

# Estudo Comparativo de *Autoencoders*

Exame de Qualificação de Mestrado

Candidato: Wilson Bagni Júnior  
Orientador: Prof. Dr. Zanoni Dias  
Coorientador: Prof. Dr. Hélio Pedrini

Instituto de Computação  
Unicamp

25 de Março de 2024

# Tópicos

1. Introdução
2. Objetivos e Contribuições
3. Questões de Pesquisa
4. Trabalhos Relacionados
5. Metodologia
6. Materiais
7. Resultados Preliminares
8. Plano de Trabalho e Cronograma

1. Introdução
2. Objetivos e Contribuições
3. Questões de Pesquisa
4. Trabalhos Relacionados
5. Metodologia
6. Materiais
7. Resultados Preliminares
8. Plano de Trabalho e Cronograma

# Autoencoders

- Redes neurais que utilizam técnica de aprendizado de máquina não supervisionado.
- Objetivo: prever na saída o mesmo dado da entrada.
- Composto por:
  - **Codificador:** realiza a compactação dos dados.
  - **Espaço Latente:** carrega a menor representação dos dados.
  - **Decodificador:** realiza a descompactação dos dados.

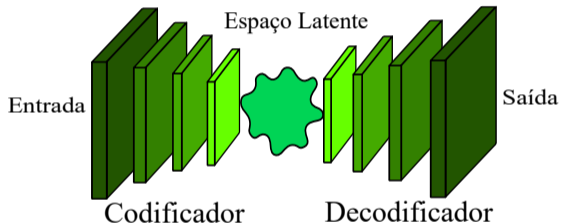


Figura: Exemplo de rede neural de um *Autoencoder*.

# Tipos de Modelos de *Autoencoders*

Existem diferentes tipos de modelos de *Autoencoders* com diferentes características e especificidades. Dentre eles, podemos citar:

- **Vanilla Autoencoder:** variação mais simples.
- **Fully Connected Autoencoder (FCAE)** <sup>[11]</sup>: composto por redes neurais totalmente conectadas.
- **Denosing Autoencoder (DAE)** <sup>[13]</sup>: realiza o treinamento com dados corrompidos.
- **Convolutional Autoencoder (CAE)** <sup>[15]</sup>: composto por redes neurais convolucionais.
- **Variational Autoencoder (VAE)** <sup>[6]</sup>: utiliza conceitos de inferência probabilística para otimizar o espaço latente.

# Tipos de Modelos de *Autoencoders*

- **Adversarial Autoencoder (AAE)** [8]: utiliza redes adversariais generativas para criar melhores representações latentes.
- **Recurrent Autoencoder (RAE)** [2]: composto por redes neurais recorrentes que lidam melhor com informações sequenciais.
- **Sparse Autoencoder (SAE)** [9]: utiliza esparsidade nas camadas de aprendizado da rede neural contribuindo para uma representação mais interpretável e econômica de recursos computacionais.
- **Contractive Autoencoder (ConAE)** [10]: utiliza “penalização por contrato” durante o treinamento.
- **Masked Autoencoder (MAE)** [5]: utiliza técnicas de mascaramento com mecanismos de auto-atenção.

# Autoencoders e suas Aplicações

São ferramentas utilizadas em várias aplicações no contexto de aprendizado de máquina, como, por exemplo:

- **Análise de imagens**
- **Classificação de imagens**
- **Compressão de dados**
- **Detecção de anomalias**
- **Extração de features**
- **Geração de dados sintéticos**
- **Processamento de texto**
- **Resolução de equações diferenciais**
- **Sistemas de recomendação**

1. Introdução
2. **Objetivos e Contribuições**
3. Questões de Pesquisa
4. Trabalhos Relacionados
5. Metodologia
6. Materiais
7. Resultados Preliminares
8. Plano de Trabalho e Cronograma



# Problema e Objetivos

**Problema:** Análise da eficácia, do espaço latente e da capacidade generativa de modelos de *Autoencoders* em diferentes conjuntos de dados.

## Objetivos:

- Revisão da literatura sobre o tema.
- Identificação e implementação dos modelos a serem estudados.
- Análise de desempenho, do espaço latente, do comportamento dos *Autoencoders* com dados distintos das bases de treinamento e de sua capacidade de gerar dados sintéticos.
- Complementação das análises aplicando *Autoencoders* em cenários específicos (por exemplo, em aumento de dados).
- Consolidação e comparação dos dados.

- Produção de um material consolidado sobre os *Autoencoders*, com estudo comparativo entre modelos, principais conceitos-chaves e aplicações.
- Publicação dos resultados obtidos.

1. Introdução
2. Objetivos e Contribuições
- 3. Questões de Pesquisa**
4. Trabalhos Relacionados
5. Metodologia
6. Materiais
7. Resultados Preliminares
8. Plano de Trabalho e Cronograma

# Questões de Pesquisa

Algumas perguntas servem como guia para as investigações realizadas:

- Quais modelos de *Autoencoders* possuem propriedades interessantes para serem estudados?
- Quais são os pontos fortes e fracos de cada modelo analisado?
- Há padrões que permitam fazer previsões ou estimativas de desempenho ou de outra propriedade de determinado modelo?
- Como é o desempenho dos modelos estudados em relação à geração de dados sintéticos?
- O quão adaptáveis são os modelos para lidar com dados diferentes daqueles utilizados no treinamento?

1. Introdução
2. Objetivos e Contribuições
3. Questões de Pesquisa
- 4. Trabalhos Relacionados**
5. Metodologia
6. Materiais
7. Resultados Preliminares
8. Plano de Trabalho e Cronograma

# Trabalhos Relacionados

Embora seja comum encontrar na literatura diversas aplicações de *Autoencoders*, não há muito material dedicado a estudar essa técnica por si ou comparar suas variantes.

Há trabalhos [1,16] em formato de *survey* com explicações sobre o funcionamento e principais características de diferentes modelos de *Autoencoders*, porém, não um estudo sistemático e comparativo entre eles.

Outros trabalhos [7,12,17] realizaram comparações em contextos específicos, como, por exemplo, *deep learning* e sistemas de recomendação.

1. Introdução
2. Objetivos e Contribuições
3. Questões de Pesquisa
4. Trabalhos Relacionados
- 5. Metodologia**
6. Materiais
7. Resultados Preliminares
8. Plano de Trabalho e Cronograma

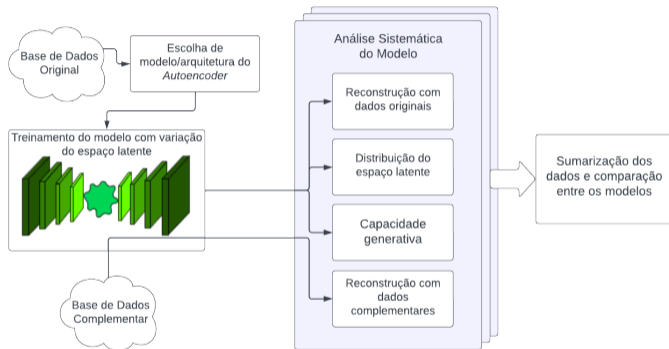
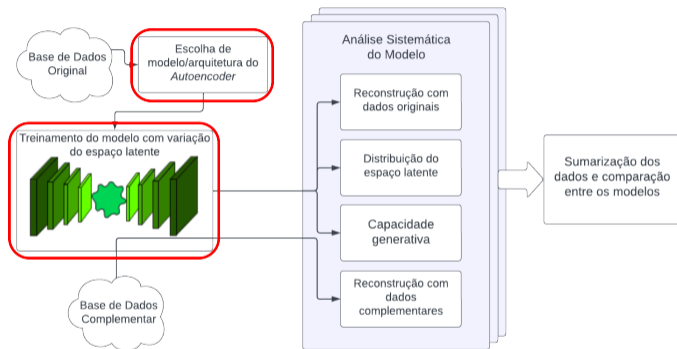


Figura: Etapas que compõem a metodologia utilizada neste projeto.



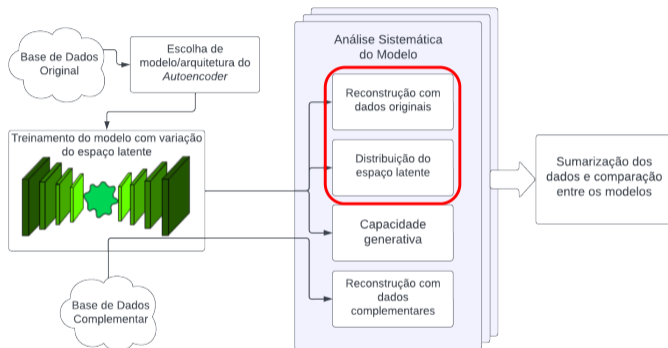
# Metodologia



1. Implementação do modelo.
2. Definição da arquitetura.

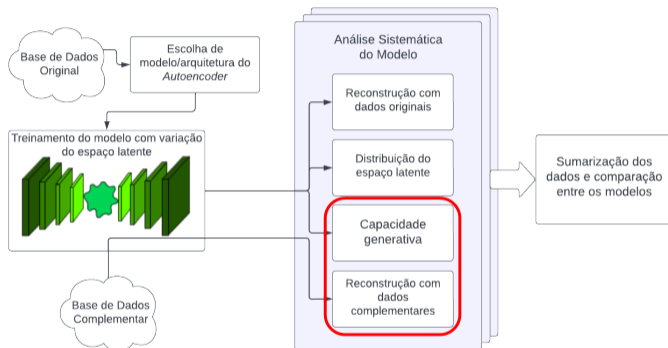
3. Treinamento do modelo com variação do espaço latente.

# Metodologia

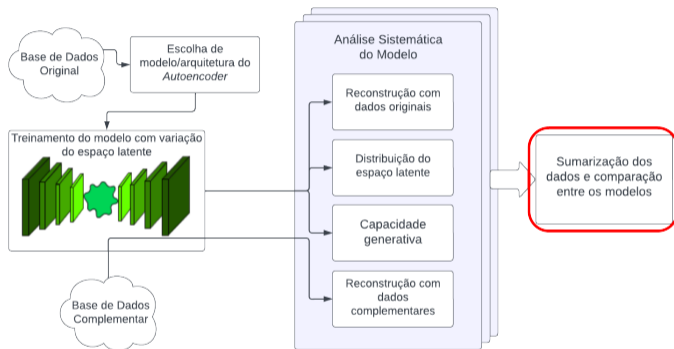


4. Análise: reconstrução com dados originais.
5. Análise: distribuição do espaço latente.

# Metodologia



6. Análise: reconstrução com dados complementares.
7. Análise: capacidade generativa.



## 8. Sumarização dos dados e comparação entre os modelos.

# Métricas de Avaliação

Para avaliar a replicação dos dados, utilizaremos o *Root Mean Squared Error* (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{d} \sum_{i=1}^d (\hat{x}_i - x_i)^2}$$

Nele,  $x_i$  e  $\hat{x}_i$  representam, respectivamente, os valores de entrada e saída dos dados processados pelo *Autoencoder* para todos os  $d$  valores que compõem os dados de entrada.

# Métricas de Avaliação

Para avaliar a redução de dimensionalidade, utilizaremos as seguintes métricas:

Métrica	Definição	Intervalo
Trustworthiness ( $M_t$ )	$1 - \frac{2}{nk(2n-3k-1)} \sum_{i \in N} \sum_{j \in U_i^{(k)}} (r(i, j) - k)$	<b>[0, 1]</b>
Continuity ( $M_c$ )	$1 - \frac{2}{nk(2n-3k-1)} \sum_{i \in N} \sum_{j \in V_i^{(k)}} (\hat{r}(i, j) - k)$	<b>[0, 1]</b>
Normalized stress ( $M_{ns}$ )	$\frac{\sum_{i,j} (\Delta o_{i,j} - \Delta q_{i,j})^2}{\max(\sum_{i,j} \Delta o_{i,j}^2, \sum_{i,j} \Delta q_{i,j}^2)}$	<b>[0, 1]</b>
Neighborhood hit ( $M_{nh}$ )	$\frac{\sum_{i \in N} K_i^{(k)}}{nk}$	<b>[0, 1]</b>
Shepard goodness ( $M_s$ )	$\frac{\text{Spearman coef.}(\Delta o_{i,j}, \Delta q_{i,j}) + 1}{2}$	<b>[0, 1]</b>

**Tabela:** Métricas utilizadas no trabalho. Na coluna “Intervalo”, está em destaque o valor de melhor eficácia.

Em sua maioria, elas trabalham com a relação entre os  $k$ -vizinhos mais próximos antes e depois da redução de dimensionalidade.

1. Introdução
2. Objetivos e Contribuições
3. Questões de Pesquisa
4. Trabalhos Relacionados
5. Metodologia
- 6. Materiais**
7. Resultados Preliminares
8. Plano de Trabalho e Cronograma

- *Modified National Institute of Standards and Technology (MNIST)* [4]: 70.000 imagens em escala de cinza de dígitos manuscritos, cada uma com dimensões de  $28 \times 28$  pixels.
- *Fashion-MNIST* [14]: 70.000 imagens em escala de cinza de artigos de vestuário, divididas em 10 classes distintas, incluindo camisetas, vestidos, tênis e outros itens relacionados à moda. Cada imagem possui dimensões de  $28 \times 28$  pixels, semelhante ao MNIST.
- *Extended MNIST (EMNIST)* [3]: 800.000 imagens em escala de cinza de letras manuscritas, cada uma com dimensões de  $28 \times 28$  pixels. Base com um formato compatível com o conjunto de dados MNIST.
- *Painter by Numbers*: cerca de 80.000 imagens de obras de arte de diferentes tamanhos e artistas.



# Recursos Computacionais

- Python e bibliotecas como: NumPy, Pandas, scikit-learn, TensorFlow, Keras, PyTorch e Matplotlib.
- Plataforma Google Colab, com CPU, GPU e armazenamento de dados.
- Recursos computacionais do Instituto de Computação (IC) da Unicamp, com CPU, GPU e armazenamento de dados.
- Supercomputador Santos Dumont, através do projeto “Aprendizado de Máquina Profundo para Problemas de Visão Computacional e Biologia Computacional” coordenado pelo Prof. Zanoni Dias, com GPUs de alto poder computacional para processamento de dados.

1. Introdução
2. Objetivos e Contribuições
3. Questões de Pesquisa
4. Trabalhos Relacionados
5. Metodologia
6. Materiais
- 7. Resultados Preliminares**
8. Plano de Trabalho e Cronograma

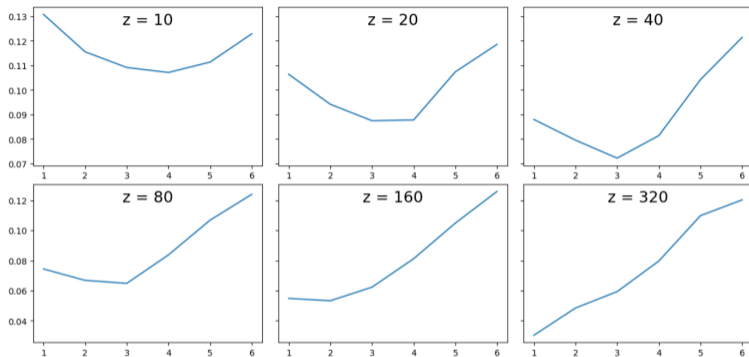
# Experimentos Iniciais com *Autoencoders* - FCAE

Base de dados: MNIST, entrada da rede de tamanho 784 ( $28 \times 28$  *pixels*)

De acordo com a metodologia:

1. **Implementação do modelo:** *Fully Connected Autoencoder* (FCAE).
2. **Definição da arquitetura:** variação da Camada Latente ( $z$ ) com valores 10, 20, 40, 80, 160 e 320 para escolha da melhor configuração de camadas.
  - **1 camada:**  $784 \rightarrow 392 \rightarrow z$ .
  - **2 camadas:**  $784 \rightarrow 523 \rightarrow 260 \rightarrow z$ .
  - **3 camadas:**  $784 \rightarrow 588 \rightarrow 392 \rightarrow 196 \rightarrow z$ .
  - **4 camadas:**  $784 \rightarrow 627 \rightarrow 470 \rightarrow 313 \rightarrow 156 \rightarrow z$ .
  - **5 camadas:**  $784 \rightarrow 654 \rightarrow 524 \rightarrow 394 \rightarrow 264 \rightarrow 134 \rightarrow z$ .
  - **6 camadas:**  $784 \rightarrow 672 \rightarrow 560 \rightarrow 448 \rightarrow 336 \rightarrow 224 \rightarrow 112 \rightarrow z$ .

# Experimentos Iniciais com *Autoencoders* - FCAE



**Figura:** Variação do RMSE no conjunto de testes de acordo com as camadas do modelo para diferentes valores  $z$  da Camada Latente.

Foi adotada a configuração de 3 camadas para a sequência dos experimentos.

# Experimentos Iniciais com *Autoencoders* - FCAE

3. **Treinamento do modelo com variação do espaço latente:** com  $z$  (Camada Latente) variando entre 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600 e 700.

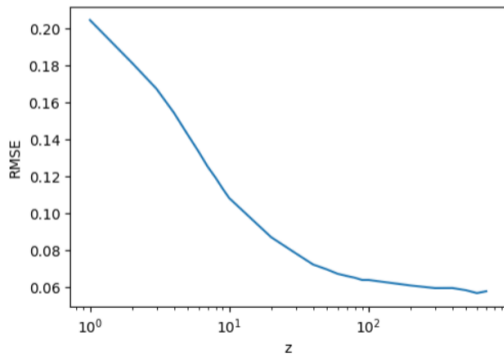


Figura: Variação do RMSE (eixo y) no conjunto de testes de acordo com a variação da quantidade de neurônios  $z$  da Camada Latente (eixo x em escala logarítmica).

# Experimentos Iniciais com *Autoencoders* - FCAE

## 4. Análise: reconstrução com dados originais.

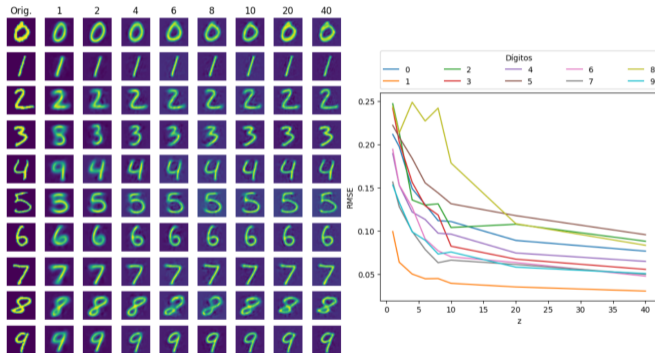


Figura: (a) à esquerda, imagens originais e processadas para diferentes valores de  $z$ ; (b) à direita, evolução do RMSE em relação a  $z$ .

# Experimentos Iniciais com *Autoencoders* - FCAE

## 5. Análise: distribuição do espaço latente através do UMAP.

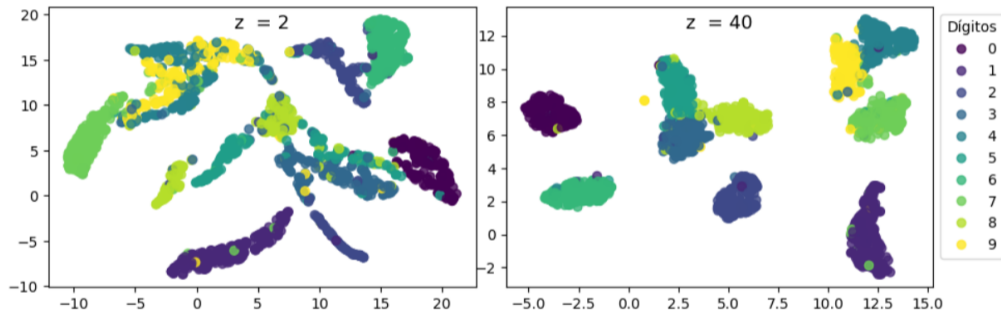


Figura: Redução de dimensionalidade realizada pelo UMAP (não supervisionado). Na figura à esquerda, é apresentado o resultado para  $z = 2$  e, à direita, para  $z = 40$ .

# Experimentos Iniciais com *Autoencoders* - FCAE

## 5. Análise: distribuição do espaço latente através das métricas.

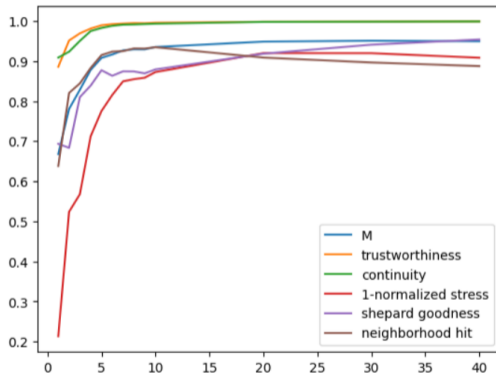


Figura: Evolução das métricas definidas na metodologia, de acordo com a variação de  $z$ .



# Experimentos Iniciais com *Autoencoders* - FCAE

## 6. Análise: reconstrução com dados complementares.

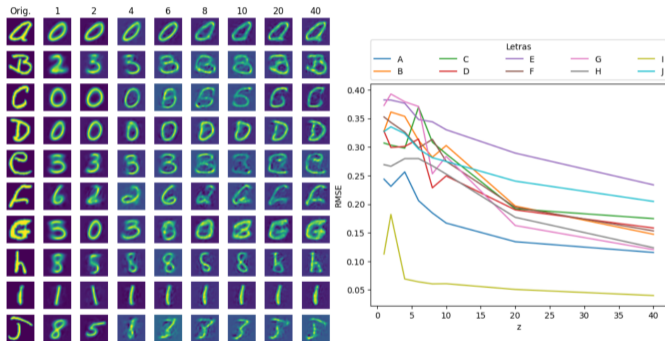


Figura: (a) à esquerda, imagens originais (EMNIST) e processadas para diferentes valores de  $z$ ; (b) à direita, evolução do RMSE em relação a  $z$ .

# Experimentos Iniciais com *Autoencoders* - FCAE

## 6. Análise: reconstrução com dados complementares.

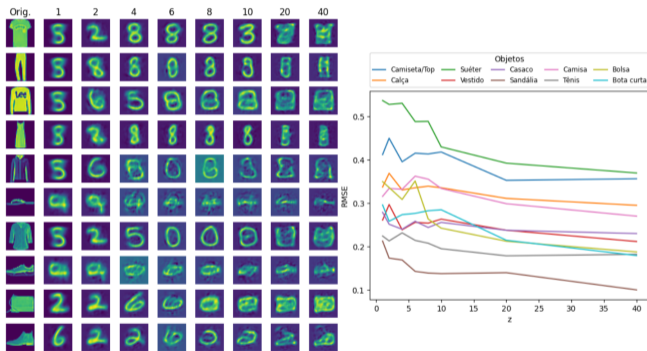


Figura: (a) à esquerda, imagens originais (*Fashion-MNIST*) e processadas para diferentes valores de  $z$ ; (b) à direita, evolução do RMSE em relação a  $z$ .

# Experimentos Iniciais com *Autoencoders* - FCAE

## 7. Análise: capacidade generativa.

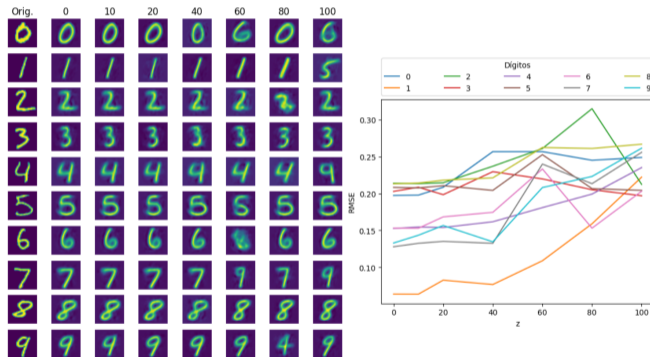


Figura: (a) à esquerda, imagens originais e processadas para  $z = 2$  com perturbação máxima do espaço latente de 0 a 100%; (b) à direita, evolução do RMSE em relação à perturbação aplicada.

# Experimentos Iniciais com *Autoencoders* - FCAE

## 7. Análise: capacidade generativa.

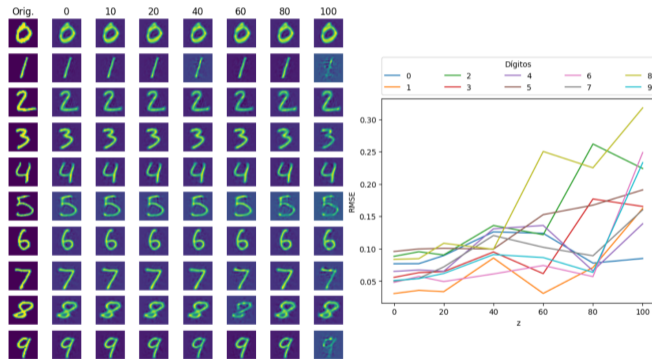


Figura: (a) à esquerda, imagens originais e processadas para  $z = 40$  com perturbação máxima do espaço latente de 0 a 100%; (b) à direita, evolução do RMSE em relação à perturbação aplicada.

# Experimentos Iniciais com *Autoencoders* - VAE

Base de dados: MNIST, entrada da rede de tamanho 784 ( $28 \times 28$  pixels)

De acordo com a metodologia:

1. **Implementação do modelo:** *Variational Autoencoder* (VAE).
2. **Definição da arquitetura:** 3 camadas ( $784 \rightarrow 588 \rightarrow 392 \rightarrow 196 \rightarrow z$ ).

# Experimentos Iniciais com *Autoencoders* - VAE

3. **Treinamento do modelo com variação do espaço latente:** com  $z$  (Camada Latente) variando entre 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 200, 300, 400, 500, 600 e 700.

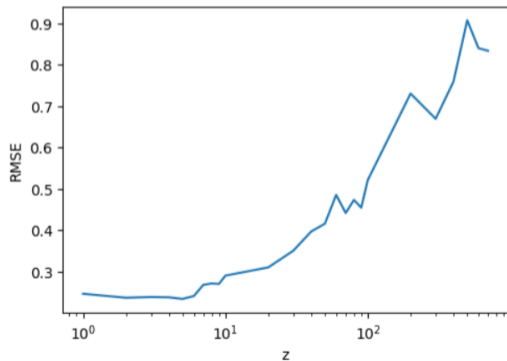


Figura: Variação do RMSE (eixo  $y$ ) no conjunto de testes de acordo com a variação da quantidade de neurônios  $z$  da Camada Latente (eixo  $x$  em escala logarítmica).

# Experimentos Iniciais com *Autoencoders* - VAE

## 4. Análise: reconstrução com dados originais.

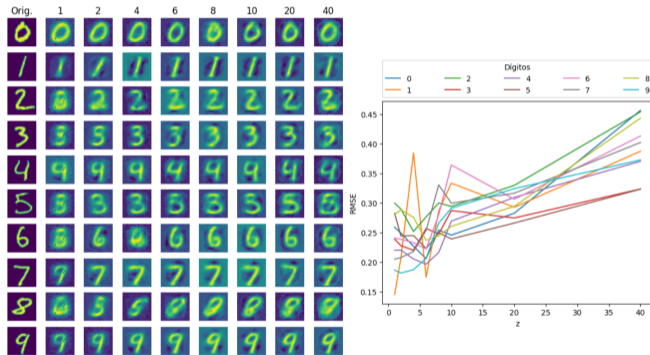


Figura: (a) à esquerda, imagens originais e processadas para diferentes valores de  $z$ ; (b) à direita, evolução do RMSE em relação a  $z$ .

# Experimentos Iniciais com *Autoencoders* - VAE

## 5. Análise: distribuição do espaço latente através do UMAP.

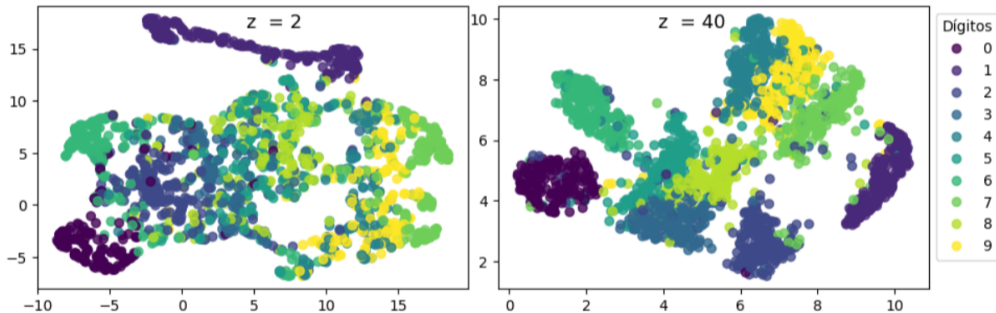


Figura: Redução de dimensionalidade realizada pelo UMAP (não supervisionado). Na figura à esquerda, é apresentado o resultado para  $z = 2$  e, à direita, para  $z = 40$ .



# Experimentos Iniciais com *Autoencoders* - VAE

## 5. Análise: distribuição do espaço latente através do UMAP.

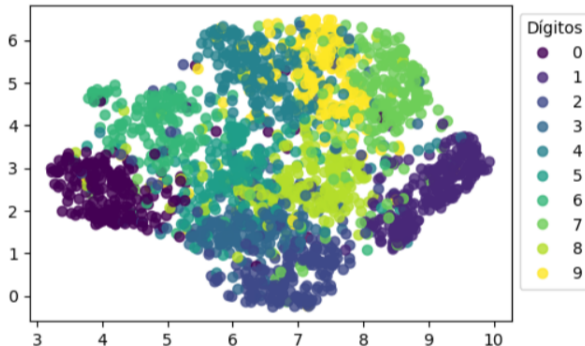


Figura: Redução de dimensionalidade realizada pelo UMAP (não supervisionado) para  $z = 100$ . Nesta imagem, a melhor organização do espaço latente está mais evidenciada.

# Experimentos Iniciais com *Autoencoders* - VAE

## 5. Análise: distribuição do espaço latente através das métricas.

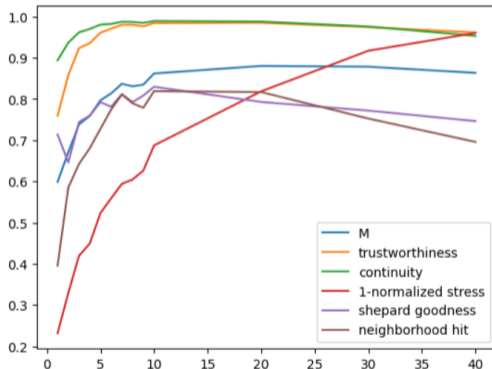


Figura: Evolução das métricas definidas na metodologia, de acordo com a variação de  $z$ .

# Experimentos Iniciais com *Autoencoders* - VAE

## 6. Análise: reconstrução com dados complementares.

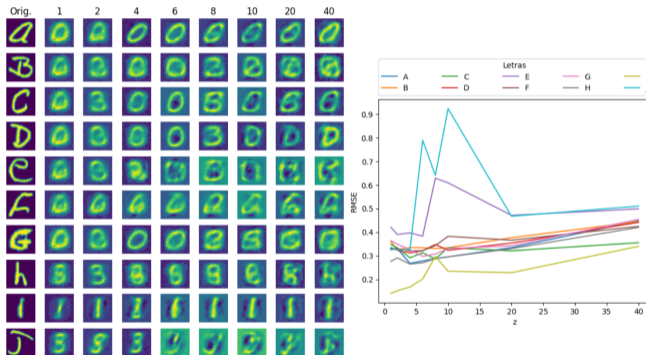


Figura: (a) à esquerda, imagens originais (EMNIST) e processadas para diferentes valores de  $z$ ; (b) à direita, evolução do RMSE em relação a  $z$ .

# Experimentos Iniciais com *Autoencoders* - VAE

## 6. Análise: reconstrução com dados complementares.

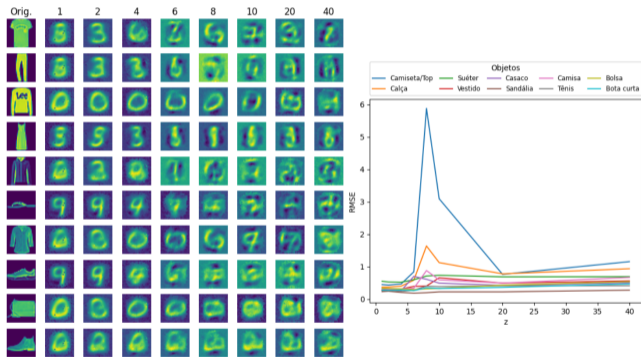


Figura: (a) à esquerda, imagens originais (*Fashion-MNIST*) e processadas para diferentes valores de  $z$ ; (b) à direita, evolução do RMSE em relação a  $z$ .

# Experimentos Iniciais com *Autoencoders* - VAE

## 7. Análise: capacidade generativa.

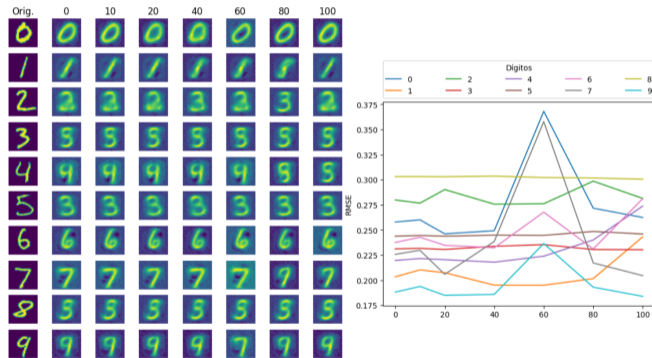


Figura: (a) à esquerda, imagens originais e processadas para  $z = 2$  com perturbação máxima do espaço latente de 0 a 100%; (b) à direita, evolução do RMSE em relação à perturbação aplicada.

# Experimentos Iniciais com *Autoencoders* - VAE

## 7. Análise: capacidade generativa.

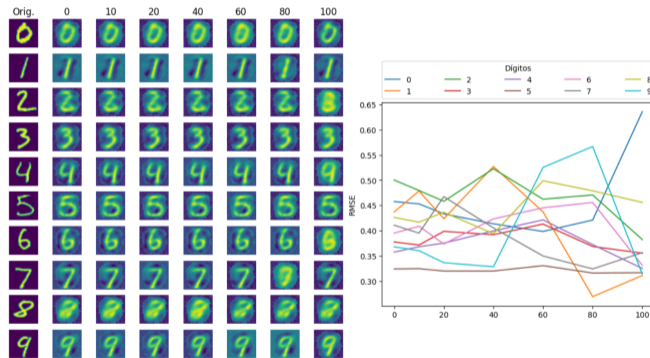


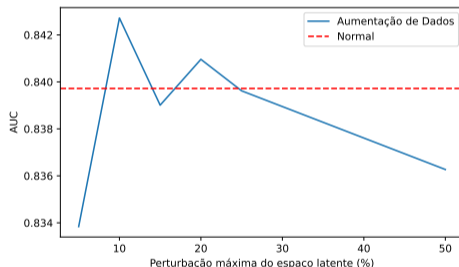
Figura: (a) à esquerda, imagens originais e processadas para  $z = 40$  com perturbação máxima do espaço latente de 0 a 100%; (b) à direita, evolução do RMSE em relação à perturbação aplicada.

# Experimentos Iniciais - Aumentação de Dados

- Avaliamos a capacidade generativa do FCAE para aumento de dados em um problema real.
- Objetivo: Criar um modelo capaz de receber duas imagens de obras de arte e dizer se elas são ou não do mesmo autor.
- Estratégia para solução: Utilizamos a rede pré-treinada Inception v3 como extrator de 2048 *features*. Treinamos um outro modelo para receber essas entradas e determinar se são ou não do mesmo artista.
- Aumento de dados: Treinamos um FCAE de 3 camadas ( $2048 \rightarrow 1536 \rightarrow 1024 \rightarrow 512 \rightarrow z$ ) e variamos o espaço latente para gerar dados sintéticos.

# Experimentos Iniciais - Aumentação de Dados

**Artistas com poucas obras de arte:** a primeira estratégia foi realizar aumento de dados em artistas com poucas amostras.

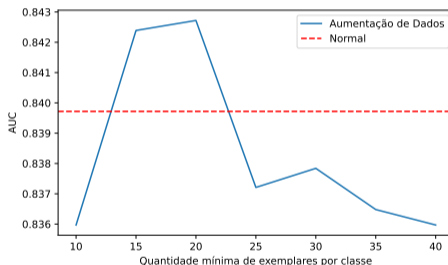


**Figura:** Variação da AUC no conjunto de teste com uso de aumento de dados de acordo com a porcentagem de perturbação máxima do espaço latente. Artistas com menos de 20 obras tiveram dados sintéticos criados até esse valor.



# Experimentos Iniciais - Aumentação de Dados

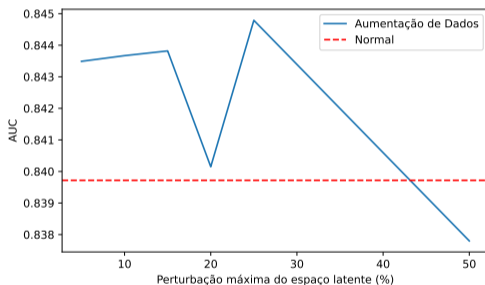
Como houve um pico de eficácia na perturbação de 10%, repetimos experimentos variando a quantidade mínima de exemplares por classe a serem aumentadas.



**Figura:** Variação da AUC no conjunto de teste com uso de aumento de dados de acordo com a quantidade mínima de exemplares por classe que foram aumentadas. A perturbação do espaço latente foi de 10% em todos os casos.

# Experimentos Iniciais - Aumentação de Dados

**Aumentação das amostras:** criamos um dado sintético para cada amostra.



**Figura:** Variação da AUC no conjunto de teste com uso de aumento de dados de acordo com a porcentagem de perturbação máxima do espaço latente. Foi gerada uma imagem sintética adicional para cada amostra original.

# Experimentos Iniciais - Conclusões

Foi possível aplicar a metodologia proposta em modelos de *Autoencoders* e obter resultados preliminares.

Realizamos experimentos complementares para avaliar a capacidade generativa de *Autoencoders* em um problema real.

1. Introdução
2. Objetivos e Contribuições
3. Questões de Pesquisa
4. Trabalhos Relacionados
5. Metodologia
6. Materiais
7. Resultados Preliminares
8. Plano de Trabalho e Cronograma

# Plano de Trabalho e Cronograma

Cronograma de execução das atividades propostas, em um prazo de 24 meses.

Atividades	1º ano						2º ano					
	1	2	3	4	5	6	1	2	3	4	5	6
Pesquisa bibliográfica	•	•	•	•	•	•		•		•		•
Aproveitamento de disciplinas	•											
Proficiência de inglês		•										
Definição dos modelos a serem analisados			•	•								
Realização do EQM				•								
Replicação dos códigos a serem comparados				•	•	•	•					
Análise e comparação dos resultados.							•	•	•			
Publicação dos resultados										•	•	
Escrita da dissertação										•	•	•
Defesa da dissertação												•

Tabela: Cronograma de atividades dividido em bimestres.

# Referências I

- [1] D. Bank, N. Koenigstein, and R. Giryes. *Autoencoders*, pages 353–374. Springer International Publishing, Cham, 2023.
- [2] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.
- [3] G. Cohen, S. Afshar, J. Tapson, and A. Van Schaik. EMNIST: Extending MNIST to handwritten letters. In *International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.
- [4] L. Deng. The MNIST database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.
- [5] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009, 2022.
- [6] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [7] P. Li, Y. Pei, and J. Li. A comprehensive survey on design and application of autoencoder in deep learning. *Applied Soft Computing*, 138:110176, 2023.

# Referências II

- [8] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey.  
Adversarial autoencoders.  
*arXiv preprint arXiv:1511.05644*, 2015.
- [9] A. Ng.  
Sparse autoencoder.  
*CS294A Lecture Notes*, 72(2011):1–19, 2011.
- [10] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio.  
Contractive auto-encoders: explicit invariance during feature extraction.  
In *International Conference on Machine Learning (ICML)*, pages 833–840, 2011.
- [11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams.  
Learning representations by back-propagating errors.  
*Nature*, 323(6088):533–536, Oct 1986.
- [12] C. C. Tan and C. Eswaran.  
Performance comparison of three types of autoencoder neural networks.  
In *2nd Asia International Conference on Modelling & Simulation (AMS)*, pages 213–218, 2008.
- [13] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol.  
Extracting and composing robust features with denoising autoencoders.  
In *International Conference on Machine Learning (ICML)*, page 1096–1103, New York, NY, USA, 2008. Association for Computing Machinery.
- [14] H. Xiao, K. Rasul, and R. Vollgraf.  
Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms.  
*arXiv preprint arXiv:1708.07747*, 2017.

# Referências III

- [15] M. D. Zeiler and R. Fergus.  
Visualizing and understanding convolutional networks.  
In *European Conference on Computer Vision (ECCV)*, pages 818–833, Cham, 2014. Springer International Publishing.
- [16] J. Zhai, S. Zhang, J. Chen, and Q. He.  
Autoencoder and its various variants.  
In *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 415–419. IEEE, 2018.
- [17] G. Zhang, Y. Liu, and X. Jin.  
A survey of autoencoder-based recommender systems.  
*Frontiers of Computer Science*, 14:430–450, 2020.



Obrigado!