یادگیری عمیق

دکتر امین گلزاری اسکوئی

a.golzari@azaruniv.ac.ir

a.golzari@tabrizu.ac.ir

https://github.com/Amin-Golzari-Oskouei

دانشگاه شهید مدنی آذربایجان

پاییز ۱۴۰۱

فصل ۴

اصول یادگیری ماشین

مطالب این فصل

- انواع دیگر یادگیری ماشین به غیر از دسته‌بندی و رگرسیون
- روال‌های ارزیابی رسمی برای مدل‌های یادگیری ماشین
- آماده‌سازی نمونه‌ها برای یادگیری عمیق
- مهندسی ویژگی
- نحوه مقابله با بیش‌برازش
- گردش کار کلی برای حل مسائل یادگیری عمیق

چهار شاخه یادگیری ماشین

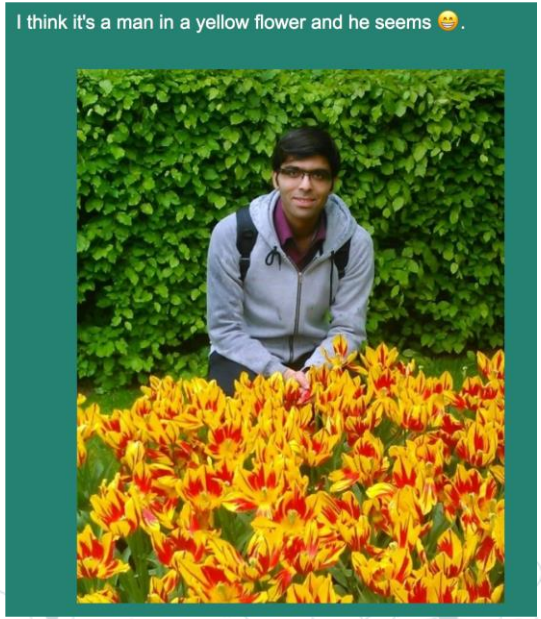
- یادگیری با ناظر
- یادگیری بدون ناظر
- یادگیری خودنظارتی
- یادگیری تقویتی

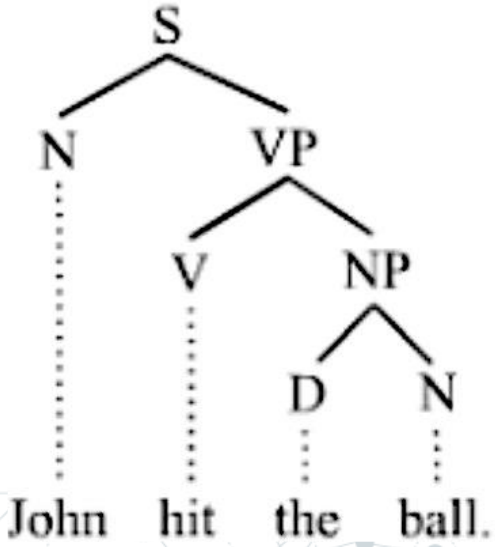
یادگیری با ناظر

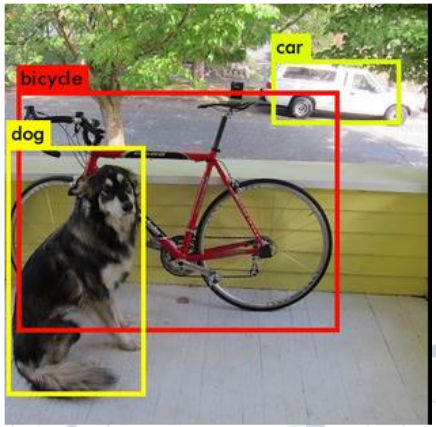
- ✓ یادگیری با ناظر متداول‌ترین نوع یادگیری است.
- ✓ تمامی چهار مثالی که در این کتاب با آن‌ها روبرو شده‌اید، مثال‌های متعارف یادگیری با ناظر هستند.
- ✓ نگاشت نمونه‌های ورودی به اهداف شناخته‌شده با استفاده از مجموعه‌ای از نمونه‌ها (که اغلب به وسیله انسان برچسب‌گذاری می‌شوند) است.
- ✓ دو دسته‌بندی مرسوم: دسته‌بندی و رگرسیون است.
- ✓ جمع‌آوری داده هزینه‌بر است.

یادگیری با ناظر

- ✓ مثال‌های خارق‌العاده از یادگیری با ناظر:
  - ✓ پیش‌بینی عنوان برای توصیف یک تصویر
  - ✓ تشخیص اشیا
  - ✓ پیش‌بینی درخت نحوی

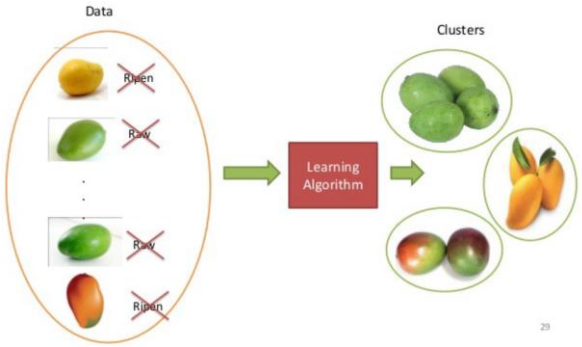




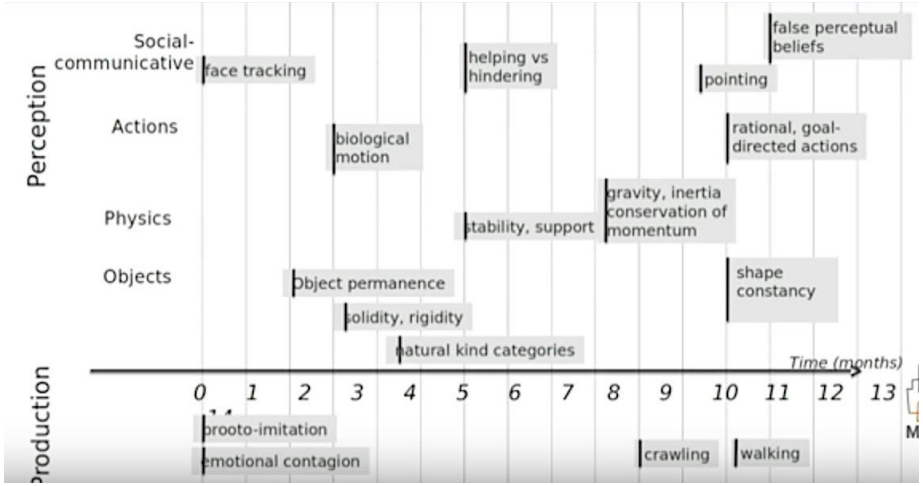


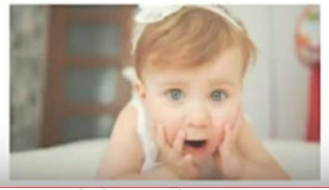
یادگیری بدون ناظر

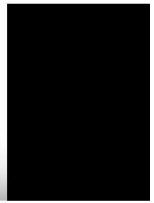
- ✓ یافتن تبدیل‌های جالب برای نمونه‌های ورودی بدون استفاده از برچسب نمونه‌ها.
- ✓ یادگیری بدون ناظر بخش اصلی تحلیل نمونه‌ها است و اغلب گامی ضروری برای درک بهتر مجموعه داده قبل از آغاز حل مسئله‌ی یادگیری با ناظر است.
- ✓ بسیاری از محققان مانند Yann Lecun پیش‌بینی می‌کنند که مرز بعدی هوش مصنوعی در یادگیری بدون نظارت است.



یادگیری بدون ناظر



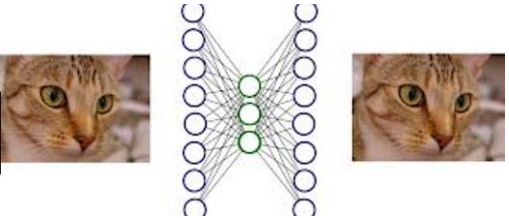




یادگیری خودنظارتی

یادگیری خودنظارتی نوعی یادگیری با ناظر بدون برچسب‌های ایجاد شده به وسیله انسان است. به عنوان مثال:

- خودکدگذارها نمونه معروفی از یادگیری خودنظارتی هستند که در آن‌ها اهداف تولید شده همان ورودی‌های بدون تغییر هستند.

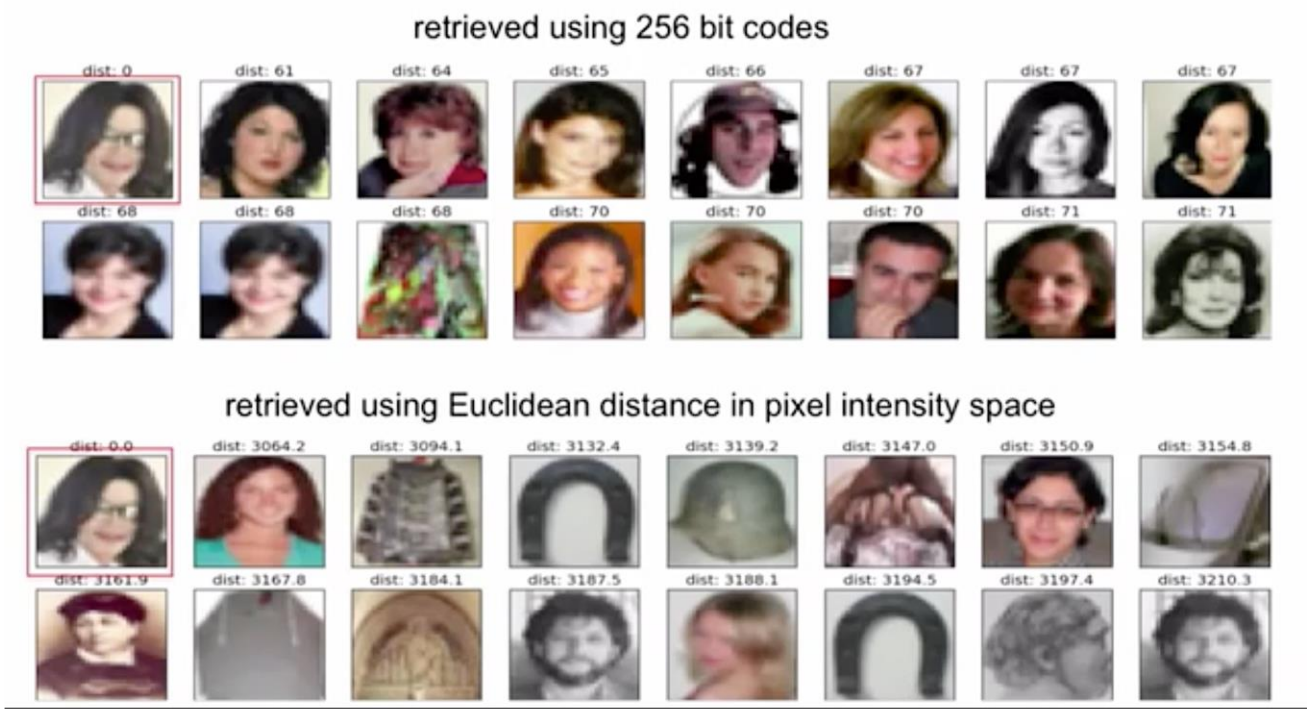


- پیش‌بینی فریم بعدی در یک ویدیو با استفاده از فریم‌های قبلی
- پیش‌بینی کلمه بعدی در یک متن با استفاده از کلمات قبلی

یادگیری خودنظارتی

- ✓ توجه داشته باشید که تفاوت بین یادگیری‌های با ناظر، خودنظارتی و بدون ناظر ممکن است برخی مواقع مبهم باشد (مرز کاملا مشخصی بین آنها قابل ترسیم نیست). یادگیری خودنظارتی را می‌توان هم به عنوان یادگیری با ناظر و هم به عنوان یادگیری بدون ناظر تعبیر و تفسیر نمود. تعبیر و تفسیر آن به در نظر گرفتن سازوکار یادگیری یا کاربرد آن بستگی دارد.

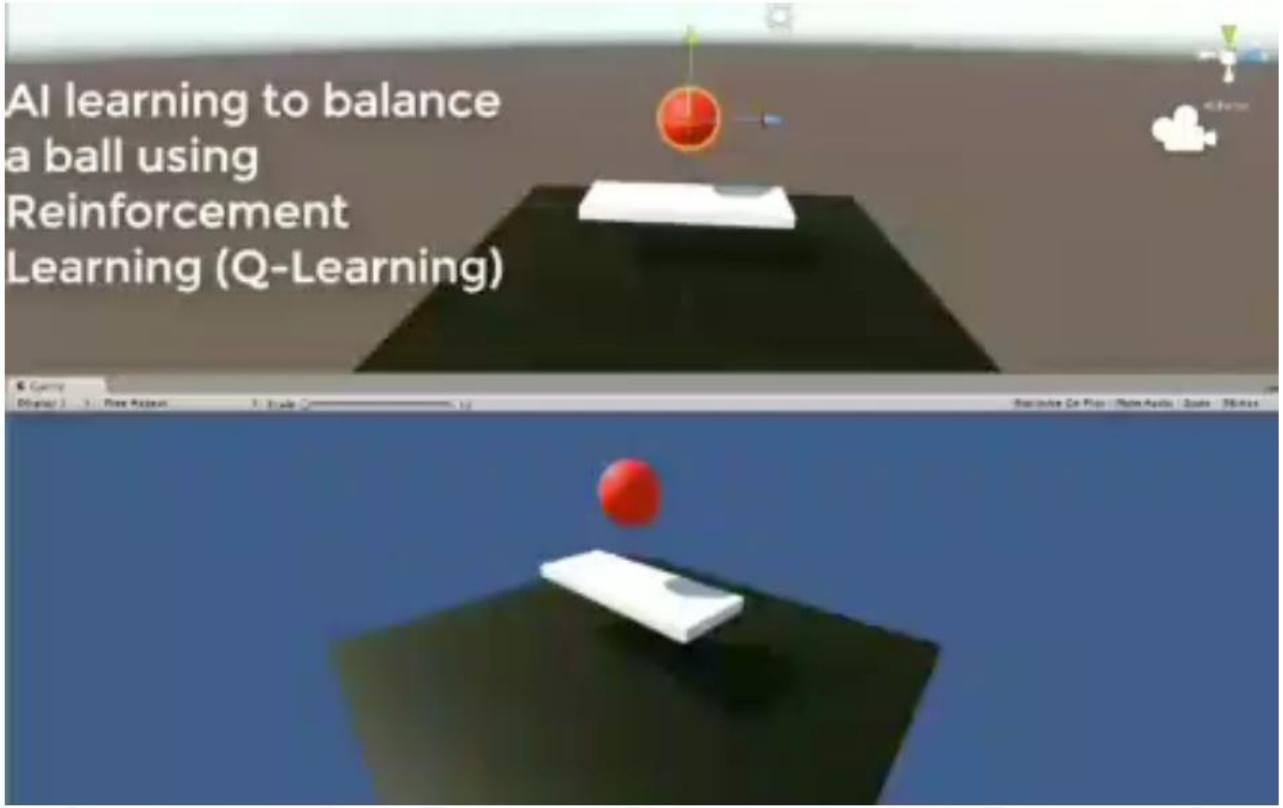
یادگیری خودنظارتی



یادگیری تقویتی

- ✓ این شاخه که برای مدت طولانی مهجور مانده بود، اخیراً بعد از موفقیت گوگل دیپ مایند در استفاده از آن برای یادگیری بازی‌های آتاری (و سپس یادگیری بازی گو در بالاترین سطح)، توجه بسیاری را به خود جلب کرده است.
- ✓ در یادگیری تقویتی، یک عامل بر اساس اطلاعات دریافتی از محیط، یاد می‌گیرد فعالیت‌هایی را انتخاب کند که بیشترین پاداش را برایش در پی داشته باشند.
- ✓ یادگیری تقویتی بیشتر یک حوزه پژوهشی است و هیچ موفقیت عملی را خارج از محدوده بازی‌ها کسب نکرده است.

یادگیری تقویتی



ارزیابی مدل‌های یادگیری ماشین

- در یادگیری ماشین، هدف به دست آوردن مدل‌هایی است که تعمیم‌پذیر هستند (روی نمونه‌هایی که هرگز ندیده‌اند خوب عمل می‌کنند) و بیش‌برازش مانع عمده آن است.

- ما تنها می‌توانیم آنچه را که اطلاعات آن را دریافت می‌کنیم کنترل کنیم، بنابراین سنجش قابلیت تعمیم مدل از اهمیت زیادی برخوردار است.

- ما عمدتاً به نحوه‌ی سنجش تعمیم مدل خواهیم پرداخت: چگونه باید مدل‌های یادگیری ماشین را ارزیابی کنیم.

ارزیابی مدل‌های یادگیری ماشین

ارزیابی مدل همواره به تقسیم نمونه‌ها به سه مجموعه خلاصه می‌شود: آموزشی، اعتبارسنجی و آزمایش. مدل را با نمونه‌های آموزشی، آموزش می‌دهیم و با نمونه‌های اعتبار سنجی، ارزیابی می‌کنیم. زمانی که مدل در بهترین حالت خود است، عملکردش را برای بار آخر روی نمونه‌های آزمایش، می‌سنجیم.

ارزیابی مدل‌های یادگیری ماشین

شاید بپرسید چرا نباید فقط دو مجموعه (مجموعه آموزش و مجموعه آزمایش) داشته باشیم؟

علت این است که توسعه یک مدل، همیشه دربرگیرنده تنظیم پیکربندی آن است: به عنوان مثال، انتخاب تعداد لایه‌ها و اندازه لایه‌ها (برای تمییز از پارامترها که وزن‌های شبکه هستند، ابر پارامترها نامیده می‌شوند). این تنظیم با استفاده از عملکرد مدل روی نمونه‌های اعتبارسنجی به عنوان سیگنال بازخورد صورت می‌گیرد. در حقیقت، این تنظیم، نوعی یادگیری است: یعنی جستجوی پیکربندی خوب در فضای پارامتر.

درنتیجه، تنظیم پیکربندی مدل بر اساس عملکرد آن روی مجموعه اعتبارسنجی می‌تواند به سرعت به بیش‌برازش مدل روی مجموعه اعتبارسنجی (نشت اطلاعات) منتهی شود، حتی اگر مدل هرگز مستقیماً روی آن آموزش ندیده باشد.

ارزیابی مدل‌های یادگیری ماشین

در پایان، با مدلی روبرو خواهید بود که به طور غیر واقعی روی نمونه‌های اعتبارسنجی خوب عمل می‌کند، چراکه شما آن را به این صورت ارتقا داده و اصلاح‌کرده‌اید. اگر به غیر از نمونه‌های اعتبارسنجی، به عملکرد مدل روی نمونه‌های کاملاً جدید اهمیت می‌دهید، پس باید برای ارزیابی مدل از مجموعه داده‌ای کاملاً متفاوت، یعنی نمونه‌های آزمایش استفاده کنید که مدل هرگز ندیده است. مدل، هرگز نباید به هیچ اطلاعاتی حتی به‌صورت غیرمستقیم درباره مجموعه آزمایش دسترسی داشته باشد.

ارزیابی مدل‌های یادگیری ماشین

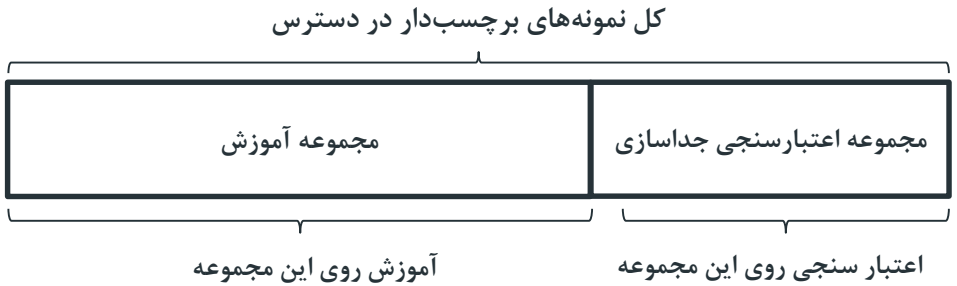
سه روش شناخته شده ارزیابی عبارتند از:

- اعتبارسنجی به روش جداسازی ساده
- اعتبارسنجی K بخشی:
- اعتبارسنجی مکرر K بخشی با درهم سازی

ارزیابی مدل‌های یادگیری ماشین

اعتبارسنجی به روش جداسازی ساده:

یک بخش از نمونه‌ها را به عنوان مجموعه آزمایش جدا کنید. مدل را با استفاده از باقیمانده نمونه‌ها آموزش داده و روی مجموعه آزمایش ارزیابی کنید. همان‌طور که در بخش قبل مشاهده کردید، برای اجتناب از نشت اطلاعات نباید مدل را بر اساس مجموعه آزمایش تنظیم کنید، بنابراین نیاز به یک مجموعه اعتبارسنجی دارید.

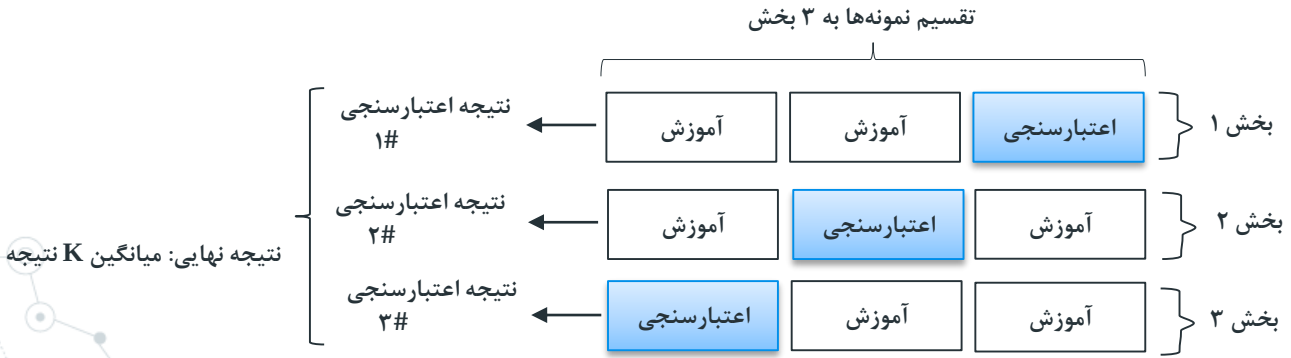


ارزیابی مدل‌های یادگیری ماشین

اعتبارسنجی K بخشی:

- در این رویکرد، نمونه‌ها را به K بخش با اندازه یکسان تقسیم کنید. برای هر بخش i، مدل را روی K-1 بخش باقیمانده آموزش داده و آن را روی بخش i اعتبارسنجی کنید. بدین ترتیب، نتیجه نهایی اعتبارسنجی برابر با میانگین نتایج به دست آمده از K اعتبارسنجی خواهد بود.

- این روش زمانی کارساز است که عملکرد مدل، واریانس قابل‌توجهی بر اساس تقسیم آموزش - آزمایش نشان دهد؛



ارزیابی مدل‌های یادگیری ماشین

اعتبارسنجی مکرر K بخشی با درهم سازی:

- این اعتبارسنجی برای مواقعی است که نمونه‌های کمی در اختیار دارید و لازم است که ارزیابی شما از مدل تا حد ممکن دقیق باشد.
- این روش عبارت است از اعمال اعتبارسنجی K بخشی به دفعات مکرر با درهم سازی هر باره نمونه‌ها قبل از تقسیم کردن به K بخش.

مواردی که باید به خاطر بسپارید

هنگام انتخاب روش اعتبارسنجی، به موارد زیر دقت کنید:

- **نماینده مناسبی از نمونه‌ها:** انتظار دارید که هر دو مجموعه آموزش و آزمایش نماینده نمونه‌های موجود باشند. به عنوان مثال، اگر هنگام دسته‌بندی تصاویر ارقام از آرایه‌ای از نمونه‌هایی شروع کنید که بر اساس کلاسشان مرتب شده‌اند، انتخاب 80 درصد آرایه به عنوان مجموعه آموزش و 20 درصد باقیمانده به عنوان مجموعه آزمایش موجب خواهد شد که مجموعه آموزش تنها شامل کلاس‌های صفر تا 7 بوده و مجموعه آزمایش دربرگیرنده کلاس‌های ۸ و ۹ باشد. به نظر اشتباه مضحکی می‌آید، اما به طور باورنکردنی‌ای متداول است. به همین دلیل، اغلب قبل از تقسیم نمونه‌ها به بخش‌های آموزش و آزمایش باید آن‌ها را به‌صورت تصادفی بُر بزنید.

مواردی که باید به خاطر بسپارید

هنگام انتخاب روش اعتبارسنجی، به موارد زیر دقت کنید:

- **ترتیب زمانی:** اگر می‌خواهید آینده را با نمونه‌های گذشته پیش‌بینی کنید (به عنوان مثال، هوای فردا، جابجایی سهام و غیره)، نباید قبل از تقسیم نمونه‌ها آن‌ها را به‌صورت تصادفی برهم بزنید، چراکه در این حالت نشت زمانی ایجاد خواهد شد: مدل با استفاده از نمونه‌های آینده آموزش خواهد دید. در چنین موقعیت‌هایی، باید همواره اطمینان حاصل کنید که نمونه‌های آزمایش به لحاظ زمانی پس از نمونه‌های آموزشی قرار داشته باشند.

مواردی که باید به خاطر بسپارید

هنگام انتخاب روش اعتبارسنجی، به موارد زیر دقت کنید:

- **تکرار نمونه‌ها:** اگر برخی از نمونه‌ها چندین بار دیده می‌شوند (در نمونه‌های دنیای واقعی کاملاً متداول است)، بُر زدن نمونه‌ها و تقسیم آن‌ها به مجموعه آموزش و اعتبارسنجی موجب تکرار نمونه‌ها بین مجموعه‌های آموزشی و اعتبارسنجی خواهد شد. در عمل، در حال آزمایش روی نمونه‌های آموزشی خواهید بود که بدترین کار ممکن است! لذا از عدم وجود نمونه مشترک میان مجموعه آموزش و مجموعه اعتبارسنجی اطمینان حاصل کنید.

پیش‌پردازش نمونه‌ها، مهندسی ویژگی و یادگیری ویژگی

- هدف از پیش‌پردازش نمونه‌ها، آماده سازی نمونه‌های خام برای شبکه‌های عصبی است. پیش‌پردازش این موارد را در برمی‌گیرد: بردارسازی، نرمال‌سازی، مدیریت مقادیر ناموجود و استخراج ویژگی.

- بسیاری از فنون پیش‌پردازش نمونه‌ها و مهندسی ویژگی حوزه محور هستند (به عنوان مثال، مختص نمونه‌های متنی یا نمونه‌های تصویری هستند).

پیش‌پردازش نمونه‌ها، مهندسی ویژگی و یادگیری ویژگی

بردارسازی: تمامی ورودی‌ها و اهداف در شبکه عصبی باید تنسورهایی از نوع ممیز شناور (در موارد خاص، تنسورهای اعداد صحیح) باشند.

- به عنوان مثال، در دو مثال پیشین دسته‌بندی متن، از بازنمایی متن به صورت لیست اعداد صحیح شروع کرده و از کدگذاری تک-یک برای تبدیلشان به تنسور داده float32 استفاده کردیم. در مثال‌های دسته‌بندی ارقام و پیش‌بینی قیمت مسکن، خود نمونه‌ها به‌صورت برداری بودند و نیازی به انجام این مرحله نبود.

نرمال‌سازی مقدار: بسیاری از فنون پیش‌پردازش نمونه‌ها و مهندسی ویژگی حوزه محور هستند (به عنوان مثال، مختص نمونه‌های متنی یا نمونه‌های تصویری هستند).

پیش‌پردازش نمونه‌ها، مهندسی ویژگی و یادگیری ویژگی

نرمال‌سازی مقدار: برای تسهیل یادگیری شبکه، نمونه‌ها باید دارای مشخصات زیر باشند:

- دارای مقادیر کوچک باشند – عموماً، اکثر مقادیر باید در دامنه 0-1 باشند.
- همگن باشند – یعنی، مقادیر تمامی ویژگی‌ها باید تقریباً در دامنه یکسانی قرار بگیرند.

علاوه بر این، با وجود اینکه روش نرمال‌سازی دقیق زیر همیشه ضروری نیست، ولی استفاده از آن متداول بوده و می‌تواند مفید واقع شود:

- هر ویژگی را به طور مستقل نرمال‌سازی کنید تا میانگین صفر داشته باشد.
- هر ویژگی را به طور مستقل نرمال‌سازی کنید تا انحراف معیار آن برابر با 1 باشد.

پیش‌پردازش نمونه‌ها، مهندسی ویژگی و یادگیری ویژگی

مدیریت مقادیر ناموجود: به طور کلی، در شبکه‌های عصبی، بهتر است مقادیر ناموجود به عنوان صفر وارد شوند؛ البته با این شرط که صفر یک مقدار معنادار نباشد.

توجه کنید که اگر برخی از مقادیر در میان نمونه‌های آزمایش ناموجود هستند، اما شبکه بدون آن‌ها آموزش دیده است، پس نادیده گرفتن آن‌ها را یاد نگرفته است! در چنین موقعیتی، باید به طور تصادفی نمونه‌های آموزشی با ورودی‌های ناموجود ایجاد کنید: برخی از نمونه‌های آموزشی را چند بار کپی کنید و برخی از ویژگی‌های آن‌ها را که فکر می‌کنید در نمونه‌های آزمایش موجود نیستند، حذف کنید.

پیش‌پردازش نمونه‌ها، مهندسی ویژگی و یادگیری ویژگی

مهندسی ویژگی: مهندسی ویژگی فرایند استفاده از دانش خودمان در مورد نمونه‌ها و الگوریتم موجود یادگیری ماشین (در این مورد، شبکه عصبی) به منظور کارایی بهتر الگوریتم است.

نمونه‌های خام: ماتریسی از پیکسل‌ها		
ویژگی‌های بهتر: مختصات عقربه‌های ساعت	{x1: 0.7, y1: 0.7} {x2: 0.5, y2: 0.0}	{x1: 0.0, y2: 1.0} {x2: -0.38, y2: 0.32}
ویژگی‌های بهتر از آن: زوایای عقربه‌های ساعت	theta1: 45, theta2: 0	theta1: 90, theta2: 140

پیش‌پردازش نمونه‌ها، مهندسی ویژگی و یادگیری ویژگی

قبل از یادگیری عمیق، مهندسی ویژگی نقشی حیاتی داشت، چراکه الگوریتم‌های سنتی به اندازه‌ای مجهز به فضای فرضیه غنی نبودند که ویژگی‌های مفید را خودشان یاد بگیرند و درنتیجه نحوه‌ی ارائه نمونه به الگوریتم برای موفقیت آن ضروری بود.

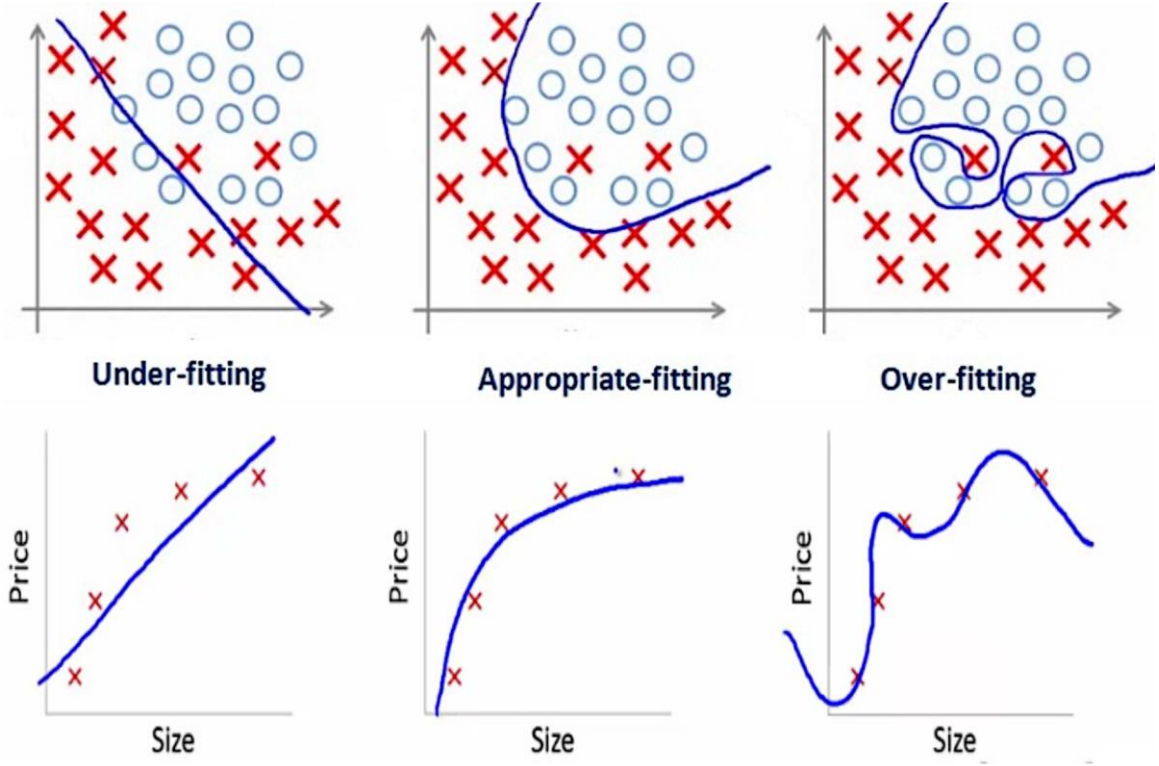
به عنوان مثال، قبل از اینکه شبکه‌های عصبی کانولوشنی در مسئله دسته‌بندی ارقام MNIST موفق عمل کنند، راه‌حل‌ها اساساً بر ویژگی‌هایی از قبیل تعداد حلقه‌ها در تصویر رقم، ارتفاع هر رقم در تصویر، هیستوگرام مقادیر پیکسل و غیره مبتنی بودند.

پیش‌پردازش نمونه‌ها، مهندسی ویژگی و یادگیری ویژگی

خوشبختانه، یادگیری عمیق در اکثر موارد نیاز به مهندسی ویژگی را مرتفع می‌سازد، چراکه شبکه‌های عصبی این توانایی را دارند که به طور خودکار ویژگی‌های مفید را از نمونه‌های خام استخراج کنند. آیا این بدین معنی است که استفاده از شبکه‌های عصبی عمیق نیاز به مهندسی ویژگی را مرتفع می‌سازد؟

- ویژگی‌های خوب این امکان را فراهم می‌سازند که مسائل را با استفاده از منابع کمتر و با ظرافت بیشتری حل کنید.
- ویژگی‌های خوب امکان حل مسئله با نمونه‌های بسیار کم را فراهم می‌آورند.

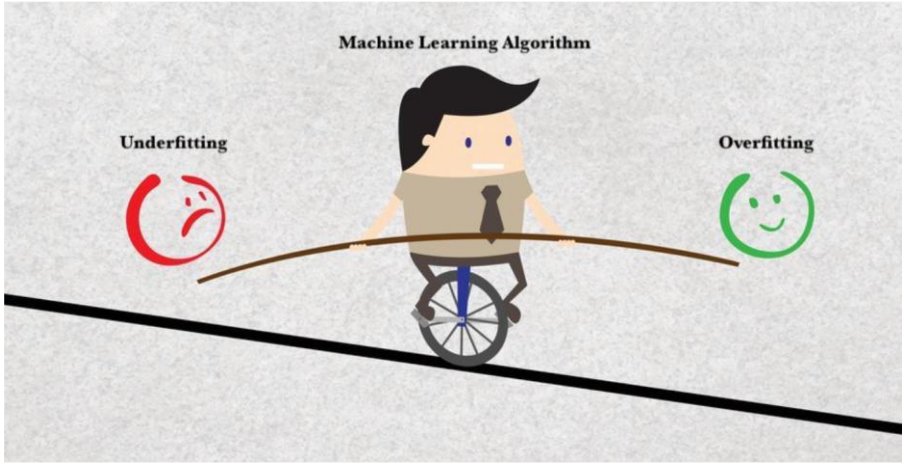
بیش‌برازش و کم‌برازش



Images courtesy of Andrew Ng

بیش‌برازش و کم‌برازش

مسئله اساسی در یادگیری ماشین ایجاد توازن بین بهینه‌سازی و تعمیم است. . بهینه‌سازی به فرایند تنظیم مدل به منظور رسیدن به بهترین عملکرد ممکن روی نمونه‌های آموزشی اطلاق می‌گردد. در حالی که تعمیم، نحوه عملکرد مدل آموزش دیده روی نمونه‌هایی است که هرگز ندیده است.



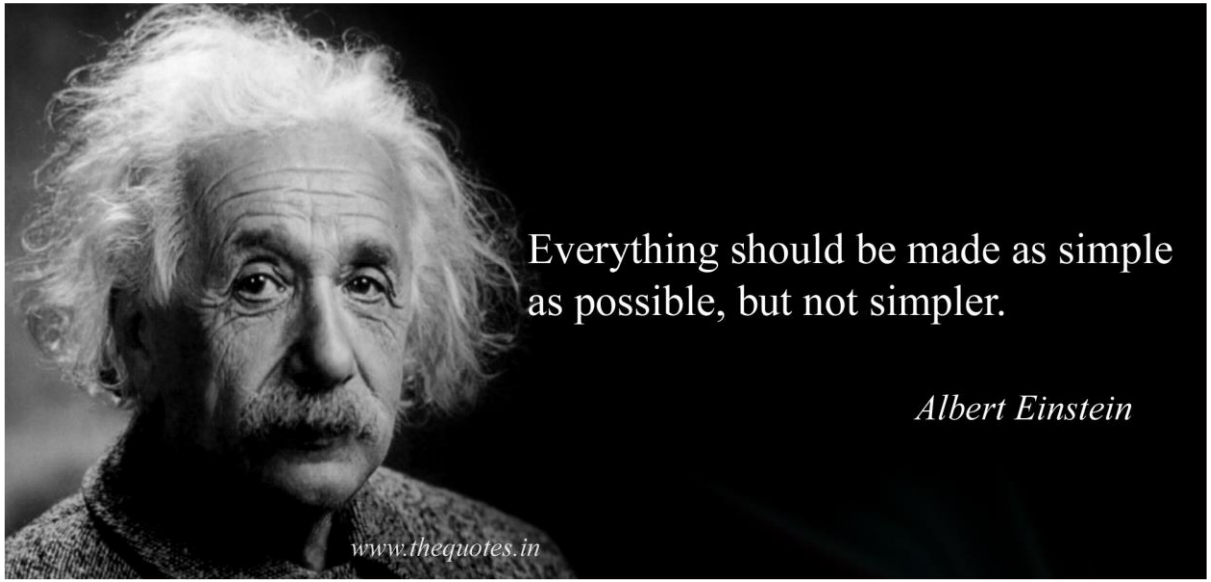
بیش‌برازش و کم‌برازش

دو الگو در یک مجموعه‌داده وجود دارد:
- الگوهای خاص این داده‌ها
- الگوهای کلی برای همه داده‌های این نوع

به عبارتی:
- کم‌برازش ناتوانی مدل در گرفتن الگوهای کلی است
- بیش‌برازش، گرفتن الگوهای خاص داده است

به عبارتی دیگر:
- بیش‌برازش شبیه به داشتن حالات بیش از حد ضروری است. به عنوان مثال، فنجان‌ها باید دسته داشته باشند.
- کم‌برازش شبیه به داشتن حالات کمتر از حد ضروری است. به عنوان مثال، فنجان‌ها نیازی به باز بودن در بالا ندارند.

بیش‌برازش و کم‌برازش



بیش‌برازش و کم‌برازش

- هدف به دست آوردن قدرت تعمیم بالا است، اما شما تعمیم را کنترل نمی‌کنید؛ فقط می‌توانید مدل را بر اساس نمونه‌های آموزشی تنظیم نمایید.

- بیش‌برازش به دلایل زیر رخ می‌دهد:
  - تعداد نمونه کم
  - مدل بیش از حد پیچیده

- مشکل اصلی در یادگری عمیق بیش‌برازش است.

بیش‌برازش و کم‌برازش

- بهترین راه‌حل برای جلوگیری از یادگیری الگوهای نامربوط یا گمراه‌کننده به وسیله مدل، فراهم کردن نمونه‌های آموزشی بیشتر است. مدلی که روی نمونه‌های بیشتری آموزش دیده باشد، طبیعتاً قابلیت تعمیم بهتری خواهد داشت.

- زمانی که امکان استفاده از نمونه‌های زیاد وجود نداشته باشد، بهترین راه‌حل، تعدیل میزان اطلاعات ذخیره‌شده به وسیله مدل و یا ایجاد محدودیت در مورد اطلاعاتی است که مدل می‌تواند ذخیره نماید. اگر شبکه‌ای قابلیت یادگیری تعداد کمی الگو را داشته باشد، فرایند بهینه‌سازی آن را وادار به تمرکز روی برجسته‌ترین الگوها خواهد نمود که شانس بهتری برای تعمیم دارند.

- فرایند مقابله با بیش‌برازش با این شیوه‌ها، تنظیم نامیده می‌شود.

بیش‌برازش و کم‌برازش

- برخی از متداول‌ترین فنون مقابله با بیش برازش (تنظیم):
  - کاهش اندازه شبکه
  - اضافه نمودن تنظیم وزن
  - حذف تصادفی

- ممکن است با اصل تیغ اوکام آشنا باشید: اگر دو توضیح برای یک مسئله مشخص داده شده باشد، به احتمال زیاد ساده‌ترین توضیح صحیح است.

کاهش اندازه شبکه

- کاهش تعداد پارامترهای قابل یادگیری در مدل (که تعداد لایه‌ها و تعداد واحدها در هر لایه آن را معین می‌سازد).
- در یادگیری عمیق، تعداد پارامترهای قابل یادگیری در مدل، اغلب به عنوان ظرفیت یاد می‌شود.
- باید بین ظرفیت بیش ازحد و ظرفیت ناکافی موازنه ایجاد شود.
- متأسفانه، برای تعیین تعداد مناسب لایه‌ها یا اندازه مناسب برای هر لایه هیچ فرمول جادویی وجود ندارد.

اضافه نمودن تنظیم وزن

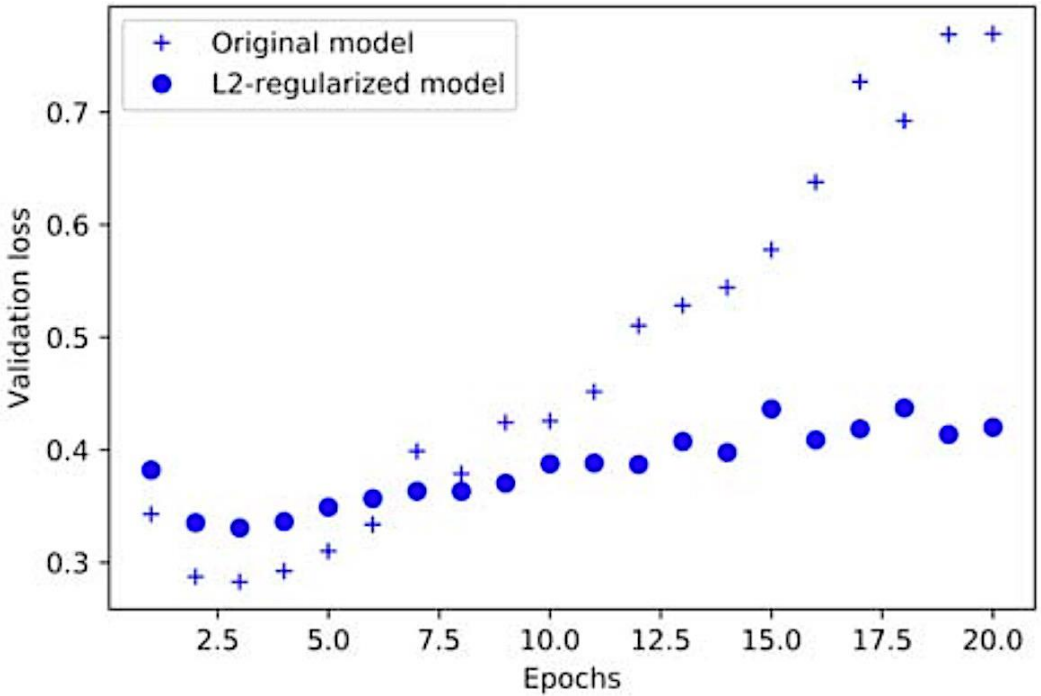
در مورد مدل‌هایی که شبکه‌های عصبی یاد می‌گیرند اصل تیغ اوکام نیز صادق است: بیش‌برازش در مدل‌های ساده‌تر (با وزن‌های کوچک) بسیار کم‌تر از مدل‌های پیچیده است.

یکی از شیوه‌های معمول کنترل بیش‌برازش، محدود کردن پیچیدگی شبکه با محدود کردن وزن‌های آن به مقادیر کوچک است که موجب توزیع منظم‌تر مقادیر وزن می‌شود. این عمل تنظیم وزن نامیده می‌شود.

تنظیم وزن با اضافه کردن هزینه‌ای به تابع خطا شبکه انجام می‌شود که با داشتن وزن‌های بزرگ همراه است:

- تنظیم L1: هزینه اضافه شده متناسب با مقدار مطلق مقادیر وزن‌ها (Loss+ λ∑|W|)
- تنظیم L2: هزینه اضافه شده متناسب با مربع مقادیر وزن‌ها (Loss+ λ∑W^2)

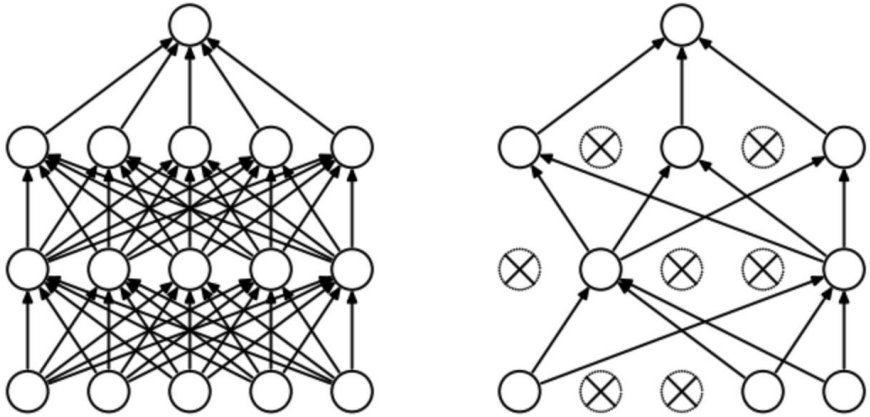
اضافه نمودن تنظیم وزن



اضافه نمودن حذف تصادفی

اعمال حذف تصادفی روی یک لایه عبارت از کنار گذاشتن تصادفی (صفر کردن) تعدادی از ویژگی‌های خروجی لایه در طول آموزش است.

نرخ حذف تصادفی برابر با کسر ویژگی‌هایی است که صفر شده‌اند و اغلب مقداری بین 0.2 و 0.5 دارد.



(a) Standard Neural Net

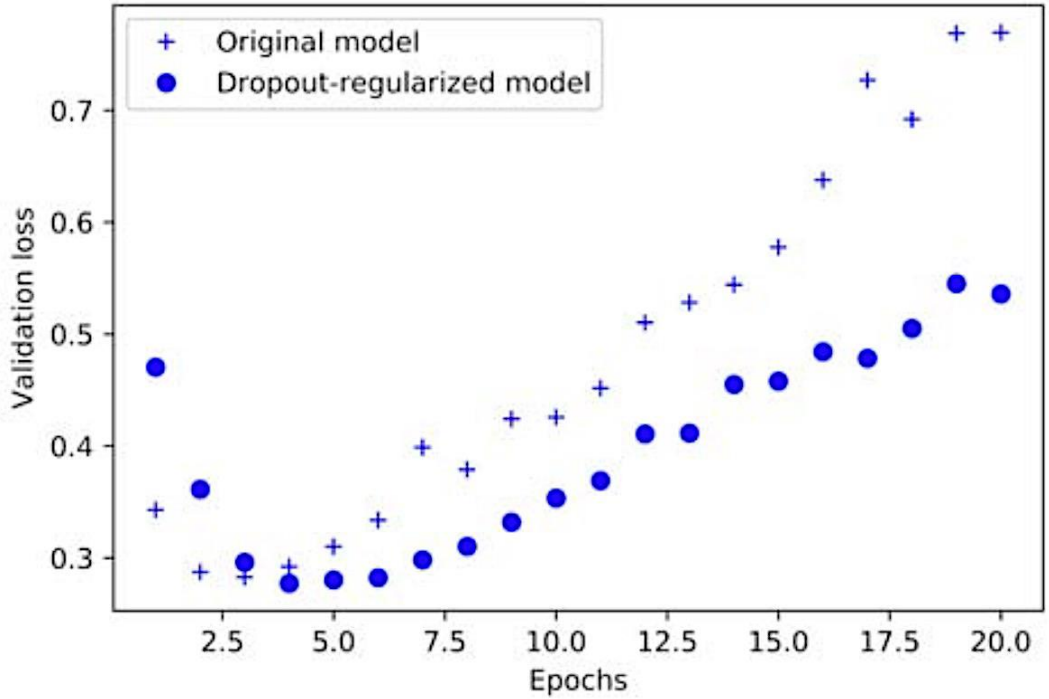
(b) After applying dropout.

اضافه نمودن تنظیم وزن

ممکن است این روش عجیب و بی‌اساس به نظر برسد. چرا این کار به کاهش بیش‌برازش کمک می‌کند؟

هینتون می‌گوید آنچه بیشتر از همه برایم الهام‌بخش بود سازوکار ممانعت از کلاه‌برداری بانک‌ها بود.

اضافه نمودن حذف تصادفی



روال جامع یادگیری ماشین

1- تعریف مسئله و فراهم نمودن مجموعه داده

2- انتخاب معیار موفقیت

3- تصمیم‌گیری در مورد روش ارزیابی

4- آماده‌سازی نمونه‌ها

5- توسعه مدلی که بهتر از یک روش پایه عمل می‌کند

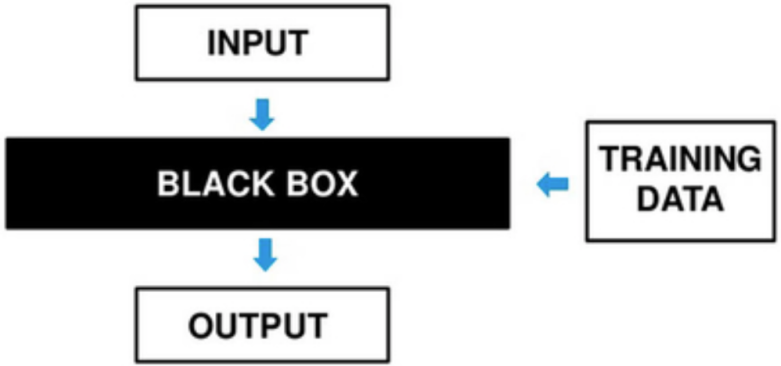
6- بالا بردن قدرت مدل: توسعه مدلی که بیش‌برازش دارد

7- تنظیم مدل و میزان کردن ابرپارامترها

تعریف مسئله و فراهم نمودن مجموعه داده

- نمونه‌های ورودی شما چه چیزهایی خواهند بود؟
- سعی دارید چه چیزی را پیش‌بینی کنید؟
- با چه نوع مسئله‌ای مواجه هستیم؟ (دسته‌بندی دودویی، رگرسیون، دسته‌بندی چند کلاسی، چند برچسبی)

شناسایی نوع مسئله، راهنمای انتخاب معماری مدل، تابع هزینه و غیره است.



تعریف مسئله و فراهم نمودن مجموعه داده

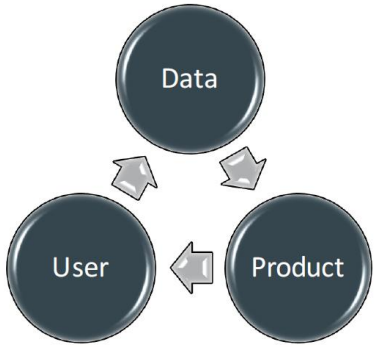
بدون شناخت ورودی‌ها و خروجی‌ها و اینکه از چه نمونه‌هایی استفاده خواهیم کرد، نمی‌توانیم به مرحله بعدی برویم. در این مرحله باید نسبت به فرضیه‌ای که می‌سازید آگاهی کامل داشته باشید:

- در مورد اینکه با این ورودی‌ها می‌توان خروجی را پیش‌بینی نمود، فرضیه‌سازی می‌کنید.
- در مورد کافی بودن اطلاعات نمونه‌های در دسترس برای یادگیری رابطه بین ورودی‌ها و خروجی‌ها فرضیه‌سازی می‌کنید.

تعریف مسئله و فراهم نمودن مجموعه داده

Andrew Ng: “Even though today I have a pretty good sense of how web search technology works, honestly have no idea how to build a small team to build a competitive nearly decent web search engine without access to that data asset.”

https://www.youtube.com/watch?v=NKpuX_yzdYs



تعریف مسئله و فراهم نمودن مجموعه داده

یک نوع از مسائل لاینحل که باید از آن‌ها مطلع باشید، مسائل ناپایدار هستند:

- به عنوان مثال: فرض کنید می‌خواهید سیستم توصیه‌گر لباس بسازید، سیستم را با نمونه‌های تابستان آموزش داده و می‌خواهید تولید پیشنهادات را در زمستان شروع کنید. مشکل بزرگ این است که نوع لباس‌هایی که مردم می‌خرند از یک فصل به فصل دیگر متغیر است.

- در این مورد، حرکت صحیح، آموزش مجدد و مداوم مدل بر اساس نمونه‌های جدیدتر یا جمع‌آوری نمونه‌ها با مقیاس زمانی است که در آن مسئله پایدار است. برای مسائل دوره‌ای مانند خرید لباس، نمونه‌های چند سال برای کسب تغییرات فصلی کافی خواهد بود، اما فراموش نکنید که زمان سال حتماً باید یکی از ورودی‌های مدلتان باشد.

انتخاب معیار موفقیت

برای کنترل یک چیز باید بتوانید بر آن نظارت کنید. برای کسب موفقیت، نیاز است آن را تعریف کنید – صحت ؟ دقت و نرخ فراخوانی ؟ نرخ بقای مشتری؟

معیار شما برای موفقیت، راهنمای انتخاب **تابع هزینه** است: همان چیزی که مدل آن را بهینه خواهد کرد. این معیارها باید مستقیماً در راستای اهداف سطح بالای شما، مانند موفقیت کسب و کارتان باشند.

برای درک تنوع معیارهای موفقیت در یادگیری ماشین و نحوه ارتباط آن‌ها با حوزه‌های مسائل مختلف، مرور رقابت‌های علوم داده در کگل مفید خواهد بود.

تصمیم‌گیری در مورد روش ارزیابی

- پس از تعیین هدف، باید نحوه سنجش پیشرفت امور نیز تعیین گردد.

- قبلاً سه نوع معمول روش‌های اعتبارسنجی را مرور کردیم:
  - کنار گذاشتن مجموعه اعتبارسنجی جداسازی
  - انجام اعتبارسنجی متقابل K بخشی
  - انجام اعتبارسنجی مکرر K بخشی

- فقط کافی است یکی از این روش‌ها را انتخاب کنید. در اکثر موارد، اولین گزینه جوابگو خواهد بود.

آماده‌سازی نمونه‌ها

- همان‌طور که قبلاً مشاهده کردید، نمونه‌ها باید به عنوان تنسور قالب‌بندی شوند.
- مقادیر این تنسورها اغلب باید به بازه‌ای کوچک نگاشت شوند: به عنوان مثال در بازه [1، 1-] یا بازه [1، 0].
- در صورتی که ویژگی‌های مختلف دارای مقادیری در دامنه‌های مختلف هستند (نمونه‌های ناهمگن)، باید نمونه‌ها را نرمال‌سازی کنید.
- ممکن است بخواهید مهندسی ویژگی را به ویژه برای مسائلی با نمونه‌های کم انجام دهید.

توسعه مدلی که بهتر از یک روش پایه عمل می‌کند

هدف در این مرحله کسب قدرت (توان) آماری است: یعنی، ایجاد یک مدل کوچک که توانایی مقابله و شکست یک روش پایه ساده را داشته باشد. در مثال دسته‌بندی ارقام MNISTهر چیزی که دقت آن بزرگ‌تر از 0.1 باشد دارای قدرت آماری است؛ در مثال IMDBقدرت آماری متعلق به چیزی است که دقت آن بالای 0.5 است.

توجه داشته باشید که کسب قدرت آماری همواره مقدور نیست. اگر بعد از چندین معماری منطقی نتوانستید روش پایه تصادفی را شکست بدهید، این احتمال وجود دارد که پاسخ مدنظر برای سؤال شما در نمونه‌های ورودی وجود ندارد.

توسعه مدلی که بهتر از یک روش پایه عمل می‌کند

با فرض اینکه همه‌چیز به خوبی پیش برود، لازم است برای ساخت اولین مدل عملی دست به سه انتخاب کلیدی بزنید:

- فعال‌ساز آخرین لایه: محدودیت‌های مفیدی را به خروجی شبکه اعمال می‌کند.
- تابع هزینه: باید با نوع مسئله‌ای که می‌خواهید حل کنید، همخوانی داشته باشد.
- پیکربندی بهینه‌سازی: از چه بهینه‌سازی‌هایی استفاده خواهید کرد؟ نرخ یادگیری آن چقدر خواهد بود؟

توسعه مدلی که بهتر از یک روش پایه عمل می‌کند

در مورد انتخاب تابع هزینه، باید بدانید که بهینه‌سازی معیاری که موفقیت مسئله را اندازه‌گیری می‌کند، همیشه به طور مستقیم امکان‌پذیر نیست. گاهی اوقات تبدیل معیار به تابع هزینه به هیچ عنوان کار ساده‌ای نیست. توابع هزینه لازم است تنها با داشتن زیردسته نمونه‌ها قابل محاسبه باشند (به طور ایده‌آل، تابع هزینه باید برای یک نمونه نیز قابل محاسبه باشد) و اینکه توابع هزینه باید مشتق‌پذیر باشند (در غیر این صورت برای آموزش شبکه نمی‌توانید از پس‌انتشار استفاده کنید).

توسعه مدلی که بهتر از یک روش پایه عمل می‌کند

نوع مسئله	فعال‌سازی لایه آخر	تابع هزینه
دسته‌بندی دودویی	سیگموید	binary_crossentropy
دسته‌بندی یک برچسبی، چند کلاسی	سافت‌مکس	categorical_crossentropy
دسته‌بندی چند برچسبی، چند کلاسی	سیگموید	binary_crossentropy
رگرسیون با مقادیر تصادفی	هیچ‌کدام	mse
رگرسیون مقادیر بین 0 و 1	سیگموید	mse or binary_crossentropy

بالا بردن قدرت مدل: توسعه مدلی که بیش‌برازش دارد

- بعد از دستیابی به مدلی با قدرت آماری، این سؤال پیش می‌آید که آیا مدل به اندازه کافی قدرتمند است؟ آیا لایه‌ها و پارامترهای کافی برای مدل‌سازی مسئله موردنظر را دارد؟

- مدل آرمانی مدلی است که دقیقاً روی مرز بین کم‌برازش و بیش‌برازش و بین کم‌ظرفیتی و بیش‌ظرفیتی قرار می‌گیرد. برای دانستن جای این مرز، باید ابتدا از آن عبور کنید.

- برای تعیین اندازه مدل موردنیاز، باید مدلی ایجاد کنید که بیش‌برازش داشته باشد. انجام این کار کاملاً آسان است:
  - لایه‌هایی را اضافه کنید.
  - لایه‌ها را بزرگ‌تر کنید.
  - برای تکرار بیشتر آموزش دهید.

تنظیم مدل و میزان کردن ابرپارامترها

این مرحله زمان برترین مرحله خواهد بود: مکرراً مدل را اصلاح خواهید کرد، آموزش خواهید داد و روی نمونه‌های اعتبارسنجی ارزیابی خواهید نمود (در این نقطه روی نمونه‌های آزمایش ارزیابی نخواهید کرد)، مجدداً آن را اصلاح‌کرده و این کارها را تا زمانی تکرار خواهید کرد که مدل به اندازه ممکن اصلاح گردد. مواردی که باید امتحان کنید به شرح زیر هستند:

- اضافه کردن حذف تصادفی
- امتحان معماری‌های مختلف: حذف و اضافه کردن لایه‌ها
- اضافه کردن تنظیم L1 و یا L2
- امتحان ابرپارامترهای مختلف (مانند تعداد واحدها برای هر لایه یا نرخ یادگیری بهینه‌ساز) برای یافتن پیکربندی بهینه
- تکرار اختیاری مهندسی ویژگی: اضافه کردن ویژگی‌های جدید یا حذف ویژگی‌هایی که ظاهراً فاقد اطلاعات مفید هستند.

GOING TIME

4.4-overfitting-and-underfitting

Presentation Time

ارائه ۲: دیتاست‌های
رایج در حوزه‌های
مختلف یادگیری عمیق
(تصویر، متن، صوت).

تشکر

سوال؟

a.golzari@azaruniv.ac.ir

a.golzari@tabrizu.ac.ir

https://github.com/Amin-Golzari-Oskouei