دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)

دانشکده ریاضی و علوم کامپیوتر

گزارش پروژه نهایی درس جبرخطی عددی

موضوع پروژه

گرادیان کاهشی طبیعی

نگارش

رقیه مهدوی لاین

استاد

دکتر مهدی دهقان

بهار ۱۴۰۴

چکیده

در این تحقیق سه روش مهم بهینه‌سازی شامل گرادیان با تندترین شیب ، گرادیان مزدوج و گرادیان طبیعی مورد بررسی قرار گرفته‌اند. ابتدا روش گرادیان با تندترین شیب به عنوان ساده‌ترین الگوریتم کاهش تابع هدف معرفی شده و نحوه‌ی جهت‌گیری آن در راستای منفی گرادیان تابع هدف توضیح داده شده است. سپس روش گرادیان مزدوج، که با در نظر گرفتن حافظه‌دار بودن جهت‌ها و استفاده از اطلاعات پیشین گرادیان به سرعت همگرایی بالاتری دست می‌یابد، تحلیل شده است. در ادامه، روش گرادیان طبیعی بررسی شده که با تکیه بر ساختار هندسی فضای پارامتر و استفاده از متریکFisher ، جهت واقعی‌ترین کاهش را در فضاهای پیچیده مانند شبکه‌های عصبی ارائه می‌دهد. در این تحقیق مزایا، پایداری عددی، سرعت همگرایی و کاربردهای هر روش مقایسه شده و نقاط قوت و ضعف آنها تحلیل شده‌اند.

فهرست مطالب

مقدمه

مروری بر بهینه سازی محدب

مسئله بهینه سازی محدب به صورت زیر تعریف می‌شود:

$$\text{minimize} \quad f_0(x)$$
$$\text{subject to} \quad f_i(x) \ll b_i \qquad i = 1, \ldots, m$$

که در آن توابع $f_0, f_1, \ldots, f_m : R^n \to R$ همگی محدب هستند به این معنا که شرط زیر را ارضا میکنند.

$$f_i(\alpha x + \beta y) \ll \alpha f_i(x) + \beta f_i(y)$$

برای همه بردارهای $x, y \in R^n$ و ضرایب $\alpha, \beta \in R$ که جمع آنها برابر با یک است، یعنی $\alpha + \beta = 1$ ، و نیز $\alpha, \beta \geq 0$.
مسائل کمترین مربعات و برنامه ریزی خطی هر دو نمونه‌هایی خاص از این قالب کلی بهینه سازی محدب هستند.

مروری بر روش گرادیان کاهشی

در طول اجرای الگوریتم گرادیان کاهشی (Gradient Descent) ، دستگاه به‌صورت تکراری نقطه‌ی بعدی به‌روزرسانی‌شده را با استفاده از گرادیان تابع هدف در موقعیت فعلی محاسبه می‌کند. این به‌روزرسانی از طریق کم‌کردن حاصل‌ضرب نرخ یادگیری در گرادیان نسبت به پارامتر فعلی صورت می‌گیرد. رابطه‌ی این به‌روزرسانی به‌صورت زیر بیان می‌شود:

$$p_{n+1} = p_n - l\nabla f(p_n)$$

$$\nabla f = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}\right]^T$$

که در آن:

- l نرخ یادگیری
- p_n پارامتر مودر بهینه سازی در مرحله n-ام است
- و $\nabla f(p_n)$ گرادیان تابع هزینه مورد انتظار در نقطه‌ی p_n را نشان می‌دهد.

مشکل اصلی

تعریف بد حالتی

یک ماتریس زمانی **بدحالت** گفته می‌شود که تغییرات جزئی در داده های ورودی منجر به تغییرات بسیار بزرگ در خروجی شود.

تعریف عدد حالت ماتریس

برای ماتریس شرطی A به مقدار عددی $\|A\|.\|A^{-1}\|$ عدد شرطی یا عدد وضعیت یا عدد حالت (Condition number) ماتریس A گفته می‌شود و با نماد $\kappa(A)$ نمایش داده می‌شود. $\kappa(A) = \|A\|.\|A^{-1}\|$.

فرض کنید $\lambda_{(max)}$ و $\lambda_{(min)}$ به ترتیب بزرگترین و کوچترین مقدار ویژه ماتریس متقارن A باشند. آنگاه:

$$\kappa_2(A) = \frac{\lambda_{(max)}}{\lambda_{(min)}}$$

ماتریس هسین

ماتریس هسین یک ماتریس مربعی از مشتق‌های مرتبه دوم یک تابع اسکالر است. اگر تابع $f : R \to R^n$ باشد آنگاه ماتریس هسین H به صورت زیر تعریف می‌شود:

$$H = (\nabla f)' = \begin{pmatrix} \frac{\partial^2 f}{\partial {x_1}^2} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{pmatrix}$$

این ماتریس که در این‌جا با H نشان داده شده است، به ماتریس **هسین تابع f** معروف است و درایه‌های آن به صورت زیر تعریف می‌شوند:

$$H_{i,j} = \frac{\partial^2 f}{\partial x_j \partial x_i} = \frac{\partial^2 f}{\partial x_i \partial x_j} = H_{j,i}$$

این‌که می‌توان مشتق‌های جزئی را به هر ترتیبی محاسبه کرد، یک نکته از «تقارن مشتق‌های جزئی مرتبه دوم» یا «برابری مشتق‌های جزئی مختلط» شناخته می‌شود و به این معناست که ماتریس هسین یک ماتریس متقارن است، یعنی: $H = H^T$

بسیاری از کاربردهای مهم مشتق‌های مرتبه دوم، ماتریس‌های هسین، و تقریب‌های درجه دوم در زمینه‌ی بهینه‌سازی ظاهر می‌شوند؛ یعنی در مسائلی مانند کمینه‌سازی (یا بیشینه‌سازی) توابع $f(x)$.

در مسئله‌ی یافتن مینیمم (یا ماکزیمم) محلی یک تابع پیچیده $f(x)$، یک رویکرد متداول ریاضی آن است که مقدار $f(x + \delta x)$ را برای بردار جابه‌جایی کوچک δx با یک تابع ساده‌تر موسوم به "مدل" تقریب زده و سپس آن مدل را بهینه کنیم تا یک گام بهینه‌سازی پیشنهاد شود.

به طور مشخص استفاده از تقریب:

$$f(x+\delta x) \approx f(x) + f'(x)[\delta x]$$

که یک مدل خطی (affine) به‌شمار می‌رود، منجر به روش‌های گرادیان کاهشی و سایر الگوریتم‌های مشابه می‌شود.

با این حال تقریب دقیق‌تر $f(x+\delta x)$ می‌تواند به الگوریتم‌هایی با نرخ همگرایی سریع‌تر بینجامد. از این رو، یک ایده‌ی طبیعی آن است که از مشتق دوم f'' برای ساخت یک مدل درجه دوم استفاده کنیم، یعنی :

$$f'(x+\delta x) \approx f'(x) + f''(x)[\delta x]$$

که در فضای $\mathbb{R}^n$ به صورت $\delta(\nabla f) \approx H\delta x$ بیان می‌شود؛ که در آن H ماتریس هسین (Hessian) است.

از این‌رو، کمینه‌سازی مدل درجه دوم متناظر با گام نیوتن است:

$$\delta x \approx -H^{(-1)}\nabla f$$

که تلاشی است برای یافتن ریشه‌ی ∇f بر اساس تقریب مرتبه اول..

با این حال دقت و پایداری این تقریب به شرایط عددی ماتریس هسین H بستگی دارد. در شرایطی که ماتریس هسین بد حالت *(ill-conditioned)*باشد، عدد شرطی آن *(condition number)*بسیار بالا باشد، محاسبه‌ی

$$\delta x \approx -H^{(-1)}\nabla f$$

با خطای عددی قابل توجهی مواجه می‌شود. در این حالت، جهت و اندازه‌ی گام پیشنهادی δx می‌تواند به‌صورت ناپایدار و نادقیق تخمین زده شود.

به‌عبارت دیگر، هنگامی که H بدشرط است، بردار گرادیان ∇f ممکن است در جهاتی با خمیدگی زیاد (بردار ویژه های متناظر با مقادیر ویژه بزرگتر) مؤثرتر از جهاتی با خمیدگی کم باشد، و همین موضوع باعث می‌شود مشتق دوم (و در نتیجه مدل درجه دوم) رفتار نامناسبی در مقیاس‌های مختلف فضا داشته باشد. این موضوع در عمل به الگوریتم‌هایی منجر می‌شود که در مسیر بهینه‌سازی دچار زیگ زاگ و نوسانات شدید می‌شوند و نرخ همگرایی بسیار کاهش می‌یابد.

بنابراین، بدحالتی هسین نه‌تنها محاسبه‌ی گام نیوتنی را بی‌ثبات می‌کند، بلکه کیفیت تقریب مشتق دوم را نیز به‌شدت تحت‌تأثیر قرار می‌دهد، چرا که بردار جابه‌جایی δx به‌شدت نسبت به خطای عددی حساس خواهد بود. به همین دلیل، در این شرایط استفاده از روش‌هایی چون گرادیان مزدوج (Conjugate Gradient) یا گرادیان طبیعی *(Natural Gradient)* که این بدشرطی را اصلاح یا به‌نحوی پیش‌شرط‌گذاری(Preconditioning) می‌کنند، می‌تواند به بهبود پایداری و سرعت همگرایی بینجامد.

نمونه تأثیر ماتریس هسین بد حالت

تابع اول: $f(x,y)=\frac{1}{2}(x^2+y^2)$

گرادیان: $\nabla f(x,y)=\begin{bmatrix}x\\y\end{bmatrix}$

ماتریس هسین تابع: $H=\begin{bmatrix}1&0\\0&1\end{bmatrix}$

عدد حالت ماتریس: $\kappa_2(A)=\frac{\lambda_{(max)}}{\lambda_{(min)}}=\frac{1}{1}=1$ ماتریس بسیار خوش حالت هست.

نقطه شروع: $(x_0,y_0)=(2,2)$

نرخ یادگیری: $\alpha=0.1$

تکرار ۱

$$\nabla f=\begin{bmatrix}2\\2\end{bmatrix},\qquad (x_1,y_1)=(2,2)-0.1\times\begin{bmatrix}2\\2\end{bmatrix}=(1.8,1.8)$$

تکرار ۲

$$\nabla f=\begin{bmatrix}1.8\\1.8\end{bmatrix},\qquad (x_2,y_2)=(1.8,1.8)-0.1\times\begin{bmatrix}1.8\\1.8\end{bmatrix}=(1.62,1.62)$$

تکرار ۳

$$\nabla f=\begin{bmatrix}1.62\\1.62\end{bmatrix},\qquad (x_3,y_3)=(1.62,1.62)-0.1\times\begin{bmatrix}1.62\\1.62\end{bmatrix}=(1.458,1.458)$$

……

هر گام به‌طور یکنواخت و مستقیم به‌سوی مبدأ، که نقطه‌ی مینیمم است، حرکت می‌کند. همگرایی سریع و پایدار است.

تابع دوم: $f(x,y)=\frac{1}{2}(x^2+1000y^2)$

گرادیان: $\nabla f(x,y)=\begin{bmatrix}x\\1000y\end{bmatrix}$

ماتریس هسین تابع: $H=\begin{bmatrix}1&0\\0&1000\end{bmatrix}$

عدد حالت ماتریس: $\kappa_2(A)=\frac{\lambda_{(max)}}{\lambda_{(min)}}=\frac{1000}{1}=1000$ ماتریس بسیار بد حالت هست.

نقطه شروع: $(x_0,y_0)=(2,2)$

نرخ یادگیری: $\alpha=0.001$ به دلیل انحنای زیاد، نرخ یادگیری باید بسیار کوچک باشد.

تکرار ۱

$$\nabla f = \begin{bmatrix} 2 \\ 2000 \end{bmatrix}, \qquad (x_1, y_1) = (2, 2) - 0.001 \times \begin{bmatrix} 2 \\ 2000 \end{bmatrix} = (1.998, 0)$$

تکرار ۲

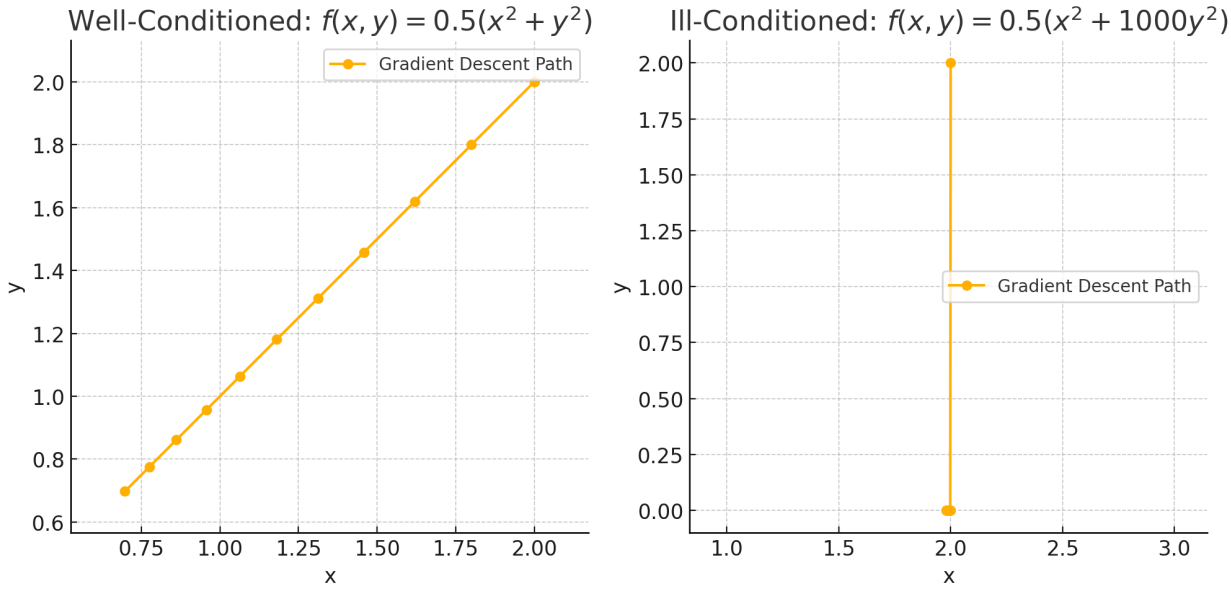
$$\nabla f = \begin{bmatrix} 1.998 \\ 0 \end{bmatrix}, \qquad (x_2, y_2) = (1.998, 0) - 0.001 \times \begin{bmatrix} 1.998 \\ 0 \end{bmatrix} = (1.996002, 0)$$

تکرار ۳

$$\nabla f = \begin{bmatrix} 1.996002 \\ 0 \end{bmatrix}, \qquad (x_3, y_3) = (1.996002, 0) - 0.001 \times \begin{bmatrix} 1.996002 \\ 0 \end{bmatrix} = (1.994006, 0)$$

.....

با اینکه از نقطه‌ی (۲، ۲) شروع کردیم، الگوریتم در همان گام اول مؤلفه‌ی y را فوراً به صفر رساند به‌دلیل گرادیان بزرگ $1000y$. سپس در جهت x پیشرفت بسیار کندی دارد. این رفتار ناشی از **عدم تعادل در انحنا** است. الگوریتم گرادیان کاهشی گام‌های زیادی را صرف حرکت زیگ‌زاگ یا نزدیک شدن آهسته به نقطه‌ی مینیمم می‌کند.



همانطور که مشاهده شد، در حضور ماتریس هسین بد حالت، عملکرد الگوریتم گرادیان کاهشی ساده به‌طور محسوسی افت می‌کند. در چنین شرایطی، مسیر حرکت الگوریتم از نقطه شروع تا نقطه بهینه، با نوسانهای شدید، زیگ‌زاگ‌های متعدد و همگرایی بسیار کند همراه است. این رفتار ناشی از اختلاف زیاد در انحنای تابع در جهات مختلف و نرخ تغییرات غیرهم‌مقیاس گرادیان است.

راهکار های پیشنهادی

توضیح فرم درجه دوم(Quadratic Form)

بیایید با چند تعریف و توضیح درباره‌ی نمادگذاری‌ها شروع کنیم. الگوریتم هایی که در ادامه به آن می‌پردازیم در واقع برای حل دستگاه های معادلات به صورت زیر به کار می‌روند:

$$Ax = b$$

در این معادله، x بردار مجهول است که باید آن را بیابیم، b بردار معلوم و A یک ماتریس مربعی، متقارن و معین مثبت (یا نیمه معین مثبت) است.چنین دستگاه‌هایی در بسیاری از مسائل مهم مهندسی و علمی پدیدار می‌شوند؛ از جمله در روش‌های اختلاف محدود و اجزای محدود برای حل معادلات دیفرانسیل جزئی، تحلیل سازه‌ها، تحلیل مدارها و حتی تمرین‌های ریاضی.

ماتریس A معین مثبت است اگر به ازای هر بردار ناصفر x داشته باشیم: $x^T Ax > 0$

فرم درجه دوم (Quadratic Form) یک تابع اسکالر درجه دوم از یک بردار است که به صورت زیر نوشته می‌شود:

$$f(x) = \frac{1}{2}x^T Ax - b^T x + c \qquad (۱)$$

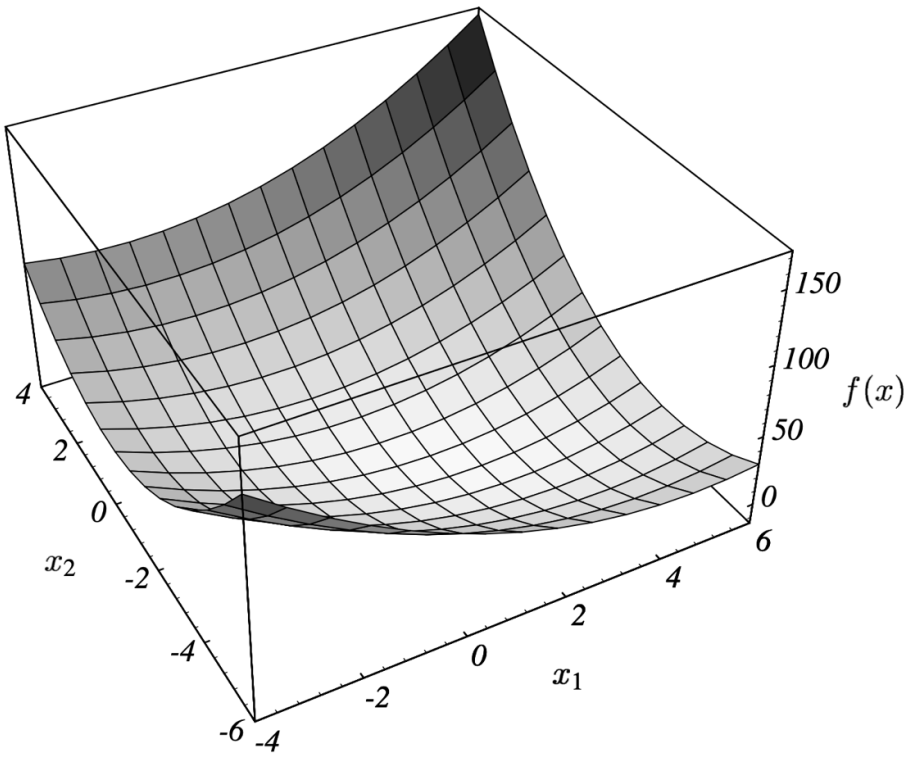
که در آن A یک ماتریس، و x و b بردارهایی هستند. اگر ماتریس A متقارن و معین مثبت (positive-definite) باشد، آنگاه این تابع در نقطه‌ای کمینه می‌شود که همان پاسخ دستگاه معادلات خطی زیر است:

$$Ax = b$$

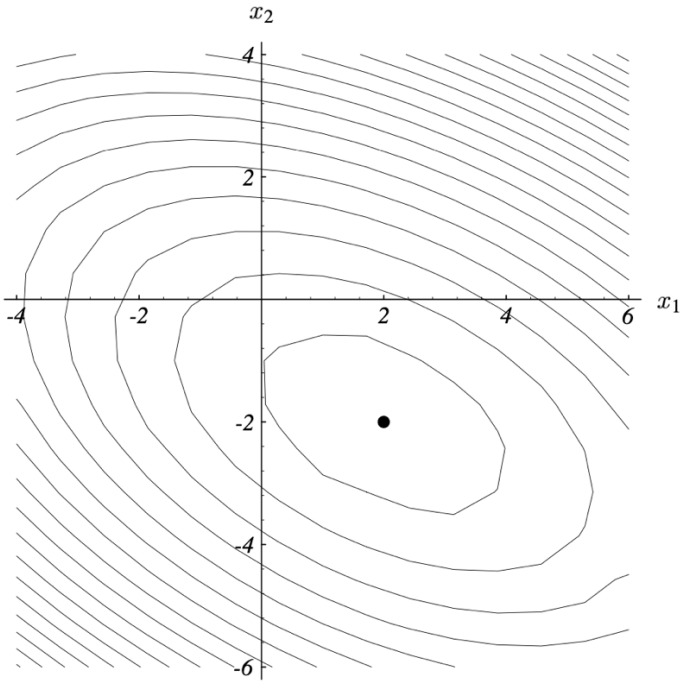
این ایده را با یک مسئله‌ی نمونه‌ی ساده نمایش خواهم داد.

$$A = \begin{bmatrix} 3 & 2 \\ 2 & 6 \end{bmatrix}, \qquad b = \begin{bmatrix} 2 \\ -8 \end{bmatrix}, \qquad c = 0$$

دستگاه $Ax = b$ در شکل ۱ نمایش داده شده است. به‌طور کلی، جواب x در نقطه‌ی تقاطع n ابرصفحه (hyperplane) قرار دارد، که هر یک بعد $n - 1$ دارند. در این مسئله، جواب برابر است با $x = [2, -2]^T$. فرم درجه دوم متناظر $f(x)$در شکل ۲ نمایش داده شده است. نمودار کانتور (خطوط تراز) $f(x)$ تابع نیز در شکل ۳ ارائه شده است.



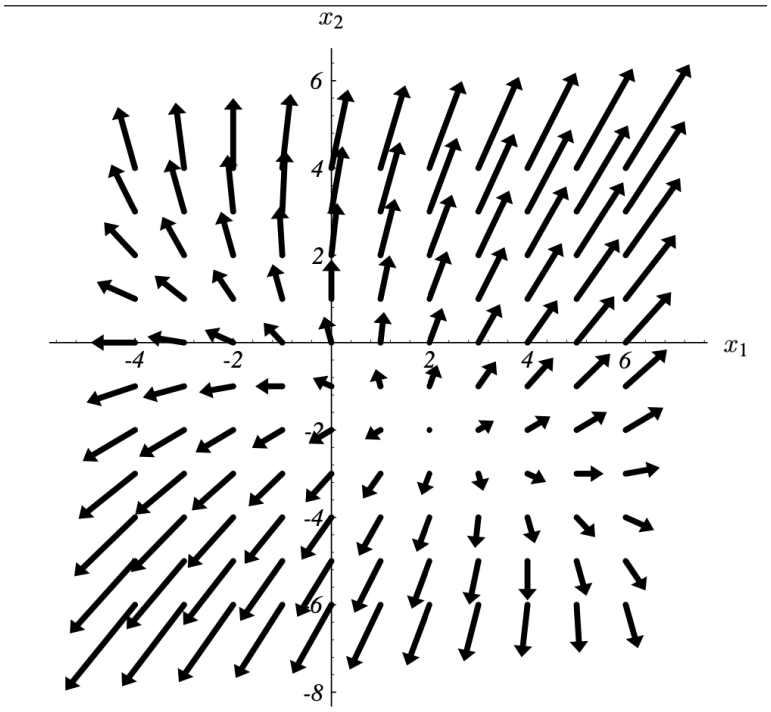
شکل (۱) نمودار شکل درجه دوم تابع $f(x)$ می‌باشد، نقطه مینیمم این سطح همان پاسخ $Ax = b$ است.



شکل(۲) کانتورهای فرم درجه دوم. هر منحنی بیضی‌شکل دارای مقدار ثابتی از $f(x)$ است.

نکته: نمودار کانتور، **منحنی‌هایی در صفحه دوبعدی (برای دو متغیر)** هستند که در آن مقدار تابع $f(x)$ برابر با یک مقدار ثابت c است : $f(x) = c$

برای تابع درجه دوم بالا، این معادله معمولاً منجر به بیضی‌هایی در صفحه می‌شود. مرکز این بیضی‌ها نقطه‌ای است که $f(x)$ در آن مینیمم می‌شود (یعنی جواب دستگاه)



شکل (۳) گرادیان $f'(x)$ برای تابع درجه دوم، جهت بیشترین افزایش مقدار تابع $f(x)$ را در هر نقطه x نشان می‌دهد و همواره بر خطوط کانتور عمود است.

نکته: نقطه مرکزی که همه فلش‌ها از آن دور می‌شوند (یا به آن نزدیک می‌شوند)، همان نقطه بهینه (کمینه) تابع و در نتیجه جواب بهینه دستگاه $Ax = b$ است. این نقطه جایی است که گرادیان صفر است، یعنی مشتق تابع در آن صفر شده و هیچ جهتی برای افزایش تابع وجود ندارد.

گرادیان فرم درجه دوم به شکل زیر است: (2)

$$f(x) = \begin{bmatrix} \frac{\partial}{\partial x_1} f(x) \\ \frac{\partial}{\partial x_2} f(x) \\ \vdots \\ \frac{\partial}{\partial x_n} f(x) \end{bmatrix}$$

با کمی محاسبات میتوان با اعمال کردن معادلات (۱) و (۲) به رابطه زیر رسید:

$$f'(x) = \frac{1}{2}A^T x + \frac{1}{2}Ax - b \qquad (۳)$$

و اگر ماتریس A متقارن باشد(یعنی $A^T = A$) آنگاه میتوان این رابطه را به صورت زیر نوشت:

$$f'(x) = Ax - b \qquad (۴)$$

و اگر این رابطه را مساوی با صفر قرار دهیم آنگاه جواب بهینه دستگاه را بدست می‌اوریم. بنابراین، جواب دستگاه یک نقطه بحرانی از تابع $f(x)$ است . اگر ماتریس A هم **متقارن** و هم **معین مثبت** باشد، آنگاه این جواب یک نقطه مینیمم برای تابع $f(x)$ خواهد بود.

روش گرادیان با بیشترین شیب (The method of Steepest Descent)

در روش گرادیان کاهشی تندترین شیب، از یک نقطه دلخواه $x_{(0)}$شروع می‌کنیم و به سمت پایین یک سطح سهمی‌وار (paraboloid)حرکت می‌کنیم. ما مجموعه‌ای از قدم‌ها به صورت$x_{(1)}, x_{(2)}, \ldots$ برمی‌داریم تا زمانی که اطمینان یابیم به اندازه کافی به جواب x نزدیک شده‌ایم.

در هر گام، جهتی را انتخاب می‌کنیم که تابع f با بیشترین سرعت کاهش می‌یابد، یعنی در جهت منفی گرادیان $f'(x_{(i)})$. طبق معادله‌ی (۴)، این جهت برابر است با:

$$-f'(x_{(i)}) = b - Ax_{(i)}$$

چند تعریف اولیه:

- خطا: $e_{(i)} = x_{(i)} - x$ برداری است که نشان می‌دهد چقدر از جواب واقعی فاصله داریم.
- باقیمانده: $r_{(i)} = b - Ax_{(i)}$ نشان می‌دهد که چقدر با مقدار صحیح بردار b فاصله داریم.(به راحتی می‌توان دید که $r_{(i)} = Ae_{(i)}$ و باید به باقیمانده به‌عنوان خطا که با A به فضای b نگاشته شده فکر کنید.)
- $r_{(i)} = -f'(x_{(i)})$ پس باید باقیمانده را برابر با جهت تندترین کاهش تابع f در نظر بگیرید.

فرض کنید از نقطه $x = [2, -2]^T$ اولین گام ما، در جهت تندترین کاهش، روی خط صافی قرار می‌گیرد که در شکل ۴ (a)نشان داده شده است. نقطه بعدی به صورت زیر به‌دست می‌آید:

$$x_{(1)} = x_{(0)} + \alpha r(0)$$

سؤال این است که چه اندازه گام (step size) باید برداریم؟

جستجوی خطی (line search) روشی است که مقدارα را طوری انتخاب می‌کند که مقدار f را در طول یک خط کمینه کند. شکل ۴ (b) این فرآیند را نشان می‌دهد: فقط اجازه داریم نقطه‌ای را روی تقاطع صفحه عمودی و سطح سهمی‌وار انتخاب کنیم.
شکل ۴ (c)سهمی‌ای (parabola) را نشان می‌دهد که از تقاطع آن دو سطح به دست آمده است. اکنون این پرسش مطرح می‌شود که مقدار α در پایه‌ی این سهمی چقدر است؟

می‌دانیم برای کمینه کردن f باید مشتق جهتی $\frac{d}{d\alpha}f(x_{(1)})$ برابر صفر باشد، با استفاده از مشتق زنجیره‌ای داریم:

$$\frac{d}{d\alpha}f(x_{(1)}) = f'(x_{(1)})^T \frac{d}{d\alpha}x_{(1)} = f'(x_{(1)})^T r_{(0)}$$

با برابر قرار دادن این عبارت با صفر، نتیجه می‌گیریم که α باید طوری انتخاب شود که $r_{(0)}$ بر $f'(x_{(0)})$ عمود باشد.

برای تعیین α، توجه کنید که $f'(x_{(1)}) = -r_{(1)}$، و داریم:

$$r_{(1)}^T r_{(0)} = 0$$

$$(b - Ax_{(1)})^T r_{(0)} = 0$$

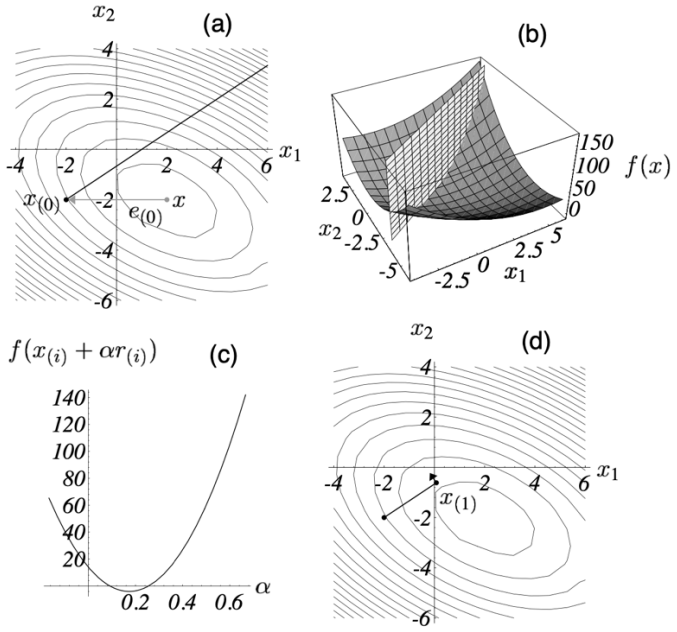
$$\left(b - A(x_{(0)} + \alpha r_{(0)})\right)^T r_{(0)} = 0$$

$$(b - Ax_{(0)})^T r_{(0)} - \alpha(Ar_{(0)})^T r_{(0)} = 0$$

$$(b - Ax_{(0)})^T r_{(0)} = \alpha(Ar_{(0)})^T r_{(0)}$$

$$r_{(0)}^T r_{(0)} = \alpha r_{(0)}^T (Ar_{(0)})$$

$$.\alpha = \frac{r_{(0)}^T r_{(0)}}{r_{(0)}^T A r_{(0)}}$$



شکل (۴) روش گرادیان نزولی:(Steepest Descent)
(a)با شروع از نقطه $[2, -2-]$، یک گام در جهت بیشترین کاهش تابع f برداشته می‌شود.
(b) نقطه‌ای روی تقاطع این دو سطح پیدا می‌شود که مقدار fرا کمینه می‌کند.
(c) این سهمی، حاصل تقاطع سطوح است؛ پایین‌ترین نقطه‌ی آن هدف ماست.
(d) گرادیان در پایین‌ترین نقطه عمود بر گرادیان گام قبلی است.

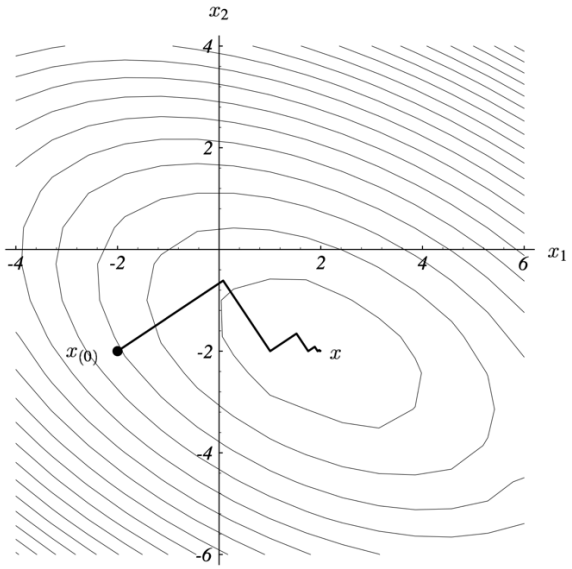
در مجموع، روش گرادیان نزولی (Steepest Descent) به‌صورت زیر است:

$$r_{(i)} = b - Ax_{(i)}$$

$$\alpha_{(i)} = \frac{r_{(i)}^T r_{(i)}}{r_{(i)}^T A r_{(i)}}$$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} r_{(i)}$$

مثال زیر را در نظر بگیرید.



شکل (۵) در این‌جا،الگوریتم از نقطه‌ی حدودی$[-2,2]$ شروع می‌شود و در نقطه‌ی حدودی$[2,2]$ همگرا می‌شود.

این مثال تا زمانی که همگرا شود، در شکل ۸ اجرا می‌شود. به مسیر زیگزاگی توجه کنید؛ این مسیر به این دلیل ظاهر می‌شود که هر گرادیان بر گرادیان قبلی عمود است.

پایداری این روش به عوامل متعددی وابسته است از جمله:

۱ .پایداری عددی و شرطی بودن ماتریس هسین

در مسائل مربعی (مانند کمینه‌سازی فرم درجه دوم)، عملکرد روش به شدت به ماتریس A بستگی دارد. اگر ماتریس بد حالت باشد، یعنی نسبت بزرگ‌ترین به کوچک‌ترین مقدار ویژه آن زیاد باشد، مسیر حرکت الگوریتم به صورت زیگزاگ و ناپایدار درمی‌آید و نرخ همگرایی کند می‌شود. این رفتار در نمودارهای کانتور به صورت حرکت‌های عمودشونده پیاپی روی ایزوکانتورهای بیضوی مشاهده می‌شود.

۲. حساسیت به انتخاب گام

روش نزول تندترین نیاز به یک جستجوی خطی دقیق (line search) دارد تا اندازه بهینه گام در هر تکرار تعیین شود. اگر این گام به‌درستی انتخاب نشود (مثلاً خیلی کوچک یا خیلی بزرگ باشد)، روش ممکن است همگرا نشود یا نوسان کند. در نتیجه، پایداری عددی این روش نسبت به روش‌هایی مانند گرادیان مزدوج یا روش نیوتن ضعیف‌تر است.

۳ .همگرایی آهسته برای مسائل بدشرط

برای سیستم‌هایی با ایزوکانتورهای کشیده (یعنی مقادیر ویژه‌ی بسیار متفاوت)، جهت‌های گرادیان در هر مرحله تقریباً عمود بر یکدیگر هستند. این موضوع باعث می‌شود الگوریتم با هر بار حرکت، بخش کمی از خطا را کاهش دهد و نیاز به تکرارهای زیاد داشته باشد. بنابراین، پایداری و سرعت همگرایی در این حالت کاهش می‌یابد

روش گرادیان مزدوج

روش **گرادیان کاهشی با بیشترین شیب (Steepest Descent)** اغلب در گام‌های متوالی خود، در همان جهتی حرکت می‌کند که در گام‌های قبلی نیز طی شده است (نگاه کنید به شکل ۸). اما آیا بهتر نبود اگر هر بار که گامی برمی‌داریم، آن گام از همان ابتدا در جهت بهینه باشد؟

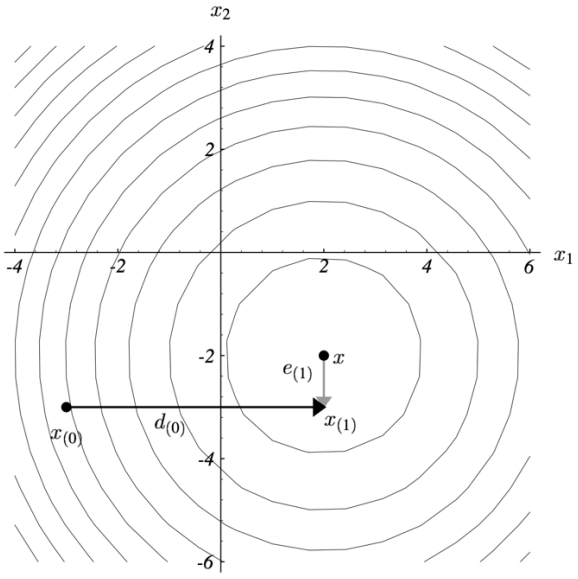
در این‌جا ایده‌ای مطرح می‌شود: فرض کنیم مجموعه‌ای از بردارهای جستجوی **متعامد** $d_{(0)}, d_{(1)}, \dots, d_{(n-1)}$ را در اختیار داریم. در هر یک از این جهت‌های جستجو فقط یک گام برمی‌داریم، و این گام با دقت طوری انتخاب می‌شود که نتیجه‌ی آن دقیقاً در راستای نقطه‌ی هدف x قرار گیرد. پس از n گام، به پاسخ نهایی خواهیم رسید.

شکل ۶ این ایده را به تصویر می‌کشد؛ در آن محورهای مختصات به‌عنوان جهت‌های جستجو در نظر گرفته شده‌اند. گام اول (در راستای محور افقی) ما را به مقدار صحیح مولفه‌ی اول x_1 می‌رساند؛ گام دوم (در راستای محور عمودی) نیز دقیقاً به مقصد می‌رسد. توجه کنید که خطای حاصل از گام دوم، یعنی $e_{(1)}$، بر جهت گام اول یعنی $d_{(0)}$عمود است.

در حالت کلی، برای هر گام، نقطه‌ای را به گونه‌ای انتخاب می‌کنیم که:

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)}$$

برای تعیین مقدار بهینه‌ی $\alpha_{(i)}$، از این نکته استفاده می‌کنیم که خطای حاصل از گام بعدی یعنی $e_{(i+1)}$، باید بر جهت گام فعلی $d_{(i)}$عمود باشد؛ در نتیجه، هیچ‌گاه نیازی نخواهد بود در همان جهت دوباره حرکت کنیم.



شکل (۶) روش جهت‌های متعامد متأسفانه، این روش تنها زمانی کارآمد است که پاسخ مسئله را از قبل بدانید.

CGصرفاً همانند روش جهت‌های مزدوج (Conjugate Directions) است، با این تفاوت که جهت‌های جستجو از باقی‌مانده‌ها (residuals)ساخته می‌شوند (یعنی با قرار دادن $u_{(i)} = r_{(i)}$).

این انتخاب از چند جهت منطقی است. اول آنکه باقی‌مانده‌ها در روش گرادیان کاهشی تند (Steepest Descent) عملکرد خوبی داشتند، پس چرا در روش جهت‌های مزدوج نیز از آنها استفاده نکنیم؟ دوم آنکه باقی‌مانده دارای ویژگی مطلوبی است: با جهت‌های جستجوی قبلی متعامد است بنابراین، استفاده از آن تضمین می‌کند که در هر مرحله، جهت جستجوی جدیدی تولید می‌شود که مستقل خطی از جهت‌های قبلی است — مگر آنکه باقی‌مانده صفر باشد، که در این صورت مسئله قبلاً حل شده است.

حال به بررسی پیامدهای این انتخاب بپردازیم. از آنجایی که بردارهای جستجو از باقی‌مانده‌ها ساخته می‌شوند، زیرفضای تولید شده توسط (مجموعه $\{r_{(0)}, r_{(1)}, \dots, r_{(i-1)}\}$ span برابر است با $D_{(i)}$ یعنی زیرفضای جستجو در مرحله‌ی i ام). همچنین، از آنجایی که هر باقی‌مانده بر جهت‌های جستجوی قبلی متعامد است، بر باقی‌مانده‌های قبلی نیز متعامد خواهد بود. بنابراین معادله به صورت زیر ساده می‌شود:

$$r_{(i)}^T r_{(j)} = 0 \qquad i \neq j$$

نکته‌ی جالب این‌جاست که معادله‌ی زیر:

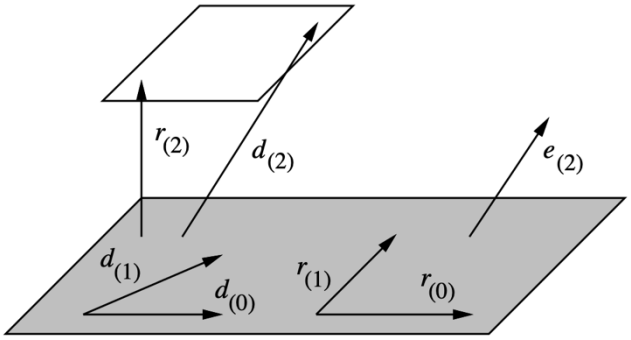
$$r_{(i+1)} = -Ae_{(i+1)}$$

$$= -A\left(e_{(i+1)} + \alpha_{(i)} d_{(i)}\right)$$

$$= r_{(i)} - \alpha_{(i)} A d_{(i)}$$

نشان می‌دهد باقی‌مانده‌ی جدید $r_{(i)}$صرفاً ترکیبی خطی از باقی‌مانده‌ی قبلی و بردار $Ad_{(i)}$است. با توجه به اینکه $d_{(i)}$ این حقیقت نشان می‌دهد که هر زیرفضای جدید $D_{(i+1)}$ از اجتماع زیرفضای قبلی $D_{(i)}$ و زیرفضای $AD_{(i)}$ تشکیل می‌شود. بنابراین:

$$D_{(i)} = span\{d_{(0)}, Ad_{(0)}, A^2 d_{(0)} \dots, A^{i-1} d_{(0)}\}$$
$$= span\{r_{(0)}, Ar_{(0)}, A^2 r_{(0)} \dots, A^{i-1} r_{(0)}\}$$



شکل (۷) در روش گرادیان مزدوج، هر باقی‌مانده‌ی جدید با تمام باقی‌مانده‌ها و جهت‌های جستجوی قبلی عمود است؛ و هر جهت جستجوی جدید نیز از باقی‌مانده ساخته می‌شود، به‌گونه‌ای که با تمام باقی‌مانده‌ها و جهت‌های جستجوی قبلی، نسبت به ماتریس A عمود باشد. نقاط پایانی بردارهای $r_{(2)}$ و $d_{(2)}$ روی صفحه‌ای قرار می‌گیرند که با زیرفضای $D_{(2)}$(که در شکل به‌صورت ناحیه‌ی سایه‌خورده نشان داده شده) موازی است. در روش CG ، بردار $d_{(2)}$ ترکیبی خطی از $r_{(2)}$ و $d_{(2)}$ است.

این زیرفضا، به نام زیرفضای کریلوف (Krylov Subspace) شناخته می‌شود. زیرفضایی که با اعمال مکرر یک ماتریس بر یک بردار تولید می‌شود. این زیرفضا ویژگی جذابی دارد: از آنجا که $AD_{(i)}$ ، و با توجه به اینکه باقی‌مانده‌ی بعدی $r_{(i)}$ بر $D_{(i)}$ عمود است، نتیجه می‌گیریم که $r_{(i+1)}$ نسبت به زیرفضای $D_{(i)}$ نیز A-متعامد خواهد بود.

در نتیجه، فرآیند متعامدسازی گرَم-اشمیت (Gram-Schmidt Conjugation) در اینجا ساده می‌شود، چرا که $r_{(i+1)}$ از پیش نسبت به تمام جهت‌های جستجوی قبلی (به جز $d_{(i)}$ نسبت به A متعامد است.

ثابت های گرام اشمیت به صورت زیر است بیایید آن را ساده تر کنیم.

$$\beta_{ij} = -\frac{r_{(i)}^T A d_{(j)}}{d_{(j)}^T A d_{(j)}};$$

$$r_{(i)}^T r_{(j+1)} = r_{(i)}^T r_{(j)} - \alpha_{(j)} r_{(i)}^T A d_{(j)}$$

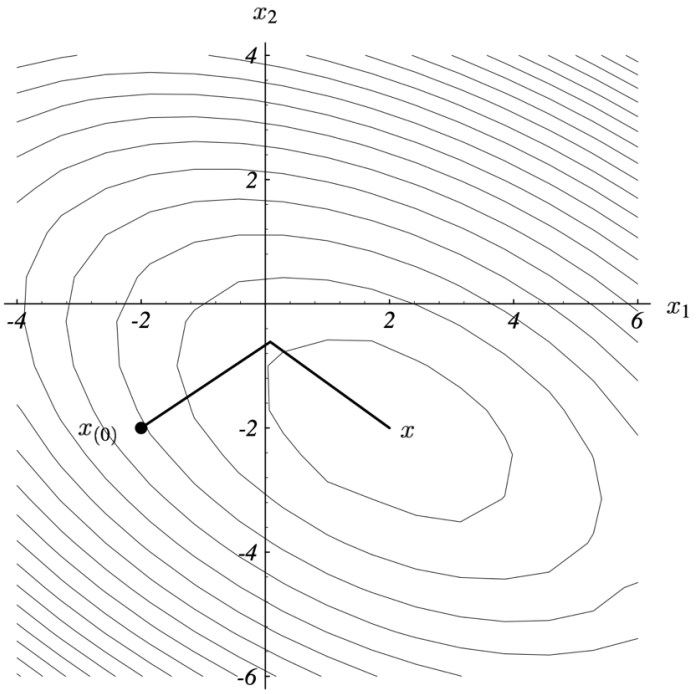
$$\Rightarrow \alpha_{(j)} r_{(i)}^T A d_{(j)} = r_{(i)}^T r_{(j)} - r_{(i)}^T r_{(j+1)}$$

$$r_{(i)}^T A d_{(i)} = \begin{cases} \frac{1}{\alpha_{(i)}} r_{(i)}^T r_{(i)}, & i = j \\ -\frac{1}{\alpha_{(i-1)}} r_{(i)}^T r_{(i)}, & i = j+1 \\ o & otherwise \end{cases}$$

دیگر نیازی به ذخیره‌سازی بردارهای جستجوی قبلی برای تضمین-A متعامد بودن بردارهای جستجوی جدید وجود ندارد. این پیشرفت عمده هماند چیزی است که الگوریتم(CG گرادیان مزدوج) را تا این حد مهم کرده است، زیرا پیچیدگی فضایی و زمانی هر تکرار از $O(n^2)$ به $O(m)$ کاهش یافته است، که در آن mتعداد مؤلفه‌های ناصفر ماتریس Aاست. از این پس، از نماد اختصاری $\beta_{(i)} =$ $\beta_{i,i-1}$ استفاده خواهم کرد. بیایید بیشتر ساده‌سازی کنیم:

$$\beta_i = \frac{r_{(i)}^T r_{(i)}}{d_{(i-1)}^T r_{(i-1)}}$$

$$= \frac{r_{(i)}^T r_{(i)}}{r_{(i-1)}^T r_{(i-1)}}$$



شکل(۸) روش گرادیان مزدوج

پس به طور کل داریم:

$$d_{(0)} = r_{(0)} = b - Ax_{(0)},$$

$$\alpha_{(i)} = \frac{r_{(i)}^T r_{(i)}}{d_{(i)}^T A d_{(i)}} \quad \text{(by Equations 32 and 42)},$$

$$x_{(i+1)} = x_{(i)} + \alpha_{(i)} d_{(i)},$$

$$r_{(i+1)} = r_{(i)} - \alpha_{(i)} A d_{(i)},$$

$$\beta_{(i+1)} = \frac{r_{(i+1)}^T r_{(i+1)}}{r_{(i)}^T r_{(i)}},$$

$$d_{(i+1)} = r_{(i+1)} + \beta_{(i+1)} d_{(i)}.$$

آنالیز پایداری گرادیان مزدوج

الگوریتم گرادیان مزدوج *(CG)* به‌صورت نظری پس از n تکرار کامل می‌شود، اما در عمل، خطاهای عددی مانند گرد کردن و حذف موجب کاهش دقت پسماند و از دست رفتن خاصیت-A متعامد بودن بردارهای جستجو می‌شوند. این مشکلات، به‌ویژه از دست رفتن مزدوج بودن در دهه‌ی ۱۹۶۰ باعث شد جامعه‌ی ریاضی *CG* را کنار بگذارد، اما در دهه‌ی ۱۹۷۰ با اثبات کارایی آن در مسائل بزرگ، دوباره مورد توجه قرار گرفت. امروزه تحلیل همگرایی اهمیت زیادی دارد، زیرا *CG* برای مسائلی استفاده می‌شود که اجرای کامل n تکرار در آنها ممکن نیست. بنابراین، تحلیل همگرایی بیشتر به‌عنوان اثباتی بر مفید بودن *CG* در مسائل بسیار بزرگ تلقی می‌شود تا ابزاری برای مقابله با خطای عددی. همچنین، چون اولین تکرار *CG* مانند روش $steepest\ descent$ است، شرایط همگرایی آن مشابه بررسی می‌شود.

مقایسه پیچیدگی

نتیجه می‌گیرم که روش Steepest Descent دارای پیچیدگی زمانی $O(mk)$ است، در حالی‌که روش گرادیان مزدوج (CG) دارای پیچیدگی زمانی $O(m\sqrt{k})$ می‌باشد. هر دو الگوریتم دارای پیچیدگی فضایی $O(m)$ هستند.

تقریبات تفاضل محدود و المان محدود برای مسائل مقدار مرزی بیضوی مرتبه دوم که روی حوزه‌هایی با ابعاد d تعریف شده‌اند، معمولاً شرط وضعیتی κ از مرتبه $O(n^{2/d})$ دارند. بنابراین، روش نزول شیب‌دار برای مسائل دوبعدی دارای پیچیدگی زمانی $O(n^2)$ است، در حالی‌که CG دارای پیچیدگی زمانی $O(n^{3/2})$ می‌باشد. همچنین، برای مسائل سه‌بعدی، پیچیدگی زمانی نزول شیب‌دار $O(n^{5/3})$ و برای CG برابر با $O(n^{4/3})$ است.

روش گرادیان طبیعی

گرادیان طبیعی (Natural Gradient) نسخه‌ای اصلاح‌شده از گرادیان معمولی است که با در نظر گرفتن ساختار هندسی فضای پارامترها، جهت واقعیِ بیشترین کاهش تابع هزینه را تعیین می‌کند. برخلاف گرادیان کلاسیک که در فضاهای پیچیده ممکن است باعث نوسان یا همگرایی کند شود، گرادیان طبیعی با استفاده از هندسه‌ی اطلاعات، مسیر یادگیری را بهینه‌تر و پایدارتر می‌سازد. در بسیاری از کاربردهای یادگیری ماشین مانند پرسپترونها، جداسازی کور منابع، و سیستم‌های دینامیکی، ثابت شده است که این روش دارای کارایی فیشر است؛ یعنی در بلندمدت به‌اندازه‌ی بهترین روش‌های تخمین دسته‌ای عملکرد دارد. همچنین، استفاده از گرادیان طبیعی می‌تواند پدیده‌ی توقف ناگهانی یادگیری (پلاتو) در الگوریتم‌هایی مانند backpropagation را کاهش داده یا حذف کند.

اجازه دهید $S = w \in \mathbb{R}^n$ یک فضای پارامتر باشد که تابعی به نام $L(w)$روی آن تعریف شده است.

هنگامی که S یک فضای اقلیدسی با دستگاه مختصات متعامد و یک‌نواخت باشد، طول مربع یک بردار افزایشی کوچک dw که نقاط w و $w + dw$ را به هم متصل می‌کند، به صورت زیر بیان می‌شود:

$$|dw|^2 = \sum_{i=1}^{n} (dw_i)^2,$$

که در آن dw_i مؤلفه‌های بردار dw هستند. با این حال زمانی که دستگاه مختصات غیرمتعامد باشد، طول مربع این بردار با استفاده از یک فرم درجه دوم (quadratic form) داده می‌شود.

$$|dw|^2 = \sum_{i,j} g_{ij}(w) dw_i dw_j.$$

وقتی S یک منیفلد خمیده باشد، دیگر دستگاه مختصات خطی متعامد وجود ندارد و طول بردار dw همیشه به صورت معادله بالا نوشته می‌شود. چنین فضایی یک فضای ریمانی (Riemannian space) نامیده می‌شود. در فضای پارامترهای شبکه‌های عصبی دارای ویژگی ریمانی است. ماتریس $n \times n$ به صورت $G = (g_{ij})$ که به طور کلی به w وابسته است، تانسور متریک ریمانی (Riemannian metric tensor)نامیده می‌شود. این ماتریس در شرایط خاص به حالت ساده‌تری کاهش می‌یابد.

$$g_{ij}(w) = \delta_{ij} = \begin{cases} 1, & i = j, \\ 0, & i \neq j \end{cases}$$

در حالت متعامد اقلیدسی، ماتریس Gبرابر با ماتریس یکه I است.

جهت بیشترین کاهش (شیب تندترین کاهش) تابع $L(w)$ در نقطه w توسط بردار dw تعریف می‌شود که مقدار $L(w + dw)$ را کمینه می‌کند، به شرطی که طول $|dw|$ ثابت باشد؛ یعنی تحت این قید که: $|dw|^2 = \varepsilon^2$

قضیه ۱.جهت بیشترین کاهش تابع $L(w)$ در یک فضای ریمانی توسط رابطه زیر داده می‌شود:

$$-\tilde{\nabla} L(w) = -G^{-1}(w) \nabla L(w)$$

که در آن $G^{-1} = \left(g^{ij}\right)$ (معکوس متریک $G = \left(g_{ij}\right)$ است و ∇L گرادیان معمولی تابع L می‌باشد.

$$\nabla L(\boldsymbol{w}) = \left(\frac{\partial}{\partial w_1} L(\boldsymbol{w}), \ldots, \frac{\partial}{\partial w_n} L(\boldsymbol{w})\right)^T,$$

Proof. We put

$$d\boldsymbol{w} = \varepsilon \boldsymbol{a},$$

and search for the $\boldsymbol{a}$ that minimizes

$$L(\boldsymbol{w} + d\boldsymbol{w}) = L(\boldsymbol{w}) + \varepsilon \nabla L(\boldsymbol{w})^T \boldsymbol{a}$$

under the constraint

$$|\boldsymbol{a}|^2 = \sum g_{ij} a_i a_j = 1.$$

By the Lagrangean method, we have

$$\frac{\partial}{\partial a_i}\{\nabla L(\boldsymbol{w})^T \boldsymbol{a} - \lambda \boldsymbol{a}^T G \boldsymbol{a}\} = 0.$$

This gives

$$\nabla L(\boldsymbol{w}) = 2\lambda G \boldsymbol{a}$$

or

$$\boldsymbol{a} = \frac{1}{2\lambda} G^{-1} \nabla L(\boldsymbol{w}),$$

where λ is determined from the constraint.
We call

$$\tilde{\nabla} L(\boldsymbol{w}) = G^{-1} \nabla L(\boldsymbol{w})$$

گرادیان طبیعی تابع L در فضای ریمانی، جهت $-\nabla\tilde{} L$ است. بنابراین، $-\nabla\tilde{} L$—جهت بیشترین کاهش تابع L را نشان می‌دهد. (اگر از نمادگذاری تانسوری استفاده کنیم، این دقیقاً شکل متضاد گرادیان $-\nabla L$ است.) زمانی که فضا اقلیدسی و دستگاه مختصات، متعامد و نرمال باشد، در این حالت ماتریس متریک G برابر با ماتریس همانی است، بنابراین گرادیان طبیعی با گرادیان معمولی برابر خواهد بود.

$$\nabla\tilde{} L = \nabla L$$

این بیانگر الگوریتم گرادیان نزولی طبیعی با فرم زیر است:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta_t \tilde{\nabla} L(\boldsymbol{w}_t),$$

که در آن η_t نرخ یادگیری است که اندازه‌ی گام را تعیین می‌کند.

پایداری گرادیان طبیعی

یکی از مهم‌ترین مزایای گرادیان طبیعی نسبت به گرادیان معمولی، پایداری عددی بالاتر آن در هنگام یادگیری است. در فضاهای پارامتری که شرطسازی (conditioning) ضعیف دارند یا متغیرها با مقیاس‌های مختلف ظاهر می‌شوند، گرادیان معمولی ممکن است باعث نوسان یا کندی همگرایی شود. اما گرادیان طبیعی با در نظر گرفتن ساختار ریمانی فضای پارامتر، جهت بهینه را به گونه‌ای تنظیم می‌کند که حرکت در آن فضا پایدارتر و کاراتر باشد. همچنین، مطالعات نظری نشان داده‌اند که گرادیان طبیعی Efficient Fisherاست؛ یعنی به صورت مجانبی (asymptotically) عملکردی معادل با برآورد بهینه‌ی دسته‌ای (Batch Estimation) دارد. به همین دلیل، پدیده‌هایی مانند "پلاتو" که در یادگیری شبکه‌های عصبی با گرادیان معمولی رخ می‌دهد، در گرادیان طبیعی یا دیده نمی‌شوند یا بسیار کاهش می‌یابند.

نتیجه

می‌توان نتیجه گرفت که هر یک از این روش‌ها برای موقعیت‌ها و ساختارهای خاصی از مسائل مناسب هستند. روش گرادیان با تندترین شیب ساده و قابل فهم است، اما در مسائل بدشرط (ill-conditioned) و با توابع پیچیده، همگرایی کندی دارد و مسیر زیگ‌زاگی طی می‌کند. روش گرادیان مزدوج با بهینه‌سازی جهت حرکت و استفاده از اطلاعات قبلی، سرعت همگرایی بهتری در مسائل مربعی یا خطی دارد و اغلب در مسائل بزرگ‌مقیاس مؤثرتر است. در مقابل، گرادیان طبیعی که بر پایه‌ی هندسه‌ی اطلاعاتی و متریک Fisherعمل می‌کند، در فضاهای پیچیده مانند آموزش شبکه‌های عصبی عملکرد بهتری دارد.

مقایسه

ویژگی	گرادیان با تندترین شیب	گرادیان مزدوج	گرادیان طبیعی
پیچیدگی محاسباتی	پایین	متوسط	بالا (به دلیل محاسبه‌ی متریکFisher)
سرعت همگرایی	کند در مسائل بدحالت	سریع‌تر در مسائل مربعی	پایدار و مناسب در فضاهای پیچیده
پایداری عددی	پایین	متوسط	بالا
کاربردها	مسائل ساده و آموزش اولیه	مسائل بزرگ‌مقیاس و خطی	شبکه‌های عصبی، سیستم‌های دینامیکی، جداسازی کور منبع

تحلیل و آنالیز پیاده سازی

در ابتدا از نظر خروجی نهایی، هر سه الگوریتم توانستند جواب دقیقی برای مسئله کمینه‌سازی تابع درجه دوم $f(x) = \frac{1}{2}x^T Ax - b^T x + c$ پیدا کنند. الگوریتم Steepest Descent معمولاً برای چنین توابعی به‌خوبی عمل می‌کند، اما در عمل به دلیل استفاده از گرادیان مستقیم ممکن است نیاز به تعداد بیشتری تکرار داشته باشد، خصوصاً اگر ماتریس A دارای عدد حالت بالایی باشد. الگوریتم Conjugate Gradient معمولاً بسیار سریع‌تر همگرا می‌شود، چون از ساختار خاص ماتریس متقارن مثبت‌معین استفاده می‌کند و جهت‌های جستجوی خود را بهینه‌تر تنظیم می‌کند. در مورد Natural Gradient Descent ، این الگوریتم از اطلاعات هندسی (یعنی ساختار ماتریس A) برای اصلاح گرادیان استفاده می‌کند و در واقع با "پیش‌شرطی‌سازی" بهبودیافته، به سمت جواب حرکت می‌کند. در این مورد خاص که A شناخته‌شده و کوچک است، Natural Gradient نیز جواب دقیقی را با نرخ همگرایی خوب ارائه می‌دهد.

از نظر پایداری عددی، Conjugate Gradient برتری دارد چون نیاز به محاسبه معکوس ماتریس ندارد و تنها ضرب ماتریسی و برداری انجام می‌دهد. در مقابل، Natural Gradient نیاز به محاسبه معکوس A دارد، که در مسائل بزرگ‌تر یا با شرط عددی ضعیف می‌تواند منجر به ناپایداری عددی شود Steepest Descent .در شرایطی که نرخ یادگیری بد انتخاب شود یا هندسه تابع بد باشد (مثل وقتی که بیضی‌های تراز بسیار کشیده‌اند)، دچار نوسان یا همگرایی کند می‌شود و از این جهت نسبت به بقیه الگوریتم‌ها حساس‌تر است.

از نظر پیچیدگی زمانی، Steepest Descent و Natural Gradient هر دو دارای پیچیدگی $O(n^2)$ در هر گام هستند (با فرض ضرب ماتریسی کامل)، اما Natural Gradient نیاز به معکوس‌گیری اولیه با پیچیدگی $O(n^3)$ دارد. این هزینه بالا در مسائل بزرگ بسیار محسوس است. در مقابل، Conjugate Gradient نه تنها از معکوس‌گیری اجتناب می‌کند بلکه اگر الگوریتم در کمتر از n گام همگرا شود، بسیار کاراتر خواهد بود؛ در واقع پیچیدگی آن تقریباً $O(nk)$ و برای مسائل بزرگ‌مقیاس بهترین گزینه است، به‌ویژه وقتی A اسپارس باشد.

مراجع

1. Natural Gradient Works Efficiently in Learning Shun-ichi Amari
RIKEN Frontier Research Program, Saitama 351-01, Japan

2. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain
Jonathan Richard Shewchuk

3. S096 Matrix Calculus for Machine Learning and Beyond
Independent Activities Period (IAP) 2023

4.Numerical Optimization, Jorge Nocedal