دانشگاه صنعتي اميركبير
(پلی‌تکنیک تهران)
دانشکده علوم ریاضی

گزارش پروژه درس داده کاوی محاسباتی
پروژه 3
رشته علوم کامپیوتر گرایش هوش مصنوعی و محاسبات نرم

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

نگارش
هومن ذوالفقاری

استاد راهنما
دکتر مهدی قطعی

تدریس یار
مهندس بهنام یوسفی مهر

آذر 1404

اینجانب هومن ذوالفقاری متعهد می‌شوم که مطالب مندرج در این پایان نامه حاصل کار پژوهشی اینجانب تحت نظارت و راهنمایی اساتید دانشگاه صنعتي اميركبير بوده و به دستاوردهاي ديگران كه در اين پژوهش از آنها استفاده شده است مطابق مقررات و روال متعارف ارجاع و در فهرست منابع و مآخذ ذكر گرديده است. اين پايان نامه قبلاً براي احراز هيچ مدرك هم‌سطح يا بالاتر ارائه نگرديده است.

در صورت اثبات تخلف در هر زمان، مدرك تحصيلي صادر شده توسط دانشگاه از درجه اعتبار ساقط بوده و دانشگاه حق پيگيري قانوني خواهد داشت.

كليه نتايج و حقوق حاصل از اين پايان نامه متعلق به دانشگاه صنعتي اميركبير مي‌باشد. هرگونه استفاده از نتايج علمي و عملي، واگذاري اطلاعات به ديگران يا چاپ و تكثير، نسخه‌برداري، ترجمه و اقتباس از اين پايان نامه بدون موافقت كتبي دانشگاه صنعتي اميركبير ممنوع است.
نقل مطالب با ذكر مآخذ بلامانع است.

در صفحه تعهدنامه اصالت اثر، در قسمت بالا سمت چپ، تاریخ دفاع خود را جایگزین تاریخ نوشته شده کنید.

همچنین در صفحه تعهدنامه اصالت اثر، در خط اول، نام و نام خانوادگی خود را به صورت کامل با نام و نام خانوادگی نمونه، جایگزین کنید. در انتهای متن تعهد، در قسمت امضا نیز باید نام و نام خانوادگی کامل خود را وارد نماید.

هومن ذوالفقاری

امضا

چکیده

این پروژه به بررسی رفتار الگوریتم‌های بهینه‌سازی از دیدگاه هندسی می‌پردازد. ابتدا نشان داده می‌شود که در توابع بدحالت، گرادیان نزولی به‌دلیل عدد وضعیت بالا دچار زیگ‌زاگ و همگرایی کند می‌شود، در حالی‌که روش‌های مرتبه‌دوم مانند نیوتون و گرادیان مزدوج عملکرد بسیار بهتری دارند. سپس با آزمایش روی یک شبکهٔ کوچک مشاهده می‌شود که روش‌های شبه‌نیوتونی در ابعاد پایین کارآمدتر از روش‌های مرتبه‌اول‌اند، اما در شبکه‌های عمیق هزینهٔ محاسباتی هسین استفاده از آنها را غیرعملی می‌کند و روش‌هایی مانند SGD و Adam تنها گزینه‌های قابل‌استفاده هستند. در پایان، با اعمال تجزیهٔ QR و ایجاد داده‌های متعامد نشان داده شد که بهبود هندسهٔ ورودی موجب کاهش عدد وضعیت و تسریع همگرایی الگوریتم‌های مرتبه‌اول می‌شود. این نتایج نقش اساسی هندسهٔ داده و مدل را در کارایی فرایند یادگیری برجسته می‌کند.

واژه‌های کلیدی:

هندسه یادگیری، گرادیان کاهشی، روش نیوتون، گرادیان مزدوج، شبکه‌های عصبی عمیق، هسین، تجزیه QR، تعامد، پیش‌شرطی‌سازی، همگرایی.

فهرست مطالب

صفحه

فصل اول
مقدمه

مقدمه

مباحث هندسهٔ یادگیری در سال‌های اخیر تبدیل به یکی از پایه‌های درک رفتار الگوریتم‌های بهینه‌سازی در شبکه‌های عصبی شده‌اند. با وجود اینکه روش‌های مرتبه‌دوم مانند نیوتون از نظر تئوری دارای همگرایی بسیار سریع‌تری هستند، در عمل مشاهده می‌شود که مدل‌های عمیق همچنان عمدتاً با روش‌های مرتبه‌اول) گرادیان ساده یا انواع مومنتوم مثل (Adam آموزش داده می‌شوند

.

هدف این پروژه بررسی این پرسش است که **چرا روش‌های مرتبه‌اول در یادگیری عمیق بهتر هستند،** و چگونه **هندسهٔ داده‌ها** (نه فقط الگوریتم) می‌تواند سرعت همگرایی را تغییر دهد.

برای این کار چهار محور اصلی بررسی شده است:

1. رفتار روش‌ها روی توابع **بدحالت** (Ill-conditioned)
2. کارایی روش‌های مرتبه‌اول و شبه‌نیوتونی روی یک **شبکهٔ کوچک**
3. محل شکست روش نیوتون روی **شبکه‌های عمیق**
4. نقش **تعامد (Orthogonality)** و **QR** در بهبود هندسه مسئله

فصل دوم
تحلیل های ریاضی (سطوح بدحالت)

تحلیل های ریاضی (سطوح بدحالت)

2-1- تعریف تابع آزمایشی

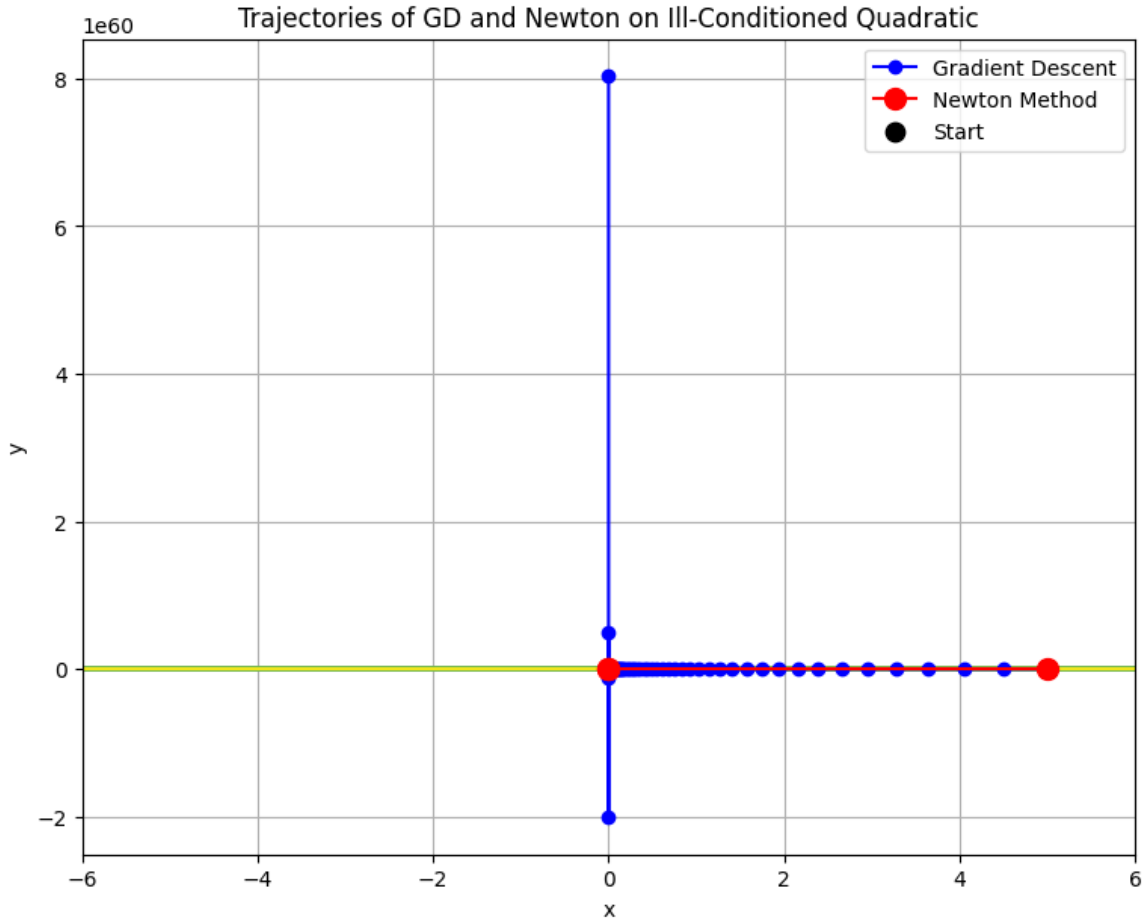
برای مشاهده رفتار بهینه‌سازها یک تابع درجه‌دو مصنوعی با ماتریس هسین زیر ساخته شد:

$$H = \begin{pmatrix} 1 & 0 \\ 0 & 50 \end{pmatrix}$$

این ماتریس دارای **عدد وضعیت بزرگ** است:

$$\kappa = \frac{\lambda_{\max}}{\lambda_{\min}} = 50$$

بنابراین کانتورهای تابع بیضی‌های بسیار کشیده هستند و گرادیان ساده در چنین دره‌هایی دچار **زیگ زاگ** می‌شود.



2-2- روش‌های پیاده‌سازی‌شده

سه روش از پایه پیاده سازی شدند:

1. گرادیان کاهشی (GD)

$$\theta_{t+1} = \theta_t - \alpha \nabla J(\theta_t)$$

در این مسئله، گرادیان برابر $H\theta$ است GD فقط جهت شیب را می‌بیند و نسبت به عدد وضعیت بسیار حساس است.

2. روش نیوتون

$$\theta_{t+1} = \theta_t - H^{-1} \nabla J(\theta_t)$$

با وارد کردن H^{-1}، گام‌ها از «بیضی» به «دایره» نگاشت می‌شوند؛ بنابراین از زیگ‌زاگ جلوگیری می‌شود.

3. گرادیان مزدوج (CG)

روش میانی که گام‌های جستجو را نسبت به ماتریس H **متعامد مزدوج** انتخاب می‌کند؛ بدون نیاز به ذخیرهٔ هسین.

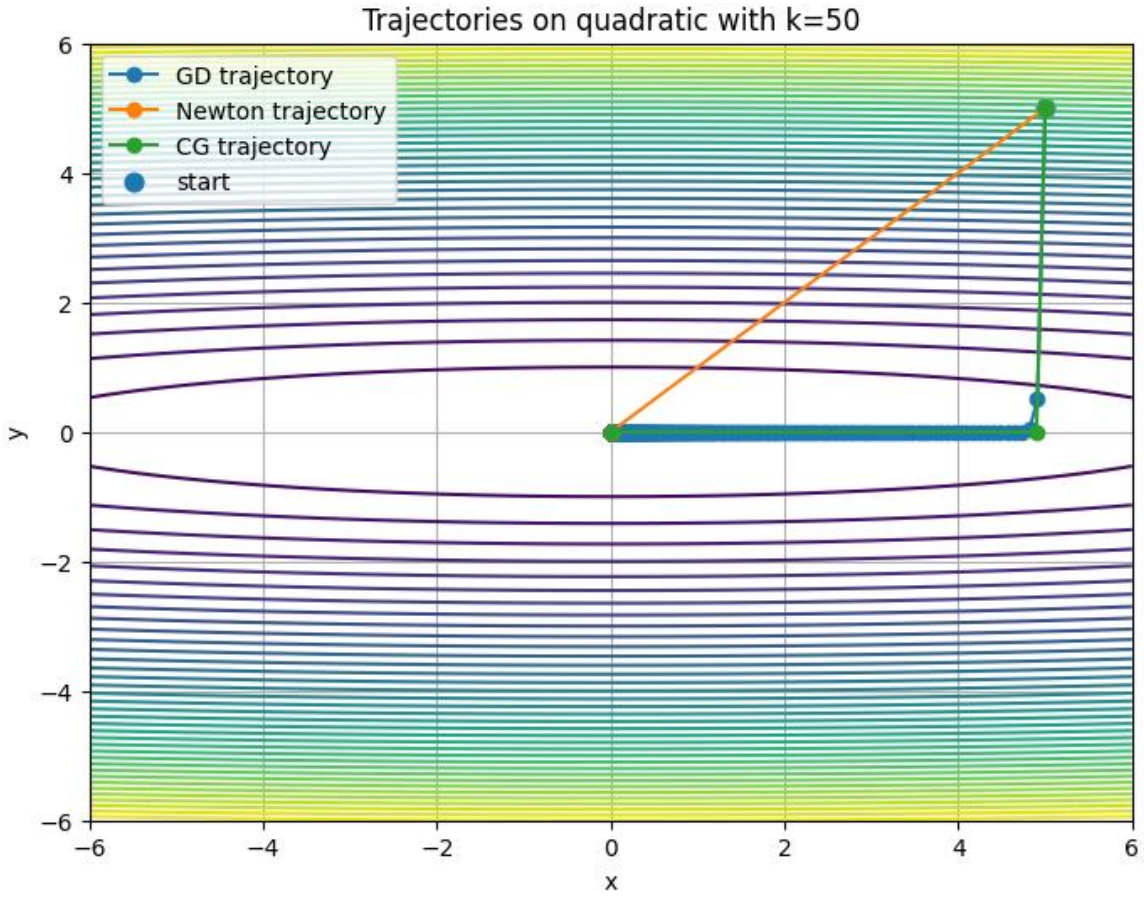
2-3- مشاهدات تجربی

- **GD** مسیر به وضوح زیگ‌زاگی است؛ هر بار در جهت عمود بر مسیر واقعی منحرف می‌شود. دلیل: محور y انحنای بسیار بزرگ دارد و گرادیان در جهت عمودی بسیار قوی‌تر است.
- **Newton** در یک گام مستقیماً به مرکز می‌رود؛ همگرایی **درجه دو** .دلیل: نیوتون اطلاعات انحنا (H^{-1}) را دارد و مستقیماً «جهت بهینه» را می‌داند.
- **CG** عملکرد بینابینی دارد؛ سریع‌تر از GD ولی کندتر از نیوتون. در حالت درجه دوم باید در 2گام (تعداد ابعاد) همگرا شود. این روش مثل نیوتون رفتار می‌کند اما **H را ذخیره نمی‌کند**.

2-4-وقتی عدد وضعیت (Condition Number) زیاد می‌شود:

GD به شدت کند می‌شود (تا صدها یا هزاران تکرار ممکن است).

نیوتون عملاً مستقل از κ در چند تکرار همگراست.
CG سرعتش کمتر از نیوتون اما تقریباً پایدار است.



فصل سوم
شبکه عصبی کلاسیک (فضای نیوتونی)

شبکه عصبی کلاسیک (فضای نیوتونی)

2-5- مدل و داده

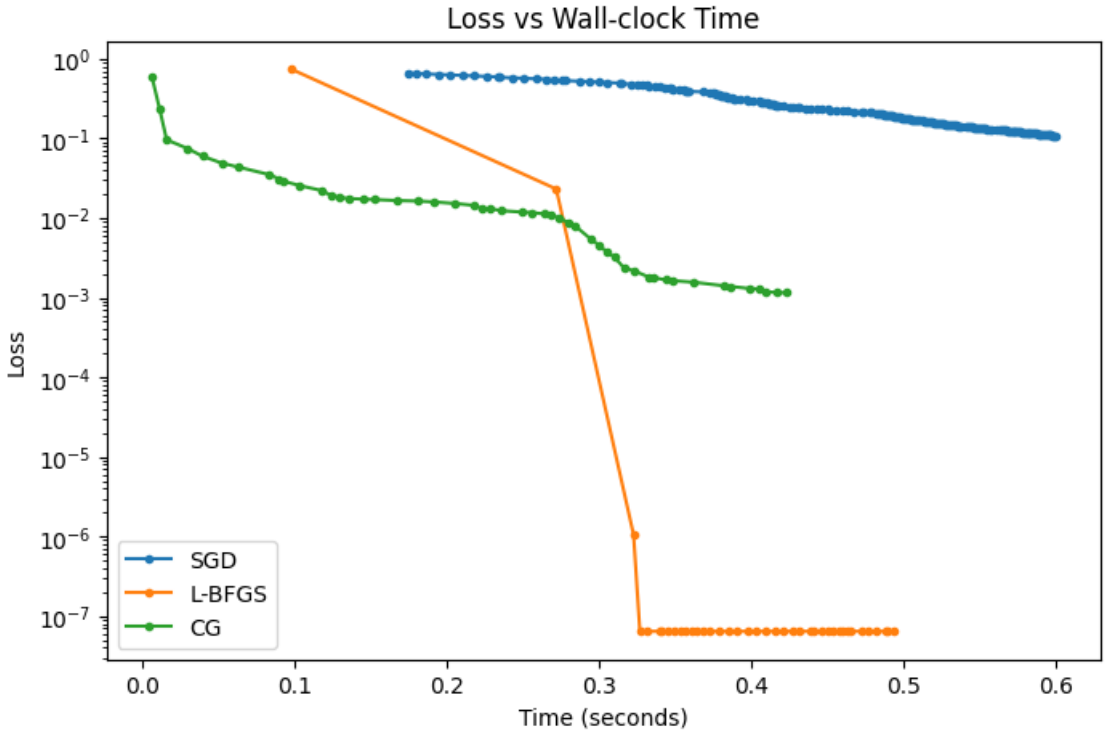
- دیتاست (**Breast Cancer Wisconsin**دودویی)
- شبکهٔ کوچک :(Shallow MLP) یک Hidden layer با ۵ نورون
- تعداد پارامترها :۱۱۵ ← مناسب برای روش‌های شبه‌نیوتونی

2-6- بهینه سازهای مقایسه شده

1. **SGD**
2. **L-BFGS** تقریب نیوتون
3. **Conjugate Gradient** با استفاده از Scipy

2-7- نتایج

در **مدل‌های کوچک**، استفاده از اطلاعات انحنا کاملاً سودمند است و حتی L-BFGS از نظر زمان واقعی از SGD جلو می‌زند .این حالت مشابه تئوری است و نشان می‌دهد مشکل اصلی روش‌های مرتبه دوم **هزینهٔ محاسباتی** آنهاست، نه کیفیت گام‌ها.



چرا در مدل کوچک، L-BFGSبهتر عمل می‌کند؟

در این ابعاد (۱۵۰ پارامتر)، حافظهٔ لازم برای ذخیرهٔ ماتریس تقریب هسین L-BFGS بسیار کم است. هر iteration آن *هزینه زیادی دارد* اما تعداد iterations آن کم است. به همین دلیل اگر loss سطحی و بدون نویز باشد (مثل دیتاست سرطان پستان)، L-BFGS معمولاً سریع‌تر از SGD به صفر می‌رسد.

رفتار CG

CG نسبت به L-BFGS کمی کندتر است، اما از SGD بسیار پایدارتر و بدون نوسان همگرا می‌شود.این روش نیازی به ذخیرهٔ هسین ندارد (فقط گرادیان). در روش CG مبتنی بر SciPy، به دلیل اینکه شبکهٔ عصبی غیرخطی با تابع زیان غیرکوادراتیک است و ماتریس هسین در طول آموزش تغییر می‌کند، الگوریتم CG کلاسیک همگرایی تضمین‌شده ندارد و معمولاً با گام‌های بسیار بزرگ و افزایش شدید گرادیان باعث overflow می‌شود. اما نسخهٔ Trust-Region CG (trust-ncg) یا استفاده از مدل پایدارسازی‌شده با Softplus و BCEWithLogitsLoss باعث رفع این ناپایداری می‌شود.

چرا SGD کندتر است؟

هر آپدیت **ارزان** است اما **تعداد آپدیت‌ها زیاد**. در دیتاست واقعی که کمی بدحالت (ill-conditioned) است، مسیر SGD زیگزاگی و کند می‌شود. با این حال در مدل‌های deep به دلیل ارزان بودن هر iteration از L-BFGS بهتر است.

الگوریتم	بازه زمانی کل (ثانیه)	سریع‌ترین؟
SGD	0.21~ثانیه	کندترین
L-BFGS	0.064~ثانیه	سریع
CG	0.162~ثانیه	متوسط

فصل چهارم
شبکه عمیق و تله ی مقیاس پذیری

شبکه عمیق و تله ی مقیاس پذیری

2-8- معماری:

شبکهٔ عمیق با:

- ۳ لایهٔ مخفی
- هر کدام ۱۰۰ نورون
- دیتاستFashion-MNIST

2-9- محاسبهٔ ابعاد مدل و هسین:

هر لایه شامل وزن‌ها و بایاس‌هاست. محاسبهٔ پارامترها گام‌به‌گام:

- لایهٔ 1 : وزن ها $784 \times 100 = 78\,400$بایاس‌ها 100 .

  مجموع لایه1. $78\,400 + 100 = 78\,500$

- لایهٔ 2 وزن ها $100 \times 100 = 10\,000$. بایاس‌ها 100.

  مجموع لایه 2. $10\,000 + 100 = 10\,100$

- لایهٔ 3 : مانند لایه 2 است

- لایهٔ خروجی: وزن ها $100 \times 10 = 1\,000$ بایاس ها 10.

  مجموع خروجی. $1\,000 + 10 = 1\,010$ =

- حالا جمع کل پارامترها N

$$N = 78\,500 + 10\,100 + 10\,100 + 1\,010 = 99\,710$$

حافظهٔ لازم برای ذخیرهٔ ماتریس هسین $N \times N$:

تعداد مؤلفه‌ها در هسین: $N^2 = 99{,}710^2$

محاسبه با روش جبری (کوتاه) . $99{,}710 = 100{,}000 - 290$:

پس $99{,}710^2 = 100{,}000^2 - 2 \cdot 100{,}000 \cdot 290 + 290^2$

$= 10{,}000{,}000{,}000 - 58{,}000{,}000 + 84{,}100 = 9{,}942{,}084{,}100$

هر عدد float چهار بایت است . پس کل بایت‌ها:

$$\text{bytes} = 9{,}942{,}084{,}100 \times 4 = 39{,}768{,}336{,}400 \text{ bytes}$$

تبدیل به گیگابایت:

بر حسب «گیبی بایت:($\text{GiB} = 2^{30} = 1{,}073{,}741{,}824$) »

$$\frac{39{,}768{,}336{,}400}{1{,}073{,}741{,}824} \approx 37.04 \text{ GiB}$$

بر حسب «گیگابایت ده دهی:($\text{GB} = 10^9$) »

$$\frac{39{,}768{,}336{,}400}{10^9} \approx 39.77 \text{ GB}$$

بنابراین ذخیرهٔ هسینِ کامل نیازمند حدود ۳۷ **GiB (≈ 40 GB)** حافظهٔ RAM است.

اگر فقط بخش مثلثیِ متقارنِ هسین را ذخیره کنیم (چون هسین متقارن است) حداقل نصف این مقدار لازم است **18.5 GiB** ≈ که باز هم بزرگ و غیرعملی است برای اکثر کارت‌ها/سرورها در عمل (و این فقط حافظهٔ نگهداری، بدون احتساب نیاز محاسباتی برای معکوس کردن است)

هزینهٔ محاسباتی معکوس/حل هسین : عملگر معکوس/حل هسین (مثلاً چولسکی یا معکوس مستقیم) در پیچیدگی تقریباً $O(N^3)$ عملیات عددی است. برای $N \approx 10^5$ داریم که $N^3 \approx 9.913 \times 10^{14}$ عمل که عددی عظیم (صدها تریلیون فلاپ)، به‌علاوه هزینهٔ حافظهٔ بالا و تبادل با حافظه. نتیجهٔ عملی: **نیوتون خالص** برای این مدل عمیق (N≈100k) غیرقابل‌اجرا است. همین‌طور، هسین در نقطه‌های مختلف آموزش تغییر می‌کند — پس حتی اگر یک‌بار هسین را محاسبه کنی، باید باز هم آن را به‌روزرسانی کنی تا روش مؤثر بماند.

نتیجه: روش نیوتون در شبکه‌های عمیق غیرقابل اجراست؛ نه به دلیل ضعف تئوریک، بلکه به دلیل هزینهٔ حافظه و زمان.

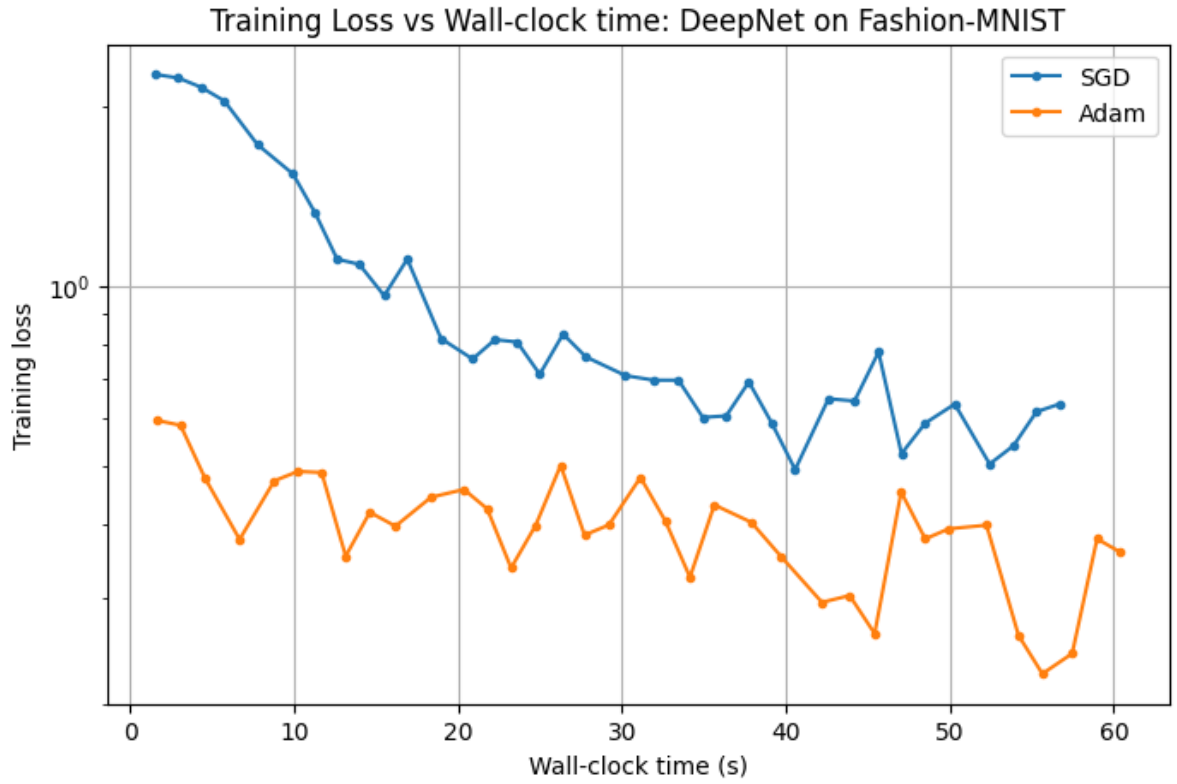
10-2-مقایسه SGD و Adam

مشاهدات:

- Adamدر ابتدا **بسیار سریع‌تر** پایین می‌آید.
- SGDرفتار لرزانی دارد.
- پس از چند Epoch ممکن است SGD برسد، اما Adam در زمان کمتر عملکرد بهتر دارد.

نتیجه:

استفاده از **اطلاعات آماری لحظه‌ای (moment estimates)** در Adam باعث می‌شود الگوریتم بدون هسین، «اثر شبه‌نیوتونی» داشته باشد.



.

فصل پنجم
تعامد و QR (رویکرد داده کاوی)

تعامد و QR (رویکرد داده کاوی)

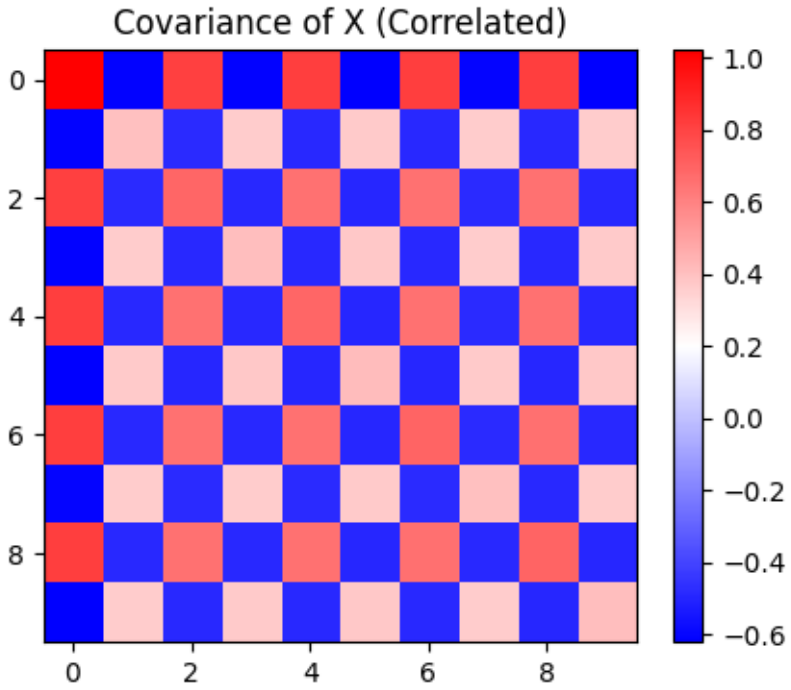
11-2-ساخت دیتاست همبسته

ویژگی‌ها به صورت خطی به یکدیگر وابسته طراحی شدند.(Multicollinearity)
برای دیتاست اولیه:

$$\kappa(\mathrm{Cov}(X)) \gg 1$$

این وضعیت باعث می‌شود سطح خطا در رگرسیون **دره‌ای و کشیده** شود و GD عملکرد ضعیف داشته باشد. در دیتاست داخل کد داریم:

Condition number of Cov(X) = 678.0130182609772

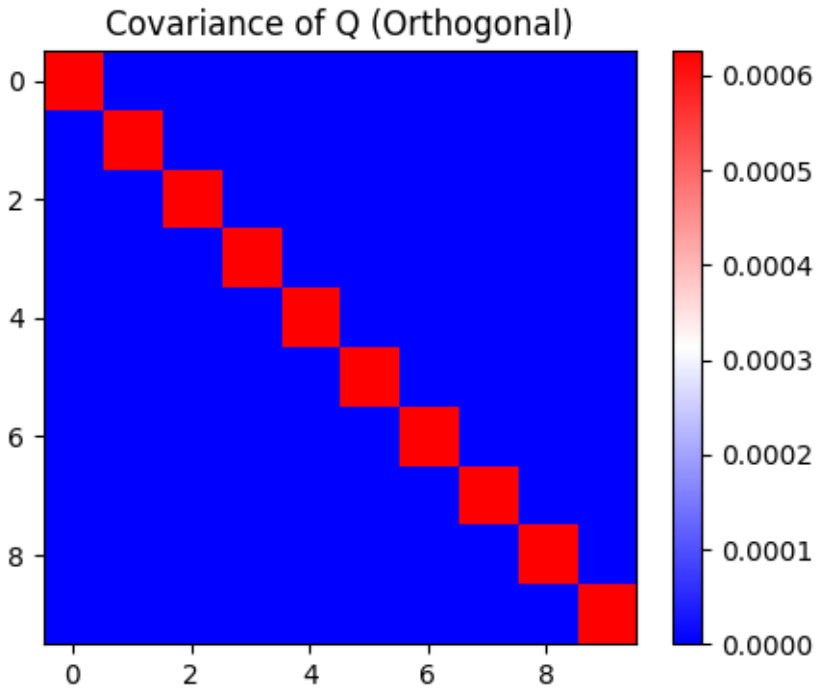


بعد از اجرای QR :

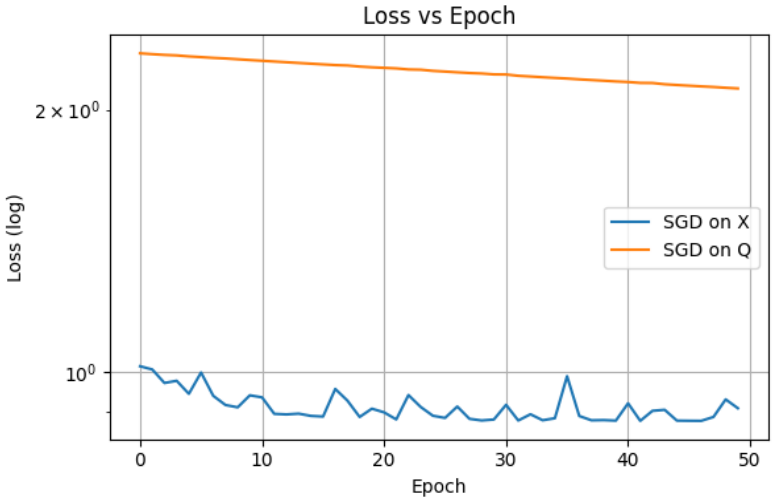
- ماتریس Q **ویژگی‌های متعامد** دارد
- کوواریانس Q تقریباً **قطری** می‌شود
- عدد **Condition Number** از مقدار بسیار بزرگ به نزدیک 1 کاهش می‌یابد
- در نتیجه سطح هزینه (Loss landscape) از کشیده به گرد (well-conditioned) میرود
- گرادیان کاهشی دیگر **زیگ‌زاگ نمی‌کند**

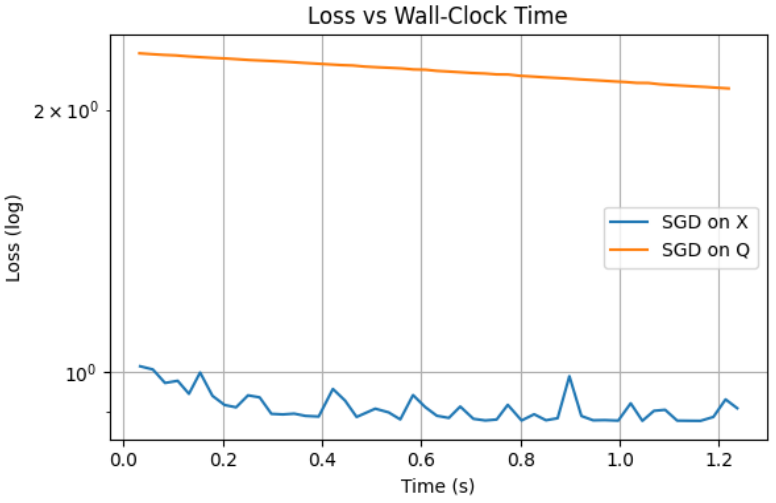
- همگرایی بسیار سریع تر و نرم تر می‌شود
- رفتار SGD روی Q شبیه یک **روش مرتبه دوم کارآمد** می‌شود
- سطح خطا تقریباً ایزوتروپیک شده است:

Condition number of Cov(Q) = 1.0018074320188883



اگرچه Loss اولیه روی دادهٔ Q به دلیل تغییر مقیاس ویژگی‌ها و کوچک بودن نُرم ستون‌ها کمی بزرگ‌تر است، اما مسیر کاهش Loss بسیار یکنواخت تر و سریع تر است. QR هندسهٔ سطح خطا را اصلاح می‌کند، نه مقدار Loss اولیه را. بنابراین معیار صحیح مقایسه، سرعت همگرایی و پایداری آن است نه مقدار اولیه Loss





فصل ششم
جمع‌بندي و نتيجه‌گيری

جمع‌بندي و نتيجه‌گيری

این پروژه به صورت تجربی چند واقعیت اساسی را در یادگیری عمیق نشان داد:

- **روش نیوتون از نظر تئوری عالی است**

ولی هزینهٔ محاسباتی $\mathcal{O}(N^2)$آن باعث می‌شود در شبکه‌های بزرگ عملاً غیرقابل استفاده باشد.

- **GD به شدت به عدد وضعیت وابسته است**

در توابع دره‌ای، GD کند و زیگ‌زاگی است؛ اما Newton و CG این مشکل را ندارند.

- **در شبکه‌های کوچک، روش‌های شبه‌نیوتونی واقعاً بهتر عمل می‌کنند**

L-BFGS در مدل Breast Cancer بسیار سریع‌تر از SGD بود.

- **در شبکه‌های بزرگ، Adamو SGD باقی می‌مانند**

Adamبا استفاده از مومنت‌ها اثر «پیش‌شرطی‌سازی» ایجاد می‌کند.

- **هندسهٔ داده‌ها همان‌قدر مهم است که انتخاب بهینه‌ساز**

تعامد (QR) عدد وضعیت را بهبود می‌دهد و بهینه‌سازی را به شدت تسریع می‌کند.

.

منابع و مراجع

[1] Nocedal & Wright — Numerical Optimization (2006 ،ویرایش دوم)

[2] (2004) Boyd & Vandenberghe — Convex Optimization

[3] (2016) Goodfellow, Bengio, Courville — Deep Learning

[4] (2022) Murphy — Probabilistic Machine Learning