دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)
دانشکده مهندسی کامپیوتر

سمینار کارشناسی ارشد

گرایش هوش مصنوعی و رباتیکز

آشنایی با مدل‌های پخشی

نگارش

عطیه غفارلوی مقدم

استاد درس

دکتر رضا صفابخش

۱۴۰۲/۶/۱۰


بسم الله الرحمن الرحيم

چکیده

مدل‌های پخشی به عنوان یک خانواده‌ی قدرتمند از مدل‌های مولد با نتایج قابل توجه در حوزه‌های مختلف تصویر، صوت، متن و گراف، در سال‌های اخیر مورد توجه بسیاری از محققین یادگیری ماشین قرار گرفته‌اند. همه‌ی مدل‌های مطرح شده در این خانواده از یک رویکرد مشابه پیروی می‌کنند و آن هم تخریب داده با نویز در یک فرآیند گام به گام و سپس بازگرداندن این فرآیند برای تولید نمونه‌های جدید از داده می‌باشد. در این گزارش ابتدا به بررسی دقیق سه دسته‌ی مطرح مدل‌های پخشی و شباهت‌ها و تفاوت‌های آنها می‌پردازیم. پس از معرفی این مدل‌ها، به بررسی رابطه ی میان آنها و انواعی دیگر از مدل‌های مولد می‌پردازیم و در نهایت به تعدادی از چالش‌های مطرح در خصوص این مدل‌ها اشاره می‌کنیم.

واژه‌های کلیدی:

مدل‌های مولد، مدل‌های پخشی، مدل‌های مولد مبتنی بر امتیاز، مدل‌های پخشی نویززدای احتمالاتی

فهرست مطالب

عنوان صفحه

فهرست تصاویر

شکل صفحه

فهرست جداول

جدول صفحه

فهرست اختصارات

عنوان اختصاری	عنوان کامل
مپنا	مدل پخشی نویززدای احتمالاتی

فصل اول

مقدمه

مدل‌های مولد کاربردهای متنوعی در یادگیری ماشین دارند، از جمله‌ی این کاربردها می‌توان به تولید عکس‌های باکیفیت، تولید قطعه‌های موسیقی و یا صوت، افزایش کارایی یادگیری نیمه نظارتی[۱]، تشخیص ناهنجاری[۲]، یادگیری تقلیدی و کمک به یادگیری تقویتی اشاره کرد. پیشرفت‌های اخیر این حوزه دو رویکرد کلی را دنبال می‌کنند: رویکردهای مبتنی بر درست‌نمایی[۳] و شبکه‌های مولد تقابلی[۴]. در رویکردهای مبتنی بر درست‌نمایی از لگاریتم درست‌نمایی یا خطاهای جایگزین به عنوان تابع هدف یادگیری استفاده می‌شود؛ در حالیکه در شبکه‌های مولد تقابلی تلاش می‌شود تا یادگیری با یک رویکرد تقابلی بین دو شبکه‌ی تمایزگر[۵] و تولیدکننده[۶] صورت گیرد.

با وجود عملکرد قابل توجه مدل‌های مبتنی بر درست‌نمایی وشبکه‌های مولد تقابلی، این مدل‌ها به واسطه‌ی نوع تعریفشان درگیر برخی محدودیت‌های ذاتی هستند. به عنوان مثال مدل‌های مبتنی بر درست‌نمایی برای ساخت توزیع‌های نرمال شده نیازمند به کارگیری معماری‌های خاص یا استفاده از توابع خطای جایگزین (مانند حد پایین مشاهده[۷] در خودکدگذارهای تغییراتی[۸]) در فرآیند یادگیری هستند. شبکه‌های مولد تقابلی مشکلات موجود در مدل‌های مبتنی بر درست‌نمایی را ندارند اما فرآیند یادگیری آنها به خاطر ماهیت تقابلی آن می‌تواند دچار ناپایداری شود. همچنین تابع هدف مدل‌های مولد تقابلی برای مقایسه و ارزیابی انواع این مدل‌ها چندان مناسب نیست[۱۴].

مدل‌های پخشی[۹] یک دسته‌ی نوظهور از مدل‌های مولد هستند که با توجه به عملکرد مناسب آنها در حوزه‌های تصویر، صوت، متن و گراف به سرعت مورد استقبال قرار گرفتند. وجه مشترک همه‌ی مدل‌های پخشی این است که در یک فرآیند رو به جلو[۱۰] داده را با نویز تخریب می‌کنند ودر نتیجه از توزیع داده به یک توزیع دلخواه قابل محاسبه می‌رسند و سپس در یک مسیر رو به عقب که توسط مدل یاد گرفته می‌شود، تلاش می‌کنند تا داده را بازسازی و از یک نمونه از توزیع دلخواه به یک نمونه از توزیع داده‌ها برسند. این موضوع درشکل ۱-۱ نشان داده شده است.

با وجود اینکه چهارچوب گفته شده در بند قبل در همه‌ی مدل‌های پخشی وجود دارد؛ رویکردهای مختلفی برای فرموله سازی و پیاده سازی این روش وجود دارد. در مدل‌های مولد مبتنی بر تابع امتیاز[۱۱] تلاش می‌شود با استفاده از گرادیان لگاریتم توزیع داده، که به آن تابع امتیاز[۱۲] می‌گوییم، و یک فرآیند

[1] Semi-supervised learning
[2] Anomaly
[3] Likelihood
[4] Generative adversarial networks
[5] Discriminatior
[6] Generator
[7] Evidence lower bound
[8] Variational autoencoders
[9] Diffusion models
[10] Forward process
[11] Score-based generative models
[12] Score function

شکل ۱-۱: روند کلی یک مدل پخشی

در هر مدل پخشی یک فرآیند ثابت گام به گام رو به جلو برای تخریب داده و یک فرآیند معکوس برای تولید یک نمونه داده از نویز وجود دارد[۹].

گام به گام حرکت در جهت این گرادیان نمونه‌ی جدید تولید کنیم. از طرفی در مدل‌های پخشی نویززدای احتمالاتی فرآیند تخریب داده و تولید نمونه با یک زنجیره‌ی مارکفی احتمالاتی مدل می‌شود و در نهایت در مدل‌های پخشی مبتنی بر معادلات دیفرانسیل تصادفی، فرآیند تخریب به صورت یک معادله دیفرانسیل تصادفی پیوسته مدل می‌شود که با پیدا کردن معکوس این معادله و گسسته سازی آن امکان تولید نمونه‌ی جدید از توزیع داده‌ها فراهم می‌گردد.

می‌توان نشان داد که مدل‌های پخشی بسیاری از محدودیت‌های مدل‌های مولد دیگر مانند نیاز به تمایزگر در شبکه‌های مولد تقابلی، نیاز به مقیدسازی شبکه در جریان‌های نرمال‌ساز[۱۳] و نیاز به تنظیم توزیع‌های پسین در خودکدگذارهای تغییراتی را ندارند و قابلیت تولید تصاویر باکیفیت و متنوع را دارند. مشکل اصلی در مدل‌های پخشی فرآیند نمونه‌برداری طولانی آنهاست که باعث عدم کارایی این مدل‌ها در کاربردهای تعاملی می‌شود. با رفع این موضوع می‌توان اطمینان حاصل کرد که این دسته از مدل‌های مولد از توانایی همه‌جانبه‌ای برای به کارگیری در حوزه‌های مختلف برخوردار هستند.

در این گزارش ابتدا هر یک از سه مدل پخشی و فرموله سازی هر کدام را بیان می‌کنیم و نشان می‌دهیم که تحت شرایطی خاص این روش‌ها هم ارز و قابل تبدیل به یکدیگر هستند. سپس در فصل پنجم به مقایسه‌ی کلی روش‌های پخشی با سایر مدل‌های مولد و همچنین مقایسه‌ی انواع مدل‌های پخشی مطالعه شده در این گزارش با یکدیگر می‌پردازیم و در نهایت به برخی جنبه‌هایی از مدل‌های پخشی می‌پردازیم که قابلیت مطالعه‌ی بیشتر و اعمال نوآوری را دارند.

[13] Normalizing flows

فصل دوم

مدل‌های مولد مبتنی بر امتیاز

در این فصل به معرفی اولین دسته از مدل‌های پخشی می‌پردازیم. این مدل‌ها برای تولید نمونه از گرادیان توزیع نسبت به داده، که تابع امتیاز نام دارد، کمک می‌گیرند. اما از آنجایی که تابع امتیاز در اکثر داده‌های دنیای واقعی خوش تعریف نیست، برای تولید نمونه‌ها نیازمند سازوکاری هستیم که بتواند ما را به تابع امتیاز مناسب‌تری برساند. در این فصل ابتدا مفاهیم اولیه‌ی به کارگیری تابع امتیاز برای تولید نمونه‌ی جدید را بررسی می‌کنیم و سپس مشکلات این روش و راه‌حل‌های پیشنهادی ارائه شده را مطرح می‌کنیم و در نهایت تکنیک‌هایی برای افزایش مقیاس پذیری این مدل معرفی می‌کنیم.

۲-۱ مفاهیم اولیه

۲-۱-۱ تابع امتیاز و فرآیند تطبیق امتیاز

فرض کنید مجموعه داده‌ی ما از تعدادی نمونه‌ی مستقل و با توزیع یکسان از توزیع ناشناخته‌ی $p_{data}(x)$ تشکیل شده است. امتیاز یک توزیع احتمالاتی $p(x)$ را گرادیان لگاریتم آن توزیع نسبت به داده‌ها تعریف می‌کنیم که به فرم $\nabla_x \log p(x)$ نشان داده می‌شود. امتیاز یک توزیع احتمالاتی در واقع یک میدان برداری[1] است که به سمتی اشاره می‌کند که درست‌نمایی داده‌ها بیشترین رشد را دارد. شکل ۲-۱ این میدان برداری را برای یک توزیع یک بعدی نشان می‌دهد. با داشتن تابع امتیاز یک توزیع می‌توانیم از آن توزیع نمونه گیری کنیم و خود توزیع را تخمین بزنیم.

شبکه‌ی امتیاز $s_\theta : \mathbb{R}^D \to \mathbb{R}^D$ یک شبکه‌ی عصبی با پارامترهای θ می‌باشد که برای تخمین تابع امتیاز یک توزیع مانند $p_{data}(x)$ آموزش داده می‌شود. در صورتی که بتوانیم تابع امتیاز توزیع را بدست آوریم می‌توانیم از این تابع برای مدل مولد و همچنین تولید نمونه‌های جدید استفاده کنیم. فرآیند تطبیق امتیاز[2] [۷] فرآیندی است که برای آموزش شبکه‌ی امتیاز به کار گرفته می‌شود و هدف اصلی آن تخمین $\nabla_x \log p(x)$ بدون نیاز به تخمین مستقیم $p(x)$ است. تابع هدف تخمین امتیاز به صورت زیر قابل تعریف است:

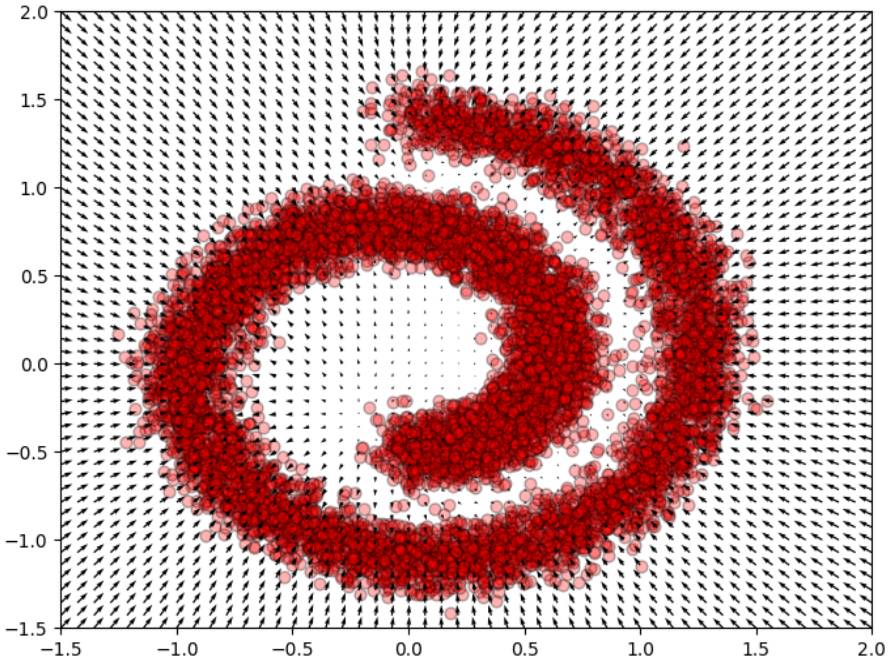
$$\frac{1}{2}\mathbb{E}_{p_{data}}[||s_\theta(x) - \nabla_x \log p_{data}(x)||_2^2] \qquad (۱-۲)$$

که می‌توان به صورت ریاضی نشان داد که با رابطه‌ی زیر هم‌ارز است:

$$\mathbb{E}_{p_{data}(x)}[tr(\nabla_x s_\theta(x)) + \frac{1}{2}\|s_\theta(x)\|_2^2] \qquad (۲-۲)$$

[1] Vector field

[2] Score matching



شکل ۲-۱: میدان برداری تابع امتیاز برای یک توزیع یک بعدی

معادله‌ی بالا به دلیل محاسبه ی ژاکوبین تابع امتیاز قابلیت مقیاس‌پذیری و پیاده‌سازی در داده‌ی با بعد بالا را ندارد بنابراین باید به دنبال جایگزینی برای تابع هدف باشیم به گونه‌ای که بتوانیم در مقیاس بزرگتر و ابعاد بالاتر نیز از رویکرد تطبیق امتیاز برای تخمین تابع امتیاز استفاده کنیم.

۲-۱-۲ تطبیق امتیاز نویززدا

تطبیق امتیاز نویززدا[3] [۱۷] یک رویکرد تطبیق امتیاز است که به هدف رفع مشکل محاسبه ی ژاکوبین توسعه داده شد. در این فرآیند ابتدا نقطه‌ی x با توزیع نویز از پیش تعریف شده‌ی $q_\sigma(\tilde{x}|x)$ تخریب می‌شود و سپس تخمین امتیاز برای توزیع داده‌ی تخریب شده‌ی $q_\sigma(\tilde{x}) \triangleq \int_{q_\sigma}(\tilde{x}|x)p_{data}(x)dx$ صورت می‌گیرد. در این حالت تابع هدف به صورت زیر می‌باشد:

$$\frac{1}{2}\mathbb{E}_{q_\sigma(\tilde{x}|x)p_{data}(x)}\left[\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}}\log q_\sigma(\tilde{x}|x)\|_2^2\right] \qquad (۳-۲)$$

که با کمینه کردن آن تابع بهینه‌ی $s_{\theta^*}(x)$ بدست می‌آید. حال می‌توان نشان داد که اگر نویز به اندازه‌ی کافی کوچک باشد، آنگاه $s_{\theta^*}(x)$ تخمینی از تابع امتیاز توزیع داده است.

[3] Denoising score matching

۲-۱-۳ دینامیک لانژوین

دینامیک لانژوین[۴] برای تولید نمونه از توزیع ناشناخته ی $p(x)$ با استفاده از تابع امتیاز این توزیع به کار می‌رود. با داشتن یک اندازه ی گام $\epsilon > 0$ و یک مقدار اولیه‌ی $\tilde{x}_0 \sim \pi(x)$ که π یک توزیع اولیه است. دینامیک لانژوین به صورت تکراری فرآیند زیر را تکرار می‌کند:

$$\tilde{x}_t = \tilde{x}_{t-1} + \frac{\epsilon}{2}\nabla_x \log p(\tilde{x}_{t-1}) + \sqrt{\epsilon} z_t \qquad (۴-۲)$$

که $z_t \sim \mathcal{N}(0, I)$ است. توزیع $\tilde{x}_T$ با توزیع $p(x)$ برابر می‌شود اگر $\epsilon \to 0$ و $T \to \infty$ باشد. با توجه به معادله‌ی (۴-۲) برای بدست آوردن نمونه‌ی جدید تنها به تابع امتیاز توزیع نیاز داریم از این رو می‌توانیم ابتدا یک شبکه‌ی امتیاز برای تخمین امتیاز در هر نقطه به دست آوریم و سپس با دینامیک لانژوین و با استفاده از تخمین‌های امتیازات به یک نمونه‌ی جدید از توزیع برسیم.

۲-۲ تولید نمونه با روش تطبیق امتیاز و دینامیک لانژوین

همان طور که پیش‌تر هم اشاره شد، با داشتن تخمین تابع امتیاز به کمک تطبیق امتیاز و دینامیک لانژوین می‌توان به یک نمونه‌ی جدید رسید. اما در ادامه نشان خواهیم داد که دومانع اساسی برای پیاده‌سازی این فرم از تولید نمونه‌ها وجود دارد که ما را ملزم می‌سازد تا روش پیشنهادی را بهبود دهیم.

۲-۲-۱ فرضیه‌ی رویه‌ها

فرضیه‌ی رویه‌ها بیان می‌کند که اکثر داده‌های دنیای واقعی در یک فضای با بعد بالا روی یک رویه‌ی با بعد کمتر قرار دارند. به صورت تجربی نشان داده شده است که این موضوع برای بسیاری از مجموعه داده‌ها برقرار است. در این صورت روش‌های مولد مبتنی بر امتیاز با دو مشکل اساسی روبرو هستند. مشکل اول این است که تابع امتیاز گرادیان گرفته شده در فضای با بعد بالاتر است و زمانی که x محدود به یک رویه‌ی با بعد پایین‌تر است این گرادیان تعریف نشده است. و دوم، تابع هدف فرآیند تطبیق امتیاز تنها در صورتی تخمین سازگاری از امتیاز می‌دهد که فضای پشتیبان[۵] توزیع داده‌ها با کل فضای تعریف شده یکی باشد که در حالتی که داده‌ها روی رویه‌ی با بعد کمتر هستند چنین نیست.

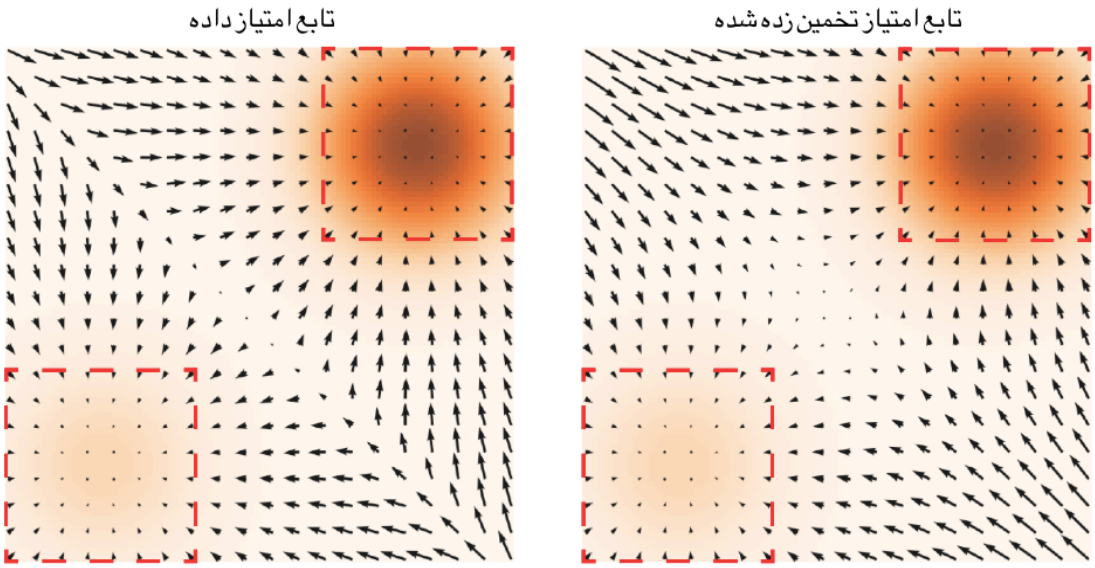
۲-۲-۲ نواحی کمتر چگال

پراکندگی داده در نواحی کمتر چگال هم برای تخمین تابع امتیاز و هم برای دینامیک لانژوین مشکل ایجاد می‌کند. در تخمین تابع امتیاز در نواحی که نمونه‌های کافی از داده‌ها وجود ندارد، به دلیل کم

[4] Langevin synamics

[5] Support

بودن شواهد توزیع داده، خطای زیادی وجود دارد. برای توضیح این موضوع از یک مثال توزیع گاوسی مخلوط استفاده می‌کنیم. همانطور که در شکل ۲-۲ مشاهده می‌شود، در نواحی بین دو قله ی گاوسی تخمین تابع امتیاز تخمین قابل اتکایی نیست.



شکل ۲-۲: مشکل نواحی کمتر چگال در تخمین تابع امتیاز

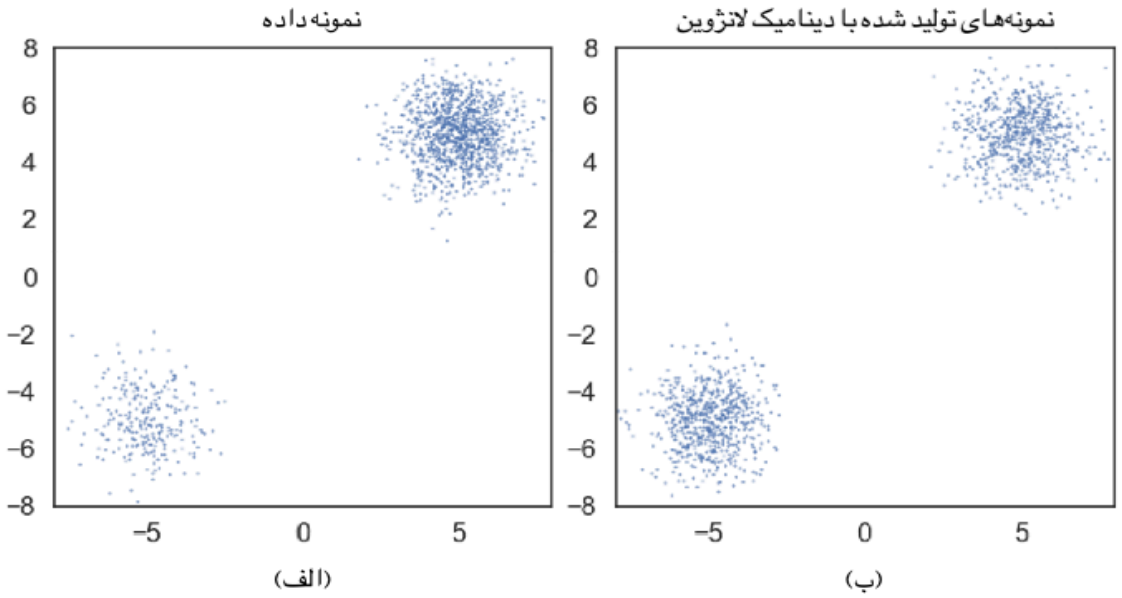
شکل سمت چپ جهت‌های میدان برداری امتیاز را در یک توزیع ترکیب گاوسی نشان می‌دهد و شکل سمت راست تخمین به دست آمده از تابع امتیاز را نشان می‌دهد. همانطور که مشاهده می‌کنید تخمین امتیاز در بین دو قله‌ی گاوسی که ناحیه‌ی کمتر چگال است با خطا همراه است[۱۴].

همچنین زمانی که دو قله‌ی توزیع یا دو ناحیه‌ی چگال داده با یک ناحیه‌ی کمتر چگال از یکدیگر جدا شده باشند، دینامیک لانژوین قابلیت درک درست وزن نسبی قله‌ها نسبت به یکدیگر را از دست می‌دهد و در واقع قدرت تمایز میان نواحی چگال تر را از دست می‌دهد و در نتیجه به توزیع درست همگرا نمی‌شود. همچنین در صورتی که دو مد داده فضای پشتیبان مجزایی داشته باشند نیز دینامیک لانژوین برای تولید نمونه‌های درست نیازمند گام‌های کوچکتر و در نتیجه زمان ترکیب[۶] زیادی می‌باشد. شکل ۲-۳ این موضوع را نشان می‌دهد.

۲-۲-۳ راه‌حل پیشنهادی

می‌توان مشاهده کرد که تخریب داده با نویز گاوسی تصادفی می‌تواند مجموعه داده‌ی مناسب تری برای اعمال روش‌های مبتنی بر امتیاز بسازد. زیرا اولا اضافه کردن نویز گاوسی محدودیت داده به یک رویه‌ی

[6] Mixing



شکل ۲-۳: مشکل نواحی کمتر چگال در دینامیک لانژوین

شکل (الف) نمونه‌های یک توزیع مخلوط گاوسی را نشان می‌دهد که از دو قله با وزن های متفاوت ایجاد شده است. شکل(ب) نمونه‌های تولید شده با دینامیک لانژوین از همین توزیع را نشان می‌دهد. همانطور که مشاهده می‌شود، دینامیک لانژوین از هر دو قله به یک نسبت نمونه تولید کرده است و قادر به درک وزن نسبی قله‌ها نسبت به یکدیگر نیست[۱۴].

با بعد پایین‌تر را از بین می‌برد چراکه فضای پشتیبان یک توزیع گاوسی کل فضای با بعد بالاست و همین امر سبب می‌شود که مشکلات ایجاد شده به واسطه‌ی فرضیه‌ی رویه‌ها برطرف شود. و دوم اینکه نویز گاوسی بزرگ امکان پرکردن ناحیه‌های کمتر چگال در توزیع داده‌ی تخریب نشده را دارد و همین امر می‌تواند مشکل تخمین امتیاز در نواحی کمتر چگال و مشکلات دینامیک لانژوین را برطرف کند. همچنین اضافه کردن نویز در چندین مرحله تا همگرا شدن به توزیع اصلی داده و بعد اعمال دینامیک لانژوین روی هر یک از این توزیع‌های میانی می‌تواند سرعت همگرایی دینامیک لانژوین را افزایش دهد. با توجه به فرضیه‌ی مطرح شده در بند قبل می‌توانیم فرآیند تولید نمونه‌ی مبتنی بر امتیاز را به دو صورت بهبود دهیم: اول، داده را با سطوح مختلف نویز تخریب کنیم و دوم، از یک شبکه‌ی شرطی شده بر روی نویز برای تخمین امتیاز در همه‌ی سطوح نویز استفاده کنیم. سپس از دینامیک لانژوین برای تولید نمونه‌ها از سطوح نویز بزرگ به سطوح نویز کوچک استفاده کنیم. در ادامه جزئیات این روش را بیشتر توضیح خواهیم داد.

۲-۲-۴ شبکه‌ی امتیاز شرطی شده با نویز

مجموعه‌ی $\{\sigma_i\}_{i=1}^{L}$ را یک دنباله‌ی هندسی مثبت در نظر بگیرید که $\frac{\sigma_1}{\sigma_2} = \cdots = \frac{\sigma_{L-1}}{\sigma_L} > 1$ و توزیع داده‌ی تخریب شده را نیز به صورت $q_\sigma(x) \triangleq \int p_{data}(t)\mathcal{N}(x|t,\sigma^2 I)dt$ تعریف می‌کنیم. مجموعه‌ی $\{\sigma_i\}$ را به گونه‌ای تعریف می‌کنیم که σ_1 به اندازه ی کافی بزرگ باشد تا بتوانیم مشکلات مطرح

شده در بخش قبل را برطرف کنیم و σ_L را به اندازه‌ای کوچک تعریف می‌کنیم که توزیع تخریب شده تقریبا با توزیع داده ها برابر باشد. هدف آموزش شبکه‌ی امتیاز شرطی شده با نویز این است که تابع امتیاز تمامی توزیع‌های تخریب شده را به گونه‌ای به دست آورد که داشته باشیم: $\forall \sigma \in \{\sigma_i\}_{i=1}^{L}$: $s_\theta(x,\sigma) \approx \nabla_x \log q_\sigma(x)$. به عبارت $s_\theta(x,\sigma)$ شبکه‌ی امتیاز شرطی شده با نویز می‌گوییم. در [۱۴] شبکه‌ی به کارگرفته شده برای معماری شبکه‌ی امتیاز شرطی شده با نویز یک شبکه‌ی یو[۷] [۱۲] می‌باشد که یک شبکه ی بسیار موفق در امر قطعه بندی معنایی[۸] تصویر است.

۲-۲-۵ به کارگیری تطبیق امتیاز برای آموزش شبکه‌ی امتیاز شرطی شده با نویز

برای آموزش شبکه‌ی امتیاز، از تابع هدف مشابه تطبیق امتیاز نویز زدا استفاده شده است و با توجه به اینکه $q_\sigma(\tilde{x}|x) = \mathcal{N}(\tilde{x}|x,\sigma^2 I)$ است و در نتیجه داریم $\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x) = -(\tilde{x}-x)/\sigma^2$ و بنابراین برای یک σی مشخص داریم:

$$l(\theta;\sigma) \triangleq \frac{1}{2}\mathbb{E}_{p_{data}(x)}\mathbb{E}_{\tilde{x}\sim\mathcal{N}(x,\sigma^2 I)}\left[\left\| s_\theta(\tilde{x},\sigma) + \frac{\tilde{x}-x}{\sigma^2}\right\|_2^2\right] \tag{۲-۵}$$

و از ترکیب معادله‌ی بالا برای همه ی سطوح نویز داریم:

$$\mathcal{L}(\theta;\{\sigma_i\}_{i=1}^{L}) \triangleq \frac{1}{L}\sum_{i=1}^{L}\lambda(\sigma_i) l(\theta;\sigma_i) \tag{۲-۶}$$

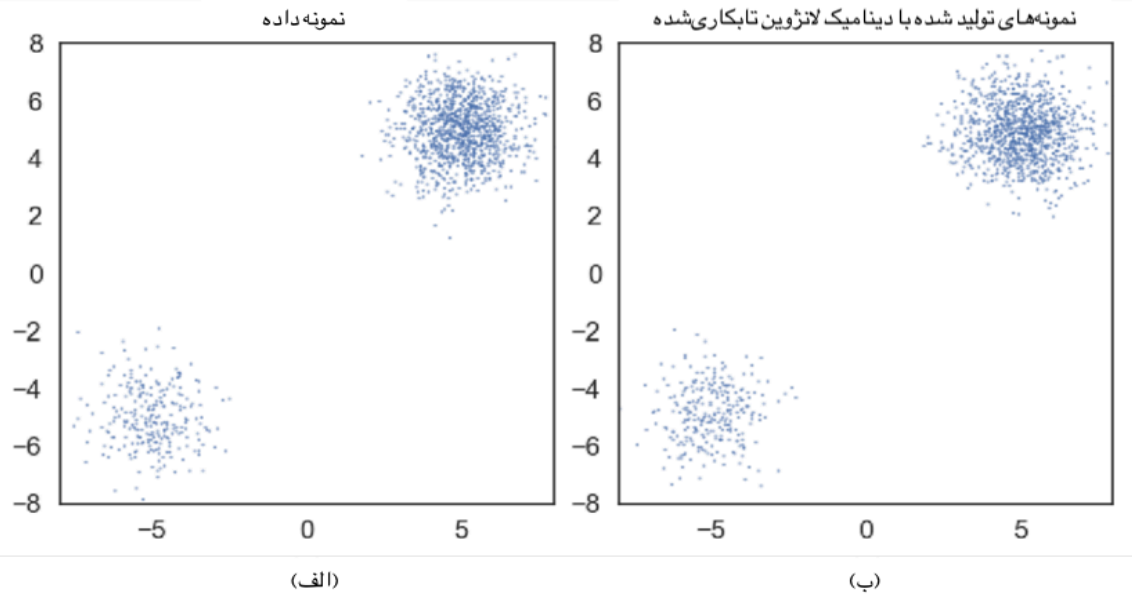
که $\lambda(\sigma_i)$ یک تابع ضریب از سطوح نویز است. تابع هدف تعریف شده برای این مساله به هیچ گونه یادگیری تقابلی[۹] و هیچ گونه نمونه برداری از تابع در فرآیند یادگیری نیاز ندارد و همین امر سبب برتری آن نسبت به سایر شبکه‌های مولد می‌شود.

۲-۲-۶ نمونه گیری از شبکه‌ی امتیاز شرطی شده با نویز

پس از آموزش شبکه‌ی امتیاز شرطی شده با نویز از یک دینامیک لانژوین تابکاری شده[۱۰] برای تولید نمونه‌ها استفاده می‌کنیم. که ایده‌ی اصلی آن از تابکاری شبیه سازی شده[۱۱] گرفته شده است. فرآیند این روش به این صورت است که با شروع از یک توزیع اولیه، دینامیک لانژوین را روی نمونه‌های این توزیع

[7] U-net
[8] Semantic segmentation
[9] Adversarial
[10] Annealed langevin dynamics
[11] Simulated annealing

شروع می‌کنیم و سپس آن را اجرا می‌کنیم تا از توزیع $q_{\sigma_1}(x)$ نمونه گیری کنیم. سپس نمونه‌های تولید شده را به عنوان نقطه‌ی شروع نمونه برداری با دینامیک لانژوین در توزیع $q_{\sigma_2}(x)$ در نظر می‌گیریم و در هر گام اندازه‌ی حرکت را کاهش می‌دهیم تا زمانی که دینامیک لانژوین را برای نمونه گیری از $q_{\sigma_L}(x)$ انجام دهیم که بسیار نزدیک به توزیع اصلی داده است و می‌تواند نمونه‌هایی از داده را تولید کند. نتایج رویکرد ارائه شده برای یک توزیع مخلوط گاوسی دو بعدی در شکل ۲-۴ آمده است که نشان می‌دهد که برخلاف دینامیک لانژوین این روش به خوبی امکان تشخیص نسبت وزنی قله‌ها به یکدیگر را دارد.



شکل ۲-۴: حل مشکل نواحی کمتر چگال در دینامیک لانژوین تابکاری‌شده

شکل (الف) نمونه‌های یک توزیع مخلوط گاوسی را نشان می‌دهد که از دو قله با وزن های متفاوت ایجاد شده است. شکل(ب) نمونه‌های تولید شده با دینامیک لانژوین تابکاری شده از همین توزیع را نشان می‌دهد. همانطور که مشاهده می‌شود، دینامیک لانژوین تابکاری شده برخلاف دینامیک لانژوین قابلیت درک وزن نسبی قله‌ها را دارد[۱۴].

۲-۳ بهبود مدل‌های مبتنی بر امتیاز

در بخش‌های قبل به معرفی و شرح جزئیات مدل‌های مبتنی بر امتیاز پرداختیم. این مدل‌ها دو مزیت اصلی نسبت به سایرین دارند که عبارتست از:

- درفرآیند آموزش نیازی به نمونه گیری نداریم و همین موضوع سبب می‌شود که برای آموزش شبکه‌های عمیق، کاراتر و مقیاس پذیرتر باشند.

- فرآیند نمونه برداری و آموزش کاملا جدا از یکدیگر هستند و این امر امکان خلاقیت و استفاده از سایر روش‌های نمونه برداری مبتنی بر امتیاز را می‌دهد.

اما با وجود این مزایا و علیرغم عملکرد مناسب این روش‌ها، مدل‌های مبتنی بر امتیاز معرفی شده تنها روی تصاویر با رزولوشن پایین عملکرد مناسبی دارند و در رزولوشن‌های بالاتر این عملکرد با عوامل محدودکننده‌ای روبرو می‌شود. از جمله‌ی این عوامل می‌توان به این موضوع اشاره کرد که برای یادگیری شبکه‌ی امتیاز شرطی شده با نویز از روش تطبیق امتیاز نویززدا استفاده می‌شود و در نتیجه در این فرآیند نیازمند تخریب داده با سطوح نویز مختلف هستیم و این سطوح نویز باید به گونه‌ای باشند که شبکه‌ی امتیاز هم کلیات و هم جزئیات تصویر را بیاموزد. از این رو انتخاب سطوح نویز بسیار اهمیت دارد. در پیاده سازی توصیف شده در قسمت‌های قبل، این عملکرد برای تصاویر با رزولوشن پایین مناسب بود اما هیچ تکنیک یا ترفندی برای رزولوشن‌های بالاتر ارائه نشد. در ادامه دو تکنیک موثر و با پشتوانه‌ی تئوری را مختصرا توضیح می‌دهیم که مدل‌های مولد مبتنی بر امتیاز را توانمند می‌سازد تا تصاویری با رزولوشن بالاتر تولید کنند.

۲-۳-۱ انتخاب دنباله‌ی نویز

انتخاب دنباله‌ی نویز برای موفقیت شبکه‌های امتیاز شرطی شده با نویز بسیار مهم است. اگرشبکه تنها با یک نویز آموزش داده شود و یا نویزها بسیار کوچک باشند، تصاویر تولید شده کیفیت مناسبی ندارند چراکه تخمین تابع امتیاز به درستی صورت نمی‌گیرد واگر نویزها خیلی بزرگ باشند نمونه‌های تولیدی خرابی دارند. برای تعیین دنباله ی نویز $\{\sigma_i\}_{i=1}^{L}$ باید دو پارامتر نویز اولیه یا σ_1 و L یا طول دنباله ی نویز را تعیین کنیم. پارامتر نویز اولیه تنوع داده‌ها را کنترل می‌کند و هر چقدر که بزرگتر باشد تنوع بیشتری را تضمین می‌کند و از طرفی بزرگتر شدن σ_1 به مقادیر نویز کاهش یافته‌ی بیشتری نیاز دارد و این امر دینامیک لانژوین را پرهزینه‌تر می‌کند. به طور تئوری می‌توان نشان داد که تعیین σ_1 باید به گونه ای باشد که از نظر عددی قابل مقایسه با حداکثر فاصله‌ی جفت داده‌ی موجود در مجموعه داده باشد[۱۵]. انتخاب دو نویز متوالی σ_i و σ_{i-1} باید به گونه ای باشد که $p_{\sigma_i}(x)$ نمونه‌های کافی در ناحیه‌های چگال $p_{\sigma_{i-1}}(x)$ داشته باشد. این امر اصل اساسی توسعه‌ی دینامیک لانژوین تابکاری شده است. می‌توان به صورت تئوری نشان داد که برای اینکه اصل بالا برقرار باشد باید یک دنباله‌ی هندسی برای ساخت نویز در نظر بگیریم به صورتی که نسبت $\lambda = \frac{\sigma_{i-1}}{\sigma_i}$ آن طوری تعیین شود که رابطه‌ی زیر را ارضا کند:

$$\Phi(\sqrt{2D}(\gamma-1)+3\gamma) - \Phi(\sqrt{2D}(\gamma-1)-3\gamma) \approx 0.5 \qquad (۷-۲)$$

که D بعد داده ها و $\phi(.)$ تابع چگالی احتمال گاوسی است[۱۵].

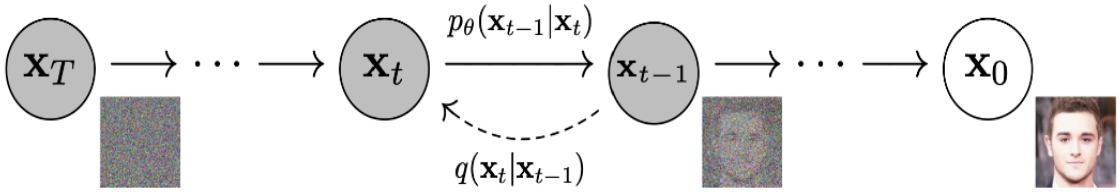
فصل سوم

مدل‌های پخشی نویززدای احتمالاتی

در این فصل به معرفی وبررسی مدل‌های پخشی نویززدای احتمالاتی[۱] یا به اختصار مپنا می‌پردازیم. این مدل‌ها نیز از همان قاعده‌ی کلی مدل‌های پخشی که اضافه کردن نویز در یک فرآیند رو به جلو و حذف آن به صورت معکوس است پیروی می‌کنند با این تفاوت که برای این فرآیند از یک زنجیره‌ی احتمالاتی مارکفی استفاده می‌کنند. در این فصل ابتدا به تعریف خود مدل و تابع هزینه‌ی آن می‌پردازیم، سپس با پارامتری سازی مجدد تابع هزینه ، هم ارزی این روش با روش معرفی شده در فصل دوم را نشان می‌دهیم و در نهایت در خصوص فرآیند یادگیری و بهبودهای این مدل مطالبی را مطرح می‌کنیم.

۳-۱ تعریف مدل

مپناها مدل‌هایی مبتنی بر متغیر پنهان[۲] هستند که از یک زنجیره‌ی مارکفی برای فرآیند پخشی استفاده می‌کنند. رویکرد کلی این مدل‌ها به این صورت است که در یک مسیر رو به جلو هر توزیع پیچیده‌ای از داده‌ها را می‌گیرد و آن را به یک توزیع اولیه که ساده و قابل محاسبه است تبدیل می‌کند و سپس یک مسیر روبه عقب محدود به زمان را یاد می‌گیرد که نمونه هایی از توزیع اولیه را به نمونه‌های توزیع داده تبدیل کند و این همان مدل مولد می‌باشد. تصویر ۳-۱ این فرآیند را نشان می‌دهد. بنابراین این مدل سه مشخصه‌ی اصلی دارد: مسیر رو به جلو، مسیر رو به عقب و تابع مدل مولد که در ادامه هر یک را توضیح می‌دهیم.



شکل ۳-۱: فرآیند پخش در مپناها

مدل گرافی جهت‌دار یک مپنا که به صورت یک زنجیره‌ی مارکف روبه جلو و معکوس آن می‌باشد[۶].

۳-۱-۱ مسیر رو به جلو

اگر توزیع داده را با $q(x^{(0)})$ نشان دهیم، این توزیع به صورت مرحله به مرحله به یک توزیع $\pi(y)$ که از نظر آنالیزی قابل بیان است تبدیل می‌شود و این تبدیل توسط یک هسته‌ی $T_\pi(y|y;\beta)$ صورت می‌گیرد

[1] Denoising diffusion probabilistic models

[2] Latent variable models

که β را نرخ پخش[3] می‌گوییم وداریم:

$$\pi(y) = \int dy' \, T_\pi(y|y';\beta)\pi(y') \tag{۱-۳}$$

$$q(x^{(t)}|x^{(t-1)}) = T_\pi(x^{(t)}|x^{(t-1)};\beta_t) \tag{۲-۳}$$

و بنابراین مسیر رو به جلو[4] با شروع از توزیع داده ها و طی T گام پخشی به صورت زیر است:

$$q(x^{(0\ldots T)}) = q(x^{(0)}) \prod_{t=1}^{T} q(x^{(t)}|x^{(t-1)}) \tag{۳-۳}$$

در ادامه‌ی این فصل هسته‌ی $q(x^{(t)}|x^{(t-1)})$ هسته ی گاوسی در نظر گرفته می‌شود اما هسته‌های دیگری چون هسته‌ی دوجمله‌ای[5] نیز وجود دارند. یک ویژگی حائز اهمیت مسیر رو به جلو این است که می‌توان به جای استفاده از فرم زنجیره‌ی مارکفی آن از یک فرم بسته استفاده کرد به این معنا که برای یافتن $x^{(t)}$ در زمان دلخواه t می‌توان از رابطه‌ی زیر کمک گرفت:

$$q(x_t|x_0) = \mathcal{N}(x_t; \sqrt{\overline{\alpha}_t}x_0, (1-\overline{\alpha}_t)I) \tag{۴-۳}$$

که $\alpha_t := 1 - \beta_t$ و $\overline{\alpha}_t := \prod_{s=1}^{t} \alpha_s$ تعریف می‌شود.

۳-۱-۲ مسیر رو به عقب

برای یادگیری توزیع مولد نیاز است که بتوانیم برعکس مسیر رو به جلو را یاد بگیریم و این به این معناست که باید بتوانیم از یک توزیع دلخواه قابل محاسبه به توزیع داده برسیم. بنابراین با تعریف توزیع اولیه به

[3] Diffusion rate

[4] Forward trajectory

[5] Binomial kernel

صورت $p(x^{(T)}) = \pi(x^{(T)})$ داریم:

$$p(x^{(0\ldots T)}) = p(x^{(T)}) \prod_{t=1}^{T} p(x^{(t-1)}|x^{(t)}) \tag{۵-۳}$$

می‌توان نشان داد که اگر فرآیند پخش گاوسی باشد و گام‌های پخش β خیلی کوچک باشند آنگاه معکوس فرآیند پخش نیز گاوسی می‌باشد[۲]. و بنابراین هر چقدر که طول این زنجیره بیشتر باشد β کوچکتر می‌شود و می‌توان از تابع گاوسی برای مسیر معکوس نیز کمک گرفت. در فرآیند یادگیری با توجه به اینکه هسته‌ی گذر[۶] زنجیره ی مارکفی یک تابع گاوسی است، برای ساخت زنجیره تنها نیاز به میانگین و کوواریانس این تابع گاوسی داریم و در ادامه سعی می‌کنیم آنها را تخمین بزنیم.

۳-۱-۳ تابع مدل مولد

تابع احتمال تولید شده توسط مپنا به صورت زیر می‌باشد:

$$p(x^{(0)}) = \int dx^{(1\ldots T)} p(x^{(0\ldots T)}) \tag{۶-۳}$$

از آنجاییکه انتگرال در حالت کلی غیرقابل محاسبه[۷] است، می‌توان با ایده گرفتن از نمونه برداری اهمیت[۸] به جای محاسبه‌ی انتگرال از میانگین احتمال نسبی مسیر رو به جلو و مسیر رو به عقب استفاده کرد که به صورت زیر قابل بیان است:

$$p(x^{(0)}) = \int dx^{(1\ldots T)} q(x^{(1\ldots T)}|x^{(0)}) \prod_{t=1}^{T} \frac{p(x^{(t-1)}|x^{(t)})}{q(x^{(t)}|x^{(t-1)})} \tag{۷-۳}$$

۳-۲ تعریف تابع خطا

با توجه به مدل تعریف شده در قسمت قبل، آموزش شبکه معادل بیشینه کردن لگاریتم درست‌نمایی[۹] تابع مدل مولد و یا کمینه کردن منفی این تابع است. بنابراین تابع هزینه را می‌توان به صورت زیر نوشت:

[6] Transition kernel

[7] Intractable

[8] Importance sampling

[9] Log-likelihood

$$L = \int dx^{(0)} q(x^{(0)}) \log p(x^{(0)}) \tag{۸-۳}$$

در اینجا برای آموزش به جای بهینه سازی مستقیم لگاریتم درست‌نمایی از بهینه سازی حد تغییراتی [۱۰] آن کمک می‌گیریم که به صورت زیر قابل تعریف است:

$$\mathbb{E}[-\log p_\theta(x^0)] \leq \mathbb{E}_q[-\log \frac{p_\theta(x^{0:T})}{q(x^{1:T}|x^0)}] = \mathbb{E}_q[-\log p(x^T) - \Sigma_{t\geq 1} \log \frac{p_\theta(x^{t-1}|x^t)}{q(x^t|x^{t-1})}] =: L \tag{۹-۳}$$

۳-۳ تغییر پارامتر تابع خطا

از آنجایی که ترکیب مسیر رو به جلو و مسیر معکوس شبیه به یک کدگذار تغییراتی است، می‌توان تابع هزینه را به صورت زیر بازنویسی کرد:

$$\mathcal{L}_{vlb} = \mathcal{L}_0 + \mathcal{L}_1 + \ldots + \mathcal{L}_{T-1} + \mathcal{L}_T \tag{۱۰-۳}$$

$$\mathcal{L}_0 = -\log p_\theta(x_0|x_1) \tag{۱۱-۳}$$

$$\mathcal{L}_{t-1} = \mathcal{D}_{KL}(q(x_{t-1}|x_t, x_0)\|p_\theta(x_{t-1}|x_t)) \tag{۱۲-۳}$$

$$\mathcal{L}_T = \mathcal{D}_{KL}(q(x_T|x_0)\|p(x_T)) \tag{۱۳-۳}$$

و از آنجایی که در صورت کوچک بودن β_t هر دوی $p_\theta(x_{t-1}|x_t))$ و $q(x_{t-1}|x_t, x_0)$ توزیع هایی گاوسی هستند در نتیجه دیورجنس KL [۱۱] آنها را می‌توان به فرم تفاضل میانگین آنها در نظر گرفت و در نتیجه برای معادله ی (۱۲-۳) داریم:

[10] Variational bound

[11] KL-divergence

$$\mathcal{L}_{t-1} = \mathbb{E}_q \left[\frac{1}{2\sigma_t^2} \|\tilde{\mu}_t(x_t, x_0) - \mu_\theta(x_t, t)\|^2 \right] + C \qquad (۱۴-۳)$$

که در واقع نشان می‌دهد که در فرآیند آموزش یک مدل پخشی بهینه سازی خطا شامل تخمین میانگین پسین[۱۲] فرآیند رو به جلو می‌باشد. از طرفی با تغییر پارامتر $x_t(x_0, \epsilon) = \sqrt{\bar{\alpha}_t}x_0 + \sqrt{(1-\bar{\alpha}_t)}\epsilon$ برای $\epsilon \sim \mathcal{N}(0, I)$ و قراردهی آن در فرمول معادله‌ی (۱۴-۳) می‌توان نشان داد که تخمین $\mu_\theta(x_t, t)$ به صورت زیر قابل پارامتری سازی است:

$$\mu_\theta(x_t, t) = \tilde{\mu}_t \left(x_t, \frac{1}{\sqrt{\bar{\alpha}_t}} \left(x_t - \sqrt{1-\bar{\alpha}_t}\epsilon_\theta(x_t) \right) \right) = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t) \qquad (۱۵-۳)$$

که ϵ_θ یک تخمین از نویز موجود در x_t می‌باشد و در نتیجه برای محاسبه ی یک نمونه ی x_{t-1} تنها کافیست که عبارت زیر را محاسبه کنیم:

$$x_{t-1} = \frac{1}{\sqrt{\alpha_t}}(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t) + \sigma_t z \qquad (۱۶-۳)$$

که z از توزیع گاوسی با میانگین صفر و کوواریانس همانی می آید.

این تغییر پارامتر دو موضوع مهم را نشان می‌دهد: اول اینکه برای آموزش یک فرآیند رو به عقب می‌توان از تخمین $\tilde{\mu}_t$ با تابع میانگین وابسته به زمان μ_θ استفاده کرد و یا مستقیما نویز اضافه شده به داده را با تخمین ϵ محاسبه کرد. دوم اینکه در حالتی از پارامتری سازی که از تخمین نویز در هر گام زمانی استفاده می‌کنیم در واقع فرآیندی مشابه تطبیق امتیاز نویززدا با دینامیک لانژوین را طی می‌کنیم و در این حالت تابع هدف شبکه را تبدیل به یک مدل تطبیق امتیاز نویززدا می‌کنیم.

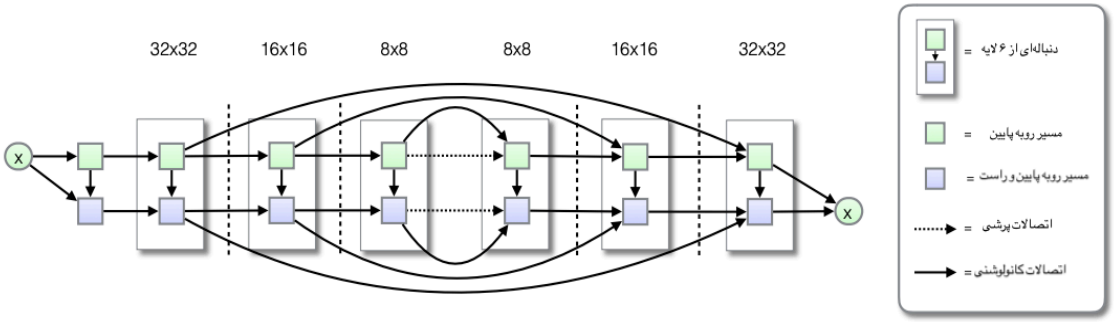
۳-۴ فرآیند یادگیری

با توجه به تغییر پارامتر انجام شده در قسمت قبل و همچنین به کارگیری آن درمعادله‌ی (۱۰-۳) می توان به تابع هزینه ای رسید که کاملا نسبت به θ مشتق پذیر است و برای آموزش مناسب است اما برای سادگی پیاده سازی و افزایش کیفیت نمونه‌های تولیدی می‌توان تابع هزینه را به شکل زیر نیز بازنویسی کرد:

[12] Posterior mean

$$L_{simple}(\theta) = \mathbb{E}_{t,x_0,\epsilon}\left[\left\|\epsilon - \epsilon_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon, t)\right\|^2\right] \quad (۱۷-۳)$$

که t از یک توزیع یونیفرم بین ۱و T می آید و واریانس β_t در این فرآیند یک پارامتر ثابت بدون یادگیری می‌باشد. در ساختار مپناها از هسته ی اصلی $pixelCNN++$ [۱۳] استفاده شده است که در واقع نوعی شبکه‌ی یو با اتصالات گسترده‌ی باقی‌مانده‌ای می‌باشد. شکل۳-۲ ساختار کلی شبکه‌ی به کار گرفته شده را نشان می‌دهد.



شکل ۳-۲: معماری $pixelCNN++$

شبکه‌ی $pixelCNN++$ که نوعی شبکه‌ی یو با اتصالات باقی‌مانده‌ای گسترده است به عنوان شبکه‌ی اصلی در مپناها استفاده می‌شود[۱۳].

۳-۵ بهبود مدل‌های پخشی نویززدای احتمالاتی

با وجود عملکرد فوق العاده‌ی مپناها در تولید نمونه‌های با کیفیت بالا، جنبه های مختلفی از این مدل‌ها نیاز به بازنگری و ارزیابی دارند. از جمله اینکه این مدل‌ها تا چه حد توانایی پیدا کردن همه‌ی مد[۱۳]های یک توزیع احتمالاتی را دارند و یا اینکه در مجموعه داده‌های متنوع مانند $ImageNET$، آیا قابلیت تولید نمونه‌های متنوع را دارند؟ لگاریتم درست‌نمایی، یک معیار پرکاربرد در بحث مدل‌های مولد است و بهینه سازی آن سبب می‌شود تا مدل مولد بتواند همه‌ی مد های یک توزیع را تولید کند[۱۱]. همچنین اخیرا نشان داده شده است که بهبود ناچیزی در بهینه سازی لگاریتم درست‌نمایی موجب تغییر چشم گیری در کیفیت نمونه و بازنمایی ویژگی های آموخته شده می‌شود[۴]. در مپنا با وجود عملکرد بسیار عالی در تولید تصاویر با کیفیت بالا مقدار لگاریتم درست‌نمایی قابل مقایسه با سایر روش‌های مولد نیست. اولین رویکردی که مسلما باعث بهبود لگاریتم درست‌نمایی در مپناها می‌شود، افزایش T یا تعداد گام است.

[13] Mode

هر چقدر T بیشتر باشد لگاریتم درست‌نمایی مپناها بهینه‌تر خواهد بود. از طرفی این افزایش موجب ناکارآمدی زمانی و هزینه‌ای مپناها می‌شود. در ادامه‌ی این فصل دو رویکرد برای بهینه‌تر کردن لگاریتم درست‌نمایی در مپناها را به اختصار معرفی می‌کنیم.

۳-۵-۱ یادگیری واریانس

در قسمت‌های قبل و درهنگام محاسبه‌ی تابع هزینه گفتیم که واریانس را ثابت و بدون یادگیری در نظر می‌گیریم. ثابت در نظر گرفتن σ_t از دیدگاه افزایش کیفیت نمونه‌های تولیدی انتخاب مناسب و کارایی است اما از آنجاییکه میزان تاثیر جملات مختلف حد تغییراتی در گام‌های ابتدایی بیشتر است پس برای بهبود لگاریتم درست‌نمایی یادگیری واریانس می‌تواند باعث بهبود این مقدار شود.

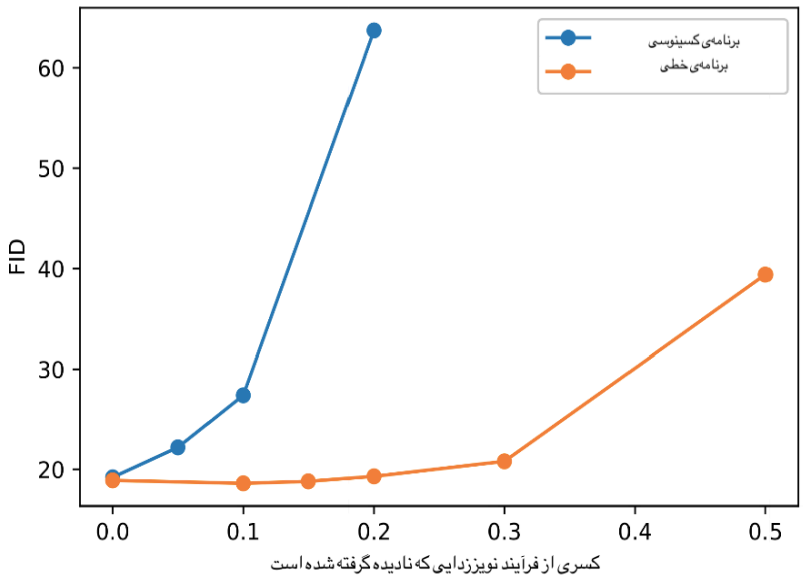
۳-۵-۲ بهبود گام‌های نویز

در مپنا مقدار نویز β_t با یک تابع خطی تغییر می‌کند. این تابع خطی برای تصاویر با رزولوشن بالا خوب عمل می‌کند اما برای تصاویر با رزولوشن پایین مناسب نیست. این امر به این علت است که انتهای فرآیند پخش شدت نویز تا حدی افزایش می‌یابد که عملا در کیفیت نمونه بی تاثیر می‌شود. این موضوع در شکل ۳-۳ به خوبی نشان داده شده است. همچنین همانطور که در شکل ۳-۴ مشاهده می‌کنید در مدلی که با نویز خطی پخش شده است، نادیده گرفتن ۲۰ درصد از فرآیند معکوس تاثیر مخربی بر کیفیت نمونه ها نمی‌گذارد.

شکل ۳-۳: تفاوت برنامه‌ی نویز خطی و کسینوسی در فرآیند پخشی

نمونه‌های فرآیند پخشی با برنامه‌ی نویز خطی (بالا) و نویز کسینوسی(پایین). نمونه‌های یک چهارم آخر برنامه‌ی نویز خطی عملا همگی نویز هستند و فرآیند نویزافزایی خیلی آهسته صورت گرفته است[۱۰].

برای حل این مشکل می‌توانیم از انواع دیگری از برنامه‌های نویز مانند برنامه‌ی نویز کسینوسی استفاده کنیم. همان طور که در شکل ۳-۳ می‌بینید، این برنامه‌ی نویز با سرعت آهسته‌تری از برنامه‌ی نویز خطی به تصویر نویز اضافه می‌کند. همچنین همانطور که در شکل ۳-۴ قابل مشاهده است حذف بخشی از فرآیند رو به عقب در این برنامه‌ی نویز باعث افت کیفیت تصویر می‌شود.



شکل ۳-۴: تفاوت برنامه‌ی نویز خطی و کسینوسی در حذف بخشی از فرآیند نویززدایی

در مدلی که با نویز خطی پخش شده است، نادیده گرفتن ۲۰ درصد از فرآیند معکوس تاثیر مخربی بر کیفیت نمونه ها نمی‌گذارد در حالیکه در مدل کسینوسی حذف هر کسری از فرآیند معکوس کیفیت را بدتر می‌کند[۱۰].

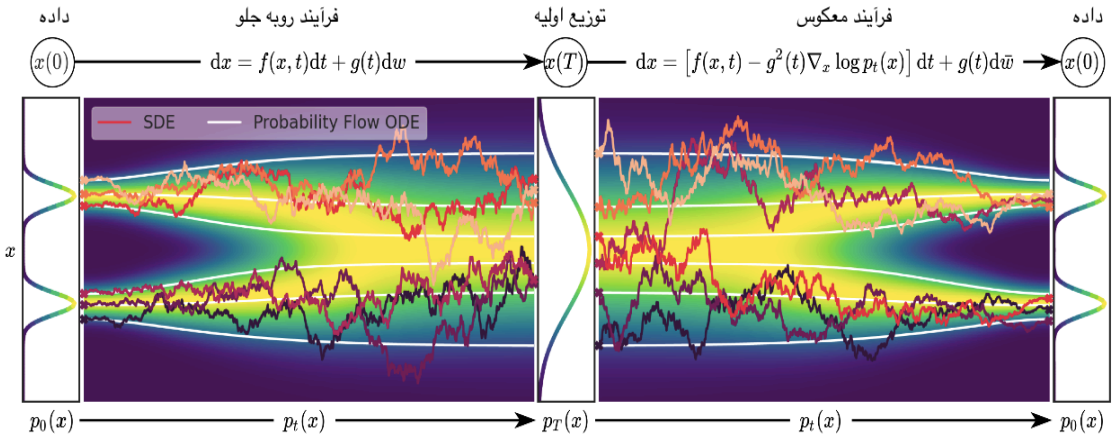
فصل چهارم

مدل‌های مولد مبتنی بر معادلات دیفرانسیل تصادفی

در دو فصل گذشته دو گونه از مدل‌های پخشی را معرفی کردیم که نقطه‌ی اشتراک هر دو این بود که در یک فرآیند پخش داده را با نویز فزاینده تخریب می‌کردند و سپس در یک فرآیند معکوس از این داده‌ی تخریب شده به مدلی مولد برای تولید داده‌ی جدید می‌رسیدند. در این فصل یک چهارچوب کلی برای فرآیند پخش تعریف می‌کنیم که به جای تخریب داده با یک سری گام گسسته‌ی متناهی، داده را با بی نهایت گام پیوسته تخریب می‌کنیم. این فرآیند تخریب از طریق یک معادله دیفرانسیل تصادفی تعریف شده صورت می‌گیرد و سپس نشان داده می‌شود که برای فرآیند تصادفی یک معادله دیفرانسیل تصادفی معکوس نیز وجود دارد که تنها وابسته به تابع امتیاز می‌باشد. در این بخش ابتدا به معرفی این چهارچوب کلی و مزایای آن می‌پردازیم، سپس ارتباط آن با دو روش پیش‌تر معرفی شده را بیان می‌کنیم و در نهایت به انواع روش‌های نمونه برداری از این توزیع می‌پردازیم.

۱-۴ معرفی فرآیند پخشی با معادلات دیفرانسیل تصادفی

ایده‌ی اصلی در مدل‌هایی که مبتنی بر معادلات دیفرانسل تصادفی هستند این است که تعداد گام‌های حرکت از توزیع داده به سمت توزیع نویز بی نهایت و به صورت حدی کوچک باشد به طوریکه توزیع داده‌های تخریب شده با افزایش نویز به صورت یک معادله دیفرانسیل تصادفی تغییر کند. مانند مدل‌های قبلی این مدل نیز از سه بخش مسیر رو به جلو، مسیر معکوس و تابع هدف یادگیری تشکیل شده است.تصویر ۱-۴ شمای کلی این روش را نشان می‌دهد.



شکل ۴-۱: مدل‌های پخشی مبتنی بر معادلات دیفرانسیل

مدل‌های پخشی مبتنی بر معادلات دیفرانسیل از فرآیندهای تصادفی و این معادلات برای مدل کردن فرآیند رو به جلو و فرآیند معکوس استفاده می‌کنند[۱۶].

۴-۱-۱ مسیر روبه جلو

همانطور که پیش‌تر هم اشاره شد در این مدل هدف توسعه‌ی فرآیند پخش $\{x(t)\}_{t=0}^{T}$ است به طوریکه $x(0) \sim p_0$ یا همان توزیع داده‌ها و $x(T) \sim p_T$ یا همان توزیع اولیه[۱] باشد و متغیر t یک متغیر پیوسته در بازه‌ی $[0, T]$ باشد. در چنین حالتی فرآیند پخش را یک راه حل معادله دیفرانسل تصادفی به فرم زیر در نظر می‌گیریم:

$$dx = f(x,t)dt + g(t)dw \tag{۴-۱}$$

که در عبارت بالا $f(t,.) : \mathbb{R}^d \to \mathbb{R}^d$ ضریب یا تابع رانش[۲] و $g(.)$ یک مقدار عددی وابسته به زمان است که ضریب پخش[۳] گفته می‌شود. w نیز فرآیند وینر استاندارد[۴] که معادل همان نویز سفید تصادفی در فضای پیوسته می‌باشد. به طور کلی توزیع اولیه ی p_T هر توزیع دلخواه قابل محاسبه‌ای است که هیچ اطلاعاتی از توزیع داده در خود ندارد. مانند یک توزیع گاوسی با میانگین و کوواریانس ثابت. حال برای هر توزیع اولیه‌ی مشخص یا هر شکلی از فرآیند پخش یک معادله دیفرانسیل تصادفی طراحی می‌شود که امکان تبدیل توزیع داده به توزیع مشخص شده را دارد.

۴-۱-۲ مسیر معکوس

می‌دانیم که معکوس یک فرآیند پخشی که با معادلات دیفرانسیل مدل می‌شود نیز یک فرآیند پخشی است که در جهت عکس زمانی حرکت می‌کند و با معادله دیفرانسیل تصادفی معکوس زیر مدل می‌شود:

$$dx = [f(x,t) - g(t)^2 \bigtriangledown_x log p_t(x)]dt + g(t)d\bar{w} \tag{۴-۲}$$

که $\bar{w}$ همان فرآیند وینر استاندارد در جهت معکوس از T به 0 است و dt نیز در جهت منفی حرکت می‌کند[۱]. همانطور که در معادله‌ی(۴-۲) مشاهده می‌شود این عبارت تنها نیازمند تابع امتیاز توزیع $p_t(x)$ می‌باشد و با داشتن این تابع می‌توانیم معکوس فرآیند پخشی را بسازیم و از توزیع داده‌ها نمونه تولید کنیم.

[1] Prior distribution
[2] Drift coefficient
[3] Diffusion coefficient
[4] Standard wiener process

۴-۱-۳ تخمین تابع امتیاز

تابع امتیاز یک توزیع همانطور که در فصل دوم مشاهده شد با فرآیند تطبیق امتیاز قابل تخمین و یادگیری است. تنها نکته‌ی حائز اهمیت این است که در این حالت نیاز به یک تعمیم پیوسته از تابع خطای این شبکه داریم. این تعمیم به صورت زیر قابل بیان است:

$$\theta^{\star} = \arg\min_{\theta} \mathbb{E}_t \left\{ \lambda(t) \mathbb{E}_{x(0)} \mathbb{E}_{(x(t)|x(0))} \left[\left\| s_\theta(x(t), t) - \bigtriangledown_{x(t)} \log p_{0t}(x(t)|x(0)) \right\|_2^2 \right] \right\} \quad (۴-۳)$$

که $\lambda(t)$ یک تابع وزن‌دهی مثبت است و t به صورت یکنواخت از بازه‌ی $[0, T]$ نمونه برداری شده است و $x(0) \sim p_0(x)$ و $x(t) \sim p(x(t)|x(0))$ می‌باشد. اگر ظرفیت مدل مناسب و تعداد داده‌ها کافی باشد معادله ی (۴-۳) ما را به پارامترهای بهینه‌ی تابع امتیاز می‌رساند[۱۶].

۴-۲ مزیت‌های مدل‌های مولد مبتنی بر معادله دیفرانسیل تصادفی

سه مزیت کلی برای این دسته از روش‌ها می‌توان نام برد که عبارتند از:

- نمونه برداری و محاسبه‌ی درست‌نمایی منعطف: در این روش از هر حل کننده ی کلی معادلات دیفرانسیل تصادفی برای نمونه گیری از معادله دیفرانسیل تصادفی معکوس می‌توان کمک گرفت. علاوه بر این می‌توان از روش‌های دیگری حل‌کننده های قطعی نیز برای نمونه گیری کمک گرفت.
- تولید قابل کنترل: از آنجاییکه معادله دیفرانسل تصادفی معکوس شرطی شده از توابع امتیاز غیر شرطی قابل محاسبه است، می‌توانیم فرآیند تولید نمونه را روی اطلاعاتی شرطی کنیم که در حین یادگیری در اختیار مدل قرار نگرفته است. این امر باعث می‌شود تا کاربردهایی مثل ترمیم تصویر[۵]، تولید شرطی شده روی کلاس و رنگ آمیزی[۶] با یک مدل امتیاز غیرشرطی و بدون آنکه نیاز به آموزش مجدد باشد، قابل انجام باشند.
- ایجاد یک چهارچوب یکپارچه: این روش قابلیت جستجو و به کارگیری انواع مختلفی از معادلات دیفرانسیل تصادفی را برای بهبود مدل‌های پخشی دارد. همچنین هر دو روش پیش‌تر معرفی شده نیز درقالب این روش قابل بیان و بهینه سازی هستند.

[5] Image inpainting
[6] Colorization

۴-۳ ارتباط روش مبتنی بر معادلات دیفرانسیل تصادفی با روش‌های پیش‌تر معرفی شده

به طور کلی می‌توان گفت فرآیند پخش به کار گرفته شده در مپنا و تطبیق امتیاز با دینامیک لانژوین، گسسته‌سازی‌هایی از روش ارائه شده در این فصل هستند. هر کدام از این روش‌ها از یک معادله دیفرانسیل تصادفی برخوردار هستند که با گسسته‌سازی آن می‌توان هم ارزی روش‌های پیش‌تر معرفی شده را با روش مبتنی بر معادله دیفرانسیل تصادفی نشان داد. در روش تطبیق امتیازبا دینامیک لانژوین با استفاده از N سطح نویز می‌توان تابع تخریب $p_{\sigma_i}(x|x_0)$ را با توزیع x_i حاصل از یک زنجیره‌ی مارکف به شکل زیر به دست آورد:

$$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} z_{i-1}, \quad i = 1, \dots, N \tag{۴-۴}$$

که $z_{i-1} \sim \mathcal{N}(0, I)$ و در صورتی که $N \to \infty$ آنگاه زنجیره‌ی مارکفی بالا به یک فرآیند تصادفی پیوسته به صورت $\{x(t)\}_{t-0}^{1}$ تبدیل می‌شود که خود t هم یک متغیر پیوسته در بازه‌ی $[0, T]$ است. در چنین صورتی می‌توانیم این فرآیند تصادفی را با یک معادله دیفرانسیل تصادفی به فرم زیر مدل کنیم:

$$dx = \sqrt{\frac{d[\sigma^2(t)]}{dt}} dw \tag{۵-۴}$$

این معادله ،که به آن معادله ی انفجار واریانس[۷] می‌گوییم، در واقع حالت خاصی از چهارچوب مطرح شده در این فصل می‌باشد.

به طور مشابه برای مپنا نیز یک زنجیره‌ی مارکفی به شکل زیر داریم:

$$x_i = \sqrt{1 - \beta_i} x_{i-1} + \sqrt{\beta_i} z_{i-1}, \quad i = 1, \dots, N \tag{۶-۴}$$

که در صورتی که $N \to \infty$، به معادله دیفرانسیل تصادفی زیر همگرا می‌شود:

$$dx = -\frac{1}{2}\beta(t) x dt + \sqrt{\beta(t)} dw \tag{۷-۴}$$

و بنابراین تخریب های نویز به کار برده شده در مپنا و در تطبیق امتیاز با دینامیک لانژوین هر دو گسسته سازی هایی از یک معادله دیفرانسل تصادفی هستند.

[7] Variance exploding

۴-۴ انواع روش‌های نمونه برداری

بعد از آموزش یک شبکه‌ی امتیاز وابسته به زمان می‌توان از آن برای ساخت یک معادله دیفرانسیل تصادفی معکوس استفاده کرد و سپس با استفاده از روش‌های عددی حل این معادله دیفرانسیل آن را شبیه‌سازی و نمونه‌هایی از توزیع p_0 تولید کرد. در این قسمت به دو دسته روش حل این معادلات اشاره خواهیم کرد:

۴-۴-۱ حل کننده‌های عام منظوره ی معادلات دیفرانسیل تصادفی

حل کننده‌های عددی مسیر تقریبی معادلات دیفرانسیل تصادفی را تخمین می‌زنند. بسیاری از این حل کننده‌های عام منظوره مانند روش اویلر-ماریاما[۸] برای تولید نمونه به کار گرفته می‌شوند.

۴-۴-۲ تبدیل به معادلات دیفرانسیل ساده

با دانستن تابع امتیاز یک راه دیگر برای حل معادلات دیفرانسیل تصادفی و نمونه برداری، تبدیل این معادلات به معادلات دیفرانسیل ساده[۹] و تبدیل فرآیند تصادفی به فرآیند قطعی[۱۰] می‌باشد. به ازای هر فرآیند پخشی تصادفی یک فرآیند قطعی متناظر آن وجود دارد که هر دو مسیر توزیع‌های حاشیه‌ای یکسانی دارند. این فرآیند قطعی به صورت زیر قابل بیان است:

$$dx = [f(x,t) - \frac{1}{2}g^2(t) \bigtriangledown_x \log p_t(x)]dt \tag{۸-۴}$$

در صورتی که تابع امتیاز را بدانیم این مسیر به راحتی از معادله دیفرانسیل تصادفی قابل محاسبه است. استفاده از معادلات دیفرانسیل ساده به جای معادلات دیفرانسیل تصادفی مزیت های متعددی از جمله توانایی محاسبه‌ی دقیق درست‌نمایی، کدگذاری یکتا و نمونه برداری کارا دارد.

[8] Euler-maruyama

[9] Ordinary differential equations

[10] Deterministic process

فصل پنجم

مقایسه‌ی روش‌های مولد

در این فصل ابتدا به بررسی ارتباط بین مدل‌های پخشی با سایر مدل‌های مولد و مقایسه‌ی آنها می‌پردازیم؛ سپس مدل‌های پخشی پیش‌تر معرفی شده را از دو جنبه‌ی عملکرد و نتایج به دست آمده و معماری با یکدیگر مقایسه می‌کنیم.

۵-۱ بررسی ارتباط مدل‌های پخشی با سایر مدل‌های مولد

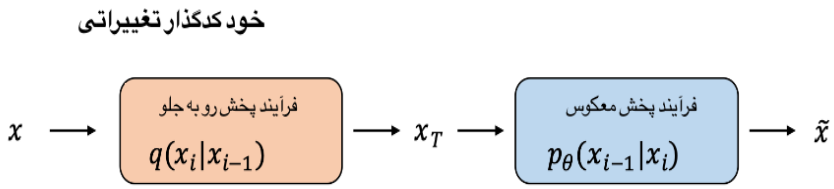
۵-۱-۱ ارتباط خودکدگذارهای تغییراتی با مدل‌های پخشی

خودکدگذارهای تغییراتی[۸] شامل یک کدگذار[۱] و یک کدگشا[۲] هستند که هر دو برای پیدا کردن نگاشتی بین داده‌های ورودی و یک فضای پنهان پیوسته آموزش داده می‌شوند. در این مدل‌ها تعبیه[۳] به دست آمده به عنوان یک متغیر پنهان در یک مدل مولد احتمالاتی در نظر گرفته می‌شود و یک کدگشای احتمالاتی آموزش داده می‌شود تا ازاین متغیر پنهان به داده برسد. بنابراین فرض می‌شود که داده‌ی x از یک متغیر پنهان مشاهده نشده‌ی z با یک توزیع شرطی $p_\theta(x|z)$ تولید می‌شود و $q_\phi(z|x)$ برای تخمین z به کار گرفته می‌شود. در این مدل برای تضمین دستیابی به تخمین مناسبی از z از یک رویکرد تغییراتی بیزی برای بیشینه کردن حد پایین مشاهده به فرم $\mathcal{L}(\phi,\theta,x) \leq \log p_\theta(x)$ که داریم:

$$\mathcal{L}(\phi,\theta;x) = \mathbb{E}_{q(z|x)}[\log p_\theta(x,z) - \log q_\phi(z|x)] \qquad (۱-۵)$$

استفاده می‌شود. برای این نوع فرموله سازی این مدل انتخاب‌های متعددی برای مدل‌های کدگذار و کدگشا وجود دارد، معمولا این مدل‌ها با خانواده‌ی توزیع‌های نمایی مدل می‌شوند که پارامترهای این توزیع از شبکه‌های عصبی چند لایه تولید می‌شود. مپناها را می‌توان به عنوان خودکدگذارهای تغییراتی سلسله مراتبی[۴] صورت بندی کرد که در آنها کدگذار ثابت است. در چنین صورت بندیی مسیر رو به جلو نقش کدگذار را ایفا می‌کند و به عنوان یک مدل گاوسی خطی در نظر گرفته می‌شود و مسیر روبه عقب نیز نقش کدگشا را ایفا می‌کند که در گام‌های مختلف نویززدایی به کار گرفته می‌شود و همچنین اندازه‌ی متغیر پنهان به کارگرفته شده در کدگشا نیز با اندازه‌ی داده‌ی ورودی برابر است. شکل ۵-۱ این موضوع را نشان می‌دهد.

[1] Encoder
[2] Decoder
[3] Embedding
[4] Hierarchical variational autoencoders



شکل ۵-۱: مدل‌های پخشی مبتنی بر معادلات دیفرانسیل

مدل‌های پخشی مشابه یک خودکدگذار تغییراتی سلسله مراتبی هستند که کدگذار در آنها ثابت است[۲۰].

۵-۱-۲ ارتباط شبکه‌های مولد تقابلی با مدل‌های پخشی

شبکه‌های مولد تقابلی[۳] از دو زیر شبکه‌ی تولید کننده‌ی G و تمایزگر D تشکیل شده‌اند که معمولا به فرم شبکه‌های عصبی پیاده سازی می‌شوند اما در هر فرمی از سیستم‌های مشتق پذیر که داده را از یک فضا به فضای دیگر می‌برند نیز قابل پیاده سازی هستند. تابع هزینه‌ی شبکه‌های مولد تقابلی به صورت زیر قابل بیان است:

$$\min_G \max_D \mathbb{E}_{x\sim p_{data}}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \qquad (۵-۲)$$

شبکه‌ی تولیدکننده تلاش می‌کند با تولید نمونه‌های جدید به صورت ضمنی توزیع داده را مدل کند و در مقابل شبکه‌ی تمایزگر تلاش می‌کند که با حداکثر دقت نمونه‌های تولید شده را از نمونه‌های واقعی جدا کند. فرآیند بهینه سازی این شبکه تا جایی ادامه پیدا می‌کند که به یک نقطه‌ی زینی[۵] که کمینه برای شبکه‌ی تولید کننده و بیشینه برای شبکه‌ی تمایزگر است برسد. در چنین نقطه‌ای شبکه‌ی مولد توزیع صحیح نمونه‌های داده را یادگرفته است.

در ساختار پایه‌ای شبکه‌های مولد تقابلی رابطه‌ی واضحی با مدل‌های پخشی وجود ندارد اما این دو مدل می‌توانند در صورت ترکیب با یکدیگر مشکلات و ضعف هایشان را برطرف کنند. به عنوان مثال یک مساله‌ی خیلی رایج در شبکه‌های مولد تقابلی، مساله‌ی عدم پایداری آنها است که عموما به علت عدم همپوشانی بین توزیع داده‌ی واقعی و توزیع داده‌ی تولید شده ایجاد می‌شود. یک راه حل این مساله این است که با اضافه کردن نویز به ورودی تمایزگر فضای پشتیبان توزیع‌های تمایزگر و تولید کننده را افزایش دهیم. این راه حل می‌تواند توسط یک مدل پخشی صورت بگیرد. در [۱۸] نویسندگان از یک مدل پخشی با برنامه‌ی نویز تطبیقی برای اضافه کردن نویز به تمایزگر استفاده کردند. از سوی دیگر شبکه‌های مولد تقابلی نیز می‌توانند در افزایش سرعت نمونه برداری مدل‌های پخشی موثر واقع شوند. در [۱۹] برای کاهش تعداد گام‌های نویززدایی از یک مدل مولد تقابلی شرطی برای مدل کردن هر گام نویززدایی استفاده شده است.

[5] Saddle point

۵-۱-۳ مشکل سه‌گانه‌ی مدل‌های مولد

با وجود توسعه‌ی حجم زیادی از مدل‌های مولد در سال‌های اخیر برای انواعی از کاربردها از جمله تصویر، صدا، متن و گراف ها، همه‌ی مدل های موجود فعلی قابلیت ارضای سه نیاز اصلی یک مدل مولد به کار گرفته شده در مسائل واقعی را ندارند[۱۹]. این سه نیاز اصلی عبارتند از:

۱. تولید نمونه‌های با کیفیت

۲. پوشش مد و تنوع نمونه‌های تولیدی

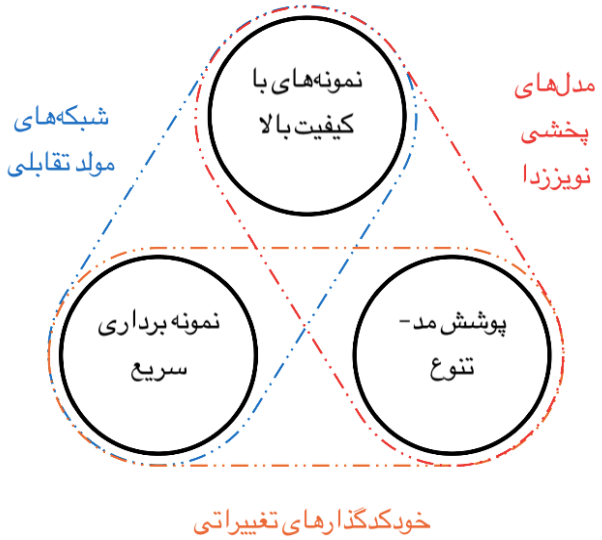
۳. نمونه برداری سریع و به صرفه از نظر محاسباتی

به عنوان مثال اکثر مدل‌های تولید تصویر فعلی روی تولید نمونه‌های با کیفیت تمرکز دارند اما تنوع داده‌ها نیز موضوع مهمی در کاهش تاثیرات اجتماعی منفی مدل‌های مولد است. همچنین برای کاربردهای تعاملی نیاز به نمونه برداری سریع یک مساله‌ی حائز اهمیت است. این چالش در ادبیات مدل‌ها مولد به عنوان مشکل سه گانه‌ی یادگیری مولد[۶] شناخته می‌شود. هر کدام از مدل‌های مولد در حل بخشی از مشکلات یادگیری مولد تبحر دارند و بخش‌هایی از این مشکل به عنوان ضعف آنها محسوب می‌شود. به عنوان مثال شبکه‌های مولد تقابلی قابلیت تولید نمونه‌های با کیفیت در زمان سریع را دارند اما پوشش مد ضعیفی دارند. از طرفی خودکدگذار های تغییراتی به خوبی مدهای داده را پوشش می‌دهند و سرعت نمونه‌برداری مناسبی دارند اما کیفیت نمونه‌های تولیدی آنها کم است. برای مقایسه‌ی عملکرد مدل‌های پخشی با سایر مدل‌های مولد خوب است بدانیم که این مدل‌ها قابلیت رفع این مشکل سه گانه را دارند یاخیر. مدل‌های پخشی با ظهور خود توانستند از شبکه‌های مولد تقابلی در تولید نمونه‌های با کیفیت پیشی بگیرند و همچنین با تغییر و تنظیم این مدل‌ها و دستیابی به درست‌نمایی بهینه، قابلیت پوشش مدهای مختلف داده را دارند اما نمونه برداری از این مدل‌ها نیازمند صدها بار استفاده از شبکه می‌باشد و همین امر سرعت نمونه برداری را کاهش می‌دهد و به کارگیری آنها در کاربرد را هزینه‌بر ودشوار می‌سازد. شکل ۵-۲ این مشکل سه گانه و حوزه ی عملکرد مناسب هر مدل را نشان می‌دهد.

همچنین جدول ۵-۱-۳ نیز مقایسه ای از ویژگی های کلی این سه مدل مولد ارائه می‌دهد. ویژگی تعیین یکتا[۷] نشان دهنده ی این است که با تغییر معماری و مقداردهی اولیه تا چه حد خروجی مدل و کدگذاری آن تغییر می‌کند. در مدل‌های پخشی به شرط داشتن داده‌ی کافی، دقت بهینه سازی مناسب و ظرفیت لازم مدل، برخلاف مدل های مولد تقابلی و خودکدگذار های تغییراتی با تغییر معماری و مقداردهی اولیه همچنان به یک خروجی و کدگذاری میرسیم[۱۶].

[6] Generative learning trilemma

[7] Unique identifiability



شکل ۵-۲: مشکل سه‌گانه‌ی مدل‌های مولد

سه نیاز اساسی در توسعه‌ی مدل‌های مولد و حوزه‌ی عملکرد موثر هر مدل[۱۹].

جدول ۵-۱: مقایسه‌ی مدل‌های مولد با یکدیگر

مدل	مدل‌های پخشی	خودکدگذارهای تغییراتی	شبکه‌های مولد تقابلی
تعداد شبکه	۱	۲	۲
نوع آموزش	نویززدایی تکراری	استنتاج تغییراتی	تقابلی
سرعت آموزش	سریع	آهسته	آهسته
کیفیت تصاویر	بالا	متوسط	بالا
تعیین یکتا	بله	خیر	خیر

۵-۲ مقایسه‌ی انواع روش‌های پخشی

در سه فصل گذشته سه مدل از انواع روش‌های پخشی نام برده شد که هرکدام می‌توانستند هم ارز یا در برگیرنده‌ی روش دیگری باشند. در حالت کلی می‌توان گفت که مدل‌های پخشی مبتنی بر معادلات دیفرانسیل تصادفی در برگیرنده ی دو مدل دیگر هستند و این دو مدل با وجود اشتراک و هم ارزی با یکدیگر تحت شرایط خاص (تخمین نویز و واریانس ثابت) در حالت کلی دو فرمولاسیون متفاوت از معادلات دیفرانسیل تصادفی هستند. در این قسمت می‌خواهیم عملکرد و نتایج، معماری این مدل‌ها و برخی از ویژگی‌های کلی آنها را با یکدیگر مقایسه کنیم.

جدول ۵-۲: مقایسه‌ی کیفیت تصاویر تولیدی با سه مدل پخشی روی مجموعه داده‌ی $CIFAR-10$

مدل	FID
مدل‌های مبتنی بر امتیاز	۱۰.۸۷
مپنا	۳.۱۷
مدل مبتنی بر معادله دیفرانسیل (۴-۵)	۲.۲۰
مدل مبتنی بر معادله دیفرانسیل (۴-۷)	۲.۴۱

۵-۲-۱ نتایج و عملکرد

برای مقایسه‌ی نتایج هر مدل از معیار کیفیت فاصله‌ی اکتسابی فرچت[۸] (FID) [۵] استفاده می‌کنیم و نتایج را بر روی مجموعه داده ی$CIFAR-10$ برای مقالات اولیه‌ی هر کدام که [۱۴] برای تطبیق امتیاز نویززدا با دینامیک لانژوین، [۶] برای مپنا و [۱۶] برای مدل‌های پخشی مبتنی بر معادلات دیفرانسیل تصادفی است، بررسی می‌کنیم. فاصله‌ی اکتسابی فرچت معیاری است که فاصله ی بین بردار ویژگی تصاویر اصلی و تصاویر تولید شده را محاسبه می‌کند. هر چقدر که این معیار کمتر باشد نشان می‌دهد که تصاویر دو گروه مشابهت آماری بیشتری دارند و همچنین کمتر بودن این معیار با تصاویر پایه‌ی مجموعه داده همبستگی خوبی با کیفیت تصاویر تولیدی دارد.

جدول ۵-۲-۱ معیار FID را برای سه روش معرفی شده بر روی مجموعه ی $CIFAR-10$ نشان می‌دهد.

۵-۲-۲ معماری

معماری شبکه‌ی یو[۱۲] رایج‌ترین معماری استفاده شده برای مدل‌های پخشی است. علت محبوبیت این معماری این است که یک ساختار متقارن دارد که بازنمایی هم ویژگی‌های سطح پایین و هم ویژگی‌های سطح بالا را یاد می‌گیرد و به نسبت، ساختاری ساده و قابل تغییر دارد. نوع شبکه‌ی یوی به کار رفته در هر یک از مدل‌ها متفاوت است. معماری شبکه بسته به رزولوشن تصویر، وظیفه ی مورد نظر و نوع مدل متفاوت است و وجه مشترک همه‌ی آنها تنها در استفاده از انواعی از شبکه ی یو است. معماری‌های ذکر شده برای تصاویر با رزولوشن تا $64x64$ هستند.

مدل تطبیق امتیاز با دینامیک لانژوین در شبکه‌ی امتیاز شرطی شده با نویز خود از یک شبکه ی یو با ۶ بلوک باقی مانده ای [۹] در هر سمت کدگذار و کدگشا استفاده می‌کند. مپناها از یک شبکه‌ی یو با ۸ بلوک باقی مانده‌ای در هر سمت کدگذار و کدگشا استفاده می‌کنند که جمعا شامل ۱۵ بلوک باقی‌مانده‌ای است. مدل‌های پخشی مبتنی بر معادله دیفرانسیل تصادفی نیز از یک شبکه‌ی یو با ۱۶ بلوک باقی‌مانده‌ای در هر سمت کدگذار و کدگشا استفاده می‌کنند.

[8] Frechet inception distance

[9] Residual block

مزیت اصلی شبکه‌ی یو به کارگیری اتصالات پرشی[10] است. اتصالات پرشی اتصالاتی هستند که از لایه‌های مختلف شبکه عبور می‌کنند و اجازه‌ی حفظ ویژگی‌های سطح پایین داده را می‌دهند که امری بسیار مهم برای تولید عکس‌هایی شبیه به واقعیت است.
انتخاب نوع معماری و تعداد پارامترها بسته به رویکرد و وظیفه‌ی مدنظر می‌تواند متفاوت باشد. به عنوان مثال اگرهدف تولید تصاویر با کیفیت بالا باشد ممکن است نیاز باشد که ساختار پیچیده‌تر با تعداد پارامتر بیشتر به کار گرفته شود و از طرفی اگر به دنبال سرعت در تولید تصاویر باشیم، نیاز داریم از یک شبکه ی ساده تر با تعداد پارامتر کمتر استفاده کنیم.

۵-۲-۳ ویژگی‌های کلی

با توجه به همه ی مواردی که پیش‌تر گفته شد و آنچه در مقالات مطالعه شده است، می‌توان چند نکته ی کلی درخصوص شباهت ها و تفاوت های سه مدل پخشی معرفی شده بیان کرد:

- هر سه مدل برای آموزش خود به منابع محاسباتی زیادی نیاز دارند و از نظر هزینه‌ی محاسباتی و زمان مورد نیاز برای یادگیری بسیار هزینه‌بر و زمان‌بر هستند.
- در مدل مپنا و مبتنی بر معادلات دیفرانسیل تصادفی، حساسیت زیادی نسبت به تعیین ابرپارامترها، تابع نویز زدا و معماری مدل وجود دارد و عملکرد مناسب آنها به انتخاب صحیح این پارامترها بستگی دارد.
- روش‌های مبتنی بر معادلات دیفرانسیل تصادفی انعطاف زیادی در تغییر فرآیند نویزافزایی دارند و می‌توان از آنها برای کاربردهایی که فرآیند تخریب نامشخص یا نیازمند تغییر است استفاده کرد.
- مدل تطبیق امتیاز با دینامیک لانژوین به دلیل معضل بعد[11] در داده‌های با بعد بالا عملکرد مناسبی ندارد.

[10] Skip connections

[11] curse of dimensionality

فصل ششم

جمع‌بندی و کارهای آتی

۶-۱ جمع‌بندی

در این گزارش تلاش شد تا اطلاعات اولیه‌ای برای آشنایی و یافتن یک دیدگاه کلی در خصوص مدل‌های پخشی ارائه شود. هدف کلی از این گزارش بررسی مقالات پایه و مبنای تئوری این مدل‌ها برای درک ساختار و سازو کار تولید داده بود و از پرداختن به حجم زیادی از کاربردها و بهبودها که با توجه به رشد روز افزون به کارگیری مدل های پخشی رو به فزونی است خودداری شده است.

در فصل اول به معرفی مدل‌های پخشی مبتنی بر امتیاز پرداختیم. این مدل‌ها در رویکرد اولیه‌ی خود تماما پخشی نیستند و تنها با استفاده از یک تطبیق امتیاز و دینامیک لانژوین قابل انجام هستند. اما رویکرد اولیه‌ی این روش‌ها با محدودیت‌هایی از جمله مشکل رویه‌ها و نواحی کمتر چگال توزیع داده روبرو بود که این محدودیت‌ها در داده‌های واقعی اکثر اوقات برقرار هستند. به همین دلیل در مدل ارائه شده که تطبیق امتیاز نویززدا با دینامیک لانژوین تابکاری شده نام دارد، به دنبال برطرف کردن این مشکل هستیم و این روش مانند تمام رویکردهای پخشی به صورت مرحله به مرحله داده را با نویز تخریب و سپس در فرآیند معکوس نمونه‌هایی از داده را تولید می‌کند.

در فصل دوم به معرفی مپنا پرداختیم، مدلی که رایج‌ترین نوع مدل‌های پخشی است و در کاربردهای تولید تصویر مطرح مانند $Dall-E$ به کار گرفته شده است. این مدل به صورت احتمالاتی و با یک زنجیره ی مارکفی فرآیند پخش را مدل می‌کند. برای آموزش این شبکه رویکرد های مختلفی وجود دارد. در یک رویکرد می‌توانیم از تخمین میانگین پسین فرآیند رو به جلو برای آموزش شبکه استفاده کنیم و در رویکرد دوم می‌توانیم مستقیما نویز اضافه شده به تصویر در یک گام از فرآیند پخش را تخمین بزنیم. می‌توان نشان داد که با تغییر پارامتر در رویکرد دوم و ثابت در نظر گرفتن واریانس این روش هم ارز روش تطبیق امتیاز با دینامیک لانژوین است.

در فصل سوم چهارچوبی کلی برای فرآیند پخشی معرفی شد که قابلیت دربرگیری روش‌های پیش‌تر معرفی شده را دارد. در این روش نیز به صورت مرحله به مرحله داده را با نویز تخریب می‌کنیم اما به جای به کارگیری تعداد گام‌های محدود از تعداد گام نامتناهی پیوسته در یک بازه استفاده می‌کنیم. برای مدل کردن فرآیند پخش در چنین حالتی از یک معادله دیفرانسیل تصادفی استفاده می‌کنیم که معکوس چنین معادله‌ای را می‌توانیم تنها با داشتن تابع امتیاز بسازیم. برای ساخت نمونه از چنین معادله‌ای تنها کافی است از یک حل کننده استفاده کنیم و آن را گسسته کنیم. همچنین در این فصل نشان دادیم که هر دو روش پیش‌تر معرفی شده با تعیین معادلات متناسب با آنها می‌توانند در چهارچوب روش‌های مبتنی بر معادله دیفرانسیل تصادفی قرار گیرند.

در فصل چهارم پس از معرفی سه دسته‌ی اصلی مدل‌های پخشی به مقایسه‌ی آنها هم با سایر مدل‌های مولد و هم با یکدیگر پرداختیم. نکته‌ی قابل تاکید در این فصل این است که برای مقایسه‌ی روش‌های معرفی شده تنها به مقالات به کار برده شده در سراسر گزارش اتکا شده است و با توجه به استقبال روزافزون از توسعه و به کارگیری این روش‌ها ممکن است به نتایج و بهبودهای بهترو یا رفع مشکلات مطرح شده پرداخته شده باشد که شرح آنها در این گزارش نیامده است.

۶-۲ کارهای آتی

با توجه به مطالعات انجام شده و با توجه به نوظهور بودن این دسته از مدل‌ها قابلیت بهبود و توسعه برای آنها در انواعی از جنبه ها وجود دارد که در ادامه آنها را نام می‌بریم.

- تئوری

از آنجاییکه مدل‌های پخشی اساسا بر پایه‌ی روابط ریاضی توسعه داده شده اند، امکان بهبود آنها از جنبه ی ریاضی و تئوری وجود دارد. به عنوان مثال ارائه‌ی فرمولاسیون جدیدی برای تخریب داده‌ها، استفاده از انواعی از حل‌کننده‌های معادله دیفرانسیل و بررسی تاثیرات آنها و همچنین به کارگیری مباحثی از فیزیک ترمودینامیک از جمله روش‌های قابل اعمال هستند.

- معماری

معماری فعلی تمام مدل‌های پخشی از هسته‌ی شبکه‌ی یو بهره می‌برد. با وجود عملکرد فوق‌العاده‌ی این شبکه‌ها همچنان مشکلاتی در استفاده از آنها در بعضی از کاربردهای خاص وجود دارد و علاوه بر این تعداد پارامترهای این شبکه‌ها آنها را از لحاظ محاسباتی بسیار سنگین می‌کند. پیاده سازی و استفاده از سایر شبکه‌ها برای شبکه‌ی نویز زدا می‌تواند پیشرفت قابل توجهی در به کارگیری این مدل‌ها در همه‌ی کاربردها فراهم آورد.

- داده

آموزش و توسعه‌ی مدل‌های پخشی با میزان داده‌ی کارا یک چالش باز در بحث توسعه‌ی مدل‌های پخشی است. این مدل‌ها برای تولید داده‌ی با کیفیت نیاز به دریافت داده‌ی با کیفیت دارند و از این رو کار روی یادگیری محدود و یادگیری با داده‌ی کم کیفیت می تواند یک رویکرد قابل پیگیری باشد.

- کاربرد

همانطور که دراین گزارش نیز مشاهده شد، اکثر توسعه‌های مدل‌های پخشی بر روی داده های تصویری انجام شده است. اما تحقیقات متعددی روی پیاده سازی و تطبیق این مدل‌ها در سایرحوزه‌های کاربرد مانند صوت، متن، یادگیری تقویتی و گراف نیز در حال انجام است. تطبیق مدل‌های پخشی با انواع اشکال داده گونه‌ای دیگر از مسائلی است که می‌تواند مورد بررسی و تغییر قرار گیرد.

کتاب‌نامه

[1] Anderson, Brian DO. Reverse-time diffusion equation models. Stochastic Processes and their Applications, 12(3):313–326, 1982.

[2] Feller, William. Retracted chapter: On the theory of stochastic processes, with particular reference to applications. In Selected Papers I, pages 769–798. Springer, 2015.

[3] Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. Advances in neural information processing systems, 27, 2014.

[4] Henighan, Tom, Kaplan, Jared, Katz, Mor, Chen, Mark, Hesse, Christopher, Jackson, Jacob, Jun, Heewoo, Brown, Tom B, Dhariwal, Prafulla, Gray, Scott, et al. Scaling laws for autoregressive generative modeling. arXiv preprint arXiv:2010.14701, 2020.

[5] Heusel, Martin, Ramsauer, Hubert, Unterthiner, Thomas, Nessler, Bernhard, and Hochreiter, Sepp. Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems, 30, 2017.

[6] Ho, Jonathan, Jain, Ajay, and Abbeel, Pieter. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020.

[7] Hyvärinen, Aapo and Dayan, Peter. Estimation of non-normalized statistical models by score matching. Journal of Machine Learning Research, 6(4), 2005.

[8] Kingma, Diederik P, Welling, Max, et al. An introduction to variational autoencoders. Foundations and Trends® in Machine Learning, 12(4):307–392, 2019.

[9] Kreis, Karsten, Gao, Ruiqi, and Vahdat, Arash. Denoising diffusion-based generative modeling: Foundations and applications, 2022.

[10] Nichol, Alexander Quinn and Dhariwal, Prafulla. Improved denoising diffusion probabilistic models. In International Conference on Machine Learning, pages 8162–8171. PMLR, 2021.

[11] Razavi, Ali, Van den Oord, Aaron, and Vinyals, Oriol. Generating diverse high-fidelity images with vq-vae-2. Advances in neural information processing systems, 32, 2019.

[12] Ronneberger, Olaf, Fischer, Philipp, and Brox, Thomas. U-net: Convolutional networks for biomedical image segmentation. In Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18, pages 234–241. Springer, 2015.

[13] Salimans, Tim, Karpathy, Andrej, Chen, Xi, and Kingma, Diederik P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. arXiv preprint arXiv:1701.05517, 2017.

[14] Song, Yang and Ermon, Stefano. Generative modeling by estimating gradients of the data distribution. Advances in neural information processing systems, 32, 2019.

[15] Song, Yang and Ermon, Stefano. Improved techniques for training score-based generative models. Advances in neural information processing systems, 33:12438–12448, 2020.

[16] Song, Yang, Sohl-Dickstein, Jascha, Kingma, Diederik P, Kumar, Abhishek, Ermon, Stefano, and Poole, Ben. Score-based generative modeling through stochastic differential equations. arXiv preprint arXiv:2011.13456, 2020.

[17] Vincent, Pascal. A connection between score matching and denoising autoencoders. Neural computation, 23(7):1661–1674, 2011.

[18] Wang, Zhendong, Zheng, Huangjie, He, Pengcheng, Chen, Weizhu, and Zhou, Mingyuan. Diffusion-gan: Training gans with diffusion. arXiv preprint arXiv:2206.02262, 2022.

[19] Xiao, Zhisheng, Kreis, Karsten, and Vahdat, Arash. Tackling the generative learning trilemma with denoising diffusion gans. arXiv preprint arXiv:2112.07804, 2021.

[20] Yang, Ling, Zhang, Zhilong, Song, Yang, Hong, Shenda, Xu, Runsheng, Zhao, Yue, Shao, Yingxia, Zhang, Wentao, Cui, Bin, and Yang, Ming-Hsuan. Diffusion models: A comprehensive survey of methods and applications. arXiv preprint arXiv:2209.00796, 2022.

واژه‌نامه‌ی فارسی به انگلیسی

ا

اتصالات پرشی . . . . . Skip connections

انفجار واریانس . . . . Variance exploding

ب

بلوک باقی‌مانده‌ای . . . . . Residual block

ت

تابع امتیاز . . . . . Score function

تابکاری شبیه‌سازی شده . . . . Simulated annealing

ترکیب . . . . . Mixing

ترمیم تصویر . . . . . Image inpainting

تعبیه . . . . . Embedding

تعیین یکتا . . . . . Unique identifiability

تطبیق امتیاز . . . . . Score matching

تطبیق امتیاز نویززدا . . . Denoising score matching

تمایزگر . . . . . Discriminatior

توزیع اولیه . . . . . Prior distribution

تولید کننده . . . . . Generator

ج

جریان‌های نرمال‌ساز Normalizing flows

ح

حد پایین مشاهده Evidence lower bound

حد تغییراتی . . . . . Variational bound

حل‌کننده‌ی اویلر ماریاما Euler-maruyama solver

خ

خودکدگذارهای تغییراتی . . . Variational autoencoders

خودکدگذارهای تغییراتی سلسله مراتبی Hierarchical variational autoencoders

د

درست‌نمایی . . . . . Likelihood

دینامیک لانژوین . . Langevin synamics

دینامیک لانژوین تابکاری شده . Annealed langevin dynamics

دیورجنس KL . . . . . KL-divergence

ر

رنگ‌آمیزی . . . . . Colorization

ش

شبکه‌های مولد تقابلی . . . . . Generative adversarial networks

شبکه‌ی یو . . . . . U-net

ض

ضریب پخش . . . . Diffusion coefficient

ضریب رانش . . . . . Drift coefficient

غ

غیرقابل محاسبه . . . . . Intractable

ف

فاصله‌ی اکتسابی فرچت . . . . . Frechet inception distance

فرآیند قطعی . . . Deterministic process

فرآیند رو به جلو . . . . . Forward process

فرآیند وینر استاندارد . . Standard wiener process

فضای پشتیبان . . . . . Support

ق

قطعه‌بندی معنایی . . . . . Semantic segmentation

ک

کدگشا . . . . . Decoder

کدگذار . . . . . Encoder

ل

لگاریتم درست‌نمایی . . . Log-likelihood

م

مد . . . . . Mode

مدل پخشی نویززدای احتمالاتی Denoising diffusion probabilistic models

مدل‌های مولد مبتنی بر امتیاز Score-based generative models

مدل‌های متغیر پنهان . . Latent variable models

مدل پخشی . . . . . Diffusion models

مسیر روبه جلو . . . . Forward trajectory

مشکل سه‌گانه‌ی یادگیری مولدGenerative learning trilemma

معادلات دیفرانسیل ساده . . . . . Ordinary differential equations

میانگین پسین . . . . . Posterior mean

میدان برداری . . . . . Vector field

ن

ناهنجاری . . . . . Anomaly

نرخ پخش . . . . . Diffusion rate

نقطه‌ی زینی . . . . . Saddle point

نمونه‌برداری اهمیت . . . . . Importance sampling

ه

هسته . . . . . Kernel

هسته‌ی دوجمله‌ای . . . Binomial kernel

هسته‌ی گذر . . . . . Transition kernel

واژه‌نامه‌ی انگلیسی به فارسی

A

Anomaly . . . . . ناهنجاری

Annealed langevin dynamics . دینامیک لانژوین تابکاری شده

B

Binomial kernel . . . هسته‌ی دوجمله‌ای

C

Colorization . . . . . رنگ‌آمیزی

D

Decoder . . . . . کدگشا

Denoising score matching . . . تطبیق امتیاز نویززدا

Denoising diffusion probabilistic models مدل پخشی نویززدای احتمالاتی

Deterministic process . . . فرآیند قطعی

Diffusion coefficient . . . . ضریب پخش

Diffusion models . . . . . مدل پخشی

Diffusion rate . . . . . نرخ پخش

Discriminatior . . . . . تمایزگر

Drift coefficient . . . . . ضریب رانش

E

Encoder . . . . . کدگذار

Embedding . . . . . تعبیه

Evidence lower bound حد پایین مشاهده

Euler-maruyama solver حل‌کننده‌ی اویلر ماریاما

F

Forward process . . . . . فرآیند رو به جلو

Forward trajectory . . . . مسیر روبه جلو

مدل‌های متغیر پنهان . . Latent variable models

درست‌نمایی . . . . . Likelihood

لگاریتم درست‌نمایی . . . Log-likelihood

M

ترکیب . . . . . Mixing

مد . . . . . Mode

N

جریان‌های نرمال‌ساز Normalizing flows

O

معادلات دیفرانسیل ساده . . . . . Ordinary differential equations

P

میانگین پسین . . . . . Posterior mean

توزیع اولیه . . . . . Prior distribution

R

بلوک باقی‌مانده‌ای . . . . . Residual block

S

نقطه‌ی زینی . . . . . Saddle point

مدل‌های مولد مبتنی بر امتیاز Score-based generative models

تابع امتیاز . . . . . Score function

فاصله‌ی اکتسابی فرچت . . . . . Frechet inception distance

G

شبکه‌های مولد تقابلی . . . . . Generative adversarial networks

مشکل سه‌گانه‌ی یادگیری مولد Generative learning trilemma

تولید کننده . . . . . Generator

H

خودکدگذارهای تغییراتی سلسله مراتبی Hierarchical variational autoencoders

I

ترمیم تصویر . . . . . Image inpainting

نمونه‌برداری اهمیت . . . . . Importance sampling

غیرقابل محاسبه . . . . . Intractable

K

هسته . . . . . Kernel

دیورجنس KL . . . . . KL-divergence

L

دینامیک لانژوین . . Langevin synamics

هسته‌ی گذر . . . . . Transition kernel

U

شبکه‌ی یو . . . . . U-net

تعیین یکتا . . . . . Unique identifiability

V

انفجار واریانس . . . . Variance exploding

خودکدگذارهای تغییراتی . . . Variational autoencoders

حد تغییراتی . . . . . Variational bound

میدان برداری . . . . . Vector field

تطبیق امتیاز . . . . . Score matching

قطعه‌بندی معنایی . . . . . Semantic segmentation

یادگیری نیمه نظارتی . Semi-supervised learning

تابکاری شبیه‌سازی شده . . . . Simulated annealing

اتصالات پرشی . . . . . Skip connections

فرآیند وینر استاندارد . . Standard wiener process

فضای پشتیبان . . . . . Support

T