



Assignment NO.11 Solutions

Deep learning | spring 1401 | Dr.Mohammadi

Teacher Assistant:

Shabnam Ezatzadeh

Student name : **Amin Fathi**

Student id : **400722102**

Problem1

مزایای مدل های end to end :

این مدل های طبق متریک های دقت و یادآوری عملکرد بهتری داشته ایند، علاوه بر این، این مدل ها از مشکل تعیین اجزای مورد نیاز برای انجام یک کار و نحوه تعامل آن اجتناب می کنند و در واقع سادگی خوبی دارند و همین سادگی تلاش کمتری برای پیاده سازی نیاز دارد . این مدل ها همچنین بهینه سازی بهتری نسبت به رقبا خود داشته و نیاز به متخصصان نیز کمتر است در آن ها .

معایب مدل های end to end :

از معایب این مدل ها می توان گفت lack of explainability و همچنین ضعف در پیش بینی عملکرد این شبکه هاست ، به سبب ساختار این مدل ها نمیتوان با قاطعیت گفت دقیقا چه اتفاقی رخ میدهد و نتیجه چه خواهد شد . بنابراین یافتن نقطه ای از مدل که عملکرد اشتباهی دارد و نیازمند تغییر است کار سختی است . هزینه محاسباتی و حافظه بالا و نیاز شدید به داده از دیگر معایب این مدل ها می باشد .

منبع :

[End-to-End Models for Complex AI Tasks | Capital One](#)

Problem 2

یکی از راه ها محدود کردن میدان دید (field of view) به یک مقدار ثابت و از پیش تعیین شده است؛ به طور مثال با استفاده از پنجره های محلی و بلوک دیاگرام ها با مقدار گام (stride) ثابت و یا می توان با استفاده از تکنیک هایی نظیر pooling و کانولوشن اقدام به کم کردن مقدار طول دنباله کرد. یکی دیگر از روش های قابل ذکر استفاده از یک ماژول حافظه جانبی قابل یادگیری است که می تواند همزمان به چندین توکن دسترسی داشته باشد . نگاشت طول ابعاد کلید ها و مقادیر به ابعاد کوچک تر و در واقع انتقال این مقادیر به طول بعد کمتر می تواند مشکل پیچیدگی حافظه مد نظر را به شدت کاهش دهد. یکی دیگر از کار های ممکن کاهش رزولوشن دنباله (sequence) هاست . هم چنین استفاده از کرنل ها در محاسبات ریاضی بسیار به صرفه تر از محاسبات مربوط به ماتریس های $N*N$ در مدل مبتنی بر مبدل است .

منبع :

[\[2009.06732\] Efficient Transformers: A Survey \(arxiv.org\)](#)

Problem3.a

زمانی مفید هستند که ما در واقع چند نوع token داشته باشیم مانند QA که در واقع قسمت context را با ۰ و قسمت question را با ۱ نشان می دهیم. به طور واضح تر برای حالتی که نیاز به ادغام چند نوع token type داشته باشیم و بخواهیم آن را تفکیک کنیم از token types id استفاده می کنیم.

منبع :

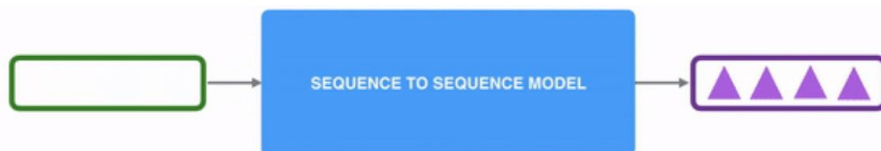
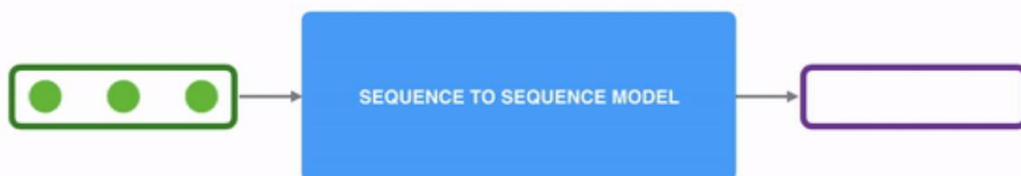
[Transformers In NLP | State-Of-The-Art-Models \(analyticsvidhya.com\)](https://huggingface.co/transformers/v3.2.0/glossary.html)

<https://huggingface.co/transformers/v3.2.0/glossary.html>

<https://jaketae.github.io/category/common-sense/#:~:text=The%20token%20type%20IDs%20are,require%20this%20two-sentence%20distinction>

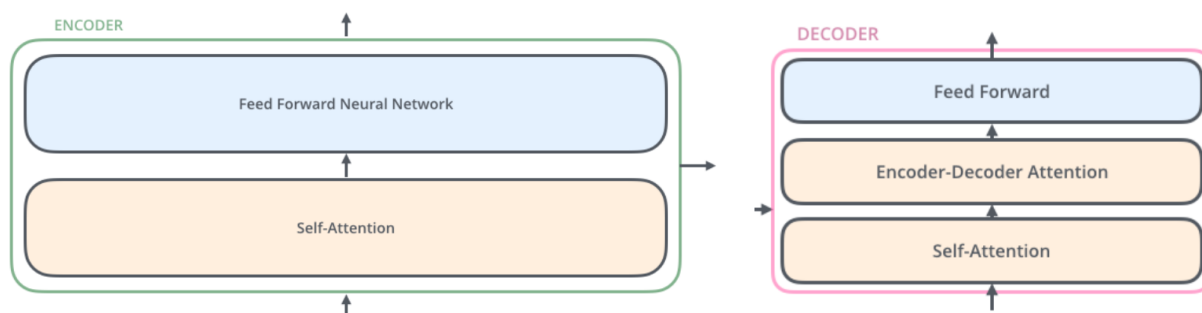
Problem3.b

مبدل ترکیب یک رمزگشا و رمزگذار است و استفاده از ایده مکانیسم توجه است. و ایده اصلی مبدل تبدیل یک دنباله ورودی به یک دنباله خروجی است و بیشتر در مباحث تشخیص گفتار و تبدیل text to speech کارآرایی دارد



ایده اصلی این شبکه ها ترکیب شبکه های کانولوشنی و مکانیسم توجه (برای سرعت بخشیدن به تبدیل دنباله ورودی به خروجی است)

شمایی از یک مبدل را در زیر می بینید ، ۶ شبکه رمزگشا و ۶ شبکه رمزگذار که رمزگشا ها معماری مشابهی با هم دارند و رمزگذار ها هم معماری مشابهی با یکدیگر ، معماری این دو را در زیر مشاهده می کنید.



Self attention در قسمت encoder باعث میشود تا شبکه به بقیه کلمات جمله نیز توجه کند ، بخش encoder-decoder attention نیز باعث می شود تا decoder بر روی قسمت های مرتبط جمله ورودی تمرکز کند .

منبع :

[How Transformers Work. Transformers are a type of neural... | by Giuliano Giacaglia | Towards Data Science](#)

Problem 3.c

مدل های مبدل به دلیل سربار محاسباتی و حافظه به خصوص در قسمت self attention معماری آن ها که در قسمت قبل توضیح داده شد؛ در طول دنباله خود ثابت هستند و بنابراین توانایی پردازش وظایفی که نیاز به طول دنباله و توالی بلندی هستند ندارد ، وضایفی مانند خلاصه سازی اسناد.

منبع :

[Google AI Blog: Constructing Transformers For Longer Sequences with Sparse Attention Methods \(googleblog.com\)](#)

Problem4

هر سه حالت ممکن است جوابگو باشد و بستگی به کاربرد دارد .

رمزگذار در مواقعی که مثلا توکن های قبلی نیز توسط دریافت کننده دریافت می شود و هر نشانه با شرطی شدن نشانه های قبلی پیش بینی میشود نیازی ندارد در معماری استفاده شود . برخی مواقع مانند LM های ماسک دار، نیازی به رمزگشا نیست یا در مدل های NMT هر توکن ترجمه با شرطی شدن نشانه های قبلی و جمله ورودی پیش بینی میشود که در این حالت وظیفه پردازش توکن های قبلی بر عهده رمزگشا و متن اصلی بر عهده رمزگذار است.

منابع:

<https://datascience.stackexchange.com/questions/65241/why-is-the-decoder-not-a-part-of-bert-architecture>

<https://www.kaggle.com/general/236840>