تمرین دوم درس یادگیری ماشین
دکتر باباعلی

سیدعلیرضا مولوی

فهرست مطالب

۱ داده ها ۱

۲ مدل ۱

۳ مسئله بهینه سازی ۲
۱.۳ معادله نرمال ۲
۲.۳ استفاده از روش کاهش گرادیان ۳

۴ بررسی نتایج ۴

۱ داده ها

داده ها به دو بخش برای آموزش و برای تست تقسیم شده اند، تعداد داده های آموزشی ۸۰۰۰ و تعداد داده های تست ۲۰۰۰ است. داده ها ورودی شامل ۲ متغیر حقیقی اند و داده های خروجی ۱ متغیر حقیقی اند، بنابر این مسئله ما رگرسیون [۱] است.

۲ مدل

چون مسئله رگرسیون سیستم مسئله را به صورت معادلات جدول ۱ تعریف میکنیم:

جدول ۱: معادلات مدل رگرسیون

$$D = \{(x^{(i)}, y_i) | i = 1, 2, \cdots, n\} \subset R^d \times R \quad (۱)$$

$$\hat{y}_i = f(\phi(x^{(i)}); \omega, b) = \phi(x^{(i)}).\omega + b \quad (۲)$$

$$Loss(D) = L(D) + \lambda\Omega(\omega) \quad (۳)$$

$$L(D) = \frac{1}{n}\Sigma_{i=1}^{n} l(y_i, \hat{y}_i) \quad (۴)$$

$$l(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2 \quad (۵)$$

$$\Omega(\omega) = L2(\omega) = \frac{1}{2m}\Sigma_i \omega_i^2 \quad (۶)$$

توضیح عبارات تعریف شده در جدول ۱:

* D: داده های آموزشی اند.
* $\hat{y}_i$: خروجی مدل به ازای $x^{(i)}$ است. $f(.; \omega, b)$ در واقع مدل رگرسیون خطی است که وزن های آن w و بایاس b است. $\phi : R^d \to R^m$ [۲] تابعی است که داده های ما را از یک فضا به فضای دیگر می برد. بنابراین:

$$\phi(x) \in R^m, \omega \in R^{m*1}, b \in R^1$$

* $L(D)$: مجموع خطای بر روی تمام داده ها اند به علاوه مقدار متعادل سازی [۳] بر روی وزن ها (ω) است؛ که مقدار λ یک تعادل بین متعادل سازی وزن ها و کاهش خطا برروی داده ها به وجود می آورد. **توجه کنید** ما مقدار b را متعادل سازی نکردیم زیرا در عمل این کار باعث underfit می شود.
* $L(D)$: مجموع خطای بر روی تمام داده ها است.
* $l(y, \hat{y})$: خطای به ازای فقط یک داده آموزشی است.

  با توجه به اینکه ما خطای مربع را انتخاب کردیم خطای $L(D)$ معادل **میانگین خطای مربعات** [۴] است.
* $\Omega(\omega)$: جریمه به ازای وزن های بزرگ است که این نوع به نام L2 است.

---

[۱] Regression
[۲] Basis function
[۳] Regularizatio
[۴] Mean Squarred Error

۳ مسئله بهینه سازی

هدف ما یافتن ω^*, b^* است به طوریکه:

$$\omega^*, b^* = argmin_{\omega,b} Loss(D) \quad (۷)$$

که برای حل معادله ۷ مشتق $Loss(D)$ را براساس متغیر های ω, b میگیرم و معادل 0 قرار می دهیم.

$$\nabla_\omega Loss(D) = 0, \nabla_b Loss(D) = 0 \quad (۸)$$

۱.۳ معادله نرمال [۵]

یک راه حل معادله ۸ به طور تحلیلی است که در این حالت بهترین جواب ممکن بدست می آید.[۶]
برای سادگی کار ما عبارات زیر ۲ تعریف میکنیم.

جدول ۲: معادلات بر پایه θ

$$\theta = \begin{pmatrix} b \\ \omega \end{pmatrix} \quad (۹)$$

$$z^{(i)} = \begin{pmatrix} 1 \\ \phi(x^{(i)}) \end{pmatrix} \quad (۱۰)$$

$$\hat{y} = f(\phi(x^{(i)}); \omega, b) = \phi(x^{(i)})\omega + b \quad (۱۱)$$

$$= z^{(i)}\theta \quad (۱۲)$$

$$(۱۳)$$

طبق مشتق گیری فرمول θ^* به صورت معادله ۱۴ تعریف میشود. دقت کنید سطر اول ماتریس متعادل سازی تماما 0 است. زیرا مقدار b را ما متعادل سازی نکردیم.

$$\theta^* = \left(Z^T.Z + \lambda \begin{pmatrix} 0,0,0,\cdots,0 \\ 0,1,0,\cdots,0 \\ 0,0,1,\cdots,0 \\ \vdots,\vdots,\vdots,\vdots,\vdots \\ 0,0,0,\cdots,1 \end{pmatrix} \right) Z^T.y \quad (۱۴)$$

پس از پیاده سازی فرمول های بالا و نتیجه گرفتن اجرا در جداول ۳ و ۴ نوشته شده است.
زمانی که درجه 1 باشد مدل نمی تواند حتی به داده آموزشی فیت شود و در واقع underfitting رخ میدهد.
اما برای درجه 3 و 5 مدل به خوبی الگو را یاد گرفته است و حتی overfitting رخ نداده است.

---
[۵] Normal Equation
[۶] Global Minimum

جدول ۳: نتایج بر روی مسئله بدون استفاده از متعادل سازی

Degree	RMSE on train	RMSE on Test	SSE on Train	SSE on Test
1	1513.1821	2544.0873	18317759690.3680	12944759955.7720
3	$2.1033e-09$	$6.4512e-09$	$3.5391e-14$	$8.3235e-14$
5	$2.1033e-09$	$6.4512e-09$	$3.5399e-14$	$8.3235e-14$

جدول ۴: نتایج بر روی مسئله با استفاده از متعادل سازی که مقدار λ از طریق GridSearch بدست آمده است.

Degree	λ	RMSE on train	RMSE on Test	SSE on Train	SSE on Test
1	1	1513.1821	2544.0988	18317759695.5464	12944877370.5125
3	0	$2.1033e-09$	$6.4512e-09$	$3.5391e-14$	$8.3235e-14$
5	0	$2.1033e-09$	$6.4512e-09$	$3.5399e-14$	$8.3235e-14$

۲.۳ استفاده از روش کاهش گرادیان[۷]

روش کاهش گرادیان یک روش تکراری است و نیاز به ابرپارامتر[۸] اضافی نرخ یادگیری و تعداد تکرار ها را دارد. من برای محاسبه گرادیان از روش کل داده استفاده کردم[۹] (کد نوشته میتواند به عنوان ورودی سایز batch size را بگیرد و مدل رو بر روی mini batch آموزش دهد.).

از روابط زیر برای آموزش مدل استفاده میکنیم. روابط زیر بر پایه بردار هستند.

$$b = b + lr * \nabla_b Loss(D) \quad (۱۵)$$

$$\omega = \omega + lr * \nabla_\omega Loss(D) \quad (۱۶)$$

$$\nabla_b Loss(D) = \frac{1}{n}\Sigma_i(\hat{y}_i - y_i) \quad (۱۷)$$

$$\nabla_\omega Loss(D) = \frac{1}{n}\Sigma_{i=1}^{n}(\hat{y}_i - y_i)x^{(i)} + \lambda\frac{1}{m}\omega \quad (۱۸)$$

مقادیر اولیه ω را معمولا به طور تصادفی و کوچک انتخاب می شود، و مقدار b را معمولا صفر لحاظ می شود.

با توجه به اینکه در ابتدای آموزش خطاها خیلی بزرگ هستند و گرادیان نیز بسیار بزرگ میشود، برای اینکه پروسه آموزش به این دلیل غیرپایدار نشود ما اندازه بردار گرادیان رو محدود میکنیم.

جدول ۵: هایپرپارامتر های پیشفرض $clipnorm = 1 lambda = 0 lr = 0.01 ،epochs = 50,000$

Degree	RMSE on train	RMSE on Test	SSE on Train	SSE on Test
1	1868.6277	3207.1338	27934156107.4889	20571413877.7332
3	17.1598	22.8040	2355649.6984	1040042.1568
5	5085.5341	9350.3283	206901251028.7528	174857280461.1025

[۷] Gradient Decsent
[۸] HyperParameter
[۹] Batch Gradient

جدول ۶: نتایج بر روی مسئله با استفاده از متعادل سازی که مقدار λ از طریق GridSearch بدست آمده است. هایپرپارامتر های پیشفرض $epochs = 50,000$،$lr = 0.001$،$clipnorm = 1$، اما برای degree=5 ما به دنبال lr مناسب تر نیز هستیم.

Degree	λ	RMSE on train	RMSE on Test	SSE on Train	SSE on Test
1	1	1845.4281	3194.4256	27244839430.7771	20408709259.3440
3	0.1	16.8115	23.8095	2261011.0345	1133784.4048
5	$\lambda = 0$ $lr = 0.0001$	679.3653	2149.1433	3692297973.0992	9237633886.3776

۴ بررسی نتایج

کاهش گرادیان یک الگوریتم بهینه سازی محلی[۱۰] است، بنابراین در این روش احتمال اینکه در یک کمینه محلی به دام بیافتیم بالا است.

با توجه به نتایج نوشته در جدول ۴ و ۶ می توان مشاهده کرد حتی با وجود متعادل سازی مقادیر λ باز مدل کاهش گرادیان نتیجه ضعیف تری داشته است، و احتمالا در کمینه محلی گیر کرده است[۱۱]. به سطر سوم جدول ۶ توجه کنید، حتی با وجود متعادل سازی مدل SGD Degree=5 نتیجه چندان خوبی بر روی داده آموزشی نداشته و همچنین به شدت overfit کرده است؛ اما روش معادله نرمال هم بر روی داده آموزش و هم داده تست به خوبی فیت شده است.

به طور کلی زمانی که تعداد داده ها کم باشد استفاده از روش معادله نرمال بازدهی بهتری دارد ولی زمانی که داده ها زیاد باشد و بعد داده ها بالا باشد روش معادله نرمال بسیار کند می شود (حتی امکان این است که همه داده ها را **نتوان** در حافظه اصلی ذخیره کرد) و در این مواقع روش های Mini-Batch GD بازدهی و سرعت بهتری دارند.

۲ روش بهترین بازده را زمانی دارند که ظرفیت (توانایی یادگیری مدل)، که در این مثال با degree کنترل می شود، متناسب با داده باشد. زمانی که degree=3 انتخاب شود هر دو مدل بهترین بازده بر روی داده آموزشی و تست می دهند و اگر کمتر باشد مدل underfit و اگر بیشتر باشد احتمالا overfit می کند.

[۱۰] Local Seach

[۱۱] البته روش هایی مثل استفاده از الگوریتم های بهینه سازی بهینه تر مانند SGD with momentum و یا روش های با نرخ یادگیری انطباقی ویا Early Stopping و ... می توانند روش SGD بهبود بدهند.