به نام خدا

عنوان:

بخش چهارم تکلیف چهارم شبکه‌های عصبی

استاد:

دکتر منصوری

دانشجو:

محمدعلی مجتهدسلیمانی - 4033904504

تاریخ:

18/10/1403

Table of Content

[**بخش اول** 3](#_Toc187272684)

[**پس به طور خلاصه** 6](#_Toc187272685)

[**جدولی از مقایسه بین RNN و Transformer:** 7](#_Toc187272686)

[**بخش دوم** 7](#_Toc187272687)

# **بخش اول**

چرا Transformer نسبت به RNN در مدل سازی دنباله‌ها بهتر عمل میکند؟

قبل از اینکه به سراغ چرایی بهتر بودن transformer ها برویم بهتر است یک نگاه سطحی به عملکرد هر کدام از این 2 شبکه عصبی داشته باشیم. RNN ها طراحی شده اند تا دنباله ها را به وسیله نگهداری یک hidden state پردازش کنند. این hidden state در هر گام از دنباله بروز رسانی میشود. این hidden state به عنوان یک حافظه عمل میکند و اطلاعات را از مراحل قبلی نگه داری میکنند تا در پردازش گام فعلی استفاده بکنند. در طرف دیگر Transformer ها با استفاده از مکانیسم self-attention دنباله ها را تغییر میدهند، با این مکانیسم آنها کل دنباله را میتوانند به صورت یکجا و موازی پردازش کنند و نیازی به پردازش ترتیبی ندارند. از اونجایی که در این مکانیسم ترتیبی برای دنباله لحاظ نمیشود از positional encoding استفاده میشود تا اطلاعات مربوط به position هر المان دنباله را نگهداری کند تا ترتیب دنباله ها بهم نخورد.

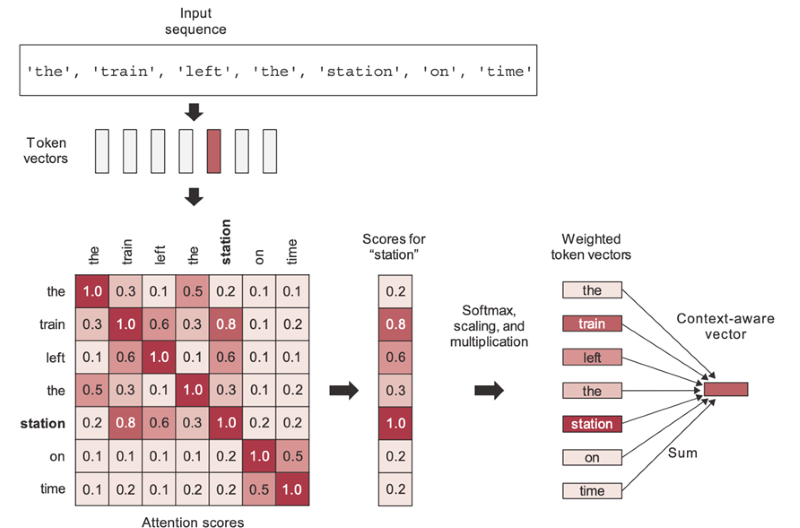
همانطور که در توضیحات قبلی اشاره کردیم، اصلی ترین مشکل RNN عدم قابلیت موازی بودن است. RNN ها در هر واحد زمان یک المان از دنباله را پردازش میکنند. برای هر المان در هر گام یک hidden state نگهداری و بروز رسانی میشود. این hidden state یک برداری است که اطلاعات گام های قبلی را در خودش نگه میدارد همانطور که در توضیحات بالاتر اشاره کردم. محاسبات hidden state در گام فعلی فقط به تمام hidden state های قبلی وابسته است. این مسئله یک زنجیره ای از وابستگی ها را بین hidden state ها و المان‌های دنباله ها میسازد. همین مسئله باعث میشود زمان آموزش این شبکه ها بسیار زیاد شود به سبب اینکه یک گلوگاه (bottleneck) در گام فعلی ایجاد میشود که به صورت ترتیبی نیاز هست تمام hidden state ها پردازش شوند.

در مقابل Transformer ها به علت استفاده از مکانیسم self-attention امکان موازی سازی دارند زیرا میتوانند به تمام المان های یک دنباله به صورت همزمان دسترسی داشته باشند. به همین دلیل دیگر hidden stateیی وجود نخواهد داشت که بخواهد به گام های بعدی منتقل شود. هر المان به صورت مستقیم میتواند با هر المان دیگر بررسی شود که باعث میشود این زنجیری از وابستگی که در RNN شکل میگرفت از بین برود. این مکانیسم باعث میشود زمان آموزش مدل ها سریعتر شود به همین دلیل ما میتوانیم مدل های بزرگتری بر روی dataset های بزرگ داشته باشیم.

دومین موضوعی که باید به آن اشاره کرد از تفاوت این 2 شبکه عصبی، توانایی آنها در کار با وابستگی های طولانی یا بلند به اصطلاح (long-range dependencies) گفته میشود به این معنی کدام شبکه عصبی میتواند گذشته های دورتری را به یاد بیاورد. مشکلی که در RNN وجود دارد این است که به علت محو شدن و انفجار گرادیان (Vanishing and Exploding Gradients) دچار مشکل فراموش میشوند در نتیجه این اتفاق شبکه توانایی در بدست آوردن ارتباطات در قسمت های دورتر دنباله را از دست میدهد و به نوعی انگار دچار فراموشی میشود. به عنوان مثال، فهم ضمیر “it” در انگلیسی نیازمند این است که ما بدانیم دقیقا در اوایل جمله کدام اسم (noun) ذکر شده است که در حالا در گام های بعدی به آن اشاره شده است. در حین آموزش RNN از الگوریتم پس انتشار (back propagation) استفاده میکند تا وزن های را تنظیم کند. وقتی دنباله بزرگ بشود این مقدار گرادیان میتواند بسیار کوچک یا بسیار بزرگ شود. اگر دچار محو شدگی گرادیان شویم باعث میشود گام های ابتدایی دنباله تاثیر بسیار ناچیز و کمی در تنظیم وزن ها داشته باشند، به این معنی که RNN دچار فراموشی نسبت به گام های ابتدایی میشود و آن اطلاعات از بین میروند. اگر گرادیان بسیار بزرگ شود باعث ناپایداری در حین فرآیند یادگیری میشود. (اگر چه که شبکه هایی مانند GRU و LSTM برای حل این مشکلات آمده اند.)

در مقابل Transformer میتوانند دسترسی مستقیم در هر گامی از دنباله داشته باشند. مکانیسم self-attention باعث میشود که بین هر المان یک دنباله ما یک رابطه مستقیم داشته باشیم و اطلاعات مربوط بین این 2 محاسبه شود. وزن ها اینجا میزان اهمیت هر المان دنباله نسبت به تمام المان های دیگر و خودش محاسبه خواهد شد. به سبب اینکه دیگر پردازش ترتیبی نداریم دیگر خبری از hidden state نداریم. این ویژگی باعث میشود که وابستگی های طولانی بسیار خوب بتوانند محاسبه بشوند و شبکه درک خیلی خوبی از مضمون کلی دنباله داشته باشد. حتی میتوانند ارتباطات پیچیده درون داده ها بسیار خوب متوجه بشوند.

همین امر سبب میشود که در محاسبات این 2 شبکه تفاوت اساسی ایجاد شود. در RNN ها تعداد محاسبات به صورت خطی با طول دنباله رشد پیدا میکند. یعنی اگر طول دنباله را 2 برابر کنیم تعداد محاسبات هم 2 برابر میشود. در مقابل Transformer ها تعداد محاسباتی ثابتی دارند نسبت به طول دنباله که دریافت میکنند. اگر چه که این شبکه ها نیاز به حافظه بیشتری دارند زیرا نیاز هست که وزن های attention ها (امتیاز که نشان میدهد هر المان چه قدر به سایر المان ها توجه دارد) ذخیره شوند. این وزن ها برای هر جفت از المان ها محاسبه میشوند.



این عکس به خوبی محاسبه attention را نشان میدهد که خروجی حاصل یک ماتریس خواهد بود که ابعاد آن برابر با ابعاد خود دنباله است. اعداد بدست آمده همان وزن های attention هستند که نشان میدهد هر المان چه قدر نسبت به خودش و سایر المان ها توجه کند و اهمیت دارد. به عنوان مثال کلمه train با کلمه station وزن بیشتری دارند به این معنا که این 2 ارتباط بیشتری با همدیگر دارند.

به سبب دلایل گفته شده RNN ناتوانی قابل توجهی نسبت به dataset های بزرگ و دنباله های بزرگ دارند و آموزش خیلی کند اتفاق میفتد.

در مقابل شبکه های Transformer به دلیل مکانیسم توجه امکان موازی سازی دارند و بسیار مقیاس پذیر خواهند بود نسبت به dataset های بزرگ و دنباله های بزرگ. تحقیقات نشان داده است که هر چه قدر مدل های این شبکه بزرگتر باشند کارایی بیشتری خواهند داشت.

یک نگاه دیگر به این 2 شبکه میتواند اینگونه گفته شود که RNN یک نگاه محلی به دنباله ها دارند یعنی با یک چشم انداز محلی به دنباله ها نگاه میکنند. پردازش هر المان به صورت ترتیبی خواهد بود و مفهوم جمله را به صورت تدریجی درک میکنند. اما شبکه های Transformer با یک چشم انداز سراسری به دنباله ها نگاه میکنند به سبب استفاده از مکانیسم self-attention. هر المان به اطلاعات سایر المان ها دسترسی دارد و میتوانند ارتباط های پیچیده بین المان ها را بهتر متوجه شوند.

## **پس به طور خلاصه:**

* شبکه های Transformer به دلیل استفاده از مکانیسم self-attention اجازه میدهند هر المان با سایر المان های دنباله ارتباط برقرار کند و فرقی نمیکند فاصله این 2 المان چه قدر باشد.
* با مشکلاتی از قبیل محو شدن گرادیان روبرو نمیشوند و اطلاعاتی را فراموش نمیکنند در ارتباطات طولانی.
* قابلیت پردازش موازی دارند برعکس پردازش ترتیبی RNNها.

## **جدولی از مقایسه بین RNN و Transformer:**

|  |  |  |
| --- | --- | --- |
| ویژگی | RNN | Transformer |
| پردازش | ترتیبی (هر المان در واحد زمان) | موازی (همه المان ها به صورت همزمان) |
| وابستگی طولانی | ضعیف (محو شدن گرادیان) | عالی (مکانیسم توجه) |
| سرعت یادگیری | آهسته | سریع |
| موازی سازی | محدود | بسیار بالا |
| مقیاس پذیری | محدود | بسیار بالا |
| حافظه | پایین | بالا (وزن attention) |
| چهارچوب | نگاه محلی | نگاه سراسری |

# **بخش دوم**

سسی