Generative Adversarial Networks

Edilton Brandão de Sousa
Escola de Matemática Aplicada - FGV/EMAp
ediltonbs5@gmail.com

Thiago Franke Melchiors
Escola de Matemática Aplicada - FGV/EMAp
thiago.franke.ms@gmail.com

Abstract

Este relatório tem como objetivo descrever as complexidades das Generative Adversarial Networks (GANs), detalhando sua estrutura única, os princípios matemáticos que orientam seu treinamento e os resultados teóricos que sustentam sua eficácia. Através de uma revisão do algoritmo fundamental e de uma implementação prática, o documento visa elucidar os desafios e as capacidades destes modelos inovadores, proporcionando um entendimento profundo desse tema tão interessante.

1 Introdução

No vasto campo da Inteligência Artificial, as Generative Adversarial Networks (GANs) surgiram como uma das grandes inovações no aprendizado de máquina nos últimos anos. Tendo como base o artigo de Goodfellow et al. [1], este relatório tem como referência o framework original proposto para estimar distribuições gerativas sem a necessidade de modelos prescritivos. Utilizando um processo que envolve duas redes neurais — um gerador e um discriminador —, as GANs são capazes de aprender as informações no qual é alimentada e gerar novos dados praticamente indistinguíveis dos originais.

As GANs podem ter diversas utilidades, desde o aprimoramento de imagens e a criação de arte até a geração de modelos em 3D. No entanto, apesar de seu grande potencial, essas redes podem enfrentar desafios relacionados à sua convergência e à qualidade das gerações. Este relatório, realizará uma análise profunda da metodologia que fundamentam as GANs, com um foco especial na matemática relacionada. Além disso, um experimento prático será apresentado, o qual não apenas ilustrará a teoria, mas também discutirá as nuances envolvidas no treinamento e na implementação desses modelos.

2 Metodologia

2.1 Estrutura das GANs

As Generative Adversarial Networks (GANs) são arquiteturas de aprendizado de máquina que consistem em duas redes neurais rivais, o Gerador (G) e o Discriminador (D), ambos usualmente formados por perceptrons multicamadas. O Gerador aprende a criar dados novos e convincentes a partir de uma distribuição de ruído inicial $p_z(z)$, tentando imitar a distribuição de dados reais p_g ao mapear o ruído z para o espaço de dados usando parâmetros θ_g. Por outro lado, o Discriminador avalia amostras recebidas para discernir se são reais, originadas do conjunto de dados de treinamento $p_{data}(x)$, ou falsificações criadas pelo Gerador. Ele é treinado para maximizar a probabilidade de fazer a classificação correta, atribuindo maior probabilidade às amostras reais e menor às geradas por G. Em essência, G é otimizado para enganar D, enquanto D é otimizado para não ser enganado por G, criando um sistema de aprendizado dinâmico e adversarial.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

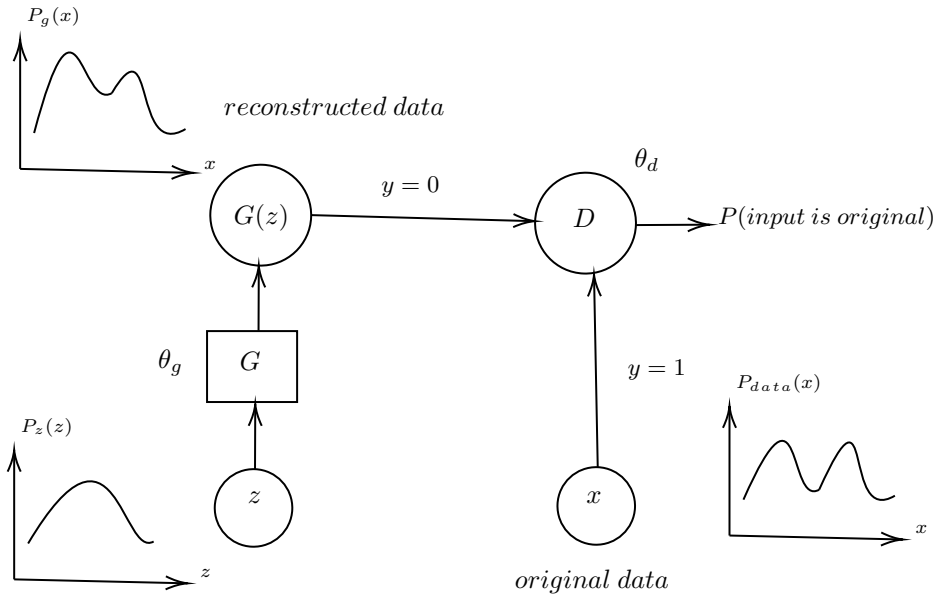

Figure 1: Representação da estrutura de uma GAN [2]

De maneira intuitiva, podemos imaginar que o Gerador e o Discriminador em uma GAN são dois personagens em um jogo de gato e rato. O Gerador é um falsificador habilidoso, dedicando-se a criar cópias tão perfeitas de obras de arte valiosas que são quase indistinguíveis dos originais. Com cada nova obra que cria, ele aprende um pouco mais sobre como replicar as nuances e detalhes que tornam as obras convincentes. Por outro lado, o Discriminador é como um experiente detetive de arte, cuja função é distinguir as falsificações originais. Inicialmente, algumas falsificações passam por autênticas, mas com o tempo, o detetive aprende a detectar as sutilezas que denunciam uma obra como falsa. Neste jogo, cada um está constantemente aprendendo com o outro: o falsificador se torna mais astuto à medida que o detetive se torna mais esperto. Este ciclo de aprimoramento contínuo é a base do aprendizado adversarial, onde a competição direta impulsiona ambos os lados a desenvolverem habilidades cada vez mais sofisticadas.

2.2 A matemática por trás das GANs

2.2.1 Algorítimo

No coração das Generative Adversarial Networks (GANs) reside um confronto matemático moldado pelo jogo de minimax, uma competição calculada entre dois agentes: o gerador G e o discriminador D. A função de valor $V(G, D)$ define o campo de batalha para este jogo, onde D busca maximizá-la, pois o output de $D(x)$ é a probabilidade de que o dado x tenha vindo dos dados reais. Por outro lado, G se esforça para minimizar $\log(1 - D(G(z)))$, ou seja, aumentar a probabilidade de que D classifique erroneamente $G(z)$ como tendo vindo dos dados reais. A equação a seguir captura a essência desta disputa:

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))] \tag{1}$$

Seguimos uma abordagem iterativa, onde k passos são dados na otimização de D, seguidos por um único passo na atualização de G. As GANs visam chegar a um ponto de equilíbrio: uma síntese onde G gera dados tão convincentes que D não consegue diferenciar entre o que é real e o que é artificial, resultando em $D(x) = 0.5$ para todas as entradas.

No algorítimo a seguir conseguimos observar com mais detalhes o processo de otimização:

Algorithm 1: Minibatch stochastic gradient descent training of generative adversarial nets. [1]

Result: Treine o discriminador e o gerador em uma GAN

for *number of training iterations* **do**

  for *k steps* **do**

    Sample minibatch of m noise samples $\{z^{(1)}, ..., z^{(m)}\}$ from noise prior $p_g(z)$;

    Sample minibatch of m examples $\{x^{(1)}, ..., x^{(m)}\}$ from data generating distribution $p_{data}(x)$;

    Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right)$$

  end

  Sample minibatch of m noise samples $\{z^{(1)}, ..., z^{(m)}\}$ from noise prior $p_g(z)$;

  Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)$$

end

Explicação do algorítimo e passos principais:

1. Atualização do Discriminador:

   - Para um número determinado de iterações de treinamento, começamos otimizando o discriminador. Por k vezes (onde k é um hiperparâmetro que define quantas vezes o discriminador é atualizado em relação ao gerador), fazemos o seguinte:
   - Primeiro, um minilote de m amostras de ruído é retirado da distribuição de ruído prior $p_g(z)$.
   - Então, um minilote de m exemplos reais é amostrado da distribuição de dados reais $p_{data}(x)$.
   - Com esses dados, o discriminador é atualizado realizando stochastic gradient ascent. Isso é feito maximizando a função de valor composta pelo logaritmo da probabilidade atribuída às amostras reais somada ao logaritmo de um menos a probabilidade atribuída às amostras geradas.

2. Atualização do Gerador:

   - Após o discriminador ter sido atualizado, passamos para o gerador. Nesta etapa, um novo minilote de m amostras de ruído é amostrado.
   - O gerador é então atualizado pelo stochastic gradient descent, onde a meta é minimizar o logaritmo de um menos a probabilidade que o discriminador atribui às amostras geradas. A intenção é melhorar o gerador para que ele produza dados que o discriminador classificará como reais.

2.2.2 Otimização global de $p_g = p_{data}$

Proposição 1. *Para G fixo, o discriminador ótimo D é:*

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \tag{2}$$

Demonstração. O critério de treinamento do discriminador é, dado um G fixo, minimizar a função objetivo $V(G, D)$:

$$V(G, D) = \int_x p_{data}(x) \log D(x) dx + \int_z p_z(z) \log(1 - D(G(z))) dz \tag{3}$$

$$= \int_x p_{data}(x) \log D(x) + p_g(x) \log(1 - D(x)) dx$$

Para qualquer (a, b), a função $f(x) = a \log x + b \log(1 - x)$ é maximizada em $x = a/(a + b)$. Portanto, o discriminador ótimo é:

$$D^*(x) = \frac{p_{data}(x)}{p_{data}(x) + p_g(x)}$$

□

Podemos agora, definir:

$$C(G) = \max_D V(G, D) = \int_x p_{data}(x) \log D^*(x) + p_g(x) \log(1 - D^*(x)) dx \tag{4}$$

$$= \int_x p_{data}(x) \log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} + p_g(x) \log \frac{p_g(x)}{p_{data}(x) + p_g(x)} dx$$

$$= E_{x \sim p_{data}} \left[\log \frac{p_{data}(x)}{p_{data}(x) + p_g(x)} \right] + E_{x \sim p_g} [\log \frac{p_g(x)}{p_{data}(x) + p_g(x)}]$$

Ou seja, acabamos de fixar D e estamos olhando a expressão apenas em termos de G.

Teorema 1. *O mínimo global do critério de $C(G)$ é obtido se, e somente se, $p_g = p_{data}$. Neste ponto, $C(G)$ atinge o valor $- \log 4$.*

Demonstração. Para $p_{data} = p_g$, temos $D^* = 1/2$, então:

$$C(G) = \log \frac{1}{2} + \log \frac{1}{2} = - \log 4$$

A expressão da divergência de Jensen-Shannon entre duas distribuições p e q é:

$$D_{JS}(p||q) = \frac{1}{2} D_{KL} \left(p \middle\| \frac{p + q}{2} \right) + \frac{1}{2} D_{KL} \left(q \middle\| \frac{p + q}{2} \right)$$

Se compararmos esta expressão com a expressão de $C(G)$, vemos que:

$$C(G) = 2 D_{JS}(p_{data} \| p_g) - \log 4.$$

Portanto, $C(G)$ é duas vezes a divergência de Jensen-Shannon entre p_{data} e p_g menos uma constante. Portanto, minimizar $C(G)$ é equivalente a minimizar a divergência de Jensen-Shannon entre p_{data} e p_g e quando isso acontece, a divergência de Jensen-Shannon é zero e $C(G)$ atinge o valor $- \log 4$.

Dessa forma, o mínimo global de $C(G)$ é obtido se, e somente se, $p_g = p_{data}$. Neste ponto, $C(G)$ atinge o valor $- \log 4$. □

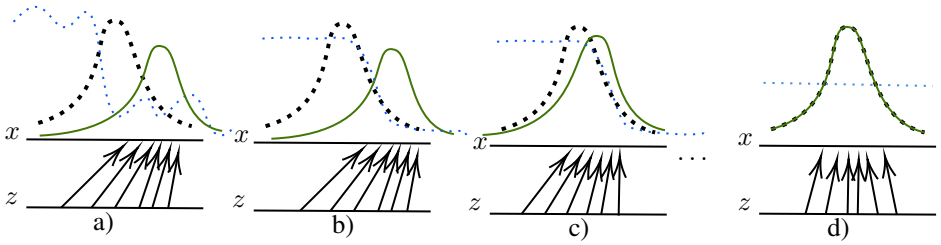

Figure 2: Redes adversárias generativas são treinadas atualizando simultaneamente a distribuição discriminativa (D) e a distribuição generativa (G). (a) G tenta imitar p_{data} enquanto D aprende a diferenciar entre dados reais e gerados. (b) D melhora na classificação de dados reais. (c) G ajusta-se baseado no gradiente de D para gerar dados que pareçam reais. (d) Após treinamento suficiente, G e D alcançam uma convergência onde $D(x) = \frac{1}{2}$, indicando que D não pode distinguir entre as distribuições. [1]

2.2.3 Convergência do Algorítimo 1

Proposição 2. *Dado um Gerador (G) e um Discriminador (D) com capacidade suficiente, isto é, redes neurais com arquitetura e parâmetros suficientes para modelar a complexidade da distribuição dos dados reais e para distinguir entre dados autênticos e gerados, respectivamente, então, a cada etapa do Algoritmo 1, se o Discriminador pode alcançar o seu ótimo em resposta a G, e p_g é atualizado de forma a melhorar o critério*

$$\mathbb{E}_{x\sim p_{data}}[\log D^*(x)] + \mathbb{E}_{x\sim p_g}[\log(1 - D^*(x))]$$

então a distribuição gerada p_g converge para a distribuição dos dados reais p_{data}.

Demonstração. Consideramos $V(G, D) = U(p_g, D)$ como a função de valor para o GAN, onde U é composta pelas expectativas dos logaritmos das probabilidades atribuídas pelo discriminador D. A função de valor V é o objetivo que o discriminador D tenta maximizar para um dado gerador G, e é definida como a soma de duas expectativas:

$$V(G, D) = \mathbb{E}_{x\sim p_{data}}[\log D(x)] + \mathbb{E}_{x\sim p_g}[\log(1 - D(x))]$$

Essa função é convexa em p_g, a distribuição de probabilidade que o gerador G está tentando aprender. A convexidade de U é devida a sua linearidade.

Quando a função $U(p_g, D)$ é maximizada em relação a D para um G fixo, obtemos o discriminador ótimo D^*. A afirmação crucial aqui é que as subderivadas do supremo de um conjunto de funções convexas incluem a derivada da função no ponto onde o máximo é atingido. Formalmente, isso é expresso como:

Se $f(x) = \sup_{\alpha\in A} f_\alpha(x)$ e cada $f_\alpha(x)$ é convexa em x, então as subderivadas de f em x incluem a derivada de $f_\beta(x)$ se $\beta = \arg\sup_{\alpha\in A} f_\alpha(x)$.

O significado desta afirmação é que, se você está tentando minimizar $f(x)$ usando o método de gradiente descendente e se em um certo ponto x a função $f_\beta(x)$ é a que contribui para o supremo, então a direção de descida mais acentuada (ou seja, a direção oposta à subderivada) é dada pela derivada negativa de $f_\beta(x)$ nesse ponto.

Isso é equivalente a realizar uma atualização de gradiente descendente para p_g no D^* ótimo para o G correspondente. Como o supremo de $U(p_g, D)$ é convexo em p_g com um único ótimo global, como visto no Teorema 1, essa atualização nos move na direção que reduz a diferença entre p_g e p_{data}, a distribuição real de dados.

Com atualizações suficientemente pequenas e precisas de p_g, garantimos que estamos seguindo o caminho de descida do gradiente sem "pular" o ótimo global devido a passos grandes demais. Isso nos leva à convergência de p_g para p_{data}, pois cada passo está orientado a diminuir a divergência entre a distribuição que G está gerando e a distribuição dos dados reais.

Portanto, se G e D têm capacidade suficiente para representar a distribuição dos dados e o processo de aprendizado permite que D seja otimizado e p_g seja atualizado corretamente em cada iteração do algoritmo, então p_g convergirá para p_{data}, concluindo a prova. □

3 Experimento

O experimento foi projetado para explorar a capacidade das GANs de gerar imagens que se assemelham a um conjunto de dados específico. O objetivo era entender a eficiência do modelo em aprender e replicar a distribuição de dados subjacente. Vale destacar que o processo de criação de imagens a partir do ruído z envolve o uso de redes neurais convolucionais (CNNs), que são fundamentais para capturar e gerar as características visuais das imagens de forma eficaz.

O dataset escolhido foi o Fashion-MNIST, que consiste em 70.000 imagens em escala de cinza de 10 categorias de roupas e acessórios, com cada imagem dimensionada em 28x28 pixels.

Os hiperparâmetros definidos para o experimento incluíram:

- Taxa de aprendizado: Determinada empiricamente para garantir a convergência efetiva durante o treinamento (10^{-4}).
- Tamanho do lote (batch size): Estabelecido para equilibrar a utilização de memória e a estabilidade do treinamento. (32)
- Número de épocas: Limitado a 200, uma quantidade suficiente para alcançar resultados razoáveis sem sobreajustes, dados os padrões relativamente simples do Fashion-MNIST.

A escolha dos hiperparâmetros foi guiada pela simplicidade do dataset, o que permitiu um treinamento eficiente em torno de 56 minutos.

Após 200 épocas, o modelo foi capaz de gerar imagens que refletiam as características visuais principais do dataset Fashion-MNIST:

Figure 3: Amostra de imagens do dataset Fashion-MNIST

Figure 4: Imagens geradas pelo modelo

Com o experimento apresentado vimos que os resultados, apesar da falta de métricas definidas no artigo original para avaliar objetivamente a fidelidade das imagens, parecem bons sob análise subjetiva. Futuras iterações poderiam explorar várias melhorias para aprimorar a qualidade das imagens geradas e a eficiência do modelo. Algumas sugestões incluem:

- Aumentar o número de épocas: Embora 200 épocas tenham sido suficientes para alcançar bons resultados com o Fashion-MNIST, aumentar o número de épocas poderia permitir que o gerador e o discriminador refinem suas capacidades, resultando em imagens de maior qualidade. Um número maior de épocas poderia ajudar especialmente a aprimorar detalhes finos e a consistência das características visuais nas imagens geradas.
- Ajuste de hiperparâmetros: Experimentar com diferentes taxas de aprendizado, tamanhos de lote ou mesmo a taxa de atualização entre o gerador e o discriminador (o parâmetro k em nosso treinamento) pode oferecer insights sobre configurações mais otimizadas para nosso modelo GAN.
- Datasets mais desafiadores: Migrar para datasets mais complexos e variados, como CIFAR-10 ou até mesmo conjuntos de dados de imagens de maior resolução, para testar a robustez e a adaptabilidade do modelo a desafios maiores.

Para uma compreensão mais profunda dos experimentos e para visualizar diretamente o código e os resultados, o repositório do trabalho está disponível através do seguinte link:

https://github.com/edilton-bs/Generative-Adversarial-Nets

4 Vantagens e desvantagens

Vantagens:

- O modelo dispensa a necessidade de inferência durante o processo de aprendizagem;
- Permite a integração de uma gama diversificada de funções, oferecendo flexibilidade no design do modelo;
- Existe uma vantagem estatística potencial, pois o gerador é atualizado através dos gradientes que passam pelo discriminador, e não diretamente pelos exemplos dos dados;

- É capaz de representar distribuições extremamente pontiagudas, até mesmo degeneradas.

Desvantagens:

- O modelo não oferece uma representação explícita da distribuição $p_g(x)$, o que pode limitar a compreensão e o controle sobre como os dados estão sendo gerados;
- O treinamento requer uma sincronização cuidadosa entre D (o discriminador) e G (o gerador), adicionando uma camada de complexidade ao processo;

5 Conclusão

Neste relatório, exploramos em profundidade as Generative Adversarial Networks (GANs), uma tecnologia que revolucionou o campo da inteligência artificial. Demonstramos como as GANs utilizam uma abordagem de aprendizado adversarial para gerar dados que são indistinguíveis dos reais, enfatizando a interação entre os modelos gerador e discriminador, que operam em um jogo dinâmico de otimização minimax.

O experimento conduzido com o dataset Fashion-MNIST serviu como uma aplicação prática das teorias discutidas no artigo de Goodfellow, ilustrando a capacidade das GANs de replicar e aprender com eficiência a distribuição de dados subjacente. No entanto, vale mencionar algumas críticas ao artigo original. Primeiramente, não define métricas objetivas para avaliar a qualidade das imagens geradas, deixando essa tarefa predominantemente para a análise subjetiva. Além disso, o artigo não explica detalhadamente o raciocínio por trás da decisão de usar o backpropagation do discriminador para treinar o gerador, uma escolha crucial para o sucesso da metodologia. Seria benéfico se o artigo contextualizasse melhor a época de sua proposta, destacando como, embora fosse possível treinar classificadores de alta qualidade, os geradores de imagens ainda enfrentavam grandes desafios, resultando frequentemente em imagens de baixa qualidade e aspecto borrado. A inovação dos GANs foi justamente utilizar o avanço dos classificadores para melhorar o treinamento dos geradores.

Para concluir, o conhecimento teórico e prático obtido neste estudo nos motiva a explorar datasets mais complexos e processos de treinamento mais robustos. Nosso objetivo é gerar resultados mais refinados e aplicáveis em contextos cotidianos, visando não apenas aprimorar a qualidade visual das imagens, mas também estabelecer métodos de avaliação mais concretos e menos subjetivos.

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. *arXiv preprint arXiv:1406.2661*, 2014. URL https://arxiv.org/abs/1406.2661.

[2] Normalized Nerd. The math behind generative adversarial networks clearly explained!, 2020. Disponível em: https://www.youtube.com/watch?v=Gib_kiXgnvA.