به نام خدا

مفاهیم پیشرفته در یادگیری ماشین

نیم‌سال دوم ۴۰۱ـ۱۴۰۰

مدرس: دکتر مهدیه سلیمانی

دانشکده‌ی مهندسی کامپیوتر

تمرین سری دوم | روش های متاـ یادگیری مبتنی بر بهینه سازی و یادگیری متریک | زمان تحویل: ۱۹ فروردین

لطفا نکات زیر را رعایت کنید:

- سوالات خود را از طریق پست مربوط به تمرین در Quera مطرح کنید.
- در هر کدام از سوالات، اگر از منابع خاصی استفاده کرده‌اید باید آن را ذکر کنید.
- پاسخ ارسالی واضح و خوانا باشد.
- تمام پاسخ‌های خود را در یک فایل با فرمت HW#_[SID]_[Fullname].zip روی کوئرا قرار دهید.
- برای ارسال هر تمرین تا ساعت ۲۳:۵۹ روز ددلاین فرصت دارید. علاوه بر آن، در هر تمرین می توانید تا سقف هفت روز از تأخیر مجاز باقیمانده‌ی خود استفاده کنید.
- برای کسب نمره کامل در این تمرین کافیست تا ۱۶۰ نمره از ۲۰۰ نمره را کسب نمایید (۴۰ نمره امتیازی).

سوال ۱: (نظری) متاـ یادگیری مبتنی بر بهینه‌سازی دو سطحی (۲۵ نمره)

همانطور که در جلسات درس مشاهده کرده‌اید، یکی از روش‌های متاـ یادگیری [۱] ، خانواده بهینه‌سازی دو سطحی [۲] بوده که مهمترین کار در این زمینه روش MAML [۳] می‌باشد. در این خانواده از روش‌ها، متاـ پارامترها [۴] (پارامترهای آهسته) همبعد با پارامترهای مختص وظیفه [۵] (پارامترهای سریع) بوده و به عنوان یک نقطه شروع برای کل وزن‌های شبکه عمل می‌کنند. به طور دقیق‌تر، اگر توزیع وظایف [۶] را به صورت $p(\mathcal{T})$ در نظر بگیریم، می‌توان رابطه زیر را برای یادگیری متاـ پارامترها ارائه داد:

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{\mathcal{T}=(S,Q)\sim p(\mathcal{T})}[\mathcal{L}(\phi, Q)] \quad (۱آ)$$

$$Where \quad \phi = \mathcal{A}\text{lg}(\theta, S) \quad (۱ب)$$

که در این رابطه S مجموعه داده‌های پشتیبان [۷] و Q مجموعه داده‌های پرسمان [۸] مربوط به هر وظیفه [۹] را نشان می‌دهد. در این رابطه، محاسبه پارامترهای سریع ϕ توسط روش $\mathcal{A}\text{lg}$ انجام می‌شود که در مقالات مربوطه به طرق مختلفی انتخاب شده و بهینه‌سازی داخلی [۱۰] نامیده می‌شود. همچنین در رابطه فوق، بهینه‌سازی خارجی [۱۱] که روی پارامترهای θ صورت می‌پذیرد، به شکل $\arg\min$ نمایش داده شده است.
برای انجام بهینه‌سازی خارجی، لازم است تا از تابع معرفی شده نسبت به θ گرادیان را به صورت زیر محاسبه کرده و θ را از طریق آن به‌روزرسانی نماییم:

$$\nabla_\theta \mathbb{E}[\mathcal{L}(\phi, Q)] = \mathbb{E}[\nabla_\theta \mathcal{L}(\mathcal{A}\text{lg}(\theta, S), Q)] \quad (۲)$$

برای محاسبه عبارت فوق، از قاعده زنجیره‌ای مشتق استفاده می‌کنیم:

$$\nabla_\theta \mathcal{L}(\mathcal{A}\text{lg}(\theta, S), Q) = \nabla_\phi \mathcal{L}(\phi, Q)|_{\phi=\mathcal{A}\text{lg}(\theta,S)} \times \frac{d}{d\theta}\mathcal{A}\text{lg}(\theta, S) \quad (۳)$$

همانطور که مشاهده می‌کنید، رابطه فوق از دو جمله تشکیل شده است؛ محاسبه جمله اول نسبتاً راحت است. چرا که کافیست تا از $\mathcal{L}$ مشتق گرفته و مقدار $\mathcal{A}\text{lg}(\theta, S)$ را در آن جایگذاری کنیم و در این صورت گرادیانی از $\mathcal{A}\text{lg}(\theta, S)$ عبور نمی‌کند. این در حالیست که برای محاسبه جمله دوم،

[1] Meta-Learning
[2] Bi-Level Optimization
[3] Model Agnostic Meta-Learning
[4] Meta-Parameters
[5] Task-Specific Parameters
[6] Tasks
[7] Support
[8] Query
[9] Task
[10] Inner-Level Optimization
[11] Outer-Level Optimization

لازم است تا عملیات مشتق‌گیری را از داخل الگوریتم $\mathcal{A}\text{lg}(\theta, S)$ عبور دهیم. این مسئله می‌تواند مشکل‌زا باشد چرا که ممکن است $\mathcal{A}\text{lg}(\theta, S)$ اصلاً قابلیت عبور گرادیان را نداشته باشد یا اگر دارد ممکن است منجر به محاسبات پرهزینه شود. به عنوان مثال، در روش MAML داریم:

$$\mathcal{A}\text{lg}(\theta, S) = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, Q) \Rightarrow \frac{d}{d\theta}\mathcal{A}\text{lg}(\theta, S) = I - \alpha \nabla^2_\theta \mathcal{L}(\theta, S) \tag{۴}$$

که در رابطه فوق، محاسبه مشتق‌های مرتبه دوم (Hessian) می‌تواند بسیار سخت یا پرهزینه باشد چرا که لازم است تا کل گراف محاسباتی در طول مسیر محاسبه $\mathcal{A}\text{lg}(\theta, S)$ نگه داشته شود تا بتوان گرادیان را از روی آن عبور داده و به مراحل قبلی رساند.
در این سوال قصد داریم تا تکنیکی معرفی کنیم که این مشکلات مرتفع شوند. برای این منظور، فرض کنید $\mathcal{A}\text{lg}(\theta, S)$ به صورت زیر پیشنهاد شده است:

$$\phi = \mathcal{A}\text{lg}(\theta, S) = \arg\min_{\phi'} \mathcal{L}(\phi', S) + \frac{\lambda}{2}||\phi' - \theta||^2 \tag{۵}$$

که λ یک هایپرپارامتر است و هر چه مقدار آن بیشتر باشد، باعث می‌شود تا جواب بهینه‌سازی درونی، به نقطه شروع خود یعنی θ نزدیک‌تر بماند. بهینه‌سازی ۵ را می‌توان با انجام چندین گام بهینه‌سازی تکرار شونده [۱۲] حل کرد اما مشکل آن جاست که امکان عبور گرادیان از چنین محاسباتی وجود ندارد. با در نظر گرفتن این مسئله، به پرسش‌های زیر پاسخ دهید:

(آ) فرض کنید ما قادر هستیم تا بهینه‌سازی ۵ را به صورت کامل حل کنیم و ϕ را به عنوان جواب بهینه دقیق آن به دست آورده‌ایم. از این مسئله استفاده کنید و $\frac{d\phi}{d\theta}$ را محاسبه کنید (راهنمایی: در نقطه بهینه دقیق، مشتق $\mathcal{L}(\phi', S) + \frac{\lambda}{2}||\phi' - \theta||^2$ نسبت به ϕ' صفر خواهد بود). (۱۰ نمره)

(ب) اگر مراحل پرسش قبل را به درستی طی کرده باشید، در جواب خود به یک عبارت حاوی مشتق مرتبه دوم $\nabla^2 \mathcal{L}$ (یا همان ماتریس Hessian) می‌رسید. محاسبه این عبارت چه تفاوتی با مشتق مرتبه دوم موجود در رابطه ۴ دارد؟ استفاده از این تابع $\mathcal{A}\text{lg}$ پیشنهادی چه مزیتی نسبت به MAML دارد؟ (۱۰ نمره)

(ج) به عنوان جمع‌بندی، الگوریتمی که متاـ یادگیر در هر اپیزود طی می‌کند را به صورت گام گام شرح دهید. (۵ نمره)

پاسخ

(آ)

$$\frac{d}{d\phi'}\mathcal{L}(\phi', S) + \frac{\lambda}{2}||\phi' - \theta||^2 = \nabla_{\phi'}\mathcal{L}(\phi', S)|_{\phi'=\phi} - \lambda(\phi' - \theta)|_{\phi'=\phi} = 0 \tag{آ۶}$$

$$\Rightarrow \phi = \theta - \frac{1}{\lambda}\nabla_\phi \mathcal{L}(\phi, S) \tag{ب۶}$$

$$\Rightarrow \frac{d}{d\theta}\phi = I - \frac{1}{\lambda}\nabla^2_\phi \mathcal{L}(\phi, S) \times \frac{d\phi}{d\theta} \tag{ج۶}$$

$$\Rightarrow \frac{d\phi}{d\theta} = (I + \frac{1}{\lambda}\nabla^2_\phi \mathcal{L}(\phi, S))^{-1} \tag{د۶}$$

(ب) مشتق مرتبه دوم به دست آمده در الگوریتم MAML به صورت $\nabla^2_\theta \mathcal{L}(\theta, S)$ می‌باشد این در حالیست که در روش معرفی شده با محاسبه $\nabla^2_\phi \mathcal{L}(\phi, S)$ مواجه می‌شویم. این بدان معناست که مشتق دوم مورد نظر، کافیست در نقطه بهینه حاصل از حل دقیق بهینه‌سازی ۵ نوشته شود و به مسیری که در گام‌های محاسبه ۵ طی شده است، هیچ وابستگی وجود ندارد. توجه کنید که θ همان نقطه شروع بهینه‌سازی است در حالی که ϕ نقطه نهایی بهینه‌سازی محسوب می‌شود. لذا در حل ۵ کافیست تا گرادیان θ و ϕ را قطع کنیم، بهینه‌سازی ۵ را حل کرده و گرادیان مورد نیاز برای آپدیت θ را با کمک روابط ۳ و ۶د به دست آوریم. این درحالیست که در رویکرد MAML (مخصوصاً زمانی که بیش از یک گام بهینه‌سازی در حلقه داخلی بر می‌داریم)، لازم است تا کل گراف محاسباتی و کل وزن‌های محاسبه شده در میان مسیر را نگه داری کنیم تا بتوانیم مشتق‌های مرتبه دو را محاسبه کنیم.

(ج)
- یک وظیفه (Task) به صورت $\mathcal{T} \in D_{meta-train}$ نمونه برداری می‌کنیم که شامل داده‌های پشتیبان (Support) S و پرسمان (Query) Q می‌باشد.
- با در دست داشتن داده‌های S، بهینه‌سازی ۵ را با روش‌های مرسوم بهینه‌سازی انجام می‌دهیم و حاصل را ϕ می‌نامیم.
- طبق رابطه ۶د مشتق ϕ نسبت به θ را محاسبه می‌کنیم.
- با کمک داده‌های Q، عبارت $\nabla_\phi \mathcal{L}(\phi, Q)|_{\phi=\mathcal{A}\text{lg}(\theta,S)}$ را محاسبه می‌کنیم.
- از ضرب دو رابطه محاسبه شده در دو مورد اخیر، $\nabla_\theta \mathcal{L}(\mathcal{A}\text{lg}(\theta, S), Q)$ را محاسبه کرده و با کمک این گرادیان متاـ پارامترهای θ را آپدیت می‌کنیم.

[12] Iterative

سوال ۲: (نظری) روش‌های مبتنی بر یادگیری متریک (۵۵ نمره)

روش‌های یادگیری متریک یکی دیگر از روش‌هایی هستند که در درس به عنوان یکی از اعضای خانواده متا_یادگیری با آن‌ها آشنا شدید. در این روش‌ها هدف آن است که یک شبکه استخراج ویژگی مثل $f_\theta(x)$ یاد گرفته شود تا داده‌ها با برچسب یکسان را در فضای نمایش مخفی در کنار یکدیگر تصویر کند. پارامترهای θ متا_پارامترهای مدل در نظر گرفته شده و در طول متا_یادگیری آموزش داده می‌شوند. از آنجایی که $f_\theta(x)$ صرفاً یک فضای نمایشی [۱۳] فراهم می‌کند، برای کامل کردن شبکه نیاز به یک دسته‌بند (یادگیر پایه [۱۴]) پارامتریک یا نان_پارامتریک داریم که روی این فضای نمایشی قرار گرفته، از داده‌های پشتیبان برای آماده‌سازی خود استفاده کرده و با کمک آن‌ها عمل دسته‌بندی نمونه‌های پرسمان را انجام دهد. پارامترهای دسته‌بند معرفی شده را به عنوان پارامترهای سریع می‌شناسند و آن‌ها را با ϕ نمایش می‌دهند. این پارامترها مختص هر وظیفه به دست آمده و برای وظیفه بعدی تغییر می‌کنند.
در این روش‌ها یکی از تصمیمات مهم در زمینه طراحی الگوریتم انتخاب مناسب همین دسته‌بند می‌باشد. از جمله انتخاب‌های موجود برای این خانواده، انتخاب روش Nearest Neighbour می‌باشد. همچنین روش دیگری که در مقاله ProtoNet معرفی شد استفاده از دسته‌بندهای مبتنی بر پروتوتایپ دسته‌ها (با میانگین گیری از داده‌های موجود در مجموعه پشتیبان از هر کلاس) می‌باشد. مشکلی که متا_یادگیری مبتنی بر پروتوتایپ دارند این است که دسته‌بند ساده‌ای دارند و تعمیم‌پذیری دسته‌بند در فاز متا_ارزیابی [۱۵] کم می‌باشد. به همین دلیل در مقاله Bertinetto 2018 به بررسی دو دسته‌بند رایج در یادگیری ماشین (Logistic Regression, Ridge Regression) و نحوه استفاده موثر آن در مسئله متا_یادگیری پرداخته شده است. با مطالعه مقاله و راهنمایی های داده شده در زیر، به سوالات پاسخ دهید:

(آ) توضیح دهید که در نگاه اول، استفاده از این دسته‌بندها چه مشکلی می‌تواند برای فرایند متا_آموزش [۱۶] ایجاد کند؟ چرا استفاده از رویکردهای پروتوتایپی یا KNN این مشکل را ایجاد نمی‌کند؟ (راهنمایی: پاسخ این مورد غیر مرتبط با پرسش قبل نیست) (۱۰ نمره)

(ب) توضیح دهید که در مقاله معرفی شده، چگونه مشکل معرفی شده را حل می‌کند؟ تفاوت رویکردی که برای Ridge Regression و Logistic Regression به کار گرفته می‌شود را توضیح دهید. (۵ نمره)

(ج) دو ماتریس $X \in \mathbb{R}^{n\times d}$ به عنوان داده‌های پشتیبان تعبیه شده در فضای نمایش و $Y \in \mathbb{R}^{n\times l}$ به عنوان برچسب‌های One-hot همان داده‌ها را در نظر بگیرید که n تعداد داده‌ها و d اندازه بردار بازنمایی هر داده و l تعداد برچسب‌های موجود در دسته می‌باشد. ثابت کنید که دو رابطه زیر در صورتی که $\lambda > 0$ باشد، با هم برابر می‌باشند و نحوه استفاده آن در دسته‌بند Ridge Regression را توضیح دهید : (۱۰ نمره)

$$\left(X^TX + \lambda I\right)^{-1}X^TY = X^T\left(XX^T + \lambda I\right)^{-1}Y$$

(د) توضیح دهید که برای انجام متا_یادگیری استفاده از کدام یک از دو رابطه بالا بهتر می‌باشد و چرا؟ (۵ نمره)

(ه) در مقاله معرفی شده، برای محاسبه وزن‌های دسته‌بند Logistic Regression از بهینه‌سازی با روش Newton استفاده شده است. دلیل این امر را بیان کنید و همچنین در مورد خود Newton's Method تحقیق کنید و رابطه به روزرسانی و نحوه بدست آوردن این رابطه را بنویسید. (۱۰ نمره)

(و) تعاریف زیر را در نظر بگیرید:

$$\begin{aligned} A_t &= diag(q_t) \\ q_t^{(i)} &= \sigma\left(\omega_t^T x^{(i)}\right)\left(1-\sigma\left(\omega_t^T x^{(i)}\right)\right) \\ B_t^{(i)} &= \sigma\left(\omega_t^T x^{(i)}\right) - y^{(i)} \end{aligned}$$

که $A \in \mathbb{R}^{n\times n}$ یک ماتریس قطری و $B \in \mathbb{R}^n$ می‌باشد.
با اعمال Newton's Method روی تابع هزینه این دسته‌بند، به رابطه به‌روزرسانی زیر برسید (۱۵ نمره) :

$$\omega_{t+1} = \left(X^TA_tX + \lambda I\right)^{-1}X^T\left(A_tX\omega_t - B_t\right)$$

(رابطه فوق مشابه رابطه ۷ مقاله می‌باشد با این تفاوت که به نظر می‌رسد روابط موجود در مقاله در برخی جزئیات از نظر Notation ایراد دارند.)

پاسخ:

(آ) دو دسته بند ذکر شده، دسته‌بندهایی پارامتریک هستند و برای آن که پارامترهای آن‌ها محاسبه شود، نیاز است تا یک دستگاه بهینه‌سازی حل شود. یکی از روش‌های حل دستگاه، استفاده از رویکردهای iterative و استفاده مستقیم از رویکرد Gradient Decent است. بدی استفاده از این نوع Solver آن است که مراحل انجام شده در رویکرد iterative قابلیت عبور گرادیان را ندارند و این باعث به مشکل خوردن متا_یادگیری می‌شود. به عبارت دقیق‌تر، فرض کنید که نمونه‌های پشتیبان را از شبکه استخراج ویژگی عبور داده‌اید و آن‌ها را در فضای

[13] Representation Space
[14] Base-Learner
[15] Meta-Test
[16] Meta-Training

نمایشی تعبیه کرده‌اید. سپس با یک رویکرد iterative بهینه‌سازی مربوط به Ridge Regression را انجام داده و وزن‌های دسته‌بند را به دست می‌آورید. مشکلی که وجود دارد آن است که نمی‌توان گرادیان Meta-Loss را از وزن‌های به دست آمده عبور داد و از طریق آن شبکه استخراج ویژگی و پارامترهای θ را آپدیت کرد. این در حالیست که با استفاده از KNN، به راحتی می‌توان وزن‌های θ را آپدیت کرد چرا که دسته‌بند مستقیماً با خود نمونه‌های هر کلاس کار می‌کند. با استفاده از این رویکرد می‌توان گرادیان را از نمونه‌های موجود هر کلاس عبور داد و از این طریق چینش نمونه‌های یک کلاس را طوری تنظیم کرد که نمونه‌های یک کلاس در فضای نمایش نزدیک یک دیگر بیفتند. رویکرد مشابهی در مورد دسته‌بند پروتوتایپی وجود دارد چرا که عملیات میانگین‌گیری روی نمونه‌های یک کلاس مشکلی برای عبور مسیر گرادیان ایجاد نمی‌کند و با آپدیت کردن θ می‌توان نمونه‌ها را طوری در فضا تعبیه کرد که نمونه‌های مربوط به یک کلاس خاص در نزدیکی پروتوتایپ کلاس خودشان قرار بگیرند.

(ب) این مقاله برای حل مشکل Ridge Regression ، به جای حل iterative آن، یک جواب Closed-form به عنوان وزن‌های بهینه معرفی می‌کند. این جواب بسته به راحتی قابلیت عبور گرادیان به لایه‌های عقبی را فراهم می‌کند.

از سویی برای دسته‌بند Logistic Regression ، از رویکرد iterative خاصی استفاده می‌کند و همزمان نشان می‌دهد که این رویکرد قابلیت عبور گرادیان از تک تک مراحل iterative را فراهم می‌کند. رویکرد مورد استفاده Newton's Method نام دارد.

(ج) باتوجه به اینکه ماتریس Y در هردو طرف مساوی از سمت راست در معادله ضرب شده است، پس تنها کافی است که اثبات کنیم:

$$\left(X^TX+\lambda I\right)^{-1}X^T=X^T\left(XX^T+\lambda I\right)^{-1}$$

برای شروع اثبات در رابطه زیر را در نظر بگیرید:

$$\lambda X^T=\lambda X^T$$

ماتریس همانی را برای یک سمت از رابطه مساوی بالا از سمت چپ ماتریس X^T و یک بار از سمت راست ماتریس، ضرب می‌کنیم:

$$\lambda I_dX^T=\lambda X^TI_n$$

حال عبارت X^TXX^T را به دو سمت مساوی اضافه می‌کنیم.

$$X^TXX^T+\lambda I_dX^T=X^TXX^T+\lambda X^TI_n$$

با فاکتورگیری داریم:

$$\left(X^TX+\lambda I_d\right)X^T=X^T\left(XX^T+\lambda I_n\right)$$

حال اگر در دو طرف تساوی، عبارت $\left(X^TX+\lambda I_d\right)^{-1}$ را از چپ و عبارت $\left(XX^T+\lambda I_n\right)^{-1}$ را از سمت راست ضرب کنیم داریم:

$$\left(X^TX+\lambda I_d\right)^{-1}\left(X^TX+\lambda I_d\right)X^T\left(XX^T+\lambda I_n\right)^{-1}=\left(X^TX+\lambda I_d\right)^{-1}X^T\left(XX^T+\lambda I_n\right)\left(XX^T+\lambda I_n\right)^{-1}$$

با ساده‌سازی داریم:

$$X^T\left(XX^T+\lambda I_n\right)^{-1}=\left(X^TX+\lambda I_d\right)^{-1}X^T$$

که مطلوب سوال می‌باشد.

این نکته لازم به ذکر می‌باشد که ماتریس‌هایی که معکوس آن را در اثبات بالا استفاده کردیم هردو ماتریس‌های مثبت نیمه‌معین می‌باشند و بنابراین حتما معکوس‌پذیر می‌باشند.

رابطه معرفی شده، همان جواب Closed-Form برای دسته‌بند Ridge Regression است و همانطور که مشاهده می‌شود، در محاسبه این وزن‌ها تنها از ضرب ماتریسی و معکوس گیری استفاده شده است که این اپراتورها قابلیت عبور گرادیان از خود را فراهم می‌کنند.

(د) برای انجام متا_یادگیری استفاده از رابطه $X^T\left(XX^T+\lambda I_n\right)^{-1}$ بهینه‌تر می‌باشد. چون در این رابطه نیاز به محاسبه معکوس یک ماتریس با ابعاد $n\times n$ داریم ولی در حالت دیگر نیاز به محاسبه معکوس ماتریسی به ابعاد $d\times d$ می‌باشد که مقدار d که بیانگر ابعاد بازنمایی در شبکه عصبی می‌باشد که به مراتب خیلی بزرگتر از تعداد نمونه‌ها در مسئله‌های متا_یادگیری می‌باشد چرا که که تعداد نمونه‌های پشتیبان در مسائل few-shot خیلی کم می‌باشد.

(ه) به دلیل محدودیت تعداد به‌روزرسانی پارامترها در حلقه درونی الگوریتم،از Newton's Method برای همگرایی سریع‌تر به نسبت گرادیان گیری ساده استفاده شده است. در Newton's Method علاوه بر گرادیان، اطلاعات مشتق دوم نیز استفاده می‌شود. توجه کنید که در تابع هزینه‌های محدب (از جمله همین Logistic Regression) استفاده از این رویکرد می‌تواند خیلی سریع ما را به نقطه بهینه همگرا کند. در این روش با نوشتن بسط Taylor برای تابع هزینه f حول یک نقطه داریم:

$$f\left(x+t\right)=f\left(x\right)+f'\left(x\right)t+\frac{f''\left(x\right)t^2}{2}$$

در رابطه بالا برای کمینه کردن مقدار تابع، نسبت به پارامتر t که جهت به‌روزرسانی و حرکت می‌باشد مشتق میگیریم که جهت بهینه به‌روزرسانی را پیدا کنیم:

$$\frac{dy}{dx}\left(f(x)+f'(x)\,t+\frac{f''(x)\,t^2}{2}\right)=f'(x)+f''(x)\,t=0$$

$$t^*=-\frac{f'(x)}{f''(x)}$$

بنابراین رابطه بهینه‌سازی را می‌توان به صورت زیر نوشت:

$$x_{new}=x_{old}-\frac{f'(x_{old})}{f''(x_{old})}=x_{old}-(\nabla_x^2 f)^{-1}\nabla_x f$$

(و) هدف سوال کمینه‌کردن منفی لگاریتم بیشینه درست‌نمایی به همراه یک جمله منظم‌سازی می‌باشد.

$$P(Y|\omega,X)=\prod_{i=1}^{N}\left(\sigma\left(\omega^T x^{(i)}\right)\right)^{y^{(i)}}\left(1-\sigma\left(\omega^T x^{(i)}\right)\right)^{1-y^{(i)}}$$

$$L=-\log P(Y|\omega,X)+\lambda\|\omega\|^2=-\sum_{i=1}^{N}\left[y^{(i)}\log\left(\sigma\left(\omega^T x^{(i)}\right)\right)+\left(1-y^{(i)}\right)\log\left(1-\sigma\left(\omega^T x^{(i)}\right)\right)\right]+\lambda\|\omega\|^2$$

حال با گرادیان گیری مقادیر گرادیان اول و دوم را حساب می‌کنیم:

$$\nabla_\omega L=\sum_{i=1}^{N}\left(\sigma\left(\omega^T x^{(i)}\right)-y^{(i)}\right)x^{(i)^T}$$

$$H=\nabla_\omega^2 L=\sum_{i=1}^{N}\sigma\left(\omega^T x^{(i)}\right)\left(1-\sigma\left(\omega^T x^{(i)}\right)\right)x^{(i)^T}x^{(i)}$$

حال اگر سیگماهای بالا را به ضرب ماتریسی تبدیل کنیم، طبق رابطه به‌روزرسانی داریم:

$$\begin{aligned}\omega_{t+1}&=\omega_t-H^{-1}\nabla L\\&=\omega_t-\left(X^TA_tX+\lambda I\right)^{-1}\left(X^TB_t+\lambda\omega_t\right)\\&=\left(X^TA_tX+\lambda I\right)^{-1}\left(\left(X^TA_tX+\lambda I\right)\omega_t-X^TB_t-\lambda\omega_t\right)\\&=\left(X^TA_tX+\lambda I\right)^{-1}\left(X^TA_tX\omega_t-X^TB_t\right)\\&\Rightarrow\omega_{t+1}=\left(X^TA_tX+\lambda I\right)^{-1}X^T\left(A_tX\omega_t-B_t\right)\end{aligned}$$

سوال ۳: (نظری) استفاده از دسته‌بند SVM در رویکرد یادگیری متریک (۳۰ نمره)

در این سوال قصد داریم مقاله MetaOptNet را مورد بررسی قرار دهیم که برای مدت قابل توجهی SOTA [۱۷] در زمینه یادگیری با تعداد نمونه کم [۱۸] به شمار می‌رفت. این مقاله شباهت بسیار زیادی به مقاله معرفی شده در پرسش قبل دارد با این تفاوت که قصد دارد به جای دسته‌بندهای معرفی شده، دسته‌بند SVM را به عنوان لایه آخر روی شبکه استخراج ویژگی سوار کند. رویکردهایی که در پرسش قبل برای Ridge Regression و Logistic Regression معرفی شدند را نمی‌توان برای SVM به کار برد. لذا این مقاله به دنبال ارائه روشی است تا این مشکل را برطرف نماید. برای این منظور، رویکردی شبیه به پرسش اول را دنبال می‌کند. با این تفاوت که در پرسش اول کل وزن‌های شبکه (اعم از دسته‌بند و Backbone) در دستگاه بهینه‌سازی قرار می‌گرفتند اما در این سوال دستگاه بهینه‌سازی فقط روی وزن‌های دسته‌بند SVM نوشته می‌شود. برای درک بهتر روش، مقاله مورد نظر را بررسی نموده و به پرسش‌های زیر پاسخ دهید:

(آ) توضیح دهید که پارامترهای سریع چگونه به کمک داده‌های پشتیبان و پارامترهای آهسته ساخته می‌شوند و دستگاه بهینه‌سازی معرفی شده در این مقاله که از حل آن پارامترهای سریع ساخته می‌شوند را به همراه دوگان آن به صورت دقیق و با ذکر جزئیات نمادگذاری معرفی کنید. (۵ نمره)

[17] State-of-the-Art
[18] Few-Shot Learning

(ب) با مطالعه صفحه ۴ این مقاله، توضیح دهید که استفاده از قضایای KKT و Implicit Function Theorem چه کمکی در راستای محاسبه گرادیان می‌کنند و آپدیت شبکه Backbone با استفاده از چه گرادیانی انجام می‌شود؟ (برای این قسمت اثبات دقیق ریاضی مد نظر نیست و به شرطی که به صورت شفاف کاربرد این دو قضیه و نحوه استفاده آن‌ها را بیان کنید نمره کامل را دریافت می‌کنید) (۱۰ نمره)

(ج) با مطالعه روابط این مقاله ضمن نوشتن دستگاه بهینه‌سازی مربوطه، توضیح دهید که چگونه می‌توان دسته‌بند Ridge Regression مطرح شده در سوال قبل را ذیل همین رویکرد جای داد. (۵ نمره)

(د) بعد از به دست آوردن وزن‌های بهینه w برای دسته‌بند SVM یا Ridge Regression ، تابع متا_هزینه به چه صورتی نوشته می‌شود؟ از این تابع هزینه برای به‌روزرسانی کدام پارامترها استفاده می‌شود؟ (از رابطه ۱۲ مقاله کمک بگیرید، ولی جزئیات به دست آوردن آن را به صورت شفاف بیان کنید) (۱۰ نمره)

توجه: در ادبیات این مقاله، نمادگذاری رایج متا_یادگیری رعایت نشده است و جای نماد θ و ϕ با هم عوض شده است. برای یکسان شدن جواب‌ها، شما از نمادگذاری معرفی شده در پرسش اول استفاده کرده و θ و ϕ را به ترتیب برای متا_پارامترها و پارامترهای مختص وظیفه به کار ببرید.

پاسخ

(آ) در این مقاله، برای به دست آوردن پارامترهای سریع، پیشنهاد شده است تا یک دستگاه بهینه‌سازی مطابق با تابع‌هزینه SVM حل شود. در مقاله مورد نظر، رابطه (۴) این دستگاه را نشان می‌دهد که در آن w ها پارامترهای سریع هستند (لازم است این دستگاه نوشته شود و متغیرهای موجود در آن معرفی شوند). در این رویکرد، پارامترهای آهسته همان وزن‌های شبکه استخراج ویژگی هستند که در این مقاله، بر خلاف نمادگذاری رایج، با ϕ نشان داده شده است. به عبارت دیگر، شبکه استخراج ویژگی یک فضای نمایشی ایجاد می‌کند که دسته‌بند SVM بتواند روی آن قرار گرفته و کلاس‌ها را دسته‌بندی نماید.

(ب) روابط (۴) و (۵) مقاله، دستگاه‌های بهینه‌سازی را نشان می‌دهد که مربوط به بهینه‌سازی Inner Loop می‌باشد. حل این دستگاه‌ها با رویکردهای Iterative شدنی است اما مشکلی که به وجود می‌آید آن است که نمی‌توان به سادگی گرادیان را از مراحل Iterative این دستگاه‌ها به عقب برگرداند تا متا_پارامترها آپدیت شوند.

برای این منظور، به جای آن که گرادیان را از مراحل میانی محاسباتی به صورت عقب گرد عبور دهند، سعی می‌شود تا گرادیان را به صورت مستقیم محاسبه کنند. (پاسخ این سوال شباهت زیادی به پرسش اول دارد) به عبارت دیگر، فرض کنید دستگاه‌های ذکر شده حل شده‌اند و پارامترهای سریع نهایی به دست‌آمده‌اند. در این صورت، طبق قضیه KKT می‌توان ادعا کرد که مشتق تابع هزینه دستگاه در نقطه بهینه صفر است. از سوی دیگر، اگر از قضیه مشتق ضمنی استفاده شود، می‌توان گرادیان پارامترهای سریع نسبت به متغیرهای نهان فضای نمایش و سپس نسبت به متا_پارامترهای مدل را محاسبه کرد. (رابطه (۸) مقاله)

(ج) اگر Ridge Regression را به صورت یک دستگاه بهینه‌سازی بنویسیم، می‌توان حتی علی‌رغم حل Iterative آن، از رویکرد معرفی شده در این سوال استفاده کرد و محاسبه گرادیان‌ها را ممکن ساخت. (رجوع شود به رابطه ۱۱ مقاله).

(د) با فرض این که پارامترهای سریع w در مراحل قبلی محاسبه شده‌اند، از تابع هزینه log-likelihood در کنار پارامتر قابل یادگیری γ به عنوان Temperature استفاده می‌شود:

$$\mathcal{L}(D^{test}, \theta, \phi, \lambda) = -\sum_{(x,y)\in D^{test}} logp(Y=y|x) = -\sum_{(x,y)log\in D^{test}} softmax(\gamma w f_\theta(x))$$

$$= -\sum_{(x,y)\in D^{test}} log\frac{exp(\gamma w_y f_\theta(x))}{\sum_k exp(\gamma w_k f_\theta(x))} = \sum_{(x,y)\in D^{test}} (-\gamma w_y f_\theta(x) + log\sum_k \gamma exp(w_k f_\theta(x)))$$

از رابطه ذکر شده برای آپدیت متا_پارامترها (θ) استفاده می‌شود. اما باید توجه داشت که روابط فوق علاوه بر وابستگی مستقیم، از طریق w نیز به θ وابستگی دارند. لذا در محاسبه مشتق‌های جزئی، رابطه محاسبه شده در رابطه (۸) مقاله مورد استفاده واقع می‌شود.

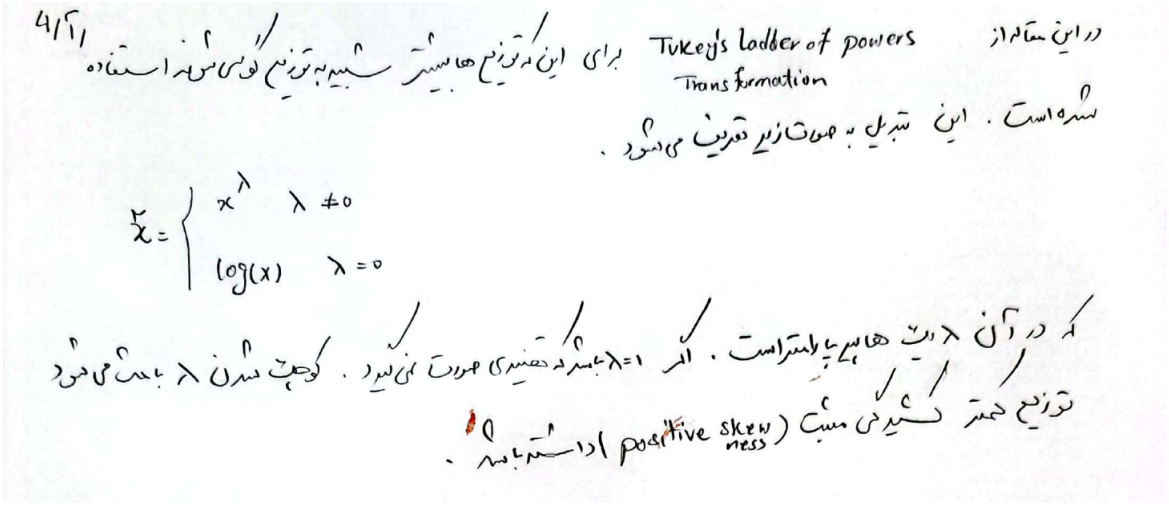
سوال ۴: (نظری) تنظیم توزیع برای یادگیری چندنمونه‌ای (۱۵ نمره)

یکی از ریسک‌های احتمالی در یادگیری چندنمونه‌ای احتمال بیش برازش [19] بر روی دادگان کم‌تعداد آموزشی است. در این مقاله روشی پیشنهاد شده است تا به کمک استخراج مشخصات آماری کلاس‌های حاضر در متا_آموزش بتوان توزیع دادگان کلاس‌های حاضر در متا_ارزیابی را تنظیم کرد. این مقاله را به دقت خوانده و به سوالات زیر به طور کامل پاسخ دهید:

(آ) از آنجایی که ممکن است توزیع دادگان هر کلاس حاضر در متا_آموزش گاوسی نباشد و دارای مقداری کشیدگی باشد؛ در نظر گرفتن این توزیع‌ها به عنوان توزیع گاوسی و استخراج میانگین و کواریانس از آن‌ها می‌تواند اشتباه باشد. توضیح دهید این مقاله چه روشی را برای حل مشکل کشیدگی توزیع دادگان متا_آموزش اتخاذ کرده است و چگونه این روش موجب حل مشکل کشیدگی توزیع می‌شود؟(۵ نمره)

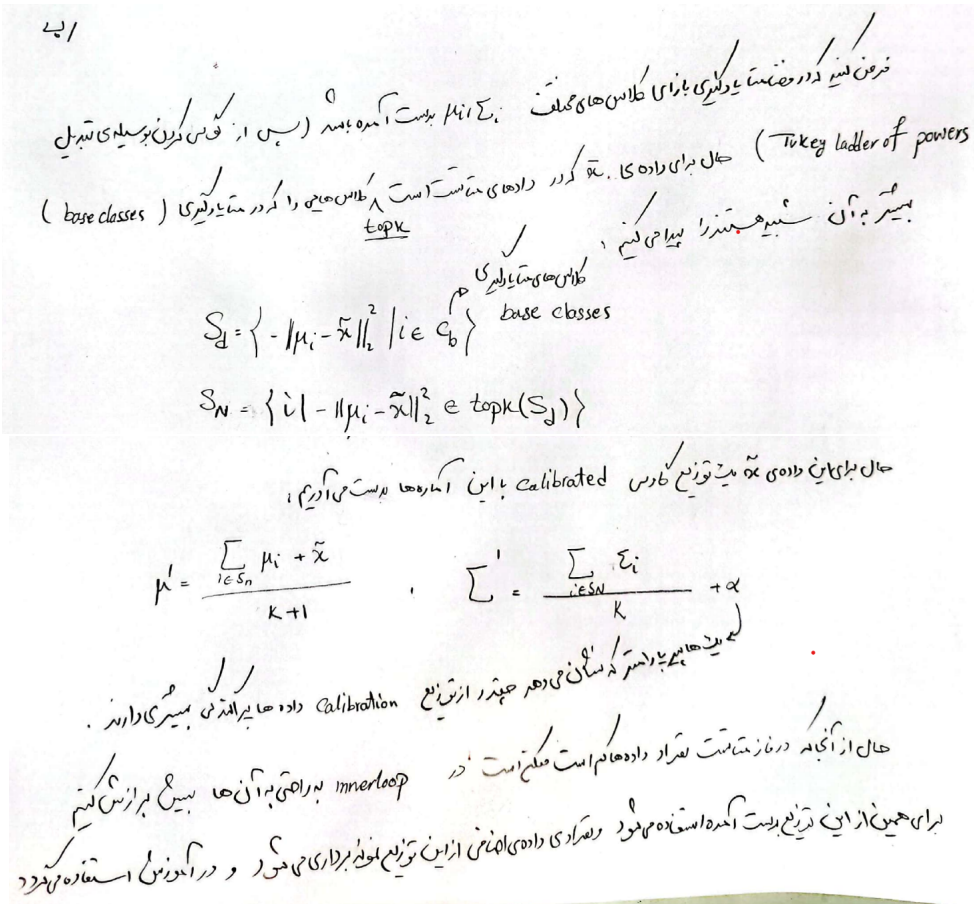
[19] Overfitting

پاسخ



(ب) پس از استخراج میانگین و کواریانس کلاس‌های حاضر در متا_آموزش، مدل ارائه شده اقدام به تنظیم توزیع دادگان حاضر در متا_ارزیابی می‌کند. به صورت کامل و با نوشتن روابط ریاضی مربوطه بیان کنید این تنظیم توزیع به چه صورت انجام می‌پذیرد و وجود کلاس‌های مشابه در متا_آموزش به کلاس منظور در متا_ارزیابی چه کمکی به تنظیم توزیع می‌کند؟ (۵ نمره)

پاسخ

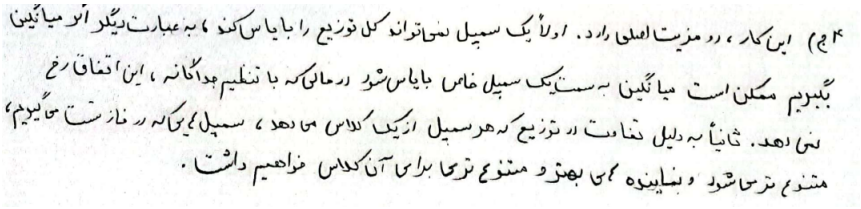


همچنین برای توضیح بیشتر فرض کنید در هنگام متاست تکلیف ما جدا کردن کلاس روباه از کلاغ در تنظیمات نمونه کم باشد. در صورتی که ما در متاآموزش کلاس های گرگ و کبوتر را داشته باشیم نمونه های کلاس های گرگ و کبوتر به ترتیب میتوانند به کلاس های روباه و کلاغ به علت شباهت کلاس ها کمک کنند و مانع از آورفیت شدن شبکه بر روی تعداد کم داده ها شوند.

(ج) در تنظیماتی که در هنگام متا_ارزیابی از هر کلاس بیش از یک نمونه آموزش داشته باشیم این مدل به جای میانگین‌گیری از نمونه‌ها، برای هر

کدام از k نمونه آموزش اقدام به تنظیم توزیع جداگانه می‌کند. توضیح دهید تنظیم توزیع جداگانه چه مزیتی نسبت به میانگین گیری نمونه‌ها و سپس یک تنظیم توزیع دارد؟(۵ نمره)

پاسخ



به علاوه در صورتی که توزیع دادگان کلاس حاضر در متاتست یک توزیع چند قله ای باشد، استفاده از یک نمونه نمی تواند مدل خوبی از آن توزیع چند قله ای ارائه دهد. در حالی که استفاده از چند نمونه این مشکل را حل می کند.

سوال ۵: (عملی) یادگیری چندنمونه‌ای از طریق یادگیری متریک (۲۵ نمره)

در این سوال قصد داریم تا مدل یادگیرنده شبکه Prototypical را مورد پیاده‌سازی و بررسی قرار دهیم. به این منظور هر دو زیر مجموعه‌های آموزش و تست دادگان CIFAR100 را دریافت کرده و سپس آن‌ها را به یکدیگر الحاق نمایید. سپس این دادگان را به سه زیر مجموعه متا_آموزش، متا_اعتبارسنجی [۲۰] و متا_ارزیابی تقسیم کنید. به این صورت که دادگان آموزش شامل دادگان ۷۰ کلاس، دادگان اعتبارسنجی شامل ۲۰ کلاس و دادگان کلاس تست شامل ۱۰ دیگر باشند. در گام بعدی بایستی یک Sampler پیاده‌سازی کنید که با گرفتن پارامترهای Way و Shot بتواند دادگان اتکا و پرسمان برای یک وظیفه را تولید کنند (در واقع این ماژول با هر فراخوانی دو مجموعه داده به اندازه $Way * Shot$ برای اتکا و پرسمان خروجی میدهد). این ماژول در واقع هر بار یک اپیزود را تولید میکند. در طی آزمایش‌های زیر از ماژول ProtoNetBack که در فایل‌های پیوستی قرار داده شده است به عنوان شبکه Backbone استخراج ویژگی استفاده کنید و در جلوی آن دو لایه تمام متصل [۲۱] با اندازه دلخواه قرار دهید.

(آ) دسته‌بند را با تنظیمات 8-shot, 10-way آموزش دهید و سپس دقت مدل را بر روی دادگان متا_ارزیابی گزارش دهید. انتظار می‌رود دقت در این بخش بیشتر از ۵۱ درصد باشد.

(ب) به ازای هر یک از تنظیمات $shot \in \{1, 2, 4, 8, 16\}$ و 10-way برای متا_آموزش و متا_ارزیابی آزمایش بالا را تکرار کرده و نمودار دقت متا_ارزیابی بر حسب shot را رسم نمایید.

(ج) حال به ازای هر یک از تنظیمات $way \in \{2, 4, 8, 16, 32\}$ و 5-shot برای متا_آموزش آزمایش را تکرار کرده و نمودار دقت متا_ارزیابی بر حسب way را رسم نمایید. (دقت کنید که در هنگام متا_ارزیابی از تنظیمات 5-shot, 10-way استفاده نمایید)

(د) حال در هنگام متا_آموزش با تنظیمات 10-shot, 10-way دسته‌بند را آموزش دهید. در هنگام متا_ارزیابی اما متا_ارزیابی را به ازای هر یک از تنظیمات $shot \in \{1, 5, 10, 15, 20\}$ انجام دهید و نمودار دقت آن را بر حسب shot رسم نمایید.

سوال ۶: (عملی) متا_یادگیری براساس بهینه‌سازی (۵۰ نمره)

در این سوال قصد داریم تا مدل معروف دسته متا_یادگیری براساس بهینه‌سازی،MAML، را پیاده‌سازی نماییم. مقاله مرتبط با این کار، این مقاله می‌باشد. در Notebook داده شده تمام پارامترهای مسئله و مراحل حل به صورت گام به گام تشریح شده است. سوال از دو بخش اصلی تشکیل شده است که در بخش اول به دلیل کاهش هزینه آموزش بخش عمده شبکه به صورت pretrained شده در اختیار شما قرار داده شده است و شما تنها روی بخش مشخص شده شبکه فرایند متا_یادگیری را انجام خواهید داد. در بخش اول قرار است تاثیر تعداد گام‌های به‌روزرسانی مدل در حلقه داخلی الگوریتم، مورد بررسی قرار گیرد. از شما خواسته شده است که به ازای مقادیر ۱ تا ۳ این مورد را انجام دهید و نتیجه هر حالت را مقایسه و گزارش کنید. در بخش دوم نیز از شما خواسته شده است که حال با یک گام به‌روزرسانی حلقه داخلی، کل ساختار مدل (مدل متا_یادگیری بخش اول + ساختار مدل pretrained داده شده) را به صورت متاپارامتر در نظر بگیرید و متا_یادگیری را روی آن انجام دهید. در نهایت نتایج بدست آمده از هردو بخش را تحلیل و گزارش نمایید.

[20] Meta-Validation
[21] Fully Connected