기계학습
02 단항 선형 회귀

SW융합학부 양희경

Ng, Machine Learning, Coursera
오렐리앙 제롱, 핸즈온 머신러닝(사이킷런과 텐서플로를 활용한 머신러닝, 딥러닝 실무), 한빛미디어, 2018.04

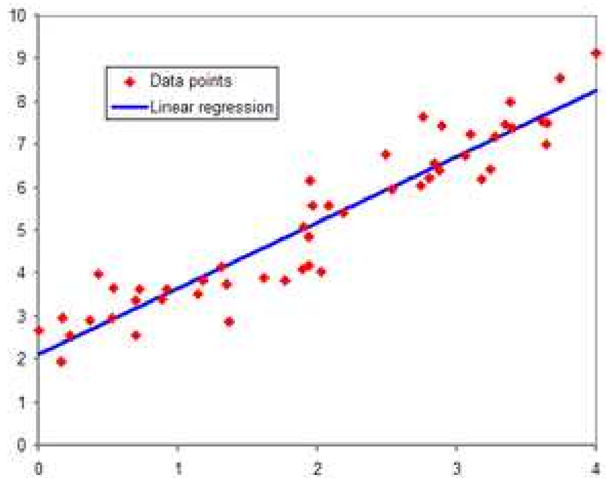
학기 내용

1. 기계 학습 소개
2. **단항 선형 회귀Linear regression with one variable**
3. 다항 선형 회귀Linear regression with multiple variables
4. 로지스틱 회귀(분류)Logistic Regression(Classification)
5. 정규화Regularization
6. Support Vector Machine (SVM)
7. 군집Clustering
8. 차원 축소Dimensionality Reduction

내용

2.1 선형 회귀란?

- 회귀(回歸)Regression
  - 연속적인 종속 변수와 한 개 이상의 독립 변수 사이의 관계를 추정하는 통계적인 과정
    - 종속 변수: **y**, 결과 변수
    - 독립 변수: **x**, (입력) **특성**
    - 관계: 모델model, 가설hypothesis



©Wikipedia

- 회귀의 종류
  - 특성의 개수에 따라
    - 단항 선형 회귀(2장): 특성 개수 한 개
    - 다항 선형 회귀(3장): 특성 개수 두 개 이상
  - 정규화 방법에 따라(5장)
    - 릿지 회귀
    - 라쏘 회귀
    - 엘라스틱넷

2.1 선형 회귀란?

- 선형 회귀
  - 특성의 가중치 합과 편향bias 이라는 상수를 더해 결과 변수를 예측하는 과정
  - (단항) $\hat{y} = h_\theta(x) = \theta_0 + \theta_1 x$
  - (다항) $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$

    $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2$

    $h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \cdots$

- 활용 예
  - 키와 체중의 관계 분석하기
  - 몸무게, 나이, 키로 기대 수명 예측하기
  - 집 면적과 집값의 관계 분석하기
  - 현재 시장 상태와 추가 정보로 미래의 주식 예측하기
  - 유튜브에서 특정 비디오를 보는 사용자의 나이 예측하기

내용

2.1 선형 회귀란?Linear regression

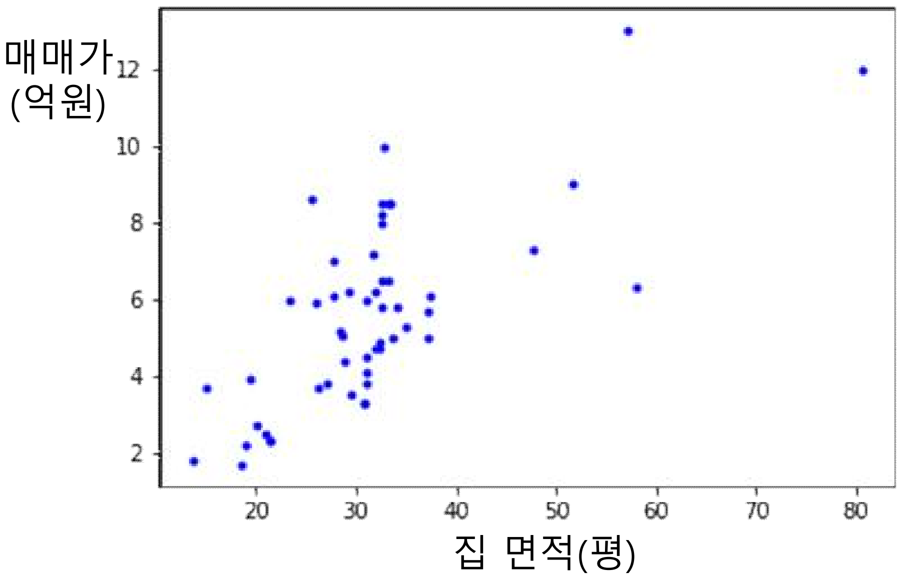
2.2 모델 설계Model representation

2.3 비용 함수Cost function

2.4 경사 하강법Gradient descent

2.2 모델 설계

- 상명대학교 근처 아파트 & 오피스텔 매매가격(57건)
  - 평창동, 신영동, 홍은동, 홍제동 등
  - 2020.2월 기준, 네이버부동산



2.2 모델 설계

- 상명대학교 근처 아파트 & 오피스텔 매매가격(57건)
  - 평창동, 신영동, 홍은동, 홍제동 등
  - 2020.2월 기준, 네이버부동산

- 기호
  - **m** = 훈련 예제training example 개수
  - **x** = "입력" 변수variable/ 특성feature
  - **y** = "출력" 변수/ "타겟" 변수target variable

집 면적(평)	매매가(억원)
27.6	6.1
27.0	3.8
37.0	5.7
13.6	5.7
…	…

m = 57

(x, y) – 훈련 예제 하나

$(x^{(i)}, y^{(i)})$ – i 번째 훈련 예제

$x^{(1)} = \boxed{}$

$x^{(2)} = \boxed{}$

$y^{(1)} = \boxed{}$

2.2 모델 설계

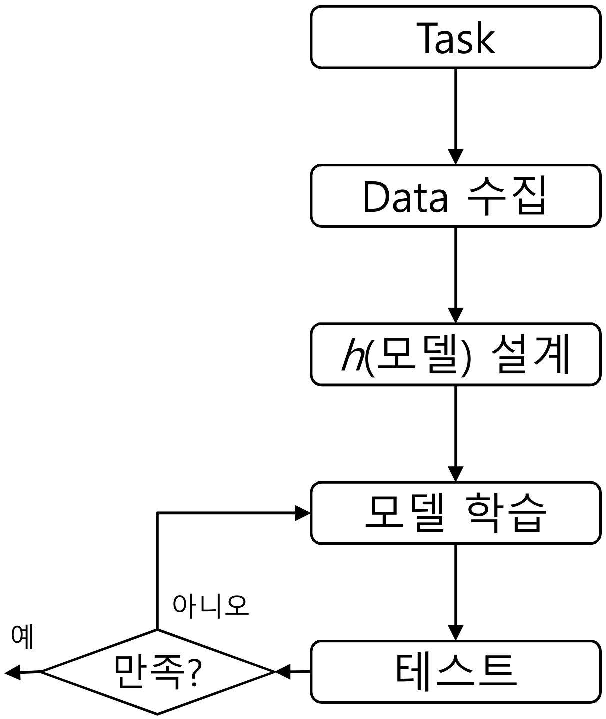
- 입력 변수(특성)가 하나 더 추가된다면?

특성 개수

$m = 57$
훈련 예제 개수

	x_1	x_2
	집 면적(평)	방 개수
$x^{(1)}$	27.6 $x^{(1)}_1$	4 $x^{(1)}_2$
$x^{(2)}$	27.0	5 $x^{(2)}_2$
$x^{(3)}$	37.0 $x^{(3)}_1$	3
	...	...

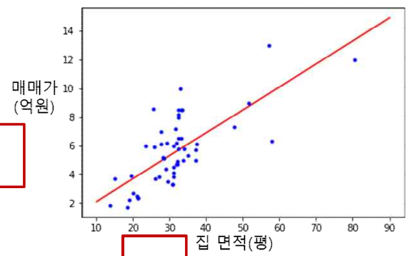
2.2 모델 설계



(집 면적-집 값)의 관계 알아보자.

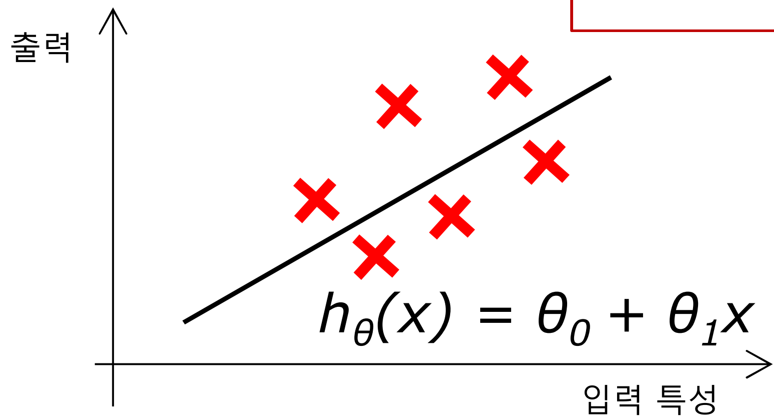
상명대 근처 (평수, 매매가) 데이터 세트 수집

단항 선형 회귀 모델



h 를 어떻게 표현할 것인가?

$h_\theta(x) = \theta_0 + \theta_1 x$, 줄여서 $h(x)$



단변수/단항 선형 회귀

Feature 1 개

Linear regression with one variable
Univariate linear regression

2.2 모델 설계

- 구현 팁
  - $h_\theta(x) = \theta_0 + \theta_1 x$

$$= (\theta_0 \quad \theta_1)\begin{pmatrix}1\\ x\end{pmatrix}$$

2.2 모델 설계

- 훈련 세트

집 면적(평)	매매가(억원)
27.6	6.1
27.0	3.8
37.0	5.7
13.6	5.7
…	…

- Hypothesis: $h_\theta(x) = \theta_0 + \theta_1 x$
  - θ_i: parameters
  - **$\boldsymbol{\theta_i}$ 는 어떻게 선택할까?**

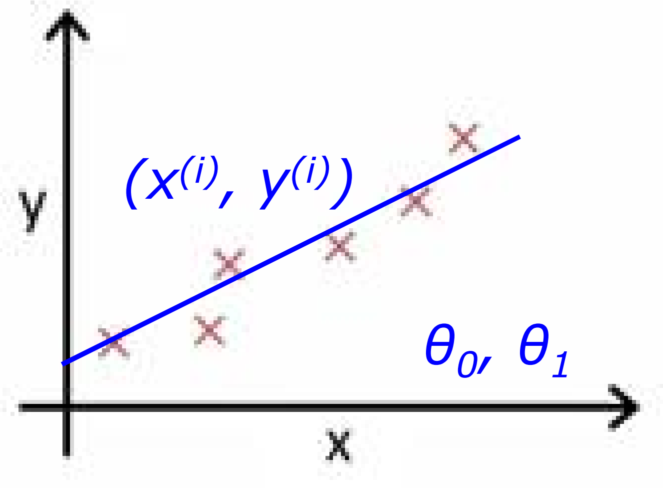
➔ 2.3 비용 함수

내용

2.3 비용 함수

- 비용 함수cost function(손실 함수loss function)
  - 주어진 데이터에 대해, 예측된 값($\hat{y}$) 과 실제 값(y) 사이의 에러를 하나의 실수값으로 수치화한 함수
  - 비용 함수
    - 모델이 틀린 만큼 비용(패널티) 를 준다는 의미에서 직관적인 표현인 '비용'함수라는 이름으로 불림.
  - 손실 함수
    - 모델이 실제 데이터와 차이를 부정적인 의미인 손실로 간주하여 '손실'함수라 불리기도 함
- 예)
  - 주어진 데이터: $y^{(1)}=1$, $y^{(2)}=3$
  - 모델 $y = \theta$ 예측

2.3 비용 함수



아이디어: 학습 예제 (x, y) 의 y와 $h_\theta(x)$가 잘 맞도록 θ_0, θ_1을 선택하자.

참고: $h_\theta(x)$ 는 $\hat{y}$ 이라 쓰기도 함

$$\underset{\theta_0,\theta_1}{\operatorname{argmin}} \; \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$$h_\theta\left(x^{(i)}\right) = \theta_0 + \theta_1 x^{(i)}$$

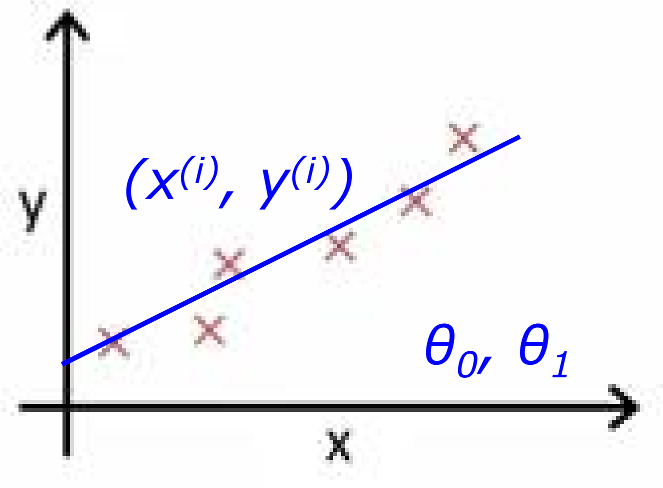
$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$J(\theta_0, \theta_1)$: 비용함수

squared error function

$$\underset{\theta_0,\theta_1}{\operatorname{argmin}} \; J(\theta_0, \theta_1)$$

2.3 비용 함수



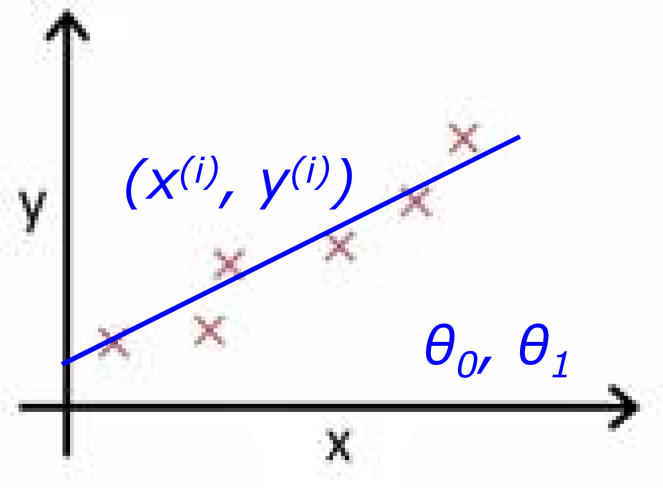
아이디어: 학습 예제 (x, y) 의 y와 $h_\theta(x)$가 잘 맞도록 θ_0, θ_1을 선택하자.

참고: $h_\theta(x)$ 는 $\hat{y}$ 이라 쓰기도 함

$$\underset{\theta_0,\theta_1}{\operatorname{argmin}} \ \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$$h_\theta\left(x^{(i)}\right) = \theta_0 + \theta_1 x^{(i)}$$

2.3 비용 함수



아이디어: 학습 예제 (x, y) 의 y와 $h_\theta(x)$가 잘 맞도록 θ_0, θ_1을 선택하자.

참고: $h_\theta(x)$ 는 $\hat{y}$ 이라 쓰기도 함

$$\underset{\theta_0,\theta_1}{\arg\min} \ \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

$$h_\theta\left(x^{(i)}\right) = \theta_0 + \theta_1 x^{(i)}$$

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

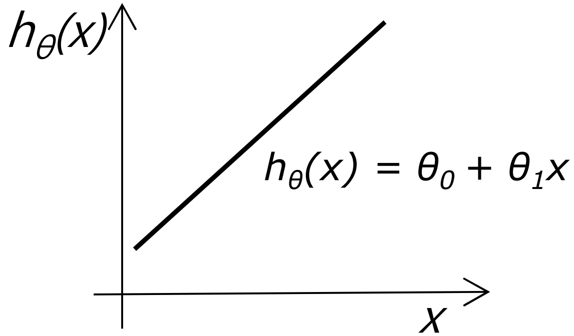
$J(\theta_0, \theta_1)$: 비용함수

squared error function

$$\underset{\theta_0,\theta_1}{\arg\min} \ J(\theta_0, \theta_1)$$

2.3 비용 함수

- 가설Hypothesis : $h_\theta(x) = \theta_0 + \theta_1 x$



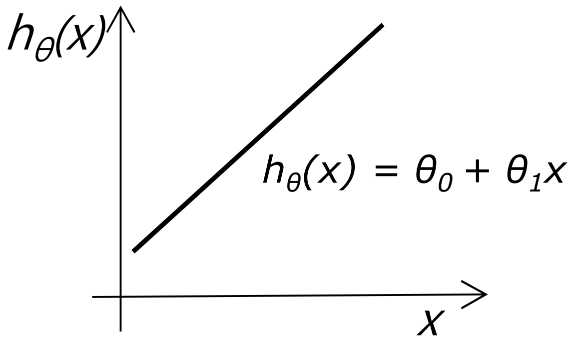
- 비용 함수Cost function : $J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$

- 목표 : $\underset{\theta_0,\theta_1}{\operatorname{argmin}} J(\theta_0, \theta_1)$

2.3 비용 함수

- 가설Hypothesis :

$$h_\theta(x) = \theta_0 + \theta_1 x$$

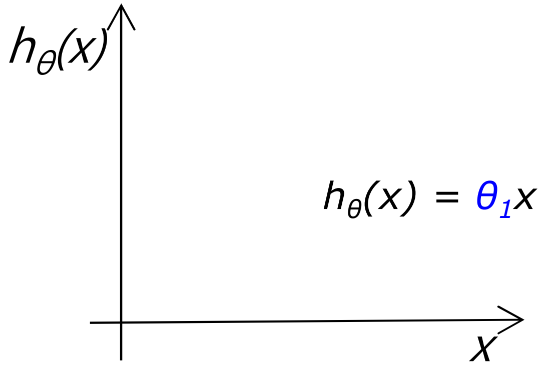


- 비용 함수Cost function :

$$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)^2$$

- 목표 : $\underset{\theta_0,\theta_1}{\arg\min}\ J(\theta_0, \theta_1)$

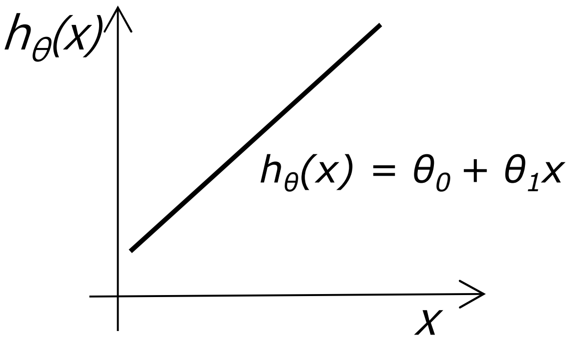
<u>단순화 하면?</u>



2.3 비용 함수

- 가설 Hypothesis :

$$h_\theta(x) = \theta_0 + \theta_1 x$$

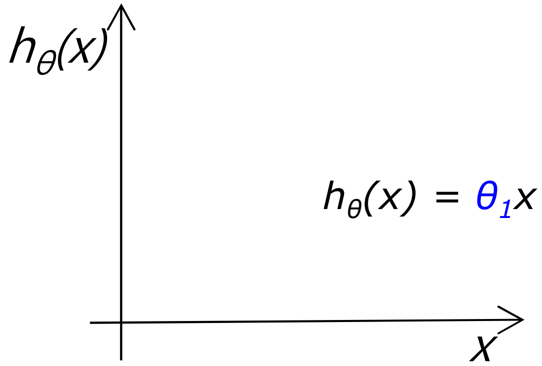


- 비용 함수 Cost function :

$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

- 목표 : $\underset{\theta_0,\theta_1}{\mathrm{argmin}}\ J(\theta_0, \theta_1)$

단순화 하면?



$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

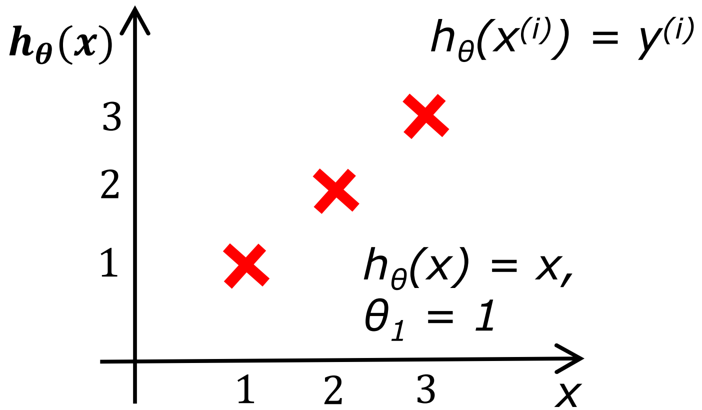
$$\underset{\theta_1}{\mathrm{argmin}}\ J(\theta_1)$$

2.3 비용 함수

- θ_1 만 이용하는 단순한 예

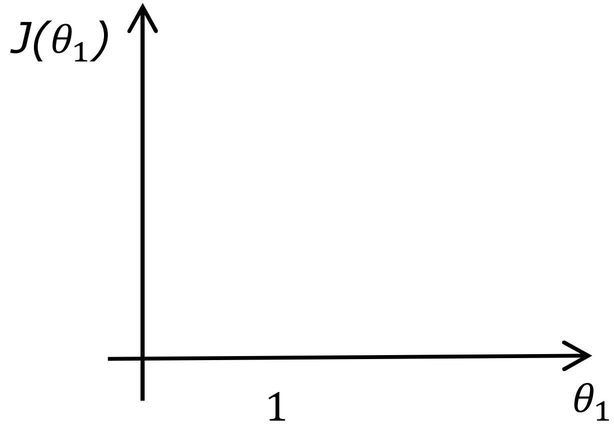
$$\boldsymbol{h_\theta(x)}$$

($\boldsymbol{\theta_1}$고정하면, x에 대한 함수)



$$\boldsymbol{J(\theta_1)}$$

(파라미터 $\boldsymbol{\theta_1}$ 에 대한 함수)



$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$$

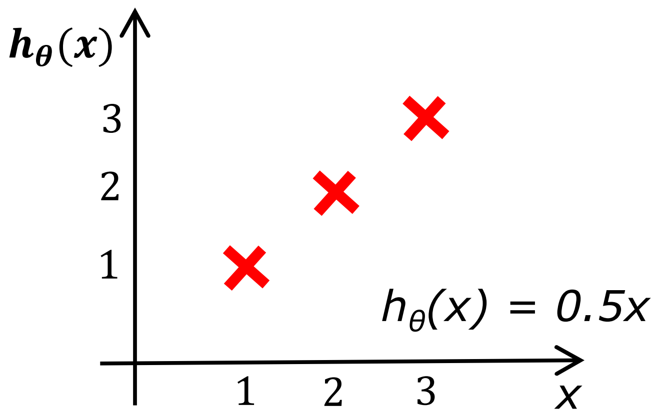
$$= \frac{1}{2m}\sum_{i=1}^{m}\left(\theta_1 x^{(i)} - y^{(i)}\right)^2$$

2.3 비용 함수

- <u>θ_1 만 이용하는 단순한 예</u>

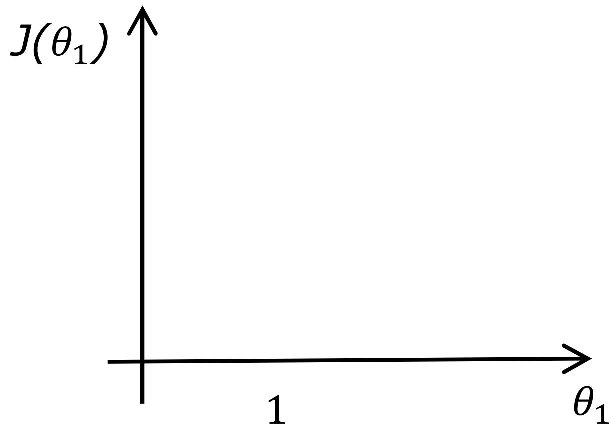
$h_\theta(x)$

($\boldsymbol{\theta_1}$고정하면, x에 대한 함수)



$J(\theta_1)$

(파라미터 $\boldsymbol{\theta_1}$ 에 대한 함수)



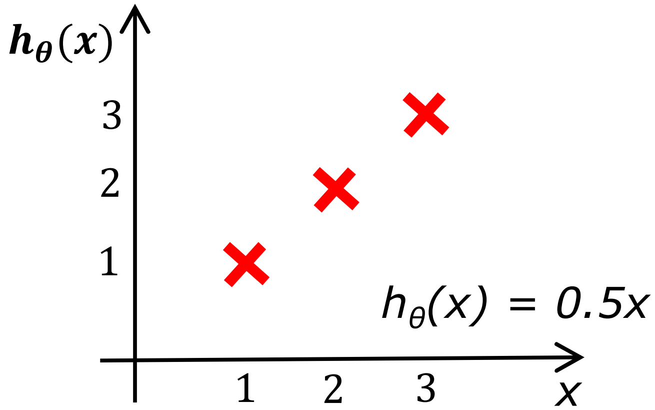
$J(0.5) =$

2.3 비용 함수

- θ_1 만 이용하는 단순한 예

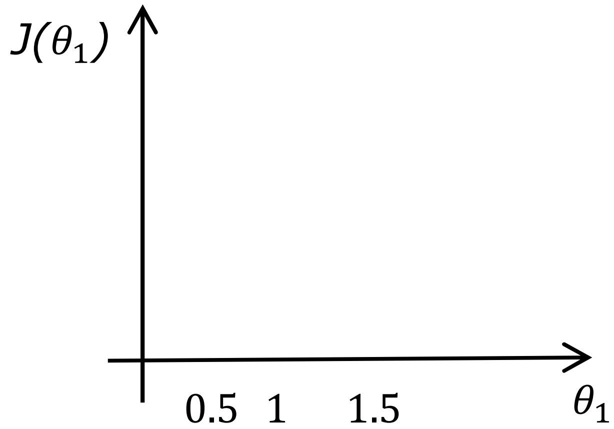
$$h_{\theta}(x)$$

(θ_1고정하면, x에 대한 함수)



$$J(\theta_1)$$

(파라미터 θ_1 에 대한 함수)



$$J(\theta_1)$$

$$\underset{\theta_1}{\operatorname{argmin}} J(\theta_1)$$

2.3 비용 함수

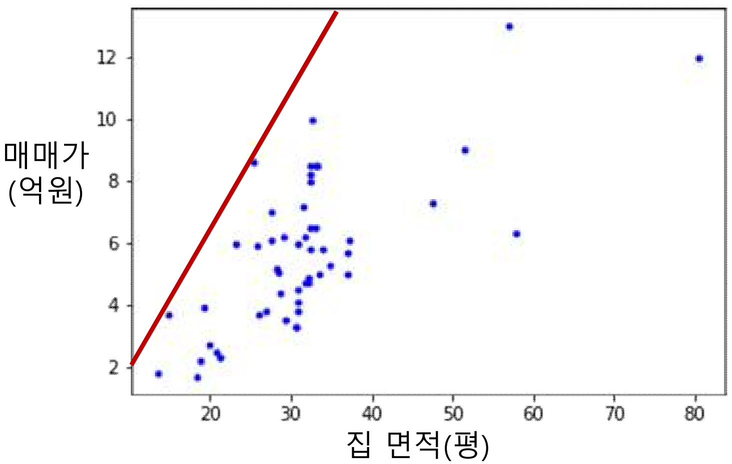
- θ_0, θ_1 을 모두 이용하는 예

2.3 비용 함수

- θ_0, θ_1 을 모두 이용하는 예

$h_\theta(x)$

(θ_0, θ_1고정하면, x에 대한 함수)

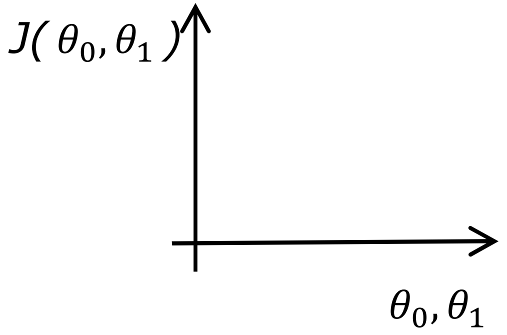


$$h_\theta(x) = 2 + 0.2x$$

$$\theta_0 = 2, \theta_1 = 0.2$$

$J(\theta_0, \theta_1)$

(파라미터 θ_0, θ_1 에 대한 함수)

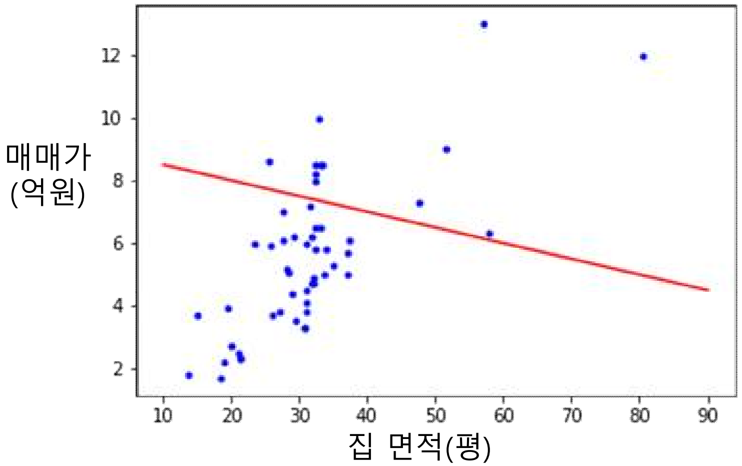


2.3 비용 함수

- θ_0, θ_1 을 모두 이용하는 예

$h_\theta(x)$

(θ_0, θ_1고정하면, x에 대한 함수)

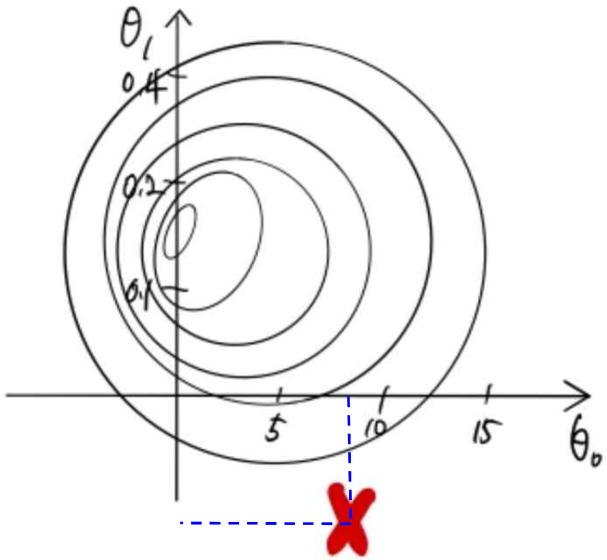


$$h_\theta(x) = 9 - 0.05x$$

$$\theta_0 = 9, \theta_1 = -0.05$$

$J(\theta_0, \theta_1)$

(파라미터 θ_0, θ_1 에 대한 함수)

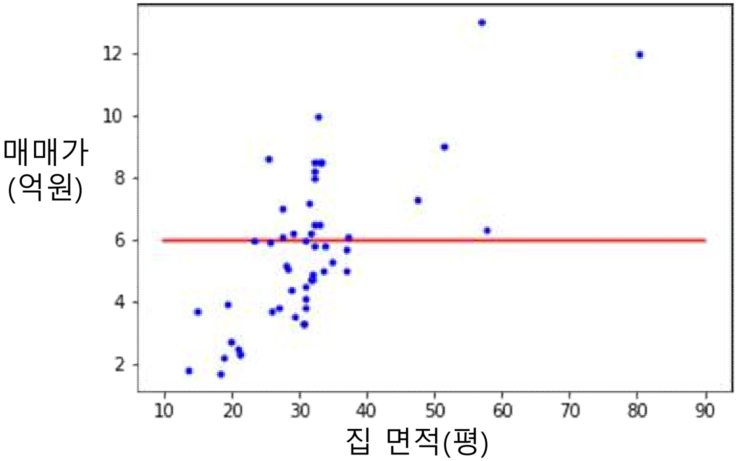


2.3 비용 함수

- θ_0, θ_1 을 모두 이용하는 예

$h_\theta(x)$

(θ_0, θ_1고정하면, x에 대한 함수)

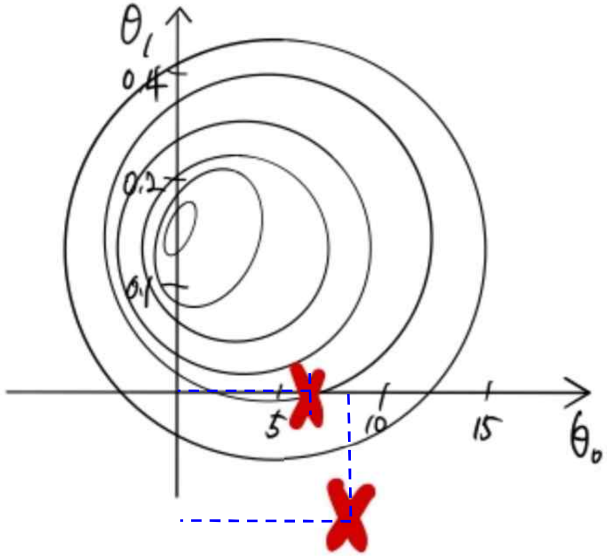


$h_\theta(x) = 6 + 0x$

$\theta_0 = 6, \theta_1 = 0$

$J(\theta_0, \theta_1)$

(파라미터 θ_0, θ_1 에 대한 함수)

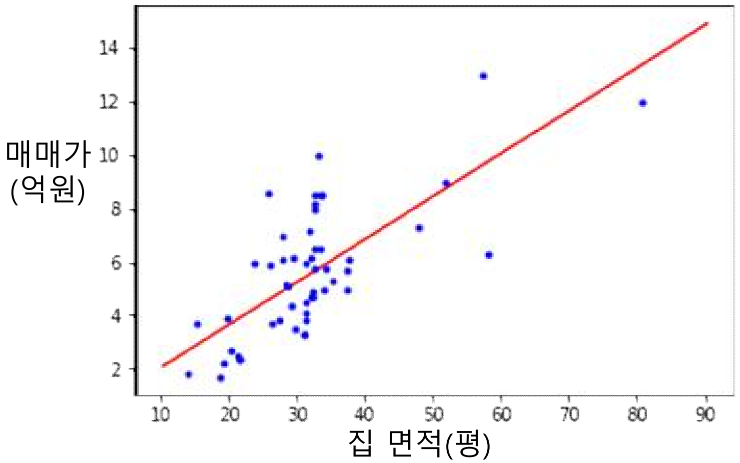


2.3 비용 함수

- θ_0, θ_1 을 모두 이용하는 예

$h_\theta(x)$

(θ_0, θ_1고정하면, x에 대한 함수)

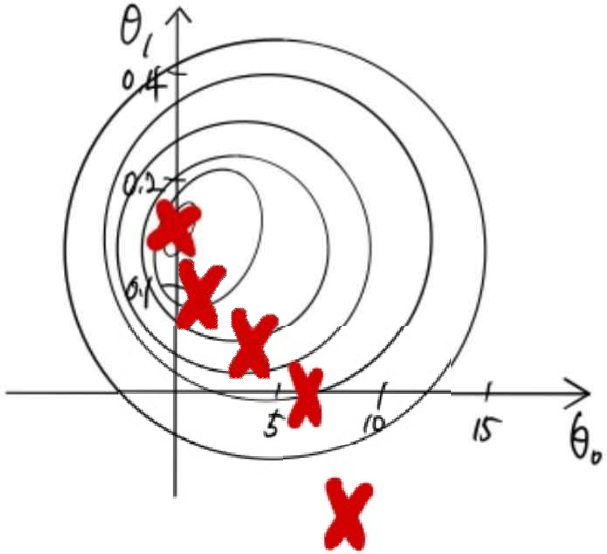


$$h_\theta(x) = 0.51 + 0.16x$$

$$\theta_0 = 0.51, \theta_1 = 0.16$$

$J(\theta_0, \theta_1)$

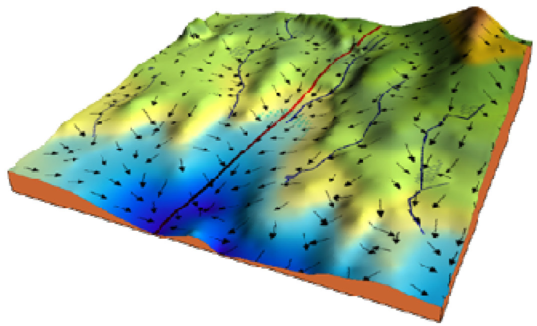
(파라미터 θ_0, θ_1 에 대한 함수)



내용

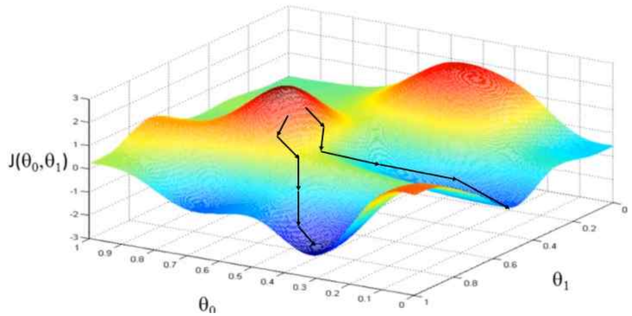
2.4 경사 하강법

- 경사 하강법Gradient descent
  - 미분 가능한 함수의 (지역) 최소값을 찾기 위해 반복적으로 미분을 이용하는 최적화 알고리즘



- 안개 낀 산 속에서 하산하려면?

- 비용 함수와 경사 하강법



©Wikipedia

2.4 경사 하강법

- 함수 $J(\theta_0, \theta_1)$ 가 있을 때,

  $\min_{\theta_0, \theta_1} J(\theta_0, \theta_1)$ 하길 원함

$J(\theta_0, \theta_1, \theta_2, \ldots, \theta_n)$

$\underset{\theta_0, \theta_1, \ldots, \theta_n}{\text{minimize}} J(\theta_0, \theta_1, \ldots, \theta_n)$

- 방법
  - 임의의 θ_0, θ_1 부터 시작
  - θ_0, θ_1 를 바꿔가며 $J(\theta_0, \theta_1)$ 를 줄이자.
    희망하는 최소값에 도달할 때까지

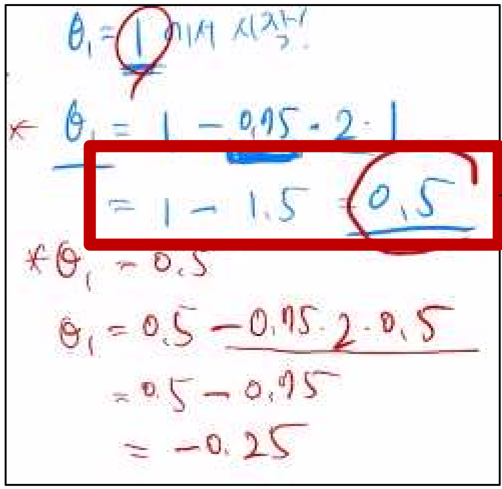
2.4 경사 하강법

- 경사 하강법 알고리즘

```
수렴할 때까지 반복{
```
$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$ (for j=0 and j=1)
```
}
```

학습률 Learning rate   *변화율 → 미분 derivative*

$$\min_{\theta_1} J(\theta_1)$$



※ 계산 상 오타 (1:38 즈음)
'1 -> -0.5 -> 0.25 ...'
이런 식으로 theta 값이 변합니다.

2.4 경사 하강법

- 경사 하강법 알고리즘

수렴할 때까지 반복{

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$$ (for j=0 and j=1)

}

학습률 Learning rate *변화율 gradient* → *미분 derivative*

– 동시에 업데이트 됨

Incorrect:

$$\theta_0 = \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

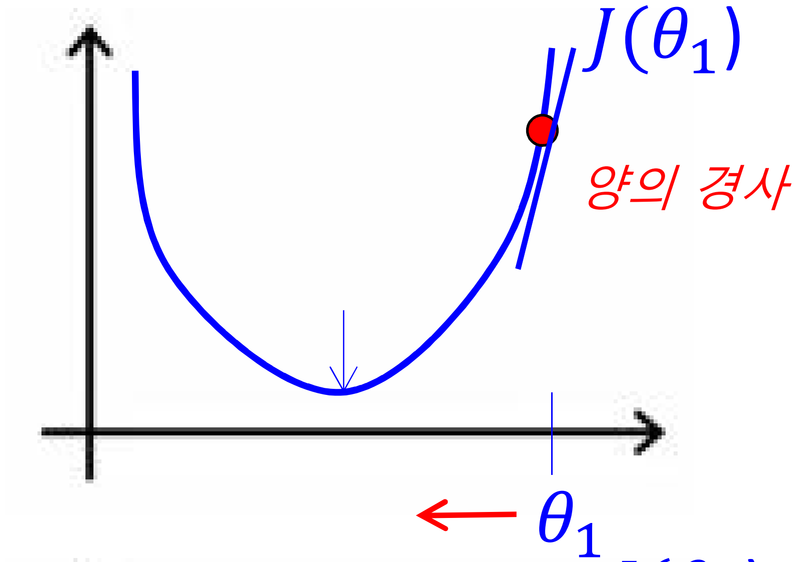
$$\theta_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

Correct: simultaneous update

$$\begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix} = \begin{pmatrix} \theta_0 \\ \theta_1 \end{pmatrix} - \alpha \begin{pmatrix} \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) \\ \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) \end{pmatrix}$$

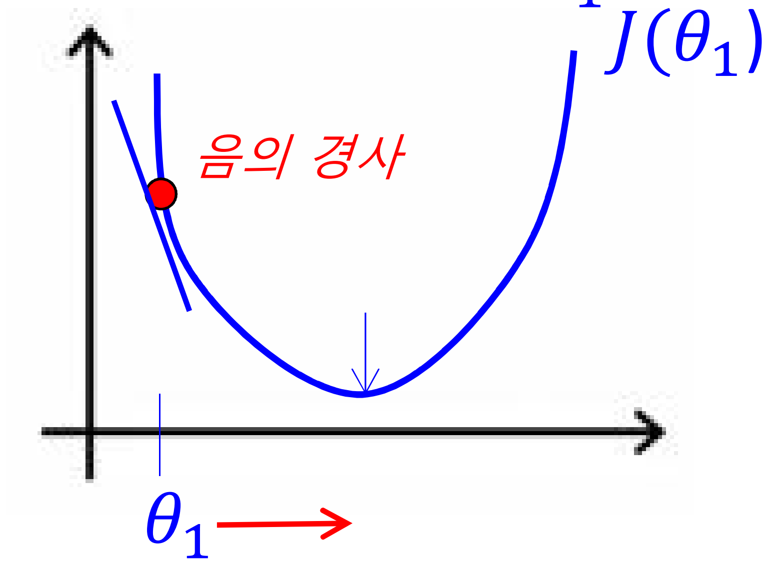
```
theta = theta - learning_rate * gradients
```

2.4 경사 하강법



$$\theta_1 = \theta_1 - \alpha \underbrace{\boxed{\frac{\partial}{\partial \theta_1} J(\theta_1)}}_{\geq 0}$$

$$\theta_1 = \theta_1 - \alpha(\text{양의 값})$$



$$\theta_1 = \theta_1 - \alpha \underbrace{\boxed{\frac{\partial}{\partial \theta_1} J(\theta_1)}}_{\leq 0}$$

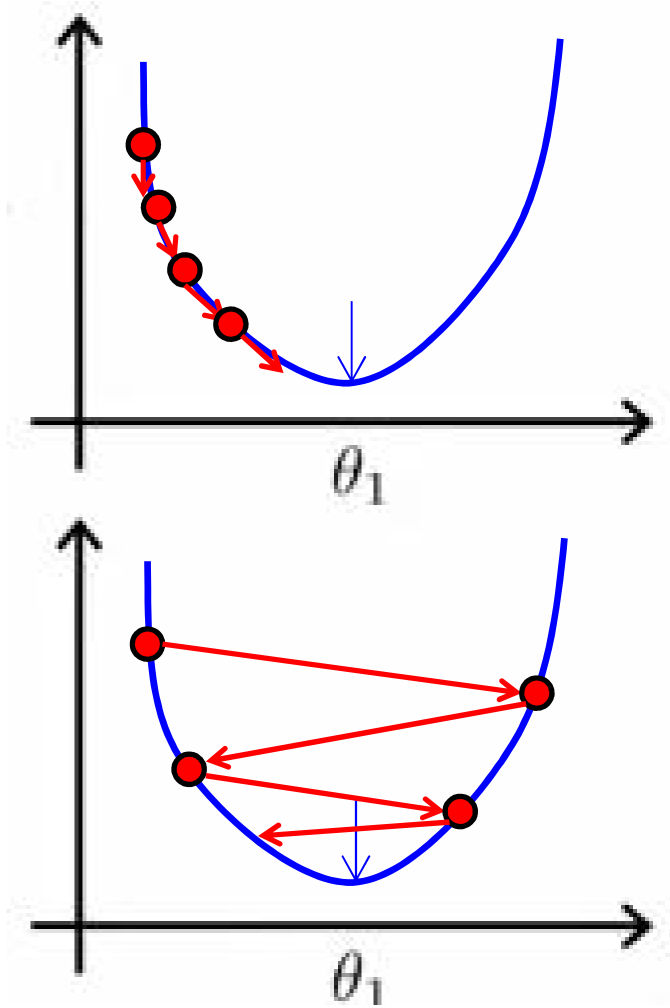
$$\theta_1 = \theta_1 - \alpha(\text{음의 값})$$

2.4 경사 하강법

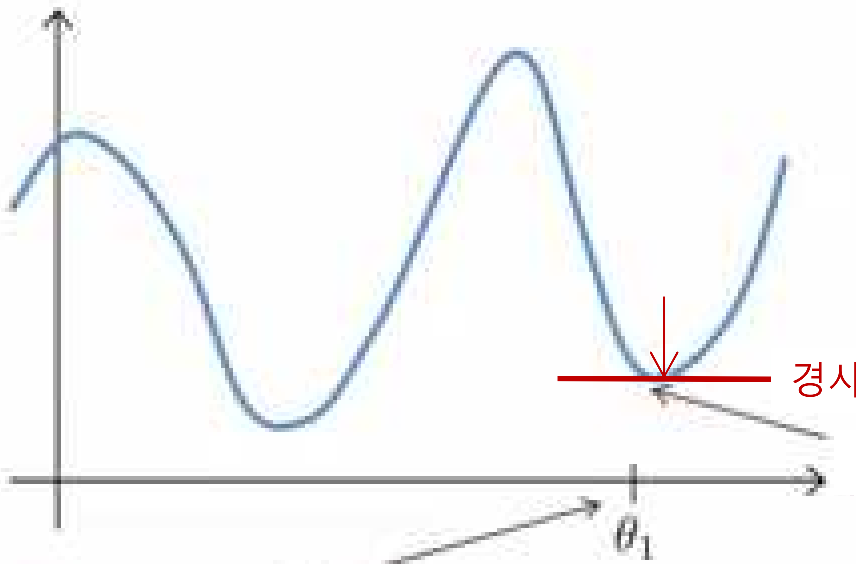
$$\theta_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

α 가 너무 작으면, 경사 하강법이 느려질 수 있음

α 가 너무 크면, 최소값을 지나칠 수 있음. 수렴에 실패하게 되고 발산되기도 함



2.4 경사 하강법



지역적 최소값local minimum (극소) 에 위치한 θ_1

θ_1 의 현재 값

$$\theta_1 := \theta_1 - \alpha \boxed{\frac{d}{d\theta_1} J(\theta_1)} = 0$$

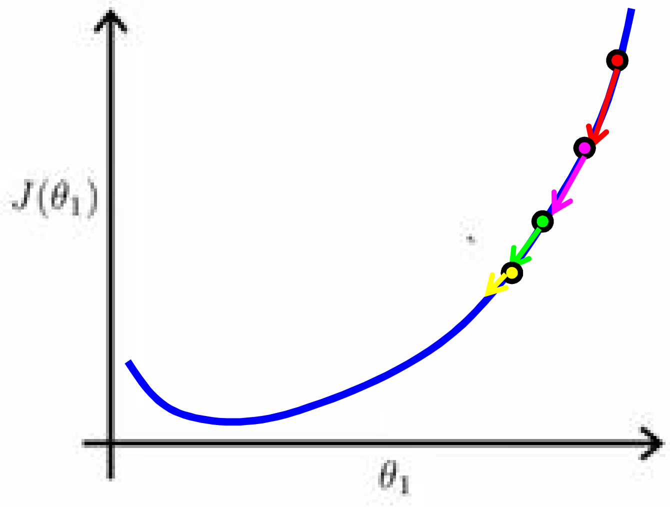
$$\theta_1 \coloneqq \theta_1 - \alpha \cdot 0$$

$$\theta_1 \coloneqq \theta_1$$

2.4 경사 하강법

- α 를 고정하더라도, 경사 하강법은 극소에 수렴할 수 있음

$$\theta_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$



  - 극소에 수렴할 때, 경사 하강법은 자동으로 조금씩 이동하게됨. 따라서 반복할 때, 후반부에 따로 α 를 줄일 필요 없음.

2.4 경사 하강법

경사 하강법 알고리즘

```
수렴할 때까지 반복{
```

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} \boxed{J(\theta_0, \theta_1)}$$

```
    (for j=1 and j=0)
}
```

선형 회귀 모델

$$h_\theta(x) = \theta_0 + \theta_1 x$$

$$\boxed{J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2}$$

2.4 경사 하강법

$$\frac{\partial}{\partial\theta_j}J(\theta_0,\theta_1) = \frac{\partial}{\partial\theta_j}\left[\frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2\right]$$
$$= \frac{\partial}{\partial\theta_j}\left[\frac{1}{2m}\sum_{i=1}^{m}\left(\theta_0+\theta_1 x^{(i)}-y^{(i)}\right)^2\right]$$

$$\theta_0(j=0): \frac{\partial}{\partial\theta_0}J(\theta_0,\theta_1) = \frac{1}{m}\sum_{i=1}^{m}\left(\theta_0+\theta_1 x^{(i)}-y^{(i)}\right)$$

$$\theta_1(j=1): \frac{\partial}{\partial\theta_1}J(\theta_0,\theta_1) = \boxed{}$$

2.4 경사 하강법

- 경사 하강법 알고리즘

수렴할 때까지 반복{

$$\theta_0 = \theta_0 - \alpha \boxed{\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)}$$

$$\theta_1 = \theta_1 - \alpha \boxed{\frac{1}{m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) \cdot x^{(i)}}$$

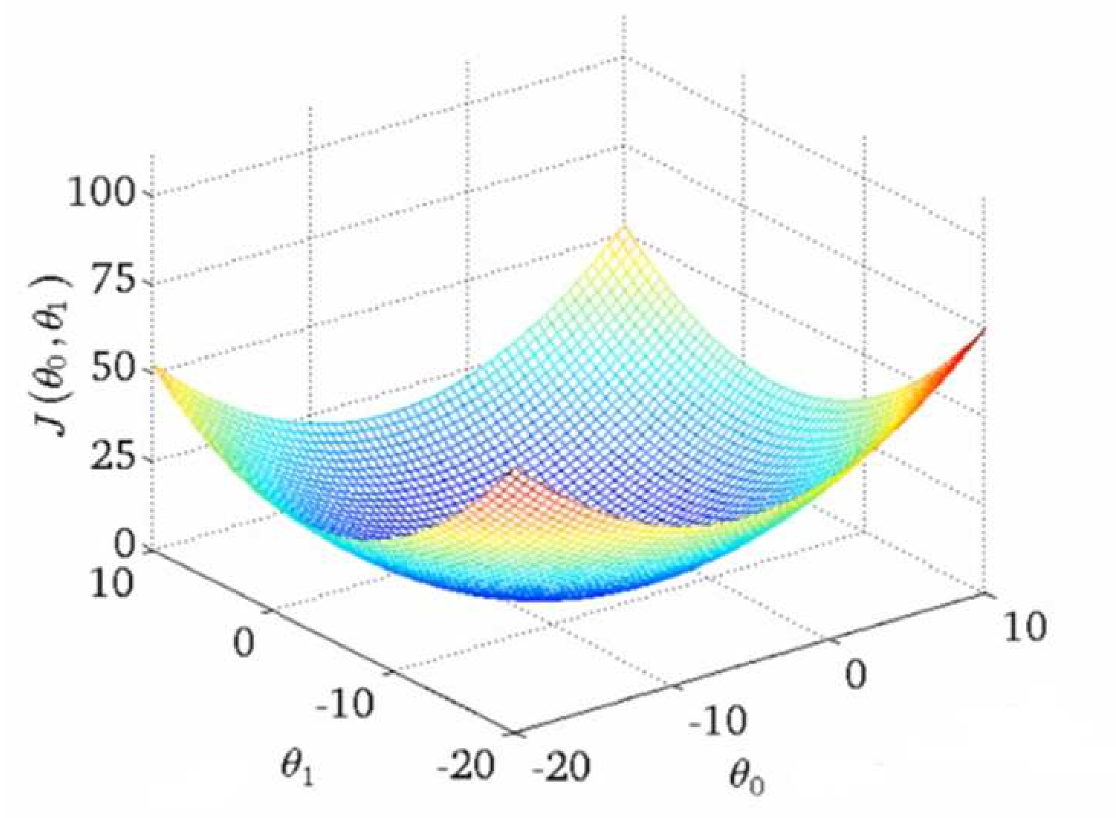
}

$$\frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$$

$$\frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$

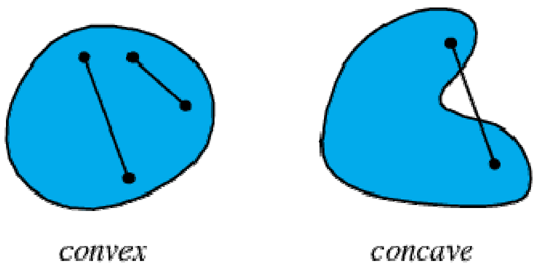
2.4 경사 하강법

- 볼록 함수Convex function
  - 그릇 모양



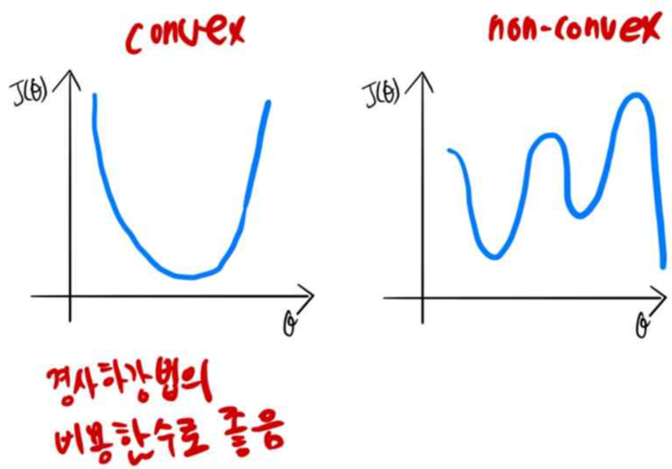
2.4 경사 하강법

- Convex 란?
  - Convex 공간은?
    - 어떤 공간의 아무 두 점을 이은 선분이 모두 그 공간에 포함되어 있어야 함



  - Convex function 의 성질
    - Initial point 를 아무 점으로 설정해도, local minimum 이 only 1

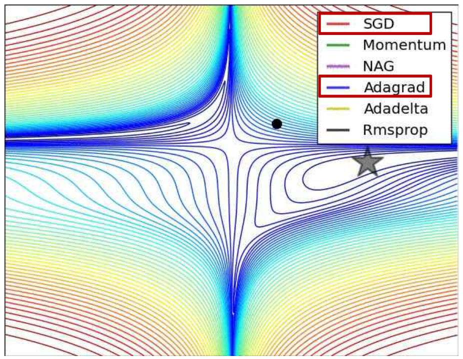
  - Convex function 인지 여부
    - 이차 미분 이용

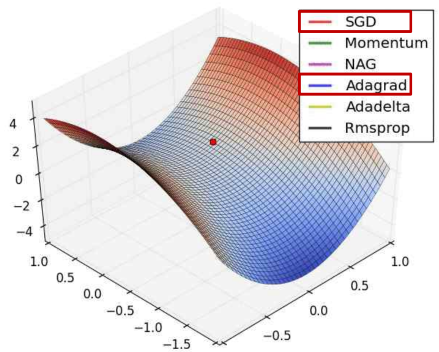


2.4 경사 하강법

$$\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)$$

- 배치 경사 하강법Batch gradient descent
  - 파라미터를 1회 업데이트 시킬 때, 전체 학습 예제들을 사용
- 확률적 경사 하강법Stochastic gradient descent(SGD)
  - 1회 업데이트 시킬 때, 무작위로 딱 한 개의 학습 예제를 사용
- 미니배치 경사 하강법Mini-batch gradient descent
  - 1회 업데이트 시킬 때, 미니배치라 부르는 임의의 작은 학습 예제 샘플을 사용





(심층학습 6~7주차 내용)

경사 하강법 알고리즘 등 비교 ©Wikipedia

내용

2.1 선형 회귀란?Linear regression

2.2 모델 설계Model representation

2.3 비용 함수Cost function

2.4 경사 하강법Gradient descent

학기 내용

1. 기계 학습 소개
2. 단항 선형 회귀 Linear regression with one variable
3. **다항 선형 회귀 Linear regression with multiple variables**
4. 로지스틱 회귀(분류) Logistic Regression(Classification)
5. 정규화 Regularization
6. Support Vector Machine (SVM)
7. 군집 Clustering
8. 차원 축소 Dimensionality Reduction