دانشگاه صنعتی اصفهان

دانشکده علوم ریاضی

پروژه کارشناسی

انقلاب هوش مصنوعی و ظهور مکانیزم توجه

واکاوی مقاله

Attention is All You Need

نگارش:

محمدجواد عبدالهی وشنوه‌ئی

استاد راهنما:

دکتر بهناز عمومی

پاییز ۱۴۰۴

چکیده

در این گزارش، به تحلیل مقاله "توجه تمام آن چیزی است که نیاز دارید" اثر تحقیقات تیمی از پژوهشگران گوگل و برخی دانشگاه‌ها بوده که یکی از نقاط عطف مهم در زمینه یادگیری عمیق و شبکه‌های عصبی به شمار می‌آید. این مقاله با معرفی معماری ترنسفورمر، وابستگی به شبکه‌های عصبی بازگشتی را به طور کامل حذف کرده و مدلی مبتنی بر مکانیزم توجه ارائه می‌دهد که نه تنها به نتایج پیشرفته‌تری در وظایف ترجمه ماشینی دست یافت، بلکه به دلیل قابلیت موازی‌سازی بالا، فرآیند آموزش ماشین را نیز به شدت تسریع بخشیده است. در این پروژه، ابتدا به بررسی محدودیت‌های مدل‌های پیشین پرداخته، سپس اجزای اصلی معماری ترنسفورمر شامل ضرب داخلی مقیاس‌شده، توجه چند سر و رمزگذاری موقعیتی را تشریح کرده و در نهایت به بررسی نتایج و تأثیرات گسترده این مقاله بر حوزه پردازش زبان طبیعی می‌پردازیم.

فهرست مطالب

مقدمه ۳

۱ عصر بازگشتی و چالش‌های زبانی (دنیای پیش از ۲۰۱۷) ۴
۱.۱ الگو پردازش متوالی: صف‌بندی کلمات ۴
۲.۱ محدودیت‌های ذاتی مدل‌های بازگشتی: تراژدی فراموشی ۵
۱.۲.۱ مشکل اول: پدیده محو شدن گرادیان ۵
۲.۲.۱ مشکل دوم: گلوگاه پردازش سریالی ۵
۳.۱ تلاش‌های اصلاحی: ظهور حافظه‌های پیشرفته ۶
۴.۱ بن‌بست اطلاعاتی در معماری رمزگذار–رمزگشا ۶

۲ انقلاب توجه و معماری ترنسفورمر ۷
۱.۲ مفهوم توجه در ادراک انسانی ۷
۱.۱.۲ تمثیل اثر مهمانی کوکتل ۷
۲.۲ تولد ترنسفورمر: شکستن زنجیر زمان ۸
۱.۲.۲ حذف کامل پردازش سریالی ۸
۲.۲.۲ جایگزینی حافظه با مکانیزم توجه ۸

۳ کالبدشکافی معماری ترنسفورمر
گذر از انتزاع به ریاضیات ۱۰
۱.۳ معماری مبتنی بر پشته‌های رمزگذار و رمزگشا ۱۰
۲.۳ توکن‌گذاری و تعبیه‌ی کلمات: تبدیل واژه به مختصات معنایی ۱۱
۱.۲.۳ مفهوم توکن: واحد بنیادین پردازش ۱۱
۲.۲.۳ لایه‌ی تعبیه‌ی کلمات: نگاشت به فضای برداری پیوسته ۱۲
۱.۲.۲.۳ ویژگی‌های ریاضی فضای تعبیه ۱۲
۳.۳ کدگذاری موقعیتی: حل معمای آشفتگی زمانی ۱۲
۴.۳ مکانیزم توجه خودکار: قلب تپنده مدل ۱۳
۱.۴.۳ تشریح فنی با مدل سه‌گانه پرس‌وجو، کلید و مقدار ۱۴
۱.۱.۴.۳ فرمول ریاضی توجه ضرب‌داخلی مقیاس‌شده ۱۴

۲.۱.۴.۳ فرآیند محاسبه توجه ۱۴
۵.۳ توجه چندسر: نگرش چندبعدی به زبان ۱۵
۱.۵.۳ تحلیل نقش ماتریس‌های وزن و تبدیل خطی ۱۵
۶.۳ شبکه‌ی پیش‌خور و اتصالات میان‌بر: تثبیت یادگیری ۱۶
۱.۶.۳ تحلیل ساختاری و هندسی لایه پیش‌خور ۱۶
۲.۶.۳ اتصالات میان‌بر و نرمال‌سازی: تضمین پایداری ریاضی ۱۷
۷.۳ تولید خروجی نهایی: لایه‌ی خطی و تابع نرم‌بیشینه ۱۷
۸.۳ تحلیل توپولوژیک و جریان داده در معماری ترنسفورمر ۱۹
۱.۸.۳ مسیر جریان در پشته رمزگذار (سمت چپ) ۱۹
۲.۸.۳ مسیر جریان در پشته رمزگشا (سمت راست) ۱۹
۳.۸.۳ مکانیزم توجه متقاطع: اتصال دو نیمکره ۲۰

۴ چرا ترنسفورمر جهان را تغییر داد؟ ۲۲
۱.۴ پیروزی موازی‌سازی و هم‌افزایی با قانون مور ۲۲
۲.۴ مدیریت وابستگی‌های بافاصله: حذف محدودیت‌های مکانی ۲۳
۳.۴ قوانین مقیاس‌پذیری: بزرگتر یعنی هوشمندتر ۲۳

۵ عصر پسا ترنسفورمر و ظهور غول‌های زبانی ۲۴
۱.۵ تبارشناسی مدل‌ها: سه شاخه تکاملی ۲۴
۱.۱.۵ معماری‌های فقط رمزگذار[1] ۲۴
۲.۱.۵ معماری‌های فقط رمزگشا[2] ۲۴
۳.۱.۵ معماری‌های ترکیبی رمزگذار رمزگشا[3] ۲۵
۲.۵ جدول زمانی تکامل: رشد نمایی پارامترها ۲۵
۳.۵ فراتر از متن: تعمیم به بینایی و مدل‌های چندوجهی ۲۶

نتیجه‌گیری ۲۷

منابع ۲۹

واژه نامه انگلیسی به فارسی ۳۲

واژه نامه فارسی به انگلیسی ۳۷

فهرست اختصارات ۴۲

[1] Encoder Only
[2] Decoder Only
[3] Encoder Decoder

مقدمه: سپیده‌دم عصر نوین ادراک ماشینی

در پهنه‌ی گسترده‌ی تاریخ تحولات تکنولوژیک، لحظاتی وجود دارند که مرز میان «پیشین» و «پسین» را چنان پررنگ و قاطع ترسیم می‌کنند که بازگشت به دوران ماقبل آن‌ها، ناممکن می‌نماید. همان‌گونه که اختراع ماشین چاپ، کشف الکتریسیته و ابداع ترانزیستور مسیر تمدن بشری را تغییر دادند، در دنیای پیچیده‌ی هوش مصنوعی[1] و به‌طور خاص در حوزه‌ی تخصصی پردازش زبان طبیعی (NLP)[2] نیز شاهد چنین نقطه عطفی بوده‌ایم.

انتشار مقاله‌ی جریان‌ساز «توجه تمام آن چیزی است که نیاز دارید[3]» در سال ۲۰۱۷ توسط تیمی از پژوهشگران برجسته‌ی گوگل، حکمی مشابه را در این عرصه داشت. این پژوهش نه تنها یک معماری نوین تحت عنوان ترنسفورمر[4] را به جهان علم معرفی کرد، بلکه الگو حاکم بر تعامل ماشین با زبان انسان را از ریشه دگرگون ساخت.

این تغییر بنیادین، زیربنای ظهور مدل‌های زبانی قدرتمند و فراگیر زیر شد که امروزه تار و پود زندگی دیجیتال بشر را احاطه کرده‌اند و مرزهای میان خلاقیت ماشینی و انسانی را کمرنگ ساخته‌اند.

Generative Pre-trained Transformer (GPT): این خانواده از مدل‌ها که بر پایه مکانیزم تولیدی بنا شده‌اند، در ساختن متن‌های منسجم و شبیه به انسان تبحر دارند. تمرکز اصلی آن‌ها بر پیش‌بینی کلمه بعدی در یک دنباله است و به همین دلیل در وظایفی همچون نگارش خلاق، ترجمه و مکالمه، عملکردی خیره‌کننده از خود به نمایش می‌گذارند.

Bidirectional Encoder Representations from Transformers (BERT): برخلاف مدل‌های پیشین که متن را تنها در یک جهت پردازش می‌کردند، این مدل با رویکردی دوطرفه به زبان نگاه می‌کند. این قابلیت به ماشین اجازه می‌دهد تا بافت و معنای کلمات را با توجه به واژگان قبل و بعد آن‌ها به صورت هم‌زمان درک کند که این امر منجر به انقلابی در وظایف درک مطلب و طبقه‌بندی متون شده است.

هدف غایی این گزارش پژوهشی، ارائه‌ی یک تحلیل جامع، عمیق و در عین حال شفاف برای مخاطبی است که لزوماً پیش‌زمینه‌ی فنی سنگینی در حوزه‌ی یادگیری عمیق[5] یا هوش مصنوعی ندارد. ما در این نوشتار، سفری اکتشافی را از دوران حکمرانی الگوریتم‌های کلاسیک و محدودیت‌های ساختاری آن‌ها آغاز می‌کنیم؛ دورانی که در آن ماشین‌ها در درک روابط طولانی و پیچیده‌ی کلامی ناتوان بودند. سپس به لحظه‌ی تولد معماری ترنسفورمر می‌رسیم و با زبانی ساده اما دقیق، مکانیزم‌های درونی آن را کالبدشکافی می‌کنیم تا دریابیم چگونه مفهوم «توجه» جایگزین محاسبات خطی و زمان‌بر شد.

این گزارش صرفاً توصیف وقایع تاریخی نیست، بلکه تلاشی تحلیلی است برای درک چرایی و چگونگی وقوع انقلابی که به ماشین‌ها اجازه داد از سطح «خواندن» کلمات عبور کرده و به لایه‌های عمیق‌تر «درک» مفهوم، بافت کلام و ظرافت‌های پنهان زبانی دست یابند. ما بررسی خواهیم کرد که چگونه این جهش تکنولوژیک، راه را برای مدل‌هایی هموار کرد که قادرند شعر بگویند، کد بنویسند و استدلال کنند.

---

[1] Artificial Intelligence
[2] Natural Language Processing (NLP)
[3] Attention Is All You Need
[4] Transformer
[5] Deep Learning

فصل ۱

عصر بازگشتی و چالش‌های زبانی (دنیای پیش از ۲۰۱۷)

برای درک عمیق از ابعاد بنیادین و عظمت تحولی که مقاله «توجه تمام آن چیزی است که نیاز دارید» در پیکره هوش مصنوعی ایجاد کرد، پیش‌نیاز اساسی آن است که وضعیت علم و فناوری را در دوران پیش از آن واکاوی کنیم. باید بدانیم مهندسان و دانشمندان علوم داده با چه ابزارهایی با پدیده زبان برخورد می‌کردند و چرا این ابزارها برای دستیابی به سطوح عالی ادراک ماشینی ناکافی و ناکارآمد بودند.

زبان انسان، برخلاف بسیاری از داده‌های ساختاریافته، پدیده‌ای ذاتا پیچیده، سیال، ابهام‌آمیز و شدیداً وابسته به بافت است. برخلاف داده‌های تصویری که در آن‌ها پیکسل‌های مجاور معمولاً دارای همبستگی معنایی و ساختاری هستند، در زبان طبیعی، ارتباط میان کلمات می‌تواند در طول جملات طولانی پراکنده باشد و ساختاری غیرخطی را در بستری خطی پنهان کند.

۱.۱) الگو پردازش متوالی: صف‌بندی کلمات

تا پیش از سال ۲۰۱۷، الگو غالب در حوزه پردازش زبان طبیعی بر پایه یک فرض بنیادین و شهودی استوار بود:

«زبان یک توالی زمانی است»

منطق این فرض ساده بود؛ هنگامی که انسان صحبت می‌کند یا متنی را می‌نگارد، کلمات در بستر زمان جاری می‌شوند؛ کلمه اول پیش از کلمه دوم، و کلمه دوم پیش از کلمه سوم قرار می‌گیرد. بر این اساس، دانشمندان به این نتیجه رسیدند که مدل‌های ماشینی نیز باید متن را دقیقاً به همین ترتیب، یعنی به صورت کلمه به کلمه و متوالی پردازش کنند.

این رویکرد منجر به ظهور و توسعه خانواده‌ای از الگوریتم‌ها تحت عنوان شبکه‌های عصبی بازگشتی (RNN)[1] گردید. این شبکه‌ها با الهام‌گیری از مکانیزم حافظه کوتاه‌مدت در مغز انسان طراحی شده بودند. ایده محوری این بود که شبکه، کلمه نخست را دریافت کرده، پردازش می‌کند و عصاره اطلاعات آن را در یک حافظه داخلی به نام

[1] Recurrent Neural Network (RNN)

«حالت پنهان[1]» ذخیره می‌نماید. سپس این اطلاعات به مرحله بعد (گام زمانی بعدی) منتقل می‌شود تا در کنار کلمه دوم پردازش شود. به عبارت دیگر، درک شبکه از کلمه جاری، تابعی از خود کلمه و دانشی بود که از کلمات پیشین کسب کرده بود.

۲.۱) محدودیت‌های ذاتی مدل‌های بازگشتی: تراژدی فراموشی

اگرچه ظهور Recurrent Neural Network (RNN) گامی بلند نسبت به مدل‌های آماری کلاسیک محسوب می‌شد، اما این معماری با چالش‌های ساختاری و مهلکی دست‌وپنج نرم می‌کرد که مانع از پیشرفت جدی در وظایف پیچیده‌ای نظیر ترجمه متون طولانی یا درک اسناد حقوقی و علمی می‌شد.

۱.۲.۱) مشکل اول: پدیده محو شدن گرادیان

بزرگ‌ترین چالش فنی در آموزش RNNها، ناتوانی آن‌ها در حفظ و انتقال اطلاعات در طول توالی‌های طولانی بود. برای تقریب ذهنی، بازی «تلفن‌بازی» را در نظر بگیرید؛ نفر اول پیامی را در گوش نفر دوم زمزمه می‌کند و این زنجیره ادامه می‌یابد. هرچه صف طولانی‌تر باشد، پیام نهایی دچار تحریف بیشتری شده و شباهت خود را به پیام اصلی از دست می‌دهد.

در شبکه‌های عصبی بازگشتی، هنگام استفاده از الگوریتم آموزش پس‌انتشار[2]، اتفاقی مشابه رخ می‌دهد. سیگنال‌های خطا که باید شبکه را اصلاح کنند، در حین حرکت به عقب در زمان (از انتهای جمله به ابتدا)، در اعداد کوچک‌تری ضرب می‌شوند و به تدریج کوچک و کوچک‌تر می‌گردند تا جایی که عملاً صفر یا محو می‌شوند. این پدیده که در ادبیات فنی "محو شدن گرادیان[3]" نامیده می‌شود، باعث می‌شود شبکه نتواند ارتباط میان کلماتی را که فاصله زیادی از یکدیگر دارند، درک کند. این ناتوانی در مدیریت وابستگی‌های طولانی‌برد[4] نقطه‌ضعف مدل‌های بازگشتی بود.

به عنوان مثال، در جمله: "آن گربه‌ای که هفته پیش در حیاط پشتی خانه همسایه قدیمی ما زیر درخت بلوط نشسته بود و با توپی قرمز بازی می‌کرد، **گرسنه است**." برای اینکه مدل دریابد چه کسی گرسنه است، باید کلمه "گربه" را که ده‌ها کلمه قبل‌تر آمده، در حافظه داشته باشد. RNNهای ساده در برقراری این ارتباطات طولانی شکست می‌خوردند و ممکن بود به اشتباه تصور کنند که "درخت" یا "توپ" گرسنه است.

۲.۲.۱) مشکل دوم: گلوگاه پردازش سریالی

چالش دوم ماهیتی محاسباتی داشت. در یک شبکه بازگشتی، پردازش کلمه دهم از نظر ریاضیاتی **نمی‌تواند** آغاز شود مگر آنکه پردازش کلمه نهم به اتمام رسیده باشد. این وابستگی زمانی[5] دقیقاً مانند یک مسابقه دوی امدادی است؛ دونده دوم باید منتظر بماند تا دونده اول چوب را به او برساند.

[1] Hidden State
[2] Backpropagation
[3] Vanishing Gradient
[4] Long-Range Dependencies
[5] Temporal Dependency

این ویژگی با روند پیشرفت سخت‌افزارهای کامپیوتری در تضاد کامل بود. کارت‌های گرافیک مدرن یا همان Graphics Processing Unit (GPU) ها، برای انجام هزاران محاسبه پردازش موازی[1] طراحی شده‌اند، نه محاسبات سریالی. ساختار خطی RNN اجازه نمی‌داد که از قدرت عظیم پردازش موازی استفاده شود، زیرا هر مرحله معطل مرحله قبل بود. این گلوگاه باعث می‌شد آموزش مدل‌های بزرگ روی حجم عظیمی از داده‌ها، بسیار کند، زمان‌بر و پرهزینه باشد.

۳.۱) تلاش‌های اصلاحی: ظهور حافظه‌های پیشرفته

پژوهشگران برای غلبه بر مشکل فراموشی، نسخه‌های تکامل‌یافته‌تری از شبکه‌های بازگشتی را با نام‌های Long Short-Term Memory (LSTM) و Gated Recurrent Unit (GRU) توسعه دادند. این معماری‌ها به مکانیزم‌هایی هوشمند مجهز بودند که مانند یک دروازه[2] عمل می‌کردند؛ آن‌ها قابلیت این را داشتند که به صورت فعال تصمیم بگیرند چه اطلاعاتی را در حافظه نگه دارند و چه اطلاعاتی را دور بریزند.

اگر RNN ساده را مانند شخصی در نظر بگیریم که سعی می‌کند همه چیز را طوطی‌وار حفظ کند و در نهایت گیج می‌شود، LSTM مانند شخصی است که یک دفترچه یادداشت ساختاریافته دارد و تنها نکات کلیدی را ثبت می‌کند. این بهبود معماری باعث شد که عملکرد سیستم‌ها در ترجمه ماشینی و تشخیص گفتار در سال‌های ۲۰۱۴ تا ۲۰۱۶ جهشی چشمگیر داشته باشد. با این حال، مشکل دوم یعنی عدم امکان پردازش موازی همچنان پابرجا بود. مدل‌ها دقیق‌تر شده بودند، اما کماکان کند بودند و مقیاس‌پذیری پایینی داشتند.

۴.۱) بن‌بست اطلاعاتی در معماری رمزگذار–رمزگشا

استاندارد طلایی آن دوران برای ترجمه ماشینی، استفاده از معماری رمزگذار–رمزگشا[3] مبتنی بر شبکه‌های بازگشتی بود. در این ساختار، بخش رمزگذار وظیفه داشت جمله ورودی را کلمه به کلمه بخواند و تمام معنا و مفهوم آن را در یک بسته ریاضی فشرده و با ابعاد ثابت به نام بردار زمینه[4] فشرده‌سازی کند. سپس بخش رمزگشا می‌بایست تنها با اتکا به همین بسته فشرده، جمله را به زبان مقصد بازسازی نماید.

تصور کنید از شما بخواهند تمام محتوا، احساسات و جزئیات یک فصل از یک رمان را تنها در یک جمله خلاصه کنید و آن را به فرد دیگری بدهید تا او عین متن اصلی را بازنویسی کند! طبیعتاً بسیاری از ظرایف در این فرآیند فشرده‌سازی از بین می‌روند. این محدودیت که به گلوگاه اطلاعاتی[5] شهرت یافت، نشان می‌داد که فشرده‌سازی تمام اطلاعات یک جمله طولانی در یک بردار ثابت، روشی بهینه برای انتقال معنا نیست. این دقیقاً همان نقطه‌ای بود که نیاز به رویکردی انقلابی همچون مکانیزم توجه احساس می‌شد.

---

[1] Parallel Processing
[2] Gate
[3] Encoder-Decoder Architecture
[4] Context Vector
[5] Information Bottleneck

فصل ۲

انقلاب توجه و معماری ترنسفورمر

در فضای علمی سال‌های پیش از ۲۰۱۷ که مملو از چالش‌های محاسباتی و محدودیت‌های حافظه در مدل‌های بازگشتی بود، انتشار مقاله «توجه تمام آن چیزی است که نیاز دارید» بسان یک صاعقه بر پیکره تحقیقات هوش مصنوعی فرود آمد. عنوان این مقاله، خود به تنهایی یک مانیفست علمی و بیانیه‌ای جسورانه علیه الگوهای سنتی حاکم بود.

نویسندگان این مقاله با طرح یک ادعای رادیکال، بیان کردند که برای توسعه مدل‌های زبانی قدرتمند و پیشرفته، دیگر نیازی به استفاده از ساختارهای پیچیده و زمان‌بر RNN یا Convolutional Neural Networks (CNN) نیست. آن‌ها استدلال کردند که می‌توان تمام معماری‌های پیشین را کنار گذاشت و تنها بر یک مفهوم شناختی تمرکز کرد:

«مکانیزم توجه[1] »

این تغییر رویکرد، نقطه عطفی بود که مسیر تکامل یادگیری عمیق را برای همیشه تغییر داد.

۱.۲) مفهوم توجه در ادراک انسانی

پیش از آنکه به تشریح ریاضیاتی و فنی معماری جدید بپردازیم، ضروری است مفهوم انتزاعی توجه را در بستر شناخت انسانی واکاوی کنیم. سیستم عصبی و شناختی انسان دارای ظرفیت محدودی برای پردازش اطلاعات است و نمی‌تواند تمام داده‌های ورودی از محیط (بصری، شنیداری، حسی) را با دقت یکسان و به صورت همزمان پردازش کند. در عوض، مغز ما مجهز به فیلترهای هوشمندی است که به طور ناخودآگاه بر روی بخش‌های مهم تمرکز کرده و سایر اطلاعات را نادیده می‌گیرد.

۱.۱.۲) تمثیل اثر مهمانی کوکتل

روانشناسان شناختی پدیده‌ای را شناسایی کرده‌اند که بهترین مدل برای تبیین مکانیزم توجه است و به «اثر مهمانی کوکتل[2] » شهرت دارد. برای درک این پدیده، تصور کنید در یک مهمانی شلوغ، پرهیاهو و مملو از صداهای

[1] Attention Mechanism
[2] The Cocktail Party Effect

متداخل حضور دارید. ده‌ها نفر به صورت همزمان در حال گفتگو هستند، موسیقی با صدای بلند پخش می‌شود و صدای برخورد ظروف نیز شنیده می‌شود.

در چنین محیط آشفته‌ای، سیستم شنوایی و مغز شما این قابلیت شگفت‌انگیز را دارد که روی صدای دوستتان که در مقابل شما ایستاده است قفل کند و تمام صداهای دیگر را به عنوان نویز پس‌زمینه[1] فیلتر نماید. اما نکته جالب‌تر اینجاست که اگر ناگهان از آن سوی سالن، کسی نام شما را صدا بزند، با وجود اینکه تمرکز شما کاملاً معطوف به دوستتان بود، بلافاصله توجهتان به سمت آن صدا جلب می‌شود (تغییر ناگهانی کانون توجه).

این قابلیت مغز برای تخصیص وزن[2] یا وزن‌دهی به ورودی‌های مختلف، اساس کار مدل‌های جدید هوش مصنوعی است. در این فرآیند:

- صدای دوست شما دارای وزن بالا (اهمیت زیاد) است.
- صدای محیط دارای وزن پایین (اهمیت کم) است.

مکانیزم توجه در هوش مصنوعی دقیقاً تلاشی ریاضیاتی برای شبیه‌سازی همین رفتار است: توانایی مدل برای اینکه یاد بگیرد در هر لحظه، کدام بخش از داده‌ها مهم است و کدام بخش‌ها باید نادیده گرفته شوند.

۲.۲) تولد ترنسفورمر: شکستن زنجیر زمان

معماری نوظهوری که در مقاله سال ۲۰۱۷ معرفی شد و ترنسفورمر نام گرفت، دو تغییر ساختاری و بنیادین را در نحوه پردازش داده‌ها ایجاد کرد. این تغییرات باعث شد محدودیت‌های مدل‌های بازگشتی که در فصل پیشین به آن‌ها اشاره شد، کاملاً مرتفع گردند.

۱.۲.۲) حذف کامل پردازش سریالی

مهم‌ترین دستاورد معماری ترنسفورمر، گسستن زنجیر وابستگی زمانی بود. برخلاف RNN که مجبور بود کلمات را به صورت متوالی (یکی پس از دیگری) بخواند، ترنسفورمر تمام جمله یا حتی یک پاراگراف کامل را به صورت یکجا و همزمان پردازش میکند.

این تغییر الگو به معنای حذف گلوگاه پردازشی بود. در این معماری، دیگر نیازی نیست برای شروع پردازش کلمه آخر، منتظر اتمام پردازش کلمات ابتدایی بمانیم. این ویژگی امکان استفاده حداکثری از قابلیت پردازش موازی در سخت‌افزارهای مدرن (GPU) را فراهم کرد. نتیجه این امر، افزایش خیره‌کننده سرعت آموزش مدل‌ها بود که اجازه داد دانشمندان مدل‌هایی با میلیاردها پارامتر را در زمانی معقول آموزش دهند.

۲.۲.۲) جایگزینی حافظه با مکانیزم توجه

دومین تغییر بزرگ، تغییر در نحوه مدیریت اطلاعات بود. در مدل‌های قدیمی مانند LSTM، شبکه تلاش می‌کرد خلاصه اطلاعات گذشته را در یک حافظه محدود و فشرده با نام (Hidden State) با خود حمل کند. اما ترنسفورمر این رویکرد را کنار گذاشت.

[1] Background Noise
[2] Weight Allocation

در معماری ترنسفورمر، مفهوم حافظه خطی جای خود را به دسترسی سراسری[1] داد. مدل در هر لحظه از پردازش، به **تمام** کلمات جمله دسترسی مستقیم دارد. مکانیزم توجه به مدل اجازه می‌دهد که برای درک هر کلمه، نگاهی به کل جمله بیندازد و تصمیم بگیرد کدام کلمات دیگر برای تفسیر کلمه فعلی ضروری هستند. این یعنی فاصله فیزیکی کلمات در جمله دیگر اهمیتی ندارد؛ کلمه اول می‌تواند به راحتی و با همان دقت کلمه دوم، با کلمه صدم ارتباط برقرار کند.

[1] Global Access

فصل ۳

کالبدشکافی معماری ترنسفورمر
گذر از انتزاع به ریاضیات

پس از بررسی مبانی نظری و چرایی نیاز به تغییر الگو، اکنون زمان آن فرا رسیده است که ساختار درونی معماری ترنسفورمر را واکاوی کنیم. پرسش بنیادین این است: ماشینی که ماهیت درک آن صرفاً بر پایه عملیات باینری و محاسبات ماتریسی است، چگونه قادر به استخراج ظرایف معنایی از ادبیات کلاسیک یا تحلیل پیچیدگی‌های متون حقوقی می‌باشد؟ پاسخ در معماری مدولار و سلسله‌مراتب ریاضیاتی این مدل نهفته است.

۱.۳) معماری مبتنی بر پشته‌های رمزگذار و رمزگشا

ساختار کلان ترنسفورمر بر پایه الگوی کلاسیک Sequence to Sequence (Seq2Seq) بنا شده است، اما با پیاده‌سازی متفاوت. این معماری از دو مؤلفه یا پشته[۱] اصلی تشکیل شده است:

۱. **پشته رمزگذار**[۲]: این بخش به عنوان تحلیل‌گر عمل می‌کند. وظیفه آن دریافت رشته ورودی (متن خام) و نگاشت آن به یک فضای نمایش انتزاعی پیوسته است. رمزگذار با خواندن متن و تحلیل روابط میان کلمات، یک درک عمیق ریاضی از محتوا ایجاد می‌کند.

۲. **پشته رمزگشا**[۳]: این بخش نقش تولیدگر را ایفا می‌کند. رمزگشا با دریافت خروجی رمزگذار (که حاوی عصاره معنایی متن است)، اقدام به تولید خروجی نهایی (مانند ترجمه متن یا پاسخ به سوال) به صورت گام‌به‌گام می‌نماید.

[1] Stack
[2] Encoder
[3] Decoder

۲.۳) توکن‌گذاری و تعبیه‌ی کلمات: تبدیل واژه به مختصات معنایی

ماشین‌ها فاقد درک ذاتی از زبان هستند و تنها اعداد را پردازش می‌کنند. گام نخست در فرآیند پردازش، نگاشت کلمات گسسته به بردارهای پیوسته در یک فضای ریاضی چندبعدی است که به آن تعبیه کلمات[۱] اطلاق می‌شود.

در این فرآیند، هر کلمه به یک بردار[۲] (لیستی از اعداد اعشاری) تبدیل می‌شود. ویژگی کلیدی این فضا آن است که فاصله هندسی بردارها با شباهت معنایی کلمات همبستگی دارد. به عنوان مثال، در این فضای برداری، مختصات کلمه "پادشاه" به "ملکه" و "سیب" به "پرتقال" بسیار نزدیک است و یا اختلاف مختصات کلمات "مادر" و "دختر" با اختلاف کلمات "پدر" و "پسر" یکسان است. این تبدیل، زیربنای درک ماشین از روابط معنایی واژگان است. پیش از آنکه مدل بتواند کلمات را به بردارهای ریاضی تبدیل کند، متن خام باید به واحدهای معنادار کوچکتری به نام توکن[۳] تقسیم شود. این فرآیند که توکن‌گذاری[۴] نامیده می‌شود، پل ارتباطی میان زبان انسانی و دنیای اعداد است.

۱.۲.۳) مفهوم توکن: واحد بنیادین پردازش

توکن‌ها لزوماً کلمات کامل نیستند؛ بسته به الگوریتم مورد استفاده، یک توکن می‌تواند یک کلمه، یک پیشوند، یک پسوند یا حتی تک‌حروف باشد.

- **چرا از کلمه کامل استفاده نمی‌شود؟** اگر هر کلمه یک توکن واحد باشد، مدل در مواجهه با کلمات جدید یا مشتقات پیچیده مانند "ورشکستگی‌ها" دچار مشکل می‌شود. این پدیده باعث بزرگ شدن بی‌رویه فضای دایره لغات[۵] و افزایش پارامترهای مدل می‌گردد.
- **رویکرد زیر–واحد**[۶]: مدل‌های مدرن متن را به قطعات کوچکتری می‌شکنند. به عنوان مثال، کلمه "بی‌برنامه" ممکن است به دو توکن "بی" و "برنامه" تقسیم شود. این کار به مدل اجازه می‌دهد با تعداد محدودی توکن در دایره لغات ثابت، بی‌نهایت کلمه و ترکیب جدید را درک و معنایابی کند.

پس از آنکه متن به توکن‌ها تجزیه شد، هر توکن به یک عدد صحیح منحصربه‌فرد (شناسه[۷]) اختصاص می‌یابد. اما از آنجا که این اعداد صرفاً شناسنامه هستند و هیچ رابطه ریاضی با هم ندارند (مثلاً عدد اختصاص یافته به کلمه "گربه" لزوماً نزدیک به عدد "سگ" نیست)، گام بعدی نگاشت این اعداد به یک فضای برداری پیوسته[۸] یا همان لایه تعبیه[۹] است.

---

[1] Word Embeddings
[2] Vector
[3] Token
[4] Tokenization
[5] Vocabulary
[6] Sub-word
[7] ID
[8] Continuous Vector Space
[9] Embedding

۲.۲.۳) لایه‌ی تعبیه‌ی کلمات: نگاشت به فضای برداری پیوسته

پس از مرحله توکن‌گذاری، ما مجموعه‌ای از اعداد صحیح (شناسه‌ها) داریم. اما این اعداد فاقد ارزش ریاضی هستند؛ برای مثال عدد اختصاص یافته به کلمه «مرد» و «زن» ممکن است بسیار دور از هم باشد، در حالی که از نظر معنایی به هم نزدیک‌اند. لایه‌ی تعبیه کلمات[۱] وظیفه دارد هر توکن را به یک بردار با ابعاد ثابت ($d_{model} = 512$) در یک فضای پیوسته تصویر کند.

۱.۲.۲.۳) ویژگی‌های ریاضی فضای تعبیه

این لایه صرفاً یک جدول جستجو[۲] بزرگ با ابعاد $V \times d_{model}$ است که در آن V نشان‌دهنده‌ی تعداد کل توکن‌های لغت‌نامه است. ویژگی‌های کلیدی این لایه عبارتند از:

۱. **تشابه معنایی و فاصله اقلیدسی[۳]:** در این فضای ۵۱۲ بعدی، بردارها به گونه‌ای آموزش می‌بینند که توکن‌های با معنای مشابه، در فاصله نزدیک‌تری از یکدیگر قرار بگیرند. برای مثال، تفاضل برداری میان "ملکه" و "زن" باید تقریباً معادل تفاضل برداری میان «پادشاه» و «مرد» باشد.

۲. **توزیع اطلاعات[۴]:** هر بعد از این بردار ۵۱۲ بعدی، جنبه‌ای از مفهوم توکن را رمزگذاری می‌کند؛ برای نمونه یک بعد ممکن است مربوط به زمان، بعدی دیگر مربوط به جنسیت و موارد مشابه باشد، هرچند که این ابعاد برای انسان مستقیماً قابل تفسیر نیستند.

۳. **هم‌مقیاس شدن[۵] متوازن:** نویسندگان مقاله بردار کلمات را در $\sqrt{d_{model}}$ ضرب می‌کنند. این کار باعث می‌شود مقادیر بردار کلمات با مقادیر «کدگذاری موقعیتی» هم‌مقیاس شوند تا هنگام جمع شدن، اطلاعات مکانی باعث محو شدن اطلاعات معنایی نشود.

در حقیقت، لایه‌ی تعبیه، کلمات را از یک بازنمایی گسسته و بی‌معنا به یک نمایش هندسی[۶] غنی تبدیل می‌کند که در آن مفاهیم زبانی با استفاده از روابط برداری قابل محاسبه هستند.

۳.۳) کدگذاری موقعیتی: حل معمای آشفتگی زمانی

یکی از چالش‌های ناشی از حذف پردازش سریالی (که در فصل قبل اشاره شد)، از دست رفتن مفهوم بود. از آنجا که مدل ترنسفورمر تمام کلمات را به صورت همزمان پردازش می‌کند، ذاتاً نمی‌تواند تفاوت میان دو جمله "علی زد حسن را" و "حسن زد علی را" را تشخیص دهد، زیرا مجموعه کلمات در هر دو یکسان است.

برای غلبه بر این چالش، نویسندگان مقاله راهکاری هوشمندانه تحت عنوان رمزگذاری موقعیتی[۷] ارائه دادند. در این روش، اطلاعات مربوط به جایگاه هر کلمه در جمله، از طریق جمع کردن بردار کلمه با یک بردار موقعیت

[1] Word Embedding
[2] Lookup Table
[3] Euclidean Distance
[4] Information Distribution
[5] Scaling
[6] Geometric Representation
[7] Positional Encoding

خاص (PE)، به مدل تزریق می‌شود. نویسندگان به جای استفاده از اعداد صحیح ساده که در جملات طولانی باعث ناپایداری محاسباتی[1] و رشد نامحدود مقادیر می‌شوند، از توابع مثلثاتی با فرکانس‌های متفاوت استفاده کردند.

فرمول‌های ریاضی این کدگذاری برای هر جایگاه pos و هر بعد[2] i به صورت زیر تعریف می‌شوند:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{۱.۳}$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \tag{۲.۳}$$

در این روابط، pos نشان‌دهنده موقعیت توکن در دنباله و i نشان‌دهنده شاخص بعد در فضای نمایش است. انتخاب این ساختار سینوسی و کسینوسی بر پایه یک استدلال منطقی بنا شده است:

این توابع سینوسی و کسینوسی به مدل اجازه می‌دهند تا با درک فرکانس سیگنال هر کلمه، جایگاه نسبی و مطلق آن را در جمله شناسایی کند.

از منظر تحلیل ریاضی، مهم‌ترین ویژگی این فرمول‌بندی آن است که برای هر فاصله ثابت k، بردار موقعیت PE_{pos+k} می‌تواند به صورت یک تبدیل خطی[3] از بردار PE_{pos} بیان شود. این ویژگی برای یک مدل ریاضیاتی حیاتی است، زیرا به شبکه عصبی اجازه می‌دهد به سادگی یاد بگیرد که چگونه با توجه به فواصل نسبی[4]، به اطلاعات مکانی[5] میان کلمات دسترسی پیدا کند. در واقع، با این روش، مدل نه تنها می‌داند کلمه در کجا قرار دارد، بلکه نسبت فاصله آن با سایر کلمات را نیز با دقت ریاضی بالایی درک می‌کند.

۴.۳) مکانیزم توجه خودکار: قلب تپنده مدل

محوری‌ترین نوآوری این مقاله، مکانیزم توجه خودکار[6] است. این مکانیزم به کلمات اجازه می‌دهد تا با یکدیگر تعامل داشته باشند و میزان وابستگی معنایی خود را در یک بافتار مشخص کنند.

برای تبیین این مفهوم، جمله زیر را در نظر بگیرید: "**بانک** پول نداشت چون **آن** ورشکسته شده بود."

ما به عنوان انسان می‌دانیم که ضمیر "آن" به "بانک" اشاره دارد. اما برای ماشین، کلمه "آن" می‌تواند از نظر ساختاری به "پول" نیز اشاره داشته باشد. مکانیزم توجه خودکار، هنگام پردازش کلمه "آن"، تمام کلمات دیگر جمله را اسکن می‌کند تا مرتبط‌ترین واژه را بیابد. در اینجا، وجود کلمه "ورشکسته" که صفتی برای نهادهای مالی است، باعث می‌شود مدل وزن توجه بالایی را میان "آن" و "بانک" برقرار کرده و ابهام ضمیر را برطرف سازد.

[1] Computational Instability
[2] Dimension
[3] Linear Transformation
[4] Relative Distances
[5] Spatial Information
[6] Self-Attention

۱.۴.۳) تشریح فنی با مدل سه‌گانه پرس‌وجو، کلید و مقدار

نویسندگان برای مدل‌سازی ریاضی این تعامل، از مفاهیم بازیابی اطلاعات در پایگاه‌های داده الهام گرفتند. برای هر کلمه، سه بردار مجزا از طریق ضرب بردار ورودی در سه ماتریس وزن آموزش‌پذیر (W^Q, W^K, W^V) تولید می‌شود:

پرس‌وجو[۱] **(Q):** نشان‌دهنده سیگنالی است که کلمه برای جستجوی اطلاعات ارسال می‌کند.

کلید[۲] **(K):** نقش شناسه یا برچسب کلمه را برای سایر کلمات ایفا می‌کند.

مقدار[۳] **(V):** حاوی محتوای اصلی و عصاره معنایی کلمه است.

۱.۱.۴.۳) فرمول ریاضی توجه ضرب‌داخلی مقیاس‌شده

فرآیند محاسبه توجه در مقاله تحت عنوان توجه ضرب داخلی مقیاس‌شده[۴] معرفی شده است. این عملیات به صورت ماتریسی با فرمول زیر بیان می‌گردد:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{۳.۳}$$

در این معادله، ضرب داخلی QK^T میزان شباهت میان پرس‌وجوها و کلیدها را محاسبه می‌کند. اما نکته حائز اهمیت برای تحلیل ریاضی، حضور عبارت $\frac{1}{\sqrt{d_k}}$ است. نویسندگان استدلال کردند که در ابعاد بالا (d_k بزرگ)، حاصل‌ضرب داخلی می‌تواند به اعداد بسیار بزرگی ختم شود. این بزرگی مقادیر باعث می‌شود تابع سافت‌مکس[۵] وارد نواحی تخت با گرادیان[۶] بسیار کوچک شود که اصطلاحاً به آن اشباع شدن[۷] می‌گویند. تقسیم بر $\sqrt{d_k}$ باعث نرمال‌سازی و پایداری واریانس این مقادیر شده و فرآیند یادگیری را بهبود می‌بخشد.

در نهایت، با اعمال تابع سافت‌مکس، وزن‌های توجه به‌دست می‌آیند که به صورت یک ترکیب خطی[۸] روی بردارهای مقدار اعمال می‌شوند تا نمایش نهایی کلمه در آن بافت خاص شکل بگیرد.

۲.۱.۴.۳) فرآیند محاسبه توجه

فرآیند ریاضیاتی توجه را می‌توان به یک مکالمه تشبیه کرد:

۱. کلمه "آن" (به عنوان پرس‌وجو) سیگنالی ارسال می‌کند: «من به دنبال مرجعی هستم که قابلیت ورشکستگی داشته باشد.»

۲. کلمه "بانک" (به عنوان کلید) پاسخ می‌دهد: «من یک نهاد مالی هستم و تطابق بالایی با درخواست تو دارم.»

۳. کلمه "پول" (به عنوان کلید) پاسخ می‌دهد: «من شیء فیزیکی هستم و تطابق کمی دارم.»

[4] Scaled Dot-Product Attention
[5] Softmax
[6] Gradient
[7] Saturation
[8] Linear Combination

مدل با محاسبه شباهت برداری (ضرب داخلی) میان Q و Kها، وزن‌های توجه را محاسبه می‌کند. در نهایت، ترکیبی وزن‌دار از بردارهای مقدار ایجاد می‌شود که معنای جدید کلمه "آن" را در بافت این جمله خاص شکل می‌دهد. این عملیات در فرمول‌بندی مقاله «توجه ضرب داخلی مقیاس‌شده» نامیده شده است.

۵.۳) توجه چندسر: نگرش چندبعدی به زبان

آیا یک کلمه در جمله تنها یک نقش ایفا می‌کند؟ قطعاً خیر. کلمات همزمان دارای نقش‌های نحوی (مانند فاعل یا مفعول)، معنایی و ارجاعی هستند. یک مکانیزم توجه واحد ممکن است تنها بر روی یکی از این جنبه‌ها تمرکز کند. برای حل این محدودیت، ترنسفورمر از تکنیک توجه چند سر[۱] بهره می‌برد.

در این روش، مدل به جای یک بار، چندین بار (مثلاً $h = 8$ بار) و به صورت موازی فرآیند توجه را انجام می‌دهد. فرمول‌بندی ریاضی این لایه به صورت زیر است:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \tag{۴.۳}$$

که در آن هر سر به صورت مستقل در یک زیرفضای متفاوت محاسبه می‌گردد:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{۵.۳}$$

۱.۵.۳) تحلیل نقش ماتریس‌های وزن و تبدیل خطی

در این ساختار، دو گام اساسی از منظر جبر خطی رخ می‌دهد که درک آن‌ها برای تبیین مدل ضروری است:

۱. **چرا Q, K, V در ماتریس‌های وزن ضرب می‌شوند؟** بردارهای ورودی در ابتدا ثابت هستند. ضرب کردن آن‌ها در ماتریس‌های W_i^Q, W_i^K, W_i^V در واقع یک نگاشت خطی[۲] است. این کار باعث می‌شود که هر سرِ توجه[۳]، بردارها را به یک زیرفضای برداری[۴] جدید با ابعاد کمتر ($d_k = d_{model}/h$) ببرد. در این زیرفضاهای جدید، مدل آزاد است تا روابط متفاوتی را یاد بگیرد؛ برای مثال، یک سر یاد می‌گیرد که در فضای خود، روی فواصل کوتاه و روابط دستوری تمرکز کند و سر دیگر روی روابط معنایی دوربرد. بدون این ضرب ماتریسی، تمام سرها عملاً یک چیز واحد را تماشا می‌کردند.

۲. **نقش ماتریس نهایی W^O چیست؟** پس از آنکه هر سر خروجی خود را تولید کرد، ما با مجموعه‌ای از بردارها روبرو هستیم که هر کدام از دیدگاه خاصی به متن نگریسته‌اند. با عمل متصل کردن[۵]، این اطلاعات در کنار هم چیده می‌شوند. اما برای آنکه این اطلاعات پراکنده دوباره با هم تعامل کرده و به ابعاد اصلی مدل بازگردند، از ماتریس W^O استفاده می‌شود. این ماتریس وظیفه تلفیق[۶] دانش استخراج شده توسط

[1] Multi-Head Attention
[2] Linear Projection
[3] Attention Head
[4] Vector Subspace
[5] Concatenate
[6] Aggregation

تمام سرها را بر عهده دارد و وزن‌های هر سر را به گونه‌ای ترکیب می‌کند که خروجی نهایی، عصاره‌ای منسجم از تمام نگاه‌های تخصصی باشد.

در واقع، هر سر مسؤل استخراج نوع خاصی از روابط است:

- **سر اول:** روی روابط دستوری و نحوی تمرکز می‌کند.
- **سر دوم:** روابط زمانی افعال را بررسی می‌کند.
- **سر سوم:** مترادف‌ها و قرابت‌های معنایی را می‌یابد.

گویی چندین متخصص همزمان در حال تحلیل جمله از زوایای مختلف هستند و در نهایت ماتریس W^O مانند یک سردبیر، نظرات تمام آن‌ها را برای رسیدن به یک نتیجه واحد و جامع ترکیب می‌کند.

۶.۳) شبکه‌ی پیش‌خور و اتصالات میان‌بر: تثبیت یادگیری

پس از آنکه مکانیزم توجه، روابط پیچیده و بافتی میان کلمات را استخراج کرد، بردار خروجی به یک شبکه‌ی پیش‌خور نقطه‌ـ‌به‌ـ‌نقطه[1] ارسال می‌شود. این شبکه به صورت مستقل و کاملاً یکسان روی هر جایگاه (توکن) اعمال می‌گردد؛ بدین معنا که پارامترها برای تمام توکن‌های یک لایه مشترک است، اما از لایه‌ای به لایه دیگر تغییر می‌کند.

۱.۶.۳) تحلیل ساختاری و هندسی لایه پیش‌خور

از منظر ساختاری، این بخش شامل دو تبدیل خطی (نگاشت آفین) است که یک تابع فعال‌ساز غیرخطی تابع فعال‌ساز خطی اصلاح‌شده[2] میان آن‌ها قرار دارد. فرمول ریاضی این فرآیند به صورت زیر تدوین می‌گردد:

$$\text{FFN}(x) = \max(0, xW_1 + b_1)W_2 + b_2 \tag{۶.۳}$$

در این معادله، مؤلفه‌های ریاضی نقش‌های زیر را ایفا می‌کنند:

- **انبساط ابعادی[3]:** نگاشت اول ($xW_1 + b_1$) بردار ورودی را از فضای مدل ($d_{model} = 512$) به یک فضای میانی بسیار بزرگ‌تر ($d_{ff} = 2048$) منتقل می‌کند. این افزایش ابعاد به مدل اجازه می‌دهد تا ویژگی‌های استخراج شده توسط مکانیزم توجه را در یک فضای با ظرفیت بالاتر، بازنمایی و تفکیک کند.
- **غیرخطی‌سازی:** تابع $\max(0, xW_1 + b_1)$ با حذف مقادیر منفی، ویژگی‌های غیرخطی را به مدل تزریق می‌کند که برای یادگیری الگوهای پیچیده زبانی ضروری است.

[1] Position-wise Feed-Forward Network
[2] ReLU
[3] Dimensional Expansion

- **انقباض و بازگشت:** نگاشت دوم (W_2) وظیفه دارد داده‌ها را مجدداً به ابعاد اصلی مدل بازگرداند تا برای ورود به لایه‌های بعدی آماده شوند.

در حالی که لایه‌ی توجه وظیفه‌ی برقراری ارتباطات «افقی» میان کلمات مختلف را بر عهده داشت، لایه Feed-Forward Network (FFN) وظیفه دارد ویژگی‌های استخراج شده برای هر کلمه را به صورت «عمودی» و مجزا پردازش، پالایش و تقویت کند.

۲.۶.۳) اتصالات میان‌بر و نرمال‌سازی: تضمین پایداری ریاضی

یک چالش بنیادین در شبکه‌های عصبی عمیق، پدیده تأثیر محوشدگی گرادیان[1] است. در فرآیند پس‌انتشار، مشتقات متوالی ممکن است به سمت صفر میل کنند و مانع از بروزرسانی وزن‌های لایه‌های اولیه شوند. ترنسفورمر برای حل این بن‌بست ریاضی، از ساختار "جمع و نرمال‌سازی" بهره می‌برد:

$$\text{Output} = \text{LayerNorm}(x + \text{Sublayer}(x)) \tag{۷.۳}$$

این ساختار از دو بخش مکمل تشکیل شده است:

۱. **اتصال میان‌بر[2]:** با جمع کردن ورودی (x) با خروجی زیرلایه، یک بزرگراه اطلاعاتی ایجاد می‌شود. از نظر محاسباتی، این کار باعث می‌شود که در هنگام مشتق‌گیری، عبارت $\frac{\partial(x+f(x))}{\partial x} = 1 + f'(x)$ حاصل شود. وجود عدد یک تضمین می‌کند که حتی اگر مشتق لایه‌ها (f') بسیار کوچک باشد، گرادیان همچنان جریان یافته و به لایه‌های ابتدایی برسد.

۲. **نرمال‌سازی لایه[3]:** این عملیات با ثابت نگاه داشتن میانگین و واریانس فعالیت‌های عصبی در هر لایه، از نوسانات شدید داده‌ها جلوگیری می‌کند. این پایداری عددی، سرعت همگرایی مدل را به شدت افزایش داده و اجازه می‌دهد نرخ یادگیری[4] بالاتری در آموزش استفاده شود.

این ترکیب هوشمندانه، امکان آموزش مدل‌هایی با ده‌ها و صدها لایه را فراهم می‌سازد، بدون آنکه پایداری ریاضی مدل بر اثر افزایش عمق از دست برود.

۷.۳) تولید خروجی نهایی: لایه‌ی خطی و تابع نرم‌بیشینه

پس از آنکه داده‌ها از آخرین بلوک پشته‌ی رمزگشا عبور کردند، خروجی همچنان به صورت مجموعه‌ای از بردارهای پیوسته در فضای نمایش انتزاعی قرار دارد. برای آنکه این بردارها به کلمات قابل فهم و توزیع احتمال تبدیل شوند، مدل دو گام ریاضی زیر را طی می‌کند:

[1] Vanishing Gradient Effect
[2] Residual Connection
[3] Layer Normalization
[4] Learning Rate

۱. **لایه خطی**[۱]: این لایه یک شبکه‌ی پیش‌خور ساده است که وظیفه‌ی نگاشت[۲] بردار خروجی رمزگشا را به فضای دایره لغات بر عهده دارد. اگر تعداد کلمات موجود در دایره لغات مدل را V در نظر بگیریم، این لایه بردار خروجی را به برداری با طول V تبدیل می‌کند. هر مؤلفه در این بردار، نشان‌دهنده یک امتیاز خام[۳] برای هر کلمه در فرهنگ‌لغت است.

۲. **تابع سافت‌مکس:** امتیازات حاصل از لایه‌ی خطی، که در ادبیات یادگیری ماشین تحت عنوان امتیازات خام[۲] شناخته می‌شوند، اعدادی حقیقی در بازه‌ی $(-\infty, +\infty)$ هستند. این مقادیر به تنهایی فاقد تفسیر احتمالی مستقیم هستند، چرا که نه در بازه‌ی واحد قرار دارند و نه مجموع آن‌ها لزوماً برابر با یک است. برای تبدیل این بردار از امتیازات خام به یک توزیع احتمالاتی مجاز، از تابع نگاشت سافت‌مکس استفاده می‌شود.

این تابع، بردار $Z = (z_1, z_2, \ldots, z_V)$ را به بردار احتمالات $P = (p_1, p_2, \ldots, p_V)$ نگاشت می‌کند، به طوری که برای هر مؤلفه z_i داریم:

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{V} e^{z_j}} \tag{۸.۳}$$

بسط تحلیلی این فرآیند از چند منظر حائز اهمیت است:

- **تضمین اصل موضوع‌پذیری احتمال:** تابع سافت‌مکس به دلیل ماهیت تابع نمایی، تمام خروجی‌ها را مثبت گردانده و با تقسیم بر مجموع (برای نرمال‌سازی)، تضمین می‌کند که $\sum p_i = 1$ برقرار باشد. این امر خروجی مدل را به یک توزیع احتمال گسسته[۳] روی دایره لغات تبدیل می‌کند.
- **تقویت تفاوت‌ها:** از منظر ریاضی، سافت‌مکس یک نسخه‌ی نرم و مشتق‌پذیر از تابع بیشینه است. این تابع به گونه‌ای عمل می‌کند که بزرگترین امتیاز در ورودی (z_{max})، بیشترین سهم از احتمال را به خود اختصاص می‌دهد و سایر امتیازات به صورت نمایی تضعیف می‌شوند. این ویژگی برای فرآیند آموزش و پس‌انتشار حیاتی است، زیرا اجازه می‌دهد سیگنال خطا از میان این تابع عبور کند.
- **تفسیر هندسی:** این تابع در واقع بردار خروجی لایه‌ی خطی را بر روی یک سیمپلکس احتمالی[۴] در فضای Vـ بعدی تصویر می‌کند. در مرحله‌ی تولید متن، مدل با نمونه‌برداری از این توزیع یا انتخاب کلمه‌ای با بیشترین احتمال از طریق رمزگشایی حریصانه[۵]، توکن بعدی را مشخص می‌سازد.

این عملیات، گام نهایی در تبدیل محاسبات انتزاعی پشته‌ی رمزگشا به تصمیمات زبانی ملموس است که دقت آن مستقیماً بر کیفیت متن تولید شده تأثیر می‌گذارد.

---

[1] Linear Layer
[2] Projection
[3] Logit
[2] Logits
[3] Discrete Probability Distribution
[4] Probability Simplex
[5] Greedy Decoding

احتمالات خروجی[1]: در نهایت، کلمه‌ای که بالاترین مقدار احتمال را به خود اختصاص داده است، به عنوان پیش‌بینی مدل برای گام زمانی فعلی انتخاب می‌شود. این کلمه برای تولید توکن‌های بعدی، مجدداً به بخش ورودی‌های رمزگشا (خروجی‌های شیفت‌یافته به راست) بازگردانده می‌شود تا فرآیند به صورت خودرگرسیو[2] ادامه یابد.

۸.۳) تحلیل توپولوژیک و جریان داده در معماری ترنسفورمر

اکنون که مؤلفه‌های بنیادین ریاضیاتی نظیر مکانیزم توجه و شبکه‌های پیش‌خور را واکاوی کردیم، زمان آن است که نگاهی جامع به نحوه اتصال این اجزا و تشکیل گراف محاسباتی ترنسفورمر بیاندازیم. نمودار زیر (شکل ۱.۳) ساختار کلی مدل را نمایش می‌دهد که می‌توان آن را به سه بخش اصلی تقسیم کرد: ستون رمزگذار (چپ)، ستون رمزگشا (راست) و پل ارتباطی میان آن‌ها.

۱.۸.۳) مسیر جریان در پشته رمزگذار (سمت چپ)

فرآیند پردازش در ستون سمت چپ که معادل رمزگذار است، طی مراحل زیر انجام می‌پذیرد:

۱. **فضای برداری اولیه:** توکن‌های ورودی ابتدا از طریق لایه تعبیه ورودی[2] به بردارهای متراکم تبدیل شده و بلافاصله با سیگنال‌های رمزگذاری موقعیتی جمع می‌شوند تا اطلاعات مکانی به آن‌ها افزوده گردد.

۲. **بلوک‌های تکرارشونده:** داده‌ها وارد بلوکی می‌شوند که N بار تکرار می‌شود (در نسخه اصلی مقاله $N = 6$). در هر بلوک، جریان داده از دو زیرلایه اصلی عبور می‌کند:

- ابتدا لایه توجه چند سر روابط درونی متن را استخراج می‌کند.
- سپس لایه شبکه پیش‌خور[3] پردازش غیرخطی را انجام می‌دهد.

۳. **اتصالات میانبر و نرمال‌سازی:** نکته حیاتی در این معماری که در نمودار با فلش‌های کناری مشخص شده است، وجود اتصالات میانبر[4] است. خروجی هر زیرلایه با ورودی همان زیرلایه جمع شده و سپس وارد لایه نرمال‌سازی لایه (بلوک‌های زرد رنگ) می‌شود. این مکانیزم از نظر ریاضیاتی گرادیان را در طول شبکه حفظ کرده و همگرایی را تضمین می‌کند.

۲.۸.۳) مسیر جریان در پشته رمزگشا (سمت راست)

ستون سمت راست یا رمزگشا، ساختاری مشابه دارد اما با دو تفاوت ساختاری مهم که ناشی از ماهیت خودرگرسیو[5] (تولید کلمه به کلمه) است:

---

[1] Output Probabilities
[2] Auto-regressive
[2] Input Embedding
[3] Feed-Forward
[4] Residual Connections
[5] Auto-Regressive

۱. **پوشش اطلاعات آینده:** در اولین لایه توجه (بلوک پایین سمت راست)، از نوعی خاص از توجه به نام توجه چندسر پوشیده[1] استفاده شده است. این لایه تضمین می‌کند که در هنگام پیش‌بینی کلمه t، مدل هیچ دسترسی به کلمات $t+1$ به بعد نداشته باشد .

۲. **تولید توزیع احتمال:** در انتهای پشته رمزگشا، خروجی نهایی از یک لایه خطی عبور کرده تا به ابعاد دایره لغات پروژه شود و سپس تابع سافت‌مکس آن را به یک توزیع احتمالاتی معتبر تبدیل می‌کند که کلمه بعدی بر اساس آن انتخاب می‌شود.

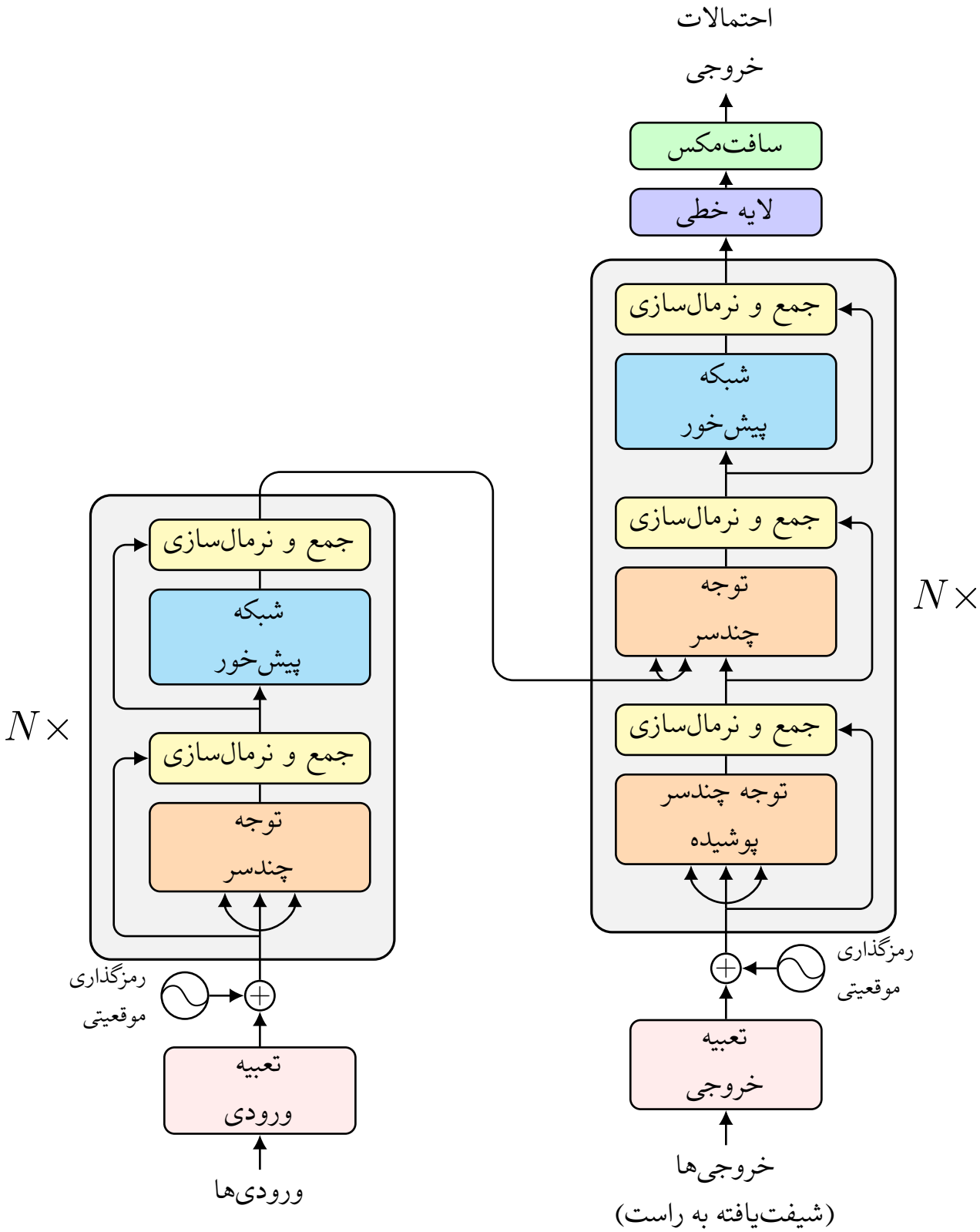
۳.۸.۳) مکانیزم توجه متقاطع: اتصال دو نیمکره

مهم‌ترین بخش نمودار، جایی است که دو ستون به هم متصل می‌شوند. اگر به بلوک میانی در سمت راست دقت کنید، متوجه می‌شوید که این لایه توجه (توجه چند سر) ورودی‌های خود را از دو منبع متفاوت دریافت می‌کند:

- **ورودی کوئری (Q):** از لایه پایین‌دستی در خودِ رمزگشا می‌آید (آنچه مدل قصد دارد بگوید).
- **ورودی کلید و مقدار (K, V):** از خروجی نهایی پشته رمزگذار (سمت چپ) می‌آید (آنچه متن اصلی بیان کرده است).

این اتصال همان نقطه‌ای است که مدل «زبان مقصد» را با «مفاهیم زبان مبدأ» هم‌تراز می‌کند.

[1] Masked Multi-Head Attention



شکل ۱.۳: نمای شماتیک معماری ترنسفورمر. ستون سمت چپ (رمزگذار) وظیفه استخراج ویژگی‌ها و ستون سمت راست (رمزگشا) وظیفه تولید دنباله خروجی را بر عهده دارد. بردارهای خروجی رمزگذار به عنوان ورودی‌های K و V به لایه‌های میانی رمزگشا تغذیه می‌شوند.

فصل ۴

چرا ترنسفورمر جهان را تغییر داد؟

اهمیت و جایگاه مقاله «توجه تمام آن چیزی است که نیاز دارید» در تاریخ علم داده، فراتر از معرفی صرف یک معماری جدید است؛ این پژوهش، دروازه‌ای را به سوی امکاناتی گشود که تا پیش از آن در زمره رویاهای محاسباتی دست‌نیافتنی طبقه‌بندی می‌شدند. این تغییر الگو، نتیجه همگرایی چندین عامل فنی و سخت‌افزاری بود که در ادامه به تشریح آن‌ها می‌پردازیم.

۱.۴) پیروزی موازی‌سازی و هم‌افزایی با قانون مور

همان‌گونه که در فصول پیشین تبیین شد، نقطه‌ضعف مدل‌های بازگشتی (RNN)، ماهیت سریالی آن‌ها بود. این وابستگی، گلوگاهی ساختاری ایجاد می‌کرد که با پیشرفت‌های سخت‌افزاری ناسازگار بود.

از منظر پیچیدگی محاسباتی[۱]، شبکه‌های بازگشتی دارای «پیچیدگی زمانی[۲]» خطی یا $O(N)$ هستند؛ بدین معنا که زمان لازم برای پردازش، رابطه‌ای مستقیم و غیرقابل شکست با طول جمله دارد. اما معماری ترنسفورمر با حذف این قید زمانی، اجازه داد تا محاسبات از حالت توالی خارج شده و به فرم موازی‌سازی ماتریسی[۳] درآیند.

این ویژگی انقلابی باعث شد پژوهشگران بتوانند از تمام ظرفیت کلاستر[۴] های عظیم GPU و نسل جدید سخت‌افزارها مانند «Tensor Processing Units (TPU)» بهره‌برداری کنند. این هم‌افزایی با قانون مور[۵] (که پیش‌بینی‌کننده پیشرفت سخت‌افزار است) باعث شد ناگهان آموزش مدل‌هایی با میلیاردها پارامتر بر روی کل داده‌های موجود در ویکی‌پدیا[۶] و اینترنت امکان‌پذیر شود.

اگر RNNها را به مانند پر کردن یک استخر با سطل آب در نظر بگیریم، ترنسفورمرها مانند باز کردن همزمان هزاران شیر فشار قوی عمل می‌کنند.

[1] Computational Complexity
[2] Time Complexity
[3] Matrix Parallelization
[4] Cluster
[5] Moore's Law
[6] Wikipedia

۲.۴) مدیریت وابستگی‌های بافاصله: حذف محدودیت‌های مکانی

در الگوهای کلاسیک پردازش زبان، فاصله فیزیکی میان کلمات، بزرگترین دشمن درک معنایی بود. هرچه فاصله بین فاعل و فعل یا ضمیر و مرجع آن بیشتر می‌شد، احتمال خطا و فراموشی مدل به صورت نمایی افزایش می‌یافت.

اما در معماری ترنسفورمر، مفهوم فاصله بازتعریف شده است. به مدد مکانیزم توجه، طول مسیر بیشینه[۱] یا مسافتی که سیگنال باید طی کند تا از کلمه اول به کلمه آخر برسد، به $O(1)$ تقلیل یافته است. این بدان معناست که دسترسی کلمه اول به کلمه صدم، همان‌قدر مستقیم، سریع و شفاف است که دسترسی به کلمه دوم.

این ویژگی باعث شده است که ترنسفورمرها در پردازش:

- مقالات علمی پیچیده با ارجاعات متعدد
- داستان‌های بلند با شخصیت‌های گوناگون
- و به‌ویژه کدهای برنامه‌نویسی که در آن‌ها تعریف یک متغیر در ابتدای فایل باید در صدها خط بعد شناسایی شود

عملکردی خیره‌کننده و بی‌رقیب از خود نشان دهند.

۳.۴) قوانین مقیاس‌پذیری: بزرگتر یعنی هوشمندتر

یکی از شگفت‌انگیزترین یافته‌های تجربی پس از ظهور ترنسفورمرها، کشف و فرمول‌بندی «قوانین مقیاس‌پذیری[۲]» بود. در مدل‌های عصبی قدیمی (مانند LSTM)، نمودار عملکرد مدل پس از رسیدن به حجم مشخصی از داده و اندازه شبکه، دچار اشباع عملکرد[۳] می‌شد؛ یعنی افزودن داده بیشتر یا بزرگ کردن مدل، تأثیر چندانی در بهبود نتایج نداشت.

اما محققان دریافتند که در معماری ترنسفورمر، این سقف شیشه‌ای شکسته شده است. عملکرد این مدل‌ها با افزایش:

- حجم داده‌های آموزشی
- تعداد پارامترهای مدل
- و توان محاسباتی

به صورت لگاریتمی و پیش‌بینی‌پذیری بهبود می‌یابد. این کشف مهم، آتش رقابت جهانی برای ساخت مدل‌های زبانی بزرگ[۴] یا همان Large Language Models (LLM)ها را شعله‌ور کرد و منجر به ظهور مدل‌های غول‌پیکری شد که امروزه مرزهای هوش ماشینی را جابجا کرده‌اند.

---

[1] Maximum Path Length
[2] Scaling Laws
[3] Performance Saturation
[4] Large Language Models

فصل ۵

عصر پسا ترنسفورمر و ظهور غول‌های زبانی

انتشار مقاله سال ۲۰۱۷، نه یک پایان، بلکه نقطه آغازی بر یک انفجار کامبرین[1] در اکوسیستم هوش مصنوعی بود. همان‌گونه که در زیست‌شناسی، دوره کامبرین شاهد ظهور سریع و متنوع گونه‌های حیاتی بود، در اینجا نیز معماری ترنسفورمر بستری را فراهم کرد که تمام مدل‌های زبانی پیشرفته امروزی، به عنوان نوادگان مستقیم یا غیرمستقیم آن طبقه‌بندی می‌شوند.

۱.۵) تبارشناسی مدل‌ها: سه شاخه تکاملی

بر مبنای معماری پایه ترنسفورمر، تحقیقات به سه شاخه اصلی تقسیم شدند که هر یک بر تقویت بخش خاصی از این ساختار تمرکز داشتند:

۱.۱.۵) معماری‌های فقط رمزگذار

این دسته از مدل‌ها تنها از پشته «رمزگذار» بهره می‌برند و تمرکز اصلی آن‌ها بر «درک معنایی» و تحلیل روابط درونی متن است.

- **کاربرد:** ایده‌آل برای وظایفی نظیر طبقه‌بندی متن[2]، تحلیل احساسات و استخراج موجودیت‌های نامدار.
- **نمونه شاخص:** مدل BERT (معرفی شده توسط گوگل در ۲۰۱۸). این مدل با ایجاد درک «دوطرفه[3]» از متن، انقلابی در موتورهای جستجو ایجاد کرد و توانست نیت کاربر را با دقتی بی‌سابقه شناسایی کند.

۲.۱.۵) معماری‌های فقط رمزگشا

این خانواده با حذف بخش رمزگذار، تمام توان خود را بر روی پشته «رمزگشا» و قابلیت‌های «تولید متن[4]» متمرکز کرده‌اند. وظیفه اصلی آن‌ها پیش‌بینی کلمه (یا توکن) بعدی در یک دنباله است.

[1] Cambrian Explosion
[2] Text Classification
[3] Bidirectional
[4] Text Generation

- **کاربرد:** تولید متن خلاقانه، تکمیل کد و مکالمه.
- **نمونه شاخص:** سری مدل‌های GPT (توسعه یافته توسط OpenAI). این مدل‌ها اثبات کردند که با افزایش مقیاس مدل و حجم داده‌ها، می‌توان به سطحی از تولید متن دست یافت که تمایز آن از نوشتار انسانی دشوار باشد.

۳.۱.۵) معماری‌های ترکیبی رمزگذار رمزگشا

این مدل‌ها به معماری اصلی و کامل مقاله ۲۰۱۷ وفادار مانده‌اند و هر دو بخش را حفظ کرده‌اند.

- **کاربرد:** وظایف دنباله به دنباله (Sequence to Sequence) مانند ترجمه ماشینی و خلاصه‌سازی متون.
- **نمونه شاخص:** مدل‌های تی‌فای[۱] (گوگل) و بارت[۲] (متا/فیس‌بوک).

۲.۵) جدول زمانی تکامل: رشد نمایی پارامترها

برای درک شتاب تحولات تکنولوژیک در این حوزه، بررسی روند تکامل مدل‌ها از سال ۲۰۱۷ تا کنون ضروری است. جدول زیر سیر صعودی ابعاد شبکه و قابلیت‌های نوظهور را نمایش می‌دهد.

جدول ۱.۵: جدول زمانی تکامل مدل‌های زبانی بزرگ و ویژگی‌های بارز آن‌ها

سال	مدل	سازنده	تعداد پارامترها	ویژگی بارز و دستاورد علمی
۲۰۱۷	ترنسفورمر	Google	۶۵ میلیون	معرفی معماری پایه و مکانیزم توجه
۲۰۱۸	GPT-۱	OpenAI	۱۱۷ میلیون	اثبات کارایی یادگیری نظارت‌نشده[۳] برای تولید متن
۲۰۱۸	BERT	Google	۳۴۰ میلیون	ایجاد درک عمیق و دوطرفه از بافت متن
۲۰۱۹	GPT-۲	OpenAI	۵.۱ میلیارد	تولید متون منسجم طولانی و طرح نگرانی‌های ایمنی
۲۰۲۰	GPT-۳	OpenAI	۱۷۵ میلیارد	ظهور یادگیری چند شات[۴]؛ عدم نیاز به بازآموزی برای وظایف جدید
۲۰۲۲	ChatGPT	OpenAI	محرمانه	استفاده از Reinforcement Learning from Human Feedback (RLHF) برای همسوسازی با قصد انسان و پذیرش عمومی
۲۰۲۳	GPT-۴	OpenAI	(تخمین: تریلیون)	توانایی‌های استدلال پیچیده و درک مدل‌های چندوجهی[۵]

[1] T5
[2] BART

۳.۵) فراتر از متن: تعمیم به بینایی و مدل‌های چندوجهی

قدرت انتزاعی و ریاضیاتی معماری ترنسفورمر به حدی فدرتمند بود که پژوهشگران دریافتند این ساختار محدود به پردازش زبان طبیعی نیست. ایده محوری این بود:

اگر بتوانیم تصاویر را نیز به مانند کلمات در نظر بگیریم، می‌توانیم از مکانیزم توجه بهره ببریم.

این نگرش منجر به تقسیم تصاویر به قطعات کوچک موزاییکی تحت عنوان تکه‌های تصویر[1] شد. این قطعات دقیقاً مانند کلمات در جمله وارد شبکه می‌شوند. نتیجه این رویکرد، ظهور ترنسفورمرهای بینایی[2] یا Vision Transformers (ViT) بود که امروزه در حوزه بینایی ماشین جایگزین بسیاری از مدل‌های کانولوشنی شده‌اند.

علاوه بر این، مدل‌های مولد تصویر مانند DALL-E و Midjourney که متن را به تصویر تبدیل می‌کنند، از همین زیرساخت برای ایجاد ارتباط معنایی میان فضای متنی و فضای بصری استفاده می‌کنند که فصل نوینی را در هوش مصنوعی چندوجهی رقم زده است.

[1] Image Patches
[2] Vision Transformers

نتیجه‌گیری: میراث ماندگار یک الگو نوین

مقاله جریان‌ساز «توجه تمام آن چیزی است که نیاز دارید» را می‌توان فراتر از یک دستاورد فنی، به عنوان نقطه عطف تاریخی و مرز جداکننده دوران هوش مصنوعی کلاسیک[1] و عصر نوین هوش مصنوعی مولد[2] قلمداد کرد. تا پیش از انتشار این اثر، الگو حاکم بر پردازش ماشینی، اسیر محدودیت‌های ذاتی زمان و خطی بودن بود؛ الگوریتم‌ها ناگزیر بودند جهان اطلاعات را به صورت پردازش متوالی[3]، کلمه به کلمه و لحظه به لحظه ادراک کنند. این رویکرد، منجر به خلق مدل‌هایی با حافظه کوتاه‌مدت و ناپایدار می‌شد که در مواجهه با دریای وسیع داده‌ها و ابعاد بالا، دچار سردرگمی و فراموشی می‌گشتند.

شکستن سد زمان و مکان

پژوهشگران گوگل با معرفی معماری ترنسفورمر و برجسته‌سازی نقش محوری مکانیزم توجه، این سد تکنولوژیک را درهم شکستند. آن‌ها اثبات کردند که اگر قیدهای زمانی را حذف کنیم و به ماشین اجازه دهیم نگاهی Global به کل داده‌ها داشته باشد، و همزمان به آن بیاموزیم که چگونه منابع پردازشی خود را مدیریت کند (یا همان تخصیص پویای منابع[4])، قادر به کشف الگوهای نهفته[5] خواهد بود که بسیار فراتر از درک سطحی زبان است.

گستره تأثیر: از ادبیات تا حیات

ترنسفورمرها نه تنها چالش دیرینه ترجمه ماشینی را با کیفیتی بی‌سابقه مرتفع ساختند، بلکه افق‌های جدیدی را در علوم کامپیوتر گشودند. امروزه این معماری‌ها زیربنای سیستم‌هایی هستند که:

- کدهای پیچیده نرم‌افزاری تولید می‌کنند،
- متون خلاقانه و ادبی می‌نگارند،
- در تشخیص‌های پزشکی دقیق عمل می‌کنند،

[1] Classical Artificial Intelligence
[2] Generative Artificial Intelligence
[3] Sequential Processing
[4] Dynamic Resource Allocation
[5] Latent Patterns

- و حتی در حوزه زیست‌شناسی محاسباتی[1]، ساختارهای پیچیده حیات مانند تاخوردگی پروتئین[2] را رمزگشایی می‌نمایند (اشاره به مدل‌هایی نظیر AlphaFold).

سخن پایانی

امروزه، هر تعاملی که با عامل‌های گفتگوگر[2] هوشمند صورت می‌گیرد، هر متنی که توسط مترجم‌های عصبی برگردان می‌شود، و هر شگفتی جدیدی که در اکوسیستم هوش مصنوعی رخ‌نمایی می‌کند، در حقیقت پژواکی از آن مقاله هشت صفحه‌ای سال ۲۰۱۷ است. مقاله‌ای که با جسارتی علمی و بیانیه‌ای قاطع اعلام کرد برای دستیابی به هوش عمومی، پیچیدگی‌های بازگشتی لازم نیست؛ بلکه **توجه**، به واقع تمام چیزی است که نیاز دارید.

[1] Computational Biology
[2] Protein Folding
[2] Conversational Agents

منابع

[1] Wikipedia Contributors. “Attention Is All You Need”. Available at: Wikipedia

[2] Wikipedia Contributors. “Cocktail party effect”. Available at: Wikipedia

[3] Wikipedia Contributors. “List of large language models”. Available at: Wikipedia

[4] DavletD. “Attention Is All You Need”. Available at: Medium

[5] Ujwal Tickoo. “Attention is all you need in plain English”. Available at: Medium

[6] “History and Evolution of LLMs”. Available at: GeeksforGeeks

[7] “From RNNs to Transformers”. Available at: Baeldung

[8] “Transformer vs RNN in NLP: A Comparative Analysis”. Available at: Appinventiv

[9] Ege Oguz. “RNN vs. Transformer: Why Parallelization Won the AI Race”. Available at: Medium

[10] “I Finally Understood Attention is All You Need After So Long”. Available at: AI Plain English

[11] Sebastian Raschka. “Understanding and Coding the Self-Attention Mechanism From Scratch”. Available at: Sebastian Raschka Blog

[12] Luv Bansal. “Transformer — Attention Is All You Need Easily Explained With Illustrations”. Available at: Medium

[13] Debasish Das. “Self-Attention: The Simple Mechanism That Made ChatGPT Possible”. Available at: Towards AI

[14] Gunjas Singh. “Positional encoding in transformers: a Visual and Intuitive guide”. Available at: Medium

[15] “A Gentle Introduction to Positional Encoding in Transformer Models”. Available at: MachineLearningMastery

[16] Animesh Basnet. “A Beginners Guide to Attention is all you need”. Available at: Medium

[17] “Deep Dive into Self-Attention by Hand”. Available at: Towards Data Science

[18] Mubashir Iqbal. “Self-Attention: Understanding with Easy Analogies”. Available at: Medium

[19] Micah Adler. “The Cocktail Party Inside a Neural Network”. Available at: Medium

[20] “Attention Is All You Need - A Deep Dive into the Revolutionary Transformer Architecture”. Available at: Towards AI

[21] “The history, timeline, and future of LLMs”. Available at: Toloka AI

[22] Justin Milner. “A Timeline Recap of LLMs (Part 1)”. Available at: Medium

[23] Dr. Alan D. Thompson. “Timeline of AI and language models”. Available at: LifeArchitect

[24] Amjad Shaikh. “Attention Is All You Need — And All We’ve Lost”. Available at: Medium

[25] Christian Versloot. “From vanilla RNNs to Transformers”. Available at: GitHub

[26] “Revolutionizing Vision: A Deep Dive into Attention Is All You Need”. Available at: ResearchGate

[27] “Enhancing English-Persian Neural Machine Translation”. Available at: EasyChair

[28] “What is an encoder-decoder model?”. Available at: IBM

[29] “How Transformers Work: A Detailed Exploration”. Available at: DataCamp

[30] “Why were/are LSTMs popular for so long for NLP?”. Available at: StackExchange

[31] “Why is it said transformers are more parallelizable than RNN’s?”. Available at: StackExchange

[32] “Are Transformers Strictly More Effective Than LSTM RNNs?”. Available at: Reddit

[33] “Why is it said that the transformer is more parallelizable than RNN’s?”. Available at: Reddit

[34] “How We Use Selective Attention to Filter Information and Focus”. Available at: Very-well Mind

[35] “The cocktail party explained”. Available at: Speech Neurolab

[36] “LLM Timeline”. Available at: LLM Timeline App

[37] “Attention is all you need (Transformer) - Model explanation”. Available at: YouTube

[38] “Transformer and Attention Mechanism”. Available at: YouTube

واژه‌نامه انگلیسی به فارسی

A

Aggregation ..... تلفیق
Auto-Regressive ..... خودرگرسیو
Auto-regressive ..... خودرگرسیو
Artificial Intelligence ..... هوش مصنوعی
Attention Head ..... سرِ توجه
Attention Is All You Need ..... توجه تمام آن چیزی است که نیاز دارید
Attention Mechanism ..... مکانیزم توجه

B

Background Noise ..... نویز پس‌زمینه
Backpropagation ..... پس‌انتشار
BART ..... بارت
Bidirectional ..... دوطرفه

C

Cambrian Explosion ..... انفجار کامبرین
Classical Artificial Intelligence ..... هوش مصنوعی کلاسیک
Cluster ..... کلاستر
Computational Biology ..... زیست‌شناسی محاسباتی
Computational Complexity ..... پیچیدگی محاسباتی
Computational Instability ..... ناپایداری محاسباتی
Concatenate ..... متصل کردن
Conversational Agents ..... عامل‌های گفتگوگر
Context Vector ..... بردار زمینه
Continuous Vector Space ..... فضای برداری پیوسته

D

Decoder ..... رمزگشا
Decoder Only ..... فقط رمزگشا
Deep Learning ..... یادگیری عمیق
Dynamic Resource Allocation ..... تخصیص پویای منابع
Dimension ..... بعد
Dimensional Expansion ..... انبساط ابعادی
Discrete Probability Distribution ..... توزیع احتمال گسسته

E

Euclidean Distance ..... فاصله اقلیدسی
Embedding ..... تعبیه
Encoder ..... رمزگذار
Encoder Decoder ..... رمزگذار رمزگشا
Encoder Only ..... فقط رمزگذار
Encoder-Decoder Architecture ..... معماری رمزگذار_رمزگشا

F

Feed-Forward ..... شبکه پیش‌خور
Few-Shot Learning ..... یادگیری چند شات

G

Gate ..... دروازه
Generative Artificial Intelligence ..... هوش مصنوعی مولد
Geometric Representation ..... نمایش هندسی
Global Access ..... دسترسی سراسری
Gradient ..... گرادیان
Greedy Decoding ..... رمزگشایی حریصانه

H

Hidden State ..... حالت پنهان

I

ID ..... شناسه
Image Patches ..... تکه‌های تصویر
Information Bottleneck ..... گلوگاه اطلاعاتی
Information Distribution ..... توزیع اطلاعات
Input Embedding ..... تعبیه ورودی

K

Key ..... کلید

L

Layer Normalization ..... نرمال‌سازی لایه
Large Language Models ..... مدل‌های زبانی بزرگ
Latent Patterns ..... الگوهای نهفته
Learning Rate ..... نرخ یادگیری
Linear Combination ..... ترکیب خطی
Linear Layer ..... لایه خطی
Linear Projection ..... نگاشت خطی
Linear Transformation ..... تبدیل خطی
Logit ..... امتیاز خام
Logits ..... امتیازات خام
Long-Range Dependencies ..... وابستگی‌های طولانی‌برد
Lookup Table ..... جدول جستجو

M

Maximum Path Length ..... طول مسیر بیشینه
Masked Multi-Head Attention ..... توجه چندسر پوشیده
Matrix Parallelization ..... موازی‌سازی ماتریسی
Multi-Head Attention ..... توجه چند سر
Multimodal Models ..... مدل‌های چندوجهی
Moore’s Law ..... قانون مور

N

Natural Language Processing (NLP) ..... پردازش زبان طبیعی

O

Output Probabilities ..... احتمالات خروجی

P

Parallel Processing ..... پردازش موازی
Performance Saturation ..... اشباع عملکرد
Positional Encoding ..... رمزگذاری موقعیتی
Position-wise Feed-Forward Network ..... شبکه‌ی پیش‌خور نقطه‌ـ‌به‌ـ‌نقطه
Probability Simplex ..... سیمپلکس احتمالی
Projection ..... نگاشت
Protein Folding ..... تاخوردگی پروتئین

Q

Query ..... پرس‌وجو

R

Recurrent Neural Network (RNN) ..... شبکه‌های عصبی بازگشتی
Relative Distances ..... فواصل نسبی
ReLU ..... تابع فعال‌ساز خطی اصلاح‌شده
Residual Connection ..... اتصال میان‌بر
Residual Connections ..... اتصالات میانبر

S

Saturation ..... اشباع شدن
Scaled Dot-Product Attention ..... توجه ضرب داخلی مقیاس‌شده
Scaling ..... هم‌مقیاس شدن
Scaling Laws ..... قوانین مقیاس‌پذیری
Self-Attention ..... توجه خودکار
Sequential Processing ..... پردازش متوالی
Sub-word ..... زیرـ واحد
Softmax ..... سافت‌مکس
Spatial Information ..... اطلاعات مکانی
Stack ..... پشته

T

T5 ..... تی‌فای
Text Classification ..... طبقه‌بندی متن
Text Generation ..... تولید متن
Temporal Dependency ..... وابستگی زمانی
The Cocktail Party Effect ..... اثر مهمانی کوکتل
Time Complexity ..... پیچیدگی زمانی
Token ..... توکن
Tokenization ..... توکن‌گذاری
Transformer ..... ترنسفورمر

U

Unsupervised Learning ..... یادگیری نظارت‌نشده

V

Value ..... مقدار
Vanishing Gradient ..... محو شدن گرادیان
Vanishing Gradient Effect ..... تأثیر محوشدگی گرادیان
Vector ..... بردار
Vector Subspace ..... زیرفضای برداری
Vision Transformers ..... ترنسفورمرهای بینایی
Vocabulary ..... فضای دایره لغات

W

Weight Allocation ..... تخصیص وزن
Wikipedia ..... ویکی‌پدیا
Word Embedding ..... تعبیه کلمات
Word Embeddings ..... تعبیه کلمات

واژه‌نامه فارسی به انگلیسی

ا

اتصال میان‌بر ..... Residual Connection
اتصالات میانبر ..... Residual Connections
اثر مهمانی کوکتل ..... The Cocktail Party Effect
احتمالات خروجی ..... Output Probabilities
اشباع شدن ..... Saturation
اشباع عملکرد ..... Performance Saturation
اطلاعات مکانی ..... Spatial Information
الگوهای نهفته ..... Latent Patterns
امتیاز خام ..... Logit
امتیازات خام ..... Logits
انبساط ابعادی ..... Dimensional Expansion
انفجار کامبرین ..... Cambrian Explosion

ب

بارت ..... BART
بردار ..... Vector
بردار زمینه ..... Context Vector
بعد ..... Dimension

پ

پردازش زبان طبیعی ..... Natural Language Processing (NLP)
پردازش متوالی ..... Sequential Processing
پردازش موازی ..... Parallel Processing
پرس‌وجو ..... Query
پس‌انتشار ..... Backpropagation
پشته ..... Stack

پیچیدگی زمانی ..... Time Complexity
پیچیدگی محاسباتی ..... Computational Complexity

ت

تأثیر محوشدگی گرادیان ..... Vanishing Gradient Effect
تابع فعال‌ساز خطی اصلاح‌شده ..... ReLU
تاخوردگی پروتئین ..... Protein Folding
تبدیل خطی ..... Linear Transformation
تخصیص پویای منابع ..... Dynamic Resource Allocation
تخصیص وزن ..... Weight Allocation
ترکیب خطی ..... Linear Combination
ترنسفورمر ..... Transformer
ترنسفورمرهای بینایی ..... Vision Transformers
تعبیه ..... Embedding
تعبیه کلمات ..... Word Embeddings
تعبیه ورودی ..... Input Embedding
تکه‌های تصویر ..... Image Patches
تلفیق ..... Aggregation
توجه تمام آن چیزی است که نیاز دارید ..... Attention Is All You Need
توجه چند سر ..... Multi-Head Attention
توجه چندسر پوشیده ..... Masked Multi-Head Attention
توجه خودکار ..... Self-Attention
توجه ضرب داخلی مقیاس‌شده ..... Scaled Dot-Product Attention
توزیع احتمال گسسته ..... Discrete Probability Distribution
توزیع اطلاعات ..... Information Distribution
توکن ..... Token
توکن‌گذاری ..... Tokenization
تولید متن ..... Text Generation
تی‌فای T5

ج

جدول جستجو ..... Lookup Table

ح

حالت پنهان ..... Hidden State

خ

خودرگرسیو Auto-regressive

د

دروازه Gate
دسترسی سراسری Global Access
دوطرفه Bidirectional

ر

رمزگذار Encoder
رمزگذار رمزگشا Encoder Decoder
رمزگذاری موقعیتی Positional Encoding
رمزگشا Decoder
رمزگشایی حریصانه Greedy Decoding

ز

زیرفضای برداری Vector Subspace
زیر_واحد Sub-word
زیست‌شناسی محاسباتی Computational Biology

س

سافت‌مکس Softmax
سرِ توجه Attention Head
سیمپلکس احتمالی Probability Simplex

ش

شبکه پیش‌خور Feed-Forward
شبکه‌های عصبی بازگشتی Recurrent Neural Network (RNN)
شبکه‌ی پیش‌خور نقطه_به_نقطه Position-wise Feed-Forward Network
شناسه ID

ط

طبقه‌بندی متن ..... Text Classification
طول مسیر بیشینه ..... Maximum Path Length

ع

عامل‌های گفتگوگر ..... Conversational Agents

ف

فاصله اقلیدسی ..... Euclidean Distance
فضای برداری پیوسته ..... Continuous Vector Space
فضای دایره لغات ..... Vocabulary
فقط رمزگذار ..... Encoder Only
فقط رمزگشا ..... Decoder Only
فواصل نسبی ..... Relative Distances

ق

قانون مور ..... Moore's Law
قوانین مقیاس‌پذیری ..... Scaling Laws

ک

کلاستر ..... Cluster
کلید ..... Key

گ

گرادیان ..... Gradient
گلوگاه اطلاعاتی ..... Information Bottleneck

ل

لایه خطی ..... Linear Layer

م

متصل کردن ..... Concatenate
محو شدن گرادیان ..... Vanishing Gradient
مدل‌های چندوجهی ..... Multimodal Models
مدل‌های زبانی بزرگ ..... Large Language Models
معماری رمزگذار_رمزگشا ..... Encoder-Decoder Architecture
مقدار ..... Value
مکانیزم توجه ..... Attention Mechanism
موازی‌سازی ماتریسی ..... Matrix Parallelization

ن

ناپایداری محاسباتی ..... Computational Instability
نرخ یادگیری ..... Learning Rate
نرمال‌سازی لایه ..... Layer Normalization
نگاشت ..... Projection
نگاشت خطی ..... Linear Projection
نمایش هندسی ..... Geometric Representation
نویز پس‌زمینه ..... Background Noise

و

وابستگی زمانی ..... Temporal Dependency
وابستگی‌های طولانی‌برد ..... Long-Range Dependencies
ویکی‌پدیا ..... Wikipedia

ه

هم‌مقیاس شدن ..... Scaling
هوش مصنوعی ..... Artificial Intelligence
هوش مصنوعی کلاسیک ..... Classical Artificial Intelligence
هوش مصنوعی مولد ..... Generative Artificial Intelligence

ی

یادگیری چند شات ..... Few-Shot Learning
یادگیری عمیق ..... Deep Learning
یادگیری نظارت‌نشده ..... Unsupervised Learning

فهرست اختصارات

B

BERT ..... Bidirectional Encoder Representations from Transformers

C

CNN ..... Convolutional Neural Networks

F

FFN ..... Feed-Forward Network

G

GPT ..... Generative Pre-trained Transformer
GPU ..... Graphics Processing Unit
GRU ..... Gated Recurrent Unit

L

LLM ..... Large Language Models
LSTM ..... Long Short-Term Memory

N

NLP ..... Natural Language Processing

R

RLHF ..... Reinforcement Learning from Human Feedback

RNN ..... Recurrent Neural Network

S

Seq2Seq ..... Sequence to Sequence

T

TPU ..... Tensor Processing Units

V

ViT ..... Vision Transformers