Information Theory

Additional mini-project in deeplearning math

Entropy Loss
(엔트로피 손실)

소프트웨어 꼰대 강의

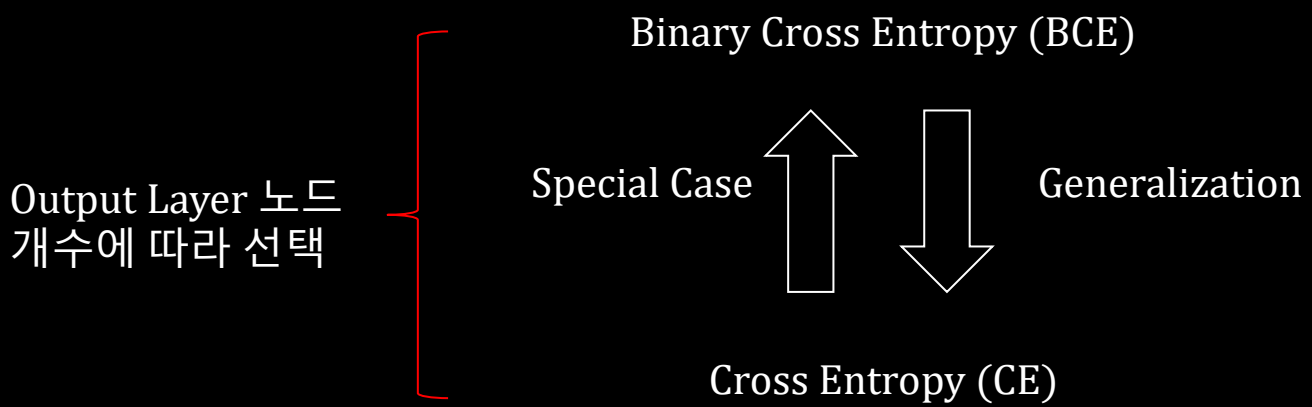
노기섭 교수

(kafa46@cju.ac.kr)

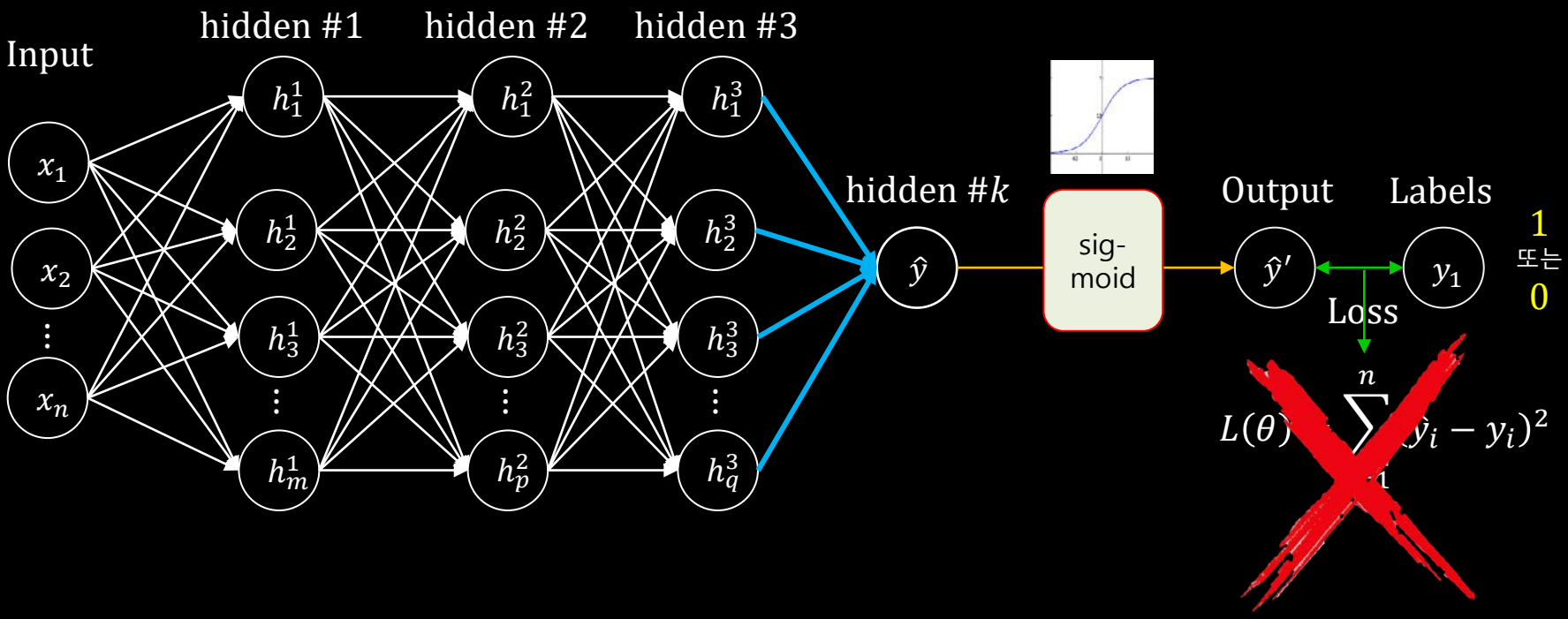
Course Overview

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 Deeplearning 딥러닝에서의 엔트로피	How to apply the information entropy into Deeplearning? 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)

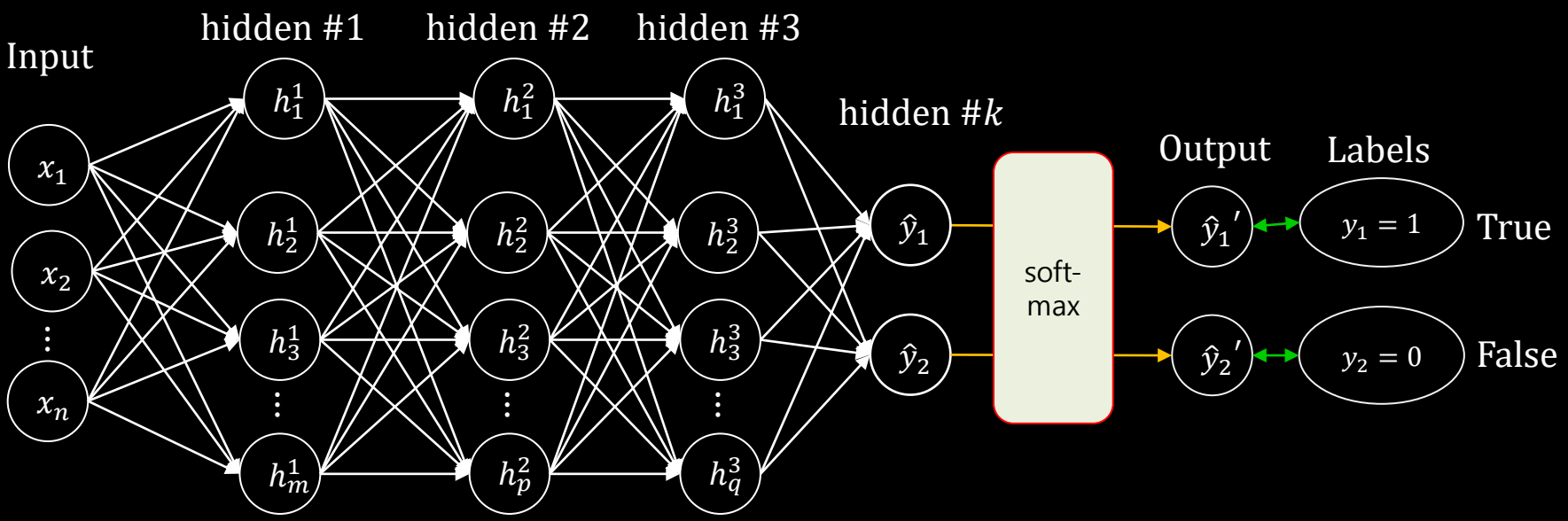


예측 전략
- sigmoid를 통과한 값이 0.5 이상이면 True 로 예측
- sigmoid를 통과한 값이 0.5 미만이면 False 로 예측

얼마나 틀린 거야?

Binary Classification (이진 분류)

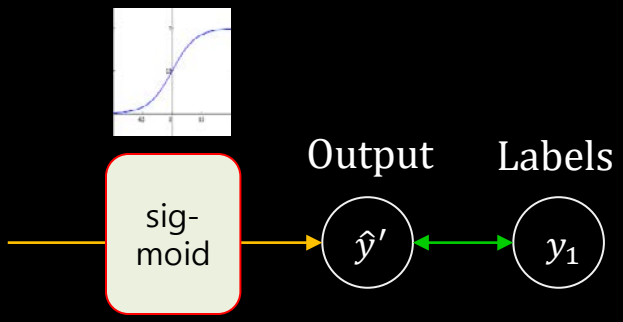
참고: 이진 분류를 softmax로 구현해도 문제 없음
→ 이런 상황에서는 cross entropy loss를 사용



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 봅니다.

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

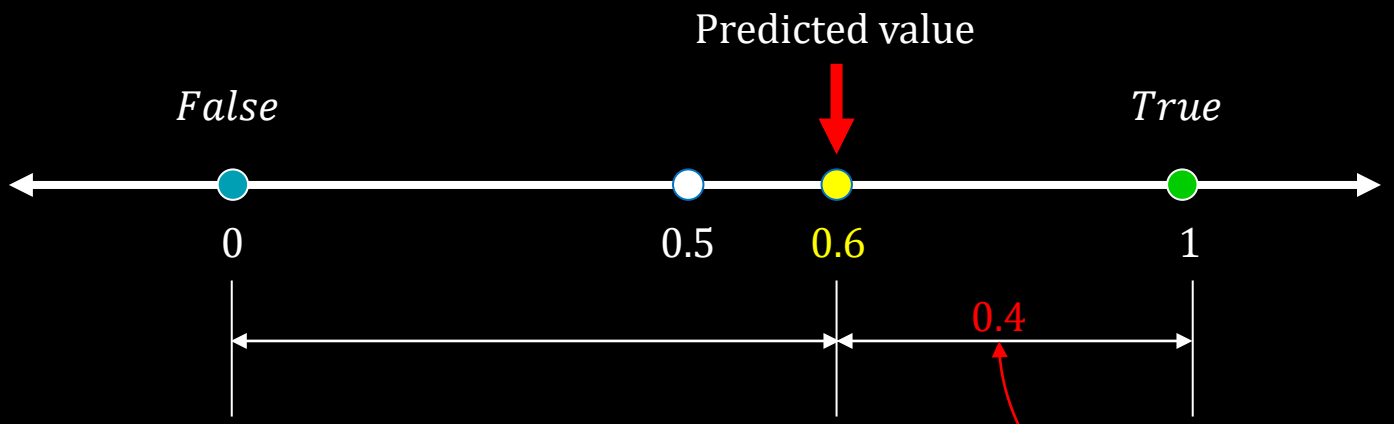
0.5 이상이므로 True 라고 판단할 것임 (맞는 답)

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



True 확률: 1.0

틀린 정도

True 맞춘 정도

0.6 (60%)

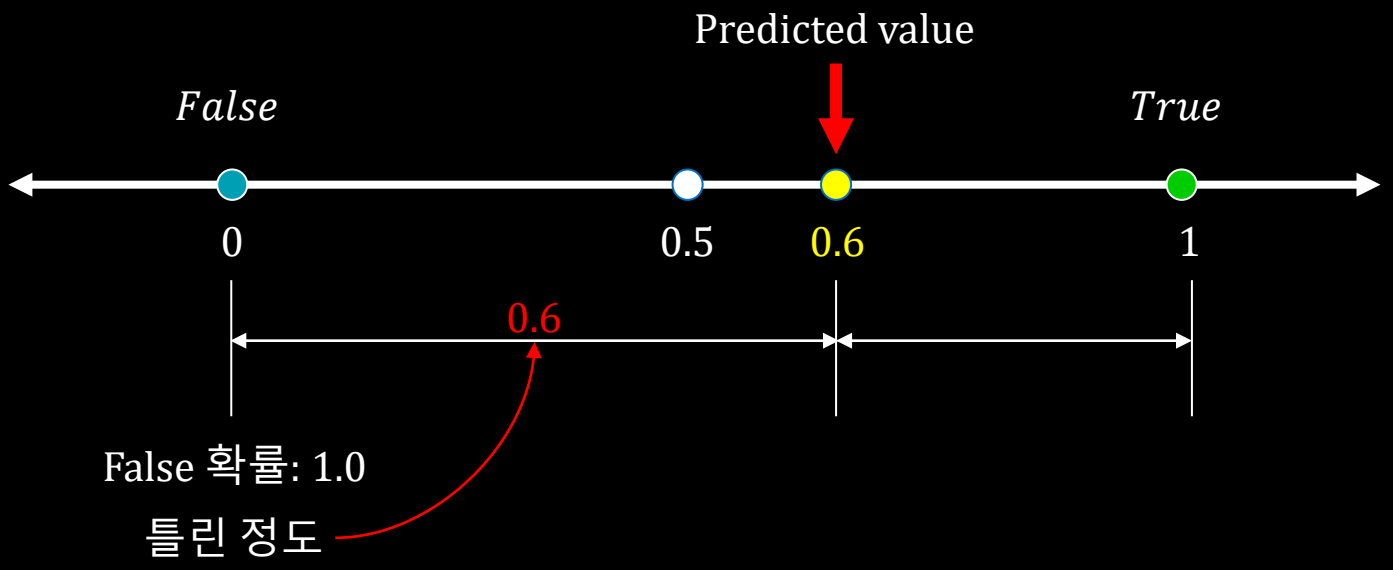
이진 분류의 반대 상황도 같이 생각하기

이번에는 정답(Label) 값이 False 이었다고 가정해 봅니다.

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

0.5 이상이므로 True 라고 판단할 것임 (틀린 답)

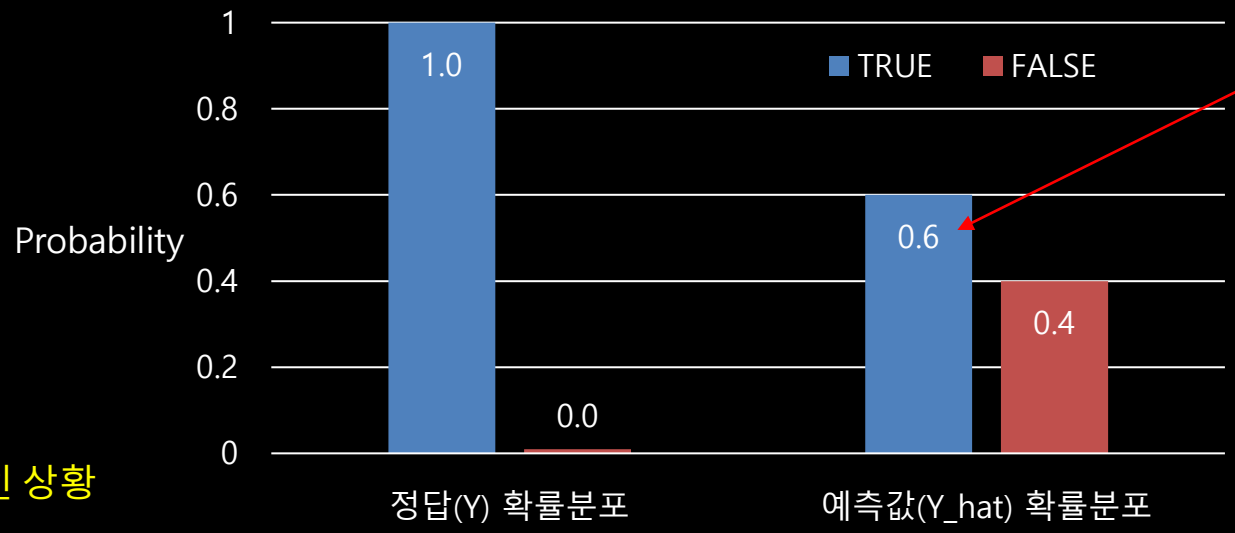
맞춘 양으로 보면, $1 - 0.6 = 0.4$ 만큼 맞췄음(틀린 양으로 보면, 0.6 만큼 틀렸음)



엔트로피 관점으로 이해하기 (True 가 정답인 상황)

헷갈리는 포인트: 확률 분포가 2개라는 점.... ㅠㅠ

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



True 가 정답인 상황

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

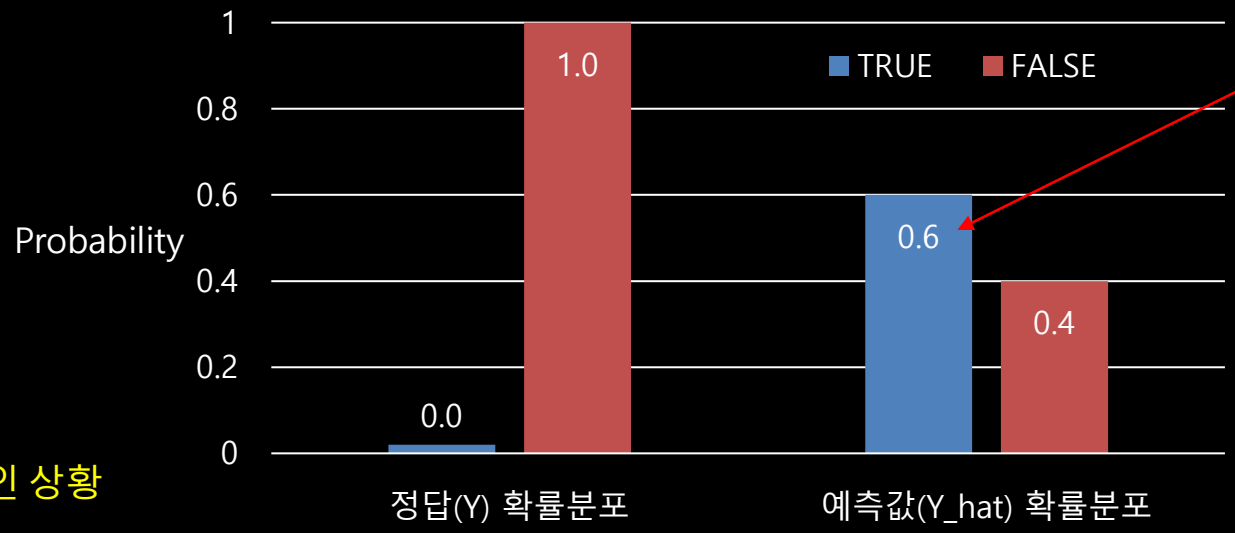
$$E(\hat{Y}) = -(0.6 \times \log 0.6) - (0.4 \times \log 0.4) = (0.6 \times 0.737) + (0.4 \times 1.322) = 0.442 + 0.529 = 0.971$$

Entropy를 이용해 얼마나 틀렸는지 표현할 수 없을까?

엔트로피 관점으로 이해하기 (False 가 정답인 상황)

헷갈리는 포인트: 여전히 확률 분포가 2개라는 점.... ㅠㅠ (True 가 정답인 상황과 똑같다)

$$Entropy:\ H(X) = E\big(I(X)\big) = -\sum_i p_i \times \log\, p_i$$



False 가 정답인 상황

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

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

False가 정답일 경우도 동일한 상황 발생
(정답이 True, False 관계없이 항상 같다??? → 문제 있음)

Concept of Cross Entropy

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

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

간단히 쓰면 다음과 같습니다 ^^

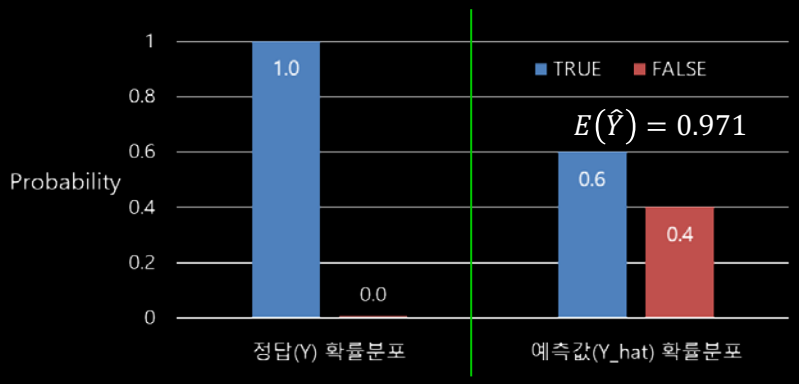
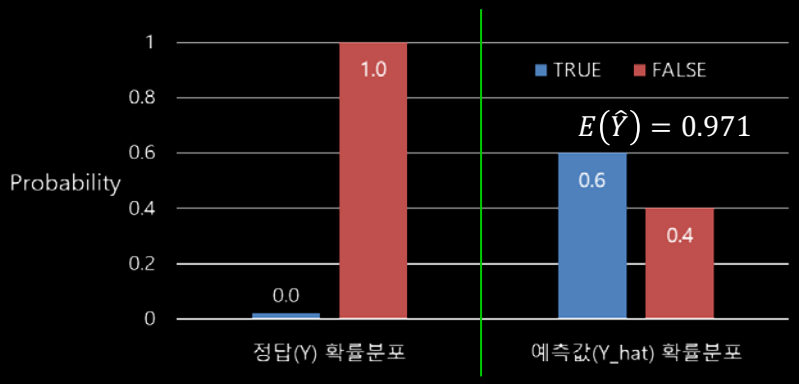
$$H(p,q) = -\sum_{x \in X} p(x) \times \log\ q(x)$$

$$,where\ p\ and\ q\ are\ probility\ distribution.$$

Binary Cross Entropy

Cross Entropy 공식 → $H(Y,\hat{Y}) = -\sum_{y \in Y} P(Y=y) \times \log\ P(\hat{Y}=y)$

이진 분류의 경우 → $H(Y,\hat{Y}) = -P(Y=True) \times \log P(\hat{Y}=True) - (P(Y=False)) \times \log\ (P(\hat{Y}=False)$

정답이 True 인 경우 확률 분포	정답이 False 인 경우 확률 분포
	
$H(Y,\hat{Y}) = -1 \times \log 0.6 - 0 \times \log(0.4))$	$H(Y,\hat{Y}) = -0 \times \log 0.6 - 1 \times \log(0.4))$
$= -(1 \times -0.737) = 0.737$	$= -(1 \times -1.322) = 1.322$

Binary Classification (이진 분류) ⇨ 출력이 하나일지라도 2개의 Case를 모두 고려해야 합니다 ^^.

Formulation with One Output

교수님 ~~~

출력이 하나인데 2가지 경우를
항상 따로 생각해야 하나요?

한번에 처리할 수 있는 방법은 없나요???

당연히 있습니다. ~~ 아주 간단해요 ^^.

이미지 저작자: FLATICON
이미지 출처:
https://www.flaticon.com/kr/free-icon/professor_1915998

$$Cross\ Entropy:\ H(Y,\hat{Y}) = E_Y\left(I(\hat{Y})\right) = -\sum_{i \in X} P(Y=i) \times \log\ P(\hat{Y}=i)$$

Binary classification → 경우의 수가 2개 뿐!!

$$-\sum_{i \in \{T,F\}} P(Y=i) \times \log\ P(\hat{Y}=i)$$

$$BCE = -[p \log p + (1-p)\log(1-p)]$$

Sigmoid 출력이 성공 확률 p 이므로

그냥... 생각없이 모두 전개한다.

$$= -\left[P(True) \times \log\ P\left(\hat{Y}=True\right) + P(False) \times \log\ P\left(\hat{Y}=False\right)\right]$$

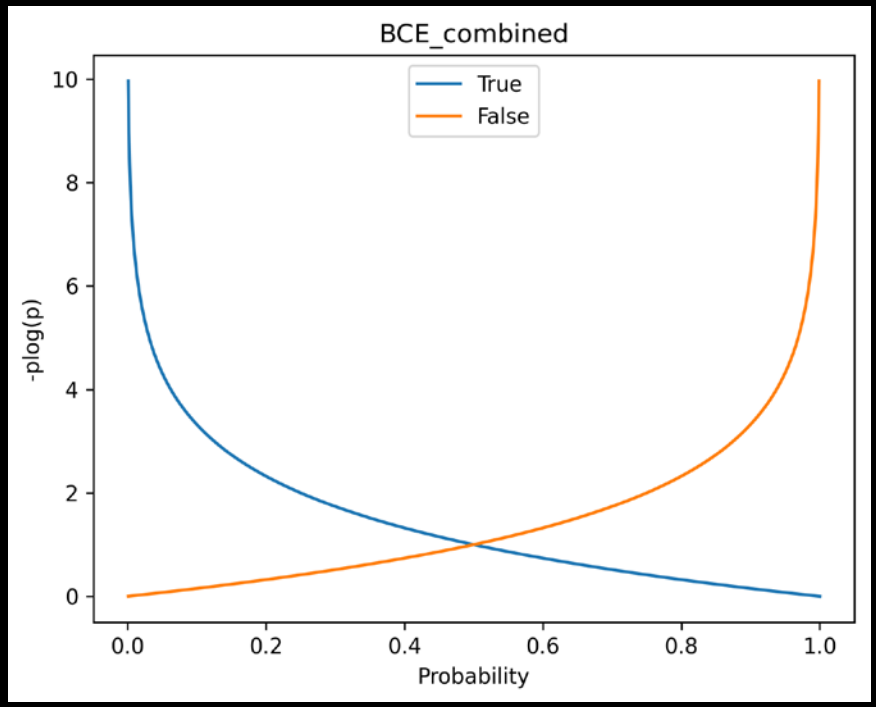
BCE 직관적으로 해석하기

$$BCE = -[\, p \log p + (1-p)\log(1-p) \,]$$

정답이 True 인 경우에 살아남는 항(Term)
(False와 관련된 term은 0이 되어 삭제됨)

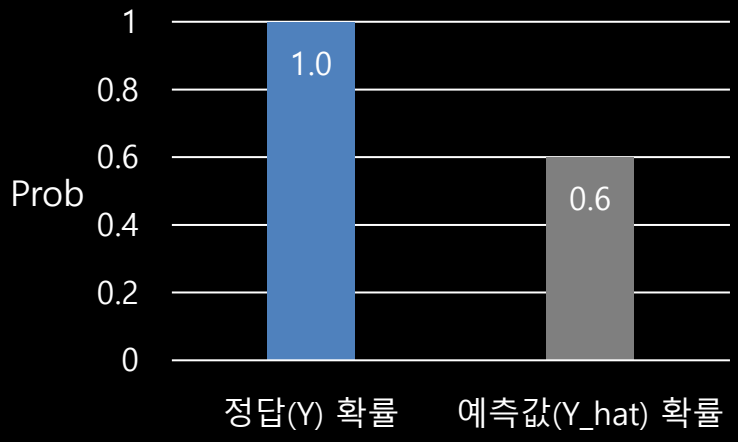
정답이 False 인 경우에 살아남는 항(Term)
(True와 관련된 term은 0이 되어 삭제됨)

최종 결과는
요런 형태를
갖게 됩니다 ^^

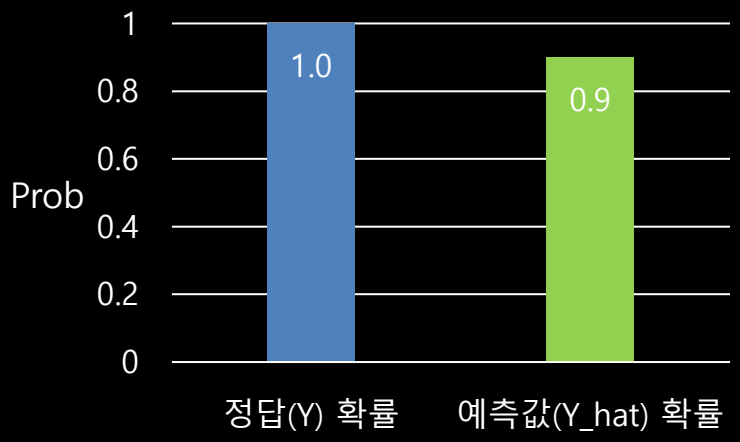


정답에서 가까워질수록, 멀어질수록 → entropy 변화량 관찰

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



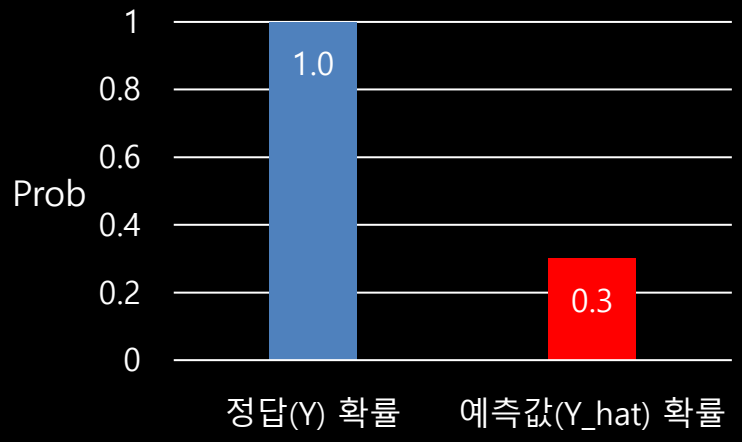
정답에 가까워 짐



$$H(Y, \hat{Y})$$
$$= -(1 \times \log 0.9 + (1-1) \times \log(1-0.9)))$$
$$= -(1 \times 0.152) = 0.152$$

Cross entropy 가 감소한다

정답에서 멀어 짐



$$H(Y, \hat{Y})$$
$$= -(1 \times \log 0.3 + (1-1) \times \log(1-0.3)))$$
$$= -(1 \times -1.737) = 1.737$$

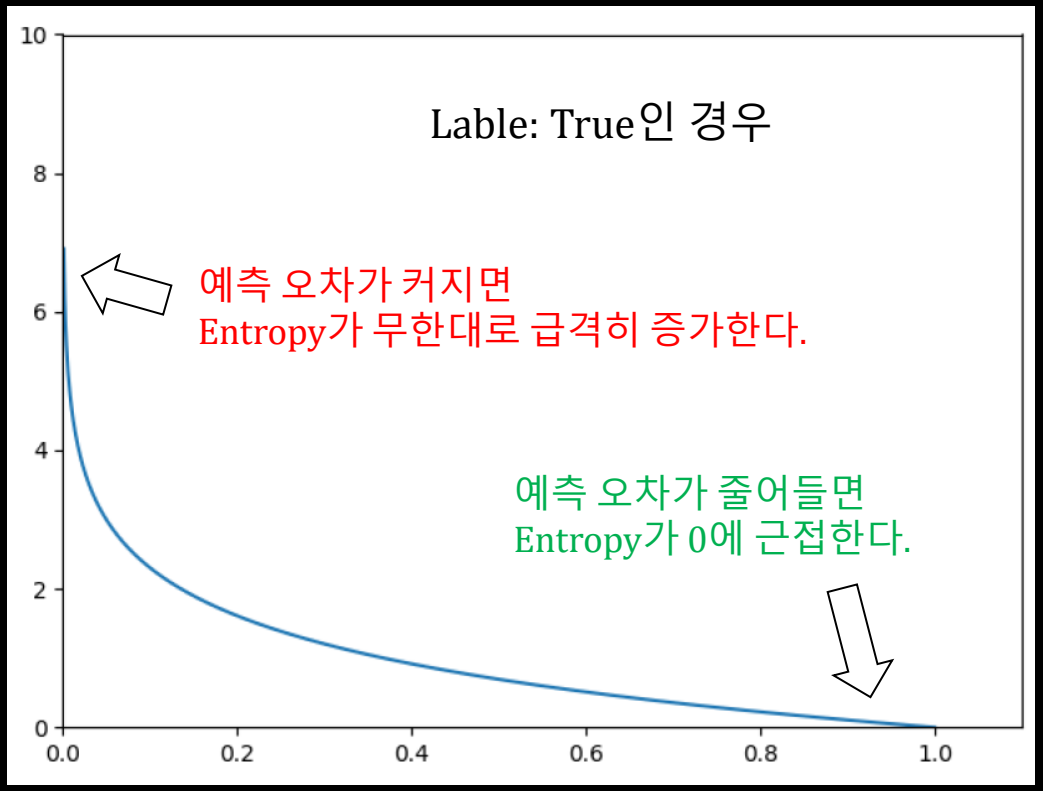
Cross entropy 가 증가한다

Binary Cross Entropy의 일반적 특징

Binary Cross Entropy를 Loss 값으로 설정하고 학습시킬 수 있다!

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

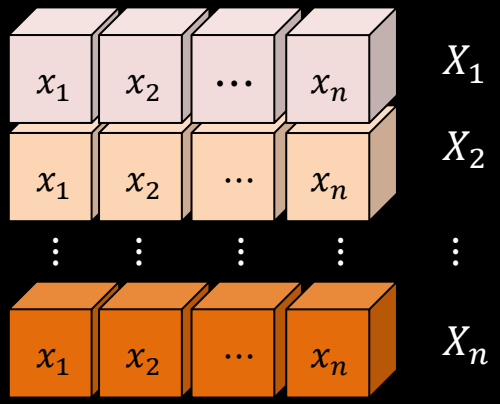
$(Cross\ Entropy)$

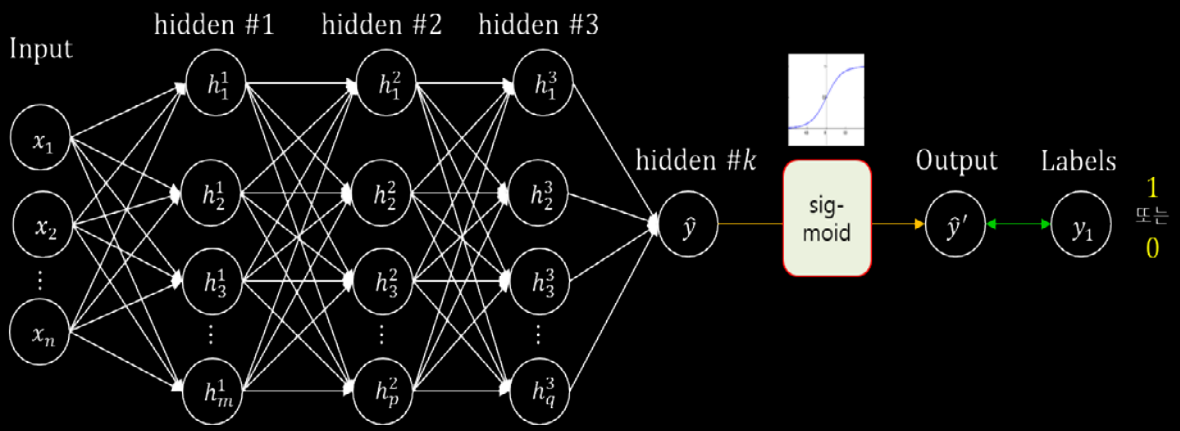


$$\hat{Y} \in [0, 1]$$

Binary Cross Entropy를 Deeplearning에 적용하는 방법

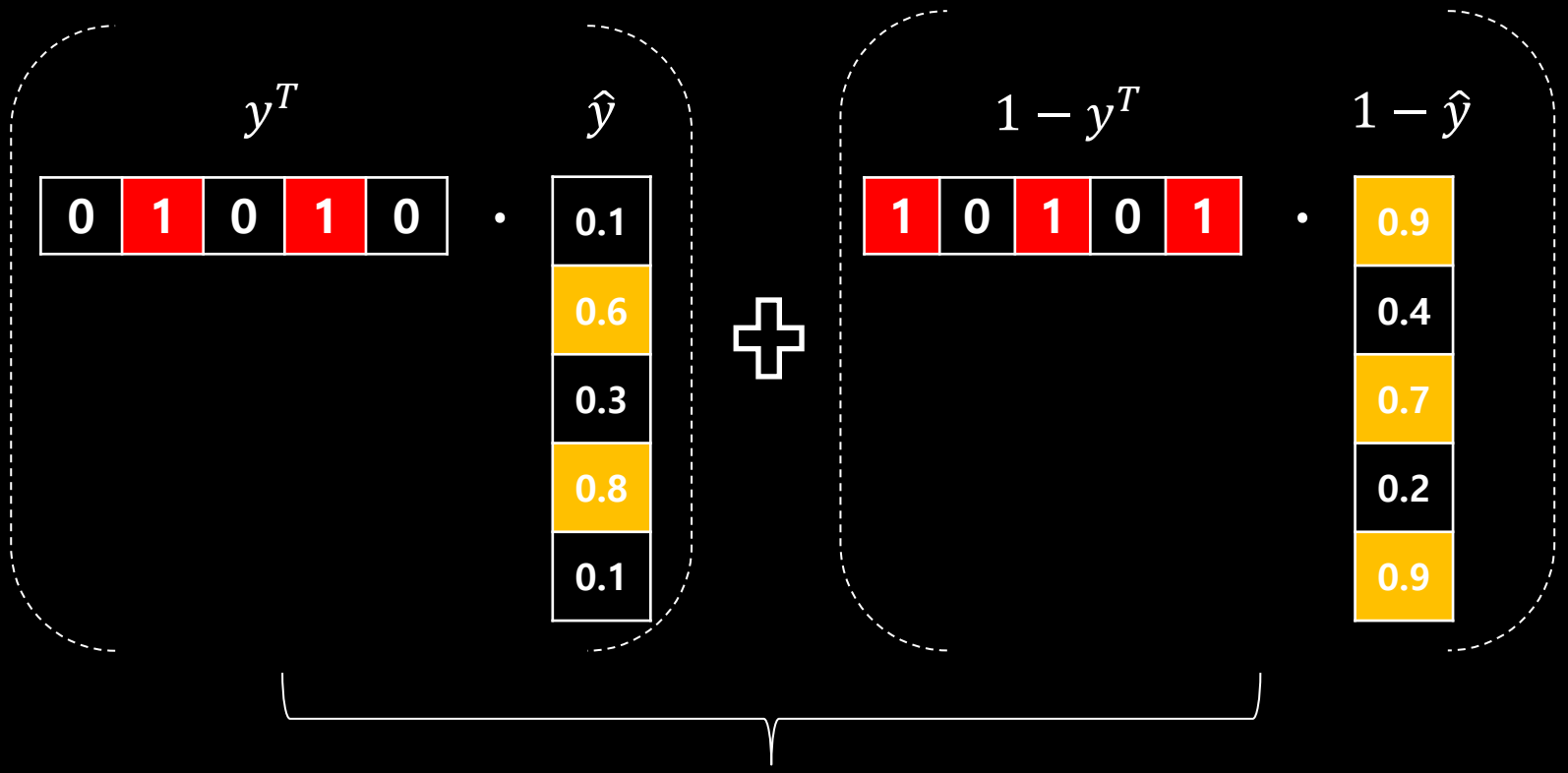
입력은 Mini batch 형태로 들어올 것임 (일반적으로 3D- / 4D-tensor 형태)





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

Binary Cross Entropy - 병렬 처리

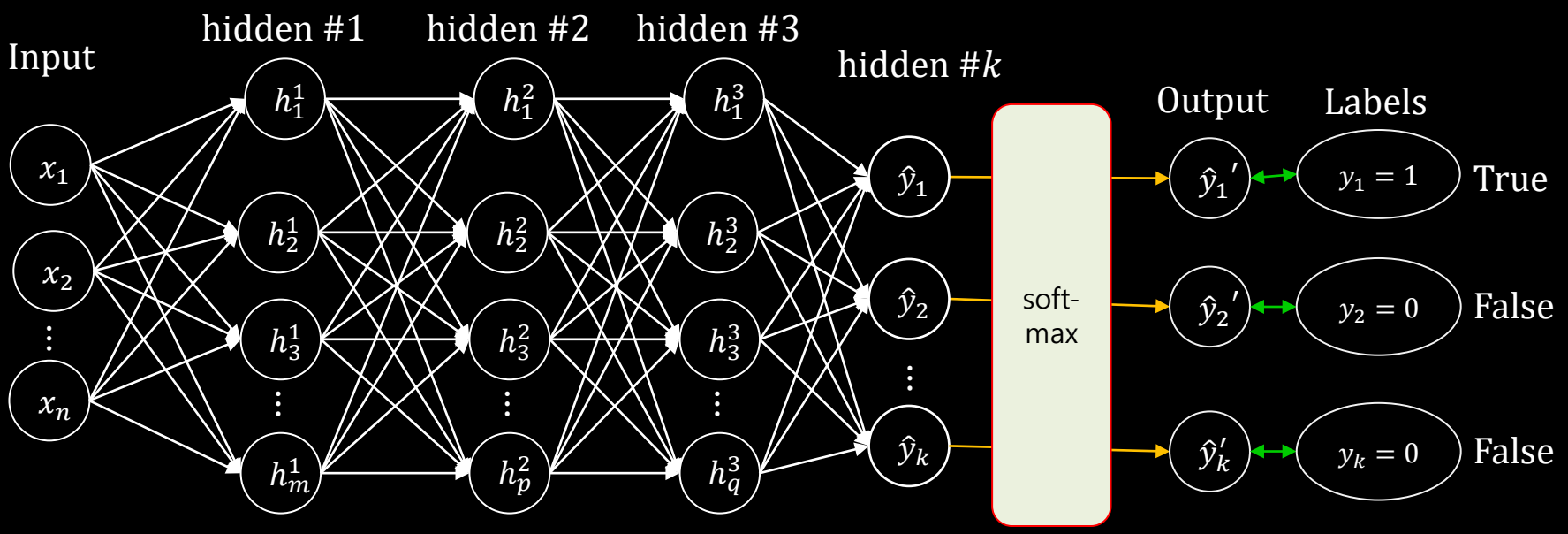


$$CE = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i^T \times \log \hat{y}_i + (1 - y_i)^T \times \log(1 - \hat{y}_i)\right]$$

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

BCE Loss의 일반화 버전 → Cross Entropy

다중 분류 네트워크 (Logistic Regression) ↔ (Ref. Linear Regression)



다중 분류에서의 엔트로피 Loss는
- 'Cross Entropy' 또는
- 'Categorical Cross Entropy' 라는

용어로 사용합니다.

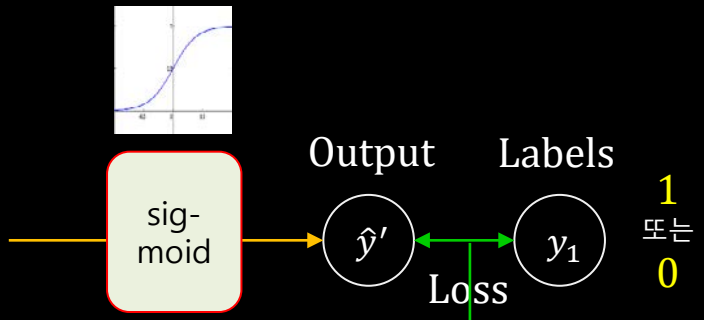
하나만 1, 나머지는 0

(One-hot Encoding)

한방에 정리하기!!!

Data Processing	1개의 데이터만 적용할 경우	병렬처리 적용할 경우 (Mini-batch)
Binary Cross Entropy	$H(Y,\hat{Y}) = -\sum_{i\in\{T,F\}} P(Y=i)\times\log P(\hat{Y}=i)$ 가능한 경우가 2개	$H(Y,\hat{Y}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j\in\{T,F\}} P(Y=j)\times\log P(\hat{Y}=j)$ $= -\frac{1}{n}\sum_{i=1}^{n}[y_i\log p_i + (1-y_i)\log(1-p_i)]$
(Categorical) Cross Entropy	$H(Y,\hat{Y}) = -\sum_{i\in Y} P(Y=i)\times\log P(\hat{Y}=i)$ 가능한 경우가 m개 $(m=\lvert Y\rvert)$	$H(Y,\hat{Y}) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j\in Y} P(Y=j)\times\log P(\hat{Y}=j)$ $= -\frac{1}{n}\sum_{i=1}^{n}\sum_{j\in Y} y_{ij}\log p_{ij}$

Cross Entropy를 사용하는 이유?



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

교수님~~

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

맞습니다 ^^.

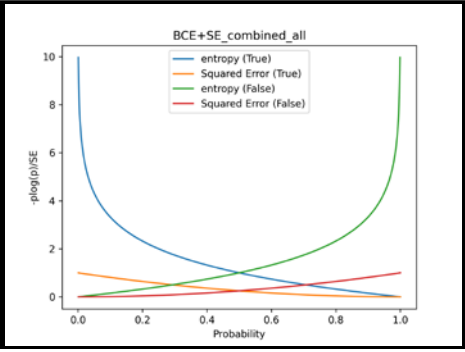
하지만, 딥러닝 신경망 학습(optimization) 관점에서
큰 차이가 있습니다.

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

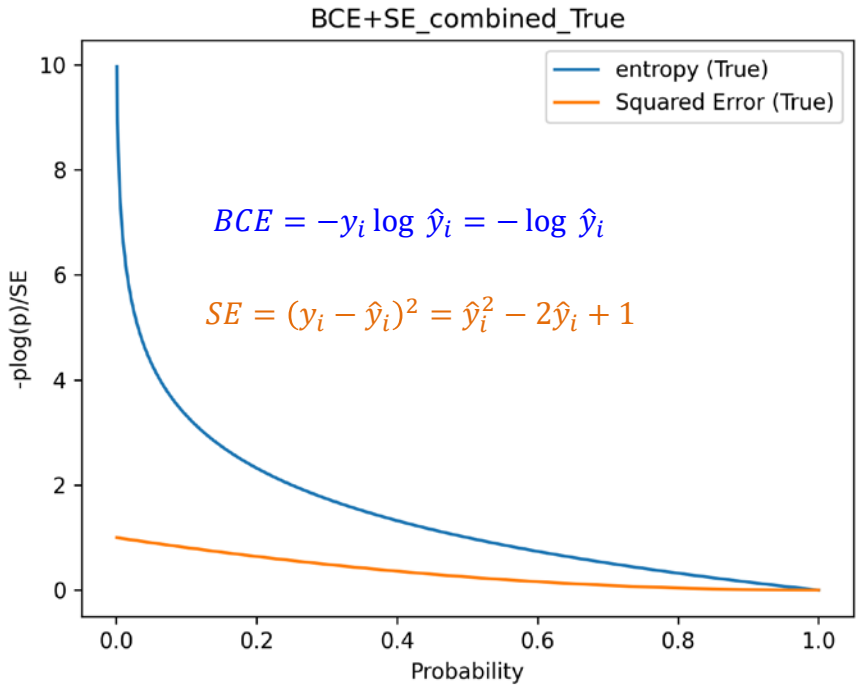
그림으로 살펴 볼까요? ^^

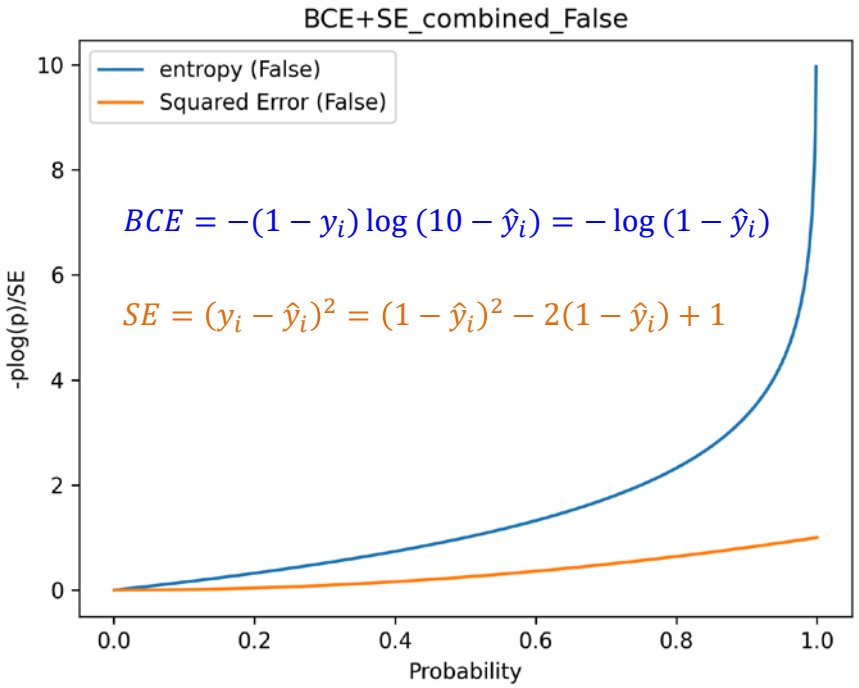
Regression Task에서는 어쩔 수 없이 MSE (Mean Squared Error)를
사용하지만, Classification Task에서는 Entropy Loss를 사용하는 것이 좋습니다. ^^

Cross Entropy를 사용해야 하는 이유 (직관적 이해)



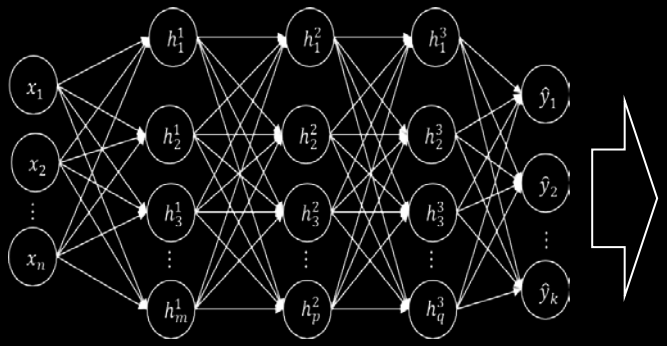
이미지 출처:
https://www.etnews.com/20160630000221



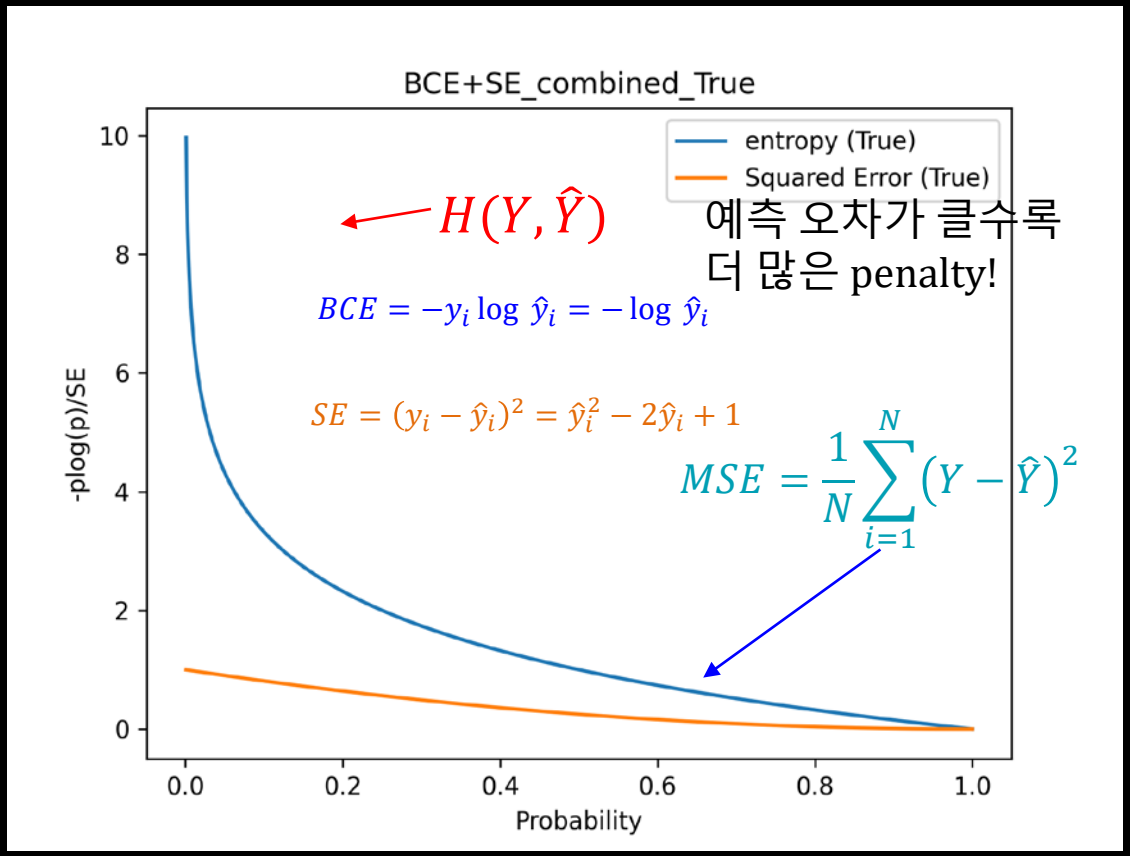


Cross Entropy 장점 - 직관적 이해

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



Back-prop 과정에서
더 큰 업데이트 수행



0 ← $\hat{Y}$ → 1

예측 확률 값의 범위

수고하셨습니다 ..^^..