

디지털 영상처리 연구실 연구보고서

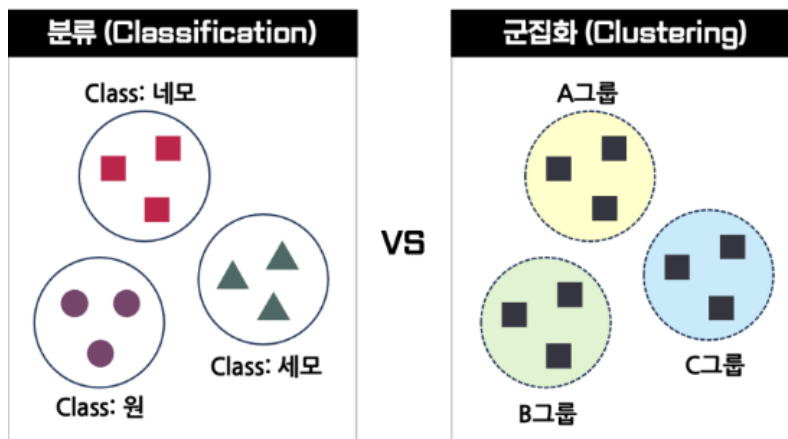
김우현

##머신러닝?

->컴퓨터가 스스로 데이터의 규칙을 찾는 것=학습,훈련

#머신러닝의 학습 방식

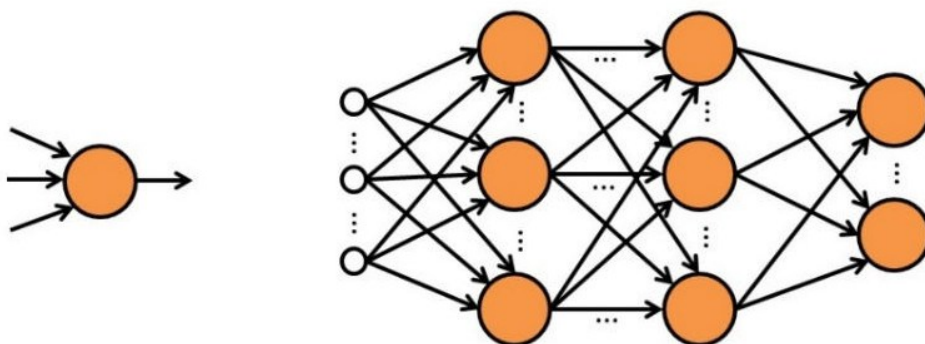
-> 지도학습, 비지도학습



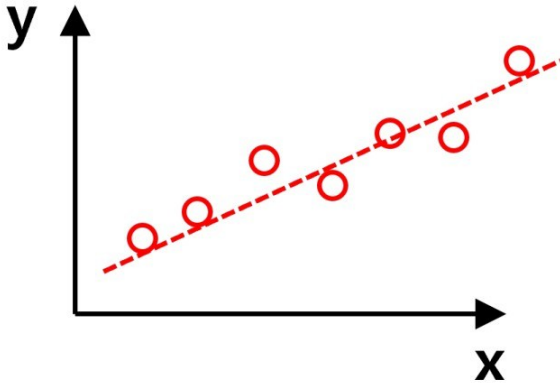
->모델 생성

$$x * 1.5 + 0.1 = y$$

#인공 신경망(딥러닝)



##선형 회귀



$$\rightarrow y = ax + b$$

#경사 하강법?

->기울기(변화율)을 사용하여 모델을 조금씩 조정

$$y^{\wedge} = wx + b$$

#경사하강법을 이용하여 모델 결정

1. 무작위 값으로 w, b 결정
2. 샘플 한 개로 예측 값 확인
3. 예측 값(y^{\wedge})과 원래의 타깃 값(y) 비교
4. 예측 값(y^{\wedge})이 원래의 타깃 값(y)에 가까워지도록 w, b 조정

● 임의의 값 이용해서 모델 생성

$w=1$, $b=1$ 일 때 ($x[0]=0.061$)

$$y^{\wedge} = x[0] * w + b$$

$$y^{\wedge} = 1.061$$

$$y = 151$$

- w값을 0.1증가 하여 예측

$w_inc = w + 0.1$

$y^{\wedge} = x[0] * (w_inc) + b$

$y^{\wedge}_inc = 1.067$

-> 약 0.006 증가

- 예측값 증가정도 확인하기

$w_rate(증가정도) = (y^{\wedge}_inc - y^{\wedge}) / (w_inc - w)$

$= 0.061$

-> $x[0]$ 에 대한 w 의 변화율 $= w_rate = x[0]$

→ 따라서 가중치 w 를 업데이트 하는 방법은 $w + w_rate$ 입니다.

→ 마찬가지로 b 의 값도 같은 방법을 사용하면 b 의 변화율은 1인 것을 확인($b+1$)

#오차 역전파로 업데이트

-> 앞서 했던 방식으로는 예측값이 실제값($y=151$)에 한참 부족하였습니다.

-> $err = y[0] - y^{\wedge}$ 즉 w, b 의 변화율($w_rate, 1$)에 오차를 곱한 다음 w, b 를 업데이트 시켜 줍니다

$err = y[0] - y^{\wedge}$

$w_new = w + w_rate * err$

$b_new = b + 1 * err$

-> $w_new = 10.2$, $b_new = 150.9$

-> 이 방법으로 두번째 샘플 $x[1]$ 값을 이용하여 업데이트 진행

$y^{\wedge} = x[1] * w_new + b_new$

$err = y[1] - y^{\wedge}$

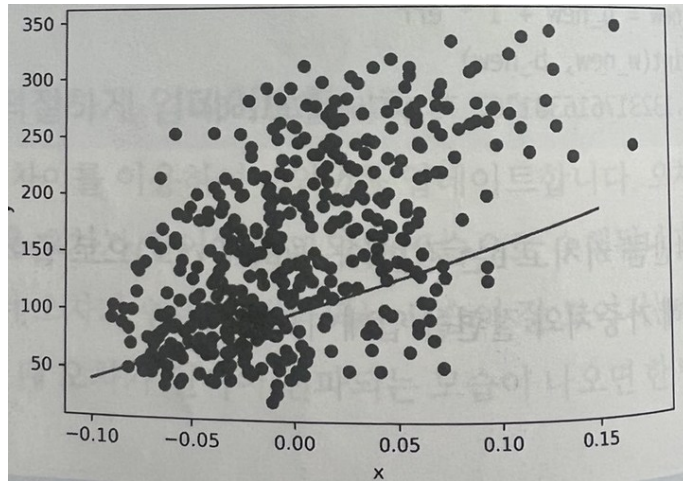
$w_rate = x[1]$

$w_new = w + w_rate * err$

$b_new = b + 1 * err$

-> $w_new = 14.1$, $b_new = 75.5$

- 전체 샘플에 대해 업데이트 진행

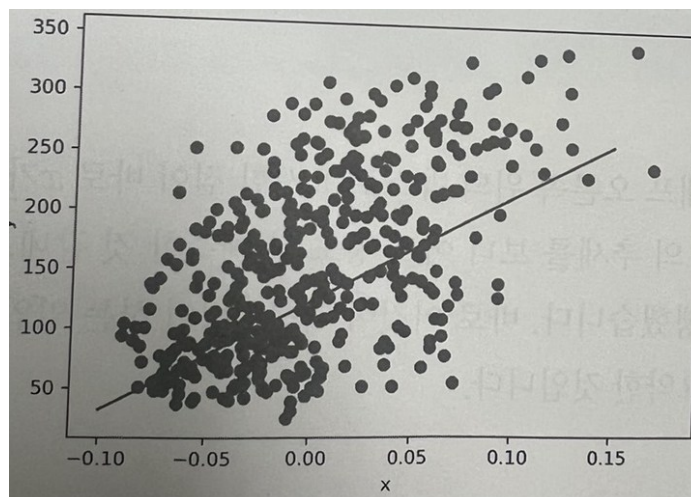


-> $\hat{y} = 587.865x + 99.4$ 의 모델을 얻음

- 여러 에포크 반복

-> 경사 하강법에서는 주어진 훈련 데이터로 여러 번 반복해서 학습=에포크

-> 100번의 에포크로 가중치와 절편 업데이트



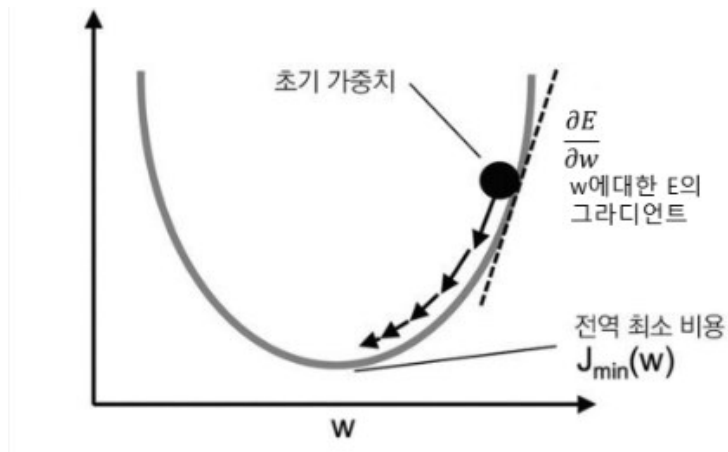
-> $\hat{y} = 913.6x + 123.4$ 의 모델을 최종적으로 얻음

#경사 하강법과 손실함수

- 제공오차함수?

-> $SE = (Y - \hat{Y})^2$

-> 제공오차가 최솟값이 되면 산점도 그래프를 가장 잘 표현한 직선



-> 제공오차를 가중치(w)에 대하여 미분하여 최솟값으로 향함

$\frac{\partial SE}{\partial w} = -2(y - \hat{y})x$ 이므로 가중치 업데이트시 w에 변화율을 빼서 가중치 업데이트!

-> 따라서 오차 역전파에서 사용한 $w_{\text{new}} = w + w_{\text{rate}} * \text{err}(\text{err} = y[0] - \hat{y})$ 은

제공오차를 미분했던 값을 이용한 것입니다