استاد: دکتر فاطمه منصوری

نیمسال دوم ۱۴۰۳-۱۴۰۲

گرادیان کاهشی و پس انتشار

یادگیری عمیق

گردآورنده: ثنا سعیدمهر

پس انتشار و گرادیان کاهشی به منظور بهبود دقت پیش‌بینی شبکه‌های عصبی استفاده می‌شود. آن‌ها با کاهش خطای خروجی در شبکه های عصبی به بهبود دقت پیش‌بینی کمک می‌کنند.

اگرچه از پس انتشار و گرادیان کاهشی برای بهبود دقت پیش‌بینی شبکه‌های عصبی استفاده می‌شود، اما نقش‌های کاملاً متفاوتی در این فرآیند دارند. پس انتشار نقش محاسبه گرادیان را ایفا می‌کند، در حالی که گرادیان کاهشی نقش نزولی از طریق گرادیان را ایفا می‌کند.

گرادیان کاهشی

گرادیان چیست؟

گرادیان نشان دهنده جهت و شیب بیشینه تابع هدف است، که در ریاضیات، معادله‌ها و توابع مختلفی را می‌تواند مشخص کند. وقتی از گرادیان در زمینه یادگیری ماشین و شبکه‌های عصبی صحبت می‌شود، به معنی تغییر در وزن‌ها با توجه به تغییر در خطا است. در واقع، گرادیان می‌تواند به عنوان شیب تابع نیز در نظر گرفته شود که اگر شیب بیشتر باشد، مدل سریع‌تر یاد می‌گیرد، و اگر شیب صفر باشد، مدل دیگر یادگیری را ادامه نمی‌دهد.
الگوریتم گرادیان کاهشی یک روش بهینه‌سازی است که اغلب برای آموزش مدل‌های یادگیری ماشین و شبکه‌های عصبی استفاده می‌شود. این الگوریتم با کاهش خطاهای بین نتایج پیش‌بینی شده و واقعی، مدل‌های یادگیری ماشین را بهبود می‌بخشد. هدف اصلی از استفاده از الگوریتم گرادیان کاهشی، کمینه کردن تابع هدف (معمولاً تابع خطا یا تابع هزینه) است. این کار با به‌روزرسانی وزن‌های شبکه به گونه‌ای انجام می‌شود که تابع هدف به حداقل برسد.

در واقع، مراحل کلی الگوریتم گرادیان کاهشی به صورت زیر است:

۱. مقداردهی اولیه: با حدس اولیه برای پارامترهای تابع شروع می‌کند.

۲. محاسبه گرادیان: ابتدا گرادیان تابع هدف نسبت به وزن‌های شبکه و با توجه به پارامترهای نقطه فعلی محاسبه می‌شود. این گرادیان نشان دهنده جهت و نرخ تغییر تابع هدف در نقطه فعلی است.

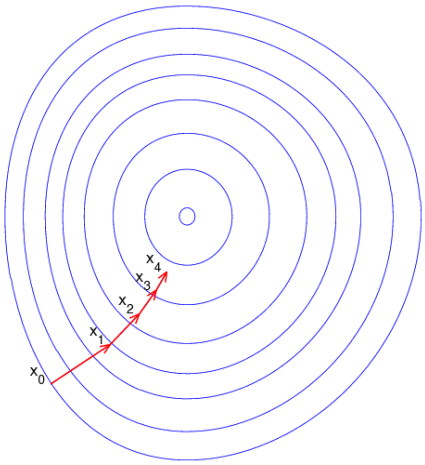
۳. به‌روزرسانی وزن‌ها: پارامترها را در جهت مخالف گرادیان تنظیم می‌کند تا به سمت حداقل حرکت کند. با استفاده از گرادیان محاسبه شده و نرخ یادگیری، وزن‌های شبکه به‌روزرسانی می‌شوند. این به‌روزرسانی باعث کاهش تابع هدف می‌شود.

۴. تکرار فرآیند: این دو مرحله (محاسبه گرادیان و به‌روزرسانی وزن‌ها) تا زمانی که به کمینه محلی یا جهانی تابع هدف برسد، تکرار می‌شود. این فرآیند به صورت مکرر انجام می‌شود تا وزن‌های شبکه بهینه شوند و تابع هدف به حداقل برسد، مانند رسیدن به تعداد مشخصی از تکرارها یا کاهش گرادیان به اندازه کافی.

مثال

مردی را تصور کنید که چشم‌بند دارد و می‌خواهد با کمترین پله‌های ممکن به بالای تپه‌ای صعود کند. او ممکن است با برداشتن گام‌های بزرگ در شیب‌دارترین مسیر، بالا رفتن از تپه را شروع کند. با نزدیک شدن به قله، قدم‌های او کوچک‌تر و کوچک‌تر می‌شوند تا از غلبه بر آن جلوگیری شود. این فرآیند را می توان با استفاده از گرادیان به صورت ریاضی توصیف کرد.

تصور کنید تصویر زیر تپه ما را از نمای بالا به پایین نشان می‌دهد و فلش‌های قرمز رنگ پله‌های کوهنورد ما هستند. شیب را در این زمینه به عنوان بردار در نظر بگیرید که جهت شیب دارترین قدمی را که مرد چشم بسته می‌تواند بردارد و همچنین طول آن قدم را در برمی‌گیرد.



توجه داشته باشید که گرادیان از $X0$ تا $X1$ بسیار طولانی تر از شیب از $X3$ به $X4$ است. دلیلش این است که شیب که طول بردار را تعیین می‌کند کمتر است. این کاملاً مثال تپه را نشان می‌دهد زیرا تپه هر چه بالاتر می‌رود شیب کمتری پیدا می‌کند. بنابراین شیب کاهش یافته همراه با کاهش اندازه پله برای تپه‌نورد است.

نحوه عملکرد گرادیان کاهشی

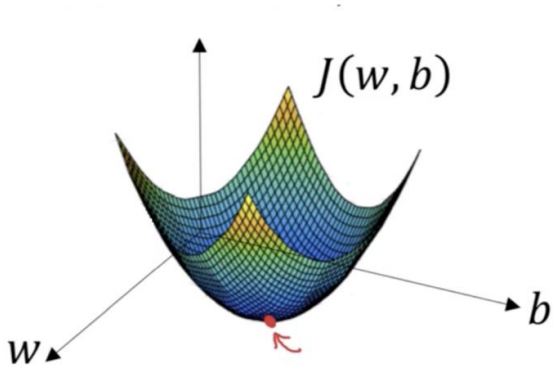
به جای بالا رفتن از یک تپه، گرادیان کاهشی را مانند پیاده‌روی به پایین دره در نظر بگیرید. این یک قیاس بهتر است زیرا یک الگوریتم کمینه‌سازی است که یک تابع معین را به حداقل می‌رساند.

معادله زیر آنچه را که الگوریتم گرادیان کاهشی انجام می‌دهد توصیف می‌کند: b موقعیت بعدی کوهنورد ما است، در حالی که a موقعیت فعلی او را نشان می‌دهد. علامت منفی به بخش کمینه‌سازی الگوریتم گرادیان کاهشی اشاره دارد. گاما یک عامل انتظار است و عبارت گرادیان $(\nabla f(a)$ شیب دارترین جهت نزولی است.

$$b = a - \gamma(\nabla f(a))$$

بنابراین این فرمول اساساً موقعیت بعدی را که باید برویم به ما می‌گوید که جهت بیشترین شیب نزول است. بیایید به مثال دیگری نگاه کنیم تا واقعاً مفهوم را به صورت کامل متوجه شویم.

تصور کنید که مسئله یادگیری ماشین دارید و می‌خواهید الگوریتم خود را با گرادیان کاهشی آموزش دهید تا تابع هزینه $J(w, b)$ خود را به حداقل برسانید و با تغییر پارامترهای آن $(b$ و $w)$ به حداقل محلی خود برسید. تصویر زیر محورهای افقی را نشان می‌دهد که پارامترهای $(b$ و $w)$ را نشان می‌دهد، در حالی که تابع هزینه $J(w، b)$ در محورهای عمودی نشان داده شده است. گرادیان کاهشی یک تابع محدب است.

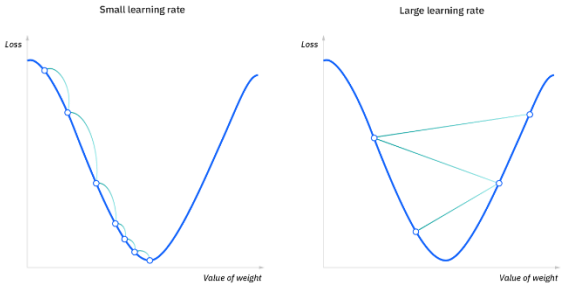


می‌دانیم که می‌خواهیم مقادیر w و b را پیدا کنیم که با حداقل تابع هزینه (که با فلش قرمز مشخص شده است) مطابقت دارد. برای شروع یافتن مقادیر مناسب، w و b را با تعدادی اعداد تصادفی مقداردهی اولیه می‌کنیم. سپس گرادیان کاهشی از آن نقطه شروع می‌شود (جایی در بالای تصویر)، و یکی پس از دیگری در شیب دارترین جهت نزولی (یعنی از بالا به پایین تصویر) طی می‌شود تا زمانی که به نقطه‌ای برسد که تابع هزینه تا حد امکان کوچک باشد.

نرخ یادگیری گرادیان

نرخ یادگیری در الگوریتم گرادیان کاهشی تعیین می‌کند که چقدر سریع یا آهسته می‌توان به سمت وزن‌های بهینه حرکت کرد. این فراپارامتر، مقدار مراحل انجام شده در طول به‌روزرسانی پارامترها را تعیین می‌کند. انتخاب نرخ یادگیری مناسب بسیار حائز اهمیت است، زیرا نرخ کم می‌تواند همگرایی را کاهش دهد و نرخ زیاد ممکن است باعث رد شدن از حداقل شود. الگوریتم گرادیان کاهشی هنگامی همگرا می‌شود که به نقطه‌ای نزدیک به صفر برسد که نشان‌دهنده حداقل محلی است. اما اگر تابع مناطق مسطح داشته باشد، ممکن است در نقطه‌ای گیر کند.

برای رسیدن به حداقل محلی با الگوریتم گرادیان کاهشی، باید نرخ یادگیری را به مقدار مناسب تنظیم کرد که نه خیلی کم و نه خیلی زیاد باشد. اگر مراحل بسیار بزرگ باشند، ممکن است به حداقل محلی نرسد زیرا بین تابع محدب گرادیان کاهشی به جلو و عقب می‌چرخد (تصویر سمت راست را ببینید). اگر نرخ یادگیری خیلی کم باشد، شیب نزول در نهایت به حداقل محلی می‌رسد اما ممکن است کمی طول بکشد (تصویر سمت چپ را ببینید).



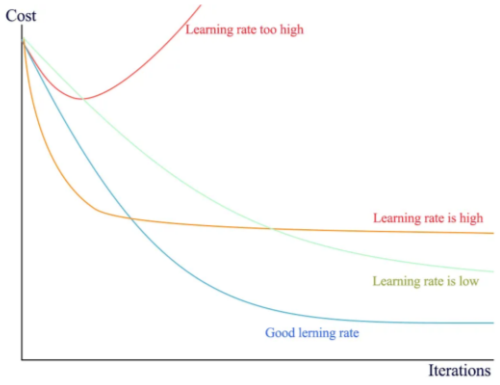
چالش‌های گرادیان کاهشی

چالش‌های گرادیان کاهشی شامل همگرایی آهسته، گیر کردن در حداقل‌های محلی و بیش از حد حداقل است. تکنیک‌هایی مانند حرکت[1]، نرخ‌های یادگیری تطبیقی[2] (مثلاً آدام[3]) و منظم‌سازی[4] می‌توانند به رفع این چالش‌ها کمک کنند.

یک راه خوب برای اطمینان از اجرای صحیح الگوریتم گرادیان کاهشی، ترسیم تابع هزینه در حین اجرای بهینه‌سازی است. این به شما کمک می‌کند ارزش تابع هزینه خود را بعد از هر بار تکرار گرادیان کاهشی ببینید و راهی برای تشخیص میزان مناسب بودن

[1] Momentum
[2] Adaptive Learning Rates
[3] Adam
[4] Regularization

نرخ یادگیری شما به راحتی فراهم می‌کند. شما می‌توانید مقادیر مختلف را برای آن امتحان کنید و همه آنها را با هم ترسیم کنید. تصویر زیر تفاوت بین نرخ یادگیری خوب و بد را نشان می‌دهد.



اگر الگوریتم گرادیان کاهشی به درستی کار کند، تابع هزینه باید پس از هر تکرار کاهش یابد. وقتی گرادیان کاهشی دیگر نمی‌تواند تابع هزینه را کاهش دهد و کم و بیش در همان سطح باقی می‌ماند، الگوریتم همگرا شده است. تعداد تکرارهایی که برای همگرایی گرادیان کاهشی نیاز دارد، گاهی اوقات می‌تواند بسیار متفاوت باشد.

درک گرادیان کاهشی برای آموزش موثر و کارآمد مدل‌های یادگیری ماشین ضروری است، زیرا اساس بسیاری از تکنیک‌های بهینه‌سازی مورد استفاده در این زمینه را تشکیل می‌دهد. تسلط بر مفاهیم و پیاده سازی های آن می‌تواند به بهبود عملکرد مدل‌ها در کاربردهای مختلف کمک کند.

انواع گرادیان کاهشی

سه نوع الگوریتم گرادیان کاهشی وجود دارد:

گرادیان کاهشی دسته ای[5]

گرادیان کاهشی دسته‌ای خطا را برای هر نقطه در یک مجموعه آموزشی محاسبه کرده و مدل را تنها پس از ارزیابی تمام نمونه‌های آموزشی به‌روز می‌کند. این فرآیند به عنوان یک دوره آموزشی شناخته می‌شود. این روش کارایی محاسباتی را فراهم می‌کند، اما ممکن است زمان زیادی برای پردازش مجموعه داده‌های بزرگ لازم باشد، زیرا همه داده‌ها باید در حافظه ذخیره شوند. گرادیان کاهشی دسته‌ای معمولاً یک گرادیان خطا و هم‌گرایی پایدار ایجاد می‌کند، اما گاهی اوقات این نقطه هم‌گرایی بهترین نقطه برای

[5] Batch Gradient Descent

مدل نیست و ممکن است حداقل محلی را در مقابل نقطه جهانی پیدا کند. (گرادیان را با استفاده از کل مجموعه داده آموزشی در هر تکرار محاسبه می‌کند.)

گرادیان کاهشی تصادفی[6]

گرادیان کاهشی تصادفی یک روش آموزشی است که برای هر نمونه در مجموعه داده اجرا می‌شود و پارامترهای هر نمونه را به ترتیب به‌روز می‌کند. این روش با داشتن تنها یک نمونه آموزشی، مدیریت حافظه را آسان‌تر می‌کند. از طرفی، این به‌روزرسانی‌های پی‌درپی ممکن است دقت و سرعت بیشتری داشته باشند، اما نسبت به روش گرادیان کاهشی دسته‌ای، باعث کاهش کارایی محاسباتی شود. به‌روزرسانی‌های مکرر ممکن است باعث ایجاد گرادیان‌های نویزی شوند، اما این می‌تواند برای فرار از حداقل محلی و یافتن جهانی مفید باشد. (محاسبه گرادیان و به‌روزرسانی پارامترها با استفاده از یک نمونه آموزشی در هر مرحله انجام می‌شود.)

Algorithm 1 Stochastic Gradient Descent

1: Hyperparameter: learning rate α, number of total iteration n_{iter}.
2: Initialize θ randomly.
3: **for** $i = 1$ to n_{iter} **do**
4: Sample j uniformly from $\{1, \ldots, n\}$, and update θ by

$$\theta := \theta - \alpha \nabla_\theta J^{(j)}(\theta)$$

گرادیان کاهشی دسته ای کوچک[7]

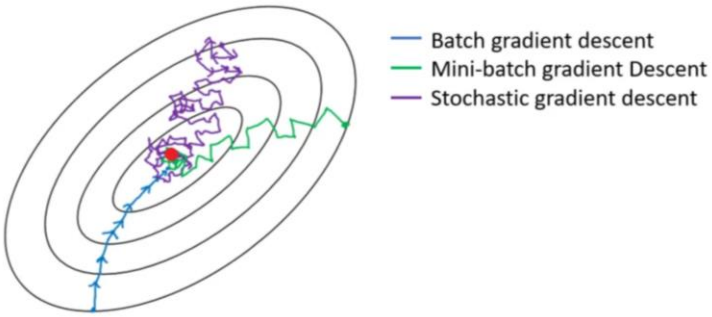
گرادیان کاهشی دسته‌ای کوچک مفاهیمی را از هر دو گرادیان کاهشی دسته‌ای و گرادیان کاهشی تصادفی ترکیب می‌کند. این روش مجموعه داده‌های آموزشی را به دسته‌های کوچک تقسیم کرده و پس از هر تقسیم، به‌روزرسانی انجام می‌دهد. این رویکرد تعادلی بین کارایی محاسباتی گرادیان کاهشی دسته‌ای و سرعت گرادیان کاهشی تصادفی ایجاد می‌کند. (پارامترها را با استفاده از یک زیرمجموعه کوچک و تصادفی از داده‌های آموزشی به روز می‌کند.)

Algorithm 2 Mini-batch Stochastic Gradient Descent

1: Hyperparameters: learning rate α, batch size B, # iterations n_{iter}.
2: Initialize θ randomly
3: **for** $i = 1$ to n_{iter} **do**
4: Sample B examples $j_1, \ldots, j_B$ (without replacement) uniformly from $\{1, \ldots, n\}$, and update θ by

$$\theta := \theta - \frac{\alpha}{B} \sum_{k=1}^{B} \nabla_\theta J^{(j_k)}(\theta) \tag{7.10}$$

[6] Stochastic gradient descent
[7] Mini-batch gradient descent



شکل ۱ : مقایسه انواع گرادیان کاهشی

پس انتشار

پس انتشار[8]، مخفف «انتشار معکوس خطاها[9]»، یک الگوریتم یادگیری نظارت شده[10] است که در آموزش شبکه های عصبی مصنوعی[11] استفاده می‌شود. این روشی برای محاسبه گرادیان تابع تلفات با توجه به وزن شبکه است. مبنای پس انتشار بر اصول ریاضیاتی از جمله قانون زنجیره‌ای حساب دیفرانسیل است که به شبکه‌ها اجازه می‌دهد تا یاد بگیرند چگونه وزن‌های خود را به‌روز کنند تا خطای خروجی را کاهش دهند. این فرآیند به شبکه اجازه می دهد تا از طریق تکرار داده‌های آموزشی، عملکرد خود را در طول زمان یاد بگیرد و بهبود بخشد.

در واقع می‌توان گفت این الگوریتم نقش مهمی در بهبود پیش‌بینی‌های انجام شده توسط شبکه‌های عصبی دارد. این به این دلیل است که پس انتشار می‌تواند خروجی شبکه عصبی را به طور مکرر بهبود بخشد.

در یک Feed Forward، ورودی از لایه ورودی به لایه خروجی به جلو حرکت می‌کند. پس انتشار به بهبود خروجی شبکه عصبی کمک می‌کند. این کار را با انتشار خطا به عقب از لایه خروجی به لایه ورودی انجام می‌دهد.

پس انتشار چگونه کار می‌کند؟

برای اینکه بفهمیم پس انتشار چگونه کار می‌کند، ابتدا بیایید بفهمیم که Feed Forward چگونه کار می‌کند.

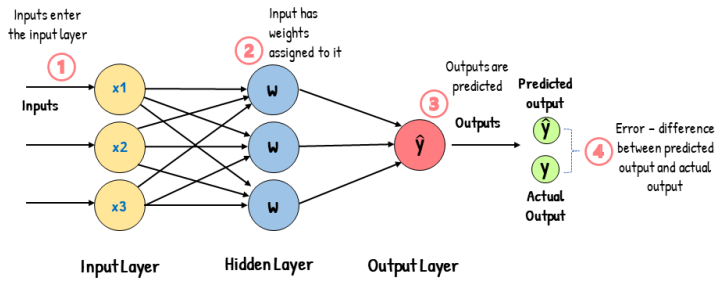
[8] Backpropagation
[9] Backward Propagation of Errors
[10] Supervised Learning Algorithm
[11] Artificial Neural Networks

Feed Forward

Feed Forward از یک لایه ورودی، یک یا چند لایه پنهان و یک لایه خروجی تشکیل شده است. لایه ورودی، ورودی شبکه عصبی را دریافت می‌کند و هر ورودی دارای وزنی است.

وزن‌های مرتبط با هر ورودی مقادیر عددی هستند. این وزن‌ها نشانگر اهمیت ورودی در پیش‌بینی خروجی نهایی هستند. به عنوان مثال، ورودی مرتبط با وزن زیاد، تأثیر بیشتری بر خروجی نسبت به ورودی مرتبط با وزن کم خواهد داشت.

هنگامی که یک شبکه عصبی برای اولین بار آموزش داده می‌شود، ابتدا با ورودی تغذیه می‌شود. از آن جایی که شبکه عصبی هنوز آموزش ندیده است، نمی‌داند از کدام وزن برای هر ورودی استفاده کند. بنابراین به هر ورودی به طور تصادفی یک وزن اختصاص داده می‌شود. از آنجایی که وزن‌ها به صورت تصادفی تخصیص داده می‌شوند، شبکه عصبی احتمالاً پیش‌بینی‌های اشتباهی انجام می‌دهد و خروجی نادرست را نشان می‌دهد.

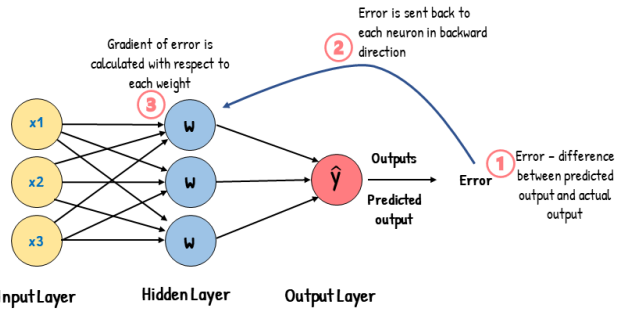


شکل ۲ : Feed Forward

هنگامی که شبکه عصبی، خروجی نادرستی می‌دهد، منجر به خطای خروجی می‌شود. این خطا تفاوت بین خروجی‌های واقعی و پیش‌بینی شده است. تابع هزینه این خطا را اندازه‌گیری می‌کند. تابع هزینه (J) نشان می‌دهد که مدل چقدر دقیق عمل می‌کند. بنابراین هدف به حداقل رساندن تابع هزینه یعنی کاهش خطای خروجی است. از آنجایی که وزن‌ها روی خطا تأثیر می‌گذارند، باید وزن‌ها را دوباره تنظیم کنیم تا تابع هزینه را به حداقل برساند.

این همان جایی است که پس انتشار وارد می‌شود...

پس انتشار به ما این امکان را می دهد که وزن را مجدداً تنظیم کنیم تا خطای خروجی را کاهش دهیم. خطا در حین پس انتشار از خروجی به لایه ورودی به عقب منتشر می شود. سپس از این خطا برای محاسبه گرادیان تابع هزینه با توجه به هر وزن استفاده می‌شود.



شکل ۳ : Backpropagation

اساساً، هدف پس انتشار محاسبه گرادیان منفی تابع هزینه است. این گرادیان منفی همان چیزی است که به تنظیم وزن کمک می‌کند. این ایده را به ما می‌دهد که چگونه باید وزن ها را تغییر دهیم تا بتوانیم تابع هزینه را کاهش دهیم.

پس انتشار از قانون زنجیره برای محاسبه گرادیان تابع هزینه استفاده می‌کند. قانون زنجیره شامل گرفتن مشتق است. این شامل محاسبه مشتق جزئی هر پارامتر است. این مشتقات با تفکیک یک وزن و در نظر گرفتن وزن (های) دیگر به عنوان یک ثابت محاسبه می‌شوند. در نتیجه انجام این کار، یک گرادیان خواهیم داشت. از آن جایی که ما گرادیان‌ها را محاسبه کرده‌ایم، می توانیم وزن‌ها را تنظیم کنیم.

عملکرد الگوریتم پس انتشار به صورت زیر است:

- در مرحله عبور به جلو[۱۲]، داده‌های ورودی لایه به لایه از شبکه عبور داده می‌شود و خروجی را تولید می‌کند.

- خروجی با استفاده از تابع تلفات برای اندازه‌گیری خطا با خروجی مورد نظر مقایسه می‌شود.

- در مرحله گذر به عقب[۱۳]، گرادیان تابع تلفات نسبت به هر وزن در شبکه با استفاده از قانون زنجیره محاسبه می‌شود.

- سپس وزن‌ها در جهت مخالف گرادیان با استفاده از الگوریتم‌های بهینه سازی مانند گرادیان کاهشی تنظیم می‌شوند تا خطا را به حداقل برسانند.

[12] Forward Pass
[13] Backward Pass

	Backpropagation	Gradient Descent
Definition	An algorithm for calculating the gradients of the cost function	Optimization algorithm used to find the weights that minimize the cost function
Requirements	Differentiation via the chain rule	•Gradient via Backpropagation •Learning rate
Process	Propagating the error backwards and calculating the gradient of the error function with respect to the weights	Descending down the cost function until the minimum point and find the corresponding weights

شکل ۴ : مقایسه گرادیان کاهشی و پس انتشار

منابع

[1] Ng, A. (2000). CS229 Lecture notes. CS229 Lecture notes, 1(1), 1-3.