On the Improvement of Generalization and Stability of Forward-Only Learning via Neural Polarization

Supplementary material

A Instability of Sigmoidal Probability Functions

As demonstrated by Gandhi et al. [4], networks trained with FFA using bounded activation functions often exhibit suboptimal performance, sometimes even rendering models incapable of learning. This reduced learning dynamic primarily stems from the vanishing gradient behavior of the sigmoidal probability function. Given that neural networks usually initialize their weights from normal distributions, the latent vectors $\boldsymbol{\ell} \in \mathbb{R}^n$ resulting from the linear operation $\mathbf{W}\mathbf{x}^T$, where $\mathbf{x} \in \mathbb{R}^m$ is an input vector and $\mathbf{W} \in \mathbb{R}^{n \times m}$, will also follow a 0-centered normal distribution for each coordinate. When computing the squared Euclidean norm of this vector after passing through the `Sigmoid` activation function to obtain the goodness, the resulting distribution will yield large goodness scores, with the expected value driven by the number of neurons. The exact expression is given by:

$$\mathbb{E}\left[\|\texttt{Sigmoid}(\boldsymbol{\ell})\|_2\right] = n \cdot \mathbb{E}[\texttt{Sigmoid}(\ell_i)] = \frac{n}{2}. \tag{A0}$$

Large expected goodness values only serve to degrade the learning dynamics in the traditional FFA. Given such large goodness scores, its becomes almost guaranteed that the value of the sigmoidal probability reach values close to 1. Under this scenario, due to the behavior of the derivative of the sigmoidal function, gradient updates will become arbitrarily small, thereby producing negligible weight updates.

To overcome this issue, a careful balance of the θ and α hyperparameters in the probability P_σ is required. While setting θ around the distribution's expected value helps for the purpose, high variance values can still render this probability function unstable, which can be fixed by tuning α. An initial proposal to mitigate this effect was given in [4], where θ was set to the number of neurons. However, due to the difference between this value and the real expected value of the goodness, this approach would still be unable of training networks using sigmoid activations, as proven by the results therein reported. Additionally, this method does not weight the impact of the variance on the probability function, which can result in models with large variance values and the sigmoid function incapable of learning, as presented in Proposition 1.

In this work, we propose the use of mean aggregation to mitigate the impact of the mean. Our hypothesis is that this strategy can reduce the correlation between the expected value of the norm and the number of neurons. However, we acknowledge that this solution still faces limitations. Due to the bounded nature of functions like `Sigmoid` or `Tanh`, the expected values of the norm can have low variance, resulting in suboptimal utilization of the probability function and inaccurate estimations of the positivity of input samples. Consequently, models employing bounded functions in FFA often achieve suboptimal performance, necessitating extensive hyperparameter tuning processes. Even when implementing such a tuning, accuracy may still be low due to distributional differences arising during model training.

B Proof of Proposition 1

Let z be the random variable obtained from the expression $G(\boldsymbol{\ell}_\oplus) - G(\boldsymbol{\ell}_\ominus)$, which computes the difference between the positive and negative goodness scores. Since $\boldsymbol{\ell}_\oplus$ and $\boldsymbol{\ell}_\ominus$ are independent, the same independence holds for their respective transformed values $G(\boldsymbol{\ell}_\oplus)$ and $G(\boldsymbol{\ell}_\ominus)$. Moreover, considering that both $G(\boldsymbol{\ell}_\oplus)$ and $G(\boldsymbol{\ell}_\ominus)$ originate from transformations of weights drawn from the same distribution, they share identical distributions, implying equal mean values. This equality implies that the mean value of z will be given by:

$$\mathbb{E}[z] = \mathbb{E}\left[G(f(\mathbf{W}_\oplus \mathbf{x}^T)) - G(f(\mathbf{W}_\ominus \mathbf{x}^T))\right] = 0 \tag{B1}$$

Additionally, it is clear that the distribution of the variable z is symmetric, as any point $z \in \mathbb{R}$ satisfies that $g(z) = g(-z)$, where g represents the probability density function of z. Given this property and the fact that the function σ satisfies that $\sigma(-x) = 1 - \sigma(x)$, we can easily verify that $\mathbb{E}[P_\sigma(z)] = 0.5$.

Given that the derivative of a sigmoidal function is $\sigma'(x) = \sigma(x)(1 - \sigma(x))$, we can express the expected value of the derivative as:

$$\mathbb{E}\left[\frac{\partial P_\sigma(z)}{\partial G(\boldsymbol{\ell}_\oplus)}\right] = \mathbb{E}\left[P_\sigma(z)(1 - P_\sigma(z))\right]. \tag{B2}$$

Using the previously stated fact that $\mathbb{E}[z] = 0.5$, we can manipulate the previous expression to obtain the following simplification:

$$\frac{1}{4} - \mathbb{E}\left[\frac{1}{4} - P_\sigma(z) + (P_\sigma(z))^2\right] = \frac{1}{4} - \mathbb{E}\left[(\mathbb{E}[P_\sigma(z)] - P_\sigma(z))^2\right]$$

In the last equation, the expected value on the right side is by definition the variance of the sigmoidal activity. Therefore, we can reformulate the expression into the desired statement, thereby completing the proof:

$$\mathbb{E}\left[\frac{\partial P_\sigma(z)}{\partial G(\boldsymbol{\ell}_\oplus)}\right] = \frac{1}{4} - \text{Var}\left[P_\sigma(z)\right] \geq 0. \tag{B3}$$

The second statement of the proposition aims to provide a lower bound for the previously mentioned expression. This lower bound is computed by employing a function that covers the original sigmoid. While there exist other covering functions that more closely approximate the original sigmoid, offering stricter lower bounds, for the purpose of this proof we limit ourselves to acknowledging the existence of such bounds. The chosen function is $x^2 + 0.25$, which can be shown to exceed $\sigma^2(x)$ over the range of real numbers. Using properties of the expected value, we prove that:

$$\mathbb{E}[P_\sigma^2(z)] \leq 0.25 + \mathbb{E}[z^2] = 0.25 + \text{Var}[z^2]. \tag{B4}$$

By expanding Equation (B2), we obtain:

$$\mathbb{E}\left[\frac{\partial P_\sigma(z)}{\partial G(\boldsymbol{\ell}_\oplus)}\right] = \underbrace{\mathbb{E}\left[P_\sigma(z)\right]}_{=0.5} - \mathbb{E}\left[P_\sigma(z))\right], \tag{B5}$$

from which, by substituting the value of $\mathbb{E}\left[P_\sigma(z))\right]$ using the inequality in Equation (B4), we obtain the desired formula:

$$\mathbb{E}\left[\frac{\partial P_\sigma(z)}{\partial G(\boldsymbol{\ell}_\oplus)}\right] \geq 0.25 - \text{Var}\left[z^2\right]. \tag{B6}$$

C Proof of Proposition 2

To prove the first statement of this proposition, we will verify that scaling the value of the goodness scores is equivalent to transforming the value of ϵ. If the given transformation of ϵ is smaller than the values of the goodness, we can verify that the function is approximately equivalent to completely removing the ϵ value. To do so, let γ be a non-negative scaling factor. Then we have that:

$$\frac{\gamma G(\boldsymbol{\ell}_\oplus) + \epsilon}{\gamma G(\boldsymbol{\ell}_\oplus) + \gamma G(\boldsymbol{\ell}_\ominus) + 2\epsilon} = \frac{G(\boldsymbol{\ell}_\oplus) + \epsilon\gamma^{-1}}{G(\boldsymbol{\ell}_\oplus) + G(\boldsymbol{\ell}_\ominus) + 2\epsilon\gamma^{-1}}. \tag{C1}$$

Given the constraint $\gamma G(\boldsymbol{\ell}_\oplus) \gg \epsilon$, we have that $G(\boldsymbol{\ell}_\oplus) \gg \epsilon\gamma^{-1}$, and, since the value of $\epsilon\gamma^{-1}$ is numerically negligible when compared to goodness scores, it follows that:

$$\frac{G(\boldsymbol{\ell}_\oplus) + \epsilon\gamma^{-1}}{G(\boldsymbol{\ell}_\oplus) + G(\boldsymbol{\ell}_\ominus) + 2\epsilon\gamma^{-1}} \approx \frac{G(\boldsymbol{\ell}_\oplus)}{G(\boldsymbol{\ell}_\oplus) + G(\boldsymbol{\ell}_\ominus)} \tag{C2}$$

The second statement of the proposition is deduced from a direct computation of the expression's derivative. Since the value of ϵ does not affect the expression significantly when $G(\boldsymbol{\ell})$ is much larger than ϵ, we will omit it for the remainder of the proof. Therefore, the derivative is expressed as:

$$\frac{\partial P_s(G(\boldsymbol{\ell}_\oplus), G(\boldsymbol{\ell}_\ominus))}{\partial G(\boldsymbol{\ell}_\oplus)} = \frac{G(\boldsymbol{\ell}_\ominus)}{G(\boldsymbol{\ell}_\oplus)} \frac{1}{G(\boldsymbol{\ell}_\oplus) + G(\boldsymbol{\ell}_\ominus)}, \tag{C3}$$

which converges to zero under two conditions: i) when the sum of the goodness values approaches infinity, or ii) when the ratio between negative and positive goodness values tends to zero. Since we assumed a upper and lower bounded sum of goodness values, the derivative's value is predominantly influenced by the ratio of goodness values. Consequently, this implies that:

$$\frac{\partial P_s(G(\boldsymbol{\ell}_\oplus), G(\boldsymbol{\ell}_\ominus))}{\partial G(\boldsymbol{\ell}_\oplus)} = \mathcal{O}\left(\frac{G(\boldsymbol{\ell}_\ominus)}{G(\boldsymbol{\ell}_\oplus)}\right), \tag{C4}$$

with $\mathcal{O}(\cdot)$ denoting asymptotic order of complexity.

D Experimental Setup: Additional Information

This appendix provides several additional details of the experimental setup, such as the choice of hyperparameter values and the total set of neural configuration chosen for the experiments in RQ1.

Hyperparameter values The hyperparameter values for the sigmoidal function in FFA were retrieved from those used in the original work of Hinton [5]: $\theta = 2$ and $\alpha = 1$. However, considering the arguments presented in Appendix A, we opt to use different hyperparameter values for models trained using the `Sigmoid` activation, as they provide better-than-random accuracy in a small set of experiments. For neural configurations employing the L_2 norm or its square variation, we use $\theta = 0.2$ and $\alpha = 5$. Configurations using the L_1 norm consider $\theta = 0.4$ and $\alpha = 2.5$. The parameters for the Polar-FFA's version of the sigmoid probability (P_σ) remained consistent with the original $\theta = 2$ and $\alpha = 1$ values for all experiments. The ϵ value of the symmetric probability P_s was set to 10^{-6} for all experiments to ensure numerical stability. No additional hyperparameter tuning was considered. As mentioned in Section 3, the division of positive and negative neurons followed a 1-to-1 relationship, meaning that each layer had the same number of positive and negative neurons.

Dataset Configuration To maintain consistency across experiments, all datasets are normalized by standardizing their values. Similarly, we employ the original train/validation/test partitions of all datasets across all experiments. No data augmentation techniques were used.

Early Stopping on CIFAR-10 To mitigate computational overhead, an early stopping strategy was applied during the experiments related to the CIFAR-10 dataset. Models that did not show improvement in accuracy for more than 10 epochs were stopped before reaching their $T = 100$ epoch limit. This strategy has been used due to the large amount of models showing suboptimum or even close-to-random performance.

Neural Configurations To provide experimental results over a comprehensive set of neural configurations, we trained models using combinations of the activation, goodness, and probability functions proposed in this appendix. This approach resulted in 108 different configurations being employed for each dataset. The set of probability functions comprises the original Sigmoid probability of FFA, the Sigmoid Probability of Polar-FFA, and the Symmetric Probability of Polar-FFA. The choice of activation functions was made to cover activations with different behaviors. For this instance, we selected: the `ReLU` function, due to its unbounded and non-negative behavior; the `Sigmoid` function, due to its bounded and non-negative behavior; and the hyperbolic tangent function `Tanh`, as its range is not limited to positive values. This selection is presented in Table D1.

Table D1. Activation and probability functions chosen to build the set of neural configurations for the training experiments aimed to answer RQ1.

Activation function	Probability function
`ReLU`, `Sigmoid`, `Tanh`	P_σ^{FFA}, P_σ, P_s

To explore a diverse set of goodness functions, we divide its functionality into three distinct components: i) the employed norm, which measures how the latent vector is evaluated; ii) the aggregation method of the norm, which can vary between the sum or the mean of the elements; and iii) the lateral inhibition mechanism, which limits the number of active neurons in the latent vector, thereby restricting the number of modified synapses during learning. While Hinton advocated for the use of the squared Euclidean norm in the original approach, mainly due to the simplicity of the gradient, we investigate the use of two additional norm functions: the $\|\cdot\|_2$ norm and the $\|\cdot\|_1$ norm. Additionally, we analyze the different behavior of the models when employing a mean-based and a sum-based aggregation method for the norm. This choice is mainly motivated by the arguments presented in Appendix A regarding possible solutions to the vanishing gradient that arises when employing a `Sigmoid` activation. In this case, replacing the sum-based with a mean-based norm implies reducing the variance of the sigmoidal distribution, which mitigates the vanishing effect in some cases.

Finally, we investigate the impact of incorporating a lateral inhibition mechanism into the latent vector. These methods were introduced to increase sparsity in the latent vectors, thereby regulating activity and limiting the number of weights updated at each forward pass. This was achieved by implementing a *Winner-Takes-All* (WTA) dynamic using a top-k selection function. Specifically, this function resets to zero all elements in a vector that are not among the k most active elements. For all experiments, the value for k was set to 15 to ensure that only a small subset of neurons was active. It is important to remark that the lateral inhibition scheme is only directly applied to the goodness function. However, in order to limit the information loss between layers, the non-inhibited latent vector is passed to subsequent layers. A summarized version of the different components of the goodness function is given in Table D2.

Table D2. List of the goodness components that are considered in the training experiments related to RQ1. Each goodness function was composed by employing a norm, an aggregation method and a lateral inhibition mechanism. Each column lists all the methods for each component.

Goodness configuration		
Norm function	Aggregation method	Lateral inhibition
$\|\|\boldsymbol{\ell}\|\|_2^2$, $\|\|\boldsymbol{\ell}\|\|_2$, $\|\|\boldsymbol{\ell}\|\|_1$	Mean, sum	No inhibition, WTA inhibition

E Effect on the Ratio of Positive and Negative Neurons

This appendix provides further analysis of the performance obtained when employing different relations of positive to negative neurons. To provide a systematic analysis of distinct scenarios, we examined both balanced and highly disproportionate polarity distributions. Specifically, we analyzed networks with the following percentages of positive neurons at each layer: 5%, 10%, 25%, 50%, 75%, 90%, and 95%. We trained multiple networks using the same neural configurations described in Appendix D, except for those using WTA mechanics. These experiments were constrained to grayscale datasets: MNIST, Fashion-MNIST, and K-MNIST. All networks were trained for 10 epochs using the same set of hyperparameters as in the RQ1 experiments. Since the only metric of interest in this appendix is the variation in accuracy across the different positive-to-negative splits, we normalized the results of each configuration relative to their mean accuracy.

The results of these experiments are depicted in Figure E1. Overall, they demonstrate that most neural configurations achieve comparable accuracy, with only a small subset of outliers exhibiting a statistically significant difference in accuracy. This finding suggests that Polar-FFA networks possess self-regulatory dynamics, where the activation strength of the two polarity sets adjusts to surpass the other set when presented with inputs of their respective polarity. However, when examining the general behavior of the outliers, a clear tendency emerges: they perform better with a balanced distribution of positive and negative neurons, while showing reduced accuracies at the extremes of polarity distribution.

From an experimental standpoint, given the observed results, it appears clear that the best polarity distribution within the layers is achieved with a near-balanced configuration. Consequently, this configuration was used in the experiments conducted for both RQ1 and RQ2.

However, to gain further insights into this set of outliers, we present evidence on the circumstances under which these outliers

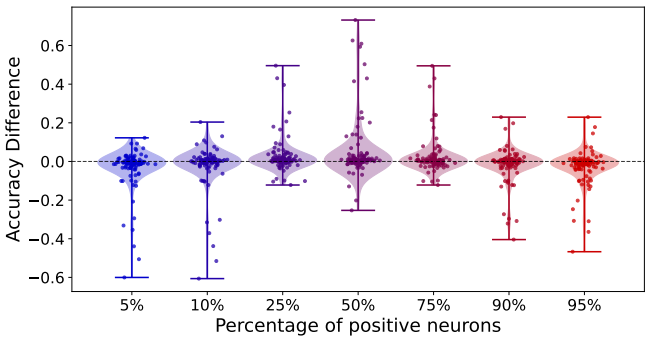

Figure E1. Distribution in the difference in accuracy between the distinct percentages of positive to negative neurons to the mean accuracy of the different percentages.

emerge. Figure E2 provides a disaggregated view of the results, showing the activation and probability functions used during training. For clarity, we exclude the set of *stable configurations*, as they do not provide relevant information regarding the outliers. This band of configurations ranges within an accuracy difference of $[-0.08, 0.08]$, encompassing 85% of the most stable experimental results.

Effects of the Probability Function. Upon early inspection, the Symmetric probability function appears to produce more stable results, with only a small set of outliers exhibiting a characteristic behavior: all result from using the `Tanh` activation function with a high percentage of positive neurons. In this scenario, the model seems incapable of balancing the high positivity of the large neural dataset with the reduced negative neural set. We believe this asymmetry in the results arises from the wider negative data distribution, which arises from the combination of input samples with all non-corresponding labels, compared to the sharper distribution of positive samples. In contrast, most outliers appear when employing the Sigmoidal probability function, especially when using bounded activation functions. Unlike the Symmetric probability, these outliers show a sharper decrease in accuracy when the positive neural set is smaller than the negative set. This effect can be attributed to the inability of the positive neural set to achieve a high enough goodness score to push the probability function towards non-vanishing output ranges.

Effects of the Activation Function. Out of the three functions, `ReLU` demonstrates the most competitive performance, with its few outliers highly concentrated in balanced polarity distributions, showing only a small accuracy difference of ± 0.20. `Tanh` exhibits a clear preference for near-equilibrium ratios of positive to negative neurons, often resulting in a negative accuracy difference when the positive-to-negative neural distributions are imbalanced. Similarly, `Sigmoid` performs better with even polarity distributions, slightly improving its results in configurations where the positive neurons outnumber the negative ones. In cases with a very low percentage of positive neurons, `Sigmoid` yields reduced accuracy, which can be attributed to the positive neurons' inability to achieve high goodness scores necessary to push the probability function into stable ranges. Among the two bounded functions, `Tanh` shows a less drastic drop in accuracy compared to `Sigmoid`, with most outliers remaining within the ± 0.40 range, while `Sigmoid` reaches the ± 0.60 range.

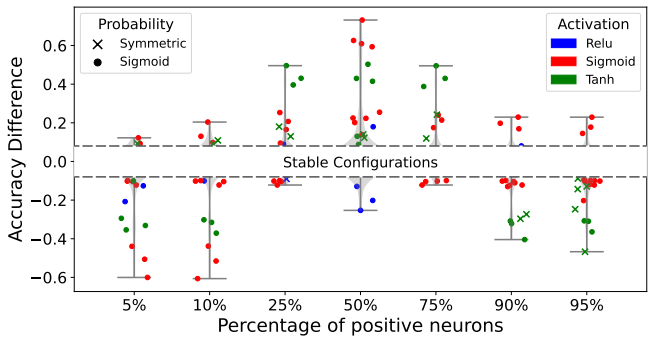

Figure E2. Plot showing the accuracy difference of the outlying elements, categorized based on the probability function and the activation function used. Points with less than 0.08 absolute accuracy difference, denoted as *stable configuration band*, have been removed to improve clarity.

F Additional Results for RQ1: goodness, probability functions and activations

In this appendix we present additional results for answering RQ1, broken down based on different configuration parameters. Table F1 shows the average accuracy scores of the different goodness functions over MNIST-like datasets, whereas Table F2 presents the average accuracy focusing solely on the activation function. Finally, the complete set of results for each neural combination and dataset is presented in Table F3 (shown in the next page).

Table F1. Mean accuracy of each goodness configurations averaged over `ReLU`, `Sigmoid` and `Tanh` activated models and all MNIST-like datasets. The highest average accuracy is highlighted in bold.

Norm	Aggregation	Inhibition	P_s	P_σ	P_σ^{FFA}
$\|\|\boldsymbol{\ell}\|\|_2^2$	Sum	–	83.80	81.28	59.08
		WTA	90.75	91.67	60.89
	Mean	–	84.05	68.37	66.26
		WTA	**90.76**	72.56	41.97
$\|\|\boldsymbol{\ell}\|\|_2$	Sum	–	83.41	87.14	55.97
		WTA	89.53	**91.96**	60.49
	Mean	–	82.87	61.88	63.99
		WTA	89.29	61.30	34.86
$\|\|\boldsymbol{\ell}\|\|_1$	Sum	–	81.34	90.33	50.01
		WTA	88.14	90.16	**83.45**
	Mean	–	81.21	67.63	65.15
		WTA	87.91	67.89	55.99

One straightforward observation from the results of the different neural configurations is a notable increase in accuracy when employing WTA inhibition dynamics on models using the symmetric probability P_s. However, this effect seems to be more dependent on the aggregation strategy when using the other probability functions. Models tend to achieve higher accuracy levels when employing a sum-based aggregation, while this score drops when a mean-based aggregation is used. We hypothesize that this effect relates to the activity bounds resulting from each neural configuration. When reducing the activity with WTA dynamics on mean-based goodness functions, it yields low goodness scores, restricting the behavior of the probability and thereby leading to degraded accuracy. Conversely, the effect is reversed in the sum-based score, where WTA reduces the variance of high-valued goodness scores. Surprisingly, the average accuracy over the three activation functions in FFA appears to achieve maximal accuracy when employing the L_1 norm, in contrast with the original approach which advocated for the usage of the squared Euclidean norm. Nevertheless, all models achieving the highest accuracy, which were used to produce the results in Table 3, employed the $\|\cdot\|_2$ norm.

Table F2. Mean accuracy of each activation function on each probability function averaged over the MNIST, KMNIST, Fashion-MNIST and CIFAR-10 datasets.

Activation	P_s	P_σ	P_σ^{FFA}
`ReLU`	87.76	**87.04**	**84.04**
`Sigmoid`	**90.78**	69.41	40.45
`Tanh`	79.72	76.59	50.03

When it comes to the results discussed in Section 5, we observe that `ReLU` achieves the highest generalization score in models trained using P_σ (Polar-FFA) or P_σ^{FFA} (FFA). Both the `Sigmoid` and the `Tanh` activation functions are found to achieve lower accuracy scores. In contrast, when examining the results of the P_s probability, the opposite appears to hold, with `Sigmoid` activation emerging as the best-performing activation function. Additionally, as hinted by previous results, the difference between the average accuracy of the different activation functions is minimized when using the symmetric probability P_s, while P_σ^{FFA} produces the highest variance in the reported performance figures.

G Additional Results for RQ2: Latent Space Taxonomy

This appendix provides an extensive overview of the various latent structures that emerge depending on the neural configurations used to train the models. As presented in the results of RQ1 in Subsection 5, neurons do not behave uniformly across the different configurations employed during training, which results in different generalization capabilities. This distinct dynamics appear to be closely linked to the geometry manifested in their latent space, resembling the strategies employed by the models to attain their optimal accuracies. The phenomenon of latent spaces acquiring a geometric structure as learning progresses was already observed by Tosato et al. [19] and by Ororbia & Mali [16], however, this work extends their analysis to a broader range of activation and probability functions, aiming to showcase the diverse dynamics that FFA-like algorithm can exhibit.

Given the extensive set of models trained for this study, totaling over 500 models, we conducted a qualitative analysis on their latent space. From this analysis, we selected a small subset of representative spaces that illustrate the features characterizing their respective learning dynamics. Our analysis focused on models achieving more than 50% accuracy to ensure clarity in the observed geometric patterns. Additionally, for each characteristic latent space described, we provide a brief discussion analyzing the dynamics contributing to the emergence of these latent structures. A detailed depiction of the different latent spaces is provided in the supplementary material.

Throughout our analysis, we identified five distinct latent structures generated by FFA-like algorithms, each driven by a specific set of neural configurations. The classification of these latent structures is as follows:

Original FFA The neural configuration of the original FFA involved using the L_2 norm over a network composed of ReLU-activated neurons, integrated with a sigmoidal probability function using a threshold value of $\theta = 2$. The resulting geometric structure is

characterized by a large cluster of points neighboring zero, representing samples detected as negative, alongside a set of clusters located farther away, each comprising points classified by the model as positive (see Figure G1). Each positive cluster predominantly consists of points belonging to distinct classes, aligning with the effect observed by Tosato et al., where latent vectors of datapoints from the same class converged into similar representations [19]. This trend does not seem to be exclusive to this configuration, as most latent geometries appear to exhibit this neural specialization phenomenon, indicating that neurons tend to exhibit neural activity primarily when presented with samples from their respective class.

Figure G1. Latent space projection using T-SNE of a network trained with FFA using ReLU activations and a L_1 norm with WTA lateral inhibition.

Sigmoid-Activated FFA In contrast to FFA models trained using ReLU, those employing the sigmoid activation function fail to cluster negative samples within the neighboring area of 0, as depicted in Figure G2. This effect arises from the non-negative output range of the sigmoid, where achieving activities close to zero requires extremely low pre-activation values. Consequently, the learning dynamic of this model shifts the objetive from having negative latent vectors close to zero to aiming for positive samples with slightly higher goodness scores than negative ones. As a result, the difference in mean activity between positive and negative vectors becomes evident, while negative vectors still produce latent vectors that closely resemble those generated by positive samples. Usually, the small clusters observed adhere to a correspondence where samples from the same pair of real class and label embedding group together. Unlike ReLU-based FFA networks, latent vectors in this category do not result in sparse vectors. Instead, they maximize their latent activity by activating a large set of neurons close to attaining their maximal value at each coordinate, as dictated by the upper bound of the sigmoid. This lack of sparsity also implies a lesser degree of neural specialization, as many neurons become specialized in a wide range of classes.

Perturbation-Based FFA Models associated with this latent geometry exhibit less predictable neural configurations. Currently, instances of this structure appear to emerge in cases utilizing the ReLU activation function together with a sum-based aggregation. This structure seems to arise from large values of the ReLU activation resulting in smaller weight changes, primarily focusing on the weights of the label. For example, such spaces can be observed in cases using the L_2 norm in MNIST but fail to appear when employing a Winner-Takes-All (WTA) inhibition mechanism, which drastically reduces layer activity. In the latter case, the resulting geometry aligns with what is expected for the original FFA.

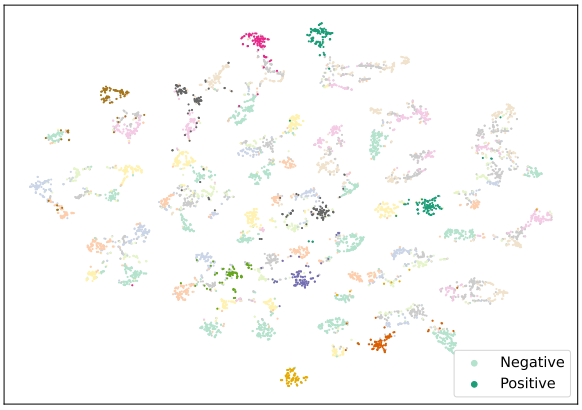

Figure G2. Latent space projection using T-SNE of a network trained with FFA using Sigmoid activations and a L_1 norm. Darker colors represent positive samples, while pale colors represent negative inputs.

The geometry arising under this category is characterized by showing no significant separation between positive and negative samples. As depicted in Figure G3, the latent space consists of numerous small clusters, corresponding to the total number of input samples. Additionally, each small cluster contains only 10 or fewer points, as shown in the small frame in the upper right corner, which presents a zoomed version of one of these clusters. This behavior can be attributed to the reduced impact of the label on the latent vector. In most models, the weights of the latent vector serve to heavily influence the latent in a certain direction. In these cases, the label weights are relatively low, and therefore, the latent is mainly positioned based on the latent generated by the input alone. Formally, we can view this effect as having two independent weight matrices $(\mathbf{W}_x, \mathbf{W}_l)$, where $\mathbf{W_x}$ represents the matrix that interacts with the input image $\mathbf{x}$ and W_l interacts with the embedded label $\mathbf{l}$. Since the latent vector is given by $f(\mathbf{W}_x\mathbf{x}^T + \mathbf{W}_l\mathbf{l}^T)$, if the matrix $\mathbf{W}_l$ has low-valued entries, the value will be approximately given by $f(\mathbf{W}_x\mathbf{x}^T + \mathbf{W}_l\mathbf{l}^T) \approx f(\mathbf{W}_x\mathbf{x}^T)$. However, models using this configuration still achieve good accuracy, as the small perturbation given by the value of $\mathbf{W}_l\mathbf{l}^T$ serves to point in the direction that maximizes the latent activity for each class.

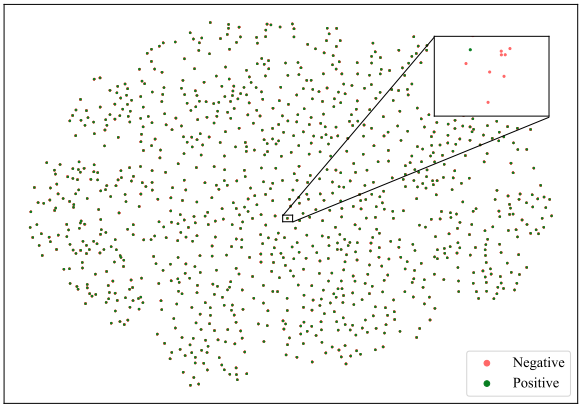

Figure G3. Latent space projection using T-SNE of a network trained with FFA using ReLU activations and a squared L_2 norm.

Sigmoid Probability Polar-FFA Similar to the original FFA, models trained using the sigmoid probability in Polar-FFA generally achieve highly clustered positive latent vectors, which also have a

high degree of separation with respect to negative samples. However, due to the additional objective of maximizing the negative neural set when being presented with a negative input, the negative set of latent vectors exhibits positive-like activity. In this case, the information loss created by sending all negatives to zero is replaced by large clouds of latent points, each containing information about the given negative sample. While the cluster structure makes this latent structure partially similar to the one seen in FFA models trained with sigmoid activations, the activity in the negative samples in these models does not overlap with the one present in the positive neuron group, resulting in a clearer separation between positive and negative samples and less overlap.

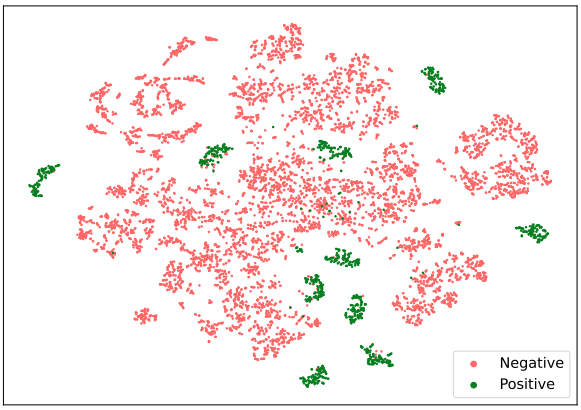

Figure G4. Latent space projection using T-SNE of a network trained with FFA using ReLU activations, the squared L_2 norm with WTA lateral inhibition, and the sigmoid probability function.

In contrast to the previous cases, where networks exhibited completely different activity when not using ReLU, this effect is not restricted to any specific activation function. For instance, similar behavior can be observed in networks trained using the Tanh activation or the Sigmoid function. These cases result in similar topologies, characterized by clusters based on real class and embedded label, akin to the sigmoid case of FFA. However, Tanh and Sigmoid appear to achieve tighter clustering, possibly due to a reduced variance in the latent vector caused by the bounded behavior of the activation functions.

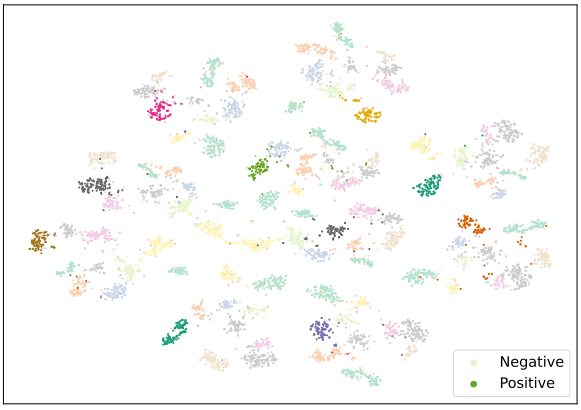

Figure G5. Latent space projection using T-SNE of a network trained with Polar-FFA using Tanh activations, the L_2 norm, and the sigmoid probability function. Each color represent the class of the input image. Darker colors represent positive samples, while pale colors represent negative inputs.

Symmetric Polar-FFA Models trained using the symmetric probability function in Polar-FFA appear to generate highly homogeneous latent spaces, as depicted in Figure G6, mostly independent of the activation or the goodness function. This homogeneous latent structure seems to correlate with the previously presented results indicating that these models have the highest generalization capabilities and the most robustness across different seeds. Similar to most other configurations, this probability function generates a high degree of separation between positive and negative samples. However, a noticeable clustering can be observed near the origin, where positive and negative points become clustered together. This clustering arises due to the scale-invariance of these models, which is not accounted for in this depiction. Employing a normalization scheme, as depicted in Figure G7, helps achieve a clearer separation between positive and negative samples. These clusters indicate that each class is related to a direction in the space, which is relative to points in the projective space.

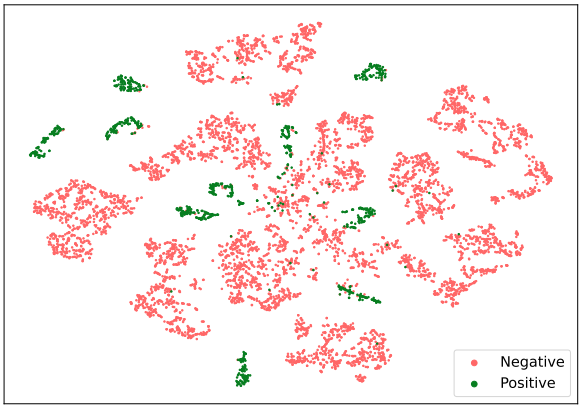

Figure G6. Latent space projection using T-SNE of a network trained with FFA using ReLU activations, the squared L_2 norm with WTA lateral inhibition, and the symmetric probability function.

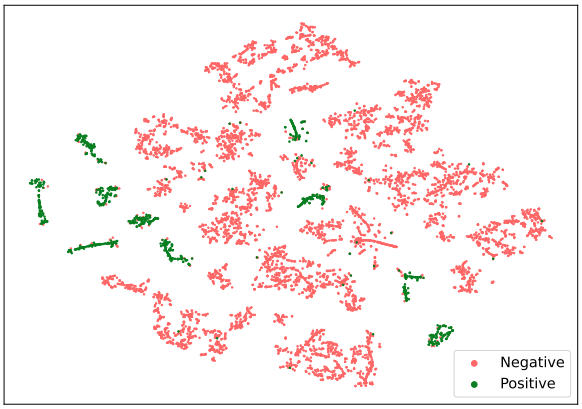

Figure G7. Normalized latent space projection using T-SNE of a network trained with FFA using ReLU activations, the squared L_2 norm with WTA lateral inhibition, and the symmetric probability function.

Table F3. Highest test accuracy obtained for each experiments employing the different neural combinations.

MNIST Dataset												
Norm	Aggregation	Inhibition	P_s			P_σ			P_σ^{FFA}			
			Sigmoid	ReLU	Tanh	Sigmoid	ReLU	Tanh	Sigmoid	ReLU	Tanh	
$\|\boldsymbol{\ell}\|_2^2$	Sum	-	97.30	96.31	79.30	43.70	87.62	97.61	9.80	96.06	91.57	
		WTA	97.27	95.79	96.94	98.32	97.37	97.98	9.80	95.91	95.86	
	Mean	-	97.41	96.52	80.37	52.84	96.87	75.25	89.64	95.99	14.02	
		WTA	97.58	96.15	97.01	64.15	97.07	86.27	11.56	94.10	7.60	
$\|\boldsymbol{\ell}\|_2$	Sum	-	96.32	91.20	80.49	96.53	98.26	96.19	9.80	95.68	75.40	
		WTA	96.44	91.99	96.60	95.94	98.15	96.76	79.21	97.33	93.91	
	Mean	-	96.34	91.21	80.89	52.64	93.20	76.29	88.08	93.77	9.94	
		WTA	96.53	91.61	96.64	65.39	93.17	75.02	90.56	91.39	4.07	
$\|\boldsymbol{\ell}\|_1$	Sum	-	97.23	95.23	82.36	97.85	73.82	97.21	9.80	93.42	88.24	
		WTA	97.13	94.51	96.58	98.10	98.25	97.59	9.80	95.00	93.72	
	Mean	-	97.15	94.48	82.04	43.94	91.42	73.69	85.18	90.88	12.76	
		WTA	96.77	93.43	96.59	69.84	85.05	60.88	9.75	75.35	7.61	
K-MNIST Dataset												
Norm	Aggregation	Inhibition	P_s			P_σ			P_σ^{FFA}			
			Sigmoid	ReLU	Tanh	Sigmoid	ReLU	Tanh	Sigmoid	ReLU	Tanh	
$\|\boldsymbol{\ell}\|_2^2$	Sum	-	87.98	87.89	52.20	89.59	87.61	87.64	10.00	81.84	67.27	
		WTA	89.69	87.86	88.01	90.58	87.99	89.80	10.00	77.15	79.10	
	Mean	-	87.80	88.54	52.99	39.02	90.35	42.18	63.52	83.72	39.76	
		WTA	89.20	88.10	87.81	37.29	86.97	67.54	8.60	79.29	10.39	
$\|\boldsymbol{\ell}\|_2$	Sum	-	86.77	79.13	51.53	84.51	91.50	82.89	10.00	84.59	46.12	
		WTA	88.83	76.94	86.04	82.54	91.57	85.25	59.80	87.60	77.88	
	Mean	-	85.85	77.29	52.05	58.28	80.86	37.58	59.59	74.21	42.41	
		WTA	87.58	76.59	85.86	35.01	80.59	53.95	67.50	72.71	9.21	
$\|\boldsymbol{\ell}\|_1$	Sum	-	88.05	86.63	51.56	88.35	89.67	87.58	10.00	71.90	58.80	
		WTA	89.09	82.41	86.15	90.41	91.07	88.86	10.00	79.14	77.02	
	Mean	-	86.76	84.95	52.57	23.78	76.45	40.04	61.43	66.66	40.66	
		WTA	87.73	84.25	86.26	29.60	69.41	48.30	10.30	59.58	9.30	
Fashion-MNIST Dataset												
Norm	Aggregation	Inhibition	P_s			P_σ			P_σ^{FFA}			
			Sigmoid	ReLU	Tanh	Sigmoid	ReLU	Tanh	Sigmoid	ReLU	Tanh	
$\|\boldsymbol{\ell}\|_2^2$	Sum	-	87.83	87.07	78.29	87.95	62.77	87.04	10.00	84.50	80.69	
		WTA	87.85	87.30	86.07	88.74	87.02	87.22	10.00	85.62	84.59	
	Mean	-	87.86	86.95	78.00	61.42	87.76	69.67	76.28	85.54	47.88	
		WTA	87.84	86.85	86.28	61.58	87.05	65.12	71.21	83.86	11.08	
$\|\boldsymbol{\ell}\|_2$	Sum	-	87.00	84.91	74.68	86.97	88.92	87.16	10.00	86.27	32.22	
		WTA	87.47	83.50	85.42	86.20	88.45	86.59	84.04	87.36	83.90	
	Mean	-	86.70	84.75	75.81	58.99	85.09	65.70	72.57	83.92	61.86	
		WTA	86.95	83.56	85.89	59.43	84.83	63.59	77.41	81.40	9.65	
$\|\boldsymbol{\ell}\|_1$	Sum	-	87.96	87.29	74.37	88.10	75.47	86.21	10.00	83.11	78.46	
		WTA	87.72	86.23	85.99	88.55	88.51	86.33	10.00	86.05	83.66	
	Mean	-	87.07	86.58	74.23	53.56	83.73	70.27	72.20	81.41	64.76	
		WTA	87.07	85.48	86.03	49.08	79.68	59.86	68.94	63.04	9.84	
CIFAR-10 Dataset												
Norm	Aggregation	Inhibition	P_s			P_σ			P_σ^{FFA}			
			Sigmoid	ReLU	Tanh	Sigmoid	ReLU	Tanh	Sigmoid	ReLU	Tanh	
$\|\boldsymbol{\ell}\|_2^2$	Sum	-	41.67	45.03	22.02	10.00	12.64	24.05	10.00	12.66	41.56	
		WTA	41.27	45.14	33.66	40.26	32.69	33.50	10.00	10.54	20.00	
	Mean	-	42.46	45.86	21.49	14.27	48.35	19.16	33.74	42.17	10.43	
		WTA	39.84	45.53	42.00	12.76	47.35	11.73	10.46	40.97	11.68	
$\|\boldsymbol{\ell}\|_2$	Sum	-	39.43	40.29	22.10	14.75	18.41	34.14	10.00	17.20	25.51	
		WTA	36.17	42.52	29.14	37.93	47.71	31.04	12.74	16.17	28.24	
	Mean	-	39.82	42.59	20.92	10.00	49.92	13.81	20.84	42.81	10.43	
		WTA	39.23	43.89	34.30	11.06	45.77	12.83	10.44	39.14	12.55	
$\|\boldsymbol{\ell}\|_1$	Sum	-	42.99	46.39	12.61	10.00	10.03	26.72	10.00	40.06	10.00	
		WTA	40.14	44.32	32.69	40.34	23.77	38.36	10.00	17.03	11.51	
	Mean	-	43.28	45.43	12.37	14.21	46.51	9.38	29.26	40.72	10.53	
		WTA	40.15	44.15	36.05	11.00	34.73	14.68	10.10	28.46	12.33	