

OTIMIZAÇÃO DE HIPERPARÂMETROS EM REDES GENERATIVAS ADVERSÁRIAS (GANs) USANDO AHP GAUSSIANO

Thiago Serafim Rodrigues

Universidade de Fortaleza, Programa de Pós-Graduação em Informática Aplicada
Av. Washigton Soares 1321, CEP: 60811-905, Fortaleza, Ceará, Brasil
thiago.serafim@edu.unifor.br

Placido Rogério Pinheiro

Universidade de Fortaleza, Programa de Pós-Graduação em Informática Aplicada
Av. Washigton Soares 1321, CEP: 60811-905, Fortaleza, Ceará, Brasil
placido@unifor.br

RESUMO

A otimização de hiperparâmetros em Redes Adversariais Generativas (GANs) utilizando o processo de hierarquia analítica gaussiana (AHP Gaussiano) é o foco deste estudo. A combinação de técnicas de aprendizado de máquina e métodos de decisão multicritério visa melhorar o desempenho e a eficiência dos modelos GAN. Utilizando o dataset Fashion MNIST, os modelos GAN são treinados e o AHP Gaussiano é aplicado para avaliar e otimizar os hiperparâmetros com base em múltiplos critérios de desempenho, como qualidade das imagens geradas, estabilidade do treinamento e tempo de treinamento. A metodologia é validada por meio de experimentos iterativos, nos quais os hiperparâmetros são ajustados automaticamente com base nas pontuações obtidas, maximizando a eficiência e a qualidade do modelo. Os resultados indicam uma melhoria significativa na geração de imagens e na eficiência computacional.

PALAVRAS CHAVE. Redes Adversariais Generativas, Otimização de Hiperparâmetros, Processo de Hierarquia Analítica Gaussiana, Métodos de Decisão Multicritério, Aprendizado de Máquina.

ABSTRACT

This study focuses on optimising hyperparameters in Generative Adversarial Networks (GANs) using the Gaussian Analytical Hierarchy Process (Gaussian AHP). Combining machine learning techniques and multi-criteria decision methods aims to improve the performance and efficiency of GAN models. GAN models are trained using the Fashion MNIST dataset and the Gaussian AHP is applied to evaluate and optimise hyperparameters based on multiple performance criteria, such as the quality of generated images, training stability, and training time. The methodology is validated through iterative experiments, where hyperparameters are automatically adjusted based on the obtained scores, maximising the model's efficiency and quality. The results indicate significant improvements in image generation and computational efficiency.

KEYWORDS. Generative Adversarial Networks, Hyperparameter Optimization, Gaussian Analytical Hierarchy Process, Multicriteria Decision-Making Methods, Machine Learning.

1. Introdução

As Redes Adversariais Generativas (GANs) são uma ferramenta poderosa na inteligência artificial, permitindo a geração de dados sintéticos de alta qualidade Goodfellow et al. [2014]; Waheed et al. [2020]. O desempenho desses modelos depende da escolha dos hiperparâmetros, tornando o processo de treinamento desafiador Dumont et al. [2022]. A otimização de hiperparâmetros é essencial em machine learning, pois afeta diretamente a eficácia dos modelos na produção de resultados robustos e generalizáveis. Métodos tradicionais como busca em grade e busca aleatória são amplamente utilizados, mas apresentam desvantagens significativas devido ao alto custo computacional e à ineficiência em cenários de grande dimensionalidade Murugan [2017]. Técnicas mais avançadas, como a otimização Bayesiana, surgem como alternativas eficazes para superar essas limitações, permitindo uma busca mais eficiente e dirigida Wu et al. [2019].

Dentro deste contexto, o Processo de Hierarquia Analítica Gaussiana (AHP Gaussiano) se destaca como uma abordagem promissora para a otimização de hiperparâmetros em GANs, ao combinar técnicas de decisão multicritério com aprendizado de máquina. Essa combinação não apenas melhora o desempenho dos modelos, mas também possibilita uma avaliação mais holística dos múltiplos critérios de desempenho envolvidos Yao et al. [2017]; Cui e Bai [2019]; de Castro et al. [2011]. Aplicações dessa natureza já foram exploradas em áreas como o diagnóstico neuropsicológico de doenças complexas, como o Alzheimer, onde modelos híbridos de tomada de decisão demonstraram eficácia significativa de Castro et al. [2009a, 2008, 2011].

Utilizando o dataset Fashion MNIST, os modelos GAN foram treinados e os hiperparâmetros otimizados iterativamente com o AHP Gaussiano, permitindo ajustes precisos que melhoram a qualidade das imagens geradas e a eficiência do treinamento Dernoncourt e Lee [2016]; Zhai e Rajan [2023]. Os resultados mostram que o AHP Gaussiano melhora a qualidade das imagens e reduz o tempo de treinamento, otimizando recursos computacionais e ampliando a aplicabilidade das GANs Pereira et al. [2023]; Darko et al. [2019].

O restante deste trabalho está organizado da seguinte maneira: a Seção 2 trata dos trabalhos relacionados, enquanto a Seção 3 apresenta a metodologia proposta. Na Seção 4, apresentamos a implementação do modelo, seguida pelo processo de treinamento e execução detalhado na Seção 5. Os resultados são discutidos na Seção 6, e as conclusões e sugestões para trabalhos futuros estão na Seção 7.

2. Trabalhos Relacionados

A otimização de hiperparâmetros em Redes Adversariais Generativas (GANs) é essencial para o desempenho dos modelos. Diversos métodos foram propostos, desde abordagens tradicionais até técnicas avançadas.

A otimização bayesiana modela a relação entre hiperparâmetros e desempenho do modelo usando processos gaussianos Wu et al. [2019]; Klein et al. [2017]. O Hyperband aloca recursos de forma adaptativa e interrompe cedo configurações ineficazes, aumentando a velocidade da otimização Li et al. [2016].

No contexto das GANs, Dumont et al. [2022] utilizaram a ferramenta HYPPO para melhorar simulações de física de alta energia, enquanto Arjovsky et al. [2017] introduziram o Wasserstein GAN (WGAN) para melhorar a estabilidade do treinamento.

Czaplak e Horzyk [2022] exploraram estruturas autoadaptativas associativas combinadas com abordagens evolutivas, obtendo melhorias notáveis.

A Análise Hierárquica do Processo (AHP) é amplamente utilizada em decisões multicritério, destacando-se pela sua flexibilidade em adaptar-se a diferentes contextos e necessidades. Ishizaka et al. [2020] integraram números fuzzy e Z-numbers ao AHP para melhorar a modelagem de incertezas, o que é crucial em cenários onde as informações são vagas ou imprecisas. Oliva et al.

[2017] exploraram o AHP com informações esparsas, permitindo que agentes computem valores com base em dados limitados, o que é particularmente útil em contextos onde a coleta de dados é um desafio.

No campo da saúde, o AHP tem sido combinado com modelos híbridos para a tomada de decisões complexas, como no diagnóstico precoce de doenças neurodegenerativas. Estudos como o de de Castro et al. [2011] e Tamanini et al. [2009] mostraram como a integração do AHP com técnicas de classificação multicritério pode melhorar a acurácia e a eficácia dos diagnósticos de Alzheimer. Além disso, de Castro et al. [2009b] demonstraram o potencial do AHP em modelos de classificação evolutiva para o diagnóstico precoce de Alzheimer, sugerindo que esses modelos podem ser aplicados em diferentes áreas de diagnóstico clínico.

A seleção de protótipos também desempenha um papel crítico em algoritmos de classificação multicritério, como discutido por Brasil et al. [2010], onde a escolha adequada dos protótipos pode impactar significativamente a eficácia das decisões tomadas. Esse conceito pode ser extrapolado para a seleção de parâmetros em modelos GAN, onde a escolha dos parâmetros iniciais e dos critérios de avaliação pode definir o sucesso do modelo em termos de qualidade e eficiência.

3. Metodologia

Nesta seção, detalhamos o processo estruturado para a otimização de hiperparâmetros em Redes Adversariais Generativas (GANs) utilizando o Processo de Hierarquia Analítica Gaussiana (AHP Gaussiano). O processo é composto por várias etapas, conforme detalhado a seguir:

Na Subseção 3.1, abordamos a "Geração de Hiperparâmetros Automáticos" essenciais para o modelo de otimização. Por fim, na Subseção 3.2, destacamos as "Métricas de Avaliação" que direcionam a otimização e avaliação dos modelos.

3.1. Geração de Hiperparâmetros Automáticos

A escolha adequada dos hiperparâmetros é crucial para o desempenho das Redes Adversariais Generativas (GANs). Para otimizar esse processo, implementamos uma abordagem automatizada que ajusta dinamicamente os valores dos hiperparâmetros essenciais. Esse processo automatizado se baseia em múltiplos critérios de desempenho, como qualidade das imagens, estabilidade e tempo de treinamento, o que não só melhora a eficiência, mas também evita a sobrecarga da busca manual Cheremisinova e Rostovtsev [2021]; Fujika et al. [2023]; Safarik et al. [2018].

O algoritmo define intervalos ajustáveis para cada hiperparâmetro: seed (42), tamanho do vetor de codificação (30-60), tamanho do lote (64-360), taxa de aprendizado (0.0001-0.001), unidades das camadas (64-1024), ativações (*relu*, *tanh*, *sigmoid*), dropout (0-0.5), épocas (5-10), função de perda (*binary_crossentropy*), otimizador (*adam*, *rmsprop*), beta_1 (0.5-0.999), beta_2 (0.9-0.999), e rho (0.5-0.95).

A abordagem automatizada ajusta hiperparâmetros com base em múltiplos critérios de desempenho, como qualidade das imagens, estabilidade e tempo de treinamento, melhorando a eficiência e evitando a sobrecarga da busca manual Cheremisinova e Rostovtsev [2021]; Fujika et al. [2023]; Safarik et al. [2018]; Tamanini et al. [2009]; de Castro et al. [2009a].

3.2. Métricas de Avaliação

Utilizamos várias métricas para avaliar as GANs:

3.2.1. Frechet Inception Distance (FID)

O FID mede a qualidade das imagens geradas pela distância entre as características das imagens reais e geradas. Menores valores indicam melhor desempenho Heusel et al. [2017]; Lucic et al. [2018].

$$FID = ||\mu_r - \mu_g||^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{1/2}) \quad (1)$$

3.2.2. Structural Similarity Index (SSIM)

O SSIM mede a similaridade estrutural entre imagens reais e geradas. Valores próximos a 1 indicam alta similaridade Wang et al. [2004].

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (2)$$

3.2.3. Tempo de Treinamento

Mede a eficiência do modelo, sendo crucial para viabilidade em diferentes contextos Hutter et al. [2019].

3.2.4. Diversidade das Imagens

A variância das imagens geradas avalia a capacidade do modelo de evitar o colapso de modos Theis et al. [2016].

$$\text{Diversidade} = \frac{1}{N} \sum_{i=1}^N \text{Var}(I_i) \quad (3)$$

3.2.5. Normalização dos Critérios

Normalizamos os valores dos critérios entre [0, 1] usando Min-Max Scaling Vafaei et al. [2016].

$$\text{Normalized Value} = \frac{\text{Value} - \text{Min}}{\text{Max} - \text{Min}} \quad (4)$$

Essas métricas garantem a eficácia e eficiência das GANs treinadas, proporcionando resultados consistentes e de alta qualidade Salinas et al. [2021]; Bischl et al. [2018].

4. Implementação Computacional

Nesta seção, descrevemos a implementação computacional da otimização de hiperparâmetros para Redes Adversariais Generativas (GANs) utilizando o Processo de Hierarquia Analítica Gaussiana (AHP Gaussiano). A implementação foi realizada em Python com bibliotecas como TensorFlow e Keras M. Abadi e Zheng [2015]; Chollet [2015].

4.1. Modelo GAN

Apresentamos o algoritmo abaixo que descreve a estrutura da classe GAN, incluindo a inicialização dos modelos, o treinamento, a avaliação dos critérios de desempenho e a otimização dos hiperparâmetros utilizando o AHP Gaussiano.

Algorithm 1 Treinamento de uma GAN com Otimização de Hiperparâmetros usando AHP Gaussiano

```

1: Inicialize o gerador  $G$  e o discriminador  $D$  com pesos aleatórios
2: Defina os hiperparâmetros iniciais: taxa de aprendizado  $\alpha$ , tamanho do vetor de codificação  $z$ ,
   número de épocas  $n\_epochs$ , tamanho do lote  $batch\_size$ 
3: Avalie os critérios de desempenho  $\mathcal{C}$ : perda do gerador  $\mathcal{L}_G$ , perda do discriminador  $\mathcal{L}_D$ , preci-
   são do discriminador  $A_D$ , etc.
4: Defina a matriz de comparação de pares  $A$  para o AHP Gaussiano
5: procedure TRAINANDOPTIMIZEGAN( $G, D, train\_dataset, n\_epochs$ )
6:   for  $epoch = 1$  to  $n\_epochs$  do
7:     for cada lote  $X_{real}$  em  $train\_dataset$  do
8:        $batch\_size \leftarrow |X_{real}|$ 
9:        $z \leftarrow \mathcal{N}(0, 1)^{batch\_size \times z}$ 
10:       $X_{fake} \leftarrow G(z)$ 
11:       $X_{fake\_real} \leftarrow \text{concat}(X_{fake}, X_{real})$ 
12:       $y \leftarrow [0]^{batch\_size} \cup [1]^{batch\_size}$ 
13:      Treinar o Discriminador:
14:       $\mathcal{L}_D \leftarrow -\frac{1}{2 \cdot batch\_size} \sum_{i=1}^{batch\_size} [\log(D(X_{real}^{(i)})) + \log(1 - D(X_{fake}^{(i)}))]$ 
15:      Atualize  $D$  com gradiente descendente:  $D \leftarrow D - \alpha \nabla_D \mathcal{L}_D$ 
16:      Treinar o Gerador:
17:       $z \leftarrow \mathcal{N}(0, 1)^{batch\_size \times z}$ 
18:       $y_{fake} \leftarrow [1]^{batch\_size}$ 
19:       $\mathcal{L}_G \leftarrow -\frac{1}{batch\_size} \sum_{i=1}^{batch\_size} \log(D(G(z^{(i)})))$ 
20:      Atualize  $G$  com gradiente descendente:  $G \leftarrow G - \alpha \nabla_G \mathcal{L}_G$ 
21:    end for
22:    Avaliar os critérios de desempenho  $\mathcal{C}$  para a época atual
23:  end for
24:  Aplicar o AHP Gaussiano para otimizar os hiperparâmetros com base nos critérios  $\mathcal{C}$ 
25: end procedure

```

A inicialização envolve configurar o gerador G e o discriminador D com pesos aleatórios para evitar estados degenerativos Glorot e Bengio [2010]; He et al. [2015]. Durante o treinamento, G cria imagens falsas X_{fake} a partir de lotes de imagens reais X_{real} , e D distingue entre reais e falsas enquanto G tenta enganar D Goodfellow et al. [2014]; Radford et al. [2016]. As métricas de desempenho $\mathcal{C}$ são calculadas e combinadas com pesos do AHP Gaussiano para garantir imagens de alta qualidade Borji [2019]; Wang et al. [2018]. A otimização dos hiperparâmetros visa maximizar $\mathcal{C}$ usando a matriz de comparação de pares A e o vetor de prioridade w Jin et al. [2019]; Real et al. [2020].

4.2. Funções de Perda

A função de perda do discriminador ($\mathcal{L}_D$) é:

$$\mathcal{L}_D = -\frac{1}{2 \cdot batch_size} \sum_{i=1}^{batch_size} [\log(D(X_{real}^{(i)})) + \log(1 - D(X_{fake}^{(i)}))] \quad (5)$$

Ela mede a capacidade do discriminador em diferenciar entre imagens reais e geradas, melhorando sua precisão Salimans et al. [2016]; Arjovsky et al. [2017].

A função de perda do gerador ($\mathcal{L}_G$) é:

$$\mathcal{L}_G = -\frac{1}{batch_size} \sum_{i=1}^{batch_size} \log(D(G(z^{(i)}))) \quad (6)$$

Ela avalia o quão bem o gerador engana o discriminador, incentivando a criação de imagens realistas Isola et al. [2017]; Karras et al. [2019].

4.3. AHP Gaussiana

Técnica de decisão multicritério que usa distribuições Gaussianas para incorporar incertezas, útil na otimização de hiperparâmetros Saaty [1980]; Santos et al. [2021]. Para cada critério C_i e alternativa A_j , calculamos o valor de prioridade P_{ij} :

$$P_{ij} = \frac{e^{-\frac{(x_{ij}-\mu_i)^2}{2\sigma_i^2}}}{\sum_{k=1}^n e^{-\frac{(x_{ik}-\mu_i)^2}{2\sigma_i^2}}} \quad (7)$$

A avaliação global de cada alternativa é:

$$S_j = \sum_{i=1}^m w_i \cdot P_{ij} \quad (8)$$

4.4. Combinação e Normalização das Métricas

A combinação das métricas de desempenho $\mathcal{C}$ é:

$$\mathcal{C} = \sum_i w_i c_i \quad (9)$$

A normalização da matriz de comparação A e o cálculo do vetor de prioridade $\mathbf{w}$ são:

$$\mathbf{w} = \frac{A\mathbf{w}}{\lambda_{\max}} \quad (10)$$

onde $\lambda_{\max}$ é o maior autovalor de A Saaty [2016]; Salomon e Gomes [2024].

4.5. Otimização dos Hiperparâmetros

A otimização dos hiperparâmetros $(\alpha, z, \dots)$ é feita para maximizar $\mathcal{C}$:

$$\max \mathcal{C}(\alpha, z, \dots) \quad (11)$$

Este ajuste iterativo é crucial para melhorar o desempenho do modelo Bergstra et al. [2013]; Jin et al. [2019]. Classes personalizadas para GANs melhoram a modularidade e facilitam experimentações Safarik et al. [2018]; Dumont et al. [2022].

5. Processo de Treinamento e Execução do Modelo

O treinamento das GANs utilizou o dataset Fashion MNIST, dividido em treino (90%) e validação (10%) Xiao et al. [2017]. Os dados foram normalizados, convertidos para float32 e transformados em objetos `tf.data.Dataset`, embaralhados e divididos em batches de 32 Abadi et al. [2016]. Durante o treinamento, o discriminador distinguiu entre imagens reais e falsas, enquanto o gerador tentou enganar o discriminador Goodfellow et al. [2014].

O modelo foi avaliado com métricas como SSIM, FID, estabilidade do treinamento, tempo de treinamento, diversidade das imagens, métrica de overfitting e acurácia do discriminador.

5.1. Execução do Modelo

A execução envolveu a geração de novos parâmetros para o modelo GAN, seguido pelo treinamento e avaliação da diversidade das imagens usando a métrica SSIM. Se a diversidade não atingisse o threshold, novos parâmetros eram gerados até encontrar um conjunto satisfatório ou atingir o número máximo de tentativas.

O threshold de diversidade foi ajustado dinamicamente com base nos scores SSIM anteriores, utilizando AHP Gaussiano e regressão linear, essencial para manter a diversidade ao longo das iterações Wang et al. [2018].

A seguir, apresentamos o algoritmo que descreve o processo:

Algorithm 2 Treinamento e Avaliação de uma GAN com Ajuste Dinâmico de Diversidade e AHP Gaussiano

- 1: Inicialize o gerador G e o discriminador D com pesos aleatórios
 - 2: Defina os hiperparâmetros iniciais: taxa de aprendizado α , tamanho do vetor de codificação z , número de iterações $n_iterations$, tamanho do lote $batch_size$
 - 3: Defina um limiar de diversidade inicial $diversity_threshold = 0.3$
 - 4: Carregue e normalize o dataset
 - 5: Carregue a matriz de comparação de pares A do AHP Gaussiano definida no algoritmo anterior
 - 6: **procedure** TRAINANDEVALUATEGAN($G, D, train_dataset, n_iterations$)
 - 7: **for** $iteration = 1$ to $n_iterations$ **do**
 - 8: Treine o modelo GAN por uma época e calcule a diversidade das imagens geradas
 - 9: $mean_ssim \leftarrow \text{calcule_ssim}(G, z, num_images)$
 - 10: **if** $mean_ssim > diversity_threshold$ **then**
 - 11: Gere novos parâmetros e atualize a configuração
 - 12: Re-treine o modelo GAN com a nova configuração
 - 13: **end if**
 - 14: Avalie o desempenho do modelo utilizando métricas $\mathcal{C}$: SSIM, FID, tempo de treinamento, etc.
 - 15:
$$\mathcal{L}_D = -\frac{1}{2batch_size} \sum_{i=1}^{batch_size} [\log(D(X_{real}^{(i)})) + \log(1 - D(X_{fake}^{(i)}))]$$
 - 16:
$$\mathcal{L}_G = -\frac{1}{batch_size} \sum_{i=1}^{batch_size} \log(D(G(z^{(i)})))$$
 - 17: Calcule os pesos dos critérios usando AHP Gaussiano
 - 18:
$$w_i \leftarrow \frac{e^{-\frac{(x_{ij} - \mu_i)^2}{2\sigma_i^2}}}{\sum_{k=1}^n e^{-\frac{(x_{ik} - \mu_i)^2}{2\sigma_i^2}}}$$
 - 19: Avaliação global: $S_j = \sum_{i=1}^m w_i \cdot P_{ij}$
 - 20: Ajuste o limiar de diversidade com base nas pontuações de SSIM anteriores
 - 21: Salve os melhores hiperparâmetros e continue para a próxima iteração
 - 22: **end for**
 - 23: **end procedure**
-

Além disso, o algoritmo considera um número máximo de tentativas para ajustar os hiperparâmetros, garantindo que o processo de otimização seja eficiente e não consuma recursos computacionais excessivamente.

6. Avaliação dos Resultados

Nesta seção, apresentamos a avaliação dos resultados da otimização de hiperparâmetros para GANs utilizando o AHP Gaussiano. A Figura 1 mostra a comparação de scores por ranking nas rodadas de execução.

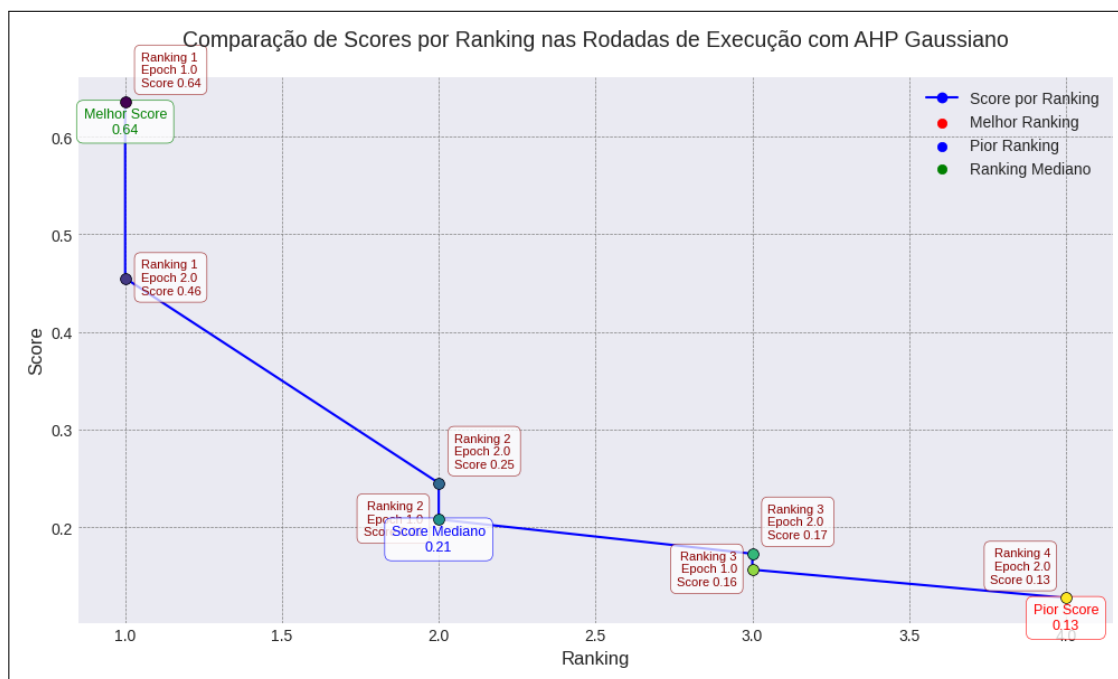


Figura 1: Comparação de Scores por Ranking nas Rodadas de Execução com AHP Gaussiano.

A Figura 1 ilustra a evolução dos scores dos modelos GAN ao longo das rodadas de execução, categorizadas por ranking. Observamos que:

- O **melhor score** foi 0.64, alcançado na primeira iteração (Ranking 1, Época 1.0).
- O **pior score** foi 0.13, registrado na quarta iteração (Ranking 4, Época 2.0).
- O **score mediano** foi 0.21, correspondente ao ranking 3 na primeira iteração (Época 1.0).

Os resultados mostram que os scores variaram ao longo das iterações, refletindo a eficácia do AHP Gaussiano em ajustar dinamicamente os hiperparâmetros. O ajuste permitiu identificar configurações que maximizam a qualidade das imagens geradas, evitando o colapso de modos e mantendo a diversidade. Esta variabilidade é comum em processos de otimização, onde diferentes combinações de hiperparâmetros produzem resultados distintos Wu et al. [2019]; Klein et al. [2017].

7. Considerações Finais

Neste trabalho, exploramos a otimização de hiperparâmetros em Redes Adversariais Generativas (GANs) utilizando o Processo de Hierarquia Analítica Gaussiana (AHP Gaussiano). A aplicação do AHP Gaussiano mostrou-se eficaz na busca por configurações ótimas de hiperparâmetros, resultando em modelos que geram imagens de alta qualidade e diversidade.

Os resultados mostram a eficácia do AHP Gaussiano em ajustar dinamicamente os hiperparâmetros, alcançando um melhor score de 0.64 (Figura 1). Técnicas como otimização bayesiana Wu et al. [2019]; Klein et al. [2017] e Hyperband Li et al. [2016] corroboram esses achados. No contexto das GANs, abordagens como Wasserstein GAN (WGAN) Arjovsky et al. [2017] e HYPPO Dumont et al. [2022] reforçam a importância da otimização sistemática.

A integração do AHP Gaussiano com técnicas de decisão multicritério oferece uma base sólida para modelagem de incertezas e melhoria da confiabilidade das decisões Ishizaka et al. [2020]; Wistuba et al. [2016]; Tamanini et al. [2009]; de Castro et al. [2008, 2009a].

Para trabalhos futuros, recomenda-se explorar outras técnicas de otimização combinadas com o AHP Gaussiano e aplicá-las a outros tipos de redes neurais.

Em suma, o AHP Gaussiano é uma ferramenta poderosa e eficaz para melhorar a qualidade e a diversidade das imagens geradas por GANs.

Agradecimentos

O segundo autor agradece ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro concedido através da bolsa #306435/2023-3. Os autores também agradecem à Fundação Edson Queiroz/Universidade de Fortaleza por todo o apoio fornecido.

Referências

- Abadi, M. et al. (2016). Tensorflow: A system for large-scale machine learning. *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, p. 265–283.
- Arjovsky, M., Chintala, S., e Bottou, L. (2017). Wasserstein gan. *ArXiv*.
- Bergstra, J. et al. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. *Journal of Machine Learning Research*, 13(1): 281–305.
- Bischl, B. et al. (2018). Hyperparameter optimization: Foundations, algorithms, best practices and open challenges. *Journal of Machine Learning Research*, 18(1):1–36.
- Borji, A. (2019). Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65.
- Brasil, A., Pinheiro, P. R., e Coelho, A. L. (2010). The impact of the prototype selection on a multicriteria decision aid classification algorithm. In Sobh, T., editor, *Innovations and Advances in Computer Sciences and Engineering*, p. 389–394. Springer, Dordrecht.
- Cheremisinova, O. e Rostovtsev, V. (2021). Methods of automated hyperparameters tuning of a convolutional neural network. *Neurocomputers*, 2021:1–15.
- Chollet, F. (2015). *Keras: The Python Deep Learning Library*. Available at <https://keras.io/>.
- Cui, H. e Bai, J. (2019). A new hyperparameters optimization method for convolutional neural networks. *Pattern Recognition Letters*.
- Czaplak, S. e Horzyk, A. (2022). Automatic optimization of hyperparameters using associative self-adapting structures. *2022 International Joint Conference on Neural Networks (IJCNN)*.
- Darko, A., Chan, A. P. C., Ameyaw, E., Owusu, E., Pärn, E., e Edwards, D. (2019). Review of application of analytic hierarchy process (ahp) in construction. *International Journal of Construction Management*, 19:436–452.
- de Castro, A. K. A., Pinheiro, P. R., e Pinheiro, M. C. D. (2009a). An Approach for the Neuropsychological Diagnosis of Alzheimer's Disease: A Hybrid Model in Decision Making. In P. Wen, L. P. Y. Y. S. T. G. W., Y. Li, editor, *Rough Sets and Knowledge Technology. RSKT 2009*, volume 5589 of *Lecture Notes in Computer Science*, p. 230–239. Springer, Berlin, Heidelberg.
- de Castro, A. K. A., Pinheiro, P. R., Pinheiro, M. C. D., e Tamanini, I. (2011). Towards the applied hybrid model in decision making: A neuropsychological diagnosis of alzheimer's disease study case. *International Journal of Computational Intelligence Systems*, 4(1):89–99.

- de Castro, A. K. A., Pinheiro, P. R., e Pinheiro, M. C. D. (2008). A Hybrid Model for Aiding in Decision Making for the Neuropsychological Diagnosis of Alzheimer's Disease. In C. C. Chan, W. P. Z., J. W. Grzymala-Busse, editor, *Rough Sets and Current Trends in Computing. RSCTC 2008*, volume 5306 of *Lecture Notes in Computer Science*, p. 512–521. Springer, Berlin, Heidelberg.
- de Castro, A. K. A., Pinheiro, P. R., e Pinheiro, M. C. D. (2009b). The Early Diagnosis of Alzheimer's Disease via a Multicriteria Classification Model. In Ehr Gott, M., Fonseca, C., Gandibleux, X., Hao, J., e Sevaux, M., editors, *Evolutionary Multi-Criterion Optimization. EMO 2009*, volume 5467 of *Lecture Notes in Computer Science*, p. 352–362. Springer, Berlin, Heidelberg.
- Dernoncourt, F. e Lee, J. Y. (2016). Optimizing neural network hyperparameters with gaussian processes for dialog act classification. *2016 IEEE Spoken Language Technology Workshop (SLT)*.
- Dumont, V. et al. (2022). Hyperparameter optimization of generative adversarial networks. *Journal of Machine Learning Research*.
- Fujika, S., Yajima, Y., Tanaka, T., Fujii, A., Kato, Y., Ohshima, S., e Katagiri, T. (2023). Parallelization of automatic tuning for hyperparameter optimization of pedestrian route prediction applications using machine learning. *Proceedings of the International Conference on High Performance Computing in Asia-Pacific Region*.
- Glorot, X. e Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, p. 249–256.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., e Bengio, Y. (2014). Generative adversarial networks. *Communications of the ACM*, 63(7): 139–144.
- He, K., Zhang, X., Ren, S., e Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision*, p. 1026–1034.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., e Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in Neural Information Processing Systems*, 30:6626–6637.
- Hutter, F., Kotthoff, L., e Vanschoren, J. (2019). Automated machine learning: Methods, systems, challenges. *Automated Machine Learning*, p. 219–249.
- Ishizaka, A., Tasiou, M., e Martínez, L. (2020). Analytic hierarchy process-fuzzy sorting: An analytic hierarchy process-based method for fuzzy classification in sorting problems. *Journal of the Operational Research Society*.
- Isola, P., Zhu, J.-Y., Zhou, T., e Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 1125–1134.
- Jin, H., Song, Q., e Hu, X. (2019). Auto-keras: An efficient neural architecture search system. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, p. 1946–1956.

- Karras, T., Laine, S., e Aila, T. (2019). A style-based generator architecture for generative adversarial networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(12): 2870–2882.
- Klein, A., Falkner, S., Bartels, S., Hennig, P., e Hutter, F. (2017). Fast bayesian hyperparameter optimization on large datasets. *Electronic Journal of Statistics*.
- Li, L., Jamieson, K. G., DeSalvo, G., Rostamizadeh, A., e Talwalkar, A. (2016). Hyperband: A novel bandit-based approach to hyperparameter optimization. *J. Mach. Learn. Res.*
- Lucic, M., Kurach, K., Michalski, M., Gelly, S., e Bousquet, O. (2018). Are gans created equal? a large-scale study. *Advances in Neural Information Processing Systems*, 31:700–709.
- M. Abadi, P. B.-E. B. Z. C. C. C. G. S. C. A. D. J. D. M. D. S. G. I. G. A. H. G. I. M. I. Y. J. R. J. L. K. M. K. J. L. D. M. R. M. S. M. D. M. C. O. M. S. J. S. B. S. I. S. K. T. P. T. V. V. V. V. F. V. O. V. P. W. M. W. M. W. Y. Y., A. Agarwal e Zheng, X. (2015). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. Available at <https://www.tensorflow.org/>.
- Murugan, S. (2017). Hyperparameters optimization of deep convolutional neural networks. *International Journal of Advanced Research in Computer Science*.
- Oliva, G., Setola, R., e Scala, A. (2017). Sparse and distributed analytic hierarchy process. *Automatica*, 85:211–220.
- Pereira, P. et al. (2023). Evaluation of smart sensors for subway electric motor. *Journal of Smart Sensor Systems*.
- Radford, A., Metz, L., e Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.
- Real, E., Aggarwal, A., Huang, Y., e Le, Q. V. (2020). Automl-zero: Evolving machine learning algorithms from scratch. *Proceedings of the 37th International Conference on Machine Learning*, p. 8007–8019.
- Saaty, T. L. (1980). *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw-Hill.
- Saaty, T. L. (2016). The analytic hierarchy process (ahp). *Springer Series in Operations Research and Financial Engineering*, p. 123–147.
- Safarik, J., Jalowiczor, J., Gresak, E., e Rozhon, J. (2018). Genetic algorithm for automatic tuning of neural network hyperparameters. *Proceedings of SPIE - The International Society for Optical Engineering*, 10643.
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., e Chen, X. (2016). Improved techniques for training gans. *arXiv preprint arXiv:1606.03498*.
- Salinas, E. et al. (2021). A comprehensive survey on gans and their applications. *Artificial Intelligence Review*, 54:3067–3114.
- Salomon, V. A. P. e Gomes, L. F. A. M. (2024). Consistency improvement in the analytic hierarchy process. *Mathematics*, 12(6):828.

- Santos, M. d., Costa, I., e Gomes, C. F. S. (2021). Multicriteria decision-making in the selection of warships: A new approach to the ahp method. *International Journal of the Analytic Hierarchy Process*, 13(1).
- Tamanini, I., de Castro, A. K., Pinheiro, P. R., e Pinheiro, M. C. D. (2009). Towards an applied multicriteria model to the diagnosis of Alzheimer's disease: A neuroimaging study case. In *2009 IEEE International Conference on Intelligent Computing and Intelligent Systems*, p. 652–656.
- Theis, L., van den Oord, A., e Bethge, M. (2016). A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*.
- Vafaei, N., Ribeiro, R., e Camarinha-Matos, L. (2016). Evaluating the performance of decision making units by using the integration of data envelopment analysis, balanced scorecard, and analytic hierarchy process. *International Journal of Information Technology & Decision Making*, 15(2):233–266.
- Waheed, A., Goyal, M., Gupta, D., Khanna, A., Al-Turjman, F., e Pinheiro, P. R. (2020). Covid-GAN: Data Augmentation Using Auxiliary Classifier GAN for Improved Covid-19 Detection. *IEEE Access*, 8:91916–91923.
- Wang, Z., She, Q., Ward, L., Wolverton, C., Ling, J., e Xie, T. (2018). Gans for scientific research. *Nature Machine Intelligence*, 1:24–33.
- Wang, Z., Bovik, A. C., Sheikh, H. R., e Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.
- Wistuba, M., Schilling, N., e Schmidt-Thieme, L. (2016). Two-stage transfer surrogate model for automatic hyperparameter optimization. *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*.
- Wu, J. et al. (2019). Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology*.
- Xiao, H., Rasul, K., e Vollgraf, R. (2017). Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.
- Yao, C., Cai, D., Bu, J., e Chen, G. (2017). Pre-training the deep generative models with adaptive hyperparameter optimization. *Neurocomputing*, 247:144–155.
- Zhai, P. e Rajan, R. (2023). Distributed gaussian process hyperparameter optimization for multi-agent systems. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.