دانشگاه اصفهان

مستندات پروژه پایانی درس یادگیری ماشین

علیرضا ساعی

۹۹۳۶۱۳۰۲۶

سامان امیدی

۹۹۳۶۶۳۰۰۹

میترا عمرانی

۹۹۳۶۱۳۰۴۷

مقاله اول

Denoising Diffusion Probabilistic Models

در این مقاله به صورت کلی به بحث دیفیوژن پرداخته شده که چگونه با استفاده از مدل های احتمالاتی دیفیوژن دارای متغیرهای پنهان میباشد، میتوان عکسهای مصنوعی را با استفاده از لایههای متعدد با دینویز کردن ساخت. این مقاله روی دیتاستهای CIFAR10 و LSUN اسکورهای قابل ملاحظهای بدست آورده است:

Inception Score = 9.46

FID = 3.17

از این اسکورهای برای ارزیابی مدلهای دیفیوژن استفاده میشود. این اسکوردهی به 3 قسمت براساس نوع دیفیژن مدل تقسیم میشود:

1. تولید عکس با متن
2. تولید عکس با متن با شرط عکس ورودی
3. تولید عکس با کلاس-شرطی

حال با توجه به توضیحات داریم که هرکدام خوبیها و بدیهایی نسبت به یکدیگر دارند:

Inception Score (IS):

یک معیار برای ارزیابی کیفیت تصاویر تولیدی توسط یک مدل مولد است. این معیار توسط مدل Inception-v3، که یک مدل کلاسیفیکیشن تصویر است، محاسبه می‌شود.

در این روش میتوان دیتاست inception v3 را از کتابخانه TensorFlow دانلود کرد، سپس عکسهای خود را تولید کرده و به مدلی که با این دیتاست train شده است میدهیم تا عمل کلاسیفیکیشن را انجام دهد. سپس یک امتیاز با توجه به احتمالات پیش‌بینی مدل Inception-

v3 برای هر تصویر محاسبه می‌شود. این احتمالات نشان‌دهنده میزان اطمینان مدل Inception-v3 در دسته‌بندی تصاویر هستند. به صورت میانگین اطمینان تقسیم شده بر لگاریتم طبیعی از میانگین احتمالات تعلق تصاویر به دسته‌ها محاسبه می‌شود.

Frechet Inception Distance (FID):

یک معیار دیگر برای ارزیابی کیفیت تصاویر تولیدی است. این معیار با مقایسه توزیع واقعی تصاویر با توزیع تولیدی توسط مدل، میزان اختلاف بین این دو توزیع را اندازه‌گیری می‌کند.

در این روش همانند روش امتیازدهی قبلی یک مدلی را با دیتاست Inception v3 عمل training را انجام میدهیم. سپس عکس تولیدی و عکسهای دیتاست را به این مدل میدهیم تا ویژگیهای عکسهارا با استفاده از یک لایهی خاص مدلمان استخراج کنیم. محاسبه فاصله Fréchet بین دو توزیع چندمتغیره گاوسی که توسط ویژگی‌های تصاویر واقعی و تصاویر تولیدی توصیف می‌شوند. میانگین و ماتریس کوواریانس این دو توزیع محاسبه می‌شود و سپس فاصله Fréchet بر اساس این میانگین و ماتریس کوواریانس محاسبه می‌گردد.

در اینجا مدل را به صورت یه زنجیره مارکوف نشان داده است که اگر از چپ به راست برویم عمل دینویز کردن رخ میدهد و اگر از راست به چپ برویم عمل نویزدار کردن تصویر رخ میدهد. بدین صورت که X_T یه تصویر کاملا نویزی و X_0 تصویر بدون نویز میباشد. در مقاله به این اشاره شده بود که این روش Lossy میباشد، یعنی همواره با از دست رفتن برخی اطلاعات در این پروسه روبهرو هستیم.

کدی که در این مقاله نوشته شده است از کتابخانه TensorFlow استفاده کرده است و معماری شبکه آنها U-Net میباشد. به طور کلی عملکرد Unet از شکل حرف U منشا گرفته است. یعنی ابتدا ورودی را بسیار abstract کرده و سپس به حالت قبلی برمیگردانیم. ابتدا یک فایل nn.py ساختهاند که لایههای مورد استفاده را با استفاده از TensorFlow در قالب تابع ساختهاند سپس از توابع این کد در شبکه عصبی استفاده شده و کد unet.pyدر این مقاله موارد زیر را داراست:

1. توابع نالینیریتی و نرمالایز کردن:

- Nonlinearity این تابع از Swish برای نالینیریتی استفاده می‌کند.
- normalize: این تابع از Group Normalization برای نرمالایز کردن ویژگی‌ها استفاده می‌کند.

2. توابع افزایش و کاهش اندازه تصویر

- Upsample: این تابع از متد نزدیکترین همسایگی برای افزایش اندازه تصویر استفاده می‌کند و در صورت نیاز به یک لایه کانولوشن اضافی نیز امکان پذیر است.
- DownSample: این تابع برای کاهش اندازه تصویر با استفاده از لایه‌های کانولوشن یا میانگین‌گیری در نظر گرفته شده است

3. قسمت ResNet:

- این تابع یک بلوک ResNet را پیاده‌سازی می‌کند که شامل دو لایه کانولوشن، توابع نالینیریتی، نرمالایزیشن، و shortcut می‌شود.

4. قسمت Attention

- این تابع یک بلوک Attention را پیاده‌سازی می‌کند که از عملیات توجه بر روی ویژگی‌های ورودی استفاده می‌کند.

5. قسمت اصلی کد

- این تابع معماری کلی مدل را پیاده‌سازی می‌کند. این مدل شامل لایه‌های کانولوشن، بلوک‌های ResNet، و بلوک‌های Attention است که با هم تصاویر را از دنباله‌های زمانی ویژگی‌ها تولید می‌کند. این مدل از توابعی مانند توابع نان‌لینیریتی، نرمالایزیشن، افزایش و کاهش اندازه تصویر، و بلوک‌های ResNet و Attention برای انجام این کارها استفاده می‌کند.

مقاله دوم

Common Diffusion Noise Schedules and Sample Steps are Flawed

در این مقاله، نویسندگان اشکالات در طراحی‌های معمول(Diffusion Noise Schedules) را مورد بررسی قرار داده‌اند که باعث عدم اجبار آخرین گام زمانی به داشتن نسبت سیگنال به نویز (SNR) صفر می‌شوند. برخی از پیاده‌سازی‌های نمونه‌بردار Diffusion نیز از گام زمانی آخرین روند شروع می‌کنند. این طراحی‌ها به اشکالی منجر می‌شوند که مدل تفاوت‌هایی بین فرایند آموزش و استنتاج داشته باشد. این تفاوت ممکن است باعث مشکلات جدی در پیاده‌سازی‌های موجود شود. به عنوان مثال، در یک مدل به نام "Stable Diffusion"، این اشکالات باعث محدود شدن مدل به تولید تصاویر با روشنایی متوسط می‌شود و اجازه تولید تصاویر بسیار روشن یا تاریک را نمی‌دهد. نویسندگان در ادامه، چندین راه‌حل ساده ارائه کرده‌اند تا این اشکالات را برطرف کنند، از جمله: (1) تغییر برنامه نویز به منظور اجبار به داشتن نسبت SNR صفر در گام زمانی آخرین؛ (2) آموزش مدل با پیش‌بینی مقدار v؛ (3) تغییر نمونه‌بردار به نحوی که همیشه از گام زمانی آخرین شروع شود؛ (4) تغییر راهنمای بدون نیاز به تصویر به منظور جلوگیری از بیش‌اندازه شدن در توسعه. این تغییرات ساده اطمینان می‌دهند که فرآیند نویز مطابقت دارد و مدل قادر است تصاویری تولید کند که با توزیع اصلی داده‌ها بهتر همخوانی دارند.

Signal To Noise Ration

این معیار نسبت مقدار سیگنال به مقدار نویز در یک سیستم یا سیگنال مشخص را اندازه‌گیری می‌کند و نمایانگر نسبت قدرت یا انرژی سیگنال به نویز در آن سیستم است. در اینجا، چند جنبه اصلی SNR با استفاده از "نویز" توضیح داده می‌شود:

- **سیگنال** (Signal): اطلاعات مورد نظر در یک سیستم یا سیگنال. در حوزه تصویر، مثلاً، سیگنال می‌تواند اطلاعات تصویر باشد.

- **نویز** (Noise): سیگنال‌های تصادفی یا اطلاعات غیرمطلوب که در سیستم حضور دارند و می‌توانند اندازه‌گیری سیگنال را مخلوط کنند.

این مقاله 4 روشی که اشاره کردیم را دقیق تر توضیح میدهد:

روش اول

در این بخش، مقاله به بررسی اشکالات برنامه‌های نوفه مشترک می‌پردازد و نشان می‌دهد که هیچکدام از این برنامه‌ها نسبت سیگنال به نویز (SNR) صفر در گام زمانی آخر را اجبار نمی‌دهند. به علاوه، برنامه نویز کوسینوس با عمدتاً کلیپ کردن βt به اندازه‌ای که از 0.999 بیشتر نشود، جلوی رسیدن SNR به صفر را می‌گیرد. مشکلات خاصی در برنامه نویز مورد استفاده در مدل Stable Diffusion نیز مشاهده می‌شود. مقاله نشان می‌دهد که SNR در گام زمانی آخر از صفر فاصله دارد. این اشکالات باعث ایجاد یک فاصله بین آموزش و استنتاج می‌شود. به عبارت دیگر، در حین آموزش (زمان $t = T$)، ورودی مدل به کلی نویز خالص نیست و مقدار کمی از سیگنال همچنان در آن وجود دارد. این سیگنال نشتی شده شامل اطلاعات با فرکانس پایین‌تر می‌باشد، مانند میانگین کلی هر کانال. در نتیجه، مدل در ادامه یاد می‌گیرد که از نویز خروجی که حاوی میانگین نشتی شده است، احیاء کند. در حین استنتاج، برای نمونه‌برداری از نویز گوسی خالص استفاده می‌شود که همیشه نسبت سیگنال به نویز صفر دارد.

روش دوم

SNR صفر است، پیش‌بینی ϵ تبدیل به یک وظیفه ساده می‌شود و زیان ϵ نمی‌تواند مدل را به یادگیری چیزهای معنی‌دار درباره داده‌ها هدایت کند. در اینجا به پیشنهاد می‌پردازیم و به جای استفاده از پیش‌بینی ϵ و زیان ϵ، از پیش‌بینی v و زیان v استفاده می‌کنیم. این روش به این صورت است که:

$$V_t = sqrt(a_t . \epsilon) - sqrt(1 - a_t . x_0)$$

در $t = T$ داریم:

$$a_T = 0$$

در این گام خاص، مدل وظیفه دی‌نویزینگ را انجام نمی‌دهد زیرا ورودی هیچ سیگنالی ندارد. به جای آن، مدل به‌طور معکوس به پیش‌بینی میانگین توزیع داده‌ها تشویق می‌شود. نتایج نشان می‌دهد که آموزش مدل Stable Diffusion با خسارت v مانند استفاده از خسارت ϵ است. توصیه می‌شود همیشه از پیش‌بینی v برای مدل استفاده شود و در صورت نیاز به تنظیم وزن خسارت (λ_t) برای دست‌یابی به وزن‌های مختلف خسارت از آن استفاده شود

روش سوم

در این بخش، مقاله به بررسی مسئله نمونه‌برداری از گام زمانی آخر می‌پردازد. بسیاری از اجراها، از جمله پیاده‌سازی‌های رسمی DDIM و PNDM ، گام زمانی آخر را به درستی در فرآیند نمونه‌برداری در نظر نمی‌گیرند. این امر نیز نادرست است چرا که مدل‌ها در گام‌های زمانی کمتر از T آموزش می‌بینند ورودی‌هایی با نسبت سیگنال به نویز (SNR) غیرصفر، بنابراین با

رفتار استنتاجی متفاوت هستند. مانند مشکل بحران نور در Stable Diffusion که در بررسی روش اول اعلام شد، عدم در نظر گرفتن گام زمانی آخر در اینجا نیز مشکلاتی ایجاد می‌کند. مقاله به این نتیجه می‌رسد که نمونه‌برداری از گام زمانی آخر در همراه با یک برنامه نویز که نسبت SNR صفر را اعمال می‌کند، حیاتی است. به این ترتیب، زمانی که نویز گوسی خالص به مدل در گام نمونه‌برداری اولیه داده می‌شود، مدل واقعاً به یادگیری این ورودی در استنتاج آموزش دیده است.

روش چهارم

در این بخش، مقاله به مشکلاتی می‌پردازد که در حالتی که نسبت SNR به صفر نزدیک است، راهنمای بدون کلاسیفایر بسیار حساس می‌شود و می‌تواند باعث اضافه شدن بیش از حد نور به تصاویر شود. مشکلات مشابهی در کارهای دیگر نیز مشاهده شده است. برای مثال، مدل Imagen از برنامه نویز کسینوس استفاده می‌کند که SNR گام زمانی آخر به صفر نزدیک است و روشی به نام "تنظیم پویا" پیشنهاد می‌دهد تا مشکل اضافه شدن بیش از حد نور را حل کند. با الهام از این روش، مقاله یک راه جدید برای تغییر مجدد راهنمای بدون دسته‌بند ارائه می‌دهد که قابل استفاده در هر دو مدل فضای تصویر و مدل فضای پنهان است.