

Relatório 27 - Prática: Modelos Generativos (III)

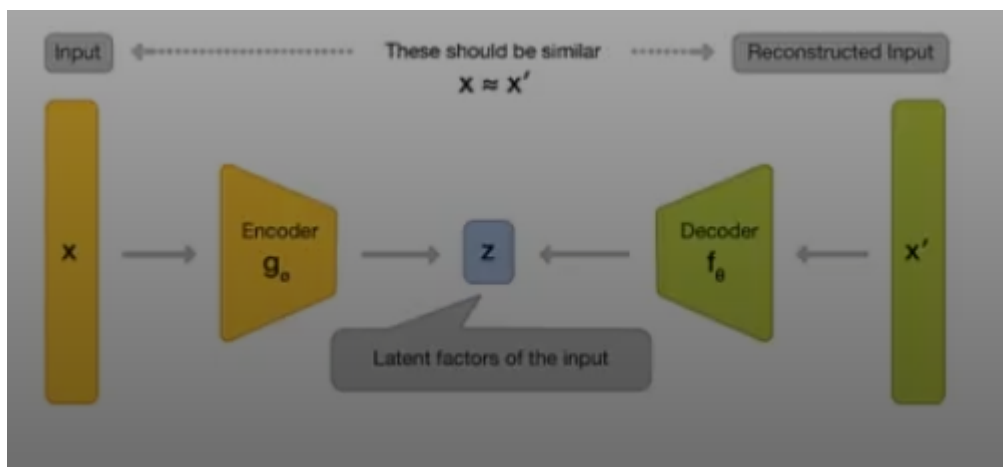
Higor Miller Grassi

Descrição da Atividade: Nesta prática sobre modelos generativos, foram explorados diferentes tipos de arquiteturas, como Auto-Encoders, Variational Auto-Encoders (VAEs) e Generative Adversarial Networks (GANs). Também foi analisado o modelo GPT, um gerador de texto baseado em aprendizado profundo, que utiliza um conjunto de obras de Shakespeare para aprender a prever a próxima palavra com base nas anteriores, criando textos no mesmo estilo.

Os modelos Generativos, basicamente “aprendem” com datasets e posteriormente criam algo sem mesmo ter existido. Como por exemplo, um modelo treinado com milhões de fotos de gatos pode gerar uma imagem nova de um gato que nunca existiu.

Auto-Encoders: Eles aprendem a reduzir a entrada para suas características latentes, como mostrado na imagem a seguir, a parte esquerda da mesma, pega uma entrada(geralmente uma imagem) e com as CNNs para destilar os padrões de entrada.

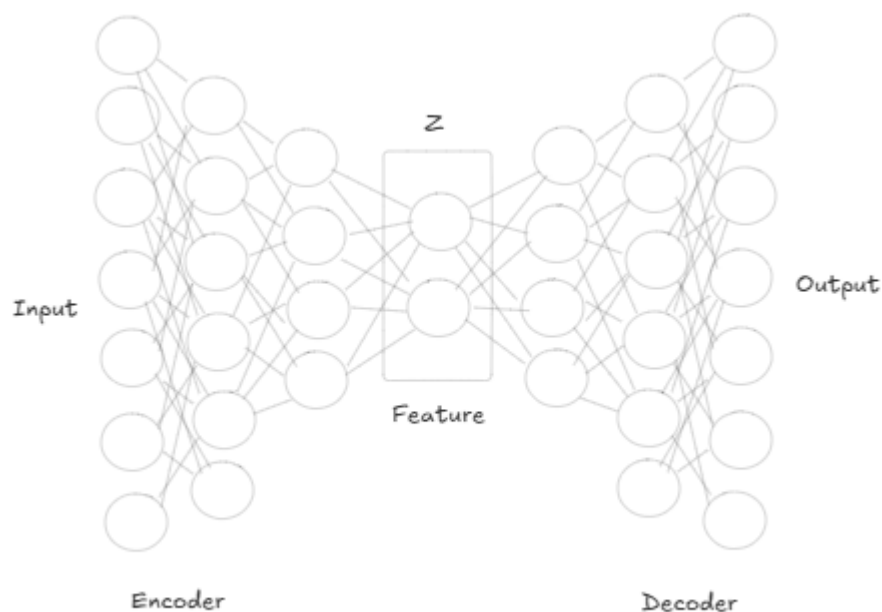
Do outro lado da imagem temos o Decoder, que basicamente seria o inverso do Encoder, sendo treinado para reconstruir imagens e/ou dados completos a partir desses mesmos vetores pertencentes na caixa “Z”, com convolução de transposição juntamente ao Max Pooling, diferentemente da convolução anteriormente ensinada. \



Estrutura dos Autoencoders

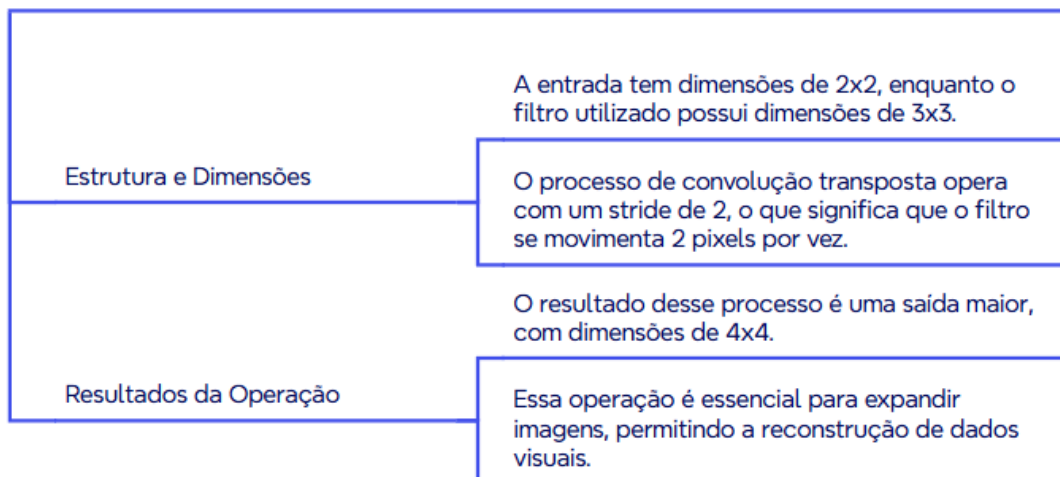
Função do Encoder	<p>O Encoder reduz a entrada para características latentes, geralmente usando Convolutional Neural Networks (CNNs).</p> <p>A parte esquerda da figura mostra como a entrada, como uma imagem, é processada para extrair padrões.</p>
Função do Decoder	<p>O Decoder inverte a função do Encoder, reconstruindo imagens a partir dos vetores latentes.</p> <p>Utiliza convolução de transposição e Max Pooling, diferentemente da convolução tradicional.</p>

Resumindo, o objetivo geral é que a entrada e a entrada reconstruída pelo decoder sejam o mais semelhante possível, sendo muito usado para: redução de dimensionalidade, detecção de anomalias, compressão de dados, entre outras muitas aplicações no contexto técnico, representado na figura a seguir.

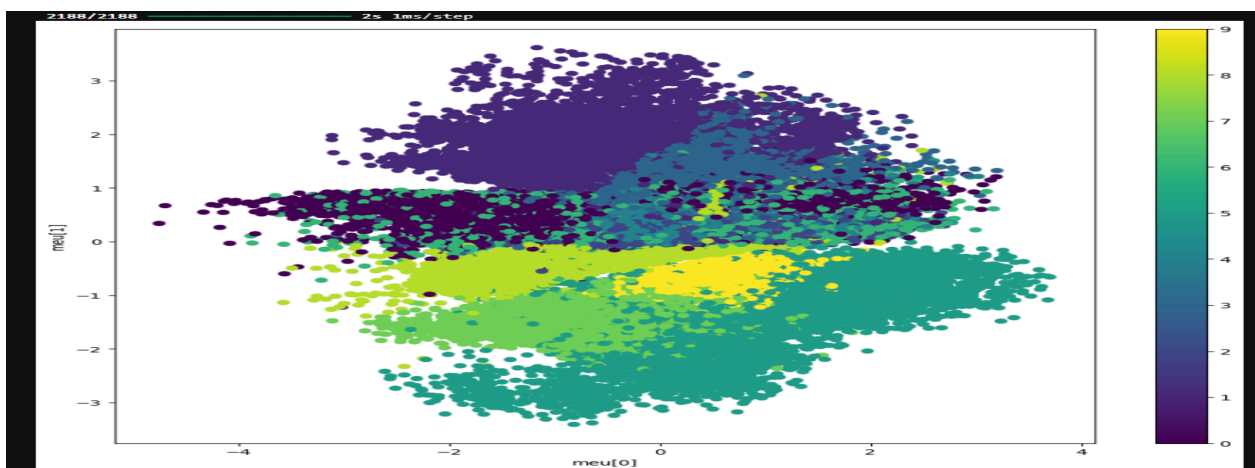


Convolução Transposta: Observando a imagem, ela funciona da seguinte maneira, com uma entrada (2x2) e um peso (3x3) aplicado sobre a mesma, e no Stride 2, o filtro é movimentado com um passo de 2 pixels, resultando em um output maior de (4x4). Em um resumo, reconstrói uma imagem maior a partir de uma menor, sendo útil para expandir dados visuais, criando ou melhorando as imagens.

Funcionamento da Convolução Transposta



Variational Auto-Encoders(VAEs): Sendo a base das redes generativas, que aprende a representar dados de forma comprimida (codificação) e depois os reconstrói (decodificação), gerando novas amostras semelhantes às do conjunto de treinamento. Estando diretamente ligados a distribuições de probabilidades porque, ao invés de mapear os dados para um ponto fixo no espaço latente (como os autoencoders tradicionais), eles aprendem uma distribuição estatística dos dados, permitindo que o modelo gere novas amostras ao escolher diferentes pontos dessa distribuição, assim como na imagem a seguir.

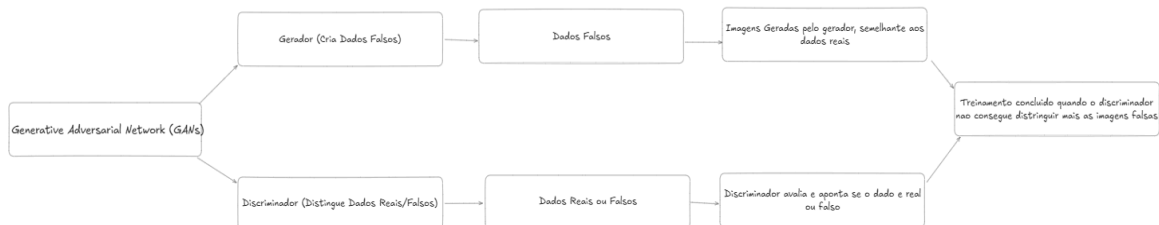
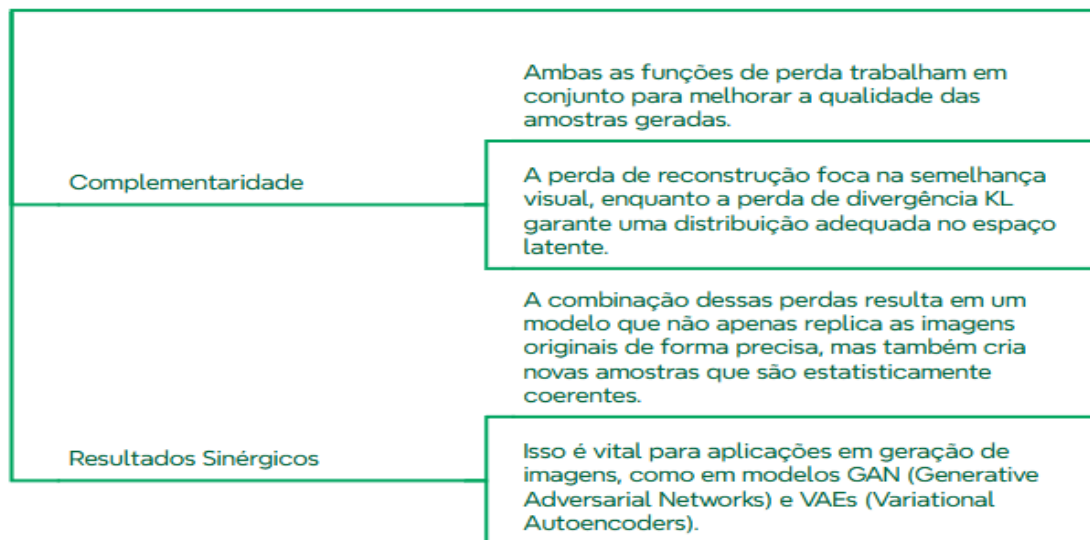


Loss Functions:

- **Reconstruction loss:** Penalizar as imagens que não são semelhantes as imagens originais.

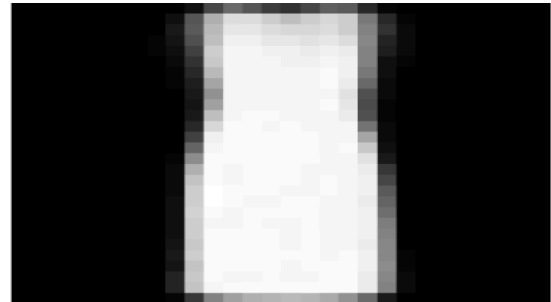
- **Divergence Loss KL:** Mede a diferença entre duas distribuições de probabilidade, forçando a distribuição latente (z) a se aproximar de uma normal padrão, assim conseguindo regularizar o espaço latente para gerar amostras mais realistas.

Relação entre as Funções de Perda



Generative Adversarial Networks(GAN's): São modelos de aprendizado de máquina compostos por duas redes neurais: o gerador, que cria dados falsos, e o discriminador, que tenta distinguir os dados reais dos falsos. O objetivo do gerador é enganar o discriminador, e o discriminador visa melhorar sua capacidade de identificar dados falsos, com eles treinando juntos de forma competitiva, com o gerador melhorando ao longo do tempo para criar dados cada vez mais realistas, e quando o discriminador não consegue mais diferenciar, o treinamento acabou, onde

na imagem a seguir temos do lado direito a foto em que está no database e no esquerdo a imagem criada pelo algoritmo onde tem uma semelhança.



O vídeo "Let's build GPT: from scratch, in code, spelled out", apresentado por Andrej Karpathy, demonstra a construção passo a passo de um modelo GPT (Generative Pre-trained Transformer) a partir do zero. Ele inicia explicando os fundamentos da arquitetura de transformers, destacando conceitos-chave como o mecanismo de self-attention, que permite ao modelo analisar relações entre palavras em um texto, e a estrutura de blocos sequenciais com camadas de atenção multi-head e redes neurais feed-forward.

No desenvolvimento prático, o vídeo aborda desde o pré-processamento de dados (como tokenização de texto) até a implementação detalhada de cada camada do modelo, codificado em tempo real funções essenciais, como a geração de embeddings, cálculos de atenção, e a otimização do treinamento com técnicas como masking (para evitar que o modelo "veja" tokens futuros durante a geração), e também sendo explorado a lógica por trás da geração de texto autoregressiva, em que o modelo prevê a próxima palavra iterativamente, e demonstra como ajustar hiperparâmetros e lidar com desafios como eficiência computacional.

Fundamentos da Arquitetura de Transformers

Conceitos-Chave	O vídeo inicia com a explicação da arquitetura de transformers, enfatizando o mecanismo de self-attention.
	Este mecanismo permite ao modelo analisar as relações entre palavras em um texto, essencial para entender o contexto.
Estrutura de Blocos	A arquitetura é composta por blocos sequenciais que incluem camadas de atenção multi-head.
	As redes neurais feed-forward são integradas para processar as informações, oferecendo robustez ao modelo.

Por fim, o vídeo conclui com resultados práticos, treinando uma versão reduzida do GPT em um corpus de texto (como obras de Shakespeare) e gerando amostras de texto coerentes.

```
ROMEO:  
But you fret, he I wish his migute:  
They now I uDWARD IS TIVER:  
God I camil: and Gen  
The sitens, and say-wretor upwor alond, liege to makes.  
Ie plant, sure here than steed thus micks.
```

Conclusão: A prática demonstrou como os modelos generativos têm um grande potencial para diversas aplicações, desde a reconstrução e compressão de dados até a criação de novas amostras realistas, seja em imagens ou textos. A competição entre geradores e discriminadores nas GANs, a representação probabilística dos VAEs e a capacidade preditiva do GPT reforçam o impacto dessas arquiteturas no avanço da inteligência artificial.