

http://lib.uliege.be https://matheo.uliege.be

**پایان‌نامه کارشناسی ارشد:** شبیه‌سازی خودکار صدای چند بلندگو

**نویسنده:** Jemine, Corentin

**ترویج دهنده:** Louppe, Gilles

**دانشکده:** دانشکده علمی کاربردی

**مدرک تحصیلی:** فوق‌لیسانس علوم داده با گرایش تخصصی

**سال تحصیلی انجام پروژه:** 2019-2018

URI/URL: <https://github.com/CorentinJ/Real-Time-Voice-Cloning> ; <http://hdl.handle.net/2268.2/6801>

**هشدار برای کاربران:**

کلیه اسنادی که در دسترسی آزاد در سایت MatheO قرار می‌گیرند توسط حق چاپ محافظت می‌شوند.   
مطابق با اصول تعیین شده توسط " ابتکار دسترسی آزاد بوداپست " (BOAI، 2002)، کاربر سایت می‌تواند متن کامل این اسناد را بخواند، دانلود، کپی، ارسال، چاپ، جستجو کند یا به متن کامل این اسناد پیوند دهد، آنها را کالبدشکافی کند. برای فهرست کردن آنها، از آنها به‌عنوان داده برای نرم‌افزار استفاده کنید، یا از آنها برای هر هدف قانونی دیگر (پیش‌بینی‌شده توسط مقررات) استفاده کنید.

هرگونه استفاده از سند برای مقاصد تجاری اکیداً ممنوع است. (مربوط به کپی‌رایت).

علاوه بر این، کاربر موافقت می‌کند که به حقوق معنوی نویسنده، عمدتاً حق یکپارچگی اثر و حق پدری و این در هر استفاده‌ای که کاربر متعهد می‌شود، احترام بگذارد؛ بنابراین، به‌عنوان‌مثال، هنگام بازتولید یک سند به‌صورت استخراج یا به طور کامل، کاربر به طور کامل منابع را همان‌طور که در بالا ذکر شد ذکر می‌کند.

هرگونه استفاده‌ای که به‌صراحت در بالا مجاز نیست (مانند، به‌عنوان‌مثال، تغییر در سند یا خلاصه آن) به مجوز قبلی و صریح نویسندگان یا جانشینان آنها در عنوان نیاز دارد.

دانشگاه لیژ

دانشکده علمی کاربردی

شبیه‌سازی صدا در زمان واقعی

**نویسنده: سرپرست:**

Corentin Jemine Prof. Gilles Louppe

سال تحصیلی 2018 – 2019



انجام مطالعات فارغ‌التحصیلی برای اخذ مدرک کارشناسی ارشد

در علم داده نوشته Corentin Jemine

**خلاصه**

پیشرفت‌های اخیر در یادگیری عمیق نتایج چشمگیری را در حوزه متن به گفتار نشان داده است.   
برای این منظور، یک شبکه عصبی عمیق معمولاً با استفاده از مجموعه‌ای از چندین ساعت گفتار ضبط شده حرفه‌ای از یک گوینده آموزش داده می‌شود.

ارائه یک‌صدای جدید به چنین مدلی بسیار گران است، زیرا نیاز به ضبط مجموعه‌داده جدید و آموزش مجدد مدل دارد.

یک تحقیق اخیر خط لوله سه‌مرحله‌ای را معرفی کرد که اجازه می‌دهد صدایی را که در طول آموزش مشاهده نمی‌شود تنها از چند ثانیه گفتار مرجع و بدون آموزش مجدد مدل شبیه‌سازی شود. نویسندگان نتایج به طور قابل‌توجهی صدای طبیعی را به اشتراک می‌گذارند، اما هیچ اجرایی ارائه نمی‌دهند.

ما این چارچوب و منبع‌باز را در اولین اجرای عمومی آن بازتولید می‌کنیم. ما چارچوب را با یک مدل Vocoder جدیدتر تطبیق می‌دهیم تا بتوانیم آن را در زمان واقعی اجرا کنیم.

فهرست

1. **مقدمه**
2. مروری بر روش‌های تبدیل متن به گفتار در یادگیری ماشینی
3. سنتز گفتار پارامتریک آماری
4. تکامل وضعیت هنر در تبدیل متن به گفتار
5. **انتقال یادگیری از تأیید بلندگو به ترکیب متن به گفتار چند سخنران**
6. بررسی اجمالی
7. تعریف مسئله
8. رمزگذار بلندگو
9. معماری مدل
10. ازدست‌دادن سرتاسری تعمیم‌یافته
11. آزمایشات
12. سینت سایزر
13. معماری مدل
14. آزمایشات
15. Vocoder
16. معماری مدل
17. آزمایشات
18. **جعبه ابزار و منبع‌باز**
19. **نتیجه‌گیری**
20. **مقدمه**

مدل‌های یادگیری عمیق در بسیاری از زمینه‌های یادگیری ماشین کاربردی غالب شده‌اند. متن به گفتار (Text-to-speech (TTS))، فرایند ترکیب گفتار مصنوعی از یک پیام متنی، از این قاعده مستثنی نیست. مدل‌های عمیقی که صدای طبیعی‌تری نسبت به رویکردهای مرسوم سنتی تولید می‌کنند، در سال 2016 ظاهر شدند. از آن زمان، بیشتر تمرکز تحقیقات بر روی کارآمدتر کردن این مدل‌های عمیق، صدای طبیعی‌تر یا آموزش آن‌ها به‌صورت سرتاسری متمرکز شده است.

استنباط از صدها بار کندتر از زمان واقعی در GPU (Graphics Processing Unit به معنای واحد پردازش گرافیکی) به‌دست‌آمده است (van den Oord et al-ون دن اوورد و همکاران -2016) به‌صورت بلادرنگ (سیستم‌عامل بلادرنگ یا Real Time Operating System (RTOS)) بر روی یک CPU (Central Processing Unite به معنای واحد پردازش مرکزی) موبایل امکان‌پذیر است. (Kalchbrenner et al-کالچبرنر و همکاران-2018) در مورد کیفیت گفتار تولید شده، طبیعی بودن نزدیک به انسان را نشان می‌دهد. (Shen et al- شن و همکاران-2017)

جالب‌توجه است که طبیعی بودن گفتار با معیارهای ذهنی به بهترین وجه رتبه‌بندی می‌شود و مقایسه با گفتار واقعی انسان به این نتیجه می‌رسد که ممکن است چیزی به نام «گفتار طبیعی‌تر از گفتار انسان» وجود داشته باشد. در واقع، برخی معتقدند که آستانه طبیعی بودن انسان قبلاً رد شده است. (Shirali-Shahreza and Penn- سجاد شیرعلی شهرضا و پن -2018)

مجموعه‌داده‌های گفتار ضبط شده حرفه‌ای منبع کمیاب است. سنتز یک‌صدای طبیعی با تلفظ صحیح، لحن پرجنب‌وجوش و حداقل نویز پس‌زمینه نیاز به داده‌های آموزشی باکیفیت‌های یکسان دارد. علاوه بر این، کارایی داده‌ها موضوع اصلی یادگیری عمیق است. آموزش یک مدل رایج تبدیل متن به گفتار مانند تاکوترون (سیستم سنتز گفتار متکی بر هوش مصنوعی) معمولاً به صدها ساعت گفتار نیاز دارد. (Wang et al \_وانگ و همکاران- 2017). بااین‌حال، توانایی تولید گفتار با هر صدایی برای طیف وسیعی از برنامه‌ها جذاب است، خواه مفید باشند یا صرفاً سفارشی‌سازی شده باشند. تحقیقات منجر به ایجاد چارچوب‌هایی برای تبدیل صدا و شبیه‌سازی صدا شده است. تفاوت آنها در این است که تبدیل صدا شکلی از انتقال سبک در یک بخش گفتار از یک‌صدا به دیگری است، درحالی‌که شبیه‌سازی صدا شامل ضبط صدای یک گوینده برای اجرای گفتار متنی در ورودی‌های دلخواه است.

درحالی‌که آموزش کامل یک مدل TTS تک بلندگو از نظر فنی نوعی شبیه‌سازی صدا است، اما علاقه بیشتر به ایجاد یک مدل ثابت است که بتواند صداهای جدیدتر را با داده‌های کمی ترکیب کند.

رویکرد متداول این است که یک مدل TTS آموزش‌دیده برای تعمیم به سخنرانان جدید بر اساس تعبیه صدا برای شبیه‌سازی شرطی شود (Arik et al 2017-2018 / Jia et al-2018). تعبیه ابعاد پایینی دارد و توسط یک مدل رمزگذار بلندگو که گفتار مرجع را به‌عنوان ورودی می‌گیرد، مشتق شده است. این رویکرد معمولاً کارآمدتر از آموزش یک مدل TTS جداگانه برای هر بلندگو است، علاوه بر این که مرتبه‌ای سریع‌تر و از نظر محاسباتی هزینه کمتری دارد. جالب‌توجه است، اختلاف زیادی بین مدت‌زمان گفتار مرجع موردنیاز برای شبیه‌سازی یک‌صدا در بین روش‌های مختلف وجود دارد که از نیم ساعت برای هر گوینده تا تنها چند ثانیه متغیر است. این عامل معمولاً تعیین‌کننده شباهت صدای تولید شده با صدای واقعی گوینده است.

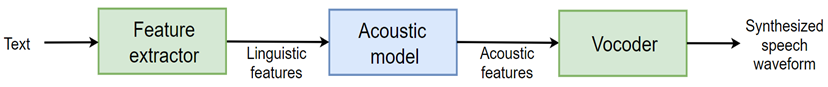
هدف ما دستیابی به شکل قدرتمندی از شبیه‌سازی صدا است. چارچوب به‌دست‌آمده باید بتواند در حالت صفر شات، یعنی برای بلندگوهایی که در طول آموزش‌دیده نمی‌شوند، کار کند. این باید صدای یک گوینده را تنها با چند ثانیه گفتار مرجع در خود داشته باشد. این نتایج موردنظر نشان‌داده‌شده است که توسط برآورده می‌شود. (Jia et al-2018)

نتایج آنها چشمگیر است، اما توسط هیچ اجرای عمومی پشتیبانی نمی‌شود. ما چارچوب آنها را بازتولید می‌کنیم و پیاده‌سازی خود را به‌صورت متن‌باز می‌کنیم. علاوه بر این، ما یک مدل مبتنی بر (Kalchbrenner et al -2018) چارچوب را ادغام می‌کنیم تا آن را در زمان واقعی اجرا کنیم، یعنی برای تولید گفتار در زمانی کوتاه‌تر یا برابر با مدت‌زمان گفتار تولید شده.

ساختار این سند به شرح زیر است. ما با مقدمه‌ای کوتاه در مورد روش‌های TTS با یادگیری ماشینی شروع می‌کنیم. مروری بر تکامل وضعیت هنر برای TTS با طبیعی بودن گفتار به‌عنوان معیار اصلی دنبال می‌کند. سپس کار (جیا و همکاران، 2018) را همراه با اجرای خود ارائه می‌کنیم. ما با ارائه جعبه ابزاری که برای رابط با چارچوب طراحی کرده‌ایم، به پایان می‌رسیم.

1. **مروری بر روش‌های تبدیل متن به گفتار در یادگیری ماشینی**
2. **سنتز گفتار پارامتریک آماری:**

سنتز گفتار پارامتریک آماری (SPSS) به گروهی از روش‌های TTS مبتنی بر داده اشاره دارد که در اواخر دهه 90 ظهور کردند. در SPSS، رابطه بین ویژگی‌های محاسبه شده بر روی متن ورودی و ویژگی‌های صوتی خروجی توسط یک مدل مولد آماری (موسوم به مدل آکوستیک) یاد می‌شود؛ بنابراین، یک چارچوب کامل SPSS همچنین شامل یک خط لوله برای استخراج ویژگی‌ها از متن برای سنتز، و همچنین سیستمی است که می‌تواند شکل موج صوتی را از ویژگی‌های صوتی تولید شده توسط مدل صوتی بازسازی کند (به چنین سیستمی Vocoder می‌گویند). برخلاف مدل آکوستیک، این دو بخش از چارچوب ممکن است به طور کامل مهندسی شده و از هیچ روش آماری استفاده نکنند. درحالی‌که مدل‌های مدرن عمیق TTS معمولاً به‌عنوان SPSS نامیده نمی‌شوند، خط لوله SPSS همان‌طور که در شکل 1 نشان‌داده‌شده است به همان اندازه برای آن روش‌های جدیدتر اعمال می‌شود.



**شکل 1- خط لوله کلیSPSS. جعبه آبی صرفاً یک مدل آماری است در حالی که جعبه‌های سبز می‌توانند فرآیندهای مهندسی یا/و مدل‌های آماری باشند.**

نقش استخراج‌کننده ویژگی این است که داده‌هایی را ارائه دهد که بیشتر نشان‌دهنده این باشد که گفتار تولید شده توسط مدل چگونه به نظر می‌رسد. گفتار فرایند پیچیده‌ای است و تغذیه مستقیم شخصیت‌ها به یک مدل آکوستیک ضعیف ثابت خواهد شد که مؤثر نیست. ارائه ویژگی‌های اضافی از تکنیک‌های پردازش زبان طبیعی (NLP) ممکن است تا حد زیادی وسعت کاری را که باید توسط مدل آکوستیک آموخته شود، کاهش دهد. بااین‌حال ممکن است در مورد طبیعی بودن، به‌خصوص برای کلمات نادر یا ناشناخته، به مبادلات منجر شود. در واقع، اکتشافی مهندسی دستی به طور کامل همه پیچیدگی‌های زبان گفتاری را مشخص نمی‌کند. به همین دلیل استخراج ویژگی را می‌توان با مدل‌های آموزش‌دیده نیز انجام داد. خط بین استخراج‌کننده ویژگی و مدل صوتی می‌تواند تار شود، به‌خصوص برای مدل‌های عمیق. در واقع، گرایشی که در همه حوزه‌هایی که مدل‌های عمیق بر تکنیک‌های یادگیری ماشین سنتی غلبه کرده‌اند، رایج است، این است که استخراج ویژگی از اکتشافات کمتری تشکیل شده باشد، زیرا مدل‌های بسیار غیرخطی قادر به کار در سطوح بالاتر انتزاع هستند.

یک تکنیک رایج استخراج ویژگی، ساخت فریم‌هایی است که زمینه‌های اطراف را به‌صورت سلسله‌مراتبی ادغام می‌کند. به‌عنوان‌مثال، یک قاب در سطح هجا می‌تواند شامل کلمه‌ای باشد که آن را تشکیل می‌دهد، موقعیت آن در کلمه، هجاهای همسایه، واج‌هایی که هجا را تشکیل می‌دهند، ... استرس و لهجه تک‌تک هجاها را می‌توان با یک مدل آماری مانند درخت تصمیم پیش‌بینی کرد. برای رمزگذاری عروض می‌توان از مجموعه‌ای از قوانین مانند ToBI (بکمن و ایلام، 1997) استفاده کرد. در نهایت، یک کار مهندسی ویژگی برای ارائه یک قاب به‌عنوان یک شی عددی به مدل باقی می‌ماند، به‌عنوان‌مثال. ویژگی‌های طبقه‌بندی معمولاً با استفاده از یک نمایش یک‌سره کدگذاری می‌شوند.

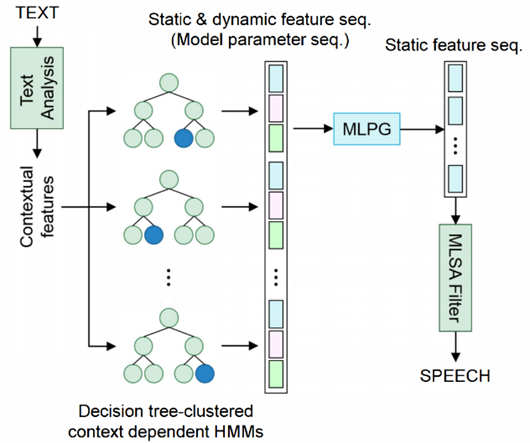
دلیل اینکه مدل آکوستیک مستقیماً یک‌شکل موج صوتی را پیش‌بینی نمی‌کند این است که مدل‌سازی صدا به طور اتفاقی دشوار است: این یک حوزه به‌ویژه متراکم است و سیگنال‌های صوتی معمولاً بسیار غیرخطی هستند. نمایشی که ویژگی‌ها را به شیوه‌ای قابل‌حمل‌تر نشان می‌دهد، حوزه زمان - فرکانس است. طیف‌نگارها نسبت به همتای شکل موج خود صاف‌تر و چگالی کمتری دارند. آنها همچنین دارای مزیت دوبعدی بودن هستند، بنابراین به مدل‌ها اجازه می‌دهد تا از اتصال فضایی بهتری استفاده کنند. متأسفانه، یک طیف‌نگار نمایشی با اتلاف شکل موج است که فاز را دور می‌زند. هیچ تابع تبدیل معکوس منحصربه‌فردی وجود ندارد، و استخراج تابعی که نتایجی با صدای طبیعی تولید می‌کند، بی‌اهمیت نیست. هنگامی که به گفتار اشاره می‌شود، این عملکرد مولد را رمزگذار صدا می‌نامند. انتخاب Vocoder فاکتور مهمی در تعیین کیفیت صدای تولید شده است.

همان‌طور که اغلب در مورد وظایفی که شامل تولید داده‌های ادراکی مانند تصاویر یا صدا هستند، ارزیابی رسمی و عینی عملکرد مدل دشوار است. در مورد ما، ما به ارزیابی طبیعی بودن گفتار و شباهت صدای صوتی تولید شده می‌پردازیم. روش‌های قدیمی‌تر TTS اغلب بر روی آمار محاسبه‌شده بر روی شکل موج یا ویژگی‌های صوتی برای مقایسه مدل‌های مختلف متکی بودند. نه‌تنها این معیارها اغلب با ادراک انسان از صدا همبستگی ضعیفی دارند، بلکه معمولاً همان چیزی هستند که مدل قصد داشت حداقل آن را کاهش دهد. "هنگامی که یک معیار تبدیل به یک هدف می‌شود، دیگر معیار خوبی نیست" (Strathern، 1997). یک گرایش اخیر در TTS انجام یک ارزیابی ذهنی با افراد انسانی و گزارش میانگین امتیاز نظر آنها (MOS) است. با یک سری بخش‌های صوتی ارائه می‌شوند و از آنها خواسته می‌شود که طبیعی بودن (شباهت آن‌ها هنگام مقایسه دو بخش) را در مقیاس لیکرت از 1 تا 5 ارزیابی کنند. ازآنجایی‌که آزمودنی‌ها لزوماً به گفتار واقعی انسان با 5 نمره نمی‌دهند، ممکن است در آینده سیستم‌های TTS از انسان‌ها در این معیار پیشی بگیرند.

(شیرعلی شهرضا و پن، 2018) استدلال می‌کنند که MOS معیاری نیست که برای ارزیابی سیستم‌های TTS اقتباس شده باشد و به‌جای آن از تست A/B استفاده می‌کنند.جایی که از آزمودنی‌ها خواسته می‌شود بگویند که کدام بخش صوتی را از بین دو مورد ترجیح می‌دهند (رأی خنثی نیز معمولاً ممکن است). برای توضیح اختلاف‌نظر در بین افراد، باید این مطالعات را در مقیاس بزرگ با صدها موضوع انجام داد. خدمات جمع‌سپاری مانند Amazon Mechanical Turk3 معمولاً درگیر هستند.

1. **تکامل وضعیت هنر در تبدیل متن به گفتار:**

وضعیت هنر در SPSS برای مدت طولانی به‌عنوان یک چارچوب مبتنی بر مدل مارکوف پنهان (HMM) باقی‌مانده است (توکودا - 2013). این رویکرد که در شکل 2 نشان‌داده‌شده است، شامل خوشه‌بندی ویژگی‌های زبانی استخراج‌شده از متن ورودی با یک درخت تصمیم، و آموزش HMM در هر خوشه است (Yoshimura-1999). HMMها وظیفه دارند توزیعی بر روی ضرایب طیف گرام، مشتق آنها، مشتق دوم و یک پرچم دودویی تولید کنند که نشان می‌دهد کدام قسمت‌های صوتی تولید شده باید حاوی صدا باشد. با الگوریتم تولید پارامتر حداکثر احتمال (MLPG) (توکودا و همکاران، 2000)، ضرایب طیف‌نگاری از این توزیع نمونه‌برداری می‌شوند و در نهایت به رمزگذار صوتی MLSA داده می‌شوند (Imai، 1983). می‌توان صدای تولید شده با شرطی کردن HMMها روی یک بلندگو یا تنظیم پارامترهای گفتاری تولید شده با تکنیک‌های تطبیق یا درون‌یابی را تغییر داد (Yoshimura و همکاران، 1997). توجه داشته باشید که درحالی‌که این چارچوب درگذشته برای SPSS پیشرفته بود، اما هنوز از نظر طبیعی بودن گفتار تولید شده در مقایسه با رویکردهای پیوسته به‌خوبی تثبیت شده، پایین‌تر بود.



**شکل 2- خط لوله عمومی TTS مبتنی برHMM شکل استخراج شده از (Hashimoto et al.- 2015)**

|  |  |
| --- | --- |
| Method | MOS |
| HMM+MLPG  HMM+DNN  DNN+MLPG  DNN+DNN | 3.08 (±0.12) 2.86 (±0.12) 3.53 (±0.12) 3.17 (±0.12) |

جدول 1: MOS روش‌های مختلف بررسی شده در (Hashimoto et al.- 2015). خط اول چارچوب مبتنی بر HMM است. برای خط دوم و چهارم، الگوریتم MLPG با یک شبکه عصبی کاملاً متصل جایگزین می‌شود.

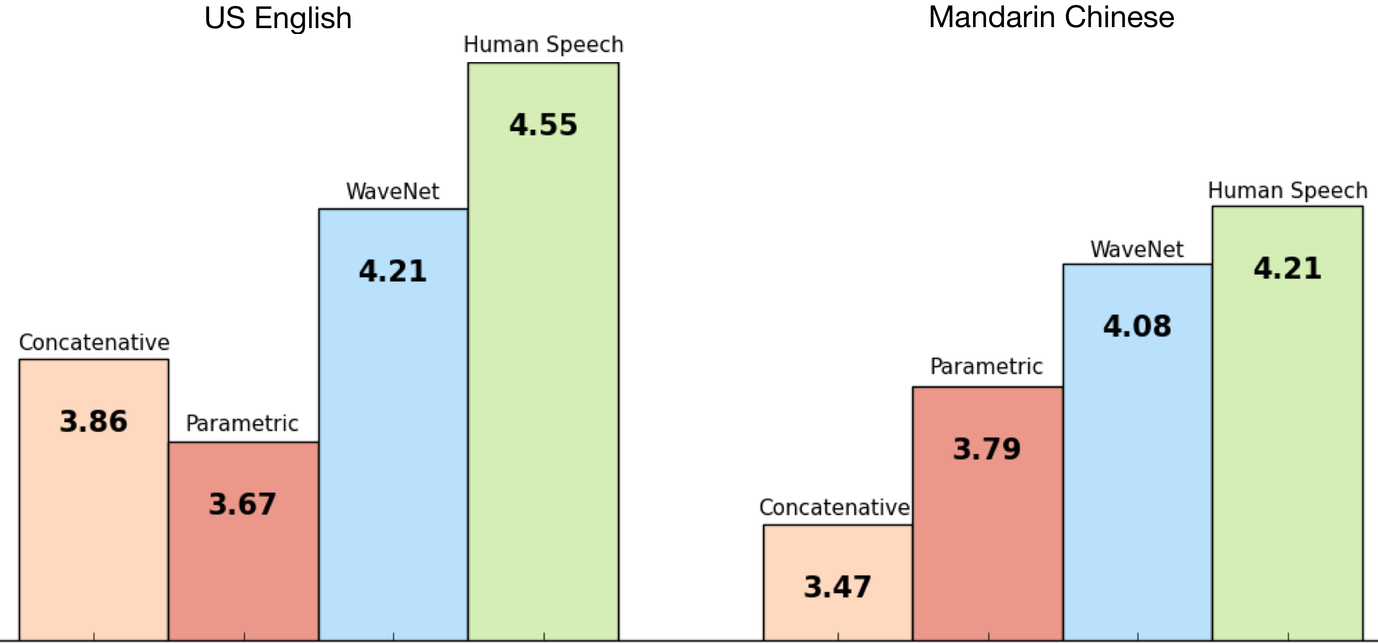
بهبودهایی در این چارچوب بعداً توسط شبکه‌های عصبی عمیق پیش‌خور (DNN) در نتیجه پیشرفت در سخت‌افزار و نرم‌افزار انجام شد. ذن و همکاران (2013) پیشنهاد می‌کنند که HMMهای خوشه‌بندی درخت تصمیم را به نفع یک DNN جایگزین کنند. آنها برای بهره‌وری بهتر داده‌ها استدلال می‌کنند؛ زیرا مجموعه آموزشی دیگر در خوشه‌های مختلف زمینه‌ها تقسیم نمی‌شود. آنها بهبودهایی را نسبت به کیفیت گفتار با تعدادی از پارامترهای مشابه با رویکرد مبتنی بر HMM نشان می‌دهند. تحقیقات بعدی این یافته‌ها را تأیید می‌کند (کیان و همکاران، 2014؛ هاشیموتو و همکاران، 2015). MOS ترکیب‌های مدل مختلف که توسط (Hashimoto و همکاران، 2015) امتحان شده است در جدول 1 گزارش شده‌اند.

(Fan et al.- 2014) پشتیبانی می‌کنند که RNNها مدل‌های آکوستیک طبیعی می‌سازند؛ زیرا می‌توانند نمایش فشرده‌ای از توابع پیچیده و طولانی را یاد بگیرند. ازآنجایی‌که RNN ‌ها برای تولید سری‌های موقتی سازگار هستند، ویژگی‌های استاتیکی را می‌توان مستقیماً توسط مدل آکوستیک تعیین کرد و نیاز به ویژگی‌های دینامیکی و MLPG را کاهش داد. آنها شبکه‌های LSTMهای دوطرفه را در برابر رویکردهای مبتنی بر HMM و DNN که قبلاً توضیح داده شد، مقایسه می‌کنند. نتایج آزمایش A/B آنها قطعی است، ما آنها را در شکل 3 گزارش می‌کنیم.



**شکل 3- آزمایش A/B مدل ها. Hybrid B یک شبکه با لایه‌های LSTM کاملاً متصل و دو طرفه است. شکل استخراج شده از (فان و همکاران، 2014)**

ظهور WaveNet (وان دن اوورد و همکاران، 2016) پیشرفت قابل‌توجهی در TTS ایجاد کرد. WaveNet یک شبکه عصبی کانولوشن عمیق است که برای یک‌شکل موج صوتی خام، توزیع یک نمونه را به‌صورت مشروط به نمونه‌های قبلی مدل می‌کند؛ بنابراین می‌توان مستقیماً با پیش‌بینی نمونه‌ها در یک‌زمان خود به‌صورت خودکار، صدا تولید کرد. WaveNet از پشته‌های پیچش گشاد شده یک‌بعدی با ضریب اتساع به طور تصاعدی با عمق لایه افزایش می‌یابد که امکان میدان دریافت بسیار بزرگ و غیرخطی بودن قوی موردنیاز برای مدل‌سازی صدای خام را فراهم می‌کند. شرطی کردن مدل بر روی ویژگی‌های زبانی برای انجام TTS لازم است؛ بنابراین WaveNet هم به‌عنوان یک مدل آکوستیک و هم به‌عنوان یک رمزگذار صدا عمل می‌کند. توجه داشته باشید که بدون شرطی‌سازی محلی، یک WaveNet آموزش‌دیده، صدایی مشابه داده‌های آموزشی، اما بدون ساختار یا معنایی (در اصل حرف‌زدن) تولید می‌کند. نویسندگان WaveNet را با رویکرد پارامتری قدیمی‌تر و با رویکرد پیوسته مقایسه می‌کنند، نتایج در شکل 4 گزارش شده است. توجه داشته باشید که چگونه نتایج بین انگلیسی ایالات متحده و چینی ماندارین متفاوت است که نشان می‌دهد عملکرد TTS به زبان آگنوستیک نیست.

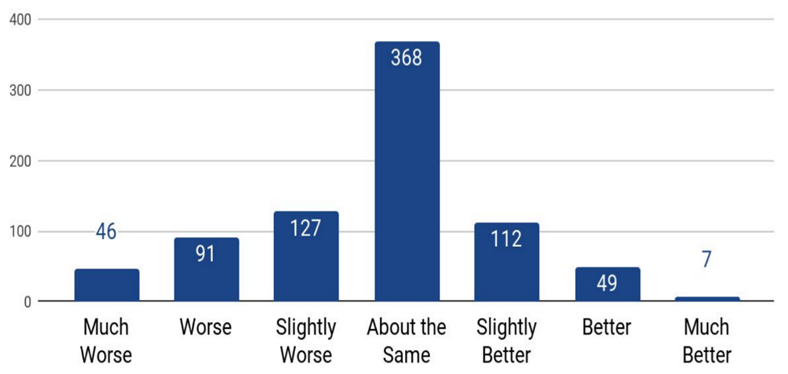


**شکل 4- MOS عملکرد WaveNet در مقایسه با رویکرد پارامتری و پیوسته و همچنین با گفتار طبیعی.**

از تاکوترون (وانگ و همکاران، 2017) پیروی می‌کند، یک مدل دنباله به دنباله که یک طیف‌نگاری را از دنباله‌ای از کاراکترها به‌تنهایی تولید می‌کند، و نیاز به تخصص حوزه را کاهش می‌دهد. در این چارچوب، Vocoder الگوریتم Griffin-Lim است. Tacotron از یک معماری رمزگذار- رمزگشا استفاده می‌کند که در هر مرحله رمزگشا بر روی مجموع وزنی خروجی‌های رمزگذار عمل می‌کند. این مکانیسم توجه که در (Bahdanau و همکاران، 2014) توضیح داده شده است، به شبکه اجازه می‌دهد تصمیم بگیرد که کدام مراحل از دنباله ورودی باتوجه‌به هر مرحله از دنباله خروجی مهم هستند.

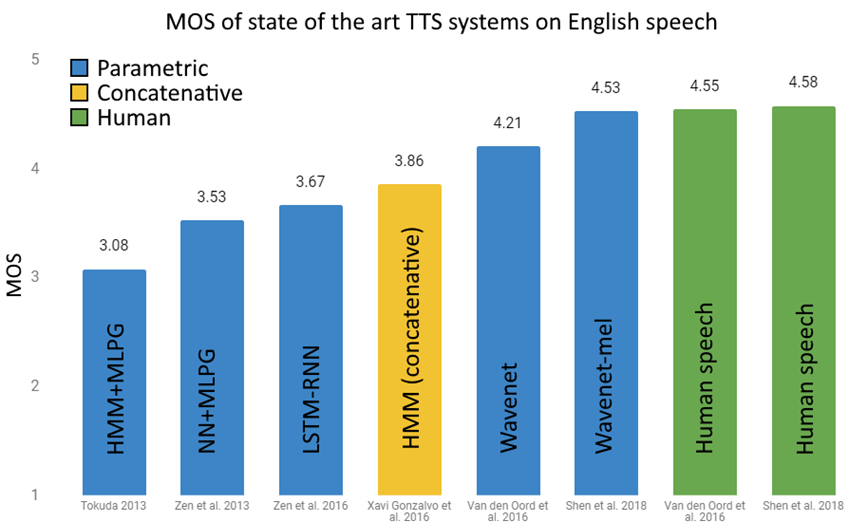
Tacotron به MOS 3.85 در مجموعه‌داده انگلیسی ایالات متحده دست می‌یابد که بیشتر از امتیاز 3.69 به‌دست‌آمده در رویکرد پارامتری (Zen et al.- 2016) است؛ اما کمتر از 4.09 به‌دست‌آمده توسط رویکرد پیوسته (Gonzalvo et al. ، 2016). نویسندگان ذکر می‌کنند که تاکوترون صرفاً گامی به‌سوی یک چارچوب بهتر است. متعاقباً، تاکوترون 2 منتشر شد (شن و همکاران، 2017). معماری Tacotron 2 مانند یک رمزگذار-رمزگشا با توجه باقی می‌ماند، اگرچه تغییرات متعددی در نوع لایه‌ها ایجاد شده است. تفاوت اصلی با Tacotron استفاده از WaveNet اصلاح شده به‌عنوان Vocoder است.

در همان مجموعه‌داده، Tacotron 2 به 4.53 MOS دست می‌یابد که در مقایسه با 4.58 برای گفتار انسان (تفاوت از نظر آماری معنی‌دار نیست)، و به بالاترین MOS برای TTS دست می‌یابد. با آزمایش A/B، مشخص شد که Tacotron 2 به طور متوسط کمی کمتر از نمونه‌های حقیقت زمینی ترجیح داده می‌شود. این رتبه‌بندی‌ها در شکل 5 نشان‌داده‌شده است.



*شکل 5- رتبه بندی های ترجیحی بین تاکوترون 2 و نمونه های حقیقت زمین. 800 رتبه بندی از 100 مورد وجود دارد. برچسب ها با توجه به تاکوترون 2 بیان می شوند. شکل استخراج شده از (شن و همکاران، 2017).*

برای مقایسه کلی تکنیک‌های شرح داده شده در این بخش، ما در شکل 6 تمام MOS هر روش پیشرفته را گزارش می‌کنیم. توجه داشته باشید که ما بیش از یک نتیجه برای گفتار انسان داریم، زیرا حداقل دو مطالعه به طور مستقل آن را ارزیابی می‌کنند. MOS برای گفتار انسان در هر دو یکسان است و تفاوت بین آنها از لحاظ آماری معنی‌دار نیست. بااین‌حال توجه داشته باشید که این دو مطالعه دارای نویسندگان مشترک هستند و بنابراین ممکن است یک سوگیری مشترک داشته باشند. روش‌های ارزیابی کمی متفاوت از MOS ممکن است نتایج متفاوتی به همراه داشته باشد که یکی از دلایلی است که MOS مورد انتقاد قرار می‌گیرد.



**شکل 6- MOS برای متدهای ارائه شده در این بخش.**

1. **انتقال یادگیری از تأیید بلندگو به ترکیب متن به گفتار چند سخنران**
2. **بررسی اجمالی**

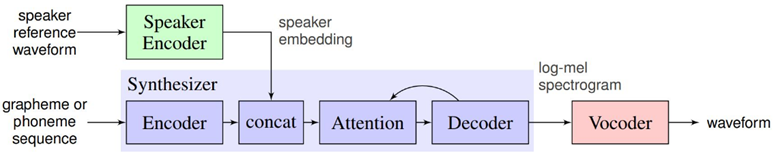
رویکرد ما به شبیه‌سازی صدای بلادرنگ عمدتاً مبتنی بر (جیا و همکاران، 2018) است (که در این سند به آن SV2TTS گفته می‌شود). چارچوبی را برای شبیه‌سازی صدای صفر شات که تنها به 5 ثانیه گفتار مرجع نیاز دارد، توصیف می‌کند. این مقاله تنها یکی از انتشارات متعدد از سری تاکوترون 5 است که در گوگل نوشته شده است. جالب اینجاست که مقاله SV2TTS به‌خودی‌خود نوآوری چندانی به همراه ندارد، بلکه بر اساس سه کار اصلی قبلی گوگل است: the GE2E loss (Wan et al., 2017) و Tacotron (Wang et al., 2017) و WaveNet (van den Oord et al., 2016)

چارچوب کامل یک خط لوله سه‌مرحله‌ای است که در آن مراحل با مدل‌های ذکر شده قبلی مطابقت دارد. بسیاری از ابزارها و قابلیت‌های فعلی TTS ارائه شده توسط Google، مانند Google assistant یا Google cloud services، از همین مدل‌ها استفاده می‌کنند. درحالی‌که بسیاری از پیاده‌سازی‌های منبع‌باز این مدل‌ها به‌صورت آنلاین وجود دارد، هیچ یک از چارچوب SV2TTS در دانش ما وجود ندارد (از ماه می 2019).

**مراحل سه‌گانه چارچوب به شرح زیر است:**

* یک رمزگذار بلندگو که از گفتار کوتاه یک بلندگو یک جاسازی می‌گیرد. تعبیه نمایش معناداری از صدای گوینده است، به‌طوری‌که صداهای مشابه در فضای نهفته نزدیک هستند. این مدل در (Wan et al., 2017) (به‌عنوان GE2E در سراسر این سند ذکر شده است) و (Heigold et al., 2015) توضیح داده شده است.
* سینت سایزری که مشروط به تعبیه یک بلندگو، یک طیف گرا از متن ایجاد می‌کند. این مدل محبوب Tacotron 2 (شن و همکاران، 2017) بدون WaveNet است (که اغلب به‌عنوان Tacotron به دلیل شباهت آن به اولین تکرار شناخته می‌شود).
* یک Vocoder که یک‌شکل موج صوتی را از طیف‌نگاری‌های تولید شده توسط سینت سایزر استنباط می‌کند. نویسندگان از WaveNet (van den Oord و همکاران، 2016) به‌عنوان یک کد صوتی استفاده کردند که به طور مؤثر کل چارچوب Tacotron 2 را مجدداً اعمال کرد.

در زمان استنتاج، رمزگذار بلندگو یک بیانیه مرجع کوتاه از گوینده را به شبیه‌سازی می‌دهد. این یک جاسازی را ایجاد می‌کند که برای شرطی‌سازی سینت سایزر استفاده می‌شود، و یک متن پردازش شده به‌عنوان یک دنباله واج به‌عنوان ورودی به سینتی سایزر داده می‌شود. Vocoder خروجی سینت سایزر را می‌گیرد تا شکل موج گفتار را تولید کند. این موضوع در شکل 7 نشان‌داده‌شده است.



**شکل 7- چارچوب SV2TTS در طول استنتاج. بلوک‌های آبی نمای سطح بالایی از معماری تاکوترون را نشان می‌دهند که به گونه‌ای اصلاح شده است که امکان تنظیم صدا را فراهم می‌کند. شکل استخراج شده از (جیا و همکاران، 2018).**

یکی از ویژگی‌های چارچوب SV2TTS این است که همه مدل‌ها را می‌توان به طور جداگانه و بر روی مجموعه‌داده‌های مجزا آموزش داد. برای رمزگذار، شخص به دنبال مدلی است که در برابر نویز مقاوم باشد و بتواند بسیاری از ویژگی‌های صدای انسان را ثبت کند؛ بنابراین، مجموعه بزرگی از بسیاری از بلندگوهای مختلف برای آموزش رمزگذار ترجیح داده می‌شود، بدون نیاز به سطح نویز صدا.

علاوه بر این، رمزگذار با از دست دادن GE2E آموزش داده شده است که به هیچ برچسبی غیر از هویت بلندگو نیاز ندارد. با GE2E، وظیفه ای که باید توسط مدل یاد شود، یک کار تایید بلندگو است، که به خودی خود ربطی به شبیه سازی صدا ندارد. با این حال، این وظیفه به گونه ای تعیین شده است که شبکه یک جاسازی را که نمایش معنی دار صدای گوینده است، خروجی دهد. این تعبیه برای شرطی کردن سینت سایزر بر روی یک صدا مناسب است، از این رو نام مقاله: انتقال یادگیری از تأیید بلندگو به ترکیب متن به گفتار چند سخنران. کیفیت صدای تولید شده فقط می تواند به خوبی داده ها باشد. بنابراین مجموعه داده های با کیفیت بالاتر و حاشیه نویسی مورد نیاز است، که اغلب به این معنی است که اندازه آنها کوچکتر است.

در بخش زیر، به طور رسمی وظیفه‌ای را که هدف SV2TTS حل آن است، تعریف می‌کنیم. در بخش‌های فرعی باقی‌مانده، ما سه بخش چارچوب را همان‌طور که در (جیا و همکاران، 2018) توضیح داده شد، پیاده‌سازی هر یک و نتیجه آزمایش‌هایمان ارائه می‌کنیم.

1. **تعريف مسئله**

مجموعه داده‌ای از گفته‌ها را که توسط گوینده آنها گروه‌بندی‌شده‌اند در نظر بگیرید. گفتار j ام گوینده i را می گوییم. گفته ها در حوزه شکل موج هستند. ما با طیف‌نگار عبارت را نشان می‌دهیم. طیف‌نگار یک تابع قطعی و غیر قابل معکوس (از دست دادن) است که ویژگی‌های گفتار را از شکل موج استخراج می‌کند تا گفتار را به شیوه‌ای قابل تحمل‌تر در یادگیری ماشین مدیریت کند.

رمزگذار تعبیه شد مربوط به بیان را محاسبه می کند، جایی که پارامترهای رمزگذار هستند. علاوه بر این، نویسندگان تعبیه‌کننده‌ی سخنران را به‌عنوان مرکز جاسازی‌های گفته‌های گوینده تعریف می‌کنند:

سینت سایزر S که توسط پارامتر شده است وظیفه دارد را با توجه به و ، رونوشت بیان ، تقریب کند. داریم. در اجرای خود، ما مستقیماً از تعبیه گفتار به جای تعبیه گوینده استفاده می کنیم (این انتخاب را در بخش 3.4 ایجاد می کنیم)، در عوض را می دهیم.

در نهایت، Vocoder که توسط پارامتر شده است، وظیفه دارد که را با تقریبی کند. داریم:

می‌توان این چارچوب را به‌صورت سرتاسری با تابع هدف زیر آموزش داد:

جایی که یک تابع ضرر در حوزه شکل موج است. این روش دارای معایبی است:

* این نیاز به آموزش هر سه مدل در یک مجموعه‌داده یکسان دارد، به این معنی که این مجموعه‌داده در حالت ایده‌آل باید الزامات همه مدل‌ها را برآورده کند: تعداد زیادی بلندگو برای رمزگذار، اما در همان زمان، رونوشت‌هایی برای سینت سایزر. یک نویز سطح پایین برای سینتی سایزر و به‌نوعی سطح نویز متوسط برای رمزگذار (به‌طوری‌که بتوان گفتار ورودی پرسروصدا را مدیریت کرد). این تضادها مشکل‌ساز هستند و منجر به مدل‌های آموزشی می‌شوند که اگر به طور جداگانه بر روی مجموعه‌داده‌های متمایز آموزش داده شوند، می‌توانند عملکرد بهتری داشته باشند. به طور خاص، یک مجموعه‌داده کوچک به‌احتمال زیاد منجر به تعمیم ضعیف و در نتیجه عملکرد ضعیف صفر شات می‌شود.
* دستیابی به همگرایی مدل ترکیبی بسیار سخت است. به طور خاص، سینت سایزر تاکوترون می‌تواند زمان قابل‌توجهی را قبل از ایجاد ترازهای صحیح ببرد.

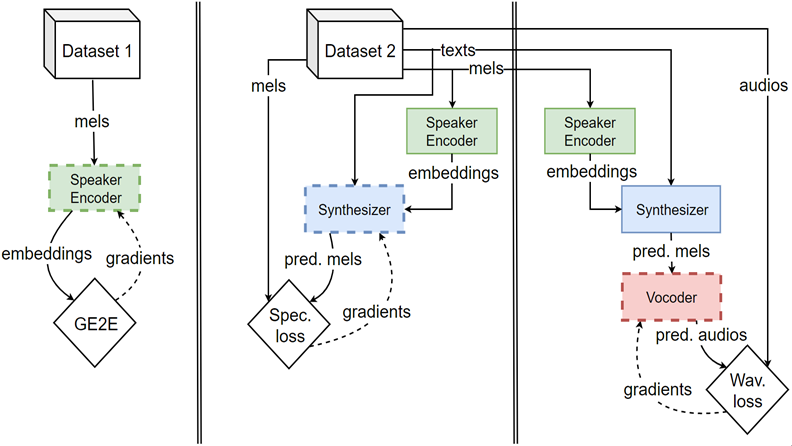
یک راه آشکار برای پرداختن به موضوع دوم، جداسازی آموزش سینت سایزر و صداگذار است. با فرض یک رمزگذار از پیش آموزش‌دیده، سینت سایزر را می‌توان آموزش داد تا به طور مستقیم طیف‌سنجی mel صدای هدف را پیش‌بینی کند:

جایی که یک تابع ضرر در حوزه زمان-فرکانس است.

سپس صداگذار مستقیماً بر روی طیف‌نگارها آموزش داده می‌شود. توجه داشته باشید که هر دو رویکرد آموزش بر روی طیف‌نگارهای حقیقت زمینی یا طیف‌نگارهای تولید شده توسط سنتزکننده معتبر هستند. دومی نیاز به یک سینت سایزر از پیش آموزش‌دیده دارد.

یا

بهینه‌سازی رمزگذار بلندگو باقی‌مانده است. بر خلاف سینت سایزر و رمزگذار صدا، رمزگذار برچسبی برای آموزش ندارد. این وظیفه به طور بدی به‌عنوان تولید جاسازی‌های معنادار تعریف می‌شود که صدا را در گفتار مشخص می‌کند. می‌توان راهی برای آموزش رمزگذار بلندگو به‌عنوان رمزگذار خودکار تصور کرد، اما برای پیش‌بینی نیاز به مدل آپ نمونه‌سازی متناظر با متن دارد. درهرصورت مجموعه‌داده به یک جمله محدود می‌شود، یا یکی به رونوشت نیاز دارد و مدل نمونه‌برداری سینت سایزر است. در هر دو مورد، کیفیت آموزش توسط مجموعه‌داده مختل می‌شود و بعید است به‌خوبی تعمیم یابد. خوشبختانه، ازدست‌دادن GE2E (Wan et al., 2017)راه حلی برای این مشکل به ارمغان می‌آورد و امکان آموزش رمزگذار بلندگو را مستقل از سینت سایزر فراهم می‌کند. این موضوع در بخش بعد توضیح داده شده است.



**شکل 8- آموزش سه مرحله‌ای متوالی SV2TTS (به دنبال انتخاب‌های ما برای پیاده‌سازی). مدل هایی با خطوط کانتور جامد منجمد شده اند. توجه داشته باشید که طیف نگارهای mel که به رمزگذار بلندگو و آنهایی که به عنوان هدف سینت سایزر استفاده می شوند با پارامترهای متفاوتی ایجاد می‌شوند.**

درحالی‌که تمام بخش‌های چارچوب به طور جداگانه آموزش داده می‌شوند، هنوز هم نیاز است که سینتی‌سایزر دارای تعبیه‌هایی از یک رمزگذار آموزش‌دیده باشد و برای Vocoder باید طیف‌نگاری‌های mel را از یک سینت‌سایزر آموزش‌دیده داشته باشد (اگر روی طیف‌نگار حقیقت زمینی آموزش داده نشود). شکل 8 نشان می‌دهد که چگونه هر مدل برای آموزش به مدل قبلی وابسته است. رمزگذار بلندگو نیاز به تعمیم کافی برای ایجاد تعبیه‌های معنادار روی مجموعه‌داده سینت سایزر دارد و حتی زمانی که بر روی یک مجموعه‌داده مشترک آموزش داده می‌شود، همچنان باید بتواند در زمان استنتاج در حالت صفر شات عمل کند.

1. **رمزگذار بلندگو**

مدل رمزگذار و روش آموزش آن در چندین مقاله توضیح داده شده است (جیا و همکاران، 2018؛ وان و همکاران، 2017؛ هیگلد و همکاران، 2015). ما این مدل را با پیاده‌سازی PyTorch خودمان بازتولید کردیم. ما قطعات مربوط به SV2TTS و همچنین انتخاب‌های خود را برای پیاده‌سازی ترکیب می‌کنیم.

1. **معماری مدل**

این مدل یک LSTM سه‌لایه با 768 گره پنهان است که به دنبال آن یک‌لایه طرح‌ریزی 256 واحدی وجود دارد. درحالی‌که در هیچ یک از مقالات اشاره‌ای به چیستی لایه طرح‌ریزی (پروجکشن) وجود ندارد، شهود ما این است که به‌سادگی یک‌لایه 256 خروجی کاملاً متصل در هر LSTM است که به طور مکرر برای هر خروجی LSTM اعمال می‌شود. هنگامی که ما برای اولین‌بار رمزگذار بلندگو را پیاده‌سازی کردیم، برای نمونه‌سازی سریع، ساده و برای بار آموزشی سبک‌تر، به طور مستقیم از 256 واحد لایه LSTM استفاده کردیم. این بخش آخر مهم است، زیرا نویسندگان مدل خود را برای 50 میلیون مرحله آموزش داده‌اند (اگرچه در یک مجموعه‌داده بزرگ‌تر) که از لحاظ فنی بازتولید آن برای ما دشوار است. ما متوجه شدیم که این مدل کوچک‌تر بسیار خوب عمل می‌کند، و زمانی را برای آموزش نسخه بزرگ‌تر در آینده پیدا نکرده‌ایم.

ورودی‌های این مدل طیف‌نگارهای log-mel چهل کاناله با عرض پنجره 25 میلی‌ثانیه و گام 10 میلی‌ثانیه است. خروجی حالت پنهان L2 نرمال شده آخرین لایه است که بردار 256 عنصر است. پیاده‌سازی ما همچنین دارای یک‌لایه ReLU قبل از عادی‌سازی است، با این هدف که تعبیه‌ها را پراکنده و در نتیجه آسان‌تر قابل تفسیر کند.

1. **ضایعات کلی از انتها به انتها**

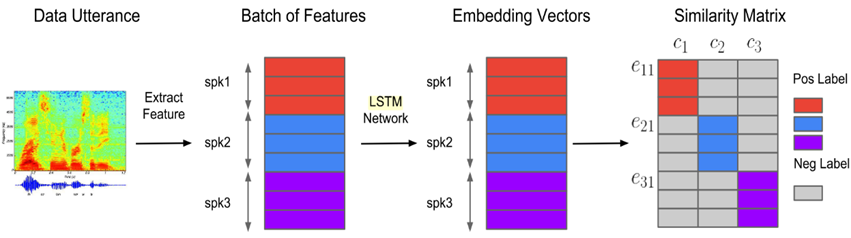
رمزگذار بلندگو در مورد یک کار تأیید بلندگو آموزش‌دیده است. تأیید بلندگو یک کاربرد معمولی از بیومتریک است که در آن هویت یک شخص از طریق صدای آنها تأیید می‌شود. با استخراج تعبیه بلندگو (به معادله 1) از چند گفته، یک الگو برای یک شخص ایجاد می‌شود. به این فرایند ثبت‌نام می‌گویند. در زمان اجرا، کاربر خود را با یک گفتار کوتاه شناسایی می‌کند و سیستم جاسازی آن گفته را با جاسازی‌های بلندگوی ثبت‌شده مقایسه می‌کند. بالاتر از یک آستانه شباهت معین، کاربر شناسایی می‌شود. ازدست‌دادن GE2E این فرایند را برای بهینه‌سازی مدل شبیه‌سازی می‌کند.

در زمان آموزش، مدل تعبیه‌های از M گفته‌های مدت ثابت از N بلندگو را محاسبه می‌کند.

یک بلندگوی جاسازی شده برای هر بلندگو مشتق شده است:

ماتریس شباهت نتیجه مقایسه دو به دو همه جاسازی‌های X در برابر هر بلندگوی جاسازی‌شده در دسته است. این معیار تشابه کسینوس مقیاس شده است:

که در آن w و b پارامترهای قابل یادگیری هستند. کل این فرایند در شکل 9 نشان‌داده‌شده است. از منظر محاسباتی، شباهت کسینوس دو بردار با هنجار L2 به‌سادگی حاصل‌ضرب نقطه آنهاست، بنابراین سمت راست معادله 2 است. انتظار می‌رود که یک مدل بهینه مقادیر تشابه بالا را زمانی که یک گفته با گوینده مطابقت دارد و مقادیر کمتری در جای دیگر تولید کند. برای بهینه سازی در این جهت، ضرر مجموع تلفات softmax ردیفی است.



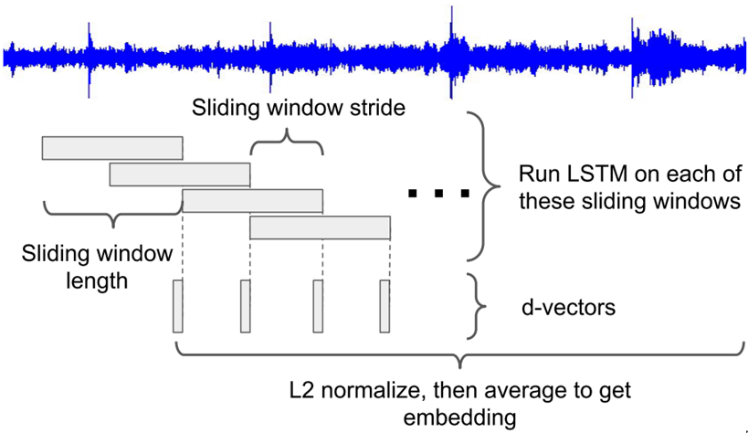
**شکل 9- ساخت ماتریس شباهت در زمان آموزش. این شکل از (وان و همکاران، 2017) استخراج شده است.**

توجه داشته باشید که هر گفته در هنگام محاسبه افت در مرکز همان بلندگو گنجانده شده است. این امر باعث ایجاد تعصب نسبت به بلندگوی صحیح مستقل از دقت مدل می شود. و نویسندگان استدلال می‌کنند که این راه‌حل برای راه‌حل‌های پیش پاافتاده نیز باقی می گذارد. برای جلوگیری از این امر، سخنی که با تعبیه بلندگوی خود مقایسه می‌شود، از جاسازی بلندگو حذف می‌شود.

سپس ماتریس شباهت به‌صورت زیر تعریف می‌شود:

که در آن مرکزهای انحصاری به صورت زیر تعریف می شوند:

مدت‌زمان ثابت گفته‌ها در یک دسته آموزشی 1.6 ثانیه است. این‌ها جملات جزئی هستند که از جملات کامل‌تر در مجموعه‌داده نمونه‌برداری شده‌اند. درحالی‌که معماری مدل قادر است ورودی‌های طول متغیر را مدیریت کند، منطقی است انتظار داشته باشیم که با گفته‌هایی با مدت‌زمان مشابه آنچه در آموزش‌دیده می‌شود بهترین عملکرد را داشته باشد؛ بنابراین، در زمان استنتاج، یک گفته به بخش‌های 1.6 ثانیه‌ای تقسیم می‌شود که 50٪ با هم همپوشانی دارند و رمزگذار هر بخش را به‌صورت جداگانه به جلو می‌برد. خروجی‌های به‌دست‌آمده میانگین می‌شوند و سپس نرمال می‌شوند تا تعبیة گفته را ایجاد کنند. این موضوع در شکل 10 نشان‌داده‌شده است. جالب اینجاست که نویسندگان SV2TTS از ویندوزهای 800 میلی‌ثانیه در زمان استنتاج حمایت می‌کنند، اما هنوز در طول آموزش 1.6 ثانیه طول می‌کشند. همان‌طور که در GE2E انجام می‌شود ترجیح می‌دهیم 1.6 ثانیه را برای هر دو نگه داریم.



**شکل 10- محاسبه تعبیه یک گفته کامل. بردارهای d صرفاً خروجی های غیر عادی مدل هستند. این شکل از (وان و همکاران، 2017) استخراج شده است.**

نویسندگان از و به عنوان پارامترهایی برای اندازه دسته استفاده می کنند. هنگام ثبت نام یک سخنران در یک برنامه کاربردی، باید انتظار داشت که چندین گفته از هر کاربر داشته باشد، اما احتمالاً مرتبه بزرگی بالاتر از 10 نیست، بنابراین این انتخاب معقول است. در مورد تعداد بلندگوها، خوب است مشاهده کنید که پیچیدگی زمانی محاسبه ماتریس تشابه است. بنابراین، این پارامتر باید نه خیلی بزرگ انتخاب شود تا به طور قابل ملاحظه‌ای سرعت آموزش را کاهش ندهد، نه صرفاً انتخاب بزرگ‌ترین سایز دسته‌ای که بر روی پردازنده گرافیکی(GPU) قرار می‌گیرد. البته هنوز هم امکان موازی سازی چندین دسته در یک GPU وجود دارد در حالی که عملیات در دسته ها برای کارایی همگام سازی می شود. ما تشخیص دادیم که هنگام محاسبه ماتریس شباهت، بردار کردن تمام عملیات‌ها برای به حداقل رساندن تعداد تراکنش‌های GPU، اهمیت ویژه‌ای دارد.

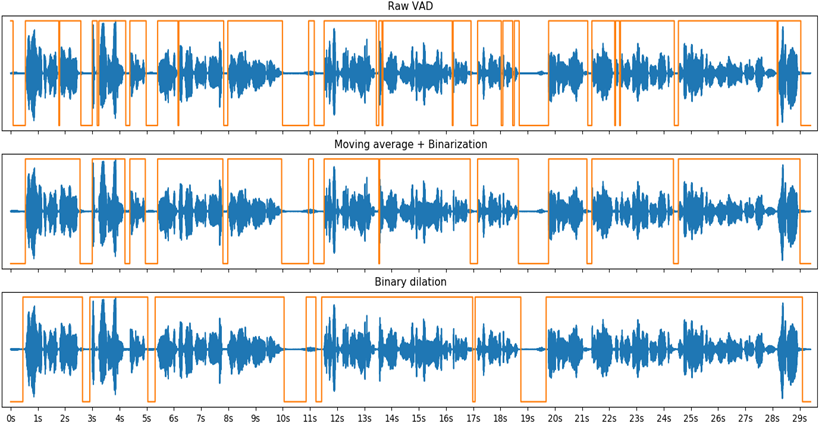
1. **آزمایشات**

برای جلوگیری از بخش‌هایی که عمدتاً هنگام نمونه‌برداری از جملات جزئی از گفته‌های کامل بی‌صدا هستند، از بسته پایتون webrtcvad برای انجام تشخیص فعالیت صوتی (VAD) استفاده می‌کنیم. این یک پرچم باینری را بر روی صدا نشان می‌دهد که مربوط به صداگذاری یا عدم صداگذاری بخش است.

ما یک میانگین متحرک را روی این پرچم باینری انجام می‌دهیم تا میخ‌های کوتاه در تشخیص را هموار کنیم که سپس دوباره آنها را باینری می‌کنیم. در نهایت، یک اتساع بر روی پرچم با اندازه هسته انجام می دهیم، که در آن حداکثر مدت زمان سکوت قابل تحمل است.

سپس صدا از قسمت‌های بی‌صدا حذف می‌شود. ما متوجه شدیم که مقدار ثانیه انتخاب خوبی است که عروض گفتاری طبیعی را حفظ می کند.

این فرایند در شکل 11 نشان‌داده‌شده است. آخرین مرحله پیش‌پردازشی که برای شکل موج‌های صوتی اعمال می‌شود، نرمال‌سازی است، تا حجم متفاوت بلندگوها در مجموعه‌داده را جبران کند.



**شکل 11- مراحل حذف سکوت با VAD از بالا به پایین. خط نارنجی پرچم صدای باینری است که در آن مقدار بالایی به این معنی است که قطعه صدا داده می‌شود و زمانی که پایین‌تر باشد، صداگذاری نشده است.**

نویسندگان چندین مجموعه‌داده پرسروصدا را با هم ترکیب کردند تا مجموعه بزرگی از گفتار با کیفیتی مشابه آنچه در طبیعت یافت می‌شود، به دست آورند. این مجموعه‌داده‌ها عبارت‌اند از LibriSpeech (Panayotov et al., 2015) و VoxCeleb1 (Nagrani et al., 2017) و VoxCeleb2 (Chung et al., 2018) و یک مجموعه‌داده داخلی که ما به آن دسترسی نداریم.

LibriSpeech مجموعه‌ای از کتاب‌های صوتی است که برای 1000 ساعت صدا از 2400 بلندگو تشکیل شده است که به طور مساوی در دو مجموعه «پاک» و «دیگر» تقسیم می‌شود. ظاهراً مجموعه تمیز از گفتار تمیزتری نسبت به مجموعه دیگر تشکیل شده است، حتی اگر برخی از قسمت‌های مجموعه تمیز همچنان دارای نویز زیادی باشد (Zen et al., 2019).  
VoxCeleb1 و VoxCeleb2 از بخش‌های صوتی استخراج شده از ویدئوهای یوتیوب افراد مشهور (اغلب در زمینه مصاحبه) ساخته شده‌اند. VoxCeleb1 دارای بلندگو است، در حالی که VoxCeleb2 حدود دارد. هر دوی این مجموعه داده ها دارای سخنرانان غیرانگلیسی هستند. ما از اکتشافی مبتنی بر ملیت گوینده برای فیلتر کردن موارد غیرانگلیسی از مجموعه آموزشی در VoxCeleb1 استفاده کردیم، اما نتوانستیم همان اکتشافات را در VoxCeleb2 اعمال کنیم، زیرا ملیت در آن مجموعه ارجاع نشده است. توجه داشته باشید که بدون آزمایش مشخص نیست که آیا داشتن غیرانگلیسی زبان به آموزش رمزگذار لطمه می زند یا خیر (نویسندگان نیز به آن توجهی نکرده اند). همه این مجموعه داده ها در 16 کیلوهرتز نمونه برداری شده اند.

نویسندگان ترکیب‌های مختلف این مجموعه‌داده‌ها را آزمایش می‌کنند و تأثیر آن را بر کیفیت جاسازی‌ها مشاهده می‌کنند. آنها اندازه خروجی مدل LSTM (اندازه جاسازی‌ها) را باتوجه‌به تعداد بلندگوها به 64 یا 256 تنظیم می‌کنند.

آنها طبیعی بودن ذهنی و شباهت را با حقیقت اصلی گفتار تولید شده توسط یک سینتی سایزر آموزش‌دیده از تعبیه‌های تولید شده توسط هر مدل ارزیابی می‌کنند. آنها همچنین میزان خطای مساوی رمزگذار را در تأیید بلندگو گزارش می‌دهند که در ادامه در این بخش به آن می‌پردازیم. این نتایج در جدول 2 آمده است.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SV-EER | Similarity | Naturalness | Embedding Dim | Speakers | Training Set |
| 16.60% | 2.23 ± 0.08 | 3.73 ± 0.06 | 64 | 1.2K | LS-Clean |
| 15.32% | 2.27 ± 0.09 | 3.60 ± 0.06 | 64 | 1.2K | LS-Other |
| 11.95% | 2.43 ± 0.09 | 3.83 ± 0.06 | 256 | 2.4K | LS-Other+VC |
| 10.14% | 2:54 ± 0:09 | 3.82 ± 0.06 | 256 | 8.4K | • LS-Other+VC+VC2 |
| 5.08% | 3.03 ± 0.09 | 4.12 ± 0.05 | 256 | 18K | Internal |

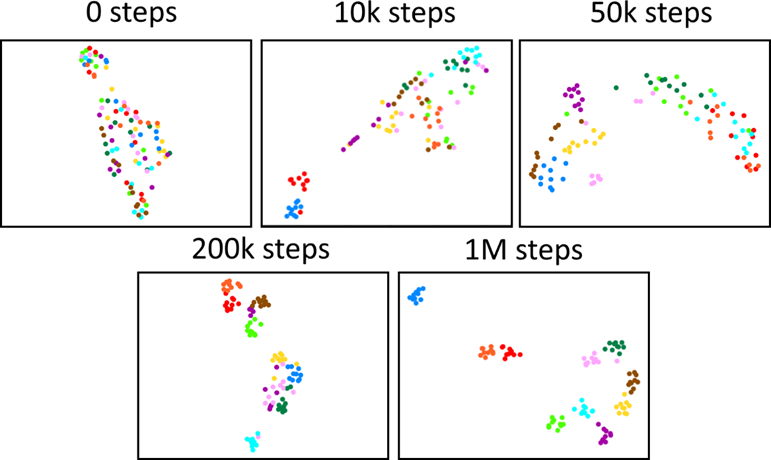
جدول 2: آموزش رمزگذار بلندگو بر روی مجموعه‌داده‌های مختلف، از (جیا و همکاران، 2018). LS LibriSpeech و VC VoxCeleb است. سینت سایزرها بر روی LS-Clean آموزش‌داده‌شده و در یک مجموعه آزمایشی ارزیابی می‌شوند. خط با گلوله پیاده‌سازی است که ما قصد داریم آن را بازتولید کنیم.

این نتایج نشان می‌دهد که تعداد سخنرانان به‌شدت با عملکرد خوب نه‌تنها رمزگذار در کار تأیید، بلکه همچنین کل چارچوب در مورد کیفیت گفتار تولید شده و توانایی آن در شبیه‌سازی صدا، همبستگی دارد. جهش کوچک در طبیعی بودن، شباهت و EER به‌دست‌آمده با گنجاندن VoxCeleb2 احتمالاً می‌تواند نشان‌دهنده این باشد که تنوع زبان‌ها به آموزش آسیب می‌رساند. مجموعه‌داده‌های داخلی نویسندگان یک مجموعه جستجوی صوتی اختصاصی از 18 هزار انگلیسی‌زبان است. رمزگذار آموزش‌داده‌شده بر روی این مجموعه‌داده به طور قابل‌توجهی بهتر عمل می‌کند، بااین‌حال ما فقط به مجموعه‌داده‌های عمومی دسترسی داریم؛ بنابراین با LibriSpeech-Other، VoxCeleb1 و VoxCeleb2 ادامه می‌دهیم.

ما رمزگذار بلندگو را برای یک میلیون مرحله آموزش می‌دهیم. برای نظارت بر آموزش، EER را گزارش می‌کنیم و توانایی مدل را برای خوشه‌بندی بلندگوها مشاهده می‌کنیم. ما به‌صورت دوره‌ای از یک دسته از 10 سخنران با 10 گفتار نمونه‌برداری می‌کنیم، جاسازی‌های گفته را محاسبه می‌کنیم و آنها را در فضایی دوبعدی با UMAP نمایش می‌دهیم (مک‌اینز و هیلی، 2018). ازآنجایی‌که انتظار می‌رود جاسازی‌های سخنرانان مختلف در فضای پنهان بیشتر از همدیگر باشند تا جاسازی‌هایی از همان بلندگوها، انتظار می‌رود که با پیشرفت آموزش، خوشه‌هایی از گفته‌ها از یک سخنران یکسان شکل بگیرند. ما پیش‌بینی‌های UMAP خود را در شکل 12 گزارش می‌کنیم، جایی که این رفتار قابل‌مشاهده است.

همان‌طور که قبلاً ذکر شد، نویسندگان مدل خود را برای 50 میلیون مرحله در مجموعه‌داده‌های اختصاصی خودآموزش داده‌اند. درحالی‌که هم مجموعه‌داده ما و هم مدل ما کوچک‌تر هستند، مدل ما هنوز در 1 میلیون مرحله همگرا نشده است. ضرر به طور پیوسته با واریانس کمی کاهش می‌یابد و همچنان می‌تواند بیشتر کاهش یابد، اما ما محدود به زمان هستیم.

بااین‌وجود، مدل به‌دست‌آمده نتایج بسیار قوی را به همراه دارد. در واقع، مجموعه آزمون EER را 4.5 درصد محاسبه کردیم. باتوجه‌به 10.14% نویسندگان برای همان مجموعه با 50 برابر مراحل بیشتر، این مقدار بسیار کم است. ما نمی‌دانیم که آیا مدل ما واقعاً به این خوبی عمل می‌کند یا اینکه آیا روش محاسبه EER نویسندگان به‌اندازه کافی با ما متفاوت است تا مقادیری را بافاصله زیاد از هم تولید کند.

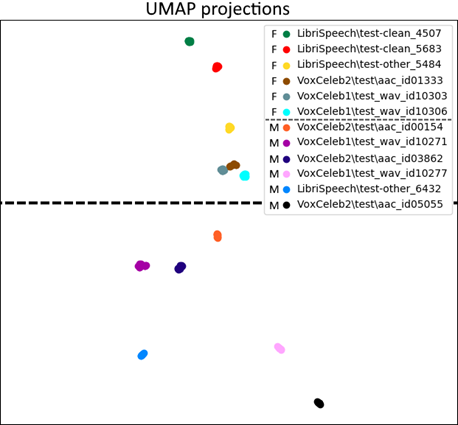


**شکل 12- پیش‌بینی‌های UMAP از جاسازی‌های گفته از دسته‌های انتخابی تصادفی از قطار مجموعه‌ای در تکرارهای مختلف مدل ما. گفته های یک بلندگو با نقطه ای از همان رنگ نشان داده می شود. ما به‌طور خاص از ارسال برچسب‌ها به UMAP صرفنظر می‌کنیم، بنابراین خوشه‌بندی به‌طور کامل توسط مدل انجام می‌شود.**

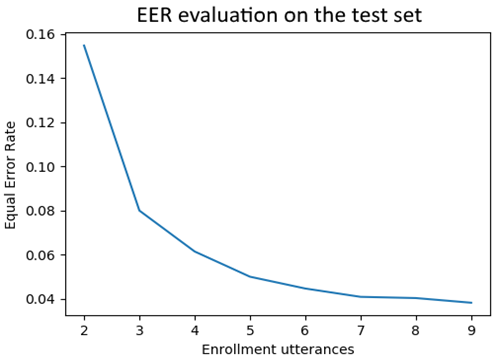
ما درمی‌یابیم که خوشه‌بندی در فضای پنهان تولید شده توسط مدل ما به طرز چشمگیری قوی است و به‌خوبی تعمیم می‌یابد. در تمام آزمایش‌های ما، پیش‌بینی‌های UMAP گفته‌ها را کاملاً از مجموعه آزمایشی هر یک از سه مجموعه‌داده با فواصل بین خوشه‌ای بزرگ و واریانس درون خوشه‌ای کوچک جدا می‌کند. نمونه‌ای در شکل 13 آورده شده است. مجموعه تست مورداستفاده برای این ارزیابی ترکیبی از مجموعه تست‌های LibriSpeech، VoxCeleb1 و VoxCeleb2 است. بلندگوهایی که با F حاشیه‌نویسی شده‌اند، گویندگان زن هستند و آنهایی که M دارند، سخنرانان مرد هستند. ما نتایج خود را با (جیا و همکاران، 2018، شکل 3) مقایسه می‌کنیم. متوجه می‌شویم که پیش‌بینی‌های ما نیز جنسیت سخنرانان را به‌صورت خطی در فضای پیش‌بینی‌شده جدا می‌کند. در شکل نویسنده، گفته‌ها فقط از LibriSpeech است. ما VoxCeleb1 و VoxCeleb2 را نیز شامل می‌شویم، با برخی از سخنرانانی که به طور تصادفی انتخاب شده‌اند که به انگلیسی صحبت نمی‌کنند. توجه داشته باشید که خوشه‌های ما متراکم‌تر از خوشه‌های نویسندگان هستند، اما این تنها نتیجه استفاده از تکنیک کاهش ابعاد متفاوت است. ما هنگام استفاده از T-SNE نتایجی شبیه به نتایج آنها پیدا می‌کنیم (ون در ماتن و هینتون، 2008).

نرخ خطای برابر (EER) معیاری است که معمولاً در سیستم‌های بیومتریک برای ارزیابی دقت سیستم استفاده می‌شود. مقدار نرخ مثبت کاذب زمانی است که برابر با نرخ منفی واقعی باشد. معادل‌سازی آن عبارات با تغییر آستانه تشابه انجام می‌شود که بالاتر از آن کاربر توسط سیستم بیومتریک شناسایی می‌شود. نویسندگان SV2TTS به روش خود برای ارزیابی EER اشاره‌ای نکرده‌اند. این مسئله مشکل‌ساز است؛ زیرا محاسبه EER مشکل است و به‌شدت به تعداد اظهارات ثبت‌نام انتخاب شده بستگی دارد.

ما به GE2E مراجعه می‌کنیم و از 6 گفته برای ثبت‌نام استفاده می‌کنیم و آن‌ها را با 7 گفته مقایسه می‌کنیم. برای تعداد مختلف گفته‌ها، به شکل 14 مراجعه کنید. نویسندگان آنها به این موضوع اشاره نمی‌کنند که گفته‌هایی که استفاده می‌کنند (هم برای ثبت‌نام و هم برای آزمایش) جملات کامل یا جزئی هستند. باتوجه‌به نتایج ما، از جملات جزئی استفاده می‌کنیم. اما گفته‌های کامل می‌تواند EER حتی کمتری را به همراه داشته باشد.



**شکل 13- پیش بینی های UMAP از 120 جاسازی، 10 عدد برای هر یک از 12 بلندگو. شش سخنران مرد و شش سخنران زن به طور تصادفی از مجموعه های آزمایشی انتخاب می شوند.**



**شکل 14- تأیید EER سخنران در مجموعه آزمون با توجه به تعداد اظهارات ثبت نام. هر سخنران در برابر n + 1 گفته ای که برای ثبت نام استفاده می شود آزمایش می شود.**

به دلیل فقدان اطلاعات ارائه شده توسط نویسندگان SV2TTS و ازآنجایی‌که ما هیچ منبع قابل اعتمادی پیدا نکردیم که نشان دهد EER چگونه برای چنین سیستمی محاسبه می‌شود، نمی‌توانیم صحت یا عادلانه بودن مقایسه نتایج خود را با نتایج نویسندگان تضمین کنیم. بااین‌حال، مستقل از آنها، آزمایش‌های ما مطمئناً نشان می‌دهد که رمزگذار بلندگوی ما در تأیید صحت بلندگو بسیار خوب عمل می‌کند. با درنظرگرفتن این موارد، ما مطمئن هستیم که رمزگذار بلندگو در حال ایجاد جاسازی (تعبیه) های معنی‌دار است.

در مورد سرعت استنتاج، رمزگذار بسیار سریع‌ترین در بین سه مدل است، زیرا تقریباً 1000× زمان واقعی بر روی GTX 1080 GPU ما کار می‌کند. در واقع، زمان اجرا با ورودی/خروجی GPU برای همه گفته‌های مجموعه‌داده ما محدود می‌شود.

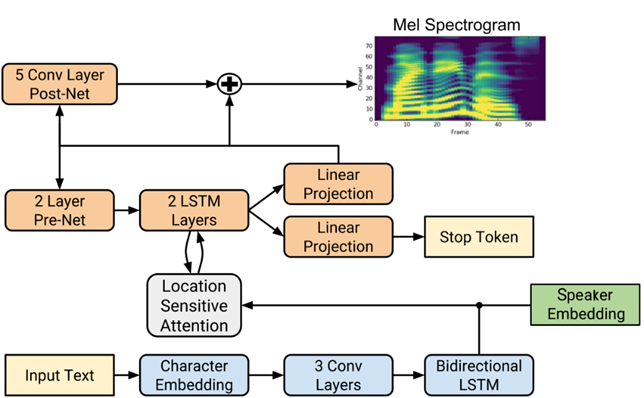
1. **سینت سایزر**

سینت سایزر Tacotron 2 بدون Wavenet است (ون دن اوورد و همکاران، 2016). ما از یک پیاده‌سازی متن‌باز Tensorflow از Tacotron 2 استفاده می‌کنیم که Wavenet را از آن جدا می‌کنیم و تغییرات اضافه شده توسط SV2TTS را پیاده‌سازی می‌کنیم.

1. **معماری مدل**

ما به طور خلاصه معماری سطح بالای تاکوترون 2 اصلاح شده را بدون Wavenet (که به‌سادگی تاکوترون از آن یاد می‌کنیم) ارائه می‌دهیم. برای جزئیات بیشتر، از خواننده دعوت می‌کنیم تا نگاهی به مقالات تاکوترون بیندازند (شن و همکاران، ۲۰۱۷؛ وانگ و همکاران، ۲۰۱۷).

تاکوترون یک مدل توالی به دنباله تکرارشونده است که یک طیف‌نگار mel را از متن پیش‌بینی می‌کند. دارای ساختار رمزگشای رمزگذار (نباید با رمزگذار بلندگوی SV2TTS اشتباه شود) که توسط مکانیزم توجه حساس به مکان پل شده است (چوروفسکی و همکاران، 2015). کاراکترهای جداگانه از دنباله متن ابتدا به‌عنوان بردار جاسازی می شوند. لایه‌های کانولوشنال به دنبال افزایش گستره یک قاب رمزگذار منفرد هستند. این فریم‌ها برای تولید فریم‌های خروجی رمزگذار از طریق یک LSTM دوطرفه عبور داده می‌شوند. اینجا جایی است که SV2TTS تغییری در معماری ایجاد می‌کند: تعبیه بلندگو به هر فریمی که توسط رمزگذار Tacotron خروجی می‌شود، متصل می‌شود. مکانیزم توجه به فریم‌های خروجی رمزگذار برای تولید فریم‌های ورودی رمزگشا توجه می‌کند. هر فریم ورودی رمزگشا با خروجی قاب رمزگشای قبلی که از یک شبکه پیش‌تر عبور داده می‌شود، الحاق می‌شود و مدل را خود هم‌بسته (Autoregressive model) می‌کند. این بردار به‌هم‌پیوسته از دولایه LSTM یک‌طرفه عبور می‌کند قبل از اینکه به یک قاب طیفی mel منفرد نمایش داده شود. طرح دیگری از همان بردار به یک اسکالر به شبکه اجازه می‌دهد تا به‌تنهایی پیش‌بینی کند که تولید فریم‌ها را با انتشار مقداری بالاتر از یک آستانه مشخص متوقف کند. کل توالی فریم‌ها قبل از تبدیل‌شدن به طیف‌نگار mel از یک پس شبکه باقی‌مانده عبور داده می‌شود. این معماری در شکل 15 نشان‌داده‌شده است.



**شکل 15- معماری تاکوترون اصلاح شده. بلوک های آبی مربوط به رمزگذار و بلوک های نارنجی مربوط به رمزگشا هستند. این رقم از (شن و همکاران، 2017) استخراج و اصلاح شد.**

طیف‌نگارهای mel هدف برای سینت سایزر، ویژگی‌های بیشتری نسبت به آنهایی که برای رمزگذار بلندگو استفاده می‌شوند، ارائه می‌دهند. آنها از یک پنجره 50 میلی‌ثانیه با گام 12.5 میلی‌ثانیه محاسبه می‌شوند و 80 کانال دارند. متون ورودی برای تلفظ در پیاده‌سازی ما پردازش نمی‌شوند و کاراکترها همان‌طور که هستند تغذیه می‌شوند. بااین‌حال، چند روش تمیزکردن وجود دارد: جایگزین‌کردن اختصارات و اعداد با فرم متنی کامل آنها، اجبار کردن همه کاراکترها به ASCII (استاندارد کدگذاری آمریکایی برای تبادل اطلاعات یا American Standard Code for Information Interchange)، عادی‌سازی فضاهای سفید و کوچک کردن همه کاراکترها. می‌توان از علائم نگارشی استفاده کرد، اما در مجموعه‌داده‌های ما وجود ندارد.

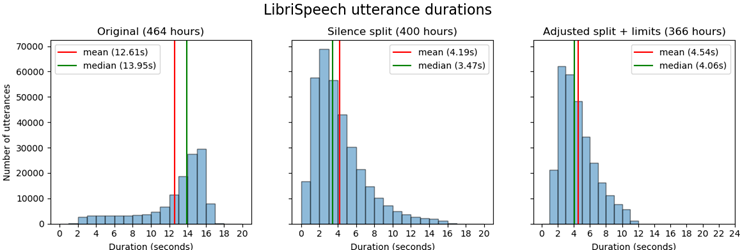
1. **آزمایشات**

در SV2TTS، نویسندگان دو مجموعه‌داده را برای آموزش سینت سایزر و رمزگذار صدا در نظر می‌گیرند. اینها LibriSpeech-Clean هستند که قبلاً به آن اشاره کردیم و VCTK که مجموعه‌ای از تنها 109 زبان مادری انگلیسی است که با تجهیزات حرفه‌ای ضبط شده است. گفتار در VCTK در 48 کیلوهرتز نمونه‌برداری می‌شود و در آزمایشات آنها به 24 کیلوهرتز نمونه‌برداری می‌شود که هنوز بالاتر از نمونه‌برداری 16 کیلوهرتز LibriSpeech است. آنها دریافتند که یک سینت سایزر آموزش‌داده‌شده بر روی LibriSpeech در مورد شباهت بهتر از VCTK تعمیم می‌یابد، اما به قیمت طبیعی بودن گفتار. آنها این را با آموزش سینت سایزر در یک مجموعه و آزمایش آن در مجموعه دیگر ارزیابی می‌کنند. این نتایج در جدول 3 آمده است. ما تصمیم گرفتیم با مجموعه داده‌ای کار کنیم که بهترین شباهت شبیه‌سازی صدا را در بلندگوهای دیده نشده ارائه می‌دهد و بنابراین LibriSpeech را انتخاب کردیم. ما همچنین سعی کرده‌ایم از مجموعه‌داده‌های جدیدتر LibriTTS (Zen و همکاران، 2019) که توسط تیم Tacotron ایجاد شده است استفاده کنیم. این مجموعه‌داده یک نسخه تمیزتر از کل مجموعه LibriSpeech با بلندگوهای پرسروصدا، نرخ نمونه‌برداری بالاتر 24 کیلوهرتز و علائم نگارشی است که LibriSpeech فاقد آن است. متأسفانه، به دلایلی که ما نادیده می‌گیریم، سینتی‌سایزر نمی‌تواند ترازهای معنی‌داری را در این مجموعه‌داده ایجاد کند. ما مجموعه‌داده اصلی LibriSpeech را به‌جای آن نگه داشتیم.

|  |  |  |  |
| --- | --- | --- | --- |
| Similarity | Naturalness | Testing Set | Synthesizer Training Set |
| 1.82 ± 0.08 | 4.28 ± 0.05 | LibriSpeech | VCTK |
| 2.77 ± 0.08 | 4.01 ± 0.06 | VCTK | LibriSpeech |

جدول 3: ارزیابی متقابل مجموعه‌داده‌ها در مورد طبیعی بودن و شباهت سخنران برای سخنرانان دیده نشده. این جدول از (جیا و همکاران، ۲۰۱۸) استخراج شده است.

به دنبال توصیه‌های پیش‌پردازش نویسندگان، ما از یک مدل تشخیص خودکار گفتار (ASR) برای تراز کردن رونوشت‌های LibriSpeech با متن استفاده می‌کنیم. ما متوجه شدیم که Montreal Forced Aligner (تراز اجباری مونترال)در این کار به‌خوبی عمل می‌کند. ما همچنین یک نسخه پاک‌تر از این ترازها را عمومی کرده‌ایم تا برای سایر کاربرانی که به آن‌ها نیاز دارند، در زمان صرفه‌جویی کنیم. با تراز کردن صوت با متن، گفته‌ها را در سکوت‌های طولانی‌تر از 0.4 ثانیه تقسیم می‌کنیم. این به سینت سایزر کمک می‌کند تا همگرا شود، هم به دلیل حذف سکوت در طیف‌نگار هدف، و هم به دلیل کاهش مدت‌زمان متوسط گفته‌ها در مجموعه‌داده، زیرا توالی‌های کوتاه‌تر فضای کمتری را برای خطاهای زمان‌بندی ارائه می‌دهند. ما اطمینان می‌دهیم که گفته‌ها کوتاه‌تر از 1.6 ثانیه، مدت‌زمان گفته‌های جزئی که برای آموزش رمزگذار استفاده می‌شوند، و بیش از 11.25 ثانیه نباشد تا حافظه GPU برای آموزش ذخیره شود. ما در سکوتی که در صورت امکان بیانی خیلی کوتاه یا خیلی طولانی ایجاد کند، تقسیم نمی‌کنیم. توزیع طول گفته‌ها در مجموعه‌داده در شکل 16 نشان‌داده‌شده است. توجه داشته باشید که چه مدت سکوت قبلاً 64 ساعت (13.7٪) از مجموعه‌داده را به خود اختصاص داده است.



**شکل 16- (سمت چپ) هیستوگرام مدت زمان گفته ها در LibriSpeech-Clean، (وسط) پس از تقسیم بر روی سکوت، (راست) پس از محدود کردن طول و تنظیم مجدد تقسیمات.**

جداسازی سکوت‌ها با هم‌تراز کردن متن با گفته‌ها، علاوه بر این امکان ایجاد نمایه‌ای از نویز را برای همه گفته‌های یک گوینده می‌دهد. ما از پیاده‌سازی پایتون 13 از الگوریتم LogMMSE استفاده می‌کنیم (افرایم و مالا، 1985). LogMMSE یک بخش گفتار صوتی را با نمایه‌سازی نویز در اولین چند فریم (که معمولاً هنوز شامل گفتار نمی‌شود) و به‌روزرسانی این نمایه نویز در فریم‌های غیرگفتاری به طور مداوم در طول گفتار پاک می‌کند. ما این پیاده‌سازی را با مشخصات نویز و پاک‌سازی گفتار در دو مرحله جداگانه تطبیق می‌دهیم. هم‌تراز با نویسندگان، ما این مرحله پیش‌پردازش اضافی را یافتیم که به کاهش نویز پس‌زمینه طیف‌نگارهای سنتز شده کمک زیادی می‌کند.

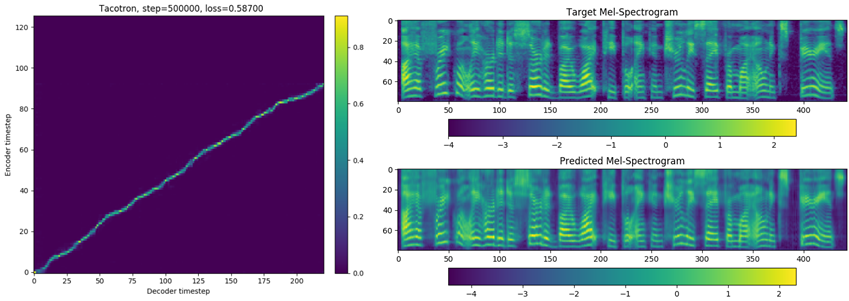
در SV2TTS، تعبیه‌هایی که برای شرطی‌سازی سینت سایزر در زمان آموزش استفاده می‌شوند، تعبیه‌های بلندگو هستند. ما استدلال می‌کنیم که تعبیه‌های بیانی همان گفتار هدف، انتخاب طبیعی‌تری را به‌جای آن ایجاد می‌کند. در زمان استنتاج، از تعبیه‌های گفته نیز استفاده می‌شود. درحالی‌که فضای بیان و تعبیه‌های بلندگو یکسان است، جاسازی‌های بلندگو با L2 نرمال‌سازی نمی‌شوند. این تفاوت در دامنه باید کوچک باشد و تأثیر کمی بر سینت سایزری که از جاسازی استفاده می‌کند، داشته باشد، همان‌طور که نویسندگان وقتی از آنها در مورد آن سؤال کردیم موافقت کردند. بااین‌حال، آنها اشاره‌ای نمی‌کنند که چه تعداد جاسازی گفته برای استخراج یک جاسازی بلندگو استفاده شده است. می‌توان انتظار داشت که از تمام گفته‌های موجود استفاده شود. اما با تعداد بیشتری از تعبیه‌های گفته، بردار متوسط (جاسازی بلندگو) بیشتر از نسخه عادی خود فاصله می‌گیرد. علاوه بر این، نویسندگان به خود اشاره می‌کنند که اغلب تغییرات زیادی از لحن و زیروبم در گفته‌های یک گوینده در مجموعه‌داده وجود دارد، زیرا آنها از شخصیت‌های مختلف تقلید می‌کنند (جیا و همکاران، 2018، پیوست B). گفته‌ها دارای تنوع درونی کمتری هستند، زیرا دامنه آنها حداکثر به یک جمله محدود می‌شود؛ بنابراین، انتظار می‌رود که تعبیه یک گفته، نمایش دقیق‌تری از صدای گفته شده در گفتار نسبت به جاسازی گوینده باشد. این امر در صورتی صدق می‌کند که گفته به‌اندازه کافی طولانی باشد تا یک جاسازی معنادار ایجاد کند. درحالی‌که مدت‌زمان «بهینه» گفتار مرجع 5 ثانیه بود، جاسازی در حال حاضر با تنها 2 ثانیه گفتار مرجع معنادار است (جدول 4 را ببینید). ما معتقدیم که در صورت کوتاه‌تر از مدت‌زمان گفته‌های جزئی (1.6 ثانیه)، جاسازی گفتار باید برای ضبط معنی‌دار صدا کافی باشد، بنابراین از جاسازی‌های گفته برای آموزش سینتی سایزر استفاده کردیم.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Reference utterance duration | | | | |  |
| 10 sec 5 sec 3 sec sec 2 1 sec | | | | |
| 4:16 ± 0:06 | 4:20 ± 0:06 | 4:18 ± 0:06 | 4:26 ± 0:05 | 4:28 ± 0:05 | Naturalness (MOS) |
| 3:18 ± 0:07 | 3:28 ± 0:07 | 3:31 ± 0:07 | 3:17 ± 0:07 | 2:85 ± 0:07 | Similarity (MOS) |
| 11.50% | 10.46% | 10.80% | 11.30% | 17.28% | SV-EER |

جدول 4: تأثیر مدت‌زمان گفتار مرجع در VCTK ارزیابی شده است. این جدول از (جیا و همکاران، 2018) استخراج شده است.

ما سینتی سایزر را برای 150 هزار مرحله آموزش می‌دهیم، با اندازه دسته‌ای 144 در 4 پردازنده گرافیکی. تعداد خروجی‌های رمزگشا در هر مرحله بر روی 2 تنظیم شده است، همان‌طور که در Tacotron 1 انجام شد. ما متوجه شدیم که مدل با یک خروجی در هر مرحله رمزگشا یا همگرا نیست یا عملکرد ضعیفی دارد.

تابع اتلاف افت L2 بین طیف‌نگارهای پیش‌بینی‌شده و پایه‌ای حقیقت مل است. در طول آموزش، مدل در حالت تراز حقیقت زمینی (GTA) تنظیم می‌شود (که به آن حالت معلم اجباری نیز می‌گویند) که در آن ورودی به شبکه پیش شبکه به‌جای طیف‌نگار پیش‌بینی‌شده، فریم قبلی طیف‌نگار حقیقت زمینی است. با GTA، زیروبم و عروض طیف گرام تولید شده با حقیقت زمینی همسو می‌شود و زمینه مشترک بین پیش‌بینی و حقیقت زمین و همچنین هم‌گرایی سریع‌تر را امکان‌پذیر می‌کند. بدون GTA، سینتی سایزر باتوجه‌به یک متن ثابت و ورودی جاسازی شده (همان‌طور که در زمان استنتاج وجود دارد) تغییرات متفاوتی از یک گفته تولید می‌کند. همان‌طور که در بخش «سنتز گفتار پارامتریک آماری» موردبحث قرار گرفت، در مورد Vocoder نیز وجود دارد، ارائه هرگونه ارزیابی کمی از عملکرد مدل دشوار است. ما می‌توانیم مشاهده کنیم که مدل خروجی‌های درستی را از طریق تست‌های شنیداری غیررسمی تولید می‌کند، اما یک ارزیابی رسمی از ما نیاز دارد که نظرسنجی‌های نمره ذهنی را برای استخراج MOS تنظیم کنیم. درحالی‌که برخی از نویسندگانی که ما به آنها اشاره کردیم می‌توانند این کار را انجام دهند، اما این کار از دسترس ما خارج است. بااین‌حال، در مورد سینتی‌سایزر، می‌توان صحت هم‌ترازی‌های ایجاد شده توسط ماژول توجه را نیز تأیید کرد. ما یک مثال را در شکل 17 ترسیم می‌کنیم. به تعداد مراحل رمزگشا (223) توجه کنید که با تعداد فریم‌های پیش‌بینی‌شده (446) با تعداد خروجی‌های رمزگشا در هر مرحله (2) مطابقت دارد. همچنین توجه کنید که طیف‌نگار پیش‌بینی‌شده چگونه از حقیقت زمین صاف‌تر است، یک رفتار معمولی از مدل که میانگین را در حضور نویز پیش‌بینی می‌کند.

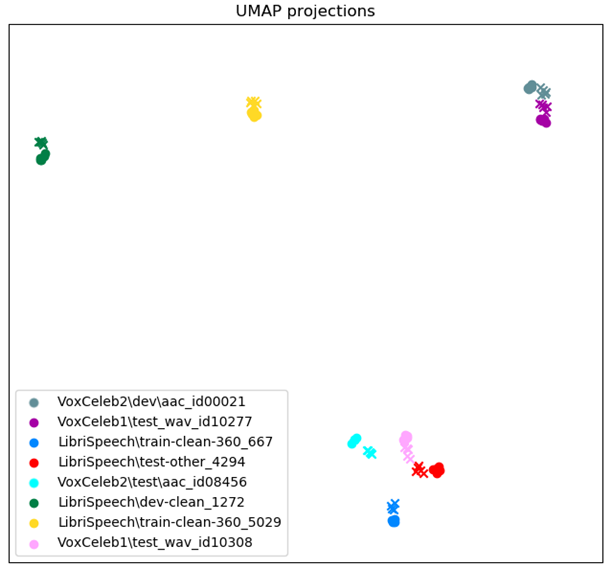


**شکل 17- (سمت چپ) نمونه ای از تراز بین مراحل رمزگذار و مراحل رمزگشا. (راست) مقایسه بین طیف‌نگار پیش‌بینی‌شده GTA و طیف‌نگار حقیقت زمین.**

قبل از آموزش رمزگذار صدا، می‌توانیم برخی از جنبه‌های سینتی‌سایزر آموزش‌دیده را با استفاده از Griffin-Lim (Griffin and Jae Lim-1984) به‌عنوان کدکننده صدا ارزیابی کنیم. Griffin-Lim یک مدل یادگیری ماشینی نیست، بلکه یک الگوریتم تکراری است که سیگنال صوتی منبع یک طیف گرام را تخمین می‌زند. صدای تولید شده به این روش معمولاً تعداد کمی از ویژگی‌های صوتی گوینده را حفظ می‌کند، اما گفتار قابل‌درک است. گفتار تولید شده توسط سینت سایزر، حتی در حضور کلمات پیچیده یا ساختگی، به‌درستی با متن مطابقت دارد. بااین‌حال، عروض گاهی غیرطبیعی است، با مکث در مکان‌های غیرمنتظره در جمله، یا عدم توقف در جایی که انتظار می‌رود. این امر به‌ویژه با تعبیه برخی از بلندگوها که به‌آرامی صحبت می‌کنند، قابل‌توجه است و نشان می‌دهد که رمزگذار بلندگو نوعی عروض را ضبط می‌کند. فقدان علائم نگارشی در LibriSpeech تا حدی مسئول این امر است و مدل را مجبور می‌کند تا تنها از متن علائم نگارشی را استنباط کند. این موضوع توسط نویسندگان نیز برجسته شده است و می‌توان آن را در برخی از نمونه‌های 14 سخنرانان LibriSpeech آنها شنید. محدودیت‌هایی که ما بر روی مدت‌زمان گفته‌ها در مجموعه‌داده اعمال کردیم (s1.6s - 11.25) احتمالاً مشکل‌ساز هستند. جملاتی که خیلی کوتاه هستند با مکث‌های طولانی کشیده می‌شوند و برای جملاتی که بیش از حد طولانی هستند صدا شتاب‌زده خواهد شد. هنگام تولید چندین جمله در زمان استنتاج، باید به‌صورت دستی وقفه‌هایی را وارد کنیم تا جایی که متن ورودی را تقسیم کنیم تا طیف‌نگار را در چندین بخش ترکیب کنیم. این مزیت ایجاد دسته‌ای از ورودی‌ها به‌جای ورودی طولانی را دارد که امکان استنتاج سریع را فراهم می‌کند.

ما می‌توانیم بیشتر مشاهده کنیم که چگونه برخی از ویژگی‌های صوتی با Griffin-Lim با محاسبه جاسازی‌های گفتار ترکیبی و نمایش آنها با UMAP همراه با جاسازی‌های حقیقت زمینی از بین می‌روند. یک مثال در شکل 18 آورده شده است. مشاهده می‌کنیم که خوشه‌های تعبیه‌شده سنتز شده نزدیک به خوشه تعبیه‌های حقیقت پایه مربوطه خود هستند. ازدست‌دادن ویژگی‌های نوظهور نیز قابل‌مشاهده است، به‌عنوان‌مثال برای گوینده‌های صورتی، قرمز و آبی، گفته‌های سنتز شده واریانس بین خوشه‌ای کمتری نسبت به همتای حقیقت پایه خود دارند. این پدیده در بلندگوهای خاکستری و بنفش نیز رخ می‌دهد.

Tacotron معمولاً سریع‌تر از زمان واقعی عمل می‌کند. ما سرعت استنتاج 5× تا 10× زمان واقعی را با مقایسه زمان تولید با طول مدت طیف‌نگار تولید شده اندازه‌گیری می‌کنیم.



**شکل 18- پیش بینی جاسازی های حقیقت زمینی و جاسازی های گفتاری سنتز شده توسط گریفین-لیم که از همان جاسازی های حقیقت زمینی ایجاد شده اند. جاسازی های حقیقت زمینی با دایره ها و جاسازی های سنتز شده با ضربدر ترسیم می شوند.**

1. **Vocoder**

در SV2TTS و در Tacotron2، WaveNet کد صوتی است. WaveNet از زمان انتشار خود در قلب یادگیری عمیق با صدا قرار داشته است و در مورد طبیعی بودن صدا در TTS همچنان در حال پیشرفت است. بااین‌حال همچنین به‌عنوان کندترین معماری یادگیری عمیق عملی در زمان استنتاج شناخته شده است. چندین مقاله بعدی بهبودهایی را در این جنبه به ارمغان آوردند تا نسل را به زمان واقعی یا سریع‌تر از زمان واقعی برساند، مثلاً. (وان دن اوورد و همکاران، 2017؛ پین و همکاران، 2016؛ کالچبرنر و همکاران، 2018)، بدون هیچ ضربه‌ای یا تقریباً هیچ ضربه‌ای به کیفیت گفتار تولید شده وارد نشده است. بااین‌وجود، WaveNet همچنان به‌عنوان صداگذار در SV2TTS باقی می‌ماند، زیرا سرعت نگرانی اصلی نیست و به دلیل اینکه پیاده‌سازی WaveNet خود گوگل با پیشرفت‌های مختلف در حال حاضر 8000 نمونه در ثانیه تولید می‌کند (Kalchbrenner و همکاران، 2018، صفحه 2). این در تضاد با WaveNet "وانیلی" است که در بهترین حالت با سرعت 172 گام در ثانیه تولید می‌کند (van den Oord و همکاران، 2017، صفحه 7). در زمان نگارش این پایان‌نامه، اکثر پیاده‌سازی‌های متن‌باز WaveNet هنوز هم پیاده‌سازی‌های وانیلی هستند.

(Kalchbrenner و همکاران، 2018) یک طرح ساده برای توصیف سرعت استنتاج مدل‌های خود هم‌بسته پیشنهاد می‌کند. باتوجه‌به بردار هدف با نمونه‌های برای پیش بینی، کل زمان استنتاج را می توان به صورت زیر تجزیه کرد:

که در آن N تعداد محصولات ماتریس- بردار (تعداد لایه‌های) موردنیاز برای تولید یک نمونه است.

زمان محاسبه لایه و سربار محاسبه (معمولاً عملیات ) برای لایه است. توجه داشته باشید که نرخ نمونه‌برداری استاندارد برای گفتار شامل 16 کیلوهرتز، 22.05 کیلوهرتز و 24 کیلوهرتز است (در حالی که موسیقی معمولا با فرکانس 44.1 کیلوهرتز نمونه‌برداری می‌شود)، به این معنی که فقط برای 5 ثانیه صدا نزدیک به 100000 نمونه است. معماری استاندارد WaveNet شامل سه دسته از 10 بلوک باقیمانده از هر کدام از دو لایه است که به منجر می شود.

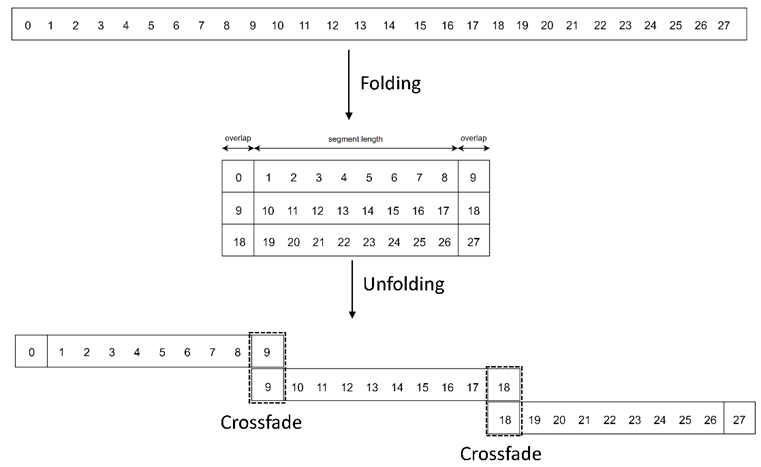
WaveRNN، مدل ارائه شده در (Kalchbrenner et al., - 2018) ، WaveNet را نه‌تنها با کاهش سهم ، بلکه از ، و بهبود می بخشد. مدل Vocoder که ما استفاده می کنیم، پیاده سازی PyTorch منبع باز 15 است که مبتنی برWaveRNN است، اما چندین انتخاب طراحی متفاوت را ارائه می دهد که توسط کاربر github fatchord انجام شده است. ما به این معماری به عنوان \جایگزین WaveRNN اشاره خواهیم کرد.

1. **معماری مدل**

در WaveRNN، کل 60 کانولوشن از WaveNet با یک‌لایه GRU جایگزین می‌شود (چو و همکاران، 2014). نویسندگان معتقدند که غیرخطی بودن بالای یک‌لایه GRU به‌تنهایی به‌اندازه کافی نزدیک است که پیچیدگی کل مدل WaveNet را در بر بگیرد. در واقع، آنها یک MOS برای Wavenet و برای بهترین مدل WaveRNN خود گزارش کردند. ورودی‌های مدل، طیف‌نگار GTA mel هستند که توسط سینت سایزر تولید می‌شوند و صدای حقیقت زمینی به‌عنوان هدف است. در زمان آموزش، مدل بخش‌های شکل موج با اندازه ثابت را پیش‌بینی می‌کند. عبور روبه‌جلو WaveRNN فقط با محصول ماتریس بردار در یک طرح درشت ریز اجرا می شود که در آن ابتدا 8 بیت پایین (درشت) نمونه 16 بیتی هدف پیش بینی می شود و سپس برای شرطی کردن پیش بینی 8 بیت بالاتر استفاده می شود. (دقیق). پیش بینی بر روی پارامترهای توزیعی است که خروجی از آن نمونه برداری شده است. برای جزئیات بیشتر، خواننده را به ( Kalchbrenneret al.,- 2018) ارجاع می دهیم.

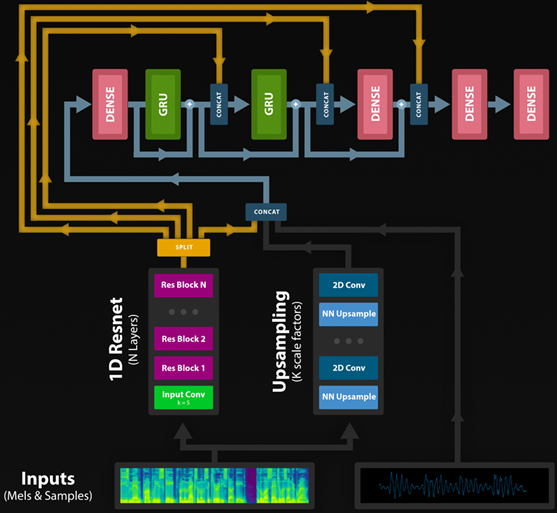
نویسندگان عوامل و را با اجرای عملیات نمونه برداری به عنوان یک عملیات GPU سفارشی بهبود می بخشند. ما این کار را تکرار نمی کنیم. آنها همچنین شبکه را با استراتژی هرس از (نارنگ و همکاران، 2017؛ ژو و گوپتا، 2017) پراکنده می کنند. این روش بر خلاف الگوریتم‌های هرس کلاسیک که بین چندین تمرین عمل می‌کنند، به تدریج وزنه‌ها را در طول تمرین هرس می‌کند. این الگوریتم یک ماسک باینری روی وزن ها ایجاد می کند که نشان می دهد وزن باید به اجبار روی 0 تنظیم شود یا به همان صورت باقی بماند. نسبت وزن های صفر نسبت به تعداد کل وزن ها در شبکه را پراکندگی می گویند. نتایج حاصل از (نارنگ و همکاران، 2017؛ ژو و گوپتا، 2017) نشان می دهد که شبکه های بزرگ با سطوح پراکندگی بین 90 تا 95 درصد به طور قابل توجهی از نسخه های متراکم خود بهتر عمل می کنند. نویسندگان WaveRNN علاوه بر این استدلال می کنند که متناسب با تعداد وزن های غیر صفر است. ما با این شکل از هرس آزمایش می کنیم و نتایج خود را در بخش 3.5.2 گزارش می کنیم.

در نهایت، آنها در جوج با نمونه‌برداری دسته‌ای بهبود می‌یابند. در نمونه‌گیری دسته‌ای، گفته به بخش‌هایی با طول ثابت تقسیم می‌شود و تولید به طور موازی در تمام بخش‌ها انجام می‌شود. برای حفظ مقداری زمینه بین انتهای یک بخش و ابتدای قسمت بعدی، قسمت کوچکی از انتهای یک بخش در ابتدای قسمت بعدی تکرار می‌شود. به این فرایند فولدینگ (folding) می‌گویند. سپس مدل بخش‌های تا شده را به جلو می‌فرستد. برای بازیابی تانسور باز شده، بخش‌های همپوشانی بخش‌های متوالی با یک محو شدن متقاطع ادغام می‌شوند. این در شکل 19 نشان‌داده‌شده است. ما از نمونه‌برداری دسته‌ای با WaveRNN جایگزین، با طول قطعه 8000 نمونه و طول همپوشانی 400 نمونه استفاده می‌کنیم. با این پارامترها، یک دسته تا شده با اندازه 2 کمی بیش از 1 ثانیه صوت برای گفتار 16 کیلوهرتز تولید می‌کند.



**شکل 19- نمونه برداری دسته ای از یک تانسور. توجه داشته باشید که چگونه همپوشانی بر روی هر بخش بعدی در تانسور تا شده تکرار می شود.**

WaveRNN جایگزین، معماری است که ما استفاده می‌کنیم. هیچ سند و مقاله‌ای برای این مدل وجود ندارد، بنابراین برای درک عملکرد درونی آن به کد منبع و نمودار نویسنده (شکل 20) تکیه می‌کنیم. در هر مرحله تمرین، یک طیف‌نگار مل و شکل موج مربوط به آن در همان تعداد قطعه بریده می‌شوند. ورودی‌های مدل عبارت‌اند از قطعه طیف گرام t برای پیش‌بینی و قطعه شکل موج . انتظار می‌رود که مدل قطعه شکل موج t را با طول یکسان خروجی دهد. طیف‌نگار mel از طریق یک شبکه نمونه‌برداری بالا می‌رود تا با طول شکل موج هدف مطابقت داشته باشد (تعداد کانال‌های mel ثابت می‌ماند). یک مدل Resnet مانند نیز از طیف‌نگار به عنوان ورودی برای تولید ویژگی‌هایی استفاده می‌کند که لایه‌ها را در طول تبدیل طیف‌نگار mel به شکل موج شرطی می‌کند. بردار حاصل تکرار می شود تا با طول قطعه شکل موج مطابقت داشته باشد. سپس این بردار شرطی‌سازی به طور مساوی به چهار جهت در امتداد بعد کانال تقسیم می‌شود و بخش اول با طیف‌نگار نمونه‌برداری شده و با بخش شکل موج مرحله زمانی قبلی پیوند می‌یابد. بردار حاصل از چندین تغییر شکل با اتصالات پرش می‌گذرد: ابتدا دو لایه GRU سپس یک لایه متراکم. بین هر مرحله، بردار شرطی‌سازی با شکل موج میانی به هم متصل می‌شود. در نهایت، دو لایه متراکم توزیعی را بر روی مقادیر گسسته ایجاد می‌کنند که مربوط به کدگذاری 9 بیتی صدای ترکیب‌شده mu-law است.



**شکل 20- معماری جایگزین WaveRNN.**

1. **آزمایشات**

هنگامی که با عبارات کوتاه سروکار دارید، رمزگذار صوتی معمولاً کمتر از زمان واقعی کار می‌کند. سرعت استنتاج به‌شدت به تعداد چین‌ها (تا ها) در نمونه‌گیری دسته‌ای بستگی دارد. در واقع، شبکه تقریباً در زمان ثابتی باتوجه‌به تعداد چین‌ها کار می‌کند و با افزایش تعداد چین‌ها، تنها زمان کمی افزایش می‌یابد. ما متوجه می‌شویم که ساده‌تر است که در مورد آستانه‌ای از مدت‌زمان گفتار صحبت کنیم که مدل در زمان واقعی بالاتر از آن اجرا می‌شود. در تنظیمات ما، این آستانه 12.5 ثانیه است. به این معنی که برای عبارات کوتاه‌تر از این آستانه، مدل کندتر از زمان واقعی اجرا می‌شود. به نظر می‌رسد که عملکرد به طور غیرمنتظره‌ای با برخی از عوامل محیطی (مانند سیستم‌عامل) در PyTorch متفاوت است، و بنابراین ما نتایج خود را باتوجه‌به یک پیکربندی واحد بیان می‌کنیم.

پیاده‌سازی روی دست ما دارای عملیات GPU سفارشی توسط ( Kalchbrenner et al.,- 2018) نیست و پیاده‌سازی آن فراتر از توانایی‌های ما است. در عوض، ما بر جنبه هرس ذکر شده توسط نویسندگان تمرکز می‌کنیم. آنها ادعا می‌کنند که یک WaveRNN پراکنده بزرگ بهتر و سریع‌تر از یک متراکم کوچک‌تر عمل می‌کند. ما الگوریتم هرس را آزمایش کرده‌ایم، اما به دلیل محدودیت‌های زمانی، آموزش مدل هرس شده را کامل نکردیم. این نقطه عطفی است که امیدواریم در آینده به آن دست پیدا کنیم.

تانسورهای پراکنده، در زمان نگارش، درعین‌حال یک ویژگی آزمایشی در PyTorch هستند. پیاده‌سازی آنها ممکن است به‌اندازه آنچه که نویسندگان استفاده می‌کردند کارآمد نباشد. از طریق آزمایش‌ها، متوجه می‌شویم که عملیات ضرب ماتریس addmm برای یک ماتریس پراکنده و یک بردار متراکم، تنها از نظر زمانی با addmm متراکم برای سطوح پراکندگی بالای 91 درصد، یکنواخت می‌شود. در زیر این مقدار، استفاده از تانسورهای پراکنده در واقع سرعت عبور به جلو را کاهش می‌دهد. نویسندگان سطوح پراکندگی 96.4٪ و 97.8٪ را گزارش می‌کنند (Kalchbrenner و همکاران، 2018، جدول 5) درحالی‌که عملکرد مناسب را حفظ می‌کنند. آزمایش‌های ما نشان می‌دهد که در بهترین حالت، سطح پراکندگی 96.4 درصد آستانه بلادرنگ را به 7.86 ثانیه و سطح 97.8 درصد را به 4.44 ثانیه کاهش می‌دهد. به دلیل فرض استنتاج زمان ثابت ما، و همچنین به این دلیل که برخی از لایه‌های مدل را نمی‌توان از هم جدا کرد، اینها مرزهای پایین‌تر خوش‌بینانه در آستانه واقعی هستند. این تجزیه‌وتحلیل اولیه نشان می‌دهد که هرس کردن Vocoder برای سرعت استنتاج مفید است.

متأسفانه، ما نتوانستیم با موفقیت نسخه نهایی Vocoder را قبل از ارسال این پایان‌نامه آموزش دهیم. ما موفق شدیم یک نمونه اولیه را تا ماه فوریه کار کنیم، اما این مدل دیگر با تغییراتی که ما در چارچوب ایجاد کرده‌ایم سازگار نیست. ما مصمم هستیم که قبل از دفاع از این پایان‌نامه یک اجرای کاری ارائه دهیم، اما فعلاً نمی‌توانیم آزمایش‌های جدیدی را گزارش کنیم. برداشت ما از نمونه اولیه این بود که پیاده‌سازی ما در ایجاد یک مدل TTS که می‌تواند اکثر صداها را شبیه‌سازی کند، اما نه برخی از صداهای غیرمعمول، موفقیت‌آمیز بوده است. برخی از مصنوعات و نویز پس‌زمینه به دلیل کیفیت پایین سینت سایزر ما وجود داشت، به همین دلیل است که ما مجبور شدیم کیفیت داده‌ها و روش‌های پیش‌پردازش خود را تجدیدنظر کنیم. یکی از اشکالات چارچوب SV2TTS، ضرورت آموزش مدل‌ها به ترتیب متوالی است. هنگامی که یک رمزگذار جدید آموزش داده می‌شود، سینت سایزر باید دوباره آموزش داده شود و همچنین رمزگذار صدا نیز باید دوباره آموزش داده شود. انتظار برای آموزش مدل‌ها برای اینکه بدانیم روی چه چیزهایی باید تمرکز کنیم، یک وضعیت تکرارشونده در توسعه چارچوب ما بوده است. بااین‌وجود، ما بر این باوریم که به‌زودی می‌توانیم نمونه‌هایی را در مخزن خود منتشر کنیم، و از خواننده دعوت می‌کنیم تا پیشرفت‌های آینده را دنبال کند، حتی اگر از تاریخ ارسال فراتر رود.

1. **جعبه ابزار و منبع باز**

بخشی از تلاش‌های توسعه برای مناسب ساختن این پروژه برای منبع‌باز انجام شد. می‌توانید از طریق این آدرس اینترنتی به مخزن دسترسی داشته باشید:

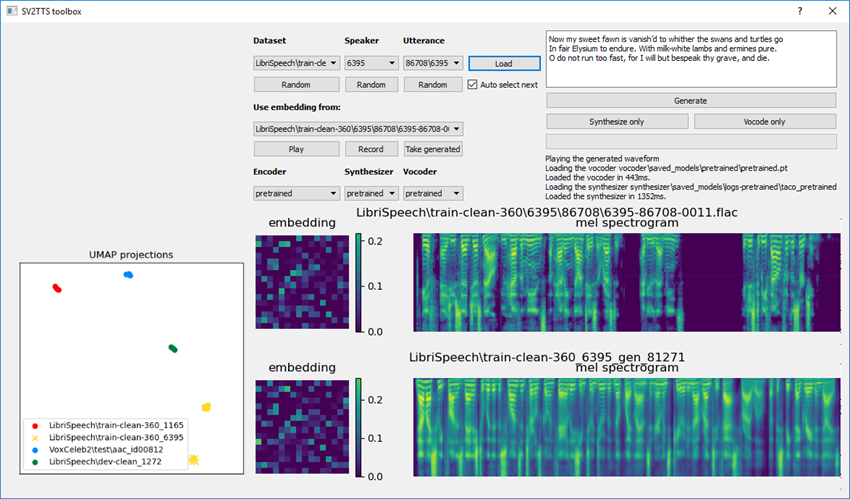
<https://github.com/CorentinJ/Real-Time-Voice-Cloning>

اگر مخزن در حال حاضر در زمان مطالعه غیر قابل دسترسی است، باید به زودی باز شود.

متن‌باز کردن پروژه به معنای نمایش یک رابط تمیز برای هر مدل، هم برای آموزش و هم برای استنتاج، و هم برای مستندسازی کدها و رویه‌ها است. ما معتقدیم که تلاش ارزش آن را دارد. انگیزه‌های ما عبارت‌اند از:

* برای بهبود اجرای خود با مشارکت کاربران
* برای آشنایی با مدیریت مخازن منبع‌باز
* برای اطمینان از تکرارپذیری نتایج
* به کاربران این امکان را می‌دهد تا با داده، زمان و تلاش کم، شبیه‌سازی صوتی را روی سخت‌افزار مصرف‌کننده انجام دهند.

برای آخرین نکته، هدف ما توسعه یک رابط گرافیکی است که به کاربران اجازه می‌دهد تا به‌سرعت و بدون نیاز به مطالعه اولیه آن چارچوب را به دست آورند. ما آن را جعبه ابزار \SV2TTS می‌نامیم. رابط جعبه ابزار را می‌توان در شکل 21 مشاهده کرد. این جعبه ابزار در پایتون با رابط گرافیکی Qt4 نوشته شده است و بنابراین کراس پلتفرم است.



**شکل 21- رابط جعبه ابزار. SV2TTS این تصویر به بهترین وجه بر روی یک پشتیبانی دیجیتال مشاهده می‌شود.**

کاربر با انتخاب یک فایل صوتی گفته از هر یک از مجموعه‌داده‌های ذخیره شده در دیسک خود شروع می‌کند. جعبه ابزار چندین مجموعه‌داده گفتاری محبوب را مدیریت می‌کند و می‌تواند برای افزودن موارد جدید سفارشی شود. علاوه بر این، کاربر همچنین می‌تواند گفته‌ها را ضبط کند تا صدای خود را شبیه‌سازی کند.

هنگامی که یک گفته بارگیری شد، جاسازی آن محاسبه می‌شود و پیش‌بینی‌های UMAP به طور خودکار به‌روز می‌شوند. طیف‌نگار mel بیان رسم شده است (ردیف وسط سمت راست)، اما این صرفاً برای مرجع است، زیرا برای محاسبه چیزی استفاده نمی‌شود. همچنین بردار تعبیه را با نمودار حرارتی (در سمت چپ طیف‌نگار) رسم می‌کنیم. توجه داشته باشید که embeddingها بردارهای یک‌بعدی هستند و بنابراین شکل مربع هیچ معنای ساختاری در مورد مقادیر جاسازی ندارد. ترسیم تعبیه‌ها نشانه‌های بصری در مورد اینکه چگونه دو جاسازی متفاوت هستند، می‌دهد.

هنگامی که یک جاسازی محاسبه شده است، می‌توان از آن برای تولید یک طیف‌نگار استفاده کرد. کاربر می‌تواند هر متن دلخواه (بالا سمت راست رابط) را برای ترکیب بنویسد. به‌عنوان یادآوری، علائم نگارشی توسط مدل ما پشتیبانی نمی‌شود و نادیده گرفته می‌شود. برای تنظیم عروض گفتار تولید شده، کاربر باید شکاف‌های خطی را بین قسمت‌هایی که باید به‌صورت جداگانه سنتز شوند وارد کند. سپس طیف‌نگار کامل الحاق آن بخش‌ها است. سنتز یک طیف‌نگار آن را در سمت راست پایین رابط نمایش می‌دهد. ترکیب جملات یکسان چندین بار خروجی‌های متفاوتی را به همراه خواهد داشت.

در نهایت، کاربر می‌تواند بخش مربوط به طیف‌نگار سنتز شده را با کد صوتی تولید کند. نوار بارگیری پیشرفت تولید را نشان می‌دهد. پس از اتمام، جاسازی گفتار سنتز شده تولید می‌شود (در سمت چپ طیف‌نگار سنتز شده) و همچنین با UMAP پیش‌بینی می‌شود. کاربر آزاد است که آن جاسازی را به‌عنوان مرجع برای نسل بعدی در نظر بگیرد.

1. **نتیجه‌گیری**

ما چارچوبی برای شبیه‌سازی صدای بلادرنگ ایجاد کردیم که اجرای عمومی نداشت. ما نتایج را با وجود برخی عروض‌های غیرطبیعی رضایت‌بخش می‌دانیم، و توانایی شبیه‌سازی صدا در چارچوب نسبتاً خوب است، اما با روش‌هایی که از زمان گفتار مرجع بیشتری استفاده می‌کنند، هم‌تراز نیست. ما امیدواریم که چارچوب خود را فراتر از محدوده این پایان‌نامه بهبود بخشیم، و احتمالاً برخی از پیشرفت‌های جدیدتر در این زمینه را که در زمان نگارش انجام شده است، پیاده‌سازی کنیم. ما بر این باوریم که در آینده‌ای نزدیک اشکال قوی‌تری از شبیه‌سازی صدا در دسترس خواهد بود.

**منابع**

Sercan Arik, Gregory Diamos, Andrew Gibiansky, John Miller, Kainan Peng, Wei  
Ping, Jonathan Raiman, and Yanqi Zhou. Deep voice 2: Multi-speaker neural textto-speech, 2017.  
Sercan O. Arik, Jitong Chen, Kainan Peng, Wei Ping, and Yanqi Zhou. Neural voice  
cloning with a few samples, 2018.  
Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation  
by jointly learning to align and translate. CoRR, abs/1409.0473, 2014. URL http:  
//arxiv.org/abs/1409.0473.  
Mary E. Beckman and Gayle Ayers Elam. Guidelines for tobi labelling, 03 1997.  
Kyunghyun Cho, Bart van Merrienboer, C¸ aglar G¨ul¸cehre, Fethi Bougares, Holger  
Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoderdecoder for statistical machine translation. CoRR, abs/1406.1078, 2014. URL http:  
//arxiv.org/abs/1406.1078.  
Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, KyungHyun Cho, and Yoshua  
Bengio. Attention-based models for speech recognition. CoRR, abs/1506.07503,  
2015. URL http://arxiv.org/abs/1506.07503.  
J. S. Chung, A. Nagrani, and A. Zisserman. Voxceleb2: Deep speaker recognition. In  
INTERSPEECH, 2018.  
Y. Ephraim and D. Malah. Speech enhancement using a minimum mean-square error  
log-spectral amplitude estimator. IEEE Transactions on Acoustics, Speech, and  
Signal Processing, 33(2):443{445, April 1985. ISSN 0096-3518. doi: 10.1109/TASSP.  
1985.1164550.  
Y Fan, Yuang Qian, Feng-Long Xie, and Frank Soong. Tts synthesis with bidirectional  
lstm based recurrent neural networks. pages 1964{1968, 01 2014.  
Xavi Gonzalvo, Siamak Tazari, Chun-an Chan, Markus Becker, Alexander Gutkin,  
and Hanna Silen. Recent advances in google real-time hmm-driven unit selection  
synthesizer. In Interspeech, pages 2238{2242, 2016.  
D. Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. IEEE Transactions on Acoustics, Speech, and Signal Processing, 32(2):236{  
243, April 1984. ISSN 0096-3518. doi: 10.1109/TASSP.1984.1164317.  
K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda. The effect of neural networks in  
statistical parametric speech synthesis. In 2015 IEEE International Conference on  
Acoustics, Speech and Signal Processing (ICASSP), pages 4455{4459, April 2015.  
doi: 10.1109/ICASSP.2015.7178813.

Georg Heigold, Ignacio Moreno, Samy Bengio, and Noam Shazeer. End-to-end textdependent speaker verification. CoRR, abs/1509.08062, 2015. URL http://arxiv.  
org/abs/1509.08062.  
S. Imai. Cepstral analysis synthesis on the mel frequency scale. In ICASSP ’83. IEEE  
International Conference on Acoustics, Speech, and Signal Processing, volume 8,  
pages 93{96, April 1983. doi: 10.1109/ICASSP.1983.1172250.  
Ye Jia, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen,  
Patrick Nguyen, Ruoming Pang, Ignacio Lopez-Moreno, and Yonghui Wu. Transfer  
learning from speaker verification to multispeaker text-to-speech synthesis. CoRR,  
abs/1806.04558, 2018. URL http://arxiv.org/abs/1806.04558.  
Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande,  
Edward Lockhart, Florian Stimberg, Aaron van den Oord, Sander Dieleman, and  
Koray Kavukcuoglu. Efficient neural audio synthesis, 2018.  
Leland McInnes and John Healy. Umap: Uniform manifold approximation and projection for dimension reduction. 02 2018.  
A. Nagrani, J. S. Chung, and A. Zisserman. Voxceleb: a large-scale speaker identification dataset. In INTERSPEECH, 2017.  
Sharan Narang, Gregory F. Diamos, Shubho Sengupta, and Erich Elsen. Exploring  
sparsity in recurrent neural networks. CoRR, abs/1704.05119, 2017. URL http:  
//arxiv.org/abs/1704.05119.  
Tom Le Paine, Pooya Khorrami, Shiyu Chang, Yang Zhang, Prajit Ramachandran,  
Mark A. Hasegawa-Johnson, and Thomas S. Huang. Fast wavenet generation algorithm, 2016.  
V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. Librispeech: An asr corpus  
based on public domain audio books. In 2015 IEEE International Conference on  
Acoustics, Speech and Signal Processing (ICASSP), pages 5206{5210, April 2015.  
doi: 10.1109/ICASSP.2015.7178964.  
Y. Qian, Y. Fan, W. Hu, and F. K. Soong. On the training aspects of deep neural  
network (dnn) for parametric tts synthesis. In 2014 IEEE International Conference  
on Acoustics, Speech and Signal Processing (ICASSP), pages 3829{3833, May 2014.  
doi: 10.1109/ICASSP.2014.6854318.  
Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly,  
Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A.  
Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural TTS synthesis by conditioning wavenet on mel spectrogram predictions. CoRR, abs/1712.05884, 2017.  
URL http://arxiv.org/abs/1712.05884.  
S. Shirali-Shahreza and G. Penn. Mos naturalness and the quest for human-like speech.  
In 2018 IEEE Spoken Language Technology Workshop (SLT), pages 346{352, Dec  
2018. doi: 10.1109/SLT.2018.8639599.

Marilyn Strathern. ‘improving ratings’: audit in the british university system. European Review, 5(3):305{321, 1997. doi: 10.1002/(SICI)1234-981X(199707)5:3h305::  
AID-EURO184i3.0.CO;2-4.  
K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech  
parameter generation algorithms for hmm-based speech synthesis. In 2000 IEEE  
International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100), volume 3, pages 1315{1318 vol.3, June 2000. doi:  
10.1109/ICASSP.2000.861820.  
Yoshihiko; Toda Tomoki; Zen Heiga; Yamagishi Junichi; Oura Keiichiro Tokuda, Keiichi; Nankaku. Speech synthesis based on hidden markov models. Proceedings of the  
IEEE, 101, 05 2013. doi: 10.1109/JPROC.2013.2251852.  
A¨aron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals,  
Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu.  
Wavenet: A generative model for raw audio. CoRR, abs/1609.03499, 2016. URL  
http://arxiv.org/abs/1609.03499.  
A¨aron van den Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George van den Driessche, Edward Lockhart, Luis C. Cobo, Florian Stimberg, Norman Casagrande, Dominik Grewe, Seb Noury, Sander Dieleman,  
Erich Elsen, Nal Kalchbrenner, Heiga Zen, Alex Graves, Helen King, Tom Walters,  
Dan Belov, and Demis Hassabis. Parallel wavenet: Fast high-fidelity speech synthesis. CoRR, abs/1711.10433, 2017. URL http://arxiv.org/abs/1711.10433.  
Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. Journal  
of Machine Learning Research, 9:2579{2605, 2008. URL http://www.jmlr.org/  
papers/v9/vandermaaten08a.html.  
Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end  
loss for speaker verification, 2017.  
Yuxuan Wang, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep  
Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc V. Le, Yannis  
Agiomyrgiannakis, Rob Clark, and Rif A. Saurous. Tacotron: A fully end-to-end  
text-to-speech synthesis model. CoRR, abs/1703.10135, 2017. URL http://arxiv.  
org/abs/1703.10135.  
Takayoshi Yoshimura, Takashi Masuko, Keiichi Tokuda, Takao Kobayashi, and  
Tadashi Kitamura. Speaker interpolation in hmm-based speech synthesis system.  
In EUROSPEECH, 1997.  
Takayoshi Yoshimura, Keiichi Tokuda, Takashi Masuko, Takao Kobayashi, and  
Tadashi Kitamura. Simultaneous modeling of spectrum, pitch and duration in hmmbased speech synthesis. In EUROSPEECH, 1999.

H. Zen, A. Senior, and M. Schuster. Statistical parametric speech synthesis using deep  
neural networks. In 2013 IEEE International Conference on Acoustics, Speech and  
Signal Processing, pages 7962{7966, May 2013. doi: 10.1109/ICASSP.2013.6639215.  
Heiga Zen, Yannis Agiomyrgiannakis, Niels Egberts, Fergus Henderson, and Przemyslaw Szczepaniak. Fast, compact, and high quality LSTM-RNN based statistical  
parametric speech synthesizers for mobile devices. CoRR, abs/1606.06061, 2016.  
URL http://arxiv.org/abs/1606.06061.  
Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J. Weiss, Ye Jia, Zhifeng Chen, and  
Yonghui Wu. Libritts: A corpus derived from librispeech for text-to-speech. CoRR,  
abs/1904.02882, 2019. URL http://arxiv.org/abs/1904.02882.  
Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of  
pruning for model compression, 2017.