**به نام خدا**

**تفاوت دو الگوریتم :**

| **روش** | **دسته‌بندی اصلی** | **زیردسته** | **کاربرد اصلی** |
| --- | --- | --- | --- |
| **معادله نرمال** | یادگیری نظارت‌شده | رگرسیون خطی تحلیلی | مدل‌سازی پارامتری |
| **Gradient Descent** | بهینه‌سازی | الگوریتم‌های تکراری | آموزش مدل‌های پارامتری |

| **مشخصه** | **معادله نرمال (Normal Equation)** | **گرادیان کاهشی (Gradient Descent)** | **گرادیان کاهشی تصادفی (SGD)** | **گرادیان کاهشی مینی‌بچ (Mini-Batch GD)** |
| --- | --- | --- | --- | --- |
| **نوع حل** | تحلیلی (یک مرحله‌ای) | عددی (تکراری) | عددی (تکراری) | عددی (تکراری) |
| **پیچیدگی محاسباتی** | O(n³) - برای معکوس ماتریس | O(n) در هر تکرار | O(1) در هر تکرار | O(k) - k اندازه بچ |
| **حافظه مورد نیاز** | بالا (ذخیره ماتریس XᵀX) | متوسط | کم | متوسط |
| **دقت** | دقیق (حل بهینه) | وابسته به نرخ یادگیری و تکرارها | نوسان بیشتر | تعادل بین دقت و سرعت |
| **مناسب برای** | داده‌های کوچک (n < 10,000) | داده‌های بزرگ | داده‌های بسیار بزرگ | داده‌های بزرگ با منابع محدود |
| **نیاز به Feature Scaling** | خیر | بله | بله | بله |
| **پارامترهای تنظیم** | بدون پارامتر | نرخ یادگیری (α)، تعداد تکرارها | نرخ یادگیری، تعداد تکرارها | نرخ یادگیری، اندازه بچ، تعداد تکرارها |
| **همگرایی** | همگرایی در یک مرحله | همگرایی بهینه با نرخ یادگیری مناسب | همگرایی با نوسان | همگرایی پایدارتر از SGD |
| **پیاده‌سازی در Scikit-Learn** | LinearRegression(solver='normal') | SGDRegressor(learning\_rate='constant') | SGDRegressor() | SGDRegressor(batch\_size=32) |
| **مزایا** | - بدون نیاز به تنظیم پارامتر - دقیق برای داده کوچک | - مناسب داده‌های بزرگ - انعطاف‌پذیر | - بسیار سریع برای داده‌های حجیم | - تعادل بین سرعت و دقت |
| **معایب** | - کند برای داده‌های بزرگ - مشکل با ماتریس‌های تکین | - نیاز به تنظیم نرخ یادگیری - همگرایی آهسته | - نوسان در همگرایی - نیاز به تنظیم دقیق | - تنظیم پیچیده‌تر |

**نکات کلیدی که شما به درستی یاد گرفتید**:

1. **هدف از Normal Equation و Gradient Descent**:
   * هر دو روش برای **یافتن پارامترهای بهینه** (وزن‌ها یا ضرایب) در مدل‌های یادگیری ماشین استفاده می‌شوند.
   * شرط اساسی: **رابطه ریاضی بین**X**(ویژگی‌ها) و**y**(هدف)** وجود داشته باشد (مثل مدل‌های خطی).
2. **مقایسه دو روش**:

| **مشخصات** | **Normal Equation** | **Gradient Descent** |
| --- | --- | --- |
| **نوع مسئله** | مدل‌های خطی (مثل رگرسیون خطی) | مدل‌های خطی و غیرخطی (شبکه‌های عصبی) |
| **پیچیدگی محاسباتی** | مناسب برای داده‌های کوچک (n\_features < 10k) | مناسب برای داده‌های بزرگ |
| **سرعت** | سریع (محاسبه مستقیم) | کندتر (تکرارشونده) |
| **مزایا** | بدون نیاز به تنظیم نرخ یادگیری | قابل استفاده در مدل‌های پیچیده‌تر |

1. **مثال‌های کاربردی**:
   * **Normal Equation**:
2. theta = np.linalg.inv(X.T @ X) @ X.T @ y *# محاسبه پارامترها*

فقط برای **رگرسیون خطی ساده** کاربرد دارد.

* **Gradient Descent**:
* theta = theta - learning\_rate \* gradient *# آپدیت پارامترها در هر تکرار*

در **رگرسیون لجستیک، شبکه‌های عصبی** و... استفاده می‌شود