

# Redes Neurais Adversárias Generativas

## 1. Introdução a modelos generativos

Até este momento, as redes neurais artificiais que já estudamos, incluindo as redes profundas, atuam essencialmente como modelos discriminativos. Tomando como exemplo o problema de classificação, a rede parte de um padrão completo em sua entrada e realiza um mapeamento para uma determinada classe ao reconhecer características típicas daquela classe.

Os modelos generativos, em contrapartida, operam no sentido oposto: a partir de um descritor de uma classe, eles devem gerar padrões completos que apresentem as características típicas daquela classe. A Figura 1 apresenta de forma pictórica esta diferença.

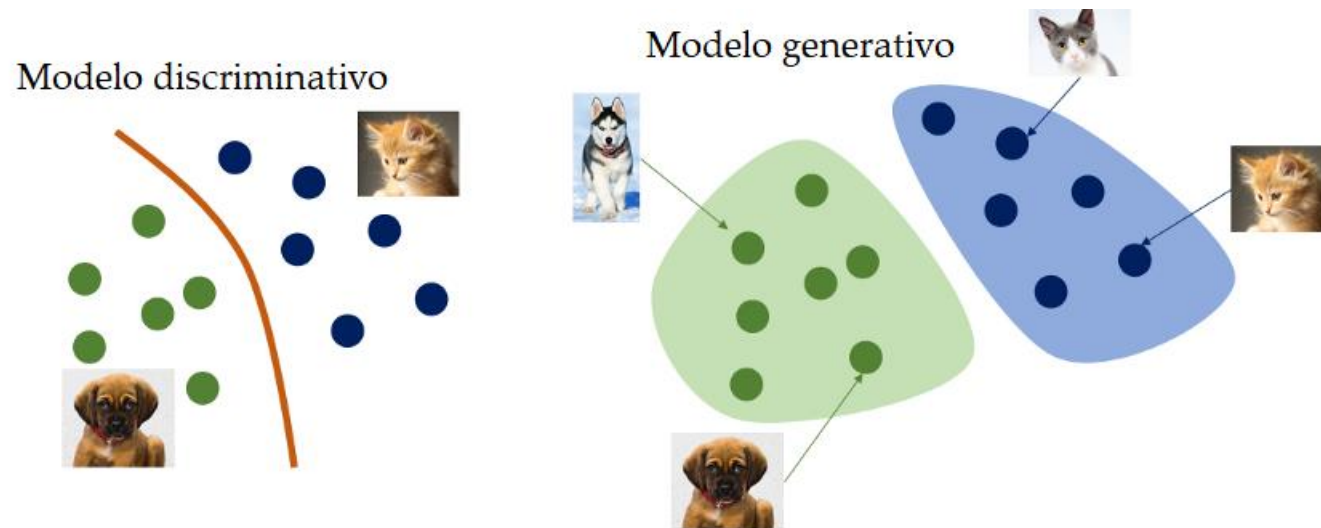


Figura 1 – Enquanto o modelo discriminativo aprende a identificar a classe de cada padrão de entrada, o modelo generativo deve aprender a modelar a distribuição dos dados para, então, poder sintetizar novos padrões consistentes com aqueles vistos durante o treinamento. Extraída de <https://medium.com/@jordi299/about-generative-and-discriminative-models-d8958b67ad32>.

A partir de seu aprendizado, um modelo generativo deve ser capaz de gerar um conjunto ilimitado de padrões sintéticos de algum domínio de interesse. O desafio, porém, é fazer com que (1) os padrões artificialmente criados não sejam facilmente distinguíveis de padrões reais, e (2) que haja algum grau de controle sobre o tipo de padrão gerado.

O modelo generativo, portanto, não só deve capturar a essência dos padrões, mas também os seus detalhes e variações de estilo e forma. Além disso, ele deve ser capaz de produzir o tipo de resultado selecionado na entrada.

Em última análise, ele deve aprender a distribuição que está por trás da classe de padrões que se quer gerar, o que envolve descobrir e mapear corretamente o *manifold*, *i.e.*, o espaço de dimensão reduzida em que se manifestam as variações presentes nos padrões de entrada.

É justamente neste contexto que surgiram as redes neurais adversárias generativas (GANs, do inglês *generative adversarial networks*) (GOODFELLOW ET AL., 2014; GUI ET AL., 2020). Nas palavras de Yann LeCun, trata-se “*da ideia mais interessante nos últimos 10 anos em aprendizado de máquina*”.

## 2. Redes Adversárias Generativas

As GANs trazem a ideia de se estabelecer um jogo entre duas redes neurais, denominadas de *geradora* e *discriminadora*.

A *rede geradora* tem como função criar dados sintéticos que sejam fidedignos em relação aos dados verdadeiros, *i.e.*, cujas características sejam plausíveis a ponto de estas amostras se confundirem com os dados verdadeiros.

Por outro lado, a *rede discriminadora* tenta justamente descobrir quando o dado recebido em sua entrada é real ou fictício (*i.e.*, se foi artificialmente criado pela rede geradora). Logo, a rede discriminadora corresponde a um classificador binário que visa detectar o que é real e o que é sintético.

Estas duas redes são chamadas de *adversárias* porque o aumento de desempenho de uma rede, mantendo a outra inalterada, acarreta uma perda de desempenho da outra. Portanto, a rede discriminadora busca se tornar cada vez mais hábil em

discriminar imagens sintéticas de imagens reais. Concomitantemente, a rede generativa tenta criar imagens sintéticas que sejam cada vez mais verossímeis, ou, em outras palavras, menos distinguíveis de imagens reais.

As Figuras 2 e 3 apresentam a estrutura das GANs em um contexto genérico e no caso de geração de imagens de dígitos manuscritos, respectivamente.

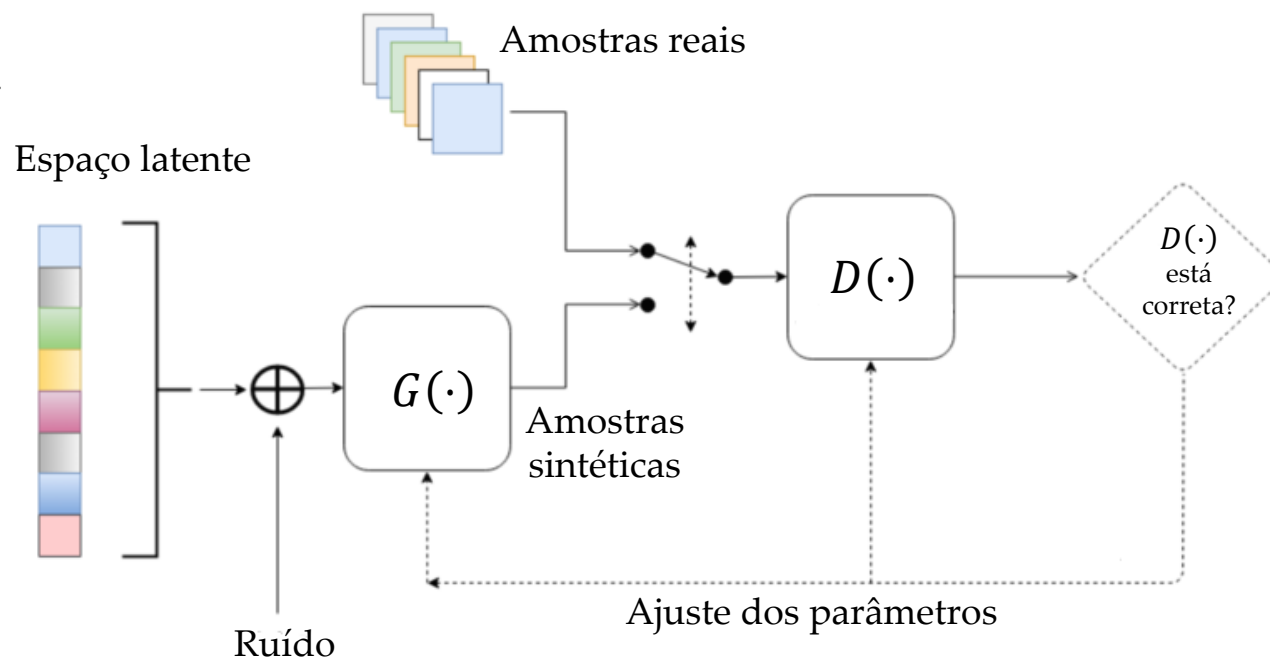


Figura 2 – Estrutura de uma GAN. Adaptada de <https://medium.com/machinelearningadvantage/create-any-image-with-c-and-a-generative-adversarial-network-6031a4b90dec>.

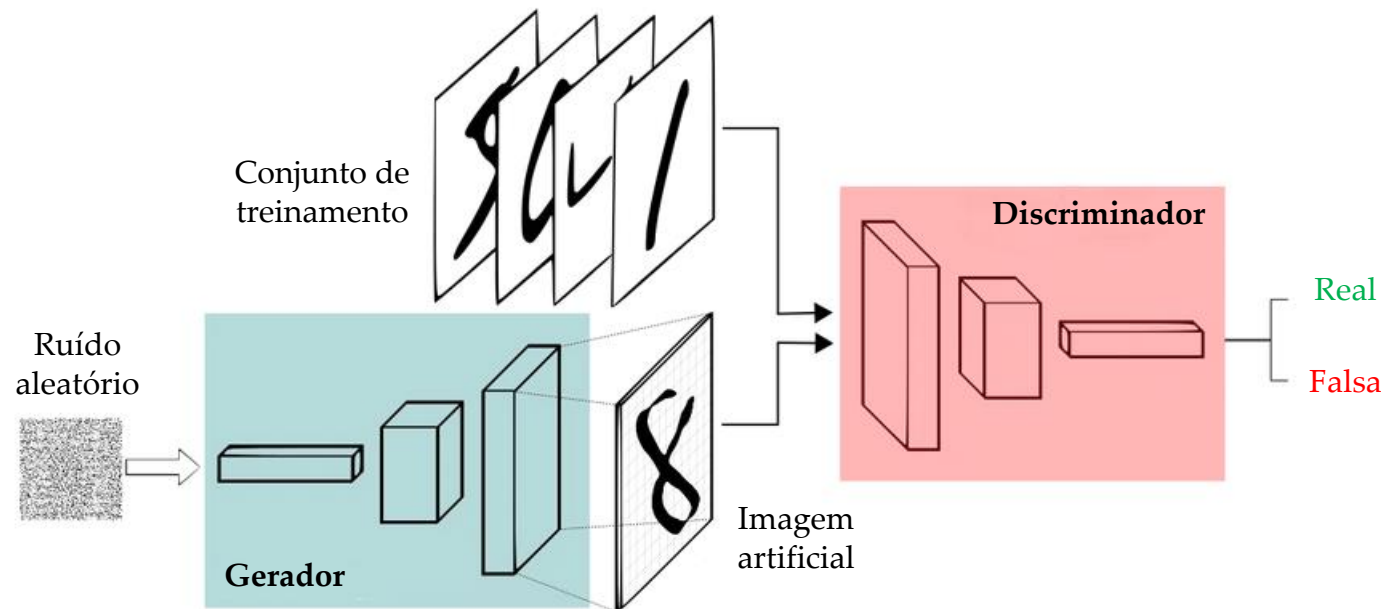


Figura 3 – GAN aplicada à síntese de imagens de dígitos manuscritos. Adaptada de <http://deeplearningbook.com.br/introducao-as-redes-adversarias-generativas-gans-generative-adversarial-networks/>.

Como podemos observar nas Figuras 2 e 3, a rede geradora recebe como entrada uma amostra aleatória de uma determinada distribuição (*e.g.*, Gaussiana), com uma determinada dimensionalidade, e gera um dado novo (*e.g.*, uma imagem). Podemos

interpretar esta entrada como sendo a representação em um espaço latente (ou o código) correspondente à imagem que será gerada.

Note também que a rede geradora nunca recebe como entrada dados retirados do conjunto de treinamento. Na verdade, ele deve aprender a gerar dados válidos apenas por meio de sua interação com a rede discriminadora (GÉRON, 2019). Com isso, quanto mais a rede discriminadora consegue perceber que os dados criados pela geradora são falsos, mais a rede geradora é levada pelo treinamento a aprimorar seus parâmetros de modo a conseguir sintetizar amostras que se confundam com os dados reais.

Idealmente, o equilíbrio deste jogo se estabelece quando a rede geradora se torna tão hábil em criar dados plausíveis que a rede discriminadora, embora muito competente em reconhecer dados falsos, já não consegue mais perceber a diferença, convergindo para uma taxa de acerto de 50% (equivalente a um classificador

aleatório) (GOODFELLOW ET AL., 2016). Logo, é possível afirmar que a rede generativa aprendeu, de fato, a distribuição subjacente à classe de padrões que se deseja gerar.

## 2.1. Estratégia de treinamento

Em essência, treinar uma GAN envolve ajustar conjuntamente os parâmetros de duas redes – a geradora e a discriminadora –, considerando o objetivo que cada uma tenta atingir. Por um lado, este processo traz à baila os mesmos conceitos e técnicas utilizados para o treinamento de redes profundas.

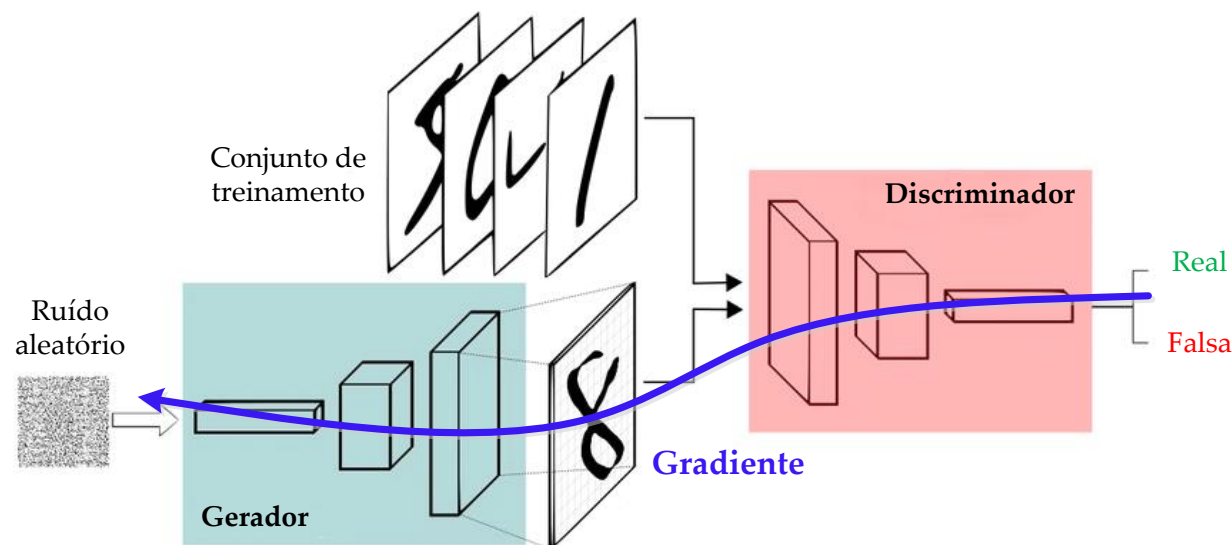


Figura 4 – Retropropagação do erro para cálculo do vetor gradiente em GANs. (Adaptação da Figura 3).



Por outro lado, uma nova função custo é explorada, a qual modela explicitamente os papéis a serem desempenhados pelas duas redes.

Durante o treinamento, é preciso “sintonizar” o aprendizado das duas redes no sentido de manter um equilíbrio entre os progressos destas redes em suas respectivas tarefas, na forma de um jogo de soma nula, com base em uma estratégia *minimax*. Caso uma rede avance muito rapidamente em relação à outra, isso pode prejudicar o aprendizado como um todo, pois a rede geradora precisa ser constantemente desafiada por um bom discriminador para progressivamente aprimorar as características dos padrões gerados.

### **Função objetivo:**

$$\min_G \max_D \mathbb{E}_{\mathbf{x} \sim p_{\text{dados}}} \{\log D(\mathbf{x})\} + \mathbb{E}_{\mathbf{z} \sim p_z} \{\log(1 - D(G(\mathbf{z})))\} \quad (1)$$

## Considerações:

- A rede discriminadora é um modelo de classificação binária que produz em sua saída a probabilidade de a entrada ser um padrão real (classe positiva) ou sintético (classe negativa).

Sendo assim, quando a entrada da rede discriminadora é um padrão do conjunto de treinamento, a saída esperada é  $D(\mathbf{x}) = 1$ . Quando, porém, a entrada é um padrão sintetizado pela rede geradora, denotado como  $G(\mathbf{z})$ , a saída deve ser zero ( $D(G(\mathbf{z})) = 0$ ).

Portanto, maximizar a função objetivo em (1) com respeito aos parâmetros da rede discriminadora corresponde a melhorar sua capacidade de distinguir os padrões reais dos sintéticos.

Obs.: Note que a expressão em (1) é o negativo da entropia cruzada binária.

- Por outro lado, o objetivo da rede geradora é justamente enganar a rede discriminadora, fazendo com que padrões sintéticos sejam considerados como padrões reais. Ou seja, o alvo da rede geradora é fazer com que suas saídas,  $G(\mathbf{z})$ , sejam classificadas como padrões reais, *i.e.*,  $D(G(\mathbf{z})) = 1$ .

Observe que somente o segundo termo da função objetivo em (1) depende da rede geradora. Assim, minimizar a função em (1) equivale a minimizar

$$\mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}} \{\log(1 - D(G(\mathbf{z})))\}, \quad (2)$$

de modo que os parâmetros de  $G(\cdot)$  são ajustados para tornar  $D(G(\mathbf{z}))$  cada vez mais próximo de 1.

Cada iteração do processo de treinamento da GAN é dividida em duas etapas:

- Na primeira etapa, a rede discriminadora é ajustada; um *batch* de amostras reais e sintéticas é oferecido para a rede, com rótulos iguais a 1 e 0, respectivamente, e

$D(\cdot)$  é ajustada tendo em vista o gradiente da entropia cruzada. Somente os pesos do discriminador são alterados nesta etapa.

- Na segunda etapa, a rede geradora é ajustada; primeiro, usamos  $G(\cdot)$  para criar um *batch* de amostras sintéticas e, uma vez mais, empregamos o discriminador para dizer se os padrões são reais ou falsos. Desta vez, não há amostras do conjunto de treinamento e todos os rótulos são iguais a 1. Os parâmetros da rede discriminadora permanecem fixos durante esta etapa, de modo que o algoritmo de gradiente afetará apenas os pesos da rede geradora.

É interessante perceber que a rede geradora nunca realmente tem contato com padrões reais e, mesmo assim, consegue produzir amostras sintéticas convincentes. Isso ocorre graças ao gradiente que flui através da rede discriminadora até a rede geradora. Além disso, quanto melhor for  $D(\cdot)$ , mais informações sobre padrões reais estarão contidas no vetor gradiente, de modo que  $G(\cdot)$  pode ser aprimorada.

A Figura 5 exibe o pseudocódigo do processo de treinamento da GAN.

---

**Algorithm 1** Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator,  $k$ , is a hyperparameter. We used  $k = 1$ , the least expensive option, in our experiments.

---

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

**for**  $k$  steps **do**

- Sample minibatch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from noise prior  $p_g(z)$ .
- Sample minibatch of  $m$  examples  $\{x^{(1)}, \dots, x^{(m)}\}$  from data generating distribution  $p_{\text{data}}(x)$ .
- Update the discriminator by ascending its stochastic gradient:

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

**end for**

- Sample minibatch of  $m$  noise samples  $\{z^{(1)}, \dots, z^{(m)}\}$  from noise prior  $p_g(z)$ .
- Update the generator by descending its stochastic gradient:


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

**end for**

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

---

Figura 5 – Resumo do treinamento de GANs. Extraída de (GOODFELLOW ET AL., 2014).



A ideia de confrontar duas redes neurais apresenta notória elegância e resultados promissores foram reportados na literatura. Não obstante, o treinamento de GANs costuma enfrentar alguns obstáculos, visto que existe a possibilidade de as duas redes oscilarem durante o processo de aprendizagem e não conseguirem desempenhar seus papéis. Além disso, também foi observado que à medida que o treinamento avança, as redes podem acabar “esquecendo” o que foi aprendido no início (GÉRON, 2019).

A partir do artigo pioneiro de Goodfellow et al. (2014), houve uma explosão de trabalhos relacionados a GANs (veja a Figura 6) e muitos esforços na comunidade para aprimorar os resultados. Por exemplo, foram propostas novas funções custo, além de estruturas mais sofisticadas para a rede geradora, como, por exemplo, as StyleGANs (KARRAS ET AL., 2019; KARRAS ET AL., 2020).

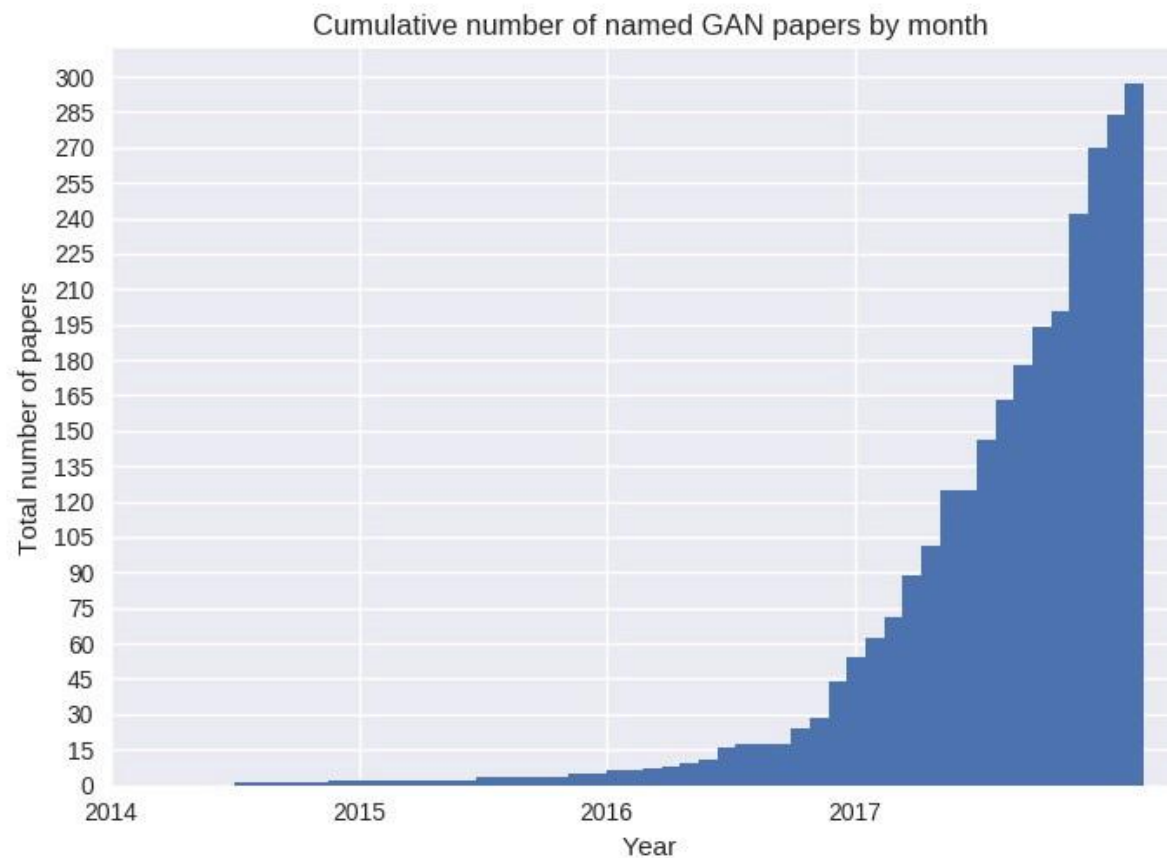


Figura 6 – Número acumulado de trabalhos relacionados a GANs desde 2014. Extraída de <https://github.com/nowozin/mlss2018-madrid-gan>.



## 3. Exemplos

### 3.1. Geração de imagens de faces



Figura 7 – Evolução na qualidade de imagens de faces humanas sintetizadas por GANs. Extraída de <https://github.com/nowozin/mlss2018-madrid-gan>.

**Sugestão:** consultar a página <https://thispersondoesnotexist.com/>.



### 3.2. Aritmética no espaço latente e conceitos visuais

A Figura 8 mostra um resultado fascinante obtido com as *deep convolutional* GANs (DCGANs) (RADFORD ET AL., 2016).

- As nove imagens no topo da Figura 8 foram manualmente selecionadas: três homens com óculos, três homens sem óculos e três mulheres sem óculos. Cada imagem é gerada pela DCGAN a partir de um vetor código de dimensão 100 (esta, portanto, é a dimensão do espaço latente).
- Para cada um destes grupos, foi calculado o vetor código médio e a respectiva imagem foi gerada (veja a última linha da Figura 8).
- Surpreendentemente, ao fazer a operação

Homem com óculos – Homem sem óculos + Mulher sem óculos

**no espaço latente**, foi obtido um vetor código que deu origem a uma imagem de uma mulher com óculos (centro do grid  $3 \times 3$ ). As oito imagens ao seu redor

foram geradas a partir do mesmo vetor código, mas corrompido por ruído, para mostrar a interpolação semântica feita pela DCGAN.

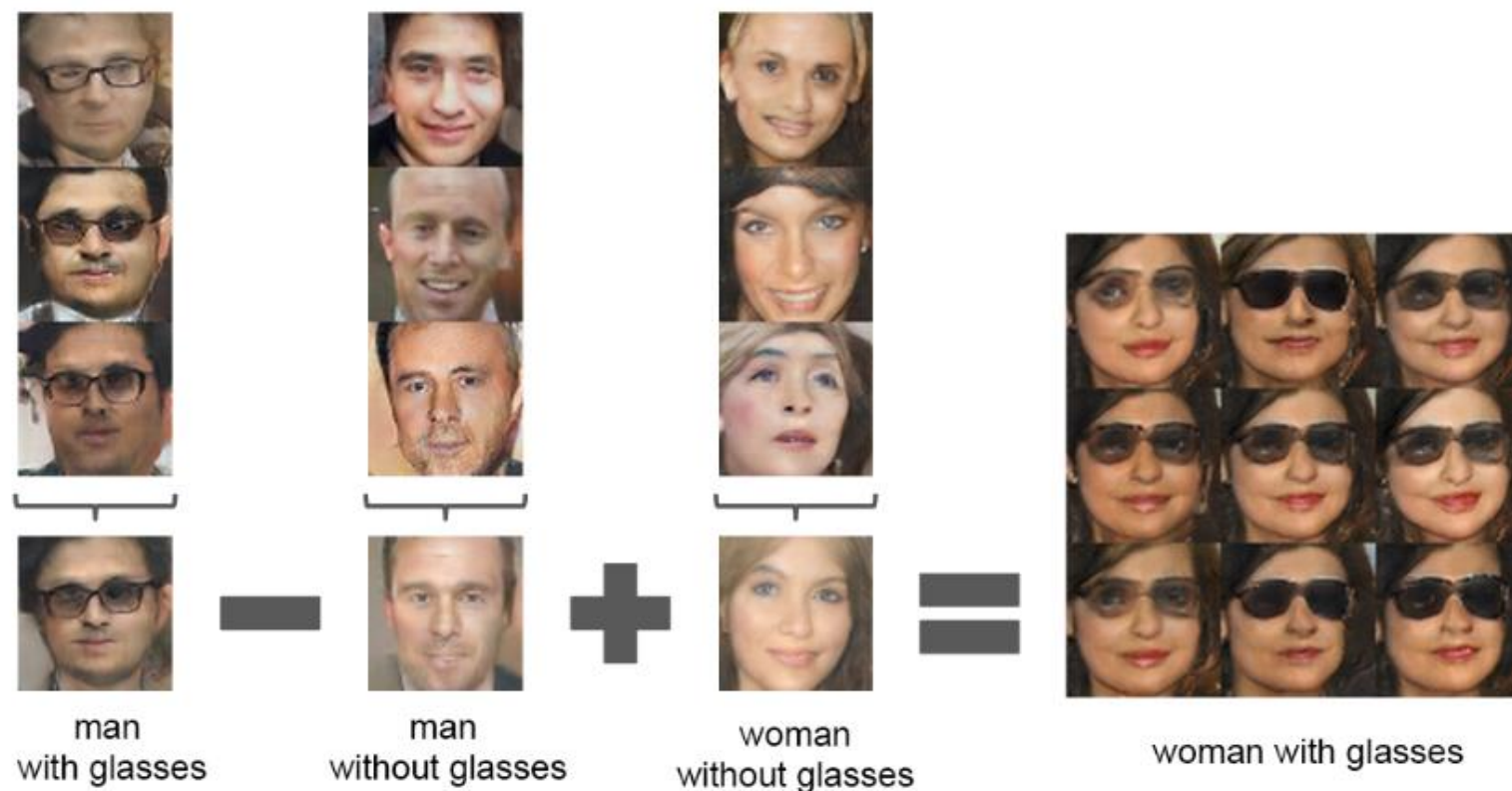


Figura 8 – Aritmética no espaço latente e seu efeito nas imagens geradas. Extraída de (Radford et al., 2016).

## 4. Referências bibliográficas

- GÉRON, A., **Hands-on Machine Learning with Scikit-Learn, Keras & Tensorflow**, O'Reilly Media, 2<sup>a</sup> ed., 2019.
- GOODFELLOW, I., BENGIO, Y., COURVILLE, A., **Deep Learning**, MIT Press, 2016.
- GOODFELLOW, I. J., POUGET-ABADIE, J., MIRZA, M., XU, B., WARDE-FARLEY, D., OZAIR, S., COURVILLE, A., BENGIO, Y., “Generative Adversarial Networks”, Proceedings of the 27<sup>th</sup> International Conference on Neural Information Processing Systems (NIPS), pp. 2672-2680, 2014.
- GUI, J., SUN, Z., WEN, Y., TAO, D., YE, J., “A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications”, *arXiv:2001.06937v1*, 2020.
- KARRAS, T., LAINE, S., AILA, T., “A Style-Based Generator Architecture for Generative Adversarial Networks”, *arXiv:1812.04948v3*, 2019.
- KARRAS, T., LAINE, S., AITTALA, M., HELLSTEN, J., LEHTINEN, J., AILA, T., “Analyzing and Improving the Image Quality of StyleGAN”, *arXiv:1912.04958v2*, 2020.
- RADFORD, A., METZ, L., CHINTALA, S., “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, *arXiv:1511.06434v2*, 2016.
- VON ZUBEN, F. J., **Notas de Aulas do Curso “Redes Neurais” (IA353)**, disponíveis em <http://www.dca.fee.unicamp.br/~vonzuben/courses/ia353.html>