5-3. 경사 하강법(Gradient Descent): 손실 함수 최소화를 위한 최적화 방법론

경사 하강법은 머신러닝과 딥러닝에서 모델의 성능을 최적화하는 가장 기본적인 알고리즘 중 하나입니다. 주로 손실 함수(Loss Function)의 값을 최소화하기 위해 사용되며, 이는 모델이 주어진 데이터를 잘 설명하도록 가중치(weight)를 조정하는 과정에서 중요한 역할을 합니다. 경사 하강법은 함수의 기울기(gradient)를 따라 이동하면서 최솟값에 도달하려고 합니다.

1. 경사 하강법의 기본 개념

   경사 하강법은 손실 함수 $f(x)$가 있을 때, 이 함수를 최소화하는 최적의 매개변수 x를 찾는 과정입니다. 여기서 $f(x)$는 함수의 값이며, 경사는 함수의 특정 지점에서의 기울기를 의미합니다. 경사 하강법은 기울기를 따라 반대 방향으로 조금씩 이동하면서 손실 함수를 최소화합니다.

   경사 하강법의 수식

   기본적인 경사 하강법의 수식은 다음과 같습니다.

   $$xnew = xold - \eta \cdot \nabla f(xold)$$

   - $\nabla f(x)$는 함수 $f(x)$의 기울기(도함수 입니다.
   - η는 학습률(learning rate)로, 이동하는 크기를 조절합니다.
   - x_{new}는 기울기를 이용해 업데이트된 새로운 매개변수입니다.

2. 경사 하강법의 딥러닝에서의 활용

   딥러닝에서는 모델의 매개변수(가중치와 편향)를 최적화하기 위해 손실 함수(Loss Function)를 최소화하는 것이 목표입니다. 이때 경사하강법을 사용하여 손실 함수의 기울기를 계산하고, 그 기울기를 따라 매개변수를 업데이트합니다.

손실 함수는 예측값과 실제값의 차일르 나타내며, 이 차이를 줄여가는 과정에서 경사 하강법이 사용됩니다. 경사 하강법의 변형으로는 배치 경사 하강법(Batch Gradient Descent), 확률적 경사 하강법(Stochastic Gradient Descent), 그리고 미니 배치 경사 하강법(Mini-batch Gradient Descent)등이 있습니다.

- 배치 경사 하강법 : 전체 데이터셋을 사용하여 한 번의 업데이트를 수행하는 방법
- 확률적 경사 하강법(SGD) : 매년 하나의 샘플만 사용하여 업데이트를 수행하는 방법
- 미니 배치 경사 하강법 : 일부 샘플(미니 배치)을 사용하여 업데이트를 수행하는 방법

3. 경사 하강법의 작동 방식

   경사 하강법은 다음과 같은 과정을 거칩니다.

   1. 초기화 : 모델의 매개변수(가중치)를 임의의 값으로 초기화합니다.
   2. 손실 함수 계산 : 현재 매개변수에 따른 손실 함수를 계산합니다.
   3. 기울기 계산 : 손실 함수의 기울기를 계산합니다. 이때 기울기는 매개변수를 업데이트할 방향을 나타냅니다.
   4. 매개변수 업데이트 : 기울기를 따라 매개변수를 업데이트합니다. 매개변수는 기울기 방향의 반대쪽으로 움직입니다.
   5. 수렴 : 손실 함수의 값이 더 이상 크게 변하지 않을 때까지 2~4 과정을 반복합니다.

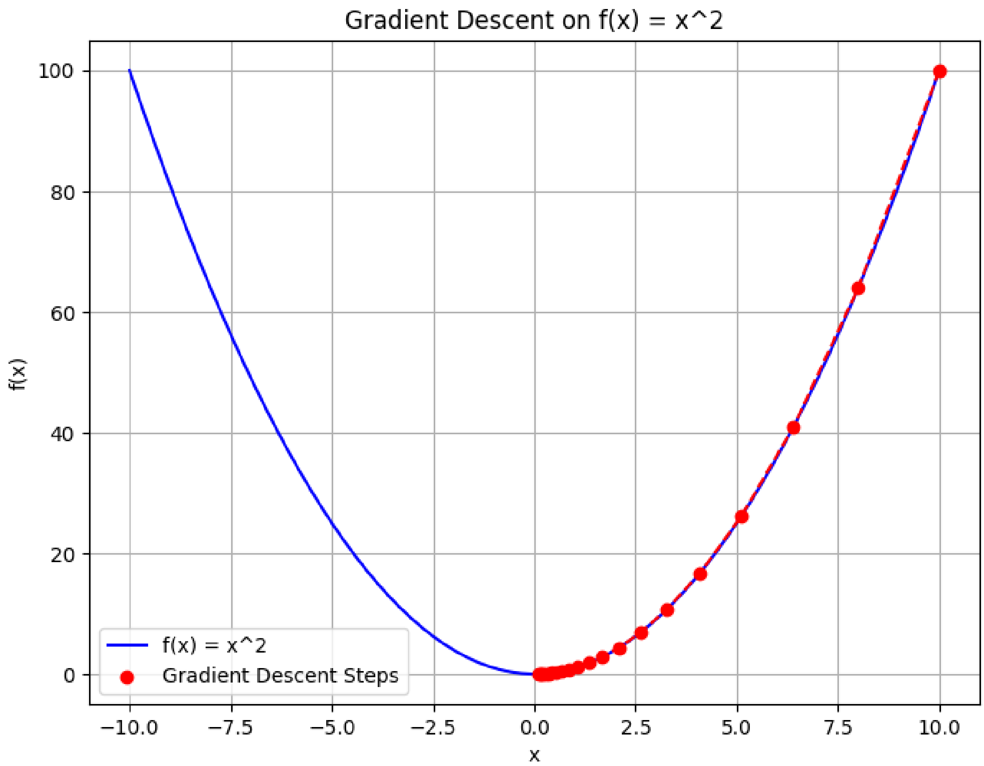
4. 경사 하강법의 문제점 및 개선

   경사 하강법은 손실 함수가 잘 정의된 경우에 잘 작동하지만, 다음과 같은 문제를 가질 수 있습니다.

   - 학습률 설정 : 학습률이 너무 크면 발산하고, 너무 작으면 수렴 속도가 느려집니다.
   - 지역 최적점 : 손실 함수가 여러 개의 지역 최소값을 가질 경우, 전역 최소값(Global Mimimum)에 도달하지 못하고 지역 최소값(Local Minimum)에 머무를 수 있습니다.
   - 지그재그 현상 : 기울기가 너무 급격히 변하는 경우, 매개변수가 지그재그로 이동하며 수렴 속도가 느려질 수 있습니다.

이러한 문제를 해결하기 위해 모멘텀(Momentum), Adam 최적화(Adam Optimizer) 등의 개선된 최적화 방법이 존재합니다.

- 그래프 : 간단한 2차 함수 $f(x) = x^2$을 최소화하는 예제 구현 , 함수의 소값은 $x = 0$에서 발생합니다.



- 목적함수 : $f(x) = x^2$는 매우 간단한 함수로, 이를 최소화하는 것이 목표입니다.
- 도함수(기울기): 이 함수의 기울기는 $f'(x) = 2x$입니다.
- 경사 하강법을 통해 x의 값을 점진적으로 업데이트하여 최솟값을 찾아갑니다. 학습률 $\eta = 0.1$로 설정했고, 초기값 $x = 10$에서 시작합니다.

```
import numpy as np
import matplotlib.pyplot as plt

# 목적 함수: f(x) = x^2
def f(x):
    return x**2

# 목적 함수의 도함수 (기울기)
```

```
def f_prime(x):
    return 2*x

# 경사 하강법 구현
def gradient_descent(learning_rate=0.1, epochs=20, initial_
    x = initial_x
    history = [x]  # 각 epoch의 x값 기록
    for i in range(epochs):
        grad = f_prime(x)  # 기울기 계산
        x = x - learning_rate * grad  # x 업데이트
        history.append(x)
        print(f"Epoch {i+1}: x = {x}, f(x) = {f(x)}")
    return history

# 경사 하강법 실행
history = gradient_descent()

# 결과 시각화
x_vals = np.linspace(-10, 10, 100)
y_vals = f(x_vals)

plt.figure(figsize=(8, 6))
plt.plot(x_vals, y_vals, label='f(x) = x^2', color='blue')
plt.scatter(history, [f(x) for x in history], color='red',
plt.plot(history, [f(x) for x in history], color='red', li
plt.title("Gradient Descent on f(x) = x^2")
plt.xlabel('x')
plt.ylabel('f(x)')
plt.legend()
plt.grid(True)
plt.show()
```

- 이 코드를 실행하면 x값이 점점 줄어들면서 최소값인 $x = 0$으로 수렴하는 과정을 확인할 수 있습니다. 또한 그래프에서는 경사 하강법이 $f(x) = x^2$함수에서 어떻게 이동하는지를 시각화할 수 있습니다.