



دانشگاه آزاد اسلامی واحد شیراز
معاونت فرهنگی دانشجویی



پیت و پایت

شماره ۱
تابستان ۱۴۰۴

نشریه فرهنگی - علمی - اجتماعی
مهندسی کامپیوتر



در این شماره خواهید خواند:

- سخنی با شما
- نشریه در نگاه دانشگاه
- گفت و گوی دوستانه با آقای دکتر تبرزد
- نورالینک
- کاهن معبد دلفی
- پرورش طوطی دیجیتال
- سنگ بنای نامرئی هوش مصنوعی
- جادوی مهندسی پرامیت
- هر آنچه ورودی های جدید باید بدانند!
- مسابقه

شناختنامه

زمینه انتشار:

فرهنگی، علمی، اجتماعی

نوع انتشار:

گاهنامه

صاحب امتیاز:

زینب شیرکول

شورای سردبیر:

علی پروینی، زهرا صحرانورد

هیئت تحریریه:

آنیتا سبحانیان، سپیده نوروزی مطلق، شیدا ستاری، علی پروینی، گودرز جعفری

ویراستار ادبی:

زینب شیرکول، شیدا ستاری

تیم گرافیک:

زهرا صحرانورد	طراحی جلد، صفحه آرایی
جواد طاهری	طراحی نقشه، صفحه آرایی
علی پروینی	امور گرافیکی
رضا نساج	امور گرافیکی
بهداد بهاری	طراحی لوگو
آرمان احراری	طراحی لوگو

عکاس:

حمیدرضا ملاحسینی

با سپاس فراوان از:

دکتر محمد رضا قاضی، ریاست محترم دانشگاه آزاد اسلامی شیراز
دکتر علیرضا بیابان نورد، معاونت محترم فرهنگی-دانشجویی
دکتر محمدعلی تبرزد، استادیار دانشکده کامپیوتر دانشگاه آزاد اسلامی شیراز
و کلیه عزیزانی که ما را یاری دادند.

فهرست

سرآغاز یک راه

- ۱ _____ سخنی با شما
- ۳ _____ نشریه در نگاه دانشگاه

یک فنجان حرف

- ۵ _____ گفت و گوی دوستانه با آقای دکتر تبرزد

زیر ذره بین علم

- ۲۳ _____ نورالینک
- ۲۷ _____ کاهن معبد دلفی
- ۳۷ _____ پرورش طوطی دیجیتال
- ۴۳ _____ سنگ بنای نامرئی هوش مصنوعی
- ۴۶ _____ جادوی مهندسی پرامپت

پردیس نامه

- ۵۰ _____ هر آنچه ورودی های جدید باید بدانند!

نبرد اندیشه ها

- ۶۶ _____ مسابقه



شیدا ستاری

مدیر مسئول

به نام خالق اندیشه و نوآوری

در این نشریه گرد هم آمده‌ایم تا پلی میان دانش، فناوری و خلاقیت بسازیم. جایی که از دنیای تکنولوژی، علوم نوین و آینده‌ای هوشمند سخن می‌گوییم. با هم در مسیری گام برمی‌داریم که نه تنها به درک عمیق‌تر مفاهیم علمی منجر می‌شود، بلکه راهی به سوی ساختن آینده‌ای روشن‌تر می‌گشاید. در این شماره، نگاه ما به یکی از تاثیرگذارترین تحولات عصر حاضر است: هوش مصنوعی و شبکه‌های یادگیری عمیق. این فناوری‌ها امروز نه تنها دروازه‌ای به کشف‌های تازه گشوده‌اند، بلکه تحولی بنیادین در بسیاری از علوم به وجود آورده‌اند. هوش مصنوعی، با شبیه‌سازی هوش انسانی، توانسته است پیچیده‌ترین مسائل را حل کند و مسیرهای نوینی برای پیشرفت در حوزه‌های مختلف از جمله پزشکی، مهندسی، اقتصاد و حتی هنر ارائه دهد. در قلب این تحول، شبکه‌های یادگیری عمیق قرار دارند که با قدرت بی‌نظیر خود در تحلیل داده‌ها و شناسایی الگوهای پیچیده، انقلابی را رقم زده‌اند. از تشخیص بیماری‌ها و پیش‌بینی بازارهای مالی گرفته تا خلق آثار هنری و بهبود زندگی روزمره، این فناوری‌ها نقشی انکارناپذیر ایفا می‌کنند. علاوه بر این، پیشرفت‌هایی مانند بازگشت بینایی به نابینایان با استفاده از دستگاه‌های پیشرفته، پروژه‌های جسورانه‌ای چون نورالینک برای اتصال مغز به ماشین، و توسعه مدل‌های زبانی بزرگ که توانایی درک و تولید متن را به سطوح بی‌سابقه‌ای رسانده‌اند، نشان می‌دهد که هوش مصنوعی تنها یک فناوری نیست، بلکه پلی است به سوی آینده‌ای که در آن محدودیت‌ها به چالش کشیده می‌شوند و فرصت‌های تازه‌ای برای بشر ایجاد می‌شود. در حقیقت، هوش مصنوعی و یادگیری عمیق، مرزهای دانش را بازتعریف کرده‌اند و با ارائه راه‌حل‌هایی نوآورانه، به ابزاری قدرتمند برای پیشبرد تمامی علوم تبدیل شده‌اند. این فناوری‌ها نه تنها به ما کمک می‌کنند تا بهتر بفهمیم و تصمیم‌گیری کنیم، بلکه به ما امکان می‌دهند جهان را به مکانی هوشمندتر و پایدارتر تبدیل کنیم. با افتخار شما را به همراهی دعوت می‌کنیم تا در این صفحات، با هم قدمی دیگر به سوی شناخت عمیق‌تر و آینده‌ای پیشرفته‌تر برداریم.



علی پروینی، زهرا صحرانورد

شورای سردبیر



چند باری شاهد بودیم که دانشجویان ترم پایینی سؤالاتی دربارهٔ ریزه‌کاری‌های رشته و زیست دانشگاهی می‌پرسیدند. این دغدغه، در کنار توجه به نیاز دانشجویان برای دسترسی به یک بستر علمی و تحلیلی، باعث شد تا در محیط دانشگاه، ایده‌ای برای شکل‌گیری یک نشریه دانشجویی مطرح شود. به پشتوانه و حمایت‌های ارزشمند آقای دکتر اسدی، مسیری آغاز شد که هدف آن نه‌تنها پاسخ به سؤالات دانشجویان، بلکه پوشش مباحث روز دنیای رایانه و ابعاد فرهنگی و اجتماعی مرتبط با زندگی دانشگاهی بود.

ایجاد این بستر، علاوه بر پاسخ‌گویی به دغدغه‌های دانشجویان، فرصتی فراهم کرد تا بتوانیم دانش و تجربیات خود را در محیطی پویا و علمی به اشتراک بگذاریم. مگر نه این است که علم تنها در انباشت دانش نیست، بلکه در اشتراک گذاشتن آن معنا می‌یابد؟ هر گام جدیدی با چالش‌های خود همراه است، و این مسیر نیز از این قاعده مستثنی نبود. خوشحالیم که پس از چند ماه تلاش، توانستیم اولین شماره از نشریه را منتشر کنیم. در کنار همه این‌ها، شروع این راه برای ما باعث شکل‌گیری ارتباطات و دوستی‌های ارزشمندی شده است. دوستی‌هایی که بدون حمایت و همدلی آنها پیشبرد نشریه غیر ممکن بود. پس، از همه کسانی که به نحوی در این مسیر همراه ما بودند، از صمیم قلب سپاسگزاریم.

در شماره‌ای که در دست دارید، به بررسی موضوعات پربحث دنیای هوش مصنوعی پرداختیم. همزیستی انسان و کامپیوتر و بحث پیرامون ورودی‌های جدید، از جمله موضوعاتی بودند که نتوانستیم از آن‌ها بگذریم. اصلاً نشریه جایی است که باید دربارهٔ این‌ها صحبت شود! این نشریه بازتابی از گوشه‌ای کوچک از دریای بیکران علم رایانه است و همواره جای رشد و تکامل دارد. از همین رو، بی‌صبرانه منتظر پیشنهادهای و انتقادهای شما دوستان گرامی هستیم، چراکه مشارکت شما همواره منبع الهام و افتخار ماست. اگر دیدگاهی یا ایده‌ای برای مطرح کردن دارید، مشتاق شنیدن صدای شما هستیم؛ هرکدام از شما می‌توانید بخشی از این جریان پویای علمی و فرهنگی باشید.

در پایان لازم می‌دانیم از اساتید گرانقدر دانشکده، نهایت تشکر را به‌جا بیاوریم که در این مسیر، همراه ما بوده‌اند.

امید داریم این نشریه به بستری تبدیل شود که از طریق تبادل تجربیات و ایده‌ها، ارتباطات ارزشمند میان نسل‌های مختلف دانشگاهی را تقویت کرده و موجبات رشد و یادگیری همگانی را فراهم آورد.

**دکتر محمدرضا قاعدی**

رئیس دانشگاه آزاد اسلامی استان فارس

«به نام خداوند لوح و قلم»

بی تردید پیشرفت، بالندگی، شکوفایی و توسعه علمی کشور عزیزمان ایران مدیون تلاش های صادقانه، هدفمند، مستمر و مبتنی بر دانش و تجربه ارزشمند دانشجویان؛ این امید های آینده کشور؛ است. دانشجویانی که خود از ارکان توسعه علمی ایران هستند و به علاوه نقش مهمی به عنوان مسببین تحقق نقشه جامع علمی کشور به عهده دارند و بدون کمک و همراهی آنان دستیابی به مرجعیت علمی از طریق انتشار دانش کیفی میسر نخواهد بود. در این بین مجلات علمی دانشجویی نماد و نشانه ای است از حضور اجتماعی و علمی دانشجویان کشور که بی شک زمینه ساز ارتقای توان و مهارت های دانشجویان در نقش های اجتماعی شان می باشد. از آن جا که نشریات دانشجویی حقیقتاً متعلق به دانشجویان بوده و هست و بستری را فراهم می آورد که تراوشات ذهنی و فکری نسل جوان به جامعه و دانشگاه منعکس گردد و زمینه ساز مناسبی برای گفتمان سازی، جریان سازی و آگاهی بخشی به دانشجویان از تحولات اساسی علمی و اجتماعی ایران و جهان است؛ نمی توان وجود و توسعه این رکن چهارم دموکراسی در میان دانشجویان را نادیده انگاشت. نشریات علمی-تخصصی و به ویژه نشریه علمی تخصصی کامپیوتر؛ با پژوهش، تحقیق و باز نشر مطالب ارزنده و به روز تخصصی این فناوری نوین جهانی، خصوصاً در عرصه های مهمی همچون هوش مصنوعی می تواند نقش مهم و سازنده ای در ارتقای سطح علمی و اندیشه ای دانشجویان داشته باشد. از خداوند قادر متعال برای همه دست اندرکاران این مجله وزین آرزوی موفقیت و سربلندی می نمایم.



دکتر علیرضا بیابان نورد

معاون فرهنگی-دانشجویی
دانشگاه آزاد اسلامی واحد شیراز

بسم الله الرحمن الرحيم

واقعیت این است که یکی از محلهای بسیار مهم در همه جای دنیا، آموزش و پرورش و دانشگاهها هستند. اگرچه مهمترین وظیفه دانشگاهها آموزش است؛ اما نباید فراموش کنیم که انسان در کنار آموزش، بایستی جنبه‌های مختلف زندگی خود را نیز پرورش دهد. فعالیت‌های فرهنگی دانشجویی در حوزه‌های مربوط به تشکلهای، کانونها و نشریات علاوه بر نقش آموزشی، به رشد و پرورش فردی دانشجویان کمک می‌کند. این فرایند پرورش، آنها را برای پذیرش مسئولیت‌های مختلف در جامعه آماده می‌کند تا مناصب مختلفی را در عرصه جامعه بپذیرند و جایگزین افرادی شوند که در سازمانها و نهادها تحت عناوین مختلف و در دانشگاه تحت عنوان استاد فعالیت می‌کنند. یکی از این نهادها، نشریات می‌باشند. نشریات دانشجویی از نهادهایی هستند که امکان مکتوب کردن ایده‌های ذهنی را فراهم می‌کنند. یکی از بخش‌های مهمی که می‌توان این دستاوردها را عرضه کرد، در نوشتن اتفاق می‌افتد و بستر نشریه دانشجویی جایی است که دانشجو می‌تواند دستاوردهای علمی، پژوهشی، فرهنگی و اجتماعی خود را در اختیار بقیه بگذارد و از این طریق هم به رشد دیگران کمک می‌کند و هم خودش پرورش پیدا خواهد کرد. با توجه به تجربیات ارزنده شما، چه راهکارهایی برای موفقیت بیشتر دانشجویان پیشنهاد می‌کنید؟

فراتر از آموزش، به دیگر فعالیت‌های دانشجویی هم بپردازند. هم‌اکنون، معاونت فرهنگی-دانشجویی کاملاً در دانشگاه فعال می‌باشد. دانشجویانی که با معاونت ارتباط دارند، می‌دانند که می‌توانند تشکل، کانون و نشریه تشکیل بدهند و در مسابقات و جشنواره‌های مختلفی شرکت کنند. پیشنهاد من به دانشجویان این است که حتماً از همین بسترها شروع کنند تا ان شاء الله در آینده نقش‌آفرین باشند. پیشنهاد دیگر من این است که دانشجویان علاوه بر درس، با مطالعه آزاد به رشد فکری خود کمک کنند. در همین زمینه، مقام معظم رهبری به مناسبت چهلمین سال انقلاب، یعنی سال ۱۳۹۷، بیانیه گام دوم انقلاب صادر کردند. این بیانیه دارای دو بخش است: بخش نخست، مروری بر چهل سال گذشته. بخش دوم، پیشنهادها و چشم‌انداز چهل سال آینده. به نظر بنده، دانشجویان این بیانیه را بخوانند تا ان شاء الله آینده خوبی برای خود و کشورشان بسازند. درنهایت از همه عزیزان ممنونم و برایتان آرزوی موفقیت دارم.



مهمان این شماره از نشریه بیت و بایت، دکتر محمدعلی تبرزد، از اساتید برجسته دانشگاه آزاد اسلامی واحد شیراز هستند. ایشان با تجربه‌ای ارزشمند در تدریس و پژوهش، به‌عنوان یکی از چهره‌های شناخته‌شده و مورد احترام در حوزه مهندسی کامپیوتر فعالیت می‌کنند.

گفت و گوی دوستانه با آقای دکتر تبرزد

مصاحبه: گودرز جعفری، شیدا ستاری
نویسنده: شیدا ستاری



بودم تا ۱۳۹۰، باز رفتم برای دکترا دانشگاه شیراز تا ۱۳۹۶، که میشه چند سال پیش. من برای طرح خدمت اومدم دانشگاه آزاد اسلامی شیراز. اون زمان خیلی معمول بود. نمیدونم الان می‌گیرن یا نه. اون موقع به جای سربازی شما دو سال یا سه سال می‌رفتی به دانشگاه هیئت علمی می‌شدی. حقوق هیئت علمی بود، همه امکانات هیئت علمی بود، اسمش سربازی بود. فرصت خوبی بود برای کسانی که مهاجرت نمی‌کردن و می‌خواستن تو ایران

**لطفاً خودتان را معرفی کنید
و کمی درباره سوابق تحصیلی
و شغلی تان توضیح دهید**

سلام، من ورودی ۱۳۷۶ دانشگاه شیراز بودم. زمان ما خیلی معمول بود که لیسانس رو چهار ساله تموم می‌کردن اما من جزو کسانی بودم که لیسانس رو شش ساله خوندم که تا ۱۳۸۲ طول کشید. سال ۱۳۸۲ که رفتم ارشد دانشگاه تهران بودم، باز اونم یه سه سالی طول کشید تا ۱۳۸۵ که اومدم همین دانشگاه اسلامی واحد شیراز، دیگه

همین رندومنسی که یک نفر بیشتر این رشته را انتخاب کرده بود، به جای عمران کامپیوتر قبول شدم. من رشته اولم رو به این دلیل که تمایلی به رفتن به شهر دیگری نداشتم و بعد فاصله برام خیلی مهم بود، انتخاب کردم. حتی بعدها یکی از گزینه هایی که برای تدریس داشتم، به عنوان هیئت علمی دانشگاه صنعتی شیراز بود که به همین علت پیگیری نکردم.

در آن زمان، ما که اصلاً نمی دونستیم کامپیوتر چی هست! در سال ۱۳۷۶ یعنی زمانیکه من وارد دانشگاه شدم، مرکز کامپیوتر دانشگاه شیراز کامپیوترهایی با نسخه خیلی قدیمی ۲۸۶ داشت، و سه کامپیوتر با نسخه ۴۸۶ به دانشجویان تحصیلات تکمیلی اختصاص داده بودند.

در آن سالهایی که با کامپیوترهای دانشگاه شیراز کار می کردیم، سرعت دانلود شبکه اینترنتی که ما داشتیم خیلی جذاب بود! مثلاً در حد پنج بایت در ثانیه دانلود می کرد. در ایام عید که دانشگاه تعطیل می شد و خلوت بود، ما مجوز می گرفتیم که در بخش بمونیم و آن موقع سرعت به کیلو بایت در ثانیه هم می رسید و خوشحال بودیم که می توانستیم یک فایل ۱۵ مگابایتی رو در یک روز دانلود کنیم.

خب در زمانی که بحث کامپیوتر تازه بود و حتی کامپیوتر در دانشگاه های تخصصی اونقدر وجود نداشت چه برسه به اینکه کسی کامپیوتر شخصی داشته باشه؛ کسی نمیدونست که کامپیوتر چیه و آیا این رشته به درد میخوره یا نه اما مهندس های عمران سابقه قدیمی تر و شرایط و پرستیژ

بمونن. یا دو سال می رفتن خدمت یا سه سال می اومدن هیئت علمی می شدن. خب، خیلی بهتر بود. من سال ۱۳۸۵ که اومدم، سه سالش حداقل طرح خدمت بود. یکی دو سال هم بعدش. خیلی برام مهم نبود برم ادامه دکترا رو. تا بعدش دیدم که خب، حالا لازمه دکترا برم.

🗨️ دلیل انتخاب این رشته چه بود؟

به طور کلی، نه فقط در سن هجده سالگی، همین الان که چهل و چند سال سن دارم، احساس می کنم رندومنسی^۱ که در جهان وجود دارد خیلی پیچیده تر از اون هست که فکر کنیم این ما هستیم که تصمیم می گیریم زندگیمون چه مسیری طی کنه. حتی همین حالا که در دانشگاه آزاد درس می دهم، آنقدر پارامترهای جانبی وجود دارد که در انتها این تصمیم که بخوام ۱۰ سال دیگر اینجا تدریس کنم، به راحتی می تونه به هر علتی تغییر کنه. مثل اینکه مشکلی پیش بیاید و مجبور شم دانشگاه رو ترک کنم یا حتی از ایران برم، یا شاید دانشگاه به این نتیجه برسه که اساتیدی مثل ما دیگه به دردش نمی خورن.

عملاً برای من در اون سن همین رندومنس وجود داشت. در دوره ما، معمولاً رشته ریاضی را به عنوان رشته اول انتخاب می کردند و به سمت رشته های خوب و پرمقتاضی مهندسی عمران، مکانیک و کامپیوتر می رفتند.

من در اولویت اول مهندسی عمران دانشگاه شیراز و در اولویت دوم کامپیوتر همین دانشگاه را انتخاب کرده بودم. در آن سال ظرفیت رشته عمران مثلاً اگر ۱۳ نفر بود، من رتبه چهاردهم شدم و عملاً به دلیل



در طی تحقیقاتان، آیا ابزار یا فناوری خاصی وجود داشته که به طور ویژه برایتان برجسته و تاثیرگذار باشد؟

ابزارهای مهندسی کامپیوتر با سرعت نمایی دارن تغییر می‌کنن و این فرق اصلیش با بقیه مهندسی‌هاست. قبلاً ما از ابزارهایی مثل متلب و R با کتابخونه‌های مخصوص به خودشون استفاده می‌کردیم، ولی حدود ۱۰ سال هست که رفتیم سراغ ابزارایی مثل پایتون و کتابخونه‌های مختلف این زبان که فقط باید یه تابع رو فیت کنی. مثلاً یه کلاسیفایر^۲ رو با یه دستور fit. به کار ببری تا همه محاسبات انجام بشه. ابزار معروفی که الان تو پیاده‌سازی برای شروع استفاده می‌شه، مجموعه Scikit-learn هست که به نظرم یه کتابخونه خوب و کامله برای کسی که می‌خواد شروع به کدنویسی کنه. از طرفی، کتابخونه‌های هوش مصنوعی شدن ابزار اصلی‌مون. قبلاً شبکه‌های عصبی به خاطر ساختاری که داشتن، قوی نبودن و در شبکه‌های fully connected لینک‌های زیادی وجود داشت که همین باعث می‌شد به خاطر زمان و داده زیاد نتونی اون رو آموزش بدی. با تغییر و تحول اساسی ساختارها و ورود به شبکه‌های عصبی کانولوشنی که به جای یک ساختار کلی، ساختار محلی رو در داده‌ها پیدا می‌کردن، امکان افزایش تعداد فیلترها فراهم شد و نتیجش این بود که با همون داده و زمان کم، مفهوم بیشتری تولید بشه. در این ساختار جدید، سخت‌افزار و به‌خصوص کارت‌های گرافیکی شدن غول‌هایی که دیگه قابل مقایسه با ۱۰ سال پیش نیستن، یه قانون کلاسیک

مشخصی داشتند و اولویت من عمران بود اما به خاطر همان رندومنسی که گفتم، مسیر من از عمران به کامپیوتر تغییر کرد.

در طول تحصیلاتان، بر روی چه پروژه یا تحقیقاتی مرتبط با حوزه هوش مصنوعی کار کرده اید؟

هوش مصنوعی به نظرم یه جور جعبه‌ابزار^۱ هست که در اختیار بقیه موضوعات قرار می‌گیره. قدیم هر الگوریتم یادگیرنده‌ای که طراحی می‌شد بهش هوش مصنوعی می‌گفتن، الان بهتره بهش بگیم هوش مصنوعی کلاسیک. مثلاً الگوریتم درخت هرس آلفا-بتا رو الانم بهش می‌گیم هوش مصنوعی. ایده این بود که هوش انسان رو جوری در الگوریتم اپلای کنی که یک مقداری هوشمند بشه.

این رو در برنامه‌نویسی‌هامون همیشه داشتیم و استفاده می‌کردیم، منتها اون موقع نسخه‌های خیلی ساده‌ترش بود. اینجوری نبود که مثل الان ساختار پیچیده‌ای داشته باشه یا انقدر پیشرفته باشه.

پایان نامه، که بیشتر به آی‌تی مربوط می‌شد، درمورد سیستم‌های چند عامله بود که نوعی تصمیم‌گیری هوشمند محسوب میشه. مقالاتی که در طی دوران ارشدم دادم یک مقداری هوشمند بودند منتها هوش‌های اولیه و کلاسیکی مثل ژنتیک در آنها استفاده می‌شد. دوره دکترا رو که شروع کردم جدی‌تر سراغ هوش مصنوعی رفتم و روی شاخه‌های جدیدی کار کردم.

پیش‌بینی کنی این آدم نبود! تو احتمالاً ساتوشی بودی. یا اگه سه سال پیش بیت‌کوین خریده بودم، الان سه برابر شده بود، در این مورد هم، فرد سه سال پیش این فکر رو نمی‌تونست بکنه که بیت‌کوین قراره ۶۰ هزار دلار، ۷۰ هزار دلار بشه. پس هیچوقت این افسوس رو نباید بخوریم. الان من از رشته‌ام خیلی خوشحالم و به نظرم کامپیوتر تخصص من هست، یعنی من مطمئناً در عمران یا مکانیک به اندازه‌ای که در کامپیوتر موفقیت به دست آوردم، موفق نمی‌شدم.

من فکر می‌کنم کسانی در این رشته موفق می‌شن که می‌تونن برای هر چیزی الگوریتم داشته باشن، یعنی از صبح که می‌خوای لباسی رو انتخاب کنی، در ذهنت با یه الگوریتمی تصمیم بگیری که این پیراهن رو بپوشم، یا مثلاً امروز که می‌خوام برم بیرون باید این کارها رو انجام بدم و اگه نشد یه کار دیگه رو جایگزین کنم، مثل اینکه برای خودت مفهوم حلقه در برنامه‌نویسی رو پیاده‌سازی کنی. به نظر من چنین شخصی می‌تونه در همه حیطه‌های کامپیوتر موفق بشه چون ایده اصلی کامپیوتر الگوریتمی فکر کردنه. به نظر خودم، مغز من هم الگوریتمی بوده و شاید کار ایده‌آل برای من همین کاره.

شاید هم در هجده سالگی عمران می‌خوندم و چیزی که بودم، ۱۰۰ برابر بهتر از الان بود یا حتی می‌تونستم نیمی از شیراز رو داشته باشم، نمی‌دونم! اما افسوس این رو نمی‌خورم و همین جا که هستم رو خیلی دوست دارم. به نظرم، یه سرمایه‌ای در زندگی آدم‌ها

هست که هر دو سال یه بار سرعت کامپیوترها دو برابر می‌شه، و عملاً شبکه‌های عصبی باعث شدن شرکت‌های گرافیکی هم از زمین بلند شن و تبدیل بشن به یه ابزار مهم در این حوزه.

اگر با دیدگاه و تجربه امروزتان امکان بازگشت به گذشته را داشتید، چه کارهایی را انجام می‌دادید و چه تصمیماتی را احتمالاً نمی‌گرفتید؟

به نظرم هیچوقت نباید از خودتون این سوال رو پرسید. این سوال خودش یه تناقض داره، اگه قرار باشه با تفکر حال حاضرتون به ۱۰ سال قبل برگردید، تفکر ۱۰ سال قبلتون باید ۱۰ سال بزرگتر می‌بود و این امکان‌پذیر نیست! چون اون موقع تصمیم دیگه‌ای می‌گرفتید و به تفکر الان خودتون نمی‌رسیدید. عملاً یه باگ مفهومی پشت این موضوع هست و باید حواستون باشه هیچوقت افسوس این رو نخورید که من ۱۰ سال پیش باید یه سری کارها رو می‌کردم و نکردم.

من حتماً در هجده سالگی یه سری تصمیماتی گرفتم که وقتی فکر می‌کنم می‌گم چه کار مسخره‌ای بوده و کلی از این تصمیمات رو می‌تونم در زندگیم پیدا کنم، اما اولین چیزی که به خودم می‌گم اینکه که من فقط اون زمان هجده ساله بودم که این تصمیم رو گرفتم و اگه برگردم به هجده سالگی، همین تصمیم رو دوباره می‌گیرم.

یه تصمیمی که خیلی‌ها افسوسش رو می‌خورن اینکه که اگه ۲۰ سال پیش بیت‌کوین خریده بودم، الان میلیاردر بودم. واقعیت اینکه که اگه تو می‌تونستی آینده رو



در حقیقت جهان همیشه بر اساس یه ساختار هرمی عمل می‌کنه. نه کمونیسم جواب می‌ده و نه دموکراسی، این‌ها فقط اسم‌های زیبایی هستن که برای توجیه یه سیستم استفاده می‌شن، سیستمی که برده‌ها برای هرم قدرت کار می‌کنن و برای اونا پول درمیارن. این ساختار طبیعی هست و همه جا دیده می‌شه ولی مردم نمی‌تونن هرم قدرت رو بپذیرن و وقتی اسمش رو می‌شنون، اذیت می‌شن.

در این ساختار، ارز رایج زمانه، و مهم اینه که چطور اون رو خرج کنی. ما هرکدوم حدود ۸۰ سال زمان داریم. حالا باید فکر کنیم که این ۸۰ سال رو چطور خرج کنیم. من اون رو صرف چیزی می‌کنم که خوشحالم کنه. اگه الان اینجا نشستم و در دانشگاه درس می‌دم، مهم نیست چقدر حقوق می‌گیرم. مهم اینه که آیا بازخورد مفیدی می‌گیرم؟ آیا از این کار خوشحال هستم؟

من خوشحالم که اینجا هستم، چون چهار تا دانشجو دارم که به هر دلیلی به دانشگاه‌های دیگه نرفتن و اومدن اینجا. وقتی می‌بینم می‌تونم یه تلنگر به اونا بزنم و الان که بیست ساله هستن و قبل از اینکه شروع به دنبال پول دویدن کنند تا در نهایت در شصت سالگی از درد ورم مفاصل نتونن پولشون رو خرج کنن، به اونا یادآوری کنم زمانتون رو هوشمندانه خرج کنید. پیشرفت شغلی و علمی داشته باشید، اما گم نشید. نباید زندگی رو فقط به پول و قدرت محدود کنید. ارز اصلی شما زمانه و در ازای اون باید شادی به دست بیارید. حالا یکی هست صدای شنیدن سکه‌هاش خوشحالش

وجود داره که از همه چیز مهم‌تره. من همیشه در مورد اینکه بعضی‌ها می‌گن که کاش به جای یک خونه ۱۰ تا یا ۱۰۰ تا خونه داشتم، یا حتی راجع به چیزی که خودم در ذهنم دارم، آیا من دوست داشتم مثلاً الان به خارج رفته باشم و تبدیل به یه غول در زمینه کامپیوتر شده بودم؟ یه جوابی دارم که با اون زندگی می‌کنم و خوشحالم آن هم اینست که سرمایه زندگی آدم‌ها یه ارز جهانیه که این ارز جهانی برای همه ثابت و اون زمانه.

در مورد هر چیز دیگه‌ای مثل پول درآوردن یا زندگی کردن، تو در هر حال داری زمانت رو خرج می‌کنی. فرقی نمی‌کنه میلیارد میلیارد درآمد داشته باشی یا میلیون میلیون! تو درواقع داری زمانت رو می‌فروشی و به جای اون پول دریافت می‌کنی. مثلاً هشت ساعت از وقت امروزت رو می‌فروشی و در عوض یک میلیون پول دریافت می‌کنی اما این هشت ساعتی رو که فروختی، در مقابلش چه چیزی گیرت اومده؟ پول که فقط یه اسکناس و عدد تو حسابت هست. تو باید به جای خرج این زمان چیزی بگیری که ارزش داشته باشه. به نظرم، خوشحال بودن خیلی مهم‌تر از هر چیز دیگه‌ایه.

ارز مشترک رو همیشه زمان در نظر بگیرید. به نظرم این ارز، مشترک و جهانیه. یعنی فرقی نداره. کسی که در آمریکا زندگی می‌کنه، قراره هشتاد سالگی بمیره و منی که تو آفریقا زندگی می‌کنم. جفتمون یک ارز تو دست داریم.

من این موضوع رو در جهان عدالت می‌بینم. خیلی‌ها می‌گن در جهان بی‌عدالتی وجود داره و باید ساختارش تغییر کنه، ولی

حتی بعضی‌ها که واقعاً به جایگاهی شبیه ایلان ماسک می‌رسن، در نهایت زندگی آرام و خوشایندی ندارن. اونا مجبورند از صبح تا شب حواسشون به اطرافشون باشه که مبادا کسی بهشون ضربه بزنه یا آسیبی برسونه. مثالش ترامپ؛ فکر نمی‌کنم او زندگی لذت‌بخشی داشته باشه. من فکر می‌کنم از او خوشحال‌تر هستم.

آیا در دوران دانشجویی خود تجربه یا خاطره‌ای خاصی دارید که همچنان در ذهن شما مانده باشد؟ خوشحال می‌شویم داستانی از این تجربیات را بشنویم

از استادهای تأثیرگذار من در دوره لیسانس، دکتر توحیدی در دانشگاه شیراز بود که تا چند سال پیش هم برنامه‌نویسی درس می‌داد. زمان ما، درس‌های زیادی می‌دادن: سیستم‌عامل، برنامه‌نویسی و چند تا درس دیگه. مغز الگوریتمیکی داشتن و وقتی می‌خواستن برنامه‌نویسی رو توضیح بدن، ساختار و داستان پشتش رو می‌ساختن. مثلاً وقتی می‌خواستن یه مفهوم رو توضیح بدن، سعی می‌کردن از حالت‌هایی که ممکن بود به نتیجه نرسن هم بگن تا بدونیم چرا این روش خوبه و بقیه روش‌ها مناسب نیستن. مثلاً وقتی می‌خواستن حلقه for رو یاد بدن، اول درباره نیاز به تکرار توضیح می‌دادن تا نیازش رو بفهمیم، بعد ساختارش رو حدس بزنیم و در نهایت حلقه for رو که همون ساختار رو داره، معرفی می‌کردن. دکتر توحیدی تأثیر زیادی رو من داشتن. من عموماً کلاس‌ها رو نمی‌رفتم یا یکی در میان می‌رفتم، ولی کلاس‌های ایشون

می‌کنه و باید تا آخر عمرش پول جمع کنه اما اگه این تو رو خوشحال نمی‌کنه، مثلاً سفر برای تو خوشحال‌کننده‌ست، زندگیت رو بر مبنای پول جمع کردن نذار.

به نظرم، زندگی کلاً دو سطح داره، یه سطح، حداقل نیازهای حیوانیه که از نظر من می‌شه در روز سه وعده غذای سالم داشته باشی بخوری، دسترسی به آب سالم داشته باشی، خونه داشته باشی. این می‌شه حداقلی که لازم نیست شب در خیابون یخ بزنی. از این سطح که عبور کنی، سطح رفاه زندگی‌ها لگاریتمی می‌شه. یعنی اگه به جای ۱ میلیارد، ۱۰ میلیارد داشته باشی، کیفیت زندگی‌ات ۱۰ برابر نمی‌شه. خوشبختی تو به شکل خطی رشد نمی‌کنه، بلکه چیزی شبیه به یه تابع لگاریتمیه.

یه کلکی که به آدم‌ها می‌زنن تا به دنبال اون هویجی که جلوی‌شون گذاشته شده بدونند، اینه که براشون داستان موفقیت تعریف می‌کنن. می‌گن ببین ایلان ماسک به کجا رسید، چرا ما نرسیم؛ اما واقعیت اینه که چنین چیزی تقریباً غیرممکنه. چطور می‌تونی ایلان ماسک بشی؟ شانس اینکه از نظر آماری تو به ایلان ماسک تبدیل بشی، ۱ در ۸ میلیارد است؛ یعنی عملاً صفر. با این حال، آدم‌ها رو با این ایده گول می‌زنن و تو هم در این مسیر می‌دوی. بعد در شصت سالگی متوجه می‌شی که نه، تو هرگز نمی‌تونستی ماسک بشی.

تنها کاری که تو می‌تونی انجام بدی اینه که خوشحال باشی، همین. خودت خوشحال باشی و کسی که کنارت نشسته هم خوشحال باشه؛ همین کافیه.



جزئیات این موضوع خیلی زیاده و برای درک دقیقش باید چند سال در حوزه هوش مصنوعی مطالعه کرد. ولی اگه بخوام خلاصه بگم، ایده اصلی هوش مصنوعی اینه که سیستمی ایجاد کنیم که بتونه تصمیم‌هایی شبیه به انسان بگیره. در ایده‌های کلاسیک، یه متخصص برنامه‌نویسی با استفاده از ساختارهای مشخص مثل درخت تصمیم‌گیری یا درخت هرس آلفا-بتا، الگوریتمی می‌نوشت که بتونه تصمیماتی شبیه به انسان بگیره. هرچی تصمیم‌ها بیشتر به تصمیمات انسانی نزدیک بودن، اون رو هوشمندتر می‌دونستن. هدف همیشه این بوده که کارهایی که انسان انجام می‌ده رو سیستم‌ها انجام بدن. کم‌کم این ساختارها پیشرفته‌تر شدن و هوش‌ها قوی‌تر شدن، ولی در نهایت، پشت همه این سیستم‌ها یه استنباط هیوریستیک وجود داره. یه ایده اکتشافی هست. مثلاً در خانواده‌ی یادگیری آماری^۱، گفته می‌شه که جامعه بزرگ دروغ نمی‌گه. بنابراین، اگه شما رفتار جامعه رو شبیه‌سازی کنید، می‌تونید پیش‌بینی کنید که رفتارهای آینده چطور خواهد بود. مثلاً اگه ۱۰۰۰ نمونه رو بررسی کنید و ببینید در اون‌ها یه رفتار خاص انجام می‌شه، پیش‌بینی می‌کنید که در نمونه هزار و یکم هم همون رفتار اتفاق می‌افته. خیلی از روش‌های هوش مصنوعی رو از طبیعت الهام گرفتن. مثلاً الگوریتم‌های ژنتیکی به رفتار ژن‌های موجودات نگاه می‌کنن و شبکه‌های عصبی از عملکرد سلول‌های مغزی انسان الهام می‌گیرن. ایده اینه که اگه انسان هوشمند و این هوشمندی در مغزش متمرکز شده

رو همیشه سعی می‌کردم برم. زمان انتخاب رشته ارشد هم پیش ایشون رفتم و گفتم: «رتبه‌ام این شده، کجا برم؟» ایشون به من گفتن که حتماً در هر مقطع تحصیلی یه محیط جدید رو تجربه کنم و از استادها و محیط جدید درس بگیرم. این باعث شد که من برم تهران و در دانشگاه تهران ادامه تحصیل بدم. اونجا هم یه استاد خیلی خوب داشتم که متأسفانه فوت کردن. دکتر کارو لوکس، ایشون هم خیلی باسواد بودن و سبک زندگی‌شون رو خیلی دوست داشتم. با اینکه خیلی باسواد بودن، هیچ‌وقت خودشون رو نمی‌گرفتن. اولین باری که دکتر لوکس رو دیدم، در پله‌ها بود. نمی‌شناختمش، ولی یه پیرمرد خوشحال و مهربون با ریش بلند دیدم که دارن بالا میان. سلام کردم و برخلاف انتظارم خیلی گرم جواب داد. هر وقت در اتاقش رو می‌زدم، ایشون همیشه در اتاق و مشغول خوندن چیزی درون کامپیوترشون بودند. این افراد واقعاً تأثیرگذار در زندگی من بودن. دکتر لوکس هم خیلی آدم بزرگی بود و خیلی زود به خاطر یه بیماری فوت کردن. ولی با وجود بزرگی و سوادشون، همیشه خاکی و مهربون بودن. آدم‌هایی که با وجود موفقیت‌هاشون خودشون رو گم نمی‌کنن، واقعاً ارزش زیادی دارن. این افراد می‌دونن که ارزش واقعی تو اخلاقیات و شخصیت آدم‌هاست، نه چیزایی که دارن.

به عنوان کسی که در زمینه هوش مصنوعی تحقیق می‌کند توضیح بدید این واقعاً به چه معناست؟ چگونه یک قطعه کد می‌تواند دارای هوش و ادراک باشد؟

... به نظرتان شبکه‌های عصبی الهام‌گرفته از مغز انسان، در چه وظایفی بهتر و در چه زمینه‌هایی ضعیف‌تر از مغز عمل می‌کنند؟

تفاوت اصلی شبکه‌های عصبی با مغز انسان در تعداد نورون‌ها و لایه‌هایش هست. ضعف اصلی شبکه‌های عصبی، حتی شبکه‌های عمیق، این هست که تعداد لایه‌هاشون هنوز به تعداد لایه‌های مغز انسان نمی‌رسه. مغز انسان لایه‌های بیشتری داره و هرچی تعداد لایه‌های شبکه عصبی بیشتر باشه، مفاهیمی که می‌تونه تشخیص بده، پیچیده‌تر می‌شن. مثلاً شبکه‌ای با چهار لایه می‌تونه چیزای ساده‌ای مثل خطوط و دایره‌ها رو پیدا کنه، اما شبکه‌ای با ۴۰ لایه می‌تونه تشخیص بده که چیزی که می‌بینه، قسمتی از دم یک گربه‌ست. به عبارتی، می‌تونه مفهوم پیچیده‌تری مثل گربه و حتی قسمتی از گربه رو تشخیص بده. بنابراین، هرچی لایه‌ها بیشتر باشه، توانایی شبکه عصبی برای تشخیص مفاهیم پیچیده بیشتر می‌شه.

مغز انسان به دلیل تعداد زیاد لایه‌ها و اتصال‌های نورون‌ها، از نظر مفهومی پیچیده‌تر بوده و می‌تونه مفاهیم پیچیده‌تری رو تشخیص بده. قدرت شبکه‌های عصبی تو پردازش اطلاعاته. با اینکه لایه‌های کمتری دارن، می‌تونن تعداد زیادی از محاسبات رو در کسری از ثانیه انجام بدن، در حالی که مغز انسان نمی‌تونه این کار رو بکنه. مثلاً چشم انسان بیشتر از ۳۰ یا ۶۰ فریم در ثانیه رو نمی‌تونه ببینه. حتی اگه ۱۲۰ فریم در ثانیه هم بهش نشون داده بشه، براش فرقی نمی‌کنه. ولی شبکه‌های عصبی

که شامل نورون‌های عصبیه، پس اگه بتونیم ساختاری شبیه نورون‌های عصبی ایجاد کنیم، که همون شبکه عصبیه، ممکنه به هوش دست پیدا کنیم. بعد این ساختارها رو طراحی و اجرا می‌کنن. ایده‌ها معمولاً به این شکله: انسان‌ها یا مکانیسم‌های طبیعی موفق رو بررسی می‌کنیم و از اونا الهام می‌گیریم تا یه سیستم هوشمند طراحی کنیم. بنابراین، الگوریتم‌های هوشمند باید یه ایده اولیه داشته باشن که توضیح بده چطور می‌تونیم مفهوم هوش رو به کد تبدیل کنیم. بعد از اون، برای این ایده یه ساختار^۱ طراحی می‌شه. مثلاً تو شبکه‌های عصبی، یه تابع مثل WX تعریف می‌شه یا در الگوریتم‌های ژنتیکی، یه ژن برای اون ساخته می‌شه و اینا با هم ترکیب می‌شن. حالا چطور یه کد می‌تونه هوشمند بشه؟ اگه کدی صرفاً تصادفی^۲ باشه، نمی‌تونه هوشمند باشه. ولی اگه یک ساختار مشخص و هدفمند به اون اضافه بشه، می‌شه گفت که این کد هوشمنده. هوشمند بودن یعنی چی؟ یعنی کدی که بتونه رفتاری که براش تعریف شده رو دوباره تکرار کنه. مثلاً در هیوریستیک آماری، رفتارهایی که از هزار نفر مشاهده شده، دوباره تکرار می‌شه. اگه این رفتار در نفر هزار و یکم هم تکرار بشه، می‌گیم سیستم هوشمند عمل کرده. فرض کنید شما رانندگی یاد می‌گیرید. رفتارهایی که در ۱۰۰ هزار کیلومتر رانندگی قبلی آموختید، اگه در کیلومتر بعدی هم همون رفتارها رو تکرار کنید، به نظر می‌رسه که دارید کار هوشمندانه‌ای انجام می‌دید.

۱. Structur

۲. Random



به پیچیدگی مفاهیم مغز انسان برس.

چه پیشنهادی برای دانشجویانی دارید که می‌خواهند کار ریسرچ را شروع کنند؟ در واقع چه گپ‌ها و موضوعات بکری وجود دارد که دانشجو‌ها هم بتوانند در آن زمینه فعالیت کنند؟

به نظرم این بیشتر به درد دانشجویان کارشناسی می‌خورد که به تحقیق و پژوهش علاقه دارند. خوشبختانه کلاس‌های تحصیلات تکمیلی معمولاً روزهای آخر هفته هست که دانشجویان کارشناسی کلاس ندارند. من تقریباً به هرکدام از دانشجویانی که به سراغم میان و می‌گن می‌خواهیم کاری انجام بدیم، این جواب رو می‌دم.

حتماً نیاز دارید که مفاهیم پیشرفته علم رو یاد بگیرید و راه‌حلش این هست که به کلاس‌های آکادمیک تحصیلات تکمیلی برید. من به کلاس‌های حضوری اعتقاد زیادی دارم چون تعامل‌پذیر هستن. مثلاً اگر استاد مطلبی رو بگه که به نظرتون اشتباه باشه، می‌تونید بپرسید چرا این رو می‌گه. کلاس‌های آنلاین یوتیوب و آپارات هم خوب هستن، ولی تعامل‌پذیر نیستن.

فکر نمی‌کنم هیچ‌کدام از اساتید از اینکه شما در کلاس‌های آنها حضور پیدا کنید و بگید که می‌خواهید گوش کنید، ناراحت بشن. معمولاً به ندرت پیش میاد که استادی بگه: نه! اجازه نداری تو کلاس من گوش بدی.

به نظرم اگه می‌خواهید کار تحقیقاتی کنید، یک یا دو سال نیاز به تحصیلات آکادمیک دارید و می‌تونید این تحصیلات

می‌تونن میلیون‌ها داده رو تو هر ثانیه پردازش کنن. بنابراین، نقطه قوت شبکه‌های عصبی تو سرعت و پردازش موازی هست.

با این حال، مغز انسان هنوز در زمینه‌هایی مثل پیدا کردن مفاهیم پیچیده جلوتره. فعلاً می‌گیم شبکه‌های عصبی خلاقیت ندارند. مثلاً در شبکه‌های دیپ‌فیک^۱، می‌تونن عکس‌های جعلی از انسان‌ها بسازن، مثلاً صورتی رو روی بدن دیگه‌ای مونتاژ کنن. ولی ممکنه خطاهایی رخ بده، مثلاً تصویری تولید بشه که انسان سه دست داشته باشه، چون شبکه نمی‌دونه که انسان نباید سه دست داشته باشه.

در مدل‌های زبانی مثل GPT هم، با اینکه جمله‌ها رو خوب می‌سازن، ولی باگ‌های مفهومی دارن. مثلاً ممکنه ارتباط مفهومی بین چیزهایی که می‌گه وجود نداشته باشه. بنابراین، ضعف اصلی هوش مصنوعی فعلاً درک مفاهیم پیچیده هست، که ما بهش می‌گیم خلاقیت! شاید اگر تعداد لایه‌ها و توان پردازش شبکه‌های عصبی بیشتر بشه، بتونن به این خلاقیت دست پیدا کنن.

عملکرد نوروها و سلول‌های شبکه عصبی تقریباً مشابه هم هست، ولی تفاوت‌هایی در ساختار و ساده‌سازی‌هایی که انسان‌ها در طراحی شبکه‌های عصبی انجام دادن، وجود داره. مثلاً، در شبکه‌های عصبی از توابعی مثل سیگموئید به عنوان تابع فعال‌سازی استفاده می‌شه، ولی مغز انسان دقیقاً از سیگموئید استفاده نمی‌کنه. این تفاوت‌های جزئی وجود داره، ولی تفاوت اصلی همون تعداد لایه‌ها هست که باعث می‌شه شبکه‌های عصبی هنوز نتونن

آمار و احتمالات مهندسی رو می‌خوندید، بعد که مسلط می‌شدید می‌رفتید سراغ هوش مصنوعی. این روش مربوط به زمانی بود که پیشرفت حوزه‌ها این قدر سریع نبود. اینکه همه چیز رو بخونید خوبه، چون علاوه بر دانش دید بهتون می‌ده، ولی در شرایط فعلی امکانش وجود نداره.

مطالعه ساختاری دو تا جنبه داره: اول، یادگیری دانش؛ و دوم، کاربرد دانش. متأسفانه، افراد اغلب به جنبه دوم توجهی نمی‌کنن. برای مثال، وقتی کتاب آمار و احتمالات مهندسی می‌خونید، معمولاً مدل‌ها رو حفظ می‌کنید؛ می‌دونید توزیع نرمال چیه، یا احتمال شرطی چطوری عمل می‌کنه. اما کمتر کسی به این فکر می‌کنه که احتمال شرطی کجا کاربرد داره یا چرا به چنین مدلی نیاز داشتیم. آیا در زندگی روزمره جایی هست که از توزیع نرمال استفاده کنید؟ این نوع از درک عمیق و توجه به کاربردها معمولاً با مطالعه وسیع و کل‌نگر حاصل می‌شه. مثلاً من در درس معماری کامپیوتر، خیلی اصرار دارم بگم چرا مانو، منظور نویسنده کتاب معماری کامپیوتر، این طراحی رو انجام داد. اینکه چرا ۲۵ تا دستور گذاشته و نه ۳۰ تا؟ چرا این طور انتخاب کرده و طور دیگری انتخاب نکرده؟

شما باید یه هرم رو تصور کنید که از مفاهیم پایه‌ای پایین شروع می‌شه و به مباحث تخصصی بالای اون می‌رسه. حالا که وقت اون نیست که از پایین‌ترین سطح هرم مفاهیم شروع کنید و به آرومی پیش برید؛ پیشنهاد من اینه که از میانه هرم شروع کنید، یعنی نه از اصول کاملاً پایه‌ای و نه از موضوعات کاملاً تخصصی.

رو در زمان خودش یعنی دوره ارشد انجام بدید یا اگر علاقمندید زودتر شروع کنید.

آیا موضوعی را پیشنهاد می‌کنید که دانشجویان حول محور آن کار کنند؟

الان بیشتر کارها با ابزار دیپ لرنینگ^۱ انجام می‌شه. اما من اصلاً دوست ندارم که موضوعات به جهت خاصی محدود بشن. مثلاً یکی دوست داره در حوزه شبکه‌های اجتماعی کار کنه، یکی دیگه در حوزه مالی. اصلاً تنوع قشنگه. سایتی به اسم Papers With Code وجود داره که State of the Art رو خوب دسته‌بندی کرده.

ایده اصلی این سایت اینه که مقالاتی قرار بده که کدهاشون هم وجود دارن. این سایت دسته‌بندی‌های خوبی ارائه می‌ده. اما اگه بخوام موضوع خاصی رو پیشنهاد بدم، واقعاً دوست ندارم همه به یک نقطه متمرکز بشن. این سایت کلمات کلیدی خوبی در هر زمینه پیشنهاد می‌ده و می‌تونید با استفاده از اونا و کنار هم قرار دادن شون، شروع به تحقیق کنید.

برای ورود به حوزه هوش مصنوعی، آیا بهتر است دانشجویان ابتدا ریاضیات و مفاهیم پایه را یاد بگیرند یا مستقیماً با کدنویسی شروع کنند و در حین مسیر، مفاهیم مرتبط را فرا بگیرند؟ نقشه راه پیشنهادی شما چیست؟

به نظرم راه‌حل نه کاملاً اینه و نه اون؛ بلکه جایی میان این دو قرار داره. روش قدیمی این‌جوری بود که شما اول کل یک ساختار رو یاد می‌گرفتید، مثلاً همه مباحث



گزینه‌ها رو ج زدم. نتیجه این شد که ۱۳ درصد معماری رو درست زدم، در حالی که اگه تصادفی جواب می‌دادم، باید صفر درصد می‌زدم. این تجربه نشون داد که مدل ذهنی طراح سؤال رو اگه بشناسید، می‌تونید بهتر عمل کنید. می‌خوام بگم، سؤال کردن پشت هر چیزی می‌تونه کلی دانش ایجاد کنه، چون در نهایت، جهانی که در آن زندگی می‌کنیم، توسط انسان‌ها ارزیابی می‌شه. ما در حال برآورد و ارزیابی با مدل‌های آماری هستیم. انسان‌ها معمولاً گرایش و تمایل به یه سمت خاص دارن. این امر باعث می‌شه که اصل locality در جهان به وجود بیاد. فرض کنید که انسان‌هایی که به هم نزدیک هستن، هم‌سطح و هم‌فکرن. این یه گرایش locality که فرض رندوم بودن رو نقض می‌کنه. نتیجه این هست که اگر شما در دانشگاهی نمونه‌گیری کنید، این نمونه با نمونه‌ای از شیراز یا با نمونه‌ای از ایران یا حتی جهان فرق داره. پشت این موضوع یه قانون locality وجود داره که تو خیلی از قسمت‌های زندگی قابل مشاهده است. اگر نگاهی به چراهای مسائل داشته باشیم، می‌تونیم تحلیل‌های خوبی از آن استخراج کنیم. حتی شاید بتونیم تمریناتی برای این تحلیل‌ها در نظر بگیریم، اما نمی‌دونم که این تحلیل‌ها به راحتی تغییر می‌کنن یا نه!

این روزها خیلی‌ها، فکر می‌کنند که دانشگاه دیگر کاربرد ندارد و با دوره‌های آنلاین سریع‌تر میتوان به اطلاعات به‌روز دسترسی داشت. به نظر شما دانشگاه در حال از دست دادن جایگاه و اعتبار خودش است؟

اول مفاهیم اولیه و کلیدی رو پیدا کنید و آنها رو یاد بگیرید و هر وقت نیاز به عمق بیشتر داشتید، می‌تونید به مبانی پایه‌ای مثل آمار و احتمالات برگردید و مفهومی که نیاز دارید رو دوباره مرور کنید. برای ورود به حوزه هوش مصنوعی، خوندن سه تا شش درس در سطح کارشناسی ارشد باعث میشه تا دانش اولیه مناسبی به دست بیارید. این مسیر بهتون کمک می‌کنه تا بدون نیاز به مطالعه همه مفاهیم پایه‌ای، کارتون رو پیش ببرید. این مسیر استعداد مخصوص به خودش رو می‌خواد و به نظرم افرادی تو این کار خیلی استثنا می‌شن که به همه چیز گیر می‌دن و هیچ مطلبی رو بدون دلیل و استدلال نمی‌پذیرن. مثلاً اگه من بگم این مدل یادگیری ماشین فلان کار رو انجام می‌ده، باید پیش خودتون بپرسید: چرا این کار رو انجام می‌ده؟ کدوم قسمت مدل باعث این نتیجه شده؟ اگه مدل رو دستکاری کنم چی می‌شه؟ این نوع سؤال‌ها خیلی مهمه. یه تجربه شخصی‌ام رو در مورد کنکور ارشد بگم. وقتی کنکور می‌دادم، به این فکر کردم که طراح سؤال یه انسانه و جواب درست رو می‌دونه. معمولاً طراحان دوست دارن جواب درست رو در گزینه‌ای مخفی کنن که راحت پیدا نشه. برخلاف اینکه می‌گن احتمال درست بودن هر گزینه در تست‌های چهارگزینه‌ای برابره، در عمل این‌طور نیست. من از کنکور سال‌های قبل نمونه‌گیری کردم و دیدم که مثلاً در بعضی درس‌ها، جواب‌های صحیح بیشتر در گزینه ج قرار دارن. وقتی این الگو رو پیدا کردم، برای درس سختی مثل معماری کامپیوتر که ضریبش برای گرایش من صفر بود، تمام

تکنولوژی‌ها اهمیت می‌دن از نظر من یا اشتباه می‌کنن، یا اینکه لایه تفکر و آنالیز و سیستم‌های پیچیده‌شون رو به افراد باکیفیت می‌سپارند و صرفاً نوشتن تابعی که آن لایه نتیجه گرفتن رو به تو می‌سپارن. حالا اگه بخوایم به سوال اصلی برگردیم، دلیل اینکه چرا تحصیل آکادمیک دیگه برای خیلی‌ها مفید به نظر نمی‌رسه اینه که آکادمیک و بیزینس دو حوزه متفاوت هستن. در آکادمیک باید دانش و مدل‌سازی پشت علم کامپیوتر رو یاد بگیری. مثلاً وقتی در دانشگاه کدنویسی یاد می‌دن، هدف فقط یاد دادن نوشتن حلقه for یا while نیست؛ بلکه باید بفهمی تفاوت این دو چیه. در کلاس‌های دیگه، استاد ممکنه ازت بخواد یه حلقه بنویسی که اعداد یک تا ده، اعداد اول، یا اعداد کامل رو چاپ کنه. این کار رو می‌کنی، ولی مهم‌تر از اینکه فقط حلقه رو نوشته باشی اینه که درک کنی چرا این حلقه به این شکل کار می‌کنه یا چرا برای حل مسئله از for به جای while استفاده نشده. برای حل یه مسئله خاص، نیازی به تحصیلات آکادمیک نداری. می‌تونی با شرکت در دوره‌های آنلاین، دیدن ویدیوهای یوتیوب، یا مراجعه به منابعی مثل Stack Overflow خیلی سریع‌تر از چهار سال برنامه نویسی یاد بگیری. مثلاً اگه هدف این باشه که فقط یه تابع ساده بنویسی که دو ورودی بگیره و خروجی بده، می‌تونی این مهارت رو ظرف یه سال یاد بگیری. اما اگه بخوای به سطح تحلیل و طراحی برسی و بفهمی چرا ساختارهای برنامه‌نویسی به شکلی که هستن طراحی شدن، اون وقت نیاز به آموزش آکادمیک داری.

در دنیای امروز که جریان داده‌ها سریع پیشرفت می‌کنه، این تغییرات اتفاق می‌افته. این روند بعد از اومدن شبکه‌ها و افزایش ارتباطات بین آدم‌ها شدت گرفت. وقتی ارتباطات زیاد شد، حجم زیادی از داده‌ها در اختیار همه قرار گرفت و این باعث تغییر ساختار شد. حالا که دسترسی به اطلاعات سریع‌تر و راحت‌تر شده، موافقم با اینکه راحت‌تر و با هزینه کمتر، می‌تونی مهارت‌های مورد نیازت رو یاد بگیری. تقسیم‌بندی که من برای این موضوع دارم، که لزوماً درست نیست، این هست که برای حل یه مسئله دو نوع تخصص نیاز داری. یکی اینکه مسئله خاص رو چطور حل کنی، یعنی یک مسئله با چه روش‌هایی حل می‌شه. نوع دیگه تخصص اینه که پشت مسئله رو تحلیل کنی و از اتفاقات جانبی جلوگیری کنی و مسائل مشابه رو هم حل کنی. یعنی اینکه در حین حل مسئله، بتونی جوانب دیگه اون رو هم در نظر بگیری. خب، تخصص دوم به‌طور واضح گران‌تر و کمیاب‌تره. آموزش‌های آنلاین مثل دوره‌های یوتیوب یا کلاس‌های موسسات و از آن واضح‌تر کلاس‌های کنکوری، برای خیلی از مسائل راه‌حل‌های کنکوری دارن، یعنی می‌تونی مسئله‌ای خاص رو حل کنی، بدون اینکه بدونی این مسئله چی بوده. اینکه در برنامه‌نویسی ممکنه با Stack Overflow و کپی کردن کدهای آماده، باگی رو حل کنی یا در نهایت تغییراتی بدی، ولی چیزی که در نهایت داری یه چیز وصله پینه‌ای شده است و اگر این مسئله در شرایط متفاوتی قرار بگیره، ممکن هست به‌درستی کار نکنه. شرکت‌هایی که به این نوع



می‌ده. این رفتارها هم آموزشی هستن و کمک می‌کنن مسیر زندگی‌ات رو راحت‌تر پیش ببری. می‌تونی ببینی که اگه چهل یا پنجاه سال مثل این استاد زندگی کنی، به کجا می‌رسی، چه چیزهایی رو باید ازش یاد بگیری و چه مواردی رو کنار بذاری. نگاه دانشگاهی یه دید استنتاجی و کلاسیکه. قرار هست مفاهیم رو یاد بگیری. ولی در دوره‌های آموزشی خصوصی، معمولاً این مفاهیم رو آموزش نمی‌دن. این موسسات ساختار رو آموزش نمی‌دن و دلیلش هم واضحه؛ هدف اصلی آنها کسب درآمد هست. موسسه‌ها می‌خوان دانشجویا خوشحال باشن تا ترم‌های بعد هم ثبت‌نام کنن. دانشجو زمانی خوشحال می‌شه که حس کنه چیزهایی رو یاد می‌گیره؛ پس سخت‌گیری نمی‌کنن و فقط چند مثال ساده میارن که دانشجو احساس کنه همه چیز رو یاد گرفته. من زمانی مدیریت صنعتی درس می‌دادم و به من گفته می‌شد که زیاد سخت نگیرم تا دانشجویا راضی باشن. مثلاً اگه درس شبکه می‌دادم، فقط چند مثال ساده از شبکه رو می‌گفتم تا دانشجو لذت ببره و احساس کنه شبکه رو یاد گرفته. ولی در دانشگاه، من درباره مدل پشت شبکه‌های کامپیوتری صحبت می‌کنم. ممکنه فقط درصد کمی از دانشجویا این مباحث رو بفهمن، ولی هدف من همون درصد هست که قراره طراح و تحلیل‌گر شن. به طور کلی، آکادمیک مفاهیم انتزاعی و بنیادی رو آموزش می‌ده، در حالی که حوزه بیزینس بیشتر به دنبال آموزش کاربردی و عملی هست. هر دو لازم هستن، ولی قابل مقایسه نیستن. یه فرد کاملاً آکادمیک

به نظرم کسی که به دانشگاه می‌ره، برای یاد گرفتن همین طرز تفکره. من به دانشجویا می‌گم در کنار دانشگاه دوره‌های عملی هم بگذرونن، چون ترکیب این دو مهارت در چهار سال دانشگاه به‌تنهایی به دست نمی‌اد. تو در دانشگاه قرار هست مفاهیمی رو یاد بگیری که کمک می‌کنن هر مسئله‌ای رو حل کنی، نه اینکه فقط یه مسئله خاص رو بلد باشی. یه فارغ‌التحصیل دانشگاهی باید توانایی حل مسائل رو داشته باشه. اگه بهش بگن به جای سی، با جاوا یا پایتون کدنویسی کن، نباید بگه: بلد نیستم. باید بتونه مفاهیم رو به زبان‌های مختلف تعمیم بده. برنامه‌نویس کسی نیست که فقط یاد گرفته یه بار حلقه‌ای برای چاپ اعداد یک تا ده بنویسه و حالا اگه ازش بخوان یک تا ۱۵ رو چاپ کنه، نتونه. تو در دانشگاه، یه مهارت یاد می‌گیری؛ البته اگه واقعاً بخوای یاد بگیری! حالا کیفیت اساتید و محیط آموزشی هم تأثیرگذار هست. همیشه در دانشگاه‌ها درصدی از اساتید، مثلاً ۲۰ تا ۳۰ درصد، مفیدن. در دانشگاه تهران که من درس خوندم، همین نسبت رو دیدم. حدود ۲۰ تا ۳۰ درصد اساتیدا برای من مفید بودن، تو دانشگاه شیراز هم همین‌طور بود. این وضعیت در بیشتر دانشگاه‌ها وجود داره. اما همین ۲۰ تا ۳۰ درصد کافی هستن. قرار هست از این اساتیدا تجربه چهل یا پنجاه ساله‌ای که به صورت چکیده بهت منتقل می‌شه رو یاد بگیری. این تجربه هم عملیه و هم رفتاری. مثلاً می‌بینی استاد چطور یه مسئله رو حل می‌کنه یا وقتی ازش سوالی می‌پرسی، چطور جواب

مشکل از ساختار داده‌ها ناشی می‌شه؛ انقدر داده‌های زیادی در اختیار آدم‌ها قرار گرفته که افراد نسل جدید بمباران اطلاعاتی می‌شن. این افرادی که شما بهشون اشاره می‌کنید، از پنج سالگی‌شون بمباران اطلاعاتی شدن. مثلاً اگر من از سی سالگی بمباران اطلاعاتی شدم، ولی این نسل از ده سالگی تبلت دستشون بوده و در انواع و اقسام شبکه‌های اجتماعی بمباران اطلاعاتی شدن. این باعث می‌شه که آدم‌ها نتونن متمرکز بشن. مثلاً یکی از چیزایی که حواس رو به راحتی پرت می‌کنه، اینستاگرام هست. موضوع هر پستی با پست بعدی کاملاً متفاوت هست و همین باعث می‌شه که نتونید روی چیزی تمرکز کنید. مثلاً یه خبر درباره جنگ می‌خونید و بهش فکر نمی‌کنید و این به سرعت به نحوه ساختن یه توپ پلاستیکی تغییر پیدا می‌کنه و بعدش هم فکر می‌کنید ظهر چی بخورم؟ بمباران اطلاعاتی باعث می‌شه که تمرکز کاهش پیدا کنه.

این موضوع تقصیر هیچ‌کس نیست؛ این روند تکامل بشریته و همه‌جا اتفاق می‌افته. با بمباران اطلاعاتی آدم‌ها از حالت متمرکز به حالت وسیع ولی سطحی تبدیل می‌شن. ایراد دیگه‌ای که در دانشجویان جدید می‌بینم، اینه که نظام آموزشی هم تست‌محور شده و همه چیز رو به صورت تستی حل می‌کنن. مثلاً در دبیرستان دیگه مهم نیست که فرمول یه مسئله جبری چیه و از کجا اومده. اگه یه مسئله جبری بهتون بدن، می‌گن این‌جوری جوابش می‌شه، بدون اینکه به مدل پشتش کاری داشته باشن. این باعث می‌شه که افراد تستی بالا بیان و مفهومی

برای بعضی کارها مناسب نیست، و یه فرد کاملاً عملی هم فقط برای کارهای خاص مفیده. ولی کسی که در هر دو حوزه تجربه داشته باشه، مثل یه الماس ارزشمند هست. چنین فردی هم مسائل رو درک می‌کنه و هم می‌تونه اونا رو حل کنه. سیستم آموزشی آکادمیک مثل جستجوی طلاست. اگه در محیطی باشی که درصد بیشتری از استادها مفید باشن، شانس یادگیری هم بالاتر می‌ره. مثلاً اگه در دانشگاهی باشی که به جای ۲۰ تا ۳۰ درصد، ۷۰ تا ۸۰ درصد از استادها خوب باشن، احتمالاً بیشتر یاد می‌گیری. مشکل ما معمولاً سرفصل‌ها نیست؛ سرفصل‌ها اغلب خوبن، ولی مسئله اینه که استاد توانایی آموزش اونا رو داره یا نه. آیا استاد خودش مطلب رو بلده و رو به شکل منظم و قابل فهم ارائه می‌ده. تفاوت دانشمند و استاد همین‌جاست. دانشمند مطلب رو بلده، ولی استاد باید بتونه اون رو به یه مطلب منطقی و قابل فهم تبدیل کنه. خیلی از استادها دانشمند خوبی هستن، ولی معلم خوبی نیستن. به همین دلیل، فقط ۲۰ تا ۳۰ درصد استادها می‌تونن واقعاً موثر باشن. این افراد رو هم معمولاً خارج از محیط آکادمیک پیدا نمی‌کنی. اگه سوال این باشه که آیا دانشگاه چهار ساله مفیده؟ جواب اینه که به تنهایی کافی نیست. در کنار دانشگاه باید مهارت‌های عملی رو هم یاد بگیری. اما آیا جایگزینی دانشگاه با آموزش‌های آنلاین هم ممکن نیست.

**نظرتان در مورد نسل Z چیست؟
چه تفاوت‌هایی با نسل‌های
قبل دارن؟ تا به حال به چالش
یا مشکل‌های خاصی برخوردید؟**



که دوست دارن چیزی یاد بگیرن. این انگیزه اصلی برای منه. جذابیت دانشگاه برای من این هست که اگر در یک سال، ۵۰ تا ۱۰۰ تا دانشجو داشته باشم، از این تعداد پنج نفر هم خوشحال‌تر از دانشگاه بیرون برن، برای من ارزش داره. حتی در مکانیزم‌های غلط آموزش و بمباران اطلاعاتی هم هنوز دانشجویان خوبی هستن.

سبک بازی عوض شده و نمی‌تونیم بگیریم باید مثل قدیم باشه. هنوز خیلی‌ها برای حل مسئله و مهارت‌هاشون تلاش می‌کنن و به نظرم کیفیتشون خوبه. اینکه شبیه به من فکر نمی‌کنن، به معنای این نیست که کیفیتشون ایراد داره. من با افراد ۱۵، ۲۰ سال کوچک‌تر از خودم کلی تعامل دارم و از اینکه این حجم دیتا رو دیدن، خوشم میاد.

یادمه که مامان و مامان‌بزرگ‌هامون هم می‌گفتن نسل جدید از دور خارج شده و دیگه درست نیستن.

هر نسلی تغییرات خودش رو داره. دنیا همینه و هر ۱۰ سال در یه دوره‌ای زندگی می‌کنیم که اتفاقات جذاب خودش رو داره.

ما یک همه‌پرسی از دانشجویان در فضای مجازی ترتیب دادیم و اغلب متن‌های فرستاده حول محور تدریس خوب و ساختار بندی شده شما بود که غالباً هر ترم یک ساختار را دنبال میکنند و از تسلط شما قدرانی کردند

من ۲۰-۱۵ ساله که دارم معماری درس می‌دم، یعنی معماری رو کامل می‌دونم چیه. به نظرم اینکه هر ترم مثل ترم قبل باشه خوب نیست، مگر اینکه علم

یاد بگیرن. مکانیزم آموزشی دانشگاه‌ها هم به همین سمت می‌ره؛ چون از ابتدا همه‌چیز رو به صورت حفظی یاد می‌دن.

بمباران اطلاعاتی و آموزش تستی باعث می‌شه که افراد نتونن متمرکز بشن و مسئله حل کنن. این دلیل اصلی که خیلی از دانشجویان جدید توانایی حل مسئله ندارن. می‌گن دانشجو‌ها بی‌سواد و ناتوان شدن، ولی این‌طور نیست. از نظر ژنتیکی، آدم‌ها طی ۱۰۰ یا ۱۰۰۰ سال ناتوان نمی‌شن. ضریب هوشی نمی‌تونه به این سرعت تغییر کنه. احتمالاً ضریب هوشی ما با ضریب هوشی انسانهای غارنشین خیلی فرقی نمی‌کنه. فقط آن‌ها اطلاعات در اختیارشون نبوده و از بچگی تعلیم ندیده بودن که در سی سالگی مهندس آی‌تی بشن. شما از روز اولی که به دنیا می‌آید، چراغ بالای سرتون روشنه و ادیسون همه‌چیز رو قبلاً اختراع کرده. شما دیگه نیازی نیست چرخ یا برق رو از اول اختراع کنید. این تفاوت اصلی هست.

برای حل مسئله، ما دانش عمقی می‌خوایم. نسل جدید دانش عمقی بهشون یاد داده نشده و مکانیزم آموزشی‌شون باگ داره. من الان در این ساختار دیگه لذتی نمی‌برم از اینکه کسی سر کلاس بیاد و ته کلاس ساندویچ بخوره یا حرف بزنه. وقتی دستشون رو بالا می‌برن، تنها سوالشون اینه که این مطلب تو امتحان میاد یا نه؟ یا خیلی بخواد هوشمندانه‌تر بپرسه، می‌گه: چه جوری از این سوال می‌دید؟ یعنی دنبال مدل‌سازی مغز من هم تا حدودی هستن!

با این حال، همین ترمی که دارم درس می‌دم، مثلاً سه چهار تا دانشجو هستن

شبکه عصبی یکی از همون جرقه‌هایی بود که اتفاق افتاد. وقتی که شبکه عصبی کانولوشنی^۴ اومد و یه فیلتر لوکال ساختن، ضعف شبکه عصبی رو پوشوندن و بعد شبکه عصبی تونست دستش باز بشه، شروع به یادگیری کنه و وارد یادگیری عمیق^۵ و شاخه‌هاش بشه. عملاً طی ۱۰ سال اخیر مهم‌ترین تغییر در علم کامپیوتر و هوش مصنوعی، همین شبکه‌های عمیق بوده. بقیه موارد کلاسیک موندن؛ تغییرات‌شون فقط در حد بهبودهای جزئی بوده، اونم در بازه‌های ۱۰ ساله. تازه این وضعیت علمی مثل کامپیوتر که سرعت تغییراتش خیلی زیاده، رخ داده. احتمالاً درمورد علوم دیگه، سرعت تغییرات از این هم کمتره. حتی برای رشته‌های مهندسی، سرعت تغییرات کمتر از اینه.

چرا در کلاس هایتان، خصوصاً کلاس معماری کامپیوتر تعامل و ارتباط کمی با دانشجویان دارید؟

باید بگم وسواس منه. کلاً در مورد اینکه سرفصل رو باید تموم کنم وسواس دارم. مخصوصاً درمورد درس‌هایی که در کنکور ارشد میاد. خودم خاطره خوبی از درس‌های دوره لیسانسم ندارم و برای ارشد مجبور شدم همه کتاب‌های مرجع رو از اول بخونم تا تازه بفهمم این درس چیه. انگار یه تجربه بد از اون زمان دارم. به همین دلیل، میشه گفت نوعی فوبیا از درس‌هایی که سرفصل‌هاشون نصفه‌ونیمه تدریس می‌شه دارم.

نه فقط معماری کامپیوتر، در همه درس‌هام همین رویه رو دارم. مثلاً درس معماری،

تو فاصله بین ترم قبل و ترم فعلی خیلی تغییر نکرده باشه، یا مثلاً در یک بازه پنج ساله هم تغییری نکرده باشه. برای درس معماری کامپیوتر رفرنسی که برای این درس داریم، موریس مانو، ۱۵-۱۰ ساله که تغییری نکرده. اون چیزی که می‌خواد درباره سخت‌افزار به دانشجو یاد بده، همون هست و پایه و اساسش تغییری نکرده و همچنان می‌شه طی یک ترم اونو یاد گرفت. بله، تکنولوژی‌های ساخت سخت‌افزار قطعاً دارن تغییر می‌کنن، که ما در این مورد بحثی نداریم، ولی در همین حد کسی که می‌خواد فردا کدنویسی کنه باید بدونه دستور اسمبلی قرار هست چه شکلی اجرا بشه یا دیدی داشته باشه از این که چرا وقتی برق به کامپیوتر وصل می‌شه برنامه اجرا می‌شه. این ایده‌ها و مفاهیم، واقعاً تو ۳۰، ۲۰ یا حتی ۴۰ سال گذشته تغییر نکردن. البته برای درس‌هایی که ماهیتاً داینامیک‌تر هستن، این ثابت موندن چندان مطلوب نیست. بیشتر درس‌های آکادمیکی که درباره‌شون صحبت می‌کنیم، معمولاً پایه‌هایی دارن که چند دهه قدمت دارن. مثلاً در دنیای کامپیوتر، تنها چیزی که می‌تونم بگم در ۲۰-۱۰ سال اخیر واقعاً جابجا شده، شبکه‌های عصبیه. این مهم‌ترین چیزیه که می‌تونم بگم. یا مباحثی مثل فازی، هنوز همون قصه پنجاه سال پیش رو دارن. یادگیری آماری، همه رگرسیون‌ها، کلاسیفایرها^۱، ”الگوریتم نزدیک‌ترین همسایه“^۲ همه این‌ها همون داستان ۴۰-۳۰ سال پیش هستن. کاربردشون ممکنه یه کم تغییر کرده باشه، ولی مفهومشون همون اندازه قبلی باقی مونده. فقط تنظیم^۳ شدن.



غذا خوردن و تشنه شدن، اما غلام هیچ واکنشی نشون نداد. کم‌کم مهمونا شروع به اعتراض کردن و یکی از اونا خودش بلند شد و رفت آب بیاره. غلام همون موقع گفت: این شخص تشنه واقعی بود.

به نظرم، تشنه واقعی خودش دنبال یادگیری میاد. این پروژه‌های کلاسی برای تشنه‌ها نیست، بلکه درصد زیادی از غیرتشنه‌ها رو جذب می‌کنه. اون دانشجویی که واقعاً علاقه‌منده، خودش میاد و می‌پرسه: می‌خوام فلان کار رو انجام بدم، چیکار کنم. منم راهنمایی می‌کنم.

پس می‌شه گفت دو عامل اصلی در این موضوع تأثیر دارن: یکی وسواس من برای تموم کردن سرفصل‌ها و یکی اینکه رویکرد من بیشتر پیدا کردن تشنه واقعیه. این روش ممکنه خوب یا بد باشه، ولی برای من جواب می‌ده.

آیا روش تدریس شما برگرفته از تجربه‌ها یا استادان خاصی است؟

من در روش تدریس الگوریتمی و معلم بودنم سعی کردم شبیه همون استاد دوره کارشناسی‌ام، دکتر توحیدی، باشم. به نظرم او ساختار خیلی خوبی در تدریس داشت؛ نیاز به مسئله رو اول ایجاد می‌کرد و بعد مسئله رو حل می‌کرد. یعنی مثلاً نیاز به حلقه for رو اول توضیح می‌داد و بعد کم‌کم اهمیتش رو نشون می‌داد. مثلاً اگه می‌خواست کاری کنه که پنج بار پرینت بگیره، پنج بار پرینت می‌گرفت تا دانشجوها خسته بشن و بپرسن چرا این کار رو می‌کنه. بعد توضیح می‌داد که برای جلوگیری از این کار نیاز به حلقه for

ترم‌های اول که درس می‌دادم، پنج تا شش فصل رو تدریس می‌کردم، اما حالا به سختی به چهار فصل می‌رسم. بعضی وقت‌ها حتی چهار فصل هم نمی‌تونم تموم کنم. این وسواس باعث می‌شه دیگه وقتی برای حاشیه رفتن نداشته باشم.

فکر می‌کنم مشکل اصلی همین فوبیاست. شاید هم هیچ اتفاق خاصی نیفته اگه دانشجو برای کنکور ارشد خودش بشینه بخونه، ولی من همچنان احساس می‌کنم که باید در کلاس، درس رو کامل یاد بگیره. این وسواس باعث می‌شه برخی فرصت‌ها رو از دست بدم، مثلاً همین تعامل‌هایی که می‌تونستم در کلاس داشته باشم.

در کل، اون دسته از دانشجو‌هایی که هدفشون یادگیریه، خودشون سراغ من میان. معمولاً بعد از کلاس میان و سؤالات و ایده‌هاشون رو مطرح می‌کنن.

اما اگه بخوام مثلاً تعامل رو به صورت تعریف پروژه تعریف کنم، فکر می‌کنم این روش به حاشیه می‌ره. خیلی از دانشجویان زرنج با کپی کردن پروژه ممکنه فقط برای گرفتن نمره یا گذروندن وقت این پروژه‌ها رو انجام بدن. این دیگه چیزی نیست که مدنظر منه.

بذارید یه حکایت براتون تعریف کنم: یه نفر می‌خواست یه غلام بخره. غلامی خیلی گران‌قیمت دید که صاحبش می‌گفت این غلام مهارت خاصی داره؛ تشنه‌شناسه! هر کسی که تشنه باشه، غلام متوجه می‌شه. خریدار غلام رو خرید و برای امتحان مهارتش، مهمونی بزرگی ترتیب داد. تو مهمونی غذای شور درست کردن و آب هم سر سفره نگذاشتن. مهمونا

نمی‌کنه. آینده هم که پیش‌بینی‌ناپذیره و الان دیدیم یه مدل خیلی پیچیده روش سواره که ما نمی‌تونیم مدلش کنیم.

پس یک هفته آینده، یک ماه آینده چی خوشحالت می‌کنه؟ یه موقع با حل مسئله خوشحالی، یه موقع با پول درآوردن، با هر چی خوشحالی. به نظر من باید زندگی رو سه قسمت کرد، درصدی برای آینده ذخیره کن، یه درصدی لذت‌های فعلی و درصدی برای لذت‌های آتی.

داریم. این ساختار نیاز ساختن رو ازش یاد گرفتیم و ایده تدریس رو ازش گرفتیم.

برای خوشحال بودن و سبک زندگی‌ام هم باز از دو استاد مهمی که داشتم تاثیر گرفتم. سعی کردم مزیت‌های خاص هرکدوم رو با هم ترکیب کنم. این دو نفر عملاً ایده‌های اصلی من رو شکل دادن.

آیا در پایان نصیحتی دارید؟

قبلاً گفتم، همین که در واقع پول واقعی‌شون رو درست خرج کنن. یعنی اینکه پول واقعی، که زمان هست، رو در مقیاس مناسبی خرج کنید که محصولش بشه یه شادی هشتاد ساله. برای هر کسی ممکنه لذت متفاوتی باشه؛ مثلاً یکی لذت می‌بره که سکه‌های پولش رو بشماره، یکی از کشف ناشناخته‌ها لذت می‌بره، یکی تو سفر، یکی تو حل مسائل پیچیده و بغرنج، یکی تو شبکه‌سازی. یه دستور به درد همه نمی‌خوره، ولی فصل مشترکشون اینه: چی خوشحالت می‌کنه؟ اون کار رو بکن. استرس گذشته رو نکش، غم آینده رو هم خیلی نخور، تو لحظه زندگی کن. از همش بهتره.

یه صحنه از فیلم پاندای کونگ فو کار هست، فکر می‌کنم پاندا زیر درخت هلو می‌خورد و استرس داشت. استاد اوگ وی یه جمله خوب گفت: گذشته، گذشته. دیگه نمی‌شه کاریش کرد. آینده رو هم کسی ندیده، پس غمش رو نخور. تو حال زندگی کن. درواقع حال ترجمه Present بود و می‌گفت به همین علت بهش می‌گن چون که هدیه فعلی‌تونه. این جمله جالب بود و واقعاً این استراتژی خیلی درسته. گذشته که گذشته، اگه افسوس بخوری، تغییر

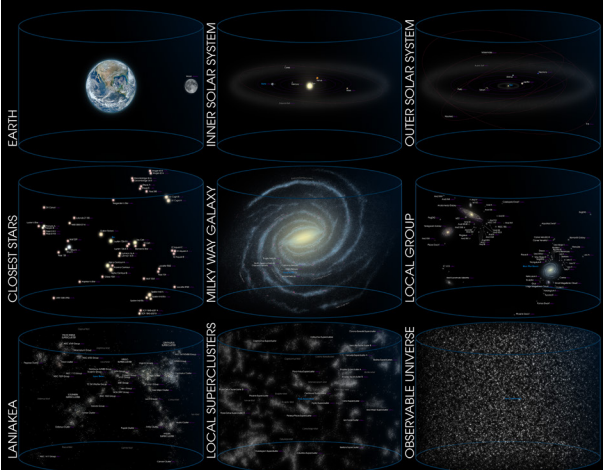
شما و تمام کسانی که دوست‌شان دارید، روزی خواهید مُرد...

تنها بخش کوچکی از چیزهایی که گفته‌اید یا کارهایی که انجام داده‌اید برای تعداد کمی از مردم اهمیت خواهند داشت، آن هم صرفاً برای یک مدت کوتاه. این حقیقت ناخوشایند زندگی‌ست. تمام مسائلی که به آن‌ها فکر می‌کنید یا کارهایی که انجام می‌دهید، تنها گریز استنادهای از این حقیقت‌اند.

ما غبارهای کیهانی بی‌اهمیتی هستیم که در یک نقطه‌ای آبی پر سه می‌زنیم و به هم برخورد می‌کنیم. عظمتی برای خودمان تجسم می‌کنیم و اهدافی برای خودمان می‌سازیم. اما راستش را بخواهید، ما هیچ نیستیم.

پس از قهوه‌ی لعتی‌تان لذت ببرید!

مارک منسن



جهت خواندن نسخه کامل
مصاحبه، اسکن کنید.



شناختی‌تان را شفاف‌تر ببینید. این علم به شما کمک می‌کند تا بفهمید مغز شما چگونه تصمیم می‌گیرد، احساسات شما را کنترل می‌کند و رفتارهای مختلف شما را هدایت می‌کند. مغز انسان از میلیاردها نورون تشکیل شده که از طریق سیناپس‌ها با یکدیگر ارتباط برقرار می‌کنند. با مطالعه این شبکه‌های پیچیده، دانشمندان قادرند رفتارها و واکنش‌های شما را بهتر درک کنند و به این ترتیب به شما کمک می‌کنند تا بفهمید چرا شما به شیوه خاصی فکر می‌کنید یا عمل می‌کنید.

همچنین، پیشرفت‌های این علم در درمان بیماری‌هایی مانند آلزایمر، پارکینسون و افسردگی می‌تواند کیفیت زندگی شما را بهبود بخشد. این روش‌ها می‌توانند به افرادی که با این بیماری‌ها دست و پنجه نرم می‌کنند، کمک کنند تا دوباره به زندگی طبیعی خود بازگردند. علاوه بر این، علم اعصاب در تعامل با رشته‌هایی مثل روان‌شناسی و پزشکی به شما کمک می‌کند تا عملکرد ذهنی و رفتاری خود را بهتر بشناسید و درک عمیق‌تری از رفتارهای انسانی پیدا کنید. در نهایت، این علم می‌تواند در بهبود سلامت روان و جسم شما نقشی مهم ایفا کند.

« هوش مصنوعی: شبیه‌سازی هوش انسانی

هوش مصنوعی (AI) به عنوان شاخه‌ای از علوم کامپیوتر، به ماشین‌ها این توانایی را می‌دهد که فرآیندهای پیچیده انسانی مانند یادگیری، تصمیم‌گیری و حتی درک زبان طبیعی را تقلید کنند. این فناوری در حال گسترش سریع است و

سپیده نوروزی مطلق

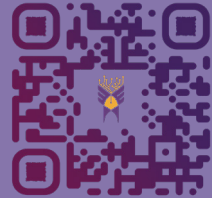
ورودی ۱۴۰۲ مهندسی کامپیوتر



منابع استفاده شده در نوشتار



برای کسب اطلاعات
بیشتر اسکن کنید



« نورالینک

« چگونه هوش مصنوعی و علوم اعصاب زندگی ما را متحول می‌کنند؟

آیا تا به حال به این فکر کرده‌اید که مغز چگونه خاطرات را ذخیره می‌کند یا چه چیزی باعث می‌شود که انسان‌ها بتوانند تصمیمات پیچیده بگیرند؟ یا شاید از خود پرسیده‌اید که چگونه ماشین‌ها می‌توانند به شکلی روزافزون هوشمندتر شوند؟ علوم اعصاب و هوش مصنوعی، هر کدام به شکلی جداگانه، به این سوالات پاسخ می‌دهند. اما وقتی این دو حوزه به هم می‌رسند، تحولی شگرف در انتظار بشر است که نه تنها سلامت و توانایی‌های انسان را تغییر می‌دهد، بلکه دنیای فناوری و ارتباطات را نیز از نو تعریف می‌کند.

« علوم اعصاب: از درک مغز تا درمان‌های نوآورانه

علوم اعصاب، یا نوروساینس، به شما این امکان را می‌دهد که چگونگی عملکرد مغز و سیستم عصبی خود را بهتر درک کنید و ارتباط آن با رفتار، احساسات و توانایی‌های

کاربردهای گسترده‌ای در علوم اعصاب، روان‌شناسی و سلامت روان داشته باشد.

«نورالینک: پلی میان مغز و ماشین»

نورالینک، شرکتی که توسط ایلان ماسک تأسیس شده، نمونه‌ای بارز از ترکیب علوم اعصاب و هوش مصنوعی است. این شرکت با توسعه رابط‌های مغز-کامپیوتر^۱، تلاش می‌کند مغز انسان را مستقیماً به ماشین‌ها متصل کند. هدف این فناوری، نه تنها درمان اختلالات عصبی، بلکه ارتقای توانایی‌های شناختی انسان است.



کاربردهای آن در زمینه‌های مختلف، از جمله بهداشت، خرید آنلاین و رباتیک، به طور چشمگیری در حال پیشرفت است.

یکی از کاربردهای مهم هوش مصنوعی، پردازش داده‌های پزشکی است. سیستم‌های پیشرفته‌ی هوش مصنوعی می‌توانند تصاویر پزشکی را تحلیل کرده و بیماری‌هایی مانند سرطان را در مراحل اولیه شناسایی کنند. این فناوری با افزایش دقت تشخیص، می‌تواند به نجات جان بیماران کمک کند.

سیستم‌های توصیه‌گر نیز یکی دیگر از کاربردهای هوش مصنوعی هستند. از پیشنهاد فیلم در سامانه‌های پخش آنلاین گرفته تا شخصی‌سازی خریدهای روزمره، این سیستم‌ها نقشی اساسی در ساده‌سازی انتخاب‌های ما دارند. این سیستم‌ها با تحلیل داده‌های رفتاری کاربران و تطبیق آن‌ها با ترجیحات مشابه، تجربه کاربری را بهینه می‌کنند. هوش مصنوعی در قلب این فناوری قرار دارد و به‌طور مداوم دقت و اثربخشی آن را افزایش می‌دهد.

در رباتیک پیشرفته، هوش مصنوعی به ربات‌ها این امکان را می‌دهد که به‌صورت مستقل و کارآمد در محیط‌های پیچیده عمل کنند. از جراحی‌های دقیق گرفته تا اکتشافات فضایی، هوش مصنوعی می‌تواند به ربات‌ها کمک کند تا وظایف حساس و دشوار را به دقت و با کمترین خطا انجام دهند.

یکی از حوزه‌های نوین هوش مصنوعی، تحلیل داده‌های مغزی برای درک بهتر عملکردهای شناختی و عاطفی انسان است. این فناوری می‌تواند به بررسی نحوه‌ی پردازش احساسات در مغز کمک کرده و



دستگاه بلایندسایت: بازگشت بینایی به نابینایان

پروژه‌ی بلایندسایت، یکی از نوآوری‌های پیشرفته‌ی نورالینک، با هدف بازگرداندن بینایی به افراد نابینا توسعه یافته است. اما این فناوری چگونه عمل می‌کند؟ ابتدا، یک دوربین هوشمند مانند چشم مصنوعی تصاویر را دریافت و پردازش می‌کند. سپس داده‌های بصری به سیگنال‌های دیجیتال تبدیل شده و توسط الگوریتم‌های پیشرفته‌ی پردازش تصویر تحلیل می‌شوند. در مرحله‌ی نهایی، این داده‌ها به صورت سیگنال‌های الکتریکی به ایمپلنت مغزی ارسال شده و با تحریک نورون‌های قشر بینایی، امکان درک تصاویر برای فرد نابینا فراهم می‌شود. این فناوری می‌تواند برای افرادی که به بیماری‌های شبکه‌ی یا آسیب‌های عصب بینایی مبتلا هستند، یک راه حل موثر باشد. حتی اگر فردی از بدو تولد نابینا باشد، در صورتی که قشر بصری او سالم باشد، می‌تواند از این سیستم بهره‌مند شود و دوباره دنیای اطراف خود را ببیند. برای بهینه‌سازی عملکرد این سیستم، فناوری‌هایی مانند الکتروانسفالوگرافی^۱ و تحریک عمقی مغز^۲ برای رمزگشایی دقیق‌تر فعالیت‌های عصبی و ارتباط بهتر مغز با دستگاه‌های دیجیتال به کار می‌روند. همچنین استفاده از الگوریتم‌های پیشرفته یادگیری ماشین به این معنا است که داده‌های عصبی به طور دقیق‌تر تحلیل می‌شوند تا دقت سیستم‌های نورالینک افزایش یابد.

پیشرفت در توسعه فناوری‌های فناوری مغز و رایانه^۳ چالش‌های زیادی را پیش

روی ما قرار داده است که شاید برای شما هم جالب باشد بدانید. یکی از مهم‌ترین چالش‌ها رمزگذاری سیگنال‌ها است: چگونه می‌توان داده‌های تصویری دوربین‌ها را به سیگنال‌هایی تبدیل کرد که مغز ما آن‌ها را به راحتی درک کند؟ شاید به نظر ساده برسد، اما این فرآیند به دلیل پیچیدگی مغز انسان بسیار دشوار است.

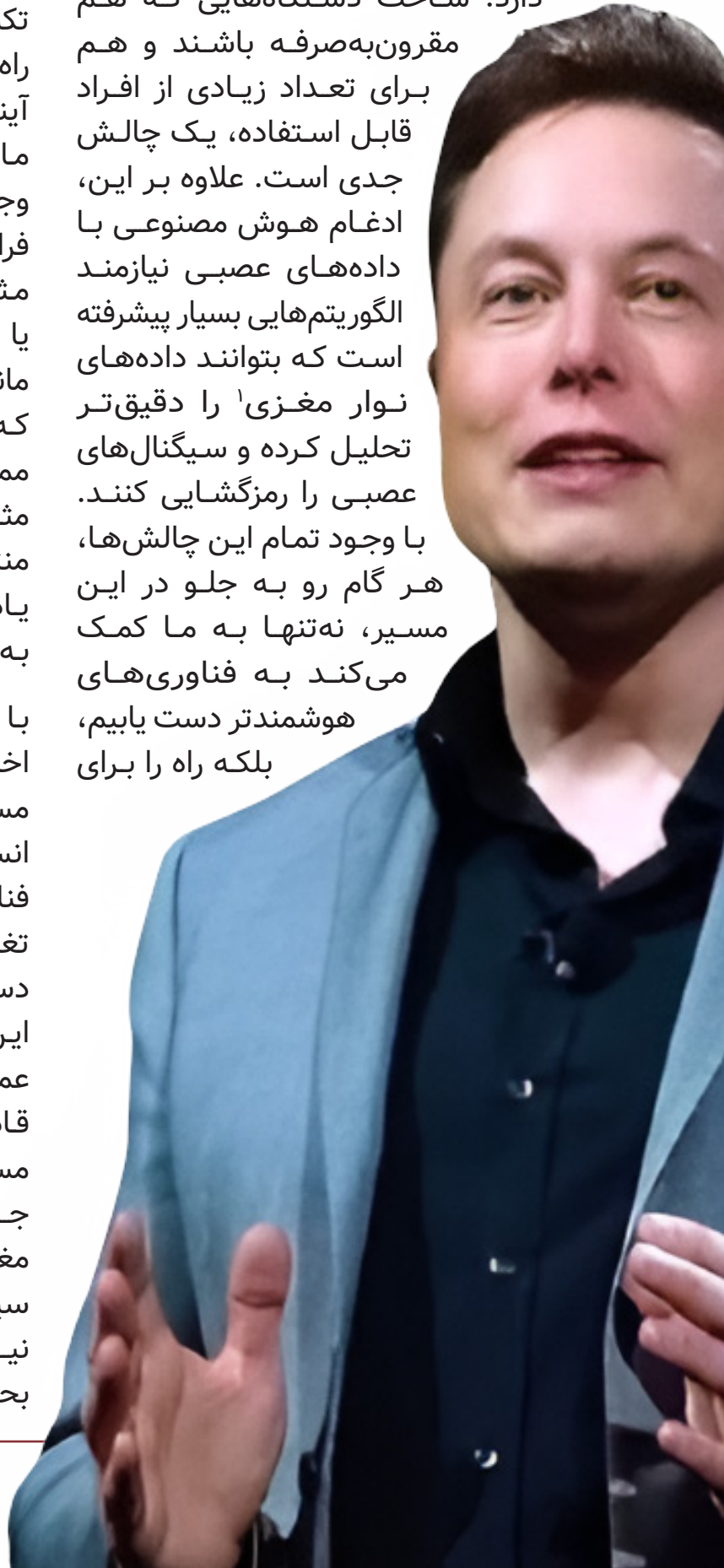
تا به حال فکر کرده‌اید که چنین فناوری‌هایی چگونه می‌توانند بدون تأخیر کار کنند؟ این همان چیزی است که ما آن را حداقل تأخیر می‌نامیم. سیستم باید در زمان واقعی عمل کند تا شما به عنوان کاربر، تجربه‌ای بی‌نقص داشته باشید. اما چالش بزرگ‌تر این است که تحریک مداوم مغز می‌تواند منجر به خستگی عصبی یا حتی آسیب به نورون‌ها شود. به همین دلیل، دانشمندان در تلاش‌اند تا پروتکل‌هایی ایمن و کارآمد طراحی کنند.

دنیا یی کاملاً جدید و شگفت‌انگیز باز می‌کند. شما چطور؟ آیا آماده‌اید که بخشی از این آینده هیجان‌انگیز باشید؟

تکنولوژی‌هایی مانند بلایندسایت تنها آغاز راه هستند و شاید شما هم تصور کنید که در آینده این فناوری‌ها چقدر می‌توانند زندگی ما را تغییر دهند. تصور کنید دستگاه‌هایی وجود داشته باشند که توانایی‌های شما را فراتر از محدودیت‌های طبیعی تقویت کنند، مثلاً دیدن در تاریکی یا بهبود حافظه‌تان. یا شاید درمان‌هایی انقلابی برای مشکلاتی مانند فلج، افسردگی، یا اضطراب ارائه شوند که زندگی بسیاری را متحول کنند. حتی ممکن است روزی بتوانید اطلاعات جدید، مثل یک زبان تازه، را مستقیماً به مغزتان منتقل کنید و تنها در چند ساعت آن را یاد بگیرید. آینده‌ای که شاید خیلی دور به نظر می‌رسد، اکنون در دسترس است.

با وجود تمام مزایا، این فناوری‌ها سوالات اخلاقی مهمی را مطرح می‌کنند. یکی از این مسائل این است که تا چه حد می‌توان انسان را با ماشین ادغام کرد و آیا چنین فناوری‌هایی مرزهای هویت انسانی را تغییر خواهند داد. علاوه بر این، موضوع دسترسی عادلانه نیز اهمیت دارد؛ آیا این فناوری‌ها به‌طور گسترده در دسترس عموم خواهند بود یا تنها افراد ثروتمند قادر به بهره‌برداری از آن‌ها خواهند شد؟ مسئله حریم خصوصی ذهنی نیز چالشی جدی است؛ زیرا حفاظت از داده‌های مغزی کاربران در برابر سوءاستفاده به سیاست‌ها و فناوری‌های پیشرفته‌ای نیاز دارد. بررسی این مسائل نیازمند بحث‌های بین‌رشته‌ای میان متخصصان

شاید از خود بپرسید، این فناوری‌ها چقدر در دسترس عموم خواهند بود؟ پاسخ این سؤال به مسائل مقیاس‌پذیری بستگی دارد. ساخت دستگاه‌هایی که هم مقرون به‌صرفه باشند و هم برای تعداد زیادی از افراد قابل استفاده، یک چالش جدی است. علاوه بر این، ادغام هوش مصنوعی با داده‌های عصبی نیازمند الگوریتم‌هایی بسیار پیشرفته است که بتوانند داده‌های نوار مغزی^۱ را دقیق‌تر تحلیل کرده و سیگنال‌های عصبی را رمزگشایی کنند. با وجود تمام این چالش‌ها، هر گام رو به جلو در این مسیر، نه تنها به ما کمک می‌کند به فناوری‌های هوشمندتر دست یابیم، بلکه راه را برای

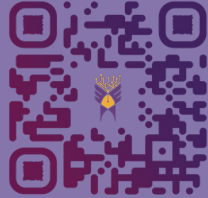


شیدا ستاری

ورودی ۱۳۹۹ مهندسی کامپیوتر



منابع استفاده شده در نوشتار

برای کسب اطلاعات
بیشتر اسکن کنید

« کاهن معبد دلفی »

فنی، فیلسوفان و سیاست‌گذاران است. علوم اعصاب و هوش مصنوعی، هر یک به تنهایی انقلابی در زندگی ما ایجاد کرده‌اند. اما وقتی این دو حوزه به هم می‌رسند، نتایجی فراتر از تصور به دست می‌آید. از بازگرداندن بینایی به نابینایان گرفته تا ارتقای توانایی‌های شناختی، این فناوری‌ها در حال تغییر مسیر آینده بشر هستند. آینده‌ای که دیگر تنها تخیل نیست، بلکه واقعیتی در حال شکل‌گیری است.

Python به چه معناست؟

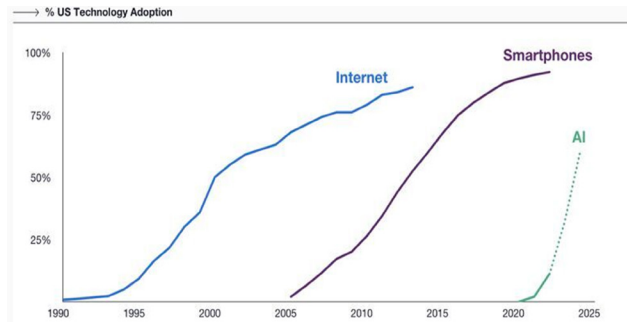
کلمه pythia برگرفته از واژه python، کاهنه معبد دلفی و پیشگوی آپولو در یونان باستان بود که پیام‌های الهی را منتقل می‌کرد.



سال ۱۹۹۷، یادگیری ماشین^۱ به عنوان زیرشاخه‌ای از هوش مصنوعی^۲ معرفی شد که یک توانایی جزیی یادگیری از داده‌ها را به ماشین می‌داد. اما این قدم به تنهایی انسان را به هدف خود نمی‌رساند! پس در سال ۲۰۱۲ ظهور یادگیری عمیق و استفاده از شبکه‌های عصبی^۳ چند لایه انقلابی را رقم زد، انقلابی که تا سال ۲۰۲۱ چندان قابل لمس و حس کردن نبود...

اما بعد از آن، با گسترش هوش مصنوعی مولد^۴، تحولی بزرگ در دنیای فناوری رخ داد. شرکت‌های بزرگ فناوری، با سرمایه‌گذاری‌های گسترده، به سمت این تکنولوژی حرکت کرده و در تلاش‌اند تا آن را در صنایع مختلف به کار گیرند برخلاف انقلاب قبلی که عموم مردم خبری از آن نداشتند، این بار می‌توانستند بدون داشتن آگاهی عمیق از روند پیچیده و تحولات پشت صحنه به آسانی از نوک کوه یخی به نام مدل‌های بزرگ زبانی^۵ که محصولی از هوش مصنوعی مولد بود، استفاده کنند.

با این حال، استفاده از این مدل‌های زبانی فقط یک روی سکه است! روی دیگر آن، مربوط به چالش‌های مرتبط با حریم خصوصی، شفافیت داده‌ها و کنترل اطلاعات است. سیاست‌های محرمانگی و خط‌مشی شرکت‌های بزرگ فناوری باعث می‌شوند که اطلاعات به صورت عمومی منتشر نشود و بسیاری از پژوهشگران به داده‌های اساسی این مدل‌ها دسترسی نداشته باشند. این عدم شفافیت، مانعی بزرگ در برابر محققان است و امکان بررسی و ارزیابی دقیق تاثیر عواملی همچون اندازه پارامترها، نوع داده‌های آموزشی و روش‌های تنظیم مدل،



این نمودار روند پیشرفت و جا افتادن سه تکنولوژی مهمی که در طول تاریخ بشریت اتفاق افتادند را در جامعه آمریکا نشان می‌دهد. اما هدف من از آوردن این نمودار صحبت از مردم آمریکا نیست!! بیا یاد مثل یک ماشین زمان از اول شروع کنیم به مرور کردن، یعنی از وقتی که اینترنت دایل‌آپ با صدای نوستالژی‌طور و سرعت کندش پای خود را به خانه‌ها باز کرد تا اینترنت پرسرعت روی گوشی‌های همراه و حالا هم که زمزمه اینترنت ماهواره‌ای و استارلینک چند وقتی بر سر زبان‌ها افتاده است.

در مورد تلفن‌های همراه هم فکر می‌کنم تقریباً همه از آن ۱۱۰۰ ها (اولین نسل چکش) تا اسمارت‌فون‌هایی که امروزه نه تنها یک وسیله ارتباطی بلکه پیشران خیلی از کسب‌وکارها در حوزه‌های مختلف هستند، دید خوبی داشته باشید.

اما می‌رسیم به نمودار سوم، یعنی هوش مصنوعی!! هوش مصنوعی به عنوان یکی از تاثیرگذارترین و پربحث‌ترین تکنولوژی که دنیای کامپیوتر به خودش دیده، در چندین دهه و با طی کردن مسیری از یک ایده بلندپروازانه در سال ۱۹۵۶ با هدف خلق ماشین‌هایی که بتوانند هوش انسانی را تقلید و یا حتی فراتر از آن عمل کنند، مطرح شد. چندین سال بعد یعنی

۱. Machine learning

۲. Artificial intelligence

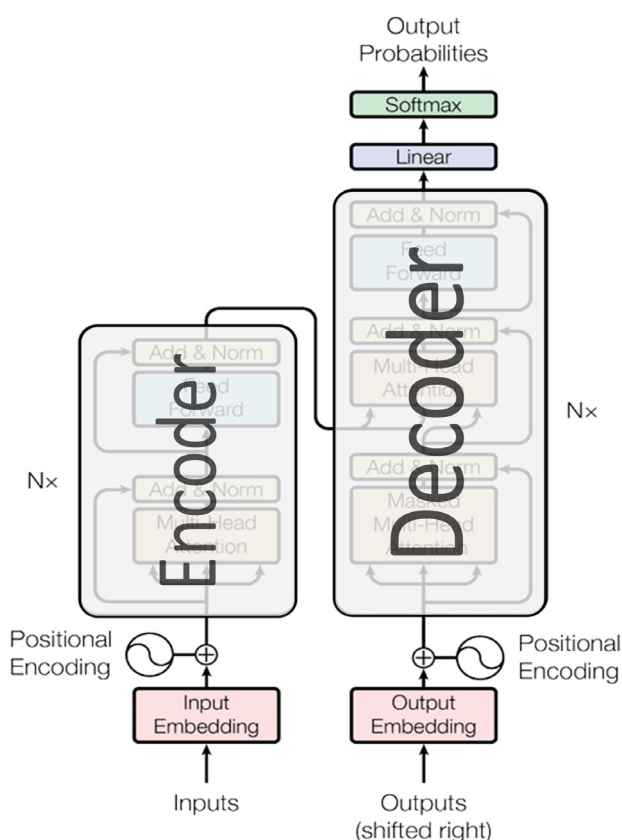
۳. Deep Learning

۴. Generative

۵. Large Language Models



قبلی طراحی شده بود. اما چیزی که امروز می‌بینیم فراتر از کاربرد اولیه آن است. با مقدار زیادی ساده‌سازی و صرف‌نظر از جزئیات، ساختاری که می‌توان برای این معماری متصور شد بدین شکل است:



نکته مهمی که وجود دارد اینست که تمامی مدل‌ها از کل ساختار این معماری استفاده نمی‌کنند بلکه بر حسب و نحوه استفاده از بخش‌های مختلف این معماری، سه دسته مدل را می‌توان ساخت که تجاری‌ترین و پرکاربردترین قسمت ماجرا مربوط به مدل‌هایی است که بر اساس رمزگشا پیاده‌سازی شده‌اند.

❖ مدل‌های رمزگشا^۱

مدل‌های رمزگشا همان‌طور که از نامشان مشخص است، تنها از قسمت “Decoder”

تأثیر مستقیمی بر عملکرد نهایی آن دارند. پروژه Pythia با هدف رفع این محدودیت‌ها و فراهم آوردن بستری باز و شفاف در زمینه تحقیقات بر روی مدل‌های زبانی بزرگ طراحی شده است.

Model Size	Layers	Model Dim	Equivalent Models
70 M	6	512	—
160 M	12	768	GPT-Neo 125M, OPT-125M
410 M	24	1024	OPT-350M
1.0 B	16	2048	—
1.4 B	24	2048	GPT-Neo 1.3B, OPT-1.3B
2.8 B	32	2560	GPT-Neo 2.7B, OPT-2.7B
6.9 B	32	4096	OPT-6.7B
12 B	36	5120	—

گروه EleutherAI مجموعه‌ای از مدل‌های زبانی با پارامترهای متنوع و داده‌های آموزشی گسترده را به صورت عمومی منتشر کرده است. از این منظر، این مقاله می‌تواند یک فرصت مناسب برای تحقیق علاقه‌مندان باشد. علاوه بر این، گروه نیز یک سری مطالعات موردی را بر روی مجموعه زبانی در جهت افزایش میزان بهینه‌سازی مدل‌ها در فاز پیش‌آموزش انجام داده‌اند که در ادامه نتایج به دست آمده را خواهیم دید.

اما پیش از آنکه به سراغ جزئیات بیشتر نتایج و دستاوردهای این گروه برویم، بهتر است ابتدا به ایده و ساختار معماری مدل‌های زبانی بپردازیم.

❖ معماری ترانسفورمر^۱

مدل‌های زبانی بزرگ، پیشرفت ویروس‌گونه خود را وام‌دار مقاله‌ای به نام Attention is All You Need هستند. در این مقاله، دانشمندان گوگل معماری به نام ترانسفورمر^۱ را معرفی کردند. معماری که در ابتدا برای حل مشکلات مدل‌های ترجمه ماشینی

شناسه‌هایی را به توکن‌ها اختصاص داده باشد

"The" → 101, "cat" → 202, "is" → 303, "playing" → 404, "." → 505

آنگاه شناسه عددی که به هر توکن در مثال بالا اختصاص خواهد داد بدین شکل می‌باشد:

[101, 202, 303, 404, 505]

حال هر یک از شناسه‌های عددی مربوط به توکن‌ها به یک بردار در فضای چند بعدی به نام منیفلد^۲ تبدیل می‌شود. منیفلد مانند یک پایگاه داده نیست؛ بلکه نقشه‌ای است از الگوهای که مدل از داده‌های آموزشی یاد گرفته است. در حقیقت، قدرت مدل‌های زبانی بزرگ در فهم الگوها است، نه حفظ داده‌ها! برای مثال، یک مدل آثار شکسپیر را حفظ نمی‌کند. در عوض، آن‌ها را به نوعی سنتز می‌کند یعنی الگوهای از گرامر، سبک و روابط بین کلمات را یاد می‌گیرد تا متنی شبیه به سبک شکسپیر تولید کند. مفهوم تعبیه‌سازی^۳ نیز دقیقاً به چنین چیزی اشاره دارد. در واقع، لایه تعبیه‌سازی^۴ اولین بخشی است که توکن‌ها پس از نگاشت عددی با آن روبرو می‌شوند، در این لایه هر توکن به یک بردار تبدیل می‌شود که اطلاعات بیشتری نسبت به یک شناسه عددی را در خود دارد.

« تک داغ!

اگر ما بخواهیم صرفاً یک‌سری بردارهای رندوم را به اعداد نسبت بدهیم اگرچه ممکن است ساده به نظر بیاید، اما نمی‌تواند اطلاعات خاصی را در مورد معنای کلمات یا حتی روابط بین کلمات

این معماری بهره می‌برند و عمدتاً با هدف تولید متن^۱ و پیش‌بینی کلمه بعدی طراحی شده‌اند. سردمداران این نوع معماری، مدل‌هایی مثل Chat GPT و Gemini و LLaMa^۲ هستند که تقریباً می‌توانم مطمئن باشم حداقل اسم یکی از آن‌ها یا به گوشتان خورده است و یا از آن‌ها استفاده کرده‌اید.

اما سوال اینجاست که چگونه این مدل‌ها بدون وجود Encoder توانایی درک متون انسانی را دارند و متوجه حرف ما می‌شوند؟

در واقع، زمانی که جمله‌ای را به یک مدل زبانی، مثل چت جی‌پی‌تی وارد می‌کنید ابتدا آن را به اجزا کوچکتر کلمه یا زیر کلمه به نام توکن تقسیم می‌کند تا به فرم مناسب و قابل هضم‌تری برای مدل تبدیل شوند.

به طور مثال: در جمله «The cat is playing.» توکن‌ها در این جمله عبارتند از:

[«The», «cat», «is», «playing», «.»]

چالشی که بعد از این مرحله با آن روبرو هستیم، این است که مدل‌ها سر از داده‌های پیچیده‌ای مانند زبان انسانی در نمی‌آورند، درواقع آنها فقط اعداد را متوجه می‌شوند و با این موجودات ریاضی سروکار دارند؛ بنابراین هر توکن تشخیص داده شده به یک شناسه عددی یکتا نگاشت می‌شود. مدل، این شناسه‌های عددی را بر اساس دیکشنری که قبلاً در طی مرحله پیش‌آموزش، جلوتر به آن خواهیم رسید، از داده‌های عظیم استخراج کرده است را به هر توکن اختصاص می‌دهد.

بطور مثال: اگر مدل در دیکشنری خود چنین



سپس از ماژول pandas برای تبدیل داده‌ها به یک ساختار جدولی استفاده می‌شود. دستور زیر این کار را انجام می‌دهد:

`pd.DataFrame(data)`

◀ نمایش داده‌ها

با استفاده از دستور `print()`، محتویات "Data Frame" در قالب یک جدول قابل خواندن به نمایش درمی‌آید.

```
python-IAUbbmag
#one hot encoding using OneHotEncoder of Scikit-Learn
import pandas as pd
from sklearn.preprocessing import OneHotEncoder
```

```
python-IAUbbmag
#Building an employee dataset for example
data = {'Employee id': [10, 20, 15, 25, 30],
        'Gender': ['M', 'F', 'F', 'M', 'F'],
        'Remarks': ['Good', 'Nice', 'Good', 'Great', 'Nice']}
#Converting into a Pandas dataframe
df = pd.DataFrame(data)
#Print the dataframe:
print(f"Employee data : \n{df}")
```

نتیجه این کد عبارتست از:

Employee data:

	Employee id	Gender	Remarks
0	10	M	Good
1	20	F	Nice
2	15	F	Good
3	25	M	Great
4	30	F	Nice

حال که ماشین از درک داده‌های غیر عددی ناتوان است، پس ما هم بوسیله این روش می‌توانیم مقادیر مربوط به ویژگی‌های جنسیت و معیار کارکرد را به برداری از اعداد باینری، مانند $y=[0, \dots, 1, 0, \dots, 0, 0, 0]$ تبدیل کنیم.

را به مدل منتقل کند. این در حالی است که هدف ما از تعبیه‌سازی دقیقاً این است که به مدل پیچیدگی‌های زبان انسانی خودمان را بفهمانیم!

یکی از روش‌های اولیه برای نمایش کلمات به صورت بردارهایی از اعداد، روش One-Hot Encoding است.

در واقع، این روش معمولاً در دیتاست‌های کوچک و با ویژگی‌های گسسته و ساده استفاده می‌شود و مقادیر یا "value" های مربوط به ویژگی‌هایی از دیتاست، موسوم به Categorical Features، را به ویژگی‌های عددی یا "Numerical Features" تبدیل می‌نماید. برای مثال، اگر یک دیتاستی از کارمندان یک شرکت را در نظر بگیریم که شامل ویژگی‌هایی به ترتیب شناسه هر کارمند، جنسیت، و معیار کارکرد آنها باشد، جنسیت و معیار کارکرد جزو Categorical Features و شناسه نوعی Numerical Features به حساب می‌آید.

ما با استفاده از این روش، "value" های مربوط به ویژگی‌های "Categorical" را به "Numerical" تبدیل می‌کنیم تا مدل و الگوریتم‌های یادگیری ماشین نیز درکی از داده‌های غیر عددی داشته باشند.

در مثال زیر، یک دیتاست ساده شامل اطلاعات سه ستون ایجاد شده است:

۱. Employee ID: شناسه عددی برای هر کارمند (مانند ۱۰ یا ۲۰).

۲. Gender: جنسیت با حروف اختصاری «M» (مذکر) و «F» (مونث).

۳. Remarks: توصیفات عملکرد کارکنان مانند «Good» یا «Great».

تا به این مرحله مدل متوجه کلمات ما شده‌است، اما اگر بخواهیم تا همین‌جا کار را متوقف کنیم، مانند اینست که برای یک انسان دنباله ای نامربوط از کلمات را تکرار کرده‌باشیم.

آیا این کافی است؟ قطعاً خیر. همانگونه که انسان‌ها در حین صحبت یا خواندن یک متن متوجه ارتباط، سنخیت و همچنین محتوای کلی کلام یکدیگر می‌شوند، بایستی این موضوع نیز برای مدل مسجل شود. در مرحله‌ی بعد، مدل از مکانیزمی به نام "Attention" استفاده می‌کند تا تشخیص دهد کدام کلمات در متن مهم‌تر هستند و چگونه با یکدیگر ارتباط دارند. طی این مرحله مدام مقادیر بردارهایی که در مرحله قبل محاسبه شده بودند، بروزرسانی می‌شوند.

به عنوان مثال، کلمه‌ی "model" در عبارت‌های "a machine learning model" و "a fashion runway model" دارای بردارهای معنایی متفاوتی است و این درک معنا و تفاوت برداری دقیقاً در همین مرحله رخ می‌دهد.

پس بردارهای به‌روز شده از لایه قبل وارد یک شبکه عصبی ساده به نام "Feed-Forward" می‌شوند. این شبکه با انجام عملیات پیچیده‌تر روی این بردارها، آن‌ها را از سطح جزئیات اولیه فراتر برده و مفاهیم عمیق‌تری از کلمات، مانند نقش دستوری آن‌ها در جمله و معنای دقیق‌ترشان، به بردارها اضافه می‌کند.

قطعه کد زیر در ابتدا شناسایی می‌کند که کدام ستون‌های دیتافریم شامل داده‌های متنی هستند. سپس از روش OneHotEncoder برای تبدیل مقادیر متنی به مقادیر عددی استفاده می‌کند. به این صورت که هر مقدار متنی به چندین ستون جدید تبدیل می‌شود، به طوری که هر ستون نمایان‌گر یکی از دسته‌های ممکن آن مقدار است و با مقدار ۰ یا ۱ نشان داده می‌شود. سپس این ستون‌های عددی جدید به دیتافریم اصلی اضافه می‌شوند و ستون‌های متنی اولیه که دیگر کاربردی ندارند، حذف می‌شوند.

در نهایت، داده‌ها آماده استفاده برای مدل‌های یادگیری ماشین خواهند بود. این روش کمک می‌کند که حتی داده‌های پیچیده و متنی هم به شکلی قابل درک برای ماشین‌ها تبدیل شوند.

```
python -i AUbbmag

categorical_columns = df.select_dtypes(include=
['object']).columns.tolist()
encoder = OneHotEncoder(sparse_output=False)
one_hot_encoded =
encoder.fit_transform(df[categorical_columns])
one_hot_df = pd.DataFrame(one_hot_encoded,
columns=encoder.get_feature_names_out(categorical_colu
mns))
df_encoded = pd.concat([df, one_hot_df], axis=1)
df_encoded = df_encoded.drop(categorical_columns,
axis=1)
print(f"Encoded Employee data : \n{df_encoded}")
```

نتیجه کد چنین چیزی خواهد شد:

Encoded Employee data:

	Employee id	Gender_F	Gender_M	Remarks_Good	Remarks_Great Remarks_Nice
0	10	M	0.0	1.0	0.0 , 0.0
1	20	F	1.0	0.0	0.0 , 1.0
2	15	F	1.0	0.0	0.0 , 0.0
3	25	M	0.0	1.0	1.0 , 0.0
4	30	F	1.0	0.0	0.0 , 1.0

«هنوز تمام نشده...»

از آن به ورودی‌های ما پاسخ می‌دهد.

حالا مدل با این ذهن عددی چگونه قرار است از داده‌های خود استفاده کند و پاسخ سوالات یک انسان را بدهد؟

فرض کنید که جمله زیر به عنوان پرامپت ورودی به یک مدل زبانی داده شده و از آن خواسته می‌شود تا جمله را تکمیل کند: "abc..." (هرکدام از این کاراکترها نماینده یک کلمه و ساختار کلی نماینده دنباله‌ای از کلمات می‌باشد).

فرضا اگر این مدل قبلا، در فرآیند پیش آموزش چنین دنباله‌هایی از کلمات را مشاهده کرده باشد: "abc"، با هزاربار تکرار در مجموعه داده، "abd" با پنج بار و "abe" با یک بار تکرار.

آنگاه در پارامترهای خود احتمال رخداد وقوع کلمه "c" را ۷۵٪، احتمال "d" را ۱۵٪ و احتمال "e" را ۱۰٪ ثبت می‌کند تا در چنین موقعیتی که کاربر از آن خروجی می‌گیرد بتواند منطقی‌ترین انتخاب از نظر یک انسان را تولید کند، هرچند که خود چیزی از کلام متوجه نمی‌شود!

محققان این مقاله توانستند نه تنها روی کاغذ، بلکه در عمل نیز متوجه شوند که تکرار بالای دنباله‌هایی از کلمات می‌تواند موجب بهبود عملکرد مدل در پیش‌بینی کلمات بعدی شود.



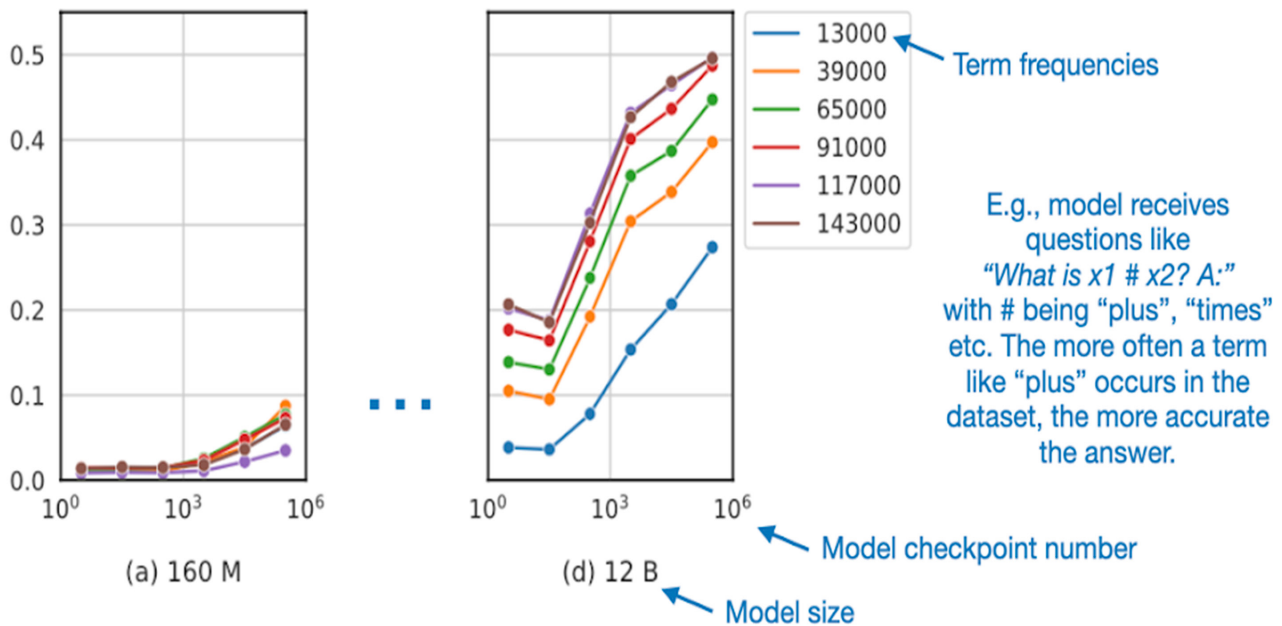
آیا ساختن یک مدل زبانی قوی فقط با معماری Transformer امکان‌پذیر است؟

در نگاه اول ممکن است به نظر برسد که معماری ترنسفورمر به تنهایی می‌تواند یک مدل هوشمند را بسازد تا مثل انسان‌ها متن را درک و به آن پاسخ دهد. اما این تنها شروع کار است! برای اینکه مدلی مانند GPT-۴ بتواند مکالمه‌ای طبیعی با شما داشته و به سؤالات پیچیده پاسخ دهد، نیازمند ۳ فاز Fine-Tuning و Data Augmentation و Pretraining است.

در این شماره به سراغ اولین فاز آموزش یک مدل زبانی، یعنی پیش‌آموزش، می‌رویم و نتایج ارائه‌شده در مقاله Pythia را درباره این فاز بررسی می‌کنیم. همچنین تلاش می‌شود تا در شماره‌های بعدی به سایر مراحل نیز پرداخته شود.

در مرحله‌ی پیش‌آموزش، مدل مبتنی بر معماری ترنسفورمر با استفاده از مجموعه‌ی گسترده‌ای از داده‌های متنی، شامل کتاب‌ها، مقالات و صفحات وب، آموزش داده می‌شود. این مرحله به مدل این امکان را می‌دهد تا داده‌ها و جملات انسانی را ببیند و ساختار جملات را یاد بگیرد.

در حقیقت، مدل با استفاده از این مجموعه داده، درست مانند یک کودک، مرحله به مرحله آموزش داده می‌شود. این داده‌ها به معماری یاد شده در بالا خوراند می‌شوند و نتیجه این آموزش در قالب یک سری پارامتر، به صورت بردار یا ماتریس، ثبت می‌شوند. این اعداد در حقیقت نقش ذهن مدل را بازی می‌کنند؛ ذهنی که با استفاده



به اتمام می‌رسید، checkpoint هایی از وضعیت فعلی مدل تهیه شد تا اگر فرایند آموزش به هر دلیلی متوقف شد، بتوانند از آخرین نقطه ذخیره شده ادامه دهند.

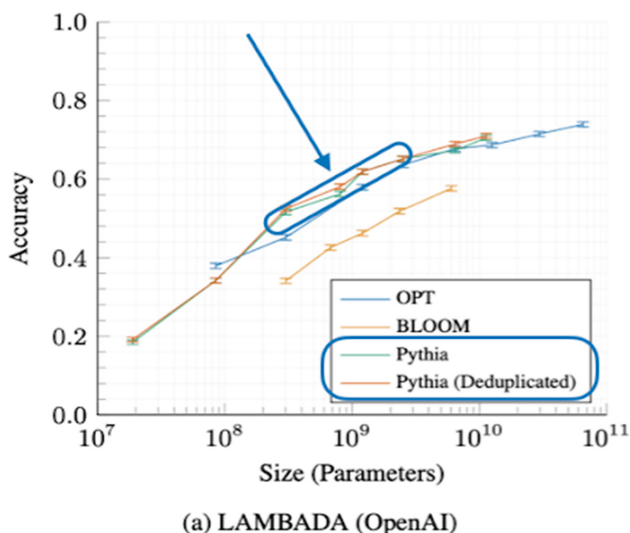
یکی از مطالعات موردی انجام گرفته در مقاله نامبرده، به بررسی برهم کنش و تاثیر افزایش Batch Size - به طور خاص میزان این پارامتر در اینجا ۲M توکن می باشد - بر روی همگرایی و کارایی مدل انجام گرفته است.

افزایش اندازه Batch به طور مستقیم باعث کاهش تعداد گام‌های لازم برای تکمیل یک دوره آموزش یا "Epoch" می‌شود. به طور مشخص، اگر پیش از این برای تکمیل یک دوره، نیاز به طی کردن x گام داشتیم، با افزایش اندازه این پارامتر، می‌توانیم همین دوره را با $x/2$ گام به اتمام برسانیم. این مسئله منجر به کاهش زمان آموزش و بهره‌گیری بهتر از ظرفیت محاسباتی سخت‌افزار (به‌ویژه GPU) می‌شود.

اما از طرفی هم افزایش آن می‌تواند نگرانی‌هایی را در خصوص همگرایی مدل

درواقع، یکی از آزمایش‌های کلیدی مقاله Pythia این بود که داده‌های پیش‌آموزش را با تکرار کنترل‌شده بررسی کردند. به این صورت که دنباله‌هایی از کلمات با تکرارهای متفاوت (برای مثال ۱۰۰ بار، ۱۰ بار، و یک بار) به مدل داده شد و سپس رفتار مدل در پیش‌بینی آن‌ها بررسی گردید. نتیجه نشان داد که دنباله‌های پرتکرار نه تنها با دقت بیشتری پیش‌بینی می‌شوند، بلکه مدل حتی در شرایطی که ترتیب کلمات کمی تغییر کند نیز، ارتباط نزدیک‌تری با این دنباله‌های پرتکرار حفظ می‌کند.

نویسندگان Pythia مجموعه هشت تایی مدل‌ها را بر روی دیتاستی به نام The Pile با حجم ۸۰۰ گیگابایت معادل با ۳۴۰ میلیارد توکن آموزش داده‌اند. از آنجایی که آموزش مدل‌های زبانی بزرگ معمولاً مدت زمان زیادی طول می‌کشد و در این بین ممکن است مشکلاتی پیش بیاید، محققان این حجم از دیتا را نه یکباره بلکه در طی batch هایی-تقسیم‌بندی‌های کوچکتری از دیتا-به مدل خوراندند. در این میان، هرگاه پردازش اصطلاحاً یک iteration مربوط به هر batch



گروه برای بررسی تاثیر این موضوع، مدل را بر روی دو مجموعه داده مختلف آموزش دادند:

۱. مجموعه داده اصلی که شامل تمام داده و بدون حذف تکرارها بود.

۲. مجموعه داده Deduplicated که حجمی معادل تقریباً ۲۰۷ میلیارد توکن دارد.

بر خلاف چیزی که انتظار می‌رفت، اگرچه حذف داده‌های تکراری از نظر تئوری می‌تواند به بهبود کیفیت داده، تاثیر بر روی کارایی یا Performance مدل و کاهش حافظه مورد نیاز کمک کند، اما نتایج این تحقیق همان طور که در شکل مشخص است، نشان داد که آموزش بر روی داده‌های deduplicated تفاوت قابل توجهی در دقت مدل ایجاد نکرده است. منحنی‌های مربوط به مدل آموزش‌دیده بر روی داده‌های اصلی و مدل آموزش‌دیده بر روی داده‌های deduplicated تقریباً همپوشانی داشتند و دقت هر دو مدل تقریباً یکسان بود. این یافته‌ها نشان می‌دهد که حذف داده‌های تکراری لزوماً منجر به بهبود عملکرد و کاهش سوگیری مدل‌های زبانی بزرگ نمی‌شود و ممکن است در بسیاری

به وجود آورد. به طور خاص، این نگرانی وجود دارد که با بزرگ‌تر کردن Batch Size، نوسانات گرادیان‌ها کاهش یابد و مدل به یک کمینه محلی گیر بیفتد، یا به درستی همگرا نشود. اما یافته‌های مطالعه Pythia نشان می‌دهد که با وجود افزایش هزینه سخت‌افزاری، همگرایی مدل آسیب نمی‌بیند، به شرطی که پارامترهای یادگیری به درستی تنظیم شوند. به بیان دیگر در این مطالعه، مشاهده شد که با تنظیم نرخ یادگیری به صورت خطی بر اساس اندازه دسته، می‌توان اثرات منفی احتمالی را جبران کرد. این افزایش به همراه افزایش مناسب نرخ یادگیری، باعث می‌شود تا مدل به همان سرعت و دقت همگرا شود که در اندازه دسته کوچک‌تر انجام می‌شد. به عبارت دیگر، هماهنگی بین Batch Size و Learning Rate کلید موفقیت در حفظ همگرایی و بهبود کارایی مدل است.

«برگردیم به روند پیش‌آموزش...»

دادگان Pile شامل داده‌هایی از منابعی مثل ArXiv, PubMed, Wikipedia و سایر داده‌های متنی است. این داده به دلیل وسعت شامل متونی است که بارها در فرایند Train مدل دیده می‌شود. به همین دلیل از لحاظ تئوری انتظار می‌رود محققان به دنبال اجرا و پیاده‌سازی فنون حذف داده^۱ تکراری در جهت کاهش هزینه سخت‌افزاری و همچنین افزایش دقت و کارایی مدل باشند. اما سوال اصلی اینست که این روش در عمل هم به همان اندازه موثر است؟

در مجموعه داده آموزشی قرار دارند. این موضوع آنقدر مهم است که تیم توسعه‌دهنده GPT-۴، حتی پس از آماده شدن مدل، به مدت ۶ ماه به بررسی و تحلیل دقیق مدل پرداختند تا مطمئن شوند که رفتارهای ناشی از Memorization به نقض حریم خصوصی کاربران منجر نشود. این مسئله نشان‌دهنده حساسیت بالای موضوع است، چرا که فاش شدن اطلاعات شخصی در پاسخ‌های مدل می‌تواند منجر به پیامدهای حقوقی و شکایت کاربران شود.

یکی از سوالات مهم این است که آیا ترتیب دیده شدن داده‌ها در طول پیش‌آموزش می‌تواند بر میزان Memorization تأثیر بگذارد؟ به عبارت دیگر، آیا مدل به دلیل مشاهده زود هنگام‌تر یک داده خاص، آن را بهتر یا بیشتر حفظ می‌کند و اطلاعات جدیدتر را فراموش می‌کند؟

در مقاله، محققان این موضوع را به طور سیستماتیک بررسی کردند و به نتایج جالبی دست یافتند. آن‌ها نشان دادند که ترتیب ارائه داده‌ها در طول فرایند آموزش، تأثیر قابل توجهی بر میزان Memorization مدل ندارد. به عبارت دیگر، حتی اگر داده‌ها به صورت تصادفی شافل شوند یا به ترتیب خاصی مرتب شوند، مدل همچنان تمایل به حفظ محتوای خاصی از داده‌ها دارد. این یافته‌ها حاکی از آن است که Memorization بیشتر به کیفیت و ماهیت داده‌ها بستگی دارد تا ترتیب مشاهده آن‌ها. به ویژه، داده‌هایی که تکرار بالایی دارند یا محتوای خاص و منحصر به فردی دارند، بیشتر در معرض حفظ شدن قرار می‌گیرند. این نتایج نشان می‌دهد که روش شافل

از موارد، تأثیر محسوسی نداشته باشد.

در حالی که محققان به دنبال بهینه‌سازی فرایند آموزش با تکنیک‌هایی همچون افزایش Batch Size و حذف داده‌های تکراری هستند تا زمان و هزینه محاسباتی را کاهش دهند، چالش دیگری به نام "Memorization" مطرح می‌شود. Memorization به معنای حفظ محتوای داده‌ها توسط مدل و تکرار آن در پاسخ‌ها است که می‌تواند به نقض حریم خصوصی کاربران منجر شود. اگرچه حذف داده‌های تکراری می‌تواند به بهبود کیفیت داده‌ها کمک کند، اما یافته‌های مقاله Pythia نشان می‌دهد که حتی با استفاده از تکنیک‌های مختلف برای بهینه‌سازی ترتیب داده‌ها، مشکل Memorization همچنان وجود دارد.

بررسی تأثیر ترتیب داده‌ها در مدل‌های Memorization زبانی بزرگ

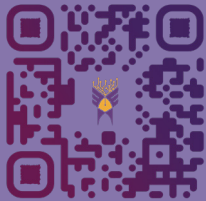
Memorization در مدل‌های زبانی بزرگ به حالتی گفته می‌شود که مدل محتوای دیده‌شده در حین آموزش را به طور دقیق و بدون تغییر، در پاسخ به درخواست‌های کاربران تکرار می‌کند. به عنوان مثال، اگر یک مدل بخشی از یک متن ۶۴ توکنی را در فرایند پیش‌آموزش خود دیده باشد، ممکن است در مواجهه با یک درخواست که شامل ۳۲ توکن اول آن متن است، دقیقاً ۳۲ توکن بعدی را بدون تغییر تولید کند. اگرچه این رفتار از دیدگاه پیش‌بینی کلمات بعدی قابل قبول است، اما می‌تواند مشکلات حریم خصوصی جدی ایجاد کند، به ویژه زمانی که داده‌های حساس یا خصوصی مانند شماره تلفن‌ها یا آدرس‌ها

گودرز جعفری

توسعه دهنده



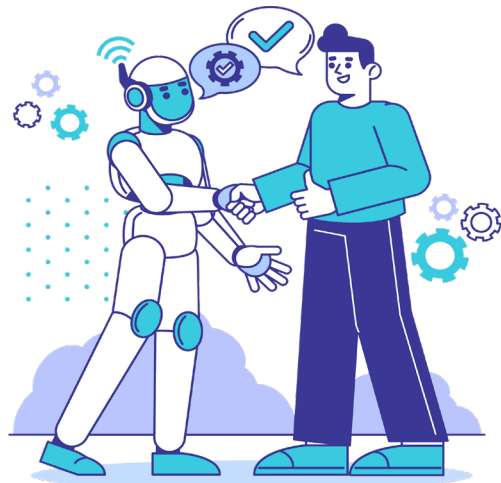
منابع استفاده شده در نوشتار

برای کسب اطلاعات
بیشتر اسکن کنید

« پرورش طوطی دیجیتالی »

آیا تا به حال پیش آمده است که برخی وظایف تکراری در روند کاری شما آنقدر زیاد و خسته‌کننده شده و زمان زیادی را از شما بگیرند؟ برای مثال، اگر صاحب یک فروشگاه اینترنتی باشید، احتمالاً با سوالات پرتکراری مانند نحوه پرداخت، راهنمای استفاده از سایت و موارد مشابه مواجه شده‌اید. با اینکه پاسخ این سوالات را از قبل تهیه کرده و در دسترس قرار داده‌اید، ممکن است کاربران کمتر به آن‌ها مراجعه کنند و بخش پشتیبانی آنلاین بیشتر وقت خود را صرف پاسخگویی به همین سوالات کند. یا فرض کنید چندین کتاب دارید، اما زمان کافی برای مطالعه ندارید و می‌خواهید در کوتاه‌ترین زمان ممکن به نکات مدنظر خود دست پیدا کنید. در چنین شرایطی، چت‌بات‌ها می‌توانند به‌عنوان راه حلی برای کاهش بار کاری و صرفه‌جویی در زمان به کمک ما بیایند. چت‌بات‌ها با استفاده از مدل‌های هوش مصنوعی می‌توانند به‌صورت خودکار به سوالات پاسخ دهند و روندهای تکراری را بدون نیاز به مداخله انسانی انجام دهند. این چت‌بات‌ها می‌توانند

کردن داده‌ها یا تغییر ترتیب آن‌ها نمی‌تواند به تنهایی مشکل Memorization را حل کند. بنابراین، محققان باید به جای تمرکز بر ترتیب داده‌ها، به بررسی کیفیت داده‌های آموزشی و استفاده از تکنیک‌های پیشرفته‌تر مانند فیلترینگ داده‌های حساس توجه بیشتری داشته باشند. در نهایت، برای کاهش ریسک فاش شدن اطلاعات حساس، تمرکز بر ارزیابی دقیق مدل‌ها و بهبود روش‌های آموزش ضروری به نظر می‌رسد. ترنسفورمرها توانسته‌اند درک ما از زبان و هوش مصنوعی را دگرگون کنند، و مدل‌هایی مانند Pythia گواهی بر این پیشرفت شگفت‌انگیز هستند. فاز پیش‌آموزش این مدل‌ها مثل سفری به اعماق زبان است، جایی که پیچیدگی‌ها و ظرافت‌های کلمات با دقت و زیبایی کشف می‌شوند. نتایج حاصل از Pythia تأیید می‌کند که با به کارگیری معماری‌های پیشرفته ترنسفورمرها و داده‌های فراگیر، می‌توان توانایی‌های هوش مصنوعی را به سطوح جدیدی از دقت و کارایی ارتقا داد. این پیشرفت‌ها نه تنها به گسترش مرزهای علمی کمک می‌کنند، بلکه چشم‌اندازهای جدیدی برای حل مسائل پیچیده زبانی و ایجاد سیستم‌های هوشمندتر و اثربخش‌تر را به نمایش می‌گذارند.



۱. شناسایی نیاز: چرا قصد دارید چت بات هوش مصنوعی بسازید؟ به طور مثال هدف من از طراحی چت بات در اینجا، آسان تر کردن دسترسی به اطلاعات آموزشی بدون نیاز به مراجعه حضوری به بخش آموزش و صرفه جویی در زمان است.

۲. تجربه کاربری: درک نیازها و اولویت های کاربران هدف، مسیر طراحی و عملکرد چت بات را مشخص می کند. این کار به شما کمک می کند تا انتظارات آن ها را برآورده کرده و تجربه بهتری برایشان فراهم کنید.

۳. محدوده عملکرد: مشخص کنید چت بات شما تا چه حد و در چه موضوعاتی می تواند پاسخ گو باشد. محدود کردن حوزه تخصص چت بات، دقت و کیفیت پاسخ های آن را به شکل قابل توجهی افزایش می دهد.

◀ تنظیم پاسخ های اولیه چت بات

برای شروع ساخت یک چت بات هوشمند، به جای اینکه از صفر شروع کنیم و چرخ را دوباره اختراع کنیم، می توانیم از مدل های زبانی پیشرفته ای مثل ChatGPT و یا مدل های موجود در HuggingFace استفاده کنیم. این سرویس ها با ارائه API، به ما این امکان را می دهند که چت بات را به یکی از این مدل های زبانی متصل کنیم تا پاسخ هایی طبیعی و مشابه به مکالمات انسانی تولید کند. علاوه بر این، استفاده از مدل های زبانی به چت بات ما این امکان را می دهد که نه تنها سوالات را به صورت انسانی زده شده ای پاسخ دهد بلکه در طی زمان یاد بگیرد و خود را بهبود دهد. یکی از این مدل ها که قابلیت های گسترده ای دارد، مدل

به طور خاص بر اساس نیازهای شما سفارشی سازی شوند تا پاسخ هایی متناسب با درخواست های خاص کاربران ارائه دهند. انواع مختلفی از چت بات های هوش مصنوعی وجود دارند که هر کدام برای کاربردهای خاصی طراحی شده اند و می توانیم متناسب با نیاز خود آن ها را پیاده سازی کنیم:

- قانون محور: مبتنی بر دستورالعمل های از پیش تعیین شده هستند.
- یادگیری ماشینی: یاد می گیرند و به مرور زبان بهبود می یابند.
- پردازش زبان طبیعی: قادر به درک و پردازش زبان انسانی هستند.
- هیبریدی: ترکیبی از روش های مختلف برای بهبود عملکرد هستند.
- مکالمه ای: بر تعامل طبیعی و مشابه انسان تمرکز دارند.

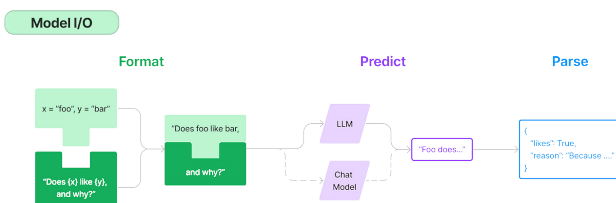
◀ ساخت چت بات هوش مصنوعی شخصی

ساخت یک چت بات هوش مصنوعی شخصی شاید در ابتدا کمی پیچیده به نظر برسد، اما اگر آن را به مراحل کوچک تر تقسیم کنید، روند کار به آسانی انجام می شود و همچنین نتیجه مطلوب تری خواهید گرفت. من در این متن، این فرآیند را با پیاده سازی یک چت بات مخصوص دانشگاه برای پاسخگویی به سوالات مربوط به آیین نامه ها طراحی کردم.

◀ شروع با یک هدف مشخص

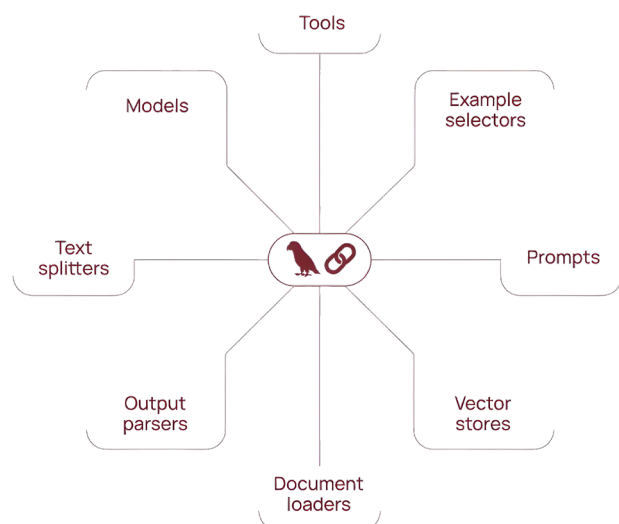
پیش از ورود به جزئیات فنی، لازم است دیدگاه روشنی نسبت به هدف خود داشته باشیم:

طراحی‌شده‌ای هستند که به مدل کمک می‌کنند تا به‌طور دقیق‌تر و مرتبط‌تر پاسخ دهد. به‌عنوان مثال، می‌توانیم یک الگوی دستور ایجاد کنیم که متغیرهایی از کاربر دریافت کند (مثل تاریخ، موضوع یا نوع سؤال) و سپس این اطلاعات را به مدل بدهد تا بهترین و دقیق‌ترین پاسخ را ارائه دهد. به همین ترتیب، این الگوها به مدل کمک می‌کنند تا نتایج بهتری در پاسخ به درخواست‌ها تولید کند. در کنار الگوهای دستوری، LangChain امکان استفاده از تجزیه‌کننده‌ها^۲ را هم فراهم می‌کند. این ابزار به ما کمک می‌کند تا خروجی‌های مدل را تجزیه و تحلیل کرده و اطلاعات مهم و کاربردی را از آن‌ها استخراج کنیم. به عبارت دیگر، مدل‌های زبانی اغلب پاسخ‌هایی می‌دهند که از نظر ساختاری کامل نیستند یا نیاز به پردازش بیشتر دارند. با استفاده از تجزیه‌کننده‌ها، می‌توانیم این خروجی‌ها را به‌صورت منظم و دقیق به کاربر نمایش دهیم.



همان‌طور که در ابتدا گفتم، یکی از چالش‌های کلیدی در دوران دانشجویی، آشنایی با آئین‌نامه‌های آموزشی پر از قوانین و تبصره‌های پیچیده است. هر بار که به این اطلاعات نیاز داریم، مجبور می‌شویم آن بخش خاص را به‌طور کامل بخوانیم و یا حضوری به بخش آموزش مراجعه کنیم. اما چون بیشتر اوقات زمان کافی نداریم، اغلب از جزئیات غافل می‌مانیم.

MinerU است که امکاناتی همچون استخراج دقیق اطلاعات از اسناد، شناسایی و پردازش PDFهای اسکن‌شده و تبدیل فرمول‌ها به فرمت LaTeX را ارائه می‌دهد. حالا که بدون نیاز به سخت‌افزار خاصی و فقط با استفاده از یک API توانستیم چت‌بات را به قابلیت‌های بیشتری مجهز کنیم، برای اینکه پاسخ‌های کاربردی و مؤثرتری به سوالاتمان دهد باید به این فکر کنیم که چگونه ورودی‌ها و خروجی‌ها را مدیریت کنیم. چرا که اگر این ورودی‌ها به‌طور واضح و دقیق طراحی نشوند، چت‌بات ممکن است نتواند پاسخ‌های درست و مورد نیاز ما را ارائه دهد. همچنین، خروجی‌هایی که از آن می‌گیریم بایستی به گونه‌ای ساختاریافته و مفهومی تنظیم شوند تا اطلاعات را به‌درستی منتقل کنند. برای انجام این کار، ابزاری به نام LangChain به دادم‌مان می‌رسد!



در واقع، ما با کمک آن می‌توانیم ورودی‌های چت‌بات را با استفاده از الگوهای دستوری^۱ سفارشی‌تر کنیم. این الگوها، دستورالعمل‌های از پیش

۱. Prompt Templates

۲. Parsers


```
pseudocode-IAUbbmag

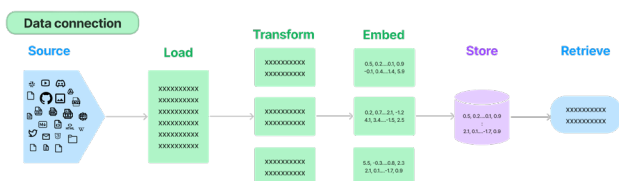
# ایجاد قالب دستور
prompt_template = PromptTemplate.from_template("در مورد {adjective} درس های (content) بهم بگو")

# تنظیم متغیرها
variables = {
    "adjective": "قوانین آموزشی",
    "content": "پیش نیاز"
}

# اعمال قالب بر اساس متغیرها
prompt = prompt_template.format(**variables)
print(prompt)
```

بازیابی

مدل‌های زبانی بزرگ، معمولاً بر اساس حجم عظیمی از داده‌های عمومی آموزش دیده‌اند و این بدان معنی است که در وظایف عمومی عملکرد خوبی دارند، اما وقتی پای مسائل خاص و تخصصی مانند بخش‌های مختلف آیین‌نامه دانشگاه آزاد به میان می‌آید، چندان کارآمد نیستند و ممکن است نتوانند به درستی به سوالات پاسخ دهند. برای حل این مشکل، ما نیاز داریم که از اطلاعات دقیق و خاص استفاده کنیم و این داده‌ها را به مدل زبانی بدهیم تا بتوانیم پاسخ‌های دقیقی دریافت کنیم. اینجا است که فرآیند تولید مبتنی بر بازیابی یا RAG^۳ وارد می‌شود. این فرایند به ما این امکان را می‌دهد که به جای اینکه فقط به دانش عمومی مدل‌های زبانی اتکا کنیم، اطلاعات خاص و مرتبط را از منابع خود (مثل داده‌های دانشگاه یا هر منبع دیگر) بازیابی کنیم و سپس آن اطلاعات را به مدل بدهیم تا پاسخ دقیق‌تری تولید کند. ابزار LangChain به ما این اجازه را می‌دهد که مدل‌های مبتنی بر RAG را به راحتی بسازیم.



برای حل این مشکل، ابتدا با استفاده از یک خزنده^۱ تمامی قوانین آموزشی دانشگاه آزاد را از سایت جمع‌آوری کردم و به صورت فایل‌های PDF ذخیره کردم. سپس با کمک ابزاری مانند LangChain، که یک سیستم Orchestration است، این داده‌ها را به مدل ۳.۵ gpt متصل کردم تا در کمترین زمان ممکن مطالب مورد نظر را پردازش کرده و اطلاعات دقیق‌تری دریافت کنم.

```
pseudocode-IAUbbmag

# ایجاد و تنظیم مدل چت
model = ChatOpenAI(
    openai_api_key="<OPENAI_API_KEY>",
    model_name="gpt-3.5-turbo",
    temperature=0.3
)

# تولید پاسخ با استفاده از مدل و دستور
response = model.generate(prompt.text)
print(response)
```

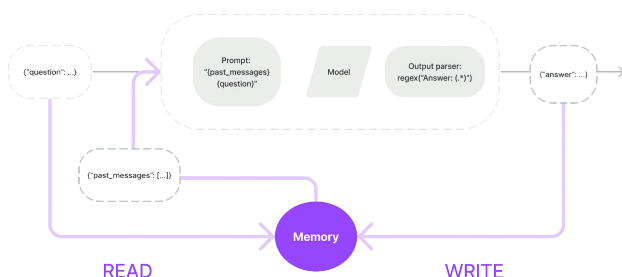
در این تصویر، یک مدل چت با استفاده از یکی از مدل‌های OpenAI و کلید API شخصی ایجاد شده است. برای دریافت این کلید API کافی است در وبسایت OpenAI ثبت‌نام کنید. پارامتر temperature نیز برای کنترل میزان تصادفی بودن پاسخ‌های مدل به کار می‌رود. حال برای شخصی سازی و سفارشی تر کردن ربات با استفاده از کد زیر، یک الگوی دستور^۲ ایجاد می‌کنیم که قادر است چندین متغیر را بپذیرد. این متغیرها از کاربر دریافت شده و به الگو دستور داده می‌شوند.

۱. Crawler

۳. Retrieval-Augmented Generation

۲. Prompt

حافظه: نرم افزارهایی که از مدل‌های زبانی بزرگ^۲ استفاده می‌کنند، معمولاً دارای یک رابط کاربری مکالمه‌ای هستند. یکی از جنبه‌های مهم در هر مکالمه، امکان اشاره به اطلاعاتی است که قبلاً در گفت‌وگو مطرح شده‌اند. این قابلیت ذخیره‌سازی و استفاده از اطلاعات مربوط به تعاملات گذشته را در اینجا حافظه می‌نامیم.



LangChain ابزارهای کلیدی را برای افزودن حافظه به نرم‌افزارمان ارائه می‌دهد. به عنوان مثال، ما بایستی قابلیت خواندن از حافظه را داشته باشیم تا بتوانیم ورودی‌های کاربران را بهینه کنیم. همچنین، لازم است امکان ذخیره‌سازی ورودی‌ها و خروجی‌های اجرای فعلی در حافظه فراهم شود.

```
pseudocode-IAUbbmag

# تنظیم حافظه چت با حداکثر تعداد توکن‌ها
chat_memory = TokenWindowMemory(max_tokens=300, tokenizer=OpenAITokenizer("gpt-3.5-turbo"))

# اضافه کردن پیام کاربر به حافظه چت
chat_memory.add_user_message("برای تغییر رشته حداقل چند واحد در کارشناسی باید قبول شده باشیم؟")

# ایجاد مدل چت
model = ChatOpenAI(openai_api_key="OPENAI_API_KEY", model_name="gpt-3.5-turbo")

# تولید پاسخ با استفاده از مدل و حافظه چت
answer = model.generate(chat_memory.messages()).content()
print(answer.text) # واحد گذرانده باشد # ۲۴ واحد کارشناسی باید در دوره کارشناسی

# اضافه کردن پاسخ به حافظه چت
chat_memory.add(answer)

# اضافه کردن پیام دیگر کاربر به حافظه چت
chat_memory.add(user_message("توابع چند بار می‌توانیم تغییر رشته کنیم؟"))

# تولید پاسخ دیگر با استفاده از مدل و حافظه چت
answer_with_name = model.generate(chat_memory.messages()).content()
print(answer_with_name.text) # ۵۹: دانشجوی در هر یک از مقاطع تحصیلی تنها یکبار می‌تواند تغییر رشته دهد

# اضافه کردن پاسخ به حافظه چت
chat_memory.add(answer_with_name)
```

به این منظور، ابتدا فایل‌های جمع‌آوری شده از وبسایت دانشگاه آزاد را با استفاده از ابزار بارگذاری اسناد^۱ موجود بارگذاری کرده و بعد از آن، با استفاده از ترنسفورمرها این فایل‌ها را برای پردازش آماده می‌کنیم. ترنسفورمرها می‌توانند اسناد بزرگ را به بخش‌های کوچکتر تقسیم کنند تا مدل راحت‌تر بتواند آن‌ها را پردازش کند. این بخش‌های کوچک می‌توانند شامل جملات یا پاراگراف‌های مجزا باشند که به مدل اجازه می‌دهند تا به‌طور دقیق‌تری اطلاعات مورد نظر را استخراج کند.

```
pseudocode-IAUbbmag

# بارگذاری سند از فایل سیستم
document = FileSystemDocumentLoader.load_document("BitByte.txt")

# ایجاد تقسیم‌کننده سند با پارامترهای مشخص
splitter = RecursiveCharacterTextSplitter(
    chunk_size=100,
    chunk_overlap=0,
    tokenizer=OpenAITokenizer("gpt-3.5-turbo")
)

# تقسیم سند به بخش‌های کوچکتر
segments = splitter.split(document)
```

ما از `FileSystemDocumentLoader` برای بارگذاری یک سند از فایل سیستم استفاده می‌کنیم. سپس با استفاده از `OpenAITokenizer`، آن سند را به بخش‌های کوچکتر تقسیم می‌کنیم. برای بهبود کارایی فرآیند بازیابی، اسناد معمولاً به `embedding` تبدیل شده و در `store`های وکتوری ذخیره می‌شوند. `LangChain` چندین ارائه‌دهنده `embedding` و روش‌های مختلف پشتیبانی می‌کند و تقریباً با تمامی `store`های وکتوری محبوب سازگار است.

قابلیت به شما امکان می‌دهد برنامه‌های پیچیده‌تری بسازید که بتوانند وظایف چندگانه را به‌طور هماهنگ انجام دهند. LangChain ابزارهای مناسبی برای ساخت چنین زنجیره‌هایی ارائه می‌دهد و همچنین مجموعه‌ای از زنجیره‌های از پیش آماده را در اختیار شما قرار می‌دهد تا بتوانید سریع‌تر و راحت‌تر کار خود را آغاز کنید.

```
pseudocode-IAUbbmag

# ایجاد مدل چت
chat_model = ChatOpenAI(openai_api_key="<OPENAI_API_KEY>", model_name="gpt-3.5-turbo")

# تنظیم بازیابی با استفاده از ذخیره ساز جاسازی و مدل جاسازی
retriever = EmbeddingStoreRetriever.from_store(embedding_store, embedding_model)

# تنظیم حافظه چت با حداکثر ۱۰ پیام
chat_memory = MessageWindowMemory(max_messages=10)

# ایجاد قالب دستور
prompt_template = PromptTemplate.from_template(
    پاسخ خود را براساس اطلاعات زیر قرار دهید:\n\n{question}\n\nبدهید:\n\n{information}"
)

# ایجاد زنجیره بازیابی مکالمه‌ای
chain = ConversationalRetrievalChain.builder(
    chat_model=chat_model,
    retriever=retriever,
    chat_memory=chat_memory,
    prompt_template=prompt_template
).build()
```

ما از یک زنجیره از پیش ساخته‌شده به نام “Conversational Retrieval Chain” استفاده می‌کنیم که به ما این امکان را می‌دهد یک مدل چت همراه با یک بازیاب^۳، حافظه و الگوی دستور^۴ به کار بگیریم. حالا می‌توانیم به سادگی از این زنجیره برای اجرای سوالات کاربر استفاده کنیم.

```
pseudocode-IAUbbmag

answer = chain.execute("مدرک دانشجویی انتقالی توسط چه واحد دانشگاهی صادر می‌شود؟")
print(answer)
```

Chain‌ها با حافظه و الگوی دستور پیش‌فرض ارائه می‌شوند که می‌توانیم آن‌ها را تغییر دهیم. ایجاد Chain‌های سفارشی بسیار راحت است. توانایی ایجاد زنجیره‌ها، پیاده‌سازی ماژولار^۵ برنامه‌های پیچیده را ساده‌تر می‌کند.

ما یک حافظه چت با پنجره ثابت با استفاده از TokenWindowChatMemory پیاده‌سازی می‌کنیم که به ما امکان می‌دهد پیام‌های چت را که با مدل زبانی تبادل می‌کنیم، بخوانیم و بنویسیم. LangChain همچنین ساختارهای داده‌ای پیچیده‌تر و الگوریتم‌هایی برای بازگرداندن پیام‌های انتخاب‌شده از حافظه به‌جای بازگرداندن همه چیز ارائه می‌دهد. برای مثال، قابلیت بازگرداندن خلاصه‌ای از چند پیام گذشته یا بازگرداندن پیام‌های مرتبط با اجرای کنونی را دارد.

برنامه‌های پیچیده LangChain

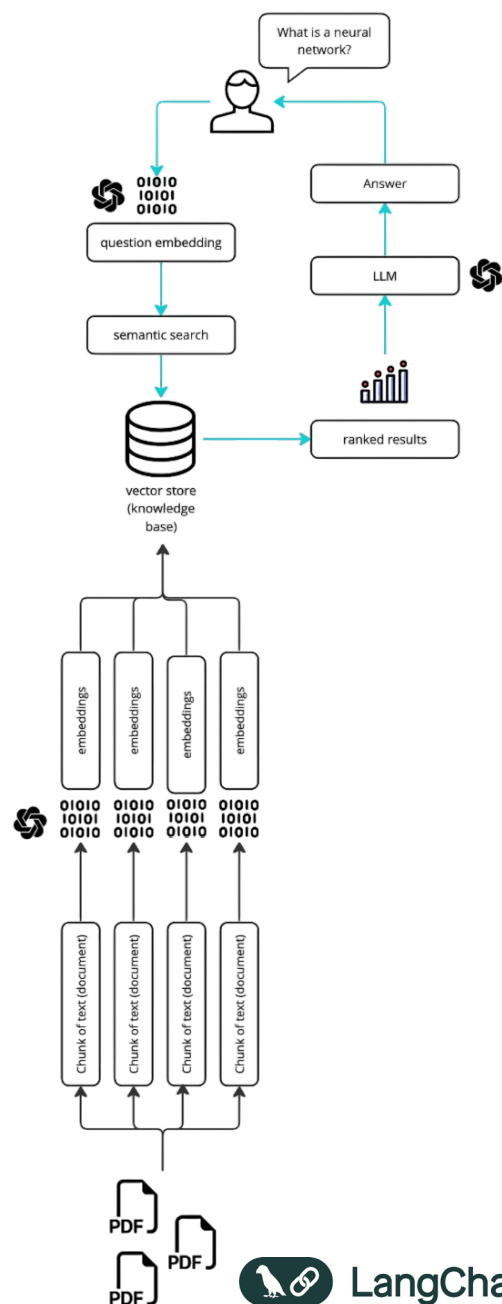
تا اینجا دیدیم که چگونه از اجزای مختلف و جداگانه برای ایجاد نرم‌افزاری مبتنی بر مدل‌های زبانی استفاده می‌کنیم. LangChain علاوه بر این، ابزارهایی برای ساخت برنامه‌های پیچیده‌تر نیز ارائه می‌دهد. به‌عنوان مثال، می‌توان از زنجیره‌ها^۱ و عامل‌ها^۲ استفاده کرد تا برنامه‌هایی پیشرفته و منطبق با نیازهای خاص خود طراحی کنیم.

زنجیره‌ها^۱

به طور کلی، هر نرم‌افزار نیاز دارد که چندین بخش را به ترتیب خاصی اجرا کند. در LangChain، به این فرآیند زنجیره گفته می‌شود. زنجیره‌ها به شما کمک می‌کنند تا توسعه برنامه‌های پیچیده‌تر را ساده‌تر کرده و مدیریت، نگهداری و بهبود آن‌ها را راحت‌تر کنید. زنجیره‌ها همچنین برای ترکیب چندین بخش یا فرآیند در یک برنامه مفید هستند، به‌ویژه زمانی که برنامه نیاز دارد با بیش از یک مدل زبانی کار کند. این

عواملها

LangChain همچنین construct های قدرتمندتری مانند عاملها ارائه می‌دهد. برخلاف زنجیره‌ها، عاملها از یک مدل زبانی به عنوان موتور reasoning برای تعیین اقداماتی که باید انجام شوند و ترتیب آنها استفاده می‌کنند. ما همچنین می‌توانیم به عاملها دسترسی به ابزارهای مناسب برای انجام action های لازم را ارائه دهیم.



آنی‌تا سبحانیان

ورودی ۱۴۰۲ مهندسی کامپیوتر



منابع استفاده شده در نوشتار

برای کسب اطلاعات
بیشتر اسکن کنید



« سنگ بنای نامرئی هوش مصنوعی »

نبرد پردازنده‌ها

در دنیای امروز، رایانه‌ها نه تنها ابزارهایی برای انجام کارها هستند، بلکه قلب تپنده عصر دیجیتال به شمار می‌آیند. این دستگاه‌ها به طور خاص برای پردازش داده‌ها طراحی شده‌اند و به ما این امکان را می‌دهند که از نرم‌افزارهای مختلف استفاده کنیم، بازی‌های ویدیویی جذاب را تجربه کنیم و شبیه‌سازی‌های پیچیده را انجام دهیم. اما چه چیزی باعث می‌شود که این دستگاه‌ها این قدر قدرتمند و کارآمد باشند؟

در پس هر محاسبه و هر تصویر زیبا، دو قهرمان پنهان قرار دارند: CPU و GPU. این دو واحد پردازش، مغز متفکر رایانه را تشکیل می‌دهند. CPU، همانند یک مدیر با تجربه، وظایف مختلف را برنامه‌ریزی و اجرا می‌کند و توانایی انجام محاسبات پیچیده را دارد. در عین حال، GPU، مانند

همزمان با پردازنده مرکزی کار می‌کند تا حجم داده‌ها را به‌طور مؤثری پردازش کرده و به وظایف تخصصی رسیدگی کند.

GPUها از زمان طراحی اولیه خود فراتر رفته و اکنون برای تسریع محاسباتی که شامل مقادیر عظیم داده است، بسیار ارزشمند شده‌اند. این ویژگی‌ها آنها را برای کارهایی مانند یادگیری ماشین، محاسبات علمی و رایانش با عملکرد بالا مناسب کرده است.

« انقلاب در پردازش هوش مصنوعی »

در دنیای امروز، که فناوری به سرعت در حال پیشرفت است، هوش مصنوعی به عنوان یکی از مهم‌ترین محرک‌های تغییرات اجتماعی و اقتصادی شناخته می‌شود. با افزایش حجم داده‌ها و نیاز به پردازش سریع و کارآمد آنها، شتاب‌دهنده‌های هوش مصنوعی به خط مقدم این انقلاب فناوری پیوسته‌اند.

شتاب‌دهنده‌های هوش مصنوعی، از جمله واحدهای پردازش عصبی «NPU»، به عنوان نسل جدید سخت‌افزارهای تخصصی، با هدف تسریع در برنامه‌های هوش مصنوعی و یادگیری ماشین طراحی شده‌اند. این فناوری‌های نوین با توانایی پردازش داده‌های با دقت پایین و معماری‌های نوآورانه، کاربردهای گسترده‌ای در رباتیک، اینترنت اشیا و محاسبات ابری پیدا کرده‌اند. از گوشی‌های هوشمند گرفته تا سرورهای ابری، این شتاب‌دهنده‌ها در حال تغییر چهره فناوری‌های روزمره ما هستند و در آینده‌ای نزدیک نقش بیشتری در دنیای دیجیتال ایفا خواهند کرد.

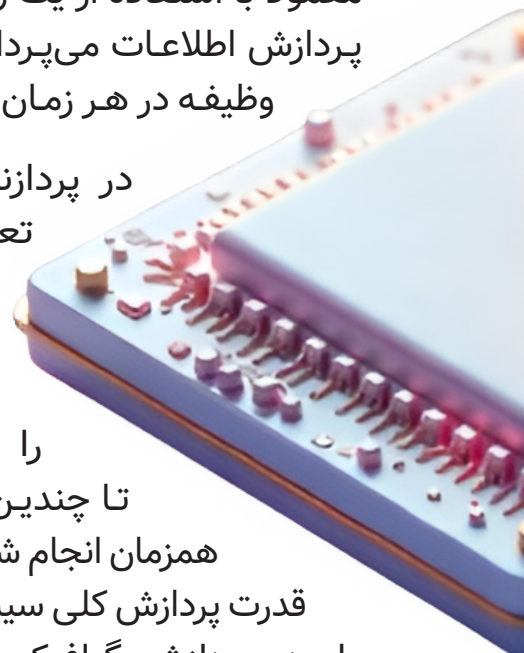
یک هنرمند خلاق، به پردازش و تولید تصاویر زیبا و واقع‌گرایانه کمک می‌کند.

در واقع، همکاری این دو واحد است که به رایانه‌ها قدرت و انعطاف‌پذیری می‌بخشد. هرکدام در حوزه خود بی‌نظیر هستند و می‌توانند تجربه‌ای غنی و بی‌نظیر را برای کاربران به ارمغان آورند. به این ترتیب، دنیای رایانه‌ها نه تنها دنیای داده‌ها، بلکه دنیای امکان‌ها و خلاقیت‌هاست. با هر بار فشردن یک کلید یا حرکت ماوس، دنیایی از ایده‌ها و نوآوری‌ها به واقعیت تبدیل می‌شود.

« مغزهای دیجیتالی »

واحد پردازش مرکزی^۱، که به عنوان "مغز" کامپیوتر شناخته می‌شود، مسئول اجرای دستورات در یک برنامه است. CPU معمولاً با استفاده از یک رویکرد سریالی به پردازش اطلاعات می‌پردازد و تنها بر یک وظیفه در هر زمان تمرکز می‌کند.

در پردازنده‌های مدرن، تعداد هسته‌ها معمولاً بین ۲ تا ۶۴ متغیر است که این امکان را فراهم می‌کند تا چندین وظیفه به‌طور همزمان انجام شود و به این ترتیب قدرت پردازش کلی سیستم افزایش یابد. واحد پردازش گرافیکی یا GPU، یک پردازنده تخصصی است که به‌طور اولیه برای بهبود کیفیت نمایش گرافیک‌ها در کامپیوترها، به‌ویژه گرافیک‌های سه‌بعدی، طراحی شده است. این واحد به‌طور



ستاپ باکس‌های عصبی؛ آینده خانه‌های هوشمند

ادغام هوش مصنوعی در خانه‌های هوشمند، تجربه زندگی روزمره را متحول کرده است و نسل جدید ستاپ باکس‌ها با «NPU»ها این تغییر را به اوج می‌رسانند. پردازنده‌های عصبی با توانایی پردازش موازی عملیات پیچیده هوش مصنوعی مانند تشخیص صدا و یادگیری ماشین، به کاربران این امکان را می‌دهند تا وظایف را به صورت محلی و بدون نیاز به اتصال مداوم به سرورهای ابری انجام دهند. این رویکرد نه تنها باعث کاهش زمان پاسخگویی و بهبود امنیت داده‌ها می‌شود، بلکه مصرف انرژی را نیز کاهش می‌دهد.

ستاپ باکس‌های مجهز به پردازنده‌های عصبی می‌توانند نقش مهمی در هوشمندسازی خانه‌ها ایفا کنند و به عنوان مرکز مدیریت دستگاه‌های متصل عمل کنند. با قدرت پردازشی ۲ تریلیون عملیات در ثانیه، این دستگاه‌ها از قابلیت‌های پیشرفته‌ای مانند تشخیص صدا در لحظه، تحلیل ویدئو برای امنیت خانه و بهینه‌سازی محتوا برخوردارند. پیشرفت‌های آینده در این حوزه می‌تواند توان پردازشی را به ۴۰ تریلیون عملیات در ثانیه برساند، که باعث افزایش کیفیت تصاویر، بهبود پاسخگویی و ارائه تجربه‌ای شخصی‌سازی شده‌تر خواهد شد.

در دنیایی که بهینه‌سازی مصرف انرژی و حفاظت از محیط زیست از اهمیت بالایی برخوردار است، استفاده از پردازنده‌های عصبی در ستاپ باکس‌ها، گامی مهم به سمت خانه‌های هوشمند و پایدارتر خواهد بود.

از TPU تا QPU

دنیای سخت‌افزارهای هوش مصنوعی با سرعت چشمگیری در حال گسترش است و پردازنده‌های جدیدی همچون TPU^۱، DPU^۲، VPU^۳، APU^۴ و QPU^۵ هر کدام نقش ویژه‌ای در این عرصه دارند. TPUها که توسط گوگل توسعه یافته‌اند، سرعت پردازش مدل‌های یادگیری عمیق را افزایش می‌دهند. DPUها به عنوان مدیران ترافیک داده‌ها در مراکز داده عمل می‌کنند و امنیت و کارایی شبکه را بهبود می‌بخشند. VPUها برای پردازش تصویر و ویدئو در زمان واقعی طراحی شده‌اند و در دستگاه‌های لبه کاربرد دارند. APUها که ترکیبی از CPU و GPU هستند، گزینه‌ای مقرون به صرفه برای کاربران عمومی و بازی‌بازان محسوب می‌شوند. و در نهایت، QPUها با استفاده از مکانیک کوانتومی، سرعتی بی‌نظیر را در محاسبات پیچیده به ارمغان می‌آورند و نویدبخش آینده‌ای درخشان برای هوش مصنوعی هستند.

مقایسه قدرت GPU و NPU

در دنیای رو به رشد هوش مصنوعی، انتخاب بین واحدهای پردازش گرافیکی و واحدهای پردازش عصبی به یک بحث داغ تبدیل شده است. GPUها به دلیل توانایی در پردازش موازی انبوه، برای آموزش مدل‌های یادگیری عمیق و مدیریت داده‌های بزرگ مناسب هستند. در مقابل، NPUها به طور خاص برای وظایف هوش مصنوعی طراحی شده‌اند و با مصرف انرژی کمتر و کارایی بالاتر، گزینه‌ای ایده‌آل برای دستگاه‌های لبه‌ای و کاربردهایی هستند که به بهره‌وری

شیدا ستاری

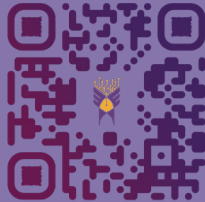
ورودی ۱۳۹۹ مهندسی کامپیوتر



منابع استفاده شده در نوشتار



برای کسب اطلاعات
بیشتر اسکن کنید



« جادوی مهندسی پرامپت

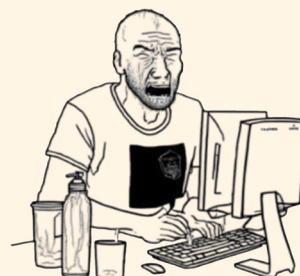
احتمالاً شما هم در استفاده از مدل‌های زبانی بزرگ^۱ با این مشکل مواجه شده‌اید: زمانی که درخواستی موسوم به prompt را به یک مدل می‌دهید، اما پاسخی که دقیقاً مدنظر شماست تولید نمی‌شود. گاهی حتی ممکن است مدل شروع به ارائه پاسخ‌های نامربوط یا هذیان‌گونه^۲ کند. این پدیده می‌تواند دلایل متعددی داشته باشد. برای مثال، ممکن است مشکل از مجموعه داده‌ای^۳ باشد که مدل بر اساس آن آموزش دیده، یا به دلایل فنی دیگری که اصلاح آن‌ها از دست کاربر خارج است. با این حال، یکی از مهم‌ترین عواملی که می‌تواند این مشکل را تشدید کند، نحوه نگارش درخواستی است که به مدل ارائه می‌دهیم. این عامل، برخلاف موارد دیگر، کاملاً در کنترل ماست و تأثیر قابل توجهی در کیفیت پاسخ‌ها دارد. مهندسی درخواست، Prompt Engineering، یکی از حوزه‌های نوظهور و رو به رشد در علم کامپیوتر و هوش مصنوعی است که به بهبود تعامل میان افراد غیرمتخصص و ماشین‌ها کمک شایانی کرده است.

انرژی نیاز دارند. انتخاب مناسب بین این دو به نیازهای خاص پروژه هوش مصنوعی شما بستگی دارد، اما هر دو در توسعه فناوری هوش مصنوعی نقش حیاتی دارند.

پی نوشت: دستگاه‌های لبه (Edge Devices) تجهیزاتی هستند که داده‌ها را همان جایی که تولید می‌شوند، پردازش می‌کنند. این کار باعث می‌شود سرعت بالا برود، مصرف اینترنت کمتر شود و امنیت اطلاعات بیشتر شود. برای مثال، دوربین‌های هوشمند، سنسورها، ماشین‌های خودران و ایستگاه‌های 5G از این دستگاه‌ها هستند.

در این قسمت قرار است با جادوی کلمات، هوش مصنوعی را رام کنیم!

Prompt Engineer



چند تا «ط» توی «قسطنطیه» هست؟
کدوم عدد بزرگ‌تره، ۹.۱۱ یا ۹.۹؟
کدوم سنگین‌تره، یک تُن پَر یا یک تُن آجر؟
چند تا کشور با حرف «الف» شروع می‌شه؟



توسعه داده شده‌اند، متصل شویم. اما یک چالش مهم وجود دارد؛ به دلیل تحریم‌های اعمال‌شده علیه ایران، امکان استفاده مستقیم از سرویس‌های OpenAI برای کاربران ایرانی وجود ندارد. برای حل این مشکل، می‌توانیم از یک سرویس‌دهنده ایرانی مانند Gilas.io استفاده کنیم. این سرویس به ما کمک می‌کند تا بدون نیاز به پروکسی، VPN، یا حتی داشتن اکانت خارجی، به مدل‌ها متصل شویم.

همان‌طور که در تصویر بالا می‌بینید، ابتدا محیط مجازی پایتون را فعال می‌کنیم، سپس کتابخانه OpenAI را نصب می‌کنیم. در نهایت، با وارد کردن کلیدی که از Gilas.io دریافت کرده‌ایم، ارتباط با سرویس برقرار می‌شود.

```
python-IAUbbmag

import os
from dotenv import load_dotenv
from openai import OpenAI
load_dotenv()
client = OpenAI(
    api_key=os.environ.get("GILAS_API_KEY"),
    base_url="https://api.gilas.io/v1/")
```

در گام بعدی، باید یک کلاینت با استفاده از پکیج OpenAI بسازیم. این کلاینت به ما امکان می‌دهد تا به مدل زبانی متصل شویم و با آن ارتباط برقرار کنیم. اما قبل از این مرحله، لازم است کلید سرویس گیلز را بارگذاری کنیم تا اتصال به سرویس برقرار شود. برای این کار از پکیج dotenv استفاده می‌کنیم. این پکیج به ما کمک می‌کند تا اطلاعات حساس، مثل کلید API، را در یک فایل env ذخیره کنیم و به‌صورت ایمن از آن بخوانیم. از آنجا که به‌جای استفاده مستقیم

تعریف رسمی که از مهندسی درخواست در منابع مربوطه آورده شده، عبارت است از فرایند طراحی، تنظیم و بهینه‌سازی درخواست‌های متنی^۱ که به مدل‌های هوش مصنوعی داده می‌شود، به گونه‌ای که این مدل‌ها قادر به تولید پاسخ‌هایی دقیق، مرتبط و مطلوب باشند. خیلی خلاصه و به دور از تعاریف، برداشت غیررسمی که از این تعریف می‌توان کرد این است که مهندسی درخواست برای کنترل کردن رفتار یک مدل زبانی می‌باشد که می‌توان به‌وسیله آن به مدل بگوییم ورودی ما را چگونه تفسیر کند و در قبال آن چه خروجی را تولید کند. هدف از نوشتن این متن در واقع ارائه روش‌ها و تکنیک‌هایی است که با استفاده از آن بتوان به بیشترین میزان بهره‌وری از جوابی که مدل تحویل می‌دهد، رسید. برای شروع، ابتدا باید محیط برنامه‌نویسی خود را آماده کنیم. در این پروژه از زبان برنامه‌نویسی پایتون استفاده می‌کنیم. علاوه بر آن، چند کتابخانه و یک سرویس‌دهنده نیز به‌کار گرفته می‌شوند تا امکان اتصال به یک مدل هوش مصنوعی فراهم شود.

```
python-IAUbbmag

## Activate virtual env
!python3 -m virtualenv .venv
!source .venv/bin/activate

## Install OpenAI package
!pip install openai

## Export Gilas.io API key
os.environ["GILAS_API_KEY"]='...'
```

اولین گام، نصب پکیج OpenAI است. این پکیج به ما امکان می‌دهد تا به مدل‌های بزرگ زبانی، که اغلب توسط شرکت OpenAI

تکنیک اول: اگر جواب واضحی از مدل انتظار دارید، به همان میزان هم واضح صحبت کنید!

مدل‌های بزرگ زبانی را به عنوان یک ای کیو سان تصور کنید که حجم عظیمی از دانش و اطلاعات جورواجور و مختلف را در خود جای داده است. اما برخلاف مغز انسان که با اتصالات و پیوندهای شیمیایی و الکتریکی کار می‌کند، بر پایه حجم عظیمی از دیتا به همراه شبکه‌ای از فرمول‌ها و الگوریتم‌های ریاضیاتی و منطقی طراحی شده تا با تمام توان بفهمد چه چیزی از او خواسته شده و باید چه هدفی را دنبال کند. اما با تمام این تفاسیر ذهن‌خوان نبوده و نمی‌تواند متوجه شود که دقیقاً چه منظور و هدفی پشت هر درخواست شما نهفته است.

در نتیجه، اگر پاسخی را احتیاج دارید که دقیقاً مطابق با درخواست‌های شما باشد، بایستی به صورت کاملاً شفاف برای آن مشخص کنید که در ازای گرفتن ورودی x انتظار دارید چه خروجی‌ای به شما تحویل دهد! مثلاً به جای گفتن «چیزی در مورد پرندگان بگو»، بهتر است بگویید: «یک پاراگراف کوتاه درباره تفاوت‌های رفتاری عقاب و شاهین به من بده.» به این شکل مدل دقیقاً می‌داند که چه اطلاعاتی را جمع‌آوری کند و چطور آن را ارائه دهد.

یکی از تکنیک‌هایی که می‌تواند در این زمینه کمک کند، استفاده از نشانه‌های جداکننده به نام delimiter است. با این روش می‌توانید بخش‌های مختلف درخواست خود را به طور شفاف برای مدل تفکیک کنید تا هم خوانایی و هم دقت را افزایش دهید.

از OpenAI، درخواست‌هایمان ابتدا از طریق یک سرویس ایرانی مثل Gilas.io ارسال می‌شود (به عنوان یک third-party provider)، باید آدرس base_url را تنظیم کنیم. این آدرس برای سرویس Gilas.io برابر است با: <https://api.gilas.io/v1>. با این تنظیمات، می‌توانیم به راحتی از خدمات Gilas.io استفاده کنیم و بدون دغدغه تحریم‌ها، به مدل‌های زبانی متصل شویم!

```
python-IAUbbmag

def get_completion(prompt, model="gpt-3.5-turbo"):
    messages = [{"role": "user", "content": prompt}]
    response = client.chat.completions.create(
        model=model,
        messages=messages,
        temperature=0, # this is the degree of
        randomness of the model's output
    )
    return response.choices[0].message.content
```

در گام آخر، ما یک تابع کمکی به نام get_completion تعریف می‌کنیم. این تابع ورودی‌ای به نام prompt دریافت می‌کند که در واقع همان درخواست یا سؤالی است که قصد داریم به مدل ارسال کنیم. در ابتدا، prompt به عنوان محتوای نقش کاربر (user role) تنظیم می‌شود. سپس، تابع پیام را با مقدار temperature=0 به مدل زبانی (LLM) ارسال می‌کند. تنظیم این پارامتر باعث می‌شود که مدل پاسخ‌های دقیق‌تر و با قطعیت بیشتری تولید کند، چرا که مقدار صفر به معنی کاهش میزان تصادفی بودن پاسخ‌هاست. در نهایت، خروجی تابع همان پاسخی است که مدل به درخواست ما ارائه می‌دهد. این ساختار ساده و کارآمد به ما امکان می‌دهد به راحتی درخواست‌هایمان را ارسال کرده و پاسخ‌های موردنظر را دریافت کنیم.



یکی از فرمت‌های داده‌ای که ساختاری خوش‌تعریف و برای ماشین‌ها خوانایی بالایی دارد، JSON نام دارد. در واقع، زمانی که داده‌ها را از یک سرویس واسط یا third-party service (در اینجا مدل زبانی) به صورت متنی دریافت می‌کنیم، این داده‌ها به شکل خام و غیرساختاریافته هستند و خوانایی بالایی ندارند. اما با استفاده از مهندسی درخواست، می‌توانیم در پرامپت خود مشخص کنیم که خروجی باید در قالبی ساختارمند و دقیق، مثل JSON، بازگردانده شود.

به عنوان مثال، در کد زیر می‌توانید ببینید که یک ورودی به مدل داده شده و از آن خواسته‌ایم خروجی را در قالب JSON ارائه دهد:

```
python-IAUbbmag

prompt = f"""
Generate a list of three made-up book titles along \
with their authors and genres. \
Provide them in JSON format with the following keys: \
book_id, title, author, genre.
"""
response = get_completion(prompt)
print(response)
```

در این شماره دو تکنیک را دیدیم و سعی کردیم با استفاده از آن‌ها از ظرفیت‌های مدل‌های زبانی به بهترین نحو ممکن استفاده کنیم. تکنیک‌های نامبرده شده ابزارهای قدرتمندی هستند که می‌توانند بهره‌وری مدل را به طرز چشمگیری افزایش دهند تا از بروز خطا تا جای ممکن جلوگیری کنند.

اما این تنها آغاز کار است! در شماره‌های آینده به سراغ معرفی استراتژی‌های پیشرفته‌تر خواهیم رفت تا به سطح بهتری از تعامل با این موجودات دیجیتالی هوشمند برسیم.

کاربرد دیگری که این نشانه‌های جداکننده دارند، جلوگیری از مفهومی به نام injection attack است. در واقع، اگر مدل نتواند تشخیص دهد که کدام بخش از پرامپت مربوط به ورودی کاربر است، ممکن است به اشتباه درخواست کاربر را به عنوان بخشی از برنامه اصلی خود در نظر بگیرد. این می‌تواند منجر به نتایج نادرست یا حتی رفتارهای ناخواسته از سوی مدل شود.

در تصویر زیر یک مهندسی درخواست با استفاده از نشانه‌های جداکننده به منظور خلاصه کردن یک متن نشان داده شده است:

```
python-IAUbbmag

text = f"""
باید اینکه می‌خواهید مدلی چه کاری انجام دهد را با ارائه دستورالعمل‌هایی که به قدر واضح و مشخص هستند، بیان کنید. این کار باعث هدایت مدل به سوی خروجی \
\مطلوب می‌شود و احتمال دریافت پاسخ‌های نامرتبط یا نادرست را کاهش می‌دهد. ترکیب نوشتن یک دستورالعمل \
\ واضح با نوشتن یک دستورالعمل کوتاه را با هم اشتباه نکنید. در بسیاری از موارد، دستورالعمل‌های بلندتر، \
وضوح و زمینه را برای مدل فراهم می‌کنند که می‌تواند به خروجی‌های جزئی و مرتبط تر منجر شود.
"""

prompt = f"""
Summarize the text delimited by triple backticks \
into a single sentence in Farsi language.
```{text}```
"""
prompt
```

## تکنیک دوم: از قالب بندی خروجی استفاده کنید

مدل‌های زبانی این امکان را دارند که خروجی خود را به فرمت‌ها و ساختارهای مختلفی تولید کنند. اهمیت این قابلیت زمانی بیشتر مشخص می‌شود که، برای مثال، از مدل زبانی در بخش بک‌اند یک وبسایت استفاده کنیم. در چنین سناریویی، خروجی مدل باید به گونه‌ای باشد که برنامه‌های کامپیوتری بتوانند به راحتی آن را تجزیه<sup>۱</sup> کنند.



## “ورودی های جدید بخوانند”



سپیده نوروزی مطلق



علی پروینی



### به هاگوارتز خوش آمدید!

درود بر شما! ورودتان را به دانشگاه، پس از سال‌ها تلاش و تحصیل را تبریک می‌گوییم. مطمئن باشید این تصمیم، یک نقطه عطف مهم در زندگی شما خواهد بود، بنابراین نگذارید هیچ‌کس باعث کم‌ارزش جلوه دادن تصمیم شما شود.

مسیری که در آن پا گذاشتید تفاوت‌چندانی با جادوگری ندارد. شما قرار است یاد بگیرید چطور با تغییر یک سنگ و اتصال یک جریان به آن باعث هوشمند شدن آن می‌شویم و با یک زبان خاص با آن صحبت می‌کنیم تا به ما در اتصال به هر نقطه از جهان و انجام کارها کمک کند. واقعاً این جادوگری نیست؟!

احتمالاً اولین احساسی که نسبت به ورود به دانشگاه داشتید هیجان بوده که البته با احتمال قریب به یقین می‌توانم بگویم بعد از ورودتان به سوله ورزشی برای ثبت‌نام، آن هیجان تازه اولش است! به هر حال، یاد گرفتن جادوی کامپیوترها و تبدیل شدن به یک مهندس کار ساده‌ای نیست، اما دور از دسترس هم نیست و با تلاش مداوم قابل دستیابی می‌باشد ولی حداقل جنبه مثبت ماجرا این است که شما قدم اول را برداشته‌اید! هویت دانشجو همان‌طور که



از اسمش پیدا است، جستن راه حل و دانش حل مسئله هاست. شما که دانشجوی این رشته هستید، بیشتر از همه باید به این مهارت عادت کنید. می توانم بگویم بیشترین کاری که شما در دانشگاه انجام می دهید حل مشکل است و این مشکل همیشه قرار نیست به تنهایی حل شود. ما در این بخش ویژه به قسمت کوچکی از راه حل های مشکلات شما اشاره می کنیم به امید اینکه کمی با خیال راحت تر قدم در این مسیر بگذارید.





## وحشت نکنید!

اول بیاید تا کمی بیشتر با دانشگاه خود آشنا شویم:  
در پایین نقشه دانشگاه را می توانید ببینید. البته همیشه یادتان باشد از  
پرسیدن نترسید. اگر جایی گم شدید روی کمک همه می توانید حساب کنید.



## قوانین دانشگاه خود را قورت دهید!

به طور کلی، دانشگاه را به عنوان یک ارگان در نظر بگیرید که مجموعه‌ای از پروتکل‌ها و ضوابط در آن حاکم است. قاعده‌تاً هرچه زودتر از این قوانین آگاه شوید، کمتر در موقعیت‌های مختلف دچار سردرگمی خواهید شد. پروتکل‌هایی مثل فارغ‌التحصیلی، سربازی پسران، انتخاب واحد و... را می‌توانید از سایت دانشگاه و پی‌دی‌اف‌های مختلفی که آنجا قرار داده شده و همچنین قسمت آموزش دانشکده متوجه شوید.

چارت نیز یکی از آن دسته ضوابط و ابزارهای کلیدی در برنامه‌ریزی دوران تحصیل شما است. در واقع، چارت آرایش ترمی دروسی است که قرار است در این چهار و اندی سال بخوانید. در دانشگاه، به جز ترم اول که مدیر گروه برای شما انتخاب واحد می‌کند، در ادامه این کار به عهده خودتان است و اگر در انتخاب واحد دقت کافی نداشته باشید و اشتباهی مرتکب شوید، همین اشتباه مصادف می‌شود با دوندگی در زمان فارغ‌التحصیلی!

اولین گام در مسیر انتخاب واحد، مشخص کردن گرایش تحصیلی است. به طور کلی، رشته مهندسی کامپیوتر در مقطع کارشناسی به سه گرایش مهندسی سخت‌افزار، مهندسی نرم‌افزار و IT (فناوری اطلاعات) ارائه می‌شود که در دانشگاه ما می‌توانید

بین گرایش‌های نرم‌افزار و آی‌تی یکی را برای دوره چهار ساله کارشناسی پیوسته یا دو ساله کارشناسی ناپیوسته انتخاب کنید. این انتخاب هم کار چندان پیچیده‌ای

نیست و تنها کافی است هنگام





انتخاب واحد، بر اساس چارت درسی گرایش موردنظرتان درس بردارید. گرایشی را که شما انتخاب می‌کنید تأثیری در مدرکتان ندارد و در آن ثبت نخواهد شد.

احتمالاً الان که چارت را باز کرده‌اید با یک سری خطوط درهم برخورد کرده باشید. در حقیقت، فلش‌هایی که از یک درس به درس دیگر رفته‌اند، نشان‌دهنده این هستند که درس اول پیش‌نیاز یا هم‌نیاز درس دوم است. به عبارت دیگر، شما باید ابتدا درس پیش‌نیاز را با موفقیت بگذرانید تا بتوانید در درس دوم ثبت‌نام کنید. برای مثال، اگر فلشی از «برنامه‌سازی پیشرفته» به «ساختمان داده» رفته باشد، این یعنی بدون گذراندن برنامه‌سازی پیشرفته، اجازه ثبت‌نام در ساختمان داده را نخواهید داشت.

علاوه بر این، برخی فلش‌ها ممکن است به‌عنوان هم‌نیاز مشخص شده باشند؛ این یعنی می‌توانید هر دو درس را به‌صورت هم‌زمان بگذرانید، اما توصیه می‌شود ترتیب پیشنهادی چارت را رعایت کنید تا در یادگیری مفاهیم به مشکل برنخورید. این ساختار چارت به گونه‌ای طراحی شده که دانشجویان بتوانند مفاهیم را به‌صورت پیوسته و گام‌به‌گام یاد بگیرند و برای درس‌های پیشرفته‌تر آماده شوند. بنابراین، مهم است که قبل از برنامه‌ریزی ترم جدید، چارت درسی و پیش‌نیازها را با دقت در اینترنت و سرفصل وزارت علوم بررسی کنید.

انتخاب واحد و مدیریت تحصیلی در دوران دانشگاه، ارتباطی تنگاتنگ با عملکرد شما در هر ترم دارد. همان‌طور که چارت درسی به شما کمک می‌کند تا مسیر تحصیلی‌تان را بر اساس پیش‌نیازها و هم‌نیازها تنظیم کنید، وضعیت





تحصیلی شما هم تأثیر مستقیمی بر نحوه استفاده از این چارت دارد. اگر انتخاب واحد به‌درستی انجام نشود یا اینکه دروسی را بیفتید، مشکلاتی پیش می‌آید که ممکن است مسیر تحصیلی را پیچیده‌تر کند.

آموزشیار، سامانه اصلی شما برای تمام کارهای مرتبط با دانشگاه است؛ از جمله انتخاب واحد، پرداخت شهریه، مشاهده نمرات و کارنامه. تنها رزرو تغذیه برای سلف از این سامانه انجام نمی‌شود و باید از اپ دانشجویار استفاده کنید.

یکی از این مشکلات، وضعیت مشروطی است. اگر معدل یک ترم شما کمتر از دوازده شود، در ترم بعد به‌عنوان دانشجوی مشروط شناخته می‌شوید و این به معنای محدود شدن تعداد واحدهای قابل انتخاب به چهارده واحد است. این محدودیت می‌تواند شما را از رعایت ترتیب چارت و گذراندن به‌موقع برخی دروس محروم کند. به‌عنوان مثال، اگر پیش‌نیاز یک درس مهم را به دلیل محدودیت واحدها نتوانید در یک ترم بگذرانید، این موضوع ممکن است به تأخیر در گذراندن درس‌های بعدی و حتی فارغ‌التحصیلی منجر شود.

ارتباط میان چارت درسی و وضعیت مشروطی بسیار مهم است. چارت درسی برای این طراحی شده که دانشجویان با یک ترتیب منطقی و پیوسته دروس را بگذرانند، اما اگر دانشجویی به دلیل ضعف عملکرد در وضعیت مشروطی قرار گیرد، این ترتیب به هم می‌ریزد و ممکن است مجبور شوید دروس بیشتری را در ترم‌های بعدی فشرده کنید. چنین فشاری می‌تواند دوباره منجر به افت تحصیلی و تداوم وضعیت مشروطی شود.

بنابراین، در انتخاب واحد باید علاوه بر رعایت پیش‌نیازها و هم‌نیازها، به تعداد واحدهای قابل انتخاب و توانایی خود در گذراندن آن‌ها نیز توجه کنید. اگر وضعیت تحصیلی شما مناسب نباشد، نه‌تنها اجرای صحیح چارت درسی دشوار می‌شود، بلکه ممکن است در شرایط پیچیده‌تری مانند مشروطی‌های متوالی گرفتار شوید که حتی می‌تواند به خطر اخراج از دانشگاه منجر شود.





## ۱. دانشگاه، میدان آزمون و کشف مسیر آینده

می‌دانم که شاید این عنوان کمی کلیشه‌ای به نظر برسد و برای دانشجوی ترم اولی که به‌تازگی از پشت میز و نیمکت و بدون تجربه آنچنانی وارد جو دانشگاه شده، یک امر سخت باشد. اما دانشگاه با یک بازه زمانی چهار-پنج ساله فرصت مناسبی است تا راه‌های مختلف را امتحان کنید و در نهایت به درک بهتری از آینده‌تان برسید. احتمالاً نتیجه این آزمون و خطا خلاصه بشود به حداقل یکی از این دو هدف: در حوزه مربوط به رشته‌تان شروع به کار کنید و یا اینکه به شکل مردانی از سرزمین پارس بروید تا ثریا علم را دنبال کنید! در هر صورت، دانشگاه جای خوبی برای محک زدن و آماده شدن برای هر دو هدف بالا می‌باشد. نظر من را بخواهید، قبل از تعیین این دو مورد، هم اصطلاحاً ریسرچ را تا حدودی امتحان کنید و هم فضای کار را از نزدیک لمس کنید (احتمالاً در درس کارآموزی تا حدودی بتوانید این را از نزدیک امتحان کنید)، چرا که بعضاً پیش می‌آید که بدون تجربه کردن، بچه‌ها هدفی را انتخاب می‌کنند و بعد که در اتمسفر و جو هدف قرار می‌گیرند، می‌بینند که ای داد بی‌داد ما اصلاً روحیاتمان همخوانی نداشت!



## ۲. آخر دانشگاه هم شد جا؟

شاید شما هم قضیه انصراف از تحصیل استیو جابز را شنیده باشید و رسانه سعی بر این داشته باشد که دانشگاه را محل بیهوده‌ای نشان دهد. اما واقعیت این است که شاید جای کاملاً بی‌نقصی نباشد، اما می‌تواند نقطه شروع خوبی برای کسانی که می‌خواهند هر دو هدف گفته‌شده در بالا را امتحان کنند، باشد. راستش را بخواهید از نظر من شاید کمتر جایی یا اصلاً هیچ‌جای دیگر دروس مورد نیاز یک مهندس کامپیوتر را به این دقت و ظرافت در کنار همدیگر قرار نداده باشد، هرچند در رشته ما انتقاداتی به قدیمی بودن دروس تدریس شده وارد باشد، اما هسته و شاخه اصلی علم کامپیوتر را می‌توانید در این درس‌ها ببینید.

## ۳. شبکه‌سازی و ارتباط ساختن را جدی بگیرید

دانشگاه یک جامعه کوچکی از افراد متخصص در زمینه تحصیلی‌تان و همچنین آدم‌هایی که مثل شما در یک مسیر هستند را گرد هم آورده. در نتیجه این فرصت خیلی خوبی است که بتوانید دایره‌ای از افراد هم‌هدف خودتان را شناسایی کنید. اساتید فقط برای درس دادن اینجا حضور ندارند، سعی کنید با آن‌ها ارتباط برقرار کنید و از تجربیاتشان استفاده کنید. مکان‌ها و تشکلهایی مثل کتابخانه و فمتو و مرکز رشد واحدهای فناور و کارآفرین و کانون‌های علمی فرهنگی و نشریه و... هم گزینه‌های خوبی هستند برای اینکه با دانشجویهای پرتلاش مثل خودتان زودتر آشنا شوید.





### ۴. خودآموزی و زبان

اگر نمی‌دانید که کدام فیلد از کامپیوتر به درد شما می‌خورد از تجربه کردن نترسید! با سرچ کردن mind map to learn sth در اینترنت با تعداد زیادی سایت مواجه می‌شوید که به شما دقیقاً نشان می‌دهند برای یادگیری یک موضوع خاص باید چه مسیری (Roadmap) را طی کنید. کمی حوزه‌های مختلف را امتحان کنید تا اوضاع دستتان بیاید. اما دریایی به عمق یک سانتی‌متر هم نباشید و از این شاخه به آن شاخه پریدن عادتتان نشود! مورد دیگر زبان انگلیسی است. احتمالاً وقتی دنبال یادگیری یک مطلب خاص هستید، متوجه خواهید شد احتمال اینکه در وب انگلیسی به نتیجه برسید خیلی بیشتر است. از طرفی ممکن است لازم شود متونی (از مقالات علمی تخصصی گرفته تا یک دیتاشیت) به همین زبان را بخوانید. در نتیجه، برای اینکه کمی گوگل ترنسلیت را راحت بگذارید و عملاً نشخوارکننده و وابسته به تفسیر و ترجمه دیگران نباشید، کم‌کم دانش زبانی خود را بالا ببرید. جدای از این‌ها، ماهیت کامپیوتر رشته‌ای است که با همفکری و مشورت با دیگران کارهایش را پیش می‌برد و اصطلاحاً open source است. شما علاوه بر یادگیری برای ارتباط مؤثر با دنیای اطرافتان و به روز بودن در زمینه کاری و درسی، حتماً به دانستن زبان نیاز دارید.

Yay!



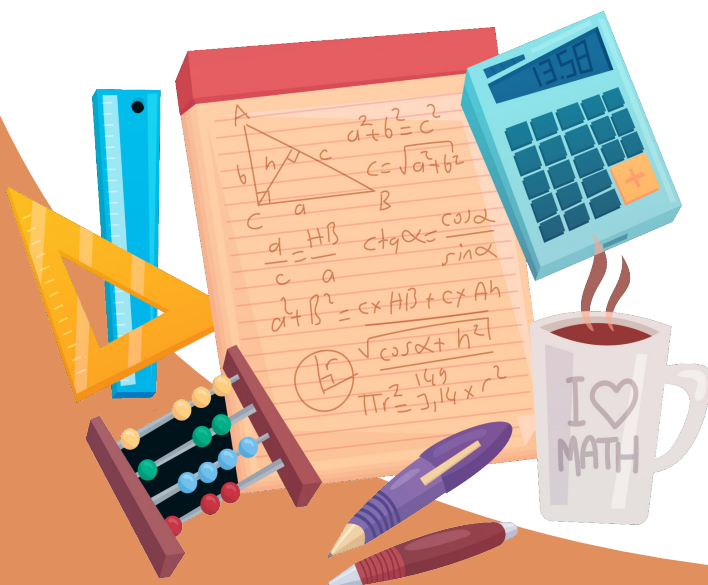
## ۵. کلاس‌هایتان را جدی بگیرید

در این مورد از من به شما نصیحت که حتی‌الامکان از کلاس درس غافل نشوید و درس را با استاد پیش ببرید. خیلی وقت‌ها ممکن است احساس کنید بودن سر کلاس چیزی به شما اضافه نمی‌کند، اما حداقل ذهنیتی که استاد از درس و مباحثش دارد در ذهن شما کم‌کم شکل می‌گیرد و می‌توانید با دیدی دقیق‌تر خودتان درس را خودآموز یاد بگیرید و دنبال یادگیری‌اش بروید. اگر هم می‌بینید که اوضاع خیلی خراب است، به رفرنسی که استاد توصیه کرده، کلاس درس و یا جزوات اساتید دیگر روی بیاورید! در نهایت، فرادرس، مکتب‌خونه و سایت‌های این‌چنینی هم می‌توانند گزینه‌های خوبی باشند!

دو توصیه کوچک:

- درس‌های ریاضیاتی چه پیوسته و چه گسسته را هم جدی بگیرید! شاید خیلی یقه شما را در کارشناسی نگیرند اما مطمئن باشید در رشته‌های ارشد مثل هوش مصنوعی و دیتا ساینس با آن‌ها خیلی کلنجار خواهید رفت. در نهایت اگر جایی به کارتان نیاید، قطعاً آن دید حل مسئله (problem solving) که ریاضیات به شما می‌دهد، همه‌جای زندگی‌تان به کارتان خواهد آمد.

- توصیه دیگر این است که در درس‌هایی که به برنامه‌نویسی مربوط هستند حتماً در کنار آموزش دیدن، پروژه زدن فراموش نشود! حتی شده پروژه‌های کوچک! این پروژه است که به شما برنامه‌نویسی، مهارت دیباگینگ و داشتن صبر ایوب را آموزش می‌دهد! جایی مثل گیت‌هاب یک سایت عالی است برای اینکه با پروژه‌های بقیه آشنا شوید و یا حتی در آن‌ها مشارکت داشته باشید.



### ۶. پایتون یا سی‌پلاس‌پلاس؟ مسئله این است...

راستش، زبان‌ها می‌آیند و می‌روند؛ این تفکر الگوریتمی است که می‌ماند! یک مشکلی که اغلب ترم اولی‌ها دارند این است که درگیر انتخاب یک زبان برنامه‌نویسی برای ابد الدهر می‌شوند! الان وقتش نیست که نگران زبان برنامه‌نویسی باشید، مفاهیم بنیادی تقریباً بین همه آن‌ها یکسان است و بهتر است به جای آن، نگران تفکر الگوریتمی باشید. سعی کنید در چهار درس مبانی، برنامه‌سازی پیشرفته، ساختمان داده و الگوریتم ایده اصلی برنامه‌نویسی کردن را دنبال کنید و تا آن موقع به زبان به‌عنوان یک ابزار برای پیاده‌سازی این موارد نگاه کنید. این که در آینده چه پیش بیاید هم خدا عالم است. ان شاء الله که خیر است! فقط هر کاری می‌کنید تعصب نداشته باشید چرا که علم مهندسی در نهایت یک علم تجربی و پر از تغییر است و بحث کردن بر سر خوب یا بد بودن یک ابزار شما را به هدف‌تان که همان حل مسئله است، نخواهد رساند. مورد دوم این است که برنامه‌نویسی یک ابزار مهم در جعبه‌ابزار کامپیوتر است اما تنها ابزارش نیست! وظیفه نهایی شما به عنوان یک مهندس حل چالش‌های دنیای واقعی است و برای انجام این کار باید در کنار برنامه‌نویسی، به مرور زمان درک عمیقی از مفاهیم کامپیوتر مثل سیستم دیجیتال، شبکه، پایگاه داده و سیستم عامل به دست بیاورید.





## ۷. آرامش در میان طوفان!

هر مسیر جدیدی با چالش‌های خودش همراه است، دانشگاه هم از این قاعده مستثنی نبوده و نیست. رویارویی با هر چالش و حل آن خودش یک مشکل است حال اگر با چاشنی استرس و اضطراب ترکیب شود که می‌شود قوز بالای قوز! من یکی خودم از آن استرسی‌های روزگار هستم و از هر پروژه کوچک یک غول بی‌شاخ و دم درست می‌کنم و گاهی به نقطه‌ای می‌رسم که اصلاً انجامشان نمی‌دادم تا خدایی ناکرده با استرس هم روبرو نشوم.

فلسفه‌ای که من برای استرس دارم اینست که باید آن را پذیرفت و دلیلش را به درستی درک کرد، چراکه در این صورت تبدیل می‌شود به یک نیروی محرکه که ما را به حرکت وا می‌دارد. در نتیجه استرس را ریشه‌یابی کنید و ببینید دلیل آن چیست. نتیجه‌ای که من در این مورد گرفتم اینست که غالباً هسته مشترک تمامی استرس‌ها نداشتن یک برنامه‌ریزی واقع‌بینانه و تصویر شفاف ذهنی از مسیر است. قبل از رسیدگی به چالش به یک درک درست و دنباله‌ای از کارهایی که باید انجام دهید تا آن را به اتمام برسانید، برای خود متصور شوید.

شما با این کار انگار طبق یک الگوریتم Divide and Conquer (تقسیم و حل) مشکل خود را به یک سری ریز مشکل می‌شکنید که حلشان ساده‌تر است و از آن غول بی‌شاخ و دم خلاص می‌شوید. بدین‌وسیله ذهن آرام‌تری خواهید داشت و می‌دانید که باید چه کارهایی را انجام دهید. در مرحله دوم برای حل این ریز مشکل‌ها یک برنامه‌ریزی درست و واقع‌بینانه‌ای که از پس آن برمی‌آیید، بریزید. در این مرحله متاسفانه نمی‌توانم بگویم از ساعت ۹ تا ۱۰ و نیم چه بخوانید و چه کار کنید! برنامه‌ریزی یک چیز کاملاً شخصی‌سازی شده است که



بایستی در طی زمان و با آزمون و خطا به آن برسید و به تدریج پرورشش دهید. اما به طور کلی هدف از برنامه‌ریزی اینست که بدانیم چه کاری در چه زمانی انجام شود تا در بازه زمانی X به نتیجه مطلوب برسیم. نکته دیگری که باید در مورد برنامه‌ریزی رعایت کنید اینست که قابل انعطاف باشد چرا که زندگی پر از عدم قطعیت هست و شما به عنوان یک انسان با کلی رویداد از پیش تعیین نشده مواجه می‌شوید. من به شخصه اطلاعات مهمی مثل تاریخ امتحانات، تحویل پروژه‌ها و تکالیف را در جایی مثل Google Calendar ثبت می‌کنم و بعد بر اساس این تواریخ یک برنامه کلی می‌ریزم. در نهایت بر اساس این برنامه کلی و با در نظر گرفتن رویدادهای از پیش تعیین نشده‌ای که انتظارش را نداشتم اما باید انجامشان بدهم، برنامه‌ای برای چند روز آینده‌ام در نظر می‌گیرم.

اگر آدم کاغذی هستید (راستش این فحش نیست ولی کلمه دیگری به ذهنم نرسید!) از یک دفتر استفاده کنید، اما حالا که توفیق اجباری نصیبتان شده و رشته‌تان کامپیوتر است، به Trello و Google Keep, Evernote, Notion, Todoist یک سری بزنید.



## ۸. خانه دوم

اصلی‌ترین قسمت زندگی کسی که از راه دوری می‌آید و مهمان این شهر و دانشگاه می‌شود، خوابگاه است. بدون مقدمه برویم سراغ چالش‌های جدید و متفاوتی که قرار است به عنوان یک خوابگاهی با آن‌ها مواجه شوی. باید این را بدانی که هم‌اتاقی‌هایت دست کمی از خانواده تو ندارند. پس بهتر است یاد بگیری در عین اینکه حساسیت‌های خودت را با دقت و حوصله به آن‌ها توضیح می‌دهی، با حساسیت‌های آن‌ها هم آشنا شوی. در عین پایبند بودن به اصول خودت انعطاف‌پذیر باشی و یادت باشد شما از جای‌جای ایران دور هم جمع شدید و هر کدامتان دنبال هدفی آمدید، پس سعی کن با چالش‌های خانواده جدیدت کنار بیایی.

این خانه و خانواده جدید برای همه بچه‌ها چه درونگرا چه برونگرا، پر از تجربه‌های جدید و جالب است؛ از آشپزی‌های دورهمی و شب‌نشینی‌ها گرفته تا دور هم درس خواندن‌ها و در سر و کله هم زدن‌ها. یادت باشد که در خوابگاه به‌جز خانواده، کلی همسایه باحال هم داری که از سراسر ایران آمدند و برای اینکه خاطره خوبی از دوران خوابگاهی بودندت ثبت کنی، بهتر است ارتباط‌های خوبی با آن‌ها بسازی. برای تشکیل این ارتباط مثلاً می‌توانی بیشتر روی نکات مثبت خودت و دوستانت دست بگذاری. هر وقت مسئله‌ای بینتان پیش آمد به عنوان اولین اقدام با یک فلاسک چای و دوتا شیرینی پیش دوستت بنشین و با او صحبت کن. هر ارتباطی بین تو و خانواده جدیدت هست را خود شما تشکیل می‌دهید،



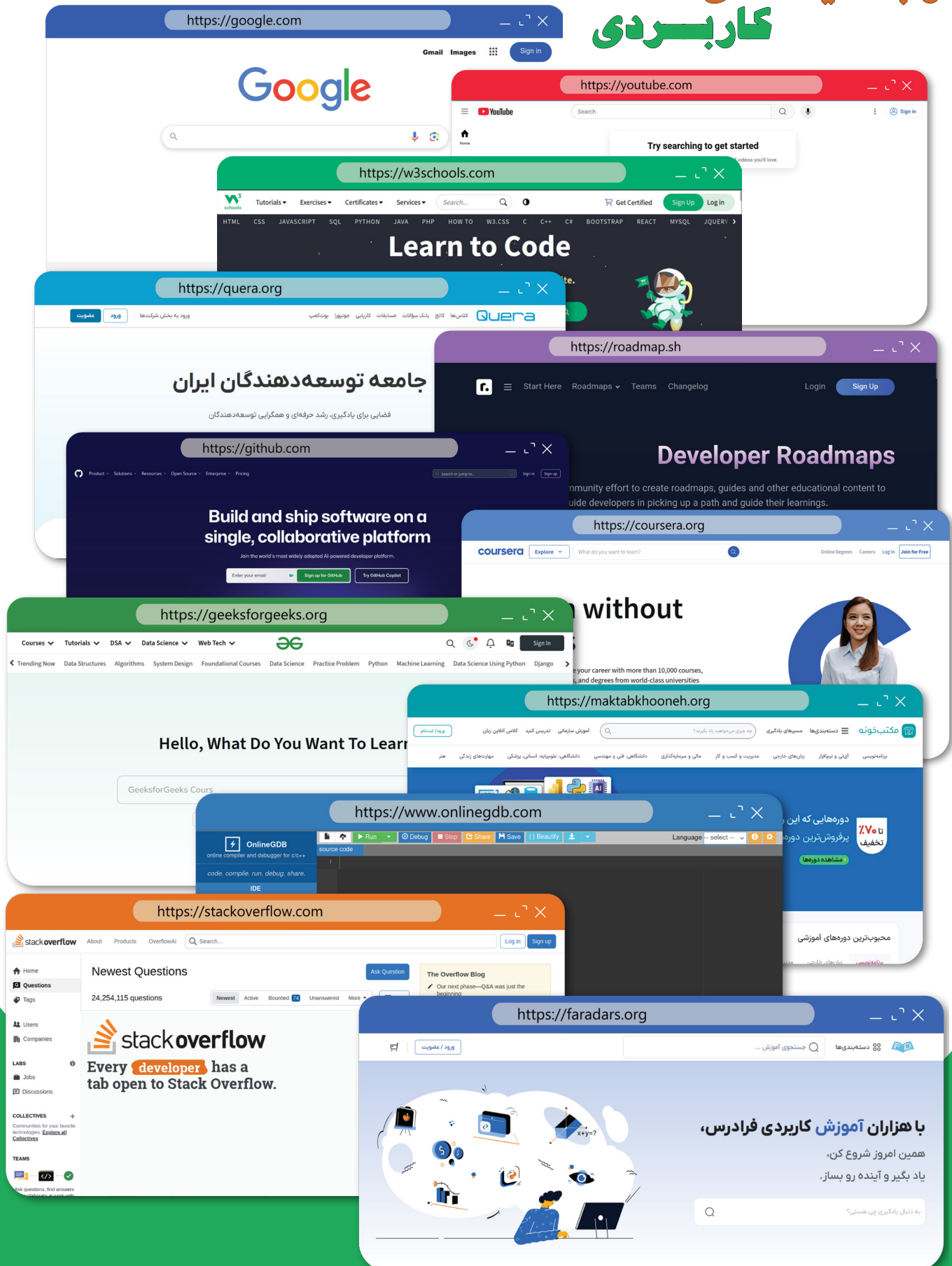


پس حواست حسابی جمع باشد. به جای پافشاری و اصرار روی حرفها (البته به جز قوانین و خط قرمز) یا به خصوص پشت هم حرف زدن که خیلی کار ناپسندیده‌ای است، بهتر است موقع اختلافها و بحثها کمی حوصله کنی و سعی کنی بیشتر کوتاه بیایی، اینگونه نتایج بهتری هم می‌گیری. به علاوه آن آستانه تحمل و مهارت زیست‌یات هم بسیار تقویت می‌شود. اما اگر صحبت نتیجه نداشت، به عنوان گزینه آخر می‌توانی اتاقت را به صورت مسالمت‌آمیز عوض کنی.

در آخر به یاد داشته باشید که موفقیت حاصل تلاشی پیوسته و متعادل است. اشتباه کردن، بخشی طبیعی از یادگیری است و هر قدمی که برمی‌دارید، شما را به انسانی تواناتر تبدیل می‌کند. از کلاس‌هایتان بهره بگیرید، با چالش‌ها روبه‌رو شوید و پروژه‌ها را به فرصتی برای رشد تبدیل کنید. دانشگاه فقط محلی برای آموختن درس‌ها نیست، بلکه جایی برای ساختن آینده و کشف توانایی‌های واقعی شماست. از این سفر لذت ببرید و با اطمینان به پیش بروید!



# وب سایت های کاربردی



# مسابقه طراحی: شیداستاری

شما، مأمور ویژه دولت و برنامه‌نویسی حرفه‌ای، مأموریتی فوق‌محرمانه دریافت کرده‌اید: یافتن و بازیابی پرونده‌ای سری که امنیت ملی را تهدید می‌کند. پس از پیگیری‌های متعدد و نفوذ به چند پایگاه اطلاعاتی، متوجه می‌شوید که این پرونده درون یک گاوصندوق فوق امنیتی در یک مرکز رهاشده نظامی پنهان شده است. با رسیدن به محل، شما در برابر گاوصندوق قرار می‌گیرید. روی گاوصندوق متنی نامفهوم حک شده که در نگاه اول احتمالا بی‌معنی به نظر می‌رسد:

NI JLIwyyx, SIO GOMN yPuFOuNy u MyLCyM Iz HOGvyLM uHx xyNyLGCHy zIL  
yuwB QByNByL CN CM u LCABN-wIOJFyx HOGvyL.U HOGvyL R CM wIHMCxyLyx  
LCABN-wIOJFyx Cz SIO wuH zCHx NQI JIMCNCpy CHNyAyLM u uHx v MOwB  
NBuN:0 < u ≤ v ≤ Ru × v = Ru / v ≥ 0.5cz MOwB u JuCL yRCMNM, NByH NBy  
HOGvyL CM LCABN-wIOJFyx uHx SIOL LyMJIHMy MBIOFx vy 1.iNByLQCMY,  
LyMJIHx QCNB 0.fyN'M vyACH QCNB NBy zIFFIQCHA HOGvyLM:66 ,55 ,120 ,37

حس امنیتی‌تان هشدار می‌دهد: با یک رمز سزار با شیف ۲۰ روبه‌رو هستید. تنها راه باز کردن گاوصندوق، رمزگشایی این متن و تحلیل مسئله‌ای است که در پس آن نهفته است. با پیاده‌سازی الگوریتم سزار و حل دقیق مسئله، می‌توانید اعداد لازم برای باز کردن قفل را بیابید. زمان محدود است و سرنوشت در دستان شماست.

## نحوه شرکت و ارسال پاسخ‌ها

پوشه‌ای در درایو شخصی خود بسازید، به ایمیل نشریه دسترسی بدهید و موارد زیر را داخل آن قرار دهید:

۱. فایل کدهای لازم برای حل مسابقه
۲. ویدئویی از توضیح نحوه رسیدن به پاسخ

سپس لینک پوشه را همراه اطلاعات زیر به یکی از ایمیل‌های نشریه بفرستید:

**نام و نام خانوادگی، شماره دانشجویی، شماره تماس**

برای مشاهده جزئیات بیشتر و کپی کردن کد مسابقه، بارکد زیر را اسکن کنید.

[info@bbmag.ir](mailto:info@bbmag.ir)

[bbmag.ir@gmail.com](mailto:bbmag.ir@gmail.com)



به شرکت‌کنندگانی که موفق به رمزگشایی پیام نهایی شوند، به قید قرعه جوایزی اهدا خواهد شد.





“

## کلام آخر

خیلی خوشحال هستیم که تا به اینجا همراه ما بودید. امیدوارم مطالب را مفید یافته و از آنها استفاده کرده باشید. حالا که تا به اینجا آمدید پس بدانید که شما هم جزئی از این دنیای پرماجرا شدید! اینجا فضایی برای شماست، جایی که می‌توانید ایده‌ها و دانشتان را به اشتراک بگذارید، از علم و هنر بگویید، و خلاقیت خود را به نمایش بگذارید. فرقی نمی‌کند که عاشق تحلیل‌های عمیق باشید یا طراحی‌های گرافیکی چشم‌نواز، اینجا می‌تواند بستری برای رشد و تعامل باشد. تا دیدار بعد، پرسشگر بمانید، خلاق باشید، و هیچ‌وقت از یادگیری دست نکشید!

”

با ما در ارتباط باشید:



[company/bbmag](https://www.linkedin.com/company/bbmag)



[IAUbbmag](https://t.me/IAUbbmag)



[bbmag.ir@gmail.com](mailto:bbmag.ir@gmail.com)



[bbmag.ir/fa](http://bbmag.ir/fa)



+98 937 850 6125



هوش ما چیزی است که ما را انسان می‌سازد،  
و هوش مصنوعی گسترش دهنده آن کیفیت است.

یان لی کان - دانشمند علوم رایانه

