Testing the Manifold Hypothesis: An Information Theoretic Approach
Adrian Lopez - PRUV Summer Research
Duke University

Introduction

High dimensional data is prevalent everywhere, with significant examples being present in speech, images, and text. Deep neural networks (DNNs) have recently been receiving a great deal of excitement due to breakthroughs in being able to "learn" high performing functions between high dimensional spaces such as in computer vision for image classification or natural language processing. The main theory surrounding its success stems from the manifold hypothesis in which it is believed that although the dataset lies in a high dimensional space, much of the relevant structure of the data (i.e. the degrees of freedom or support of the data) lies along a low dimensional manifold [1][2], and many experiments have been conducted suggesting this is the case [1]. In this hypothesis, it is believed that the reason why DNNs are able to perform exceedingly well in creating functions between high dimensional spaces is in large part due to the hidden layers being able to "disentangle" highly tangled manifolds lying in input space to flattened manifolds in its hidden space (i.e. in its hidden layers)[2].

Many studies suggest that depth, which is determined by the number of layers in a network, is key to the high performance of DNNs [3][4][5][6], more so than width such as in [3] where they show that shallow networks are able to approximate a class of functions requiring an exponential number of neurons, whereas deep networks only require a polynomial number of neurons and [6] where they show that deep networks are better able to create more complex decision boundaries for spaces with more topological complexity in the form of a higher sum of Betti numbers than shallow networks. To test this hypothesis, we first created a sampled dataset from different manifolds and aimed to see how this sample dataset got transformed as it moved through the layers of the DNN. We used techniques from topological data analysis (TDA) and information theory to qualitatively and quantitatively measure this movement.

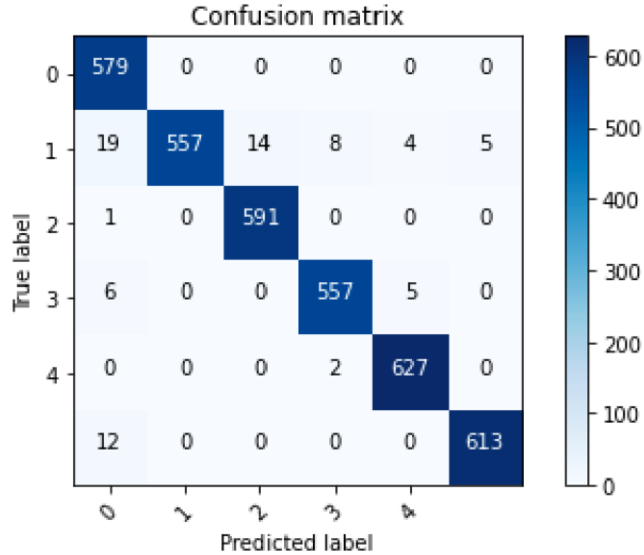
Method

Dataset: To test this experiment we generated a dataset of 6 different randomly sampled manifolds with different degrees of freedom living in 32-dimensional space with added noise, where each training sample consisted of 1,000 randomly sampled points along the surface of a manifold. This was in order to simulate an embedded manifold living in a high dimensional space, which would be the case as in the manifold hypothesis. The 5 different manifolds consisted of ellipses, ellipsoids, tori, linked ellipses, helices, and an 8-dimensional elliptical manifold to exhibit a peculiar structure (See Appendix A1 for a more detailed explanation), and the dataset consisted of 3000 of each type of manifold, i.e. 3000 helices images, etc. ***TDA***: We also calculated the 1D persistent homology for all of these manifolds and the 2D persistent homology for the ellipsoids in order to be able to qualitative see how the topology changes of the manifolds as they go through the DNN. ***DNN:*** Our DNN was composed of 3 convolutional layers, each with kernel size 4 and a dense layer with the convolutional layers having 128, 64, and 32 filters respectively and the final dense layer having 6 nodes used for classification. The convolutional layers had a Leaky ReLU activation function with alpha parameter 0.1, and the dense layer had a SoftMax activation function. We trained these networks with a batch size of 20, a learning rate of 1e-6, and a sparse categorical cross-entropy loss function. ***Mutual Information Plots:*** The mutual information plots were generated in similar accordance to [7], and to run the experiment, the same network architecture was trained except leaving out the ellipses and the mutual information of the layers during training was recorded. After this finished, we then ran this again this time adding back the ellipses and leaving out the linked ellipses and the mutual information was again recorded and displayed below.

Results:

A: Training the Model

After training the DNN, we got the following performance.

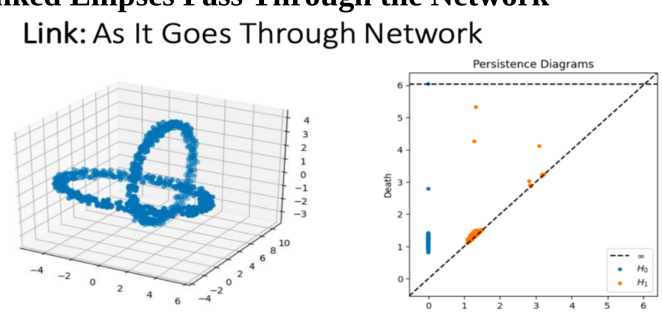


In this figure above, label 0 were the ellipses, label 1 were the ellipsoids, label 2 were the tori, label 3 were the 8D links, label 4 were the helices, and label 5 (the last row and column) were the linked circles. Note most of the error lies in classification of the sphere which is interesting since it is the only manifold with significant 2D persistent homology.

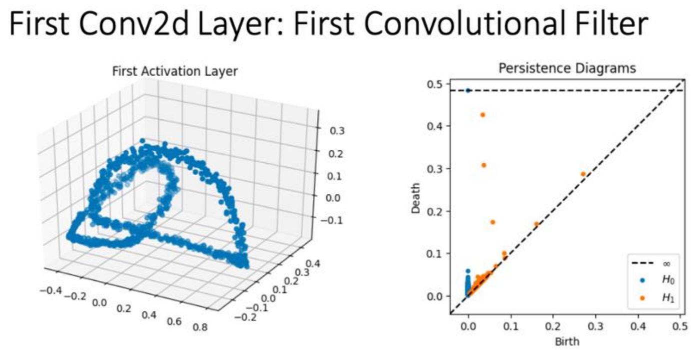
B: Watching Manifolds Pass Through the Network

Since we had a high number of filters in our model, testing to see how the data looked as it passed through all of them would have been unrealistic and repetitive. As such, it was decided to see how the data looked like as it passed through the first and final filter of each layer. We sought to see specifically 3 manifolds of interest, the linked ellipses to see the progression of a tangled manifolds in input space moving through the layers, the ellipses to be able to compare them to the linked ellipses(i.e. as a control) and the ellipsoids since they were responsible for most of the classification error that the model had.

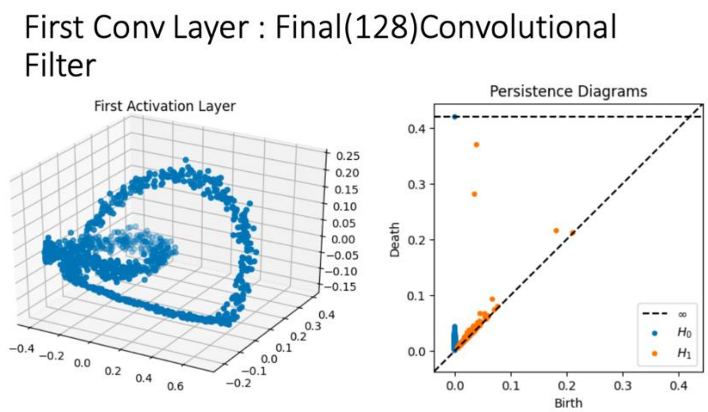
B1: Watching the Linked Ellipses Pass Through the Network



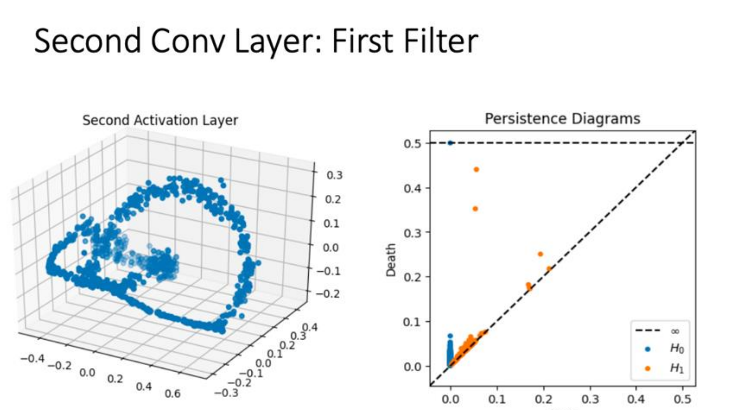
Linked Ellipses in Input Space(left). 0D (blue) and 1D(orange) persistent homology of the linked ellipses(right).



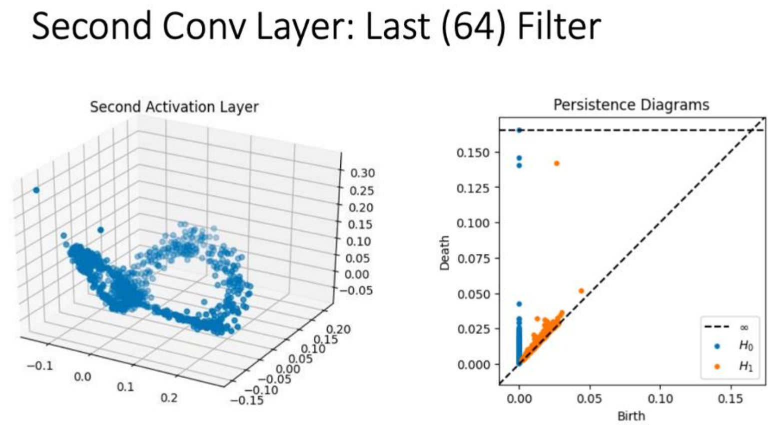
Linked Ellipses after passing through the first convolutional filter in the first layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).



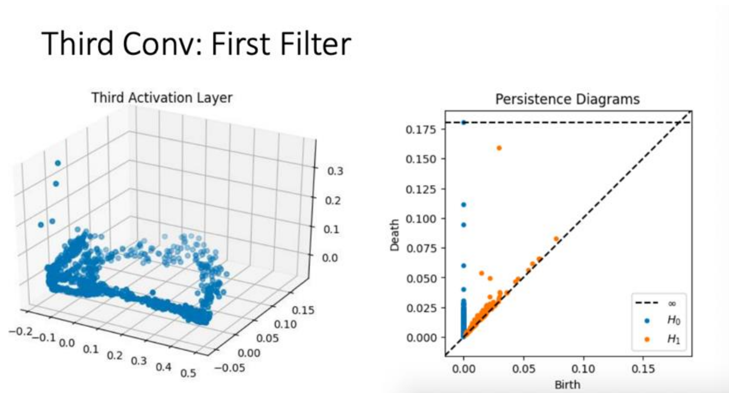
Linked Ellipses after passing through the 128th convolutional filter in the first layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).



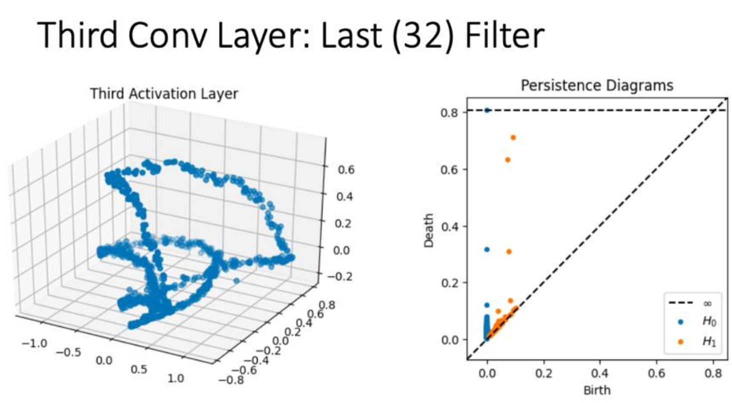
Linked Ellipses after passing through the first convolutional filter in the second layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).



Linked Ellipses after passing through the 64th convolutional filter in the second layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).

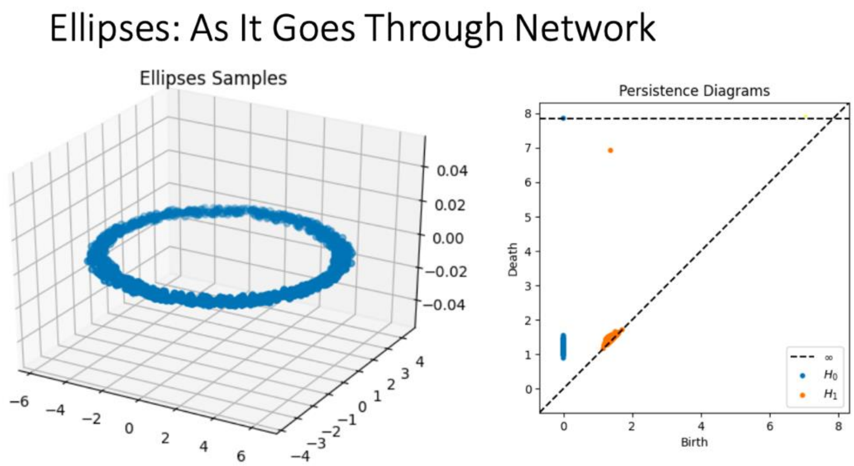


Linked Ellipses after passing through the first convolutional filter in the third layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).

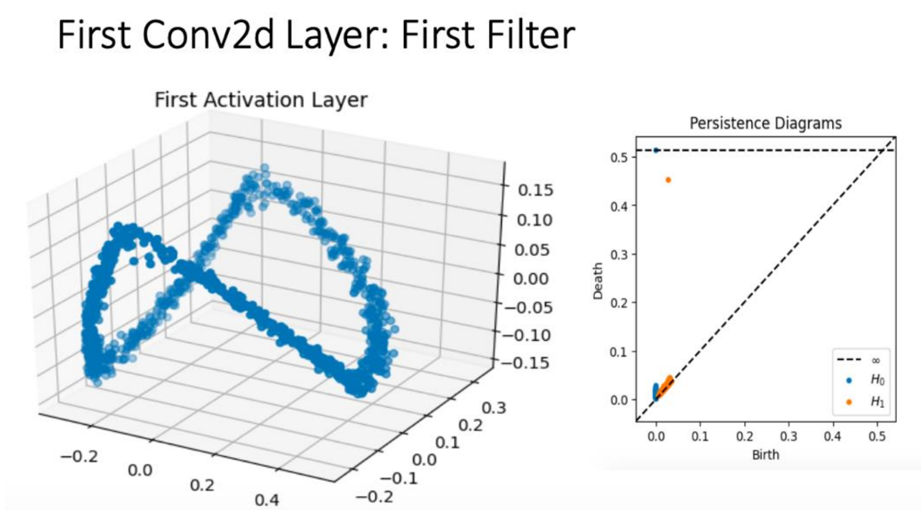


Linked Ellipses after passing through the last convolutional filter in the third layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).

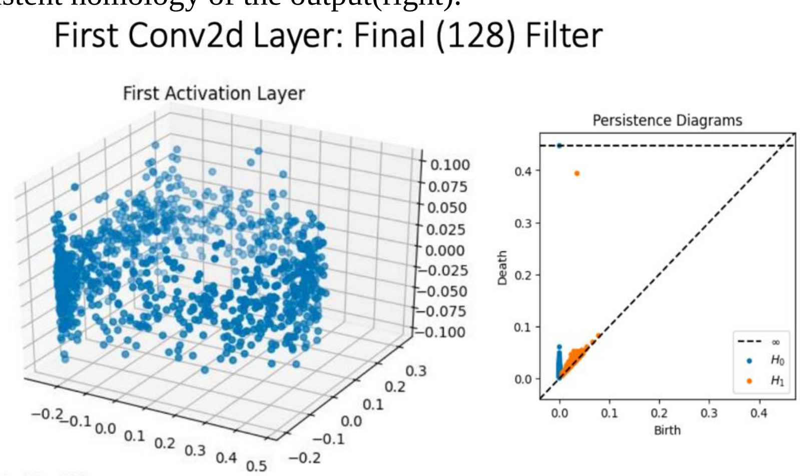
B2: Watching the Ellipses Pass Through the Network



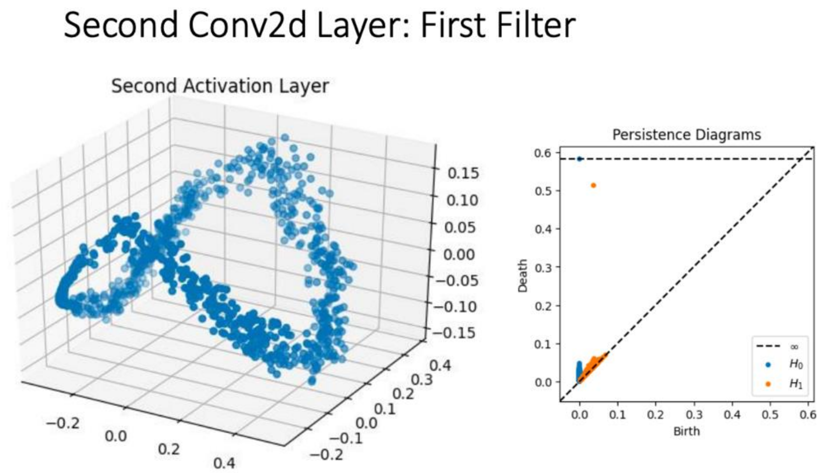
Ellipses in Input Space(left). 0D (blue) and 1D(orange) persistent homology of the linked ellipses(right).



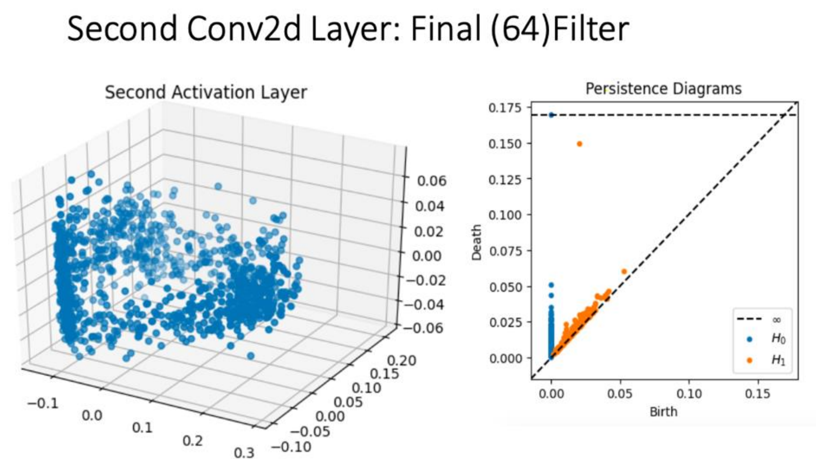
Ellipses after passing through the first convolutional filter in the first layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).



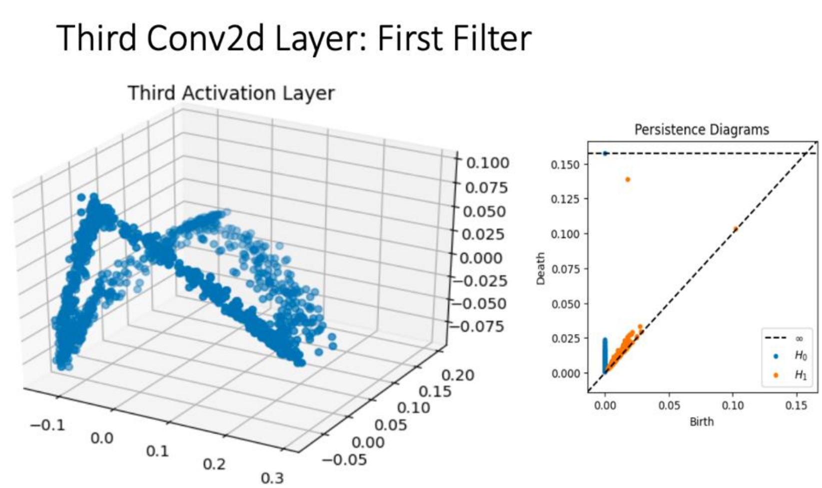
Ellipses after passing through the 128^{th} convolutional filter in the first layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).



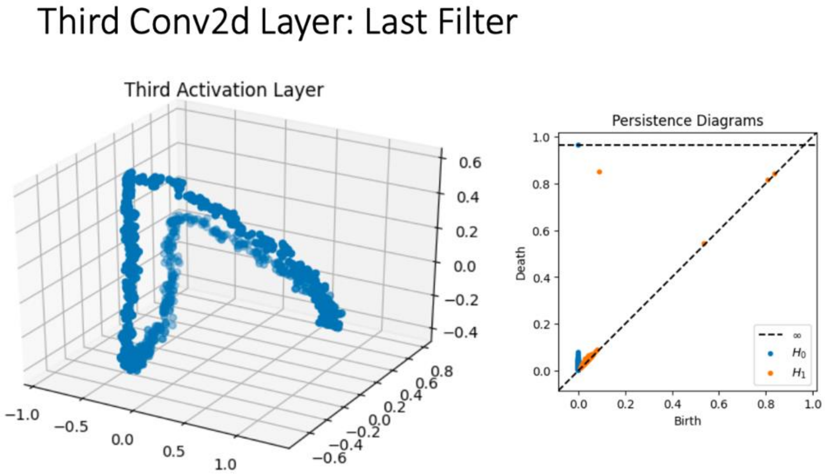
Ellipses after passing through the first convolutional filter in the second layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).



Ellipses after passing through the 64th convolutional filter in the second layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).

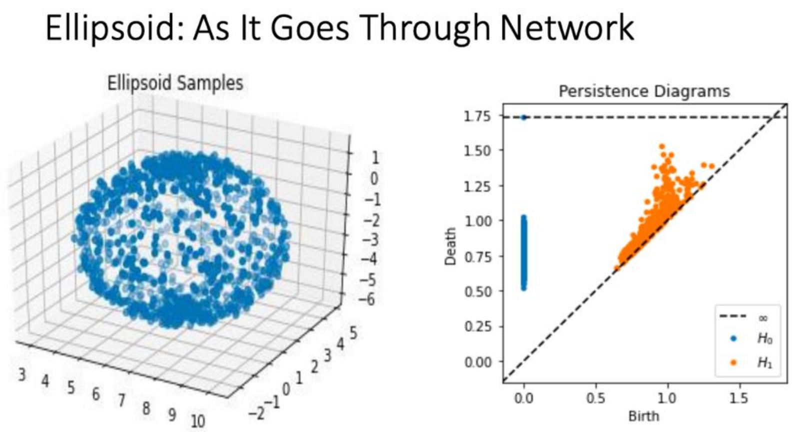


Ellipses after passing through the first convolutional filter in the third layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).

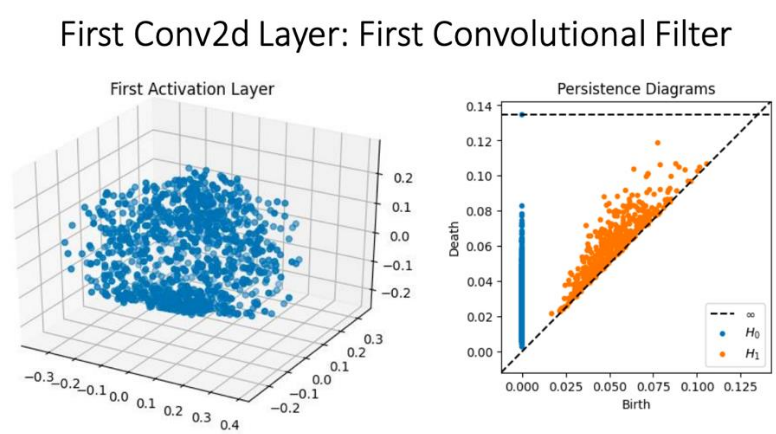


Ellipses after passing through the 32th convolutional filter in the third layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).

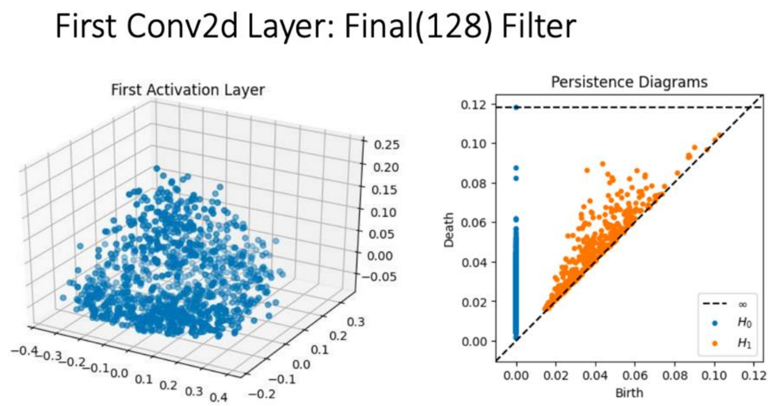
B3: Watching the Ellipsoids Pass Through the Network



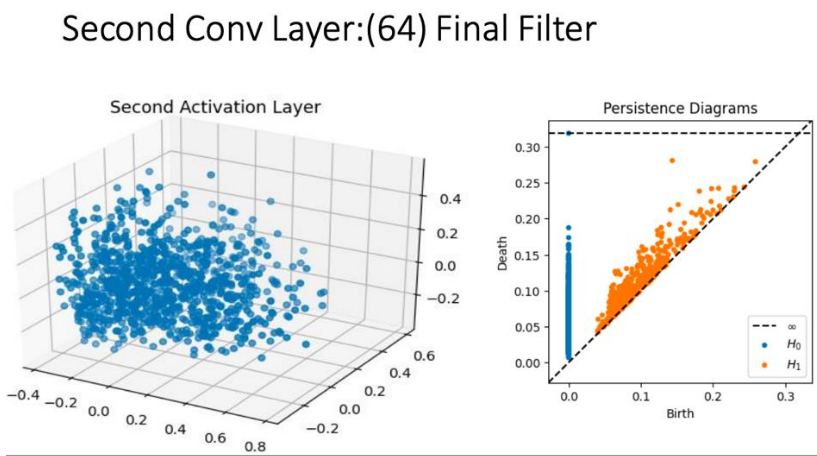
Ellipsoids in Input Space(left). 0D (blue) and 1D(orange) persistent homology of the linked ellipses(right).



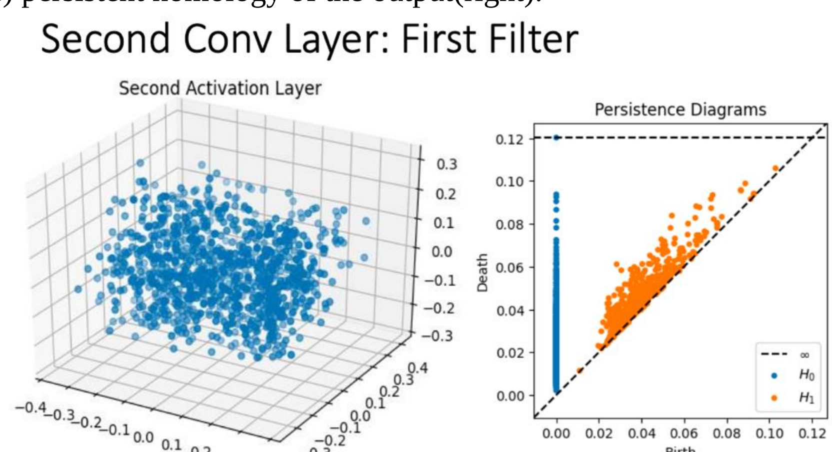
Ellipsoids after passing through the first convolutional filter in the first layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).



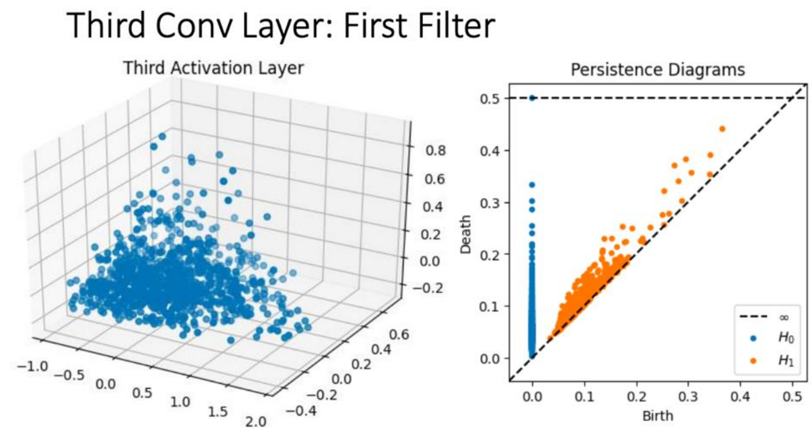
Ellipsoids after passing through the 128th convolutional filter in the first layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).



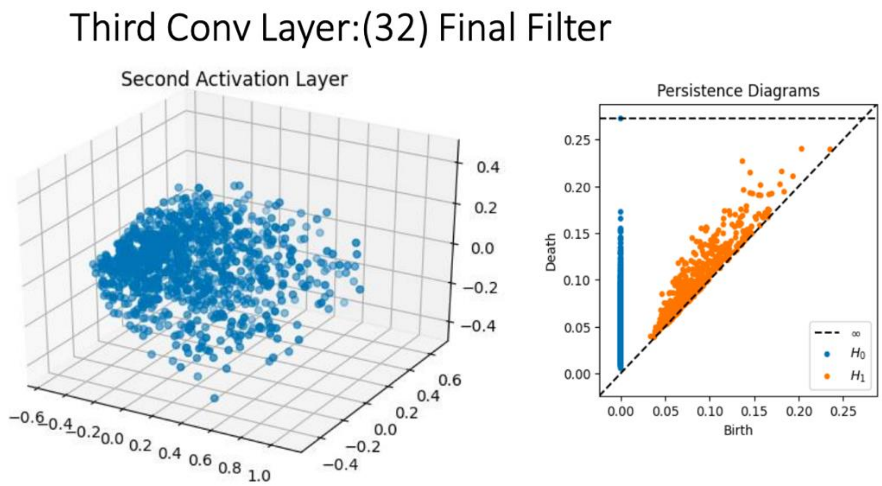
Ellipsoids after passing through the first convolutional filter in the second layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).



Ellipsoids after passing through the 64th convolutional filter in the first layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).

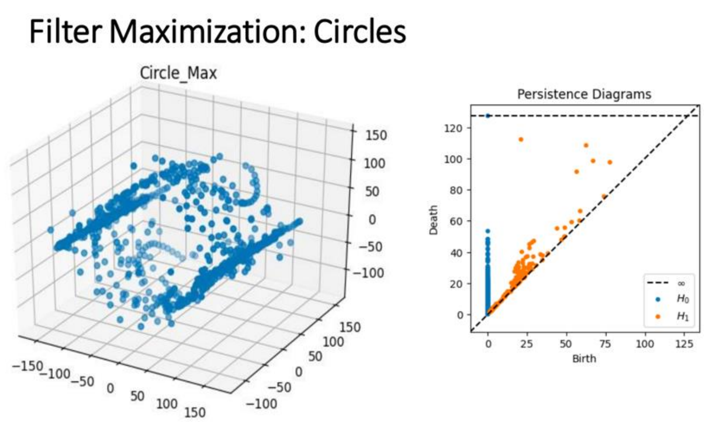


Ellipsoids after passing through the first convolutional filter in the third layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).

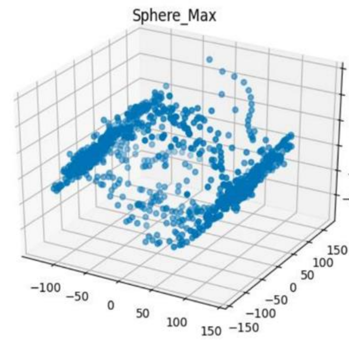


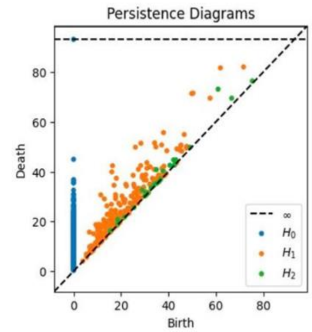
Ellipsoids after passing through the 32th convolutional filter in the third layer (left). 0D (blue) and 1D(orange) persistent homology of the output(right).

C: Filter Maximization

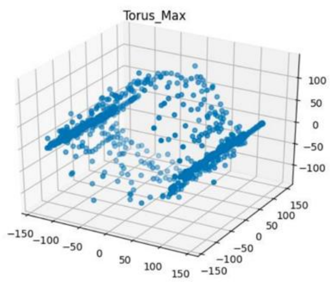


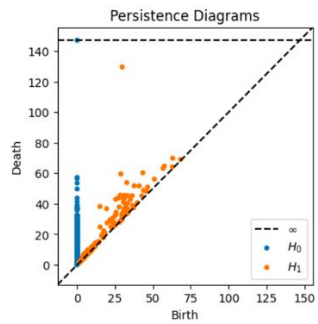
Filter Maximization: Ellipsoids



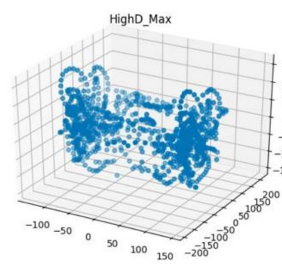


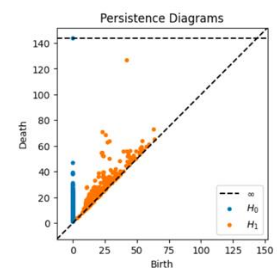
Torus



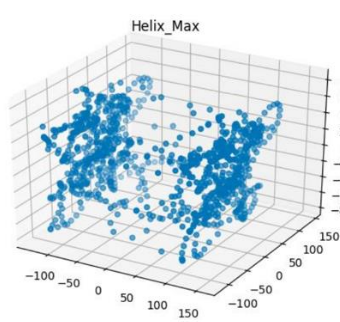


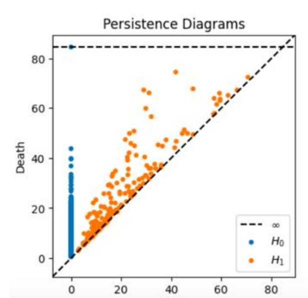
High D Shape

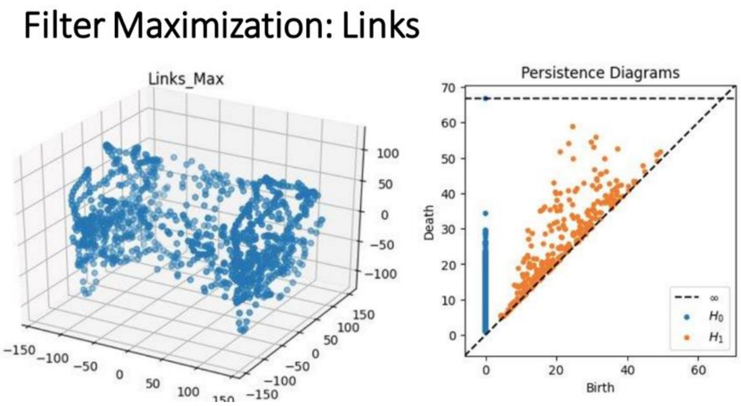




Helix

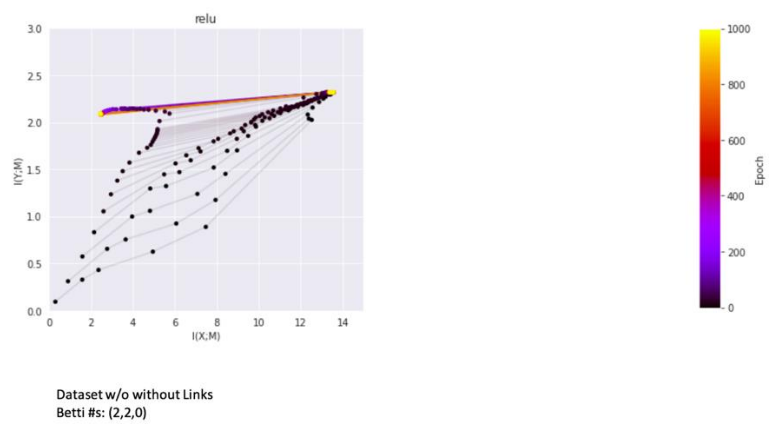




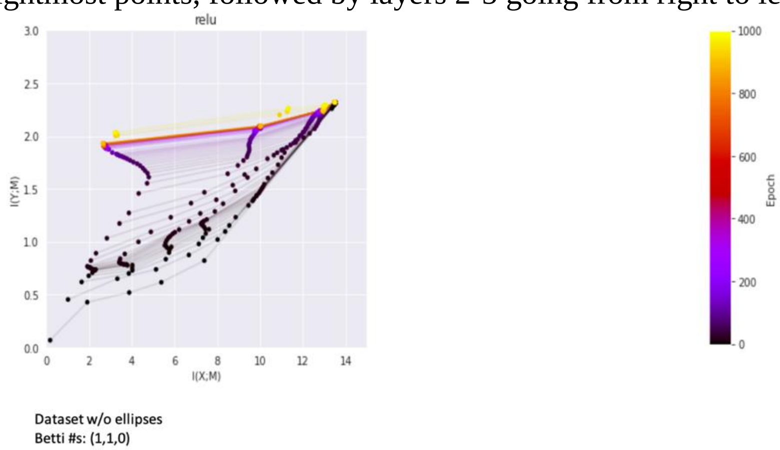


As expected for filter maximization, these filters were trained to look for parts of the manifold that were unique to shape. Looking at the High-D case as an example, you can see that the filters were expecting a highly curved shape having a large 1st betti number which resembles the topology of the manifold.

D: Mutual Information Plots



Mutual information plot of dataset not included the linked ellipsoids. Note that the first layer is depicted on the rightmost points, followed by layers 2-5 going from right to left.



Mutual information plot of dataset not included the ellipsoids.

Discussion:

As you can clearly see from all of these plots, the structure of the manifolds changes as they go through the layers. The plots in section B1and B2 give strong evidence for the DNN being able to disentangle curved manifolds into flattened manifolds since in the last layer of the network, the network disentangled the curved linked manifolds into 2 elliptical like manifolds and in the last layer of the DNN the network only transformed the ellipses into one elliptical like structure supporting the hypothesis. To better test this hypothesis however, highly tangled manifolds should be sampled instead (like 20 linked ellipsoids for example) and the ability for DNNs to disentangle these highly tangled manifolds should be examined in a similar fashion to better model image data in particular. Finally, this experiment should be done using image data to be able to generalize to the practical applications of DNNs. Likewise, the effects of depth and width on the ability for these networks to disentangle these manifolds should also be tested which is where I plan on going with this. In addition, the mutual information plots suggest that it requires more information to learn a more topologically complex structure since it required more information to learn the linked ellipses case as opposed to just a regular ellipse. This is possibly due to it requiring more complexity to disentangling the manifold in between the layers. A more comprehensive experiment into the information required will be done when we incorporate more highly curved manifolds to better approximate real images, and the effects of depth and width of a network will be examined.

Bibliography

[1] JOURNAL OF THE AMERICAN MATHEMATICAL SOCIETY Volume 29, Number 4, October 2016, Pages 983–1049 http://dx.doi.org/10.1090/jams/852 Article electronically published on February 9, 2016

[2] http://colah.github.io/posts/2014-03-NN-Manifolds-Topology/

[3] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In Advances in neural information processing systems, pages 2924–2932, 2014.

[4] Olivier Delalleau and Yoshua Bengio. Shallow vs. deep sum-product networks. In Advances in Neural Information Processing Systems, pages 666–674, 2011.

[5] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. arXiv preprint arXiv:1512.03965, 2015.

[6] Hrushikesh Mhaskar, Qianli Liao, and Tomaso Poggio. Learning real and boolean functions: When is deep better than shallow. arXiv preprint arXiv:1603.00988, 2016.

[7] Ravid. Shwartz-Ziv, Naftali Tishby, Opening the Black Box of Deep Neural Networks via Information, 2017, Arxiv.

Appendix:

For access to the code for a more detailed description, check the link below:
https://colab.research.google.com/drive/1QP9RDJXRuEM2f-4X8DEQJ3MYuQP230tH?usp=sharing

A1: Images of the Dataset: All of these images had either 2 degrees of freedom as in the case of the ellipse, 3 degrees of freedom as in the case of the ellipsoid, torus, helix, and linked ellipsoids, or 8 degrees of freedom as in the case of the 8D ellipses. In the case of the 8D ellipses, 4 of the degrees of freedom expressed a A*cos(t) relationship whereas the other 4 expressed a A*sin(t) relationship making each degree of freedom either linear or elliptical to the other one. Note A for each degree of freedom was randomly selected making it elliptical. The data was then embedded into a 32D space and inputted into the DNN.

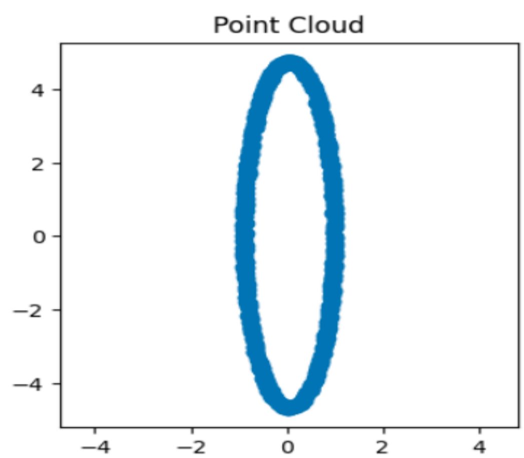


Figure 1: Ellipse

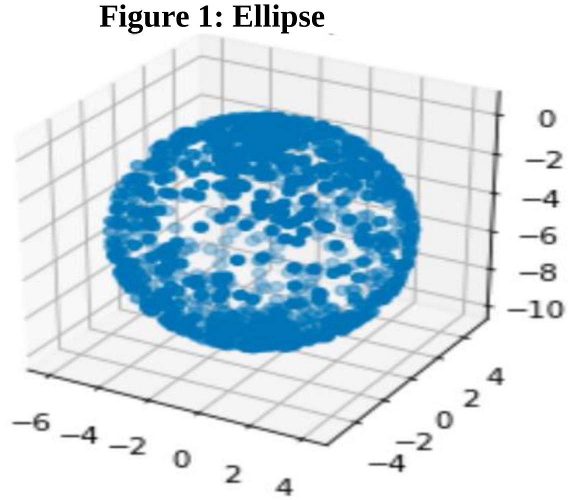


Figure 2: Ellipsoid

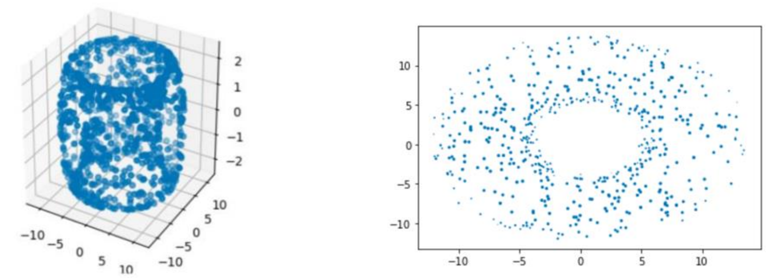


Figure 3: Torus

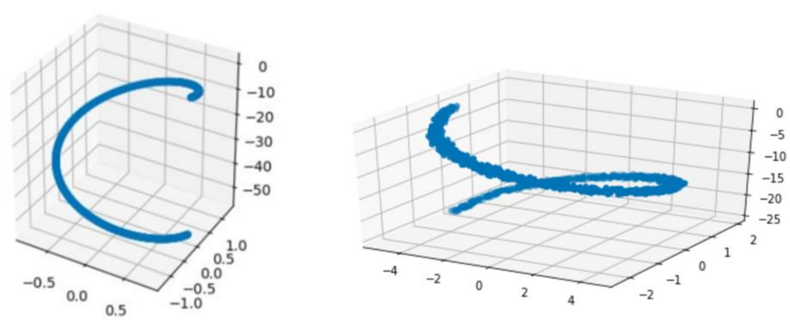


Figure 4: Helix

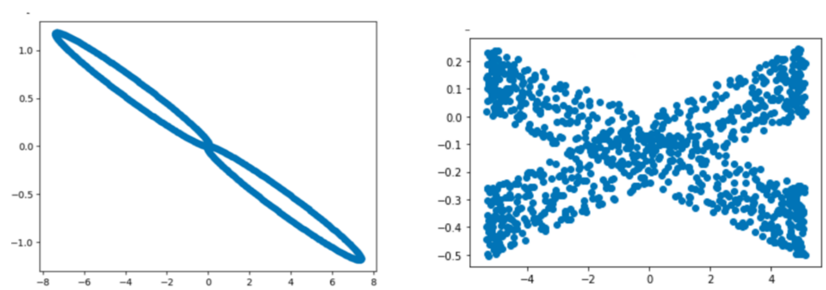


Figure 5: 8D Ellipses

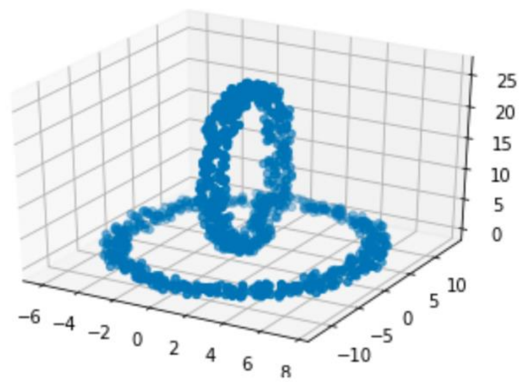


Figure 6: Linked ellipses