

Week2-2

1. 학습정리

1. 경사하강법

1. 미분

1. 코드

```
1. import sympy as sym
2. from sympy.abc import x
3. sympy.diff(sym.poly(x**2 + 2*x + 3), x)
```

2. 미분은 함수의 최적화에 사용

1. 함수를 증가 시키고 시으면 미분값을 더해주고 반대면 빼줌

3. 경사상승법 (gradient ascent)

1. 미분값을 더해줌

2. 함수의 극대값의 위치를 구할 때 사용

4. 경사하강법 (gradient descent)

1. 미분값을 빼줌

2. 함수의 극소값의 위치를 찾을 때 사용

3. 극 값에 도달하면 미분값이 0이 되서 이제 움직이지 않게 됨

5. 변수가 벡터일 때

1. 각 변수별로 편미분을 계산한 그레디언트 벡터를 이용

$$\nabla f = (\partial_{x_1} f, \partial_{x_2} f, \dots, \partial_{x_d} f)$$

요 기호를 **nabla** 라 부릅니다



앞서 사용한 미분값인 $f'(x)$ 대신 벡터 ∇f 를 사용하여 변수 $\mathbf{x} = (x_1, \dots, x_d)$ 를 동시에 업데이트 가능합니다

6. 선형회귀

1. 가장 근사한 선형 모델을 찾는 것

2. 비선형회귀는 목적식이 블록하지 않을 수 있어 수렴이 항상 보장되지는 않음

7. 확률적 경사하강법 (SGD : Stochastic Gradient Descent)

1. 모든 데이터 대신 데이터 일부(미니배치)만 사용

2. 극소점에서 탈출할 수 있어서 최소점을 찾을 수 있음

3. 그냥 경사하강법 보다 범용성있고 효율적

4. 하드웨어의 한계를 벗어날 수 있음

2. 피어세션

1. 수업 리뷰 & 궁금한 점 질문

1. 경사하강법이 무엇인가

2. 경사하강법 수식에 대한 질답

2. 알고리즘 문제 코드 설명 & 리뷰

1. <https://leetcode.com/problems/remove-duplicate-letters/>

2. <https://programmers.co.kr/learn/courses/30/lessons/42884>