



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

پایان نامه کارشناسی ارشد
گرایش هوش مصنوعی

عنوان

تولید دنباله با استفاده از شبکه‌های مقابله‌ای

نگارش

احسان منتهائی

استاد راهنما

دکتر مهدیه سلیمانی

زمستان ۱۳۹۷

دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

پایان نامه کارشناسی ارشد

تولید دنباله با استفاده از شبکه‌های مقابله‌ای

نگارش: احسان منتهائی

امضاء:

دکتر مهدیه سلیمانی

استاد راهنما:

امضاء:

دکتر حسین صامتی

استاد ممتحن داخلی:

امضاء:

دکتر عمادالدین فاطمی‌زاده

استاد ممتحن خارجی:

چکیده

دنباله‌ها بخش زیادی از اطلاعات دنیای واقعی را تشکیل می‌دهند، که بارزترین نمونه‌ی آن زبان‌های طبیعی است. بسیاری از ساختارهای مهم نیز قابلیت مدل شدن به عنوان دنباله را داشته و داده‌هایی مثل ساختار مولکول، گراف و نت‌های موسیقی را نیز می‌توان به عنوان دنباله در نظر گرفت. از سوی دیگر تولید دنباله‌های جدید و بامعنی در هر حوزه، موضوعی با اهمیت و کاربردی است، مثلاً در ترجمه‌ی زبان طبیعی یا کشف ساختار داروی جدید با ویژگی‌های خاص، مساله‌ی تولید دنباله وجود دارد. با این وجود مشکلات مهم زیادی برای حل مساله‌ی تولید دنباله مطرح است، از جمله این مشکلات می‌توان به گسسته بودن جنس داده‌ها و انتخاب تابع هدف مناسب اشاره کرد. روش‌های پایه‌ی ارائه‌شده در این حوزه، دارای مشکلاتی نظیر اُریبی مواجهه میان زمان آموزش و آزمون و تابع هدف نامناسب هستند، از این رو نیاز به روش‌های جدید در این حوزه احساس می‌شود.

در چند سال اخیر پیشرفت‌های زیادی در حوزه‌ی تولید تصویر به وسیله‌ی شبکه‌های مولد مقابله‌ای انجام شده است. همین موضوع باعث شده که استفاده از شبکه‌های مولد مقابله‌ای در تولید دنباله‌ها نیز به تازگی مورد توجه قرار گیرد. اما به دلیل گسسته بودن جنس دنباله‌ها، این امر به سادگی میسر نبوده و برای حل آن نیاز به استفاده از راهکارهایی مثل یادگیری تقویتی و استفاده از تقریب وجود دارد. به علاوه ناپایداری شبکه‌های مولد مقابله‌ای باعث ایجاد چالش‌های جدید و زیاد شدن پیچیدگی مساله می‌شود.

در این پژوهش، با بیان رویکردی جدید و مبتنی بر ایده‌ی شبکه‌های مولد مقابله‌ای، به ارائه‌ی روشی برای حل مساله‌ی تولید دنباله با رویکردی تکرار شونده می‌پردازیم، به طوری که مدل در هر گام الگوریتم، با آموزش در مقابل نمونه‌های تولیدی خودش بهبود می‌یابد. اساس روش پیشنهادی تخمین نسبت چگالی احتمال بوده و با این رویکرد روشی بدون مشکل در برابر گسستگی دنباله‌ها ارائه شده است. راهکار ارائه شده نسبت به روش‌های شبکه‌های مولد مقابله‌ای در حوزه‌ی دنباله، آموزشی پایدارتر دارد؛ هم‌چنین مشکل اُریبی مواجهه نیز در روش پیشنهادی وجود ندارد.

از آنجا که ارزیابی مدل‌های مولد خود چالشی مورد تحقیق است، در بخش دیگری از پایان‌نامه به بررسی معیارهای موجود پرداخته و با ارائه سه نحوه‌ی ارزیابی جدید، سعی در رفع مشکل معیارهای موجود و بهره بردن از نتایج پژوهش‌های مربوط به حوزه‌ی تصویر داریم. روش‌های ارزیابی پیشنهادی برخلاف معیارهای پیشین که فقط کیفیت نمونه‌ها را بررسی می‌کنند، همزمان به تنوع نمونه‌های تولیدی در کنار کیفیت اهمیت می‌دهند. آزمایش‌های این پژوهش نشان‌دهنده‌ی برتری روش پیشنهادی در مقابل روش‌های پیشین است.

کلمات کلیدی: شبکه‌های مقابله‌ای، ایجاد دنباله، شبکه‌های عصبی، یادگیری عمیق

فهرست مطالب

۱	مقدمه	۱
۱-۱	تعریف مساله	۱
۲-۱	اهمیت و کاربرد	۲
۳-۱	یادگیری مقابله‌ای	۳
۱.۳-۱	دلیل توجه به یادگیری مولدمقابله‌ای	۳
۴-۱	چالش‌ها	۵
۱.۴-۱	مشخص نبودن معیار ارزیابی مناسب	۵
۲.۴-۱	مشخص نبودن تابع هزینه مناسب	۵
۳.۴-۱	ناهماهنگی آموزش و آزمون	۶
۴.۴-۱	مشکل انتقال گرادیان	۷
۵.۴-۱	ناپایداری آموزش مقابله‌ای	۸
۵-۱	هدف پژوهش	۹
۶-۱	ساختار پایان‌نامه	۱۰
۲	پژوهش‌های پیشین	۱۱
۱-۲	مقدمه	۱۱
۱.۱-۲	نمادها	۱۲
۲-۲	مدل‌های مولد	۱۳
۱.۲-۲	مدل مولد بازگردنده مبتنی بر نمونه‌گیری	۱۳
۲.۲-۲	مدل‌های مولد مبتنی بر فضای نهان	۱۴

۱۷	۳-۲ مدل‌های دسته‌بند
۱۷	۱.۳-۲ مدل دسته‌بند بازگردنده
۱۷	۲.۳-۲ مدل دسته‌بند پیچشی
۱۸	۴-۲ یادگیری مبتنی بر بیشینه درست‌نمایی
۱۸	۱.۴-۲ روش جبر معلم
۱۹	۲.۴-۲ روش نمونه برداری زمان‌بندی شده
۲۱	۳.۴-۲ روش‌های مبتنی بر خودرمزگذار وردشی
۲۳	۵-۲ شبکه‌های مولد مقابله‌ای
۲۴	۱.۵-۲ بررسی تئوری
۲۵	۶-۲ یادگیری مقابله‌ای با استفاده از Gumbel Softmax
۲۵	۱.۶-۲ تکنیک تنظیم مجدد برای توزیع دسته‌ای
۲۶	۲.۶-۲ تقریبی مشتق‌پذیر برای نمونه‌گیری از توزیع دسته‌ای
۲۷	۳.۶-۲ جزئیات روش
۲۹	۷-۲ یادگیری مقابله‌ای با استفاده از فضای ویژگی
۲۹	۱.۷-۲ روش جبر استاد
۳۱	۲.۷-۲ روش TextGAN
۳۲	۸-۲ یادگیری مقابله‌ای با استفاده از یادگیری تقویتی
۳۳	۱.۸-۲ روش SeqGAN
۳۵	۲.۸-۲ روش ORGAN
۳۶	۳.۸-۲ روش RankGAN
۳۸	۴.۸-۲ روش LeakGAN
۴۰	۵.۸-۲ روش MaliGAN
۴۲	۹-۲ یادگیری مقابله‌ای با رویکرد تولید دنباله‌ی «توزیع کلمه»
۴۳	۱۰-۲ جمع‌بندی
۴۵	۳ معیارهای ارزیابی پیشین و پیشنهادی
۴۵	۱-۳ مقدمه

۴۶	۲-۳	معیارهای مبتنی بر احتمال مدل
۴۶	۱.۲-۳	منفی لگاریتم درست‌نمایی
۴۷	۲.۲-۳	Perplexity
۴۷	۳-۳	معیارهای مبتنی بر داده‌ی مصنوعی
۴۷	۱.۳-۳	منفی لگاریتم درست‌نمایی پیشگو
۴۸	۲.۳-۳	معیار پیشنهادی - فاصله با پیشگو
۵۰	۳.۳-۳	گرامر مستقل از متن
۵۰	۴-۳	معیارهای مبتنی بر n -گرام
۵۱	۱.۴-۳	BLEU
۵۲	۲.۴-۳	Self-BLEU
۵۲	۳.۴-۳	معیار پیشنهادی - MS-Jaccard
۵۳	۵-۳	معیار پیشنهادی - Fréchet BERT Distance
۵۴	۶-۳	جمع‌بندی

۵۶ مراجع

۶۲ واژه‌نامه فارسی به انگلیسی

۶۴ واژه‌نامه انگلیسی به فارسی

فهرست شکل‌ها

- ۱-۱ نتایج یکی از آخرین پژوهش‌های انجام شده در حوزه‌ی تصویر با کمک شبکه‌های مولد مقابله‌ای ۴
- ۲-۱ مقایسه رفتار بیشینه درست‌نمایی و معکوس فاصله‌ی ۶
- ۳-۱ نمایش یک نمونه تابع از مقدار پیوسته‌ی دو بعدی به یک مقدار گسسته ۸
- ۴-۱ نمایش ناپایداری آموزش شبکه‌های مولد مقابله‌ای و پایدارکردن آن ۹
- ۱-۲ مدل مولد بازگردنده مبتنی بر نمونه‌گیری ۱۴
- ۲-۲ دو نمونه مدل بازگردنده‌ی مبتنی بر فضای نهان که به صورت غیرقطعی بعد از ورودی گرفتن نمونه را ایجاد می‌کنند. ۱۵
- ۳-۲ مدل بازگردنده‌ی مبتنی بر فضای نهان که به صورت قطعی بعد از ورودی گرفتن نمونه را ایجاد می‌کنند. ۱۶
- ۴-۲ ساختار شبکه‌ی رمزگشای پیچشی متسع شده ۱۷
- ۵-۲ ساختار شبکه‌ی پیچشی یک بعدی به عنوان دسته‌بند ۱۸
- ۶-۲ روش جبر معلم در دو فاز آموزش و آزمون ۱۹
- ۷-۲ نمای کلی روش نمونه‌برداری زمان‌بندی شده ۲۰
- ۸-۲ نمای کلی شبکه‌ی خودرمزگذار وردشی با رمزگذار و رمزگشای LSTM برای یک نمونه ورودی . ۲۲
- ۹-۲ ساختار کلی شبکه‌ی خودرمزگذار وردشی با رمزگذار LSTM و رمزگشای پیچشی ۲۳
- ۱۰-۲ نمای کلی روش یادگیری مقابله‌ای ۲۴
- ۱۱-۲ نمایش مشکل محاسبه گرادیان به دلیل وجود عملگر تصادفی و حل این مشکل توسط Gumbel Softmax ۲۶
- ۱۲-۲ نمایش اثر پارامتر τ بر روی تقریب Gumbel Softmax ۲۷
- ۱۳-۲ نمای کلی یک نمونه روش که از Gumbel Softmax برای حل مشکل انتقال گرادیان استفاده کرده است. ۲۸

۲۹	۱۴-۲ نمایش وضعیت مخفی شبکه بازگردنده در دو بعد به وسیله‌ی T-SNE
۳۰	۱۵-۲ ساختار کلی نحوه‌ی آموزش در روش جبر استاد
۳۲	۱۶-۲ نمای کلی روش TextGAN
۳۴	۱۷-۲ تخمین پاداش میانی دنباله با روش جستجوی مونت کارلو
۳۶	۱۸-۲ نمایش ساختار کلی نحوه‌ی آموزش در روش RankGAN
۴۰	۱۹-۲ ساختار کلی روش LeakGAN
۴۴	۲۰-۲ نمایش زمانی روش‌های پیشین ارائه‌شده در حوزه‌ی تولید دنباله

فهرست جدول‌ها

۱-۲ جدول نمادها ۱۲

۱-۳ مقیاس‌ی معیارهای موجود و پیشنهاد شده ۵۵

فصل ۱

مقدمه

مطالب این فصل

۱-۱	تعریف مساله	۱
۲-۱	اهمیت و کاربرد	۲
۳-۱	یادگیری مقابله‌ای	۳
۴-۱	چالش‌ها	۵
۵-۱	هدف پژوهش	۹
۶-۱	ساختار پایان‌نامه	۱۰

در این فصل ابتدا به معرفی مساله‌ی تولید دنباله می‌پردازیم، اهمیت و کاربردهای این مساله را مورد بررسی قرار می‌دهیم. پس از آن به صورت مختصر درباره‌ی یادگیری مقابله‌ای^۲ و دلیل توجه به آن توضیحاتی می‌دهیم. سپس چالش‌هایی ذاتی مساله، چالش‌هایی که در روش‌های پایه برای حل مساله وجود داشته و چالش‌های ایجاد شده با استفاده از یادگیری مقابله‌ای بیان می‌شود و فصل حاضر را با تشریح ساختار پایان‌نامه و فصول مختلف آن به پایان می‌بریم.

۱-۱ تعریف مساله

در این پژوهش هدف تولید دنباله‌های گسسته با آموزش بر روی نمونه‌هایی که از این دنباله داریم است. دنباله‌ی گسسته به معنی تعدادی متغیر گسته است که ترتیب دارند و گسسته بودن دنباله به این معنی است که هر مقدار دنباله مقادیر محدودی می‌گیرد. در این پژوهش داده‌ی گسسته از نوع دسته‌ای^۳ مدنظر است (در مقابل داده‌های عددی مثل مجموعه‌ی اعداد صحیح). در داده‌های دسته‌ای همه‌ی مقادیر ممکن هم‌فاصله است، از این رو معمولاً این مقادیر به صورت بردار یک-فعال^۴ کد می‌شوند. هر یک از این مقادیر در دنباله را کلمه می‌نامیم.

^۲ Adversarial

^۳ Categorical

^۴ One-hot

۳-۱ یادگیری مقابله‌ای

شبکه‌های مولد مقابله‌ای^۱ [۶] روشی برای آموزش مدل مولد^۲ است بر این مبنا که شبکه‌هایی با هم رقابت می‌کنند و به نوعی با هم بازی انجام می‌دهند؛ نتیجه‌ی این رقابت آموزش مدل مولد است. روش‌های آموزش مبتنی بر این ایده را یادگیری مقابله‌ای می‌نامیم. در ادامه به نحوه‌ی کار روش شبکه‌های مولد مقابله‌ای معرفی شده در [۶] می‌پردازیم و سپس چند دلیل برای اهمیت یادگیری مقابله‌ای ذکر می‌کنیم.

در شبکه‌های مولد مقابله‌ای برای آموزش مدل مولد از شبکه‌ی دومی به عنوان تمیزدهنده^۳ استفاده می‌شود، این شبکه یک دسته‌بند^۴ دو دسته‌ای است. روال آموزش به این صورت است که تمیزدهنده بین دو مجموعه داده‌ی واقعی و داده‌های تولید شده توسط مولد دسته‌بندی انجام می‌دهد، سپس شبکه‌ی مولد به سمتی سوق داده می‌شود که تمیزدهنده را به اشتباه بیاندازد و این دو مرحله به صورت تکراری انجام می‌شود. در این روش آموزش تمیزدهنده مشابه یک دسته‌بند عادی انجام می‌شود. از آنجا که شبکه‌ی تمیزدهنده نسبت به ورودی مشتق‌پذیر است برای آموزش مدل مولد، گرادینان از شبکه‌ی تمیزدهنده وارد شده و پارامترهای مدل مولد را به‌روزرسانی می‌کند، به عبارت دیگر شبکه‌ی تمیزدهنده مشابه تابع هزینه برای مولد عمل کرده و مشخص می‌کند با تغییر پارامترهای مولد در چه راستایی تمیزدهنده به اشتباه می‌افتد.

۱.۳-۱ دلیل توجه به یادگیری مولد مقابله‌ای

برای حل مسأله‌ی تولید دنباله توجه زیادی به آموزش مقابله‌ای شده است [۲، ۵، ۷-۱۳]، در ادامه دلایلی برای این موضوع ذکر می‌کنیم.

- پیشرفت‌های چشم‌گیر یادگیری مقابله‌ای در حوزه‌ی تصویر: بعد از معرفی یادگیری مقابله‌ای، این روش در حوزه‌ی تولید تصویر نتایج مرز دانش^۵ را کسب کرده است و هر روزه شاهد پیشرفت‌های بیشتری در این حوزه با کمک یادگیری مقابله‌ای هستیم.
- برای مثال در یکی از جدیدترین پژوهش‌هایی که توسط یک شرکت بزرگ ساخت کارت‌های گرافیکی بر روی تولید تصویر صورت انسان انجام شده است [۱۴]، نتیجه‌ی این پژوهش علاوه بر کسب امتیاز بالایی در معیار ارزیابی مربوط به آن حوزه، موفق به تولید تصاویر بسیار نزدیک به تصاویر واقعی شده است. به طوری که نمونه‌های ارائه شده در مقاله بسیار باکیفیت هستند و مصنوعی بودن آن غیرقابل تشخیص است، در شکل ۱-۱ (آ) دو نمونه از این تصاویر نمایش داده شده است. به علاوه فضای نهان^۶ که بر روی تصاویر صورت به دست آمده با معنی بوده و قابلیت ترکیب تصاویر تولید شده را می‌دهد، برای مثال شکل ۱-۱ (ب) نشان دهنده‌ی چند تصویر ورودی در سطر اول است که با کمک فضای ویژگی به دست آمده، به سبک^۷ تصویر سمت چپ تبدیل شده‌اند. به این ترتیب، به واسطه‌ی عملکرد مناسب این روش‌ها در حوزه‌ی

^۱ Generative Adversarial Networks

^۲ Generative

^۳ Discriminator

^۴ Classifier

^۵ State-of-the-Art

^۶ Latent

^۷ Style



(آ) تصاویر مصنوعی تولید شده



(ب) ادغام تصاویر در فضای ویژگی و تبدیل تصاویر مبدا به سبک تصویر مقصد

شکل ۱-۱: نتایج یکی از آخرین پژوهش‌های انجام شده در حوزه‌ی تصویر با کمک شبکه‌های مولد مقابله‌ای [۱۴]

تولید تصویر، در دو سال اخیر توجه به سمت استفاده از مدل‌های مولد مقابله‌ای در حوزه‌ی تولید متن هم جلب شده است.

- قابلیت استفاده از دسته‌ی بزرگی از فاصله‌ها: اثبات شده است که آموزش به وسیله‌ی شبکه‌های مولد مقابله‌ای، معادل کاهش فاصله‌ی Jensen-Shannon بین توزیع اصلی و توزیع مولد است [۶]. هم‌چنین روش‌های مبتنی بر یادگیری مقابله‌ای برای کاهش فاصله‌های دیگر هم پیشنهاد شده است [۱۵، ۱۶]. این فاصله‌ها روش مستقیم برای استفاده به عنوان تابع هزینه^۱ ندارند و این امر با کمک یادگیری مقابله‌ای محقق شده است.

- کمک به حل بعضی مشکلات موجود در روش‌های پایه: در روش‌های پایه‌ی تولید دنباله مشکلاتی مثل اُریبی مواجهه^۲ وجود دارد که دلیل آن رفتار متفاوت مولد در دو فاز آموزش^۳ و آزمون^۴ است. در ادامه در بخش چالش‌ها بیشتر به این موضوع می‌پردازیم. معمولاً روش یادگیری مقابله‌ای دچار این مشکل نمی‌شوند، زیرا اکثر این روش‌ها فاز آموزش و آزمون یکسانی برای تولید دنباله دارند.

^۱ Cost Function

^۲ Exposure Bias

^۳ Train

^۴ Test

۴-۱ چالش‌ها

در این بخش ابتدا به چالش معیار ارزیابی و تابع هزینه که چالش ذاتی مساله است می‌پردازیم، در ادامه نمونه‌ای از مشکل روش پایه را بیان کرده و در پایان مشکلات استفاده از یادگیری مقابله‌ای را در مساله بیان می‌کنیم.

۱.۴-۱ مشخص نبودن معیار ارزیابی مناسب

در حوزه‌ی مدل‌های مولد معیار ارزیابی مشخصی وجود ندارد و نحوه‌ی ارزیابی این مدل‌ها خود یک چالش است [۱۷]. ارزیابی مدل‌های مولد دنباله هم از این قاعده مستثنی نیست و معیاری استاندارد به جز قضاوت انسانی وجود ندارد. به دلیل اهمیت معیار ارزیابی، فصل ۳ به این موضوع اختصاص داده شده است.

۲.۴-۱ مشخص نبودن تابع هزینه مناسب

مشکلی مشابه در حوزه‌ی انتخاب تابع هزینه مورد استفاده وجود دارد و بخش عمده‌ای از پژوهش‌های انجام شده در حوزه‌ی تولید دنباله به روش‌های آموزش و تابع‌های هزینه اختصاص دارد. تابع هدف^۱ اولیه مورد استفاده در روش‌های پایه مبتنی بر بیشینه درست‌نمایی^۲ است. این تابع هدف به این معنی است که در آموزش مدل، هدف افزایش احتمالی است که مدل به داده‌های آموزش می‌دهد و به سمتی می‌رود که از دید مدل، داده‌های واقعی احتمال بالایی بگیرند؛ ولی اینکه به دنبال مدل‌هایی باشیم که نمونه‌های تولیدی آن در توزیع واقعی داده‌ها (که در دسترس نیست) احتمال بالایی بگیرند، رویکرد منطقی‌تری است. با این وجود راه‌حلی برای استفاده از این تابع هدف وجود ندارد. روش‌های یادگیری مقابله می‌توانند استفاده از تابع هدف‌های بهتر را ممکن کنند [۱۸]. درست‌نمایی مدل را به سمتی می‌برد که به داده‌های آموزش احتمال بالایی دهد، ولی این موضوع را کمتر در نظر می‌گیرد که به نمونه‌های غیر از داده‌های آموزش احتمال کمی نسبت دهد. این ویژگی می‌تواند منجر به رفتاری شود که به نمونه‌ی نامعتبری احتمال بالا نسبت داده شود، این رفتار به نام میانگین-جستجوگری^۳ شناخته می‌شود که در ادامه بیشتر توضیح داده شده است.

رفتار میانگین-جستجوگری

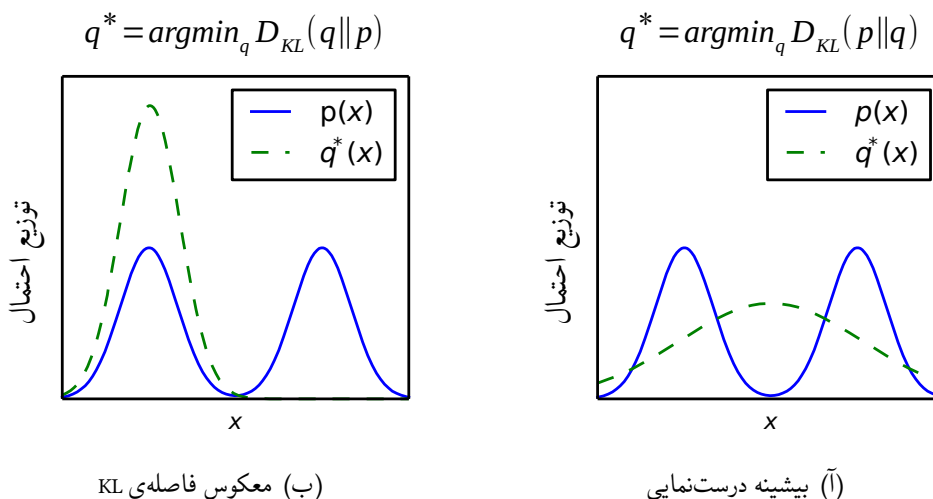
در حالتی که ظرفیت مدل به اندازه‌ی کافی است با در نظر گرفتن تابع درست‌نمایی به عنوان تابع هدف می‌توان توزیع داده‌ی اصلی را یاد گرفت. ولی در حالتی که ظرفیت مدل در مقابل داده‌های آموزش کم است مدل نمی‌تواند توزیع اصلی را کاملاً یاد بگیرد و بسته به تابع هزینه رفتارهای متفاوتی را بروز می‌دهد. رفتاری که با تابع هدف درست‌نمایی بروز داده می‌شود باعث می‌شود توزیع آموزش دیده کل توزیع داده اصلی را در بر بگیرد و در این بین ممکن است به نقاطی از فضا بیشترین احتمال را نسبت دهد که داده‌های اصلی در آنجا احتمال کمی دارند. این رفتار میانگین-جستجوگری نامیده می‌شود. در حوزه‌ی مدل‌های مولد این رفتار به این معنی است که مدل می‌خواهد

^۱ Objective Function

^۲ Maximum Likelihood

^۳ Mean-Seeking

داده‌هایی شبیه به تمام داده‌های آموزش تولید کند، حتی به این قیمت که بعضی از داده‌های تولیدی شبیه به داده‌های آموزش نباشد [۱۸]. به صورت کلی‌تر رفتار میانگین-جستجوگری مربوط به استفاده فاصله‌ی Kullback-Leibler (KL) بین توزیع داده اصلی و توزیع مدل به عنوان تابع هزینه است، در ادامه دلیل این رفتار بیان شده است.



شکل ۲-۱: مقایسه رفتار بیشینه درست‌نمایی و معکوس فاصله‌ی

برای توجیه رفتار میانگین-جستجوگری، حالتی را در نظر بگیرید که این رفتار رخ ندهد یعنی محلی در فضا توزیع احتمال داده‌ی اصلی مقدار دارد و توزیع مدل به آن نقطه احتمال صفر نسبت داده است، در این شرایط تابع هزینه به سمت بی‌نهایت می‌رود و بنابراین در آموزش از چنین حالتی دوری می‌شود و هر نقطه از فضای داده که احتمال دارد، باید مدل هم احتمالی به آن نقطه نسبت دهد.

به عنوان یک مثالی از این رفتار، اگر فرض کنیم داده‌های واقعی توزیع دو قله‌ای گاوسی دارند و خانواده‌ی مدل مولدی که در نظر گرفته‌ایم گاوسی تک قله‌ای باشد، در شکل ۲-۱(آ) نتیجه آموزش با شرایط گفته شده نشان داده شده است، که $p(x)$ توزیع واقعی داده‌ها و $q^*(x)$ بهینه است. نتیجه آموزش با درست‌نمایی در این حالت باعث شده مدل به نقطه‌ای از فضا احتمال زیادی دهد که داده‌های اصلی احتمال کمی دارند. شکل ۲-۱(ب) حالتی است که معکوس فاصله‌ی KL کمینه شده و به سمتی رفته که یکی از قله‌های توزیع اصلی را یاد بگیرد که از یک جهت رفتار مناسب‌تری است. به عبارت دیگر این حالت باعث تولید نمونه‌ی نامعتبر نمی‌شود ولی می‌تواند تنوع در نمونه‌های تولیدشده را محدود کند.

۳.۴-۱ ناهماهنگی آموزش و آزمون

در برخی از روش‌ها نمی‌توان رفتار مشابه در آموزش و آزمون داشت که این مساله باعث می‌شود که خطایی در آموزش ایجاد شود. در ادامه به ذکر یک نمونه از این مشکل می‌پردازیم.

در اکثر روش‌ها برای تولید دنباله، مساله ساده‌سازی می‌شود و به این شکل تبدیل می‌شود که با داشتن یک زیردنباله عنصر بعدی دنباله چگونه باید باشد. بنابراین اگر در گام آموزش این زیردنباله‌ها فقط داده‌های واقعی باشند مشکل

ناهماهنگی^۱ آموزش و آزمون پیش می‌آید، زیرا مدل فقط تصمیم‌گیری با دنباله‌های کاملاً صحیح را آموزش دیده ولی در گام آزمون با زیردنباله‌های تولید خودش مواجه می‌شود، این مشکل اُریبی مواجهه نامیده می‌شود.

۴.۴-۱ مشکل انتقال گرادیان

راهکارهای زیادی برای استفاده از یادگیری مقابله‌ای در حوزه‌ی داده‌های پیوسته پیشنهاد و استفاده شده است، ولی به دلیل مشکل انتقال گرادیان در داده‌های گسسته، مستقیماً امکان استفاده از این روش‌ها در حوزه‌ی داده‌های گسسته وجود ندارد.

روش بهینه‌سازی که برای آموزش شبکه‌های عصبی استفاده می‌شود مبتنی بر محاسبه‌ی گرادیان نسبت به پارامترهای شبکه است و روش‌های یادگیری مقابله‌ای هم نیازمند محاسبه‌ی این گرادیان هستند. برای به‌روزرسانی شبکه مولد لازم است که نمونه‌های تولید شده توسط شبکه‌ی مولد وارد دسته‌بند شده، سپس با کمک دسته‌بند تغییر پارامترهای شبکه‌ی مولد در جهتی که شبکه‌ی دسته‌بند فریب بخورد (یا همان گرادیان)، محاسبه شود. برای اینکه گرادیان نسبت به پارامترهای شبکه‌ی مولد قابل محاسبه باشد، باید تمام عملیات ایجاد نمونه و محاسبه‌ی خروجی دسته‌بند دارای مشتق تعریف شده باشند.

در مساله‌ی تولید دنباله معمولاً در انتهای شبکه‌ی مولد نیاز به نمونه‌گیری از یک توزیع است، این عملیات مشتق تعریف شده‌ای ندارد که این موضوع باعث بروز مشکل انتقال گرادیان می‌شود [۵]. مشکل انتقال گرادیان به معنی عدم توانایی انتقال گرادیان تولید شده توسط دسته‌بند به شبکه‌ی مولد است. این چالش تنها مربوط به حوزه‌ی یادگیری مقابله‌ای نبوده و برای تقریب گرادیان پژوهش‌هایی مثل [۱۹] انجام شده است. در ادامه دلایل این مشکل با جزئیات بیشتر تشریح شده است.

تعریف نشده بودن مشتق عملیات تصادفی

زمانی که متغیر تصادفی z تابعی از θ است و این رابطه تصادفی است، در حالت کلی مشتق z نسبت به θ تعریف نشده است. این مشکل در خود رمزگذار وردشی^۲ هم وجود داشته و راه حل تنظیم مجدد^۳ برای آن پیشنهاد شده است [۱۹].

برای مثال، یک نمونه استفاده از این تکنیک در رابطه با توزیع گوسی است، فرض کنید متغیر تصادفی z به طوری تعریف می‌شود که $z \sim N(f_\theta, 1)$ باشد، در حالت عادی مشتق z به پارامتر θ تعریف نشده است ولی با بازنویسی z به صورت $f_\theta + N(0, 1)$ مشکل حل شده و با حذف قسمت تصادفی از مسیر تبدیل θ به z مشتق تعریف شده است. متأسفانه در شبکه‌های مولد مقابله‌ای برای تولید دنباله، این مشکل به راحتی قابل حل نیست. در بخش ۶-۲ توضیحات بیشتری در این رابطه بیان شده است.

^۱ Inconsistency

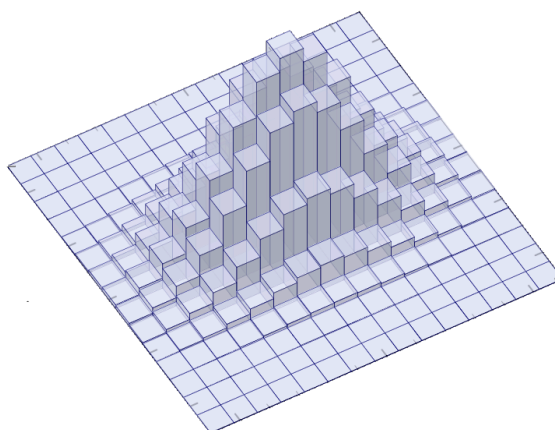
^۲ Variational Autoencoder

^۳ Reparametrization

گسسته بودن خروجی شبکه

مشکل عدم وجود مشتق در مثال گاوسی به این دلیل قابل حل بوده که بعد از تکنیک تنظیم مجدد در مسیر تبدیل θ به z تمام عملیات مشتق‌پذیر است. در حالی که در تولید دنباله گسستگی مقادیر، تعریف دقیق مشتق را ناممکن می‌کند و تنها راهکارهای تقریبی ممکن هستند. شبکه‌ای که در مساله‌ی تولید دنباله گسسته استفاده می‌شود نیازمند تولید داده‌های گسسته است و بنابراین خروجی شبکه‌ی مولد نوع گسسته دارد.

مشکل گسسته بودن خروجی عملیات را از دو دیدگاه می‌توان بیان کرد. دیدگاه اول اینکه برای مشتق این عملیات راه حلی نداریم و راه عملی برای انجام آن وجود ندارد. برای مثال در نظر بگیرید رابطه‌ی $z = \arg \max_i f_i(\theta)$ برای z برقرار است و می‌خواهیم مشتق z نسبت به θ را محاسبه کنیم که روش برای آن نداریم. دیدگاه دوم به صورت بررسی هندسی است، فرض کنید تابع f به صورت $z = f(\theta)$ تعریف شده است و خروجی آن گسسته است، یعنی z دارای N حالت ممکن است. فضایی که پارامترهای θ می‌سازند را در نظر بگیرید، از آنجا که در کل N حالت خروجی وجود دارد، بنابراین فضای پارامترهای θ به N بخش افراز می‌شود. اگر θ نقطه‌ای در یکی از نواحی فضا باشد با تغییرات محلی در مقدار θ خروجی تغییری نمی‌کند و مشتق معنی ندارد [۵]، شکل ۱-۳ توصیفی از چنین فضایی است.



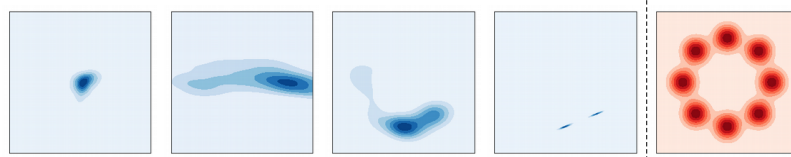
شکل ۱-۳: نمایش یک نمونه تابع از مقدار پیوسته‌ی دو بعدی به یک مقدار گسسته، در چنین فضایی روش‌های مبتنی بر گرادیان برای پیدا کردن نقطه‌ی بیشینه کارساز نیست.

۵.۴-۱ ناپایداری آموزش مقابله‌ای

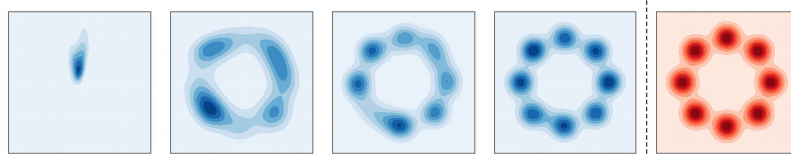
بزرگترین چالشی که در روش‌های آموزش مقابله‌ای وجود دارد ناپایداری آموزش آن است. دو نمونه از این ناپایداری به شرح زیر است:

- آموزش مقابله‌ای می‌تواند به رفتار ناپایداری منجر شود که توزیع مدل مولد بین چند قله از توزیع اصلی به صورت تناوبی جابجا شود که این رفتار چسبیدگی به قله^۱ نام دارد. یک مثال از چسبیدگی به قله در شکل ۱-۴(آ) نشان داده شده است و نشان می‌دهد که در آموزش یک توزیع چندقله‌ای به عنوان توزیع اصلی،

^۱ Mode Collapsing



(آ) آموزش ناپایدار



(ب) آموزش پایدار

شکل ۱-۴: نمایش ناپایداری آموزش شبکه‌های مولد مقابله‌ای و پایدارکردن آن با روش معرفی شده در [۲۰]. تصویر سمت راست توزیع داده‌ی اصلی را نشان می‌دهد و بقیه تصاویر از چپ به راست توزیع مدل مولد در حین آموزش و در گام‌های ۵۰۰۰، ۱۰۰۰۰ و ۲۰۰۰۰ آموزش است.

مدل مولد بین قله‌ها به صورت تناوبی جابجا می‌شود. هم‌چنین شکل ۱-۴ (ب) نشان دهنده‌ی حالتی است که با تکنیک معرفی شده در [۲۰]، این آموزش پایدار شده است.

- مشکل دیگر در تکنیک مطرح‌شده در تنظیم نسبت آموزش مدل مولد و تمیزدهنده است. به این صورت که اگر مدل تمیزدهنده به خوبی آموزش ببیند تمام نمونه‌های مدل مولد را به راحتی تشخیص می‌دهد و این موضوع باعث سخت شدن آموزش مولد می‌شود. مثلاً فرض کنید شبکه‌ی مولد بر اساس امتیازی که از تمیزدهنده می‌گیرد آموزش می‌بیند و در حالتی که تمیزدهنده خیلی قوی‌تر باشد به همه‌ی حرکت‌های خوب مولد امتیاز بسیار کوچکی می‌دهد که این موضوع باعث آموزش ندیدن مولد در عمل می‌شود. در واقع نزدیک نقطه‌ی بهینه برای تمیزدهنده، گرادیان نزدیک صفر است و چندان نمی‌تواند به بهبود مولد کمک کند.

برای پایدار کردن آموزش روش‌هایی ارائه شده است [۲۰-۲۵]، ولی بسیاری از این روش‌ها خاص حوزه‌ی داده‌های پیوسته است و نمی‌توان از آنها در تولید دنباله مستقیماً بهره برد.

۵-۱ هدف پژوهش

با توجه به تعریف مساله در بخش ۱-۱ و چالش‌هایی که برای حل آن وجود دارد، در ادامه قصد داریم روشی برای تولید دنباله ارائه کنیم که بر پایه‌ی شبکه‌های مولد مقابله‌ای است، اما با مشکلاتی نظیر انتقال گرادیان رو به رو نیست. هم‌چنین در جهت ارزیابی موثرتر مدل پیشنهادی، معیارهای سنجش کیفیت را مورد مطالعه‌ی دقیق قرار داده و معیارهای جدیدی را برای این منظور پیشنهاد می‌کنیم.

۱-۶ ساختار پایان‌نامه

در ادامه‌ی مستند حاضر، در فصل دوم به تشریح کارهای پیشین در زمینه‌ی تولید دنباله می‌پردازیم و با بررسی مزایا و معایب این روش‌ها، رویکردهای مختلف حل این مساله را با یکدیگر مقایسه می‌کنیم و در فصل سوم راهکار پیشنهادی برای تولید دنباله را معرفی می‌کنیم. از آنجا که معیار ارزیابی خود موضع چالش برانگیزی در این حوزه است فصل چهارم را به تشریح این معیارها می‌پردازیم و معیارهای جدیدی را پیشنهاد می‌کنیم. با ارائه آزمایش‌های انجام شده در فصل پنجم، به ارزیابی روش پیشنهادی می‌پردازیم. در نهایت با جمع‌بندی مطالب ارائه شده و ارائه پیشنهاداتی برای ادامه پژوهش، پایان‌نامه را به پایان می‌بریم.

فصل ۲

پژوهش‌های پیشین

مطالب این فصل

۱-۲	مقدمه	۱۱
۲-۲	مدل‌های مولد	۱۳
۳-۲	مدل‌های دسته‌بند	۱۷
۴-۲	یادگیری مبتنی بر بیشینه درست‌نمایی	۱۸
۵-۲	شبکه‌های مولد مقابله‌ای	۲۳
۶-۲	یادگیری مقابله‌ای با استفاده از Gumbel Softmax	۲۵
۷-۲	یادگیری مقابله‌ای با استفاده از فضای ویژگی	۲۹
۸-۲	یادگیری مقابله‌ای با استفاده از یادگیری تقویتی	۳۲
۹-۲	یادگیری مقابله‌ای با رویکرد تولید دنباله‌ی «توزیع کلمه»	۴۲
۱۰-۲	جمع‌بندی	۴۳

در این فصل بعد از بیان مدل‌های مولد و دسته‌بند مورد استفاده در حوزه‌ی تولید دنباله، به بیان روش‌های پایه و مشکلات آن‌ها می‌پردازیم. سپس با بیان دقیق‌تر شبکه‌های مولد مقابله‌ای که بر اساس ایده‌ی آموزش مقابله‌ای هستند، روش‌های مبتنی بر یادگیری مقابله‌ای را شرح می‌دهیم. روش‌های یادگیری مقابله‌ای را در چهار دسته در نظر می‌گیریم؛ این دسته‌بندی شامل روش‌های مبتنی بر Gumbel Softmax، روش‌های مبتنی بر فضای ویژگی، روش‌های مبتنی بر یادگیری تقویتی و در آخر روش‌های با رویکرد تولید دنباله‌ی کلمات می‌شود. در پایان با جمع‌بندی این فصل را به اتمام می‌رسانیم.

۱-۲ مقدمه

بیشتر پژوهش‌های پیشین برای استفاده از شبکه‌های مولد مقابله‌ای در تولید دنباله به موضوع رفع مشکل انتقال گرایان (که در بخش ۴.۴-۱ مورد بحث قرار گرفت) اختصاص داشته است. رویکردهای مختلفی برای این منظور

وجود داشته است.

برخی از روش‌ها به صورت تقریبی گرادیان انتقال یافته به مولد را به دست می‌آورند. راه حل دیگر که رایج تر است، تبدیل مساله‌ی آموزش مولد به یک مساله‌ی یادگیری تقویتی است. این روش‌ها با تعیین پاداش برای مولد آموزش را انجام داده و گرادیان شبکه‌ی مولد را از این طریق محاسبه می‌کنند. در این دسته از پژوهش‌ها هدفی دیگر که دنبال می‌شود، افزایش اطلاعات موجود در پاداش است. همچنین برخی از پژوهش‌ها سعی کرده اند با تغییر صورت مساله و تبدیل آن به مساله‌ای پیوسته، تولید دنباله را انجام دهند. در این فصل به هر کدام از این روش‌ها پرداخته می‌شود. در ادامه نماد گذاری معرفی شده که برای یک دست شدن بیشتر فصل در تمام فصل استفاده شده است.

۱.۱-۲ نمادها

در این بخش به بیان قراردادهای و نمادهایی می‌پردازیم که در توضیح روش‌ها از این نمادگذاری استفاده شده است. این قراردادهای و نمادگذاری‌ها در جدول ۱-۲ مشخص شده است.

جدول ۱-۲: جدول نمادها

نماد	توضیحات
کلمه	کوچک‌ترین واحد دنباله کلمه است.
\mathcal{L}	برای نشان دادن تابع هزینه استفاده می‌شود که در صورت ذکر نکردن، آن را کمینه می‌کنیم.
L	طول دنباله‌ها یکسان فرض می‌شود و L نشان دهنده‌ی طول دنباله‌ها است.
V	اندازه‌ی واژگان ^۱ یعنی تعداد حالاتی که هر کلمه‌ی دنباله می‌تواند به خود بگیرد.
N	تعداد نمونه‌های آموزش
$x^{(n)}$	داده‌ی آموزش n ام ($1 \leq n \leq N$)
x_l	کلمه‌ی l از دنباله‌ی x ($1 \leq l \leq L$)
$x_{a:b}$	زیردنباله‌ی شامل کلمات a تا b ام ($1 \leq a < b \leq L$)
$x_{:b}$	زیردنباله‌ی شامل b کلمه‌ی اول ($1 < b \leq L$)
P, P_{data}	نشان دهنده‌ی توزیع اصلی
$p(x)$	تابع چگالی توزیع اصلی
Q, P_{model}	نشان دهنده‌ی توزیع مدل است.
$q(x)$	تابع چگالی توزیع مدل
$D_\phi(x)$	نمایش تمیزدهنده به عنوان یک تابع است.
G_θ	نشان دهنده‌ی شبکه‌ی مولد است.

۲-۲ مدل‌های مولد

در تمامی روش‌های ارائه‌شده، دنباله‌ها به صورت احتمالی توسط مدل مولد مدل‌سازی می‌شوند به طوری که توزیع احتمال دنباله به توزیع‌های ساده‌تری شکسته می‌شود که به صورت زیر است:

$$q(x; \theta) = q_1(x_1; \theta) \prod_{l=2}^L q_l(x_l | x_{1:l-1}; \theta) \quad (1.2)$$

که این عبارت با کمک قاعده‌ی زنجیره‌ای^۱ به دست آمده است و توزیع احتمال دنباله به طول L به توزیع‌های ساده‌تر تک متغیره $q_l(x_l | x_{1:l-1}; \theta)$ شکسته شده است.

لازم به ذکر است که هیچ استقلال‌ی در مدل‌سازی به فرم گفته‌شده در رابطه‌ی ۱.۲ فرض نشده است و این موضوع باعث افزایش قابلیت مدل برای یادگیری توزیع‌های پیچیده و شامل ارتباطات طولانی در دنباله می‌شود [۲۶]. معمولاً توزیع‌های Q_l با استفاده از یک شبکه‌ی عصبی بازگردنده^۲ مدل می‌شود، ولی به تازگی شبکه‌های پیچشی^۳ نیز برای این منظور استفاده شده‌اند.

مدل‌های مولد را می‌توان به دو دسته تقسیم کرد: دسته‌ی اول مولدهایی هستند که با استفاده از نمونه‌گیری، دنباله‌های مختلف تولید می‌کنند. تمام قسمت تصادفی عملیات تولید نمونه در نمونه‌گیری خلاصه شده است. دسته‌ی دوم مولدهایی هستند که برای تولید نمونه در یک فضای نهان، از یک توزیع ساده نمونه‌گیری کرده و سپس به کمک یک تابع تبدیل، این نمونه را از فضای نهان به نمونه‌ی نهایی تبدیل می‌کنند. دسته‌ی اول توزیع دنباله را به صورت صریح مدل می‌کنند، این در حالی است که توزیع در دسته‌ی دوم به شرط فضای نهان مدل می‌شود. در بخش‌های بعد به جزئیات مدل‌ها می‌پردازیم.

۱.۲-۲ مدل مولد بازگردنده مبتنی بر نمونه‌گیری

این مدل‌ها که از پرستفاده‌ترین مدل‌های مولد هستند، بر اساس رابطه‌ی ۱.۲ توزیع دنباله را با کمک یک ساختار بازگردنده مدل می‌کنند. ساختار LSTM [۲۷] به عنوان یکی از انواع مرسوم شبکه‌های بازگردنده استفاده می‌شود. به صورت دقیق‌تر اگر در شبکه‌ی مورد استفاده اندازه‌ی وضعیت مخفی^۴ برابر h ، فضای نهان مورد استفاده برای کلمات e و تعداد کلمات V باشد، علاوه بر شبکه‌ی بازگردنده دو ماتریس دیگر به پارامترها اضافه می‌شود. ماتریس اول برای تبدیل کلمات به بردار ویژگی و دارای اندازه‌ی $V \times e$ است و سطر l -ام در آن شامل بردار ویژگی کلمه‌ی l -ام است. ماتریس دوم برای تبدیل وضعیت مخفی شبکه بازگردنده به توزیع روی کلمات است، اندازه این ماتریس $h \times V$ است و با ضرب وضعیت مخفی در این ماتریس برداری در اندازه‌ی کلمات به دست می‌آید. با اعمال تابع بیشینه‌ی هموار^۵ بر روی این بردار توزیع‌ای دسته‌ای^۶ بر روی کلمات به دست می‌آید. با وجود پرستفاده بودن این معماری، در پژوهشی جدید از نظر عملی و تئوری نشان داده شده است که این مدل‌های دارای محدودیت‌هایی برای مدل کردن دنباله‌هایی مثل دنباله‌ی زبان طبیعی هستند [۲۸].

^۱ Chain Rule

^۲ Recurrent Neural Network

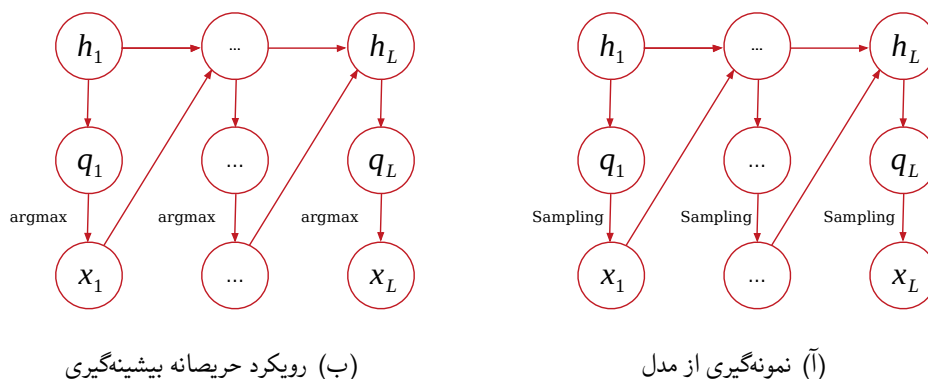
^۳ Convolutional

^۴ Hidden State

^۵ Softmax

^۶ Categorical

نمونه‌گیری از این مدل‌ها به این صورت است که در هر مرحله بخشی از دنباله‌ی تولید شده به مدل وارد می‌شود و مدل توزیعی بر روی کلمه‌ی بعدی ایجاد می‌کند. کلمه‌ی بعدی دنباله با نمونه‌گیری از این توزیع، مشخص شده و این روال تکرار می‌شود. معمولاً کلمه‌ای به عنوان کلمه‌ی «پایان» وجود دارد که هر زمان این کلمه تولید شود به معنی اتمام جمله است. در شکل ۱-۲ (آ) نحوه‌ی نمونه‌گیری از این مدل نمایش داده شده است.



شکل ۱-۲: مدل مولد بازگردنده مبتنی بر نمونه‌گیری

سختی بیشینه گرفتن از مدل

از مشکلات مدل احتمالی تشریح شده، حالتی است که در گام آزمون به جای نمونه‌گیری به بیشینه‌گیری نیاز باشد، این مساله بیش‌تر در مدل‌های شرطی مطرح می‌شود. در این حالت یافتن نمونه با بیشترین احتمال بسیار هزینه‌بر است، زیرا پیچیدگی محاسباتی آن برحسب تعداد عناصر دنباله، نمایی است. به همین دلیل رویکرد ساده‌ای مثل روش حریصانه مورد استفاده قرار گرفته و از هر توزیع Q_n عنصر با بیشترین احتمال انتخاب می‌شود. در شکل ۱-۲ (ب) این روش نمایش داده شده است. می‌دانیم این نحوه‌ی نمونه‌گیری باعث انتخاب نمونه‌ها با احتمال کم‌تر از بیشینه می‌شود زیرا:

$$\prod_{l=1}^L \max_{x_l} q_n(x_l | x_{1:l-1}; \theta) \leq \max_x \prod_{l=1}^L q_n(x_l | x_{1:l-1}; \theta). \quad (2.2)$$

در روشی دیگر برای تقلیل این مشکل، به جای انتخاب یک لغت در هر مرحله، k بهترین دنباله در هر مرحله انتخاب و نگهداری می‌شود که به این روش جستجوی پرتویی^۱ گفته می‌شود [۲۹].

۲.۲-۲ مدل‌های مولد مبتنی بر فضای نهان

مدل‌های مولدی که دارای فضای نهان هستند، قابلیت مدل کردن صریح توزیع را از دست می‌دهند ولی در عوض امکان مشاهده و تغییر ویژگی‌های اصلی دنباله را به کمک فضای نهان دارند. به علاوه در روش‌های مثل خودرمزگذار وردشی^۲، این مدل‌ها لازمه‌ی روش هستند [۳۰].

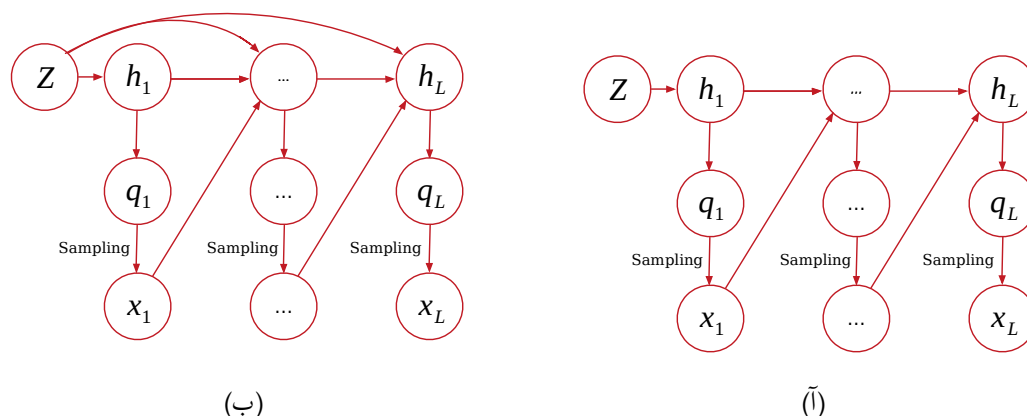
^۱ Beam Search

^۲ Variational Autoencoder

در این مدل‌ها فرایند تولید نمونه به این صورت است که مقدار z از فضای نهان نمونه گرفته می‌شود و مدل مولد به شرط z نمونه را تولید می‌کند. این مدل‌ها بیش‌تر برای استفاده در شبکه‌های مولد مقابله‌ای در حوزه‌ی داده‌های پیوسته استفاده می‌شوند، ولی در حوزه‌ی داده‌های گسسته، استفاده از این مدل‌ها با مشکلاتی همراه است. دو ساختار مطرح شده برای این مدل‌ها را در ادامه توضیح می‌دهیم.

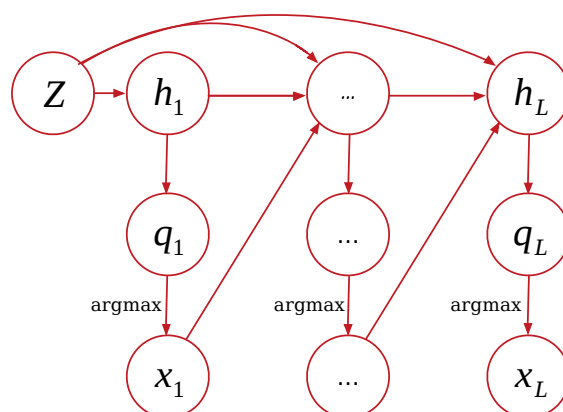
مدل مولد بازگردنده مبتنی بر فضای نهان

این مدل‌ها ساختاری مشابه **مدل مولد بازگردنده مبتنی بر نمونه‌گیری** (بخش ۲-۱.۲) دارند، با این تفاوت که متغیر نهان z به مدل اضافه شده است. در ادامه دو نمونه از این مدل‌ها را توضیح می‌دهیم. ساختاری که برای این شبکه‌ها در [۲۶] پیشنهاد شده، با شکل ۲-۲(آ) نشان داده شده است. تنها تفاوتی که این مدل با **مدل مولد بازگردنده مبتنی بر نمونه‌گیری** دارد، در وضعیت مخفی اولیه‌ی شبکه است و وضعیت مخفی اولیه‌ی شبکه از مقدار نهان z به دست می‌آید. نوع دیگری از این ساختار در شکل ۲-۲(ب) نمایش داده شده است که در آن علاوه بر وضعیت اولیه، متغیر نهان هم به عنوان ورودی به تمام مراحل شبکه‌ی بازگردنده وارد شده است. این ساختارها را مدل‌های غیرقطعی می‌نامیم، زیرا در این مدل‌ها با ثابت بودن مقدار z ، تولید نمونه قطعیت نداشته و مجدداً بعد از ورودی دادن z ، فرایند تصادفی نمونه‌گیری در آن‌ها انجام می‌شود. دو مدل غیرقطعی ارائه شده در پژوهش انجام شده در [۲۶] با روش خودرمزگذار وردشی آموزش دیده و برتری‌ای به هم دیگر ندارند. از نقدهای وارد شده به مدل غیرقطعی این است که در آموزش تمایل به نادیده گرفتن مقدار نهان دارند [۲۶، ۳۱].



شکل ۲-۲: دو نمونه مدل بازگردنده مبتنی بر فضای نهان که به صورت غیرقطعی بعد از ورودی گرفتن نمونه را ایجاد می‌کنند.

در ساختار دیگری که برای مدل‌های بازگردنده مبتنی بر فضای نهان در [۹] پیشنهاد شده است، با مشخص شدن مقدار z ، نمونه به صورت قطعی تولید شده و دیگر عملیات تصادفی در نمونه‌گیری وجود ندارد. همان‌طور که این مدل در شکل ۲-۳ نمایش داده شده است، متغیر نهان به تمام مراحل شبکه وارد می‌شود. نکته‌ای که در مورد این مدل وجود دارد این است که در عمل، مشتق‌ناپذیری $\arg \max$ باعث ایجاد مشکل در بعضی از روش‌های آموزش می‌شود. برای رفع این مشکل، عملیات $\arg \max$ به وسیله‌ی تابع بیشینه‌ی هموار تخمین زده می‌شود. نحوه‌ی این تخمین در بخش ۲-۲.۶ بیان خواهد شد.



شکل ۲-۳: مدل بازگردنده مبتنی بر فضای نهان که به صورت قطعی بعد از ورودی گرفتن نمونه را ایجاد می‌کند.

مدل مولد پیچشی مبتنی بر فضای نهان

نوع دیگری از مدل‌های مبتنی بر فضای داده در [۳۱] استفاده شده است. این مدل رفتار غیرقطعی دارد و بعد از مشخص شدن مقدار نهان z ، نمونه‌ی نهایی به صورت قطعی مشخص نیست. استفاده از شبکه‌ی عصبی پیچشی^۱ به عنوان شبکه‌ی مولد، موضوع کمتر بررسی شده‌ای در حوزه‌ی تولید دنباله است. در ادامه به توضیح این ساختار می‌پردازیم.

شبکه‌ی پیچشی یک بعدی برای تولید دنباله مورد استفاده قرار می‌گیرد. در حالت عادی شبکه‌های پیچشی شرط علی^۲ بودن را ندارند؛ علی بودن به این معنی است که بخشی از شبکه که کلمه‌ی l -ام را پیش‌بینی می‌کند، فقط تابع کلمات ورودی ۱ تا $l-1$ باشد. در گام آموزش، برای جلوگیری از رسیدن شبکه به حالت‌های نامطلوب نیازمند مدلی علی هستیم، به همین دلیل شبکه‌ی پیچشی پیشنهاد شده، نوع علی شبکه‌های پیچشی است.

به علاوه، برای افزایش اندازه‌ی میدان دریافتی^۳ از نوعی از شبکه‌های پیچشی به نام پیچشی متسع شده^۴ استفاده شده است. شبکه‌ی پیچشی متسع شده در کنار افزایش اندازه‌ی میدان دریافتی، هزینه محاسباتی را زیاد نمی‌کند و به این صورت عمل می‌کند که اگر مقدار اتساع^۵ را d بنامیم، پیچش^۶ روی ورودی‌ها به صورتی اعمال می‌شود که ورودی‌ها d درمیان در نظر گرفته می‌شود و $d-1$ ورودی را در هر بخش در نظر نمی‌گیرد. شبکه پیچشی معمولی را می‌توان حالت خاص شبکه‌ی پیچشی متسع شده با مقدار $d=1$ در نظر گرفت [۳۲]. در شکل ۲-۴ این مدل برای حالت چهار ورودی نمایش داده شده که در لایه‌ی اول $d=1$ و در لایه‌ی دوم $d=2$ است. همان‌طور که در شکل مشخص است، مدل در تمام مراحل به مقدار نهان متصل است.

^۱ Convolutional Neural Networks

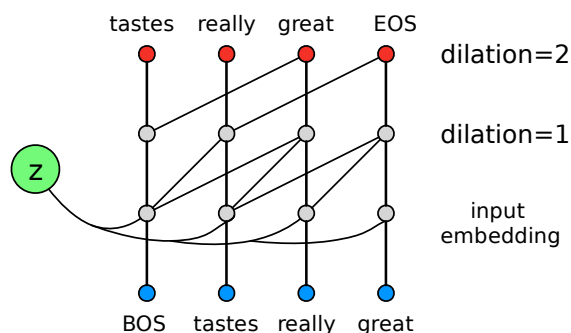
^۲ Causality

^۳ Receptive Field Size

^۴ Dilated Convolution

^۵ Dilation

^۶ Convolution



شکل ۲-۴: ساختار شبکه‌ی رمزگشای پیچشی متسع شده برای چهار ورودی [۳۱]

۳-۲ مدل‌های دسته‌بند

مدل‌های دسته‌بند دو دسته‌ای در روش‌های مبتنی بر یادگیری مقابله‌ای مورد استفاده هستند. این شبکه‌ها دنباله را به عنوان ورودی گرفته و عددی بین صفر و یک را خروجی می‌دهد. دو نوع شبکه‌ی پرکاربرد برای این منظور وجود دارد که در ادامه معرفی شده‌اند.

۱.۳-۲ مدل دسته‌بند بازگردنده

این مدل‌ها یک شبکه‌ی بازگردنده که معمولاً LSTM است، استفاده کرده و بعد از بردن کلمات دنباله به فضای ویژگی، این دنباله‌ی ویژگی‌ها را به مدل بازگردنده وارد کرده و در نهایت برای ساختن مقدار خروجی، تبدیلی خطی بر روی آخرین وضعیت مخفی در نظر گرفته و با اعمال تابع فعال‌ساز سیگموئید خروجی بین صفر و یک ایجاد می‌کنند.

۲.۳-۲ مدل دسته‌بند پیچشی

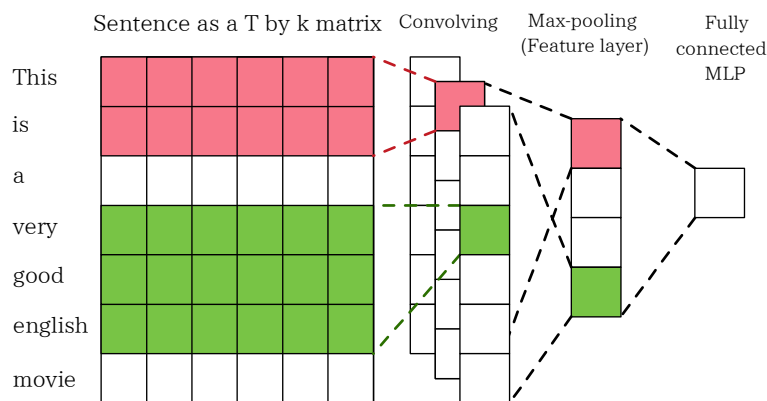
روش دیگر دسته‌بندی، مبتنی بر استفاده از شبکه‌های پیچشی یک بعدی است. همان‌طور که در شکل ۲-۵ نمایش داده شده است، این شبکه‌ها در ابتدا کلمات دنباله را به فضای ویژگی برده و سپس به وسیله‌ی اعمال پیچش‌هایی با فیلترهای با اندازه‌های مختلف، از دنباله ویژگی استخراج می‌کنند. این ویژگی‌ها با طول دنباله رابطه داشته و اندازه‌ی ثابتی ندارند، از این رو در طول دنباله بر روی این ویژگی‌ها با بیشینه‌گیری رای‌گیری^۱ انجام می‌شود و در نهایت به تعداد فیلترها از دنباله ویژگی استخراج می‌شود. در گام نهایی با یک یا چند لایه‌ی تمام‌متصل^۲، این ویژگی‌ها به یک عدد تبدیل شده که به عنوان خروجی دسته‌بند استفاده می‌شود [۳۳].

به بیان ساده‌تر این مدل ویژگی‌هایی را از n -گرام^۳‌های دنباله استخراج می‌کند و مستقل از جایی که این n -گرام‌ها قرار دارند، از روی این ویژگی‌ها خروجی نهایی را پیش‌بینی می‌کند.

^۱ Pooling

^۲ Fully Connected

^۳ n-gram



شکل ۲-۵: ساختار شبکه‌ی پیچشی یک بعدی به عنوان دسته‌بند [۹]

۴-۲ یادگیری مبتنی بر بیشینه درست‌نمایی

بیشینه کردن درست‌نمایی از قدیم جزء روش‌های پراستفاده بوده است. در این بخش بعد از توضیحات اولیه به بررسی روش‌هایی که تابع هزینه‌ی مبتنی بر بیشینه درست‌نمایی دارند، می‌پردازیم. در ابتدا روش جبر معلم^۱ که روش پایه‌ی تولید دنباله محسوب می‌شود، توضیح داده، سپس به بیان مشکل آریبی مواجهه پرداخته و راهکار ارائه شده برای رفع آن را بررسی می‌کنیم. در انتها به بررسی روش‌های خودرمزگذار وردشی می‌پردازیم، که هدفشان علاوه بر تولید دنباله یافتن نمایشی در فضای نهان برای دنباله‌ها است.

در روش بیشینه کردن درست‌نمایی مدلی پارامتری برای تخمین توزیع احتمال ارائه می‌شود، که θ پارامتر آن است. درست‌نمایی مدل به ازای پارامتر θ برابر $\prod_{n=1}^N P_{model}(x^{(n)}; \theta)$ است، که $x^{(i)}$ ها داده‌های آموزش هستند. اصل بیشینه درست‌نمایی به سادگی می‌گوید که باید پارامتر θ به صورتی انتخاب شود که درست‌نمایی بیشینه شود [۳۴]. به بیانی دیگر بیشینه کردن درست‌نمایی هم‌ارز کم کردن فاصله‌ی Kullback-Leibler (KL) بین توزیع داده‌های واقعی و توزیع مدل است:

$$\theta^* = \arg \max_{\theta} D_{KL}(P_{data}(x) || P_{model}(x; \theta)), \quad (3.2)$$

که در عمل به $P_{data}(x)$ دسترسی نداشته و فقط N نمونه از این توزیع داریم، و به همین دلیل به جای استفاده از $P_{data}(x)$ از تقریب $\hat{P}_{data}(x)$ استفاده می‌کنیم. یک توزیع تجربی است که چگالی احتمال فقط روی N نمونه وجود دارد. کمینه کردن فاصله‌ی KL بین $\hat{P}_{data}(x)$ و P_{model} معادل بیشینه درست‌نمایی روی داده‌های آموزش است [۳۴].

۱.۴-۲ روش جبر معلم

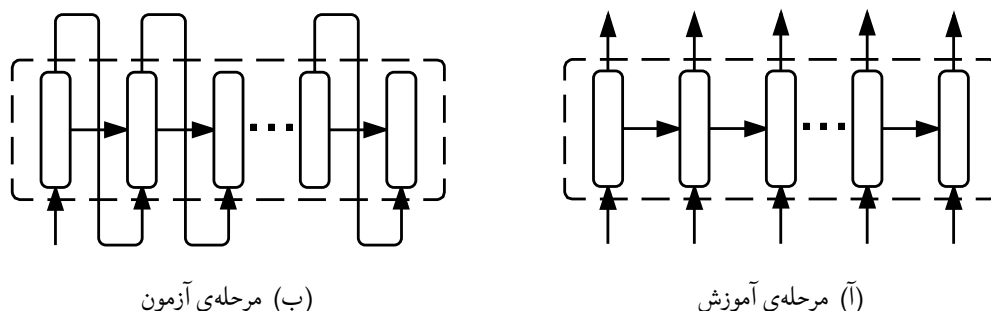
روش جبر معلم پایه‌ای‌ترین راه برای آموزش توزیع دنباله و تولید دنباله‌های جدید است. در روش جبر معلم تابع هزینه براساس بیشینه درست‌نمایی است و معمولاً مدل مولد بازگردنده مبتنی بر نمونه‌گیری (بخش ۲-۱.۲) به

^۱ Teacher Forcing

عنوان مولد استفاده می‌شود. بنابراین تابع هزینه از رابطه ۱.۲ به صورت زیر به دست می‌آید:

$$\mathcal{L} = \sum_{n=1}^N \sum_{l=1}^L \log Q_l(x_l^{(n)} | x_{1:n-1}^{(n)}) \quad (4.2)$$

این تابع هزینه نشان می‌دهد که برای آموزش، زیر دنباله‌ی $x_{1:n}$ به مدل وارد می‌شود و انتظار می‌رود که مدل کلمه‌ی x_{n+1} را پیش‌بینی کند. در شکل ۲-۶ (آ) روال آموزش نشان داده شده است.



شکل ۲-۶: روش جبر معلم در دو فاز آموزش و آزمون [۸]

آرایی مواجهه

همان‌طور که توضیح داده شد، در آموزش روش جبر معلم فقط داده‌های آموزش وارد شبکه می‌شود، بنابراین شبکه به شرط زیر دنباله‌های واقعی و درست، آموزش دیده تا بتواند کلمه‌ی بعدی را پیش‌بینی کند. ولی در فاز آزمون که از شبکه‌ی آموزش دیده نمونه‌گیری می‌کنیم، داده‌های واقعی را نداریم که از آن به عنوان ورودی مدل استفاده شود و راه‌حلی که وجود دارد ورودی دادن دنباله‌ی تولید شده توسط خود مدل است. نحوه‌ی تولید نمونه‌ها در فاز آزمون در شکل ۲-۶ (ب) نمایش داده شده است. بنابراین توزیع ورودی که برحسب آن مدل آموزش دیده در فاز آزمون تغییر کرده است و این تغییر ورودی موجب خطا در خروجی پیش‌بینی شده می‌شود. این خطا به صورت تجمعی در طول دنباله بیشتر شده و باعث کاهش کیفیت نمونه‌های تولیدی می‌شود [۵، ۱۸، ۲۹]. این رفتار باعث تولید دنباله‌هایی می‌شود که دارای کلمات مناسب و با کیفیتی در ابتدای دنباله هستند، اما در کلمات جلوتر کیفیت کاهش می‌یابد [۹].

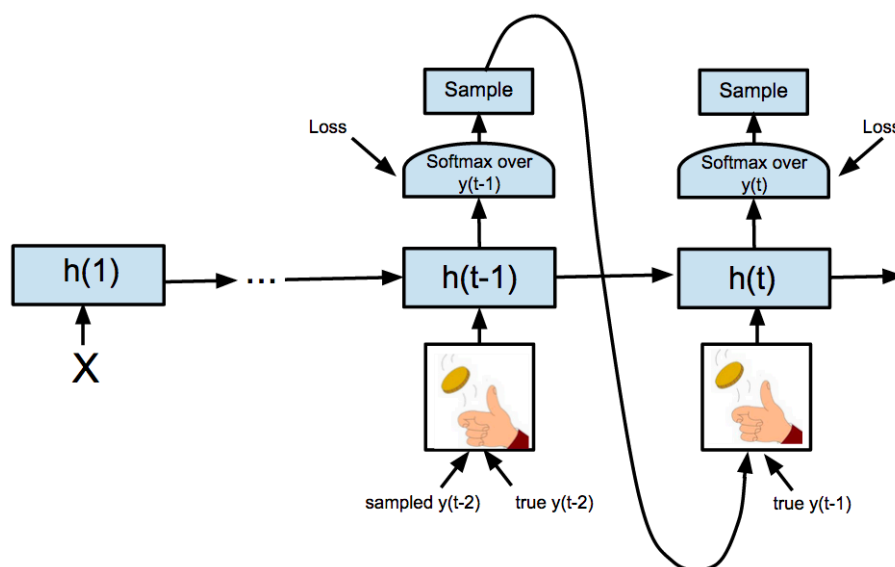
از آنجا که در گام آموزش شبکه فقط در مواجهه داده‌های کاملاً درست قرار گرفته و در آزمون در مواجهه داده‌های تولید شده قرار گرفته، این مشکل آرایی مواجهه نامیده می‌شود [۱۳، ۲۹].

۲.۴-۲ روش نمونه برداری زمان‌بندی شده

برای حل مشکل آرایی مواجهه در مقاله‌ی [۲۹] راه‌کاری به نام نمونه‌برداری زمان‌بندی شده^۱ پیشنهاد شده است که در عمل باعث بهبود می‌شود ولی دارای مشکلاتی است که در ادامه بعد از معرفی روش، به آن می‌پردازیم. این

^۱ Scheduled Sampling

روش هم‌چنین به نام داده به عنوان اثبات‌گر^۱ شناخته می‌شود [۳۵]. همانطور که در بخش ۲-۱.۴ توضیح داده شد، مشکل اُربیی مواجهه ریشه در تفاوت پیکربندی شبکه بین دو فاز آموزش و آزمون دارد؛ روش جبر استاد^۲ برای حل این مشکل، در گام‌های آموزش نیز بعضی از عناصر دنباله ورودی به شبکه را از داده‌های مصنوعی و تولید شده توسط خود شبکه انتخاب می‌کند. برای ساخت دنباله‌ی ورودی در مرحله‌ی آموزش، به ازای هر کلمه با احتمال ϵ از داده‌های واقعی استفاده شده و با احتمال $1 - \epsilon$ از کلمه نمونه‌گیری شده از خود مدل استفاده می‌شود. پارامتر ϵ در ابتدای آموزش مقدار برابر یک دارد و این مقدار به تدریج کاهش پیدا می‌کند تا به صفر برسد. با این کار مدل به تدریج برای تولید دنباله در ادامه‌ی نمونه‌های تولید شده از خودش در زمان آزمون آماده می‌شود. این روال در شکل ۲-۷ نمایش داده شده است.



شکل ۲-۷: نمای کلی روش نمونه‌برداری زمان‌بندی شده، انتخاب هر عنصر دنباله مشابه پرتاب یک سکه است که تصمیم بگیریم از نمونه‌ی تولید شده توسط مدل و یا نمونه‌ی با مقدار واقعی استفاده کنیم [۲۹].

مشکلات این روش به شرح زیر است:

- در این روش در هر جایگاه از دنباله، مستقل از دنباله‌ی تولید شده تا آن کلمه، مقدار هدف برای کلمه‌ی بعدی (یعنی مقداری که می‌خواهیم بیشترین احتمال را در خروجی داشته باشد) برابر مقداری است که داده واقعی در جایگاه متناظر دارد. این رفتار ممکن است در بعضی حالات باعث سوق دادن مدل به پیش‌بینی اشتباه شود [۳۵].

برای مثال فرض کنید دنباله‌ی واقعی جمله‌ی «من یک پیاده‌روی طولانی داشتم» باشد. در آموزش با عملیات تصادفی که در روش وجود دارد سه کلمه‌ی اول ورودی از داده‌ی واقعی انتخاب شده و کلمه آخر توسط مدل تولید شود. پیشوند تولید شده در این حالت می‌تواند برابر «من یک پیاده‌روی داشتم» شود (کلمه‌ی آخر توسط مدل پیش‌بینی شده است). در این شرایط در آموزش مدل را به سمتی می‌بریم که کلمه‌ی بعدی از داده‌ی اصلی یعنی کلمه «داشتم» را پیش‌بینی کند، به بیان دیگر مدل را به سمتی سوق می‌دهیم که در کل عبارت «من یک پیاده‌روی داشتم داشتم» را تولید کند.

^۱ Data As Demonstrator (DAD)

^۲ Professor Forcing

- به صورت تئوری در [۱۸] بررسی شده است که این روش تخمین‌گر صحیحی برای بیشینه درست‌نمایی نیست، یعنی اگر ظرفیت مدل و تعداد داده‌های آموزش به بی‌نهایت میل کند، مدل آموزش دیده به سمت مدل با بیشینه درست‌نمایی نمی‌رود و تخمین‌گر آریبی^۱ دارد.

۳.۴-۲ روش‌های مبتنی بر خودرمزگذار وردشی

روش‌هایی مبتنی بر خودرمزگذار وردشی در کاربرد تولید دنباله بیش‌تر به این منظور توسعه یافته‌اند که علاوه بر تولید دنباله، فضای نهان برای دنباله‌ها تولید کنند که در آن فضا، مفاهیم اساسی دنباله بیان شود. مثلاً در زبان طبیعی موضوع جمله و یا ویژگی‌های معنایی آن از جمله مفاهیمی هستند که در فضا نهان به دنبال آن هستیم. حالت موفق این نمایش در فضای نهان می‌تواند به این صورت باشد که بین نمایش دنباله‌ها بتوان روابط جمع و تفریق به دست آورد، مثلاً بتوان در فضای نهان دنباله‌های زبان طبیعی برداری پیدا کرد که با اضافه کردن آن به یک جمله، با حفظ مفهوم، آن را به سمت غیر رسمی‌تر شدن ببرد و یا با تفریق آن متن را به سمت رسمی‌تر شدن ببرد. برای این هدف مدل ارائه شده در بخش ۲-۱.۲ مناسب نیستند زیرا فضای نهانی ندارند و ویژگی‌هایی هم که در هر گام این روش‌ها به دست می‌آید، به هدف تولید کلمه‌ی بعدی است و نه برای دربرداشتن مفاهیم دنباله [۲۶]. برای به دست آوردن فضای نهان می‌توان از روش خودرمزگذار وردشی استفاده کرد. به عبارت دیگر این روش‌ها راهی برای یادگیری در مدل‌های معرفی‌شده در بخش ۲-۲.۲ هستند.

روش خودرمزگذار وردشی شامل دو شبکه‌ی رمزگذار^۲ و رمزگشا^۳ است و تابع هزینه‌ی آن کران پایینی برای درست‌نمایی است که با بیشینه کردن این کران سعی در بیشینه کردن درست‌نمایی داریم، این کران به صورت زیر است [۲۶، ۳۰]:

$$\mathcal{L}(\theta; x) = -\mathcal{D}_{KL}(q_{\theta}(z|x)||p(z)) + \mathbb{E}_{q_{\theta}(z|x)}[\log p_{\theta}(x|z)]. \quad (5.2)$$

در رابطه‌ی ۵.۲ توزیع $q_{\theta}(z|x)$ مدل احتمالی رمزگذار، $p_{\theta}(x|z)$ مدل احتمالی رمزگشا است و $p(z)$ توزیع احتمالی است که فرض کرده‌ایم متغیر نهان z از آن می‌آید؛ سمت راست رابطه‌ی ۵.۲ شامل دو عبارت می‌شود. اولی عبارتی بر حسب فاصله‌ی KL است که توزیع پسین^۴ یعنی $q(z|x)$ را به توزیع پیشین^۵ یعنی $p(z)$ نزدیک می‌کند. عبارت دوم را تابع هزینه‌ی بازسازی^۶ می‌نامیم و سعی دارد شبکه را به این سمت سوق دهد که ورودی را در خروجی تولید کند. معمولاً توزیع پیشین و پسین گاوسی فرض می‌شود، توزیع پیشین گاوسی با میانگین صفر و کواریانس همانی است و توزیع پسین با گاوسی با کواریانس قطری است که پارامترهای آن توسط شبکه بر حسب ورودی مشخص می‌شود. لازم به ذکر است که در حالتی که توزیع پسین و پیشین گاوسی است، برای فاصله‌ی KL در رابطه‌ی ۵.۲ فرم بسته وجود دارد [۲۶، ۳۰].

در مقاله‌های [۲۶] و [۳۱] دو نمونه استفاده از خودرمزگذار وردشی در حوزه‌ی دنباله پیشنهاد شده است، هر دوی این روش‌ها شبکه‌ی رمزگذاری از نوع LSTM دارند ولی در روش اول شبکه‌ی رمزگشا LSTM و در روش دوم شبکه‌ای پیچشی است، در ادامه به این دو روش می‌پردازیم.

^۱ Bias

^۲ Encoder

^۳ Decoder

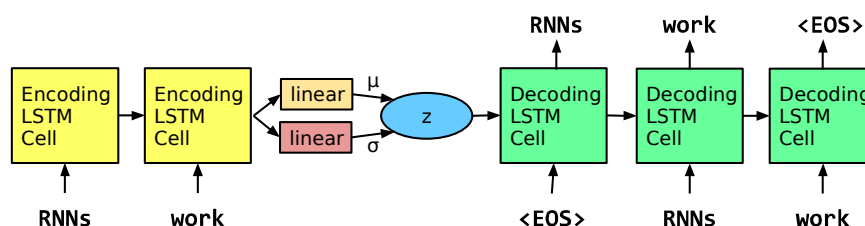
^۴ Posterior

^۵ Prior

^۶ Reconstruction Loss

شبکه‌ی LSTM به عنوان رمزگشا

همان‌طور که گفته شد روش ارائه شده در [۲۶] برای رمزگذار و رمزگشا از شبکه‌ی LSTM استفاده کرده است، این روش را در ادامه VAE-LSTM-LSTM می‌نامیم. روش VAE-LSTM-LSTM همان‌طور که گفته شد رمزگذار و رمزگشا از نوع LSTM دارد. شبکه رمزگذار به وسیله‌ی تبدیلی خطی بر روی آخرین وضعیت مخفی شبکه‌ی بازگردنده، پارامترهای توزیع پسین را محاسبه می‌کند. هم‌چنین از مدل مولد بازگردنده مبتنی بر فضای نهان (بخش ۲-۲.۲) در حالت غیرقطعی، به عنوان شبکه‌ی رمزگشا استفاده شده است. ساختار توضیح داده شده در شکل ۸-۲ برای یک نمونه داده‌ی ورودی بیان شده است.



شکل ۸-۲: نمای کلی شبکه‌ی خودرمزگذار وردشی با رمزگذار و رمزگشای LSTM برای یک نمونه ورودی [۲۶]

در حالت عادی مدلی که اطلاعات مفید رمزگذار را در z ذخیره می‌کند، دارای مقدار غیر صفر برای جمله‌ی فاصله‌ی KL در رابطه‌ی ۵.۲ است، یعنی دو توزیع پسین و پیشین دقیقاً برابر نمی‌شود. ولی در عمل زمانی که شبکه‌ی VAE-LSTM-LSTM با روش معمول آموزش شبکه‌های خودرمزگذار وردشی آموزش داده می‌شود، در بیشتر موارد این رفتار دیده شده و مقدار KL صفر می‌شود که به این معنی است که دو توزیع $p(z)$ و $q_\theta(z|x)$ یکسان می‌شوند. زمانی که مدل این رفتار را بروز می‌دهد، می‌تواند هر توزیع دلخواهی را به عنوان خروجی مستقل از z تولید کرده و مشابه روش جبر معلم عمل کند [۲۶]. دلیل این رفتار کاملاً مشخص نیست ولی ساختار LSTM استفاده شده به عنوان رمزگشا این خاصیت را دارد که اطلاعات ورودی رمز شده از رمزگذار را نادیده می‌گیرد [۲۶، ۳۱]. در VAE-LSTM-LSTM دو راهکار برای رفع مشکل صفر شدن KL پیشنهاد شده است:

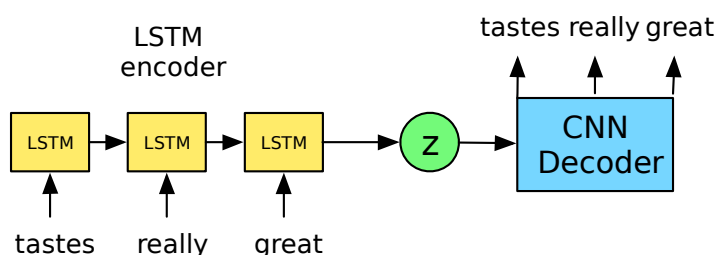
- با وزن دار کردن جمله‌ی KL در رابطه‌ی ۵.۲، به طوری که این وزن در ابتدا صفر باشد تا این موضوع باعث شود مدل بدون مشکلی در افزایش تابع هزینه، اطلاعات رمزگذار را در z رمز کند و خطای بازسازی کمتر شود. سپس در ادامه‌ی آموزش، این وزن افزایش یافته تا هدف دیگر آموزش که نزدیک کردن دو توزیع $p(z)$ و $q_\theta(z|x)$ به هم است، برآورده شود.
- با جایگذاری بعضی از کلمات ورودی، شبکه رمزگشا به سمتی سوق داده می‌شود که به متغیر نهان اهمیت بیشتری دهد. در این جایگذاری کلمات ورودی به صورت تصادفی به کلمه‌ی UNK (به معنی نامعلوم) تبدیل می‌شوند. بنابراین رمزگشا مجبور شود به سمتی برود که اطلاعات حذف شده را از متغیر نهان استخراج کند و به متغیر نهان اهمیت بیشتری دهد [۲۶].

نتایج این روش با وجود به دست آوردن نمایش مخفی از دنباله، در تولید دنباله‌ی زبانی ضعیف‌تر از مدل‌های ساده‌ی زبانی مثل روش جبر معلم بوده است [۲۶]؛ یکی از دلایلی که برای ضعف این روش ذکر می‌شود این است که

توزیع پسینی که شبکه‌ی رمزگذار می‌سازد، کل فضای نهان را پوشش نمی‌دهد و بخش زیادی از این فضا به دنباله‌ی معتبری متناظر نمی‌شود [۹].

شبکه‌ی پیچشی به عنوان رمزگشا

در راهکاری دیگر که در [۳۱] ارائه شده، با جایگذاری ۲-۲.۲ به صورت غیرقطعی به عنوان رمزگشا، نتایج بهبود داده شده است. نمای کلی شبکه در شکل ۲-۹ نشان داده شده است، این روش را VAE-LSTM-CNN می‌نامیم. با ساختار جدید علاوه بر بهبود روش نسبت به مدل VAE-LSTM-LSTM، نتایج در حوزه‌ی تولید دنباله نسبت به روش پایه‌ی جبر معلم هم بهبود یافته است.



شکل ۲-۹: ساختار کلی شبکه‌ی خودرمزگذار وردشی با رمزگذار LSTM و رمزگشای پیچشی [۳۱]

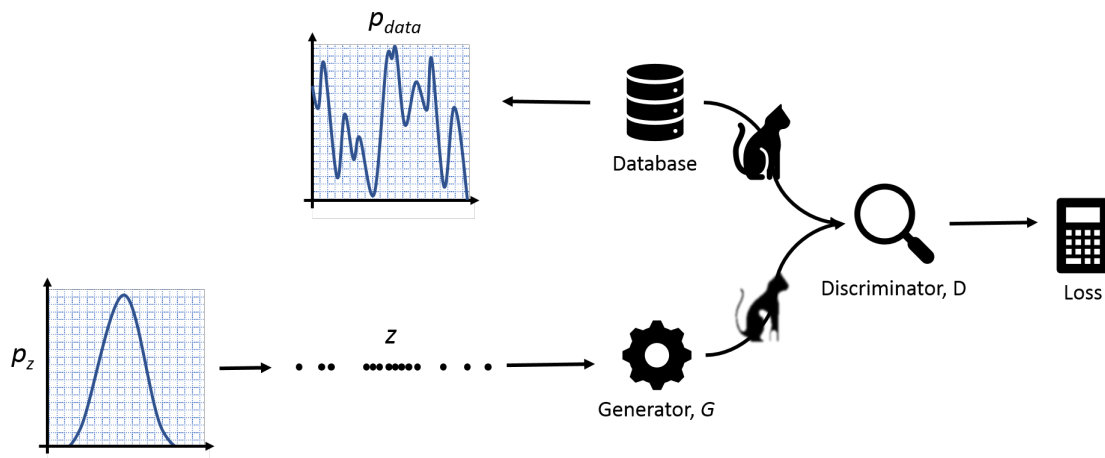
۵-۲ شبکه‌های مولد مقابله‌ای

در بخش ۱-۳ به بیان مختصری از شبکه‌های مولد مقابله‌ای پرداختیم، در این بخش به صورت دقیق‌تر به این موضوع می‌پردازیم. لازم به ذکر است که منظور از یادگیری مقابله‌ای ایده‌ای مشابه با پژوهش شبکه‌های مولد مقابله‌ای [۶] است.

اساس کار شبکه‌های مولد مقابله‌ای [۶] بازی بین دو شبکه است؛ شبکه‌ی اول مولد است که نمونه تولید می‌کند و شبکه‌ی دوم یک دسته‌بند احتمالاتی و دو دسته‌ای است که بر روی نمونه‌های تولید شده و واقعی قضاوت کرده و با این قضاوت شبکه‌ی مولد را به سمت بهبود راهنمایی می‌کند. به صورت دقیق‌تر، شبکه‌ی مولد به سمتی می‌رود که دسته‌بند را به خطا بیاندازد و دسته‌بند هم به سمتی می‌رود که دسته‌بندی درست انجام دهد؛ یک حالت تعادل این بازی زمانی است که مولد نمونه‌های مشابه نمونه‌های واقعی تولید کند و دسته‌بند هم برای داده‌های واقعی و هم برای داده‌های تولید شده احتمال $\frac{1}{2}$ نسبت دهد. زیرا در حالتی که دو دسته‌ی داده دارای توزیع یکسان هستند، تشخیص درست برای دسته‌بند دادن احتمال $\frac{1}{2}$ است.

مدل مولد در این روش، در واقع یک تابع تبدیل است که تبدیلی از فضای نهان به فضای نمونه‌ها انجام می‌دهد. فضای نهان توزیع ساده‌ای مثل گاوسی در نظر گرفته می‌شود. بنابراین روال تولید نمونه به این صورت می‌شود که بعد از نمونه‌گیری از توزیع گاوسی، شبکه‌ی مولد بر روی این نمونه اعمال شده و نمونه‌ی نهایی تولید می‌گردد. مدل دسته‌بند که به نام تمیزدهنده شناخته می‌شود، یک شبکه‌ی دسته‌بند دو دسته‌ای عادی است. در شکل ۲-۱۰ نمایی

از یادگیری مقابله‌ای نمایش داده شده است.



شکل ۲-۱۰: نمای کلی روش یادگیری مقابله‌ای

تابع هزینه‌ی این روش به صورت زیر است:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}} [\log D(x)] + \mathbb{E}_{z \sim P_z} [\log(1 - D(G(z)))]. \quad (6.2)$$

در رابطه‌ی ۶.۲ عبارت سمت راست، منفی تابع هدف دسته‌بندی بین مجموعه داده‌های $\{x | x \sim P_{data}\}$ به عنوان دسته‌ی با برجسب صفر و مجموعه داده‌ی $\{G(z) | z \sim P_z\}$ به عنوان دسته‌ی با برجسب یک است، که P_z نشان دهنده‌ی توزیع فضای نهان است. این تابع هدف، توسط D که تمیزدهنده است، بیشینه می‌شود یعنی تمیزدهنده دسته‌بندی درست را انجام می‌دهد و از سمت دیگر تابع هدف نسبت به G که مولد است کمینه می‌شود. بنابراین هدف آن است که نمونه‌های تولیدی مولد طوری باشد که تمیزدهنده به اشتباه بیافتد.

در عمل تابع هزینه‌ی ۶.۲ به طور مستقیم قابل استفاده نیست و روش حلی که برای آن ارائه شده آموزش گام به گام شبکه‌ی مولد و تمیزدهنده است؛ به این صورت که در یک مرحله مدل مولد و در مرحله‌ی دیگر مدل تمیزدهنده آموزش ببیند. با تکرار این مراحل، آموزش انجام می‌شود. بر حسب راه‌حل گفته شده تابع هزینه‌ی دو شبکه به صورت زیر در می‌آید:

$$\begin{aligned} \mathcal{L}_D &= -\mathbb{E}_{x \sim P_{data}} [\log D_\phi(x)] - \mathbb{E}_{z \sim P_z} [\log(1 - D_\phi(G_\theta(z)))], \\ \mathcal{L}_G &= \mathbb{E}_{z \sim P_z} [\log(1 - D_\phi(G_\theta(z)))]. \end{aligned} \quad (7.2)$$

نکته‌ای که در رابطه‌ی ۷.۲ وجود دارد تابع هزینه‌ی مولد است؛ زمانی که شبکه‌ی مولد به‌روز رسانی می‌شود، گرادیان \mathcal{L}_G نسبت به θ لازم است، برای معتبر بودن این گرادیان باید $D_\phi(x)$ نسبت به x مشتق‌پذیر باشد.

۱.۵-۲ بررسی تئوری

در این بخش نشان می‌دهیم که آموزش شبکه مولدمقابله‌ای در حالتی که شبکه‌ها ظرفیت کافی دارند و در بهینه‌سازی به نقطه‌ی بهینه برسند معادل کاهش فاصله‌ی Jensen-Shannon بین توزیع تولیدی توسط مولد و توزیع داده‌ی واقعی است.

قضیه ۱.۲. برای یک مولد ثابت، تمیزدهنده بهینه به صورت زیر است [۶]:

$$D_G^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}, \quad (۸.۲)$$

که P_g نشان‌دهنده‌ی توزیع مولد است.

با توجه به قضیه‌ی ۱.۲ می‌توان تابع هزینه گفته شده در رابطه‌ی ۶.۲ را با جایگذاری دسته‌بند بهینه ساده کرد و به عبارت زیر رسید [۶]:

$$\min_G V(D^*, G) = -\log(4) + \mathcal{D}_{JS}(P_{data} \| P_g), \quad (۹.۲)$$

که $\mathcal{D}_{JS}(P_{data} \| P_g)$ به معنی فاصله‌ی Jensen-Shannon بین توزیع تولیدی توسط مولد و توزیع داده‌ی واقعی است.

نتیجه ۱. بنابراین آموزش شبکه‌های مولد مقابله‌ای در حالت حدی، معادل کاهش فاصله‌ی Jensen-Shannon بین دو توزیع مولد و داده‌ی واقعی است.

۶-۲ یادگیری مقابله‌ای با استفاده از Gumbel Softmax

زمانی که می‌خواهیم از روش شبکه‌های مولد مقابله‌ای^۱ در حوزه‌ی داده‌های گسسته استفاده کنیم؛ همان‌طور که در بخش ۵-۲ و ۴-۴ توضیح داده شد، مشکل انتقال گرادیان به وجود می‌آید. این مشکل ناشی از عملیات نمونه‌گیری در خروجی شبکه مولد است، زیرا اکثر شبکه‌ها برای ایجاد توزیع دسته‌ای در خروجی شبکه از یک تابع بیشینه‌ی هموار استفاده می‌کنند. روش پیشنهاد شده در [۷] که GSGAN نامیده می‌شود، برای حل مشکل انتقال گرادیان، فرایند انتهایی شبکه را با تقریبی مشتق‌پذیر جایگذاری می‌کند. در ادامه پس از معرفی تکنیک تنظیم مجدد برای توزیع دسته‌ای و نحوه‌ی تقریب نمونه‌گیری، به جزئیات روش GSGAN می‌پردازیم.

۱.۶-۲ تکنیک تنظیم مجدد برای توزیع دسته‌ای

اگر یک توزیع دسته‌ای داشته باشیم که احتمال هر حالت با π_i بیان شود (برای برقرار شدن شرط توزیع باید $\sum_i \pi_i = 1$ شود). یک راه معادل برای نمونه‌گیری از این توزیع به صورت زیر است [۱۹]:

$$y = \arg \max_i \{ \log(\pi_i) + g_i \}, \quad (۱۰.۲)$$

$$g_i \sim \text{Gumbel}(\cdot, 1)$$

که g_i ها نمونه‌های مستقل از توزیع $\text{Gumbel}(\cdot, 1)$ با چگالی $f(z) = e^{-z - \exp(-z)}$ هستند^۲. این نوع نمونه‌گیری که به نام تکنیک تنظیم مجدد^۳ شناخته می‌شود، قسمت تصادفی نمونه‌گیری را از مسیر بازگشت مشتق

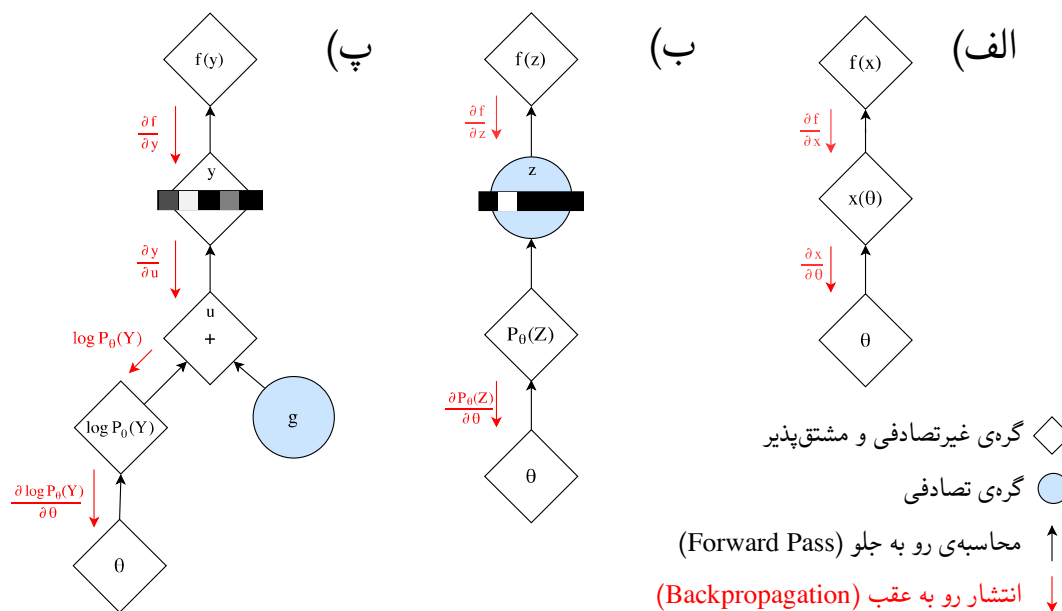
^۱ Generative Adversarial Networks

^۲ برای نمونه‌گیری از توزیع گامبل می‌توان از روش نمونه‌گیری تبدیل معکوس استفاده کرد، به این صورت که از توزیع یکنواخت به صورت $u \sim \text{Uniform}(\cdot, 1)$ نمونه گرفته شود و به وسیله‌ی رابطه‌ی $g = -\log(-\log(u))$ به نمونه‌های توزیع گامبل تبدیل می‌شود [۱۹].

^۳ Reparametrization

خارج می‌کند.

شکل ۲-۱۱ (الف) نشان دهنده عملیاتی است که عملگرهای آن قطعی و مشتق‌پذیراند؛ در این حالت برای محاسبه‌ی مشتق به صورت انتشار رو به عقب^۱ مشکلی وجود ندارد. شکل ۲-۱۱ (ب) نشان دهنده‌ی حالتی است که در محاسبات، عملگری تصادفی وجود دارد و این عملگر باعث می‌شود در محاسبه‌ی مشتق به صورت انتشار رو به عقب، با عنصر مشتق‌ناپذیر مواجه شویم و مشتق قابل محاسبه نباشد. شکل ۲-۱۱ (پ) نشان دهنده‌ی راهکاری است با تکنیک تنظیم مجدد که برای توزیع دسته‌ای استفاده می‌شود. همان‌طور که دیده می‌شود، قسمت تصادفی محاسبات از مسیر بازگشت مشتق خارج شده است و برای محاسبه‌ی مشتق کافی است که عملگر $\arg \max$ با عملگر مشتق‌پذیر تقریب زده شود.



شکل ۲-۱۱: نمایش مشکل محاسبه گرادیان به دلیل وجود عملگر تصادفی و حل این مشکل توسط Gumbel Softmax [۱۹]

۲.۶-۲ تقریبی مشتق‌پذیر برای نمونه‌گیری از توزیع دسته‌ای

در این بخش به نحوه‌ی تقریب فرایند نمونه‌گیری از یک توزیع دسته‌ای می‌پردازیم. نحوه‌ی بیانی که برای نمونه‌های گرفته شده استفاده می‌شود، کد شده‌ی آن‌ها به صورت یک-فعال^۲ است؛ زیرا معمولا ورودی شبکه‌ی تمیزدهنده نمونه‌های کد شده به صورت یک-فعال است. در این بخش ابتدا نحوه‌ی تقریب عمل $\arg \max$ را نشان می‌دهیم و سپس با کمک آن عملیات نمونه‌گیری را تقریب می‌زنیم.

فرض کنید که برداری به اندازه‌ی v داریم که x نام دارد، و تابع $f(x)$ بر روی آن به صورت زیر تعریف می‌شود:

$$f(x) = \text{onehot}(\arg \max_i \{x_i\}). \quad (۱۱.۲)$$

^۱ Backpropagation

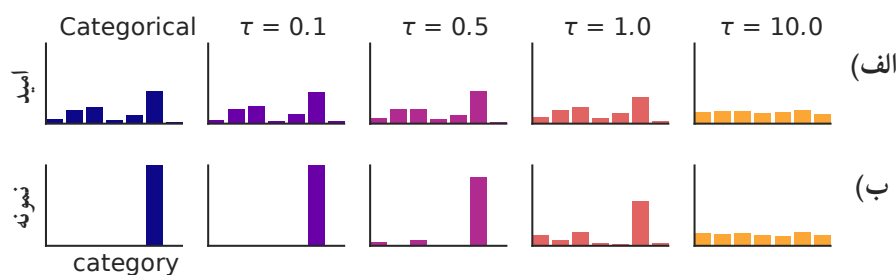
^۲ One-hot

می‌خواهیم تابع $f(x)$ را با عبارتی مشتق‌پذیر تقریب بزنیم. یک راه آن استفاده از تابع بیشینه‌ی هموار است؛ به این صورت که تمام مقادیر بردار x به عدد τ تقسیم شده و سپس وارد یک تابع بیشینه‌ی هموار می‌شود؛ زمانی که $\tau \rightarrow 0$ ، خروجی تابع بیشینه‌ی هموار تقریبی از $f(x)$ می‌شود:

$$f(x) \underset{\tau \rightarrow 0}{\approx} \text{softmax}\left(\frac{x}{\tau}\right). \quad (۱۲.۲)$$

در تقریب ارائه شده مشتق تابع بیشینه‌ی هموار هم تعریف شده است و تقریبی مشتق‌پذیر برای عملگر $\arg \max$ به دست می‌آید؛ این روش تقریب Gumbel Softmax نامیده می‌شود [۱۹]. از طرفی با تکنیک تنظیم مجدد که در بخش ۲-۱.۶ معرفی شد، برای نمونه‌گیری از توزیع دسته‌ای، نیازمند انجام عملگر $\arg \max$ به صورت مشتق‌پذیر هستیم. بنابراین با ترکیب این دو موضوع تقریبی مشتق‌پذیر از نمونه‌گیری توزیع دسته‌ای به دست می‌آید.

لازم به ذکر است که اگر $\tau \rightarrow \infty$ برود، حاصل به جای برداری یک-فعال (که فقط یک مقدار غیر صفر دارد)، برداری است که تمام درایه‌های آن غیر صفر و مساوی هم هستند. در شکل ۲-۱۲ تاثیر پارامتر τ نشان داده شده است. در سطر اول، نمودار سمت چپ یک توزیع دسته‌ای است که با مقدار احتمال آن توصیف شده است. بقیه نمودارهای سطر اول نشان دهنده‌ی توزیع جدید تقریب زده شده بر حسب مقادیر مختلف τ هستند. همان‌طور که مشخص است با افزایش مقدار τ ، توزیع به سمت توزیع یکنواخت می‌رود و دقت تقریب کم می‌شود. در سطر دوم، بردار نمونه‌های تولید شده از توزیع متناظر در سطر اول نشان داده شده است، در یک تقریب خوب انتظاری که وجود دارد این است که نمونه‌ها به صورت یک-فعال باشد، در حالی که با افزایش τ این ویژگی هم تضعیف می‌شود. با وجود این که مقدار τ با مقادیر کوچک بهتر است، ولی در این حالت واریانس مشتق تخمینی این عملگر افزایش می‌یابد، این موضوع باعث می‌شود بین مقدار τ و واریانس مشتق تخمینی مصالحه^۱ به وجود آید [۱۹].

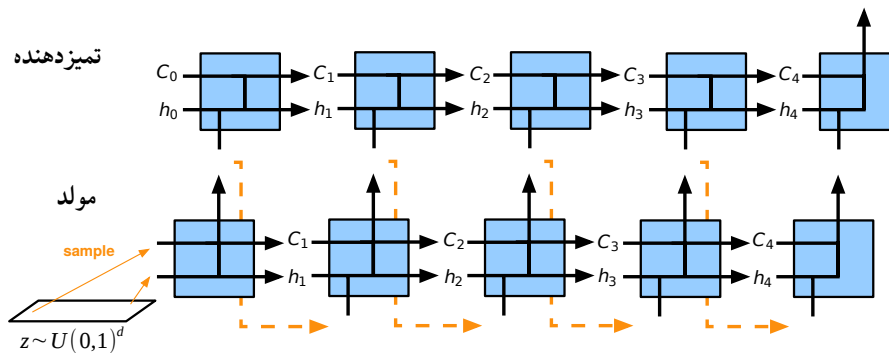


شکل ۲-۱۲: نمایش اثر پارامتر τ بر روی تقریب Gumbel Softmax [۱۹]

۳.۶-۲ جزئیات روش

در روش GSGAN، همان‌طور که در شکل ۲-۱۳ نشان داده شده است، از معماری LSTM برای شبکه‌ی مولد و تمیزدهنده استفاده شده و آخرین خروجی LSTM برای ایجاد خروجی تمیزدهنده استفاده می‌شود. در شبکه‌های معمول و مورد استفاده در مدل کردن دنباله‌ی گسسته، خروجی شبکه‌ی مولد در هر مرحله یک توزیع دسته‌ای است که این توزیع با کمک یک تابع بیشینه‌ی هموار ایجاد می‌شود. در روش GSGAN خروجی‌های شبکه با کمک Gumbel

^۱ Tradeoff



شکل ۲-۱۳: نمایی از روش [۷] که از Gumbel Softmax برای حل مشکل انتقال گرادیان استفاده کرده است [۷].

Softmax به صورت مشتق‌پذیر تقریب زده شده‌اند. این موضوع باعث می‌شود تمام توابع هزینه و روش‌هایی که مبتنی بر روش کاهش گرادیان هستند، بر روی شبکه قابل اعمال باشد. در ادامه جزئیات تقریب استفاده شده در خروجی شبکه توضیح داده می‌شود و سپس تابع هزینه‌ی مورد استفاده بیان می‌گردد. اگر h بردار خروجی مولد قبل از تابع بیشینه‌ی هموار باشد، احتمال کلمه‌ی i -ام از توزیع دسته‌ای که شبکه تولید می‌کند، برابر مقدار زیر است:

$$\frac{\exp(h_i)}{\sum_{j=1}^V \exp(h_j)}. \quad (13.2)$$

حال با توجه به روش نمونه‌گیری که در بخش ۲-۱.۶ گفته شد، می‌توان نمونه‌گیری از این توزیع را به صورت زیر نوشت:

$$y = \arg \max_i \{h_i + g_i\} \quad (14.2)$$

اگر نمونه‌های یک-فعال را به کمک تابع بیشینه‌ی هموار تقریب بزنیم، به ازای یک τ ثابت، حاصل به صورت زیر می‌شود [۷، ۱۹]:

$$y_{onehot} = \text{softmax}\left(\frac{h + g}{\tau}\right) \quad (15.2)$$

مقدار τ در روال آموزش تغییر می‌کند. روالی که در روش استفاده شده است به این صورت است که در ابتدای آموزش τ مقدار زیادی دارد و در حین آموزش مقدار آن کاهش پیدا می‌کند و بعد از رسیدن به یک مقدار مشخص، ثابت می‌ماند [۷]. با شبکه‌ی گفته شده، معماری کاملاً مشتق‌پذیر می‌شود و استفاده از روش مطرح شده در حوزه‌ی پیوسته ممکن می‌شود. تابع هزینه‌ی مورد استفاده برای شبکه‌ی مولد و تمیزدهنده به صورت زیر است:

$$\begin{aligned} \mathcal{L}_D &= -\mathbb{E}_{x \sim P}[\log D(x)] - \mathbb{E}_{z \sim P_z}[\log(1 - D(G(z)))], \\ \mathcal{L}_G &= -\mathbb{E}_{z \sim P_z}[\log \frac{D(G(z))}{1 - D(G(z))}]. \end{aligned} \quad (16.2)$$

تابع آموزش تمیزدهنده مشابه روش GAN عادی، آنتروپی متقاطع^۱ بین نمونه‌های واقعی و داده‌ی تولید شده است. برای تولید نمونه از شبکه‌ی مولد، تعدادی نمونه از توزیع گامبل لازم است که z نمایش‌دهنده‌ی آن‌ها است. تابع هزینه‌ی شبکه‌ی مولد با شبکه‌های مولد مقابله‌ای عادی تفاوت دارد، این تابع از مقاله‌ی [۳۶] اقتباس شده که در

^۱ Cross Entropy

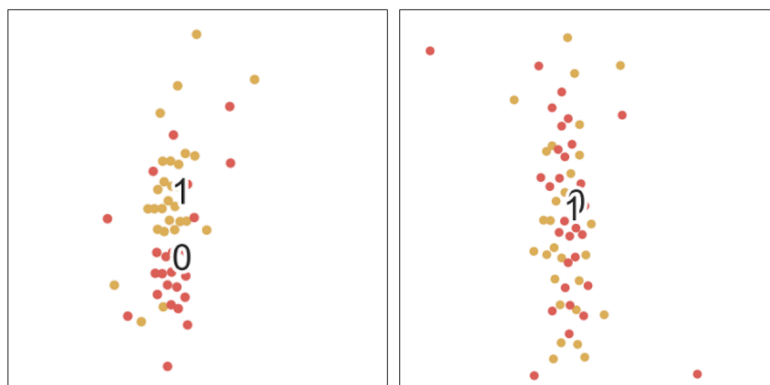
حوزه‌ی تولید تصویر باعث بهبود گشته است. روال آموزش، مشابه شبکه‌های مولد مقابله‌ای، به این صورت است که در یک گام شبکه‌ی مولد ثابت فرض می‌شود و تمیزدهنده آموزش می‌بیند، در گام بعد با ثابت فرض شدن تمیزدهنده شبکه‌ی مولد آموزش داده می‌شود.

۷-۲ یادگیری مقابله‌ای با استفاده از فضای ویژگی

یکی از راهکارهایی که برای استفاده از شبکه‌های مولد مقابله‌ای در حوزه‌ی داده‌های گسسته وجود دارد، بردن فضای آموزش دنباله به یک فضای پیوسته مثل یک فضای ویژگی است. راه‌حل اولیه‌ای که در این حوزه به نظر می‌رسد، آموزش یک شبکه‌ی برای استخراج ویژگی، مثل خودرمزگذار^۱ و استفاده از شبکه‌های مولد مقابله‌ای در فضای جدید است. ولی این روش در تولید دنباله‌های واقعی عملکردی ضعیف دارد، زیرا معمولاً فضای ویژگی‌ای که یاد گرفته می‌شود ساختار دارد، ولی بخش‌های کوچکی از این فضا متناظر با یک دنباله‌ی معتبر است و در عمل دیده شده که نمونه‌گیری در این فضا به تولید دنباله‌های بی‌مفهومی منجر می‌شود [۹]. در ادامه روش‌هایی توضیح داده می‌شوند که از فضای ویژگی استفاده کرده‌اند. روش جبر استاد با کمک آموزش در فضای ویژگی، مشکل انتقال گرادیان را از بین برده و با یادگیری مقابله‌ای سعی در حل آریبی مواجهه دارد. روش TextGAN با کمک فضای ویژگی، پایداری آموزش یادگیر مقابله‌ای را بهبود داده است.

۱.۷-۲ روش جبر استاد

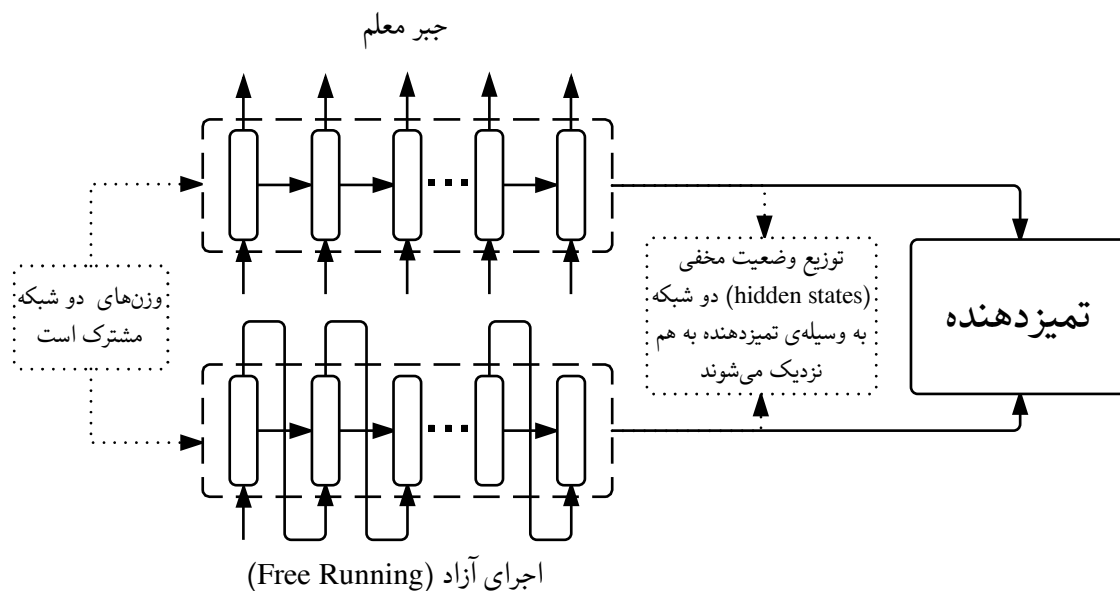
روش معرفی شده در [۸] که با نام جبر استاد شناخته می‌شود، با هدف رفع مشکل آریبی مواجهه روش جبر معلم (بخش ۱.۴-۲) ارائه شده است. این روش از ساختار مشابه شبکه‌های مولد مقابله‌ای برای این منظور استفاده کرده است. راهکار پیشنهادی روش جبر استاد مشکلی برای انتقال گرادیان از تمیزدهنده به مولد ایجاد نمی‌کند، زیرا شبکه‌ی دسته‌بند بر روی فضای نهان شبکه‌ی مولد کار می‌کند.



شکل ۱۴-۲: نمایش وضعیت مخفی شبکه بازگردنده در دو بعد به وسیله‌ی T-SNE. تصویر سمت راست مربوط به شبکه‌ی آموزش دیده با روش جبر استاد و تصویر سمت چپ مربوط به شبکه‌ی آموزش دیده با روش جبر معلم است. نقاط قرمز بیانگر مقدار مخفی شبکه در حین آموزش و نقاط زرد بیانگر مقدار مخفی شبکه در زمان نمونه برداری و آزمون است [۸].

^۱ Autoencoder

ایده‌ی روش از آنجا نشأت می‌گیرد که در جبر معلم اختلاف در رفتار شبکه بین زمان آموزش و آزمون وجود دارد، این موضوع در سمت چپ شکل ۲-۱۴ قابل مشاهده است که وضعیت شبکه در زمان آموزش و آزمون متفاوت است، بنابراین روش جبر استاد از شبکه‌ای تمیزدهنده استفاده می‌کند تا رفتار شبکه در دو فاز آموزش و آزمون شبیه به هم شود. برای این کار دو حالت شبکه را در نظر می‌گیرد، حالت اول که در آموزش جبر معلم است و دنباله پیشوندی از داده‌های واقعی به شبکه وارد می‌شود و حالت دیگر همان شبکه با ورودی‌های متفاوت است به طوری که ورودی هر مرحله از داده‌های تولید شده از خود مولد در مرحله‌ی قبل است. در نهایت روش کلی آموزش به این صورت می‌شود که در یک مرحله، مولد مشابه جبر معلم آموزش داده می‌شود و در مرحله‌ی دیگر شبکه‌ها به سمتی برده می‌شوند که دو حالت مذکور شبکه از هم قابل تمایز نبوده و در این جهت به‌روزرسانی شبکه‌های تمیزدهنده و مولد انجام می‌شود. این دو مرحله به صورت متوالی انجام می‌شود. نمای کلی آموزش در شکل ۲-۱۵ مشخص است که قسمت بالا حالت اول و قسمت پایین حالت دوم را نشان می‌دهد. لازم به ذکر است شبکه مولد به دو صورت پیکربندی شده است و پارامترهای هر دو پیکربندی مشترک است. با توجه به توضیحات داده شده برای



شکل ۲-۱۵: ساختار کلی نحوه‌ی آموزش در روش جبر استاد [۸]

حالت غیر شرطی، توابع هزینه‌ی استفاده شده برای تمیزدهنده و مولد به صورت زیر هستند:

$$\begin{aligned}\mathcal{L}_D &= -\mathbb{E}_{x \sim P}[\log D(B(x))] - \mathbb{E}_{x \sim Q}[\log(1 - D(B(x)))] \\ \mathcal{L}_G &= -\mathbb{E}_{x \sim P}[\log q(x)] - \mathbb{E}_{x \sim Q}[\log D(B(x))],\end{aligned}\quad (17.2)$$

که $B(x)$ نشان‌دهنده‌ی خروجی و وضعیت مخفی شبکه‌ی مولد به ازای ورودی x است.

در واقع روش جبر استاد مشابه همان جبر معلم عمل کرده ولی شبکه‌ی تمیزدهنده مانند نرمال‌ساز، در آموزش دخیل است. در سمت راست شکل ۲-۱۴ می‌توان وضعیت شبکه‌ای را که با روش جبر استاد آموزش دیده، مشاهده کرد، که دیگر دو حالت شبکه بر خلاف شکل سمت چپ که مرتبط با روش جبر معلم است، از هم قابل تمیز نیستند. اطلاعاتی که به شبکه تمیزدهنده وارد می‌شود، شامل وضعیت مخفی و خروجی شبکه قبل از نمونه‌برداری است. این موضوع باعث می‌شود از این روش بتوان بدون مشکل بر روی داده‌های گسسته استفاده کرد. این روش باعث بهبود

و تعمیم پذیری مدل نسبت به روش پایه‌ی جبر معلم شده است، ولی تفاوت نتیجه بیش‌تر در دنباله‌های طولانی است [۸].

۲.۷-۲ روش TextGAN

روش TextGAN که در [۹] ارائه شده است مبتنی بر ایده‌ی شبکه‌های مولد مقابله‌ای مطرح شده است، ولی بر اساس نزدیک کردن توزیع مدل و توزیع اصلی در فضای ویژگی کار می‌کند. ایجاد فضای ویژگی با کمک شبکه‌ی تمیزدهنده در [۳۷] پیشنهاد شده است. روش TextGAN با تعمیم این روش به وسیله‌ی شبکه‌ی تمیزدهنده، ویژگی‌های مناسبی از دنباله به دست می‌آورد و شبکه‌ی مولد در این فضا دو توزیع اصلی و مولد را به هم نزدیک می‌کند. موضوعی که این پژوهش روی آن تاکید داشته، بهبود مشکل چسبیدگی به قله در روش پیشنهادی نسبت به دیگر روش‌های مبتنی بر یادگیری مقابله‌ای است. این بهبود با استفاده از نزدیک کردن ممان‌های^۱ دو توزیع حاصل می‌شود. در این روش از ویژگی‌های لایه‌های اولیه‌ی شبکه‌ی تمیزدهنده برای ساخت فضای ویژگی استفاده می‌شود. برای این فضای ویژگی سه هدف در نظر گرفته شده است؛ اول این که ویژگی‌ها، تمیزدهنده‌ای بین داده‌های مصنوعی و واقعی باشند. دوم، ویژگی‌ها شامل اطلاعات دنباله بوده و نمایشی از آن در فضای نهان باشند. سومین هدف داشتن ویژگی‌هایی است که برای شبکه‌ی مولد فعلی چالش ایجاد کند تا به آموزش بهتر منجر شود. برای رسیدن به این اهداف تابع هزینه زیر برای شبکه‌ی تمیزدهنده تعریف شده است:

$$\begin{aligned}\mathcal{L}_D &= \mathcal{L}_{GAN} + \lambda_r \mathcal{L}_{recon} - \lambda_m \mathcal{L}_{MMD^2}, \\ \mathcal{L}_{GAN} &= -\mathbb{E}_{x \sim P}[\log D(x)] - \mathbb{E}_{z \sim P_z}[\log(1 - D(G(z)))], \\ \mathcal{L}_{recon} &= \|\hat{z} - z\|^2,\end{aligned}\quad (18.2)$$

که \mathcal{L}_D تابع هزینه‌ی تمیزدهنده است و \mathcal{L}_{MMD^2} اختلاف توزیع مدل و داده اصلی بر اساس فاصله‌ی بیشینه میانگین اختلاف^۲ (MMD) با هسته‌ی گاوسی^۳ در فضای ویژگی است، z نشان‌دهنده‌ی مقدار نهانی است که به وسیله‌ی آن دنباله‌ی مصنوعی تولید شده و \hat{z} مقدار بازسازی شده از روی فضای ویژگی به وسیله‌ی شبکه‌ی مولد است، ضرایب λ_r و λ_m فوق پارامترهای^۴ مدل هستند.

در تابع هزینه‌ی تمیزدهنده، جمله‌ی \mathcal{L}_{GAN} که تابع هزینه عادی برای تمیزدهنده است، باعث آموزش ویژگی‌هایی می‌شود که بین داده‌های واقعی و تولید شده تمایز ایجاد می‌کند. جمله‌ی \mathcal{L}_{recon} که خطای بازسازی است باعث سوق دادن فضای ویژگی به سمتی می‌شود که بیش‌ترین اطلاعات دنباله در فضای ویژگی بیان شود. جمله‌ی $-\mathcal{L}_{MMD^2}$ باعث می‌شود ویژگی‌های چالشی تری در مقابل شبکه‌ی مولد به دست آید، زیرا منفی این جمله تابع هزینه شبکه‌ی مولد است:

$$\mathcal{L}_G = \mathcal{L}_{MMD^2}. \quad (19.2)$$

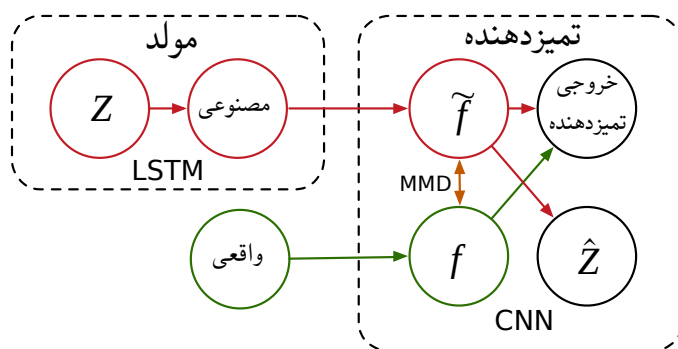
در شکل ۱۶-۲ نمای کلی روش نشان داده شده است. f و \tilde{f} به ترتیب نشان‌دهنده‌ی داده‌ی واقعی و داده‌ی تولیدشده در فضای ویژگی هستند و \hat{z} نشان‌دهنده‌ی مقدار z بازسازی شده از روی \tilde{f} است.

^۱ Moments

^۲ Maximum Mean Discrepancy

^۳ Gaussian kernel

^۴ Hyperparameter



شکل ۲-۱۶: نمای کلی روش TextGAN [۹]

از مدل مولد بازگردنده مبتنی بر فضای نهان (بخش ۲-۲-۲) به عنوان مولد استفاده شده است؛ در روش TextGAN از حالت قطعی استفاده شده و با معلوم شدن مقدار z نمونه‌ی تولیدی مشخص است. برای این منظور از تقریب $\arg \max$ برای مشتق‌پذیری استفاده شده است. از مدل دسته‌بند پیچشی (بخش ۲-۲-۳) به عنوان تمیزدهنده استفاده می‌شود و فضای ویژگی مورد استفاده که بر روی آن MMD محاسبه می‌شود، حاصل شبکه‌ی دسته‌بند بعد از قسمت پیچش و رای‌گیری (قبل از لایه‌ی تمام‌متصل) است. روش TextGAN مشابه اکثر روش‌های مبتنی بر یادگیری مقابله‌ای در تولید دنباله، برای شروع از مدل پیش-آموزش^۱ دیده استفاده می‌کند.

۲-۸ یادگیری مقابله‌ای با استفاده از یادگیری تقویتی

دسته‌ای دیگر از روش‌ها برای استفاده از یادگیری مقابله‌ای بر روی داده‌های گسسته، رویکرد جدیدی به مساله داشته‌اند و مساله‌ی تولید دنباله را به عنوان یک محیط مساله‌ی یادگیری تقویتی دیده‌اند و با کمک این دید از روش‌های یادگیری تقویتی برای آموزش مولد استفاده کرده‌اند. در حالتی که مساله با رویکرد یادگیری تقویتی بررسی می‌شود، مشکل انتقال گرادیان به وجود نمی‌آید، زیرا دیگر در آموزش گرادیان منتقل نمی‌شود بلکه پاداشی تعریف می‌شود که آن برای آموزش مولد منتقل می‌شود.

تولید دنباله را می‌توان یک مساله‌ی یادگیری تقویتی دید، به این صورت که در هر مرحله، وضعیت^۲، همان دنباله‌ی تولید شده تا به آن لحظه است و عمل^۳ که عامل^۴ می‌تواند انجام دهد، انتخاب عنصر بعدی دنباله است. محیط توصیف شده یک محیط قطعی است و با انجام هر عمل، وضعیت جدید به صورتی قطعی مشخص می‌شود، زیرا با انجام عمل، دنباله‌ی جدید که با اضافه شدن کلمه‌ی انتخاب شده به دست آمده، نشان دهنده‌ی وضعیت جدید است. عامل به این صورت مدل می‌شود که یک سیاست^۵ وجود دارد که برای هر وضعیت توزیع احتمالی روی عمل‌های ممکن مشخص می‌کند و برای انتخاب عمل در هر وضعیت از این توزیع نمونه‌گیری می‌شود. در مساله

^۱ Pre-train^۲ State^۳ Action^۴ Agent^۵ Policy

یادگیری تقویتی به دنبال یافتن سیاستی هستیم که امید پاداش^۱ دریافتی را بیشینه کند:

$$\theta^* = \arg \max_{\theta} \mathbb{E}[R(x)|\pi_{\theta}] \quad (20.2)$$

که $R(x)$ نشان دهنده‌ی پاداش نسبت داده شده به دنباله‌ی x است و π_{θ} بیانگر سیاست است. پاداش به این صورت که تعریف می‌شود که فقط برای دنباله‌های کامل پاداش وجود دارد.

گرایان سیاست

یکی از روش‌های ساده و پراستفاده‌ی حل مساله‌ی گفته شده، گرایان سیاست^۲ است. این روش بر مبنای صعود در راستای گرایان^۳ کار می‌کند. برای این منظور در هر گام نیاز به محاسبه‌ی گرایان تابع هزینه نسبت به پارامترهای سیاست وجود دارد. در مساله‌ی تعریف شده شبکه‌ی مولد، مشخص کننده‌ی همان سیاست است و $\pi_{\theta}(x_l|x_{1:l-1}) = q_l(x_l|x_{1:l-1}; \theta)$ طبق رابطه‌ی ۱.۲ برقرار است. با توجه به قضیه‌ی گرایان سیاست داریم [۵، ۱۰، ۳۸]:

$$\nabla_{\theta} J = \sum_{l=1}^L \mathbb{E}_{x_{1:l-1} \sim Q_{\theta}} \left[\sum_{x_l} V_{\theta}(x_{1:l}) \nabla_{\theta} q_l(x_l|x_{1:l-1}; \theta) \right] \quad (21.2)$$

$$V_{\theta}(x_{1:l}) = \mathbb{E}_{x_{l+1:L} \sim Q_{\theta}} [R(x_{1:L})],$$

که $V_{\theta}(x_{1:l})$ بیانگر امید میزان پاداشی است که با ادامه دادن وضعیت $x_{1:l}$ طبق سیاست مشخص شده با θ به دست می‌آید و Q_{θ} توزیع دنباله است که در عبارت ۲-۲ تعریف شده است. جمع روی x_l به معنی جمع روی تمام عمل‌های ممکن (در اینجا به معنی کلمات ممکن) است. برای استفاده از گرایان سیاست نیاز به مشخص شدن $V_{\theta}(x_{1:l})$ است، این مقدار به صورت تخمینی به دست می‌آید.

۱.۸-۲ روش SeqGAN

روش SeqGAN که در [۵] معرفی شده است، از اولین راهکارهای استفاده از یادگیری مقابله‌ای در حوزه‌ی تولید دنباله بوده است. این روش مساله‌ی تولید دنباله را با در نظر گرفتن مساله به عنوان یک مساله‌ی یادگیری تقویتی حل می‌کند. در این مساله پاداش، خروجی شبکه‌ی تمیزدهنده است. خروجی شبکه‌ی تمیزدهنده، احتمالی است که این شبکه به واقعی بودن دنباله‌ی ورودی می‌دهد. این خروجی عددی بین صفر و یک است که هرچه به یک نزدیک‌تر باشد داده از دید تمیزدهنده واقعی‌تر و هرچه به صفر نزدیک‌تر باشد غیر واقعی‌تر است. این روش برای حل مساله‌ی یادگیری تقویتی تعریف شده، از راه کار گرایان سیاست استفاده می‌کند و با استفاده از پاداش دریافت شده، گرایانی را برای شبکه‌ی مولد تخمین می‌زند که با گرایان تخمینی، شبکه‌ی مولد آموزش می‌بیند. با ایده گرفتن از شبکه‌های مولد مقابله‌ای، در روش SeqGAN دو شبکه‌ی مولد و تمیزدهنده وجود دارد و آموزش به این صورت است که در یک گام تمیزدهنده بر روی داده‌های واقعی و داده‌های تولید شده آموزش می‌بیند و در گام دیگر با کمک دسته‌بند آموزش، شبکه‌ی مولد آموزش داده می‌شود.

^۱ Reward

^۲ Policy Gradient

^۳ Gradient Ascent

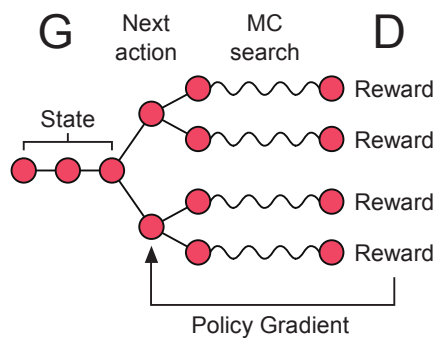
روش SeqGAN برای تمیزدهنده از مدل دسته‌بند پیچشی (بخش ۲-۲.۳) استفاده می‌کند و به همین دلیل فقط به ازای هر دنباله‌ی کامل شده، مقدار خروجی تمیزدهنده، که همان پاداش است، در دسترس است. این موضوع که فقط پاداش برای دنباله‌ی کامل وجود دارد و پاداشی میانی برای دنباله وجود ندارد باعث ایجاد چالش می‌شود. به همین دلیل با استفاده از جستجوی مونت کارلو^۱ پاداش میانی برای دنباله‌ها تعریف می‌شود که در ادامه به جزئیات آن می‌پردازیم. تابع هزینه‌ی تمیزدهنده، تابع عادی شبکه‌های مولد مقابله‌ای است و به صورت زیر تعریف می‌شود:

$$\mathcal{L}_D = -\mathbb{E}_{x \sim P}[\log D(x)] - \mathbb{E}_{x \sim Q}[\log(1 - D(x))]. \quad (22.2)$$

این روش از مدل مولد بازگردنده مبتنی بر نمونه‌گیری (بخش ۲-۱.۲) استفاده می‌کند. گرادین اعمالی به مدل مولد بر اساس رابطه‌ی ۲۱.۲ به صورت زیر محاسبه می‌شود و به عبارتی تبدیل می‌شود که با مونت کارلو قابل تخمین باشد:

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_G &= - \sum_{l=1}^L \mathbb{E}_{x_{:l-1} \sim Q_{\theta}} \left[\sum_{x_l} V_{\theta}(x_{:l}) \nabla_{\theta} q_l(x_l | x_{:l-1}; \theta) \right] \\ &= - \sum_{l=1}^L \mathbb{E}_{x_{:l-1} \sim Q_{\theta}} \left[\sum_{x_l} V_{\theta}(x_{:l}) q_l(x_l | x_{:l-1}; \theta) \nabla_{\theta} \log q_l(x_l | x_{:l-1}; \theta) \right] \\ &= - \sum_{l=1}^L \mathbb{E}_{x_{:l-1} \sim Q_{\theta}} \left[\mathbb{E}_{x_l \sim Q_{\theta} | x_{:l-1}} [V_{\theta}(x_{:l}) \nabla_{\theta} \log q_l(x_l | x_{:l-1}; \theta)] \right] \\ &= - \sum_{l=1}^L \mathbb{E}_{x_{:l} \sim Q_{\theta}} [V_{\theta}(x_{:l}) \nabla_{\theta} \log q_l(x_l | x_{:l-1}; \theta)]. \end{aligned} \quad (23.2)$$

در معادله‌ی ۲۳.۲، $V_{\theta}(x_{:l})$ نشان دهنده‌ی امتیاز نسبت داده شده به زیر دنباله‌ی با l کلمه‌ی اولیه‌ی دنباله است. در حالتی که $l = L$ باشد، مقدار امتیاز مشخص بوده و به صورت $V_{\theta}(x_{:L}) = D(x)$ است. ولی در حالتی که پاداش میانی را می‌خواهیم، به این صورت عمل می‌کنیم که برای تخمین امتیاز $x_{:l}$ ، توسط مدل نمونه‌هایی تولید می‌کنیم که در l کلمه‌ی اول مشابه $x_{:l}$ باشند، از آنجا که دنباله‌های تولید شده دنباله‌های کاملی هستند، با کمک تمیزدهنده امتیازشان مشخص می‌شود. میانگین این امتیازها را به عنوان $V_{\theta}(x_{:l})$ در نظر می‌گیریم. این عملیات در شکل ۲-۱۷ و رابطه‌ی ۲۴.۲ نشان داده شده است.



شکل ۲-۱۷: تخمین پاداش میانی دنباله با روش جستجوی مونت کارلو [۵]

^۱ Monte Carlo Search

$$V_{\theta}(x_{:l}) = \begin{cases} D(x), & \text{if } l = L \\ \frac{1}{M} \sum_{x \in MC^{\theta}(x_{:l}; M)} D(x), & \text{otherwise.} \end{cases} \quad (24.2)$$

که $MC^{\theta}(x_{:l}; M)$ نشان دهنده‌ی مجموعه‌ای M عضوی از دنباله‌های تولید شده توسط مدل است که دارای پیشوند $x_{:l}$ هستند. دنباله‌های این مجموعه دنباله‌هایی با طول کامل‌اند.

از جزئیات روش SeqGAN چند مورد قابل ذکر است، برای بهبود تمیزدهنده از معماری شبکه‌های شاهرایی^۱ [۳۹] بر روی تمیزدهنده استفاده می‌شود. این مدل برای شروع آموزش، مدل مولد را با روش جبر معلم آموزش می‌دهد تا نقطه‌ی شروع تصادفی نداشته باشد. دلیل این موضوع این است که از دید مساله‌ی تعریف شده‌ی یادگیری مقابله‌ای، تعداد عمل‌های ممکن بسیار زیاد است و مساله فضای جستجوی بزرگی دارد و شروع تصادفی در محیطی پیچیده نیازمند زمان زیادی برای آموزش است [۱۲].

۲.۸-۲ روش ORGAN

روش معرفی شده در [۲] با نام ORGAN تعمیمی از روش SeqGAN (بخش ۲-۱.۸) است. هدف این روش وارد کردن دانش خبره به تولید دنباله است. مثلاً زمانی که برای دنباله معیاری داریم که به کیفیت و صحت دنباله تولیدی امتیاز می‌دهد. مثالی از این امتیاز در حوزه تولید دنباله‌های ساختار مولکولی برای دارو است. در این حوزه ویژگی‌هایی مثل حلال بودن در آب و سختی تولید این ساختار در عمل عوامل مهمی هستند. برای ارزیابی این ویژگی‌ها، روش‌هایی وجود دارد که مربوط به خبرگی آن حوزه است. مثلاً بسته‌ی نرم‌افزاری به نام RDKit^۲ وجود دارد که با کمک آن می‌توان به ویژگی حلالیت در آب یک ساختار مولکولی امتیاز داد.

روش ORGAN دو نسخه دارد، نسخه‌ی اول شبیه به روش SeqGAN است و نسخه‌ی دوم برای پایداری آموزش از فاصله‌ی Wasserstein-1 استفاده شده است، در ادامه به جزئیات این دو نسخه از روش می‌پردازیم. نسخه اول کاملاً مشابه SeqGAN عمل کرده با این تفاوت که به پاداش، جمله‌ی جدید اضافه کرده است. در این روش پاداش دنباله‌ی $x_{:l}$ به صورت زیر تعریف می‌شود:

$$V_{\theta}(x_{:l}) = \begin{cases} S(x), & \text{if } l = L \\ \frac{1}{M} \sum_{x \in MC^{\theta}(x_{:l}; M)} S(x), & \text{otherwise} \end{cases} \quad (25.2)$$

$$S(x) = \lambda D(x) + (1 - \lambda) O(x),$$

که $MC^{\theta}(x_{:l}; M)$ نشان دهنده‌ی مجموعه‌ای M عضوی از دنباله‌های تولید شده توسط مدل است که دارای پیشوند $x_{:l}$ هستند و دنباله‌های این مجموعه دنباله‌هایی با طول کامل‌اند. تفاوتی که نسبت به SeqGAN وجود دارد در تابع $S(x)$ است که جمله‌ای بر حسب $O(x)$ به آن اضافه شده است؛ منظور از $O(x)$ امتیازی است که از طریق خبره به دنباله‌ی کامل x داده می‌شود. این روش در حالتی که $\lambda = 0$ است مشابه یادگیری تقویتی بر حسب امتیاز خبرگی می‌شود و در حالتی که $\lambda = 1$ مشابه روش SeqGAN است.

شبکه‌ی استفاده شده به عنوان مولد، مدل مولد بازگردنده مبتنی بر نمونه‌گیری (بخش ۲-۱.۲) بوده و تمیزدهنده نیز براساس مدل دسته‌بند پیچشی (بخش ۲-۲.۳) است.

^۱ Highway Networks

^۲ <https://www.rdkit.org/>

استفاده از فاصله‌ی Wasserstein-1

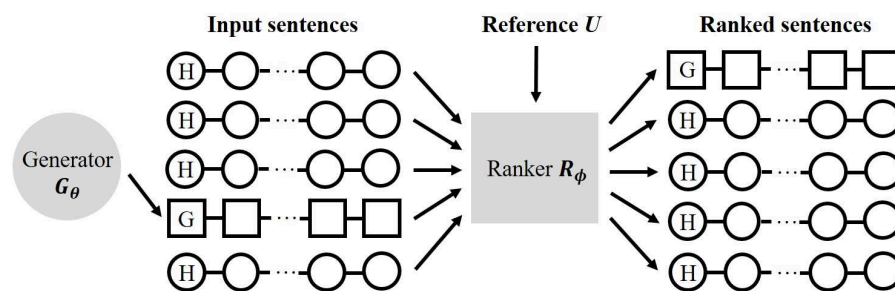
در نسخه‌ی دوم این روش آموزش، مدل مولد مشابه نسخه‌ی اول است و تفاوت در آموزش تمیزدهنده است. تابع هزینه‌ی تمیزدهنده از کارهای [۲۱] و [۲۲] ایده گرفته و به صورت زیر است:

$$\mathcal{L}_D = \frac{1}{K} \sup_{\|D\| \leq K} \mathbb{E}_{x \sim Q}[D(x)] - \mathbb{E}_{x \sim P}[D(x)], \quad (26.2)$$

که $\|D\| \leq K$ به معنی پیوستگی K-Lipschitz تابع $D(x)$ است، برای اعمال این پیوستگی بر روی شبکه‌ی تمیزدهنده راهکاری مبتنی بر اضافه کردن جمله‌ی جریمه‌کننده^۱ به تابع هزینه پیشنهاد شده است [۲۲].

۳.۸-۲ روش RankGAN

روش‌های یادگیر مقابله‌ای که تشریح شد همه محدود به دسته‌بندی دو دسته‌ای به عنوان تمیزدهنده هستند. روش RankGAN [۱۰] به جای استفاده از تمیزدهنده از شبکه‌ای رتبه‌بند^۲ استفاده می‌کند. شبکه‌ی رتبه‌بند نسبت به دسته‌ای از دنباله‌های واقعی رتبه‌بندی می‌کند. این موضوع باعث کمک بیش‌تر به شبکه‌ی مولد می‌شود. روال آموزش این روش به این صورت است که رتبه‌بند در یک گام آموزش داده می‌شود و به سمتی سوق داده می‌شود که رتبه‌ی کم‌تری به داده‌ی مصنوعی دهد، در گام دیگر مولد سعی در به اشتباه انداختن شبکه‌ی رتبه‌بند دارد به طوری که دنباله‌های تولید شده رتبه‌ی بهتری نسبت به دنباله‌های واقعی بگیرند. به نوعی می‌توان گفت این نحوه‌ی آموزش نسبت به روش SeqGAN باعث انتقال اطلاعات بیشتری به مولد برای آموزش می‌شود. نمایشی از ساختار کلی این روش در شکل ۱۸-۲ نشان داده شده است.



شکل ۱۸-۲: نمایش ساختار کلی نحوه‌ی آموزش در روش RankGAN [۱۰]. H نشان‌دهنده‌ی دنباله‌های واقعی و G دنباله‌ی تولید شده توسط مولد است. ورودی رتبه‌بند یک دنباله‌ی تولید شده و مجموعه‌ای از دنباله‌های واقعی است. در شکل حالتی که شبکه‌ی مولد تلاش دارد به داده‌ی تولیدی بالاترین رتبه داده شود و رتبه‌بند را گمراه کند نمایش داده شده است. این مقایسه بر حسب مجموعه‌ی مرجع U است.

RankGAN بر مبنای ایده‌ی یادگیری مقابله‌ای روال آموزش زیر را پیشنهاد می‌دهد:

$$\min_{\phi} \max J(Q_\phi, R_\theta) = \mathbb{E}_{x \sim P}[\log R_\theta(x|U, C^-)] + \mathbb{E}_{x \sim Q_\phi}[\log(1 - R_\theta(x|U, C^+))]. \quad (27.2)$$

^۱ Penalty

^۲ Ranker

در رابطه‌ی ۲۷.۲، عبارت $R_\theta(x|U, y)$ به معنی رتبه‌ی x نسبت به مجموعه نمونه‌های y است. این رتبه‌بندی بر اساس شباهت به مجموعه نمونه‌های U تعیین می‌شود. U مجموعه‌ی مرجعی از دنباله‌های واقعی است که شباهت بیش‌تر به این مجموعه به معنی کیفیت بالاتر نمونه است. دو مجموعه‌ی C^+ و C^- به ترتیب نشان‌دهنده‌ی مجموعه نمونه‌های تولید شده و واقعی است. در ادامه جزئیات تابع R بیان شده است.

شبکه‌ی رتبه‌بند از ساختاری مشابه با شبکه‌های پیچشی [۴۰] استفاده کرده و با استفاده از تبدیل‌های غیرخطی برای هر دنباله‌ی ورودی ویژگی استخراج می‌کند. بخش استخراج ویژگی رتبه‌بند را با $\mathfrak{F}(x)$ نمایش می‌دهیم. در ادامه، شبکه‌ی رتبه‌بند با ویژگی‌های به دست آمده رتبه‌بندی را انجام می‌دهد. شبکه‌ی رتبه‌بند با کمک مجموعه‌ی مرجعی از دنباله‌های واقعی که در اختیار دارد، کار می‌کند. امتیازی که به دنباله‌ی x بر حسب $u \in U$ بین مجموعه جملات C (منظور از مجموعه‌ی C مجموعه‌های C^+ و C^- است) می‌دهد، به صورت زیر تعریف می‌شود:

$$P(x|u, C) = \frac{\exp(\gamma\alpha(x|u))}{\sum_{x' \in C \cup \{x\}} \exp(\gamma\alpha(x'|u))}, \quad (28.2)$$

که $\alpha(x|u)$ نشان‌دهنده‌ی تابع امتیازی است که به دنباله‌ی x بر حسب مرجع u داده می‌شود. γ فوق پارامتر روش است که برای تنظیم میزان همواری رتبه‌بندی استفاده می‌شود. تابع امتیازی که به عنوان $\alpha(x|u)$ در نظر گرفته می‌شود، شباهت کسینوسی ویژگی به دست آمده از دو دنباله است:

$$\alpha(x|u) = \cosine(\mathfrak{F}(x), \mathfrak{F}(u)) = \frac{\mathfrak{F}(x) \cdot \mathfrak{F}(u)}{\|\mathfrak{F}(x)\| \|\mathfrak{F}(u)\|}. \quad (29.2)$$

خروجی نهایی رتبه‌بند با امید روی تمام u های ممکن به صورت زیر به دست می‌آید:

$$\log R_\theta(x|U, C) = \mathbb{E}_{u \in U} \log P(x|u, C), \quad (30.2)$$

که $R_\theta(x|U, C)$ نشان‌دهنده‌ی رتبه‌ی x در بین مجموعه‌ی U است. همان‌طور که گفته شد، رتبه‌بند برای داده‌های واقعی، مجموعه‌ی C را از داده‌های تولیدی نمونه‌گیری کرده و به سمتی می‌رود که رتبه‌ی داده‌ی واقعی افزایش یابد. برای داده‌های تولیدی، مجموعه‌ی C از داده‌های واقعی انتخاب شده و رتبه‌بند به سمتی سوق داده می‌شود که رتبه‌ی داده‌ی تولیدی کم شود، بنابراین تابع هزینه‌ی دسته‌بند به صورت زیر می‌شود:

$$\mathcal{L}_R = -\mathbb{E}_{x \sim P}[\log R_\theta(x|U, C^-)] - \mathbb{E}_{x \sim Q_\phi}[\log(1 - R_\theta(x|U, C^+))]. \quad (31.2)$$

ولی در عمل دید شده که استفاده از تابع هزینه‌ی زیر باعث آموزش بهتر می‌شود:

$$\mathcal{L}_R = -\mathbb{E}_{x \sim P}[\log R_\theta(x|U, C^-)] + \mathbb{E}_{x \sim Q_\phi}[\log R_\theta(x|U, C^+)]. \quad (32.2)$$

مدل مولد بازگردنده مبتنی بر نمونه‌گیری (بخش ۲-۱.۲) به عنوان شبکه‌ی مولد در نظر گرفته شده است. برای حل مشکل گسسته بودن داده‌ها از روش یادگیری تقویتی برای آموزش استفاده شده است و پاداش تعریف شده برای دنباله‌های کامل تولید شده، رتبه‌ی دنباله تولید شده در رتبه‌بند است. روش RankGAN مشابه روش SeqGAN از گرایان سیاست برای آموزش مولد استفاده کرده و گرایان تابع هزینه‌ی مولد به صورت زیر می‌شود:

$$\nabla_\theta \mathcal{L}_G = -\sum_{l=1}^L \mathbb{E}_{x_{:l-1} \sim Q_\theta} \left[\sum_{x_l} V_\theta(x_l) \nabla_\theta q_l(x_l|x_{:l-1}; \theta) \right] \quad (33.2)$$

که تابع ارزش V_θ با جستجوی مونت کارلو به صورت زیر به دست می‌آید:

$$V_\theta(x_{:l}) = \begin{cases} R_\theta(x|U, C^+), & \text{if } l = L \\ \frac{1}{M} \sum_{x \in MC^\theta(x_{:l}; M)} D(x), & \text{otherwise} \end{cases} \quad (34.2)$$

و $MC^\theta(x_{:l}; M)$ نشان دهنده‌ی مجموعه‌ای M عضوی از دنباله‌های تولید شده توسط مدل است که دارای پیشوند $x_{:l}$ هستند.

روش RankGAN نیز برای شروع اولیه نیاز به گام پیش-آموزش مولد دارد.

۴.۸-۲ روش LeakGAN

همانگونه که گفته شد، روش‌هایی مثل SeqGAN به وسیله‌ی یادگیری تقویتی درصدد حل مشکل انتقال گرادیان هستند و برای آموزش شبکه‌ی مولد از خروجی عددی تمیزدهنده به عنوان پاداش استفاده می‌کنند. یکی از مشکلات این روش میزان اطلاعات کمی است که از تمیزدهنده به مولد می‌رسد، به طوری که برای یک دنباله‌ی کوتاه یا طولانی فقط یک عدد برای آموزش مولد وجود دارد. این شرایط مقایسه شود با شبکه‌های مولد مقابله‌ای بر روی داده‌ی پیوسته‌ای مثل تصویر که گرادیان نسبت به تک تک پیکسل‌های تصویر به مولد می‌رسد. از سمت دیگر، در مساله‌ی یادگیری تقویتی تعریف شده، محیط به صورت جعبه سیاه^۱ نیست و محیط مساله (که در واقع همان تمیزدهنده است) شناخته شده است؛ بنابراین می‌توان اطلاعات بیشتری به جای فقط یک عدد از آن استخراج کرد [۱۱]. روش LeakGAN که در [۱۱] معرفی شده است با استفاده‌ی بیشتر از تمیزدهنده، سعی در بهبود آموزش به خصوص برای تولید جملات طولانی‌تر دارد. برای این منظور از یک روش سلسله‌مراتبی^۲ به نام شبکه‌های فئودالی^۳ [۴۱] استفاده شده است؛ این شبکه‌ها در حوزه‌ی یادگیری تقویتی برای محیط‌هایی با پاداش با تاخیر^۴ استفاده می‌شوند. در ادامه ابتدا به معرفی شبکه‌های فئودالی پرداخته و سپس جزئیات روش LeakGAN را بر مبنای آن بیان می‌کنیم.

شبکه‌های فئودالی

در یک مساله‌ی یادگیری تقویتی عامل باید دنباله‌ای از عمل‌ها را انجام دهد. می‌توان این دنباله‌های عمل‌ها را به صورت سطح بالاتر دید و آن‌ها را نتیجه‌ی تصمیم‌گیری‌های کلی‌تری دانست. به عبارت دیگر برای حل مساله، دو شبکه وجود داشته باشد. شبکه‌ی اول تصمیم‌های سطح بالا و بدون جزئیات در طول زمان را مشخص کند و شبکه‌ی دوم با توجه به تصمیم شبکه‌ی اول، عمل‌های نهایی مساله را انجام دهد. در شبکه‌های فئودالی چنین هدفی دنبال می‌شود. شبکه‌ی اول که مدیر^۵ نام دارد کلیات عمل را مشخص و برای شبکه‌ی دوم زیرهدف^۶ هایی تعیین می‌کند. شبکه‌ی دوم، کارگر^۷ نام دارد و عمل‌های نهایی را با توجه به زیر هدف‌های مشخص شده از سمت مدیر انجام می‌دهد. راهکاری برای این دیدگاه توسط [۴۱] ارائه شده است که در ادامه آن را شرح می‌دهیم.

^۱ Black Box

^۲ Hierarchical

^۳ Feudal Net

^۴ Delayed Reward

^۵ Manager

^۶ Sub-goal

^۷ Worker

برای انجام روال گفته شده، به یک فضای نهان نیاز است که توصیف کننده‌ی وضعیت محیط باشد. در این فضای نهان، وضعیت در زمان t با s_t مشخص می‌شود و g_t نشان دهنده‌ی خروجی شبکه‌ی مدیر برای این زمان است. g_t برداری در فضای نهان است و نشان دهنده‌ی عمل مدیر برای تغییر و رفتن به وضعیت بهتر است. یعنی در حالت ایده‌آل، اگر در زمان t وضعیت s_t باشد، مدیر با انتخاب g_t ، هدف رفتن به وضعیت $s_t + g_t$ را دارد. بنابراین، روال آموزش شبکه‌ی کارگر مشخص بوده و آموزش آن در راستای پیروی از تصمیم‌های مدیر است. شبکه‌ی کارگر با روش گرادیان سیاست، طبق پاداش زیر آموزش داده می‌شود:

$$R_t^I = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}^I \quad (35.2)$$

$$r_t^I = \frac{1}{c} \sum_{i=0}^c \cos(\sin(s_t - s_{t-i}, g_{t-i})),$$

که R_t^I نشان دهنده‌ی پاداش است، γ و c فوق‌پارامترهای روش است. γ مقداری بین صفر و یک می‌گیرد. c نشان دهنده‌ی میزان دقت زمانی تصمیمات مدیر بوده که در آموزش مدیر هم دخیل است و در ادامه بیشتر توضیح داده می‌شود.

برای آموزش مدیر نیز از روش گرادیان سیاست استفاده می‌شود، به این صورت که پاداش، همان پاداش اصلی مساله است و در آموزش می‌خواهیم مدیر با تصمیمات خود کارگر را به سمتی برود که پاداش افزایش یابد. آموزش مدیر با گرادیان سیاست به سادگی ممکن نیست، زیرا بعد از مشخص شدن عمل مدیر، تصمیم کارگر دخیل است تا پاداش مشخص شود. به همین دلیل در آموزش مدیر فرض می‌شود که کارگر عملکرد خوبی دارد و با فرض توزیعی برای رفتار کارگر، گرادیان آموزش مدیر به صورت زیر می‌شود:

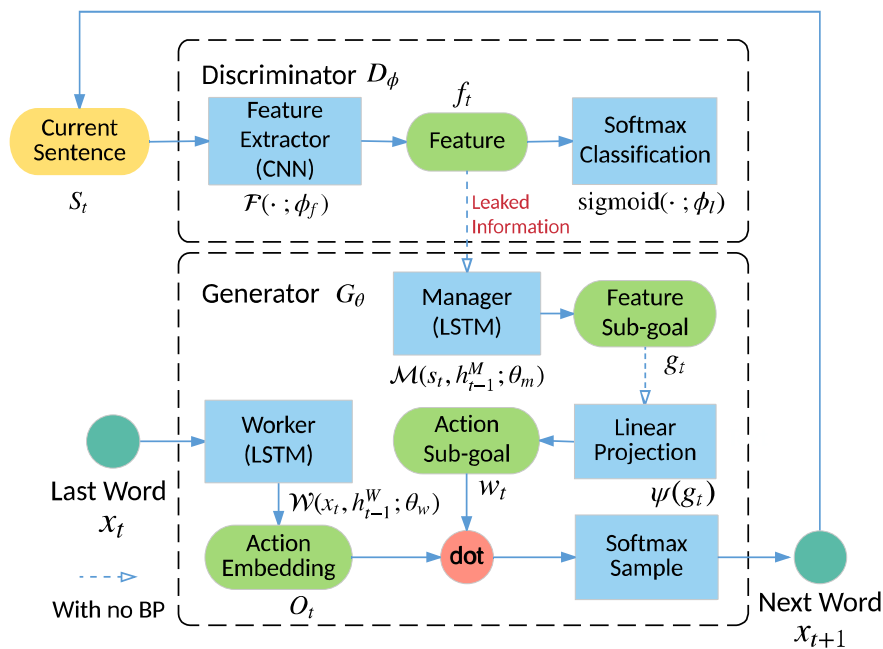
$$\nabla_{\theta} \mathcal{L}_G = R_t \nabla \cos(\sin(s_{t+c} - s_t, g_t)). \quad (36.2)$$

در عبارت ۳۶.۲، R_t نشان دهنده‌ی پاداش اصلی از لحظه‌ی t به بعد است. c نشان دهنده‌ی میزان تاثیر عمل مدیر است و مدیر، عمل‌های خود را برای وضعیت در c مرحله‌ی آینده مشخص می‌کند. در کل روال آموزش به این صورت است که دو شبکه‌ی مدیر و کارگر به صورت تکرارشونده آموزش می‌بینند.

جزئیات روش

روش LeakGAN [۱۱] برای استفاده‌ی بیشتر از تمیزدهنده، از شبکه‌های فتودالی به عنوان مولد استفاده می‌کند. یعنی شبکه‌ی مدیر، ویژگی‌های کلی دنباله‌ای را که مولد تولید می‌کند، مشخص کرده و شبکه‌ی کارگر، تولید کلمات را بر این اساس انجام می‌دهد. بقیه‌ی موارد مشابه دیگر شبکه‌های مبتنی بر یادگیری مقابله‌ای بوده که در یک گام تمیزدهنده آموزش داده می‌شود و در گام دیگر مولد.

در این روش فضای ویژگی‌ای که برای شبکه‌های فتودالی لازم است، بر حسب شبکه‌ی تمیزدهنده به دست می‌آید. شبکه‌ای از نوع مدل دسته‌بند پیچشی (بخش ۲-۲.۳) به عنوان تمیزدهنده استفاده می‌شود و ویژگی‌های تولید شده توسط این شبکه به عنوان فضای نهان در نظر گرفته می‌شود. به این ترتیب از وضعیت داخلی تمیزدهنده در آموزش استفاده می‌شود. یعنی مدیر پیش‌بینی می‌کند که تغییر در فضای ویژگی‌های تمیزدهنده در چه سمتی به تولید دنباله‌های واقعی‌تر منجر می‌شود و کارگر با کمک مدیر، دنباله‌هایی تولید می‌کند که این امر محقق شود. شبکه‌ی کارگر و مدیر هر کدام یک شبکه‌ی LSTM هستند. نمای کلی روش در شکل ۲-۱۹ نشان داده شده است.



شکل ۲-۱۹: ساختار کلی روش LeakGAN [۱۱]

پاداش اصلی مساله که در رابطه‌ی ۳۶.۲ با R_t نشان داده شده است، مشابه روش SeqGAN با کمک جستجوی مونت کارلو تخمین زده می‌شود:

$$R_l = \begin{cases} D(x), & \text{if } l = L \\ \frac{1}{M} \sum_{x \in MC^\theta(x_{:l}; M)} D(x), & \text{otherwise.} \end{cases} \quad (37.2)$$

که $MC^\theta(x_{:l}; M)$ نشان دهنده‌ی مجموعه‌ای M عضوی از دنباله‌های تولید شده توسط مدل است که دارای پیشوند $x_{:l}$ هستند، دنباله‌های این مجموعه دنباله‌هایی با طول کامل‌اند.

روش LeakGAN از تکنیک‌هایی مثل رتبه‌بندی در روش RankGAN، پیش-آموزش و غیره برای آموزش استفاده می‌کند. یکی از مشکلات این روش پیچیدگی آن است که همزمان سه شبکه در طی آموزش وجود دارد. همان‌طور که گفته شد، ویژگی‌های شبکه‌ی تمیزدهنده به عنوان فضای نهان مورد استفاده برای شبکه‌ی فتودالی در نظر گرفته می‌شود. از آنجا که تمیزدهنده در حال تغییر و آموزش است، فضای نهان مدام در حال تغییر است و این موضوع می‌تواند منجر به ناپایداری روش شود [۱۱].

۵.۸-۲ روش MaliGAN

روش معرفی شده در [۱۲] که MaliGAN نام دارد، با تعریف تابع هدفی جدید به نحوی آموزش شبیه به روش SeqGAN می‌رسد، ولی با تفاوت‌هایی که دارد آموزش آن پایداری بیشتری پیدا می‌کند. همچنین این روش در مقایسه با روش پایه‌ی جبر معلم مقاومت بیشتری نسبت به مشکل بیش‌برازش^۱ دارد [۱۲].

تابع هدف تمیزدهنده، مشابه شبکه‌های مولد مقابله‌ای، تابع هزینه‌ی دسته‌بند عادی است. ولی تابع هدفی که برای

^۱ Overfit

مولد استفاده می‌شود برابر $\mathcal{D}_{KL}(\tilde{P}||Q)$ است، که \tilde{P} نشان‌دهنده‌ی یک تخمین از توزیع واقعی است. برای یافتن توزیع تخمینی \tilde{P} از قضیه‌ی ۱.۲ استفاده شده است. طبق این قضیه در حالتی که تمیزدهنده بهینه است، داریم:

$$\begin{aligned} D^*(x) &= \frac{p_{data}(x)}{p_{data}(x) + q(x)} \\ \Rightarrow p_{data}(x) &= \frac{D^*(x)}{1 - D^*(x)} q(x), \end{aligned} \quad (38.2)$$

طبق این رابطه، \tilde{P} به عنوان تقریبی از توزیع واقعی به کمک تمیزدهنده و چگالی توزیع فعلی مولد (که با $q(x)$ مشخص می‌شود) به دست می‌آید، چگالی توزیع تقریبی را با $\tilde{p}(x)$ نشان می‌دهیم که به صورت زیر می‌شود:

$$\tilde{p}(x) = \frac{D(x)}{1 - D(x)} q(x). \quad (39.2)$$

برای آموزش مولد، بعد از آموزش تمیزدهنده از شبکه‌ی مولد فعلی رونوشت^۱ گرفته می‌شود. تابع چگالی و توزیع رونویسی شده به ترتیب با $q'(x)$ و Q' نشان داده می‌شود. سپس شبکه‌ی مولد با تابع هدف $\mathcal{D}_{KL}(\tilde{P}||Q)$ آموزش می‌بیند و $\tilde{p}(x)$ براساس Q' با کمک رابطه‌ی ۳۹.۲ به دست می‌آید. بنابراین تابع هزینه‌ی مولد به صورت زیر می‌شود:

$$\begin{aligned} \mathcal{L}_G &= \mathcal{D}_{KL}(\tilde{P}||Q_\theta) \\ \tilde{p}(x) &= \frac{1}{Z(\theta)} \frac{D(x)}{1 - D(x)} q'(x) \end{aligned} \quad (40.2)$$

که در رابطه‌ی گفته شده برای اطمینان از توزیع بودن \tilde{P} ضریب نرمال‌سازی در نظر گرفته شده است. روش MaliGAN برای استفاده از تابع هدف بیان شده، با کمک نمونه‌گیری براساس اهمیت^۲ به جای نمونه‌گیری از توزیع \tilde{P} ، نمونه‌گیری از مدل مولد را انجام می‌دهد. با این راهکار، گرادیان تابع هزینه به صورت زیر محاسبه می‌شود:

$$\begin{aligned} \nabla_\theta \mathcal{L}_G &= \nabla_\theta \mathcal{D}_{KL}(\tilde{P}||Q_\theta) \\ &= \mathbb{E}_{x \sim \tilde{P}} \nabla_\theta \log q_\theta(x) \\ &= \mathbb{E}_{x \sim Q'} \left[\frac{\tilde{p}(x)}{q'(x)} \nabla_\theta \log q_\theta(x) \right] \\ &= \frac{1}{Z(\theta)} \mathbb{E}_{x \sim Q'} \left[\frac{D(x)}{1 - D(x)} \nabla_\theta \log q_\theta(x) \right], \end{aligned} \quad (41.2)$$

که در گام آخر از عبارت ۳۹.۲ استفاده شده است. روش MaliGAN به جای نمونه‌گیری از Q' ، نمونه‌گیری را از مدل مولد به روز انجام می‌دهد و با نمونه‌های آن گرادیان تابع هزینه را تخمین زده و آموزش مولد انجام می‌شود. با داشتن مجموعه‌ی $\{x^{(i)}\}_1^m$ از نمونه‌های مولد، تخمین به صورت زیر می‌شود:

$$\begin{aligned} \nabla_\theta \mathcal{L}_G &\approx \frac{\sum_{i=1}^m r_D(x^{(i)}) \nabla_\theta \log q_\theta(x)}{\sum_{i=1}^m r_D(x^{(i)})} \\ r_D(x) &= \frac{D(x)}{1 - D(x)} \end{aligned} \quad (42.2)$$

^۱ Copy

^۲ Importance Sampling

همان‌طور که در عبارت ۴۲.۲ دیده می‌شود، نحوه‌ی به روزرسانی شبکه‌ی مولد، شبیه به روش SeqGAN است، ولی نمونه‌ها هرکدام به وسیله‌ی تمیزدهنده، وزن گرفته‌اند. تابع هزینه‌ی ارائه شده علاوه بر افزایش درست‌نمایی رفتارهای خوب، درست‌نمایی رفتارهای نادرست را هم کاهش می‌دهد [۱۲].

۹-۲ یادگیری مقابله‌ای با رویکرد تولید دنباله‌ی «توزیع کلمه»

رویکری که در روش‌های گفته شده برای تولید دنباله وجود داشته دیدن هر کلمه‌ی دنباله به عنوان یک داده‌ی گسسته و نمایش آن با یک بردار یک-فعال است که این موضوع باعث ایجاد مشتق‌ناپذیری مدل می‌شود. می‌توان کلمات دنباله را حالتی کلی‌تر در نظر گرفت، به این صورت که هر کلمه با یک بردار با اندازه‌ی V بیان شود به طوری که جمع عناصر آن یک شده و یک دنباله‌ی با طول l به وسیله‌ی l بردار V تایی بیان شود. در این حالت اگر بردارها به صورت یک-فعال شوند، به همان نمایش مورد استفاده قبل می‌رسیم. در واقع هدف این است که روی هر کلمه‌ی دنباله توزیعی را تولید شود. با این بیان جدید، برای تولید دنباله‌های معتبر، روش باید یاد بگیرد بردار مربوط به هر کلمه را به سمت یک-فعال شدن سوق دهد [۱۳، ۲۲، ۴۲]. این نحوه‌ی حل مساله‌ی تولید دنباله، تولید دنباله‌ی «توزیع کلمه» نامیده می‌شود.

رویکرد تعریف شده با این که مشکل مشتق‌ناپذیری را ندارد با روش عادی شبکه‌های مولد مقابله‌ای قابل آموزش نیست، که دو دلیل برای آن ذکر شده است:

- طبق نتیجه‌ی ۱ در بخش ۲-۱.۵ فاصله‌ای که شبکه‌های مولد مقابله‌ای بین دو توزیع مولد و داده‌ی واقعی کم می‌کند، فاصله‌ی Jensen-Shannon است. این فاصله زمانی که دو توزیع با هم تفاوت زیادی دارند و توزیع آن‌ها در فضای نمونه‌ها اشتراک کمی دارد، اشباع^۱ می‌شود و مشتق آن صفر می‌شود. از طرفی داده‌ی اصلی بردارهای یک-فعال است و نمونه‌های تولید شده در فضای خیلی بزرگ‌تری قرار می‌گیرند که نمونه‌های یک-فعال بخش کوچکی از آن است. این موضوع باعث می‌شود که فاصله‌ی این دو توزیع زیاد شود و مشکل اشباع شدن تابع هزینه پیش آید، بنابراین گرادیان به شبکه برنگردد و آموزش انجام نشود [۲۲].
- در عمل هم تفاوت زیاد دو دسته داده باعث مشکل می‌شود. به این صورت که داده‌های واقعی یک-فعال هستند و این تفاوت بارز توسط دسته‌بند تشخیص داده می‌شود. در این حالت دسته‌بند به عنوان تمیزدهنده تمام داده‌های تولیدشده را به راحتی با این ویژگی تشخیص می‌دهد و گرادیان بی‌معنی به مولد برمی‌گردد [۲۲].

از طرف دیگر روش جدیدی در حوزه‌ی شبکه‌های مولد مقابله‌ای به نام WGAN [۲۱] معرفی شده است، که مشکلات گفته شده برای حالتی که دو توزیع از هم دور هستند در آن کمتر است و گرادیان آموزش در حالتی که دو توزیع از هم دور هم هستند، قابلیت آموزش مولد را دارد. این روش بر مبنای فاصله‌ی Wasserstein است و این فاصله را کاهش می‌دهد. تابع هزینه‌ی تعریف شده برای این روش شرطی دارد که برقراری آن سخت است. در این روش شبکه‌ی تمیزدهنده باید خاصیت پیوستگی K-Lipschitz را داشته باشد. راهکارهایی برای برقراری این پیوستگی مطرح شده است. راهکار پر استفاده از این بین، اضافه کردن جمله‌ی جریمه^۲ به تابع هزینه‌ی شبکه‌ی دسته‌بند است که به نام

^۱ Saturate

^۲ Penalty

WGAN-GP [۲۲] شناخته می‌شود. رابطه‌ی به‌روزرسانی شبکه‌ی مولد و تمیزدهنده در این روش به صورت زیر است:

$$\begin{aligned}\mathcal{L}_D &= \mathbb{E}_{x \sim Q_\theta}[D(x)] - \mathbb{E}_{x \sim P}[D(x)] + \lambda \mathbb{E}_{x \sim O}[(\|\nabla_x D(x)\|_2 - 1)^2] \\ \mathcal{L}_G &= -\mathbb{E}_{x \sim Q_\theta}[D(x)],\end{aligned}\quad (۴۳.۲)$$

که جمله‌ی آخر در تابع هزینه‌ی تمیزدهنده جریمه‌ای برای کنترل پیوستگی K-Lipschitz است. نمونه‌های توزیع O به این صورت ایجاد می‌شوند که نمونه‌ای از توزیع P و نمونه‌ای از توزیع Q در نظر گرفته می‌شود، بین این دو نمونه خطی رسم شده و با نمونه‌گیری یکنواخت از این خط نمونه‌های توزیع O به دست می‌آید [۲۲]. در ادامه دو روش را که به‌وسیله‌ی WGAN-GP مساله تولید دنباله را به‌صورت تولید «توزیع کلمه» حل می‌کنند، بررسی می‌کنیم.

- روش اول که در [۲۲] معرفی شده است، یک شبکه‌ی پیچشی را به عنوان مولد در نظر می‌گیرد که تابع تبدیلی از فضای نهان به دنباله‌ای از «توزیع کلمه» است. به عبارت دیگر برداری از نمونه‌های $z \sim N(0, 1)$ را به ماتریسی $L \times V$ تبدیل می‌کند. شبکه‌ی مولد نیز با ساختار پیچشی یک بعدی مدل شده است. این روش روی داده‌های طبیعی، بررسی شده و نتایج مطلوبی نداشته است [۱۳، ۲۲، ۴۳]. این روش در مقاله‌ی WGAN-GP معرفی شده و به همین دلیل با نام WGAN-GP می‌شناسیم.
- روش دوم، در مدل پایه با تغییر مدل مولد و تمیزدهنده به شبکه‌ی بازگردنده‌ی GRU [۴۴] نتایج نامطلوبی کسب می‌کند [۱۳]. در ادامه با تکنیک‌هایی مثل آموزش گام به گام مدل (در گام اول دنباله‌های با طول یک، در گام بعد دنباله‌های با طول دو و به همین صورت تا رسیدن به بیشینه اندازه‌ی مدنظر) کیفیت دنباله‌های تولیدی را بهبود می‌بخشد [۱۳]. این روش را VDG-GRU می‌نامیم.

این روش‌ها در فاز آزمون برای تولید دنباله‌ها نیازمند تبدیل توزیع کلمه به بردار یک-فعال هستند که برای این کار بیشینه‌گیری انجام می‌دهند [۱۳، ۲۲]. از آن‌جا که بیشینه‌گیری در فاز آموزش اتفاق نمی‌افتد این موضوع خود مشکلی است که باعث اختلاف در فاز آموزش و آزمون می‌شود. نکته‌ای که در رابطه‌ی دو روش گفته شده وجود دارد، عدم استفاده از پیش-آموزش برای تولید نمونه‌ها است. اکثر روش‌های دیگر مبتنی بر یادگیری تقویتی بدون پیش-آموزش نتایج بهتری از دنباله‌های تصادفی تولید نمی‌کنند.

۱۰-۲ جمع‌بندی

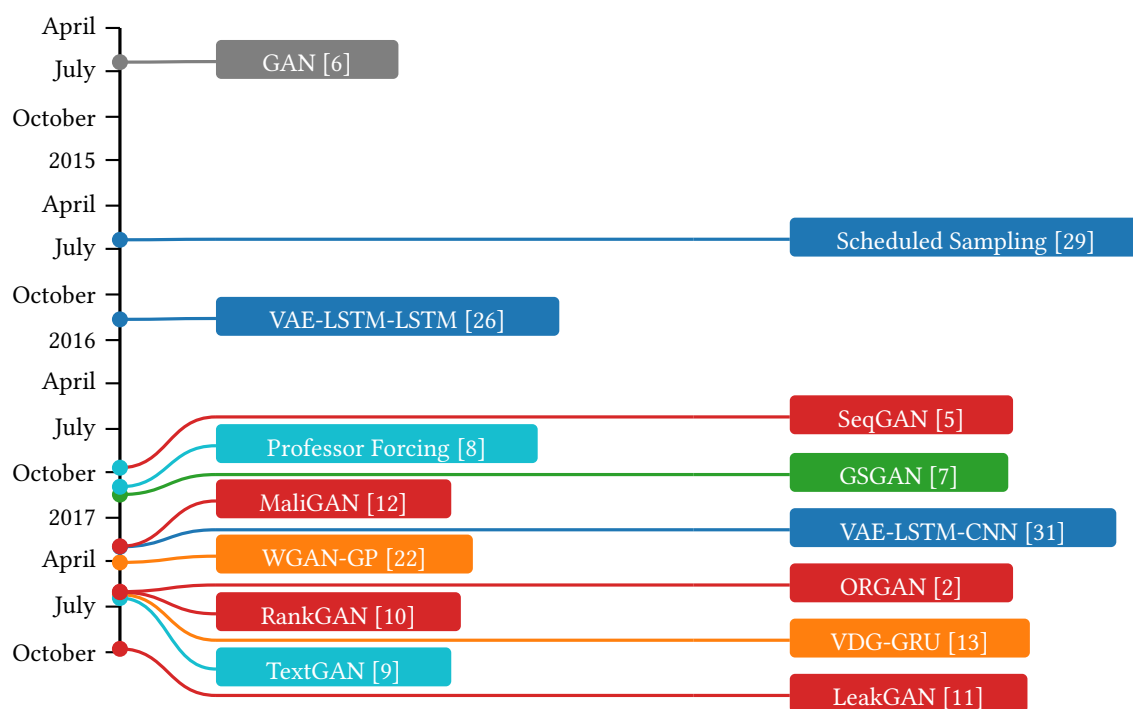
در این فصل به روش‌های موجود در حوزه‌ی تولید دنباله پرداختیم. برای این منظور، ابتدا ساختارهای شبکه‌عصبی مورد استفاده در این حوزه را معرفی کردیم. پس از آن، به بیان روش‌های مبتنی بر بیشینه‌درست‌نمایی پرداخته و مشکل آریبی مواجهه که در روش پایه‌ی جبر معلم وجود دارد، تشریح شد. سپس با بیان جزئیات شبکه‌های مولد مقابله‌ای، در چهار دسته به تشریح روش‌های مبتنی بر یادگیری مقابله‌ای در حوزه‌ی تولید دنباله پرداخته‌ایم. همانطور که گفته شد مشکل اصلی استفاده از شبکه‌های مولد مقابله‌ای در انتقال گرادیان است و هر یک از این چهار دسته راهکاری برای حل آن ارائه می‌دهند. دسته‌ی اول با تقریبی فضای گسسته مساله را به صورت یک فضای پیوسته درآورده است. دسته‌ی دوم با کار در فضای ویژگی این مشکل را حل کرده‌اند. سومین دسته که بیشتر روش‌های

پیشین را در برمی‌گیرد، مساله‌ی آموزش مولد را به یک مساله‌ی یادگیری تقویتی تبدیل کرده‌اند و بدون مشکل انتقال گرادیان، تکنیک‌های این حوزه را برای حل مساله به کار بسته‌اند. در دسته‌ی آخر روش‌هایی تشریح شد که با تغییر تعریف صورت مساله، مشکل انتقال گرادیان را حل می‌کنند و به جای دنباله‌ی گسسته یک دنباله از توزیع‌های دسته‌ای تولید می‌کنند.

در روش‌های پیشینه درست‌نمایی علاوه بر روش پایه‌ی جبر معلم، دو روش که به وسیله‌ی آن فضای نهانی برای دنباله‌ها ایجاد می‌شود، بیان شد.

در روش‌های مبتنی بر یادگیری تقویتی، ابتدا روش اولیه بیان شد که با در نظر گرفتن مولد به عنوان عامل مساله‌ی یادگیری تقویتی و تعریف پاداش بر اساس خروجی تمیزدهنده (میزان به اشتباه افتادن تمیزدهنده) مساله را حل می‌کند. سپس تعمیمی از این روش تشریح شد که دانش خبرگی را در پاداش اضافه می‌کند. رویکردی دیگر گفته شد که به جای تمیزدهنده از یک رتبه‌بند استفاده می‌شود و با این کار پاداش غنی‌تری به عامل (مولد) می‌رسد. بعد از آن، راه‌کاری معرفی شد که از شبکه‌های فئودالی به عنوان مولد استفاده می‌کند، این رویکرد که در یادگیری تقویتی مطرح شده برای مدل کردن عامل به صورت سلسله‌مراتبی است. در پایان روش‌های این دسته، راه‌کاری توضیح داده شد که با تعریف تابع هدفی جدید به رویکردی مشابه دیگر روش‌های یادگیری مقابله‌ای رسیده است.

در ادامه نمایش زمانی روش‌های پیشین نشان داده شده است. زمان‌های این نمایش، براساس اولین انتشار مقاله بر روی شبکه‌ی Arxiv مشخص شده و دسته‌بندی چهارگانه روش‌های پیشین با رنگ از یکدیگر تفکیک شده است.



شکل ۲-۲۰: نمایش زمانی روش‌های پیشین ارائه‌شده در حوزه‌ی تولید دنباله

فصل ۳

معیارهای ارزیابی پیشین و پیشنهادی

مطالب این فصل

۴۵	۱-۳ مقدمه
۴۶	۲-۳ معیارهای مبتنی بر احتمال مدل
۴۷	۳-۳ معیارهای مبتنی بر داده‌ی مصنوعی
۵۰	۴-۳ معیارهای مبتنی بر n -گرام
۵۳	۵-۳ معیار پیشنهادی - Fréchet BERT Distance
۵۴	۶-۳ جمع‌بندی

ارزیابی مدل‌های مولد به تنهایی چالش بوده و پژوهش‌های زیادی روی این موضوع انجام شده است. در این فصل معیارهای ارزیابی پیشین تشریح می‌شود، مشکلات آنها بررسی شده و معیارهای جدیدی برای ارزیابی معرفی می‌شود.

۱-۳ مقدمه

از بین معیارهای مورد استفاده در پژوهش‌های پیشین، دسته‌ای از معیارها فقط کیفیت جملات را در نظر می‌گیرد و دسته‌ای دیگر فقط تنوع جملات تولیدی را ارزیابی می‌کنند. این موضوع باعث می‌شود که مقایسه‌ی دو مدل به راحتی میسر نشود. در این فصل علاوه بر تشریح معیارهای پیشین، چهار معیار جدید معرفی شده است. دو مورد از این معیارها در ارزیابی با داده‌ی مصنوعی (بخش ۲.۳-۳) و یک مورد در حوزه‌ی ارزیابی دنباله‌ها از دید n -گرام است (بخش ۳.۴-۳). در آخر معیاری خاص دنباله‌های زبان طبیعی پیشنهاد شده (بخش ۵-۳) که براساس معیاری معتبر در حوزه‌ی ارزیابی تصاویر بنا نهاده شده است.

لازم به ذکر است که معیارهای خاص یک حوزه فقط محدود به زبان طبیعی نیست. برای مثال دنباله‌هایی که روی ساختار مولکولی هستند، می‌توان با خواص شیمیایی مثل پایداری آن، ارزیابی شود. حتی این ارزیابی می‌تواند محاسباتی برای شبیه‌سازی رفتار ساختار تولید شده باشد. یا به عنوان مثالی دیگر، برای دنباله‌های نت موسیقی

می‌توان با خبرگی معیارهای برای طبیعی بودن آن تعریف کرد [۲]. برای مشخص شدن معیارهای معرفی شده در این پژوهش، عنوان بخش‌های مربوط به معیار پیشنهادی با پیشوند «معیار پیشنهادی» مشخص شده‌اند.

۲-۳ معیارهای مبتنی بر احتمال مدل

قدیمی‌ترین معیار مورد استفاده معیارهایی بوده که بر اساس درست‌نمایی مدل بنا نهاده شده است، این معیار مشکلاتی دارد. بزرگترین مشکل این معیار در ارزیابی روش‌های جدید ناعادلانه بودن آن است، زیرا مدل پایه این معیار را به عنوان تابع هدف در نظر می‌گیرد و کمینه می‌کند. از این رو، قرار دادن این معیار به عنوان مبنای قضاوت به مقایسه‌ای عادلانه منجر نمی‌شود. در ادامه دو معیار معروف از این دسته معرفی شده است و مشکلات آن به صورت جزئی مورد بررسی قرار می‌گیرد.

۱.۲-۳ منفی لگاریتم درست‌نمایی

معیار منفی لگاریتم درست‌نمایی^۱ که به اختصار NLL می‌نامیم، فقط برای مدل‌های احتمالی قابل استفاده است؛ یعنی باید بتوان احتمالی که مدل به دنباله می‌دهد را محاسبه کرد. معیار منفی لگاریتم درست‌نمایی برابر منفی لگاریتم میزان احتمالی است که مدل به داده‌های واقعی می‌دهد؛ بنابراین کمتر بودن آن به معنی نتیجه‌ی بهتر است. این معیار به صورت زیر تعریف می‌شود:

$$\text{NLL} = -\frac{1}{N} \sum_{n=1}^N \log q(x^{(n)}), \quad (1.3)$$

که $q(x)$ بیانگر احتمالی است که مدل به نمونه‌ی x می‌دهد و $x^{(n)}$ نمونه‌هایی است که از توزیع اصلی در اختیار داریم.

مشکلاتی برای این معیار وجود دارد که به شرح زیر است:

- فقط برای مدل‌های احتمالی قابل استفاده است.
- این معیار در روش‌های تولید دنباله که با کمک قاعده‌ی زنجیره‌ای دنباله‌ها را مدل می‌کنند حساسیتی نسبت به مشکل اُریبی مواجهه (بخش ۲-۱.۴) ندارد. زیرا برای محاسبه‌ی این معیار همواره نمونه‌هایی از داده واقعی وارد مدل شده و توزیع احتمال شرطی به شرط دنباله‌ی صحیح بررسی می‌شود. از این رو روش‌هایی که مدعی برطرف کردن مشکل اُریبی مواجهه هستند، به این معیار اهمیت کمتری می‌دهند.
- مشکلی دیگری که وجود دارد امتیاز پایینی است که این معیار به روش‌هایی با رفتار میانگین-جستجوگری می‌دهد. در نظر بگیرید که یک مدل آموزش دیده و به اکثر داده‌های واقعی احتمال بالایی نسبت داده ولی به بخش کوچکی از داده‌های واقعی احتمالاً نزدیک صفر نسبت دهد. در این حالت معیار منفی لگاریتم

^۱ Negative Log Likelihood

درست‌نمایی حاصل جمع عباراتی است که یکی از این عبارات به مثبت بی‌نهایت میل می‌کند و اثر بقیه جملات از بین می‌رود. به همین دلیل ممکن است امتیاز مدل آموزش دیده که بخشی از توزیع واقعی را آموزش دیده از مدلی که تصادفی به تمام جملات احتمال یکسان می‌دهد کمتر شود. به بیان دیگر، در عمل عبارت $\log \prod_{x_i} q(x_i)$ برای محاسبه‌ی معیار محاسبه می‌شود، که با نزدیک به صفر شدن یکی از $q(x_i)$ ها کل مقدار سمت راست تساوی به سمت صفر می‌رود.

- این معیار خود تابع هزینه‌ی بعضی از روش‌ها است؛ بنابراین مقایسه روش‌هایی که این تابع هزینه را به عنوان هزینه در نظر می‌گیرند، در مقابل روش‌هایی که از این تابع هزینه استفاده نمی‌کنند (مثل شبکه‌های مولد مقابله‌ای) غیرعادلانه است.

۲.۲-۳ Perplexity

این معیار کاملاً مشابه معیار منفی لگاریتم درست‌نمایی است، با این تفاوت که به‌نوعی رفتار لگاریتمی آن حذف شده است. این معیار در حالت عادی به صورت زیر تعریف می‌شود:

$$PP = b^{-\frac{1}{N} \sum_{n=1}^N \log_b q(x^{(n)})}, \quad (2.3)$$

که معمولاً b یکی از مقادیر ۲ یا e است.

اگر در رابطه‌ی ۲.۳ به جای N که تعداد نمونه‌ها (دنباله‌ها) را نشان می‌دهد، تعداد کلمات کل دنباله‌های جایگذاری شود و نسبت به تعداد کلمات نرمال شود، آنگاه معیار حاصل Perplexity Per Word نام دارد.

۳-۳ معیارهای مبتنی بر داده‌ی مصنوعی

این دسته از معیارها برای سنجش آموزش بر روی مجموعه داده‌ی واقعی که در اختیار داریم قابل استفاده نیستند. این معیارها فقط برای مقایسه‌ی مدل‌ها در مقابل همدیگر کاربرد دارد. در این روش ارزیابی، داده‌های مصنوعی تولید می‌شود که ویژگی‌های آن را به خوبی می‌دانیم و با آموزش مدل‌ها بر روی این داده‌ها، کیفیت مدل آموزش دیده بررسی می‌شود. در ادامه چند نمونه از این معیارهای ارزیابی بیان شده است.

۱.۳-۳ منفی لگاریتم درست‌نمایی پیشگو

این روش ارزیابی یکی از پراستفاده‌ترین روش‌ها در بررسی مدل‌های مولد دنباله است. در این معیار، برای تولید داده‌های مصنوعی توزیع احتمالی به صورت تصادفی می‌سازیم؛ با نمونه‌گیری از این توزیع، داده‌های مصنوعی ساخته می‌شود. در حوزه‌ی دنباله یکی از راهکارهای ساخت توزیع احتمال در نظر گرفتن شبکه‌ی عصبی با مقادیر پارامتر تصادفی به عنوان توزیع هدف است. در مقاله‌ی [۵] از یک شبکه‌ی LSTM با مقادیر پارامتر تصادفی استفاده می‌شود؛ آن توزیع را، پیشگو^۱ می‌نامیم. بعد از آموزش مدل به‌وسیله‌ی داده‌های مصنوعی، ارزیابی به این صورت

^۱ Oracle

انجام می‌شود که تعدادی داده با نمونه‌گیری از مدل به دست می‌آید؛ داده‌های تولید شده به شبکه‌ی پیشگوی ورودی داده می‌شود و به وسیله‌ی آن مقدار $p(x)$ محاسبه می‌شود. با کمک پیشگو، درست‌نمایی داده‌های تولیدی از نظر پیشگو محاسبه می‌شود و به عنوان معیار در نظر گرفته می‌شود:

$$\text{NLL-Oracle} = -\mathbb{E}_{x \sim Q} \log p(x), \quad (۳.۳)$$

که این معیار منفی لگاریتم درست‌نمایی پیشگو یا به اختصار NLL-Oracle نامیده می‌شود. از مشکلات این معیار عدم توجه به تنوع نمونه‌های تولید شده توسط مدل است. به این صورت که اگر مدل فقط یک نمونه تولید کند و آن یک نمونه، نمونه‌ی با کیفیتی از دید پیشگو باشد، امتیاز مدل بسیار بالا می‌شود؛ در حالی که چنین مدلی مطلوب نیست. اشکال دیگری که به این روش وارد است، در رابطه با نحوه‌ی ساخت توزیع پیشگو است. زمانی که توزیع پیشگو با شبکه‌ی LSTM تصادفی ساخته می‌شود، کنترلی بر روی پیچیدگی دنباله‌های تولید شده وجود ندارد [۴۲].

۲.۳-۳ معیار پیشنهادی - فاصله با پیشگو

همانطور که گفته شد در مقالات حوزه‌ی تولید دنباله معیار درست‌نمایی پیشگو پراستفاده است؛ در حالی که این معیار با در نظر نگرفتن تنوع داده‌ی تولیدی می‌تواند اشکالاتی جدی ایجاد کند. مثلاً احتمال بروز مشکل چسبیدگی به قله در شبکه‌های مولد مقابله‌ای زیاد است و اگر این اتفاق روی دهد، این معیار نه تنها جریمه‌ای نمی‌کند بلکه امتیاز بالایی نسبت می‌دهد.

در بیشتر روش‌ها، مدل مولد خود یک مدل احتمالی است و روش منفی لگاریتم درست‌نمایی پیشگو از این موضوع سودی نمی‌برد. در این پژوهش استفاده از فاصله‌های متقارن بین مدل آموزش دیده و پیشگو، برای ارزیابی مدل پیشنهاد می‌کنیم. در واقع زمانی که دو مدل احتمالی در اختیار داریم طیف وسیعی از فاصله‌ها قابل تخمین است. دو فاصله‌ی Bhattacharyya و Jeffrey برای این منظور انتخاب شده است، زیرا با بررسی‌های که انجام شد تخمین با خطای کمتری داشته‌اند.

بنابراین نحوه‌ی ارزیابی به این صورت است که بعد از ایجاد یک پیشگو با توزیع احتمال تصادفی مشابه بخش ۱.۳-۳، از توزیع تصادفی به دست آمده نمونه‌گیری می‌شود و داده‌های آموزش به دست می‌آید. سپس با داده‌های به دست آمده، مدل آموزش داده می‌شود و در نهایت با در اختیار داشتن هر دو مدل احتمالی مولد و پیشگو، فاصله‌ی بین این دو به صورت تقریبی محاسبه می‌شود. در ادامه دو فاصله و نحوه‌ی تخمین آن بیان شده است.

فاصله‌ی Bhattacharyya

این فاصله به صورت زیر تعریف می‌شود:

$$\begin{aligned} \mathcal{D}_{BC}(P \parallel Q) &= -\ln BC(P, Q) \\ BC(P, Q) &= \sum_{x \in X} \sqrt{p(x)q(x)}. \end{aligned} \quad (۴.۳)$$

برای تخمین این فاصله، آن را به فرم امید می‌بریم:

$$\begin{aligned}
 BC(P, Q) &= \sum_{x \in X} \sqrt{p(x)q(x)} \\
 &= \sum_{x \in X} \sqrt{p(x)q(x) \frac{p(x)}{p(x)}} \\
 &= \sum_{x \in X} p(x) \sqrt{\frac{q(x)}{p(x)}} \\
 &= \mathbb{E}_{x \sim P} \sqrt{\frac{q(x)}{p(x)}}.
 \end{aligned} \tag{۵.۳}$$

به صورت مشابه، داریم:

$$\begin{aligned}
 BC(P, Q) &= \sum_{x \in X} \sqrt{p(x)q(x)} \\
 &= \sum_{x \in X} \sqrt{p(x)q(x) \frac{q(x)}{q(x)}} \\
 &= \sum_{x \in X} q(x) \sqrt{\frac{p(x)}{q(x)}} \\
 &= \mathbb{E}_{x \sim Q} \sqrt{\frac{p(x)}{q(x)}}.
 \end{aligned} \tag{۶.۳}$$

با توجه به روابط ۵.۳ و ۶.۳ تخمین‌گر زیر به دست می‌آید:

$$\mathfrak{D}_{BC}(P \parallel Q) \approx -\frac{1}{2} \left(\ln \frac{1}{N} \sum_{i=1}^N \sqrt{\frac{q(x^{(i)})}{p(x^{(i)})}} + \ln \frac{1}{M} \sum_{j=1}^M \sqrt{\frac{p(x^{(j)})}{q(x^{(j)})}} \right), \tag{۷.۳}$$

که $x^{(i)}$ ها نمونه‌هایی از توزیع P و $x^{(j)}$ ها نمونه‌هایی از توزیع Q است.

فاصله‌ی Jeffrey

این فاصله به صورت زیر تعریف می‌شود:

$$\mathfrak{D}_J(P \parallel Q) = \sum_{x \in X} (p(x) - q(x)) (\ln p(x) - \ln q(x)). \tag{۸.۳}$$

برای این فاصله تخمین‌گر زیر را داریم:

$$\begin{aligned}
 \mathfrak{D}_J(P \parallel Q) &= \sum_{x \in X} (p(x) - q(x)) (\ln p(x) - \ln q(x)) \\
 &= \sum_{x \in X} p(x) \ln \frac{p(x)}{q(x)} - \sum_{x \in X} q(x) \ln \frac{p(x)}{q(x)} \\
 &= \mathbb{E}_{x \sim P} \ln \frac{p(x)}{q(x)} - \mathbb{E}_{x \sim Q} \ln \frac{p(x)}{q(x)} \\
 &= \mathbb{E}_{x \sim P} \ln \frac{p(x)}{q(x)} + \mathbb{E}_{x \sim Q} \ln \frac{q(x)}{p(x)} \\
 \Rightarrow \mathfrak{D}_J(P \parallel Q) &\approx \frac{1}{N} \sum_{i=1}^N \ln \frac{p(x^{(i)})}{q(x^{(i)})} + \frac{1}{M} \sum_{j=1}^M \ln \frac{q(x^{(j)})}{p(x^{(j)})},
 \end{aligned} \tag{۹.۳}$$

که $x^{(i)}$ ها نمونه‌هایی از توزیع P و $x^{(j)}$ ها نمونه‌هایی از توزیع Q است.

۳.۳-۳ گرامر مستقل از متن

در برخی پژوهش‌ها [۷، ۴۲]، برای بررسی قدرت مدل و اینکه چقدر ساختارهای دنباله را یاد گرفته است از یک گرامر مستقل از متن^۱ استفاده می‌شود. به این صورت که با تعریف گرامری از این نوع، زبانی تعریف می‌شود. با ساخت دنباله‌هایی از این زبان داده‌های آموزش مصنوعی تولید شده و مدل با آن آموزش می‌بیند. از مدل آموزش دیده نمونه‌هایی تولید می‌شود و درصد قبول شدن این نمونه‌ها در گرامر به عنوان معیار در نظر گرفته می‌شود. در این روش می‌توان با تعریف گرامرهای مختلف، سطوح مختلف پیچیدگی دنباله‌ها را ایجاد کرد.

۴-۳ معیارهای مبتنی بر n -گرام

این معیار مناسب دنباله‌هایی مثل زبان‌های طبیعی است که الگوهای محلی غنی دارند. با وجود اینکه این معیارها برای زبان‌های طبیعی پراستفاده است ولی معنای جملات را در نظر نمی‌گیرند و فقط به بررسی n -گرام‌های زبان می‌پردازد. منظور از n -گرام زیر دنباله‌های n تایی یک دنباله است. در این بخش ابتدا دو معیار مورد استفاده در پژوهش‌های پیشین و مشکلات آن‌ها را تشریح کرده، سپس معیار جدیدی که در این پژوهش معرفی شده است، را بیان می‌کنیم. با معیار جدید سعی بر رفع مشکلات دو معیار پیشین داشته‌ایم.

^۱ Context Free Grammar

۱.۴-۳ BLEU

معروف‌ترین معیار بر حسب n -گرام‌ها معیار BLEU است؛ این معیار در اصل در حوزه‌ی ترجمه زبان پیشنهاد شده و صحت این معیار، در ارزیابی ترجمه اثبات شده است. با ارزیابی انسانی مشخص شده که BLEU همبستگی^۱ زیادی با قضاوت انسانی دارد [۴۵]. در معیار BLEU هدف محاسبه‌ی کیفیت ترجمه براساس چند متن مرجع است. در این ارزیابی برای هر جمله در زبان مبدا، چندین جمله در زبان مقصد وجود دارد. پس از ترجمه متن به زبان مقصد، این ترجمه با معیار BLEU بر اساس مجموعه‌ای از جملات که در زبان مقصد در اختیار داریم ارزیابی می‌شود. این ارزیابی بر اساس شباهت n -گرام‌های حاصل ترجمه و جملات نمونه در زبان مقصد است و مقداری بین صفر و یک دارد؛ افزایش آن به معنی شباهت بیشتر است.

این معیار در حوزه‌ی ارزیابی تولید متن‌های تولید شده استفاده زیادی داشته است؛ به این صورت که مجموعه‌ای از داده‌های اصلی (بجای مجموعه متن مرجع در ترجمه) به عنوان مرجع در نظر گرفته می‌شود و متن تولیدی بر اساس این مرجع ارزیابی می‌شود. این معیار بیشتر برای دنباله‌های زبان طبیعی کاربرد دارد، ولی برای دیگر دنباله‌ها نیز قابل استفاده است.

از مشکلات این معیار در بررسی متن تولیدی علاوه بر محدود بودن به بررسی الگوهای محلی، تاثیرپذیری این معیار از اندازه‌ی مجموعه مرجع است [۴۵]. این حساسیت باعث می‌شود در مقایسه‌ی دو روش طبق این معیار، نیاز باشد اندازه‌ی مجموعه مرجع یکسان باشد. مشکل دیگر این ارزیابی عدم توجه به تنوع نمونه‌های تولید شده توسط مدل است. به این صورت که اگر مدل فقط یک نمونه تولید کند و آن یک نمونه، نمونه‌ی با کیفیتی از دید معیار باشد؛ آنگاه امتیاز مدل بسیار بالا می‌رود، در حالی که چنین مدلی مطلوب نیست.

در ادامه به نحوه‌ی محاسبه BLEU که در [۴۵] ارائه شده، برای محاسبه‌ی امتیاز یک تک جمله می‌پردازیم. برای ارزیابی جملات تولید شده، پس از محاسبه‌ی امتیاز BLEU مربوط به تک تک جملات برای تجمیع این امتیازها از میانگین حسابی استفاده می‌شود.^۲

نحوه‌ی محاسبه‌ی BLEU

این معیار بر حسب شباهت‌های n -گرام جمله‌ی مورد بررسی و جملات مرجع کار می‌کند. ابتدا معیار را برای یک n خاص بررسی می‌کنیم و سپس نحوه‌ی امتیازدهی نهایی بیان می‌شود.

این معیار n -گرام‌های جمله‌ی مورد بررسی را با مجموعه‌ی n -گرام‌های موجود در جملات مرجع را به صورت یک به یک جفت می‌کند و نسبتی از n -گرام‌های جمله‌ی مورد بررسی که با n -گرام‌های جملات مرجع یکی می‌شود را امتیاز برای حالت n در نظر می‌گیرد؛ ولی محدودیتی وجود دارد که برای هر n -گرام فقط یک جمله (بهترین جمله) از مجموعه مرجع استفاده می‌شود. یعنی حداکثر تعداد جفت شدن‌های یک n -گرام که با x نشان می‌دهیم، برابر $\max_i \{Count_n(r_i, x)\}$ است که $Count_n(r_i, x)$ به معنی تعداد تکرار n -گرام x در جمله‌ی مرجع i ام است. به صورت دقیق‌تر امتیاز برای یک n خاص برای جمله‌ی مورد بررسی s به صورت زیر محاسبه می‌شود.

$$p_n = \frac{\sum_{x \in G_n} \min \left(Count_n(s, x), \max_i \{Count_n(r_i, x)\} \right)}{\sum_{x \in G_n} Count_n(s, x)}, \quad (10.3)$$

^۱ Correlation

^۲ در مقاله‌ی اصلی BLEU [۴۵] که در حوزه‌ی ترجمه است، راهکاری دیگر برای حالت چند جمله معرفی شده است؛ ولی این راهکار در ارزیابی روش‌های تولید دنباله استفاده نمی‌شود.

که مجموعه‌ی G_n تمام n -گرام‌های ممکن و $Count_n(x, y)$ به معنی تعداد تکرار n -گرام y در جمله‌ی x است. در رابطه ۱۰.۳ مخرج کسر در واقع تعداد n -گرام‌های جمله‌ی مورد بررسی است. بری تجمیع امتیازها به ازای n های مختلف و به دست آوردن یک عدد برای جمله‌ی مورد بررسی، از میانگین هندسی p_n ها استفاده می‌شود؛ زیرا p_n نسبت به n به صورت نمایی کاهش می‌یابد. میانگین هندسی p_n برابر میانگین حسابی $\log p_n$ است.

در معیاری که تا به اینجا گفته شده است، جملات طولانی‌تر از مرجع بدلیل بررسی n -گرام‌های آن در صورت بد بودن جریمه می‌شوند؛ ولی جملات کوچک می‌توانند به نادرست امتیاز بالایی بگیرند. بنابراین با اعمال تاثیر طول جمله‌ی مورد بررسی معیار BLEU محاسبه می‌شود. برای آنکه جملات کوچکتر از مرجع امتیاز نادرست و زیاد نگیرند، تاثیر طول جملات تولید شده به صورت ضربی در معیار اضافه می‌شود. به این صورت که جملات که جمله‌ی هم‌اندازه و یا کوچکتر در مرجع دارند تغییری نمی‌کنند و جملات که از مرجع کوچکتر هستند، جریمه می‌شوند. برای حالتی که n -گرام‌های با n برابر ۱ تا N دارند، لگاریتم معیار BLEU به صورت زیر می‌شود [۴۵]:

$$\min(\cdot, 1 - \frac{l_s}{l_r}) + \frac{1}{N} \sum_{n=1}^N \log P_n, \quad (11.3)$$

که l_s و l_r به ترتیب طول جمله مورد بررسی و طول نزدیکترین جمله^۱ در مرجع به جمله‌ی مورد بررسی است. در حالتی که جمله هم‌طول در مجموعه مرجع وجود دارد، تساوی $\min(\cdot, 1 - \frac{l_s}{l_r}) = 0$ برقرار است و تاثیری در امتیاز محاسبه شده توسط n -گرام‌ها نمی‌گذارد.

۲.۴-۳ Self-BLEU

همانطور که اشاره شده معیار BLEU از تنوع نمونه‌های تولید شده توسط مدل تاثیری نمی‌پذیرد، از این رو در [۴۶] معیاری معرفی شده است که بصورت جدا گانه فقط تنوع نمونه‌های تولید شده را بررسی می‌کند. این معیار بر اساس BLEU است و برای محاسبه‌ی آن، هر یک از جملات تولیدی را به عنوان جمله‌ی مورد بررسی در نظر گرفته و باقی جملات را به عنوان مرجع در نظر می‌گیرد؛ سپس معیار BLEU را در هر حالت محاسبه کرده و میانگین این امتیازها به عنوان Self-BLEU تعریف می‌شود. کم بودن این معیار بیانگر تنوع بیشتر جملات تولیدی است، در بهترین حالت این مقدار صفر و بدترین مقدار آن یک است.

۳.۴-۳ معیار پیشنهادی - MS-Jaccard

همان‌طور که در بخش‌های قبل گفته شد، معیار BLEU فقط کیفیت را در نظر گرفته و تنوع را در نظر نمی‌گیرد؛ از سمت دیگر معیار Self-BLEU کیفیت را در نظر نمی‌گیرد و تنوع را فقط در نظر می‌گیرد. در این بخش معیاری جدید را مبتنی بر n -گرام معرفی می‌کنیم، که به صورت همزمان تنوع و کیفیت را در نظر بگیرد. معیار پیشنهادی MS-Jaccard نام دارد.

همانطور که گفته شد، معیار BLEU تک تک جملات تولید شده را با کل مجموعه‌ی مرجع مقایسه کرده و سپس میانگین امتیازها را استفاده می‌کند. در مقابل روش پیشنهادی کل جملات تولید شده را همزمان با کل جملات

^۱ طبق کد کتابخانه NLTK در حالتی که دو جمله هم فاصله وجود دارد، جمله‌ی با طول کمتر انتخاب می‌شود.

مرجع مقایسه می‌کند تا همزمان کیفیت و تنوع جملات سنجیده شود. برای محاسبه‌ی معیار همه‌ی n -گرام‌های جملات تولید شده را به عنوان یک چندمجموعه^۱ (یعنی یک n -گرام می‌تواند چند بار تکرار شود) و n -گرام‌های جملات مجموعه مرجع را به عنوان چندمجموعه‌ای دیگر در نظر گرفته؛ فاصله‌ی Jaccard این دو چندمجموعه مبنای معیار MS-Jaccard است. به ازای یک n خاص، امتیاز معیار به صورت زیر است:

$$p_n = \frac{\sum_{x \in S_1 \cup S_2} \min\{Count_n(S_1, x), Count_n(S_2, x)\}}{\sum_{x \in S_1 \cup S_2} \max\{Count_n(S_1, x), Count_n(S_2, x)\}}, \quad (12.3)$$

که S_1 و S_2 به ترتیب نشان دهنده‌ی چندمجموعه‌ی شامل n -گرام‌های جملات تولیدی و جملات مرجع است. تابع $Count_n(S, x)$ به معنی تعداد n -گرام x در چندمجموعه‌ی S است. از آنجا که این معیار بسته به اندازه‌ی مجموعه‌ها وابسته است، برای رفع این مشکل تابع $Count_n(S, x)$ را بازتعریف می‌کنیم و آن را به معنی نسبت n -گرام x به کل تعداد در چندمجموعه‌ی S در نظر می‌گیریم. در نهایت معیار MS-Jaccard میانگین هندسی امتیازهای p_n برای n برابر ۱ تا N است. این معیار مقداری بین صفر و یک دارد، افزایش آن به معنی نزدیک شدن جملات تولیدی به جملات مرجع است.

دلیلی که باعث می‌شود معیار MS-Jaccard تنوع دنباله‌ها را نیز در نظر بگیرد این است که دنباله‌های مرجع تنوعی ذاتی در خود دارند و این معیار فاصله‌ای بین نمونه‌های تولید شده و نمونه‌های مرجع محاسبه می‌کند. به همین دلیل تنوعی مشابه با داده‌های مرجع مقادیر بهتری در معیار دارد. برای مثال اگر یک n -گرام در نمونه‌های تولیدی از نمونه‌های مرجع بیشتر شود، آنگاه در صورت کسر ۱۲.۳ مقدار اثرگذار برابر تعداد تکرار n -گرام در نمونه‌های مرجع است و مقدار اثرگذار در مخرج کسر با تعداد n -گرام در نمونه‌های تولید شده زیاد می‌شود و باعث کاهش مقدار معیار می‌شود. به همین صورت کاهش تعداد n -گرام نسبت به نمونه‌های مرجع باعث کاهش صورت کسر می‌شود و صورت ثابت می‌ماند، در نتیجه معیار کم می‌شود.

۵-۳ معیار پیشنهادی - Fréchet BERT Distance

معیار Fréchet BERT Distance (FBD) که در این پژوهش معرفی شده است، مربوط به حوزه‌ی زبان‌های طبیعی است و بیشترین کیفیت را در زبان انگلیسی دارد. در ابتدا معیار Fréchet Inception Distance (FID) [۴۷] و شبکه‌ی عمیق برت^۲ [۴۸] که پایه‌های این معیار است، معرفی می‌شود؛ سپس به تشریح معیار FBD می‌پردازیم. مشکل ارزیابی نتایج مدل‌های مولد، تنها محدود به حوزه‌ی دنباله نیست. روش‌های مولد تصویر نیز در ارزیابی خود چالش دارند و برای رفع آن پژوهش‌های زیادی شده است. معیار پر استفاده و جدیدی که در این حوزه وجود دارد، FID است. معیار FID بر اساس ویژگی‌هایی که از شبکه‌ی عمیق Inception استخراج می‌شود، کار می‌کند. این شبکه یکی از معروف‌ترین دسته‌بندهای حوزه‌ی تصویر است. معیار FID در ابتدا با کمک شبکه‌ی Inception تصاویر مورد ارزیابی را به فضای ویژگی می‌برد، سپس در این فضا فاصله‌ی توزیع نمونه‌های آزمون و نمونه‌های تولید شده را محاسبه کرده و به عنوان معیار در نظر می‌گیرد. برای محاسبه‌ی فاصله توزیع نمونه‌های آزمون و نمونه‌های تولید شده، برای هر کدام یک توزیع گاوسی فرض می‌شود و فاصله‌ای که بین این دو توزیع محاسبه می‌شود، براساس فاصله‌ی Fréchet است. فاصله‌ی Fréchet که به نام Wasserstein-2 هم شناخته می‌شود، در حالت کلی قابل محاسبه

^۱ Multiset

^۲ BERT

به فرم بسته نیست؛ ولی در اینجا چون با تقریب دو توزیع گاوسی فرض شده است، فاصله به صورت زیر قابل محاسبه است:

$$\|m_1 - m_2\|_2^2 + \text{Tr}(C_1 + C_2 - 2(C_1 C_2)^{1/2}), \quad (13.3)$$

که C_i و m_i به ترتیب میانگین و کواریانس توزیع‌ها را نشان می‌دهند. تا چندی پیش در حوزه‌ی زبان طبیعی، شبکه‌ای به قدرت Inception وجود نداشت. ولی به تازگی شبکه‌ای توسط شرکت گوگل بر روی کل متون ویکی‌پدیا آموزش دیده و منتشر شده است. این شبکه که برت نام دارد، یک شبکه‌ی عمیق و پر قدرت است که آموزش آن نیاز به چندین ماه محاسبه توسط کارت گرافیک دارد. شبکه‌ی برت توانسته در بسیاری چالش‌های حوزه‌ی متن زبان طبیعی نتایج مرز دانش را کسب کند [۴۸]. از آنجا که شبکه‌های آموزش دیده‌ی برت بر روی بیشتر زبان‌ها به صورت عمومی منتشر شده است^۱، می‌توان از آن به جای شبکه‌ی Inception در معیار FID استفاده کرد و معیاری در حوزه‌ی زبان‌های طبیعی داشت. بنابراین معیار FBD را به این صورت تعریف می‌کنیم که ابتدا جملات تولید شده و جملات آزمون با استفاده از شبکه‌ی برت به فضای ویژگی برده می‌شوند؛ سپس با محاسبه‌ی میانگین و کواریانس هر کدام از این دو مجموعه داده، فاصله‌ی Fréchet بین آنها محاسبه می‌شود.

۳-۶ جمع‌بندی

در این بخش معیارهای پیشین و معیارهای پیشنهادی در قالب جدول ۳-۱ جمع‌بندی شده است. لازم به ذکر است که معیارهای مشخص شده با ستاره، برای ارزیابی مدل مولد دنباله، در این پژوهش معرفی شده‌اند.

به این ترتیب، در این پژوهش هم برای ارزیابی مبتنی بر داده‌ی مصنوعی، هم برای ارزیابی مبتنی بر n -گرام و هم برای ارزیابی زبان طبیعی معیارهایی ارائه کردیم که به طور هم‌زمان تنوع و کیفیت نمونه‌ها را در نظر می‌گیرند.

^۱ <https://github.com/google-research/bert>

جدول ۳-۱: مقیاس‌های معیارهای موجود و پیشنهاد شده. «مدل احتمالی» بیانگر نیاز به احتمالی بودن مدل است. «تاثیر تنوع» به معنی در نظر گرفتن تنوع دنباله‌های تولید شده از نظر معیار است. «تاثیر کیفیت» به معنی در نظر گرفتن کیفیت دنباله‌های تولید شده از دید معیار است. «تاثیر آریبی مواجهه» به معنی تاثیر پذیری معیار از وجود این مشکل است. «سمت بهبود» به معنی جهتی است که در آن جمت تغییر معیار مطلوب است. «بازه» نشان دهنده‌ی بازه‌ی ممکن برای معیار است. H به معنی آنتروپی^۱ داده‌های آموزش است. معیارهای مشخص شده با ستاره، برای ارزیابی مدل مولد دنباله، در این پژوهش معرفی شده‌اند.

دسته	نام مختصر	نام کامل	بخش مرتبط	مدل احتمالی	تاثیر تنوع	تاثیر کیفیت	تاثیر آریبی مواجهه	جهت بهبود	بازه
مبتنی بر احتمال مدل	NLL-Model	منفی لگاریتم درست‌نمایی مدل	۱.۲-۳	*	*	*		↓	H تا ∞
	PP	Perplexity	۲.۲-۳	*	*	*		↓	b^H تا ∞
مبتنی بر داده‌ی مصنوعی	NLL-Oracle	منفی لگاریتم درست‌نمایی پیشگو	۱.۳-۳			*	*	↓	H تا ∞
	*Bhattacharyya	Bhattacharyya	۲.۳-۳	*	*	*	*	↓	\cdot تا ∞
	*Jeffrey	Jeffrey	۲.۳-۳	*	*	*	*	↓	\cdot تا ∞
	CFG	گرامر مستقل از متن	۳.۳-۳			*	*	↑	\cdot تا ۱
	BL	BLEU	۱.۴-۳			*	*	↑	\cdot تا ۱
مبتنی بر n -گرام	SBL	Self-BLEU	۲.۴-۳		*			↓	\cdot تا ۱
	*MSJ	MS-Jaccard	۳.۴-۳		*	*	*	↑	\cdot تا ۱
زبان طبیعی	*FBD	Fréchet BERT Distance	۵-۳		*	*	*	↓	\cdot تا ∞

مراجع

- [1] D. Weininger, "Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules," *Journal of chemical information and computer sciences*, vol. 28, no. 1, pp. 31–36, 1988.
- [2] G. L. Guimaraes, B. Sanchez-Lengeling, P. L. C. Farias, and A. Aspuru-Guzik, "Objective-reinforced generative adversarial networks (ORGAN) for sequence generation models," *CoRR*, vol. abs/1705.10843, 2017. [Online]. Available: <http://arxiv.org/abs/1705.10843>
- [3] A. Bojchevski, O. Shchur, D. Zügner, and S. Günnemann, "Netgan: Generating graphs via random walks," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018, pp. 609–618. [Online]. Available: <http://proceedings.mlr.press/v80/bojchevski18a.html>
- [4] J. You, R. Ying, X. Ren, W. L. Hamilton, and J. Leskovec, "Graphrnn: Generating realistic graphs with deep auto-regressive models," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, 2018, pp. 5694–5703. [Online]. Available: <http://proceedings.mlr.press/v80/you18a.html>
- [5] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, 2017, pp. 2852–2858. [Online]. Available: <http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14344>
- [6] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, 2014, pp. 2672–2680. [Online]. Available: <http://papers.nips.cc/paper/5423-generative-adversarial-nets>
- [7] M. J. Kusner and J. M. Hernández-Lobato, "GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution," *arXiv e-prints*, p. arXiv:1611.04051, Nov. 2016. [Online]. Available: <https://arxiv.org/pdf/1611.04051.pdf>

- [8] A. M. Lamb, A. G. ALIAS PARTH GOYAL, Y. Zhang, S. Zhang, A. C. Courville, and Y. Bengio, "Professor forcing: A new algorithm for training recurrent networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 4601–4609. [Online]. Available: <http://papers.nips.cc/paper/6099-professor-forcing-a-new-algorithm-for-training-recurrent-networks.pdf>
- [9] Y. Zhang, Z. Gan, K. Fan, Z. Chen, R. Henao, D. Shen, and L. Carin, "Adversarial feature matching for text generation," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 4006–4015. [Online]. Available: <http://proceedings.mlr.press/v70/zhang17b.html>
- [10] K. Lin, D. Li, X. He, Z. Zhang, and M.-T. Sun, "Adversarial ranking for language generation," in *Advances in Neural Information Processing Systems*, 2017, pp. 3155–3165.
- [11] J. Guo, S. Lu, H. Cai, W. Zhang, Y. Yu, and J. Wang, "Long text generation via adversarial training with leaked information," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 2018, pp. 5141–5148. [Online]. Available: <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16360>
- [12] T. Che, Y. Li, R. Zhang, R. D. Hjelm, W. Li, Y. Song, and Y. Bengio, "Maximum-likelihood augmented discrete generative adversarial networks," *arXiv preprint arXiv:1702.07983*, 2017. [Online]. Available: <https://arxiv.org/pdf/1702.07983.pdf>
- [13] O. Press, A. Bar, B. Bogin, J. Berant, and L. Wolf, "Language generation with recurrent generative adversarial networks without pre-training," *arXiv preprint arXiv:1706.01399*, 2017.
- [14] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," *arXiv preprint arXiv:1812.04948*, 2018.
- [15] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," in *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016, pp. 271–279. [Online]. Available: <http://papers.nips.cc/paper/6066-f-gan-training-generative-neural-samplers-using-variational-divergence-minimization>
- [16] B. Poole, A. A. Alemi, J. Sohl-Dickstein, and A. Angelova, "Improved generator objectives for gans," *arXiv preprint arXiv:1612.02780*, 2016. [Online]. Available: <https://arxiv.org/pdf/1612.02780.pdf>

- [17] L. Theis, A. van den Oord, and M. Bethge, “A note on the evaluation of generative models,” *CoRR*, vol. abs/1511.01844, 2015. [Online]. Available: <http://arxiv.org/abs/1511.01844>
- [18] F. Huszar, “How (not) to train your generative model: Scheduled sampling, likelihood, adversary?” *CoRR*, vol. abs/1511.05101, 2015. [Online]. Available: <http://arxiv.org/abs/1511.05101>
- [19] E. Jang, S. Gu, and B. Poole, “Categorical reparametrization with gumbel-softmax,” in *International Conference on Learning Representations (ICLR)*, 2017. [Online]. Available: <https://openreview.net/pdf?id=rkE3y85ee>
- [20] L. Mescheder, S. Nowozin, and A. Geiger, “The numerics of gans,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 1825–1835. [Online]. Available: <http://papers.nips.cc/paper/6779-the-numerics-of-gans.pdf>
- [21] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein GAN,” *CoRR*, vol. abs/1701.07875, 2017. [Online]. Available: <http://arxiv.org/abs/1701.07875>
- [22] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5767–5777. [Online]. Available: <http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf>
- [23] K. Roth, A. Lucchi, S. Nowozin, and T. Hofmann, “Stabilizing training of generative adversarial networks through regularization,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 2018–2028. [Online]. Available: <http://papers.nips.cc/paper/6797-stabilizing-training-of-generative-adversarial-networks-through-regularization.pdf>
- [24] L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein, “Unrolled generative adversarial networks,” *CoRR*, vol. abs/1611.02163, 2016. [Online]. Available: <http://arxiv.org/abs/1611.02163>
- [25] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, “Improved techniques for training gans,” in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2234–2242. [Online]. Available: <http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf>
- [26] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, and S. Bengio, “Generating sentences from a continuous space,” in *Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016*, 2016, pp. 10–21. [Online]. Available: <http://aclweb.org/anthology/K/K16/K16-1002.pdf>

- [27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [28] Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen, "Breaking the softmax bottleneck: A high-rank RNN language model," in *International Conference on Learning Representations*, 2018. [Online]. Available: <https://openreview.net/forum?id=HkwZSG-CZ>
- [29] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 1171–1179. [Online]. Available: <https://papers.nips.cc/paper/5956-scheduled-sampling-for-sequence-prediction-with-recurrent-neural-networks.pdf>
- [30] C. Doersch, "Tutorial on variational autoencoders," *CoRR*, vol. abs/1606.05908, 2016. [Online]. Available: <http://arxiv.org/abs/1606.05908>
- [31] Z. Yang, Z. Hu, R. Salakhutdinov, and T. Berg-Kirkpatrick, "Improved variational autoencoders for text modeling using dilated convolutions," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 3881–3890. [Online]. Available: <http://proceedings.mlr.press/v70/yang17d.html>
- [32] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations (ICLR)*, 2016. [Online]. Available: <https://arxiv.org/pdf/1511.07122.pdf>
- [33] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 2014, pp. 1746–1751. [Online]. Available: <http://aclweb.org/anthology/D/D14/D14-1181.pdf>
- [34] I. J. Goodfellow, "NIPS 2016 tutorial: Generative adversarial networks," *CoRR*, vol. abs/1701.00160, 2017. [Online]. Available: <http://arxiv.org/abs/1701.00160>
- [35] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," in *International Conference on Learning Representations (ICLR)*, 2016. [Online]. Available: <https://arxiv.org/pdf/1511.06732.pdf>
- [36] C. K. Sønderby, J. Caballero, L. Theis, W. Shi, and F. Huszár, "Amortised map inference for image super-resolution," in *International Conference on Learning Representations (ICLR)*, 2017. [Online]. Available: <https://openreview.net/pdf?id=S1RP6GLle>
- [37] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems 29*, D. D. Lee,

- M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 2234–2242. [Online]. Available: <http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf>
- [38] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, “Policy gradient methods for reinforcement learning with function approximation,” in *Advances in Neural Information Processing Systems 12, [NIPS Conference, Denver, Colorado, USA, November 29 - December 4, 1999]*, 1999, pp. 1057–1063. [Online]. Available: <http://papers.nips.cc/paper/1713-policy-gradient-methods-for-reinforcement-learning-with-function-approximation>
- [39] R. K. Srivastava, K. Greff, and J. Schmidhuber, “Highway networks,” *CoRR*, vol. abs/1505.00387, 2015. [Online]. Available: <https://arxiv.org/pdf/1505.00387.pdf>
- [40] X. Zhang and Y. LeCun, “Text understanding from scratch,” *CoRR*, vol. abs/1502.01710, 2015. [Online]. Available: <http://arxiv.org/abs/1502.01710>
- [41] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu, “Feudal networks for hierarchical reinforcement learning,” in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, 2017, pp. 3540–3549. [Online]. Available: <http://proceedings.mlr.press/v70/vezhnevets17a.html>
- [42] S. Subramanian, S. Rajeswar, F. Dutil, C. Pal, and A. Courville, “Adversarial generation of natural language,” in *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 2017, pp. 241–251.
- [43] S. Semeniuta, A. Severyn, and S. Gelly, “On accurate evaluation of gans for language generation,” *CoRR*, vol. abs/1806.04936, 2018. [Online]. Available: <http://arxiv.org/abs/1806.04936>
- [44] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder-decoder approaches,” in *Proceedings of SSST@EMNLP 2014, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, Doha, Qatar, 25 October 2014*, 2014, pp. 103–111. [Online]. Available: <http://aclweb.org/anthology/W/W14/W14-4012.pdf>
- [45] K. Papineni, S. Roukos, T. Ward, and W. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA.*, 2002, pp. 311–318. [Online]. Available: <http://www.aclweb.org/anthology/P02-1040.pdf>
- [46] Y. Zhu, S. Lu, L. Zheng, J. Guo, W. Zhang, J. Wang, and Y. Yu, “Texygen: A benchmarking platform for text generation models,” *SIGIR*, 2018. [Online]. Available: <https://arxiv.org/pdf/1802.01886.pdf>
- [47] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “Gans trained by a two time-scale update rule converge to a local nash equilibrium,” in *Advances in Neural Infor-*

- mation Processing Systems*, 2017, pp. 6626–6637. [Online]. Available: <https://papers.nips.cc/paper/7240-gans-trained-by-a-two-time-scale-update-rule-converge-to-a-local-nash-equilibrium.pdf>
- [48] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018. [Online]. Available: <https://arxiv.org/abs/1810.04805>

واژه‌نامه فارسی به انگلیسی

Beam Search	جستجوی پرتویی	n-gram	گرام- n
Black Box	جعبه سیاه	Test	آزمون
Variational Autoencoder	خود رمزگذار وردشی	Train	آموزش
Autoencoder	خود رمزگذار	Entropy	آنترپی
Variational Autoencoder	خود رمزگذار وردشی	Cross Entropy	آنترپی متقاطع
Data As Demonstrator (DAD)	داده به عنوان اثبات‌گر	Dilation	اتساع
Categorical	دسته‌ای	Saturate	اشباع
Classifier	دسته‌بند	Backpropagation	انتشار رو به عقب
Pooling	رای‌گیری	Receptive Field Size	اندازه‌ی میدان دریافتی
Ranker	رتبه‌بند	Vocabulary Size	اندازه‌ی واژگان
Encoder	رمزگذار	Bias	اُریبی
Decoder	رمزگشا	Exposure Bias	اُریبی مواجهه
Copy	رونوشت	BERT	برت
Sub-goal	زیرهدف	Maximum Likelihood	بیشینه درست‌نمایی
Style	سبک	Maximum Mean Discrepancy	بیشینه میانگین اختلاف
Hierarchical	سلسله‌مراتبی	Softmax	بیشینه‌ی هموار
Policy	سیاست	Overfit	بیش‌برازش
Highway Networks	شبکه‌های شاهراهی	Objective Function	تابع هدف
Feudal Net	شبکه‌های فئودالی	Cost Function	تابع هزینه
Generative Adversarial Networks	شبکه‌های مولد مقابله‌ای	Reconstruction Loss	تابع هزینه‌ی بازسازی
Recurrent Neural Network	شبکه‌ی عصبی بازگردنده	Fully Connected	تمام‌متصل
Convolutional Neural Networks	شبکه‌ی عصبی پیچشی	Discriminator	تمیزدهنده
Gradient Ascent	صعود در راستای گرادیان	Reparametrization	تنظیم مجدد
Agent	عامل	Professor Forcing	جبر استاد
Causality	علّی	Teacher Forcing	جبر معلم
Action	عمل	Penalty	جریمه
		Penalty	جریمه‌کننده
		Monte Carlo Search	جستجوی مونت کارلو

Reward	پاداش	Hyperparameter	فوق پارامتر
Delayed Reward	پاداش با تاخیر	Chain Rule	قاعده‌ی زنجیره‌ای
Posterior	پسین	Manager	مدیر
Pre-train	پیش-آموزش	State-of-the-Art	مرز دانش
Oracle	پیشگو	Tradeoff	مصالحه
Prior	پیشین	Adversarial	مقابله‌ای
Convolution	پیچش	Moments	ممان‌های
Convolutional	پیچشی	Negative Log Likelihood	منفی لگاریتم درست‌نمایی
Dilated Convolution	پیچشی متسع شده	Generative	مولد
One-hot	یک-فعال	Mean-Seeking	میانگین-جستجوگری
Mode Collapsing	چسبیدگی به قله	Inconsistency	ناهماهنگی
Multiset	چندمجموعه	Scheduled Sampling	نمونه‌برداری زمان‌بندی شده
Worker	کارگر	Importance Sampling	نمونه‌گیری براساس اهمیت
Policy Gradient	گرادیان سیاست	Latent	نهان
Knowledge Graph	گراف دانش	Gaussian kernel	هسته‌ی گاوسی
Context Free Grammar	گرامر مستقل از متن	Correlation	همبستگی
Random Walk	گشت‌زنی تصادفی	State	وضعیت
		Hidden State	وضعیت مخفی

واژه‌نامه انگلیسی به فارسی

Encoder	رمزگذار	Action	عمل
Entropy	آنتروپی	Adversarial	مقابله‌ای
Exposure Bias	اُریبی مواجهه	Agent	عامل
Feudal Net	شبکه‌های فئودالی	Autoencoder	خودرمزگذار
Fully Connected	تمام‌متصل	Backpropagation	انتشار رو به عقب
Gaussian kernel	هسته‌ی گاوسی	Beam Search	جستجوی پرتویی
Generative	مولد	BERT	برت
Generative Adversarial Networks	شبکه‌های مولد	Bias	اُریبی
	مقابله‌ای	Black Box	جعبه‌سیاه
Gradient Ascent	صعود در راستای گرادیان	Categorical	دسته‌ای
Hidden State	وضعیت مخفی	Causality	علّی
Hierarchical	سلسله‌مراتبی	Chain Rule	قاعده‌ی زنجیره‌ای
Highway Networks	شبکه‌های شاهراهی	Classifier	دسته‌بند
Hyperparameter	فوق پارامتر	Context Free Grammar	گرامر مستقل از متن
Importance Sampling	نمونه‌گیری براساس اهمیت	Convolution	پیچش
Inconsistency	ناهماهنگی	Convolutional	پیچشی
Knowledge Graph	گراف دانش	Convolutional Neural Networks	شبکه‌ی عصبی
Latent	نهان		پیچشی
Manager	مدیر	Copy	رونوشت
Maximum Likelihood	بیشینه درست‌نمایی	Correlation	همبستگی
Maximum Mean Discrepancy	بیشینه میانگین اختلاف	Cost Function	تابع هزینه
Mean-Seeking	میانگین-جستجوگری	Cross Entropy	آنتروپی متقاطع
Mode Collapsing	چسبیدگی به قله	Data As Demonstrator (DAD)	داده به عنوان اثبات‌گر
Moments	ممان‌های	Decoder	رمزگشا
Monte Carlo Search	جستجوی مونت کارلو	Delayed Reward	پاداش با تاخیر
Multiset	چندمجموعه	Dilated Convolution	پیچشی متسع شده
Negative Log Likelihood	منفی لگاریتم درست‌نمایی	Dilation	اتساع
n-gram	n-گرام	Discriminator	تمیزدهنده

Reparametrization	تنظیم مجدد	Objective Function	تابع هدف
Reward	پاداش	One-hot	یک-فعال
Saturate	اشباع	Oracle	پیشگو
Scheduled Sampling	نمونه‌برداری زمان‌بندی شده	Overfit	بیش‌پرازش
Softmax	بیشینه‌ی هموار	Penalty	جریمه‌کننده
State	وضعیت	Policy	سیاست
State-of-the-Art	مرز دانش	Policy Gradient	گرادیان سیاست
Style	سبک	Pooling	رای‌گیری
Sub-goal	زیرهدف	Posterior	پسین
Teacher Forcing	جبر معلم	Pre-train	پیش-آموزش
Test	آزمون	Prior	پیشین
Tradeoff	مصالحه	Professor Forcing	جبر استاد
Train	آموزش	Random Walk	گشت‌زنی تصادفی
Variational Autoencoder	خود رمزگذار وردشی	Ranker	رتبه‌بند
Vocabulary Size	اندازه‌ی واژگان	Receptive Field Size	اندازه‌ی میدان دریافتی
Worker	کارگر	Reconstruction Loss	تابع هزینه‌ی بازسازی
		Recurrent Neural Network	شبکه‌ی عصبی بازگردنده

Abstract

Lots of essential structures can be modeled as sequences and sequences can be utilized to model the structures like molecules, graphs and music notes. On the other hand, generating meaningful and new sequences is an important and practical problem in different applications. Natural language translation and drug discovery are examples of sequence generation problem. However, there are substantial challenges in sequence generation problem. Discrete spaces of the sequence and challenge of the proper objective function can be pointed out.

On the other, the baseline methods suffer from issues like exposure bias between training and test time, and the ill-defined objective function. So, the necessity of new methods is available.

In recent years, there has been development in image generation by usage of generative adversarial networks (GANs). These networks have attracted attention for sequence generation lately, but since sequences are discrete, this cannot be done easily, and new approaches like reinforcement learning and approximation should be utilized. Furthermore, the problem of instability in generative adversarial networks causes new challenges.

In this research, based on the idea of generative adversarial models, a new iterative method is proposed for sequence generation problem, such that in every step of the algorithm, the model is trained against itself by using the generated samples. The idea of the proposed method is based on the ratio estimation which enables the model to overcome the problem of discreteness in data. Also, the proposed method is more stable than the other GAN-based methods. It also should be noted that the exposure bias problem does not exist in the proposed method.

Since the evaluation of the generated sequences is also an essential challenge in the field of sequence generation, we reviewed the evaluation criterion for sequence generation and also proposed three new methods for evaluating the new sequences which in contrast to previous criterions, examines both the quality and diversity of the new samples.

Experiments show the superiority of the proposed method to previous methods.

Keywords: *Sequence Generation, Adversarial Networks, Neural Network, Deep Learning*



Sharif University of Technology
Department of Computer Engineering

M.Sc. Thesis
Artificial Intelligence

Topic
Adversarial Networks for Sequence Generation

By
Ehsan Montahaei

Supervisor
Dr. M. Soleymani

Winter 2018

