**آموزش یادگیری ماشین**

این آموزش به صورت ساختار یافته و گام‌به‌گام، مفاهیم یادگیری ماشین را توضیح می‌دهد. محتوای این آموزش ترجمه کامل و دقیق سند ارائه‌شده است و به گونه‌ای سازماندهی شده که مانند یک راهنمای آموزشی عمل کند. هر بخش با عنوان‌های اصلی و فرعی، توضیحات، مثال‌ها و نکات کلیدی همراه است تا یادگیری آسان‌تر شود. اصطلاحات فنی به فارسی ترجمه شده‌اند، اما در پرانتز معادل انگلیسی آن‌ها برای وضوح ذکر شده است.

**بخش ۱: یادگیری ماشین چیست؟**

یادگیری ماشین (Machine Learning) زیرمجموعه‌ای از هوش مصنوعی (AI) است که بر الگوریتم‌هایی تمرکز دارد که می‌توانند الگوهای داده‌های آموزشی را “یاد بگیرند” و سپس، استنتاج‌های دقیق درباره داده‌های جدید انجام دهند. این توانایی تشخیص الگو، مدل‌های یادگیری ماشین را قادر می‌سازد تا تصمیم‌گیری یا پیش‌بینی کنند بدون نیاز به دستورالعمل‌های صریح و کدگذاری‌شده.

یادگیری ماشین امروزه بر حوزه هوش مصنوعی تسلط یافته است: این فناوری ستون فقرات بیشتر سیستم‌های هوش مصنوعی مدرن را تشکیل می‌دهد، از مدل‌های پیش‌بینی تا وسایل نقلیه خودران، مدل‌های زبانی بزرگ (LLMs) و سایر ابزارهای هوش مصنوعی تولیدی.

اصل مرکزی یادگیری ماشین (ML) این است که اگر عملکرد یک مدل را روی مجموعه‌ای از وظایف بهینه کنید که به اندازه کافی شبیه مشکلات واقعی جهانی باشد—از طریق فرآیندی به نام آموزش مدل (model training)—مدل می‌تواند پیش‌بینی‌های دقیق روی داده‌های جدیدی که در کاربرد نهایی خود می‌بیند، انجام دهد.

آموزش خود به عنوان وسیله‌ای برای رسیدن به هدف است: تعمیم‌پذیری (generalization)، که ترجمه عملکرد قوی روی داده‌های آموزشی به نتایج مفید در سناریوهای واقعی است، هدف اساسی یادگیری ماشین است. در اصل، یک مدل آموزش‌دیده الگوهایی را که از داده‌های آموزشی یاد گرفته، اعمال می‌کند تا خروجی صحیح برای یک وظیفه واقعی را استنتاج کند: بنابراین، استقرار یک مدل هوش مصنوعی را استنتاج هوش مصنوعی (AI inference) می‌نامند.

یادگیری عمیق (Deep Learning)، زیرمجموعه‌ای از یادگیری ماشین که توسط شبکه‌های عصبی مصنوعی بزرگ—یا بهتر بگوییم “عمیق”— هدایت می‌شود، در چند دهه اخیر به عنوان معماری مدل هوش مصنوعی پیشرفته در تقریباً هر حوزه‌ای که هوش مصنوعی استفاده می‌شود، ظاهر شده است. در مقابل الگوریتم‌های صریحاً تعریف‌شده یادگیری ماشین سنتی، یادگیری عمیق بر شبکه‌های توزیع‌شده عملیات ریاضی تکیه دارد که توانایی بی‌نظیری برای یادگیری ظرافت‌های پیچیده داده‌های بسیار پیچیده فراهم می‌کند. از آنجایی که یادگیری عمیق به مقادیر بسیار زیادی داده و منابع محاسباتی نیاز دارد، ظهور آن با اهمیت فزاینده “داده‌های بزرگ” (big data) و واحدهای پردازش گرافیکی (GPUs) همزمان بوده است.

رشته یادگیری ماشین با علم داده (data science) ارتباط نزدیکی دارد. در واقع، یادگیری ماشین را می‌توان به عنوان مجموعه‌ای از الگوریتم‌ها و تکنیک‌ها برای خودکارسازی تحلیل داده و (مهم‌تر از آن) اعمال یادگیری‌های حاصل از آن تحلیل برای اجرای خودکار وظایف مرتبط درک کرد.

منشأ اصطلاح (هرچند نه مفهوم اصلی آن) اغلب به مقاله آرتور ال. ساموئل در سال ۱۹۵۹ در مجله IBM، با عنوان “برخی مطالعات در یادگیری ماشین با استفاده از بازی چکرز” نسبت داده می‌شود. در مقدمه مقاله، ساموئل به طور خلاصه نتیجه ایده‌آل یادگیری ماشین را بیان می‌کند: "یک کامپیوتر می‌تواند برنامه‌ریزی شود تا یاد بگیرد بازی چکرز را بهتر از کسی که برنامه را نوشته، بازی کند."¹

در این تصویر یک جدول دیده می‌شود که چند الگوریتم پرکاربرد یادگیری ماشین را همراه با توضیح، تابع خطا (Loss Function) و نوع کاربر (Classification یا Regression) معرفی کرده.

ترجمه و بازنویسی فارسی آن به شکل جدول به این صورت است:

|  |  |  |  |
| --- | --- | --- | --- |
| توضیح | تابع خطا (Loss Function) | نوع کاربرد | نام الگوریتم |
| پیش‌بینی خروجی پیوسته با برازش یک رابطه خطی بین ورودی‌ها و خروجی‌ها. | MSE (میانگین مربعات خطا – Mean Squared Error) | رگرسیون | **رگرسیون خطی (Linear Regression)** |
| مدل‌سازی احتمال یک خروجی دو مقداری (باینری) با استفاده از تابع لجستیک (سیگموید). | آنتروپی متقاطع باینری (Binary Cross-Entropy یا Log Loss) | دسته‌بندی | **رگرسیون لجستیک (Logistic Regression)** |
| تقسیم داده به شاخه‌ها بر اساس مقادیر ویژگی‌ها به منظور پیش‌بینی. | معیارهای ناخالصی (Impurity Measures) مانند جینی (Gini)، آنتروپی (Entropy)، یا MSE | هر دو (دسته‌بندی و رگرسیون) | **درخت تصمیم (Decision Tree)** |
| مجموعه‌ای از درخت‌های تصمیم که نتایجشان میانگین‌گیری (در رگرسیون) یا رأی‌گیری (در دسته‌بندی) می‌شود. | مشابه تابع خطای درخت تصمیم | هر دو | **جنگل تصادفی (Random Forest)** |
| چارچوب گرادیان بوستینگ که درخت‌ها را به‌صورت ترتیبی برای اصلاح خطاهای قبلی می‌سازد. | قابل سفارشی‌سازی؛ اغلب Log Loss یا MSE | هر دو | **ایکس‌جی‌بست (XGBoost)** |
| یافتن ابرصفحه بهینه که کلاس‌ها را جدا یا داده‌ها را برازش کند. | Hinge Loss (برای دسته‌بندی) یا Epsilon-insensitive Loss (برای رگرسیون) | هر دو | **ماشین بردار پشتیبان (SVM – Support Vector Machine)** |

**بخش ۲: یادگیری ماشین در مقابل هوش مصنوعی**

هرچند “یادگیری ماشین” و “هوش مصنوعی” اغلب به جای یکدیگر استفاده می‌شوند، اما کاملاً مترادف نیستند. به طور خلاصه: همه یادگیری ماشین هوش مصنوعی است، اما همه هوش مصنوعی یادگیری ماشین نیست.

در تصور عمومی، “هوش مصنوعی” معمولاً با داستان‌های علمی-تخیلی مرتبط است—معمولاً از طریق توصیفاتی از آنچه به درستی هوش مصنوعی عمومی (AGI) نامیده می‌شود، مانند HAL 9000 در اودیسه فضایی یا آوا در اکس ماشینا—یا اخیراً، با هوش مصنوعی تولیدی. اما “هوش مصنوعی” یک اصطلاح کلی برای هر برنامه‌ای است که می‌تواند از اطلاعات برای تصمیم‌گیری یا پیش‌بینی بدون دخالت فعال انسانی استفاده کند.

ساده‌ترین سیستم‌های هوش مصنوعی مجموعه‌ای از دستورات اگر-پس-درغیراینصورت هستند، با قوانین و منطقی که صریحاً توسط یک دانشمند داده برنامه‌ریزی شده است. در سطح ساده‌ترین، حتی یک ترموستات ابتدایی یک سیستم هوش مصنوعی مبتنی بر قوانین است: وقتی با قوانین ساده‌ای مانند اگر دمای اتاق < ۶۷، سپس بخاری را روشن کن و اگر دمای اتاق > ۷۲، سپس کولر را روشن کن برنامه‌ریزی شود، ترموستات قادر به تصمیم‌گیری خودکار بدون دخالت بیشتر انسانی است. در سطح پیچیده‌تر، یک درخت تصمیم‌گیری بزرگ و پیچیده مبتنی بر قوانین که توسط متخصصان پزشکی برنامه‌ریزی شده، می‌تواند علائم، شرایط و بیماری‌های همراه را تجزیه و تحلیل کند تا به تشخیص یا پیش‌آگهی کمک کند.²

برخلاف سیستم‌های متخصص، منطقی که یک مدل یادگیری ماشین بر اساس آن عمل می‌کند، صریحاً برنامه‌ریزی نمی‌شود—بلکه از طریق تجربه یاد گرفته می‌شود. یک برنامه فیلتر اسپم ایمیل را در نظر بگیرید: هوش مصنوعی مبتنی بر قوانین نیاز به یک دانشمند داده دارد تا معیارهای دقیق و جهانی برای اسپم را به صورت دستی طراحی کند؛ یادگیری ماشین تنها نیاز به انتخاب یک الگوریتم مناسب و یک مجموعه داده کافی از ایمیل‌های نمونه دارد. در آموزش، مدل ایمیل‌های نمونه را می‌بیند و پیش‌بینی می‌کند کدام‌ها اسپم هستند؛ خطای پیش‌بینی‌های آن محاسبه می‌شود و الگوریتم آن تنظیم می‌شود تا خطا را کاهش دهد؛ این فرآیند تکرار می‌شود تا مدل دقیق شود. مدل یادگیری ماشین آموزش‌دیده جدید به طور ضمنی یاد گرفته است چگونه اسپم را شناسایی کند.

با پیچیده‌تر شدن وظایفی که یک سیستم هوش مصنوعی باید انجام دهد، مدل‌های مبتنی بر قوانین شکننده‌تر می‌شوند: اغلب غیرممکن است که هر الگو و متغیری را که مدل باید در نظر بگیرد، صریحاً تعریف کرد. سیستم‌های یادگیری ماشین به عنوان حالت غالب هوش مصنوعی ظاهر شده‌اند زیرا یادگیری ضمنی الگوها از خود داده‌ها ذاتاً انعطاف‌پذیرتر، مقیاس‌پذیرتر و قابل دسترس‌تر است.

**بخش ۳: یادگیری ماشین چگونه کار می‌کند؟**

یادگیری ماشین از طریق منطق ریاضی کار می‌کند. بنابراین، ویژگی‌های مرتبط (یا “ویژگی‌ها”) هر نقطه داده باید به صورت عددی بیان شوند تا خود داده‌ها بتوانند به یک الگوریتم ریاضی که “یاد می‌گیرد” ورودی داده‌شده را به خروجی مورد نظر نگاشت کند، وارد شوند.

نقاط داده در یادگیری ماشین معمولاً به صورت بردار نشان داده می‌شوند، که در آن هر عنصر (یا بعد) از جاسازی بردار نقطه داده به ارزش عددی آن برای یک ویژگی خاص对应 دارد. برای حالت‌های داده که ذاتاً عددی هستند، مانند داده‌های مالی یا مختصات جغرافیایی، این کار نسبتاً ساده است. اما بسیاری از حالت‌های داده، مانند متن، تصاویر، داده‌های گراف رسانه‌های اجتماعی یا رفتارهای کاربر اپلیکیشن، ذاتاً عددی نیستند و بنابراین نیاز به مهندسی ویژگی کمتر شهودی برای بیان به صورت آماده برای یادگیری ماشین دارند.

فرآیند (اغلب دستی) انتخاب اینکه کدام جنبه‌های داده در الگوریتم‌های یادگیری ماشین استفاده شود، انتخاب ویژگی (feature selection) نامیده می‌شود. تکنیک‌های استخراج ویژگی داده‌ها را به تنها ابعاد مرتبط و معنادار خود پالایش می‌کنند. هر دو زیرمجموعه مهندسی ویژگی (feature engineering) هستند، رشته گسترده‌تر پیش‌پردازش داده‌های خام برای استفاده در یادگیری ماشین. یکی از تمایزهای قابل توجه یادگیری عمیق این است که معمولاً روی داده‌های خام عمل می‌کند و بسیاری از فرآیند مهندسی ویژگی—یا حداقل استخراج ویژگی—را خودکار می‌کند. این امر یادگیری عمیق را مقیاس‌پذیرتر، هرچند کمتر قابل تفسیر، از یادگیری ماشین سنتی می‌کند.

**پارامترهای مدل یادگیری ماشین و بهینه‌سازی**

برای یک مثال عملی، یک الگوریتم رگرسیون خطی ساده برای پیش‌بینی قیمت فروش خانه بر اساس ترکیب وزنی سه متغیر را در نظر بگیرید: متراژ مربع، سن خانه و تعداد اتاق‌خواب‌ها. هر خانه به عنوان یک جاسازی بردار با ۳ بعد نشان داده می‌شود: [متراژ مربع, اتاق‌خواب‌ها, سن]. یک خانه ۳۰ ساله با ۴ اتاق‌خواب و ۱۹۰۰ متر مربع می‌تواند به عنوان [۱۹۰۰, ۴, ۳۰] نشان داده شود (هرچند برای اهداف ریاضی، این اعداد ممکن است ابتدا مقیاس‌بندی یا نرمال‌سازی شوند تا به محدوده یکنواخت‌تری برسند).

الگوریتم یک تابع ریاضی ساده است:

در اینجا، A، B و C پارامترهای مدل هستند: تنظیم آن‌ها تعیین می‌کند که مدل هر متغیر را چقدر سنگین وزن‌دهی کند. هدف یادگیری ماشین یافتن مقادیر بهینه برای چنین پارامترهای مدل است. در حالی که بیشتر موارد واقعی یادگیری ماشین الگوریتم‌های پیچیده‌تر با تعداد بیشتری متغیر ورودی را شامل می‌شود، اصل همان است: بهینه‌سازی پارامترهای قابل تنظیم الگوریتم برای دستیابی به دقت بیشتر.

**بخش ۴: انواع یادگیری ماشین**

همه روش‌های یادگیری ماشین می‌توانند به عنوان یکی از سه پارادایم یادگیری متمایز دسته‌بندی شوند: یادگیری نظارت‌شده، یادگیری بدون نظارت یا یادگیری تقویتی، بر اساس طبیعت اهداف آموزشی آن‌ها و (اغلب اما نه همیشه) نوع داده‌های آموزشی که شامل می‌شوند.

* **یادگیری نظارت‌شده**: مدل را برای پیش‌بینی خروجی “صحیح” برای یک ورودی داده‌شده آموزش می‌دهد. این روش برای وظایفی اعمال می‌شود که نیاز به درجه‌ای از دقت نسبت به “حقیقت زمینی” خارجی دارند، مانند طبقه‌بندی یا رگرسیون.
* **یادگیری بدون نظارت**: مدل را برای تشخیص الگوهای ذاتی، وابستگی‌ها و همبستگی‌ها در داده‌ها آموزش می‌دهد. برخلاف یادگیری نظارت‌شده، وظایف یادگیری بدون نظارت شامل هیچ حقیقت زمینی خارجی نیستند که خروجی‌های آن باید با آن مقایسه شوند.
* **یادگیری تقویتی (RL)**: مدل را برای ارزیابی محیط خود و انجام اقدامی که بیشترین پاداش را به دست آورد، آموزش می‌دهد. سناریوهای RL شامل وجود یک حقیقت زمینی واحد نیستند، اما شامل وجود اقدامات “خوب” و “بد” (یا خنثی) هستند.

فرآیند آموزش انتها به انتها برای یک مدل داده‌شده می‌تواند و اغلب شامل رویکردهای ترکیبی است که بیش از یکی از این پارادایم‌های یادگیری را بهره می‌برد. برای مثال، یادگیری بدون نظارت اغلب برای پیش‌پردازش داده‌ها برای استفاده در یادگیری نظارت‌شده یا تقویتی استفاده می‌شود. مدل‌های زبانی بزرگ (LLMs) معمولاً آموزش اولیه (پیش‌آموزش) و تنظیم دقیق خود را از طریق انواع یادگیری نظارت‌شده انجام می‌دهند، و سپس تنظیم دقیق بیشتری از طریق تکنیک‌های RL مانند یادگیری تقویتی از بازخورد انسانی (RLHF) انجام می‌دهند.

در یک عمل مشابه اما متمایز، روش‌های مختلف یادگیری گروهی (ensemble learning) خروجی‌های چندین الگوریتم را جمع‌آوری می‌کنند.

**یادگیری نظارت‌شده**

الگوریتم‌های یادگیری نظارت‌شده مدل‌ها را برای وظایفی آموزش می‌دهند که نیاز به دقت دارند، مانند طبقه‌بندی یا رگرسیون. یادگیری نظارت‌شده هم مدل‌های یادگیری عمیق پیشرفته و هم طیف گسترده‌ای از مدل‌های یادگیری ماشین سنتی که هنوز در صنایع مختلف استفاده می‌شوند، را قدرت می‌بخشد.

* **مدل‌های رگرسیون**: مقادیر پیوسته را پیش‌بینی می‌کنند، مانند قیمت، مدت زمان، دما یا اندازه. مثال‌های الگوریتم‌های رگرسیون سنتی شامل رگرسیون خطی، رگرسیون چندجمله‌ای و مدل‌های فضای حالت هستند.
* **مدل‌های طبقه‌بندی**: مقادیر گسسته را پیش‌بینی می‌کنند، مانند دسته (یا کلاس) که یک نقطه داده به آن تعلق دارد، یک تصمیم باینری یا یک اقدام خاص که باید انجام شود. مثال‌های الگوریتم‌های طبقه‌بندی سنتی شامل ماشین‌های بردار پشتیبان (SVMs)، بیز ساده و رگرسیون لجستیک هستند.
* بسیاری از الگوریتم‌های یادگیری ماشین نظارت‌شده می‌توانند برای هر دو وظیفه استفاده شوند. برای مثال، خروجی آنچه اسماً یک الگوریتم رگرسیون است، می‌تواند سپس برای اطلاع‌رسانی به پیش‌بینی طبقه‌بندی استفاده شود.

برای اندازه‌گیری و بهینه‌سازی برای دقت، خروجی‌های مدل باید با یک حقیقت زمینی مقایسه شوند: خروجی ایده‌آل “صحیح” برای هر ورودی داده‌شده. در یادگیری نظارت‌شده سنتی، آن حقیقت زمینی توسط داده‌های برچسب‌دار فراهم می‌شود. یک مدل تشخیص اسپم ایمیل روی مجموعه داده‌ای از ایمیل‌ها آموزش داده می‌شود که هر کدام به عنوان SPAM یا NOT SPAM برچسب‌گذاری شده‌اند. یک مدل تقسیم‌بندی تصویر روی تصاویر آموزش داده می‌شود که پیکسل به پیکسل حاشیه‌نویسی شده‌اند. هدف یادگیری نظارت‌شده تنظیم پارامترهای مدل تا زمانی است که خروجی‌های آن به طور مداوم با حقیقت زمینی ارائه‌شده توسط آن برچسب‌ها مطابقت داشته باشد.

ضروری برای یادگیری نظارت‌شده استفاده از یک تابع ضرر (loss function) است که واگرایی (“ضرر”) بین خروجی مدل و حقیقت زمینی را در یک دسته از ورودی‌های آموزشی اندازه‌گیری می‌کند. هدف یادگیری نظارت‌شده به صورت ریاضی به عنوان کمینه‌سازی خروجی یک تابع ضرر تعریف می‌شود. پس از محاسبه ضرر، الگوریتم‌های بهینه‌سازی مختلف—که بیشتر آن‌ها شامل محاسبه مشتق(های) تابع ضرر هستند—برای شناسایی تنظیمات پارامتر که ضرر را کاهش دهند، استفاده می‌شوند.

از آنجایی که این فرآیند سنتی نیاز به یک انسان در حلقه برای ارائه حقیقت زمینی به شکل حاشیه‌نویسی داده‌ها دارد، “یادگیری نظارت‌شده” نامیده می‌شود. بنابراین، استفاده از داده‌های برچسب‌دار historically به عنوان ویژگی تعیین‌کننده یادگیری نظارت‌شده در نظر گرفته می‌شد. اما در سطح اساسی‌ترین، نشانه یادگیری نظارت‌شده وجود برخی حقیقت زمینی و هدف آموزشی کمینه‌سازی خروجی تابع ضرری است که واگرایی از آن را اندازه‌گیری می‌کند.

برای سازگاری با مفهوم انعطاف‌پذیرتر یادگیری نظارت‌شده، اصطلاح‌شناسی مدرن یادگیری ماشین از “نظارت” یا “سیگنال‌های نظارتی” برای اشاره کلی به هر منبع حقیقت زمینی استفاده می‌کند.

**یادگیری خودنظارت‌شده**

برچسب‌گذاری داده‌ها می‌تواند برای وظایف پیچیده و مجموعه داده‌های بزرگ، پرهزینه و زمان‌بر شود. یادگیری خودنظارت‌شده شامل آموزش روی وظایفی است که سیگنال نظارتی مستقیماً از داده‌های بدون برچسب به دست می‌آید—بنابراین “خود” نظارت‌شده.

برای مثال، خودرمزگذارها (autoencoders) آموزش داده می‌شوند تا داده‌های ورودی را فشرده (یا کدگذاری) کنند، سپس ورودی اصلی را با استفاده از آن نمایش فشرده بازسازی (یا رمزگشایی) کنند. هدف آموزشی آن‌ها کمینه‌سازی خطای بازسازی است، با استفاده از خود ورودی اصلی به عنوان حقیقت زمینی. یادگیری خودنظارت‌شده همچنین روش آموزشی اصلی برای مدل‌های زبانی بزرگ است: مدل‌ها نمونه‌های متنی با کلمات پنهان یا ماسک‌شده دریافت می‌کنند و وظیفه پیش‌بینی کلمات گم‌شده را دارند.

یادگیری خودنظارت‌شده اغلب با یادگیری انتقال (transfer learning) مرتبط است، زیرا می‌تواند مدل‌های پایه را با قابلیت‌های گسترده فراهم کند که سپس برای وظایف خاص‌تر تنظیم دقیق می‌شوند.

**یادگیری نیمه‌نظارت‌شده**

در حالی که یادگیری خودنظارت‌شده اساساً یادگیری نظارت‌شده روی داده‌های بدون برچسب است، روش‌های یادگیری نیمه‌نظارت‌شده از هر دو داده برچسب‌دار و بدون برچسب استفاده می‌کنند. به طور گسترده، یادگیری نیمه‌نظارت‌شده شامل تکنیک‌هایی است که از اطلاعات داده‌های برچسب‌دار موجود برای فرضیات درباره نقاط داده بدون برچسب استفاده می‌کنند تا این نقاط بتوانند در جریان‌های کاری یادگیری نظارت‌شده گنجانده شوند.

**یادگیری بدون نظارت**

الگوریتم‌های یادگیری ماشین بدون نظارت الگوهای ذاتی در داده‌های بدون برچسب را تشخیص می‌دهند، مانند شباهت‌ها، همبستگی‌ها یا گروه‌بندی‌های بالقوه. آن‌ها در سناریوهایی مفیدتر هستند که چنین الگوهایی لزوماً برای ناظران انسانی آشکار نیستند. از آنجایی که یادگیری بدون نظارت فرض نمی‌کند که یک خروجی “صحیح” شناخته‌شده از قبل وجود دارد، نیاز به سیگنال‌های نظارتی یا توابع ضرر سنتی ندارند—بنابراین “بدون نظارت”.

بیشتر روش‌های یادگیری بدون نظارت یکی از عملکردهای زیر را انجام می‌دهند:

* **الگوریتم‌های خوشه‌بندی**: نقاط داده بدون برچسب را به “خوشه‌ها” یا گروه‌بندی‌ها بر اساس نزدیکی یا شباهت آن‌ها به یکدیگر تقسیم می‌کنند. آن‌ها معمولاً برای وظایفی مانند تقسیم‌بندی بازار یا تشخیص تقلب استفاده می‌شوند. الگوریتم‌های خوشه‌بندی برجسته شامل خوشه‌بندی K-means، مدل‌های مخلوط گاوسی (GMMs) و روش‌های مبتنی بر چگالی مانند DBSCAN هستند.
* **الگوریتم‌های انجمن**: همبستگی‌ها را تشخیص می‌دهند، مانند بین یک اقدام خاص و شرایط خاص. برای مثال، کسب‌وکارهای تجارت الکترونیک مانند آمازون از مدل‌های انجمن بدون نظارت برای قدرت‌دهی به موتورهای توصیه استفاده می‌کنند.
* **الگوریتم‌های کاهش بعد**: پیچیدگی نقاط داده را با نشان دادن آن‌ها با تعداد کمتری ویژگی—یعنی در ابعاد کمتر—کاهش می‌دهند، در حالی که ویژگی‌های معنادار آن‌ها را حفظ می‌کنند. آن‌ها اغلب برای پیش‌پردازش داده‌ها، و همچنین برای وظایفی مانند فشرده‌سازی داده یا可视‌سازی داده استفاده می‌شوند. الگوریتم‌های کاهش بعد برجسته شامل خودرمزگذارها، تحلیل مؤلفه اصلی (PCA)، تحلیل تمایز خطی (LDA) و جاسازی همسایه تصادفی توزیع t (t-SNE) هستند.

همان‌طور که نامشان نشان می‌دهد، الگوریتم‌های یادگیری بدون نظارت را می‌توان به طور گسترده به عنوان “بهینه‌سازی خودشان” درک کرد. برای مثال، این انیمیشن نشان می‌دهد چگونه یک الگوریتم خوشه‌بندی K-means به طور تکراری مرکز هر خوشه را به تنهایی بهینه می‌کند. بنابراین، چالش آموزش مدل‌های بدون نظارت بر پیش‌پردازش مؤثر داده‌ها و تنظیم مناسب هایپرپارامترهایی تمرکز دارد که بر فرآیند یادگیری تأثیر می‌گذارند اما خودشان قابل یادگیری نیستند، مانند نرخ یادگیری یا تعداد خوشه‌ها.

**یادگیری تقویتی (RL)**

در حالی که یادگیری نظارت‌شده مدل‌ها را با بهینه‌سازی آن‌ها برای مطابقت با مثال‌های ایده‌آل آموزش می‌دهد و الگوریتم‌های یادگیری بدون نظارت خودشان را با مجموعه داده تطبیق می‌دهند، مدل‌های یادگیری تقویتی به طور کلی از طریق آزمون و خطا آموزش داده می‌شوند. آن‌ها به طور برجسته در رباتیک، بازی‌های ویدئویی، مدل‌های استدلال و سایر موارد استفاده که فضای راه‌حل‌ها و رویکردهای ممکن بسیار بزرگ، باز یا دشوار برای تعریف است، استفاده می‌شوند. در ادبیات RL، یک سیستم هوش مصنوعی اغلب به عنوان یک “عامل” (agent) اشاره می‌شود.

به جای جفت‌های مستقل ورودی-خروجی داده که در یادگیری نظارت‌شده استفاده می‌شود، یادگیری تقویتی (RL) روی تاپل‌های داده وابسته حالت-اقدام-پاداش عمل می‌کند. به جای کمینه‌سازی خطا، هدف یادگیری تقویتی بهینه‌سازی پارامترها برای بیشینه‌سازی پاداش است.

چارچوب ریاضی برای یادگیری تقویتی عمدتاً بر اساس اجزای زیر ساخته شده است:

* **فضای حالت**: شامل تمام اطلاعات مرتبط با تصمیماتی است که مدل ممکن است بگیرد. حالت معمولاً با هر اقدامی که مدل انجام می‌دهد، تغییر می‌کند.
* **فضای اقدام**: شامل تمام تصمیماتی است که مدل مجاز به انجام آن در یک لحظه است. برای مثال، در یک بازی تخته، فضای اقدام شامل تمام حرکات قانونی موجود در یک زمان داده‌شده است. در تولید متن، فضای اقدام شامل کل “واژگان” توکن‌های موجود برای یک مدل زبانی بزرگ است.
* **سیگنال پاداش**: بازخورد—مثبت یا منفی، معمولاً به عنوان یک مقدار اسکالر بیان‌شده—است که به عامل در نتیجه هر اقدام ارائه می‌شود. ارزش سیگنال پاداش می‌تواند توسط قوانین صریح، یک تابع پاداش یا یک مدل پاداش آموزش‌دیده جداگانه تعیین شود.
* **سیاست**: “فرآیند فکری” است که رفتار عامل RL را هدایت می‌کند. از نظر ریاضی، یک سیاست (π) تابعی است که یک حالت (s) را به عنوان ورودی می‌گیرد و یک اقدام (a) را برمی‌گرداند: π(s)→a \pi(s) \to a π(s)→a.

در روش‌های مبتنی بر سیاست مانند بهینه‌سازی سیاست تقریبی (PPO)، مدل مستقیماً یک سیاست را یاد می‌گیرد. در روش‌های مبتنی بر ارزش مانند Q-learning، عامل یک تابع ارزش را یاد می‌گیرد که امتیازی برای چقدر “خوب” بودن هر حالت محاسبه می‌کند، سپس اقداماتی را انتخاب می‌کند که به حالت‌های با ارزش بالاتر منجر شوند. یک پیچ را در نظر بگیرید: یک عامل مبتنی بر سیاست ممکن است “در این گوشه، به چپ بپیچ” یاد بگیرد، در حالی که یک عامل مبتنی بر ارزش امتیازی برای هر موقعیت یاد می‌گیرد و به سادگی به موقعیت مجاور با امتیاز بهتر حرکت می‌کند. رویکردهای ترکیبی، مانند روش‌های بازیگر-منتقد، یک تابع ارزش را یاد می‌گیرند که سپس برای بهینه‌سازی یک سیاست استفاده می‌شود.

در یادگیری تقویتی عمیق، سیاست به عنوان یک شبکه عصبی نشان داده می‌شود.

**بخش ۵: یادگیری عمیق**

یادگیری عمیق از شبکه‌های عصبی مصنوعی با لایه‌های زیاد—بنابراین “عمیق”—به جای الگوریتم‌های صریحاً طراحی‌شده یادگیری ماشین سنتی استفاده می‌کند. هرچند شبکه‌های عصبی زود در تاریخ یادگیری ماشین معرفی شدند، اما تا اواخر دهه ۲۰۰۰ و اوایل ۲۰۱۰، که تا حدی توسط پیشرفت‌های GPUs امکان‌پذیر شد، بر بیشتر زیرحوزه‌های هوش مصنوعی تسلط نیافتند.

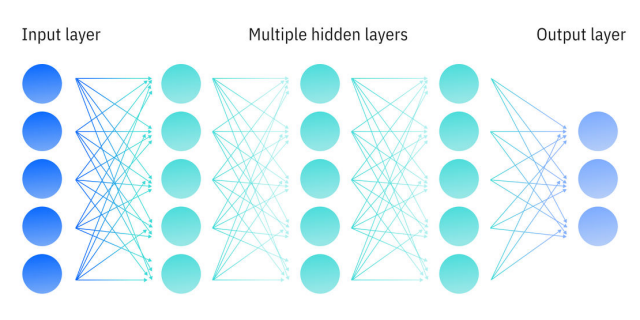
شبکه‌های عصبی، که به طور سست از مغز انسان الهام گرفته شده‌اند، شامل لایه‌های به‌هم‌پیوسته “نورون‌ها” (یا گره‌ها) هستند، که هر کدام عملیات ریاضی خود (به نام “تابع فعال‌سازی”) را انجام می‌دهند. خروجی تابع فعال‌سازی هر گره به عنوان ورودی به هر گره لایه بعدی خدمت می‌کند و این فرآیند ادامه می‌یابد تا لایه نهایی، جایی که خروجی نهایی شبکه محاسبه می‌شود. حیاتی است که توابع فعال‌سازی انجام‌شده در هر گره غیرخطی هستند، که شبکه‌های عصبی را قادر می‌سازد الگوها و وابستگی‌های پیچیده را مدل‌سازی کنند.

هر اتصال بین دو نورون یک وزن منحصربه‌فرد اختصاص داده می‌شود: یک ضریب که سهم یک نورون را در نورون لایه بعدی افزایش یا کاهش می‌دهد. این وزن‌ها، همراه با اصطلاحات بایاس بین لایه‌ها، پارامترهایی هستند که باید از طریق یادگیری ماشین بهینه شوند.

الگوریتم پس‌انتشار (backpropagation) محاسبه اینکه هر گره فردی چقدر به خروجی کلی تابع ضرر کمک می‌کند را امکان‌پذیر می‌کند، و اجازه می‌دهد حتی میلیون‌ها یا میلیاردها وزن مدل از طریق الگوریتم‌های گرادیان نزولی به صورت جداگانه بهینه شوند. به دلیل حجم و دانه‌بندی به‌روزرسانی‌های مورد نیاز برای دستیابی به نتایج بهینه، یادگیری عمیق به مقادیر بسیار زیادی داده و منابع محاسباتی در مقایسه با یادگیری ماشین سنتی نیاز دارد.

آن ساختار توزیع‌شده قدرت و تطبیق‌پذیری باورنکردنی مدل‌های یادگیری عمیق را فراهم می‌کند. داده‌های آموزشی را به عنوان نقاط داده پراکنده روی یک گراف دوبعدی تصور کنید. اساساً، یادگیری ماشین سنتی هدف دارد یک منحنی واحد پیدا کند که از هر یک از آن نقاط داده عبور کند؛ یادگیری عمیق تعداد دلخواهی از خطوط کوچک‌تر و قابل تنظیم جداگانه را برای تشکیل شکل مورد نظر کنار هم قرار می‌دهد. شبکه‌های عصبی تقریب‌گرهای جهانی هستند: از نظر نظری ثابت شده است که برای هر تابعی، یک آرایش شبکه عصبی وجود دارد که می‌تواند آن را بازتولید کند.

با این حال، فقط چون چیزی از نظر نظری ممکن است، به معنای آن نیست که از طریق روش‌های آموزشی موجود عملی باشد. برای سال‌ها، عملکرد کافی روی برخی وظایف حتی برای مدل‌های یادگیری عمیق دور از دسترس باقی ماند—اما با گذشت زمان، تغییرات در معماری شبکه عصبی استاندارد قابلیت‌های جدیدی برای مدل‌های یادگیری ماشین باز کرده است.



**شبکه‌های عصبی کانولوشنی (CNNs)**

شبکه‌های عصبی کانولوشنی (CNNs) لایه‌های کانولوشنی به شبکه‌های عصبی اضافه می‌کنند. در ریاضیات، کانولوشن عملیاتی است که یک تابع شکل دیگری را تغییر می‌دهد (یا کانولوش می‌کند). در CNNها، لایه‌های کانولوشنی برای استخراج ویژگی‌های مهم از داده‌ها با اعمال “فیلترهای” وزنی استفاده می‌شوند. CNNها عمدتاً با مدل‌های بینایی کامپیوتری و داده‌های تصویری مرتبط هستند، اما موارد استفاده مهم دیگری نیز دارند.

توضیح شکل:

**۱. تصویر ورودی (Input Image)**

تصویری که می‌خواهیم پردازش کنیم، اینجا به شکل یک جدول اعداد نمایش داده شده (این اعداد در واقع روشنایی یا شدت پیکسل‌ها هستند).

**۲. فیلتر یا کرنل (Filter / Kernel)**

یک جدول کوچک‌تر (اینجا **۳×۳**) که مثل یک ذره‌بین مخصوص روی تصویر حرکت می‌کند.

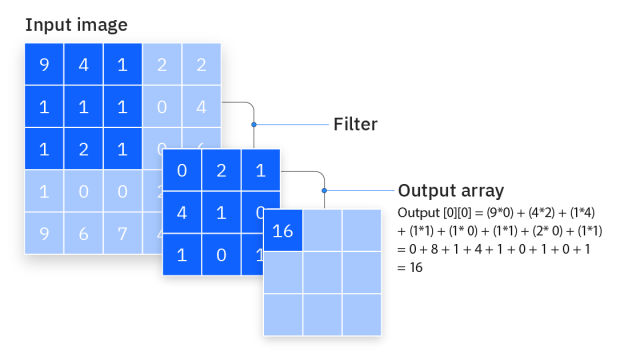
این فیلتر به دنبال الگوهای خاصی در تصویر می‌گردد؛ مثلا خط‌ها، لبه‌ها یا شکل‌ها.

**۳. نحوه‌ی محاسبه**

فیلتر را روی یک بخش کوچک از تصویر ورودی می‌گذاریم.

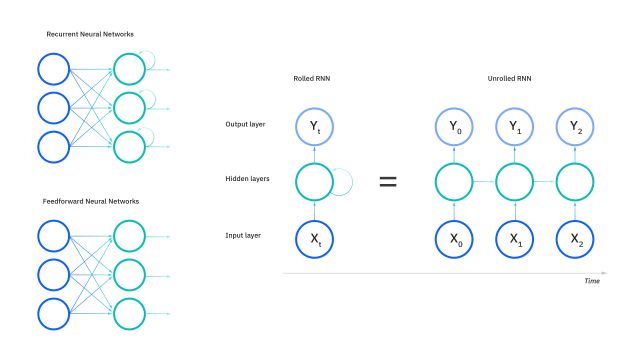
سپس:

* هر عدد از فیلتر را ضربدر عدد متناظر آن در تصویر می‌کنیم.
* همه این حاصل‌ضرب‌ها را با هم جمع می‌کنیم.
* Output[0][0] = (9×0) + (4×2) + (1×4)
* + (1×1) + (1×0) + (1×1)
* + (2×0) + (1×1) + (1×1)
* = 0 + 8 + 4 + 1 + 0 + 1 + 0 + 1 + 1
* = 16
* این **۱۶** اولین عدد در **آرایه خروجی** است.
* **۴. حرکت فیلتر**
* فیلتر کمی به سمت راست یا پایین حرکت می‌کند و همین کار را دوباره تکرار می‌کند تا کل جدول خروجی ساخته شود.
* **۵. نتیجه**
* آرایه خروجی در واقع یک نقشه‌ی ویژگی (Feature Map) است.
* این نقشه به CNN کمک می‌کند تا بخش‌های مهم تصویر را تشخیص دهد.
* اگر بخواهیم خیلی خلاصه بگوییم:
* این عملیات مثل وقتی است که یک شابلون کوچک را روی بخش‌های مختلف یک عکس می‌گذارید و فقط چیزهایی را یادداشت می‌کنید که با الگوی شابلون جور در می‌آید.



**شبکه‌های عصبی بازگشتی (RNNs)**

شبکه‌های عصبی بازگشتی (RNNs) برای کار روی داده‌های متوالی طراحی شده‌اند. در حالی که شبکه‌های عصبی پیش‌خور معمولی یک ورودی واحد را به یک خروجی واحد نگاشت می‌کنند، RNNها یک توالی از ورودی‌ها را به خروجی با عمل در یک حلقه بازگشتی نگاشت می‌کنند که در آن خروجی برای یک گام در توالی ورودی به عنوان ورودی به محاسبه گام بعدی خدمت می‌کند. در واقع این یک “حافظه” داخلی ایجاد می‌کند، به نام حالت پنهان، که RNNها را قادر می‌سازد زمینه و ترتیب را درک کنند.



**مدل‌های ترانسفورمر**

مدل‌های ترانسفورمر، که اولین بار در سال ۲۰۱۷ معرفی شدند، عمدتاً مسئول ظهور مدل‌های زبانی بزرگ و سایر ستون‌های هوش مصنوعی تولیدی هستند و نتایج پیشرفته در بیشتر زیرحوزه‌های یادگیری ماشین به دست می‌آورند. مانند RNNها، ترانسفورمرها ظاهراً برای داده‌های متوالی طراحی شده‌اند، اما راه‌حل‌های هوشمند اجازه داده‌اند بیشتر حالت‌های داده توسط ترانسفورمرها پردازش شوند. قدرت منحصربه‌فرد مدل‌های ترانسفورمر از مکانیسم توجه نوآورانه آن‌ها ناشی می‌شود، که مدل‌ها را قادر می‌سازد به طور انتخابی روی بخش‌های ورودی داده که در یک لحظه خاص در توالی مرتبط‌ترین هستند، تمرکز کنند.

**مدل‌های مامبا**

مدل‌های مامبا یک معماری شبکه عصبی نسبتاً جدید هستند که اولین بار در سال ۲۰۲۳ معرفی شدند و بر اساس یک واریاسیون منحصربه‌فرد از مدل‌های فضای حالت (SSMs) هستند. مانند ترانسفورمرها، مدل‌های مامبا وسیله نوآورانه‌ای برای اولویت‌بندی انتخابی اطلاعات مرتبط‌ترین در یک لحظه داده‌شده فراهم می‌کنند. مامبا اخیراً به عنوان رقیبی برای معماری ترانسفورمر ظاهر شده است، به ویژه برای مدل‌های زبانی بزرگ.

**بخش ۶: موارد استفاده یادگیری ماشین**

بیشتر کاربردهای یادگیری ماشین در یکی یا بیشتر از دسته‌های زیر قرار می‌گیرند، که عمدتاً بر اساس موارد استفاده و حالت‌های داده‌ای که روی آن‌ها عمل می‌کنند، تعریف می‌شوند.

**بینایی کامپیوتری**

بینایی کامپیوتری زیرحوزه هوش مصنوعی است که با داده‌های تصویری، داده‌های ویدئویی و سایر حالت‌های داده که نیاز به “دیدن” توسط مدل یا ماشین دارند، سروکار دارد، از تشخیص‌های بهداشتی تا تشخیص چهره تا خودروهای خودران. زیرحوزه‌های قابل توجه بینایی کامپیوتری شامل طبقه‌بندی تصویر، تشخیص شیء، تقسیم‌بندی تصویر و تشخیص کاراکتر نوری (OCR) هستند.

**پردازش زبان طبیعی (NLP)**

حوزه پردازش زبان طبیعی (NLP) طیف متنوعی از وظایف مربوط به متن، گفتار و سایر داده‌های زبانی را پوشش می‌دهد. زیرحوزه‌های قابل توجه NLP شامل چت‌بات‌ها، تشخیص گفتار، ترجمه زبان، تحلیل احساسات، تولید متن، خلاصه‌سازی و عامل‌های هوش مصنوعی هستند. در NLP مدرن، مدل‌های زبانی بزرگ همچنان با سرعتی بی‌سابقه وضعیت هنر را پیش می‌برند.

**تحلیل سری زمانی**

مدل‌های سری زمانی برای تشخیص ناهنجاری، تحلیل بازار و وظایف مرتبط با تشخیص یا پیش‌بینی الگو اعمال می‌شوند. آن‌ها از یادگیری ماشین روی داده‌های تاریخی برای انواع موارد استفاده پیش‌بینی استفاده می‌کنند.

**تولید تصویر**

مدل‌های انتشار (diffusion models)، خودرمزگذارهای واریانسی (VAEs) و شبکه‌های مولد خصمانه (GANs) می‌توانند برای تولید تصاویر اصلی استفاده شوند که الگوهای پیکسلی یادگرفته‌شده از داده‌های آموزشی را اعمال کنند.

**بخش ۷: عملیات یادگیری ماشین (MLOps)**

عملیات یادگیری ماشین (MLOps) مجموعه‌ای از شیوه‌ها برای پیاده‌سازی رویکرد خط مونتاژ برای ساخت، استقرار و نگهداری مدل‌های یادگیری ماشین است.

انتخاب دقیق و پیش‌پردازش داده‌های آموزشی، و همچنین انتخاب مدل مناسب، مراحل حیاتی در خط لوله MLOps هستند. اعتبارسازی پس از آموزش متفکرانه، از طراحی مجموعه داده‌های معیار تا اولویت‌بندی معیارهای عملکرد خاص، برای اطمینان از اینکه مدل خوب تعمیم می‌یابد (و نه فقط بیش‌تناسب با داده‌های آموزشی) لازم است.

پس از استقرار، مدل‌ها باید برای رانش مدل (model drift)، مشکلات کارایی استنتاج و سایر پیشرفت‌های نامطلوب نظارت شوند. یک عمل خوب تعریف‌شده治理 مدل برای کارایی مداوم ضروری است، به ویژه در صنایع تنظیم‌شده یا سریع‌التغییر.

**بخش ۸: کتابخانه‌های یادگیری ماشین**

تعداد زیادی ابزار، کتابخانه و چارچوب منبع‌باز برای ساخت، آموزش و آزمایش پروژه‌های یادگیری ماشین وجود دارد. در حالی که چنین کتابخانه‌هایی مجموعه‌ای از ماژول‌های پیش‌تنظیم‌شده و انتزاع‌ها را برای ساده‌سازی فرآیند ساخت مدل‌ها و جریان‌های کاری مبتنی بر یادگیری ماشین ارائه می‌دهند، متخصصان نیاز به آشنایی با زبان‌های برنامه‌نویسی رایج—به ویژه پایتون—برای استفاده کامل از آن‌ها دارند.

کتابخانه‌های منبع‌باز برجسته، به ویژه برای ساخت مدل‌های یادگیری عمیق، شامل PyTorch، TensorFlow، Keras و کتابخانه Transformers Hugging Face هستند.

کتابخانه‌ها و ابزارهای منبع‌باز قابل توجه یادگیری ماشین سنتی شامل Pandas، Scikit-learn، XGBoost، Matplotlib، SciPy و NumPy در میان بسیاری دیگر هستند.

IBM خود مجموعه قابل توجهی از آموزش‌ها برای مبتدیان و متخصصان پیشرفته یادگیری ماشین حفظ و به‌روزرسانی می‌کند.