

03장 머신러닝의 기초를 다집니다

— 수치 예측

Draft

- 03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다
- 03-2 경사 하강법으로 학습하는 방법을 알아봅니다
- 03-3 손실 함수와 경사 하강법의 관계를 알아봅니다
- 03-4 선형 회귀를 위한 뉴런을 만듭니다

Draft

- 03-1 선형 회귀에 대해 알아보고 데이터를 준비합니다
- **03-2 경사 하강법으로 학습하는 방법을 알아봅니다**
- 03-3 손실 함수와 경사 하강법의 관계를 알아봅니다
- 03-4 선형 회귀를 위한 뉴런을 만듭니다

03-2 경사 하강법으로 학습하는 방법을 알아보니다

그래프로 경사 하강법의 의미를 알아보시다

선형 회귀와 경사 하강법의 관계

선형 회귀의 목표

입력 데이터(x)와 타깃 데이터(y)를 통해 기울기(a)와 절편(b)을 찾는 것

경사 하강법

모델이 데이터를 잘 표현할 수 있도록 기울기(변화율)를 사용하여 모델을 조금씩 조정하는 최적화 알고리즘

경사 하강법 : 데이터로 모델을 훈련하기 위한 핵심 최적화 알고리즘

03-2 경사 하강법으로 학습하는 방법을 알아봅니다

그래프로 경사 하강법의 의미를 알아봅시다

예측값과 변화율

예측값

모델을 통해 예측한 값(ex. $y=7x+4$ 라는 모델의 x 에 7을 넣으면 53이 예측 됨)

변화율

모델의 가중치와 절편을 **효율적**으로 업데이트시키기 위한 값

03-2 경사 하강법으로 학습하는 방법을 알아봅니다

그래프로 경사 하강법의 의미를 알아봅시다

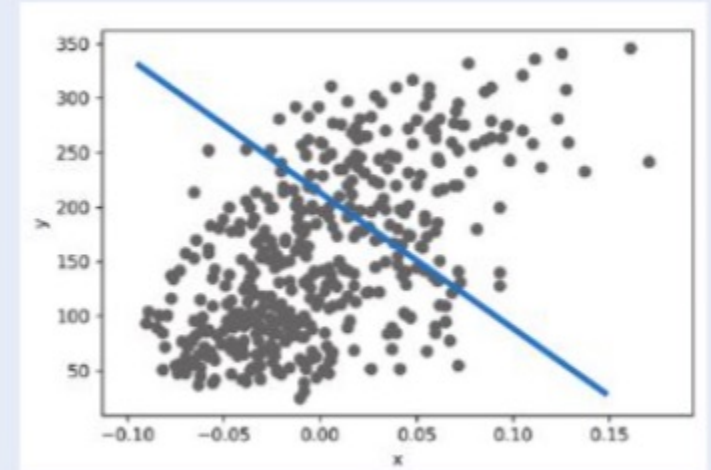
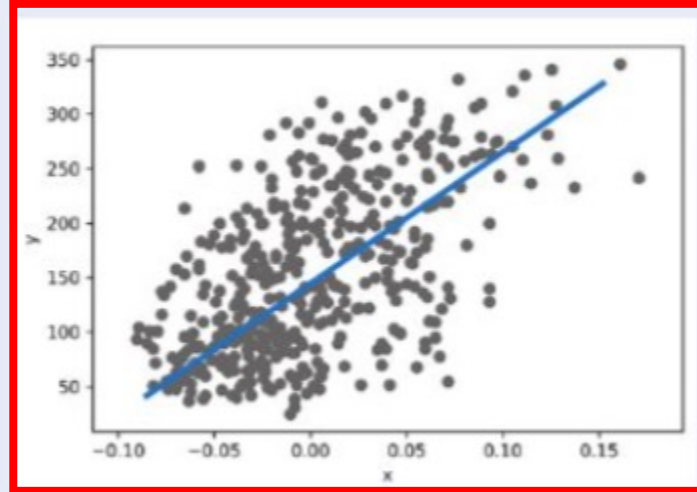
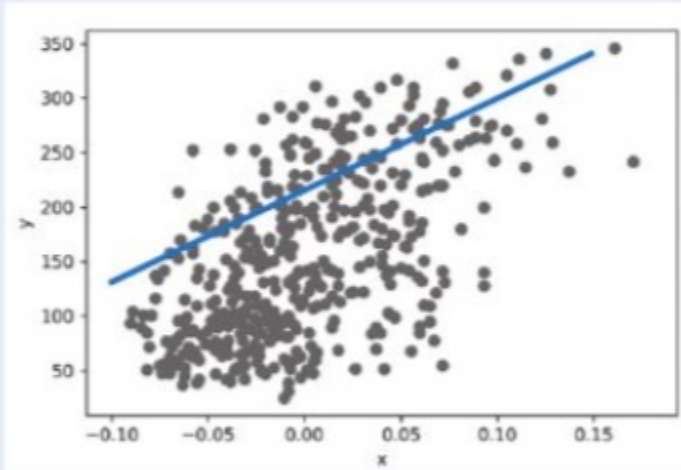
경사 하강법

모델이 데이터를 잘 표현할 수 있도록 기울기(변화율)를 사용하여 모델을 조금씩 조정하는 최적화 알고리즘

- 1. Heuristic
 - 2. 변화율
 - 3. 오차역전파
- } 실습

03-2 경사 하강법으로 학습하는 방법을 알아봅시다 그래프로 경사 하강법의 의미를 알아봅시다

산점도에 가장 잘 맞는 직선을 그린다면 어떤 모습일까?
다음 그림 중 어느 직선이 데이터를 가장 잘 표현하고 있을까?

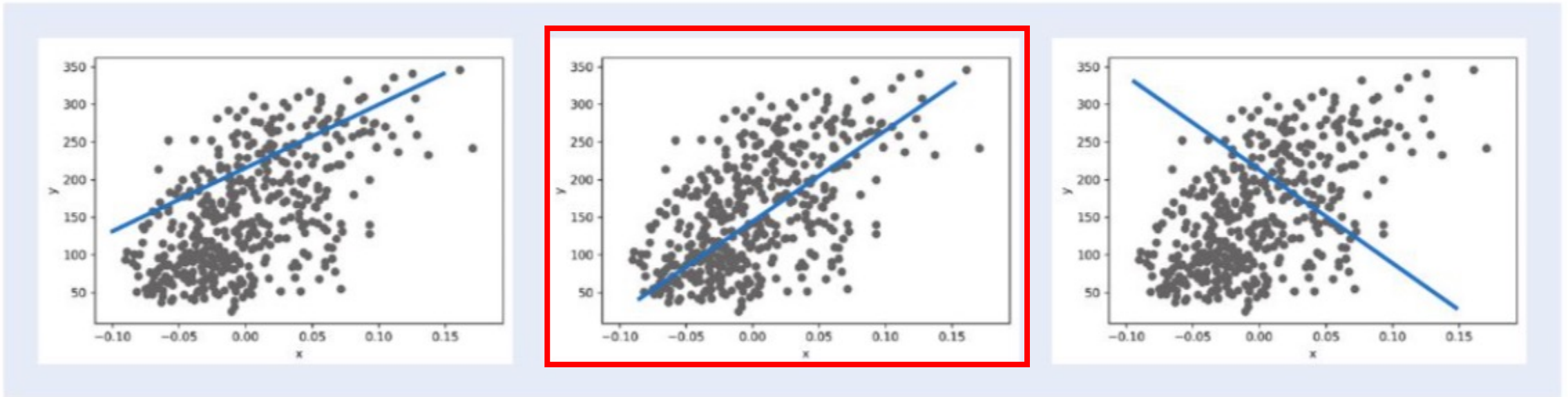


가운데 그림이 가장 데이터를 잘 표현하고 있는 직선!!

03-2 경사 하강법으로 학습하는 방법을 알아보시다 그래프로 경사 하강법의 의미를 알아보시다

- 위 그래프는 입력 데이터 1개의 특성에 대한 타깃 데이터를 그림으로 표현
- 만약 10개의 특성으로 그래프를 그린다면? -> 고차원 공간을 가로지르는 초평면은 상상하기도 어려움
- 보통은 1, 2개의 특성을 사용하여 2차원이나 3차원 그래프로 그리는 경우가 많음

=>알고리즘에 대한 직관을 쉽게 얻을 수 있음
(낮은 차원에서 얻은 직관은 높은 차원으로 확장 될 수 있음)



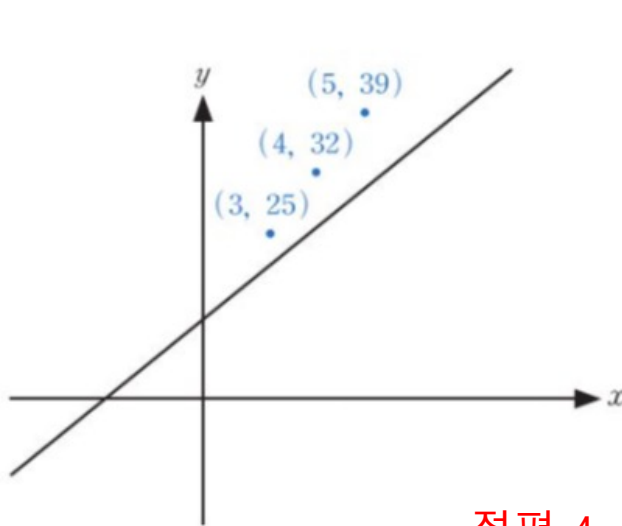
가운데 그림이 가장 데이터를 잘 표현하고 있는 직선!!

03-2 경사 하강법으로 학습하는 방법을 알아봅시다

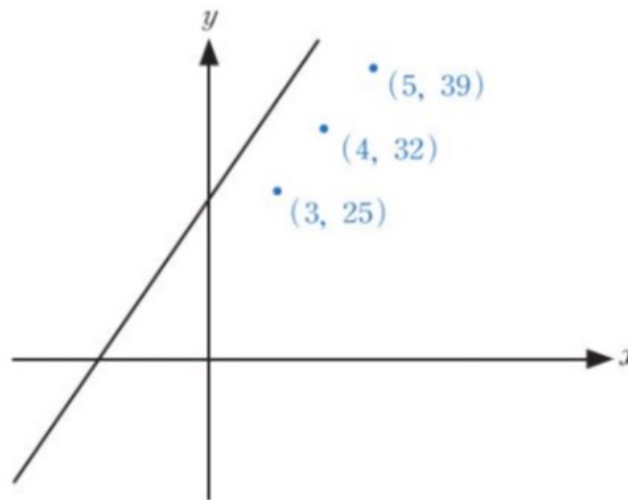
그래프로 경사 하강법의 의미를 알아봅시다

선형회귀와 경사 하강법의 관계 이해

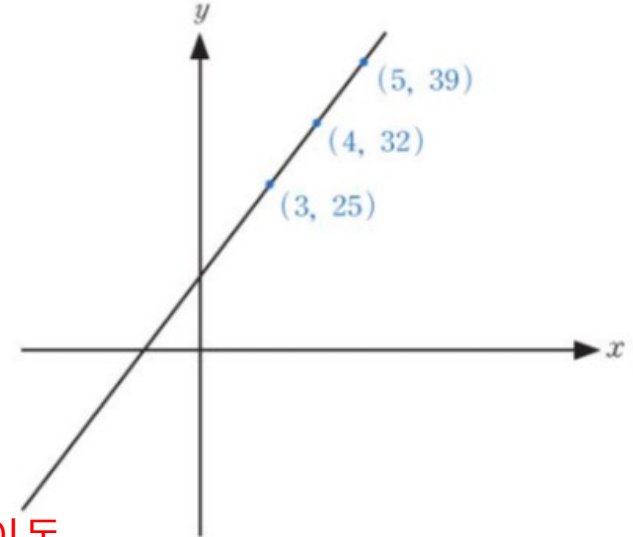
- 선형 회귀의 목표는 입력 데이터와 타겟 데이터를 통해 기울기와 절편을 찾는 것
- 즉, 산점도 그래프를 잘 표현하는 직선의 방정식을 찾는 것이 회귀 알고리즘의 목표
- 경사 하강법이 바로 그 방법 중 하나
- 경사 하강법이란, 모델이 데이터를 잘 표현할 수 있도록 기울기(변화율)를 사용하여 모델을 조금씩 조정하는 최적화 알고리즘
- 회귀의 한 방법인 경사 하강법은 많은 양의 데이터를 처리하기에 좋은 알고리즘
- 회귀 문제를 푸는 알고리즘은 정규 방정식, 결정 트리, 서포트 벡터 머신 등 아주 많음



절편 4 -> 5 이동
기울기 6 -> 7 이동



절편 5 -> 4 이동
기울기 유지



03-2 경사 하강법으로 학습하는 방법을 알아보니다

예측값과 변화율에 대해 알아보시다

예측값이란?

세타

- 딥러닝 분야에서는 기울기 a 를 종종 가중치를 의미하는 w 나 계수를 의미하는 θ 로 표기
- y 는 \hat{y} 으로 표기하여 읽을 때는 와이-햇이라고 읽음
- 즉, 앞으로는 $y = ax + b$ 로 알고 있던 모델을 $\hat{y} = wx + b$ 로 이해하기
- 가중치 w 와 절편 b 는 알고리즘이 찾은 규칙이고 \hat{y} 는 우리가 예측한 값을 의미
- 예측값이란 모델에 새로운 입력값을 넣으면 어떤 출력이 나오는데 이 값이 모델을 통해 예측한 값
- 예를 들어 $y = 7x + 4$ 라는 모델의 에 7을 넣으면 53이 나오는데 이 값이 예측 값

$$y = ax + b \rightarrow \hat{y} = wx + b$$

예측값의 의미를 이해했으니 예측값으로 어떻게 모델을 조정할 수 있는지 알아보자

03-2 경사 하강법으로 학습하는 방법을 알아봅시다 예측값으로 올바른 모델 찾기

- 아래 식에서 우리가 찾고 싶은 것은 훈련 데이터 (x, y) 에 잘 맞는 w 와 b

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

- w 와 b 를 찾기 위한 방법은 다음과 같음

훈련 데이터에 잘 맞는 w 와 b 를 찾는 방법

① 무작위로 w 와 b 를 정합니다(무작위로 모델 만들기).

② x 에서 샘플 하나를 선택하여 \hat{y} 를 계산합니다(무작위로 모델 예측하기).

③ \hat{y} 와 선택한 샘플의 진짜 y 를 비교합니다(예측한 값과 진짜 정답 비교하기, 틀릴 확률 99%).

④ \hat{y} 이 y 와 더 가까워지도록 w , b 를 조정합니다(모델 조정하기).

⑤ 모든 샘플을 처리할 때까지 다시 ②~④ 항목을 반복합니다.

조정시 휴리스틱, 변화율, 오차역전파를 활용하여
기울기 조정 가능

- 위의 방법은 아주 직관적
- 4번의 경우 단순한 모델에서는 부호를 보고 이 커지거나 작아지는 방향을 짐작하기 쉬움
- 하지만 모델이 복잡할수록 체계적인 방법이 필요
- 첫번째 방법대로 모델을 만들어보고 체계적인 방법을 소개

03-2 경사 하강법으로 학습하는 방법을 알아봅시다

예측값으로 올바른 모델 찾기

훈련데이터에 맞는 w 와 b 찾기

w 와 b 가 바뀌었을 때 \hat{y} 어떻게 변하는지 알아내는 가장 간단한 방법은 실제로 계산을 해보는 것!!

1. w 와 b 초기화하기

- w 와 b 를 무작위로 초기화 => 실수 1.0으로 초기화

```
w = 1.0  
b = 1.0
```

2. 훈련 데이터의 첫 번째 샘플 데이터로 \hat{y} 얻기

- 임시로 만든 모델로 훈련 데이터의 첫 번째 샘플 $x[0]$ 에 대한 \hat{y} 를 계산

선형 회귀 공식

```
y_hat = x[0] * w + b  
print(y_hat)  
1.0616962065186886
```

3. 타겟과 예측 데이터 비교하기

- 첫 번째 샘플 $x[0]$ 에 대응하는 타겟값 $y[0]$ 을 출력하여 y_{hat} 의 값과 비교

진짜 정답

```
print(y[0])  
151.0
```

03-2 경사 하강법으로 학습하는 방법을 알아봅시다

예측값으로 올바른 모델 찾기

훈련데이터에 맞는 w 와 b 찾기

4. w 값 조절하여 예측값 바꿔보기

- 예측한 y_{hat} 은 1.06 / 타겟은 151.0 => 차이가 큼 => 이유 w 와 b 를 무작위 값으로 정했기 때문
- w 와 b 를 좀 더 좋은 방향으로 바꾸려면? => 어떤 방향으로 바꾸면 y_{hat} 이 $y[0]$ 에 가까워질 수 있을까? => 가장 쉬운 방법은 w 와 b 를 조금씩 변경해서 y_{hat} 이 증가하는지 또는 감소하는지 보기

03-2 경사 하강법으로 학습하는 방법을 알아봅시다

예측값으로 올바른 모델 찾기

훈련데이터에 맞는 w 와 b 찾기

실습 1) 방금 배운 코드 돌려보기 (변화율 적용 x)

03-2 경사 하강법으로 학습하는 방법을 알아봅시다

예측값으로 올바른 모델 찾기

훈련데이터에 맞는 w와 b찾기

실습 2) 임의로 w 변경하여 오차 계산해보기 (아래 코드를 실행해보기. Add를 0.1, 1, 100, 1000, 10000으로 변화시켜보기)

```
In [42]: 1 #1 . 훈련 데이터에 맞는 w, b 찾아보기
2 i = 0
3 add = 0.1
4 #add = 0.1, 1, 100, 1000, 10000 로 변화시키면서 y - y_hat_inc 수치 어떻게 변화하는지 확인
5 #사람이 인위적으로 결과값과 예측치가 적어지는 방향으로 가중치를 조절해야함 => 딱히 기준이 없고 노동력이 많이 들
6 while(i < 10):
7     w_inc = w + add
8     y_hat_inc = x[0] * w_inc + b
9     print("updated y_hat_inc : ", y_hat_inc, "y - y_hat_inc : ", y[0]-y_hat_inc)
10     add += 0.1
11     i += 1
```

```
updated y_hat_inc : 1.0678658271705574 y - y_hat_inc : 149.93213417282945
updated y_hat_inc : 1.074035447822426 y - y_hat_inc : 149.92596455217756
updated y_hat_inc : 1.080205068474295 y - y_hat_inc : 149.9197949315257
updated y_hat_inc : 1.0863746891261639 y - y_hat_inc : 149.91362531087384
updated y_hat_inc : 1.0925443097780327 y - y_hat_inc : 149.90745569022198
updated y_hat_inc : 1.0987139304299016 y - y_hat_inc : 149.9012860695701
updated y_hat_inc : 1.1048835510817705 y - y_hat_inc : 149.89511644891823
updated y_hat_inc : 1.1110531717336394 y - y_hat_inc : 149.88894682826637
updated y_hat_inc : 1.1172227923855083 y - y_hat_inc : 149.88277720761448
updated y_hat_inc : 1.123392413037377 y - y_hat_inc : 149.87660758696262
```

03-2 경사 하강법으로 학습하는 방법을 알아봅시다

예측값으로 올바른 모델 찾기

훈련데이터에 맞는 w 와 b 찾기

실습 3) 임의로 b 변경하여 오차 계산해보기

03-2 경사 하강법으로 학습하는 방법을 알아봅시다 변화율로 가중치 업데이트하기

최적의 선형 회귀 모델을 찾기 위해 변화율로 가중치 업데이트해봄. 다른 방법은 없을까? -> 변화율 사용

변화율
계산
공식

```
w_rate = (y_hat_inc - y_hat) / (w_inc - w)
print(w_rate)
```

0.061696206518688734

변화율 예측값 진짜정답 변경후w 변경전w
w가 0.1만큼 증가할때 y는 0.0616만큼 증가

03-2 경사 하강법으로 학습하는 방법을 알아보니다 변화율로 가중치 업데이트하기

- 변화율로 가중치 업데이트하기
- w 가 0.1만큼 증가했을때 y hat이 얼마나 증가했는지?
=> y hat이 증가한 양을 w 가 증가한 양으로 나누면 됨 **변화율**

$$\begin{aligned} w_rate &= \frac{y_hat_inc - y_hat}{w_inc - w} = \frac{(x[0] * w_inc + b) - (x[0] * w + b)}{w_inc - w} \\ &= \frac{x[0] * ((w + 0.1) - w)}{(w + 0.1) - w} = x[0] \end{aligned}$$

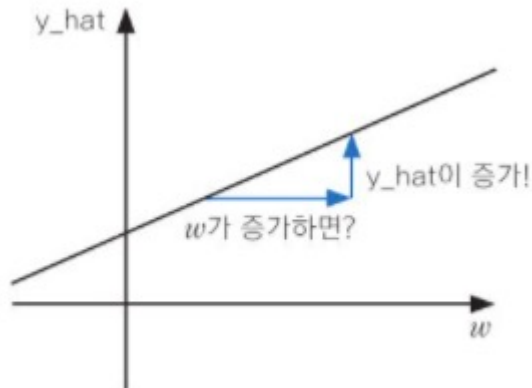
**변화율은 결국 훈련 데이터의
첫 번째 샘플인 $x[0]$ 과 동일!!**

03-2 경사 하강법으로 학습하는 방법을 알아봅시다

변화율로 가중치 업데이트하기

변화율이 양수일 때 가중치를 업데이트하는 방법

- w 와 b 를 변화율로 업데이트하는 방법 (\hat{y} 이 증가해야 y 와 가까워짐)
- 변화율이 0보다 큰 경우를 가정한 그래프



변화율이 0보다 큰 직선의 방정식

- w 증가하면 \hat{y} 증가, w 감소하면 \hat{y} 증가

$$w_{\text{inc}} = w + w_{\text{rate}}$$

03-2 경사 하강법으로 학습하는 방법을 알아봅시다

변화율로 가중치 업데이트하기

변화율이 양수일 때 가중치를 업데이트하는 방법

실습 4) 변화율이 양수일 때 가중치 업데이트 하기

```
In [14]: 1 #2. 변화율로 가중치 업데이트하기 (변화율 양수일때)
          2 w = 1.0
          3 b = 1.0
          4
          5 w_rate = (y_hat_inc - y_hat) / (w_inc-w)
          6 w_inc = w + w_rate
          7 print(w_rate)
          8 y_hat_inc = x[0] * w_inc + b
          9 print("updated y_hat_inc : ", y_hat_inc, "y - y_hat_inc : ", y[0]-y_hat_inc)
          10
```

0.061696206518686264

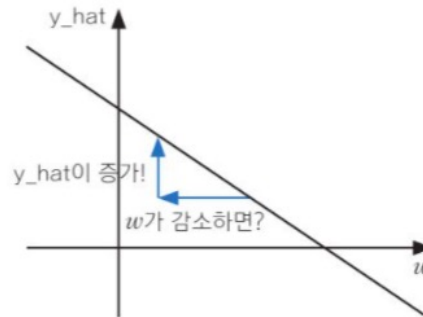
updated y_hat_inc : 1.065502628417485 y - y_hat_inc : 149.93449737158252

03-2 경사 하강법으로 학습하는 방법을 알아봅시다

변화율로 가중치 업데이트하기

변화율이 음수일 때 가중치를 업데이트하는 방법

- 변화율이 0보다 작을 때 \hat{y} 을 증가시키려면 어떻게?



변화율이 0보다 작은 직선의 방정식

- w 가 증가하면 \hat{y} 은 감소, w 가 감소하면 \hat{y} 은 증가
- \hat{y} 을 증가시키려면? $w + w_{rate}$

W 가 1이고
 W_{rate} 가 -0.1이라면
 W 는 0.9로 업데이트 되므로
 \hat{y} 이 증가됨

```
w_new = w + w_rate  
print(w_new)  
1.0616962065186888
```

03-2 경사 하강법으로 학습하는 방법을 알아봅시다 변화율로 절편 업데이트하기

- 절편 b 에 대한 변화율로 b 업데이트
- b 를 0.1만큼 증가시킨 후 y_{hat} 이 얼마나 증가했는지(0.1증가) 계산하고 변화율도 계산 => 변화율은 1

$$b_{\text{rate}} = \frac{y_{\text{hat_inc}} - y_{\text{hat}}}{b_{\text{inc}} - b} = \frac{(x[0] * w + b_{\text{inc}}) - (x[0] * w + b)}{b_{\text{inc}} - b}$$
$$= \frac{(b + 0.1) - b}{(b + 0.1) - b} = 1$$

```
b_inc = b + 0.1
y_hat_inc = x[0] * w + b_inc
print(y_hat_inc)
1.1616962065186887

b_rate = (y_hat_inc - y_hat) / (b_inc - b)
print(b_rate)
1.0
```

- 변화율의 값은 1. 즉 b 가 1만큼 증가하면 y_{hat} 도 1만큼 증가
- 1차함수 그래프에서 절편이 1 증가하면 그래프의 위치가 y 방향으로 1만큼 올라가니 당연한 결과

03-2 경사 하강법으로 학습하는 방법을 알아봅시다 변화율로 절편 업데이트하기

- 즉, b 를 업데이트 하기 위해서는, 변화율이 1이므로 단순히 1을 더하면 됨!

```
b_new = b + 1  
print(b_new)  
2.0
```

- \hat{y} 를 증가시켜야 하는 상황을 가정하고 w 와 b 를 업데이트하는 방법에 대해 알아봄 (수동적 방법)
- 그 이유는?

- \hat{y} 가 y 에 한참 미치지 못 하는 값인 경우, w 와 b 를 더 큰 폭으로 수정할 수 없습니다(앞에서 변화율 만큼 수정을 했지만 특별한 기준을 정하기가 어렵습니다).
- \hat{y} 가 y 보다 커지면 \hat{y} 를 감소시키지 못 합니다.

- 다음 에는 w 와 b 를 더 능동적으로 업데이트하는 방법인 '오차 역전파'에 대해 알아보자

03-2 경사 하강법으로 학습하는 방법을 알아봅니다 변화율로 절편 업데이트하기

실습 5) 변화율로 절편 업데이트하기 -> 공유한 코드 보면서 이해하기

정리

- 경사하강법
 - 훈련데이터에 잘맞는 w 와 b 를 찾는 방법
 1. 휴리스틱
 2. 변화율로 가중치과 절편값 조정
 1. 가중치 업데이트
 2. 절편 업데이트
 3. 오차역전파