



## Speech Enhancement and Denoising

پروژه پایانی درس پردازش گفتار

دانشکده مهندسی کامپیوتر

استاد: دکتر صامتی

بهار ۱۴۰۳

---

بهار بهزادی پور ۴۰۲۲۰۷۱۵۵

مهتا فطرت ۴۰۲۲۱۲۲۲۵

# فهرست مطالب

۲	۱	مقدمه
۲	۱.۱	شرح مسئله
۲	۲.۱	رویکردها
۳	۳.۱	چالش‌ها
۳	۴.۱	اهداف و دستاوردهای این پروژه
۴	۲	مرور کارهای پیشین
۴	۱.۲	Metricgan
۵	۲.۲	Diff-TTS
۶	۳.۲	A Study on Speech Enhancement Based on Diffusion Probabilistic Model
۹	۳	دادگان
۹	۱.۳	دیتاست synthesized
۹	۲.۳	دیتاست recorded

# فصل ۱

## مقدمه

بهبود کیفیت گفتار و رفع نویز آن همواره مورد توجه بوده است و جزء معدود مسائل حوزه‌ی پردازش گفتار است که می‌توان گفت هنوز کاملاً حل شده 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 به 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 سیستم‌های تشخیص گفتار، ارتباطات تلفنی، و کمک به افراد با مشکلات شنوایی اهمیت دارد. در زبان فارسی، به دلیل تفاوت‌های صوتی و ساختاری با دیگر زبان‌ها، نیاز به تحقیق و توسعه ویژه‌ای 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 عموماً 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 راه‌حلی‌هایی بر اساس هر یک ارائه می‌دهیم.

### ۳.۱ چالش‌ها

همانطور که برای سایر راه‌حل‌های مبتنی بر شبکه‌های عصبی چالش داده مطرح است، در این جا هم با این مشکل مواجه هستیم. در واقع تهیه‌ی دادگانی طبیعی متشکل از زوج‌های تمیز و نویزی یک گفتار واحد، امری دشوار و زمان‌بر است. به همین جهت غالباً 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 ۳۰٪ افزایش دهد و به نتایج برتر دست یابد (امتیاز PESQ = ۱۵.۳). 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 سیگموئید قابل یادگیری برای تخمین ماسک

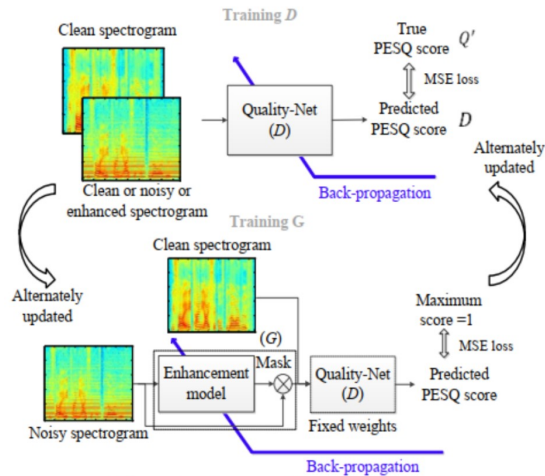


Figure 1: Training flow of MetricGAN.

## 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 سیگموئید قابل یادگیری، مفیدترین تکنیک‌ها هستند. ما MetricGAN+ ما نتایج پیشرفته‌ای را بر روی مجموعه داده‌های VoiceBank DEMAND به دست می‌آورد و امتیازهای PESQ می‌تواند به ترتیب ۳.۰ و ۴۵.۰ نسبت به MetricGAN و BLSTM (MSE) افزایش یابد.

## ۲.۲ Diff-TTS

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 اتورگرسو 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. به منظور 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 NVIDIA GPU ۲۰۸۰Ti به‌طور ۲۸ برابر سریع‌تر از زمان واقعی تولید می‌کند. Diff-TTS توزیع نویز را به توزیع مل-اسپکتروگرام متناظر با متن داده شده تبدیل می‌کند. همان‌طور که در شکل ۱ نشان داده شده است، مل-اسپکتروگرام به تدریج با نویز گوسی تخریب شده و به متغیرهای نهان تبدیل می‌شود. این فرآیند، فرآیند انتشار نامیده می‌شود.

فرض کنید،  $x_1, \dots, x_T$  یک دنباله از متغیرها با ابعاد یکسان باشد که در آن  $t = 0, 1, \dots$ .  $T$ ، شاخصی برای مراحل زمانی انتشار است. سپس، فرآیند انتشار مل-اسپکتروگرام  $x_0$  را از طریق

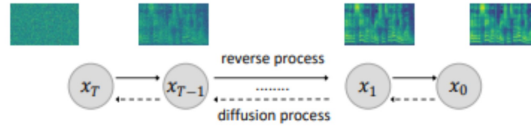


Figure 1: Graphical model for the reverse process and the diffusion process.

شکل ۲.۲: فرایند diffusion

یک زنجیره انتقال‌های مارکوفی به نویز گوسی  $x_T$  تبدیل می‌کند. هر مرحله انتقال با یک برنامه واریانس  $x_1, \dots, x_T$  از پیش تعیین شده است. به طور خاص، هر تبدیل مطابق با احتمال انتقال مارکوفی  $q(x_t|x_{t-1}, c)$  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 فرایند انتشار  $q(x_1:T|x_0, c)$  یک فرایند مارکوفی است و می‌تواند به صورت زیر تجزیه شود:

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

فرایند معکوس یک روش تولید مل-اسپکتروگرام است که دقیقاً برخلاف فرایند انتشار عمل می‌کند. برخلاف فرایند انتشار، هدف فرایند معکوس بازسازی یک مل-اسپکتروگرام از نویز گوسی است. فرایند معکوس به عنوان توزیع شرطی  $p(x_0:T-1|x_T, c)$  تعریف می‌شود و می‌تواند بر اساس خاصیت زنجیره مارکوف به چندین انتقال تجزیه شود: دیف-TTS شامل یک رمزگذار متن، رمزگذار مرحله، پیش‌بینی‌کننده مدت زمان، و رمزگشا است. شبکه رمزگشا شامل یک پشته از ۱۲ بلوک مقاوم با  $\text{Conv1D}$ ،  $\text{sigmoid}$   $\text{tanh}$  و کانولوشن‌های  $1 \times 1$  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  $\text{Conv1D}$  اضافه می‌شود. لایه  $\text{Conv1D}$  اندازه هسته‌ای برابر ۳ بدون گشادگی دارد. پس از عبور از این بلوک مقاوم، خروجی‌ها قبل از پس-نت جمع می‌شوند. در نهایت، رمزگشا نویز گوسی متناظر با دنباله فونم و مرحله زمانی انتشار را به دست می‌آورد.

## 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 فرایندهای جفت‌شده انتشار و معکوس نشان داده‌اند. ویژگی منحصر به فرد فرایند معکوس (یعنی حذف سیگنال‌های غیرهدف از نویز گوسی و سیگنال‌های نویزی) می‌تواند برای بازسازی سیگنال‌های تمیز استفاده شود. بر اساس این ویژگی، ما مدل بهبود گفتار مبتنی بر مدل احتمالاتی انتشار (DiffuSE)

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

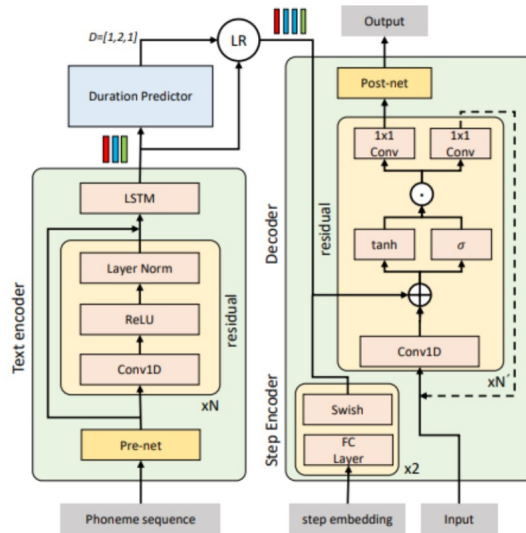


Figure 3: The network architecture of Diff-TTS

### شکل ۳.۲: معماری diff-tts

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  $x_t$  ممکن است در طول فرآیند معکوس از مرحله  $T$  تا  $t+1$  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.

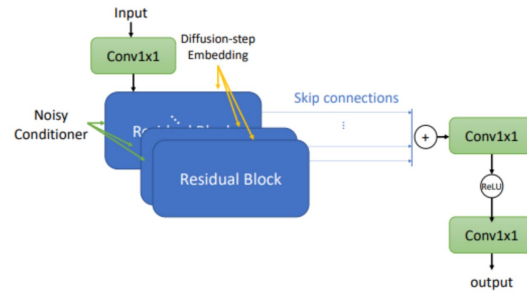


Fig. 2. The architecture of the proposed DiffuSE model

#### 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 *fine-tuning* بر روی زبان فارسی انجام شود. طی جستجوهای که انجام شد، دیتاست بزرگ و شناخته شدهای برای تسک *Denoising and Enhancement Speech* زبان 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 نهایتاً 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 *crowd* از نوع *gaussian* ترکیب می‌شود و با احتمال یک‌دوم هم با یک صوت صدای *crowd* 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 سنتز شده می‌باشد 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 دقیقاً همین اصوات را در محیط‌هایی با یک صدای *background* پخش کردیم و مجدداً ضبط کردیم. به این ترتیب هر دوی داده‌ی تمیز و نویزی شامل گفتار عیناً یکسانی می‌شدند اما دیگر صدای *background* به صورت مصنوعی به داده اضافه نشده بود. ایده‌ی اصلی این بود که تمام صداهای محیطی نیز کاملاً 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 مورد 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 شده است و برای fine-tuning مورد استفاده قرار گرفته است.