3-5. 편미분(Parial Derivatives): 다변수 함수의 미분, 딥러닝에서 사용되는 다변수 최적화와 연결

편미분은 다변수 함수에서 각 변수에 대해 독립적으로 미분하는 과정입니다.다변수 함수는 여러 입력 변수에 의존하는 함수이기 때문에, 특정 변수에 대한 변화율을 계산할 때 나머지 변수들은 고정된 상태로 고려됩니다.

편미분은 딥러닝에서 다변수 최적화 문제를 해결하는데 필수적인 역할을 합니다.

1. 편미분의 정의

   다변수 함수 $f(x, y, z, ...)$에서 특정 변수에 대한 편미분을 계산할 때는 그 변수만 변화시키고 나머지 변수들은 고정된다고 가정합니다. 편미분의 표기는 다음과 같습니다.

   $$\frac{\partial f}{\partial x}$$

   이는 함수 f를 변수 x에 대해 미분하되, 나머지 변수들은 상수로 취급한다는 말입니다.

   예를 들어, 함수 $f(x, y) = x^2 + 3xy + y^2$의 경우

   - x에 대한 편미분은 $\frac{\partial f}{\partial x} = 2x + 3y$
   - y에 대한 편미분은 $\frac{\partial f}{\partial y} = 3x + 2y$

2. 딥러닝에서의 편미분

   딥러닝에서의 신경망은 다변수 함수로 나타낼 수 있습니다. 네트워크의 각 층의 가중치와 편향은 여러 변수로 구성된 함수로 표현되며, 이 가중치들을 최적화하는 것이 학습 과정의 핵심입니다. 신경망의 손실 함수$L(\hat{y}, y)$는 입력 데이터와 네트워크의 가중치에 의해 결정되므로, 각 가중치에 대한 편미분을 계산하여 경사하강법을 통해 최적화를 진행합니다.

   예시 : 단일 층 신경망

   가정

   - 입력 : x_1, x_2

- 가중치 : w_1, w_2
- 출력 : $\hat{y} = w_1 x_1 + w_2 x_2$

손실 함수 $L(\hat{y}, y)$를 최적화하려면 각 가중치 w_1, w_2에 대한 편미분을 계산하여 경사하강법을 사용해 가중치를 업데이트합니다.

3. 다변수 최적화와 경사하강법

   다변수 최적화에서 목표는 다변수 함수의 최소값을 찾는 것입니다. 신경망의 손실 함수 $L(\theta)$에서 매개 변수 θ에 대해 손실을 최소화하는 것이 목표입니다. 이를 위해서는 각 변수에 대한 편미분들 계산하여 기울기를 얻고, 기울기의 반대 방향으로 가중치를 조정합니다.

   경사하강법(Gradient Descent)

   경사하강법은 함수의 기울기를 사용하여 최소값을 찾는 방법입니다. 다변수 함수의 기울기는 모든 편미분들의 벡터로 구성되며, 이를 기울기 벡터(Gradient Vector)라고 부릅니다.

$$\nabla L(\theta) = (\frac{\partial L}{\partial \theta_1}, \frac{\partial L}{\partial \theta_2} ...)$$

   경사하강법에서는 각 매개변수 θ_i를 다음과 같이 업데이트 합니다.

$$\theta_i = \theta_i - \eta \frac{\partial L}{\partial \theta_i}$$

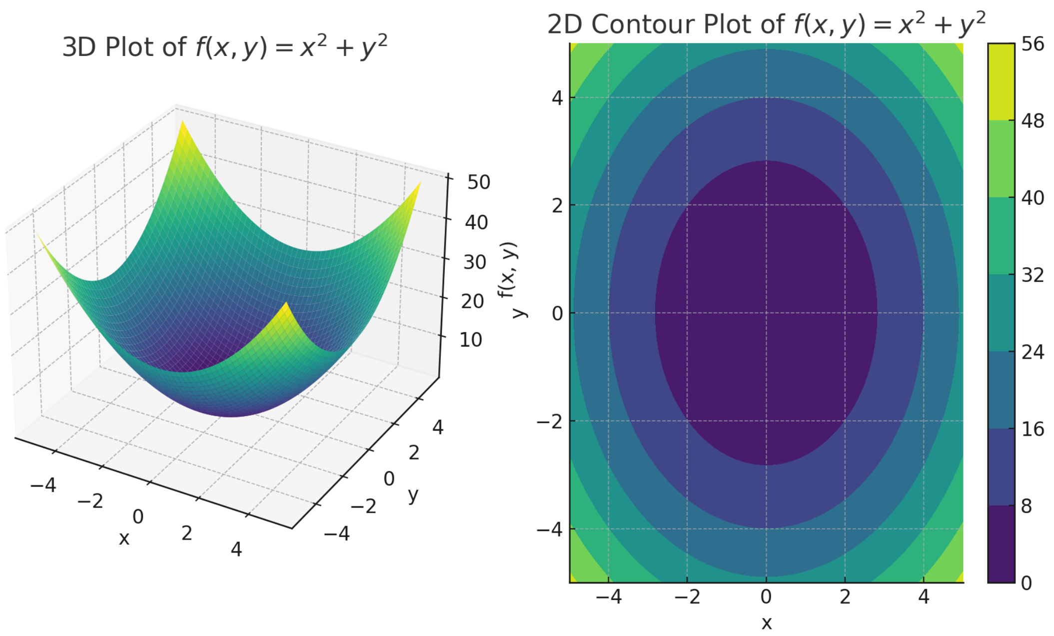
   여기서 η는 학습률 입니다.

4. 편미분의 예제와 그래프

   - 예제함수

     다음과 같은 2변수 함수 $f(x, y) = x^2 + y^2$을 사용하여 편미분을 계산하고 시각화 하겠습니다.

     - $f_x = \frac{\partial f}{\partial x} = 2x$
     - $f_y = \frac{\partial f}{\partial y} = 2y$
       - 이 함수는 2차원에서 볼록한 형태로 (0, 0)에서 최소값을 가집니다.



- 3D 그래프는 $f(x, y) = x^2 + y^2$함수의 모양을 직관적으로 보여줍니다. 이 함수는 원형 대칭성을 가지며, 중심에서 최솟값을 가집니다.
- 2D 등고선 그래프는 높이가 일정한 선을 그리며, 각 등고선은 같은 함수 값을 나타냅니다. 중심으로 갈수록 낮은 값을 나타내며, 이는 경사하강법에서 최솟값으로 수렴하는 경향을 보여줍니다.

위의 그래프는 다변수 함수 $f(x, y) = x^2 + y^2$에 대한 시각화입니다.

- 3D그래프는 함수의 표면을 보여주며 이 함수는 원형 대칭성을 가지고 있으며 중심에서 최솟값을 가집니다. 이는 경사하강법을 사용할 때 기울기가 작아지는 방향으로 이동하는 형태를 잘 보여줍니다.
- 2D 등고선 그래프는 높이가 일정한 선을 그려 각 등고선이 같은 함수 값을 나타냅니다. 등고선이 밀집된 부분은 기울기가 큰 부분을 의미하고, 넓은 부분은 완만한 구간을 나타냅니다.

편미분을 통해 함수의 변화율을 계산하고, 경사하강법을 적용하여 함수의 최소값을 찾는 과정에서 이러한 시각화가 유용합니다.

```
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import sympy as sp
```

```
# 변수 정의
x, y = sp.symbols('x y')

# 함수 정의
f = x**2 + y**2

# 편미분 계산
f_x = sp.diff(f, x)
f_y = sp.diff(f, y)

print(f"x에 대한 편미분: {f_x}")
print(f"y에 대한 편미분: {f_y}")

# 2D 및 3D 그래프용 데이터 생성
x_vals = np.linspace(-5, 5, 400)
y_vals = np.linspace(-5, 5, 400)
X, Y = np.meshgrid(x_vals, y_vals)
Z = X**2 + Y**2

# 3D 그래프
fig = plt.figure(figsize=(12, 6))
ax = fig.add_subplot(121, projection='3d')
ax.plot_surface(X, Y, Z, cmap='viridis')

ax.set_title("3D Plot of $f(x, y) = x^2 + y^2$")
ax.set_xlabel("x")
ax.set_ylabel("y")
ax.set_zlabel("f(x, y)")

# 2D 그래프
ax2 = fig.add_subplot(122)
contour = ax2.contourf(X, Y, Z, cmap='viridis')
fig.colorbar(contour)

ax2.set_title("2D Contour Plot of $f(x, y) = x^2 + y^2$
ax2.set_xlabel("x")
ax2.set_ylabel("y")
```

```
plt.show()
```

- SymPy를 사용한 편미분 계산 : sp.diff(f, x)와 sp.diff(f, y)를 사용해 $f(x, y) = x^2 + y^2$함수의 편미분을 각각 계산합니다.
- 2D 및 3D 그래프 시각화 : matplotlib을 사용하여 함수의 3차원 표면과 2차원 등고선(contour)그래프를 시각화합니다.
    - 3D 그래프에서는 함수 $f(x, y)$의 곡면을 시각화합니다.
    - 2D 그래프는 함수의 등고선을 보여주며, 각 높이를 색상으로 구분합니다.