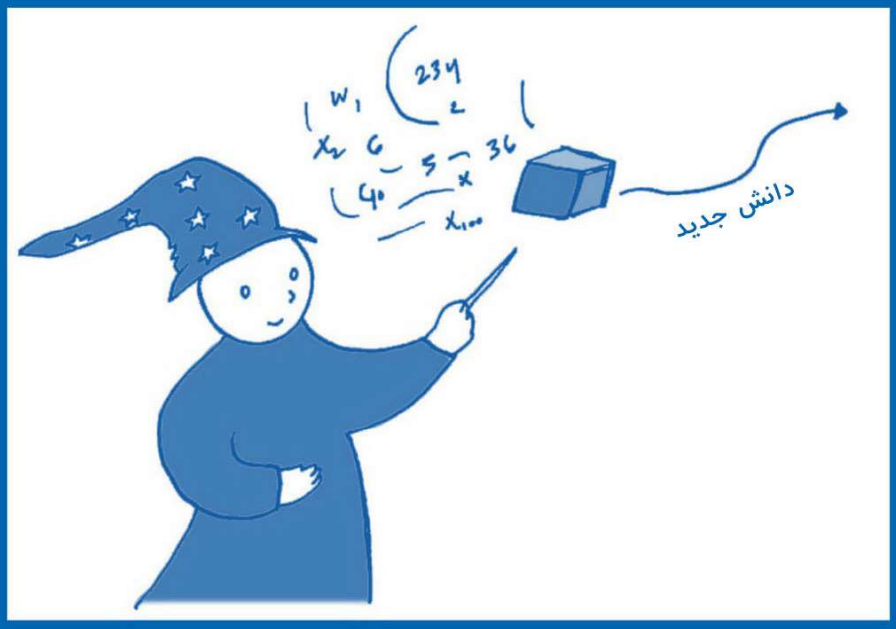
ماشین ها چگونه می آموزند

راهنمای مصور یادگیری ماشین

هلن ادواردز

مترجم: جهاد فرحمند خانقاه



INTELLIGENTSIA RESEARCH

درباره‌ی INTELLIGENTSIA RESEARCH

INTELLIGENTSIA RESEARCH یک شرکت تحقیقاتی در زمینه بازار و سرمایه‌گذاری است که بر هوش مصنوعی و یادگیری ماشین تمرکز دارد. هدف ما ساده‌سازی پیچیدگی‌ها، تفکیک واقعیت از اغراق و قدردانی از پیشرفت‌های کنونی در بستر تاریخ و چشم‌انداز بلندمدت هوش مصنوعی است.

ما با بهره‌گیری از تجربه‌های گسترده در حوزه فناوری اطلاعات، بازاریابی محصولات مصرفی و تحقیقات مالی، دیدگاهی منحصربه‌فرد ارائه می‌دهیم که تجربه کاربر و زیرساخت‌های فناوری موردنیاز را به‌خوبی درک می‌کند.

تحلیل‌های ما انسانی است. ما ایده‌ها و مدل‌هایی را توسعه می‌دهیم که به مشتریان کمک می‌کنند تا بفهمند چرا عصر هوش مصنوعی متفاوت است و چه عواملی برای پذیرش و توسعه موفق یادگیری ماشین ضروری هستند. این تحلیل‌ها تلاش دارند از چرخه‌های رونق و رکود گذشته عبور کرده و منافع قابل‌توجه ماشین‌های هوشمند را به جامعه ارائه دهند.

فهرست موضوعات

- مقدمه
- هوش مصنوعی چیست؟
  - تاریخچه مختصر هوش مصنوعی
- انواع هوش مصنوعی
- یادگیری ماشین
  - یادگیری تحت نظارت
  - یادگیری بدون نظارت
  - یادگیری تقویتی
- آموزش ماشین برای یادگیری
  - استنتاج معکوس
  - شبکه‌های عصبی و یادگیری عمیق
  - ماشین‌ها چگونه می‌بینند؟
  - الگوریتم‌های تکاملی و ژنتیکی
  - الگوریتم‌های بیزین
  - ماشین‌ها چگونه گفتگو می‌کنند؟
  - قیاس‌گرها
- راهنمایی انسان برای ماشین هایی که یاد می گیرند
  - ورودی انسانی ۱: دانش حوزه
  - ورودی انسانی ۲: مهندسی ویژگی‌ها
  - ورودی انسانی ۳: ارزیابی عملکرد مدل
- شش نکته پیش از شروع
  - شماره ۱: بدانید چه مسئله‌ای را می‌خواهید حل کنید
  - شماره ۲: داده خوب < داده بزرگ، هنوز
  - شماره ۳: این بیشتر علم است تا نرم‌افزار
  - شماره ۴: این بیشتر حکمت است تا کدنویسی ساده
  - شماره ۵: هوش مصنوعی تمایل به قطبی کردن افراد دارد، پیام مهم است
  - شماره ۶: ارزش‌ها و شفافیت، تعصب و اخلاق
- کلام پایانی
- تقدیر و تشکر

مقدمه

هوش مصنوعی در حال تغییر دادن زندگی ما از راه‌هایی است که باید آن‌ها را درک کنیم و بشناسیم. الگوریتم‌ها مشخص می‌کنند چگونه اطلاعات را به دست آوریم، چگونه یاد بگیریم، چگونه حرکت کنیم، چگونه خرید کنیم، چه چیزهایی بخریم، چگونه سلامت بمانیم، چگونه ملاقات کنیم، با چه کسانی ملاقات کنیم، چگونه با ما رفتار شود و با چه چیزی با ما رفتار شود. بازاریابی، تجزیه و تحلیل، تشخیص، تولید، رانندگی، جستجو، صحبت کردن، دیدن و شنیدن همگی در حال دگرگون شدن و تغییر شکل به‌وسیله ماشین‌هایی هستند که یاد می‌گیرند. الگوریتم‌هایی که اکنون می‌توانند با سرعت و مقیاسی که داده‌ها تولید می‌شوند، عمل کنند و اکنون آنچه را که زمانی غیرممکن بود، به یک واقعیت عملی تبدیل کنند. برای مثال، الگوریتم‌هایی که می‌توانند:

> الگوهای ظریف در تراکنش‌ها را تشخیص دهند و در حجم‌های انبوهی از متن، اطلاعات مالی و موقعیت مکانی، تقلب را دقیقا پیش‌بینی کنند.
>
> میلیون‌ها نقطه‌داده[1] را از منابع داده‌ای متفرق پردازش کنند تا بطور قابل اعتمادی، الزامات و بایدها و نبایدهای نگهداری تجهیزات پیچیده مانند موتورهای جت را پیش‌بینی کنند.
>
> مخازن بزرگ داده‌های پراکنده را که در منابع اجتماعی امروزی، وبلاگ‌ها، سیستم‌های تراکنش، سیستم‌های جغرافیایی فضایی و دستگاه‌های شخصی افراد یافت می‌شود، ترکیب کنند تا بینش‌های عملی برای بازاریابان ایجاد شود.

و همچنان که هوش مصنوعی بیشتری را در زندگی روزمره خود ادغام می‌کنیم، بیشتر در مورد خودمان، ترجیحاتمان، چگونگی انتخابهای‌مان، اینکه چه چیزی را می‌خواهیم کنترل کنیم، و اینکه چگونه ممکن است تحت تاثیر ماشین‌ها قرار بگیریم، یاد می‌گیریم.

اما ماشین‌ها چگونه یاد می‌گیرند؟

همه‌اش فقط ریاضیات است.

- اورن اتزیونی

که دلالت دارد بر اینکه برای درک هوش مصنوعی، شما باید درک کنید که ماشین‌ها چگونه یاد می‌گیرند و برای درک اینکه ماشین‌ها چگونه یاد می‌گیرند، باید ریاضیات را درک کنید. آموختن ریاضیات پیچیده، چیزی نیست که همه به آن علاقمند باشند، بنابراین من این کتاب را نوشتم تا آن قسمتی از ریاضیات را که مهم است به شما نشان دهم.

چرا این مهم است؟

کامپیوتر می‌گوید نه.

- کارول بیر، سریال Little Britain

این عبارت طعنه آمیز که در متن نمایش کمدی سریال Little Britain محبوب شد، زیربنای نگرانی مشترکی در هوش مصنوعی است؛ فقدان شفافیت و سلب قدرت که سیستم‌های خودکار می‌توانند ایجاد کنند. اکنون که برنامه‌های رایانه‌ای می‌توانند خود را تطبیق دهند، بر [و در] پایگاه‌های کاربری بزرگ‌تر و بزرگ‌تر عمل کنند، به صورت فیزیکی در روبات‌ها یا وسایل نقلیه تعبیه شوند و نیازی به ورود انسان برای نتیجه‌گیری یا حفظ نتایج نداشته باشند، این نگرانی دیگر تنها به خدمات ضعیف مشتری یا عدم کمک محدود نمی‌شود، بلکه می‌تواند بسیار جدی‌تر باشد. سوگیری، اخلاق، تبعیض و بی‌انصافی (یا ناعدالتی)، از پیامدهای اجتناب‌ناپذیر سیستم‌های هوش مصنوعی‌ای هستند که به صورت ضعیفی طراحی و اجرا شده‌اند. برای درک اینکه

۱- نقطه‌داده یا Data Point یک مقدار یا مشاهده‌ی خاص از داده است که یک ویژگی یا خصوصیت را نشان می‌دهد. به عبارت دیگر، هر نقطه‌داده یک "قطعه کوچک اطلاعات" است که می‌تواند بخشی از یک مجموعه‌داده بزرگ‌تر باشد.

چگونه تصمیمات توسط انسان‌ها اتخاذ می‌شود و ریاضیات چگونه یک الگوریتم را هدایت می‌کند، به فهم عمیق‌تری از هوش مصنوعی نیاز داریم.
هدف این کتاب این است که شما را با عواملی آشنا کند که هوش مصنوعی امروز را هدایت می‌کنند، تا متوجه شوید چه چیزی این حوزه از علوم کامپیوتر را از مهندسی نرم‌افزار سنتی متمایز می‌کند. این کتاب برای مدیرانی طراحی شده که مایلند از یادگیری ماشین در کسب‌وکار خود استفاده کنند و مکانیزم‌های اساسی آن را درک کنند و همچنین برای هر فردی که علاقه‌مند به فهم معماری‌های هوش مصنوعی و یادگیری ماشین است..

توضیحات مترجم:

در این ترجمه، هر مفهوم یا تصویری را که متوجه نشدم و یا احساس کردم که به توضیح بیشتری نیاز دارد، به صورت پانوشت، داخل قلاب یا به صورت متن توضیحی سرمه‌ای‌رنگ در کنار متن اصلی گنجانده‌ام.

هوش مصنوعی چیست؟

اولاً، هوش چیست؟ ما ویژگی‌های متعددی را به رفتار هوشمندانه نسبت می‌دهیم، از جمله:

- داشتن توانایی ساختن، حفظ کردن و استفاده از یک مخزن بزرگ دانش

  حلزون‌ها با نرم‌تنان صدفی خویشاوندند، آنها در همه جای زمین زندگی می‌کنند، برخی می‌توانند به عنوان سازهای موسیقی استفاده شوند[۲]، برخی می‌توانند «تیرهای عشق[۳]» شلیک کنند.

- استدلال عرفی / عقل سلیم

  اگر سوزنی را در بازوی کسی فرو کنید، آیا سوراخ در بازو ایجاد می‌شود یا در سوزن؟

- تعیین روابط بین حقایق

  تیدلز یک گربه است. گربه‌ها میرا هستند. تیدلز میرا است.

- داشتن توانایی تحلیل

  مسئله: چند هواپیما در حال حاضر در آسمان هستند؟

  رویکرد: چند فرودگاه وجود دارد؟ چند پرواز در هر فرودگاه وجود دارد؟ هر پرواز چقدر طول می‌کشد؟

- داشتن ظرفیت یادگیری

  اولین گام‌ها، اولین کلمه، اولین جایزه

- انتقال ایده‌ها به دیگران و درک ارتباطات آنها

  -.- --- ...---...

  (این کد مورس است که در آن شخص ۱ می‌گوید SOS و شخص ۲ می‌گوید Ok)

- درک جهان

  آسمان آبی است

- معنی دادن به جهان [درک و تفسیر جهان]

  آسمان خاکستری است. ممکن است باران ببارد.

هوش مصنوعی ماشینی است که این کارها را انجام می‌دهد. در درجه اول برای اتوماسیون کارها استفاده شده است. بسیاری از کارهای تکراری، خودکار شده‌اند - اپراتورهای تلفن، متصدیان بانک، کارگران مونتاژ کارخانه - و بسیاری از وظایف به سمت رایانه‌ای شدن می‌روند زیرا از رویه‌های کاملاً تعریف شده و تکراری تشکیل شده‌اند که به راحتی در نرم‌افزار کدگذاری می‌شوند. هوش مصنوعی بیشتر در مورد کارهای روتین بوده است، به عنوان مثال، حذف یک انسان از توالی تکراری جوش نقطه‌ای در خط مونتاژ خودرو. در طول نیم قرن گذشته، افزایش قدرت محاسباتی، امکان اتوماسیون پیچیده‌تر و پیشرفته‌تری را فراهم کرده است. اما کارهای غیر روتین، مانند رانندگی با ماشین، تشخیص اشیا در تصاویر و هر چیزی که شامل تعامل در زبان طبیعی است، در برابر اتوماسیون مقاومت کرده‌اند زیرا ماشین‌ها به راحتی به بسیاری از چیزهایی که ما به صورت شهودی[۴] درمی‌یابیم، دست نیافته‌اند.

برای مثال این جمله ساده را در نظر بگیرید:

پت به صدا درآمدن زنگ هشدار را شنید (Pat heard the alarm go off)

برای ما بدیهی است که به صدا آمدن (go off) زنگ هشدار همان زنگ خوردن (ring) است، اما برای یک ماشین اینطور نیست. ما همچنین به طور شهودی درک می‌کنیم که پت (pat) در این زمینه یک شخص است. اما یک ماشین "پت" را به عنوان یک فعل [به معنای tap] نیز در نظر می‌گیرد، در این صورت جمله کاملاً بی معنی می‌شود.

۲- برخی گونه‌های حلزون (مثلاً نوعی صدف دریایی به نام "شاخ حلزون" یا Conch) می‌توانند به عنوان ابزار موسیقی بادی مورد استفاده قرار گیرند که اشاره به کاربرد غیرعادی آن‌هاست.

۳- برخی از گونه‌های حلزون‌ها رفتاری دارند که به آن «پرتاب تیر عشق» گفته می‌شود. این تیرها ساختارهای آهکی کوچکی هستند که در طول جفت‌گیری به بدن حلزون دیگر پرتاب می‌شوند و حاوی هورمون‌هایی هستند که شانس موفقیت باروری را افزایش می‌دهند.

٤- شهود (Intuition): درک مستقیم و ناخودآگاه از چیزی بدون نیاز به استدلال آگاهانه.

هم‌زمان با اینکه الگوریتم‌ها به روش‌های جدیدی به تحلیل جریان‌های عظیم داده می‌پردازند، نحوه درک ماشین‌ها از جهان در حال تغییر است. داده‌های جدید، مانند تصاویر خانه‌ها در خیابان‌ها، عکس‌های بی‌شمار از دوستان، حیوانات خانگی و مکان‌ها، داده‌های حسگرها از گجت‌های پوشیدنی تناسب اندام یا سنسورهای متصل به خودروها، به ماشین‌ها این امکان را می‌دهند که تفاوت یک گربه و سگ، دوست و دشمن و آنچه نمی‌خواهید با چیزی را که احتمالاً خریداری خواهید کرد بیاموزند. با کمی کمک از انسان‌ها، ماشین‌ها داده‌ها را دریافت می‌کنند و خودشان درباره جهان یاد می‌گیرند، برنامه‌هایشان را به‌روزرسانی کرده و به‌طور خودکار عملکردشان را بهبود می‌بخشند.

بنابراین، درست است که هوش مصنوعی درباره خودکارسازی است، اما با افزایش چشمگیر منابع داده و قدرت محاسبات، دامنه خودکارسازی به سرعت در حال گسترش است. اکنون می‌توانیم وظایفی را که پیشتر منحصراً انسانی تصور می‌شدند، خودکار کنیم. اتوماسیون دیگر فقط خودکارسازی نیست، بلکه به شخصی‌سازی نیز تبدیل شده است.

آیا باید نگران باشیم؟

آینده هوش مصنوعی اغلب به صورت طیفی بین آرمان‌شهر و ویران‌شهر تصور می‌شود. ایلان ماسک و استیون‌هاوکینگ از جمله متخصصانی هستند که نگرانی‌های جدی درباره ساخت هوش مصنوعی هوشیاری که بتواند سریع‌تر و مؤثرتر از انسان فکر و عمل کند، مطرح کرده‌اند. نیک بوستروم، فیلسوف دانشگاه آکسفورد، درباره «مسئله کنترل» یعنی چگونگی طراحی یک هوش مصنوعی که پس از رسیدن به هوشی فرابشری، علیه ما عمل نکند، صحبت می‌کند. در کوتاه‌مدت، ملاحظات مهم از جنس کنترل کردن مانند طراحی ایمن خودروهای خودران، قوانین بین‌المللی برای جنگ خودکار و ملاحظات اخلاقی در الگوریتم‌های آماری وجود دارند. همچنین نگرانی‌های موجهی در مورد افزایش بیکاری ناشی از فناوری با افزایش اتوماسیون وجود دارد. بعداً به برخی از آنها اشاره می‌کنم، اما تمرکز من در این کتاب بر معرفی ساده علم، ریاضیات و مهندسی است که چگونگی یادگیری ماشین‌ها را تعریف می‌کند.

نکته اصلی این است که هوش مصنوعی برای بهبود زندگی ما طراحی شده است.

ما به طور معمول چندین بار در روز از نتایج الگوریتم‌های یادگیری ماشین استفاده می‌کنیم. خواه با سیری (SIRI) صحبت کنیم یا روی پیشنهادی از آمازون کلیک کنیم یا از نحوه شخصی‌سازی فید خبری خود در فیس‌بوک آگاه شویم، تجربه‌های مصرفی روزمره ما در حال تغییر است.

درک زبان طبیعی، دستیاران شخصی‌ای را در اختیار ما قرار می‌دهد که واقعاً نیازهایمان را می‌فهمند.

تحلیل و پیش‌بینی سریع‌تر و دقیق‌تر، اعتماد ما را به مشاوره‌های تخصصی که می‌گیریم، بیشتر می‌کند.

دستگاه‌های هوشمندتر، خانه‌هایمان را راحت‌تر، امن‌تر و سرگرم‌کننده‌تر می‌کنند.

انتخاب‌های بهتر، ما را کارآمدتر می‌کنند. چه در حمل‌ونقل، تجارت، اجتماع، امور مالی، سرگرمی یا آموزش، کنترل بیشتری بر نحوه استفاده از وقتمان داریم.

کامپیوترها در کارهایی که هدفشان مشخص است، مانند کشیدن نمودار از یک جدول اعداد، بسیار خوب عمل می‌کنند. از آنجا که بخش زیادی از تعاملات انسانی و فعالیت‌های خلاقانه به صورت آنلاین انجام می‌شود و می‌توان آنها را در قالب داده‌ها ثبت نمود، به اشتراک گذاشت و با الگوریتم‌ها تحلیل کرد، حتی کارهایی که بیشتر به انسان مربوط می‌شوند و شامل مهارت‌های اجتماعی و خلاقیت هستند، اکنون توسط ماشین‌ها قابل یادگیری‌اند. شبکه‌های پیچیده تعامل و رفتار انسان اکنون به طور معمول تحلیل می‌شوند و از نتایج آنها برای شخصی‌سازی موارد برای شما استفاده می‌شود.

به تدریج، دیگر هیچ مانع فنی برای دستیابی ماشین به هر هدفی بدون راهنمایی انسان وجود ندارد. این وضعیتی کاملاً متفاوت با نحوه تعامل سنتی ما با کامپیوترهاست و دلیل اصلی اهمیت درک چگونگی یادگیری ماشین‌های امروزی است.
اما ابتدا، نگاهی کوتاه به تاریخچه هوش مصنوعی؛ از آزمون مشهور آلن تورینگ تا آلفاگو، از وعده‌های هوش مصنوعی تا دوران رکود متعدد آن.
سفری صد ساله...

تاریخچه‌مختصر هوش مصنوعی

انواع هوش مصنوعی

روش‌های گوناگونی برای دسته‌بندی هوش مصنوعی وجود دارد. به دلیل تداخل‌ها و ارتباطات پیچیده بین آنها، گاهی این دسته‌بندی‌ها می‌توانند گیج‌کننده باشند.

یکی از راه‌های تقسیم‌بندی هوش مصنوعی، دو شاخه اصلی زیر است:

- سیستم‌های دانش یا خبره
- سیستم‌های یادگیری ماشین

سیستم‌های دانش یا خبره، ساختاری برای ثبت و کدگذاری دانش یک متخصص انسانی در یک حوزه خاص فراهم می‌کنند. سیستم‌های دانش، اساساً زیربنای تمام سیستم‌های قانون "اگر-آنگاه" سنتی هستند و اولین نمونه‌های واقعاً موفق هوش مصنوعی به شمار می‌روند. این سیستم‌ها در دهه ۱۹۷۰ ابداع و در دهه ۱۹۸۰ رایج شدند. برای نمونه...

اگر فلان، آنگاه بهمان

اگر برنامه پیش‌بینی آب‌وهوا نشان داد فردا بارانی است،
آن‌گاه به من با یک نوتیفیکیشن موبایل هشدار بده.

اگرچه بیشتر فناوری‌های امروزی همچنان مبتنی بر قواعد هستند، این سیستم‌ها ظرفیت و انعطاف‌پذیری محدودی در دنیای مدرنِ متصل به اینترنت دارند. آنها به اندازه کافی سریع تغییر نمی‌کنند تا برای مدت طولانی مفید باشند و در نهایت، آنچه مردم می‌توانند با اطلاعات انجام دهند را محدود می‌کنند. برای مثال، سیستم‌های خبره سال‌ها برای کمک به پزشکان در تصمیم‌گیری‌های درمانی درست استفاده می‌شدند. در حالت حداکثری، یک سیستم خبره برای مجموعه‌ای مشخص از ورودی‌ها، یک نتیجه ارائه می‌دهد که برای همه افراد در جهان یکسان است. اما اکنون هدف پزشکی شخصی[سازی شده]، ارائه نتایجی به تعداد افراد جهان است.

با ظهور هوش مصنوعی جدید، یعنی یادگیری ماشین، تغییری بنیادین در مقیاس[۵]، سرعت[۶] و دامنه[۷] رخ داده است.

۵- مقیاس (Scale): در اینجا، مقیاس به دو جنبه اصلی اشاره دارد:

- حجم داده‌ها :یادگیری ماشین، به ویژه یادگیری عمیق، برای عملکرد مؤثر به حجم عظیمی از داده‌ها نیاز دارد. پیشرفت‌های اخیر در جمع‌آوری و ذخیره‌سازی داده‌ها (به عنوان مثال، داده‌های بزرگ) به سیستم‌های یادگیری ماشین اجازه داده است تا با داده‌های بسیار بیشتری نسبت به گذشته آموزش ببینند و در نتیجه، عملکرد بهتری داشته باشند. بنابراین، "مقیاس" به افزایش چشمگیر حجم داده‌های مورد استفاده اشاره دارد.

- پیچیدگی مدل‌ها :با افزایش قدرت محاسباتی، امکان ساخت مدل‌های پیچیده‌تر و بزرگ‌تر در یادگیری ماشین فراهم شده است. این مدل‌های بزرگ‌تر (مانند شبکه‌های عصبی عمیق با میلیون‌ها یا حتی میلیاردها پارامتر) می‌توانند الگوهای پیچیده‌تری را در داده‌ها یاد بگیرند. "مقیاس" در اینجا به افزایش اندازه و پیچیدگی این مدل‌ها اشاره دارد.

۶- سرعت (Speed): سرعت به چند جنبه مربوط می‌شود:

- سرعت آموزش :با پیشرفت سخت‌افزاری (مانند GPUها و TPUها) و الگوریتم‌های بهینه‌سازی، سرعت آموزش مدل‌های یادگیری ماشین به طور چشمگیری افزایش یافته است. این بدان معناست که می‌توان مدل‌ها را در زمان کوتاه‌تری آموزش داد و سریع‌تر به نتایج رسید.

- سرعت پردازش و پیش‌بینی :مدل‌های آموزش‌دیده می‌توانند با سرعت بسیار بالایی داده‌های جدید را پردازش کرده و پیش‌بینی انجام دهند. این امر برای کاربردهایی مانند تشخیص تصویر، پردازش زبان طبیعی و سیستم‌های توصیه‌گر بسیار مهم است.

۷- دامنه (Scope): دامنه به گستردگی کاربردهای یادگیری ماشین اشاره دارد. با پیشرفت‌های اخیر، یادگیری ماشین در حوزه‌های بسیار متنوعی مورد استفاده قرار می‌گیرد، از جمله: بینایی کامپیوتر (تشخیص چهره، تشخیص اشیا، تجزیه و تحلیل تصاویر پزشکی.)، پردازش زبان طبیعی (ترجمه ماشینی، تحلیل احساسات، چت‌بات‌ها.)، خودروسازی (خودروهای خودران.)، بهداشت و درمان (تشخیص بیماری‌ها، توسعه داروها.) و مالی (تشخیص تقلب، پیش‌بینی بازار.)

سیستم‌های یادگیری ماشین به کامپیوترها توانایی یادگیری بدون برنامه‌نویسی صریح را می‌دهند. الگوریتم‌ها مدل‌هایی می‌سازند که با قرار گرفتن در معرض داده‌های جدید، به‌روز می‌شوند و تغییر می‌کنند. تاریخچه یادگیری ماشین به دهه ۱۹۵۰ برمی‌گردد، اما پیشرفت‌های اخیر، نتیجه همگرایی بین توسعه ابزارهای احتمالاتی[۸] در دهه ۱۹۹۰ و دسترسی به داده‌های بزرگ و تراشه‌های قدرتمند مورد استفاده در پردازش گرافیکی در سال‌های اخیر است.

یکی از کاربردهای یادگیری ماشین، جستجوی اینترنتی است. گوگل از الگوریتم‌های پیچیده‌ای استفاده می‌کند که نتایج جستجو را بر اساس اطلاعاتی که از شما و همچنین عوامل متعدد دیگر درباره افرادی شبیه شما و درباره جهان دارد، شخصی‌سازی می‌کند. به همین دلیل، افراد مختلف هنگام جستجوی یک عبارت یکسان، نتایج متفاوتی می‌بینند. حتی پیشنهادهای خودکار متفاوتی در حین جستجو دریافت می‌کنیم که می‌تواند بسته به زمان و مکان جستجو تغییر کند. این آزمایش را انجام دهید: عبارت «بهترین .... چیست» را در دستگاه‌های مختلف تایپ کنید. آیا هر بار پیشنهاد مشابهی می‌بینید؟

گوشی من: «بهترین راه برای بیرون کشیدن لنگر چیست»

لپ‌تاپ من: «بهترین پوکمون در بازی پوکمون گو چیست»

مثال دیگر، پیش‌بینی قیمت‌هاست. فرض کنید می‌خواهید قیمت فروش یک خانه را پیش‌بینی کنید. می‌توانید مدلی بسازید که این قیمت را بر اساس ویژگی‌های مختلف (متراژ، تعداد اتاق خواب، تعداد حمام و غیره) تخمین بزند. اگر از یک مدل ثابت استفاده کنید، الگوریتم فقط یک بار اجرا می‌شود و یک همبستگی تک‌باره[۹] ارائه می‌دهد که احتمالاً فقط در یک منطقه محدود کاربرد دارد. اما خانه‌ها دائماً خرید و فروش می‌شوند و الگوهای زیادی در یک مکان دیده می‌شود که می‌تواند قیمت‌ها را در همه مناطق مشخص کنند، الگوهایی که نمی‌توانیم ببینیم زیرا متغیرهای بسیار زیاد و روابط پنهان بسیار زیادی وجود دارند. یک الگوریتم یادگیری ماشین می‌تواند مدلی را در متغیرهای بسیار بسیار زیاد بسازد، به هر فروش جدید خانه نگاه کند، آن را با قیمت پیش‌بینی شده مدل برای آن خانه مقایسه کند و سپس مدل را به طور خودکار تنظیم کند تا دقیق‌تر شود. همچنین سیستم‌های یادگیری ماشین می‌توانند داده‌های بسیار بیشتری را در اشکال مختلف و با سرعت مدیریت کنند. به هر حال این سیل داده چقدر مهم است؟

رشد داده‌ها به صورت تصاعدی نمایی است...

> رشد داده‌ها در سطح جهان، منحنی رشد چشمگیری را طی می‌کند و پیش‌بینی می‌شود از ۴.۴ زتابایت در سال ۲۰۱۳ به ۴۴ زتابایت تا سال ۲۰۲۰ برسد (یک زتابایت برابر با ۱ تریلیون گیگابایت است). حتی این رقم نیز مرتباً افزایش داده می‌شود. تصور این حجم از داده دشوار است؛ برای بیشتر ما، فقط به معنی «خیلی خیلی زیاد» است.

...و بخش عمده این رشد داده از منابع جدید ناشی می‌شود.

> ترافیک جهانی داده‌های موبایل بین سال‌های ۲۰۱۵ و ۲۰۲۰ تقریباً هشت برابر افزایش خواهد یافت. سرعت اتصال شبکه‌های موبایل تا سال ۲۰۲۰ بیش از سه برابر خواهد شد. تا سال ۲۰۲۰، بیش از سه پنجم دستگاه‌های متصل به شبکه موبایل، دستگاه‌های «هوشمند» خواهند بود. این فقط شامل گوشی‌های هوشمند نمی‌شود، بلکه خودروهای خودران، حسگرهای خانه‌های هوشمند، شهرهای هوشمند و طیف گسترده‌ای از حسگرها و کنترلرهای صنعتی را نیز در بر می‌گیرد.

قدرت پردازش نیز با نرخ نمایی در حال افزایش است...

۸- اصطلاح "probabilistic tools" یا "ابزارهای احتمالاتی" در زمینه هوش مصنوعی و به ویژه یادگیری ماشین به مجموعه‌ای از روش‌ها، تکنیک‌ها و مدل‌هایی اشاره دارد که بر پایه نظریه احتمال و آمار بنا شده‌اند. این ابزارها برای مدل‌سازی و تحلیل داده‌هایی که دارای عدم قطعیت یا تصادف هستند، استفاده می‌شوند. به عبارت دیگر، وقتی با داده‌هایی روبرو هستیم که نمی‌توان به طور قطعی در مورد آن‌ها صحبت کرد (مثلاً احتمال وقوع یک رویداد، میزان خطا در یک اندازه‌گیری، یا توزیع داده‌ها)، از ابزارهای احتمالی برای درک و پیش‌بینی رفتار آن‌ها استفاده می‌کنیم. این ابزارها به کامپیوترها و سیستم‌ها کمک می‌کنند تا با داده‌های غیرقطعی (uncertain)، متغیر یا دارای نویز (noise) کار کنند و پیش‌بینی‌های دقیق‌تری انجام دهند.

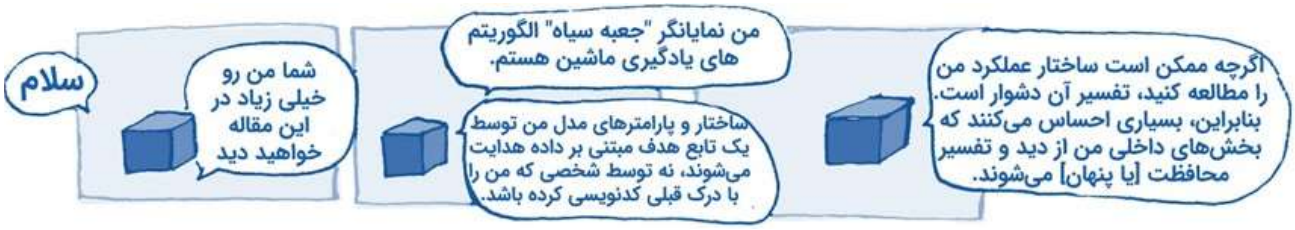
۹- در این متن، عبارت "one-time correlation" به معنای "همبستگی تک‌باره" یا "همبستگی در یک زمان مشخص" است و به نوعی از تحلیل همبستگی اشاره دارد که فقط یک بار و در یک نقطه زمانی خاص انجام می‌شود و نتایج آن فقط برای همان زمان و احتمالاً همان محدوده جغرافیایی معتبر است

حتی با وجود نزدیک شدن به محدودیت فیزیکی سرعت تراشه‌ها و رسیدن قانون مور[۱۰] به نقطه پایانی خود، همچنان پیش‌بینی می‌شود که ابرکامپیوترها در سال ۲۰۲۰ حدود ۳۰ برابر قدرتمندتر از سال ۲۰۱۵ باشند. بسیاری از پیشرفت‌های اخیر در هوش مصنوعی، به دلیل استفاده از واحدهای پردازش گرافیکی (GPUها[۱۱]، تراشه‌های قدرتمندتری که اساساً برای پردازش گرافیکی طراحی شده‌اند) بوده است. اکنون تراشه‌ها به طور خاص برای کاربردهای یادگیری ماشین طراحی می‌شوند، به گونه‌ای که سرعت و مصرف انرژی بهینه شوند.

...این امکان را به دانشمندان داده می‌دهد تا حتی پیچیده‌ترین داده‌های «بدون ساختار» را هم پردازش کنند.

تا همین اواخر، داده‌ها برای استفاده توسط کامپیوترها به ساختار نیاز داشتند، اما الگوریتم‌ها به طور فزاینده‌ای قادر به پردازش این نوع داده‌ها هستند. داده‌های بدون ساختار، داده‌هایی بدون برچسب و سازمان‌دهی هستند که اغلب پر از ابهام و ناهماهنگی‌اند. بیشتر داده‌های جهان بدون ساختارند؛ مانند تصاویر، ویدیوها و متن موجود در اسناد.
موضوع بعدی: یادگیری ماشین، چیستی، چگونگی عملکرد و کاربردهای آن.
اما قبل از آن...

در بخش بعدی، یک جعبه سیاه کوچک در تصاویر خواهید دید. بسیاری از مراحلی که ماشین‌ها برای یادگیری طی می‌کنند، شامل فرایندهایی هستند که چندان شفاف نیستند. ممکن است مراحلی وجود داشته باشد که محاسبات در آنها بسیار پیچیده و تجسم آنها دشوار باشد یا محاسبات ریاضی بسیار پیچیده‌ای که حتی متخصصان هم به طور کامل آنها را درک نمی‌کنند. هر زمان که این جعبه سیاه ظاهر شود، به این معنی است که نوعی «جادو» وجود دارد که باعث کار کردن الگوریتم می‌شود. این ابهام، الگوریتم‌های یادگیری را از کدهای سنتی بسیار متفاوت می‌کند و یکی از دلایل اصلی اهمیت درک ریاضیات پشت یادگیری ماشین است.



۱۰- قانون مور (Moore's Law) یک پیش‌بینی است که توسط گوردون مور، یکی از بنیان‌گذاران شرکت اینتل، در سال ۱۹۶۵ مطرح شد. این قانون بیان می‌کند که تعداد ترانزیستورها در یک مدار مجتمع (IC) تقریباً هر دو سال یک‌بار دو برابر می‌شود. این افزایش تعداد ترانزیستورها معمولاً به معنای بهبود عملکرد و افزایش سرعت پردازش در تراشه‌های کامپیوتری است.

۱۱- در یادگیری ماشین و هوش مصنوعی، حجم زیادی از محاسبات ریاضی باید روی داده‌ها انجام شود. GPU می‌تواند این محاسبات را به‌طور هم‌زمان و بسیار سریع انجام دهد، به همین دلیل برای آموزش مدل‌های هوش مصنوعی استفاده می‌شود. فرض کنید می‌خواهید تعداد زیادی نقطه را روی یک صفحه رسم کنید. CPU هر نقطه را یکی‌یکی رسم می‌کند اما GPU همه نقاط را هم‌زمان رسم می‌کند. این باعث می‌شود GPU در کارهایی که نیاز به پردازش سنگین و هم‌زمان دارند، بسیار سریع‌تر باشد.

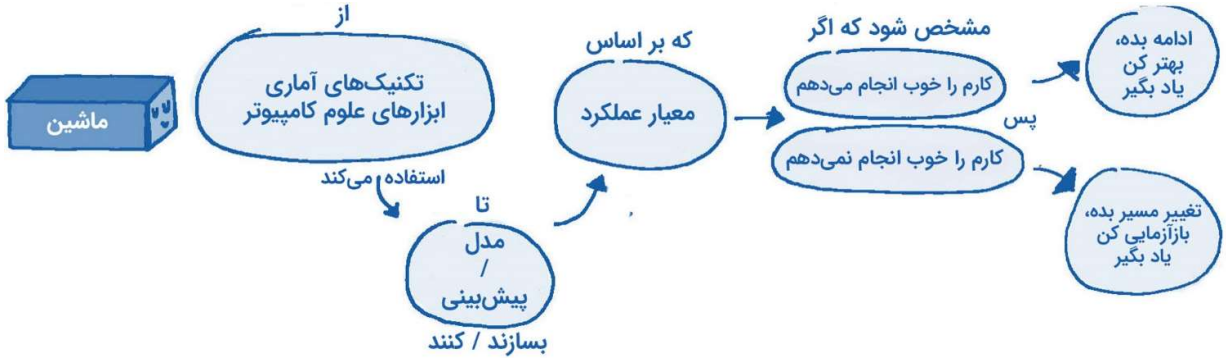
یادگیری ماشین

از X (داده های مرتبط) یاد بگیر تا Y (پیش‌بینی دقیق) را انجام بدی

این تابع ریاضی ساده در قلب یادگیری ماشینی قرار دارد. ما می‌خواهیم با استفاده از متغیرهای ورودی، امکان پیش‌بینی را پیدا کنیم، اما تابعی را که به لحاظ ریاضی ورودی‌ها را به خروجی‌ها مرتبط کند، نمی‌شناسیم. اگر آن را می‌شناختیم، از آن استفاده می‌کردیم اما چون آن را نمی‌دانیم، باید آن را از داده‌ها بیاموزیم. یک ماشین رابطه بین ورودی‌ها و خروجی‌ها را می‌آموزد و سپس به‌طور خودکار برنامه‌های خود را برای انجام پیش‌بینی‌های جدید از داده‌های جدید بهبود می‌بخشد.

راه‌های زیادی وجود دارد که ماشین‌ها می‌توانند برنامه‌های خود را بازنویسی کنند. بیشتر آنها بر نوعی حساب دیفرانسیل و انتگرال یا تکنیک آماری تکیه می‌کنند. همه الگوریتم‌های یادگیری ماشین از داده‌ها برای آموزش یک مدل استفاده می‌کنند تا بتواند بر اساس معیاری از عملکرد، پیش‌بینی‌ها یا تصمیماتی را اتخاذ کند. داده‌ها، ماده خام هستند؛ سبک یادگیری به نوع داده، نوع مسئله و نوع الگوریتم بستگی دارد.

در عبارت ریاضی بالا، عملکرد بر اساس میزان دقت برنامه در پیش‌بینی مقداری برای خروجی که قبلاً شناخته شده بود، اندازه‌گیری می‌شود. تفاوت بین خروجی پیش‌بینی شده و خروجی واقعی، خطا است. روش‌های مختلفی برای برنامه‌ریزی یک رایانه برای تنظیم تابع خود بر اساس این خطا وجود دارد.



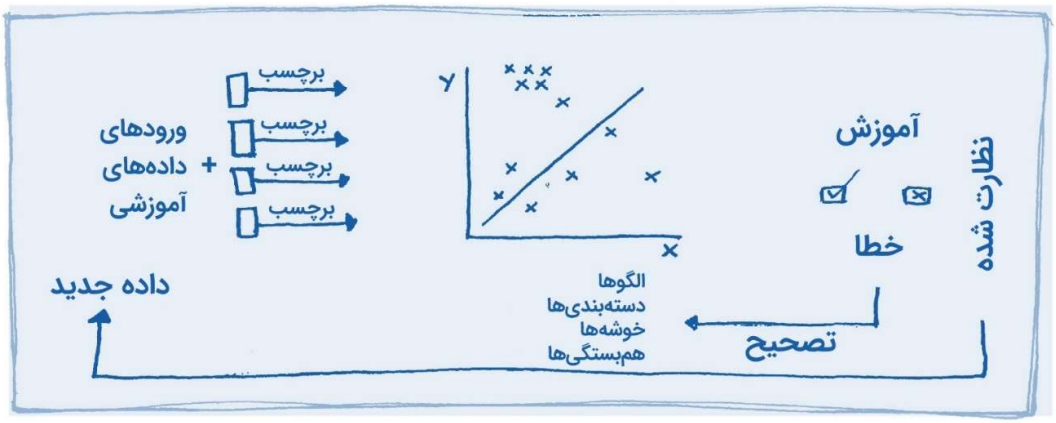
قبل از ورود به چگونگی انجام این کار، بیائید ابتدا بررسی کنیم که چگونه ماشین می‌فهمد خطایی وجود دارد که باید بر اساس آن عمل کند. سه دسته کلی یادگیری ماشین وجود دارد: یادگیری نظارت شده، یادگیری بدون نظارت و یادگیری تقویتی.

یادگیری نظارت شده

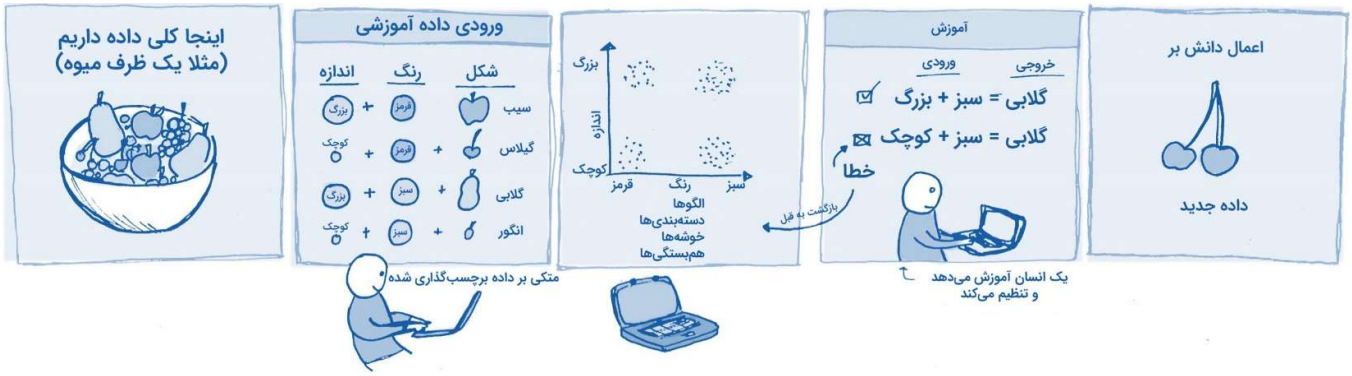
اغلب الگوریتم‌های یادگیری ماشین، الگوریتم‌های یادگیری نظارت شده هستند. الگوریتم‌های یادگیری نظارت شده بر اساس مجموعه‌ای از مثال‌هایی که توسط انسان ارائه و برچسب گذاری شده است، پیش بینی می‌کنند. این بدان معناست که یک انسان تصمیم می‌گیرد از چه داده‌هایی استفاده کند (که به این عمل، مهندسی ویژگی[۱۲] می‌گویند)، داده‌هایی را برای آموزش مدل تأمین می‌کند (به این نوع داده‌ها، داده‌های آموزشی می‌گویند) و داده‌هایی را برای آزمایش مدل ارائه می‌دهد (به این نوع داده‌ها، داده‌های آزمایشی می‌گویند).

۱۲- الگوریتم‌های یادگیری ماشین نمی‌توانند مستقیماً با داده‌های خام (مانند متن، تصاویر یا داده‌های جدولی بدون پردازش) کار کنند. آن‌ها به داده‌هایی نیاز دارند که به شکل عددی و قابل فهم برایشان باشند. همچنین، داده‌های خام ممکن است حاوی اطلاعات اضافی، نویز یا ویژگی‌های نامربوط باشند که عملکرد مدل را کاهش می‌دهند. مهندسی ویژگی (Feature Engineering) به زبان ساده، هنر و علمِ تبدیل داده‌های خام به ویژگی‌هایی است که برای الگوریتم‌های یادگیری ماشین قابل فهم و مفید باشند. به عبارت دیگر، شما با استفاده از دانش خود در مورد مسئله و داده‌ها، ویژگی‌های جدیدی می‌سازید که به مدل یادگیری ماشین کمک می‌کنند تا الگوها را بهتر شناسایی کند و پیش‌بینی‌های دقیق‌تری انجام دهد.

در یادگیری نظارت شده، داده‌ها با مقدار مورد نظر برچسب گذاری می‌شوند، به عنوان مثال، قیمت یک سهام خاص. الگوریتم به دنبال الگوهایی در مقادیر است. می‌تواند از هر اطلاعاتی که ممکن است مرتبط باشد (نتایج مالی، آب و هوا، رویدادهای جهانی) برای یافتن الگو استفاده کند. هنگامی‌که یک الگو پیدا می‌شود، پیش بینی برای، مثلاً، قیمت فردا انجام می‌دهد. هنگامی‌که مدل در حال آموزش است، تا زمانی که سطح دقت مورد نظر به دست آید، اصلاح و تصحیح صورت می‌گیرد.



این تصویر، داده‌هایی را نشان می‌دهد که همگی بر اساس ویژگی‌های x و y برچسب‌گذاری شده‌اند. این نقطه‌داده‌های برچسب‌گذاری‌شده رسم شده‌اند و خطی کشیده شده که بهترین برازش[۱۳] را با داده‌ها دارد. ما خطا را با یافتن نقطه‌ای روی خط برای x و y معلوم (پیش‌بینی) و مقایسه‌ی آن با نتیجه‌ی واقعی (آزمون) بررسی می‌کنیم. برنامه خط را تغییر شکل می‌دهد تا برازش بهتری ایجاد کند، تا زمانی که به سطح قابل قبولی از خطا برسد. این مثالی از رگرسیون خطی ساده است. یادگیری ماشین این تکنیک را گرفته و آن را بر اساس چرخه‌ی آزمون، پیش‌بینی، تنظیم و آزمون مجدد، خودکار کرده است.
بیایید به طبقه بندی یک کاسه میوه در کلاس‌های مختلف بر اساس آموزش نظارت شده نگاه کنیم.

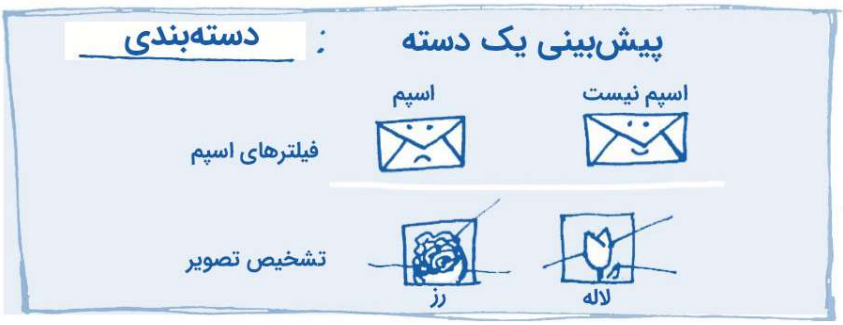


در این مثال، ظرفی حاوی چهار نوع میوه است. یک انسان داده‌های ورودی (اندازه و رنگ) و داده‌های خروجی (شکل) را برچسب‌گذاری می‌کند. الگوریتم از مجموعه داده‌ی آموزشی برای یافتن خوشه‌هایی از اشکال مشابه استفاده می‌کند، سپس یک انسان این نتایج را با استفاده از مجموعه‌ی داده‌ی دیگری، مجموعه‌ای که برای اهداف آزمایشی نگه داشته شده بود، آزمایش می‌کند. انسان مدل را برای عملکرد بهینه تنظیم می‌کند، داده‌های جدید را جمع‌آوری می‌کند و از مدل برای پیش‌بینی استفاده می‌کند. یک گیلاس، گلابی نیست، اما دو گیلاس که ساقه‌هایشان هنوز به هم متصل است، به عنوان گیلاس شناخته می‌شوند.

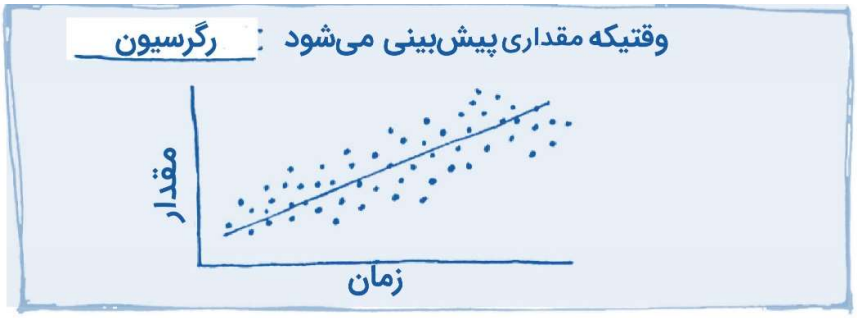
سه کاربرد رایج یادگیری نظارت‌شده وجود دارد:

۱۳- برازش (Fitting) به زبان ساده یعنی پیدا کردن بهترین تطابق یا انطباق بین یک مدل (مانند یک خط، منحنی یا تابع ریاضی) و یک سری داده. فرض کنید شما یک سری نقطه روی یک نمودار دارید. برازش یعنی پیدا کردن خط یا منحنی‌ای که به بهترین شکل از بین این نقاط عبور کند یا به آن‌ها نزدیک باشد. به عبارت دیگر، برازش به دنبال یافتن مدلی است که داده‌های موجود را به خوبی توصیف کند. این مدل می‌تواند یک خط مستقیم (در رگرسیون خطی)، یک منحنی (در برازش منحنی) یا هر نوع تابع ریاضی دیگری باشد.

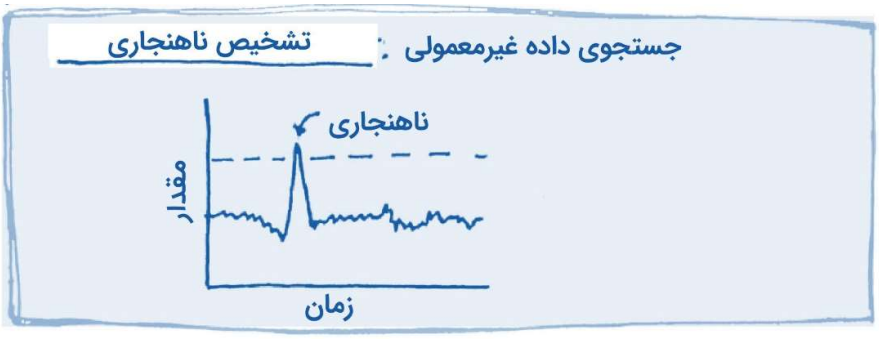
طبقه بندی :زمانی که داده‌ها برای پیش بینی یک دسته استفاده می‌شوند، به عنوان مثال، فیلترهای اسپم و تشخیص تصویر مانند تشخیص بین لاله یا رز



رگرسیون :زمانی که یک مقدار پیش بینی می‌شود، به عنوان مثال، قیمت سهام



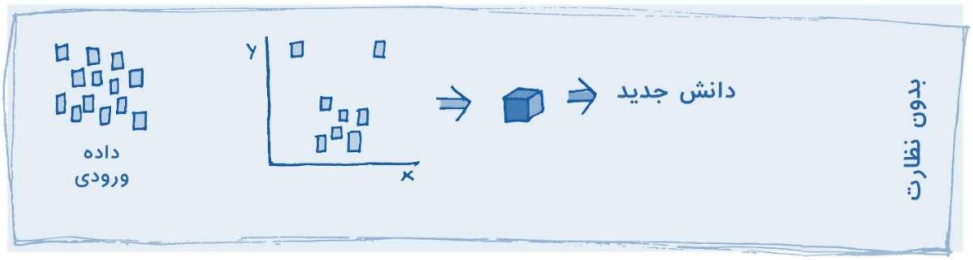
تشخیص ناهنجاری :زمانی که به دنبال داده‌های غیرعادی است، به عنوان مثال، تشخیص تقلب



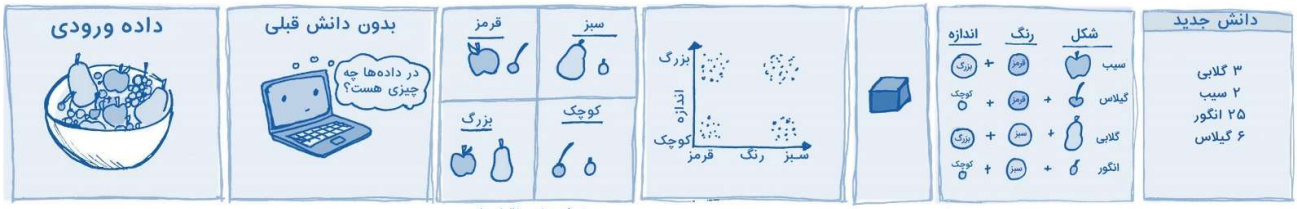
یادگیری بدون نظارت

یادگیری بدون نظارت از رویکردهای یادگیری متفاوت و فرآیندهای ریاضی مختلف برای دستکاری داده‌ها و ارائه نتیجه استفاده می‌کند.

در یادگیری بدون نظارت، نقاط داده هیچ برچسبی ندارند و در عوض، هدف سازماندهی داده‌ها بر اساس شباهت و درک ساختار آن است. هیچ نتیجه شناخته شده یا پاسخ صحیحی وجود ندارد و یک مدل با استنتاج ساختارها در داده‌ها (به جای آزمایش در برابر پاسخ‌های صحیح) آماده می‌شود. یک هدف مهم یادگیری بدون نظارت این است که ماشین را وادار به یافتن الگوهای داده‌ای کند که انسان‌ها از آنها اطلاعی ندارند.



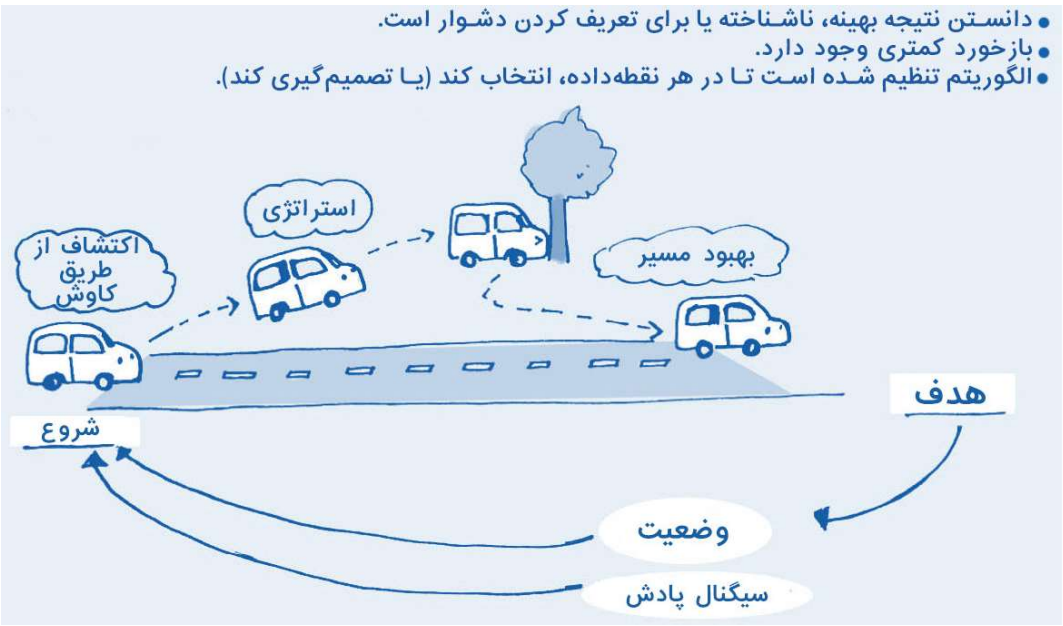
بیایید به کاسه میوه برگردیم. در یادگیری بدون نظارت، ماشین ویژگی‌هایی مانند اندازه، رنگ و شکل را کشف می‌کند. ماشین داده‌ها را بر اساس ویژگی‌هایی که پیدا کرده است خوشه بندی می‌کند و به عنوان خروجی می‌تواند چهار نوع میوه مختلف و همچنین تعداد هر کدام را در هر کلاس مشخص کند.



یادگیری تقویتی

یادگیری تقویتی در موقعیت‌هایی استفاده می‌شود که نتایج بهینه مشخص نیستند یا تعریف آن‌ها دشوارتر است و بازخورد کمتری وجود دارد. این بازخورد ممکن است بسیار دیرتر در فرآیند ارائه شود و حتی اطلاعاتی درباره مراحل خود فرآیند ندهد. مراحل باید از طریق کاوش کشف شوند یعنی آزمایش استراتژی‌های مختلف و اندازه‌گیری عملکرد بر مبنای هدف بلندمدت.

در یادگیری تقویتی، الگوریتم طوری تنظیم می‌شود که در پاسخ به هر داده، تصمیماتی بگیرد. مدت کوتاهی بعد، الگوریتم اطلاعاتی درباره ارزش تصمیمی که گرفته است یک «پاداش» دریافت می‌کند، سپس برای اصلاح استراتژی خود از آن استفاده می‌کند. یادگیری تقویتی به طور معمول در رباتیک و نرم‌افزارهای مربوط به خودروهای خودران یا اتوماتیک به کار می‌رود.



تعریف مسیر بهینه برای یک وسیله نقلیه جهت حرکت امکان‌پذیر نیست. در دنیا جاده‌ها و موانع زیادی وجود دارند که تعریف نتیجه مثبت در هر حالت را دشوار می‌کنند. اما می‌توانیم موفقیت را ماندن در جاده و شکست را خارج شدن از جاده تعریف کنیم. با استفاده از یادگیری تقویتی، خودرو وقتی در جاده بماند پاداش می‌گیرد و وقتی از جاده خارج شود هیچ پاداشی نمی‌گیرد. خودرو با توجه به بازخوردی که در شرایط مختلف دریافت می‌کند، مدل خود را برای نحوه حرکت و رسیدن به موفقیت تعریف‌شده، یعنی ماندن در جاده، ایجاد می‌کند.

آموزش ماشین برای یادگیری

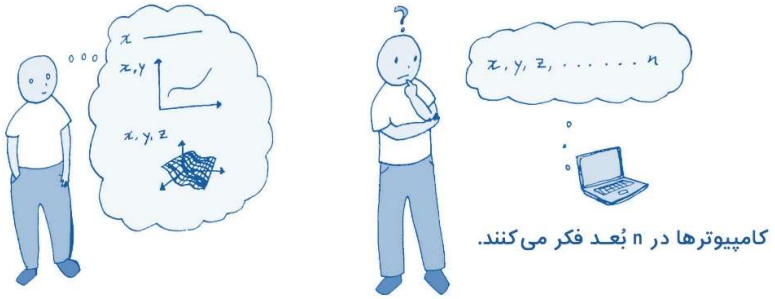
شهود و تجربه انسان قدرتمند است. ما در انتقال دانش و مفاهیم از یک حوزه به حوزه دیگر ماهر هستیم. هنگامی‌که مصرف انرژی را با قدرت پردازش مقایسه می‌کنیم، ما متفکران کارآمدی هستیم. سریعترین ابررایانه جهان می‌تواند ۹۳ کوادریلیون محاسبه در ثانیه انجام دهد، در حالی که اندازه گیری معادل برای مغز انسان ۳٫۵ کوادریلیون محاسبه در ثانیه است. اما وقتی نوبت به انرژی می‌رسد، ابررایانه ۱۵٫۳ میلیون وات مصرف می‌کند در حالی که مغز ما بخش کوچکی از این مقدار، فقط ۲۰ وات، مصرف می‌کند[۱۴]. هنگام مقایسه اتصالات، مغز انسان حدود ۹۰ میلیارد سلول دارد که با تریلیون‌ها اتصال به هم متصل شده اند. وقتی چنین مغزی به صورت دیجیتالی شبیه سازی شد، دانشمندان برای بازتاب فقط ۱ ثانیه از فعالیت طبیعی مغز انسان به بیش از ۸۲۰۰۰ پردازنده نیاز داشتند که روی یکی از سریعترین ابررایانه‌های جهان کار می‌کردند. این بدان معناست که سریعترین ابررایانه جهان با ۱۰٫۶۶ میلیون پردازنده فقط می‌تواند فعالیت مغزی ۱۳۰ نفر را در یک زمان شبیه سازی کند. اما ما از مجموعه‌ای از سوگیری‌های شناختی رنج می‌بریم؛ گرایش‌هایی به تفکر به شیوه‌های خاص که می‌تواند منجر به انحراف سیستماتیک و تکرارپذیر از آنچه ریاضیات می‌گوید، شود. چند مثال رایج:

اثر لنگر (Anchoring effect): تمایل به تکیه بیش از حد بر یک قطعه اطلاعات هنگام تصمیم‌گیری که معمولاً اولین قطعه از اطلاعات است. خرده‌فروشان از این اثر زمانی استفاده می‌کنند که یک قیمت «اصلی» به طور مصنوعی بالا را با تخفیف نشان می‌دهند و باعث می‌شوند احساس کنیم معامله خوبی انجام می‌دهیم[۱۵].

در دسترس‌پذیری (Availability heuristic): تمایل به بیش از حد تخمین زدن احتمال وقوع یک رویداد به دلیل اینکه خاطرات مربوط به آن رویداد، تازه یا غیرمعمول یا از نظر احساسی پرشور هستند. اگر شما دغدغه زیادی در مورد تصادفات رانندگی داشته باشید، احتمالاً این امر درک شما از احتمال تصادف را افزایش می‌دهد[۱۶].

توهم خوشه‌بندی (Clustering illusion): تمایل به ارزیابی بیش از حد اهمیت الگوهای پیوسته یا خوشه‌ها در نمونه‌های تصادفی بزرگ داده‌ها و در نتیجه، دیدن الگوهای خیالی[۱۷].

میانبرهای ذهنی بسیار بیشتری وجود دارند که در نحوه ارزیابی داده‌ها، اختلال ایجاد می‌کنند. یک محدودیت مهم دیگر این است که ما نمی‌توانیم بیش از چند متغیر را در ذهن خود در هر زمان معین نگه داریم.



۱۴- به بیان ساده، ابرکامپیوتر حدوداً ۲۶ برابر سریع‌تر از مغز انسان، محاسبه اما ۷۶۵ هزار برابر بیشتر از مغز انسان انرژی مصرف می‌کند.

۱۵- فرض کنید می‌خواهید یک لپ‌تاپ بخرید. وارد یک فروشگاه می‌شوید و اولین لپ‌تاپی که می‌بینید، با قیمت ۲۰ میلیون تومان برچسب خورده است. سپس لپ‌تاپ دیگری را می‌بینید که ۱۶ میلیون تومان قیمت دارد. به احتمال زیاد، لپ‌تاپ دوم را "ارزان" و "به صرفه" تلقی می‌کنید، زیرا قیمت آن را با اولین لپ‌تاپی که دیدید (لنگر) مقایسه می‌کنید. در حالی که شاید قیمت واقعی و منصفانه لپ‌تاپ دوم، ۱۴ میلیون تومان باشد. فروشنده با قرار دادن لپ‌تاپ ۲۰ میلیونی، ذهن شما را روی قیمت بالاتری "لنگر" می‌اندازد تا لپ‌تاپ ۱۶ میلیونی، معامله خوبی به نظر برسد.

۱۶- فرض کنید اخیراً اخبار زیادی در مورد سقوط هواپیما شنیده‌اید. در این صورت، ممکن است احساس کنید که احتمال سقوط هواپیما برای شما بسیار زیاد است، حتی اگر آمار نشان دهد که سفر با هواپیما بسیار امن‌تر از سفر با خودرو است. دلیل این امر این است که خاطرات مربوط به سقوط هواپیما به دلیل تازگی و پوشش رسانه‌ای گسترده، به راحتی در ذهن شما "در دسترس" هستند و باعث می‌شوند که احتمال وقوع آن را بیش از حد واقعی تخمین بزنید.

۱۷- فرض کنید در حال پرتاب سکه هستید. بعد از چند بار پرتاب، به این ترتیب نتیجه می‌گیرید: رو، رو، پشت، پشت، رو، رو. ممکن است فکر کنید که یک الگوی خاص وجود دارد و پرتاب بعدی حتماً "پشت" خواهد بود. در حالی که هر پرتاب سکه، یک رویداد مستقل است و هیچ ارتباطی با پرتاب‌های قبلی ندارد. توهم خوشه‌بندی باعث می‌شود که در داده‌های تصادفی، الگوهای خیالی ببینیم و اهمیت رگه‌ها یا خوشه‌های تصادفی را بیش از حد ارزیابی کنیم.

"اگر مردم می‌توانستند ابعاد بالاتر را ببینند، یادگیری ماشین ضرورتی نداشت."

- پدرو دومینگوس

بیایید به مثال پیش‌بینی قیمت خانه برگردیم و یک آزمایش فکری انجام دهیم. فرض کنید همسایه‌ی شما خانه‌اش را به قیمت ۵۰۰,۰۰۰ دلار می‌فروشد. خانه‌ی او یک سرویس بهداشتی و دو اتاق خواب دارد؛ خانه‌ی شما دو سرویس بهداشتی و چهار اتاق خواب دارد و دو برابر بزرگتر است. حالا با توجه به این اطلاعات، پیش‌بینی شما از قیمت خانه‌تان چقدر خواهد بود؟ شاید حدود ۸۰۰,۰۰۰ دلار؟ آیا متوجه می‌شوید که نسبتاً راحت است که سه عامل مختلف (تعداد حمام، اتاق خواب و اندازه) را در ذهن خود نگه دارید و حداقل به یک محدوده قیمتی برسید؟ حالا فرض کنید عوامل بیشتری را اضافه کنیم: خانه‌ی شما ده سال قدیمی‌تر است و کمی فرسوده شده در حالی که خانه‌ی همسایه اخیراً بازسازی شده است، شما یک حیاط دارید ولی حیاط او طراحی مینیمالیستی مدرن دارد، خانه‌ی شما یک ایوان خوب دارد و نمای بهتری دارد ولی خانه‌ی او یک استخر شنا و شومینه‌ی بیرونی دارد. آیا حالا متوجه می‌شوید که دیگر نمی‌توانید همه‌ی این متغیرها را دنبال کنید، چه برسد به اینکه یک پیش‌بینی دقیق جدید انجام دهید؟

یادگیری ماشین، حوزه‌ای علمی‌است که در آن الگوریتم‌های ریاضی روابط خاص بین متغیرهای متعدد را محاسبه می‌کنند. بسیاری از این الگوریتم‌ها بر اساس روش‌های آماری هستند و همه آنها نوعی روش مرتب‌سازی تکراری یا خودکار دارند که به الگوریتم اجازه می‌دهد به یک راه‌حل بهینه، همگرا شود.

در حقیقت صدها الگوریتم مختلف وجود دارد، از جمله تغییرات متعدد بر اساس تغییرات دیگر، و این حوزه به سرعت در حال تغییر است زیرا افراد روش‌های هوشمندانه‌ای برای ترکیب رویکردهای مختلف پیدا می‌کنند. من تصمیم گرفتم از یک دسته‌بندی ساده استفاده کنم که توسط پدرو دومینگوس، یکی از برترین پژوهشگران یادگیری ماشین در دنیا، توسعه یافته است. وی در کتاب خود با عنوان الگوریتم اصلی، الگوریتم‌ها را به پنج دسته [یا پنج قبیله] تقسیم می‌کند.

استنتاج قیاسی معکوس (Inverse deduction): در این روش، الگوریتم به صورت تکراری (iteratively)عمل می‌کند تا شکاف‌های موجود در دانش فعلی را پر کند.

فرض کنید شما یک پازل دارید که چند قطعه آن گم شده است. روش استنتاج قیاسی معکوس مانند این است که شما با توجه به قطعات موجود و شکل کلی پازل، سعی می‌کنید حدس بزنید که قطعات گم‌شده چگونه بوده‌اند و جای خالی آن‌ها را پر کنید. این کار به صورت تکراری انجام می‌شود، یعنی ابتدا یک حدس می‌زنید، سپس آن را بررسی می‌کنید و در صورت نیاز حدس خود را اصلاح می‌کنید تا به جواب درست برسید.

شبکه‌های عصبی و یادگیری عمیق (Neural networks and deep learning): در این رویکرد، یک شبکه ریاضی از ساختار مغز انسان تقلید می‌کند.

این روش از نحوه کارکرد مغز انسان الهام گرفته است. مغز ما از شبکه‌ای از سلول‌های عصبی (نورون‌ها) تشکیل شده که با هم ارتباط دارند. شبکه‌های عصبی مصنوعی هم به همین شکل هستند، یعنی از واحدهای محاسباتی ساده‌ای تشکیل شده‌اند که به هم متصل هستند و با دریافت داده‌ها، الگوها و روابط بین آن‌ها را یاد می‌گیرند. یادگیری عمیق نوع پیشرفته‌تری از شبکه‌های عصبی است که از لایه‌های بیشتری از این واحدها استفاده می‌کند و به همین دلیل می‌تواند مسائل پیچیده‌تر را حل کند.

تکاملی (Evolutionary): در این روش، الگوریتم نتایج را به گونه‌ای مرتب و غربال می‌کند که به نوعی شبیه‌ساز تکثیر DNA باشد.

این روش از فرایند تکامل در طبیعت الهام گرفته است. در طبیعت، موجودات زنده با گذشت زمان و از طریق جهش‌های ژنتیکی و انتخاب طبیعی، تغییر می‌کنند و به گونه‌ای سازگارتر با محیط خود تبدیل می‌شوند. الگوریتم‌های تکاملی هم به همین شکل عمل می‌کنند، یعنی ابتدا مجموعه‌ای از راه‌حل‌های تصادفی برای یک مسئله ایجاد می‌کنند، سپس بهترین آن‌ها را انتخاب می‌کنند و با

ایجاد تغییرات تصادفی در آن‌ها (مانند جهش ژنتیکی)، نسل جدیدی از راه‌حل‌ها را تولید می‌کنند. این فرایند تکرار می‌شود تا به یک راه‌حل بهینه برسیم.

بیزی (Bayesian): در این رویکرد، الگوریتم از نوعی نظریه احتمال برای حل مسئله کاهش عدم قطعیت استفاده می‌کند.

این روش بر اساس نظریه احتمال بیز کار می‌کند. این نظریه به ما کمک می‌کند تا با داشتن اطلاعات اولیه (دانش قبلی) و دریافت اطلاعات جدید، احتمال وقوع یک رویداد را محاسبه کنیم. الگوریتم‌های بیزی هم به همین شکل عمل می‌کنند، یعنی با داشتن یک دانش اولیه در مورد مسئله، با دریافت داده‌های جدید، دانش خود را به‌روز می‌کنند و احتمال درست بودن فرضیات خود را محاسبه می‌کنند. این روش به خصوص در مواردی که با عدم قطعیت روبرو هستیم، بسیار مفید است.

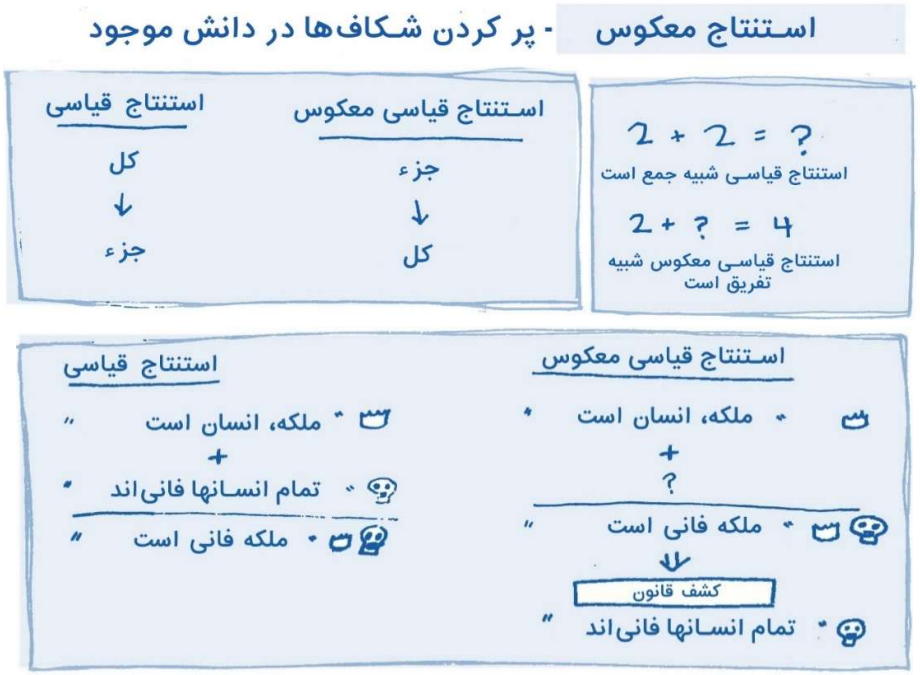
قیاس‌گرها (Analogizers): در این روش، الگوریتم، تضاد بین مجموعه‌های اطلاعات قدیمی و جدید را کاهش می‌دهد.

این روش بر اساس پیدا کردن شباهت‌ها بین مسائل مختلف کار می‌کند. فرض کنید شما یک مسئله جدید دارید که قبلاً با آن روبرو نشده‌اید. اگر بتوانید این مسئله را با یک مسئله قدیمی که قبلاً حل کرده‌اید، مقایسه کنید و شباهت‌هایی بین آن‌ها پیدا کنید، می‌توانید از راه‌حل مسئله قدیمی برای حل مسئله جدید استفاده کنید. الگوریتم‌های قیاس‌گر هم به همین شکل عمل می‌کنند، یعنی با پیدا کردن شباهت‌ها بین داده‌های جدید و داده‌های قدیمی، سعی می‌کنند الگوها و روابط موجود در داده‌های قدیمی را به داده‌های جدید تعمیم دهند.

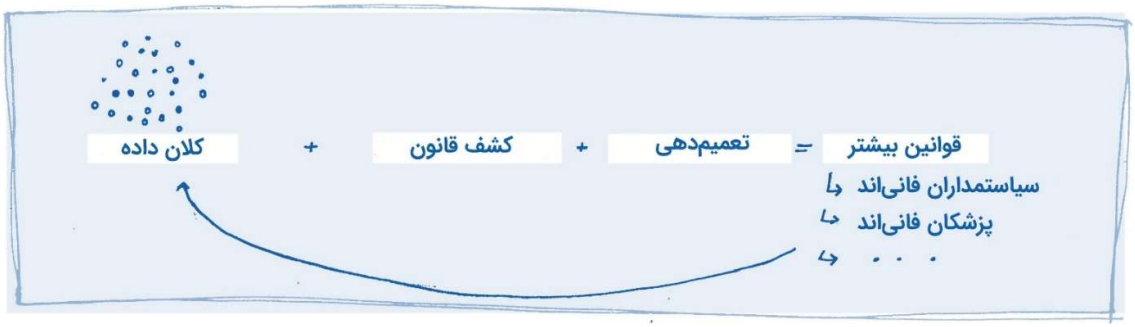
استنتاج قیاسی معکوس

استنتاج معکوس اساساً از نتایج یا «یادگیری از طریق مثال» به عقب برمی‌گردد. یک الگوریتم شامل برخی مقدمات معلوم یا مفروض است و از سیستم می‌پرسد «چه دانشی کم است؟»

این الگوریتم‌ها می‌توانند به طور خودکار، قوانینی را برای توجیه مشاهدات با تلاش برای یافتن الگوهای کلی و سپس استنتاج یک قاعده، ایجاد کنند. یعنی ورودی‌هایی انتخاب یا شناسایی می‌شوند که در تقسیم مجموعه داده‌ها به بخش‌های مشابه بهترین عملکرد را داشته باشند. از بسیاری جهات، استنتاج معکوس، روشی علمی را به ذهن متبادر می‌کند، زیرا مشاهدات خاص از طریق آزمایش و تکرار در برابر یک فرضیه، به یک قاعده کلی‌تر مرتبط می‌شوند.



در استنتاج معکوس، مجموعه‌داده‌های بسیار بزرگ می‌توانند با تعمیم هر قاعده در کل مجموعه داده، حجم زیادی از دانش جدید را به دست آورند. در بسیاری از حوزه‌های علمی، داده‌ها بسیار سریع‌تر از آنچه که قابل تحلیل باشند، تولید می‌شوند. ماشین‌ها قادرند به‌طور خودکار فرضیه‌هایی را ایجاد کنند که مشاهدات را توضیح می‌دهند و سپس آزمایش‌هایی را برای تولید داده‌های بیشتر آغاز کنند، بدین ترتیب فرضیه را آزمایش کرده و سپس این چرخه را تکرار می‌کنند. این فرآیند برای خودکارسازی فرایند علمی مورد استفاده قرار گرفته و کاربردهای عملی زیادی در بسیاری از حوزه‌ها دارد، به‌عنوان مثال، در تعیین عملکرد ژن‌ها و تسریع در کشف دارو با خودکارسازی فرآیند «فهرست کوتاه[۱۸]» ساختارهای مولکولی ایمن و مؤثر. این امر می‌تواند سال‌ها از زمان‌بندی توسعه را کاهش دهد، زیرا بسیاری از گزینه‌های بالقوه از فرایند تولید و سپس آزمایش‌های حیوانی و انسانی حذف می‌شوند.



بسیاری از سیستم‌های هوش مصنوعی، گونه‌هایی از استنتاج هستند. درخت‌های تصمیم در بسیاری از سیستم‌های رایج تعبیه شده‌اند، مانند اینکه چگونه یک بانک تصمیم می‌گیرد وام‌بدهد یا اینکه چگونه یک تماس از طریق یک مرکز خدمات مشتری به مسئول مربوطه هدایت شود. بزرگترین محدودیت آنها انعطاف پذیری است. قانون، قانون است و محاسبه و استفاده از آنها با این روش‌ها بسیار زیاد است.

شبکه‌های عصبی و یادگیری عمیق

شبکه‌های عصبی از ساختار و عملکرد مغز الهام گرفته شده‌اند. انواع مختلفی از شبکه‌های عصبی وجود دارد که از طرح‌های «نورونی» متفاوت، توابع آماری تعبیه‌شده و ترفندهای محاسباتی استفاده می‌کنند. در حالی که آنها مدتها پیش اختراع شده بودند، در چند سال اخیر پیشرفت‌های قابل توجهی در آنها حاصل شده است. شبکه‌های عصبی، اساس یادگیری عمیق هستند که در حال حاضر فعال‌ترین و متنوع‌ترین شاخه توسعه یادگیری ماشین است.

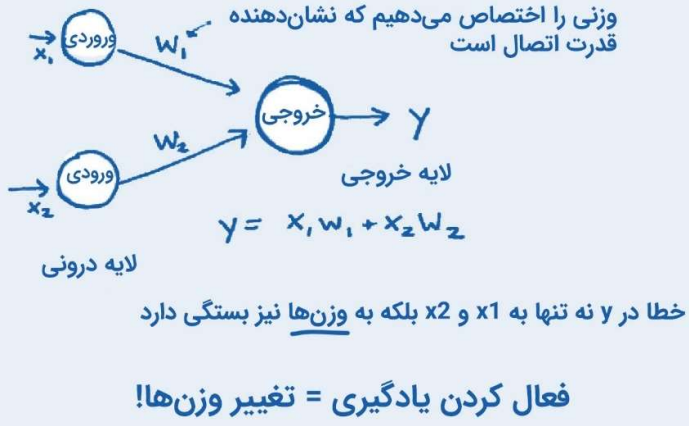
۱۸- short list (فهرست کوتاه) شامل یک گروه کوچک از گزینه‌ها یا مواردی است که بعد از بررسی اولیه و کنار گذاشتن موارد دیگر، انتخاب شده‌اند. به عبارت دیگر، فهرست کوتاه به مجموعه‌ای از بهترین یا امیدوارکننده‌ترین گزینه‌ها اشاره دارد که برای بررسی یا اقدام نهایی انتخاب می‌شوند. برای مثال، در زمینه کشف دارو، فهرست کوتاه به مجموعه‌ای از مولکول‌های خاص اشاره دارد که از بین تعداد زیادی گزینه، به عنوان ماده ایمن و مؤثر شناخته شده‌اند و برای آزمایش‌های بیشتر انتخاب می‌شوند.

اجزای یک شبکه عصبی چیست؟

آنچه که با هم آتش می‌گیرد، با هم سیم‌کشی می‌شود[۱۹].

یک نورون[۲۰] زیستی، سلولی است که از نظر الکتریکی تحریک‌پذیر است و می‌تواند سیگنال‌ها را به سلول‌های دیگر منتقل کند. سیگنال‌هایی که از سلول‌های دیگر (از طریق سیناپس[۲۱]) دریافت می‌شوند، ممکن است محرک[۲۲] یا بازدارنده[۲۳] باشند. اگر سیگنال محرک که یک نورون در یک دوره زمانی کوتاه دریافت می‌کند به اندازه کافی بزرگ باشد، نورون یک پالس کوتاه، به نام پتانسیل عمل[۲۴]، تولید می‌کند که سپس سیناپس‌ها و نورون‌های دیگر را فعال می‌کند. بنابراین، آنچه در پاسخ به یک محرک خاص رخ می‌دهد، به قدرت سیگنال‌ها و پیکربندی شبکه بستگی دارد.

یک نورون ریاضی[۲۵] نیز ورودی‌هایی دریافت می‌کند. به هر ورودی یک وزن[۲۶] اختصاص داده می‌شود. وزن بالاتر، محرک است و وزن پایین‌تر، بازدارنده است. هر نورون می‌تواند ورودی‌های متعددی دریافت کند که هرکدام وزن متفاوتی دارند. قدرت ارتباطات، «فعال‌سازی[۲۷]» نامیده می‌شود و به‌عنوان مجموع وزنی[۲۸] فعال‌سازی‌های تمام نورون‌هایی که به آن ورودی می‌دهند، محاسبه می‌شود. تابع فعال‌سازی[۲۹] درون نورون، ورودی را دریافت کرده و فعال‌سازی خروجی را محاسبه می‌کند. این خروجی به‌عنوان ورودی برای نورون‌های دیگر در بالاترین سطح شبکه استفاده می‌شود.



در متن بالا، «فعال‌سازی» به زبان ساده به این معنی است که هر نورون (سلول عصبی مصنوعی) در شبکه عصبی، ورودی‌هایی را از نورون‌های دیگر دریافت می‌کند و این ورودی‌ها با اوزان خاصی ضرب می‌شوند. این اوزان نشان‌دهندهٔ اهمیت هر ورودی هستند: وزن بیشتر یعنی ورودی تأثیر بیشتری دارد (محرک است) و وزن کمتر یعنی تأثیر کمتری دارد (بازدارنده است).

۱۹- در علوم اعصاب neurons that fire together wire together به این مفهوم که هرقدر شما یک مدار نورونی را در مغز خود بیشتر اجرا کنید، آن جریان قوی تر می شود، یعنی هرقدر که بیشتر پیانو تمرین کنید، به یک زبان صحبت کنید یا شعبده بازی کنید این جریان های نورونی قوی‌تر می شوند.

۲۰ نورون (Neuron): سلول عصبی که واحد اصلی سیستم عصبی است و وظیفه انتقال پیام‌های الکتریکی و شیمیایی را بر عهده دارد.

۲۱ سیناپس (Synapse): محل اتصال بین دو نورون که در آن انتقال سیگنال‌ها صورت می‌گیرد.

۲۲ محرک (Excitatory): سیگنالی که احتمال فعال شدن نورون بعدی را افزایش می‌دهد.

۲۳ بازدارنده (Inhibitory): سیگنالی که احتمال فعال شدن نورون بعدی را کاهش می‌دهد.

۲۴ پتانسیل عمل (Action potential): یک تغییر ناگهانی و کوتاه در پتانسیل الکتریکی غشای نورون که به عنوان سیگنال برای انتقال پیام عصبی عمل می‌کند.

۲۵ نورون ریاضیاتی/مصنوعی (Mathematical/Artificial Neuron): مدلی ساده‌شده از نورون بیولوژیکی که در شبکه‌های عصبی مصنوعی استفاده می‌شود.

۲۶ وزن (Weight): عددی که نشان‌دهنده اهمیت یا قدرت یک ورودی است. وزن‌های مثبت نشان‌دهنده ورودی‌های محرک و وزن‌های منفی نشان‌دهنده ورودی‌های بازدارنده هستند.

۲۷ فعال‌سازی (Activation): قدرت سیگنال یا خروجی یک نورون.

۲۸ مجموع وزن‌دار (Weighted sum): مجموع حاصل‌ضرب هر ورودی در وزن مربوط به آن.

۲۹ تابع فعال‌سازی (Activation function): تابعی ریاضی که ورودی (مجموع وزن‌دار ورودی‌ها) را به خروجی (فعال‌سازی نورون) تبدیل می‌کند. این تابع نقش مهمی در تعیین رفتار نورون دارد.

»فعال‌سازی» در واقع حاصل جمع این ورودی‌های وزن‌دار است. یعنی هر نورون، ابتدا تمام ورودی‌هایی که از نورون‌های قبلی گرفته را در وزن‌های مربوط به خودشان ضرب می‌کند و سپس همهٔ این حاصل‌ضرب‌ها را با هم جمع می‌کند. این جمع، «قدرت ارتباط» یا همان «فعال‌سازی» نامیده می‌شود.

به عبارت دیگر، «فعال‌سازی» نشان می‌دهد که یک نورون چقدر «تحریک» شده است. فرض کنید یک نورون سه ورودی با وزن‌های ۲، ۱ و ۰.۵ دریافت کند. اگر مقادیر ورودی‌ها به ترتیب ۳، ۴ و ۲ باشند، فعال‌سازی به این صورت محاسبه می‌شود:

(۳ ۲) + (۴ ۱) + (۲ ۰.۵) = ۶ + ۴ + ۱ = ۱۱

عدد ۱۱ نشان‌دهندهٔ میزان فعال شدن این نورون است.

اما این پایان کار نیست. نورون یک «تابع فعال‌سازی» هم دارد. این تابع، مقدار فعال‌سازی (در مثال بالا ۱۱) را به عنوان ورودی می‌گیرد و یک خروجی جدید تولید می‌کند. این خروجی، سپس به عنوان ورودی برای نورون‌های لایهٔ بعدی شبکه استفاده می‌شود.

تابع فعال‌سازی نقش مهمی در تعیین خروجی نورون دارد. این تابع می‌تواند خطی یا غیرخطی باشد و انواع مختلفی دارد (مانند سیگموئید، ReLUو غیره). وظیفهٔ اصلی تابع فعال‌سازی این است که تصمیم بگیرد آیا نورون باید «فعال» شود (یعنی خروجی قابل توجهی داشته باشد) یا «غیرفعال» بماند (یعنی خروجی ناچیزی داشته باشد).

به طور خلاصه، «فعال‌سازی» در یک نورون به دو مرحله تقسیم می‌شود:

۱. محاسبهٔ مجموع وزن‌دار ورودی‌ها: ورودی‌ها در وزن‌هایشان ضرب شده و با هم جمع می‌شوند. این حاصل جمع، «قدرت ارتباط» یا «فعال‌سازی» اولیه نامیده می‌شود.
۲. اعمال تابع فعال‌سازی: تابع فعال‌سازی روی مقدار فعال‌سازی اولیه اعمال می‌شود و خروجی نهایی نورون را تعیین می‌کند.

بنابراین، «فعال‌سازی» نشان‌دهندهٔ میزان تحریک یک نورون و خروجی نهایی آن است که به نورون‌های بعدی ارسال می‌شود.

برای تجسم ورودی‌ها و خروجی‌ها، تصور کنید می‌خواهید با استفاده از یک ظرف رنگ آبی و یک ظرف رنگ قرمز، یک ظرف رنگ بنفش درست کنید. مقدار هر رنگی که انتخاب می‌کنید، وزن‌ها هستند و تفاوت بین رنگ مطلوب و خروجی واقعی، خطا است. اگر رنگ مطلوب قرمزتر باشد، وزن قرمز را افزایش می‌دهید، اگر رنگ مطلوب بنفش‌تر باشد، وزن آبی را افزایش می‌دهید. نورون‌ها خودشان شامل توابع ریاضی هستند که می‌توانند ماهیت خروجی را تغییر دهند. آن‌ها می‌توانند شامل توابع پله‌ای با رفتارهای ساده روشن/خاموش یا توابع آماری باشند که امکان تغییر تدریجی بیشتری را با پاسخ نورون به ورودی‌ها فراهم می‌کنند. فشار انگشت خود را روی انتهای یک شلنگ آبیاری تصور کنید و اینکه چگونه می‌توان با تنظیم فشار و زاویه انگشت خود، حالت پاشش آب را از خاموش به روشن و سپس به یک فواره کامل تنظیم کرد.

نورون‌ها به صورت لایه لایه[۳۰] به هم متصل می‌شوند تا یک شبکه را تشکیل دهند. هیچ اتصالی درون لایه‌ها وجود ندارد، فقط اتصال بین لایه‌ها[۳۱] وجود دارد. لایه‌هایی که خروجی آن‌ها فقط به عنوان ورودی برای نورون‌های دیگر استفاده می‌شود، لایه‌های پنهان[۳۲] نامیده می‌شوند. فعال‌سازی‌ها[۳۳] در لایه‌های بالاتر توسط ورودی‌های لایه‌های پایین‌تر، تا لایه خروجی محاسبه می‌شوند. ساده‌ترین شبکه عصبی، یک شبکه پیش‌خور[۳۴] است.

---

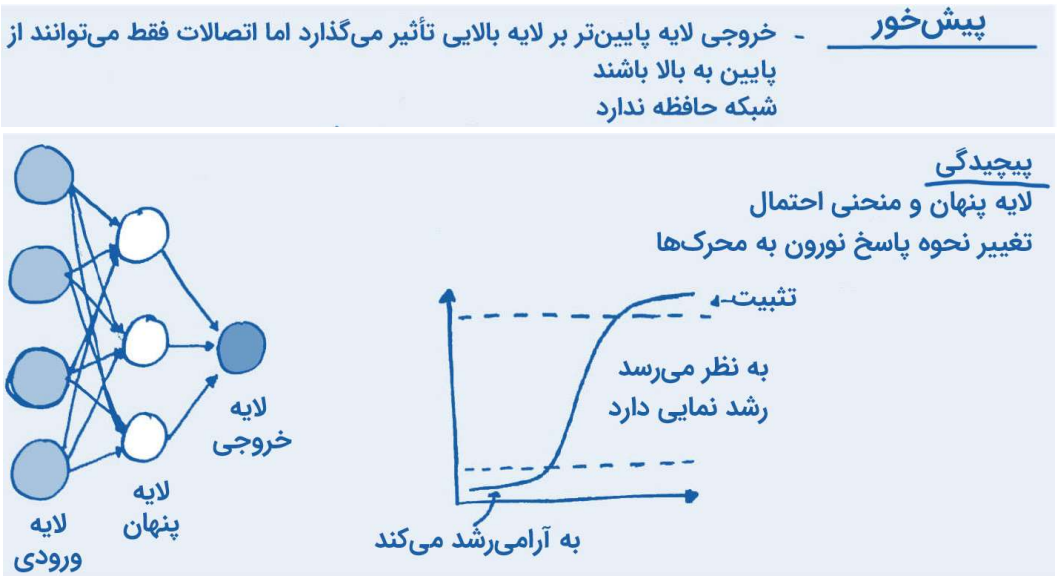
۳۰ لایه (Layer): دسته‌ای از نورون‌ها که به صورت موازی عمل می‌کنند. در یک شبکه عصبی، نورون‌ها در لایه‌های مختلف سازماندهی می‌شوند.

۳۱ اتصال بین لایه‌ها :(Connections between layers) ارتباط بین نورون‌های لایه‌های مختلف. این اتصالات با وزن‌هایی همراه هستند که قدرت تأثیر یک نورون بر نورون دیگر را مشخص می‌کنند.

۳۲ لایه‌های پنهان (Hidden layers): لایه‌هایی که بین لایه ورودی و لایه خروجی قرار دارند و مستقیماً با دنیای بیرون در ارتباط نیستند. این لایه‌ها وظیفه استخراج ویژگی‌های پیچیده از داده‌ها را بر عهده دارند.

۳۳ فعال‌سازی (Activations): خروجی یک نورون که به عنوان ورودی برای نورون‌های لایه‌های بعدی استفاده می‌شود.

۳۴ شبکه پیش‌خور (Feed forward network): نوعی از شبکه عصبی که در آن اطلاعات فقط در یک جهت، از لایه ورودی به لایه‌های پنهان و سپس به لایه خروجی جریان دارد و هیچ بازگشتی یا حلقه‌ای وجود ندارد.



شبکه پیش‌خور، حافظه ندارد[۳۵].

شبکه چگونه یاد می‌گیرد؟

یک اختراع مهم در حوزه شبکه‌های عصبی، پس‌انتشار (backpropagation) یا «بک‌پروپ[۳۶]» بود. بک‌پروپ مخفف عبارت «انتشار معکوس خطاها[۳۷]» است و برای رسیدن به آن از برخی مفاهیم حساب دیفرانسیل و انتگرال استفاده می‌شود. زمانی که داده‌ها به شبکه‌های عصبی وارد می‌شوند، خروجی می‌تواند یک معیار خطا تولید کند. در شبکه‌های عصبی، این خطا به شکل یک عبارت ریاضی است که می‌تواند برای تنظیم وزن‌ها در شبکه با جستجوی تدریجی خطای کمتر استفاده شود. این خطا به نام تابع زیان[۳۸] (loss function) شناخته می‌شود و برای تنظیم وزن‌ها از طریق بازگشت در شبکه استفاده می‌شود. در پایان این فرآیند، شبکه تمامی وزن‌ها را بهینه[۳۹] کرده و قادر خواهد بود ورودی‌ها را به خروجی‌ها به‌درستی نگاشت کند. پس‌انتشار یک تکنیک مهم است زیرا به این معنی است که الگوریتم به‌طور خودکار وزن‌های مدل را تغییر می‌دهد و بدین ترتیب از طریق خودتصحیحی و بازخورد[۴۰] «یاد می‌گیرد».

---

۳۵ این ویژگی‌ها باعث می‌شوند که شبکه‌های پیش‌خور برای مسائلی که نیاز به پردازش اطلاعات به صورت مرحله‌ای و بدون وابستگی به داده‌های قبلی دارند، مناسب باشند. به عنوان مثال، دسته‌بندی تصاویر یا تشخیص الگوهایی که وابستگی زمانی ندارند. در مقابل، برای مسائلی که نیاز به پردازش داده‌های ترتیبی یا دارای وابستگی زمانی دارند (مانند پردازش زبان طبیعی یا پیش‌بینی سری‌های زمانی)، شبکه‌های بازگشتی (Recurrent Neural Networks) مناسب‌تر هستند.

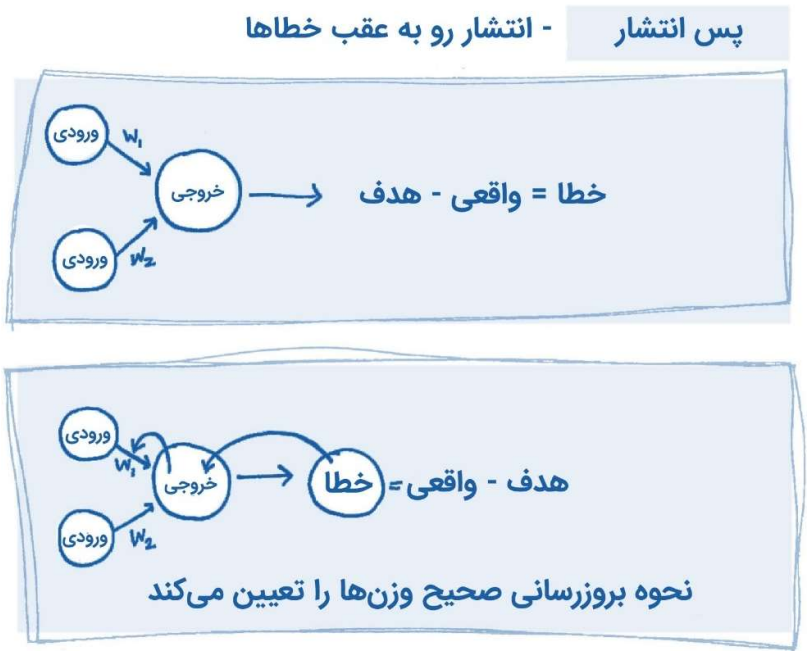
۳۶ پس‌انتشار/پس‌بروپ (Backpropagation/Backprop): یک الگوریتم مهم در یادگیری ماشین، به ویژه در آموزش شبکه‌های عصبی مصنوعی است که برای محاسبه گرادیان تابع زیان نسبت به وزن‌های شبکه استفاده می‌شود. سپس این گرادیان برای به‌روزرسانی وزن‌ها و کاهش خطا استفاده می‌شود.

۳۷ انتشار وارونه خطاها (Backward propagation of errors): نام کامل الگوریتم پس‌انتشار که به نحوه محاسبه و انتشار خطا از لایه خروجی به لایه‌های قبلی اشاره دارد.

۳۸ تابع زیان (Loss function): یک تابع ریاضی که میزان اختلاف بین خروجی پیش‌بینی شده توسط مدل و خروجی واقعی را اندازه‌گیری می‌کند. هدف در آموزش مدل، کمینه‌کردن این تابع است.

۳۹ تابع زیان :(Loss function) یک تابع ریاضی که میزان اختلاف بین خروجی پیش‌بینی شده توسط مدل و خروجی واقعی را اندازه‌گیری می‌کند. هدف در آموزش مدل، کمینه‌کردن این تابع است.

۴۰ خود-اصلاحی و بازخورد :(Self-modification and feedback) اشاره به این دارد که الگوریتم پس‌انتشار با استفاده از بازخورد خطا، به طور خودکار وزن‌های شبکه را تنظیم می‌کند و به این ترتیب "یاد می‌گیرد."



فرض کنید شما می‌خواهید یک توپ را به هدف مشخصی پرتاب کنید. در ابتدا ممکن است پرتاب شما دقیق نباشد و توپ به هدف نخورد. برای بهبود دقت پرتاب، شما به طور غریزی تغییراتی در نحوه پرتاب خود (مانند زاویه پرتاب، قدرت پرتاب و غیره) ایجاد می‌کنید. اگر پرتاب شما خیلی کوتاه باشد، شما آن را قوی‌تر پرتاب می‌کنید و اگر خیلی بلند باشد، آن را آرام‌تر پرتاب می‌کنید. این فرآیند آزمون و خطا را آنقدر تکرار می‌کنید تا در نهایت توپ به هدف بخورد. الگوریتم پس‌انتشار در شبکه‌های عصبی دقیقاً مشابه این فرآیند عمل می‌کند. در اینجا:

- شبکه عصبی: مانند بدن شما است که توپ را پرتاب می‌کند.
- وزن‌های شبکه: مانند تنظیمات بدن شما (زاویه پرتاب، قدرت پرتاب و غیره) هستند که نتیجه پرتاب را تعیین می‌کنند.
- تابع زیان/خطا: مانند فاصله توپ از هدف است. هرچه فاصله بیشتر باشد، خطا بیشتر است.
- گرادیان: مانند راهنمایی است که به شما می‌گوید چگونه تنظیمات خود (وزن‌ها) را تغییر دهید تا خطا کاهش یابد.

به عبارت دیگر، الگوریتم پس‌انتشار به این صورت عمل می‌کند:

۱. ورودی: داده‌ها به شبکه عصبی داده می‌شوند و شبکه یک خروجی تولید می‌کند.
۲. محاسبه خطا: خروجی شبکه با خروجی واقعی (هدف) مقایسه می‌شود و میزان خطا (تابع زیان) محاسبه می‌شود.
۳. پس‌انتشار خطا: خطا به صورت معکوس در شبکه منتشر می‌شود و «گرادیان» برای هر وزن محاسبه می‌شود. گرادیان نشان می‌دهد که هر وزن چقدر در خطا نقش داشته است و چگونه باید تغییر کند تا خطا کاهش یابد.
٤. به‌روزرسانی وزن‌ها: با استفاده از گرادیان، وزن‌های شبکه به گونه‌ای تغییر می‌کنند که خطا کاهش یابد. این مرحله مانند تنظیم زاویه و قدرت پرتاب شما برای دقیق‌تر شدن پرتاب است.

این فرآیند بارها تکرار می‌شود تا شبکه عصبی به تدریج یاد بگیرد که خروجی‌های دقیق‌تری تولید کند و خطا به حداقل برسد. شاید متوجه چیزی شده باشید... بک‌پروپ[٤١] به یک خروجی شناخته‌شده نیاز دارد، چیزی که بتوان آن را حل کرد. به‌طور کلی، بک‌پروپ در یادگیری تحت نظارت (supervised learning) استفاده می‌شود. پس چگونه شبکه‌های عصبی در یک روش بدون نظارت (unsupervised) یاد می‌گیرند؟

آیا نمی‌توانید جنگل را در میان درختان ببینید؟

به دنبال سوزن در انبار کاه هستید؟

اطلاعات زیاد...

---

٤١ الگوریتم پس‌انتشار یک مثال کلاسیک از روش‌های یادگیری نظارت‌شده است. این الگوریتم با مقایسه خروجی پیش‌بینی شده با خروجی واقعی (برچسب)، خطا را محاسبه کرده و وزن‌های شبکه را برای کاهش خطا تنظیم می‌کند.

گاهی اوقات اطلاعات زیادی وجود دارد. گاهی کاهش مقدار اطلاعات و ساده‌سازی[۴۲] داده‌ها می‌تواند اطلاعات بیشتری را آشکار کند.

فرض کنید می‌خواهیم از روی چک‌لیستی از سه علامت - سرفه، دمای بالا، درد مفاصل - پیش‌بینی کنیم که آیا فردی آنفولانزا دارد یا خیر. مقدار «۱» را به «بله» اختصاص می‌دهیم و زمانی که فرد حداقل دو مورد از این علائم را داشته باشد، او را به عنوان بیمار در نظر می‌گیریم.

اما می‌خواهیم چیزهای دیگری را نیز در مورد بیمار بدانیم که ممکن است به این معنی باشد که احتمال بیمار بودن او کمتر است، به عنوان مثال، اینکه واکسن آنفولانزا زده باشد یا قرص ویتامین D مصرف کرده باشد یا اخیراً ورزش کراس‌فیت سنگین را شروع کرده باشد (که می‌تواند دلیلی بر احساس درد باشد). در اینجا علائم مخالف[۴۳] را به همین ترتیب برچسب‌گذاری می‌کنیم یعنی «۱» برای «بله» و «۰» برای «خیر». زمانی که بیمار حداقل دو مورد از این موارد را داشته باشد، او را سالم در نظر می‌گیریم.

حالا این ایده را در یک شبکه عصبی قرار می‌دهیم. یک لایه ورودی و یک لایه خروجی، هر کدام با ۶ ویژگی داده وجود دارد: سرفه، دمای بالا، درد مفاصل، واکسن آنفولانزا، ویتامین D، کراس‌فیت سنگین. اگر فقط دو نورون در لایه پنهان[۴۴] داشته باشیم، لایه پنهان فقط می‌تواند پیام «بیمار» یا «سالم» را به جلو بفرستد. در طول تکرارهای متوالی، دو واحد پنهان مجبور می‌شوند حساسیت‌های (فعال‌سازی‌های) متفاوتی[۴۵] نسبت به ورودی‌ها داشته باشند. نتیجه نهایی این است که ما شبکه‌ای ایجاد کرده‌ایم که یک نمایش فشرده[۴۶] از داده‌ها را یاد گرفته است در حالی که پیش‌بینی‌های دقیقی در مورد اینکه آیا فردی بر اساس علائم و سابقه خود بیمار است یا سالم، انجام می‌دهد.

---

۴۲ کاهش ابعاد/ساده‌سازی داده‌ها (Reducing the amount of information/Simplifying the data): ایده اصلی این بخش این است که گاهی اوقات با کاهش تعداد ویژگی‌ها یا متغیرها، می‌توان به درک بهتری از داده‌ها رسید و از پیچیدگی بیش از حد جلوگیری کرد.

۴۳ علائم و علائم‌مخالف (Symptoms and counter symptoms): منظور از «علائم» نشانه‌هایی است که احتمال بیماری را افزایش می‌دهند (سرفه، تب، درد مفاصل) و منظور از «علائم مخالف» عواملی است که احتمال بیماری را کاهش می‌دهند (واکسن، ویتامینD ، کراس‌فیت).

۴۴ لایه پنهان با دو نورون (Hidden layer with two neurons): محدود کردن لایه پنهان به دو نورون، شبکه را مجبور می‌کند که اطلاعات را به صورت فشرده‌تر و با تمرکز بر ویژگی‌های اصلی‌تر ذخیره کند.

۴۵ حساسیت‌های متفاوت (Different sensitivities/activations): به این معنی است که هر نورون در لایه پنهان به ترکیب متفاوتی از ورودی‌ها حساس خواهد بود و نقش متفاوتی در تشخیص «بیمار» یا «سالم» ایفا خواهد کرد.

۴۶ نمایش فشرده (Compressed representation): نتیجه نهایی این فرآیند، یادگیری یک نمایش فشرده از داده‌ها توسط شبکه است. به عبارت دیگر، شبکه با استفاده از تعداد کمتری نورون در لایه پنهان، قادر به ذخیره و پردازش اطلاعات مهم است.

در یادگیری ماشین، ترفندهای[۴۷] زیادی وجود دارد که برای کشف الگو یا رابطه‌ای به کار می‌رود که انسان قادر به دیدن آن نیست. بسیاری از تکنیک‌ها به دنبال تجزیه یک چیز پیچیده به اجزای کوچک‌تر برای یافتن معنا هستند. اما ترفندهای دیگری نیز وجود دارند که با مجبور کردن رایانه به فشرده‌سازی اطلاعات[۴۸]، تجزیه آن و دور ریختن بخش‌هایی از آن و سپس مجبور کردن دوباره آن به بازسازی شکل اصلی، قوانین پنهان را کشف می‌کنند.

---

۴۷ ترفندها (Tricks): منظور از ترفندها در این متن، روش‌های ابداعی و غیرمستقیمی است که برای حل مسائل پیچیده یادگیری ماشین استفاده می‌شود.

۴۸ فشرده‌سازی اطلاعات (Compressing information): به معنای کاهش ابعاد داده‌ها و حذف اطلاعات اضافی به گونه‌ای است که اطلاعات مهم حفظ شود

هنگامی که این کار را داخل یک شبکه عصبی انجام می‌دهیم، ساختار حاصل، حاوی خودرمزگذارها[۴۹] (Autoencoders) است (یعنی یک شبکه عصبی بدون نظارت که به صورت معکوس کار می‌کند) که ابتدا خروجی را تجزیه می‌کند و سپس ورودی‌های صحیح را بازسازی می‌کند. با مجبور کردن ماشین به تجزیه اطلاعات و سپس بازسازی آن، شبکه، همبستگی‌هایی[۵۰] را در ساختار ساده‌سازی شده پیدا می‌کند. مزیت انجام این کار فقط مربوط به قدرت پردازش یا حافظه نیست، بلکه برای امر یادگیری، اساسی است: شبکه به تنهایی نحوه فشرده‌سازی اطلاعات را کشف می‌کند و از یک تصویر نویزدار و مخدوش، یک تصویر تمیز ایجاد می‌کند.

خودرمزگذار (AUTOENCODER):
یک شبکه عصبی بدون نظارت است که برای یادگیری رمزگذاری‌های کارآمد استفاده می‌شود
(لایه خروجی تعداد گره‌های مشابه با لایه ورودی دارد)
برای پیش‌بینی مقدار هدف Y با ورودی‌های X آموزش داده نمی‌شود،
بلکه برای بازسازی ورودی‌های خود (X) آموزش داده می‌شود

به زبان ساده‌تر:

خودرمزگذار مانند یک دستگاه کپی عمل می‌کند که تلاش می‌کند یک کپی دقیق از ورودی خود ایجاد کند. اما این کپی‌برداری به صورت مستقیم انجام نمی‌شود. خودرمزگذار ابتدا ورودی را به یک نمایش فشرده (مانند یک کد فشرده) تبدیل می‌کند و سپس از این نمایش فشرده، سعی در بازسازی ورودی اصلی دارد. این فرآیند دو بخش اصلی دارد:

۱. رمزگذار (Encoder): ورودی را به یک نمایش فشرده (کد) تبدیل می‌کند. این بخش معمولاً شامل چندین لایه عصبی است که ابعاد داده‌ها را کاهش می‌دهند.
۲. رمزگشا (Decoder): از نمایش فشرده (کد) برای بازسازی ورودی اصلی استفاده می‌کند. این بخش نیز معمولاً شامل چندین لایه عصبی است که ابعاد داده‌ها را افزایش می‌دهند.

هدف از این کار چیست؟

هدف اصلی خودرمزگذار، یادگیری ویژگی‌های مهم و اصلی داده‌ها و حذف نویز و اطلاعات اضافی است. با مجبور کردن شبکه به فشرده‌سازی و سپس بازسازی داده‌ها، شبکه مجبور می‌شود تا مهم‌ترین ویژگی‌ها را شناسایی و حفظ کند. این ویژگی‌ها می‌توانند برای کارهای مختلفی مانند کاهش ابعاد داده‌ها، حذف نویز، تولید داده‌های جدید و غیره استفاده شوند.

به عنوان مثال، فرض کنید می‌خواهیم تصاویر چهره را فشرده کنیم. یک خودرمزگذار می‌تواند با یادگیری ویژگی‌های اصلی چهره (مانند شکل چشم‌ها، بینی و دهان)، یک نمایش فشرده از هر تصویر ایجاد کند. سپس با استفاده از این نمایش فشرده، می‌تواند تصویر اصلی را با دقت قابل قبولی بازسازی کند.

در نتیجه، خودرمزگذار یک ابزار قدرتمند برای یادگیری نمایش‌های فشرده و کارآمد از داده‌ها است که در زمینه‌های مختلفی از جمله پردازش تصویر، پردازش زبان طبیعی و غیره کاربرد دارد.

یک جمع‌بندی...

اکنون می‌دانیم که چگونه خروجی یک نورون تحت تأثیر وزن ورودی‌ها قرار می‌گیرد،

و می‌دانیم که چگونه خروجی نورون‌ها می‌تواند به جلو حرکت کند و ورودی برای نورون‌های سطح بالاتر باشد،

و می‌دانیم که چگونه خروجی یک نورون می‌تواند با استفاده از پس‌انتشار (بک‌پروپ) آزمایش، ارزیابی و اصلاح شود،

و می‌دانیم که لایه‌های پنهان می‌توانند داده‌ها را در سطح مفهومی یاد بگیرند و ویژگی‌های اصلی داده‌هایی که ما قادر به دیدن آن‌ها نیستیم را شبیه‌سازی کنند.

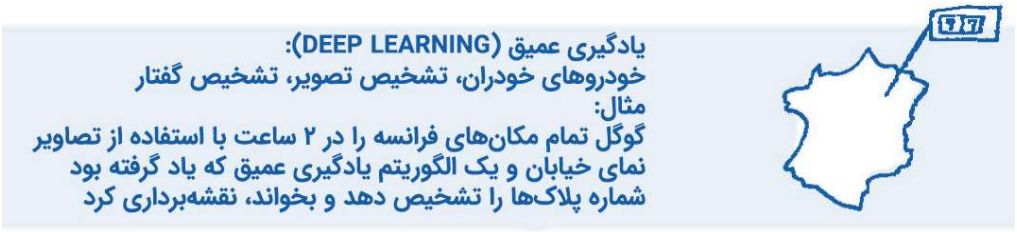
---

۴۹ خودرمزگذار :(Autoencoder) نوع خاصی از شبکه عصبی بدون نظارت است که سعی می‌کند ورودی را دریافت کرده و خروجی‌ای تولید کند که تا حد ممکن شبیه به ورودی اصلی باشد. در این فرآیند، شبکه مجبور می‌شود ویژگی‌های کلیدی ورودی را شناسایی کرده و آن‌ها را در یک نمایش فشرده (کد) ذخیره کند. سپس با استفاده از این کد، سعی در بازسازی ورودی اصلی می‌کند.

۵۰ همبستگی‌ها :(Correlations) روابط و وابستگی‌های بین ویژگی‌های مختلف داده‌ها است.

پس، اگر لایه‌های بیشتری را روی هم قرار دهیم چه اتفاقی می‌افتد؟ این چیزی است که به آن یادگیری عمیق گفته می‌شود.

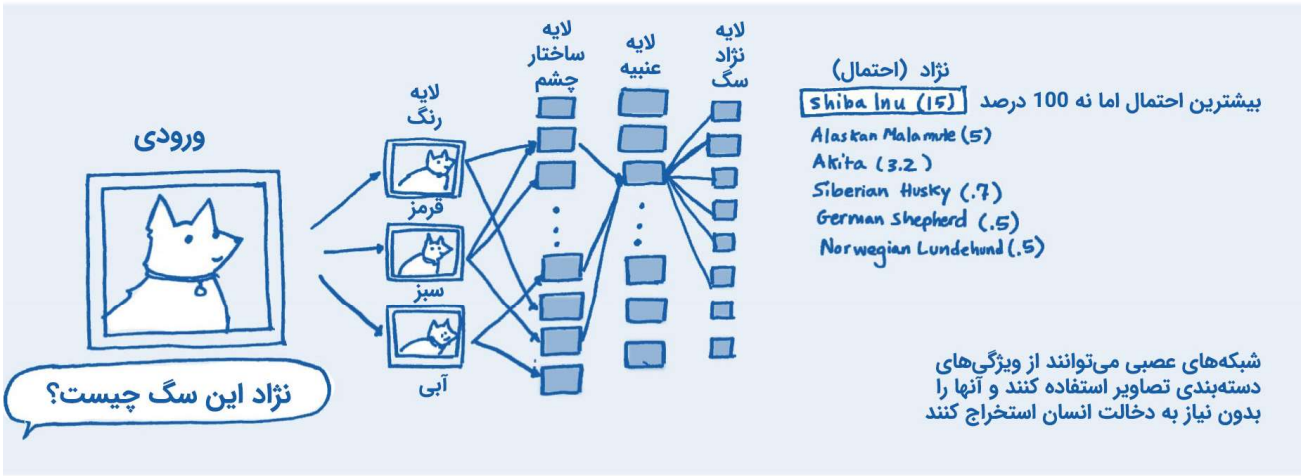
یادگیری عمیق[51]، یک به‌روزرسانی جدید برای شبکه‌های عصبی است. در یادگیری عمیق، لایه‌های بسیار بیشتری از نورون‌ها به هم متصل می‌شوند و هر لایه می‌تواند به نوبه خود آموزش داده شود. هر لایه پنهان، پیچیده‌تر و پیچیده‌تر می‌شود و می‌تواند نمایش‌های پیچیده‌تر یا انتزاعی‌تری را رمزگذاری کند. برای مثال، اگر یک شبکه مجموعه بزرگی از تصاویر چهره را دریافت کند، اولین خودرمزگذار یاد می‌گیرد که ویژگی‌های کوچک را در یک ناحیه محلی از تصویر (مانند یک گوشه یا یک خط یا یک نقطه) را رمزگذاری کند. لایه بعدی این را به عنوان ورودی می‌گیرد و ویژگی سطح بعدی مانند لبه یک چشم را یاد می‌گیرد. در نهایت، لایه بالایی می‌تواند همه اینها را کنار هم قرار دهد و مثلا تصویر را به عنوان تصویری از ملکه تشخیص دهد.



یادگیری عمیق می‌تواند تحت نظارت یا بدون نظارت باشد. در یادگیری تحت نظارت، شبکه آموزش می‌بیند و با تنظیم وزن‌ها و تمرکز بر پاسخ صحیح، یاد می‌گیرد.

در یادگیری بدون نظارت، مدل مقادیر عظیمی از داده‌های ورودی غنی را می‌گیرد و از تمام ترفندهایی که دارد برای رمزگذاری آنچه می‌بیند، استفاده می‌کند.

در حالی که بزرگترین دستاوردهای عملکردی با یادگیری نظارت‌شده به دست آمده است، امروزه توانایی یادگیری عمیق برای کشف ویژگی‌های جدید برای خود، در خط مقدم هوش مصنوعی قرار دارد.



شبکه عصبی توسط پیکسل‌های یک تصویر فعال می‌شود ‹ نورون‌ها وزن‌دهی می‌شوند و توسط یک تابع f(x) تبدیل می‌شوند ‹ فعال‌سازی این نورون‌ها به نورون‌های دیگر (لایه‌ها) منتقل می‌شود ‹ نورون خروجی که نژاد را تعیین می‌کند، فعال می‌شود

ماشین‌ها چگونه می‌بینند؟

اگر ماشین‌هایی داشته باشیم که بتوانند اطلاعات بصری را پردازش کنند، می‌توانیم گستره‌ی کارهایی که با ماشین‌ها می‌توانیم انجام دهیم را متحول کنیم. فقط به انبوه عظیم اطلاعات بصری که اکنون به صورت آنلاین به آن دسترسی داریم و آنچه بیشتر می‌تواند ایجاد شود فکر کنید؛ تصاویر پزشکی، تصاویر فضایی، هر تعداد تصویر از افراد، مکان‌ها، چیزها. ماشین‌ها در پردازش و طبقه‌بندی تصاویر، سریع‌تر، ارزان‌تر و در بسیاری از موارد، دقیق‌تر از انسان‌ها هستند.

51- Deep Learning

یادگیری عمیق این حوزه را متحول کرده است، عمدتاً به این دلیل که در کشف ساختارهای پیچیده در داده‌های چندبعدی بسیار خوب است. تصاویر، حاوی ویژگی‌های بسیاری با تفاوت‌های ظریف اما مهم هستند که کدگذاری آن‌ها به روش سنتی غیرممکن بوده است.

یکی از روش‌هایی که ماشین‌ها با آن یاد می‌گیرند تا ببینند، استفاده از نوعی شبکه عصبی به نام شبکه عصبی کانولوشن (CNN یا ConvNet) است. شبکه عصبی کانولوشن نوعی شبکه عصبی پیش‌خور است که از ساختار قشر بینایی الهام گرفته شده است. این شبکه‌ها شبیه مجموعه‌های عظیمی از فیلترها عمل می‌کنند که اطلاعات ورودی را از طریق فیلترها و لایه‌های پیچیده‌تر به‌طور پیوسته پردازش می‌کنند.

شبکه‌های عصبی کانولوشن عناصر طراحی مهمی دارند که آنها را به ویژه در بینایی کامپیوتر مفید می‌سازد. آنها به طور منحصر به فرد دارای موارد زیر هستند:

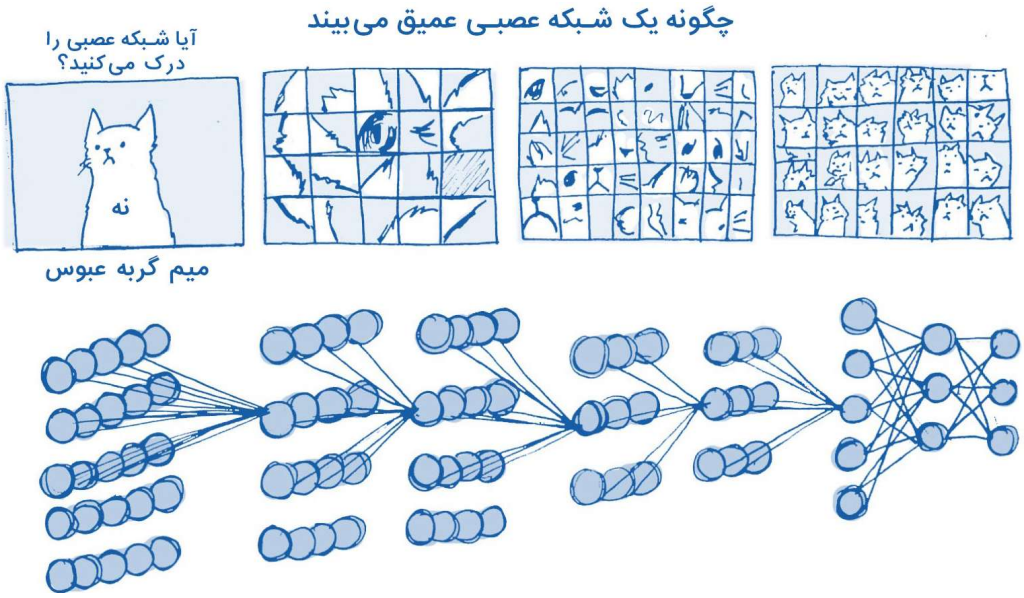
اتصالات محلی: که آنها را قادر می‌سازد به راحتی مقادیر همبسته بالا را تشخیص دهند و موتیف‌های محلی متمایز مانند لبه چشم را تشکیل دهند.

وزن‌های مشترک: به طوری که یک الگو پس از تشخیص در بخش دیگری از آرایه، راحت‌تر قابل تشخیص باشد.

لایه‌های تجمیع (Pooling): که می‌توانند از نظر ریاضی مسائل را ساده کنند و در نتیجه تعداد ابعاد را کاهش دهند و به کاهش حساسیت شبکه به اعوجاج‌های کوچک کمک کنند.

مراحل کانولوشن (Convolution): یک فرآیند نمونه‌برداری فرعی که کارایی شبکه را افزایش می‌دهد.

هر تصویر به عنوان آرایه‌ای از مقادیر پیکسل به شبکه وارد می‌شود. در لایه اول، شبکه ویژگی‌های مرتبه پایین‌تر، معمولاً وجود یا عدم وجود لبه‌ها را یاد می‌گیرد. سپس لایه دوم ممکن است این لبه‌ها و جهت‌ها و مکان‌های آنها را در ترکیبی با مرتبه بالاتر قرار دهد تا با بخش‌هایی از اشیاء مطابقت داشته باشد. شبکه به «پیش‌خور کردن» ادامه می‌دهد، بنابراین لایه‌های بعدی نمایش اشیاء در تصویر را به سطح بالاتر و انتزاعی‌تر می‌سازند: از یک مرز رنگ به لبه یک چشم، و از لبه یک چشم به خود چشم.



پیشرفت بزرگ در شبکه‌های عصبی کانولوشن در سال ۲۰۱۲ رخ داد، زمانی که با در دسترس بودن تراشه‌های پردازشی قدرتمند، این شبکه‌ها روی مجموعه داده‌ای بسیار بزرگ‌تر از قبل اعمال شدند؛ حدود یک میلیون تصویر. نتیجه شگفت‌انگیز بود: نرخ خطای رویکردهای قبلی به نصف کاهش یافت. امروزه شبکه‌های عصبی کانولوشن روش غالب برای شناسایی و تشخیص هستند. این شبکه‌ها بسیار بزرگ‌اند و می‌توانند شامل ۱۰ تا ۲۰ لایه، صدها میلیون وزن و میلیاردها اتصال باشند.

یک مرور کلی ...

- اکنون می‌دانیم که خروجی یک نورون چگونه توسط وزن‌های ورودی‌های آن تأثیر می‌پذیرد .
- همچنین می‌دانیم که خروجی نورون‌ها می‌تواند به لایه‌های بالاتر تغذیه شود و به‌عنوان ورودی نورون‌های سطح بالاتر عمل کند.
- می‌دانیم که خروجی یک نورون می‌تواند توسط الگوریتم بازپخش خطا (Backpropagation) آزمایش، ارزیابی و اصلاح شود.
- درک کرده‌ایم که لایه‌های مخفی می‌توانند داده‌ها را در سطح مفهومی بیاموزند و ویژگی‌های اصلی داده‌ها را، حتی آن‌هایی که برای ما قابل مشاهده نیستند، استخراج کنند.
- و اکنون می‌دانیم که چگونه می‌توانیم لایه‌های بسیاری را روی هم قرار دهیم و شبکه‌های یادگیری عمیق ایجاد کنیم که قادر به انجام وظایف پیچیده‌ای مانند کمک به ماشین‌ها در «دیدن» باشند.

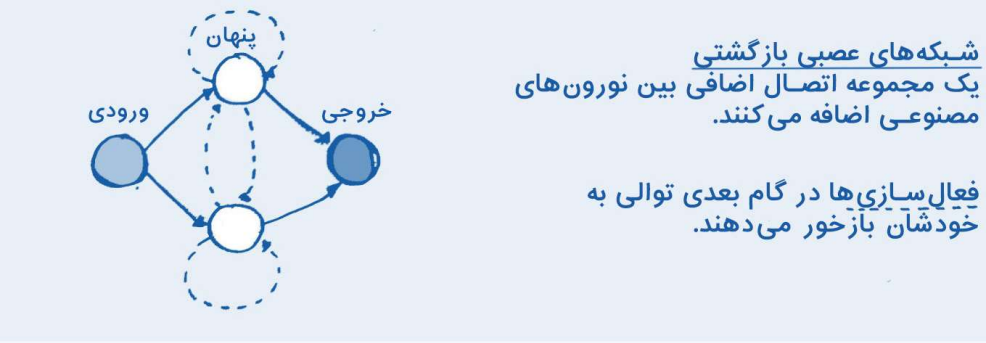
اما چه می‌شود اگر ماشین بتواند خروجی چندین تکرار را به خاطر بسپارد و از این دانش برای بهبود عملکرد خود استفاده کند؟ این همان چیزی است که به آن شبکه‌های عصبی بازگشتی (Recurrent Neural Networks) گفته می‌شود.

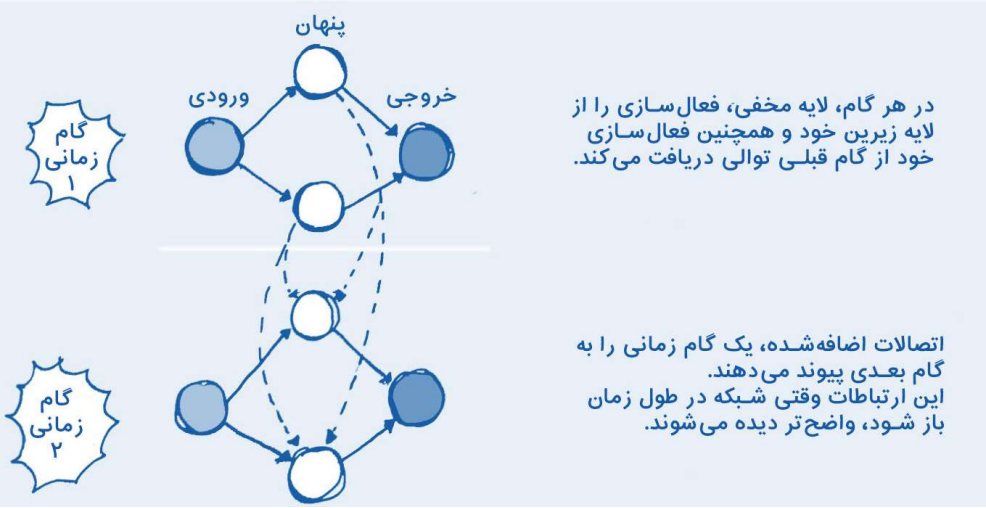
شبکه‌های عصبی بازگشتی[52] می‌توانند به جای یک طبقه‌بندی ساده، ورودی‌های متعددی را دریافت و خروجی‌های متعددی ارائه کنند. آنها قادر به کار با دنباله‌ها هستند، جایی که نورون فعال‌سازی قبلی خود را به "خاطر می‌آورد"، نوعی حافظه کوتاه مدت مصنوعی و همچنان قادر به دریافت ورودی تازه هم است.

در انواع دیگر شبکه‌های عصبی، ورودی‌ها و خروجی‌ها ثابت هستند، یعنی یک ورودی قرار می‌دهید و مجموعه ای از احتمالات کلاس‌های مختلف را دریافت می‌کنید. برای مثال، یک میلیون عکس از پرندگان قهوه ای وارد کنید و خروجی احتمالات نژادهای مختلف پرنده قهوه‌ای را دریافت کنید. شبکه‌های عصبی بازگشتی می‌توانند دنباله‌هایی از ورودی‌ها و خروجی‌ها را مدیریت کنند که آنها را بسیار انعطاف پذیر و سازگار می‌کند. به جای اینکه به ورودی‌ها و خروجی‌های ثابت محدود شوند، شبکه‌ها می‌توانند یاد بگیرند که خروجی متفاوتی ارائه دهند یا ورودی را اصلاح کنند. چه هوشمندانه!

به خاطر دارید که گفتم هیچ اتصالی بین نورون‌ها در یک لایه مشابه در یک شبکه عصبی وجود ندارد؟ شبکه‌های عصبی بازگشتی با اضافه کردن یک حلقه در زمان، نوعی لایه ایجاد می‌کنند. این روش هوشمندانه‌ای برای دادن «حافظه» به یک شبکه است، آنچنانکه تکرار قبلی را به خاطر می‌آورد و از آن برای به‌روزرسانی خروجی خود استفاده می‌کند.

52- RNN: Recurrent Neural Networks





این تصویر مفهوم شبکه‌های عصبی بازگشتی (RNN) را به صورت شماتیک نشان می‌دهد. این شبکه‌ها نوعی از شبکه‌های عصبی هستند که برای پردازش داده‌های ترتیبی (مانند متن، صدا، ویدیو و سری‌های زمانی) بسیار مناسبند، چرا که «حافظه» دارند و می‌توانند اطلاعات مراحل قبلی را به یاد بسپارند. برای توضیح ساده‌تر تصویر، آن را به دو بخش تقسیم می‌کنیم:

بخش اول: نمایش کلی یک RNN. این بخش، ساختار کلی یک شبکه عصبی بازگشتی را نشان می‌دهد. در اینجا چند نکته کلیدی وجود دارد:

ورودی (Input): داده‌ای که به شبکه داده می‌شود. مثلاً اگر بخواهیم یک جمله را پردازش کنیم، هر کلمه می‌تواند یک ورودی باشد.

لایه پنهان (Hidden): این لایه، قلب تپنده RNN است. در این لایه، پردازش اصلی روی داده‌ها انجام می‌شود و «حافظه» شبکه در اینجا ذخیره می‌شود.

خروجی (Output): نتیجه پردازش شبکه. مثلاً اگر بخواهیم کلمه بعدی در یک جمله را پیش‌بینی کنیم، خروجی، پیش‌بینی شبکه خواهد بود.

اتصالات بازگشتی (حلقه‌های فلش‌دار): این اتصالات، ویژگی اصلی RNNها هستند. آن‌ها نشان می‌دهند که خروجی لایه پنهان، دوباره به عنوان ورودی به خود لایه پنهان در مرحله بعدی برمی‌گردد. این بازخورد باعث می‌شود که شبکه، اطلاعات مراحل قبلی را به خاطر بسپارد.

به زبان ساده، فرض کنید شما در حال خواندن یک کتاب هستید. هر جمله‌ای که می‌خوانید، ورودی به مغز شما (لایه پنهان) است. مغز شما (به دلیل حافظه‌اش) اطلاعات جملات قبلی را نیز در نظر می‌گیرد تا جمله فعلی را بهتر درک کند. اتصالات بازگشتی در RNN دقیقاً همین کار را انجام می‌دهند.

بخش دوم: باز شدن RNN در طول زمان

این بخش، نحوه عملکرد RNN در طول زمان را نشان می‌دهد. برای درک بهتر، شبکه «باز» شده است و مراحل زمانی مختلف را به صورت جداگانه نشان داده‌ایم:

مرحله زمانی ۱ (Time Step 1): در این مرحله، اولین ورودی به شبکه داده می‌شود. لایه پنهان، پردازش را انجام می‌دهد و یک خروجی و همچنین یک «حالت پنهان» (که همان اطلاعات ذخیره شده در حافظه است) تولید می‌کند.
مرحله زمانی ۲ (Time Step 2): در این مرحله، ورودی دوم به شبکه داده می‌شود، اما نکته مهم اینجاست که «حالت پنهان» تولید شده در مرحله قبلی نیز به عنوان ورودی به لایه پنهان داده می‌شود. این یعنی شبکه، اطلاعات مرحله قبلی را به یاد دارد و از آن در پردازش مرحله فعلی استفاده می‌کند.
این روند در مراحل بعدی نیز تکرار می‌شود. به این ترتیب، RNNمی‌تواند وابستگی‌ها و الگوها را در طول زمان یاد بگیرد. به عنوان مثال، فرض کنید می‌خواهیم جمله «هوا امروز خیلی خوب است» را به شبکه بدهیم. در مرحله اول، کلمه «هوا» به شبکه داده می‌شود. در مرحله دوم، کلمه «امروز» به همراه اطلاعات مربوط به کلمه «هوا» (که در «حالت پنهان» ذخیره شده) به شبکه داده می‌شود. به همین ترتیب، شبکه در هر مرحله، اطلاعات کلمات قبلی را در نظر می‌گیرد تا کل جمله را درک کند.
به طور خلاصه، RNNها با داشتن اتصالات بازگشتی، حافظه دارند و می‌توانند داده‌های ترتیبی را به خوبی پردازش کنند. این ویژگی آن‌ها را برای کاربردهایی مانند ترجمه ماشینی، تشخیص گفتار، و تحلیل سری‌های زمانی بسیار مفید می‌سازد.
از نظر عملی، این کار آنها را قادر می‌سازد تا کارهای بسیار هوشمندانه ای مانند تولید خودکار عنوان برای یک تصویر و ترجمه همزمان زبان را انجام دهند.

یادگیری عمیق به کجا می‌رود؟
سه حوزه توسعه یادگیری عمیق به‌ویژه هیجان‌انگیز وجود دارند:
داده‌های پراکنده (Sparse Data): پیشرفت‌های یادگیری عمیق به دلیل دسترسی به مجموعه داده‌های کلان رخ داد. مشخص شده که بزرگ‌تر بودن داده‌ها برای شبکه‌های عصبی بهتر بوده است. اما این رویکرد «یک اندازه برای همه»[۵۳] ممکن است همیشه کارآمدترین راه نباشد. اکنون روش‌های جدیدی در حال توسعه هستند که به یادگیری عمیق اجازه می‌دهند حتی در شرایطی که داده‌های کمتری در دسترس است، کارایی خوبی داشته باشد.
شبکه‌های مولد تخاصمی (Generative Adversarial Networks): این ایده بر اصل «اگر نتوانی چیزی را بسازی، آن را درک نکرده‌ای» استوار است. در این حوزه، ایده این است که به‌طور همزمان دو شبکه عصبی آموزش داده شوند: یکی روی داده‌های واقعی آموزش می‌بیند و دیگری سعی می‌کند تصاویر مشابهی را تولید کند تا شبکه اول بتواند روی آن‌ها آموزش ببیند. این فرآیند شبیه یک مسابقه مچ‌اندازی در یادگیری ماشین است، یعنی شبکه «تولیدکننده» تصویر تلاش می‌کند شبکه «یادگیرنده» را بهبود دهد.
سخت‌افزار (Hardware): تمامی این پردازش‌ها از نظر محاسباتی بسیار پرهزینه هستند. تمرکز جدید بر محاسباتی که انرژی کمتری مصرف کند و بهینه‌سازی طراحی تراشه برای هوش مصنوعی، به این معناست که یادگیری عمیق اکنون در سخت‌افزار (سیلیکون) تعبیه شده است. واحد پردازش تانسور (TPU) گوگل، که به‌طور خاص برای یادگیری عمیق طراحی شده، مختص محصول یادگیری عمیق گوگل یعنی TensorFlow است. گوگل ادعا می‌کند که این فناوری تقریباً معادل پیش‌بُرد هفت ساله فناوری یا سه نسل از قانون مور است.

---

۵۳ منظور از "one size fits all" در مورد یادگیری عمیق این است که تکیه صرف بر داده‌های بسیار زیاد (massive datasets) به عنوان تنها راه حل برای بهبود عملکرد شبکه‌های عصبی، این فرض را دارد که هر چه داده بیشتر باشد، همیشه نتیجه بهتری حاصل می‌شود. در حالی که این فرض تا حدی درست است، اما همیشه کارآمدترین یا بهترین راه نیست. به عبارت دیگر، این رویکرد نادیده می‌گیرد که: (۱) جمع‌آوری داده‌های بسیار زیاد همیشه امکان‌پذیر نیست: در بسیاری از موارد، دسترسی به داده‌های حجیم یا محدود است یا هزینه بالایی دارد. (۲) داده‌های زیاد لزوماً به معنای داده‌های با کیفیت نیست: داده‌های زیاد ممکن است شامل نویز، داده‌های نامربوط یا داده‌های مغرضانه باشند که می‌توانند عملکرد شبکه را تحت تأثیر منفی قرار دهند. (۳) پردازش داده‌های بسیار زیاد نیازمند منابع محاسباتی زیادی است: آموزش شبکه‌های عصبی بر روی داده‌های حجیم نیازمند سخت‌افزار قدرتمند و زمان زیادی است.
بنابراین، رویکرد "one size fits all" در یادگیری عمیق به معنای تکیه صرف بر داده‌های بسیار زیاد بدون در نظر گرفتن روش‌های دیگر مانند یادگیری با داده‌های کم (sparse data) یا بهینه‌سازی الگوریتم‌ها است. رویکردهایی که به دنبال کارایی بیشتر با داده‌های کمتر هستند، تلاش می‌کنند تا از این محدودیت‌ها عبور کنند و راه حل‌های بهینه‌تری ارائه دهند.

یک جمع بندی نهایی...

اکنون می‌دانیم که خروجی یک نورون چگونه تحت تأثیر وزن ورودی‌های آن قرار می‌گیرد،
و می‌دانیم که چگونه خروجی نورون‌ها می‌تواند به لایه‌های بالاتر تغذیه شده و به‌عنوان ورودی برای نورون‌های سطح بالاتر عمل کند،
و می‌دانیم که خروجی یک نورون چگونه می‌تواند با استفاده از بک‌بروپ (Backpropagation) آزمایش، ارزیابی و اصلاح شود،
و می‌دانیم که لایه‌های مخفی می‌توانند داده‌ها را در سطح مفهومی بیاموزند و ویژگی‌های اصلی داده‌ها را که قابل مشاهده نیستند، استخراج کنند،
و می‌دانیم که چگونه می‌توانیم لایه‌های بسیاری را روی هم قرار دهیم و شبکه‌های یادگیری عمیقی ایجاد کنیم که قادر به انجام وظایف پیچیده‌ای مانند کمک به ماشین‌ها در «دیدن» باشند .
و می‌دانیم که زمانی که ماشین‌ها بتوانند خروجی چندین تکرار را به خاطر بسپارند و از این دانش برای بهبود عملکرد خود استفاده کنند، می‌توانیم رابطهای بسیار هوشمند و شبیه انسان بسازیم که بدون دخالت انسان توانایی بهبود خود را داشته باشند.

الگوریتم‌های تکاملی و ژنتیکی

الگوریتم‌های تکاملی، از اصل «بقای اصلح» برای فرآیند تکثیر DNA در یک شبیه‌سازی کامپیوتری تقلید می‌کنند. این الگوریتم‌ها با آزمایش در برابر یک نتیجه بهبود می‌یابند و به همین دلیل نوعی یادگیری تقویتی محسوب می‌شوند که آزمون و خطا را با سه مفهوم اصلی ترکیب می‌کنند:

- انتخاب (Selection): جایی که نتایج بهتر، بقا پیدا می‌کنند و پیشرفت می‌کنند.
- تقاطع (Crossover): جایی که بخش‌هایی از این نتایج خوب با هم ترکیب می‌شوند.
- جهش (Mutation): جایی که گاهی اوقات ترکیب‌های تصادفی و جدید رخ می‌دهند.

الگوریتم‌های تکاملی برای بهینه‌سازی در برابر یک تابع برازش (Fitness Function) طراحی می‌شوند که معمولاً توسط یک انسان تعریف می‌شود. برای مثال، درصد پیام‌های اسپم که به درستی فیلتر می‌شوند می‌تواند یک تابع برازش باشد.

به زبان ساده، این الگوریتم‌ها برای رسیدن به یک هدف خاص تلاش می‌کنند و این هدف توسط یک «تابع برازش» (fitness function) تعریف می‌شود.

حالا این «تابع برازش» چیست؟ فرض کنید شما می‌خواهید یک برنامه بنویسید که ایمیل‌های اسپم را تشخیص دهد. هدف شما این است که برنامه تا حد امکان ایمیل‌های اسپم را به درستی تشخیص دهد و از ورود آن‌ها به صندوق ورودی کاربر جلوگیری کند.

در اینجا، «تابع برازش» می‌تواند «درصد ایمیل‌های اسپمی که به درستی فیلتر می‌شوند» باشد. یعنی هر چقدر برنامه بهتر عمل کند و درصد بیشتری از ایمیل‌های اسپم را درست تشخیص دهد، مقدار «تابع برازش» بالاتر خواهد بود.

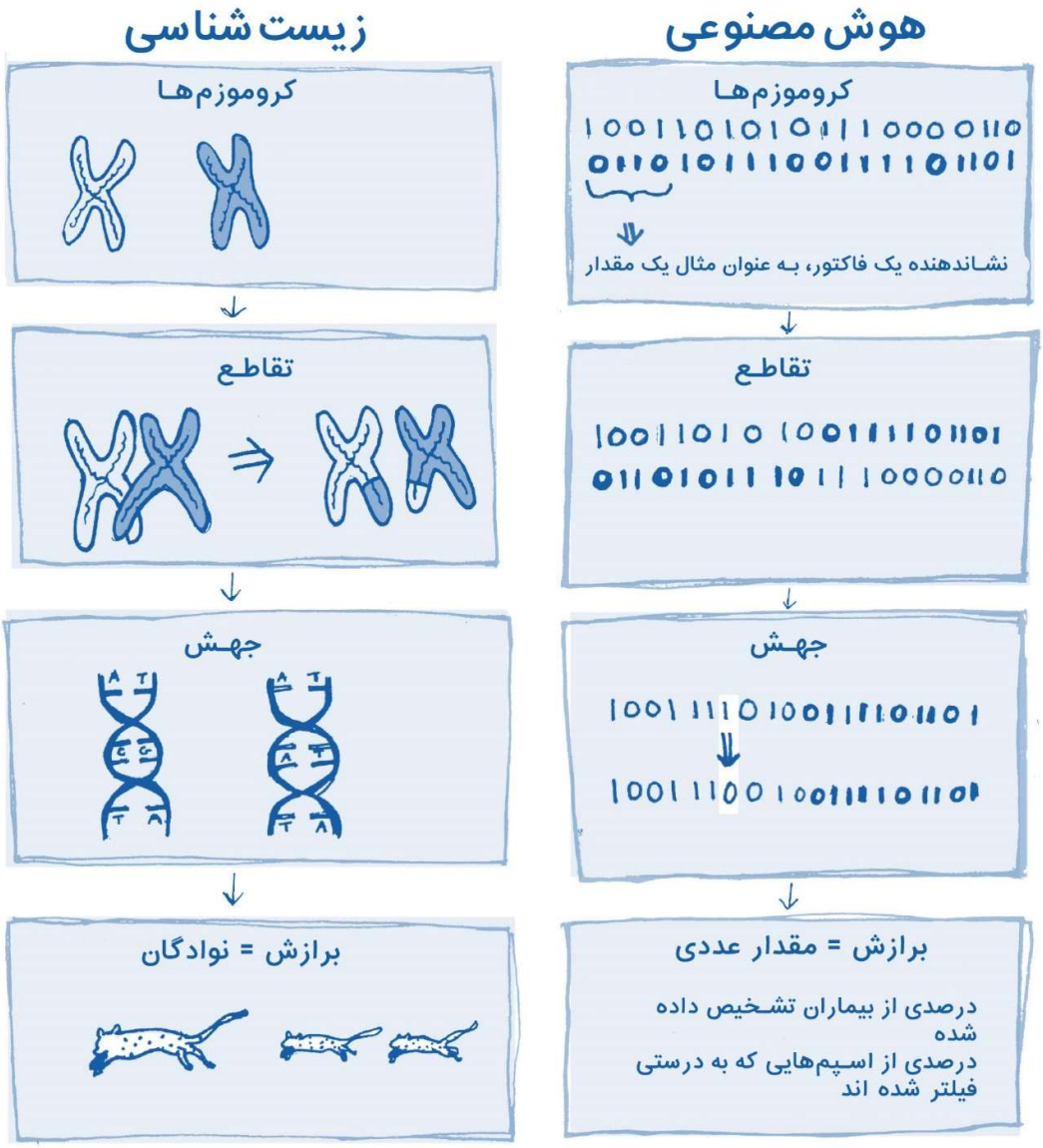
الگوریتم‌های تکاملی با استفاده از فرآیندهای انتخاب، تقاطع و جهش، سعی می‌کنند به طور مداوم «نسل‌های» جدیدی از برنامه‌ها (یا راه حل‌ها) ایجاد کنند که «تابع برازش» بالاتری داشته باشند. یعنی برنامه‌هایی که در تشخیص ایمیل‌های اسپم عملکرد بهتری دارند.

به عبارت دیگر، انسان یک معیار (همان «تابع برازش») را برای الگوریتم تعریف می‌کند و الگوریتم به طور خودکار تلاش می‌کند تا با استفاده از فرآیندهای تکاملی، به بهترین نتیجه ممکن بر اساس آن معیار دست یابد.

برای مثال در مورد فیلتر اسپم:

- نسل اول: تعدادی برنامه (یا راه حل) به صورت تصادفی ایجاد می‌شوند که هر کدام به شکلی ایمیل‌ها را بررسی می‌کنند.

- ارزیابی: هر برنامه با بررسی تعدادی ایمیل واقعی، ارزیابی می‌شود و «درصد ایمیل‌های اسپمی که به درستی فیلتر می‌شوند» برای هر کدام محاسبه می‌شود. این همان محاسبه «تابع برازش» است.
- انتخاب: برنامه‌هایی که عملکرد بهتری داشته‌اند (یعنی «تابع برازش» بالاتری داشته‌اند) برای تولید نسل بعدی انتخاب می‌شوند.
- تقاطع و جهش: بخش‌هایی از کدهای برنامه‌های برتر با هم ترکیب می‌شوند (تقاطع) و تغییرات تصادفی کوچکی نیز در کدها ایجاد می‌شود (جهش). این کار باعث ایجاد نسل جدیدی از برنامه‌ها می‌شود.
- این چرخه بارها تکرار می‌شود تا برنامه‌ای با بالاترین «تابع برازش» (یعنی بهترین عملکرد در تشخیص اسپم) به دست آید.



همانطور که الگوریتم‌های تکاملی می‌توانند راه‌حل‌های کاملاً جدیدی پیدا کنند، در زمانی هم که مدلی برای یافتن پاسخ درست وجود ندارد، عملکرد خوبی دارند. برای مثال، از آنجایی که هیچ راهی برای برنامه‌ریزی جهت راه رفتن یک ربات روی برف وجود ندارد، ربات می‌تواند با استفاده از یک الگوریتم ژنتیکی آزمایش کند و «مناسب‌ترین» روش را برای راه رفتن بدون افتادن پیدا کند. یکی از معایب الگوریتم‌های تکاملی این است که ممکن است محاسبات زیادی برای همگرایی (رسیدن به یک راه‌حل) نیاز داشته باشند که از نظر زمان و هزینه گران است. و درست مانند ربات، آن‌ها می‌توانند در تکرارهای بی‌پایان گم شوند و در ابرفضای تغییرات سرگردان شوند.

بیایید به مثال قیمت خانه خود بازگردیم و یک مزیت کلیدی الگوریتم‌های تکاملی را در نظر بگیریم یعنی کشف راه‌حل جدید از بین متغیرهای زیاد. تا کنون می‌دانیم که می‌توانیم چند متغیر را در ذهن خود نگه داریم و می‌دانیم که آمار سنتی روشی امتحان‌پس‌داده و واقعی برای پیش‌بینی قیمت بر اساس چندین متغیر بر اساس تکنیک‌هایی مانند رگرسیون است. همچنین می‌دانیم که می‌توانیم یک الگوریتم یادگیری بسازیم که بتواند تحلیل رگرسیون را به طور خودکار در پاسخ به داده‌های جدید به‌روزرسانی کند. اما اگر بخواهیم معادله‌ای برای نشان دادن روابط بسیار پیچیده‌تر، به ویژه روابطی که زمان در آن‌ها عامل است، پیدا کنیم چه؟ مدل‌سازی داده‌های سری زمانی، که در آن داده‌ها به ترتیب زمانی نمایش داده می‌شوند، دشوار است. اثرات تأخیر، همپوشانی داده‌های سری زمانی، فواصل نامنظم بین متغیرها و اثرات رقابتی، همه با مدل‌سازی خطی به سختی قابل تفکیک هستند.

الگوریتم‌های تکاملی قادر به مدیریت این نوع داده‌ها هستند و می‌توانند برای یافتن رابطه‌ای به پیچیدگی زیر استفاده شوند:

پیش‌بینی قیمت خانه ما = (قیمت‌های فروش خانه‌های اخیر در شعاع یک مایلی)^{۳} / (قیمت‌های فروش خانه‌های اخیر در شعاع ۱۰۰ مایلی)^{۳} + ۲.۵۵ × (بازاریابی انجام شده سه هفته پیش) + ۳ + × (تعداد عکس‌های رنگی در بروشور)^{۲} + ۰.۰۹۹۲ × (بازدید از وب‌سایت در دو هفته گذشته) $\log_2$(تعداد دفعات بازدید آزاد) × ۱/(۱+(باران در روز بازدید آزاد)^{۲}).

واضح است که این یک مثال خیالی با متغیرهای ساختگی است، اما الگوریتم‌های تکاملی این نوع فرمول‌های پیچیده و غیرخطی را کشف کرده‌اند. آن‌ها برای انجام همه چیز از مدیریت لجستیک در عملیات خرده‌فروشی گرفته تا بهینه‌سازی در فرآیندهای بزرگ صنعتی استفاده می‌شوند.

پیشرفت‌های اخیر در عملکرد، علاقه به الگوریتم‌های تکاملی را احیا کرده است. مشکلات همگرایی با استفاده از یک نسخه یادگیری ماشینی از «تیغ اوکام» (اصلی که بیان می‌کند، در میان فرضیه‌های رقیب، فرضیه‌ای که کمترین فرضیات را دارد باید انتخاب شود) حل شده است، یعنی راه‌حل‌ها به شدت به سمت ساده‌ترین نتیجه متمایل می‌شوند و نتایج پیچیده‌تر در اوایل محاسبات «حذف می‌شوند». کارایی با استفاده از محاسبات موازی بهبود یافته است، اکنون الگوریتم‌ها در مقیاس بسیار بزرگ استفاده می‌شوند و متغیرهایی را که به طور سنتی استفاده از آن‌ها دشوار بوده است، مانند داده‌های سری زمانی، پردازش می‌کنند. آن‌ها همچنین به طور مؤثر در رویکردهای ترکیبی استفاده می‌شوند، مثلا الگوریتم‌های تکاملی برای یافتن وزن‌های اولیه برای مقداردهی اولیه یک شبکه عصبی استفاده می‌شوند.

تیغ اوکام (Occam's Razor) که به آن «اصل یا قانون صرفه‌جویی» هم گفته می‌شود، یک اصل حل مسئله در فلسفه و علم است که به زبان ساده می‌گوید: از بین توضیحات مختلف برای یک پدیده یا مسئله، ساده‌ترین توضیح که با شواهد موجود سازگار باشد، به احتمال زیاد درست است.

به عبارت دیگر، وقتی چند فرضیه یا نظریه برای توضیح یک چیز وجود دارد، فرضیه‌ای که کمترین فرضیات اضافی را دارد و پیچیدگی کمتری دارد، ارجحیت دارد. انگار که با یک تیغ، اضافات و پیچیدگی‌های غیرضروری را از توضیحات حذف می‌کنیم.

مثال: فرض کنید صبح از خواب بیدار می‌شوید و می‌بینید که چمن جلوی خانه‌تان خیس است. دو توضیح احتمالی وجود دارد:

۱. دیشب باران آمده است.
۲. یک نفر شبانه آمده و با شلنگ چمن‌ها را آبیاری کرده و سپس ردپای خود را کاملاً پاک کرده و هیچ اثری از خود به جا نگذاشته است.

طبق تیغ اوکام، توضیح اول (باران) محتمل‌تر است، زیرا ساده‌تر است و فرضیات کمتری دارد. برای اثبات توضیح دوم، باید فرضیات زیادی را در نظر بگیریم (وجود شخصی که شبانه آمده، شلنگ داشته، ردپایش را پاک کرده و غیره) که هیچ شاهدی برای آن‌ها نداریم.

کاربردهای تیغ اوکام: تیغ اوکام در زمینه‌های مختلفی کاربرد دارد، از جمله:

- علم: در انتخاب نظریه‌های علمی، نظریه‌ای که ساده‌تر باشد و توضیحات کمتری نیاز داشته باشد، ترجیح داده می‌شود.
- هوش مصنوعی: در طراحی الگوریتم‌ها و مدل‌ها، سعی می‌شود از پیچیدگی‌های غیرضروری پرهیز شود.

- طراحی رابط کاربری (UI/UX): در طراحی رابط‌های کاربری، سعی می‌شود از پیچیدگی‌های غیرضروری و المان‌های اضافی پرهیز شود تا کاربر تجربه بهتری داشته باشد.
- تصمیم‌گیری: در زندگی روزمره، هنگام مواجهه با گزینه‌های مختلف، می‌توان از تیغ اوکام برای انتخاب گزینه ساده‌تر و منطقی‌تر استفاده کرد.

نکته مهم: تیغ اوکام به این معنی نیست که همیشه ساده‌ترین توضیح درست است. بلکه به این معنی است که در صورت برابری سایر شرایط، توضیح ساده‌تر ارجحیت دارد. گاهی اوقات، واقعیت پیچیده‌تر از آن چیزی است که به نظر می‌رسد و نیاز به توضیحات پیچیده‌تری دارد. اما به عنوان یک اصل کلی، تیغ اوکام ابزاری مفید برای ساده‌سازی مسائل و انتخاب گزینه‌های منطقی‌تر است. در متن، استفاده از «تیغ اوکام» به این معنی است که در بین راه‌حل‌های مختلفی که الگوریتم پیدا می‌کند، راه‌حل‌هایی که ساده‌تر هستند و پیچیدگی کمتری دارند (مثلاً فرمول‌های ساده‌تر با متغیرهای کمتر)، نسبت به راه‌حل‌های پیچیده‌تر ارجحیت داده می‌شوند و احتمال بیشتری برای بقا و تکثیر در نسل‌های بعدی الگوریتم دارند. این کار باعث افزایش سرعت همگرایی الگوریتم و رسیدن به راه‌حل‌های کارآمدتر می‌شود.

الگوریتم‌های بیزی

بیشتر فرایندهای تحلیلی که تا به اینجا در موردشان صحبت کردیم، از مقادیر به عنوان ورودی استفاده می‌کنند، برای مثال، رنگ یک پیکسل، قیمت یک خانه، یا دمای بدن یک بیمار. اگر ورودی به جای مقدار، یک احتمال باشد، چه؟ چگونه از ورودی‌ای مانند «۱۰٪ از افراد خوشحال، ثروتمند هستند» در یک الگوریتم استفاده کنیم؟ برای چنین مسائلی می‌توانیم از الگوریتم‌های بیزی استفاده کنیم.

ما معمولاً احتمال را بر حسب فراوانی یا گرایش در نظر می‌گیریم. برای مثال، احتمال آمدن «شیر» هنگام پرتاب یک سکه یک‌دوم است و احتمال آمدن عدد ۱ هنگام پرتاب یک تاس، یک‌ششم است. فراوانی برای موقعیت‌های کاملاً تصادفی مانند پرتاب سکه به خوبی کار می‌کند اما برای موقعیت‌هایی که نوعی الگو در آن‌ها وجود دارد، نه آن‌چنان. در این موقعیت‌ها، احتمال وقوع یک رویداد بر اساس چیزهایی است که ممکن است با آن رویداد، مرتبط باشند.

تصور کنید در جایی زندگی می‌کنید که طوفان‌ها به صورت جبهه‌ای می‌آیند. یک دقیقه روی ایوان نشسته‌اید و از منظره‌ی آسمان آبی بی‌نقص لذت می‌برید، سپس دسته‌ای از ابرهای تیره را می‌بینید. دقیقه‌ی بعد، در یک باد شدید هستید و با باران که تقریبا به صورت افقی می‌بارد، می‌جنگید و به سمت بند رخت می‌دوید تا لباس‌های شسته و پهن شده را نجات دهید. شما به این الگو عادت کرده‌اید و می‌دانید وقتی توده‌ای از ابر را در یک افق پایین می‌بینید، شانس باران تقریباً ۱۰۰٪ است. حالا تصور کنید که در تعطیلات تابستان به سان‌فرانسیسکو می‌روید. اواسط بعدازظهر با آسمان آبی روشن و دمای مطبوع ۲۴ درجه‌ی سانتی‌گراد (۷۵ درجه‌ی فارنهایت) به فرودگاه بین‌المللی سان فرانسیسکو (SFO) می‌رسید. همانطور که اوبر شما به مرکز شهر نزدیک می‌شود، به غرب نگاه می‌کنید و دسته‌ای از تیرگی خاکستری را می‌بینید که به طرز شومی برج‌های توئین‌پیکس را پوشانده است. وقتی راننده‌ی اوبر شما را در هتل پیاده می‌کند، هوا ۱۷ درجه‌ی سانتی‌گراد (۶۲ درجه‌ی فارنهایت) با بادی نسبتاً سرد که ۱۳ درجه‌ی سانتی‌گراد (۵۶ درجه‌ی فارنهایت) است!!

می‌گوئید: «به نظر می‌رسه بارون بیاد، به نظرم احتمال بارون ۹۵ درصده.»

راننده اوبر فقط شانه‌ای بالا می‌اندازه و می‌گوید: «می‌دونی چی می‌گن، سردترین زمستونی که داشتم، تابستون سان‌فرانسیسکو بود.»

هیچ بارانی نمی‌بارد، اما روز بعد همان الگو تکرار می‌شود. این بار خودتان مه معروف سان‌فرانسیسکو را تجربه می‌کنید، در حالی که با عجله از روی پل گلدن گیت عبور می‌کنید و همراه با بسیاری از گردشگران سردرگم و لرزان دیگر، باد سرد و مرطوب را تحمل می‌کنید احتمال بارش باران را ۵۰/۵۰ تخمین می‌زنید.

تا پایان تعطیلات یک هفته‌ای‌تان، به وجود مه عادت کرده‌اید و می‌دانید که تقریباً هیچ شانسی برای بارش باران وجود ندارد. هفته را با احتمال بارش ۹۵ درصد شروع کردید، آن را به ۵۰ درصد کاهش دادید و در نهایت به چیزی حدود ۱۰ درصد رسیدید.

این همان چیزی است که در بیز (Bayes) اتفاق می‌افتد. ما به احتمال «قبل» به‌عنوان *احتمال پیشین* (Prior Probability) و به احتمال «بعد» به‌عنوان *احتمال پسین* (Posterior Probability) اشاره می‌کنیم.

قضیه بیز : احتمال با توجه به شرایط مرتبط با یک رویداد توصیف می‌شود، نه بر اساس فراوانی وقوع آن رویداد.

احتمال = میزانی که یک رویداد احتمال وقوع دارد، که با نسبت موارد مطلوب به کل تعداد موارد ممکن اندازه‌گیری می‌شود.

$$\text{احتمال} = \frac{\text{موارد مطلوب}}{\text{کل تعداد موارد ممکن}}$$

۵ بیاید	$\frac{1}{6} = 16.67\%$
خط بیاید	$\frac{1}{2} = 50\%$

* این روش برای موقعیت‌های کاملاً تصادفی خوب عمل می‌کند، اما نه برای موقعیت‌هایی که الگویی در آن وجود دارد (مانند وضعیت آب وهوا).

← در تفسیر بیزین، احتمال، یک مقدار انتزاعی است که به یک فرضیه اختصاص می‌دهیم؛ به عنوان "میزان باور" ما به درستی آن فرضیه.

فرضیه : فردا باران خواهد بارید

هر چه احتمال بیشتر باشد ← بیشتر فکر می‌کنید که به چتر نیاز دارید.

تنظیم مقدار بر اساس شواهدی که درباره وضعیت به دست می‌آورید:

مقدار اولیه = احتمال پیشین (Prior Probability) و استفاده از شواهد برای رسیدن به

مقدار جدید برای احتمال = احتمال پسین (Posterior Probability) ← که امیدواریم به حقیقت نزدیک‌تر از احتمال اولیه باشد

آمار بیزی احتمال باور (در یک دانش خاص) را به‌عنوان نقطه شروع در نظر می‌گیرد. این احتمال پیشین (prior probability) نشان‌دهنده این است که چقدر یک مدل پیش از مشاهده هرگونه داده واقعی معتبر است. پس از جمع‌آوری داده‌ها، توزیع احتمالی دیگری به نام احتمال پسین (posterior probability) ایجاد می‌شود. بنابراین، ما با یک مقدار اولیه (احتمال پیشین) شروع می‌کنیم و سپس با توجه به داده‌های تولیدشده، این مقدار را تنظیم می‌کنیم (احتمال پسین). با پیشرفت تکرارها، فرضیه به‌طور تدریجی به‌روز می‌شود و عدم قطعیت کاهش می‌یابد.

پیش از آن‌که به الگوریتم‌های بیزی بپردازیم، باید با قضیه بیز (Bayes' Theorem) آشنا شویم، که روشی برای توصیف احتمال وقوع یک رخداد بر اساس عواملی است که ممکن است با آن رخداد مرتبط باشند. تفکر بیزی لزوماً به‌صورت شهودی قابل درک نیست و بهترین روش برای توضیح آن استفاده از یک مثال است.

دوست شما، مکس، نتیجه آزمایش مثبت سرطان دریافت کرده و می‌خواهد بداند احتمال واقعی ابتلا به سرطان چقدر است. او می‌داند که آزمایش، خود رخداد نیست؛ یعنی داشتن نتیجه مثبت آزمایش، معادل با داشتن سرطان نیست. همچنین، او متوجه است که آزمایش‌ها کامل و بی‌نقص نیستند و ممکن است سرطان را در زمانی که وجود ندارد تشخیص دهند (نتیجه مثبت کاذب) یا نتوانند سرطان را در زمانی که وجود دارد شناسایی کنند (نتیجه منفی کاذب). پس از دریافت نتیجه مثبت آزمایش، احتمال ابتلا به سرطان تنها به دقت آزمایش وابسته نیست، بلکه به احتمال یک نتیجه مثبت واقعی تقسیم بر احتمال هرگونه نتیجه مثبت بستگی دارد. احتمال اینکه مکس به سرطان مبتلا باشد چقدر است، با توجه به اینکه (۱) ۱٪ از مردم سرطان دارند، (۲) ۸۰٪ از آزمایش‌ها، سرطان را در زمانی که وجود دارد تشخیص می‌دهند، (۳) و ۹.۶٪ از آزمایش‌ها، سرطان را در زمانی که وجود ندارد تشخیص می‌دهند

	* ابتلاء	* عدم ابتلاء
	CANCER 1٪	NO CANCER 99٪
TEST +	80٪	9.6٪
TEST −	20٪	90.4٪

۱٪ مردم به سرطان مبتلاء هستند

* اگر سرطان داشته باشد: در ستون اول جدول قرار می‌گیرد. *

۸۰٪ احتمال دارد نتیجه آزمایش مثبت باشد.
۲۰٪ احتمال دارد نتیجه آزمایش منفی باشد.

* اگر سرطان نداشته باشد: در ستون‌دوم جدول قرار می‌گیرد. *

۹.۶٪ احتمال دارد نتیجه آزمایش مثبت باشد
۹۰.۴٪ احتمال دارد نتیجه آزمایش منفی باشد

چقدر آزمایش دقیق است؟
احتمال اینکه سرطان داشته باشد چقدر است؟

آزمایش مثبت: شما در ردیف بالای جدول قرار دارید،

ممکن است یک مثبت واقعی باشد.
ممکن است یک مثبت کاذب باشد.

احتمال مثبت واقعی: = احتمال اینکه سرطان داشته باشید × احتمال اینکه آزمایش آن را تشخیص دهد = 1٪ × 80٪ = 0.008

احتمال مثبت کاذب: = احتمال اینکه سرطان نداشته باشید × احتمال اینکه آزمایش همچنان مثبت نشان دهد = 99٪ × 9.6٪ = 0.09509

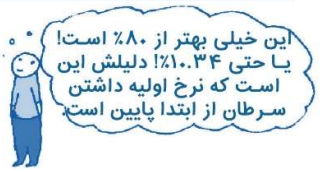
	CANCER 1٪	NOT CANCER 99٪
TEST +	مثبت واقعی 1٪ × 80٪ = 0.008	مثبت کاذب 99٪ × 9.6٪ = 0.09509
TEST −	منفی کاذب 1٪ × 20٪ = 0.00002	منفی واقعی 99٪ × 90.4٪ = 0.89601

احتمـال دریافت هر نتیجـه مثبت (واقعی یا کاذب)=

0.008 + 0.09509 = .10304

احتمال داشتن سرطان = احتمـال نتیجه مثبت واقعی / احتمال هـر نتیجه مثبت

= 0.008 / .10304 = 0.0776 ≈ 7.8٪



جالب است! اگر مکس فقط دقت آزمایش را در نظر می‌گرفت، فرض می‌کرد که احتمال ابتلای او به سرطان ۸۰٪ است، در حالی که در واقع، احتمال ابتلای او تنها ۷.۸٪ است. نرخ پایه ابتلا به سرطان در ابتدا بسیار پایین بود و نرخ نتایج مثبت کاذب آن‌قدر بالا است که در هر جمعیتی، تعداد زیادی نتایج مثبت کاذب وجود خواهد داشت.

قضیه بیز بیان ریاضی این ایده است :

احتمال مدل با توجه به داده‌ها برابر است با احتمال داده‌ها با توجه به مدل، ضرب در احتمال پیشین مدل، تقسیم بر احتمال داده‌ها.

بنابراین، اگر بخواهیم مثال سرطان را به‌صورت یک فرمول ریاضی بررسی کنیم:

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

که در آن A و B دو رویداد هستند و P(B) مخالف صفر است

P(A) و P(B):
احتمال مشاهده رویدادهای A و B بدون توجه به یکدیگر هستند.

P(A|B):
یک احتمال شرطی است و احتمال وقوع رویداد A را به شرط اینکه B درست باشد، نشان می‌دهد.

P(B|A):
احتمال مشاهده رویداد B به شرط اینکه A درست باشد.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

$$= \frac{80\% \times 1\%}{10.34\%} = \frac{0.008}{.10304} = 0.0776 \approx 7.8\%$$

یا به طور کلی تر...

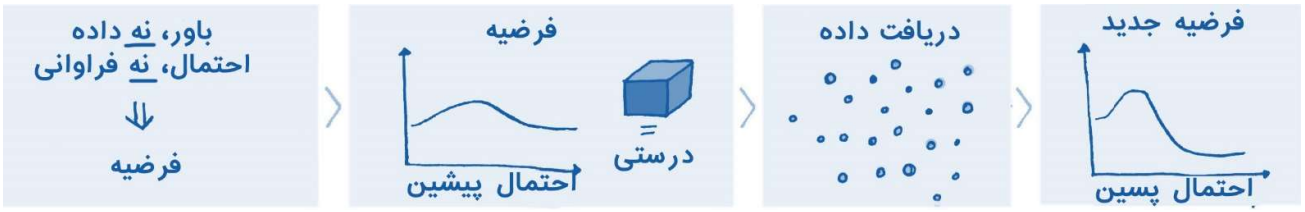
احتمال پسین $P(A|B)$: چه مقدار احتمال دارد فرضیه ما با توجه به داده‌ها درست باشد؟

احتمال $P(B|A)$: چقدر احتمال دارد داده‌ها با فرض درست بودن فرضیه‌ی ما مشاهده شوند؟

احتمال پیشین $P(A)$: قبل از بررسی داده‌ها، چقدر فرضیه محتمل بوده است؟

احتمال حاشیه‌ای $P(B)$: احتمال داده‌های جدید تحت تمام فرضیه‌های ممکن چقدر است؟

احتمال فرضیه‌ی ما با توجه به داده‌ها =
(احتمال داده‌ها با فرض درست بودن فرضیه‌ی ما × احتمال اولیه‌ی فرضیه‌ی ما) / احتمال کل داده‌ها



پنل اول (از سمت چپ) - نقطه شروع - باور اولیه: این پنل نشان می‌دهد که در روش بیزی، ما با یک «باور» یا «حدس اولیه» شروع می‌کنیم، نه صرفاً با داده‌ها. این باور اولیه را «فرضیه» می‌نامیم. نکته‌ی مهم این است که این روش بر مبنای «احتمال» است، نه «فراوانی». یعنی به جای اینکه ببینیم یک اتفاق چند بار رخ داده، به این فکر می‌کنیم که چقدر به درستی آن باور داریم. مثلا فرض کنید می‌خواهیم بفهمیم که آیا فردا باران می‌بارد یا نه. باور اولیه‌ی ما (فرضیه) می‌تواند این باشد که «فردا احتمالاً باران نمی‌بارد».

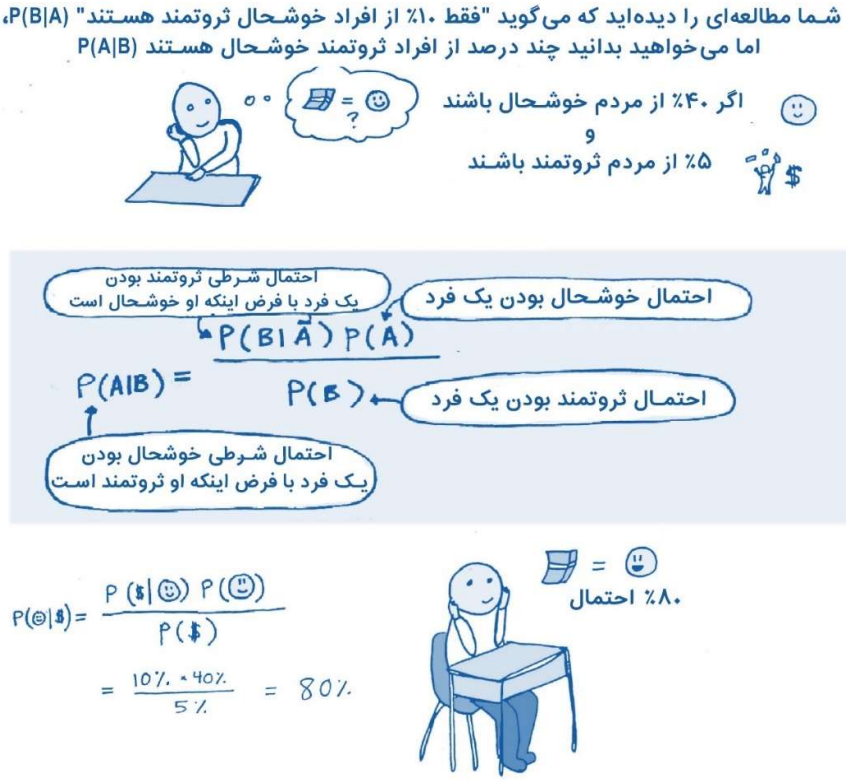
پنل دوم - نمایش باور اولیه - احتمال پیشین: این پنل این باور اولیه (فرضیه) را به صورت گرافیکی نشان می‌دهد. منحنی نشان‌دهنده‌ی «احتمال پیشین» است. این احتمال نشان می‌دهد که قبل از دیدن هرگونه مدرک یا داده‌ی جدید، چقدر به فرضیه‌ی خود باور داشتیم. در مثال باران، اگر معمولاً در این فصل باران نبارد، احتمال پیشین ما برای «نباریدن باران» بالا خواهد بود و منحنی به سمت راست متمایل می‌شود.

پنل سوم - دریافت داده و ارزیابی آن: در این مرحله، «داده» یا «مدرک» جدیدی به دست می‌آوریم. سپس این داده را ارزیابی می‌کنیم تا ببینیم چقدر با باور اولیه‌ی ما همخوانی دارد یا آن را رد می‌کند. «اعتبار» یا «درستی» نشان می‌دهد که این داده چقدر برای به‌روزرسانی باور ما مهم است. فرض کنید آسمان پر از ابرهای تیره شده است. این «داده‌ی جدید» است. «اعتبار» این داده بالاست، زیرا ابرهای تیره معمولاً نشانه‌ی باران هستند.

پنل چهارم - به‌روزرسانی باور - احتمال پسین: در نهایت، با توجه به داده‌ی جدید و ارزیابی آن، باور اولیه‌ی خود را به‌روز می‌کنیم. این باور به‌روز شده را «احتمال پسین» می‌نامیم. منحنی جدید نشان می‌دهد که بعد از دیدن داده‌ی جدید، چقدر به فرضیه‌ی خود باور داریم. مثلا با دیدن ابرهای تیره، باور خود را از «احتمالاً باران نمی‌بارد» به «احتمالاً باران می‌بارد» تغییر می‌دهیم. احتمال پسین برای «باریدن باران» بالاتر می‌رود و منحنی تغییر شکل می‌دهد.

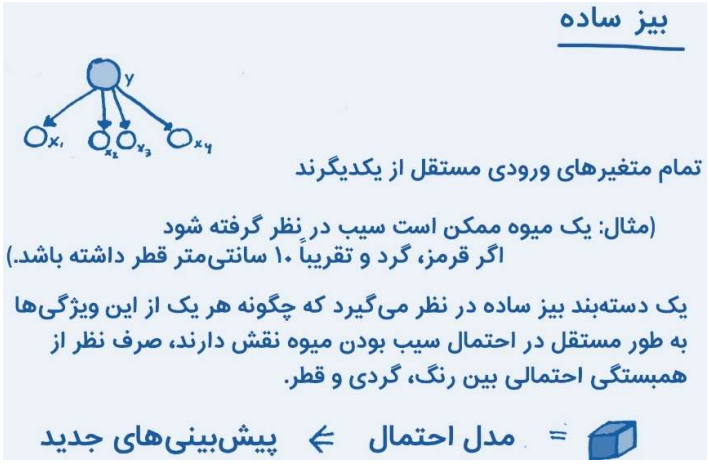
به طور خلاصه، استنتاج بیزی روشی برای به‌روزرسانی باورهایمان با استفاده از داده‌های جدید است. ما با یک باور اولیه (احتمال پیشین) شروع می‌کنیم، داده‌های جدید را جمع‌آوری می‌کنیم و سپس باور خود را بر اساس این داده‌ها به‌روز می‌کنیم و به یک باور جدید (احتمال پسین) می‌رسیم.

یک مثال دیگر...



تصویر می‌خواهد رابطه‌ی بین خوشحالی و ثروت را با استفاده از احتمال شرطی و قضیه‌ی بیز نشان دهد.
ابتدا گفته می‌شود که می‌دانیم «اگر کسی خوشحال باشد، فقط ۱۰٪ احتمال دارد که ثروتمند باشد. اما سوال اینجاست: «اگر کسی ثروتمند باشد، چقدر احتمال دارد که خوشحال باشد؟
برای پاسخ به این سوال، اطلاعات دیگری هم داریم: ۴۰٪ از مردم خوشحال هستند و ۵٪ از مردم ثروتمند هستند.
با استفاده از فرمول بیز، می‌توانیم این احتمال را محاسبه کنیم:
احتمال (خوشحال بودن به‌شرط ثروتمند بودن) = (احتمال (ثروتمند بودن به‌شرط خوشحال بودن) x احتمال (خوشحال بودن)) / احتمال (ثروتمند بودن)
با جایگذاری اعداد:
احتمال (خوشحال بودن به‌شرط ثروتمند بودن) = (۱۰٪ x ۴۰٪) / ۵٪ = ۸۰٪
یعنی اگر کسی ثروتمند باشد، ۸۰٪ احتمال دارد که خوشحال باشد.

از تئوری تا عمل... چگونه این ایده به الگوریتم‌های یادگیری تبدیل می‌شود؟
الگوریتم‌های بیزی از استنتاج احتمالی، یک روش آماری برای تخمین پارامترهای یک توزیع زیربنایی یا پنهان بر اساس یک توزیع پیشین، استفاده می‌کنند. بنابراین، در حالی که در یک شبکه‌ی عصبی وزن‌هایی را به ورودی‌ها اعمال می‌کنیم، در یک الگوریتم بیزی، به جای آن احتمالات را اختصاص می‌دهیم. با بازگشت به مثال مخلوط کردن رنگ‌های قرمز و آبی، یک شبکه‌ی عصبی وزن‌هایی را به هر رنگ اعمال می‌کند و رنگی از طیف بنفش ایجاد می‌کند، در حالی که یک الگوریتم بیزی احتمال قرمز یا آبی بودن را اعمال می‌کند. «یادگیری» فرآیند به‌روزرسانی مداوم احتمالات ورودی‌ها بر اساس خروجی جدید است که به نوبه‌ی خود احتمال خروجی را به‌روز می‌کند.
یکی از کاربردها در یادگیری ماشین، الگوریتم‌های بیز ساده است...
الگوریتم‌های بیز ساده این ایده را می‌گیرند و آن را در موقعیت‌هایی به کار می‌برند که چندین کلاس وجود دارد و ما به یک روش سریع و آسان برای ساخت یک مدل پیش‌بینی بر اساس احتمال منفرد این کلاس‌ها نیاز داریم. به آن «ساده» گفته می‌شود زیرا مدل این فرض را می‌کند که یک ویژگی خاص در یک کلاس با هیچ ویژگی دیگری مرتبط نیست. برای مثال، اگر یک میوه قرمز، گرد و به اندازه‌ی یک توپ بیسبال باشد، ممکن است سیب در نظر گرفته شود. حتی اگر این ویژگی‌ها به یکدیگر یا ویژگی‌های دیگر وابسته باشند، همه‌ی آنها به طور مستقل در احتمال اینکه این میوه سیب باشد، سهیم هستند.



۱ .تفاوت الگوریتم‌های بیزی با شبکه‌های عصبی:

- شبکه‌های عصبی: مثل این است که شما به هر ورودی (مثلاً رنگ قرمز و آبی) یک «وزن» بدهید. این وزن‌ها نشان می‌دهند که هر ورودی چقدر در خروجی (مثلاً رنگ بنفش) تأثیر دارد. با تغییر این وزن‌ها، خروجی هم تغییر می‌کند.
- الگوریتم‌های بیزی: به جای وزن، به هر ورودی یک «احتمال» اختصاص می‌دهند. یعنی احتمال اینکه ورودی قرمز باشد، چقدر و احتمال اینکه آبی باشد چقدر است. این احتمالات نشان می‌دهند که چقدر به وجود هر ورودی باور داریم.

مثال رنگ‌ها: فرض کنید می‌خواهیم رنگ بنفش را با ترکیب رنگ قرمز و آبی بسازیم.

- شبکه‌ی عصبی: به رنگ قرمز وزن ۰.۶ و به رنگ آبی وزن ۰.۴ می‌دهد. این یعنی رنگ قرمز تأثیر بیشتری دارد و بنفش به قرمزی می‌زند.
- الگوریتم بیزی: احتمال می‌دهد که ۶۰٪ اوقات رنگ قرمز و ۴۰٪ اوقات رنگ آبی استفاده شده است.

۲ .مفهوم «یادگیری» در الگوریتم‌های بیزی:

«یادگیری» در این الگوریتم‌ها یعنی اینکه با دیدن خروجی‌های جدید، احتمالات مربوط به ورودی‌ها را به‌روز کنیم. مثلاً اگر ببینیم خروجی خیلی بنفش نیست و بیشتر به آبی می‌زند، احتمال آبی بودن ورودی را بیشتر می‌کنیم. این به‌روزرسانی احتمالات ورودی‌ها، به نوبه‌ی خود احتمال خروجی را هم تغییر می‌دهد.

۳ .الگوریتم بیز ساده (Naïve Bayes):

این الگوریتم یک نوع خاص از الگوریتم‌های بیزی هست که برای دسته‌بندی داده‌ها به چند گروه مختلف استفاده می‌شود. «ساده» است چون یک فرض ساده‌انگارانه دارد: فرض می‌شود که ویژگی‌های مختلف یک داده هیچ ربطی به هم ندارند.

مثال: فرض کنید می‌خواهیم تشخیص بدهیم که یک میوه سیب هست یا نه. ویژگی‌های این میوه می‌تواند رنگ قرمز، گرد بودن و اندازه‌ی توپ بیسبال باشد. الگوریتم بیز ساده فرض می‌کند که این ویژگی‌ها کاملاً مستقل از هم هستند. یعنی اینکه میوه قرمز باشه، هیچ ربطی به گرد بودنش ندارد و هر دو به طور جداگانه در احتمال سیب بودن میوه تأثیر دارند.

چرا این فرض ساده‌انگارانه مفید است؟

این فرض باعث می‌شود تا محاسبات خیلی ساده‌تر شده و الگوریتم بتواند خیلی سریع یک مدل پیش‌بینی بسازد. البته این فرض همیشه درست نیست، اما در بسیاری از موارد به خوبی کار می‌کند.

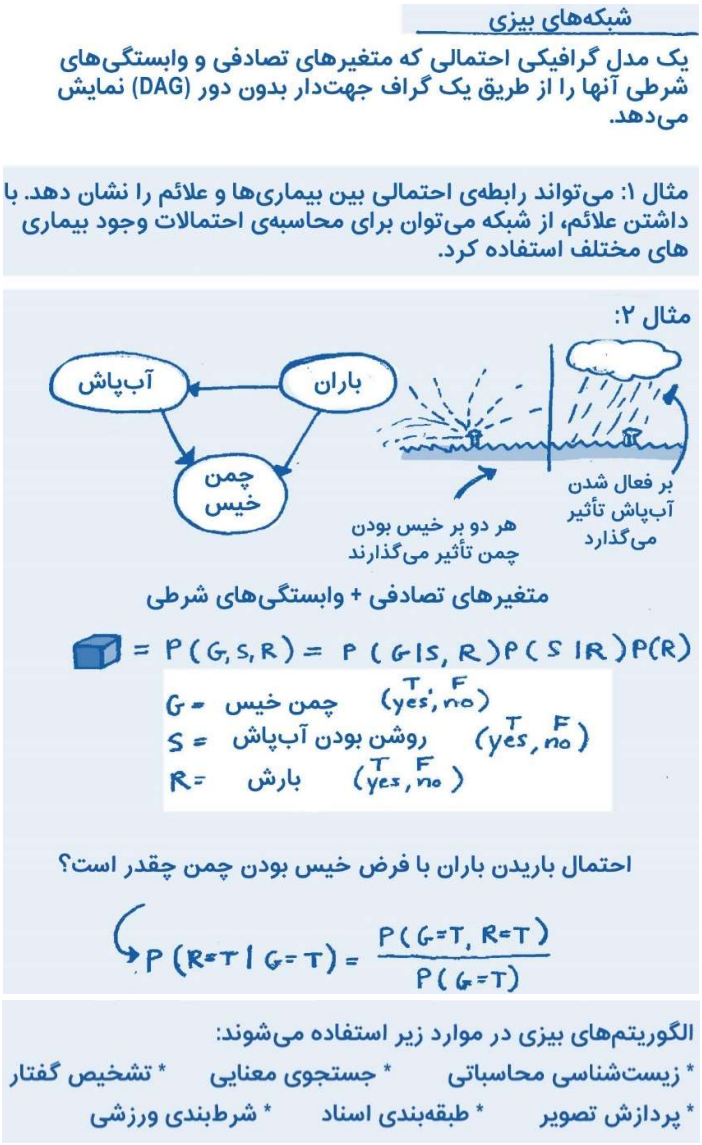
الگوریتم‌های بیز ساده برای هر کلاس در مجموعه داده، احتمال پسین را محاسبه می‌کنند. کلاسی با بالاترین احتمال پسین به عنوان خروجی پیش‌بینی در نظر گرفته می‌شود. برای مثال، اگر می‌خواهید با توجه به بارندگی، پیش‌بینی کنید که آیا بازی فوتبال دخترتان لغو می‌شود یا خیر، رویکرد بیز ساده شامل محاسبه‌ی احتمالات ترکیبات منفرد، مانند احتمال ابری بودن، بارانی بودن یا آفتابی بودن است و متعاقبا محاسبه‌ی احتمالات پسین منفرد برای احتمال لغو به دلیل بارندگی.

به دلیل فرض استقلال بین ویژگی‌ها، از بیز ساده به طور گسترده در طبقه‌بندی متن و فیلتر کردن ایمیل‌های ناخواسته استفاده می‌شود یعنی جایی که به خوبی عمل می‌کند. همچنین در تحلیل احساسات رسانه اجتماعی محبوب است، زیرا می‌تواند به طور کارآمد احساسات مثبت و منفی مشتری را شناسایی کند.

روش دیگری برای استفاده از قضیه‌ی بیز در شبکه‌های بیزی است...

شبکه‌های بیزی ترکیبی از نظریه گراف (ساختارهای ریاضی برای مدل‌سازی روابط دوتایی بین اشیاء) و نظریه احتمال است. آنها زمانی به خوبی عمل می‌کنند که عدم قطعیت و پیچیدگی به طور همزمان اتفاق بیفتند. در این مشارکت، نظریه گراف مدولار بودن را به ارمغان می‌آورد، جایی که یک سیستم پیچیده می‌تواند با ترکیب اجزای ساده‌تر ساخته شود و نظریه احتمال به عنوان چسب عمل می‌کند و اطمینان حاصل می‌کند که سیستم در کل سازگار است. همچنین احتمال رابط کاربری را دیکته می‌کند و راه‌هایی را برای رسیدن داده‌ها به مدل‌ها فراهم می‌کند. آنها برای بسیاری از کاربردها، به ویژه در مواردی که نیاز به مدل‌سازی مجموعه‌های متغیر با تعامل بالا وجود دارد، استفاده می‌شوند.

رایج‌ترین مشکلی که می‌خواهیم با استفاده از شبکه‌های بیزی حل کنیم، استنتاج احتمالی است. برای مثال، شبکه‌ای از آب‌پاش‌های چمن را در نظر بگیرید و فرض کنید می‌بینیم که چمن خیس است. برای این موضوع دو دلیل احتمالی وجود دارد: یا باران می‌بارد یا آب‌پاش روشن است. می‌توانیم از قانون بیز برای محاسبه‌ی احتمال پسین هر توضیح و پیش‌بینی اینکه آیا خیس بودن چمن به دلیل آب‌پاش یا باران محتمل‌تر است، استفاده کنیم.

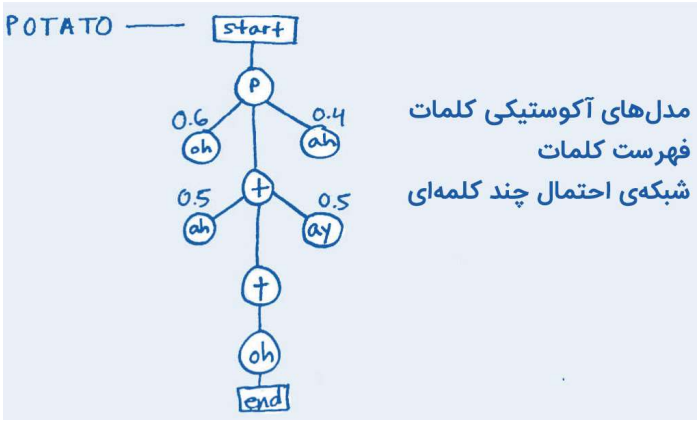


چگونه ماشین‌ها صحبت می‌کنند؟

داشتن ماشین‌هایی که بتوانند به طور هوشمندانه با انسان‌ها صحبت کنند، نشان‌دهنده‌ی یک پیشرفت بزرگ در رابطه ما با فناوری است. اما یک گفتگوی خوب به همان اندازه که به توانایی شنیدن، دیدن و ترجمه‌ی کلمات نیاز دارد، به درک زمینه و منظور نیز نیازمند است.

دستیاران صوتی به سرعت در حال رشد هستند و اکنون بیش از ۱۵٪ از کل ترافیک جستجو را تشکیل می‌دهند. با افزایش دقت آنها، هوش مصنوعی مکالمه‌ای، مانند چت‌بات‌ها، در همه جا حاضر می‌شود.

گفتار انسان پیچیده است، با شاخه‌ها، پیچیدگی‌ها و چرخش‌های بسیار، بنابراین مجموعه‌ی داده‌ی آموزشی برای زبان طبیعی بسیار بزرگ است. انتخاب، گردآوری و آماده‌سازی داده‌های آموزشی در پردازش زبان طبیعی تا حدودی یک هنر باقی مانده است و جزئیات می‌توانند تفاوت بین یک سیستم با عملکرد خوب و یک سیستم با عملکرد ضعیف را ایجاد کنند. و در زبان، دقت مهم است. دقت بیش از ۹۵٪ اساساً تجربه‌ی کاربر را تغییر می‌دهد و نقطه‌ای است که در آن هوش مصنوعی مکالمه‌ای به رابط ترجیحی برای بسیاری از تعاملات تبدیل می‌شود.



این تصویر تشخیص گفتار را نشان می‌دهد. سیستم با صدای "پ (P) "شروع می‌کند. سپس، با توجه به مدل‌های آکوستیکی و احتمالات، تصمیم می‌گیرد که صدای بعدی چیست. احتمال دارد صدای "اُ (oh) "با احتمال ۰.۶ یا صدای "آ (ah) "با احتمال ۰.۴ بیاید.

اگر صدای "آ (ah) "انتخاب شود، دوباره یک انتخاب وجود دارد: صدای "آ (ah) "با احتمال ۰.۵ یا صدای "اِی (ay) "با احتمال ۰.۵. در نهایت، به صدای "اُ (oh) "می‌رسیم و کلمه کامل می‌شود.

در سمت راست تصویر، سه منبع اطلاعاتی که سیستم برای تشخیص کلمات استفاده می‌کند، ذکر شده است:

۱. **مدل‌های آکوستیکی کلمات :**این مدل‌ها صداهای مختلف زبان را مدل‌سازی می‌کنند.
۲. **فهرست کلمات/لیست کلمات :**این فهرست شامل کلماتی است که سیستم قادر به تشخیص آنهاست.
۳. **شبکه‌ی احتمال چند کلمه‌ای :**این شبکه احتمال وقوع چند کلمه پشت سر هم را نشان می‌دهد.

یکی از راه‌هایی که ماشین‌ها می‌توانند گفتار را یاد بگیرند، با استفاده از حالت خاصی از شبکه‌های بیزی به نام مدل پنهان مارکوف است. مدل‌های پنهان مارکوف می‌توانند داده‌های سری زمانی را مدل‌سازی کنند و توزیع‌های احتمال را بر روی دنباله‌های مشاهدات نشان دهند. این فرض که یک رویداد می‌تواند باعث رویداد دیگری در آینده شود، اما نه برعکس، به مدل اجازه می‌دهد تا در طول زمان «به جلو حرکت کند».

در شکل بعدی، یک حالت (x) وجود دارد که با زمان تغییر می‌کند و ما می‌خواهیم آن را تخمین بزنیم یا ردیابی کنیم، که در این مورد کلمات در یک جمله هستند. کلمات، پنهان هستند بنابراین نمی‌توانیم مستقیماً آنها را مشاهده کنیم. در عوض، می‌خواهیم آنها را پیش‌بینی کنیم. چیزی که می‌توانیم مشاهده کنیم چیزی مرتبط با کلمات واقعی است: حالت(y) ، که مثلاً ساختار دستوری را نشان می‌دهد. هر حالت یک توزیع احتمال را بر روی خروجی اعمال می‌کند. ممکن است کلمات (پنهان) را ندانیم، اما می‌دانیم که آنها با نحوه‌ی قرارگیری کلمات در ساختار یک جمله مرتبط هستند.

Hidden Markov Model یا HMM

مدل پنهان مارکوف Hidden Markov Model) یا (HMM یک مدل آماری است که برای مدل‌سازی سیستم‌هایی به کار می‌رود که در آن‌ها یک سری حالت‌های پنهان وجود دارند که مستقیماً قابل مشاهده نیستند، اما می‌توان مشاهداتی را از آن‌ها به دست آورد. این مدل در زمینه‌های مختلفی از جمله تشخیص گفتار، تشخیص دست‌خط، و بیوانفورماتیک کاربرد دارد.

تشریح مفهوم مدل پنهان مارکوف با یک مثال:

تصور کنید فردی پشت پرده‌ای ایستاده است و تنها سایه او قابل مشاهده است. شما نمی‌توانید خود فرد را ببینید (حالت پنهان)، اما می‌توانید حرکات و سایه او را مشاهده کنید (مشاهدات). مدل پنهان مارکوف نیز بر همین اساس عمل می‌کند:

- حالت‌های پنهان: این حالت‌ها مانند خود شخص پشت پرده هستند که مستقیماً قابل مشاهده نیستند. در زمینه تشخیص گفتار، این حالت‌ها می‌توانند کلمات یا واحدهای آوایی تشکیل‌دهنده کلمات باشند.
- مشاهدات: این مشاهدات مانند سایه شخص هستند که قابل مشاهده‌اند. در تشخیص گفتار، این مشاهدات امواج صوتی هستند که توسط میکروفون ضبط می‌شوند.

هدف مدل پنهان مارکوف، با توجه به مشاهدات (امواج صوتی)، حدس زدن حالت‌های پنهان (کلمات) است.

نحوه عملکرد مدل پنهان مارکوف:

این مدل بر اساس دو مفهوم کلیدی عمل می‌کند:

۱. احتمال انتقال (Transition Probability): احتمال انتقال از یک حالت پنهان به حالت پنهان دیگر را نشان می‌دهد. به عنوان مثال، در کلمه «سیب»، احتمال اینکه پس از صدای «س» صدای «ی» بیاید چقدر است؟ این احتمال نشان‌دهنده چگونگی توالی حالت‌های پنهان است.

۲. احتمال انتشار (Emission Probability): احتمال تولید یک مشاهده خاص توسط یک حالت پنهان را نشان می‌دهد. به عنوان مثال، احتمال اینکه کلمه «سیب» باعث ایجاد یک الگوی صوتی خاص شود چقدر است؟ این احتمال رابطه بین حالت‌های پنهان و مشاهدات را مشخص می‌کند.

مدل پنهان مارکوف با استفاده از این احتمالات، به دنبال بهترین توالی حالت‌های پنهان (کلمات) می‌گردد که با مشاهدات (امواج صوتی) بیشترین تطابق را داشته باشد.

مثال تشخیص یک جمله: برای درک بهتر، تشخیص جمله «من سیب می‌خورم» را در نظر بگیرید:

حالت‌های پنهان: کلمات «من»، «سیب» و «می‌خورم» هستند.

مشاهدات: امواج صوتی ضبط شده از این جمله هستند.

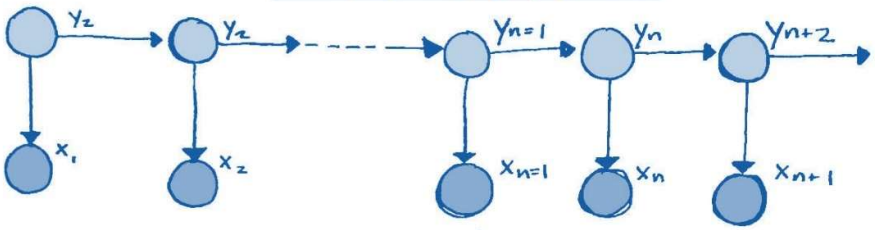
مدل پنهان مارکوف با بررسی احتمالات انتقال (مانند احتمال اینکه پس از «من» کلمه «سیب» بیاید) و احتمالات انتشار (مانند اینکه کلمه «سیب» چه صدایی تولید می‌کند)، تلاش می‌کند تا بهترین حدس را در مورد کلمات جمله بزند.

دلایل نامگذاری:

- «پنهان»: به این دلیل از واژه «پنهان» استفاده می‌شود که حالت‌های واقعی (کلمات) مستقیماً از مشاهدات (امواج صوتی) قابل تشخیص نیستند و باید توسط مدل حدس زده شوند.
- «مارکوف»: این نامگذاری به دلیل مبنای ریاضی و نظری این مدل است که بر اساس کارهای ریاضیدان روسی، آندری مارکوف (Andrey Markov)، بنا شده است. مفهوم «زنجیره‌های مارکوف» که توسط وی مطرح شد، بیان می‌کند که وضعیت فعلی یک سیستم فقط به وضعیت قبلی آن بستگی دارد و مستقل از گذشته‌های دورتر است. این خاصیت در مدل پنهان مارکوف نیز به کار می‌رود و فرض می‌شود که هر حالت پنهان فقط به حالت پنهان قبلی خود وابسته است، که این امر محاسبات را ساده‌تر می‌کند.

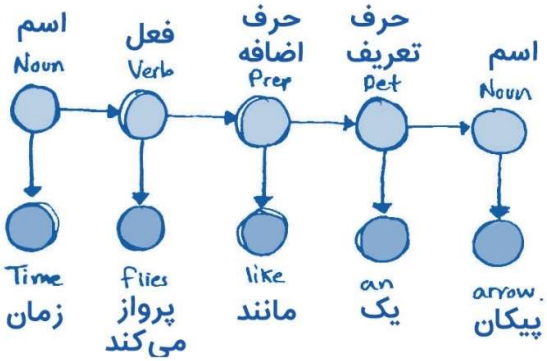
در نتیجه، مدل پنهان مارکوف با استفاده از احتمالات انتقال و انتشار و با تکیه بر خاصیت مارکوف، قادر به حدس زدن حالت‌های پنهان بر اساس مشاهدات است و در کاربردهای مختلفی مانند تشخیص گفتار نقش مهمی ایفا می‌کند.

مدل پنهان مارکوف

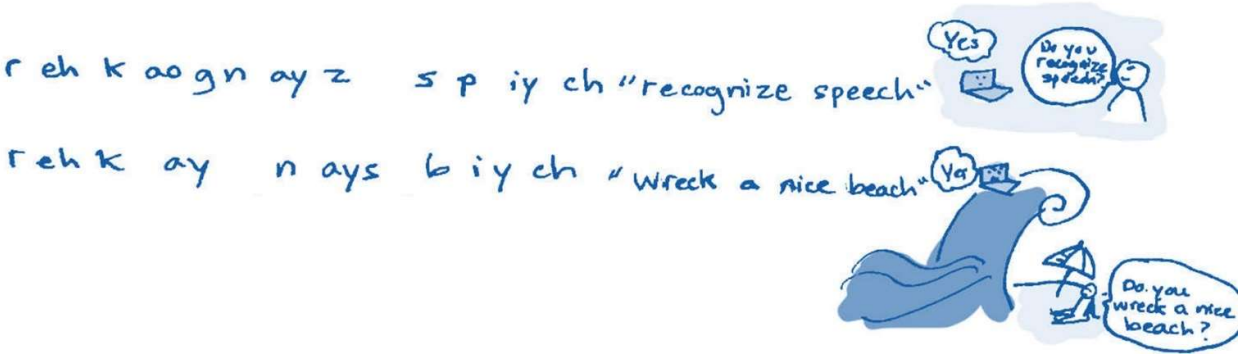


Y نشان‌دهنده‌ی حالت پنهان (مثلاً برچسب دستوری) در زمان‌های مختلف است.
X نشانگر مشاهدات (مثلاً کلمات) هستند

هدف: با داشتن یک جمله، برچسب دستوری هر کلمه را تعیین کنید



سیستم باید تشخیص دهد که کلمه از کجا شروع و در کجا تمام می‌شود



با قیاس دو عبارت "recognize speech" (به معنای «صحبت را تشخیص بده» با آوای «reh kaogn ay z sp iy ch») و "Wreck a nice beach"(به معنای «ساحل زیبایی را خراب کن» با آوای «reh k ay n ays b iy ch») نشان می‌دهند که چگونه سیستم باید مرز بین کلمات را هم تشخیص دهد، زیرا صداهای مشابه می‌توانند به کلمات مختلفی تبدیل شوند.

یک دستیار هوشمند می‌تواند کلمه‌ی بعدی در جمله (عنصر پنهان) را بر اساس ساختار شبکه و احتمالات پیشین عناصر غیرپنهان پیش‌بینی کند. سیستم‌های تشخیص گفتار نحوه‌ی یافتن کلمات پنهان، ساختن آنها به صورت عبارات و تبدیل عبارات به جملات را پیدا می‌کنند. آنها به شدت به آموزش بر روی واژگان مرتبط و به هر چقدر داده که بتوانند در مورد اینکه کدام کلمه به احتمال زیاد بعد از کلمه‌ی دیگر می‌آید، تکیه می‌کنند. با افزایش داده‌ها، سیستم احتمالات جدید را محاسبه می‌کند و دقت آن بهبود می‌یابد. هر چه بیشتر با Siri صحبت کنید، او بهتر خواهد شد.

فرض کنید می‌خواهید بگویید "هوا امروز..." دستیار هوشمند با توجه به کلمه‌ی "امروز" و با استفاده از اطلاعاتی که قبلاً یاد گرفته، حدس می‌زند که احتمالاً کلمه‌ی بعدی "خوب"، "بارانی" یا کلمات مشابه باشد. هرچه داده‌های بیشتری به سیستم داده شود (یعنی افراد بیشتری با آن صحبت کنند)، سیستم بهتر یاد می‌گیرد که کدام کلمات معمولاً پشت سر هم می‌آیند و در نتیجه دقت آن در پیش‌بینی کلمات بعدی و فهمیدن منظور ما بیشتر می‌شود. به همین دلیل گفته می‌شود که «با Siri صحبت کنید، او بهتر خواهد شد». یعنی هرچه بیشتر از آن استفاده شود، عملکردش بهبود پیدا می‌کند.

مقایسه‌گران

"شمشیر برای جنگجو مانند قلم است برای...؟"

"حرکت برای سبز مانند ایست است برای...؟"

"میو برای گربه مانند پارس است برای...؟"

این‌ها تشبیهات ساده‌ای هستند. پر کردن جای خالی در این جمله‌ها با کلمات مناسب: نویسنده، قرمز، سگ، یک کار پیش پا افتاده است. از بسیاری جهات، تشبیه قدیمی‌ترین نوع هوش است و چیزی است که ما انسان‌ها به‌طور غریزی همیشه انجام می‌دهیم.

در یادگیری ماشین، مقایسه‌گران با استفاده از توابع ریاضی قطعات داده‌ها را تطبیق می‌دهند تا بپرسند، "چه چیزهایی شبیه چیزهایی هستند که قبلاً دیده‌ام؟"

قیاس‌ها

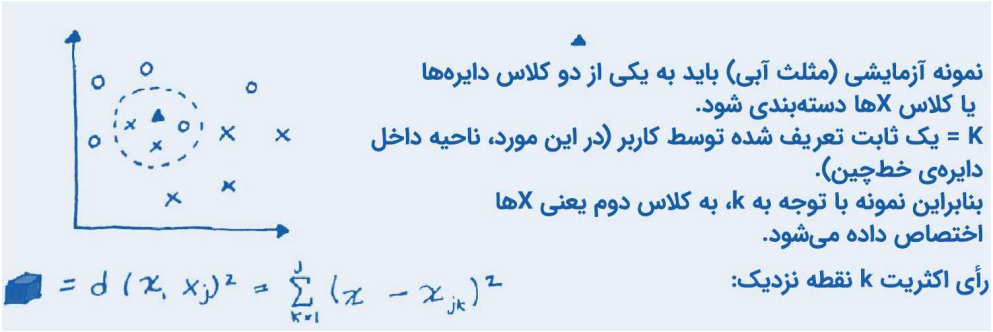
یادگیری از طریق قیاس، تشخیص شباهت‌ها بین موقعیت‌ها و استنباط شباهت‌های دیگر است.
مسئله‌ی کلیدی: قضاوت در مورد میزان شباهت دو چیز است.

بسیاری از سیستم‌های توصیه‌گر[۵۴] و دسته‌بندی[۵۵] بر پایه الگوریتم‌های قیاسی[۵۶] (مشابهت-محور) بنا شده‌اند. در یک سطح کلی، همه آن‌ها بر یک اصل مشترک کار می‌کنند و همگی در تلاش برای یافتن یا ساخت چیزی هستند که «مرز تصمیم[۵۷]» نامیده می‌شود؛ یک خط (یا سطح یا ناحیه) که یک کلاس را از کلاس دیگر جدا می‌کند.

ساده‌ترین این الگوریتم‌ها، k نزدیک‌ترین همسایه [58](k-NN) نامیده می‌شود. k به تعداد k (عدد) نزدیک‌ترین نمونه‌های آموزشی[۵۹] اشاره دارد. شکل زیر نحوه استفاده از k-NN را برای دسته‌بندی مثلث آبی (نمونه آزمایشی[۶۰]) به عنوان نزدیک‌ترین به X یا O (دو کلاس) نشان می‌دهد. در این مثال با استفاده از k=3، سه همسایه نزدیک مثلث آبی، دو X و یک O هستند که به این معنی است که مثلث آبی به کلاس X، یعنی نزدیک‌ترین کلاس خود، تخصیص داده می‌شود.

این روش نوعی «رأی اکثریت[۶۱]» است و اگر توزیع کلاس‌ها نامتوازن[۶۲] باشد، یعنی داده‌های یک کلاس بسیار بیشتر از کلاس دیگر باشد، دارای معایبی است.

k نزدیک‌ترین همسایه: یافتن نزدیک‌ترین (یا مشابه ترین) نقاط



---

۵۴ - سیستم‌های توصیه‌گر (Recommender Systems): سیستم‌هایی که به کاربران پیشنهادهایی مانند فیلم، محصول یا موسیقی می‌دهند (مانند سیستم‌های پیشنهاد فیلم نتفلیکس)

۵۵ - سیستم‌های دسته‌بندی (Classification Systems): سیستم‌هایی که ورودی‌ها را به دسته‌های از پیش تعریف شده اختصاص می‌دهند (مانند تشخیص ایمیل‌های اسپم).

۵۶ - الگوریتم‌های قیاسی (Analogous Algorithms): الگوریتم‌هایی که بر اساس شباهت بین داده‌ها عمل می‌کنند.

۵۷ - مرز تصمیم (Decision Boundary): مرزی که در فضای ویژگی‌ها، نمونه‌های متعلق به کلاس‌های مختلف را از هم جدا می‌کند.

58 - kنزدیک‌ترین همسایه (k-NN): یک الگوریتم یادگیری نظارت‌شده که برای دسته‌بندی یا رگرسیون استفاده می‌شود. در دسته‌بندی، یک نمونه جدید بر اساس کلاس k نزدیک‌ترین همسایه خود در داده‌های آموزشی، دسته‌بندی می‌شود.

۵۹ - نمونه آموزشی (Training Example): داده‌هایی که برای آموزش مدل استفاده می‌شوند

۶۰ - نمونه آزمایشی (Test Sample): داده‌هایی که برای ارزیابی عملکرد مدل آموزش دیده استفاده می‌شوند.

۶۱ - رأی اکثریت (Majority Vote): روشی برای تصمیم‌گیری که در آن، گزینه‌ای که بیشترین آرا را کسب کند، انتخاب می‌شود.

۶۲ - توزیع نامتوازن (Skewed Distribution): وضعیتی که در آن تعداد نمونه‌های یک کلاس بسیار بیشتر از کلاس‌های دیگر است. این موضوع می‌تواند باعث شود مدل به نفع کلاس غالب، سوگیری پیدا کند.

تصویر، مفهوم الگوریتم k نزدیک‌ترین همسایه (k-NN) را به صورت تصویری نشان می‌دهد. این الگوریتم یکی از ساده‌ترین و پرکاربردترین الگوریتم‌های یادگیری ماشین است که برای مسائل دسته‌بندی و رگرسیون استفاده می‌شود.

- **نمونه آزمایشی (Test Sample):** داده‌ای جدید که می‌خواهیم کلاس آن را پیش‌بینی کنیم (در اینجا مثلث آبی).
- **کلاس‌ها (Classes):** دسته‌های از پیش تعریف شده‌ای که داده‌ها می‌توانند به آن‌ها تعلق داشته باشند (در اینجا دایره‌ها وX ها).
- **K:** تعداد همسایه‌های نزدیک که برای تصمیم‌گیری در مورد کلاس نمونه آزمایشی در نظر گرفته می‌شوند. مقدار k توسط کاربر تعیین می‌شود.
- **رأی اکثریت (Majority Vote):** روشی برای تعیین کلاس نمونه آزمایشی بر اساس کلاس غالب در بین k همسایه نزدیک آن. به این معنی که کلاسی که بیشترین تعداد همسایه را در میان k همسایه نزدیک دارد، به عنوان کلاس نمونه آزمایشی انتخاب می‌شود.
- **فاصله اقلیدسی (Euclidean Distance):** معیاری برای محاسبه فاصله بین دو نقطه در فضای چند بعدی. فرمول ارائه شده در تصویر، نحوه محاسبه این فاصله را نشان می‌دهد. در واقع این فرمول، طول خط مستقیم بین دو نقطه را محاسبه می‌کند.

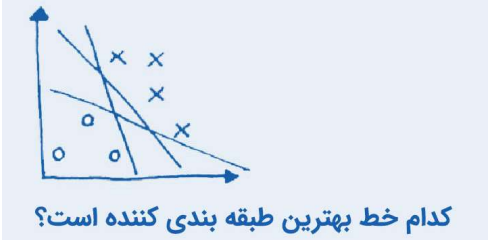
نحوه عملکرد الگوریتم k-NN در این مثال:

۱. ابتدا k (در اینجا ۳) همسایه نزدیک مثلث آبی پیدا می‌شوند. این همسایه‌ها نقاطی هستند که کمترین فاصله را با مثلث آبی دارند (داخل دایره‌ی خط‌چین).
۲. سپس کلاس این k همسایه بررسی می‌شود. در این مثال، دو X و یک دایره در بین سه همسایه نزدیک وجود دارد.
۳. با استفاده از روش رأی اکثریت، کلاسی که بیشترین تعداد همسایه را دارد (یعنی X)، به عنوان کلاس مثلث آبی انتخاب می‌شود.

نوع دیگری از قیاس‌گران، ماشین‌های بردار پشتیبان (SVM)[۶۳] هستند...

به جای تعیین مرز تصمیم‌گیری از طریق تنظیم یک مقدار برای k و محاسبه فواصل بین نمونه آزمایشی و نقاط داده (مرکز جرم داده‌ها)، می‌توان داده‌ها را با خطوط (یا سطوح در ابعاد بالاتر) جدا کرد. اولین گام این است که خطی را محاسبه کنیم که بهترین تفکیک را بین داده‌ها انجام دهد، یعنی مرز تصمیم‌گیری. اما اگر بخواهید دقیقاً نرخ مثبت‌های کاذب[۶۴] و منفی‌های کاذب[۶۵] را مشخص کنید، چه می‌شود؟

ماشین‌های بردار پشتیبان الگوریتم‌های پیچیده‌تری هستند که در آن داده‌ها به وکتورها تبدیل شده‌اند، که اساساً نمایش‌های فشرده‌ای از داده‌های خام هستند. وکتورها می‌توانند برای نمایندگی عددی انواع مختلف داده‌ها، مانند کلمات در یک سند، استفاده شوند. به عنوان مثال، جمله کوتاه "من از خرگوش‌ها متنفرم، اما خرگوش‌ها مرا دوست دارند" می‌تواند در یک بردار شمارش کلمات[۶۶] (۱، ۱، ۲، ۱، ۱، ۱) برای (من، متنفرم، خرگوش‌ها، اما، دوست دارند، من) نمایش داده شود.



کدام خط بهترین طبقه بندی کننده است؟

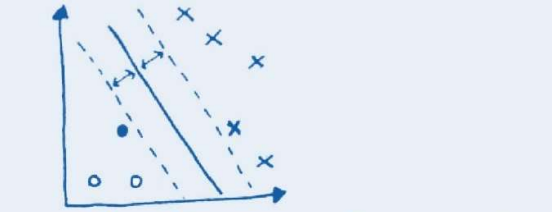
---

۶۳ - ماشین‌های بردار پشتیبان (SVM): یک الگوریتم یادگیری نظارت‌شده قدرتمند است که برای دسته‌بندی و رگرسیون استفاده می‌شود. SVM به دنبال یافتن بهترین ابرصفحه (خط در دو بعد، صفحه در سه بعد و ابرصفحه در ابعاد بالاتر) است که داده‌ها را با حداکثر حاشیه از هم جدا کند.

۶۴ - مثبت کاذب (False Positive): حالتی که در آن مدل، یک نمونه را به اشتباه به یک کلاس مثبت نسبت می‌دهد، در حالی که در واقع منفی است.

۶۵ - منفی کاذب (False Negative): حالتی که در آن مدل، یک نمونه را به اشتباه به یک کلاس منفی نسبت می‌دهد، در حالی که در واقع مثبت است.

۶۶ - بردار شمارش کلمات (Word Count Vector): روشی برای نمایش متن به صورت عددی که در آن، تعداد تکرار هر کلمه در متن شمرده می‌شود.



اضافه کردن یک "حاشیه ایمنی" (خطا) "حمایت شده" توسط "بردارها" از هر طرف.

در الگوریتم SVM، هدف پیدا کردن بهترین خط (یا ابرصفحه در ابعاد بالاتر) است که دو کلاس مختلف داده را از هم جدا کند. «بهترین» به این معنی است که این خط باید بیشترین فاصله را از نزدیک‌ترین نقاط داده هر دو کلاس داشته باشد. این فاصله، «حاشیه ایمنی» یا به اصطلاح «حداکثر حاشیه» نامیده می‌شود.

- خط جداکننده (Decision Boundary/Hyperplane): خطی که دو کلاس را از هم جدا می‌کند (در تصویر، خط مورب).
- حاشیه ایمنی (Margin): ناحیه‌ای بین خط جداکننده و نزدیک‌ترین نقاط داده هر کلاس. هدف SVM، بیشینه‌سازی این حاشیه است.
- بردارهای پشتیبان (Support Vectors): نقاط داده‌ای که دقیقاً روی حاشیه یا در داخل آن قرار دارند. این نقاط، در تعیین موقعیت و جهت خط جداکننده نقش حیاتی دارند. در واقع، اگر سایر نقاط داده حذف شوند، موقعیت خط جداکننده همچنان توسط این بردارها تعیین می‌شود.
- خطا (Error): اشاره به این دارد که ممکن است برخی از نقاط داده به درستی دسته‌بندی نشوند و در داخل حاشیه یا حتی در طرف اشتباه خط جداکننده قرار بگیرند. با این حال، هدفSVM ، کاهش این خطا در عین بیشینه‌سازی حاشیه است.

چرا حاشیه ایمنی مهم است؟

تعمیم بهتر (Better Generalization): حاشیه بزرگتر باعث می‌شود مدل به داده‌های جدیدی که در آینده با آن‌ها مواجه می‌شود، بهتر تعمیم پیدا کند و عملکرد بهتری داشته باشد. به عبارت دیگر، احتمال اینکه مدل داده‌های جدید را به درستی دسته‌بندی کند، بیشتر می‌شود.

مقاومت در برابر داده‌های پرت (Robustness to Outliers): حاشیه بزرگتر، مدل را در برابر داده‌های پرت (نقاط داده‌ای که از الگوی کلی داده‌ها پیروی نمی‌کنند) مقاوم‌تر می‌کند.

ماشین‌های بردار پشتیبان، مرز تصمیم را بر اساس بردارهای پشتیبان[۶۷] (به معنای واقعی کلمه، چیزی که خط برای «حمایت» به آن تکیه می‌کند) در هر طرف که به عنوان «حاشیه ایمنی[۶۸]» عمل می‌کنند، حل می‌کنند. این مرز هموارتر به آن‌ها کمک می‌کند تا دسته‌بندهای بسیار دقیقی باشند. همچنین باعث می‌شود که آن‌ها در مجموعه‌های داده بزرگ بسیار سریع و کارآمد باشند، زیرا پس از شناسایی بردارهای پشتیبان، می‌توان بسیاری از داده‌های دیگر را دور ریخت. این ممکن است در مجموعه‌های داده کوچک اهمیت زیادی نداشته باشد، اما وقتی میلیون‌ها نقطه داده قابل حذف باشند، می‌توان زمان زیادی را صرفه‌جویی کرد.

قیاس‌گران (الگوریتم‌های قیاسی) وقتی صحبت از مقابله با ویژگی‌های غیرخطی[۶۹] در داده‌ها باشد، قدرتمند هستند، زیرا می‌توانند داده‌ها را در ابعاد مختلف مدیریت کنند. هنگامی که ویژگی‌های غیرخطی وجود داشته باشد، این الگوریتم‌ها می‌توانند از تکنیک‌های ریاضی استفاده کنند که ابعاد[۷۰] کافی برای جداسازی داده‌ها به کلاس‌های متمایز ایجاد می‌کنند. این

---

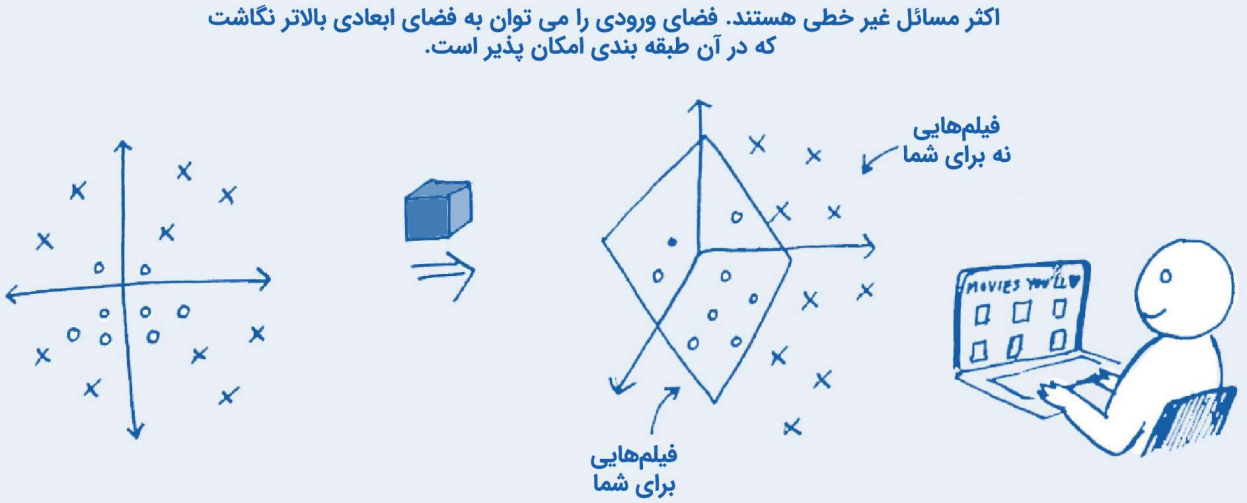
۶۷ - بردارهای پشتیبان (Support Vectors): نقاط داده‌ای هستند که نزدیک‌ترین فاصله را با مرز تصمیم دارند و در تعیین موقعیت و جهت آن نقش کلیدی ایفا می‌کنند.

۶۸ - حاشیه ایمنی (Safety Margin): ناحیه‌ای در اطراف مرز تصمیم که توسط بردارهای پشتیبان تعریف می‌شود. بیشینه‌سازی این حاشیه، هدف اصلی SVM است.

۶۹ - ویژگی‌های غیرخطی (Non-linear Features): ویژگی‌هایی در داده‌ها که رابطه خطی با یکدیگر ندارند. در این حالت، نمی‌توان داده‌ها را با یک خط مستقیم یا یک صفحه ساده از هم جدا کرد.

۷۰ - ابعاد (Dimensions): تعداد ویژگی‌های موجود در داده‌ها. افزایش ابعاد می‌تواند به جداسازی بهتر داده‌ها در فضایی با ابعاد بالاتر کمک کند.

بدان معنی است که آن‌ها می‌توانند به راحتی و به سرعت حجم زیادی از ورودی‌های در حال تغییر را مرتب کنند و ورودی‌ها را به کلاس‌های متمایز ارسال کنند. تصور کنید این کمی شبیه بازیکنان راگبی[۷۱] در یک درگیری نزدیک (راک) باشد. وقتی سوت زده می‌شود، آن‌ها از درهم‌تنیدگی خود جدا می‌شوند و به عنوان دو گروه متمایز در داخل خط تیم خود دوباره شکل می‌گیرند. یک ترفند عالی.



توضیح ساده این تصویر این است:

بسیاری از مسائل در دنیای واقعی به گونه‌ای هستند که نمی‌توان آنها را با یک خط مستقیم یا یک صفحه ساده (یعنی به صورت خطی) حل کرد. به این مسائل، مسائل «غیرخطی» می‌گویند. اما یک ترفند وجود دارد: می‌توان داده‌های این مسائل را به یک «فضای ابعادی بالاتر» برد. در این فضای جدید، که ابعاد بیشتری دارد، ممکن است بتوان داده‌ها را به راحتی با یک خط یا صفحه از هم جدا کرد و دسته‌بندی کرد.

به عبارت دیگر، فرض کنید می‌خواهید دو دسته از توپ‌ها را از هم جدا کنید. اگر توپ‌ها روی یک خط باشند و به صورت درهم قرار گرفته باشند، جدا کردن آنها با یک خط ساده ممکن نیست (مسئله غیرخطی است). اما اگر این توپ‌ها را به هوا پرتاب کنید (یعنی به فضای سه بعدی ببرید)، احتمالاً می‌توانید با یک صفحه آنها را از هم جدا کنید. این مثال ساده نشان می‌دهد که چگونه رفتن به فضای ابعادی بالاتر می‌تواند مسائل غیرخطی را به مسائل خطی تبدیل کند که راحت‌تر حل می‌شوند.

در حوزه هوش مصنوعی و یادگیری ماشین، این «نگاشت به فضای ابعادی بالاتر» با استفاده از توابع ریاضی خاص انجام می‌شود. به این ترتیب، الگوریتم‌ها می‌توانند مسائل پیچیده غیرخطی را با استفاده از تکنیک‌های ساده‌تر در فضای جدید حل کنند.

قیاس‌گران (الگوریتم‌های قیاسی) در زندگی روزمره ما بسیار رایج هستند. آن‌ها فناوری پشت بسیاری از برنامه‌های کاربردی رایج هوش مصنوعی در تجارت الکترونیک را فراهم می‌کنند. آن‌ها از طریق داده‌های جدید یاد می‌گیرند و در یک چشم به هم زدن، مرزهای تصمیم جدید را دوباره محاسبه می‌کنند.

فرض کنید شما فیلم‌های هیجان‌انگیز را ترجیح می‌دهید. ارائه‌دهنده محتوای پخش آنلاین شما، این را می‌داند و دائماً نسخه‌های جدید را بررسی می‌کند تا فیلم‌هایی از این ژانر را که فکر می‌کند می‌خواهید تماشا کنید، به شما ارائه دهد. پیش‌بینی می‌کند که چون شما فیلم Momento را تماشا کرده‌اید، از Argo نیز لذت خواهید برد. اما به جای انتخاب Argo، شما The

---

۷۱ - ترفند راگبی :(Rugby Trick) تشبیه به بازیکنان راگبی در یک "راک" (درگیری نزدیک) برای نشان دادن نحوه جداسازی سریع و کارآمد داده‌ها توسط الگوریتم‌های قیاسی، به خصوص SVM ، استفاده شده است.

این متن به مزایای کلیدی SVM و الگوریتم‌های قیاسی اشاره می‌کند:

دقت بالا: به دلیل وجود حاشیه ایمنی و استفاده از بردارهای پشتیبان.

کارایی در داده‌های بزرگ: به دلیل امکان حذف داده‌های غیرضروری پس از شناسایی بردارهای پشتیبان.

توانایی مدیریت ویژگی‌های غیرخطی: با استفاده از تکنیک‌های ریاضی برای افزایش ابعاد و جداسازی داده‌ها در فضایی با ابعاد بالاتر.

این ویژگی‌ها SVM و سایر الگوریتم‌های قیاسی را به ابزاری قدرتمند برای مسائل دسته‌بندی و تشخیص الگو تبدیل می‌کند، به خصوص در مواردی که با داده‌های پیچیده و غیرخطی سروکار داریم.

Good, the Bad and the Ugly را انتخاب می‌کنید. بنابراین الگوریتم دوباره مرتب‌سازی می‌کند و دفعه بعد پیشنهادی برای No Country for Old Men اضافه می‌کند. سپس شخصی از حساب شما برای تماشای Thelma and Louise استفاده می‌کند و دفعه بعد که وارد سیستم می‌شوید، پیشنهادی برای Gone Girl دارید.

دفعه بعد که به دنبال فیلم می‌گردید، در نظر داشته باشید که الگوریتمی وجود دارد که با ترجیحات شما، آنچه قبلاً تماشا کرده‌اید، موارد جدید و آنچه جستجو کرده‌اید، سازگار می‌شود. تصور کنید اگر الگوریتم به داده‌های بیشتری در مورد شما دسترسی داشته باشد - اینکه با چه کسی هستید، آن روز کجا بوده‌اید، چه سفری را برنامه‌ریزی می‌کنید، چه اخباری را خوانده‌اید - و انتخاب خود را بر اساس اطلاعات شخصی بسیار بیشتر تنظیم کند.

قیاس‌گران، شخصی‌سازی را تقویت می‌کنند.

راهنمایی انسان برای ماشین هایی که یاد می گیرند

«یادگیری ماشین به طور شگفت‌انگیزی خوب عمل می‌کند، اما ریاضیدان‌ها دقیقاً نمی‌دانند چرا.»

اینگرید دابچی

در مقابل

«٪۹۹ از یادگیری ماشین، کار انسان‌هاست.»

اورن ایتزیونی

من در مورد ریاضیات داخل جعبه سیاه صحبت کرده‌ام: یادگیری عمیق، بردارهای پشتیبان، احتمال‌های پسین. حالا وقت آن است که در مورد شرایط انسانی صحبت کنم.

راه‌اندازی یک الگوریتم یادگیری ماشین خوب اغلب به عنوان فرآیندی که نیاز به شهود فراوان دارد و تا حدی شبیه به یک «هنر سیاه[۷۲]» توصیف شده است. درک اهرم‌هایی که انسان‌ها با آن‌ها تعامل دارند و قضاوت‌هایی که در هوش ماشینی صورت می‌گیرد، به شما کمک می‌کند تا عمیق‌تر بفهمید چه چیزهایی لازم است تا یک ماشین «هوشمند» شود.

تصمیمات زیادی وجود دارد که هنگام ساخت یک الگوریتم یادگیری مفید، انسان‌ها می‌گیرند؛ این تصمیمات هم مهم هستند و هم ظریف.

راه اندازی و تنظیم یک الگوریتم خوب یادگیری ماشین، به عنوان چیزی که نیازمند شهود زیاد است که به نوعی یک "هنر سیاه" توصیف شده است. داشتن بینشی در مورد اهرم‌هایی[۷۳] که انسان ها با آن بازی می کنند و قضاوت‌هایی[۷۴] که در هوش ماشین انجام می شود، درک شما را در مورد اینکه چه چیزی برای ساختن یک ماشین "هوشمند" لازم است، عمیق تر می کند. هنگامی که نوبت به ساخت یک الگوریتم یادگیری مفید می رسد، تصمیمات مهم و ظریف بسیاری وجود دارد که انسان می‌گیرد.

72- a technique or practice considered mysterious and sinister. a skill or ability that seems mysterious or magical often used humorously:
تکنیک یا عملی که مرموز تلقی می‌شود. مهارت یا توانایی که مرموز یا جادویی به نظر می رسد اغلب به صورت طنز استفاده می‌شود

۷۳- Levers (اهرم‌ها): در اینجا، «اهرم‌ها» به پارامترها و تنظیماتی اشاره دارند که انسان در یک الگوریتم یادگیری ماشین می‌تواند تغییر دهد تا عملکرد آن را بهبود بخشد که می‌تواند شامل انتخاب نوع الگوریتم، تنظیم پارامترهای آن، انتخاب ویژگی‌های داده‌ها و غیره باشد.

۷۴- Judgments (قضاوت‌ها): در این زمینه، «قضاوت‌ها» به تصمیماتی اشاره دارند که انسان در طول فرآیند ساخت و تنظیم یک الگوریتم یادگیری ماشین می‌گیرد. این می‌تواند شامل انتخاب روش ارزیابی، تصمیم‌گیری در مورد میزان داده‌های آموزشی، و تفسیر نتایج باشد.

چیزی که یادگیری ماشین را از منطق سخت[۷۵] سیستم‌های خبره[۷۶] متمایز می‌کند این است که تعداد ترکیب‌های ممکن ورودی‌ها و خروجی‌ها بسیار زیاد و گاهی حتی نامحدود است. داشتن یک روش هوشمندانه برای اعتبارسنجی (آزمایش و ارزیابی)[۷۷] عملکرد، کاملاً حیاتی است. چه در شخصی‌سازی صفحات وب، جایی که الگوریتم‌ها می‌توانند روزانه میلیاردها صفحه را به‌صورت آنی[۷۸] سفارشی کنند، و چه در شناسایی ساختارهای مولکولی جدید با سرعت میلیون‌ها مورد در ثانیه برای کشف دارو، اطمینان از اینکه ماشین در حال بررسی داده‌های درست است، الگوریتم‌های صحیح را با دقت اجرا می‌کند و پیش‌بینی‌های مناسبی ارائه می‌دهد، امری کلیدی است.

بزرگ‌تر بودن لزوماً بهتر نیست، سادگی می‌تواند بر پیچیدگی غلبه کند ...

با پیچیده‌تر شدن الگوریتم‌ها، ممکن است انتخاب مدل‌های پیچیده‌تر، وسوسه‌انگیز باشد. در هر الگوریتم، تنظیمات و دستگیره‌های[۷۹] زیادی برای تغییر وجود دارد، اما پیچیدگی لزوماً بر مدل‌های ساده‌تر که توسط انسان‌های باهوش و خلاق به‌خوبی اجرا می‌شوند، برتری ندارد. در حالی که برخی ماشین‌ها بدون کنترل مستقیم انسان به‌طور موفقیت‌آمیزی در حال یادگیری و کشف هستند، اکثریت قریب به اتفاق کاربردهای یادگیری ماشین همچنان با دخالت زیاد انسان[۸۰] همراه هستند.

ورودی انسان شماره ۱: دانش حوزه

یادگیری ماشین یک اهرم دانش[۸۱] است؛ راهی برای دستیابی به نتایج بیشتر با منابع کمتر. این بدان معناست که یک معیار کلیدی برای انتخاب رویکرد، داشتن درک خوبی از انواع دانشی است که با آن روش سازگار هستند.

گاهی اوقات در آغاز کار، دانش و داده کافی برای انتخاب یک الگوریتم مناسب وجود ندارد. در این حالت، الگوریتم ممکن است تنها الگوهای تصادفی و عجیب را رمزگذاری کند. این پدیده «بیش‌برازش[۸۲]» نامیده می‌شود و دومینگوس آن را «مشکل آزاردهنده و رایج[۸۳] یادگیری ماشین» توصیف می‌کند. این پدیده برخلاف شهود است و به این معناست که استفاده از یک یادگیرنده قدرتمندتر لزوماً بهتر از یک یادگیرنده کم‌قدرت‌تر نیست.

این نکته زمانی اهمیت پیدا می‌کند که می‌خواهید محصولی مبتنی بر یادگیری ماشین را از یک فروشنده ارزیابی کنید، کسی که داده‌های شما را گرفته و یک نسخه آزمایشی به شما ارائه می‌دهد. در دنیای یادگیری ماشین، این ممکن است یک برازش

۷۵- Hard logic (منطق سخت): در اینجا به منطق قطعی و از پیش تعریف شده در سیستم‌های خبره اشاره دارد که بر اساس قوانین "اگر-آنگاه" (if-then) عمل می‌کنند. در مقابل، یادگیری ماشین بر اساس داده‌ها یاد می‌گیرد و الگوها را شناسایی می‌کند، نه قوانین از پیش تعیین شده.

۷۶- Expert system (سیستم خبره): یک سیستم کامپیوتری که دانش یک یا چند متخصص انسانی را در یک حوزه خاص شبیه‌سازی می‌کند و می‌تواند برای حل مسائل در آن حوزه مورد استفاده قرار گیرد.

۷۷- Validate performance (اعتبارسنجی عملکرد): فرآیندی برای ارزیابی و تأیید صحت و کارایی یک سیستم یا الگوریتم است.

۷۸- Real time (بی‌درنگ): به پردازش و پاسخگویی به داده‌ها در همان لحظه دریافت آن‌ها اشاره دارد، بدون تأخیر قابل توجه.

۷۹- Knobs and dials (دستگیره‌ها و تنظیمات): این اصطلاح به پارامترها و تنظیمات مختلفی اشاره دارد که می‌توان در یک الگوریتم تغییر داد تا عملکرد آن را تنظیم کرد. این اصطلاح، تصویری از دستگاهی با دکمه‌ها و چرخ‌های تنظیم را ارائه می‌دهد که کاربر می‌تواند آن‌ها را برای رسیدن به نتیجه مطلوب تغییر دهد.

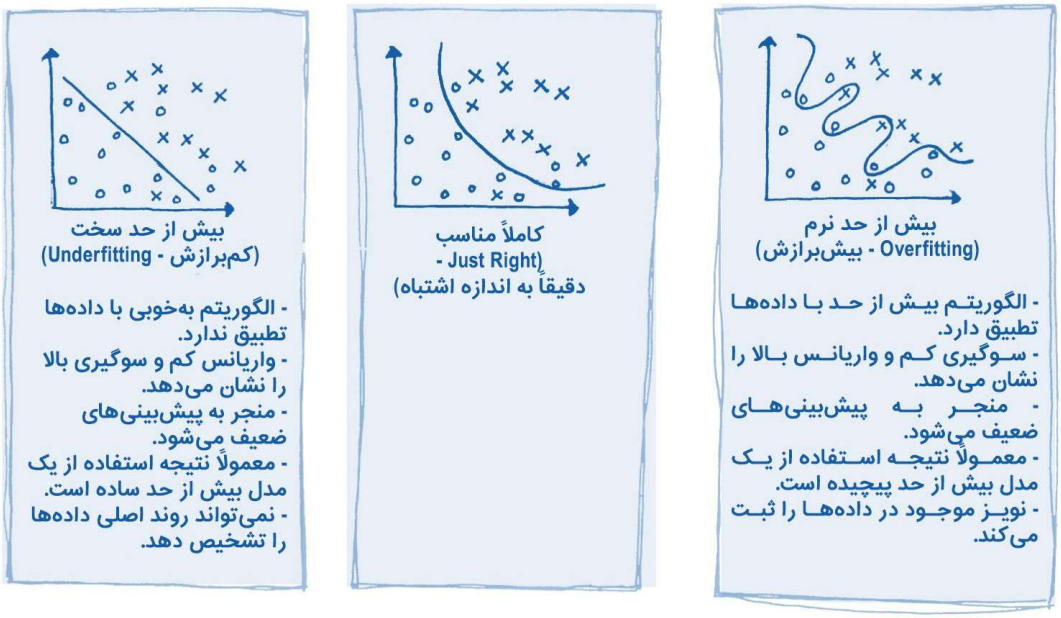
۸۰- Human involvement (دخالت انسان): این عبارت به نقش انسان در فرآیند توسعه، آموزش، و استفاده از الگوریتم‌های یادگیری ماشین اشاره دارد. این شامل مواردی مانند انتخاب داده‌ها، طراحی الگوریتم، تنظیم پارامترها، و تفسیر نتایج می‌شود.

۸۱- Knowledge lever (اهرم دانش): این اصطلاح به این ایده اشاره دارد که یادگیری ماشین می‌تواند به ما کمک کند تا از داده‌های موجود، دانش بیشتری استخراج کنیم، مانند استفاده از یک اهرم برای بلند کردن وزنی سنگین با نیروی کمتر

۸۲- Overfitting (بیش‌برازش): وضعیتی که در آن یک مدل یادگیری ماشین به جای یادگیری الگوهای کلی و واقعی در داده‌ها، الگوهای تصادفی و نویز موجود در داده‌های آموزشی را حفظ می‌کند. در نتیجه، مدل در داده‌های آموزشی عملکرد بسیار خوبی دارد، اما در داده‌های جدید عملکرد ضعیفی از خود نشان می‌دهد.

۸۳- Bugbear (غول بی شاخ و دُم): این اصطلاح به مشکلی رایج و آزاردهنده اشاره دارد.

سفارشی[۸۴] نباشد، بلکه می‌تواند صرفاً یک بیش‌برازش باشد. دیدن اینکه برنامه چگونه با پیش‌بینی داده‌های جدید عمل می‌کند، تنها چیزی است که اهمیت دارد.



این تصویر به زبان ساده می‌خواهد توضیح دهد که در یادگیری ماشین، پیدا کردن تعادل بین سادگی مدل و پیچیدگی مدل بسیار مهم است. سه حالت کلی بررسی شده است:

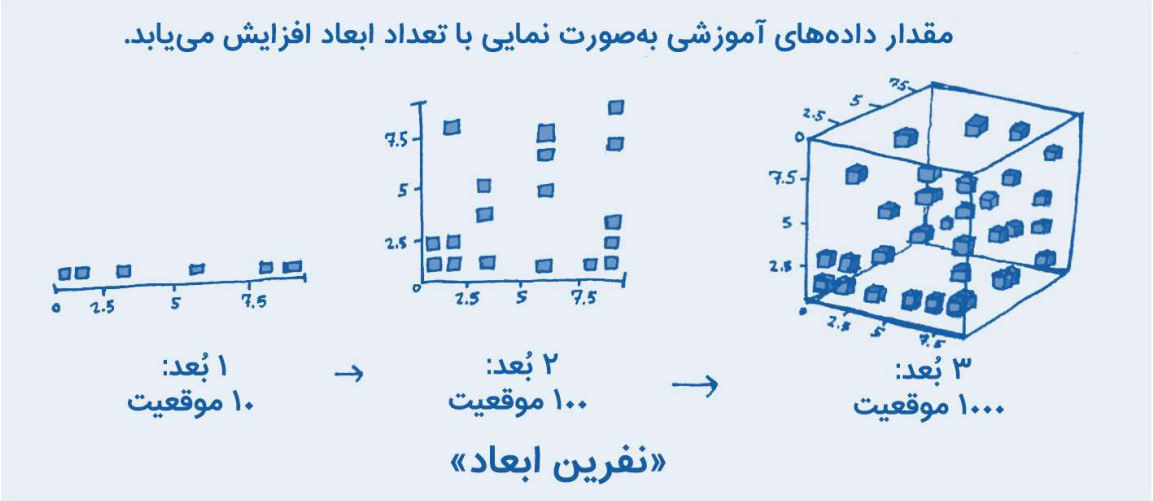
۱. کم‌برازش (Underfitting): وقتی مدل بیش از حد ساده باشد، نمی‌تواند الگوهای اصلی داده‌ها را درک کند. به زبان ساده، انگار الگوریتم «درس را خوب یاد نگرفته» و پیش‌بینی‌های آن معمولاً اشتباه است.
۲. بیش‌برازش (Overfitting): وقتی مدل بیش از حد پیچیده شود، تمام جزئیات و حتی نویزهای بی‌اهمیت داده‌ها را یاد می‌گیرد. این یعنی الگوریتم «بیش از حد وسواسی» شده و وقتی داده جدیدی ببیند، نمی‌تواند خوب پیش‌بینی کند.
۳. تعادل مناسب (Just Right): حالتی که مدل نه بیش از حد ساده است و نه بیش از حد پیچیده. در این حالت، مدل به‌اندازه کافی الگوهای اصلی داده را یاد می‌گیرد و می‌تواند روی داده‌های جدید عملکرد خوبی داشته باشد.

به زبان خیلی ساده، هدف این است که مدل «درس را خوب یاد بگیرد»، اما نه آنقدر کم‌خوانی کند که مفهوم را نفهمد، و نه آنقدر جزئیات غیرضروری را حفظ کند که در امتحان (داده‌های جدید) گیج شود.

مشکل بزرگ بعدی «نفرین ابعاد» (Curse of Dimensionality) است. با افزایش تعداد ویژگی‌ها (و در نتیجه ابعاد)، مقدار داده‌ای که برای «پوشش دادن» فضای حاصل نیاز است، به‌صورت نمایی افزایش می‌یابد. در برخی مسائل، ممکن است هرگز داده کافی وجود نداشته باشد و اضافه کردن ویژگی‌های بیشتر به‌منظور اضافه کردن داده، مشکلات بیشتری ایجاد کند تا اینکه آن‌ها را حل کند.

داده‌ها، ماده خام یادگیری ماشین هستند و به‌طور کلی، داده‌های بیشتر بهتر است. اما نکته کلیدی این است که بدانیم چه زمانی این اصل صدق نمی‌کند. برای این کار، نیاز به افرادی دارید که مسئله‌ای را که قصد حل آن را دارید، به‌خوبی درک کنند.

۸۴- Custom fit (برازش سفارشی): به مدلی اشاره دارد که به طور خاص برای یک مجموعه داده یا مسئله خاص طراحی شده و عملکرد خوبی دارد.



این افزایش نمایی در حجم داده مورد نیاز با افزایش ابعاد، همان «نفرین ابعاد» است. به عبارت دیگر، هرچه تعداد ویژگی‌هایی که برای توصیف داده‌ها استفاده می‌کنیم بیشتر شود، برای آموزش مؤثر یک مدل یادگیری ماشین، به داده‌های بسیار بیشتری نیاز داریم. در برخی موارد، جمع‌آوری این حجم از داده عملاً غیرممکن است.

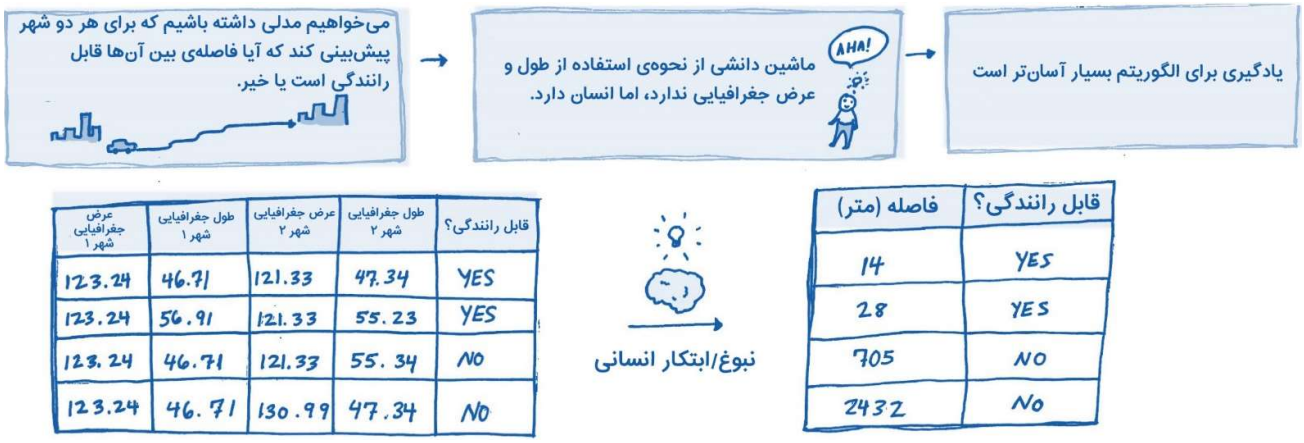
نکته مهم این است که افزایش ابعاد، فضای داده را بسیار پراکنده می‌کند و باعث می‌شود که داده‌ها به طور مؤثر فضای داده را پوشش ندهند. این امر منجر به مشکلاتی مانند بیش‌برازش (overfitting) و کاهش دقت مدل می‌شود.

بنابراین، در طراحی مدل‌های یادگیری ماشین، انتخاب ویژگی‌های مناسب و کاهش ابعاد داده‌ها از اهمیت بالایی برخوردار است.

ورودی انسان شماره ۲: مهندسی ویژگی ها

مهندسی ویژگی‌ها بخش عمده‌ای از تلاش در یک پروژه یادگیری ماشین را به خود اختصاص می‌دهد یعنی به‌طور متوسط ۷۰٪ از زمان پروژه. مهندسی ویژگی همان جایی است که انسان‌ها برتری دارند، جایی که مهارت‌های ذاتی خلاقیت و شهود به موفقیت منجر می‌شوند. انسان‌ها می‌توانند ببینند که چگونه اطلاعات را کنار هم قرار دهند و داده‌ها و ورودی‌هایی طراحی کنند که الگوریتم‌ها را قادر سازد تا راحت‌تر تابع مناسب را پیدا کنند.

یک مثال زیبا از مهندسی ویژگی این است که ویژگی‌ها به گونه‌ای طراحی شوند که کامپیوتر بتواند پیش‌بینی کند آیا امکان رانندگی بین دو شهر وجود دارد یا خیر، بر اساس عرض جغرافیایی و طول جغرافیایی شهرها. در این مثال، کامپیوتر دانش استفاده از عرض و طول جغرافیایی را ندارد، زیرا هیچ‌کدام به طور مستقیم با «قابل رانندگی بودن» همبستگی نزدیکی ندارند. اما یک انسان می‌تواند بفهمد که تغییر ویژگی‌ها می‌تواند به الگوریتم کمک کند تا یک مدل خوب بسازد.



این تصویر مفهوم مهندسی ویژگی (Feature Engineering) را به زبان ساده توضیح می‌دهد. هدف، ساخت مدلی است که بتواند تشخیص دهد آیا بین دو شهر می‌توان رانندگی کرد یا خیر.

- در ابتدا، داده‌ها به صورت عرض و طول جغرافیایی دو شهر ارائه می‌شوند. مشکل اینجاست که الگوریتم به طور مستقیم نمی‌تواند از این داده‌ها برای تشخیص «قابل رانندگی بودن» استفاده کند. زیرا هیچ رابطه‌ی مستقیمی بین عرض/طول جغرافیایی و قابل رانندگی بودن وجود ندارد.
- اینجاست که «نبوغ انسانی» وارد می‌شود. انسان با دانش خود می‌تواند ویژگی جدیدی به نام «فاصله» (Distance) را از روی عرض و طول جغرافیایی محاسبه کند.
- با اضافه شدن ویژگی «فاصله»، یادگیری برای الگوریتم بسیار ساده‌تر می‌شود. زیرا اکنون یک رابطه‌ی مستقیم بین «فاصله» و «قابل رانندگی بودن» وجود دارد. فاصله‌های کم معمولاً قابل رانندگی هستند و فاصله‌های زیاد معمولاً قابل رانندگی نیستند (حداقل بدون پرواز).

به عبارت دیگر، مهندسی ویژگی به معنای تبدیل داده‌های خام به ویژگی‌هایی است که برای الگوریتم معنادارتر و قابل استفاده‌تر باشند. این کار اغلب نیازمند دانش و شهود انسانی است.

داده‌های خام نامرتب هستند. جمع‌آوری، پاکسازی، یکپارچه‌سازی[۸۵] و پیش‌پردازش[۸۶] داده‌ها بسیار زمان‌بر است و حتی پس از آن، اغلب به شکلی نیستند که بتوان از آن‌ها یاد گرفت. مهندسی ویژگی روشی است که داده‌های خام از یک شکل به شکل دیگری تبدیل می‌شوند که همبستگی بهتری با مسئله‌ای که به آن پرداخته می‌شود، دارد.

برچسب‌گذاری داده‌ها و مهندسی ویژگی با بهبود پردازش داده‌های بدون ساختار[۸۷] توسط ماشین‌ها، کمتر به یک مسئله تبدیل خواهند شد. رویکردهای ترکیبی بسیاری در حال تحقیق هستند که شامل روش‌های برون‌سپاری جمعی[۸۸] یا یادگیری عمیق برای استخراج ویژگی‌ها می‌شوند. با گذشت زمان، ممکن است اهمیت مهندسی ویژگی کاهش یابد، اما در حال حاضر، برای بیشتر هوش ماشینی، این یک جایگاه مهم برای هدایت انسانی است.

ورودی انسانی شماره ۳: برخورد با اینکه آیا مدل به اندازه کافی خوب کار می‌کند یا خیر

هنگامی که داده‌ها آماده استفاده می‌شوند، یادگیرنده بر روی بخش کوچکی از داده‌ها، «مجموعه آموزشی[۸۹]»، آموزش داده می‌شود. بخشی از داده‌ها نگه داشته می‌شوند به نام «مجموعه آزمایشی[۹۰]». غیرمعمول نیست که انسان‌ها این مرحله را نادیده بگیرند و نتایج عالی از مجموعه آموزشی بدون آزمایش کافی بر روی داده‌های نگه داشته شده به دست آورند. به محض اینکه داده‌های جدید به یادگیرنده ارائه می‌شوند، نتایج در نهایت، چیزی بهتر از حدس زدن تصادفی[۹۱] نخواهند بود.

اگر تلاشی برای ساخت ویژگی‌های خوب انجام شده باشد و الگوریتم بر روی داده‌های مناسب آموزش داده و آزمایش شده باشد، اما خروجی دقیق نباشد، انسان‌ها دو انتخاب دارند: طراحی یک الگوریتم یادگیری بهتر یا جمع‌آوری داده‌های بیشتر. به طور کلی، داده‌های بیشتر برنده می‌شوند، اما محدودیت‌هایی وجود دارد.

۸۵- Integrate (یکپارچه‌سازی): ترکیب داده‌ها از منابع مختلف در یک مجموعه داده واحد.

۸۶- Pre-process (پیش‌پردازش): آماده‌سازی داده‌ها برای استفاده در الگوریتم‌های یادگیری ماشین، شامل پاکسازی، تبدیل و نرمال‌سازی داده‌ها.

۸۷- Unstructured data (داده‌های بدون ساختار): داده‌هایی که فرمت مشخص و از پیش تعریف شده‌ای ندارند، مانند متن، تصاویر و ویدیوها.

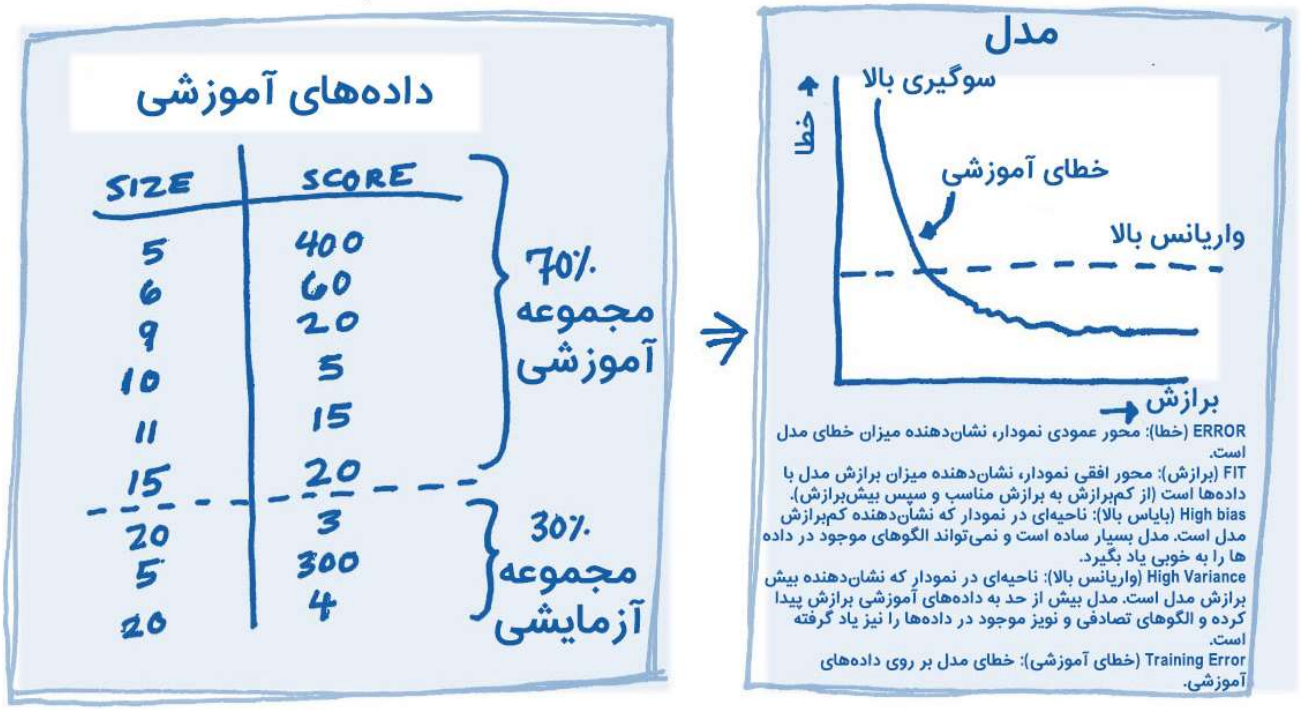
۸۸- Crowd sourcing (برون‌سپاری جمعی): استفاده از تعداد زیادی از افراد برای انجام یک کار یا حل یک مشکل.

۸۹- Training set (مجموعه آموزشی): بخشی از داده‌ها که برای آموزش مدل یادگیری ماشین استفاده می‌شود.

۹۰- Testing set (مجموعه آزمایشی): بخشی از داده‌ها که برای ارزیابی عملکرد مدل آموزش دیده استفاده می‌شود و در طول آموزش به مدل نشان داده نمی‌شود.

۹۱- Random guessing (حدس زدن تصادفی): پیش‌بینی بدون هیچگونه مبنای منطقی یا اطلاعاتی، به طوری که احتمال صحت آن برابر با شانس است.

اکنون که جهان مملو از داده است، محدودیت به زمان منتقل شده است. زمان کافی برای پردازش تمام داده‌های موجود وجود ندارد و بسیاری از آن‌ها بدون استفاده باقی می‌مانند. نقش هوشمندانه انسان، انتخاب تعادل بهینه بین پیچیدگی الگوریتم[۹۲]، پیچیدگی داده[۹۳] و قدرت محاسباتی[۹۴] است. مثل همیشه، دقت پیش‌بینی چیزی است که اهمیت دارد.



برای جمع بندی...

شما اکنون می‌دانید که هوش مصنوعی امروزی در مورد استفاده از ماشین‌هایی است که یاد می‌گیرند،

شما می‌دانید که ماشین‌هایی که یاد می‌گیرند توسط مجموعه‌ای از الگوریتم‌های ریاضی هدایت می‌شوند،

۹۲- Algorithm complexity (پیچیدگی الگوریتم): میزان منابع محاسباتی (مانند زمان و حافظه) که یک الگوریتم برای اجرا نیاز دارد.

۹۳- Data complexity (پیچیدگی داده): میزان پیچیدگی و تنوع داده‌ها.

۹۴- Computing power (قدرت محاسباتی): توانایی یک سیستم کامپیوتری برای پردازش داده‌ها.

شما می‌دانید که آن‌ها از داده‌ها برای به‌روزرسانی مداوم برنامه‌های خود بدون ورودی انسانی استفاده می‌کنند،

شما همچنین می‌دانید که عملکرد داخلی آن‌ها لزوماً شفاف نیست،

و اینکه وجود انسان‌ها برای دستیابی به نتیجه درست، کلیدی هستند.

در نهایت، شما اکنون می‌دانید که این الگوریتم‌ها چه نوع کاربردهایی را هدایت می‌کنند.

به طور فزاینده‌ای آشکار می‌شود که یادگیری ماشین محرک جدیدی برای مزیت رقابتی است. شرکت‌هایی که از سرعت و مقیاس یادگیری ماشین برای ارائه مثلا توصیه‌های بهتر یا دستیابی به پیشرفت‌های چشمگیر در پزشکی استفاده می‌کنند، پیشرفت خواهند کرد. پیشرفت، ایجاد داده‌های بیشتر را به دنبال خواهد داشت، داده‌های بیشتر، یادگیری بیشتر را به دنبال خواهد داشت. درست مانند دوچرخه‌سوارانی که با هم مسابقه میدهند و کسانی را که نمی‌توانند پشت سر می‌گذارند، یادگیری ماشین مزیتی ایجاد می‌کند که اگر آن را نداشته باشید، ادامه دادن کار دشوار خواهد بود.

شش نکته پیش از شروع

شماره ۱: بدانید چه مشکلی را می‌خواهید حل کنید

هر چند این نکته واضح به نظر می‌رسد، اما شگفت‌آور است که چقدر افراد زیادی هستند که از این مرحله به سرعت عبور می‌کنند و فرض می‌کنند که مشکل را به خوبی تعریف کرده‌اند یا همه بر سر آب با هم توافق دارند. تمام پیچیدگی‌های پیش رو و تصمیمات کلیدی - چه داده‌ای، چه مقیاسی، چه دقتی برای پیش‌بینی - به وضوحِ صورت مسئله بستگی خواهد داشت. اگر تصمیم بگیرید با این فناوری بازی کنید اما دیدگاه روشنی از مشکلی که می‌خواهید حل کنید نداشته باشید، در بیابان فضای اَبَرمکعب[۹۵] (hyperspace) گم خواهید شد، الگوریتم‌های شما نیز همینطور.

من روی این نکته زیاد تاکید نمی‌کنم؛ احتمالاً روش خودتان را برای بررسی این مورد دارید. فقط مطمئن شوید که انجام بشود.

در متن بالا، عبارت «چه داده‌ای، چه مقیاسی، چه دقتی برای پیش‌بینی» به این معناست که قبل از شروع هرگونه پیش‌بینی، باید به طور دقیق مشخص کنید که:

- چه نوع داده‌ای برای پیش‌بینی مورد نیاز است؟ (چه داده‌ای)
- پیش‌بینی شما در چه مقیاسی انجام می‌شود؟ (چه مقیاسی)
- چه میزان دقت برای پیش‌بینی شما کافی است؟ (چقدر دقت)

بیایید هر کدام را با جزئیات بیشتری بررسی کنیم:

۱ .چه داده‌ای؟ (چه نوع داده‌ای مورد نیاز است؟)

این بخش به نوع داده‌هایی اشاره دارد که برای انجام پیش‌بینی خود به آن‌ها نیاز دارید. برای مثال:

- اگر می‌خواهید فروش یک محصول را پیش‌بینی کنید، به داده‌هایی مانند تاریخچه فروش، قیمت محصول، تبلیغات، فصلی بودن فروش و غیره نیاز دارید.
- اگر می‌خواهید ترافیک یک وب‌سایت را پیش‌بینی کنید، به داده‌هایی مانند بازدیدهای قبلی، منابع ترافیک، کلمات کلیدی و غیره نیاز دارید.
- اگر می‌خواهید آب و هوا را پیش‌بینی کنید، به داده‌هایی مانند دما، رطوبت، فشار هوا، سرعت باد و غیره نیاز دارید.

بنابراین، اولین قدم این است که مشخص کنید دقیقاً به چه نوع داده‌هایی برای پیش‌بینی مورد نظر خود نیاز دارید.

۲ .چه مقیاسی؟ (پیش‌بینی در چه مقیاسی انجام می‌شود؟)

این بخش به بازه زمانی یا سطح جزئیات پیش‌بینی اشاره دارد. برای مثال:

- بازه زمانی: آیا می‌خواهید فروش را برای روز بعد، هفته بعد، ماه بعد یا سال بعد پیش‌بینی کنید؟
- سطح جزئیات: آیا می‌خواهید فروش کل محصولات را پیش‌بینی کنید یا فروش هر محصول به صورت جداگانه؟ آیا می‌خواهید ترافیک کل وب‌سایت را پیش‌بینی کنید یا ترافیک هر صفحه به صورت جداگانه؟

مقیاس پیش‌بینی تأثیر زیادی بر نوع داده‌های مورد نیاز و روش‌های پیش‌بینی دارد.

۳ .چقدر دقت؟ (چه میزان دقت کافی است؟)

هیچ پیش‌بینی‌ای کاملاً دقیق نیست. میزان دقت مورد نیاز به کاربرد پیش‌بینی بستگی دارد. برای مثال:

- اگر می‌خواهید موجودی انبار را مدیریت کنید، دقت بالایی در پیش‌بینی فروش مورد نیاز است تا از کمبود یا انباشت کالا جلوگیری شود.
- اگر می‌خواهید روند کلی بازار را پیش‌بینی کنید، دقت کمتر کفایت می‌کند.
- تعیین میزان دقت مورد نیاز به شما کمک می‌کند تا روش‌های پیش‌بینی مناسب را انتخاب کنید و از صرف زمان و منابع برای دستیابی به دقتی که واقعاً لازم نیست، جلوگیری کنید.

۹۵- Hyperspace (فضای اَبَرمکعب): در ریاضیات و فیزیک به فضایی با بیش از سه بعد گفته می‌شود. در اینجا به صورت استعاری برای اشاره به پیچیدگی و سردرگمی ناشی از عدم تعریف دقیق مسئله استفاده شده است.

شماره ۲: باز هم، داده‌ی خوب < داده‌ی بزرگ

برخی از افراد ممکن است باور داشته باشند که کیفیت داده دیگر اهمیتی ندارد و تا زمانی که داده بزرگ وجود داشته باشد، خوب است. در حالی که این الگوریتم‌ها قطعاً روی مجموعه داده‌های بزرگتر عملکرد بهتری دارند، اما همچنان باید داده‌ی *خوب* وجود داشته باشد. یادگیری عمیق، یک فرآیند پاکسازی داده نیست. در واقع، بیشترین دستاوردها در یادگیری عمیق، به کمک یادگیری نظارت‌شده با داده‌های ساختاریافته حاصل شده است.

بسیاری از شرکت‌ها برای برآورده کردن بازگشت سرمایه (ROI) و سایر انتظارات ناملموس برای پروژه‌های داده‌ی بزرگ خود با مشکل مواجه شده‌اند. خواه کمبود دانشمندان داده باشد، خواه مشکلات واقعی حاکمیت بر دریاچه‌های داده[۹۶] یا پیچیدگی‌های مدیریت چندین فروشنده، مشاور و ارائه‌دهنده خدمات، مدیریت داده دشوار است.

یک دلیل مهم و منطقی وجود دارد که چرا یادگیری ماشین و داده‌ی بزرگ در یک رابطه‌ی بلندمدت بی‌نقص نیستند. در استفاده از داده‌های غیرایستا[۹۷] و مدل‌های تطبیقی[۹۸]، پیچیدگی‌های خاص و چالش‌های اضافی وجود دارد. هنگامی که مدل‌ها ساخته می‌شوند، تمایل دارند فقط روی داده‌هایی (دنیایی) که روی آن‌ها آموزش داده شده‌اند، به خوبی کار کنند. اگر دنیا تغییر کند، ممکن است آن‌ها از کار بیفتند و نتایج نادرستی ارائه دهند. اگر داده‌ای بزرگ و معتبر در روند تغییرات دنیا واقعی از بین برود [مثلا دیگر معتبر نباشد]، اساساً دوباره به داده‌ی کوچک تبدیل می‌شود و تشخیص اینکه آیا فقط در ارتباط با فراز و نشیب‌های دنیای عادی است یا دنیای تغییر کرده، بسیار دشوار است. در واقع، محققان و استارتاپ‌های زیادی به طور فعال روی این موضوع (که داده‌ی پراکنده نامیده می‌شود) کار می‌کنند، زیرا اقتصاد داده‌ی بزرگ و غیرایستا در یادگیری ماشین، پیشرفت را در برخی زمینه‌های مهم محدود خواهد کرد.

در هر پروژه یادگیری ماشین، مهم است که تشخیص دهیم بیشتر تلاش‌ها مصروف فرآیند بسیار غیرجذاب انتخاب، پاکسازی و مرتب‌سازی داده‌هاست. مطمئن شوید که داده‌ی بزرگ شما، داده‌ی بزرگ *خوبی* است.

شماره ۳: [هوش مصنوعی] بیشتر علم است تا نرم افزار

گروه تحقیقاتی یادگیری ماشین فیس‌بوک از واژه «آزمایش» استفاده می‌کند، و این انتخاب کاملاً آگاهانه است. یادگیری ماشین فرآیندی بر اساس آزمون و خطا، مهندسی ویژگی‌های خاص حوزه، و طراحی آزمایش است. این فرآیند از استانداردهای توسعه نرم‌افزار پیروی نمی‌کند و نتیجه‌ای واضح و مشخص در قالب منطق بولی، که معمولاً نمایانگر دیدگاه برنامه‌نویسان است، ارائه نمی‌دهد.

پروژه‌های یادگیری ماشین نیاز به تعریف دقیق‌تر مسئله نسبت به پروژه‌های نرم‌افزاری دارند، زیرا تأثیر پیش‌بینی و تعامل آن با محیط، خود مسئله را تغییر می‌دهد. این مسئله مشابه نظریه بازی‌ها در مقیاسی بزرگ‌تر است.

بین مدل‌های نرم‌افزاری قدیمی با برنامه‌نویس و منطق قابل ردیابی و کدگذاری جدید هوش مصنوعی شکاف‌هایی وجود دارد. در یادگیری ماشین، کنترل نسخه یا انتشار خاص یا تست وجود ندارد. با نفوذ یادگیری ماشین به سراسر سیستم‌ها، شرکت‌ها باید دریابند که چگونه چیزی را اشکال‌زدایی کنند که مدولار نیست، دائما در حال تغییر است و ممکن است ناخواسته سوگیری‌هایی را بر اساس ماهیت الگوریتم‌های آماری و داده‌های آموزشی در خود جای بدهد.

یادگیری ماشین مرتب، مرحله‌بندی‌شده یا قطعی نیست. این یک فرآیند شامل نمونه‌سازی (پروتوتایپ)، ارزیابی، آزمایش و مقیاس‌دهی است. همان‌طور که شرکت‌های داروسازی نمی‌توانند تمام منابع خود را در مرحله پیش‌آزمایش بالینی مصرف کنند، شما هم نمی‌توانید تمام بودجه خود را برای اولین نسخه صرف کنید. انجام آزمایش‌های متعدد ضروری است و هزینه‌های نگهداری همچنان یک عامل ناشناخته باقی می‌ماند. بسیاری از مسائل باید در قالب‌هایی غیر از قطعی بازتعریف شوند، مثلاً با تعیین میزان پذیرش برای دقت یا نرخ خطای مثبت/منفی کاذب.

۹۶- Data lakes (دریاچه‌های داده): مخازن بزرگی از داده‌های خام در قالب‌های مختلف.

۹۷- Non-stationary data (داده‌های غیرایستا): داده‌هایی که توزیع آماری آن‌ها با گذشت زمان تغییر می‌کند.

۹۸- Adaptive models (مدل‌های تطبیقی): مدل‌هایی که می‌توانند با تغییر داده‌ها، خود را به‌روزرسانی کنند.

این دنیای پر از عدم قطعیت برای شرکت‌هایی که به تفکر قطعی عادت دارند، چندان سازگار نیست. برای موفقیت در این حوزه، باید با تفکر در قالب احتمالات راحت باشید.

شماره ٤: [هوش مصنوعی] بیشتر [بر اساس نوعی] خِرد است تا یک کدنویسیِ سرسری[۹۹] (Hack-dom)

دلیلی وجود دارد که موی سر بسیاری از چهره‌های سرشناس در هوش مصنوعی، خاکستری شده است. مهارت[۱۰۰] در یادگیری ماشین سال‌ها زمان و معمولاً یک یا دو مدرک دکترا می‌طلبد. طی بیست تا سی سال گذشته، زمانی که پایه‌های یادگیری ماشین امروزی در ریاضیات کاربردی و هوش مصنوعی آماری بنا نهاده شد، افرادی که در این حوزه ماندگار شدند فرصت داشتند تا روش‌های مختلفی را بیازمایند. امروز که نسل جدیدی از دانشمندان داده و مهندسان در حال ساخت هوش مصنوعی مدرن هستند، تجربیات پیشین نقشی حیاتی و فعال دارند. هوش مصنوعی مدرن به شدت بر اصول و پایه‌هایی تکیه دارد که تاریخچه‌ای طولانی در علوم کامپیوتر دارند.

بازیگران اصلی در حوزه هوش مصنوعی و یادگیری ماشین، مانند فیسبوک، گوگل، بایدو، مایکروسافت، IBM، اوبر و تویوتا، همگی گروه‌های تحقیقاتی دارند، زیرا تحقیق و توسعه (R&D) یک استراتژی سرمایه‌گذاری مهم است. به عنوان نمونه، فیسبوک سیستم ترجمه زبان طبیعی خود را توسعه داده است. زبان کاربران فیسبوک به دلیل اصطلاحات عامیانه[۱۰۱]، اختصارات و محلی‌سازی‌های گسترده، بسیار متفاوت است. همچنین، مأموریت OpenAI ایجاد هوش مصنوعی ایمن و قابل دسترس برای همه است.

ابزارهای فناوری اطلاعات مدرن خود را به ذهنیت کدزنیِ سرسری قرض داده‌اند. در شرکت‌ها، هکاتون‌ها و توسعه‌های مخفیانه (Skunk work) تقریباً رکود معماری فناوری اطلاعات نصب شده را مسخره می‌کنند. بخشی از این احساس در یادگیری ماشین نیز وجود دارد. TensorFlow گوگل و یادگیری ماشین AWS و همچنین انبوهی از کتابخانه‌های متن‌باز یادگیری ماشین ممکن است این تصور را ایجاد کنند که این صرفاً یک افزونه‌ی آسان برای تسلط یک مهندس نرم‌افزار است. حتی بهتر، رایگان است. اما در واقع هیچ چیز، رایگان نیست.

فرآیند یادگیری ماشین یک فرآیند علمی است:

- بر پایه فرضیه‌سازی،
- طراحی آزمایش‌های مستحکم،
- درک عمیق از اصول پایه،
- تسلط بر پیچیدگی‌های تحلیل‌های چندبعدی آماری،
- و مهارت ذاتی در آزمودن خلاقانه نتایج در برابر گروه کنترل.

پیش‌بینی زمان و هزینه یک پروژه یادگیری ماشین دشوار است. همان‌طور که در علم، رویکرد مبتنی بر آزمایش، فرضیه‌سازی و اثبات مفهومی (Proof-of-Concept) بهترین راه برای شروع است، در یادگیری ماشین نیز چنین رویکردی ضروری است.

شماره ۵: هوش مصنوعی تمایل به دوقطبی[۱۰۲] کردن مردم دارد؛ پیام [شما] مهم است

تکنولوژی همیشه نیروی کار را دگرگون کرده است. هر کسی که با او صحبت می‌کنید، نظری درباره ربات‌ها و تأثیر آن‌ها بر مشاغل یا آینده زندگی حرفه‌ای دارد. زمانی که پژوهشگران دانشگاه آکسفورد، فری و آزبورن، اعلام کردند که ٤٧٪ از مشاغل در ایالات متحده در معرض تهدید تکنولوژی قرار دارند، بسیاری از افراد نگران شدند. اقتصاددانان شرکت Deloitte نیز تغییرات شغلی ناشی از تکنولوژی در بریتانیا را طی ۱۵۰ سال گذشته بررسی کردند و نتیجه گرفتند که بحث درباره تکنولوژی بیشتر بر

۹۹- Hack-dom/Hack mentality (کدنویسی سرسری/ذهنیت کدنویسی سرسری): اشاره به رویکردی سطحی و سریع در کدنویسی و توسعه‌ی نرم‌افزار دارد که اغلب بدون در نظر گرفتن اصول مهندسی نرم‌افزار و با هدف دستیابی سریع به نتیجه انجام می‌شود.

۱۰۰- Artisanry (مهارت): مهارت و استادی در انجام یک کار، به خصوص کارهای دستی. در اینجا به مهارت عمیق در یادگیری ماشین اشاره دارد.

۱۰۱- Colloquialisms (اصطلاحات عامیانه): کلمات و عباراتی که در گفتار روزمره استفاده می‌شوند و معمولاً در نوشتار رسمی دیده نمی‌شوند.

۱۰۲- Polarize (دوقطبی کردن): تقسیم کردن افراد به دو گروه کاملاً مخالف

جنبه‌های تخریبی[۱۰۳] آن متمرکز است. آن‌ها دریافتند که تکنولوژی اثرات خلاقانه زیادی دارد، اما این اثرات به دلیل ماهیتشان سخت‌تر تعریف و پیش‌بینی می‌شوند.

مشکل برای افراد این است که تکنولوژی بسیار سریع‌تر از انسان‌ها پیشرفت می‌کند. ارتباط مستقیمی بین جایگاه شغلی و اشتیاق به هوش مصنوعی وجود دارد. یک مطالعه نشان داده است که این شکاف چقدر عمیق است: حدود ۴۲% از مدیران ارشد از هوش مصنوعی هیجان‌زده‌اند، در حالی که تنها ۱۵% از کارکنان خط مقدم[۱۰۴] چنین احساسی دارند.

در تکنولوژی نوعی عدم تقارن پیش‌بینی[۱۰۵] وجود دارد. ما نمی‌توانیم به سادگی آنچه را که خلق خواهد شد تصور کنیم، اما می‌توانیم به وضوح آنچه را که از بین خواهد رفت شناسایی کنیم. علاوه بر این، ما بیشتر از آنکه به دستاوردها اشتیاق داشته باشیم، از زیان‌ها می‌ترسیم. اگر مردم باور داشته باشند که هوش مصنوعی به زیان آن‌ها منجر می‌شود، احتمال بیشتری وجود دارد که از آن بترسند و نسبت به کسانی که از آن استفاده می‌کنند احساس نارضایتی کنند.

شما نمی‌توانید همه دستاوردها و زیان‌های ناشی از هوش مصنوعی را پیش‌بینی کنید، اما می‌توانید در نحوه صحبت درباره آن‌ها آگاهانه و مدبرانه عمل کنید.

شماره ٦: ارزش ها و شفافیت، سوگیری و اخلاق

کلمه «الگوریتم» در حال تبدیل شدن به یک مفهوم مرموز در ذهن جمعی ماست؛ موجودیتی که خارج از کنترل ما عمل می‌کند و توسط افرادی مریض و مطرود، برای دستکاری ما استفاده می‌شود. چه در مورد نحوه نمایش اخبار در فیسبوک و چه در مورد نتایج جستجوی گوگل، الگوریتم‌ها به‌نوعی به عنوان موجوداتی با ویژگی‌های اخلاقی، به خودی خود، دیده می‌شوند.

تا اینجا شما می‌دانید که الگوریتم‌ها در واقع مجموعه‌ای از قوانین یا روتین‌ها هستند که مدتی است آن‌ها را داریم. قبل از ظهور داده‌های بزرگ و الگوریتم‌های یادگیری ماشین، الگوریتم‌ها توسط برنامه‌نویسان به‌طور دستی کدنویسی می‌شدند و پیش‌داوری‌ها از طریق سیاست‌ها مدیریت می‌شدند. سیاست‌ها بر اساس آمار تنظیم می‌شدند. مردم تصمیماتی درباره سیاست‌ها می‌گرفتند که مسائلی مانند تأیید وام‌ها را تعیین می‌کرد. در جداول اکتوئری[۱۰۶] هر شرکت بیمه، داده‌هایی وجود دارد که نشان می‌دهد آیا مردان یا زنان رانندگان ایمن‌تری هستند یا اینکه چه تفاوتی باید بین حق بیمه‌های افراد با درآمد کم و بالای یک خانوار باشد. در آن زمان، ما به الگوریتم‌ها به عنوان موجوداتی با پیش‌داوری نگاه نمی‌کردیم. بلکه از آمار، ریسک و سیاست‌ها یا قوانین مربوط به رفتار منصفانه یا جلوگیری از تبعیض صحبت می‌کردیم.

ما می‌توانیم از یادگیری ماشین برای تولید دانش جدید از مقدار نسبتاً کمی از دانش قبلی استفاده کنیم. این امر باعث می‌شود که این الگوریتم‌ها موتورهای بسیار قدرتمندی برای کشف مسائل جدید شوند. اما این می‌تواند به انتشار هرگونه پیش‌داوری نیز منجر شود زمانی که این الگوریتم‌ها گسترش یابند و در مسائل مختلف، کامپیوترها یا شبکه‌های اجتماعی مورد استفاده قرار گیرند.

پس با گسترش الگوریتم‌های یادگیری ماشین، باید نگران چه منابع جدیدی از بی‌عدالتی باشیم؟

- نابرابری

اقتصادهای توسعه‌یافته، با زیرساخت‌های پیشرفته تکنولوژی و سیستم‌های داده‌ای خوب‌توسعه‌یافته، قادر خواهند بود که از مزایای یادگیری ماشین و هوش مصنوعی بسیار سریع‌تر و با اثرات بیشتری نسبت به کشورهای در حال توسعه بهره‌برداری کنند. کشورهای ثروتمند انبوهی از داده‌های ساختار یافته دیجیتال در اختیار دارند. از طرف دیگر، با پیشرفت‌هایی که در

۱۰۳- Disrupt (مختل کردن): ایجاد اختلال و تغییرات اساسی در یک سیستم یا فرآیند.

۱۰٤- Front line workers (کارکنان خط مقدم): کارکنانی که مستقیماً با مشتریان یا عملیات اصلی شرکت در ارتباط هستند.

۱۰۵- Prediction asymmetry (عدم تقارن در پیش‌بینی): ناتوانی در پیش‌بینی نتایج مثبت فناوری به همان اندازه که می‌توانیم نتایج منفی آن را پیش‌بینی کنیم.

۱۰٦- Actuarial tables (جداول محاسباتی): جداولی که توسط شرکت‌های بیمه برای محاسبه ریسک و حق بیمه استفاده می‌شوند.

پردازش داده‌های غیرساختار یافته، بینایی کامپیوتری و پردازش زبان طبیعی به وجود آمده است، ممکن است کشورهای در حال توسعه فرصتی پیدا کنند که از بسیاری از زیرساخت‌های داده‌ای سنتی و پرهزینه عبور کنند (به اصطلاح "پرش از پله‌ها"). اما نمی‌توان نادیده گرفت که اقتصادهای توسعه‌یافته از مزیت شروع زودتر برخوردارند و این خطر وجود دارد که نابرابری اقتصادی میان کشورهای مختلف بیشتر شود.

- تبعیض

برچسب‌گذاری توهین‌آمیز گوگل برای سیاه‌پوستان به عنوان گوریل؟

تبعیض فروشگاه Staples علیه افراد فقیر؟

استریوتایپ‌های جنسیتی در نتایج جستجو بر اساس شغل جستجو شده؟

تبلیغات آنلاین تبعیض‌آمیز که کارهای کم‌درآمد را با زنان مرتبط می‌کند؟

تمام این‌ها اتفاق افتاده‌اند.

برنامه‌نویسان به‌طور معمول داده‌های کاربران را در الگوریتم‌های پیچیده، روش‌های ابتکاری[۱۰۷] و برنامه‌های کاربردی وارد می‌کنند. بیشتر اوقات آنچه که به دست می‌آید مفید است و اطلاعات دقیق‌تری به ما می‌دهد. اما ممکن است عواقب ناخواسته‌ای مانند تبعیض به همراه داشته باشد، به‌ویژه اگر داده‌هایی که برای آموزش الگوریتم استفاده شده‌اند شامل اطلاعات کافی از اقلیت‌ها نباشند (بازنمایی[۱۰۸] کافی نداشته باشد).

باید این ارتباطات نادرست را مانند بلاگ‌ها (اشکال نرم‌افزاری) در نظر بگیریم. و مانند آنچه که قبلاً در سیاست‌ها انجام می‌دادیم - چک کردن صحت آن - با استفاده از روش‌های جدید دیباگ کردن نرم‌افزارها. تبعیض در این زمینه، سوگیری ماشین است. و سوگیری ماشین یک اشکال [Bug] است.

- ضعف شفافیت

درست است که ما دقیقاً نمی‌دانیم چگونه یادگیری ماشین کار می‌کند، اما آن کار می‌کند. فقط لحظه‌ای به این فکر کنید: وقتی خروجی یک الگوریتم دیگر قابل پیگیری[۱۰۹] نباشد، وقتی پاسخ در احتمال‌های پیچیده و چندبعدی[۱۱۰] نهفته باشد، و فردی که مسئول ست نتواند توضیح دهد چرا تصمیم خاصی گرفته شده، باید چه فکری بکنیم؟

بسیاری از آنچه که ما به عنوان انصاف در نظر می‌گیریم، به همان اندازه که به نمایش واضح حق و باطل [و درست و غلط] بستگی دارد، به «اعتماد» نیز وابسته است. تصمیمات و اقدامات همیشه سیاه و سفید نیستند. حس عدالت یا تصمیم‌گیری منصفانه ما به همان اندازه به نحوه رفتار با ما و اینکه آیا کسی به طور صادقانه عمل کرده است، بستگی دارد که به یک مرز قطعی و مشخص. اگر مردم به یک الگوریتم رجوع کنند، شانه‌هایشان را بالا بیندازند و بگویند «خب من هم آن را نمی‌فهمم»، اعتماد به شدت آسیب خواهد دید.

- اخلاق

هر مسئله اخلاقی پیچیده‌ای را که نام ببرید، احتمالاً هیچ راه‌حل شناخته‌شده‌ای برای آن وجود ندارد، چه برسد به اینکه راه‌حل ساده‌ای هم داشته باشد. قانون یک کد مبهم از اخلاقیات جامعه است که بر اساس زمینه و پرونده‌های قبلی شکل می‌گیرد. وقتی با یک معضل مواجه می‌شویم، شهود ما نمی‌تواند به ما کمک کند. ما می‌توانیم دو دیدگاه متناقض را در ذهن خود نگه داریم، اما یک ماشین نمی‌تواند چنین کاری بکند.

۱۰۷- روش‌های ابتکاری (Heuristics): روش‌هایی برای حل مسئله که لزوماً به یافتن بهترین جواب تضمین نمی‌دهند، اما در بیشتر موارد جواب‌های قابل قبولی ارائه می‌دهند و سریع‌تر از روش‌های دقیق عمل می‌کنند.

۱۰۸- بازنمایی کافی (Underrepresented): به این معنی است که داده‌های مربوط به یک گروه خاص (مانند اقلیت‌ها) به اندازه کافی در مجموعه داده آموزشی الگوریتم وجود ندارد، که می‌تواند منجر به سوگیری الگوریتم شود.

۱۰۹- قابل ردیابی (Traceable): به این معنی است که می‌توان مراحل و منطق تصمیم‌گیری یک الگوریتم را به طور واضح دنبال کرد و فهمید که چگونه به یک خروجی خاص رسیده است.

۱۱۰- احتمالات پیچیده و چندبعدی (Complex, Multidimensional Probability): در الگوریتم‌های پیچیده یادگیری ماشین، به ویژه شبکه‌های عصبی عمیق، تصمیم‌گیری‌ها بر اساس احتمالات چندگانه و روابط پیچیده بین متغیرها انجام می‌شود که درک و تفسیر آنها دشوار است.

مسائل اخلاقی پنهانی نیز وجود دارند که بر زندگی روزمره ما تأثیر می‌گذارند. در طراحی هر الگوریتم، به‌طور ضروری تعداد معینی از مثبت‌های کاذب و منفی‌های کاذب وجود دارد. طراح الگوریتم باید بین این دو تعادل برقرار کند. اگر این الگوریتم برای شناسایی تصاویری از گربه‌ها باشد، این تعادل ممکن است چندان حیاتی نباشد، اما اگر الگوریتم مربوط به تصویربرداری پزشکی برای شناسایی یک وضعیت خطرناک باشد، طراحی این تعادل پیامدهای عمیقی خواهد داشت. چنین الگوریتمی مسائل اخلاقی را برای پزشک تشخیص‌دهنده در بر دارد. طراح الگوریتم، نه پزشک، تصمیم می‌گیرد که خطر از دست دادن تشخیص سرطان در تعداد کمی از موارد را در مقابل خطر هشداردادن غیرضروری به گروهی از افراد که در واقع بیمار نیستند، چگونه مدیریت کند.

- شخصی‌سازی

هدف اصلی شخصی‌سازی[۱۱۱]، ایجاد تجربه‌ای است که کاملاً متناسب با وضعیت فردی شما باشد، خواه در پزشکی، تجارت الکترونیک یا در یک تجربه دنیای واقعی. معضل، در حریم خصوصی و امنیت و در ایجاد تجربه‌ای است که مورد اعتماد باشد. شخصی‌سازی بیش از حد می‌تواند به شدت ناخوشایند باشد. طراحی هوش مصنوعی واقعاً انسان‌محور[۱۱۲]، دشوارتر از آن چیزی است که به نظر می‌رسد و خارج از داستان‌های علمی تخیلی، به ندرت به آن دست یافته‌ایم.

به طور خلاصه، بسیاری از چیزهایی که توسط هوش مصنوعی متحول می‌شوند، بدون تغییر باقی می‌مانند. آنها همچنان طبق قوانینی که از قبل وجود داشته‌اند، عمل می‌کنند. اما به طور ویژه به مکان‌های جدیدی که قوانین باید در آنها اعمال شوند توجه کنید؛ نحوه جمع‌آوری و استفاده از داده‌ها، چه نسبتی از نتایج نادرست یا غلط توسط الگوریتم‌ها تولید می‌شود و چه کسی آن نتایج را می‌بیند، نگرش مردم نسبت به دریافت تصمیمات تولید شده خودکار، نحوه واکنش مشتریان به شخصی‌سازی دقیق. اینها همه حوزه‌های جدید نظارت و حکمرانی[۱۱۳] در عصر هوش ماشینی هستند.

خلاصه‌ای از مطالب...

مسئله خود را بشناسید.

داده‌های خود، داده‌های جدید مورد نیازتان و هزینه به دست آوردن آنها را بشناسید.

موفقیت را به صورت آماری تعریف کنید و با کمک احتمالات و قوانین و نظریه‌هایش فکر کنید.

رویکردی تجربی اتخاذ کنید، با یک فرضیه و اثبات مفهوم[۱۱۴] شروع کنید.

بر قابلیت‌های جدیدی که به آنها نیاز خواهید داشت تمرکز کنید و گفتگویی حول آنها شکل دهید.

۱۱۱- شخصی‌سازی (Personalization): تطبیق محصولات، خدمات یا تجربیات با نیازها و ترجیحات فردی کاربران.

۱۱۲- انسان‌محور (Human-centric): رویکردی در طراحی که بر نیازها، توانایی‌ها و محدودیت‌های انسان تمرکز دارد.

۱۱۳- حکمرانی (Governance): مجموعه‌ای از قوانین، سیاست‌ها و رویه‌هایی که برای مدیریت و کنترل یک سیستم یا سازمان استفاده می‌شوند.

۱۱۴- اثبات مفهوم (Proof of Concept): آزمایشی اولیه و کوچک‌مقیاس برای نشان دادن عملی بودن یک ایده یا فناوری.

کلام آخر

هوش مصنوعی برق جدید است.

- اندرو نگ، دانشگاه استنفورد

یادگیری ماشین به اندازه اختراع برق یک پیشرفت بزرگ است. یادگیری ماشین فناوری بنیادینی است که در تمام زمینه‌هایی که با آن‌ها در ارتباطیم نفوذ خواهد کرد. کامپیوترها دیگر از طریق برنامه‌نویسی یاد نمی‌گیرند، بلکه از طریق تجربه یاد می‌گیرند. معامله سهام شما، شناسایی دارویی که بهترین اثر را برای شما دارد، رانندگی شما به محل کار، نوشتن مقالات خبری شخصی، پیش‌بینی خدماتی که به آن‌ها نیاز دارید و زمان آن‌ها، ارائه مجموعه‌ای از محصولات انتخاب شده؛ همه این‌ها به‌طور فزاینده‌ای سریع‌تر و دقیق‌تر انجام می‌شوند به دلیل یادگیری ماشین.

حالا شما مکانیزم‌های ریاضی و دامنه تصمیمات مهندسی را که در این الگوریتم‌ها به کار رفته می‌فهمید. اکنون می‌بینید که این الگوریتم‌ها هنوز آماده رسیدن به نصاب هوش عمومی در سطح انسان نیستند و برای رسیدن به این هدف به پیشرفت‌های بسیاری در علوم کامپیوتر، علوم اجتماعی، علوم اعصاب و سایر رشته‌های فناوری نیاز دارند. به این فکر کنید که چگونه یک کودک یاد می‌گیرد یک گربه را شناسایی کند. با کمتر از دوازده تصویر، یک کودک نه تنها قادر است گربه را شناسایی کند، بلکه می‌تواند این دانش را تعمیم دهد و گربه را از زاویه‌های مختلف یا در نور ضعیف تشخیص دهد. امروز حتی بهترین سیستم‌های یادگیری عمیق نیاز به تصاویری با مقیاس‌های بزرگتر دارند. یا به این نکته توجه کنید که در حالی که هر یک از این الگوریتم‌ها می‌توانند بسیار قدرتمند باشند، محققان هنوز در تلاش‌اند تا آن‌ها را به گونه‌ای ترکیب کنند که مزایای هرکدام را در یک الگوریتم به حداکثر برسانند.

یادگیری ماشین هنوز نتواسته است مشکلات دشوار «عقل سلیم[۱۱۵]» را حل کند؛ دانش پیش‌زمینه‌ای که ما از طریق تجربیات‌مان از جهان به دست می‌آوریم و توانایی درک زمینه و به تبع آن، استدلال. توانایی تشخیص این که یک توالت و یک صندلی، اگرچه از نظر ظاهری مشابه‌اند، در بیشتر مواقع کاربردهای کاملاً متفاوتی دارند. اما این وضعیت به سرعت در حال تغییر است زیرا یادگیری ماشین در حال تسریع کشف راه‌های جدید برای حل چالش‌های بزرگ هوش مصنوعی است.

یادگیری ماشین در یک نقطه هیجان‌انگیز قرار دارد. حالا زمان آن است که یاد بگیرید این فناوری چیست (و در نتیجه چه چیزی نیست) تا بتوانید با چشمی نقاد به تبلیغات آن نگاه کنید، سوالات درست را بپرسید و از پتانسیل آن برای بروز سوگیری[۱۱۶] آگاه باشید. حالا زمان آن است که یاد بگیرید ماشین‌ها چگونه یاد می‌گیرند تا بتوانید تأثیر آن را بر زندگی روزمره‌تان درک کنید و در آینده، قادر باشید پیش‌بینی کنید که چگونه این فناوری ارتباطات شخصی ما، آموزش، مشاغل و تجربه مشترک خلاقیت انسانی را تغییر خواهد داد.

۱۱۵- عقل سلیم (Common Sense): درک پایه‌ای از جهان که بیشتر انسان‌ها به طور طبیعی دارند و شامل دانش ضمنی در مورد نحوه عملکرد اشیا و تعاملات اجتماعی است.

۱۱۶- سوگیری (Bias): تمایل سیستم به ارائه نتایج خاص به دلیل نقص در داده‌ها یا طراحی الگوریتم

تقدیر و تشکر

مؤسسه تحقیقات هوش مصنوعی Intelligentsia Research از موارد زیر قدردانی می‌کند:

- از «غول‌های» هوش مصنوعی که آثارشان هم به ما آگاهی بخشیده و هم در دقت و دسترس‌پذیری این راهنمای اولیه مؤثر بوده است.
- از Celeste Knudsen برای تصویرسازی ماهرانه و جذاب مفاهیم پیچیده یادگیری ماشین.
- از Pedro Domingos، نویسنده کتاب "The Master Algorithm" که به شدت توصیه می‌کنیم آن را مطالعه کنید. توسعه ساختار "Five Tribes" باعث شده است یادگیری ماشین برای همه بسیار قابل دسترس‌تر شود.
- از BigML به‌ویژه برای ایده‌هایشان در زمینه مهندسی ویژگی‌ها.
- از Jason در MachineLearningMastery برای کمک به درک اصول اولیه.
- از Zachary Lipton در UCSD برای توضیح چگونگی عملکرد زمان در شبکه‌های عصبی.
- از Bob Van den Hoek برای پاسخ‌هایش در Quora.
- از Norm Jouppi، گوگل.
- از Alex Gray، Skytree.
- از Alex Gray، مدیر تحقیقات هوش مصنوعی در فیسبوک.
- از Kalid Azad، BetterExplained، برای شاید بهترین توضیحی که تا به حال از بیز دیدیم (و ما بسیاری را مطالعه کرده‌ایم).
- از Ivan Vasilev برای flu.
- و از تمامی افرادی که با آن‌ها در کنفرانس‌ها و از طریق تماس‌های تلفنی صحبت کرده‌ایم و در تلاشند تا یادگیری ماشین را به نقطه‌ای عالی برسانند.