· 03/28

A·I

- Strong AI
- Weak AI
  - 따로 개발들 (?)
  - machine Learning

Neural Network : 비정형 데이터 학습

NN을 제외한 나머지 ⇒ 정형 데이터 학습 예측

**

(지도학습)
- supervised Learning
- unsupervised Learning (비지도 학습)
- Reinforcement Learning (강화학습)

- Regression (회귀)
- SVM (support Vector machine)
- Decision Tree → Random Forest
- Naive Bayes
- KNN (K-Nearest Neighbor)
- Neural Network (신경망)

* 예측모델이 어떤 종류의 값을 예측하나요?

- Regression (연속적인 숫자값) (회귀)
- classification (분류) (어떤 분류에 속하는가 → discrete 숫자값)

- K-Means
- DBScan

clustering (군집)

Deep Learning

X

*
- DNN
- CNN
- RNN(LSTM)
- GAN, RCNN
...

Regression (회귀) → "평균" ~로 돌아가요

프란시스 골턴 (1800~1900) → **Regression Toward Mean**

→ 데이터가 회귀모델이 제시한 추세선에 수렴

찰스 다윈 (종의기원) → 인간 개선을 목적으로 우량종 육성을 목적으로 선택적 출산 → 우생학 (인간을 개선하는 과학)

* 훌륭한 사람은 자연환경보다는 유전적 영향을 많이 받는다.

키가 커요 → 자식 → 키가 커요

키가 작아요 → 키가 작아요

---

* **Regression**

* 어떤 데이터 대해 그 데이터에 영향을 주는 조건들의 평균적인 영향력을 이용해 데이터에 대한 조건부 평균을 구하는 기법

아파트 평균가격 = (1040 × 평수) + (350 × 역세권 여부) + (70 × 층수) + ···

- 평균은 대표성을 가져요
  - 여러 데이터를 가장 잘 표현하는.

* Regression Model

→ 어떤 데이터에 대해
그 값에 영향을 주는 조건들을 고려하여
그 데이터를 가장 잘 표현하는 함수.

독립변수가 1개인 경우

→ $y = \beta_1 x + \beta_0$

$y$: 종속변수 (아파트가격)

$x$: 독립변수 (지역)

$\beta_0$: 후렌치, 실제값을 보정하기 위한 상수.

Classical Linear Regression 기반

→ $y = ax + b$ (기울기: $a$, 절편: $b$)

└→ 왜 이런 Regression Model을 만드나요?

① 정보파악이 쉽다
② 조건의 영향력
③ 예측을 파악

- Regression Model

여러가지 복잡한 조건하에서
평균을 집계할 수 있도록하는 효율적인 방법
조건들을 이용해 평균을 추정할 수 있도록 해주는 기법.

"한가지 주의해야 할점"

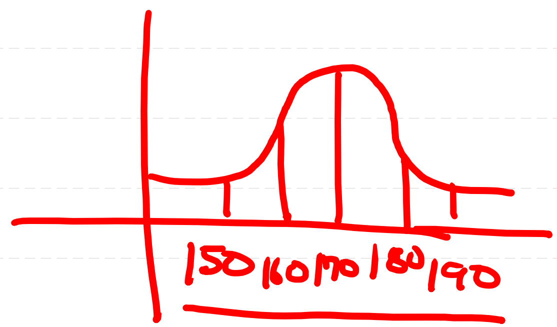
↳ 평균을 구할때 주의해야 할 사항이 regression에도 적용

① 우리나라 성인 남성의 키

② 우리나라 전체 근로자의 연봉

이 중 평균을 대푯값으로 사용할 수 있는 경우는?

정규분포




(3300)
전체평균


중위
(2600)

① 회귀 모델 = 영향력1 × 조건1 + 영향력2 × 조건2 + 영향력n × 조건n

회귀계수 (regression coefficient)

?

② 고전적 선형 회귀 모델 (Classical Linear Regression Model)

$$\hat{y} = \beta_0 + \sum_{i=1}^{p} \beta_i x_i$$

일반식

종속변수

- 단순 선형 회귀 (Simple Linear Regression)
  → 독립변수가 1개
- 다중 선형 회귀 (Multiple Linear Regression)
  → 독립변수가 2개 이상

# Machine Learning에서 Linear Regression

| x(시간) | t(점수) |
|---|---|
| 10 | 95 |
| 8 | 80 |
| 5 | 63 |
| 2 | 15 |
| 1 | 5 |

Training Data Set

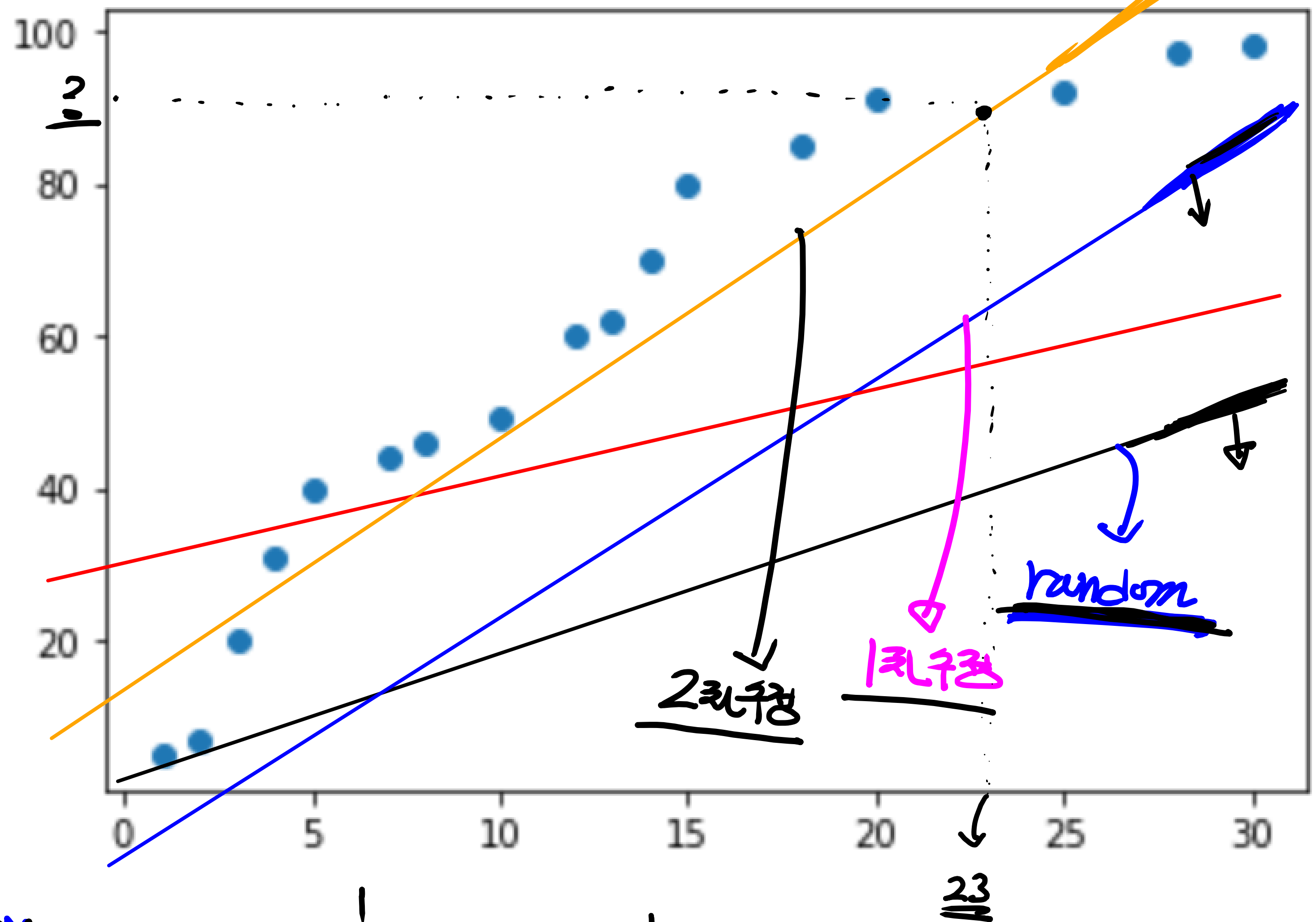
* Linear Regression (Machine Learning System) → ??점

7시간 공부하면 몇점?

단순 선형 회귀(simple Linear Regression)의 이문제를

도식(그래프)로 표현해 봐요.

→ scatter(산점도)




★ 고전적 선형 회귀 모델
(Classical Linear Regression model)

↓

$$\hat{y} = \beta_0 + \sum_{i=1}^{p} \beta_i x_i \quad \text{(일반식)}$$

↓ 독립변수 1개

$$\hat{y} = \beta_0 + \beta_1 x_1$$

$$y = ax + b \quad \text{(직선)}$$

? ? 어떻게 찾나요?

처음에는 a, b 값은 random

a와 b를 수정 → 조금더 데이터를 잘 표현 직선

a와 b를 수정 → 조금더 적합한 직선

★ $y = Wx + b$

W: weight (가중치)

b: bias (바이어스)

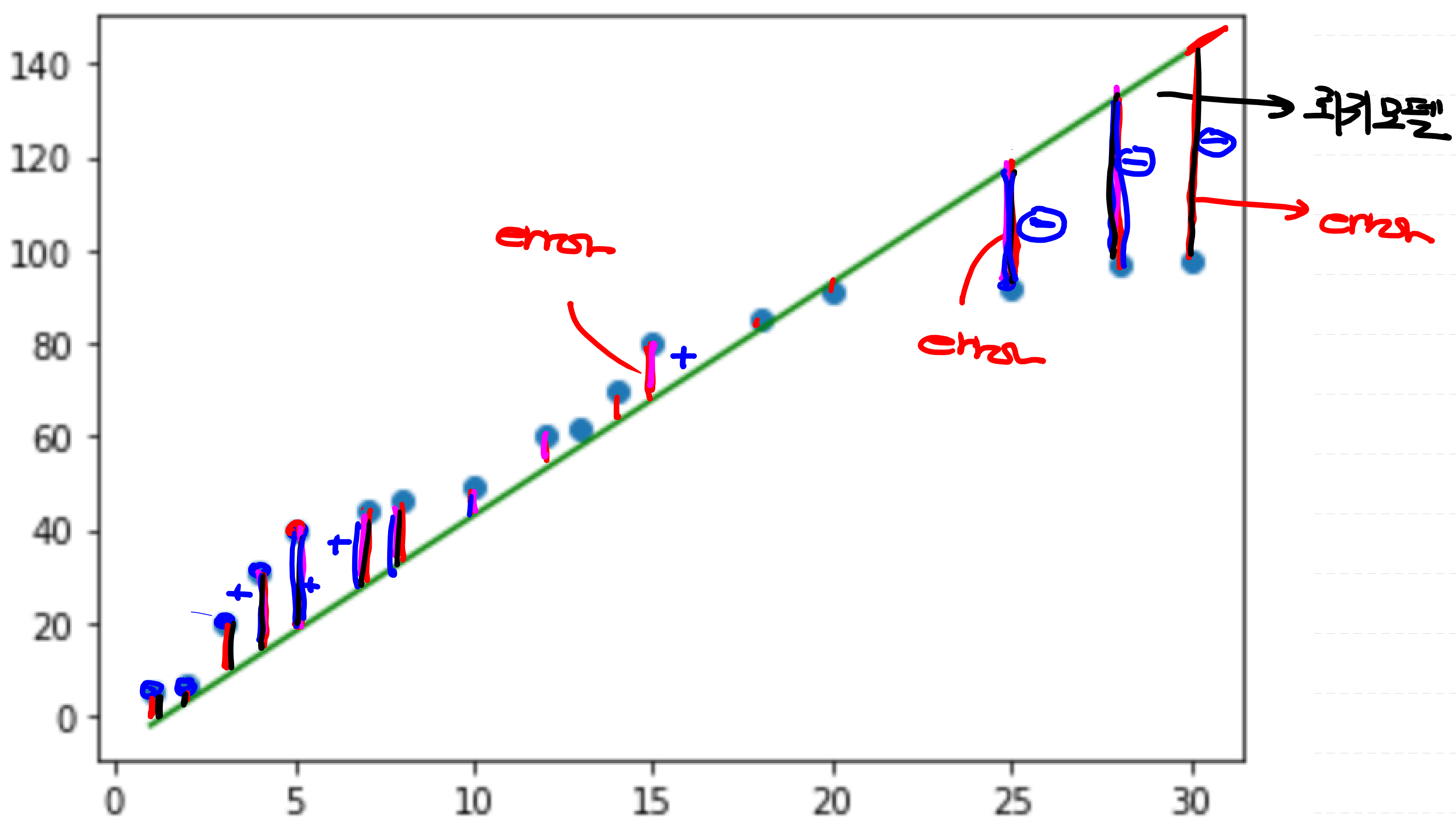
찾아나가는 과정

⇓

"learning" (학습)

$y = wx + b$

(5) (-7)




* 오차(error) : 실제값(정답, $t$) 과 예측값($y$)의 차이 → 작을수록 좋아요

$$Error = t - y = t - (wx + b)$$

부호문제로 X

오차의 합이 당연히 작아야 좋아요!! → 오차들은 값이 여러개 → 하나의 숫자로 표현하고싶어! → 흠 ㅎㅎㅎ

- 앗 그러면 각 error의 절대값의 합은 어때요?!

하나의 방법이기는 해요. 하지만 우리는 다른방법을 이용.

- 다른 방식으로 우리는 오차의 제곱의 평균을 이용.

장점?? → 오차에 대한 가중치를 줄 수 있어요.

⇒ 평균제곱오차 (MSE : Mean Squared Error)

↳ 작으면 작을수록 좋아요

↳ 평균제곱오차(MSE)를 가장 작게 만드는 W, b를 찾는 문제로 생각할 수 있어요.!

↳ loss function (cost function)

★ 손실함수 or 비용 함수

이 함수를 찾는 방법을 "최소제곱법"이라고 해요 (Least squared Method)

"오차"

$$\text{loss function} = \frac{(t_1 - y_1)^2 + (t_2 - y_2)^2 + (t_3 - y_3)^2 + \cdots (t_n - y_n)^2}{n}$$

$y = wx + b$

$$= \frac{(t_1 - (wx_1 + b))^2 + (t_2 - (wx_2 + b))^2 + \cdots (t_n - (wx_n + b))^2}{n}$$

$$= \frac{1}{n} \sum_{i=1}^{n} [t_i - (wx_i + b)]^2$$

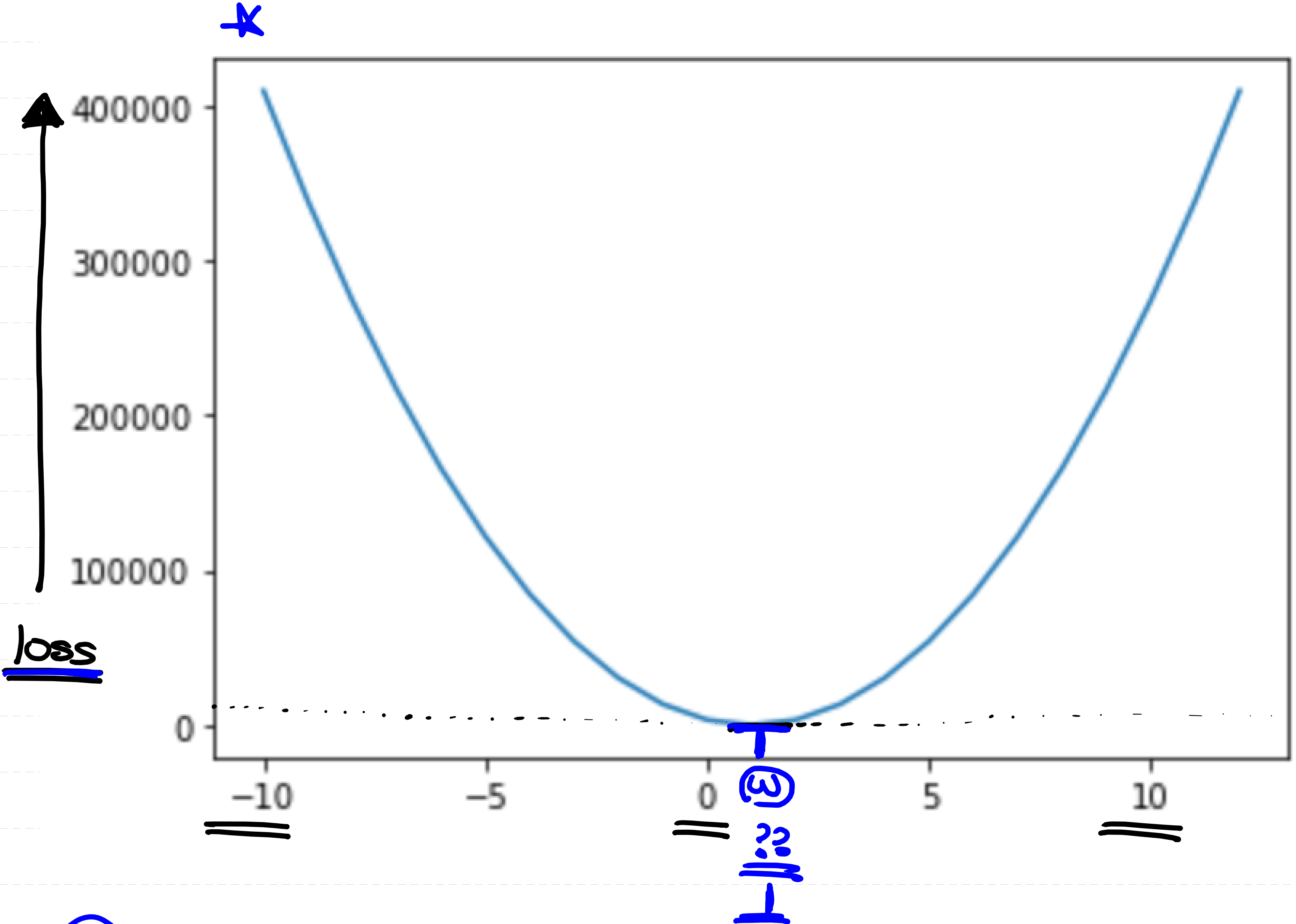
$$E(w, b) = \frac{1}{n} \sum_{i=1}^{n} [t_i - (wx_i + b)]^2 \Rightarrow \text{loss function}$$

최소로 만드는

w, b를 찾는거예요!!

모양은 좀

알아보아요!!




* loss는 w, b의 함수

실제 값을 이용해서 loss function의 모양 확인!

⇓

양의 포물선

→ ω

loss function의 값이 최소가 되게하는 W를 구하려면 어떻게 해야 하나요 ??

(경사하강법) Gradient Descent Alogrithm 이용

# ★ gradient descent ★

① $w$ 값은 랜덤하게 설정

② $w$에서 편미분해서 값을 구함.

③ $w' = w - \alpha \frac{\partial E(w,b)}{\partial w}$

숫자

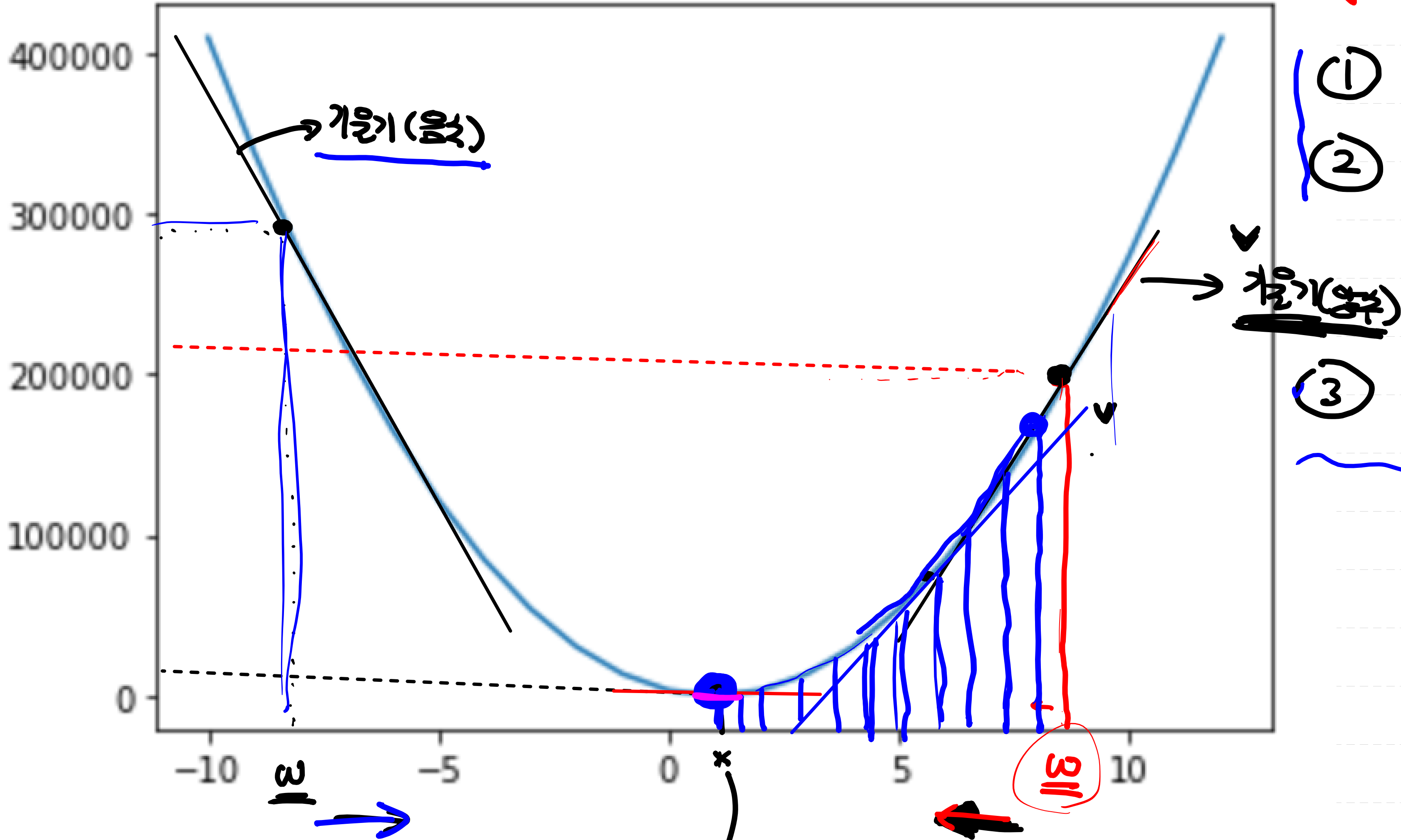
1000

learning rate

학습률

우리가 설정하는값

< 0.001 >




접선의 기울기 "0"

$$\boxed{y = wx + b}$$

정리해 보면,

독립변수가 1개인 "Simple Linear Regression"의 목표는

Training Data Set의 특정값 분포를 잘 나타내는 임의의 직선인

$y = Wx + b$를 구하는 것이고 결국 $(W, b)$를 구하는게 목표.

→ MSE(평균제곱오차)를 이용한 loss function을 만들었어요

* loss function을 최소로 만드는 $W, b$를 구하려고 해요!

$W, b$를 구하기위해 $E(W, b)$를 $W, b$에 대해 편미분해요

편미분한값(접선의 기울기) × learning rate ⇒ 이 값을 $W, b$에서 (학습률) 차감

↓

더 좋은 $W, b$가 계산

반복해서 최적의 $W, b$를 찾아요!

gradient Descent 알고리즘.

★★

(Model)

$y = Wx + b$

$\overset{w,b}{\hookrightarrow}$ 처음은 random

(loss Function)

$$E(w,b) = \frac{1}{n}\sum_{i=1}^{n}\left[t_i - (wx_i + b)\right]^2$$

| 시간(x) | 성적(t) |
|---|---|
| 1 | 1 |
| 2 | 10 |
| 5 | 30 |
| 8 | 70 |
| 10 | 90 |

* Training Data set

⇒ Learning: loss Function의 값 계산 ⇒ 결과값 ⇒ 계산된 loss 값이 최적인가요? (최소) ⇒ Y ⇒ 학습종료

N ⇒ Weight, bias update

$$w' = w - \alpha \frac{\partial E(w,b)}{\partial w}, \quad b' = b - \alpha \frac{\partial E(w,b)}{\partial b}$$