Discovering the underlying topological space of extracted features

Hamid Karim[1] Harrison LeFrois[2]

[1]Department of Computer Science and Engineering
Michigan State University

[2]Department of Mathematics
Michigan State University

April 2017

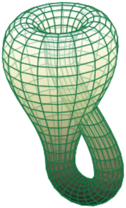
Table of Contents

The Big Problem

What does our data look like?

Source: www.gnuplotting.org/klein-bottle/

The Big Problem

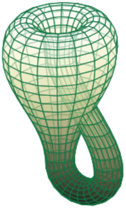


What does our data look like?

- Data is maybe from a manifold?

Source: www.gnuplotting.org/klein-bottle/

The Big Problem

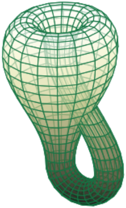


What does our data look like?

- Data is maybe from a manifold?
- Manifold learning makes this assumption

Source: www.gnuplotting.org/klein-bottle/

The Big Problem

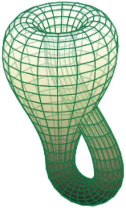


What does our data look like?

- Data is maybe from a manifold?
- Manifold learning makes this assumption
- Question 1: What if our data isn't from a manifold? Could the features be parametrized by a manifold?

Source: www.gnuplotting.org/klein-bottle/

The Big Problem



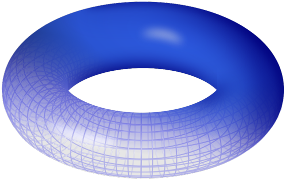
What does our data look like?

- Data is maybe from a manifold?
- Manifold learning makes this assumption
- Question 1: What if our data isn't from a manifold? Could the features be parametrized by a manifold?
- Question 2: Neural networks often become a black box. What is happening inside?

Source: www.gnuplotting.org/klein-bottle/

Why do we care?

Source:
https://en.wikipedia.org/wiki/Torus

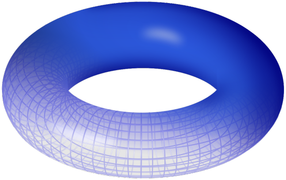


Why do we care?

- Manifolds have nice mathematical structure (locally Euclidean, lots of structure, etc)

Source: https://en.wikipedia.org/wiki/Torus

The Big Problem

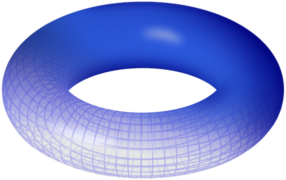


Why do we care?

- Manifolds have nice mathematical structure (locally Euclidean, lots of structure, etc)
- Can assign coordinates on the manifold rather than in $\mathbb{R}^n$

Source: https://en.wikipedia.org/wiki/Torus

The Big Problem



Why do we care?

- Manifolds have nice mathematical structure (locally Euclidean, lots of structure, etc)
- Can assign coordinates on the manifold rather than in $\mathbb{R}^n$
- This might help with prediction!

Source: https://en.wikipedia.org/wiki/Torus

Topology and Machine Learning

- Persistent homology: used to deduce topological structure from data points. Comes from Topological Data Analysis (TDA)
- Convolutional Neural Network (CNN): Great for feature extraction from images

Quick Topology Intro

Concerned with properties that don't change under continuous deformations

- Like stretching, shrinking, and bending
- But NOT cutting, gluing, etc.

Example: Circle $=$ Square $=$ Triangle

Source:
https://www.shapeways.com/blog/archives/
a-3d-printed-topology-joke.html

Quick Topology Intro

Homology is a sequence of groups (an algebraic structure) that keeps track of "holes" and their higher dimensional analogues. So if X is a topological space, then:

- $H_0(X)$ counts the number of connected pieces
- $H_1(X)$ counts the number of loops
- $H_2(X)$ counts the number of "voids" (like the inside of a beach ball)

Inspiration

Persistent homology of rotations of everyday objects

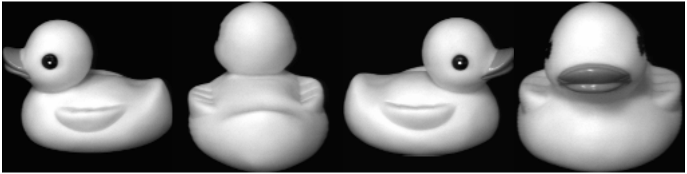
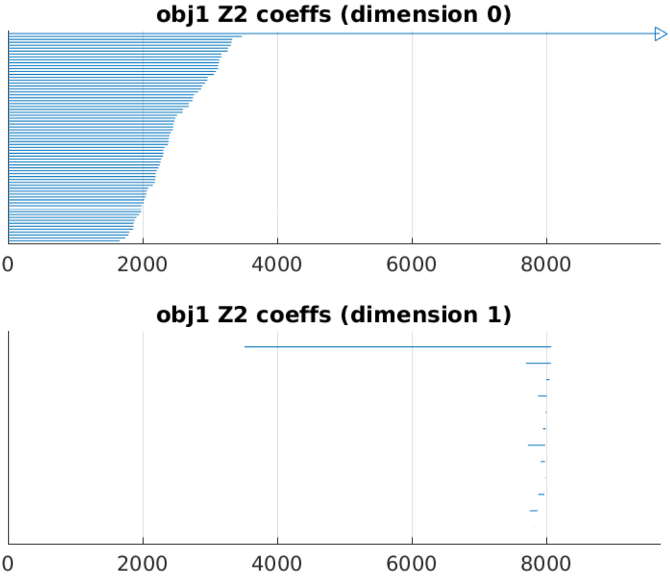


Figure: Rotation of rubber duck

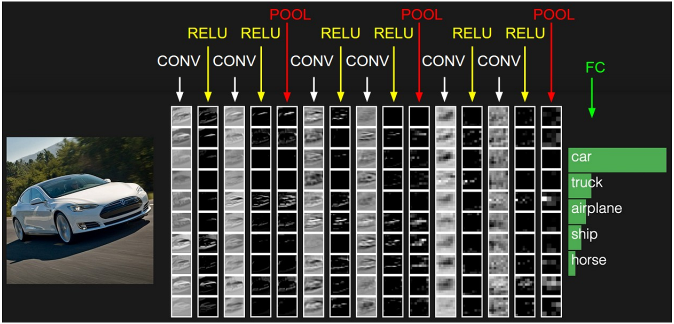
Inspiration



Our Data

Needed thousands of photos to train a neural network - so we used the CASIA-Webface dataset. It is a public dataset consisting of of 494,414 images [2].

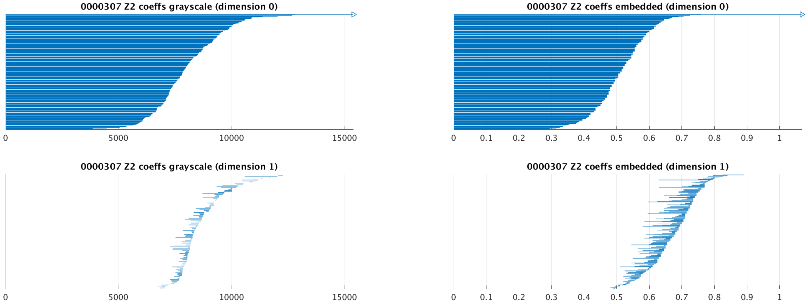
Convolutional neural network



Architecture

Name	Type	Filter Size /Stride	Output size	Depth	#Params
Conv11	convolution	3×3 / 1	100×100×32	1	0.28K
Conv12	convolution	3×3 / 1	100×100×64	1	18K
Pool1	max pooling	2×2 / 2	50×50×64	0	
Conv21	convolution	3×3 / 1	50×50×64	1	36K
Conv22	convolution	3×3 / 1	50×50×128	1	72K
Pool2	max pooling	2×2 / 2	25×25×128		
Conv31	convolution	3×3 / 1	25×25×96	1	108K
Conv32	convolution	3×3 / 1	25×25×192	1	162K
Pool3	max pooling	2×2 / 2	13×13×192	0	
Conv41	convolution	3×3 / 1	13×13×128	1	216K
Conv42	convolution	3×3 / 1	13×13×256	1	288K
Pool4	max pooling	2×2 / 2	7×7×256	0	
Conv51	convolution	3×3 / 1	7×7×160	1	360K
Conv52	convolution	3×3 / 1	7×7×320	1	450K
Pool5	avg pooling	7×7 / 1	1×1×320		
Dropout	dropout (40%)		1×1×320	0	
Fc6	fully connection		10575	1	3305K
Cost1	softmax		10575	0	
Cost2	contrastive		1	0	
Total				11	5015K

Helena Bonham Carter - 254 images - Original features on left (pixels), 128 features on right (embedding from CNN)



New loss function

- We are basically interested in function f which embeds our data X in d-dimensional Euclidean space.
  $\mathcal{C}^i \leftarrow$ Data with same labels as i-th instance
  $C^i \leftarrow$ centroid of $\mathcal{C}^i$ in embedded space

$$
\begin{aligned}
\text{[1]Previous,} \quad L_{i,j,k} =& \lambda d(f^i, f^j) + (1-\lambda)[1 + d(f^i, f^j) - d(f^i, f^j)]_+ \\
\text{Current,} \quad L_{i,j,k} =& d(f^i, f^j) + [1 + d(f^i, f^j) - d(f^i, f^k)]_+ \\
& + d(f^i, C^i) + d(f^j, C^i) + d(f^k, C^k) \\
& + [1 - d(C^i, C^k)]_+ \\
\text{such that,} \quad C^i =& \frac{1}{|\mathcal{C}^i|} \sum_{s \in \mathcal{C}^i} f^s \quad \text{centroid in embedded space}
\end{aligned}
$$

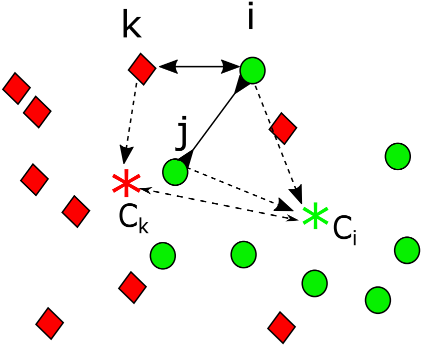
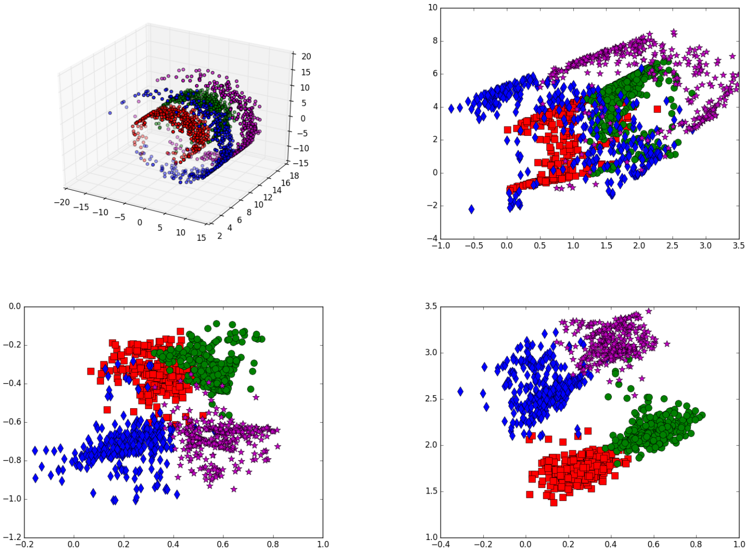


Figure: Graphical representation of loss function in the embedded space

Results on a toy data



Conclusions and Future Work

Conclusions thus far:

- From homology perspective, data appears to be connected with no loops
- Maybe other structure - like a dendrite? Clusters?

Future work:

- Apply more tools from TDA (mapper algorithm - check for dendrite structures)
- Other standard datasets (MNIST, natural images, spoken language) with appropriate NN architectures (CNN, RNN, etc)
- Scale space instead of pixel space for images

Bibliography

- Florian Schroff, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015, pp. 815–823.
- Dong Yi et al. "Learning Face Representation from Scratch". In: *CoRR* abs/1411.7923 (2014). URL: http://arxiv.org/abs/1411.7923.