

Objetivo da atividade

Objetivo desta atividade é compreender os principais tópicos teóricos sobre *Autoencoders* (AEs). Além disso, adicionalmente, propõe-se um exercício prático para prática de programação.

1 Exercícios teóricos

Exercício 1. Quais são as principais tarefas que os diferentes tipos de *Autoencoders* (AEs) podem ser aplicados?

Exercício 2. Suponhamos que você quer treinar um classificador, mas você tem acesso a um dataset com uma grande quantidade de dados não-rotulados e algumas centenas de exemplos rotulados. Como os *Autoencoders* podem ajudar? Como você procederia?

Exercício 3. Se um *Autoencoder* reconstrói perfeitamente um conjunto de entradas, ele é necessariamente um bom AE? Como você pode avaliar o desempenho de um AE?

Exercício 4. O que são AEs subcompletos e supercompletos? Qual é o risco de um AE excessivamente subcompleto? E quais os riscos de se ter um AE supercompleto?

Exercício 5. Como se “amarram” os pesos em um *Stacked Autoencoder* (SAE)? Quais são os motivos de se realizar tal abordagem?

2 Exercício Prático - Opcional

Tente explorar um *Denoising Autoencoder* (dAE) para pré-treinar um classificador de imagens usando o dataset do MNIST. Siga os seguintes passos:

- Divida o dataset em conjuntos de treinamento e teste. **Dica:** na biblioteca *keras* do Python o dataset MNIST já se encontra dividido em conjuntos de treinamento e teste;
- Treine o dAE usando todo o conjunto de treinamento;
- Verifique se as imagens são razoavelmente reconstruídas;
- Cria uma Rede Neural Profunda (DNN), reusando os pesos das camadas do dAE. Treine-a usando apenas 500 imagens aleatoriamente selecionadas do conjunto de treinamento.
- Avalie o modelo induzido no conjunto de teste.

Uma vez realizado o experimento, pergunta-se: a DNN tem desempenho melhor com ou sem a etapa de pré-treinamento via dAE? Elabore uma discussão/comentário crítico sobre o experimento realizado, destacando todas as escolhas de implementação (arquiteturas, medidas de desempenho, etc). Opcionalmente, caso queira, encaminhe também o código ou link para o mesmo (git, notebook, etc).

3 Links

Alguns links que podem ajudar no desenvolvimento desta atividade:

- **Keras** - framework para implementação de redes neurais e deep learning com Python e Tensor Flow. É possível criar MLPs com as funções e componentes existentes. <https://keras.io>
- <https://www.pluralsight.com/guides/use-autoencoders-to-denoise-images>
- <https://www.pyimagesearch.com/2020/02/24/denoising-autoencoders-with-keras-tensorflow-and-deep-learning/>
- <https://github.com/wikibook/keras/blob/master/chapter3-autoencoders/denoising-autoencoder-mnist-3.3.1.py>
- <https://www.machinecurve.com/index.php/2019/12/20/building-an-image-denoiser-with-a-keras-autoencoder-neural-network/>