

نویسنده: آندرو ان جی.



آموزش یادگیری ماشین
به سبک Andrew Ng.

مترجمان:

دکتر مهدی اسماعیلی

حمیدرضا فلاح کتک لاهیجانی

میینا قرائی

آموزش یادگیری ماشین به سبک Andrew Ng.

مترجمان: دکتر مهدی اسماعیلی، حمیدرضا فلاح کتک لاهیجانی، مبینا قرائی
ویراستار علمی: دکتر رامین مولاناپور
طراحی جلد و صفحه‌آرایی: همتا بیداریان
ناشر: انتشارات آتی‌نگر
ناشر همکار: انتشارات وینا
چاپ اول، ۱۴۰۰
شمارگان: ۱۰۰۰ نسخه
قیمت به همراه DVD: ۶۴۰,۰۰۰ ریال
شابک: ۹۷۸-۶۲۲-۷۵۷۱-۳۶-۳

ISBN: 978-622-7571-36-3

حق چاپ برای انتشارات آتی‌نگر محفوظ است.

نشانی دفتر فروش: خیابان جمالزاده جنوبی، روبه‌روی کوچه رشتچی، پلاک ۱۴۴، واحد ۱
تلفن: ۸-۶۶۵۶۵۳۳۶ ۶۶۵۶۵۳۳۷
نمابر: ۶۶۵۶۵۳۳۷



www.ati-negar.com * info@ati-negar.com

سرشناسه: آندرو، ان جی، Andrew Ng.
آموزش یادگیری ماشین به سبک Andrew Ng؛ مترجمان: مهدی اسماعیلی، حمیدرضا فلاح کتک لاهیجانی، مبینا قرائی
تهران: آتی‌نگر، وینا ۱۴۰۰
۱۶۰ ص: مصور، جدول، نمودار.
ISBN: 978-622-7571-36-3
فیپا.
یادداشت: عنوان اصلی کتاب: Machine Learning Yearning is a deeplearning.ai project.
موضوع: فراگیری ماشینی - Machine learning - هوش مصنوعی - Artificial intelligence
موضوع: شبکه‌های عصبی (کامپیوتر (Computer science) Neural networks) (-Neural networks)
شناسه‌افزوده: اسماعیلی، مهدی، ۱۳۵۰، - مترجم
شناسه‌افزوده: فلاح کتک لاهیجانی، حمیدرضا، ۱۳۶۱، - مترجم
شناسه‌افزوده: قرائی، مبینا، ۱۳۷۴، - مترجم
شناسه‌افزوده: مولاناپور، رامین، ۱۳۵۲، - ویراستار
رده‌بندی کنگره
رده‌بندی دیویی
شماره کتابشناسی ملی
QC۳۲۵/۵
۰۰۶/۳۱
۸۵۴۰۸۰

فهرست مطالب

- ۷ ۱ چرا استراتژی یادگیری ماشین
- ۹ ۲ چگونگی استفاده از این کتاب برای کمک به تیم شما
- ۱۰ ۳ پیش‌نیازها و نمادگذاری
- ۱۱ ۴ مقیاس، پیش‌ران پیشرفت یادگیری ماشین است

راه‌اندازی مجموعه‌های اعتبارسنجی و آزمایشی

- ۱۷ ۵ مجموعه‌های اعتبارسنجی و آزمایشی شما
- ۲۰ ۶ مجموعه‌های آزمایشی و اعتبارسنجی شما باید از توزیع یکسانی آمده باشند
- ۲۲ ۷ مجموعه‌های آزمایشی و اعتبارسنجی باید چه قدر بزرگ باشند؟
- ۲۳ ۸ یک معیار ارزیابی تک عددی جهت بهینه‌سازی برای تیم خود وضع کنید
- ۲۵ ۹ معیارهای بهینه‌سازی و رضایت‌مندی
- ۲۷ ۱۰ داشتن معیار و مجموعه اعتبارسنجی تکرارها را تسریع می‌کند
- ۲۹ ۱۱ زمان تغییر معیارها و مجموعه‌های اعتبارسنجی / آزمایشی
- ۳۲ ۱۲ نکات کلیدی در ایجاد مجموعه‌های آزمایشی و اعتبارسنجی

تحلیل پایه‌ای خطا

- ۳۷ ۱۳ نخستین سیستم خود را به سرعت بسازید و سپس این کار را تکرار کنید
- ۳۸ ۱۴ تحلیل خطا: نگاه به نمونه‌های موجود در مجموعه اعتبارسنجی برای ارزیابی ایده‌ها
- ۴۰ ۱۵ ارزیابی موازی ایده‌های متعدد در طی تحلیل خطا
- ۴۳ ۱۶ پاک‌سازی نمونه‌هایی از مجموعه آزمایشی و اعتبارسنجی که به اشتباه برچسب‌گذاری می‌شوند

- ۱۷ اگر یک مجموعه اعتبارسنجی بزرگ دارید، آن را به دو زیرمجموعه تقسیم کنید و فقط یکی از آن‌ها را بررسی کنید ۴۶
- ۱۸ مجموعه‌های اعتبارسنجی مردمک چشم و جعبه سیاه باید چقدر بزرگ باشند؟ ۴۸
- ۱۹ نکات کلیدی در تحلیل خطای پایه ۵۱

بایاس و واریانس

- ۲۰ بایاس و واریانس: دو منبع بزرگ خطا ۵۵
- ۲۱ مثال‌هایی از بایاس و واریانس ۵۷
- ۲۲ مقایسه با نرخ خطای بهینه ۵۹
- ۲۳ رفع بایاس و واریانس ۶۲
- ۲۴ موازنه بایاس-واریانس ۶۴
- ۲۵ تکنیک‌های کاهش بایاس اجتناب‌پذیر ۶۵
- ۲۶ تحلیل خطا در مجموعه آموزشی ۶۶
- ۲۷ تکنیک‌های کاهش واریانس ۶۸

منحنی‌های یادگیری

- ۲۸ تشخیص بایاس و واریانس: منحنی‌های یادگیری ۷۳
- ۲۹ نمایش خطای آموزش ۷۶
- ۳۰ تفسیر منحنی‌های یادگیری: بایاس بالا ۷۸
- ۳۱ تفسیر منحنی‌های یادگیری: سایر موارد ۸۰
- ۳۲ ترسیم منحنی‌های یادگیری ۸۲

مقایسه با عملکرد سطح انسانی

- ۳۳ چرا با عملکرد سطح انسانی مقایسه می‌کنیم؟ ۸۷

- ۳۴ چگونه عملکرد سطح انسانی را تعریف کنیم؟ ۸۹
- ۳۵ سبقت گرفتن از عملکرد سطح انسانی ۹۱

توزیع‌های متفاوت مجموعه‌های آموزشی و آزمایشی

- ۳۶ چه زمانی باید آموزش و آزمایش را روی توزیع‌های مختلف انجام داد ۹۵
- ۳۷ چگونه تصمیم می‌گیرید که آیا از تمام داده‌های خود استفاده کنید یا خیر ۹۷
- ۳۸ چگونه تصمیم می‌گیرید که داده‌های ناسازگار را در نظر بگیرید یا خیر ۱۰۰
- ۳۹ وزندهی داده‌ها ۱۰۱
- ۴۰ تعمیم‌دهی از مجموعه آموزشی به مجموعه اعتبارسنجی ۱۰۳
- ۴۱ تشخیص خطاهای بایاس، واریانس و عدم‌انطباق داده‌ها ۱۰۵
- ۴۲ رفع عدم‌انطباق داده‌ها ۱۰۸
- ۴۳ تولید داده‌های مصنوعی ۱۱۰

اشکال‌زدایی الگوریتم‌های استنباطی

- ۴۴ آزمون صحت‌سنجی بهینه‌سازی ۱۱۵
- ۴۵ شکل کلی آزمون صحت‌سنجی بهینه‌سازی ۱۱۸
- ۴۶ مثال یادگیری تقویتی ۱۲۰

یادگیری ژرف انتها به انتها

- ۴۷ پیدایش یادگیری انتها به انتها ۱۲۵
- ۴۸ مثال‌های بیشتر از یادگیری انتها به انتها ۱۲۷
- ۴۹ مزایا و معایب یادگیری انتها به انتها ۱۲۹
- ۵۰ انتخاب مؤلفه‌های مجرا: دسترس‌پذیری داده‌ها ۱۳۱
- ۵۱ انتخاب مؤلفه‌های مجرا: ساده‌سازی وظیفه ۱۳۳

۱۳۸

۵۲ یادگیری خروجی‌های غنی به‌طور مستقیم

تحلیل خطای بخش‌ها

۱۴۳

۵۳ تحلیل خطای بخش‌ها

۱۴۷

۵۴ نسبت دادن خطا به یک بخش

۱۵۰

۵۵ مورد کلی انتساب خطا

۱۵۳

۵۶ تحلیل خطای بخش‌ها و مقایسه با عملکرد سطح انسانی

۱۵۵

۵۷ تشخیص مجرای معیوب

نتیجه‌گیری

۱۵۹

۵۸ تشکیل تیم ابرقهرمان - هم‌تیمی‌های خود را به مطالعه این کتاب وادار کنید



چرا استراتژی یادگیری ماشین

یادگیری ماشین^۱ شالوده‌ی بسیاری از کاربردهای مهم از جمله جستجو در وب، شناسایی هرزنامه‌ها، تشخیص گفتار، پیشنهاد کالاها و غیره است. فرض را بر این می‌گیریم که شما یا تیم شما روی یک برنامه‌ی کاربردی یادگیری ماشین کار می‌کند و می‌خواهید به پیشرفت‌های سریع دست پیدا کنید. این کتاب به شما در انجام این کار کمک خواهد کرد.

مثال: استارت‌آپ ارائه‌ی عکس‌های گربه

فرض کنید شما در صدد راه‌اندازی استارت‌آپی هستید تا سیلی بی‌پایان از عکس‌های گربه‌ها را به دوستداران این حیوان خانگی عرضه کنید.



1- Machine Learning



مجموعه‌های آزمایشی و اعتبارسنجی باید چه قدر بزرگ باشند؟

مجموعه اعتبارسنجی باید به اندازه‌ای بزرگ باشد که بتوانید تفاوت الگوریتم‌های منتخب را تشخیص دهید. به‌عنوان مثال، اگر دسته‌بند A دارای مقدار درستی ۹۰٪ و دسته‌بند B دارای درستی ۹۰/۱٪ باشد، پس مجموعه اعتبارسنجی که حاوی ۱۰۰ نمونه است نمی‌تواند این تفاوت ۰/۱٪ را شناسایی کند. در مقایسه با سایر مسائل یادگیری ماشین که شاهد آن‌ها بوده‌ام، مجموعه اعتبارسنجی حاوی ۱۰۰ نمونه، مجموعه کوچکی قلمداد می‌شود. مجموعه‌های اعتبارسنجی در اندازه‌های ۱۰۰۰ تا ۱۰,۰۰۰ نمونه، متداول هستند. شما با برخورداری از ۱۰,۰۰۰ نمونه، شانس خوبی برای تشخیص بهبود ۰/۱٪ خواهید داشت.^۱

برای برنامه‌های کاربردی مهم و بالغ (به‌عنوان مثال در حوزه تبلیغات، جستجوی وب و پیشنهاد محصولات) با تیم‌هایی نیز برخورد داشته‌ام که حتی به تفاوت ۰/۱٪ بهبود هم بسیار رغبت نشان می‌دهند، زیرا این امر تأثیر مستقیم بر سوددهی شرکت دارد. در این مورد ممکن است مجموعه‌ای بسیار بزرگ‌تر ۱۰,۰۰۰ نمونه انتخاب کنید تا بتواند حتی کوچک‌ترین مقدار بهبود را تشخیص دهد.

اندازه مجموعه آزمایشی چه قدر باید باشد؟ این مجموعه باید به اندازه‌ای بزرگ باشد که اطمینان بالا را در مورد عملکرد کلی سیستم شما ایجاد کند. یک هیوریستیک رایج برای مقدار اندازه مجموعه آزمایشی، نسبت ۳۰٪ داده‌های شما را پیشنهاد می‌کند. این کار هنگامی به‌خوبی جواب می‌دهد که تعداد معدودی از نمونه‌ها را (مثلاً بین ۱۰۰ تا ۱۰,۰۰۰ مورد) در اختیار داشته باشید. اما حالا در عصر داده‌های بزرگ که با مسائل یادگیری ماشین مواجه هستیم و تعداد نمونه‌ها گاه به بیش از یک میلیارد می‌رسد، نسبت داده‌های اختصاص داده شده به مجموعه‌های آزمایشی و اعتبارسنجی کوچک شده است حتی هنگامی که تعداد نمونه‌ها در مجموعه‌های آزمایشی/اعتبارسنجی رو به افزایش بوده است. داشتن مجموعه‌های آزمایشی/اعتبارسنجی بسیار بزرگ که فراتر از نیازمندی‌های ارزیابی عملکرد الگوریتم‌های شما باشند ضرورتی ندارد.

۱- به لحاظ نظری نیز می‌توان عملکرد یک الگوریتم را روی مجموعه اعتبارسنجی از زوایه اختلاف معنادار آماری بررسی کرد. بیشتر تیم‌ها در عمل با این مسئله مواجه نیستند (مگر این‌که قصد انتشار مقاله‌های پژوهشی دانشگاهی را داشته باشند) و من معمولاً آزمایش‌های آماری معنادار را پیدا نمی‌کنم که برای سنجش پیشرفت‌های مقدماتی مفید باشند.



یک معیار ارزیابی تک عددی جهت بهینه‌سازی برای تیم خود وضع کنید

درستی دسته‌بند در واقع مثالی از یک معیار ارزیابی تک عددی است: شما دسته‌بند خود را روی مجموعه اعتبارسنجی (یا مجموعه آزمایشی) اجرا کرده و به یک عدد دست پیدا می‌کنید که نشان می‌دهد چه نسبتی از نمونه‌ها را به درستی دسته‌بندی کرده است. مطابق این معیار، اگر دسته‌بند A به درستی ۹۷٪ و دسته‌بند B به درستی ۹۰٪ دست پیدا کند، پس ما دسته‌بند A را به عنوان دسته‌بند برتر قلمداد می‌کنیم.

در مقابل، دقت^۱ و بازخوانی^۲ جزء معیارهای ارزیابی تک عددی نیستند: آن‌ها برای ارزیابی دسته‌بند شما دو عدد را ارائه می‌دهند. داشتن معیارهای ارزیابی چند عددی کار مقایسه الگوریتم‌ها را سخت‌تر می‌کند. فرض کنید الگوریتم‌های شما به این صورت عمل می‌کنند:

دسته‌بند	دقت	بازخوانی
A	۹۵٪	۹۰٪
B	۹۸٪	۸۵٪

در اینجا، هیچ کدام از این دو دسته‌بند برتری محسوسی نسبت به دیگری ندارد، بنابراین شما را بلافاصله برای انتخاب یکی از آن‌ها راهنمایی نمی‌کند.

دسته‌بند	دقت	بازخوانی	امتیاز F1
----------	-----	----------	-----------

1- Precision

۲- دقت دسته‌بند گره‌ها در واقع عبارت است از نسبت تصاویری که دسته‌بند شما به درستی آن‌ها را به عنوان گره شناسایی کرده است (داده‌های اعتبارسنجی یا آزمایشی). بازخوانی در واقع به درصدی از تمام تصاویر گره در مجموعه اعتبارسنجی (یا آزمایشی) اشاره دارد که دسته‌بند شما آن‌ها را به درستی به عنوان گره برچسب‌گذاری می‌کند. غالباً توازی میان دقت بالا و بازخوانی بالا وجود دارد.

برعکس، فرض کنید که شما معیار و مجموعه اعتبارسنجی خاصی ندارید. بنابراین هر بار که تیم شما دسته‌بند جدیدی برای شناسایی گره‌ها ایجاد می‌کند، شما باید آن را در برنامه کاربردی خود قرار داده و برای چند ساعت با برنامه کاربردی کار کنید تا متوجه شوید که آیا با دسته‌بند جدید پیشرفتی حاصل شده است یا خیر. این فرایند به طرز باورنکردنی به کندی پیش خواهد رفت. مضاف بر این، اگر تیم شما درستی دسته‌بند را از ۹۵٪ به ۹۵٫۱٪ بهبود دهد، ممکن است نتوانید این بهبود ۰٫۱٪ را با اجرای برنامه کاربردی تشخیص دهید. با این حال با انباشت تدریجی ده‌ها مورد از این بهبودهای ۰٫۱٪، سیستم شما به پیشرفت‌های زیادی خواهد رسید. داشتن معیار و مجموعه اعتبارسنجی به شما امکان می‌دهد تا با سرعت تشخیص دهید که کدام ایده‌ها بهبودهای کوچک یا بزرگی را با موفقیت برای شما رقم می‌زنند و در نتیجه امکان می‌دهد به سرعت تعیین کنید که کدام ایده‌ها را برای اصلاح نگه دارید یا کنار بگذارید.

زمان تغییر معیارها و مجموعه‌های اعتبارسنجی / آزمایشی

هنگامی که کار با پروژه جدیدی را شروع می‌کنیم، سعی ما بر این است تا به سرعت مجموعه‌های اعتبارسنجی/آزمایشی را انتخاب کنیم زیرا این کار هدف معینی را پیش روی تیم قرار می‌دهد.

ما معمولاً از تیم‌های خود می‌خواهیم تا ظرف کمتر از یک هفته (به ندرت طولانی‌تر از این مدت زمان) به سراغ مجموعه اعتبارسنجی یا آزمایشی اولیه و نیز معیار اولیه بروند. به جای تمرکز روی موارد کامل، بهتر است کار خود را با موارد ناکامل شروع کرده و کار را عملاً به دست بگیرید. ولی این دوره زمان‌بندی یک هفته‌ای در مورد برنامه‌های کاربردی بالغ کاربرد ندارد. برای مثال، برنامه شناسایی هرزنامه‌ها نوعی برنامه کاربردی بالغ یادگیری ژرف محسوب می‌شود. ما تیم‌هایی را دیده‌ایم که برای دستیابی به مجموعه‌های اعتبارسنجی/آزمایشی بهتر، ماه‌ها روی سیستم‌های بالغ کار کرده‌اند.

در صورتی که بعدها متوجه شدید که مجموعه اعتبارسنجی/آزمایشی یا معیار شما به نتیجه لازم دست پیدا نکردند، پس حتماً آن‌ها را به سرعت تغییر دهید. برای مثال، اگر مقدار معیار روی مجموعه اعتبارسنجی، دسته‌بند A را بالاتر از دسته‌بند B قرار می‌دهد، ولی تیم شما، جایگاه واقعاً بالاتری را برای دسته‌بند B قائل است، پس احتمالاً این نشانه‌ای است که می‌گوید شما باید مجموعه‌های اعتبارسنجی/آزمایشی یا معیار ارزیابی خود را تغییر دهید.

در مثال بالا به سه دلیل احتمالی عمده، دسته‌بند A به اشتباه بالاتر از دسته‌بند B رتبه‌بندی می‌شود؛ این دلایل عبارت‌اند از:

۱- توزیع واقعی که شما برای عملکرد مطلوب به آن نیاز دارید با توزیع مجموعه‌های اعتبارسنجی/آزمایشی متفاوت است.

فرض کنید مجموعه اعتبارسنجی/آزمایشی اولیه عمدتاً حاوی عکس‌های گربه‌هایی با سن بالا باشد. وقتی شما به برنامه کاربردی گربه خود مراجعه می‌کنید، متوجه می‌شوید که کاربران برخلاف انتظار شما، تصاویر بسیار بیشتری از بچه گربه‌ها را آپلود کرده‌اند. بنابراین توزیع مجموعه اعتبارسنجی/آزمایشی نماینده مناسبی

از توزیع واقعی داده‌هایی نیست که شما برای عملکرد مطلوب به آن نیاز دارید. در این مورد، مجموعه‌های اعتبارسنجی / آزمایشی خود را به‌روز کنید تا بیشتر معرف توزیع واقعی باشند.



۲- شما مجموعه اعتبارسنجی را بیش‌برازش کرده‌اید

فرایند ارزیابی مکرر ایده‌ها با کمک مجموعه اعتبارسنجی موجب می‌شود تا الگوریتم شما به‌تدریج در مجموعه اعتبارسنجی بیش‌برازش شود. پس از تکمیل این فرایند، سیستم خود را روی مجموعه آزمایشی ارزیابی خواهید کرد. اگر متوجه شدید که عملکرد مدل روی مجموعه اعتبارسنجی بسیار بهتر از عملکرد روی مجموعه آزمایشی است، نشان می‌دهد که با بیش‌برازش در مجموعه اعتبارسنجی روبه‌رو شده‌اید. در این مورد، مجموعه اعتبارسنجی تازه‌ای را پیدا کنید.

اگر لازم است روند پیشرفت تیم ردیابی شود، می‌توانید سیستم خود را به‌طور منظم با کمک مجموعه آزمایشی نیز ارزیابی کنید (مثلاً یک بار در هفته یا در ماه). با این حال، از مجموعه آزمایشی برای تصمیم‌گیری در مورد الگوریتم از جمله برگشت به سیستم هفته گذشته استفاده نکنید. اگر این کار را انجام دهید، در مجموعه آزمایشی نیز با بیش‌برازش روبه‌رو خواهید شد و دیگر نمی‌توانید روی این مجموعه برای برآورد کاملاً بی‌طرفانه از عملکرد سیستم خود حساب کنید (اگر قصد داشته باشید مقالات تحقیقاتی خود را منتشر کرده یا شاید از این معیار برای اتخاذ تصمیمات مهم کسب‌وکار استفاده کنید، آنگاه شما به آن نیاز پیدا خواهید کرد).

۳- معیار، مواردی را اندازه‌گیری می‌کند که جز نیازمندی‌های پروژه برای بهینه‌سازی نیستند

فرض کنید معیار شما برای برنامه کاربردی گربه، درستی دسته‌بندی باشد. این معیار در حال حاضر، دسته‌بند A را بالاتر از دسته‌بند B ارزیابی و رتبه‌بندی می‌کند. ولی فرض کنید که هر دو الگوریتم را امتحان کرده و متوجه شدید که دسته‌بند A تصاویر پورنوگرافی را گاه‌وبی‌گاه به نادرست دسته‌بندی می‌کند. اگرچه درستی دسته‌بند A بیشتر است، ولی به هر حال، تأثیر بد ناشی از عملکرد سیستم روی این تصاویر، به معنای عملکرد غیرقابل قبول آن است. خوب شما در مواجهه با چنین شرایطی چه کار می‌کنید؟

معیار مزبور در این جا نمی‌تواند این واقعیت را تشخیص دهد که الگوریتم B برای محصول شما، حقیقتاً بهتر از الگوریتم A است. بنابراین نمی‌توانید به این معیار برای انتخاب بهترین الگوریتم اعتماد کنید. اکنون زمان تغییر معیارهای ارزیابی فرا رسیده است. برای مثال، می‌توانید معیار را به گونه‌ای تغییر دهید که در مواجهه با تصاویر ذکر شده، جریمه سنگینی را متحمل شود. توصیه اکید می‌کنم که معیار جدیدی را انتخاب کرده و از آن برای تعریف هدف جدید تیم استفاده کنید، تا این که کار را برای مدتی طولانی بدون داشتن معیار قابل اعتماد ادامه داده و با تغییر آن دوباره به انتخاب دستی از میان دسته‌بندها مبادرت کنید.

تغییر مجموعه‌های اعتبارسنجی/آزمایشی اولیه یا معیارهای ارزیابی در جریان انجام یک پروژه کاملاً رایج است. داشتن معیار و مجموعه اعتبارسنجی/آزمایشی اولیه، به شما در تکرار سریع کمک می‌کند. اگر متوجه شدید که مجموعه‌های اعتبارسنجی/آزمایشی یا معیار فعلی، دیگر تیم شما را به سمت مسیر درست هدایت نمی‌کنند، جای نگرانی ندارد. کافی است آن‌ها را تغییر داده و اطمینان حاصل کنید که تیم شما با مسیر جدید آشنا است.

نکات کلیدی در ایجاد مجموعه‌های آزمایشی و اعتبارسنجی

- ◀ مجموعه‌های اعتبارسنجی و آزمایشی را از توزیعی انتخاب کنید که منعکس‌کننده داده‌های موردانتظار شما برای آینده بوده و می‌خواهید روی آن‌ها کار کنید. این رویه ممکن است مشابه توزیع داده‌های آموزشی شما نباشد.
- ◀ در صورت امکان، مجموعه‌های اعتبارسنجی و آزمایشی را از یک توزیع انتخاب کنید.
- ◀ به منظور بهینه‌سازی، یک معیار ارزیابی تک عددی را برای تیم خود انتخاب کنید. اگر هدف‌های چندگانه‌ای مدنظر شما هستند، پس آن‌ها را ترکیب و به فرمولی واحد تبدیل کرده (نظیر محاسبه میانگین معیارهای خطای چندگانه) یا معیارهای رضایت‌مندی و بهینه‌سازی را تعریف کنید.
- ◀ یادگیری ماشین فرایندی به شدت تکرارشونده است: یعنی شما ممکن است بیش از ده‌ها ایده را قبل از جلب رضایت خود آزمایش کنید.
- ◀ داشتن مجموعه‌های اعتبارسنجی و آزمایشی و نیز معیار ارزیابی تک عددی به شما در ارزیابی سریع الگوریتم‌ها و بنابراین تکرار سریع‌تر ایده‌ها کمک می‌کند.
- ◀ هنگام شروع کار روی برنامه کاربردی جدید تلاش کنید مجموعه‌های اعتبارسنجی و آزمایشی و نیز معیار را ظرف کمتر از یک هفته ایجاد کنید. صرف زمان بیشتر روی برنامه‌های کاربردی بالغ می‌تواند اقدام مطلوبی باشد.
- ◀ هیوریستیک قدیمی افزایش داده‌ها به دو بخش آزمایشی و آموزشی به نسبت ۳۰ و ۷۰ درصد برای حل مشکلاتی که شما در آن‌ها، با داده‌های زیادی روبه‌رو هستید، کاربردی نیست. در اینجا مجموعه‌های اعتبارسنجی و آزمایشی ممکن است بسیار کمتر از ۳۰٪ داده‌ها باشند.
- ◀ مجموعه اعتبارسنجی شما باید به اندازه‌ای بزرگ باشد که بتواند تغییرات معنادار در خصوص درستی الگوریتم را تشخیص دهد، ولی لزومی ندارد از این مقدار بسیار بزرگ‌تر باشد. مجموعه آزمایشی شما باید به اندازه‌ای بزرگ باشد تا برآورد مطمئنی از عملکرد نهایی سیستم ارائه کند.

همان‌طور که در بالا ذکر شد، فرض کنید الگوریتم شما ۱۶٪ خطا (درستی ۸۴٪) در مجموعه اعتبارسنجی دارد. ما این خطای ۱۶٪ را به دو مؤلفه تقسیم می‌کنیم:

- ◀ مؤلفه اول، نرخ خطای الگوریتم در مجموعه آموزشی. در این مثال، نرخ خطا ۱۵٪ است. ما آن را به‌طور غیررسمی به‌عنوان بایاس الگوریتم قلمداد می‌کنیم.
- ◀ مؤلفه دوم، عملکرد الگوریتم در مجموعه اعتبارسنجی (یا آزمایشی) چه‌قدر بدتر از مجموعه آموزشی است. در این مثال، اختلاف عملکرد الگوریتم در مجموعه اعتبارسنجی نسبت به مجموعه آموزشی برابر با ۱٪ است. ما آن را به‌طور غیررسمی واریانس الگوریتم قلمداد می‌کنیم.^۱

برخی تغییرات در الگوریتم یادگیری می‌تواند به مؤلفه اول خطا یعنی بایاس بپردازد و عملکرد الگوریتم را در مجموعه آموزشی بهبود دهد. برخی تغییرات به مؤلفه دوم یعنی واریانس می‌پردازد و به تعمیم بهتر آن از مجموعه آموزشی به مجموعه‌های آزمایشی/اعتبارسنجی کمک می‌کند.^۲ برای انتخاب امیدوارکننده‌ترین تغییرات، بسیار مفید خواهد بود که بدانیم پرداختن به کدام یک از این دو مؤلفه خطا از اهمیت بالاتری برخوردار است.

ایجاد بینش خوب در مورد بایاس و واریانس به شما کمک خواهد کرد تا تغییرات مؤثری را برای الگوریتم خود رقم بزنید.

۱- حوزه آمار دارای تعاریف رسمی‌تری از بایاس و واریانس است که ما بابت آن‌ها نگرانی نخواهیم داشت. به‌طور کلی بایاس عبارت است از نرخ خطای الگوریتم شما در مجموعه آموزشی در مواقعی که یک مجموعه آموزشی بسیار بزرگ داشته باشید. واریانس عبارت است از این که عملکرد شما در مجموعه آزمایشی در مقایسه با مجموعه آموزشی در این محیط چقدر بدتر است. هنگامی که معیار خطای شما، میانگین مربعات خطا باشد، می‌توانید فرمول‌های بنویسید که این دو کمیت را مشخص می‌کنند و همچنین ثابت می‌شود که خطای کل از مجموع بایاس و واریانس بدست می‌آید. اما در مورد اهداف تصمیم‌گیری ما راجع به نحوه پیشبرد کار در مسئله یادگیری ماشین، تعریف غیررسمی‌تری که اینجا از بایاس و واریانس ارائه شده کفایت خواهد کرد.

۲- همچنین روش‌هایی نیز وجود دارد که به‌طور همزمان با ایجاد تغییرات اساسی در معماری سیستم، می‌تواند هر دو خطای بایاس و واریانس را کاهش دهد. البته شناسایی و پیاده‌سازی آن‌ها دشوارتر است.

مثال‌هایی از بایاس و واریانس

وظیفه ما برای دسته‌بندی گربه‌ها را تصور کنید. دسته‌بند «ایده‌ال» (مانند یک انسان) ممکن است عملکرد تقریباً کاملی در این کار داشته باشد.

فرض کنید الگوریتم شما به صورت زیر عمل می‌کند:

◀ خطای آموزش = ۱٪

◀ خطای اعتبارسنجی = ۱۱٪

خوب، مشکل چیست؟ ما با استفاده از تعاریف فصل قبل، بایاس را ۱٪ و واریانس را (۱۰٪=۱٪-۱۱٪) برآورد می‌کنیم. بنابراین، با مقدار واریانس بالایی روبه‌رو هستیم. دسته‌بند دارای خطای آموزش بسیار پایینی است، اما نمی‌تواند آن را به مجموعه اعتبارسنجی تعمیم دهد. اسم این را بیش‌برازش نیز می‌گویند.

حالا این مثال را در نظر بگیرید:

◀ خطای آموزشی = ۱۵٪

◀ خطای اعتبارسنجی = ۱۶٪

ما بایاس را ۱۵٪ و واریانس را ۱٪ برآورد می‌کنیم. این دسته‌بند مجموعه آموزشی را با کیفیت ضعیف و خطای ۱۵٪ برآزش می‌کند، اما خطای آن در مجموعه اعتبارسنجی کمی بالاتر از خطای آموزشی است. بنابراین این دسته‌بند دارای بایاس بالا و در عین حال واریانس پایین است. این شرایط با نام کم‌برازش شناخته می‌شود.

حالا این مثال را در نظر بگیرید:

◀ خطای آموزش = ۱۵٪

◀ خطای اعتبارسنجی = ۳۰٪



تکنیک‌های کاهش واریانس

اگر الگوریتم یادگیری شما با مسئله واریانس بالا مواجه است، می‌توانید تکنیک‌های زیر را امتحان کنید:

- ◀ **افزودن داده‌های آموزشی بیشتر:** این ساده‌ترین و قابل‌اعتمادترین روش برای پرداختن به واریانس است، البته تا زمانی که به داده‌های بسیار بیشتر و قدرت محاسباتی کافی برای پردازش داده‌ها دسترسی داشته باشید.
- ◀ **افزودن تنظیم (تنظیم L2، تنظیم L1، تکنیک دورریزی):** این تکنیک واریانس را کاهش و بایاس را افزایش می‌دهد.
- ◀ **افزودن توقف زودرس (یعنی الگوریتم کاهش گرادیان را پیش از موعد بر مبنای خطای مجموعه اعتبارسنجی متوقف کنید):** این روش واریانس را کاهش داده اما بایاس را افزایش می‌دهد. توقف زودرس بسیار شبیه روش‌های تنظیم رفتار می‌کند و برخی مؤلفان آن را یک روش تنظیم می‌نامند.
- ◀ **انتخاب ویژگی برای کاهش تعداد/نوع ویژگی‌های ورودی:** این روش ممکن است به حل مسائل واریانس کمک کرده اما ممکن است بایاس را نیز افزایش دهد. کاهش ناچیز تعداد ویژگی‌ها (مثلاً از ۱۰۰۰ ویژگی به ۹۰۰ ویژگی) بعید است که تأثیر زیادی روی بایاس داشته باشد. کاهش بسیار بالایی تعداد ویژگی‌ها (مثلاً کاهش ۱۰۰۰ ویژگی به ۱۰۰ ویژگی یعنی کاهش ده برابری)، تا زمانی که ویژگی‌های مفید زیادی را حذف نکنید، می‌تواند احتمال تأثیرگذاری را بیشتر کند. در یادگیری ژرف مدرن، هنگامی که با داده‌های زیادی روبه‌رو هستند، این انتخاب ویژگی توسط الگوریتم صورت می‌گیرد و به الگوریتم اجازه می‌دهیم تا از برخی ویژگی‌ها براساس نوع داده‌ها استفاده کند. با این حال، وقتی مجموعه آموزشی شما کوچک است، انتخاب ویژگی می‌تواند بسیار مفید باشد.
- ◀ **کاهش اندازه مدل (مانند کاهش تعداد نورون‌ها/لایه‌ها):** از این روش با احتیاط استفاده کنید. این روش ممکن است واریانس را کاهش داده ولی بایاس را افزایش دهد. با این حال، من این تکنیک را برای رفع مسئله واریانس توصیه نمی‌کنم. اضافه کردن تنظیم معمولاً عملکرد بهتری را برای

دسته‌بندی به دنبال دارد. مزیت کاهش اندازه مدل این است که هزینه‌های محاسباتی را کاهش می‌دهد و در نتیجه به آموزش مدل‌ها سرعت می‌بخشد. اگر سرعت بخشیدن به فرایند آموزش مدل مفید است، پس حتماً کاهش اندازه مدل را مدنظر داشته باشید. اما اگر هدف شما کاهش واریانس است و دغدغه‌ای بابت هزینه محاسباتی ندارید در عوض، اضافه کردن تنظیم را مدنظر داشته باشید.

در اینجا دو تاکتیک دیگر از فصل قبل در مورد بایاس تکرار می‌شود:

- ◀ **اصلاح ویژگی‌های ورودی براساس بینش‌های تحلیل خطا:** اغلب تحلیل خطا به شما الهام می‌دهد تا ویژگی‌های بیشتری را ایجاد کنید تا به الگوریتم در حذف دسته خاصی از خطاها کمک کند. این ویژگی‌های جدید می‌توانند تماماً بایاس و واریانس را کاهش دهند. از جنبه نظری، افزودن ویژگی‌های بیشتر ممکن است واریانس را افزایش دهد؛ اما اگر با چنین موردی روبه‌رو شدید، به سراغ تکنیک‌های تنظیم بروید که معمولاً از افزایش واریانس جلوگیری می‌کند.
- ◀ **معماری مدل را اصلاح کنید تا تناسب بیشتری با مسئله شما داشته باشد (مانند معماری شبکه عصبی):** این تکنیک می‌تواند به‌طور هم‌زمان بایاس و واریانس را تحت تأثیر قرار دهد.



ترسیم منحنی‌های یادگیری

فرض کنید مجموعه آموزشی بسیار کوچک شما مشتمل بر ۱۰۰ نمونه است. شما الگوریتم خود را با استفاده از یک زیرمجموعه تصادفی با تعداد ۱۰ نمونه، ۲۰ نمونه، ۳۰ نمونه و ... آموزش می‌دهید. در واقع تعداد نمونه‌های ۱۰ تا ۱۰۰ را تا رسیدن به عدد ۱۰۰ ادامه دهید. سپس از این زیرمجموعه‌های ۱۰ نمونه‌ای برای ترسیم منحنی یادگیری خود استفاده کنید. ممکن است متوجه شوید که منحنی در اندازه‌های کوچک‌تر مجموعه آموزشی، کمی نویزدار است (یعنی اینکه مقادیر، بالاتر یا پایین‌تر از حد انتظار هستند).

هنگامی که برای آموزش فقط از ۱۰ نمونه منتخب تصادفی استفاده می‌کنید، ممکن است بدشانس باشید و یک مجموعه آموزشی مشخصاً «بد» مانند مجموعه‌ای شامل نمونه‌هایی با برچسب اشتباه یا مبهم در اختیار شما قرار بگیرد؛ یا ممکن است خوش‌شانس باشید و با یک مجموعه آموزشی مشخصاً «خوب» برخورد کنید. داشتن مجموعه آموزشی کوچک به این معنا است که ممکن است خطاهای آموزشی و اعتبارسنجی به صورت تصادفی دچار نوسان شوند.

اگر برنامه کاربردی یادگیری ماشین شما به شدت به سمت یک کلاس یا دسته چولگی دارد (مانند وظیفه دسته‌بندی گربه‌ها که در آن نسبت نمونه‌های منفی بسیار بیشتر از نمونه‌های مثبت است) یا اگر این برنامه دارای تعداد بسیار زیادی کلاس است (مانند شناسایی ۱۰۰ گونه حیوان مختلف)، پس شانس انتخاب یک مجموعه آموزشی بد یا مجموعه‌ای که نماینده خوبی از داده‌ها نیست نیز افزایش پیدا می‌کند. به‌عنوان مثال، اگر ۸۰٪ از نمونه‌های شما متعلق به کلاس منفی ($y=0$) و تنها ۲۰٪ از آن‌ها متعلق به نمونه‌های مثبت ($y=1$) باشد، آنگاه این احتمال وجود دارد که یک مجموعه آموزشی منتخب ۱۰ عضوی از این مجموعه، تنها شامل نمونه‌های منفی باشد؛ بدین ترتیب برای الگوریتم بسیار دشوار است که بتواند روابط معناداری را از این داده‌ها استخراج کند.

اگر نویز موجود در منحنی آموزشی، مشاهده روندهای واقعی را دشوار کند، پس دو راه‌حل وجود دارد:

◀ به جای اینکه مدل خود را تنها با تعداد ۱۰ نمونه آموزش دهید، با استفاده از تکنیک نمونه‌گیری با جایگذاری^۱ چندین مجموعه آموزشی متفاوت (مثلاً ۳ تا ۱۰ مجموعه) را به صورت تصادفی از مجموعه داده‌های اصلی خود انتخاب کنید که هر یک دارای ۱۰ نمونه است. با کمک هر یک از این مجموعه داده‌ها مدل‌های متفاوتی را آموزش دهید و خطای مجموعه آموزشی و اعتبارسنجی مدل‌های حاصل را محاسبه کنید. میانگین خطای مربوط به هر دو مجموعه اعتبارسنجی و مجموعه آموزشی را محاسبه و ترسیم کنید.

◀ اگر مجموعه آموزشی شما به سمت یک کلاس چولگی دارد یا دارای کلاس‌های متعددی است، پس به جای انتخاب تصادفی ۱۰ نمونه، زیرمجموعه «متعادلی^۲» را تشکیل دهید. به عنوان مثال، می‌توانید اطمینان حاصل کنید که ۲۰ درصد از نمونه‌های منتخب مربوط به کلاس مثبت و ۸۰ درصد آن مربوط به کلاس منفی است. به طور کلی‌تر، می‌توانید اطمینان حاصل کنید که نسبت کلاس نمونه‌های موجود در داده‌های منتخب شما تا حد ممکن به نسبت کلی آن‌ها در مجموعه آموزشی اصلی نزدیک است.

من با هیچ کدام از این تکنیک‌ها مشکلی ندارم، مگر این که شما قبلاً منحنی‌های یادگیری را رسم کرده و به این نتیجه برسید که این منحنی‌ها برای مشاهده روندهای اصلی، بیش از حد نویزدار هستند. اگر مجموعه آموزشی شما بزرگ باشد (مثلاً شامل بیش از ۱۰,۰۰۰ مثال نمونه) و توزیع کلاس شما به سمت دسته یا کلاس خاصی چولگی نداشته باشد، احتمالاً به این تکنیک‌ها نیازی نخواهید داشت.

در نهایت، ممکن است رسم منحنی یادگیری از نظر محاسباتی پرهزینه باشد؛ به عنوان مثال، ممکن است شما مجبور باشید ده مدل را به همراه ۱,۰۰۰، سپس ۲,۰۰۰ و به همین منوال تا ۱۰,۰۰۰ نمونه آموزش دهید. آموزش مدل‌هایی با مجموعه داده‌های کوچک بسیار سریع‌تر از آموزش مدل‌هایی با مجموعه داده‌های بزرگ‌تر انجام می‌شود. بنابراین، به جای انتخاب یکسان اندازه‌های مجموعه آموزشی در مقیاس خطی مانند مثال بالا، شما می‌توانید مدل‌هایی با تعداد ۱,۰۰۰، ۲,۰۰۰، ۴,۰۰۰، ۶,۰۰۰ و ۱۰,۰۰۰ نمونه را آموزش دهید. این امر همچنان می‌تواند درک روشنی از روند منحنی‌های یادگیری به شما ارائه دهد. البته، این تکنیک تنها در صورتی کاربرد دارد که هزینه محاسباتی آموزش در همه مدل‌های اضافی چشمگیر باشد.

۱- نمونه‌گیری با جایگذاری در اینجا به این معنا است: شما به طور تصادفی ۱۰ نمونه متفاوت از ۱۰۰ نمونه را انتخاب می‌کنید تا اولین مجموعه آموزشی خود را شکل دهید. سپس برای تشکیل دومین مجموعه آموزشی، بدون توجه به این که برای اولین مجموعه آموزشی خود چه نمونه‌هایی را انتخاب کرده‌اید، دوباره ۱۰ نمونه را انتخاب می‌کنید. بنابراین این امکان وجود دارد که یک نمونه خاص در هر دو مجموعه آموزشی اول و دوم ظاهر شود. در مقابل، اگر از تکنیک نمونه‌گیری بدون جایگذاری استفاده کنید، مجموعه آموزشی دوم تنها از ۹۰ نمونه‌ای انتخاب خواهد شد که اولین بار انتخاب نشده‌اند. در عمل، نتیجه استفاده از این دو تکنیک نباید تفاوت بزرگی ایجاد کند، اما استفاده از روش اول رایج‌تر است.

2- Balanced



چگونه تصمیم می‌گیرید که آیا از تمام داده‌های خود استفاده کنید یا خیر

فرض کنید مجموعه آموزشی آشکارساز گریه شما شامل ۱۰,۰۰۰ تصویر بارگذاری شده توسط کاربر است. توزیع این داده‌ها با توزیع یک مجموعه مجزای آزمایشی/اعتبارسنجی یکسان است و بیان‌گر توزیعی هستند که شما برای عملکرد مطلوب به آن بها می‌دهید. شما ۲۰,۰۰۰ تصویر دیگر که از اینترنت دانلود کرده‌اید نیز دارید. آیا شما باید از تمام این ۳۰,۰۰۰ تصویر به‌عنوان داده‌های آموزشی برای الگوریتم یادگیری خود استفاده کنید یا این که آن ۲۰,۰۰۰ تصویر اینترنتی را به دلیل ترس از بایاس الگوریتم یادگیری کنار می‌گذارید؟

وقتی از نسل‌های قبل‌تر الگوریتم‌های یادگیری (مانند ویژگی‌های بینایی کامپیوتری که به صورت دستی طراحی شده و به دنبال آن یک دسته‌بند خطی ساده دیده می‌شود) استفاده می‌کردید، با این ریسک واقعی روبه‌رو بودید که ادغام هر دو نوع داده باعث بدتر شدن عملکرد شما می‌شود. بنابراین، برخی مهندسان به شما در مورد اضافه کردن ۲۰,۰۰۰ تصویر اینترنتی هشدار می‌دادند.

اما این خطر در عصر امروزی الگوریتم‌های یادگیری قدرتمند و انعطاف‌پذیر-مانند شبکه‌های عصبی بزرگ-به شدت کاهش پیدا کرده است. اگر بتوانید از عهده ساخت یک شبکه عصبی با تعداد کافی از واحدها و لایه‌های پنهان بر بیابید، می‌توانید با اطمینان ۲۰,۰۰۰ تصویر را به مجموعه آموزشی خود اضافه کنید. اضافه کردن این تصاویر احتمال افزایش سطح عملکرد شما را بالا می‌برد.

این مشاهده بر این واقعیت متکی است که ما با نگاهت‌هایی چون $X \rightarrow Y$ روبه‌رو هستیم که در هر دو نوع داده به‌خوبی جواب می‌دهد. به عبارت دیگر، سیستم‌هایی وجود دارند که یک تصویر اینترنتی یا یک تصویر برنامه کاربردی موبایل را به‌عنوان ورودی پذیرفته و حتی بدون دانستن منبع تصویر، برچسب آن را با اطمینان پیش‌بینی می‌کند.

اضافه کردن ۲۰,۰۰۰ تصویر اضافی اثرات زیر را به دنبال دارد:

۱- شبکه عصبی شما با تعداد بیشتری از نمونه روبه‌رو می‌شود که در آن نشان داده می‌شود که گربه‌ها به چیزی شبیه هستند و به چه چیزی شبیه نیستند. این کار مفید است، زیرا تصاویر اینترنتی و تصاویر برنامه موبایل بارگذاری شده توسط کاربر به‌طور قطع با یکدیگر وجه اشتراک‌هایی دارند. شبکه عصبی شما می‌تواند بخشی از دانش حاصل از تصاویر اینترنتی را در تصاویر برنامه کاربردی موبایل اعمال کند.

۲- شبکه عصبی را وادار می‌کند تا بخشی از ظرفیت خود را برای آشنایی با خصوصیات مختص تصاویر اینترنتی توسعه دهد (مانند وضوح بالاتر، توزیع‌های مختلف در مورد شیوه چارچوب‌بندی تصاویر و غیره). اگر این خصوصیات تفاوت فاحشی با تصاویر برنامه کاربردی موبایل داشته باشند، بخشی از ظرفیت شبکه عصبی را به خود تخصیص خواهند داد. بدین ترتیب ظرفیت کمتری برای تشخیص داده‌های حاصل از توزیع تصاویر برنامه کاربردی موبایل وجود خواهد داشت، داده‌هایی که شما واقعاً به آن‌ها اهمیت می‌دهید. از جنبه نظری، این ممکن است به عملکرد الگوریتم‌های شما آسیب برساند.

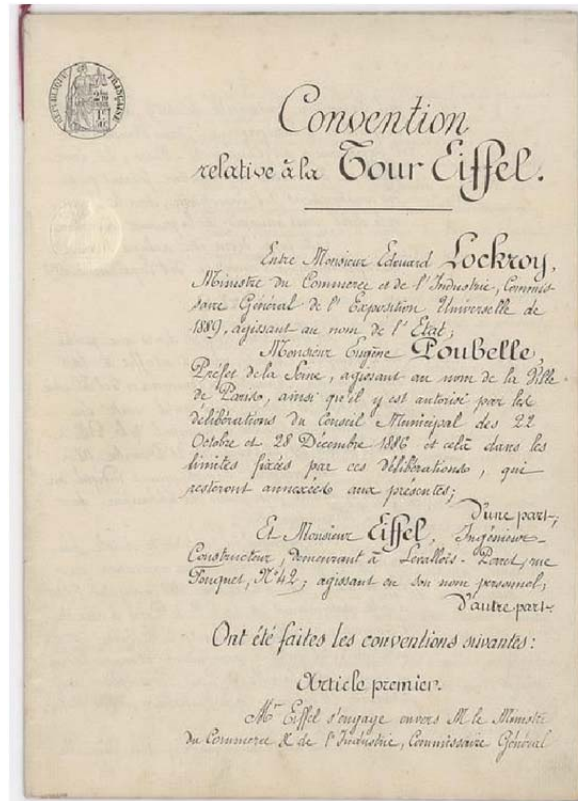
ما برای تشریح تأثیر دوم به زبانی دیگر می‌توانیم به شخصیت داستانی شرلوک هولمز گریزی بزنیم که می‌گوید مغز شما مانند یک اتاق زیرشیروانی با فضای بسیار محدود است. او می‌گوید: «شما به ازای هر دانش اضافی، چیزی را فراموش می‌کنید که قبلاً می‌دانستید. بنابراین، نبود حقایق غیرمفید که حقایق مفید را به حاشیه می‌راند از همه چیز مهم‌تر است.»^۱

خوشبختانه، اگر شما ظرفیت محاسباتی لازم برای ساخت یک شبکه عصبی بسیار بزرگ را دارید (مثل یک اتاق زیر شیروانی بزرگ)، پس جای نگرانی نیست. شما ظرفیت کافی برای یادگیری چه به کمک داده‌های اینترنتی و چه به کمک تصاویر برنامه کاربردی موبایل را در اختیار دارید، بدون اینکه دو نوع داده موجود، برای کسب ظرفیت با هم رقابت کنند. «مغز» الگوریتم شما به قدری بزرگ هست که لازم نیست نگران تمام شدن فضای زیر شیروانی باشید.

اما اگر شما یک شبکه عصبی بزرگ ندارید (یا یک الگوریتم یادگیری بسیار انعطاف‌پذیر دیگر را در اختیار ندارید)، پس باید توجه بیشتری به داده‌های آموزشی خود کنید تا با توزیع مجموعه‌های اعتبارسنجی و آزمایشی شما مطابقت داشته باشند.

۱- مطالعه در اسکارلت نوشته شده توسط آرتور کانن دوئل

اگر داده‌هایی را در اختیار دارید که کاملاً بی‌فایده است، پس باید آن‌ها را به دلایل محاسباتی کنار بگذارید. به‌عنوان مثال، فرض کنید مجموعه‌های آزمایشی/اعتبارسنجی شما عمدتاً شامل تصاویر غیررسمی از افراد، مکان‌ها، نشانه‌ها و حیوانات هستند. در ضمن فرض کنید مجموعه بزرگی از اسناد تاریخی اسکن شده را نیز در اختیار دارید:



این اسناد حاوی چیزهایی نیستند که به گربه شباهت داشته باشد. آن‌ها در ضمن کاملاً متفاوت از توزیع مجموعه اعتبارسنجی/آزمایشی شما به نظر می‌رسند. هیچ نشانه‌ای وجود ندارد که از این داده‌ها به‌عنوان نمونه‌های منفی استفاده کنید، زیرا فواید تأثیر اول شرح داده شده در بالا اندک است؛ تقریباً هیچ نکته‌ای در مورد این داده‌ها وجود ندارد که شبکه عصبی شما بتواند فرا بگیرد و در توزیع مجموعه آزمایشی/مجموعه اعتبارسنجی شما کاربرد داشته باشد. گنجاندن آن‌ها باعث هدر رفتن منابع محاسباتی و ظرفیت شبکه عصبی خواهد شد.



پیدایش یادگیری انتها به انتها

فرض کنید می‌خواهید سیستمی را برای تحلیل نظرات آنلاین کاربران در مورد محصولات بسازید تا به‌طور خودکار به شما بگوید که آیا نویسنده آن، محصول را دوست دارد یا خیر. برای مثال، شما امیدوار هستید که این سیستم نظر زیر را در میان نظرات بسیار مثبت قرار دهد:

This is a great mop!

و نظر زیر را در دسته نظرات بسیار منفی تشخیص دهد:

This mop is low quality—I regret buying it.

مسئله تشخیص نظرات مثبت در مقابل نظرات منفی را «دسته‌بندی احساسات»^۱ می‌نامند. برای ساخت این سیستم، می‌توانید مجرای شامل دو مؤلفه زیر را بسازید:

۱- تجزیه‌گر^۲: سیستمی است که متن را با اطلاعاتی حاشیه‌نویسی می‌کند که مهم‌ترین کلمات در آن شناسایی می‌شوند^۳. به‌عنوان مثال، شما ممکن است از این تجزیه‌گر برای برچسب‌گذاری تمام صفت‌ها و اسم‌ها استفاده کنید. بنابراین در مثال بالا با متن حاشیه‌نویسی شده زیر روبه‌رو خواهید شد:

This is a great_{Adjective} mop_{Noun}!

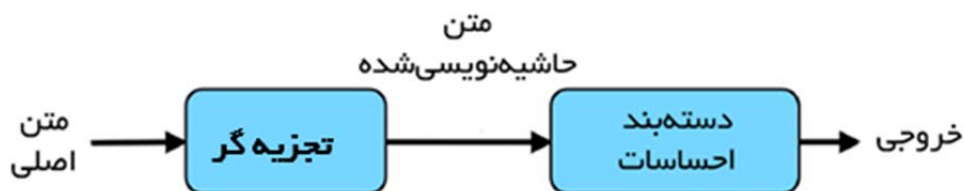
۲- دسته‌بند احساسات: نوعی الگوریتم یادگیری است که متن حاشیه‌نویسی شده را به‌عنوان ورودی دریافت کرده و احساسات نهفته در متن را پیش‌بینی می‌کند. حاشیه‌نویسی تجزیه‌گر می‌تواند کمک بزرگی به این الگوریتم یادگیری کند: با وزن‌دهی بیشتر به صفات، الگوریتم شما می‌تواند به‌سرعت روی کلمات مهمی مانند «great» تمرکز کند و کلمات کم‌اهمیت‌تر مانند «this» را نادیده بگیرد.

1- Sentiment Classification

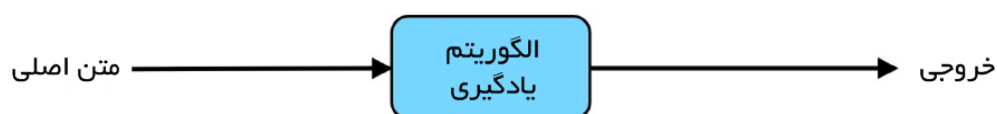
2- Parser

۳- یک تجزیه‌گر، حاشیه‌نویسی بسیار غنی‌تری را از متن ارائه می‌دهد، اما در اینجا این تشریح ساده برای توضیح یادگیری ژرف انتها به انتها کافی است.

این مجرا همراه با دو مؤلفه مزبور به صورت زیر به تصویر کشیده می‌شود:



اخيراً تمایل به جایگزینی چنین فرایندی با یک الگوریتم یادگیری واحد وجود دارد. الگوریتم یادگیری انتها به انتها^۱ برای این کار جمله "This is a great mop!" را به سادگی به عنوان ورودی متن اصلی و خام دریافت می‌کند و تلاش می‌کند احساسات را مستقیماً تشخیص دهد:



شبکه‌های عصبی معمولاً در سیستم‌های یادگیری انتها به انتها مورد استفاده قرار می‌گیرند. عبارت «انتها به انتها» به این واقعیت اشاره دارد که ما از الگوریتم یادگیری درخواست می‌کنیم که مستقیماً از ورودی به خروجی مطلوب برود؛ یعنی الگوریتم یادگیری مستقیماً «انتهای ورودی» سیستم را به «انتهای خروجی» متصل می‌کند.

در مواجهه با مسائلی که در آن‌ها، با داده‌های فراوان سروکار دارید، سیستم‌های انتها به انتها بسیار موفق بوده‌اند. اما آن‌ها همیشه انتخاب مناسبی نیستند. در فصل‌های بعدی مثال‌های بیشتری از سیستم‌های انتها به انتها را بررسی خواهیم کرد و شما را در مورد زمان استفاده یا عدم‌استفاده از آن‌ها راهنمایی می‌کنیم.

1- End-to-End Learning Algorithm

مثال‌های بیشتر از یادگیری انتها به انتها

فرض کنید می‌خواهید یک سیستم تشخیص گفتار بسازید. ممکن است برای انجام این کار سیستمی با سه مؤلفه زیر را در نظر داشته باشید:

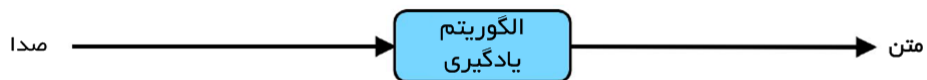


مؤلفه‌های مزبور این موارد را انجام می‌دهند:

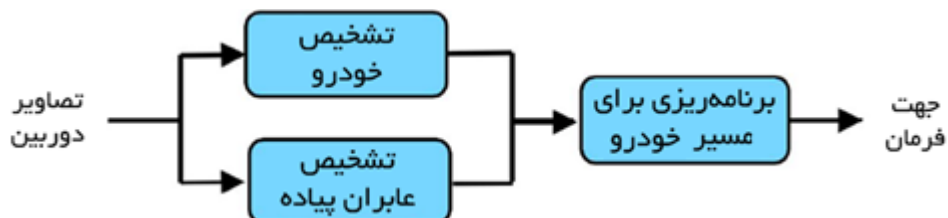
- ۱- محاسبه ویژگی‌ها: ویژگی‌های طراحی شده دستی مانند ویژگی‌های MFCC (ضرایب کپستروم فرکانسی مل) را استخراج می‌کند که با کمک آن محتوای یک گفتار ثبت می‌شود در حالی که خصوصیات کمتر مرتبط، مانند گام صدای گوینده را نادیده می‌گیرد.
- ۲- آشکارساز واج: برخی از زبان شناسان قائل به وجود واحدهای اساسی صدا به نام «واج‌ها» هستند. برای مثال، صدای آغازین "k" در کلمه "keep" همان واج "c" در کلمه "cake" است. این سیستم تلاش می‌کند واج‌های موجود در کلیپ صوتی را شناسایی کند.
- ۳- آشکارساز نهایی: در اینجا توالی واج‌های تشخیص داده شده دریافت و تلاش می‌شود به صورت یک متن پیوسته در خروجی قرار بگیرد.

در مقابل، یک سیستم انتها به انتها ممکن است یک کلیپ صوتی را به‌عنوان ورودی دریافت و سعی کند مستقیماً خروجی خود را تولید کند:

1- Mel-Frequency Cepstrum Coefficients
2- Phoneme



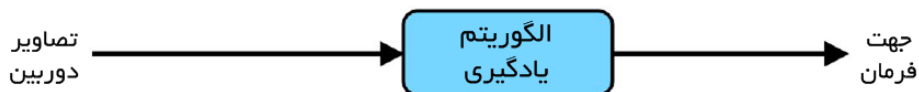
ما تاکنون فقط مجراهای یادگیری ماشین را توصیف کرده‌ایم که کاملاً خطی هستند: خروجی آن‌ها به ترتیب از یک مرحله به مرحله بعدی منتقل می‌شود. مجراها ممکن است پیچیده‌تر باشند. به عنوان مثال، در اینجا یک معماری ساده برای یک خودرو خودران ارائه شده است:



این سیستم سه مؤلفه دارد: یک مؤلفه خودروهای دیگر را با استفاده از تصاویر دوربین تشخیص می‌دهد. مؤلفه دیگر عابران پیاده را شناسایی می‌کند و سپس مؤلفه آخر مسیری را برای خودرو ما در نظر می‌گیرد که از برخورد با خودروها و عابران پیاده جلوگیری می‌کند.

نیازی به یادگیری همه مؤلفه‌های یک مجرا نیست. برای مثال، مستندات علمی مربوط به «برنامه‌ریزی حرکت ربات» شامل الگوریتم‌های متعدد برای مرحله طراحی مسیر نهایی خودرو می‌شوند. بسیاری از این الگوریتم‌ها مستلزم یادگیری نیستند.

در مقابل، رویکرد انتها به انتها ممکن است ورودی‌های خود را از حسگرها دریافت کرده و مستقیماً جهت فرمان را در خروجی خود داشته باشد:



به رغم این که یادگیری انتها به انتها موفقیت‌های زیادی دارد، ولی همیشه بهترین رویکرد نیست. برای مثال، تشخیص گفتار انتها به انتها به خوبی جواب می‌دهد. با این حال، من در مورد کارایی یادگیری انتها به انتها برای رانندگی خودکار تردید دارم. در چند فصل بعدی، دلیل آن را توضیح می‌دهم.

تشکیل تیم ابرقهرمان – هم‌تیمی‌های خود را به مطالعه این کتاب وادار کنید

تبریک می‌گوییم، این کتاب به پایان رسید!

در فصل دوم کتاب درباره اینکه چگونه این کتاب می‌تواند به شما کمک کند تا به قهرمان تیم خود بدل شوید صحبت کردیم.



بهتر از ابرقهرمان بودن، فقط این است که بخشی از یک تیم ابرقهرمان باشید. امیدوارم شما نسخه‌هایی از این کتاب را به دوستان و هم‌تیمی‌های خود بدهید و به خلق ابرقهرمانان دیگر کمک کنید!