ETH zürich

D-BSSE
Department of Biosystems Science and Engineering

Topological representation learning

Michael Moor

Machine Learning and Computational Biology Group, ETH Zurich

DataSig Seminar, Mathematical Institute, University of Oxford.

Ad personam

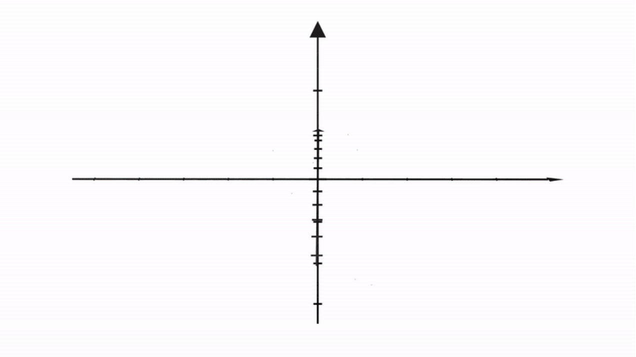
michaelmoor.ml @Michael_D_Moor

2018 - now	PhD at the Machine Learning and Computational Biology lab, ETH Zurich.
2016 - 2018	MD at University of Basel.
2011 - 2017	Medical studies University of Basel.

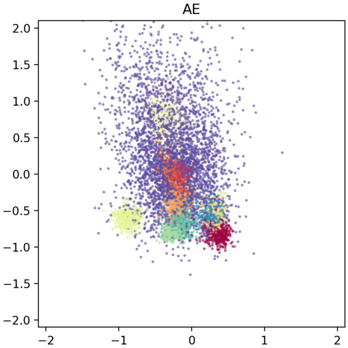
My research interests

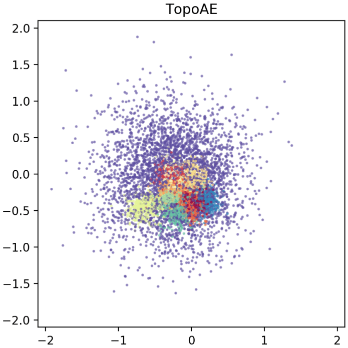
		Featured publications
Methods	Topological ML	Topological Autoencoders *(ICML 2020)*, Neural Persistence *(ICLR 2019)*
	Time series	Set Functions for Time Series *(ICML 2020)*, Imputing Signature Models *(Artemiss, ICML 2020)*
Applications	Clinical ML	Early prediction of sepsis *(MLHC 2019)*, Early prediction of circulatory failure *(Nature Medicine)*

Motivation

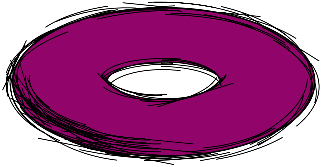


Motivation

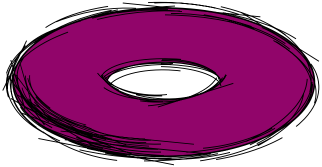

AE
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2


TopoAE
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2

Background



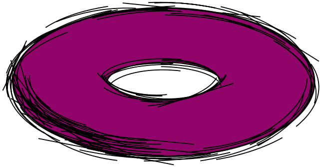
Background

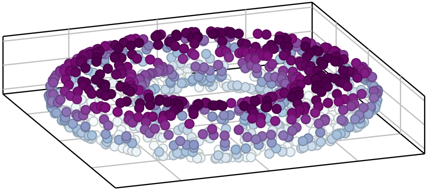


Betti numbers characterize topological spaces

- β_0 connected components
- β_1 cycles
- β_2 voids

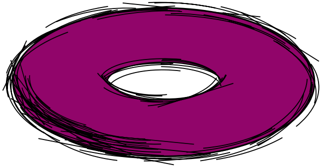
Background

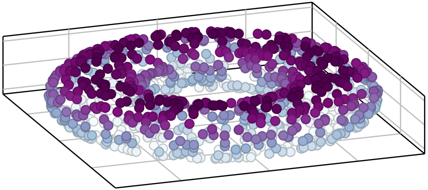




Betti numbers characterize topological spaces

- β_0 connected components
- β_1 cycles
- β_2 voids





Betti numbers characterize topological spaces

- β_0 connected components
- β_1 cycles
- β_2 voids

Issues

- Great for manifolds (which are usually unknown)
- But instead *approximated* via samples
- Topology on samples is noisy

Background

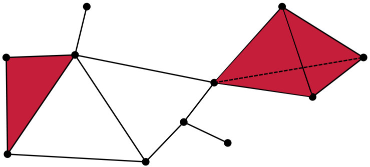
- In simplicial homology, Betti numbers can be calculated[1] from a *simplicial complex*.
- To define a simplicial complex we first need to define simplices:
- A k-simplex is the convex hull of $k + 1$ vertices.

[1] formally, the i-th Betti number is the rank of the i-th homology group of the simplicial complex
[2] image source: `https://umap-learn.readthedocs.io/en/latest/_images/simplices.png`

Background

- A simplicial complex K is a set of simplices fulfilling two criteria:
  1. Every face of a simplex in K is also in K.
  2. Any non-empty intersection of two simplices in K is a face of both simplices.

- Example:



[1] image source: http://bastian.rieck.me/research/talks/an_introduction_to_persistent_homology.pdf

Background

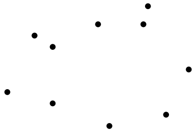
- How do we arrive at a simplicial complex from a point cloud? Which points should be connected?
- Problem: Adding or removing single points would change the Betti numbers of the resulting simplicial complex.
- This issue motivated *persistent* homology: Using a varying distance threshold ϵ, we can extract a nested sequence of simplicial complexes to extract topological features over varying scales ('multi-scale Betti numbers').

Persistent homology (PH)[3]

Vietoris-Rips Complex[2]: We ‘grow‘ a neighbourhood graph (simplicial complex for higher dimensions) and keep track of the appearance and disappearance of topological features.

Filtration:

$$\emptyset = \mathrm{K}_0 \subseteq \mathrm{K}_1 \subseteq \cdots \subseteq \mathrm{K}_{n-1} \subseteq \mathrm{K}_n = \mathrm{K}$$



$$E := \{ (u, v) \mid \mathrm{dist}\,(p_u, p_v) \leq \epsilon \}$$

[2] Vietoris [1927]

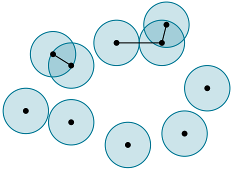
[3] Edelsbrunner and Harer [2008]

Persistent homology (PH)[3]

Vietoris-Rips Complex[2]: We 'grow' a neighbourhood graph (simplicial complex for higher dimensions) and keep track of the appearance and disappearance of topological features.

Filtration:

$$\emptyset = \mathrm{K}_0 \subseteq \mathrm{K}_1 \subseteq \cdots \subseteq \mathrm{K}_{n-1} \subseteq \mathrm{K}_n = \mathrm{K}$$



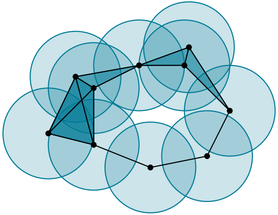
$$E := \left\{ (u, v) \mid \mathrm{dist}\left(p_u, p_v\right) \leq \epsilon \right\}$$

[2] Vietoris [1927]

[3] Edelsbrunner and Harer [2008]

Persistent homology (PH)[3]

Vietoris-Rips Complex[2]: We 'grow' a neighbourhood graph (simplicial complex for higher dimensions) and keep track of the appearance and disappearance of topological features.



Filtration:

$$\emptyset = \mathrm{K}_0 \subseteq \mathrm{K}_1 \subseteq \cdots \subseteq \mathrm{K}_{n-1} \subseteq \mathrm{K}_n = \mathrm{K}$$

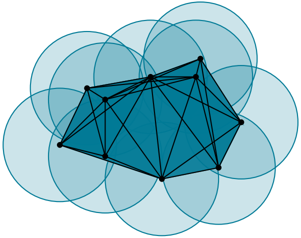
$$E := \{(u, v) \mid \operatorname{dist}(p_u, p_v) \leq \epsilon\}$$

[2] Vietoris [1927]

[3] Edelsbrunner and Harer [2008]

Persistent homology (PH)[3]

Vietoris-Rips Complex[2]: We 'grow' a neighbourhood graph (simplicial complex for higher dimensions) and keep track of the appearance and disappearance of topological features.



Filtration:

$$\emptyset = \mathrm{K}_0 \subseteq \mathrm{K}_1 \subseteq \cdots \subseteq \mathrm{K}_{n-1} \subseteq \mathrm{K}_n = \mathrm{K}$$

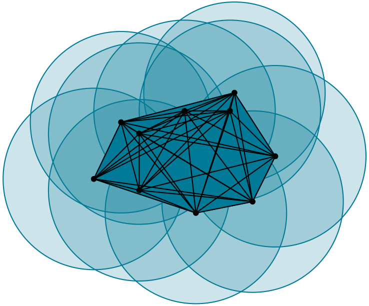
$$E := \left\{ (u, v) \mid \mathrm{dist}\left(p_u, p_v\right) \leq \epsilon \right\}$$

[2] Vietoris [1927]

[3] Edelsbrunner and Harer [2008]

Persistent homology (PH)[3]

Vietoris-Rips Complex[2]: We 'grow' a neighbourhood graph (simplicial complex for higher dimensions) and keep track of the appearance and disappearance of topological features.



Filtration:

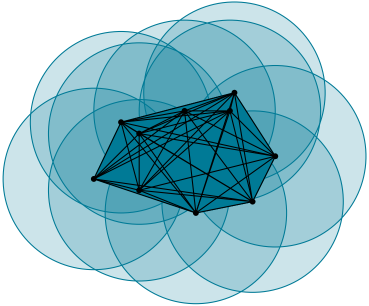
$$\emptyset = \mathrm{K}_0 \subseteq \mathrm{K}_1 \subseteq \cdots \subseteq \mathrm{K}_{n-1} \subseteq \mathrm{K}_n = \mathrm{K}$$

$$E := \left\{ (u, v) \mid \mathrm{dist}\left(p_u, p_v\right) \leq \epsilon \right\}$$

[2] Vietoris [1927]

[3] Edelsbrunner and Harer [2008]

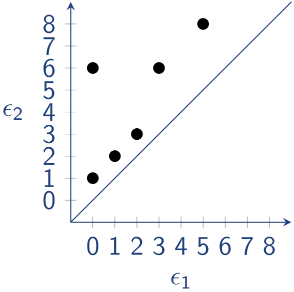
Vietoris-Rips Complex[2]**:** We ‘grow‘ a neighbourhood graph (simplicial complex for higher dimensions) and keep track of the appearance and disappearance of topological features.



$$E := \{ (u, v) \mid \mathrm{dist}(p_u, p_v) \leq \epsilon \}$$

Filtration:

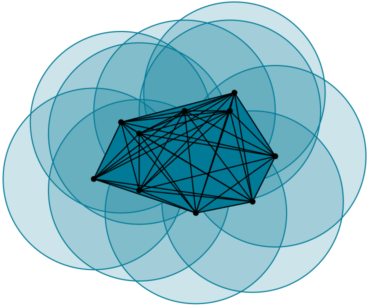
$$\emptyset = \mathrm{K}_0 \subseteq \mathrm{K}_1 \subseteq \cdots \subseteq \mathrm{K}_{n-1} \subseteq \mathrm{K}_n = \mathrm{K}$$



[2] Vietoris [1927]
[3] Edelsbrunner and Harer [2008]

Persistent homology (PH)[3]

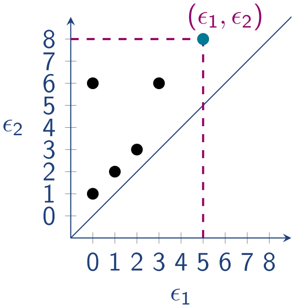
Vietoris-Rips Complex[2]: We 'grow' a neighbourhood graph (simplicial complex for higher dimensions) and keep track of the appearance and disappearance of topological features.



$$E := \{ (u, v) \mid \mathrm{dist}(p_u, p_v) \leq \epsilon \}$$

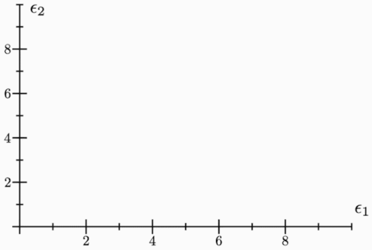
Filtration:

$$\emptyset = \mathrm{K}_0 \subseteq \mathrm{K}_1 \subseteq \cdots \subseteq \mathrm{K}_{n-1} \subseteq \mathrm{K}_n = \mathrm{K}$$

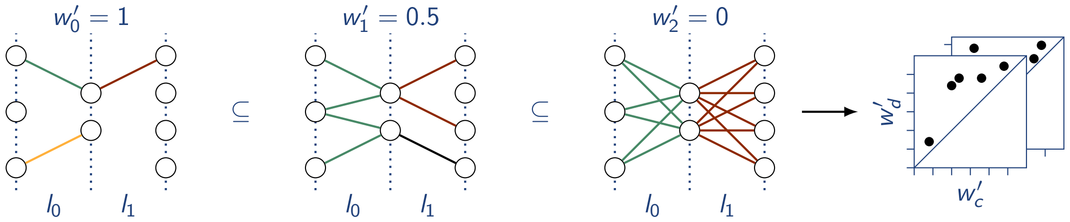


[2] Vietoris [1927]
[3] Edelsbrunner and Harer [2008]

Persistent homology II



Neural persistence: A complexity measure for deep neural networks using algebraic topology [4]

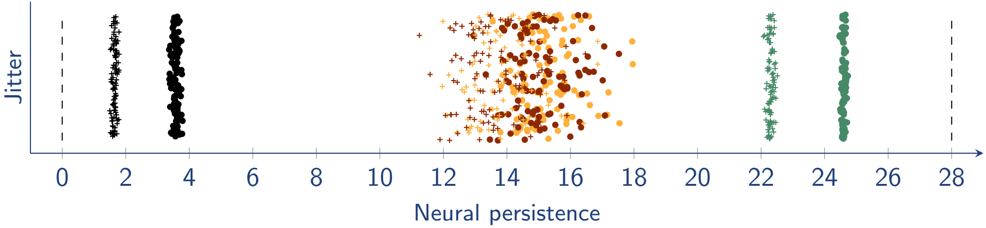


Illustrating the neural persistence calculation of a network with two layers (l_0 and l_1). Colours indicate connected components per layer. The filtration process is depicted by colouring connected components that are created or merged when the respective weights are greater than or equal to the threshold w'_i.

$$\mathrm{NP}(G_k) := \|\mathcal{D}_k\|_p := \left(\sum_{(c,d)\in\mathcal{D}_k} \mathrm{pers}(c,d)^p \right)^{\frac{1}{p}} \tag{1}$$

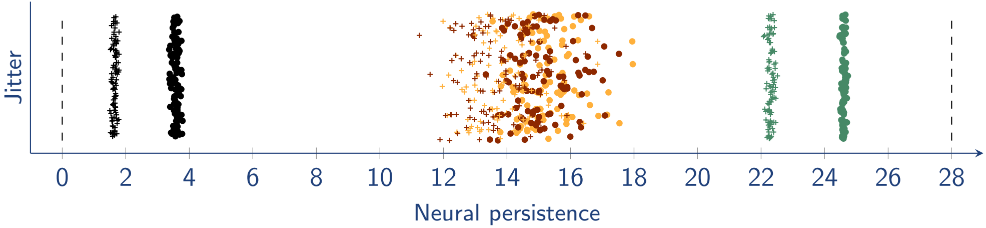
[4] Rieck et al. [2018]

Neural persistence for monitoring neural network training



Neural persistence values of trained perceptrons (green), diverging ones (yellow), random Gaussian matrices (red), and random uniform matrices (black). Dots indicate actually computed NP values while crosses indicate a predicted lower bound.

Neural persistence for monitoring neural network training



Neural persistence values of trained perceptrons (green), diverging ones (yellow), random Gaussian matrices (red), and random uniform matrices (black). Dots indicate actually computed NP values while crosses indicate a predicted lower bound. This was joint work with Bastian Rieck, Matteo Togninalli, Christian Bock, Max Horn, Thomas Gumbsch, and Karsten Borgwardt.

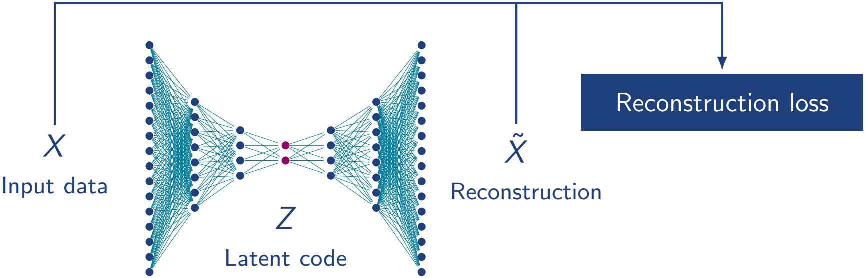
So we can observe and monitor topological features of neural networks, but can we *influence* them?

Topological autoencoders

A method for preserving topological features of the data space in low-dimensional representations [Moor et al., 2019].

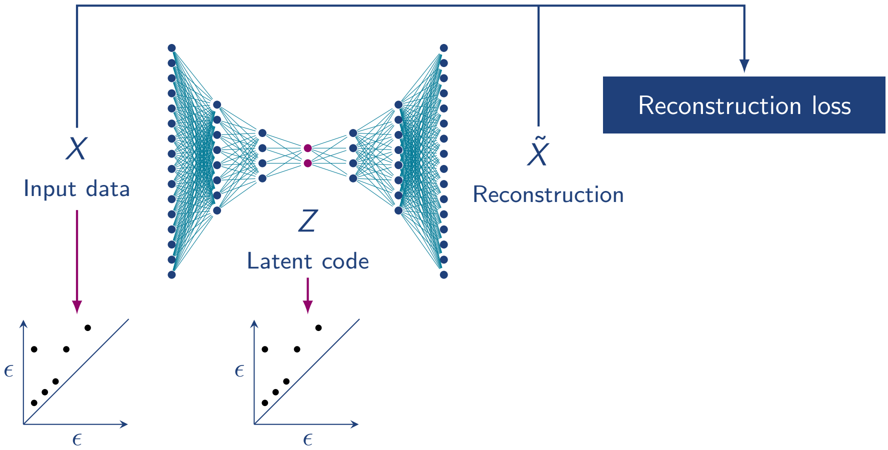
This was joint work with Max Horn, Bastian Rieck, and Karsten Borgwardt.

Overview



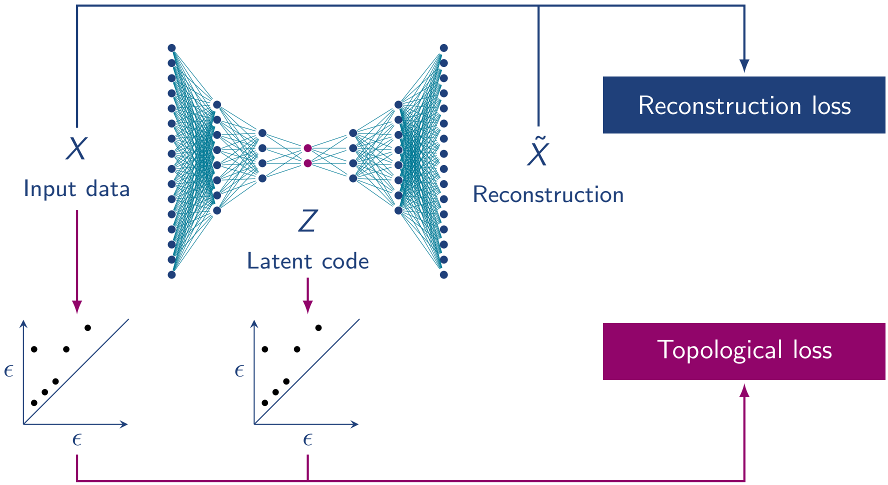
X
Input data
Z
Latent code
$\tilde{X}$
Reconstruction
Reconstruction loss

Overview



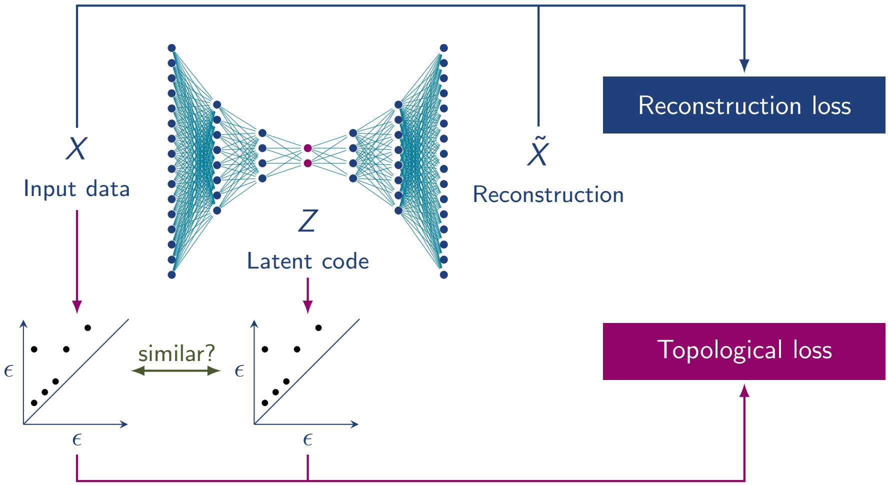
Reconstruction loss
X
Input data
Z
Latent code
$\tilde{X}$
Reconstruction
ϵ
ϵ
ϵ
ϵ

Overview



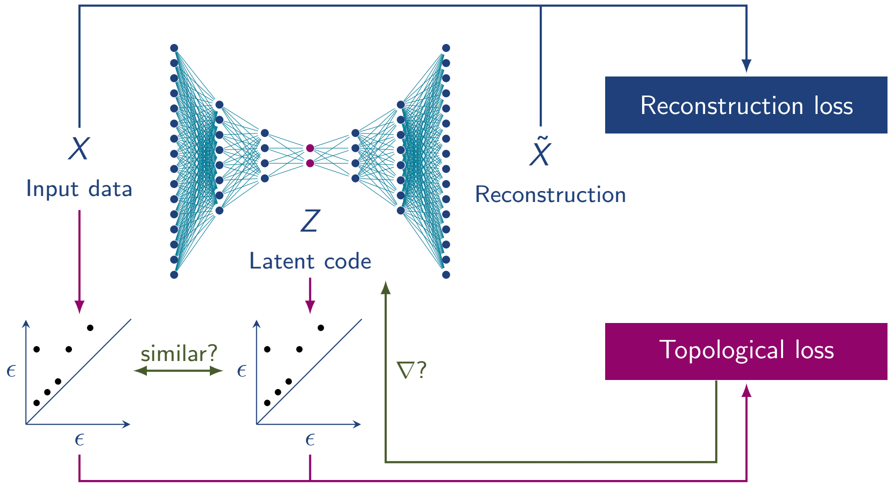
Reconstruction loss
X
Input data
$\tilde{X}$
Reconstruction
Z
Latent code
ϵ
ϵ
ϵ
ϵ
Topological loss

Overview



X
Input data
Z
Latent code
$\tilde{X}$
Reconstruction
Reconstruction loss
similar?
ϵ
ϵ
ϵ
ϵ
Topological loss

Overview



Reconstruction loss
X
Input data
Z
Latent code
$\tilde{X}$
Reconstruction
ϵ
ϵ
similar?
ϵ
ϵ
∇?
Topological loss

Proposed Method

- Given a point cloud X, we denote the persistent homology (PH) calculation of its Vietoris-Rips complex $\mathfrak{R}_\epsilon(X)$ as:

$$\left(\mathcal{D}^X, \pi^X\right) := \mathrm{PH}(\mathfrak{R}_\epsilon(X)) \tag{2}$$

where $\mathcal{D}^X$ refers to the resulting persistence diagram (0-dimensional for now), and π^X stands for the corresponding persistence *pairings*, i.e. the set of indices pointing to the subset of simplices in $\mathfrak{R}_\epsilon(X)$ which the PH calculation identified as topologically relevant.

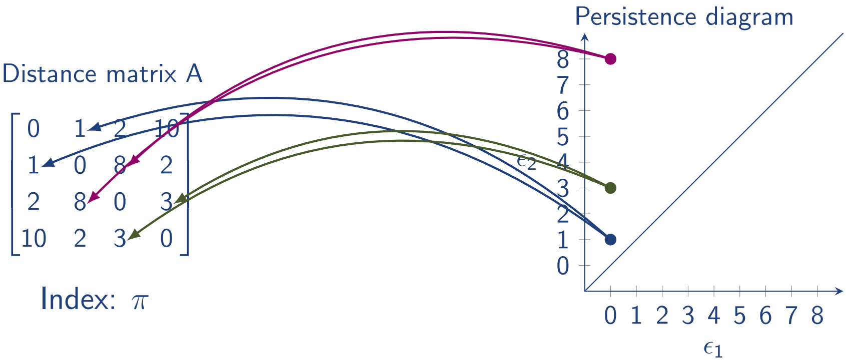
Proposed Method

- Introducing this persistence pairing π^X allows for a notational trick. We can access the values of the persistence diagram by selecting the corresponding entries in the pointcloud's distance matrix A^X.
- $A^X\left[\pi^X\right]$ is treated as a vector in $\mathbb{R}^{|\pi^X|}$.

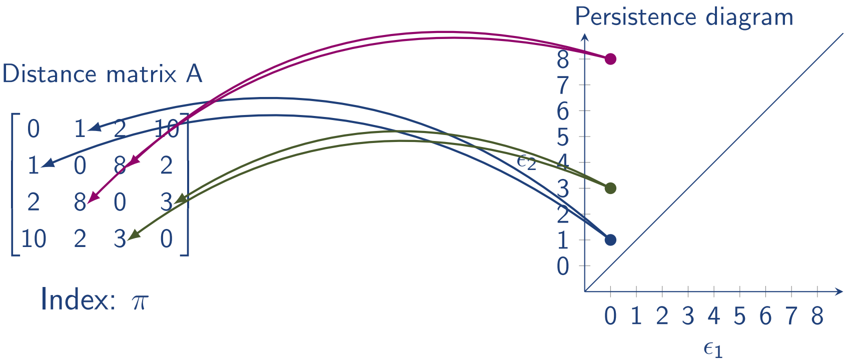
Distance matrix vs persistence diagram

Distance matrix A

$$\begin{bmatrix} 0 & 1 & 2 & 10 \\ 1 & 0 & 8 & 2 \\ 2 & 8 & 0 & 3 \\ 10 & 2 & 3 & 0 \end{bmatrix}$$


Distance matrix A
0 1 2 10
1 0 8 2
2 8 0 3
10 2 3 0
Index: π
Persistence diagram
ϵ_2
ϵ_1
0 1 2 3 4 5 6 7 8

Distance matrix vs persistence diagram



Notation:

A^X = distance matrix of mini-batch in data space

π^X = index set resulting from PH calculation in data space

$A^X\left[\pi^X\right]$ = vector of distances selected with π^X

Proposed Method

- Let X be a point cloud representing a mini-batch from the data space $\mathcal{X}$.
- Now we define an autoencoder as the composition of two functions $h \circ g$, where $g\colon \mathcal{X} \to \mathcal{Z}$ represents the *encoder* and $h\colon \mathcal{Z} \to \mathcal{X}$ represents the *decoder*. We denote latent codes with $Z := g(X)$.
- During a forward pass of the autoencoder, we compute the persistent homology of the mini-batch in both the data as well as the latent space, yielding the following set of tuples:

$$\left(\mathcal{D}^X, \pi^X\right) := \mathrm{PH}(\mathfrak{R}_\epsilon(X)) \quad \text{and} \quad \left(\mathcal{D}^Z, \pi^Z\right) := \mathrm{PH}(\mathfrak{R}_\epsilon(Z)) \tag{3}$$

Proposed Method

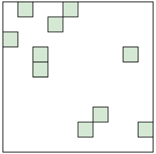
- Both diagrams $\mathcal{D}^X$ and $\mathcal{D}^Z$ are compared in order to construct a topological loss term $\mathcal{L}_t$
- We add $\mathcal{L}_t$ to the standard reconstruction loss term $\mathcal{L}_r$ to arrive at the following optimisation objective

$$\mathcal{L} = \mathcal{L}_r\big(X, h(g(X))\big) + \lambda\mathcal{L}_t, \tag{4}$$

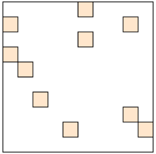
where $\lambda \in$ is a parameter to control the strength of the regularisation.

Proposed Method

- Before diving into the topological loss term, let's visualize *selected* distances:



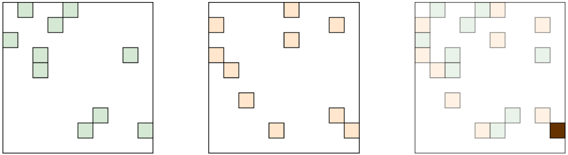
$A^X\left[\pi^X\right]$



$A^Z\left[\pi^Z\right]$

Proposed Method

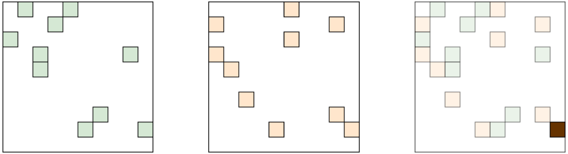
- Before diving into the topological loss term, let's visualize *selected* distances:



$A^X[\pi^X]$ $A^Z[\pi^Z]$ Intersection

- Problem: At the beginning, a randomly initialized latent space shows little overlap in terms of which distances are selected (1 in expectation). How to create a non-naive loss term that still matches the 'edges' in both spaces?

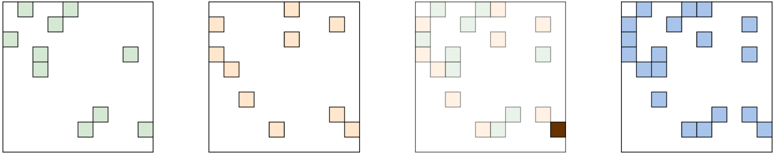
- Constraints:
  1. In the latent space, we wish to preserve the input topology as represented by $A^X\left[\pi^X\right]$
  2. Only A^Z depends on the autoencoder's parameters and leads to informative gradients.



(a) $A^X\left[\pi^X\right]$ **(b)** $A^Z\left[\pi^Z\right]$ **(c)** Intersection

Proposed Method

- Constraints:
  1. In the latent space, we wish to preserve the input topology as represented by $A^X\left[\pi^X\right]$
  2. Only A^Z depends on the autoencoder's parameters and leads to informative gradients.



(a) $A^X\left[\pi^X\right]$ **(b)** $A^Z\left[\pi^Z\right]$ **(c)** Intersection **(d)** Union

- We propose to consider the *union* of all selected distances / edges both in $A^X\left[\pi^X\right]$ and $A^Z\left[\pi^Z\right]$.

Proposed Method

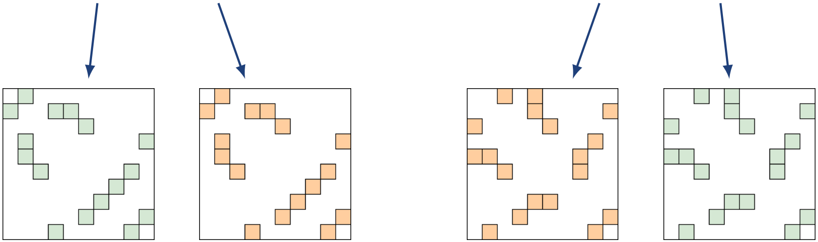
- To implement this, our topological loss term decomposes into two components, each handling the "directed" loss occurring as topological features in one of the two spaces, i.e. either the data space X or the latent code Z, remain fixed.
- We have $\mathcal{L}_t = \mathcal{L}_{\mathcal{X}\to\mathcal{Z}} + \mathcal{L}_{\mathcal{Z}\to\mathcal{X}}$, where

$$\mathcal{L}_{\mathcal{X}\to\mathcal{Z}} := \frac{1}{2}\left\|\mathrm{A}^X\left[\pi^X\right] - \mathrm{A}^Z\left[\pi^X\right]\right\|^2 \quad \text{and} \quad \mathcal{L}_{\mathcal{Z}\to\mathcal{X}} := \frac{1}{2}\left\|\mathrm{A}^Z\left[\pi^Z\right] - \mathrm{A}^X\left[\pi^Z\right]\right\|^2, \qquad (5)$$

Topological loss term

$$\mathcal{L}_t = \mathcal{L}_{\mathcal{X}\to\mathcal{Z}} + \mathcal{L}_{\mathcal{Z}\to\mathcal{X}}$$

$$\mathcal{L}_{\mathcal{X}\to\mathcal{Z}} := \frac{1}{2}\left\|\mathsf{A}^X\left[\pi^X\right] - \mathsf{A}^Z\left[\pi^X\right]\right\|^2 \qquad \mathcal{L}_{\mathcal{Z}\to\mathcal{X}} := \frac{1}{2}\left\|\mathsf{A}^Z\left[\pi^Z\right] - \mathsf{A}^X\left[\pi^Z\right]\right\|^2$$



Proposed Method

This topological loss term is differentiable under the following assumption:

Assumption

There is an infinitesimal neighbourhood around each point in a persistence diagram that only contains this single point. Thus, the corresponding persistence pairing π does not change upon a small perturbation of the underlying distances.

Gradient Derivation

- Letting $\boldsymbol{\theta}$ refer to the parameters of the *encoder*, we have

$$\frac{\partial}{\partial \boldsymbol{\theta}} \mathcal{L}_{\mathcal{X} \to \mathcal{Z}} = \frac{\partial}{\partial \boldsymbol{\theta}} \left(\frac{1}{2} \left\| \mathrm{A}^X \left[\pi^X \right] - \mathrm{A}^Z \left[\pi^X \right] \right\|^2 \right) = -\left(\mathrm{A}^X \left[\pi^X \right] - \mathrm{A}^Z \left[\pi^X \right] \right)^\top \left(\frac{\partial \mathrm{A}^Z \left[\pi^X \right]}{\partial \boldsymbol{\theta}} \right) \quad (6)$$

$$= -\left(\mathrm{A}^X \left[\pi^X \right] - \mathrm{A}^Z \left[\pi^X \right] \right)^\top \left(\sum_{i=1}^{\left| \pi^X \right|} \frac{\partial \mathrm{A}^Z \left[\pi^X \right]_i}{\partial \boldsymbol{\theta}} \right), \quad (7)$$

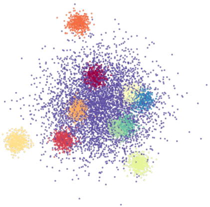
where $\left| \pi^X \right|$ denotes the cardinality of a persistence pairing and $\mathrm{A}^Z \left[\pi^X \right]_i$ refers to the *i*-th entry of the vector of paired distances.

Proposed Method: Stability

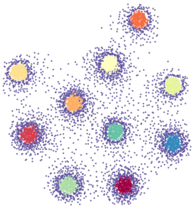
- We aim to capture topological features of the data and latent space. Yet, we only calculate topological features on the mini-batch level.
- In two theorems, we address whether this is approximation is stable:
  1. In Theorem 1, we show that the bottleneck distance between persistence diagrams of a point cloud X and its subsample X^m of m points is bounded by the Hausdorff distance between X and X^m.
  2. In Theorem 2, we derive an upper bound of the expected Hausdorff distance between X and X^m.

Experiments

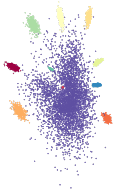
Spheres



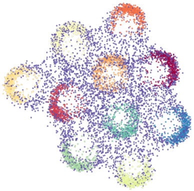
PCA



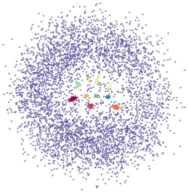
t-SNE



Autoencoder



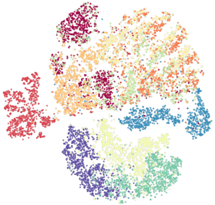
UMAP



Topo-AE

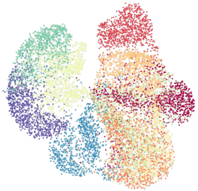
FashionMNIST Xiao et al. [2017]

PCA



t-SNE

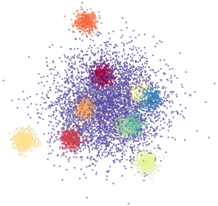
Autoencoder



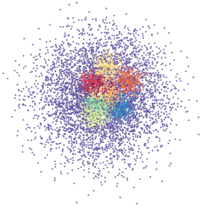
UMAP

Topo-AE

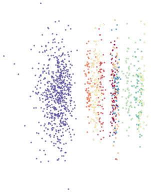
Further architectures



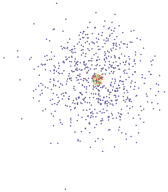
PCA



TopoPCA



VAE



Topo-VAE

Insights and Summary

- Novel method for preserving topological information of the input space in dimensionality reduction

Insights and Summary

- Novel method for preserving topological information of the input space in dimensionality reduction
- Under weak theoretical assumptions our topological loss term is differentiable and permits the training of neural networks via backpropagation.

Insights and Summary

- Novel method for preserving topological information of the input space in dimensionality reduction
- Under weak theoretical assumptions our topological loss term is differentiable and permits the training of neural networks via backpropagation.
- Approximating topological features on the mini-batch level is robust.

Insights and Summary

- Novel method for preserving topological information of the input space in dimensionality reduction
- Under weak theoretical assumptions our topological loss term is differentiable and permits the training of neural networks via backpropagation.
- Approximating topological features on the mini-batch level is robust.
- Our method was uniquely able to capture spatial relationships of nested high-dimensional spheres

Insights and Summary

- Novel method for preserving topological information of the input space in dimensionality reduction
- Under weak theoretical assumptions our topological loss term is differentiable and permits the training of neural networks via backpropagation.
- Approximating topological features on the mini-batch level is robust.
- Our method was uniquely able to capture spatial relationships of nested high-dimensional spheres
- The proposed loss term is highly generic, can be employed in various architectures, and merely requires distances between data objects.

Outlook

- A current bottleneck for many PH-based approaches in ML is to scale up the dimensionality of the persistence calculation. This could be achieved with approximations or parallelism.
- Applications, where the structure of high-dimensional data is relevant but currently hard to recover, e.g. in the life sciences.
- Topological data analysis (TDA) is officially "taking off" in the ML community, with the first Neurips 2020 Workshop Topological Data Analysis and Beyond!

Further TDA projects from (or with) our lab

- Graph Filtration Learning *(ICML 2020)*
- A Persistent Weisfeiler–Lehman Procedure for Graph Classification *(ICML 2019)*

For further information, please check out our

Paper:

https://arxiv.org/abs/1906.00722

Code:

Credits:

- Aleph for TDA calculations https://github.com/Pseudomanifold/Aleph
- manim for animations https://github.com/3b1b/manim

References

H. Edelsbrunner and J. Harer. Persistent homology—a survey. In J. E. Goodman, J. Pach, and R. Pollack, editors, *Surveys on discrete and computational geometry: Twenty years later*, number 453 in Contemporary Mathematics, pages 257–282. American Mathematical Society, Providence, RI, USA, 2008.

M. Moor, M. Horn, B. Rieck, and K. Borgwardt. Topological autoencoders. *arXiv preprint arXiv:1906.00722*, 2019.

B. Rieck, M. Togninalli, C. Bock, M. Moor, M. Horn, T. Gumbsch, and K. Borgwardt. Neural persistence: A complexity measure for deep neural networks using algebraic topology. *arXiv preprint arXiv:1812.09764*, 2018.

L. Vietoris. Über den höheren Zusammenhang kompakter Räume und eine Klasse von zusammenhangstreuen Abbildungen. *Mathematische Annalen*, 97(1):454–472, 1927.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Appendix

Bound of bottleneck distance between persistence diagrams on subsampled data

Theorem

Let X be a point cloud of cardinality n and $X^{(m)}$ be one subsample of X of cardinality m, i.e. $X^{(m)} \subseteq X$, sampled without replacement. We can bound the probability of the persistence diagrams of $X^{(m)}$ exceeding a threshold in terms of the bottleneck distance as

$$\mathbb{P}\left(\mathsf{d_b}\left(\mathcal{D}^X, \mathcal{D}^{X^{(m)}}\right) > \epsilon\right) \leq \mathbb{P}\left(\mathsf{d_H}\left(X, X^{(m)}\right) > 2\epsilon\right),$$

where $\mathsf{d_H}$ refers to the Hausdorff distance between the point cloud and its subsample.

Expected value of Hausdorff distance

Theorem

Let $\mathrm{A} \in^{n \times m}$ *be the distance matrix between samples of* X *and* $X^{(m)}$*, where the rows are sorted such that the first* m *rows correspond to the columns of the* m *subsampled points with diagonal elements* $a_{ii} = 0$*. Assume that the entries* a_{ij} *with* $i > m$ *are random samples following a distance distribution* F_D *with* $\mathrm{supp}(f_D) \in_{\geq 0}$*. The minimal distances* δ_i *for rows with* $i > m$ *follow a distribution* F_Δ*. Letting* $Z := \max_{1 \leq i \leq n} \delta_i$ *with a corresponding distribution* F_Z*, the expected Hausdorff distance between* X *and* $X^{(m)}$ *for* $m < n$ *is bounded by:*

$$\mathbb{E}\left[\mathrm{d_H}(X, X^{(m)})\right] = \mathbb{E}_{Z \sim F_Z}[Z] \leq \int_0^{+\infty} \left(1 - F_D(z)^{(n-1)}\right) \mathrm{d}z \leq \int_0^{+\infty} \left(1 - F_D(z)^{m(n-m)}\right) \mathrm{d}z$$

Density distribution error

Definition (Density distribution error)

Let $\sigma \in_{>0}$. For a finite metric space $\mathcal{S}$ with an associated distance $\operatorname{dist}(\cdot, \cdot)$, we evaluate the density at each point $x \in \mathcal{S}$ as

$$\mathrm{f}_\sigma^{\mathcal{S}}(x) := \sum_{y \in \mathcal{S}} \exp\left(-\sigma^{-1} \operatorname{dist}(x, y)^2\right),$$

where we assume without loss of generality that $\max \operatorname{dist}(x, y) = 1$. We then calculate $\mathrm{f}_\sigma^X(\cdot)$ and $\mathrm{f}_\sigma^Z(\cdot)$, normalise them such that they sum to 1, and evaluate

$$\mathrm{KL}_\sigma := \mathrm{KL}\left(\mathrm{f}_\sigma^X \parallel \mathrm{f}_\sigma^Z\right), \tag{8}$$

i.e. the Kullback–Leibler divergence between the two density estimates.

Quantification of performance

Data set	Method	$KL_{0.01}$	$KL_{0.1}$	KL_{1}	ℓ-MRRE	ℓ-Cont	ℓ-Trust	ℓ-RMSE	Data MSE
Spheres	Isomap	0.181	**0.420**	**0.00881**	**0.246**	**0.790**	**0.676**	10.4	–
	PCA	0.332	0.651	0.01530	0.294	0.747	0.626	11.8	0.9610
	TSNE	**0.152**	0.527	0.01271	**<u>0.217</u>**	0.773	**<u>0.679</u>**	**<u>8.1</u>**	–
	UMAP	0.157	0.613	0.01658	0.250	0.752	0.635	**9.3**	–
	AE	0.566	0.746	0.01664	0.349	0.607	0.588	13.3	**<u>0.8155</u>**
	TopoAE	**<u>0.085</u>**	**<u>0.326</u>**	**<u>0.00694</u>**	0.272	**<u>0.822</u>**	0.658	13.5	**0.8681**
F-MNIST	PCA	**<u>0.356</u>**	**<u>0.052</u>**	**<u>0.00069</u>**	0.057	0.968	0.917	**<u>9.1</u>**	0.1844
	TSNE	0.405	0.071	0.00198	**<u>0.020</u>**	0.967	**0.974**	41.3	–
	UMAP	0.424	0.065	0.00163	0.029	**<u>0.981</u>**	0.959	**13.7**	–
	AE	0.478	0.068	0.00125	**0.026**	0.968	**<u>0.974</u>**	20.7	**<u>0.1020</u>**
	TopoAE	**0.392**	**0.054**	**0.00100**	0.032	**0.980**	0.956	20.5	**0.1207**
MNIST	PCA	0.389	0.163	0.00160	0.166	0.901	0.745	**<u>13.2</u>**	0.2227
	TSNE	**<u>0.277</u>**	**0.133**	0.00214	**<u>0.040</u>**	0.921	**<u>0.946</u>**	22.9	–
	UMAP	**0.321**	0.146	0.00234	**0.051**	**<u>0.940</u>**	**0.938**	**14.6**	–
	AE	0.620	0.155	**0.00156**	0.058	0.913	0.937	18.2	**<u>0.1373</u>**
	TopoAE	0.341	**<u>0.110</u>**	**<u>0.00114</u>**	0.056	**0.932**	0.928	19.6	**0.1388**

Quantification of performance - 2

Data set	Method	$\mathrm{KL}_{0.01}$	$\mathrm{KL}_{0.1}$	KL_{1}	ℓ-MRRE	ℓ-Cont	ℓ-Trust	ℓ-RMSE	Data MSE
	PCA	**0.591**	**0.020**	<u>**0.00023**</u>	0.119	<u>**0.931**</u>	0.821	<u>**17.7**</u>	0.1482
	TSNE	0.627	0.030	0.00073	<u>**0.103**</u>	0.903	**0.863**	**25.6**	–
CIFAR	UMAP	0.617	0.026	0.00050	0.127	0.920	0.817	33.6	–
	AE	0.668	0.035	0.00062	0.132	0.851	<u>**0.864**</u>	36.3	**0.1403**
	TopoAE	<u>**0.556**</u>	<u>**0.019**</u>	**0.00031**	**0.108**	**0.927**	0.845	37.9	<u>**0.1398**</u>