Topology-Aware Activation Functions in Neural Networks

Pavel Snopov[1] and Oleg R. Musin[1]

University of Texas Rio Grande Valley - School of Mathematical and Statistical Sciences
Brownsville - USA

Abstract. This study explores novel activation functions that enhance the ability of neural networks to manipulate data topology during training. Building on the limitations of traditional activation functions like ReLU, we propose SmoothSplit and ParametricSplit, which introduce topology «cutting» capabilities. These functions enable networks to transform complex data manifolds effectively, improving performance in scenarios with low-dimensional layers. Through experiments on synthetic and real-world datasets, we demonstrate that ParametricSplit outperforms traditional activations in low-dimensional settings while maintaining competitive performance in higher-dimensional ones. Our findings highlight the potential of topology-aware activation functions in advancing neural network architectures. The code is available via `https://github.com/Snopoff/Topology-Aware-Activations`.

1 Introduction

Despite significant advancements in neural networks across various domains, a comprehensive explanation of the learning mechanism remains an open and crucial question. Analyzing internal data representations and studying the dynamics of changes in the geometry and topology of data in hidden layers can provide valuable insights into these mechanisms.

Such analysis was performed in a seminal paper [1], in which the authors investigate how the topology of a dataset $M = M_a \cup M_b$, representing two classes a and b, changes as it passes through the layers of a neural network well-trained for binary classification. They found that neural networks transform «topologically complex» datasets into «topologically trivial» datasets as the data passes through the layers. These topological transformations occur significantly faster in neural networks that utilize ReLU as the activation function. The authors attributed this to the topological nature of ReLU, which, being a non-injective function, fails to be a homeomorphism. As a result, ReLU can effectively «glue» points, unlike tanh, which is a homeomorphism and thus preserves the topology of the input.

Building on this perspective, activation functions that do not preserve topology could be further explored. Assuming that topology simplification is a primary task during neural network training, new activation functions could be designed to enhance this process more efficiently.

In this study, we propose novel activation functions that, like ReLU, are non-homeomorphic. However instead of «gluing» points these functions «split»

them, destructing the underlying topology. We evaluate these functions against ReLU and other widely-used activation functions on synthetic and real-world datasets. Our experiments demonstrate that these novel activation functions often outperform or match the performance of existing methods, suggesting their potential for broader application in neural network architectures.

2 Related Work

Beyond the foundational work in [1], several studies have explored the dynamics and changes in the geometric and topological properties of internal data representations in neural networks. For instance, [2] demonstrates that the training process of deep neural networks is linked to untangling the latent manifolds on which data resides. The authors of [3] investigate how the geometry of these latent manifolds, including their dimensionality, changes during training. Their findings suggest that well-trained networks transform the latent manifolds of different classes into linearly separable representations.

In [4], the authors utilize persistent landscapes, a vectorization technique for persistence diagrams, to analyze the dynamics of topological complexity in the network's hidden layers, demonstrating that topological characteristics do not always simplify during training. Work [5] examines internal representations in various modern neural networks for computer vision and natural language processing tasks from a topological perspective, observing that the learning process is directly linked to changes in topological characteristics. Moreover, it shows that model quality and generalization capabilities may be associated with the topology and geometry of the data.

3 Non-Homeomorphic Activation Functions

As mentioned in the introduction, the effectiveness of ReLU, as discussed in [1], is attributed to its topological properties when considered as a function $\mathbb{R} \to \mathbb{R}$. It fails to be a homeomorphism, therefore it alters the topology by by compressing it. Specifically, it «eliminates» non-trivial cycles in homology of underlying data manifold, simplifying its topology.

However, topology simplification can also be achieved by «cutting» the data manifold. This process, if applied appropriately (e.g., along non-trivial cycles), divides the manifold into simpler components. A function capable of such cutting must also be non-homeomorphic, however, unlike ReLU, it must be non-surjective rather than non-injective. For example, consider the the following function, which termed Split:

$$\text{Split}(x) = x + \text{sign}(x)c,$$

where c controls the distortion of that cut at 0. When used as an activation function, Split divides the original data manifold along each dimension.

While effective in splitting data, Split is non-differentiable and thus unsuitable as an activation function in neural networks. To address this limitation, we

propose a smooth approximation, SmoothSplit, defined as:

$$\text{SmoothSplit}(x) = x + \tanh(\alpha x)c.$$

where α determines the sharpness of the approximation. For sufficiently large α, SmoothSplit closely approximates Split while maintaining differentiability, making it compatible with the neural network training pipeline.

While splitting data manifolds simplifies their topology, neural networks may still need to «glue» points together, particularly in the final layers. To enable both splitting and compressing operations, we propose a parametric activation function, ParametricSplit, defined as:

$$\text{ParametricSplit}(x) = \begin{cases} bx + b\cos a - \sin a, & \text{if } x \leq -\cos a \\ x\tan(a), & \text{if } -\cos a \leq x \leq \cos a \\ x + \sin a - \cos a, & \text{if } x \geq \cos a \end{cases}$$

This function can emulate ReLU, Split, or SmoothSplit under appropriate parameter settings:

- For $a = 0, b = 0$, $\text{ParametricSplit}(x) = \text{ReLU}(x - 1)$
- For $a = \frac{\pi}{2}, b = 1$, $\text{ParametricSplit}(x) = \text{Split}(x)$
- For $a \in [\frac{\pi}{4}, \frac{\pi}{2}), b = 1$, $\text{ParametricSplit}(x) = \text{SmoothSplit}(x)$, with α uniquely defined by a (see fig. 1).

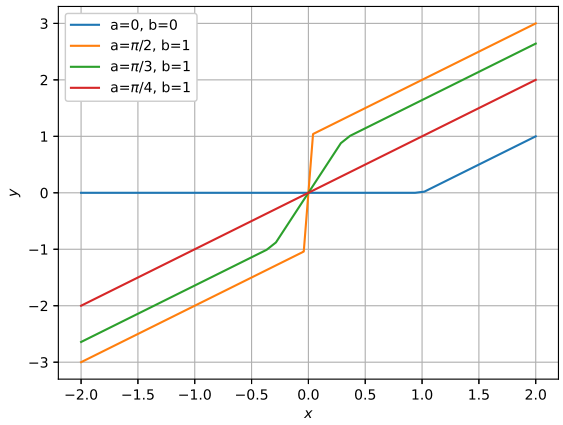


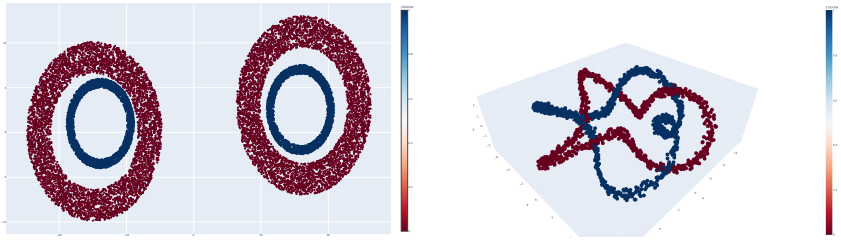
Fig. 1: ParametricSplit with different parameters

In summary, there are two primary types of manifold deformations: compression and splitting. While ReLU performs the former, Split and SmoothSplit

achieve the latter. The ParametricSplit function unifies these operations, enabling both compression and splitting depending on its parameters. Since these parameters can be learned during training, ParametricSplit integrates seamlessly into the neural network pipeline. In the following sections, we compare ParametricSplit with ReLU and other popular activation functions on synthetic and real-world datasets.

4 Experiments

We evaluated the proposed activation functions in binary classification tasks, comparing their performance with ReLU, tanh, and PReLU. The experiments involved two synthetic datasets, CurvesOnTorus and Circles, and one real-world dataset, the Breast Cancer Wisconsin dataset. In the synthetic datasets, each class is sampled from a distinct manifold that is intertwined with others, making linear separability impossible. The data manifolds in these synthetic datasets are one-dimensional and immersed in $\mathbb{R}^2$ and $\mathbb{R}^3$ for Circles and CurvesOnTorus, respectively (illustrated in Figure 2).



(a) Dataset Circles (b) Dataset CurvesOnTorus

Fig. 2: Synthetic datasets

We utilized fully connected neural networks for the experiments. The activation functions being compared were applied to all layers except the last one, which used ReLU. This design ensured that the network «glued» data components at the final layer, as mentioned earlier. However, further ablation studies are needed to assess the impact of this configuration.

To investigate the effects of network depth and layer dimensionality on performance, we varied the number of layers and the dimensions of each layer. Specifically, the number of layers was set to $\{1, 2, 3\}$, and layer dimensions varied depending on the dataset:$\{2, 3, 4\}$ for Circles,$\{3, 4, 5, 6, 7\}$ for CurvesOnTorus, and $\{30, 40, 80, 100\}$ for Breast Cancer Wisconsin.

The models were trained for 100 epochs with weights initialized using Xavier normal initialization, except for networks with ReLU, which used Xavier uniform initialization. The learning rate was set to 0.05, and datasets were split into training and test sets in a 70/30 ratio. To ensure robustness, each experiment was repeated 10 times.

Activation functions	# of layers	**Circles**			**CurvesOnTorus**		
		2	3	4	3	4	5
tanh	1	0.53608	0.50629	0.45689	0.4638	0.30247	**0.1277**
	2	0.59075	0.53033	0.49025	0.45167	0.24247	**0.11148**
	3	0.53842	0.50128	0.46227	0.46198	**0.28846**	**0.13182**
ReLU	1	0.56848	0.56269	0.52233	0.59894	0.42909	0.29224
	2	0.57535	0.51272	0.51093	0.5043	0.3833	0.28264
	3	0.60052	0.50569	0.47942	0.52122	0.48321	0.24112
PReLU	1	**0.51215**	0.50723	0.50507	0.48369	0.33667	0.16796
	2	**0.51127**	0.46182	0.42828	0.42597	0.22896	0.15549
	3	0.52945	**0.4134**	0.44995	0.49129	0.32966	0.17116
SmoothSplit	1	0.54291	**0.48158**	0.4609	0.53448	0.37739	0.27432
	2	0.55634	0.51703	0.4963	0.47907	0.41282	0.20849
	3	0.57883	0.52049	0.48448	0.50015	0.3845	0.31222
ParametricSplit	1	0.52717	0.49387	**0.39754**	**0.46232**	**0.25139**	0.20235
	2	0.55464	**0.45533**	**0.38826**	**0.29508**	**0.21714**	0.17218
	3	**0.51639**	0.53109	**0.38475**	**0.38731**	0.326	0.13416

Activation functions	# of layers	**CurvesOnTorus**		**Breast Cancer**			
		6	7	30	40	80	100
tanh	1	**0.07998**	**0.04432**	0.659	0.65911	0.65907	0.65931
	2	0.075843	0.06734	0.6592	0.65921	0.65922	0.6591
	3	**0.09893**	0.03521	0.65919	0.6592	0.65914	0.65911
ReLU	1	0.22594	0.06334	0.45753	0.29907	0.4866	0.53435
	2	0.16865	0.08815	**0.24669**	0.33424	**0.33288**	**0.38068**
	3	0.28498	**0.03179**	0.34099	0.3926	**0.3872**	0.50572
PReLU	1	0.10352	0.07216	0.35806	0.36374	0.3136	**0.28632**
	2	0.07196	**0.01427**	0.35431	0.31485	0.41877	0.6172
	3	0.30733	0.08859	0.31048	**0.37636**	0.61726	0.52685
SmoothSplit	1	0.22501	0.12244	0.48983	0.38327	**0.29216**	0.41555
	2	0.24841	0.09748	0.61399	0.53298	0.45029	2.97077
	3	0.29647	0.26787	0.5737	0.59591	8.13304	6.08318
ParametricSplit	1	0.14366	0.11116	**0.3265**	**0.22968**	0.44336	0.35209
	2	**0.05791**	0.03494	0.30993	**0.24246**	0.56696	0.54031
	3	0.12654	0.0851	**0.283**	0.38585	0.46578	**0.50018**

Table 1: Val. loss averaged in 10 runs

5 Results

The results of the experiments, summarized in Table 1, indicate that the proposed ParametricSplit activation function performs exceptionally well, particularly in low-dimensional scenarios. It outperformed traditional activation functions like ReLU, tanh, and PReLU, especially when the layer dimensions were small relative to the data manifold's dimensionality. In higher-dimensional settings, ParametricSplit performed comparably to the best-performing activation functions. Similarly, SmoothSplit achieved competitive results, often on par with traditional choices such as ReLU and tanh.

One notable observation is the strong performance of tanh and PReLU on the CurvesOnTorus dataset in configurations with larger dimensions. This can be attributed to the one-dimensional nature of the class manifolds in this dataset. Immersion in $\mathbb{R}^5$ or higher dimensions provides sufficient freedom for the neural network to disentangle the curves through homeomorphic transformations.

These findings suggest an underlying explanation for the observed phenomena: when the network's layer dimensions are low relative to the data manifold's dimensionality, the network has limited degrees of freedom and lacks sufficient «room» to disentangle the data through homeomorphic transformations. In such

cases, the network must manipulate the data topology to solve the classification task. The learnable parameters of ParametricSplit provide this capability, enabling the network to «cut» the topology and adjust it as needed.

6 Conclusion

In this study, we introduced novel activation functions, ParametricSplit and SmoothSplit, designed to manipulate the topology of data representations within neural networks. These functions extend the capabilities of traditional activations like ReLU by allowing for both topology «cutting» and «gluing» operations, providing networks with greater flexibility to adapt to the underlying data manifold. Through experiments on synthetic and real-world datasets, we demonstrated that ParametricSplit consistently outperformed traditional activations in low-dimensional settings, where topology transformation is crucial for solving classification tasks. In higher-dimensional scenarios, ParametricSplit matched or exceeded the performance of established functions, showcasing its versatility.

Future work could explore the broader applicability of these functions in other machine learning tasks, or investigate the impact of topological transformations on network generalization. Additionally, integrating these functions into architectures beyond fully connected networks, such as convolutional or transformer-based models, presents an interesting topic for further research too.

References

[1] Gregory Naitzat, Andrey Zhitnikov, and Lek-Heng Lim. Topology of deep neural networks. *J. Mach. Learn. Res.*, 21(1), jan 2020.

[2] Pratik Brahma, Dapeng Wu, and Yiyuan She. Why deep learning works: A manifold disentanglement perspective. *IEEE Transactions on Neural Networks and Learning Systems*, 27:1–12, 12 2015.

[3] Uri Cohen, SueYeon Chung, Daniel D. Lee, and Haim Sompolinsky. Separability and geometry of object manifolds in deep neural networks. *Nature Communications*, 11(1):746, 2020.

[4] Matthew Wheeler, Jose Bouza, and Peter Bubenik. Activation landscapes as a topological summary of neural network performance. In *2021 IEEE International Conference on Big Data (Big Data)*. IEEE, dec 2021.

[5] German Magai. Deep neural networks architectures from the perspective of manifold learning, 2023.