경사하강법

- 권수태 교수

JEONJU UNIVERSITY
The Place for Superstars

1. 경사하강법

❖ 개요

- 머신러닝 및 딥러닝 알고리즘을 학습시킬 때 사용하는 방법 중 하나
- 1차 근삿값 발견용 최적화 알고리즘
- 함수 값이 낮아지는 방향으로 독립 변수 값을 변형시켜가면서 최종적으로는 최소 함수 값을 갖도록 하는 독립 변수 값을 찾는 방법
- Steepest descent method 라고도 함
- 함수의 기울기를 구하여 기울기가 낮은 쪽으로 계속 이동시켜 극값(최적값)에 이를 때까지 반복하는 것

1. 경사하강법

❖ 경사하강법 사용 이유

➢ 왜 미분 계수가 0 인 지점을 찾지 않고 경사하강법을 쓰는지?

- ✓ 함수가 닫힌 형태가 아닌 경우
- ✓ 함수가 너무 복잡해 미분 계수를 구하기 어려운 경우
- ✓ Gradient Descent Method를 사용하는 것이 미분 계수를 구하는 것보다 더 쉬운 경우
- ✓ 데이터 양이 너무 많은 경우 (효율적으로 계산하기 위해)

1. 경사하강법

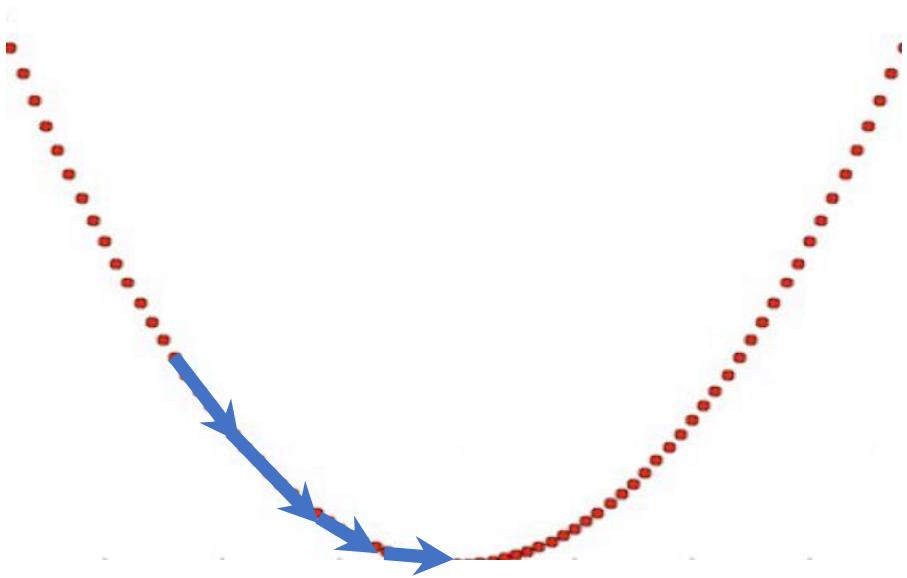
❖ 필요성

- ➢ 기계학습의 최적화
  - ✓ 수학적 함수의 최소값을 찾는 것과 달리
  - ✓ 학습자료가 주어지고 학습자료의 집합을 이용하여 얻어지는 목적함수의 최저점을 찾는 과정
- ➢ 학습 모델의 매개변수 공간
  - ✓ 학습자료의 집합이 한정되어 정답을 찾는 것이 어려움
  - ✓ 다중회귀분석의 경우 입력 특징변수가 많아지면 한정된 학습자료로부터 정답을 알아내는데 문제가 있음
  - ✓ 적절한 모델을 선택하고 목적함수를 정의하여, 모델의 매개변수 공간을 탐색하면서 목적함수가 최저가 되는 값을 찾는 과정이 필요

1. 경사하강법

❖ 원리

- ➢ 임의의 비용함수가 주어질 때 가장 낮은 곳을 찾는 방법
  - ✓ 지도없이 산위에서 내려오는 과정
  - ✓ 경사도가 없는 곳, 즉 기울기가 0인 곳까지 내려오는 원리



1. 경사하강법

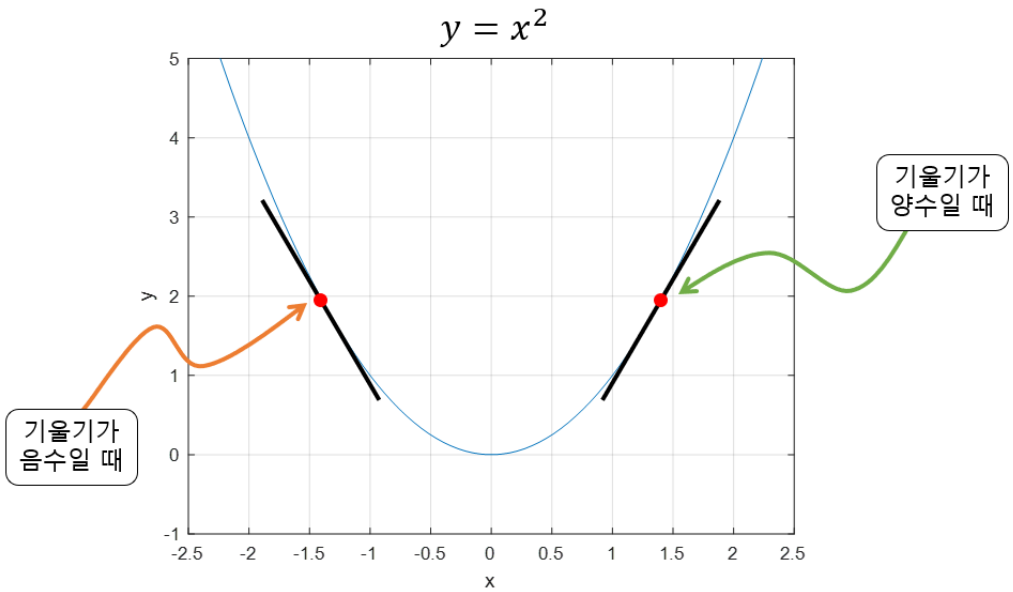
❖ 수식

➢ 함수의 기울기(gradient)를 이용

✓ 기울기가 양수 : x 의 값이 커질수록 함수 값이 커짐

✓ 기울기가 음수 : x 의 값이 커질수록 함수 값이 작아 짐

✓ $x_{i+1} = x_i$ – 이동거리 × 기울기의 부호



1. 경사하강법

❖ 수식

- ➢ 이동거리 : 최솟값에 가까워질수록 기울기의 크기는 작아짐
  - ✓ 극소값에 가까울 수록 조심히 움직여야 발산하지 않음
  - ✓ 이동거리에 사용할 값을 gradient의 크기와 비례하는 factor를 이용하면 현재 x 의 값이 극소값에서 멀 때는 많이 이동하고, 극소값에 가까워졌을 때는 조금씩 이동
  - ✓ $x_{i+1} = x_i$ – (이동거리 × 기울기의 부호)
  - ✓ $x_{i+1} = x_i$ – (기울기의 크기 × 기울기의 부호)
  - ✓ $x_{i+1} = x_i$ – 기울기

1. 경사하강법

❖수식

- Step Size ($=\eta$)
  - ✓사용자의 필요에 맞게 이동 거리를 조절할 수 있도록 해주기 위하여 조절 인자를 넣기도 함

$$x_{i+1} = x_i - \text{기울기}$$

$$\downarrow$$

$$x_{i+1} = x_i - \eta * \nabla f(x_i)$$

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \cdots, \frac{\partial f}{\partial x_n}\right)$$

1. 경사하강법

❖수식

➢ Step Size 주의사항

$$x_{i+1} = x_i - \eta * \nabla f(x_i)$$

✓ 인자 η 값이 너무 작을 경우

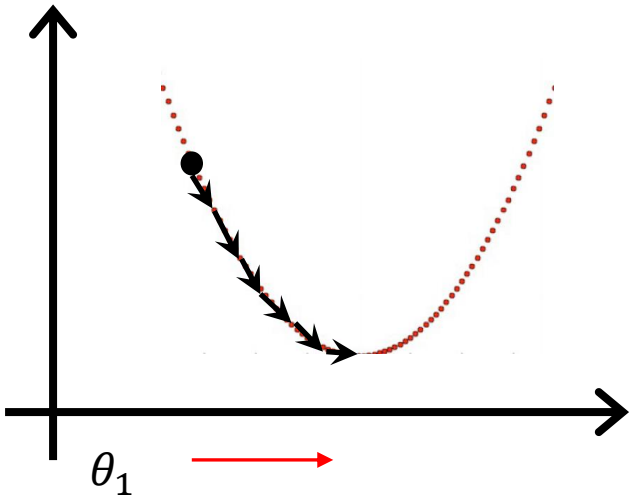
- 반복해야 하는 값이 많으므로 학습 시간이 오래 걸림
- 지역 최소값(local minimum)에 수렴할 수 있음

✓ 인자 η 값이 너무 클 경우

- 빠르게 수렴할 수 있지만, 최소값으로 수렴되지 못하고 발산할 여지 있음
- 스텝이 너무 커서 전역 최소값(global minimum)을 가로질러 반대편으로 건너 뛸 수 있음

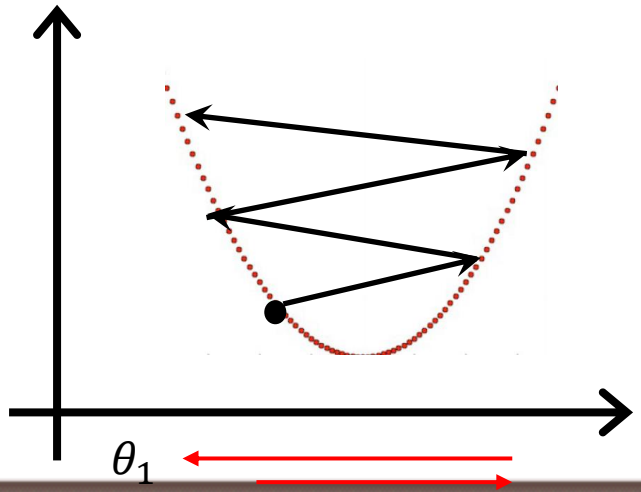
1. 경사하강법

❖학습률 η 의 영향



$$x_i = x_i - \eta * \frac{\partial}{\partial x_i} J(x_0, x_1)$$

학습률이 너무 적으면 최저점을 찾아 수렴하는 시간이 매우 느림



학습률이 너무 크면 최저점을 벗어나게 되고 수렴하지 않을 수 있음

1. 경사하강법

❖미분과 기울기, 그리고 경사 하강법의 개념

➢ 머신러닝에 사용되는 여러 기법들을 이해하기 위해 필요한 수학적인 개념 중에서 가장 중요한 개념은 바로 미분(derivative)

✓ 미분이란 순간 변화량을 구하는 것

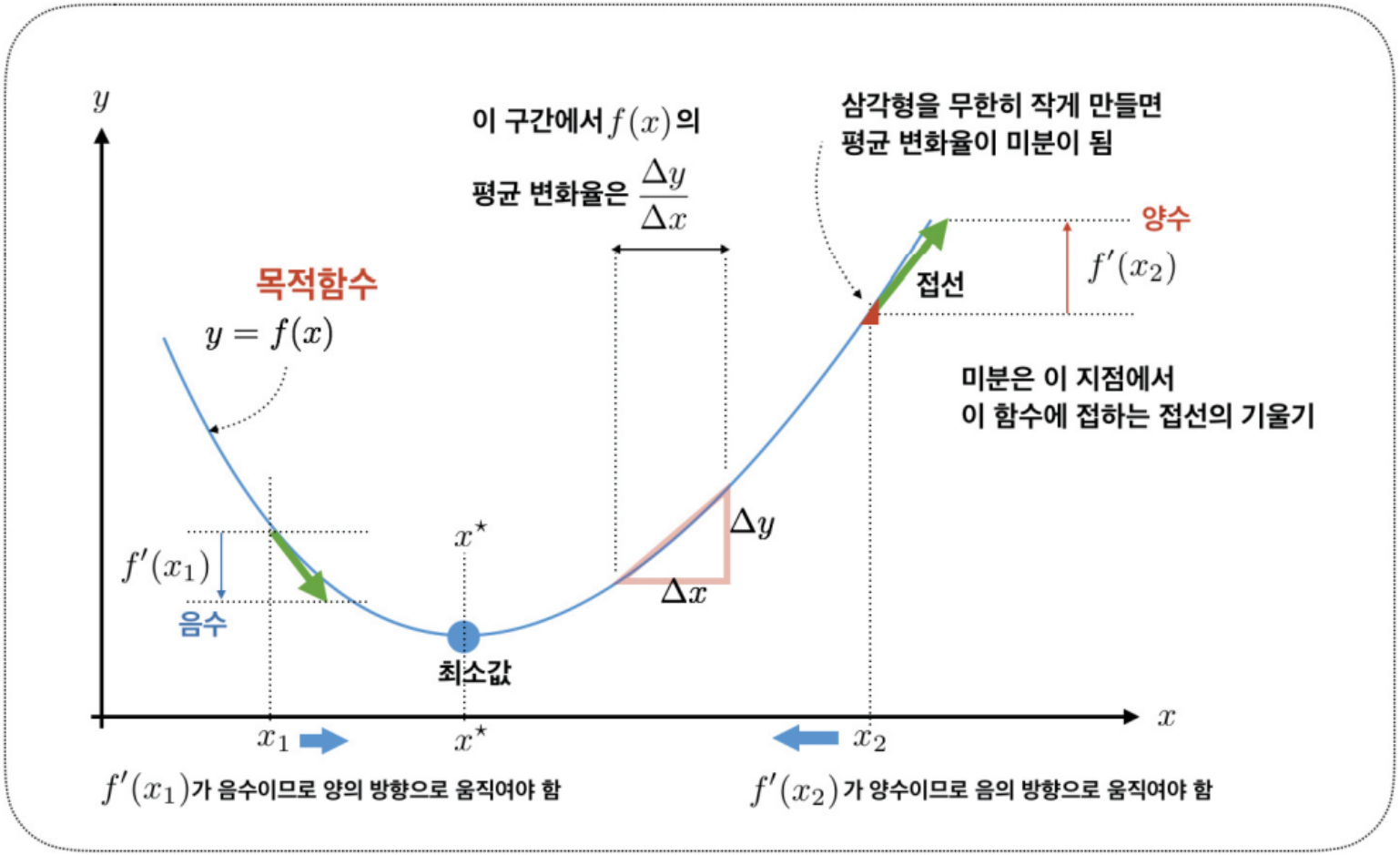
✓ 독립 변수값의 변화량에 대한 함수값 변화량 비의 극한

$$f'(a) = \lim_{\Delta x \to 0} \frac{f(a+\Delta x) - f(a)}{\Delta x}$$

1. 경사하강법

❖미분과 기울기, 그리고 경사 하강법의 개념

➢미분

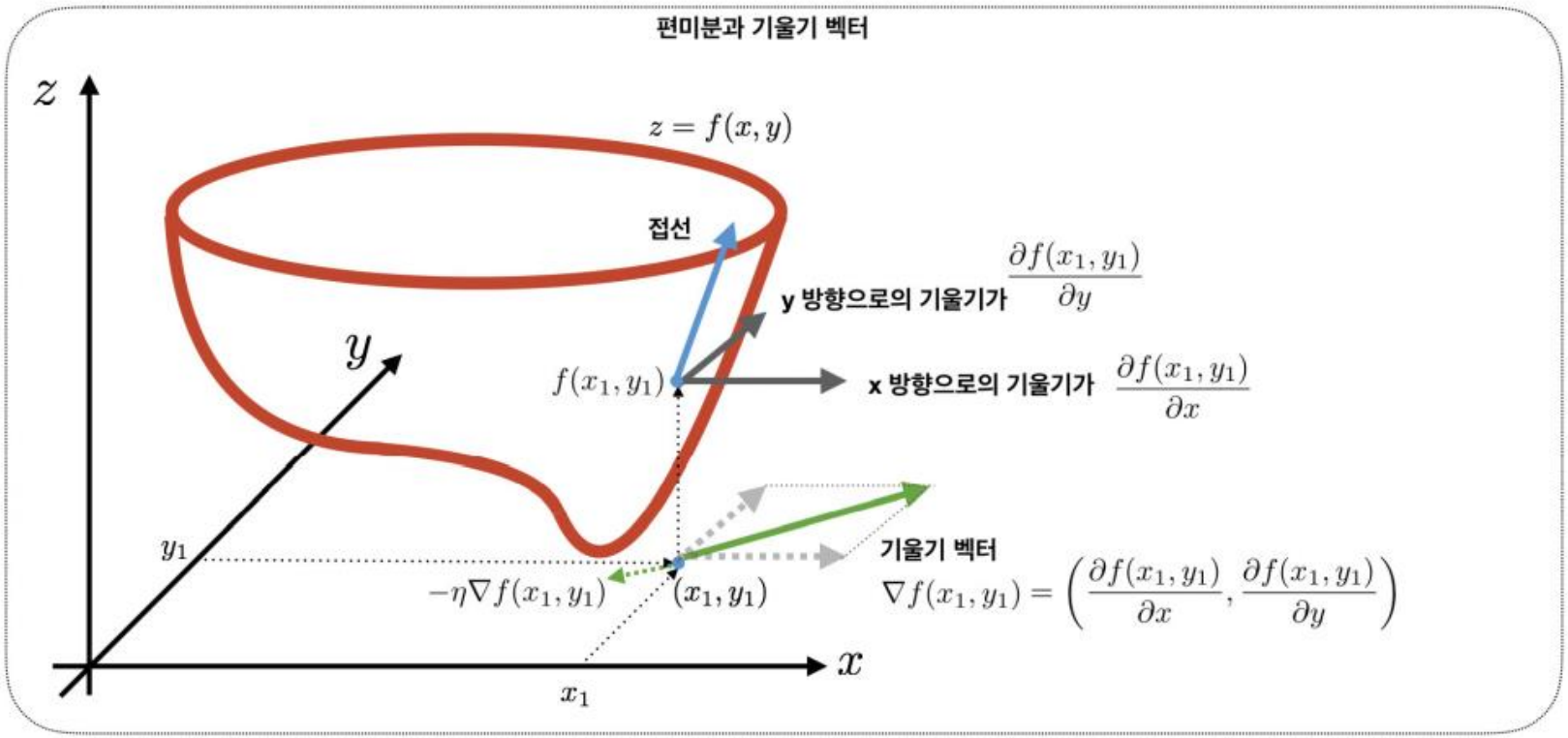


JEONJU UNIVERSITY

1. 경사하강법

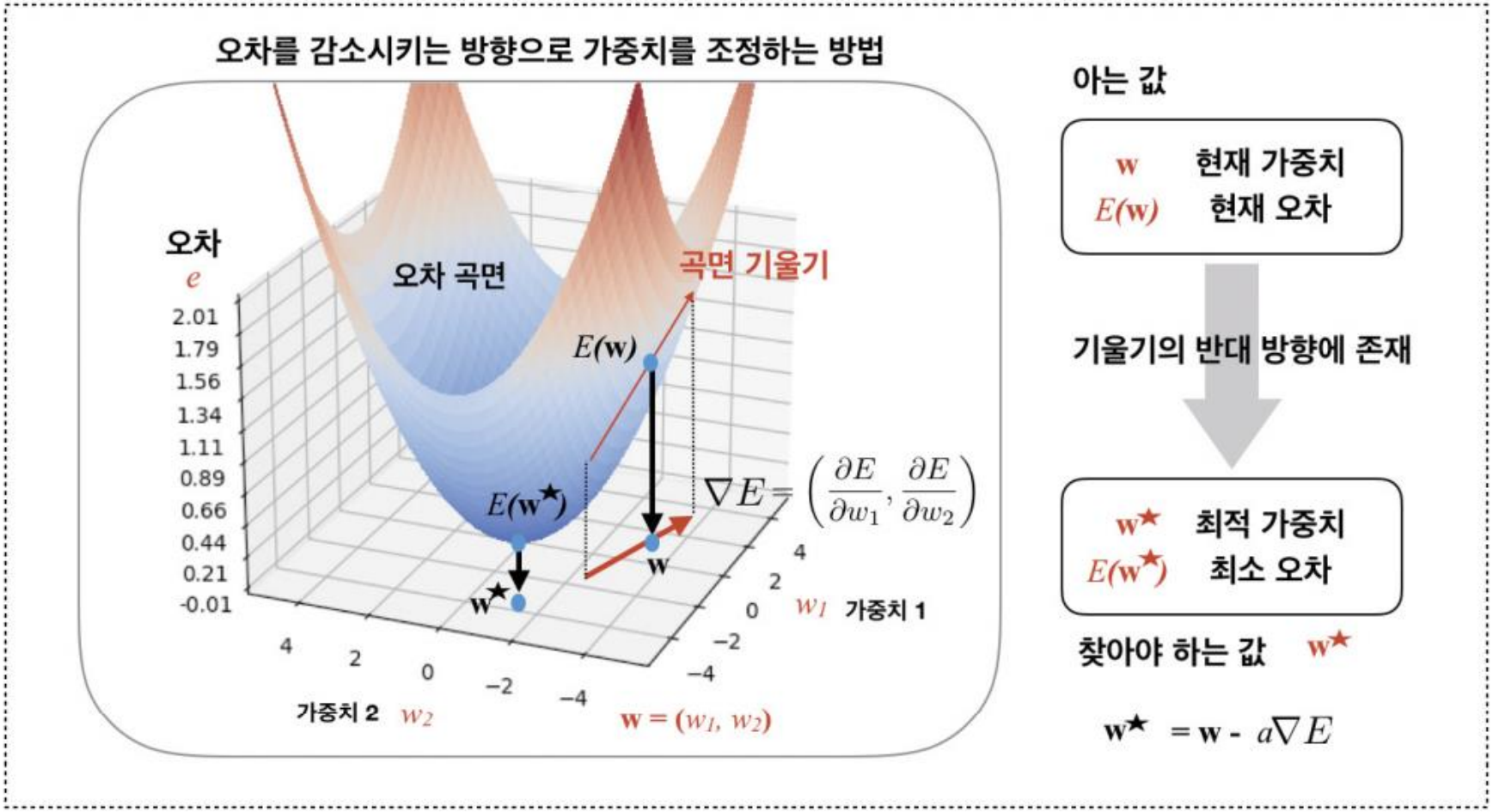
❖미분과 기울기, 그리고 경사 하강법의 개념

➢ 편미분(partial derivative)이란 둘 이상의 변수들을 가지는 함수 f가 있을 경우, 이 함수를 각각의 변수에 대해서 독립적으로 미분을 하는 방식



1. 경사하강법

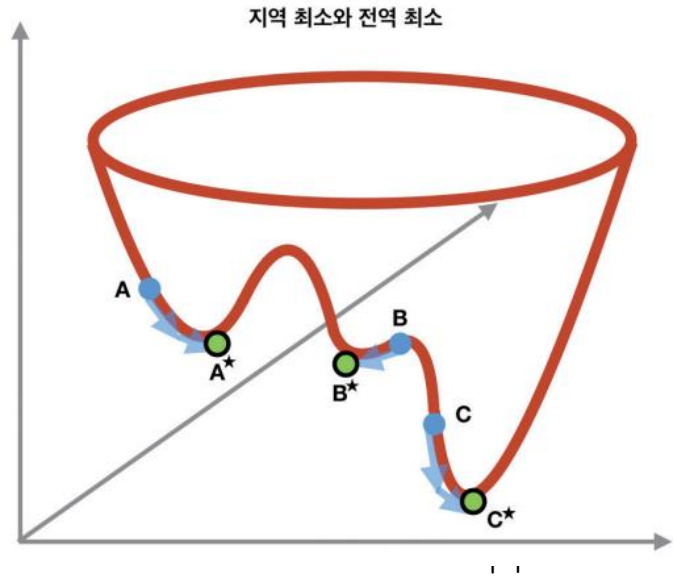
❖미분과 기울기, 그리고 경사 하강법의 개념



1. 경사하강법

❖경사하강법

- ➢ 현재 가중치 w 위치에서 오차 곡면의 기울기를 안다면 기울기를 따라 내려가면 곡면을 낮은 곳으로 향할 수 있음
- ➢ 경사 하강법을 통해 얻은 답은 일정한 영역내에서 가장 좋은 답
- ➢ 전체 공간에서 가장 좋은 해는 전역 최소값global minimum



1. 경사하강법

❖경사하강법

- 경사 하강법을 통한 학습이라는 것은 현재의 가중치 w 를 더 좋은 가중치 w'로 바꾸는 일

$$\mathbf{w}' = \mathbf{w} - \eta \nabla E$$

- 지역 최적값에 붙잡혀 전역 최적값을 찾지 못할 위험이 존재
- η(eta)는 기울기의 반대 방향으로 얼마나 이동할 것인지를 결정하는 학습률
- 학습 과정에 영향을 미치는 값들을 하이퍼파라미터(hyperparameter)라고 함

1. 경사하강법

❖경사하강법

➢ 알고리즘

✓수렴할 때까지 아래의 과정을 반복 수행

- $w_j = w_j - \eta \frac{\partial}{\partial w_j} J(w_0, w_1)$

✓단순회귀분석에서의 회귀변수 학습 과정

$w_0 = init0, \quad w_1 = init1$

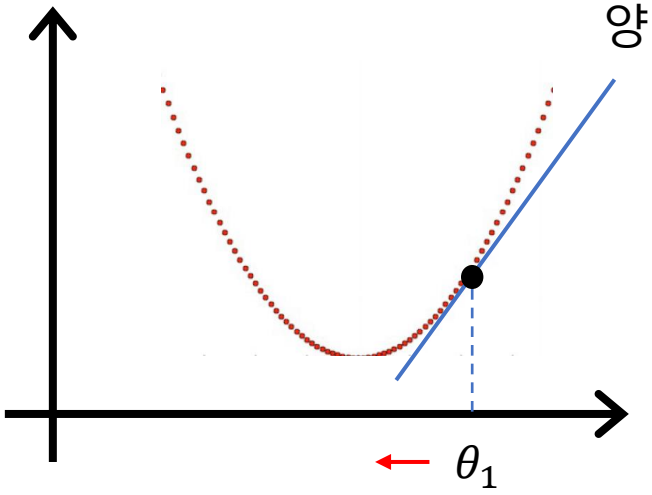
Temp0 = $w_0 - \eta \frac{\partial}{\partial w_0} J(w_0, w_1)$, Temp1 = $w_1 - \eta \frac{\partial}{\partial w_1} J(w_0, w_1)$

$w_0 = temp0, \quad w_1 = temp1$

Temp0 = $w_0 - \eta \frac{\partial}{\partial w_0} J(w_0, w_1)$, Temp1 = $w_1 - \eta \frac{\partial}{\partial w_1} J(w_0, w_1)$

1. 경사하강법

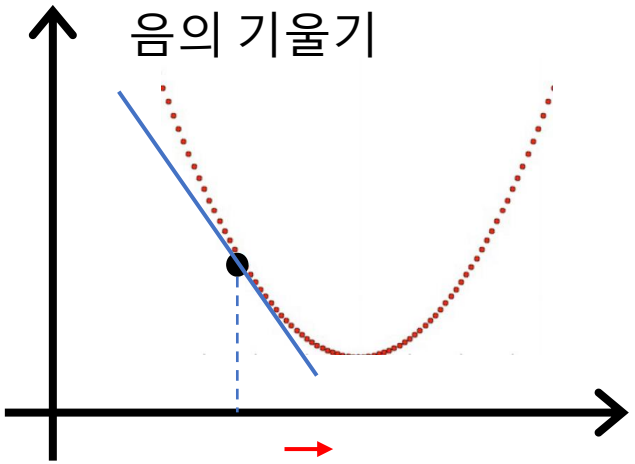
❖ 경사하강법



$$w_1 = w_1 - \eta \frac{\partial}{\partial w_1} J(w_0, w_1)$$

$$w_1 = w_1 - \eta \nabla$$

w_1이 증가하여 최저점으로 이동



$$w_1 = w_1 - \eta \frac{\partial}{\partial w_1} J(w_0, w_1)$$

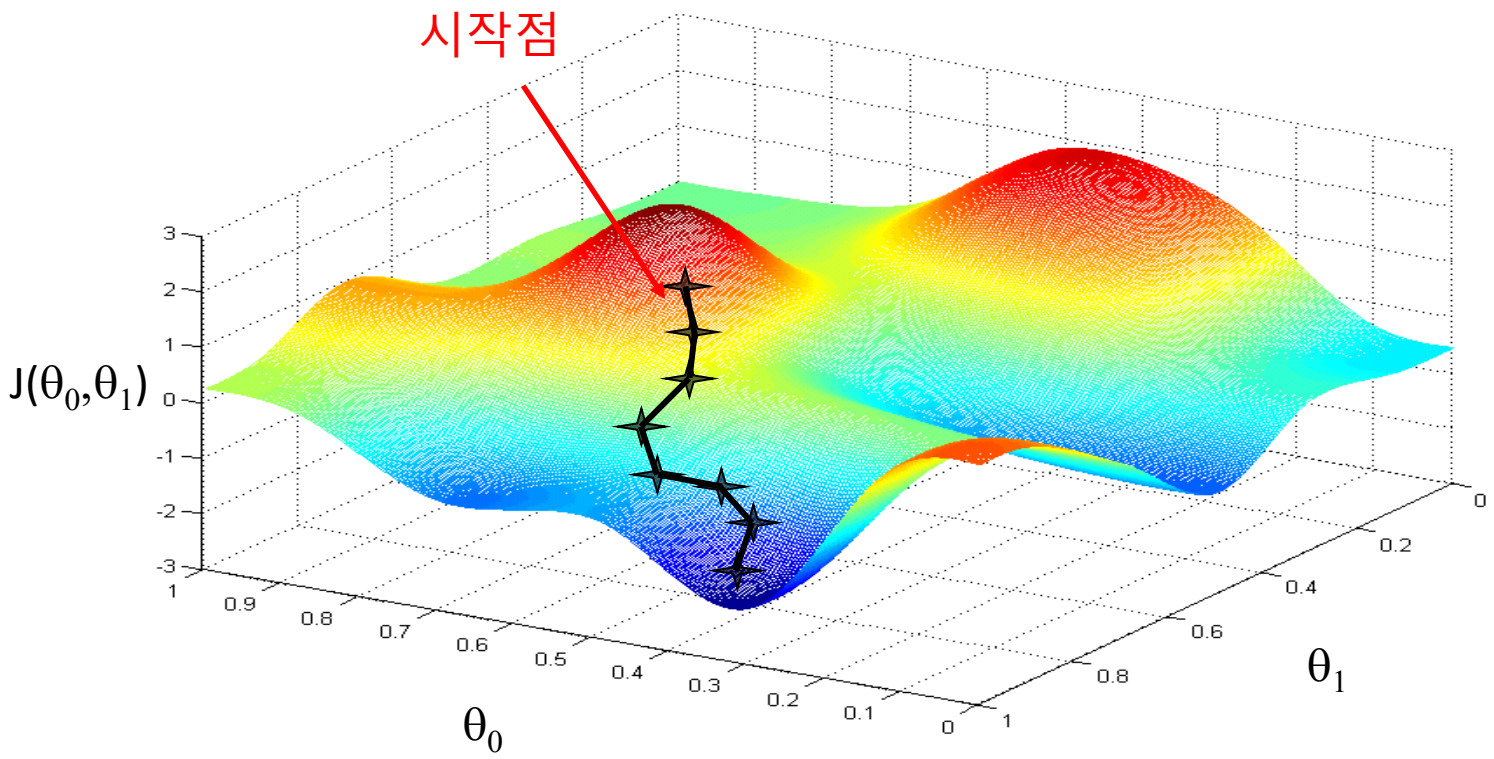
$$w_1 = w_1 + \eta \nabla$$

w_1이 감소하여 최저점으로 이동

1. 경사하강법

❖ 경사하강법의 약점

➢ 시작점의 차이에 따라 다른 결과를 가져올 수 있음

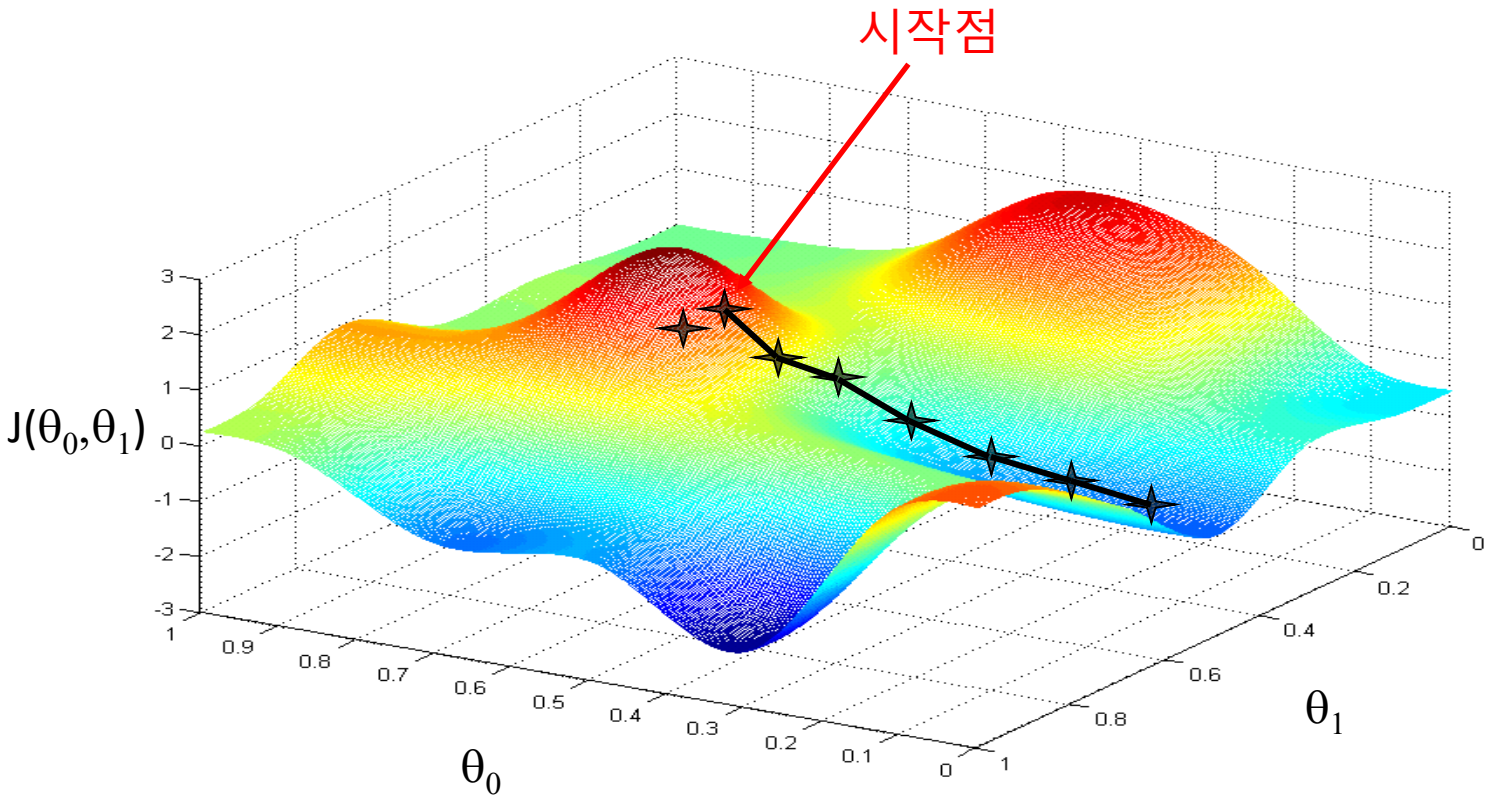


JEONJU UNIVERSITY

1. 경사하강법

❖ 경사하강법의 약점

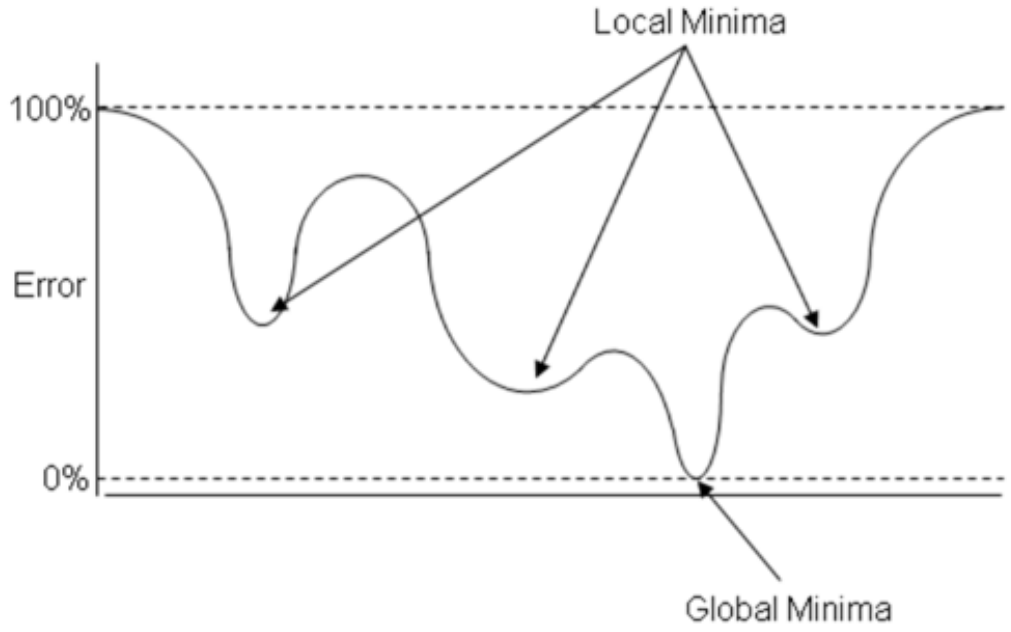
➢ 시작점의 차이에 따라 다른 결과를 가져올 수 있음



1. 경사하강법

❖ 경사하강법의 약점

- ➢ 경사 하강법은 현재 위치에서의 기울기를 사용하기 때문에 지역 최소값에 빠질 수 있음
- ➢ 평탄한 지역을 지날 때는 시간이 오래 걸리고 일찍 멈춰서 전역 최소값에 도달하지 못 할 수도 있음



JEONJU UNIVERSITY

1. 경사하강법

❖ 경사하강법의 약점 보완방법

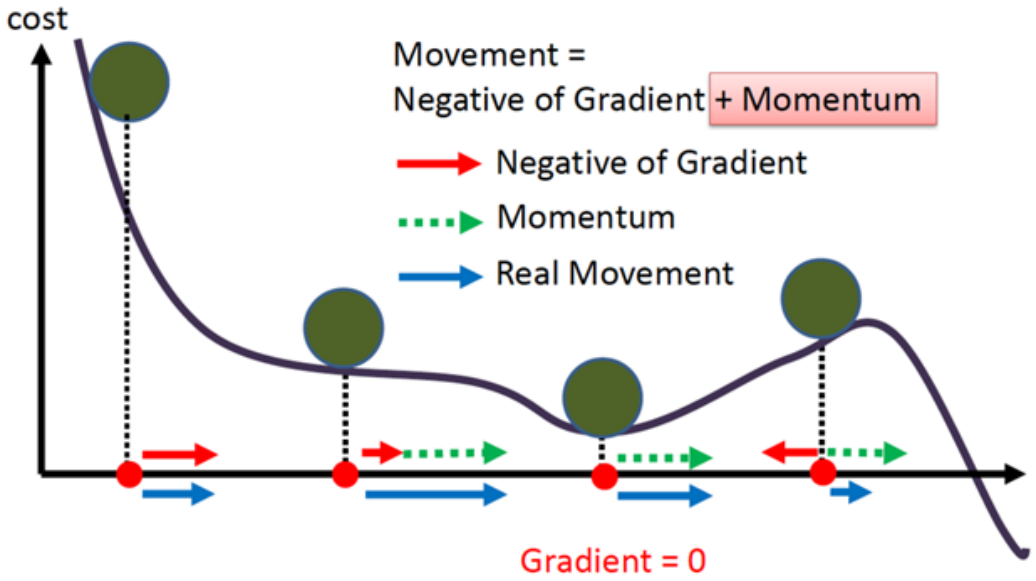
- 모멘텀 (Momentum)
- Batch Gradient Descent
- 확률적 경사 하강법 (Stochastic Gradient Descent)
- Mini-batch Gradient Descent

1. 경사하강법

❖ 경사하강법의 약점 보완방법

➢ 모멘텀 (Momentum)

- ✓ 기울기에 관성을 부과하여 작은 기울기는 쉽게 넘어가도록
- ✓ 예) 언덕에서 공을 굴렸을 때, 낮은 언덕은 공의 관성을 이용하여 쉽게 넘어갈 수 있게 하여 지역 최소값을 탈출 할 수 있게 함



JEONJU UNIVERSITY

1. 경사하강법

❖ 경사하강법의 약점 보완방법

- 경사하강법 약점(2)
  - ✓ 전체 데이터를 모두 사용해서 기울기를 계산
  - ✓ 데이터가 큰 경우 학습하는데 많은 시간 필요
- 배치 사이즈(Batch size)
  - ✓ 경사하강법에서 배치는 단일 반복에서 기울기를 계산하는 데 사용하는 data의 총 개수
  - ✓ 경사하강법에서의 배치는 전체 데이터 셋이라고 가정함
  - ✓ 연산 한번에 들어가는 데이터의 크기
  - ✓ mini Batch
    - 1 Batch size에 해당하는 데이터 셋

1. 경사하강법

❖ 경사하강법의 약점 보완방법

- ➢ 에포크(Epoch)
  - ✓ 훈련 데이터셋에 포함된 모든 데이터들이 한 번씩 모델을 통과한 횟수로, 모든 학습 데이터셋을 학습하는 횟수
  - ✓ 1 epoch: 전체 학습 데이터셋이 한 신경망에 적용되어 순전파와 역전파를 통해 신경망을 한번 통과했다는 의미
  - ✓ epoch를 높일수록, 다양한 무작위 가중치로 학습을 해보는 것이므로 적합한 파라미터를 찾을 확률이 올라감. 즉, 손실값이 내려감
  - ✓ 지나치게 epoch를 높이게 되면, 그 학습 데이터셋에 과적합(overfitting)되어 다른 데이터에 대해선 제대로 된 예측을 하지 못할 수 있음

1. 경사하강법

❖ 경사하강법의 약점 보완방법

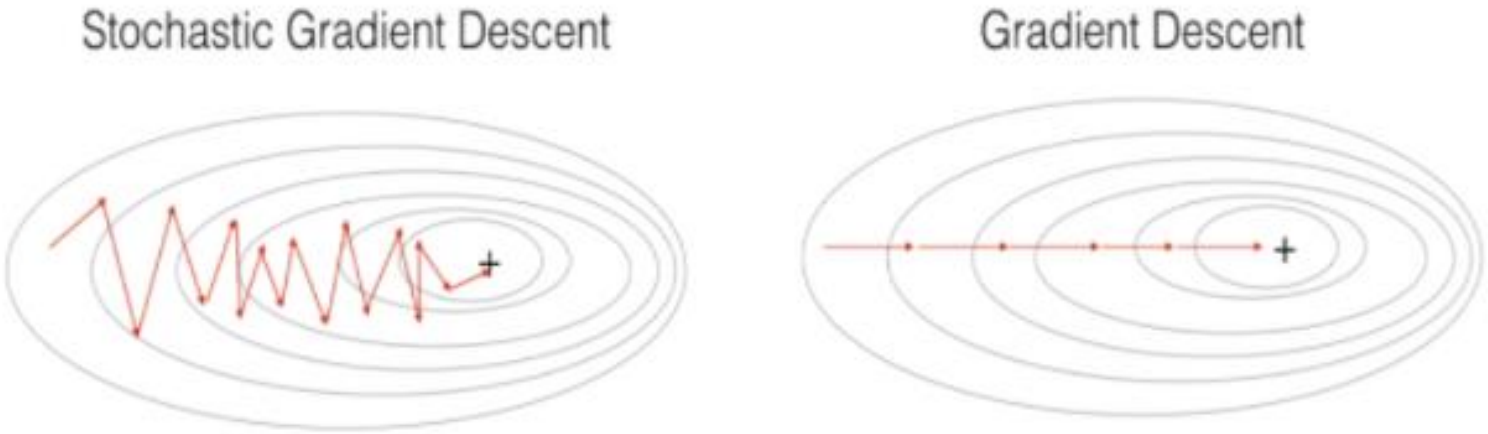
- ➢ 확률적 경사 하강법 (Stochastic Gradient Descent)
  - ✓ 매 step에서 딱 한개의 샘플을 무작위로 선택하고 그 하나의 샘플에 대한 기울기 계산
  - ✓ 배치 크기가 1인 경사하강법 알고리즘
  - ✓ 특징
    - 하나의 샘플만 메모리에 있으면 되기 때문에 큰 데이터셋도 학습 가능하며 학습 속도가 빠름
    - Cost Function이 매우 불규칙할 경우, 알고리즘이 local minimum을 건너뛰도록 도와주어 global minimum을 찾을 가능성이 높음
    - 샘플의 선택이 확률적이기 때문에 배치 경사 하강법에 비해 불안정
    - 반복이 충분하면 SGD가 효과 있지만 노이즈가 매우 심함

1. 경사하강법

❖ 경사하강법의 약점 보완방법

- ➢ 확률적 경사 하강법 (Stochastic Gradient Descent)
  - 부드럽게 감소하지 않고 위아래로 요동치며 평균적으로 감소

< 최소값을 찾는 과정 >



1. 경사하강법

❖ 경사하강법의 약점 보완방법

➢ 확률적 경사 하강법 (Stochastic Gradient Descent)

- 학습자료 마다 미분값을 계산하여 업데이트

Repeat {

x^i 순서를 섞음

모든 x^i에 대해 순차적으로
  미분값 ∇^i 계산
  i번째 미분값을 이용하여 업데이드 : $\Theta = \Theta - \alpha\nabla^i$

} until convergence

1. 경사하강법

❖ 경사하강법의 약점 보완방법

배치경사하강법
모든 학습자료의 미분값을 평균하여 한꺼번에 업데이트

Repeat {

모든 x^i에 대해 미분값 $\nabla^1, \nabla^2, \nabla^3, \ldots, \nabla^m$계산
미분값의 평균 계산 : $\nabla = \frac{1}{m}\sum_{i=1}^{m} \nabla^i$
미분값의 평균을 이용하여 업데이트 : $\Theta = \Theta - \alpha\nabla$

} until convergence

1. 경사하강법

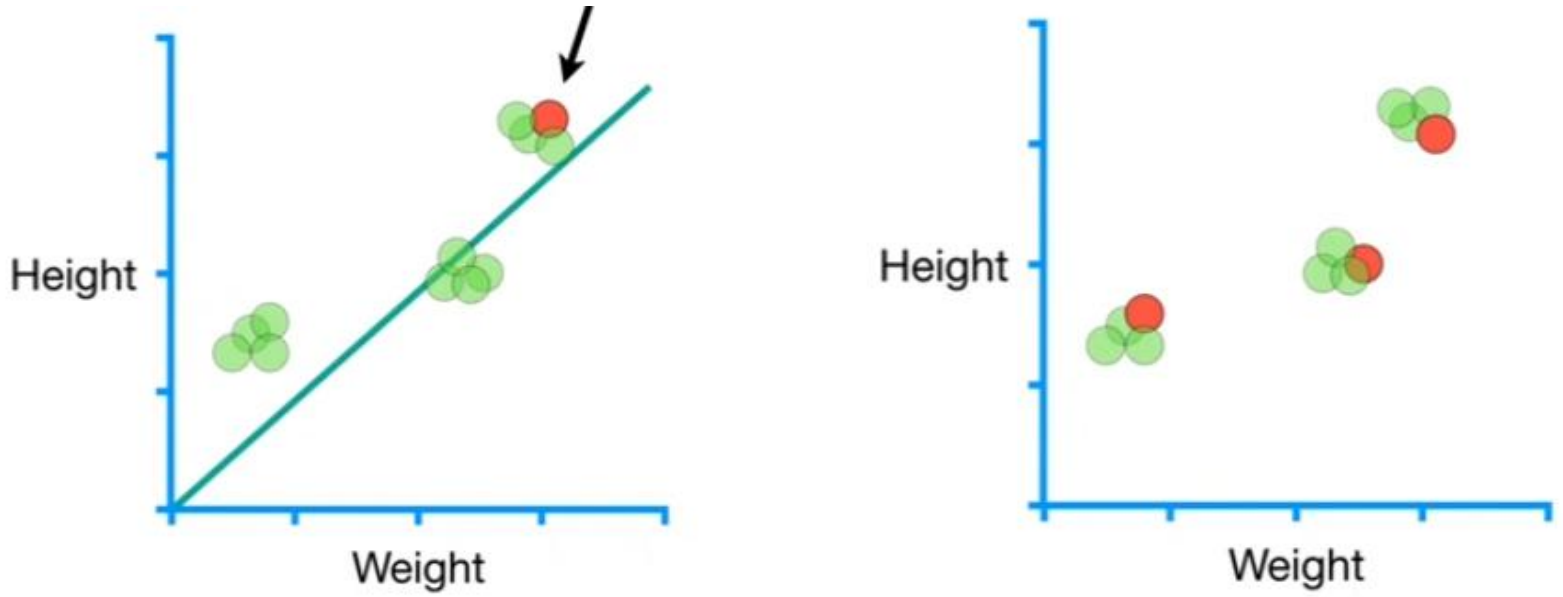
❖ 경사하강법의 약점 보완방법

- Mini-batch Gradient Descent
  - ✓ 임의의 작은 샘플세트에 대해 계산
  - ✓ 확률적 경사 하강법에 비해 matrix 연산에 최적화
  - ✓ SGD와 BGD의 절충안으로 배치 크기를 줄여 확률적 경사하강법을 이용하는 방법
  - ✓ 전체 데이터를 batch_size개씩 나눠 배치로 학습 시키는 방법이고, 배치 크기는 사용자가 지정
    - 예를 들어, 1,000개인 학습 데이터 셋에서 batch_size를 100으로 잡았으면 총 10개의 mini batch가 나오게 됨
    - 이 100개씩의 mini batch를 갖고 한 번씩 SGD를 진행
  - ✓ 파라미터 공간에서 덜 불규칙하게 학습
  - ✓ Local minimum에 빠지면 빠져나오기 힘듬

1. 경사하강법

❖ 경사하강법의 약점 보완방법

➢ Mini-batch Gradient Descent



1. 경사하강법

❖ 예제

```
def gradient_descent(f, fp, x0, learning_rate, max_Iter):
    paths = []   # 비어 있는 list

    print("{0:02d} : {1:5.5f}, {2:6.5f}".format(0, x0, f(x0)))

    for i in range(max_Iter):
        x1 = x0 - learning_rate * fp(x0)
        # x를 학습률로 지정한 거리만큼 이동
        print('{0:02d} : {1:5.5f}, {2:6.5f}'.format(i+1, x1, f(x1)))

        # x0에 x1의 값을 대입해주며 갱신해주고, 다시 반복
        x0 = x1
        paths.append(x0)   # 최적값을 찾아가는 각 점을 모은 것
    return(x0, f(x0), np.array(paths))
```


1. 경사하강법

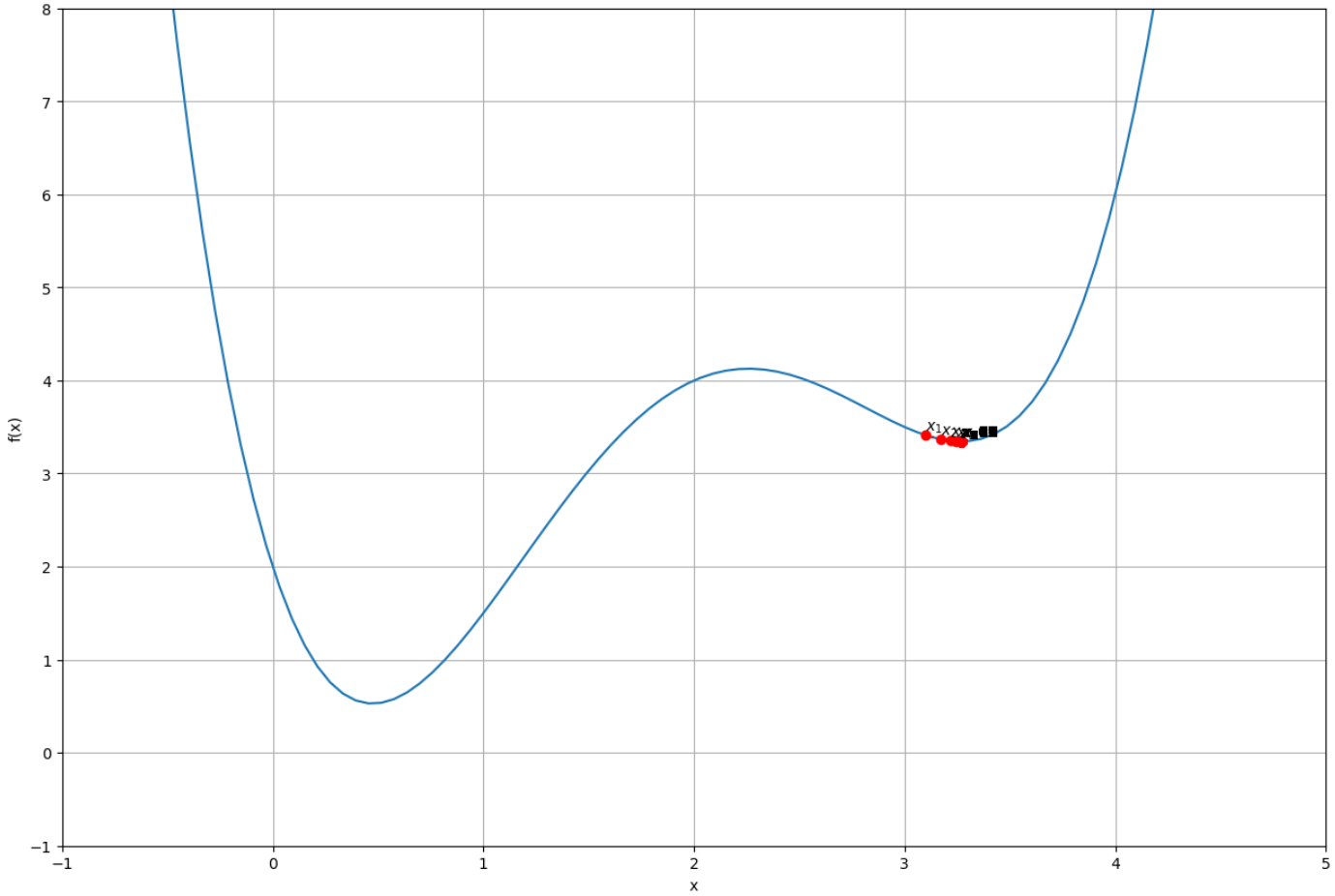
❖ 예제

```
def f(x):
    return  0.5*x**4 - 4*x**3 + 10*x**2 - 7*x + 2
def fp(x):
    return 2*x**3 - 12*x**2 + 20*x – 7

xopt, fopt, paths = gradient_descent(f, fp, 4, 0.1, 200)

x = np.linspace(-1, 5, 100)
y = f(x)
plt.figure(figsize=(15, 10))
plt.plot(x, f(x))
plt.plot(paths, f(paths), 'ro:')
for k, point in enumerate(paths):
    plt.text(point, f(point), '$x_{0}$'.format(k+1), verticalalignment='bottom')
plt.grid()
plt.xlabel('x')
plt.ylabel('f(x)')
plt.xlim(-1, 5)
plt.ylim(-1.0, 8.0)
plt.show()
```


1. 경사하강법

❖ 예제

xopt, fopt, paths = gradient_descent(f, fp, 4, 0.1, 200)

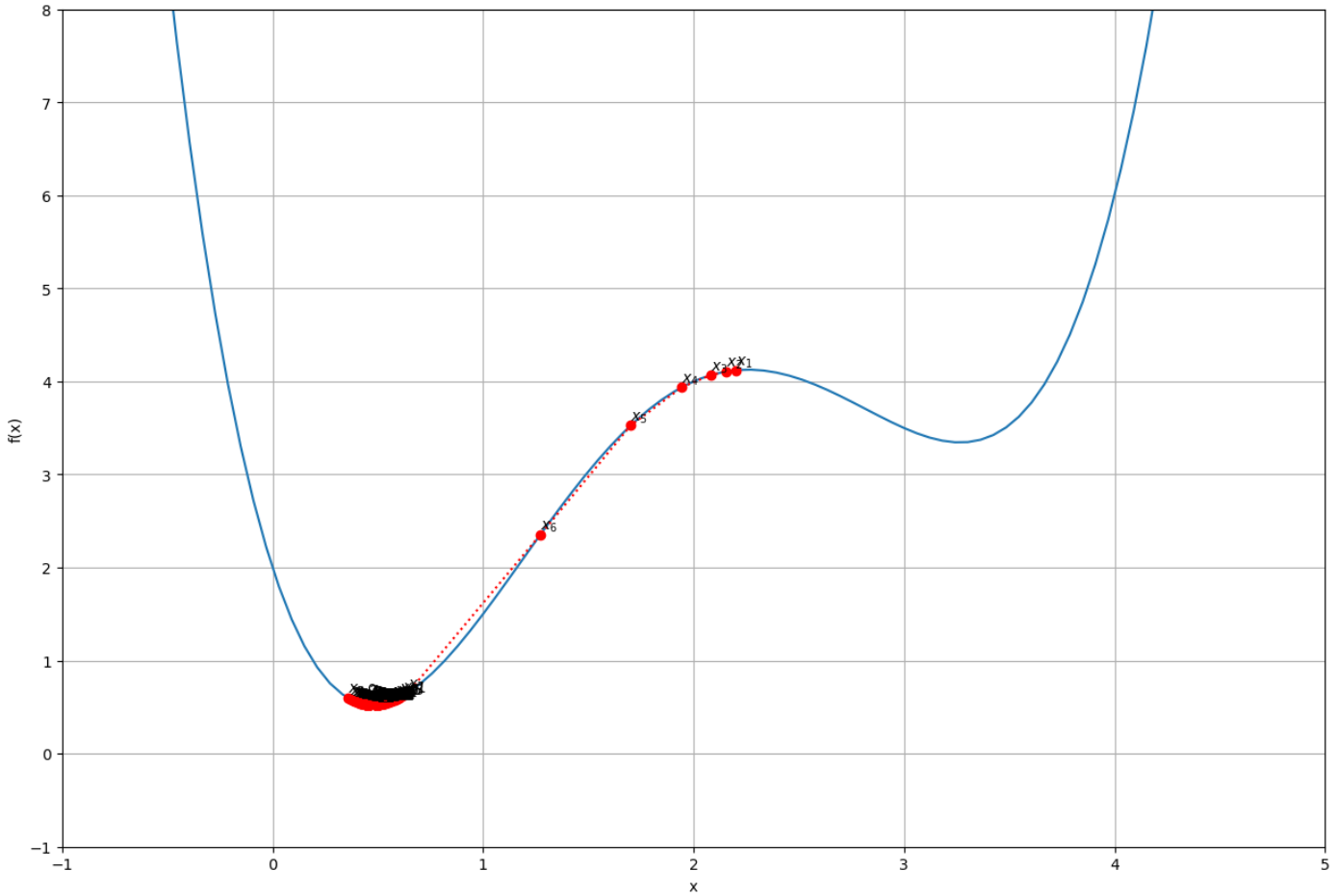


JEONJU UNIVERSITY

1. 경사하강법

❖ 예제

xopt, fopt, paths = gradient_descent(f, fp, 4, 0.2, 200)

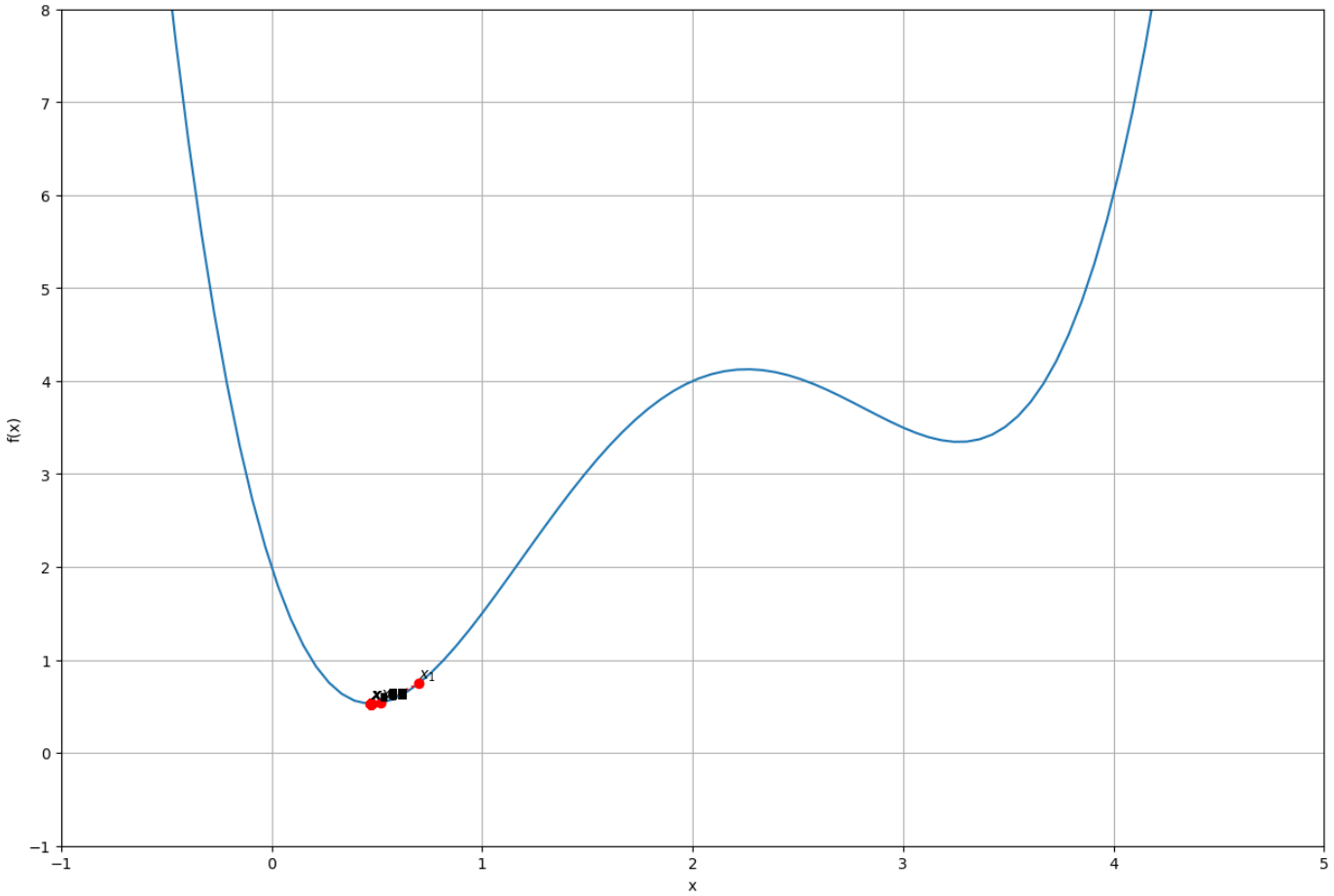


JEONJU UNIVERSITY

1. 경사하강법

❖ 예제

xopt, fopt, paths = gradient_descent(f, fp, 5, 0.1, 200)



JEONJU UNIVERSITY

Thank You !

JEONJU UNIVERSITY
The Place for Superstars