Rede Adversária Generativa
(Generative Adversarial Network - GAN)

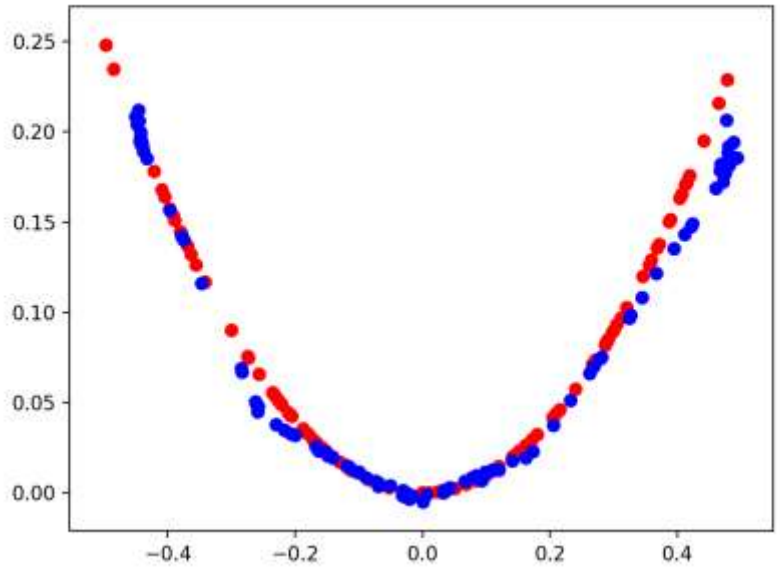
Aluizio Fausto Ribeiro Araújo
Universidade Federal de Pernambuco
Centro de Informática

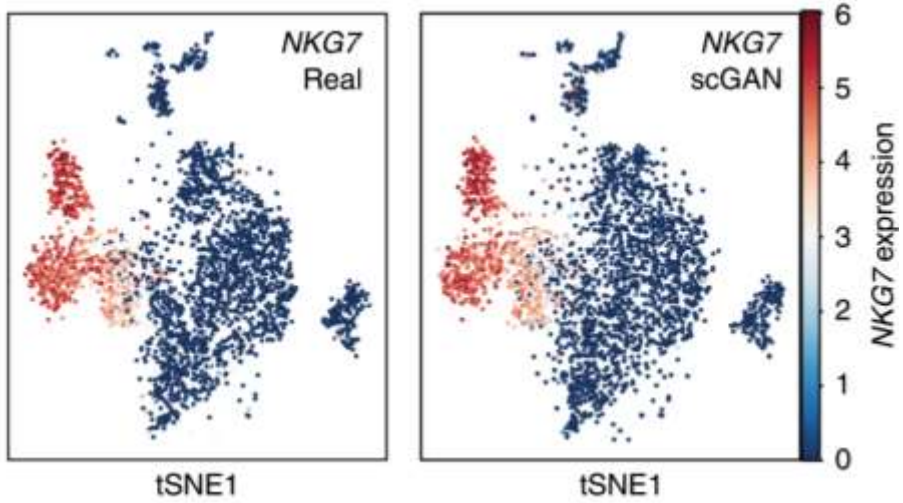
Conteúdo

- Introdução
  - Motivação, apresentação, objetivos, aplicações
- Modelos Generativos
- Redes Adversárias Generativas
  - Arquitetura de GAN
  - Treinamento de GAN
- Extensões
- Aplicações
- Softwares e Referências

Introdução

- Considere dois conjuntos de dados reais e aumentado:
  - Em uma dimensão duas dimensões;

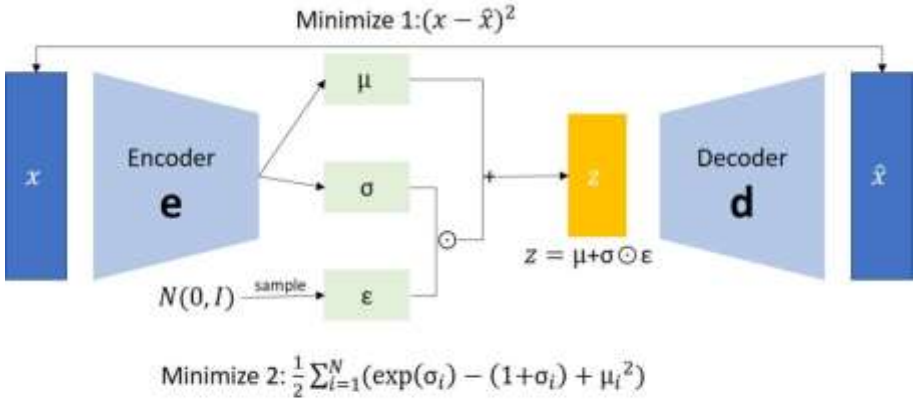




  - Dados reais complementados por dados artificialmente gerados;

Introdução

- A modelagem generativa visa construir um modelo, criando uma distribuição a partir de um conjunto de dados:
  - Os modelos generativos aprendem uma função de densidade de probabilidade a partir de um conjunto de treinamento e geram novas instâncias produzidas a partir da mesma distribuição;
- A qualidade do modelo pode ser avaliada por exemplos que ele produza e posterior comparação com dados reais;
- Visão esquemática geral de um modelo generativo:

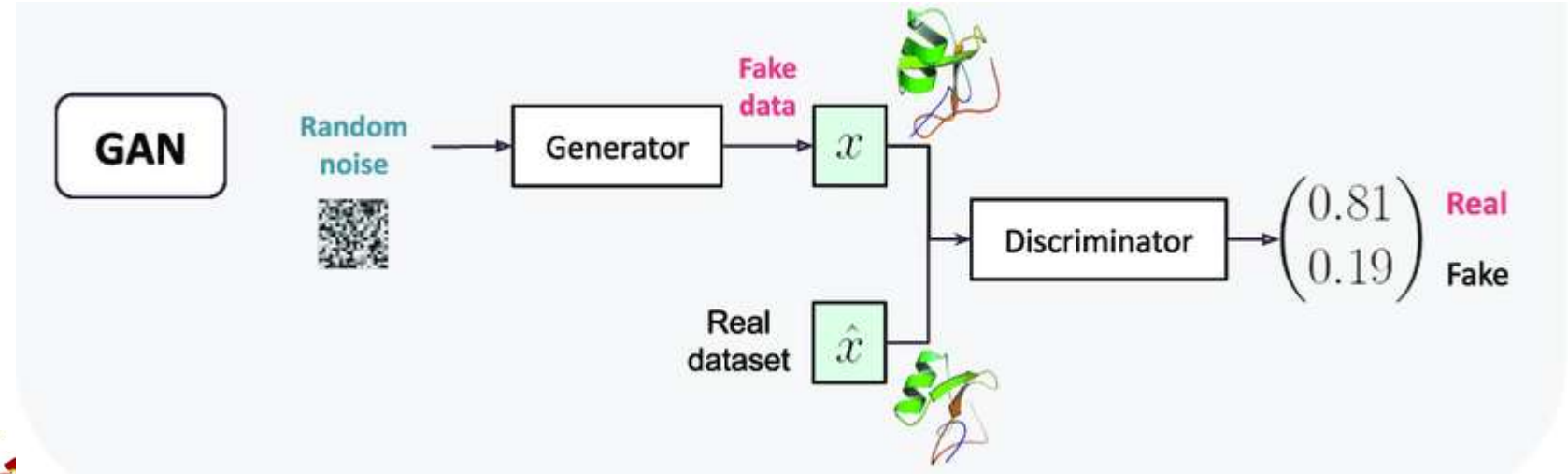


Introdução

- Uma rede adversária generativa (GAN) é formada por duas redes neurais que interagem entre si na forma de um jogo de soma zero (vitória de um agente implica na derrota do outro);
- As GANs geram novos dados sintéticos semelhantes aos reais (mesmas estatísticas do conjunto de treinamento), colocando duas redes neurais (gerador e discriminador) uma contra a outra:
  - O Gerador tenta capturar a verdadeira distribuição de dados para gerar novas amostras (de qualquer origem como imagens ou textos),
    - Gerador produz padrões para tentar enganar o discriminador, i.e., ser considerado um exemplo verdadeiro;
  - O discriminador, geralmente um classificador binário visa classificar amostras geradas reais e falsas com precisão,
    - O discriminador usa treinamento indireto, ele avalia quão realista uma entrada parece;

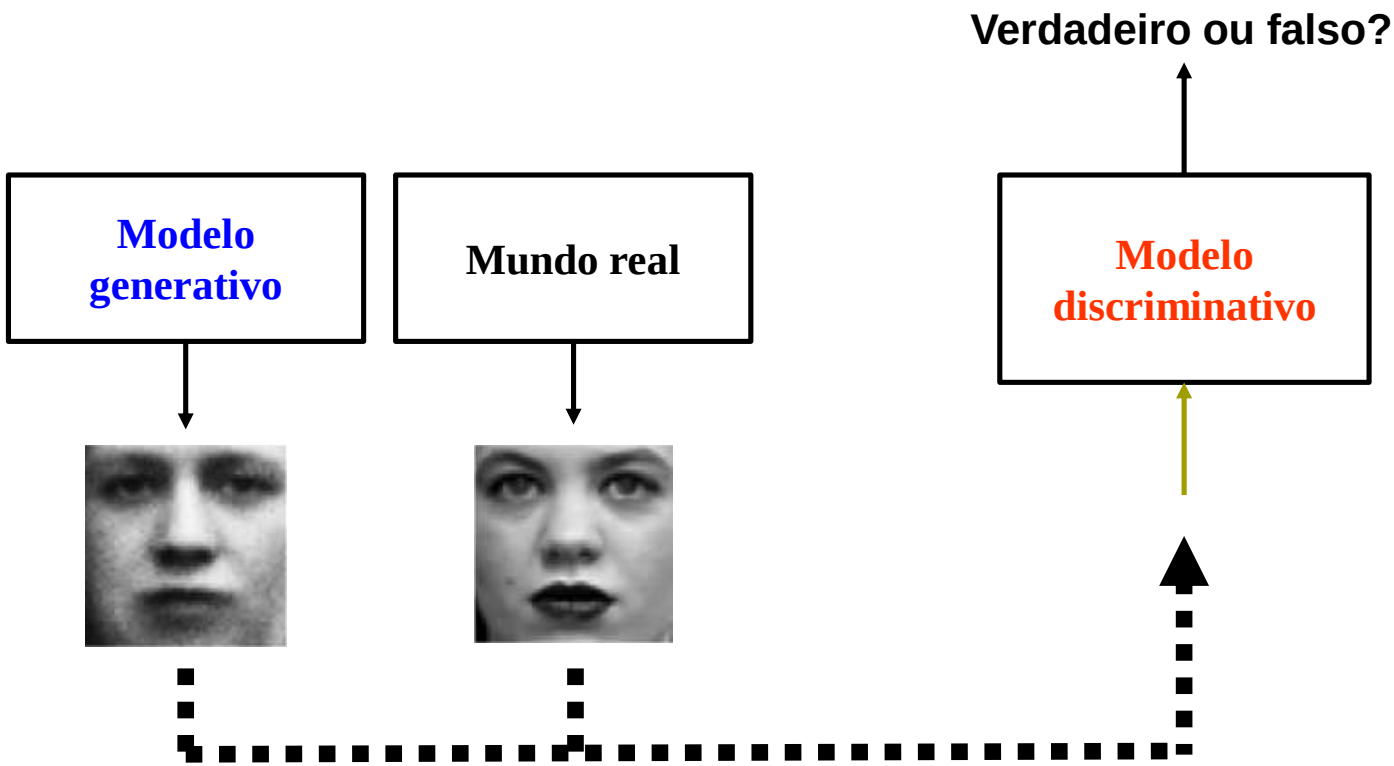
Introdução

- Uma GAN aprende a gerar novos dados com as mesmas estatísticas de seu conjunto de treinamento,
  - E.g., GAN treinado em fotografias pode gerar novas fotografias que parecem autênticas para observadores humanos;
- Originalmente emprega aprendizagem não-supervisionada,
  - Também há modelos com aprendizagem semi-supervisionada, supervisionada e por reforço;



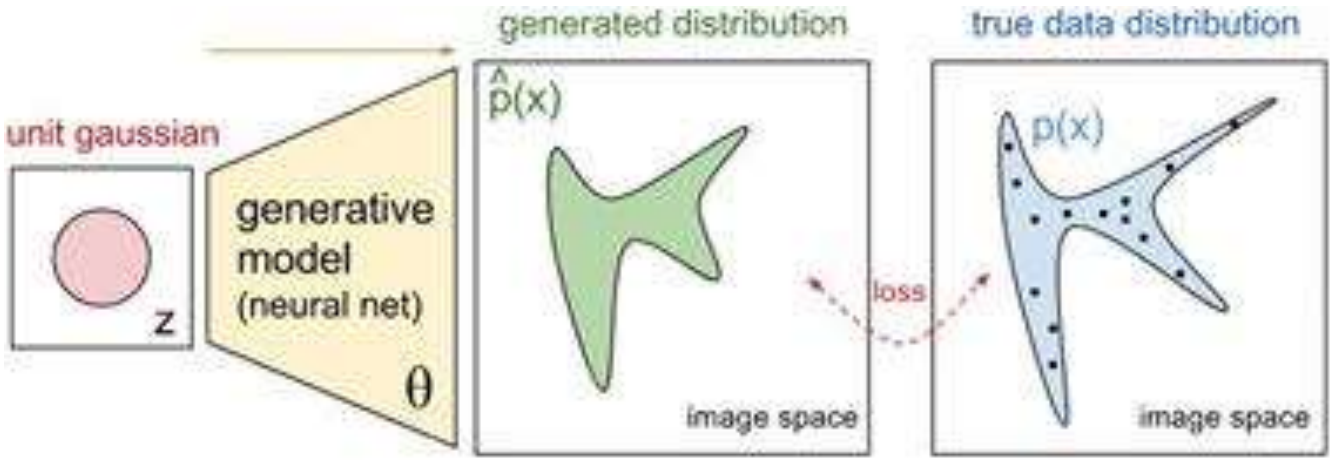
Introdução

- Ilustração da atuação das GANs:
  - Geração de uma amostra e avaliação de seu realismo;



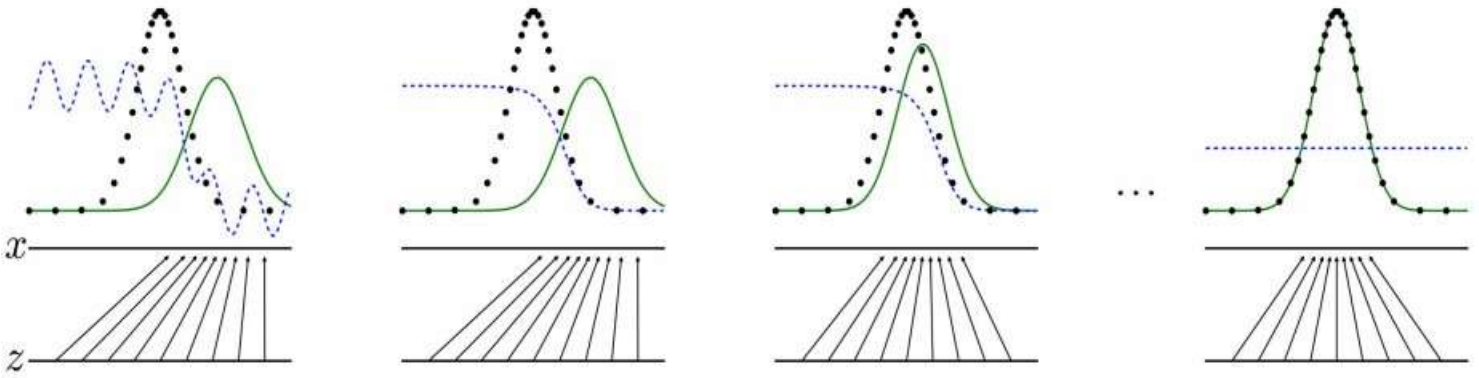
Introdução

- Treinamento de GAN:
  - A rede geradora gera dados que parecem reais depois de aprender um distribuição dos dados de treinamento;
  - A rede discriminadora, tipicamente uma CNN, aprende a rotular os exemplos como verdadeiros (reais) ou falsos (artificiais);
  - A categorização é usada para ajustar a rede geradora;



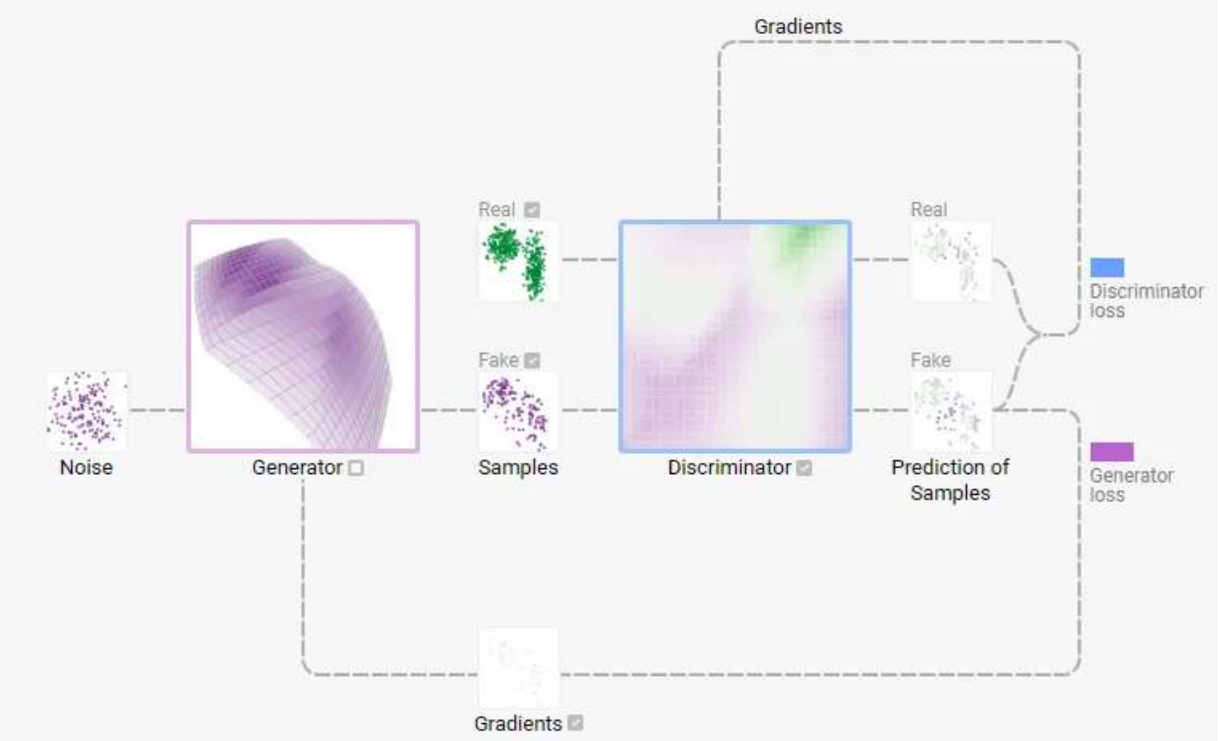
Introdução

- GANs são treinadas até que iguale a distribuição generativa (verde) à distribuição geradora de dados (linha pontilhada) e o discriminador (linha azul pontilhada) seja incapaz de diferenciar entre as duas distribuições;



Introdução

- GANs visam replicar distribuição de probabilidade,
- Para uma GAN, emprega-se funções de perda (f.p.):
  - Gerador: f.p. mede diferença entre dados gerados e de treinamento;
  - Discriminador : f.p. mede diferença entre dados gerados e reais;

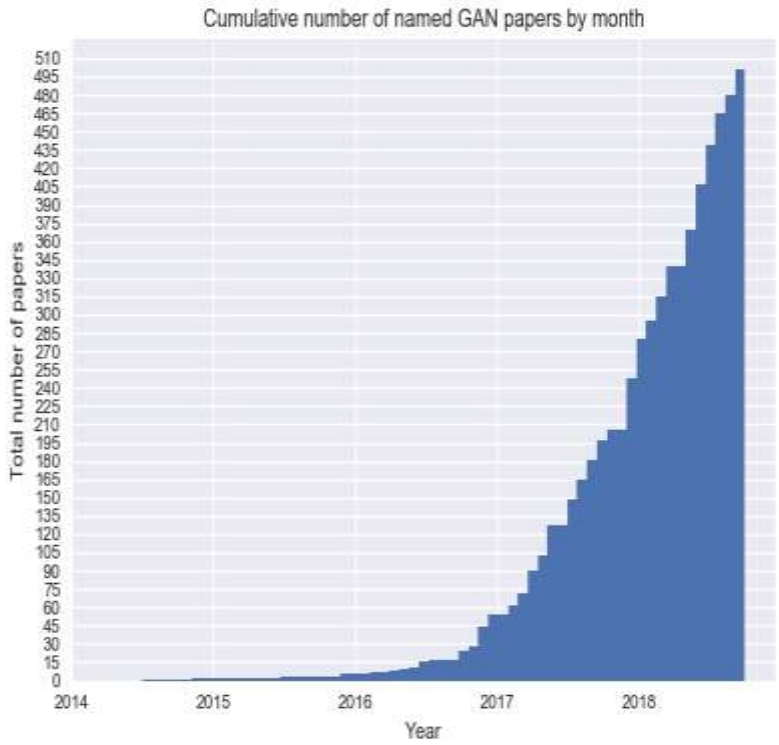


Introdução

- Algumas aplicações de GANs:
  - Geração de exemplos para conjuntos de dados de imagem, fotografias de rostos humanos, fotografias realistas, visão facial frontal, novas poses humanas, personagens de animação;
  - Tradução de imagem para imagem, de texto para imagem, semântica de imagem para foto, de vestuário;
  - Edição de fotografia, mesclagem de fotos;
  - Fotos para emojis;
  - Envelhecimento facial;
  - Pintura interna de fotos;
  - Previsão de vídeo;
  - Geração de objetos 3D;

Introdução

- Números de publicações e ilustração de uma GAN:



https://github.com/hindupuravinash/the-gan-zoo

Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *arXiv preprint arXiv:1609.04802* (2016).

Modelos Generativos

- Modelos discriminativos se caracterizam por
  - Dado um **x**, determina-se uma resposta **y** ou se estima a probabilidade p(**y**|**x**);
- Os modelos discriminativos apresentam limitações:
  - Não é possível modelar p(**x**), portanto, não se pode gerar novos dados (amostrar p(**x**));
- Modelos generativos, em geral, se caracterizam por:
  - Podem modelar p(**x**);
  - Podem gerar novos dados.

Modelos Generativos

- Modelos generativos estimam o processo probabilístico que gera um conjunto de observações $\mathcal{D}$:
  - $\mathcal{D} = \{\mathbf{x}^i, \mathbf{y}^i\}$, $i=1, ..., n$; modelos generativos supervisionados aprendem distribuição de probabilidade conjunta $p(\mathbf{x}^i, \mathbf{y}^i)$, frequentemente para calcular $p(\mathbf{y}^i|\mathbf{x}^i)$;
  - $\mathcal{D} = \{\mathbf{x}^i\}$, $i=1, ..., n$; modelos generativos não-supervisionados aprendem a distribuição de $\mathcal{D}$ para agrupamento ou amostragem de duas maneiras:
    - Por estimativa direta de $p(\mathbf{x}^i)$;
    - Introdução de variável latente $\mathbf{y}^i$ e estimativa de $p(\mathbf{x}^i, \mathbf{y}^i)$;

Modelos Generativos

- Modelos generativos estimam o processo probabilístico que gera um conjunto de observações $\mathcal{D}$:
  - Escolhe-se uma família parametrizada $p(\mathbf{x} \mid \theta)$ e aprende-se θ maximizando a log-verossimilhança:

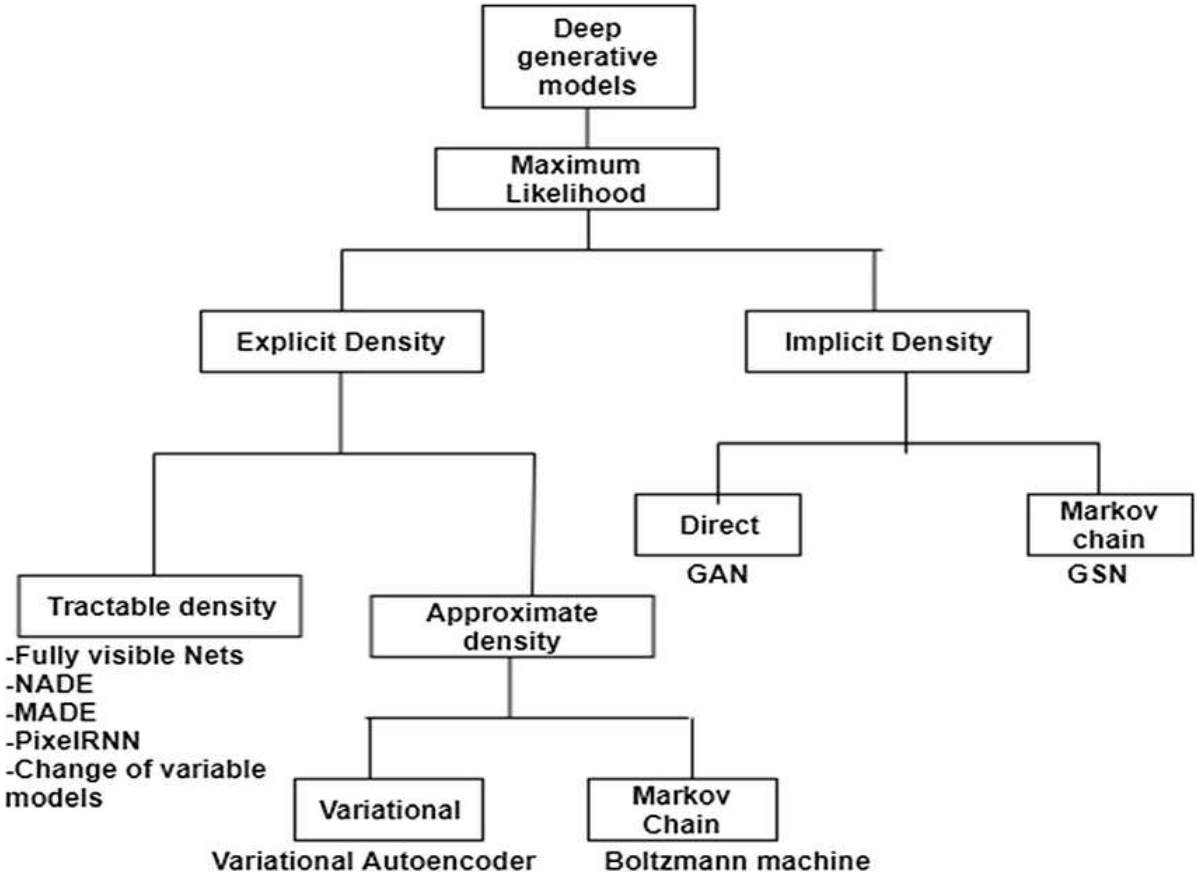
$$\theta^* = \arg\max_\theta \sum_{i=1}^{n} \log p(\mathbf{x}^i \mid \theta).$$

  - Modelos com variáveis latentes: Define-se uma distribuição conjunta $p(\mathbf{x},\mathbf{y} \mid \theta)$ e aprende-se θ maximizando a verossimilhança log-marginal:

$$\theta^* = \arg\max_\theta \sum_{i=1}^{n} \log \int p(\mathbf{x}^i, \mathbf{z}^i \mid \theta)\, \mathrm{d}\,\mathbf{z}.$$

Modelos Generativos

- Opções para estimação função de densidade:
  - Densidade explícita;
  - Densidade implícita;
- Nos modelos generativos implícitos pode-se avaliar a qualidade das amostras, logo calcula-se seu gradiente em relação aos parâmetros da rede e estes são atualizados;

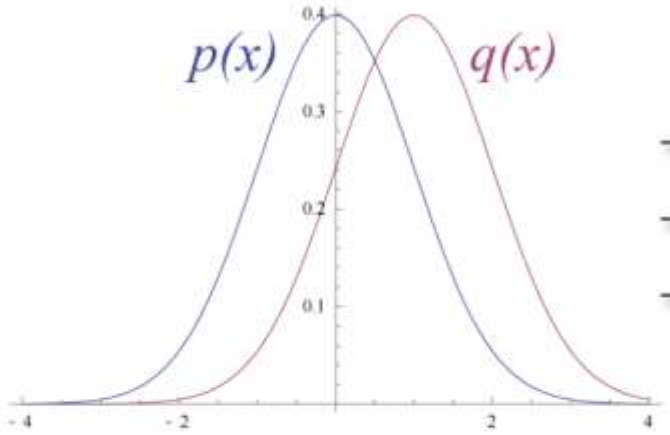


Redes Adversárias Generativas (GANs)

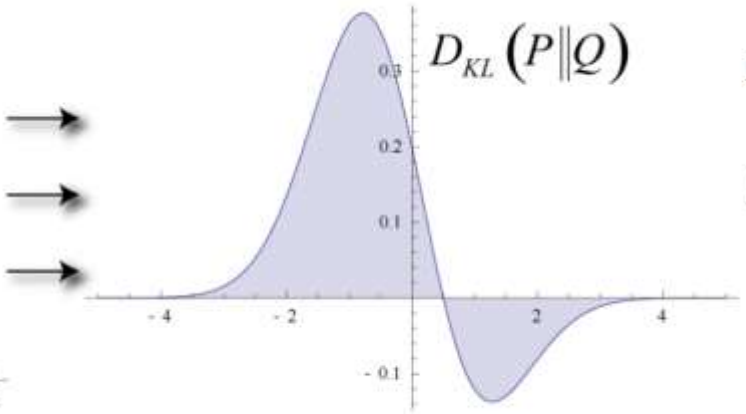
- Redes adversárias generativas (GANs) (Goodfellow et al., 2014) são baseadas em um suposto cenário de jogo no qual a rede geradora deve competir contra um adversário:
  - Uma rede geradora produz amostras;
  - A rede discriminadora (adversária) tenta distinguir amostras de dados de treinamento e amostras vindas do gerador;
  - O discriminador estima uma probabilidade que uma amostra seja um exemplo de treinamento real em vez de gerada pelo modelo;
- Redes adversárias generativas (GANs):
  - Generativa porque aprende a distribuição subjacente dos dados;
  - Adversária por ser formada por duas redes concorrentes;
  - Redes por empregar redes neurais profundas.

Redes Adversárias Generativas (GANs)

- As *Generative Adversarial Networks* (GANs) visa aproximar duas distribuições para gerar uma saída realista e de qualidade:
  - A distribuição de probabilidade de um gerador de instâncias, $P_G(.)$;
  - A distribuição de probabilidade das imagens reais, $P_{data}(.)$;
- A similaridade entre as duas distribuições é calculada por:
  - Divergência de Kullback-Leibler (KLD): Métrica não simétrica que afere entropia relativa ou diferença na informação de duas distribuições;
  - KLD mede a distância entre duas distribuições de dados;



Original Gaussian PDF's



KL Area to be Integrated

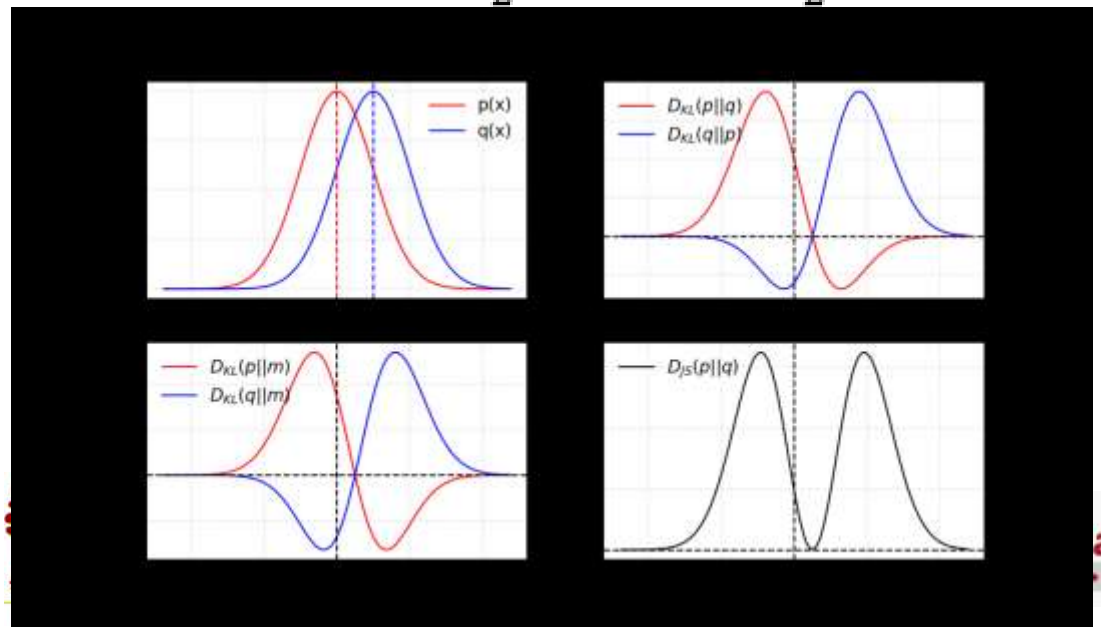
$$KL(P||Q) = \sum p(x) \log \frac{p(x)}{q(x)}$$

$$KL(Q||P) = \sum q(x) \log \frac{q(x)}{p(x)}$$

Redes Adversárias Generativas (GANs)

- ... a similaridade entre as duas distribuições é calculada por:
  - Divergência de Jensen-Shannon (JSD): Método de medir a similaridade entre duas pdfs, é simétrica e sempre tem um valor finito,
    - A raiz quadrada da JSD é chamada de distância de Jensen-Shannon;
  - Por ser simétrica, JSD compara pdfs, independentemente de qual distribuição é referência e qual é a ser comparada;

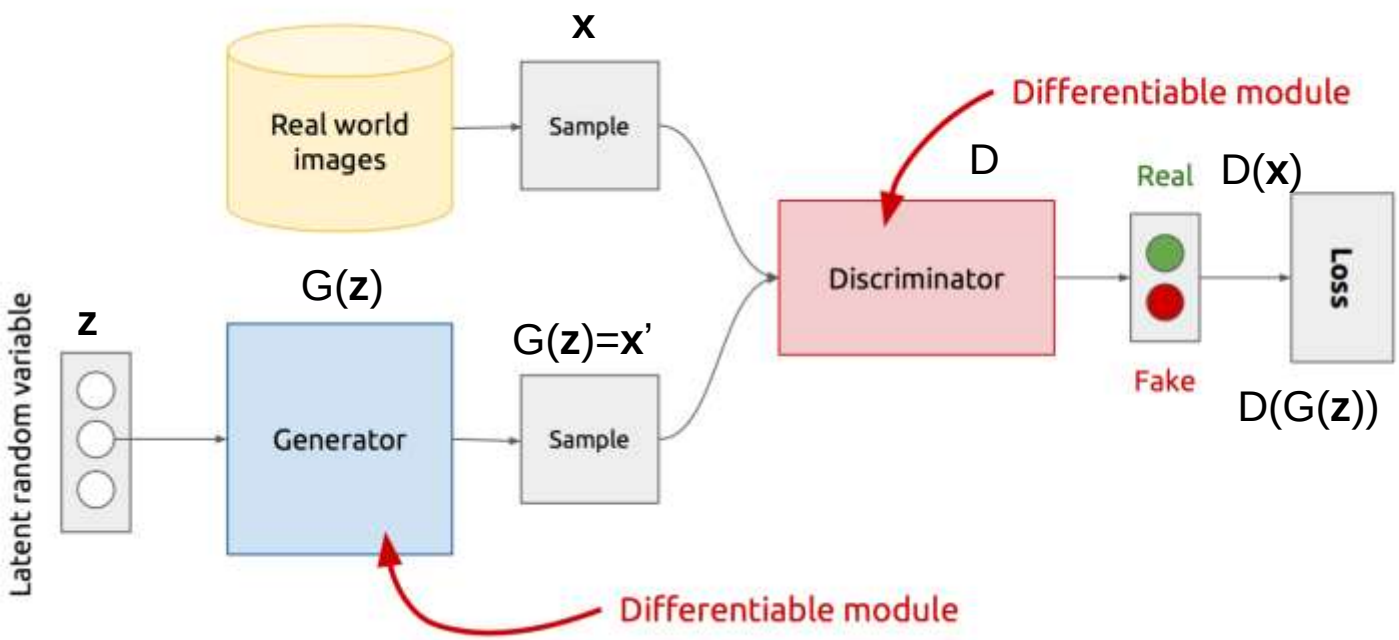
$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2} D(P \parallel M) + \frac{1}{2} D(Q \parallel M),$$ onde M= (P+Q)/2 e D=KLD;



- Distribuições gaussianas:
  - $\mathbb{N}_1(\mu=0, \sigma=1)$ = p(.);
  - $\mathbb{N}_2(\mu=1, \sigma=1)$ = q(.);
  - Média: M= $(\mathbb{N}_1+\mathbb{N}_2)/2$;
- KLD é assimétrica mas JSD é simétrica;

Redes Adversárias Generativas (GANs)

- Arquitetura geral de uma GAN:
  - **z** é um ruído aleatório (Gaussiano/Uniforme) que é entendido como a representação latente da imagem;



Redes Adversárias Generativas (GANs)

- Treinamento de uma GAN:
  - Gerador G: Entrada **z** e saída G(**z**)= **x**';
    - Dada uma distribuição a priori $P_{prior}(\mathbf{z})$, uma distribuição de probabilidade $P_G(\mathbf{z})$ é determinada pela função G;
  - Discriminador D: Entradas **x** e G(**z**)= **x**', e saída escalar, D(.);
    - A saída avalia a diferença entre $P_G(\mathbf{z})$ e $P_{data}(\mathbf{x})$;
- Define-se uma função de custo V(G,D) para D determinar diferença entre P_{data} de P_G: $G^*=\arg \min_G \max_D V(G,D)$,
  - A distribuição G pode ser alterada e não apenas seus parâmetros;
- Formulação como um jogo MinMax:
  - O discriminador visa maximizar sua recompensa enquanto o gerador busca minimizar a recompensa do discriminador;

Redes Adversárias Generativas (GANs)

- Formulação de jogo MinMax (Jensen–Shannon divergence - JSD):
  - Aproximação inspirada na teoria dos jogos:

$$\mathrm{Min}_G\mathrm{Max}_D\ V(G,D) = E_{\mathbf{x}\sim Pdata(\mathbf{x})}[\log D(\mathbf{x})] + E_{\mathbf{z}\sim PG(\mathbf{x})}[\log(1- D(G(\mathbf{z})))]$$

verossimilhança de dados reias — verossimilhança de dados gerados

  - Discriminador D (saída de 0 a 1) ajusta pesos para maximizar V:
    - Deseja-se que D(**x**) tenda a um e D(G(**z**)) tenda a zero;
  - Gerador G (saída de imagem) procura confundir o discriminador D, a fim de minimizar V:
    - Deseja-se que D(**x**) tenda a zero e D(G(**z**)) tenda a um;
- Lembrando que log (1)= 0; log (0,1)= –1; log(0) = – ∞;
- O equilíbrio de Nash desse jogo em particular é obtido em:
  - $P_{data}(\mathbf{x}) = P_G(\mathbf{z})$;
  - $D(\mathbf{x}) = ½,\ \forall \mathbf{x}$:

Redes Adversárias Generativas (GANs)

- Formulação a função de custo V(G,D) definida, alterna-se entre:
  - Elevação do gradiente para o discriminador :

$$\text{Max}_D\ V(G,D) = E_{\mathbf{x}\sim Pdata(\mathbf{x})}[\log D(\mathbf{x})] + E_{\mathbf{z}\sim PG(\mathbf{x})}[\log(1 - D(G(\mathbf{z})))]$$

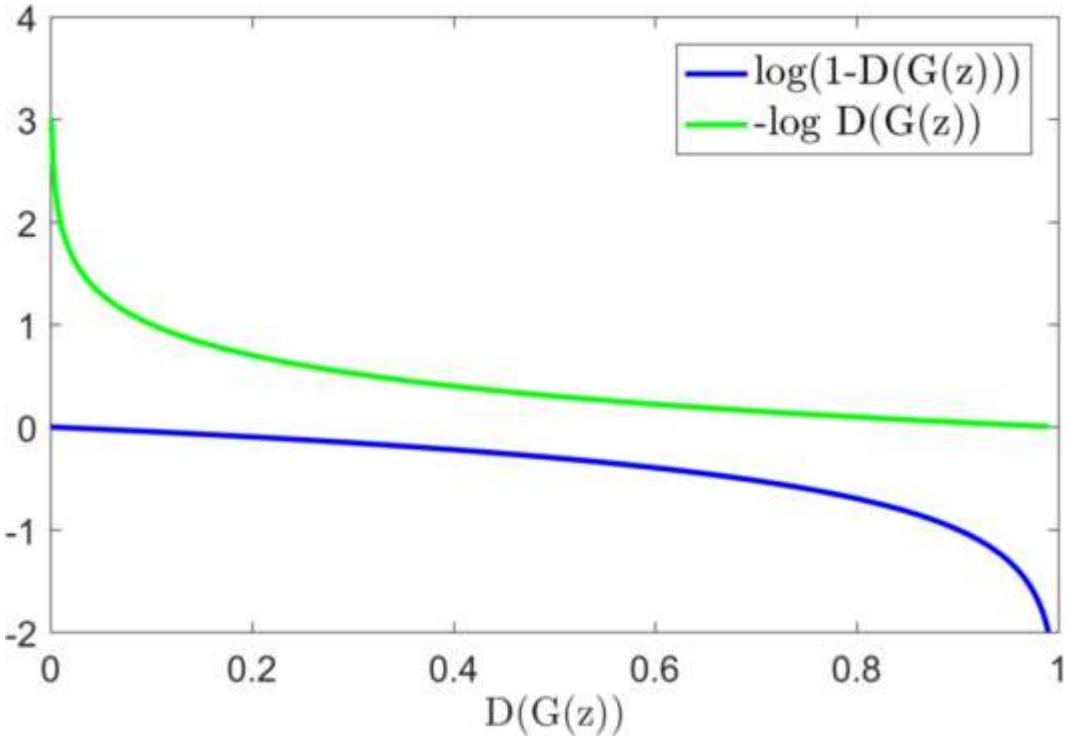
  - Redução do gradiente para o gerador :

$$\text{Min}_G\ V(G,D) = E_{\mathbf{z}\sim PG(\mathbf{x})}[\log(1 - D(G(\mathbf{z})))]$$

  - Uma amostra gerada ruim tem $D(G(\mathbf{z}))\approx 0$ e $D(\mathbf{x})\approx 1$, situação que geralmente ocorre no início do treinamento, logo o discriminador tem previsão próxima a zero, portanto, sua derivada é nula (gradiente que desaparece);

Redes Adversárias Generativas (GANs)

- Nova função de custo para o gerador:
  - $\text{Min}_G\ V(G,D) = E_{\mathbf{z}\sim PG(\mathbf{x})}[-\log D(G(\mathbf{z}))]$



Redes Adversárias Generativas (GANs)

- Aprendendo D

- Inicialize θ_d para D e θ_g para G
- Repita K vezes
- Amostre m exemplos $\{\mathbf{x}^1, \mathbf{x}^2, \ldots \mathbf{x}^m\}$ da distribuição de dados $P_{data}(\mathbf{x})$;
- Amostre m exemplos de ruído $\{\mathbf{z}^1, \ldots, \mathbf{z}^m\}$ da distribuição a priori dos dados $P_{prior}(\mathbf{z})$;
- Obtenha os dados gerados $\{\mathbf{x}^{*1}, \ldots, \mathbf{x}^{*m}\}$, $\mathbf{x}^{*i}=G(\mathbf{z}^i)$, $i= 1, ..., m$;
- Atualize os parâmetros do discriminador θ_d para maximização
- $V' \approx 1/m \, \Sigma_{i=1, ..., m} \log D(\mathbf{x}^i) + 1/m \, \Sigma_{i=1, ..., m} -\log(D(\mathbf{x}^{*i}))$;
- $\theta_d \leftarrow \theta_d + \eta \nabla V'(\theta_d)$ (gradiente ascendente)
- Fim-do-Repita

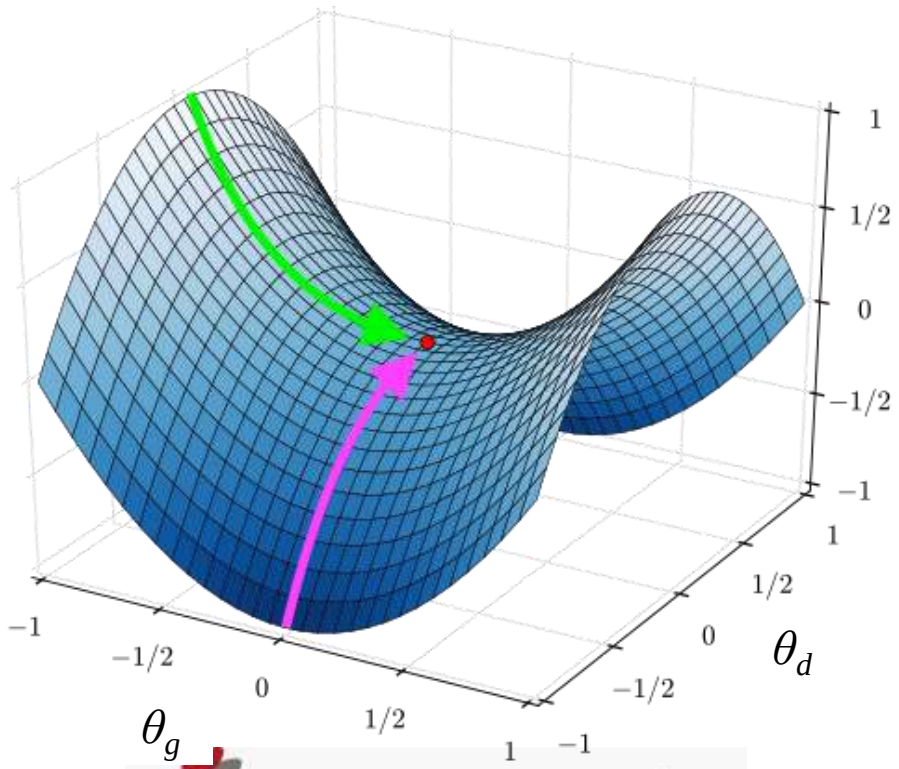
Redes Adversárias Generativas (GANs)

- Aprendendo G

- Amostre m novos exemplos de ruído $\{\mathbf{z}^1, \ldots, \mathbf{z}^m\}$; dos dados da distribuição a priori $P_{prior}(\mathbf{z})$;
- Obtenha os dados gerados $\{\mathbf{x}^{*1}, \ldots, \mathbf{x}^{*m}\}$, $\mathbf{x}^{*i}=G(\mathbf{z}^i)$, $i= 1, ..., m$;
- Atualize os parâmetros do gerador θ_g para minimização
- $V' \approx 1/m \, \Sigma_{i=1, ...,m \,)}[- \log D(\mathbf{x}^{*i})]$;
- $\theta_g \leftarrow \theta_g - \eta \nabla V'(\theta_g)$ (gradiente descendente)

Redes Adversárias Generativas (GANs)

- Visualização do aprendizado de D e G:
  - Max_D V(G,D) e Min_G V(G,D);



Redes Adversárias Generativas (GANs)

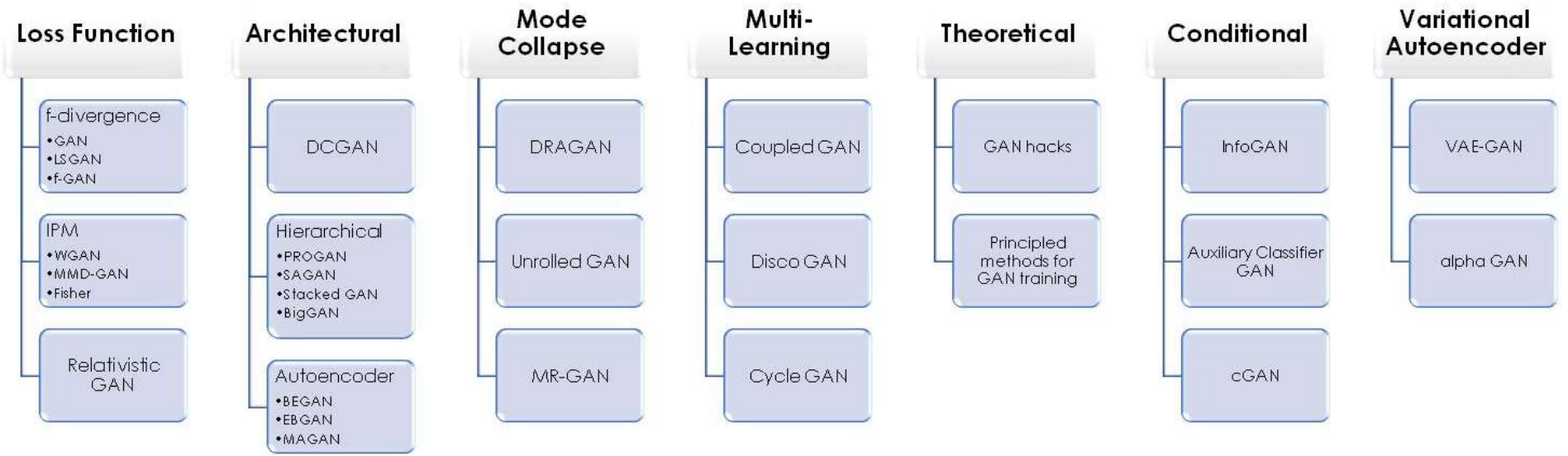
- Métricas de desempenho:
  - Acuracidade: A = (VN + VP)/(VN+FN+FP+VP);
  - Precisão: P = VP/(VP+FP);
  - Revocação (*Recall*): R = TP/(TP+FN);
  - F1-Score é a média harmônica entre precisão e revocação: F1 = 2[*(P*R)]/(P+R);
  - Curva característica operacional do receptor (ROC): Plota o gráfico entre a taxa de TP (sensibilidade) e a taxa de FP (1-especificidade) com respeito a um limiar de aceitação de um exemplo como verdadeiro positivo;
  - Área sob a curva ROC (AUC): Permite comparar diferentes curvas ROC;

Redes Adversárias Generativas (GANs)

- Algumas limitações de GAN:
  - Gradiente que desaparece para um discriminador bom demais;
  - Ausência de convergência pois o gerador e o discriminador oscilam sem atingir o equilíbrio;
  - Colapso da distribuição do gerador que é reduzida a um pequeno conjunto de exemplos;
  - Queda da distribuição do gerador que não cobre totalmente a distribuição de dados;
- Tratamentos das limitações:
  - Mudança de objetivos: Emprego de objetivo heurístico que não satura, custo de máxima verossimilhança, etc;
  - Discriminador limitado: restringir a capacidade do discriminador;
  - Agendar aprendizagem: balancear aprendizagem de D_{θ_d} e G_{θ_g};

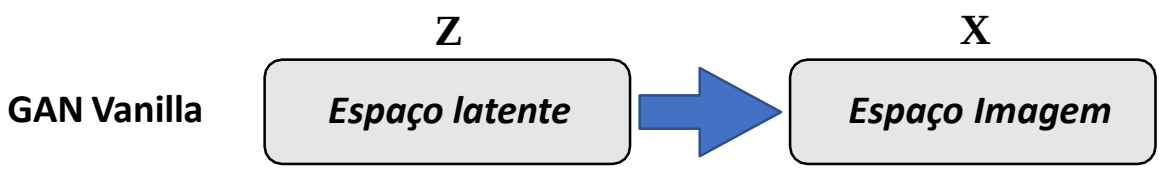
GAN – Extensões

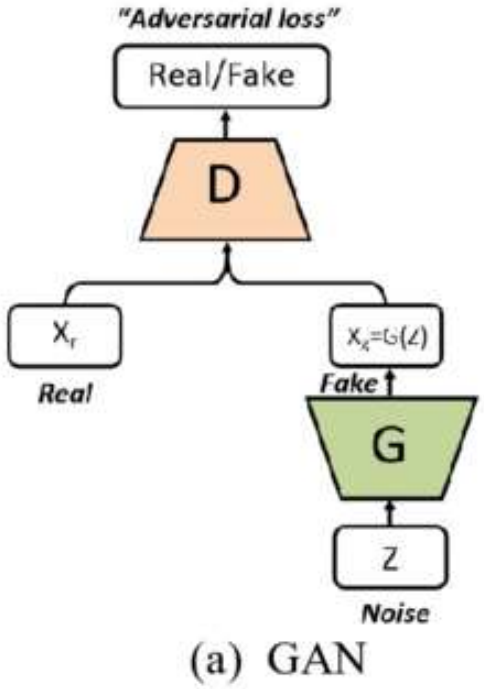
- Sete diferentes perspectivas para extensões:
  - Função de perda, arquitetura, colapso de modo, multi-aprendizagem, teórico, condicional, auto-encoder variacional;



GAN – Extensões

- (a) GAN Vanilla é o modelo GAN básico:
  - D é o Discriminador e G é o gerador;





(a) GAN

  - Seja um G pré-treinado e uma saída **x**;
  - Otimize $\min_{\mathbf{z}} \|\mathbf{x} - G(\mathbf{z})\|^2$
  - Treinamento lento;

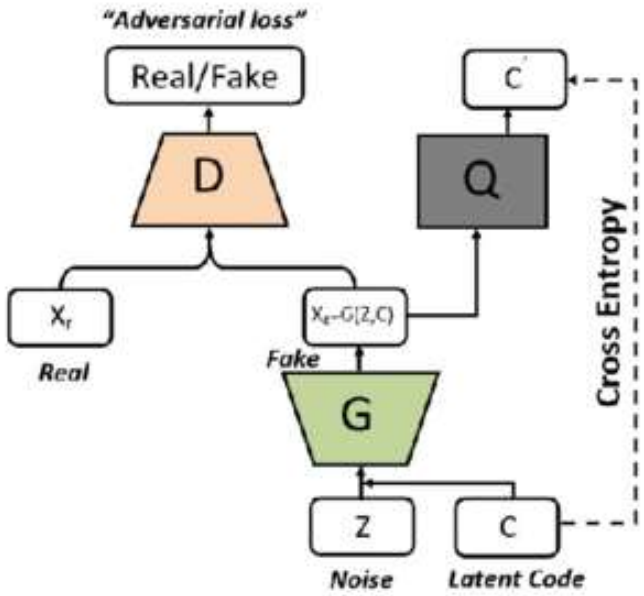
GAN – Extensões

- (b) Na InfoGAN, a função objetivo aprende representações interpretáveis e significativas maximizando informação mútua entre subconjunto fixo de variáveis de ruído e as observações:

  $Min_{G,Q}Max_D\ V(G,D,Q) = V(G,D) - \lambda L_I(G,Q)$

  – Q é distribuição auxiliar próxima da posterior; L_I é limite inferior variacional da informação mútua entre o código latente e as observações;

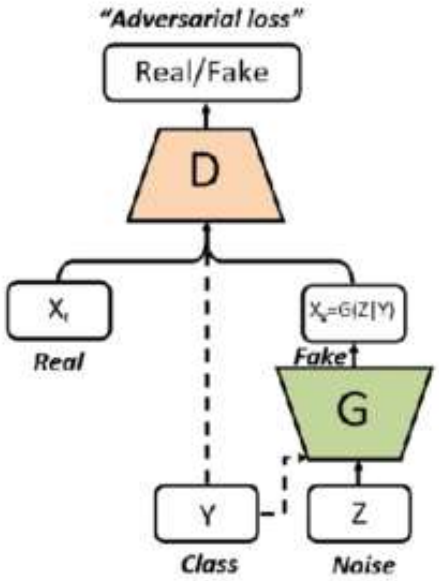
  – Sua implementação compreende outra camada totalmente conectada aos parâmetros de saída para a distribuição



(b) InfoGAN

GAN – Extensões

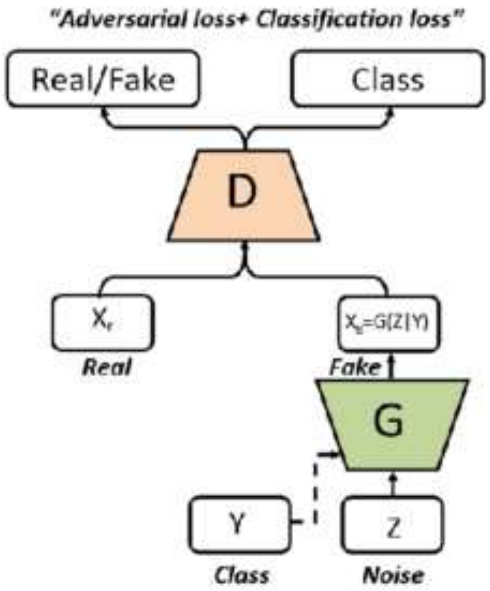
- (d) GAN condicional (cGAN) gera condicionalmente os padrões de saída;
- Nas cGANs, gerador e discriminador estão condicionados a informação auxiliar (rótulos de classe ou dados),
  - O modelo pode aprender mapeamento multimodal de entradas para saídas devido às informações contextuais;
  - A maioria das variantes GAN pode ser modificada para incluir cGAN;
  - cGAN faz G criar amostras específicas, corrigindo assim o problema de colapso do modo;



(d) CGAN

GAN – Extensões

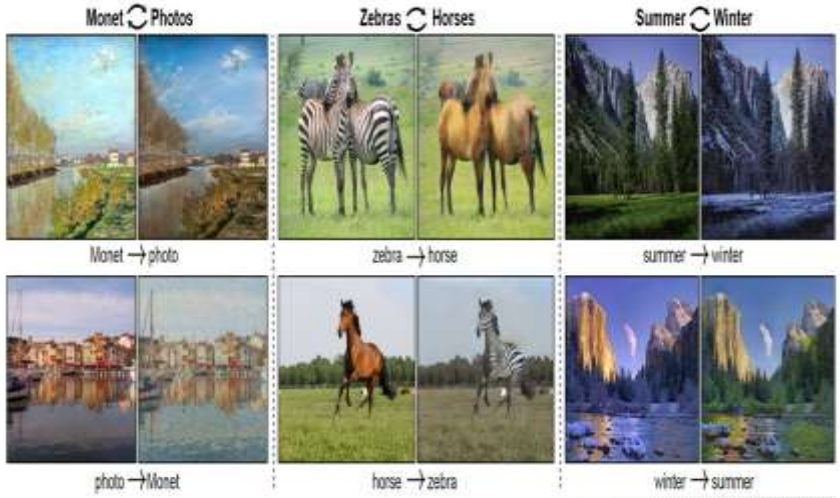
- (e) GAN com Classificador Auxiliar (ACGAN), como o cGAN, tem gerador que recebe um ponto no espaço latente e um rótulo, porém, seu discriminador não recebe a informação de contexto;
- Modelo do Gerador:
  - Entrada: Ponto aleatório do espaço latente e o rótulo da categoria;
  - Saída: Imagem gerada;
- Modelo de discriminador:
  - Entrada: Imagem;
  - Saída: Probabilidade de que a imagem fornecida seja real;



(e) ACGAN

GAN - Aplicações

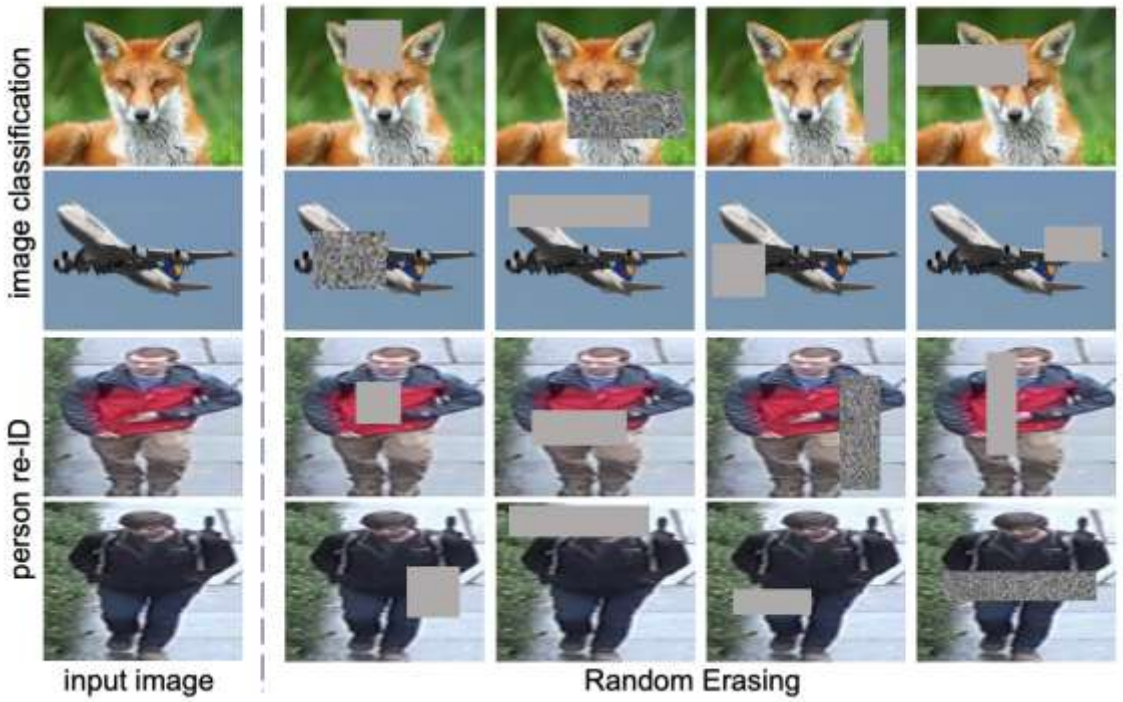
- Visão computacional:
  - Geração de imagens (CGAN, DCGAN);
  - Tradução de imagem (Ciclo GAN);
  - Super-resolução (SRGAN, ProGAN);
  - Envelhecimento facial (StarGAN, AgeC-GAN);
  - Síntese de textura (Estilo GAN);
  - Texto para imagens (StackGAN, PSGAN);
  - Detecção de objetos (GAN perceptual);



GAN - Aplicações

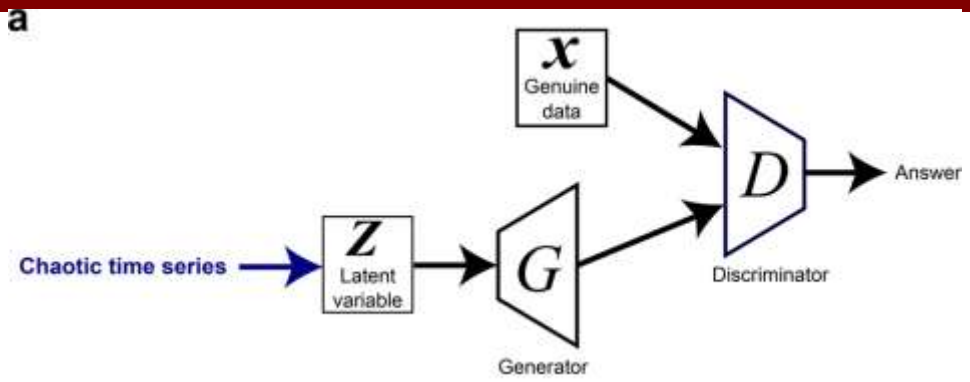
- Geração de dados:
  - Aumento dos conjuntos de dados de treinamento;
  - Criação de novos dados pelo gerador da GAN;
  - Uso mais frequente em imagens que em dados tabulares;

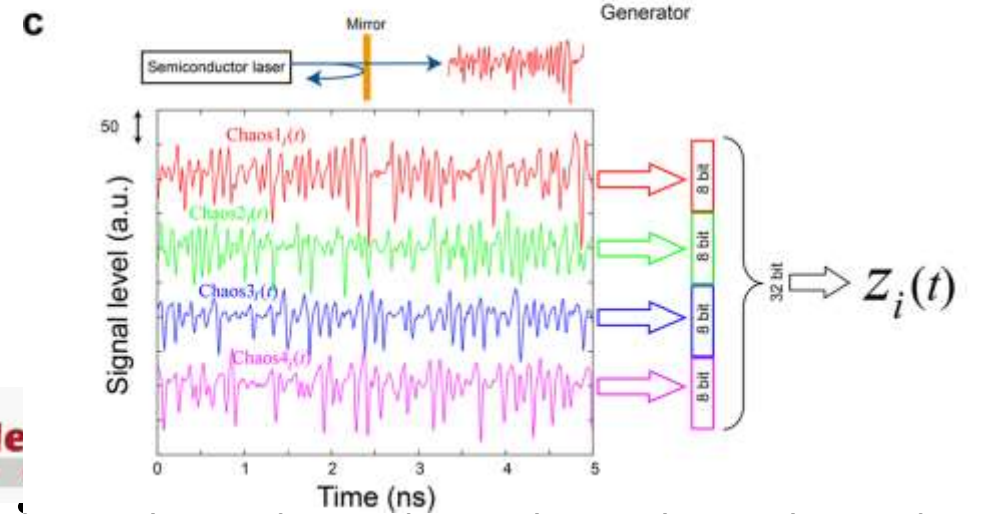
Aumento de dados
Para o YOLOv4



GAN - Aplicações

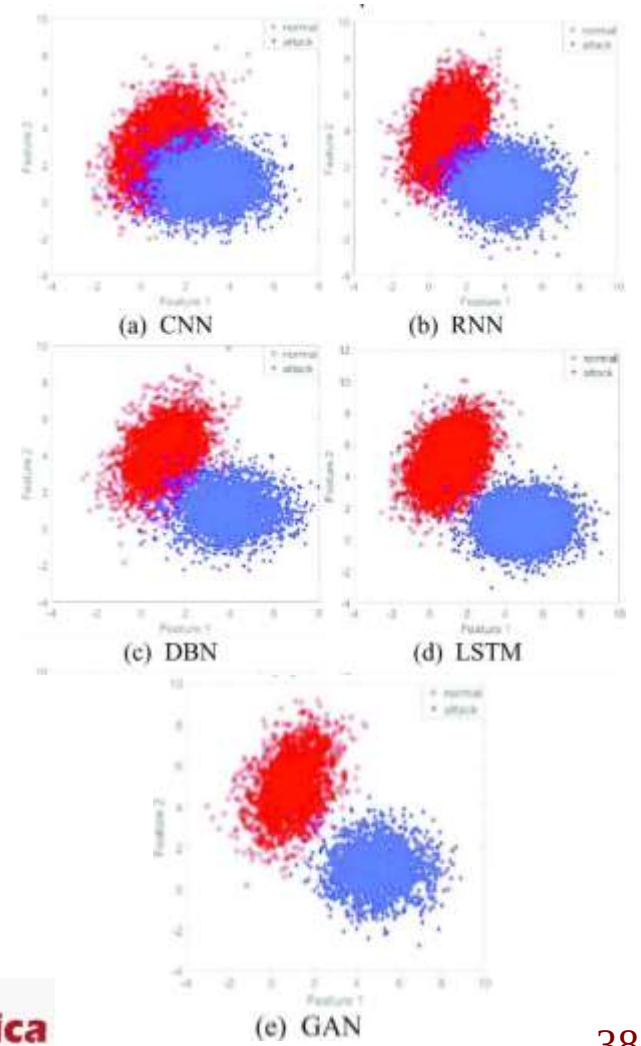
- Séries temporais:
  - GANs condicionais permite uso de redes neurais recorrentes para gerar dados de séries temporais;
  - Aplicações: modelagem econômica, previsão de estoque, previsão de mortalidade, valor em risco, projeções financeiras, etc.





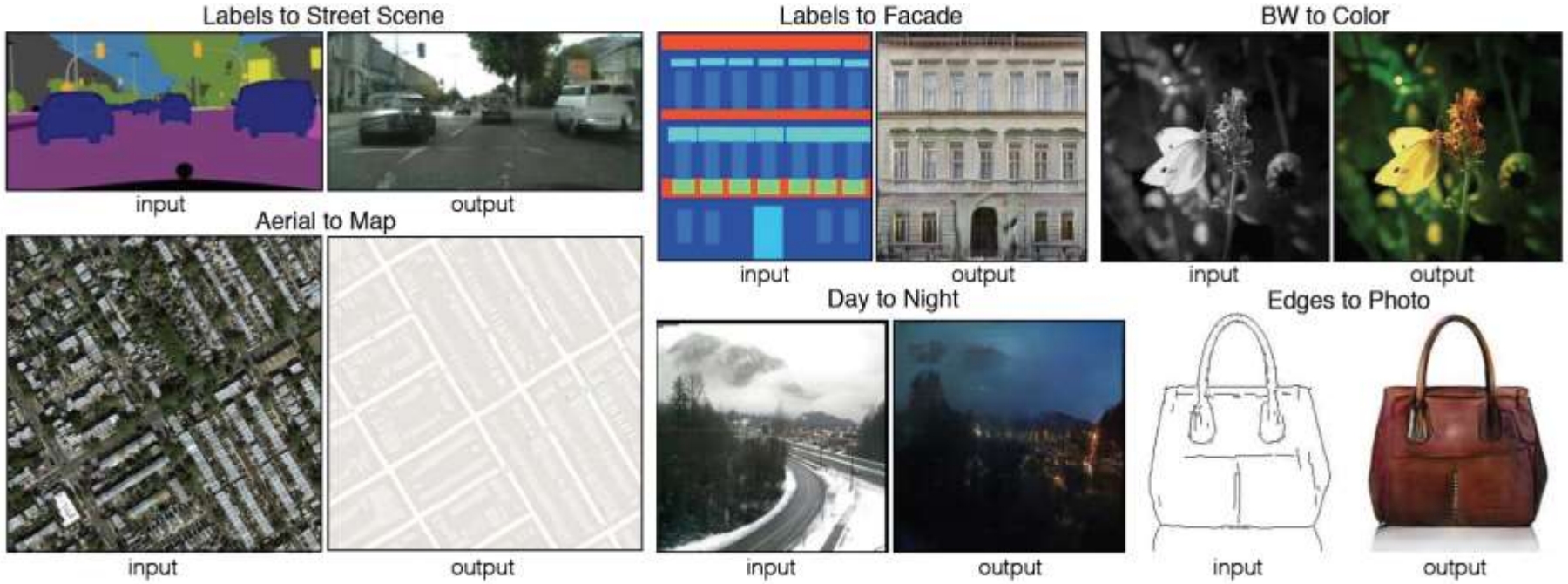
GAN - Aplicações

- Detecção de anomalia:
  - Identificação de itens raros ou eventos suspeitos com respeito à maioria dos dados;
  - Exemplo: Simulação de tráfego em rede local de computadores que apresenta tráfego normal e comportamentos anormais;



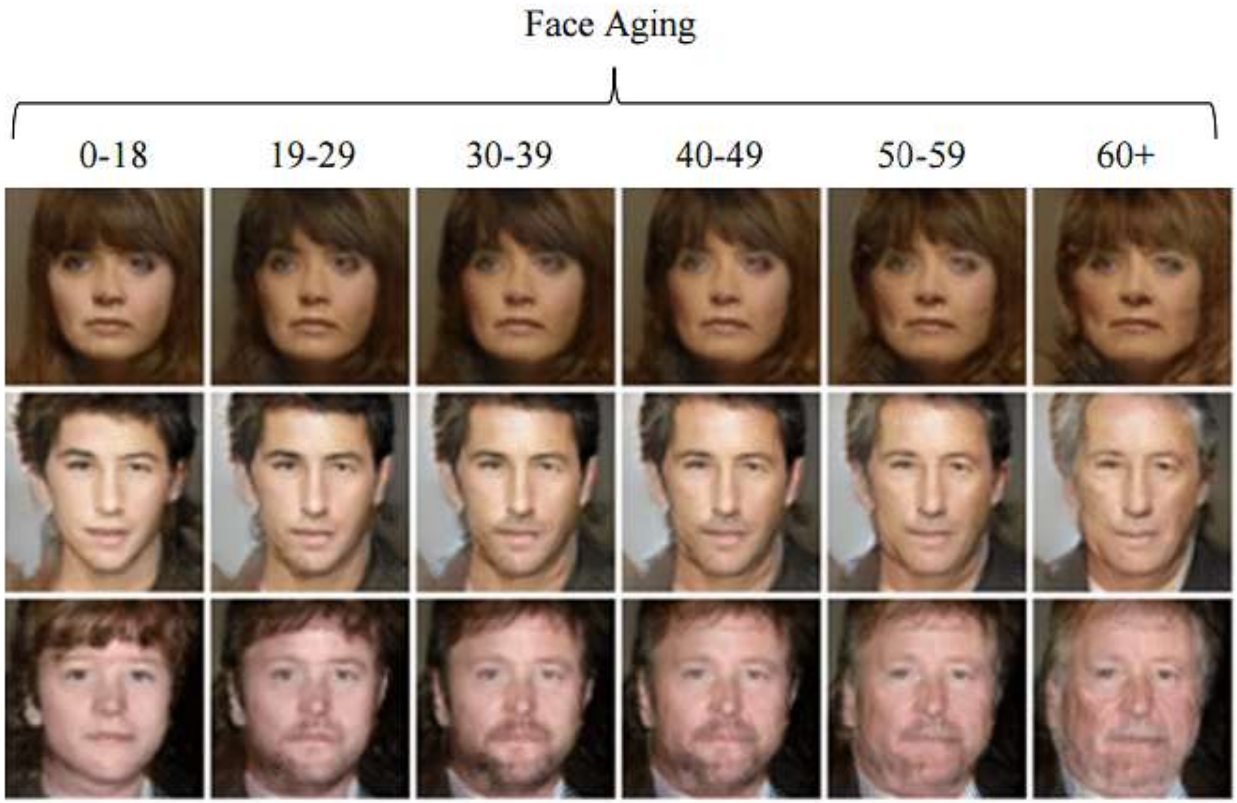
GAN - Aplicações

- Tradução de imagem para imagem:
  - DCGAN tem treino condicionado às imagens do domínio fonte;



GAN - Aplicações

- Envelhecimento facial:
  - GANs condicional pode renderizar rosto envelhecido;
    - Pode ser útil na busca de crianças desaparecidas;



Softwares

- GAN Lab: https://poloclub.github.io/ganlab/
- TF-GAN: https://github.com/tensorflow/gan
- Keras GAN: https://github.com/keras-team/keras-contrib/tree/master/examples/improved_wgan
- DCGAN TensorFlow: https://github.com/carpedm20/DCGAN-tensorflow
- Deep Convolutional GANs: https://github.com/carpedm20/DCGAN-tensorflow
- Wasserstein GAN: https://github.com/martinarjovsky/WassersteinGAN

Referências

- 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).
- Asimopoulos, D. C., Nitsiou, M., Lazaridis, L., & Fragulis, G. F. (2022). Generative Adversarial Networks: a systematic review and applications. In *SHS Web of Conferences* (Vol. 139, p. 03012). EDP Sciences.
- Dash, A., Ye, J., & Wang, G. (2021). A review of Generative Adversarial Networks (GANs) and its applications in a wide variety of disciplines--From Medical to Remote Sensing. *arXiv preprint arXiv:2110.01442*.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M.,Xu, B.,Warde-Farley, D.,Ozair, S.,Courville, A., & Bengio,Y. (2014). Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 27: 2672–2680.
- Hong, Y., Hwang, U., Yoo, J., & Yoon, S. (2019). How generative adversarial networks and their variants work: An overview. *ACM Computing Surveys*, 52(1), 1-43.
- Jabbar, A., Li, X., & Omar, B. (2021). A survey on generative adversarial networks: Variants, applications, and training. *ACM Computing Surveys*, 54(8), 1-49.
- Li, Y., Wang, Q., Zhang, J., Hu, L., & Ouyang, W. (2021). The theoretical research of generative adversarial networks: an overview. *Neurocomputing*, 435, 26-41.