به نام خدا

نظریه یادگیری ماشین

دکتر سیدصالحی

جلسه هشتم

دانشکده ریاضی و علوم کامپیوتر

بهار ۱۴۰۳

۱ مقدمه

به طور معمول، هنگامی که یک مدل یادگیری ماشین را آموزش می‌دهیم به یک مجموعه از داده‌های آموزش[۱] دسترسی داریم که به کمک آن می‌توانیم میزان خطا را بر روی این داده‌ها محاسبه کنیم. به این خطا، خطای یادگیری[۲] گفته می‌شود و ما به دنبال کاهش این خطا هستیم.

تا به اینجای کار یک مسئله بهینه‌سازی را تعریف کرده‌ایم. آنچه یادگیری ماشین را نسبت به بهینه‌سازی[۳] متمایز می‌کند آن است که در یادگیری ماشین ما می‌خواهیم که خطای تعمیم[۴] یا خطای آزمون[۵] نیز کاهش پیدا کند. خطای تعمیم را به صورت میزان خطای مورد انتظار بر روی داده‌های دیده نشده تعریف می‌کنیم. برای تخمین خطای تعمیم یک مدل یادگیری ماشین، می‌توانیم میزان خطای این مدل را بر روی یک مجموعه از داده‌های آزمون[۶] محاسبه کنیم که مستقل از داده‌های آموزش باشد.

حال این پرسش مطرح می‌شود که چگونه می‌توانیم میزان خطای مدل را بر روی داده‌های آزمون کاهش دهیم در صورتی که تنها به داده‌های آموزش دسترسی داریم؟ بر اساس نظریه یادگیری آماری[۷] می‌توانیم به این پرسش پاسخ دهیم. اگر داده‌های آموزش و آزمون به صورت کاملاً دلخواهانه انتخاب شده باشند، و فرض خاصی بر روی داده‌ها نداشته باشیم، نمی‌توانیم کار زیادی انجام دهیم. اگر بتوانیم نسبت به نحوه‌ی جمع‌آوری داده‌های آموزش و آزمون فرضیاتی داشته باشیم، می‌توانیم کمی پیشروی کنیم. فرض می‌کنیم داده‌های آموزش و آزمون با استفاده از یک توزیع احتمالاتی مشترک تحت فرآیندی به نام فرآیند تولید داده[۸] ایجاد شده باشند و هر یک از نمونه‌های جمع‌آوری شده مستقل از سایر نمونه‌ها باشد.[۹] با کمک این

[1] training set
[2] training error
[3] optimization
[4] generalization error
[5] test error
[6] test set
[7] statistical learning theory
[8] data-generating process
[9] independent and identically distributed (i.i.d.) assumptions

فرض می‌توانیم در یک چارچوب احتمالاتی به مطالعه رابطه میان خطای یادگیری و خطای آزمون بپردازیم.

در فرآیندهای یادگیری ماشین ما ابتدا یک نمونه‌برداری انجام می‌دهیم تا مجموعه داده‌های یادگیری بسازیم، سپس با استفاده از الگوریتم‌های یادگیری ماشین تلاش می‌کنیم تا خطای آموزش را کاهش دهیم. در این صورت انتظار می‌رود که مقدار خطای آزمون بزرگ‌تر یا مساوی با مقدار خطای آموزش باشد. در این صورت می‌توانیم عوامل مؤثر در عملکرد الگوریتم‌های یادگیری ماشین را به صورت زیر خلاصه کنیم:

۱. کاهش خطای آموزش

۲. کاهش اختلاف خطای آموزش و خطای آزمون

هر یک از این دو عامل، به چالش‌های بنیادینی که در یادگیری ماشین با آن‌ها مواجه هستیم، یعنی کم‌برازش[10] و بیش‌برازش[11]، اشاره می‌کنند که در ادامه به معرفی آن‌ها می‌پردازیم.

۱-۱ کم‌برازش

کم‌برازش هنگامی رخ می‌دهد که نمی‌توانیم مقدار خطای مدل را بر روی داده‌های یادگیری کاهش دهیم. در این صورت خطای آموزش نسبتاً مقداری بزرگ خواهد داشت. با توجه به فرض‌هایی که در مورد داده‌ها داشتیم، انتظار می‌رود که مقدار خطای آزمون نیز نسبتاً بزرگ باشد.

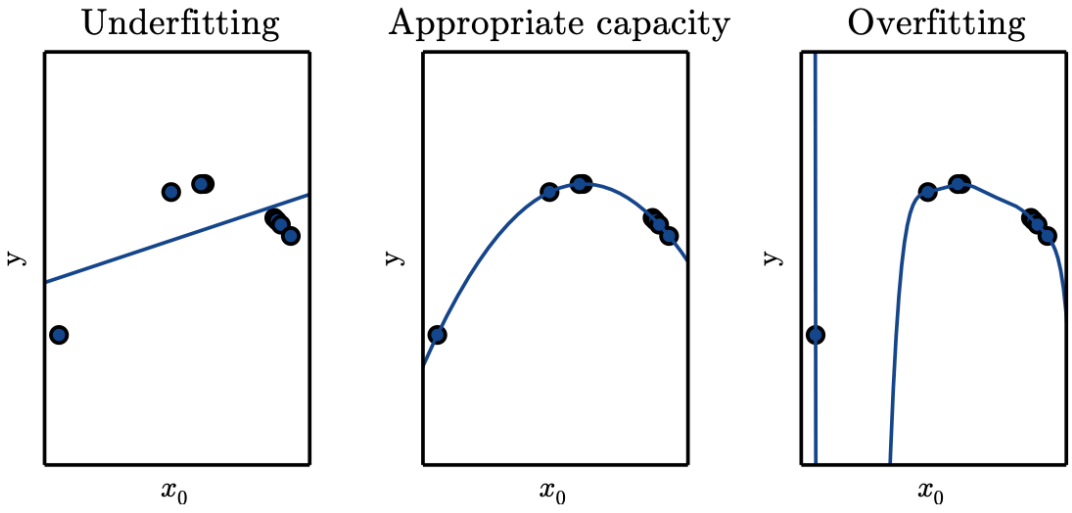
۱-۲ بیش‌برازش

بیش‌برازش هنگامی رخ می‌دهد که مدل آموزش داده شده عملکرد مناسبی در برابر داده‌های یادگیری داشته باشد، ولی در برابر داده‌های دیده نشده قادر به تعمیم نباشد و عملکرد نامناسبی از خود نشان دهد. در این صورت می‌توان گفت که خطای آموزش مدل نسبتاً کوچک است ولی خطای آزمون آن نسبتاً بزرگ است. به بیانی دیگر خطای آموزش و خطای آزمون اختلاف نسبتاً بزرگی دارند.

۱-۳ ظرفیت*

می‌توانیم احتمال بیش‌برازش یا کم‌برازش یک مدل را توسط ظرفیت[12] آن کنترل کنیم. به زبان ساده ظرفیت یک مدل بیانگر قابلیت برازش گستره‌ای از توابع مختلف توسط مدل است. مدل‌هایی که ظرفیت بالایی دارند می‌توانند با حفظ خواص حاضر در مجموعه داده‌های آموزش دچار بیش‌برازش شوند و عملکرد مناسبی بر روی داده‌های تست نداشته باشند.

[10] underfitting
[11] overfitting
[12] capacity



شکل ۱: در تصویر فوق ۳ مدل را بر روی یک مجموعه از داده‌ها آموزش می‌دهیم. داده‌های فوق با یک نمونه برداری تصادفی از مقادیر x و محاسبه مقادیر y توسط یک تابع درجه دو بدست آمده‌اند. (چپ) یک مدل خطی آموزش داده شده بر روی داده‌های فوق، از کم‌برازش رنج می برد و نمی‌تواند انحنای موجود در داده‌ها را ثبت کند. (وسط) یک مدل درجه دوم آموزش داده شده بر روی داده‌های فوق، به خوبی به نقاط دیده نشده تعمیم می‌یابد و از مقدار قابل توجهی کم‌برازش یا بیش‌برازش رنج نمی‌برد. (راست) یک مدل چندجمله‌ای از درجه ۹ آموزش داده شده بر روی داده‌های فوق، از بیش‌برازش رنج می‌برد ولی اگرچه مدل آموزش داده شده از همه‌ی نقاط به صورت دقیق عبور می‌کند، قادر به تعمیم بر روی داده‌های دیده نشده نیست.

یک روش برای کنترل ظرفیت یک الگوریتم یادگیری، انتخاب فضای فرضیه[۱۳] آن است. فضای فرضیه یک الگوریتم یادگیری، مجموعه توابعی است که مدل مجاز به گزینش آن‌ها به عنوان پاسخ مسئله است.

برای مثال فضای فرضیه الگوریتم رگرسیون خطی[۱۴] شامل تمامی توابع خطی ورودی‌های آن می‌باشد. ما می‌توانیم این الگوریتم را به گونه‌ای تعمیم دهیم (با افزودن درجات بالاتر ورودی‌ها به مجموعه داده‌های آموزش) که فضای فرضیه آن علاوه بر توابع خطی، توابع چندجمله‌ای را نیز در برگیرد. با انجام این کار ظرفیت مدل افزایش می‌یابد. برای نمونه شکل ۱ را در نظر بگیرید.

تا به اینجای کار تنها با روش تغییر تعداد ویژگی‌ها[۱۵] و افزودن پارامترهای جدید متناظر با این ویژگی‌ها ظرفیت مدل را تغییر دادیم. اما روش‌های دیگری برای تغییر ظرفیت یک مدل نیز وجود دارد. به همین منظور به سراغ مفهومی به نام ظرفیت توصیفی[۱۶] می‌رویم. در واقع هنگامی که یک مدل را برای آموزش بر می‌گزینیم، در اصل خانواده‌ای از توابعی که الگوریتم یادگیری می‌تواند آن‌ها را انتخاب کند تا خطا را کاهش دهد مشخص می‌شود. در عمل الگوریتم‌های یادگیری اغلب قادر نخواهند بود که بهترین تابع را پیدا

[13] hypothesis space
[14] linear regression
[15] feature
[16] representational capacity

کنند بلکه صرفاً قادر به یافتن توابعی هستند که مقدار خطای آموزش را به اندازه قابل توجهی کاهش دهند. بی‌نقص نبودن فرآیند بهینه‌سازی باعث می‌شود که ظرفیت مؤثر[17] یک مدل کمتر از ظرفیت توصیفی آن باشد.

۲ بایاس و واریانس

در این بخش ابتدا نگاهی به مفاهیم بایاس و واریانس می‌اندازیم. سپس به مصالحه بایاس و واریانس[18] می‌پردازیم و در انتها تجزیه بایاس و واریانس[19] را بررسی می‌کنیم.

۲-۱ بایاس

بایاس یک تخمین‌گر به صورت زیر تعریف می‌شود:

$$\text{Bias}(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta$$

که امید ریاضی بر روی داده‌ها است و θ مقدار واقعی است که توزیع تولید داده را تعریف می‌کند. پس بایاس بیانگر میزان اختلاف مورد انتظار میان مقدار واقعی و مقدار پیش‌بینی شده توسط مدل است. خطای بایاس خطای ناشی از فرضیات اشتباه در الگوریتم یادگیری است. بایاس بالا می‌تواند باعث شود که یک الگوریتم روابط مربوطه بین ویژگی‌ها و خروجی‌های هدف را از دست بدهد.

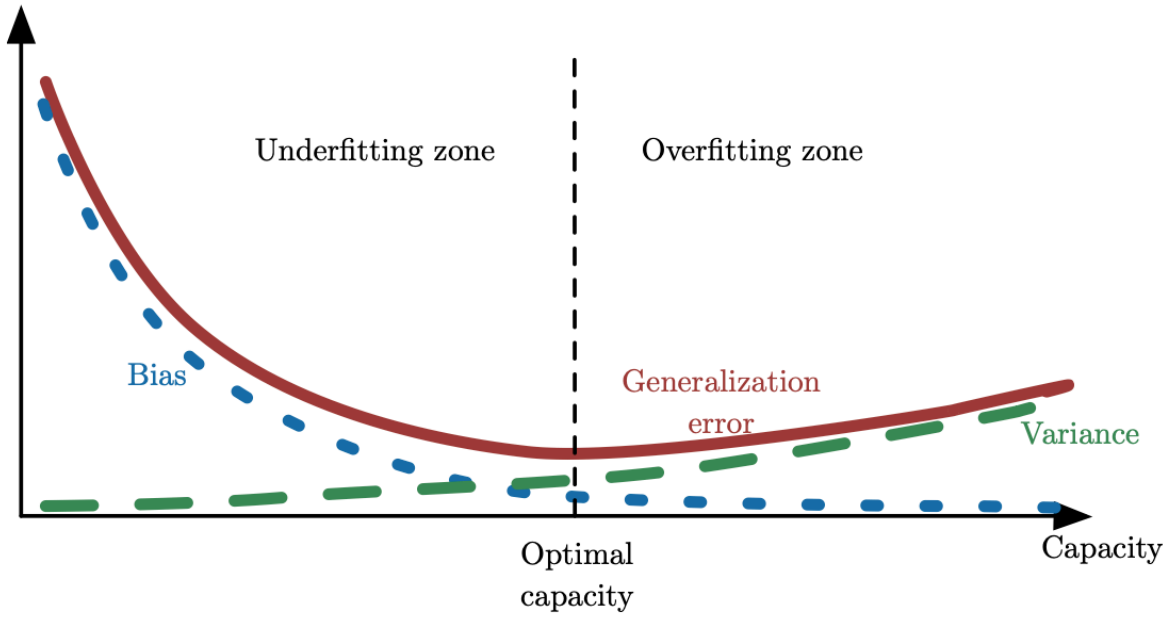
۲-۲ واریانس

واریانس یک تخمین‌گر به صورت زیر تعریف می‌شود:

$$\text{Var}(\hat{\theta}) = \mathbb{E}[\hat{\theta}^2] - \mathbb{E}[\hat{\theta}]^2$$

واریانس یک مدل بیانگر میزان تغییرات مورد انتظار در صورت آموزش با یک مجموعه داده‌ی جدید با نمونه برداری دوباره از فرآیند ایجاد تولید داده می‌باشد. واریانس یک خطا از حساسیت به نوسانات کوچک در مجموعه داده‌های آموزش است. واریانس بالا می‌تواند باعث شود که یک الگوریتم نویز حاضر در مجموعه داده‌های یادگیری را مدل کند و آن‌ها را یاد بگرید.

[17] effective capacity
[18] bias–variance tradeoff
[19] bias-variance decomposition



شکل ۲: با افزایش ظرفیت (محور افقی)، بایاس (نقطه‌چین) کاهش یافته و واریانس (خط‌چین) افزایش می‌یابد. در این صورت خطای تعمیم به فرم یک منحنی U شکل در می‌آید. پس می‌توان ادعا کرد که یک ظرفیت بهینه وجود دارد به گونه‌ای که اگر مقدار آن را کاهش دهیم کم‌برازش، و اگر مقدار آن را افزایش دهیم بیش‌برازش رخ دهد.

۲-۳ مصالحه بایاس و واریانس

بایاس و واریانس دو نوع خطای متفاوت را برای یک تخمین‌گر اندازه‌گیری می‌کنند. بایاس میزان انحراف مورد انتظار از مقدار واقعی تابع و یا پارامتر را اندازه‌گیری می‌کند. واریانس معیاری از انحراف از مقدار تخمین‌گر مورد انتظار را ارائه می‌کند که هر نمونه‌گیری خاص از داده‌ها ممکن است باعث ایجاد آن شود.

رابطه بین بایاس و واریانس با مفاهیم یادگیری ماشین نظیر ظرفیت، کم‌برازش و بیش‌برازش ارتباط تنگاتنگی دارد. هنگامی که خطای تعمیم توسط میانگین مربعات خطا[۲۰] اندازه گیری می‌شود (که در آن بایاس و واریانس اجزای معنی‌داری در خطای تعمیم هستند)، افزایش ظرفیت اغلب منجر به افزایش واریانس و کاهش بایاس می‌شود.

۲-۴ تجزیه بایاس و واریانس

تجزیه بایاس و واریانس روشی برای تجزیه و تحلیل خطای تعمیم مورد انتظار الگوریتم یادگیری با توجه به یک مسئله خاص به عنوان مجموع سه عبارت، بایاس، واریانس و کمیتی به نام خطای کاهش ناپذیر است که ناشی از نویز موجود در مجموعه داده‌های آموزش حاصل می شود. در صورتی که در داده‌های آموزش

[20] mean squared error

نویزی نداشته باشتیم می‌توانیم میانگین مربعات خطای یک مدل را به صورت زیر بنویسیم:

$$\text{MSE} = \mathbb{E}\left[(\hat{\theta} - \theta)^2\right] = \text{Bias}(\hat{\theta})^2 + \text{Var}(\hat{\theta})$$

می‌توانیم این رابطه را تعمیم دهیم. فرض کنید که یک مجموعه از داده‌های آموزش شامل نقاط $x_1, \cdots, x_n$ و مقادیر حقیقی y_i را که منتناظر با x_iها هستند داریم. همچنین در نظر بگیرید که داده‌ها توسط تابع $f(x)$ تولید شده باشند و $y = f(x) + \epsilon$ باشد که ϵ مقداری نویز با میانگین 0 و واریانس σ^2 می‌باشد. در این صورت خواهیم داشت:

$$\begin{aligned}
\text{MSE} &= \mathbb{E}\left[(\hat{y} - y)^2\right] \\
&= \mathbb{E}\left[(f(x) + \epsilon - y)^2\right] \\
&= \text{Bias}(\hat{y})^2 + \text{Var}(\hat{y}) + \sigma^2
\end{aligned}$$

۳ منظم‌سازی

برای جلوگیری از بیش‌برازش مدل‌ها می‌توانیم از منظم‌سازی[۲۱] استفاده کنیم. امکان دارد گاهی با شرایطی مواجه شویم که ۲ یا چند تابع قادر به کاهش خطای مورد نظر بر روی داده‌های آموزش شوند. حال پرسشی که مطرح می‌شود این است که کدام یک از این توابع عملکرد بهتری بر روی داده‌های آزمون خواهد داشت؟

برای پاسخ به این پرسش به یک اصل فلسفی به نام تیغ اوکام[۲۲] رجوع می‌کنیم. این اصل به زبان ساده بیان می‌کند که «میان دو نگره که توان توصیف و پیش‌بینی یکسانی دارند، ساده‌ترین را برگزین» و در اینجا با اتکا بر این اصل، تلاش می‌کنیم که در انتخاب توابع و مدل‌ها، ساده‌ترین را ترجیح دهیم.

اگر بتوانیم میزان پیچیدگی و سادگی مدل‌های مختلف را به صورت کمی مقایسه کنیم، با افزودن ضریبی از آن مقدار کمی به تابع هزینه اولیه می‌توانیم ترجیح توابع ساده‌تر را به زبان ریاضی بیان کنیم. برای نمونه اگر تابع هزینه $J(w)$ را داشته باشیم، می‌توانیم آن را به صورت زیر تغییر دهیم:

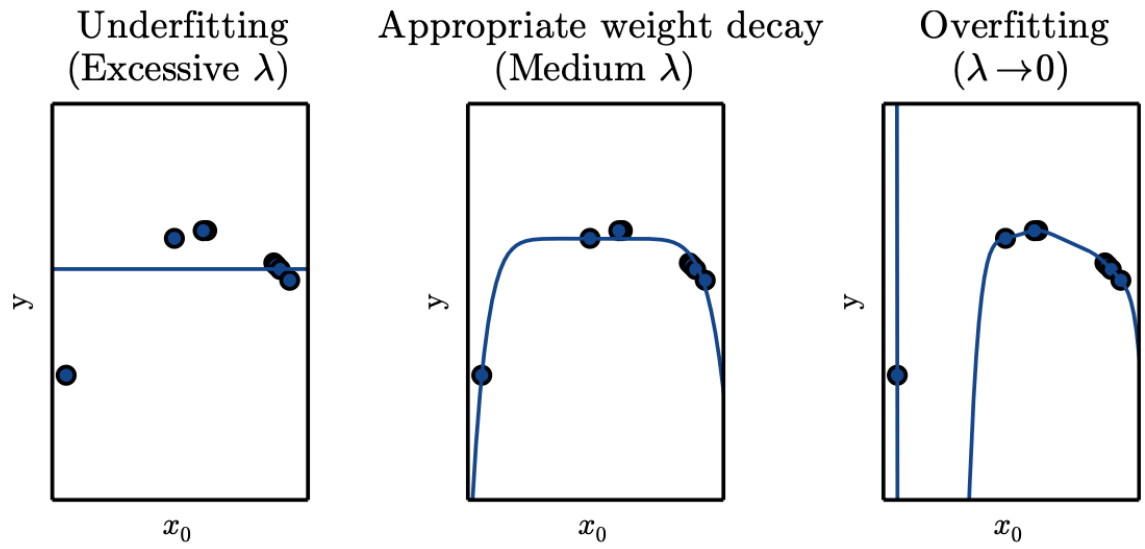
$$J_\lambda(w) = J(w) + \lambda R(w)$$

که در اینجا λ یک ابرپارامتر[۲۳] است و $R(w)$ یک تابع منظم‌سازی است که در روش‌های کلاسیک اغلب

[21] Regularization

[22] Occam's razor (Ockham's razor)

[23] hyperparameter



شکل ۳: مسئله‌ای که در شکل ۱ بررسی کردیم را به یاد بیاورید. در اگر ندانیم تابع واقعی از درجه ۲ است، مجبور می‌شویم که برای یافتن ظرفیت مناسب درجات مختلف را بررسی کنیم. هنگامی که مسائل پیچیده‌تر مواجه شویم این امر امکان‌پذیر نخواهد بود و یافتن ظرفیت ایده‌آل بسیار وقت‌گیر و یا حتی غیرممکن خواهد شد. یک راه برای جلوگیری از این مشکل این است که مدلی با ظرفیت بالا را انتخاب کرده و با منظم‌سازی تلاش کنیم تا به پاسخی قابل‌قبول برسیم که عملکرد تقریباً مشابهی نسبت به حالت ایده‌آل داشته باشد. در اینجا یک چندجمله‌ای درجه ۹ را با مقادیر مختلف λ آموزش دادیم. (راست) در صورتی که مقدار λ برابر با صفر باشد عملاً هیچ منظم‌سازی رخ نداده و همچنان بیش‌برازش رخ می‌دهد. (چپ) در صورتی که مقدار λ بیش از اندازه زیاد باشد هیچ یادگیری صورت نگرفته و کم‌برازش رخ می‌دهد. (وسط) در صورتی که مقدار معقولی برای λ انتخاب شود، مدل نهایی برخلاف ظرفیت بالاتری که نسبت به ظرفیت واقعی دارد، عملکرد مناسبی خواهد داشت.

تابعی از پارامترهای مدل می‌باشد. اگر مقادیر زیاد $R(w)$ متناظر با پیچیدگی بیشتر و مقادیر کم متناظر با سادگی باشند، تابع هزینه جدید از میان مدل‌هایی که $J(w)$ تقریباً برابر دارند، مدلی که ساده‌تر است را به عنوان پاسخ بهینه معرفی می‌کند. مقدار λ تعیین می‌کند که تا چه اندازه سادگی مدل برای ما اهمیت دارد. اگر این مقدار برابر با صفر باشد هیچ محدودیتی بر روی پیچیدگی مدل اعمال نمی‌شود و در صورتی که مقدار آن را افزایش دهیم،اهمیت بیشتری به سادگی مدل داده خواهد شد.

برای منظم‌سازی روش‌های متعددی وجود دارد. یک روش متداول برای انجام این کار روش کاهش وزن‌ها[۲۴] است که در رگرسیون خطی، یادگیری ژرف[۲۵] و ... استفاده می‌شود. با نظر گرفتن نرم‌های ℓ_2 وزن‌ها به عنوان تابع منظم‌سازی می‌باشد. با انجام این کار وزن‌های کوچک‌تر نسبت به وزن‌های بزرگ‌تر ترجیح داده می‌شوند. در این صورت وزرن‌های بزرگ‌تر تنها در صورتی انتخاب می‌شوند که بتوانند مقدار خطای آموزش را به طرز چشمگیری در مقایسه با وزن‌های کوچک‌تر کاهش دهند. اگر بجای نرم ℓ_2 از نرم

[24] weight decay
[25] deep learning

ℓ_1 استفاده کنیم، وزن‌های تنک[۲۶] ترجیح داده می‌شوند. در شکل ۳ می‌توانید اثرات این نوع منظم‌سازی که از نرم ℓ_2 استفاده می‌کند را ببینید.

از دیگر روش‌های منظم‌سازی می‌توان به توقف زودهنگام[۲۷] اشاره کرد. بسیاری از الگوریتم‌های یادگیری ماشین به صورت مرحله به مرحله بهبود می‌یابند. در این موارد اغلب با بروزرسانی پارامترها تلاش می‌کنیم تا خطای آموزش را به صورت پی در پی کاهش دهیم تا به پارامترها و یا وزن‌های بهینه برسیم. اما امکان دارد که پارامترهایی که متناظر با خطای کمینه باشند، با حفظ خواص مجموعه داده‌های آموزش به این دستاورد برسند و نتوانند به خوبی بر روی داده‌های آزمون تعمیم یابند. یک روش برای پیشگیری از بروز چنین مشکلاتی توقف زودهنگام الگوریتم است. در این صورت شرایط توقف زودهنگام (تعداد مراحل، خطای توقف، ...) را می‌توان به صورت یک ابرپارامتر در نظر گرفت. این روش از منظم‌سازی در آموزش درخت‌های تصمیم[۲۸]، جنگل‌های تصادفی[۲۹]، شبکه‌های عصبی[۳۰] و ... استفاده می‌شود.

اکنون امکان دارد که این پرسش مطرح شود که چگونه می‌توانیم ابرپارامترهای مناسب برای آموزش مدل‌هایمان را پیدا کنیم؟ در بخش بعدی نگاهی به روش‌هایی برای یافتن ابرپارامترهای بهینه و مفهوم صحت‌سنجی[۳۱] می‌اندازیم.

۴ صحت‌سنجی

در بخش پیش با مفهوم منظم‌سازی و ابرپارامتر آشنا شدیم. ابرپارامترها بر خلاف پارامترهای معمولی، که از طریق آموزش به دست می‌آیند و عملکرد مدل را کنترل می‌کنند، برای کنترل خود فرآیند آموزش بکار گرفته می‌شوند و با استفاده درست از آن‌ها می‌توان سرعت و کیفیت فرآیند یادگیری را افزایش داد. برای مثال درجه‌ی چندجمله‌ای در مسئله رگرسیون، تعداد لایه‌های شبکه‌های عصبی، ضریب منظم‌سازی، تعداد مراحل اجرای الگوریتم، و ... ابرپارامتر به شمار می‌آیند ولی وزن‌های مسئله رگرسیون و یا وزن‌های شبکه‌های عصبی پارامترهای مدل می‌باشند.

پارامترهای بهینه در فرآیند آموزش و بر اساس الگوریتم یادگیری، به گونه‌ای یادگرفته می‌شوند که خطای مدل را بر روی داده‌های آموزش کاهش دهند. برای اینکه بتوانیم ابرپارامترهای مناسب را پیدا کنیم، می‌توانیم داده‌های آموزش را به دو دسته تقسیم کرده و با استفاده از یکی از آن‌ها پارامترها را پیدا کنیم و با استفاده از دیگری ابرپارامترها را بیابیم. در این تقسیم‌بندی جدید به مجموعه داده‌های دسته اول، مجموعه

[26] sparse
[27] early stopping
[28] decision tree
[29] random forest
[30] neural network
[31] validation

داده‌های آموزش گفته می‌شود (که زیرمجموعه‌ای از داده‌های آموزش سابق است)، به دسته دوم مجموعه داده‌های صحت‌سنجی می‌گوییم.

با استفاده از این تقسیم‌بندی می‌توانیم برای تعداد ابرپارامتر دلخواه، فرآیند آموزش را به کمک داده‌های آموزش انجام دهیم، و به کمک داده‌های صحت‌سنجی ابرپارامتر را که منجر به یافتن بهترین پارامترها می‌شود را بیابیم. در واقع میزان خطای مدل آموزش داده شده بر روی داده‌های صحت‌سنجی معیاری برای تخمین خطای مدل بر روی داده‌های آزمون است. پس به این روش می‌توانیم بهترین ابرپارامترهای مدل را پیدا کنیم.

اما این روش بدون ایراد نبوده و ما را با چالش‌های جدیدی مواجه می‌کند. در واقع با تقسیم داده‌ها به دو مجموعه آموزش و صحت‌سنجی، تعداد داده‌های آموزش محدود شده و این منجر به کاهش عملکرد مدل خواهد شد. اثرات این پدیده در صورت کم بودن داده‌های در دسترس، و یا بزرگ بودن مجموعه داده‌های صحت‌سنجی تشدید می‌شود. اگر هم تعداد داده‌های صحت‌سنجی را کاهش دهیم، تخمین ما از عملکرد مدل افت پیدا کرده و غیر قابل اطمینان می‌شود. یک راه حل برای این موضوع استفاده از صحت‌سنجی ضربدری[۳۲] است که در ادامه به آن می‌پردازیم.

۱-۴ صحت‌سنجی ضربدری

یک روش برای انجام صحت‌سنجی هنگامی که به تعداد اندکی داده دسترسی داریم، این است که داده‌هایمان را به طور تصادفی به k قسمت تقسیم کنیم، سپس با توجه به تقسیم‌بندی انجام شده به تعداد k بار، هر دفعه یک بخش از داده‌ها را به عنوان داده‌های صحت‌سنجی کنار گذاشته و مدل را بر روی $k-1$ بخش دیگر آموزش دهیم. در این صورت می‌توانیم از میانگین نتایج صحت‌سنجی استفاده کرده و مدلی که بهترین عملکرد را دارد را به عنوان مدل نهایی انتخاب کنیم. به این الگوریتم $k-$ fold cross-validation می‌گویند.

این روش از صحت‌سنجی نیز معایبی دارد. برای نمونه هزینه محاسباتی به صورت چشمگیری افزایش می‌یابد. به همین دلیل استفاده از این روش برای مدل‌هایی که آموزش آن‌ها زمان زیادی می‌برد (برای مثال در آموزش شبکه‌های عصبی ژرف) امکان‌پذیر نیست. علاوه بر این قضیه انتخاب مقدار مناسب برای k نیز می‌تواند تبدیل به کار دشواری شود.

[32] cross-validation