Design of Activation Function Using Topological Framework in Deep Neural Networks

Editors: Emtiyaz Khan and Mehmet Gonen

Abstract

Success of deep neural networks in diverse tasks across domains of computer vision, speech recognition and natural language processing, has necessitated understanding the dynamics of training process and analysis of trained models. For classification tasks, one of the main requirements of the layer transfer function is to transform data such that only information required for classification is preserved and all other information is discarded. Further, the data should be transformed such that samples belonging to a single class should come closer in the transformed space which will help in drawing a clear decision boundary across classes. This enforces the need for topological simplification of the point cloud data set formed using samples from each class. We analyze the impact of the activation function on the topological transformation of the space of training samples as it gets transformed by each successive layer during training. For the task of binary classification, the impact of various activation functions on the convergence during training is studied. In this work, the afore-mentioned convergence is considered as the criterion for the design of a novel activation function, aimed at faster convergence for binary classification task. Here, Betti numbers are used to quantify topological complexity of data. Results of experiments on synthetic binary classification datasets with large Betti numbers (> 150) using Multi Layer Perceptron(MLP) are reported. Results show that the proposed novel activation function results in faster convergence requiring fewer epochs by a factor of 1.5 to 2 as compared to the popularly used activation function Leaky ReLU. Faster convergence occurred because Betti numbers reduce faster across layers with the proposed activation function. The proposed methodology is verified on benchmark image datasets: fashion MNIST, CIFAR-10 and cat-vs-dog images, using CNN.

Keywords: Training Convergence; Activation function; Topological Data Analysis; Betti Numbers

1. Introduction

Deep neural networks have become the default choice for solving many complex tasks across several domains Krizhevsky et al. (2012); Redmon et al. (2016); Chen et al. (2017); Ronneberger et al. (2015). This is mainly because: 1) many tasks in domains such as computer vision, speech analysis and NLP, involve high dimensional datasets for which a complex model is required 2)for tasks performed by humans, which they acquired by birth, it is not well now how humans perform the task and what features they use. Deep neural networks, originally inspired by biological neural networks in cat's, is computing the features in a hierarchical fashion.

However, choosing the right architecture (like the selection of hyper parameters activation function, number of layers and number of units per layer) for a specific task is mostly based on 1) trial and error or 2) previous empirical results of tasks of similar complexity. The transfer function of each layer and the entire neural network is just treated as a complex, unknown and nonlinear function parameterized by weights and biases.

In this study, we investigate some of the desired characteristics a neural network architecture should have for solving a classification task. We derive our results based on the topology of the space of training data and how this topology changes as data is transformed by each layer.

Topology is a field of mathematics that studies the shape of objects and associated invariances like connectedness, number of holes of different dimensions etc. Weibel (1999). It is observed that many real datasets when viewed as point cloud dataset in a high dimensional space follow certain topology. For example the study in Carlsson (2009) shows that the image patches obtained from natural images follow the topology of a Klein bottle Stillwell (1993).

Topological data analysisCarlsson (2009); Chazal and Michel (2021) uses topological tools, like persistent homology Edelsbrunner et al. (2008), for analyzing point cloud dataset in order to identify and characterize underlying structures in a dataset.

In their recent study, Naitzat et al. (2020) uses Betti numbers Weibel (1999) to quantify topological complexity. In their study, they make the following important observations:

1. Topological changes to data across layers of a network remain robust under different instances of training.
2. Compared to smooth activation functions like sigmoid and tanh, non-homeomorphic activation functions like ReLU helps in changing the topology of data faster.

Our work is motivated by Naitzat et al. (2020) where the transfer function of each layer is looked at, based on how the layer changes the topology of the data. Most real world datasets have non-trivial complex topology, and in order to perform classification each layer of the neural network transforms the entire space of data to a simpler topology. This leads us to the conclusion that in order to achieve classification, each layer of the neural networks should be able to change the topology of data and hence a non-homeomorphic transformation is needed at each layer. This is achieved by activation functions with discontinuity like ReLU. However, ReLU has only a single point of discontinuity. We hypothesize that an activation function with multiple discontinuities can result in faster training (less number of epochs).

We followed the approach in Naitzat et al. (2020) and used Betti numbers to quantify topological complexity of the point cloud dataset.

In addition to the above insight, that the activation function should be non-homeomorphic, we also hypothesize that a non-bijective (many-to-one) transfer function can help to bring samples from the same class closer in the transformed space.

Based on the above two hypotheses, we introduce a new parameterized family of activation functions with multiple **many-to-one** regions and **multiple** discontinuities.

Results of our experiments show that, with the proposed activation function, the network converges faster as compared to commonly used activation functions like ReLU. It is also observed that the Betti numbers, computed using persistent homology Naitzat et al. (2020), reduce faster with the proposed activation function.

The main contribution of this paper is to provide new guidelines for designing activation functions for supervised classification tasks. We illustrate the guideline by proposing a new family of activation functions.

2. Related Work

2.1. Topological Data Analysis

Topological Data Analysis, Chazal and Michel (2021); Smith et al. (2021), is an approach for characterizing a dataset topologically using persistent homology Edelsbrunner et al. (2008). In 2008, Carlsson et al. (2008) conducted a qualitative study on 3×3 image patches taken from natural images and results of their study showed that the manifold of high contrast natural image patches is homeomorphic to that of Klein bottle.

With the use of deep neural networks for implementing machine learning tasks, like image classification, object detection and segmentation, when the data dimensionality and sample size is huge the challenge of determining the right architecture for a given dataset became a hot area of research interest. Geometric deep learning, refers to the application of deep neural networks for huge datasets with complex manifold space, not necessarily Euclidean. The Study in Bronstein et al. (2017) provides a survey of geometric deep learning. Saucan et al. (2007) introduces a new sampling technique for sampling manifolds in high dimensional spaces.

2.2. Activation function and Training convergence

Since the introduction of ReLU activation function in Krizhevsky et al. (2012) as an alternative for sigmoid and tanh functions, many different variations of it like Leaky ReLU, PReLU, ELU, Threshold ReLU etc. has been tried out for faster training convergence and better classification accuracy. Bounded ReLU activation function was suggested by Liew et al. (2016) for better generalizability and training convergence. In 2017, Ramachandran et al. (2017) used automatic search techniques to look for new activation functions. They evaluated their best reported activation function $f(x) = x.sigmoid(\beta x)$ on Imagenet using existing best performing architecture and reported 0.9% improvement on classification accuracy.

Our work differs from all of these as we are using topological simplification as a basis for deriving new activation. We propose that an activation function with many-to-one regions can reduce topological complexity of data and hence can result in faster training convergence.

3. Proposed Method

3.1. Problem Formulation

We restrict our analysis to the task of supervised classification of 1) 3-dimensional synthetic datasets using Multi Layer Perceptron(MLP) and 2) images using Convolutional Neural Network. The classification task can be viewed as a many-to-one mapping, $f : R^d \mapsto \{c_1, c_2, \dots c_k\}$. The set of all samples forms a point cloud dataset on d-dimensional space ($d = H \times W$ in the case of an image of height H and width W). As identified in Naitzat

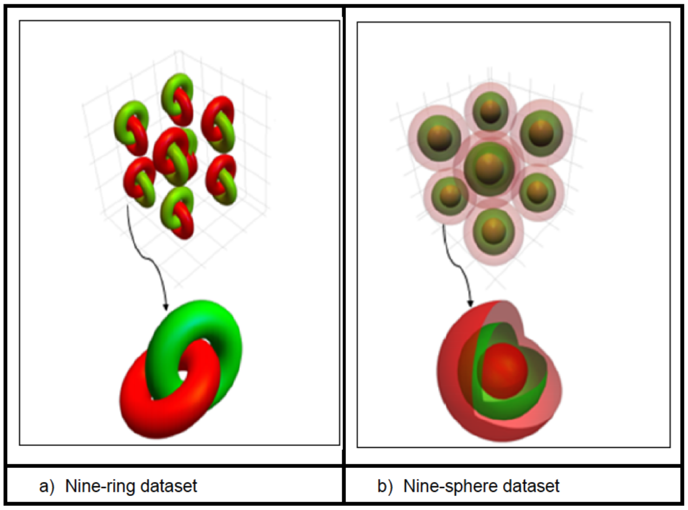


Figure 1: 3-dimensional synthetic datasets with two different classes(Red and Green) used for training MLP.
Source: https://arxiv.org/abs/2004.06093

et al. (2020), for classification task a non-homeomorphic transfer function is required for each layer as it can change the topology of the point cloud dataset. In order to achieve classification, it is necessary to change the topology from an initial complex topology to a simpler and contractible topology for each class. Another important characteristic of layer transfer function, in a classification network, is that each layer reduces the dimensionality of the input data. This helps the networks to carry forward only the information relevant for classification while discarding all other information.

This study focuses on an important aspect of neural network design – design of an optimum non-linear activation function for supervised classification tasks. See subsection 3.5 for details.

3.2. Dataset

We use two 3-dimensional simulated datasets, nine ring dataset and nine sphere dataset used in Naitzat et al. (2020). The nine ring dataset, as shown in Figure 1a, consists of two classes of data, colored Green and Red, interlocked together. The nine sphere dataset consists of nine Green spheres and 18 Red Spheres enclosing each other as shown in Figure 1b. Both the datasets contain 16000 samples for training and 2000 samples for testing.

In addition, we also perform experiments and provide results on the following real datasets:

1. Cat-Dog dataset in Kaggle – The dataset consists of images of size 32 × 32. The dataset is divided into training and testing sets, with 8000 training images and 2000 testing images, each set with an equal number of images of cats and dogs.

source:https://www.kaggle.com/c/dogs-vs-cats/

Figure 2: Sample images from Cat-Dog dataset

2. Fashion MNIST – The dataset consists of 70,000 grayscale images of size 28×28 with 10 different categories. The dataset is divided into training and testing sets, with 60,000 training images and 10,000 test images.

3. CIFAR-10 – The dataset consists of 60,000 color images of size 32×32 with 10 different categories. Each category consists of 6000 images. The dataset is divided into training and testing sets, with 50,000 training images and 10,000 test images.

Sample images from each of the above datasets are shown in Figures 2, 3 and 4. The datasets were converted to gray scale and normalized. No other preprocessing was performed.

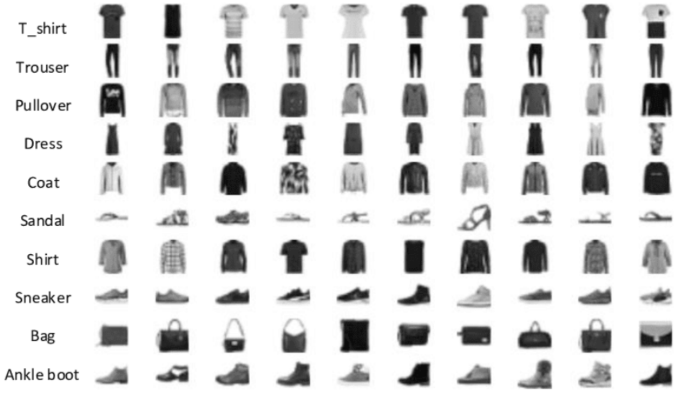


Figure 3: Sample images from each of the 10 categories in Fashion MNIST dataset.

3.3. Layer Transfer Function

The layer transfer function is composed of an affine transformation and a non-linear activation function. The generic form of the layer transfer function in MLP and CNN is given in Equations 1 and 2 below.

$$X^{(l+1)} = f(W^T X^{(l)} + b) \tag{1}$$

where:

$$\begin{aligned}
X^{(l+1)} &= \text{Output of layer } (l+1) \\
X^{(l)} &= \text{Output of layer } l \\
W &= \text{Weight matrix connecting nodes between layers } l \text{ and } (l+1) \\
b &= \text{Bias vector for layer } l \text{ and} \\
f(.) &= \text{Non-linear activation function}
\end{aligned}$$

$$x_{ij}^{(l+1)} = f(\sum_{m,n \in N(i,j)} w_{mn} x_{mn}^{(l)} + b) \tag{2}$$

where:

$$\begin{aligned}
x_{ij}^{(l+1)} &= \text{Output value of layer } (l+1) \text{ at pixel location } (i,j) \\
N(i,j) &= \text{Set of pixels in a square neighborhood centered at pixel } (i,j). \\
w_{mn} &= \text{Convolutional filter weight at location } (m,n) \\
b &= \text{Bias}
\end{aligned}$$

In the subsections below, we provide the details of desired characteristics for activation function from a topological point of view.

3.4. Significance of Betti numbers on classification

In order to perform classification task, the network needs to transform the point data cloud into separate disjoint blobs, one for each of the classes. Further, the less the numbers of holes/voids in these resulting blobs, the easier it gets to find a classification mapping function. In the present context, Betti numbers Weibel (1999) are used to describe the shape of the data distribution.

Betti numbers, denoted as $\beta_k(X)$, are used to quantify the topological complexity of a d-dimensional topological space X, where $0 \leq k \leq d-1$. Each of these descriptors, $\beta_k(X)$, captures the defining aspect of the shape of the data, X, in k^{th} dimension. The 0^{th} Betti number, $\beta_0(X)$, is the number of connected components, the first Betti number, $\beta_1(X)$, is the number of 1D holes, the second Betti number, $\beta_2(X)$, is the number of 2D holes and so on. In general, for $k > 0$, $\beta_k(X)$ represents the number of k-dimensional holes. Figure 5 shows some well-studied topological spaces and their corresponding Betti numbers Naitzat et al. (2020). Hence the Betti descriptors with N disjoint blobs should be of the form

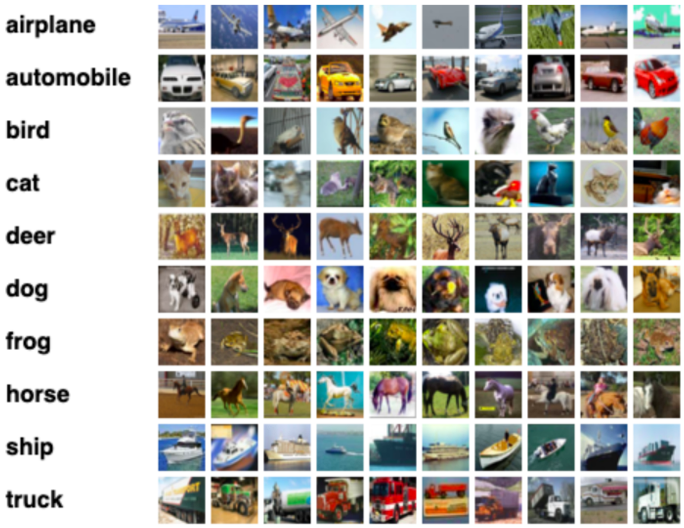
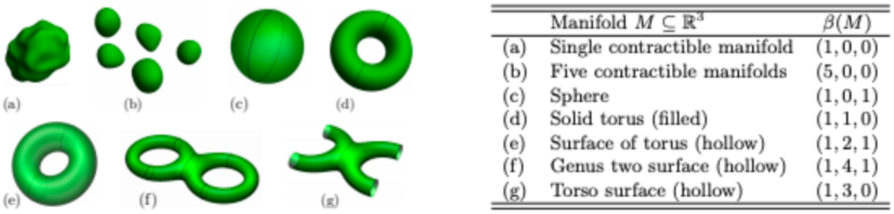


Figure 4: Sample images from CIFAR-10 dataset

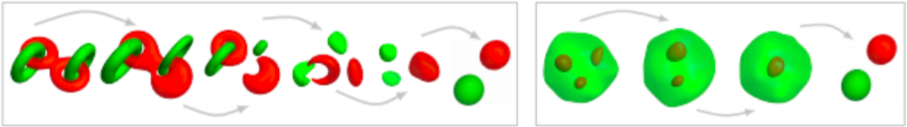
(N,0,0,..., 0) as this corresponds to the data shape with N connected components without any holes in any of them. Transforming the point data cloud of samples to a topological space with the descriptor, (N,0,0,...,0), is the goal of any classification network with N different classes.



	Manifold $M \subseteq \mathbb{R}^3$	$\beta(M)$
(a)	Single contractible manifold	$(1, 0, 0)$
(b)	Five contractible manifolds	$(5, 0, 0)$
(c)	Sphere	$(1, 0, 1)$
(d)	Solid torus (filled)	$(1, 1, 0)$
(e)	Surface of torus (hollow)	$(1, 2, 1)$
(f)	Genus two surface (hollow)	$(1, 4, 1)$
(g)	Torso surface (hollow)	$(1, 3, 0)$

Figure 5: Examples of some well studied topological spaces and corresponding Betti numbers.
Source: https://arxiv.org/abs/2004.06093

For efficient classification, one needs to quickly transform (fewer number epochs) the original point data cloud, X, from a high dimensional space with large Betti numbers to progressively smaller Betti numbers. As evident from Figure 6, this ensures that each connected component corresponds to a single class (either Red or Green) and there are no holes within any of the connected components. Hence each connected component is contractible to a single point. Hence finding a decision boundary becomes faster if there are no holes on the manifold formed by samples from a single class.



Source:https://arxiv.org/pdf/2004.06093.pdf

Figure 6: Simplification of topological space as the data is transformed by successive layers of neural network. The input data (leftmost) has a complex topology with samples from Red and Green classes interlocked together and contains holes. As the data is transformed by successive layers of the classification network, samples from the two classes get disconnected to form multiple connected components. Each connected component has samples from the same class. Transformation by further layers joins different connected components of samples from the same class into one single connected component and removes the holes from each connected component. The final transformed space has two connected components one for each class (right most).

3.5. Design of activation function

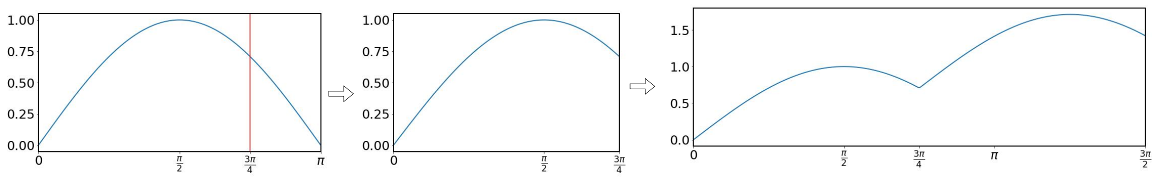
It is observed that non-homeomorphic activation functions like ReLU reduce the Betti numbers faster as opposed to traditional activation functions like sigmoid and tanh which are homeomorphic Naitzat et al. (2020). Further, the more the number of discontinuities, the more powerful the activation function will be in terms of reducing the topological complexity. In addition to these findings, we also hypothesize that multiple many-to-one regions in the layer transfer function can reduce the Betti descriptors of samples within a single class, as it tries to bring more samples together hence forming a single blob without holes. As shown in Figure 7, we start with a positive half cycle of a sine function, and select a cut-off point $x = \frac{3\pi}{4}$ so that the selected interval $[0, \frac{3\pi}{4}]$ has a many-to-one mapping region. The selected portion of the sine function is repeated and superimposed on a linearly increasing function like ReLU to get the final activation function as shown in Figure 7c.

3.6. Neural Network Architecture

We performed experiments using MLP on nine-sphere and nine-rings datasets. For image datasets, we used Convolutional Neural Networks for running the experiments. The MLP for simulated dataset consists of 9 layers with 25 neurons in each layer. We performed experiments with Leaky ReLU and the proposed activation function. Our empirical results show that the proposed activation function results in faster training convergence compared to Leaky ReLU.

4. Results and Discussion

Figure 8 shows comparison of progression of first Betti numbers (number of connected components) as the data is transformed across layers when Leaky ReLU is is used in all

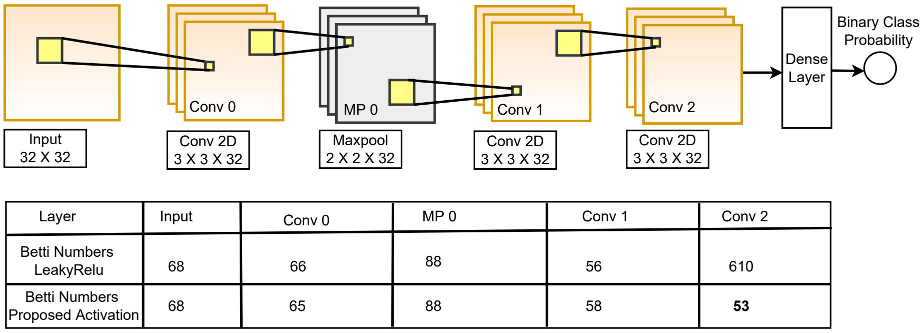


a) Positive half cycle of sine wave with cut-off point marked

b) Interval selected for design of generating activation function

c) Interval repeated and superimposed on top of a linearly increasing function for multiple many-to-one regions

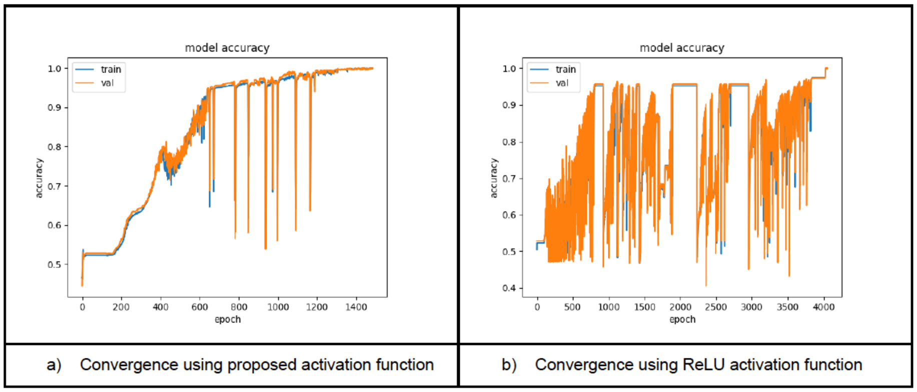
Figure 7: For classification task a many-to-one activation function is required, as it maps samples from a given class to the respective class label. Steps involved in the design of the proposed novel activation function are outlined. We start with a single half cycle of a sinusoidal function and choose a cutoff-point $\frac{3\pi}{4}$ such that the interval $[0, \frac{3\pi}{4}]$ contains many-to-one region. The values of the function in this interval are repeated to get the final activation function.



Layer	Input	Conv 0	MP 0	Conv 1	Conv 2
Betti Numbers LeakyRelu	68	66	88	56	610
Betti Numbers Proposed Activation	68	65	88	58	**53**

Figure 8: Architecture of CNN for classifying images in CIFAR-10 dataset along with the comparison of progression of first Betti number (Number of connected components) of samples from a single class across the layers using Leaky ReLU and proposed activation function. First row of the table shows Betti numbers using Leaky ReLU. Second row shows the Betti numbers with the proposed activation function added in layer Conv 2. It is observed that Betti number reduce significantly in layer Conv 2.

the layers Vs the proposed activation function is used is the final convolutional layer. It is observed, from Figures 8, that with the proposed activation function Betti numbers decrease by a significantly larger amount than Leaky ReLU activation function. Figure 9 shows the comparison of convergence of MLP with Leaky ReLU and proposed activation function. Nine-sphere and nine-ring datasets are used for this comparative study. As evident from this figure, the reduction in Betti numbers directly translates to faster training convergence. The new activation functions with multiple many-to-one regions seem to work well on various datasets and help in achieving at-par or better trainability.



a) Convergence using proposed activation function

b) Convergence using ReLU activation function

Figure 9: Comparison of training convergence using proposed activation function and Leaky ReLU activation function. It is observed that convergence is faster with the proposed activation function.

Also, The performance impact becomes more pronounced at larger batch-sizes i.e. with larger batch-sizes the gain tends to reach a factor of 2. On increasing the batch size the training for even simpler datasets takes longer (more epochs) for legacy activation functions. Even though the increase in training epochs is seen for the proposed activation function also, the increase is less pronounced.

5. Conclusion

In this work, we look at topological complexity of data at the output of each layer of deep neural network for binary classification tasks. We propose a novel activation function that simplifies the topological complexity of point cloud data (measured using Betti numbers) at each hidden layer which translates to faster training convergence. We evaluate the proposed methods on popular image classification datasets and our results show that the proposed activation function results in faster training convergence. It is observed that the Betti numbers of point topological space formed by samples from each class reduces drastically with the proposed activation function.

References

Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

Gunnar Carlsson. Topology and data. *Bulletin of the American Mathematical Society*, 46 (2):255–308, 2009.

Gunnar Carlsson, Tigran Ishkhanov, Vin De Silva, and Afra Zomorodian. On the local behavior of spaces of natural images. *International journal of computer vision*, 76(1): 1–12, 2008.

Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Frontiers in artificial intelligence*, 4, 2021.

Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. doi:https://doi.org/10.1109/TPAMI.2017.2699184.

Herbert Edelsbrunner, John Harer, et al. Persistent homology-a survey. *Contemporary mathematics*, 453:257–282, 2008.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. Available:https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Shan Sung Liew, Mohamed Khalil-Hani, and Rabia Bakhteri. Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems. *Neurocomputing*, 216:718–734, 2016.

Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. *J. Mach. Learn. Res.*, 21(184):1–40, 2020.

Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. doi:https://doi.org/10.1109/CVPR.2016.91.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. doi:https://link.springer.com/chapter/10.1007/978-3-319-24574-4_28.

Emil Saucan, Eli Appleboim, and Yehoshua Y Zeevi. Geometric sampling of manifolds for image representation and processing. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 907–918. Springer, 2007.

Alexander D Smith, Paweł Dłotko, and Victor M Zavala. Topological data analysis: concepts, computation, and applications in chemical engineering. *Computers & Chemical Engineering*, 146:107202, 2021.

John Stillwell. Curves on surfaces. In *Classical Topology and Combinatorial Group Theory*, pages 185–215. Springer, 1993.

Charles A. Weibel. Chapter 28 - history of homological algebra. In I.M. James, editor, *History of Topology*, pages 797–836. North-Holland, Amsterdam, 1999. ISBN 978-0-444-82375-5. doi: https://doi.org/10.1016/B978-044482375-5/50029-8. URL https://www.sciencedirect.com/science/article/pii/B9780444823755500298.