

الحمد لله



موضوع تحقیق:

(Sequence Generative Adversarial Nets (SeqGAN))
And
Objective-Reinforced Generative Adversarial Networks (ORGAN)

مربط به درس :

یادگیری عمیق

استاد:

جناب آقای دکتر محمد علی کیوان راد

دانشجو:

حسین عباسی

شماره دانشجویی:

۹۵۳۸۳۹۰۲۴

اسفند ۹۶

فهرست

۵.....	چکیده :
۶.....	مقدمه
۶.....	شبکه های مولدی ممیزی دنباله ای (GENERATIVE ADVERSARIAL NETS (GAN))
۷.....	الگوریتم های مولد در مقابل الگوریتم های تشخیصی یا متمایز کننده
۸.....	چگونه GAN ها کار می کنند
۱۱.....	کاربردها
۱۲.....	در زیر سه کاربرد از کاربردهای بالا را بررسی شده است.
۱۵.....	کاربردهای عملیاتی انجام شده توسط GAN
۱۷.....	پیاده سازی یک نمونه کد از GAN با تنسورفلو بر روی دیتاست MNIST
۲۲.....	مدل SEQGAN
۲۶.....	مدل ORGAN
۲۹.....	مقایسه ORGAN و روشهای دیگر
۳۰.....	نتیجه گیری
۳۱.....	مراجع

فهرست شکل ها

۹ شکل ساختار GAN
۲۱ خروجی تولید شده توسط GAN
۲۲ ساختار SEQGAN
۲۶ ساختار ORGAN
۲۷ شبه کد ORGAN
۲۸ تاثیر پارامتر یادگیری تقویتی
۲۹ مقایسه ORGAN و روشهای دیگر

چکیده:

در این گزارش ساختار روش GAN بررسی شده است و سپس چند کاربرد مطرح شده در مقالات را نام برده شده است و خروجی ها شرکت انویدیا با استفاده از این روش گفته شده است و یک پیاده سازی از این روش بر روی دیتاست MNIST را بررسی شده است که با استفاده از تنسورفلو پیاده سازی شده می باشد. در آخر دو روش که توسعه داده شده GAN بوده اند با عنوان SeqGAN و ORGAN بررسی شده اند. در مواردی که دنباله ای و به صورت متوالی می باشند روش GAN نمی تواند نتیجه مطلوبی بدهد زیرا این روش وقتی مدل به طور کامل ساخته می شود آن را ارزیابی می کند و در جزها هیچ نظر را ارائه نمی دهد در حالی که در روش SeqGAN برای هر جز که تولید می شود میزان هزینه حساب می شود و گرادیان زده می شود و اصلاح وزن ها صورت می گیرد. در مسائلی مانند تولید موسیقی، شعر و متن که از بخش های مختلفی تشکیل شده است و جزیهای کوچکتری تشکیل شده بررسی در هر جز باعث نتیجه بهتر و حتی دقیق تر می شود. اما در روش ORGAN برخلاف روش قبلی که فقط به تولید محتوا فقط تمرکز داشت. نوع محتوا تولید براساس یادگیری تقویتی نیز در نظر گرفته می شود به طوری که محدود فضای مرجع را نسبت به یک ناحیه یا هدف خواست محدود می کند و نمی گذارد محتوای شامل کل فضای مرجع باشد از طرفی متمایزکننده از دانش قبلی تولید شده را در تولید محتوا در نظر می گیرد به طور کلی تعادل و تنظیم معیار تاثیر یادگیری تقویتی نیز باید در نظر گرفته شود.

شبکه های مولدی ممیزی دنباله ای ((Generative Adversarial Nets (GAN))

این الگوریتم از سیستم شبکه های عصبی مصنوعی بهره می گیرد. سیستم های شبکه های عصبی مصنوعی با تقلید از کارکرد نورون های مغز انسان ساخته شده اند. در الگوریتم GAN دو شبکه عصبی وجود دارد که "در برابر" (بر علیه) یکدیگر کار می کنند. یکی از این شبکه ها الگوریتمی است که محتوا می سازد و شبکه دوم آنچه ساخته شده است را به چالش می گیرد و نقش مخالف را بازی می کند.

شبکه های GANs معماری خالصی از شبکه های عصبی عمیق هستند که از دو شبکه تشکیل شده اند. GANs در یک مقاله توسط آقای Goodfellow و دیگر محققان در دانشگاه مونترال، از جمله Yoshua Bengio، در سال ۲۰۱۴ معرفی شد. Yann LeCun، مدیر تحقیقات آی.بی.سی فیس بوک، "آموزش رقابتی" را "جالب ترین ایده در ۱۰ سال گذشته در یادگیری ماشین دانسته است.

پتانسیل GAN بسیار زیاد است، زیرا آنها می توانند یادگیری هر نوع توزیع داده را تقلید کنند. به این معنا، GAN ها می توانند آموزش داده شوند تا جهان هایی را که در هر حوزه ای شبیه به خودمان هستند، ایجاد کنیم: تصاویر، موسیقی، سخنرانی، پروسه. آنها یک ربات هنرمند هستند، و خروجی آنها چشمگیر است.



الگوریتم های مولد در مقابل الگوریتم های تشخیصی یا متمایز کننده

برای درک GAN ها، باید بدانید که چگونه الگوریتم های مولد کار می کنند، و برای این که آنها را با الگوریتم های تشخیصی مقایسه می کند، آموزنده است. الگوریتم های تشخیصی سعی دارند اطلاعات ورودی را طبقه بندی کنند؛ به عبارت دیگر، با توجه به ویژگی های یک نمونه داده، آنها یک برچسب یا دسته ای را که به آن داده ها تعلق دارد، پیش بینی می کنند.

به عنوان مثال، با توجه به تمام کلمات در یک ایمیل، یک الگوریتم تبعیض آمیز می تواند پیش بینی کند که آیا پیام اسپم یا غیراسپم باشد. اسپم یکی از برچسب هاست و مجموعه ای از کلمات که از ایمیل جمع آوری شده، ویژگی هایی هستند که داده های ورودی را تشکیل می دهند. هنگامی که این مشکل به صورت ریاضی بیان می شود، برچسب نامیده می شود و ویژگی ها X نامیده می شوند. فرمول $p(y|x)$ به معنای "با توجه به ایمیل x برچسب y را پیش بینی کنید" است، که در این مورد به "احتمال اینکه یک ایمیل هرزنانه باشد با توجه به کلمات موجود در آن بررسی می شود" ترجمه شود، استفاده می شود.

بنابراین الگوریتم های تشخیصی ویژگی های نقشه را به برچسب ها می رسانند. آنها تنها با این همبستگی نگران هستند. یک راه برای فکر کردن در مورد الگوریتم های مولد این است که آنها مخالف می کنند. به جای پیش بینی یک برچسب با توجه به ویژگی های خاص، آنها تلاش می کنند ویژگی هایی را با توجه به یک برچسب خاص پیش بینی کنند.

سوال الگوریتم تولیدی تلاش می کند جواب دهد: فرض کنید این ایمیل به عنوان هرزنانه باشد، احتمال این ویژگی ها چقدر است؟ در حالی که مدل های اختیاری در مورد رابطه بین y و x ، مدل های نسبی در مورد "چگونگی x شما" را مورد توجه قرار می دهند.

تقسیم بندی دیگری در مورد عملکرد مدل های مولدها و تشخیصی به شرح زیر است:

- مدل های تشخیصی مرز بین کلاس ها را می آموزند
- مدل های تولیدی توزیع کلاس های فردی را مدل می کنند

چگونه GAN ها کار می کنند

یک شبکه عصبی که از دو قسمت تشکیل شده است یک تولید کننده که داده های را تولید می کند که شبیه به داده های واقعی باشد و تشخیص دهنده را فریب دهد و تشخیص دهنده تلاش می کند تا داده ها جعلی تولید شده توسط تولید کننده را از داده های واقعی متمایز کند.

در یک مثال که بر روی عده های دست نوشته قصد داریم مطرح کنیم به این شرح است که یک مجموعه ارقام دست نویس مانند مجموعه داده MNIST که از دنیای واقعی گرفته شده است را به کار ببریم که در این مسئله هدف تشخیص دهنده این باشد که معتبر بودن مجموعه داده ورودی که از داده های واقعی است را داده واقعی تشخیص دهد و تصاویری را که از تولید کننده به عنوان جعلی ساخته شده است را جعلی شناسایی کنند. در عین حال، تولید کننده تصاویر جدیدی را ایجاد می کند که به تشخیص دهنده داده شود. این کار به این امید است که آنها نیز معتبر تشخیص داده شوند، هرچند که آنها جعلی هستند. هدف تولید کننده این است که تصاویر تولید شده توسط تشخیص دهنده معتبر تشخیص داده شود و به طور کلی تشخیص دهنده را فریب دهد. در اینجا مراحل GAN انجام می شود:

ژنراتور عدد تصادفی را می گیرد و یک تصویر را تولید می کند.

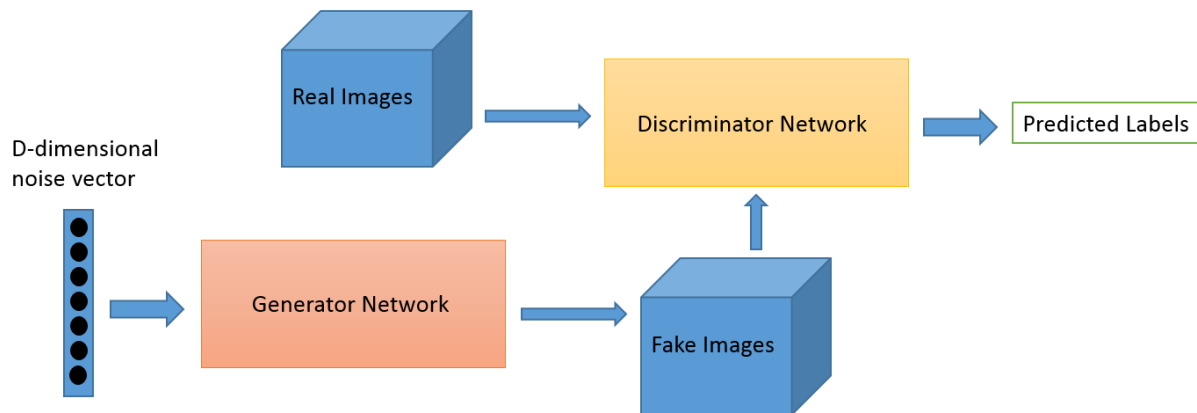
این تصویر تولید شده در کنار یک جریان تصاویر گرفته شده از مجموعه داده های واقعی، به تشخیص دهنده داده می شود.

تشخیص دهنده یا متمایز کننده هر دو تصویر واقعی و جعلی را می گیرد و عدد بین ۰ و ۱ بر می گرداند ۱ نشانگر پیش بینی اعتبار و ۰ نشان دهنده جعلی بودن تصویر است.

بنابراین شما یک حلقه بازنگری دوگانه دارید:

متمایز دهنده در حلقه بازخورد با برچسب واقعی از تصاویر است که ما می دانیم.

ژنراتور در حلقه بازخورد با متمایزدهنده است.



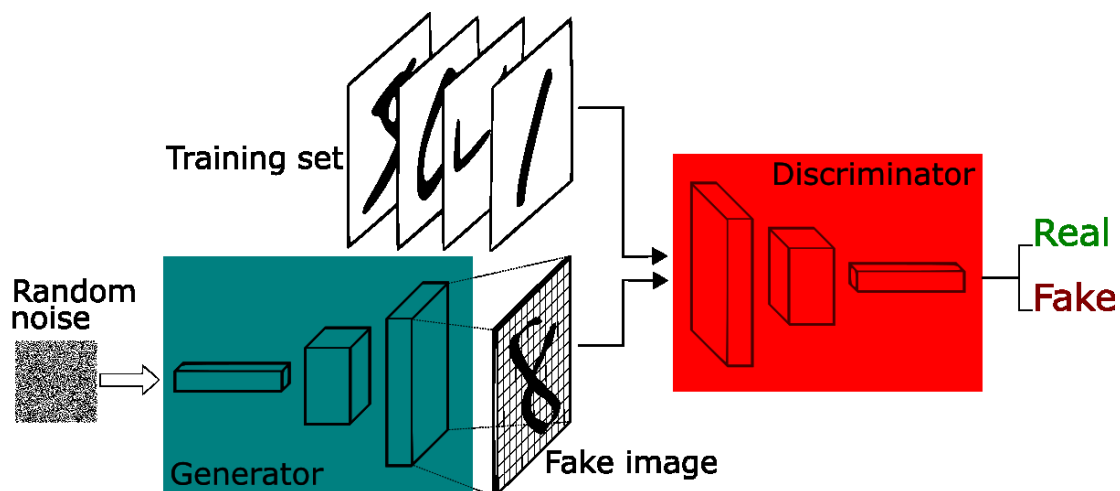
شکل ساختار GAN

شما می توانید GAN را به عنوان ترکیبی از دزد و پلیس در یک بازی تعقیب و گریز فکر کنید، جایی که دزد در حال یادگیری تولید پول تقلبی است و پلیس در حال یادگیری تمایز دادن بین پول واقعی و تقلبی است. هر دو پویا هستند یعنی پلیس نیز در حال آموزش است و هر کدام در تلاش برای کشف ترفندهای یکدیگر می باشند.

در این مثال متمایزکننده یک شبکه convolutional استاندارد است که می تواند تصاویری را که به آن متصل شده است را برچسب زند و به دو گروه واقعی یا جعلی طبقه بندی کند، تولیدکننده یک شبکه convolutional معکوس است: در حالی که یک طبقه بندی استاندارد کانولاسیون یک تصویر را می گیرد و آن را برای تولید یک احتمال می کند، ژنراتور یک بردار نویز تصادفی را می گیرد و نمونه آن را به یک تصویر می

برد. اولین داده ها را از طریق تکنیک های downsampling مانند maxpool انجام می دهد و دومین داده های جدید را تولید می کند.

هر دو شبکه در تلاش برای بهینه سازی یک سری تابع از اهداف متفاوت و مخالف هستند، و یا بدست آوردن عملکرد بهتر در یک بازی مجموع صفر هستند. همانطور که تبعیض کننده رفتار خود را تغییر می دهد، تولید کننده هم این کار انجام می دهد و نیز بالعکس. ضرر و زیان آنها بر علیه یکدیگر است.



شکل ساختار پیشنهادی برای ارقام دست نویس براساس معماری GAN

کاربردها

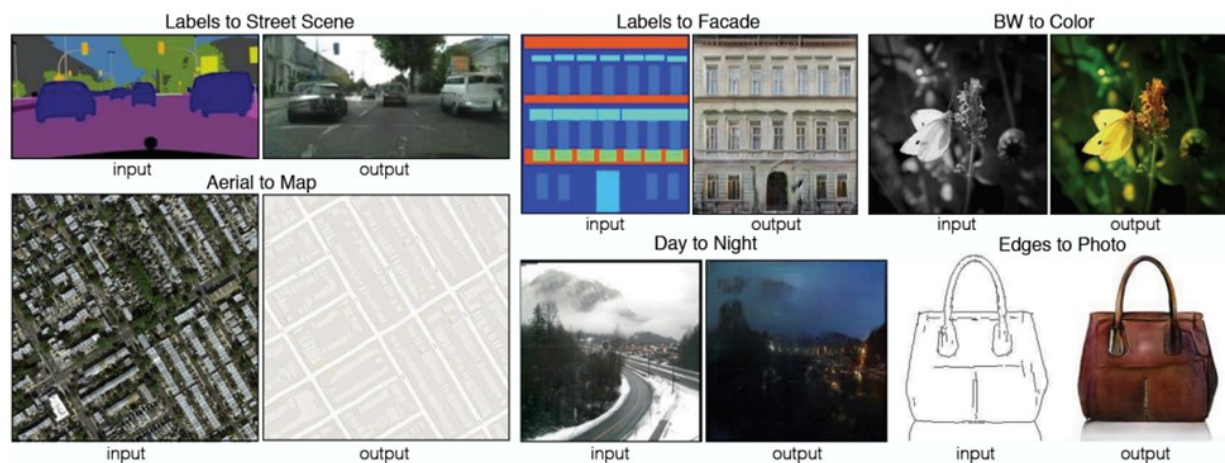
بعضی از کاربردهای GAN به شرح زیر است

- تولید فونت
- تولید شخصیت انیمیشنی
- تولید تصویر تعاملی
- Text2Image (متن به تصویر)
- Obejct 3D تولید
- ویرایش تصویر
- پیری صورت
- بررسی زست‌های انسان
- انتقال دامنه (مانند انتقال سبک، pix2pix، sketch2image)
- Image inpainting
- تقویت رزولوشن
- تولید تصویر با وضوح بالا (تصویر بزرگ در مقیاس)
- مثالهای رقابتی (دفاع در برابر حمله)
- تشخیص شی
- رباتیک
- ویدئو (تولید / پیش بینی)
- تولید داده های مصنوعی
- Image-to-Image Translation
- و کاربردها دیگر

در زیر سه کاربرد از کاربردهای بالا را بررسی شده است.

Image-to-Image Translation

شبکه های GANs به عنوان یک راه حل عمومی برای مشکلات ترجمه تصویر به تصویر به کار برده می-شوند. این شبکه ها نه تنها نقشه برداری از تصویر ورودی به تصویر خروجی را یاد می گیرند، بلکه یک تابع هزینه را برای آموزش این نقشه برداری یاد می گیرد. این باعث می شود که یک رویکرد عمومی را برای مسائلی که به طور سنتی نیاز به فرمولاسیون های مختلف داشته اند به کار ببریم. این روش در تلفیق عکس ها از نقشه های برجسب، بازسازی اشیاء از نقشه های لبه و رنگ آمیزی تصاویر در میان کارهای دیگر موثر است. علاوه بر این، از زمان انتشار نرم افزار pix2pix ، صدها نفر از کاربران توییتر، آزمایش های هنری خود را با استفاده از این سیستم انجام داده اند.



Text-to-Image Synthesis

سنتز خودکار تصاویر واقعی از متن جالب و مفید خواهد بود، اما سیستم های فعلی هوش مصنوعی هنوز از این هدف دور هستند. با این حال، در سال های اخیر، معماری های شبکه عصبی قدرتمند برای شناسایی برای این کار توسعه یافته اند. در عین حال، شبکه GANs شروع به تولید تصاویر بسیاری از دسته های خاصی مانند چهره ها، اتاق های داخلی و غیره کرده اند



Face Aging

اخیرا نشان داده شده است که شبکه های GANs می توانند تصاویر مصنوعی از فضای بصری استثنایی ایجاد کنند. در این کاربرد، روش مبتنی بر GAN را برای پیر شدن اتوماتیک صورت استفاده شده است. برخلاف آثار قبلی که با استفاده از GAN ها برای تغییر ویژگی های صورت، تاکید خاصی بر حفظ هویت اصلی فرد در نسخه سالانه چهره او داشته اند. برای این منظور، رویکرد جدیدی را برای بهینه سازی هویت از بردارهای پنهان GAN معرفی شد. ارزیابی عینی تصاویر صورت چهره ساله و جوانسازی شده توسط روش های تشخیص چهره و تخمین سن، نشان دهنده پتانسیل بالایی از روش GANs است.

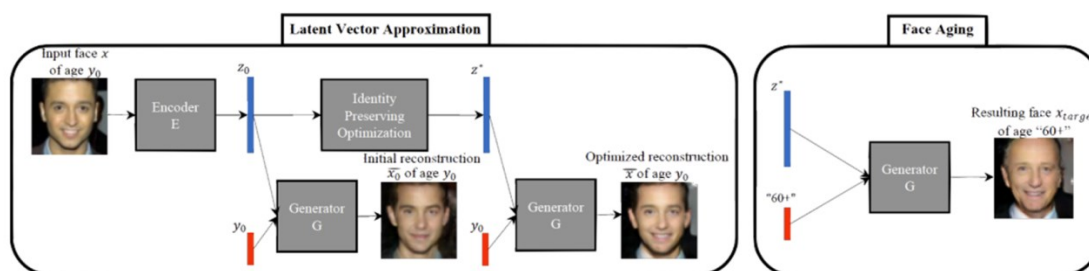


Figure 1 in the original paper.

Antipov, G., Baccouche, M., & Dugelay, J. L. (2017). "Face Aging With Conditional Generative Adversarial Networks". arXiv preprint arXiv:1702.01983.

کاربردهای عملیاتی انجام شده توسط GAN

تغییر شرایط آب و هوایی، جابجایی شب و روز

کمپانی انویدیا که اخیراً سیستمی را برای تولید تصاویر ساختگی توسعه داده بود، حالا سیستم مبتنی بر هوش مصنوعی دیگری را برای تغییر شرایط آب و هوایی، جابجایی شب و روز و تبدیل گربه سانان به یکدیگر طراحی کرده است. سیستم جدید یک روش یادگیری بدون نظارت است که امکان تغییر محتوای ویدئو را با دقتی بالا فراهم می کند.



روش های فعلی نیازمند مقایسه مقادیر زیادی داده است که هنگام طراحی سیستم های بدون ناظر چالش های زیادی را پیش روی توسعه دهندگان قرار می دهد. به عبارت دیگر سیستم های مشابه برای تولید داده های مورد نیاز به مجموعه داده های برچسب گذاری شده نیاز دارند. با این حال سیستم جدید انویدیا به داده های برچسب گذاری شده نیاز ندارد. این پروژه برای تولید تصاویر از " شبکه عصبی عمیق مولد (GAN) " استفاده می کند. یکی دیگر از قابلیت های این برنامه تبدیل تصاویر گربه سانان مختلف از قبیل گربه، پلنگ، گربه وحشی و غیره به یکدیگر است که با ظرافت خاصی صورت می گیرد.

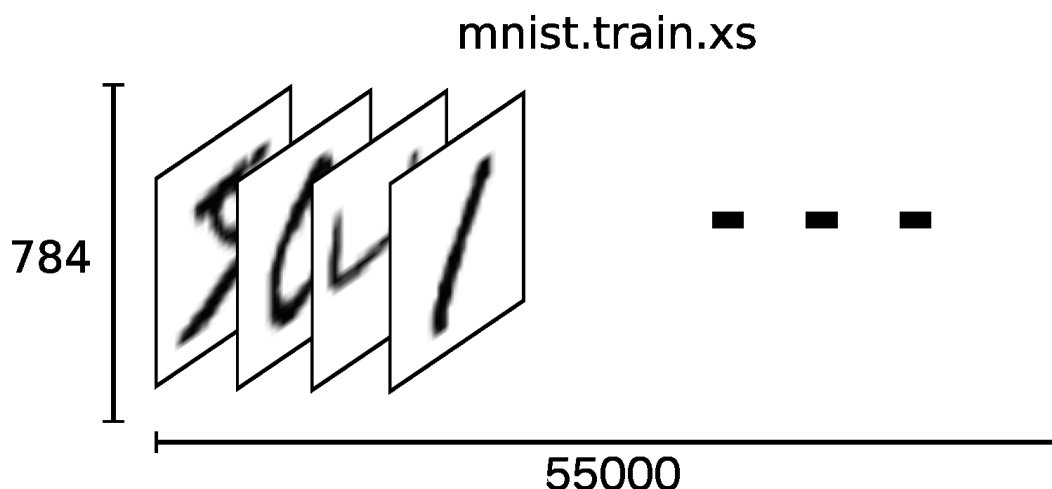
انسانهایی که هرگز نبوده‌اند و نخواهند بود



همه‌ی این تصویرها را هوش مصنوعی شرکت NVIDIA با بهره‌گیری از الگوریتمی به نام GAN ساخته است. اکنون الگوریتم GAN شرکت NVIDIA با بهره‌گیری از پایگاه داده عکس‌های CelebA-HQ که عکس آدم‌های مشهور در آن قرار دارد عکس‌های انسان‌هایی را می‌سازد که برآستی وجود ندارند. این تصویرها آنچنان به واقعیت نزدیک هستند که بیشتر مردم نمی‌توانند ساختگی بودن آنها را تشخیص دهند و باور خواهند کرد که تصویری از انسان زنده است.

پیاده سازی یک نمونه کد از GAN با تنسورفلو بر روی دیتاست MNIST

در این پیاده سازی از داده های آموزشی MNIST که شامل تصاویر ۵۵۰۰۰ تصویر 28×28 پیکسل است (LeCun, Bottou, Bengio, & Haffner, 1998) استفاده شده است. هر تصویر به عنوان یک بردار با اندازه ۷۸۴ عنصر نمایش داده می شود و هر عنصر دارای مقدار یک شدت پیکسل بین ۰ و ۱ است.



مقدار دهی ها اولیه مانند مسیر داده ورودی و خروجی و اندازه دسته ها برای هر مرحله آموزش و اندازه لایه مخفی به شکل زیر انجام می شود.

```
M = 128 # batch size during training
d = 100 # latent dimension

data_dir = "/tmp/data"
out_dir = "/tmp/out"
```

هدف این است که یک مدل ایجاد کنیم که بتواند تصاویر با کیفیت بالا از ارقام دست نوشته تولید کند.

در طی تمرین ما دسته ای از ارقام MNIST را خوانده می شود یک محلول نگهدارنده TensorFlow را با اندازه دسته ای ثابت از تصاویر M ترسیم خواهد شد.

یک تابع کمکی را برای انتخاب دسته بعدی از نقاط داده ها از مجموعه کامل نمونه ها تعریف شده است.

```
from observations import mnist

def generator(array, batch_size):
    """Generate batch with respect to array's first
    axis."""
    start = 0 # pointer to where we are in iteration
    while True:
        stop = start + batch_size
        diff = stop - array.shape[0]
        if diff <= 0:
            batch = array[start:stop]
            start += batch_size
        else:
            batch = np.concatenate((array[start:],
            array[:diff]))
            start = diff
        batch = batch.astype(np.float32) / 255.0 # normalize
        pixel intensities
        batch = np.random.binomial(1, batch) # binarize
        images
        yield batch

(x_train, _), (x_test, _) = mnist(data_dir)
x_train_generator = generator(x_train, M)
x_ph = tf.placeholder(tf.float32, [M, 784])
```

مدل

کد قسمت تولید کننده

```
from edward.models import Uniform

def generative_network(eps):
    net = tf.layers.dense(eps, 128, activation=tf.nn.relu)
    net = tf.layers.dense(net, 784, activation=tf.sigmoid)
    return net

with tf.variable_scope("Gen"):
    eps = Uniform(tf.zeros([M, d]) - 1.0, tf.ones([M, d]))
```

```
x = generative_network(eps)
```

کد قسمت متمایز کننده

```
def discriminative_network(x):  
    """Outputs probability in logits."""  
    net = tf.layers.dense(x, 128, activation=tf.nn.relu)  
    net = tf.layers.dense(net, 1, activation=None)  
    return net
```

اجازه دهید P توزیع داده های واقعی را نشان دهد. مشکل بهینه سازی در GAN ها استفاده می شود

$$\min_{\theta} \max_{\phi} \mathbb{E}_{p^*(x)} [\log D(x; \phi)] + \mathbb{E}_{p(x; \theta)} [\log(1 - D(x; \phi))].$$

این مسئله بهینه سازی دابلکس است: برای یک پارامتر نسبی و یک راه حل حداکثر با توجه به پارامترهای تشخیصی نیاز به یک راه حل مینیمم دارد. در عمل، الگوریتم با تکرار به روز رسانی شیب در هر یک انجام می شود.

بسیاری از منابع شواهدی در پشت آموزش GAN وجود دارد. یکی، که انگیزه اصلی است، بر این باور است که دو شبکه عصبی بازی می کنند. تمایز دهنده تلاش می کند تا نمونه های واقعی را از نمونه های تولید شده تولید کننده جدا کند. تولیدکننده تلاش می کند نمونه هایی را تولید کند که توسط تمایز دهنده قابل تشخیص نباشد هدف آموزش، دستیابی به تعادل نش است.

الگوریتم GAN (GANInference) به سادگی مدل تراکم ضخیم بر روی X را به عنوان ورودی می گیرد، و به اهداف آن x_{ph} متصل می شود. علاوه بر این، تمایز کننده تابع پارامتریک برای تشخیص نمونه هایشان ارائه می شود.

```
inference = ed.GANInference( data={x: x_ph}, discriminator=discriminative_network)
```

ما از ADAM به عنوان بهینه ساز برای هر دو ژنراتور و تبعیض کننده استفاده می کنیم. ما الگوریتم را برای ۱۵۰۰۰ تکرار اجرا خواهیم کرد و هر ۱۰۰۰ تکرار را به صورت پیشرفته چاپ می کنیم.

```
optimizer = tf.train.AdamOptimizer()
optimizer_d = tf.train.AdamOptimizer()

inference.initialize(
    optimizer=optimizer, optimizer_d=optimizer_d,
    n_iter=15000, n_print=1000)
```

ما اکنون حلقه اصلی را تشکیل می دهیم که GAN را آموزش می دهد. در هر تکرار پارامترها را با توجه به الگوریتم بروز رسانی می کند. در هر ۱۰۰۰ تکرار، پیشرفت را چاپ می کند و همچنین یک نمونه از نمونه تولید شده از مدل را ذخیره می کند.

```
sess = ed.get_session()
tf.global_variables_initializer().run()

idx = np.random.randint(M, size=16)
i = 0
for t in range(inference.n_iter):
    if t % inference.n_print == 0:
        samples = sess.run(x)
        samples = samples[idx, ]

        fig = plot(samples)
        plt.savefig(os.path.join(out_dir, '{}.png').format(
            str(i).zfill(3)), bbox_inches='tight')
        plt.close(fig)
        i += 1

    x_batch = next(x_train_generator)
    info_dict = inference.update(feed_dict={x_ph: x_batch})
    inference.print_progress(info_dict)
```

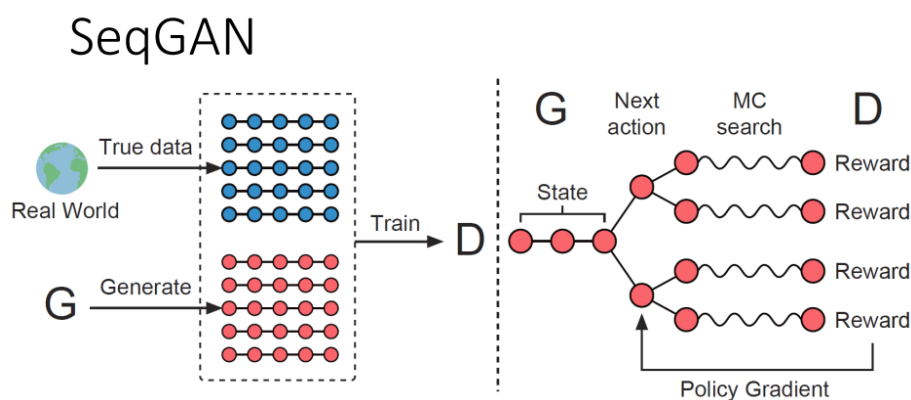
خروجی بدست آمده از پیاده سازی بالا بر روی دیتاست MNIST به شکل زیر می باشد.



خروجی تولید شده توسط GAN

مدل SeqGAN

به عنوان یک روش جدید برای آموزش مدل های مولد، شبکه های GAN که از یک مدل متمایزکننده (مدل ممیزی) برای هدایت آموزش مدل مولد استفاده می کنند، موفقیت های زیادی در تولید داده های واقعی به دست آورده است. با این حال، زمانی که هدف برای تولید توالی های نشانه های گسسته است، دارای محدودیت می شود. یکی از دلایل اصلی این است که خروجی های گسسته از مدل مولد باعث می شود تا به روزرسانی گرادینت از مدل متمایزکننده (مدل ممیزی) به مدل مولد منتقل شود. همچنین، مدل متمایزکننده تنها می تواند توالی کامل را ارزیابی کند، در حالیکه برای دنباله ای که به طور جزئی تولید شده است، نمره فعلی و آتی یک بار برای تمام سیکل تولید شده است. در این مقاله، ما یک چارچوب تولید دنباله ای به نام SeqGAN پیشنهاد می کنیم تا مشکلات را حل کنیم. مدل سازی مولد به عنوان یک سیاست تصادفی در یادگیری تقویتی است. سیگنال پاداش RL از متمایزکننده GAN بدست می آید و در مراحل تکمیل حالت متوسط با استفاده از جستجو مونت کارلو به مرحله بعدی می رسد. آزمایش های گسترده در مورد داده های مصنوعی و وظایف دنیای واقعی نشان دهنده پیشرفت قابل توجهی نسبت به خطوط اصلی قوی است. کارهای انجام شده با SeqGAN می توان شعرهای چینی و سخنان سیاسی باراک اوباما را تولید شده است. و همین طور موسیقی ها با استفاده از این روش تولید شده است



ساختار SeqGAN

تصویر بالا ساختار یک SeqGAN است که در قسمت چپ تصویر: متمایزکننده (D) بر روی داده های واقعی و داده های تولید شده توسط تولیدکننده (G) آموزش داده شده است و در قسمت راست تصویر تولیدکننده (G) توسط روش سیاست گرادیان که در آن سیگنال پاداش نهایی توسط متمایزکننده (D) ارائه شده است با استفاده از روش جستجو مونت کارلو و بازگشت به عقب به مقدار متوسط پاداش می رسد و آموزش دیده می شود.

در این روش تولید دنباله ای SeqGAN پیشنهاد شده است تا به طور مؤثر شبکه های متمرکز مولدی را برای ایجاد توالی ساختاری از طریق gradient policy ارائه دهد. برای کسب اطلاعات بیشتر، این اولین کار گسترش GAN ها برای تولید توالی های نشانه های گسسته است. در آزمایشات داده های مصنوعی از یک مکانیزم ارزیابی اوراکل برای نشان دادن برتر بودن SeqGAN از مدل های پایه های قوی استفاده شده است. برای سه کاربرد در دنیای واقعی، یعنی اشعار، زبان گفتاری و نسل موزیک، SeqGAN عملکرد بسیار خوبی در تولید توالی های خلاقانه ایفا کرده اند. همچنین مجموعه ای از آزمایشات را برای بررسی استحکام و پایداری آموزش SeqGAN انجام داده اند.

Algorithm 1 Sequence Generative Adversarial Nets

Require: generator policy G_θ ; roll-out policy G_β ; discriminator D_ϕ ; a sequence dataset $\mathcal{S} = \{X_{1:T}\}$

- 1: Initialize G_θ, D_ϕ with random weights θ, ϕ .
- 2: Pre-train G_θ using MLE on \mathcal{S}
- 3: $\beta \leftarrow \theta$
- 4: Generate negative samples using G_θ for training D_ϕ
- 5: Pre-train D_ϕ via minimizing the cross entropy
- 6: **repeat**
- 7: **for** g-steps **do**
- 8: Generate a sequence $Y_{1:T} = (y_1, \dots, y_T) \sim G_\theta$
- 9: **for** t in $1 : T$ **do**
- 10: Compute $Q(a = y_t; s = Y_{1:t-1})$ by Eq. (4)
- 11: **end for**
- 12: Update generator parameters via policy gradient Eq. (8)
- 13: **end for**
- 14: **for** d-steps **do**
- 15: Use current G_θ to generate negative examples and combine with given positive examples \mathcal{S}
- 16: Train discriminator D_ϕ for k epochs by Eq. (5)
- 17: **end for**
- 18: $\beta \leftarrow \theta$
- 19: **until** SeqGAN converges

شبکه کد SeqGAN

در الگوریتم بالا به طور خلاصه، جزئیات کامل SeqGAN پیشنهاد شده را نشان می دهد. در ابتدای آموزش، ما از برآورد حداکثر احتمال (MLE) برای پیش آزمون تولیدکننده (G_θ) استفاده می کنیم و در مجموعه آموزشی S یک سیگنال نظارت شده از یک متمایزکننده پیش آموزش دیده کشف شده است که در آموزش تولیدکننده مفید است. پس از پیش آموزش تولیدکننده و متمایزکننده به طور متناوب آموزش دیده اند. همانگونه که تولیدکننده از طریق آموزش در g-steps مرحله بروزرسانی شده تاست، متمایزکننده هم مجبور به آموزش مجددا شده است تا باتولیدکننده هماهنگ باشد. هنگام آموزش متمایز کننده، مثالهای مثبت از مجموعه داده S است، در حالی که نمونه های منفی از تولیدکننده تولید می شود. برای حفظ تعادل، تعداد نمونه های منفی که ما برای هر مرحله d تولید می کنیم همان نمونه مثبت را انتخاب می کنیم. و برای کاهش

تنوع برآورد، ما از مجموعه های مختلف نمونه های منفی همراه با مثبت استفاده می کنیم که شبیه به بوت استرپینگ است (Quinlan 1996).

که در آزمون t-test مقدار p-value برای SeqGAN بررسی شده است که نتیجه زیر حاصل شده است.

Algorithm	Random	MLE	SS	PG-BLEU	SeqGAN
NLL	10.310	9.038	8.985	8.946	8.736
p-value	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$	

بخش متمایزکننده در روش SeqGAN

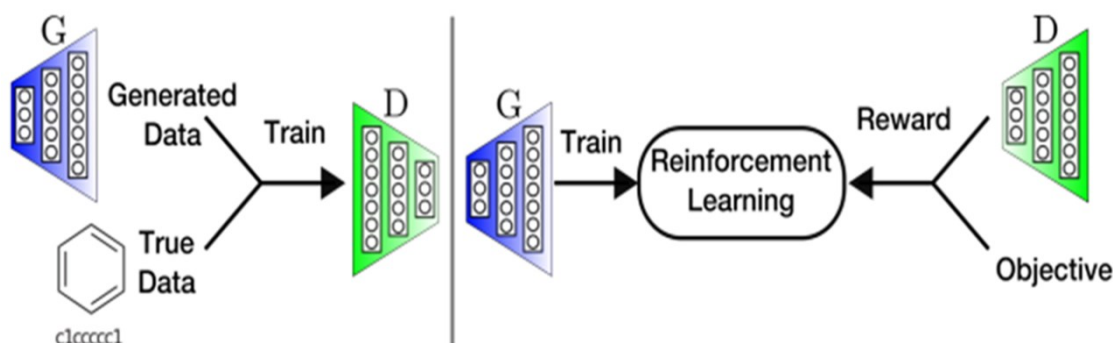
برای این بخش می توانیم از شبکه های عصبی عمیق مانند شبکه های عصبی عمیق (Vesely et al. 2013)، شبکه عصبی کانولوشن (CNN) کیم (۲۰۱۴) و شبکه عصبی کانولوشن مجدد (RCNN) (Lai et al. 2015) کارایی بالا را نشان داده اند که در مسائل طبقه بندی پیوسته مناسب هستند. در مسائلی مانند تولید متن از CNN به عنوان متمایز کننده می توان استفاده کرد زیرا CNN به تازگی اثربخشی متناسب با متن (توالی نشانه) (Zhang and LeCun 2015) نشان داده شده است.

بخش تولیدکننده در روش SeqGAN

در قسمت تولیدکننده بیشتر روش های که از دنباله گسسته از نشانه ها تشکیل شده اند و از داده های گذشته هم در تولید داده استفاده می کنند مثل روش RNN و LSTM در این بخش استفاده می شود.

مدل ORGAN

در وظایف تولید اطلاعات بدون نظارت، علاوه بر تولید یک نمونه بر اساس مشاهدات قبلی، اغلب مایل به دادن نکات به مدل به منظور تعصب نسل به معیارهای مطلوب است. روشی از ترکیب شبکه های GANs و یادگیری تقویتی (RL) را برای به دست آوردن دقیق هدف فوق پیشنهاد شده است در حالی که RL فرآیند تولید داده ها را نسبت به معیارهای دلخواه منحرف می کند، جزء متمایزکننده GAN که تابع پاداش را تعیین می کند، تضمین می کند که مدل هنوز اطلاعاتی را که از داده ها یاد گرفته است، به یاد آورد. بر اساس نتایج قبلی که GAN ها و RL به منظور تولید داده های متوالی ایجاد شده اند و این مدل را در چندین تنظیم برای تولید مولکول های کد گذاری شده به عنوان توالی های متن و در زمینه تولید موسیقی، به کار رفته است، نشان داده شده که برای هر مورد می توانیم به طور موثر فرآیند تولید را نسبت به معیارهای هدف تغییر داده شود.



ساختار ORGAN

در این کار، یک چارچوب کلی ارائه شده است که بر پیشرفت های اخیر شبکه های GAN برای بهینه سازی یک هدف دلخواه در یک کار تولید متوالی ارائه شده است. نشان داده شده که ORGAN معیارهای مورد نظر را بهبود داده تا نتایج بهتر از RNN هایی که از طریق MLE یا SeqGAN آموزش دیده اند را بهبود بخشد. مهمتر از همه، در هنگام استفاده از تنظیمات رقابتی برای حفظ داده ها غیر تکراری، بتوان تولید داده ها را به سمت

یک عملکرد پاداش خاص تنظیم کرد. برای رسم ساختن ORGAN ها از دیدگاه نظری برای درک اینکه چه زمانی و چگونه آنها همگرا هستند بسیار مهم است که درک کنیم که GAN ها به طور کلی همگرا هستند،

Algorithm 1: Objective-Reinforced Generative Adversarial Networks (ORGAN)

require : Generator G_θ ; Discriminator D_ϕ ; dataset $S = \{Y_{1:T}\}$; **objective** $O : Y \rightarrow (0, 1)$
Initialize G_θ, D_ϕ with random weights θ, ϕ ;
Pre-train G_θ using MLE on S ;
Generate negative samples using G_θ for training D_ϕ ;
Pre-train D_ϕ by minimizing the cross entropy;
repeat
 for g -steps **do**
 Generate a sequence $Y_{1:T} = (y_1, \dots, y_T) \sim G_\theta$;
 for t in $1 : T$ **do**
 Compute $Q(s = Y_{1:t-1}, a = y_t)$ by Eq. 4 using D_ϕ and O to calculate rewards per Eq. 6
 end
 Update generator parameters θ via policy gradient by Eq 5.;
 end
 for d -steps **do**
 Use G_θ to generate negative examples and combine with given positive examples S ;
 Train discriminator D_ϕ for k epochs by Eq. 1;
 end
until model converges;

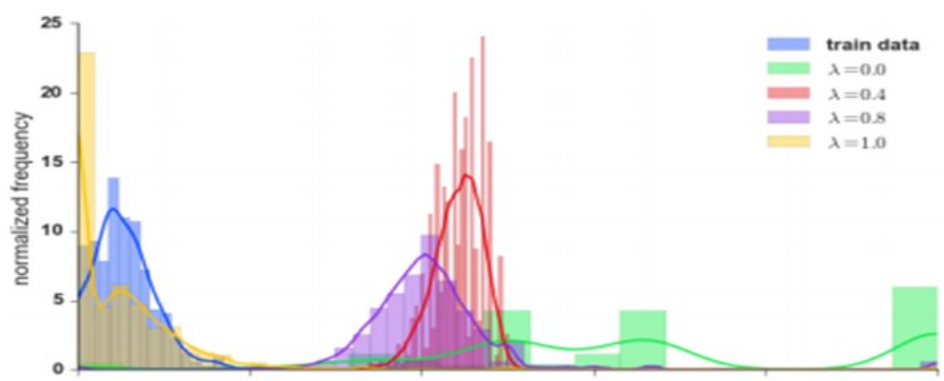
شبهه کد ORGAN

در این روش همان الگوریتم روش SeqGAN می باشد که بخش به رنگ آبی نشان دهنده تفاوت الگوریتم این روش با روش SeqGAN است جایی که D متمایزکننده است و O اهدافی است که نشان دهنده هر گونه اکتشافی است که اهداف ما می باشد.

$$R(Y_{1:T}) = \lambda \cdot D_\phi(Y_{1:T}) + (1 - \lambda) \cdot O(Y_{1:T})$$

زمانی که مقدار لاندا مساوی صفر باشد مدل D را نادیده می گیرد و RL ساده لوحانه می شود، در حالی که زمانی مقدار آن مساوی یک باشد یک مدل SeqGAN می شود.

تأثیر پارامتر یادگیری برای یادگیری تقویتی توزیع دیتا واقعی توسط تولید کننده در شکل زیر نشان داده شده است همانطور که می بینید نرخ یادگیری تقویتی باید براساس هدف خاص باید تنظیم شود نسبت به توزیع دیتا واقعی است.



تأثیر پارامتر یادگیری تقویتی

نتیجه حاصل از مقایسه ORGAN و الگوریتم‌های دیگر بر روی دیتاست‌ها که شامل مولکول‌های مختلف و معیارهای اندازه‌گیری مختلف در شیمی در شکل زیر نشان داده شده است.

Dataset	Algorithm	Novelty	Diversity	Solubility	Synth.	Drug.
<i>Tiny mols</i>	Naive RL	0.94	0.69	0.40	0.73	0.90
	ORGAN ($\lambda = 0.2$)	0.92	0.51	0.38	0.73	0.90
	SeqGAN	0.22	0.41	0.21	0.30	0.59
	MLE	0.52	0.24	0.12	0.16	0.42
<i>Small mols</i>	Naive RL	0.96	0.69	0.70	0.83	0.93
	ORGAN ($\lambda = 0.8$)	0.23	0.41	0.22	0.30	0.6
	SeqGAN	0.32	0.11	0.20	0.17	0.28
	MLE	0.38	0.14	0.24	0.24	0.38
<i>Drug-like</i>	Naive RL	0.95	0.64	0.41	0.82	0.91
	ORGAN ($\lambda = 0.8$)	0.97	0.56	0.78	0.81	0.92
	SeqGAN	0.24	0.09	0.13	0.12	0.19
	MLE	0.32	0.13	0.20	0.20	0.29

مقایسه ORGAN و روش‌های دیگر

نتیجه حاصل از مقایسه ORGAN و الگوریتم‌های دیگر بر روی دیتاست‌ها موسیقی در شکل زیر نشان داده شده است.

Objective	Algorithm	Tonality	Melodicity	Ratio of Steps	Diversity
<i>Tonality</i>	Naive RL	0.629	0.640	0.023	0.086
	ORGAN ($\lambda = 0.6$)	0.449	0.452	0.010	0.274
	SeqGAN	0.048	0.101	0.317	0.377
	MLE	0.055	0.097	0.280	0.560
<i>Melodicity</i>	Naive RL	0.467	0.629	0.062	0.238
	ORGAN ($\lambda = 0.6$)	0.594	0.679	0.033	0.386
	SeqGAN	0.057	0.103	0.304	0.535
	MLE	0.055	0.097	0.280	0.560
<i>Ratio of Steps</i>	Naive RL	0.001	0.003	0.901	0.278
	ORGAN ($\lambda = 0.8$)	0.010	0.030	0.771	0.551
	SeqGAN	0.044	0.084	0.302	0.648
	MLE	0.055	0.097	0.280	0.560

مقایسه ORGAN و روش‌های دیگر

نتیجه‌گیری

با توجه به این که در روش های GAN معمولی متمایزکننده داده تولید شده نهایی را مورد ارزیابی قرار می‌دهد و براساس آن مقدار زیان بررسی می‌شود. میزان اشتباه تولیدکننده به طور کلی بررسی می‌شود و مرحله ای که تولیدکننده در هنگام تولید داده اشتباه کرده است بررسی نمی‌شود تا آن در آن مرحله اصلاح انجام شود تا در مراحل بعد آن نیز این اشتباه ادامه داده نشود برای همین جز بررسی کردن مواردی که به صورت دنباله ای از نشانه ها می‌باشد مدنظر قرار گرفت. که در مسائلی مانند تولید متن و موسیقی می‌توان از روشهای SeqGAN استفاده کرد اما باز در این روش داده از تمام فضای مرجع تولید می‌شود و براساس هدف خاصی متمرکز نیست که برای حل این مشکل از روش GAN ترکیب شده با یادگیری تقویتی استفاده شده که باعث شد داده‌های تولید شده متمرکز با هدف در نظر گرفته باشد و محدود به دامنه مورد نظر ما در فضای مرجع باشد. کارهای که می‌توان انجام داد بر روی داده‌های غیر دنباله‌ای نیز این روش‌ها را استفاده کنیم با بکارگیری متمایزکننده محلی در داخل تولیدکننده استفاده کنیم یا اینکه کلاً متمایزکننده را در داخل تولیدکننده بکار ببریم.

Goodfellow; Generative Adversarial Networks. *arXiv Preprint arXiv:1701.00160*;2016

Goodfellow, I. J. (2014). On distinguishability criteria for estimating generative models. In *ICLR workshop*.

Lantao Yu, Weinan Zhang, Jun Wang, Yong Yu; SeqGAN: Sequence Generative Adversarial Nets with Policy Gradient; Submitted, 25 Aug 2017.

Gabriel Lima Guimaraes, Benjamin Sanchez-Lengeling, Carlos Outeiral, Pedro Luis Cunha Farias, Alán Aspuru-Guzik; Objective-Reinforced Generative Adversarial Networks (ORGAN) for Sequence Generation Models; arXiv:1705.10843v3 [stat.ML] 7 Feb 2018

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. (2014). Generative adversarial nets. In *Neural information processing systems*.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., & Smola, A. (2012). A kernel two-sample test. *The Journal of Machine Learning Research*, 13, 723–773.

Gutmann, M. U., Dutta, R., Kaski, S., & Corander, J. (2014). Statistical Inference of Intractable Generative Models via Classification. *arXiv Preprint arXiv:1407.4981*.

Gutmann, M., & Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Artificial intelligence and statistics*.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278–2324.

Marin, J.-M., Pudlo, P., Robert, C. P., & Ryder, R. J. (2012). Approximate Bayesian computational methods. *Statistics and Computing*, 22(6), 1167–1180.

Mohamed, S., & Lakshminarayanan, B. (2016). Learning in Implicit Generative Models. *arXiv Preprint arXiv:1610.03483*.

Rubin, D. B. (1984). Bayesianly justifiable and relevant frequency calculations for the applied statistician. *The Annals of Statistics*, 12(4), 1151–1172.

Sugiyama, M., Suzuki, T., & Kanamori, T. (2012). Density-ratio matching under the Bregman divergence: a unified framework of density-ratio estimation. *Annals of the Institute of Statistical*