선형 회귀(linear regression) - 가장 훌륭한 예측선 긋기

선의 방향을 잘 정하면 그 선을 따라가는 것만으로도 보이지 않는 미래의 것을 예측할 수 있음!!

1. 선형 회귀의 정의

"학생들의 중간고사 성적이 [(정보)]에 따라 다 다르다."

머신러닝과 딥러닝은 정보가 필요 -> 정보가 있으면 성적을 예측하는 방정식을 만들 수 있음

'정보' -> x -> **독립 변수**
'성적' -> y -> **종속 변수**

x 값이 변함에 따라 y 값도 변한다!!

선형 회귀 - 독립변수 x를 사용해 종속 변수 y의 움직임을 예측하고 설명하는 작업

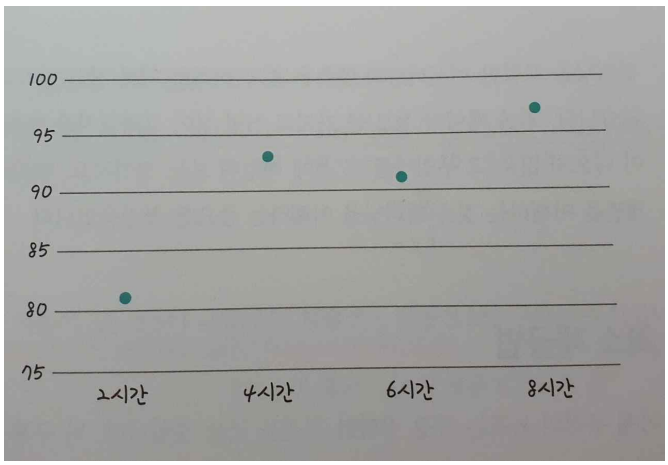
하나의 x값만으로도 y값을 설명할 수 있을 때 -> 단순 선형 회귀(simple linear regression)
여러 개의 x값으로 y값을 설명할 때 -> 다중 선형 회귀(multiple linear regression)

2, 가장 훌륭한 예측선이란?

성적 결정하는 요소를 '공부한 시간' 한 가지만 놓고 생각해 보기

공부한 시간	2시간	4시간	6시간	8시간
성적	81점	93점	91점	97점

$X = \{2, 4, 6, 8\}$ $Y = \{81, 93, 91, 97\}$



왼쪽이 아래로 향하고 오른쪽이 위를 향하는 일종의 '선형(직선으로 표시될 만한 형태)'을 보임.

선형 회귀를 공부하는 과정은 이 점들의 특징을 가장 잘 나타내는 선을 그리는 과정과 일치!!

여기서 선은 직선 즉 일차 함수 그래프로 나타남 ($y = ax + b$)
정확하게 계산하려면 상수 a(직선의 기울기)와 b(y절편)의 값을 알아야 함
선형 회귀는 최적의 a 값과 b 값을 찾아내는 작업

정확한 직선을 그어 놓았다면 이를 통해 위 표에 나와 있지 않은 또 다른 학생의 성적을 예측할 수 있음

(딥러닝을 포함한 머신러닝의 예측은 결국 이러한 기본 접근 방식과 크게 다르지 않음)

3. 최소 제곱법(method of least squares)

최소 제곱법을 통해 일차 함수의 기울기 a와 y 절편 b를 바로 구할 수 있음
(주어진 x의 값이 하나일 때 적용 가능. 여러 개의 x가 주어지는 경우에는 경사 하강법 적용)

$$a = \frac{(x - x\text{평균})(y - y\text{평균})\text{의 합}}{(x - x\text{평균})^2\text{의 합}}$$

$$b = y\text{의 평균} - (x\text{의 평균} \times \text{기울기}\, a)$$

ex) 공부한 시간(x) 평균: (2 + 4 + 6 + 8) ÷ 4 = 5

성적(y) 평균: (81 + 93 + 91 + 97) ÷ 4 = 90.5

$$a = \frac{(2-5)(81-90.5)+(4-5)(93-90.5)+(6-5)(91-90.5)+(8-5)(97-90.5)}{(2-5)^2+(4-5)^2+(6-5)^2+(8-5)^2} = \frac{46}{20} = 2.3$$

$$b = 90.5 - (2.3 \times 5) = 79$$

$$y = 2.3x + 79$$

공부한 시간	2	4	6	8
성적	81	93	91	97
예측 값	**83.6**	**88.2**	**92.8**	**97.4**

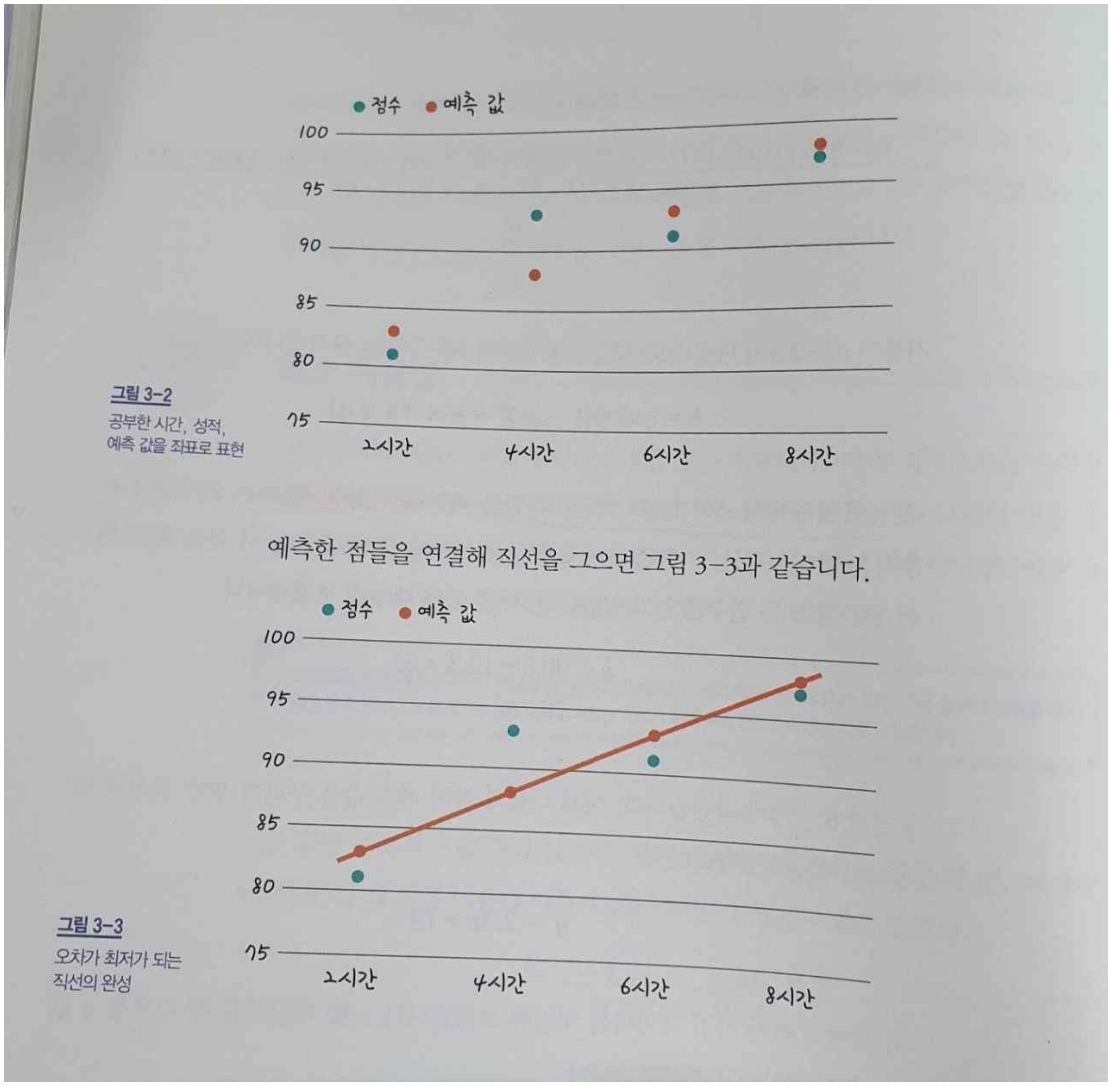


그림 3-2
공부한 시간, 성적, 예측 값을 좌표로 표현

예측한 점들을 연결해 직선을 그으면 그림 3-3과 같습니다.

그림 3-3
오차가 최저가 되는 직선의 완성

주황색 선이 바로 오차가 가장 적은, 주어진 좌표의 특성을 가장 잘 나타내는 직선!!

4. 코딩으로 확인하는 최소 제곱

(실습코드는 github 참고)

5. 평균 제곱 오차

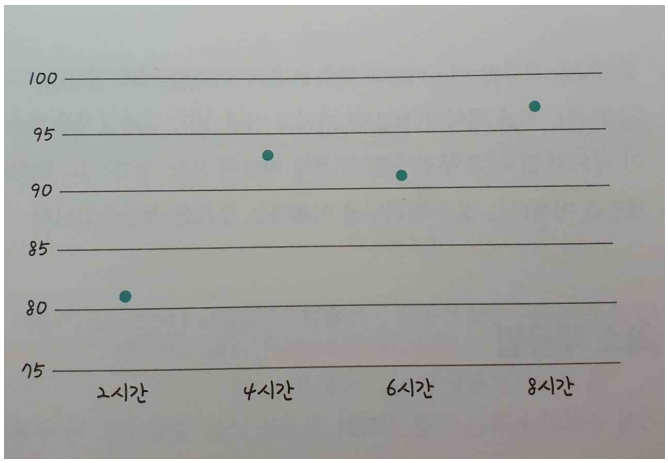
최소 제곱법으로만으로 모든 상황을 해결하기는 어려움. 여러 개의 입력을 처리하기에는 무리가 있기 때문이다. 딥러닝은 대부분 입력 값이 여러 개인 상황에서 이를 해결하기 위해 실행됨.

여러 개의 입력값을 계산할 때는 임의의 선을 그리고 난 후, 이 선이 얼마나 잘 그려졌는지를 평가하여 조금씩 수정해 가는 방법을 사용 (오차 평가 알고리즘) -> **평균 제곱 오차(mean square error, MSE)**

6. 잘못 그은 선 바로잡기

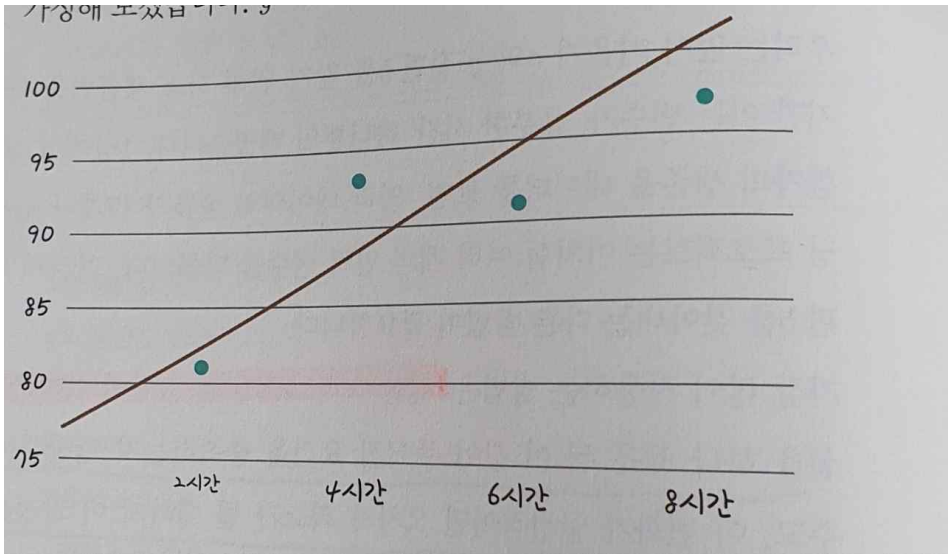
입력 데이터가 많을 때 주로 사용하는 방법은 **'일단 그리고 조금씩 수정해 나가기'** 방식이다. 가설을 하나 세운 뒤 이 값이 주어진 요건을 충족하는지 판단하여 조금씩 변화를 주고, 이 변화가 긍정적이면 오차가 최소가 될 때까지 이 과정을 계속 반복하는 방법이다.

선을 긋고 나서 수정하는 과정에서는 '나중에 그린 선이 먼저 그린 선보다 더 좋은지 나쁜지를 판단하는 방법'이 필요!

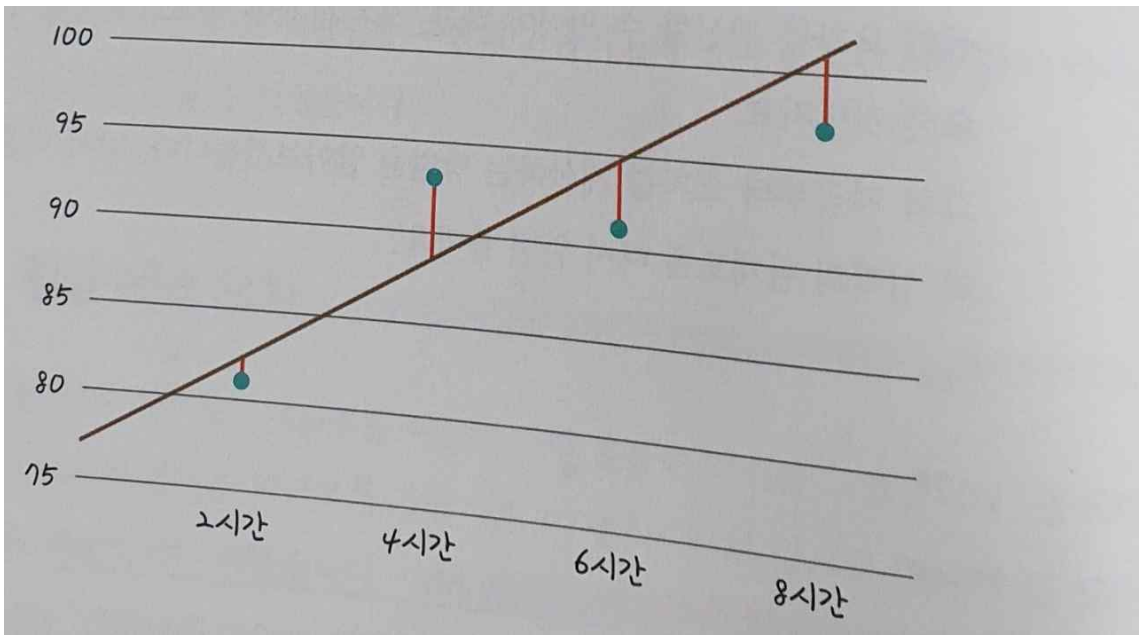


이번에는 최소 제곱법을 사용하지 않고 아무 값이나 a와 b에 대입해 본다. 임의의 값을 대입한 뒤 오차를 구하고 이 오차를 최소화하는 방식을 사용해서 최종 a와 최종 b의 값을 구해보자.

(a = 3, b = 76인 경우)

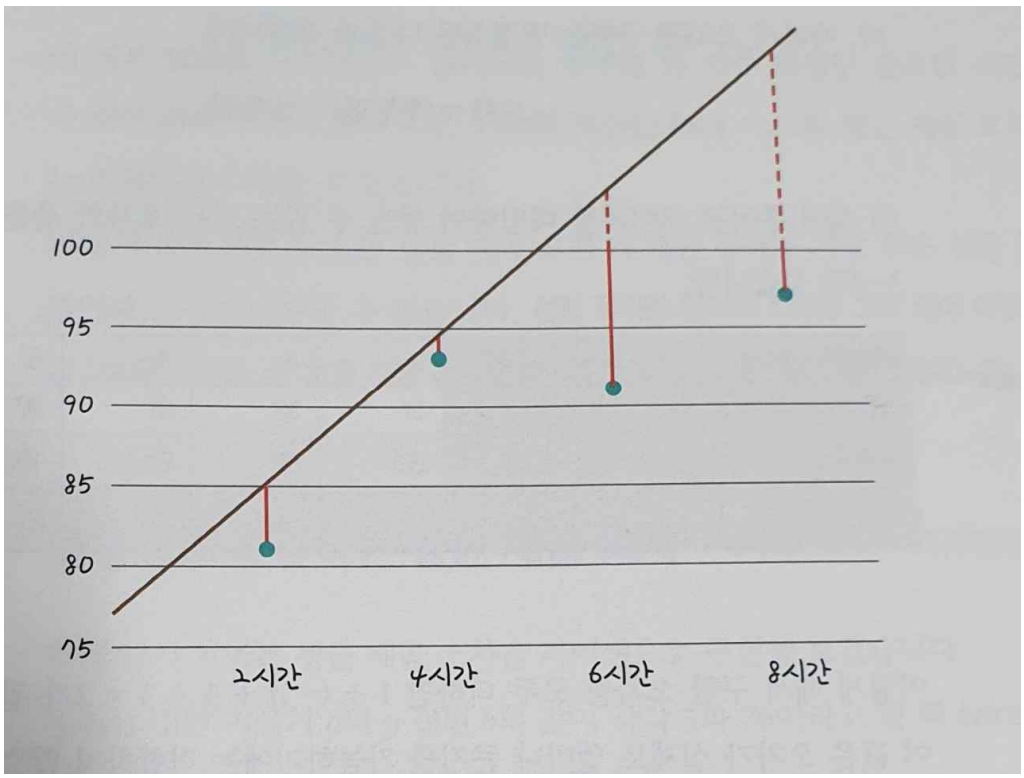


임의의 직선과 각 점 사이의 거리를 재 오차를 측정

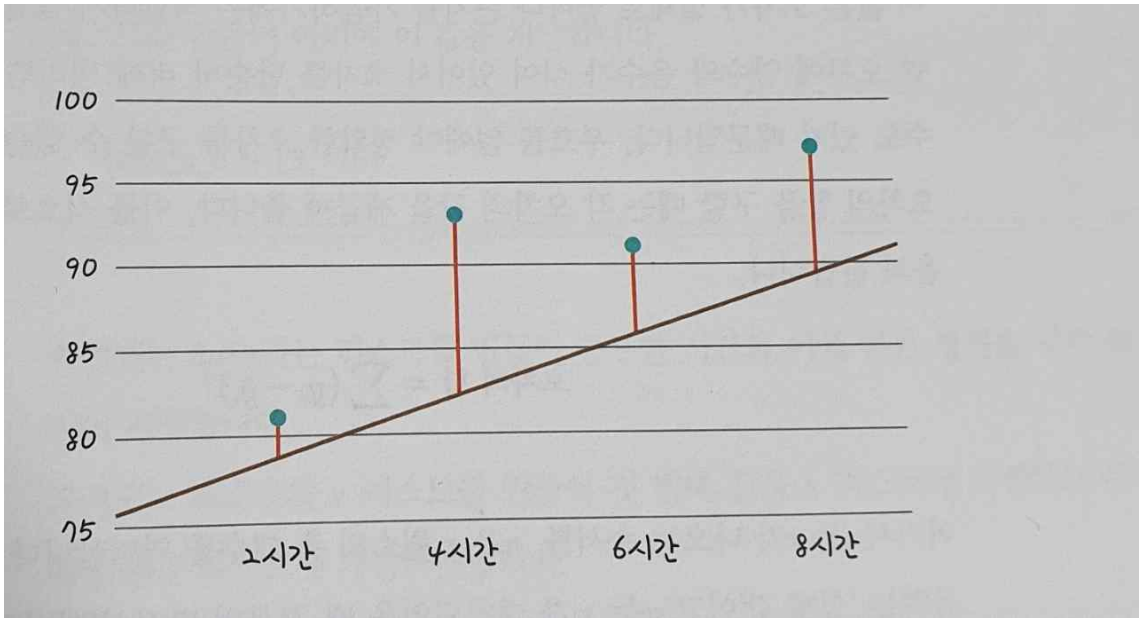


빨간색 선들의 합이 작을수록 잘 그어진 직선이고 합이 클수록 잘못 그어진 직선이 된다.

(기울기를 너무 크게 잡았을 때의 오차)



(기울기를 너무 작게 잡았을 때의 오차)



그래프의 기울기가 잘못되었을수록 빨간색 선의 거리의 합, 즉 오차의 합도 커진다.

오차 = 실제 값 - 예측 값

공부한 시간(x)	2	4	6	8
성적(실제 값, y)	81	93	91	97
예측 값	82	88	94	100
오차	1	-5	3	3

단순히 오차를 그대로 더하는 것으로 오차가 얼마나 큰지 가늠할 수 없음. 왜냐하면 양수와 음수가 섞여 있기 때문이다. -> 부호를 없애야 함 -> 각 오차의 값을 제곱

$$\text{오차의 합} = \sum_{i}^{n}(y_i - \hat{y_i})^2 \quad y_i \text{ (실제 값)} \quad \hat{y_i} \text{ (예측 값)}$$

오차의 합에 이어 각 x 값의 평균 오차를 이용 -> **평균 제곱 오차(Mean Squared Error, MSE)**

$$\text{평균 제곱 오차} = \frac{1}{n}\sum(y_i - \hat{y_i})^2$$

즉, 선형 회귀란 임의의 직선을 그어 이에 대한 평균 제곱 오차를 구하고, 이 값을 가장 작게 만들어 주는 a와 b 값을 찾아가는 작업이다.

7. 코딩으로 확인하는 평균 제곱 오차
(실습코드는 github 참고)

이제 남은 것은 오차를 줄이면서 새로운 선을 긋는 것 -> a,b 값을 적절히 조절하면서 오차의 변화를 살펴보고, 그 오차가 최소화되는 a와 b의 값을 구해야 함 (**경사 하강법**)