

파이썬(Python)으로 구현하는

# Classification

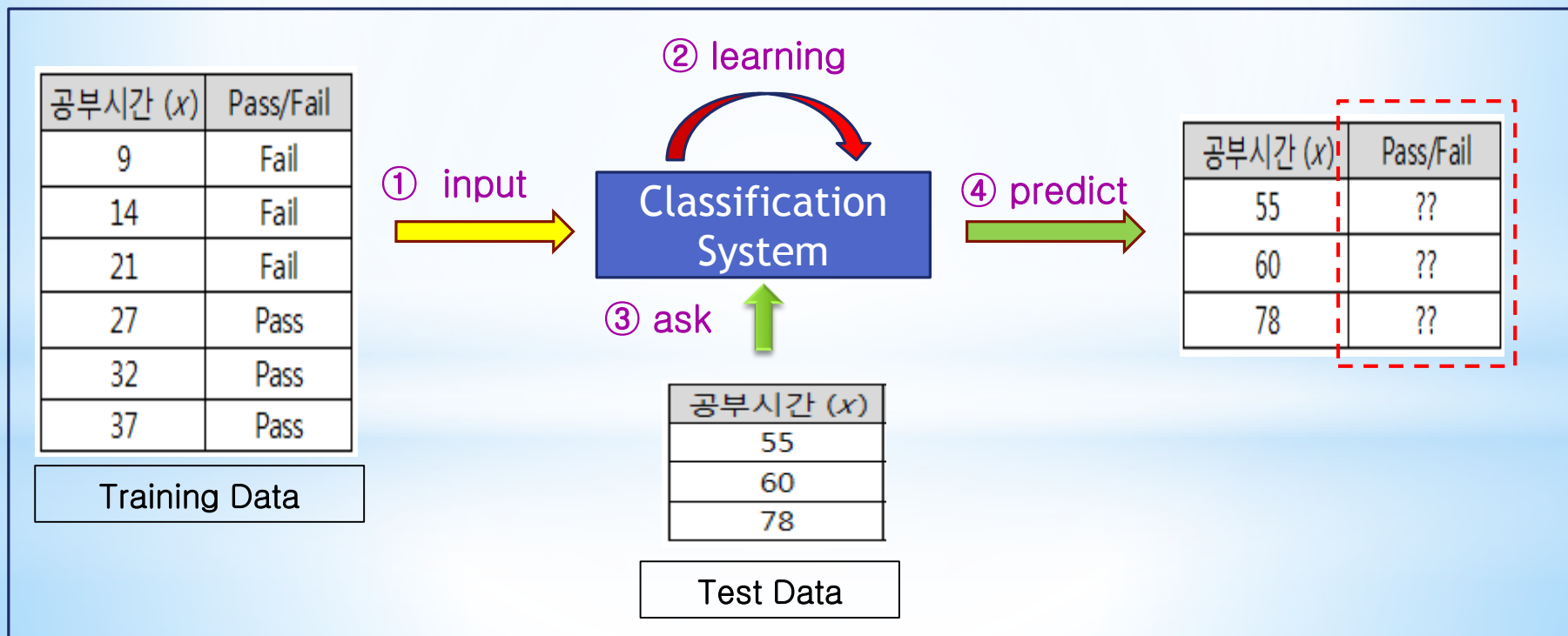
– 개념 • Cross-Entropy –

# Logistic Regression – Classification

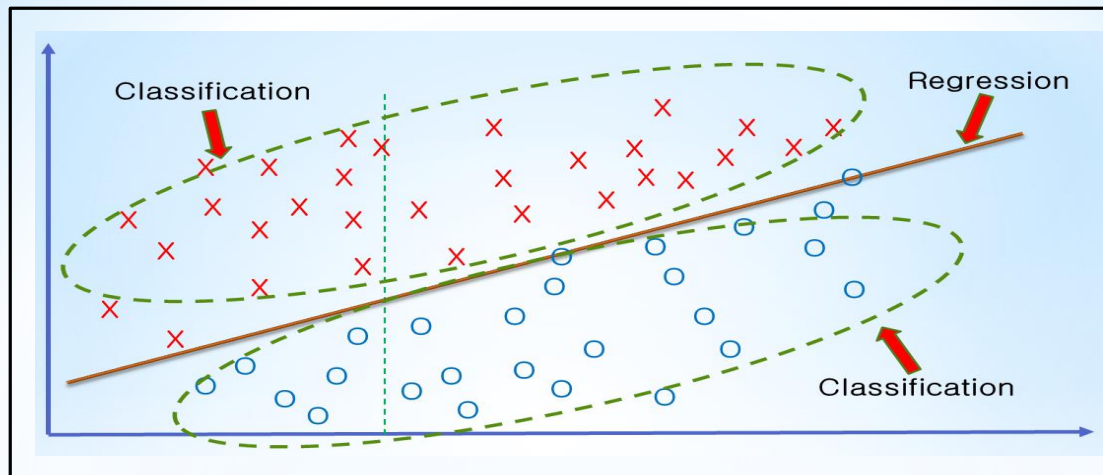
## ➤ 분류 (Classification)

- Training Data 특성과 관계 등을 파악 한 후에, 미지의 입력 데이터에 대해서 결과가 어떤 종류의 값으로 분류 될 수 있는지를 예측하는 것

(예) 스팸문자 분류 [Spam(1) or Ham(0)], 암 판별 [악성종양(1) or 종양(0)]



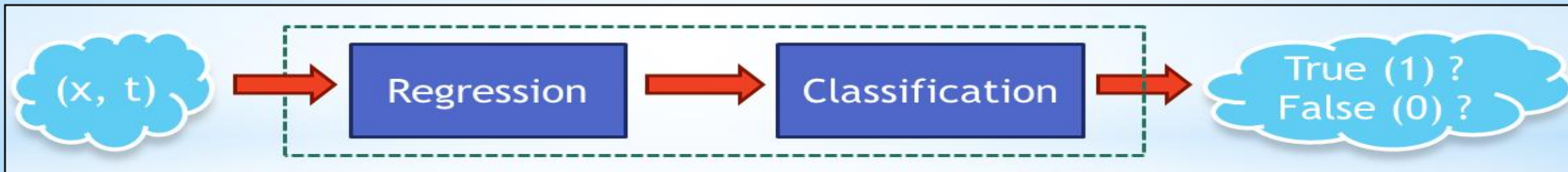
# Classification



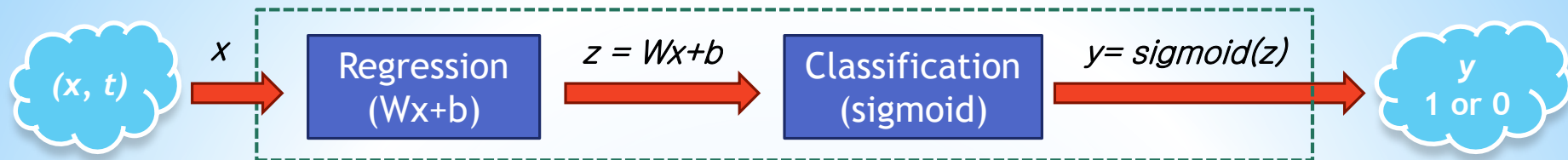
➤ 즉, **Logistic Regression 알고리즘**은,

- ① Training Data 특성과 분포를 나타내는 최적의 직선을 찾고(**Linear Regression**)
- ② 그 직선을 기준으로 데이터를 위(1) 또는 아래(0) 등으로 분류(**Classification**) 해주는 알고리즘

⇒ 이러한 Logistic Regression은 Classification 알고리즘 중에서도 정확도가 높은 알고리즘으로 알려져 있어서 Deep Learning에서 기본 Component로 사용되고 있음



# Classification – sigmoid function



➤ 출력 값  $y$  가 1 또는 0 만을 가져야만 하는 분류(classification) 시스템에서, 함수 값으로 0~1 사이의 값을 가지는 sigmoid 함수를 사용 할 수 있음

⇒ 즉, linear regression 출력  $Wx+b$  가 어떤 값을 갖더라도, 출력 함수로 sigmoid 를 사용해서 ① sigmoid 계산 값이 0.5보다 크면 결과로 1 이 나올 확률이 높다는 것이기 때문에 출력 값  $y$  는 1 을 정의하고 ② sigmoid 계산 값이 0.5 미만이면 결과로 0 이 나올 확률이 높다는 것이므로 출력 값  $y$  는 0 정의하여 classification 시스템을 구현할 수 있음

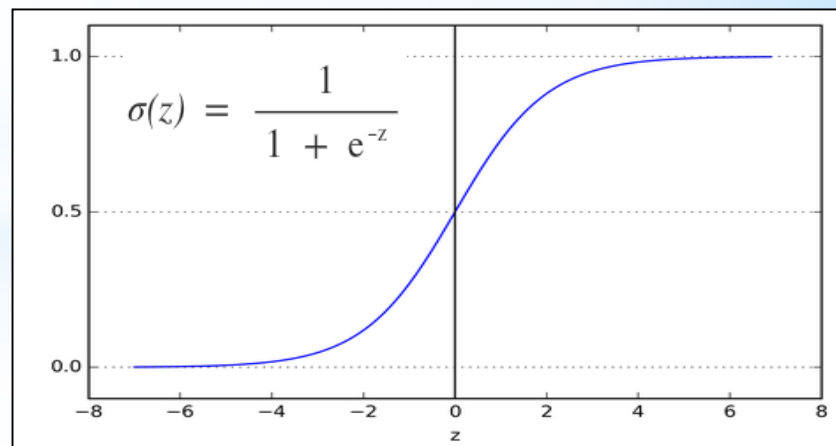
※ sigmoid 함수의 실제 계산 값  $\text{sigmoid}(z)$ 는 결과가 나타날 확률을 의미함

$$y = \frac{1}{1 + e^{-(Wx+b)}}$$

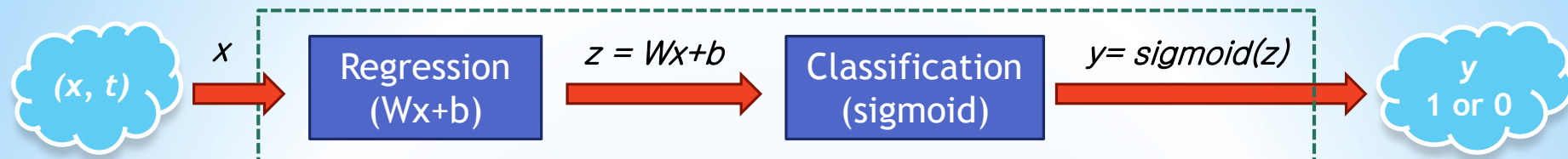
또는

$$z = Wx + b$$

$$y = \text{sigmoid}(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$



# Classification – 손실함수(loss function), W, b



- 분류시스템(classification) 최종 출력 값  $y$  는 sigmoid 함수에 의해 논리적으로 1 또는 0 값을 가지기 때문에, 연속 값을 갖는 선형회귀 때와는 다른 손실함수 필요함

$$y = \frac{1}{1 + e^{-(Wx+b)}} \quad , \quad t_i = 0 \text{ or } 1$$

손실함수  
(cross-entropy)

$$E(W,b) = - \sum_{i=1}^n \{t_i \log y_i + (1-t_i) \log(1-y_i)\}$$

손실함수  $E(W,b)$  최소값을 갖는  $W$

$$W = W - \alpha \frac{\partial E(W,b)}{\partial W}$$

손실함수  $E(W,b)$  최소값을 갖는  $b$

$$b = b - \alpha \frac{\partial E(W,b)}{\partial b}$$

※  $\alpha$  는 학습율(learning rate) 이라고 부르며,  $W$  값의 감소 또는 증가 되는 비율을 나타냄

## [참고] 손실함수(loss function) cross-entropy 유도

- classification 최종 출력 값  $y$  는 sigmoid 함수 에 의해 0~1 사이의 값을 갖는 확률적인 분류 모델임. 그래서 다음과 같이 확률 변수  $C$  를 이용해 출력 값을 나타낼 수 있음

수식	수식 설명
$p(C=1 x) = y = \text{sigmoid}(Wx+b)$	입력 $x$ 에 대해 출력 값이 1 일 확률을 $y$ 로 정의. $y$ 는 1 또는 0 이므로 $y = \text{sigmoid}(Wx+b)$ 나타낼 수 있음
$p(C=0 x) = 1 - p(C=1 x) = 1 - y$	입력 $x$ 에 대해 출력 값이 0 일 확률이며, 확률은 모두 더한 것이 1 이므로, 출력 값이 0 일 확률은 $1 - y$ 임 ( $y$ 는 출력 값이 1 일 확률로, $\text{sigmoid}(Wx+b)$ 로 정의함)
$p(C=t x) = y^t(1-y)^{1-t}$	확률변수 $C$ 는 0 이나 1 밖에는 값을 가질 수 없으므로, (즉, 정답 $t = 0$ or $1$ ) 다음처럼 나타냄
<div style="display: flex; justify-content: space-between; margin-bottom: 10px;"> <div style="border: 1px solid red; padding: 5px; width: 45%;"> <p>우도함수란, 입력 <math>x</math> 에 대해 정답 <math>t</math> 가 발생될 확률을 나타낸 함수</p> </div> <div style="border: 1px solid red; padding: 5px; width: 45%;"> <p>확률은 독립적이므로, 각 입력 데이터의 발생 확률을 곱해서 우도함수 나타냄</p> </div> </div> <div style="display: flex; align-items: center;"> <div style="color: red; font-size: 2em; margin-right: 10px;">↙</div> <math display="block">L(W,b) = \prod_{i=1}^n p(C=t_i   x_i) = \prod_{i=1}^n y_i^{t_i} (1-y_i)^{1-t_i}</math> </div>	가중치 $W$ 와 바이어스 $b$ 를 최우추정하기 위한 우도함수 (likelihood function)은 다음과 같이 나타낼 수 있으며, 이 우도함수값이 최대가 되도록(최우추정) $W$ 와 $b$ 를 업데이트 해 나가면 머신러닝에서 학습이 잘 된 것임.
$E(W,b) = -\log L(W,b)$ $= -\sum_{i=1}^n \{t_i \log y_i + (1-t_i) \log(1-y_i)\}$	함수가 최대값이 되는 것을 알기 위해서는 $W$ 와 $b$ 에 대해 편미분을 해야 하는데, 곱하기는 미분이 불편하므로 양변에 $\log$ 를 취해 덧셈 형태로 바꾸어 주고, 함수의 부호를 바꾸어 주면 함수의 최대화 문제는 최소화 문제로 바꿀 수 있으므로 다음처럼 나타냄 (파라미터 최적화를 위해서는 함수 최소값을 구하는 것이 일반적임)



# simple logistic regression (classification) 예제

[1] 먼저 loss\_func, error\_val, predict, sigmoid 함수를 구현하고, numerical\_derivative 함수를 이용하여 다음의 분류 시스템을 구현하시오

공부시간 (x)	Fail/Pass (t)
2	0
4	0
6	0
8	0
10	0
12	0
14	1
16	1
18	1
20	1

training data

