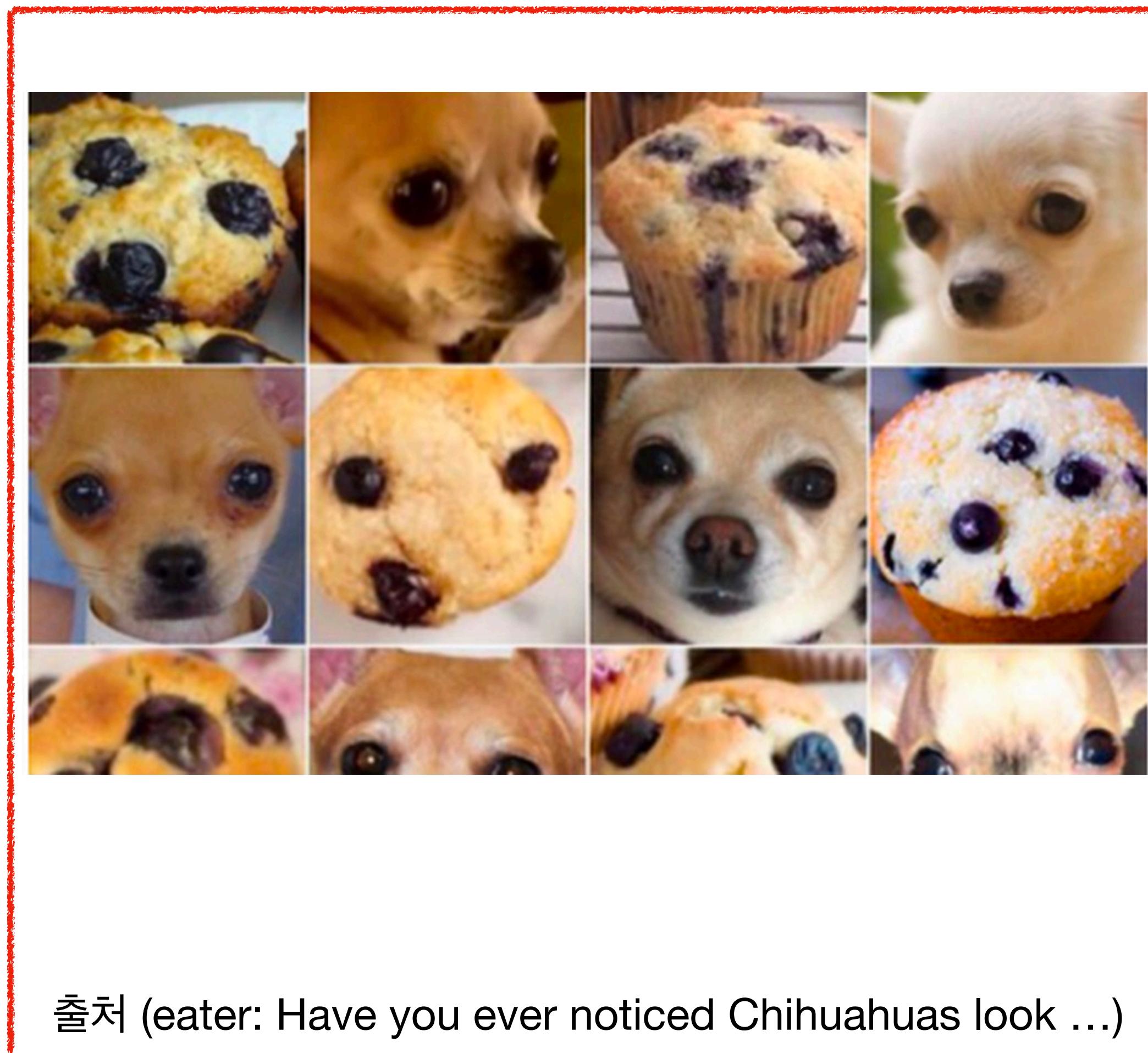
Images labelled as dog



- mislabelled image
- correctly labelled image

Anomaly란...?

Anomaly = 비정상적인 데이터



Images labelled as cat



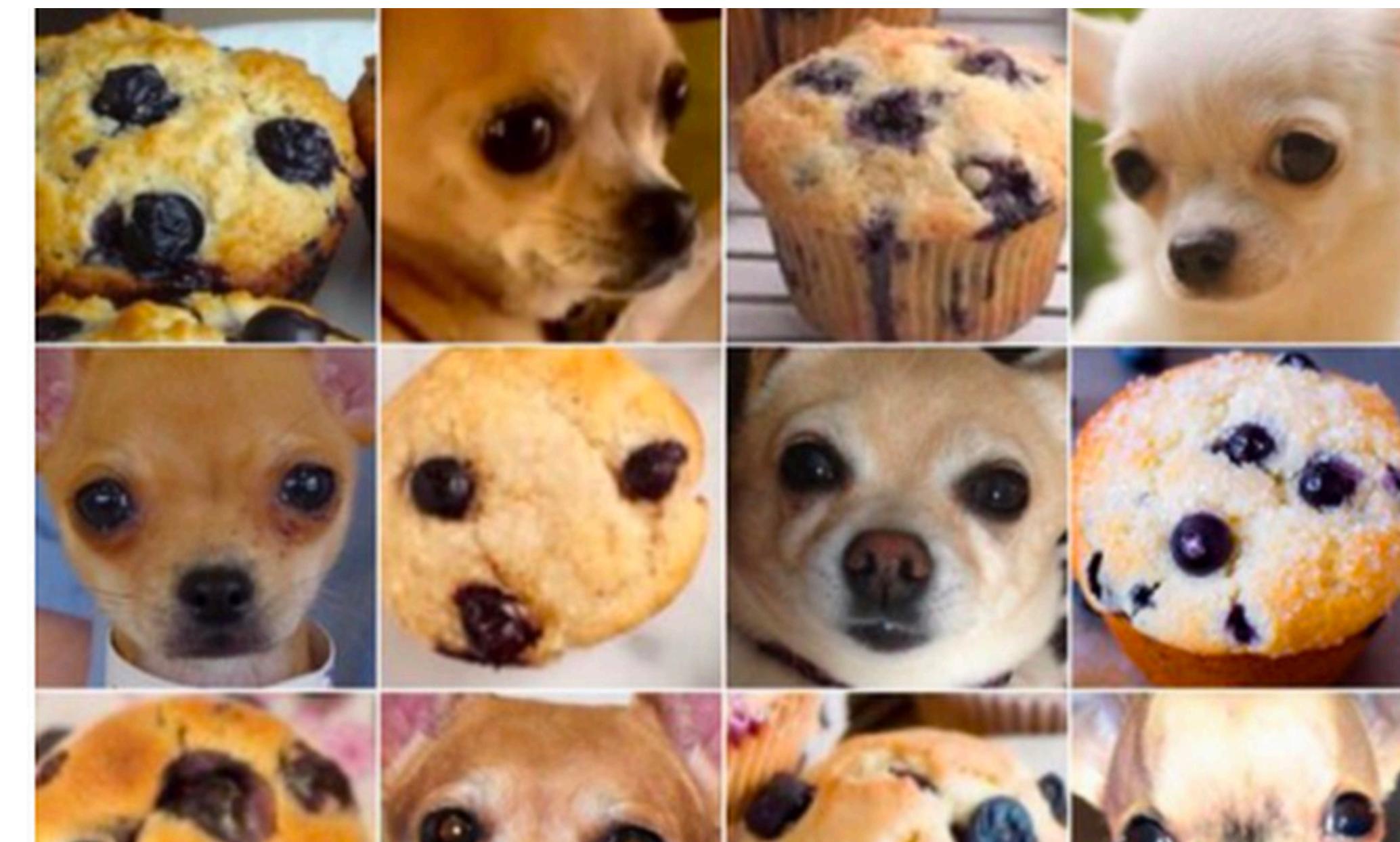
Images labelled as dog



- mislabelled image
- correctly labelled image

Anomaly란...?

Anomaly = 비정상적인 데이터



Images labelled as cat



Images labelled as dog



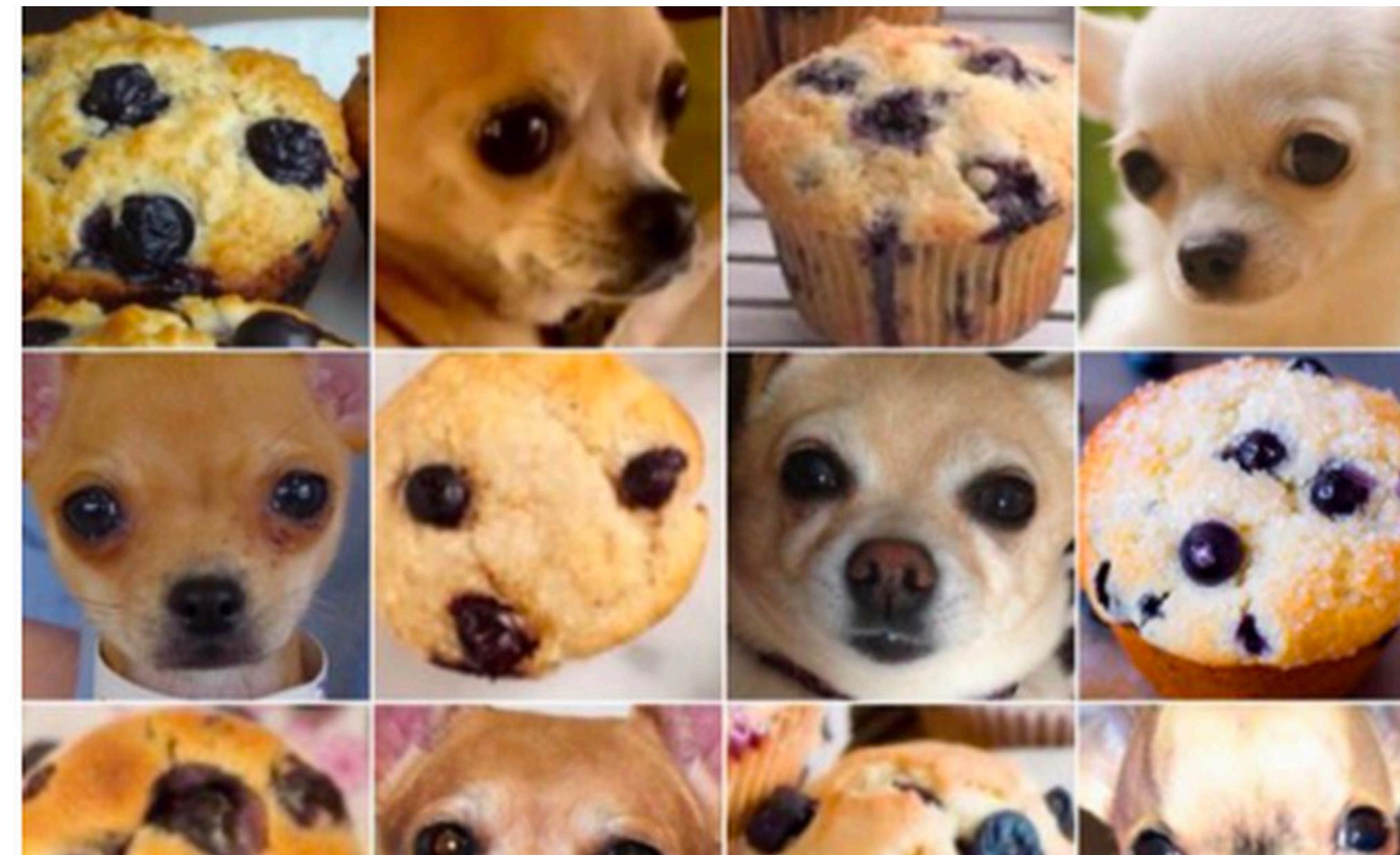
- mislabelled image
- correctly labelled image

출처 (eater: Have you ever noticed Chihuahuas look ...)

36 출처: Allegro tech blog: Trust no one, not even your training data!

Anomaly란...?

Anomaly = 비정상적인 데이터 = $|Y_i - \hat{Y}_i|$ 가 크다!



Images labelled as cat



Images labelled as dog

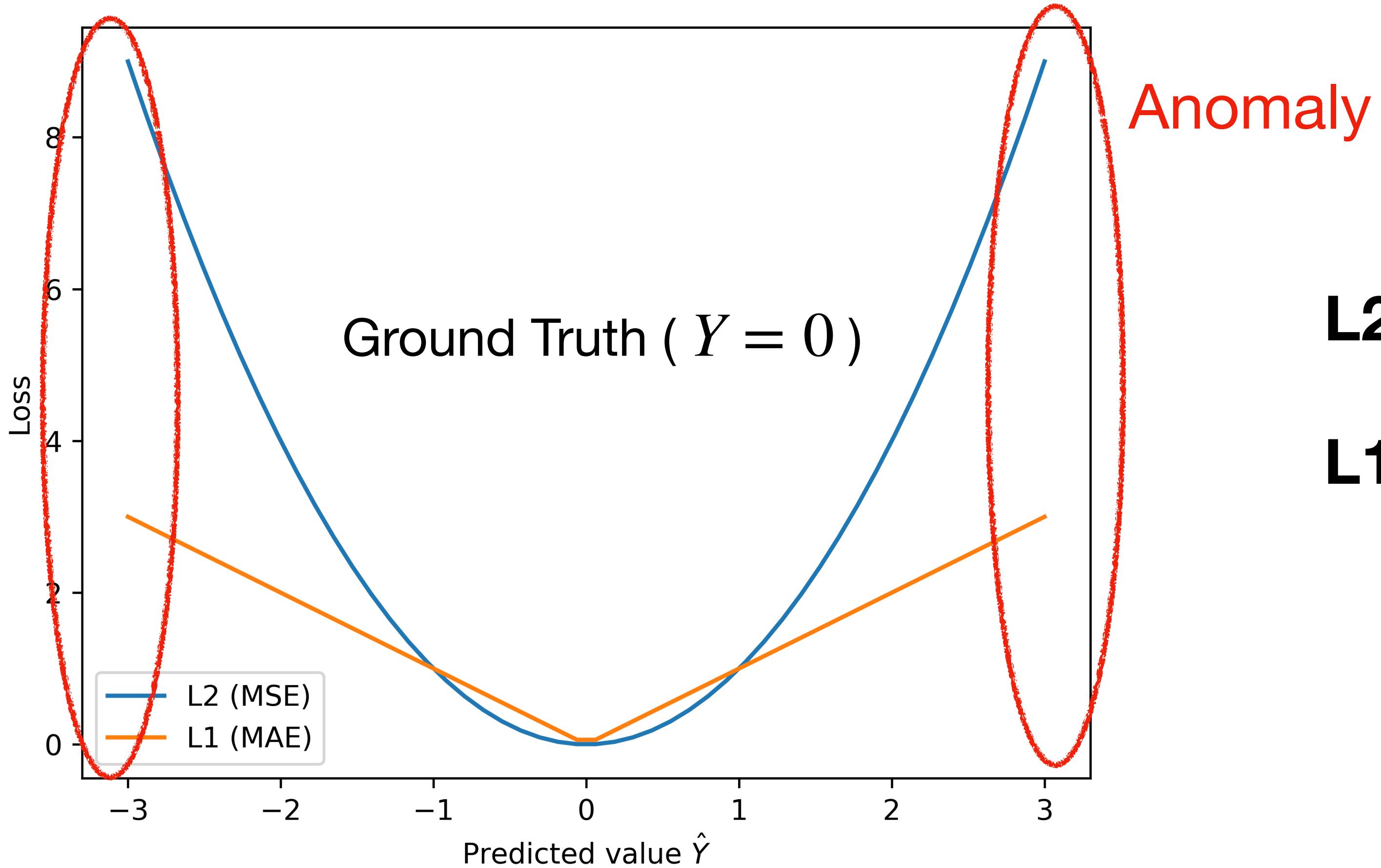


- mislabelled image
- correctly labelled image

다시 돌아와서

Loss for Regression task

L2 Loss (MSE) vs. L1 Loss (MAE)



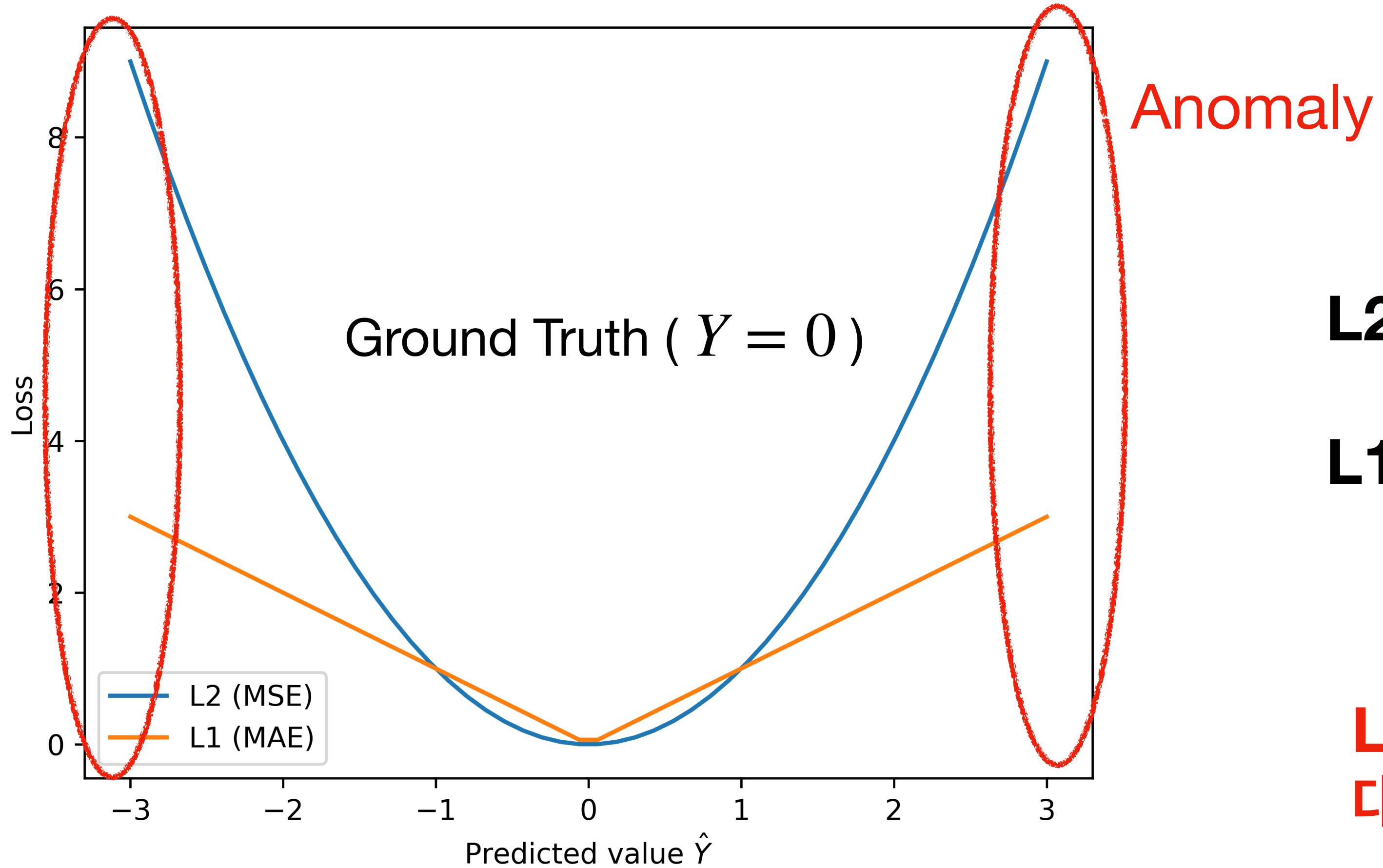
$$\textbf{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

$$\textbf{L1 Loss (MAE Loss)} = | Y_i - \hat{Y}_i |$$

Diagram Drawn using matplotlib

Loss for Regression task

L2 Loss (MSE) vs. L1 Loss (MAE)



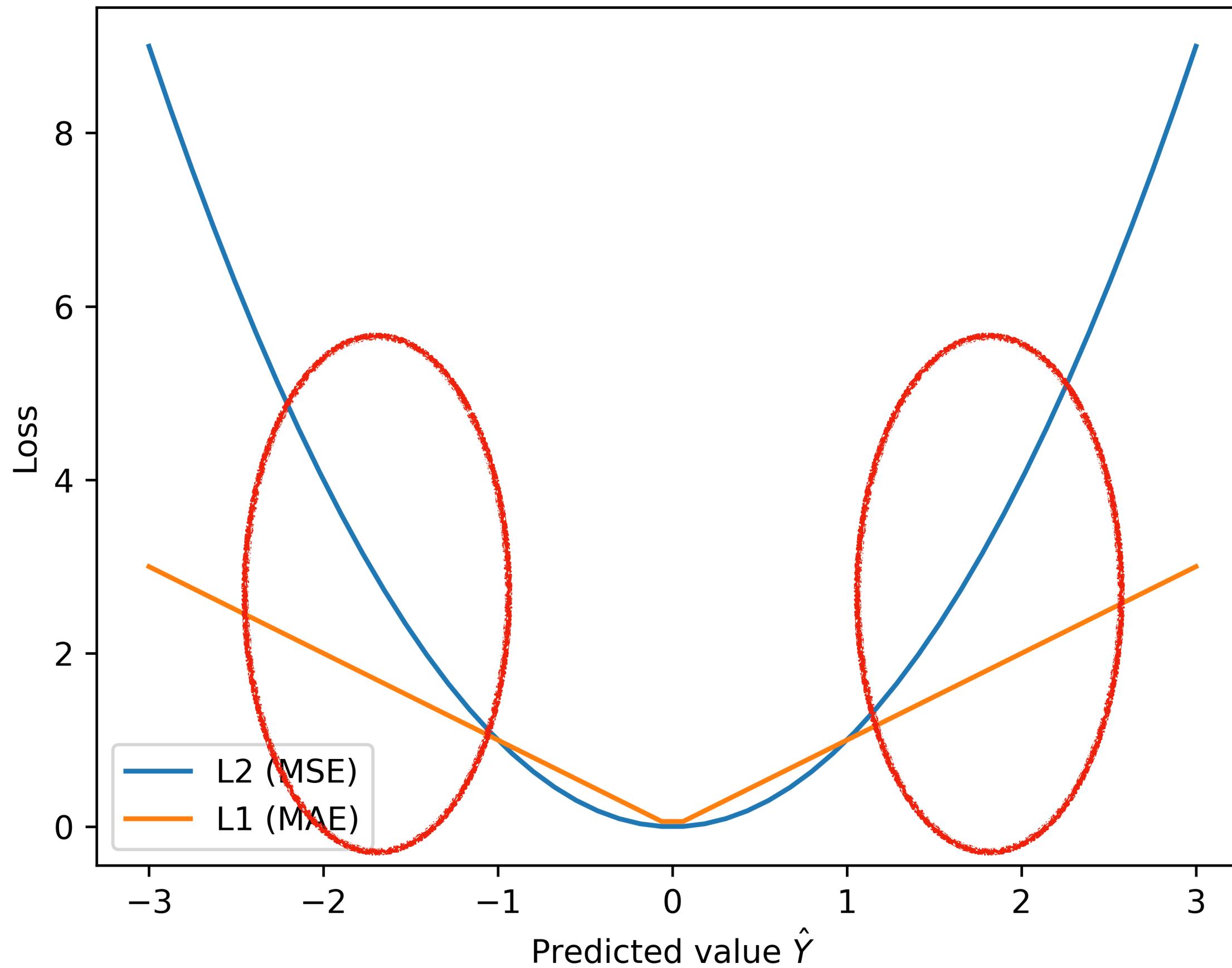
$$\textbf{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

$$\textbf{L1 Loss (MAE Loss)} = |Y_i - \hat{Y}_i|$$

L2 Loss은 L1 Loss보다 Anomaly에 대해서 더 취약하다!

Loss for Regression task

L2 Loss (MSE) vs. L1 Loss (MAE)



$$\textbf{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

$$\textbf{L1 Loss (MAE Loss)} = | Y_i - \hat{Y}_i |$$

L2 Loss은 L1 Loss보다 Neural Network
가 더 빠르게 학습하는데 도움이 된다.

Loss for Regression Tasks

각 Loss의 장단점 및 특징

1. MSE Loss (L2 Loss)

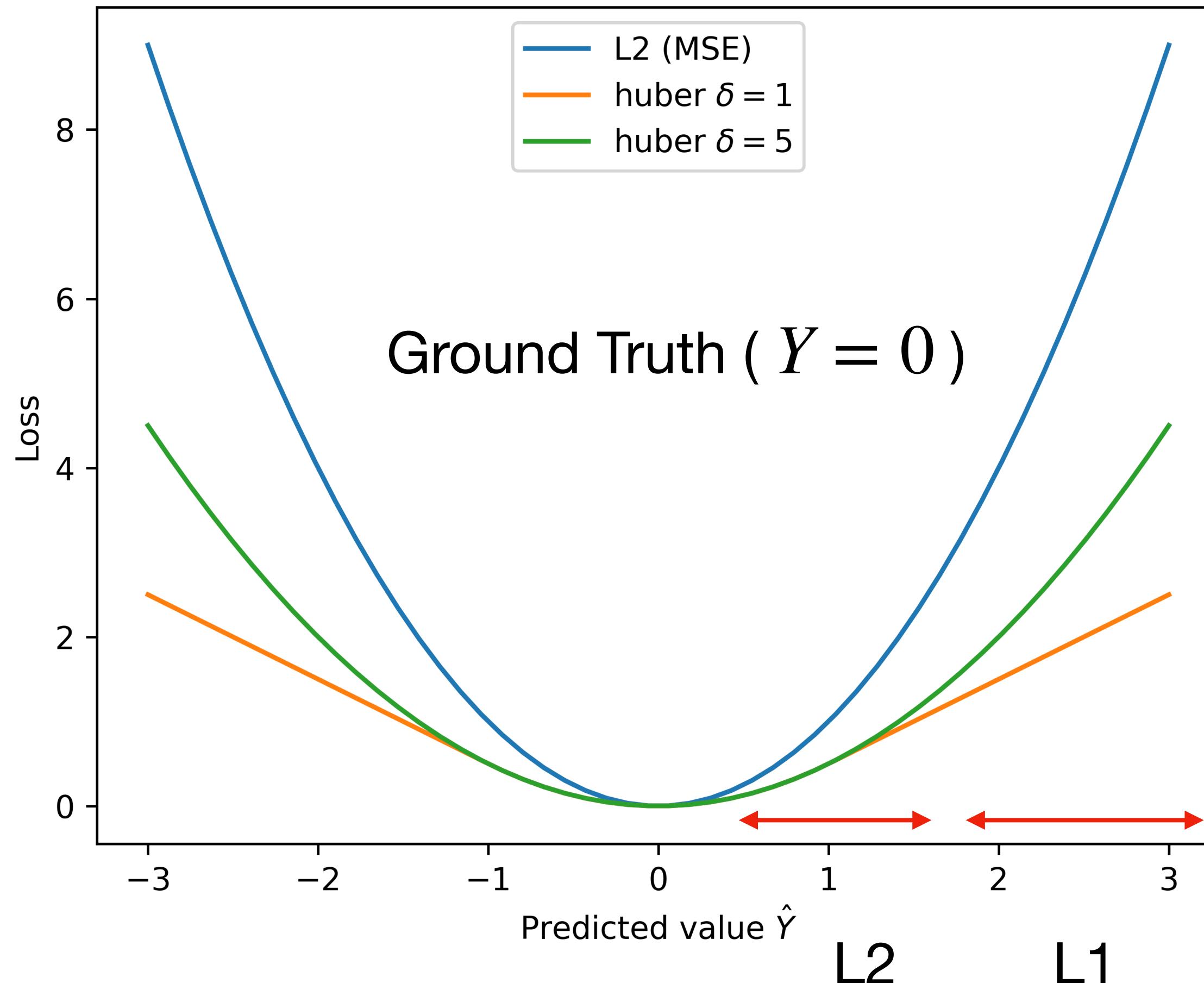
- (a) 장점: Convergence가 더 빠르다.
- (b) b/c 실제값과 차이가 크면 해당 Loss 값은 더욱 크다. (MSE은 Quadratic 하므로)
- (c) 단점: Anomaly (Outlier)에 대해서 더 민감하다.

2. MAE Loss (L1 Loss)

- (a) 장점: MSE와 비교했을때 Anomaly에 대해서 비교적 더 robust하다.
- (b) 단점: 0에서 미분가능하지 않다.

Loss for Regression task

L2 Loss (MSE) vs. Huber Loss ($\sigma = 1, \sigma = 5$)



$$\text{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

Huber Loss =

$$\frac{1}{2}(\hat{y}_i - y_i)^2$$

$$\delta \cdot \left(|\hat{y}_i - y_i| - \frac{\delta}{2} \right)$$

if $|\hat{y}_i - y_i| < \delta$

otherwise

Loss for Regression Tasks

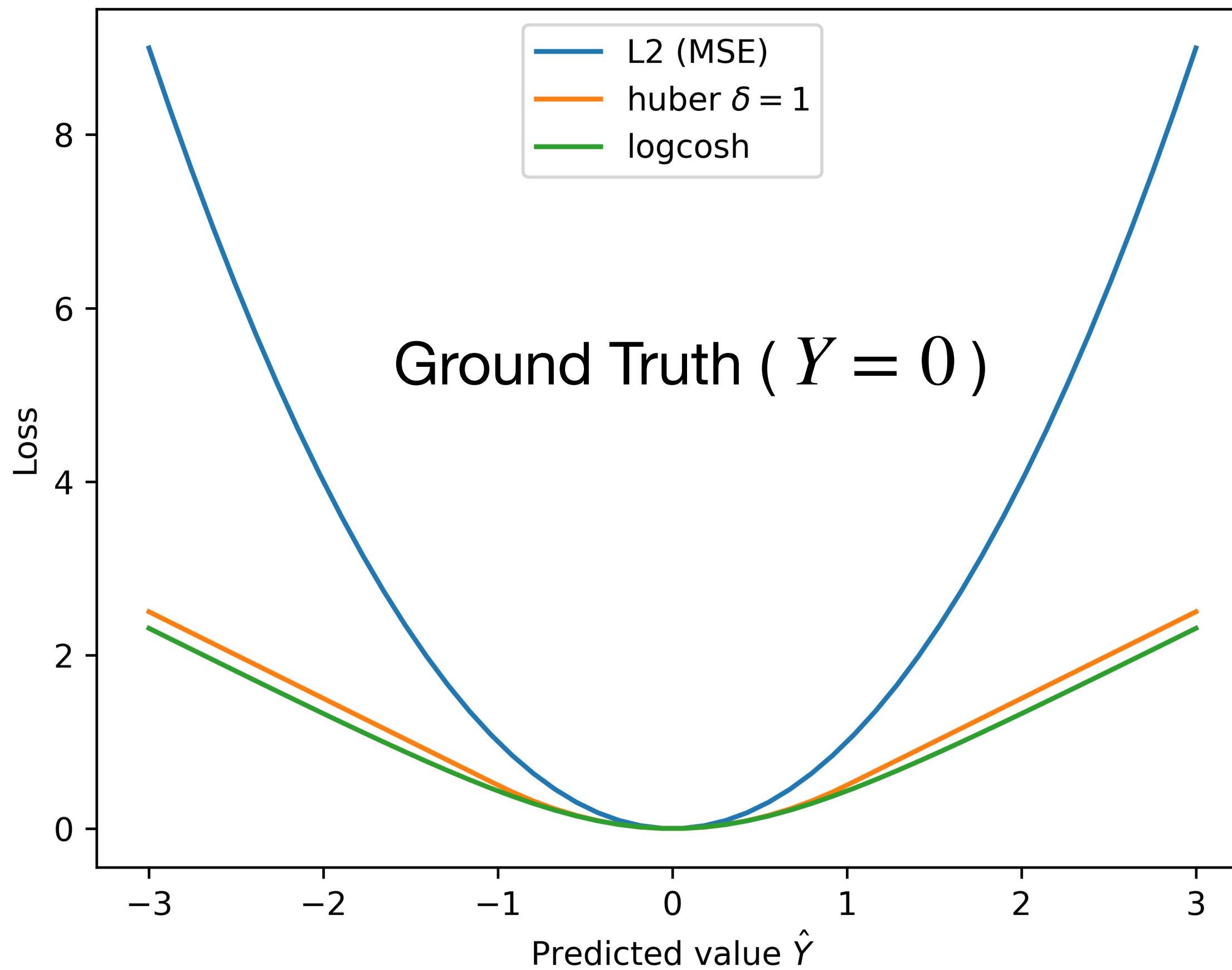
각 Loss의 장단점 및 특징

3. Huber Loss

- (a) 장점: MSE의 faster convergence의 장점과 MAE의 outlier에 대한 robustness 장점을 둘 다 가지고 있다.
- (b) 단점: 한 번만 전구간 미분가능하다.

Loss for Regression task

L2 Loss (MSE) vs. Huber Loss ($\sigma = 1$) vs. Log Cosh Loss



$$\text{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

Huber Loss =

$$\frac{1}{2}(\hat{y}_i - y_i)^2 \quad \text{if } |\hat{y}_i - y_i| < \delta$$

$$\delta \cdot \left(|\hat{y}_i - y_i| - \frac{\delta}{2} \right) \quad \text{otherwise}$$

$$\text{Log-Cosh Loss} = \log (\cosh(\hat{y}_i) - y_i)$$

Loss for Regression Tasks

각 Loss의 장단점 및 특징

3. Huber Loss

- (a) 장점: MSE의 faster convergence의 장점과 MAE의 outlier에 대한 robustness 장점을 둘 다 가지고 있다.
- (b) 단점: 한 번만 전구간 미분가능하다.

4. Log Cosh

- (a) Huber Loss와 비슷한 성격을 가졌지만 전구간 두 번 이상 미분가능하다.

참고 사항: 왜 미분가능함이 중요한가?

Copyright©2023. Acadential. All rights reserved.

Smoothness & Well-behaved function

1. “미분가능함” = **Smooth** (매끄럽다)고 **Well-behaved** (얌전하다)
2. 함수의 출력값이 입력값의 변화에 따라 “매끄럽고 부드럽게” 변함.
3. 매끄럽고 부드러운 함수 → extrapolate, interpolate하기 수월함

b/c 학습된 딥러닝 모델이 학습에서 보지 못한 input data space에 대해서 interpolate 혹은 extrapolate 가능해진다.

3-4. Soft Label vs. Hard Label

Soft Label vs. Hard Label

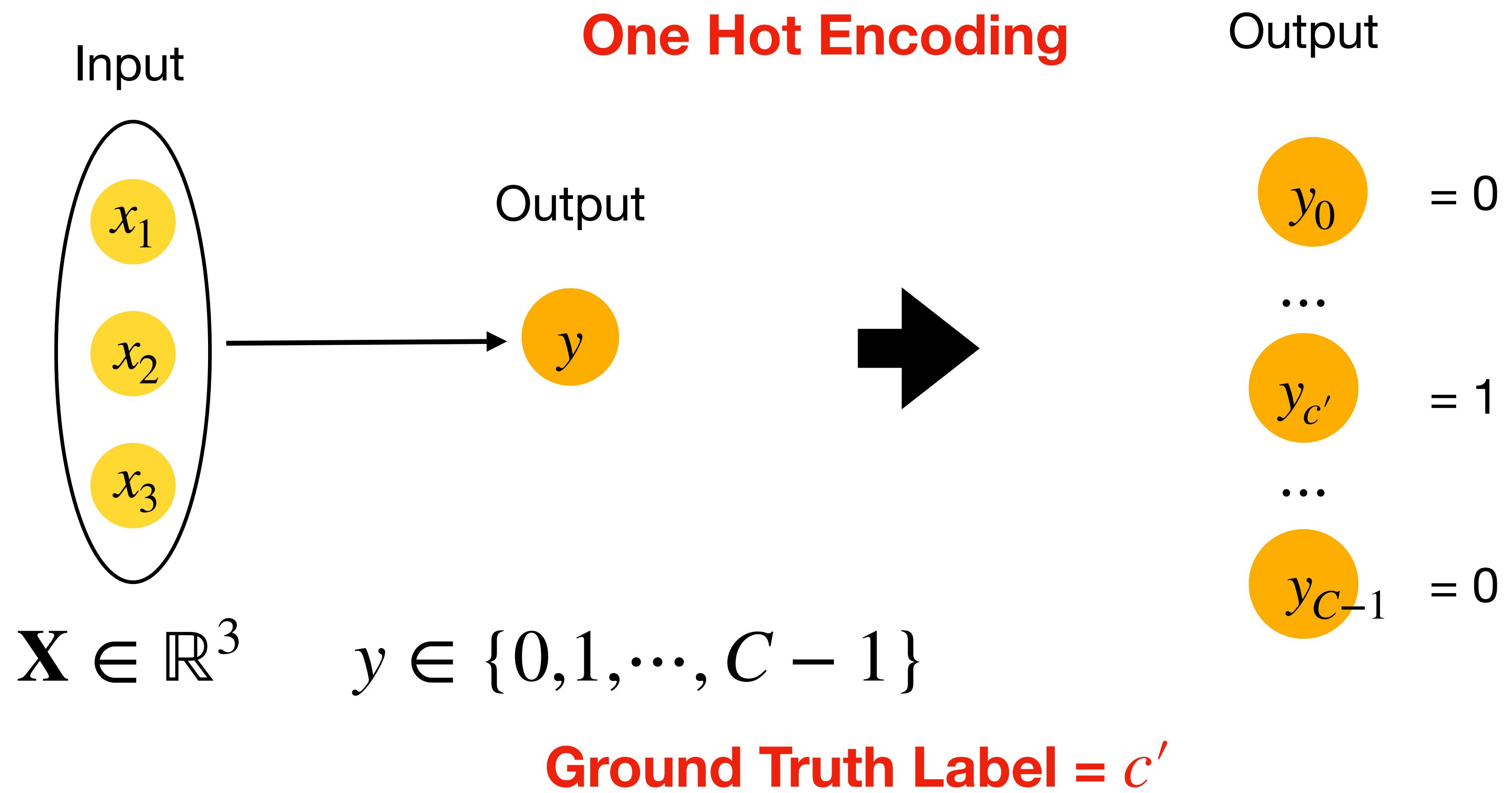
Hard Label

- 데이터 값이 여러 Class 중 하나로 명확하게 분류되는 경우.

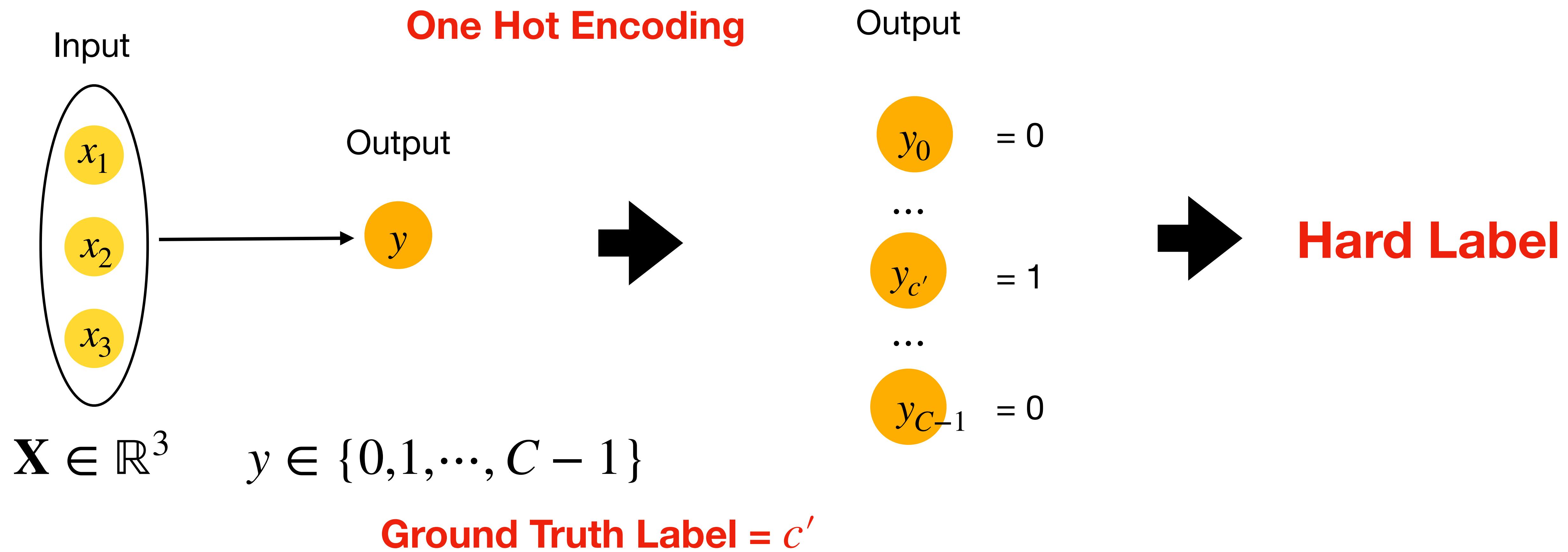
Soft Label

- 데이터 값이 여러 Class 중 하나로 명확하게 분류되지 않고, Label이 확률로 주어지는 경우.

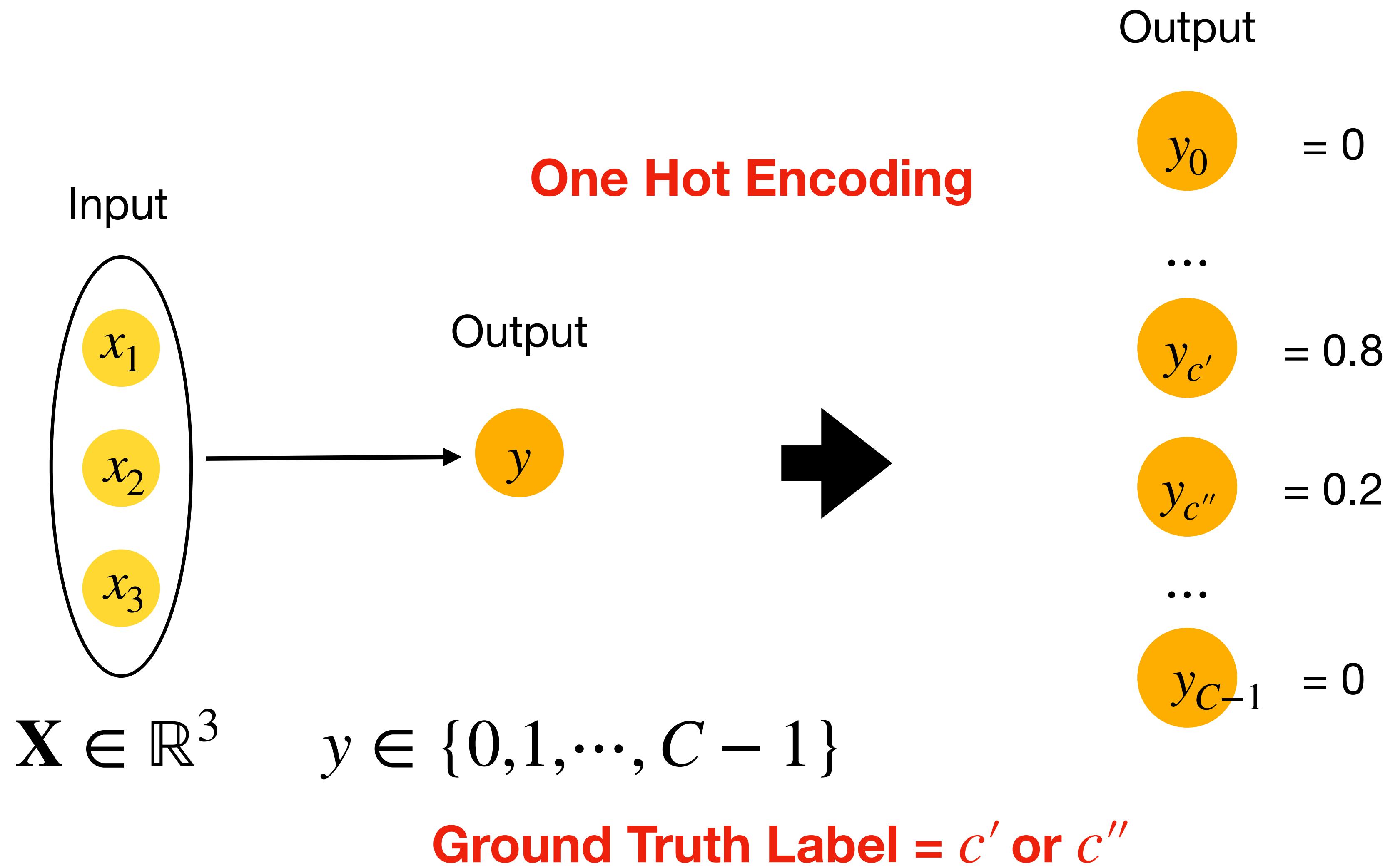
Soft Label vs. Hard Label



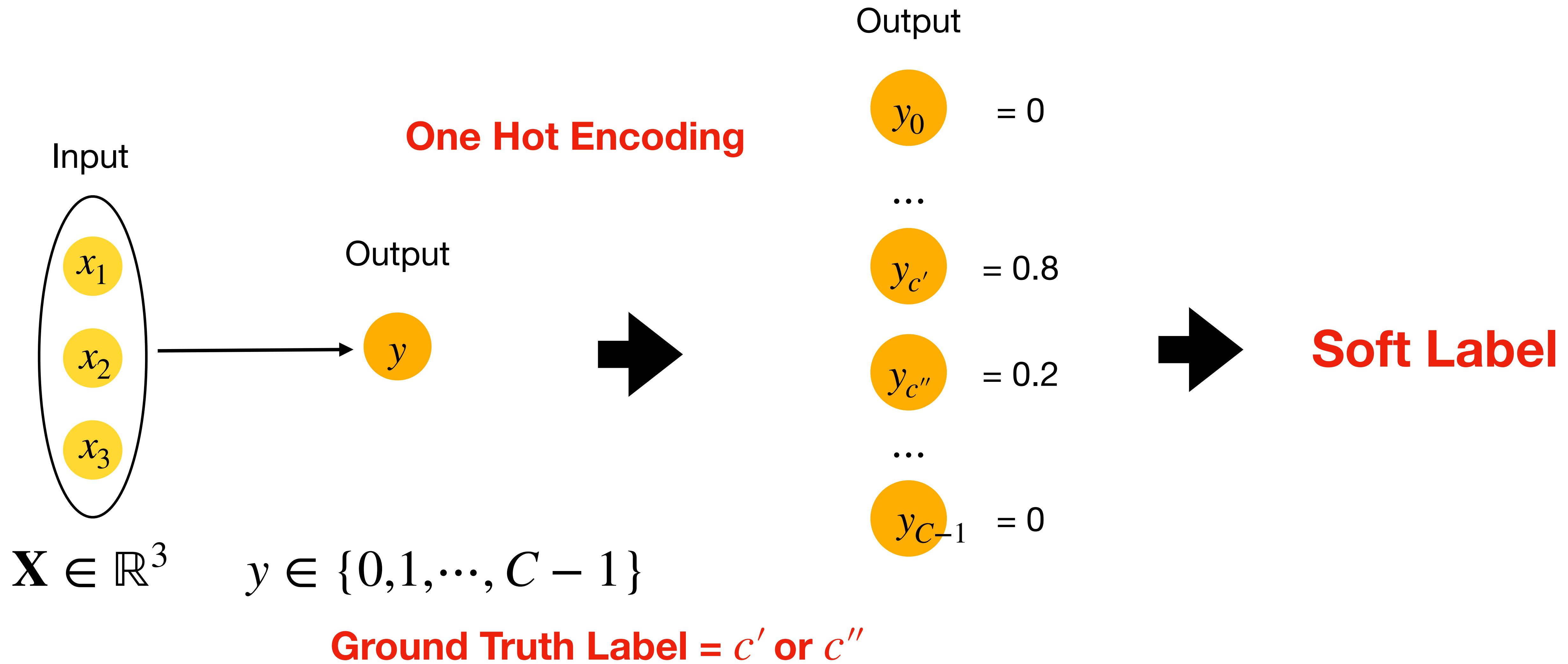
Soft Label vs. Hard Label



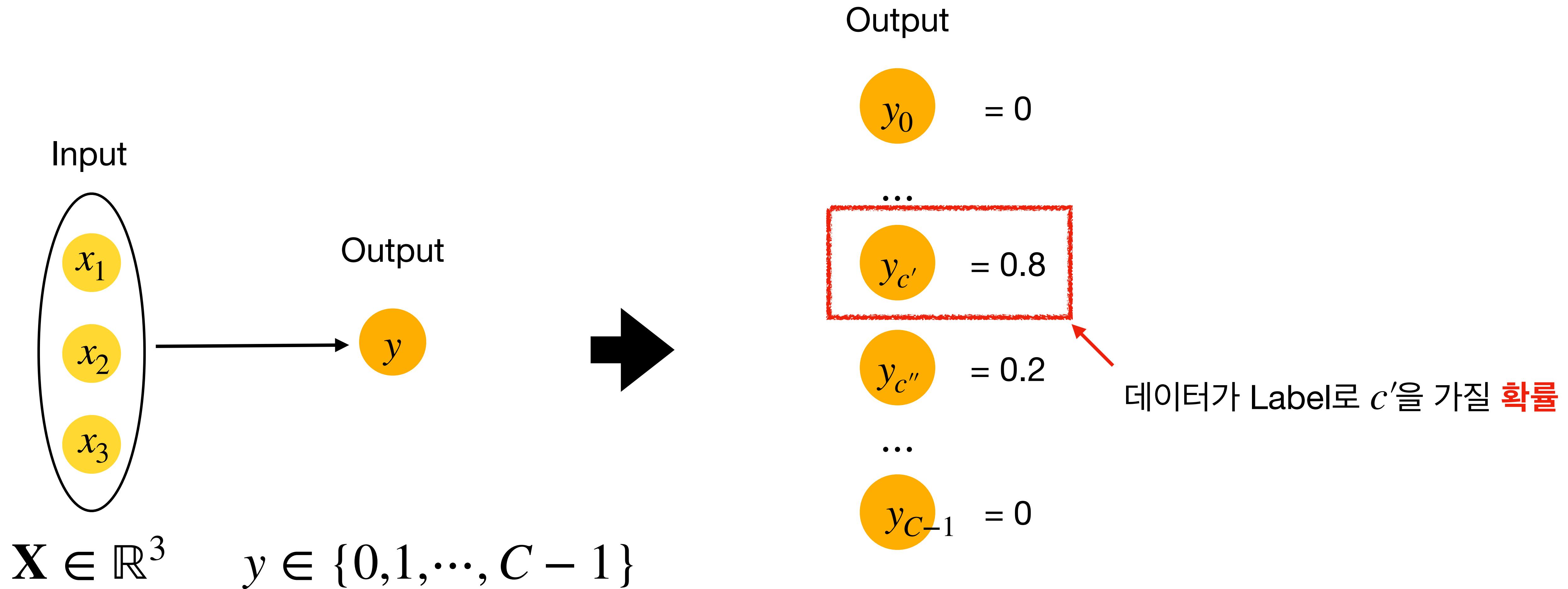
Soft Label vs. Hard Label



Soft Label vs. Hard Label



Soft Label vs. Hard Label



3-5. Classification Task의 Loss Function

Classification task의 Loss 종류

Copyright©2023. Acadential. All rights reserved.

- L1 Loss (MAE Loss)
- Kullback-Leibler Divergence Loss (KL Divergence)
- Cross Entropy Loss

Classification task의 Loss 종류

Ground Truth Label

$Y_{i,c}$ = i번째 data의 label이 class c일 확률.

Predicted Label

$\hat{Y}_{i,c}$ = i번째 data의 label이 class c일 것이라고 모델이 예측한 확률.

Classification task의 Loss 종류

Ground Truth Label

$Y_{i,c}$ = i번째 data의 label이 class c일 확률.

Predicted Label

$\hat{Y}_{i,c}$ = i번째 data의 label이 class c일 것이라고 모델이 예측한 확률.

Hard Label일 경우

$$Y_{i,c} \in \{0,1\}$$

Soft Label일 경우

$$Y_{i,c} \in [0,1]$$

Classification task의 Loss 종류

- L1 Loss (MAE Loss)

$$\sum_{c=1}^C |Y_{i,c} - \hat{Y}_{i,c}|$$

- Cross Entropy Loss

$$\sum_{c=1}^C -Y_{i,c} \log \hat{Y}_{i,c}$$

- Kullback-Leibler Divergence Loss (KL Divergence)

$$\sum_{c=1}^C Y_{i,c} \cdot \log \left(\frac{Y_{i,c}}{\hat{Y}_{i,c}} \right)$$

Classification task의 Loss 종류

Copyright©2023. Acadential. All rights reserved.

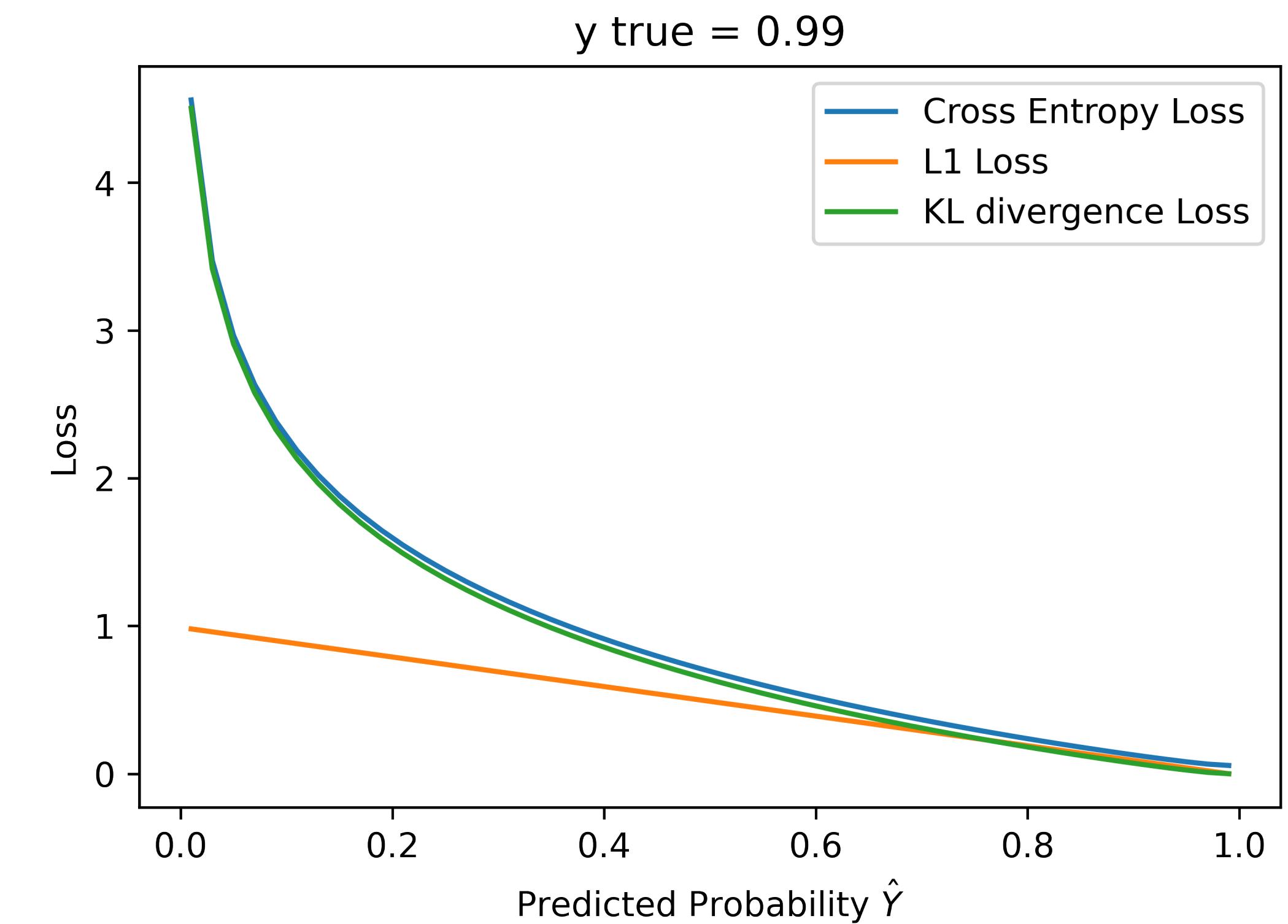
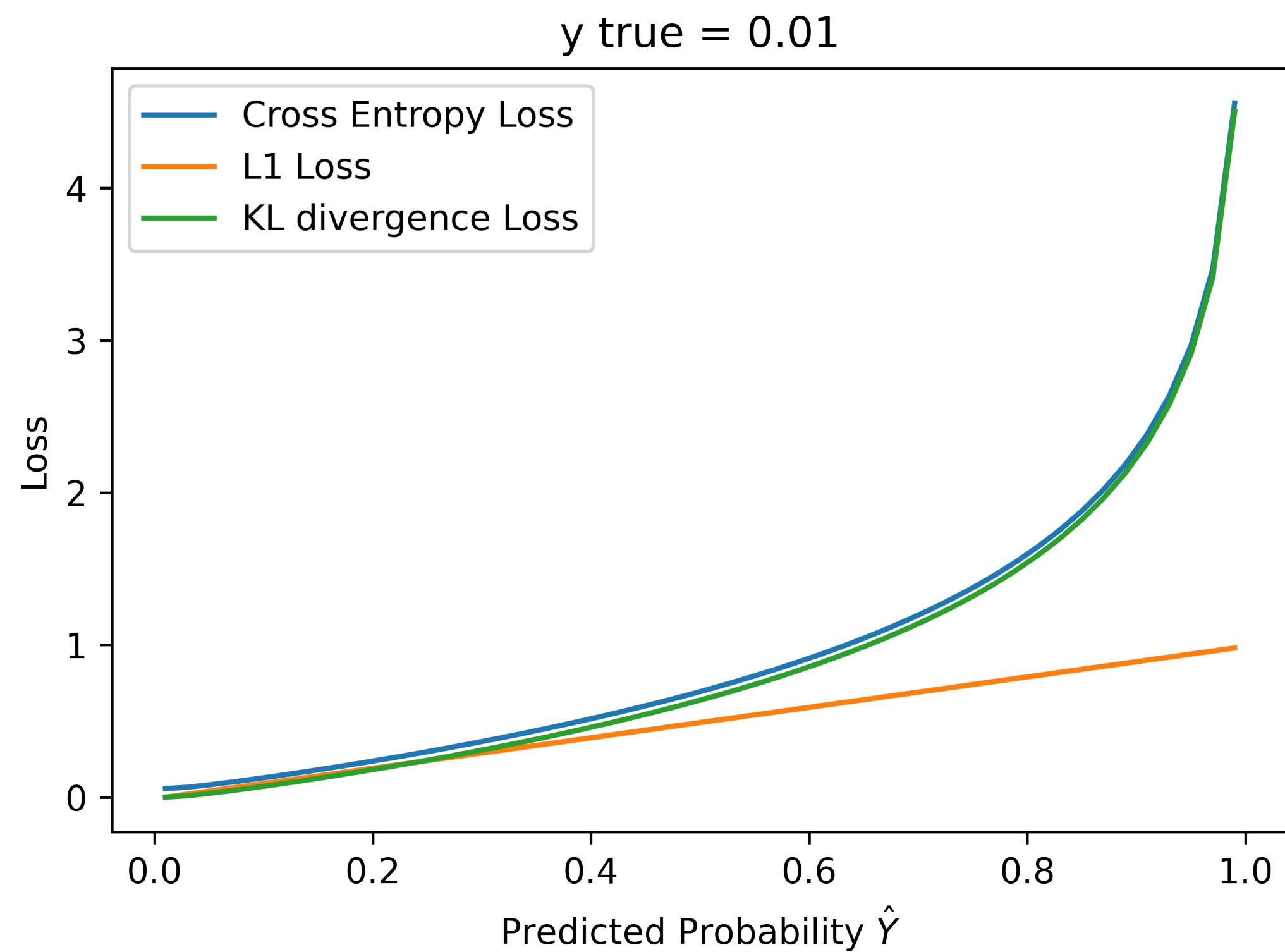


Diagram Drawn using matplotlib

Classification task의 Loss 종류

- **Cross Entropy Loss, KL Divergence Loss**
 - (a) 장점: MAE Loss에 비해서 학습 속도가 더 빠르다
 - (b) 단점: Outlier 및 Noise에 대해서 더 취약, 민감하다.
- **MAE Loss**
 - (c) 장점: Outlier 및 Noise에 대해서 더 robust하다.
 - (d) 단점: Cross Entropy Loss에 비해서 학습 속도가 느리다.

Classification task의 Loss 종류

Copyright©2023. Acadential. All rights reserved.

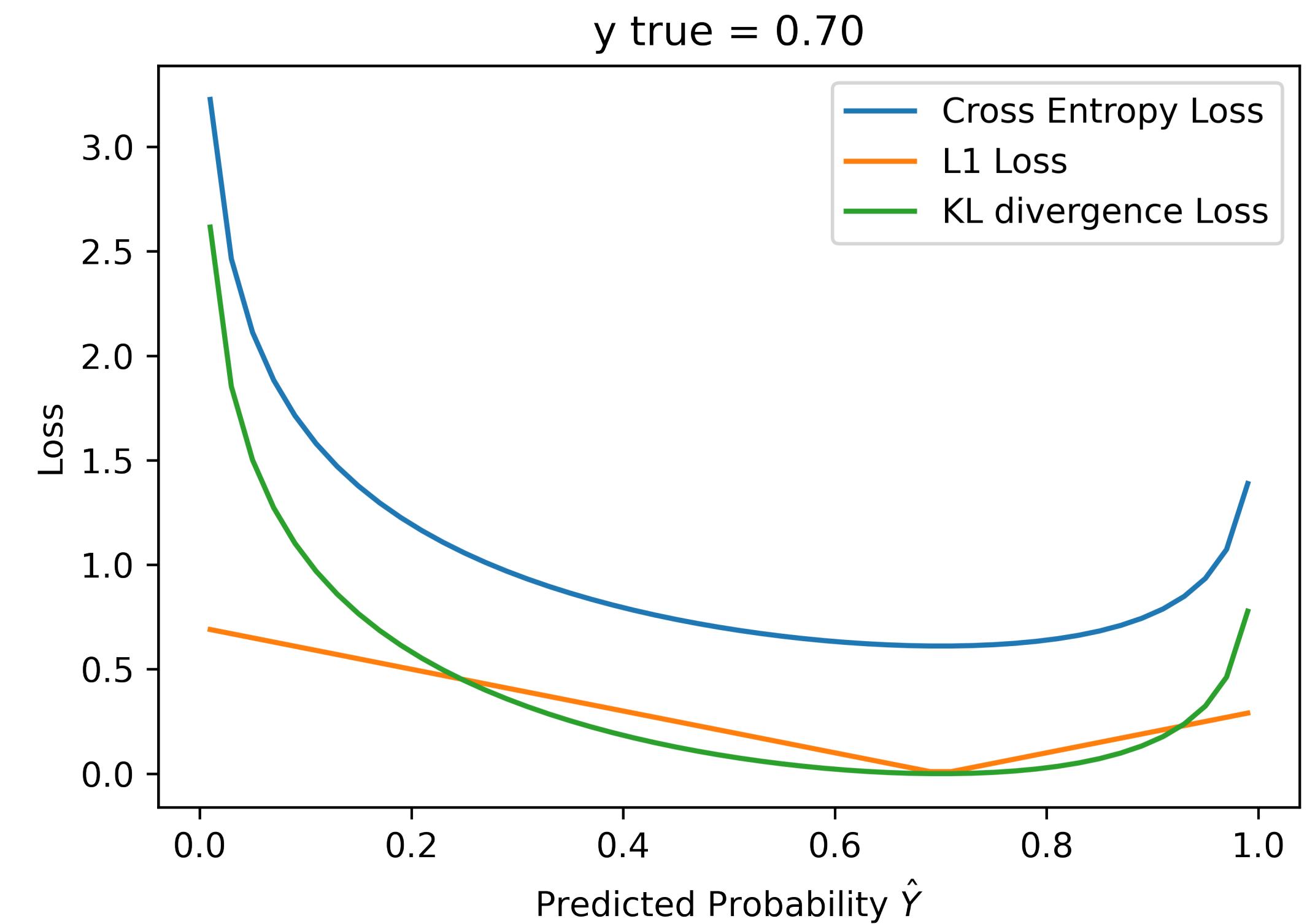
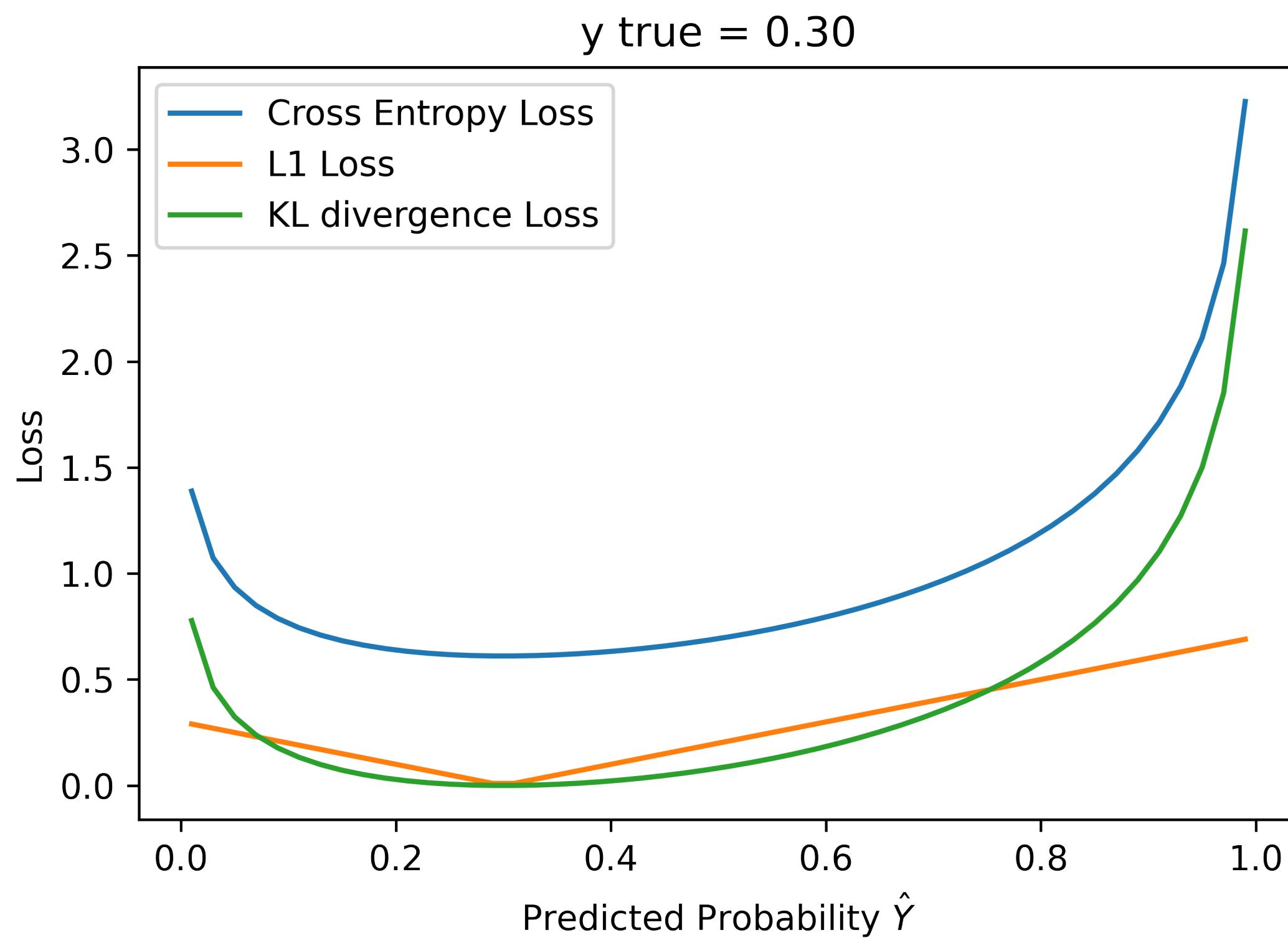


Diagram Drawn using matplotlib

3-6. [실습] PyTorch로 구현해보는 Loss Function

PyTorch로 구현해보는 Loss Function

- 학습 단계에 대한 예시 Pseudo Code

```
# Pseudo Code
# initialize loss function
loss_fn = nn.MSELoss(reduction="mean")

for batch in dataloader:
    input, ground_truth = batch
    optimizer.zero_grad()
    predicted = model(input) # forward pass

    # compute loss
    loss = loss_fn(predicted, ground_truth)

    loss.backward() # backward pass
    optimizer.step() # step function
```

- loss.backward()**을 통해서 loss에 대한 gradient $\nabla_w L$ 을 계산한다.
- optimizer.step()**을 통해서 $\nabla_w L$ 으로 gradient descent한다

PyTorch로 구현해보는 Loss Function

reduction (“mean”과 “none”의 차이)

- Loss을 처음 initialize할 때
 - reduction=“mean”
 - reduction=“none”
- 으로 지정할 수 있다.

```
# initialize loss function
loss_fn = nn.MSELoss(reduction="mean")

for batch in dataloader:
    predicted = model(batch) # forward pass

    # compute loss
    loss = loss_fn(predicted, ground_truth)

    loss.backward() # backward pass
    optimizer.step() # step function
```

PyTorch로 구현해보는 Loss Function

reduction (“mean”과 “none”의 차이)

- 예측값 $\hat{Y} \in \mathbb{R}^{B \times \dots}$, 실제값 $Y \in \mathbb{R}^{B \times \dots}$
- (reduction = “none”)의 Loss 함수의 출력값:

$$L(\hat{Y}, Y) \in \mathbb{R}^B$$

- (reduction = “mean”)의 Loss 함수의 출력값:

$$\frac{1}{B} \sum_i^B L(\hat{Y}_i, Y_i) \in \mathbb{R}$$

```
# initialize loss function
loss_fn = nn.MSELoss(reduction="mean")

for batch in dataloader:
    predicted = model(batch) # forward pass

    # compute loss
    loss = loss_fn(predicted, ground_truth)

    loss.backward() # backward pass
    optimizer.step() # step function
```

PyTorch로 구현해보는 Loss Function

reduction (“mean”과 “none”의 차이)

- 예측값 $\hat{Y} \in \mathbb{R}^{B \times \dots}$, 실제값 $Y \in \mathbb{R}^{B \times \dots}$
- (reduction = “none”)의 Loss 함수의 출력값:
$$L(\hat{Y}, Y) \in \mathbb{R}^B$$
- (reduction = “mean”)의 Loss 함수의 출력값:

$$\frac{1}{B} \sum_i^B L(\hat{Y}_i, Y_i) \in \mathbb{R}$$

각 data sample 별로 Loss을 구한다.

각 data sample 별로 Loss을 기록하는 용도로
주로 **evaluation** 단계에서 사용하는 mode.

PyTorch로 구현해보는 Loss Function

reduction (“mean”과 “none”의 차이)

- 예측값 $\hat{Y} \in \mathbb{R}^{B \times \dots}$, 실제값 $Y \in \mathbb{R}^{B \times \dots}$
- (reduction = “none”)의 Loss 함수의 출력값:

$$L(\hat{Y}, Y) \in \mathbb{R}^B$$

- (reduction = “mean”)의 Loss 함수의 출력값:

$$\frac{1}{B} \sum_i^B L(\hat{Y}_i, Y_i) \in \mathbb{R}$$

각 data sample 별로 Loss을 구해서 average을 취한다.

주로 **training** 단계에서 사용된다. 왜냐하면,

.backward()은 scalar 값 $\in \mathbb{R}$ 에 대해서만 적용할 수 있기 때문에 **average된 loss**을 사용하게 된다.

3-7. Section 3 요약

Section Summary

Loss Function

Loss Function L (손실 함수) 의 정의

Neural Network 모델이 예측한 값 \hat{Y} 과 원래 정답 Y 간의 차이 (오차)의 지표

손실 함수의 값이 최소화하도록 모델의 weight을 최적화하면 모델의 정확도가 높아진다!

Section Summary

Loss Function

- “**Regression (회귀)**” = 어떤 연속적인 값을 예측하는 task
- “**Classification (분류)**” = “**discrete**” (불연속적인), “**categorical**” (범주형) 라벨을 예측하는 task

Section Summary

Loss for Regression Task:

$$\text{L2 Loss (MSE Loss)} = (Y_i - \hat{Y}_i)^2$$

\hat{Y}_i predicted value

$$\text{L1 Loss (MAE Loss)} = | Y_i - \hat{Y}_i |$$

Y_i ground truth value

$$\text{Huber Loss} = \frac{1}{2}(\hat{y}_i - y_i)^2$$

if $|\hat{y}_i - y_i| < \delta$

$$\delta \cdot \left(|\hat{y}_i - y_i| - \frac{\delta}{2} \right)$$

otherwise

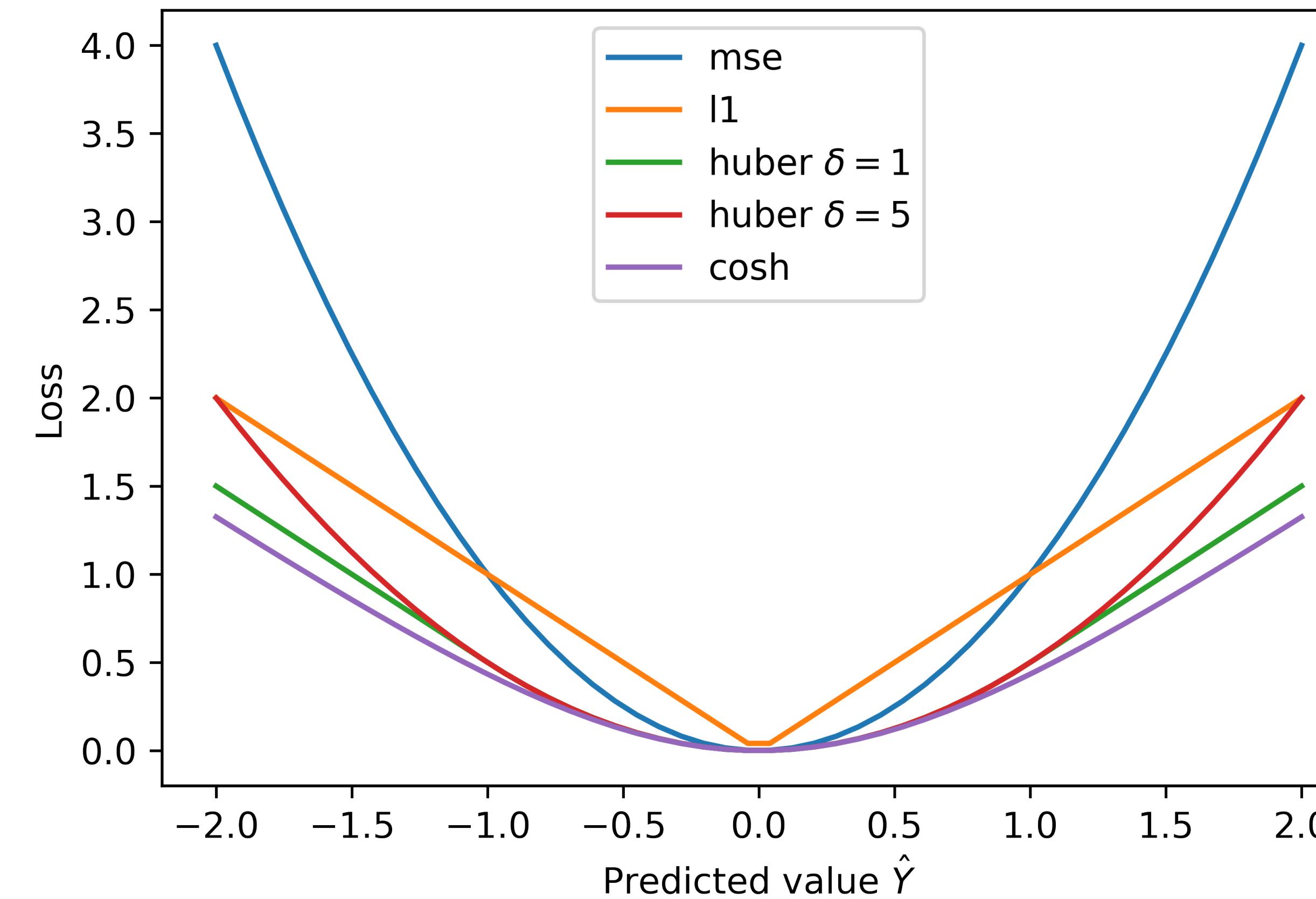
$$\text{Log-Cosh Loss} = \log (\cosh(\hat{y}_i) - y_i))$$

$$\cosh(x) = \frac{e^x + e^{-x}}{2}$$

Loss for Regression task

MSE, L1, Huber Loss, Log-cosh Loss Visualization

Copyright©2023. Acadential. All rights reserved.



Ground Truth ($Y = 0$)

Diagram Drawn using matplotlib

Section Summary

각 Loss의 장단점 및 특징

1. MSE Loss (L2 Loss)

- (a) 장점: Convergence가 더 빠르다.
- (b) b/c 실제값과 차이가 크면 해당 Loss 값은 더욱 크다. (MSE은 Quadratic 하므로)
- (c) 단점: Anomaly (Outlier)에 대해서 더 민감하다.

2. MAE Loss (L1 Loss)

- (a) 장점: MSE와 비교했을때 Anomaly에 대해서 비교적 더 robust하다.
- (b) 단점: 0에서 미분가능하지 않다.

Section Summary

각 Loss의 장단점 및 특징

3. Huber Loss

- (a) 장점: MSE의 faster convergence의 장점과 MAE의 outlier에 대한 robustness 장점을 둘 다 가지고 있다.
- (b) 단점: 한 번만 전구간 미분가능하다.

4. Log Cosh

- (a) Huber Loss와 비슷한 성격을 가졌지만 전구간 두 번 이상 미분가능하다.

Section Summary

Loss for Classification task

- L1 Loss (MAE Loss)

$$\sum_{c=1}^C |Y_{i,c} - \hat{Y}_{i,c}|$$

- Cross Entropy Loss

$$\sum_{c=1}^C -Y_{i,c} \log \hat{Y}_{i,c}$$

- Kullback-Leibler Divergence Loss (KL Divergence)

$$\sum_{c=1}^C Y_{i,c} \cdot \log \left(\frac{Y_{i,c}}{\hat{Y}_{i,c}} \right)$$

Section Summary

Loss for Classification task

- **Cross Entropy Loss, KL Divergence Loss**
 - (a) 장점: L1 Loss (MAE Loss)에 비해서 학습 속도가 더 빠르다
 - (b) 단점: Outlier 및 Noise에 대해서 더 취약, 민감하다.
- **L1 Loss (MAE Loss)**
 - (c) 장점: Outlier 및 Noise에 대해서 더 robust하다.
 - (d) 단점: Cross Entropy Loss에 비해서 학습 속도가 느리다.

Section Summary

Loss Function: Regression

Copyright©2023. Acadential. All rights reserved.

Loss (Regression)	장점	단점	공식
MSE	Convergence (수렴)이 빠르다.	Outlier에 대해서 민감하다.	$(Y_i - \hat{Y}_i)^2$
MAE	Outlier에 대해서 MSE보다 비교적 더 robust하다.	0에서 미분가능하지 않다. Convergence (수렴)이 느리다	$ Y_i - \hat{Y}_i $
Huber Loss	MSE의 빠른 수렴 속도 MAE의 Outlier에 대한 robustness	전구간에 대해 1번만 미분 가능하다.	$\begin{cases} \frac{1}{2}(\hat{y}_i - y_i)^2 & \text{if } \hat{y}_i - y_i < \delta \\ \delta \cdot \left(\hat{y}_i - y_i - \frac{\delta}{2} \right) & \text{otherwise} \end{cases}$
Log Cosh	MSE의 빠른 수렴 속도 MAE의 Outlier에 대한 robustness		$\log (\cosh(\hat{y}_i) - y_i)$

Section Summary

Loss Function: Classification

Loss (Classification)	장점	단점	공식
Cross Entropy Loss	L1 Loss에 비해서 학습 속도가 더 빠르다	Outlier 및 Noise에 더 취약, 민감하다.	$\sum_{c=1}^C -Y_{i,c} \log(\hat{Y}_{i,c})$
L1 Loss (MAE Loss)	Outlier 및 Noise에 대해서 더 Robust하다.	Cross Entropy Loss에 비해서 학습 속도가 느리다.	$\sum_{c=1}^C Y_{i,c} - \hat{Y}_{i,c} $
Kullback-Leibler Divergence Loss (KL Divergence)	Cross Entropy Loss의 장점과 유사.	Symmetric하지 않다. $KL(Y, \hat{Y}) \neq KL(\hat{Y}, Y)$	$\sum_{c=1}^C Y_{i,c} \cdot \log\left(\frac{Y_{i,c}}{\hat{Y}_{i,c}}\right)$