

# Relatório Deep Learning

## Assignment 4: GANs+Autoencoders

Ana Carolina Erthal & Felipe Lamarca

[Link para o Google Colab](#)

### 1 Modificações no código

A partir da classe já definida para o Autoencoder, criamos uma nova função, `BasicAutoencoderPipeline`, que para uma dimensão do espaço latente, realiza o treinamento do autoencoder e a síntese de dados para imagens de treino, teste e a partir de ruído aleatório (síntese estocástica).

Definimos, também, a classe VAE, que constrói o variational autoencoder, com base principalmente nas implementações do [keras](#). Na prática, o modelo consiste em um encoder-decoder, mas considerando que o encoder, ao invés de utilizar diretamente as imagens inseridas, cria uma distribuição a partir delas e utiliza uma amostra dessa distribuição. O VAE também utiliza a divergência KL para regularização, e implementamos esse termo de regularização na loss. Novamente, realizamos a reconstrução utilizando dados de treino, teste, e estocásticos, e também diferentes dimensões para o espaço latente.

Geramos, para ambas as distribuições, plots das matrizes de reconstrução a partir do espaço latente de tamanho 2, mas apresentaremos o resultado, junto com as imagens reconstruídas em cada modelo e variação de hiperparâmetro mais adiante.

Por fim, construímos a GAN, definindo um generator e um discriminator, e também um modelo que sequencia as duas etapas, realizando o pipeline completo do modelo generativo. Treinamos a rede para que o generator aprenda a gerar imagens parecidas com as de input, e para que o discriminator aprenda a distinguir as falsas das reais. Por fim, plotamos as imagens produzidas pela epoch mais recente do modelo, e também um gráfico apresentando a curva de aprendizado.

## 2 Hiperparâmetros

Para todos os resultados apresentados abaixo, utilizamos os seguintes hiperparâmetros:

Hiperparâmetro	Valor
espaços latentes utilizados	AE e VAE: 2, 32 e 64
# epochs	AE e VAE: 10    GAN: 30
optimizer	AE, VAE, GAN: 'adam'
batch size	GAN: 800
GAN distributions	normal, uniform, weibull

Tabela 1 – Hiperparâmetros

## 3 Resultados

Realizamos testes para o Autoencoder e Variational Autoencoder sintetizando imagens a partir de dados de treino, teste e dados de ruído aleatório, utilizando diferentes tamanhos de espaço latente.

### 3.1 Autoencoder (Simples)

Veja abaixo, de início, as sínteses realizadas utilizando o autoencoder padrão e um espaço latente de tamanho 64. É interessante observar que, nessas configurações, não há diferença expressiva na qualidade das reconstruções ao compararmos a síntese a partir do conjunto de treino e de teste. Isto provavelmente se deve a uma boa escolha de tamanho do espaço latente, representando um bom meio termo na dualidade entre complexidade e generalização. Isto é, temos certa complexidade, já que conseguimos sintetizar características mais detalhadas das imagens, mas mantemos generalização uma vez que a síntese de teste tem bom resultado.

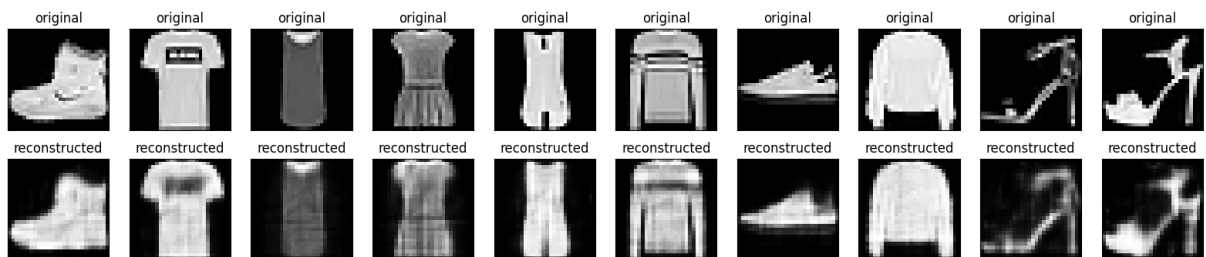


Figura 1 – Autoencoder com síntese de treino e espaço latente de tamanho 64

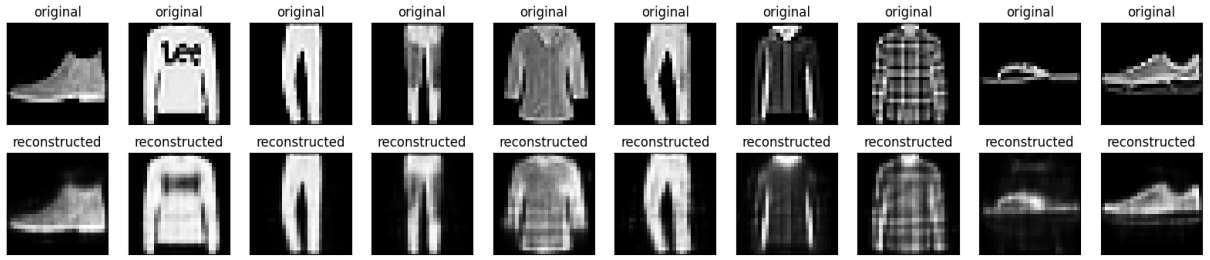


Figura 2 – Autoencoder com síntese de teste e espaço latente de tamanho 64

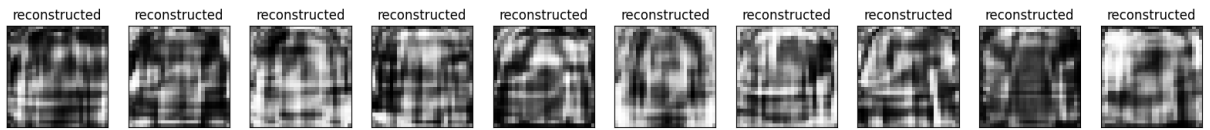


Figura 3 – Autoencoder com síntese estocástica e espaço latente de tamanho 64

Se tratando da síntese estocástica, é possível observar a pouca variabilidade das imagens produzidas, e principalmente qualidade (a imagem em si não é muito informativa). Isso de fato representa o quanto o modelo está se ajustando aos dados de treinamento (overfitting), e nos evidencia a necessidade de utilizar um modelo que traga mais variabilidade aos dados, justificando a importância do Variational Autoencoder.

Quanto à dimensão em si, utilizar um tamanho 64 parece ser bastante efetivo para reconstrução de imagens. Tendo uma dimensão grande, mais características das imagens são levadas em conta para depois serem reconstruídas, melhorando o resultado.

Vamos analisar os resultados obtidos na síntese utilizando espaço latente de tamanho 32, ainda tratando dos Autoencoders simples:

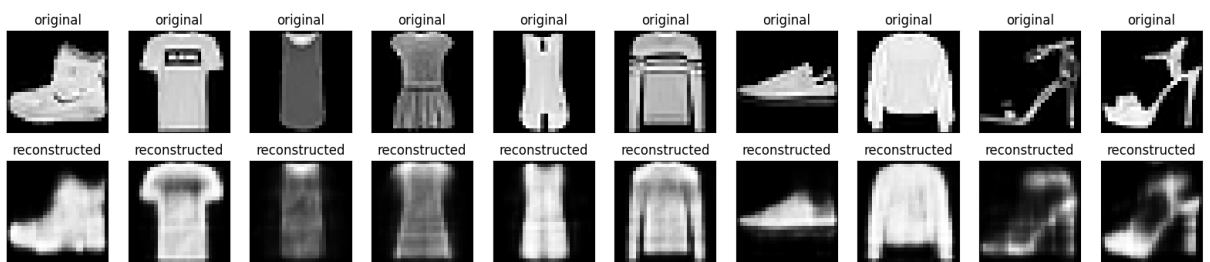


Figura 4 – Autoencoder com síntese de treino e espaço latente de tamanho 32

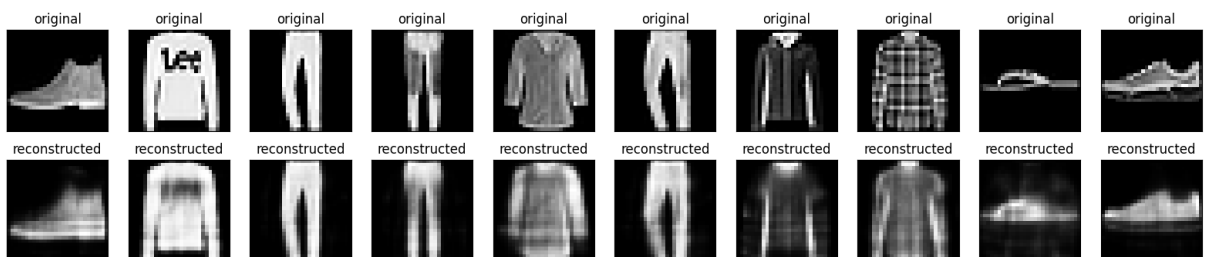


Figura 5 – Autoencoder com síntese de teste e espaço latente de tamanho 32

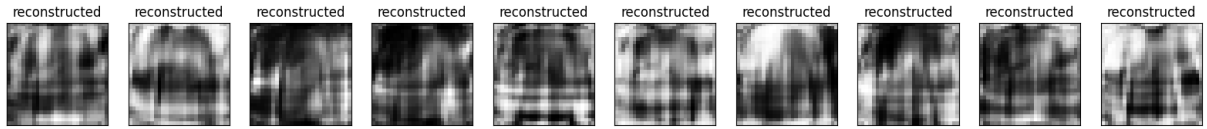


Figura 6 – Autoencoder com síntese estocástica e espaço latente de tamanho 32

As diferenças entre sínteses para treino, teste se mantém em relação às destacadas anteriormente: não há benefício claro, observando as imagens reproduzidas, entre as sintetizadas a partir de treino e teste.

No entanto, ao comparar os resultados entre o tamanho 32 e 64, evidencia-se uma performance pior da síntese realizada com um espaço latente de tamanho 32. Isto é, as imagens produzidas são piores, menos definidas quando comparadas às de `latent_dim=64`. A síntese estocástica, por sua vez, se mantém com pouquíssima variabilidade.

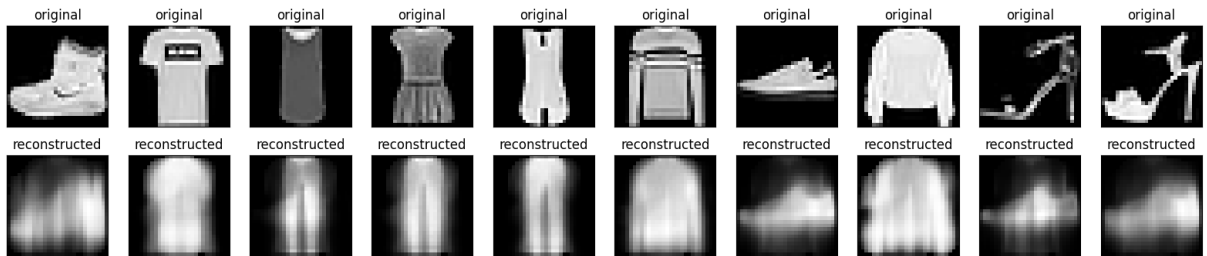


Figura 7 – Autoencoder com síntese de treino e espaço latente de tamanho 2

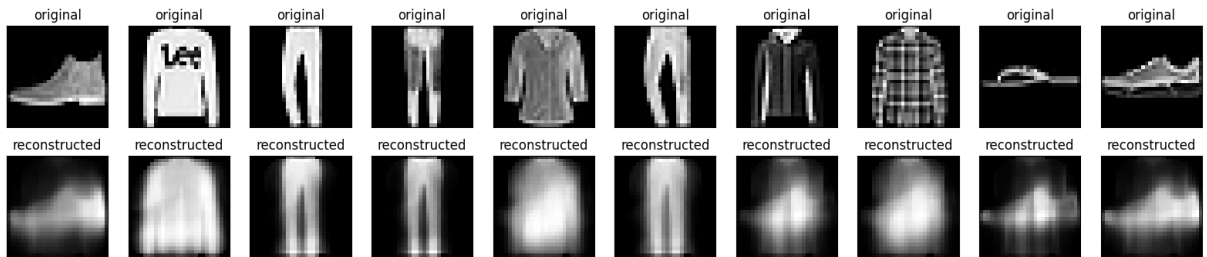


Figura 8 – Autoencoder com síntese de teste e espaço latente de tamanho 2

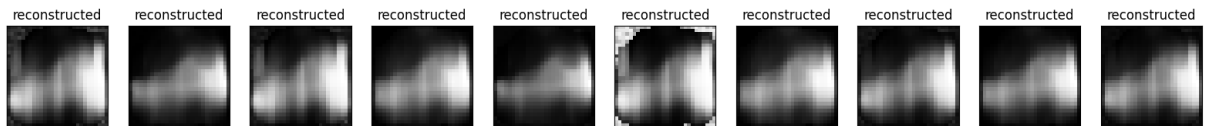


Figura 9 – Autoencoder com síntese estocástica e espaço latente de tamanho 2

Para a dimensão 2 no espaço latente é importante observarmos como as imagens reconstruídas se tornam ruins. A reconstrução não é efetiva nem no conjunto de treino, representando como a compressão não conseguiu capturar informações suficientes do conjunto de entrada do encoder, isto é, não conseguimos, com um espaço latente de tamanho 2, realizar uma compressão efetiva da imagem. Observe que para a síntese estocástica

produzimos uma imagem mais interpretável que as anteriores, já que é possível distinguir um calçado, mas quase não há variabilidade! É possível observar melhor esse fenômeno através da matriz de reconstrução a partir do espaço latente de tamanho 2:

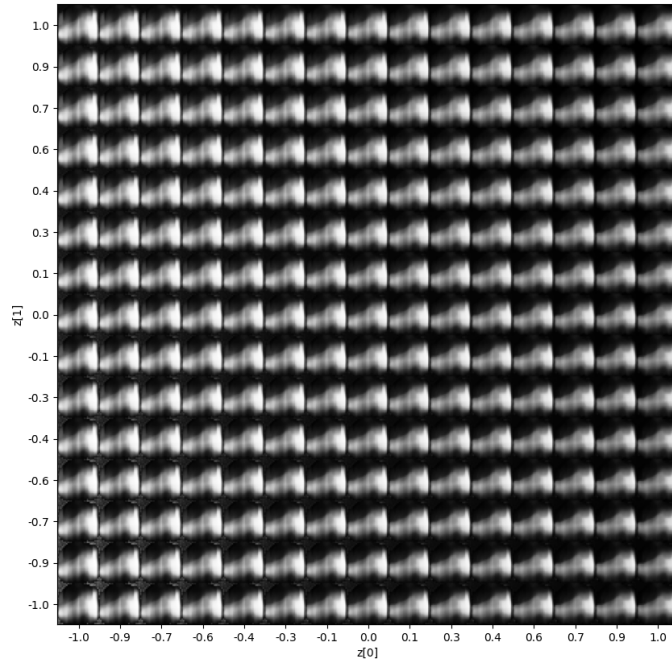


Figura 10 – Autoencoder - Matriz de reconstrução espaço latente (dim=2)

Note que obtemos sempre a mesma reconstrução, com variabilidade mínima. Essa imagem exemplifica uma questão relevante do Autoencoder, que destacamos anteriormente: a falta de variabilidade para reconstrução. Essa situação é explicada pela limitação desse modelo na geração de dados: conseguimos produzir apenas imagens muito próximas às vistas no treinamento, já que o modelo é apropriado apenas para a reconstrução (não para geração de novas imagens).

### 3.2 Variational Autoencoder (VAE)

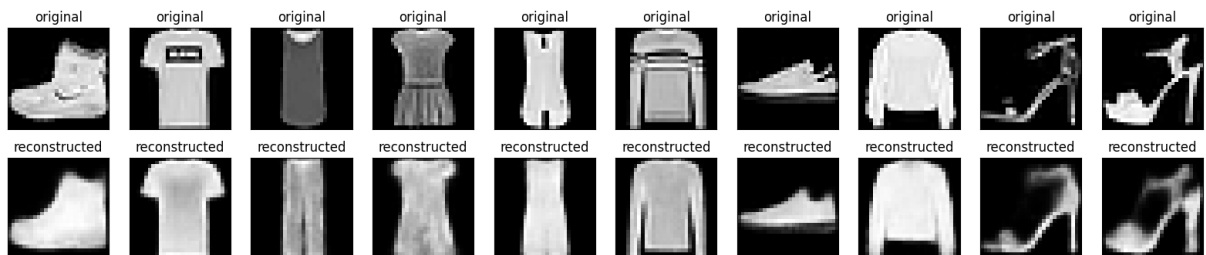


Figura 11 – Variational Autoencoder com síntese de treino e espaço latente de tamanho 64

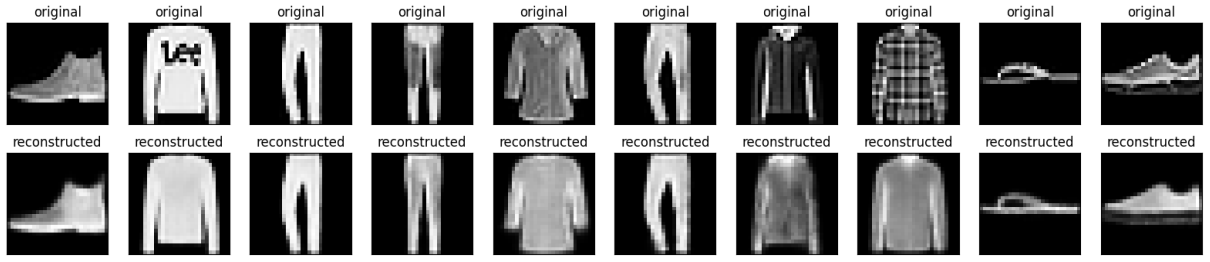


Figura 12 – Variational Autoencoder com síntese de teste e espaço latente de tamanho 64

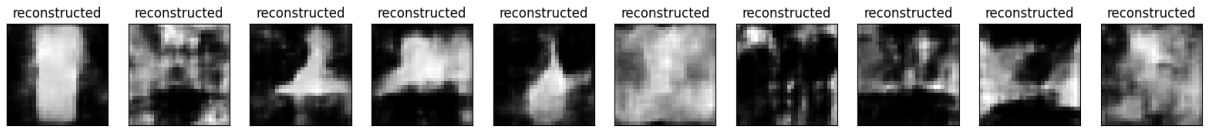


Figura 13 – Variational Autoencoder com síntese estocástica e espaço latente de tamanho 64

Ao utilizarmos o Variational Autoencoder definindo 64 como a dimensão do espaço latente para gerar novas imagens, observamos que as imagens reconstruídas captam características mais gerais das imagens — por exemplo, determinados traços no sapato ou a escritura na blusa não são reconstruídas —, e esse fato se deve à maior variabilidade determinada pelo modelo para a reconstrução, evitando a produção de imagens idênticas às originais. É importante levarmos em conta que, embora a variabilidade seja bastante desejada para geração de novos dados, quando nossa tarefa é simplesmente reconstruir imagens, o Autoencoder simples parece performar melhor.

Comparando a síntese nos conjuntos de treino e teste, novamente não observamos mudanças substantivas nas imagens reconstruídas, de maneira que o padrão observado nos dois casos é o mesmo. Para essa dimensão de espaço latente, a síntese estocástica ainda não permite interpretações.

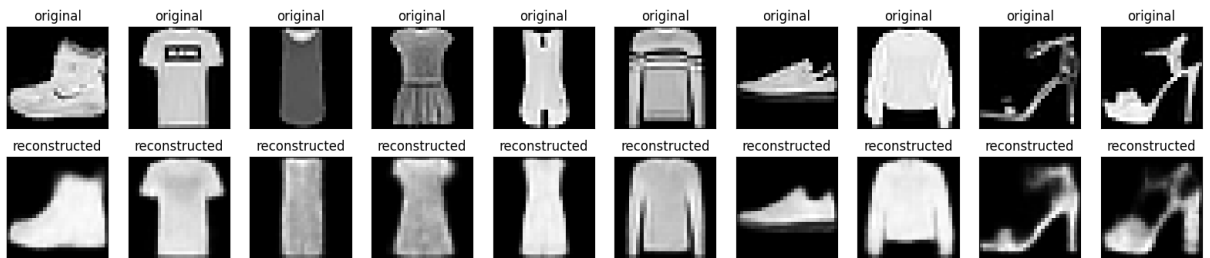


Figura 14 – Variational Autoencoder com síntese de treino e espaço latente de tamanho 32

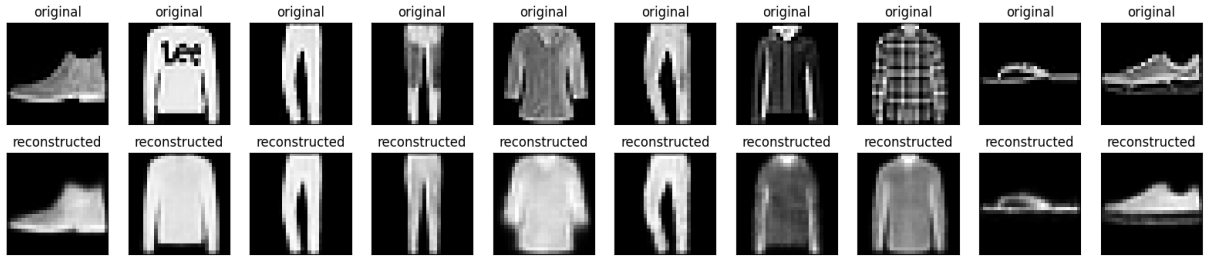


Figura 15 – Variational Autoencoder com síntese de teste e espaço latente de tamanho 32

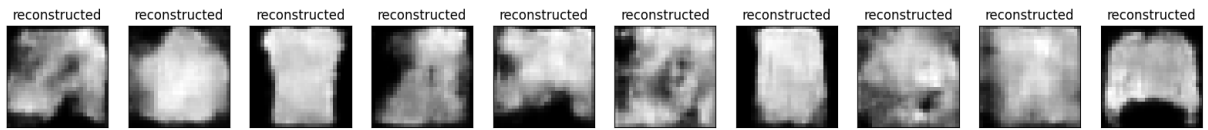


Figura 16 – Variational Autoencoder com síntese estocástica e espaço latente de tamanho 32

Alterando a dimensão do espaço latente para 32, novamente temos uma redução na exatidão das imagens produzidas, dessa vez em maior escala, já que também adicionamos variação. A grande diferença está na síntese estocástica, que começa a reconstruir imagens que se assemelham de alguma forma a peças de roupa. De fato, esse resultado faz sentido na medida em que, definindo dimensões menores para o espaço latente, forçamos que o modelo aprenda apenas as características mais críticas dos dados.

Vejamos, por fim, para o espaço latente de tamanho 2:

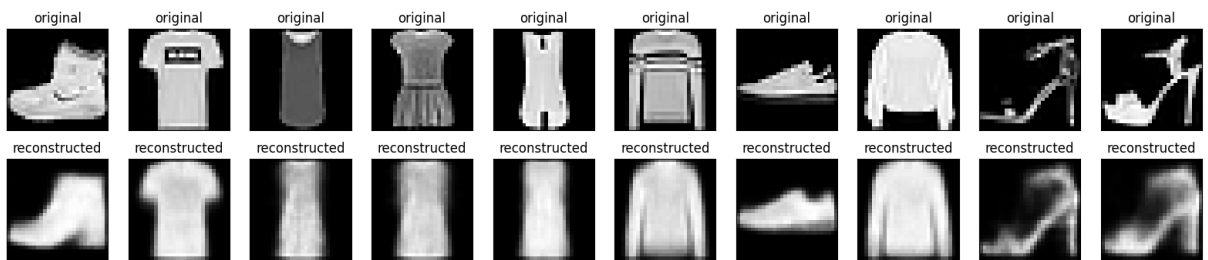


Figura 17 – Variational Autoencoder com síntese de treino e espaço latente de tamanho 2

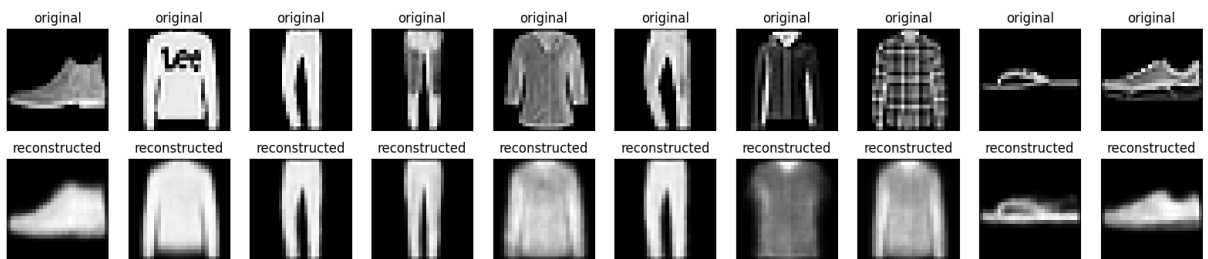


Figura 18 – Variational Autoencoder com síntese de teste e espaço latente de tamanho 2

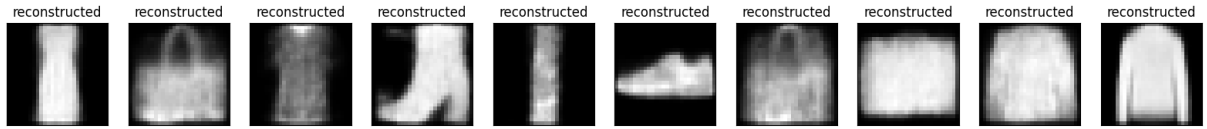


Figura 19 – Variational Autoencoder com síntese estocástica e espaço latente de tamanho 2

Observe que para o espaço latente com dimensão 2, o resultado mais interessante é o da síntese de dados estocásticos. Produzimos, por fim, imagens muito interpretáveis (ainda que pouco nítidas, já que partimos de dados aleatórios) e bastante diferenciadas. Portanto, assim como ocorreu no Autoencoder, a melhor síntese estocástica se apresenta quando temos um espaço latente de dimensão baixa.

Como nesse modelo também adicionamos variabilidade, conseguimos produzir, a partir da síntese de dados estocásticos, um conjunto mais variado, com interpretabilidade mais clara. Esse fato se evidencia claramente quando visualizamos a matriz de reconstrução do espaço latente:

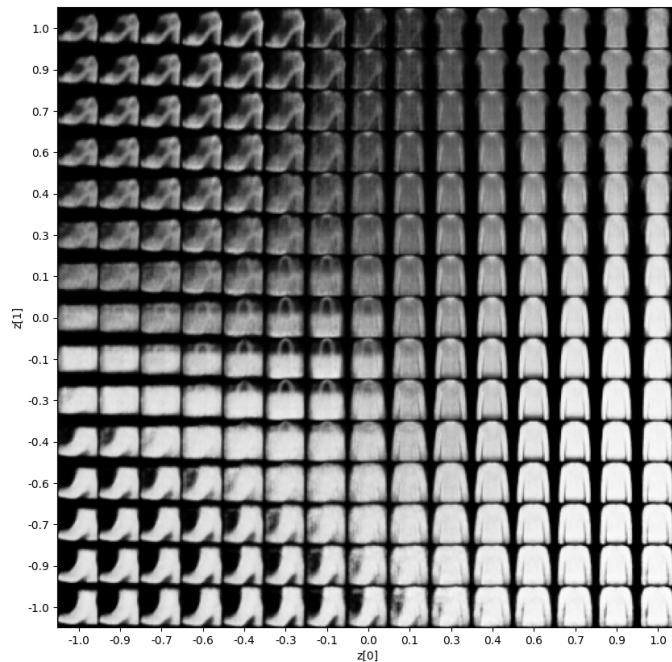


Figura 20 – Variational Autoencoder - Matriz de reconstrução espaço latente (dim=2)

Observamos uma situação muito diferente do Autoencoder, já que agora que adicionamos variabilidade, nosso espaço latente (para dimensão 2, isto é, bem baixa) passa a variar entre diferentes peças de roupas, mostrando a diferença da compressão das características mais relevantes quando inserimos variabilidade nos dados (imagem 20) e quando utilizamos o Autoencoder padrão (imagem 10).



## 4 Comparação das arquiteturas (AE e VAE)

Apesar de a princípio servirem ao mesmo propósito, é bom atentar-se a diferenças conceituais e estruturais dos AEs e VAEs, que tornam suas performances bastante distintas.

Como os autoencoders simples não contam com adição de variabilidade (como ocorre com os Variacionais através da amostragem e regularização pela divergência KL), a reconstrução de imagens, em geral, se torna bastante próxima às originais. Assim sendo, quando tivermos tarefas desse tipo, utilizar o modelo mais simples é uma boa opção — apesar da necessidade de atentar-se ao overfitting, que pode ocorrer facilmente, dificultando a reconstrução para dados de treino. No entanto, essa falta de variabilidade torna o modelo impróprio para geração de novos dados (como vimos na imagem 10).

Portanto, quando há interesse em gerar novas imagens, utilizar o VAE é uma estratégia mais adequada, pois como vimos, VAEs permitem a geração de dados latentes, explorando diferentes regiões do espaço a fim de criar amostras diversificadas.

Quando tratamos de geração de dados, no entanto, é importante explorarmos as GANs.

## 5 Generative Adversarial Networks (GANs)

Discutiremos, por fim, o modelo especializado em geração de imagens. Realizamos o treinamento utilizando três distribuições de probabilidade.

Observe abaixo as imagens iniciais e as obtidas após 30 epochs utilizando a distribuição normal, além das curvas de aprendizado do generator e do discriminator de nossa GAN:

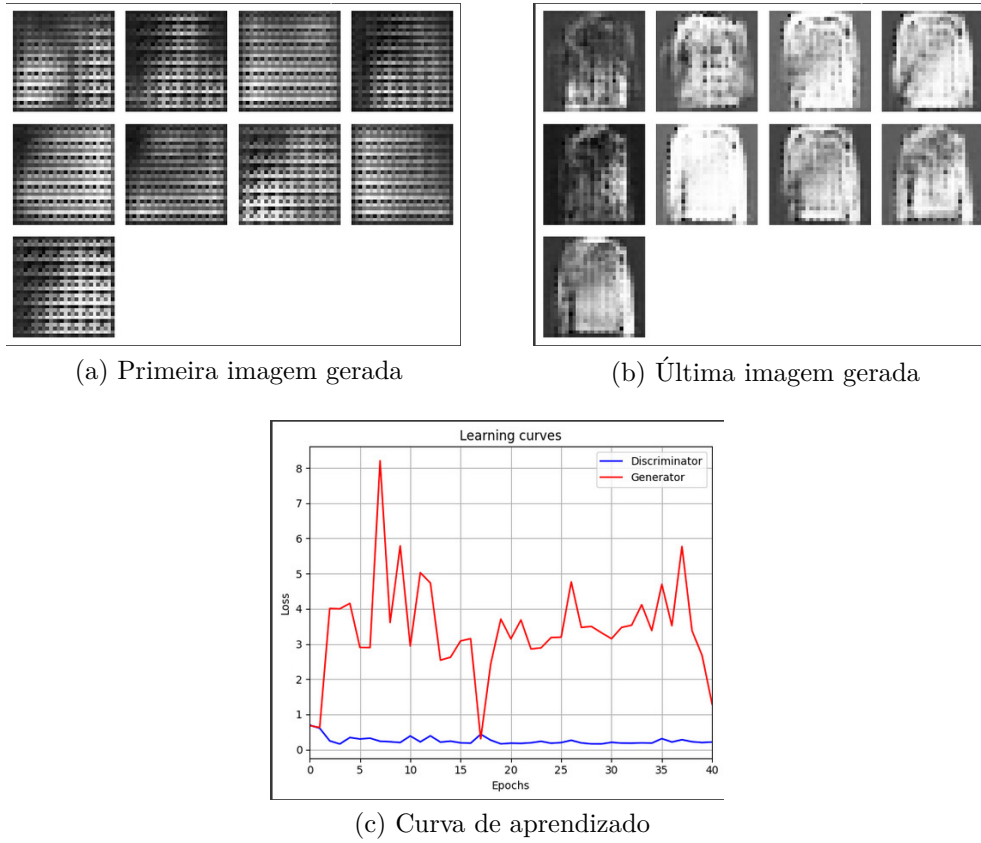
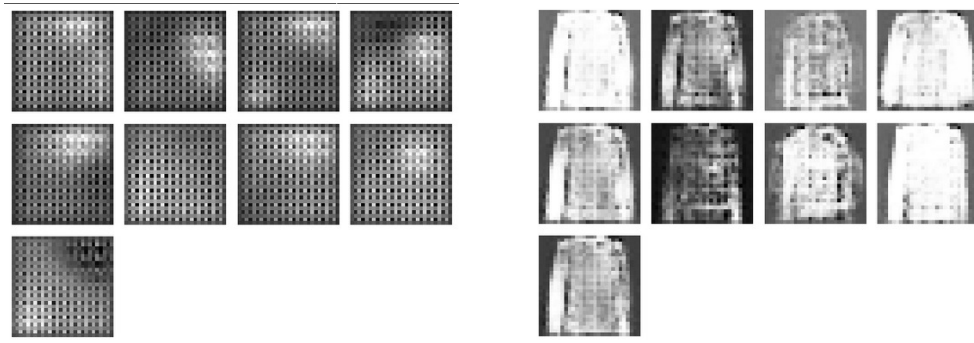


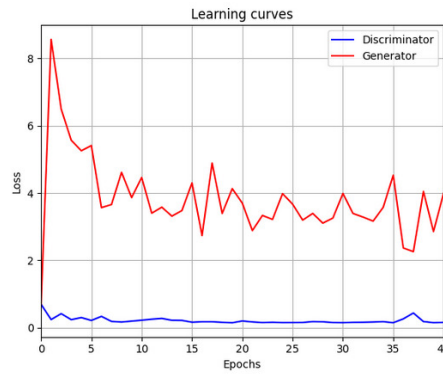
Figura 21 – GAN com amostragem por distribuição  $\mathcal{N}(0, 1)$

É interessante observar que, em poucas epochs, obtivemos resultados muito bons, e a peça de roupa que está sendo gerada já é muito evidente. Esse modelo é conhecido por sua eficácia em produzir imagens de muita qualidade, frequentemente indistinguíveis do conjunto real, já que sua arquitetura leva justamente a essa situação (generator tenta "enganar" o discriminator). É evidente que em nossos experimentos a imagem produzida não tem essa qualidade, pois a GAN conta com um processamento de alto custo computacional e que em geral leva bastante tempo, mas mesmo com as limitações, as imagens obtidas são bastante satisfatórias.



(a) Primeira imagem gerada

(b) Última imagem gerada



(c) Curva de aprendizado

Figura 22 – GAN com amostragem por distribuição  $\mathcal{U}(0, 1)$

Para a distribuição uniforme também obtivemos resultados visualmente muito bons, e observando a curva de aprendizado, é possível notar que o generator aprendeu de forma significativa (mais explícita que para as outras distribuições).

Por fim, utilizando a distribuição Weibull (abaixo), obtivemos a pior geração dentre as GANs. As últimas imagens geradas ainda são muito confusas visualmente (apesar de a peça de roupa ser identificável) quando comparamos aos resultados anteriores.

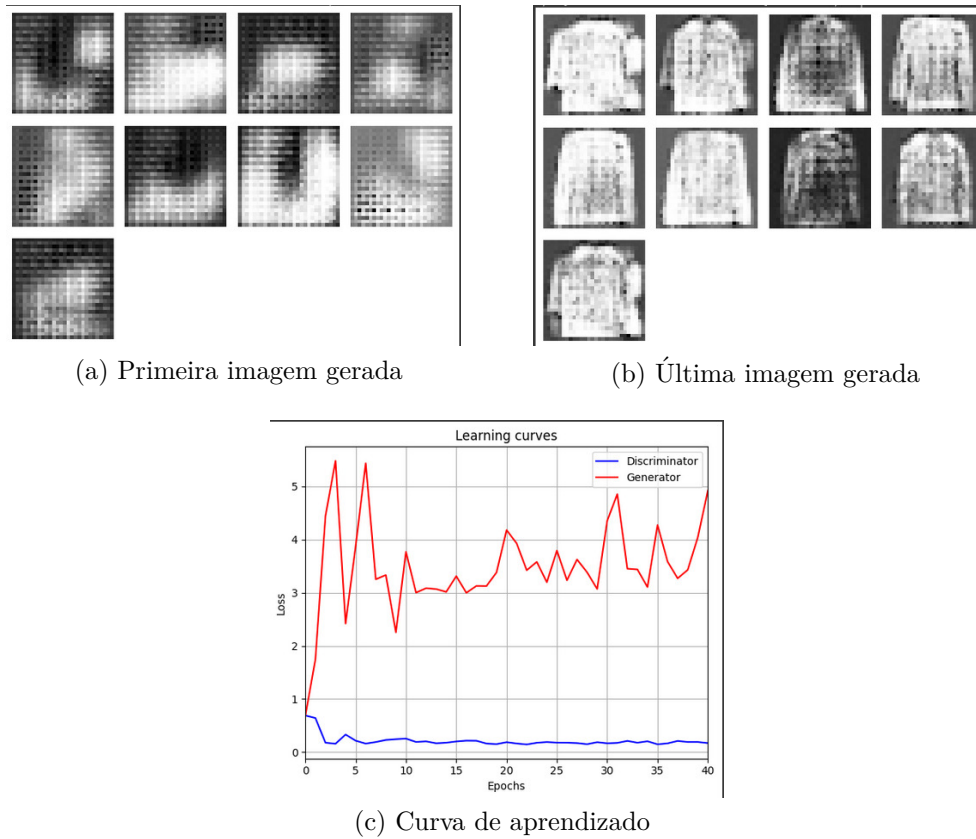


Figura 23 – GAN com amostragem por distribuição Weibull

De forma geral, a GAN, quando comparada aos modelos discutidos anteriormente, parece ser bastante eficaz. O Variational Autoencoder, apesar de realizar geração de dados, não performou com tantos detalhes quanto a GAN (especialmente ao tratarmos do que teria ocorrido ao utilizarmos a GAN com maior poder computacional).

No entanto, os outros modelos são certamente mais simples e menos custosos, o que pode justificar seu uso em muitos casos. Em especial, quando tratamos de tarefas de reconstrução de imagens, os modelos de Autoencoder reconstroem dados de entrada com fidelidade. Sendo assim, a melhor escolha de modelo depende bastante da tarefa que deseja-se performar.