بسم الله الرحمن الرحیم

دانشگاه صنعتی شریف
دانشکده‌ی مهندسی کامپیوتر

یادگیری عمیق

نیم‌سال دوم ۰۴-۰۳

مدرس: مهدیه سلیمانی

تمرین چهارم | ددلاین تمرین : ۹ خرداد

- برای ارسال هر تمرین تا ساعت ۲۳:۵۹ روز ددلاین فرصت دارید. مهلت تاخیر (مجاز و غیر مجاز) برای این تمرین، ۷ روز است (یعنی حداکثر تاریخ ارسال تمرین ۱۶ خرداد است)
- در هر کدام از سوالات، اگر از منابع خارجی استفاده کرده‌اید باید آن را ذکر کنید. در صورت همفکری با افراد دیگر هم باید نام ایشان را در سوال مورد نظر ذکر نمایید.
- پاسخ تمرین باید ماحصل دانسته‌های خود شما باشد. در صورت رعایت این موضوع، استفاده از ابزارهای هوش مصنوعی با ذکر نحوه و مصداق استفاده بلامانع است.
- پاسخ ارسالی واضح و خوانا باشد. در غیر این صورت ممکن است منجر به از دست دادن نمره شود.
- پاسخ ارسالی باید توسط خود شما نوشته شده باشد. به اسکرین‌شات از منابع یا پاسخ افراد دیگر نمره‌ای تعلق نمی‌گیرد.
- در صورتی که بخشی از سوال‌ها را جای دیگری آپلود کرده و لینک آن را قرار داده باشید، حتما باید تاریخ آپلود مشخص و قابل اتکا باشد.
- محل بارگذاری سوالات نظری و عملی در هر تمرین مجزا خواهد بود. به منظور بارگذاری بایستی تمارین تئوری در یک فایل pdf با نام HW4_[First-Name]_[Last-Name]_[Student-Id].pdf و تمارین عملی نیز در یک فایل مجزای زیپ با نام HW4_[First-Name]_[Last-Name]_[Student-Id].zip بارگذاری شوند.
- در صورت وجود هرگونه ابهام یا مشکل، در کوئرای درس آن مشکل را بیان کنید و از پیغام دادن مستقیم به دستیاران آموزشی خودداری کنید.

بخش نظری (۱۰۰ نمره)

پرسش ۱. طراحی گام به گام VAE (۲۰ نمره)

در این تمرین قصد داریم به صورت گام به گام یک مدل VAE را توسعه دهیم. به دلیل وابستگی بین صورت سوال هر بخش به جواب بخش قبل، در صورت هرگونه ابهام یا سوال می‌توانید در تلگرام ابهامات خود را بر طرف نمایید (ایدی طراح: @Alireza_1197). در ادامه پیش‌فرض‌های مسئله آمده است:

فرضیات:

ما مدل خود را بر پایه‌ی یک معماری **encoder–decoder** با یک **متغیر نهان پیوسته** توسعه خواهیم داد. مؤلفه‌های کلیدی عبارتند از:

- **داده‌ها**: مجموعه‌ای از مشاهدات $\mathcal{D} = \{X^{(i)}\}_{i=1}^{N}$، که در آن هر داده‌ی $X^{(i)} \in \mathbb{R}^n$ است.
- **فضای نهان (Latent Space)**: یک متغیر نهان $Z \in \mathbb{R}^m$ که اطلاعات فشرده‌شده‌ای از ورودی را نمایش می‌دهد.

- **مدل مولد (Decoder)**: توزیع خروجی را یک توزیع گوسی به شرط توابعی از متغیر نهان تعریف می‌کند:

$$p_\theta(X \mid Z) = \mathcal{N}(X \mid \mu_\theta(Z), \Sigma_\theta(Z))$$

که در آن Decoder نگاشت Z به پارامترهای خروجی را انجام می‌دهد:

$$\mu_\theta(Z) = \mathrm{NN}_\alpha(Z), \quad \Sigma_\theta(Z) = \mathrm{diag}(\sigma(\mathrm{NN}_\beta(Z)))$$

- **توزیع پیشین (Prior)**: متغیر پنهان از یک توزیع گاوسی چندمتغیره نمونه گیری می‌شود:

$$p_\theta(Z) = \mathcal{N}(Z \mid \mu_z, \sigma_z^2 I)$$

- **پارامترهای مدل**: مجموعه پارامترهای قابل آموزش $\theta = \{\mu_z, \sigma_z^2, \alpha, \beta\}$، که α و β وزن‌های شبکه Decoder هستند.

گام ۱: برآورد درست‌نمایی (Likelihood Estimation)

در مدل‌های مولد، هدف بیشینه کردن درست‌نمایی داده‌های مشاهده‌شده در توزیع خروجی مدل است.
(الف) یک عبارت برای **درست‌نمایی حاشیه‌ای** $p_\theta(X)$ برای یک نمونه‌ی داده X استخراج کنید.
برای مدل VAE باید تابع درست‌نمایی حاشیه‌ای داده‌های مشاهده‌شده را با استفاده از توزیع نهان مدل محاسبه کنیم:

$$p_\theta(X) = \int p_\theta(X \mid Z)\, p_\theta(Z)\, dZ$$

(ب) **چالش اصلی** در محاسبه یا بهینه‌سازی مستقیم $p_\theta(X)$ در عمل چیست؟

چالش اصلی در محاسبه یا بهینه‌سازی مستقیم $p_\theta(X)$ این است که انتگرال بالا معمولاً قابل محاسبه‌ی دقیق نیست، زیرا فضای Z پر بعد و پیچیده است. به‌همین‌دلیل، محاسبه‌ی مستقیم این مقدار بسیار دشوار یا غیرعملی است.

(ج) از آنجا که محاسبه‌ی $p_\theta(X)$ عملی نیست، یک **تقریب** برای آن پیشنهاد دهید.
راهنمایی: سعی کنید درست‌نمایی حاشیه‌ای را به صورت امید ریاضی بیان کنید.

برای تقریب مقدار $p_\theta(X)$ داریم:

$$p_\theta(X) = \int p_\theta(X \mid Z)\, p_\theta(Z)\, dZ = \mathbb{E}_{Z \sim p_\theta(Z)} \left[p_\theta(X \mid Z) \right]$$

برای تخمین این امید ریاضی می‌توان از روش نمونه‌گیری مونت‌کارلو استفاده کرد:

$$\mathbb{E}_{Z \sim p_\theta(Z)} \left[p_\theta(X \mid Z) \right] \approx \frac{1}{N} \sum_{i=1}^{N} p_\theta(X \mid Z^{(i)}), \quad Z^{(i)} \sim p_\theta(Z)$$

که در آن N تعداد نمونه‌ها است و $Z^{(i)}$ها نمونه‌هایی مستقل از توزیع $p_\theta(Z)$ هستند.

گام ۲: کاهش واریانس با Importance Sampling

چالش اصلی در برآورد $p_\theta(X)$ از طریق نمونه‌گیری، **واریانس بالا** است، زیرا نمونه‌های Z ممکن است از نواحی‌ای بیایند که به داده‌ی مشاهده‌شده‌ی X ارتباطی ندارند.
(الف) برای حل این مسئله، از importance sampling استفاده می‌کنیم. $p_\theta(X)$ را با استفاده از یک توزیع پیشنهادی $q(Z)$ بازنویسی کنید.

برای محاسبه‌ی عبارت زیر:

$$p_\theta(X) = \int p_\theta(X|Z)\, p_\theta(Z)\, dZ$$

در حالت عادی ممکن است نمونه‌گیری از $p_\theta(Z)$ و محاسبه‌ی انتگرال دشوار باشد یا منجر به واریانس بالا شود. برای حل این مشکل، از روش **نمونه‌گیری با اهمیت** (Importance Sampling) استفاده می‌کنیم:

$$p_\theta(X) = \int \frac{q(Z)}{q(Z)}\, p_\theta(X|Z)\, p_\theta(Z)\, dZ = \int q(Z)\, \frac{p_\theta(Z)}{q(Z)}\, p_\theta(X|Z)\, dZ = \mathbb{E}_{Z\sim q(Z)}\left[\frac{p_\theta(Z)}{q(Z)} p_\theta(X \mid Z)\right]$$

حال اگر از $q(Z)$ نمونه‌گیری کنیم، می‌توانیم این انتگرال را به‌صورت میانگین نمونه‌ای (Monte-Carlo) تقریب بزنیم:

$$p_\theta(X) \approx \frac{1}{N}\sum_{i=1}^{N} \frac{p_\theta(Z^{(i)})}{q(Z^{(i)})}\, p_\theta(X|Z^{(i)}) \quad \text{که } Z^{(i)} \sim q(Z)$$

این فرمول، پایه‌ای برای روش‌های یادگیری مولد و تقریب‌های واریشنال است، چرا که توزیع پیشنهادی $q(Z)$ معمولاً طوری انتخاب می‌شود که هم نمونه‌گیری از آن آسان باشد و هم نواحی مرتبط با X را پوشش دهد.

(ب) یک **انتخاب شهودی** برای توزیع پیشنهادی $q(Z)$ معرفی کنید.

برای اینکه تقریب ما از $p_\theta(X)$ واریانس کمتری داشته باشد، باید توزیع پیشنهادی $q(Z)$ طوری انتخاب شود که نمونه‌هایی از نواحی مرتبط با مقدار مشاهده‌شده‌ی X تولید کند. به بیان دیگر، ترجیح می‌دهیم $q(Z)$ بیشتر در نواحی‌ای که $p_\theta(X|Z)\, p_\theta(Z)$ مقدار بالایی دارد، متمرکز باشد. یک انتخاب شهودی مناسب برای $q(Z)$، توزیع پسین $p_\theta(Z|X)$ است. زیرا این توزیع دقیقاً نواحی از فضای Z را مشخص می‌کند که بیشترین نقش را در تولید X دارند:

$$q(Z) = p_\theta(Z|X)$$

گام ۳: از Likelihood به ELBO

برای توجیه انتخاب توزیع پیشنهادی (Proposal Distribution)، دوباره به تخمین تابع هدف $p_\theta(X)$ بازمی‌گردیم.
(الف) چرا در یادگیری ماشین معمولاً با $\log p_\theta(X)$ کار می‌کنیم و نه با خود $p_\theta(X)$؟

در یادگیری ماشین، به جای تابع درست‌نمایی $p_\theta(X)$، اغلب با لگاریتم آن یعنی $\log p_\theta(X)$ کار می‌کنیم. دلایل این انتخاب عبارت‌اند از:

۱. **پایداری عددی:** مقدار تابع درست‌نمایی برای داده‌های پیچیده بسیار کوچک است (مثلاً در حد 10^{-100}) و ضرب این مقادیر برای چند داده باعث underflow می‌شود. لگاریتم گرفتن این مقادیر آن‌ها را در بازه‌ی عددی قابل پردازش نگه می‌دارد.

۲. **سادگی محاسبات مشتق‌گیری:** مشتق لگاریتم تابع احتمال ساده‌تر است:

$$\frac{d}{d\theta}\log p_\theta(X) = \frac{1}{p_\theta(X)}\frac{d}{d\theta}p_\theta(X)$$

۳. **هم‌ارزی با بیشینه‌سازی درست‌نمایی (MLE):** چون لگاریتم یک تابع صعودی است، بیشینه کردن $\log p_\theta(X)$ معادل بیشینه کردن خود $p_\theta(X)$ است.

(ب) چرا نمی‌توانیم $\log p_\theta(X)$ را مستقیماً برآورد کنیم؟ با استفاده از Jensen's inequality یک **کران پایین** برای آن استخراج کنید (این کران به عنوان Evidence Lower Bound یا ELBO شناخته می‌شود).

چون نمی‌توانیم مستقیما لگاریتم امید ریاضی را محاسبه کنیم، برای رفع این مشکل، یک توزیع پیشنهادی $q(Z)$ را وارد می‌کنیم و از نابرابری ینسن استفاده می‌کنیم:

$$\log p_\theta(X) = \log \int p_\theta(X,Z)\,dZ = \log \sum_Z p_\theta(X,Z)\frac{q(Z)}{q(Z)} \geq \sum_Z q(Z)\log\left(\frac{p_\theta(X,Z)}{q(Z)}\right)$$

$$= \sum_Z q(Z)\log\left(\frac{p_\theta(X \mid Z)p_\theta(Z)}{q(Z)}\right) = \sum_z q(Z)\left[\log p_\theta(X \mid Z) + \log p_\theta(Z) - \log q(Z)\right]$$

$$= \sum_Z q(Z)\log p_\theta(x \mid Z) + \sum_Z q(Z)\log p_\theta(Z) - \sum_Z q(Z)\log q(Z)$$

$$= \mathbb{E}_{q(Z)}[\log p_\theta(X \mid Z)] - \mathrm{KL}(q(Z) \parallel p_\theta(Z))$$

که به آن **کران پایین شواهدی (ELBO)** می‌گویند:

$$\log p_\theta(X) \geq \mathbb{E}_{q(Z)}[\log p_\theta(X \mid Z)] - \mathrm{KL}(q(Z)) \parallel p_\theta(Z))$$

(ج) آیا بیشینه کردن این کران پایین در حین آموزش به‌تنهایی کافی است؟ چه مشکلاتی ممکن است ایجاد شود؟

خیر، بیشینه کردن ELBO به‌تنهایی نمی‌تواند تضمین‌کننده عملکرد بهینه مدل باشد. یکی از مشکلات اصلی این است که اگر توزیع تقریبی $q(Z)$ نتواند به خوبی توزیع پسین واقعی $p_\theta(Z|X)$ را تقریب بزند، حتی با مقدار زیاد ELBO نیز مدل می‌تواند عملکرد ضعیفی داشته باشد. همچنین، پدیده‌ای به نام posterior collapse ممکن است رخ دهد، که در آن مدل عملاً استفاده از متغیر پنهان Z را کنار می‌گذارد و صرفاً از شبکه بازسازی $p_\theta(X|Z)$ استفاده می‌کند، که معمولاً زمانی اتفاق می‌افتد که KL-divergence به صفر میل کند. علاوه بر این، ساختار ELBO ذاتاً شامل یک trade-off بین دو مؤلفه است: دقت بازسازی و فاصله KL و در حین آموزش ممکن است یکی از این دو قربانی دیگری شود. در نهایت، انتخاب فرم محدودکننده برای $q(Z)$ مانند توزیع گاوسی ساده، می‌تواند موجب شود که مدل نتواند ساختار پیچیده‌تری از پسین واقعی را به درستی مدل کند.

(د) **تفاضل** بین $\log p_\theta(X)$ و کران پایین استخراج‌شده چیست؟
راهنمایی: این فاصله را به صورت KL divergence بنویسید.

تفاضل بین $\log p_\theta(X)$ و کران پایین (ELBO) برابر است با واگرایی کولبک‌ـلایبلر بین توزیع پیشنهادی $q(Z)$ و توزیع پسین واقعی $p_\theta(Z|X)$. این رابطه به صورت زیر بیان می‌شود:

$$\log p_\theta(X) - \mathrm{ELBO} = D_{\mathrm{KL}}(q(Z) \parallel p_\theta(Z|X))$$

این مقدار همیشه نامنفی است، زیرا KL واگرایی یک معیار فاصله است که هیچ‌گاه منفی نمی‌شود. بنابراین، هر چه $q(Z)$ بهتر بتواند $p_\theta(Z|X)$ را تقریب بزند، مقدار KL کمتر شده و در نتیجه ELBO به $\log p_\theta(X)$ نزدیک‌تر می‌شود. در حالت ایده‌آل، اگر $q(Z) = p_\theta(Z|X)$، آنگاه KL برابر صفر شده و ELBO دقیقاً با لگاریتم درست‌نمایی برابر می‌شود. بنابراین، این تفاضل نشان‌دهنده میزان خطای ناشی از تقریب پسین است. برای بدست آوردن این عبارت (تفاضل بین ELBO و تابع مورد نظر) به صورت زیر عمل می‌کنیم:

$$\begin{aligned}
\log p_\theta(X) &= \mathbb{E}_{Z\sim q(Z)}[\log p_\theta(X)] \\
&= \mathbb{E}_{Z\sim q(Z)}\left[\log \frac{p_\theta(X|Z)p_\theta(Z)}{p_\theta(Z|X)}\right] \\
&= \mathbb{E}_{Z\sim q(Z)}\left[\log \left(\frac{p_\theta(X|Z)p_\theta(Z)}{p_\theta(Z|X)} \times \frac{q(Z)}{q(Z)}\right)\right] \\
&= \mathbb{E}_{Z\sim q(Z)}[\log p_\theta(X|Z)] - \mathbb{E}_{Z\sim q(Z)}\left[\log \left(\frac{q(Z)}{p_\theta(Z)}\right)\right] + \mathbb{E}_{Z\sim q(Z)}\left[\log \left(\frac{q(Z)}{p_\theta(Z|X)}\right)\right] \\
&= \mathbb{E}_{Z\sim q(Z)}[\log p_\theta(X|Z)] - D_{\mathrm{KL}}(q(Z)\|p_\theta(Z)) + D_{\mathrm{KL}}(q(Z)\|p_\theta(Z|X))
\end{aligned}$$

عبارت سوم به دلیل Intractable بودن $p_\theta(Z|X)$ غیر قابل محاسبه و تفاضل مور نظر است.

گام ۴: پارامتری کردن و بهینه‌سازی Posterior Approximation

تا اینجا انتخاب توزیع پیشنهادی را توجیه کردیم، اما با یک چالش جدید مواجه هستیم: posterior واقعی $p_\theta(Z \mid X)$ غیرقابل محاسبه است. بنابراین نمی‌توانیم مستقیماً KL divergence را کمینه کنیم. برای رفع این مشکل باید فرم توزیع تقریبی $q(Z)$ را مشخص کنیم. یک انتخاب رایج، توزیع گاوسی است:

$$q_\lambda(Z) = \mathcal{N}(Z \mid \mu, \sigma^2 I), \quad \lambda = \{\mu, \sigma^2\}$$

(الف) چگونه می‌توان KL divergence را به صورت **غیرمستقیم** کمینه کرد؟
راهنمایی: به گرادیان $\nabla_\lambda \log p_\theta(X)$ فکر کنید.

از آن‌جایی که توزیع پسین واقعی $p_\theta(Z|X)$ غیرقابل محاسبه است، نمی‌توان KL-divergence را مستقیماً کمینه کرد. در عوض، از استراتژی غیرمستقیمی استفاده می‌شود: به جای کمینه کردن مستقیم KL ما کران پایین شواهد (ELBO) را بیشینه می‌کنیم، چرا که طبق رابطه زیر:

$$\log p_\theta(X) = \mathrm{ELBO} + D_{\mathrm{KL}}(q_\lambda(Z) \parallel p_\theta(Z|X))$$

بیشینه کردن ELBO معادل با کمینه کردن KL است (در صورتی که $\log p_\theta(X)$ ثابت در نظر گرفته شود). برای بهینه‌سازی ELBO نسبت به پارامترهای λ توزیع تقریبی $q_\lambda(Z) = \mathcal{N}(Z|\mu, \sigma^2 I)$، نیاز به محاسبه گرادیان داریم. ELBO را به صورت زیر بانویسی می‌کنیم:

$$\mathbb{E}_{Z\sim q_\lambda(Z)}[\log p_\theta(X|Z)] - D_{\mathrm{KL}}(q_\lambda(Z)\|p_\theta(Z))$$

در اینجا دو جزء داریم. بخش اول:

$$\mathbb{E}_{Z\sim q_\lambda(Z)}[\log p_\theta(X|Z)]$$

این عبارت به صورت مستقیم شامل توزیع قابل کنترل $q_\lambda(Z)$ است و قابل تقریب مونت‌کارلو است. اما چون Z وابسته به λ است، محاسبه گرادیان نسبت به λ به سادگی انجام نمی‌شود (بعداً reparameterization trick به کمک می‌آید) و بخش دوم:

$$D_{\mathrm{KL}}(q_\lambda(Z)\|p_\theta(Z))$$

اگر فرض کنیم که توزیع prior برابر با $\mathcal{N}(0, I)$ است و

$$q_\lambda(Z) = \mathcal{N}(\mu, \sigma^2 I)$$

آنگاه KL بسته است و فرمول آن به‌صورت زیر به دست می‌آید:

$$D_{\mathrm{KL}}(q_\lambda(Z)\|p_\theta(Z)) = \frac{1}{2}\sum_j \left(\log \sigma_j^2 - 1 + \frac{1}{\sigma_j^2} + \mu_j^2\right)$$

که گرادیان آن نسبت به μ و σ نیز به‌سادگی قابل محاسبه است. در نتیجه می‌توان با بیشینه کردن کران پایین شواهد به هدف خواسته شده در صورت سوال دست یافت.

(ب) گرادیان ELBO ∇_θ و تخمین Monte Carlo آن را محاسبه کنید.

کران پایین شواهد (ELBO) به صورت زیر تعریف می‌شود:

$$\mathrm{ELBO}(\theta, \lambda) = \mathbb{E}_{Z\sim q_\lambda(Z)}[\log p_\theta(X|Z)] - D_{\mathrm{KL}}(q_\lambda(Z)\|p(Z))$$

برای محاسبه گرادیان ELBO نسبت به پارامترهای مدل θ (که معمولاً پارامترهای شبکه مولد هستند)، کافی است گرادیان بخش اول یعنی $\log p_\theta(X|Z)$ را محاسبه کنیم، زیرا KL-term به θ وابسته نیست (فرض بر این است که $p(Z)$ و $q_\lambda(Z)$ مستقل از θ هستند). پس داریم:

$$\nabla_\theta \mathrm{ELBO} = \mathbb{E}_{Z\sim q_\lambda(Z)}[\nabla_\theta \log p_\theta(X|Z)]$$

از آن‌جا که محاسبه این امیدریاضی دقیقاً ممکن نیست، از تخمین مونت‌کارلو استفاده می‌کنیم. با نمونه‌گیری $Z^{(l)} \sim q_\lambda(Z)$ برای $l = 1, \ldots, L$، می‌توان تخمین زیر را نوشت:

$$\nabla_\theta \mathrm{ELBO} \approx \frac{1}{L}\sum_{l=1}^{L} \nabla_\theta \log p_\theta(X|Z^{(l)})$$

(ج) آیا می‌توانید ∇_λ ELBO را نیز محاسبه کنید؟ چه چالش‌هایی در هنگام تخمین این گرادیان با نمونه‌گیری Monte Carlo وجود دارد و چگونه می‌توان آن‌ها را حل کرد؟
راهنمایی: بررسی کنید چگونه می‌توان نمونه‌گیری $Z \sim q_\lambda(Z)$ را طوری نوشت که تصادفی بودن از پارامترهای λ جدا شود.

بله، می‌توان گرادیان کران پایین شواهد (ELBO) را نسبت به پارامترهای λ توزیع تقریبی $q_\lambda(Z)$ نیز محاسبه کرد. اما همانطور که در بخش الف گفته شد، چالش اصلی در اینجا این است که هنگام نمونه‌گیری از $Z \sim q_\lambda(Z)$، خود نمونه‌ها به λ وابسته هستند و در نتیجه نمی‌توان به‌راحتی از درون امیدریاضی نسبت به λ مشتق گرفت. این موضوع باعث افزایش واریانس تخمین گرادیان و ناپایداری در آموزش می‌شود. برای حل این مشکل، از تکنیکی به نام Reparameterization Trick استفاده می‌شود. در این تکنیک، به جای اینکه مستقیماً از $Z \sim q_\lambda(Z) = \mathcal{N}(Z|\mu, \sigma^2 I)$ نمونه‌برداری کنیم، نمونه‌گیری را به صورت تابعی از یک نویز تصادفی مستقل از λ بازنویسی می‌کنیم:

$$Z = \mu + \sigma \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

در این صورت، تصادفی بودن نمونه‌ها از پارامترهای $\lambda = \{\mu, \sigma^2\}$ جدا می‌شود و می‌توان از مشتق‌گیری زنجیره‌ای (backpropagation) برای محاسبه دقیق و پایدار گرادیان ELBO نسبت به λ استفاده کرد. این بازنویسی باعث کاهش واریانس تخمین مونت‌کارلو و تسهیل آموزش مدل‌هایی مانند VAE می‌شود.

گام ۵: تعمیم به کل مجموعه داده

تمام محاسبات قبلی بر اساس یک نمونه داده انجام شدند. اما اگر بخواهیم $\log p_\theta(X)$ را روی کل مجموعه داده بیشینه کنیم، چه تغییراتی ایجاد می‌شود؟
(الف) این کار چه تأثیری روی **تابع هدف** و **گرادیان‌ها** نسبت به θ و λ دارد؟

در صورت استفاده از کل مجموعه داده $\mathcal{D} = \{X^{(i)}\}_{i=1}^N$، هدف دیگر بهینه‌سازی $\log p_\theta(X)$ برای یک نمونه منفرد نیست، بلکه میانگین لگاریتم احتمال روی کل مجموعه داده باید بیشینه شود. بنابراین، تابع هدف به شکل زیر تغییر می‌کند:

$$\mathcal{L}(\theta, \lambda) = \frac{1}{N}\sum_{i=1}^{N} \log p_\theta(X^{(i)}) \geq \frac{1}{N}\sum_{i=1}^{N} \text{ELBO}(X^{(i)}; \theta, \lambda)$$

یعنی ما میانگین کران پایین شواهد (ELBO) را روی کل داده‌ها بیشینه می‌کنیم. این تغییر باعث می‌شود که: ۱. تابع هدف به جای یک ELBO، به مجموع یا میانگین هاELBO تبدیل شود. ۲. گرادیان نسبت به θ و λ نیز به صورت میانگین گرادیان‌های مربوط به هر نمونه محاسبه می‌شود:

$$\nabla_\theta \mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \nabla_\theta \text{ELBO}(X^{(i)}) \quad \text{و} \quad \nabla_\lambda \mathcal{L} = \frac{1}{N}\sum_{i=1}^{N} \nabla_\lambda \text{ELBO}(X^{(i)})$$

در عمل، به دلیل هزینه محاسباتی بالا، به جای استفاده از کل داده، از **مینی‌بچ‌ها (mini-batches)** استفاده می‌شود و میانگین ELBO روی یک زیرمجموعه کوچک از داده‌ها محاسبه و گرادیان‌ها بر اساس آن تخمین زده می‌شوند. این تکنیک اساس روش Stochastic Gradient Descent است.

(ب) پارامترهای رمزگشا (θ) به‌سادگی قابل تعمیم به کل مجموعه هستند. اما λ، که پارامترهای توزیع تقریبی $q_\lambda(Z)$ است، برای هر داده به طور جدا تعریف شده. چگونه می‌توان λ را طوری مدل کرد که قابلیت تعمیم به همه داده‌ها را داشته باشد؟
راهنمایی: اینجا نقش شبکه encoder مطرح می‌شود (پارامترهای آن را با ϕ نام‌گذاری کنید).

از آنجا که پارامترهای $\lambda = \{\mu, \sigma^2\}$ مربوط به توزیع تقریبی $q_\lambda(Z)$ هستند و برای هر نمونه داده متفاوت‌اند، تعریف جداگانه‌ی آن‌ها برای هر $X^{(i)}$ غیربهینه و غیرقابل تعمیم است. برای حل این مشکل، به‌جای اینکه λ را به صورت صریح برای هر داده تعریف کنیم، از یک **شبکه عصبی** استفاده می‌کنیم که با گرفتن ورودی X، مقادیر $\mu(X)$ و $\sigma^2(X)$ را به عنوان خروجی تولید کند. این شبکه نقش **رمزگذار (encoder)** را ایفا می‌کند و پارامترهای آن را با ϕ نمایش می‌دهیم. در این حالت، توزیع تقریبی به شکل زیر بازنویسی می‌شود:

$$q_\phi(Z|X) = \mathcal{N}(Z|\mu_\phi(X), \sigma_\phi^2(X) \cdot I)$$

مزیت این روش آن است که اکنون به‌جای داشتن پارامترهای جداگانه برای هر داده، تنها پارامترهای شبکه ϕ را یاد می‌گیریم که برای تمام داده‌ها تعمیم‌پذیر هستند. این کار باعث می‌شود توزیع پشتیبان (posterior approximation) به صورت آماری شرطی بر داده باشد و در حین آموزش بتوانیم آن را با داده‌های جدید نیز ارزیابی کنیم. در نتیجه، مدل encoder به ما اجازه می‌دهد که λ را به‌صورت تابعی از X مدل کرده و به‌جای به‌روزرسانی تعداد زیادی پارامتر مجزا، تنها پارامترهای شبکه ϕ را آموزش دهیم.

گام ۶: ELBO نهایی و آموزش مدل

اکنون که کل هدف VAE استخراج شده است، فرآیند آموزش را توصیف کنید.
(الف) نشان دهید که ELBO را می‌توان به صورت زیر نوشت و مفهوم هر ترم از عبارات را توضیح دهید:

$$\mathbb{E}_{z\sim q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right] - \text{KL}\left(q_\phi(z \mid x) \,\|\, p_\theta(z)\right)$$

از نامساوی ینسن و با توجه به روابط بخش‌های قبل داریم:

$$\log p_\theta(X) \geq \mathbb{E}_{q_\phi(Z|X)}\left[\log \frac{p_\theta(X,Z)}{q_\phi(Z|X)}\right]$$

همچنین داریم: $p_\theta(X,Z) = p_\theta(X|Z)p_\theta(Z)$ پس سمت راست عبارت بالا برابر است با:

$$\mathbb{E}_{q_\phi(Z|X)}\left[\log\left(\frac{p_\theta(X|Z)p_\theta(Z)}{q_\phi(Z|X)}\right)\right]$$

در نتیجه:

$$\mathbb{E}_{q_\phi(Z|X)}[\log p_\theta(X|Z)] + \mathbb{E}_{q_\phi(Z|X)}\left[\log \frac{p_\theta(Z)}{q_\phi(Z|X)}\right]$$

جمله دوم را می‌توان به صورت زیر نوشت:

$$\mathbb{E}_{q_\phi(Z|X)}[\log p_\theta(X|Z)] - \mathrm{KL}(q_\phi(Z|X) \parallel p_\theta(Z))$$

فرم نهایی ELBO به این صورت در می‌آید:

$$\mathbb{E}_{Z\sim q_\phi(Z|X)}[\log p_\theta(X|Z)] - \mathrm{KL}(q_\phi(Z|X) \parallel p_\theta(Z))$$

این فرمول شامل دو بخش اصلی است:
۱. $\mathbb{E}_{Z\sim q_\phi(Z|X)}[\log p_\theta(X|Z)]$: این ترم، امیدریاضی لگاریتم احتمال بازسازی داده X با استفاده از نمونه‌ی نهان Z است که از توزیع تقریبی $q_\phi(Z|X)$ استخراج شده. این بخش به عنوان خطای بازسازی (Reconstruction Error) شناخته می‌شود و تلاش می‌کند مدل بازساز (Decoder) را آموزش دهد تا بتواند داده‌ی ورودی را از روی متغیرهای نهان به‌درستی بازسازی کند.
۲. $\mathrm{KL}(q_\phi(Z|X) \parallel p_\theta(Z))$: این ترم، واگرایی KL بین توزیع نهان تقریبی $q_\phi(Z|X)$ و توزیع پیشین (Prior) $p_\theta(Z)$ است. این بخش به عنوان ترم منظم‌کننده (Regularization Term) عمل می‌کند و تضمین می‌کند که فضای نهان Z به توزیع پیشین (معمولاً توزیع نرمال استاندارد) نزدیک بماند. در نتیجه، فرآیند آموزش VAE شامل یادگیری پارامترهای ϕ (رمزگذار) و θ (بازساز) به گونه‌ای است که این تابع ELBO را برای داده‌های آموزشی بیشینه کند. این فرآیند همزمان هم مدل‌سازی بازسازی داده و هم ساختاردهی فضای نهان را شامل می‌شود.

(ب) حال که مدل VAE کامل شده است، الگوریتم آموزش آن را بنویسید. در این الگوریتم نشان دهید:

- چگونه از encoder و decoder استفاده می‌شود؟
- نمونه‌گیری چگونه انجام می‌شود؟
- تابع هدف چگونه بهینه می‌شود؟

مدل VAE از دو بخش اصلی تشکیل شده است: **رمزگذار (Encoder)** که تابع $q_\phi(z|x)$ را مدل می‌کند، و **رمزگشا (Decoder)** که تابع $p_\theta(x|z)$ را مدل می‌کند. در طول آموزش، پارامترهای این دو بخش با استفاده از کران پایین شواهد (ELBO) به‌روزرسانی می‌شوند.
الگوریتم آموزش VAE:

۱. **ورودی:** داده‌های آموزشی $\{X^{(i)}\}_{i=1}^N$
۲. **مقداردهی اولیه:** پارامترهای ϕ (encoder) و θ (decoder) را به صورت تصادفی مقداردهی اولیه کن.
۳. **برای هر اپوک** تکرار کن:

- داده‌ها را به مینی‌بچ‌هایی به اندازه B تقسیم کن.
- برای هر batch از داده‌ها:

(آ) برای هر نمونه x در batch از encoder برای محاسبه‌ی $\mu_\phi(x)$ و $\sigma^2_\phi(x)$ استفاده کن.

(ب) نمونه‌گیری نهان: از Reparameterization Trick استفاده کن:

$$z = \mu_\phi(x) + \sigma_\phi(x) \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I)$$

(ج) خروجی z را به decoder بده تا بازسازی $\hat{x} \sim p_\theta(x|z)$ را محاسبه کند.

(د) محاسبه‌ی تابع هدف ELBO:

$$\text{ELBO} = \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z)] - \text{KL}(q_\phi(z|x)||p(z))$$

(ه) گرادیان‌های ELBO نسبت به ϕ و θ را محاسبه کن و با استفاده از بهینه‌سازی (Adam) پارامترها را به‌روزرسانی کن.

۴. پایان اپوک

در این الگوریتم، encoder وظیفه دارد که توزیع تقریبی $q_\phi(z|x)$ را تولید کند و decoder وظیفه دارد که نمونه‌ی نهان z را به فضای داده بازگرداند.نمونه‌گیری از z با استفاده از Reparameterization Trick انجام می‌شود تا گرادیان‌گیری مستقیم از طریق backpropagation ممکن باشد. با بهینه‌سازی ELBO کاهش و فضای نهان ساختار مناسبی می‌یابد.

مدل Hierarchical VAE (HVAE) با دو لایه متغیرهای نهان z_1 (لایه پایین) و z_2 (لایه بالا) را در نظر می‌گیریم، به طوری که $z_1 \geq z_2$. لگاریتم درست‌نمایی حاشیه‌ای را به صورت زیر تعریف می‌کنیم:

$$\log p(x) = \log \int_{z_1} \int_{z_2} p(x, z_1, z_2)\, dz_2\, dz_1$$

حال با استفاده از نامساوی ینسن توزیع تقریبی را معرفی می‌کنیم:

$$\log p(x) \geq \mathbb{E}_{q(z_1,z_2|x)} \left[\log \frac{p(x, z_1, z_2)}{q(z_1, z_2|x)}\right] \equiv L(x)$$

این عبارت، کران پایین شواهد (ELBO) را تعریف می‌کند. با جایگزینی مدل مولد و تقریب پسین ELBO را به فرم قابل تفسیر تبدیل می‌کنیم:

$$L(x) = \mathbb{E}_{q(z_1,z_2|x)} \left[\log \frac{p(x|z_1)p(z_1|z_2)p(z_2)}{q(z_1|x)q(z_2|z_1)}\right]$$

این عبارت را می‌توان به سه بخش اصلی تقسیم کرد:

$$
\begin{aligned}
&= \mathbb{E}_{q(z_1,z_2|x)}\left[\log p(x|z_1) + \log p(z_1|z_2) + \log p(z_2) - \log q(z_1|x) - \log q(z_2|z_1)\right] \\
&= \mathbb{E}_{q(z_1,z_2|x)}\left[\log p(x|z_1)\right] + \mathbb{E}_{q(z_1,z_2|x)}\left[\log \frac{p(z_1|z_2)}{q(z_1|x)}\right] + \mathbb{E}_{q(z_1,z_2|x)}\left[\log \frac{p(z_2)}{q(z_2|z_1)}\right] \\
&= \iint q(z_1|x)q(z_2|z_1)\log p(x|z_1)dz_1dz_2 \\
&\quad + \iint q(z_1|x)q(z_2|z_1)\log \frac{p(z_1|z_2)}{q(z_1|x)}dz_1dz_2 \\
&\quad + \iint q(z_1|x)q(z_2|z_1)\log \frac{p(z_2)}{q(z_2|z_1)}dz_1dz_2 \\
&= \int q(z_1|x)\log p(x|z_1)dz_1 \underbrace{\int q(z_2|z_1)dz_2}_{=1} \\
&\quad + \int q(z_1|x)\log \frac{p(z_1|z_2)}{q(z_1|x)}dz_1 \underbrace{\int q(z_2|z_1)dz_2}_{=1} \\
&\quad + \int q(z_1|x)\left[\int q(z_2|z_1)\log \frac{p(z_2)}{q(z_2|z_1)}dz_2\right]dz_1 \\
&= \mathbb{E}_{q(z_1|x)}\left[\log p(x|z_1)\right] - D_{\mathrm{KL}}\big(q(z_1|x)\|p(z_1|z_2)\big) - \mathbb{E}_{q(z_1|x)}\left[D_{\mathrm{KL}}\big(q(z_2|z_1)\|p(z_2)\big)\right]
\end{aligned}
$$

(ب)

معنای هر یک از اجزای موجود در ELBO به‌دست آمده در قسمت (الف) را توضیح دهید و تأثیر آن‌ها را بررسی کنید.

- $\mathbb{E}_{q(z_1|x)}\left[\log p(x|z_1)\right]$: این بخش دقت بازسازی داده‌ها از طریق لایه پایین z_1 را اندازه‌گیری می‌کند و مسئول کیفیت خروجی مدل است. هرچه مقدار این ترم بیشتر باشد، مدل در بازسازی داده‌ها بهتر عمل می‌کند.
- $D_{KL}(q(z_1|x) \parallel p(z_1|z_2))$: این بخش تفاوت توزیع تقریبی z_1 از توزیع مولد شرطی $p(z_1|z_2)$ را اندازه‌گیری می‌کند و تنظیم‌کننده توزیع متغیرهای نهان سطح بالا است. به عبارت دیگر، بررسی می‌کند که توزیع z_1 پیش‌بینی‌شده توسط انکدر چقدر با توزیع مولد آن (با توجه به z_2) تفاوت دارد. این ترم به مدل کمک می‌کند تا سلسله‌مراتب متغیرهای نهان را به درستی یاد بگیرد. اگر مقدار این واگرایی زیاد باشد، نشان‌دهنده عدم تطابق بین انکدر و دیکدر در لایه z_1 است.
- $\mathbb{E}_{q(z_1|x)}\left[D_{KL}(q(z_2|z_1) \parallel p(z_2))\right]$: در واقع این بخش تضمین می‌کند که توزیع z_2 بیش‌ازحد از توزیع پیشین $p(z_2)$ (معمولاً گاوسی استاندارد) فاصله نگیرد. اگر مقدار این واگرایی بسیار زیاد باشد، نشان‌دهنده عدم کارایی در یادگیری لایه بالایی مدل است. این جزء نقش تنظیم‌کننده (Regularizer) را ایفا می‌کند و از یادگیری توزیع‌های نامناسب برای z_2 جلوگیری می‌نماید.

(ج)

در Hierarchical VAEها، هدف این است که هر لایه‌ی نهان، اطلاعات مفیدی را کدگذاری کند که در بازسازی داده نقش داشته باشد. با این حال، در عمل اغلب مشاهده می‌شود که متغیرهای نهان لایه‌های بالایی مانند z_2 معمولاً توسط مدل نادیده گرفته می‌شوند. به این معنا که خروجی encoder برای z_2، یعنی $q(z_2 \mid z_1)$، تقریباً مستقل از ورودی x شده و با توزیع پیشین $p(z_2)$ هم‌راستا می‌شود.

- توضیح دهید که چرا این پدیده که به آن «فروریزش توزیع پسین» (Posterior Collapse) یا «متغیرهای نهان غیرفعال» (Inactice Latent Variables) گفته می‌شود، در زمان بهینه‌سازی ELBO رخ می‌دهد.
- این رفتار چه چیزی را در مورد جریان اطلاعات در ساختار سلسله‌مراتبی مدل نشان می‌دهد؟
- حداقل دو تکنیک (اعم از فرایند آموزش مدل یا تغییر در معماری مدل) پیشنهاد دهید که می‌توانند به جلوگیری از این مسئله کمک کنند و توضیح دهید چرا این تکنیک‌ها مؤثر هستند.

پدیده فروریزش توزیع پسین در مدل‌های سلسله‌مراتبی: در مدل‌های Hierarchical VAE، مشاهده می‌شود که متغیرهای نهان در لایه‌های بالاتر (z_2 مانند) اغلب در فرآیند یادگیری نادیده گرفته می‌شوند. این پدیده زمانی رخ می‌دهد که توزیع پسین تقریبی $q(z_2|z_1)$ به توزیع پیشین $p(z_2)$ نزدیک شده و عملاً مستقل از ورودی x می‌گردد. دلیل اصلی این رفتار را می‌توان در ماهیت تابع هدف ELBO جستجو کرد. هنگام بهینه‌سازی، مدل به طور طبیعی به سمت راه‌حلی تمایل پیدا می‌کند که در آن لایه‌های پایین‌تر (z_1 مانند) به تنهایی قادر به بازسازی مناسب داده‌ها باشند. در این حالت، از آنجا که ترم KL بین $q(z_2|z_1)$ و $p(z_2)$ در ELBO حضور دارد، مدل ترجیح می‌دهد این واگرایی را با نزدیک کردن توزیع پسین به پیشین به حداقل برساند. این امر منجر به وضعیتی می‌شود که در آن متغیرهای لایه‌های بالاتر اطلاعات مفیدی درباره ورودی رمزگذاری نمی‌کنند و نقش موثری در فرآیند تولید ایفا نمی‌نمایند. این پدیده به ویژه زمانی تشدید می‌شود که ظرفیت مدل در لایه‌های پایین‌تر برای بازسازی داده‌ها کافی باشد، یا هنگامی که توزیع‌های پیشین بیش از حد محدودکننده انتخاب شده باشند. برای مقابله با این مشکل، محققان راهکارهای مختلفی از جمله استفاده از ضرایب تعادلی برای ترم‌های KL، طراحی معماری‌های جایگزین برای انکودر، یا معرفی توزیع‌های پیشین پیچیده‌تر را پیشنهاد داده‌اند. درک این پدیده و راهکارهای مقابله با آن برای طراحی مدل‌های سلسله‌مراتبی کارآمد ضروری است.

تحلیل جریان اطلاعات در معماری سلسله‌مراتبی: پدیده فروریزش توزیع پسین در مدل‌های HVAE نشان‌دهنده اختلال در جریان اطلاعات در ساختار سلسله‌مراتبی مدل است. این رفتار حاکی از آن است که اطلاعات ورودی x به صورت مؤثر از لایه‌های پایین‌تر (z_1) به لایه‌های بالاتر (z_2) منتقل نمی‌شود و در واقع سلسله‌مراتب یادگیری شده با سلسله‌مراتب طراحی شده در مدل تناسبی ندارد. وقتی z_2 مستقل از ورودی x شود، نشان می‌دهد که مسیر اطلاعاتی بین انکدر و دیکدر در لایه‌های بالایی قطع شده است. در چنین حالتی، مدل به جای استفاده از تمام ظرفیت خود و ایجاد نمایش‌های توزیع‌یافته در سطوح مختلف، تنها بر روی لایه‌های پایینی تکیه می‌کند. این امر منجر به یادگیری ناقص روابط سلسله‌مراتبی بین متغیرهای نهان می‌شود. از دیدگاه تئوری اطلاعات، این پدیده نشان‌دهنده کاهش اطلاعات متقابل بین ورودی و متغیرهای نهان لایه‌های بالاتر است. در حالت ایده‌آل، هر لایه می‌بایست اطلاعاتی در سطوح متفاوتی از داده‌ها را کدگذاری کند، اما در صورت رخداد فروریزش، این تقسیم‌بندی سلسله‌مراتبی اطلاعات به درستی اتفاق نمی‌افتد. این رفتار همچنین نشان می‌دهد که تعادل قدرت بین اجزای مختلف مدل به هم خورده است. لایه‌های پایین‌تر به اندازه‌ای قوی شده‌اند که می‌توانند بدون کمک لایه‌های بالاتر، وظیفه بازسازی را انجام دهند. این موضوع می‌تواند ناشی از ظرفیت نامتناسب لایه‌ها یا تنظیم نادرست پارامترهای مدل باشد. در نهایت، این پدیده بر کیفیت نمایش‌های یادگرفته شده تأثیر منفی می‌گذارد. در یک مدل سلسله‌مراتبی سالم، انتظار داریم که لایه‌های مختلف سطوح متفاوتی از ویژگی‌ها را استخراج کنند، اما در صورت رخداد فروریزش، این تقسیم کار طبیعی بین لایه‌ها اتفاق نمی‌افتد و مدل از مزایای ساختار سلسله‌مراتبی خود بهره کامل نمی‌برد.

راهکارهای مقابله با فروریزش توزیع پسین:

۱. آموزش تدریجی با گرم کردن (KL Annealing): روش KL Annealing با معرفی تدریجی ترم‌های KL در طول فرآیند آموزش عمل می‌کند. در این روش، ابتدا به مدل اجازه داده می‌شود تا بر روی بازسازی داده‌ها تمرکز کند (با ضریب صفر برای ترم‌های KL)، سپس به تدریج اهمیت این ترم‌ها افزایش می‌یابد. این رویکرد به مدل فرصت می‌دهد تا قبل از اینکه محدودیت‌های KL اعمال شوند، نمایش‌های معناداری در لایه‌های مختلف یاد بگیرد. به ویژه برای لایه‌های بالاتر، این روش مفید است زیرا به z_2 اجازه می‌دهد قبل از هم‌راستا شدن با توزیع پیشین، اطلاعات مفیدی از داده‌ها را یاد بگیرد. پارامتر β که وزن ترم KL را کنترل می‌کند، معمولاً از صفر شروع شده و به تدریج به یک افزایش می‌یابد.

۲. طراحی پیشین‌های وابسته به داده (Data-Dependent Priors): به جای استفاده از توزیع پیشین ثابت (مثلاً گاوسی استاندارد) برای لایه‌های بالاتر، می‌توان از پیشین‌هایی استفاده کرد که به صورت پویا بر اساس داده ورودی تعیین می‌شوند. در این روش، پارامترهای توزیع پیشین $p(z_2)$ به جای اینکه ثابت باشند، از طریق یک شبکه عصبی دیگر که به ورودی x وابسته است محاسبه می‌شوند. این کار باعث می‌شود توزیع پیشین با ویژگی‌های داده سازگار شده و فاصله بین توزیع پسین و پیشین کمتر شود. در نتیجه، مدل انگیزه بیشتری برای استفاده از لایه‌های بالاتر پیدا می‌کند، زیرا دیگر مجبور نیست توزیع پسین را به یک پیشین ثابت و ممکن‌نا مناسب نزدیک کند.

هر دو این روش‌ها با تغییر در فرآیند آموزش یا معماری مدل، به حفظ جریان اطلاعات در طول سلسله‌مراتب و فعال نگه داشتن تمام لایه‌های نهان کمک می‌کنند. KL Annealing با کنترل هوشمندانه تعادل بین بازسازی و تنظیم‌کننده‌های KL عمل می‌کند، در حالی که پیشین‌های وابسته به داده با حذف تنش بین توزیع پسین و پیشین، استفاده از لایه‌های بالاتر را تشویق می‌کنند. ترکیب این دو روش می‌تواند نتایج بهتری در جلوگیری از فروریزش توزیع پسین ایجاد کند.

پرسش ۳. GAN (۲۰ نمره)

در این سوال، قصد داریم بین مسئله‌ی "تشخیص داده واقعی و داده تولید شده" و مسئله‌ی "فاصله توزیع تصاویر واقعی و تصاویر تولید شده" ارتباطی برقرار کنیم.
توزیع تصاویر واقعی را با P_d و توزیع تصاویر تولید شده را با P_g نشان می‌دهیم. همچنین توزیع P که داده‌ای تصادفی را به ما می‌دهد، به‌صورت زیر تعریف می‌شود:

$$P(x) = 0.5P_g(x) + 0.5P_d(x)$$

بنابراین، هر داده با احتمال مساوی از یکی از دو توزیع P_d یا P_g می‌آید.
حال، مجموعه‌ای از نمونه‌ها (x, y) تشکیل می‌دهیم، به این ترتیب که:

- x را از توزیع P نمونه‌گیری می‌کنیم.
- اگر x از توزیع واقعی P_d باشد، برچسب آن را $y = 1$ قرار می‌دهیم، و در غیر این صورت (اگر از P_g آمده باشد) $y = -1$.

اکنون، $\mathcal{D}$ را مجموعه تمامی تمایزدهنده‌ها در نظر می‌گیریم. هدف ما یافتن بهترین تمایزدهنده در این مجموعه است. فرض می‌کنیم این مجموعه هیچ محدودیتی ندارد و تمایزدهنده‌ها دارای ظرفیت نامحدود هستند. منظور از ظرفیت نامحدود این است به ازای هر تابع $y(x)$ دلخواه، تابعی در $\mathcal{D}$ با چنین رفتاری وجود دارد. بنابراین اگر نقطه بهینه تابع برای هر x را بیابید میتوانید از وجود تمایزدهنده‌ای با این اتخاذها مطمئن باشید. بهترین تمایزدهنده D در مجموعه $\mathcal{D}$ تابع هزینه زیر را کمینه می‌کند:

$$R_l(D) = \mathbb{E}_{(x,y)\sim P}[l(yD(x))]$$

در اینجا، هزینه به‌صورت امیدریاضی روی داده‌های (x, y) تعریف شده است. اکنون مقدار بهینه این تابع را تعریف می‌کنیم:

$$R_l(\mathcal{D}) = \inf_{D\in\mathcal{D}} R_l(D)$$

با توجه به صورت مسئله، امیدریاضی را به فرم انتگرالی زیر بازنویسی می‌کنیم:

$$\begin{aligned} R_l(D) &= \mathbb{E}_{(x,y)\sim P}\left[l(yD(x))\right] \\ &= \mathbb{E}_{y\sim Y, x\sim X|y}\left[l(yD(x))\right] \\ &= \frac{1}{2}\int p_d(x) l(D(x))\, dx + \frac{1}{2}\int p_g(x) l(-D(x))\, dx \\ &= \int \left(\frac{1}{2} l(D(x)) p_d(x) + \frac{1}{2} l(-D(x)) p_g(x)\right) dx \end{aligned}$$

با فرض ظرفیت نامحدود برای D، می‌توانیم بهینه‌سازی را به صورت نقطه‌ای انجام دهیم:

$$\inf_D R_l(D) = \int \inf_{D(x)} \left\{\frac{1}{2} l(D(x)) p_d(x) + \frac{1}{2} l(-D(x)) p_g(x)\right\} dx$$

با تعریف $t = \frac{p_d(x)}{p_g(x)}$، می‌توان نوشت:

$$\inf_D R_l(D) = -\frac{1}{2}\int \left[-\inf_{D(x)} \left\{t l(D(x)) + l(-D(x))\right\}\right] p_g(x)\, dx$$

تابع f را به صورت زیر تعریف می‌کنیم:

$$f(t) = -\inf_{D(x)} \{tl(D(x)) + l(-D(x))\}$$

در نتیجه مقدار بهینه به صورت زیر درمی‌آید:

$$R_l(D^*) = -\frac{1}{2}\int f\left(\frac{p_d(x)}{p_g(x)}\right) p_g(x)\, dx = -\frac{1}{2}\mathbb{I}_f(\mathbb{P}_d, \mathbb{P}_g)$$

که در آن $\mathbb{I}_f(\mathbb{P}_d, \mathbb{P}_g)$ واگرایی f بین توزیع‌های واقعی و تولید شده است.

بخش دوم

در بخش قبل، نشان دادیم که این مسئله معادل کمینه کردن یک واگرایی بین دو توزیع است. اکنون می‌خواهیم بررسی کنیم که با انتخاب توابع هزینه مختلف $l(x)$، چه نوع واگرایی‌هایی حاصل می‌شوند.
برای هر تابع هزینه $l(x)$:

۱. ابتدا محدب بودن و یکنوایی تابع f متناظر را بررسی کنید.

۲. سپس، تمایزدهنده بهینه، تابع f، و واگرایی متناظر را استخراج کنید.

حالت‌های مورد بررسی:

- **الف)** $l(x) = \mathbb{I}(x \leq 0)$
- **ب)** $l(x) = (1-x)^2$
- **ج)** $l(x) = \log(1 + e^{-x})$

راهنمایی: واگرایی های متناظر توابع هزینه به صورت نامرتب در ادامه آمده‌اند. می‌توانید جهت بررسی درستی حل خود از این بخش استفاده کنید.

- $R(\mathcal{D}) = \frac{1}{2}(1 - \mathbb{I}_{\mathsf{TV}}(\mathbb{P}_d, \mathbb{P}g))$ که منظور از TV تابع Total Variation است.
- $R(\mathcal{D}) = \log 2 - \mathbb{I}_{\mathsf{JS}}(\mathbb{P}_d, \mathbb{P}_g)$ که منظور از JS تابع Jensen Shannon است.
- $R(\mathcal{D}) = 1 - \mathbb{I}_g(\mathbb{P}_d, \mathbb{P}_g)$ که منظور از g تابع $\frac{-4t}{t+1}$ است.

۱ تابع هزینه $l(x) = \mathbb{I}(x \leq 0)$

:

$$\begin{aligned}
&\inf_{D(x)} \{(tl(D(x)) + l(-D(x)))p_g(x)\} \\
&\quad = \inf_{D(x)} \{(t\mathbb{I}(D(x) \leq 0) + \mathbb{I}(D(x) \geq 0))\}\, p_g(x) \\
&\quad = p_g(x) \inf_{D(x)} \begin{cases} t, & D(x) \leq 0 \\ 1, & D(x) \geq 0 \end{cases}
\end{aligned}$$

تمایزدهنده بهینه و تابع f به صورت زیر تعیین می‌شوند:

$$D^*(x) = \begin{cases} -1, & t < 1 \\ 1, & t \geq 1 \end{cases} = \begin{cases} -1, & p_d(x) < p_g(x) \\ 1, & p_d(x) \geq p_g(x) \end{cases}$$

$$f(t) = -\min(1, t) = \begin{cases} -t, & t \leq 1 \\ -1, & t > 1 \end{cases}$$

بررسی ویژگی‌های تابع f:
۱. یکنوایی نزولی:

$$f'(t) = \begin{cases} -1, & t \leq 1 \\ 0, & t > 1 \end{cases}$$

از آنجا که مشتق همیشه نامثبت است، تابع f نزولی است.
۲. محدب بودن: برای بررسی محدب بودن، شرط زیر را برای همه $t_1, t_2 \in \mathbb{R}$ و $\lambda \in [0, 1]$ بررسی می‌کنیم:

$$f(\lambda t_1 + (1-\lambda)t_2) \leq \lambda f(t_1) + (1-\lambda) f(t_2)$$

حالات مختلف:

- **حالت ۱**: $t_1, t_2 \leq 1$
$$f(\lambda t_1 + (1-\lambda)t_2) = -(\lambda t_1 + (1-\lambda)t_2) = \lambda f(t_1) + (1-\lambda) f(t_2)$$
- **حالت ۲**: $t_1, t_2 > 1$
$$f(\lambda t_1 + (1-\lambda)t_2) = -1 = \lambda(-1) + (1-\lambda)(-1) = \lambda f(t_1) + (1-\lambda) f(t_2)$$
- **حالت ۳**: $t_1 \leq 1 < t_2$
برای $m = \lambda t_1 + (1-\lambda)t_2$:
  - اگر $m \leq 1$: $f(m) = -m \leq -\lambda t_1 - (1-\lambda) = \lambda f(t_1) + (1-\lambda) f(t_2)$
  - اگر $m > 1$: $f(m) = -1 \leq -\lambda t_1 - (1-\lambda) = \lambda f(t_1) + (1-\lambda) f(t_2)$

واگرایی متناظر و ارتباط با Total Variation:
واگرایی f به صورت زیر محاسبه می‌شود:

$$\mathbb{I}_f(P_d, P_g) = -\int \min\left(\frac{p_d(x)}{p_g(x)}, 1\right) p_g(x) dx = -\int \min(p_d(x), p_g(x)) dx$$

واگرایی Total Variation به صورت زیر تعریف می‌شود:

$$\mathbb{I}_{TV}(P_d, P_g) = \frac{1}{2}\int |p_d(x) - p_g(x)| dx = 1 - \int \min(p_d(x), p_g(x)) dx$$

بنابراین ارتباط بین آنها:

$$\mathbb{I}_f(P_d, P_g) = \mathbb{I}_{TV}(P_d, P_g) - 1$$

و در نهایت مقدار بهینه:

$$R_l(D^*(x)) = -\frac{1}{2}\mathbb{I}_f(P_d, P_g) = -\frac{1}{2}(\mathbb{I}_{TV}(P_d, P_g) - 1) = \frac{1}{2}(1 - \mathbb{I}_{TV}(P_d, P_g))$$

۲ تابع هزینه $l(x) = (1-x)^2$

:

$$\begin{aligned}
&\inf_{D(x)} \left\{(tl(D(x)) + l(-D(x)))p_g(x)\right\} \\
&\quad = \inf_{D(x)} \left\{(t(1-D(x))^2 + (1+D(x))^2)\right\} p_g(x) \\
&\quad = p_g(x) \inf_{D(x)} \left\{t(1-2D(x)+D(x)^2) + 1 + 2D(x) + D(x)^2\right\} \\
&\quad = p_g(x) \inf_{D(x)} \left\{(t+1)D(x)^2 + 2(1-t)D(x) + (t+1)\right\}
\end{aligned}$$

برای یافتن مینیمم، از مشتق‌گیری استفاده می‌کنیم:

$$\frac{d}{dD(x)}\left[(t+1)D(x)^2 + 2(1-t)D(x) + (t+1)\right] = 2(t+1)D(x) + 2(1-t) = 0$$

حل برای $D^*(x)$:

$$D^*(x) = \frac{t-1}{t+1} \quad \text{که در آن} \quad t = \frac{p_d(x)}{p_g(x)}$$

با جایگذاری $D^*(x)$ در عبارت اصلی:

$$\begin{aligned}
f(t) &= -\inf_{D(x)} \left\{(t+1)D(x)^2 + 2(1-t)D(x) + (t+1)\right\} \\
&= -\left[(t+1)\left(\frac{t-1}{t+1}\right)^2 + 2(1-t)\left(\frac{t-1}{t+1}\right) + (t+1)\right] \\
&= -\frac{4t}{t+1}
\end{aligned}$$

بررسی ویژگی‌های تابع f

۱. یکنوایی نزولی:

$$f'(t) = -\frac{4}{(t+1)^2} < 0 \quad \text{برای همه} \quad t \geq 0$$

۲. محدب بودن:

$$f''(t) = \frac{8}{(t+1)^3} > 0 \quad \text{برای همه} \quad t > -1$$

واگرایی متناظر

واگرایی f به صورت زیر محاسبه می‌شود:

$$\mathbb{I}_f(P_d, P_g) = -4 \int \frac{p_d(x)}{p_d(x) + p_g(x)} p_g(x) dx$$

و مقدار بهینه تابع هزینه:

$$R_l(D^*(x)) = -\frac{1}{2}\mathbb{I}_f(P_d, P_g) = 2 \int \frac{p_d(x)p_g(x)}{p_d(x) + p_g(x)} dx$$

۳ تابع هزینه $l(x) = \log(1 + e^{-x})$

:

$$\inf_{D(x)} \{(tl(D(x)) + l(-D(x)))p_g(x)\}$$
$$= \inf_{D(x)} \left\{t \log(1 + e^{-D(x)}) + \log(1 + e^{D(x)})\right\} p_g(x)$$

برای یافتن مینیمم، از مشتق‌گیری استفاده می‌کنیم:

$$\frac{d}{dD(x)} \left[t \log(1 + e^{-D(x)}) + \log(1 + e^{D(x)})\right]$$
$$= t \cdot \frac{-e^{-D(x)}}{1 + e^{-D(x)}} + \frac{e^{D(x)}}{1 + e^{D(x)}} = 0$$

با تعریف تابع سیگموئید $\sigma(D(x)) = \frac{1}{1+e^{-D(x)}}$:

$$-t(1 - \sigma(D(x))) + \sigma(D(x)) = 0 \implies \sigma(D^*(x)) = \frac{t}{t+1}$$

در نتیجه تمایزدهنده بهینه برابر است با:

$$D^*(x) = \log\left(\frac{\sigma(D^*(x))}{1 - \sigma(D^*(x))}\right) = \log(t) = \log\left(\frac{p_d(x)}{p_g(x)}\right)$$

با جایگذاری $D^*(x)$ در عبارت اصلی:

$$f(t) = t \log\left(1 + \frac{1}{t}\right) + \log(1 + t)$$
$$= t \log\left(\frac{t+1}{t}\right) + \log(t + 1)$$
$$= (t + 1) \log(t + 1) - t \log t$$

یا به شکل معادل:

$$f(t) = t \log t - (t + 1) \log(t + 1)$$

بررسی ویژگی‌های تابع f
۱. یکنوایی نزولی:

$$f'(t) = \log t + 1 - \log(t + 1) - 1 = -\log\left(1 + \frac{1}{t}\right) < 0 \quad \forall t > 0$$

۲. محدب بودن:

$$f''(t) = \frac{1}{t} - \frac{1}{t+1} = \frac{1}{t(t+1)} > 0 \quad \forall t > 0$$

واگرایی متناظر و ارتباط با JSD
واگرایی f به صورت زیر محاسبه می‌شود:

$$\mathbb{I}_f(P_d, P_g) = \int \left[p_d(x) \log\left(\frac{p_d(x)}{p_d(x) + p_g(x)}\right) + p_g(x) \log\left(\frac{p_g(x)}{p_d(x) + p_g(x)}\right)\right] dx$$

واگرایی Jensen-Shannon (JSD):

$$\mathbb{I}_{JS}(P_d, P_g) = \frac{1}{2}\mathbb{I}_f(P_d, P_g) + \log(2)$$

مقدار بهینه تابع هزینه:

$$R_l(D^*(x)) = -\frac{1}{2}\mathbb{I}_f(P_d, P_g) = \log(2) - \mathbb{I}_{JS}(P_d, P_g)$$

تابع هزینه اصلی به صورت زیر تعریف شده است:

$$l_1(\theta) = \mathbb{E}_{q(x)}\left[\frac{1}{2}\|s_\theta(x) - \nabla_x \log q(x)\|_2^2\right]$$

عبارت داخل امید ریاضی را بسط می‌دهیم:

$$\frac{1}{2}\|s_\theta(x) - \nabla_x \log q(x)\|_2^2 = \underbrace{\frac{1}{2}\|s_\theta(x)\|_2^2}_{A} \underbrace{- s_\theta(x)^T.\nabla_x \log q(x)}_{B} + \underbrace{\frac{1}{2}\|\nabla_x \log q(x)\|_2^2}_{C}$$

عبارت B را می‌توان به صورت زیر بازنویسی کرد:

$$B = s_\theta(x)^T.\nabla_x \log q(x) = s_\theta(x)^T \frac{\nabla_x q(x)}{q(x)}$$

در نتیجه:

$$\mathbb{E}_{q(x)}[B] = \int s_\theta(x)^T.\nabla_x q(x) dx$$

در اینجا باید به کمک انتگرال جز به جز عبارت بالا را محاسبه کنیم. در حالت یک بعدی، فرمول انتگرال جز به جز به صورت زیر است:

$$\int u\, dv = uv - \int v\, du$$

برای حالت چندبعدی که با میدان‌های برداری کار می‌کنیم، فرمول به این صورت می‌شود:

$$\int_{\mathbb{R}^d} \mathbf{u}(x)^T \cdot \nabla q(x)\, dx = \mathbf{u}(x)q(x)|_{-\infty}^{\infty} - \int_{\mathbb{R}^d} q(x)(\nabla \cdot \mathbf{u}(x))\, dx$$

که در آن:

- $\mathbf{u}(x) = s_\theta(x)$
- $v = q(x)$
- $dv = \nabla q(x) dx$

با این جایگزینی داریم:

$$\int s_\theta(x)^T \cdot \nabla q(x) dx = s_\theta(x) q(x)|_{-\infty}^{\infty} - \int q(x)(\nabla \cdot s_\theta(x)) dx$$

با فرض $q(x)s_\theta(x) \to 0$ وقتی $x \to \infty$، جملات مرزی صفر می‌شوند:

$$\mathbb{E}_{q(x)}[B] = \int s_\theta(x)^T . \nabla_x q(x) dx = 0 - \int q(x) \nabla_x \cdot s_\theta(x) dx$$

$$\mathbb{E}_{q(x)}[B] = -\mathbb{E}_{q(x)}[\nabla_x \cdot s_\theta(x)] = -\mathbb{E}_{q(x)}[\mathrm{Tr}(\nabla_x s_\theta(x))]$$

با جایگزینی نتایج در تابع هزینه اصلی داریم:

$$l_1(\theta) = \mathbb{E}_{q(x)} \left[\frac{1}{2} \|s_\theta(x)\|_2^2 + \mathrm{Tr}(\nabla_x s_\theta(x)) \right] + C_1$$

که در آن C_1 مستقل از θ است. این تابع هزینه به دلایل زیر برای آموزش مناسب نیست:

- محاسبه $\mathrm{Tr}(\nabla_x s_\theta(x))$ نیاز به محاسبه ماتریس ژاکوبین کامل دارد.
- برای ابعاد بالا، این محاسبه بسیار پرهزینه است.
- تخمین دقیق آن در عمل دشوار است.

بخش (ب): ارتباط با تخمین نویز اکنون هدف ما این است که نشان دهیم تابع هزینه بالا معادل با تابع هزینه‌ای است که بر اساس تخمین نویز تعریف می‌شود:

$$l_3(\theta) = \mathbb{E}_{x \sim q(x),\ \epsilon \sim \mathcal{N}(0, \mathcal{I})} \left[\frac{1}{2} \left\| s_\theta(\underbrace{x + \sigma\epsilon}_{\tilde{x}}) + \frac{\epsilon}{\sigma} \right\|^2 \right]$$

جفت داده سالم و نویزی را $(x, \tilde{x})$ می‌نامیم. طبق رابطه زیر، برای کرنل گاوسی داریم:

$$\frac{\partial \log q(\tilde{x}|x)}{\partial \tilde{x}} = \frac{1}{\sigma^2}(x - \tilde{x})$$

در نتیجه:

$$l_3(\theta) = \mathbb{E}_{q(x,\tilde{x})} \left[\frac{1}{2} \left\| s_\theta(\tilde{x}) - \frac{\partial \log q(\tilde{x}|x)}{\partial \tilde{x}} \right\|^2 \right]$$

ثابت کنید که:

$$l_3 = l + C$$

که C مستقل از θ است.
اگر نیاز به راهنمایی دارید، می‌توانید به پیوست مقاله زیر مراجعه کنید:
A Connection Between Score Matching and Denoising Autoencoders
اما سعی کنید مراحل اثبات را خودتان کامل نوشته و توضیح دهید.

در تابع هزینه بالا به جای x عبارت $\tilde{x}$ را قرار می‌دهیم و داریم:

$$\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\left\|s_\theta(\tilde{x})-\nabla_{\tilde{x}}\log q(\tilde{x})\right\|_2^2\right]=\mathbb{E}_{q(\tilde{x})}\left[\underbrace{\frac{1}{2}\left\|s_\theta(\tilde{x})\right\|_2^2}_{A}-\underbrace{s_\theta(\tilde{x})^\top\nabla_{\tilde{x}}\log q(\tilde{x})}_{B}+\underbrace{\frac{1}{2}\left\|\nabla_{\tilde{x}}\log q(\tilde{x})\right\|_2^2}_{C}\right]$$

عبارت دوم را به صورت زیر بازنویسی می‌کنیم:

$$\begin{aligned}
\mathbb{E}_{q(\tilde{x})}[B] &= \mathbb{E}_{\tilde{q}(\tilde{x})}\left[\left\langle s_\theta(\tilde{x}),\frac{\partial\log q(\tilde{x})}{\partial\tilde{x}}\right\rangle\right]\\
&=\int_{\tilde{x}} q(\tilde{x})\left\langle S_\theta(\tilde{x}),\frac{\partial\log q(\tilde{x})}{\partial\tilde{x}}\right\rangle d\tilde{x}\\
&=\int_{\tilde{x}} q(\tilde{x})\left\langle s_\theta(\tilde{x}),\frac{1}{q(\tilde{x})}\frac{\partial q(\tilde{x})}{\partial\tilde{x}}\right\rangle d\tilde{x}\\
&=\int_{\tilde{x}}\left\langle s_\theta(\tilde{x}),\frac{\partial q(\tilde{x})}{\partial\tilde{x}}\right\rangle d\tilde{x}\\
&=\int_{\tilde{x}}\left\langle s_\theta(\tilde{x}),\frac{\partial}{\partial\tilde{x}}\int_x q(x)q(\tilde{x}|x)dx\right\rangle d\tilde{x}\\
&=\int_{\tilde{x}}\left\langle s_\theta(\tilde{x}),\int_x q(x)\frac{\partial q(\tilde{x}|x)}{\partial\tilde{x}}dx\right\rangle d\tilde{x}\\
&=\int_{\tilde{x}}\int_x q(x)q(\tilde{x}|x)\left\langle s_\theta(\tilde{x}),\frac{\partial\log q(\tilde{x}|x)}{\partial\tilde{x}}\right\rangle dx\,d\tilde{x}\\
&=\int_x\int_{\tilde{x}} q(x,\tilde{x})\left\langle s_\theta(\tilde{x}),\frac{\partial\log q(\tilde{x}|x)}{\partial\tilde{x}}\right\rangle d\tilde{x}\,dx\\
&=\mathbb{E}_{q(x,\tilde{x})}\left[\left\langle s_\theta(\tilde{x}),\frac{\partial\log q(\tilde{x}|x)}{\partial\tilde{x}}\right\rangle\right]
\end{aligned}$$

حال داریم:

$$\ell_1(\theta)=\mathbb{E}_{q(\tilde{x})}\left[\frac{1}{2}\left\|s_\theta(\tilde{x})\right\|^2\right]-\mathbb{E}_{q(x,\tilde{x})}\left[\left\langle s_\theta(\tilde{x}),\frac{\partial\log q(\tilde{x}|x)}{\partial\tilde{x}}\right\rangle\right]+C_2$$

سپس تابع هزینه $\ell_3(\theta)$ را به صورت زیر بازنویسی می‌کنیم:

$$\begin{aligned}
\ell_3(\theta) &= \mathbb{E}_{q(x,\tilde{x})}\left[\frac{1}{2}\left\|s_\theta(\tilde{x})-\frac{\partial\log q(\tilde{x}|x)}{\partial\tilde{x}}\right\|^2\right]\\
&=\mathbb{E}_{q(x,\tilde{x})}\left[\frac{1}{2}\left\|s_\theta(\tilde{x})\right\|^2-\left\langle s_\theta(\tilde{x}),\frac{\partial\log q(\tilde{x}|x)}{\partial\tilde{x}}\right\rangle+\frac{1}{2}\left\|\frac{\partial\log q(\tilde{x}|x)}{\partial\tilde{x}}\right\|^2\right]\\
&=\mathbb{E}_{q(x,\tilde{x})}\left[\frac{1}{2}\left\|s_\theta(\tilde{x})\right\|^2\right]-\mathbb{E}_{q(x,\tilde{x})}\left[\left\langle s_\theta(\tilde{x}),\frac{\partial\log q(\tilde{x}|x)}{\partial\tilde{x}}\right\rangle\right]\\
&\quad+\underbrace{\mathbb{E}_{q(x,\tilde{x})}\left[\frac{1}{2}\left\|\frac{\partial\log q(\tilde{x}|x)}{\partial\tilde{x}}\right\|^2\right]}_{C_3}
\end{aligned}$$

از معادله $\ell_1(\theta)$ و $\ell_3(\theta)$ نتیجه می‌گیریم:

$$\ell_1(\theta) = \ell_3(\theta) + C$$

پرسش ۵. آیا فرض مارکوف در Diffusion الزامی است؟ (۲۰ نمره)

در ساختار فروارد دیفیوژن، فرض کردیم که:

$$q(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1} q(\mathbf{x}_t \mid \mathbf{x}_{t-1})$$

این فرض مارکوف باعث می‌شود که برای تولید، نیاز به انجام T مرحله به صورت ترتیبی داشته باشیم. برای درستی فرض گاوسی بودن می‌دانیم، T باید بزرگ باشد که باعث هزینه زمانی زیاد می‌شود. آیا می‌توانیم بدون فرض مارکوف بودن هم به همان توزیع حاشیه $q(x_t \mid x_0)$ برسیم؟
دقت کنید که در فرم تابع هدف (Objective) نهایی، یعنی:

$$L(\epsilon_\theta) := \sum_{t=1}^{T} \gamma_t \, \mathbb{E}_{\mathbf{x}_0 \sim q(\mathbf{x}_0),\, \boldsymbol{\epsilon}_t \sim \mathcal{N}(0,\mathbf{I})} \left[\left\| \epsilon_\theta^{(t)} \left(\sqrt{\bar{\alpha}_t} \mathbf{x}_0 + \sqrt{1-\bar{\alpha}_t} \boldsymbol{\epsilon}_t \right) - \boldsymbol{\epsilon}_t \right\|_2^2 \right]$$

تنها این حاشیه تاثیر دارد و توزیع مشترک x_i ها تاثیری ندارد.
در مدل استاندارد، این مارجین به صورت زیر است:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0, (1-\bar{\alpha}_t)\mathbf{I})$$

بنابراین، می‌توان ساختاری غیرمارکوفی تعریف کرد که همین مارجین را تولید کند.
برای یک بردار $\sigma \in \mathbb{R}^T_{\geq 0}$ توزیع اینفرنس را به‌صورت زیر تعریف می‌کنیم:

$$q_\sigma(x_{1:T} \mid x_0) := q_\sigma(x_T \mid x_0) \prod_{t=2}^{T} q_\sigma(x_{t-1} \mid x_t, x_0)$$

که در آن:

$$q_\sigma(x_T \mid x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_T} x_0, (1-\bar{\alpha}_T)\mathbf{I})$$

و برای $t > 1$:

$$q_\sigma(x_{t-1} \mid x_t, x_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}} x_0 + \sqrt{1-\bar{\alpha}_{t-1}-\sigma_t^2} \cdot \frac{x_t - \sqrt{\bar{\alpha}_t} x_0}{\sqrt{1-\bar{\alpha}_t}}, \sigma_t^2 \mathbf{I} \right)$$

الف) اثبات برابری مارجین‌ها با استفاده از خواص توزیع گاوسی و به کمک استقرا ثابت کنید:

$$q_\sigma(x_t \mid x_0) = \mathcal{N}(\sqrt{\bar{\alpha}_t}x_0, (1-\bar{\alpha}_t)\mathbf{I})$$

توجه کنید که فرایند فروارد ما دیگر مارکوف نیست، زیرا x_t به x_0 نیز وابسته است:

$$q_\sigma(x_t \mid x_{t-1}, x_0) = \frac{q_\sigma(x_{t-1} \mid x_t, x_0)\, q_\sigma(x_t \mid x_0)}{q_\sigma(x_{t-1} \mid x_0)}$$

مشاهده جالب: اگر $\sigma_t \to 0$، واریانس گاوسی به صفر میل می‌کند و x_{t-1} به‌صورت قطعی از x_0 و x_t به دست می‌آید. بنابراین می‌توان فرآیند جریشن را به یک فرآیند قطعی تبدیل کرد که تنها منبع تصادفی آن نویز اولیه x_T است. این دیدگاه امکان نوعی تناظر بین فضای پنهان و خروجی را فراهم می‌کند، و دقت کنید که توزیع حاشیه‌ای که حاصل می‌شود همان توزیع مدل دیفیوژن تصادفی خواهد بود.

هدف ما اثبات فرم زیر برای توزیع $q(x_t \mid x_0)$ است:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0,\ (1-\bar{\alpha}_t)\mathbf{I})$$

که در آن:

$$\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$$

پایه استقرا ($t=1$): در گام اول فرآیند انتشار، داریم:

$$q(x_1 \mid x_0) = \mathcal{N}(x_1;\ \sqrt{\alpha_1}x_0,\ (1-\alpha_1)\mathbf{I})$$

و چون:

$$\bar{\alpha}_1 = \alpha_1$$

پس رابطه برقرار است:

$$q(x_1 \mid x_0) = \mathcal{N}(x_1;\ \sqrt{\bar{\alpha}_1}x_0,\ (1-\bar{\alpha}_1)\mathbf{I})$$

فرض استقرا: فرض می‌کنیم که برای زمان $t-1$ رابطه برقرار باشد:

$$q(x_{t-1} \mid x_0) = \mathcal{N}(x_{t-1}; \sqrt{\bar{\alpha}_{t-1}}x_0,\ (1-\bar{\alpha}_{t-1})\mathbf{I})$$

می‌خواهیم نشان دهیم:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}x_0,\ (1-\bar{\alpha}_t)\mathbf{I})$$

می‌دانیم که:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}x_{t-1},\ (1-\alpha_t)\mathbf{I})$$

و از فرض استقرا داریم:

$$q(x_{t-1} \mid x_0) = \mathcal{N}(x_{t-1}; \sqrt{\bar{\alpha}_{t-1}}x_0,\ (1-\bar{\alpha}_{t-1})\mathbf{I})$$

حال می‌خواهیم انتگرال زیر را محاسبه کنیم:

$$q(x_t \mid x_0) = \int q(x_t \mid x_{t-1})\, q(x_{t-1} \mid x_0)\, dx_{t-1}$$

این ترکیب دو توزیع نرمال با نگاشت خطی است. فرض می‌کنیم:

$$x_{t-1} \sim \mathcal{N}(\mu,\ \Sigma), \quad \text{که در اینجا:} \quad \mu = \sqrt{\bar{\alpha}_{t-1}}x_0,\ \Sigma = (1-\bar{\alpha}_{t-1})\mathbf{I}$$

و:

$$x_t = \sqrt{\alpha_t}x_{t-1} + \epsilon, \quad \epsilon \sim \mathcal{N}(0,\ (1-\alpha_t)\mathbf{I})$$

پس طبق خواص توزیع گاوسی، داریم:

- **میانگین:**

$$\mathbb{E}[x_t \mid x_0] = \sqrt{\alpha_t} \cdot \mathbb{E}[x_{t-1} \mid x_0] = \sqrt{\alpha_t} \cdot \sqrt{\bar{\alpha}_{t-1}} x_0 = \sqrt{\bar{\alpha}_t} x_0$$

- **کوواریانس:**

$$\mathrm{Var}(x_t \mid x_0) = \alpha_t \cdot (1 - \bar{\alpha}_{t-1}) + (1 - \alpha_t) = 1 - \bar{\alpha}_t$$

در نتیجه:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0,\ (1 - \bar{\alpha}_t)\mathbf{I})$$

با استفاده از استقرا و خواص ترکیب خطی توزیع نرمال، نشان دادیم که برای هر t داریم:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t} x_0,\ (1 - \bar{\alpha}_t)\mathbf{I})$$

فرآیند جنریشن ابتدا با استفاده از عبارت زیر که مشابه مدل استاندارد است، نمونه‌ای نویزی ساخته و بدون دسترسی به x_0 نویز آن را تخمین می‌زنیم:

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

با مدل تخمین نویز می‌توانیم نمونه‌ی بدون نویز را به صورت زیر بازسازی کنیم:

$$f_\theta^{(t)}(x_t) := \frac{x_t - \sqrt{1 - \bar{\alpha}_t} \cdot \epsilon_\theta^{(t)}(x_t)}{\sqrt{\bar{\alpha}_t}}$$

بنابراین ابتدا داریم که

$$p_\theta(x_T) = \mathcal{N}(0, \mathbf{I})$$

و پس از آن

$$p_\theta^{(t)}(x_{t-1} \mid x_t) = \begin{cases} \mathcal{N}(f_\theta^{(1)}(x_1), \sigma_1^2 \mathbf{I}) & t = 1 \\ q_\sigma(x_{t-1} \mid x_t, f_\theta^{(t)}(x_t)) & \text{غیر این صورت} \end{cases}$$

حال با این مدل جنریشن، آموزش به چه صورتی می‌شود؟ ابتدا دقت می‌کنیم که هدف آموزش مدل ما چه بود؟ مشابه قبل، ما به دنبال پارامترهایی هستیم که عبارت زیر اتخاذ شود.

$$\arg\max_\theta\ \mathbb{E}_{x_0 \sim q}[\log p_\theta(x_0)]$$

بنابراین می‌توان همان استدلال‌های مدل استاندارد را برای یافت تابع هزینه در این مدل نیز کرد.

در حالت کلی، مدل p_θ توزیع گاوسی‌ای است که میانگین آن با استفاده از تخمین x_0 محاسبه می‌شود و واریانس یا به صورت ثابت یا قابل یادگیری انتخاب می‌شود. هدف آموزش این مدل، بیشینه‌سازی لاگ‌احتمال بازسازی نمونه اصلی است:

$$\arg\max_\theta\ \mathbb{E}_{x_0, \epsilon, t}\left[\log p_\theta(x_0 \mid x_t)\right]$$

که به‌صورت معادل در مدل‌های انتشار کلاسیک تبدیل می‌شود به:

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{x_0, \epsilon, t}\left[\left\| \epsilon - \epsilon_\theta^{(t)}(x_t) \right\|^2\right]$$

این تابع هزینه به مدل یاد می‌دهد که نویز ϵ را از روی نمونه‌ی نویزی x_t پیش‌بینی کند.

ب) تطابق تابع هدف (Objective) با دیفیوژن کلاسیک با تکرار مراحل ذکر شده در اسلایدهای درس و تحلیل ELBO نشان دهید که تابع هدف این مدل مولد با مدل بررسی‌شده در اسلایدها یکسان است (به جز یک ثابت مستقل از θ). در برابری فرض کنید که در تابع هدف مدل استاندارد تابع هزینه t های مختلف، ضریب برابری دارند.

پرسش: به نظر شما چگونه می‌توان از این نگاه برای کاهش زمان طولانی جنریشن استفاده کرد؟

با شروع از درست‌نمایی لگاریتمی و استفاده از نابرابری ینسن:

$$\begin{aligned}\mathbb{E}_{x_0} \log p_\theta(x_0) &\geq \mathbb{E}_{q(x_{0:T})}\left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right] \\ &= \mathbb{E}_q\left[\log p_\theta(x_T) + \sum_{t>1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)} + \log p_\theta(x_0|x_1)\right]\end{aligned}$$

جمله دوم مربوط به واگرایی کولبک‌ـلیبلر بین توزیع‌های گاوسی است:

$$\mathbb{E}_q\left[\sum_{t>1} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_{t-1}|x_t, x_0)}\right] = -\sum_{t>1} \frac{1}{2\sigma_t^2}\mathbb{E}\left[\|\tilde{\mu}(x_t, x_0) - \mu_\theta(x_t, t)\|^2\right] + C$$

با جایگزینی فرمول‌های میانگین:

$$\tilde{\mu}(x_t, x_0) = \frac{1}{\sqrt{\alpha_t}} x_t + \left(\frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\alpha_t}} + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2}\right)\epsilon$$

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} x_t + \left(\frac{\sqrt{1-\bar{\alpha}_t}}{\sqrt{\alpha_t}} + \sqrt{1 - \bar{\alpha}_{t-1} - \sigma_t^2}\right)\epsilon_\theta(x_t, t)$$

پس از ساده‌سازی به این شکل درمی‌آید:

$$L = \mathbb{E}_{t, x_0, \epsilon}\left[\gamma_t' \|\epsilon - \epsilon_\theta(x_t, t)\|^2\right]$$

که در آن γ_t' یک ضریب ثابت است.

کاهش زمان نمونه‌برداری: در مدل‌های مارکوفی استاندارد، فرآیند معکوس نیاز به T مرحله (معمولاً ۱۰۰۰) دارد:

$$p_\theta(x_{t-1}|x_t) = \mathcal{N}(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

با تعریف فرآیند معکوس به صورت:

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}\left(\sqrt{\bar{\alpha}_{t-1}} x_0 + \gamma_t(x_t - \sqrt{\bar{\alpha}_t} x_0), \sigma_t^2 I\right)$$

با انتخاب $\sigma_t = 0$ فرآیند معکوس قطعی می‌شود:

$$x_{t-1} = \sqrt{\bar{\alpha}_{t-1}} x_0 + \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}}(x_t - \sqrt{\bar{\alpha}_t} x_0)$$