تمرین دوم

نام درس: یادگیری عمیق

استاد درس: دکتر محمدرضا محمدی

نام: محمد حقیقت

شماره دانشجویی: 403722042

گرایش: هوش مصنوعی

دانشکده: مهندسی کامپیوتر

نیم سال دوم 1404-1403

سوال اول

آ)

مدل‌های امتیازی (Score-Based Models)

ایده اصلی این مدل‌ها، یادگیری یک "تابع امتیاز" (Score Function) برای داده‌های واقعی (مثلاً تصاویر) است.

تابع امتیاز چیست؟ تابع امتیاز در هر نقطه (مثلاً برای هر تصویر)، یک بردار را مشخص می‌کند که جهتِ بیشترین افزایش احتمال را نشان می‌دهد. به عبارت ساده‌تر، این تابع مانند یک راهنما عمل می‌کند و به ما می‌گوید برای اینکه یک تصویر نویزی یا تصادفی، به یک تصویر واقعی و باکیفیت نزدیک‌تر شود، باید در چه جهتی تغییر کند. این تابع از نظر ریاضی، گرادیان لگاریتم چگالی احتمال داده است($\nabla_x \log p(x)$).

مشکل اصلی: یادگیری این تابع در مناطقی که داده واقعی وجود ندارد (مثلاً در فضای بین تصاویر مختلف) بسیار دشوار است. اگر فرآیند تولید نمونه از یک نویز خالص شروع شود، در این مناطق "خالی" سرگردان می‌شود و نمی‌تواند به یک نمونه باکیفیت همگرا شود.

راه‌حل: شرطی کردن با نویز (Noise-Conditioning)

برای حل مشکل بالا، این مقاله و کارهای مشابه یک ایده هوشمندانه را به کار می‌گیرند:

به جای یادگیری یک تابع امتیاز برای تصاویر تمیز، ما چندین نسخه نویزی از داده‌ها با مقادیر مختلف نویز (از زیاد تا کم) ایجاد می‌کنیم و یک شبکه عصبی واحد را آموزش می‌دهیم تا تابع امتیاز را برای تمام این سطوح نویز یاد بگیرد.

چرا این کار مفید است؟

نویز زیاد: وقتی نویز زیادی به تصاویر اضافه می‌کنیم، فضای خالی بین داده‌ها پر می‌شود. این کار به مدل کمک می‌کند تا ساختار کلی و ویژگی‌های درشت (coarse features) تصاویر را یاد بگیرد.

نویز کم: با افزودن نویز کم، مدل می‌تواند جزئیات دقیق و ویژگی‌های ظریف (fine-grained features) را یاد بگیرد.

در نتیجه، یک شبکه عصبی (که در مقاله NCSN نامیده می‌شود) آموزش داده می‌شود که دو ورودی می‌گیرد: یک تصویر نویزی و مقدار نویزی که به آن اضافه شده است (σ). خروجی آن، همان بردار راهنما (تابع امتیاز) برای آن سطح نویز خاص است.

Langevin Dynamics چیست؟

Langevin Dynamics یک الگوریتم تکرارشونده برای نمونه‌برداری از یک توزیع احتمالاتی است، به شرطی که تابع امتیاز آن را در اختیار داشته باشیم. این فرآیند به صورت زیر عمل می‌کند:

از یک نقطه کاملاً تصادفی (مثلاً یک تصویر نویز خالص) شروع می‌کنیم.

در هر مرحله، یک قدم کوچک در جهت تابع امتیاز برمی‌داریم (یعنی در جهتی که احتمال را افزایش می‌دهد).

سپس مقدار کمی نویز تصادفی به آن اضافه می‌کنیم. این نویز از گیر افتادن در بهینه‌های محلی جلوگیری کرده و به کاوش بهتر فضا کمک می‌کند.

با تکرار این فرآیند، نمونه به تدریج به سمت مناطق با احتمال بالا (یعنی تصاویر واقعی) حرکت می‌کند و در نهایت به یک نمونه از توزیع داده اصلی تبدیل می‌شود.

کاربرد Langevin Dynamics در مدل‌های شرطی‌شده با نویز (نقش کلیدی):

اینجاست که همه چیز به هم متصل می‌شود. در این مدل‌ها از نسخه‌ای به نام "دینامیک لانژون بازپخت شده" (Annealed Langevin Dynamics) استفاده می‌شود:

شروع با نویز زیاد: فرآیند تولید نمونه از یک تصویر نویز خالص آغاز می‌شود. سپس، دینامیک لانژون را با استفاده از تابع امتیازی که برای بالاترین سطح نویز (σ_max) آموزش دیده، اجرا می‌کنیم. این کار به سرعت ساختار کلی و کلیات تصویر را شکل می‌دهد.

کاهش تدریجی نویز: خروجی مرحله قبل را به عنوان ورودی مرحله بعد در نظر می‌گیریم. این بار دینامیک لانژون را با تابع امتیازی که برای سطح نویز کمی پایین‌تر (σ_i) آموزش دیده اجرا می‌کنیم. این کار باعث پالایش تصویر و اضافه شدن جزئیات بیشتر می‌شود.

تکرار تا رسیدن به نویز کم: این فرآیند کاهش تدریجی نویز (مانند فرآیند بازپخت یا Annealing) ادامه می‌یابد تا به پایین‌ترین سطح نویز (σ_min) برسیم.

در پایان، نمونه‌ای که از نویز خالص شروع شده بود، به تدریج از یک تصویر تار و بی‌معنی به یک تصویر باکیفیت و پر از جزئیات تبدیل می‌شود. این روش به مدل اجازه می‌دهد تا بر چالش تولید نمونه در رزولوشن‌های بالا غلبه کند و تصاویری با کیفیتی مشابه مدل‌های GAN تولید نماید.

دینامیک لانژون، موتورِ حرکتی است که با راهنماییِ تابع امتیازِ شرطی‌شده با نویز، یک نمونه تصادفی را به تدریج و مرحله به مرحله به یک نمونه باکیفیت تبدیل می‌کند.

ب)

مقابله با چالش "چگالی پایین داده" (Low Data Density)

مشکل چیست؟

فضای تصاویر بسیار بزرگ و "خالی" است. اگر تصاویر موجود در دیتاست خود را مانند جزیره‌هایی در یک اقیانوس بی‌کران در نظر بگیریم، اکثر نقاط این اقیانوس خالی از داده هستند. مدل‌های تولیدگر قدیمی (مانند GANها) در یادگیری اینکه در این فضاهای خالی چه چیزی باید وجود داشته باشد، با مشکل مواجه می‌شوند. آن‌ها ممکن است فقط روی خود جزیره‌ها تمرکز کنند و نتوانند به‌خوبی بین آن‌ها پل بزنند. به این مشکل "چگالی پایین داده" می‌گویند.

راه حل مدل‌های دیفیوژن:

مدل‌های دیفیوژن به جای تلاش برای پریدن از یک جزیره به جزیره دیگر، کل اقیانوس را با "مه" پر می‌کنند!

فرآیند رو به جلو (Forward Process): مدل به صورت عمدی و در طی مراحل متوالی، به تصاویر واقعی دیتاست (جزیره‌ها) نویز گوسی اضافه می‌کند. با هر گام، تصویر کمی نویزی‌تر می‌شود تا در نهایت به نویز خالص (یک توده ابر یا مه) تبدیل شود. این فرآیند باعث می‌شود که تمام فضای بین جزیره‌ها با نسخه‌های نویزی از تصاویر پر شود. دیگر هیچ فضای خالی‌ای وجود ندارد!

فرآیند رو به عقب (Reverse Process): حالا وظیفه اصلی مدل شروع می‌شود: یادگیری مسیریابی در این مه. مدل یاد می‌گیرد که از هر نقطه‌ای در این فضای نویزی، چگونه یک گام کوچک به سمت "واضح‌تر شدن" و نزدیک شدن به یک تصویر واقعی بردارد. در واقع، مدل "میدان گرادیان" (یا به قول مقاله NCSN، Score Function) را در کل فضا یاد می‌گیرد. این میدان مانند یک قطب‌نما عمل می‌کند که همیشه جهت "داده‌های با چگالی بالا" (جزیره‌ها) را نشان می‌دهد.

نتیجه: از آنجایی که مدل مسیر را از هر نقطه‌ای در فضا بلد است و نه فقط از روی داده‌های اصلی، مشکل فضای خالی و کم‌چگالی به طور کامل حل می‌شود. مدل به جای یک جهش بزرگ، یک سفر تدریجی و هموار را از میان مه به سمت خشکی طی می‌کند.

مقابله با چالش "چندفرضیه‌ای" (Multi-hypothesis)

مشکل چیست؟

برای یک ورودی یا شرط خاص، ممکن است چندین خروجی معتبر و متفاوت وجود داشته باشد. برای مثال، در تولید بدون شرط، از یک نقطه شروع تصادفی، بی‌نهایت تصویر معتبر می‌توان تولید کرد. بسیاری از مدل‌ها یا به یک خروجی واحد همگرا می‌شوند (deterministic) یا دچار "فروپاشی مُد" (Mode Collapse) شده و تنها تنوع محدودی از خروجی‌ها را تولید می‌کنند.

راه حل مدل‌های دیفیوژن:

ماهیت فرآیند تولید (رو به عقب) در مدل‌های دیفیوژن ذاتاً تصادفی (Stochastic) است.

گام‌های نویزدایی تصادفی: وقتی مدل در هر مرحله یک گام به سمت واضح‌تر شدن برمی‌دارد (مانند تراشیدن سنگ توسط مجسمه‌ساز)، این کار را کاملاً قطعی انجام نمی‌دهد. بلکه پس از برداشتن گام اصلی، مقدار بسیار کمی نویز جدید اضافه می‌کند (در الگوریتم سمپلینگ مقاله DDPM با σz+ نشان داده شده است). این مانند لرزش جزئی دست مجسمه‌ساز است که باعث می‌شود هر ضربه کمی متفاوت از دیگری باشد.

ایجاد مسیرهای انشعابی: این عنصر تصادفی در هر یک از صدها یا هزاران گام تکرار می‌شود. این انحرافات کوچک در طول مسیر با هم جمع شده و باعث می‌شوند که حتی اگر از یک نقطه شروع کاملاً یکسان (همان توده نویز اولیه) شروع کنیم، مدل مسیرهای کمی متفاوتی را طی کند و در نهایت به خروجی‌های کاملاً متفاوتی (مثلاً چهره‌های مختلف، گربه‌های متفاوت) برسد.

نتیجه: این فرآیند تصادفی تضمین می‌کند که مدل می‌تواند کل فضای احتمالات را کاوش کند و تنوع گسترده‌ای از نمونه‌ها را تولید نماید. این رویکرد به طور طبیعی از فروپاشی مُد جلوگیری کرده و چالش چندفرضیه‌ای را به زیبایی حل می‌کند. این مکانیزم ارتباط نزدیکی با نمونه‌برداری به روش "دینامیک لانژون" (Langevin Dynamics) دارد که در مقاله NCSNv2 به آن پرداخته شده است.

ج)

ایده اصلی دیفیوژن این است که یادگیری یک تبدیل بسیار پیچیده (از نویز خالص به یک تصویر باکیفیت) دشوار است. در عوض، مدل فرآیند را به صدها یا هزاران گام کوچک و ساده تقسیم می‌کند. هر گام، یک وظیفه بسیار جزئی و قابل یادگیری دارد: "حذف مقدار کمی نویز".

تعداد گام‌ها (T) و اندازه نویز در هر گام (β_t) مستقیماً روی کیفیت، سرعت و پایداری این فرآیند تأثیر می‌گذارند.

تعداد گام‌های کم، اما با اندازه نویز بزرگ (مثلاً ۳ گام)

فرض کنید به جای ۱۰۰۰ گام کوچک، می‌خواهیم با ۳ گام بزرگ از نویز به تصویر برسیم.

مزایا:

سرعت بسیار بالا در نمونه‌برداری: مزیت اصلی و بارز این روش، سرعت است. به جای هزاران بار فراخوانی شبکه عصبی، تنها ۳ بار آن را فراخوانی می‌کنیم. این امر تولید نمونه را به شدت سریع‌تر می‌کند.

معایب:

وظیفه یادگیری بسیار دشوار: این بزرگترین عیب است. در هر گام، مدل باید مقدار بسیار زیادی نویز را حذف کند. این دیگر یک "گام کوچک" نیست، بلکه یک "جهش بزرگ" است. شبکه عصبی باید یاد بگیرد که چگونه از یک تصویر بسیار نویزی، یک تصویر تقریباً تمیز بسازد. این کار به مراتب سخت‌تر از نویزدایی جزئی است و به مدل بسیار بزرگتر و پیچیده‌تری نیاز دارد.

افت شدید کیفیت و جزئیات: از آنجایی که وظیفه بسیار دشوار است، مدل به احتمال زیاد خطاهای بزرگی مرتکب خواهد شد. این خطاها در گام‌های بعدی تشدید شده و نتیجه نهایی احتمالاً تار، فاقد جزئیات دقیق و دارای آرتیفکت‌های (artifacts) زیاد خواهد بود. جزئیات ظریف مانند بافت پوست یا مو در چنین جهش‌های بزرگی از بین می‌روند.

ناپایداری در آموزش: آموزش مدلی که باید چنین وظیفه سنگینی را انجام دهد، بسیار ناپایدار است. مدل ممکن است به سختی همگرا شود یا اصلاً یاد نگیرد.

کاهش تنوع: مسیر تولید، تصادفی بودن خود را از دست می‌دهد. وقتی تنها چند گام بزرگ وجود دارد، تزریق نویز تصادفی در هر گام تأثیر کمتری بر مسیر کلی دارد و ممکن است مدل به تولید نمونه‌های مشابه تمایل پیدا کند (کاهش تنوع).

مثال: این مانند آن است که یک مجسمه‌ساز بخواهد با سه ضربه پتک، یک توده سنگ را به یک مجسمه کامل تبدیل کند. نتیجه احتمالاً یک سنگ شکسته خواهد بود تا یک اثر هنری!

تعداد گام‌های زیاد، اما با اندازه نویز کوچک (مثلاً ۸۰ گام یا بیشتر)

این رویکرد استاندارد در مدل‌های دیفیوژن مدرن با کیفیت بالا است (که اغلب از ۱۰۰۰ گام استفاده می‌کنند).

مزایا:

کیفیت و دقت بسیار بالا: این بزرگترین مزیت است. از آنجایی که در هر گام مقدار بسیار کمی نویز حذف می‌شود، وظیفه یادگیری برای شبکه بسیار ساده است. مدل می‌تواند با دقت فوق‌العاده‌ای این کار را انجام دهد. این امر به حفظ و بازسازی جزئیات بسیار ظریف کمک می‌کند و منجر به تولید نمونه‌هایی با کیفیت خیره‌کننده می‌شود.

آموزش پایدارتر: چون هر گام ساده است، فرآیند آموزش پایدارتر است و مدل راحت‌تر همگرا می‌شود.

تنوع بالا در نمونه‌ها: تعداد زیاد گام‌ها به این معناست که تزریق‌های کوچک نویز تصادفی در هر مرحله فرصت کافی برای انباشته شدن و ایجاد انحرافات معنادار در مسیر را دارند. این امر منجر به تولید نمونه‌هایی بسیار متنوع می‌شود.

معایب:

سرعت پایین در نمونه‌برداری: این عیب اصلی است. برای تولید یک نمونه، باید شبکه عصبی را صدها یا هزاران بار فراخوانی کرد که فرآیند را بسیار کند می‌کند. (البته تکنیک‌هایی مانند DDIM برای سرعت بخشیدن به این فرآیند توسعه یافته‌اند).

مثال: مجسمه‌ساز با هزاران ضربه ظریف و کوچک، سنگ را می‌تراشد. این کار زمان‌بر است، اما نتیجه نهایی یک شاهکار با جزئیات بی‌نقص خواهد بود.

سوال دوم

الف)

صورت سؤال:

فرض کنید:

$$p_{\text{data}} \sim \mathcal{N}(\theta_0, \epsilon^2), \quad p_g \sim \mathcal{N}(\theta, \epsilon^2)$$

یعنی هر دو توزیع نرمال (Gaussian) هستند، با واریانس (انحراف معیار به توان ۲) برابر ϵ^2، ولی میانگین‌های متفاوت دارند:

یکی میانگین θ_0

یکی دیگر میانگین θ

میخواهیم ثابت کنیم که:

$$D_{\text{KL}}(P|Q) = \frac{(\mu_1 - \mu_2)^2}{2\sigma^2}$$

مرحله 1: فرمول KL بین دو توزیع نرمال

اگر:

$$P = \mathcal{N}(\mu_1, \sigma^2)$$

$$Q = \mathcal{N}(\mu_2, \sigma^2)$$

و واریانس‌ها برابرند (همان‌طور که اینجا داریم : ϵ^2 فرمول KL ساده می‌شود:

$$D_{\text{KL}}(P|Q) = \frac{(\mu_1 - \mu_2)^2}{2\sigma^2}$$

مرحله 2: جایگذاری در این مسئله

در اینجا:

$$\mu_1 = \theta$$

$$\mu_2 = \theta_0$$

$$\sigma^2 = \epsilon^2$$

پس:

$$D_{\mathrm{KL}}(p_g|p_{\mathrm{data}}) = \frac{(\theta - \theta_0)^2}{2\epsilon^2}$$

ب)

مرحله 1: یادآوری فرمول KL (از سؤال قبلی)

ما داریم:

$$D_{\mathrm{KL}}(p_g|p_{\mathrm{data}}) = \frac{(\theta - \theta_0)^2}{2\epsilon^2}$$

مرحله 2: گرفتن حد وقتی $\epsilon \to 0$

از این فرمول:

$$\lim_{\epsilon \to 0} \frac{(\theta - \theta_0)^2}{2\epsilon^2}$$

اگر $\theta \neq \theta_0$، آنگاه صورت ثابت است و مخرج صفر می‌شود.

پس:

$$\lim_{\epsilon \to 0} D_{\mathrm{KL}}\left(p_g|p_{\mathrm{data}}\right) = \infty$$

یعنی:

هر چه انحراف معیار کوچک‌تر شود (توزیع‌ها تیزتر و متمرکزتر شوند)، اگر $\theta \neq \theta_0$، آنگاه KL شدیداً بالا می‌رود.

مرحله 3: مشتق KL نسبت به θ

فرمول KL:

$$D_{\mathrm{KL}} = \frac{(\theta - \theta_0)^2}{2\epsilon^2}$$

مشتق نسبت به θ:

$$\frac{d}{d\theta} D_{\mathrm{KL}} = \frac{1}{2\epsilon^2} \cdot 2(\theta - \theta_0) = \frac{\theta - \theta_0}{\epsilon^2}$$

مرحله 4: حد مشتق وقتی $\epsilon \to 0$

اگر $\theta \neq \theta_0$، آنگاه:

$$\lim_{\epsilon \to 0} \frac{\theta - \theta_0}{\epsilon^2} = \infty$$

(با علامت بسته به اینکه $\theta > \theta_0$ یا $\theta < \theta_0$ باشد)

یعنی:

شیب تابع KL در جهت θ بسیار بزرگ می‌شود، پس بهینه‌سازی ممکن است ناپایدار یا با گرادیان‌های انفجاری مواجه شود.

نتیجه کلی:

اگر $\theta \neq \theta_0$، آنگاه:

$$\lim_{\epsilon \to 0} D_{\mathrm{KL}}\left(p_g | p_{\mathrm{data}}\right) = \infty$$

$$\lim_{\epsilon \to 0} \frac{d}{d\theta} D_{\mathrm{KL}} = \pm\infty$$

یعنی KL هم بسیار حساس می‌شود و هم مشتق آن منفجر می‌شود!

ج)

بله، قطعاً این امر یک مشکل اساسی و شناخته‌شده برای GANهایی است که با این تابع ضرر (معروف به تابع ضرر min-max یا binary cross-entropy) آموزش می‌بینند.

چرا؟

دلیل اصلی این است که بهینه‌سازی این تابع ضرر در GAN، ارتباط مستقیمی با واگرایی جنسن-شنون (Jensen-Shannon Divergence - JSD) دارد که خود آن نیز ارتباط تنگاتنگی با واگرایی KL دارد و از مشکلات مشابهی رنج می‌برد.

بیایید این ارتباط را بررسی کنیم:

1. ارتباط تابع ضرر GAN با JSD:

در مقاله اصلی GAN، نویسندگان ثابت کرده‌اند که اگر تمایزدهنده D (Discriminator) بهینه باشد، بهینه‌سازی مولد G (Generator) با این تابع ضرر، معادل کمینه کردن واگرایی جنسن-شنون بین توزیع داده‌های واقعی (p_data) و توزیع داده‌های تولیدی (p_g) است.

به طور دقیق‌تر:

برای یک مولد G ثابت، تمایزدهنده بهینه D*(x) برابر است با:

D*(x) = p_data(x) / (p_data(x) + p_g(x))

اگر این D*(x) را در تابع هدف کلی GAN جایگذاری کنیم، به رابطه‌ی زیر می‌رسیم:

C(G) = -2 * log(2) + 2 * JSD(p_data || p_g)

این رابطه نشان می‌دهد که آموزش مولد برای کمینه کردن تابع ضرر، معادل کمینه کردن JSD بین دو توزیع است.

2. مشکل JSD (و KL) در عمل:

همانطور که در بخش (ب) دیدیم، وقتی دو توزیع همپوشانی بسیار کمی دارند (که معادل حالت $\varepsilon \to 0$ در مثال ماست)، واگرایی KL به بی‌نهایت میل می‌کند و گرادیان آن یا ناپدید می‌شود (vanishes) یا منفجر می‌شود (explodes).

واگرایی JSD نیز مشکل مشابهی دارد:

اگر تکیه‌گاه (support) دو توزیع p_data و p_g کاملاً از هم جدا باشند، JSD بین آنها یک مقدار ثابت، یعنی log(2)، خواهد بود.

3. اتفاقی که در آموزش GAN می‌افتد:

در ابتدای آموزش GAN، یا زمانی که تمایزدهنده بسیار قوی‌تر از مولد است، توزیع داده‌های تولیدی (p_g) و واقعی (p_data) به راحتی قابل تفکیک هستند (همپوشانی کم یا صفر).

تمایزدهنده (Discriminator) کامل می‌شود: تمایزدهنده به سرعت یاد می‌گیرد که با اطمینان کامل بین داده‌های واقعی و جعلی تمایز قائل شود. یعنی برای داده‌های واقعی، D(x) ≈ 1 و برای داده‌های جعلی، D(x) ≈ 0.

گرادیان مولد ناپدید می‌شود: حالا به تابع ضرر مولد نگاه کنیم. بخش اصلی که مولد سعی در بهینه‌سازی آن دارد (log(D(G(z)))) است (در فرمول سوال، بخش log(1 - D(x))) است که همین رفتار را دارد).

وقتی D(x) برای داده‌های جعلی به 0 نزدیک می‌شود، تابع log(D(x)) به -∞ میل می‌کند. مهم‌تر از آن، شیب (گرادیان) تابع لگاریتم در نزدیکی صفر بسیار کم می‌شود.

در فرمول ارائه شده در سوال، وقتی D(x) -> 0، آنگاه log(1 - D(x)) به log(1) = 0 نزدیک می‌شود. این تابع در این نقطه بسیار "تخت" (flat) است و گرادیان آن نزدیک به صفر است.

نتیجه‌گیری نهایی:

وقتی تمایزدهنده خیلی خوب عمل می‌کند، تابع ضرر مولد صاف می‌شود و گرادیان آن به صفر میل می‌کند. در نتیجه، مولد هیچ سیگنال یا راهنمایی برای بهبود پارامترهای خود (θ) دریافت نمی‌کند و یادگیری متوقف می‌شود. این پدیده به "مشکل گرادیان ناپدیدشونده" (Vanishing Gradient Problem) در GANها معروف است.

بنابراین، نتایجی که در بخش (ب) برای واگرایی KL به دست آمد، مستقیماً به مشکل ناپایداری و توقف یادگیری در GANهای استاندارد ترجمه می‌شود. این دقیقاً همان چالشی است که WGAN با جایگزین کردن این معیار فاصله (JSD/KL) با فاصله واسرشتاین، آن را برطرف می‌کند.

د)

عالی. این سوال به نقطه کلیدی و تفاوت اصلی WGAN با GANهای کلاسیک می‌پردازد.

شرایط بخش (ب) این بود:

$p\_data \sim N(\theta_0, \epsilon^2)$ و $p\_g \sim N(\theta, \epsilon^2)$

$\theta \neq \theta_0$

$\epsilon \to 0$ (انحراف معیار به صفر میل می‌کند)

در این شرایط، توزیع‌ها به توابع دلتای دیراک (نقطه‌ای) در مکان‌های θ و θ_0 تبدیل می‌شوند و هیچ همپوشانی ندارند. بیایید رفتار سه معیار را در این شرایط مقایسه کنیم.

1. واگرایی کولبک-لیبلر (KL Divergence)

همانطور که در بخش (ب) دیدیم:

$$D_{KL}(p_g \,||\, p_{data}) = (\theta - \theta_0)^2 / (2\epsilon^2)$$

مقدار: وقتی $\epsilon \to 0$، مخرج به صفر میل می‌کند و صورت کسر یک عدد ثابت مثبت است.

$$\lim_{(\epsilon \to 0)} D_{KL} = +\infty$$

رفتار: این معیار بسیار "شکننده" یا "خشن" (brittle/harsh) است. به محض اینکه توزیع‌ها همپوشانی نداشته باشند، مقدار آن به بی‌نهایت میل می‌کند و هیچ اطلاعات مفیدی در مورد اینکه "چقدر" از هم دور هستند ارائه نمی‌دهد. گرادیان آن نیز منفجر می‌شود.

2. واگرایی جنسن-شنون (JS Divergence)

واگرایی JS (که تابع هزینه GAN اصلی آن را کمینه می‌کند) به صورت زیر تعریف می‌شود:

$JSD(p \,||\, q) = \frac{1}{2} D_{KL}(p \,||\, m) + \frac{1}{2} D_{KL}(q \,||\, m)$ که در آن $m = \frac{1}{2}(p + q)$ توزیع میانگین است.

مقدار: یک نتیجه شناخته‌شده این است که وقتی دو توزیع p و q هیچ همپوشانی (disjoint supports) نداشته باشند، مقدار واگرایی JS بین آنها یک عدد ثابت است:

$$\lim_{(\epsilon \to 0)} JSD(p_g \,||\, p_{data}) = \log(2)$$

رفتار: گرچه JSD مانند KL به بی‌نهایت میل نمی‌کند، اما به مشکل دیگری برمی‌خورد. از آنجایی که مقدار آن ثابت می‌شود (log 2)، مشتق (گرادیان) آن نسبت به پارامتر θ صفر می‌شود. این به معنای گرادیان ناپدیدشونده (vanishing gradient) است. تابع هزینه یک سیگنال ثابت و بی‌فایده تولید می‌کند که به مولد نمی‌گوید چگونه خود را بهبود بخشد.

3. فاصله واسرشتاین-۱ (Wasserstein-1 Distance)

فاصله واسرشتاین (یا فاصله حمل خاک - Earth-Mover's Distance) به طور شهودی "هزینه" تبدیل یک توزیع به توزیع دیگر را اندازه‌گیری می‌کند. در حالت یک‌بعدی ما:

p_g یک توده خاک با جرم ۱ در نقطه θ است.

p_{data} یک توده خاک با جرم ۱ در نقطه θ_0 است.

کمترین هزینه برای جابجایی توده خاک از θ به θ_0 چقدر است؟ این هزینه برابر است با (مقدار جرم) × (فاصله جابجایی).

مقدار:

$$W(p_g, p_{data}) = |\theta - \theta_0|$$

رفتار: این نتیجه فوق‌العاده است!

مقدار فاصله واسرشتاین یک تابع خطی از فاصله بین میانگین‌هاست. این یک معیار فاصله معقول و معنادار است.

گرادیان آن نسبت به θ (به جز در نقطه $\theta = \theta_0$) همیشه 1 یا -1 است. این یک گرادیان ثابت و کاملاً مفید است که هرگز ناپدید یا منفجر نمی‌شود. این گرادیان همیشه به مولد می‌گوید که پارامتر θ را به سمت θ_0 حرکت دهد.

نتیجه‌گیری نهایی:

در حالی که معیارهای KL و JS در شرایطی که توزیع‌ها همپوشانی ندارند (که در عمل بسیار رایج است) از کار می‌افتند، فاصله واسرشتاین یک تابع هزینه "خوش‌رفتار" (well-behaved) و یک گرادیان معنادار ارائه می‌دهد که به مولد اجازه می‌دهد به طور پیوسته یاد بگیرد و به توزیع داده‌های واقعی همگرا شود.

سوال سوم

الف)

در مدل‌های DDPM، فرآیند پیش‌روی (forward diffusion) داده‌ی اصلی (مثلا تصویر) را به تدریج با اضافه کردن نویز گاوسی خراب می‌کند. هدف نهایی این است که بعد از چند مرحله، داده به نویز خالص برسد.

متغیرها:

- x_{t-1}: داده در مرحله‌ی قبلی از فرآیند.
- x_t: داده‌ی جدید پس از اعمال نویز در مرحله t.
- β_t: مقدار نویز تزریقی در مرحله‌ی t پارامتر کوچکی بین ۰ و ۱.
- $\epsilon \sim N(0,I)$: نویز گاوسی استاندارد.

تفسیر رابطه:

فرمول بالا نشان می‌دهد که چگونه داده‌ی مرحله‌ی قبلی x_{t-1} با وزن $\sqrt{1-\beta t}$ حفظ می‌شود و سپس نویز تصادفی با ضریب $\sqrt{\beta_t}$ به آن افزوده می‌شود.

این به این معناست که در هر مرحله‌ی t، داده کمی از ساختار اصلی خود را از دست داده و نویز بیشتری دریافت می‌کند. به تدریج، با افزایش t، داده به نویز کامل نزدیک‌تر می‌شود.

هدف این مرحله در DDPM:

فرآیند forward، داده‌ی اصلی را به نویز تبدیل می‌کند، و فرآیند reverse (که با یک مدل یادگیری یاد گرفته می‌شود) تلاش می‌کند از نویز دوباره داده را بازسازی کند. بنابراین، فهمیدن این معادله کلید درک چگونگی خراب شدن تدریجی تصویر است.

ب)

برای حل قسمت (ب)، باید نشان دهیم که فرمول اصلی در قسمت (آ):

$$x_t = \sqrt{1-\beta_t} \cdot x_{t-1} + \sqrt{\beta_t} \cdot \epsilon$$

می‌تواند به صورت زیر نوشته شود (با استفاده از ترفند تغییر پارامتر):

$$x_t = \sqrt{\overline{\alpha_t}} \cdot x_0 + \sqrt{1-\overline{\alpha_t}} \cdot \epsilon$$

راه‌حل:

تعریف پارامترها:

ابتدا پارامتر جدید α_t و $\overline{\alpha_t}$ را تعریف می‌کنیم:

$$\alpha_t = 1 - \beta_t$$

$$\overline{\alpha_t} = \prod_{s=1}^{t} \alpha_s = \prod_{s=1}^{t} (1-\beta_s)$$

نشان دهیم که x_t را می‌توان مستقیما بر حسب x_0 (داده اولیه) نوشت:

$$x_t = \sqrt{\overline{\alpha_t}} \cdot x_0 + \sqrt{1-\overline{\alpha_t}} \cdot \epsilon$$

اثبات به صورت بازگشتی (ایده اصلی):

اگر فرمول (آ) را به صورت بازگشتی بنویسیم، خواهیم داشت:

$$x_t = \sqrt{1-\beta_t} \cdot x_{t-1} + \sqrt{\beta_t} \cdot \epsilon$$

و همین‌طور:

$$x_{t-1} = \sqrt{1-\beta_{t-1}} \cdot x_{t-2} + \sqrt{\beta_{t-1}} \cdot \epsilon_{t-1}$$

و آن را در رابطه‌ی اول جایگذاری کنیم:

$$x_t = \sqrt{1-\beta_t}\left(\sqrt{1-\beta_{t-1}} \cdot x_{t-2} + \sqrt{\beta_{t-1}} \cdot \epsilon_{t-1}\right) + \sqrt{\beta_t} \cdot \epsilon_t$$

اگر این فرآیند را به صورت بازگشتی ادامه دهیم، به رابطه‌ای از این نوع می‌رسیم:

$$x_t = \sqrt{\bar{\alpha}_t} \cdot x_0 + \text{ترکیبی خطی از نویزها}$$

که با ساده‌سازی نویزها (و با فرض استقلال نویزهای گاوسی)، می‌توان آن ترکیب را به صورت یک نویز گاوسی معادل نوشت:

$$x_t = \sqrt{\bar{\alpha}_t} \cdot x_0 + \sqrt{1-\bar{\alpha}_t} \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

فرمول جدید ساده‌تر است و مستقیما به ما می‌گوید که اگر بخواهیم از نویز خالص ϵ و تصویر اصلی x_0، داده‌ی نویزی x_t را بسازیم، فقط کافی‌ست از پارامتر میانگین‌گیری تجمعی $\bar{\alpha}_t$ استفاده کنیم.

ج)

برخلاف فرایند روبه‌جلو، ما نمی‌توانیم از $(x_{t-1}|x_t)$ برای معکوس کردن نویز استفاده کنیم، زیرا این توزیع غیرقابل محاسبه (intractable) است.

بنابراین نیاز داریم یک شبکه عصبی $p_\theta(x_{t-1}|x_t)$ را آموزش دهیم تا $q(xt-1|xt)$ را تقریب بزند. تقریب $p_\theta(x_{t-1}|x_t)$ از یک توزیع نرمال پیروی می‌کند و میانگین و واریانس آن به شکل زیر تنظیم می‌شوند:

$$\begin{cases} \mu_\theta(x_t, t) & := \tilde{\mu}_t(x_t, x_0) \\ \Sigma_\theta(x_t, t) & := \tilde{\beta}_t I \end{cases}$$

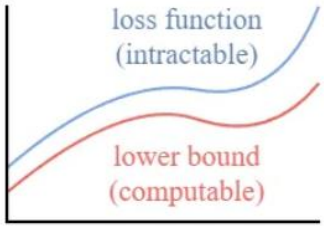
تابع هزینه (Loss Function):

ما می‌توانیم تابع هزینه را به صورت منفی لگاریتم احتمال (Negative Log-Likelihood) تعریف کنیم:

$$\text{Loss} = -\log(p_\theta(x_0))$$

Depends on $x_1, x_2, ..., x_T$
Therefore it is intractable!

این ساختار بسیار مشابه با آنچه در VAE وجود دارد، می‌باشد. به‌جای بهینه‌سازی مستقیم تابع هزینه‌ی غیرقابل محاسبه، می‌توانیم کران پایین واریاتیو (Variational Lower Bound) را بهینه کنیم. با بهینه‌سازی یک کران پایین قابل محاسبه، می‌توانیم به‌طور غیرمستقیم تابع هزینه‌ی اصلی را بهینه کنیم.



با بسط دادن کران پایین واریاتیو، می‌توان آن را با سه جمله زیر نمایش داد:

$$-\log p_\theta(x_0) \le -\log p_\theta(x_0) + D_{\mathrm{KL}}(q(x_{1:T}|x_0) \| p_\theta(x_{1:T}|x_0))$$

$$-\log p_\theta(x_0) \le \mathbb{E}_q\left[\log \frac{q(x_{1:T}|x_0)}{p_\theta(x_{0:T})}\right]$$ Variational Lower Bound

$$-\log p_\theta(x_0) \le \mathbb{E}_q[\underbrace{D_{\mathrm{KL}}(q(x_T|x_0) \| p_\theta(x_T))}_{L_T} + \sum_{t=2}^{T} \underbrace{D_{\mathrm{KL}}(q(x_{t-1}|x_t, x_0) \| p_\theta(x_{t-1}|x_t))}_{L_{t-1}} \underbrace{- \log p_\theta(x_0|x_1)}_{L_0}]$$

① ② ③

• No learnable parameters • Just a Gaussian noise — Constant ⇩ Ignorable

Stepwise denoising term

Reconstruction term

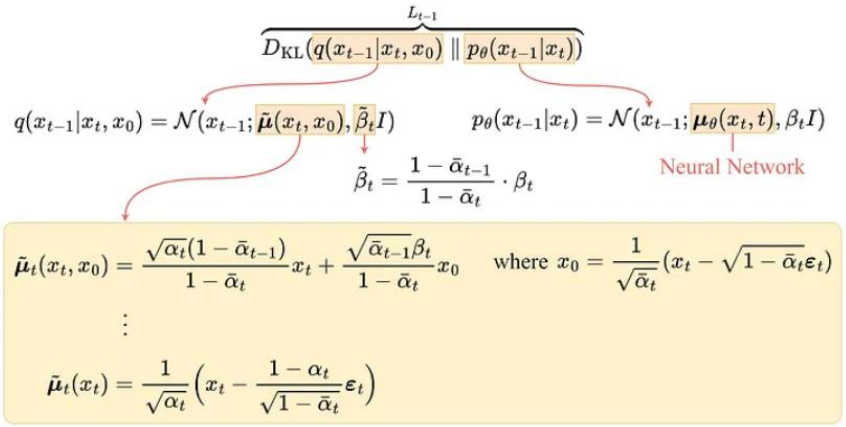
LT: جمله ثابت

از آنجایی که q هیچ پارامتر قابل یادگیری‌ای ندارد و p صرفاً یک توزیع نویز گاوسی است، این جمله در طول آموزش ثابت می‌ماند و می‌توان آن را نادیده گرفت.

Lt-1: جمله کاهش نویز مرحله‌ای

این جمله مرحله‌ی کاهش نویز هدف یعنی q را با تقریب آن توسط p_θ مقایسه می‌کند.

نکته: اگر روی x_0 شرط ببندیم، آنگاه $q(x_{t-1}|x_t,x_0)$ قابل محاسبه (tractable) می‌شود.



پس از یک‌سری استنتاجات، میانگین$\tilde{\mu}$ برای $q(x_{t-1}|x_t,x_0)$ به‌دست می‌آید.

برای تقریب مرحله‌ی کاهش نویز هدف q، تنها کافی است میانگین آن را با یک شبکه عصبی تقریب بزنیم. بنابراین، میانگین تقریبی μ_θ را در همان قالب $\tilde{\mu}$ تنظیم می‌کنیم (با استفاده از یک شبکه‌ی قابل یادگیری ϵ_θ).

$$\tilde{\mu}_t(x_t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_t\right)$$

$$\text{Set} \quad \mu_\theta(x_t,t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t,t)\right)$$

مقایسه بین میانگین هدف و میانگین تقریبی را می‌توان با استفاده از خطای میانگین مربعات(MSE) انجام داد.

$$
\begin{aligned}
L_t &= \mathbb{E}_{x_0,\varepsilon}\left[\frac{1}{2\sigma_t^2}\left\|\tilde{\mu}_t(x_t) - \mu_\theta(x_t,t)\right\|^2\right] \\
&= \mathbb{E}_{x_0,\varepsilon}\left[\frac{1}{2\sigma_t^2}\left\|\frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_t\right) - \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\varepsilon_\theta(x_t,t)\right)\right\|^2\right] \\
&= \mathbb{E}_{x_0,\varepsilon}\left[\frac{(1-\alpha_t)^2}{2\alpha_t(1-\bar{\alpha}_t)\sigma_t^2}\left\|\varepsilon_t - \varepsilon_\theta(x_t,t)\right\|^2\right] \\
&\qquad \text{ignorable} \\
&\qquad \Downarrow \\
L_t^{\text{simple}} &= \mathbb{E}_{t\sim[1,T],x_0,\varepsilon_t}\left[\left\|\varepsilon_t - \varepsilon_\theta(x_t,t)\right\|^2\right]
\end{aligned}
$$

به‌صورت تجربی، نتایج بهتری با نادیده گرفتن ضریب وزنی و مقایسه مستقیم نویز هدف و پیش‌بینی‌شده با MSE حاصل می‌شود.

در نتیجه، برای تقریب مرحله کاهش نویز دلخواه q، تنها کافی است نویز ϵ_t را با استفاده از شبکه عصبی ϵ_θ تقریب بزنیم.

L0: جمله بازسازی

این جمله نشان‌دهنده‌ی خطای بازسازی مرحله‌ی آخر کاهش نویز است و می‌توان آن را در طول آموزش به دلایل زیر نادیده گرفت:

می‌توان آن را با استفاده از همان شبکه عصبی که در L_{t-1} به‌کار رفته، تقریب زد.

نادیده گرفتن آن، کیفیت نمونه‌ها را بهتر می‌کند و پیاده‌سازی را ساده‌تر.

تابع هزینه ساده‌شده

در نتیجه، هدف نهایی آموزش ساده‌شده به صورت زیر است:

$$
\begin{aligned}
x_t &= \sqrt{\bar{a}_t}\, x_0 + \sqrt{1-\bar{a}_t}\, \varepsilon \\
L_{\text{simple}} &= \mathbb{E}_{t,x_0,\varepsilon}\left[\left\|\varepsilon - \varepsilon_\theta(x_t, t)\right\|^2\right]
\end{aligned}
$$

دریافتیم که آموزش مدل‌ها با استفاده از کران پایین واریاتیو واقعی، نسبت به هدف ساده‌شده، طول کد بهتری تولید می‌کند (همان‌طور که انتظار می‌رود)، اما هدف ساده‌شده، بالاترین کیفیت نمونه‌ها را ارائه می‌دهد.

منابع:

https://roysubhradip.hashnode.dev/a-beginners-guide-to-diffusion-models-understanding-the-basics-and-beyond