به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

مدل‌های مولد عمیق

تمرین سوم

نام و نام خانوادگی	سید محمد جزایری
شماره دانشجویی	810101399
تاریخ ارسال گزارش	1404/10/02

فهرست

فهرست تصاویر 3
سؤال اول..... 4
بخش اول 4
زیر بخش اول)..... 4
زیر بخش دوم)..... 4
زیر بخش سوم)..... 5
بخش دوم 5
زیر بخش اول)..... 5
زیر بخش دوم)..... 6
زیر بخش سوم) 11
زیر بخش چهارم)..... 12
سؤال دوم 15
بخش اول 15
زیر بخش اول)..... 15
زیر بخش دوم)..... 15
زیر بخش سوم)..... 16
زیر بخش چهارم)..... 16
بخش دوم 17
زیر بخش اول)..... 17
زیر بخش دوم)..... 20
مراجع 23

فهرست تصاویر

تصویر 1. چند نمونه تصادفی از بخش آموزش.....5
تصویر 2. چند نمونه تصادفی از بخش تست.....6
تصویر3. چند نمونه تصادفی تولید شده در epoch اول 9
تصویر4. چند نمونه تصادفی تولید شده در epoch پنجم.....9
تصویر5. چند نمونه تصادفی تولید شده در epoch دهم 9
تصویر 6. تصاویر اصلی که از روی آن ها مدل تصاویر را خواهد ساخت.....10
تصویر7. خروجی مدل EBM روی تصاویر تصویر 6.....10
تصویر 8. نمونه های تولید شده از نویز خالص.....11
تصویر9. نمونه های واقعی 11
تصویر 10. نمونه های واقعی.....12
تصویر 11. نمونه های واقعی نویزی شده با سطح 0.2 12
تصویر 12. خروجی دینویز شده سطح 0.2 مدل.....13
تصویر 13. نمونه های واقعی نویزی شده با سطح 0.4.....13
تصویر 14. خروجی دینویز شده سطح 0.4 مدل.....13
تصویر 15. نمونه های واقعی نویزی شده با سطح 0.6.....14
تصویر 16. خروجی دینویز شده سطح 0.6 مدل.....14
تصویر 17. نمودار زیان مدل 17
تصویر 18. تصویر Grid شامل 16 عکس تولید شده.....18
تصویر 19. فرآیند تبدیل 3 تصویر از نویز به عکس نهایی.....19
تصویر 20. نمودار زیان مدل شرطی 20
تصویر 21. تصاویر خروجی مدل شرطی 21

سؤال اول

بخش اول

زیر بخش اول)

برای اینکه مدل بتواند ویژگی‌های خاصی (مانند مرد جوان) را تولید کند، باید از مدل‌های مبتنی بر انرژی شرطی (Conditional EBMs) استفاده کنیم. در این روش، تابع انرژی علاوه بر ورودی x (تصویر)، به یک متغیر کلاس یا ویژگی y (برچسب مرد جوان) نیز وابسته می‌شود: $E(x, y)$. برای آموزش راحت‌تر، می‌توان برچسب‌های ویژگی را به عنوان ورودی به مدل داد تا مدل یاد بگیرد انرژی جفت‌های سازگار (x, y) را کمینه (یا در فرمول این سوال، تابع f را بیشینه) کند. این کار باعث می‌شود هنگام نمونه‌برداری، بتوانیم با فیکس کردن y، نمونه‌هایی تولید کنیم که انرژی متناظر با آن کلاس خاص را داشته باشند.

زیر بخش دوم)

روش Rejection Sampling تکنیکی برای نمونه‌برداری از یک توزیع هدف $P(x)$ است که نمونه‌برداری مستقیم از آن دشوار است (مثل EBMها که ثابت نرمال‌سازی Z را نداریم)، اما می‌توانیم مقدار متناسب با آن $\widehat{P}(x)$ را محاسبه کنیم.

مراحل انجام کار:

1. یک توزیع پیشنهادی $Q(x)$ (مثل نرمال) انتخاب می‌کنیم که نمونه‌برداری از آن ساده باشد.
2. یک ثابت M پیدا می‌کنیم به طوری که برای همه xها داشته باشیم: $Mq(x) \geq \widehat{p(x)}$.
3. یک نمونه x از $Q(x)$ و یک عدد تصادفی u از توزیع یکنواخت [0, 1] تولید می‌کنیم.
4. اگر $u \leq \frac{\widehat{P}(x)}{MQ(x)}$، نمونه را می‌پذیریم؛ در غیر این صورت آن را رد می‌کنیم.

محاسبه Optimal Acceptance Rate:

نرخ پذیرش بهینه برابر است با $M/1$ (در حالتی که توزیع‌ها نرمالایز باشند). برای پیدا کردن M باید ماکزیمم نسبت دانسیته‌ها را پیدا کنیم:

$$M = max_x \frac{p(x)}{q(x)}$$

برای دو توزیع گوسی با میانگین برابر در ابعاد d:

$$\frac{p(x)}{q(x)} \propto \left(\frac{\sigma_q}{\sigma_p}\right)^d exp\left(-\frac{||x-\mu||^2}{2}\left(\frac{1}{\sigma_p^2} - \frac{1}{\sigma_q^2}\right)\right)$$

چون $\sigma_q > \sigma_p$، توان نمایی منفی است و ماکزیمم این کسر در $x = \mu$ رخ می‌دهد (جایی که توان نمایی صفر می‌شود). بنابراین مقدار M برابر است با نسبت ثابت‌های نرمال‌سازی:

$$M = \left(\frac{\sigma_q}{\sigma_p}\right)^d$$

نرخ پذیرش (Acceptance Rate) برابر است با:

$$\frac{1}{M} = \left(\frac{\sigma_p}{\sigma_q}\right)^d = \left(\frac{1}{1.01}\right)^{784} \simeq 0.0004$$

زیر بخش سوم)

الف)با توجه به اینکه فرمول احتمال در صورت سوال به صورت $e^{f(x)}$ تعریف شده است (یعنی f متناسب با احتمال است):

- **ترم اول ($\nabla f(x_{train})$):** سعی می‌کند مقدار تابع f (انرژی منفی یا امتیاز) را برای داده‌های واقعی آموزشی افزایش دهد. یعنی مدل یاد می‌گیرد به داده‌های واقعی امتیاز بالاتری بدهد.
- **ترم دوم ($-E[\nabla f(x_{sample})]$):** سعی می‌کند مقدار تابع f را برای نمونه‌های تولید شده توسط مدل (که هنوز واقعی نیستند) کاهش دهد. این کار باعث می‌شود احتمال تولید داده‌های پرت کم شود.

ب)

- چالش: محاسبه ترم دوم نیازمند محاسبه امید ریاضی روی تمام نمونه‌های ممکن از توزیع مدل p_θ است. نمونه‌برداری دقیق از مدل نیازمند اجرای زنجیره مارکوف (MCMC) طولانی تا زمان همگرایی است که بسیار زمان‌بر و محاسباتی سنگین می‌باشد.
- راه حل (Contrastive Divergence (CD): روش CD این چالش را با یک تقریب ساده می‌کند: به جای اجرای MCMC تا بی‌نهایت (همگرایی کامل)، زنجیره نمونه‌برداری را با داده‌های آموزشی مقداردهی اولیه می‌کند (نه نویز تصادفی) و فقط تعداد کمی گام (مثلاً k=1) اجرا می‌کند. این روش سریع‌تر است و تقریب خوبی از گرادیان ارائه می‌دهد.

بخش دوم

زیر بخش اول)

Random Train Images

تصویر 1. چند نمونه تصادفی از بخش آموزش

Random Test Images

تصویر 2. چند نمونه تصادفی از بخش تست

زیر بخش دوم)

در پایین زیان ها و انرژی هایی که در صورت مسئله قید شده بودند را می بینیم:

Epoch [1/10], Step [0], Data Loss: 0.0008, Reg Loss: 0.0000, E_real: -0.0078, E_fake: -0.0086

Epoch [1/10], Step [200], Data Loss: -0.0831, Reg Loss: 0.0026, E_real: -1.1297, E_fake: -1.0466

Epoch [1/10], Step [400], Data Loss: -0.2097, Reg Loss: 0.0012, E_real: -0.7756, E_fake: -0.5659

Epoch [1/10], Step [600], Data Loss: 0.1764, Reg Loss: 0.0008, E_real: -0.4410, E_fake: -0.6174

Epoch [1/10], Step [800], Data Loss: -0.5856, Reg Loss: 0.0009, E_real: -0.8142, E_fake: -0.2285

Epoch [2/10], Step [0], Data Loss: -0.0977, Reg Loss: 0.0006, E_real: -0.4649, E_fake: -0.3673

Epoch [2/10], Step [200], Data Loss: -0.0147, Reg Loss: 0.0033, E_real: -1.2621, E_fake: -1.2474

Epoch [2/10], Step [400], Data Loss: -0.2774, Reg Loss: 0.0016, E_real: -0.9816, E_fake: -0.7042

Epoch [2/10], Step [600], Data Loss: -0.3208, Reg Loss: 0.0004, E_real: -0.5356, E_fake: -0.2149

Epoch [2/10], Step [800], Data Loss: -0.1877, Reg Loss: 0.0018, E_real: -1.0069, E_fake: -0.8193

Epoch [3/10], Step [0], Data Loss: 0.0399, Reg Loss: 0.0012, E_real: -0.6914, E_fake: -0.7314

Epoch [3/10], Step [200], Data Loss: -0.0058, Reg Loss: 0.0010, E_real: -0.6864, E_fake: -0.6806

Epoch [3/10], Step [400], Data Loss: -0.0843, Reg Loss: 0.0002, E_real: -0.2409, E_fake: -0.1566

Epoch [3/10], Step [600], Data Loss: 0.0640, Reg Loss: 0.0002, E_real: -0.1862, E_fake: -0.2502

Epoch [3/10], Step [800], Data Loss: 0.0283, Reg Loss: 0.0001, E_real: -0.1256, E_fake: -0.1539

Epoch [4/10], Step [0], Data Loss: 0.0129, Reg Loss: 0.0001, E_real: -0.1220, E_fake: -0.1349

Epoch [4/10], Step [200], Data Loss: 0.0780, Reg Loss: 0.0001, E_real: -0.0543, E_fake: -0.1323

Epoch [4/10], Step [400], Data Loss: -0.0932, Reg Loss: 0.0001, E_real: -0.0101, E_fake: 0.0831

Epoch [4/10], Step [600], Data Loss: -0.0205, Reg Loss: 0.0000, E_real: -0.0629, E_fake: -0.0424

Epoch [4/10], Step [800], Data Loss: 0.0891, Reg Loss: 0.0001, E_real: -0.1209, E_fake: -0.2100

Epoch [5/10], Step [0], Data Loss: -0.0026, Reg Loss: 0.0001, E_real: 0.0551, E_fake: 0.0577

Epoch [5/10], Step [200], Data Loss: -0.1215, Reg Loss: 0.0001, E_real: -0.0804, E_fake: 0.0411

Epoch [5/10], Step [400], Data Loss: 0.0308, Reg Loss: 0.0001, E_real: 0.0970, E_fake: 0.0663

Epoch [5/10], Step [600], Data Loss: -0.0235, Reg Loss: 0.0001, E_real: 0.0659, E_fake: 0.0894

Epoch [5/10], Step [800], Data Loss: 0.0309, Reg Loss: 0.0001, E_real: -0.1169, E_fake: -0.1478

Epoch [6/10], Step [0], Data Loss: -0.0410, Reg Loss: 0.0001, E_real: 0.0489, E_fake: 0.0899

Epoch [6/10], Step [200], Data Loss: 0.0455, Reg Loss: 0.0001, E_real: -0.1238, E_fake: -0.1693

Epoch [6/10], Step [400], Data Loss: -0.0231, Reg Loss: 0.0001, E_real: -0.0970, E_fake: -0.0739

Epoch [6/10], Step [600], Data Loss: 0.0945, Reg Loss: 0.0001, E_real: 0.1775, E_fake: 0.0830

Epoch [6/10], Step [800], Data Loss: -0.0707, Reg Loss: 0.0001, E_real: -0.0260, E_fake: 0.0448

Epoch [7/10], Step [0], Data Loss: -0.0031, Reg Loss: 0.0001, E_real: 0.0709, E_fake: 0.0740

Epoch [7/10], Step [200], Data Loss: 0.0565, Reg Loss: 0.0001, E_real: 0.1467, E_fake: 0.0903

Epoch [7/10], Step [400], Data Loss: 0.0599, Reg Loss: 0.0001, E_real: -0.0484, E_fake: -0.1082

Epoch [7/10], Step [600], Data Loss: -0.0425, Reg Loss: 0.0001, E_real: 0.0756, E_fake: 0.1181

Epoch [7/10], Step [800], Data Loss: 0.0128, Reg Loss: 0.0001, E_real: -0.0665, E_fake: -0.0794

Epoch [8/10], Step [0], Data Loss: 0.0414, Reg Loss: 0.0001, E_real: -0.0406, E_fake: -0.0820

Epoch [8/10], Step [200], Data Loss: -0.1011, Reg Loss: 0.0001, E_real: -0.0461, E_fake: 0.0550

Epoch [8/10], Step [400], Data Loss: 0.0001, Reg Loss: 0.0001, E_real: -0.0730, E_fake: -0.0731

Epoch [8/10], Step [600], Data Loss: -0.0233, Reg Loss: 0.0001, E_real: 0.1501, E_fake: 0.1734

Epoch [8/10], Step [800], Data Loss: 0.0358, Reg Loss: 0.0001, E_real: -0.0905, E_fake: -0.1263

Epoch [9/10], Step [0], Data Loss: -0.0103, Reg Loss: 0.0001, E_real: 0.0663, E_fake: 0.0765

Epoch [9/10], Step [200], Data Loss: 0.1913, Reg Loss: 0.0001, E_real: -0.0111, E_fake: -0.2025

Epoch [9/10], Step [400], Data Loss: -0.0039, Reg Loss: 0.0001, E_real: -0.0531, E_fake: -0.0492

Epoch [9/10], Step [600], Data Loss: -0.1338, Reg Loss: 0.0001, E_real: -0.1126, E_fake: 0.0212

Epoch [9/10], Step [800], Data Loss: 0.0977, Reg Loss: 0.0001, E_real: -0.0752, E_fake: -0.1729

Epoch [10/10], Step [0], Data Loss: -0.0172, Reg Loss: 0.0001, E_real: -0.0045, E_fake: 0.0126

Epoch [10/10], Step [200], Data Loss: 0.0756, Reg Loss: 0.0001, E_real: -0.0432, E_fake: -0.1187

Epoch [10/10], Step [400], Data Loss: -0.0959, Reg Loss: 0.0001, E_real: 0.0591, E_fake: 0.1550

Epoch [10/10], Step [600], Data Loss: 0.1238, Reg Loss: 0.0001, E_real: -0.0296, E_fake: -0.1535

Epoch [10/10], Step [800], Data Loss: -0.0960, Reg Loss: 0.0001, E_real: -0.1330, E_fake: -0.0370

حال نگاهی به خروجی ها در epoch های مختلف می اندازیم:

Generated Epoch 1

تصویر3. چند نمونه تصادفی تولید شده در epoch اول

Generated Epoch 5

تصویر4. چند نمونه تصادفی تولید شده در epoch پنجم

Generated Epoch 10

تصویر5. چند نمونه تصادفی تولید شده در epoch دهم

از خروجی این تصاویر به خوبی می توان مشاهده کرد که مدل در ابتدا اصلا تصاویر خوب و مطلوبی تولید نمی کند اما رفته رفته کیفیت تصاویر بهتر می شود و واقعا شبیه به عدد می شوند، البته شایان ذکر است که بدلیل تعداد epoch های کم حتی در آخر هم مدل خیلی عملکرد خوبی ندارد.

Original Real Images

تصویر 6. تصاویر اصلی که از روی آن ها مدل تصاویر را خواهد ساخت

Langevin Output (initialized with Real)

تصویر7. خروجی مدل EBM روی تصاویر تصویر 6

در این مرحله، هدف بررسی این فرضیه است که اگر مدل EBM به درستی آموزش دیده باشد، داده‌های واقعی باید در نقاط کمینه انرژی (Local Minima) یا بسیار نزدیک به آن قرار داشته باشند. بنابراین، اگر فرآیند Langevin Sampling را با یک تصویر واقعی آغاز کنیم، تصویر نباید تغییر فاحشی کند، زیرا گرادیان انرژی در آن نقطه باید نزدیک به صفر باشد ($\nabla_x E(x) \approx 0$).

با مقایسه تصویر ورودی (Original Real Images) و خروجی (Langevin Output):

1. حفظ هویت (Identity Preservation):
   - اکثر ارقام هویت خود را کاملاً حفظ کرده‌اند. برای مثال، ارقام **۹**، **۶**، **۷** و **۱** در تصویر خروجی تقریباً بدون تغییر باقی مانده‌اند.
   - این نشان می‌دهد که این داده‌ها دقیقاً در "حوضچه جذب" (Basin of Attraction) مدل قرار دارند و مدل یاد گرفته است که این پیکربندی‌ها دارای انرژی پایین هستند.
2. تغییرات جزئی (Deterioration/Thinning):
   - برخی ارقام دچار تغییرات ظریفی شده‌اند. به طور خاص، ضخامت قلم کاهش یافته است.
   - به عنوان مثال، عدد **۵** (اولین تصویر از سمت چپ) و عدد **۸** (هشتمین تصویر) و عدد **۷** چهارمی از سمت راست در خروجی بسیار نازک‌تر شده‌اند و کمی از فرم استاندارد فاصله گرفته‌اند.
   - عدد **۳** (پنجمین تصویر) در خروجی کمی دفرمه شده و شبیه به یک خط عمودی یا عدد ۱ شده است.

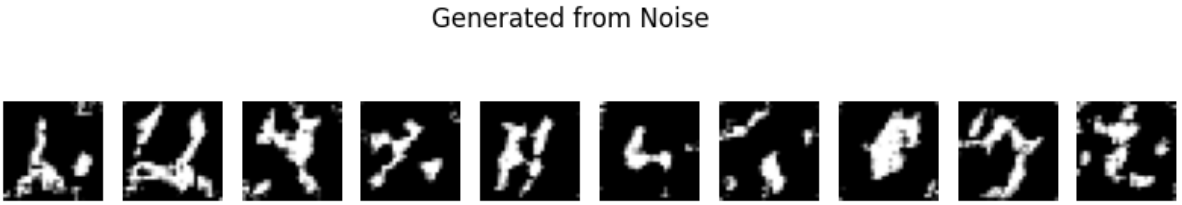
پایداری مدل با آغاز فرآیند Langevin Sampling از داده‌های واقعی ارزیابی شد. نتایج نشان می‌دهند که مدل با موفقیت توانسته است منظره انرژی (Energy Landscape) را به گونه‌ای شکل دهد که داده‌های واقعی در نواحی با انرژی پایین قرار گیرند.

۱. پایداری در کمینه‌های محلی: عدم تغییر ماهیت اکثر ارقام (مانند ۹، ۶ و ۷) تایید می‌کند که داده‌های آموزشی در نزدیکی نقاط مینیمم محلی تابع انرژی یاد گرفته شده قرار دارند. گرادیان انرژی در این نقاط کوچک است، بنابراین دینامیک Langevine تغییر زیادی در پیکسل‌ها ایجاد نمی‌کند.

۲. پدیده نازک‌شدن (Thinning): مشاهده می‌شود که مدل تمایل دارد تصاویر را به سمت نسخه نازک‌تر آن‌ها سوق دهد (مثلاً تبدیل ۸ ضخیم به یک ۸ بسیار نازک). این پدیده معمولاً ناشی از ماهیت تابع هزینه و نحوه تقریب گرادیان در روش Contrastive Divergence است که باعث می‌شود مدل دره‌های انرژی بسیار باریکی را در اطراف منیفولد داده‌ها حفر کند.

مدل EBM به پایداری قابل قبولی رسیده است، هرچند تمایل مدل به نازک کردن خطوط نشان می‌دهد که ممکن است برای جلوگیری از Overfitting روی ویژگی‌های فرکانس بالا (لبه‌های تیز)، نیاز به تنظیم دقیق‌تر پارامترهای رگولاریزیشن یا کاهش گام‌های نمونه‌برداری در زمان تست باشد.

زیر بخش سوم)



تصویر 8. نمونه های تولید شده از نویز خالص

تصویر9. نمونه های واقعی

با مقایسه تصاویر تولید شده با داده‌های واقعی ، موارد زیر قابل مشاهده است:

- تشخیص کلی: مدل توانسته است "مفاهیم کلی" ارقام را یاد بگیرد. ما می‌توانیم حدس بزنیم که برخی تصاویر شبیه به ۴، ۱، ۸ یا ۹ هستند. این یعنی مدل یاد گرفته است که تجمع جوهر در وسط تصویر انرژی کمتری نسبت به پراکندگی تصادفی دارد.
- کیفیت پایین و نویز: تصاویر تولید شده به شدت "دندانه‌دار" (Jagged)، "تکه تکه" (Fragmented) و دارای نویزهای فرکانس بالا هستند. برخلاف داده‌های واقعی که خطوطی صاف و پیوسته دارند، خروجی‌های مدل شبیه به قطرات جوهر پاشیده شده یا اشکال ناهموار هستند.
- بریدگی (Discontinuity): در بسیاری از موارد (مانند تصویر چهارم یا هشتم از سمت چپ در ردیف تولید شده)، خطوط عدد قطع شده‌اند و پیوستگی توپولوژیک ارقام حفظ نشده است.

علت کیفیت پایین چیست؟ :

- ناهمواری منظره انرژی (Rugged Energy Landscape): ما مدل را با روش **Contrastive Divergence** آموزش دادیم. این روش یک تقریب است. به دلیل ماهیت تقریبی CD و تعداد محدود گام‌های نمونه‌برداری در حین آموزش (مثلاً ۶۰ گام)، مدل نتوانسته است یک سطح انرژی کاملاً هموار بسازد. در عوض، مدل تعداد زیادی "چاه انرژی کاذب" (Spurious Local Minima) در اطراف داده‌های واقعی حفر کرده است.

- وقتی الگوریتم Langevin را از نویز شروع می‌کنیم، نمونه‌بردار در این چاله‌های کم‌عمق و ناهموار گیر می‌کند و نمی‌تواند به "قعرِ اصلی دره" (که تصویر صاف و تمیز است) برسد.

- مشکل اختلاط (Mixing Problem) در Langevin Dynamics: الگوریتم استاندارد Langevin Dynamics (بدون مکانیزم Annealing که در سوال دوم داریم)، توانایی کمی در عبور از موانع انرژی دارد. وقتی نمونه‌بردار از نویز شروع می‌کند، به سرعت به نزدیک‌ترین مد (Mode) جذب می‌شود، اما اگر آن مد یک تصویر دفرمه یا نویزی باشد، انرژی کافی برای خروج از آن و اصلاح تصویر را ندارد.
- عدم وجود قیدهای ساختاری: مدل‌های EBM کانولوشنی ساده، اگرچه توزیع مکانی پیکسل‌ها را یاد می‌گیرند، اما لزوماً "پیوستگی خطوط" را به عنوان یک ویژگی با انرژی بسیار پایین یاد نمی‌گیرند، مگر اینکه آموزش بسیار طولانی‌تر شود یا از روش‌های پیشرفته‌تر (مثل Regularization خاص) استفاده شود.

به عنوان نتیجه گیری می توان گفت خروجی‌های تولید شده نشان می‌دهند که مدل EBM توانسته است توزیع احتمالاتی داده‌های MNIST را تا حدودی تقریب بزند و اشکال شبیه به ارقام تولید کند (عدم وجود نویز خالص). با این حال، کیفیت پایین تصاویر، لبه‌های تیز و عدم پیوستگی خطوط نشان‌دهنده محدودیت‌های روش آموزش Contrastive Divergence و نمونه‌برداری Langevin ساده است. مدل به جای یادگیری یک منیفولد هموار، دچار مینیمم‌های محلی نویزی شده است که نمونه‌بردار در زمان Inference در آن‌ها به دام می‌افتد.

زیر بخش چهارم)

Original Images

تصویر 10. نمونه های واقعی

Noisy Input (Level 0.2)

تصویر 11. نمونه های واقعی نویزی شده با سطح 0.2

Denoised Output (Level 0.2)

تصویر 12. خروجی دینویز شده سطح 0.2 مدل

در تصاویر، نویز به صورت دانه‌های برفک روی تصویر دیده می‌شود، اما ارقام تا حدودی برای چشم انسان قابل تشخیص هستند. در تصاویر خروجی ، مدل عملکرد موفقی برای برخی اعداد داشته است. نویز پس‌زمینه کاملاً حذف شده و ارقام (مانند ۲، ۱، ۰) با وضوح بالا و خطوط پیوسته بازیابی شده‌اند. در این سطح نویز، داده‌ها همچنان در حوضچه جذب (Basin of Attraction) صحیحِ تابع انرژی قرار دارند. گرادیان‌های انرژی ($\nabla_x E(x)$) به درستی جهت حرکت به سمت منیفولد داده‌های واقعی را نشان می‌دهند و مدل توانسته تصویر را تمیز کند.

Noisy Input (Level 0.4)

تصویر 13. نمونه های واقعی نویزی شده با سطح 0.4

Denoised Output (Level 0.4)

تصویر 14. خروجی دینویز شده سطح 0.4 مدل

در تصاویر ورودی، نویز شدت گرفته و برخی خطوط نازک (مثل عدد ۱) کم‌رنگ شده‌اند. در خروجی، افت کیفیت محسوسی مشاهده می‌شود. مدل تمایل دارد ارقام را بسیار نازک کند. به عنوان مثال، عدد ۲ (دومین تصویر) و عدد ۰ (چهارمین تصویر) دچار شکستگی شده‌اند و پیوستگی خود را از دست داده‌اند. عدد ۹ (آخرین تصویر) به شدت آسیب دیده و شبیه به یک خط کج شده است. مدل در مرز شکست قرار دارد. اگرچه نویز پس‌زمینه حذف شده، اما مدل قادر به حفظ توپولوژی (Topology) ارقام نیست. این نشان می‌دهد که دره‌های انرژی یاد گرفته شده توسط مدل EBM بسیار "باریک" هستند و نویز ۰.۴ گاهی داده را از این دره به بیرون پرتاب می‌کند یا مدل به اشتباه "پاک کردن پیکسل‌ها" را راهی برای کاهش انرژی می‌داند.

Noisy Input (Level 0.6)

تصویر 15. نمونه های واقعی نویزی شده با سطح 0.6

Denoised Output (Level 0.6)

تصویر 16. خروجی دینویز شده سطح 0.6 مدل

ورودی‌ها بسیار نویزی هستند و تشخیص برخی ارقام دشوار است. خروجی نشان‌دهنده شکست کامل مدل در اکثر موارد است. عدد ۲ (دومین تصویر) تبدیل به یک خط مارپیچ بی‌معنی شده است. عدد ۴ (پنجمین تصویر) تقریباً محو شده است. عدد ۰ (چهارمین تصویر) از دو قطعه جدا از هم تولید شده. تقریباً تمام ارقام خروجی، تکه تکه، بسیار نازک و غیرقابل تشخیص هستند.

در این سطح، نویز آنقدر زیاد است که داده را از محدوده گرادیان‌های مفید خارج کرده است. مدل EBM (برخلاف Score-Based Models که با نویزهای بزرگ آموزش دیده‌اند) در مواجهه با نقاط دور از منیفولد، گرادیان دقیقی ارائه نمی‌دهد. همچنین، فرآیند Langevin Sampling به جای بازیابی عدد، در مینیمم‌های محلی نامطلوب گیر کرده است.

سؤال دوم

بخش اول

زیر بخش اول)

تابع امتیاز اینگونه تعریف می شود:

$$s_\theta(x) = \nabla_x log P_\theta(x)$$

برای اثبات استقلال از $Z(\theta)$ فرض میکنیم که $P_\theta(x) = \frac{e^{-E_\theta(x)}}{Z(\theta)}$. پس داریم:

$$\log P_\theta(x) = -E_\theta(x) - \log(Z(\theta))$$

حال تابع امتیاز را محاسبه می کنیم:

$$s_\theta(x) = \nabla_x log P_\theta(x) = -\nabla_x(E_\theta(x)) - \nabla_x log(Z(\theta))$$

عبارت دوم بر اساس x نیست پس گرادیانش صفر است. پس:

$$s_\theta(x) = -\nabla_x(E_\theta(x))$$

بزرگترین چالش در آموزش مدل‌های انرژی‌محور و احتمالاتی، محاسبه $Z(\theta)$ (انتگرال روی کل فضای داده) است که معمولاً غیرممکن (Intractable) است. با استفاده از مدل‌سازی تابع امتیاز (Score Matching)، ما مستقیماً گرادیان را یاد می‌گیریم و نیازی به محاسبه ثابت نرمال‌سازی $Z(\theta)$ نداریم. این باعث می‌شود آموزش بسیار سریع‌تر و پایدارتر شود.

زیر بخش دوم)

تابع هزینه Score Matching شامل ترم $E_{P_{data}}[tr(\nabla_x s_\theta(x))]$ است. این ترم نیازمند محاسبه Trace of Jacobian خروجی مدل نسبت به ورودی است. برای داده‌های با ابعاد بالا (مثل تصاویر)، محاسبه این ترم بسیار پرهزینه است زیرا نیاز به D بار انتشار به عقب (Backpropagation) دارد (که D تعداد پیکسل‌هاست). به بیان ساده، محاسبات از مرتبه $O(D^2)$ یا نیازمند محاسبات سنگین مشتقات مرتبه دوم است که برای شبکه‌های عصبی عمیق عملی نیست.

روش DSM پیشنهاد می‌کند که به جای تطبیق امتیاز روی داده‌های خام، داده‌ها را با یک نویز (مثلاً گوسی) مخدوش کنیم ($\tilde{x} \sim q_\sigma(\tilde{x}|x)$) و سپس سعی کنیم امتیازی را یاد بگیریم که بتواند نویز را تخمین بزند (یا مسیر برگشت به داده سالم را پیدا کند).

وینسنت (Vincent) ثابت کرد که مینیمم کردن خطای تطبیق امتیاز روی داده‌های نویزی، معادل مینیمم کردن خطای اصلی است.

منطق آن این است که:

$$\nabla_{\tilde{x}} \log q_\sigma(\tilde{x}|x) \approx \frac{x-\tilde{x}}{\sigma^2}$$

یعنی گرادیان لگاریتم توزیع نویزی (Score)، متناسب است با برداری که از داده نویزی $\tilde{x}$ به سمت داده سالم x اشاره می‌کند. پس یادگیری Score در اینجا عملاً معادل یادگیری یک Denoising Autoencoder است که سعی می‌کند نویز را حذف کند. چون ما x (داده سالم) و $\tilde{x}$ (داده نویزی) را داریم، محاسبه گرادیان هدف ساده است و نیازی به محاسبه اثر ماتریس ژاکوبین نیست.

زیر بخش سوم)

الف)اگر x در ناحیه ۱ باشد: $p_{data}(x) \approx \pi p_1(x)$

$$\nabla_x \log p_{data}(x) = \nabla_x \log \pi + \nabla_x p_1(x) = \nabla_x p_1(x)$$

مشاهده می‌کنیم که ضریب π (که وزن کلاس‌ها را تعیین می‌کند) در مشتق لگاریتم حذف می‌شود (چون عدد ثابت است). بنابراین، تابع امتیاز در هر ناحیه هیچ اطلاعاتی درباره نسبت وزنی (π) آن ناحیه نسبت به ناحیه دیگر ندارد.

ب)الگوریتم Langevin Dynamics برای حرکت از یک مُد (Mode) به مُد دیگر نیاز دارد که از ناحیه بین آن‌ها عبور کند. چون تکیه‌گاه‌ها جدا هستند، احتمال حضور داده در ناحیه بین دو مُد تقریباً صفر است و گرادیان (Score) در آنجا یا صفر است یا تعریف نشده (ناحیه کم‌چگالی). بنابراین، وقتی نمونه‌بردار در یکی از مُدها شروع به کار می‌کند، هیچ سیگنال گرادیانی وجود ندارد که آن را به سمت مُد دیگر هدایت کند. در نتیجه، مدل نمی‌تواند نسبت‌های صحیح ($\pi, 1 - \pi$) را بازتولید کند و تمام نمونه‌ها در همان مُد اولیه گیر می‌افتند (Mixing Problem).

زیر بخش چهارم)

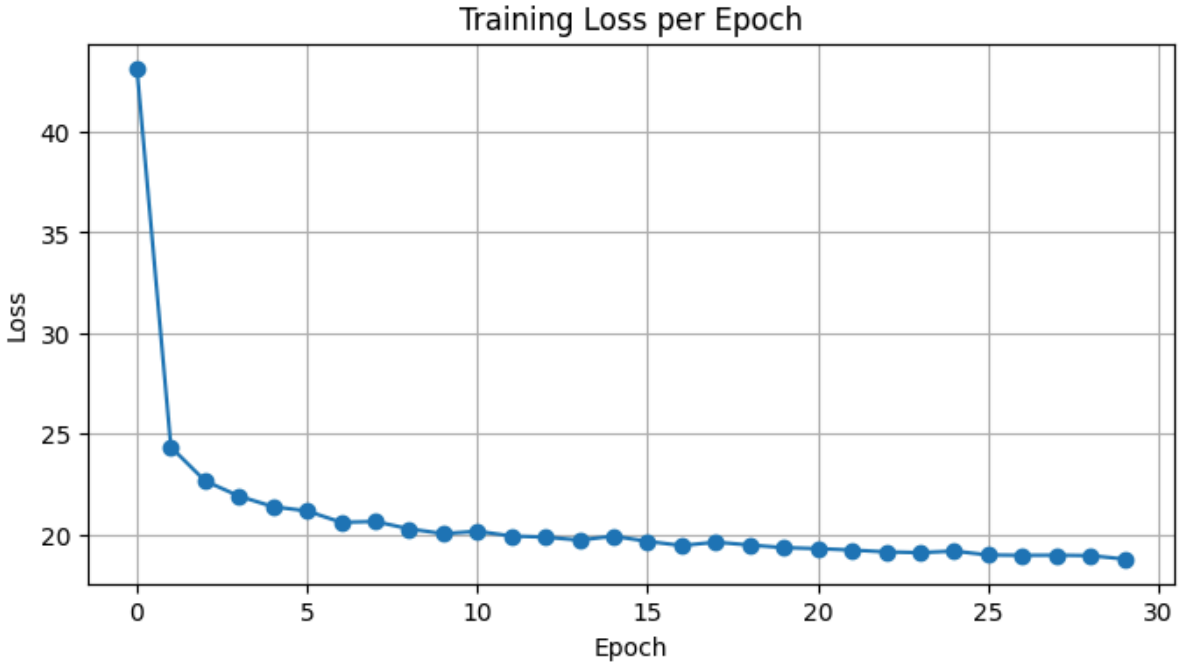
الف) مشکلات روی داده‌های واقعی (Manifold Hypothesis): داده‌های واقعی (مثل تصاویر) معمولاً روی یک منیفولد با ابعاد پایین در فضای پیکسلی قرار دارند.

- مشکل ۱: تابع امتیاز فقط روی منیفولد تعریف شده است و در فضای خارج از آن (Ambient Space) تعریف‌نشده است. این باعث می‌شود مدل در خارج از منیفولد نداند به کدام سمت برود.
- مشکل ۲: نمونه‌برداری Langevin به دلیل نواحی با چگالی پایین (Low Density Regions) بین مُدها، نمی‌تواند به خوبی در کل فضا جابجا شود.

ب) یک دنباله از نویزها $\sigma_1 > \sigma_2 > \cdots > \sigma_L$ در نظر می‌گیرد. نویز بزرگ (σ_1) کل فضا را پر می‌کند و مُدها را به هم وصل می‌کند. این باعث می‌شود گرادیان در همه جا تعریف شود و نمونه‌بردار بتواند گام‌های بزرگ بردارد و از یک مُد به مُد دیگر بپرد. نویز کوچک (σ_L): نویز را کم می‌کند تا جزئیات دقیق تصویر شکل بگیرد و نمونه روی منیفولد داده‌های واقعی بنشیند.

بخش دوم

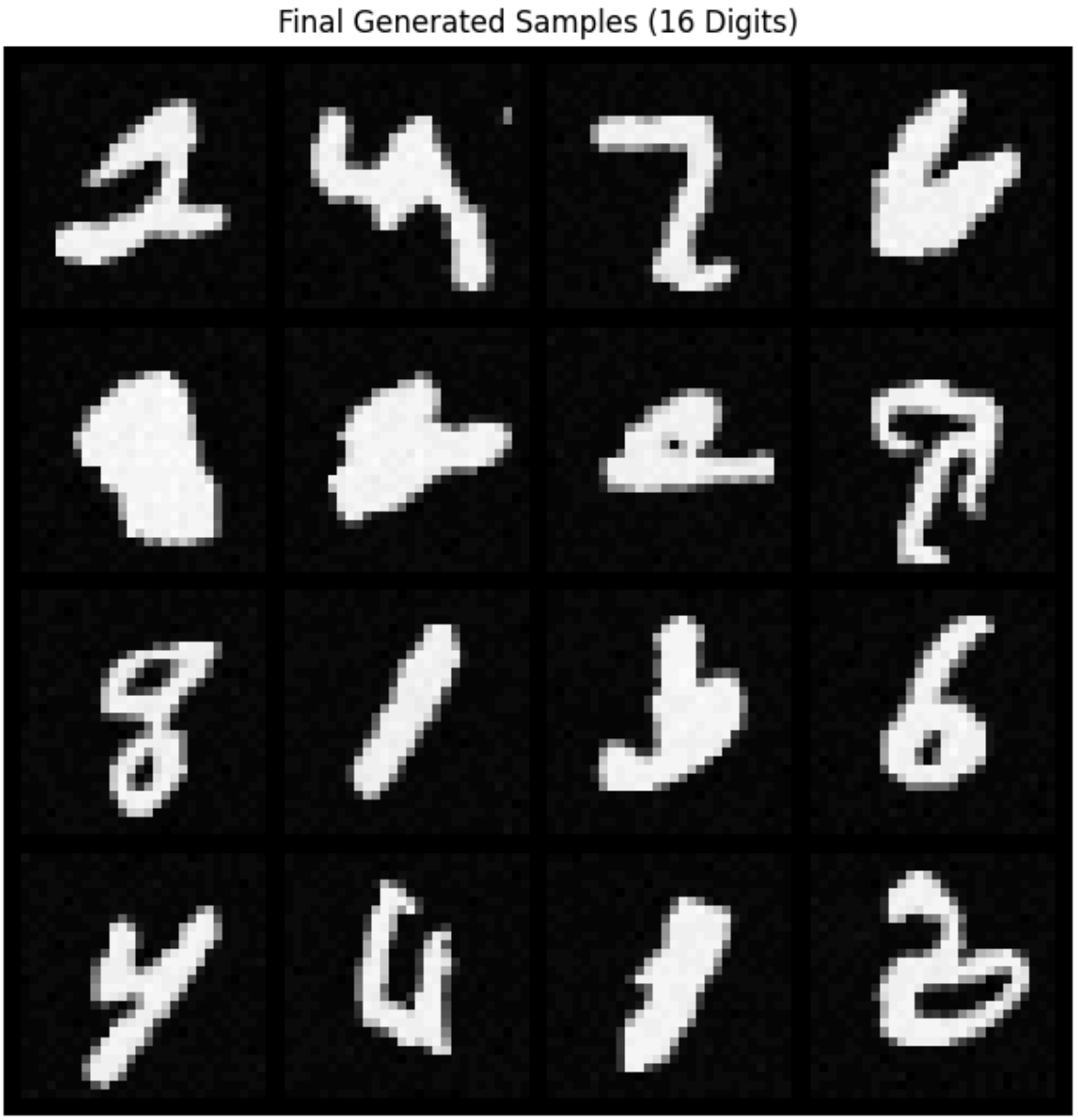
زیر بخش اول)



تصویر 17. نمودار زیان مدل

نمودار نشان‌دهنده یک روند نزولی بسیار سالم و پایدار است. خطا از مقدار اولیه حدود ۴۳ در ایپاک اول به سرعت به حدود ۲۴ کاهش یافته و سپس با یک شیب ملایم تا ایپاک ۳۰ به مقدار حدود ۱۹ می‌رسد.

افت شدید اولیه نشان می‌دهد که شبکه به سرعت یاد گرفته است جهت‌های کلی گرادیان را (برای حذف نویزهای بزرگ) تشخیص دهد. شیب ملایم در ادامه نشان‌دهنده یادگیری جزئیات دقیق (Fine-tuning) برای سطوح نویز پایین (σ_{end}) است. عدم وجود نوسانات شدید (Spikes) نشان می‌دهد که نرخ یادگیری ($2e^{-4}$) و اندازه بچ مناسب انتخاب شده‌اند.



تصویر 18. تصویر Grid شامل 16 عکس تولید شده

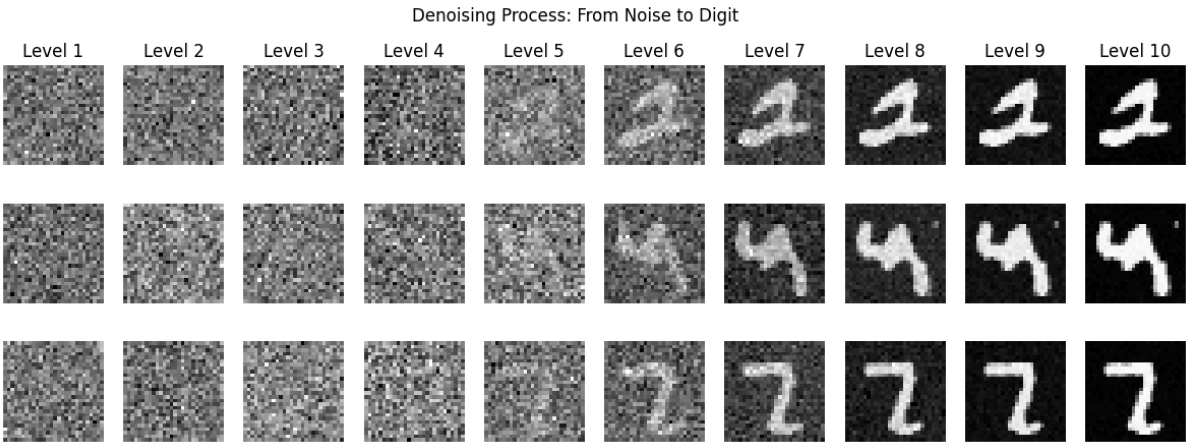
کیفیت (Quality): کیفیت تصاویر بسیار بالاتر از بخش اول (EBM) است. خطوط پیوسته هستند (بدون بریدگی که در EBM دیدیم) و اعداد کاملاً واضح‌اند.

تنوع (Diversity): در گرید ۱۶ تایی، تنوع خوبی از اعداد دیده می‌شود (۱، ۲، ۴، ۵، ۶، ۷، ۸، ۹). این نشان می‌دهد مدل دچار Mode Collapse نشده است.

مقایسه با واقعیت: تصاویر تولید شده (مانند عدد ۸ در ردیف سوم یا عدد ۴ در ردیف دوم) بسیار شبیه به دست‌خط‌های واقعی هستند و اعوجاج غیر طبیعی ندارند.

نویز پس‌زمینه: در برخی تصاویر (مثل گوشه‌های گرید) هنوز مقدار بسیار کمی نویز دانه‌دانه (Speckle noise) در پس‌زمینه سیاه دیده می‌شود که طبیعی است و با افزایش تعداد گام‌ها (T) قابل رفع است.

در کل تصاویر نهایی تولید شده توسط مدل ScoreNet کیفیت بصری بسیار بالایی دارند. برخلاف روش‌های سنتی، خطوط ارقام پیوسته و بدون شکستگی هستند. استفاده از معماری U-Net به همراه مکانیزم FiLM باعث شده است که مدل بتواند همزمان ویژگی‌های کلی و جزئیات دقیق را مدیریت کند. تنوع ارقام تولید شده نشان می‌دهد مدل توزیع احتمال فضای داده را به خوبی پوشش داده است.



تصویر 19. فرآیند تبدیل 3 تصویر از نویز به عکس نهایی

مراحل اولیه (Level 1-4): تصاویر تقریباً نویز خالص هستند. در این مرحله σ بزرگ است. این نویز زیاد باعث می‌شود نمونه‌بردار بتواند آزادانه در فضا حرکت کند و از موانع انرژی عبور کند (حل مشکل Mixing).

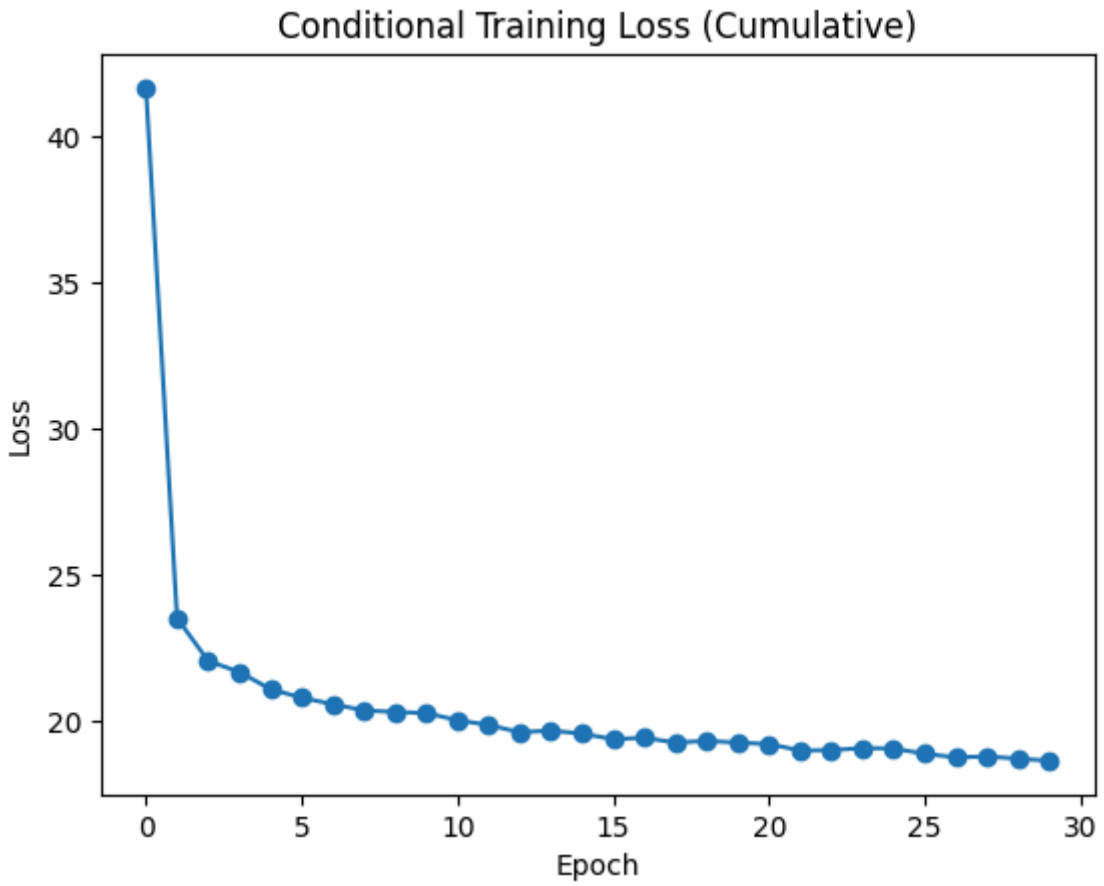
مراحل میانی (Level 5-7): توده‌های جوهری (Blobs) شروع به شکل‌گیری می‌کنند. شبکه شروع به هل دادن نویز به سمت منیفولد داده‌ها می‌کند. مثلاً در ردیف اول، شکل کلی عدد "2" و در ردیف دوم "4" ظاهر می‌شود.

مراحل پایانی (Level 8-10): نویزهای ریز پس‌زمینه حذف می‌شوند و لبه‌های اعداد تیز (Sharp) می‌شوند. نمونه نهایی کاملاً روی توزیع داده‌های MNIST نشسته است.

روند تبدیل نویز خالص به عدد نهایی طی ۱۰ سطح نویز، اثربخشی روش Annealed Langevin Dynamics را نشان می‌دهد. همانطور که مشاهده می‌شود، در سطوح اولیه (نویز بالا)، مدل ساختارهای کلی را جستجو می‌کند و با کاهش تدریجی σ، نمونه‌ها به سمت چگالی‌های بالای داده هدایت می‌شوند. این روش سلسله‌مراتب نویز باعث می‌شود مدل برخلاف EBM های ساده، در مینیمم‌های محلی کم‌عمق گیر نکند و بتواند ارقام پیچیده را به درستی بازسازی کند.

البته واضح است که مدل همچنان نواقصی دارد و سطر آخر هم گواه این مدعاست چرا که عدد خروجی چیزی بین 7 و 2 است و تشخیص دقیق آن کمی مشکل است.

زیر بخش دوم)



تصویر 20. نمودار زیان مدل شرطی

نمودار نشان می‌دهد که خطا از مقدار حدود ۴۲ شروع شده و با یک شیب تند در ۵ ایپاک اول به حدود ۲۱ می‌رسد. سپس با یک روند نزولی ملایم و بسیار پایدار ادامه می‌یابد و در ایپاک ۳۰ به مقدار زیر ۱۹ می‌رسد.

عدم وجود نوسانات شدید در نمودار نشان‌دهنده تنظیم مناسب نرخ یادگیری و اندازه Batch است.

اگر دقت کنید، مقدار نهایی خطا (Loss ~18.5) تقریباً مشابه یا اندکی بهتر از حالت غیرشرطی است. این منطقی است؛ زیرا اگرچه معماری کمی پیچیده‌تر شده (اضافه شدن امبدینگ کلاس)، اما اطلاعات اضافی (لیبل کلاس) به شبکه کمک می‌کند تا ابهام کمتری در فرآیند نویززدایی داشته باشد (مثلاً شبکه می‌داند که این نویز باید تبدیل به یک "۸" شود، نه اینکه بین "۸" و "۳" مردد بماند).

تصویر 21. تصاویر خروجی مدل شرطی

این تصویر مهم‌ترین خروجی این بخش است که نشان‌دهنده موفقیت آمیز بودن شرطی‌سازی است.

- موفقیت در کنترل تولید:
  - ردیف ۰: تمام تصاویر تولید شده، بدون استثنا، عدد صفر هستند.
  - ردیف ۱ تا ۹: به همین ترتیب، هر ردیف دقیقاً و منحصراً شامل همان عددی است که درخواست شده است.
  - نتیجه: این نشان می‌دهد که بردار امبدینگ کلاس (e^y) به طور موثری بر خروجی شبکه تأثیر گذاشته و جهت حرکت Langevin Dynamics را به سمت منیفولد خاص آن کلاس هدایت کرده است.
- تنوع درون کلاسی (Intra-class Diversity):

- به ردیف ۲ نگاه کنید: برخی ۲ ها حلقه دارند، برخی ساده هستند.
- به ردیف ۷ نگاه کنید: برخی ۷ ها خطخورده هستند و برخی نه.
- تحلیل: مدل دچار Overfitting نشده است و توانسته است تنوع موجود در هر کلاس را یاد بگیرد. مدل صرفاً یک تصویر خاص از عدد ۷ را کپی نمی‌کند، بلکه از توزیع شرطی $p(x|y = 7)$ نمونه‌برداری می‌کند.

- کیفیت تصاویر:
  - ارقام بسیار واضح، با کنتراست بالا و خطوط پیوسته هستند. کیفیت تصاویر شرطی در مقایسه با تصاویر تولید شده از نویز در بخش قبل (Unconditional) معمولاً کمی بالاتر و شارپ‌تر به نظر می‌رسد. دلیل نظری این است که فضای جستجو برای مدل محدودتر شده است (جستجو در زیر-فضای "عدد پنج" راحت‌تر از جستجو در کل فضای "همه اعداد" است).

در کل میتوان گفت که در تصویر گرید شرطی، هر سطر متناظر با یک کلاس خاص (۰ تا ۹) است. مشاهده می‌شود که مدل با موفقیت توانسته است فرآیند تولید را کنترل کند و در هر سطر دقیقاً عدد خواسته شده را تولید نماید. هیچ‌گونه تداخل کلاسی (مثلاً ظاهر شدن عدد ۳ در سطر ۵) دیده نمی‌شود که نشان‌دهنده قدرت تفکیک بالای امبدینگ‌های کلاس است.

مقایسه با مدل پایه (Unconditional):

۱. کنترل‌پذیری: برخلاف مدل بخش قبل که خروجی کاملاً تصادفی بود، در اینجا ما تسلط کامل بر خروجی داریم.

۲. کیفیت و قطعیت: تصاویر تولید شده در حالت شرطی، ساختار منسجم‌تر و قوی‌تری دارند. اطلاعات کلاس مانند یک راهنما عمل کرده و از سردرگمی مدل در نواحی مرزی (مثلاً بین عدد ۱ و ۷) جلوگیری می‌کند.

۳. تنوع: همانند مدل پایه، تنوع حفظ شده است و Mode Collapse رخ نداده است.

مراجع