

Topology-Aware Activation Functions in Neural Networks

Pavel Snopov¹ and Oleg R. Musin¹

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 and Related Work

Despite significant advances, the underlying mechanisms of neural network learning remain only partially understood. Studies such as [1] have shown that a well-trained network (i.e., one achieving near-zero generalization error) progressively transforms a complex dataset $M = M_a \cup M_b$ into a topologically simpler one. This transformation is largely attributed to the non-injective nature of ReLU, which tends to «glue» points together, whereas functions like tanh preserve the input topology.

Additional work [2, 3, 4, 5] suggests that untangling latent manifolds is crucial for improving classification performance, emphasizing the role of topological transformations in deep learning. Moreover, topology-aware activation functions have been applied successfully in tasks such as image segmentation [6] and graph neural networks [7].

Motivated by these insights, we propose novel non-homeomorphic activation functions that «split» data manifolds—effectively «cutting» topology rather than simply compressing it. The functions, SmoothSplit and ParametricSplit, are designed to enhance a network’s ability to restructure its internal representations in a topology-aware manner. Through experiments on both synthetic and real-world datasets, we demonstrate that ParametricSplit notably improves performance in low-dimensional settings while remaining competitive in higher dimensions, underscoring the potential of *topology-aware activation functions* for advancing neural network architectures.

2 Non-Homeomorphic Activation Functions

As discussed in the introduction, the effectiveness of ReLU (as detailed in [1]) is largely due to its topological properties when considered as a function from \mathbb{R} to \mathbb{R} . Since ReLU is not a homeomorphism, it alters the topology by compressing it; specifically, it «eliminates» non-trivial cycles in the homology of the underlying data manifold, thereby simplifying its structure.

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

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

where c is a learnable parameter, initialized randomly between 0 and 1, that controls the distortion at $x = 0$. When used as an activation function, Split divides the original data manifold along each dimension.

Although effective for splitting data, Split is non-differentiable and therefore unsuitable for use in neural networks. To overcome this limitation, we propose a smooth approximation, SmoothSplit, defined as

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

where α (optimized during training and initialized randomly between 0 and 1) determines the sharpness of the approximation. For sufficiently large α , SmoothSplit closely approximates Split while maintaining differentiability, thus making it compatible with gradient-based training.

While splitting data manifolds can simplify their topology, neural networks may still require the ability to «glue» points together, particularly in the final layers. To enable both splitting and compressing operations, we introduce 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 $a = 0$, $b = 0$, $\text{ParametricSplit}(x)$ approximates $\text{ReLU}(x - 1)$.
- For $a = \frac{\pi}{2}$, $b = 1$, $\text{ParametricSplit}(x)$ recovers $\text{Split}(x)$.
- For $a \in [\frac{\pi}{4}, \frac{\pi}{2})$ with $b = 1$, $\text{ParametricSplit}(x)$ approximates $\text{SmoothSplit}(x)$, with α uniquely defined by a .

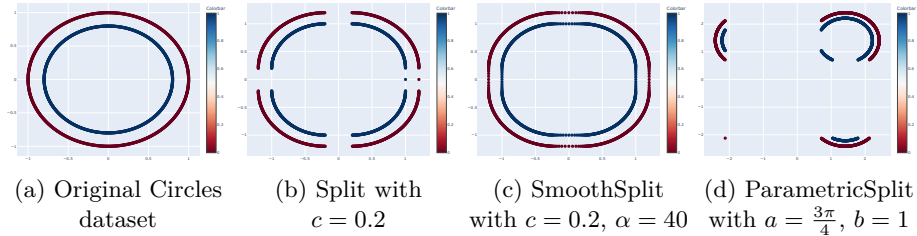


Fig. 1: Comparison of different transformations applied to the Circles dataset.

In summary, there are two primary types of manifold deformations: compression and splitting. While ReLU primarily compresses the data manifold, Split and SmoothSplit effect splitting. The proposed ParametricSplit unifies these operations by enabling both compression and splitting, depending on its parameter settings. Figure 1 illustrates these transformations on the Circles dataset. Since the parameters of ParametricSplit are learnable during training, it integrates seamlessly into the neural network pipeline. In the following sections, we compare ParametricSplit with ReLU and other widely used activation functions on both synthetic and real-world datasets.

3 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). As for the loss function, both in training setting and validation setting, we used binary cross-entropy loss.

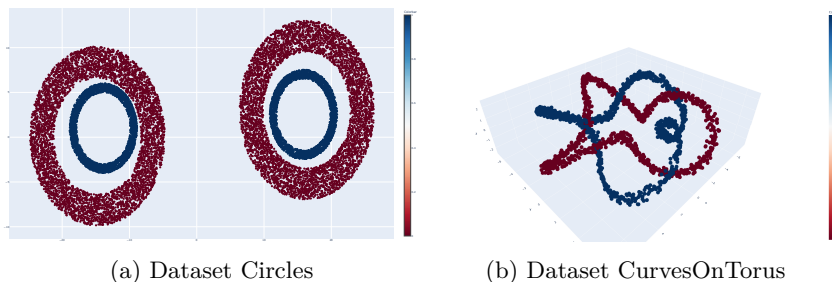


Fig. 2: Synthetic datasets

We utilized fully connected neural networks for the experiments. The acti-

vation 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 hidden layers and the dimensions of each layer. The number of hidden 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.

4 Results

The experimental results summarized in Table 1 demonstrate that the proposed ParametricSplit activation function performs exceptionally well, particularly in low-dimensional settings. It outperforms traditional activation functions such as ReLU, tanh, and PReLU when the layer dimensions are small relative to the data manifold’s intrinsic dimensionality. In higher-dimensional scenarios, ParametricSplit achieves performance comparable to the best-performing conventional activations. Similarly, SmoothSplit delivers competitive results, often matching those of ReLU and tanh.

A notable observation is the strong performance of tanh and PReLU on the CurvesOnTorus dataset in configurations with larger dimensions. This phenomenon is likely due to the one-dimensional nature of the class manifolds in this dataset; immersion in \mathbb{R}^5 or higher affords the network sufficient degrees of freedom to disentangle the curves via homeomorphic transformations. Conversely, in low-dimensional settings, the network benefits from explicitly manipulating the data topology. The learnable parameters of ParametricSplit provide the necessary flexibility to «cut» the topology and adapt to the constraints imposed by limited layer dimensions.

5 Conclusion

In this study, we introduced novel activation functions, ParametricSplit and SmoothSplit, which enhance neural networks by enabling explicit manipulation of data topology. These functions extend the capabilities of traditional activations such as ReLU by facilitating both topological «cutting» and «gluing» operations, thereby offering greater flexibility in adapting to the underlying data manifold. Our experiments on synthetic and real-world datasets demonstrate that ParametricSplit consistently outperforms conventional activations in

# of layers	Activation functions	Circles			CurvesOnTorus		
		2	3	4	3	4	5
1	tanh	0.536 (±0.066)	0.506 (±0.059)	0.457 (±0.038)	0.464 (±0.068)	0.302 (±0.056)	0.128 (±0.057)
	ReLU	0.569 (±0.075)	0.563 (±0.074)	0.522 (±0.053)	0.599 (±0.070)	0.429 (±0.112)	0.292 (±0.156)
	PReLU	0.512 (±0.093)	0.507 (±0.071)	0.505 (±0.076)	0.484 (±0.112)	0.337 (±0.118)	0.168 (±0.107)
	SmoothSplit	0.543 (±0.085)	0.482 (±0.081)	0.461 (±0.052)	0.534 (±0.045)	0.377 (±0.140)	0.202 (±0.106)
	ParametricSplit	0.527 (±0.012)	0.494 (±0.087)	0.398 (±0.128)	0.462 (±0.144)	0.251 (±0.193)	0.202 (±0.122)
	tanh	0.591 (±0.048)	0.530 (±0.066)	0.490 (±0.086)	0.452 (±0.118)	0.242 (±0.091)	0.111 (±0.069)
2	ReLU	0.576 (±0.064)	0.513 (±0.096)	0.511 (±0.123)	0.504 (±0.105)	0.383 (±0.134)	0.283 (±0.120)
	PReLU	0.511 (±0.140)	0.462 (±0.087)	0.428 (±0.073)	0.426 (±0.090)	0.230 (±0.160)	0.155 (±0.127)
	SmoothSplit	0.556 (±0.068)	0.517 (±0.084)	0.496 (±0.049)	0.479 (±0.078)	0.413 (±0.121)	0.208 (±0.122)
	ParametricSplit	0.555 (±0.076)	0.455 (±0.142)	0.388 (±0.152)	0.295 (±0.132)	0.217 (±0.193)	0.172 (±0.211)
	tanh	0.538 (±0.084)	0.501 (±0.069)	0.462 (±0.109)	0.462 (±0.093)	0.288 (±0.110)	0.132 (±0.055)
	ReLU	0.601 (±0.048)	0.506 (±0.076)	0.479 (±0.116)	0.521 (±0.085)	0.483 (±0.173)	0.241 (±0.188)
3	PReLU	0.530 (±0.079)	0.413 (±0.069)	0.450 (±0.106)	0.491 (±0.097)	0.330 (±0.220)	0.171 (±0.148)
	SmoothSplit	0.579 (±0.060)	0.521 (±0.068)	0.485 (±0.052)	0.500 (±0.124)	0.385 (±0.110)	0.312 (±0.114)
	ParametricSplit	0.516 (±0.086)	0.531 (±0.063)	0.385 (±0.171)	0.387 (±0.168)	0.326 (±0.236)	0.134 (±0.158)
	tanh	0.076 (±0.062)	0.067 (±0.027)	0.659 (±0.012)	0.659 (±0.013)	0.659 (±0.013)	0.659 (±0.012)
	ReLU	0.169 (±0.130)	0.088 (±0.101)	0.458 (±0.149)	0.299 (±0.214)	0.534 (±0.227)	0.534 (±0.248)
	PReLU	0.104 (±0.120)	0.072 (±0.103)	0.358 (±0.214)	0.364 (±0.209)	0.314 (±0.189)	0.286 (±0.137)
1	SmoothSplit	0.225 (±0.091)	0.122 (±0.096)	0.490 (±0.214)	0.383 (±0.176)	0.292 (±0.148)	0.416 (±0.220)
	ParametricSplit	0.144 (±0.089)	0.111 (±0.106)	0.327 (±0.190)	0.230 (±0.066)	0.443 (±0.228)	0.352 (±0.183)
	tanh	0.285 (±0.220)	0.032 (±0.034)	0.341 (±0.229)	0.393 (±0.229)	0.387 (±0.230)	0.506 (±0.205)
	ReLU	0.307 (±0.462)	0.089 (±0.072)	0.310 (±0.189)	0.310 (±0.206)	0.617 (±0.164)	0.527 (±0.223)
	PReLU	0.300 (±0.245)	0.268 (±0.109)	0.574 (±0.370)	0.596 (±0.545)	8.133 (±18.376)	6.083 (±14.711)
	SmoothSplit	0.127 (±0.190)	0.085 (±0.177)	0.283 (±0.161)	0.386 (±0.216)	0.465 (±0.272)	0.500 (±0.255)
2	ParametricSplit	0.099 (±0.093)	0.035 (±0.032)	0.659 (±0.013)	0.659 (±0.012)	0.659 (±0.013)	0.659 (±0.013)
	ReLU	0.285 (±0.220)	0.032 (±0.034)	0.341 (±0.229)	0.393 (±0.229)	0.387 (±0.230)	0.506 (±0.205)
	PReLU	0.307 (±0.462)	0.089 (±0.072)	0.310 (±0.189)	0.310 (±0.206)	0.617 (±0.164)	0.527 (±0.223)
	SmoothSplit	0.300 (±0.245)	0.268 (±0.109)	0.574 (±0.370)	0.596 (±0.545)	8.133 (±18.376)	6.083 (±14.711)
	ParametricSplit	0.127 (±0.190)	0.085 (±0.177)	0.283 (±0.161)	0.386 (±0.216)	0.465 (±0.272)	0.500 (±0.255)
	tanh	0.076 (±0.062)	0.067 (±0.027)	0.659 (±0.012)	0.659 (±0.013)	0.659 (±0.013)	0.659 (±0.012)
3	ReLU	0.169 (±0.130)	0.088 (±0.101)	0.458 (±0.149)	0.299 (±0.214)	0.534 (±0.227)	0.534 (±0.248)
	PReLU	0.104 (±0.120)	0.072 (±0.103)	0.358 (±0.214)	0.364 (±0.209)	0.314 (±0.189)	0.286 (±0.137)
	SmoothSplit	0.225 (±0.091)	0.122 (±0.096)	0.490 (±0.214)	0.383 (±0.176)	0.292 (±0.148)	0.416 (±0.220)
	ParametricSplit	0.144 (±0.089)	0.111 (±0.106)	0.327 (±0.190)	0.230 (±0.066)	0.443 (±0.228)	0.352 (±0.183)
	tanh	0.076 (±0.062)	0.067 (±0.027)	0.659 (±0.012)	0.659 (±0.013)	0.659 (±0.013)	0.659 (±0.012)
	ReLU	0.169 (±0.130)	0.088 (±0.101)	0.458 (±0.149)	0.299 (±0.214)	0.534 (±0.227)	0.534 (±0.248)
1	PReLU	0.104 (±0.120)	0.072 (±0.103)	0.358 (±0.214)	0.364 (±0.209)	0.314 (±0.189)	0.286 (±0.137)
	SmoothSplit	0.225 (±0.091)	0.122 (±0.096)	0.490 (±0.214)	0.383 (±0.176)	0.292 (±0.148)	0.416 (±0.220)
	ParametricSplit	0.144 (±0.089)	0.111 (±0.106)	0.327 (±0.190)	0.230 (±0.066)	0.443 (±0.228)	0.352 (±0.183)
	tanh	0.076 (±0.062)	0.067 (±0.027)	0.659 (±0.012)	0.659 (±0.013)	0.659 (±0.013)	0.659 (±0.012)
	ReLU	0.169 (±0.130)	0.088 (±0.101)	0.458 (±0.149)	0.299 (±0.214)	0.534 (±0.227)	0.534 (±0.248)
	PReLU	0.104 (±0.120)	0.072 (±0.103)	0.358 (±0.214)	0.364 (±0.209)	0.314 (±0.189)	0.286 (±0.137)
2	SmoothSplit	0.248 (±0.129)	0.097 (±0.095)	0.614 (±1.113)	0.533 (±0.339)	0.450 (±0.227)	2.971 (±5.311)
	ParametricSplit	0.058 (±0.111)	0.035 (±0.067)	0.310 (±0.192)	0.243 (±0.096)	0.567 (±0.391)	0.541 (±0.194)
	tanh	0.076 (±0.062)	0.067 (±0.027)	0.659 (±0.012)	0.659 (±0.013)	0.659 (±0.013)	0.659 (±0.012)
	ReLU	0.169 (±0.130)	0.088 (±0.101)	0.458 (±0.149)	0.299 (±0.214)	0.534 (±0.227)	0.534 (±0.248)
	PReLU	0.104 (±0.120)	0.072 (±0.103)	0.358 (±0.214)	0.364 (±0.209)	0.314 (±0.189)	0.286 (±0.137)
	SmoothSplit	0.248 (±0.129)	0.097 (±0.095)	0.614 (±1.113)	0.533 (±0.339)	0.450 (±0.227)	2.971 (±5.311)
3	ParametricSplit	0.058 (±0.111)	0.035 (±0.067)	0.310 (±0.192)	0.243 (±0.096)	0.567 (±0.391)	0.541 (±0.194)
	tanh	0.076 (±0.062)	0.067 (±0.027)	0.659 (±0.012)	0.659 (±0.013)	0.659 (±0.013)	0.659 (±0.012)
	ReLU	0.169 (±0.130)	0.088 (±0.101)	0.458 (±0.149)	0.299 (±0.214)	0.534 (±0.227)	0.534 (±0.248)
	PReLU	0.104 (±0.120)	0.072 (±0.103)	0.358 (±0.214)	0.364 (±0.209)	0.314 (±0.189)	0.286 (±0.137)
	SmoothSplit	0.248 (±0.129)	0.097 (±0.095)	0.614 (±1.113)	0.533 (±0.339)	0.450 (±0.227)	2.971 (±5.311)
	ParametricSplit	0.058 (±0.111)	0.035 (±0.067)	0.310 (±0.192)	0.243 (±0.096)	0.567 (±0.391)	0.541 (±0.194)

Table 1: Val. loss averaged in 10 runs

low-dimensional settings—where topological transformations are critical—while remaining competitive in higher dimensions.

Future work will explore the broader applicability of these functions in diverse machine learning tasks and assess the impact of topological transformations on network generalization. Additionally, integrating these activation functions into more complex architectures, such as convolutional or transformer-based models, represents an interesting direction for further research.

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.
- [6] J.S.H. Baxter and Sami Jannin. Topology-aware activation layer for neural network image segmentation. In Isgum I.Landman B.A., editor, *Progress in Biomedical Optics and Imaging - Proceedings of SPIE*, volume 11313 of *Progress in Biomedical Optics and Imaging - Proceedings of SPIE*, page 2548303, Houston, United States, February 2020. SPIE.
- [7] Bianca Iancu, Luana Ruiz, Alejandro Ribeiro, and Elvin Isufi. Graph-adaptive activation functions for graph neural networks. In *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, 2020.