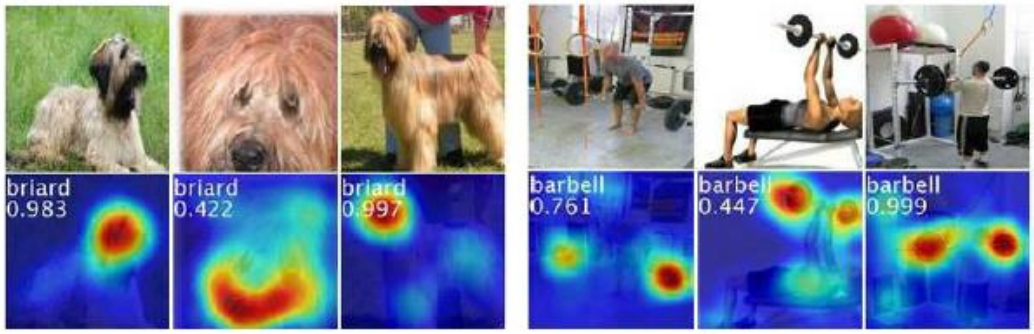
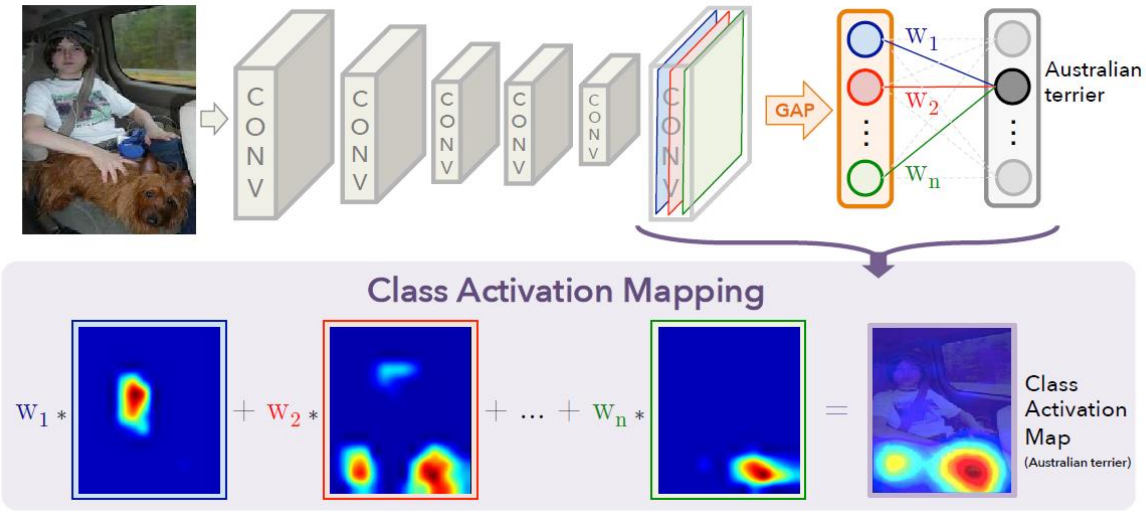
(۱

class activation map یا CAM با استفاده از لایه‌ی global average pooling در شبکه‌های CNN ساخته می‌شود. CAM برای یک دسته خاص، نواحی متمایز کننده تصویر استفاده شده توسط CNN برای شناسایی آن دسته را نشان می‌دهد. به عنوان مثال در شکل زیر CAMها نواحی متمایز کننده تصویر که برای طبقه‌بندی تصویر استفاده شده یعنی، سر حیوان برای سگ برید و صفحات در هالتر، را برجسته می‌کنند.



از معماری مشابه Network in Network و گوگل نت استفاده می کنیم. شبکه عمدتاً از لایه‌های کانولوشنال تشکیل شده است و درست قبل از لایه خروجی نهایی (در مورد طبقه بندی softmax)، global average pooling را روی نقشه‌های ویژگی‌های کانولوشن انجام می‌دهیم و از آن‌ها به عنوان ویژگی برای یک لایه کاملاً متصل که خروجی مورد نظر را تولید می‌کند استفاده می‌کنیم. با توجه به این ساختار اتصال ساده، می‌توانیم اهمیت نواحی تصویر را با بازتاب دادن وزن‌های لایه خروجی بر روی نقشه‌های ویژگی کانولوشنال شناسایی کنیم، تکنیکی که آن را نگاشت فعال‌سازی کلاس (class activation mapping) می‌نامیم.



همانطور که در شکل فوق نشان داده شده است، global average pooling میانگین مکانی نقشه ویژگی هر واحد را در آخرین لایه کانولوشنی به دست می دهد. از مجموع وزن‌دار این مقادیر برای تولید خروجی نهایی استفاده می‌شود. به طور مشابه، ما مجموع وزن‌دار از نقشه های ویژگی آخرین لایه کانولوشن را محاسبه می‌کنیم تا class activation mapهای خود را به دست آوریم. این تکنیک برای softmax توضیح داده می‌شود اما برای رگرسیون و سایر توابع ضرر نیز به همین شکل هست. برای یک تصویر $f_k(x,y)$ خروجی فعالسازی واحد kام در آخرین لایه کانولوشنی است. پس از آن نتیجه انجام global average pooling بر روی واحد kام برابر است با:

$$F^k = \Sigma_{x,y} f_k(x,y)$$

بنابراین برای کلاس c ورودی تابع softmax به صورت زیر است:

$$S_c = \Sigma_k w_k^c F_k$$

که w_k^c وزن متناظر با واحد kام برای کلاس c است که اهمیت F_k را برای کلاس c نشان می‌دهد. در نهایت خروجی softmax برای کلاس c به صورت زیر محاسبه می‌شود:

$$P_c = (\exp(S_c)) / (\Sigma_c \exp(S_c))$$

با جایگذاری $F^k = \Sigma_{x,y} f_k(x,y)$ در فرمول S_c داریم:

$$S_c = \Sigma_k w_k^c \Sigma_{x,y} f_k(x,y) = \Sigma_{x,y} \Sigma_k w_k^c f_k(x,y)$$

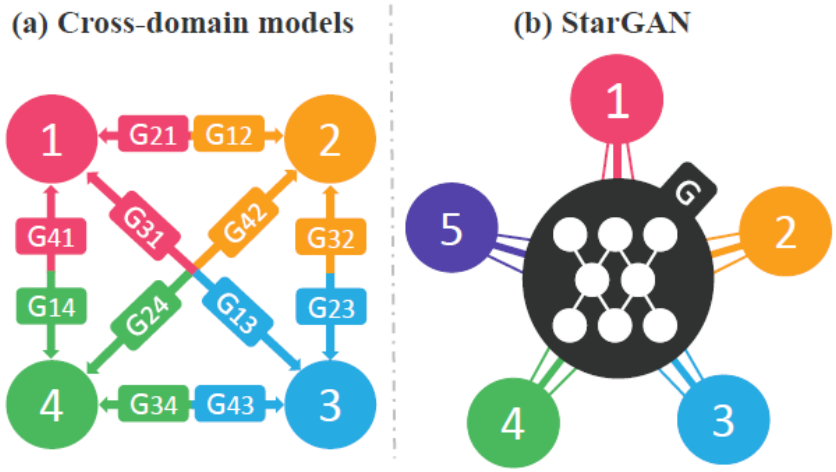
حال class activation map برای هر کلاس با توجه به مختصات فضایی آن به صورت زیر تعریف می‌شود:

$$M_c(x,y) = \sum_k w_k^c f_k(x,y).$$

در نتیجه $S_c = \sum_{x,y} M_c(x,y)$، و از این رو $M_c(x, y)$ مستقیماً اهمیت فعال سازی در گرید فضایی (x، y) را نشان می دهد که منجر به طبقه بندی یک تصویر به کلاس c می شود.

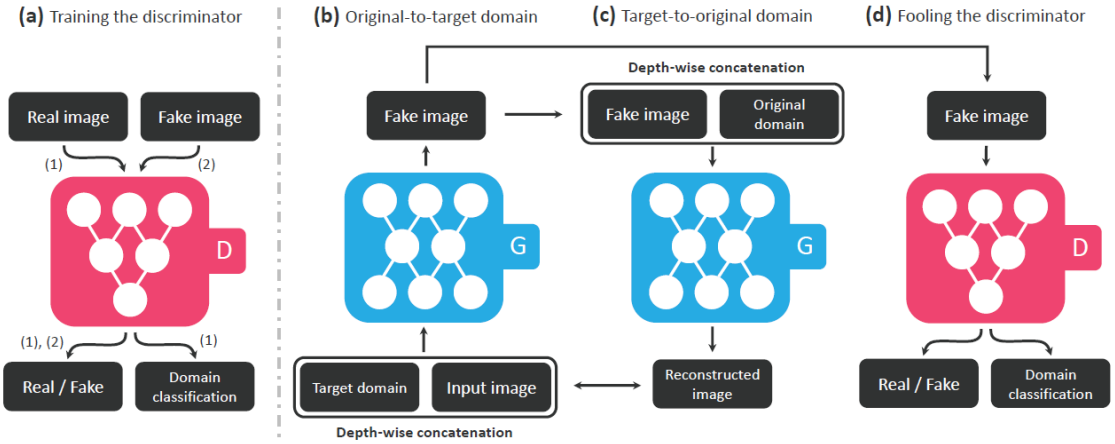
وظیفه ترجمه تصویر به تصویر این است که یک جنبه خاص از یک تصویر داده شده را به دیگری تغییر دهد، به عنوان مثال، تغییر حالت چهره یک فرد از خنده به اخم. این کار پس از معرفی شبکه‌های مولد رقابتی (GANs)، با نتایجی از تغییر رنگ مو، بازسازی عکس‌ها از نقشه‌های لبه و تغییر فصول تصاویر مناظر، پیشرفت‌های قابل‌توجهی را تجربه کرده است. با داشتن داده‌های آموزشی از دو دامنه متفاوت، این مدل ها یاد می گیرند که تصاویر را از یک دامنه به دامنه دیگر ترجمه کنند. ما اصطلاحات attribute را به عنوان یک ویژگی معنادار ذاتی در یک تصویر مانند رنگ مو، جنسیت یا سن، و مقدار ویژگی را به عنوان مقدار خاصی از یک ویژگی، به عنوان مثال، سیاه/بور/قهوه ای برای رنگ مو یا مرد/زن برای جنسیت نشان می دهیم. ما همچنین دامنه را به عنوان مجموعه ای از تصاویر با ارزش مشخصه یکسان نشان می دهیم. به عنوان مثال، تصاویر زنان می تواند یک حوزه را نشان دهد در حالی که تصاویر مردان نشان دهنده حوزه دیگری است. در ترجمه تصویر به تصویر چند دامنه تصاویر را با توجه به ویژگی‌های چندین دامنه تغییر می‌دهیم. مدل‌های موجود در چنین کارهای ترجمه تصویر چند دامنه‌ای کم‌بازده (inefficient) و ناکارآمد (ineffective) هستند. کم‌بازده بودن آن‌ها از این واقعیت ناشی می‌شود که برای یادگیری همه نگاشت‌ها در میان k دامنه، k(k-1) مولد باید آموزش داده شوند. همانطور که از شکل زیر پیداست دوازده شبکه مولد مجزا باید برای ترجمه تصاویر بین چهار دامنه مختلف آموزش داده شوند. در همین حال، آنها ناکارآمد هستند یعنی اگرچه ویژگی‌های سراسری وجود دارد که می‌توان از تصاویر همه دامنه‌ها یاد گرفت، مانند اشکال چهره، اما هر مولد نمی‌تواند به طور کامل از کل داده‌های آموزشی استفاده کند و فقط می‌تواند از دو دامنه از k دامنه یاد بگیرد. عدم استفاده کامل از داده های آموزشی احتمالاً کیفیت تصاویر تولید شده را محدود می کند. علاوه بر این، آنها قادر به آموزش مشترک دامنه‌ها از مجموعه داده های مختلف نیستند، زیرا هر مجموعه داده به صورت جزئی برچسب گذاری شده است. به عنوان راه‌حلی برای چنین مشکلاتی، StarGAN را پیشنهاد می‌کنیم، یک شبکه مولد رقابتی که قادر به یادگیری نگاشت میان دامنه‌های متعدد است. همانطور که در شکل زیر قسمت(b) نشان داده شده است، این مدل داده‌های آموزشی چندین دامنه را دریافت می‌کند و نگاشت بین تمام دامنه های موجود را تنها با استفاده از یک مولد یاد می‌گیرد. StarGAN به جای یادگیری یک ترجمه ثابت (مثلاً موهای سیاه به بلوند)، هم تصویر و هم اطلاعات دامنه را به عنوان ورودی می گیرد و یاد می گیرد که به طور انعطاف پذیر تصویر ورودی را به دامنه مربوطه ترجمه کند. در طول آموزش، به‌طور تصادفی یک برچسب دامنه هدف را تولید می‌کنیم و به مدل آموزش می‌دهیم تا به‌طور انعطاف‌پذیر یک تصویر ورودی را به دامنه هدف ترجمه کند. با انجام این کار، می‌توانیم برچسب دامنه را کنترل کرده و تصویر را در مرحله آزمون به هر دامنه دلخواه ترجمه کنیم. همچنین یک رویکرد ساده اما مؤثر را معرفی می‌شود که آموزش مشترک بین دامنه‌های مجموعه داده‌های مختلف را با

افزودن یک بردار ماسک به برچسب دامنه امکان‌پذیر می‌سازد. روش پیشنهادی تضمین می‌کند که مدل می‌تواند برچسب‌های ناشناخته را نادیده بگیرد و بر روی برچسب ارائه شده توسط یک مجموعه داده خاص تمرکز کند.



همانطور که از شکل فوق پیداست برای مدیریت چندین دامنه، CycleGAN که یک مدل بین دامنه‌ای است باید برای هر جفت دامنه تصویر ساخته شود، در حالی که StarGAN قادر به یادگیری نگاشت بین چندین دامنه با استفاده از یک مولد واحد است. این باعث می‌شود تعداد پارامترهایی که StarGAN برای ترجمه تصویر احتیاج دارد بسیار کمتر از CycleGAN باشد.

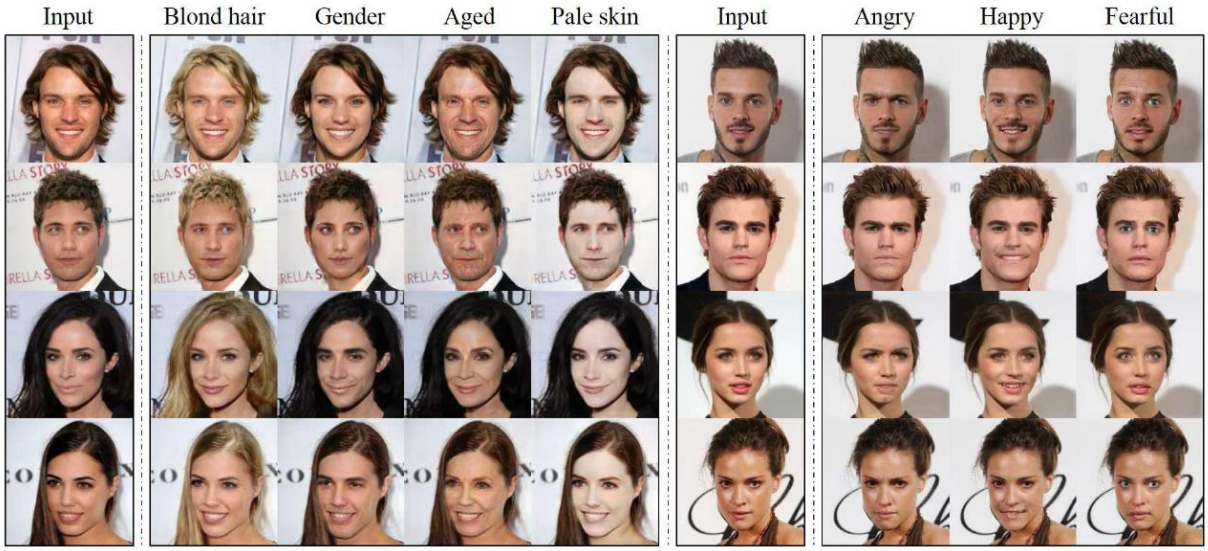
همچنین معماری StarGAN به شکل زیر است:



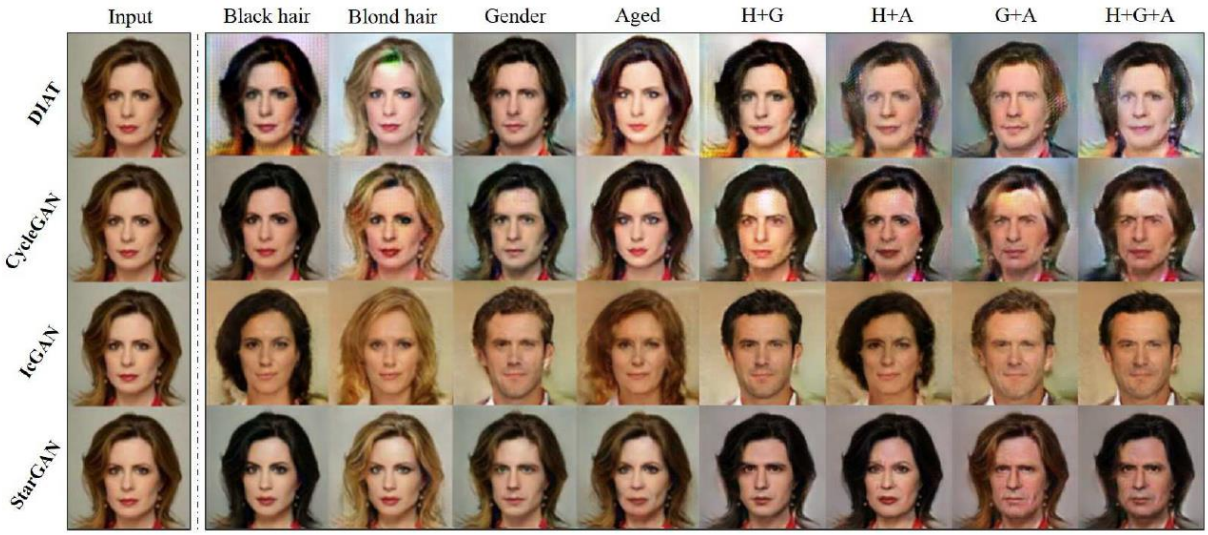
D (a) یاد می گیرد که بین تصـــاویر واقعی و جعلی تمایز قائل شـــود و تصـــاویر واقعی را به حوزه مربوطه خود طبقه‌بندی کند. G (b) هم تصـــویر و هم برچســـب دامنه هدف را به عنوان ورودی می‌گیرد و یک تصـــویر جعلی

تولید می‌کند. برچسب دامنه هدف به صورت مکانی تکرار شده و با تصویر ورودی الحاق می‌شود. (c) G سعی می کند تصویر اصلی را از تصویر جعلی با توجه به برچسب دامنه اصلی بازسازی کند. (d) G سعی می کند تصاویر غیر قابل تشخیص از تصاویر واقعی و طبقه‌بندی به عنوان دامنه هدف توسط D ایجاد کند.

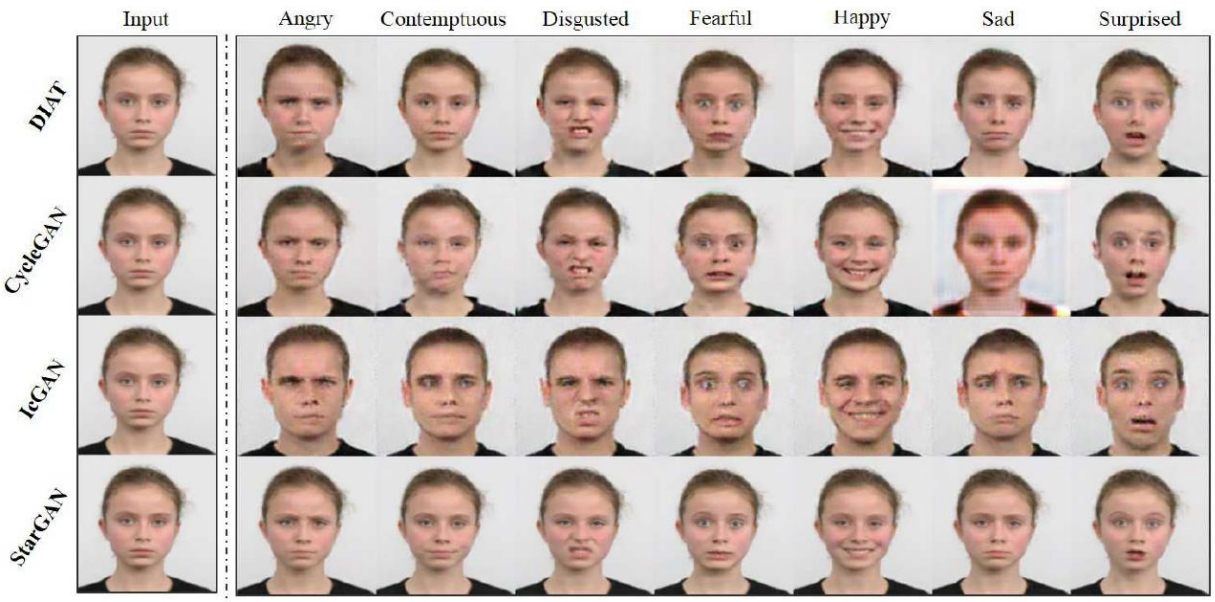
مقاله آزمایشات خود را بر روی دو مجموعه داده CelebA و RaFD انجام می‌دهد که در اولی رنگ مو، جنسیت و سن افراد تغییر می‌کند اما در دومی حالت‌های هیجانی مختلف چهره فرد ضبط شده است. شکل زیر نتایج ترجمه تصویر به تصویر چند دامنه ای در مجموعه داده CelebA از طریق انتقال دانش آموخته شده از مجموعه داده RaFD را نشان می‌دهد. ستون های اول و ششم تصاویر ورودی را نشان می دهند در حالی که ستون های باقیمانده تصاویر تولید شده توسط StarGAN هستند. توجه داشته باشید که تصاویر توسط یک شبکه مولد واحد تولید می‌شوند.



همچنین در شکل زیر می‌توان نتیاج StarGAN در تغییر رنگ مو، جنسیت و سن افراد یا ترکیبی از آن‌ها را با سایر مدل‌ها مقایسه کرد.



در شـکل زیر نیز می‌توان نتیاج StarGAN در تغییر حالت‌های هیجانی چهره افراد را با سـایر مدل‌ها مقایسـه کرد.



تفاوت اصلی starGAN و cycleGAN بالاتر با هایلات زرد مشخص شده‌است.

۳)

توضیحات مربوط به چگونگی کارکرد کد مربوط به DCGAN در فایل نوتبوک مربوطه نوشته شده است.

(۴

شبهه کد اصلی برای لاس به صورت زیر است:

$$L_{genAB} = E_{realA \sim Pdata(realA)} [1 - \log(discB(genAB(realA)]$$
$$+ \lambda * E_{realA \sim Pdata(realA)} [||genBA(genAB(realA) - realA||_1]$$

اما با توجه به اینکه این فرمول در زمان آپدیت گرادیان باعث می‌شـــود زمانی که به نقطه بهینه نزدیک شـــویم تغییرات شـدید در آپدییت گرادیان به وجود بیاید و بهینه‌ســازی را با مشــکل روبرو کند بنابراین بهتر اســت به شبهه کد زیر تغییر کند:

$$L_{genAB} = - E_{realA \sim Pdata(realA)} [\log(discB(genAB(realA)]$$
$$+ \lambda * E_{realA \sim Pdata(realA)} [||genBA(genAB(realA) - realA||_1]$$

(۵

mode collapse: معمولاً می‌خواهید GAN شما خروجی‌های متنوعی تولید کند. برای مثال، برای هر ورودی تصـــادفی به مولد چهره خود، یک چهره متفاوت می‌خواهید. با این حال، اگر یک مولد خروجی قابل قبولی تولید کند، مولد ممکن اســـت یاد بگیرد که فقط آن خروجی را تولید کند. در واقع، مولد همیشـــه در تلاش اســـت تا خروجی‌ای را بیابد که برای ممیز (discriminator) قابل قبول‌ترین به نظر می‌رســـد. اگر مولد بارها و بارها شروع به تولید همان خروجی (یا مجموعه کوچکی از خروجی ها) کند، بهترین اسـتراتژی ممیز این است که یاد بگیرد همیشه آن خروجی را رد کند. اما اگر نسل بعدی ممیز در یک مینیمم محلی گیر کند و بهترین استراتژی را پیدا نکند، برای تکرار بعدی مولد یافتن معقول‌ترین خروجی برای ممیز فعلی بسـیار آسـان اسـت. هر تکرار از مولد برای یک ممیز خاص بیش از حد بهینه (over-optimize) می‌شـود و ممیز هرگز نمی‌تواند راه خود را از تله بیاموزد. در نتیجه مولدها از طریق مجموعه کوچکی از انواع خروجی می چرخند. این شکل از خرابی GAN، mode collapse نامیده می‌شود. بنابراین mode collapse زمانی است که GAN تعداد کمی از تصاویر را با موارد تکراری (حللت‌های) زیاد تولید می‌کند. این زمانی اتفاق می‌افتد که مولد قادر به یادگیری یک نمایش ویژگی غنی نباشـد، زیرا یاد می‌گیرد خروجی‌های مشـابه را به چندین ورودی مختلف مرتبط کند. برای بررسـی mode collapse ، تصاویر تولید شده را بررسی کنید. اگر تنوع کمی در خروجی وجود داشته باشد و برخی از آنها تقریباً یکســـان باشـــند، احتمال mode collapse وجود دارد. اگر مشـــاهده کردید که این اتفاق می افتد، سعی کنید توانایی مولد برای ایجاد خروجی‌های متنوع‌تر را با موارد زیر افزایش دهید:

افزایش ابعاد داده‌های ورودی به مولد

افزایش تعداد فیلترهای مولد برای ایجاد تنوع گسترده‌تری از ویژگی‌ها

آسیب رساندن به ممیز با دادن برچسب‌های نادرست تصادفی به تصاویر واقعی

https://ch.mathworks.com/help/deeplearning/ug/monitor-gan-training-progress-and-identify-common-failure-modes.html

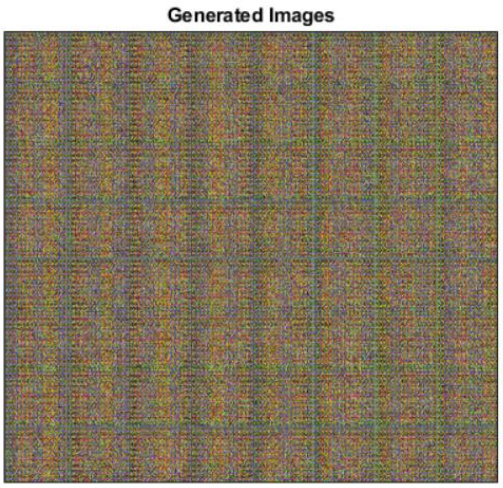
https://wandb.ai/authors/DCGAN-ndb-test/reports/Measuring-Mode-Collapse-in-GANs--VmlldzoxNzg5MDk

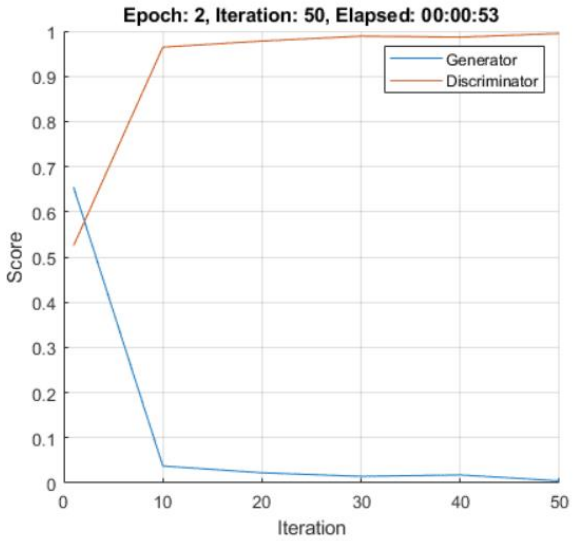
convergence failure:

convergence failure زمانی اتفاق می‌افتد که مولد و ممیز در طول فرایند آموزش به تعادل نرسند.

ممیز غلبه می‌کند:

این سناریو زمانی اتفاق می افتد که score مولد به صفر یا نزدیک به صفر برسد و score برای ممیز به یک یا نزدیک به یک برسد. نمودار زیر نمونه‌ای از غلبه ممیز بر مولد را نشان می‌دهد. توجه داشته باشید که score مولد به صفر نزدیک می شود و بازیابی نمی شود. در این حالت، ممیز بیشتر تصاویر را به درستی طبقه‌بندی می کند. به نوبه خود، مولد نمی تواند هیچ تصویری تولید کند که ممیز را فریب دهد و در نتیجه در یادگیری شکست می‌خورد.





اگر برای تکرارهای زیادی score از این مقادیر بازیابی نشد، بهتر است فرایند آموزش را متوقف کنید. اگر این اتفاق افتاد، سعی کنید عملکرد مولد و ممیز را با روش زیر متعادل کنید:

آسیب رساندن به ممیز با دادن برچسب های نادرست تصادفی به تصاویر واقعی

افزودن لایه‌های dropout ، به ممیز

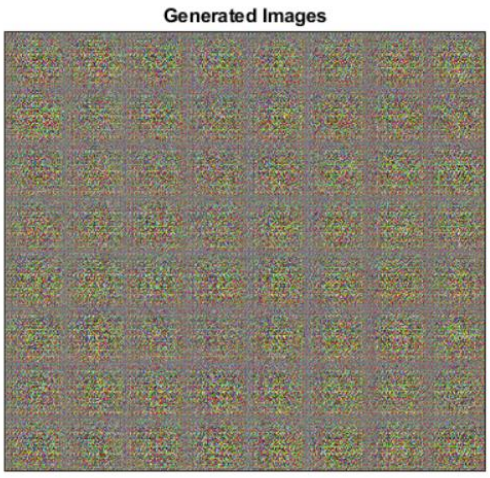
بهبود توانایی مولد برای ایجاد ویژگی‌های بیشتر با افزایش تعداد فیلترها در لایه‌های کانوولوشنی آن

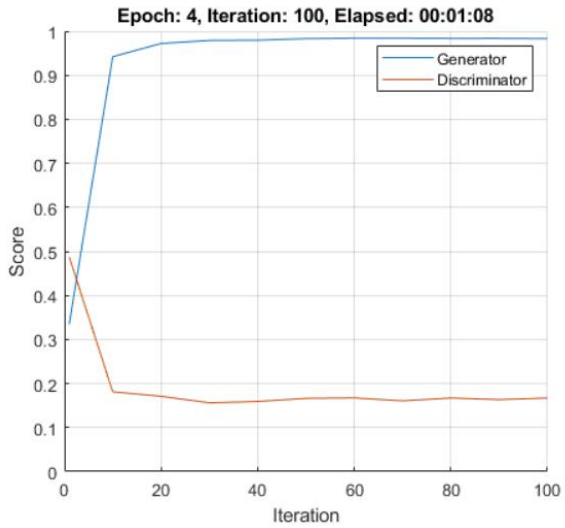
آسیب رساندن به ممیز با کاهش تعداد فیلترهای آن

مولد غلبه می‌کند:

این سناریو زمانی اتفاق می افتد که score مولد به یک یا نزدیک به یک برسد.

نمودار زیر نمونه‌ای از غلبه مولد بر ممیز را نشان می‌دهد. توجه داشته باشید که score مولد برای چندین تکرار به یک می‌رسد. در این مورد، مولد تقریباً همیشه یاد می‌گیرد که چگونه ممیز را فریب دهد. وقتی این خیلی زود در فرآیند آموزش اتفاق می‌افتد، احتمالاً مولد یک بازنمایی ویژگی بسیار ساده را یاد می‌گیرد که ممیز را به راحتی فریب می‌دهد. این بدان معناست که تصاویر تولید شده علیرغم داشتن امتیاز بالا می‌توانند بسیار ضعیف باشند. توجه داشته باشید که در این مثال، امتیاز ممیز خیلی به صفر نزدیک نمی‌شود، زیرا همچنان می‌تواند برخی از تصاویر واقعی را به درستی طبقه‌بندی کند.





اگر برای تکرارهای زیاد score از این مقادیر بازیابی نشد، بهتر است فرایند آموزش را متوقف کنید. اگر این اتفاق افتاد، سعی کنید عملکرد مولد و ممیز را با روش زیر متعادل کنید:

بهبود توانایی ممیز برای یادگیری ویژگی های بیشتر با افزایش تعداد فیلترها

آسیب رساندن به مولد با افزودن لایه‌های dropout

آسیب رساندن به مولد با کاهش تعداد فیلترهای آن

حالت زیر نیز یکی از موارد غلبه مولد است:

همانطور که مولد با آموزش بهبود می‌یابد، عملکرد ممیز بدتر می شـــود زیرا ممیز نمی‌تواند به راحتی تفاوت بین واقعی و جعلی را تشـــخیص دهد. اگر مولد به طور کامل موفق شـــود، پس ممیز دارای دقت ۵۰٪ اســـت. در واقع انگار ممیز یک ســـکه می‌اندازد و با شـــیر یا خط پیش‌بینی خود را انجام می‌دهد. این پیشـــرفت یک مشـــکل برای همگرایی GAN ایجاد می کند: بازخورد ممیز با گذشـــت زمان کمتر معنادار می‌شـــود. اگر GAN از نقطه‌ای که ممیز بازخورد کاملاً تصـــادفی می‌دهد به آموزش ادامه دهد، مولد شـــروع به آموزش بر روی بازخورد ناخواســـته می‌کند و کیفیت خودش ممکن است سقوط کند.

https://ch.mathworks.com/help/deeplearning/ug/monitor-gan-training-progress-and-identify-common-failure-modes.html

https://developers.google.com/machine-learning/gan/training