



دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

پایان‌نامه‌ی کارشناسی
رشته مهندسی کامپیوتر

عنوان:

تولید شرطی گراف

نگارش:

امیر مجتبی صبور

استاد راهنما:

دکتر مهدیه سلیمانی

بهمن ۱۳۹۹

سلام افلا

سپاس

از دکتر سلیمانی که به عنوان استاد راهنما با کمک‌ها و راهنمایی‌های بی‌دریغشان مرا در انجام این پروژه یاری کردند و از خانم فائزه فائز که مرا در تمام طول انجام پروژه راهنمایی کردند؛ تشکر و قدردانی می‌کنم.

چکیده

شبکه‌های مولد عمیق در سال‌های اخیر موفقیت چشمگیری در زمینه تولید گراف بدست آورده‌اند. اما تاکنون تمرکز محققان بر روی تولید بدون شرط گراف بوده است، در حالی که در زمینه های تصاویر و متون پیشرفت‌های بسیار زیادی در تولید مشروط آنها دیده می‌شود. از آنجا که اطلاعات گراف‌ها بسیار وابسته به محیط تعریف‌شان است، تولید شرطی گراف اهمیت زیادی پیدا می‌کند و سوال مطرح شده این است که آیا میتوان پیشرفت‌های دو حوزه متن و عکس را به حوزه گراف منتقل کرد. در این پروژه، تعدادی از راهکارهای مورد استفاده در حوزه‌ی متن و عکس را استفاده می‌کنیم و روشی نظارتی برای یادگیری شبکه‌های مولد عمیق در راستای تولید شرطی گراف ارائه می‌دهیم. سپس، روش مطرح‌شده را روی چندین مجموعه‌داده مختلف آزمایش کرده و نتایج آن را گزارش می‌کنیم.

کلیدواژه‌ها: یادگیری عمیق، تولید گراف، تولید شرطی گراف

فهرست مطالب

۹	۱ مقدمه
۹	۱-۱ تعریف مسئله
۱۰	۲-۱ اهمیت موضوع
۱۰	۳-۱ اهداف تحقیق
۱۱	۴-۱ ساختار پایان نامه
۱۲	۲ مفاهیم اولیه
۱۲	۱-۲ یادگیری نظارتی
۱۲	۲-۲ مدل های مولد و مولد مشروط
۱۳	۳-۲ دادگان گراف
۱۳	۱-۳-۲ تعریف
۱۴	۲-۳-۲ چالش ها
۱۴	۴-۲ آموزش مدل های مولد
۱۶	۵-۲ ارزیابی مدل های مولد
۱۸	۳ کارهای پیشین
۱۸	۱-۳ تولید مشروط در حوزه متن

۱۹	۱-۱-۳ روش‌های مبتنی بر طراحی کدگذار و کدگشا
۲۱	۱-۲-۳ روش‌های مبتنی بر مدل‌های مولد خصمانه
۲۳	۲-۳ تولید در حوزه گراف
۲۷	۴ مدل و نتایج
۲۷	۱-۴ بررسی اجمالی مدل
۲۹	۲-۴ مشکل گسسته بودن و عدم جریان گرادیان
۳۰	۳-۴ آزمایش‌ها و نتایج
۳۰	۱-۳-۴ مجموعه داده‌ها
۳۱	۲-۳-۴ بررسی نتایج آزمایش
۳۴	۵ نتیجه‌گیری

فهرست شکل‌ها

۱۶	۱-۲ تاثیر چینش رئوس گراف در تصویر آن‌ها
۲۰	۱-۳ معماری مدل مولد مشروط متن بر اساس مدل‌های خودکدگذار متغیر
۲۲	۲-۳ معماری مدل‌های مولد خصمانه
۲۳	۳-۳ معماری مدل SentiGAN
۲۵	۴-۳ معماری مدل GraphRNN
۳۱	۱-۴ توزیع تعداد رئوس مجموعه داده PROTEINS
۳۲	۲-۴ توزیع تخمین مقادیر p در گراف‌های تولید شده در آزمایش مجموعه داده Erdos Renyi

فهرست جدول‌ها

۳۰	۴-۱ خلاصه اطلاعات مجموعه داده‌های PROTEINS و مجموعه داده ساختگی-ErdosRenyi-0.25-0.75	50
۳۲	۴-۲ نتایج مدل روی مجموعه داده Erdos Renyi	
۳۳	۴-۳ بررسی صحت گراف‌های تولید شده در آزمایش مجموعه داده PROTEINS	
۳۳	۴-۴ بررسی صحت گراف‌های تولید شده در آزمایش مجموعه داده PROTEINS	

فصل ۱

مقدمه

۱-۱ تعریف مسئله

مدل‌های مولد عمیق^۱ یکی از زمینه‌های بسیار فعال تحقیقاتی در زمینه یادگیری ماشین است. گرچه این مدل‌ها پیشرفت‌های چشم‌گیری در حوزه‌های تصویر و متن، همچون تولید تصاویر و متون و یادگیری نمایش‌های قابل تفسیر^۲، بدست آورده‌اند، اما بکارگیری آن‌ها در حوزه گراف کمتر مورد مطالعه قرار گرفته شده است. در حال حاضر، اکثر دستاوردهای این حوزه در راستای تولید بدون شرط گراف می‌باشد و گام بعد این است که چگونه می‌توان این روش‌ها را تعمیم داد تا بتوان مقدار کنترل انسانی را در آن‌ها افزایش داد.

در حالت عادی هدف آموزش دادن مدل‌های مولد عمیق آن است که بتوان به واسطه آن‌ها از توزیع یادگرفته شده نمونه برداری کرد. اما اگر بخواهیم یک قدم جلوتر برویم، به مدل‌های مولد شرطی می‌رسیم. در این مدل‌ها، هر داده تعدادی شرط دارد و هدف، یادگیری توزیع دادگان مشروط به این شرط‌ها است. به عنوان مثال، در زمینه تولید تصاویر، مدل‌هایی وجود دارند که تصاویر متنوع‌ای از انسان‌ها تولید می‌کنند و همچنین مدل‌هایی وجود دارند که با ورودی گرفتن تعدادی شرط، همچون جنسیت و رنگ پوست، تصویر انسان‌هایی که دارای این شرط‌ها هستند را تولید می‌کند. در این پروژه هدف ما ارائه مدل مولدی برای مسئله تولید مشروط گراف است. طرح دقیق حالت کلی مسئله به شکل زیر است:

مسئله ۱-۱ فرض کنید $D_{train} = \{(g_i, c_i) | i = 1, 2, \dots, n\}$ دادگان آموزش باشند که هر داده یک گراف (g_i) و یک شرط (c_i) است. هدف پیدا کردن شبکه عصبی F_θ با پارامترهای θ است که با ورودی گرفتن شرط c ، در خروجی

^۱ Deep Generative Models
^۲ Interpretable Representations

گراف y را به گونه‌ای تولید کند که y از توزیع $P(g|c)$ بیاید.

۲-۱ اهمیت موضوع

از آنجا که تولید مشروط گراف‌ها بسیار انتزاعی است، می‌توان از آن در هر حوزه‌ای که داده‌ها نمایش گراف داشته باشند، استفاده کرد. به عنوان مثال، در زمینه داروسازی و مولکول‌سازی داده‌ها به صورت گراف هستند و می‌توان در این حوزه‌ها با آموزش دادن یک مدل مولد، مولکول‌های جدیدی با ویژگی‌های فیزیکی مشخصی تولید کرد. مثلاً می‌توان مدلی آموزش داد که بتواند مولکول‌هایی تولید کند که با احتمال بالا دارای بوی خوشی باشند و چنین مدلی در صنعت عطرسازی خیلی کمک‌کننده خواهد بود. همچنین، می‌توان مدلی تولید کرد که مولکول‌هایی که عایق بسیار خوبی در برابر سرما و گرما هستند پیشنهاد بدهد، و این مولکول‌ها در زمینه‌هایی مانند پنجره‌سازی می‌توانند خیلی موثر باشند.

زمینه‌ی دیگری که در آن می‌توان داده‌ها را به صورت گراف نمایش داد، زمینه تولید محتوا^۳ است. به عنوان مثال، می‌توان چیدمان جاده‌ها در یک شهر را به صورت گراف نشان داد که در آن رئوس گراف چهارراه‌ها و تقاطع‌ها هستند و یال‌ها جاده‌ها هستند. در نتیجه، می‌توان مدلی را تصور کرد که با ورودی گرفتن ویژگی‌هایی درباره نحوه چیدمان شهر، شبکه جاده‌ای جدیدی تولید کند که آن ویژگی‌ها را داراست. همچنین، می‌توان به هر درخت به صورت یک گراف نگاه کرد که رئوس آن مکان‌هایی هستند که شاخه‌ای در آن‌ها جدا می‌شود و مدلی را تصور کرد که با ورودی گرفتن نوع درخت، مدل سه‌بعدی از آن نوع درخت تولید می‌کند. از تولید محتوا می‌توان در صنعت‌هایی همچون تولید بازی، تولید فیلم، شبیه‌سازی، معماری، و رباتیک به صورت مستقیم استفاده کرد.

۳-۱ اهداف تحقیق

در این پروژه هدف بررسی ایده‌های موجود در زمینه تولید مشروط داده در حوزه‌های تصویر و متن است. سپس از آن‌ها استفاده می‌شود تا یک مدل مولد عمیق برای مسئله تولید مشروط گراف ساخته شود، به طوری که گراف‌های تولید شده توسط مدل ارائه شده، شباهت زیادی به گراف‌های واقعی مجموعه داده‌ها داشته باشند.

۴-۱ ساختار پایان نامه

این پایان نامه شامل ۵ فصل است. فصل دوم درباره‌ی تعاریف اولیه‌ای که در ادامه به آن‌ها خواهیم خورد است و به توضیح آن‌ها می‌پردازد. در فصل سوم، تعدادی از کارهای پیشین مربوط به تولید و تولید مشروط در حوزه متون و گراف‌ها را بررسی می‌کنیم. در فصل چهارم مدل ارائه شده این پایان نامه به تفصیل بیان می‌شود و نتایج آزمایش‌های این مدل روی دو مجموعه داده واقعی و تولید شده ارائه و تجزیه و تحلیل می‌شود. در انتها، در فصل پنجم، به نتیجه‌گیری و بیان پیشنهادهایی برای کارهای آتی می‌پردازیم.

فصل ۲

مفاهیم اولیه

۱-۲ یادگیری نظارتی

یادگیری نظارتی^۱ در یادگیری ماشین به آموزش مدل‌هایی گفته می‌شود که برای آموزش دادن آن‌ها از دادگان برچسب‌دار استفاده می‌شود. در این روش هر داده تشکیل شده است از یک ورودی (معمولاً یک بردار است) و یک خروجی موردانتظار برای آن ورودی. یادگیری نظارتی ساده‌ترین و رایج‌ترین نوع آموزش در یادگیری ماشین است که مزیت اصلی آن، بیشتر بودن دقت مدل‌های حاصل است. اما لازمه‌ی این روش، دادگان برچسب خورده است که ممکن است باعث دشواری جمع‌آوری آن‌ها بشود.

۲-۲ مدل‌های مولد و مولد مشروط

یکی از مهم‌ترین وظایف هوش مصنوعی، تولید کردن الگوریتم‌هایی است که به رایانه‌ها امکان درک دنیا را می‌دهند و مدل‌های مولد^۲ یکی از امیدوارکننده‌ترین رویکردها برای رسیدن به این هدف هستند. در این مدل‌ها، هدف یادگیری توزیع دادگان است و به طور معمول، این مدل‌ها متغیرهای نهفته‌ای^۳ دارند که با توجه به داده‌های مشاهده‌شده استنباط^۴ می‌شوند. پس از یادگیری، از مدل‌های مولد هم برای تجزیه و تحلیل داده‌ها با استفاده از این متغیرهای نهفته می‌توان استفاده کرد و هم با استفاده از آن‌ها می‌توان از توزیع داده‌ها نمونه برداری انجام داد.

Supervised Learning^۱
Generative Models^۲
Latent Variables^۳
Infer^۴

در مدل‌های مولد مشروط اوضاع کمی متفاوت است و هر داده، علاوه بر خود نوع داده‌ای که به دنبال تولید آن هستیم، دارای یک شرط نیز است و هدف مدل، یادگیری توزیع دادگان مشروط به شرط ورودی است. مزیت این مدل‌ها نسبت به مدل‌های بدون شرط این است که در این مدل‌ها کنترل بیشتری روی خروجی مدل وجود دارد و می‌توان با تغییر دادن شرط ورودی داده شده، خروجی مدل را تنظیم کرد.

۲-۳ دادگان گراف

۲-۳-۱ تعریف

برای استفاده از مجموعه داده‌های گراف، باید با نحوه نمایش و خصوصیات کلی آن‌ها آشنا باشیم. یک داده گراف شامل اطلاعات زیر می‌باشد:

۱. ماتریس مجاورت^۵

این ماتریس، به ما شکل کلی شبکه و نحوه اتصال رئوس به یکدیگر را می‌دهد. در سطر i و ستون j آن اگر عدد ۱ قرار داشته باشد، به معنی وجود یالی از راس i به راس j است و اگر ۰ باشد، به معنی عدم وجود این یال است. گراف‌های مورد بررسی در این پروژه همگی بدون جهت هستند و این به این معنی است که

$$A = A^T$$

در آن‌ها برقرار است.

۲. برچسب گراف^۶

در مجموعه داده‌های برچسب‌گذاری شده گراف، هر گراف یک برچسب دارد که مربوط به کل گراف است. به طور مثال، اگر گراف‌ها نمایانگر مولکول‌ها باشند، برچسب گراف می‌تواند نشانگر خوش بو بودن یا نبودن مولکول باشد.

۳. ویژگی‌ها و برچسب رئوس^۷

در بعضی از مجموعه داده‌های گراف، هر کدام از رئوس گراف نیز می‌توانند برچسب و بردار ویژگی مربوط به خود داشته باشند.

Adjacency Matrix^۵
Graph Label^۶
Node Attributes and Labels^۷

۴. ویژگی‌ها و برچسب یال‌ها^۸

در بعضی از مجموعه داده های گراف، هر کدام از یال‌های گراف نیز می‌توانند برچسب و بردار ویژگی مربوط به خود داشته باشند. در مجموعه داده های بررسی شده در این پایان‌نامه، هیچ‌کدام از یال‌ها برچسب و بردار ویژگی نخواهند داشت.

۲-۳-۲ چالش‌ها

زمانی که دادگان ما به صورت گراف باشند، با چالش‌هایی روبرو می‌شویم که هنگامی که دادگان به صورت تصویر یا متن باشند، وجود ندارند. مشکل اصلی گراف‌ها این است که یک گراف با **جایگشت** خوردن رئوسش تغییر پیدا نمی‌کند، در حالی که چنین چیزی هنگام استفاده از دادگان تصویر و متن درست نیست. این نکته از این جهت دشواری به همراه دارد که هنگام ورودی دادن به مدل، بر اساس جایگشت رئوس یک گراف، می‌توان یک داده را به انواع مختلفی نمایش داد و انتظار می‌رود یکسان بودن اطلاعات در تمامی این نمایش‌ها به نحوی درون مدل به آن پرداخته شود. همچنین، از آنجایی که مسئله ما تولید گراف می‌باشد، ارزیابی خروجی مدل نیز به سادگی ارزیابی کردن یک تصویر و متن نخواهد بود و باید به دنبال روش‌های بهتری برای آن گشت.

برای رفع چالش اول راهکار پیشنهادی و استفاده شده در این پروژه، جایگشت دادن گراف قبل از هر ورودی و مرتب کردن براساس الگوریتم جستجوی اول عمق^۹ است. از آنجایی که مدل ما براساس یادگیری عمیق است، هر داده را بارها خواهد دید و در هر تکرار، رئوس گراف جایگشتی تصادفی خواهند داشت. این باعث می‌شود هر چقدر تعداد دفعات تکرار داده‌ها در زمان آموزش بیشتر شود، هر داده با نمایش‌های بیشتری توسط مدل دیده می‌شود که باعث بهبود تعمیم‌پذیری^{۱۰} مدل می‌شود. از طرف دیگر، با چیدن رئوس براساس الگوریتم جستجوی اول عمق، می‌توان سختی کار را **بسیار** کم کرد و تعداد نمایشی که یک گراف را بسیار محدودتر کرد.

۲-۴ آموزش مدل‌های مولد

همان‌طور که قبل‌تر به آن اشاره شد، خطا در مدل‌های مولد به سادگی بررسی یک پارامتر مانند دقت نیست و برای آموزش دادن آن‌ها نیازمند راهکارهای هوشمندانه‌تری هستیم. به طور مثال، اگر فرض کنید هدف یک مدل مولد تولید تصاویر انسان باشد، پارامتری مانند میزان انسان بودن وجود ندارد که با استفاده از آن بتوان تابع خطا ساخت. از این جهت، روش‌های متعددی برای

^۸ Edge Attributes and Labels^۹ Depth First Search (DFS)^{۱۰} Generalization

برطرف کردن این مشکل مطرح شده است. در این جا به دو مورد از روش های مشهور برای حل این مشکل اشاره می کنیم:

۱. روش های مبتنی بر کدگذار^{۱۱} و کدگشا^{۱۲}

در این روش ها، مدل حاوی دو بخش کلی کدگذار و کدگشا است. هدف کدگذار تبدیل ورودی به نماینده ای^{۱۳} در فضای با ابعاد پایین تر است، به طوری که نماینده یافته شده حاوی بخش اعظمی از اطلاعات ورودی باشد. هدف کدگشا نیز تبدیل یک نماینده از فضای با بعد پایین به خروجی ای در فضای اولیه است. با وجود این دو بخش، مدل در مرحله آموزش سعی می کند هر ورودی را به فضای با بعد پایین برده و تبدیل به یک نماینده در این فضا کند و سپس با استفاده از این نماینده، ورودی را بازتولید کند^{۱۴}.

پس از آموزش مدل، بخش کدگشا را می توان به چشم یک مدل مولد دید که در ورودی برداری تصادفی می گیرد و خروجی اش داده ی تولید شده مدنظر ما است.

۲. روش های مبتنی بر نمونه برداری

در روش های مبتنی بر نمونه برداری، مدل حاوی دو بخش مولد^{۱۵} و تفکیک کننده^{۱۶} است. هنگام آموزش در این روش ها، بخش مولد در تلاش است خروجی هایی تولید کند که مشابه داده های واقعی باشند. بخش تفکیک کننده نیز خروجی های تولید شده توسط مولد را مورد ارزیابی قرار می دهد. با رقابت این دو بخش مدل با همدیگر یادگیری انجام می شود. در این روش ها، بخش تفکیک کننده می تواند از پیش آموزش داده شده باشد و در طول آموزش مولد، تغییر نکند، و یا می تواند در حین آموزش مولد، تغییر کرده و آموزش ببیند. مدل های GAN^{۱۷} که بار اول در سال ۲۰۱۴ در مقاله [۱] مطرح شدند و موفقیت بسیار زیادی کسب کردند، بر اساس روش های مبتنی بر نمونه برداری عمل می کنند.

در این پروژه، ما از تفکر پشت هر دو این روش ها استفاده می کنیم و در نهایت خطایی برای مدل در نظر می گیریم که

شباهت هایی به هر دو این روش ها دارد. این خطا به طور دقیق تر به آن پرداخته خواهد شد.

^{۱۱} Encoder

^{۱۲} Decoder

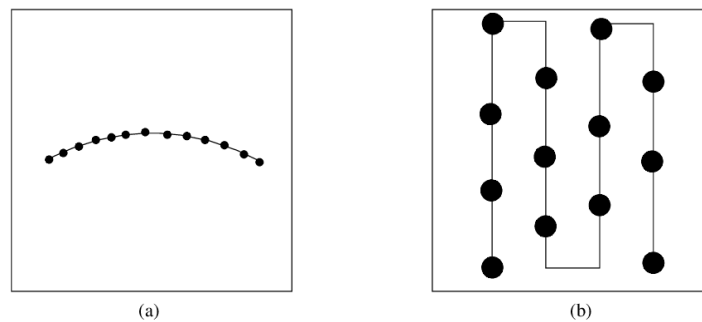
^{۱۳} representation

^{۱۴} reproduce

^{۱۵} Generator

^{۱۶} Discriminator

^{۱۷} Generative Adversarial Network



شکل ۲-۱: گراف‌های تصاویر بالا یکسان هستند اما نحوه‌ی چیده‌شدن رئوس آن‌ها تشخیص یکسان بودن‌شان را دشوار می‌کند.

۲-۵ ارزیابی مدل‌های مولد

در خیلی از مسائل مربوط به یادگیری ماشین، ارزیابی مدل پس از آموزش، بخش بسیار ساده‌ای محسوب می‌شود و معمولاً فقط چندین مشخصه^{۱۸} مانند صحت مورد بررسی قرار می‌گیرند و نمایانگر عملکرد مدل هستند. اما در مسئله تولید مشروط گراف، از آنجایی که هم مدل مولد است و هم خروجی گراف است، به این سادگی نیست. به طور خاص، در مسائل مربوط به تولید مشروط در حوزه تصویر و متن، اولین و ساده‌ترین روش ارزیابی خروجی‌های مدل، نگاه کردن به تصاویر و خواندن متن‌های تولید شده است. اما هنگامی که خروجی به صورت گراف است، از آنجایی که می‌توان یک گراف را به روش‌های مختلفی رسم کرد، رسم گراف‌های خروجی و سنجش چشمی آن‌ها، ارزیابی درستی نخواهد بود.

برای حل این مشکلات و ارزیابی گراف‌های خروجی، چندین مولفه را با هم در نظر می‌گیریم:

- محاسبه‌ی صحت^{۱۹} گراف‌های تولید شده با استفاده از دسته‌بند^{۲۰} های گراف

اولین نحوه‌ی ارزیابی استفاده شده برای یک مدل مشروط، محاسبه صحت گراف‌های خروجی آن با بهره‌گیری از دسته‌بند های گراف است. بدین جهت، پیشاپیش مستقل از روند آموزش مدل، تعدادی دسته‌بند برای مجموعه داده ورودی آموزش می‌دهیم که وظیفه هر کدام تخمین شرط از روی گراف تولید شده است. پس از اتمام آموزش مدل، از مدل تعداد زیادی خروجی با ورودی دادن شرط های متفاوت می‌گیریم. سپس به هر گراف خروجی به همراه شرط مدنظر ما برای آن، به چشم یک داده نگاه می‌کنیم. برای این مجموعه داده جدید تولید شده توسط مدل، صحت هر کدام از دسته‌بندها را روی آن می‌سنجیم. در نهایت، اگر صحت دسته‌بندها روی مجموعه داده اصلی و مجموعه داده تولید شده، نزدیک باشند، نمایانگر قدرت مدل و اگر از هم دور باشند، نمایانگر نقص مدل خواهد بود.

^{۱۸} parameter
^{۱۹} Accuracy
^{۲۰} Classifier

• مقایسه آماره‌های گراف‌های خروجی و گراف‌های مجموعه داده

دومین نحوه‌ی ارزیابی گراف‌های خروجی، مقایسه آماره^{۲۱}‌های آن‌ها با آماره‌های گراف‌های مجموعه داده اصلی است. پنج آماره مورد بررسی قرار می‌گیرند که به ترتیب عبارت‌اند از: **بزرگترین مجموعه همبند^{۲۲}**، **طول مسیر مشخصه^{۲۳}**، **معیار Gini**، **تعداد مثلث‌های گراف**، و **میانگین درجه رئوس گراف**. مشابه حالت قبل، در صورتی که این آماره‌ها برای گراف‌های تولید شده نزدیک آماره‌ها در گراف‌های مجموعه داده اصلی باشند، نمایانگر عملکرد مناسب مدل است و در غیر این صورت، نمایانگر عملکرد بد مدل خواهد بود.

MMD
هم حساب می‌کنیم؟

^{۲۱}Statistic
^{۲۲}Largest Connected Component (LCC)
^{۲۳}Characteristic Path Length (CPL)

فصل ۳

کارهای پیشین

در این قسمت چند کار مرتبط انجام شده در زمینه مدل‌های مولد و تولید مشروط را بررسی می‌کنیم و به بررسی کارهای انجام شده در حوزه‌های متن و گراف بسنده می‌کنیم. علت عدم بررسی کارهای مرتبط در حوزه تصویر، تفاوت بسیار زیاد راهکارها و چالش‌های بین دو حوزه تصویر و گراف است. اما از آنجا که در هر دو حوزه متن و گراف با داده‌های گسسته سرو کار داریم، رویکردهای مورد استفاده در این دو حوزه شباهت‌های زیادی دارند و بررسی آن‌ها کمک‌کننده خواهد بود.

۳-۱ تولید مشروط در حوزه متن

تولید مشروط متن حوزه تحقیقاتی مهمی در پردازش زبان طبیعی^۱ است که به طور گسترده‌ای مطالعه شده است. چند ایده پرتکرار در این حوزه عبارت‌اند از: استفاده از شبکه‌های عصبی بازگشتی^۲ برای تعبیه ورودی [۲، ۳، ۴]، آموزش دادن شبکه‌های مبتنی بر مدل‌های مولد خصمانه^۳ با استفاده از ایده‌های یادگیری تقویتی^۴ [۵، ۶، ۷]، شبکه‌های خودکدگذار متغیر^۵ [۸، ۹، ۱۰]، و بهره‌گیری از شبکه‌های ترانسفرمر^۶ [۱۱، ۱۲] درون مدل‌ها. در این بخش، نگاهی به روش‌های مبتنی بر معماری کدگذار و کدگشا و روش‌های مبتنی بر مدل‌های مولد خصمانه می‌کنیم.

^۱ Natural Language Processing

^۲ Recurrent Neural Networks (RNNs)

^۳ Generative Adversarial Networks

^۴ Reinforcement Learning (RL)

^۵ Variational Autoencoders

^۶ Transformer

۳-۱-۱ روش‌های مبتنی بر طراحی کدگذار و کدگشا

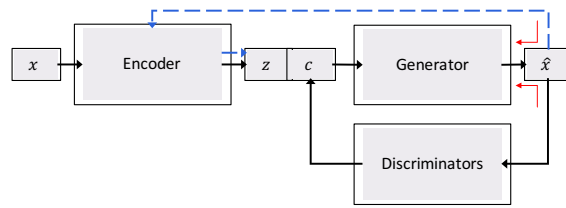
در سال ۲۰۱۴ مقاله [۲] چارچوب^۷ کدگذار و کدگشا را معرفی کرد. در این چارچوب، مدل‌ها از دو بخش کدگذار و کدگشا تشکیل می‌شوند. نحوه‌ی کار این مدل‌ها به این صورت است که ورودی مدل با استفاده از شبکه‌های عصبی بازگشتی، که بخش کدگذار مدل را تشکیل می‌دهد، تبدیل به برداری با طول ثابت می‌شود. سپس بردار بدست آمده به عنوان حالت اولیه شبکه‌ی کدگشا قرار داده شده و شبکه‌ی کدگشا متن خروجی را تولید می‌کند. چارچوب ارائه شده در حوزه متن موفقیت بسیار زیادی بدست آورده و افراد زیادی مدل‌هایشان را بر اساس این چارچوب طراحی کرده و آموزش می‌دهند. به طور مثال، در سال ۲۰۱۹ مقاله [۳] از این چارچوب استفاده کرد و توانست برای مسئله رزرو شخصی‌سازی شده رستوران راه‌حلی ارائه بدهد.

گرچه این چارچوب معماری سطح بالای مدل را مشخص و معلوم می‌کند، اما درباره‌ی معماری بخش‌های جزئی آن حرفی نمی‌زند. به عنوان مثال، در مقاله [۲] بردار با طول ثابت، که بردار نهان نامیده می‌شود، آزاد و بدون هیچ شرطی در نظر گرفته شده بود و هدف مدل، فشردن و بازتولید جملات ورودی بود. به این دسته از مدل‌ها، مدل‌های خودکدگذار^۸ گفته می‌شود. اما بعدتر، در مقاله [۸] مدل‌های خودکدگذار متغیر ارائه شد که در آن‌ها، مدل تلاش می‌کند که بردارهای نهان را به گونه‌ای تولید کند که علاوه بر اینکه کدگشا بتواند از آن‌ها جمله‌ی ورودی را بدست آورد، از توزیع نرمال حول مبدا آمده باشند و با این کار استقلال بین اعضای بردار نهان ایجاد شود. اضافه کردن این تغییر در مدل اولیه باعث شد که علاوه بر اینکه در بعضی از مسئله‌ها، مدل‌های خودکدگذار متغیر نتایج بهتری کسب کردند، ویژگی‌های بدست آمده در بردار نهان‌شان از هم گسیخته‌تر^۹ شوند [۱۳] که این باعث قابل فهم‌تر شدن ویژگی‌ها می‌شود.

یکی از روش‌های مطرح شده برای مسئله تولید مشروط متن که مبتنی بر مدل‌های خودکدگذار متغیر است، روش ارائه شده در مقاله [۴] است. در این روش ابتدا با استفاده از یک شبکه عصبی بازگشتی جملات ورودی، که با x نمایش داده می‌شوند، تبدیل به بردار نهان با نام z می‌شود. سپس شرط ورودی، که c نام دارد، را به بردار نهان بدست آمده می‌چسبانند. اعضای بردار c برخلاف بردار z می‌توانند هم گسسته و هم پیوسته باشند تا انواع شروط را پوشش دهند. به عنوان مثال، در مقاله یکی از شرط‌های بردار c در بخش آزمایش‌ها، مثبت یا منفی بودن پیام جمله است. پس از تشکیل بردار (z, c) ، این بردار به عنوان حالت اولیه بخش مولد، که خود یک شبکه عصبی بازگشتی LSTM^{۱۰} است، استفاده می‌شود تا جمله \hat{x} که خروجی ساخته مولد است، بدست آید. در گام آخر، برای بررسی عملکرد جمله \hat{x} دو کار انجام می‌گیرد.

اولین کار، بررسی میزان پایبند بودن جمله \hat{x} به شرط c است. برای این کار، به تعداد اعضای c تفکیک‌کننده قرار

Framework^vAutoencoder^۸Disentangled^۹Long short-term memory^{۱۰}



شکل ۳-۱: مدل مولد مشروط متن [۴]. بردارهای قرمز نمایانگر جریان گرادیان هستند و بردار آبی نمایانگر شرط استقلال است.

می‌دهند که هر یک وظیفه دارد میزان خطای بدست آوردن مولفه مربوطه‌اش در c را از جمله \hat{x} محاسبه کند. چالش اصلی این قسمت، گسسته بودن جمله \hat{x} است که بازگشت گرادیان را غیر ممکن می‌کند. برای رفع این مشکل، هنگام محاسبه \hat{x} ، به جای استفاده از بردارهای یک‌گرم^{۱۱}، از بردار احتمال پیوسته با دمای کم‌شونده استفاده می‌کنند. به طور دقیق خروجی \hat{x}_t برابر است با:

$$\hat{x}_t \sim \text{softmax}(o_t / \tau)$$

که o_t ورودی‌های تابع softmax و پارامتر τ دمای سیستم است که در ابتدا برابر ۱ است. در طول آموزش، با $\tau \rightarrow 0$ حاصل \hat{x}_t به بردار یک‌گرم نزدیک‌تر می‌شود و در نهایت مشابه حالت گسسته عمل می‌کند.

دومین کار، ایجاد استقلال بین متغیرهای شرط c و بردار نهان z است. در حالت ایده‌آل، انتظار از مدل آن است که بتوان متغیرهای شرطی c را تغییر داد بدون آن که تغییری در بردار نهان z رخ دهد. راهکار آن‌ها برای بدست آوردن این استقلال آن است که بخش مولد طوری خروجی \hat{x} را تولید کند که بتوان با استفاده از آن، مستقل از آن که مقدار c چیست، تمامی اطلاعات موجود در z را بازتولید کرد. ازین جهت، خروجی \hat{x} به بخش کدگذار مدل داده می‌شود تا متغیر نهان z نظیر آن محاسبه شود و خطایی برحسب فاصله z, \hat{z} به خطای کل مدل اضافه می‌شود. این شرط استقلال را می‌توان در شکل ۳-۱ به صورت بردار آبی‌رنگ دید.

با این‌که شبکه‌های بازگشتی پرکاربردترین شبکه‌ها در تولید متن هستند، اما دو اشکال عمده دارند. اولین مشکل آن‌ها این است که در روند آموزش، تابع درست‌نمایی^{۱۲} را بیشینه می‌کنند که این مشکلی به نام Exposure Bias را به وجود می‌آورد. به طور ساده شده، در شبکه‌های بازگشتی، در زمان آموزش بخش کدگشا کلمات واقعی را به عنوان ورودی برای تولید کلمه بعد می‌گیرند، در حالی که هنگام ارزیابی مدل، کلمه خروجی مدل در مرحله قبل به عنوان ورودی داده می‌شود. این تفاوت بین نحوه آموزش و ارزیابی مشکلاتی را به همراه دارد. به عنوان مثال، اگر مدل در یک گام خروجی نامناسبی تولید کند، این خروجی نامناسب باعث نامناسب شدن خروجی‌های مراحل بعد نیز می‌شود. دومین مشکل شبکه‌های بازگشتی مربوط

^{۱۱} one-hot
^{۱۲} Likelihood

به توابع خطای آن‌ها است. اکثر توابع خطا، خطا را کلمه به کلمه محاسبه می‌کنند در حالی که ارزیابی بر اساس کل جمله است. مشابه قبل، این تفاوت بین آموزش و ارزیابی باعث ایجاد مشکلاتی می‌شود. برای رفع این مشکلات، روش‌های مبتنی بر مدل‌های مولد خصمانه مورد استفاده قرار گرفتند.

۳-۱-۲ روش‌های مبتنی بر مدل‌های مولد خصمانه

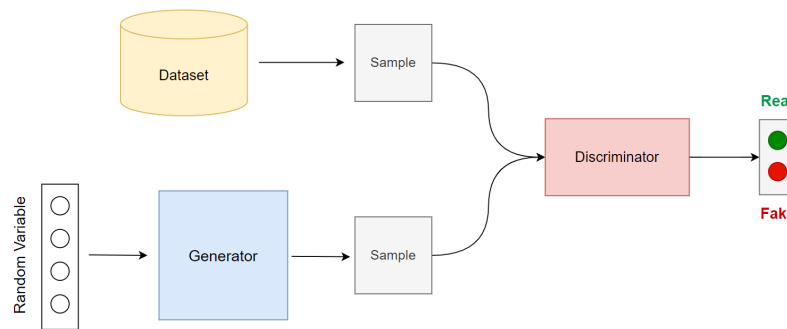
مشابه چارچوب کدگشا و کدگذار، در سال ۲۰۱۴ مقاله [۱] برای اولین بار شبکه‌های مولد خصمانه^{۱۳} را برای تولید تصاویر معرفی کرد. روش ارائه شده دیدگاه جدیدی برای تولید داشت که با دیدگاه‌های معمول آن زمان بسیار فرق داشت. تا قبل از آن، اکثر مدل‌های ارائه شده برای تولید تصاویر، در درون خود به دنبال تقریب زدن توزیع عکس‌ها و مدل‌سازی آن‌ها بودند و در مرحله ارزیابی، از توزیع تخمین‌زده نمونه‌برداری انجام می‌دادند. اما ایده اصلی شبکه‌های مولد خصمانه این بود که به جای یادگیری و مدل‌سازی توزیع داده‌ها هنگام آموزش و نمونه‌برداری از توزیع تخمین‌زده شده هنگام ارزیابی، مدل به صورت مستقیم در مرحله آموزش یاد بگیرد از توزیع اصلی داده‌ها نمونه‌برداری انجام بدهد. این شبکه‌ها در ابتدا چالش‌های زیادی برای یادگیری داشتند و گزینه استواری^{۱۴} نبودند اما با تحقیقات تکمیلی انجام شده روی آن‌ها، هم اکنون به یکی از پرکاربردترین و مشهورترین شبکه‌ها تبدیل شده‌اند.

در حالت ساده، شبکه‌های مولد خصمانه دارای دو بخش مولد و تفکیک‌کننده هستند. بخش مولد معمولاً در ورودی یک بردار تصادفی می‌گیرد و این بردار تصادفی را پس از گذراندن از لایه‌های متوالی، تبدیل به نمونه‌ای از داده مد نظر می‌کند. بخش دوم، تفکیک‌کننده‌ای با دو دسته است. وظیفه تفکیک‌کننده دسته‌بندی نمونه‌های ورودی داده شده، به دو دسته واقعی و ساختگی^{۱۵} است. برای آموزش دادن تفکیک‌کننده، به آن نمونه‌هایی از داده‌های واقعی مجموعه داده و همچنین نمونه‌هایی از داده‌های تولید شده توسط بخش مولد نشان داده می‌شود و تفکیک‌کننده باید بتواند این دو گروه از نمونه‌ها را از یک‌دیگر جدا کند. از طرف دیگر، بخش مولد در تلاش خواهد بود تا داده‌های خروجی‌اش را طوری ایجاد کند تا بتواند تفکیک‌کننده را فریب دهد و داده‌هایش را به عنوان داده‌های واقعی از دیدگاه تفکیک‌کننده جا بزند. برخلاف روش‌های قبلی، در این روش دو بخش مدل به جای همکاری، با یکدیگر رقابت می‌کنند و توابع خطایشان بر علیه همدیگر عمل می‌کنند.

^{۱۳} Generative Adversarial Networks

^{۱۴} Stable

^{۱۵} Fake



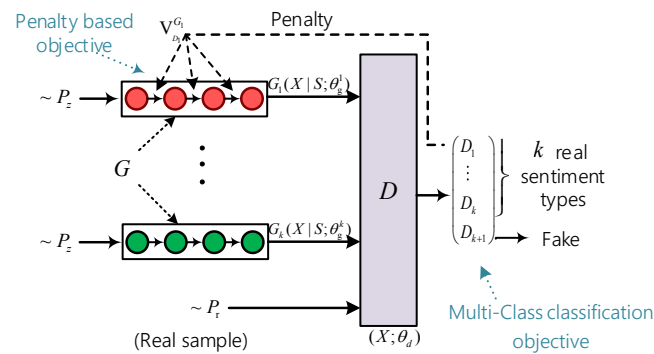
شکل ۳-۲: معماری مدل‌های مولد خصمانه.

یکی از روش‌های مطرح شده در زمینه تولید مشروط متن که بر اساس مدل‌های مولد خصمانه است، مدل ارائه شده در مقاله [۷] است. در این مدل، مسئله تولید مشروط با شرط گسسته بررسی و برایش راه‌حل پیشنهاد می‌شود. در این مسئله، فرض بر این است که هر جمله دارای یک برجسب از k برجسب موجود است. به طور مثال، در حالت $k = 2$ هر جمله می‌تواند پیامش مثبت یا منفی باشد. چارچوب مطرح شده در این مقاله که نام **SentiGAN** را دارد تشکیل شده است از k مولد و یک تفکیک‌کننده با $k + 1$ دسته. در شکل ۳-۳ می‌توان تصویری از معماری مدل دید که در آن G_1, \dots, G_k مولدها و D تفکیک‌کننده‌ی گفته شده است.

نحوه کار مدل به این صورت است که مولد G_i مسئول تولید نمونه‌هایی با برجسب i ام است. در مثال $k = 2$ این به معنی وجود ۲ مولد است که مولد اول مسئول تولید جملات با معنی مثبت و مولد دوم مسئول تولید جملات با معنی منفی است. تفکیک‌کننده D هم دارای $k + 1$ دسته است که k دسته اول آن مربوط به k برجسب موجود و دسته $k + 1$ مربوط به نمونه‌های ساختگی است. تفکیک‌کننده وظیفه‌اش این است که با دیدن داده‌های واقعی، هر کدام را در دسته مربوط به برجسب آن قرار بدهد و با دیدن داده‌های تولید شده توسط هر کدام از مولدها، آن‌ها را در دسته $k + 1$ ام مربوط به داده‌های ساختگی قرار دهد. از طرف دیگر، وظیفه مولد G_i آن است که نمونه‌هایش را به گونه‌ای تولید کند که تفکیک‌کننده آن‌ها را در دسته i ام قرار بدهد.

در این مقاله، برای حل مشکل جریان نیافتن گرادینان از انتخاب‌های گسسته، به مسئله به صورت یک فرآیند تصمیم‌گیری پی‌درپی نگاه می‌کنند [۱۴]. در ساختار مسئله، این به این معنی است که هر مولد در زمان t احتمال تولید کلمه x_{t+1} را با فرض وجود کلمات x_1, x_2, \dots, x_t ، محاسبه می‌کند. اما مشکلی که به وجود می‌آید آن است که تفکیک‌کننده مورد استفاده فقط آموزش دیده است تا جملات کامل را تفکیک کند در حالی که ممکن است $x_{1:t+1} = \{x_1, x_2, \dots, x_{t+1}\}$ جمله‌ی کاملی نباشد. برای دور زدن این مشکل، به ازای هر t ، با استفاده از روش مونت کارلو^{۱۶} تعدادی جمله کامل که شروع آن‌ها

^{۱۶} Monte Carlo



شکل ۳-۳: معماری مدل SentiGAN با k مولد و یک تفکیک‌کننده با $k + 1$ دسته [۷].

است، با استفاده از احتمال‌های تولید شده توسط مولد، نمونه‌برداری می‌شوند و میانگین خطای جملات نمونه‌برداری شده به عنوان تخمینی از خطای واقعی استفاده می‌شود.

در این مقاله، برای مولدها از شبکه‌های بازگشتی LSTM استفاده شده است. این مولدها با داشتن t کلمه اول، احتمال‌های کلمه $t + 1$ را محاسبه می‌کنند. همچنین برای تفکیک‌کننده از شبکه‌ی عصبی کانولوشنی^{۱۷} استفاده می‌شود.

راهکار پیشنهاد شده در این مقاله خوبی‌های زیادی دارد. به عنوان مثال، یکی از خوبی‌های این مدل آن است که می‌توان با استفاده از هر یک از مولدها، جملاتی با هر کدام از k برچسب تولید کرد به طوری که همگی t کلمه اولشان یکسان باشند که این عملکردی مشابه سامانه‌های تکمیل خودکار متن است. اما یکی از مشکلات اصلی آن، عدم امکان استفاده از آن برای مواقعی است که شرط‌های پیوسته وجود دارد و یا تعداد دسته‌ها (k) زیاد است. علت این مشکل آن است که تعداد پارامترهای مدل به طور خطی با افزایش k افزایش پیدا می‌کند. این باعث می‌شود در روند آموزش، از آنجایی که از هر مولد تعدادی جمله نمونه‌برداری می‌کنیم و همچنین هر مولد باید در هر گام تغییر داده شود، زمان لازم برای آموزش دادن بسیار زیاد می‌شود.

۳-۲ تولید در حوزه گراف

استفاده از یادگیری ماشین در حوزه‌ی گراف، زمینه جدیدی است که اخیراً افراد زیادی در آن شروع به کار کرده‌اند. در حوزه گراف، به دلیل شباهت‌هایی که گراف‌ها به متون از لحاظ داده‌ای دارند، مدل‌های ارائه شده برای تولید گراف شباهت ساختاری زیادی به مدل‌های موجود در زمینه متون دارند. در اینجا به بررسی دو نمونه از آن‌ها می‌پردازیم.

^{۱۷}Convolutional Neural Network

اولین مدل مورد بررسی ما در مقاله [۱۵] ارائه شده که نامش **GraphRNN** است. همان‌طور که از اسم آن می‌توان حدس زد، مدل ارائه شده مدلی بر اساس معماری مدل‌های بازگشتی است که برای کار با داده‌های گراف طراحی شده است. این روش، به یک گراف با جایگشت رئوس ثابت، به چشم یک دنباله نگاه می‌کند و مدل ترتیبی^{۱۸} برای تولید رئوس و یال‌های گراف پیشنهاد می‌دهد. در این راه، مدل مولد شامل ۲ بخش اصلی است که عبارت‌اند از بخش مولد رئوس و بخش مولد یال‌ها.

نحوه‌ی عملکرد مدل به این صورت است که بخش مولد رئوس با توجه به رئوس و یال‌های تولید شده در زمان t دو خروجی h_t و θ_t را تولید می‌کند. خروجی h_t نماینده وضعیت گراف تا مرحله t ام و θ_t نماینده توزیع یال‌های راس t با رئوس قبل از آن است. در مقاله S_t^π نمایانگر $t - 1$ ستون اول از سطر t ام در ماتریس مجاورت است. به عبارت دیگر، S_t^π برداری به طول $t - 1$ است که خانه‌ی i آن ۱ است اگر راس t به راس i وصل باشد و ۰ است اگر وصل نباشد. در این صورت، روابط بین h_t, θ_t عبارت‌اند از:

$$h_t = f_{trans}(h_{t-1}, S_{t-1}^\pi) \quad (۱-۳)$$

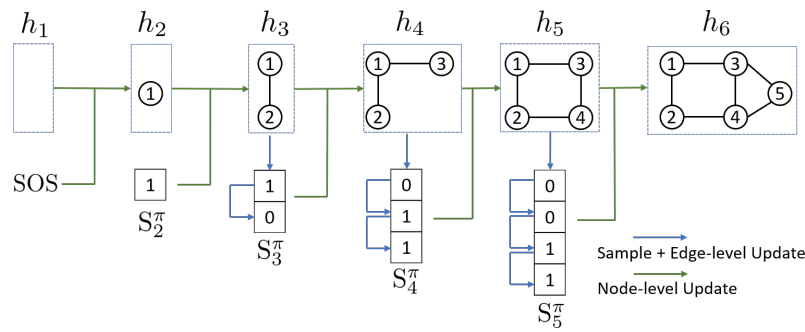
$$\theta_t = f_{out}(h_t) \quad (۲-۳)$$

که در آن‌ها f_{out} و f_{trans} شبکه‌های عصبی دلخواه هستند. خروجی θ_t پس از تولید، به عنوان حالت اولیه شبکه بازگشتی یال‌ها قرار می‌گیرد و این شبکه S_t^π را مرحله به مرحله تولید می‌کند. خروجی مرحله i شبکه‌ی یال‌ها ۱ یا ۰ بوده و این مقدار برابر مقدار موجود در خانه‌ی i ام S_t^π است. به طور خلاصه، این مدل، ماتریس مجاورت یک گراف را سطر به سطر تولید می‌کند.

راه‌کار پیشنهادی این مقاله برای مشکل جایگشت رئوس در گراف‌ها این است که هنگام ورودی دادن گراف به مدل، گراف را با استفاده از الگوریتم جستجوی سطح اول^{۱۹} ترتیب می‌دهد. با استفاده از این کار، می‌تواند تعداد جایگشت‌های نمایی گراف‌ها را به شدت کاهش دهد. برای مطالعه دقیق‌تر خوبی‌های این نوع ترتیب دادن به رئوس، به مقاله [۱۵] مراجعه کنید.

مدل مطرح شده در این مقاله، یکی از اولین مدل‌های مولد برای تولید گراف است و انتظار می‌رود مشکلات زیادی به همراه داشته باشد. یکی از بزرگترین مشکلات این مدل (که مشکل رایجی در شبکه‌های بازگشتی است) این است که اکثر عملیات مدل به صورت پشت سر هم و خطی انجام می‌گیرند و مدل، مدلی ترتیبی است. این وابستگی هر مرحله به نتایج مراحل قبل باعث می‌شود نتوانیم از قابلیت اصلی کارت‌های گرافیکی، که موازی‌سازی کارها است، برای افزایش سرعت

^{۱۸} Sequential
^{۱۹} Breadth First Search (BFS)



شکل ۳-۴: معماری مدل GraphRNN که شامل دو بخش مولد رئوس و مولد یال‌ها است [۱۵].

آموزش استفاده موثری بکنیم. ازین جهت مقدار زمان مورد نیاز هم برای آموزش این مدل و هم برای تولید گراف توسط آن بسیار زیاد است و برای گراف‌های با تعداد رئوس بالا قابل استفاده نیست.

روش دیگر مورد بررسی ما مدل ارائه شده در مقاله [۱۶] است که یکی از اهدافش بهبود سرعت مدل‌های ترتیبی مانند GraphRNN است و همچنین مدل مورد استفاده ما در این پروژه خواهد بود. در این مدل، ساده و بدون جهت بودن گراف‌ها فرض می‌شود. ویژگی‌های مهم راه‌حل ارائه شده در این مقاله عبارت‌اند از ارائه چارچوب تنظیم اندازه بلوک و اندازه گام^{۲۰} برای سبک و سنگین کردن بین کیفیت و سرعت مدل، ارائه شبکه عصبی گراف بر اساس مکانیزم توجه^{۲۱} برای کاستن تاثیر جایگشت رئوس و بهبود کارایی، و مدل‌سازی خروجی به صورت ترکیب توابع برنولی^{۲۲}.

نحوه‌ی عملکرد مدل شبیه روش‌های مبنی بر کدگشا و کدگذار می‌باشد. روند اجرای برنامه در هر گام به این صورت است که B راس تولید می‌کند که B اندازه بلوک است. سپس از B راس تولید شده، S تای اول که S اندازه گام است را نگه می‌دارد و بقیه را دور می‌اندازد. با انجام این کار، افزایش و کاهش دادن پارامترهای S , B باعث تغییر در سرعت و کیفیت مدل می‌شود. از آنجایی که در اکثر آزمایشات $B = S = 1$ در نظر گرفته شده است، ما هم همین فرض را برای ادامه این توضیحات می‌کنیم. با وجود این فرض، $b_t = \{t\}$ خواهد شد.

مدل در ابتدا برای هر یک از رئوس با استفاده از یک لایه خطی نمایش اولیه‌ای محاسبه می‌کند که نام آن را $h_{b_i}^*$ می‌گذارد. سپس این نمایش‌های اولیه را به یک شبکه‌ی عصبی گراف که با ارسال پیام^{۲۳} کار می‌کند، به عنوان ورودی می‌دهد و پس از طی R لایه انتقال پیام و توجه، نمایش‌های نهایی رئوس t را که نام آنها $h_{b_i}^R$ است به عنوان خروجی از شبکه عصبی گراف می‌گیرد. در انتها برای مدل‌سازی توزیع یال‌ها، از ترکیب K توزیع برنولی استفاده می‌شود. روابط محاسبه خروجی به صورت

^{۲۰}Block Size and Sample Stride

^{۲۱}Attention

^{۲۲}Mixture of Bernoulli Distributions

^{۲۳}Message Passing

زیر در می‌آیند:

$$\alpha_1, \alpha_2, \dots, \alpha_K = \text{Softmax}(\sum_{i \in b_t, 1 \leq j \leq i} MLP_\alpha(h_i^R - h_j^R)) \quad (3-3)$$

$$\theta_{1,i,j}, \theta_{2,i,j}, \dots, \theta_{K,i,j} = \text{Sigmoid}(MLP_\theta(h_i^R - h_j^R)) \quad (4-3)$$

$$p(L_{b_t}^\pi | L_{b_1}^\pi, \dots, L_{b_{t-1}}^\pi) = \sum_{k=1}^K \alpha_k \prod_{i \in b_t} \prod_{1 \leq j \leq i} \theta_{k,i,j} \quad (5-3)$$

در ابتدا بسته به آنکه نماینده‌های نهایی رئوس چقدر با هم‌دیگر فاصله دارند، ضرایب K توزیع برنولی را بدست می‌آورند. سپس برای دو راس مانند i, j احتمال وجود یالی بین این دو را در هر کدام از توزیع‌های برنولی محاسبه می‌کنند. در انتها، احتمال وجود یال‌ها برابر ضرب تعدادی برنولی می‌شود و هر کدام از برنولی‌ها با اعمال ضریب مربوطه‌شان با هم‌دیگر مخلوط شده و احتمال نهایی بوجود می‌آید. در این مدل، فرض می‌شود یال‌های متصل به راس t با فرض معلوم بودن $t - 1$ راس قبل از هم‌دیگر مستقل هستند.

شایان ذکر است که این مدل به طور مستقیم در چارچوب کدگشا و کدگذار قرار نمی‌گیرد، زیرا به جای بردن کل گراف به فضای نهان، این کار را برای هر راس انجام می‌دهد. اما روند کلی اجرای آن بسیار شبیه روند کار در آن چارچوب است.

فصل ۴

مدل و نتایج

در این فصل ابتدا بررسی اجمالی از مدل انجام می‌شود. سپس نحوه‌ی دور زدن مشکل گسستگی در مدل ارائه شده توضیح داده می‌شود. در نهایت نتایج بدست‌آمده مدل در تعدادی آزمایش ارائه و بررسی می‌شود.

۴-۱ بررسی اجمالی مدل

ایده‌ی اصلی رویکرد ما برای مسئله تولید مشروط گراف، تبدیل مولد بدون شرط گراف به مولدی مشروط است. همان‌طور که در کارهای پیشین مربوط به حوزه متون دیدیم، بسیاری از مدل‌های مطرح شده برای تولید مشروط، از تغییر دادن مدلی بدون شرط بوجود می‌آیند و تغییر داده شده معمولاً اضافه کردن بردار نماینده شرط به بردار نهان است. در اینجا کار مشابهی انجام می‌شود.

مدل مطرح شده ما حاوی ۲ بخش اصلی است که عبارت‌اند از مولد و دسته‌بند. بخش مولد مدل، مولد گراف بدون شرط GRAN است که کمی تغییر در آن داده شده است.

به مدل در ورودی یک زیرگراف تصادفی به همراه شرط مربوط به گراف کامل داده می‌شود. نحوه‌ی کار مدل به این صورت است که در ابتدا شرط داده‌شده در ورودی پس از یک تبدیل خطی، تبدیل به بردار نماینده شرط در فضایی نهان می‌شود. این بردار، به بردار نهان تمامی رئوس، قبل از ورودی داده شدن به شبکه عصبی گراف درون GRAN، متصل شده و حاصل به عنوان ورودی به شبکه عصبی گراف داده می‌شود. با انجام این کار می‌توانیم وجود رابطه‌ی بین شرط گراف با نحوه‌ی اتصال رئوس به یک‌دیگر را مدل‌سازی کرده و بردارهای نهان انتهایی هر راس را به نحوی تولید کنیم که شرط گراف

آن در بردار نهانش لحاظ شده باشد. به طور دقیق اگر شرط گراف را c بنامیم، روابط زیر برقرار خواهند بود:

$$\hat{c} = Wc + b \quad (۱-۴)$$

$$\hat{h}_{b_t}^* = (h_{b_t}^*, \hat{c}) \quad (۲-۴)$$

$$h_{b_1}^R, \dots, h_{b_t}^R = GNN(\hat{h}_{b_1}^*, \dots, \hat{h}_{b_t}^*) \quad (۳-۴)$$

با استفاده از $h_{b_i}^R$ های جدید، مشابه قبل خطای GRAN محاسبه می شود. ما این خطا را \mathcal{L}_{adj} می نامیم.

علاوه بر این کار، از راهکار نمونه برداری و تفکیک کنندگی استفاده می کنیم تا مدل را مشروط کنیم. ابتدا قبل از شروع آموزش مدل، یک دسته بند گراف آموزش داده می شود تا بتواند برای گراف های مجموعه داده، شرط را تخمین بزند. در مجموعه داده های بررسی شده، بردارهای شرط، بردارهایی یک گرم هستند و هر گراف در یکی از k دسته قرار می گیرند. پس از آموزش دسته بند، در مرحله یادگیری مدل مشروط، دسته بند بارگذاری^۱ می شود و در روند آموزش مدل مشروط، ثابت می ماند. در آموزش مدل مشروط، دو پارامتر تعداد نمونه ها ($samples$) و تعداد گام ها ($steps$) وجود دارد. در هر مرحله آموزش، به اندازه $samples$ گراف با استفاده از مدل تولید می کنیم به طوری که هر کدام از آن ها در گام صفرم همان زیرگراف ورودی داده شده اند و به تعداد $steps$ راس جدید به همراه یال های متصل به آن ها به هر کدام اضافه می شود. این کار، مشابه کاری است که در SentiGAN انجام می شود اما لزوماً مشابه آن، تا تولید کامل گراف جلو نمی رویم. سپس، نمونه های تولید شده به دسته بند داده شده و مقدار خطای هر کدام محاسبه می شوند. از آنجایی که تولید کردن راس t گراف به نحوی که گراف بدست آمده دارای شرط مناسب باشد، با افزایش t راحت تر و با کاهش آن سخت تر می شود، بسته به آن که زیرگراف ورودی چند راس از گراف اصلی را ندارد، پارامتر کاهنده ای در خطای این نمونه ها ضرب می شود. برای روشن شدن این موضوع، فرض کنید زیرگراف ورودی فقط ۱ راس از گراف اصلی را ندارد. در آن صورت تولید ۱ راس به طوری که گراف حاصل دارای شرط گراف اصلی باشد کار ساده ای محسوب می شود. اما اگر زیرگراف ورودی نصف رئوس گراف اصلی اش را نداشته باشد، تولید راس بعدی به طوری که گراف بدست آمده شرط نهایی را داشته باشد، کار به مراتب دشوارتری است. به عبارت دیگر، هر چقدر فاصله بین تعداد رئوس زیرگراف و گراف اصلی کمتر شود، بازتولید گرافی مشابه گراف اصلی، کار ساده تری است. ازین جهت، هنگام محاسبه خطا، بسته به آنکه زیرگراف ورودی چقدر با گراف اصلی فاصله دارد، باید خطایش کم شود. به طور دقیق اگر \hat{G} زیرگراف ورودی، G گراف کامل آن، و γ پارامتر کاهنده باشند، آنگاه

$$\mathcal{L}_{conditional} = \gamma^{|G|-|\hat{G}|} \times \mathcal{L}_{classifier}(samples) \quad (۴-۴)$$

خطای نهایی بخش مشروط کردن مدل خواهد بود.

برای مجموعه داده هایی که در آن ها رئوس نیز دارای برجسب اند، خطای نوع سومی نیز در نظر گرفته می شود. هنگام کار با این مجموعه داده ها، به مدل، یک بخش تخمین زن برجسب رئوس که نام آن را Node Classifier قرار می دهیم، اضافه می شود که ورودی آن، بردار نهان نهایی هر راس $h_{b_i}^R$ است و هدف آن تخمین زدن برجسب آن راس است. تابع خطای استفاده شده برای آن نیز مشابه توابع خطا دسته بند های معمولی خواهد بود. مشابه استدلال بخش قبل، این مقدار نیز در ضریب کاهنده مربوط به زیرگراف بودن ورودی ضرب می شود. به صورت دقیق

$$\mathcal{L}_{node-label} = \gamma^{|G|-|\hat{G}|} \times \mathcal{L}_{node-classifier}(nodes) \quad (5-4)$$

خطای مربوط به این بخش خواهد شد.

خطای نهایی مدل که گرادیان ها با استفاده از آن بدست می آید، ترکیب خطی از سه خطای فوق است. به طور دقیق

$$\mathcal{L} = \mathcal{L}_{adj} + \lambda_1 \mathcal{L}_{conditional} + \lambda_2 \mathcal{L}_{node-label} \quad (6-4)$$

خطای نهایی مدل است. در این رابطه، پارامترهای λ_1 ، λ_2 جزو ابرپارامتر^۲ های سیستم خواهد بود که بسته به مجموعه داده ها، می توان آن ها را تنظیم کرد.

۲-۴ مشکل گسسته بودن و عدم جریان گرادیان

در جریان کار مدل، در خطای نوع دوم $\mathcal{L}_{conditional}$ با مشکل گسسته بودن گراف روبرو می شویم. مشکل موجود آن است که برای افزودن راس جدید به زیرگراف ورودی و انتخاب یال های متصل به آن، از آنجایی که بودن یا نبودن یال امری گسسته است، گرادیان حاصل از خطای دسته بند نمی تواند از آن جریان پیدا کند. برای رفع این مشکل، از تکنیک ارائه شده در [۱۷] به نام نورون تصادفی دودویی^۳ استفاده می کنیم. با استفاده از این تکنیک، می توان از یک توزیع برنولی دوتایی، به نحوی نمونه برداری کرد که جریان گرادیان به هم نخورد. به طور کلی زمانی که می خواهیم راس جدیدی تولید کنیم، برای تولید یال هایش اگر در حال آموزش باشیم، با استفاده از نورون تصادفی دودویی نمونه برداری را انجام می دهیم و در صورتی که در حال ارزیابی باشیم، از توزیع برنولی نمونه گیری می کنیم.

^۲ Hyperparameters

^۳ Binary Stochastic Neuron

جدول ۴-۱: خلاصه اطلاعات مجموعه داده‌های PROTEINS و مجموعه داده ساختگی ErdosRenyi-0.25-0.75-50.

مجموعه داده	PROTEINS	ErdosRenyi-0.25-0.75-50
تعداد گراف‌ها	۱۱۱۳	۲۰۰۰۰
بیشینه تعداد رئوس	۶۲۰	۵۰
میانگین تعداد رئوس	۳۹/۰۵	۵۰
تعداد دسته‌ها	۲	۲
توزیع فراوانی دسته‌ها	۰/۶, ۰/۴	۰/۵, ۰/۵

۳-۴ آزمایش‌ها و نتایج

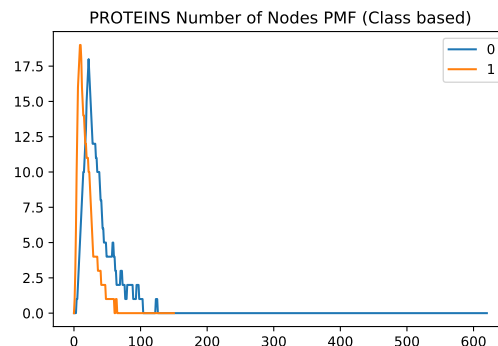
در این قسمت مدل ارائه شده را روی داده‌های واقعی آزمایش کرده و نتایج را بررسی می‌کنیم.

۱-۳-۴ مجموعه داده‌ها

دو مجموعه داده در این پایان‌نامه مورد بررسی قرار گرفته‌اند که عبارت‌اند از مجموعه داده‌های مولکولی **PROTEINS** و مجموعه داده‌های ساختگی گراف‌های Erdos Renyi با مقادیر p مختلف. خلاصه‌ای از این دو مجموعه داده را می‌توان در جدول ۴-۱ دید.

مجموعه داده‌های PROTEINS همانطور که در جدول ۴-۱ می‌توان دید، گراف‌هایش دو دسته دارند. نمودار توزیع تعداد رئوس گراف‌ها بر حسب دسته‌شان را می‌توان در شکل ۴-۱ مشاهده کرد. بنابر شکل، میانگین تعداد رئوس گراف‌های مربوط با دسته اول بیشتر است که این مطلب در ادامه هنگام بررسی نتایج اثرگذار خواهد بود.

مجموعه داده‌های Erdos Renyi نیز مجموعه داده‌ای ساختگی است که از ۲۰۰۰۰ گراف با ۵۰ راس تشکیل شده است. گراف‌های Erdos Renyi دسته‌ای از گراف‌های تصادفی‌اند که دارای یک پارامتر p هستند که نمایانگر احتمال وجود هر یال است. برای تولید یک گراف از این دسته، به ازای هر یال، نمونه‌ای از توزیع برنولی با پارامتر p کافیست برداشته شود تا وجود یا عدم وجود یال مشخص شود. در مجموعه داده مورد آزمایش، ۱۰۰۰۰ گراف با $p = ۰/۲۵$ و ۱۰۰۰۰ گراف با $p = ۰/۷۵$ دسته ۱ را تشکیل می‌دهند و هدف مدل، تولید گراف مشروط بر این دسته‌ها است. آزمایش مدل روی این مجموعه داده ساختگی از آنجایی که این مجموعه داده بسیار ساده است، صرفاً برای بررسی سالم بودن مدل است.



شکل ۴-۱: توزیع تعداد رئوس مجموعه داده PROTEINS بر حسب کلاس.

۴-۳-۲ بررسی نتایج آزمایش

هنگام ارزیابی تمامی آزمایش‌ها، دقت مجموعه داده تولید شده توسط مدل به وسیله چند تفکیک‌کننده ارزیابی می‌شود. این تفکیک‌کننده‌ها دارای یکی از معماری‌های GraphSAGE [۱۸]، DiffPool [۱۹]، و DGCNN [۲۰] هستند. همچنین، در تمامی آزمایش‌ها تفکیک‌کننده استفاده شده درون مدل دارای معماری GraphSAGE است. در تمام آزمایش‌ها پارامترهای $samples = ۱$ ، $steps = ۳$ ، $\gamma = ۰/۸$ ثابت است که این پارامترها با جستجوی شبکه‌ای^۴ در فضای ابرپارامترها پیدا شده‌اند.

نتایج آزمایش روی مجموعه داده Erdos Renyi

این سری آزمایش‌ها جزو اولین آزمایش‌های انجام شده توسط مدل هستند و هدف آن‌ها مطمئن شدن از درستی پیاده‌سازی مدل است. نتایج بدست آمده پس از ارزیابی گراف‌های تولید شده توسط مدل آموزش داده شده با دسته‌بند از نوع GraphSAGE را می‌توان در جدول ۴-۲ دید. همچنین، نمودار توزیع تخمین مقدار p در گراف‌های دو دسته را می‌توانید در شکل ۴-۲ ببینید. از آنجا که مقدار واقعی p برای گراف‌های دسته ۰ برابر ۰/۲۵ و برای گراف‌های دسته ۱ برابر ۰/۷۵ است، نتایج این آزمایش‌ها خوبی عملکرد مدل را نشان می‌دهند.

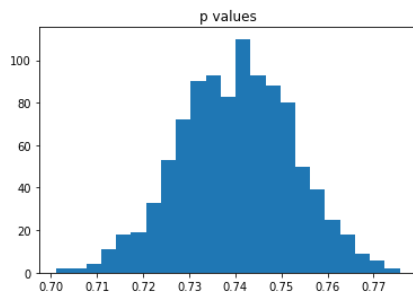
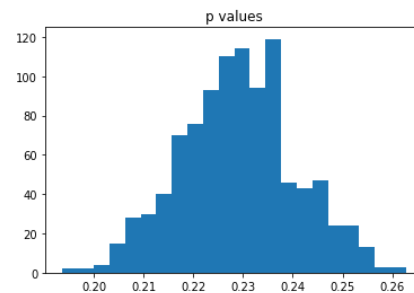
نتایج آزمایش روی مجموعه داده PROTEINS

در این سری آزمایش‌ها برخلاف دسته قبل، برای ارزیابی مدل پس از آموزش، گراف‌های تولید شده توسط مدل به وسیله سه دسته‌بند با معماری‌های GraphSAGE، DiffPool، و DGCNN ارزیابی شده و صحت عملکرد هر دسته‌بند برای هر

^۴Grid Search

جدول ۴-۲: نتایج مدل روی مجموعه داده Erdos Renyi.

مجموعه داده	صحت دسته‌بند روی نمونه‌های کلاس ۰	صحت دسته‌بند روی نمونه‌های کلاس ۱
گراف‌های واقعی	۱۰۰٪	۱۰۰٪
گراف‌های تولید شده	۹۹/۷٪	۱۰۰٪

(ب) توزیع تخمین مقادیر p در کلاس ۱.(آ) توزیع تخمین مقادیر p در کلاس ۰.شکل ۴-۲: توزیع تخمین مقادیر p در گراف‌های تولید شده در آزمایش مجموعه داده Erdos Renyi.

کلاس بدست می‌آید. همان‌طور که قبل‌تر بهش اشاره شد، هر چقدر صحت‌های بدست آمده این دسته‌بندها روی گراف‌های تولید شده به صحت آن‌ها روی گراف‌های واقعی نزدیک‌تر شود، می‌توان به این معنی برداشت کرد که مدل عملکرد بهتری دارد. علاوه بر ارزیابی توسط دسته‌بندها، برای گراف‌های تولید شده توسط مدل، آماره‌های پیش‌تر معرف شده محاسبه شده و مشابه صحت‌ها، با آماره‌های گراف‌های واقعی مقایسه می‌شوند. بار دیگر، هر چقدر این آماره‌ها به همدیگر نزدیک‌تر باشند، مدل عملکرد بهتری دارد.

در جدول ۴-۳ می‌توانید نتایج مربوط به بررسی صحت را روی گراف‌های تولید شده و گراف‌های واقعی ببینید. همچنین، در جدول ۴-۴ می‌توان آماره‌های گراف‌های واقعی و تولید شده رو بر حسب کلاس مشاهده کرد. اگر به نتایج صحت بدست آمده دقت کنیم، می‌بینیم که صحت در گراف‌های تولید شده و گراف‌های واقعی حدود ۵٪ با هم اختلاف دارند. این در حالی است که میانگین صحت دسته‌بندها روی گراف‌های واقعی حدود ۸۰٪ است. همچنین، اگر به آماره‌هایی همچون بزرگترین مولفه همبند و میانگین درجه رئوس در گراف‌های تولید شده و واقعی نگاه کنیم، باری دیگر نزدیکی زیادی بین دسته‌ها در دو گروه واقعی و تولید شده مشاهده می‌شود که بسیار نتیجه امیدوار کننده‌ای است.

اگر به نتایج دسته‌بندها توجه کنید، می‌توان دید که دسته‌بندهای از پیش آموزش داده شده مورد استفاده، همگی تمایلی به سمت دسته ۰ دارند. این به دلیل نسبت ۶۰ درصدی آن‌ها در مجموعه داده‌هاست که باعث می‌شود دسته‌بندها برای

افزایش صحت خود روی تمامی مجموعه داده‌ها ترجیح بدهند نوعی آموزش ببینند که دقت‌شان روی دسته بزرگتر بهتر شود

جدول ۴-۳: بررسی صحت گراف‌های تولید شده در آزمایش مجموعه داده PROTEINS.

مجموعه داده / دسته‌بند	صحت دسته‌بند روی نمونه‌های کلاس ۰	صحت دسته‌بند روی نمونه‌های کلاس ۱
گراف‌های واقعی / GraphSAGE	۸۲/۰٪	۶۵/۵٪
گراف‌های تولیدشده / GraphSAGE	۷۴/۹٪	۵۶/۱٪
گراف‌های واقعی / DGCNN	۸۱/۶٪	۶۳/۳٪
گراف‌های تولیدشده / DGCNN	۷۴/۸٪	۵۵/۷٪
گراف‌های واقعی / DiffPool	۹۲/۴٪	۴۲/۶٪
گراف‌های تولیدشده / DiffPool	۹۲/۵٪	۳۷/۶٪

جدول ۴-۴: بررسی صحت گراف‌های تولید شده در آزمایش مجموعه داده PROTEINS.

مجموعه داده / دسته	میانگین درجه رئوس	بزرگترین مولفه همبند	تعداد مثلث‌ها	Gini	طول مسیر مشخصه
گراف‌های واقعی / دسته ۰	۳/۷۹۸	۴۷/۶۷	۳۴/۳۰	۰/۰۵۲۲	۵/۶۰۵
گراف‌های واقعی / دسته ۱	۳/۶۴	۲۲/۲۷	۱۷/۲۴	-۰/۰۳۴۸	۳/۳۳
گراف‌های تولید شده / دسته ۰	۴/۱۰۴	۴۶/۳۲	۳۰/۹۲	۰/۱۱۷	۳/۱۷۷
گراف‌های تولید شده / دسته ۱	۳/۵۹۵	۲۳/۵۷۶	۱۶/۸۱۸	۰/۰۵۴۸	۲/۶۹۵۳

اگرچه منجر به باعث کاهش دقت در دسته کوچک‌تر می‌شود. برای بهبود این مشکل، هنگام آموزش دسته‌بندها، خطای دو دسته را ضریب‌دار کردیم اما همچنان این تمایل باقی می‌ماند.

باز فکر کنیم ببینیم چه جوری می‌تونیم مشکل رو حل کنیم
مثلا از هر دسته تعداد یکسانی داده بدیم بهش

فصل ۵

نتیجه‌گیری

در این پروژه به مسئله تولید مشروط گراف پرداختیم. ابتدا چندین راهکار مربوط به مسئله تولید و تولید مشروط در حوزه‌های گراف و متن را مورد بررسی قرار دادیم. سپس با بکارگیری ایده‌های موجود در روش‌های پیشین، نحوه‌ی تبدیل یک مدل مولد بدون شرط گراف به یک مدل مولد مشروط را بیان کرده و چالش‌های مربوط به مرحله آموزش و ارزیابی آنرا مطرح کرده و راه‌حل‌های پیشنهادی برای هر کدام را بررسی کردیم. در انتها، آزمایش‌هایی روی مجموعه داده‌های واقعی و ساختگی اجرا کرده و نتایج آن‌ها را گزارش کردیم که نشان‌دهنده کارایی خوب مدل ارائه شده بودند.

با وجود مطالب بیان شده، هنوز چالش‌ها و مشکلات زیادی در مسئله تولید مشروط گراف وجود دارد. به عنوان مثال، یکی از مشکلات مدل ارائه شده اتکای بسیار زیاد آن بر روی دسته‌بند از قبل آموزش دیده شده مورد استفاده است. به طور مثال، در آزمایش‌های مجموعه داده‌گان واقعی دیدیم که وجود جانبداری^۱ در دسته‌بند، موجب ایجاد عدم تعادل در مولد مشروط حاصل می‌شود. یکی دیگر از مشکلات مدل ارائه شده این است که در مرحله آموزش، از مدل انتظار می‌رود با وجود یک زیرگراف، چندین راس را طوری تولید کند که گراف تشکیل شده در دسته مناسب قرار بگیرد، در حالی که در هنگام ارزیابی، از مدل انتظار داریم بدون هیچ زیرگراف داده‌شده، یک گراف کامل در دسته مدنظر تولید کند. این مشکل، مشابه مشکل Exposure Bias است که برای حل آن می‌توان از به روش‌های مبنی بر مدل‌های مولد خصمانه روی آورد که این در کارهای آتی قرار می‌گیرد.

^۱ Bias

مراجع

- [1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.
- [2] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA, 2014. MIT Press.
- [3] L. Luo, W. Huang, Q. Zeng, Z. Nie, and X. Sun. Learning personalized end-to-end goal-oriented dialog. In *AAAI*, 2019.
- [4] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, and E. P. Xing. Toward controlled generation of text, 2017.
- [5] Y. Zhang, Z. Gan, and L. Carin. Generating text via adversarial training. 2016.
- [6] K. Mo, S. Li, Y. Zhang, J. Li, and Q. Yang. Personalizing a dialogue system with transfer reinforcement learning, 2016.
- [7] K. Wang and X. Wan. Sentigan: Generating sentimental texts via mixture adversarial networks. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4446–4452. International Joint Conferences on Artificial Intelligence Organization, 7 2018.
- [8] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014.
- [9] I. V. Serban, A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. Courville, and Y. Bengio. A hierarchical latent variable encoder-decoder model for generating dialogues. In *Pro-*

- ceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 3295–3301. AAAI Press, 2017.
- [10] W. Wang, Z. Gan, H. Xu, R. Zhang, G. Wang, D. Shen, C. Chen, and L. Carin. Topic-guided variational auto-encoder for text generation. *Proceedings of the 2019 Conference of the North*, 2019.
 - [11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805, 2019.
 - [12] A. Radford. Improving language understanding by generative pre-training. 2018.
 - [13] I. Higgins, L. Matthey, X. Glorot, A. Pal, B. Uria, C. Blundell, S. Mohamed, and A. Lerchner. Early visual concept learning with unsupervised deep learning, 2016.
 - [14] P. Bachman and D. Precup. Data generation as sequential decision making, 2015.
 - [15] J. You, R. Ying, X. Ren, W. L. Hamilton, and J. Leskovec. Graphrnn: Generating realistic graphs with deep auto-regressive models, 2018.
 - [16] R. Liao, Y. Li, Y. Song, S. Wang, C. Nash, W. L. Hamilton, D. Duvenaud, R. Urtasun, and R. S. Zemel. Efficient graph generation with graph recurrent attention networks, 2020.
 - [17] A. Z. Pervaiz, S. Datta, and K. Y. Camsari. Probabilistic computing with binary stochastic neurons. In *2019 IEEE BiCMOS and Compound semiconductor Integrated Circuits and Technology Symposium (BCICTS)*, pages 1–6, 2019.
 - [18] W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs, 2018.
 - [19] R. Ying, J. You, C. Morris, X. Ren, W. L. Hamilton, and J. Leskovec. Hierarchical graph representation learning with differentiable pooling, 2019.
 - [20] M. Zhang, Z. Cui, M. Neumann, and Y. Chen. An end-to-end deep learning architecture for graph classification. In *AAAI*, 2018.

واژه‌نامه

generalization	تعمیم‌پذیری	الف	
fine-tuning	تنظیم دقیق	hyperparameter	ابریارامتر
loss function	تابع خطا	infer	استنباط
content generation	تولید محتوا	statistic	آماره
generalization	تعمیم‌پذیری	stable	استوار
discriminator	تفکیک‌کننده	message passing	انتقال پیام
reinforcement	تقویتی		
sequential	ترتیبی	ب	
attention	توجه	loading	بارگذاری
mixture	ترکیب	label	برچسب
bias	نمایل	maximum	بیشینه
		reproduce	بازتولید
		recurrent	بازگشتی
ج			
brute-force	جست‌وجوی جامع		
breadth first search	جستجوی اول سطح	پ	
grid search	جستجوی شبکه‌ای	backpropagation	پس‌انتشار
چ		ت	
framework	چارچوب		

ص

accuracy صحت

ق

transferable قابل انتقال

strong قوی

interpretable قابل تفسیر

ک

minimum کمینه

گ

stride گام

م

parallel موازی

deep models مدل‌های عمیق

generative مولد

latent variable متغیرهای نهفته

adjacency matrix ماتریس مجاورت

parameter مشخصه

characteristic path مسیر مشخصه

ن

representation نمایش، نماینده

ه

ح

action حرکت

خ

autoencoder خودکدگذار

adversarial خصمانه

د

dataset داده‌ها، دادگان

precision دقت

binary دودویی

classifier دسته‌بند

likelihood درست‌نمایی

ر

gradient descent method روش کاهش گرادیان

node راس

encoder کدگذار

decoder کدگشا

س

fake ساختگی

ش

graph neural networks شبکه‌های عصبی گراف

supervised learning	یادگیری نظارتی	connected	همبند
edge	یال		
one hot	یک‌گرم		ی

Abstract

In recent years, deep generative models have achieved considerable success in graph generation. However, past work has primarily focused on unrestricted generation, which is in contrast with many advances in conditional generation in text and images. Due to the fact that graphs become more meaningful when associated with a context, the question is whether we can transfer the progress from text and images to the domain of graphs. In this project, we utilize several ideas from the domain of images and text and tackle the challenge of conditional graph generation. We propose a supervised approach to training deep generative models for conditional graph generation and evaluate the model on several different network datasets.

Keywords: Deep Learning, Graph Generation, Conditional Graph Generation



Sharif University of Technology

Department of Computer Engineering

B.Sc. Thesis

Conditional Graph Generation

By:

Amir Mojtaba Sabour

Supervisor:

Dr. Mahdiah Soleymani

January 2021