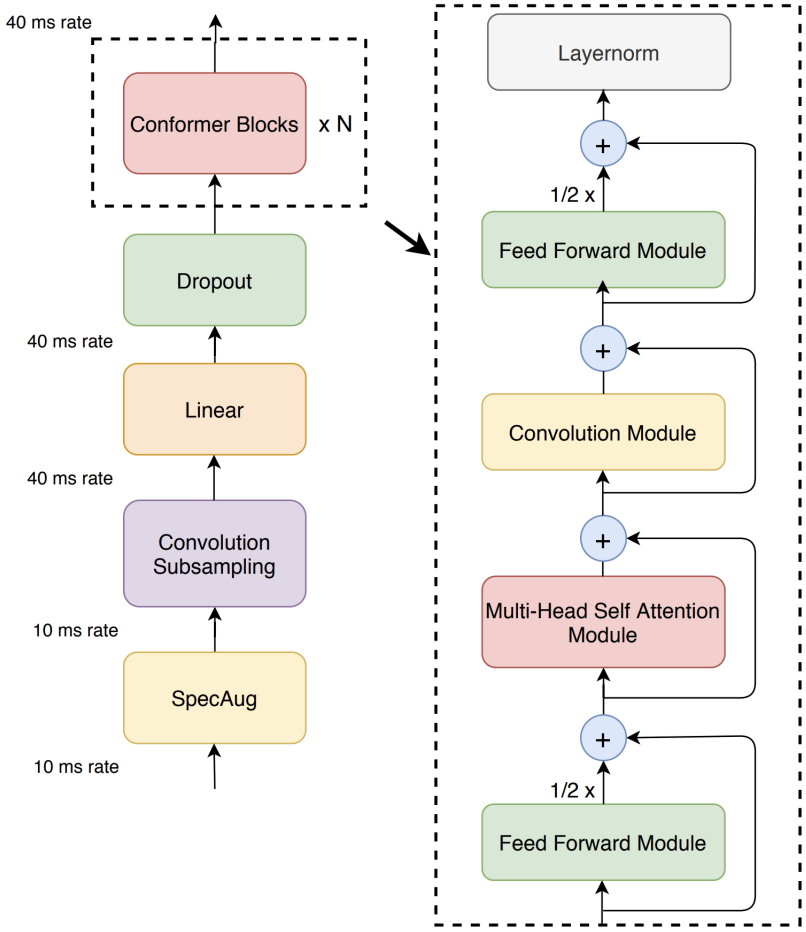
پروژه کانفورمر فارسی

دوره کارآموزی عصرگویش پرداز

محمد مهدی میررشید، حامد آجورلو

۱۹ شهریور ۱۴۰۱



۱ مقدمه

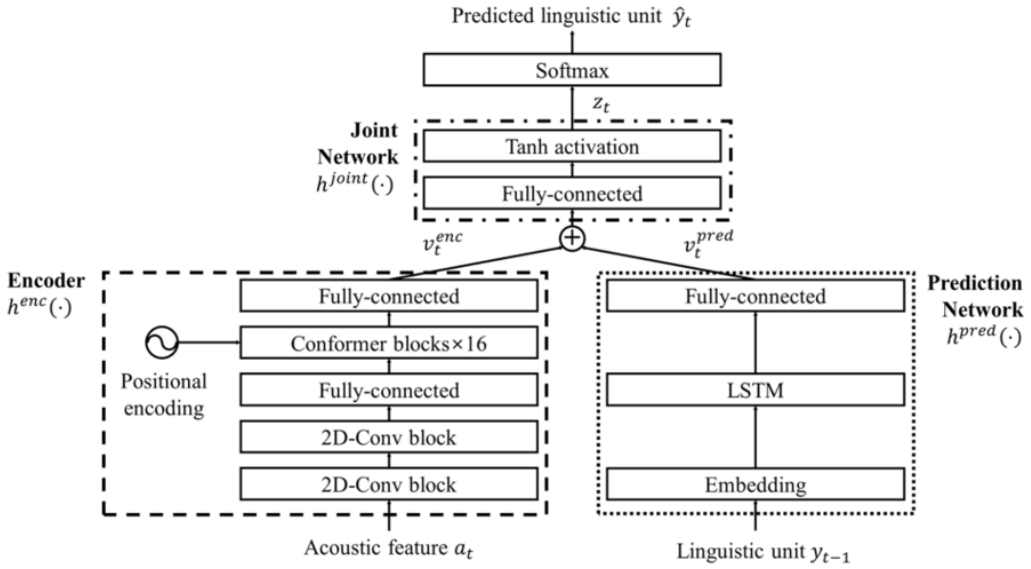
مدل کانفورمر (Conformer) مدل جدیدی در حوزه پردازش صوت است که شبکه های Transformerای را باCNN ترکیب کرده است. به نظر میرسد این مدل کار ارزشمندی باشد ولی به دلیل ماهیت ساختاری آن نتوانسته بر روی مدل های شرکت HuggingFace کار بکند و مستندات کمتری درباره آن وجود دارد.
در این پروژه ابتدا مقاله ی اصلی کانفورمر بررسی میشود، سپس مدل اولیه ای بر روی زبان انگلیسی از ابتدا آموزش داده میشود و در آخر هم تلاش بر این بوده که برای زبان فارسی بتوانیم مدلی داشته باشیم. با توجه به وجود مدل های رقیب برای کانفورمر، در این مقاله مدل ContextNet هم برای مقایسه بررسی میشود.
دیتاست استفاده شده برای آموزش اولیه روی زبان انگلیسی دیتاست AN4 است که با با آموزش اولی و بدون جست و جو برای پارامترهای بهینه و یا استفاده از مدل زبان، خطای $WER = 0.27$ را داشتیم. در ادامه برای آموزش مدل فارسی از دیتاست Mozilla Common Voice استفاده شد که شامل ۶ ساعت صوت و متن فارسی است.
در انجام آزمایش ها از جعبه ابزار Nemo و کتابخانه های مرتبط با آن استفاده شده. Nemo کتابخانه با انواع مدل های پردازش صوت است که توسط شرکت Nvidia توسعه داده شده. مدل های این کتابخانه روی زبان انگلیسی با دیتاستی چندین هزارساعته آموزش داده شده اند و اپن سورس هستند.
با توجه به محدودیت های سخت افزاری موجود و عدم تخصیص سرور برای آموزش مدل، آموزش این مدل کامل نشد ولی کد آموزش تکمیل شده و همینطور در اولین تلاش ها برای آموزش به خطای $WER = 0.62$ برای زبان فارسی رسیدیم. مشخصا این خطا مناسب نیست ولی آموزش کامل انجام نشده و جست و جویی هم برای انتخاب پارامتر بهینه صورت نگرفته است.
در ادامه هم تاثیرات انتخاب پارامتر های مختلف، مدل های مختلف و همینطور نکاتی برای ادامه کار بیان میشود.

۲ بررسی مدل های Conformer و ContextNet

مدل های Conformer و ContextNet هر دو مدل های تبدیل وویس به متن هستند و هر دو ساختار ترانسفورمری دارند. تفاوت اصلی این دو مدل در نحوه انجام انکودینگ است. در ادامه ابتدا با بررسی مقاله [۱] ساختار مدل جدید کانفورمر را میبینیم. سپس مقاله [۲] را بررسی میکنیم و با ساختار یکی از رقیب های اصلی کانفورمر آشنا میشویم.

۱.۲ مدل Conformer

شبکه های ترانسفورمری در چند سال اخیر موفقیت چشم گیری به عنوان مدل های پردازش صوت داشته اند. مزیت این مدل‌ها، توانایی درک ارتباطات کلی بین کلمات است (content-based global interactions). این توانایی در مقابل توانایی بالای مدل های مبتنی بر شبکه های



شکل ۱: ساختار کلی یک مدل کانفورمر

عصبی کانولوشنی برای درک ویژگی‌های محلی جملات است.
در مدل کانفورمر تلاش بر این است که با ترکیب هر دوی این رویکردها، مدلی با توانایی درک ویژگی‌های کلی و محلی ساخته شود. اتکا به صرفا شبکه‌های عصبی برای درک ویژگی های کلی محدودیت‌های واضحی مثل نیاز به تعداد لایه زیاد وجود دارد. راه حل مناسبی برای این موضوع، استفاده از مکانیزم توجه است که در حال حاضر به طور گسترده در مدل های مبتنی بر ترانسفورمر استفاده میشود.
ساختار مدل Conformer-Transducer معرفی شده در [۱] از یک انکودر کانفورمر و یک دیکودر یک لایه LSTM نشکیل شده. انکودر کانفورمر را در شکل ۱ میبینیم. انکودر ابتدا با روش‌های معمول مدل های مشابه مثل انجام کانولوشن و Spectogram Augmentation، صوت ورودی را پردازش میکند.
در ادامه خروجی حاصل از پردازش صوت وارد بلوک کانفورمر میشود. تفاوت اصلی انکودر این مدل با مدل های مشابه، استفاده از بلوک کانفورمر به جای ترانسفورمر است. بلوک کانفومر از چهار ماژول پشت هم تشکیل شده. یک ماژول خودتوجهی چندسر و یک ماژول کانولوشن که توسط دو لایه feed-forward احاطه شده اند. ایده استفاده از دو لایه به جای یک لایه از مدل ارائه شده در [۳] الهام گرفته شده و نتیجه بهتری میدهد.
خطاهای ارائه شده در مرجع [۱] در شکل ۲ آمده است. میبینیم خطای مدل کانفورمر با سایز بزرگ کمترین مقدار بین مدل های کانفورمر و همینطور بین تمامی مدل های بررسی شده است. همچنین میبینیم که افزایش سایز مدل کانفورمر باعث بهبود دقت مدل میشود.

Method	#Params (M)	WER Without LM		WER With LM	
		testclean	testother	testclean	testother
Hybrid					
Transformer [33]	-	-	-	2.26	4.85
CTC					
QuartzNet [9]	19	3.90	11.28	2.69	7.25
LAS					
Transformer [34]	270	2.89	6.98	2.33	5.17
Transformer [19]	-	2.2	5.6	2.6	5.7
LSTM	360	2.6	6.0	2.2	5.2
Transducer					
Transformer [7]	139	2.4	5.6	2.0	4.6
ContextNet(S) [10]	10.8	2.9	7.0	2.3	5.5
ContextNet(M) [10]	31.4	2.4	5.4	**2.0**	4.5
ContextNet(L) [10]	112.7	**2.1**	4.6	**1.9**	4.1
Conformer (Ours)					
Conformer(S)	10.3	**2.7**	**6.3**	**2.1**	**5.0**
Conformer(M)	30.7	**2.3**	**5.0**	**2.0**	**4.3**
Conformer(L)	118.8	**2.1**	**4.3**	**1.9**	**3.9**

شکل ۲: خطای انواع مدل‌های کانفورمر و ContextNet

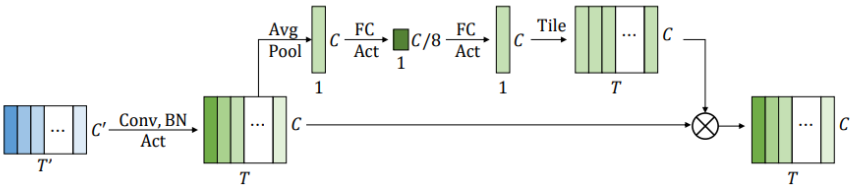
۲.۲ مدل ContextNet

برای درک بهتر مدل کانفورمر و مقایسه آن با دیگر مدل‌های ارائه شده، در این قسمت مدل ContextNet را بررسی میکنیم. همانند مدل کانفورمر، مدل ContextNet هم با هدف درک ارتباطات بین کلمات دور از هم ساخته شده. رویکرد این مدل استفاده از ماژول squeeze-and-excitation (SE) است که در [۴] معرفی شده.

عملکرد این ماژول به این صورت است که برای درک تعاملات بین کلمات دور از هم، ابتدا تمام خروجی‌های حاصل از عبور پنجره کانولوشن از روی ورودی را در یک بردار ذخیره میکند و سپس با انجام ضرب، این بردار را با بردارهای ویژگی های محلی حاصل از کانولوشن ترکیب میکند.

در شکل ۳ ماژول SE را مشاهده میکنیم. در ابتدا با عبور پنجره کانولوشن از روی ورودی، تعدادی بردار داریم که شامل ویژگی های محلی ورودی هستند. با میانگین‌گیری روی این بردارها، یک بردار حاصل میشود. سپس این بردار از دو لایه و تابع فعالسازی عبور میکند و سپس در بردارهای خروجی کانولوشن ضرب میشود.

ساختار نهایی مدل ContextNet همان مدل RNN-Transducer است [۵] که از سه بخش انکودر صوت، انکودر برچسب و یک شبکه مشترک برای دیکود کردن و ترکیب انکودرها تشکیل شده. تفاوت مدل با مدل های قبلی در انکودر صوت است که از مکانیزم توضیح داده شده برای انجام کانولوشن و پیداکردن ارتباطات بین کلمات تشکیل شده است.



شکل ۳: ماژول SE

در شکل ۲ هم دقت مدل ContextNet با سایز های مختلف را میبینیم. در بیشتر حالت ها دقت مدل ContextNet اختلاف خیلی کمی با مدل کانفورمر دارد.

۳ آزمایش‌ها

در ابتدا برای آشنایی با مدل کانفورمر، تلاش کردیم مدل را از ابتدا روی زبان انگلیسی آموزش بدهیم. در آزمایش اول آموزش‌ها روی مدل‌های پیش‌فرض انجام شدند. سپس تغییراتی روی پارامترهای مدل اعمال شد و دوباره آموزش با تنظیمات جدید انجام شد.
سپس آموزش روی زبان فارسی با حدود ۶ ساعت داده آموزش یک بار روی مدل پیش‌فرض و یک بار روی مدل با تنظیمات جدید انجام شد. در ادامه تنظیمات tokenizer تغییر کردند و دوباره آموزش با مدل کانفورمر کوچک پیش‌فرض انجام شد.
در مدل متداول کانفورمر کوچک شبکه ی پیوندی، دیکودر و انکودر به ترتیب دارای ۳۲۰، ۳۲۰ و ۱۷۶ بعد می باشد. مدل کانفورمر دوم با پارامتر های متفاوت در شبکه ی پیوندی، دیکودر و انکودر به ترتیب برابر ۲۸۰، ۱۴۰ ، ۲۸۰ آموزش داده شد.
این مدل ها یک بار با batch-size با اندازه ۱۶ و در ۸۰ epoch با نرخ یادگیری ۲.۵ آموزش داده شدند. یک بار هم بعد از تغییر تنظیمات پیش‌پردازش با batch-size با اندازه ۲۱ و نرخ یادگیری ۵ برای شروع و ۰.۲۵ بعد از ۳۰ epoch آموزش داده شدند.

۱.۳ ابزار مورد استفاده

برای انجام آموزش و آزمایش روی مدل‌ها، از ابزار Nemo [۶] استفاده شده است. Nemo مجموعه‌ای از کتابخانه ها و ماژول ها است که برای توسعه و آموزش مدل‌های پردازش صوت استفاده میشود. Nemo به صورت پیش‌فرض در مدل‌های آماده خود مدل کانفورمر و همینطور مدل ContextNet را دارد. این مدل‌ها هم به صورت آموزش‌دیده روی زبان انگلیسی و هم به صورت مدل بدون آموزش در Nemo موجود هستند.

برای انجام آموزش از محیط Google Colab استفاده شد. در نسخه رایگان این پلتفرم، سروری با پردازنده گرافیکی Nvidia K80 و حافظه گرافیکی 12GB و یا 16GB در به کاربر اختصاص پیدا میکند. زمان استفاده از سرور به چند ساعت در روز محدود شده و به همین دلیل نتوانستیم آموزش را به مدت طولانی انجام دهیم.

۲.۳ دیتاست های مورد استفاده

برای آموزش‌های اولیه روی مدل زبان انگلیسی از دیتاست AN4 استفاده شد. AN4 دیتاست کوچکی است که شامل کلیپ‌های صوتی از خواندن آدرس، اسم و غیره میبااشد [۷].
در ادامه برای آموزش فارسی از نسخه فارسی دیتاست Mozilla Common Voice 5.1 استفاده شد. این دیتاست شامل حدود ۳۰۰ ساعت صوت و متن فارسی است و به صورت متن‌باز در اختیار عموم قرار دارد. بخش آموزش این دیتاست شامل ۶ ساعت صوت فارسی و متن است که از این داده‌ها برای آموزش مدل استفاده شده [۸].

۳.۳ نتایج

۱. آموزش روی زبان انگلیسی: با مدل پیش‌فرض کانفورمر کوچک، خطا روی دیتاست AN4 به حدود WER=0.47 رسید. مشخصا این مقدار خطا با توجه به سایز کوچک دیتاست مناسب نبود و این بار پارامترهای مدل روی مقادیر کوچک‌تری تنظیم شدند. با تنظیمات جدید خطا بعد از ۷۰ epoch به WER=0.27 رسید که برای مدل اولیه دقت مناسبی بود. پارامترهای این مدل جدید به صورت ابعاد ۶۴ برای انکودر، دیکودر و شبکه پیوندی تنظیم شدند.

۲. آموزش روی زبان فارسی: در این بخش چندین آزمایش انجام شدند. آزمایش‌های اولیه با استفاده از یک tokenizer از نوع spe با vocab-size = 35 انجام شدند. در این آزمایش‌ها مدل پیش‌فرض کانفورمر بعد از ۸۰ epoch به خطای WER=0.72 رسید و مدل با تنظیمات متفاوت کانفورمر به WER=0.88. در هر دو حالت میبینیم که خطای خیلی بالایی داریم. همینطور زمان آموزش هم نسبتا طولانی است و نتوانستیم مدل را روی محیط colab به مدت مناسب آموزش بدهیم.
در این آزمایش‌ها batch-size روی ۱۶ و نرخ یادگیری روی مقدار ثابت ۵ تنظیم شده بود.
در ادامه بعد از بررسی مدل، ابتدا نحوه آموزش به این صورت تغییر کرد که از مدل آموزش‌داده شده انگلیسی برای شروع آموزش استفاده شد. همینطور در تنظیمات tokenizer مورد استفاده، اندازه vocab-size به ۱۰۲۴ تغییر کرد. با انجام این تغییرات این بار بعد از ۲۰ epoch به خطای WER=0.66 رسیدیم که هم از نظر زمانی و هم از نظر خطا نسبت به حالت اولیه بهبود پیدا کرده بود. در اینجا به نظر میرسید که خطای مدل در حال کاهش نیست و به همین خاطر نرخ یادگیری به ۰.۲۵ کاهش پیدا کرد. با انجام این کار با انجام آموزش برای ۱۰ epoch بیشتر خطا به WER=0.62 رسید.

مشخصا این بار هم هنوز خطا مناسب نبود ولی با توجه به دسترسی به دیتاست محدود و همینطور عدم تخصیص سرور، به نظر میرسد تنظیمات مدل مناسب هستند و فقط نیاز به آموزش با دیتاست طولانی‌تر و زمان بیشتر داریم.
در شکل ۴ دو نمونه از پیش‌بینی های مدل را میبینیم. مشخصا برای جمله‌های ساده‌تر و کوتاه‌تر مدل دقت بالاتری دارد و وقتی جمله‌ها پیچیده باشند دقت مدل مناسب نیست.

۴.۳ مقایسه با مدل ContextNet

به طور کلی مدل ContextNet در آزمایش‌های انجام شده خطای مشابهی با مدل کانفورمر داشت. در آزمایش‌های اولیه با مدل‌های پیش‌فرض روی دیتاست AN4، خطای ContextNet به حداقل ۰.۵ رسید. در ادامه کاهش سایز مدل ContextNet با کاهش تعداد فیترها به ۱۲۸، و همچنین کاهش ابعاد لایه های پیش‌بینی و پیوندی به ۶۴ انجام شد. در این حالت خطا به WER=0.22 رسید.
در ادامه با آموزش روی مدل فارسی، خطای مدل ContextNet-512 که یکی از انواع پیش‌فرض مدل در Nemo است به حدود WER=0.82 رسید که همچنان مقدار مناسبی نبود. البته این آزمایش بدون افزایش اندازه vocab-size در tokenizer مورد استفاده انجام شد.
میبینیم که خطای بالای مدل ویژگی مشترکی بین هر دو نوع مدل است. از این موضوع میتوان حدس زد که دیتاست کوچک استفاده‌شده و همینطور عدم تخصیص سرور برای انجام آموزش طولانی، مشکلات اصلی در آزمایش‌ها بوده اند.

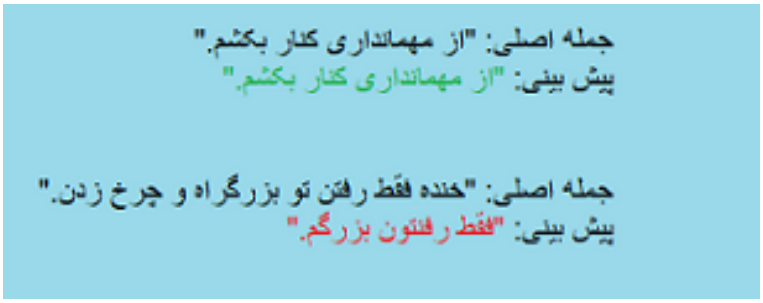
۵.۳ بررسی پارامترها با توجه به مطالعات و آزمایش ها

به طور کلی پارامترهای موثر به سه دسته تقسیم میشوند:

۱. پارامترهای پیش‌پردازش: اینجا پارامتر پیش‌پردازشی که بیشترین تاثیر روی خطای مدل را داشت عدد Vocab-Sizeدر Tokenizer مورد استفاده بود. ابتدا Vocab-Size روی حدود ۳۰ تنظیم شده بود که معادل داشتن دایره لغاتی است که فقط شامل حروف الفبا میشود. ولی با افزایش سایز دایره‌لغات، کلمات و حروف پشت هم پرتکرار هم میتوانند به صورت مجزا و مستقل در نظر گرفته بشوند.

۲. پارامترهای مدل: پارامترهای مدل شامل پارامترهای انکودر، دیکودر و شبکه‌پیوندی می‌شود. با آزمایش‌های انجام شده به خصوص روی دیتاست AN4 میبینیم که برای داده کم نیاز به کاهش همه پارامترهای مدل داریم. با افزایش‌داده‌آموزش مورد استفاده، نیاز به استفاده از مدل با سایز بزرگتر داریم. همینطور دقت میکنیم که تعداد پارامترهای انکودر مدل کانفورمر بیشترین تاثیر را روی سایز مدل دارد و برای داشتن مدل با حجم معقول نیاز به کنترل تعداد پارامترهای این مدل داریم.

۳. پارامترهای آموزش‌دهنده: در این قسمت پارامترهایی مثل نرخ یادگیری و batch-size را داریم. مقدار پیش‌فرض برای نرخ یادگیری کانفورمر ۵ است که عدد بزرگی است.



شکل ۴: دو نمونه از پیش‌بینی های مدل کانفورمر. میبینیم برای جمله کوتاه‌تر دقت بهتری داریم.

طبق آزمایش‌های مختلفی که انجام دادیم بعد از حدود ۲۰ epoch نیاز به کاهش نرخ یادگیری داریم. همینطور در حالت پیش‌فرض نرخ یادگیری کاهش پیدا نمیکند که باز هم طبق آزمایشات انجام شده نتیجه مطلوبی نمی‌دهد.
همینطور طبق جست‌وجوی انجام شده مدل موجود در ابزار Nemo با batch-size حداقل ۲۵۶ آموزش داده شده [۹] که عدد بسیار بزرگتری از مقدار استفاده شده در آزمایشات ما است. افزایش این عدد به بیش از ۲۱ که در آزمایش های ما انجام شد نیاز به حافظه گرافیکی بزرگتر از حافظه تخصیص‌یافته در Colab داشت و به همین دلیل نتوانستیم این عدد را روی مقدار بزرگتری تنظیم کنیم. این موضوع میتواند روی خطای بدست آمده را به طور قابل توجهی تحت تاثیر قرار بدهد چون batch-size تخمینی از جهت مناسب برای حرکت در الگوریتم gradient-descent می‌دهد و اگر این تخمین به اندازه کافی دقیق نباشد ممکن است به نقطه بهینه نرسیم [۱۰].

۴ نتیجه‌گیری

با آزمایش‌های انجام‌شده میبینیم که خطای مدل‌ها بالاست. با توجه به سایز دیتاست مورداستفاده که فقط شامل ۶ ساعت داده آموزش فارسی بود و همینطور با توجه به محدودیت‌های موجود برای استفاده از سرور، انتظار این موضوع را داشتیم (برای مثال در مستندات Nemo در [۱۱] میبینیم که دقت مدل ژاپنی آموزش دیده شده روی دیتاست Mozilla Common Voice با چند ساعت داده آموزشی به حداقل خطای حدود WER=0.5 میرسد.) و برای آموزش یک مدل مناسب نیاز به داده‌های بیشتر و زمان آموزش بیشتر داریم.
همینطور دیدیم که این خطای بالا به مدل کانفورمر محدود نمیشود و مدل ContextNet که از مدل‌های رقیب کانفورمر است هم همین خطاها را داشت.
کدهای استفاده‌شده در گزارش در گیت‌هاب شرکت عصر گویش‌پرداز قرار خواهند گرفت و برای ادامه کار قابل دسترسی هستند. همینطور یک مدل آموزش داده شده با خطای WER=0.62 موجود است که میتوان از آن برای ادامه آموزش استفاده کرد. به طور کلی به نظر میرسد

پارامترهای مدل به درستی تنظیم شده‌اند و برای داشتن یک مدل مناسب کانفورمر فارسی فقط نیاز به زمان اجرای بیشتر و دیتاست بزرگتر داریم.

مراجع

[۱] Gulati, A., Qin, J., Chiu, C.-C., Parmar, N., Zhang, Y., Yu, J., Han, W., Wang, S., Zhang, Z., Wu, Y., & Pang, R. (2020, May 16). Conformer: Convolution-augmented transformer for speech recognition. arXiv.org. Retrieved September 8, 2022, from https://arxiv.org/abs/2005.08100

[۲] Han, W., Zhang, Z., Zhang, Y., Yu, J., Chiu, C.-C., Qin, J., Gulati, A., Pang, R., & Wu, Y. (2020, May 16). ContextNet: Improving convolutional neural networks for automatic speech recognition with Global Context. arXiv.org. Retrieved September 8, 2022, from https://arxiv.org/abs/2005.03191

[۳] Y. Lu, Z. Li, D. He, Z. Sun, B. Dong, T. Qin, L. Wang, and T.-Y. Liu, "Understanding and improving transformer from a multi-particle dynamic system point of view," arXiv preprint arXiv:1906.02762, 2019.

[۴] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 7132–7141.

[۵] A. Graves, "Sequence transduction with recurrent neural networks," arXiv preprint arXiv:1211.3711, 2012

[۶] https://developer.nvidia.com/nvidia-nemo

[۷] https://huggingface.co/datasets/espnet/an4

[۸] https://commonvoice.mozilla.org/fa

[۹] https://github.com/NVIDIA/NeMo/issues/3288

[۱۰] https://deeplizard.com/learn/video/U4WB9p6ODjM

[۱۱] colab.research.google.com/github/NVIDIA/NeMo/blob/stable/tutorials/asr/ASR_CTC_Language_Finetuning.ipynb