



دانشگاه صنعتی امیرکبیر
(پای تکنیک تهران)

پاسخ تمرین هفتم شبکه های عصبی

استاد درس جناب آقای دکتر صفابخش

تدریس یار جناب آقای مهندس اسدی

نیمسال دوم سال تحصیلی ۱۴۰۰-۱۴۰۱

محسن عبادپور

شماره دانشجویی: ۴۰۰۱۳۱۰۸۰

ایمیل: m.ebadpour@aut.ac.ir

فهرست پاسخ ها

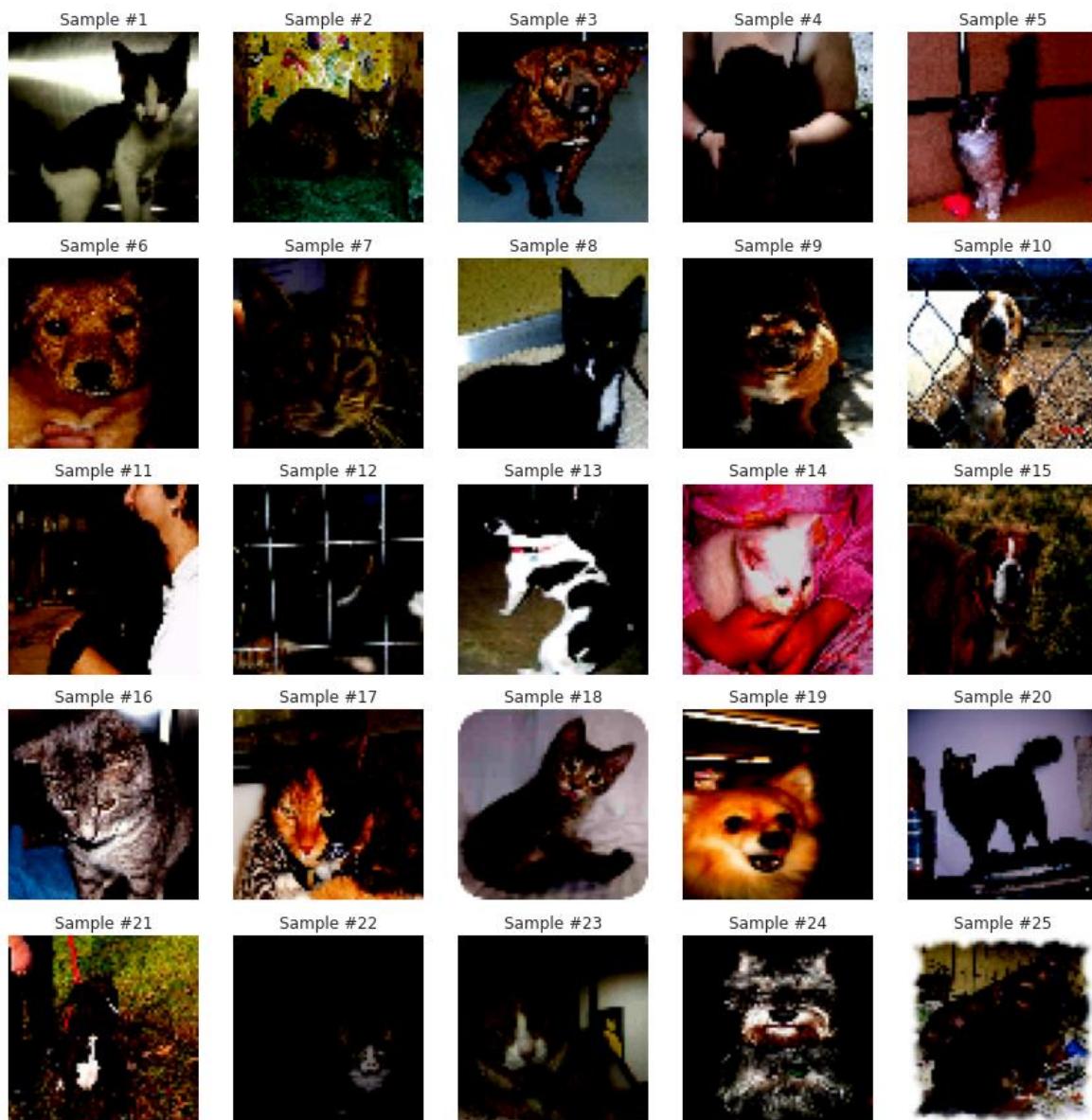
۲	توضیحات اولیه، بارگزاری داده ها و پیش پردازش
۳	مسئله اول
۴	مسئله دوم
۵	مسئله سوم
۶	مسئله چهارم
۱۴	مسئله پنجم
۱۵	مسئله ششم
۱۹	مسئله هفتم
۲۰	مسئله هشتم

توضیحات

توضیحات اولیه، بارگزاری داده ها و پیش پردازش

برای پیاده سازی و پاسخ گویی به این سری از تمرینات از زبان پایتون و کتابخانه keras برای ایجاد و آموزش مدل ها استفاده شده است.

مجموعه داده مورد نظر دارای تصاویر سگ و گربه در انواع سایز و موقعیت ها می باشد لذا برای یکسان سازی ابعاد، کل تصاویر مجموعه داده به سایز 64×64 تبدیل شده است. همچنین جهت همگرایی بهتر و سریع تر، مقادیر پیکسل های تصاویر در رنج -1 تا $+1$ نرمال شده اند. دلیل این کاهش سایز در این سطح محدودیت زمانی برای آموزش مدل و بالا نرفتن نمایی تعداد پارامتر های قابل یادگیری می باشد. برای آموزش شبکه ها در طول تمرینات از 10000 تصویر بصورت تصادفی استفاده شده است و تعداد تکرار در آزمایش های مختلف بین 100 تا 250 تکرار بوده است. همچنین سایز دسته های آموزشی، 256 نمونه در هر دسته به ازای داده های واقعی در نظر گرفته شده است.



مسئله اول

مسئله اول

همانطور که در درس آموختیم شبکه های مولد تقابلی از دو جز مولد و تمایزگر تشکیل شده است و آموزش این دو شبکه توامان و کنار هم دیگر انجام می شود و هدف این دو در مقابل همدیگر قرار دارد. شبکه مولد تلاش میکند داده ای تولید کند تا شبکه تمایزگر نتواند تشخیص دهد که آیا داده تولیدی می باشد یا داده واقعی می باشد و شبکه تمایزگر نیز تلاش می کند داده ای ورودی را بدرستی تشخیص دهد که آیا داده واقعی است یا تولیدی. لازم به ذکر است که معماری شبکه های مولد و تمایزگر کاملا مستقل از هم بوده و می تواند هر نوعی داشته باشد. این تلاش برای تشخیص داده واقعی از فیک برای تمایزگر و همچنین تلاش برای فریب تمایزگر توسط مولد یک فضای رقابتی بین این دو به وجود می آورد که میتوان آن را به یک مسئله min/max دو نقشی تشییه کرد که در آن تمایزگر تلاش میکند خطای خود یعنی تشخیص داده واقعی را کم کند و در مقابل مولد تلاش می کند خطای تمایزگر را افزایش داده و داده ای تولید کند که قابل تفکیک نباشد.

آموزش دو شبکه مولد و تمایزگر بدین صورت انجام می شود که اگر به فرض سایز هر دسته آموزشی را ۳۲ در نظر بگیریم؛ ۳۲ داده توسط مولد تولید شده(لیل ۰) و به همراه ۳۲ داده اورجینال(لیل ۱) به مدل تمایزگر به تنهایی داده می شود تا آموزش بیند و بتواند داده ها را تشخیص دهد که کدام داده واقعی است و کدام داده تولیدی.

سپس مولد و تمایزگر پشت سر هم پشته شده و تمایزگر non-trainable می شود. حال همان ۳۲ تصویر فیک با لیل تصویر اورجینال(لیل ۱) به شبکه پشته شده داده می شود تا آموزش بیند. چون تمایزگر می تواند داده فیک را جدا کند و یک گام از مولد جلو می باشد، لیل را تشخیص داده و به ازای لیل ۱ برای ازای داده فیک، خطایی در شبکه ایجاد شده و این خطای در شبکه انتشار پیدا کرده و وزن های مولد برای تولید تصاویری با کیفیت تر و نزدیک به تصاویر اورجینال بروزرسانی می شود. بدین صورت آموزش یک دسته پایان می یابد و این روند برای آموزش دسته دیگر مجدد شروع می شود.

مسئله دوم

مسئله دوم

در طی درس و در مباحث CNN ملاحظه کردیم که از لایه های کانولوشنی برای استخراجی ویژگی خاص من downsampling (در صورت valid در نظر گرفتن) استفاده می شود و بر اساس همسایگی های هر پیکسل صفحات ویژگی جدید حاصل می شود. در لایه های معکوس کانولوشنی نیز به جهت upsampling انجام می گیرد تا صفحات ویژگی حاصل گسترده تر باشد و فرآیند کلی آن تقریباً شبیه به عادی ولی به ترتیب برعکس می باشد.

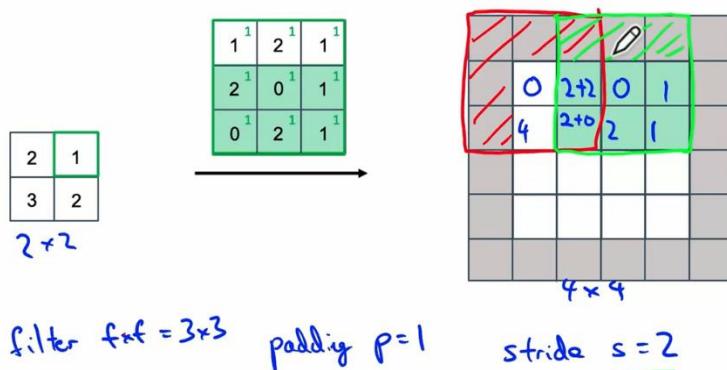
$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1$$

- n_{in} : number of input features
- n_{out} : number of output features
- k : convolution kernel size
- p : convolution padding size
- s : convolution stride size

اگر رابطه اصلی Convolution برای محاسبه سایز تصویر خروجی را در نظر گرفته و سایز ورودی را بر حسب سایز خروجی بنویسیم، ابعاد خروجی حاصل از Transpose Convolution حاصل می شود و با اختصاص padding و stride مورد نظر می تواند سایز تصویر خروجی را تعیین نماییم.

برای توضیح چگونگی انجام Andrew Ng از این [لینک](#) استفاده شده است که مربوط به آقای Andrew Ng می باشد:

Transpose Convolution



حال که ابعاد تصویر خروجی به همراه padding مشخص می شود، در هر مرحله مقدار متناظر با پیکسل پردازشی را در کل مقادیر فیلتر ضرب کرده و پیکسل هایی که در سلول های غیر حاشیه قرار می گیرند را در تصویر خروجی می نویسیم. به ازای پیکسل هایی که چندین پیکسل در تصویر ورودی مقدار برای آن تولید کرده باشند، مجموع آنها را برای پیکسل در نظر می گیریم.

مسئله سوم

مسئله سوم

در اسلاید های درس دیدیم که تابع هزینه یا همان loss function بصورت توام و در یک تابع برای مولد و تمایزگر در نظر گرفته می شود که در قالب یک مسئله دو نقشی \min/\max میتوان متصور شد؛ اما همچنین در ادامه ملاحظه کردیم که می توان تابع هزینه شبکه های مولد و تمایزگر را از هم جدا کرده و مجزا تعریف نمود اما آیا واقعاً بایستی نوع توابع هزینه این دو باهم فرق داشته باشد؟

شبکه تمایزگر را به تنها ی می تواند یک شبکه عصبی classifier تلقی کرد چرا که صرفاً ورودی را دریافت نموده و مشخص می کند یا داده ورودی واقعی است یا تقلبی و از این رو، این شبکه با یک مسئله طبقه بندی دو کلاسه مواجه است لذا می توان توابع هزینه مورد بحث در خصوص مسائل دو کلاسه را به این زمینه(شبکه تمایزگر) نیز تعمیم داد و از آن استفاده نمود که Binary Cross-Entropy را می توان انتخاب مناسبی دانست چرا که خروجی آن بر اساس میانگین فاصله احتمالات واقعی و پیش بینی شده به ازای کلاس یک می باشد و لذا مینیمم کردن آن یعنی کاهش فاصله احتمال پیش بینی شده و احتمال واقعی که معادل است با تقویت این شبکه.

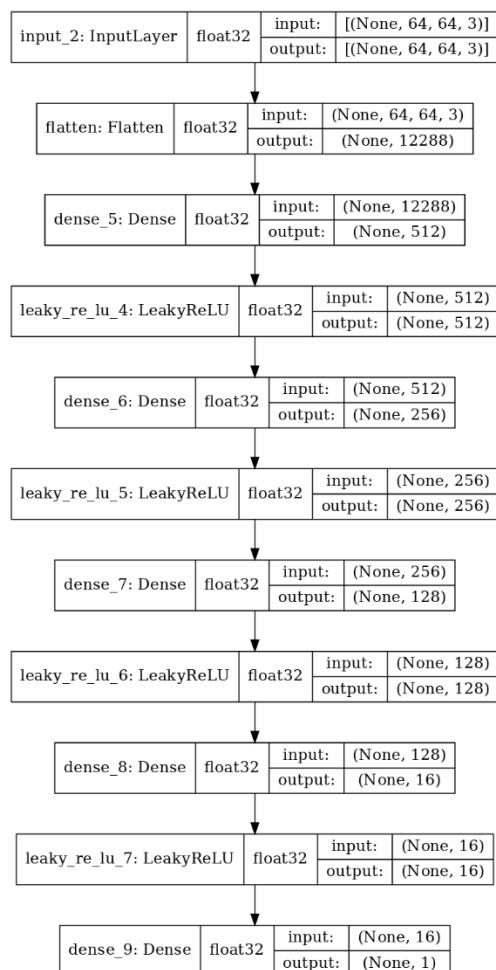
همچنین در پاسخ به مسئله اول دیدیم که آموزش شبکه مولد چگونه انجام می گیرد و ملاحظه کردیم که خطای موجود حاصل از این است که آیا تصویر تولیدی توانسته است که تمایزگر را فریب بدهد یا خیر لذا معیار ارزیابی شبکه مولد را نیز بدین گونه میتوان به عنوان یک مسئله طبقه بندی دو کلاسه در نظر گرفته و تابع هزینه فوق را برای آن استفاده نمود.

مسئله چهارم

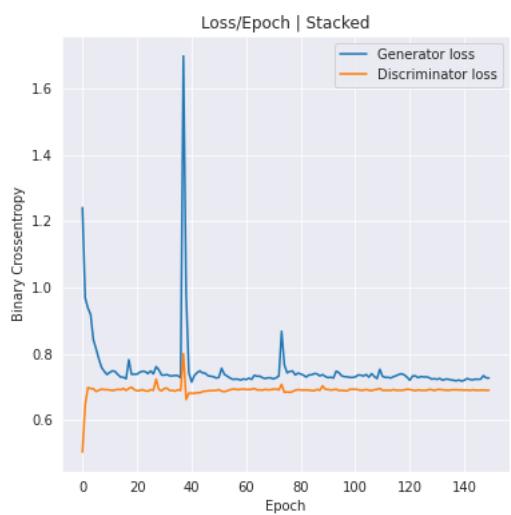
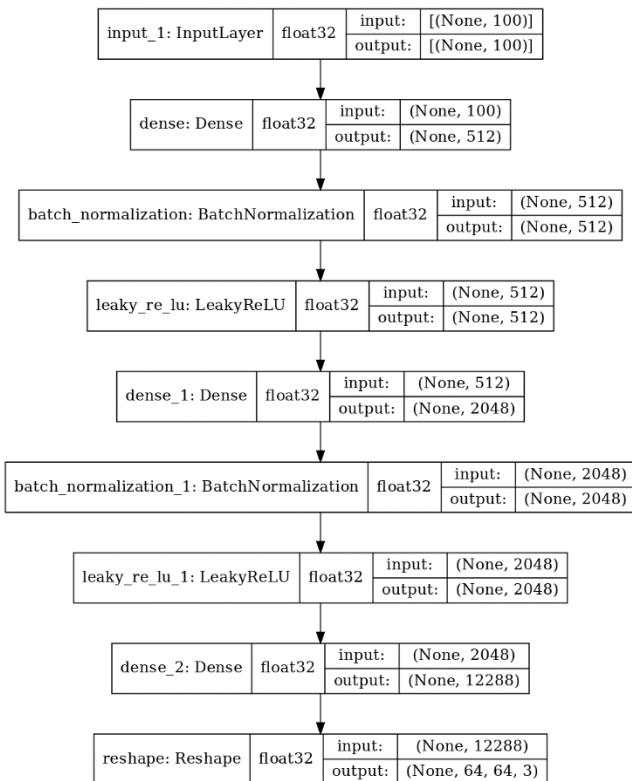
مسئله چهارم

همانطور که در مسئله بعدی مورد بحث قرار خواهد گرفت، یک از مشکلات رایج در آموزش شبکه های مولد تقابلی همگرایی سریع تمایزگر نسبت به مولد می باشد. لذا به جهت تئوری می توان انتظار داشت اگر تمایزگر را ثابت فرض کنیم؛ در صورت کاهش تعداد لایه های مولد، پیچیدگی آن نسبت به تمایزگر کمتر شده و سریع می تواند فرآیند یادگیری را طی کند و فرصت کافی برای آموزش دارد چرا که تمایزگر پیچیده بوده و آموزش آن نیازمند زمان می باشد و مولد از این زمان استفاده کرده و خطای خود را کاهش می دهد اما در مقابل مسلمان میتوان انتظار داشت کیفیت تصاویر تولیدی نیز دچار کاهش خواهد شد. به جهت نشان دادن مورد فوق و بصورت عملیاتی به ازای ۲، ۳ و ۴ لایه ای تکرار آموزش FCGAN شبکه fully-connected می دهیم که در آن از بهینهساز adam و با ۱۵۰ تکرار آموزش انجام شده است. در طی آزمایشات، معماری شبکه تمایزگر ثابت بوده و صرفاً معماری شبکه مولد شامل تغییرات بوده است. همانطور که در نمودار های حاصل قابل مشاهده می باشد، با کاهش تعداد لایه ها و پیچیدگی شبکه مولد، خطای آن نیز کمتر نتیجه شده است.

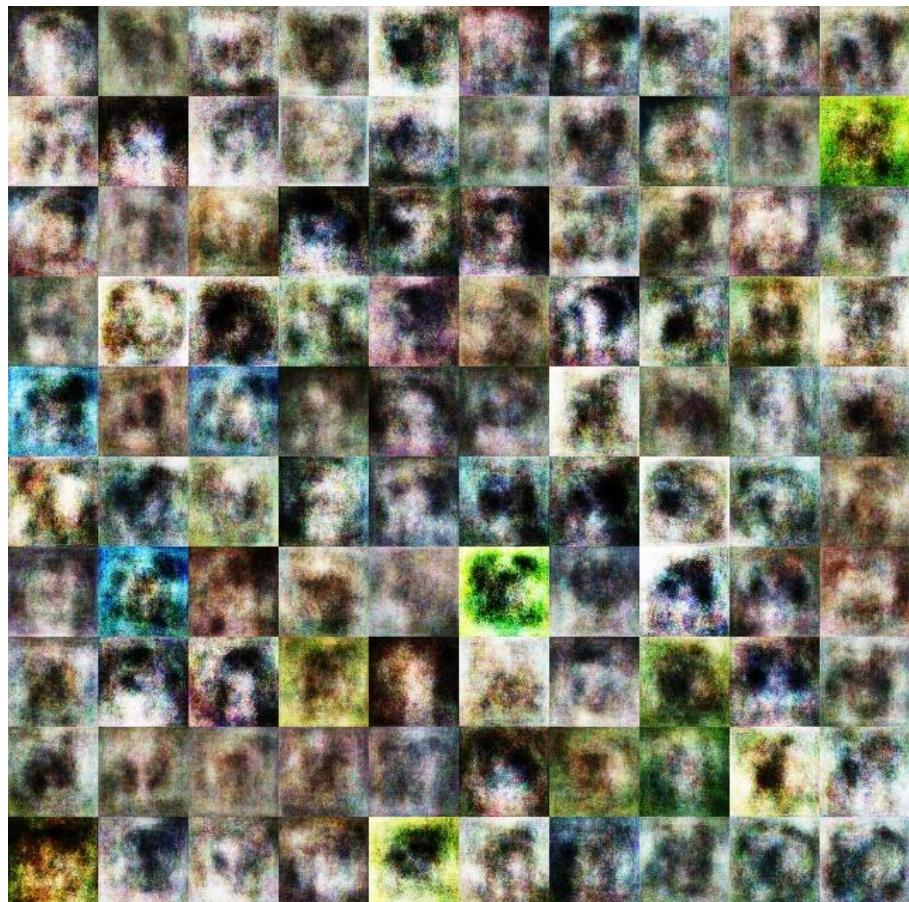
معماری تمایزگر:



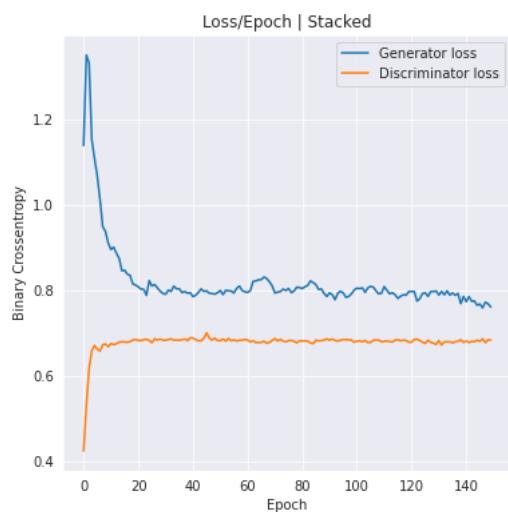
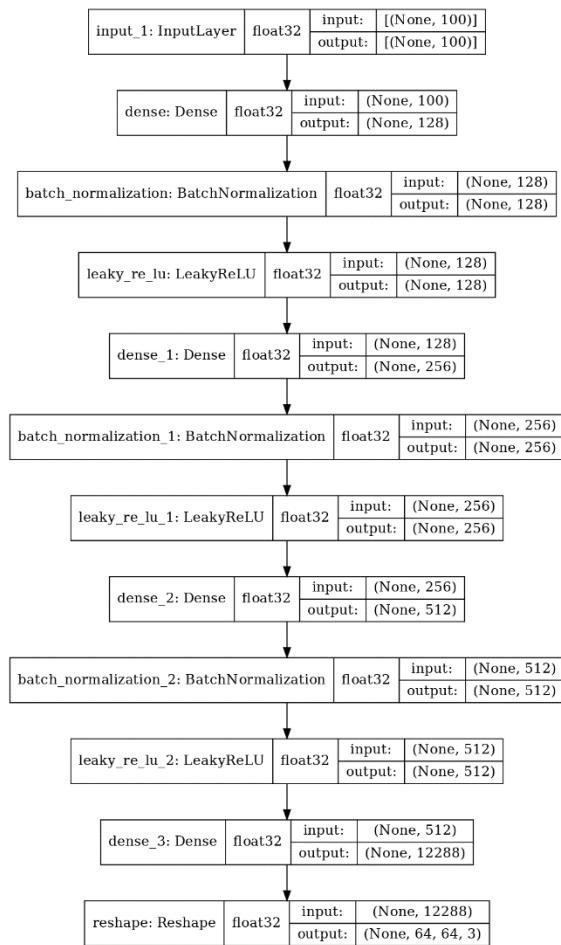
آزمایش اول به ازای دو لایه مخفی تماماً متصل برای مولد:



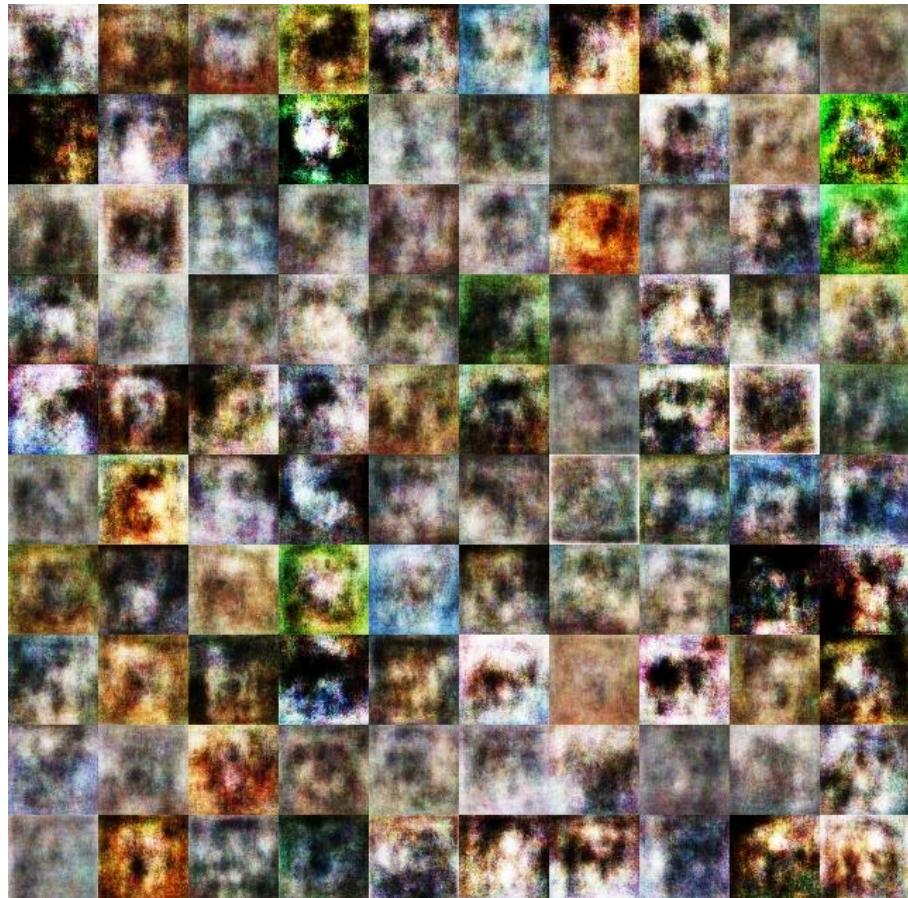
نمونه تصاویر تولید شده در گام نهایی:



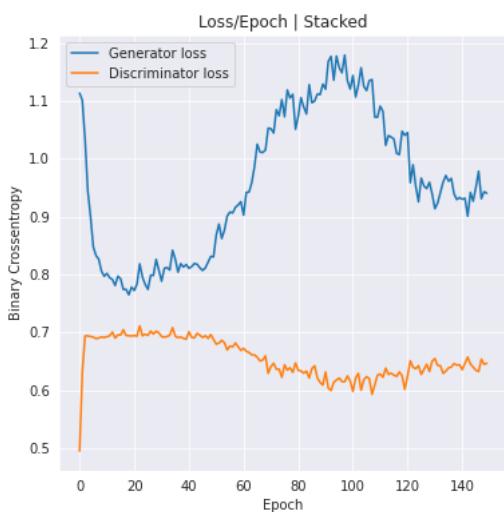
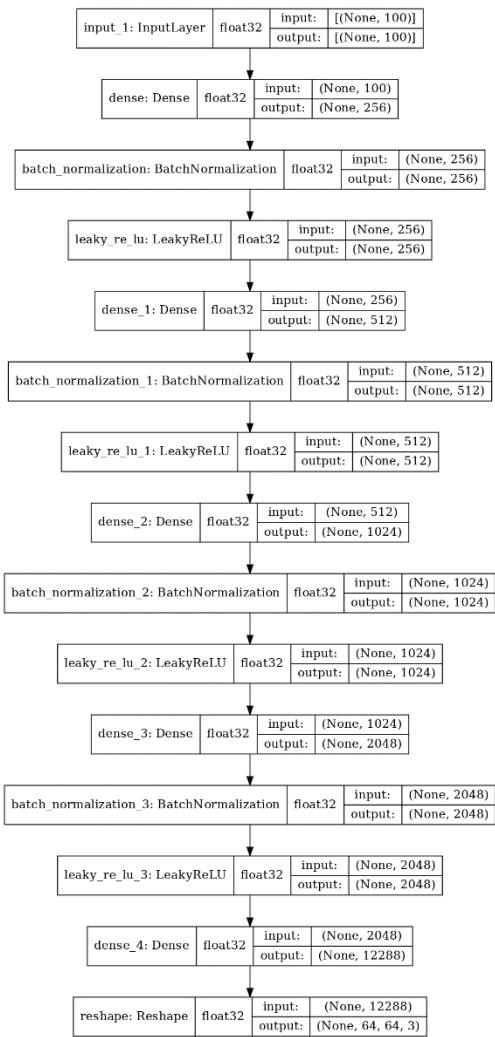
آزمایش دوم به ازای سه لایه مخفی تماماً متصل برای مولد:



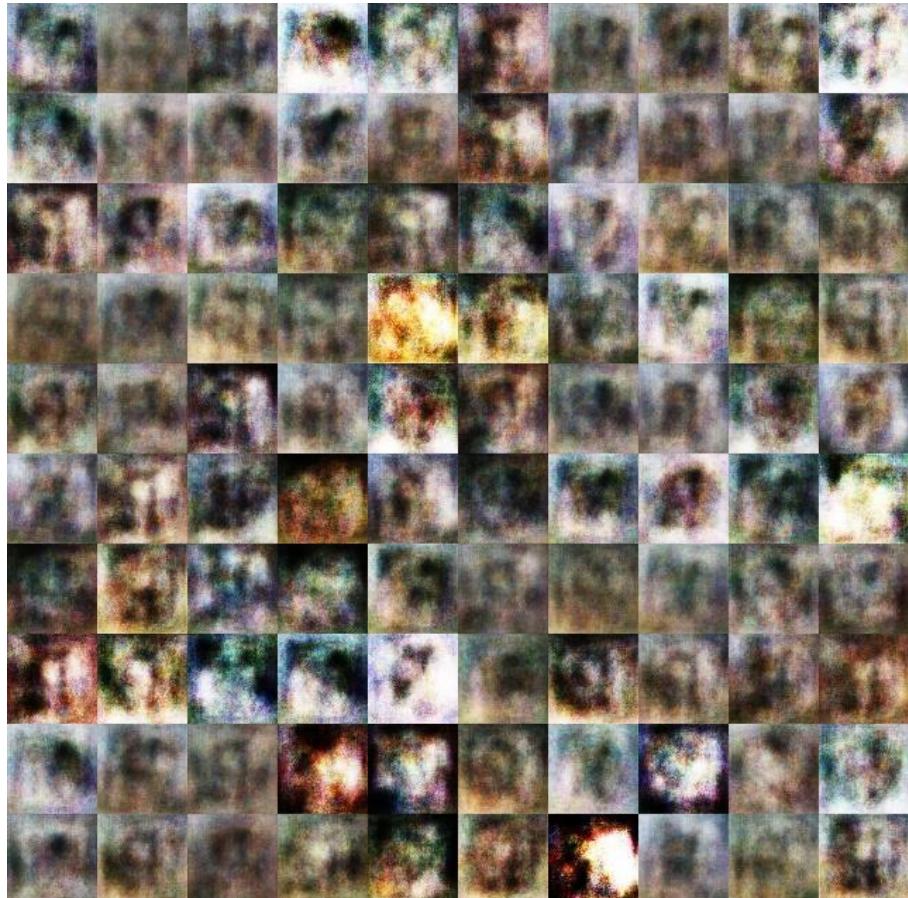
نمونه تصاویر تولید شده در گام نهایی:



آزمایش سوم به ازای چهار لایه مخفی تماماً متصل برای مولد:

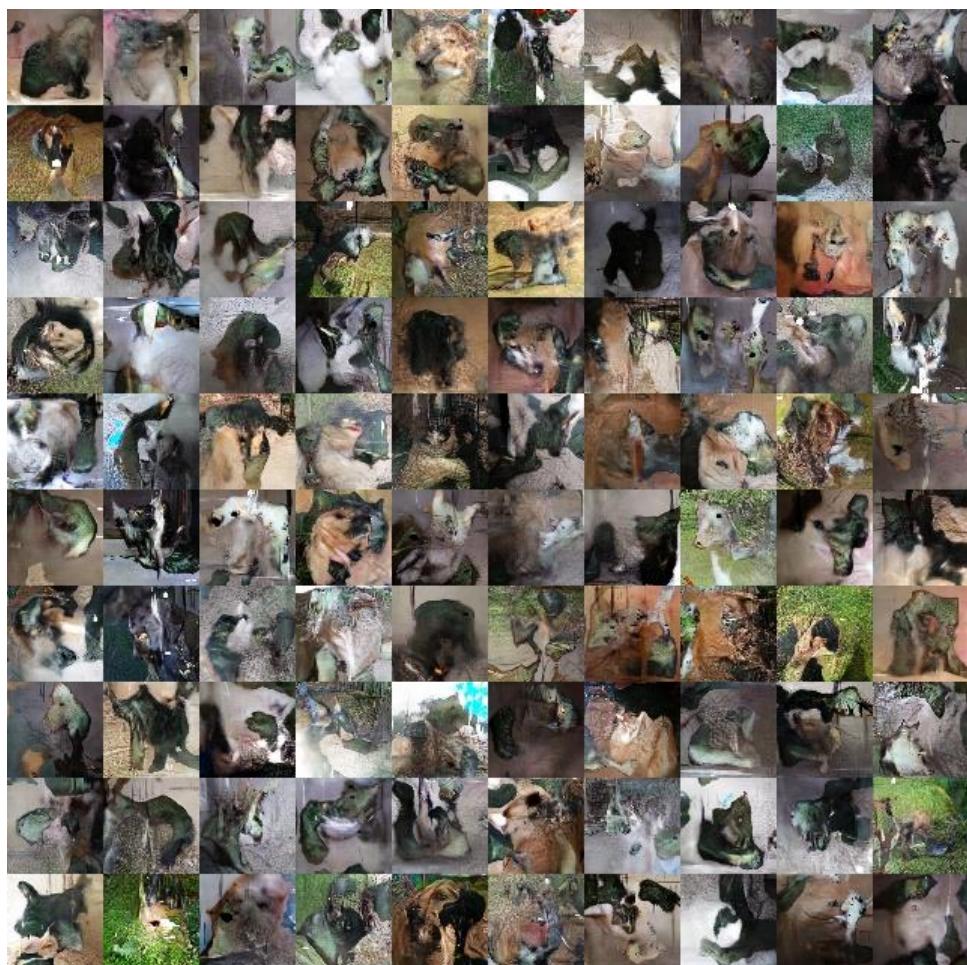
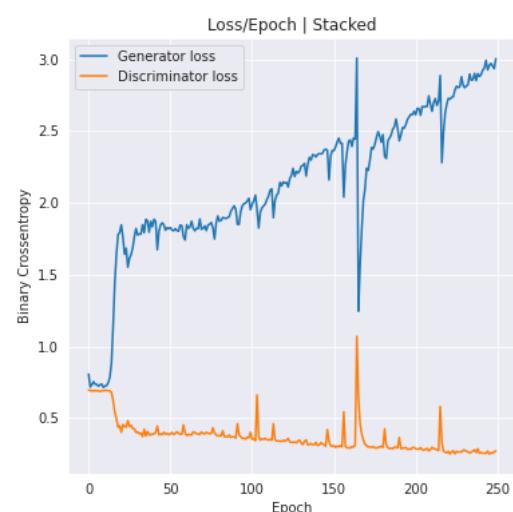


نمونه تصاویر تولید شده در گام نهایی:

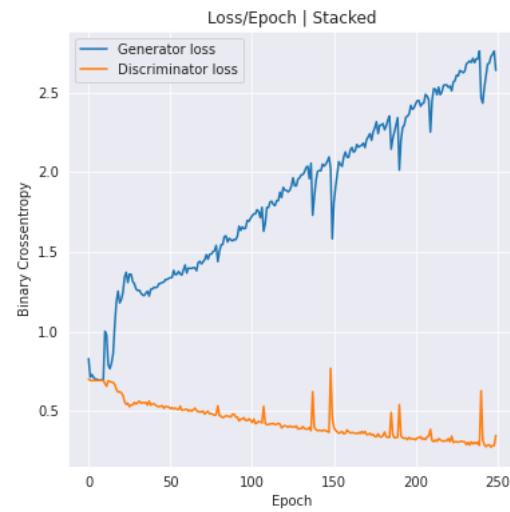


حال آزمایش تغییر تعداد لایه های شبکه مولد شامل Transposed Convolution ها را به ازای مقادیر ۱، ۲، ۳ و ۴ برای معماری DCGAN انجام دهیم. برای آزمایش های انجام شده ۲۵۰ تکرار در نظر گرفته شده و همچنین از بهینه‌ساز adam بهره گرفته شده است. در این آزمایش ها معماری تمایزگر ثابت بوده و فقط معماری مولد تغییر یافته است. برای اینکه فضای گزارش شلوغ نشود، تصاویر دقیق معماری ها در فولدر ضمیمه قرار گرفته شده است و در زیر نمودار خطای ها و تصاویر حاصل قابل ملاحظه است.

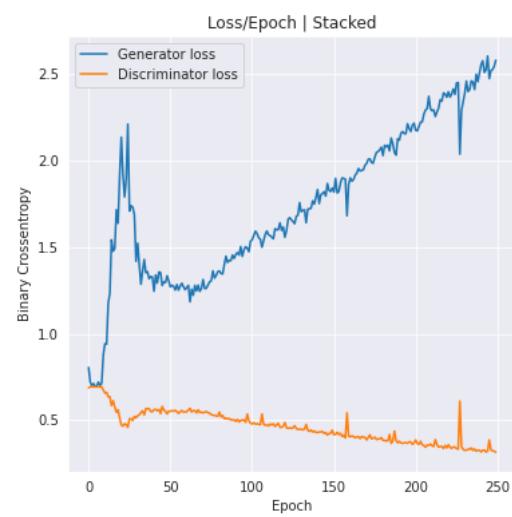
آزمایش اول به ازای یک Transposed Convolution برای مولد:



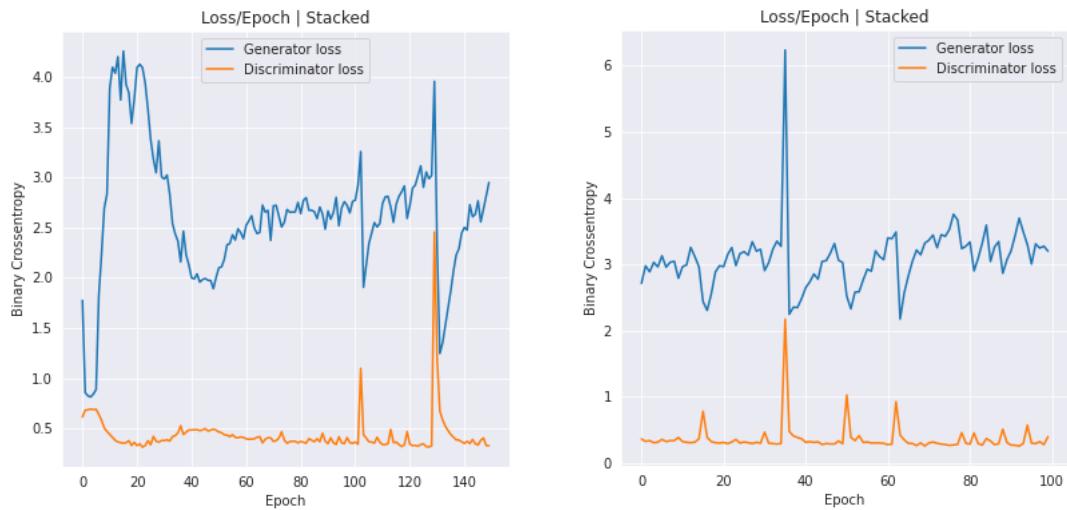
آزمایش دوم به ازای دو Transposed Convolution برای مولد:



آزمایش سوم به ازای سه Transposed Convolution برای مولد:



آزمایش چهارم به ازای چهار Transposed Convolution برای مولد(آموزش در دو مرحله تکمیل شده است):



همانطور که از نتایج مشخص است با افزایش لایه های مولد، مقدار خطای شبکه مولد افزایش یافته و فرصت کافی برای آموزش نداشته است چرا که تعداد پارامتر های مورد یادگیری نیز افزایش داشته است. کیفیت تصاویر خروجی نیز پایین تر حاصل شده است.

مسئله پنجم

مسئله پنجم

برای متعادل سازی می توان تکنیک های مختلف در مراحل مختلف آموزش شبکه ها در نظر گرفت که به چند مورد در زیر اشاره می شود:

- **تزریق نویز به داده های ورودی:** روی داده های ورودی و به صورت تصادفی در هر تکرار از آموزش نویز وارد شود و روند

تزریق نویز طی زمان بصورت کاهشی انجام گیرد و با گذر زمان تزریق نویز به حداقل ترین حالت برسد. این کار باعث می شود

یادگیری تمایزگر به تعویق افتاد چرا که برای تشخیص داده های واقعی نیز با چالش مواجه است. (با استدلالی شبیه به این میتوان

بر برچسب ها نیز نویز اندک وارد کرد تا مولد همواره در حال آموزش و یادگیری باشد)

- **استفاده از تمایزگر با معماری پیچیده تر:** اگر در هر سطحی تمایزگر را یک گام پیچیده تر از مولد در نظر بگیریم، در آن

صورت تمایزگر زمان بیشتری برای آموزش و یادگیری نیاز خواهد داشت.

- **استفاده از نرخ یادگیری متفاوت:** اگر نرخ یادگیری مولد را بیشتر از تمایزگر در نظر بگیریم، تاخیر حاصل می تواند فرصت

یادگیری برای مولد را ایجاد کند. البته باید بحث جهش و عبور از مینیمم را مد نظر گرفت که دچار آن نشد.

- **تعداد آموزش کمتر تمایزگر:** در هر تکرار از آموزش شبکه می توانیم تعداد آموزش شبکه مولد را بیشتر از تمایزگر انجام داده

و تعداد اصلاح وزن های بیشتری را برای آن شاهد باشیم در این صورت می توان آموزش این دو شبکه را متعادل نمود؛ مثلا در

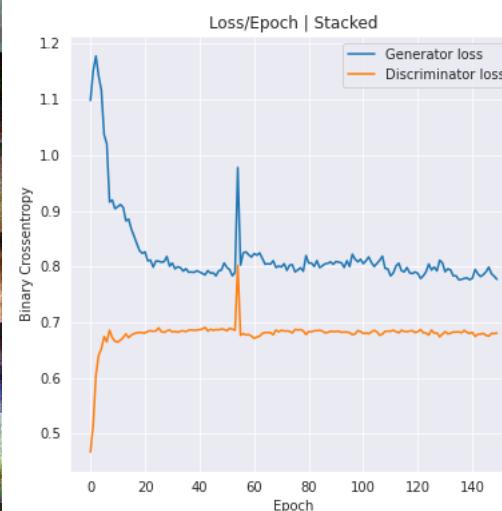
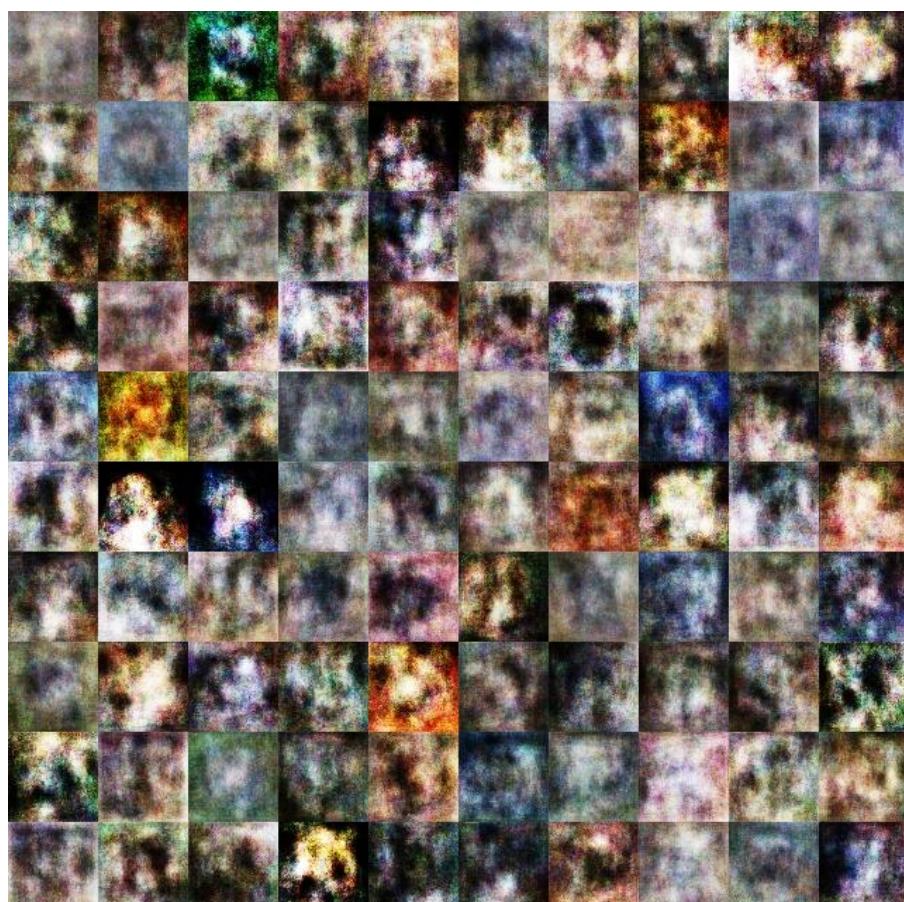
هر epoch به ازای یک بار اصلاح وزن و آموزش شبکه تمایزگر، سه بار شبکه مولد را آموزش دهیم.

مسئله ششم

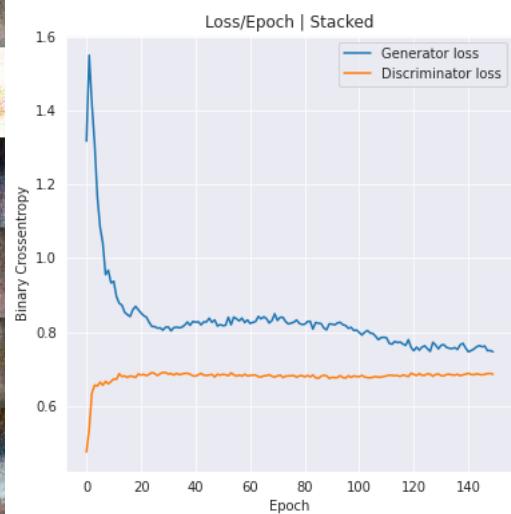
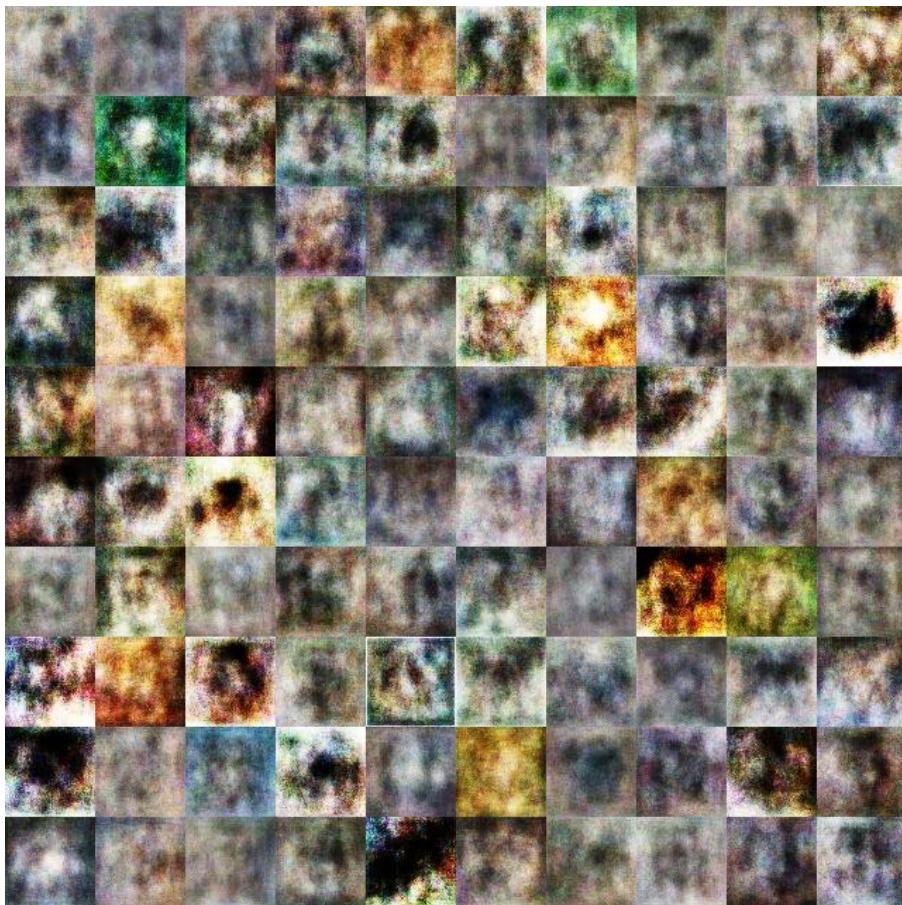
مسئله ششم

به ازای یک شبکه از FCGAN (معماری سه لایه‌ی ۱۲۸-۲۵۶-۵۱۲ نورونی) و به ازای یک شبکه از DCGAN (معماری سه لایه‌ی هر کدام را با سه مقدار شدت نویز آموزش می‌دهیم تا تاثیر شدت نویز ورودی را بر آموزش شبکه مولد تقابی ارزیابی نماییم. در زیر نمودار خط و همچنین تصاویر خروجی برای هر آزمایش آورده شده است.

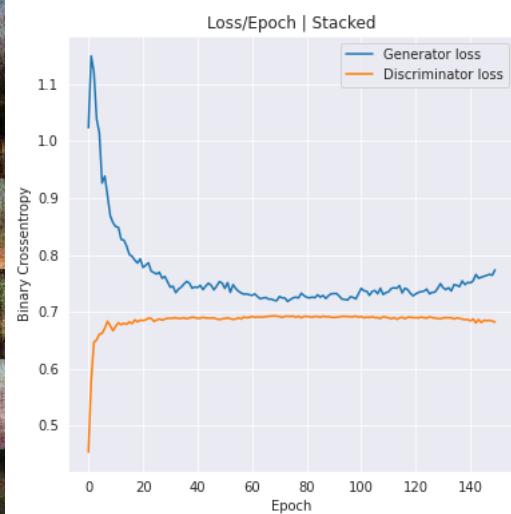
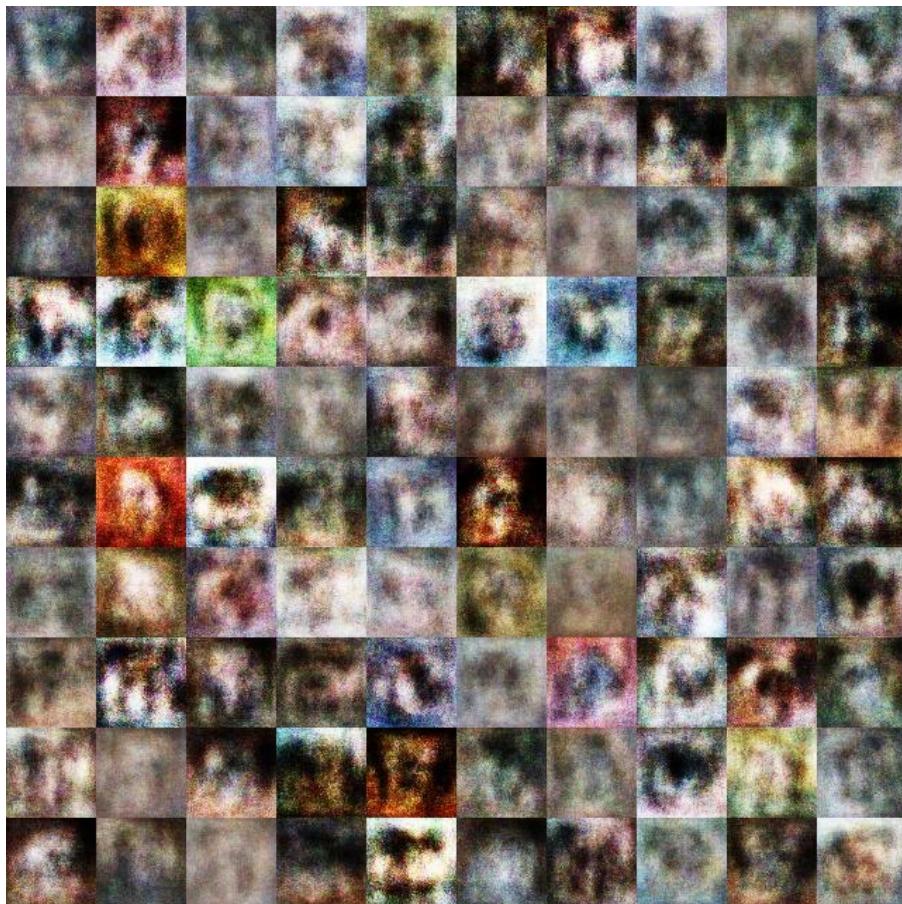
آموزش شبکه FCGAN به ازای شدت نویز ورودی ۰.۱ :



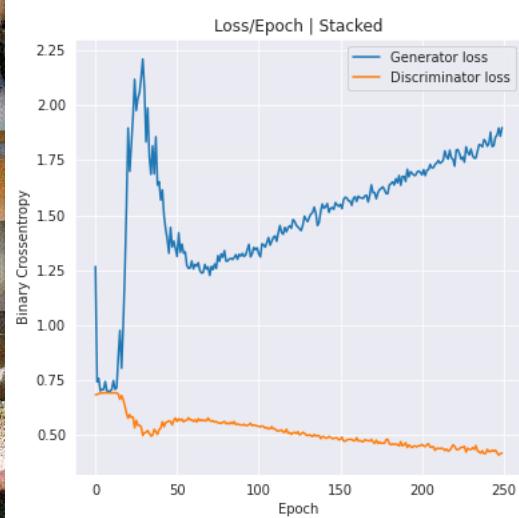
آموزش شبکه FCGAN به ازای شدت نویز ورودی ۰.۲ :



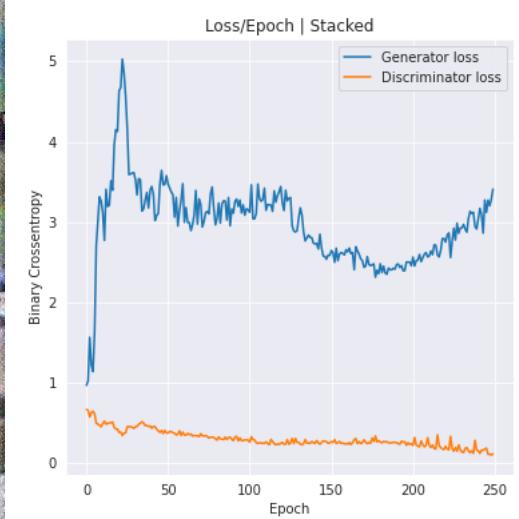
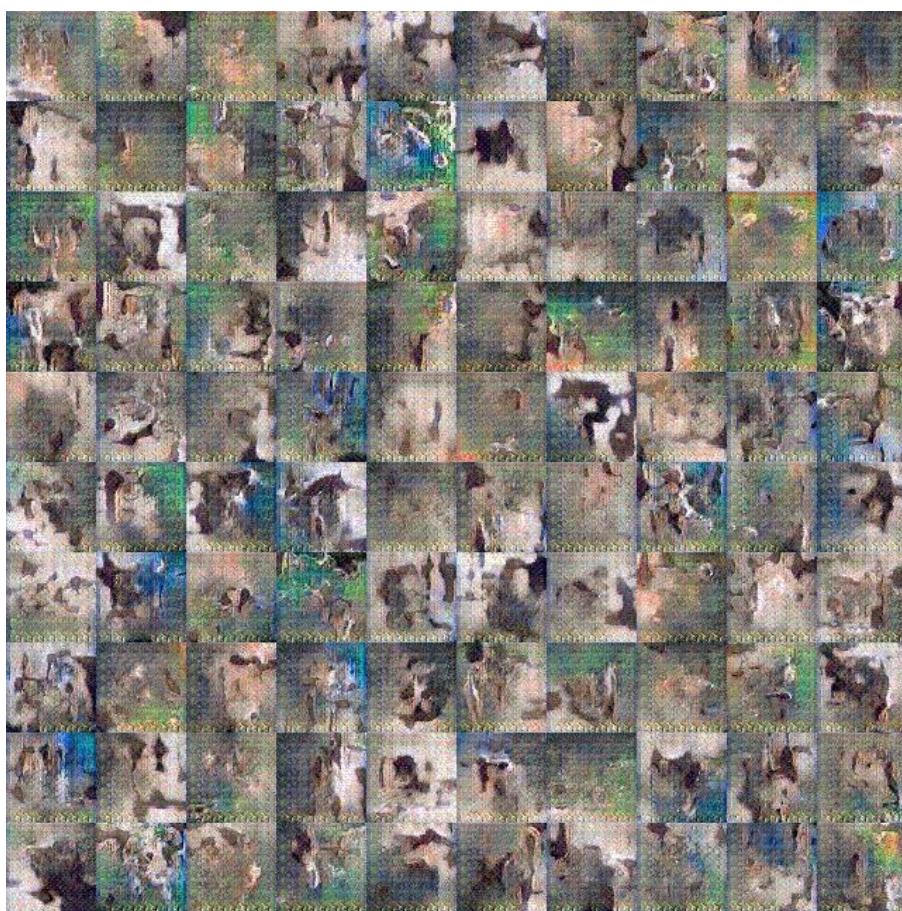
آموزش شبکه FCGAN به ازای شدت نویز ورودی ۰.۴ :



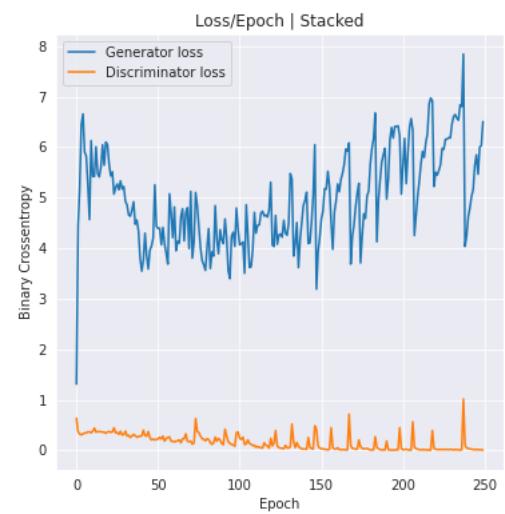
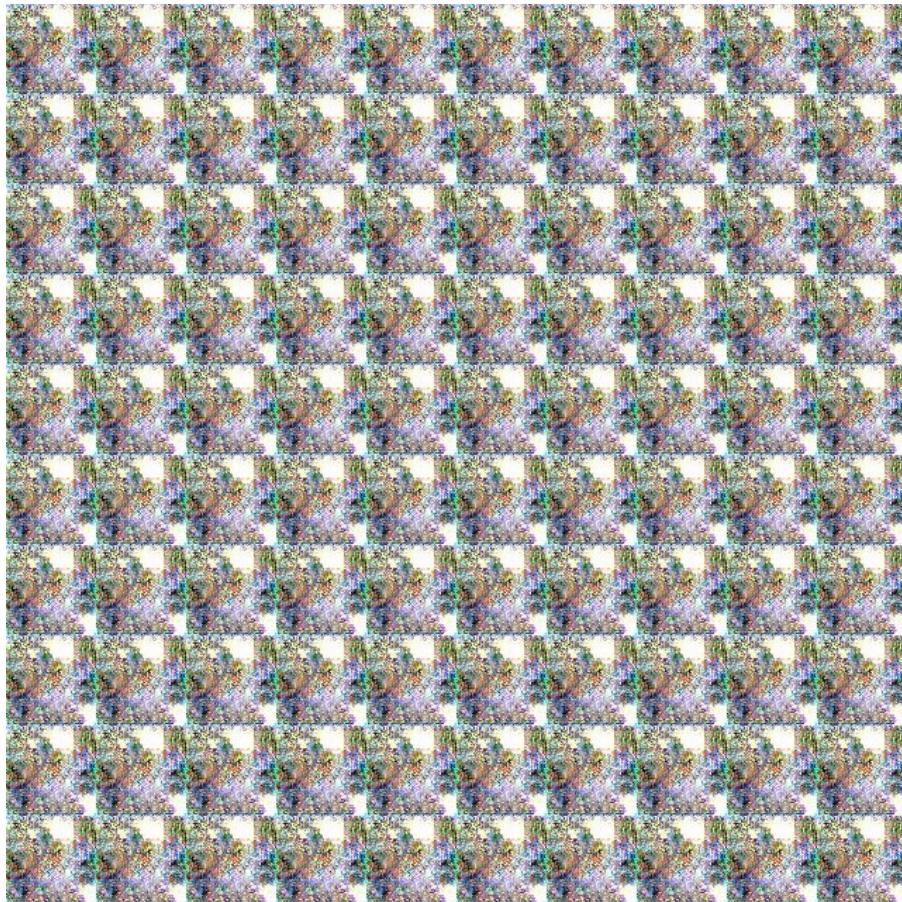
آموزش شبکه DCGAN به ازای شدت نویز ورودی ۰.۱ :



آموزش شبکه DCGAN به ازای شدت نویز ورودی ۰.۲ :



آموزش شبکه DCGAN به ازای شدت نویز ورودی : ۰.۴



نتیجه گیری:

نویز اعمالی یک نویز با احتمال نرمال بوده و منظور از نویز با شدت ۰.۲ این می باشد که نویز نرمال با میانگین صفر و انحراف معیار ۰.۲ به مقادیر ورودی افروده شده است.(لازم به اشاره مجدد است که داده ها در رنج $-1 + 1$ اسکیل شده اند) اگر نمودار های شبکه FCGAN را مورد مطالعه و ارزیابی قرار دهیم قابل مشاهده است که نمودار های خطای مولد و تمایزگر با افزایش نویز به هم نزدیک شده و به حالت متعادل تری همگرا می شوند و مدل می تواند نتایج بهتری حاصل کند و این در تصاویر تولیدی نیز قابل مشاهده است(در مقایسه ۰.۱ و ۰.۲ نزدیکتر شدن خطوط نمودار محسوس است) اما افزایش نویز خود دارای حد آستانه می باشد چرا که اگر نمودار نویز اعمالی ۰.۲ و ۰.۴ را مقایسه کنیم می بینیم پس از حدی، فاصله خطاهای شروع به افزایش کردن می کند چرا که آموزش تمایزگر مختل شده و مسیر یادگیری به درستی ادامه پیدا نمی کند(همواره با نویز مواجه است). حال اگر نمودار های شبکه DCGAN را مورد ارزیابی قرار دهیم می بینیم که اعمال نویز ۰.۱ خصم کاهش شبی افزایش خطای مولد، تصاویر تولیدی نیز جالب تر حاصل شده است. اما اعمال نویز ۰.۲ و ۰.۴ باعث بهبود نشده و مجدد باعث بهم خوردن تعادل شده است. دلیل این امر را می تواند اعمال فیلتر های مکانی دانست چرا که وقتی فیلتر convolution اعمال می کنیم یک پنجره ای از ورودی ها مورد پردازش قرار میگیرد و وجود این همبستگی اجازه افزایش نویز بیشتر از این را نمی دهد(نویز تومان اثر می گذارد). پس می توان نتیجه گرفت که اعمال نویز مطلوب بوده و باعث تعادل سازی می شود و تصاویر خروجی نیز با کیفیت تر می باشد اما باید به خاطر داشت افزایش نویز دارای حد آستانه بوده و نمیتوان افزایش را کنترل نشده انجام داد چرا که باعث اثر سو می شود.

مسئله هفتم

مسئله هفتم

برای مقایسه بین دو شبکه DCGAN و FCGAN که به ازای معماری های متفاوت در سوالات فوق آموزش داده شده است را معیار ارزیابی قرار می دهیم؛ بصورت کاملا محسوس قابل ملاحظه است که تصاویر تولیدی به جهت بصری به ازای DCGAN از کیفیت بالاتری برخوردار است و تصاویر تولیدی را بیشتر می توان به سگ یا گربه تشبیه نمود و این در حالی است که تعداد پارامتر های شبکه DCGAN نیز نسبت به معماری تقریبا مشابه در FCGAN کمتر است چرا که اتصالات تماما متصل نمی باشد و این خود دلیل می شود پیچیدگی کمتر شده و در نتیجه آموزش نیازمند زمان کمتری باشد.

بر اساس توضیحات فوق، به نظر بnde تولید تصاویر توسط شبکه مولد تقابلی با معماری DCGAN بهتر بوده و از کیفیت بالاتری برخوردار می باشد لذا انتخاب مناسب می باشد؛ دلیل بهتر بودن کیفیت را می توان لحاظ نمودن ارتباط پیکسل های همسایه با یکدیگر در ارتباط داشته باشند.(تغییرات sharp رنگی بین پیکسل های همسایه وجود نداشته باشد)

در درس پردازش تصویر آموخته ایم که بصورت کلی در تصاویر دیجیتال، پیکسل های همسایه همبسته بوده و دارای رنگ تقریبا مشابه به هم می باشند؛ در شبکه FCGAN ارتباط مقداری و رنگی بین پیکسل های همسایه لحاظ نمی شود ولی در شبکه DCGAN بواسطه لایه کانولوشنی معکوس این ارتباط لحاظ می شود و یک دلیل اساسی برای تولید تصاویر جالب و معنی دار تر همین مورد می باشد.

مسئله هشتم

مسئله هشتم

در طی پاسخ گویی به مسائل فوق و به ازای هر شبکه‌ی آموزش داده شده، تصاویر تولیدی(ماتریس 10×10) در گام نهایی نمایش داده شده است؛ همچنین تصاویر تولیدی برای هر مدل آموزش داده در هر شش epoch ذخیره شده و در فولدر ضمیمه قابل ملاحظه است که روند تولید تصاویر چگونه بوده است(ارزیابی با نویز تولیدی یکسان انجام شده است تا بتوان روند تغییرات را رهگیری نمود). لازم به ذکر است وزن شبکه‌های آموزش داده شده نیز ذخیره شده و قابل بارگزاری و تولید مجدد تصاویر می‌باشد اما به جهت اجتناب از افزایش نمایی سایز فایل آپلودی آنها در فولدر ضمیمه قرار نگرفته اند.(حجم مدل‌های آموزش داده شده تا 250 مگابایت می‌باشد)

با عنایت به موارد ذکر شده همه تصاویر قبل مشاهده است، تصاویر تولید شده برای شبکه DCGAN با شدت نویز 0.1 و معماری سه لایه

در زیر آورده شده است که میتوان گفت تقریباً جزو تصاویر تولیدی مطلوب در این تمرين می‌باشد:

