دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده ریاضی و علوم کامپیوتر

گزارش تمرین چهارم درس داده‌کاوی محاسباتی

هندسه یادگیری (انحنا، تعامد و هزینه محاسباتی در شبکه‌های عصبی)

نگارش
زهرا براتی

استاد درس
دکتر مهدی قطعی

تدریسیار
آقای بهنام یوسفی‌مهر

آذر ۱۴۰۴

چکیده

این پژوهش، به بررسی رفتار الگوریتم‌های بهینه‌سازی در شبکه‌های عصبی و نقش هندسه مسئله در کارایی آن‌ها می‌پردازد. در ابتدا با استفاده از یک تابع درجه‌دوم بدحالت نشان داده شد که گرادیان کاهشی به دلیل ضعف در مقیاس‌دهی، انحنا همگرایی کند و نوسانی دارد، درحالی‌که روش نیوتون و گرادیان مزدوج با بهره‌گیری از اطلاعات مرتبه دوم یا جهت‌های مزدوج، مسیر همگرایی مستقیم‌تر و سریع‌تری ایجاد می‌کنند. در ادامه، یک شبکه عصبی کم‌عمق روی دیتاست Breast Cancer آموزش داده شد و مشاهده شد که روش‌های شبه‌نیوتونی مانند L-BFGS و نیز گرادیان مزدوج، نسبت به SGD با زمان و تعداد تکرار بسیار کمتر به خطای پایین می‌رسند.

بخش سوم محدودیت‌های محاسباتی روش‌های مرتبه دوم در شبکه‌های عمیق را نشان داد: محاسبه و ذخیره هسین برای مدلی با حدود صد هزار پارامتر به حجمی در حد ده‌ها گیگابایت نیاز دارد و اجرای نیوتون خالص را غیرعملی می‌سازد؛ بنابراین استفاده از روش‌های مرتبه اول مانند SGD و Adam در عمل ناگزیر است. نتایج تجربی نیز نشان دادند که Adam به دلیل بهره‌گیری از ممان و نرخ یادگیری تطبیقی، همگرایی سریع‌تر و پایدارتری نسبت به SGD دارد. در پایان، بررسی تجزیه QR در یک مسئله رگرسیونی نشان داد که تعامدسازی داده‌ها، اگرچه لزوماً به کاهش سریع‌تر خطا منجر نمی‌شود، اما نوسانات گرادیان را کاهش داده و رفتار بهینه‌سازی را پایدارتر می‌سازد. مجموعه این یافته‌ها تأکید می‌کند که انتخاب روش بهینه‌سازی وابسته به هندسه داده، ابعاد مدل و ساختار هسین است و درک این عوامل برای طراحی الگوریتم‌های کارآمد ضروری است.

واژه‌های کلیدی:

گرادیان مزدوج، گرادیان کاهشی استاندارد، گرادیان کاهشی تصادفی، روش‌های نیوتونی و شبه‌نیوتونی، تجزیه QR

فهرست مطالب

صفحه

فصل اول

مقدمه

مقدمه

در بسیاری از مسائل، بدحالتی هسین و وجود انحناهای نامتقارن سبب می‌شود روش‌های مرتبه اول مانند گرادیان کاهشی همگرایی کند یا ناپایدار داشته باشند، در حالی‌که روش‌های مرتبه دوم در مدل‌های کوچک می‌توانند همگرایی بسیار سریع‌تری ارائه دهند. با این حال، ابعاد عظیم شبکه‌های عمیق و هزینه ذخیره و پردازش هسین، استفاده مستقیم از روش‌های نیوتونی را عملاً ناممکن می‌سازد و اهمیت روش‌های سبک‌تر و مقیاس‌پذیر مانند SGD و Adam را آشکار می‌کند.

این پروژه با هدف بررسی دقیق این پدیده‌ها طراحی شده است و از چهار منظر به موضوع می‌پردازد: تحلیل هندسی رفتار الگوریتم‌ها بر روی یک تابع بدحالت مصنوعی؛ مقایسه تجربی روش‌های مرتبه اول و شبه‌نیوتونی در یک شبکه عصبی کم‌عمق؛ ارزیابی محدودیت‌های محاسباتی روش‌های مرتبه دوم در شبکه‌های عمیق و درنهایت بررسی نقش تعامد و QR در بهبود وضعیت عددی داده‌ها و اثر آن بر پایداری گرادیان. این مراحل تصویری منسجم از تأثیر هندسه مسئله بر کارایی روش‌های بهینه‌سازی ارائه می‌دهد و ضرورت انتخاب رویکرد مناسب را در شرایط مختلف روشن می‌سازد.

فصل دوم

تحلیل‌های ریاضی (سطوح بدحالت)

تحلیل‌های ریاضی (سطوح بدحالت)

در مسائل بهینه‌سازی، شکل سطح تابع هزینه نقشی اساسی در رفتار و سرعت همگرایی الگوریتم‌های بهینه‌سازی ایفا می‌کند. در برخی از مدل‌های یادگیری ماشین، سطوحی با انحنای بسیار متفاوت در جهت‌های مختلف ظاهر می‌شوند که اصطلاحاً بدحالت[1] نامیده می‌شوند. در چنین شرایطی، الگوریتم‌های مرتبه اول مانند گرادیان کاهشی استاندارد[2] معمولاً با حرکت‌های زیگ‌زاگی و نرخ همگرایی کند مواجه می‌شوند؛ درحالی‌که روش‌های مرتبه دوم، مانند روش نیوتون، با استفاده از ماتریس هسین قادر به اصلاح مقیاس انحنا بوده و همگرایی بسیار سریع‌تری ارائه می‌کنند.

هدف این بخش، تحلیل بصری و ریاضی رفتار این روش‌ها بر روی یک تابع درجه دوم مصنوعی است که با هسینی بدحالت طراحی شده است.

۲-۱- پیاده‌سازی دستی و مصورسازی مسیر

در این بخش، ابتدا تابع هزینه مورد استفاده که یک تابع درجه دوم دومتغیره به صورت $f(v) = \frac{1}{2}\, v^T H v$ و دارای ماتریس هسین $H = \begin{bmatrix} 1 & 0 \\ 0 & 50 \end{bmatrix}$ است را وارد کرده و فرم گرادیان آن را نیز اضافه می‌کنیم. نقطه v اولیه برای اجرای الگوریتم‌ها را برابر $[4,4]$ در نظر گرفتیم. در ادامه به پیاده‌سازی الگوریتم‌های خواسته شده پرداختیم.

۲-۱-۱- روش گرادیان کاهشی استاندارد

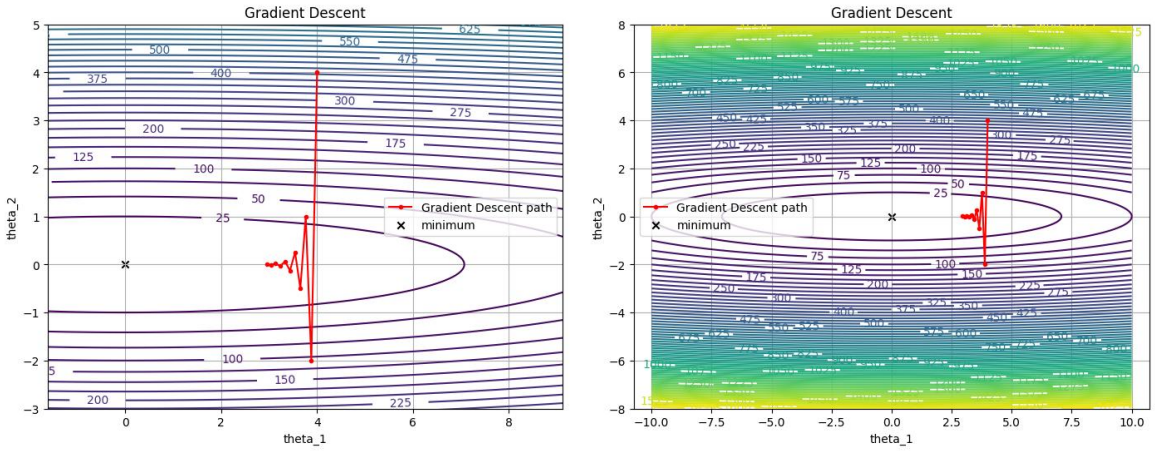
در روش گرادیان کاهشی، به‌روزرسانی پارامترها بر اساس شیب نزولی تابع، به شکل $\theta_{k+1} = \theta_k - \alpha \nabla f(\theta_k)$ صورت می‌گیرد. همچنین برای تابع درجه دوم تعریف شده، گرادیان خطی است. $(\nabla f(\theta) = H\theta)$

در پیاده‌سازی، در هر تکرار مقدار گرادیان محاسبه شده و سپس گام کاهشی با نرخ یادگیری ثابت اعمال شده است. به‌دلیل بدحالت‌بودن هسین، جهت گرادیان عمود بر جهت حرکت مطلوب قرار گرفته و الگوریتم

[1] Ill-conditioned

[2] Gradient Descent

در هر گام می‌کوشد خطا را در یک جهت تصحیح کند اما در جهت دیگر نوسان ایجاد می‌شود. این وضعیت منجر به مسیر زیگ‌زاگی در دره می‌شود.



مسیر نمودارهای مسیر گرادیان کاهشی نشان می‌دهند که الگوریتم با وجود شروع در نقطه $[4,4]$، به‌جای حرکت مستقیم، وارد نوسان‌های شدید در راستای θ_2 می‌شود. دلیل این پدیده آن است که هسین در این راستا مقدار ۵۰ دارد و بنابراین تابع در این جهت بسیار تندتر است. در نتیجه، حتی یک گام کوچک می‌تواند باعث پرش‌های زیاد و بازگشت‌های مکرر شود که به وضوح مخصوصاً در تصویر با بزرگ‌نمایی مشهود است. پس از تعداد محدودی تکرار، θ_2 تقریباً صفر شده اما θ_1 هنوز با سرعتی بسیار کم به سمت نقطه بهین حرکت می‌کند.

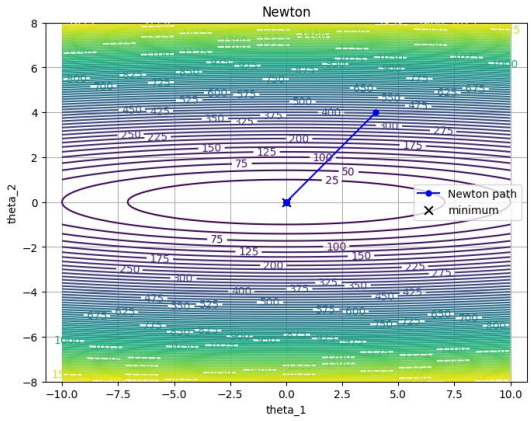
۲-۱-۲- روش نیوتون

روش نیوتون از تقریبی درجه دوم به شکل زیر برای تابع هزینه استفاده می‌کند.

$$\theta_{k+1} = \theta_k - H^{-1}\nabla f(\theta_k)$$

از آن‌جا که هسین در این مثال ثابت و قابل محاسبه است، نیوتون قادر است جهت حرکت را بر اساس انحنای واقعی سطح اصلاح کند.

در پیاده‌سازی، در هر تکرار گرادیان محاسبه شده و سپس با ضرب در ماتریس معکوس هسین اصلاح شده است. نتیجه آن در این مثال، حرکت مستقیم به سمت نقطه بهینه $[0,0]$ تنها در یک گام است.



مسیر روش نیوتون در نمودار نشان می‌دهد که الگوریتم مستقیماً به سمت مرکز بیضی‌ها حرکت می‌کند و پس از یک گام، عملاً در نقطه بهینه قرار می‌گیرد. علت آن است که H^{-1} دامنه انحنا در جهت θ_2 را ۵۰ برابر کوچک می‌کند و مسیر گرادیان را به شکلی مقیاس‌دهی می‌کند که نوسان‌ها کاملاً حذف می‌شوند.

۲-۱-۳- روش گرادیان مزدوج[3]

روش گرادیان مزدوج برای حل مسائل مربعی با هسین متقارن و معین‌مثبت طراحی شده است و بدون نیاز به ذخیره‌سازی هسین به‌صورت صریح، به‌صورت تکراری جهت‌هایی را می‌سازد که نسبت به ماتریس هسین مزدوج هستند.

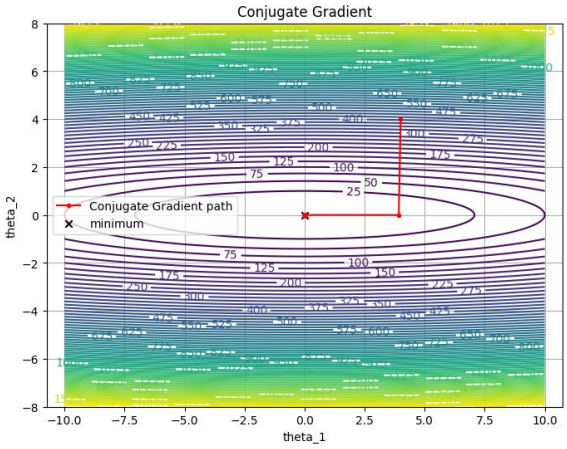
مراحل اجرای این الگوریتم به شرح زیر است:

$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0$
if $\mathbf{r}_0$ is sufficiently small, then return $\mathbf{x}_0$ as the result
$\mathbf{p}_0 := \mathbf{r}_0$
$k := 0$
repeat
$\alpha_k := \frac{\mathbf{r}_k^\mathsf{T}\mathbf{r}_k}{\mathbf{p}_k^\mathsf{T}\mathbf{A}\mathbf{p}_k}$
$\mathbf{x}_{k+1} := \mathbf{x}_k + \alpha_k\mathbf{p}_k$
$\mathbf{r}_{k+1} := \mathbf{r}_k - \alpha_k\mathbf{A}\mathbf{p}_k$
if $\mathbf{r}_{k+1}$ is sufficiently small, then exit loop
$\beta_k := \frac{\mathbf{r}_{k+1}^\mathsf{T}\mathbf{r}_{k+1}}{\mathbf{r}_k^\mathsf{T}\mathbf{r}_k}$
$\mathbf{p}_{k+1} := \mathbf{r}_{k+1} + \beta_k\mathbf{p}_k$
$k := k + 1$
end repeat
return $\mathbf{x}_{k+1}$ as the result

از طرفی با در نظر گرفتن $\nabla f(x_k) = Ax_k - b$ می‌توان گفت $r_0 = -\nabla f(x_k)$.

[3] Conjugate Gradient

پیاده‌سازی این روش با این منطق انجام گرفت.



نتایج مشاهده‌شده از روش CG نشان می‌دهد که الگوریتم، در دو گام به نقطه بهینه می‌رسد. مسیر طی‌شده فاقد نوسانات شدید گرادیان کاهشی است و با وجود آن‌که H^{-1} محاسبه نمی‌شود، جهت‌های جستجو به‌گونه‌ای ساخته شده‌اند که متعامد نسبت به هسین باشند و به‌همین دلیل، مسیر حرکت بسیار کارآمدتر از GD است.

۲-۲- تحلیل

می‌توان گفت بدحالتی سطح تابع باعث کندی و نوسان در روش‌های مرتبه اول می‌شود. از طرفی، روش نیوتون با استفاده از اطلاعات انحنا، مسیر بهینه را اصلاح می‌کند و با سرعت بسیار بالا همگرا می‌شود. در این روش با اصلاح صورت گرفته هر دو مؤلفه با سرعت یکسان به سمت مبدأ حرکت می‌کنند و به تعبیری، سطح بیضی در فضای نیوتون به دایره تبدیل می‌شود. همچنین، روش گرادیان مزدوج بدون نیاز به هسین صریح، در مسائل مربعی عملکردی معادل روش‌های مرتبه دوم ارائه می‌دهد.

فصل سوم

شبکه عصبی کلاسیک (فضای نیوتونی)

شبکه عصبی کلاسیک (فضای نیوتونی)

در این بخش، سه الگوریتم بهینه‌سازی در یک شبکه عصبی کوچک مورد بررسی قرار می‌گیرد. این الگوریتم‌ها شامل گرادیان کاهشی تصادفی[4]، روش شبه‌نیوتونی L-BFGS[5] و روش گرادیان مزدوج است.

۳-۱- دیتاست و مدل

در این مرحله از یک مسئله طبقه‌بندی باینری سرطان سینه استفاده شده است. ابتدا داده‌ها بارگذاری و در یک DataFrame قرار گرفتند. برای استانداردسازی ویژگی‌ها، از StandardScaler استفاده و هر ویژگی به میانگین صفر و واریانس یک نگاشته شد. در نهایت، داده‌ها به‌صورت تصادفی به مجموعه‌های آموزش و آزمون با نسبت ۳۰/۷۰ تقسیم شدند.

در هر سه روش، شبکه با یک لایه مخفی شامل ۵ نورون ساخته شد.

۳-۲- مسابقه بهینه‌سازها

در این گام، به بررسی روش‌های بهینه‌سازی مذکور می‌پردازیم.

۳-۲-۱- روش SGD

برای پیاده‌سازی این بخش از MLPClassifier بهره بردیم. همچنین پس از fit، مقادیر loss_curve_ و زمان اجرا ثبت شده‌اند تا هم روند کاهش loss بر حسب تکرار و هم رابطه loss و زمان در نمودارها نمایش داده شود.

۳-۲-۲- روش L-BFGS

در این بخش نیز برای پیاده‌سازی این بخش از MLPClassifier بهره بردیم. در MLPClassifier هنگامی

[4] stochastic gradient descent (SGD)

[5] Limited-memory Broyden–Fletcher–Goldfarb–Shanno

که solver برابر lbfgs است، خود کتابخانه روند بهینه‌سازی و تعداد تکرارها را مدیریت می‌کند. تنها مقدار loss نهایی از طریق ویژگی loss_ در دسترس است؛ لذا در نمودارها L-BFGS تنها به‌صورت یک نقطه (و نه یک منحنی کامل) نمایش داده شده است.

۳-۲-۳- روش CG

برای استفاده از CG به کمک scipy.optimize.minimize، لازم است مسئله آموزش شبکه عصبی را به‌صورت دستی بنویسیم.

ابتدا تمام وزن‌ها و بایاس‌ها در یک بردار یک‌بعدی θ کنار هم چیده شده‌اند.

$$\theta = (W1, b1, W2, b2)$$

که در آن:

- $W_1 \in \mathbb{R}^{d\times h}$:وزن‌های ورودی به لایه مخفی با ۵ نورون،
- $b_1 \in \mathbb{R}^h$:بایاس لایه مخفی،
- $W_2 \in \mathbb{R}^{h\times 1}$:وزن‌های لایه خروجی،
- $b_2 \in \mathbb{R}$:بایاس خروجی.

تابع unpack_theta با گرفتن بردار θ این چهار ماتریس/بردار را بازسازی می‌کند.

در ادامه نیاز به تعریف یک تابع فعال‌ساز داریم. از توابع فعال‌ساز رایج می‌توان به تابع خطی، تابع سیگموید، تابع tanh، تابع ReLU و مواردی از این دست اشاره کرد.

در ادامه پیاده‌سازی این بخش، نگاشت‌های $Z_1 = XW_1 + b_1, A_1 = \tanh(Z_1)$ و $Z_2 = A_1W_2 + b_2, \hat{y} = \sigma(Z_2)$ تعریف می‌شود که در آن σ تابع سیگموید است.

همچنین برای طبقه‌بندی باینری، از تابع هزینه کراس‌انتروپی استفاده شده است:

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N}[y_i log\,(\hat{y}_i) + (1-y_i) log\,(1-\hat{y}_i)],$$

که در آن N تعداد نمونه‌های آموزش و $y_i \in \{0,1\}$ است. در کد برای جلوگیری از خطاهای عددی، یک مقدار کوچک $\varepsilon = 10^{-12}$ به آرگومان لگاریتم اضافه شده است.

در ادامه گرادیان تابع هزینه نسبت به θ از طریق روش استاندارد پس‌انتشار خطا[6] به‌دست آمده است. روابط مشتق‌ها به‌صورت زیر است:

گرادیان نسبت به خروجی خطی لایه دوم: $\frac{\partial J}{\partial z_2} = \frac{1}{N}(\hat{y} - y)$

مشتق‌ها نسبت به وزن‌ها و بایاس‌های لایه دوم: $\frac{\partial J}{\partial W_2} = A_1^T \frac{\partial J}{\partial z_2},\ \frac{\partial J}{\partial b_2} = \sum_i \frac{\partial J}{\partial z_{2,i}}$

سیگنال خطا برای لایه اول: $\frac{\partial J}{\partial A_1} = \frac{\partial J}{\partial z_2} W_2^T,\ \frac{\partial J}{\partial z_1} = \frac{\partial J}{\partial A_1} \odot (1 - A_1^2)$

گرادیان نسبت به W_1 و b_1: $\frac{\partial J}{\partial W_1} = X^T \frac{\partial J}{\partial z_1},\ \frac{\partial J}{\partial b_1} = \sum_i \frac{\partial J}{\partial z_{1,i}}$

تمام این گرادیان‌ها در انتها به‌صورت یک بردار کنار هم چیده شده‌اند تا شیب کامل نسبت به θ ساخته شود $\nabla_\theta J(\theta) = \left(\frac{\partial J}{\partial W_1}, \frac{\partial J}{\partial b_1}, \frac{\partial J}{\partial W_2}, \frac{\partial J}{\partial b_2}\right)$.

تابع loss_and_grad این محاسبات را انجام داده و در اختیار scipy.optimize.minimize قرار می‌دهد.

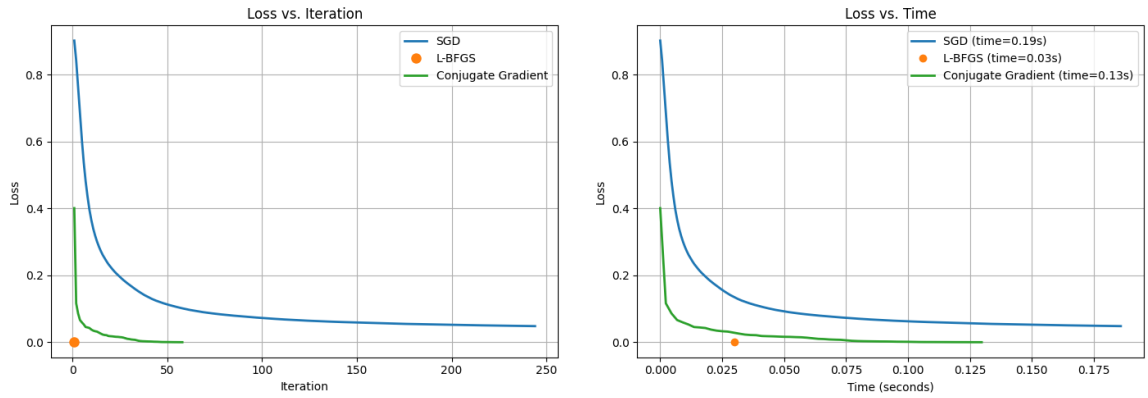
۳-۳- نمودار مقایسه‌ای و تحلیل

در این بخش به بررسی خروجی‌ها می‌پردازیم.

روش	Accuracy	Time (s)
SGD	0.9649	0.19
L-BFGS	0.9591	0.03
CG	0.9532	0.13

[6] Backprop

همان‌طور که در جدول قابل مشاهده است، هر سه روش به دقت‌های بسیار مشابه (در حدود ۹۵–۹۶ درصد) رسیده‌اند و تفاوت اصلی آن‌ها در سرعت همگرایی است، نه در کیفیت نهایی مدل.



در نمودار Loss-Time مشاهده می‌شود که کاهش خطا در SGD به‌صورت آهسته و تدریجی رخ می‌دهد و حتی پس از حدود ۰.۱۷۵ ثانیه همچنان به مقدار صفر نرسیده است. این رفتار کاملاً با ماهیت مرتبه‌اول این روش سازگار است؛ زیرا SGD تنها از جهت شیب استفاده می‌کند و به دلیل عدم اصلاح انحنا ناچار است با گام‌های کوچک و پیوسته در فضای پارامترها حرکت کند.

در مقابل، روش L-BFGS که از اطلاعات شبه‌هسین بهره می‌برد، تنها در مدت حدود ۰.۰۳ ثانیه به loss بسیار کوچک و حدود صفر می‌رسد. این سرعت ناشی از توانایی روش در مقیاس‌دهی درست جهات جستجو است. روش گرادیان مزدوج نیز رفتاری مشابه نشان می‌دهد؛ در چند صدم ثانیه نخست بخش عمده خطا کاسته می‌شود و در حدود ۰.۱۳ ثانیه loss تقریباً به صفر می‌رسد. دلیل این عملکرد، تولید جهت‌های جستجویی است که نسبت به ماتریس هسین مزدوج‌اند و از بازگشت‌های زائد و نوساناتی که SGD تجربه می‌کند جلوگیری می‌شود.

تحلیل نمودار Loss-Iteration نیز پیام مشابهی دارد. درSGD ، اگرچه کاهش اولیه loss سریع است، اما پس از چند تکرار سرعت همگرایی به‌شدت افت می‌کند و منحنی به‌صورت دنباله‌دار ادامه می‌یابد. در مقابل، L-BFGS در همان نخستین تکرار lossی به‌دست آورده که SGD تنها پس صدها تکرار به آن نزدیک می‌شود. روش گرادیان مزدوج نیز در کمتر از ۶۰ تکرار loss را تقریباً به صفر می‌رساند، که نسبت به رفتار طولانی SGD نشان‌دهنده کارایی بسیار بالاتر آن است.

از این شواهد می‌توان نتیجه گرفت که در شبکه‌های کوچک، روش‌های مبتنی‌بر اطلاعات انحنا به‌طور واضح سریع‌تر از SGD به خطای ناچیز می‌رسند. علت نظری این تفاوت نیز روشن است SGD فاقد سازوکار اصلاح

انحنا است و در دره‌های باریک فضای پارامترها ناچار به حرکت زیگ‌زاگی و گام‌های کوچک می‌شود، درحالی‌که L-BFGS با استفاده از هسین و CG با ساخت جهت‌های مزدوج، مقیاس‌بندی صحیحی میان ابعاد مختلف مسئله برقرار کرده و مسیرهای کوتاه‌تر و مستقیم‌تری به سوی نقطه بهین پیدا می‌کنند.

فصل چهارم

شبکه عمیق و تله مقیاس‌پذیری

شبکه عمیق و تله مقیاس‌پذیری

در این بخش، برای بررسی تله مقیاس‌پذیری، یک شبکه عمیق روی دیتاست Fashion-MNIST تعریف شده است.

۴-۱- مدل عمیق

تصاویر ۲۸×۲۸ ابتدا نرمال‌سازی شده و به بردارهای ۷۸۴بعدی تبدیل شده‌اند. سپس مدلی با سه لایه مخفی ۱۰۰تایی و تابع فعال‌سازی ReLU و یک لایه خروجی ۱۰تایی با softmax ساخته شده است.

۴-۲- محاسبه ابعاد هسین

در ادامه با مشاهده خلاصه‌ای از مدل، داریم:

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 100)	78,500
dense_1 (Dense)	(None, 100)	10,100
dense_2 (Dense)	(None, 100)	10,100
dense_3 (Dense)	(None, 10)	1,010

Total params: 99,710 (389.49 KB)
Trainable params: 99,710 (389.49 KB)
Non-trainable params: 0 (0.00 B)

همان‌طور که مشاهده می‌شود، تعداد کل پارامترها برابر ۹۹۷۱۰ است.

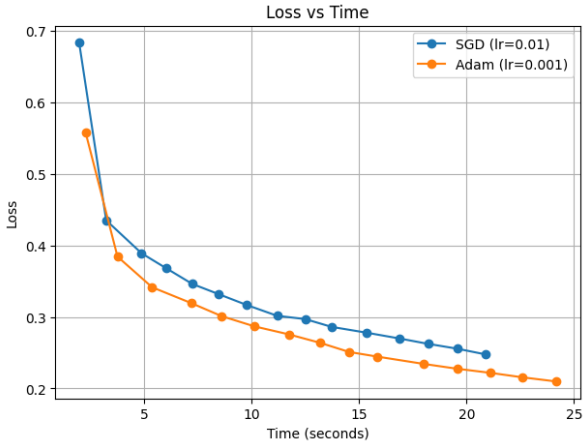
اگر بخواهیم روش نیوتون خالص را روی این مدل اجرا کنیم، به‌طور نظری باید ماتریس هسین تابع هزینه نسبت به تمام این پارامترها را تشکیل دهیم. هسین در این حالت ماتریسی با ابعاد $N \times N$ خواهد بود که حجم حافظه لازم حدوداً برابر ۳۷ گیگابایت است. این فقط حافظه لازم برای هسین است؛ در روش نیوتون علاوه بر ذخیره H، باید آن را در هر گام معکوس کرده یا حداقل حل دستگاه خطی $\nabla J = H\Delta\theta$ را انجام دهیم که پیچیدگی زمانی آن در بدترین حالت از مرتبه $O(N^3)$ است. برای N در حد 10^5، این هزینه هم از نظر حافظه و هم از نظر زمان عملاً غیرقابل‌قبول است. به همین دلیل است که روش نیوتون خالص در شبکه‌های عمیق غیرعملی محسوب می‌شود.

۴-۳- جایگزین‌ها

در این گام به آموزش مدل با Adam و SGD می‌پردازیم.

تابع train_with_timing ابتدا مدل را با بهینه‌ساز دلخواه و loss کراس‌آنتروپی کامپایل می‌کند و سپس حلقه‌ای روی epochها اجرا می‌کند که در هر تکرار یک epoch کامل fit می‌شود، از زمان شروع آموزش تا پایان آن epoch اندازه‌گیری شده و مقدار loss همان epoch از شیء history استخراج و در آرایه‌ها ثبت می‌شود.

در آزمایش حاضر، یک‌بار مدل با SGD(learning_rate=0.01, momentum=0.9) و یک‌بار با Adam(learning_rate=0.001) آموزش داده شده است و خروجی‌های چاپ‌شده نشان می‌دهند که برای هر دو روش loss به‌طور یکنواخت کاهش می‌یابد، اما Adam تقریباً در تمام زمان‌ها مقدار loss پایین‌تری نسبت به SGD دارد.



فصل پنجم

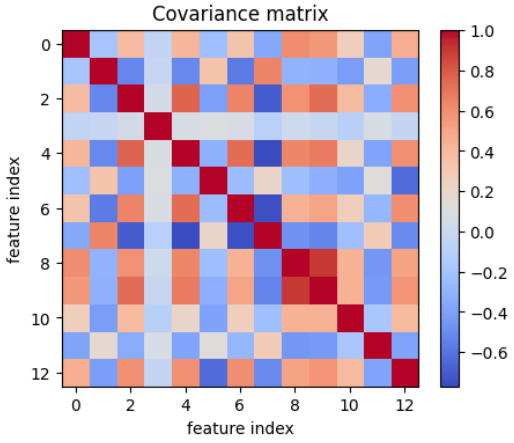
تعامد و QR (رویکرد داده کاوی)

تعامد و QR (رویکرد داده کاوی)

در این بخش، به‌جای تغییر الگوریتم بهینه‌سازی، هندسه مسئله را تغییر می‌دهیم تا اثر تعامد بر وضعیت مسئله و بر رفتار گرادیان کاهشی بررسی شود. هدف نشان دادن این نکته است که در مسائل رگرسیونی با ویژگی‌های هم‌بسته، استفاده از تبدیل‌های متعامد مانند تجزیه QR می‌تواند چه اثراتی بر مسیر همگرایی داشته باشد.

۵-۱- آماده‌سازی داده‌های همبسته

در ابتدا دیتاست Boston Housing بارگذاری و داده‌های ناقص حذف شد. سپس ویژگی‌ها استانداردسازی و ماتریس کوواریانس آن‌ها محاسبه و ترسیم شد. تصویر حاصل نشان می‌دهد که بسیاری از ویژگی‌ها دارای هم‌بستگی‌های مثبت یا منفی قابل‌توجهی هستند؛ وجود این هم‌بستگی‌ها مستقیماً نشان‌دهنده بدحالتی ماتریس $X^T X$ است.



۵-۲- تجزیه QR

در ادامه با استفاده از numpy.linalg.qr ماتریس ویژگی‌ها تجزیه شده و ماتریس Q ویژگی‌های جدیدی هستند که متعامدند.

این بدان معناست که ویژگی‌های جدید (ستون‌های Q) نسبت به یکدیگر هم‌بستگی ندارند و هر ستون حامل اطلاعاتی مستقل است. این تبدیل ساختار هندسی مسئله را بدون تغییر در فضای پاسخ دگرگون

می‌کند. بدین شکل که تابع هزینه به‌جای بیضی‌های کشیده، کانتورهایی نزدیک‌تر به دایره پیدا می‌کند و انتظار می‌رود گرادیان کاهشی در چنین فضایی رفتاری منظم‌تر داشته باشد.

۵-۳- تاثیر بر گرادیان کاهشی

برای بررسی تجربی رفتار گرادیان کاهشی در چنین فضایی، یک مدل رگرسیون خطی ساده با استفاده از گرادیان کاهشی بر هر دو ماتریس X و Q آموزش داده شد. تابع بهینه‌سازی در هر epoch مقدار خطا را ثبت و همچنین زمان سپری‌شده را محاسبه کرد.

۵-۴- تحلیل

نمودار کاهش خطا نشان می‌دهد که گرادیان کاهشی بر روی داده‌های خام X با سرعت بیشتری به مقدارهای بسیار کوچک loss می‌رسد، درحالی‌که همان بهینه‌ساز روی داده‌های متعامدسازی‌شده Q با آهنگی کندتر کاهش می‌یابد و در بازه‌های زمانی برابر، خطای بیشتری دارد. با وجود این تفاوت در سرعت، نکته مهم دیگری در رفتار Q قابل مشاهده است و آن این است که نمودار مربوط به Q نوسان‌های بسیار کمتر و روندی یکنواخت‌تر نسبت به X دارد. این تفاوت رفتاری ناشی از ماهیت تعامد در ستون‌های Q است؛ زیرا وقتی ویژگی‌ها متعامد هستند، اثرات متقابل میان ابعاد مختلف از بین می‌رود و گرادیان هر جهت بدون مزاحمت جهات دیگر تغییر می‌کند. نتیجه آن است که مسیر گرادیان کاهشی روی Q منظم‌تر است.

در مقابل، داده‌های اولیهX ، هرچند در این مسئله خاص منجر به کاهش سریع‌تری در loss شده‌اند، اما این کاهش همراه با نوسان در شیب نمودار است. این رفتار را می‌توان اثر هم‌بستگی میان ویژگی‌ها دانست؛ در چنین شرایطی ماتریس X^TX بدحالت‌تر می‌شود و گرادیان در برخی جهت‌ها بیش از اندازه بزرگ و در برخی جهت‌ها بیش از اندازه کوچک می‌شود، که همین موضوع نوسانات دوره‌ای در کاهش خطا را ایجاد می‌کند.

شاید بتوان گفت، شدت این بدحالتی در دیتاست مورد استفاده زیاد نیست. یعنی هم‌بستگی‌ها وجود دارند، اما در حدی نیستند که سبب انفجار یا فروپاشی شدید گرادیان شوند. همین موضوع توضیح می‌دهد که چرا سرعت همگرایی روی X در این مثال خاص بهتر از Q ظاهر شده است. اگر داده‌ها دارای هم‌بستگی‌های

بسیار قوی یا تقریباً خطی بودند، معمولاً منحنی Q سریع‌تر و یکنواخت‌تر کاهش پیدا می‌کرد؛ اما در این دیتاست، هم‌بستگی‌ها متوسطاند و ساختار اصلی X همچنان برای مدل رگرسیون خطی مفیدتر است.

درنهایت می‌توان گفت که تجزیه QR الزاماً همگرایی را سریع‌تر نمی‌کند، اما تقریباً همیشه آن را پایدارتر و قابل پیش‌بینی‌تر می‌سازد .کاهش نوسان در منحنی Q نشانه مستقیم همین پایداری است. در مقابل، X سرعت بیشتر ولی پایداری کمتر دارد، که بازتابی از هم‌بستگی‌های موجود میان ویژگی‌هاست. بنابراین تفاوت رفتار دو نمودار را می‌توان نتیجه ترکیبی از هندسه مسئله، شدت هم‌بستگی‌ها و مناسب‌بودن مقیاس ویژگی‌ها برای بهینه‌ساز دانست.

فصل ششم

جمع‌بندی و نتیجه‌گیری

جمع‌بندی و نتیجه‌گیری

این پروژه نشان داد که رفتار الگوریتم‌های بهینه‌سازی در شبکه‌های عصبی به‌طور مستقیم تحت‌تأثیر هندسه تابع هزینه، وضعیت عددی هسین و ابعاد مدل قرار دارد. در مسئله مصنوعی بدحالت مشاهده شد که گرادیان کاهشی به‌دلیل مقیاس‌دهی نامناسب انحنا همگرایی کند و نوسانی دارد، در حالی‌که روش نیوتون و گرادیان مزدوج، با استفاده از اطلاعات مرتبه دوم یا جهت‌های مزدوج، مسیر هموارتر و سریع‌تری به سمت نقطه بهینه پیدا می‌کنند. در شبکه کم‌عمق نیز روش‌های L-BFGS و CG در زمان و تکرار بسیار کمتر از SGD به خطای پایین رسیدند، که تأییدکننده مزیت روش‌های مبتنی‌بر انحنا در مدل‌های کوچک است.

در شبکه عمیق، محاسبه و ذخیره هسین با ابعاد بسیار بالا، استفاده مستقیم از روش‌های نیوتونی را غیرممکن کرد و نشان داد که روش‌های سبک‌تر مانند SGD و Adam تنها گزینه عملی‌اند. نتایج تجربی نیز نشان داد که Adam، رفتار سریع‌تر و پایدارتری نسبت به SGD دارد. در نهایت، بررسی QR در مسئله رگرسیونی نشان داد که تعامدسازی داده‌ها اگرچه سرعت همگرایی را الزاماً افزایش نمی‌دهد، اما نوسان گرادیان را کاهش داده و پایداری بهینه‌سازی را بهبود می‌بخشد. مجموع این نتایج بیانگر آن است که انتخاب روش بهینه‌سازی باید مبتنی‌بر هندسه مسئله، ابعاد مدل و ساختار داده انجام شود.

منابع و مراجع

https://scikit-learn.org

https://docs.scipy.org

https://lamastex.github.io

https://python-data-science.readthedocs.io

https://h1ros.github.io

https://en.wikipedia.org

https://www.youtube.com/@TechWithHasanAbbasi

پیوست‌ها

لینک گیت‌هاب پروژه:

https://github.com/ZahraBarati99/Geometry-of-Learning