



گزارش تمرین ششم درس شبکه‌های عصبی،

فاطمه غلامزاده

۹۹۱۳۱۰۰۳



بخش اول: FCGAN

(سوال اول)

برای آموزش شبکه، ابتدا تعدادی تصاویر جعلی توسط Generator تولید میشود سپس ترکیبی از این تصاویر جعلی با برچسب 0 و تصاویر واقعی با برچسب 1 به Discriminator داده می‌شود تا این شبکه برای تشخیص تصاویر جعلی از واقعی آموزش ببیند. در مرحله بعد آموزش Discriminator غیر فعال میشود و تنها بخش Generator آموزش می‌بیند. در واقع هنگامی که Discriminator آموزش داده می‌شود، می‌توان پیش‌بینی‌های آن را دریافت کرده و از آن به عنوان یک هدف برای آموزش Generator استفاده کرد و آن را با این هدف که متمایز کننده را فریب دهد آموزش می‌دهیم. به این صورت که یک بردار نویز (با توزیع تصادفی نرمال) تولید می‌شود و به عنوان ورودی به شبکه داده می‌شود. Generator یک تصویر در خروجی تولید کرده و Discriminator که قبلاً آموزش دیده جعلی بودن این تصویر را تشخیص می‌دهد و بدین ترتیب وزن لایه‌های Generator به گونه‌ای آپدیت می‌شود که میزان خطا را کاهش دهد و بتواند تصاویر نزدیک به واقعی تولید کند تا حدی که Discriminator نتواند جعلی بودن تصاویر را تشخیص دهد.

ساختار شبکه تولید شده در این سوال به این صورت است:

Discriminator

- لایه ورودی: تصاویری با ابعاد $32 \times 32 \times 3$ را دریافت می‌کند
- لایه پنهان: سه لایه پنهان از نوع Dense که به ترتیب ۲۰۴۸، ۱۰۲۴، ۵۱۲، ۲۵۶ نورون دارند و تابع فعالیت آنها LeakyReLU می‌باشد.
- لایه خروجی: لایه Dense با یک نورون با تابع فعالیت sigmoid که با دو مقدار صفر و یک در خروجی، جعلی/واقعی بودن تصویر دریافتی را مشخص می‌کند.

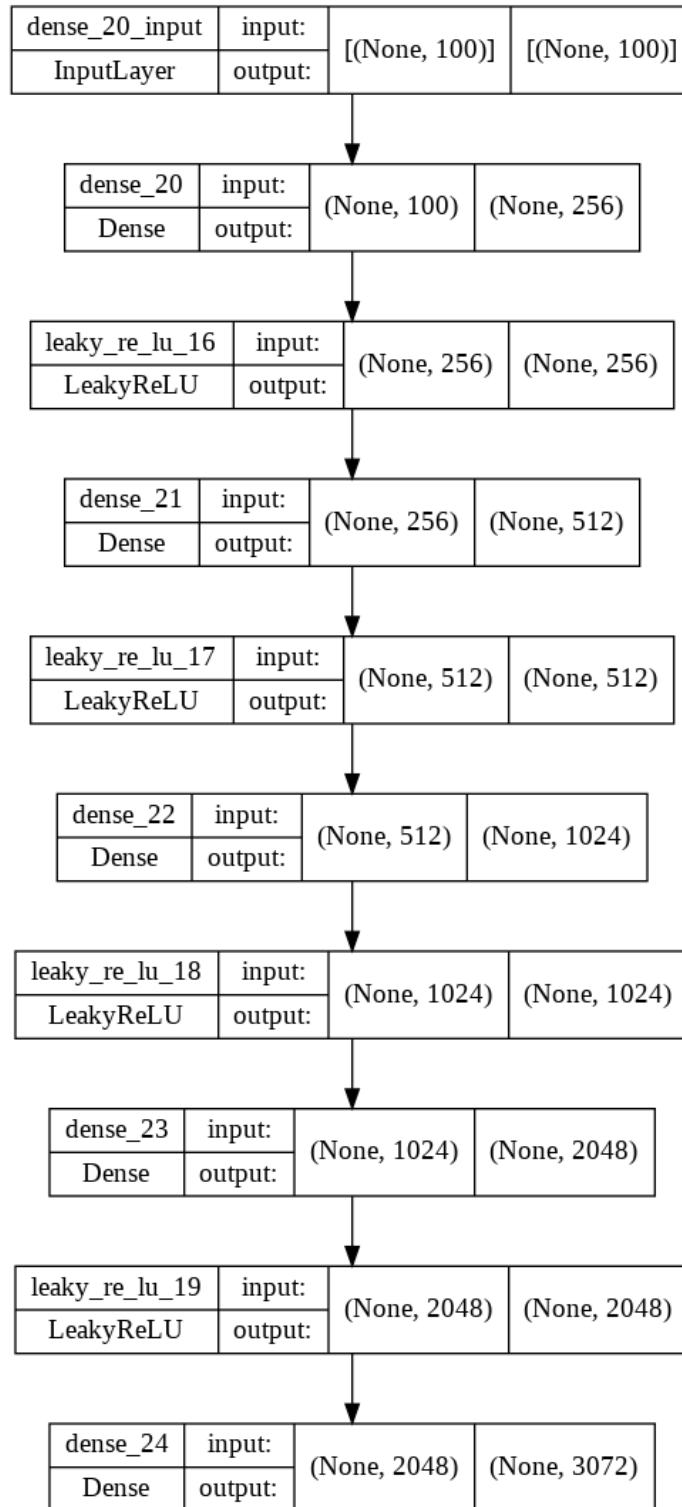
در شکل زیر ساختار discriminator به نمایش گذاشته شده است.

Generator

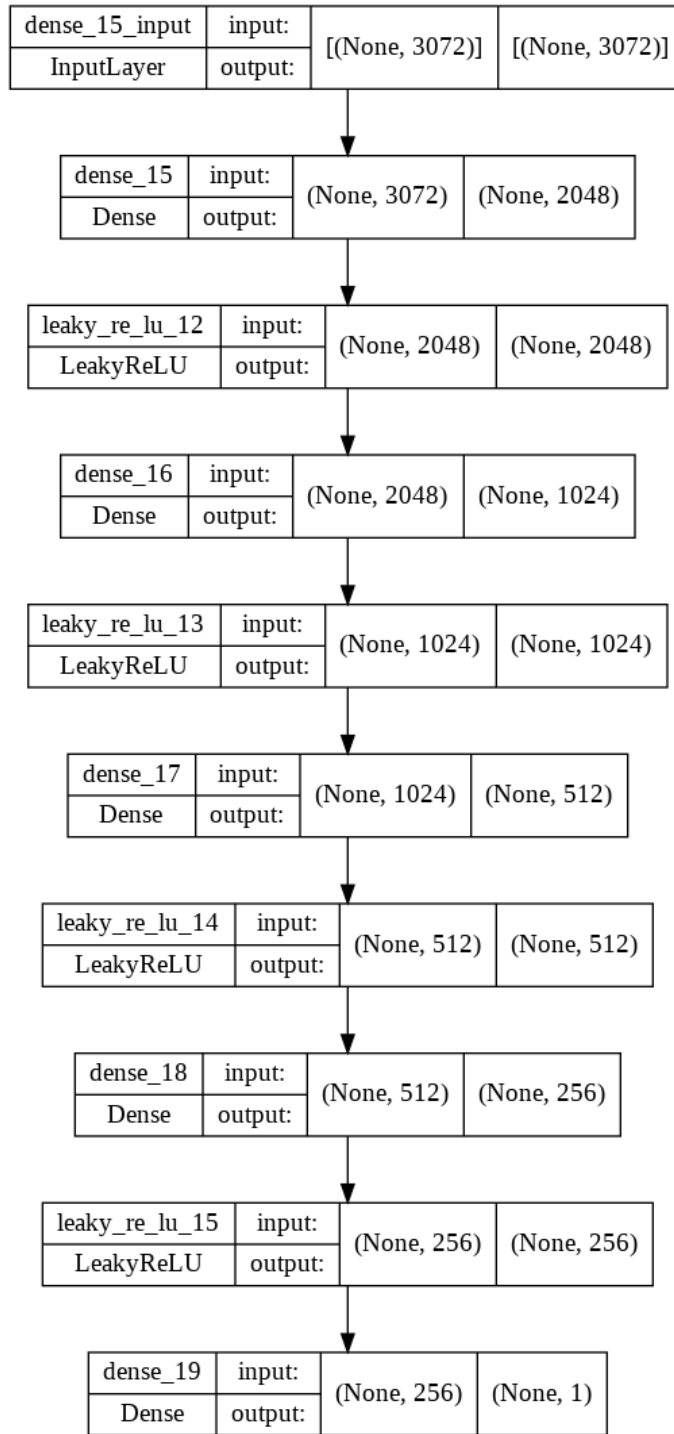
- لایه ورودی: ۱۰۰ نورون (برای دریافت یک بردار نویزی به طول ۱۰۰)

- لایه پنهان: سه لایه پنهان از نوع Dense که به ترتیب ۲۵۶، ۵۱۲، ۱۰۲۴، ۲۰۴۸ نورون دارند و تابع فعالیت آنها LeakyReLU می باشد

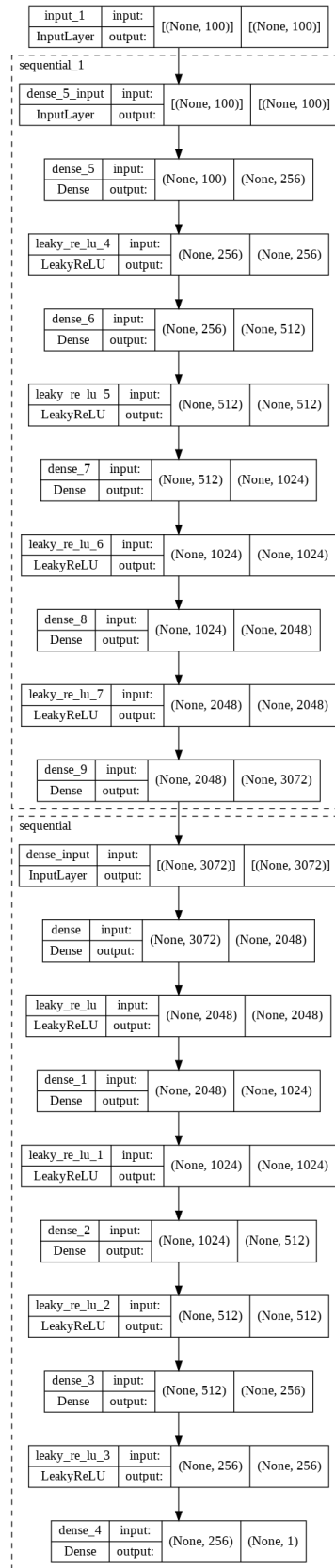
- لایه خروجی: لایه Dense $3 \times 32 \times 32 = 3072$ نورون می باشد
تابع فعالیت لایه خروجی تابع \tanh می باشد که مقادیر بردار خروجی را به بازه $[-1, 1]$ تبدیل میکند. (در فرایند نرمالسازی داده ها، مقادیر هر پیکسل به بازه $[-1, 1]$ تبدیل شده اند)
ساختارهای discriminator و generator و مدل کلی در شکل های ۱ و ۳ آورده شده است.



شكل ١ ساختار Generator



شکل ۲ ساختار Discriminator



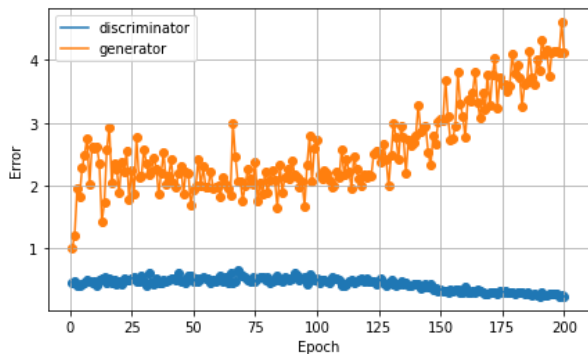
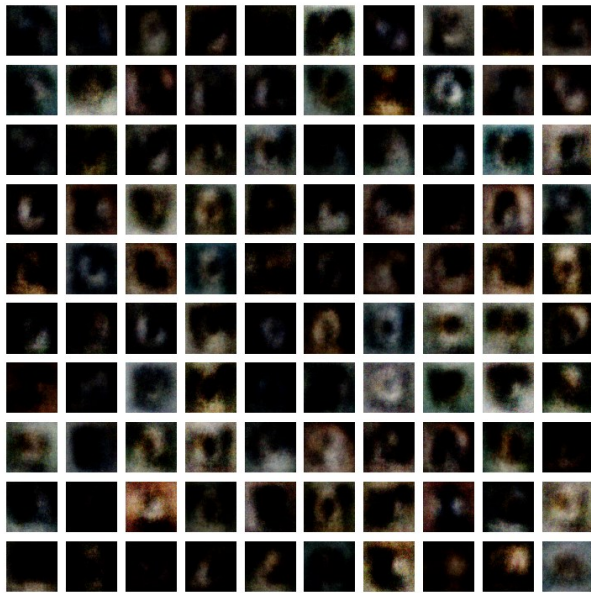
شکل ۳ ساختار DCGAN

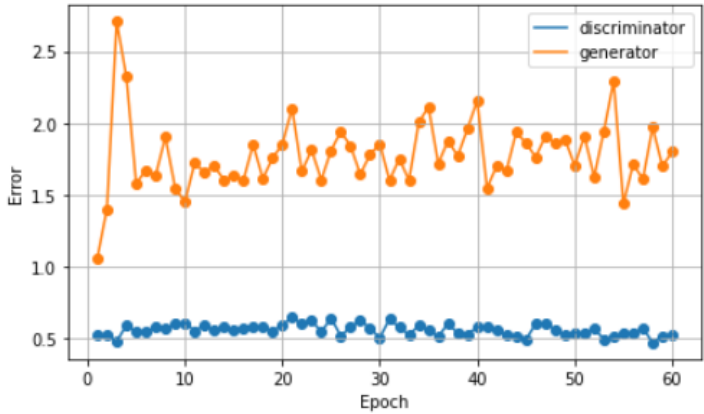
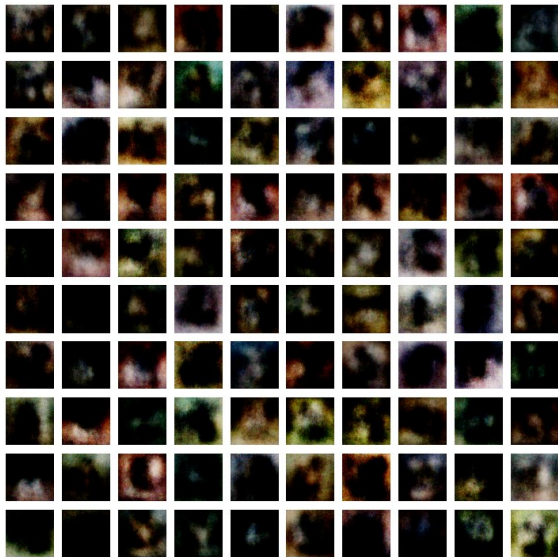
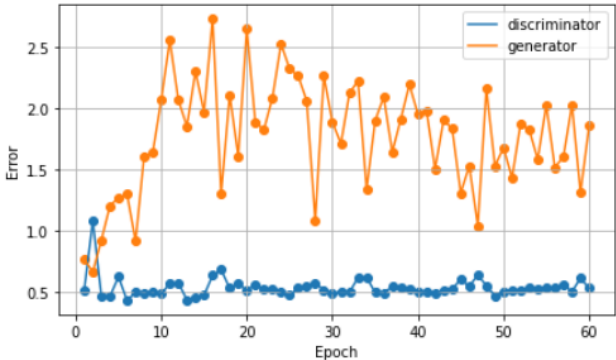
سوال دوم)

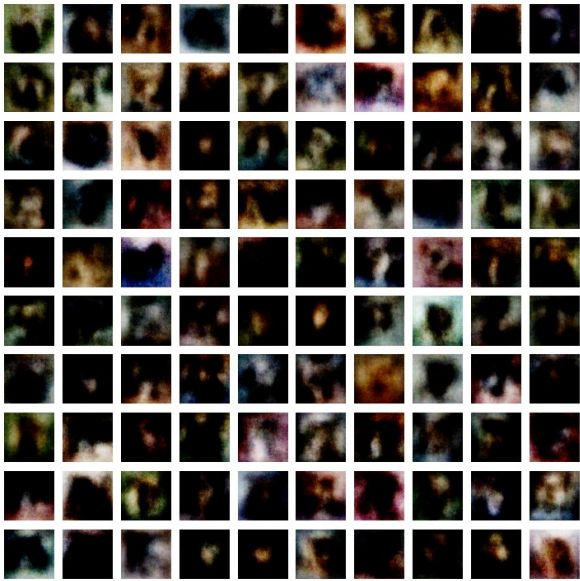
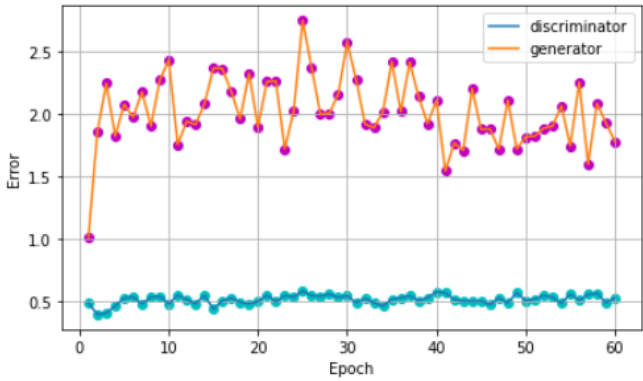
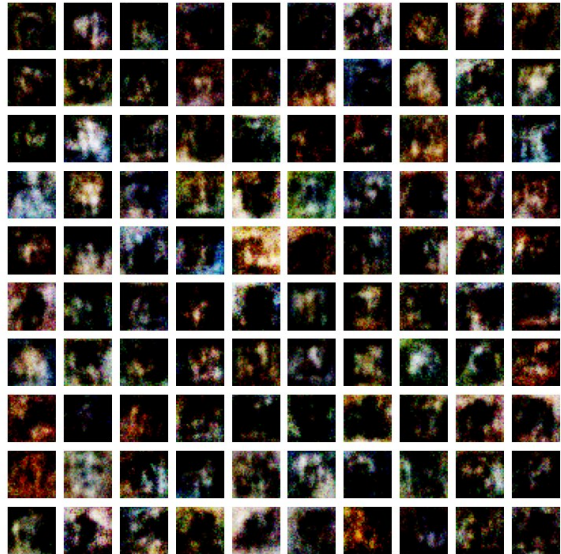
برای شبکه متمایز کننده از تابع هزینه **binary cross entropy** استفاده می‌شود زیرا این شبکه مشابه یک دسته بند باینری عمل می‌کند و تصاویر جعلی و واقعی که دارای برچسب‌های ۰ و ۱ هستند را از هم متمایز می‌کند. برای شبکه مولد نیز از تابع هزینه مشابه **binary cross entropy** استفاده می‌شود چون این شبکه نیز فعالیت مشابه اما متقابلی با شبکه متمایزگر دارد به این شکل که در این شبکه فیدبکی دریافت می‌شود مبنی بر اینکه آیا قادر به تولید یک تصویر جعلی نزدیک با هدف فریب شبکه متمایزگر، بوده است یا خیر؟!

سوال سوم)

برای انجام این بخش در شبکه مولد تعداد مختلفی لایه‌های پنهان قرار دادیم و نتایج در زیر گزارش شده است:

تعداد لایه ها	۴
تعداد نورون ها	۱۰۲۴-۵۱۲-۲۵۶-۱۲۸
نمودار Loss	
تصاویر تولید شده	
تعداد لایه ها	۵
تعداد نورون ها	۲۰۴۷-۱۰۲۴-۵۱۲-۲۵۶-۱۲۸

	<p>نمودار Loss</p>
	<p>تصاویر تولید شده</p>
<p>۶</p>	<p>تعداد لایه ها</p>
<p>۴۰۹۶-۲۰۴۸-۱۰۲۴-۵۱۲-۲۵۶-۱۲۸</p>	<p>تعداد نوروں ها</p>
	<p>نمودار Loss</p>

	تصاویر تولید شده
۷	تعداد لایه ها
۸۱۹۲-۴۰۹۶-۲۰۴۸-۱۰۲۴-۵۱۲-۲۵۶-۱۲۸	تعداد نورون ها
	نمودار LOSS
	تصاویر تولید شده

نتیجه:

همانطور که مشاهده می‌شود با افزایش لایه ها نتایج بهتری حاصل شده اند. تصاویر تولید شده کیفیت بیشتری دارند و واضح تر هستند هم چنین مقدار loss کاهش یافته است.

سوال چهارم)

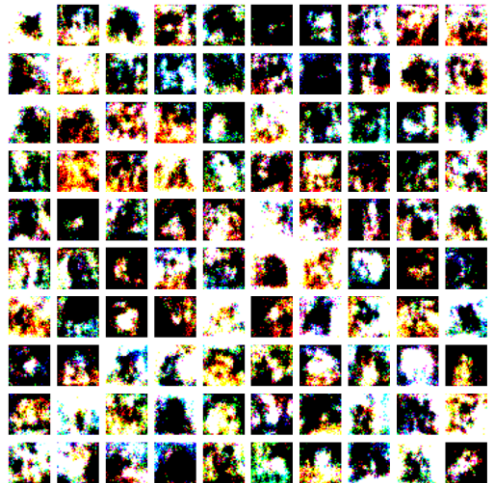
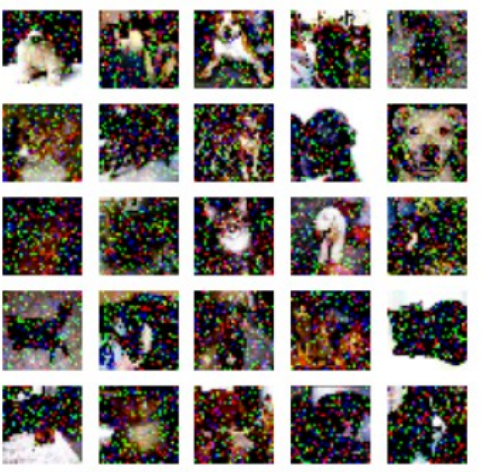
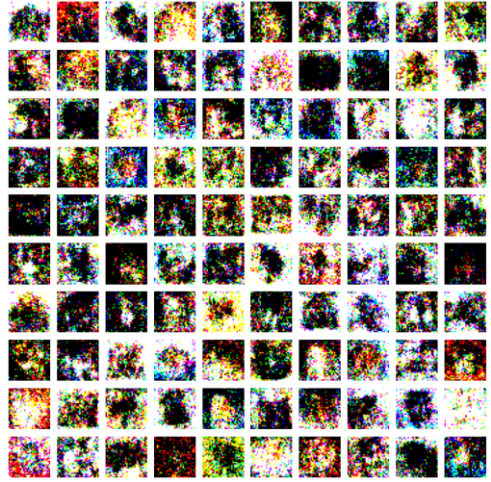
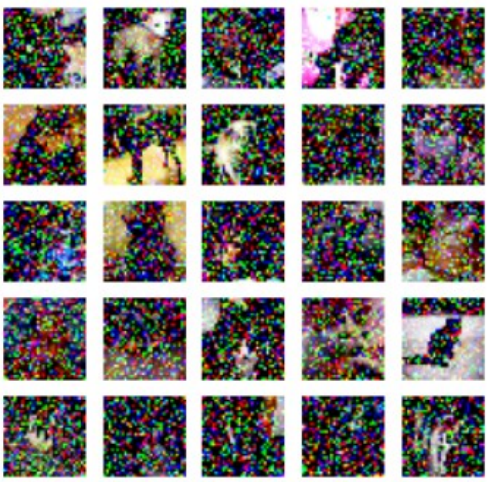
یکی از راههای ایجاد تعادل، اضافه کردن یک نویز گاوسی به تصویر موجود در مجموعه داده است. تعدادی از راه حل های دیگر شامل موارد زیر است:

- استفاده از معماری DCGAN
- نرمال کردن ورودی ها در محدوده -۱ و ۱ و استفاده از تابع \tanh به عنوان تابع فعال سازی
- استفاده از Leaky ReLU در مولد و متمایز کننده
- استفاده از mean pooling و برای نمونه برداری و استفاده از ConvTranspose2D
- اضافه کردن نویز تصادفی به برچسب ها در متمایز کننده
- در صورت داشتن برچسب از آنها استفاده کنید.

سوال پنجم)

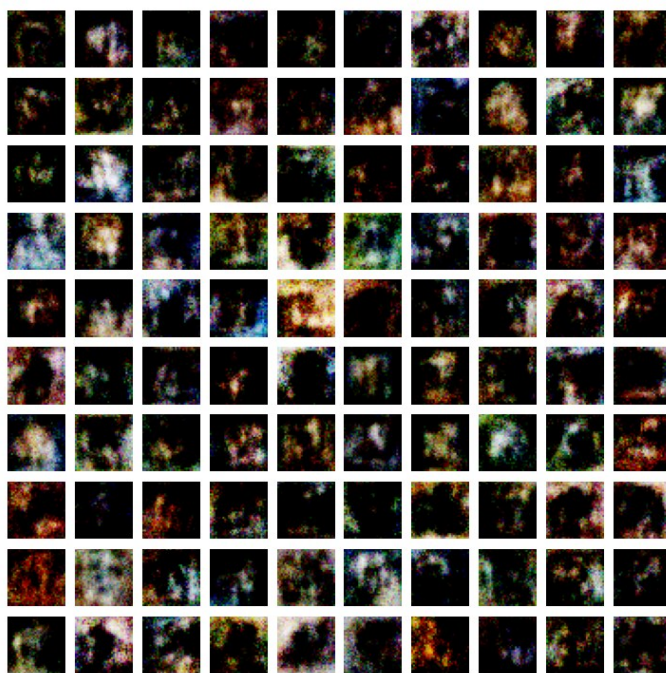
برای انجام این قسمت تصاویر را نویزی می‌کنیم به این صورت که با یک احتمال مشخص هر پیکسل نویزی می‌شود. در جدول زیر مقدار احتمال p و تصاویر نویزی شده و تصاویر تولیدی حاصل نشان داده شده اند. با افزودن مقدار کمی نویز تعادل میزان مولد و تمایزگر تا حد زیادی افزایش پیدا می کند و کیفیت تصاویر تولیدی بهتر می‌شود اما اگر احتمال p را بیشتر از ۰,۲ بدهیم تصاویر بی کیفیتی تولید می‌شوند.

نمونه تصاویر تولید شده با تصاویر نویزی	نمونه تصاویر نویزی	p
		0.01

		0.1
		0.2

سوال ششم)

۱۰۰ نمونه از تصاویر تولید شده:



بخش دوم: DCGAN

آموزش شبکه:

برای آموزش شبکه، ابتدا تعدادی تصاویر جعلی توسط Generator تولید می‌شود سپس ترکیبی از این تصاویر جعلی با برچسب صفر و تصاویر واقعی با برچسب یک به Discriminator داده می‌شود تا این شبکه برای تشخیص تصاویر جعلی از واقعی آموزش ببیند. در مرحله بعد آموزش Discriminator غیر فعال می‌شود و بخش Generator آموزش می‌بیند. به این صورت که یک بردار نویز با توزیع تصادفی نرمال تولید می‌شود و به عنوان ورودی به شبکه داده می‌شود. Generator یک تصویر در خروجی تولید می‌کند و Discriminator که قبلاً آموزش دیده جعلی بودن این تصویر را تشخیص می‌دهد و بدین ترتیب وزن لایه های Generator به گونه ای آپدیت می‌شود که میزان خطا را کاهش دهد و بتواند تصاویر نزدیک به واقعی تولید کند تا حدی که Discriminator نتواند جعلی بودن تصاویر را تشخیص دهد.

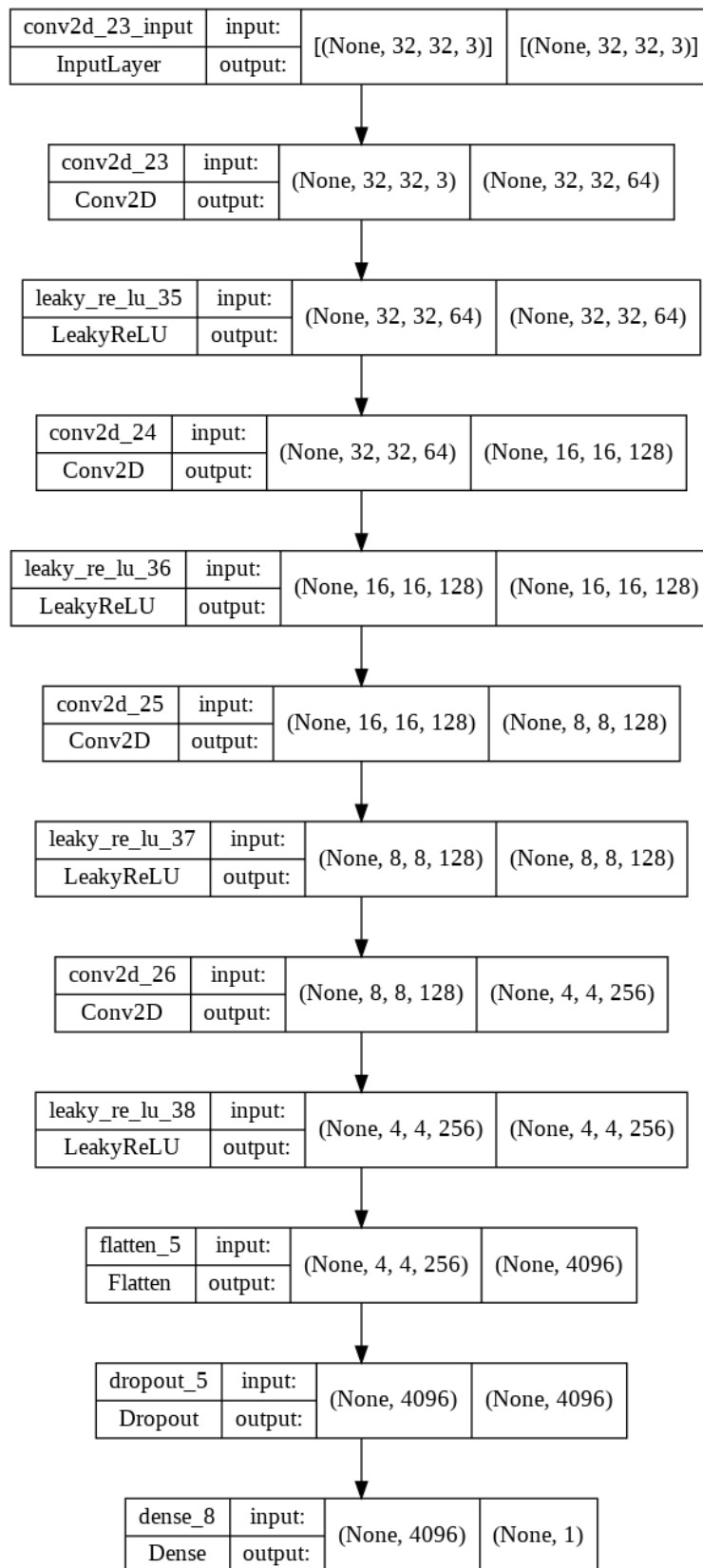
ساختار شبکه DCGAN ایجاد شده به این صورت است:

Discriminator

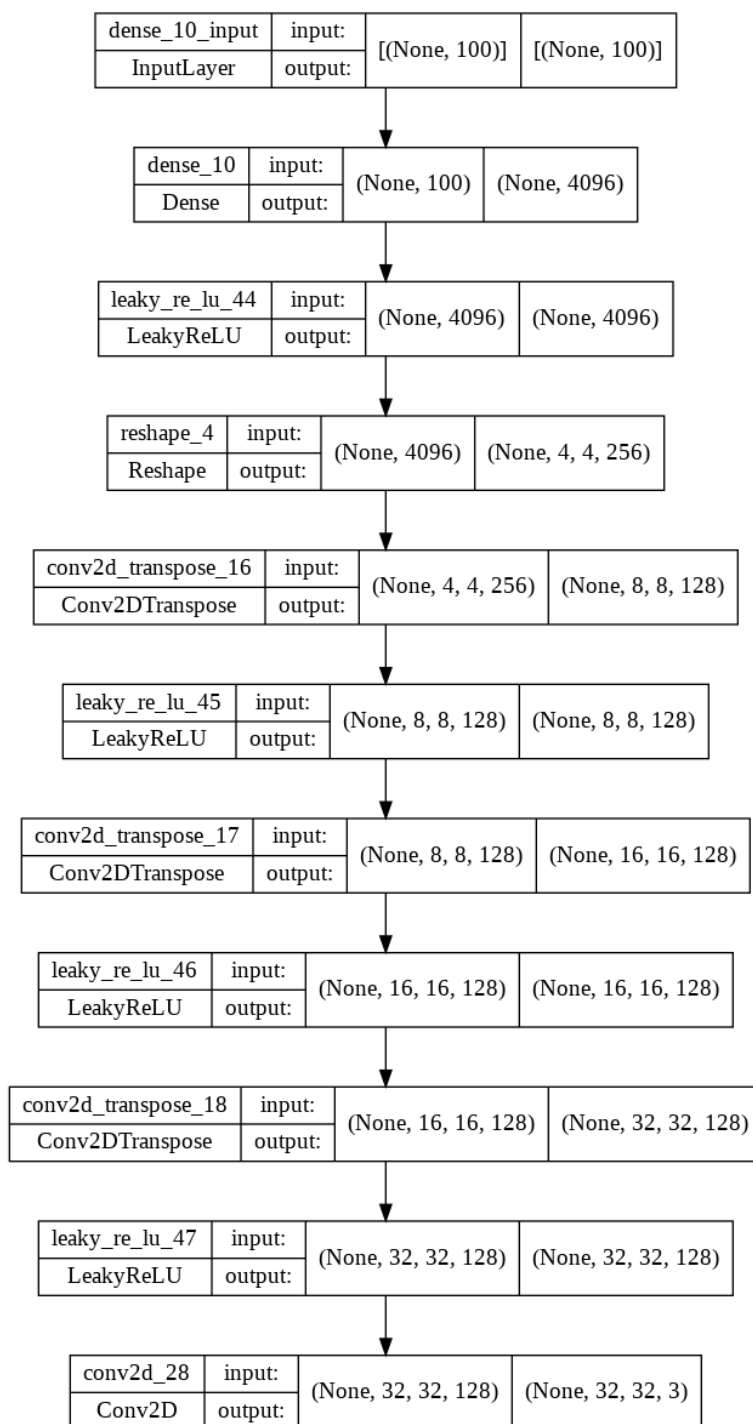
- لایه ورودی: تصاویری با ابعاد 32×32 را دریافت می‌کند
- لایه پنهان: چهار لایه پنهان از نوع Conv2D که به ترتیب $۲۵۶, ۱۲۸, ۱۲۸, ۶۴$ کرنل ۳×۳ دارند و تابع فعالیت آنها LeakyReLU می‌باشد.
- لایه خروجی: لایه Dense با یک نورون با تابع فعالیت sigmoid که با دو مقدار صفر و یک در خروجی، جعلی/واقعی بودن تصویر دریافتی را مشخص می‌کند.

Generator

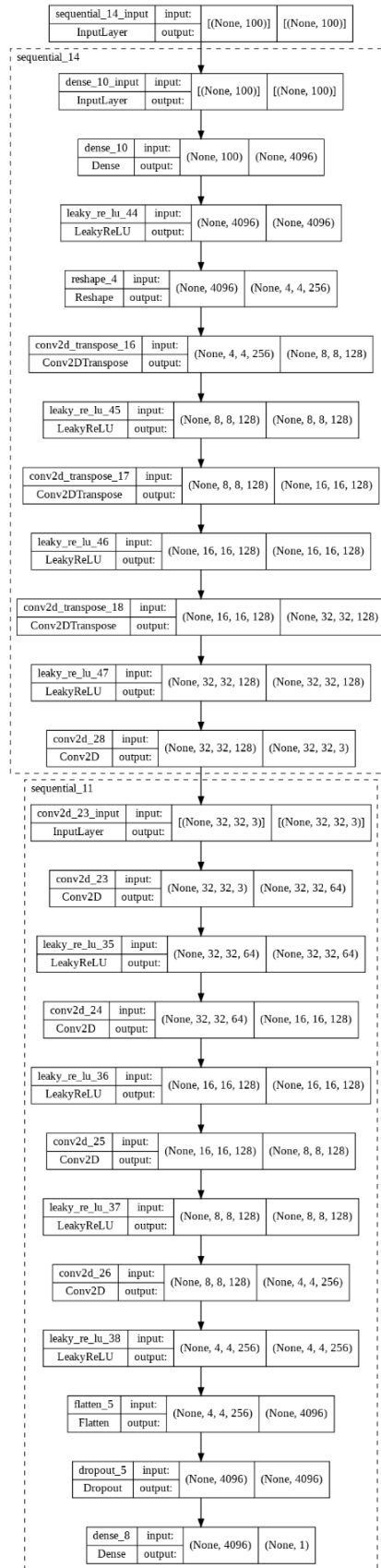
- لایه ورودی: ۱۰۰ نورون (برای دریافت یک بردار نویزی به طول ۱۰۰)
 - لایه پنهان: یک لایه Dense با $256 \times 4 \times 4$ نورون و تابع فعالیت LeakyRelu
 - سه لایه Conv2DTranspose که هر یک ۱۲۸ کرنل 4×4 دارند و تابع فعالیت هر دو LeakyReLU میباشد.
 - لایه خروجی: لایه Conv2D که از کرنلهای 3×3 استفاده میکند و تعداد کرنلها برابر است با تعداد کانالهای تصاویر دیتاست. این لایه تصاویری با ابعاد $32 \times 32 \times 3$ تولید می کند.
- ساختارهای generator و discriminator و مدل کلی در شکل های ۴ و ۵ آورده شده است.



شکل ۴ ساختار Discriminator تولید شده برای DCGAN



شکل ۵ ساختار Generator تولید شده برای DCGAN



شکل ۶ ساختار DCGAN

برای بررسی تاثیر تعداد لایه ها ۳ حالت اجرا شده و نتایج در جدول زیر آورده شده است:

تعداد لایه	تصاویر تولیدشده
۴	
۵	



نتیجه: همانطور که مشاهده می‌شود در شبکه DCGAN با افزایش تعداد لایه های مولد تصاویر کم کیفیت‌تری تولید می‌شوند و دقت مدل پایین می‌آید.

۱۰۰ نمونه از تصاویر ایجاد شده توسط بهترین مدل:

