Assignment 4: GANs + Autoencoders

Juan Belieni e Juliana Carvalho

4 de novembro de 2023

Sumário

1 Introdução

O código refente a este assignment está disponível no Colab.

2 Conceitos e implementação

2.1 GAN

GANs (Generative Adversarial Nets) são redes que possuem duas sub-redes associadas: uma generativa G que aprende a distribuição do dado por meio de uma transformação $g : \mathbb{R}^n \to \mathcal{X}$, que leva um ruído aleatório gerado por uma distribuição conhecida para o espaço dos dados, e uma discriminadora D que classifica se um certo dado $x \in \mathcal{X}$ é real (amostrado do conjunto de dados reais) ou falsa (gerada pela rede G) [1].

A implementação dessa rede é feita por meio da criação da classe `GAN`, que herda a classe base `Model` do *Keras* [2]. A instância dessa classe recebe 4 parâmetros:

- `generator_dims`: dimensões da rede generativa G;
- `discriminator_dims`: dimensões da rede discriminadora D;
- `noise_dim`: dimensão do ruído que vai ser amostrado;
- `noise_dist`: método que amostra valores de uma certa distribuição.

2.2 VAE

VAE (Variational Autoencoders), são estruturas propostas por [3], onde as redes neurais são usadas como codificadores e decodificadores probabilísticos.

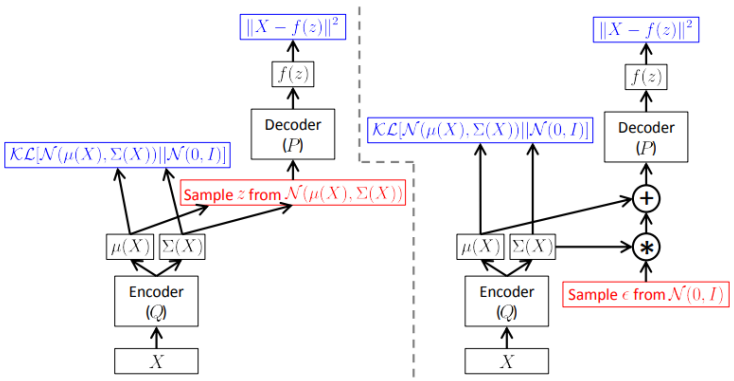

Figura 1: VAE sem (esquerda) e com (direita) o truque da reparametrização

Melhoramos o código disponibilizado no Keras [4], modificando a classe do VAE para implememntar os códigos do encoder e o decoder dentro dela. Essa classe permite alterarmos o tamanho do espaço latente ao construí-la, já que o tamanho do espaço latente é um dos seus parâmetros de inicialilzação.

- `plot_reconstrucao(original, reconstruido)` encapsulamos o primeiro código de plot, disponibilizado para o AE em na função `plot_reconstrucao(original, reconstruido)`, possibilitando, dessa forma, sua reusabilidade no notebook para plotar os resultados do VAE.
- `plot_latent_space` modificamos a função original para aceitar mínimos e máximos de x e y, invés de aceitar apenas o parâmetro `scale`.
- `sintetiza_autoencoder`: recebendo a dimensão e o dataset, usa o autoencoder para sintetizar as imagens, mostrando a original e a sintetizada.

Ademais, calculamos também a loss no treino e no teste antes de sintetizar os dados para o AE.eue

Para o VAE, reproduzimos os mesmos passos do espaço latente de tamanho 2 para o espaço latente de tamanho 64

3 Experimentos e resultados

3.1 GAN

Os experimentos com GANs foram feitas a partir de um modelo base, onde:

- a rede generativa G possui uma camada oculta com 256 nós;
- a rede discriminadora D possui uma camada oculta com 256 nós;
- o ruído possui dimensão 16 e é amostrado de uma normal.

Foram realizados 6 experimentos, onde:

- 3 redes variaram o tamanho da dimensão do ruído (8, 16 e 32);
- 3 redes variaram a distribuição da qual o ruído é amostrado (normal, uniforme e Poisson).

Os resultados para a *loss* das duas redes podem ser encontrados na tabela 1. Também é possível ver uma imagem gerada para cada experimento na figura 2.

		Ruído		*Loss*	
Exp.	Nome	Dimensão	Distribuição	Rede generativa (G)	Rede discriminadora (D)
1	`gan_dim_8`	8	Normal	2.264	0.400
2	`gan_dim_16`	16	Normal	1.968	0.425
3	`gan_dim_32`	32	Normal	2.255	0.294
4	`gan_dist_normal`	16	Normal	2.319	0.396
5	`gan_dist_uniform`	16	Uniforme	14.545	0.000
6	`gan_dist_poisson`	16	Poisson	14.074	0.000

Tabela 1: resultados dos experimentos com GANs.

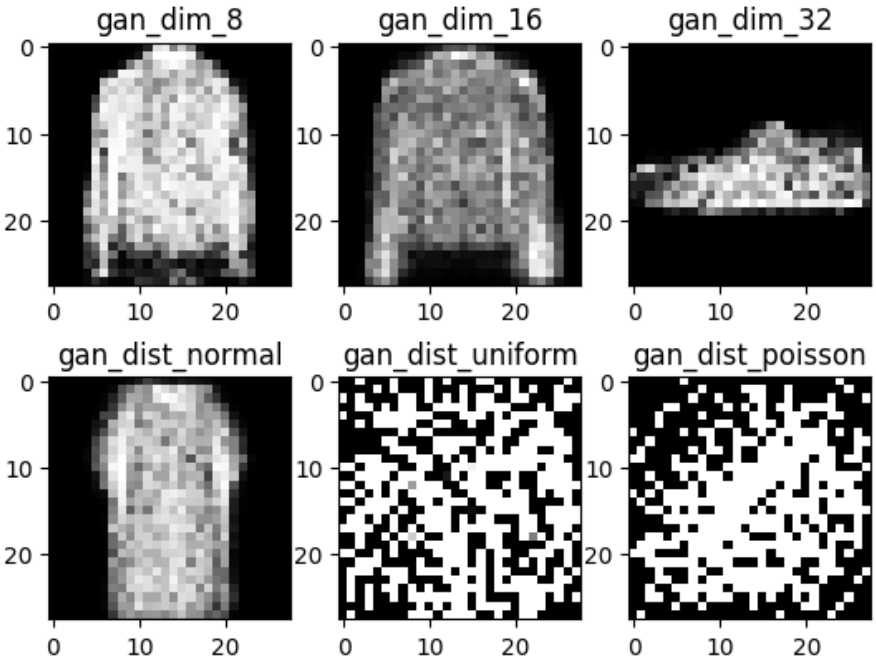

Figura 2: imagens geradas a partir das GANs treinadas nos experimentos.

Como é possível observar, apenas as GANs que amostram dados de uma distribuição normal para o ruído conseguem produzir resultados satisfatórios. Possivelmente, é possível encontrar bons resultados para outras distribuições ao variar os parâmetros dessas distribuições.

No entanto, já que as GANs com ruídos amostrados de uma uniforme e de uma Poisson não conseguiram convergir, é interessante notar que a *loss* das respectivas redes discriminadoras ficou muito próximo de zero.

3.2 AE

Arquitetura do Autoencoder:

Encoder:

- Camada de achatamento (Flatten) para transformar a entrada em um vetor.
- Camada densa com `latent_dim` unidades e ativação ReLU para gerar a representação latente.

Decoder:

- Camada densa com 784 unidades (para reconstruir a entrada original de 28×28) e ativação sigmoide para valores entre 0 e 1.
- Camada de remodelagem (Reshape) para transformar o vetor de saída de volta em uma imagem 28×28.

A entrada passa pelo encoder para reduzir a dimensionalidade para `latent_dim`, e depois é reconstruída pelo decoder para produzir uma saída que se assemelha à entrada original.

Autoencoder treinado no train:

Experimento	Dimensão	Val Loss	Loss
1	16	0.0176	0.0174
2	64	0.0089	0.0088
3	128	0.0056	0.0056

Tabela 2: Autoencoder com dimensões e valores de loss

Observe que a loss final é próxima da loss de validação. O autoencoder de tamanho 128 tem a menor loss.

3.3 VAE

O VAE consiste das seguints camadas:

Encoder:

- Camada de convolução 2D com 32 filtros, ativação ReLU e stride 2.
- Camada de convolução 2D com 64 filtros, ativação ReLU e stride 2.
- Camada de achatamento (flatten).
- Camada densa com 16 unidades e ativação ReLU para gerar z_{mean}.
- Camada densa com 16 unidades e ativação ReLU para gerar $z_{\text{log\_var}}$.

Decoder:

- Camada densa com $7 \times 7 \times 64$ unidades e ativação ReLU para expandir z de volta.

- Camada de remodelagem (reshape)
- Camada deconvolutiva (ou transposta) 2D com 64 filtros, ativação ReLU e stride 2.
- Camada deconvolutiva 2D com 32 filtros, ativação ReLU e stride 2.
- Camada deconvolutiva 2D com 1 filtro, ativação sigmoide e mesma dimensionalidade da entrada ($28 \times 28 \times 1$).

Experimento	Dimensão	Reconstruction Loss	KL-Loss	Loss
1	2	6.3740	255.5308	262.1913
2	64	289.8445	4.1817	294.3857

Tabela 3: VAE com dimensões e valores de loss

Obserrve que a Loss final dass duas dimensões é relativamente próxima, sendo a de dimensão 2 menor que a de dimensão 64. O experimento de dimensão 2 apresentou loss menor do que o de tamanho 64, enquanto a loss da desigualdade KL foi maior no primeiro experimento.

4 Síntese dos dados

Para cada caso abaixo, mostramos 10 imgens sintetizadas e suas respectivas originais. De modo geral, a síntese de dados de treino e teste capturam os detalhes mais importantes da imagem, descartando os detalhes e preservando os contornos. A síntese de dados a partir de um ruído aleatório parece mostrar alguns padrões como se vê abaixo:

4.1 AE

4.1.1 A partir do treino

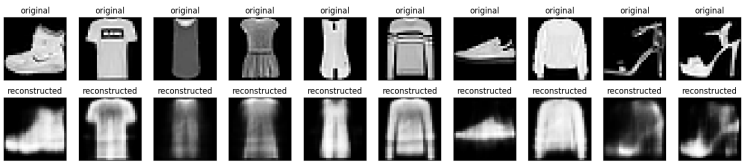

Figura 3: AE: Síntese treino de dimensão 16

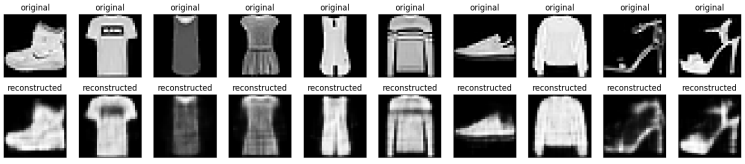

Figura 4: AE: Síntese treino de dimensão 64

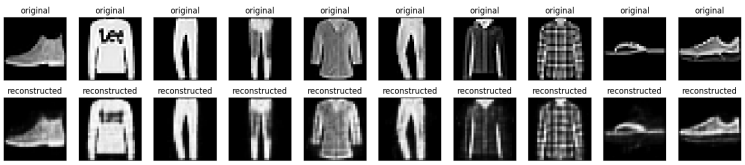

Figura 5: AE: Síntese treino de dimensão 128

4.1.2 A partir do teste

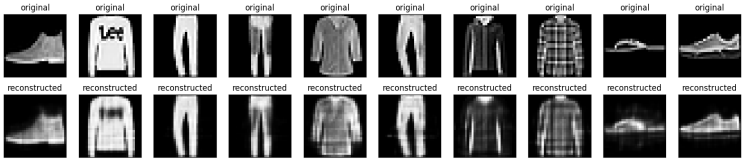

Figura 6: AE: Síntese teste de dimensão 64

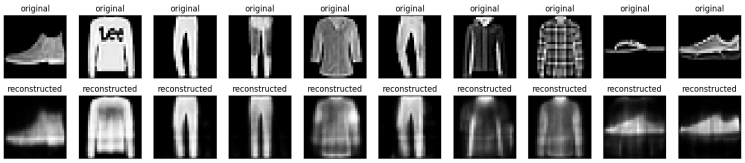

Figura 7: AE: Síntese teste de dimensão 16

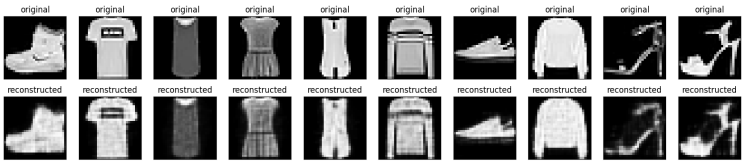

Figura 8: AE: Síntese teste de dimensão 128

4.1.3 Ruído aleatório

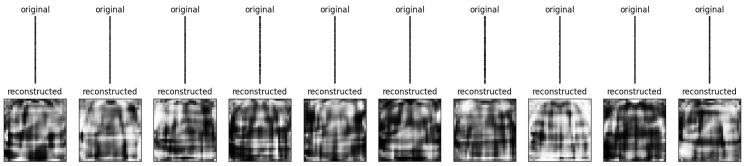

Figura 9: AE: Síntese ruído de dimensão 64

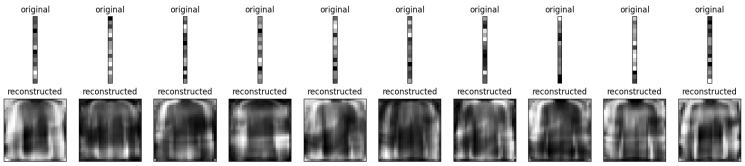

Figura 10: AE: Síntese ruído de dimensão 16

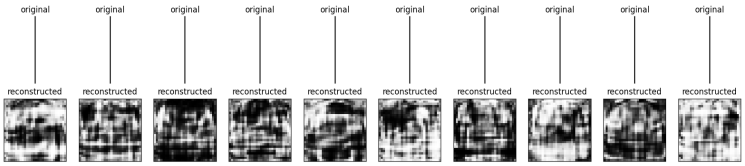

Figura 11: AE: Síntese ruído de dimensão 128

4.2 VAE

4.2.1 A partir do treino

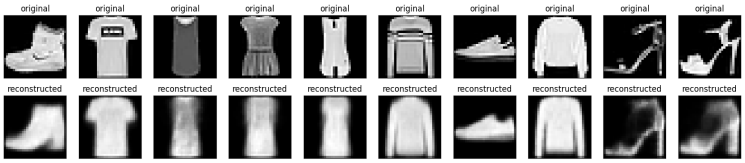

Figura 12: VAE: Síntese de imagem a partir dos dados de treino, dimensão latente de tamanho 2. Vê-se que as imagens perdem os detalhes, como a marca no tênis e na blusa, mas mesmo assim mantêm um contorno bem parecido com as originais. As imagens sintetizadas parecem ser mais claras.

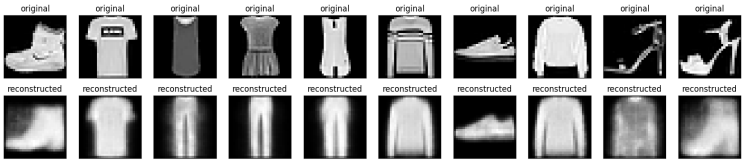

Figura 13: VAE: Síntese treino de dimensão 64 para comparação

4.2.2 A partir do teste

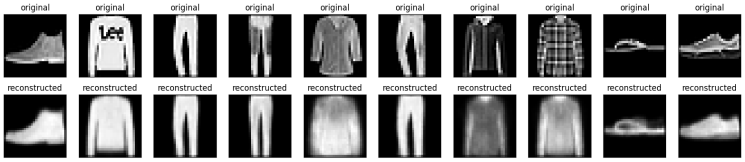

Figura 14: VAE: Síntese de imagem a partir dos dados de teste, dimensão latente de tamanho 2. Vê-se que as imagens perdem os detalhes, como a marca na blusa, mas mesmo assim mantêm um contorno bem parecido com as originais. As imagens sintetizadas parecem ser mais claras, como no caso da blusa xadrez que não mostra seus contornos.

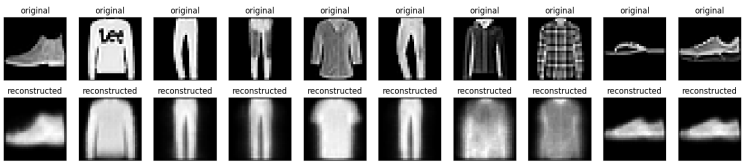

Figura 15: VAE: Síntese teste de dimensão 64 para comparação

4.2.3 Ruído aleatório

Dimensão 2

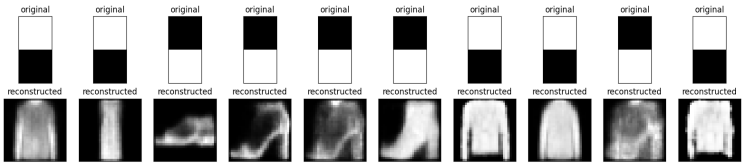

Figura 16: Síntese de imagem a partir dos dados de ruído aleatório, dimensão latente de tamanho 2. É possível ver que os sapatos (3 à 5 imagem, 9a imagem) estão associados ao original onde a parte de cima é preta, as blusas mais associadas às partes de baixo pretas.s

Dimensão 64

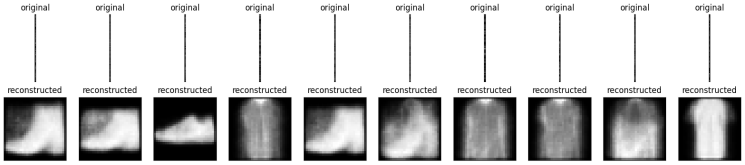

Figura 17: Síntese de imagem a partir dos dados de ruído aleatório, dimensão latente de tamanho 64. Note que as imagens estão mais borradas que a versão de dimensão 2.

5 Matrizes de reconstrução do espaço latente de tamanho 2

Considerando o plot para dados igualmente espaçados no intervalo de -1 a 1 nos eixos horizontais e verticais, temos os seguintes plots:

5.1 VAE

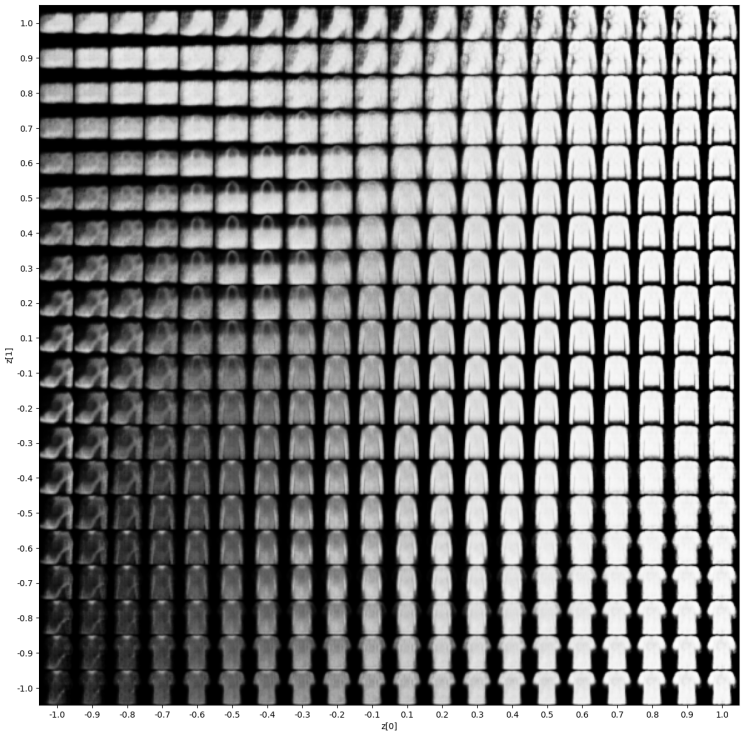

Figura 18: Reconstrução a partir do VAE

5.2 AE

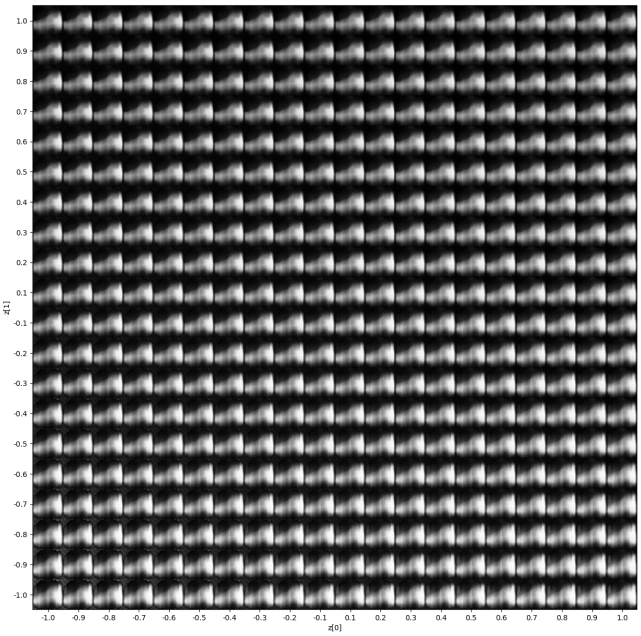

Figura 19: Reconstrução a partir do AE, no range -1 e 1 para os dois eixos.

Como para o conjunto de teste a média e o desvio padrão do espaço latente encodificado no conjunto de teste, são:

- Média: [5.6604557, 18.774605]
- Desvio Padrão: [4.5718417, 13.367]

Podemos atualizar o plot para compreender os novos ranges dos eixos horizontais e verticais, representando 95% dos dados, tomando a média + ou − duas vezes o desvio padrão.

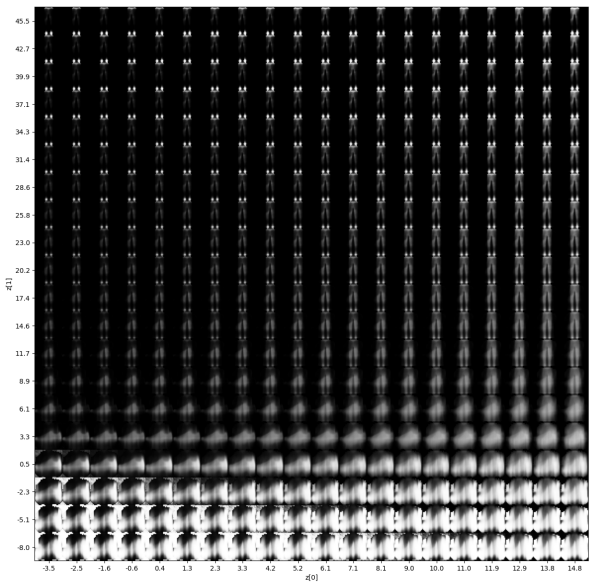

Figura 20: Reconstrução a partir do AE, no caso observando um espaço linear com um range que representa 95% dos dados. É possível ver botas e calças.

6 Comparação dos resultados das três arquiteturas

6.1 Síntese a partir dos conjuntos de teste e treino

No AE, considerando o espaço latente de tamanho 64, podemos ver que as imagens sintetizadas no conjunto de treino, são tão borradas quanto em relação ao conjunto de teste. Isso porque a loss avaliada no teste é 0.00891 e no treino 0.00875, ou seja, a de teste é só um pouco maior que a de treino. Lembrando que no AE é descobrir F e G tal que $F(G(x)) = x$, logo a loss pretende minimizar $||F(G(x)) - x||$. A loss a se minimizar no test seria $L(G(F(x_{test})), x_{test})$ e a do treino é definida de forma análoga.

No VAE, considerando o espaço latente de tamanho 2, podemos notar que as imagens sintetizadas no conjunto de teste parecem ser menos borradas quanto às de treino. Isso ocorre porque a reconstruction loss de test 0.35783845 é menor que a de treino 0.30580848, como computamos após a síntese dos dados.

6.2 Síntese estocástica

A síntese estocástica acontece de forma similar entre as redes treinadas, pois amostramos um ruído ou representação latente do dado, que é posteriormente passada por uma rede gerado ou por ou *decoder* para ser transformada em uma imagem.

Nas GANs, os resultados conseguem ser mais ou menos satisfatórios, mas vai depender bastante da distribuição e dimensão do ruído escolhidas no treinamento. No entanto, uma vez treinada, é bem trivial gerar novos dados a partir do ruído, o que pode não ser verdade para os AEs.

Os AEs sãe redes que aprendem uma transformação para um certo espaço latente (o *encoder*) e uma transformação do espaço latente para o espaço dos dados (o *decoder*). Por causa disso, podemos nos aproveitar da estrutura do *decoder* para gerar novas imagens a partir do espaço latente. No entanto, essa tarefa acaba não sendo tão trivial, pois teríamos que, de alguma forma, amostrar de alguma distribuição que representa os dados no espaço latente. A forma encontrada foi de gerar novas imagens a partir de dados ao redor da média.

Esse último problema é resolvido nos VAEs, já que temos uma distribuição para amostrar dados no espaço latente, alcançando melhores resultados em comparação aos AEs treinados.

7 Prós e Contras das Arquiteturas

Neste assignment, vimos a implementação da arquitetura de três modelos generativos distintos de aprendizagem não supervisionada.

O Autoencoder Básico (AE) é um modelo simples que aprende a representação dos dados, comprimindo a informação em um espaço latente e depois reconstruindo os dados a partir dele. Consideramos simples de implementar e rápido para treinar, mas o AE pode perder detalhes complexos das imagens devido à sua simplicidade.

O Variational Autoencoder (VAE) é uma amplicação do autoencoder básico. O VAE que não apenas comprime e reconstrói dados, mas também aprende a distribuição dos dados no espaço latente. Isso proporciona interpretabilidade e habilidade para lidar com variação nos dados. Neste caso uma vantagem é poder ter um função objetivo e uma loss clara. No entanto, a introdução de conceitos estatísticos, como o truque da reparametrização, adiciona complexidade ao modelo. Além disso, há um ruído injetado, tornando reconstrução imperfeita, de forma que o resultado é desfocado em comparação com GAN.

A Generative Adversarial Network (GAN) consiste em duas redes neurais, um gerador e um discriminador, que competem entre si. GANs são conhecidas por produzir resultados muito bons visualmente,

como vimos neste assignment, funcionando muito bem até com imagegns borradas, mas seu treinamento é desafiador, pois não tem uma loss ou uma função objetivo claras para comparar, e pode ser instável.

8 Conclusão

No geral, os três métodos possuem capacidades generativas satisfatórias para esse conjunto de dados. Porém, mais coisas poderiam ter sido feitas para melhorar os resultados. Por exemplo, concatenar a classificação de cada imagem à entrada na forma de *one hot encoding* para poder controlar o tipo de imagem gerada. Além disso, poderíamos ter estimado uma distribuição para os dados no espaço latente dos AEs para obtermos melhores resultados na hora de gerar novas imagens a partir do ruído.

Referências

[1] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 10 de jun. de 2014. arXiv: 1406.2661[cs,stat]. URL: http://arxiv.org/abs/1406.2661 (acesso em 31/10/2023).

[2] Keras. *The Model class*. URL: https://keras.io/api/models/model/.

[3] Diederik P Kingma e Max Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: 1312.6114[stat.ML].

[4] Keras. *Variational AutoEncoder*. URL: https://keras.io/examples/generative/vae/.