Redes Adversárias Generativas

Guilherme Bittencourt, Guilherme Fonseca, Yan Andrade

Junho 2023

1 Introdução

As redes generativas adversariais (GANs) oferecem uma abordagem para aprender representações profundas sem a necessidade de ter um conjunto de dados de treinamento extensivamente anotado. Elas alcançam isso por meio de um processo competitivo envolvendo um par de redes, permitindo a geração de sinais de retropropagação. As representações aprendidas pelas GANs têm ampla aplicabilidade em várias áreas, como síntese de imagens, edição semântica de imagens, transferência de estilo, melhoria da qualidade de imagens e classificação [1].

As GAN's são uma técnica emergente tanto para o aprendizado semissupervisionado quanto para o não supervisionado. Elas alcançam isso por meio da modelagem implícita de distribuições de dados de alta dimensão. Propostas em 2014 [2], as GANs podem ser caracterizadas pelo treinamento de um par de redes em competição uma com a outra. Uma analogia comum, adequada para dados visuais, é pensar em uma rede como um falsificador de arte e a outra como um especialista em arte. O falsificador, conhecido na literatura das GANs como o gerador, G, cria falsificações com o objetivo de gerar imagens realistas. O especialista, conhecido como o discriminador, D, recebe tanto as falsificações quanto as imagens reais e tem como objetivo distingui-las [3]. Ambos são treinados simultaneamente e em competição um com o outro. N

Na próxima seção, faremos uma breve análise sistemática da literatura sobre as arquiteturas mais famosas de GAN's propostas, assim como exemplos de algoritmos e suas aplicações. Em seguida, propomos uma avaliação experimental de duas arquiteturas comuns no cenário de geração de imagens, GAN e DCGAN [4].

2 Referencial Teórico

2.1 Fully Connected GAN

No artigo original de Goodfellow [2], foram utilizadas redes neurais totalmente conectadas para o gerador e o discriminador. Essa arquitetura foi aplicada em conjuntos de dados de imagens simples, como MNIST[5], CIFAR-10[6] e um conjunto de dados de rostos de Toronto. Os autores sugeriram otimizar o discriminador (D) em k passos e o gerador (G) em um único passo, para evitar o overfitting do discriminador. Na prática, os pesquisadores perceberam que a equação original usada para treinar o gerador pode causar problemas de desaparecimento do gradiente. Então, eles fizeram uma modificação no treinamento para maximizar uma função logarítmica. Essa modificação visa melhorar a qualidade das imagens geradas, tornando-as mais próximas das imagens reais. No entanto, essa mudança também trouxe um problema de assimetria para o modelo. O discriminador, que é responsável por distinguir entre as imagens reais e as geradas, utiliza uma técnica chamada maxout em sua arquitetura. Já o gerador usa uma combinação de ativações ReLU e sigmoid. Infelizmente, essa abordagem não se mostrou eficaz para lidar com tipos de imagens mais complexos, dificultando a geração de resultados realistas.

2.2 Semi-supervised

O Semi-supervised GAN (SGAN) [7] foi desenvolvido para abordar o desafio da aprendizagem semi-supervisionada, que está entre a aprendizagem supervisionada e não supervisionada. Ao contrário da abordagem supervisionada, que requer rótulos para todas as amostras, e da abordagem não supervisionada, que

não utiliza rótulos, a aprendizagem semi-supervisionada tem rótulos disponíveis apenas para um pequeno conjunto de exemplos. O SGAN utiliza um discriminador com múltiplas saídas, incluindo uma função softmax para classificar dados reais e uma função sigmoid para distinguir entre amostras reais e falsas. Os pesquisadores treinaram o SGAN no conjunto de dados MNIST e observaram melhorias tanto no discriminador quanto no gerador em comparação com o GAN original. No entanto, a arquitetura simples do discriminador de múltiplas saídas pode limitar a diversidade do modelo, especialmente porque os experimentos foram realizados apenas no conjunto de dados MNIST. Uma arquitetura mais complexa para o discriminador poderia potencialmente melhorar o desempenho do modelo em cenários mais desafiadores.

2.3 Bidirectional GAN

Os GANs tradicionais não conseguem aprender a reverter o processo de transformação dos dados de volta para a forma original. Mas o Bidirectional GAN (BiGAN) foi criado exatamente para isso [8]. Imagine que temos um codificador que transforma os dados reais em uma representação abstrata e um gerador que transforma essa representação abstrata de volta em dados reais. O objetivo do discriminador é identificar a diferença entre os dados originais e os dados gerados. No BiGAN, o codificador e o gerador trabalham em conjunto para enganar o discriminador, aprendendo a reverter um ao outro. Essa abordagem foi testada nos conjuntos de dados MNIST e ImageNet. Foram utilizadas técnicas de otimização para ajustar os parâmetros do modelo.

2.4 Conditional GAN

O CGAN (Conditional GAN) traz uma novidade ao adicionar informações extras, como rótulos de classe, tanto para o discriminador quanto para o gerador [9]. Essas informações extras são usadas para ajudar o modelo a aprender a gerar imagens específicas de acordo com as classes desejadas. Por exemplo, se estivermos trabalhando com o conjunto de dados MNIST, onde cada imagem é um número manuscrito, o CGAN receberia tanto o número que queremos gerar quanto um rótulo indicando qual número é. Dessa forma, o modelo é capaz de criar imagens mais precisas e coerentes para cada classe.

Os experimentos foram realizados com os conjuntos de dados MNIST e Yahoo Flickr Creative Common 100M (YFCC 100M). No caso do MNIST, o modelo foi treinado usando uma técnica chamada descida de gradiente estocástica (SGD), ajustando os parâmetros para aprender a melhor maneira de gerar as imagens corretas. No conjunto de dados YFCC 100M, foram utilizados os mesmos métodos de treinamento.

Essa abordagem do CGAN traz melhorias significativas na capacidade do modelo de gerar imagens realistas e coerentes, especialmente quando se trata de conjuntos de dados com várias classes ou características específicas. No entanto, é importante ressaltar que, em alguns casos, pode haver uma desconexão entre as informações codificadas e as imagens geradas, o que pode afetar a qualidade dos resultados.

2.5 Deep Convolutional GAN

O DCGAN (Deep Convolutional GAN) foi um trabalho pioneiro que introduziu uma nova arquitetura de rede neural chamada deconvolução para o gerador [10]. A deconvolução é uma técnica que permite visualizar características de redes neurais convolucionais e tem mostrado bom desempenho nesse contexto [11]. No caso do DCGAN, a deconvolução é usada para aumentar o tamanho das imagens geradas, possibilitando a geração de imagens de alta resolução usando GANs.

Existem algumas modificações importantes na arquitetura do DCGAN em comparação com o GAN convencional, o que traz benefícios para a modelagem de alta resolução e estabilidade do treinamento. Primeiro, o DCGAN substitui camadas de pooling por convoluções com passos para o discriminador e convoluções fracionadas para o gerador. Segundo, é utilizado o batch normalization tanto para o discriminador quanto para o gerador, o que ajuda a manter as amostras geradas e as amostras reais próximas de zero, ou seja, com estatísticas semelhantes. Terceiro, é usada a ativação ReLU para todas as camadas do gerador, exceto a camada de saída, que utiliza a função Tanh. Já para o discriminador, é usada a ativação Leaky ReLU em todas as camadas. Essa ativação evita que a rede fique presa em um estado de "inatividade", por exemplo, quando as entradas das camadas ReLU são menores que zero, pois o gerador recebe gradientes do discriminador.

Os DCGANs foram treinados com conjuntos de dados como LSUN, ImageNet e um conjunto de dados personalizado de faces. Todos os modelos foram treinados usando a técnica de descida de gradiente estocástica (SGD) com um tamanho de lote (batch size) de 128. Os pesos foram inicializados a partir de uma distribuição normal com média zero e desvio padrão de 0.02. Foi utilizado o otimizador Adam com uma taxa de aprendizado de 0.0002 e um termo de momentum de 0.5. A inclinação da Leaky ReLU foi definida como 0.2 para todos os modelos. Os modelos foram treinados usando imagens de tamanho 64×64 pixels.

O DCGAN é um marco muito importante na história das GANs, e a ideia da deconvolução se tornou uma base fundamental para a arquitetura principal dos geradores de GANs. No entanto, devido às limitações de capacidade do modelo e à otimização utilizada no DCGAN, ele é mais eficaz em imagens de baixa resolução e menos diversificadas.

2.6 Outras arquiteturas

Outros tipos de arquiteturas generativas incluem o InfoGAN [12], que aprende representações interpretáveis de forma não supervisionada ao maximizar a informação mútua entre variáveis condicionais e os dados gerados. O AC-GAN (Auxiliary Classifier GAN) [13] contém um classificador auxiliar na arquitetura. Já o LAPGAN (Laplacian Pyramid of Adversarial Networks) [14] utiliza operadores locais em várias escalas, mas com a mesma forma básica. O LAPGAN utiliza uma cascata de CNNs (Redes Neurais Convolucionais) dentro de um framework de pirâmide laplaciana para gerar imagens de alta qualidade.

3 Arquitetura Básica da GAN

Uma Rede Generativa Adversarial (GAN) consiste em dois componentes principais: o gerador e o discriminador. O objetivo é treinar esses dois componentes em conjunto para que o gerador seja capaz de produzir amostras sintéticas que sejam indistinguíveis das amostras reais, de acordo com o discriminador [2].

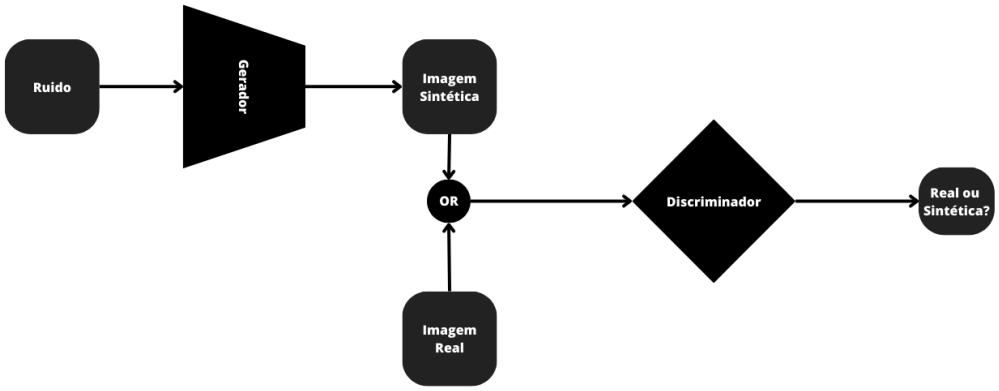


Figura 1: Arquitetura Básica da GAN

3.1 Treinamento

O gerador recebe como entrada um ruído aleatório (normalmente, um vetor de números) e o utiliza para gerar uma amostra sintética. Inicialmente, o gerador produzirá amostras de baixa qualidade, mas à medida

que o treinamento avança, ele aprenderá a gerar amostras mais realistas.

Por outro lado, o discriminador é treinado para distinguir entre amostras reais e sintéticas. Ele recebe como entrada uma amostra e avalia a probabilidade de ser real ou falsa. O discriminador é inicialmente treinado com amostras reais e sintéticas já existentes, para que ele possa aprender a distinguir entre elas.

Durante o treinamento, o gerador e o discriminador são atualizados em etapas alternadas. Primeiro, o discriminador é treinado em um lote de amostras reais e sintéticas, ajustando seus parâmetros para melhor distinguir entre elas. Em seguida, o gerador é treinado para produzir amostras que enganem o discriminador, ou seja, que o discriminador classifique erroneamente como reais.

Essa competição entre o gerador e o discriminador continua em ciclos sucessivos de treinamento, aprimorando tanto a capacidade do gerador de produzir amostras realistas quanto a habilidade do discriminador de distinguir entre amostras reais e sintéticas. Idealmente, ao final do treinamento, o gerador será capaz de produzir amostras sintéticas de alta qualidade, quase indistinguíveis das amostras reais.

4 Arquitetura da DCGAN

Para a nossa avaliação experimental, decidimos fazer uma comparação entre a arquitetura básica da GAN e uma arquitetura mais sofisticada que utiliza camadas convolucionais para extrair informações mais intrínsecas da imagem.

Uma DCGAN (Deep Convolutional GAN) é uma abordagem de redes neurais artificiais que visa gerar imagens realistas. Ela funciona através da interação entre dois componentes principais: o gerador e o discriminador.

O gerador recebe um vetor de ruído aleatório como entrada e o transforma em uma imagem sintética. Ele utiliza camadas convolucionais para aprender a mapear o vetor de ruído em uma representação visualmente semelhante a uma imagem real.

Por outro lado, o discriminador é responsável por distinguir entre imagens reais e imagens geradas pelo gerador. Ele recebe uma imagem como entrada e utiliza camadas convolucionais para extrair características e classificar se a imagem é real ou falsa.

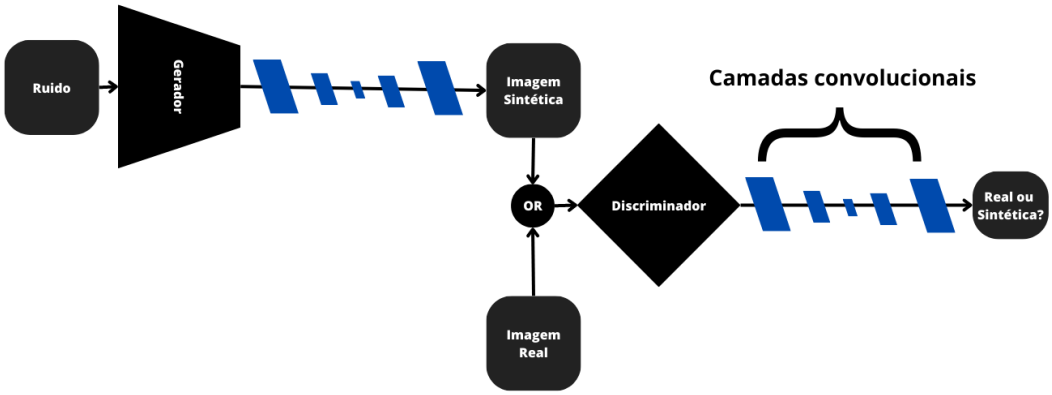


Figura 2: Arquitetura da DCGAN

5 Avaliação Experimental

Para realizar a análise experimental dos algoritmos, decidimos implementar tanto a GAN quanto a DCGAN descritas em seus respectivos artigos[6][10]. Disponibilizamos o código-fonte utilizado no nosso repositório no GitHub, que pode ser acessado através das referências. Para a base de dados, utilizamos a Fashion MNIST, que tem diversas imagens de roupas e acessórios. As imagens originais não possuem uma alta resolução e estão em escala de cinza para que seja mais fácil a obtenção dos resultados das redes.

5.1 Treinamento

Realizamos o treinamento das redes com diferentes configurações. Foram testadas três durações de treinamento: 20, 50 e 100 épocas. Além disso, variamos o tamanho do lote (batch size) entre 32 e 128. Utilizamos o otimizador ADAM [15], com taxa de aprendizado de 0.0002 e 0.0005, um weight decay de 0.00001 e betas entre 0.5 e 0.99. A função de perda utilizada foi a binary cross entropy.

A tabela abaixo mostra as diferentes configurações utilizadas durante o treinamento das redes:

Fine Tunning	**Valores**
Épocas de treinamento	20, 50, 100
Função de perda	Binary Cross Entropy (BCE)
Otimizador	ADAM
Batch Size	32a 128
Learning Rate	0.0002e 0.0005
Weight decay	$1e^{-3}$
Quantidade de filtros nas camadas convolucionais	100

Tabela 1: Configuração dos algoritmos.

5.2 Resultados

A seguir na figura 3, são apresentadas as imagens que ilustram a evolução da função de perda do gerador e do discriminador da GAN ao longo de 20, 50 e 100 épocas de treinamento. Essas imagens demonstram claramente o comportamento adversarial entre as redes, conforme proposto nos artigos. Observa-se que o erro do gerador diminui ao longo do tempo, enquanto o erro do discriminador aumenta. Essa dinâmica é indicativa do processo de aprendizado adversarial em que as redes estão envolvidas.

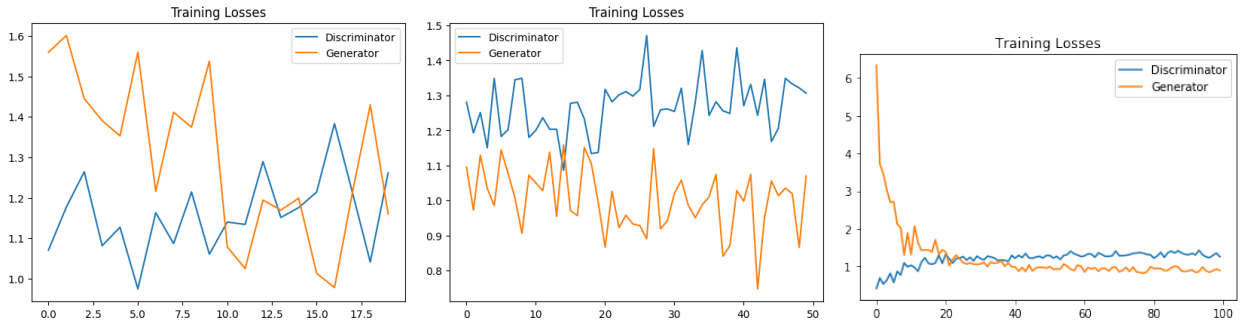


Figura 3: Evolução do BCE entre as épocas.

Na figura 4 são apresentadas as imagens resultantes do teste do algoritmo. É importante destacar que, devido à implementação simples da GAN e à limitação dos hiperparâmetros utilizados, é perceptível a presença

de ruídos nas imagens geradas. No entanto, mesmo com essas limitações, é possível vislumbrar o potencial criativo do algoritmo, evidenciando sua capacidade de geração de novas imagens. Apesar dos ruídos, as imagens produzidas demonstram a habilidade da GAN em criar conteúdo visual de forma promissora.

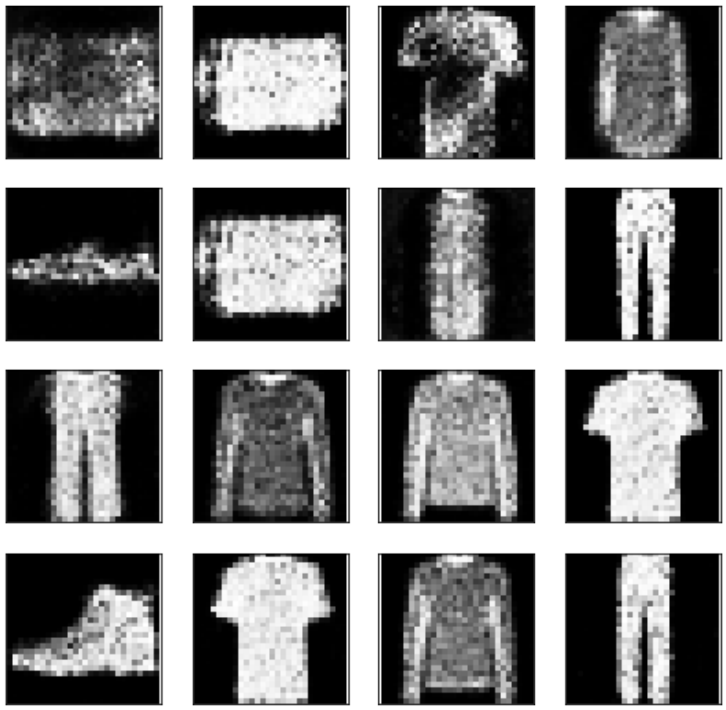


Figura 4: Imagens geradas pela GAN

No entanto, observa-se que a DCGAN apresentou gráficos de perda mais turbulentos em comparação à GAN, dificultando a identificação clara do comportamento adversarial entre a rede geradora e discriminadora. Essa observação é surpreendente, uma vez que a DCGAN utiliza camadas convolucionais para extração de características, o que se esperaria resultar em um comportamento semelhante ao da GAN em questão. No entanto, os resultados obtidos demonstraram uma dinâmica diferente entre as redes, indicando que a DCGAN pode requerer uma abordagem mais específica para atingir um equilíbrio satisfatório entre as duas redes.

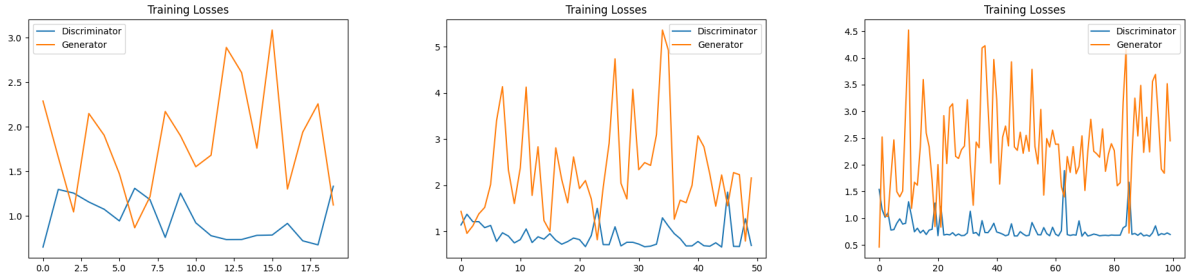


Figura 5: Evolução do BCE entre as épocas.

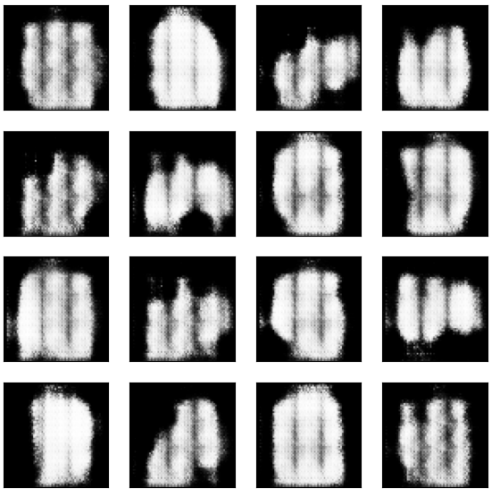
Ao analisar o gráfico, fica evidente que o discriminador convergiu para um limiar específico, sendo raramente ultrapassado pelo gerador. Existem algumas possíveis explicações para esse comportamento. Primeiramente, a arquitetura simples implementada pode limitar a capacidade da rede em aprender representações

mais complexas. Além disso, o overfitting nas redes devido ao treinamento em uma base de dados específica pode afetar o desempenho do algoritmo. Por fim, a base de dados utilizada pode conter imagens com muito ruído e padrões, o que torna o processo de aprendizado mais desafiador.

Na Figura 6, são apresentados os resultados da DCGAN na geração de imagens. É possível observar que, apesar dos desafios mencionados anteriormente, a DCGAN ainda conseguiu gerar imagens com maior nível de semelhança e menos ruidos em relação as geradas pela GAN tradicional. Esses resultados mostram o potencial da DCGAN em sintetizar novas amostras, embora haja espaço para melhorias e otimizações.

Figura 6: Imagens geradas pela DCGAN

Como parte da análise final, apresentamos os resultados das imagens geradas pela rede DCGAN após o treinamento com 100 épocas. Nas imagens a seguir, você pode ver a progressão das imagens geradas ao longo do tempo, representando as épocas 0, 20, 40, 60, 80 e 100.



Observando a sequência de imagens, é possível notar uma evolução na qualidade e na semelhança das imagens geradas em relação aos dados originais. Essa progressão indica que a DCGAN está aprendendo e refinando seus parâmetros ao longo do treinamento.

No entanto, ainda podem ser identificados ruídos e imperfeições nas imagens geradas, o que sugere a necessidade de ajustes e otimizações adicionais no algoritmo. Essa análise visual das imagens é fundamental para compreender o desempenho e as limitações da DCGAN.

Esses resultados destacam a importância de continuar explorando e aprimorando as técnicas de redes generativas como a DCGAN, buscando obter resultados cada vez mais realistas e de alta qualidade.

6 Conclusão

De fato, as redes generativas demonstram uma capacidade impressionante de criar imagens a partir de cálculos matemáticos, reproduzindo com precisão imagens existentes. Esse campo das redes generativas tem recebido grande atenção na literatura devido às inúmeras aplicações que oferece, como criação de conjuntos de dados e geração de arte digital.

A capacidade dessas redes em gerar imagens realistas tem despertado o interesse dos usuários da internet, assim como de ferramentas online, como o Mifjourney e Kapwing, que utiliza inteligência artificial para gerar imagens a partir de texto. Essas tecnologias têm se tornado cada vez mais populares e acessíveis, permitindo que os usuários explorem sua criatividade e produzam conteúdo visual de maneira rápida e fácil.

É importante ressaltar que, embora as redes generativas tenham mostrado resultados impressionantes, ainda há desafios a serem enfrentados, como a melhoria da qualidade das imagens geradas e a prevenção do surgimento de artefatos indesejados. No entanto, os avanços nesse campo prometem abrir novas possibilidades e revolucionar a forma como criamos e interagimos com o conteúdo visual.

Como resultado, as redes generativas têm se tornado uma ferramenta poderosa tanto para profissionais criativos quanto para usuários comuns, proporcionando oportunidades inovadoras de expressão e produção de conteúdo visualmente cativante.

7 Bibliografias

Link para o repositório: https://github.com/guibitten03/UFSJ—GAN-IA-/tree/main

[1] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., Bharath, A. A. (2018). Generative adversarial networks: An overview. IEEE signal processing magazine, 35(1), 53-65.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, andYoshua Bengio. 2014. Generative adversarial nets. In Advances in Neural Information Processing Systems. 2672–2680..

[3] Aggarwal, A., Mittal, M., Battineni, G. (2021). Generative adversarial network: An overview of theory and applications. International Journal of Information Management Data Insights, 1(1), 100004.

[4] Wang, Z., She, Q., Ward, T. E. (2021). Generative adversarial networks in computer vision: A survey and taxonomy. ACM Computing Surveys (CSUR), 54(2), 1-38.

[5]Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. 1998. Gradient-based learning applied to document recognition. Proc. IEEE 86, 11 (1998), 2278–2324.

[6] Alex Krizhevsky and Geoffrey Hinton. 2009. Learning Multiple Layers of Features from Tiny Images. Technical Report. Citeseer.

[7] Augustus Odena. 2016. Semi-supervised Learning with Generative Adversarial Networks. arXiv:1606.01583. Retrieved from https://arxiv.org/abs/1606.01583.

[8] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. 2016. Adversarial Feature Learning. arXiv:1605.09782. Retrieved from https://arxiv.org/abs/1605.0978.

[9] Mehdi Mirza and Simon Osindero. 2014. Conditional Generative Adversarial Nets. arXiv:1411.1784. Retrieved from https://arxiv.org/abs/1411.1784.

[10] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv:1511.06434. Retrieved from https://arxiv.org/abs/1511.06434.

[11] Matthew D. Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision. Springer, 818–833.

[12] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. 2016. InfoGAN: Interpretablerepresentation learning by information maximizing generative adversarial nets. In Advances in Neural Information Processing Systems. 2172–2180.

[13] Augustus Odena, Christopher Olah, and Jonathon Shlens. 2017. Conditional image synthesis with auxiliary classifier gans. In Proceedings of the 34th International Conference on Machine Learning, Vol. 70. 2642–2651.

[14] Emily L. Denton, Soumith Chintala, Rob Fergus, et al. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. In Advances in Neural Information Processing Systems. 1486–1494.

[15] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization."arXiv preprint arXiv:1412.6980 (2014).