

# Information Theory

Additional mini-project in deeplearning math

## Entropy Loss (엔트로피 손실)

소프트웨어 끈대 강의

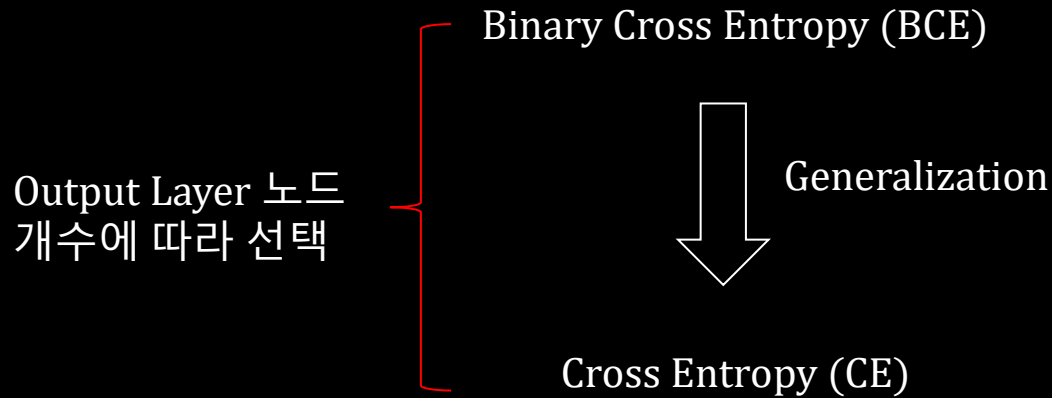
노기섭 교수

(kafa46@cju.ac.kr)

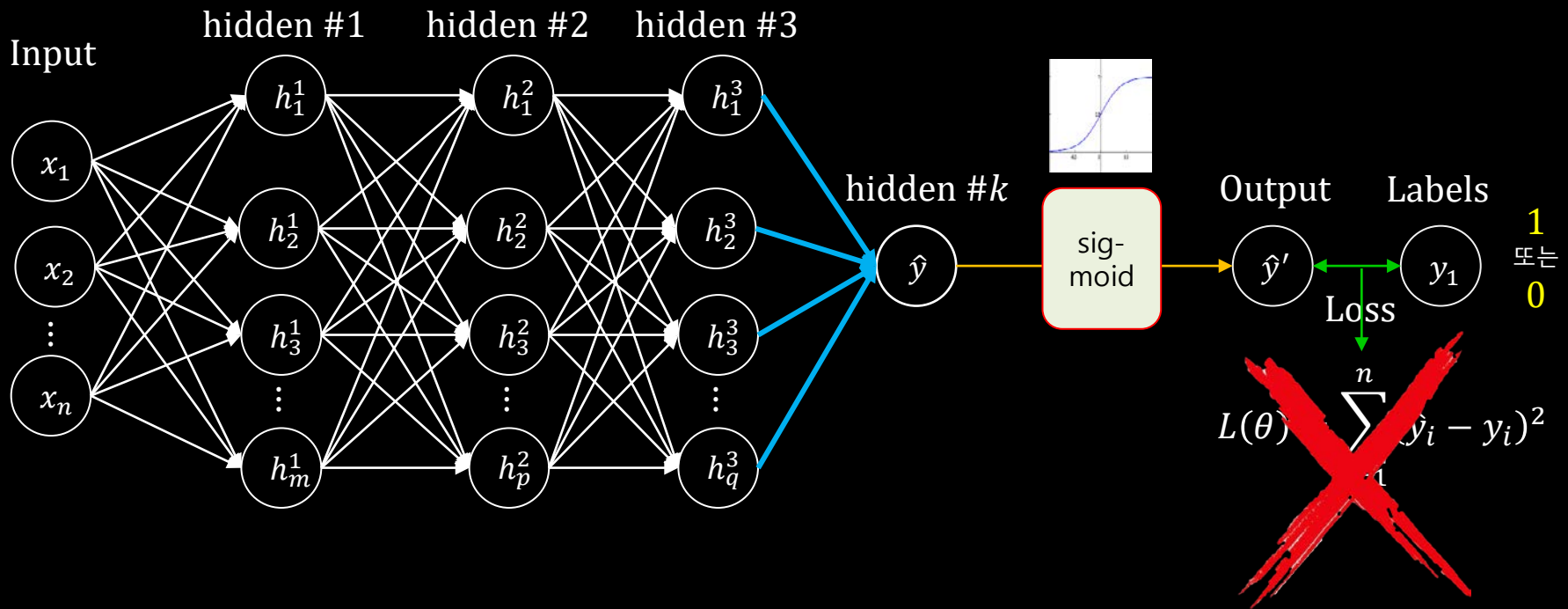
# Course Overview

Topic	Contents
01. Orientation 오리엔테이션	Motivations & Course introduction 동기부여, 과정 소개
02. Information 정보	What is the information? Concept & definition 정보란 무엇인가? 개념과 정의
03. Information Entropy 정보 엔트로피	Concepts, notation, and operations on information entropy 정보 엔트로피의 개념, 표기, 연산
04. Entropy in Deep learning 딥러닝에서의 엔트로피	How to apply the information entropy into Deep learning? 어떻게 정보 엔트로피를 딥러닝에 적용하는가?
05. Entropy Loss 엔트로피 손실	Loss function using entropy, BCE, and cross entropy 엔트로피를 이용한 손실 함수, BCE, 크로스 엔트로피
06. KL Divergence KL 발산	Concept & definition of KL divergence KL 발산의 개념과 정의
07. Summary & Closing 요약 및 마무리	Summary & closing on this project, 'Information Theory' 정보 이론 요약 및 마무리

# 딥러닝에서 흔히 사용되는 entropy



# Binary Classification (이진 분류)



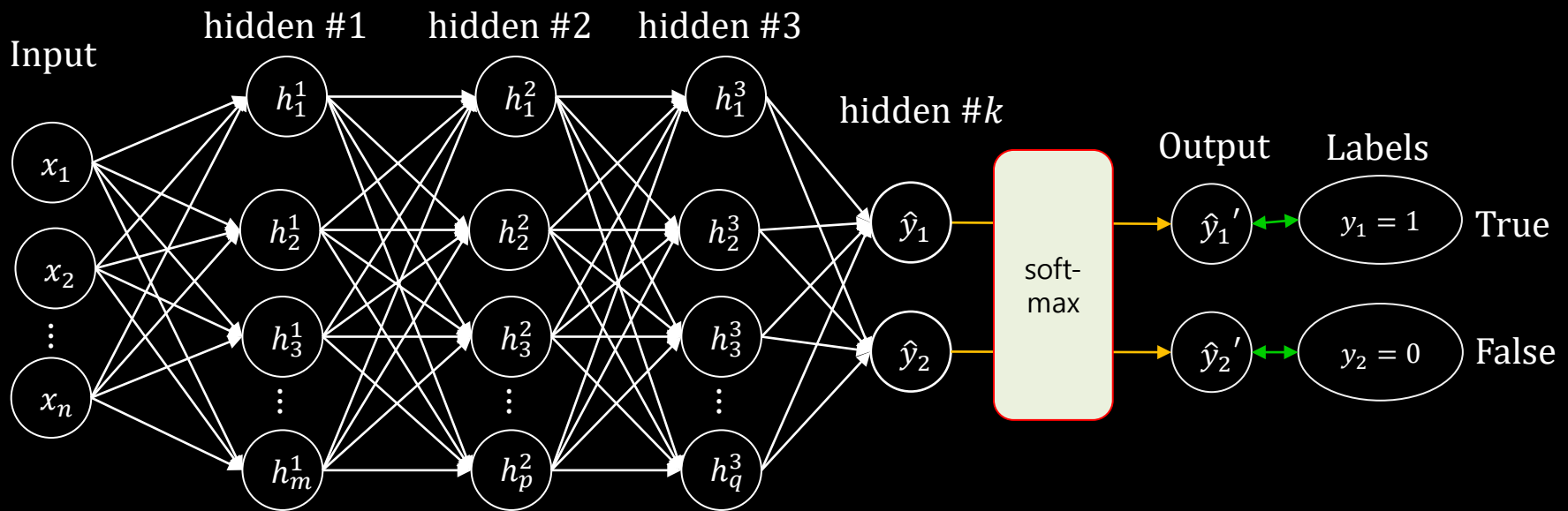
예측 전략

- sigmoid를 통과한 값이 0.5 이상이면 **True** 로 예측
- sigmoid를 통과한 값이 0.5 미만이면 **False** 로 예측

얼마나 틀린 거야?

# Binary Classification (이진 분류)

참고: 이진 분류를 softmax로 구현해도 문제 없음  
→ 이런 상황에서는 cross entropy loss를 사용

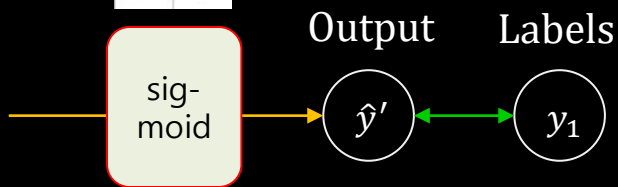
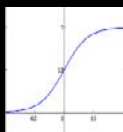


# Binary Cross Entropy

예측 전략

- sigmoid를 통과한 값이 0.5 이상이면 **True (1)** 로 예측
- sigmoid를 통과한 값이 0.5 미만이면 **False (0)**로 예측

얼마나 틀린 거야?



**Solution:** 둘 다 측정하면 되지 않을까?

정답(Label) 값이 True (1) 이었다고 가정해 봅시다.

Sigmoid를 통과한 값이  $\hat{y}' = 0.6$  이라고 가정

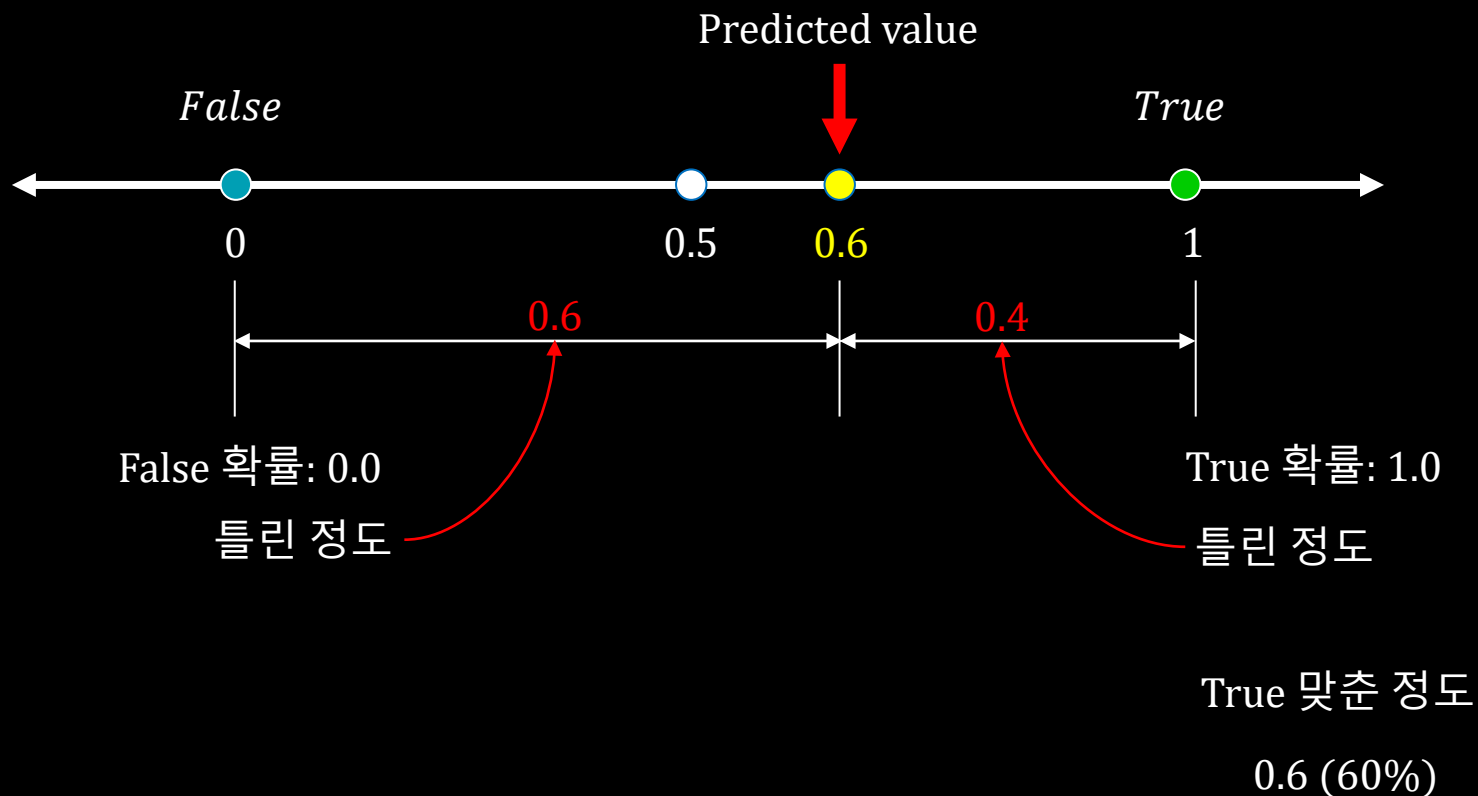
0.5 이상이므로 True 라고 판단할 것임 (맞는 답)

맞춘 양으로 보면, 0.6 만큼 맞췄음

틀린 양으로 보면,  $1 - 0.6 = 0.4$  만큼 틀렸음

# 직관적으로 살펴보기

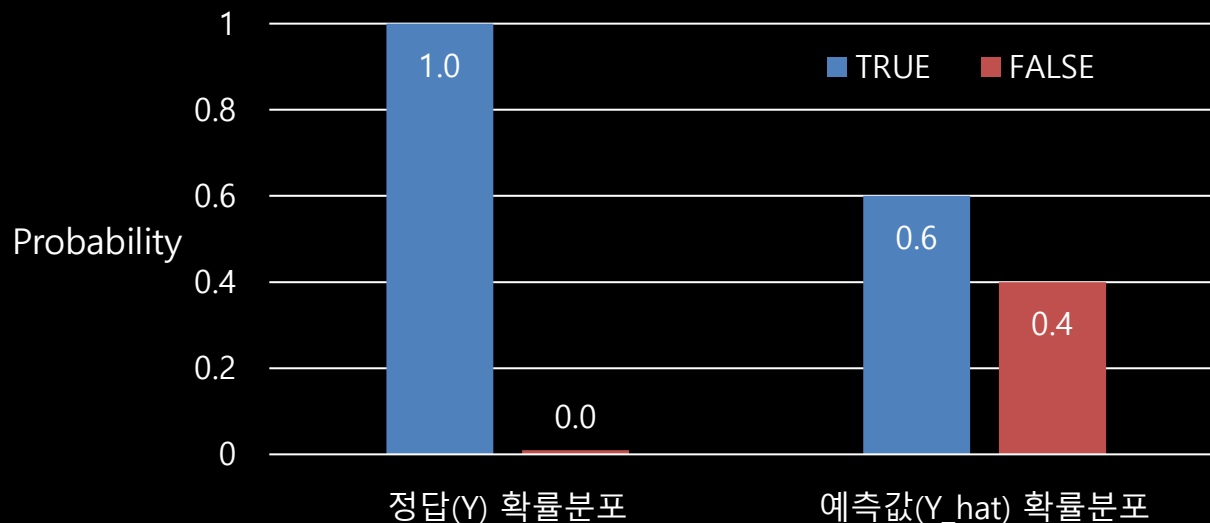
첫 번째 문제에 대한 답 ⇨ 시각적으로 살펴보기



# 엔트로피 관점으로 이해하기

헛갈리는 포인트: 확률 분포가 2개라는 점....  $\pi\pi$

$$\text{Entropy: } H(X) = E(I(X)) = - \sum_i p_i \times \log p_i$$



$$E(Y) = -(0 \times \log 0) - (1 \times \log 1) = 0$$

$$\begin{aligned} E(\hat{Y}) &= -(0.6 \times \log 0.6) - (0.4 \times \log 0.4) \\ &= (0.6 \times 0.737) + (0.4 \times 1.322) \\ &= 0.442 + 0.529 = 0.971 \end{aligned}$$

Entropy를 이용해 얼마나 틀렸는지 표현할 수 있을까?



# Concept of Cross Entropy

$$\text{Entropy: } H(X) = E(I(X)) = - \sum_i P(X = i) \times \log P(X = i) = - \sum_i p_i \times \log p_i$$

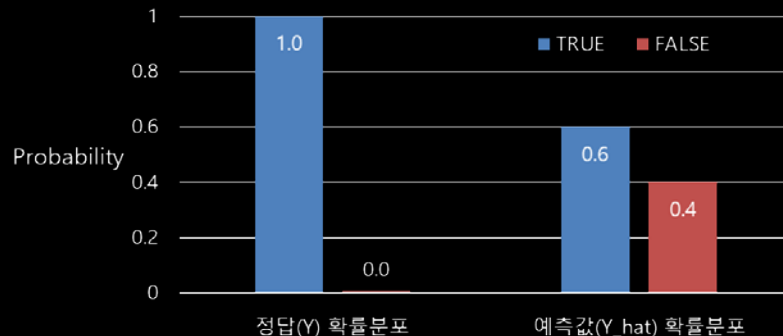
$$\text{Cross Entropy: } H(X, Y) = E_X(I(Y)) = - \sum_{i \in X} P(X = i) \times \log P(Y = i)$$

간단히 쓰면 다음과 같습니다 ^^

$$H(p, q) = - \sum_{x \in \mathcal{X}} p(x) \times \log q(x)$$

, where  $p$  and  $q$  are probability distribution.

# Binary Cross Entropy



$$\begin{aligned}
 E(\hat{Y}) &= -(0.6 \times \log 0.6) - (0.4 \times \log 0.4) \\
 &= (0.6 \times 0.737) + (0.4 \times 1.322) \\
 &= 0.442 + 0.529 = 0.971
 \end{aligned}$$

$$H(Y, \hat{Y}) = - \sum_{x \in \mathcal{X}} P(Y = x) \times \log P(\hat{Y} = x)$$

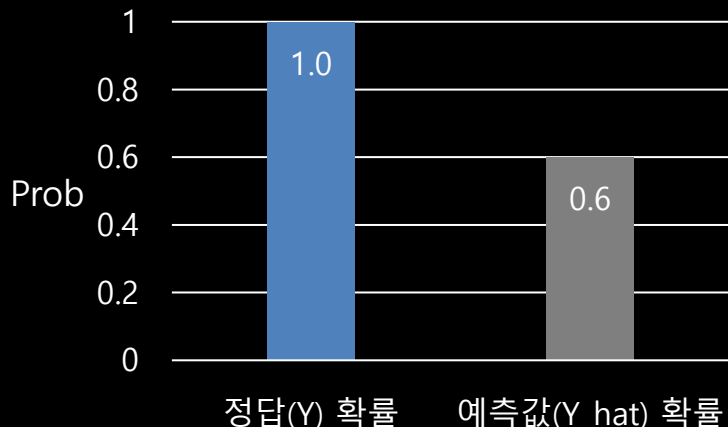
$$= -(P(Y = \text{True}) \times \log P(\hat{Y} = \text{True}) + (1 - P(Y = \text{True})) \times \log (1 - P(\hat{Y} = \text{True})))$$

$$= -(1 \times \log 0.6 + (1 - 1) \times \log(1 - 0.6))$$

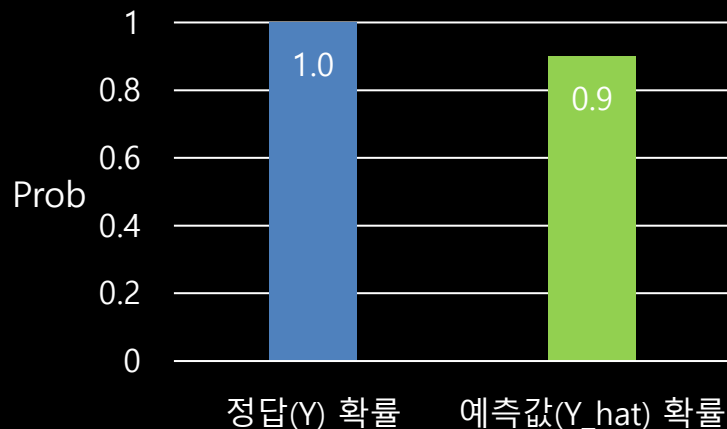
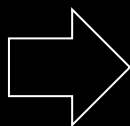
$$= -(1 \times -0.737) = 0.737$$

# 정답에서 가까워질수록, 멀어질수록 → entropy 변화량 관찰

$$H(Y, \hat{Y}) = 0.737$$



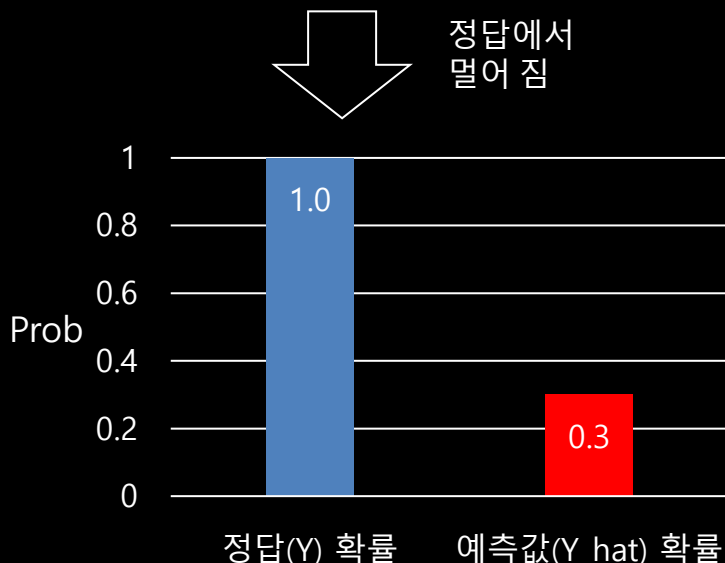
정답에  
가까워 짐



$$H(Y, \hat{Y})$$

$$\begin{aligned} &= -(1 \times \log 0.9 + (1 - 1) \times \log(1 - 0.9)) \\ &= -(1 \times 0.152) = 0.152 \end{aligned}$$

Cross entropy 가 감소한다



$$H(Y, \hat{Y})$$

$$\begin{aligned} &= -(1 \times \log 0.3 + (1 - 1) \times \log(1 - 0.3)) \\ &= -(1 \times -1.737) = 1.737 \end{aligned}$$

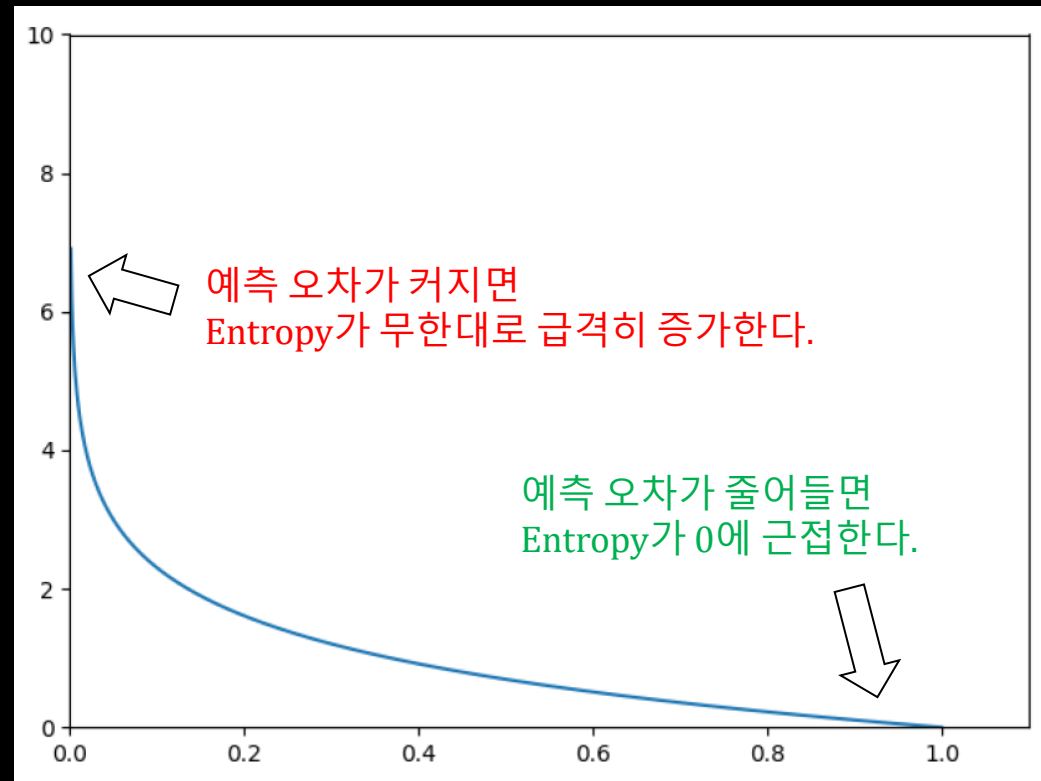
Cross entropy 가 증가한다

# Binary Cross Entropy의 일반적 특징

Binary Cross Entropy를 Loss 값으로 설정하고 학습시킬 수 있다!

$$H(Y, \hat{Y}) \in [0, \infty]$$

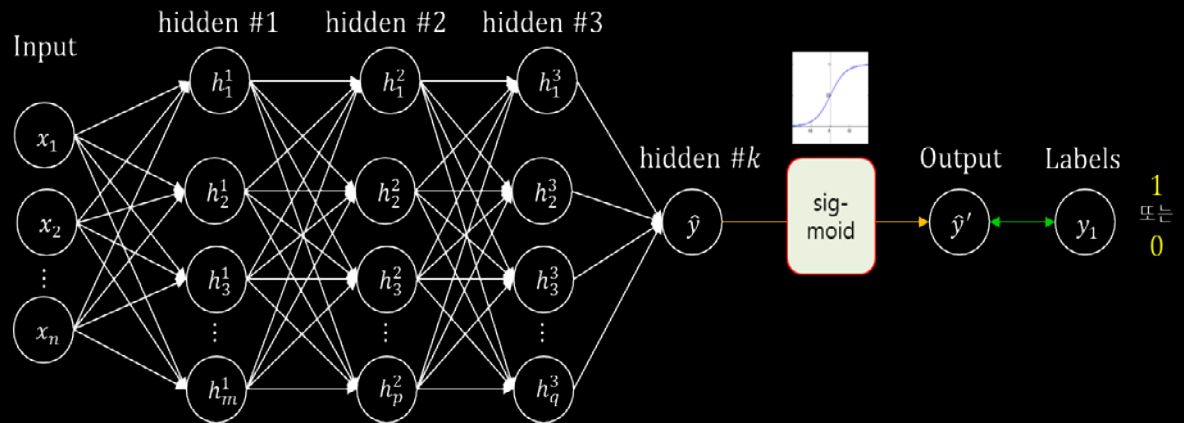
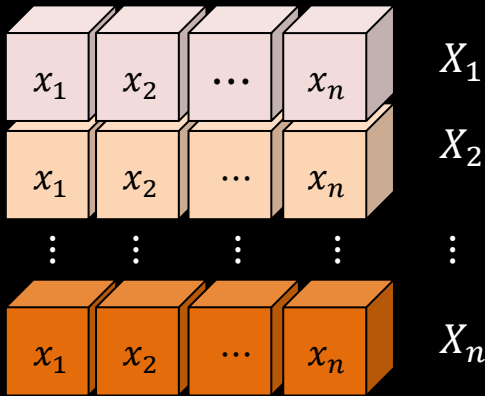
(Cross Entropy)



$$\hat{Y} \in [0, 1]$$

# Binary Cross Entropy를 Deeplearning에 적용하는 방법

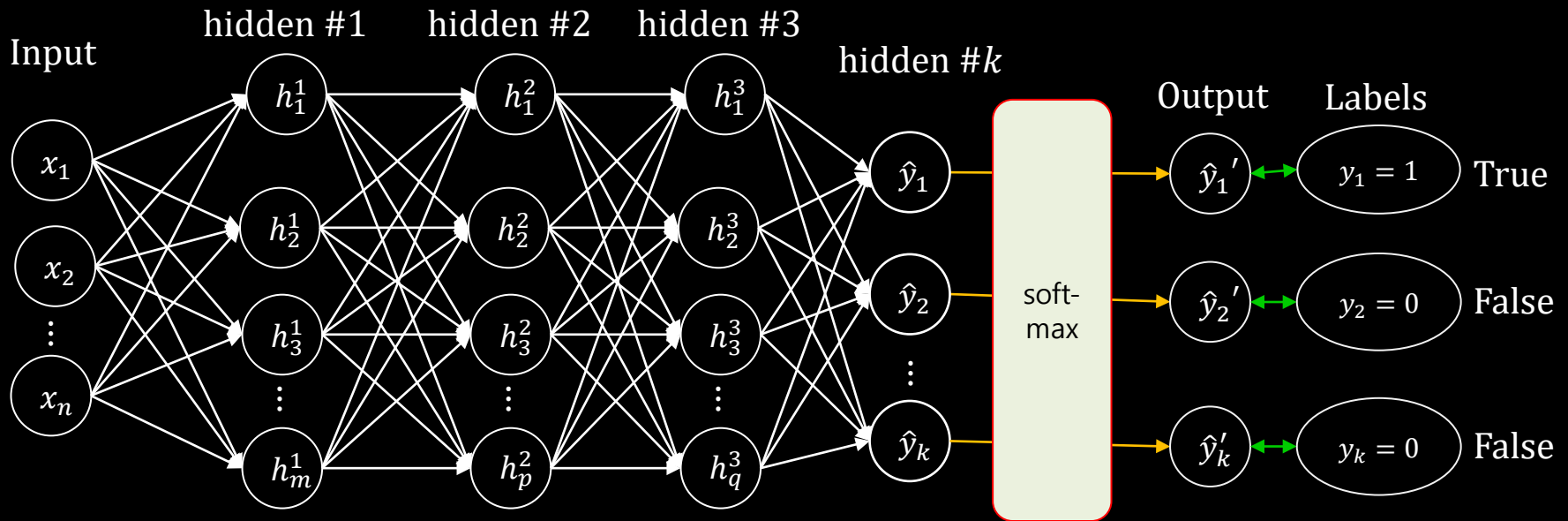
입력은 Mini batch 형태로 들어올 것임 (일반적으로 3D- / 4D-tensor 형태)



$$BCE = -\frac{1}{n} \sum_{i=1}^n [y_i \times \log \hat{y}_i + (1 - y_i) \times \log(1 - \hat{y}_i)]$$

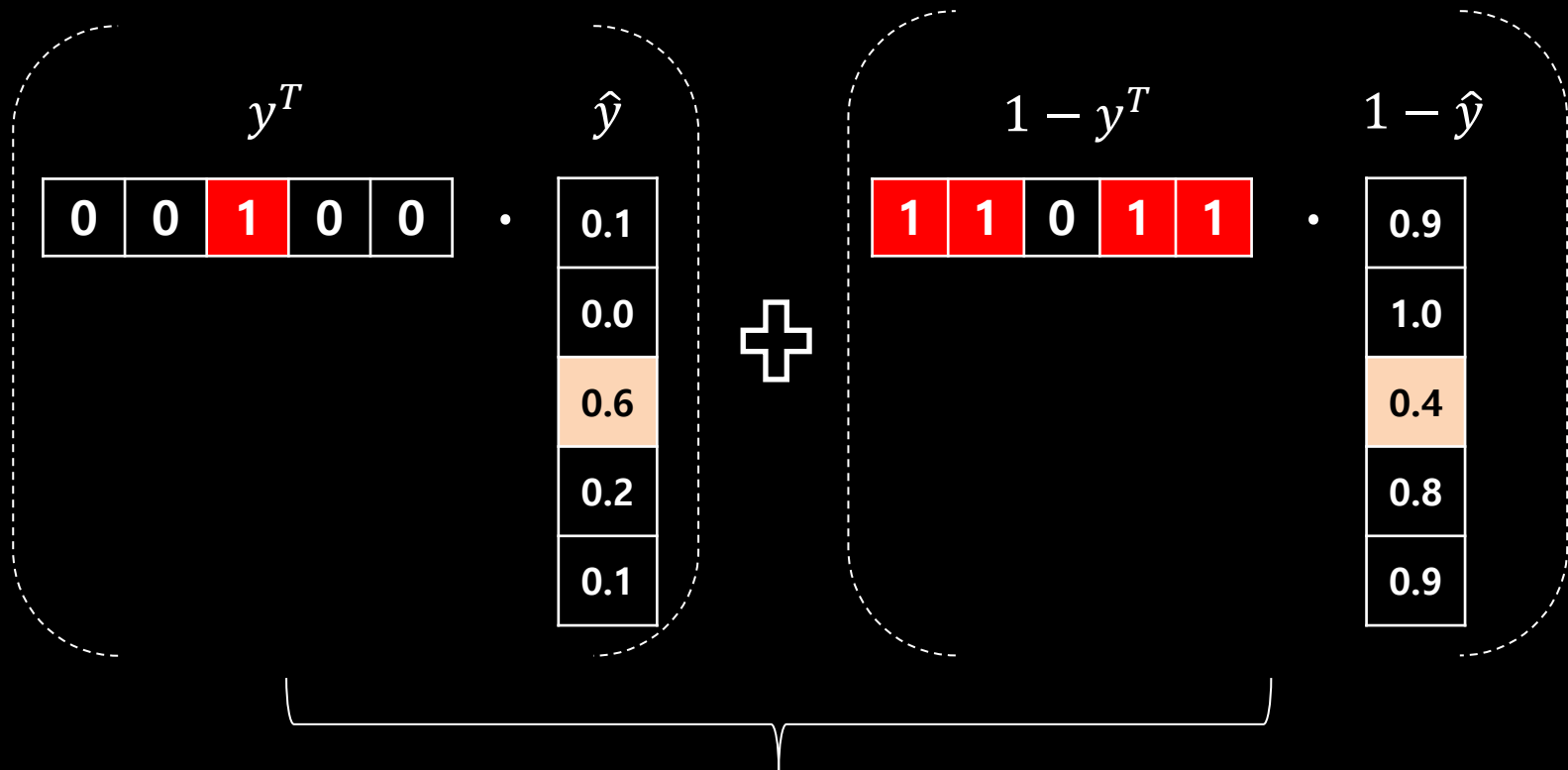
# BCE Loss의 일반화 버전 $\rightarrow$ Cross Entropy

다중 분류 네트워크 (Logistic Regression)  $\leftrightarrow$  (Ref. Linear Regression)



하나만 1, 나머지는 0  
(One-hot Encoding)

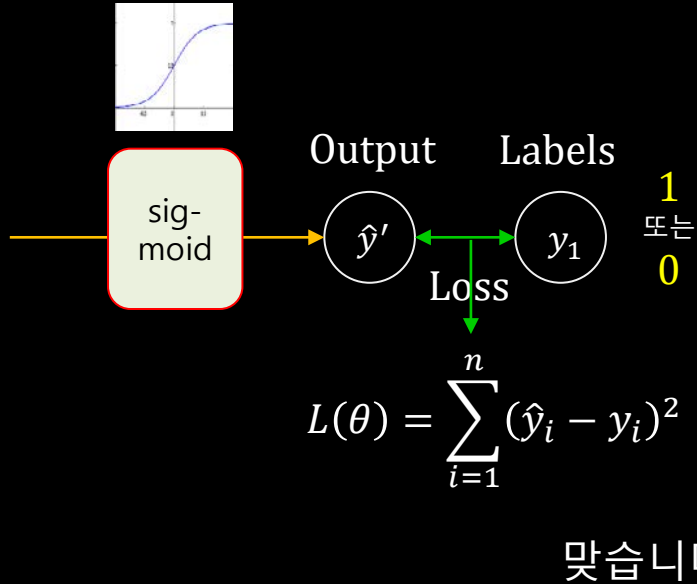
# Cross Entropy - 병렬 처리



$$CE = -\frac{1}{n} \sum_{i=1}^n [y_i^T \times \log \hat{y}_i + (1 - y_i)^T \times \log(1 - \hat{y}_i)]$$

Note: 1개 입력에 대한 Cross Entropy Loss  
→ Batch 입력일 경우 평균내서 사용

# Cross Entropy를 사용하는 이유?



교수님~~

제곱오차(Squared Error)를 사용하는 것이  
더 직관적이고 간단할 것 같은데요?

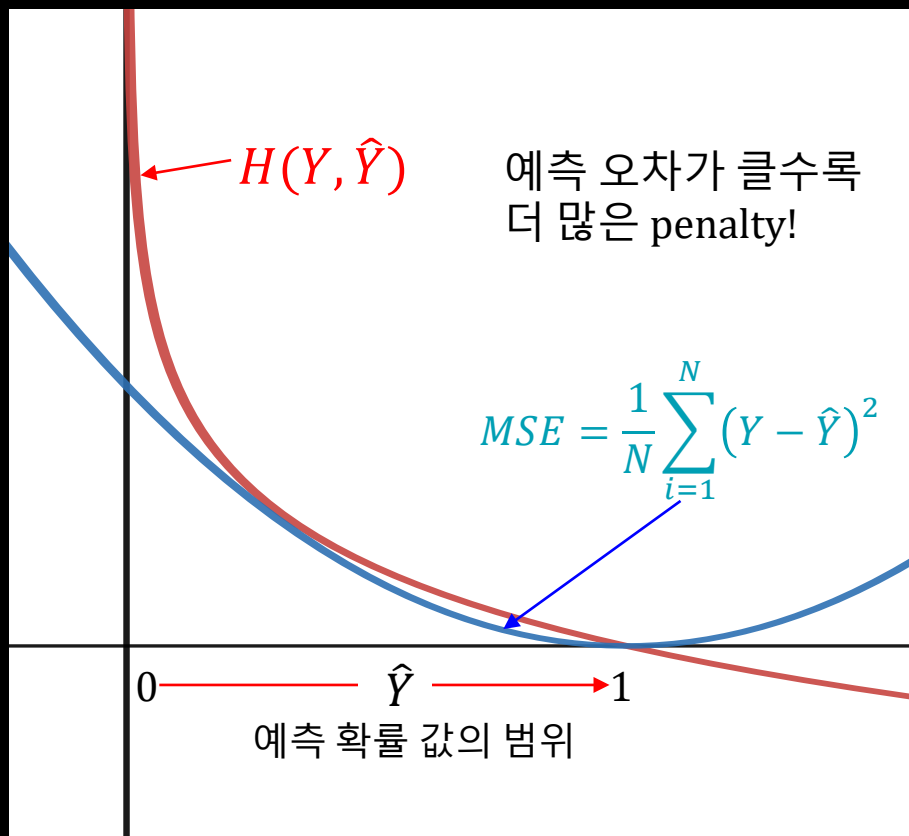
하지만, 딥러닝 신경망 학습(optimization) 관점에서  
큰 차이가 있습니다.

Cross Entropy를 사용할 경우  
학습을 효율적이고, 빠르게 진행할 수 있습니다.

Regression Task에서는 어쩔 수 없이 MSE (Mean Squared Error)를  
사용하지만, Classification Task에서는 Entropy Loss를 사용하는 것이 좋습니다. ^^

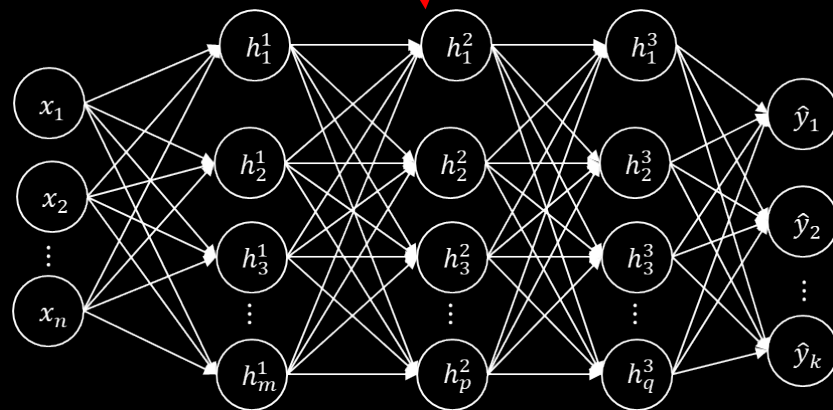


# Cross Entropy 장점 - 직관적 이해



Back-prop 과정에서  
더 큰 업데이트 수행

$$\theta \leftarrow \theta - \eta \times \frac{L(\theta)}{\partial \theta}$$





수고하셨습니다 ..^^..