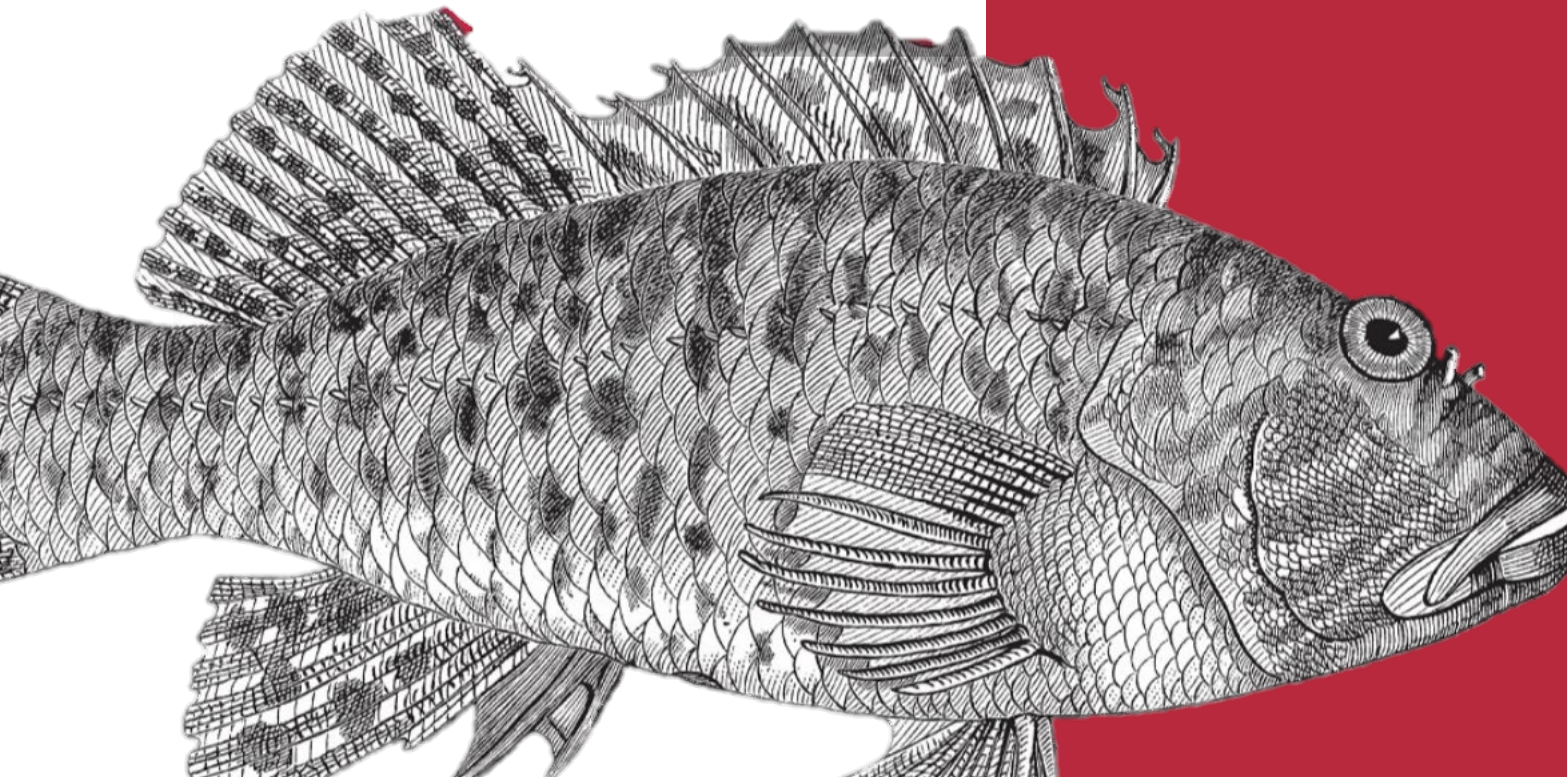


DSL 23-1 딥러닝 B조

Ch4.

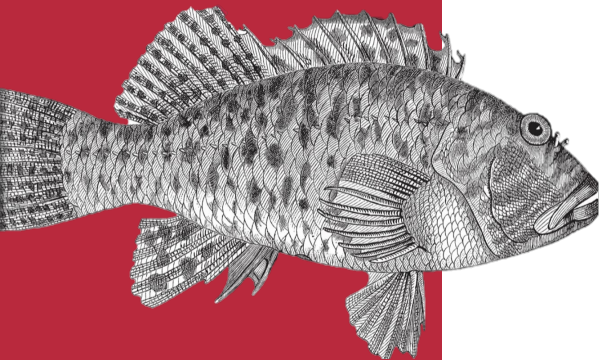
신경망 학습



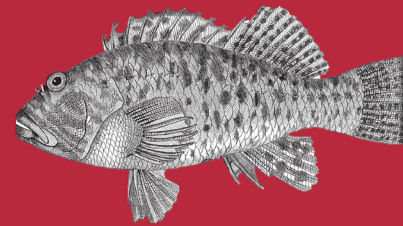
9기 이성균

목차

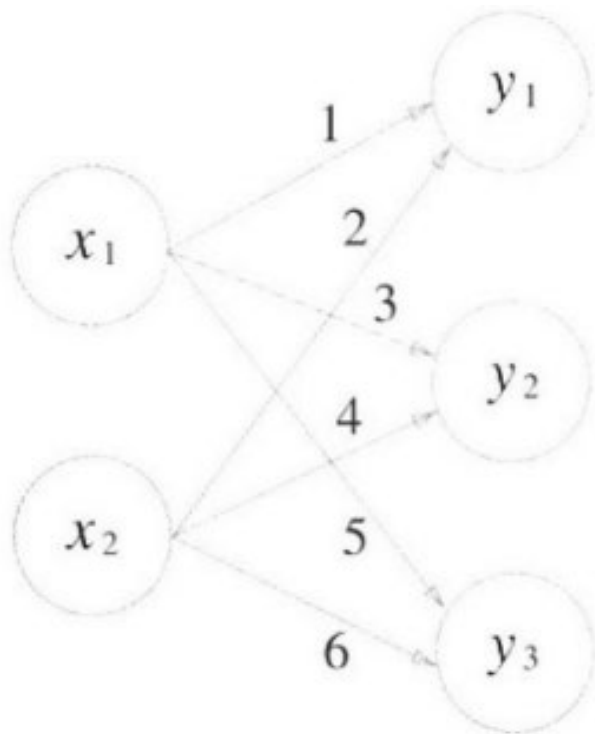
- 신경망이란? 딥러닝이란? (복습)
- 손실함수(loss function)
 - MSE
 - CEE (feat. Entropy)
 - MiniBatch
- 수치 미분 & 기울기
- 경사하강법(Gradient-descent method)



신경망이란?



- 간단히 말하면, 활성화 함수가 비선형함수인 다층 퍼셉트론
- 활성화 함수 예시 : 지시함수^{indicator function}, ReLU함수, 시그모이드함수



$$\begin{pmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{pmatrix}$$
$$\mathbf{X} \mathbf{W} = \mathbf{Y}$$

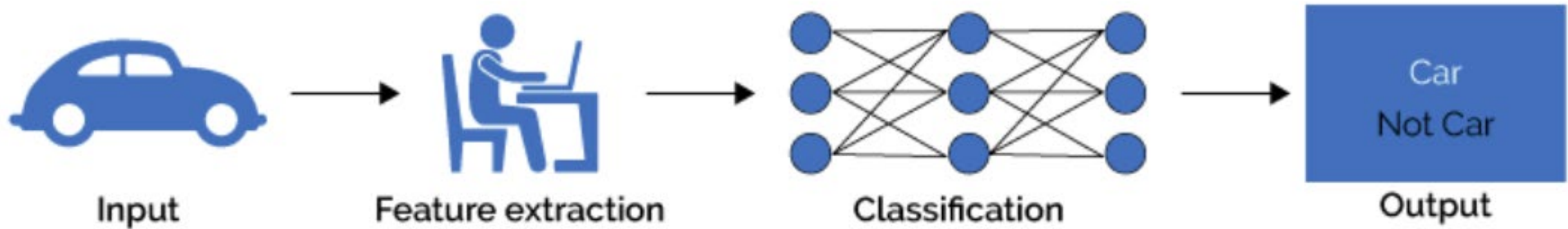
2 2 × 3 3

일치

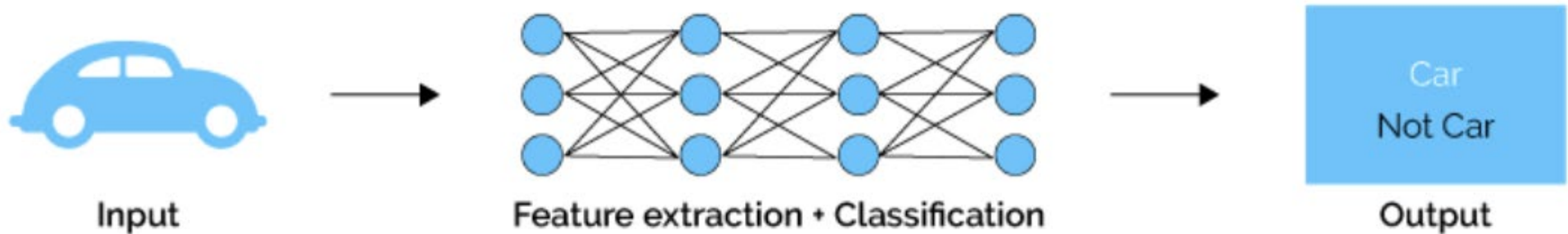


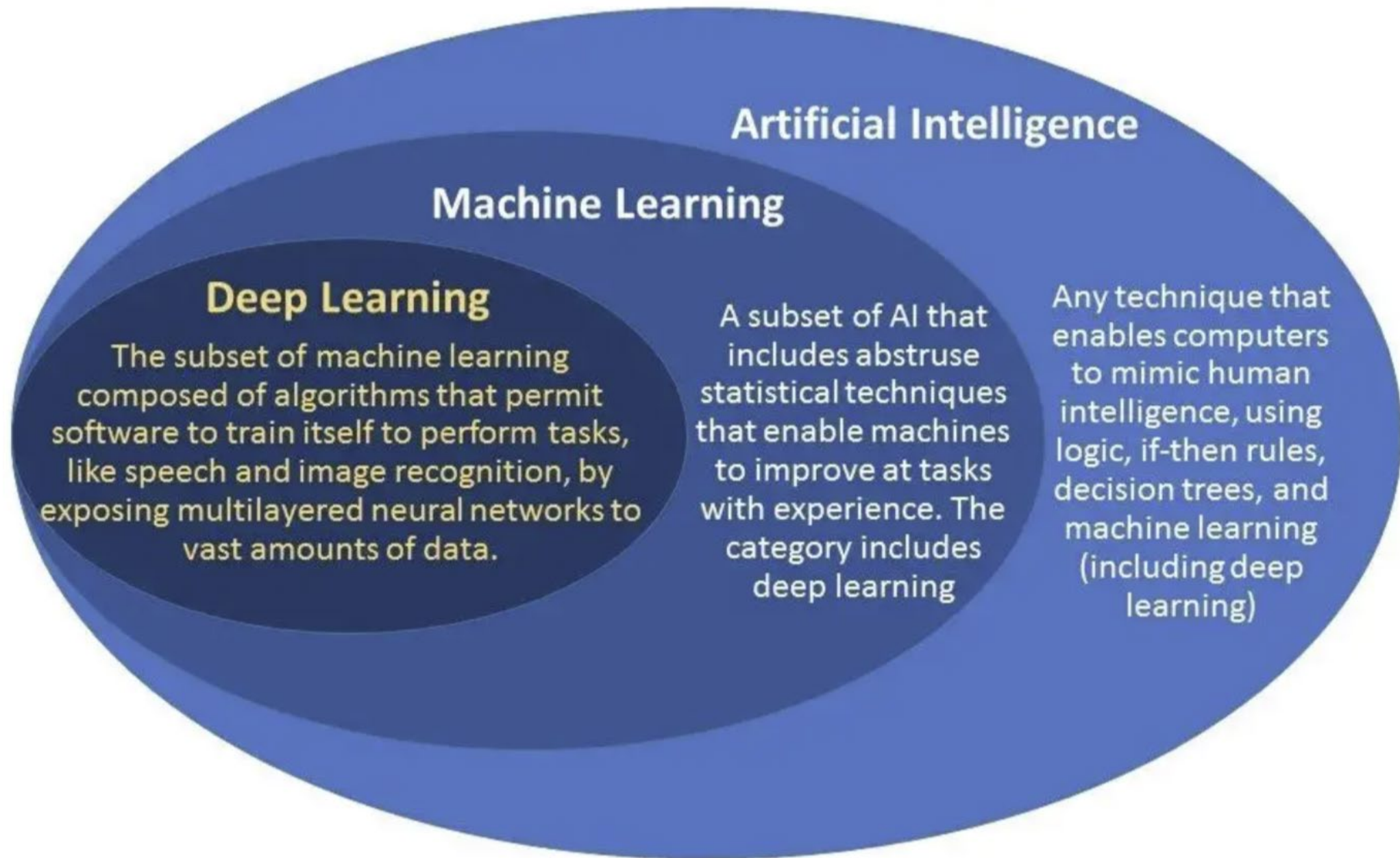
- 학습 : 훈련 데이터로부터 가중치 매개변수의 최적값을 자동으로 획득하는 것
- 언제까지 수천,수만개의 가중치 매개변수를 수작업으로 설정할 것인가?
→ 사람의 개입 없이 스스로 데이터를 얻고 매개변수를 조정해서 결과를 얻는 모델을 만들어보자! (= 딥러닝 = 종단간 기계학습)

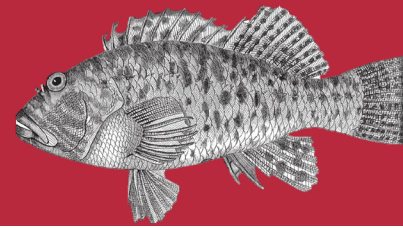
Machine Learning



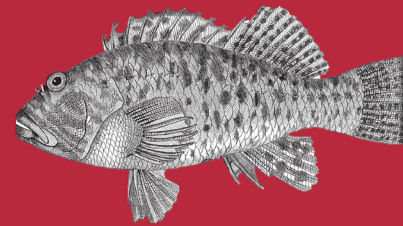
Deep Learning








- 인공신경망(ANN, Artificial Neural Networks)의 **범용 능력**이 중요!
→ **훈련 데이터**로 최적의 매개변수 지정, **시험 데이터**로 범용능력 측정
- 그러나 한 개의 훈련 데이터와 한 개의 시험 데이터에만 최적화되지 않도록(**오버피팅** Overfitting 되지 않도록) 유의해야 함



- 어떻게 최적의 매개변수를 찾아낼 것인가?
→ 손실함수 loss function, cost function를 이용! (신경망의 성질이 나쁘다는 것
을 파악하는 지표)

- 대표적으로 평균제곱오차(MSE), 교차 엔트로피 오차(CEE)
- 큰 데이터를 처리하기 위해 미니배치 학습법을 자주 이용함



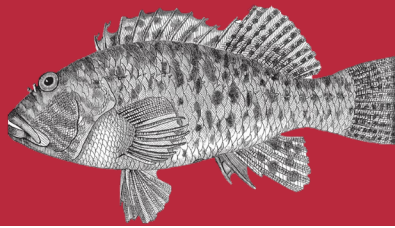
평균제곱오차 Mean Squared Error



$$E = \frac{1}{2} \sum_k (\overset{\hat{y}_i}{y_k} - \overset{y_i}{t_k})^2$$

y_k k번째 추정값
 t_k k번째 실제값(정답 레이블)
(one-hot encoding)
 k 데이터의 차원(벡터 원소의 개수)

연속형 데이터에 자주 활용 ▶ E가 클수록 신경망의 성능이 나쁘다!

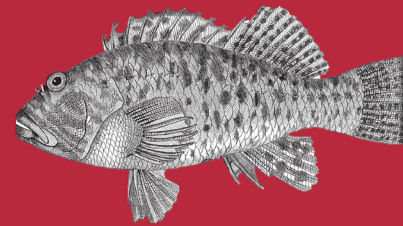


- 정보 이론에서 엔트로피(entropy) : 사건을 반복하여 얻은 정보량의 기댓값 (**평균 정보량**)

- 정보를 많이 알수록, 새롭게 알게 되는 정보량은 감소한다.
- 어떤 사건이 발생할 확률이 크면, 새롭게 알게 될 정보량은 적다.
- 어떤 사건이 발생할 확률이 작으면, 엔트로피가 커진다.
- 엔트로피가 크다 = 어떤 상태에서의 불확실성이 크다



교차 엔트로피 오차 Cross Entropy Error



$$E = - \sum_k t_k \log y_k$$

t_k : k번째 실제값(정답 레이블)
(one-hot encoding)
데이터의 차원(벡터 원소의 개수)

y_k : k번째 추정값

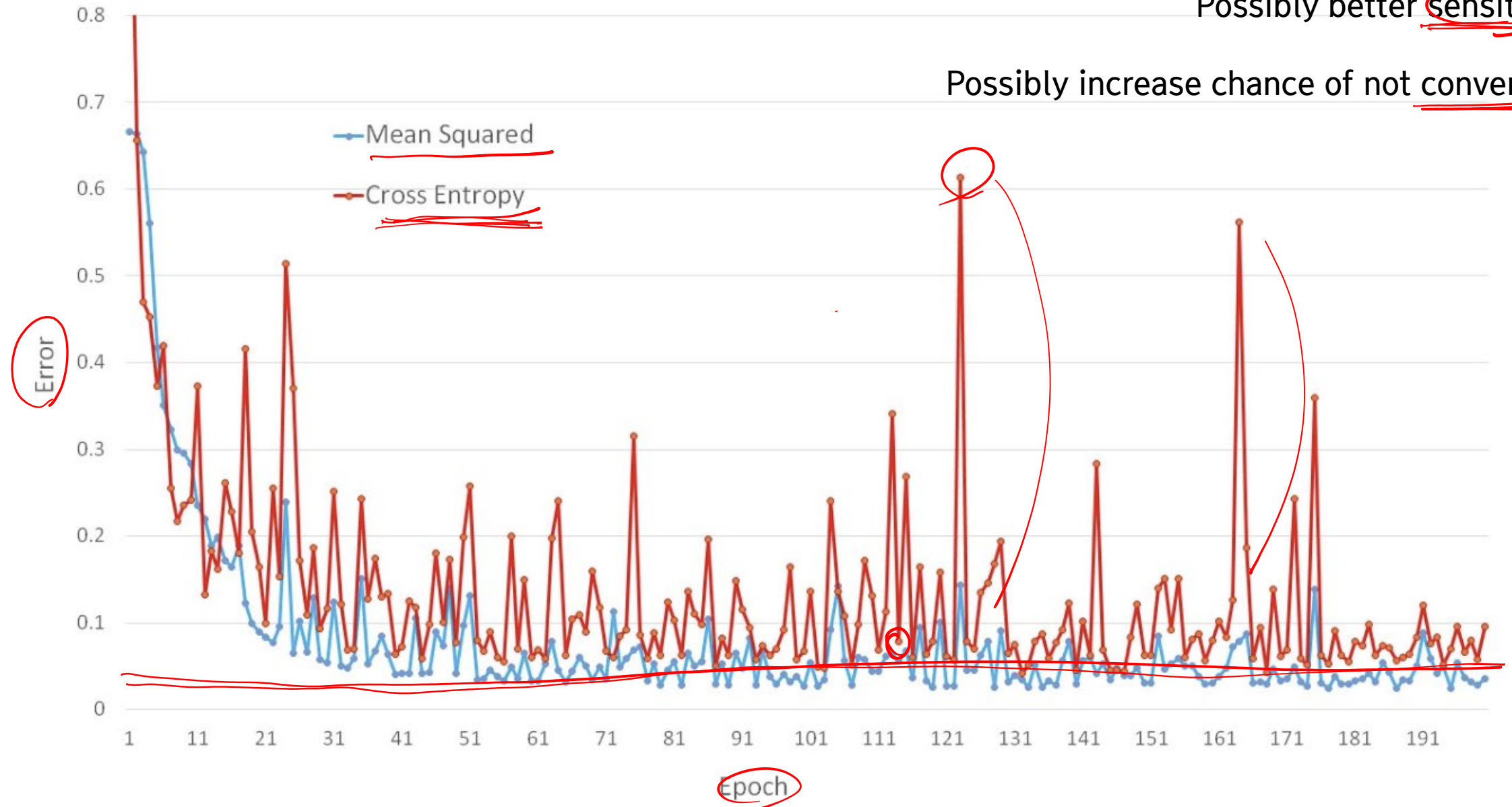
Handwritten notes: y_i , y_i , $0, 1$, Indicator, $\sum (y_i \log x_i)$, $0 \times \square$, $1 \times \square$, \log

범주형 데이터에 자주 활용 ▶ E가 클수록 신경망의 성능이 나쁘다!
(정보량이 0에 가까워져서 발생확률이 1에 가깝도록 해야 성능이 좋다)

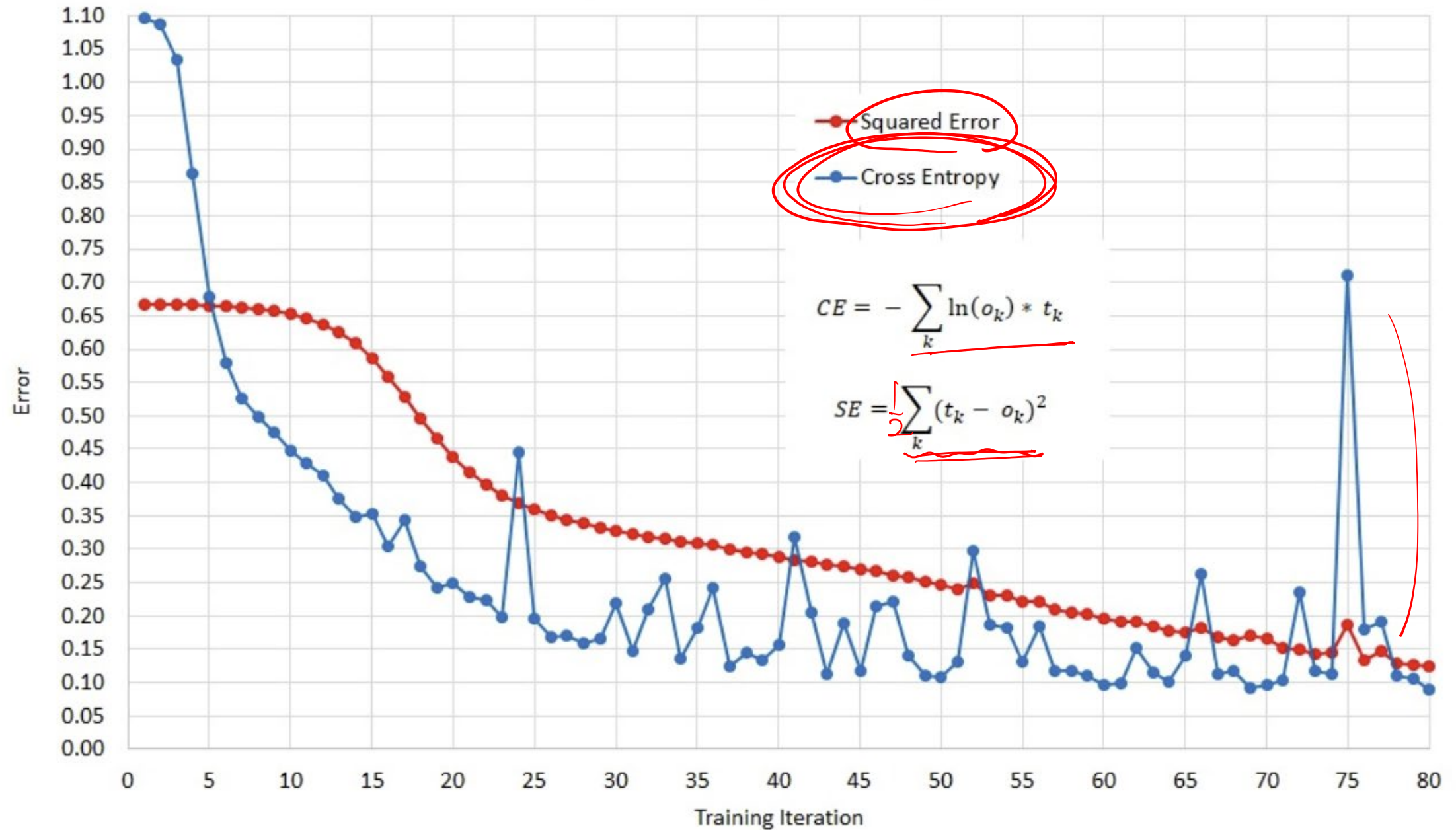
Mean Squared Error vs. Cross Entropy Error

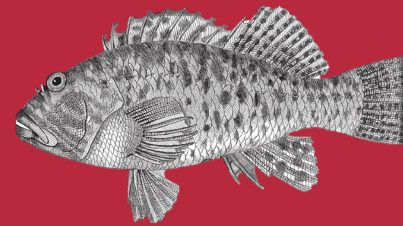
Possibly better sensitivity
vs.

Possibly increase chance of not converging



Squared Error vs. Cross Entropy Error





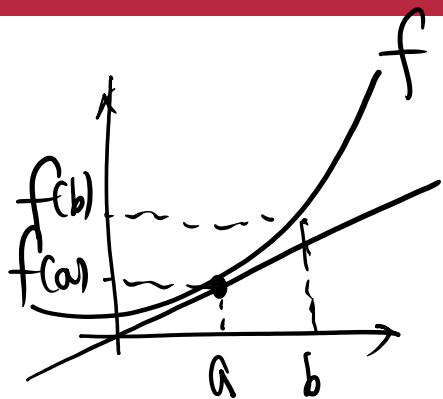
- 훈련 데이터 모두의 손실 함수의 합을 표본집단의 개수(N)으로 나눠?

너무 오래걸린다...

0.0001 60,000개 vs 1,000개

- 따라서 표본집단 전체(N 개) 중 'minibatch' n 개만 뽑아서 **평균 손실 함수의 근사치**를 구한다! (n/N 의 확률로 무작위 비복원 추출)

차분(평균변화율), 미분(순간변화율), 수치 미분



$$\text{평균변화율} = \frac{f(b) - f(a)}{b - a}$$

$$\longrightarrow \lim_{b \rightarrow a} \frac{f(b) - f(a)}{b - a} = f'(a) = \text{순간변화율}$$

$$= (a, f(a))$$

이때의 기울기

= 미분계수

\Downarrow

$$\lim_{x \rightarrow a} \frac{f(x) - f(a)}{x - a} = f'(a) \text{로 바꾸기}$$

수치 미분
 \downarrow
해석적 미분

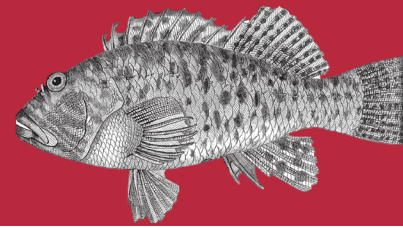
$$\lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

$$f(x) = x^2$$

$$f'(x) = 2x \quad \therefore f'(1) = 2$$

$$h \neq 0 \quad \frac{f(x+h=0.00001) - f(x)}{0.00001}$$

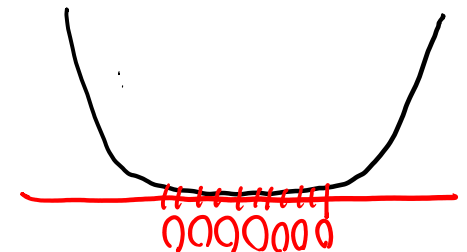
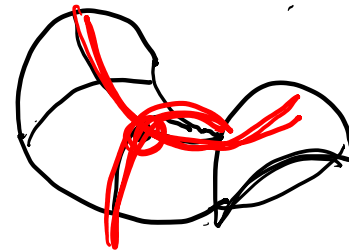
"실제" $\frac{h=0.00001}{1e-05}$



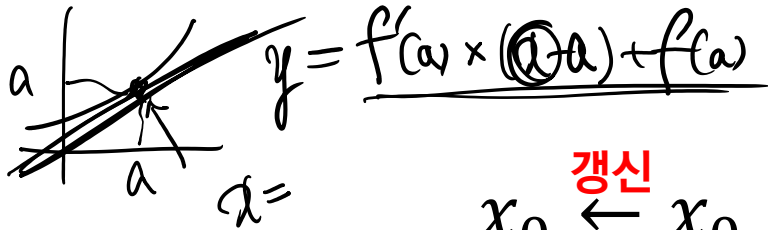
- 손실함수의 최솟값(y)을 만드는 값($x =$ 가중치 매개변수)를 찾기 위해, 기울기를 활용해보자.

→ 기울기가 가리키는 쪽이 함숫값을 가장 크게 줄이는 방향!

- 다만 안장점(saddle point), 고원(plateau) 모양을 가진 함수의 경우 정체기가 생길 수 있으니 주의해야 함.



경사하강법 Gradient-descent method



$$x_0 \xleftarrow{\text{갱신}} x_0 - \eta \frac{\partial f}{\partial x_0} \bigg|_{x_0}$$

Newton's method

$x_0 \leftarrow x_0 - \frac{f'(x_0)}{f''(x_0)}$

$$x_1 \xleftarrow{\text{갱신}} x_1 - \eta \frac{\partial f}{\partial x_1} \bigg|_{x_1}$$

$f'(x_0, x_1) = x_0^2 + x_1^2$

• η = 학습률(learning rate) ... Hyperparameter

미리 지정해두어야 함 (0.1, 0.01 등)

• 경사법으로 얼마나 반복할 것인지 횟수도 지정해주어야 함

• 학습률이 너무 크면 전혀 다른 값이 나오고 (발산), 너무 작으면 갱신X

경사하강법 알고리즘

