

# 가우스 소거법과 경사하강법

👤 생성자	👤 재환 김
🏷️ 태그	엔지니어링

## 1. 가우스 소거법 (Gaussian Elimination)

### 1.1. 목적

가우스 소거법은 **선형 방정식의 해를 구하는 방법**입니다. 주로 **선형 대수학**에서 시스템의 방정식을 풀 때 사용되며, 여러 개의 선형 방정식을 행렬 형태로 정리한 후 이를 **행렬의 계단식 형태(Row-Echelon Form)** 또는 기약 행렬 형태(Reduced Row-Echelon Form)로 변환하여 해를 구합니다.

### 1.2. 동작 방식

가우스 소거법은 행렬의 연산을 통해 시스템을 간단하게 만들고, 이를 통해 변수들의 값을 차례대로 구해나가는 과정입니다. 기본적으로 다음 단계를 거칩니다:

- **행 연산**: 하나의 행을 다른 행에 더하거나 빼서 원하는 계단식 형태로 변환합니다.
- **전진 소거(Forward Elimination)**: 상삼각 행렬로 변환하여 아래쪽에 있는 항들을 소거합니다.
- **후진 대입(Back Substitution)**: 전진 소거 후 남은 방정식을 역으로 계산하여 변수 값을 구합니다.

가우스 소거법은 **정확한 해**를 구하는 데 중점을 둔 알고리즘입니다.

### 1.3. 적용 예시

$$2x + 3y = 54x + 6y = 10$$

이런 선형 방정식이 주어졌을 때, 가우스 소거법을 사용하여 행렬의 형태로 변환하고, 상삼각 행렬을 만들어서  $x$ 와  $y$ 의 값을 구합니다.

### 1.4. 특징

- **목표**: 선형 방정식의 **정확한 해**를 구하는 것.
- **사용 범위**: 선형 시스템에서 매우 유용.

- **연산 방식:** 행렬의 행 연산을 통해 시스템을 간단히 만들어 해를 찾음.
- **해:** 유일하거나 여러 개일 수 있으며, 정확한 값을 구함.

## 2. 경사 하강법 (Gradient Descent)

### 2.1. 목적

경사 하강법은 **최적화 문제**에서 **함수의 최소값**(또는 최대값)을 찾기 위해 사용되는 방법입니다. 주로 **비선형 함수**의 최적화에서 많이 쓰이며, 특히 딥러닝에서는 **오차를 최소화**하기 위해 경사 하강법을 사용하여 가중치를 학습합니다.

### 2.2. 동작 방식

경사 하강법은 함수의 기울기(그래디언트)를 계산하여 기울기가 낮은 방향(최소값 방향)으로 점진적으로 이동하는 방식입니다. 기본적인 경사 하강법의 과정은 다음과 같습니다:

- **초기화:** 임의의 초기 값을 선택.
- **기울기 계산:** 각 점에서 함수의 기울기(변화율)를 계산.
- **값 갱신:** 기울기가 가장 낮은 방향으로 값을 갱신하며 함수의 값을 점점 줄여나감.
- **수렴:** 값이 더 이상 크게 변하지 않을 때까지 반복.

함수  $f(x)$ 의 기울기를 따라 이동하는 것은 다음과 같이 표현됩니다:

$$x_{new} = x_{old} - \eta \cdot \nabla f(x)$$

여기서,  $\eta$ 는 학습률(learning rate),  $\nabla f(x)$ 는 기울기(gradient)입니다.

### 2.3. 적용 예시

비선형 함수  $f(x) = (x-3)^2 + 2f(x)$ 의 최소값을 찾고자 할 때, 경사 하강법은 초기 값을 정하고, 기울기를 계산하여 점차  $x=3$ 에서 최소값을 찾는 방식으로 진행됩니다.

### 2.4. 특징

- **목표:** 함수의 **최소값**(또는 최대값)을 찾는 것.
- **사용 범위:** 비선형 문제에서 매우 유용하며, 딥러닝에서 주로 사용.
- **연산 방식:** 기울기를 계산하여 점진적으로 최소값 방향으로 이동.
- **해:** 최적화된 해를 점진적으로 구하며, 정확한 해가 아니라 근사값을 구할 수 있음.

## 3. 가우스 소거법과 경사 하강법의 차이점

구분	가우스 소거법	경사 하강법
목적	선형 방정식의 <b>정확한 해</b> 를 구함	비선형 함수의 최소값(또는 최대값)을 찾음
적용 분야	선형 대수학에서 선형 방정식 풀이	최적화 문제, 머신러닝, 딥러닝
알고리즘 방식	행렬의 행 연산을 통해 직접적으로 해를 계산	기울기를 따라 함수의 최소값을 점진적으로 찾음
해	<b>정확한 해</b> (유일하거나 여러 개)	<b>근사 해</b> (점진적으로 최적화된 값)
속도	정확한 해를 구하기 위해 행렬의 크기에 따라 시간이 소요됨	학습률에 따라 속도가 달라지며, 반복적으로 근사값을 찾음
특징	선형 시스템에서 사용, 행렬 연산 기반	비선형 함수에서 사용, 반복적 최적화 알고리즘

## 4. 결론

- **가우스 소거법**은 **선형 방정식**을 풀기 위한 방법으로, 행렬을 단순화하여 **정확한 해**를 구하는 것이 목표입니다.
- **경사 하강법**은 주로 **비선형 최적화**에서 사용되며, 함수의 기울기를 따라 **최소값**을 점진적으로 찾아가는 방식입니다.