Geração de imagens minerais sintéticas para modelos de aprendizado profundo usando GANs

Generating synthetic mineral images for deep learning models using GANs

Nancy Baygorrea
Bolsista PCI, Sistemas e computação, D.Sc.

Otávio da Fonseca Martins Gomes
Supervisor, Eng. de Materiais, D.Sc.

Resumo

Estratégias de aprendizado profundo têm sido cada vez mais utilizados em diversas aplicações, inclusive na mineralogia aplicada. Todavia, a questão de escassez de dados, particularmente a dificuldade de obtenção de dados anotados, é um dos principais desafios para a utilização mais ampla do aprendizado profundo em rotinas de caracterização mineral. Neste trabalho, propomos o emprego de redes adversárias generativas (GANs) para gerar imagens minerais sintéticas a fim de aumentar os dados disponíveis para treinamento de modelos de aprendizado profundo.

Palavras-chave: aprendizado profundo; redes adversárias generativas; data augmentation; caracterização mineral.

Abstract

Deep learning strategies have been increasingly used in various applications, including applied mineralogy. However, the issue of data scarcity, particularly the difficulty of obtaining annotated data, is one of the main challenges for the broader use of deep learning in mineral characterization routines. In this work, we propose the use of generative adversarial networks (GANs) to generate synthetic mineral images in order to increase the data available for training deep learning models.

Keywords: deep learning; generative adversarial networks; data augmentation; mineral characterization.

1. Introdução

Grandes quantidades de dados são fundamentais para o treinamento de modelos de aprendizado profundo de alto desempenho, pois permitem aos modelos extrair características mais generalizadas e, assim, obter maior precisão. No entanto, os problemas de dados insuficientes e desbalanceados (ou seja, um tipo de dado ou classe é significativamente maior que outro) são comuns em muitas aplicações. Uma solução promissora para isso é a utilização de técnicas de aumento de dados (*data augmentation*). Desse modo, é possível aumentar a quantidade de dados disponíveis e, ao mesmo tempo, balanceá-los, a fim de aprimorar o treinamento dos modelos (LIU et al., 2023).

O reconhecimento de partículas, grãos e fases minerais depende de diversas características, como textura, cor, forma, associações, entre outras, o que implica particularmente na demanda de grande quantidade de imagens para treinamento de modelos de aprendizado profundo. Os métodos clássicos de *data augmentation* utilizados para geração de imagens sintéticas baseiam-se em operações simples de processamento de imagens, como, por exemplo, rotação, inversão, *zoom*, translação, contraste, corte e adição de ruído. No entanto, nem sempre esses métodos são efetivos para a geração de imagens minerais sintéticas.

2. Objetivos

O presente trabalho propõe o emprego de redes adversárias generativas (GANs) para gerar imagens minerais sintéticas a fim de aumentar os dados disponíveis para treinamento de modelos de aprendizado profundo.

São objetivos específicos:

- Desenvolver uma variação do modelo GAN para a geração de imagens minerais sintéticas;
- Avaliar o desempenho do treinamento de um modelo de aprendizado profundo usando as imagens minerais sintéticas geradas pelo GAN como conjunto de treinamento.

3. Material e Métodos

3.1 Redes Adversárias Generativas (GANs)

Redes Adversárias Generativas (GANs) são uma arquitetura baseada em redes neurais de aprendizado profundo que foi proposta por GOODFELLOW et al. (2014). O resultado dessa arquitetura é uma rede neural profunda geradora que, a partir de um vetor ruído aleatório, é capaz de gerar dados sintéticos únicos. De fato, o autor propôs um "jogo competitivo" entre duas redes neurais: o gerador (G) e o discriminador (D). A função do gerador é criar novos dados a partir de um ruído aleatório como entrada, enquanto o discriminador comprova se os dados aleatórios são consistentes com as instâncias de dados reais, devolvendo uma probabilidade entre 0 e 1, onde o valor 1 indica uma predição da realidade (*real*) e o valor 0 indica uma falsificação (*fake*). Como toda rede neural, precisa-se de alguma função perda para medir a fase do treinamento. Nesse caso, duas funções para cada uma dessas redes, gerador e discriminador. A função perda do gerador (GLoss) e a função perda do discriminador (DLoss) são definidos como:

$$GLoss(z) = -log\,(D(G(z)))$$
$$DLoss(x, z) = -log(D(x)) - log\,(1 - D(G(z)))$$

onde x é algum dado real e z é um vetor ruido. A interpretação dessas funções é simples: quando um gerador G tenta enganar o discriminador D, classificando o dado *fake* como um dado real, o objetivo é $D(G(z))$ = 1, o qual equivale a minimizar .$GLoss(z)$. A perda para o discriminador consiste em duas partes desde que ele tenha que analisar em dois dados diferentes. Minimizar a primeira parte consiste em obter $D(x)$ = 1 o qual corresponde a classificar o dado x como real, enquanto minimizar a segunda parte é equivalente a obter 1 − $D(G(z))$ = 1 respectivamente $D(G(z))$ = 0, o qual corresponde a classificar o dado *fake* $G(z)$ como *fake*. Então, as funções perda podem ser escritos com a função *Binary-Cross_Entropy*:

$$BCE(y, d) = -dlog(y) - (1 - d)log(1 - y),$$

onde y é a saída da rede e d o rótulo *target.* Para $GLoss$ tem-se dado *fake*, mas o objetivo do gerador é obter o dado f*ake* classificado como real porque o *traget label* é 1. Tudo isso conduz à seguinte fórmula:

$$GLoss(z) = -log\,(D(G(z))) = -log\,(D(G(z))) - 0 = BCE(D(G(z)), 1)$$

A função perda do discriminador trabalha analogamente, mas tendo em conta que tem-se duas partes: em princípio, tem-se um dado real que o discriminador quer classificar como real e um outro dado *fake* que o discriminador quer classificar como *fake*. Isso conduz à seguinte equação:

$$DLoss(x, z) = -log\,((D(x)) - log(1 - D(G(z)) = BCE(D(x), 1) + BCE(D(G(z)), 0).\,))$$

O segredo de toda essa dinâmica está no espaço latente (um espaço de vetores aleatórios de distribuição uniforme), de fato, na prática esse espaço pode não ser usado totalmente, especialmente quando se usa um espaço maior que 100 dimensões. Geralmente tem uma variedade (espaço topológico) estruturada no espaço latente onde muito das imagens de qualidade podem ser aplicadas. Fora dessa variedade, encontra-se ruídos, onde imagens inconvenientes podem não ser convenientemente particular. De fato, a ideia do espaço latente é o mais próximo à ideia de variáveis latentes em estatísticas. Outro importante ponto é que não importa que tão grande o espaço latente seja. Espaços latentes muito pequenos limitarão a performance do modelo para capturar a distribuição total de dados reais. Nesse trabalho, vamos mostrar uma aplicação de uma variação de modelo generativo proposto por RADFORD et al. (2016) para resolver um problema de segmentação binária usando aprendizado profundo.

3.2 Dataset

O conjunto de dados (*dataset*) considerado neste experimento é composto por imagens de um minério de ferro itabirítico, composto de hematita e quartzo com pequenas quantidades de magnetita e goethita. O *dataset* e sua descrição podem ser encontrados em repositório público através do link: https://zenodo.org/records/5014700.

O *dataset* é composto de 81 pares de imagens correlacionadas. Cada par contém uma imagem adquirida por microscopia de luz refletida (RGB, 24 bits) e a imagem de anotação em 8 bits, obtida através de microscopia

eletrônica de varredura (MEV), contendo dois valores de pixel que dependem à fase que correspondem: pixel de valor 0 corresponde ao minério e pixel de valor 255, à resina. As imagens máscara (ou anotadas) foram aplicadas nas imagens RGB a fim de obter imagens com as fases minerais mantidas, mas com *background* 0. Isso permite a obtenção das anotações das imagens sintéticas. Figura 1 mostra as imagens alimentadas no modelo GAN.

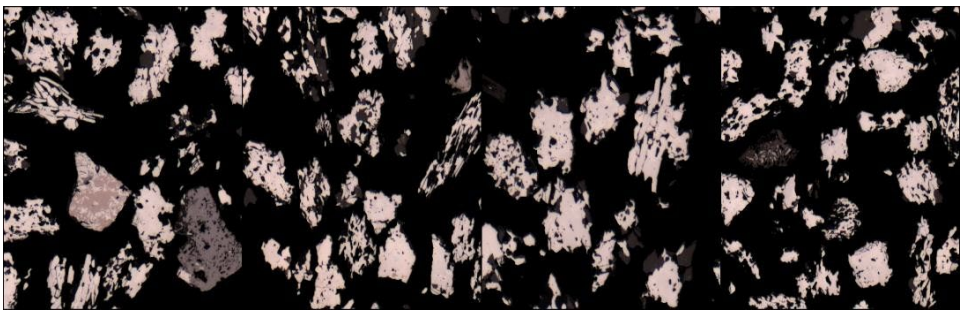

Figura 1. Amostras do conjunto original após converter o valor dos pixels correspondentes à resina para o valor 0 correspondente ao *background.*

3.3 Arquitetura do Modelo

Como descrito na seção 3.1, a dinâmica da GAN clássica proposta por GOODFELLOW et al. (2014) demostrou o grande potencial das GANs no problema da geração sintética de imagens. Não obstante, ainda apresentava algumas dificuldades como o "*mode colapso"* ou quando o gradiente da função perda se aproximava muito para zero. Para nosso experimento, uma variação do modelo GAN, chamado de *"Deep Convolutional GAN"* (DCGAN), que foi proposto por RADFORD et al. (2016), foi usado. As diferenças do GAN clássico e do DCGAN são: camadas *pooling* das GANs são substituídas por convolucionais para o gerador, o qual ajuda o modelo aprender sua própria representação espacial ao invés de ter camadas *pooling* determinísticas não treináveis; a substituição de camadas completamente conectadas em favor de arquiteturas profundas; o uso de *"Batch normalization"* para garantir o fluxo do gradiente em redes profundas; o sucesso de funções de ativação limitada de ReLU e *Leaky* ReLU com o fim de ajudar a aprender sobre a distribuição de treinamento mais rápido. Assim, desde que o conjunto de imagens é composto por imagens de boa resolução, foram utilizadas 7 camadas convolucionais (acompanhadas de "*BatchNormalization"* e funções de ativação "*LeakyReLu"*) na construção do gerador quanto no discriminador. Otimizador tanto do modelo gerador quanto discriminador: Adam com taxa de aprendizado de 0.0002. Épocas (ou iterações para o treinamento de todo o conjunto de dados): 9000. Função perda: "*Binary Cross Entropy"*. O processo demorou aproximadamente 10 h em um computador com 12 GB de GPU.

4. Resultados e Discussão

O treinamento do modelo DCGAN sobre o conjunto de dados foi sobre 9000 épocas, de maneira proposital para mostrar que pode ter muitos valores ótimos para o gerador (G) quanto para o discriminador (D) ao longo do treinamento. A Figura 2 apresenta imagens geradas nas épocas 100, 200, 2000 e 8000. Na Figura 3, é mostrado as funções dos modelos G e D durante o treinamento, onde podemos observar que todas as perdas são um

tanto erráticas no início da execução antes de se estabilizarem por volta da época 100 até a época 300. As perdas permanecem “estáveis” após esse ponto, embora a variância aumente ao longo do treinamento.

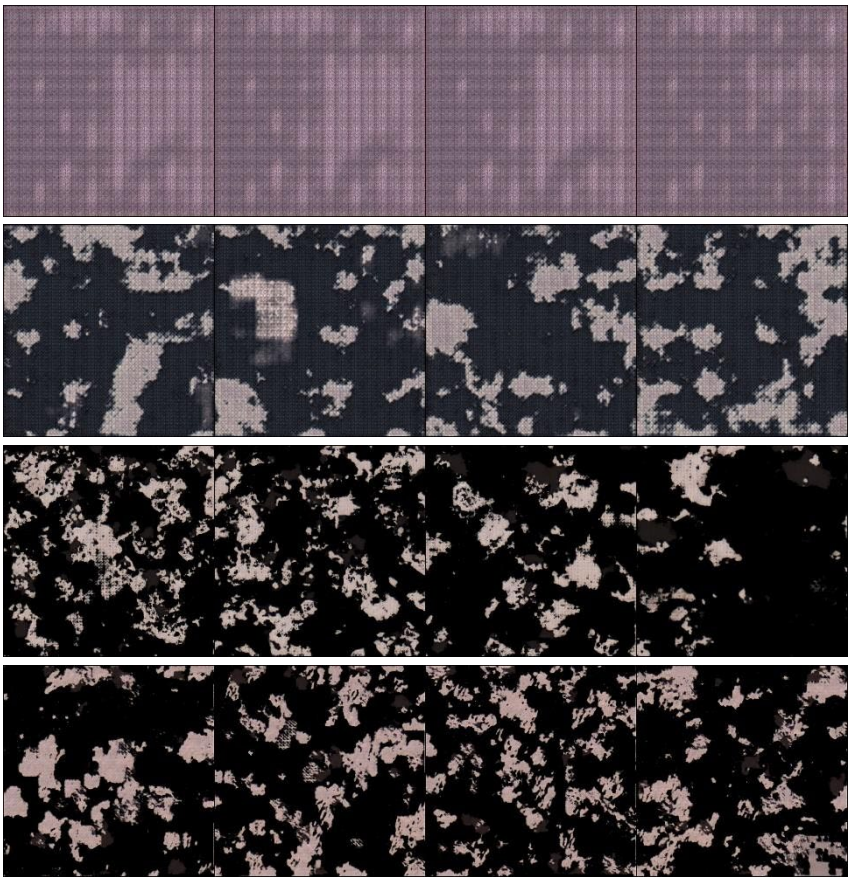

Figura 2. De acima para abaixo, cada fila de imagens corresponde a imagens geradas pelo gerador nas épocas 100, 200, 2000 e 8000, respectivamente.

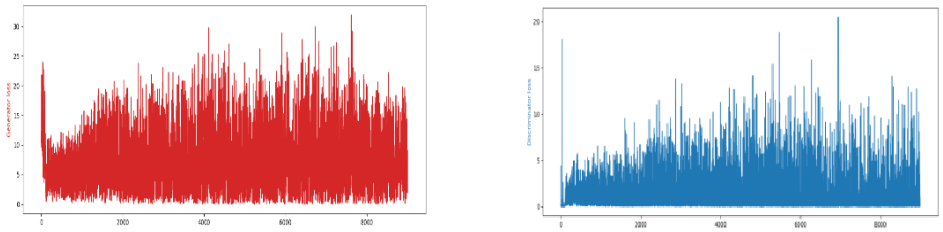

Figura 3. Curvas da função de perda do gerado quanto do discriminador no modelo DCGAN.

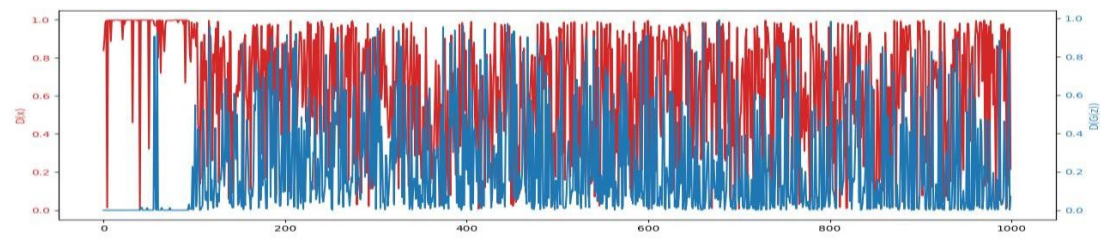

Figura 4. *Scores* do discriminador sobre imagens reais (DX) e sobre imagens *fake* (D(G(z))).

Desde que a maior parte do treinamento se apresenta estável, vamos analisar os *"score real"* e *"score fake"* em uma faixa do eixo x (correspondente às épocas) que estão representados na Figura 4. Este é um exemplo da perda normal ou esperada durante o treinamento. Ou seja, a perda do discriminador para amostras reais e falsas é aproximadamente em torno de (0,3 e 0,6). Da teoria do modelo, o discriminador tenta distinguir entre amostras reais e sintéticas, atribuindo pontuações que refletem a confiança na autenticidade de cada amostra, onde D(x) representa a pontuação do discriminador sobre imagens reais e D(G(z)) corresponde a pontuação do discriminador em relação a imagens geradas (ou sintéticas). Quanto mais próximo de 1 para amostras reais e 0 para amostras geradas, melhor o discriminador está desempenhando sua função no treinamento do modelo GAN. Uma vez tendo os "*checkpoints"* (dicionário onde são armazenadas informações tais como os pesos, otimizadores e os modelos para serem utilizados no processo de inferência) salvos no treinamento de algumas épocas, vai ser escolhido alguma época de qualquer um que contenha o *score real* e *score fake* próximos de 0,5. Para a inferência, carregamos os pesos salvos naquele "*checkpoint*" e o modelo gerador para gerar as imagens que serão utilizadas em um problema de segmentação de imagens binárias com o fim de testar a melhora no desempenho de algum modelo de segmentação sobre os dados sintéticos. Logo após esse procedimento, na construção das anotações das imagens sintéticas, precisa-se substituir o *background* com valores de pixel aleatórios que corresponde à distribuição dos valores de pixel da resina.

Para finalizar com o experimento, um modelo de segmentação será considerado para testar o desempenho do treinamento para esses dois conjuntos de dados: os originais dados iniciais com 81 imagens com suas anotações correspondentes e o conjunto de imagens sintéticas geradas pelo modelo DCGAN com 3 imagens e suas respectivas anotações. Assim, as Figura 5 e 6 mostram o desenvolvimento do treinamento de um modelo de segmentação (UNet clássico) sobre esses dois conjuntos de dados. Veja que a função perda no primeiro está indo para um *overfitting*, enquanto no segundo percebe-se que, pelo fato da acurácia flutuante crescente, o modelo tem um melhor aprendizado, mas ainda para um melhor aprendizado tem que se ajustar os parâmetros. De todo modo, cabe ressaltar que esses resultados preliminares são consistentes e por vezes até superiores aos obtidos por FILIPPO et al. (2021) no treinamento de modelos a partir do mesmo *dataset*.

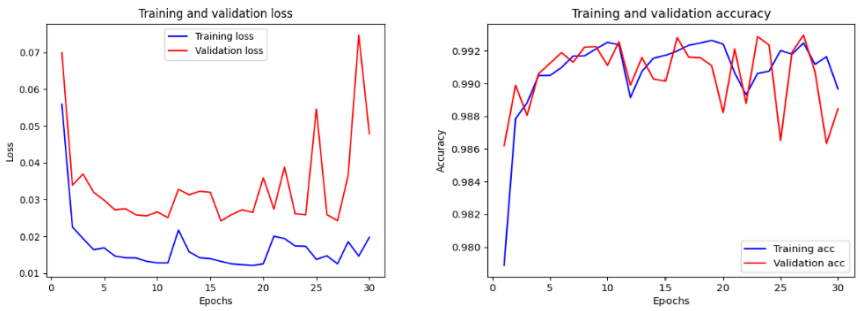

Figura 5. Curvas da função perda e da acurácia do modelo UNet para o conjunto de imagens original.

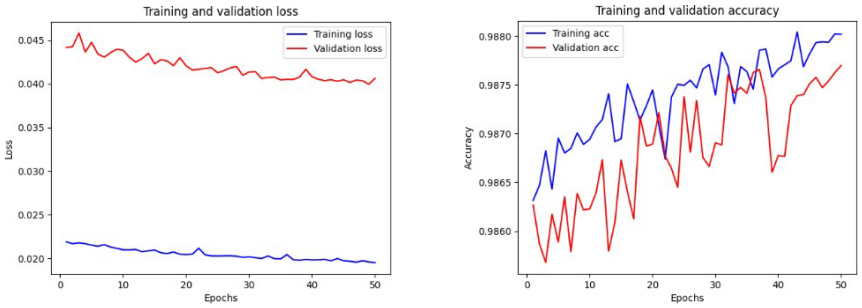

Figura 6. Curvas da função perda e da acurácia do modelo UNet para o conjunto de imagens sintéticas.

5. Conclusão

A geração de imagens sintéticas indica uma boa estratégia útil no treinamento de modelos de aprendizado profundo. No nosso experimento, as curvas da função perda e a acurácia do modelo de segmentação considerando um conjunto totalmente sintético teve melhor performance em comparação com os conjuntos de imagens iniciais. É importante notar que a eficácia da geração de imagens sintéticas depende, além da qualidade e da representatividade das imagens geradas em relação ao problema específico, da construção da arquitetura do modelo generativo adversário.

Sobre o modelo, está tudo bem se a função perda, tanto do gerador quanto do discriminador, oscilar – é apenas evidência do modelo tentando se aprimorar a cada iteração. A estabilidade do treinamento pode se deteriorar em períodos de alta variância na perda e imagens geradas de qualidade inferior. Para modelos estáveis, modificações para produzir alguns casos específicos de falha: colapso de modo e falha de convergência são tópicos de pesquisa futura.

Os resultados obtidos nesse experimento são importantes, pois destacam que a qualidade gerada pode e de fato varia ao longo da execução, mesmo após o processo de treinamento se tornar estável. Mais iterações de treinamento, além de algum ponto de estabilidade de treinamento, podem ou não resultar em imagens de maior qualidade. Por fim, ao aplicar GANs na geração de imagens de minérios, é muito importante validar os resultados gerados para garantir que eles se alinhem com as características reais dos minérios a serem estudados. Além disso, a colaboração entre especialistas em geologia, aprendizado de máquina e processamento de imagens pode ser benéfica para garantir a precisão e relevância de imagens geradas.

6. Agradecimentos

Ao professor Otávio Gomes, pelo apoio, disponibilidade, aconselhamento assertivo e pelo estímulo permanente que contribuem para aumentar o desafio e melhorar a profundidade da pesquisa, pela sua amizade. Ao setor SCT pela parceria. Ao MCTI pela bolsa PCI concedida.

7. Referências Bibliográficas

FILIPPO, M.P.; GOMES, O.D.M.; DA COSTA, G.A.O.P.; MOTA, G.L.A. Deep learning semantic segmentation of opaque and non-opaque minerals from epoxy resin in reflected light microscopy images. Minerals Engineering, v. 170, 107007, 2021.

GOODFELLOW, I.J. et al; **Tutorial: Generative Adversarial Nets**, NIPS, 2014.

LIU, Y.; WANG, X.; ZHANG, Z., DENG, F. Deep learning based data augmentation for large-scale mineral image recognition and classification. **Minerals Engineering**, v. 204, 108411, 2023.

RADFORD, A.; METZ, L.; CHINTALA, S. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv:1511.06434v2, 2016.