6-3. Hessian 행렬: 2차 미분을 이용한 함수의 곡률과 최적화 문제에서의 응용

1. Hessian 행렬의 개념

   Hessian 행렬은 다변수 함수의 2차 미분으로 구성된 행렬입니다. 이 행렬은 함수의 곡률(변화율의 변화)을 이해하고 최적화 문제에서 중요한 역할을 합니다.

   Hessian 행렬의 정의

   함수 $f(x_1, x_2, ..., x_n)$가 n개의 변수 $x_1, x_2, ..., x_n$를 가질 때, Hessian 행렬은 2차 미분의 모든 조합으로 구성된 nxn 행렬입니다. 수학적으로 다음과 같이 정의됩니다.

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

   Hessian 행렬은 다변수 함수의 두 변수에 대한 모든 2차 미분 값을 포함합니다. 각 항목은 특정 변수 쌍에 대한 이차 미분을 나타냅니다.

2. Hessian 행렬과 햄수의 곡률

   Hessian 행렬은 함수의 곡률을 분석하는 데 유용합니다. 특정 점에서 Hessian 행렬을 통해 해당 점에서 함수의 변화 양상을 분석할 수 있습니다.

   - 양의 정부호 (Positive Definite): 모든 고유값이 양수일 때, 함수는 해당 점에서 볼록(convex)하게 변합니다.
   - 음의 정부호 (Negative Definite): 모든 고유값이 음수일 때, 함수는 해당 점에서 오목(concave)하게 변합니다.
   - 양과 음이 섞인 경우 (Indefinite): 함수가 해당 점에서 안장점을 가집니다.

3. 최적화 문제에서의 Hessian 행렬 응용

   최적화 문제에서는 주어진 함수가 최소값 또는 최대값을 가지는 점을 찾는 것이 목표입니다.

   Hessian 행렬을 통해 해당 점이 극값인지, 그리고 그 극값이 최소값인지, 최대값인지를 판별할 수 있습니다.

   - 극값의 존재 : Hessian 행렬의 고유값이 모두 양수인 경우, 해당 점은 지역 최소값을 나타냅니다. 고유값이 모두 음수인 경우, 해당 점은 지역 최대값을 나타냅니다.
   - 안장점 : 고유값이 양수와 음수가 섞여 있으면, 해당 점은 안장점이며 극값이 아닙니다.

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
from sympy import symbols, diff, Matrix

# 윈도우 사용자는 아래 주석 해제
plt.rc('font', family='Malgun Gothic')

# 마이너스 폰트 깨지는 문제에 대한 대처
plt.rcParams['axes.unicode_minus'] = False


# 심볼릭 변수를 정의
x, y = symbols('x y')

# 다변수 함수 정의: f(x, y) = x^2 + y^2
f = x**2 + y**2

# 1차 미분 구하기
f_x = diff(f, x)
f_y = diff(f, y)

# 2차 미분을 구해 Hessian 행렬 계산
f_xx = diff(f_x, x)
f_xy = diff(f_x, y)
```

```
f_yx = diff(f_y, x)
f_yy = diff(f_y, y)

# Hessian 행렬 정의
Hessian = Matrix([[f_xx, f_xy], [f_yx, f_yy]])

print("Hessian 행렬:")
print(Hessian)

# 특정 점에서 Hessian 행렬의 값을 계산 (예: x=1, y=1)
Hessian_at_point = Hessian.subs({x: 1, y: 1})
print("\nHessian 행렬 at (1, 1):")
print(Hessian_at_point)

# Hessian의 고유값을 계산하여 곡률 분석
eigenvalues = Hessian_at_point.eigenvals()
print("\nHessian 행렬의 고유값:")
print(eigenvalues)



Hessian 행렬:
Matrix([[2, 0], [0, 2]])

Hessian 행렬 at (1, 1):
Matrix([[2, 0], [0, 2]])

Hessian 행렬의 고유값:
{2: 2}
```

Hessian행렬을 통해 다음과 같은 정보를 알 수 있습니다.

- 함수의 곡률 : 고유값이 모두 양수이면 해당 점에서 함수는 볼록합니다.
- 최적화 문제에서의 응용: 뉴턴법과 같은 알고리즘에서 Hessian 행렬은 함수의 2차 정보(곡률)를 이용해 수렴 속도를 빠르게 하는데 기여합니다.