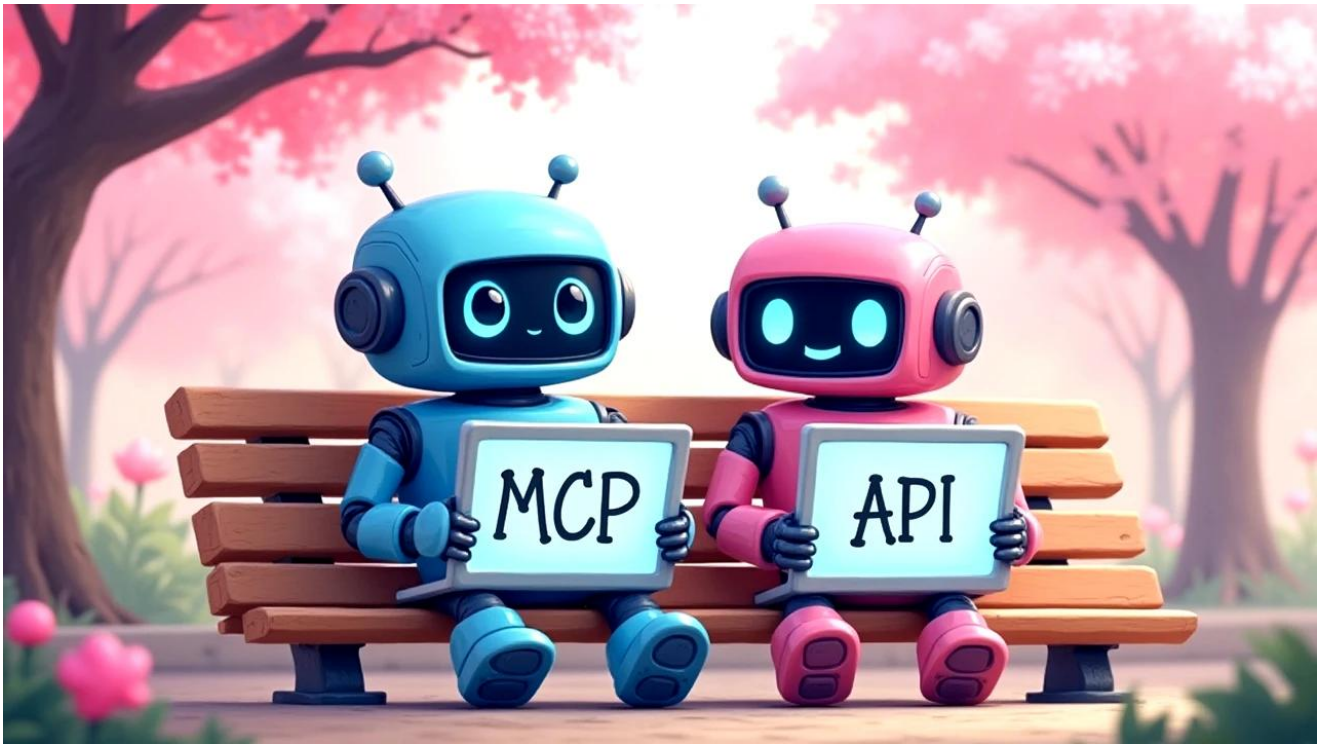




ادغام هوش مصنوعی با داده ها

ادغام هوش مصنوعی با داده‌ها، نیازمند تعامل موثر میان مدل‌های زبانی بزرگ (LLM) و منابع داده خارجی است تا دقت و کارایی این فرایند، به حداکثر برسد. تا پیش از سال ۲۰۲۴، این ارتباط معمولاً از طریق رابط‌های برنامه‌نویسی کاربردی (API) انجام می‌شد که امکان اتصال آسان داده‌ها و سرویس‌های مختلف به مدل‌های هوش مصنوعی را فراهم می‌کرد.

اخیراً شرکت **Anthropic** استاندارد جدیدی به نام Model Context Protocol (MCP) معرفی کرده است که انقلابی در نحوه ادغام هوش مصنوعی با داده‌ها ایجاد می‌کند. MCP امکانات فراتر از API‌ها ارائه می‌دهد و تجربه‌ای یکپارچه و انعطاف‌پذیر در اتصال به منابع خارجی فراهم می‌سازد.

در این مقاله، به بررسی دقیق ادغام هوش مصنوعی با داده‌ها از طریق MCP و API، تفاوت‌ها و کاربردهای هر یک و نقش MCP در آینده تعامل میان داده‌ها و هوش مصنوعی می‌پردازیم. هدف ما نشان دادن این است که چگونه این دو فناوری

می‌توانند در کنار هم، مرزهای جدیدی از درک و تصمیم‌گیری داده‌محور را در سیستم‌های هوشمند ایجاد کنند.

MCP چیست و چرا اهمیت دارد؟

برای درک بهتر MCP، می‌توان آن را به یک پورت USB-C برای برنامه‌های هوش مصنوعی تشبیه کرد. همان‌طور که پورت USB-C استاندارد است که انواع دستگاه‌ها و کابل‌ها را به لپ‌تاپ یا کامپیوتر متصل می‌کند، MCP نیز استاندارد است که ارتباط بین برنامه‌های هوش مصنوعی، مدل‌های زبان بزرگ و منابع داده خارجی را یکپارچه می‌سازد و جایگزینی مدرن برای بسیاری از API‌های سنتی محسوب می‌شود.

معماری MCP (Multi-Client Platform)

در معماری MCP، یک میزبان MCP Host نقش مدیریت‌کننده کل سیستم را بر عهده دارد. این میزبان می‌تواند چندین کلاینت MCP Clients را به صورت همزمان اجرا کند که هرکدام به صورت مستقل و با استفاده از پروتکل ۲.۰ JSON-RPC با سرورهای خارجی ارتباط برقرار می‌کنند.

خدمات ارائه شده توسط سرورها

سرورها می‌توانند خدمات مختلفی ارائه دهند، از جمله:

- دسترسی به پایگاه داده برای ذخیره و مدیریت داده‌ها
- دسترسی به مخزن کد (Code Repository) برای مدیریت نسخه‌های برنامه‌ها
- دسترسی به سرور ایمیل برای ارسال و دریافت ایمیل‌ها

نکات کلیدی در MCP

- میزبان MCP همه کلاینت‌ها را مدیریت می‌کند.
- هر کلاینت MCP می‌تواند به یک یا چند سرور متصل شود.

• ارتباط بین کلاینت‌ها و سرورها از طریق استاندارد JSON-RPC ۲.۰ برقرار می‌شود که به برنامه‌ها امکان می‌دهد بدون نیاز به تعداد زیادی API متفاوت، به منابع خارجی دسترسی یابند.

قابلیت‌های اصلی MCP برای کاربردهای هوش مصنوعی

MCP به دو نیاز اصلی برنامه‌های LLM، به‌ویژه عوامل هوش مصنوعی، پاسخ می‌دهد:

۱. **ارائه داده‌های زمینه‌ای (Contextual Data):** شامل اسناد، ورودی‌های پایگاه دانش، رکوردهای پایگاه داده و موارد مشابه است.

۲. **فعال‌سازی ابزارها و عملکردها (Tools):** امکان اجرای عملیاتی مانند جستجوی وب، فراخوانی سرویس‌های خارجی یا انجام محاسبات.

این قابلیت‌ها از طریق سرور MCP فراهم می‌شوند که مجموعه‌ای از عملیات پایه و ابتدایی را ارائه می‌دهد.

۱. ابزارها (Tools)

ابزارها عملکردهای مجزایی هستند که هوش مصنوعی می‌تواند فراخوانی کند؛ به‌عنوان مثال سرویس آب و هوا ممکن است ابزاری به نام `get weather` و سرویس تقویم، ابزاری برای ایجاد رویداد (`create event`) داشته باشد. هر سرور، نام ابزار، توضیحات، قالب ورودی و خروجی و قابلیت‌های آن را به‌صورت فهرست در دسترس قرار می‌دهد. وقتی LLM از طریق کلاینت MCP ابزاری را فراخوانی می‌کند، سرور MCP عملکرد مربوطه را اجرا می‌کند.

۲. منابع (Resources)

منابع داده‌ها (Data Resources)، مجموعه‌ای از اطلاعات فقط خواندنی هستند که توسط سرور MCP ارائه می‌شوند. این منابع می‌توانند شامل موارد زیر باشند:

- فایل‌های متنی (مثلا فایل README یا دستورالعمل‌ها)
- طرح پایگاه داده (Database Schema)
- محتوای فایل‌ها (مانند کد منبع، مستندات یا سایر داده‌ها)

کلاينت‌ها می‌توانند این منابع را هنگام نیاز، به‌صورت درخواست (Request) از سرور دریافت کنند و در سیستم خود استفاده کنند. این طراحی باعث می‌شود داده‌های ثابت و مرجع به‌صورت متمرکز از سرور MCP در اختیار همه کلاينت‌ها قرار گیرد.

۳. قالب‌های درخواست (Prompt Templates)

این قالب‌ها به‌صورت پیش‌تعریف شده طراحی شده‌اند و درخواست‌های پیشنهادی (Prompts) آماده‌ای را برای هوش مصنوعی فراهم می‌کنند تا کار با آن راحت‌تر و سریع‌تر انجام شود.

با این حال:

- همه سرورهای MCP از هر سه نوع ابتدایی (Primitives) استفاده نمی‌کنند.
- بسیاری از سرورها در حال حاضر بیشتر روی ارائه ابزارها (Tools) تمرکز دارند تا روی همه انواع ابتدایی.

به عبارت دیگر، بسته به سرور، ممکن است فقط برخی از قالب‌ها و قابلیت‌های پایه در دسترس باشند و تمرکز اصلی روی ابزارهای خاصی باشد که عملکردهای مشخصی را ارائه می‌دهند.

نکته مهم این است که عوامل هوش مصنوعی در زمان اجرا می‌توانند از سرور MCP بپرسند چه قابلیت‌هایی در دسترس است و سپس آن‌ها را به‌صورت یکنواخت و استاندارد فراخوانی کنند.

هر سرور MCP یک فهرست قابل‌خواندن را منتشر می‌کند، مانند:

- `tools/list` (فهرست ابزارها)

• [resources/list](#) (فهرست منابع)

• [prompts/list](#) (فهرست قالب‌های پیشنهادی)

این ساختار به عوامل هوش مصنوعی اجازه می‌دهد بدون نیاز به به‌روزرسانی کد، خیلی راحت قابلیت‌های جدید را کشف و استفاده کنند.

API چیست و چگونه کار می‌کند؟

رابط برنامه‌نویسی کاربردی یا API، روشی است که یک سیستم می‌تواند به عملکردها یا داده‌های سیستم دیگر دسترسی پیدا کند. API مجموعه‌ای از قوانین و پروتکل‌ها است که نحوه درخواست اطلاعات یا خدمات را تعریف می‌کند. به کمک API، توسعه‌دهندگان می‌توانند قابلیت‌های سیستم‌های خارجی را بدون ساخت همه چیز از ابتدا، در برنامه‌های خود ادغام کنند.

برای مثال، یک فروشگاه الکترونیکی می‌تواند از API پرداخت برای پردازش کارت‌های اعتباری استفاده کند. API به‌عنوان یک لایه انتزاعی عمل می‌کند؛ یعنی برنامه درخواست‌کننده (کلاينت) نیازی به دانستن جزئیات داخلی سرور ندارد و فقط باید بداند چگونه درخواست را ارسال کند و پاسخ را تفسیر نماید.

رایج‌ترین سبک API

Restful API، یکی از رایج‌ترین سبک‌های API است که بر پایه پروتکل HTTP کار می‌کند. در این سبک، کلاينت‌ها با استفاده از متدهای استاندارد HTTP با سرور ارتباط برقرار می‌کنند، از جمله:

- GET برای دریافت داده‌ها
- POST برای ایجاد داده‌های جدید
- PUT برای به‌روزرسانی داده‌ها
- DELETE برای حذف داده‌ها

پاسخ‌های سرور معمولا در قالب JSON ارسال می‌شوند که فرمت ساده و استاندارد
برای تبادل داده است.

برای مثال، در یک سیستم کتابخانه:

- برای دریافت جزئیات کتاب شماره ۱۲۳، یک درخواست GET به آدرس `books/۱۲۳/` ارسال می‌شود.

- برای ثبت امانت گرفتن یک کتاب، یک درخواست POST به آدرس `loans/` فرستاده می‌شود.

شباهت‌های MCP و API

هر دو MCP و API بر پایه معماری کلاینت-سرور ساخته شده‌اند. در API، کلاینت درخواست HTTP ارسال می‌کند و سرور پاسخ می‌دهد. در MCP، کلاینت درخواست‌هایی مانند `tools/call` به سرور MCP می‌فرستد و پاسخ دریافت می‌کند. در هر دو، جزئیات پیاده‌سازی پنهان است و فقط «رابط استاندارد» اهمیت دارد.

تفاوت‌های بنیادین MCP و API

هدف‌گذاری تخصصی MCP مقابل هدف‌گذاری عمومی API

MCP به‌طور ویژه برای ادغام برنامه‌های مبتنی بر مدل‌های زبان بزرگ (LLM) با داده‌ها و ابزارهای خارجی طراحی شده است. این پروتکل، استانداردهایی مانند ارائه داده‌های زمینه‌ای و فراخوانی ابزارها را طوری تعریف می‌کند که با نحوه عملکرد عوامل هوش مصنوعی هماهنگ باشد.

در مقابل، API‌ها برای کاربردهای عمومی‌تر ساخته شده‌اند و الزاما برای نیازهای هوش مصنوعی یا مدل‌های زبان بزرگ بهینه نیستند.

کشف پویا (Dynamic Discovery) در MCP

یکی از مزایای بزرگ MCP پشتیبانی از کشف پویا است. یعنی کلاینت MCP می‌تواند در زمان اجرا از سرور بپرسد: «چه قابلیت‌هایی داری؟» و سرور فهرستی از عملکردها و داده‌های موجود را ارائه می‌دهد. این امکان به عوامل هوش مصنوعی اجازه می‌دهد به‌صورت خودکار و بدون نیاز به به‌روزرسانی کد، با قابلیت‌های جدید سازگار شوند.

در مقابل، API‌های REST معمولاً چنین مکانیزمی ندارند و اگر API تغییر کند یا قابلیت جدید اضافه شود، باید کلاینت توسط توسعه‌دهنده به‌روزرسانی شود.

استانداردسازی رابط در MCP در برابر تنوع زیاد در API

در معماری MCP، همه سرورها، بدون توجه به نوع داده یا سرویس، از یک پروتکل و الگوهای استاندارد و یکسان استفاده می‌کنند. این یعنی نحوه درخواست و دریافت پاسخ در همه سرورهای MCP شبیه هم است.

اما در دنیای API‌ها، هر API روش خاص خودش را دارد:

- پارامترهای متفاوت

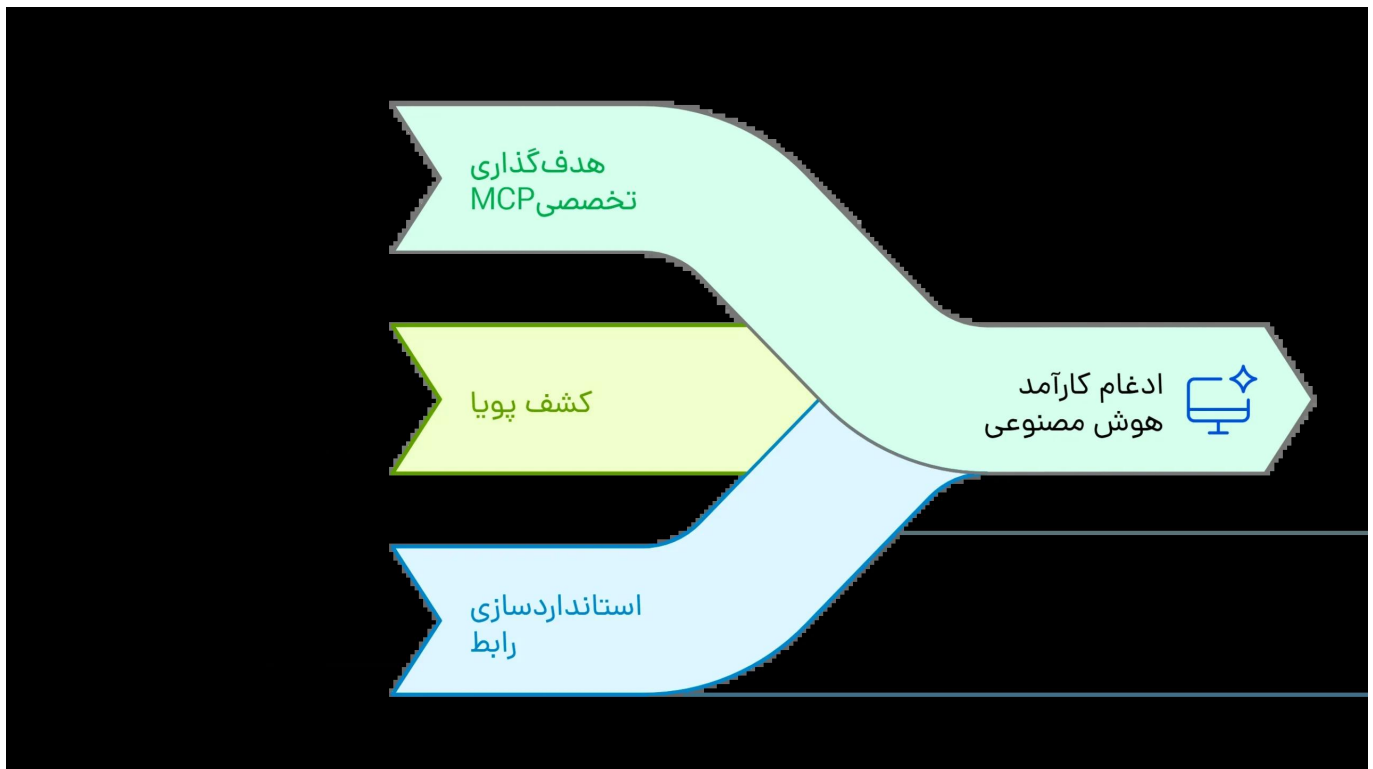
- آدرس‌های مختلف (نقاط انتهایی یا Endpoints)

- روش‌های متفاوت احراز هویت

این باعث می‌شود اگر یک عامل هوش مصنوعی بخواهد با چند API مختلف کار کند، مجبور باشد برای هر کدام یک آداپتور جداگانه بسازد که API را بشناسد و بتواند با آن ارتباط برقرار کند.

در مقابل، با MCP چون همه سرورها طبق یک استاندارد کار می‌کنند، کافی است یک بار نحوه تعامل با MCP را یاد بگیریم و بتوانیم با همه سرورهای MCP به‌شکل یکسان کار کنیم.

به عبارتی ساده: «یک بار بساز، بارها استفاده کن!»



MCP و API چگونه باهم کار می کنند؟

نکته جالب اینکه بسیاری از سرورهای MCP در واقع خودشان از APIهای سنتی برای انجام کارهایشان استفاده می کنند؛ به عبارت دیگر سرور MCP مانند یک لایه پوششی است که ترجمه بین فرمت MCP و رابط API اصلی سرویس را انجام می دهد.

برای مثال، سرور MCP گیت هاب، ابزارهایی مانند repository/list را به عنوان ابتدایی های MCP ارائه می دهد؛ اما در واقع، هر فراخوانی این ابزارها در پس زمینه، به یک درخواست REST API گیت هاب تبدیل می شود.

بنابراین MCP و APIها رقیب نیستند؛ بلکه لایه های مختلف در معماری هوش مصنوعی محسوب می شوند. MCP رابطی دوستانه تر برای هوش مصنوعی فراهم می کند که می تواند به صورت استاندارد و پویا با داده ها و ابزارهای مختلف ارتباط برقرار کند؛ در حالی که APIها زیرساخت های لازم را برای این ارتباط فراهم می آورند.

نمونه‌های کاربردی MCP در دنیای واقعی

امروزه سرویس‌های MCP برای منابع متنوعی در دسترس هستند، از جمله:

- سیستم‌های فایل
- Google Maps
- Docker
- Spotify
- فهرست روبه‌رشدی از منابع داده سازمانی

با کمک MCP، این سرویس‌ها می‌توانند به شکل استانداردتری در عوامل هوش مصنوعی ادغام شوند و قابلیت‌های جدید سریع و بدون دردسر به هوش مصنوعی اضافه گردد.

در نهایت، MCP یک گام مهم و انقلابی در حوزه ادغام هوش مصنوعی با داده‌ها و ابزارهای خارجی است. این پروتکل به طور خاص برای نیازهای مدل‌های زبان بزرگ و عوامل هوش مصنوعی طراحی شده و امکاناتی مانند کشف پویا، استانداردسازی رابط و پشتیبانی از ابزارهای تخصصی را فراهم می‌کند. در مقابل، API‌ها که پیش از این استاندارد اصلی تعامل بین سیستم‌ها بودند، بیشتر عمومی و متنوع هستند و برای هوش مصنوعی بهینه نشده‌اند.

با این حال، MCP و API‌ها مکمل هم هستند؛ MCP به عنوان لایه‌ای بالاتر و دوستانه‌تر برای هوش مصنوعی، از API‌های سنتی بهره می‌برد تا بهترین تجربه کاربری و توسعه را فراهم کند. در آینده نزدیک، MCP نقش کلیدی در تسهیل توسعه و استفاده از عوامل هوش مصنوعی خواهد داشت و به توسعه‌دهندگان اجازه می‌دهد سریع‌تر و ساده‌تر با دنیای پیچیده داده‌ها و سرویس‌ها ارتباط برقرار کنند.

اگر از هوش مصنوعی در کسب‌وکار خود استفاده می‌کنید و به دنبال راه‌هایی برای

اتصال هوشمندانه‌تر داده‌ها و ابزارهای خارجی هستید، آشنایی با MCP فقط آغاز راه است. پیشنهاد می‌کنیم برای درک عمیق‌تر کاربردهای عملی این فناوری، مقاله «اسکرپیت وبسایت با هوش مصنوعی Thunderbit» را نیز در وبسایت مدیر سبز مطالعه کنید.