

لَكَ الْمُصْرِفُ



دانشگاه شهید باهنر کرمان

دانشگاه شهید باهنر کرمان
دانشکده فنی و مهندسی

پروژه کارشناسی رشته مهندسی کامپیوتر گرایش مهندسی نرم افزار

شبکه های مولد تخاصمی (GAN)

استاد راهنما: جناب آقای دکتر مهدی افتخاری

فاطمه ایرانمنش
فاطمه طاهر

۱۴۰۰ بهار

چکیده

در این پژوهه ما، به بررسی شبکه های مولد متخصص که به اختصار GAN نامیده میشوند خواهیم پرداخت که بر اساس رویکرد تئوری بازی ها بنا شده است.

هم چنین به عنوان کار عملی، مجموعه داده ای از نقوش اسلامی را برای آموزش این گونه از شبکه ها ایجاد می کنیم.

نقوش اسلامی نوعی از نقوش تزئینی به شکل شاخ و برگ گیاهان، گل پیچک یا خطوط صاف و منظم است که گاه با عناصر دیگری همراه می شود.^[۱]

اسلامی گونه ای از نقش و نگار است که شامل خط های پیچیده و منحنی ها و قوس های دورانی مختلف که در تزئینات و کتبیه ها و بعضی دیگر از کار های نقاشی ترسیم می کنند. اسلامی گاهی به صورت نقش اصلی و زمانی همراه با دیگر طرح ها، در معماری، کتاب آرایی و هنر های صناعی به کار رفته است.

طرح های اسلامی انواع گوناگون دارد: «اسلامی ساده»، «اسلامی توپر»، «اسلامی توحالی»، «دهن اژدری»، «خرطوم فیلی»، «ماری»، «اسلامی پیچک» و انواع دیگر. که ما در این پژوهه از نوع اسلامی ساده استفاده کرده ایم.

هدف ما استفاده از این مجموعه داده برای آموزش شبکه و سپس استفاده از شبکه مورد نظر برای تولید طرح هایی مشابه با این نقوش می باشد.

فهرست مطالب

صفحه	عنوان
۲	کلیات پژوهه
۲	۱-۱-۱- ساخت مجموعه داده ای
۲	۱-۱-۱- جمع آوری تصاویر
۲	۱-۱-۱-۱- توضیحات مربوط به crawler
۳	۱-۱-۱-۲- توضیحات مربوط به merge
۳	۱-۱-۱-۳- توضیحات مربوط به downloader
۴	۱-۱-۲-۱- ایجاد مجموعه داده ای
۴	۱-۲-۱- آموزش شبکه
۶	۱-۳-۱- چالش های ایجاد شده
۸	انواع شبکه های مولد متخصص
۸	۱-۲-۱- تاریخچه
۹	۱-۱-۱-۱- مدل تفکیک کننده
۹	۱-۱-۱-۲- معماری تفکیک کننده
۱۰	۱-۲-۱-۲- مدل مولد
۱۰	۱-۲-۱-۳-۱- معماری مولد گن
۱۱	۱-۲-۲- گن ها چگونه کار می کنند.
۱۱	۱-۳-۲- انواع گن
۱۱	Wasserstein GAN - ۱-۳-۲
۱۲	PixelRNN - ۲-۳-۲
۱۲	text-2-image - ۳-۳-۲
۱۵	lsGAN - ۴-۳-۲
۱۶	SrGAN - ۵-۳-۲
۱۷	CycleGAN - ۶-۳-۲
۱۸	StyleGAN - ۷-۳-۲
۲۱	InfoGAN - ۸-۳-۲
۲۱	DiscoGAN - ۹-۳-۲
۲۳	CGAN - ۱۰-۳-۲
۲۳	DCGAN - ۱۱-۳-۲
۲۴	F-GAN - ۱۲-۳-۲
۲۶	نتایج
۲۶	۱-۳- ورودی ها
۲۶	۲-۳- خروجی ها
۳۱	منابع

پیوست

۳۴

فهرست اشکال

صفحه	عنوان
	انواع شبکه های مولد متخصص
۱۰	۱-۲- مولد و تفکیک کننده به عنوان بلوک های سازنده گن
۱۱	۲-۲- مولد و تفکیک کننده به عنوان بلوک های سازنده گن
۱۳	۳-۲- تولید تصویر براساس توصیف متنی
۱۳	۴-۲- استفاده از فرمول بندی برای تولید تصویر
۱۴	۵-۲- تفکیک کننده
۱۵	۶-۲- مجموعه داده آموزشی
۱۵	۷-۲- نتایج حاصل از مدل text-2-image
۱۷	۸-۲- تصاویر تولید شده توسط SRGAN
۱۷	۹-۲- ساختار CycleGAN
۱۸	۱۰-۲- تابع خطای consistency loss
۱۸	۱۱-۲- تصاویر تولید شده توسط CycleGAN
۱۹	۱۲-۲- معماری مولد StyleGAN
۲۱	۱۳-۲- معماری شبکه InfoGAN
۲۲	۱۴-۲- ساختار شبکه DiscoGAN
۲۳	۱۵-۲- ساختار شبکه CGAN
۲۴	۱۶-۲- ساختار شبکه DCGAN
	نتایج
۲۶	۱-۳- نمونه هایی از تصاویر ورودی
۲۷	۲-۳- خروجی در تکرار ۱۰۰۰
۲۷	۳-۳- خروجی در تکرار ۵۰۰۰
۲۸	۴-۳- خروجی در تکرار ۷۰۰۰
۲۸	۵-۳- خروجی در تکرار ۱۴۰۰۰
۲۹	۶-۳- خروجی در تکرار ۲۵۰۰۰
۲۹	۷-۳- خروجی در تکرار ۳۸۰۰۰
۳۰	۸-۳- خروجی در تکرار ۴۶۰۰۰
۳۰	۹-۳- خروجی در تکرار ۵۰۰۰۰

فصل اول

کلیات پروژه

شبکه های مولد متخاصم یکی از جذاب ترین ایده ها در زمینه یادگیری ماشین طی بیست سال اخیر می باشد.

در این سیستم ، دو شبکه عصبی در یک بازی با یکدیگر به رقابت پرداخته و با بهره گیری از یک مجموعه داده آموزشی ، می آموزد تا داده های جدیدی را با ویژگی های مشابه مجموعه آموزشی ، تولید کند. در حال حاضر مجموعه داده های آزمایشی متفاوتی مانند MNIST برای آموزش و کار با این گونه شبکه ها به صورت آماده و از قبل تهیه شده ، وجود دارد. در این پروژه قصد داریم تا با جمع آوری مجموعه تصاویری از نقوش اسلامی، یک مجموعه داده ای جدید از این نقش ها ایجاد کرده و شبکه مولد متخاصم خود را با آن ها آموزش دهیم. در ادامه به بیان جزئیات این روند می پردازیم:

۱- ساخت مجموعه داده ای

در این مرحله از انجام کار به جمع آوری تصاویر و ساخت مجموعه داده ای می پردازیم.

۱-۱-۱- جمع آوری تصاویر

برای این کار ، با نوشتن crawler ، اقدام به جستجو در گوگل با تیتر های متفاوت کرده ، لینک عکس ها را استخراج نموده و آن ها را داخل فایل قرار می دهیم. حال توسط قطعه کد merge یک های تکراری ناشی از جستجوهای متفاوت را حذف کرده و با نوشتن یک script downloader آن ها را دانلود می کنیم. تعداد عکس های به دست آمده از این طریق ۳۷۹ عدد می باشد.

۱-۱-۱- توضیحات مربوط به crawler :
(کد کامل در پیوست شماره ۱)

برای استفاده از مرورگر chrome و تنظیم آن، از کد زیر استفاده می کنیم:
options = webdriver.ChromeOptions()

با تابع زیر به آدرس داده شده می رویم:

```
browser.get("https://www.google.com/search?q=%D9%86%D9%82%D8%B4%20%D9%87%D8%A7%DB%8C%20%D8%A7%D8%B3%D9%84%DB%8C%D9%85%DB%8C%20%D8%A7%DB%8C%D8%B1%D8%A7%D9%86%DB%8C&tbs=isch&hl=en&tbs=rimg:CT6XkwxCQO_1ZYZfDNT10xEz7&bih=1299&biw=2708&client=opera-gx&sa=X&ved=0CBsQuIIBahcKEwiQ6eaGh4PvAhAAAAAHQAAAAAQBg")
```

در قطعه کد زیر تابعی را داریم که با استفاده از آن المان هایی که اسم کلاس^۱ آن ها rg_i می باشد را پیدا کرده و به صورت لیست بر می گرداند:

Class name¹

```
images = browser.find_elements_by_class_name("rg_i")
```

در تگ مربوط به هر یک از عکس های لیست ، ویژگی src آن ها که همان آدرس url می باشد را در متغیر imageUrl قرار می دهیم و آن را به لیست imageUrls اضافه می کنیم:

```
imageUrl = browser.find_elements_by_class_name("n3VNCb")[0].get_attribute("src")
imageUrls.append(imageUrl)
```

با استفاده از () آدرس ها را به رشته های حرفی تبدیل می کنیم:
with open("./imageUrls_"+str(time.time())+"_"+imgNum+"_"+str(len(imageUrls))+".json","w") as f:

```
    json.dump(imageUrls,f)
```

۱-۱-۲- توضیحات مربوط به merge :
(کد کامل در پیوست شماره ۲)

در ابتدا محل فایل json مرحله قبل را پیدا کرده ، سپس محتوای آن را در یک لیست قرار می دهیم:
کد مربوط به یافتن محل فایل:

```
imageUrlsFiles = glob.glob('.\imageUrls\\*.json')
```

کد مربوط به قرار دادن آدرس ها در لیست:

```
imageUrlArrays.append(json.load(f))
```

حال با دو حلقه for ، آدرس های داخل لیست را جدا کرده و در لیست دیگری به نام imageUrlMerged قرار می دهیم:
imageUrlMerged.append(j)

لیست را به آرایه تبدیل کرده و باتابع unique آدرس هایی که تکراری نیستند را جدا می کنیم:
imageUrlMergedNp = np.array(imageUrlMerged)
imageUrlMergedUnified = np.unique(imageUrlMergedNp)

۱-۱-۳- توضیحات مربوط به downloader :
(کد کامل در پیوست شماره ۳)
با قطعه کد زیر هر یک از عکس ها را با استفاده از آدرس آن دانلود می کنیم:
r = requests.get(imageUrl, stream=True)

عکس را تک کاناله می کنیم و آن را به صورت عددی در آرایه arr قرار می دهیم:
grayscale = image.convert("L")
arr = np.array(grayscale)

در مرحله بعد با استفاده از دو حلقه for ، عکس ها را به صورت سیاه و سفید در می آوریم. به این صورت که اگر مقدار آن پیکسل^۲ از مقدار ۱۵۵ بیشتر بود آن را به ۲۵۵ تغییر داده و در غیر این صورت آن را صفر می کنیم:

```
if arr[i][j] >= 155:
    arr[i][j] = 255
```

² منظور در ارایه های آرایه می باشد.

```
else:  
    arr[i][j] = 0
```

حال با استفاده از تابع `fromarray()` اعداد داخل آرایه را خوانده و به عکس تبدیل می کنیم:

```
img = Image.fromarray(arr)
```

و در آخر ، عکس ها را به ابعاد 28×28 تبدیل می کنیم:

```
img.thumbnail((28,28),Image.ANTIALIAS)
```

۱-۱-۲- ایجاد مجموعه داده ای

(کد کامل در پیوست شماره ۴)

در این قسمت ابتدا در یک حلقه `for` تک تک عکس ها را با دستور `imread` از فایل مربوطه خوانده و آن ها را با تابع `IMREAD_GRAYSCALE` به صورت سیاه و سفید درآورده تا کار با آن ها ساده تر شود. سپس ابعاد آن ها را با دستور `resize` به 28×28 تغییر داده و در لیستی به نام `training_data` قرار می دهیم. دستور های مربوطه به صورت زیر می باشند:

```
pic = cv2.imread(os.path.join(path, img) , cv2.IMREAD_GRAYSCALE)  
pic = cv2.resize(pic,(225,225))  
training_data.append(pic)
```

حال یک آرایه به نام `X_train` ایجاد کرده و با یک حلقه عکس ها را از لیست به این آرایه منتقل می کنیم. دستور انتقال به صورت زیر می باشد: (`axis=0` به این معناست که عکس ها را در یک ستون و زیر هم قرار دهد)

```
X_train = np.append(X_train, [training_data[i]], axis=0)
```

۱-۲- آموزش شبکه

(کد کامل در پیوست شماره ۵)

این شبکه نیز مانند سایر شبکه های گن از دو قسمت مولد و تفکیک کننده تشکیل شده است که در ابتدا این دو را تشکیل می دهیم:

در این شبکه ، تابع مولد `LeakyReLU` و `Dense` می باشد ، تشکیل شده است:

```
model.add(Dense(128,input_dim=zdim))  
model.add(LeakyReLU(alpha=0.01))  
model.add(Dense(28*28*1,activation='tanh'))
```

به علاوه تابع تفکیک کننده `build_dis` از چهار لایه که از نوع `Dense` و `Flatten` می باشد ، تشکیل شده است. هم چنین به دلیل وجود دو کلاس (جعلی یا حقیقی بودن)، تابع فعال سازی `sigmoid` می باشد:

```

model.add(Flatten(input_shape=img_shape))
model.add(Dense(128))
model.add(LeakyReLU(alpha=0.01))
model.add(Dense(1,activation='sigmoid'))

```

یک تابع برای ایجاد شبکه گن تعریف کرده (build_gan) و شبکه مولد و تفکیک کننده را به آن اضافه می کنیم:

```

model.add(gen)
model.add(dis)

```

قبل از آموزش شبکه ابتدا باید توابع مولد و تفکیک کننده را کامپایل کنیم:

```

dis_v.compile(loss='binary_crossentropy', optimizer=Adam(), metrics=['accuracy'])
gan_v.compile(loss='binary_crossentropy', optimizer=Adam())

```

برای آموزش شبکه گن ، وزن های تفکیک کننده باید ثابت بمانند:

```
dis_v.trainable=False
```

اکنون به سراغ آموزش شبکه می رویم. در این قسمت پس از نرمال سازی مجموعه داده ایی که قبلا ساخته بودیم ، شبکه را به تعداد تکرارها³ ، آموزش دهیم. برای عکس های واقعی و جعلی ، دو آرایه به عنوان مرجع (هدف) برای آموزش شبکه تفکیک کننده تعریف می کنیم:

```

real = np.ones((batch_size,1))
fake = np.zeros((batch_size, 1))

```

با استفاده از تابع () random.randint() ، از بین کل عکس ها به تعداد اندازه دسته⁴ ، عدد تصادفی تولید کرده و از آرایه ، عکس ها با این اندیس را استخراج می کنیم:

```

ids = np.random.randint(0,Xtrain.shape[0],batch_size)
imgs = Xtrain[ids]

```

سپس یک نویز اولیه ایجاد می کنیم تا مولد با استفاده از این ورودی های نویزی ، تصاویر جعلی را تولید کند:

```

z=np.random.normal(0,1,(batch_size,100))
gen_imgs = gen_v.predict(z)

```

قبل از آموزش کل شبکه ، تابع تفکیک کننده را یک بار با تصاویر واقعی و بار دیگر با تصاویر جعلی آموزش می دهیم. این کار کمک می کند که برسی کنیم آیا مدل ما روی داده های واقعی و جعلی خوب عمل می کند یا خیر:

```

dloss_real = dis_v.train_on_batch(imgs,real)
dloss_fake = dis_v.train_on_batch(gen_imgs, fake)

```

بعد از هر کدام از آموزش های بالا ، یک آرایه دو مقداره بازگردانده می شود که اولین مقدار آن نشان دهنده خطای تشخیص (loss) و دومین مقدار نشان دهنده دقت تشخیص^۵ می باشد. در این جا مقادیر متناظر هر یک از آموزش ها را با یکدیگر جمع کرده و در متغیر های مربوطه قرار می دهیم:

$$\text{dloss,accuracy} = 0.5 * \text{np.add(dloss_real,dloss_fake)}$$

اینک یک ورودی نویزی را ایجاد کرده و آن را به عنوان ورودی اصلی جا زده و شبکه گن را با آن آموزش می دهیم. در پایان خطای نهایی را در متغیر gloss قرار می دهیم:
 $\text{z} = \text{np.random.normal(0, 1, (batch_size, 100))}$
 $\text{gloss} = \text{gan_v.train_on_batch(z,real)}$

در آخر ، در هر ۱۰۰۰ تکرار ، خطاهای دقت عملکرد را نمایش می دهیم:
 $\text{print("%d [D loss: %f , acc: %.2f] [G loss: %f]" %}$
 $\text{(iteration+1,dloss,100.0*accuracy,gloss))}$

۱-۳- چالش های ایجاد شده

چالش هایی که در این پروژه با آن ها مواجه شدیم به صورت زیر می باشد:
 اولین چالش ما جمع آوری عکس ها از داخل سایت گوگل بود که این مشکل را با نوشتن crawler برطرف کردیم.
 دومین چالش ما به این دلیل بود که عکس های دانلود شده برای تهیه مجموعه داده ای، دارای اندازه های متفاوت و هم چنین تعدادی به صورت رنگی و تعدادی سیاه و سفید بودند، بنابراین مجموعه داده ایی که با آن ها ایجاد کردیم مانند مجموعه داده ای MNIST نشده و آموزش شبکه گن با مشکل رو به رو می شد که برای حل این مشکل، ما ابتدا عکس ها را به صورت grayscale در آورده و سپس اندازه آن ها را به $28*28$ تغییر داده و در آخر مجموعه داده ای را ایجاد کردیم.

فصل دوم

انواع شبکه های مولد متخصص

در این فصل، مرواری بر تعریف شبکه های مولد متخصص (GAN) و انواع آن خواهیم داشت.

۱-۲ - تاریخچه

صریح‌ترین الهام برای گن‌ها تخمین مقابله‌ای نویز بود^[۲]، که از همان تابع هزینه گن‌ها استفاده می‌کرد و «گودفلوو»^۶ در دوره دکترای خود در سال‌های ۲۰۱۴-۲۰۱۰ آن را مطالعه کرد.^[۳] افراد دیگر نیز ایده‌های مشابهی داشتند اما آن‌ها را توسعه ندادند. در سال ۲۰۱۰، ایده‌ای شامل شبکه‌های خصمانه^۷ در یک پست وبلاگ توسط «اولی نیمیتالو»^۸ منتشر شد.^[۴] این ایده هرگز پیاده‌سازی نشد و عامل تصادفی بودن^۹ را در مولد خود نداشت، بنابراین یک مدل مولد^{۱۰} نبود. امروزه این گن، به عنوان گن شرطی^{۱۱} یا CGAN شناخته می‌شود.^[۵] ایده‌ای مشابه گن‌ها برای مدل‌سازی رفتار حیوانات توسط «لی»^{۱۲}، «گاوی»^{۱۳} و «گروس»^{۱۴} در سال ۲۰۱۳ استفاده شد.^[۶] یادگیری ماشین متخصص^{۱۵} علاوه بر مدل‌سازی مولد، کاربردهای دیگری نیز دارد و می‌تواند در مدل‌هایی غیر از شبکه‌های عصبی^{۱۶} نیز به کار رود. در سال ۲۰۱۷، از گن برای بهبود تصویر با تمرکز بر بافت‌های واقعی به جای دقت بر روی پیکسل‌ها استفاده شد که کیفیت بالاتری را برای تصویر با بزرگنمایی زیاد تولید کرد.^[۷] در سال ۲۰۱۷، اولین چهره‌ها تولید شدند. آن‌ها در فوریه ۲۰۱۸ در «گرند پلی»^{۱۷} به نمایش درآمدند.^[۸] چهره‌های تولید شده توسط «styleGAN»^{۱۸} در سال ۲۰۱۹ با دیپ فیک‌ها^{۱۹} قرار گرفتند.^[۹] از سال ۲۰۱۷، فناوری گن با پیدایش یک پیاده‌سازی جدید توسعه یافت که از آستانه توانایی تولید نقاشی‌های انتزاعی منحصر به فرد و جذاب عبور کرد و حضور خود را در عرصه هنرهای زیبا اعلام کرد که «گن»^{۲۰} مخفف «شبکه خلاقانه خصمانه» ترجمه می‌شود.^[۱۱] برای ایجاد نقاشی «ادموند دی بلامی»^{۲۱} در سال ۲۰۱۸ از یک سیستم گن استفاده شد که به قیمت ۴۳۲,۵۰۰ دلار آمریکا فروخته شد.^[۱۲]

Goodfellow ^۶	adversarial networks ^۷
Olli Niemitalo ^۸	Stochasticity ^۹
Generative model ^{۱۰}	conditional GAN ^{۱۱}
Li ^{۱۲}	Gauci ^{۱۳}
Gross ^{۱۴}	adversarial machine learning ^{۱۵}
Grand Palais ^{۱۷}	neural networks ^{۱۶}
StyleGAN ^{۱۸}	Deepfakes ^{۱۹}
CAN ^{۲۰}	Edmond de Belamy ^{۲۱}

در مقاله‌ای در اوایل سال ۲۰۱۹ توسط اعضای تیم اصلی گن، پیشرفت بیشتر با این سیستم مورد بحث قرار گرفت و همچنین چشم‌انداز کلی یک هنر مجهر به هوش مصنوعی، مورد توجه واقع شد.^[۱۳] در ماه مه سال ۲۰۱۹، محققان سامسونگ یک سیستم مبتنی بر گن را نشان دادند که تنها با دادن یک عکس از شخص به سیستم، فیلم‌های در حال صحبت آن شخص را تولید می‌کند.^[۱۴] در آگوست ۲۰۱۹، یک مجموعه داده بزرگ متشکل از ۱۲،۱۹۷ آهنگ MIDI هر کدام با متن و تنظیم مlodی جفت شده برای مولد مlodی عصبی از متن با استفاده از GAN-LSTM شرطی ایجاد شد.^[۱۵] در ماه مه سال ۲۰۲۰، محققان انویدیا²² به یک سیستم هوش مصنوعی (اصطلاحاً «گیم گن»²³) آموزش دادند تا بازی پکمن²⁴ را به سادگی و با تماشای بازی، از نو بسازد.^[۱۶]

یک شبکه مولد خصمانه (گن) یک کلاس از چارچوب‌های²⁵ یادگیری ماشین است. دو شبکه عصبی در یک بازی به رقابت با یکدیگر می‌پردازند که در این رقابت، سود یک عامل، ضرر عامل دیگر است. با در اختیار داشتن یک مجموعه آموزشی، این تکنیک می‌آموزد که داده‌های جدیدی را با همان آمار مجموعه آموزشی تولید کند. به عنوان مثال، گن آموزش دیده بر روی عکس‌ها، می‌تواند عکس‌های جدیدی تولید کند که از نظر بینندگان انسانی (حداقل در نگاه اول) معتبر به نظر رسیده و دارای بسیاری از ویژگی‌های واقعی هستند. گرچه در ابتدا به عنوان شکلی از مدل مولدی برای یادگیری بدون نظارت²⁶ پیشنهاد شده‌اند، اما گن‌ها برای یادگیری نیمه نظارت شده²⁷[۱۷]، کاملاً تحت نظارت²⁸[۱۸] و تقویتی²⁹[۱۹] نیز مفید واقع شده‌اند.

شبکه‌های مولد خصمانه از دو مدل تشکیل شده است: مولد و تفکیک کننده.

۱-۱-۲- مدل تفکیک کننده: مدل تفکیک کننده مانند یک طبقه‌بندی کننده باینری عمل می‌کند که قادر است تصاویر را در دسته‌های مختلف طبقه‌بندی کند. این مدل تعیین می‌کند که آیا یک تصویر واقعی و از یک مجموعه داده مشخص است یا به طور مصنوعی تولید شده است.^[۲۰]

۱-۱-۲- معناری تفکیک کننده: تفکیک کننده خواهان انجام کار خود به بهترین شکل ممکن است. هنگامی که یک نمونه جعلی (که توسط مولد تولید می‌شود) به تفکیک کننده داده می‌شود، وی می‌خواهد آن را جعلی بنامد اما مولد می‌خواهد به گونه‌ای نمونه‌ها را تولید کند تا تفکیک کننده اشتباه کرده و آن را به عنوان یک نمونه اصلی اعلام کند. به تعبیر دیگر، مولد در قلاش است که تفکیک کننده را گول بزند.

Nvidia²²

GameGAN²³

Pac-Man²⁴

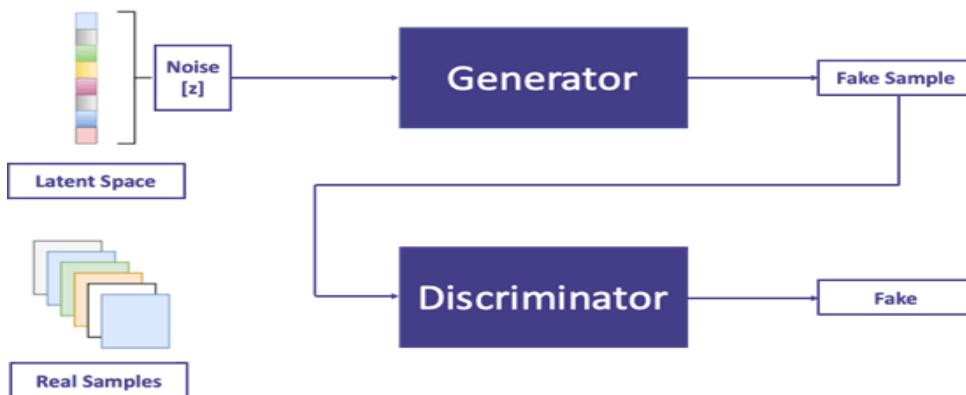
Frameworks²⁵

Unsupervised²⁶

semi-supervised²⁷

fully supervised²⁸

Reinforcement²⁹



شکل ۱-۲
مولد و تفکیک کننده به عنوان بلوک‌های سازنده گَن [۲۱]

برای بھینه‌سازی، از فرمول بھینه سازی حداقل-حداکثر^{۳۰} استفاده می‌شود که در آن، مولد می‌خواهد تابع هدف (objective function) را به حداقل برساند در حالی که تفکیک کننده می‌خواهد همان تابع هدف را، به حداکثر برساند. [۲۱]

۱-۲-۲-۱-۲- مدل مولد:

مدل تفکیک کننده سعی می‌کند با توجه به ویژگی‌های خاص، کلاس خاصی را پیش‌بینی کند. مدل مولد سعی می‌کند ویژگی‌های ارائه شده در کلاس‌ها را پیش‌بینی کند. این شامل تعیین احتمال ویژگی داده شده به یک کلاس است.

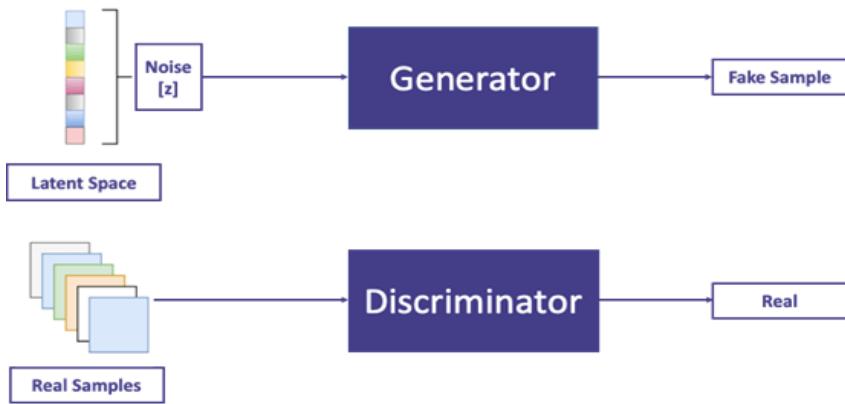
ایده اصلی گَن‌ها مبتنی بر آموزش «غیرمستقیم» از طریق تفکیک کننده است که خود نیز به صورت پویا در حال بهروزرسانی است. این اساساً به این معنی است که مولد آموزش ندیده است تا فاصله از یک تصویر خاص را به حداقل برساند، بلکه تمایل دارد تا تفکیک کننده را گول بزند. این کار مدل را قادر می‌سازد تا با روش بدون نظارت یاد بگیرد. [۲۰]

۱-۲-۱-۲- معماری مولد گَن:

مولد با تولید یک نویز تصادفی (نمونه‌برداری شده از یک فضای پنهان) نمونه‌های مصنوعی را تولید می‌کند و تفکیک کننده بین نمونه واقعی (خروجی با مقدار ۱) یا جعلی (خروجی با مقدار ۰) بودن نمونه ورودی تفاوت قائل می‌شود.

نمونه تولید شده توسط مولد، یک نمونه جعلی نامیده می‌شود. همانطور که در شکل ۱-۲ و شکل ۲-۲ مشاهده می‌کنید، هنگامی که یک داده از مجموعه داده‌های آموزشی به عنوان ورودی به تفکیک کننده داده می‌شود، آن را یک نمونه واقعی می‌خواند در حالی که داده دیگر را که توسط مولد تولید شده است، جعلی می‌نامد. [۲۱]

^{۳۰} min-max optimization function



شکل ۲-۲
مولد و تفکیک کننده به عنوان بلوک‌های سازنده گن [۲۱]

۲-۲- گن‌ها چگونه کار می‌کنند.

همان طور که در قسمت قبل گفته شد شبکه گن دارای دو قسمت مولد و تفکیک کننده می‌باشد. یک مولد نمونه‌های جدیدی از یک شی را تولید می‌کند در حالی که تفکیک کننده تعیین می‌کند که آین نمونه جدید به مجموعه داده واقعی تعلق دارد یا خیر. فرض کنیم که یک مجموعه داده حاوی تصاویر کفش داریم و می‌خواهیم کفش‌های «جعلی» تولید کنیم. نقش مولد تولید کفش‌های جدید است در حالی که هدف تفکیک کننده این است که تصاویر تولید شده توسط مولد را جعلی تشخیص دهد.

در طی فرایند آموزش، وزن‌ها و پایاس‌ها از طریق پس‌انتشار^{۳۱} تعدیل می‌شوند تا زمانی که تفکیک کننده یاد بگیرد که تصاویر واقعی کفش را از تصاویر جعلی تشخیص دهد. مولد از تفکیک کننده باز خورد می‌گیرد و از آن برای تولید تصاویر واقعی تر استفاده می‌کند. شبکه تفکیک کننده یک شبکه عصبی کانولوشن است که تصاویر را به صورت جعلی یا واقعی طبقه‌بندی می‌کند. مولد تصاویر جدید را از طریق یک شبکه عصبی ضد کانولوشن^{۳۲} تولید می‌کند. [۲۰]

۲-۳- انواع گن:

برای درک عمیق‌تر گن‌ها، انواع مختلف شبکه‌های مولد خصمانه را بررسی کرد و شیوه کارکرد و نحوه استفاده از آن‌ها را بیان خواهیم کرد:
:Wasserstein GAN - ۱-۳-۲

این گن توسط مارتین آریوفسکی³³ و همکاران در سال ۲۰۱۷ معرفی شد و یک چارچوب نظری جدید برای گن‌ها، که مبتنی بر یافته‌های تجربی است، معرفی کرد.³⁴ با استفاده از این چارچوب جدید، WGAN را ایجاد کردند که تقریب فاصله واسراشتاین³⁴ را به حداقل می‌رساند.³⁵ این نوع، گرادیان بهتری در تفکیک کننده ایجاد می‌کند که مولد می‌تواند راحت‌تر از آن بیاموزد.

back-propagation³¹
de-convolutional³²

Martin Arjovsky et. al.³³
فاصله بین دو توزیع احتمال³⁴

در این مدل جدید، نشان می‌دهیم که می‌توانیم ثبات یادگیری را بهبود بخشیم، از مشکلاتی مانند «حالت فروپاشی»³⁵ خلاص شویم.
کاربرد آن در هنر:

کنی جونز³⁶ و دریک بونافیلیا³⁷، دانشجویان علوم کامپیوتر بودند که به گن و اسراشتاین (WGAN) برخورد کردند و به این نتیجه رسیدند که این گن، ابزاری عالی برای یادگیری سبک هنری و تولید هنر جدید است.

متاسفانه، گن‌ها به شکل بدی ناپایدار هستند. ولی خوشبختانه، تغییرات اعمال شده توسط این افراد روی و اسراشتاین، باعث شد تا گن‌ها به اندازه کافی پایدار شوند و بتوانند هنری قابل شناسایی تولید کنند. این تیم از پایگاه داده ویکی‌آرت³⁸ متشکل از ۱۰۰,۰۰۰ نقاشه برچسب‌دار به عنوان داده‌های آموزشی استفاده کردند.

برای به حد اکثر رساندن عملکرد مدل، این تیم یک مولفه اضافی را به تفکیک کننده برای پیش‌بینی ژانر نقاشی‌ها اضافه کرد و برای جلوگیری از تمکز زیاد مدل بر روی «واقعی» و «جعلی» بودن تصاویر تولید شده، از پیش‌آموزش استفاده و شرطی سازی را اضافه کرد. این امر باعث شد تا تفکیک کننده بتواند در ک درستی از تفاوت بین سبک‌های هنری داشته باشد.^[۲۴]

۳-۲-۲- شبکه‌های PixelRNN:

در این مدل، تعدادی از تکنیک‌ها برای تولید تصویر را باهم ترکیب کردند که در نتیجه، چندین مدل مولد جدید (PixelCNN) و دو نوع PixelRNN در این کار معرفی شد.^[۲۵] این مدل‌ها با استفاده از نوآوری‌های جدید و تلفیق هوشمندانه تکنیک‌های موجود، وضعیت هنر را در تولید تصویر بسیار بهبود بخشید که یک کلاس پیشرفته در ساخت مدل‌های مولد هستند.

در عصر رسانه‌های اجتماعی، تصاویر بسیاری وجود دارد. اما یادگیری طبیعی توزیع تصاویر در محیطی بدون نظارت بسیار دشوار است. PixelRNN نمونه‌ای از مدل مولد خود-بازگشتی⁴⁰ بوده و قادر به مدل کردن توزیع احتمالی گستته تصویر و پیش‌بینی پیکسل تصویر در محیط دو بعدی می‌باشد.

در مقایسه با گن‌ها، مدل‌های خود-بازگشتی مانند PixelRNN توزیع داده‌ای صریح را یاد می‌گیرند در حالیکه گن‌ها توزیع احتمال ضمنی را می‌آموزند. به همین دلیل گن صریحاً توزیع احتمال را آشکار نمی‌کند بلکه به ما اجازه می‌دهد تا مشاهده خود را از توزیع احتمال آموخته شده، نمونه‌برداری کنیم. در یک تصویر، به طور کلی یک پیکسل به تمام پیکسل‌های پیش‌بینی شده قبلی بستگی دارد که منجر به ایجاد دامنه وابستگی زیادی در میان پیکسل‌ها می‌شود. همانطور که RNN‌ها ثابت کردند که در پیش‌بینی سری‌های زمانی که به دامنه وابستگی زیادی دارند، موثرند. بنابراین، RNN برای تولید پیکسل در تصویر استفاده می‌شود.^[۲۱]

۳-۳-۲- text-2-image :

شبکه‌های مولد تخاصمی برای ایجاد تصاویر تصادفی مناسب هستند. به عنوان مثال، یک گن که برای ایجاد تصاویر گربه‌ها آموزش دیده، می‌تواند تصاویر تصادفی از یک گربه ایجاد کند. اما الگوهای رنگی تصویر گربه می‌تواند طیف گسترده‌ای را شامل شود. بنابراین، تصاویر تصادفی برای حل مشکلات موارد تجاری مفید نیستند. در حال حاضر، استفاده از گن برای ایجاد تصاویر مبتنی بر تصور ما، فعالیت بسیار سختی است.

در این قسمت، در مورد معماری گن که پیشرفت عمدتی در ایجاد تصاویر بامعنی، مبتنی بر توصیف دقیق متن، صحبت می‌کنیم. این فرمول گن یک توصیف متنی را به عنوان ورودی دریافت می‌کند و یک تصویر RGB بر اساس توضیحات داده شده، تولید می‌کند.^[۲۱]

mode collapse³⁵

Kenny Jones³⁶

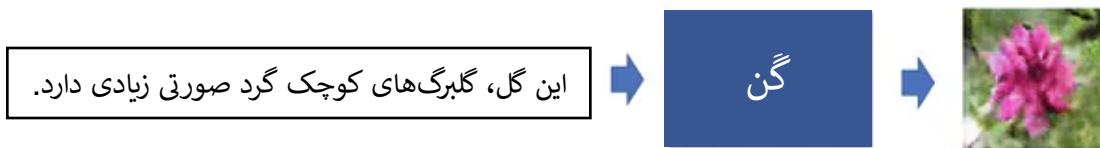
Derrick Bonafilia³⁷

WikiArt³⁸

Recurrent Neural Networks³⁹

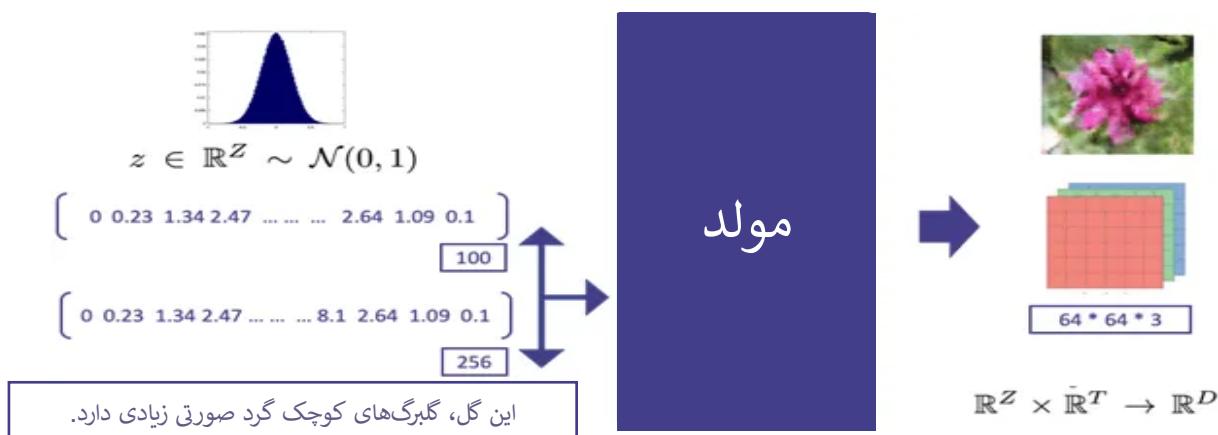
auto-regressive⁴⁰

به عنوان مثال (شکل ۳-۲)، توصیف متنی «این گل، گلبرگ‌های کوچک گرد صورتی زیادی دارد.» را به عنوان ورودی دریافت و تصویری از یک گل با گلبرگ‌های صورتی گرد ایجاد می‌کند.

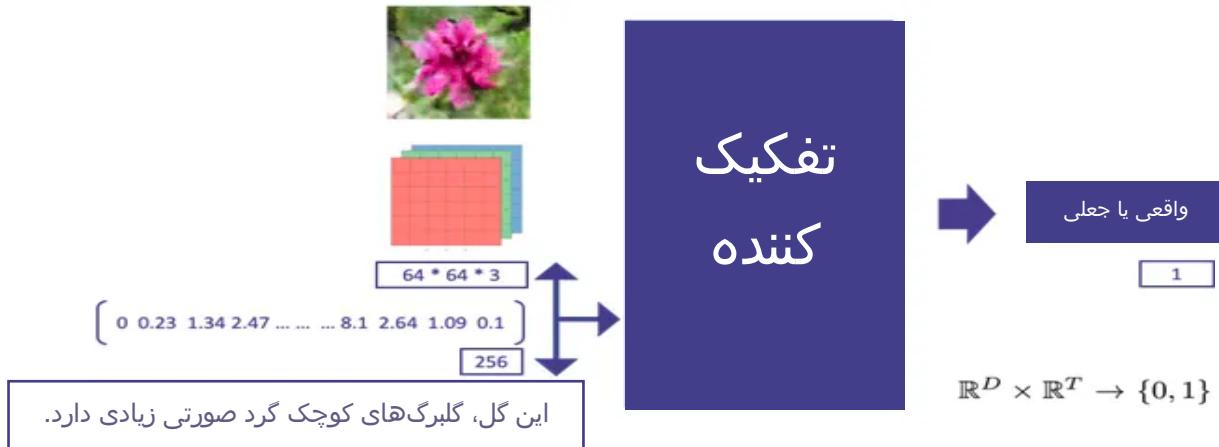


شکل ۳-۲
تولید تصویر براساس توصیف متنی [۲۱]

به عنوان مثال، برای این کار، توصیف متنی به یک تعبیه^{۴۱} ۲۵۶ بعدی تبدیل شده و با یک بودار نویز ۱۰۰ بعدی الحق می‌شود. این فرمول بندی به مولد کمک می‌کند که به جای ایجاد تصاویر تصادفی، تصاویری که با توصیف ورودی متناسب است، ایجاد کند. (شکل ۴-۲)



جفت مربوط به تصویر واقعی و متن واقعی به این دلیل آورده شده تا مدل یاد بگیرد که تصویر با متن مدنظر تطابق داشته باشد. تصویر اشتباه و متن واقعی بدین منظور آورده شده که مدل تشخیص دهد تصویر مطابق با توصیف متنی نیست. در این حالت، متغیر هدف بر روی صفر تنظیم می‌شود تا مدل یاد بگیرد که تصویر و متن داده شده با یکدیگر تابعی ندارند. در حالت سوم، تصویر جعلی به معنای تصویری است که توسط مولد تولید می‌شود. در این حالت، متغیر هدف بر روی صفر تنظیم شده است تا مدل تفکیک کننده بتواند بین تصاویر جعلی و واقعی تمایز قائل شود.^[۲۶]



شکل ۵-۲
تفکیک کننده [۲۱]

شکل ۶-۶ مجموعه داده‌های آموزشی استفاده شده دارای یک تصویر به همراه ۱۰ توضیح متنی مختلف است که خصوصیات تصویر را توصیف می‌کند.

۱. گلبرگ‌ها بلند و بنفش هستند و نوکشان بیضی‌شکل است.
۲. این گل صورتی رنگ است با گلبرگ‌های بیضی‌شکل .
۳. گلبرگ‌های بلند و باریک صورتی با یک مادگی سبک و به رنگ بنفش تیره و زرد .
۴. این گل گلبرگ‌های بنفش دارد که با بسک زرد و رشته آبی احاطه شده است .
۵. این گل، گلبرگ‌های دراز و کشیده دارد با بسک زرد در وسطش .
۶. یک گل بزرگ مخلع روشن با یک مرکز طلای-بنفش .
۷. این گل یک نک لایه از گلبرگ‌های کوچک بنفش مستطیلی دارد .
۸. این گل، گلبرگ‌های صورتی دارد که کلاله‌هایی زرد دارند .
۹. گلبرگ‌های صورتی روشن این گل، لبه‌های جهت‌دار دارند که گرد می‌باشند و با گروهی از پرچم‌های کوتاه در بالای آنچن صورتی روشن قرار گرفته‌اند .
۱۰. گلبرگ‌های صاف و صورتی، صفحه‌ای گرد از پرچم‌ها را با بسکی طلای‌رنگ، شکل داده‌اند.



شکل ۶-۲
مجموعه داده آموزشی [۲۱]

شکل ۷-۲ برخی از نتایج حاصل از یک مدل text-2-image آموزش دیده را نشان می‌دهد:



این گل، گلبرگ‌های بنفش دارد.



من گل میخواهم که گلبرگ‌های قرمز داشته باشد.



رنگ گلبرگ‌ها زرد می‌باشد.



این گل صورتی رنگ است و گلبرگ‌های گردی دارد.

شکل ۷-۲

نتایج حاصل از مدل [۲۱] text-2-image

:lsGAN -۴-۳-۲

در چند وقت اخیر، شبکه‌های مولد خصمانه عملکرد چشمگیری را برای کارهای بدون نظارت نشان داده‌اند.

در GAN معمولی، تفکیک کننده ازتابع خطای آنتروپی-متقطع⁴² استفاده می‌کند که گاهی اوقات منجر به محوشدگی گرادیان⁴³ می‌شود. به جای آن lsGAN پیشنهاد می‌کند که ازتابع خطای مجموع کمترین مربعات⁴⁴ برای تفکیک کننده استفاده کند. این فرمول کیفیت بالاتری از تصاویر تولید شده توسط گن را فراهم می‌کند.

پیش از این، در گن، از بهینه سازی حداقل-حداکثر صحبت کردیم که در آن تفکیک کننده یک طبقه‌بندی کننده باینری است و در حین بهینه سازی ازتابع خطای آنتروپی-متقطع سیگموئید استفاده می‌کند.

(۱-۲) $\min_G \max_D V_{GAN}(D, G) = \mathbb{E}_{x \sim P_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim P_z(z)}[\log(1 - D(G(z)))]$ همانطور که قبلاً ذکر شد، غالباً این فرمول باعث مساله محوشدگی گرادیان نقطه‌ای از داده می‌شود که در سمت صحیح مرز تصمیم‌گیری قرار دارند اما از منطقه تراکم بسیار فاصله دارند. فرمول حداقل مربعات به این موضوع پرداخته و یادگیری پایدارتری از مدل و تولید تصاویر بهتر را فراهم می‌کند. در زیر فرمول بهینه‌سازی شده برای lsGAN آمده است:

- a برچسب نمونه جعلی است.
- b برچسب نمونه واقعی است.
- c ارزشی را نشان می‌دهد که مولد می‌خواهد تفکیک کننده برای یک نمونه جعلی باور کند.

$$\min_D V_{LSGAN}(D) = \frac{1}{2} \mathbb{E}_{x \sim P_{data}(x)}[(D(x) - b)^2] + \frac{1}{2} \mathbb{E}_{z \sim P_z(z)}[(D(G(z)) - a)^2] \quad (2-2)$$

$$\min_G V_{LSGAN}(G) = \frac{1}{2} \mathbb{E}_{z \sim P_z(z)}[(D(G(z)) - c)^2] \quad (3-2)$$

اکنون، ما دوتابع خطای منحصر به فرد داریم که در حال بهینه‌سازی هستند. یکی با توجه به تفکیک کننده به حداقل رسیده و دیگری با توجه به مولد به حداقل می‌رسد.^[۲۷]

lsGAN در مقایسه با گن معمولی یک مزیت بزرگ دارد. در گن معمولی، همانطور که تفکیک کننده ازخطای آنتروپی-متقطع باینری استفاده می‌کند، تا زمانی که یک مشاهده سمت صحیح مرز تصمیم‌گیری باشد، خطابرای آن مشاهده صفر است.

⁴² cross-entropy

⁴³ vanishing gradient

⁴⁴ least-squares

اما در مورد lsGAN، اگر یک مشاهده از مرز تصمیم‌گیری فاصله زیادی داشته باشد، حتی اگر در سمت صحیح مرز تصمیم‌گیری باشد، این مدل آن را جریمه می‌کند.
این جریمه کردن، مولد را مجبور می‌کند تا نمونه‌ها را نزدیک به مرز تصمیم‌گیری تولید کند. همراه با آن، مساله محشودگی گرادیان نیز برطرف می‌شود؛ زیرا هنگام بهروزرسانی مولد، نقاط بسیار دور باعث ایجاد گرادیان بیشتری می‌شوند.^[۲۱]

۵-۳-۲: SrGAN

در اینجا یک GAN به گونه‌ای آموزش داده می‌شود که بتواند با گرفتن تصویر باوضوح پایین، تصویری با رزوشن بالا را ایجاد کند که یکی از کاربردهای آن در بزرگ نمایی^{۴۵} می‌باشد.
معماری SRGAN از سه شبکه عصبی تشکیل شده است: یک شبکه مولد بسیار عمیق، یک شبکه تفکیک کننده و یک شبکه VGG-16 از پیش آموزش دیده.^[۲۸]

SRGAN‌ها ازتابع خطای perceptual استفاده می‌کنند.^[۲۹] هم‌چنین، دوتابع خطای content loss و adversarial loss هم اضافه شده اند تا تصاویر تولید شده طبیعی‌تر باشند و جزئیات دقیق‌تری داشته باشند.

در شکل زیر اولین مقدار سمت راست، خطای content loss را نشان می‌دهد که توسط شبکه 19 VGG به دست می‌آید. این خطای میزان تقاضا بین تصویر بازسازی شده (که توسط مولد ایجاد شده است) و تصویر اصلی باوضوح بالا را نشان می‌دهد.

دومین مقدار سمت راست، خطای adversarial loss می‌باشد که نشان می‌دهد آیا تصاویر تولید شده توسط مولد، قادر به گمراه کردن تفکیک کننده هستند یا خیر.

در شکل ۸-۲ مشاهده می‌کنیم که تصویر تولید شده توسط SRGAN بسیار نزدیک به تصویر اصلی باوضوح بالا می‌باشد:



شکل ۸-۲
[۲۸] SRGAN تصاویر تولید شده توسط

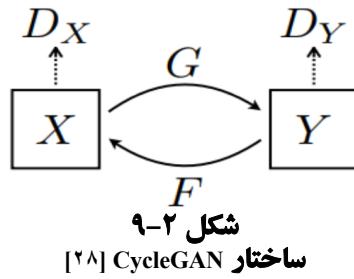
۶-۳-۲: CycleGAN

CycleGAN یک معماری GAN بسیار محبوب است که بیشتر برای تغییر شکل تصاویر شبکه‌های مختلف استفاده می‌شود.^[۳۰] به عنوان مثال با این نوع شبکه می‌توان تصویر اسب را به گورخر تبدیل کرد (یا بالعکس).

برای این کار از دو GAN استفاده شده و مولد هر GAN، عمل تبدیل را انجام می‌دهد. در واقع، اگر ورودی را X در نظر بگیریم، مولد اولین GAN می‌آموزد که تصویر X را به تصویر Y تبدیل کند ($G: X \rightarrow Y$)، بنابراین خروجی آن $Y = G(X)$ خواهد بود. مولد دومین GAN معکوس $F: Y \rightarrow X$ را به تصویر X تبدیل می‌کند ($F: Y \rightarrow X$) و نتیجه آن $X = F(Y)$ می‌باشد. به همین صورت، ما دوتابع تفکیک کننده Dx و Dy خواهیم یافت.

Zoom⁴⁵

تفکیک کننده D_X نشان می دهد که آیا خروجی مولد اول (تولید شده) از تصویر Y اصلی قابل تشخیص می باشد یا خیر. به همین ترتیب، تفکیک کننده D_Y نشان می دهد که خروجی مولد دوم (تولید شده) از تصویر X اصلی قابل تشخیص است یا خیر. این ساختار در شکل ۹-۲ نشان داده شده است:

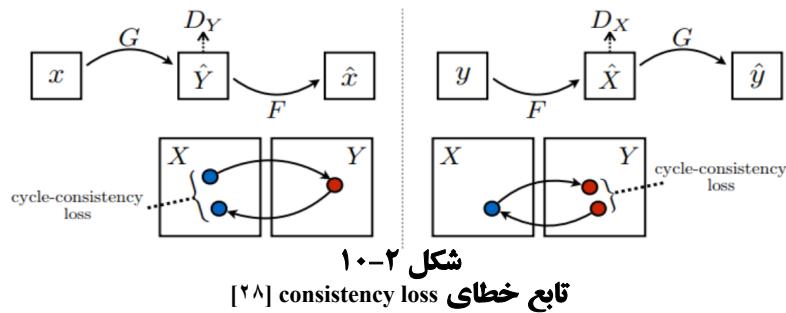


در $CycleGAN$ ، هیچ داده جفت شده ای برای آموزش وجود ندارد (یعنی مثلا عکس اسب به همراه گورخر به شبکه داده نمی شود)، بنابراین هیچ تضمینی برای صحیح بودن ورودی x و هدف y وجود ندارد. به این علت و برای اطمینان از اینکه شبکه نگاشت های صحیح را می آموزد، از تابع خطای consistency loss استفاده می شود. در چرخه این تابع:

- تصویر X توسط مولد G ، به تصویر \hat{Y} تبدیل می شود.
- تصویر \hat{Y} تولید شده، توسط مولد F به تصویر \hat{x} تبدیل می شود.
- میانگین خطای مطلق بین X و \hat{x} محاسبه می شود.

forward cycle consistency loss : $X \rightarrow G(X) \rightarrow F(G(X)) \sim \hat{X}$

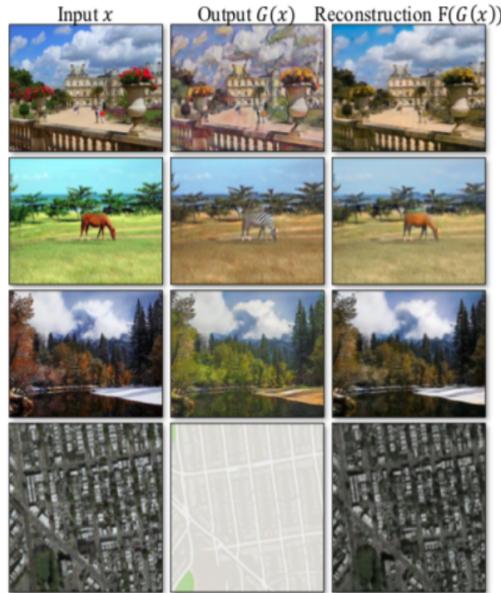
backward cycle consistency loss : $Y \rightarrow F(Y) \rightarrow G(F(Y)) \sim \hat{Y}$



همان طور که در شکل ۱۰-۲ نشان داده شد، مولد G وظیفه تبدیل تصویر X به تصویر Y را دارد. طبق خطای consistency، اگر تصویر Y را به عنوان ورودی به مولد G بدهیم، باید تصویر واقعی Y یا چیزی نزدیک به آن را به عنوان خروجی تولید کند.

$$Identity loss = |G(Y) - Y| + |F(X) - X| \quad (۴-۲)$$

در شکل ۱۱-۲ برحی از موارد موفقیت آمیز آورده شده است:



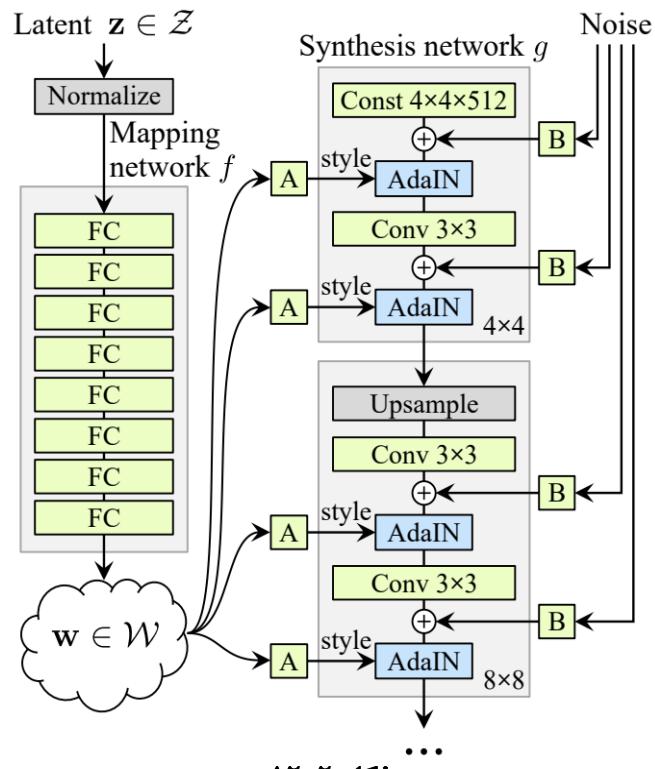
شکل ۱۱-۲
تصاویر تولید شده توسط [۲۸] CycleGAN

۷-۳-۲: StyleGAN

از زمان آغاز به کار، پیشرفت‌های زیادی ایجاد شده است که باعث شده این یک روش پیشرفته برای تولید داده‌های مصنوعی از جمله تصاویر مصنوعی باشد. با این وجود بیشتر این پیشرفت‌ها در بخش تفکیک کننده مدل ایجاد می‌شود که توانایی تولید مولد را بهمود می‌بخشد. این موضوع همچنین نشان می‌دهد که تمرکز زیادی بر روی قسمت مولد وجود ندارد، که باعث عدم کنترل قسمت تولید کننده گن می‌شود. برخی از پارامترها وجود دارد که می‌توانند در قسمت تولید تغییر کند، مانند پس‌زمینه، پیش‌زمینه و سبک یا برای چهره‌های انسانی ویژگی‌های بسیاری مانند ژست، رنگ مو، رنگ چشم و غیره وجود دارد، که می‌توانند در ایجاد تصاویر مختلف تغییر کند.

StyleGAN از معماری گن پیشونده پایه استفاده می‌کند و برخی تغییرات را در قسمت مولد پیشنهاد می‌دهد.^[۳۱] با این حال، معماری تفکیک کننده کاملاً شبیه معماری گن پیشونده پایه است.

شکل ۱۲-۲ خلاصه معماری مولد StyleGAN می‌باشد:



شکل ۱۲-۲
معماری مولد [۲۱] StyleGAN

StyleGAN به عنوان یک معماری تدریجی در حال رشد با پنج تغییر ایجاد شده، توصیف شده است.
لیست افزایشی تغییرات در مولد عبارتند از: [۳۲]

- گن پیشونده پایه

- افزودن تنظیم و نمونه برداری افزایشی دو خطی.

- افزودن شبکه نگاشت و (styles)AdaIN.

- حذف بردار ورودی نهان مولد.

- افزودن نویز به هر بلوک.

- افزودن منظم سازی ترکیبی.

1. گن پیشونده پایه [۳۳]

مدل های مولد و تفکیک کننده StyleGAN با استفاده از روش گن پیشونده در حال رشد، آموزش می بینند.

این بدان معناست که هر دو مدل با تصاویر کوچک شروع می کنند، مثلا تصاویر 4×4 . مدل ها تا زمان پایداری، سازگار هستند. سپس هر دو عامل تفکیک کننده و مولد تا دو برابر عرض و ارتفاع، گسترش می یابند، به عنوان مثال 8×8 .

برای پشتیبانی از اندازه هی بزرگتر تصویر به هر مدل یک بلوک جدید اضافه می شود که به آرامی در حین آموزش ناپدید می شوند. پس از ناپدید شدن، مدل ها دوباره آموزش می بینند تا به ثبات قابل قبولی

برسند و روند با اندازه‌های بزرگتر از تصویر تکرار می‌شود تا اندازه تصویر مورد نظر مانند 1024×1024 برآورده شود.

۲. نمونه‌برداری دو خطی [۳۱]

ما از نمونه‌برداری دو خطی به جای نمونه‌برداری نزدیک ترین همسایه بالا و پایین، هم در مولد و هم در تفکیک کننده استفاده می‌کنیم. ما این نمونه‌برداری دو خطی را به وسیله فیلتر پایین‌گذر فعال ساز، به همراه فیلتر دوجمله‌ای مرتبه دوم قابل تفکیک، بعد و قبل از هر یک از لایه‌های نمونه‌برداری، پیاده‌سازی می‌کنیم.

۳. نگاشت شبکه AdaIN [۳۱]

در مرحله بعد، از یک شبکه نگاشت مستقل استفاده می‌شود که یک نقطه نمونه‌برداری تصادفی از فضای نهان، به عنوان ورودی می‌گیرد و یک بردار استایل تولید می‌کند.

شبکه نگاشت از هشت لایه کاملاً متصل⁴⁶ تشکیل شده است، مانند شبکه عصبی عمیق استاندارد. برای سادگی، ابعاد هر دو فضای (نهان و نیمه نهان) را روی ۵۱۲×۵۱۲ تنظیم می‌کنیم و نگاشت با استفاده از لایه MLP اجرا می‌شود.

بردار استایل پس از لایه‌های کانولوشن از طریق عملیاتی به نام فرمال‌سازی نمونه تطبیقی یا AdaIN، در هر بلوک از مدل مولد تبدیل و ترکیب می‌شود.

لایه‌های AdaIN شامل استاندارد کردن خروجی ویژگی نگاشت به یک استاندارد گوسی⁴⁷ و سپس افزودن بردار استایل به عنوان بایاس است.

تحولات وابسته آموخته شده، سپس (بردار نهان میانی) به استایل‌های $y = (y_s, y_b)$ اختصاص می‌یابد که عملیات عادی سازی تطبیقی (AdaIN) را، پس از هر لایه کانولوشن شبکه‌ی سنتز شده g یا مولد، کنترل می‌کند.

$$AdaIN(x_i, y) = y_{s,i} \frac{x_i - \mu(x_i)}{\sigma(x_i)} + y_{b,i} \quad (5-2)$$

۴. حذف نقطه ورودی نهان

تغییر بعدی شامل اصلاح مدل مولد است به طوری که دیگر نقطه‌ای از فضای نهان به عنوان ورودی نمی‌گیرد. در عوض، برای شروع روند سنتز تصویر، مدل دارای مقدار ثابت $3^{*}4^{*}512$ است.

۵. افزودن نویز [۳۱]

قبل از عملیات AdaIN به هر یک از این نگاشتهای فعال ساز یک نویز گاووسی اضافه می‌شود. نمونه متفاوتی از نویز برای هر بلوک تولید و بر اساس فاکتورهای مقیاس‌گذاری آن لایه تفسیر می‌شود. نمودهای زیادی در چهره افراد وجود دارد که کوچک‌اند و می‌توانند تصادفی باشند، مانند کک و مک، محل دقیق موها، چین و چروک‌ها، ویژگی‌هایی که تصویر را واقعی تر و تنوع خروجی را بیشتر می‌کند. روش معمول برای قرار دادن این ویژگی‌های کوچک در تصاویر گن، افزودن نویز تصادفی به بردار ورودی است.

۶. منظم‌سازی ترکیبی [۳۱]

مولد استایل در هر سطح از شبکه‌ی سنتز، از بردار میانی استفاده می‌کند که ممکن است باعث شود شبکه، ارتباط بین سطوح مختلف را بیاموزد. به منظور کاهش همبستگی، مدل به طور تصادفی دو بردار ورودی $(z1, z2)$ را انتخاب کرده و بردار میانی $(w1, w2)$ را برای آن‌ها تولید می‌کند. سپس برخی از سطوح را با سطح اول آموختش می‌دهد و در یک نقطه تقسیم تصادفی به سطح دیگر تغییر پیدا کرده تا بقیه سطوح نیز آموختش بیینند. این تغییر در نقاط تقسیم تصادفی تضمین می‌کند که شبکه همبستگی را به میزان مناسب یاد می‌گیرد.

۷-۳-۲: InfoGAN

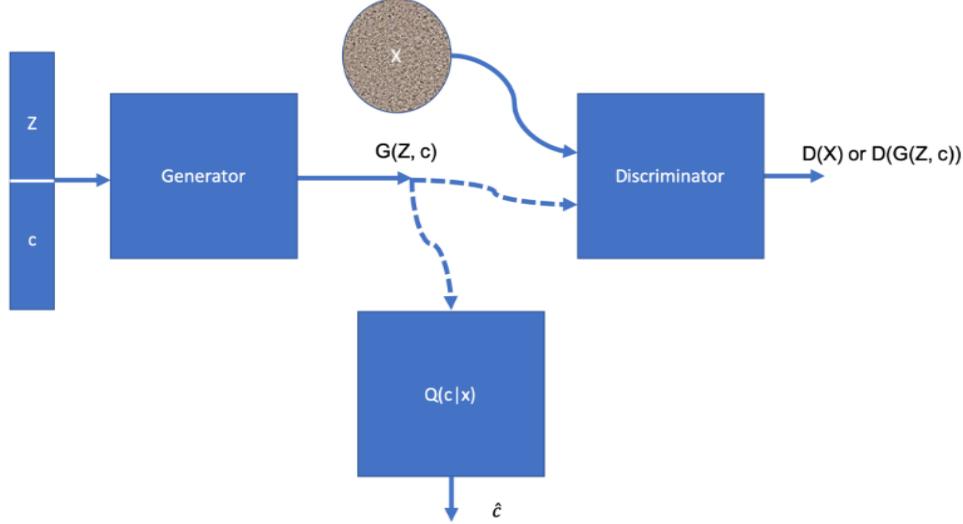
معماری گن معمولاً کنترل کمی بر روی تصاویر تولید شده در اختیار ما قرار می‌دهد. InfoGAN این امر را تغییر می‌دهد؛ به صورتیکه کنترل بر ویژگی‌های مختلف تصاویر تولید شده را فراهم می‌کند.

⁴⁶ fully connected
⁴⁷ standard Gaussian

از مفاهیم تئوری اطلاعات استفاده می‌کند به طوریکه عبارت نویز به کدهای نهان تبدیل شده و باعث ایجاد کنترلی قابل پیش‌بینی و منظم بر خروجی می‌شود.

هنجاری استفاده می‌شود که مجموعه داده ما بسیار پیچیده باشد یا وقتی می‌خواهیم InfoGAN را آموزش دهیم و مجموعه داده برچسب‌گذاری نشده است و یا زمانی که می‌خواهیم مهمترین ویژگی‌های تصاویر خود را بینیم.

مولد در InfoGAN دو ورودی می‌گیرد، فضای نهان Z و کد نهان c ، بنابراین خروجی مولد $G(Z, c)$ است. گن به گونه‌ای آموزش دیده است که اطلاعات مشترک بین کد پنهان c و تصویر تولید شده $G(Z, c)$ را به حد اکثر می‌رساند. شکل ۱۳-۲ معماری InfoGAN را نشان می‌دهد:



شکل ۱۳-۲
معماری شبکه

[۲۸] InfoGAN

بردار بهم پیوسته (Z, c) به مولد داده می‌شود. $Q(c|X)$ همچنین یک شبکه عصبی، همراه با مولد، می‌باشد که برای ایجاد نگاشت بین نویز تصادفی Z و کد پنهان آن یعنی \hat{c} کار می‌کند، هدف آن تخمین c با توجه به X است. این امر با افزودن یک عبارت منظم‌سازی بهتابع هدف گن معمولی، محقق می‌شود:

$$\min_G \max_D V_I(D, G) = V(D, G) - \lambda I(c; G(z, c)) \quad (6-2)$$

عبارت $VG(D, G)$ تابع خطای گن معمولی است و عبارت دوم برای منظم‌سازی می‌باشد که λ یک ثابت است. مقدار آن معمولاً روی یک تنظیم می‌شود و $I(c; G(z, c))$ اطلاعات مشترک بین کد پنهان c و تصویر تولید شده توسط مولد $G(z, c)$ است.

۹-۳-۲: DiscoGAN

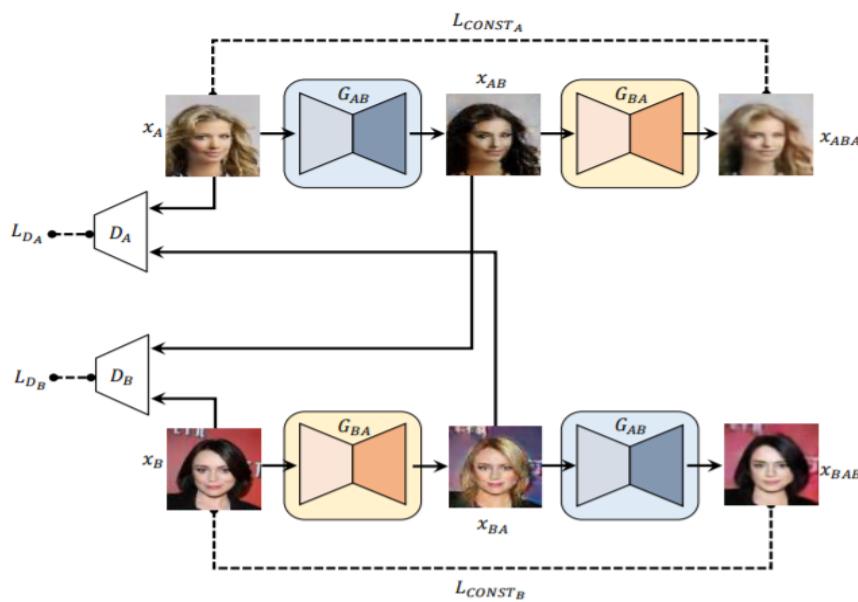
در حالی که انسان‌ها بدون هیچ گونه نظارتی به راحتی روابط بین داده‌های حوزه‌های مختلف را تشخیص می‌دهند، یادگیری کشف خودکار آن‌ها به طور کلی بسیار چالش برانگیز است و به تعداد زیادی از جفت‌های حقیقی نیاز دارد که روابط را نشان دهند. برای جلوگیری از جفت شدن پوهزینه، ما به کشف روابط بین دامنه‌ای با توجه به داده‌های غیرجفت می‌پردازیم. ما روشی مبتنی بر شبکه‌های مولد خصم‌انه را پیشنهاد می‌دهیم که یاد می‌گیرد روابط بین حوزه‌های مختلف را کشف کند. با استفاده از روابط کشف شده، شبکه پیشنهادی ما با حفظ ویژگی‌های کلیدی مانند هویت چهره، استایل را با موفقیت از یک دامنه به دامنه دیگر منتقل می‌کند.

مفهوم اصلی DiscoGAN بسیار شبیه CycleGAN است:

- هر دوی آنها دوتابع تغییرشکل منحصر به فرد را یاد می‌گیرند، یکی تغییر شکل از دامنه X به دامنه Y و دیگری نگاشت معکوس که هر دو از تابع خطای بازسازی⁴⁸ به عنوان معیاری برای سنجش بازسازی تصویر اصلی پس از دو بار تغییر شکل در بین دامنه‌ها، استفاده می‌کنند.
- هر دو از این اصل پیروی می‌کنند که اگر تصویری را از دامنه اول به دامنه دوم تبدیل کنیم و سپس دوباره آن را به دامنه اول برگردانیم، باید با تصویر اصلی مطابقت داشته باشد.
- تفاوت اصلی بین DiscoGAN و CycleGAN این است که DiscoGAN از دو خطای بازسازی استفاده می‌کند، یکی برای هر کدام از دامنه‌ها، در حالی که CycleGAN تنها از خطای ثبات چرخه⁴⁹ استفاده می‌کند.

مانند DiscoGAN، نیز بر اساس خطای بازسازی ساخته شده است. ایده به این صورت است که وقتی تصویری از یک دامنه به دامنه دیگر منتقل می‌شود و سپس به دامنه اصلی بازمی‌گردد، تصویر تولید شده باید تا جای ممکن به تصویر اصلی نزدیک باشد. در این حالت، این تفاوت کم به عنوان خطای بازسازی در نظر گرفته می‌شود و در حین آموزش، مدل سعی می‌کند این اختلاف را به حداقل برساند.

از نظر ریاضی، رابطه این مدل به عنوان تابع GAB تعریف می‌شود که عناصر را از دامنه خود یعنی A به عناصر موجود در دامنه‌ی مشترک خود یعنی B و GBA (که به طور مشابه تعریف شده است) نگاشت می‌کند. در شرایط کاملاً بدون نظارت، GAB و GBA می‌توانند به دلخواه تعریف شوند. برای یافتن یک رابطه معنی دار، باید شرطی را، به رابطه دلخواه اعمال کنیم. در اینجا، ما رابطه را محدود می‌کنیم تا یک تنازنگار یک به یک باشد. این بدان معنی است که GAB نگاشت معکوس GBA است. دامنه عملکرد GAB ، مجموعه کاملی از همه مقادیر حاصله ممکن ($GAB(x_A)$ برای همه x_A در دامنه A ، باید در دامنه B و به طور مشابه برای $GAB(x_B)$ موجود باشد).^[۲۱]

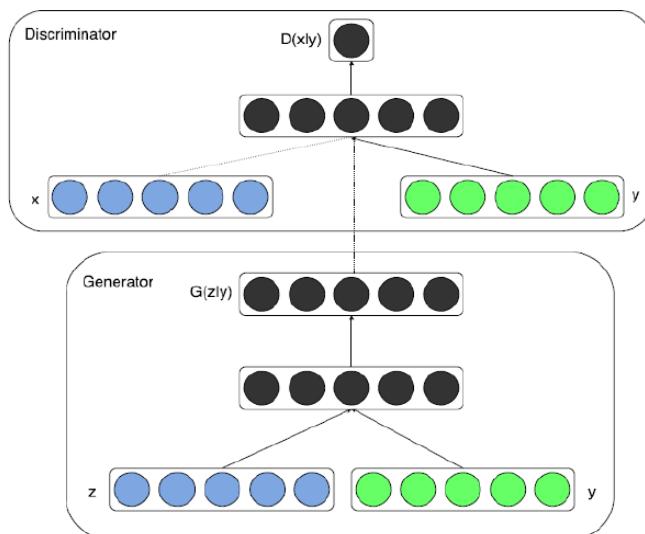


شکل ۱۴-۲
[۳۴] DiscoGAN ساختار شبکه

reconstruction loss⁴⁸
Cycle consistency loss⁴⁹

۵۰:CGAN - ۱۰-۳-۲

روشی است برای تولید شرطی خروجی ها ، به این معنی که هم مولد و هم تفکیک کننده به نوعی اطلاعات کمکی نیاز پیدا می کنند.^[۳۵] در این شبکه ، مولد می تواند به گونه ای آموزش بینند که تصاویر جدید را تولید کند در حالی که ورودی آن یک بردار تصادفی (z)، به همراه بعضی از ورودی های اضافی (اطلاعات کمکی)، به صورت شرطی ایجاد شده است. این ورودی های اضافی (y)، می توانند هر گونه از اطلاعات کمکی ، مانند برچسب های کلاس یا داده مدل های دیگر باشد. تفکیک کننده نیز به صورت شرطی می باشد. به این صورت که تصاویر تولید شده یا واقعی به همراه اطلاعات کمکی (برچسب کلاس) را به عنوان ورودی گرفته و برای آموزش ارتباط بین عکس ها و برچسب ها استفاده می کند. با تعريف متغیر شرطی y در مدل سازی و افزودن شرط هایی به مدل از طریق y ، می توان فرایند تولید داده را هدایت کرد.



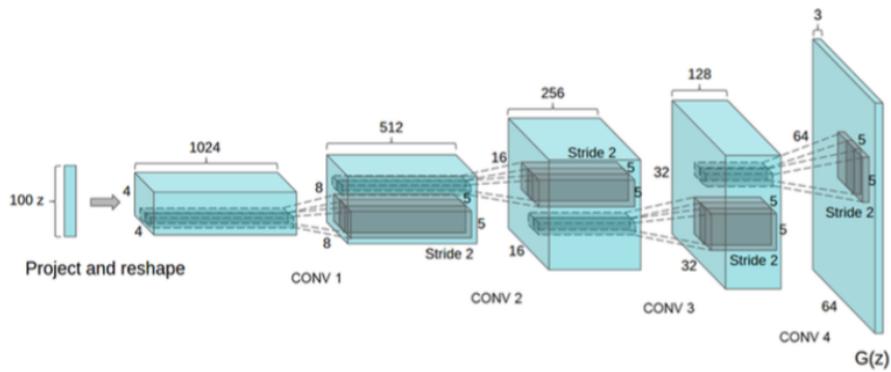
شکل ۱۵-۲
[۲۰] CGAN ساختار شبکه

۵۱:DCGAN - ۱۱-۳-۲

در DCGAN، معماری GAN اصلی با شبکه های کانولوشن عمیق (CNN) بهبود یافته است. در مقایسه با GAN اصلی، DCGAN تقریباً به طور کامل از لایه کانولوشن (convolution) به جای لایه کاملاً متصل (fully connected) استفاده می کند.^[۳۶] DCGAN ها پایدارتر هستند و تصاویر با کیفیت بالاتری تولید می کنند. در DCGAN، نرمال سازی دسته ای در هر دو شبکه مولد و تفکیک کننده انجام می شود. می توان از این شبکه ها برای انتقال سبک استفاده کرد. به عنوان مثال ، می توان از مجموعه ای از کیف های دستی ، برای تولید کفش به همان سبک استفاده کرد.

در این مدل ، در همه لایه های شبکه مولد (به غیر از لایه آخر که از \tanh استفاده می شود) از تابع فعال ساز ReLU استفاده می شود. به علاوه ، در شبکه تفکیک کننده ، برای همه لایه ها از تابع فعال ساز LeakyReLU استفاده می شود.

Conditional GAN^{۵۰}
Deep convolutional GAN^{۵۱}



شکل ۱۶-۲
ساختار شبکه [۳۶] DCGAN

پروژه های مختلف نشان داده اند که آموزش تحت نظارت CNN ها ، در مجموعه داده ای های بزرگ منجر به ایجاد ویژگی های بسیار قدرتمند آموزش دیده ، می شود. هم چنین ، یک DCGAN بدون نظارت که روی مجموعه داده ای های بزرگ آموزش دیده است ، می تواند مجموعه ای از ویژگی های جذاب را نیز ییاموزد.

۱۶-۳-۲ :F-GAN

نمونه گیرهای مولد عصبی، مدل های احتمالی هستند که با استفاده از شبکه های عصبی پیش رونده نمونه برداری را انجام می دهند: آنها یک بردار ورودی تصادفی می گیرند و از توزیع احتمال تعریف شده توسط وزن شبکه، نمونه ای تولید می کنند. این مدل ها تماشی هستند و امکان محاسبه کارآمد نمونه ها و مشتقات را دارند، اما نمی توانند برای محاسبه احتمالات یا حاشیه گذاری استفاده شوند. روش آموزش مولد - خصمانه^{۵۲} این امکان را برای آموزش چنین مدل هایی از طریق استفاده از شبکه عصبی تفکیک کننده کمکی فراهم می کند. ما نشان می دهیم که از هر واگرایی f می توان برای آموزش نمونه گیرهای عصبی مولد استفاده کرد. همچنان در مورد مزایای قابلیت انتخاب بین توابع مختلف واگرایی در پیچیدگی آموزش و کیفیت مدل های مولد به دست آمده بحث می شود.^[۳۷]

⁵² generative-adversarial

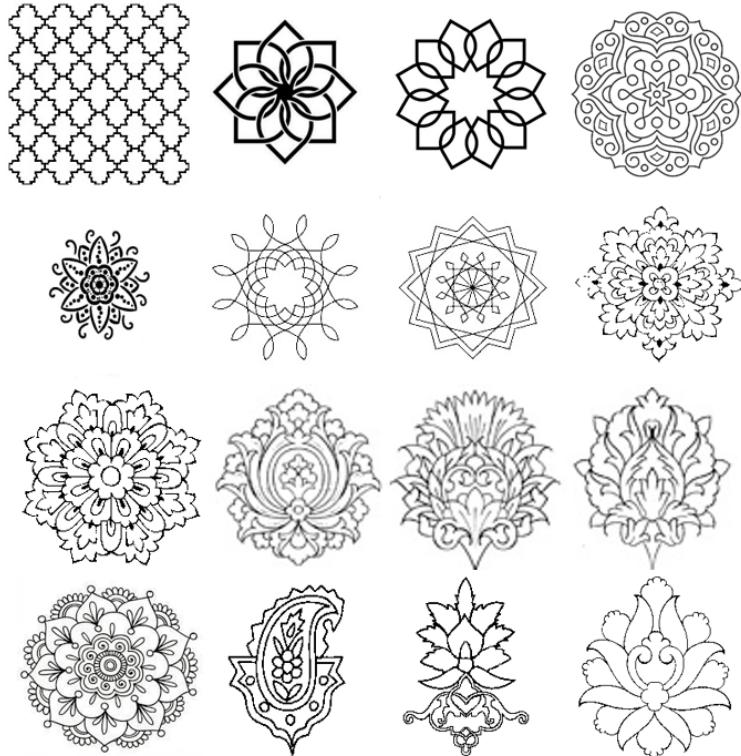
فصل سوم

نتایج

در این فصل، تعدادی از خروجی هایی که از این پروژه به دست آوردهیم را نمایش می دهیم.

۱-۳- ورودی ها

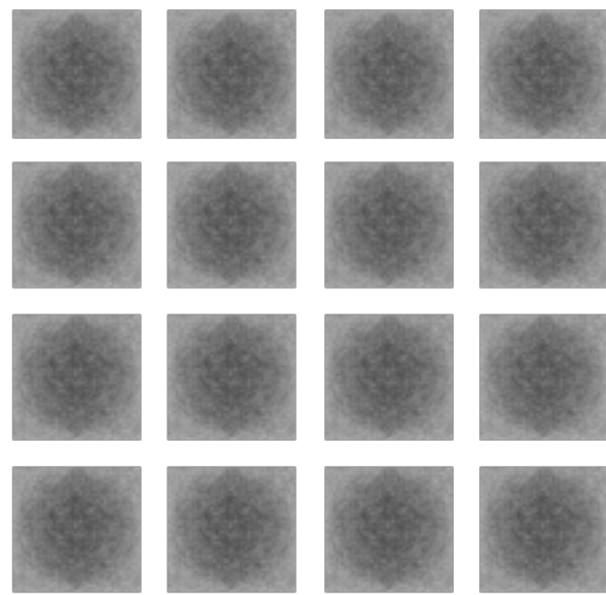
نمونه ای از تصاویر جمع آوری شده که برای آموزش شبکه استفاده شده اند در شکل ۱-۳ نشان داده شده اند:



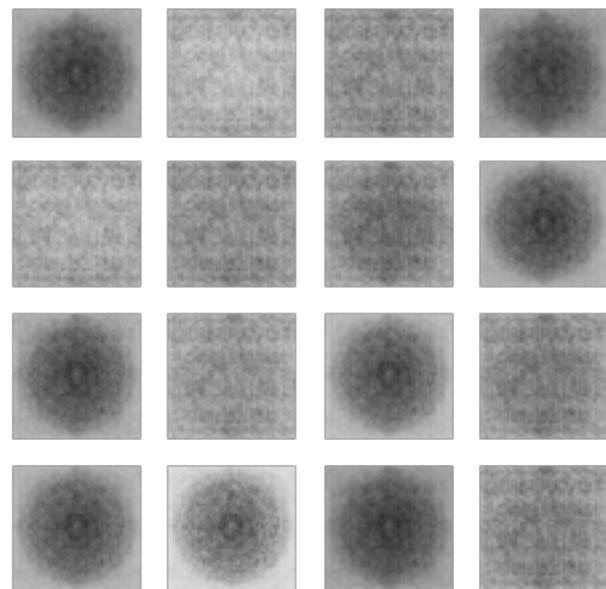
شکل ۱-۳
نمونه هایی از تصاویر ورودی

۲-۳- خروجی ها

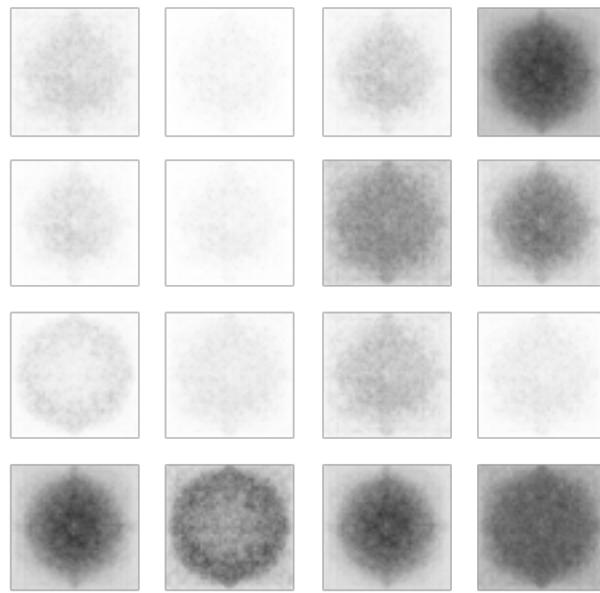
تعداد کل تکرار ها در این شبکه، ۵۰۰۰۰ تکرار می باشد که از بین تمام خروجی ها تعدادی را به نمایش می گذاریم:



1000 [D loss: 2.495899 , acc: 15.23] [G loss: 0.171027]
شکل ۲-۳
خروجی در تکرار ۱۰۰۰

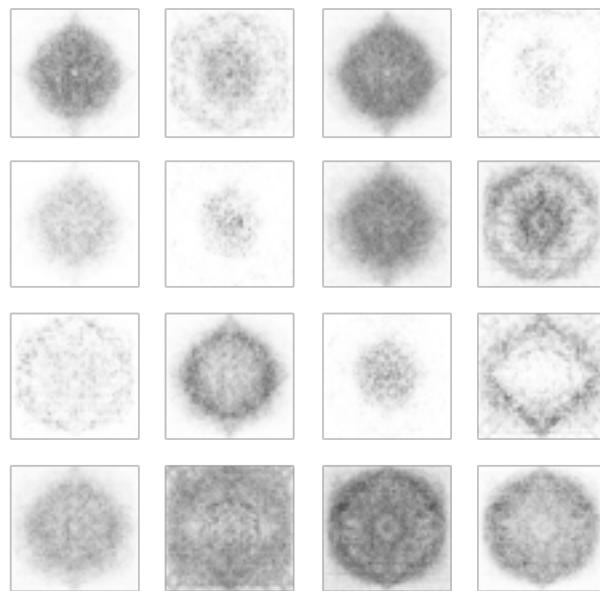


5000 [D loss: 0.336581 , acc: 90.62] [G loss: 1.321354]
شکل ۳-۳
خروجی در تکرار ۵۰۰۰



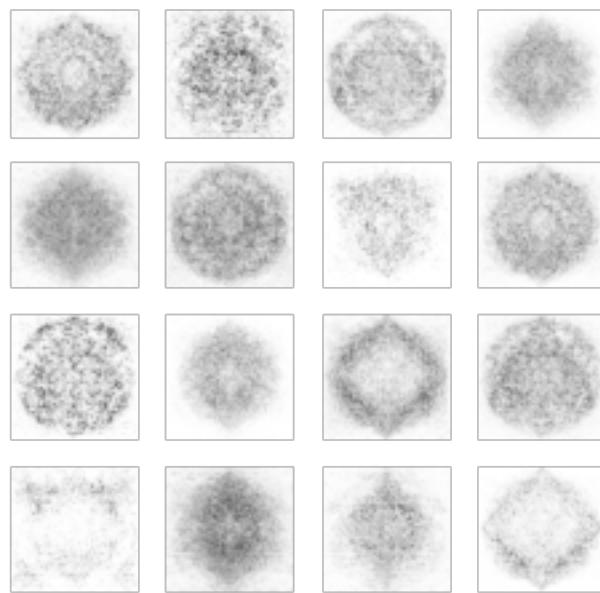
7000 [D loss: 0.397588 , acc: 83.20] [G loss: 1.917125]

شکل ۴-۳
خروجی در تکرار ۲۰۰۰



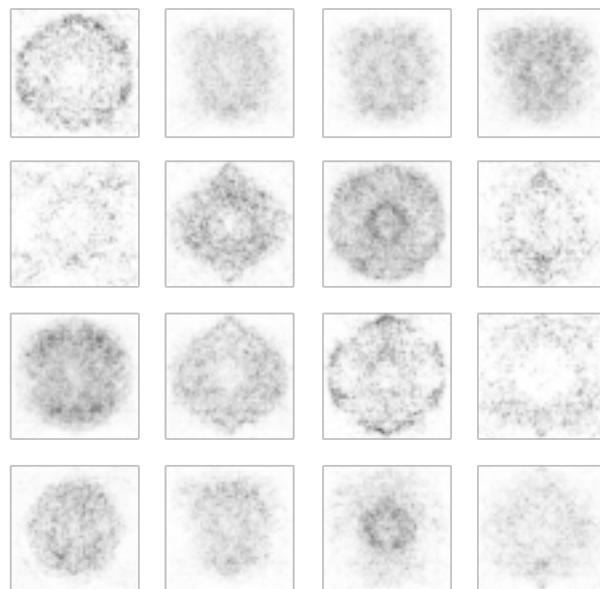
14000 [D loss: 0.504322 , acc: 73.05] [G loss: 2.941812]

شکل ۵-۳
خروجی در تکرار ۱۴۰۰۰



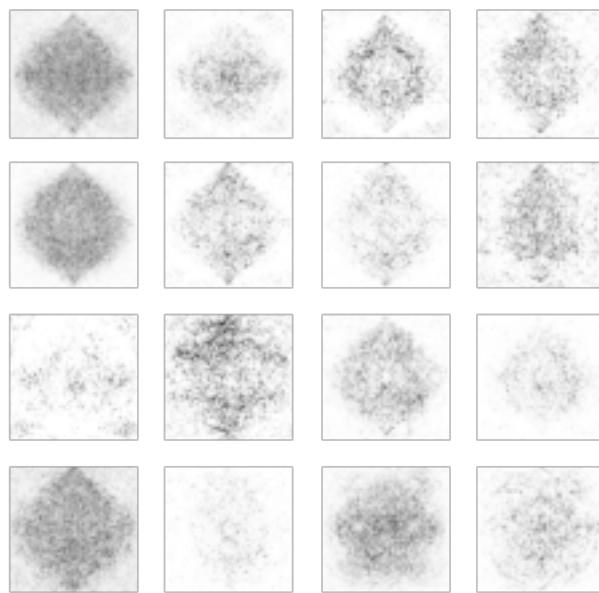
25000 [D loss: 0.407579 , acc: 80.86] [G loss: 3.604243]

شکل ۳-۶
خروجی در تکرار ۲۵۰۰۰



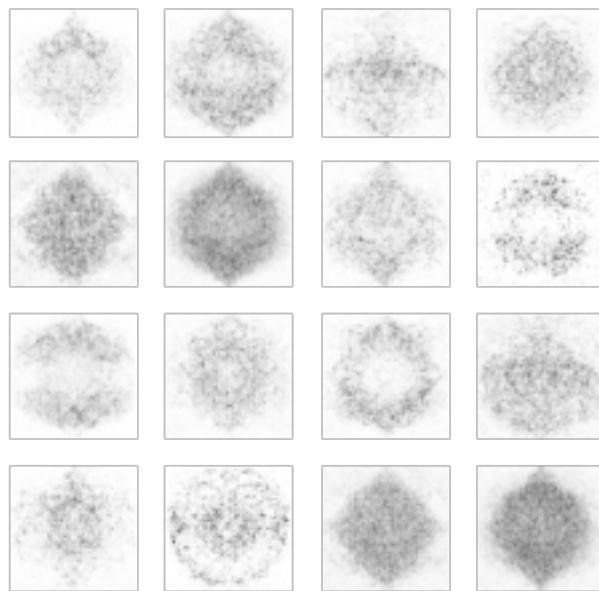
38000 [D loss: 0.145300 , acc: 94.14] [G loss: 5.922412]

شکل ۳-۷
خروجی در تکرار ۳۸۰۰۰



46000 [D loss: 0.110716 , acc: 96.09] [G loss: 3.275170]

شکل ۸-۳
خروجی در تکرار ۴۶۰۰۰



50000 [D loss: 0.096146 , acc: 97.66] [G loss: 5.306379]

شکل ۹-۳
خروجی در تکرار ۵۰۰۰۰

منابع

- [١] Fleming, John; Honour, Hugh (1977). Dictionary of the Decorative Arts.
- [٢] Gutmann, Michael; Hyvärinen, Aapo. Noise-Contrastive Estimation.
- [٣] Goodfellow, Ian; Pouget-Abadie, Jean; Mirza, Mehdi; Xu, Bing; Warde-Farley, David; Ozair, Sherjil; Courville, Aaron; Bengio, Yoshua (2014). Generative Adversarial Nets.
- [٤] Niemitalo, Olli (February 24, 2010). A method for training artificial neural networks to generate missing data within a variable context.
- [٥] reddit r/MachineLearning (2019). GANs were invented in 2010?
- [٦] Li, Wei; Gauci, Melvin; Gross, Roderich (July 6, 2013). A Coevolutionary Approach to Learn Animal Behavior Through Controlled Interaction.
- [٧] Sajjadi, Mehdi S. M.; Schölkopf, Bernhard; Hirsch, Michael (December 23, 2016). EnhanceNet: Single Image Super-Resolution Through Automated Texture Synthesis.
- [٨] Tom Février (February 17, 2019). Le scandale de l'intelligence ARTificielle.
- [٩] Paez, Danny (February 13, 2019). This Person Does Not Exist Is the Best One-Off Website of 2019.
- [١٠] BESCHIZZA, ROB (February 15, 2019). This Person Does Not Exist.
- [١١] Elgammal, Ahmed; Liu, Bingchen; Elhoseiny, Mohamed; Mazzone, Marian (2017). CAN: Creative Adversarial Networks, Generating "Art" by Learning About Styles and Deviating from Style Norms.
- [١٢] Cohn, Gabe (October 25, 2018). AI Art at Christie's Sells for \$432,500.
- [١٣] Mazzone, Marian; Ahmed Elgammal (February 21, 2019). Art, Creativity, and the Potential of Artificial Intelligence.
- [١٤] Kulp, Patrick (May 23, 2019). Samsung's AI Lab Can Create Fake Video Footage From a Single Headshot.
- [١٥] Yu, Yi; Canales, Simon (2021). Conditional LSTM-GAN for Melody Generation from Lyrics.
- [١٦] Seung Wook Kim; Zhou, Yuhao; Phlion, Jonah; Torralba, Antonio; Fidler, Sanja (2020). Learning to Simulate Dynamic Environments with GameGAN.
- [١٧] Salimans, Tim; Goodfellow, Ian; Zaremba, Wojciech; Cheung, Vicki; Radford, Alec; Chen, Xi (2016). Improved Techniques for Training GANs.
- [١٨] Isola, Phillip; Zhu, Jun-Yan; Zhou, Tinghui; Efros, Alexei (2017). Image-to-Image Translation with Conditional Adversarial Nets.
- [١٩] Ho, Jonathon; Ermon, Stefano (2016). Generative Adversarial Imitation Learning.

- [`·] Derrick MwitiJul (Jul 2, 2018). Introduction to Generative Adversarial Networks (GANs): Types, and Applications, and Implementation.
- [`·] Shibsankar Das (Updated Jul 26, 2021). 6 GAN Architectures You Really Should Know.
- [`·] Arjovski (2017). Wasserstein Generative Adversarial Networks.
- [`·] Olkin, I. and Pukelsheim, F. (1982). The distance between two random vectors with given dispersion matrices.
- [`·] John Mannes (June 21, 2017). These undergrads built a Generative Adversarial Net that creates art.
- [`·] Aaron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu (Aug 2016). Pixel Recurrent Neural Networks.
- [`·] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, Honglak Lee (Jun 2016). Generative Adversarial Text to Image Synthesis.
- [`·] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, Stephen Paul Smolley (Apr 2017). Least Squares Generative Adversarial Networks.
- [`·] Packt Editorial Staff (January 8, 2020). 3 different types of generative adversarial networks (GANs) and how they work.
- [`·] Johnson et al (Mar 2016). Perceptual Losses for Real-Time Style Transfer and Super-Resolution.
- [`·] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros (24 Aug, 2020). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks.
- [`·] Tero Karras, Samuli Laine, Timo Aila (29 Mar, 2019). A Style-Based Generator Architecture for Generative Adversarial Networks.
- [`·] Jason Brownlee (August 19, 2019). A Gentle Introduction to StyleGAN the Style Generative Adversarial Network.
- [`·] Tero Karras, Timo Aila, Samuli Laine, Jaakko Lehtinen (26 Feb, 2018). Progressive Growing of GANs for Improved Quality, Stability, and Variation.
- [`·] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, Jiwon Kim (15 May, 2017). Learning to Discover Cross-Domain Relations with Generative Adversarial Networks.
- [`·] Mehdi Mirza (6 Nov, 2014). Conditional Generative Adversarial Nets.
- [`·] Radford A, Metz L, Chintala S (2016). Unsupervised representation learning with deep convolutional generative adversarial networks

[၃၇] Sebastian Nowozin, Botond Cseke, Ryota Tomioka (2 Jun, 2016). f-GAN: Training Generative Neural Samplers using Variational Divergence Minimization.

پیوست

تمامی کدها به صورت فایل زیپ الصاق شده اند.

**University of Shahid Bahonar
Faculty of Engineering
Department of Computer Engineering**

B.Sc. Project

Generative Adversarial Networks

Supervisor: Dr.Eftekhari

**Fatemeh Iranmanesh
Fatemeh Taher**

Spring 1400

