به نام خدا

عنوان:

سوال پنجم تکلیف چهارم شبکه‌های عصبی

استاد:

دکتر منصوری

دانشجو:

محمدعلی مجتهدسلیمانی

تاریخ:

۱۴۰۳/۱۰/۲۰

Table of Content

# بخش اول

## Self-attention چیست و چگونه به مدل کمک میکند که اطلاعات وابسته به کلمات مختلف در جمله را پردازش کند؟

### مکانیسم self-attention

مکانیسم **self-attention**، روشی است که به مدل امکان می‌دهد اهمیت کلمات مختلف در یک دنباله (مانند یک جمله) را هنگام پردازش یک کلمه خاص تعیین کند. این مکانیزم به مدل کمک می‌کند تا با در نظر گرفتن روابط بین تمام کلمات دنباله، چه نزدیک و چه دور، مفهوم هر کلمه را بهتر درک کند.

به عنوان مثال:

وقتی یک جمله را می‌خوانیم، کلمات را به‌صورت جداگانه تفسیر نمی‌کنیم، بلکه معنای هر کلمه را بر اساس ارتباطش با سایر کلمات اطرافش متوجه می‌شویم، **Self-Attention** به شبکه عصبی امکان انجام کاری مشابه را می‌دهد.

این مکانیزم روی روابط بین کلمات در یک دنباله تمرکز می‌کند. این مکانیزم امتیازهایی را محاسبه می‌کند که اهمیت رابطه بین هر جفت کلمه را نشان می‌دهد. امتیاز بالاتر نشان‌دهنده رابطه قوی‌تر و تأثیر بیشتر است. مدل بر اساس این امتیازها میتواند وزن هایی را بدست بیاورد که با توجه به ارتباطشان تعیین شده است.

این مکانیزم به مدل کمک می‌کند روابط بین کلماتی که فاصله زیادی از هم دارند را بفهمد، که برای درک ساختارهای پیچیده زبانی ضروری است. با در نظر گرفتن کل دنباله به‌صورت هم‌زمان، مدل می‌تواند درک جامع‌تری از معنای هر کلمه پیدا کند.

برخلاف برخی روش‌های قدیمی که کلمات را به‌صورت ترتیبی پردازش می‌کردند، Self-Attention امکان پردازش سریع‌تر را فراهم می‌کند، زیرا محاسبات برای تمام کلمات می‌تواند به‌طور هم‌زمان انجام شود.

## Queries, Keys and Values:

سه بردار در Transformer ها وجود دارد که به هر کلمه مرتبط هستند:

**Query (Q)**: نشان دهنده اطلاعاتی است که یک کلمه به دنبال آن است.

**Key (K)**: نشان‌دهنده "برچسب" یا "موضوع" یک کلمه است و مشخص می‌کند که چه اطلاعاتی ارائه می‌دهد.

**Value (V)**: نمایانگر محتوای واقعی یا اطلاعات مرتبط با یک کلمه است.

## محاسبه امتیازهای توجه (attention Scores)

برای هر کلمه، بردار Query آن با بردار Key تمامی کلمات دیگر (شامل خودش) مقایسه می‌شود. این مقایسه از طریق ضرب داخلی انجام می‌شود که شباهت بین Query و Key را اندازه‌گیری می‌کند. این فرایند اساساً پاسخ می‌دهد: ((چقدر این Key برای Query مرتبط است؟)).

امتیازهای حاصل از ضرب داخلی با تقسیم بر توان ۲ ابعاد بردارهای Key کوچک‌تر می‌شوند. این مقیاس‌گذاری به تثبیت فرایند آموزش کمک می‌کند و از بیش‌ازحد بزرگ شدن امتیازها جلوگیری می‌کند. تابع Softmax روی امتیازهای مقیاس‌شده اعمال

می‌شود. این تابع امتیازها را به توزیع احتمالات تبدیل می‌کند، به‌طوری‌که هر امتیاز نشان‌دهنده وزن یا اهمیت Value یک کلمه نسبت به Query فعلی است. این مقادیر، امتیازهای توجه هستند.

بعد از این مرحله جمع وزن دار اعمال میشود. امتیازهای توجه برای ایجاد یک جمع وزن دار از بردارهای Value همه کلمات استفاده می‌شوند. هر بردار Value در امتیاز توجه متناظر با خود ضرب می‌شود و این مقادیر وزن با هم جمع می‌شوند.

این وزن نه‌تنها معنای اصلی کلمه را در بر می‌گیرد، بلکه رابطه آن با تمامی کلمات دیگر جمله را نیز با توجه به میزان اهمیتشان منعکس می‌کند.

فرمول این مکانیسم به شکل زیر است:

$$Attention(Q, K, V) = softmax(QK^T / sqrt(d_k)) * V$$

## ویژگی multi-head attention

برای درک روابط پیچیده‌تر transformer ها از این ویژگی استفاده میکنند.

چندین مجموعه از Q,K و V وجود دارند برای هر کدام از headها. هر کدام از headها جنبه‌های مختلفی از دنباله ورودی را یاد می‌گیرد. برای مثال، یک head ممکن است روی روابط دستوری تمرکز کند، درحالی‌که head دیگر روی ارتباطات معنایی تمرکز دارد. با این ویژگی ما میتوانیم ویژگی های غنی تری استخراج کنیم و بدست بیاوریم.

## Positional encoding

ازآنجاکه Self-Attentionبه‌طور ذاتی ترتیب کلمات را در نظر نمی‌گیرد، transformer از **کدگذاری موقعیتی** برای افزودن اطلاعات مربوط به موقعیت هر کلمه در دنباله استفاده می‌کنند. کدگذاری‌های موقعیتی معمولاً با استفاده از توابع سینوسی و کسینوسی با فرکانس‌های مختلف تولید می‌شوند. این روش به مدل اجازه می‌دهد تا بین کلماتی که معنای یکسانی دارند اما موقعیت متفاوتی در جمله دارند، تمایز قائل شود.

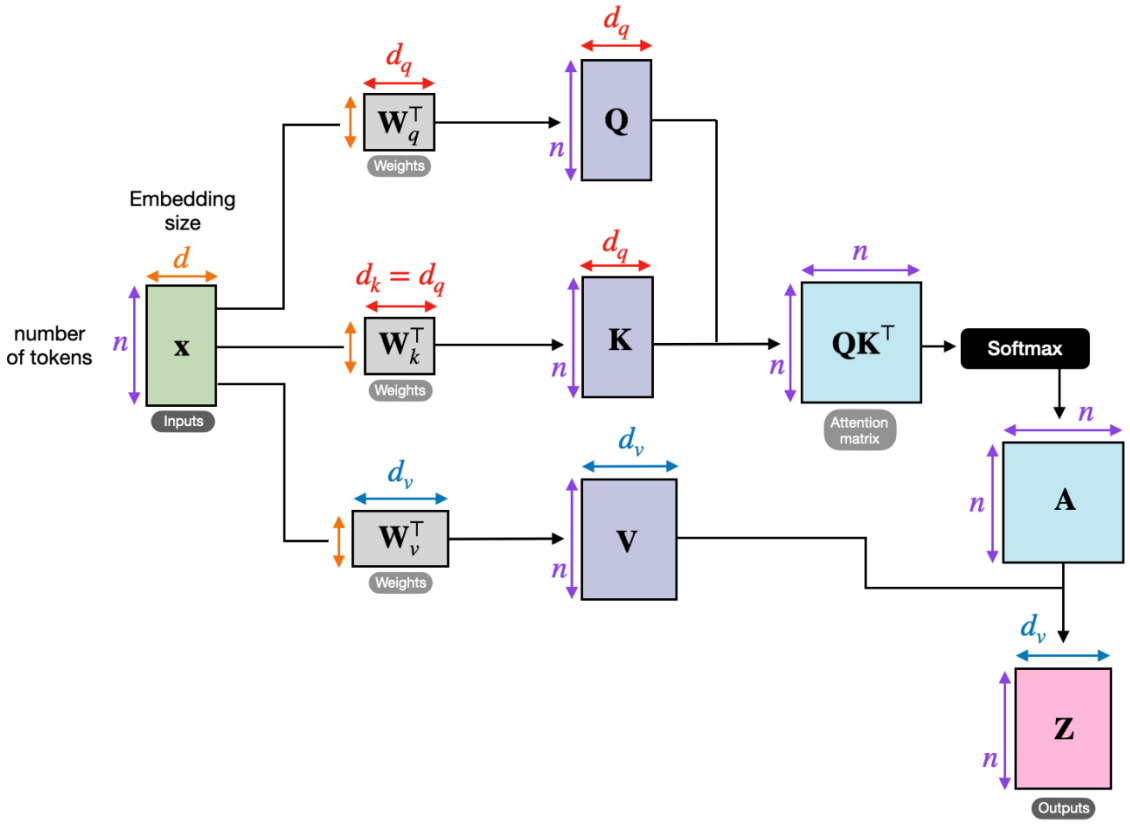
## مزایای این مکانیسم

بدست آوردن وابستگی های طولانی مدت: Self-Attentionبه‌مراتب بهتر از RNNها قادر به درک روابط بین کلمات دور از هم است، چراکهRNN ها معمولاً با مشکل محوشدگی گرادیان در دنباله‌های بلند مواجه می‌شوند. موازی‌سازی: محاسبات Self-Attention می‌توانند به‌صورت موازی برای تمامی کلمات در یک جمله انجام شوند، که این امر به‌طور قابل‌توجهی سرعت آموزش را نسبت به پردازش ترتیبیRNN ها افزایش می‌دهد. قابلیت تفسیر: امتیازهای توجه، سطحی از تفسیرپذیری را فراهم می‌کنند و به ما نشان می‌دهند که مدل هنگام پیش‌بینی بر روی کدام کلمات تمرکز کرده است.

## چگونه به مدل کمک میکند؟

این مکانیزم با استفاده از وزن هایی و امتیاز توجهی که بدست میاورد کمک میکند که مدل بتواند معنای کلمات متناسب با زمینه ای که آن کلمه در آن بیان شده است یاد گرفته شود و به آن توجه میشود. این مکانیزم تعیین می‌کند که هر کلمه تا چه حد باید به سایر کلمات در جمله "توجه" کند. این ترکیب وزنی به طور موثری روابط بین کلمه هدف و سایر کلمات را کدگذاری می‌کند. یکی دیگر از مزایای این مکانیزم توانایی آن در درک روابط

بین کلماتی است که در جمله فاصله زیادی از یکدیگر دارند. در مدل‌های سنتی مانند RNNها، اطلاعات کلمات دور ممکن است در طول مراحل مختلف پردازش کم‌رنگ شوند. برخلاف مدل‌های ترتیبی، ارتباطات مستقیم بین تمام کلمات برقرار می‌کند و به مدل اجازه می‌دهد روابط را بدون نیاز به پردازش گام‌به‌گام ارزیابی کند. امتیازات به عنوان وزن‌های پویا عمل می‌کنند و به مدل اجازه می‌دهند بر مرتبط‌ترین کلمات برای پردازش هر بخش از جمله تمرکز کند.



# بخش دوم

چرا مکانیسم **Attention** برای داده های دنباله‌ای موثرتر از مکانیزم های بازگشتی است؟

## مدیریت وابستگی طولانی

مکانیسم‌های بازگشتی :

**RNN**ها داده‌های ترتیبی را به صورت گام به گام پردازش می‌کنند و اطلاعات مراحل قبلی را از طریق حالت مخفی ***(hidden state)*** منتقل می‌کنند. با این حال، با طولانی شدن دنباله، اطلاعات مراحل ابتدایی ممکن است به دلیل مشکل ***vanishing gradient***(محو شدگی گرادیان) در طول آموزش از بین بروند. این امر یادگیری روابط بین کلمات دور از هم در دنباله را برای **RNN** ها دشوار می‌کند.

مکانیسم‌توجه:

در مقابل، مکانیسم توجه اتصالات مستقیمی بین تمام کلمات یک دنباله ایجاد می‌کنند، بدون توجه به فاصله آن‌ها. این امکان را به مدل می‌دهد تا روابط بین هر دو کلمه را، بدون توجه به فاصله آن‌ها، به صورت مستقیم ارزیابی کند. این دسترسی مستقیم به اطلاعات از تمام بخش‌های دنباله، به مدل اجازه می‌دهد وابستگی‌های طولانی را بسیار مؤثرتر درک کند.

## موازی سازی

مکانیسم‌های بازگشتی:

**RNN**ها ذاتاً ترتیبی هستند. آن‌ها باید یک کلمه را در یک زمان پردازش کنند و برای پردازش مرحله بعدی، منتظر خروجی مرحله قبلی بمانند. این ماهیت ترتیبی باعث کندی آموزش آن‌ها می‌شود، به ویژه در دنباله‌های طولانی.

مکانیسم توجه:

مکانیسم توجه امکان محاسبات موازی را فراهم می‌کنند. روابط بین تمام کلمات یک دنباله می‌توانند به صورت هم‌زمان محاسبه شوند. این موازی‌سازی سرعت آموزش را به طور قابل توجهی افزایش می‌دهد و مدل‌های مبتنی بر توجه را بسیار مؤثرتر می‌کند.

## مشکل کاهش و افزایش شدید گرادیان:

مکانیسم‌های بازگشتی:

RNNها مستعد مشکلات کاهش (*vanishing*) و افزایش شدید (*exploding*) گرادیان در طول آموزش هستند. این مشکلات زمانی رخ می‌دهند که گرادیان‌ها از میان مراحل زمانی متعددی بازگشت داده شوند، که می‌تواند باعث شود آن‌ها یا بسیار کوچک (کاهش یابند) یا بسیار بزرگ (افزایش یابند). این امر آموزش مؤثر RNN ها را، به خصوص در دنباله‌های طولانی، چالش‌برانگیز می‌کند. هرچند LSTM ها و GRU ها تا حدی این مشکل را کاهش می‌دهند، اما به طور کامل آن را برطرف نمی‌کنند.

مکانیسم توجه:

مکانیسم توجه کمتر در معرض این مشکلات قرار دارند، زیرا به پردازش ترتیبی و بازگشت گرادیان از میان مراحل زمانی متعددی متکی نیستند. اتصالات مستقیم بین کلمات باعث جریان پایدارتر گرادیان در طول آموزش می‌شود.

## تفسیرپذیری:

مکانیسم‌های بازگشتی:

فهم اینکه چرا یک RNN پیش‌بینی خاصی انجام می‌دهد، دشوار است زیرا اطلاعات در یک حالت مخفی پیچیده که به مرور زمان تکامل یافته است، کدگذاری می‌شود.

مکانیسم توجه:

مکانیسم توجه درجه‌ای از تفسیرپذیری را ارائه می‌دهند. امتیازات توجه که قدرت رابطه بین کلمات را نشان می‌دهند، بینشی درباره بخش‌هایی از دنباله ورودی که مدل هنگام انجام پیش‌بینی روی آن‌ها تمرکز کرده است، فراهم می‌کنند. این موضوع می‌تواند در درک فرآیند تصمیم‌گیری مدل و رفع خطاها مفید باشد.

## درک زمینه‌ای

مکانیسم‌های بازگشتی:

هرچندRNN ها کلمات قبلی را در نظر می‌گیرند، اما "میدان دید" محدودی در هر گام دارند. عمدتاً کلمات اخیر تأثیر بیشتری بر گام فعلی دارند.

مکانیسم توجه:

با ایجاد ترکیب وزنی از تمام کلمات دنباله، مکانیسم توجه به مدل اجازه می‌دهند درک جامع‌تر و دقیق‌تری از بافت پیرامون هر کلمه ایجاد کند. مدل می‌تواند برای پیش‌بینی هر نقطه خاص، به صورت پویا بر مرتبط‌ترین کلمات تمرکز کند، صرف نظر از موقعیت آن‌ها در دنباله.