
Autoencoders

Prof. Mateus Mendelson
mendelson.mateus@gmail.com

mmendelson.com

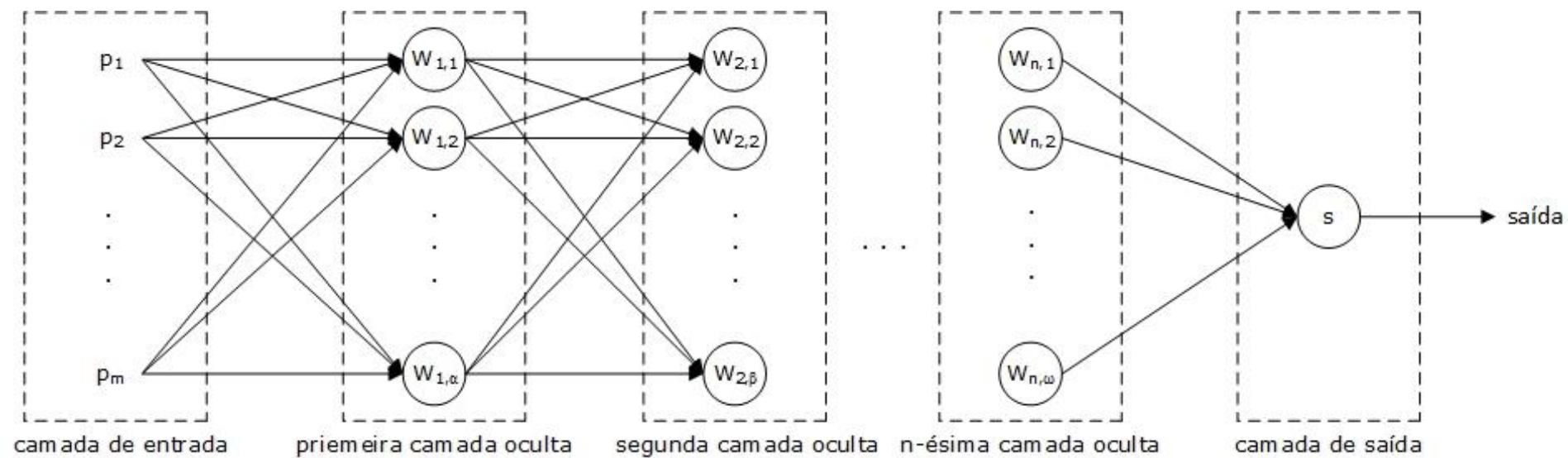


1. Introdução

- **Redes neurais artificiais** surgiram inspiradas nas redes neurais biológicas presentes no sistema nervoso central humano.
- Tais redes consistem em um arranjo de neurônios interconectados que trocam informações entre si, permitindo a detecção, aprendizado e aplicação de padrões.
- Redes neurais artificiais são muito utilizadas em áreas ligadas à inteligência artificial, principalmente para resolver problemas de classificação, de ajuste de função (regressão), de robótica, de controle, etc.

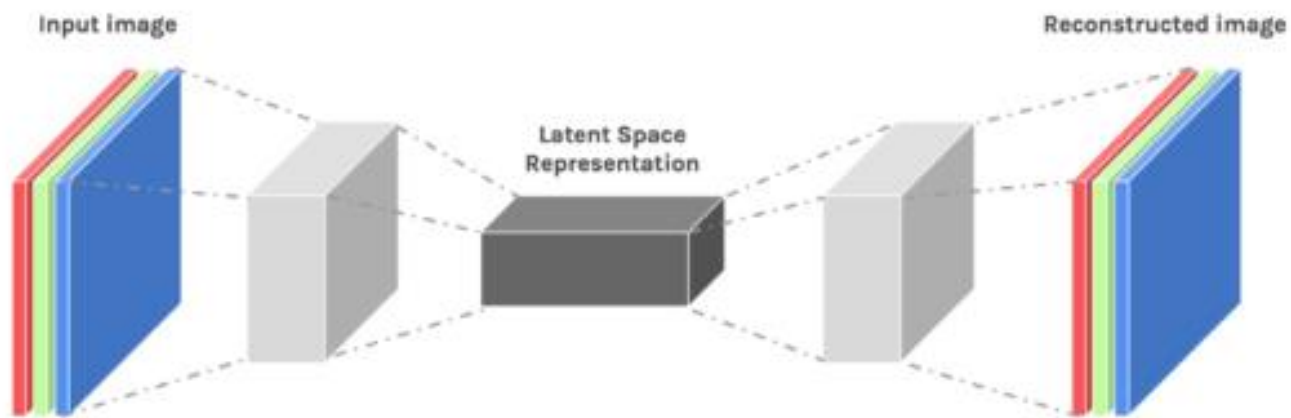


1. Introdução



1. Introdução

- Autoencoders são redes neurais não-supervisionadas que utilizam aprendizado de máquina para comprimir dados.
- Na verdade, eles aplicam o algoritmo de treinamento *Backpropagation* para calibrar os pesos e bias de tal forma que a saída da rede seja igual à entrada.

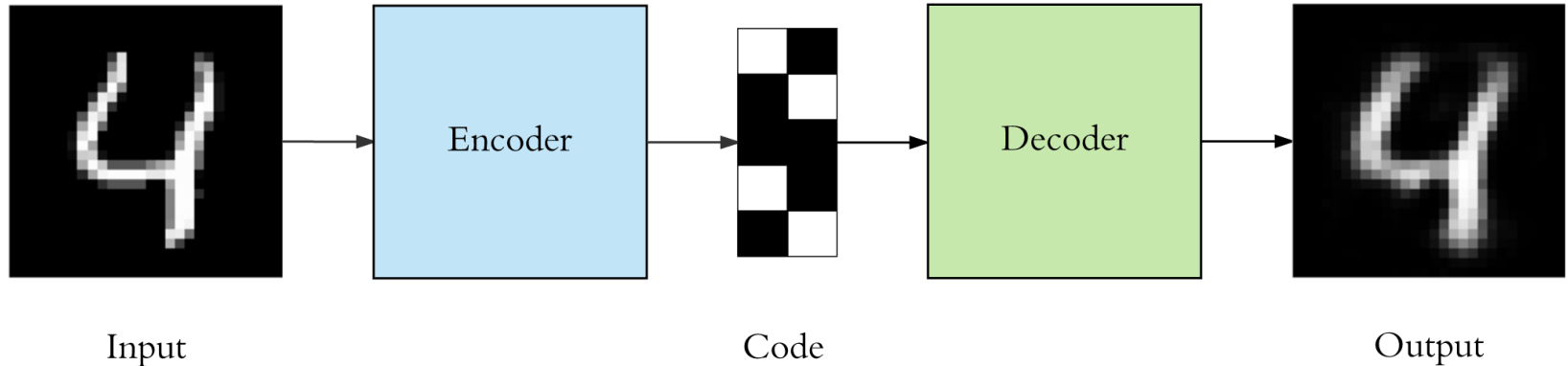


1. Introdução

- Um autoencoder é composto por três partes.
 - ✓ **Codificador:** é a parte da rede que comprime a entrada, a transformando em uma representação espacial latente.
 - ✓ **Código:** é a parte da rede que representa a entrada comprimida. Necessariamente possui dimensões menores do que a entrada. O quão menor ela é define o grau de compressão.
 - ✓ **Decodificador:** é a parte da rede que recebe o código como entrada e, então, reconstrói a imagem de volta para suas dimensões originais. A imagem resultante possui perdas e, por isso, não é idêntica à imagem original.



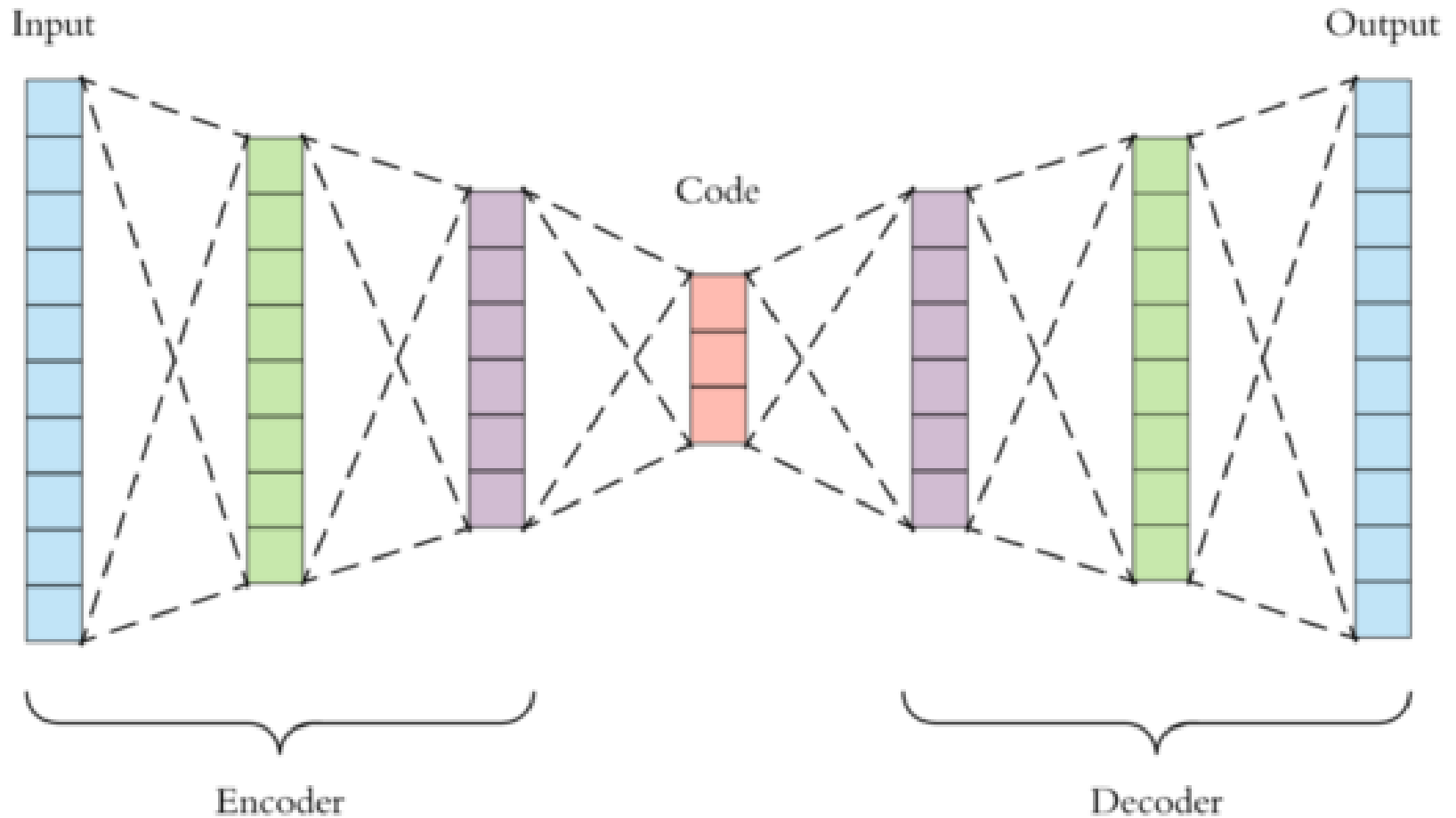
1. Introdução



Fonte: <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798>



1. Introdução



Fonte: <https://towardsdatascience.com/applied-deep-learning-part-3-autoencoders-1c083af4d798>



2. Exemplo

- Foram realizados testes com 4 situações (todos os neurônios utilizam a função de ativação relu, exceto pela última camada do decodificador, que utiliza sigmoid):
 - ✓ **Situação 1:** codificador e decodificador com 2 camadas cada. O código gerado é uma matriz de números reais de dimensões 14×14 (6272 bits).
 - ✓ **Situação 2:** codificador e decodificador com 3 camadas cada. O código gerado é uma matriz de números reais de dimensões 7×7 (1568 bits).



2. Exemplo

- Foram realizados testes com 4 situações (todos os neurônios utilizam a função de ativação relu, exceto pela última camada do decodificador, que utiliza sigmoid):
 - ✓ **Situação 3:** codificador e decodificador com 3 camadas cada. O código gerado é uma matriz de números reais de dimensões 2×2 (128 bits).
 - ✓ **Situação 4:** codificador e decodificador com 3 camadas cada. O código gerado é uma matriz de números reais de dimensões 1×1 (32 bits).
- A base de dados utilizada é a not-MNIST, que é composta por 70000 imagens em níveis de cinza de dimensões 28×28 . As imagens são representações das letras A até J.



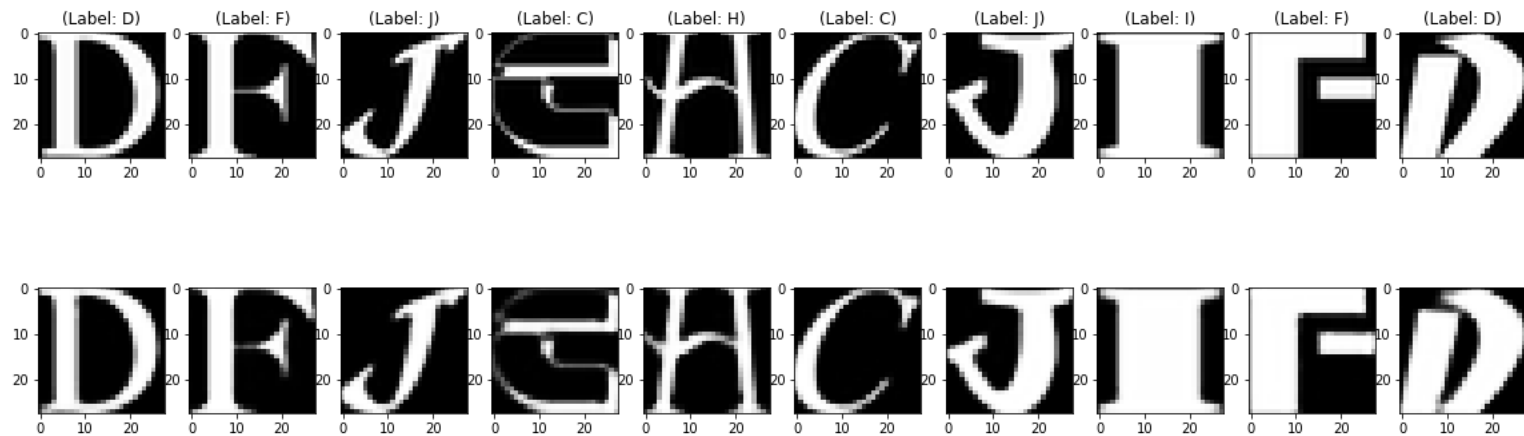
2. Exemplo

- A base de dados foi dividida em 3 conjuntos:
 - ✓ **Treinamento:** composto por 42000 imagens, que são utilizadas para calibrar a rede.
 - ✓ **Validação:** composto por 18000 imagens, que são utilizadas para evitar *overfitting*.
 - ✓ **Teste:** composto por 10000 imagens, que são utilizadas para avaliar o desempenho da rede.



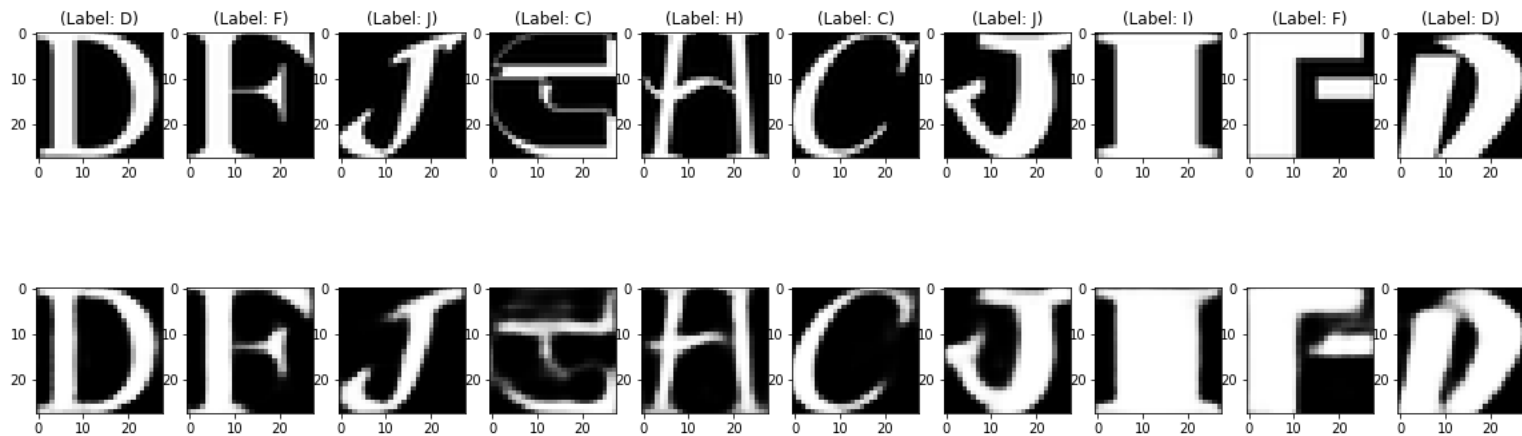
2. Exemplo

- Situação 1



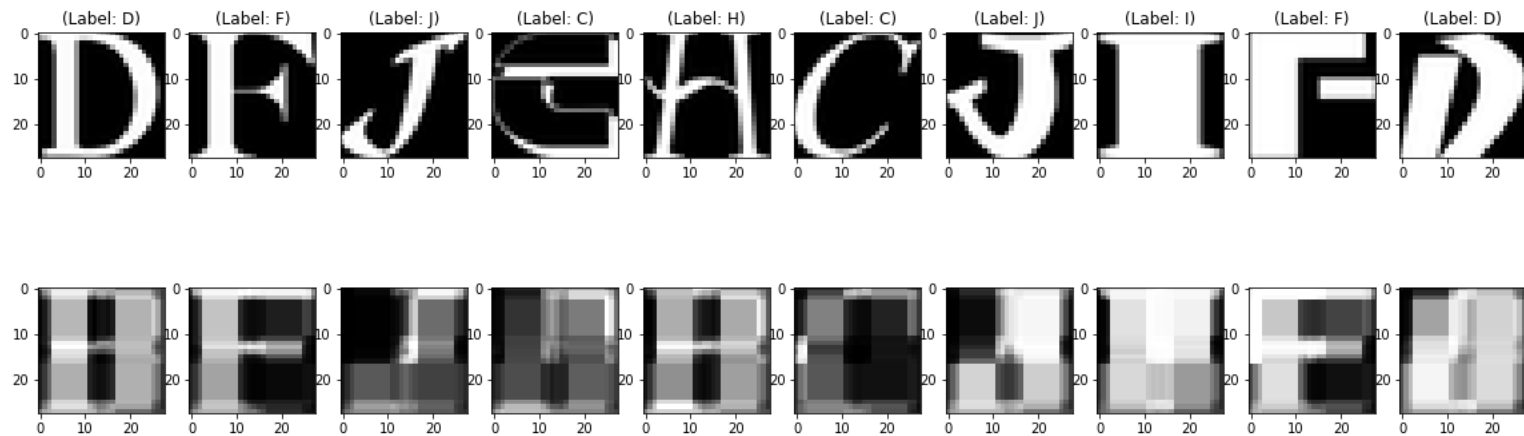
2. Exemplo

- Situação 2



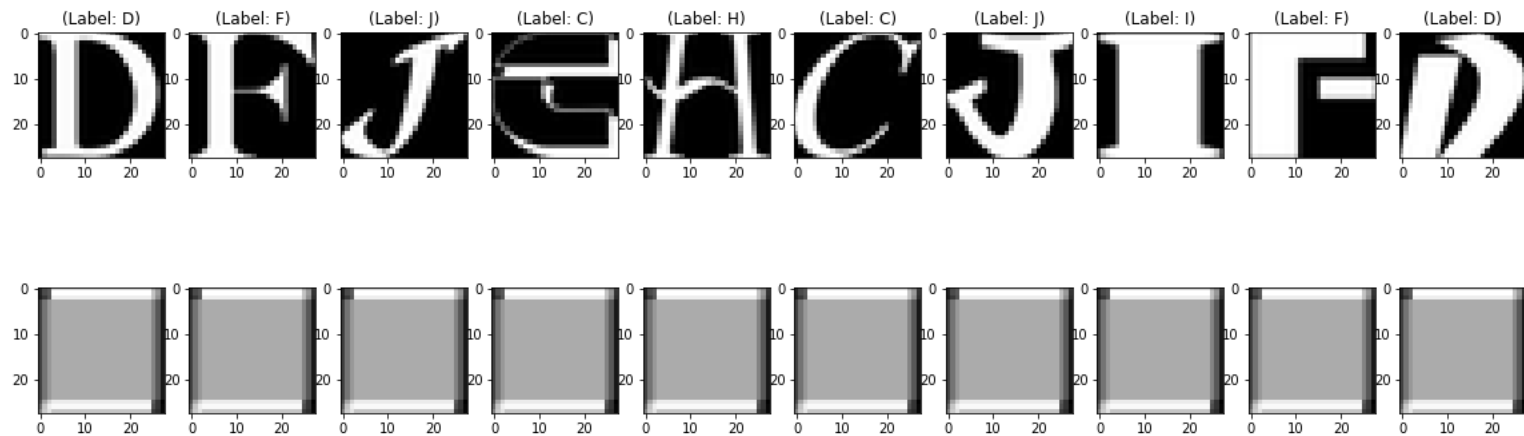
2. Exemplo

- Situação 3



2. Exemplo

- Situação 4



3. Algumas Observações

- Autoencoders são eficientes ao reduzir a dimensionalidade de suas entradas.
- Dependendo da aplicação, ainda não são bons o suficiente para realizar compressão de imagens. Estamos no início de seu uso.
- É uma compressão com perdas perceptíveis ao olho nu.
- Necessita de uma grande base de dados para que possa alcançar bons resultados.
- Uma vez treinada, a mesma rede pode ser utilizada no codificador e no decodificador, sendo necessário armazenar apenas os bits da camada de código.



4. Codificando

- Bem, vamos codificar e ver como aplicar essas coisas.
- Abram seus notebooks!



5. Mini-projeto

- Para este projeto, iremos utilizar o dataset do Street View para comprimir imagens, que pode ser encontrado aqui: <http://ufldl.stanford.edu/housenumbers/>
- Tarefa 1: implementar um autoencoder para esse dataset. Note que essas imagens são coloridas, em contraste com o MNIST que utilizamos em aula.
- Tarefa 2: utilizando esse mesmo dataset, crie imagens ruidosas e implemente um autoencoder que realiza o denoising dessas imagens.
- Desafio: implemente as duas tarefas com o uso de uma GPU em vez de realizar o treinamento em CPU. Note a diferença absurda no tempo de execução!



5. Mini-projeto

- O seu relatório será o notebook exportado para um arquivo HTML e deve conter:
 - ✓ Visualização da arquitetura de cada autoencoder.
 - ✓ Visualização de exemplos das imagens originais, das imagens ruidosas (para a tarefa 2) e suas respectivas imagens processadas pelo autoencoder.

