Speech Enhancement and Denoising

پروژه پایانی درس پردازش گفتار

دانشکده مهندسی کامپیوتر

استاد: دکتر صامتی

بهار ۱۴۰۳

بهار بهزادی‌پور ۴۰۲۲۰۷۱۵۵

مهتا فطرت ۴۰۲۲۱۲۲۲۵

مخزن پروژه

فهرست مطالب

فصل ۱

مقدمه

بهبود کیفیت گفتار و رفع نویز آن همواره مورد توجه بوده‌است و جزء معدود مسائل حوزه‌ی پردازش گفتار است که می‌توان گفت هنوز کاملا حل‌شده تلقی نمی‌شو و جای پیشرفت دارد. با این حال، مدل‌ها و ابزارهای بسیاری برای بهبود گفتار زبان انگلیسی وجود دارد که از کیفیت مطلوبی نیز برخوردار هستند. در این میان، زبان فارسی نسبت له تکنولوژی‌های اخیر در این زمینه جای پیشرفت زیادی دارد. بررسی‌های اولیه نشان می‌دهد که مدل بهبود گفتاری که برای زبان فارسی مناسب‌سازی شده باشد در حال حاضر وجود ندارد. در این پروژه ما در تلاش هستیم تا وضعیت ابزارهای حاضر را برای بهبود گفتار فارسی بررسی کنیم و بتوانیم در راه بهبود آن‌ها گام‌های موثری برداریم.

۱.۱ شرح مسئله

بهبود گفتار (Speech Enhancement) حذف نویز (Denoising) به فرآیندهایی اطلاق میشود که کیفیت و وضوح سیگنال گفتار را بهبود میبخشند و نویزهای پس زمینه را کاهش میدهند. این مسئله در کاربردهای مختلفی از جمله سیستم های تشخیص گفتار، ارتباطات تلفنی، و کمک به افراد با مشکلات شنوایی اهمیت دارد. در زبان فارسی، به دلیل تفاوت های صوتی و ساختاری با دیگر زبان ها، نیاز به تحقیق و توسعه ویژه ای در این زمینه وجود دارد.

۲.۱ رویکردها

در مورد مسئله‌ی بهبود گفتار و رفع نویز مانند بسیاری از مسائل دیگر در حوزه‌ی گفتار، دو رویکرد کلی وجود دارد. اولین رویکرد مربوط به روش‌های rule-based می‌باشد. این روش‌ها عموما مبتنی بر تکنیک‌های پردازش سیگنال می‌باشند و به مشاهده‌ی نمونه‌ی گفتارهای تمیز و نویزی وابسته نیستند. رویکرد دوم اما مبتنی بر شبکه‌های عصبی می‌باشد. در این روش‌ها، با داشتن یک معماری مناسب و تعداد زیادی از نمونه‌های ورودی و خروجی مطلوب، مدل می‌آموزد که تسک بهبود گفتار را انجام دهد. در این پروژه، ما هر دوی این روش‌ها را بررسی می‌کنیم و راه‌حل‌هایی بر اساس هر یک ارائه می‌دهیم.

۳.۱ چالش‌ها

همانطور که برای سایر راه‌حل‌های مبتنی بر شبکه‌های عصبی چالش داده مطرح است، در این‌جا هم با این مشکل مواجه هستیم. درواقع تهیه‌ی دادگانی طبیعی متشکل از زوج‌های تمیز و نویزی یک گفتار واحد، امری دشوار و زمان‌بر است. به همین جهت غالبا شاهد دیتاست‌هایی هستیم که به صورت اتوماتیک generate شده‌اند. در ادامه به این موضوع بیشتر پرداخته می‌شود.

۴.۱ اهداف و دستاوردهای این پروژه

در این پروژه ما ابتدا روش‌های موجود را برای زبان فارسی ارزیابی کردیم و امکان خاص‌سازی آن‌ها برای زبان فارسی را بررسی نمودیم. سپس از بین روش‌های موجود، سعی در بهبود برخی از این روش‌ها به کمک تکنیک‌هایی چون fine-tuning داشتیم. همچنین به عنوان یک جایگزین، ابزاری rule-based برای بهبود گفتار فارسی نیز ارائه دادیم که حتی بدون داده‌های آموزش نیز قابل استفاده است. لازم به ذکر است که در این پروژه، دو نوع دیتاست نیز برای تسک Speech Enhancement and Denoising برای زبان فارسی ارائه می‌شود که to the best of our knowledge اولین دیتاست‌های این زبان در این تسک، حتی از نوع ساختگی آن می‌باشد.

فصل ۲

مرور کارهای پیشین

در این بخش به بررسی برخی از مهم‌ترین راه‌حل‌های موجود برای Speech Enhancement and Denoising می‌پردازیم.

۱.۲ Metricgan

تفاوت بین تابع هزینه‌ای که برای آموزش مدل بهبود گفتار استفاده می‌شود و درک شنیداری انسان معمولاً باعث می‌شود که کیفیت گفتار بهبود یافته رضایت‌بخش نباشد. معیارهای ارزیابی عینی که درک انسان را در نظر می‌گیرند می‌توانند به عنوان پلی برای کاهش این فاصله عمل کنند. مدل MetricGAN که قبلاً پیشنهاد شده بود، برای بهینه‌سازی معیارهای عینی با اتصال معیار به یک تفکیک‌کننده طراحی شده بود. از آنجا که در طول آموزش تنها به امتیازات توابع ارزیابی هدف نیاز است، معیارها حتی می‌توانند غیرقابل تفکیک باشند. در این مطالعه، ما MetricGAN+ را پیشنهاد می‌کنیم که در آن سه تکنیک آموزشی که دانش حوزه پردازش گفتار را در خود دارند، پیشنهاد شده است. با این تکنیک‌ها، نتایج آزمایشی بر روی مجموعه داده VoiceBank-DEMAND نشان می‌دهد که MetricGAN+ می‌تواند امتیاز PESQ را نسبت به مدل قبلی MetricGAN به میزان ۰.۳ افزایش دهد و به نتایج برتر دست یابد (امتیاز PESQ = 3.15). ایده اصلی MetricGAN شبیه‌سازی رفتار یک تابع ارزیابی هدف (مثلاً تابع PESQ) با یک شبکه عصبی (مثلاً Quality-Net [۲۰]) است. تابع ارزیابی جانشین از امتیازات خام یاد گرفته می‌شود و تابع ارزیابی هدف را به عنوان یک جعبه سیاه در نظر می‌گیرد. هنگامی که ارزیابی جانشین آموزش داده شد، می‌توان از آن به عنوان یک تابع هزینه برای مدل بهبود گفتار استفاده کرد. متأسفانه، یک جانشین ایستا به راحتی توسط نمونه‌های تقلبی فریب می‌خورد. برای بهبود عملکرد چارچوب MetricGAN، برخی تکنیک‌های پیشرفته یادگیری پیشنهاد شده‌اند. در طی این تحقیق، عواملی که به‌طور قابل‌توجهی بر عملکرد یا کارایی آموزش تأثیر می‌گذارند نیز بررسی می‌شوند. بهبود MetricGAN+ عمدتاً از طریق سه تغییر زیر حاصل می‌شود.

۱. یادگیری امتیازات معیار برای گفتار نویزی

۲. نمونه‌ها از بافر بازپخش تجربه

۳. تابع سیگموید قابل یادگیری برای تخمین ماسک

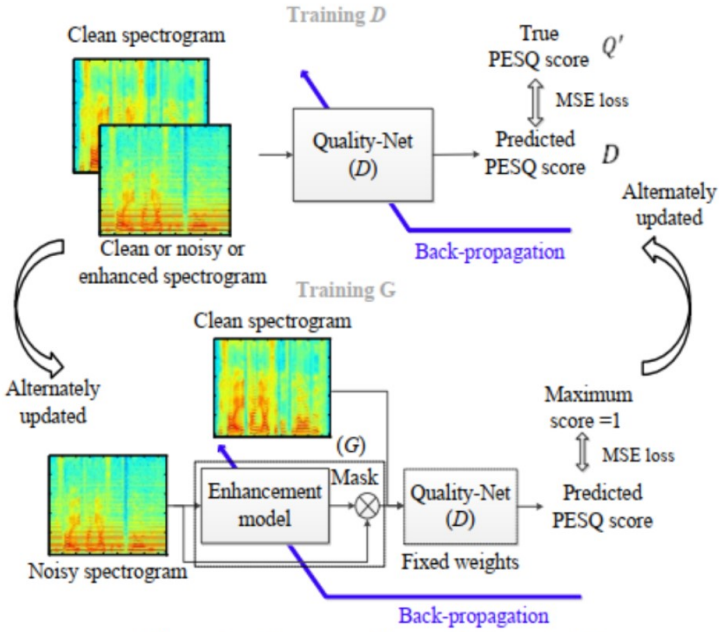


Figure 1: *Training flow of MetricGAN.*

شکل ۱.۲: آموزش metricgan

در این مطالعه، چندین تکنیک برای بهبود عملکرد چارچوب MetricGAN پیشنهاد کردیم. ما متوجه شدیم که شامل کردن گفتار نویزی برای آموزش تفکیک‌کننده و استفاده از سیگموید قابل یادگیری، مفیدترین تکنیک‌ها هستند. MetricGAN+ ما نتایج پیشرفته‌ای را بر روی مجموعه داده‌های VoiceBank-DEMAND به‌دست می‌آورد و امتیازهای PESQ می‌تواند به ترتیب 0.3 و 0.45 نسبت به MetricGAN و BLSTM (MSE) افزایش یابد.

Diff-TTS ۲.۲

با وجود اینکه مدل‌های تبدیل متن به گفتار (TTS) عصبی توجه زیادی را جلب کرده و در تولید گفتار شبیه به انسان موفق بوده‌اند، هنوز جای پیشرفت‌هایی برای طبیعی‌تر و کارآمدتر کردن آن‌ها وجود دارد. در این کار، ما یک مدل TTS غیر اتورگرسیو جدید به نام Diff-TTS پیشنهاد می‌کنیم که به تولید گفتار با کیفیت بالا و کارآمدی بالا دست می‌یابد. با توجه به متن، Diff-TTS از یک چارچوب دیفیوژن نویززدایی برای تبدیل سیگنال نویز به طیف‌نگار مل از طریق مراحل زمان دیفیوژن استفاده می‌کند. به‌منظور یادگیری توزیع طیف‌نگار مل با شرط متن، ما یک روش بهینه‌سازی مبتنی بر احتمال برای TTS ارائه می‌دهیم. علاوه بر این، برای افزایش سرعت استنتاج، ما از روش نمونه‌برداری تسریع‌شده استفاده می‌کنیم که به Diff-TTS امکان می‌دهد تا موج‌نگاشت‌های خام را به‌طور بسیار سریع‌تری تولید کند بدون اینکه کیفیت ادراکی به‌طور قابل توجهی کاهش یابد. از طریق آزمایش‌ها، تایید کردیم که Diff-TTS با یک GPU NVIDIA 2080Ti به‌طور ۲۸ برابر سریع‌تر از زمان واقعی تولید می‌کند. دیف-TTS توزیع نویز

را به توزیع مل‌ـاسپکتروگرام متناظر با متن داده شده تبدیل می‌کند. همان‌طور که در شکل ۱ نشان داده شده است، مل‌ـاسپکتروگرام به تدریج با نویز گوسی تخریب شده و به متغیرهای نهان تبدیل می‌شود. این فرآیند، فرآیند انتشار نامیده می‌شود.

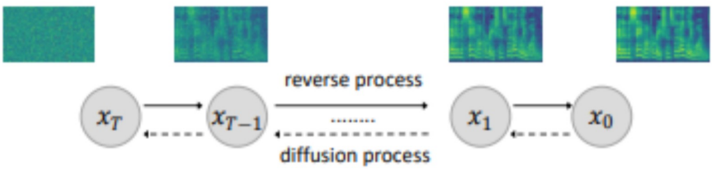


Figure 1: *Graphical model for the reverse process and the diffusion process.*

شکل ۲.۲: فرایند diffusion

فرض کنید x۱، . . . ، xT یک دنباله از متغیرها با ابعاد یکسان باشد که در آن t = ۰، ۱، . . . ، T شاخصی برای مراحل زمانی انتشار است. سپس، فرآیند انتشار مل‌ـاسپکتروگرام x۰ را از طریق یک زنجیره انتقال‌های مارکوفی به نویز گوسی xT تبدیل می‌کند. هر مرحله انتقال با یک برنامه واریانس $\beta_1, \beta_2, ..., \beta_T$ از پیش تعیین شده است. به طور خاص، هر تبدیل مطابق با احتمال انتقال مارکوفی q(xtlxt-۱، c) انجام می‌شود که مستقل از متن c فرض می‌شود و به صورت زیر تعریف می‌شود:

$$q(x_t|x_{t-1}, c) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t I).$$

کل فرآیند انتشار $q(x1 : Tx0, c)$ یک فرآیند مارکوفی است و می‌تواند به صورت زیر تجزیه شود:

$$q(x_1 \ldots, x_T|x_0, c) = \prod_{t=1}^{T} q(x_t|x_{t-1}).$$

فرآیند معکوس یک روش تولید مل‌ـاسپکتروگرام است که دقیقاً برخلاف فرآیند انتشار عمل می‌کند. برخلاف فرآیند انتشار، هدف فرآیند معکوس بازیابی یک مل‌ـاسپکتروگرام از نویز گوسی است. فرآیند معکوس به عنوان توزیع شرطی $p\theta(x0 : T1|xT, c)$ تعریف می‌شود و می‌تواند بر اساس خاصیت زنجیره مارکوف به چندین انتقال تجزیه شود:
دیف‌ـTTS شامل یک رمزگذار متن، رمزگذار مرحله، پیش‌بینی‌کننده مدت زمان، و رمزگشا است. شبکه رمزگشا شامل یک پشته از ۱۲ بلوک مقاوم با Conv۱D، sigmoid tanh و کانولوشن‌های ۱x۱ با ۵۱۲ کانال مقاوم است [۲۹]. همان‌طور که در شکل ۳ نشان داده شده است، تعبیه فونم توسط تنظیم‌کننده طول گسترش می‌یابد. سپس، تعبیه فونم و خروجی رمزگذار مرحله به ورودی پس از لایه Conv۱D اضافه می‌شود. لایه Conv۱D اندازه هسته‌ای برابر ۳ بدون گشادگی دارد. پس از عبور از این بلوک مقاوم، خروجی‌ها قبل از پس‌ـنت جمع می‌شوند. در نهایت، رمزگشا نویز گوسی متناظر با دنباله فونم و مرحله زمانی انتشار را به دست می‌آورد.

$$p_\theta(x_0 \dots, x_{T-1}|x_T, c) = \prod_{t=1}^{T} p_\theta(x_{t-1}|x_t, c).$$

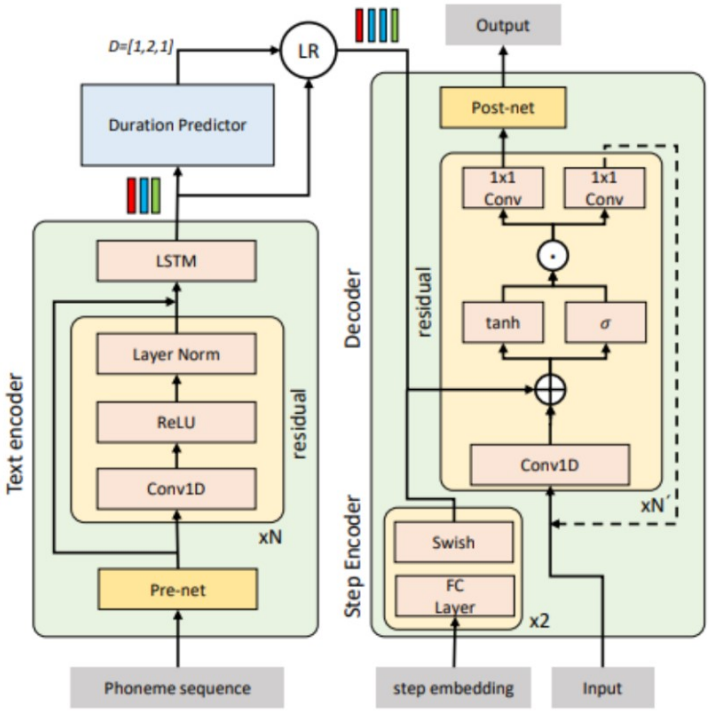


Figure 3: *The network architecture of Diff-TTS*

شکل ۳.۲: معماری diff-tts

۳.۲ A Study on Speech Enhancement Based on Diffusion Probabilistic Model

مدل‌های احتمالاتی انتشار توانایی فوق‌العاده‌ای در مدل‌سازی تصاویر طبیعی و فرم‌های صوتی خام از طریق فرآیندهای جفت‌شده انتشار و معکوس نشان داده‌اند. ویژگی منحصر به فرد فرآیند معکوس (یعنی حذف سیگنال‌های غیرهدف از نویز گوسی و سیگنال‌های نویزی) می‌تواند برای بازیابی سیگنال‌های تمیز استفاده شود. بر اساس این ویژگی، ما مدل بهبود گفتار مبتنی بر مدل احتمالاتی انتشار (DiffuSE) را پیشنهاد می‌کنیم که هدف آن بازیابی سیگنال‌های گفتاری تمیز از سیگنال‌های نویزی است. معماری اساسی مدل DiffuSE پیشنهادی مشابه معماری DiffWave است—مدل تولید فرم صوتی با کیفیت بالا که هزینه محاسباتی و ردپای نسبتاً پایینی دارد. برای دستیابی به عملکرد بهتر در بهبود، ما فرآیند معکوس پیشرفته‌ای طراحی کردیم که به آن فرآیند معکوس حمایتی گفته می‌شود و در هر مرحله زمانی، گفتار نویزی را به گفتار پیش‌بینی‌شده اضافه می‌کند. نتایج تجربی نشان می‌دهد که DiffuSE عملکردی معادل

با مدل‌های تولید صوت مرتبط در وظیفه SE مجموعه داده Bank Voice استاندارد شده دارد. علاوه بر این، نسبت به برنامه نمونه‌برداری کامل معمولاً پیشنهاد شده، فرآیند معکوس حمایتی پیشنهادی به ویژه سرعت نمونه‌برداری سریع را بهبود بخشیده و با انجام چندین مرحله نتایج بهبود بهتری نسبت به فرآیند استنتاج کامل سنتی ارائه می‌دهد. در مدل پیشنهادی، DiffuSE ما یک فرآیند معکوس حمایتی جدید را استخراج می‌کنیم تا جایگزین فرآیند معکوس اصلی شود و سیگنال‌های نویز را به طور مؤثرتری از ورودی نویزی حذف کنیم.

در مدل احتمالی انتشار اصلی، نویز گوسی در فرآیند معکوس اعمال می‌شود. از آنجا که سیگنال گفتار تمیز در طول فرآیند معکوس دیده نمی‌شود، سیگنال گفتار محاسبه شده xt ممکن است در طول فرآیند معکوس از مرحله T تا t+۱ تحریف شود. برای حل این مشکل، ما فرآیند معکوس حمایتی را پیشنهاد دادیم، که فرآیند نمونه‌برداری را از سیگنال گفتار نویزی y آغاز می‌کند و y را در هر مرحله معکوس ترکیب می‌کند در حالی که سیگنال گوسی اضافی را کاهش می‌دهد. شکل ۲ ساختار مدل DiffuSE را نشان می‌دهد. همانند DiffWave، تنظیم‌کننده در DiffuSE هدفش حفظ شباهت سیگنال خروجی به سیگنال گفتار هدف است، که اجازه می‌دهد نویز و گفتار تمیز را از داده‌های مخلوط جدا کند.

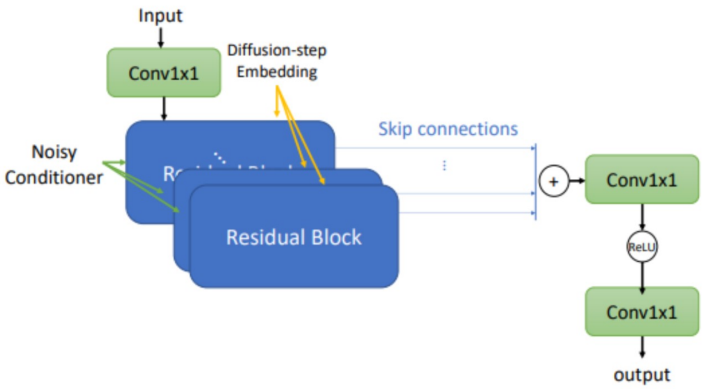


Fig. 2. The architecture of the proposed DiffuSE model

شکل ۴.۲: معماری diffuSE

برای تولید سیگنال‌های گفتار با کیفیت بالا، مدل DiffuSE را با ویژگی‌های Mel-spectral تمیز پیش‌آموزی کردیم. در DiffWave، اطلاعات شرطی مستقیماً از گفتار تمیز گرفته می‌شود که به مدل اجازه می‌دهد تا گفتار تمیز و نویز را از سیگنال‌های مخلوط جدا کند. پس از پیش‌آموزش، تنظیم‌کننده را از ویژگی‌های Mel-spectral تمیز به ویژگی‌های spectral نویزی تغییر دادیم، پارامترهای رمزگذار تنظیم‌کننده را مجدداً تنظیم کردیم و دیگر پارامترها را برای آموزش SE حفظ کردیم. در نهایت از نمونه‌برداری سریع استفاده شد تا تعداد مراحل حذف نویز کاهش پیدا کند.

فصل ۳

دادگان

اگرچه دادگان انگلیسی میتوانند برای آموزش مدل پایه از صفر به کار آیند، لازم است که حداقل یک مرحله ی fine-tuning بر روی زبان فارسی انجام شود. طی جستجوهایی که انجام شد، دیتاست بزرگ و شناخته شدهای برای تسک Denoising and Enhancement Speech برایزبان فارسی وجود ندارد. بنابراین نیاز است تا دیتاستی شامل جفت صوت های تمیز و نویزدار بسازیم. ما در این پروژه دو رویکرد را در پیش گرفتیم و نهایتا دو دیتاست synthesized و recorded را ارائه نمودیم.

۱.۳ دیتاست synthesized

در این بخش ما از صداهای validated بخش فارسی دیتاست CommonVoice استفاده کردیم. به این ترتیب که تعدادی از این صداها را به عنوان صوت تمیز در نظر گرفتیم و سپس معادل نویزی آن‌ها را به صورت اتوماتیک تولید کردیم. در این فرایند، هر صوت با احتمال یک‌دوم با یک نویز بک‌گراند از نویز gaussian ترکیب می‌شود و با احتمال یک‌دوم هم با یک صوت صدای بک‌گراند از نوع crowd talking. برای این اصوات، ما مجموعه‌ی کوچکی از فایل‌های صوتی crowd talking را از pixabay جمع‌آوری کردیم و به صورت رندوم، هر بار یک مورد از آن‌ها را برگزیده و با صوت تمیز تلفیق کردیم. نتیجه‌ی این بخش ۶۰۰۰ زوج صوت تمیز و نویزی سنتزشده می باشد که ۱۰۰۰ مورد از آن‌ها به عنوان مجموعه‌ی تست و ۵۰۰۰ مورد از آن‌ها به عنوان مجموعه‌ی train جدا شدند. subset تست این دادگان در مخزن پروژه بارگزاری شده‌است.

۲.۳ دیتاست recorded

در این بخش ما از یک مجموعه داده شامل صوت های تمیز فارسی استفاده کردیم و دقیقا همین اصوات را در محیط‌هایی با یک صدای بکگراند پخش کردیم و مجددا ضبط کردیم. به این ترتیب هر دوی داده‌ی تمیز و نویزی شامل گفتار عینا یکسانی می‌شدند اما دیگر صدای بک‌گراند به صورت مصنوعی به داده اضافه نشده بود. ایده‌ی اصلی این بود که تمام صداهای محیطی نیز کاملا طبیعی باشند. اما به علت محدودیت‌های موجود، بخشی از این نمونه‌ها با صدای طبیعی محیط (مانند صدای طبیعت یا فن) هستند و برخی دیگر با یک صدای آماده که در محیط پخش شده‌است.

برای دیتاست اصوات تمیز گزینه‌های زیادی از میان دیتاست‌ها وجود داشت. از جمله:

- persian-tts-dataset
- persian-texttospeech-audio
- ParsiGoo
- Persian SpeechCorpus
- PersianSpeech
- farsi ASR youtube
- CommonVoice
- Shenasa ai

ما در این پروژه از صوت های دیتاست persian-tts-dataset به عنوان صوت های تمیز بدون نویز استفاده کردیم. برای ساختن صوت های نویز دار متناظر، از ۵ صدای پس زمینه زیر استفاده کردیم:

- صدای محیط (صدای طبیعی محیط) –> ۵ صوت
- صدای فن (صدای طبیعی فن) –> ۲۰ صوت
- صدای باران (ضبط شده آماده) –> ۲۴ صوت
- صدای شهر (ضبط شده آماده) –> ۲۵ صوت
- صدای جنگل (ضبط شده آماده) –> ۱۸ صوت
- صدای تشویق ورزشگاه (ضبط شده آماده) –> ۳۰ صوت

دو صدای نویز اول مربوط به صدای محیط و فن هستند که در این دو محیط قرار گرفته و صدای تمیز را پخش کرده و ضبط کردیم. برای ۴ مورود دیگر که صداهای از قبل ضبط شده پس زمینه هستند، به طور همزمان صدای تمیز و نویز را اجرا کرده و صدا را ضبط کردیم تا صدای نویزدار را به دست آوریم. در نهایت توانستیم ۱۲۲ جفت داده صدای تمیز و صدای نویزدار را به دست آوریم. این دیتاست نیز در مخزن پروژه بارگزاری شده‌است و برای fine-tuning مورد استفاده قرار گرفته است.

فصل ۴

متدولوژی

همانطور که در بخش مقدمه ذکر شد، ما دو رویکرد rule-based و رویکرد مبتنی بر شبکه‌های عصبی را در پیش گرفتیم که در ادامه شرح داده می‌شود.

۱.۴ راهکار rule-based

در این بخش از دو ابزار پایتونی به شرح زیر استفاده کردیم:

- **کتاب خانه‌ی :librosa** در یک بخش از این ابزار، صدای ورودی بررسی می‌شود تا هر گونه موزیک بک‌گراندی از آن حذف شود و تنها گفتار باقی بماند. این تسک source separation نیز نام دارد و در کتاب‌خانه‌ی librosa با کیفیت بالایی پیاده‌سازی شده‌است.
- **کتاب‌خانه‌ی :noisereduce** یکی دیگر از کتاب‌خانه‌های پایتون نیز برای حذف نویزهای بی‌معنای بک‌گراند به کار می‌آیند. این کتابخانه noisereduce نام دارم و به عنوان بخش دوم ابزار rule-based ایفای نقش می‌کند.

این ابزار نیز همانند ابزارهای مبتنی بر شبکه‌های عصبی مورد ارزیابی قرار گرفته و در chapter بعد نتایج آن ذکر می‌شود.

۲.۴ راهکار مبتنی بر شبکه‌ی عصبی

فعالیت ما در این بخش به دو فاز تقسیم می‌شود. در فاز اولیه، ما تعداد زیادی از معماری‌های پیشین موجود را برای بررسی کیفیت و عملکر آن‌ها و اینکه آیا قابلیت بهبود دارند یا خیر آزمایش کردیم. در فاز بعدی نیز در جهت بهبود سه مورد از بهترین این معماری‌ها کوشیدیم.

۱.۲.۴ آزمون معماری‌های موجود

معماری‌هایی که در پروپوزال پروژه معرفی شده‌بودند از قرار زیر هستند:

- metricGAN speechbrain

- CleanUnet
- DiffSinger
- FullSubNet
- resemble-enhance
- speech-denoising-wavenet
- DTLN

ما در این فاز از پروژه، سعی کردیم بسیاری از این معماری‌ها و برخی که بعد از پروپوزال به آن‌ها رسیدیم آزمایش کنیم. اول از این جهت که عملکرد آن‌ها روی گفتار فارسی چگونه است و دوم از این جهت که آیا می‌توان آن‌ها را به روش‌هایی چون training و fine-tuning برای زبان فارسی بهبود داد یا خیر. معماری‌های بررسی شده در این بخش شامل MetricGAN نسخه‌ی speechbrain و Met-ricGAN نسخه‌ی مخزن MetricGAN-plus-pytorch و دینویزر DNS-Challenge و معماری موفق resemble-enhance و Clean-Unet و DTLN و speech-denoising-wavenet و denoiser می‌شوند. نوتبوک مربوط به آزمایش این مدل‌ها در مخزن پروژه موجود است. نهایتا در این بخش، سه مخزن زیر علاوه بر داشتن خروجی مناسب، برای آموزش یا fine-tuning نیز مناسب تشخیص داده شدند.

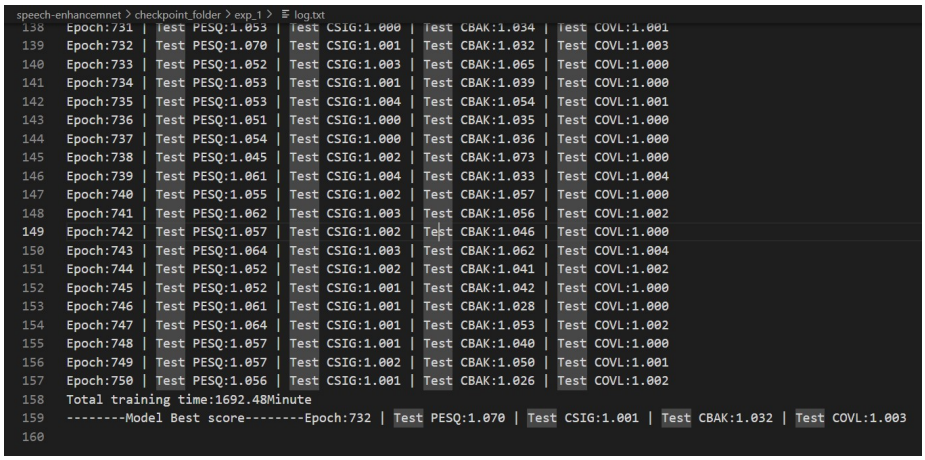
- https://github.com/wooseok-shin/MetricGAN-plus-pytorch
- https://github.com/speechbrain/speechbrain/recipes/Voicebank/enhance/MetricGAN
- https://github.com/resemble-ai/resemble-enhance

۲.۲.۴ بهبود معماری‌های موجود

در این بخش ما سه معماری ذکر شده را train یا fine-tune کردیم که شرح آن در ادامه می‌آید.

MetricGAN-plus-pytorch از آن‌جا که مدل pre-trained ارائه شده توسط این معماری قابلیت fine-tuning را نداشت، ما مدل را از ابتدا روی دیتاست VCTK آموزش دادیم. این مدل به تعداد ۷۵۰ ایپاک و به مدت حدود ۲ روز آموزش دید و مدل نهایی آن به همراه log های آموزش آن در مخزن پروژه بارگزاری گردیده‌است و در تصویر زیر نیز آمده‌است. اما متاسفانه مدل خروجی performance خوبی بر روی دادگان تست نداشت. بنابراین به سراغ مدل MetricGAN ای که در قالب framework پرکاربرد speechbrain ارائه شده بود رفتیم.

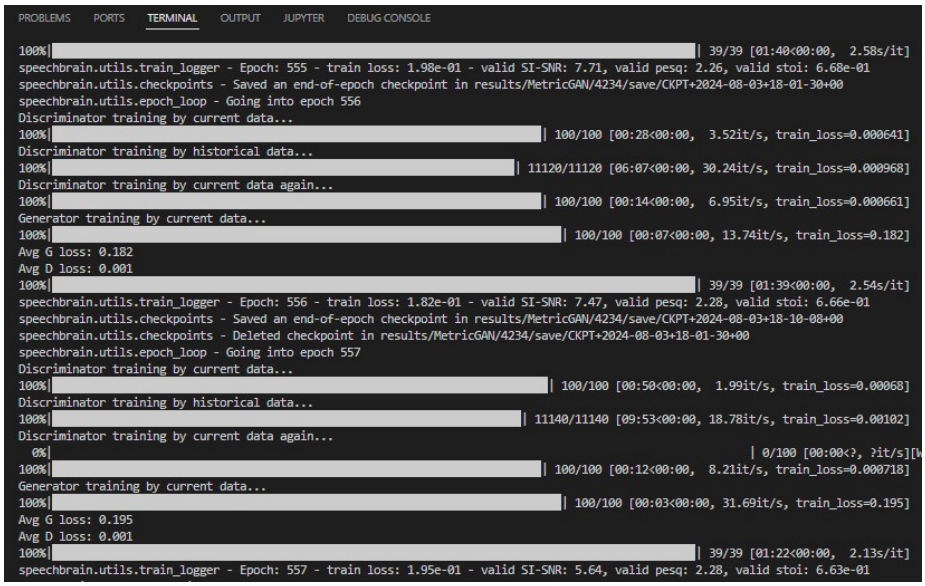
MetricGAN-speechbrain ما این مدل را نیز که نیازمند ورژن دیگری از دیتاست VCTK بود ستاپ کردیم و به مدت حدود ۲ روز و به همان تعداد ایپاک آموزش دادیم. نوتبوک مربوط به ستاپ و اسکرین‌شاتی از مراحل نهایی آموزش آن در مخزن پروژه بارگزاری شده‌است و در تصویر زیر آمده‌است.
این مدل عملکرد بهتری نسبت به مدل قبل داشت اما همچنان performance آن به نسبت مدل pre-trained ای که خودشان روی مخزن پروژه قرار داده بودند inferior بود. بنابراین ما هم در این ورژن را به مدل حاصله ترجیح دادیم و به عوان مدل core در application خود آن را Embed کردیم.
در یک تلاش دیگر، ما این مدل را برای fine-tuning ستاپ کردیم و این بار به کمک داده‌هایی که خودمان record کرده بودیم، به تعداد ایپاک کمی با یک learning-rate کوچک مناسب برای



شکل ۱.۴: training screenshot

fine-tuning آن را آموزش دادیم. خروجی‌های این مدل از دو مورد قبل بهتر بودند و در بخش ارزیابی مورد بررسی قرار می‌گیرند. مدل fine-tune شده در این بخش به همراه script های آموزش آن نیز در مخزن پروژه قرار دارند.

resemble-enhance در انتها با توجه به خروجی‌های بسیار با کیفیت resemble-enhance تصمیم گرفتیم که این مدل را نیز بر روی دادگان فارسی خود آموزش دهیم تا نتیجه را مقایسه کنیم. در مخزن پروژه، یک نوتبوک نیز به این امر اختصاص داده شده و نتایج آن در بخش‌های بعدی خواهد آمد.

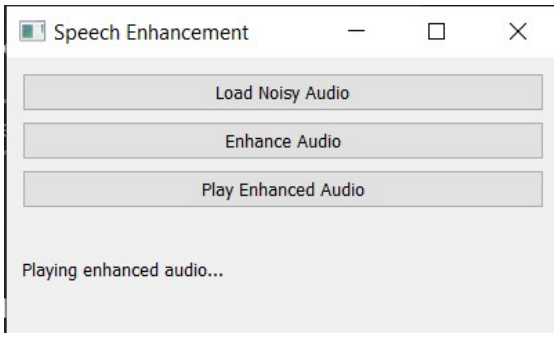


شکل ۲.۴: training screenshot

فصل ۵

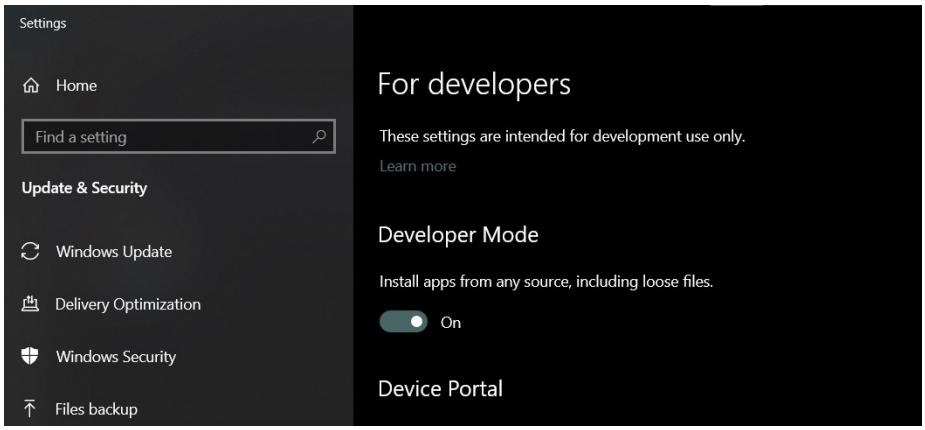
application آزمون مدل منتخب

در بخشی از این پروژه ما یک اپلیکیشن ساده بر بستر PyQt ایجاد کردیم که می‌تواند برای بررسی خروجی مدل منتخب بر روی صداهای نویزی مورد استفاده قرار بگیرد. این اپلیکیشن مینیمال، برای بارگذاری، بهبود و شنیدن صدای بهبود یافته مانند شکل زیر طراحی شده‌است.



شکل ۱.۵: testing application

برای راه اندازی این اپلیکیشن روی device خود کافی است requirement های موجود در requirements.txt را نصب کنید و دستور Py app.py را اجرا کنید. همچنین اگر سیستم شما ویندوز است، لازم است تنظیمات خود را به صورت تصویر زیر تغییر دهید.



شکل ۲.۵: windows app settings

فصل ۶

ارزیابی

در این بخش نتایج ارزیابی هر دو رویکرد rule-based و رویکرد شبکه‌های عصبی شامل شبکه‌های train شده و fine-tune شده را می‌آوریم. معیار ارزیابی مورد استفاده‌ی ما pesq با استفاده از کتابخانه‌ی pesq پایتون می‌باشد. یک مقدار خوب برای این معیار در بازه‌ی $0.45-$ و 4.5 قرار می‌گیرد. برای ارزیابی، ما اصوات نویزی از دیتاست recorded را توسط مدل rule-based و مدل آموزش‌داده‌شده‌ی resemble-enhance و مدل fine-tune شده‌ی MetricGAN بهبود دادیم و معیار pesq را از مقایسه‌ی اصوات تمیز و بهبوددادەشده محاسبه کردیم. اسکریپت‌های این محاسبات نیز در محزن پروژه موجود هستند. نتیجه‌ی ارزیابی مدل‌ها به شرح زیر می‌باشد.

- **rule-based model avg pesq:** 1.3148984869321187
- **trained resemble-enhance model avg pesq:** 1.3394993129703734
- **fine-tuned MetricGAN model avg pesq:** 1.1008675870348195

فصل ۷

نتیجه‌گیری

طبق بررسی‌های انجام‌شده، حتی مدل‌های بهبود گفتار برای زبان انگلیسی نیز دارای خطای بسیاری هستند و صدای اصلی را به خوبی حفظ نمی کنند. در این میان، پرفورمنس مدل‌های فارسی بسیار پایین‌تر نیز هست و تقریبا هیچ پروژه‌ی متن‌باز موفقی در این حوزه وجود ندارد. اما نتایج کسب‌شده نشان می‌دهد که با یک دیتاست به نسبت بزرگ و با داشتن منابع پردازشی کافی می‌توان این مدل ها را بهبود داد و حداقل هم‌تراز با مدل‌های انگلیسی نمود. همچنین این پروژه نشان می‌دهد که در صورت عدم دسترسی به دادگان مناسب، روش‌های rule-based نیز می‌توانند دقتی هم‌رده با مدل‌های عصبی نشان دهند و از این جهت می‌توانند جایگزین خوبی باشند. امید است که نتایج این پروژه و دادگان ارائه شده در آن، راه را برای بهبود مدل‌های بهبود گفتار فارسی هموارتر کند!