Topological optimization

October 4, 2021

Abstract

The purpose of this project is to perform simple topological optimization tasks. For this, gradient descent will be used on persistence diagrams in order to solve specific regularization problems, such as moving points around so that they form loops, or preventing classifiers to overfit. The effect of using different types of losses will also be studied.

- Implement an algorithm that uses the `persistence_pairs()` function of `Gudhi` to produce the gradient (seen in class) of Vietoris-Rips persistence diagrams. Alternatively, use autodiff libraries such as `TensorFlow` or `PyTorch` to compute this gradient automatically. If you want to skip this part, you can compile the `difftda` branch of `Gudhi`, which contains `TensorFlow` models for various filtrations:

  https://github.com/MathieuCarriere/gudhi/tree/diff

- Test your algorithm by running gradient descent on a point cloud (initialized randomly) so that the 1-dimensional Vietoris-Rips persistence diagram contains as many points as possible, i.e., the points form as many loops as possible. Use the regularization penalties seen in class.

- Download and read this article (section 3):

  https://drive.google.com/file/d/1a3KXbVi-5MKKaiuozZDtePcZNb-saKVx/view?usp=sharing

- Compute and implement the gradient of DTM-weighted Vietoris-Rips persistence diagrams.

- Test your algorithm by running gradient descent on a point cloud (initialized randomly) so that the 1-dimensional DTM-weighted Vietoris-Rips persistence diagram contains as many points as possible, i.e., the points form as many loops as possible even in the presence of noise. Use the regularization penalties seen in class.

- Download and read this article (section 3):

  https://drive.google.com/file/d/1UBVTTx0BGF0K2tgzKyRJf0r1Wl9hraKd/view?usp=sharing

- (Requires `TensorFlow` or `PyTorch`). Train a neural network architecture on MNIST dataset

  http://yann.lecun.com/exdb/mnist/

  and use many parameters so that it overfits. Then, use a persistent homology regularization term on the loss function to prevent overfitting.