



به نام خدا
دانشگاه تهران
دانشکده مهندسی
برق و کامپیوتر



درس شبکه‌های عصبی و یادگیری عمیق تمرین ششم

سپهر آزدادار – پوریا تاج محربی	نام و نام خانوادگی
810199357 - 810199395	شماره دانشجویی
1402.09.29	تاریخ ارسال گزارش

۱-۱. مقدمه

۱-۲. بیاده سازی VAE

۱-۳. ارزیابی مدل VAE

۱-۴. بیاده سازی CONTROL VAE

پاسخ ۱. Control VAE

1-1. مقدمه

مقاله "ControlVAE: مدل اتوانکدر واریاسیونی قابل کنترل" یک چهارچوب جدید، ControlVAE را معرفی می‌کند که یک کنترلر الهام گرفته از نظریه کنترل خودکار را با مدل اتوانکدر واریاسیونی استاندارد (VAE) تلفیق می‌کند. این چهارچوب با هدف رفع محدودیت‌های VAE های سنتی، مانند مشکل ناپدید شدن KL در مدل‌سازی زبان و کیفیت پایین بازسازی در تولید تصویر و یادگیری نمایش جداسازی شده تعریف شده است.

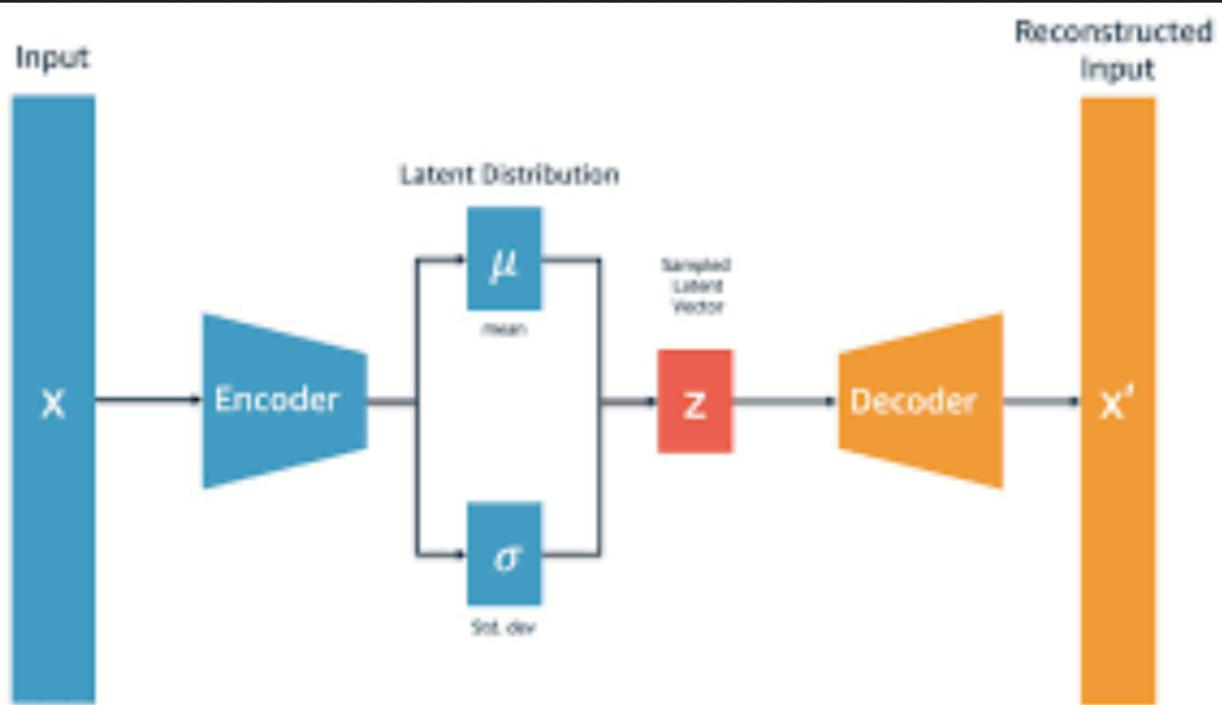
VAE از یک کنترل غیرخطی متناسب-انتگرال (PI) برای تنظیم خودکار هایپرپارامتر (وزن) در هدف ControlVAE استفاده می‌کند. این تنظیم براساس بازخورد تفاوت KL خروجی در طول آموزش است. مهم‌ترین کمک توانایی آن در کنترل دقیق تعادل بین دقت بازسازی داده‌ها و محدودیت‌های خاصی مانند تنوع خروجی یا نمایش عامل پنهان جداشده است. این کنترل عملکرد کاربردهای مختلف را بهبود می‌بخشد، منجر به بهتر شدن کیفیت بازسازی و تنوع بیشتر در داده‌های تولیدی، از جمله تصاویر و مدل‌های زبانی می‌شود.

با تنظیم پویای هایپرپارامتر در طول آموزش مدل، ControlVAE به طور مؤثری تفاوت KL را در مقدار مطلوب ثابت نگه می‌دارد، و خروجی‌های تولیدی با کیفیت بالا و تنوع دار را به دست می‌آورد. مقاله اثربخشی ControlVAE را در کاربردهای مختلف نشان می‌دهد، و برتری آن را در دستیابی به تعادل بین کیفیت بازسازی و محدودیت‌های خاص مورد نظر کاربرد نشان می‌دهد.

ایده کلی مقاله "ControlVAE: مدل اتوانکدر واریاسیونی قابل کنترل" بهبود عملکرد مدل‌های اتوانکدر واریاسیونی (VAEs) با استفاده از یک مکانیزم کنترلی است. VAE های سنتی اغلب با مشکل تعادل بین بازسازی دقیق داده‌ها و عوامل دیگری مانند تنوع یا جداسازی در داده‌های تولید شده روبرو هستند. ControlVAE با ادغام رویکرد مبتنی بر نظریه کنترل خودکار، به خصوص یک کنترل متناسب-انتگرال (PI)، این موضوع را برطرف می‌کند. این کنترلر به طور پویا یک هایپرپارامتر کلیدی در VAE را در طول آموزش، براساس بازخورد تفاوت KL خروجی، تنظیم می‌کند. این تنظیم امکان مدیریت بهتر تعادل را فراهم می‌کند و منجر به بهبود کیفیت و تنوع در خروجی‌های تولید شده، مانند تصاویر و مدل‌های زبانی می‌شود.

در زمینه مدل ControlVAE، "هایپرپارامتر" اشاره شده به وزن یا ضریبی اشاره دارد که دو جزء تابع زبان VAE را متعادل می‌کند: زیان بازسازی و تفاوت KL. در یک VAE استاندارد، این هایپرپارامتر ثابت است، که اغلب منجر به تعادل‌های ذکر شده می‌شود. ControlVAE با تنظیم پویای این هایپرپارامتر در طول آموزش، به دنبال بهینه‌سازی تعادل بین بازسازی دقیق داده‌ها و اهداف دیگر مانند نمایش جداسازی شده، بر اساس عملکرد واقعی مدل است. این رویکرد تطبیقی امکان آموزش مؤثرتر و نتایج بهبود یافته در کاربردهای مختلف را فراهم می‌کند.

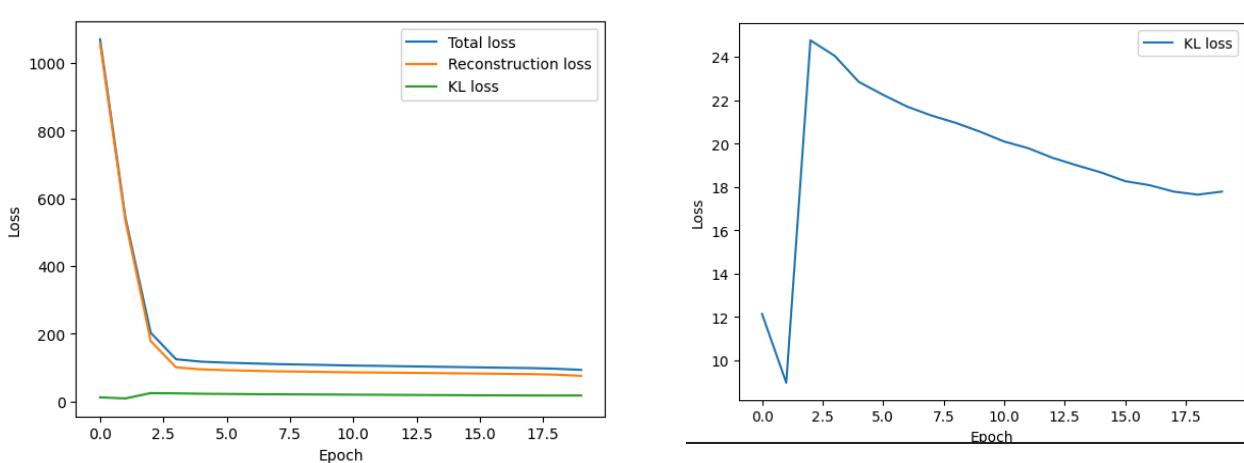
۱-۲. پیاده سازی VAE



همان طور که در شکل بالا مشاهده میکنید، ساختار VAE ها به این صورت میباشد که فضای LATENT آن ها شامل دو وکتور میانگین و انحراف از معیار میباشد. و باتوجه به آن Z ساخته میشود. در واقع طول دو وکتور μ و σ باهم برابر است و این طول LATENT SPACE ما میباشد. و اینجوری هست که μ و σ توزیع نرمال LATENT SPACE، به ازای هر بعد را مشخص میکند.

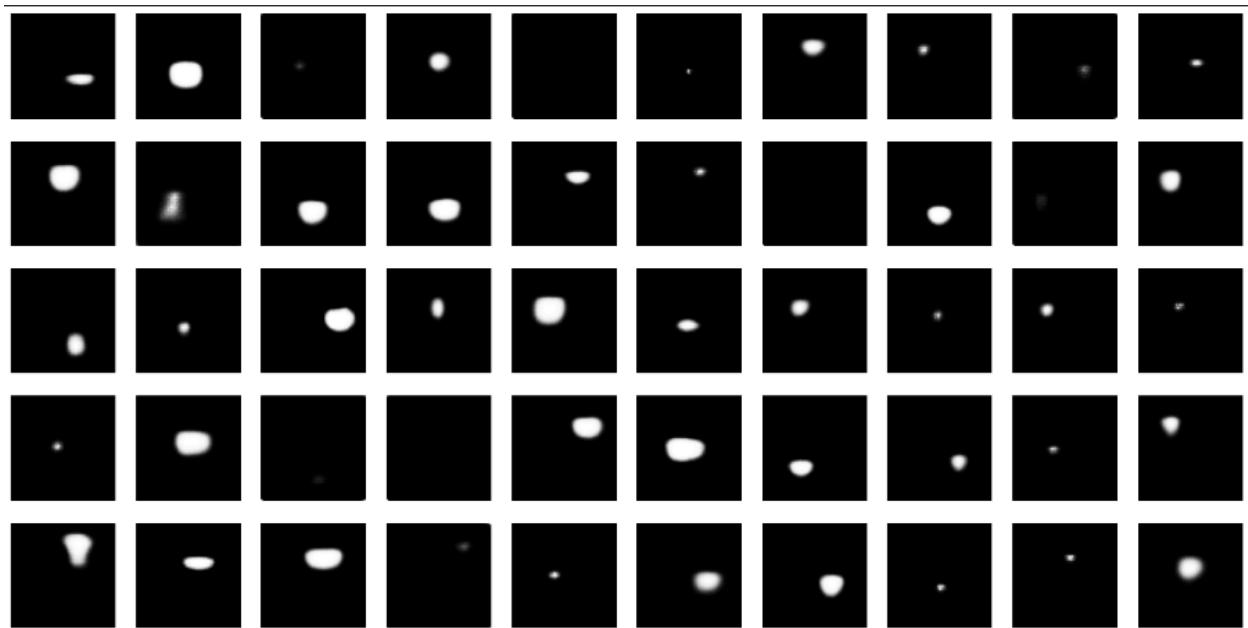
پس برای پیاده سازی دو وکتور به طول 10 برای μ و σ در نظر گرفته ایم، همان طور که در جدول آمده است. یه لایه ۲۵۶ داریم و یک لایه ۲۰، یا همان دو تا ۱۰ که μ و σ را برای LATENT DIS مشخص میکند.

۱-۳. ارزیابی مدل VAE

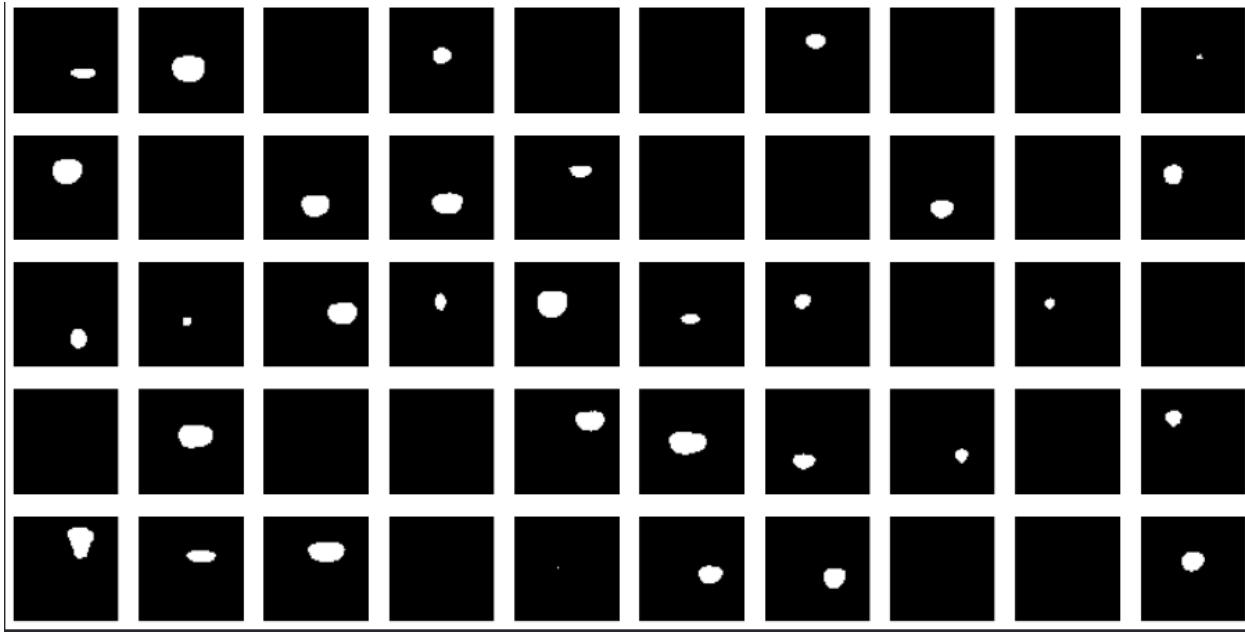


مدل VAE دو نوع loss داریم، یکی، reconstruction_loss و دیگری KL_loss . Reconstruction_loss تطابق ورودی و خروجی را بررسی میکند. سعی دارد که مدل را تشویق کند که از روی latent space به ورودی برسد. اما loss دیگر سعی دارد که توزیع آماری latent_space با توزیع نرمال استاندارد باشد. این کمک میکند که مدل فضا ورودی و distribution داده های ورودی را به درستی بشناسد. و سعی دارد که مدل این توزیع را بهتر یاد بگیرد. و همچنین Generalizaition بهتری داشته باشد، و همچنین بتواند در آینده با سمپل کردن از Z به داده های بهتری برسیم.

ممکن است رسیدن به هر دو اینا به طور همزمان ممکن نباشد. و به نوعی یک trade-off بین این دو لاس وجود دارد. و ما میتوانیم با وزن دادن به آن ها، ارزش نسبی آن ها را اصلاح کنیم تا مدل را به جایی که میخواهیم ببریم. همان طور که میبینید، در اینجا مدل بیشتر سعی کرده که تطابق ورودی و خروجی را در ابتدا بهتر کند، زیرا که در ابتدا loss خیلی زیادی از بابت آن دارد میپذارد و احتمالا مشتق از سمت بسیار بالاس. از طرفی اگر مدل را در نرم استاندارد نگه داریم، لاس kl آن صفر میشود. پس مدل سعی میکند در ایپاک های اولیه به reconstruction_loss کند. و در نتیجه همان طور که میبینید، این لاس پایین میاید و kl به شدت افزایش میباید. مدل همواره سعی میکند، برایند این دو را کاهش دهد. در لحظه های دیگر هم با همین استدلال میتوان، نتیجه گیری کرد.



:threshold بعد از اعمال



:FID

معیار Fréchet Inception Distance شباهت بین دو مجموعه تصاویر، معمولاً تصاویر واقعی و تصاویر تولید شده توسط مدل‌هایی مانند GAN‌ها را اندازه‌گیری می‌کند. FID با استفاده از بردارهای ویژگی استخراج شده، توزیع تصاویر تولیدی را با توزیع تصاویر واقعی مقایسه می‌کند. فاصله با استفاده از میانگین و کوواریانس بردارهای ویژگی از هر دو مجموعه محاسبه می‌شود. FID پایین‌تر نشان می‌دهد که توزیع‌ها شبیه‌تر هستند، که نشان‌دهنده کیفیت بهتر تصاویر تولید شده است. محاسبه دقیق شامل محاسبه FID (همچنین به عنوان Wasserstein-2 distance شناخته می‌شود) بین دو گویی چند متغیره، تعریف شده توسط میانگین‌ها و کوواریانس‌های بردارهای ویژگی است.

$$FID = \|\mu_x - \mu_y\|_2^2 + Tr(\Sigma_x + \Sigma_y - 2(\Sigma_x \Sigma_y)^{1/2})$$

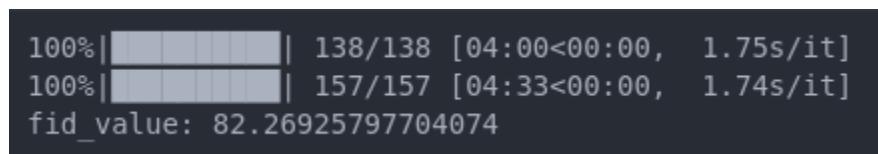
که در آن:

- μ_x, μ_y میانگین‌های ویژگی به ترتیب برای تصاویر واقعی و تولید شده هستند.
- Σ_x, Σ_y ماتریس‌های کوواریانس برای تصاویر واقعی و تولید شده به ترتیب هستند.
- $\|\mu_x - \mu_y\|_2^2$ نرم L2 مربع شده بین میانگین‌ها است.
- $\text{Tr}(\Sigma_x \Sigma_y)$ نشان‌دهنده اثر یک ماتریس است (مجموع تمام عناصر قطری).
- عبارت نهایی، $\frac{1}{2}(\Sigma_x \Sigma_y)$ ، شامل محاسبه ریشه دوم حاصلضرب ماتریس‌های کوواریانس است.

$$\text{FID} = |\mu_1 - \mu_2| + \text{Tr}(\sigma_1 + \sigma_2 - 2\sqrt{\sigma_1 * \sigma_2})$$

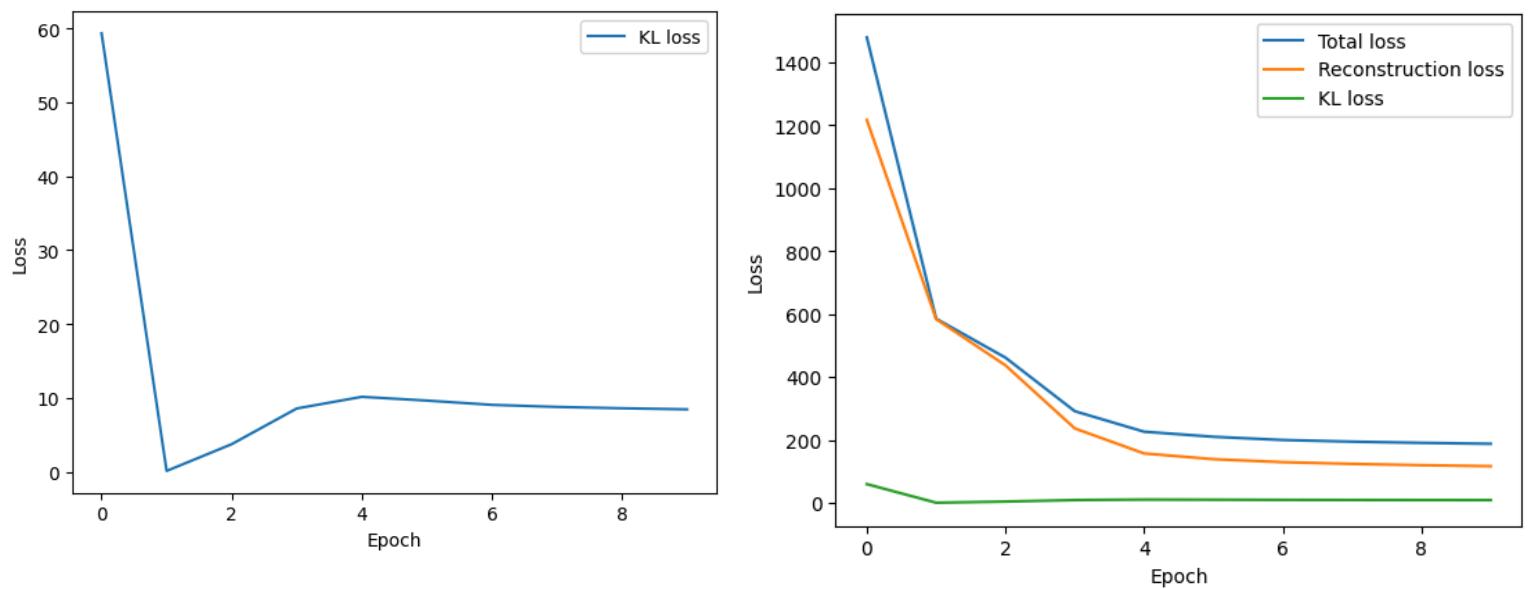
همان طور که مشاهده میکنید، در صورتی که دو توزیع با یکدیگر برابر باشند، آنگاه FID برابر با ۰ میشود. پس هر چقدر مقدار این متربک کمتر باشد، بهتر است و نشان دهنده شباهت بیشتر دو توزیع میباشد.

:VAE برای FID محاسبه شده



۱-۴. پیاده سازی CONTROL VAE

الگوریتم گفته شده در مقاله را پیاده سازی کردیم، باید `set beta max` و `beta min` را نیز میکردیم که به ترتیب ۵-۲۵ برای آنها در نظر گرفتیم. این دو مقدار حد بالا و پایین برای `beta` را تعیین میکنند. و `KL divergence` ضریب `beta` را تعیین میکنند. پس حد بالا و پایین شدت اهمیت دادن یا ندادن را با این دو مقدار تعیین میکنیم. بقیه `constant` ها الگوریتم با توجه به مقاله و خواسته صورت سوال تعیین شد. همچنین تابع `loss` را اندکی تغییر دادیم و ضریب `beta` را در پشت `kl` قرار دادیم. ابتدا در هر بتج `beta` را حساب میکنیم و با توجه به آن `loss` مدل را بدست میاریم و `back propagate` میکنیم. همچنین به جای سمپل کردن آن طوری که در الگوریتم ذکر شده بود، میانگین `kl_loss` در `btach` فعلی در نظر گرفته شده است. این روی `robust` تر میباشد و `varaince` کمتری نسب به اینکه به `kl` را از بین نمونه های یک بتج سمپل کنیم دارد.

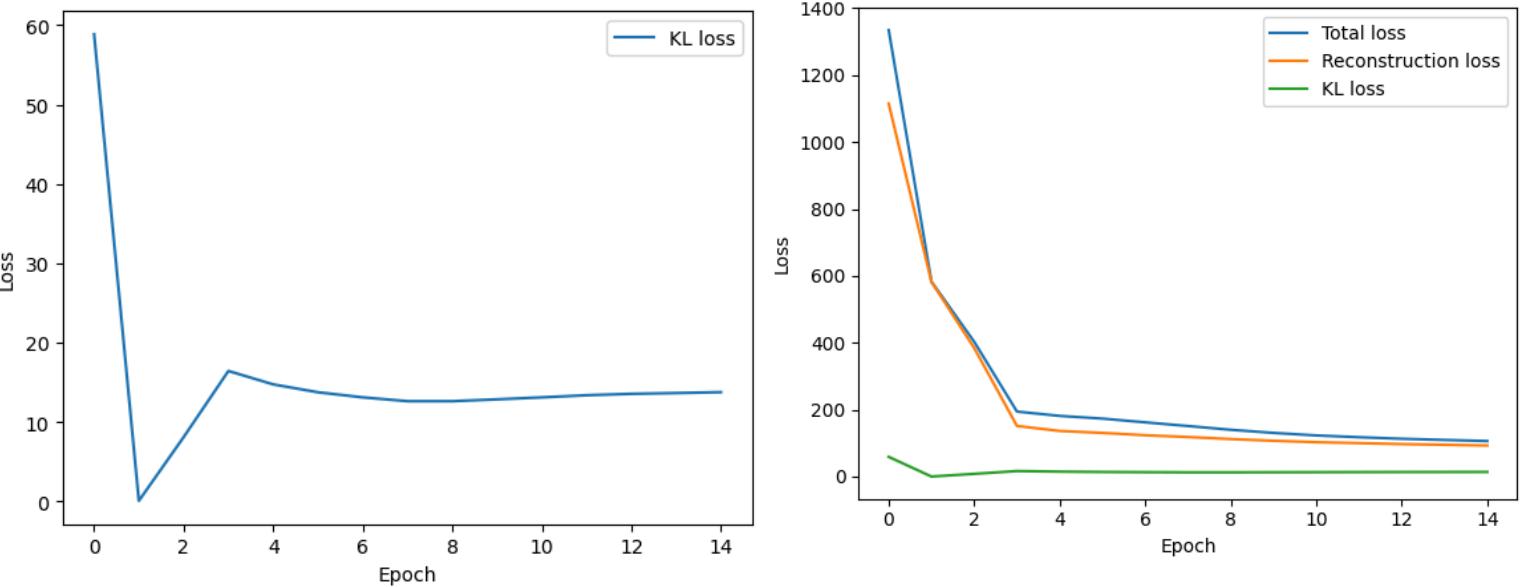


Desired KL = 8

همان طور که مشاهده میکنید، با استفاده از الگوریتم داده شده در Control VAE میتوانیم، KL را به مقداری که مد نظرمان هست ببریم. در اینجا Desired KL = 8 بود و نمودار kl_loss نیز به این مقدار میل میکند.

```
100%|██████████| 138/138 [01:19<00:00, 1.74it/s]
100%|██████████| 157/157 [01:30<00:00, 1.74it/s]
fid_value: 80.64809384890567
```

از آنجایی که میخواستیم kl_loss مان کم باشد، در نتیجه مدل reconstruction_loss بالاتری دارد، اما آن بهتر است، زیرا که توزیع به توزیع داده های ورودی نزدیک تر است.

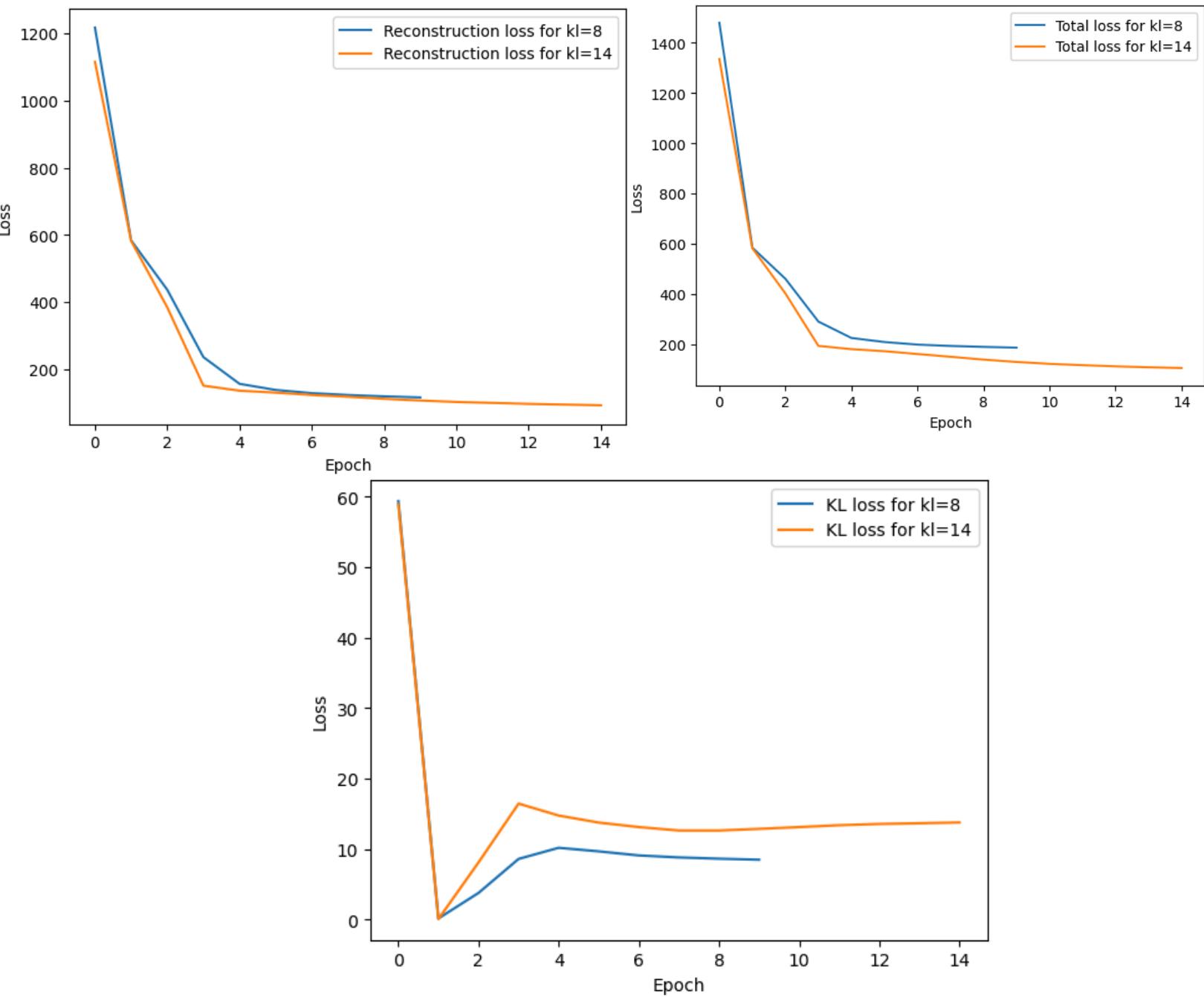


Desired KL = 14

در اینجا همان طور که میخواستیم, kl_loss به ۱۴ میل میکند. از طرفی چون اجازه دادیم kl_loss بالاتری داشته باشیم، در ازای آن کمتری خواهیم داشت.

```
100%|██████████| 138/138 [01:19<00:00, 1.73it/s]
100%|██████████| 157/157 [01:31<00:00, 1.72it/s]
fid_value: 136.78780765001687
```

این موضوع باعث میشود که توزیع داده ها در latent space نسبت به حالت قبلی بیشتر از توزیع داده های ورودی فاصله بگیرد. زیرا ما در latent space صورت نرمال استاندارد سمت پل میکنیم، اگر توزیع در این فضا غیر از این باشد، این مسئله باعث میشود که عکس های خروجی با عکس های ورودی بیشتر تفاوت داشته باشند و در راستای همین fid افزایش میابد. مدل در این حالت کمتری خواهد داشت.



همان طور که بالاتر گفته شد، kl_loss با لاتر در اینجا 14 (نمودار نارنجی) reconstruction loss کمتری خواهد داشت و در نتیجه از آنجا که از لحاظ عددی reconstruction loss بزرگ تر است، پس total_loss نیز کمتر خواهد بود. و این در نمودار های خود را نشان داده است.

پاسخ ها GAN.۲

2-آموزش مدل GAN بر روی دیتاست MNIST

با استفاده از پارامتر های زیر شبکه را آموزش داده ایم .

Optimizier	Learning Rate Disc	Learning Rate Gen	epochs
Adam	lr = 0.0001	lr = 0.0001	۱۵۰

همچنین در هر epoch 20 به تعداد ۱۰۰ عکس تولید کرده ایم تا progress GAN را به صورت چشمی چک کنیم که به وضوح آموزش دادن شبکه در ان دیده می شود .

حال در بررسی شبکه های دیگر میبینیم که این شبکه بسیار بدتر نسبت به آن ها عمل کرده و بسیار کند تر نیز آموزش میبیند. در پایین نمودار های Loss مرتبط دیده می شود . لازم به ذکر است که همگرایی بسیار کند اتفاق افتاده ولی در حال انجام است

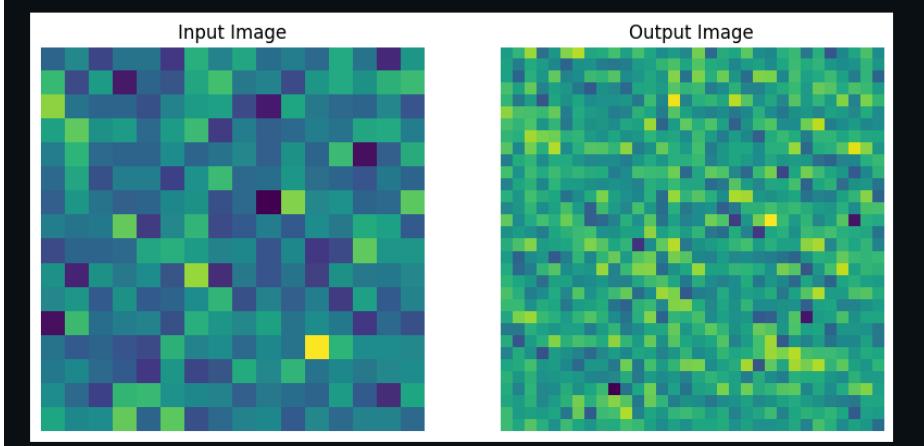
۱-۲ پیاده سازی

پرسش ۱

در عکس زیر کد کامل کننده جدول هایی که به عنوان معماری داده شده اند را قرار داده ایم .
در بخش GAN باید عکس، های ۸ در ۸ تولید کنیم حون و رو دی ۶۴، قرار داده شده است .

با توجه به خروجی لایه Convolutional بخش لایه Linear را در این قسمت متناسب قرار دادیم ($7 * 7 * 64$) شاید این به این دلیل باشد که تولید کردن sample هایی مثل ۸ و ۳ که Discriminator نتواند بین آن ها را تشخیص بدهد ممکن است باعث افتادن در نقاط کمینه محلی شود .

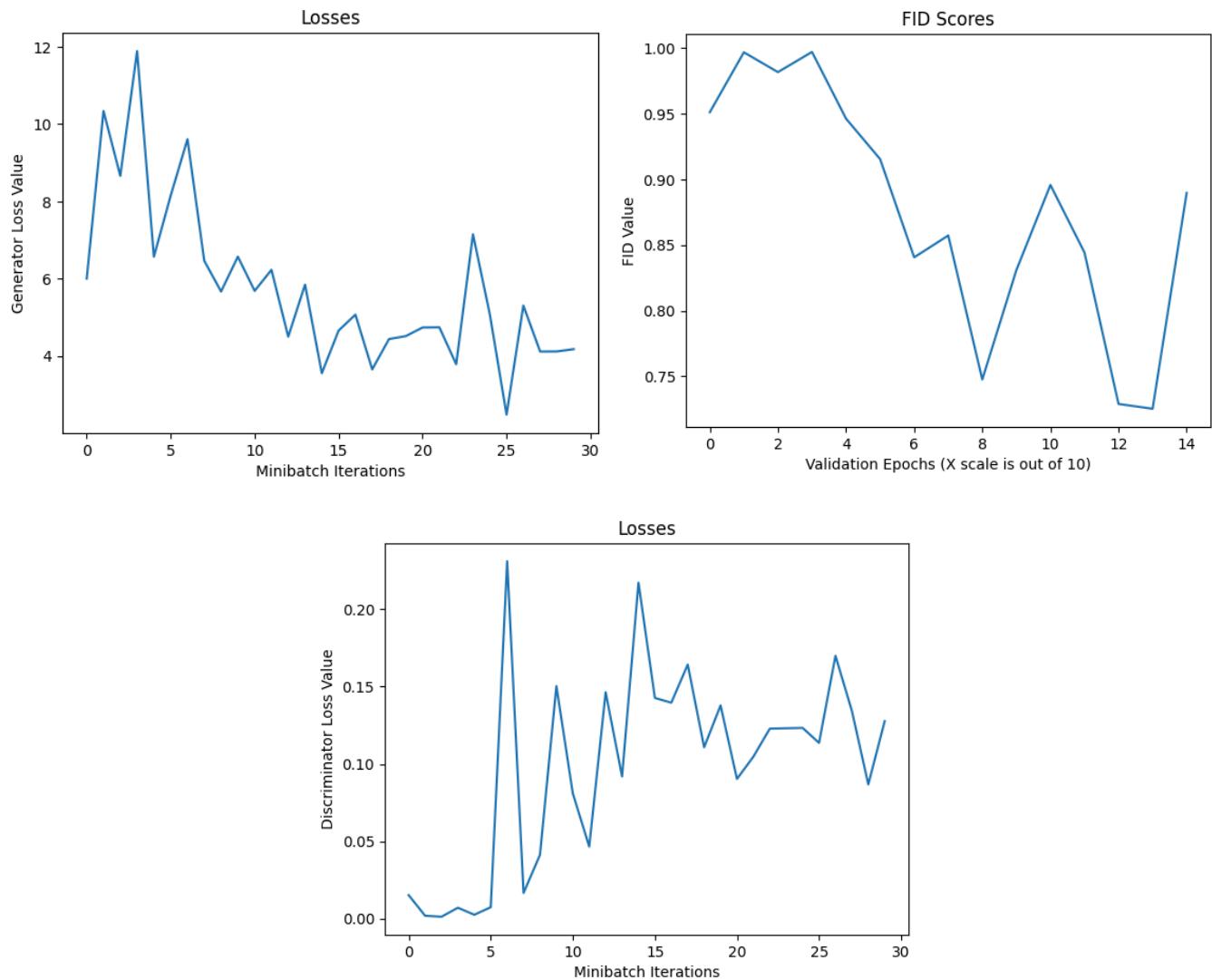
```
Input image size: torch.Size([1, 1, 16, 16])
After convolution: torch.Size([1, 4, 16, 16])
After pixel shuffle: torch.Size([1, 1, 32, 32])
```



پرسش ۲: خروجی PixelShuffle پرای یک عکس رندم را مشاهده می کنید.

توضیح: این لایه برای تولید عکس مبنی بر Superresolution استفاده می شود و کاربرد آن این است که یک ورودی با سایز کوچک تر را گرفته و بعد آن را به یک تصویر High Resolution تبدیل کند. الگوریتم این لایه در ابتدا برای Upsampling با هزینه کمتر استفاده می

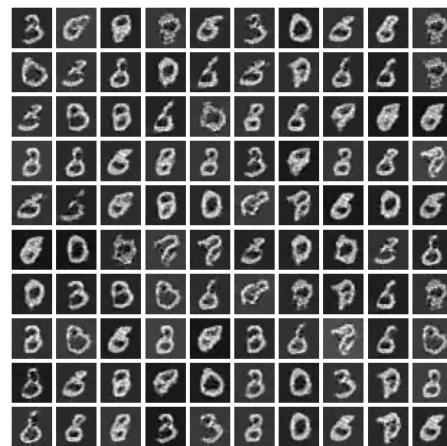
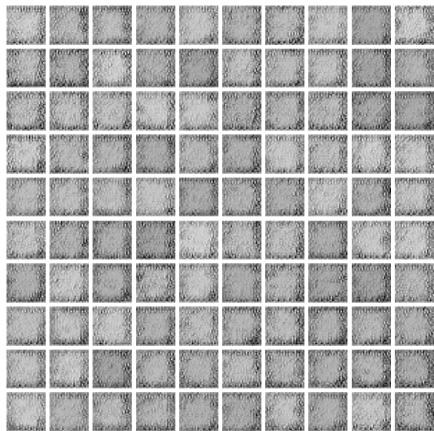
شد تا روی دستگاههای با منابع کمتر محاسباتی استفاده شود. در مقاله ای ابتدایی این لایه با استفاده از ارزیابی ریاضیاتی ادعا می‌شود که PixleShuffle با از دست دادن Minimal Information می‌تواند همان کار لایه های قدیمی Upsampler در شبکه های Convolutional را انجام بدهد و در واقع با آن هم ارز است. ولی در ابتدا این لایه اصلاً در کاربرد های Deep Learning به کار نمی‌رفت و به عنوان یک راه برای Upsample کردن و انجام Convolution های سریع و کم کردن هزینه این امر استفاده می‌شد که بعد در DeepLearning برای اتوماتیک کردن این موضوع استفاده شد.



ارزیابی

در مورد Score FID همبستگی مناسبی بین خروجی های مدل و بهتر شدن آن در طول آموزش میبینیم (عکس های پایین) در کل مشخص است که این شبکه آنقدر خوب عمل نمی کند. (FID) در واقع فاصله بین توزیع داده های آموزش و داده های تولید شده توسط GAN را نشان می دهد . در نهایت مقدار FID هر چه قدر کوچک تر باشد بهتر است که آموزش شبکه همراستا با کاهش این متریک اتفاق افتاده است ولی Trend آن ایده آل نیست . همچنین همبستگی خوبی بین کاهش فاصله Loss (در واقع Convergance GAN) و شبکه ما و کاهش FID دیده می شود .

خروجی مدل در طول آموزش (ربع اول - دوم - سوم و چهارم آموزش epoch ۱۵۰ ای)



۲-۲. مدل Wasserstein GAN

الف) این مدل را با استفاده از پارامتر های زیر آموزش داده ایم

Optimizier	Learning Rate Disc	Learning Rate Gen	epochs
RMSprop	lr=0.001	lr=0.001	۱۵۰

نکته اول این معماری و پیاده سازی در Stable Loss تر کردن یادگیری است به این شکل که در Loss های قبلی به شکلی به دلیل حساب کردن Entropy حاوی توابع لگاریتمی است که به راحتی می تواند باعث به وجود آمدن Vanishing Gradients یا Exploding Gradients شود . در این نوع GAN از یک Loss function مناسب تر بدون توابع لگاریتمی استفاده شده است که همچنین عدد نهایی آن Interpetable است .

یکی از مشکلاتی که در GAN قابل دیدیم ، تنوع کمتر خروجی های این GAN بود در این نوع GAN از این مفهوم که

نامیده می شود کمتر دیده می شود) می تواند به این دلیل باشد که یک سری ورودی های تکراری به راحتی می توانند Discriminator را گول بزند و GAN ما روی تولید این نوع ورودی ها بایاس می شود) و در این زمینه بهتر عمل می کند.

یکی از دلایل امر بالا می تواند جلوگیری طبیعی این نوع GAN از به وجود آمدن Exploding Gradient یا Vanishing Gradient هایی شود که باعث گیر کردن GAN ما روی یک محدوده خاصی از خروجی می شوند.

در این نوع GAN از مفهومی به نام Weight Clipping نیز استفاده شده است که از رشد کردن وزن های مدل جلوگیری کرده تا مفهومی به نام محدودیت Lipschitz را enforce کند.

همچنین Optimizer استفاده شده برای Generator در این GAN پیشنهادی متفاوت نسبت به Adam است و با Learning Rate بسیار کمتر استفاده می شود.

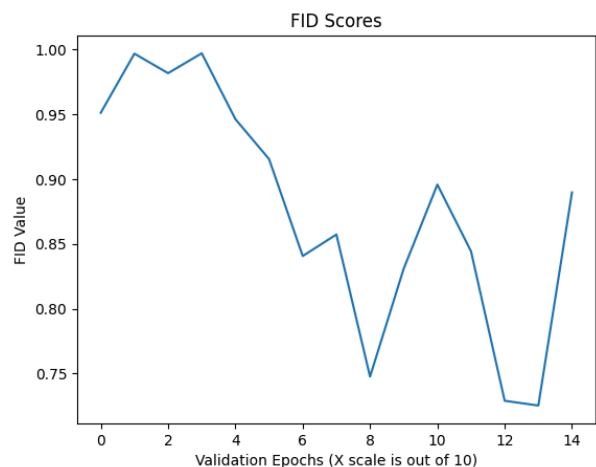
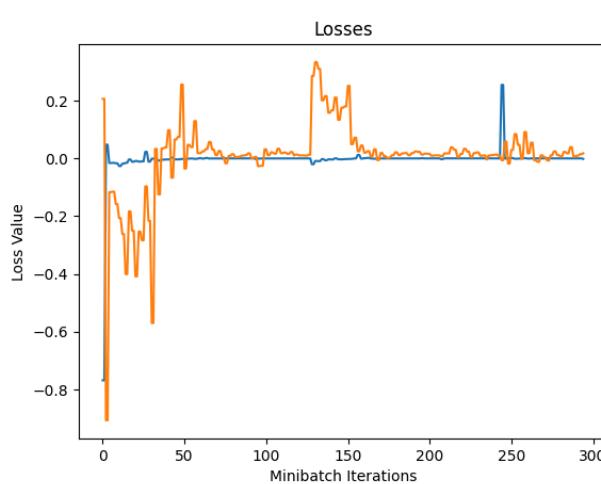
ب) تغییرات کد شامل تغییر دادن Loss function

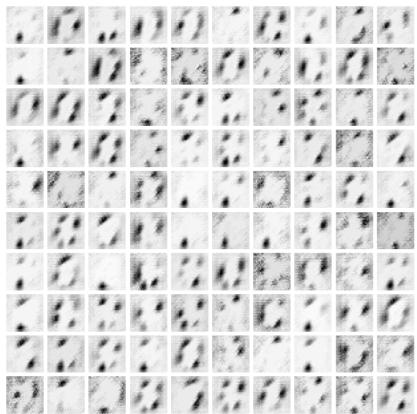
```
24
25 def wgan_critic_loss(critic_real, critic_fake):
26     return torch.mean(critic_fake) - torch.mean(critic_real)
27
28 def wgan_generator_loss(critic_fake):
29     return -torch.mean(critic_fake)
30
```

در این نوع GAN پیشنهاد می شود که Loss function ها طوری انتخاب شوند که Minimize FID Score را کنند از آنجایی که مستقیماً استفاده کردن از FID بسیار پر هزینه است . می توانیم با استفاده از فرمول بالا عملیاتی سریع تر داشته باشیم که دقیقاً همان کار ردن این Distance را داشته باشد.

```
clip_weights(  
    wgan_discriminator, -0.01, 0.01  
) # Clip weights after updating critic
```

```
31
32 wgan_d_optimizer = optimizer = optim.RMSprop(
33     wgan_discriminator.parameters(), lr=0.001
34 )
35 wgan_g_optimizer = optimizer = optim.RMSprop(wgan_generator.parameters(), lr=0.001)
36
```





6	8	6	2	3	1	2	9	0	8
7	9	9	0	7	0	5	9	5	0
0	4	5	5	0	5	0	1	0	3
3	3	9	9	9	7	9	8	2	9
1	9	6	0	0	7	0	5	5	8
4	9	9	9	5	3	0	3	6	0
5	9	0	0	0	9	0	1	3	6
6	9	3	2	7	3	5	6	2	4
4	1	7	8	4	6	3	9	7	0
7	8	5	9	1	6	8	5	3	6

5	4	0	6	5	0	7	5	7	9	7
3	6	2	3	9	9	7	8	6	9	9
0	0	6	0	3	7	8	6	9	5	5
6	5	6	7	5	9	5	2	0	9	9
0	9	0	0	7	7	5	9	1	3	3
0	3	3	1	6	0	6	6	6	9	2
0	6	1	0	8	2	6	9	9	7	7
0	9	3	9	9	8	0	6	9	9	9
6	5	7	7	5	2	9	9	9	3	3
6	1	2	8	1	9	3	5	3	9	5

یک سری از عکس های مریبوط به آموزش را در بالا مشاهده می کنیم

توضیح عملکرد و ارزیابی عملکرد این GAN به وضوح بسیار بهتر از Convergence GAN قبل است . سرعت آن بسیار بیشتر و همچنین عکس دوم که نشان دهنده Epoch 6 است نشان دهنده Progress بسیار قوی این GAN در زمان کمتر است . همچنین شبکه کمتر شدن FID بسیار بیشتر از GAN قبلی است . در خروجی های این GAN تنوع بیشتری دیده می شود که با promise های قسمت الف مبنی بر تنوع و بیشتر بودن کلاس های ورودی منطبق است . نزدیک شدن Loss های دو شبکه و Smooth بودن این Transition نیز با توضیحات بالا منطبق است .

Self Supervised GAN ۲-۳

این GAN با توجه به معماری پیشنهاد شده Train نمی شد و احتمالاً اشتباهی در پیاده سازی معماری آن یا طرح معماری پیشنهاد شده وجود داشته باشد . به هر حال کد پیاده سازی و آموزش آن وجود در notebook دارد .