
بسم الله الرحمن الرحيم

دانشگاه صنعتی اصفهان

دانشکده برق و کامپیوتر

گزارش پروژه کارشناسی با موضوع:

ساخت داده مصنوعی با شبکه عصبی GAN

استاد پروژه:

دکتر سید محمد علی خسروی فرد

گردآورنده:

پارسا حقیقی نایینی

بهمن ماه ۱۳۹۹

فهرست

چکیده

در دهه‌های اخیر رشد سریع تولید داده و توسعه سخت‌افزارهای محاسباتی باعث پیشرفت روزافزون هوش مصنوعی شده است و انتظار می‌رود در سال‌های نه چندان دور تغییرات گسترده‌تری در زندگی انسان‌ها به‌وجود بیاورند.

یکی از چالش‌ها و شاید فانتزی‌هایی که از زمان تولد هوش مصنوعی وجود داشته، توانایی ماشین در درک و تشخیص دنیای اطراف و مهمتر از آن تولید داده‌هایی مشابه داده‌های واقعی است. الگوریتم‌های رایج یادگیری ماشین تنها قادر بودند اشیاء و داده‌های متفاوت را تمیز دهند ولی کماکان مسئله درک و شناخت دست نخورده باقی مانده بود.

در سالیان اخیر دسته‌ای از مدل‌ها به نام مدل‌های مولد توسعه یافته‌اند که علاوه بر درک از محیط می‌توانند داده‌های جدیدی نیز تولید کنند. یکی از جدیدترین و کارآمدترین این مدل‌ها، شبکه‌های متخاصم مولد یا به اختصار شبکه‌های گن هستند که نتایج آن‌ها بسیار قابل قبول و واقعی‌ای بوده است به طوری که گاهی حتی با چشم انسان مصنوعی بودن آن‌ها قابل تشخیص نیست.

در این پروژه، هدف شناخت بیشتر در مورد شبکه‌های گن، مزیت‌ها و چالش‌ها و کاربردهای آن است. در ادامه معماری‌های مختلف خانواده گن که در طی این سال‌ها معرفی شده‌اند و توانسته‌اند چالش‌های مدل پایه را برطرف کنند و خروجی‌های بسیار بهتر و با وضوح بیشتری تولید کنند توضیح داده شده‌اند. در انتهای گزارش نیز در مورد چالش‌ها و موانع شبیه‌سازی و راه‌هایی که برای فائق آمدن بر آن‌ها در طی انجام پروژه انجام گرفته توضیحاتی ارائه شده است.

فصل اول

معرفی مقدمات و ملزومات پروژه

۱-۱-مقدمه

در این فصل مقدمات و پیش نیازهای معرفی پروژه توضیح داده شده‌اند. تا حد امکان سعی شده از ابتدایی‌ترین مفاهیم که در بخش‌های بعد مورد نیاز هستند به ساده‌ترین و کوتاه‌ترین صورت توضیح داده شوند. از جمله در این فصل در مورد مجموعه داده، هوش مصنوعی، یادگیری ماشین، یادگیری عمیق و مدل‌های تفکیک‌کننده و مدل‌های مولد توضیحاتی ارائه شده است.

در پایان این فصل، در مورد تعدادی از مدل‌های مولد از جمله شبکه اتوانکدر ، اتوانکدر متنوع، شبکه‌های متخاصم مولد (گن)، شبکه خود رگرسیون، مدل‌های جریان و مدل‌های ترکیبی توضیحاتی ارائه شده است.

۱-۲- داده

هر عامل هوشـــمند من جمله انســـان از طریق داده‌های ورودی که از محیط دریافت می‌کنند آموزش می‌بیند. یکی از چالش‌هایی که از ابتدای شکل‌گیری علوم مرتبط با داده و به طور خاص در یادگیری ماشین وجود داشته است یافتن مجموعه داده متنوع، با کیفیت و با تعداد مناسب برای استفاده در فرآیند آموزش است.

تنوع داده‌های آموزش اهمیت زیادی دارد چرا که ماشـــینی که روی تعداد دســـته محدودی آموزش دیده اســـت دچار سـوگیری خواهد شـد و نمی‌تواند کارایی مناسـبی داشـته باشـد. هر نمونه داده متشـکل اسـت از یک بردار ویژگی و در برخی موارد برچسب.

۱-۲-۱- بردار ویژگی[1]

بردار ویژگی شامل چندین درایه است که هر کدام از درایه‌ها نماینده ویژگی مشخصی در بین داده‌ها است. برای مثال در مجموعه داده قیمت ملک‌های یک شـهر، مسـاحت، تعداد اتاق و مکان جغرافیایی ملک از جمله ویژگی‌های قابل تصـور برای این مجموعه داده هستند.

در داده‌های تصویری هر پیکسل نماینده یک ویژگی است. در تصاویر رنگی که هر پیکسل دارای سه کانال رنگی است، هر کانال رنگی هر پیکسل نماینده یک ویژگی هستند

۱-۲-۲- برچسب[2]

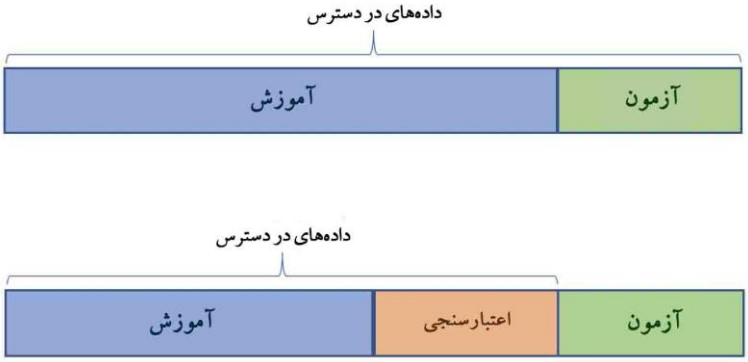
در مجموعه داده‌های برچســـب‌دار، برچســـب هر داده مشـــخص می‌کند آن نمونه به چه دســـته‌ای تعلق دارد. در یادگیری نظارت نشده که در بخش بعدی توضیح داده شده است، مطرح شده که در این نوع یادگیری نیازی به داده برچسب‌دار نیست و یادگیری تنها بر اساس بردار ویژگی انجام‌پذیر است.

برای مثال، برچسـب داده‌های مربوط به ملک‌های یک شـهر، قیمت آن‌هاسـت. یا برچسـب تصـویر یک چهره، جنسـیت یا بازه سنی است.

[1] feature vector
[2] Label

۱-۲-۳- داده آموزش[۳]، داده آزمون[۴] و داده اعتبارسنجی[۵]

در فرآیند یادگیری یک ماشین، باید توجه داشت که به هیچ عنوان نباید با همان داده‌هایی که آموزش داده‌ایم، مدل را آزمایش و ارزیابی کنیم. با این کار ماشین کارآمدی خود را برای نمونه‌های جدید از دست می‌دهد چرا که تنها توانسته ویژگی‌های نمونه‌های خود را مدل کند. برای جلوگیری از این مشکل، از ابتدا مجموعه داده خود را به دو قسمت تقسیم می‌کنیم. بخش عمده نمونه‌ها را صرف آموزش و بقیه را برای آزمون مدل استفاده می‌کنیم.



شکل ۱-۱ : تقسیم مجموعه داده‌های به داده‌های آموزش، داده‌های آزمون و اعتبارسنجی

پس از چند مرحله ارزیابی با داده‌های آزمون، این نمونه‌ها نیز برای ماشین تکراری می‌شوند. برای اطمینان بیشتر از ارزیابی صحیح و کامل می‌توان یک قسمت دیگر از داده‌ها را نیز جدا کرد و از آن برای ارزیابی مدل استفاده کرد و از داده‌های آزمون تنها برای سنجش میزان دقت ماشین استفاده کرد.

۱-۳- یادگیری ماشین[۶]

یادگیری ماشین شاخه‌ای از علم هوش مصنوعی است که به دنبال ایجاد هوشمندی در ماشین با استفاده از استخراج دانش از داده‌ها است.

یادگیری ماشین به چند طریق امکان پذیر است:

[3] training data
[4] test data
[5] validation data
[6] Machine Learning

- **یادگیری نظارت شده:** یادگیری و آموزش با داده‌های برچسب‌دار انجام می‌شود. در واقع دسته‌های مختلف به ماشین آموزش داده می‌شود و وظیفه ماشین دسته بندی داده‌های جدید در یکی از دسته‌های از پیش تعیین شده است.
- **یادگیری نظارت نشده:** آموزش ماشین با داده‌های بدون برچسب انجام می‌شود. پرهزینه بودن و زمان‌بر بودن برچسب‌گذاری داده‌ها، متخصصان علوم داده را به استفاده از این روش یادگیری مجاب کرده است.
- **یادگیری نیمه نظارتی:** در این روش یادگیری، درصد کمی از داده‌ها برچسب‌دار هستند و عمده آن‌ها بدون برچسب‌اند. ثابت شده است که همین میزان کم داده برچسب‌دار، باعث نتایج بسیار بهتری از یادگیری نظارت نشده بدست شده است. این موضوع باعث توجه هر چه بیشتر متخصصان در سال‌های اخیر به یادگیری نیمه نظارتی شده است.
- **یادگیری تقویتی:** در مسائلی که پیش از مدل‌سازی اطلاع کاملی از محیط وجود ندارد از یادگیری تقویتی استفاده می‌شود. در این مدل تقریبا آموزش اولیه‌ای به ماشین داده نمی‌شود و پس از هر فرآیند بر اساس اینکه تا چه حد کارکرد عامل نزدیک به دنیای واقعی بوده جریمه یا تشویق می‌شود.

۱-٤- شبکه‌های عصبی مصنوعی[۷]

با توسعه ماشین‌ها و نیاز به محاسبات، متخصصین رایانه و علوم محاسباتی به دنبال جستجوی راه‌حل‌های بهتری گشتند. همچون اکثر مصنوعات بشری برای پاسخگویی به این نیاز، از خلقت و آفرینش شبکه عصبی هوشمندترین عامل در دنیای شناخته شده ما یعنی انسان‌ها الهام گرفته شد و نام شبکه عصبی مصنوعی برای این مدل جدید انتخاب شد.

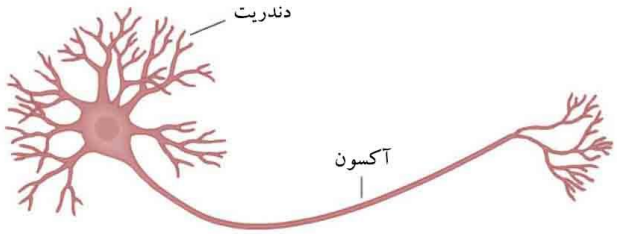
ویژگی‌ای که شبکه عصبی را از دیگر مدل‌های ریاضی و محاسباتی که امکان پیاده‌سازی نرم‌افزاری دارند متمایز می‌کند توانایی یادگیری آن است. در بخش‌های بعد این موضوع توضیح داده شده است.

[7] Artificial Neural Networks

۱-٤-۱- تاریخچه و معرفی شبکه‌های عصبی

مدل پایه شبکه‌های عصبی مصنوعی در اوایل دهه ۱۹۴۰ میلادی توسط دو ریاضی‌دان و با الهام از ساختار شبکه عصبی انسان پیاده سازی شد.

اعصاب انسان که از مغز به سرتاسر بدن رفته، شبکه‌ای است از نورون‌ها که پیام‌های الکتریکی را از مغز به اعصاب و از اعصاب به مغز مخابره می‌کنند. هر نورون از طریق گیرنده‌های خود که دندریت نامیده می‌شود، پیام‌های الکتریکی را دریافت می‌کند و پس از جمع آن‌ها در هسته خود نتیجه را از طریق آکسون به نورون‌های لایه بعدی تحویل می‌دهد.

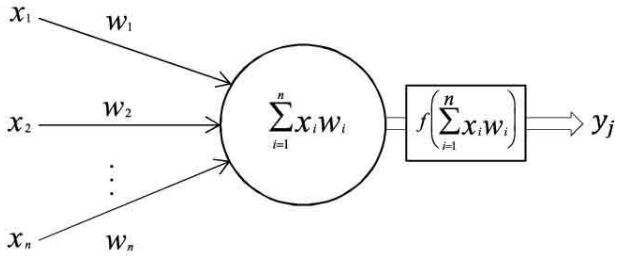


شکل ۱-۲ : ساختار نورون‌های شبکه عصبی انسان

۱-٤-۲- معماری شبکه‌های عصبی

اجزای اصلی شبکه‌های عصبی مصنوعی همچون شبکه اعصاب انسان نورون‌ها هستند. مشابه شبکه عصبی انسان هر نورون تعدادی ورودی و تعدادی خروجی دارد. در این شبکه اطلاعات از نورون‌های لایه اول وارد شده و به سمت نورون‌های لایه انتهایی می‌رود.

در معماری شبکه عصبی مصنوعی، هر یک از ورودی‌های نورون با وزن‌هایی که در ورودی متناظرشان است وزن‌دهی می‌شوند و پس از ورود به هسته نورون، مجموع آن‌ها محاسبه می‌شود. وزن‌ها اهمیت زیادی در فرآیند یادگیری شبکه عصبی دارند چرا که تنها مشخصه‌هایی هستند که در این فرآیند تغییر می‌کنند.

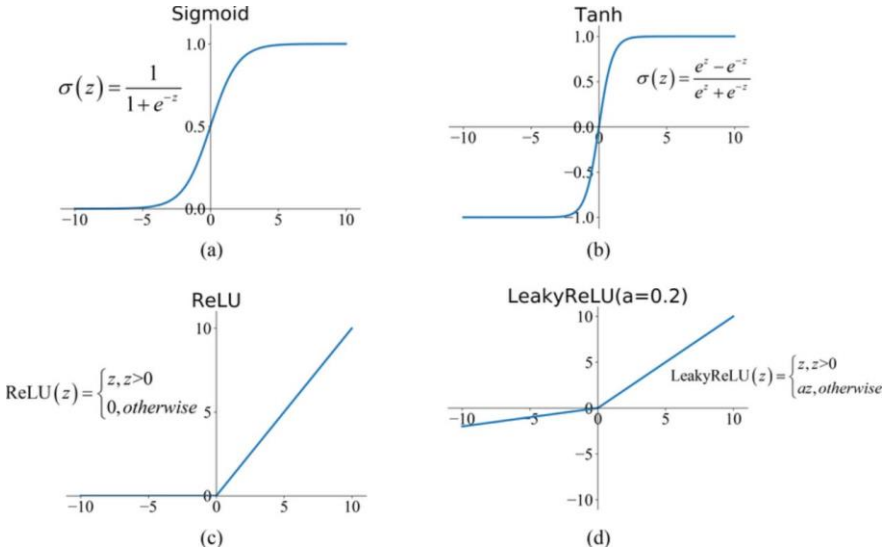


شکل ۱-۳ : ساختار نورون‌های شبکه عصبی مصنوعی

۱-٤-۲-۱- تابع فعال‌ساز

یکی از اجزاء مهم نورون‌ها، توابع فعال‌ساز هستند که بسته به مجموع ورودی‌های وزن‌دهی شده، مقدار خروجی هر نورون را مشخص می‌کنند. با کمک توابع فعال ساز خروجی هر نورون به بازه ۰ تا ۱ یا بازه -۱ تا ۱ تصویر می‌شود.

۴ نوع پرکاربرد توابع فعال ساز عبارت‌اند از: ReLU, LeakyReLU, Sigmoid, Tanh

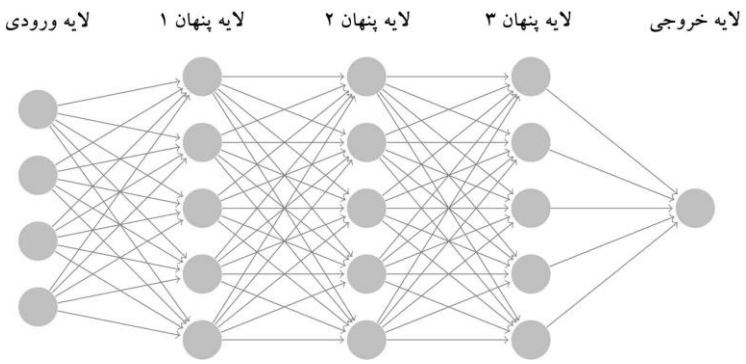


شکل ۱-۴ : چهار تابع فعال ساز پرکاربرد

۱-٤-۲-۲- لایه‌های پنهان[8]

نورون‌هایی که میان لایه ورودی و لایه انتهایی قرار گرفته‌اند روی یک یا چند لایه می‌توانند واقع شوند. ویژگی مهم نورون‌های یک لایه این است که به هم متصل نیستند. این نورون‌ها ورودی‌هایشان را تنها از لایه قبل می‌گیرند و خروجی‌هایشان را تنها به لایه بعد تحویل می‌دهند.

[8] Hidden Layers



شکل ۱-۵ : ساختار لایه‌های شبکه عصبی مصنوعی

۱-٤-۳- فرآیند یادگیری شبکه‌های عصبی

در هر فرآیند آموزش، شبکه عصبی با توجه به تفاوت میان خروجی‌اش با مقدار مطلوب سعی می‌کند همه وزن‌ها را به گونه‌ای تغییر دهد تا این فاصله کمینه شود.

طبق الگوریتم پس‌انتشار خطا[9]، خطای خروجی به سمت لایه‌های قبلی منتشر می‌شود و با محاسبه تاثیر هر وزن بر خروجی، هر وزن را تنظیم می‌کند.

۱-٤-۳-۱- تابع هزینه[10]

به فاصله میان خروجی شبکه عصبی با مقدار مطلوب تابع هزینه گفته می‌شود. هدف یادگیری در همه مسائل، کمینه کردن تابع هزینه است. برای مثال در مسائل رگرسیون که هدف یافتن خطی از بین نقاط داده شده است، تابع هزینه بر اساس فاصله نقاط تا خط تابع هزینه محاسبه می‌شود. همچنین در مدل‌های مولد که شبکه خروجی می‌سازد، تابع هزینه بر اساس فاصله داده تولید شده با داده‌های اصلی محاسبه می‌شود.

۱-٤-۳-۲- نرمال‌سازی

یکی از مسائل مهم در فرآیند یادگیری شبکه، پایدار و معتبر بودن تابع هزینه است. مشاهده می‌شود که با تغییر مشخصه‌های توزیع آماری ورودی‌ها، تابع هزینه نیز تغییر می‌کند که این وضعیت نامطلوب است. لذا لازم است قبل از ورود

[9] Backpropagation
[10] Cost Function

داده‌ها به شبکه نرمال‌سازی روی آن‌ها انجام گیرد. بر این اساس میانگین داده‌ها صفر و انحراف معیار آن‌ها روی ۱ نگه داشته می‌شود. این مسئله در لایه‌های میانی نیز با تغییر وزن‌های شبکه به‌وجود می‌آید. برای اجتناب از وقوع این مشکل لازم است ورودی‌های هر نورون نیز نرمال شوند. به این عمل نرمال‌سازی دسته‌ای[11] گفته می‌شود.

۱-۵- یادگیری عمیق[12]

یادگیری عمیق نوعی از یادگیری ماشین است که از چندین لایه شبکه عصبی برای استخراج ویژگی‌های سطح بالا از ورودی‌های خام استفاده می‌کند. در یادگیری عمیق از چندین لایه پردازشی و به‌ویژه اطلاعات غیرخطی استفاده می‌شود تا عملیات تبدیل یا استخراج ویژگی با اهدافی چون تحلیل یا بازشناخت الگو، کلاس‌بندی، خوشه‌بندی و ... انجام شود.

یادگیری عمیق در حوزه‌هایی متفاوتی همچون بینایی ماشین، پردازش متن، پردازش صوت، تولید داده و ... به کار می‌رود. دو دسته مهم یادگیری عمیق عبارت‌اند از:

- مدل‌های تفکیک‌کننده: برای کلاس‌بندی و تمییز داده‌ها از هم استفاده می‌شود.
- مدل‌های مولد: برای تولید داده‌های جدید استفاده می‌شود.

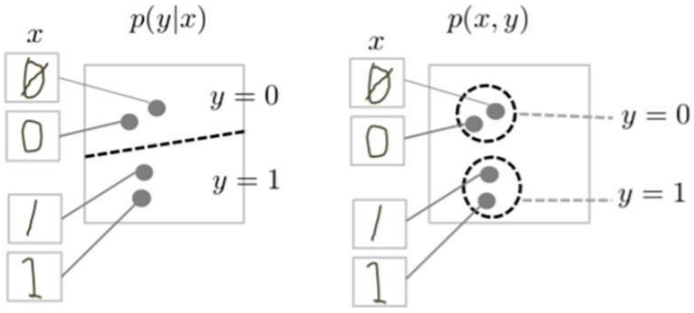
۱-۵-۱- مدل‌های تفکیک‌کننده[13]

در این مدل‌ها هدف تشخیص نوع و برچسب داده‌ها است. برای مثال در مورد مجموعه داده اعداد دست‌نویس، مدل تفکیک‌کننده قادر است اعداد مختلف را از هم تمییز دهد. یا اینکه در مورد مجموعه تصاویر چهره این مدل‌ها قادرند جنسیت و محدوده سنی افراد را تشخیص دهند. در سیستم تشخیص هویت نیز از این مدل‌ها استفاده می‌شود.

[11] Batch Normalization
[12] Deep Learning
[13] Discriminative Models

۱-۵-۲- مدل‌های مولد[14]

در مدل‌های مولد به‌جای تمرکز بر تمایز بین دسته‌های مختلف، بر روی مدل قرار گیری داده‌های هر کلاس در فضا و همبستگی داده‌ها تمرکز می‌شود. برای مثال با بررسی تصاویر چهره می‌تواند تشخیص دهد اجزاء صورت به چه ترکیبی کنار هم قرار می‌گیرند. مدل مولد با یادگیری شکل داده‌های آموزش می‌تواند داده‌های جدیدی مشابه آن‌ها تولید کند.



شکل ۱-۶ : مدل‌های مولد و مدل‌های تفکیک‌کننده

مدل تفکیک‌کننده تنها کافی است بتواند میان چند کلاس مختلف تمایز قائل شود در حالی که مدل‌های مولد موارد بیشتری را باید مدل کنند. برای مثال مدل تفکیک‌کننده قادر است یک سگ را از یک گربه تشخیص دهد. این در حالی است که مدل مولد قادر است تصاویر جدیدی از حیوانات تولید کند که همانند تصاویر واقعی باشد.

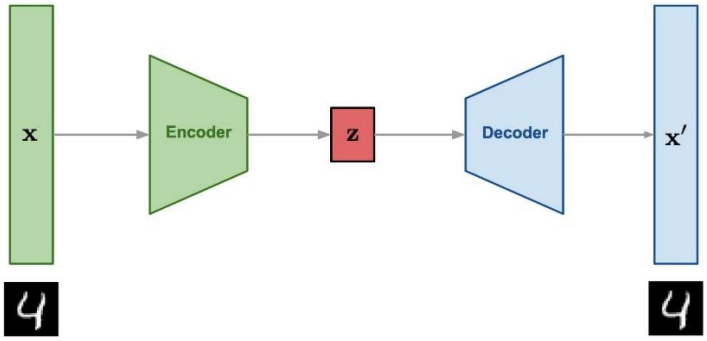
از نظر ریاضی مدل‌های تفکیک‌کننده احتمال شرطی $p(Y|X)$ و مدل‌های مولد توزیع احتمال مشترک $p(X, Y)$ را بدست می‌آورند. البته اگر یادگیری نظارت‌نشده باشد مدل مولد تنها $p(X)$ را باز می‌گرداند.

در ادامه چند معماری معروف مدل‌های مولد معرفی شده‌اند.

۱-۵-۲-۱- شبکه خودرمزگذار[15]

این شبکه از دو بخش رمزگذار (انکدر) و رمزگشا (دیکدر) ساخته شده است. داده‌های ورودی ابتدا وارد شبکه رمزگذار شده و ابعاد آن‌ها بسیار کم می‌شود. خروجی حاصل از شبکه رمزگذار سپس به شبکه رمزگشا وارد می‌شود و نهایتا تصویری هم ابعاد تصویر اولیه توسط آن تولید می‌شود.

[14] Generative Models
[15] Autoencoders



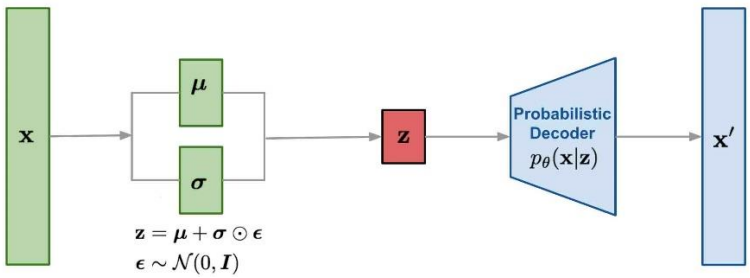
شکل ۱-۷: ساختار شبکه خودرمزگذار

تابع هزینه با محاسبه اختلاف تصویر اصلی با تصویر تولیدشده توسط خودرمزگذار بدست می‌آید و به کمک آن فرآیند یادگیری خودرمزگذار انجام می‌گیرد. یادگیری خودرمزگذار نظارت‌نشده است چرا که از داده برچسب‌دار استفاده نمی‌کند.

از شبکه خودرمزگذار می‌توان در مواردی چون فشرده‌سازی، شناسایی ناهنجاری، حذف نویز و واترمارک، رنگی کردن تصاویر، تولید داده جدید و ... استفاده می‌شود. پس از آموزش شبکه‌ها، برای تولید داده جدید تنها از شبکه رمزگشا استفاده می‌شود. با این تفاوت که به‌جای دادن خروجی رمزگذار، بردار نویز تصادفی را به آن می‌دهیم تا از طریق آن بتواند داده‌های جدید و متنوعی تولید کند.

۱-۵-۲-۲- شبکه خودرمزگذار متنوع[۱۶]

شبکه خودرمزگذار متنوع نوعی از شبکه خودرمزگذار است با این تفاوت که در بخش رمزگذار آن بردار میانگین و انحراف معیار تولید می‌شوند و در بخش رمزگشا مورد استفاده قرار می‌گیرند. خودرمزگذار متنوع قابلیت بالاتری در تولید داده دارد چرا که با استفاده از میانگین و انحراف معیار می‌تواند داده‌های متفاوتی از داده‌های ورودی تولید کند.

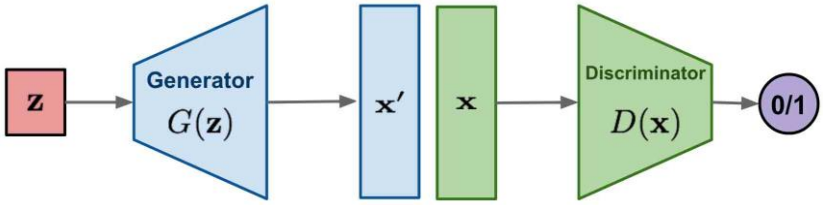


شکل ۱-۸: ساختار شبکه خودرمزگذار متنوع

[16] Variational Autoencoders (VAEs)

۱-۵-۲-۳- شبکه‌های متخاصم مولد[17]

این معماری بر پایه دو شبکه مصنوعی تفکیک‌کننده و مولد ساخته شده است. شبکه مولد داده‌های نویز تصادفی می‌گیرد و داده‌های مشابه با داده واقعی تولید می‌کند. داده‌های خروجی مولد به همراه داده‌های مجموعه داده اصلی به تفکیک‌کننده داده می‌شود و این شبکه واقعی یا تقلبی بودن آن‌ها را تشخیص می‌دهد. دو شبکه هدف متضادی دارند اما در عمل به یادگیری یک دیگر کمک می‌کنند و عملکرد یکدیگر را مدام بهتر می‌کنند.



شکل ۱-۹ : ساختار شبکه متخاصم مولد (گن)

شبکه گن به طرز چشمگیری نتایج واقعی‌تری از سایر مدل‌های مولد دارد و این امر باعث توجه بسیاری از متخصصان و ابرشرکت‌های فناوری به این معماری شده است. این مدل از دو تابع هزینه استفاده می‌کند. این در حالی است که خودرمزگذار متنوع تنها یک تابع هزینه دارد. موضوع اصلی این پژوهش معماری گن و خانواده‌های مختلف آن است. در فصل بعد به تفصیل معماری‌های مختلف خانواده گن توضیح داده شده‌اند.

۱-۵-۲-٤- شبکه خود رگرسیون[18]

این مدل برای تولید داده‌های جدید از یک گوشه تصویر شروع کرده و با ساخت پیکسل‌های جدید از پیکسل‌های قبلی تصویر نهایی را تولید می‌کند. یادگیری در این روش برخلاف سایر مدل‌های مولد، یادگیری نظارت شده است چرا که پیکسل‌های ابتدایی تصویر باید به آن داده شود.

[17] Generative Adversarial Networks (GANs)
[18] Autoregressive Models

فصل دوم

شبکه‌های متخاصم مولد (گن)

۲-۱- مقدمه

شـــبکه‌های متخاصـــم مولد یا به اختصـــار شـــبکه‌های گن[19] یکی از جدیدترین و موفق‌ترین معماری‌های یادگیری عمیق هستند. شبکه‌های گن قادرند برداری از نویز تصادفی را به تصاویری مشابه با تصاویر واقعی تبدیل کنند.

در این فصـل سـاختار این شـبکه، نحوه آموزش آن، چالش‌ها و مزیت‌ها و روش‌های مختلف ارزیابی خروجی شـبکه گن توضیح داده شده‌اند. همچنین در پایان این فصل، مجموعه‌ای از مهمترین کاربردهای شبکه‌های گن بیان شده‌اند.

[19] Generative Adversarial Networks (GANs)

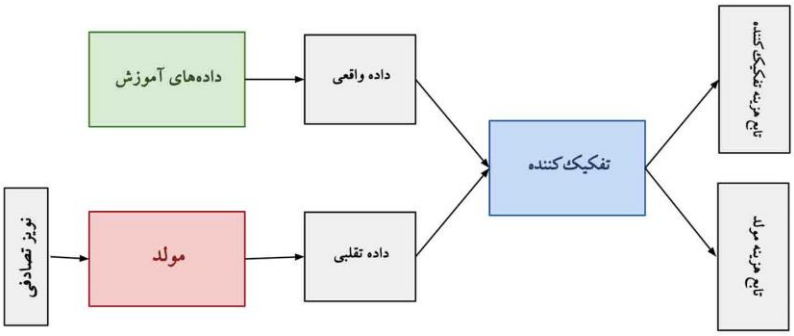
۲-۲- ساختار شبکه‌های گن [۱]

در یک معماری گن، دو شبکه عصبی یکی به عنوان مولد داده جدید و یکی به عنوان تفکیک‌کننده داده تقلبی از داده اصلی وجود دارند. در این معماری در هر فرآیند آموزش شبکه مولد سعی می‌کند داده‌های بهتری تولید کند تا شبکه تفکیک‌کننده نتواند آن‌ها را به عنوان داده تقلبی شناسایی کند و در مقابل شبکه تفکیک‌کننده سعی می‌کند در تشخیص داده تقلبی از داده اصلی هوشمندتر عمل کند. در واقع هر یک از دو شبکه سعی دارند خطای خود را کمینه و خطای دیگری را بیشینه کنند. چنین مسئله‌ای به‌نظر می‌رسد امکان‌پذیر نباشد.

جان نش ریاضی‌دان با اثبات ریاضی نشان داده است در مسائلی که دو یا چند عامل هر یک سعی دارند مطلوبیت خود را بیشینه کنند برخلاف تصور پیشین، عامل‌ها با حفظ مطلوبیت دیگر عامل‌ها می‌توانند مطلوبیت خود را افزایش دهند. جواب بهینه این مسائل زمانی رخ می‌دهد که هیچ یک از عامل‌ها نتواند مطلوبیت خود را بیش از پیش افزایش دهد. به احترام جان نش به این وضعیت، تعادل نش[۲۰] گفته می‌شود.

۲-۳- آموزش شبکه گن

فرآیند تولید تصاویر تقلبی با داده شدن یک بردار نویز تصادفی به شبکه مولد که هم‌بُعد با ورودی آن است شروع می‌شود. این بردار در شبکه عصبی مولد بسته به وزن‌ها و اتصالات نورون‌ها پردازش می‌شود و در پایان یک تصویر را می‌سازد. مقادیر نورون‌های لایه خروجی در واقع پیکسل‌های تصویر تولید شده‌اند لذا لازم است تعداد آن‌ها با تعداد پیکسل داده‌های آموزش برابر باشد.



شکل ۲-۱ : بخش تفکیک‌کننده و مولد شبکه‌های گن

[20] Nash equilibrium

تصاویر تولید شده و تصاویر واقعی به صورت تصادفی به تفکیک‌کننده داده می‌شوند و مشخص می‌کند کدام یک تقلبی و کدام واقعی است. لذا ورودی شبکه عصبی بایستی تفکیک‌کننده هم‌بُعد با تصاویر باشد و خروجی آن تنها یک نورون داشته باشد. تفکیک‌کننده توسط این نورون واقعی یا تقلبی بودن تصویر ورودی را مشخص می‌شود.

از آنجا که واقعی یا تقلبی بودن تصاویر برای خود ما مشخص است، می‌توانیم میزان خطای شبکه تفکیک‌کننده و با کمک آن تابع هزینه مولد و تفکیک‌کننده را محاسبه کنیم. آموزش تفکیک‌کننده و مولد با الگوریتم پس انتشار خطا و بر اساس تابع هزینه متناظرشان محاسبه می‌شود.

از مجموع تابع هزینه تفکیک‌کننده و مولد، تابع هزینه BCE [۲۱] یا تابع هزینه کمینه-بیشینه[۲۲] بدست می‌آید. مولد تلاش می‌کند تابع بدست آمده را بیشینه کند در حالی که تفکیک کننده سعی دارد مقدار این تابع را کمینه کند.

$$E_x\left[\log\left(D(x)\right)\right] + E_z\left[\log\left(1 - D\left(G(z)\right)\right)\right]$$

D(x): احتمال تشخیص داده‌های اصلی به عنوان واقعی

D(G(z)): احتمال تشخیص داده‌های تولید شده به عنوان واقعی

۲-۳-۱- آموزش شبکه تفکیک‌کننده

تفکیک‌کننده داده‌های هر دو منبع یعنی داده‌های واقعی و داده‌های تولید شده توسط مولد را کلاس‌بندی می‌کند. تابع هزینه تفکیک کننده، شبکه تفکیک‌کننده را به خاطر اشتباه در کلاس بندی داده‌های واقعی به عنوان داده تقلبی یا داده تقلبی به عنوان داده واقعی جریمه می‌کند.

میزان خطای تفکیک‌کننده طبق الگوریتم پس انتشار خطا به سمت لایه‌های عقب منتشر می‌شود و وزن‌های لایه‌های مختلف شبکه را تنظیم می‌کند. در زمان آموزش تفکیک‌کننده، وزن‌های شبکه مولد ثابت می‌مانند.

[21] BCE Loss
[22] Minimax Loss

۲-۳-۲- آموزش شبکه مولد

تابع هزینه مولد، شبکه مولد را به خاطر تولید داده‌هایی که تفکیک‌کننده توانسته تقلبی بودن آن‌ها را تشخیص دهد جریمه می‌کند. بر این اساس مولد یاد می‌گیرد چگونه داده‌ای تولید کند که تفکیک‌کننده آن را به عنوان داده واقعی شناسایی کند.

خروجی تابع هزینه مولد از طریق تفکیک‌کننده به عقب باز می‌گردد تا به خروجی مولد برسد و بر اساس الگوریتم پس‌انتشار خطا وزن‌های این شبکه را مجددا تنظیم کند. در فرآیند آموزش مولد، وزن‌های تفکیک‌کننده ثابت هستند.

۲-٤- چالش‌ها و مشکلات رایج گن

شبکه‌های گن تعدادی مشکل رایج دارند. همه این مشکلات حوزه‌های تحقیق فعال هستند و هیچ یک از این مشکلات به طور کامل حل نشده است. در ادامه برخی از مهمترین این چالش‌ها و راه‌حل‌هایی که برای آن‌ها پیشنهاد شده است معرفی شده‌اند.

۲-٤-۱- شیب‌های محو شونده[۲۳]

در صورتی که شبکه تفکیک‌کننده خیلی بیشتر از مولد آموزش دیده باشد، آموزش مولد ممکن است در یک شیب محو شونده گیر بیافتد. چرا که تفکیک‌کننده می‌تواند اکثر داده‌های تولید شده توسط مولد را به عنوان تقلبی شناسایی کند و با این کار اطلاعات کافی برای بهبود در اختیار مولد قرار نمی‌دهد.

تا به حال تلاش‌هایی برای اجتناب از این وضعیت معرفی شده‌اند. در ادامه به دو مورد از آن‌ها اشاره شده است:

- تابع هزینه واسرستاین: این تابع هزینه حتی در حالتی که تفکیک کننده به حالت بهینه خود رسیده است از وقوع شیب‌های محو شونده جلوگیری می‌کند. در معماری گن واسرستاین که در بخش‌های بعدی توضیح داده شده است، از این تابع هزینه استفاده می‌شود.
- تابع هزینه کمینه-بیشینه اصلاح شده: در مقاله اصلی معرفی گن این تغییر برای جلوگیری از شیب‌های محوشونده معرفی شده است.

[23] Vanishing Gradients

۲-٤-۲- حالت فروپاشی مُد[24]

مطلوب این اســت که مولد بتواند داده‌های بســیار متنوع در دســته‌های مختلف تولید کند. برای مثال بتواند چهره‌هایی با رنگ پوست و مو متفاوت، زاویه چهره متفاوت، حالت بینی و دهان متفاوت و ... تولید کند.

گاهی ممکن است مولد بتواند تعداد کمی داده قابل قبول تولید کند که هیچ موقع تفکیک‌کننده نتواند آن‌ها را شناسایی کند. در این وضعیت که به آن فروپاشی مد گفته می‌شود، مولد می‌آموزد تنها همان داده‌های محدود را تولید کند تا بتواند تفکیک‌کننده را همواره فریب دهد.

راه‌حل‌های پیشنهاد شده زیر با جلوگیری از بهینه شدن مولد در یک محدوده خاص، از فروپاشی مد جلوگیری می‌کنند:

- تابع هزینه واسرستاین: این تابع هزینه با جلوگیری از شیب‌های محوشونده، به تفکیک‌کننده می‌آموزد خروجی‌هایی که مولد روی آن‌ها تثبیت شده است را رد کند. با این کار مولد مجبور می‌شود موارد تازه‌ای را امتحان کند.
- گن‌های تثبیت‌نشــده[25]: در این مدل از معماری گن، مولد از یک تابع هزینه‌ای که طبقه‌بندی‌هایی فعلی و آینده تفکیک‌کننده را در بر می‌گیرد اســتفاده می‌کند. بنابراین مولد نمی‌تواند برای یک تفکیک‌کننده بیش از حد بهینه‌سازی کند.

۲-٤-۳- شکست در همگرایی[26]

گن دائما در همگرا شدن شکست می‌خورد. چرا که پس از مرور زمان بازخوردی که تفکیک‌کننده به مولد می‌دهد کمتر معنی‌دار می‌شود. در این شرایط اگر مولد بخواهد آموزش را با بازخوردهای تصادفی تفکیک‌کننده ادامه دهد، کیفیت آن کمتر می‌شود.

پژوهشگران راه‌حل‌های مختلفی را برای بهبود همگرایی گن آزموده‌اند، از جمله:

- افزودن نویز به ورودی تفکیک کننده
- جریمه کردن وزن‌های شبکه تفکیک کننده

[24] Mode Collapse
[25] Unrolled GANs
[26] Failure to Converge

۲-۵- ارزیابی خروجی گن

ارزیابی مدل‌های مولد از جمله معماری گن دشوار است چرا که خروجی مشخصی ندارند. دو مشخصه مهم که در ارزیابی خروجی‌های مدل گن مدنظر قرار می‌دهیم عبارت‌اند از: کیفیت تصاویر، تنوع تصاویر.

برای مقایسه تصاویر تولید شده با تصاویر واقعی دو روش پیشنهاد شده است:

- محاسبه فاصله پیکسل‌ها: مجموع فاصله تک‌تک پیکسل‌ها می‌تواند به عنوان معیار ارزیابی قلمداد شود. این روش چندان نتایج قابل قبولی نمی‌دهد.
- محاسبه فاصله ویژگی‌ها: اگر بتوان به طریقی ویژگی‌های سطح بالای تصاویر را استخراج کرد، مقایسه این ویژگی‌ها میان تصویر اصلی و تصویر ساختگی می‌تواند معیار قابل قبولی برای ارزیابی خروجی باشد. در بخش بعدی نحوه استخراج ویژگی‌ها توضیح داده شده است.

۲-۵-۱- استخراج ویژگی‌ها[۲۷]

برای استخراج ویژگی‌ها، از کلاس‌بندهایی استفاده می‌شود که لایه‌های انتهایی آن بریده شده است. از آنجا که آخرین لایه قبل از لایه‌ی آخر دارای بیشترین اطلاعات از ویژگی‌ها است، معمولا با بریدن آخرین لایه‌ی پولینگ استخراج‌کننده ویژگی بدست می‌آید.

برای ساخت استخراج‌کننده ویژگی به‌جای آموزش یک کلاس‌بند معمولا از کلاس‌بندهای از پیش آموخته استفاده می‌شود. یکی از جامع‌ترین مجموعه‌های داده که در علوم داده و هوش مصنوعی بسیار استفاده می‌شود، مجموعه داده ImageNet است که شامل ۱۴ میلیون تصویر و ۲۰ هزار دسته است. یک استخراج‌کننده ویژگی بسیار متداول که بر روی این مجموعه داده آموزش دیده و لایه آخر آن حذف شده است، با نام inception-v3 یا به اختصار inception (سرآغاز) شناخته می‌شود.

دو روش استفاده از inception معرفی شده است: فاصله سرآغازین فریچت (FID) و امتیاز سرآغاز (IS)

[27] Feature Extraction

۲-۵-۱-۱- فاصله سرآغازین فریچت (FID)[۲۸]

در این روش فاصله بین توزیع نرمال چندمتغیره داده‌های مصنوعی و اصلی محاسبه می‌شود. کمتر بودن این مقدار نمایانگر نزدیک‌تر بودن توزیع داده‌های تولید شده به داده‌های اصلی است.

$$\mathrm{FID} = ||\mu_x - \mu_y||^2 + \mathrm{Tr}\left(\Sigma_x + \Sigma_y - 2\sqrt{\Sigma_x \Sigma_y}\right)$$

Σ_x: ماتریس کواریانس

μ_x: میانگین

Tr (trace): مجموع عناصر قطر اصلی

این روش دارای ضعف است. از آن‌جا که از یک کلاس‌بند پیش آموخته استفاده می‌کند نمی‌تواند الزاما همه ویژگی‌ها را به خوبی مشخص کند. همچنین برای پاسخ‌دهی مناسب نیاز به داده با ابعادی زیاد دارد که این باعث زمان‌بر شدن این روش می‌شود. دیگر مشکل آن این است که نمی‌تواند همه تفاوت دو مجموعه داده را مشخص کند چرا که تنها از دو مشخصه آماری میانگین و کواریانس استفاده می‌کند.

۲-۵-۱-۲- امتیاز سرآغاز (IS) [۲۹]

یک روش دیگر که برای ارزیابی از طریق کلاس‌بند inception معرفی شده است، روش امتیاز سرآغاز است.

$$IS(G) = \exp\left(E_{x\sim P_g}\, D_{KL}\big(P(Y|X) \parallel P(Y)\big)\right)$$

در این رابطه دیورژانس KL از رابطه زیر بدست می‌آید:

$$D_{KL}\big(P(Y|X) \parallel P(Y)\big) = P(Y|X)\ \log\left(\frac{P(Y|X)}{P(Y)}\right)$$

[28] Fréchet Inception Distance

[29] Inception Score

استفاده از این روش مشکلات زیادی داشته است:

- به آسانی در یک مد گیر می‌افتد. (حالت فروپاشی مد)
- تنها به داده‌هایی که به عنوان تقلبی شناسایی شده‌اند، توجه می‌کند. با این کار نمی‌تواند متوجه نقاط قوت داده‌هایی که توانسته تفکیک‌کننده را فریب دهد بشود.
- مجموعه داده ImageNet نمی‌تواند همه چیز را پوشش دهد. لذا خیلی از ویژگی‌ها را ممکن است از دست دهد.

بررسی‌ها نشان داده استفاده از روش اول نتایج قابل قبول‌تری دارد.[۲]

۲-۵-۲- روش ارزیابی ادراکی چشم انسان: هایپ [۳۰] [۳]

علی رغم پیشرفت‌های زیاد هوش مصنوعی کماکان راه‌های موثری برای شناسایی داده‌های تقلبی معرفی نشده است و قوه تشخیص انسان برتری قابل توجهی دارد. البته این کار نیاز به زمان و هزینه زیادی دارد. یک راه حل مناسب و عملی شکستن پروژه‌ای به این عظمت به وظایف کوچک بین تعداد زیادی انسان است.

یکی از این تلاش‌ها روش هایپ است که توسط جمعی از محققان دانشگاه آکسفورد در سایت ترک مکانیکی آمازون[۳۱] پیاده سازی شده است. در این سایت پروژه‌های سنگین که نیاز به کار انسانی دارد با هزینه کم بین تعداد افراد زیادی برون‌سپاری می‌شود.

الگوریتم هایپ به دو طریق پیاده سازی و اجرا شده است:

- Hype time: بررسی زمان تقریبی برای تشخیص تقلبی بودن داده‌های تولید شده توسط گن
- Hype infinity: بررسی درصد خطا در تشخیص تقلبی بودن داده‌ها

[30] Human eYe Perceptual Evaluation (HYPE)
[31] https://worker.mturk.com

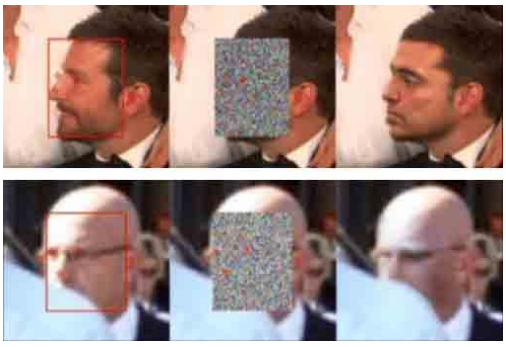
۲-٦- موارد کاربرد گن

توانایی خارق‌العاده خانواده گن در تولید داده‌های واقعی موجب تعریف کاربردهای بسیار متنوعی برای آن شده است. در ادامه به تعدادی از کاربردهای مهم خانواده گن اشاره شده است:

- ایجاد تصاویر جدید برای مجموعه‌های داده کم: یکی از چالش‌های محدودکننده در یادگیری ماشین، کمبود تعداد داده‌های مناسب است. گن می‌تواند تا حد زیادی این مشکل را برطرف کند و به توسعه سایر الگوریتم‌های یادگیری ماشین کمک شایانی کند.
- افزایش وضوح تصاویر: معماری گن قادر است تصاویر با وضوح کم را به وضوح بالا برساند. این توانایی گن می‌تواند برای کاهش حجم تصاویر با اطمینان از توانایی بازیابی آن در سمت گیرنده استفاده شود.

شکل ۲-۲ : بازیابی تصویر اصلی از تصویر کم حجم شده

- حفظ حریم خصوصی: برای استفاده از تصاویر افراد و داده‌های پزشکی همواره حفظ حریم خصوصی و محرمانگی یک محدودیت در یادگیری ماشین محسوب می‌شود. معماری گن می‌تواند با تولید تصاویر جدید از افراد غیرواقعی این چالش را برطرف کند.
- حفظ گمنامی: به کمک گن افرادی که نمی‌خواهند یا نباید تصاویرشان دیده شود می‌توانند هویت خود را در مصاحبه‌ها حفظ کنند. این کاربرد برای قربانیان حملات، شاهدان جرایم و فعالین اجتماعی می‌تواند استفاده شود.

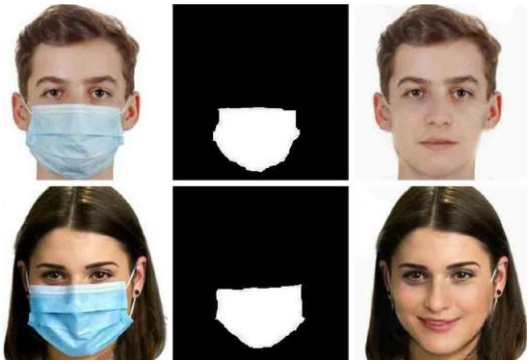


شکل ۲-۳: استفاده از گن برای حفظ گمنامی

- ایجاد شـخصـیت‌های کارتونی: در صـنعت انیمشـین‌سـازی یکی از پرچالش‌ترین و هزینه‌برترین کارها سـاخت شخصیت‌های جدید است. گن قادر است این کار را با کمترین هزینه و زمان انجام دهد.

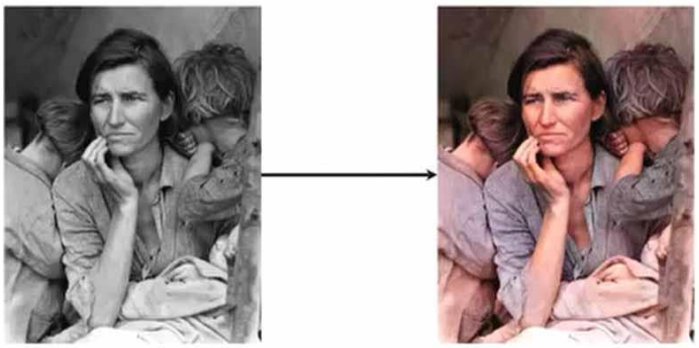
شکل ۲-۴ : تولید کاراکترهای کارتونی با استفاده از گن

- نقاشی صورت: گن قادر است تصاویر ناقص چهره را تکمیل کند که نتایج نسبتا قابل قبولی داشته است.



شکل ۲-۵ : بازسازی تصاویر ناقص یا ماسک شده توسط گن

- رنگی‌کردن تصاویر سیاه سفید: مدل پایه گن که تنها قادر بود تصاویر سیاه سفید با وضوح پایین تولید کند الآن به حدی پیشرفت کرده است که قادر است سایر عکس‌های سیاه و سفید را نیز رنگی کند.



شکل ۲-۶ : تبدیل تصاویر سیاه و سفید به رنگی با استفاده از شبکه‌های عصبی گن

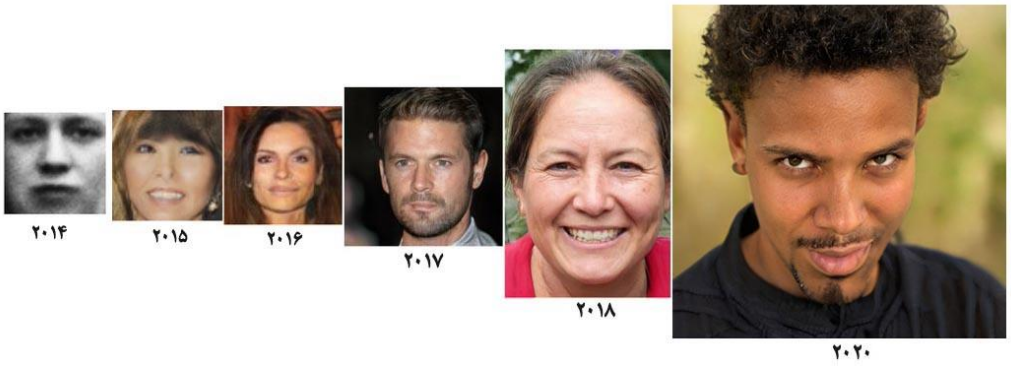
فصل سوم

معماری‌های مختلف شبکه‌های مصنوعی گن

۳-۱- مقدمه

معماری شبکه‌های گن برای اولین بار در سال ۲۰۱۴ معرفی شده است و تا به امروز معماری‌های جدیدی از این خانواده معرفی شده‌اند. معماری‌های جدید این خانواده در تولید داده‌های واقعی پیشرفت‌های چشم‌گیری داشته‌اند. همان طور که در شکل ۳-۱ نشان داده شده است، این اولین بار است که هوش مصنوعی قادر است تصاویری به این اندازه واقعی تولید کند که حتی از نگاه انسان به‌راحتی قابل تشخیص نباشند.

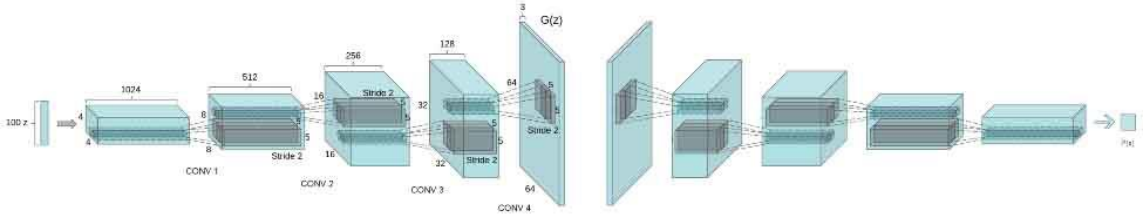
در فصل قبل، مدل پایه معماری گن و چالش‌ها و مزایای آن توضیح داده شد. در این فصل با مطرح شدن چالش‌ها و مزایای هر معماری، قدم‌به‌قدم معماری‌های مختلفی که به ترتیب این سال‌ها معرفی شده‌اند توضیح داده شده‌اند.



شکل ۳-۱ : پیشرفت خانواده معماری‌های گن

۳-۲- شبکه‌های گن کانولوشنی[۳۲] [٤]

گن کانولوشنی بر پایه معماری گن معمولی است با این تفاوت که در مولد و تفکیک‌کننده از شبکه‌های عصبی کانولوشنی استفاده شده است. در شبکه‌های کانولوشنی بلوک‌های پشت‌سرهم قرار گرفته شده‌اند و اندازه ورودی را تغییر می‌دهند. در هر بلوک، معمولا دو لایه کانولوشن و یک لایه پولینگ قرار گرفته است.



شکل ۳-۲ : ساختار شبکه‌های گن کانولوشنی

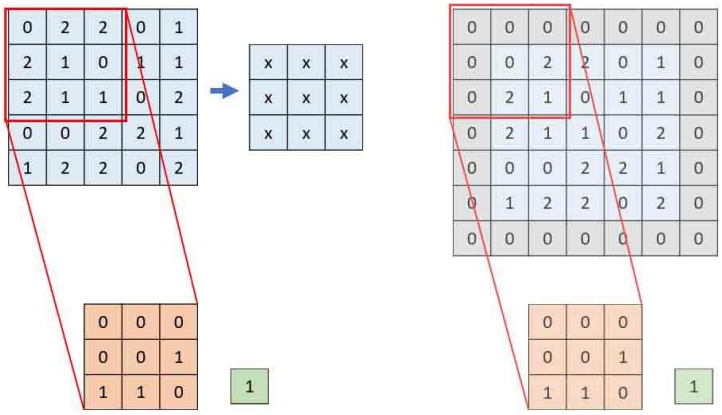
یکی از چالش‌های معماری گن پایه، عدم توانایی آن بر کنترل خروجی است. چرا که مشخص نیست هر وزن باعث چه تغییراتی در خروجی می‌شود. معماری گن کانولوشنی اولین تلاش برای کنترل بیشتر بر روی آموزش شبکه و همچنین بهتر کردن نتایج بوده است.

[32] Deep Convolutional GAN

۳-۲-۱- لایه‌ی کانولوشن

در لایه‌های کانولوشنی روی ماتریس ورودی عمل کانولوشن انجام می‌شود. برای انجام عمل کانولوشن روی ماتریس ورودی نیاز به یک ماتریس مربعی کم‌بعد با مقادیر از پیش تعیین شده به نام ماتریس هسته[۳۳] است. ماتریس هسته در جابه‌جای ماتریس اصلی قرار می‌گیرد و در هر مکان قرارگیری، مجموع حاصل ضرب تک‌تک عناصر نظیر دو ماتریس بدست آمده و در درایه نظیر ماتریس خروجی قرار می‌گیرد.

هسته ماتریس متحرک نمی‌تواند روی عناصر کناری ماتریس اصلی قرار گیرد و این باعث کاهش ابعاد ماتریس نهایی می‌شود. برای اجتناب از این مشکل قبل از کانولوشن عملیات لایه‌گذاری[۳۴] انجام می‌شود. برای انجام این کار یک قاب خالی دور ماتریس ورودی قرار می‌گیرد تا ماتریس هسته بتواند در عناصر کناری نیز قرار بگیرد و ابعاد یکسان باقی بماند.



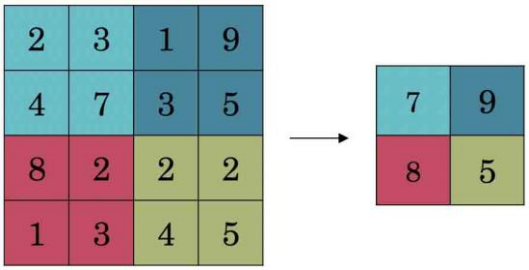
شکل ۳-۳: عملیات کانولوشن با استفاده از لایه‌گذاری (Padding) و بدون آن

۳-۲-۲- لایه‌ی پولینگ[۳۵]

در شبکه‌های عصبی کانولوشنی برای کاهش حجم محاسبات بعد از دو لایه کانولوشن یک لایه پولینگ قرار داده می‌شود. این لایه از بلوک‌های مربعی به ابعاد یکسان یک عنصر آن را در خروجی قرار می‌دهد. این کار باعث کاهش ابعاد تصویر می‌شود.

[33] Kernel
[34] Padding
[35] Pooling

در لایه پولینگ نماینده هر بلوک می‌تواند میانگین، کمینه یا بیشینه آن بلوک باشد. در شبکه عصبی کانولوشنی به طور معمول از مقدار بیشینه استفاده می‌شود چرا که اطلاعات ارزشمندتری را بازنمایی می‌کند. به این روش پولینگ بیشینه[۳۶] گفته می‌شود.

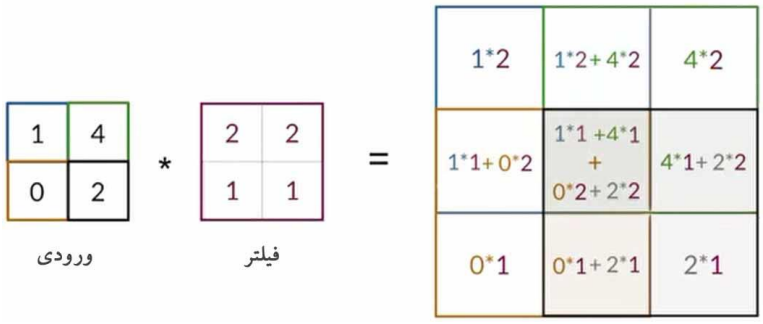


شکل ۳-۴ : عملیات پولینگ بیشینه (Max Pooling)

کاهش ابعاد می‌تواند در لایه کانولوشن نیز انجام داد. این کار با تغییر گام[۳۷] حرکت هسته امکان‌پذیر است. در واقع گام تعیین می‌کند ماتریس هسته در هر حرکت چند خانه جابه‌جا شود.

۳-۲-۳- لایه کانولوشن معکوس[۳۸]

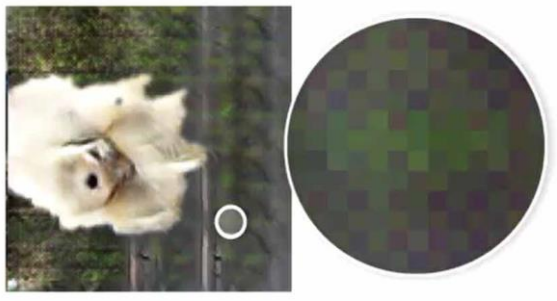
لایه کانولوشن معکوس ابعاد تصویر را افزایش می‌دهد. به این خاطر در شبکه مولد که برای ساخت تصویر نیاز است یک ورودی کم بعد به ابعاد بالاتر تبدیل شود از لایه‌های کانولوشن معکوس استفاده می‌شود. در این لایه عملیات کانولوشن معکوس انجام می‌شود. نحوه انجام این عملیات در شکل۳-۵ نشان داده شده است.



شکل ۳-۵ : عملیات کانولوشن معکوس

[36] Max Pooling
[37] Stride
[38] deconvolution

همان‌طور که در شـکل۳-۵ دیده می‌شـود در ماتریس خروجی برخی از درایه‌ها از چند مجموع سـاخته شـده‌اند که این باعث افزایش غلظت برخی نقاط در تصویر خروجی می‌شود. شکل ۳-۶ خروجی یک شبکه گن کانولوشنی است که در آن در بخش‌هایی از تصویر اعوجاج‌های رنگی دیده می‌شود.



شکل ۳-۶ : اعوجاج رنگی در خروجی تصاویر گن کانولوشنی

برای جلوگیری از این اعوجاج رنگی دو راه حل وجود دارد:

- در عمل کانولوشن معکوس، ابعاد بلوک مضربی از مقدار گام انتخاب شود.
- به‌جای استفاده از لایه کانولوشن معکوس ابتدا با روش‌های جایگزین تصویر را به ابعاد بالاتر برده شده و سپس عمل کانولوشن معکوس انجام شود.

۳-۳- شبکه‌های گن مشروط[۳۹] [۵]

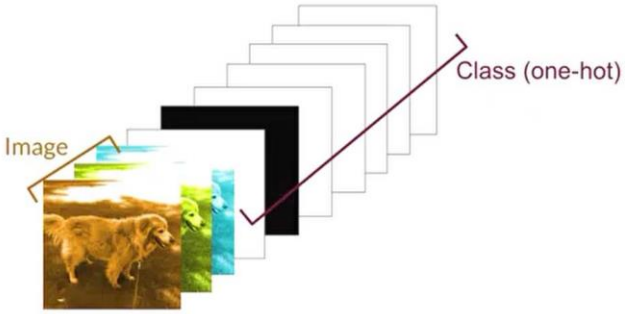
گن مشروط نوعی از خانواده گن است که از داده‌های برچسب‌دار استفاده می‌کند. یکی از چالش‌های شبکه گن معمولی عدم توانایی آن بر کنترل روی فرآیند تولید داده‌ها در کلاس‌های مختلف اسـت. در گن مشـروط با کمک برچسـب داده‌ها می‌توان شبکه مولد را موظف کرد که داده‌هایی در کلاس‌های مختلف تولید کند.

در این معماری شـبکه تفکیک‌کننده وظیفه سـخت‌گیرانه‌تری دارد. چرا که تنها در صـورتی تصـویر را به عنوان اصـلی شناسایی می‌کند که علاوه بر واقعی بودن، برچسب مشابهی نیز داشته باشد.

شماره کلاس از طریق بردار hot-۱ به مولد و از طریق ماتریس hot-۱ به تفکیک‌کننده اعمال می‌شود.

[39] Conditional GAN

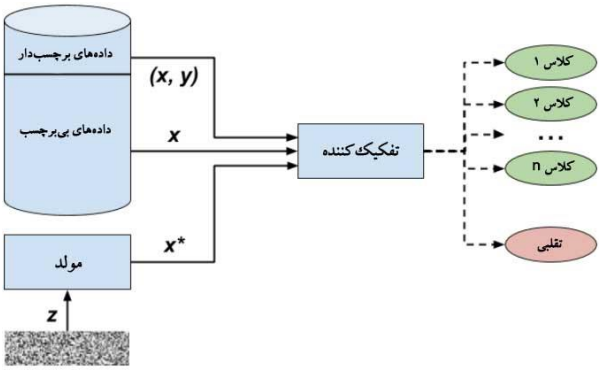
شکل ۳-۷: بردار ۱-hot



شکل ۳-۸: ماتریس ۱-hot

۳-٤- شبکه‌های گن نیمه نظارتی[40] [٦]

شبکه گن نیمه نظارتی خاصی از خانواده گن است که در آن از یادگیری نیمه نظارتی استفاده می‌شود. به این معنا که تعداد اندکی از داده‌ها برچسب‌دار است. در این معماری در بخش تفکیک کننده یک کلاس‌بند با n+۱ کلاس وجود دارد (n کلاس نوع داده و یک کلاس برای داده‌های تقلبی).



شکل ۳-۹: ساختار شبکه‌های گن نیمه نظارتی

شبکه گن نیمه نظارتی وظیفه سنگین‌تری نسبت به گن معمولی دارد چرا که علاوه بر تشخیص تقلبی یا اصلی بودن تصاویر، داده‌های اصلی را باید کلاس بندی نیز بکند. به این خاطر بخش تفکیک‌کننده این معماری اهمیت ویژه‌تری دارد.

[40] Semi Supervised GAN

۳-۵- شبکه‌های گن واسرستاین[41] [۷, ۸]

مشکلات تابع هزینه BCE که مدل پایه گن بر اساس آن بنا شده بود، باعث روی آوردن به توابع هزینه جایگزین شد. یکی از بهترین جایگزین‌های تابع هزینه BCE، تابع هزینه واسرستاین است. معماری گن واسرستاین بر اساس این تابع هزینه بنا شده است و به این علت این نام برای آن انتخاب شده است.

۳-۵-۱- تابع هزینه واسرستاین[42]

تابع هزینه واسرستاین قادر است سه اشکال اساسی تابع هزینه BCE یعنی فروپاشی مُد و شیب محو شونده و یادگیری سریع‌تر تفکیک‌کننده را حل کند. از این رو گن واسرستاین در یک مد گیر نمی‌افتد و حتی با وجود تفاوت بسیار زیاد بین دو توزیع آماری، ناحیه‌های مسطح به‌وجود نمی‌آورد.

فرآیند یادگیری مولد به نسبت پیچیده‌تر از تفکیک‌کننده است چرا که وظیفه سنگین‌تری برعهده دارد. این مسئله در معماری پایه گن که از تابع هزینه BCE استفاده می‌کند باعث می‌شود تفکیک کننده خیلی زودتر از مولد آموزش ببیند و مجالی برای یادگیری به مولد ندهد چرا که همه خروجی‌هایش را می‌تواند به عنوان تقلبی شناسایی کند.

فاصله توزیع حرکت زمین[43] از نظر ریاضی بهینه‌ترین جواب برای فاصله بین توزیع دو مجموعه را می‌دهد.[۹] تابع هزینه واسرستاین تخمینی است از این فاصله.

$$\min_g \max_c \mathrm{E}\big(c(x)\big) - \mathrm{E}\Big(c\big(g(z)\big)\Big)$$

۳-۵-۲- بخش منتقد[44]

خروجی این بخش بر خلاف مدل پایه گن که بین ۰ و ۱ است، می‌تواند هر مقداری داشته باشد. شبکه منتقد برای استفاده از تابع هزینه واسرستاین محدودیت دارد چرا که حتما باید 1-L continuous باشد تا بتواند به صورت صحیح مقدار EMD را تخمین بزند. بر اساس این شرط قدر مطلق شیب تابع هزینه در تمام نقاط باید کمتر از ۱ باشد.

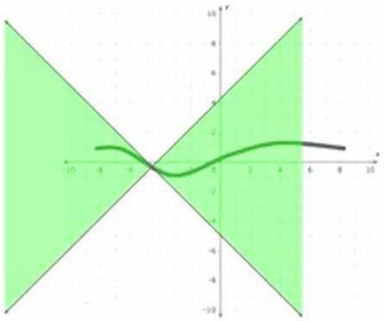
---

[41] Wasserstein GAN

[42] W-loss

[43] Earth Mover's Distribution (EMD)

[44] در معماری واسرستاین به‌جای استفاده از لفظ تفکیک‌کننده از کلمه منتقد یا Critic استفاده می‌شود.



شکل ۳-۱۰ : نحوه بررسی 1-L continuous بودن یک تابع

برای وادار کردن تابع هزینه منتقد برای اجابت این شرط دو راه کار وجود دارد:

- بریدن وزن‌ها: با محدود کردن وزن شبکه منتقد و در واقع بریدن مازاد آن می‌توان به این خواسته رسید.
- جریمه کردن شیب: با افزودن یک عبارت تنظیم‌کننده به تابع هزینه واسرستاین

$$\min_g \max_c \mathrm{E}\big(\mathrm{c}(\mathrm{x})\big) - \mathrm{E}(\mathrm{c}(\mathrm{g}(\mathrm{z})) + \lambda\, \mathrm{E}(\| \nabla \mathrm{c}(\hat{\mathrm{x}}) \|_2 - 1)$$

بین این دو روش، روش جریمه شیب نتایج نسبتا بهتری دارد.

۳-۵-۳- بخش مولد

بخش مولد در شبکه‌های گن واسرستاین مشابه بخش مولد مدل پایه معماری گن است. بر این اساس به شبکه مولد بردار نویز تصادفی با ابعاد کم داده می‌شود و پس از عبور آن از بین لایه‌های شبکه عصبی، در خروجی تصویری هم ابعاد با تصاویر اصلی تولید می‌شود. تعداد نورون‌های لایه خروجی این شبکه، به تعداد پیکسل‌های تصویر است.

۳-٦- شبکه‌های گن استایل[45] [۱۰, ۱۱]

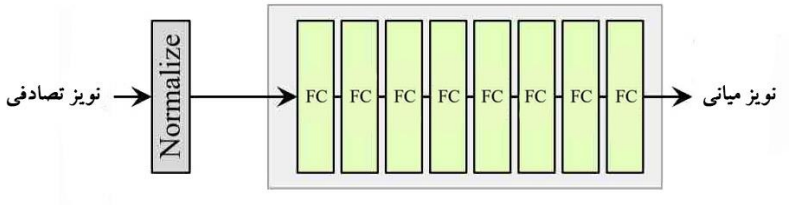
از جدیدترین معماری‌های معرفی شده‌ی خانواده گن، معماری گن استایل است. در این معماری چند هدف مهم همزمان دنبال شده و باعث شده به کمک آن بتوان تصاویر بی‌نظیری از چهره انسان تولید کرد.[46] این اهداف عبارت‌اند از:

- تولید تصاویر واقعی‌تر در رزولوشن بالا
- افزایش تنوع خروجی‌ها
- کنترل بیشتر روی ویژگی‌های تصاویر تولید شده

در این معماری منظور از استایل هر گونه تغییرات روی ظاهر چهره‌های تولید شده است. استایل در چندین سطح از کم تا زیاد قابل تعریف است. برای مثال استایل‌های چهره عبارت‌اند از زاویه و حالت چهره، رنگ پوست و مو، حالت چشم و دهان و بینی و ... .

۳-٦-۱- شبکه تصویرکننده[47]

در معماری گن استایل، بردار نویز به صورت مستقیم به مولد وارد نمی‌شود و لازم است در ابتدا توسط شبکه تصویرکننده که متشکل از هشت لایه تمام متصل[48] است پردازش اولیه‌ای روی انجام گیرد. خروجی این شبکه نویز میانی نامیده می‌شود.



شکل ۳-۱۱ : تبدیل بردار نویز تصادفی به نویز میانی با شبکه تصویرکننده

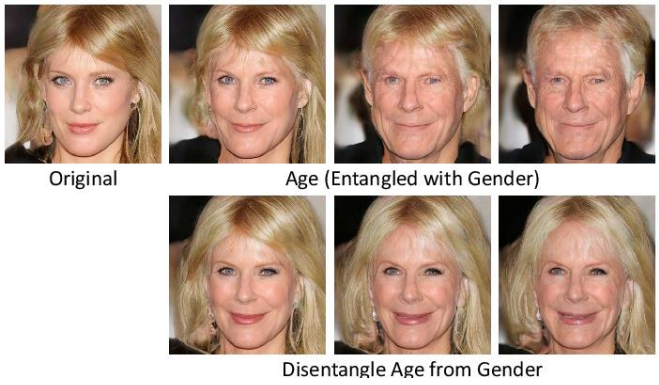
[45] StyleGAN

[46] نمونه‌هایی از تصاویر تولید شده توسط معماری گن استایل در سایت thispersondoesnotexist.com قابل مشاهده است.

[47] Mapping Network

[48] Fully Connected

نویز میانی که ابعاد بالاتری از نویز تصـــادفی اولیه دارد تا حد بســـیار خوبی این امکان را به ما می‌دهد که بتوانیم ویژگی خاصـی از تصـویر نهایی را بدون تغییر یافتن سـایر ویژگی‌ها تغییر دهیم. در شـکل ۳-۱۲ هدف تغییر دادن سـن بوده اما در ردیف اول جنسیت هم دچار تغییر شـده اسـت چرا که امکان تغییر درایه‌های خاصـی از بردار نویز اولیه برای رسـیدن به چنین خواسته‌ای میسر نبوده است. در ردیف دوم این تصویر که از بردار میانی استفاده شده است، این اتفاق رخ نمی‌دهد.



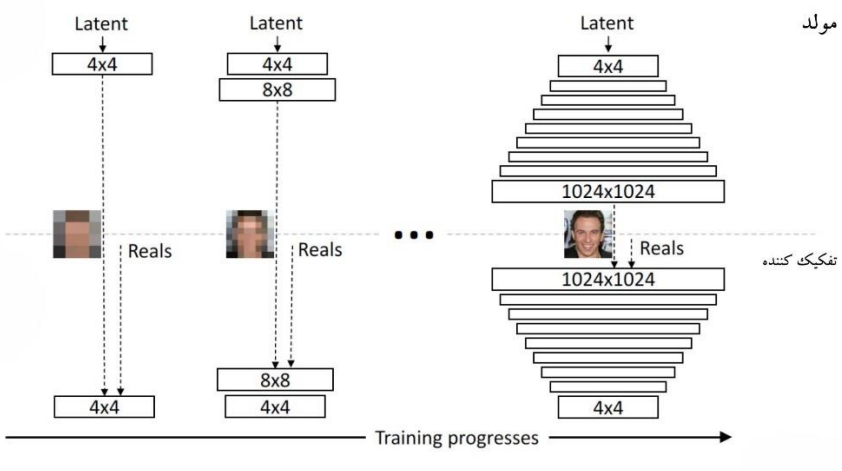
شکل ۳-۱۲ : تغییر سن تصویر در حالت استفاده از نویز میانی و بدون استفاده از آن

۳-۶-۲- بخش رشد تدریجی[۴۹]

در معماری گن استایل، شبکه مولد کار خود را با تولید تصاویر با وضوح بسیار کم (۴×۴ پیکسل) شروع می‌کند. این تصـــاویر به همراه تصـــاویر اصـــلی که به همین ابعاد درآمده‌اند به بخش تفکیک‌کننده می‌روند و با محاســـبه میزان خطای تفکیک‌کننده هر دو شبکه مولد و تفکیک کننده آموزش می‌بینند.

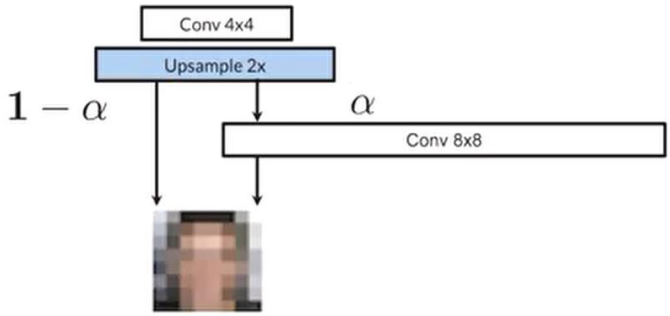
به تدریج و به طور مداوم ابعاد تصـاویر دو برابر می‌شـود تا نهایتا به تصـاویر با وضـوح بالا (۱۰۲۴×۱۰۲۴ پیکسـل) برسـد. افزایش تدریجی وضـوح تصـاویر با وارد شـدن تدریجی لایه‌های کانولوشـنی ابعاد بالاتر به مولد و تفکیک کننده حاصـل می‌شود.

[49] Progressive Growing



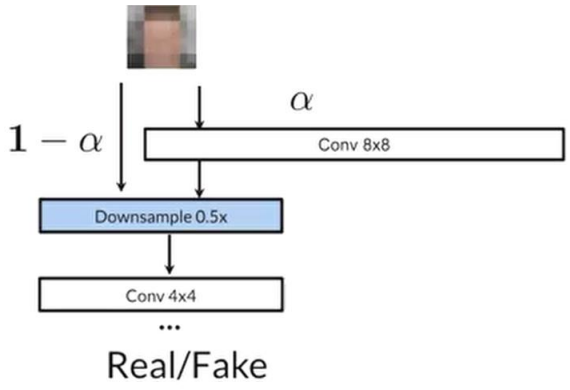
شکل ۳-۱۳ : بخش رشد تدریجی در معماری گن استایل

افزوده شـــدن لایه‌های کانولوشـــنی ابعاد بالاتر به یک باره رخ نمی‌دهد و در چندین مرحله آموزش و به تدریج انجام می‌شود. در شکل ۳-۱۴، در ابتدا مقدار α صفر است به این معنا که تنها لایه کانولوشنی ۴×۴ در شبکه قرار دارد. به تدریج مقدار α زیاد می‌شـود تا به یک برسـد. در این شـرایط لایه کانولوشـنی ۸×۸ جای لایه کانولوشـنی ۴×۴ را به کلی می‌گیرد. همین فرآیند تا لایه کانولوشنی ۱۰۲۴×۱۰۲۴ و رسیدن به تصاویر با این ابعاد ادامه می‌یابد.



شکل ۳-۱۴ : قسمت مولد در بخش رشد تدریجی

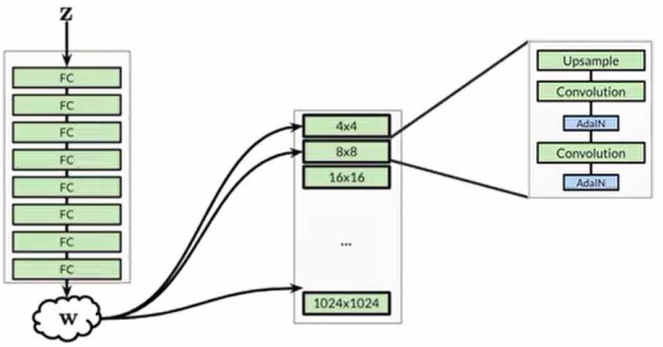
همراه با تغییر تدریجی لایه‌های کانولوشنی در مولد و در نتیجه تصاویر تولید شده‌ی با وضوح بیشتر، فرآیند مشابهی در شـــبکه تفکیک کننده نیز انجام می‌شـــود. به این ترتیب به تدریج لایه‌های کانولوشـــنی کم ابعاد جای خود را به لایه‌های کانولوشنی بعدی می‌دهند.



شکل ۳-۱۵ : قسمت تفکیک‌کننده در بخش رشد تدریجی

۳-۶-۳- بلوک‌های نرمال‌کننده انطباقی (AdaIN)[50]

نویز میانی به همه بلوک‌های مولد وارد می‌شود. همان‌طور که در شکل ۳-۱۶ نشان داده شده است، هر کدام از بلوک‌های مولد از یک لایه افزایش ابعاد و دو لایه کانولوشنی ساخته شده است و بعد از یک از لایه‌های کانولوشنی یک بلوک AdaIN قرارگرفته است. نویز میانی از طریق بلوک‌های AdaIN وارد ساختار مولد می‌شوند.



شکل ۳-۱۶ : ساختار کلی معماری گن استایل

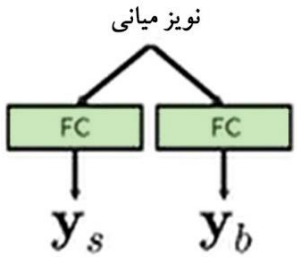
بلوک نرمال‌کننده انطباقی دو وظیفه را بر عهده دارد:

۱. نرمال‌کردن خروجی لایه کانولوشنی

۲. اعمال استایل انطباقی با استفاده از نویز میانی

[50] Adaptive Instance Normalization

نویز میانی پیش از ورود به بلوک‌های نرمال‌کننده انطباقی وارد دو لایه تمام متصل می‌شود که خروجی یکی y_s است که به عنوان مقیاس و خروجی دیگری y_b است که به عنوان بایاس به AdaIN اعمال می‌شوند.



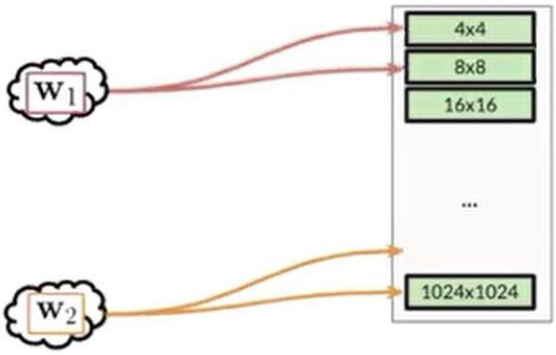
شکل ۳-۱۷ : نحوه اعمال نویز میانی به لایه‌های تمام متصل پیش از ورود آن به AdaIN

$$AdaIN(x, y) = y_{s,i}\frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i}$$

۳-٦-٤- ترکیب دو استایل

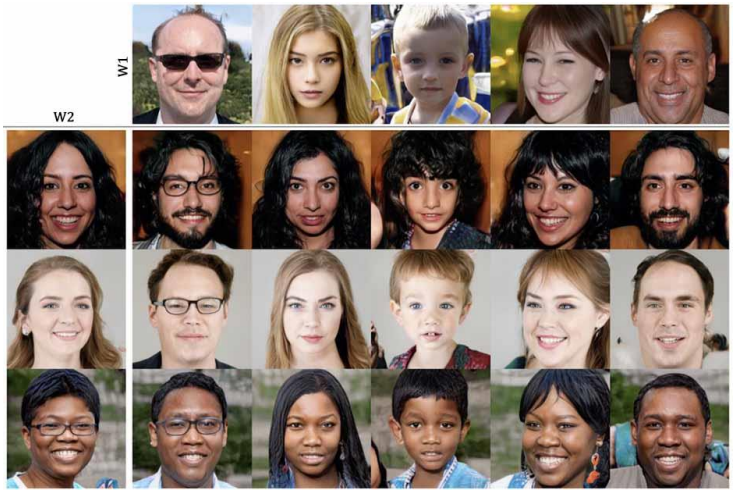
یکی از ویژگی‌های منحصربه‌فرد معماری گن استایل امکان ترکیب دو تصویر برای ساخت تصویری جدید است. حتی می‌توان مشخص کرد هر کدام از ویژگی‌های چون رنگ مو، پوست، حالت و زاویه چهره و ... از کدام یک به ارث برسد.

برای ترکیب دو تصـــویر به‌جای این‌که به تمامی بلوک‌های مولد یک بردار نویز میانی وارد شـــود، می‌توان به لایه‌های ابتدایی بردار نویز میانی یکی از تصـــاویر و به لایه‌های پایینی بردار نویز میانی تصـــویر دیگر را اعمال کرد. لایه‌های ابتدایی کلیت تصویر را می‌سازند. مثل جنسیت و زاویه صورت و لایه‌های پایینی جزئیات تصویر را می‌سازند. مثل رنگ چهره و مو، حالت چهره و ...



شکل ۳-۱۸ : نحوه ترکیب نویزهای میانی دو تصویر برای ترکیب استایل آن‌ها

در شکل ۳-۱۹ ترکیب دو دسته از تصاویر با هم نمایش داده شده است. تصاویر ردیف بالا تعیین کننده حالت صورت و جنسیت هستند چرا که بردار میانی آن‌ها به لایه‌های ابتدایی مولد وارد شده است و تصاویر ستون چپ جزئیات صورت مثل رنگ مو، پوست، چشم و ... را در تصویر ترکیب شده ساخته است.

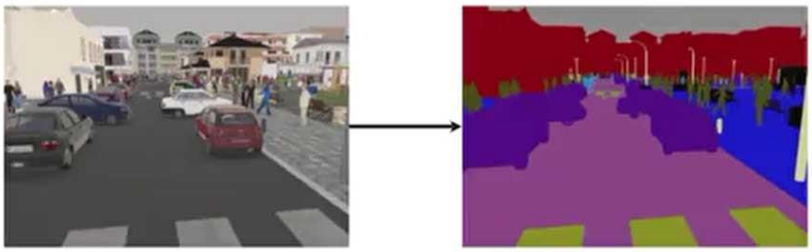


شکل ۳-۱۹ : ترکیب استایل تصاویر

۳-۷- معماری Pix2Pix [۱۲]

این معماری نوعی از گن مشروط است با این تفاوت که به مولد به‌جای دادن بردار کلاس‌ها، نقشه بخش‌بندی‌شده[51] تصاویر اصلی به مولد داده می‌شود و لذا نظارت‌شده است. در این معماری دیگر بردار نویز به مولد داده نمی‌شود و تنها کار خود را با ترجمه تصویر داده شده انجام می‌دهد.

نقشه بخش‌بندی‌شده، تصویری است هم ابعاد با تصویر اصلی که بخش‌های مختلف تصویر اصلی با رنگ‌های متفاوت روی آن مشخص شده است. در شکل ۳-۲۰ خیابان، پیاده‌رو، خط عابر پیاده، ماشین‌ها، عابرین و ساختمان‌ها هر یک با رنگ‌های جدا مشخص شده‌اند.

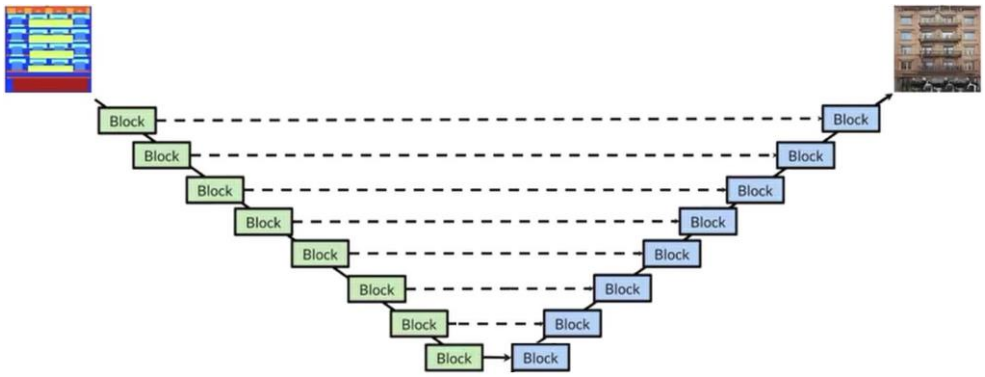


شکل ۳-۲۰ : تبدیل تصاویر واقعی به نقشه بخش‌بندی شده (Segmentation Map)

[51] Segmentation Map

۳-۷-۱- بخش مولد (U-Net)

مولد معماری Pix۲Pix از هشت رمزگذار (انکدر) و هشت رمزگشا (دیکدر) تشکیل شده است. ابعاد تصویر پس از عبور از خر انکدر کمتر می‌شود تا اینکه وارد انکدر هشت‌ام شود. پس از عبور تصویر از گلوگاه و وارد شدن به بخش دیکدرها این عمل به صورت وارونه انجام می‌شود.



شکل ۳-۲۱: معماری U-Net بخش مولد معماری Pix2Pix

در معماری U-Net ، زوج انکدر-دیکدرهای هم‌بُعد از طریق اتصالات پرشی[۵۲] مستقیما به هم وصل شده‌اند. این لینک‌ها به انکدرها و دیکدرها اجازه می‌دهد بتوانند اطلاعات مهمی را مستقیما از یکدیگر بیاموزند.

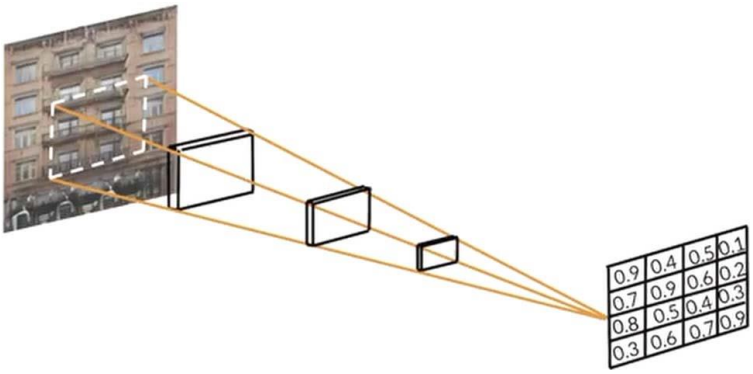
هر بلوک انکدر از یک لایه کانولوشنی، یک لایه نرمال‌کننده بسته‌ای تشکیل شده است. تابع فعال ساز آن Leaky ReLU است. همچنین هر بلوک دیکدر از یک لایه کانولوشن معکوس، یک لایه نرمال‌کننده تشکیل شده است.. تابع فعال ساز انکدرها ReLU است. در ضمن، سه دیکدر اول که بعد از گلوگاه قرار گرفته‌اند، دارای لایه حذف تصادفی[۵۳] نیز هستند.

۳-۷-۲- بخش تفکیک‌کننده (PatchGAN)

بخش تفکیک‌کننده این معماری یک تفاوت اساسی با سایر مدل‌های خانواده گن دارد و آن این است که به‌جای تولید یک عدد که نشان‌دهنده واقعی یا مصنوعی بودن تصویر است، ماتریسی از این اعداد را باز می‌گرداند. هر درایه این ماتریس

[52] Skip connections
[53] Dropout Layer

عددی بین ۰ و ۱ است که از بررسی واقعی یا مصنوعی بودن قسمتی از تصویر بدست آمده است. همچون معماری پایه گن از تابع هزینه BCE در تشخیص صحت عکس‌ها استفاده شده است.

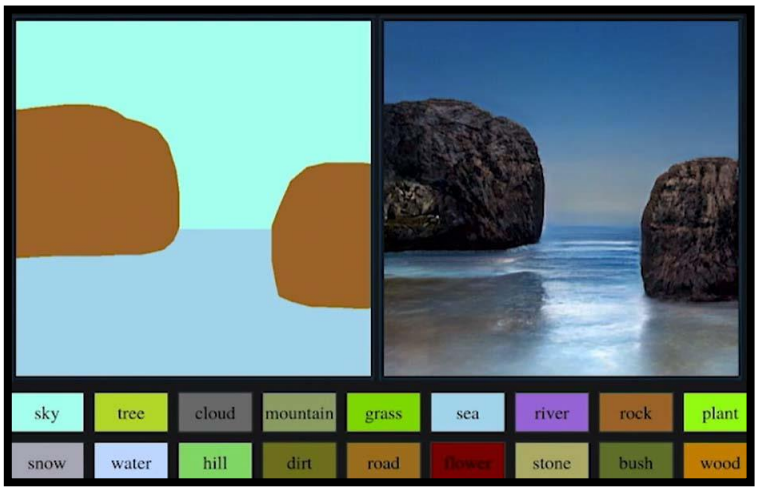


شکل ۳-۲۲: خروجی بخش تفکیک‌کننده معماری Pix2Pix

۳-۷-۳- پیشرفت‌های معماری Pix2Pix

دو مورد از معماری‌های موفقی که از معماری پایه Pix۲Pix استفاده می‌کنند، Pix۲PixHD [۱۳] و GauGAN [۱۴] هستند. هر دوی این معماری‌ها قادر هستند تصاویر با وضوح بالا تولید کنند. Pix۲PixHD با استفاده از چند رنگ که هر کدام نشان‌دهنده یک جزء صورت است، می‌تواند تصاویر چهره با وضوح بالا متناسب با این بخش‌بندی تولید کند. GauGAN با فرآیند مشابه می‌تواند تصاویر منظره بسیار واقعی تولید کند.

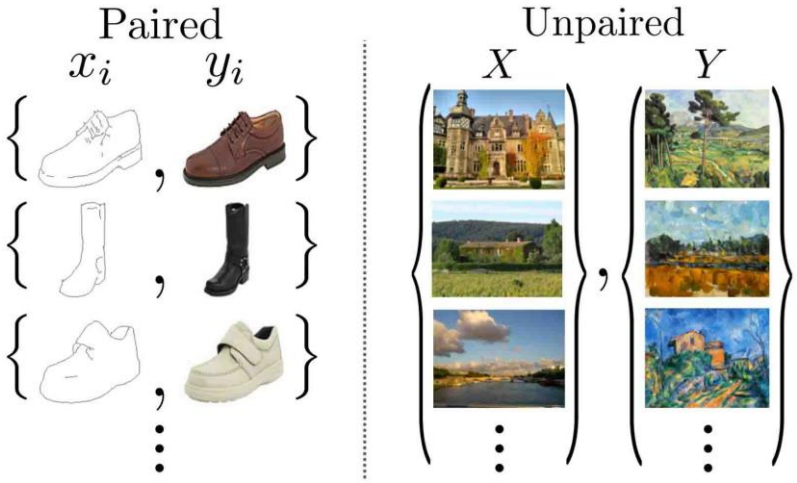
شکل ۳-۲۳: تولید تصویر چهره از نقشه برچسب با معماری Pix2PixHD



شکل ۳-۲۴ : تولید تصویر منظره از نقشه برچسب با معماری GauGAN

۳-۸- شبکه‌های گن چرخشی[54] [۱۵]

این معماری ترجمه غیرجفتی اســـت به این معنا که مجموعه‌ای از تصـــاویر را حوزه‌ای به حوزه دیگر تبدیل می‌کند. این مدل با بررسی دو مجموعه تصویر که به آن داده می‌شود می‌تواند ویژگی‌های مشترک و ویژگی متمایزکننده آن‌ها را بشناسد. در مقابل، ترجمه‌های جفتی مثل Pix2Pix تنها یک عکس را از حوزه‌ای به حوزه دیگر تبدیل می‌کند.



شکل ۳-۲۵ : تصاویر جفتی و غیر جفتی

[54] CycleGAN

گن چرخشی از به هم پیوستن دو شبکه گن ساخته می‌شود لذا دو مولد و دو تفکیک‌کننده دارد. مولدها بر اساس معماری U-Net و تفکیک‌کننده‌ها بر اساس معماری PathGAN پیاده‌سازی شده‌اند.

۳-۸-۱- توابع هزینه گن چرخشی

فرآیند یادگیری معماری گن چرخشی سخت‌تر از معماری‌های قبلی است چرا که هیچ خروجی مطلوبی مستقیما وجود ندارد. به همین خاطر برای ارزیابی تصاویر خروجی از سه تابع هزینه استفاده می‌شود که هر کدام وظیفه خاصی در ارزیابی تصاویر خروجی دارند.

۳-۸-۱-۱- تابع حداقل هزینه مربع‌ها[۵۵]

این تابع هزینه برای واقعی‌تر کردن خروجی مولدها استفاده می‌شود. در مدل پایه گن از تابع هزینه BCE برای این هدف استفاده می‌شود. تابع حداقل هزینه مربع‌ها به نسبت از BCE پایدارتر است چرا که مشکلاتی چون شیب‌های محوشونده را تا حد زیادی کاهش می‌دهد.

برای تفکیک‌کننده:

$$E_x[(D(x)-1)^2] + E_z\left[\left(D\left(G(z)\right)-0\right)^2\right]$$

برای مولد:

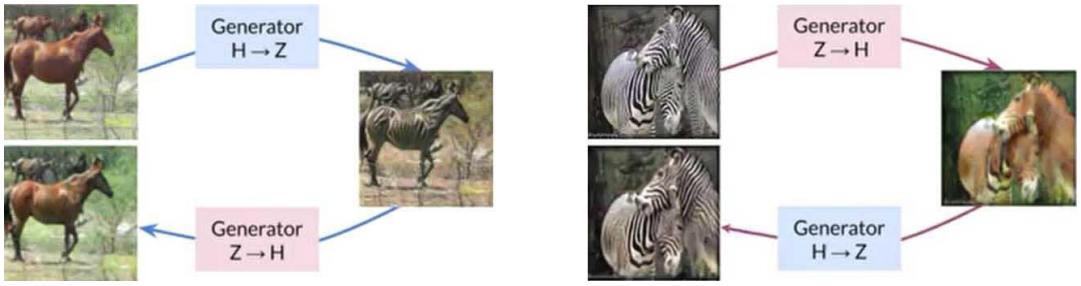
$$E_x[D(G(z))-1^2]$$

۳-۸-۱-۲-تابع هزینه ثبات چرخه[۵۶]

این تابع هزینه در انتقال اجزاء غیر مشترک تصاویر بین دو شبکه گن کمک می‌کند. برای محاسبه تابع هزینه ثبات چرخه یک تصویر به دامنه دیگر انتقال می‌یابد و مجددا به دامنه ابتدایی باز می‌گردد. این فرآیند در جهت عکس نیز انجام شده و

[55] Least Squares Loss
[56] Cycle Consistency Loss

فاصله میان تصویر ابتدایی و انتهایی محاسبه می‌شود. تابع هزینه ثبات چرخه از مجموع دو فاصله ذکر شده بدست می‌آید. حالت بهینه زمانی است که این حاصل جمع کمینه شود. این تابع هزینه فقط برای مولدها استفاده می‌شود.



شکل ۳-۲۶ : تبدیل تصویر به حوزه دیگر و تبدیل مجدد از آن حوزه به حوزه اصلی

۳-۸-۱-۳- تابع هزینه اصلیت[۵۷]

اگر تصویری از یک دامنه را به مولد شبکه روبرویی‌اش بدهیم انتظار می‌رود تغییری در آن ایجاد نکند. برای مثال اگر تصویر اسبی را به مولد ترجمه‌کننده تصویر گورخر به اسب بدهیم نباید روی آن تغییری ایجاد کند. اما در خیلی از مواقع، گن چرخشی باعث تغییر رنگ تصویر می‌شود. تابع هزینه اصلیت تلاش می‌کند رنگ‌های تصویر اصلی را حفظ کند. با این حال استفاده از آن اختیاری است.

برای محاسبه تابع هزینه اصلیت، تصویری از یک دامنه را به مولد شبکه مقابل می‌دهند. همین فرآیند در جهت عکس نیز انجام می‌شود و فاصله پیکسل‌های تصویر اصلی و تصویر تغییررنگ‌یافته بدست می‌آید. تابع هزینه ثبات چرخه از مجموع دو فاصله ذکر شده بدست می‌آید. این تابع هزینه فقط برای مولدها استفاده می‌شود.

[57] Identity Loss

فصل چهارم

نتایج شبیه سازی، موانع و مشکلات

٤-١- مقدمه

محدودیت سخت‌افزاری کامپیوترهای شخصی سد بزرگی برای پردازش‌های سنگین آموزش شبکه‌های عصبی گن است. در این فصل ابتدا محیط برنامه‌نویسی پروژه معرفی شده‌اند و پس از آن تلاش‌ها و راه‌کارهایی که برای مقابله با این محدودیت در طی انجام این پروژه انجام شده است، توضیح داده شده‌اند.

در ادامه این فصل، مجموعه داده‌هایی که در پروژه استفاده شده‌اند معرفی شده‌اند و در پایان نتایج شبیه‌سازی‌های انجام شده، ارائه شده‌اند.

٤-٢- معرفی محیط برنامه‌نویسی

برای نوشتن دستورات مربوط به ساخت لایه‌های شبکه‌های عصبی و آموزش آن با مجموعه داده از زبان برنامه‌نویسی پایتون استفاده شده است.

ماژولار بودن و شیء گرایی زبان پایتون، باعث شده عمده کارهای یادگیری ماشین و هوش مصنوعی با این زبان انجام شود. علاوه بر این بسیاری از کتابخانه‌های پرکاربرد مورد استفاده در علوم داده و هوش مصنوعی برای این زبان نوشته شده و توسعه یافته‌اند. در ادامه کتابخانه‌های مورد استفاده در این پروژه معرفی شده‌اند.

٤-٢-١- تنسورفلو[58]

تنسورفلو یک فریم ورک جامع و انعطاف پذیر از ابزارها، کتابخانه‌ها و منابع است که به محققان اجازه می دهد کارهای سنگین در حوزه یادگیری ماشین را با آن انجام دهند. تنسورفلو یک پلتفرم متن باز[59] برای یادگیری ماشین است که به صورت انتها-به-انتها[60] توسط شرکت گوگل پیاده‌سازی شده است.

٤-٢-٢- کراس[61]

کتابخانه کراس یک رابط نرم‌افزاری[62] سطح بالا برای تنسورفلو است که به محققان اجازه می‌دهد بدون درگیری با پیچیدگی‌های کد زدن با تنسورفلو از همان بستر ولی به زبان ساده‌تری استفاده کنند. همچنین کتابخانه کراس دارای چندین مجموعه داده است.

[58] Tensorflow
[59] Open Source
[60] End-to-end
[61] Keras
[62] API

٤-٢-٣- پانداس[٦٣]

کتابخانه پانداس، یک کتابخانه متن باز است که کارایی بالا، ساختاری با قابلیت استفاده آسان و ابزارهای تحلیل داده را در زبان پایتون فراهم می‌کند. کاربرد اصلی این کتابخانه برای پیش پردازش داده‌ها و وارد کردن یا خروجی گرفتن با فرمت‌های گوناگونی مثل MS Excel، TSV و CSV است.

٤-٢-٤- نامپای[٦٤]

از کتابخانه نامپای برای کار با وکتورها، توابع ریاضی و آرایه‌های چند بعدی و ... استفاده می‌شود. توابع و متدهای مختلف و کابردی این کتابخانه باعث شده که نامپای جزئی جدایی ناپذیر از اکثر برنامه و کدها در زمینه علوم داده باشد. از جمله از نامپای در زمینه‌هایی چون علوم شناختی، پردازش سیگنال، بینایی ماشین، شبکه‌ها و گراف، محاسبات کوانتومی، انواع شبیه‌سازی و یادگیری ماشین استفاده می‌شود.

٤-٣- معرفی مجموعه‌های داده مورد استفاده

در این پروژه از دو مجموعه داده رایگان و پرکاربرد در یادگیری ماشین استفاده شده است.

- مجموعه داده celebrities١٠٠k [٦٥]: این مجموعه داده شامل ١٠٠ هزار تصویر بدون برچسب از چهره با ابعاد ١٢٨×١٢٨ پیکسل است.

شکل ۴-۱ : چند تصویر از مجموعه داده celebrities-100k

[63] Pandas
[64] Numpy
[65] https://www.kaggle.com/greg115/celebrities-100k

- مجموعه داده MNIST : این مجموعه داده مجموعه‌ای برچسب‌دار از اعداد دست‌نویس لاتین شامل ۶۰ هزار داده آموزش ۲۸×۲۸ پیکسل و ۱۰ هزار داده تست با همین ابعاد است.

شکل ۴-۱ : چند تصویر از مجموعه داده MNIST

٤-٤- راه‌کارهای مقابله با محدودیت‌های سخت‌افزاری

شبکه‌های عصبی گن در میان همه معماری‌های مختلف هوش مصنوعی، می‌توان گفت سنگین‌ترین پردازش را داراست. به‌خصوص معماری‌های جدید گن که قادرند تصاویر با وضوح بسیار بالا تولید کنند. محدودیت سخت‌افزاری کامپیوترهای شخصی سد بزرگی برای پردازش‌های سنگین آموزش شبکه‌های عصبی گن است.

در این بخش تلاش‌ها و راه‌کارهایی که برای مقابله با این محدودیت سخت‌افزاری در طی انجام این پروژه انجام شده است، توضیح داده شده‌اند.

٤-٤-١- رایانش ابری[٦٦]

استفاده از رایانش ابری قدمی بزرگ در راستای اشتراکی‌سازی منابع و کاهش به شمار می‌رود. به طوری که پیش‌بینی می‌شود پارادایم اشتراکی‌شدن در مقابل پارادایم مالکیت که پیش از این بر زندگی بشر سیطره داشت بتواند در چند سال آینده تغییرات شگرفی در نوع زندگی انسان‌ها ایجاد کند.

ابرشرکت‌های بزرگی چون آمازون و مایکروسافت با ایجاد مراکز داده ابری عظیم و قدرتمند قادر هستند به مشتریان خود توان پردازشی، دیسک، نرم‌افزار و بسیاری دیگر از خدمات و سرویس‌ها را عرضه کنند و از آن‌ها هزینه اشتراک بگیرند.

[66] Cloud Computing

برای مثال فردی که نیاز به توان پردازشـــی خیلی زیاد اما در مدت کوتاه دارد می‌تواند به‌جای خریدن ســـخت‌افزارهای عظیم‌الجثه و گران قیمت، از رایانش ابری اســتفاده کند و تنها به میزان نیاز خود هزینه پرداخت کند. این کار به خصــوص در مورد پردازش‌های سنگینی مثل آموزش شبکه‌های عصبی بسیار راهبردی و کمک کننده است.

در راســتای انجام این پروژه به چندین ســایت ارائه دهنده خدمات ابری مراجعه شــد. متاســفانه با وجود اینکه همگی آن‌ها فرصــت کوتاه چند ماهه‌ای را به صــورت رایگان در اختیار کاربران قرار می‌دادند ولی به دلیل نیاز داشــتن آن‌ها به اطلاعات کریدیت کارت معتبر نتوانستم از این طریق پردازش‌های خود را انجام دهم.

٤-٤-٢- سرویس گوگل کلاب[٦٧]

گوگل محیط کامپایل و نوشتن کدهای برنامه‌نویسی به زبان‌های مختلف را با نام گوگل کلاب ارائه داده است. این محیط تقریبا مشابه محیط برنامه‌نویسی ژوپیتر است و می‌توان در آن هم کد نوشت و هم توضیحاتی را ثبت کرد.

گوگل کلاب به صورت آنلاین و در محیط مرورگر اجرا می‌شود و برای کامپایل کردن کدها و آموزش شبکه نیازی به سخت‌افزار کامپیوتر کاربر ندارد.

مشخصات و توان پردازشی که گوگل کلاب در اختیار کاربر قرار می‌دهد در شکل ۴-۳ نشان داده شده است.

Parameter	Google Colab
GPU	Nvidia K80 / T4
GPU Memory	12GB / 16GB
GPU Memory Clock	0.82GHz / 1.59GHz

شکل ۴-۳: مشخصات و توان پردازشی Google Colab

در استفاده از این سرویس چند مشکل اساسی وجود داشت که ادامه پروژه با آن را ناممکن می‌کرد:

[67] Google Colaboratory (Colab)

- لزوم آنلاین بودن در تمام مدت کامپایل: در تمام مدت چند ساعته کامپایل کامپیوتر باید به اینترنت وصل باشد و قطع شدن آنی آن به مثابه از دست رفتن تمام نتایج و شروع دوباره است.
- محدودیت زمانی اجرای کدها: در این سرویس کامپایل شدن هر کد نهایتا می‌تواند ۱۲ ساعت طول بکشد. با توجه به توان پردازشی محدود آن امکان آموزش شبکه‌های گن در این مدت زمانی مقدور نیست.
- دسترسی نداشتن مستقیم گوگل کلاب به داده‌های آموزشی: برای این کار لازم است تمام مجموعه آموزش روی ابر گوگل ذخیره شود که این کار علاوه بر زمان‌بر بودن، انعطاف پذیری برای ایجاد تغییر در داده‌های آموزش را بسیار سخت می‌کند.

علاوه بر همه این موارد، مقایسه بین مشخصات کارت گرافیکی کامپیوتر مورد استفاده با اطلاعات سخت‌افزاری سرویس گوگل کلاب که در شکل ۴-۳ نشان داده شده است، نشان از قوی‌تر بودن کارت گرافیکی کامپیوتر بود. لذا به جای این سرویس گزینه پردازش روی کارت گرافیکی که در بخش بعد توضیح داده شده است، انتخاب شد.

٤-٤-٣- پردازش روی کارت گرافیکی

توانایی خارق العاده کارت‌های گرافیک در پردازش موازی، باعث شده در سال‌های اخیر در بسیاری از کاربردهای محاسباتی مثل بلاک چین و شبکه‌های عصبی از آن‌ها استفاده شود. رشد سریع تولید داده و در کنار آن توان پردازشی حاصل از کارت‌های گرافیکی قوی در سال‌های اخیر باعث رشد چشمگیری در هوش مصنوعی و یادگیری ماشین شده است.

از کارت گرافیک کامپیوترهای شخصی نیز می‌توان برای محاسبات سنگین و البته آموزش شبکه‌های عصبی استفاده کرد. استفاده از کارت گرافیک به‌جای پردازنده مرکزی تا حد خیلی زیادی زمان پردازش‌ها را کم می‌کند و پیاده‌سازی آن‌ها را ممکن می‌کند.

گوگل، نسخه ویژه‌ای از تنسورفلو را برای همین کاربرد معرفی کرده و توسعه داده است. استفاده از کارت گرافیک برای پردازش کدها و آموزش شبکه عصبی، کمی پرچالش است. چرا که نسخه‌های مختلفی از تنسورفلو و کتابخانه‌های مورد نیاز آن ارائه شده‌اند که با هم هماهنگ نیستند. نسخه‌های استفاده شده از زبان‌ها، کتابخانه‌ها و نرم‌افزارهای مورد نیاز در این پروژه در جدول زیر نمایش داده شده است.

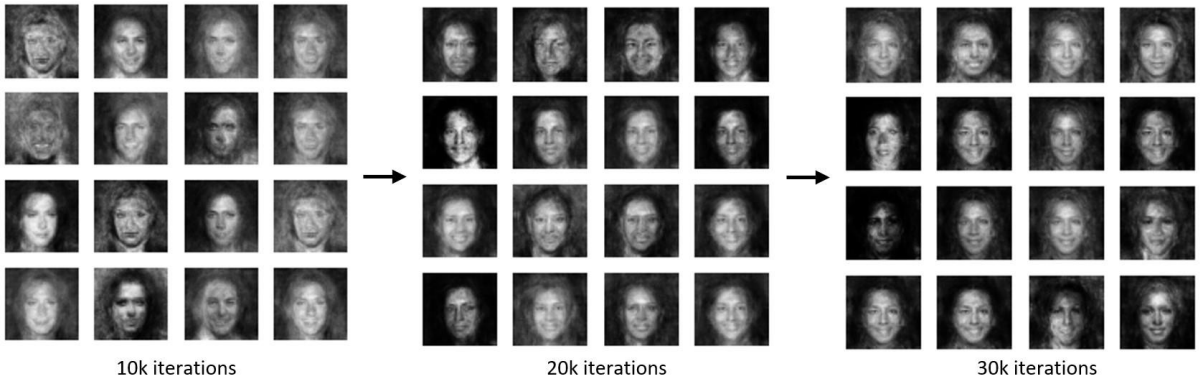
Version	Python version	Compiler	Build tools	cuDNN	CUDA toolkit
tensorflow_gpu-2.3.0	3.6	MSVC 2019	Bazel 3.1.0	7.6	10.1

پس از برطرف کردن چالش هماهنگ کردن نسخه‌های مختلف، امکان پردازش بر روی کارت گرافیک مقدور شد. با استفاده از کارت گرافیک تا حد زیادی محدودیت سخت‌افزاری برای پردازش کدها و آموزش شبکه برطرف شد. اما کماکان توانایی محاسبات سنگین با لایه‌ها و نورون‌های زیاد مقدور نبود. علاوه بر این، محدودیت فضای حافظه موقت امکان استفاده از تمام مجموعه داده را غیرممکن کرد. در بخش بعد تعدیل‌ها و تخفیفاتی که برای میسر شدن امکان شبیه‌سازی انجام گرفت، توضیح داده شده‌اند.

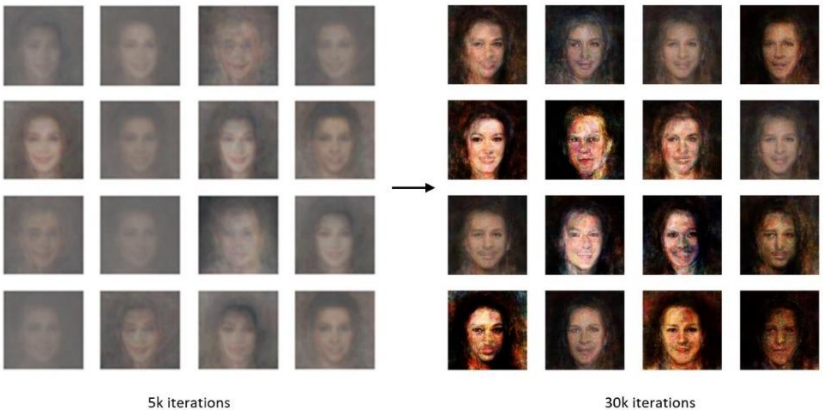
٤-٥- ارائه نتایج

با توان پردازشی کارت گرافیکی موجود تنها امکان آموزش شبکه‌های ساده‌تر خانواده گن میسر بود. بر این اساس پروژه با سه معماری گن معمولی، گن کانولوشنی و گن مشروط انجام شد. در این بخش نتایج و خروجی‌های آن‌ها آورده شده است.

در اولین شبیه‌سازی آموزش با شبکه گن معمولی بر روی مجموعه داده celebrities انجام گرفت. شبکه استفاده شده دارای ۱ لایه پنهان شامل ۱۲۸ نورون است. به دلیل محدودیت حافظه موقت تنها از ۱ درصد مجموعه داده مذکور برای آموزش استفاده شد. نتایج خروجی این شبیه‌سازی در شکل ۴-۴ و ۴-۵ به نمایش درآمده است.

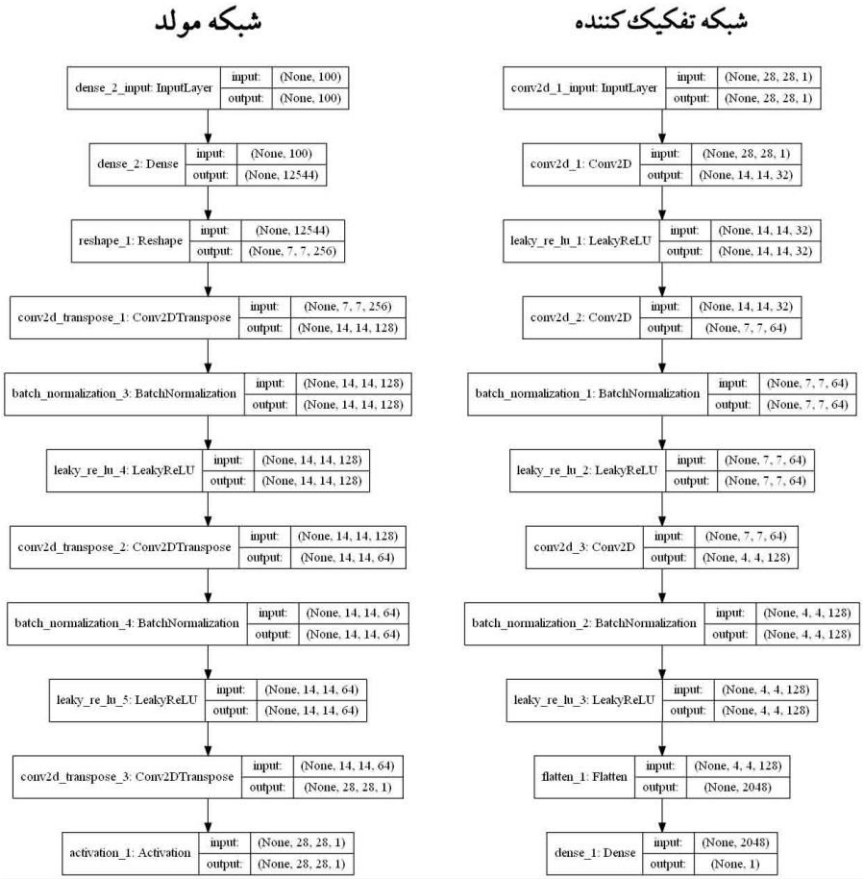


شکل ۴-۴ : تصاویر سیاه و سفید تولید شده توسط مدل پایه گن

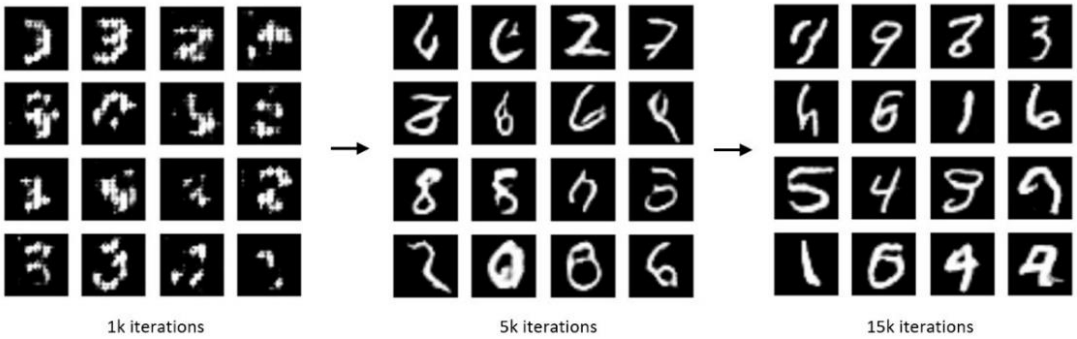


شکل ۴-۵ : تصاویر رنگی (سه کاناله) تولید شده توسط مدل پایه گن

شبیه‌سازی دوم بر روی شبکه گن کانولوشنی انجام گرفت. مشخصات ساختاری این شبکه در شکل ۴-۶ نمایش داده شده است. به دلیل سنگین بودن پردازش در آموزش این شبکه امکان استفاده از مجموعه داده celebrities نبود. به این خاطر از مجموعه ارقام دست‌نویس MNIST برای این شبیه‌سازی استفاده شد. نتایج خروجی‌های این شبکه در شکل ۴-۷ نمایش داده شده است.

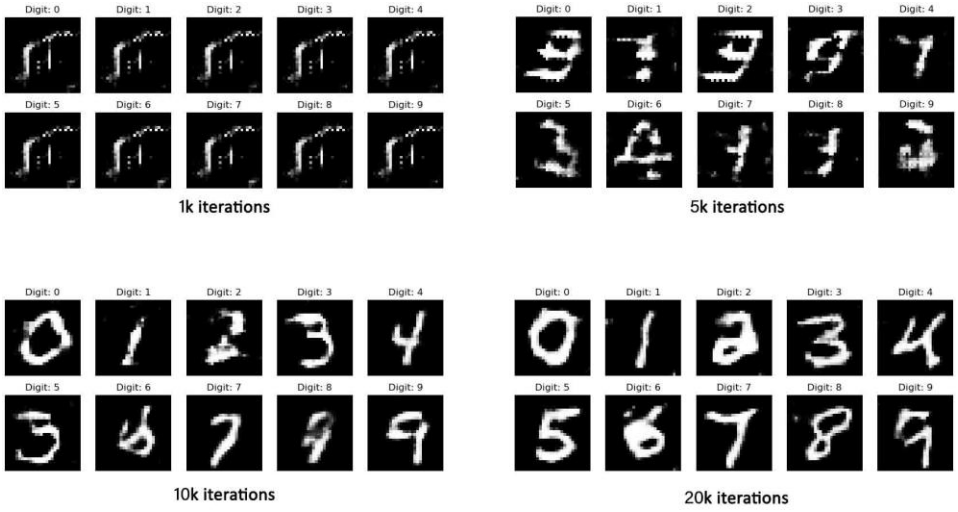


شکل ۴-۶ : ساختار شبکه گن کانولوشنی استفاده شده



شکل ۴-۷ : تصاویر تولید شده توسط شبکه گن کانولوشنی

در شبیه‌سازی سوم، آموزش بر روی شبکه گن مشروط انجام گرفت. به‌دلیل نیاز این معماری به داده‌های برچسب‌دار از مجموعه داده MNIST برای این شبیه‌سازی استفاده شد. در شکل ۴-۸ خروجی‌های این شبیه‌سازی نمایش داده شده است.



شکل ۴-۸ : مجموعه ارقام تولید شده توسط شبکه گن مشروط

در این شبیه‌سازی، آموزش شبکه گن مشروط با وجود اینکه وظیفه دسته‌بندی تصاویر خروجی را نیز بر عهده داشت نسبت به گن کانولوشنی در زمان بسیار کمتری انجام شد. دلیل این امر تعداد زیاد لایه‌های کانولوشنی در گن کانولوشنی است که توان محاسباتی بسیاری را می‌طلبد.

گن کانولوشنی بیشتر در مواردی استفاده می‌شود که تصاویر مجموعه داده ویژگی‌های زیادی داشته باشند و شبکه باید بتواند به طرز مناسبی آن‌ها را بیاموزد. از این لحاظ در مواردی مثل این شبیه‌سازی که بر روی مجموعه داده با ویژگی‌های کم انجام گرفت بهتر است بسته به نیاز از گن معمولی یا گن مشروط استفاده شود.

جمع بندی

در گزارش ارائه شده ابتدا مقدمات و کلیات از جمله مجموعه داده، یادگیری ماشین، یادگیری عمیق، شبکه‌های عصبی، مدل‌های مولد و مدل‌های تفکیک‌کننده توضیح داده شده‌اند. در فصل دوم، در مورد شبکه‌های عصبی متخاصم مولد (گن) و چالش‌ها و مزیت‌ها و کاربردهای آن توضیح داده شده است. همچنین در این فصل در مورد فرآیند آموزش این شبکه و دو بخش آن یعنی بخش مولد و بخش تفکیک‌کننده توضیحاتی ارائه شده است.

در فصل سوم، در مورد پیشرفت‌های شبکه‌های گن و معماری‌هایی که در طی این سال‌ها از زمان معرفی گن اولیه معرفی شده‌اند توضیحاتی ارائه شده است و چالش‌ها و مزیت‌های هر یک توضیح داده شده‌اند. معماری‌هایی که در این فصل معرفی شده‌اند عبارت‌اند از: گن کانولوشنی، گن نیمه نظارتی، گن واسرستاین، گن استایل، Pix2Pix و شبکه‌های گن چرخشی.

در فصل چهارم، محیط برنامه‌نویسی و کتابخانه‌های استفاده شده در پروژه معرفی شده‌اند. همچنین در این فصل مجموعه‌های داده‌ای که در شبیه‌سازی استفاده شده‌اند معرفی شده‌اند. محدودیت‌های سخت‌افزاری مانع پیشبرد پروژه بود، به همین خاطر در ادامه این فصل مجموعه اقدامات و راه‌کارهایی که برای فائق آمدن بر این مشکلات در طی انجام پروژه صورت گرفته‌اند معرفی شده‌اند. در انتهای گزارش مجموعه‌ای از تصاویر تولید شده با معماری‌های گن معمولی، گن کانولوشنی و گن مشروط آورده شده‌اند.

مراجع

١. Goodfellow, I., et al., *Generative adversarial nets.* Advances in neural information processing systems, 2014. **27**: p. 2672-2680.

٢. Barratt, S. and R. Sharma, *A note on the inception score.* arXiv preprint arXiv:1801.01973, 2018.

٣. Zhou, S., et al. *Hype :A benchmark for human eye perceptual evaluation of generative models*. in *Advances in Neural Information Processing Systems*. 2019.

٤. Radford, A., L. Metz, and S. Chintala, *Unsupervised representation learning with deep convolutional generative adversarial networks.* arXiv preprint arXiv:1511.06434, 2015.

٥. Mirza, M. and S. Osindero, *Conditional generative adversarial nets.* arXiv preprint arXiv:1411.1784, 2014.

٦. Odena, A., *Semi-supervised learning with generative adversarial networks.* arXiv preprint arXiv:1606.01583, 2016.

٧. Arjovsky, M., S. Chintala, and L. Bottou, *Wasserstein gan.* arXiv preprint arXiv:1701.07875, 2017.

٨. Gulrajani, I., et al. *Improved training of wasserstein gans*. in *Advances in neural information processing systems*. 2017.

٩. Rubner, Y., C. Tomasi, and L.J. Guibas, *The earth mover's distance as a metric for image retrieval.* International journal of computer vision, 2000. **40**(2): p. 99-121.

١٠. Mehrabi, N., et al., *A survey on bias and fairness in machine learning.* arXiv preprint arXiv:1908.09635, 2019.

١١. Karras, T., et al. *Analyzing and improving the image quality of stylegan*. in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

١٢. Isola, P., et al. *Image-to-image translation with conditional adversarial networks*. in *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

١٣. https://github.com/nvidia/pix2pixHD

.۱۴ https://blogs.nvidia.com/blog/2019/03/18/gaugan-photorealistic-landscapes-nvidia-research

.۱۵ Zhu ,J.-Y., et al. *Unpaired image-to-image translation using cycle-consistent adversarial networks*. in *Proceedings of the IEEE international conference on computer vision*. 2017.

16. https://www.coursera.org
17. https://faradars.org
18. https://developers.google.com
19. https://www.tensorflow.org