دانشگاه صنعتی امیرکبیر
(پلی‌تکنیک تهران)
دانشکده مهندسی کامپیوتر

پروژه هفتم درس رایانش عصبی

نگارش
سیدمهدی میرفندرسکی
مدرس
دکتر رضا صفابخش
بهمن ۱۴۰۱

فهرست مطالب

۱ سوالات تشریحی

۱.۱ سوال اول

برای توضیح نقش نویز در نوع شبکه‌ها فرض می‌کنیم که نویزی برای ورودی به بخش مولد این نوع شبکه‌ها وجود نداشت. اتفاقی که خواهد افتاد این است که قسمت مولد خروجی یکسانی برای تمام برچسب‌ها تولید خواهد کرد. درواقع نویز عامل ایجاد کننده خروجی‌های متفاوت به ازای برچسب‌های یکسان خواهد بود. به بیان دیگر این نویز به عنوان منبع تغییرات در نمونه‌های تولید شده عمل می‌کند. با اضافه‌کردن نویز، مولد می‌تواند طیف متنوعی از نمونه‌ها را تولید کند که باعث می‌شود روند آموزش قوی‌تر و مقاوم‌تر شود. همچنین اضافه کردن نویز از mode collapse جلوگیری خواهد کرد که در سوال بعد به آن پرداخته شده است.

تغییر پارامتر یا توزیع نویز به مولد در این نوع شبکه‌ها می‌تواند تأثیر قابل توجهی بر کیفیت و تنوع نمونه‌های تولید شده و همچنین بر پایداری فرآیند آموزش بگذارد. هرچه نویز بسته به پارامتر و توزیع تصادفی‌تر باشند، می‌تواند منجر به تنوع بیشتر در نمونه‌های تولید شده شود. زیرا مولد تشویق می‌شود تا مناطق مختلف فضای ورودی را کاوش کند. و این اتفاق، جلوگیری از mode collapse در فضای ورودی را به دنبال دارد. با این حال، تصادفی‌تر بودن نویز می‌تواند یادگیری ساختار داده‌ها را برای مولد دشوارتر کند و در نتیجه نمونه‌های با کیفیت پایین‌تری تولید شوند. درواقع پیداکردن پارامتر و توزیع مناسب نویز یک نوع trade-off میان تشویق مولد برای کاوش در فضای ورودی و یادگیری ساختار داده‌ها در مولد خواهد بود.

۲.۱ سوال دوم

این دو نوع مسئله دو مشکل رایجی هستند که می‌توانند هنگام آموزش شبکه‌های مولد تقابلی رخ دهند.

mode collapse مشکلی است که در آن مولد به جای تولید طیف متنوعی از نمونه‌ها، تغییرات محدودی از یک نمونه تولید می‌کند. این زمانی اتفاق خواهد افتاد که مولد در فریب دادن تمایزگر بیش از حد خوب شود و گرادیان‌ها از تمایزگر به مولد بسیار کوچک شود و باعث شود که مولد در فضای ورودی به یک نقطه (زیرمجموعه ای از داده‌های آموزشی) همگرا شود. سپس باعث می‌شود که مولد قادر به تولید نمونه‌های جدید و متفاوت از داده‌های آموزشی نباشد.

diminishing gradients مشکل دیگری است که می‌تواند در طول آموزش رخ دهد. این مشکل زمانی اتفاق می‌افتد که گرادیان‌ها از تمایزگر به مولد بسیار کوچک شده و باعث می‌شوند که مولد به کمینه محلی همگرا شود. این مشکل ناشی از رسیدن مولد و تمایزگر به تعادلی است که در آن مولد قادر به تولید نمونههایی است که از نمونه‌های واقعی قابل تشخیص نیستند، اما گرادیان‌های تشخیص دهنده برای ادامه بهبود مولد بسیار کوچک خواهد بود.

برای جلوگیری از این مشکلات، چندین تکنیک وجود دارد که می‌توان از آنها استفاده کرد:

- منظم‌سازی: تکنیک‌هایی مانند Drop out و کاهش وزن را می‌توان برای جلوگیری از بیشبرازش مولد و تمایزگر به داده‌های آموزشی استفاده کرد.
- هموارسازی یک طرفه برچسب‌ها: اگر خروجی تمایزگر را برای اطمینان کمتر در پیش بینی‌های خود به صورت مقدار پیوسته‌ای بین ۰ و ۱ به‌جای خود اعداد ۰ و ۱ انتخاب کنیم، می‌توانیم به بهبود گرادیان‌ها از تمایزگر به مولد کمک کند
- استفاده از نویز: افزودن نویز تصادفی‌تر به ورودی مولد می‌تواند با تشویق مولد به کاوش در مناطق مختلف فضای ورودی به جلوگیری از mode collapse کمک کند.
- برنامه آموزشی: استفاده از یک زمانبندی آموزشی برای تمایزگر و مولد به صورت متناوب با افزایش دوره‌ای نرخ یادگیری به جلوگیری از diminishing gradients کمک می‌کند.

- تغییرات معماری: استفاده از معماری‌هایی که بهینه‌سازی مولد و تمایزگر را سخت‌تر (استفاده از لایه‌های عادی) می‌کند، می‌تواند به بهبود پایداری فرآیند آموزش کمک کند.

ترکیبی از این تکنیکها می‌تواند برای بهبود عملکرد این نوع شبکه‌ها مورد استفاده قرار گیرد.

۳.۱ سوال سوم

لایه معکوس کانولوشن عملکردی مخالف لایه کانولوشنی دارد. به بیان دیگر با ورود یک feature map به این نوع لایه، یک تصویر ساخته می‌شود. می‌توان گفت لایه معلکوس کانوولوشنی اطلاعات خلاصه شده چند پیکسل (خلاصه شده توسط لایه کانوولوشنی) را توسعه و بسط می‌دهد. این نوع لایه‌ها با استفاده از فرمولی برای تابع گسترش نقطه، تخمین بهبود یافته‌ای از تصویر ایجاد ارائه می‌کنند.

معماری یک DCGAN از دو جزء اصلی مولد و تمایزگر تشکیل می‌شود. شبکه مولد یک بردار نویز تصادفی را به عنوان ورودی می‌گیرد و با عبور دادن آن از یک سری لایه معکوس کانولوشن، که برای افزایش وضوح فضایی ورودی طراحی شده‌اند، تصویر جدیدی تولید می‌کند. خروجی مولد یک تصویر مصنوعی است که از یک تصویر واقعی قابل تشخیص نیست. شبکه تمایزگر هم تصاویر واقعی و هم تصاویر مصنوعی تولید شده توسط مولد را به عنوان ورودی می‌گیرد. سپس از یک سری لایه‌های کانولوشنی برای تعیین واقعی یا مصنوعی بودن هر تصویر استفاده می‌کند. خروجی تمایزگر یک مقدار اسکالر بین ۰ و ۱ است که ۰ نشان دهنده یک تصویر مصنوعی و ۱ نشان‌دهنده یک تصویر واقعی است. این دو شبکه با هم در یک فرآیند رقابتی آموزش می‌بینند، به طوری که مولد سعی می‌کند تصاویری را تولید کند که تمایزگر آنها را به عنوان واقعی طبقه بندی کند، و تمایزگر سعی می‌کند به درستی تصاویر تولید شده را به عنوان مصنوعی طبقه بندی کند. با پیشرفت آموزش، تولید کننده در تولید تصاویر واقعی بهتر می‌شود و تمایزگر در شناسایی تصاویر مصنوعی بهتر می شود.

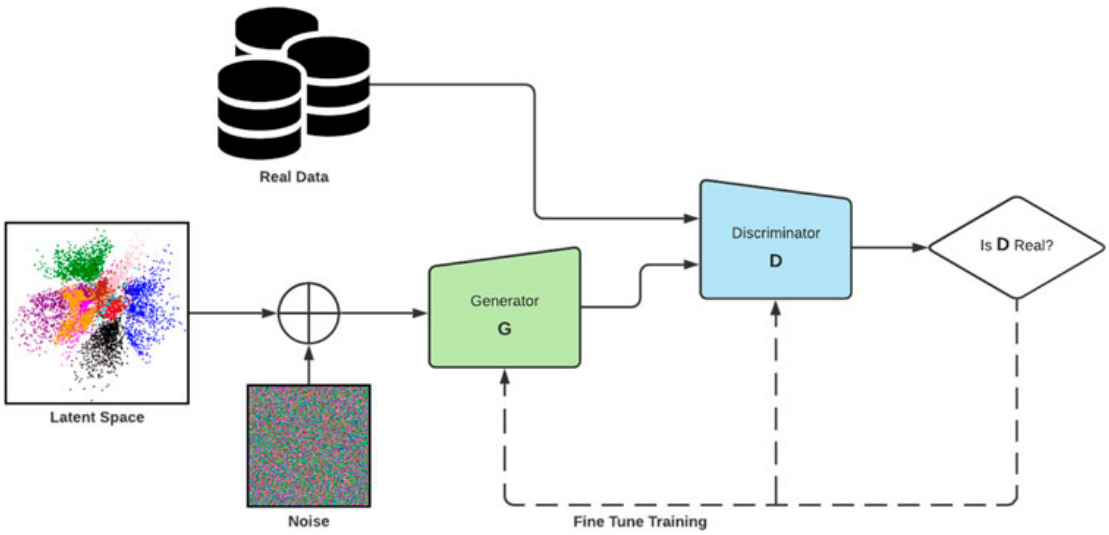
ایده اصلی تولید تصویر از متن این است که مولد و تمایزگر را در یک GAN با کدگذاری متنی مناسب توضیحات تقویت کنیم. از نظر مفهومی، این شبیه به شرطی کردن عملکرد مولد و تمایزگرها بر روی توضیحات متن است. کار اصلی پیاده‌سازی را با استفاده از شبکه‌های عصبی کانولوشنی عمیق توصیف می‌کند و از این رو DCGAN نامیده می‌شود. مولد یک شبکه لایه‌های معکوس کانولوشنی است که تصویری را از متن بر اساس توزیع نویز تولید می‌کند. تمایزگر یک شبکه کانولوشنی است که با توجه به کدگذاری متن، احتمال تعلق تصویر ورودی به توزیع داده اصلی را خروجی می‌دهد. البته جزئیات معماری در لینک موجود است (تعداد لایه‌ها، توابع فعالیت و ...). اما نکته بدیهی که لازم به ذکر است که آموزش تمایزگر بر اساس کپشن تصاویر اتفاق خواهد افتاد. یعنی باید داده‌های آموزشی با متن موجود باشند.

۴.۱ سوال چهارم

شبکه‌های مولد تقابلی از دو جزء اصلی تشکیل شده‌اند: یک شبکه مولد و یک شبکه تمایزکر. روند آموزش این دو شبکه متفاوت و مکمل یکدیگر است. معماری این نوع شبکه‌ها در شکل زیر آورده شده است.

همانطور که در شکل مشاهده می‌شود، شبکه مولد برای تولید نمونه‌های جدید که مشابه داده‌های آموزشی هستند آموزش داده می‌شود. این کار با به حداقل رساندن تفاوت بین نمونه‌های تولید شده و نمونه‌های واقعی انجام می‌شود (با استفاده از یک الگوریتم بهینه‌سازی مانند GD برای به حداقل رساندن یک تابع هزینه). تابع هزینه مورد استفاده در GAN معمولاً cross-entropy خواهد بود.

از طرفی شبکه تمایزگر برای تشخیص تمایز بین نمونه‌های واقعی و تولیدشده آموزش داده می‌شود. این کار با به حداکثر رساندن تفاوت بین احتمال یک نمونه واقعی و احتمال یک نمونه تولیدشده جدید انجام می‌شود. هر دو شبکه به طور همزمان آموزش می‌بینند، به طوری که مولد در تلاش برای تولید نمونه‌هایی است که می‌تواند تمایزگر را فریب دهد و تمایزگر تلاش می‌کند تا نمونه‌های تولید شده را به درستی شناسایی کند. به خاطر همین فرایند یک فرایند رقابتی خواهد بود.



شکل ۱: معماری شبکه‌های مولد تقابلی

تابع خطا در مدل GAN بدین صورت تعریف می‌شود که هر چه تصاویر تولید شده توسط مولد واقعی‌تر باشد و تمایزگر را به خطا بیاندازد به مولد هزینه کمتری داده میشود و هرچه تمایزگر بهتر بتواند تصاویر را تشخیص دهد هزینه کمتری برای آن در نظر گرفته می‌شود. تابع هزینه به شرح زیر است:

$$\min_G \max_D E_{x\sim p_{data}(x)}[\log D(x)] + E_{z\sim p_z(z)}[۱ - \log D(G(z))]$$

۲ CGAN

۱.۲ سوال اول

در CGAN، برای ادغام یک برچسب در یک تصویر، باید هم برچسب و هم یک بردار نویز را به عنوان ورودی به مولد ارسال کرد. سپس مولد از این اطلاعات برای تولید تصویری که بر روی برچسب شرطی شده است استفاده می‌کند. برچسب را می‌توان قبل از ارسال آن به مولد به بردار نویز متصل کرد یا می‌توان آن را به عنوان ورودی جداگانه به مولد ارسال کرد. سپس مولد از اطلاعات برچسب برای کنترل ویژگی‌های تصویر تولید شده استفاده می‌کند. از طرفی در تمایزگر نیز به همین دو روش می‌توان برای آموزش برچسب را با ورودی ادغام کرد.

۲.۲ سوال دوم

در یک GAN استاندارد، تمایزگر معمولاً یک مقدار اسکالر واحد را خروجی می‌دهد که احتمال اینکه ورودی یک نمونه واقعی (تولید نشده) باشد را نشان می‌دهد. در یک CGAN، تمایزگر یک مقدار اسکالر واحد را نیز خروجی می‌دهد که احتمال اینکه ورودی یک نمونه واقعی باشد را نشان می‌دهد. با این حال، در یک CGAN، تمایزگر همچنین یک برچسب یا سایر اطلاعات شرطی را به عنوان ورودی می‌گیرد و از این اطلاعات برای طبقه‌بندی خود استفاده می‌کند. بنابراین، خروجی تمایزگر در CGAN یک مقدار اسکالر است که نشان دهنده احتمال این است که ورودی چقدر واقعی است، پس برچسب ورودی در آن جایی نخواهد داشت.

۳.۲ سوال سوم

ابتدا لازم به ذکر است که، برای پیاده‌سازی این قسمت، با توجه به سوال نیاز به سعی و خطا نبود و چند مقاله پیاده‌سازی شده بررسی شد. راه‌های مختلفی برای این مسئله پیشنهاد شده‌اند (استفاده از لایه کانولوشنی و یا لایه Dense ساده). با پیاده‌سازی فعلی بر اساس لایه‌های Dense ساده نتیجه مطلوبی گرفته شد.

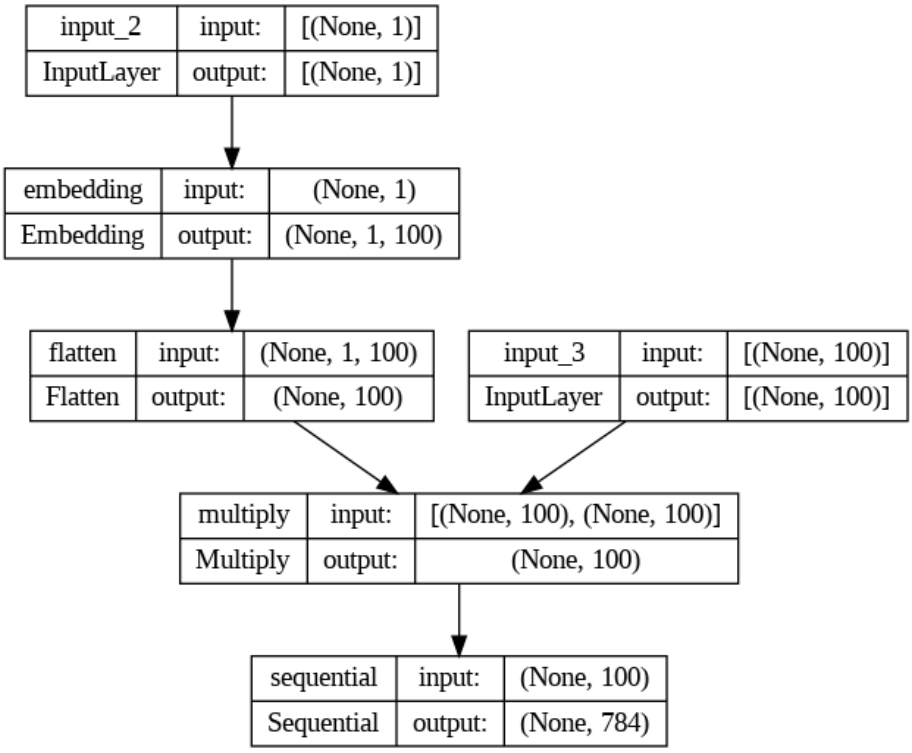
روال کار بدین صورت است که ابتدا تنها داده‌ها (بدون برچسب) را به بازه −۱ و ۱ انتقال می‌دهیم (تقسیم بر ۲۵۵، منهای ۱ و ضربدر ۲). سپس سه مرحله اساسی ساخت مدل‌های تمایزگر، مولد و آموزش ترکیبی این دو را خواهیم داشت، که در ادامه اقدام به توضیح هر یک می‌پردازیم:

- مدل تمایزگر یک تصویر به همراه برچسب آن از مجموعه داده می‌گیرد و تشخیص می‌دهد که آیا یک تصویر واقعی یا ساختگی است. اما برای ورودی مدل باید ابتدا بتوان آن‌ها را به نحوی ترکیب کرده و به مدل دهیم. ابتدا برچسب کلاس از یک لایه Embedding با خروجی ۷۸۴ عبور داده می‌شود (با اندازه حداکثر ۱۰). این کار به دلیل آن است که برای ادغام با تصاویر در نهایت در با تصاویر ضرب خواهند شد. سپس سه لایه Dense قرار می‌گیرد. برای هر لایه نیز مانند اکثر پیاده‌سازی‌ها از تابع فعالیت LeakyReLU استفاده شد. درنهایت برای تولید مقدار احتمال بین ۰ و ۱ برای واقعی یا جعلی بودن تصاویر از یک نورون به همراه تابع فعالیت sigmoid استفاده شد.

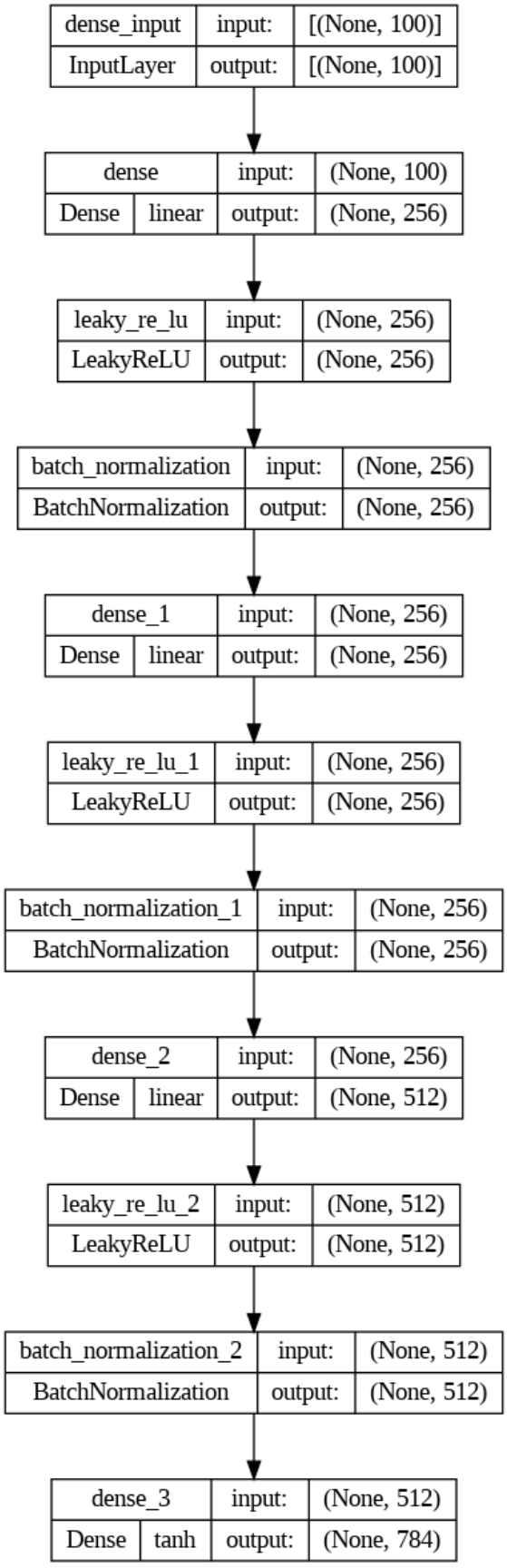
- مدل مولد یک مقدار از فضای پنهان (latent space) را به عنوان ورودی می‌گیرد و یک تصویر را خروجی می‌دهد. مشابه مدل تمایزگر این فضای پنهان با برچسب تصویر خواسته شده ادغام می‌شود. روال کار به همان صورت خواهد بود. اما بعد از ادغام فضای پنهان (معمولا اندازه آن را ۱۰۰ درنظر می‌گیرند) یک فضای برداری از مقادیر تصادفی (توزیع گاوسی) تشکیل شده است که با برچسب نیز در آن نهفته شده است. درنهایت پس از آماده‌سازی ورودی سه لایه Dense قرار گرفته‌اند که بعد از هر سه لایه یک لایه BatchNormalization با تکانه ۰.۸ قرار گرفته است. این کار برای آموزش بهتر شبکه انجام می‌شود (بدون لایه‌های ذکر شده تست شد و نتیجه خوبی گرفته نشد.). درنهایت نیز برای خروجی یک لایه به اندازه ۷۸۴ در نظر گرفته شد.

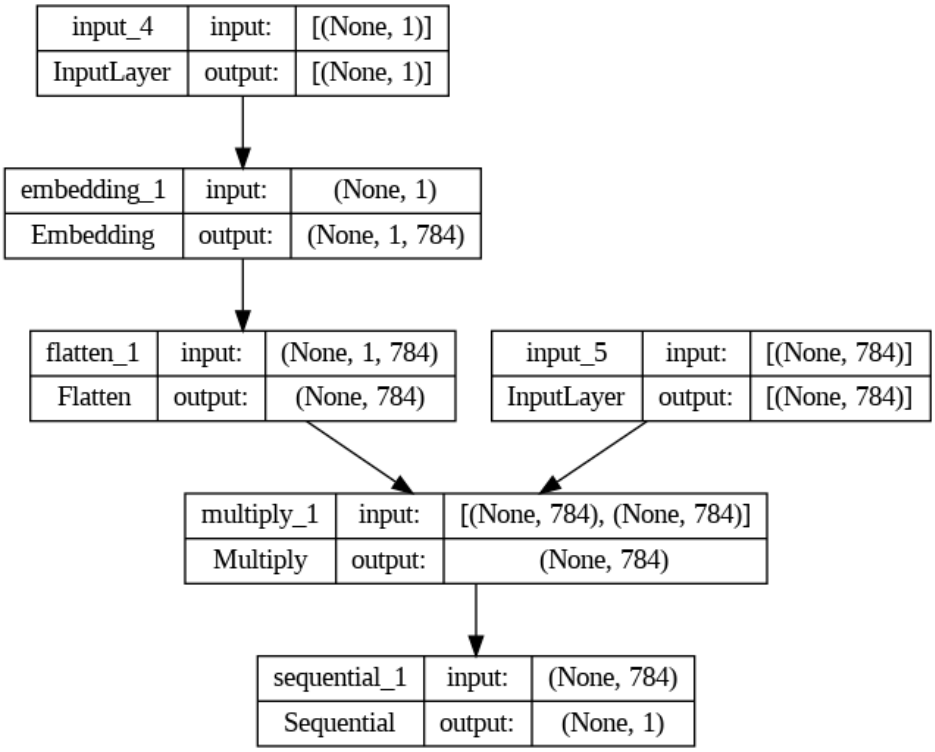
- یک مدل نهایی می‌تواند یک مدل مولد با ورودی‌های مدل تمایزگر و داده‌های آموزشی به همراه برچسب است. برای ساخت این مدل ابتدا مدل فوق را ساخته و به صورت نوبتی تمایزگر و مولد را آموزش می‌دهیم. برای اینکار به صورت دستی ای‌پاک‌ها و دسته‌ها را پیاده‌سازی می‌کنیم. همچنین باید در هر حلقه ابتدا تمایزگر را یکبار برای داده‌های آموزشی و یکبار برای داده‌های جعلی آموزش داده و سپس وزن‌های آن را ثابت نگه داریم و آموزش مولد را شروع کنیم. همچنین برای بهبود آموزش از تکنیک هموارسازی برچسب‌ها (سوال ۲) با پارامتر ۰.۱ استفاده شد. با توجه به ساختار درنظر گرفته شده نمودار خطای هریک (مولد و تمایزگر) در ادامه مشخص شده است (نمودار خطای کل مدل همان خطای مولد درنظر گرفته شده است.). آموزش با ۲۰ ای‌پاک انجام شد که تصاویر آن به همراه gif مربوطه ضمیمه شده است.

همانطور که در نمودار خطای آموزش مشاهده می‌شود، مدل پس از چند ای‌پاک به یک محدوده‌ای همگرا می‌شود. در واقع در این محدوده به طور پیوسته مولد و تمایزگر دچار نوسان یا همان رقابت هستند تا بر دیگری غالب شوند. این یکی از معماری از یکی از پیاده‌سازی‌های موجود الهام گرفته است. برای دستیابی به خطای کمتر باید معماری یا تکنیک‌های دیگری بکار برد.

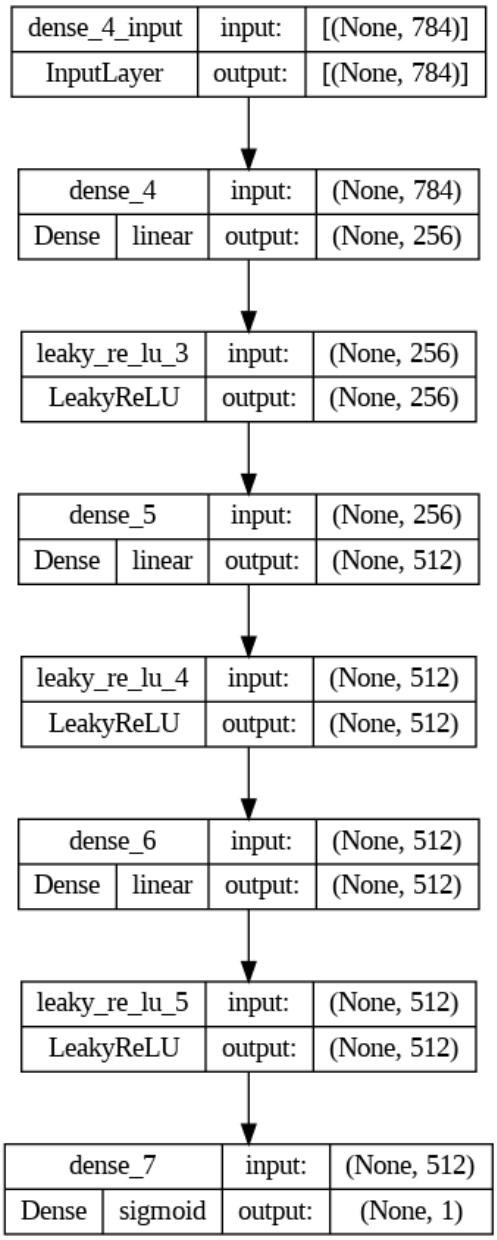


شکل ۲: گراف مصور مولد


dense_input | input: | [(None, 100)]
InputLayer | output: | [(None, 100)]
dense | input: | (None, 100)
Dense | linear | output: | (None, 256)
leaky_re_lu | input: | (None, 256)
LeakyReLU | output: | (None, 256)
batch_normalization | input: | (None, 256)
BatchNormalization | output: | (None, 256)
dense_1 | input: | (None, 256)
Dense | linear | output: | (None, 256)
leaky_re_lu_1 | input: | (None, 256)
LeakyReLU | output: | (None, 256)
batch_normalization_1 | input: | (None, 256)
BatchNormalization | output: | (None, 256)
dense_2 | input: | (None, 256)
Dense | linear | output: | (None, 512)
leaky_re_lu_2 | input: | (None, 512)
LeakyReLU | output: | (None, 512)
batch_normalization_2 | input: | (None, 512)
BatchNormalization | output: | (None, 512)
dense_3 | input: | (None, 512)
Dense | tanh | output: | (None, 784)

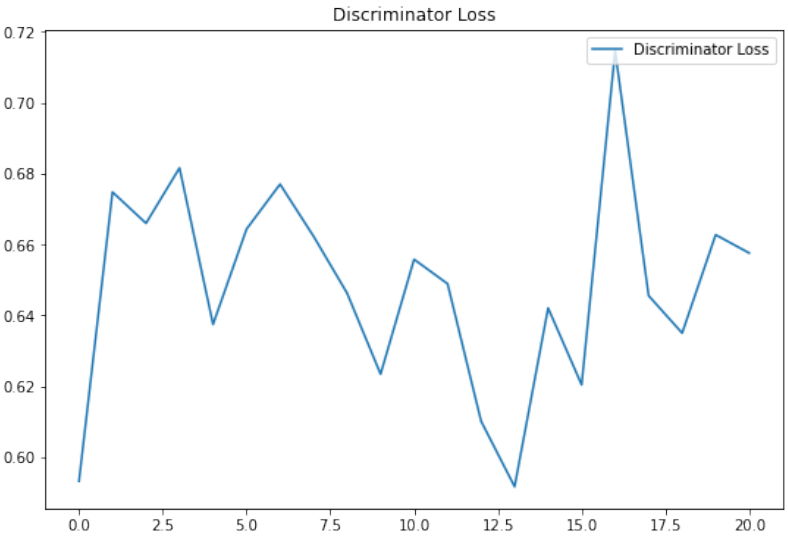


شکل ۴: گراف مصور تمایزکر



شکل ۵: گراف مصور تمایزکر

شکل ۶: نمودار خطا



شکل ۷: نمودار خطا