



## هوش مصنوعی

مدرس: دکتر عادله بیطرافان<sup>۱</sup>

دستیاران آموزشی: هلیا رنجبر<sup>۲</sup>, مهرناز حسینی<sup>۳</sup> و روزین پناو<sup>۴</sup>

موعد تحويل: ۳۰ بهمن ماه

۱۰۰ نمره

پروژه پایانی: فاز اول

هدف از این پروژه حل مسئله رگرسیون خطی با استفاده از مسئله بهینه‌سازی است. این پروژه در ۲ فاز طراحی شده است که فاز دوم آن اختیاری است. در فاز اول می‌آموزید هدف از بهینه‌سازی چیست، چه طور انجام می‌شود و رفتار آن را روی یک مجموعه داده کوچک تحلیل خواهید کرد. در فاز دوم مسئله رگرسیون غیرخطی را روی یک مجموعه داده واقعی با استفاده از الگوریتم گرادیان کاهشی که در فاز اول آموختید حل خواهید کرد.

### فاز اول: مفهوم بهینه‌سازی و الگوریتم گرادیان کاهشی

در این فاز، هدف این است که بفهمیم ماشین‌ها چگونه «یاد می‌گیرند». یادگیری در دنیای هوش مصنوعی معمولاً به معنای پیدا کردن بهترین مقادیر برای پارامترهای یکتابع است.

#### بخش ۱. مقدمه

در علوم داده و علوم کامپیوترا، ما همواره در حال بهینه‌سازی<sup>۵</sup> هستیم. به عنوان مثال، در رگرسیون خطی ما به دنبال بهترین خطی هستیم که از روی داده‌ها عبور می‌کند، در الگوریتم‌های خوشبندی، محل خوشها را بهینه می‌کنیم و در شبکه‌های عصبی به دنبال پیدا کردن بهترین پارامترها برای شبکه هستیم.

در یک جمله، «بهینه‌سازی» به معنای یافتن مقدار بهینه (کمینه<sup>۶</sup> یا بیشینه<sup>۷</sup>) یکتابع است. به عنوان مثال، در تابع زیر چه مقداری از  $x$  باعث می‌شود تابع کمترین مقدار خود را داشته باشد؟

$$f(x) = (x - 2)^2 + 5$$

پاسخ این است که اگر  $x = 2$  باشد، تابع کمترین مقدار خود یعنی 5 را خواهد داشت.

[adeleh.bitarafan@sharif.edu](mailto:adeleh.bitarafan@sharif.edu)<sup>۱</sup>

[helia.ranjbar04@ut.ac.ir](mailto:helia.ranjbar04@ut.ac.ir)<sup>۲</sup>

[mehrnazhosseini4@ut.ac.ir](mailto:mehrnazhosseini4@ut.ac.ir)<sup>۳</sup>

[rozhin.panaw@ut.ac.ir](mailto:rozhin.panaw@ut.ac.ir)<sup>۴</sup>

Optimization<sup>۵</sup>

Minimum<sup>۶</sup>

Maximum<sup>۷</sup>

در این پژوهه، برای حل مسئله بهینه‌سازی از الگوریتم گرادیان کاوهشی<sup>۱</sup> بهره می‌بریم که در ادامه این فاز الگوریتم را توضیح خواهیم داد.

در حالت کلی، فرآیند یادگیری ماشین را می‌توان یک فرآیند سه مرحله‌ای در نظر گرفت:

- **انتخاب مدل:** تعیین می‌کند که چه نوع تابع یا رابطه‌ای قرار است بین داده‌های ورودی ( $X$ ) و خروجی ( $y$ ) برقرار باشد (مثلاً یک مدل خطی که فقط می‌تواند روابط خطی بین ورودی و خروجی را یاد بگیرد).
- **انتخاب تابع زیان<sup>۲</sup>:** معیاری است که به ما می‌گوید مدل فعلی ما چقدر بد عمل می‌کند (به عنوان مثال، آیا مدل A بهتر از مدل B است؟)
- **انتخاب الگوریتم بهینه‌سازی:** روشی است که می‌گردد تا مدل با کمترین مقدار تابع زیان را پیدا کند (مثلاً پیدا کردن بهترین مقادیر برای  $w_1$  و  $w_0$  در مدل خطی  $y = w_0 + w_1 x$ )

## بخش ۲. درک و مفهوم تابع زیان

تابع زیان (که گاهی تابع هزینه یا تابع هدف نیز نامیده می‌شود) ابزاری است که عملکرد مدل ما را به یک عدد واقعی تبدیل می‌کند. هدف ما در یادگیری ماشین، به حداقل رساندن<sup>۳</sup> این عدد است. یکی از رایج‌ترین توابع زیان در مسئله رگرسیون، میانگین مجدد خطاهای<sup>۴</sup> (MSE) است که در حالت کلی فرمول آن به شرح زیر است:

$$MSE = f(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (1)$$

که در فرمول فوق:

- $y_i$  خروجی واقعی<sup>۵</sup> ورودی  $x_i$  است.
- $\hat{y}_i$  خروجی پیش‌بینی شده<sup>۶</sup> توسط مدل است.
- $n$  تعداد کل داده‌هاست.

Gradient Descent<sup>۱</sup>

Loss function<sup>۲</sup>

Minimize<sup>۳</sup>

Mean Squared Error<sup>۴</sup>

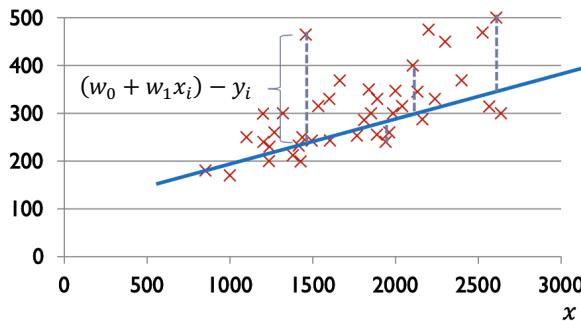
Actual<sup>۵</sup>

Predicted<sup>۶</sup>

اگر مدل ما یک رگرسیون خطی ساده، تابع زیان ما به صورت زیر نوشته خواهد شد که همان‌طور که گفته شد به دنبال کمینه کردن آن هستیم.

$$\min_{w_0, w_1} f(w_0, w_1) = \min_{w_0, w_1} \frac{1}{n} \sum_{i=1}^n ((w_0 + w_1 x_i) - y_i)^2 \quad (2)$$

معادله (۲) به این معناست که در رگرسیون خطی به دنبال پیدا کردن خطی هستیم که فاصله هر داده نسبت به آن کمینه باشد و یا به عبارتی به دنبال پیدا کردن خطی هستیم که با کمترین خطای داده‌ها قرار گیرد. شکل (۱) مسئله رگرسیون خطی را نشان می‌دهد.



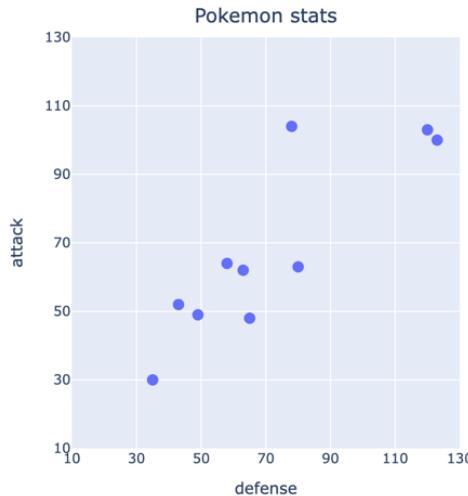
شکل(۱). مسئله رگرسیون خطی. محور افقی داده ورودی  $x$  و محور عمودی خروجی  $y$  را نشان می‌دهد. خط آبی رنگ خط تخمین زده شده را نشان می‌دهد.

بنابراین مسئله بهینه‌سازی در رگرسیون خطی با یک ویژگی، پیدا کردن بهترین مقادیر برای پارامترهای  $w_0$  و  $w_1$  است به طوری که مقدار MSE در معادله (۲) را به حداقل برسانند. در ادامه در بخش ۱، ۳ سعی در پیدا کردن این مقادیر به صورت دستی داریم و در نهایت در بخش ۱، ۴ بهترین مقدار پارامترها را با استفاده از الگوریتم گرادیان کاهشی به دست خواهیم آورد.

### بخش ۳. یافتن بهترین مقادیر پارامترها به صورت دستی (۱۵ نمره)

در این بخش، می‌خواهیم مفهوم بهینه‌سازی را با استفاده از مدل محبوب رگرسیون خطی و داده‌های واقعی بررسی و پیاده‌سازی کنیم. بدین منظور کد نیمه آماده‌ای در فایل Jupyter Notebook در اختیار شما قرار داده شده است که باید آن را کامل کنید.

در این فاز از مجموعه داده کوچک پوکمون<sup>۱</sup> با ۱۰ داده و یک ویژگی (دفاع<sup>۲</sup>) شروع می‌کنیم. داده‌های موجود در این مجموعه داده در شکل (۲) آمده است، همان‌طور که مشاهده می‌شود به ازای هر داده ورودی در محور افقی، خروجی (حمله<sup>۳</sup>) را در محور عمودی داریم. هدف یافتن بهترین خطی است که از میان این ۱۰ داده عبور می‌کند.



شکل(۲). ۱۰ داده‌های موجود در مجموعه داده پوکمون.

بنابراین هدف پیدا کردن خط  $xw_0 + w_1 = y$  و یا به عبارتی پیدا کردن پارامترهای مدل ( $w_0$  و  $w_1$ ) است. راه حل اول می‌تواند تست کردن مقادیر مختلف  $w_0$  و  $w_1$  به صورت شهودی و دستی باشد تا از بین مقادیر مختلف، خطی انتخاب گردد که کمترین تابع زیان MSE را دارد.

- با تغییر دادن پارامترهای مدل ( $w_0$  و  $w_1$ ) به صورت دستی، سعی کنید چند خط رسم کنید که از نظر شهودی با کمترین خطا از میان داده‌های پوکمون عبور می‌کند. آن‌ها را رسم کرده و تابع زیان MSE در معادله (۲) را برای آن‌ها محاسبه کنید تا متوجه شوید هر خط چقدر "بد" عمل کرده است (توجه: تابع MSE را خود پیاده‌سازی کنید و از توابع آماده Sklearn استفاده نکنید).
- تحلیل کنید که خطوط به دست آمده چقدر به نقاط نزدیک هستند؟ و با چند بار تلاش، می‌توانید خطی پیدا کنید که آن بسیار کوچک باشد؟ MSE

---

Pokemon<sup>۱</sup>

Defense<sup>۲</sup>

Attack<sup>۳</sup>

واضح است که پیدا کردن بهترین مقادیر برای پارامترهای  $w_0$  و  $w_1$  در میان بی‌نهایت مقادیری که می‌توانند بگیرند به صورت شهودی و دستی ممکن نخواهد بود و این مشکل با افزایش تعداد داده‌ها و ویژگی‌های آن‌ها چندین برابر می‌شود.

راه حل دوم می‌تواند پیدا کردن بهترین مقادیر  $w_0$  و  $w_1$  به صورت جستجو در فضای جستجو باشد. به عنوان مثال می‌توان از جستجو شبکه‌ای<sup>۱</sup> استفاده کرد. بدین منظور، برای ساده‌تر شدن مسئله، فرض کنید عرض از مبدأ صفر است ( $w_0 = 0$ ) و فقط می‌خواهیم بهترین شبی خط (پارامتر  $w_1$ ) را پیدا کنیم.

- مجموعه‌ای از اعداد مختلف را برای شبی امتحان کنید (به عنوان مثال مقادیر مختلف  $w_1$  را در بازه  $[5, -5]$  در نظر بگیرید) و نموداری رسم کنید که محور افقی آن مقادیر مختلف  $w_1$  و محور عمودی تابع زیان MSE مربوط به خط موردنظر است. سپس بررسی کنید که کدام مقدار از  $w_1$  کمترین خط را تولید می‌کند.
- پایین‌ترین نقطه این منحنی چه چیزی را نشان می‌دهد؟ در نقاطی که منحنی بسیار تند است (دور از جواب بهینه)، شبی خط مماس بر منحنی (مشتق) بزرگ است یا کوچک؟ در پایین‌ترین نقطه منحنی، شبی خط مماس (مشتق) چقدر است؟
- به نظر شما چرا جستجوی شبکه‌ای برای پیدا کردن مقادیر پارامترهای مدل مناسب نیست؟ و توضیح دهید که چرا این روش برای مدل‌های بزرگ (با میلیون‌ها پارامتر) غیرممکن است.

#### بخش ۴. یافتن بهترین مقادیر پارامترها با استفاده از گرادیان کاهشی

در این بخش، ابتدا مفهوم گرادیان<sup>2</sup> و گرادیان کاهشی را خواهیم گفت، سپس به معرفی الگوریتم گرادیان کاهشی خواهیم پرداخت. درنهایت بهترین خطی که می‌توان از مجموعه داده پوکمون عبور داد را با استفاده از الگوریتم گرادیان کاهشی پیدا خواهیم کرد.

**درک و مفهوم گرادیان:**

همان‌طور که می‌دانید برای پیدا کردن نقاط بهینه یک تابع از مشتق<sup>3</sup> استفاده می‌کنیم. از طرف دیگر متوجه شدیم که هدف از حل مسئله بهینه‌سازی معادله (۲)، پیدا کردن مقادیر بهینه پارامترهای  $w_0$  و  $w_1$  است به طوری که تابع هزینه MSE را حداقل کند. بنابراین می‌توان با مشتق‌گیری از تابع موجود در معادله (۲) مقادیر بهینه برای  $w_0$  و  $w_1$  را پیدا کنیم.

---

Grid Search<sup>1</sup>

Gradient<sup>2</sup>

Derivative<sup>3</sup>

توجه کنید، زمانی که ما بیش از یک پارامتر داریم (در اینجا  $w_0$  و  $w_1$ )، به جای مشتق، از کلمه گرادیان استفاده می‌کنیم. گرادیان در واقع برداری است که شامل مشتقات جزئی تابع نسبت به تمام پارامترهاست. در مسئله رگرسیون خطی با توجه به تابع زیان MSE در معادله (۲)، گرادیان نسبت به پارامترهای مدل (یعنی  $w_0$  و  $w_1$ ) به صورت زیر محاسبه می‌شود:

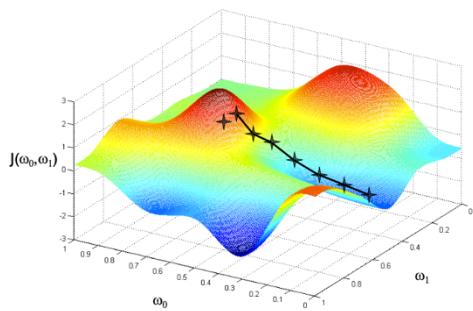
- مشتق نسبت به عرض از مبدأ (و یا به عبارتی گرادیان پارامتر  $w_0$ )

$$\frac{\partial f(w_0, w_1)}{\partial w_0} = \frac{2}{n} \sum_{i=1}^n ((w_0 + w_1 x_i) - y_i) \quad (3)$$

- مشتق نسبت به شیب خط (و یا به عبارتی گرادیان پارامتر  $w_1$ )

$$\frac{\partial f(w_0, w_1)}{\partial w_1} = \frac{2}{n} \sum_{i=1}^n x_i ((w_0 + w_1 x_i) - y_i) \quad (4)$$

### درک و مفهوم گرادیان کاهشی:



از نظر شهودی تصور کنید در مه غلیظی روی کوه هستید و هدف دارید تا به دره (کمترین مقدار خطا) برسید. در این مثال، گرادیان در واقع شیب زمین زیر پای شماست و شما برای رسیدن به دره، باید در هر قدم خلاف جهت شیب (خلاف جهت گرادیان) حرکت کنید. همین دید را برای به روزرسانی پارامترها در نظر بگیرید که با شروع از یک مقدار اولیه، پارامترها طوری به روزرسانی شوند تا خلاف جهت گرادیان حرکت کنند. این همان الگوریتم گرادیان کاهشی است.

در الگوریتم گرادیان کاهشی با شروع از یک مقدار اولیه، پارامترها در هر قدم با فرمول زیر به روزرسانی می‌گردند:

$$w_{new} = w_{old} - (\alpha \times \text{Gradient}) \quad (6)$$

در رابطه (۶)،  $\alpha$  (نرخ یادگیری<sup>۱</sup>) تعیین می‌کند که اندازه قدم‌های ما برای رسیدن به هدف چقدر بزرگ باشد. در واقع گرادیان کاهشی الگوریتمی است که به جای امتحان کردن تصادفی اعداد، با شروع از یک مقدار اولیه برای پارامترها و با استفاده از گرادیان‌ها به ما می‌گوید در کدام جهت و با چه سرعتی حرکت کنیم تا به نقطه کمینه برسیم.<sup>۲</sup>

<sup>1</sup> Learning rate

<sup>2</sup> اگر می‌خواهید تئوری این بخش را دقیق‌تر یاد بگیرید، می‌توانید از [ویدئو ۱](#) و [ویدئو ۲](#) نیز کمک بگیرید.

بنابراین می‌توان از الگوریتم گرادیان کاهشی جهت پیدا کردن مقادیر بردار پارامتر  $w$  در رگرسیون خطی به منظور کمینه کردن تابع هزینه MSE استفاده کرد. در زیر گام‌های مورد نیاز برای پیدا کردن مقادیر بهینه برای پارامترهای  $w_0$  و  $w_1$  در رگرسیون خطی با یک ویژگی با استفاده از الگوریتم گرادیان کاهشی آمده است:

**گام ۱:** مقداردهی اولیه پارامترها  $w_0^0$  و  $w_1^0$  را با  $t = 0$  می‌گذاریم.

**گام ۲:** مقدار پارامترها را با توجه به معادله (۶) به روزرسانی کن. یعنی:

$$w_0^{t+1} = w_0^t - \alpha \times \frac{\partial f(w_0, w_1)}{\partial w_0^t}$$

$$w_1^{t+1} = w_1^t - \alpha \times \frac{\partial f(w_0, w_1)}{\partial w_1^t}$$

**گام ۳:**  $t \leftarrow t + 1$

**گام ۴:** گام‌های ۲ و ۳ را تکرار کن تا زمانی که اختلاف بردار پارامترها در دو گام متوالی (یعنی  $w^t$  و  $w^{t+1}$ ) نزدیک به صفر ( $\epsilon$  (اپسیلون)) شود و یا تعداد تکرارها بیشتر از حد تعیین شده شود.

## بخش ۱.۴. پیاده‌سازی و تحلیل رفتار گرادیان کاهشی با یک پارامتر (۳۰ نمره)

تا اینجا متوجه شدیم که گرادیان کاهشی چگونه به صورت گام‌به‌گام پارامترها را به روزرسانی می‌کند تا به بهترین مقادیر خود برسند. اکنون نوبت شماست که این فرآیند را بررسی و تحلیل کنید. بدین منظور نیاز است که نیمه آماده‌ای که در فایل Jupyter Notebook در اختیار شما قرار داده شده است را کامل کنید.

در ابتدا فرض کنید عرض از مبدأ ثابت است ( $w_0 = 0$ ) و تنها هدف داریم بهترین مقادیر برای پارامتر  $w_1$  را پیدا کنیم. بدین منظور قدم‌های زیر را جلو رفته و به سوالات خواسته شده پاسخ دهید:

- فرمول محاسبه گرادیان پارامتر  $w_1$  (براساس معادله (۴)) را پیاده‌سازی کنید.
- به ازای تعدادی از تکرارها پارامتر  $w_1$  را به روزرسانی کرده و پس از هر تکرار مقدار MSE را محاسبه و ذخیره کنید.
- نمودار خطای خط پیش‌بینی شده در هر تکرار از الگوریتم را رسم کرده (نمودار MSE-slop) و تحلیل کنید در هر تکرار از به روزرسانی پارامتر  $w_1$ ، مقدار MSE چه تغییری می‌کند؟
- متغیر  $\epsilon$  (اپسیلون) در الگوریتم گرادیان کاهشی چه نقشی دارد؟ اگر آن را صفر قرار دهیم، چه خطری الگوریتم ما را تهدید می‌کند؟
- بهترین خط پیش‌بینی شده با استفاده از الگوریتم گرادیان کاهشی را به همراه داده‌ها رسم کنید.

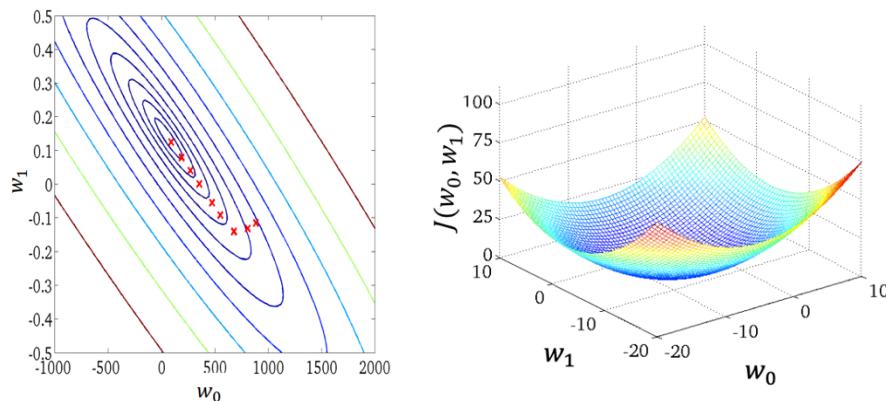
## بخش ۲،۴. بررسی تاثیر نرخ یادگیری در الگوریتم گرادیان کاهشی (۱۵ نمره)

در الگوریتم گرادیان کاهشی، یکی از مهم‌ترین فاکتورها تنظیم هایپرپارامتر<sup>۱</sup>  $\alpha$  (نرخ یادگیری) است که به صورت دستی و قبل از اجرای الگوریتم باید تنظیم گردد. به منظور بررسی تاثیر مقدار نرخ یادگیری، الگوریتم گرادیان کاهشی را به ازای مقادیر مختلف  $\{\alpha = 0.5, 0.1, 0.0001\}$  اجرا کنید و به سوالات زیر پاسخ دهید:

- نمودار MSE-slop را به ازای مقادیر مختلف نرخ یادگیری رسم کنید و نتایج را با یکدیگر مقایسه و تحلیل کنید.
- به ازای مقادیر مختلف از نرخ یادگیری، چند تکرار (Iteration) طول می‌کشد تا الگوریتم متوقف شود؟
- به ازای مقادیر مختلف از نرخ یادگیری، آیا مسیر حرکت به سمت پایین منحنی هموار است؟ سرعت همگرایی چه تغییری می‌کند؟
- چرا الگوریتم به جای رسیدن به دره، در بعضی از حالات از آن دور می‌شود؟
- عبارت "Divergence" یا واگرایی را با توجه به نمودارهایی که رسم کردید، توضیح دهید.

## بخش ۳،۴. پیاده‌سازی و تحلیل رفتار گرادیان کاهشی با دو پارامتر (۲۵ نمره)

بیشتر مدل‌هایی که در دنیای واقعی با آن‌ها سروکار داریم، بیش از یک پارامتر دارند. بنابراین در این بخش قصد داریم مدل قبلی خود را کمی واقعی‌تر کرده و علاوه بر شبی خط (پارامتر  $w_1$ ، عرض از مبدأ (پارامتر  $w_0$ ) را نیز به دست آوریم. در این حالت، تابع زیان MSE تابعی از دو متغیر خواهد بود. بنابراین به جای یک منحنی دو بعدی، تابع زیان شبیه به یک کاسه یا سطح سه‌بعدی (شکل (۳)) است و هدف پیدا کردن پایین‌ترین نقطه از این کاسه است.



شکل (۳). نمودار تابع زیان برای مسئله رگرسیون خطی با دو پارامتر  $w_0$  و  $w_1$ . بعد اول و دوم از نمودار فوق به ترتیب مقادیر مختلف پارامترهای  $w_0$  و  $w_1$  را نشان می‌دهد. بعد سوم مقدار MSE به ازای مقادیر مختلف پارامترهای  $w_0$  و  $w_1$  است. رنگ‌های گرم (قرمز و نارنجی) خطای بالا را نشان می‌دهند و هر چه به سمت رنگ‌های سرد (مثل سبز و آبی) می‌رویم خطای کمتر می‌شود. شکل سمت چپ نمودار را در حالت دو بعدی و شکل سمت راست نمودار را در حالت سه‌بعدی نمایش می‌دهد.

به منظور تحلیل رفتار گرادیان کاهشی با دو پارامتر، قدمهای زیر را جلو رفته و به سوالات خواسته شده پاسخ دهید:

- تابع محاسبه گرادیان برای پارامتر  $w_0$  را براساس معادله (۳) پیاده‌سازی کنید و الگوریتم گرادیان کاهشی را برای به دست آوردن دو پارامتر شبیه و عرض از مبدأ تغییر دهید.
- نمودار خطای خط پیش‌بینی شده در هر تکرار از الگوریتم گرادیان کاهشی را رسم کرده (نمودار MSE-slop) و تحلیل کنید.
- در هر تکرار از الگوریتم گرادیان کاهشی، مقادیر و گرادیان به دست آمده برای پارامترها و همچنین خطا به ازای آن‌ها را ذخیره کرده و نموداری مشابه با نمودار سمت چپ از شکل (۳) رسم کنید تا مسیر تغییرات گرادیان‌ها را نشان دهد. در این نمودار رفتار به روزرسانی پارامترها را بررسی کنید و تحلیل کنید چه مشکلی در این رفتار وجود دارد و چرا این اتفاق افتاده است.
- در نهایت، بهترین خط پیش‌بینی شده با استفاده از الگوریتم گرادیان کاهشی را به همراه داده‌ها رسم کنید.

#### بخش ۴،۴. برسی تاثیر استانداردسازی داده‌ها (۱۵ نمره)

برای حل مشکل بالا، باید تمام داده‌ها را به یک مقیاس مشترک بباوریم و آن‌ها را نرمال‌سازی<sup>۱</sup> کنیم. یکی از رایج‌ترین روش‌ها برای نرمال‌سازی داده‌ها این است که میانگین داده‌ها را صفر و واریانس را ۱ کنیم.

فرض کنید میانگین داده‌ها  $\mu$  و واریانس داده‌ها ( $\sigma$  انحراف معیار) باشد. به منظور نرمال‌سازی هر داده  $x^i$ ، از رابطه زیر استفاده می‌کنیم:

$$x_{new}^i = \frac{x_{old}^i - \mu}{\sigma} \quad (8)$$

بنابراین، ابتدا داده‌ها را با استفاده از رابطه (۸) نرمال‌سازی کنید، سپس به سوالات در بخش ۳،۴ مجدداً پاسخ دهید. همچنین توضیح دهید نرمال‌سازی چگونه به بهتر شدن خروجی کمک می‌کند.<sup>۲</sup>

---

<sup>1</sup> Normalization

<sup>2</sup> برای درک بهتر این مفهوم می‌توانید [ویدئو ۳](#) را مشاهده نمایید.

## نکات پایانی

- برای پیاده‌سازی تنها از Python استفاده کنید.
- گزارش شما باید کامل و جامع بوده و تمامی فعالیت‌های شما را پوشش دهد. نمایش نتایج و تحلیل‌ها نقش مهمی در ارزیابی فعالیت شما دارند.
- نتایج، گزارش و کدهای خود را در قالب یک فایل فشرده با فرمت AI\_Phase1\_[stdNumber].zip در کوئرا بارگذاری کنید (در صورت دسترسی نداشتن به کوئرا به دستیاران آموزشی خود میل کنید).
- در صورت کشف هرگونه تقلب (شباهت بالای ۶۰ درصد با کدهای موجود در اینترنت و دوستان خود و یا استفاده از ابزارهای هوش مصنوعی) در هر یک از قسمت‌های پروژه، نمره آن قسمت را از دست خواهد داد.
- امکان ارسال پروژه با تأخیر وجود ندارد.
- در صورت سوال از هر فاز پروژه با مدرس درس و یا دستیاران آموزشی در ارتباط باشید.

موفق باشید