ECAI 2024
U. Endriss et al. (Eds.)

doi:10.3233/FAIA240706

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

Erik B. Terres-Escudero[a,*], **Javier Del Ser**[b,c] **and Pablo Garcia-Bringas**[a]

[a]University of Deusto, 48007 Bilbao, Spain
[b]TECNALIA, Basque Research & Technology Alliance (BRTA), 48160 Derio, Spain
[c]University of the Basque Country (UPV/EHU), 48013 Bilbao, Spain

Abstract. Forward-only learning algorithms have recently gained attention as alternatives to gradient backpropagation, replacing the backward step of this latter solver with an additional contrastive forward pass. Among these approaches, the so-called Forward-Forward Algorithm (FFA) has been shown to achieve competitive levels of performance in terms of generalization and complexity. Networks trained using FFA learn to contrastively maximize a layer-wise defined goodness score when presented with real data (denoted as positive samples) and to minimize it when processing synthetic data (corr. negative samples). However, this algorithm still faces weaknesses that negatively affect the model accuracy and training stability, primarily due to a gradient imbalance between positive and negative samples. To overcome this issue, in this work we propose a novel implementation of the FFA algorithm, denoted as Polar-FFA, which extends the original formulation by introducing a neural division (*polarization*) between positive and negative instances. Neurons in each of these groups aim to maximize their goodness when presented with their respective data type, thereby creating a symmetric gradient behavior. To empirically gauge the improved learning capabilities of our proposed Polar-FFA, we perform several systematic experiments using different activation and goodness functions over image classification datasets. Our results demonstrate that Polar-FFA outperforms FFA in terms of accuracy and convergence speed. Furthermore, its lower reliance on hyperparameters reduces the need for hyperparameter tuning to guarantee optimal generalization capabilities, thereby allowing for a broader range of neural network configurations.

1 Introduction

Biologically plausible algorithms are emerging as alternative learning approaches focused on addressing several well-known shortcomings inherent in the backpropagation algorithm (BP) [23]. Among them, forward-only learning techniques stand out in the recent literature by leveraging error-driven local learning, thereby solving the weight transport and update lock problems [16]. These algorithms replace the backward pass of BP with an additional contrastive forward pass, modulated by carefully crafted layer-specific loss functions. Due to their local design, these algorithms allow training neural networks with a reduced memory footprint, suitable for scenarios with non-centralized computing capabilities, such as edge computing [13, 2].

One of the most prominent algorithms within forward-only learning approaches is the so-called Forward-Forward Algorithm (FFA) [4]. FFA advocates for the concept of fitness to contrastively learn to discriminate between real (also referred to as *positive*) data and synthetic (corr. *negative*) data. In doing so, FFA aims to maximize a goodness score when the network processes positive data, while minimizing this score when predicting negative data. Several works published since its inception have shown that FFA performs competitively when compared to BP. Unfortunately, FFA still faces several downsides that hinder the ability of this algorithm to achieve optimal generalization bounds. The cause of this weakness is primarily attributed to the formulation of the probability function that determines how the fitness score modulates whether a sample belongs to the positive set, as it has been shown to showcase vanishing gradient behavior [11].

This work aims to advance towards addressing this issue by introduces Polar-FFA, a novel forward-only learning algorithm that extends the original FFA by incorporating neural polarization within each layer. This mechanism introduces the concept of positive and negative *neurons*, which are shown to enhance the expressiveness of the probability function mentioned previously. We assess the benefits of Polar-FFA through extensive experiments over image classification datasets, showing that our approach enhances both the generalization capabilities of the trained network and the convergence speed of the learning process. Additionally, Polar-FFA is proven to allow for a broader set of neural configurations, thereby increasing the flexibility to build neural architectures based on FFA-like algorithms, which is critical when using bounded activation functions.

The rest of the manuscript is structured as follows: Section 2 introduces relevant literature to place in context the contribution of this work. Next, Section 3 motivates and describes the proposed Polar-FFA, together with examples of alternative probability functions. Section 4 follows by posing research questions and the experimental setup used to inform their responses with evidence. Section 5 presents the obtained experimental results and discusses on the improvements and limitations of our method. Finally, Section 6 draws the main conclusions of the work and outlines potential research directions rooted in our findings here reported. In addition, the appendices and source code needed to execute the experiments, have been included into the supplementary material [17].

* Corresponding Author. Email: e.terres@deusto.es.

2 Related Work

Before proceeding with the description of Polar-FFA, we briefly overview prior work on forward-only learning and FFA, ending with a statement of the contribution of Polar-FFA to the state of the art:

Forward-only Learning BP is arguably the most widely used algorithm for training neural networks. However, a recent surge in neuro-inspired learning algorithms has gained momentum in the Artificial Intelligence community [23]. These algorithms aim at addressing well-known algorithmic weaknesses of other learners by studying the learning dynamics in biological brains, including the usage of sparse latent activity and local learning rules. As a result, neuro-inspired learning algorithms have achieved competitive generalization capabilities [7, 12]. Among them, forward-only learning techniques present a novel credit assignment mechanism heavily inspired by the learning dynamics present in Hebbian update rules. They replace the backward pass of BP with a secondary forward pass, used in a layer-wise manner, to contrastively learn relevant features from input data [16]. The first implementation of this technique can be attributed to Kohan et al. [8], whose approach involved connecting the obtained classification error with the input layer. This allows the network to forward this error during a second forward pass, updating the weights without employing backward connections. An alternative forward-only approach was developed by Dellaferrera & Kreiman in [3], where a novel error-driven update rule was proposed to modulate the input perturbation and contrastively train each layer.

Forward-Forward Algorithm Further within the family of forward-only learning algorithms, FFA is a recently proposed neuroinspired approach based on the maximization of a *layer fitness* [4]. In doing so, FFA resorts to a contrastive learning process, where models are trained to distinguish between real (*positive*) data and synthetic (*negative*) data. To this end, FFA requires the definition of i) a *goodness function*, which measures the fitness of a sample to belong to the positive set of data; and ii) a probability function, which is used to map the fitness scores to the range $\mathbb{R}[0,1]$. Formally, a goodness function $G : \mathbb{R}^n \mapsto \mathbb{R}[0,\infty)$ maps a latent vector $\boldsymbol{\ell} \in \mathbb{R}^n$ to a non-negative fitness value. Common choices for the goodness function in the literature include the square Euclidean norm:

$$G(\boldsymbol{\ell}) = \|\boldsymbol{\ell}\|_2^2 = \sum_{i=1}^{n} \ell_i^2, \tag{1}$$

where $\boldsymbol{\ell} = (\ell_1, \ldots, \ell_n)$. Building upon this goodness function, FFA utilizes a probability function $P : \mathbb{R}[0,\infty] \mapsto \mathbb{R}[0,1]$, enabling the use of probabilistic loss functions (e.g. binary cross entropy). In his seminal work, Hinton suggested using a sigmoidal function as this mapping, with a hyper-parameter θ that shifts the center of the distribution:

$$P(G(\boldsymbol{\ell})) = \sigma(G(\boldsymbol{\ell}); \theta) = \frac{1}{1 + e^{-G(\boldsymbol{\ell})+\theta}}. \tag{2}$$

Due to its layer-wise dynamics, FFA emerges as a highly competitive alternative to other learning algorithms, especially in scenarios where memory and energy are highly constrained. For example, this algorithm has found practical applications in two relevant edge systems: optical neural networks, achieving competitive accuracy with a reduced number of parameters [13]; and microcontrollers, enabling on-device training for multivariate regression tasks [2]. Some extensions of the algorithm have been proposed to incorporate greater biological plausibility, such as the integration of predictive coding heuristics [15], and its adaptation to spiking neural networks [14]. In their work on predictive FFA (PFFA) [15], Ororbia & Mali also highlighted an additional key property of models trained with FFA: the resulting latent space is composed of distinct clusters consisting of points of the same class. A similar effect was exposed by Tosato et al. in [19], underscoring the apparent sparsity of latent vectors and its inherent latent structure. This effect was further explored by Yang [22], who provided a mathematical foundation for this phenomenon in ReLU-based networks using squared Euclidean distance as a goodness function.

3 Proposed Polar Forward-Forward Algorithm

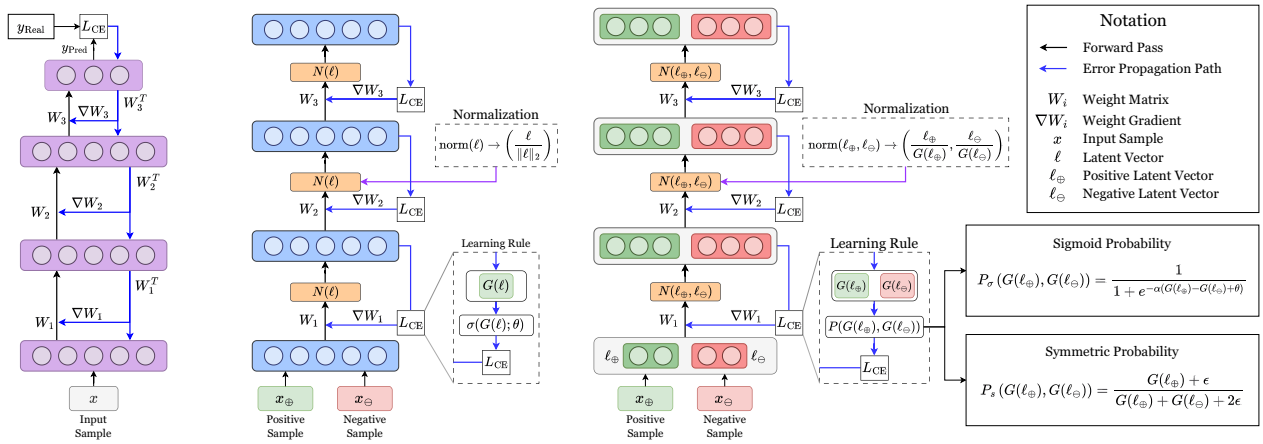
Polar-FFA introduces an extension to the FFA formulation by integrating a neural division where each neuron is assigned either a positive or negative polarization. The fundamental learning mechanism remains quite similar to FFA, as neurons within each set are trained to maximize their goodness score when exposed to samples of their corresponding polarity, and to minimize it when presented with the opposite polarity. For example, when employing a activity based goodness function, a positive neuron is expected to maximize its activity when presented with positive data, and to minimize it when presented with negative samples. However, due to this neural partitioning, the probability function measuring whether sample belongs to the positive set must be adapted from a single goodness score to a formulation including positive and negative goodness values. To provide a formal description of our algorithm, we recall the theoretical framework of FFA outlined in Section 2 for the sake of consistent notation and conceptual clarity. Since FFA-like algorithms train models on a layer by layer basis, the formulation of the proposed Polar-FFA focuses on the mechanisms involved in training a single layer. We hereafter denote the set of neurons in a given network layer as $\mathcal{L}$, so that $\boldsymbol{\ell}$ refers to the latent vector at the output of the layer at hand.

The definition of Polar-FFA departs from the assignment of a polarity to each neuron, depending on the expected goodness behavior desired for this neuron. Similarly to the original FFA, we define the subset of positive neurons as $\mathcal{L}_\oplus$, which aims at maximizing its goodness score when exposed to positive samples. The novel concept introduced in Polar-FFA is the negative neural set, denoted as $\mathcal{L}_\ominus$, which, in contrast to its counterpart, aims to maximize its goodness score when presented with negative samples. While the relative sizes of $\mathcal{L}_\oplus$ and $\mathcal{L}_\ominus$ can be arbitrarily specified whenever $\mathcal{L}_\oplus \cup \mathcal{L}_\ominus = \mathcal{L}$, for the sake of simplicity in this paper, we limit our discussion to scenarios where $|\mathcal{L}_\oplus| = |\mathcal{L}_\ominus|$. Under this split architecture, the goodness score is reformulated from a single scalar measuring the fitness of the input within the positive data distribution, to a pair of goodness scores, each measuring the suitability of the input with respect to the data distribution of their respective polarity. Since the goodness function only processes information contained in the latent vector, Polar-FFA can naturally consider the same set of goodness functions as those considered for FFA in the literature. Consequently, the goodness function $G : \mathbb{R}^n \mapsto \mathbb{R}[0,\infty) \times \mathbb{R}[0,\infty)$ evaluates each group independently as:

$$G(\boldsymbol{\ell}) = G(\boldsymbol{\ell}_\oplus \cup \boldsymbol{\ell}_\ominus) = \left(G(\boldsymbol{\ell}_\oplus), G(\boldsymbol{\ell}_\ominus)\right), \tag{3}$$

where $\boldsymbol{\ell} = \boldsymbol{\ell}_\oplus \cup \boldsymbol{\ell}_\ominus \in \mathbb{R}^n$ is the latent vector at the output of layer L, which contains n neurons, and $\boldsymbol{\ell}_\oplus$ ($\boldsymbol{\ell}_\ominus$) denote the activations corresponding to positive (negative) neurons in $\mathcal{L}_\oplus$ ($\mathcal{L}_\ominus$).

The second step in the adaptation from FFA to Polar-FFA involves replacing the scalar-based probability function with a probability function $P : \mathbb{R}[0,\infty) \times \mathbb{R}[0,\infty) \mapsto \mathbb{R}[0,1]$, which receives a pair of goodness scores at its input. This function should maximize its value as the discrepancy between positive and negative goodness



(a) BP. (b) FFA. (c) Proposed Polar-FFA.

Figure 1. Forward and backward propagation's paths on (a) Backpropagation (BP); (b) Forward-Forward Algorithm (FFA); and (c) Polar-FFA. Black lines denote the forward direction of the information flowing from the input through each of the networks. Blue lines indicate the error BP path, which has a local behavior in FFA and Polar-FFA. Additionally, the two adapted probability functions and the normalization process are included in the plot.

scores increases. Once this probability function is defined, Polar-FFA can be trained under the Binary Cross-Entropy (BCE) loss L_{CE} using an analogous training process as the standard FFA. Formally, the loss function of any forward-like supervised learning process is given by:

$$L_{\text{CE}} = -\sum_i \rho_i \log\left[P(G(\boldsymbol{\ell}_i))\right] + (1-\rho_i)\log\left[1 - P(G(\boldsymbol{\ell}_i))\right], \quad (4)$$

where $\rho_i \in \{0,1\}$ is a binary variable referring to the polarity of the data, taking value 1 if the sample is positive and 0 otherwise; and $\boldsymbol{\ell}_i$ extends the above notation to refer to the latent vector obtained by processing the i-th input. Using this loss function, weight updates at each step of Polar-FFA can be manually computed by applying the chain rule. Given a weight w_{ij} associated to a neuron that belongs to the positive neural set $\mathcal{L}_\oplus$, the loss gradient yields as:

$$\frac{\partial L_{\text{CE}}}{\partial w_{ij}} = -\sum_k \frac{1}{P(G(\boldsymbol{\ell}^k_\oplus), G(\boldsymbol{\ell}^k_\ominus))} \frac{\partial P}{\partial G_\oplus} \frac{\partial G_\oplus}{\partial \ell^k_{\oplus,j}} \frac{\partial \ell^k_{\oplus,j}}{\partial a^k_{\oplus,j}} x^k_i, \quad (5)$$

where we expand the notation of the latent vector further as $(\boldsymbol{\ell}^k_\oplus)_j = \ell^k_{\oplus,j}$ to denote the activation vector obtained from forwarding the input sample $\mathbf{x}^k$, $(\mathbf{a}^k_\oplus)_j = a^k_{\oplus,j}$ refers to the preactivation latent vector, and we employ the $\square_j$ subindex to refer to the j-th coordinate of the respective vector. Similarly, we abreviate $G(\boldsymbol{\ell}^k_\oplus)$ to $G_\oplus$. An equivalent expression of the loss can be obtained for the weight update rule of neurons in $\mathcal{L}_\ominus$ by replacing $\boldsymbol{\ell}_\oplus$ by $\boldsymbol{\ell}_\ominus$. It is important to note the heavy symmetry present in these expressions, as the behavior of neurons in the negative neural set with negative data corresponds to the same maximization objective as positive neurons within the positive dataset.

Characterization of FFA and Polar-FFA As shown in Expression (5), any update rule in a FFA-like algorithm is controlled by three key factors: the probability function, the goodness function, and the activation function. Therefore, understanding the behavior of any standard FFA-like model boils down to examining the specific behavior of these three functions. We refer as *network configurations* to any combination of activation, goodness and probability function used to define and train a specific model. Within this configuration, various choices are possible; for instance, any norm can serve as a goodness function, whereas any within the plethora of neural activation functions defined in the literature can be employed at each layer. This categorization has been crucial in providing a comprehensive set of neural configurations for the experimental setup, as is further detailed in Appendix D in the supplementary material [17].

Layer normalization Given the layer-wise learning dynamics of FFA and Polar-FFA, it is crucial to ensure that goodness information from previous layers is not overly influential in the decision making of subsequent layers. This issue was addressed in the original FFA work, where Hinton proposed introducing a normalization process between layers to equalize the goodness scores of all latent vectors [4]. Since our paper explores a broader set of goodness functions, this approach must be expanded to guarantee that this property is preserved for any such choice. Since the probability function in Polar-FFA only analyzes the relationship between positive and negative goodness values, the normalization scheme must involve updating the positive and negative latent vectors to yield equal goodness scores. This transformation ensures that latent vectors are treated equally by the probability function, removing any bias in subsequent layers. To avoid additional complexity, we restrict goodness functions to be absolutely homogeneous, meaning that $G(\lambda\boldsymbol{\ell}) = |\lambda| G(\boldsymbol{\ell})$ $\forall \boldsymbol{\ell} \in \mathbb{R}^n$ and $\forall \lambda \in \mathbb{R}$. Subject to this constraint, we can verify that the following normalization function $\texttt{norm} : \mathbb{R}^n \mapsto \mathbb{R}^n$ satisfies the sought normalization properties:

$$\texttt{norm}(\boldsymbol{\ell}_\oplus, \boldsymbol{\ell}_\ominus) = \left(\frac{\boldsymbol{\ell}_\oplus}{G(\boldsymbol{\ell}_\oplus)}, \frac{\boldsymbol{\ell}_\ominus}{G(\boldsymbol{\ell}_\ominus)}\right), \quad (6)$$

namely, the latent vectors corresponding to positive and negative neurons are normalized by their associated goodness value.

Definition of probability function Due to the distinct formulation of the probability function in Polar-FFA, the standard sigmoidal probability function cannot be directly employed to train the model. As previously discussed, the probability function in Polar-FFA must yield a probability based on the relationship between the pair of goodness scores $(G(\boldsymbol{\ell}_\oplus), G(\boldsymbol{\ell}_\ominus))$ at its input. High probability scores (values close to 1) should be produced when presented with scores satisfying $G(\boldsymbol{\ell}_\oplus) \gg G(\boldsymbol{\ell}_\ominus)$, and conversely, low probability values (close to 0) when $G(\boldsymbol{\ell}_\oplus) \ll G(\boldsymbol{\ell}_\ominus)$.

We present two alternative probability functions that leverage different relationships between positive and negative goodness scores, thus creating two distinct learning dynamics. Our first proposal expands the original sigmoid probability to incorporate negative goodness. The second approach involves computing the ratio of positive goodness to the total goodness score:

1. **Polar sigmoid probability** $P_\sigma(\cdot)$: The first proposed probability function extends the original sigmoidal probability in FFA by substituting the value $G(\boldsymbol{\ell})$ with the difference in scores $G(\boldsymbol{\ell}_\oplus) - G(\boldsymbol{\ell}_\ominus)$, thereby determining the probability based on the disparity between positive and negative goodness values. We denote this probability function as:

$$P_\sigma\left(G(\boldsymbol{\ell}_\oplus), G(\boldsymbol{\ell}_\ominus); \alpha, \theta\right) = \frac{1}{1 + e^{-\alpha \cdot (G(\boldsymbol{\ell}_\oplus) - G(\boldsymbol{\ell}_\ominus) + \theta)}}. \quad (7)$$

Similarly to the original sigmoid probability in FFA, α scales the difference in activity, whereas θ shifts the center of the function. This function exhibits a learning behavior similar to its predecessor, but incorporates two crucial properties. Firstly, since the function is defined by the difference of two opposite activities, the probability function is not biased towards the positive probability, thereby avoiding the gradient asymmetry identified by Lee et al. [11]. Secondly, as the split version does not directly employ the latent activity, the model's learning is independent of the mean of the latent activity, increasing the flexibility of the architecture. The effect of the mean on the learning dynamics is thoroughly examined in Appendix A in the supplementary material [17].
Notably, our formulation of $P_\sigma(\cdot)$ is solely influenced by the variance of the sigmoidal value of the difference between the positive and negative goodness values. Under these conditions, the mean value of the derivative can be guaranteed to be greater than any small number if the value of the difference in the goodness has low variance. This theoretical result implies that models with naturally low variance can learn, independently of their mean value. These insights are mathematically stated in Proposition 1, which is mathematically proven in Appendix B in the supplementary material [17]:

Proposition 1. *Let $\mathbf{W} \in \mathbb{R}^{n \times m}$ be a randomly initialized weight matrix, containing two independent sub-matrices of weights $(\mathbf{W}_\oplus, \mathbf{W}_\ominus)$ connecting an input $\mathbf{x}$ to their respective neural groups. Let $f(\cdot)$ be a continuous activation function, and let $\boldsymbol{\ell} = (\boldsymbol{\ell}_\oplus, \boldsymbol{\ell}_\ominus) = \left(f(\mathbf{W}_\oplus \mathbf{x}^T), f(\mathbf{W}_\ominus \mathbf{x}^T)\right) = f(\mathbf{W}\mathbf{x}^T)$ be the output vector of the layer. Let z be a random variable defined by $z = G(\boldsymbol{\ell}_\oplus) - G(\boldsymbol{\ell}_\ominus)$, such that $P_\sigma(G(\boldsymbol{\ell}_\oplus), G(\boldsymbol{\ell}_\ominus); \alpha, \theta) \equiv P_\sigma(z; \alpha, \theta)$. Then, for $\theta = 0$ and $\alpha = 1$:*

(a) The expected value of the derivative sigmoid probability function is determined by the variance of the sigmoid of z as:

$$\mathbb{E}\left[\frac{\partial P_\sigma(z)}{\partial G(\boldsymbol{\ell}_\oplus)}\right] = \frac{1}{4} - \mathrm{Var}\left[P_\sigma(z)\right] \geq 0. \quad (8)$$

(b) The expected value of the derivative is bounded below by the variance of z:

$$\mathbb{E}\left[\frac{\partial P_\sigma(z)}{\partial G(\boldsymbol{\ell}_\oplus)}\right] \geq \frac{1}{4} - \mathrm{Var}[z]. \quad (9)$$

2. **Symmetric probability** $P_s(\cdot)$: The second proposed probability function replaces the difference in goodness with the ratio between the positive goodness and the sum of the two scores:

$$P_s(G(\boldsymbol{\ell}_\oplus), G(\boldsymbol{\ell}_\ominus)) = \frac{G(\boldsymbol{\ell}_\oplus) + \epsilon}{G(\boldsymbol{\ell}_\oplus) + G(\boldsymbol{\ell}_\ominus) + 2\epsilon}, \quad (10)$$

where $\boldsymbol{\ell}$ follows the previously introduced notation for latent vectors, and ϵ is a small number introduced to avoid division by zero. The motivation behind adding ϵ in both the numerator and the denominator, with the denominator being doubled, is to ensure that the function remains symmetric for negative samples in latent vectors with low activity. This symmetry is guaranteed by the expression $P_s(G(\boldsymbol{\ell}_\oplus), G(\boldsymbol{\ell}_\ominus)) = 1 - P_s(G(\boldsymbol{\ell}_\ominus), G(\boldsymbol{\ell}_\oplus))$. Similar to the previously defined sigmoid probability function $P_\sigma(\cdot)$, this function maintains stable initial learning dynamics, ensuring that its gradient is nonzero during early training. Furthermore, a more robust theoretical result can be attributed to this function, as the update behavior is driven by the ratio between negative and positive activities, which is tightly related to the model's accuracy. For instance, the only points at which the derivative of this function is close to zero are when the model is remarkably accurate or when the values of the goodness score reach large values. The second situation can be mitigated by either clipping the goodness value or by including additional regularization terms into the loss function. Therefore, as opposed to models trained with FFA using the sigmoid function, models with low accuracy levels are ensured not to get stuck during the forward-only learning process regardless of their activity or variance, thereby making this function theoretically more robust. This property is formally expressed in Proposition 2, proven in Appendix C in the supplementary material [17].

Proposition 2. *Let $\boldsymbol{\ell} = (\boldsymbol{\ell}_\oplus, \boldsymbol{\ell}_\ominus)$ be a latent vector, and let $G(\cdot)$ be an arbitrary goodness function. The following properties hold:*

(a) $P_s(G(\boldsymbol{\ell}_\oplus), G(\boldsymbol{\ell}_\ominus))$ is approximately scale invariant for all $G(\boldsymbol{\ell}_\oplus)$ such that $G(\boldsymbol{\ell}_\ominus) \gg \epsilon$.

(b) Given two values $a, b \in \mathbb{R}[0, \infty)$ such that $a < G(\boldsymbol{\ell}_\oplus) + G(\boldsymbol{\ell}_\ominus) < b$, then the order of growth of the derivative is:

$$\frac{\partial P_s(G(\boldsymbol{\ell}_\oplus), G(\boldsymbol{\ell}_\ominus))}{\partial G(\boldsymbol{\ell}_\oplus)} = \mathcal{O}\left(\frac{G(\boldsymbol{\ell}_\ominus)}{G(\boldsymbol{\ell}_\oplus)}\right), \quad (11)$$

where $\mathcal{O}(\cdot)$ denotes asymptotic complexity.

4 Experimental Setup

To empirically assess the performance of our Polar-FFA approach, we formulate two Research Questions (RQs) that will be analyzed through an extensive set of experiments:

- **RQ1**: *Does neural polarization enhance the convergence and generalization with respect to the original FFA?*
- **RQ2**: *Which insights can be obtained by analyzing the latent space induced by neural polarization?*

To ensure that the results within RQ1 are not biased towards specific network configurations, exhaustive tests have been done with both FFA and Polar-FFA across a diverse range of architectural configurations. These configurations yield from the combinations of 3 different activation functions (`ReLU`, `Sigmoid`, and `Tanh`), 24 goodness functions $G(\cdot)$ (including $\|\cdot\|_2$ and $\|\cdot\|_1$, among others) and 3 probability functions, namely, the original sigmoid function used in FFA, hereafter denoted as $P_\sigma^{\text{FFA}}(\cdot)$, and the proposed $P_\sigma(\cdot)$ and $P_s(\cdot)$. The detailed list of network configurations utilized for experiments related to RQ1 is reported in Appendix D in the supplementary material [17].

The selected datasets for the experiments include MNIST [10], Fashion MNIST [21], KMNIST [1], and CIFAR-10 [9]. Models

trained on MNIST-like datasets (MNIST, Fashion MNIST, or KMNIST) use a 2-layer architecture comprising 1000 neurons each, whereas models trained on CIFAR-10 contain 2000 neurons per layer. In networks trained via Polar-FFA, the first half of the neurons of each layer are assigned to the positive neural group $\mathcal{L}_{\oplus}$ and the second half to the negative set $\mathcal{L}_{\ominus}$, as this polarity distribution can be empirically shown to be the most stable (see Appendix E in the supplementary material [17]). All models are trained using the ADAM optimizer with a learning rate of 0.001 for 50 epochs and a batch size of 512, except for CIFAR-10, in which the training process is extended to 100 epochs. Following [11], labels are embedded into images by concatenating them to the end of the image, using a Bernoulli distribution with probability 0.1 and a label pattern size of 100 pixels. Furthermore, we adopt Hinton's original approach to generate negative data instances [4], which involved embedding incorrect data (selected from the set of possible classes) to a real positive instance. Each (activation,goodness,probability) function combination is used to train the network on each dataset using FFA and Polar-FFA.

Each of these configurations is evaluated for accuracy and convergence speed. As for the latter, we rely on what we denote as *convergence area* (CA), defined as the area above the accuracy curve across epochs, upper bounded by the maximum accuracy attained over all epochs. Formally, given the accuracy levels $\mathbf{acc} = (\text{acc}_1, \ldots, \text{acc}_T)$ obtained by the learning algorithm at hand over T epochs, this score is given by:

$$\text{CA}(\mathbf{acc}) = \frac{1}{T \cdot \max_t \text{acc}_t} \sum_{i=t}^{T} \left[(\max_t \text{acc}_t) - \text{acc}_t \right]. \tag{12}$$

To evaluate the statistical significance of the performance gaps between FFA and Polar-FFA, we select a diverse set of neural configurations and trained them using 10 different seeds over the MNIST, KMNIST, and Fashion MNIST datasets. Accuracy measurements are conducted for all three experiments to calculate their generalization capability and its standard error of the mean. The selection of these configurations is independently performed for each probability function. Specifically, we select the two goodness functions with the highest average accuracy on MNIST-like datasets for each of the three activation functions.

RQ2 aims to offer deeper insights and arguments regarding the results obtained in response to RQ1. This analysis is conducted by exploring the latent space of models trained using the previously defined set of neural configurations. By extracting a representative sample of their latent space, we compute several geometric properties, primarily focusing on sparsity and separability indices. These indices are then inspected jointly with the accuracy and convergence speed of the trained network to analyze the relationship between the geometric properties of the latent space and the training dynamics of the networks. Moreover, we resort to T-SNE [20] as a dimensionality reduction algorithm to visualize the distribution of the latent space $\boldsymbol{\ell}$.

The sparsity of the latent space is quantified in terms of the Hoyer metric [5], which is known to meet most properties expected for sparsity metrics [6]. Given a latent vector $\boldsymbol{\ell} \in \mathbb{R}^n$, this metric returns a value between 0 and 1 measuring the sparsity of the vector, where 0 implies a uniform distribution and 1 a totally sparse vector. The Hoyer Index $\text{HI} : \mathbb{R}^n \to \mathbb{R}[0, 1]$ is defined as:

$$\text{HI}(\boldsymbol{\ell}) = \left(\sqrt{n} - (\|\boldsymbol{\ell}\|_1 / \|\boldsymbol{\ell}\|_2)\right) / \left(\sqrt{n} - 1\right). \tag{13}$$

Based on this index, we introduce a score denoted as *neural usage*, which measures how well distributed is the contribution of the different neurons to the latent output of the layer. We define it as the Hoyer index of the latent vector obtained by averaging over the output latent space. A highly sparse average latent vector would imply the existence of a large set of neurons that do not actively participate in the inference forward pass of the model, while low sparsity values would indicate that all neurons contribute equally to the prediction of the model. It is important to note that this score is particularly relevant for FFA-like methods, where the activity of neurons directly influences model predictions.

Similarly, separability focuses on the distance in the latent space between clusters of different classes. In FFA, this score quantifies the overlap between positive and negative latent vectors. To achieve this, we use the geometric separability index [18]. This metric analyzes the ratio of samples belonging to the same class as a given point within their nearest neighbors. Given a dataset composed of pairs of samples and labels, we denote the class of the k-th sample as C_k. Similarly, we denote the polarity type of its j-th nearest neighbor as $\text{KNN}_j(C_k)$. Values of separability closer to 1 imply that the different classes do not overlap, while values closer to 0 imply a total overlap of the classes. The separability index $\text{SI}(\boldsymbol{\ell})$ is computed as:

$$\text{SI}(\boldsymbol{\ell}) = \frac{1}{K \cdot J} \sum_{k=1}^{K} \sum_{j=1}^{J} \delta(C_k, \text{KNN}_j(C_k)), \tag{14}$$

where K represents the number of input samples, and J is the total number of neighbors selected, which we set to $J = 5$ for our experiments. Additionally, $\delta(x, y)$ denotes the Kronecker delta function, which outputs 1 if $x = y$ and 0 otherwise.

5 Results and Discussion

In this section, we present and discuss on the results obtained for each of the previously introduced research question:

RQ1: Does neural polarization enhance the convergence and generalization with respect to FFA?

The results of the average accuracy of the distinct models on MNIST, Fashion MNIST and KMNIST are presented in Table 1. Due to the large accuracy difference between these datasets and CIFAR-10, the results corresponding to the latter are presented in Table 2. However, the disaggregated results of all the experiments can be found in Table F3 in Appendix F in the supplementary material [17].

Table 1. Maximum, median, average and minimum values of the accuracy [%] of the different probability functions of FFA and Polar-FFA, averaged over the MNIST, Fashion-MNIST and KMNIST datasets. We also present the amount of configurations achieving an accuracy greater than 80% and the total number of configurations. Best results for each score in bold.

Score	P_s	P_σ	P_σ^{FFA}
Maximum accuracy	91.60	**92.89**	90.76
Median accuracy	**89.67**	86.29	74.76
Average accuracy	**86.09**	77.68	58.18
Minimum accuracy	**68.90**	40.43	7.64
# top-80% accuracy configurations/total	**30/36**	20/36	13/36
Convergence area top-80% configurations	**0.0122**	0.0177	0.0208

The results obtained for this first research question confirm our hypothesis regarding the improved performance of Polar-FFA compared to FFA. Firstly, analyzing the accuracy scores obtained for MNIST-like datasets in Table 1, it is evident that models trained using

Polar-FFA outperform those trained using FFA in terms of accuracy, especially in cases where the neural configuration renders the model incapable of learning, as detailed in Appendix A in the supplementary material [17]. Moreover, when comparing the results between P_σ and P_σ^{FFA} individually, a significant number of models outperform their FFA counterparts. In terms of robustness, Polar-FFA demonstrates a significant advantage over FFA, with more than a 30-point difference in average accuracy observed in the worst-performing network configuration. This difference showcases the importance of ensuring that the derivative of the probability achieves non-zero values, as highlighted in Proposition 1 and Proposition 2. Notably, while the highest generalization capabilities are attributed to P_σ, the symmetric probability function P_s achieves the most robust results, with a remarkable minimum average accuracy of 68.90% in MNIST-like datasets, surpassing both sigmoidal functions by more than 28 points of accuracy. Furthermore, this minimal accuracy value increases to 83.92% when using a lateral inhibition scheme, demonstrating high accuracy across all configurations. This trend is also reflected on the total number of models reaching the minimum accuracy of 80%, where FFA models are revealed to attain poor generalization capabilities. This effect highlights that the scale invariance of the probability function significantly alleviates the activity constraints required for sigmoid-like probability functions. Similarly, the theoretical effect stated in Proposition 1 is also observed in these results, as the minimum accuracy in MNIST-like datasets is significantly higher than random chance. In contrast, the lack of adaptability of FFA to the different latent activity distributions hinders the network's capacity to learn, making FFA highly ineffective when not employing ReLU-like networks (see Table F2 in Appendix F in the supplementary material [17]). Additionally, when considering the median convergence area (last row in the table), the symmetric probability P_s converges in almost half the time compared to both sigmoidal probabilities. However, this faster convergence is outweighed by the reduced maximum accuracy of models trained by Polar-FFA using this probability.

Similar conclusions can be drawn from the results obtained over CIFAR-10, which are shown in Table 2. Once again, Polar-FFA with P_σ is the best-performing approach. Likewise, using our symmetric probability defined in Expression (10) produces very robust models, achieving a median accuracy of 40.22% in comparison with the median accuracy of 12.70% scored by the standard FFA. When it comes to convergence speed, in this case the naive FFA algorithm scores best, followed by Polar-FFA with the symmetric probability P_s. Since the stopping criterion for the learning process is the same for all algorithms (early-stopping detailed in Appendix D in the supplementary material [17]), the significantly lower median accuracy of FFA (12.70%) compared to Polar-FFA (21.46%) suggests that FFA undergoes premature convergence.

Table 2. Maximum, median, average and minimum values of the average accuracy of the different probability functions of FFA and Polar-FFA over the CIFAR-10 dataset. We also present the amount of configurations achieving an accuracy greater than 35% and the total number of configurations. Best results marked in bold.

Score	P_s	P_σ	P_σ^{FFA}
Maximum accuracy	46.39	**49.92**	42.81
Median accuracy	**40.22**	21.46	12.70
Average accuracy	**37.14**	25.39	20.29
Minimum accuracy	**12.37**	9.38	10.00
# top-35% accuracy configurations/total	**26/36**	10/36	7/36
Convergence area top-35% configurations	0.0651	0.0529	**0.0469**

We proceed by analyzing the convergence speed of the sigmoid probability P_σ in Polar-FFA, which closely resembles the original probability function of FFA. Figure 2 depicts the difference in accuracy (horizontal axis) and convergence area (vertical axis) between Polar-FFA and FFA and the configurations using P_σ and P_σ^{FFA} achieving more than 70% accuracy in both cases. Differences in accuracy and convergence speed are predominantly inside the same quadrant, characterized by a positive difference in accuracy and a negative difference in convergence area. These results imply that incorporating neural polarization into the learning process leads to improvements in both accuracy and training speed.

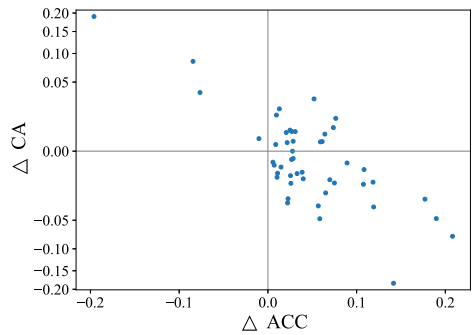


Figure 2. Distribution of the difference in convergence area ΔCA (vertical axis) and accuracy ΔACC (horizontal axis) between Polar-FFA with P_σ and FFA with P_σ^{FFA}. The vertical axis is in square-root scale for the sake of readability. Only models with an accuracy greater than 70% have been plotted, filtering those achieving fast convergence due to having suboptimal maximum accuracy.

To further investigate the impact of neural configuration on the variability between training runs, we present the results of the two best performing configurations for each activation function (`ReLu`, `Tanh` and `Sigmoid`) over 10 different seeds in Table 3. In this analysis, models trained using Polar-FFA (namely, those using P_s and P_σ) exhibit a lower standard deviation across different seeds, except when using the `Sigmoid` action together with P_σ. Moreover, the results evince a clear pattern regarding the choice of the activation function in sigmoidal probability functions: models trained using `ReLu` and `Tanh` activations yield less noisy accuracy levels than models trained with `Sigmoid` activations.

Table 3. Average accuracy and standard error of the mean over 10 runs of the best performing neural configurations with `ReLu`, `Sigmoid` and `Tanh` activations, over MNIST, Fashion-MNIST and KMNIST datasets. The highest average accuracy for each probability is highlighted in bold.

Activation, configuration	P_s	P_σ	P_σ^{FFA}
`ReLu`, best conf.	90.62 ± 0.76	$\mathbf{92.83 \pm 0.74}$	$\mathbf{90.66 \pm 0.87}$
`ReLu`, 2nd best conf.	90.72 ± 0.75	92.81 ± 0.73	88.61 ± 0.88
`Sigmoid`, best conf.	91.51 ± 0.79	80.76 ± 3.22	76.01 ± 0.79
`Sigmoid`, 2nd best conf.	$\mathbf{91.60 \pm 0.80}$	79.18 ± 3.21	71.79 ± 0.80
`Tanh`, best conf.	90.27 ± 0.89	91.52 ± 0.85	86.61 ± 1.25
`Tanh`, 2nd best conf.	90.31 ± 0.89	90.93 ± 0.93	79.73 ± 1.85

Out of all the results, the average accuracy remained consistent with the initial experiments, providing evidence of the robustness of FFA-like algorithms under carefully selected neural configurations. We note that experiments using the `Sigmoid` activation function give rise to a slight decrease in accuracy compared to the initially presented values in Table 1 when using the sigmoidal probability function P_σ. The high variance and the accuracy decrease associated

with the `Sigmoid` activation function can be attributed to a small subset of training sessions that yielded suboptimal performance, resulting in accuracy ranges between 40% and 60%. Nevertheless, the vast majority of the training seeds produced competitive accuracy levels. In contrast, when employing the symmetric probability this effect is reversed, with the `Sigmoid` activation achieving the most accurate results. Similarly, in addition to being the most robust when working with different network configurations, this probability function also achieves the lowest deviation in accuracy between different seeds.

RQ2: Which insights can be obtained by analyzing the latent space induced by neural polarization?

To address this second research question, we pause at Figure 3, which depicts the difference in accuracy and separability between Polar-FFA and FFA for models attaining an accuracy higher than 20%. Conceptually, FFA is designed to maximize the separation between positive and negative samples through a contrastive learning process. Our results reveal that this goal pursued by FFA is not fulfilled in all neural configurations. For instance, a small subset of neural configurations learns to contrast between positive and negative samples by small directional perturbations driven by the embedded labels. As shown in Appendix G in the supplementary material [17], this results in small clusters of points grouped closely, obtained from the same sample but with different embedded labels, but with the positive sample achieving a slightly greater goodness scores. The evidence of this geometrical structure points to a more diverse latent representation inherent to the algorithm, yet highly dependent on the choice of the neural configuration. Nevertheless, the major trend points towards a clear correlation between the distance between positive and negative latent spaces.

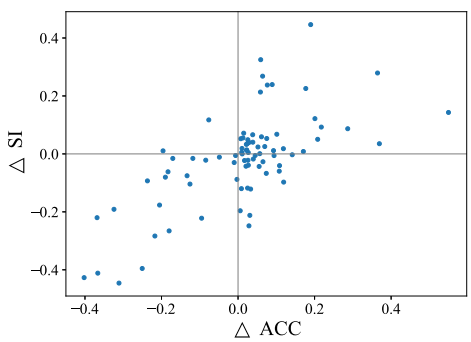


Figure 3. Distribution of the difference in the separability index ΔSI (y-axis) and accuracy ΔACC (x-axis) between Polar-FFA with P_σ and FFA with P_σ^{FFA}. Only models with accuracy higher than 20% are included, thereby filtering out near-random networks.

Sparsity also reveals information about the behavior of the model. As observed by the results in Table 4, models trained using `Tanh` as activation function result in the least sparse latent spaces, affecting the neural-level sparsity of most models except those trained using the symmetric probability function. As discussed in RQ1, these models were proven to perform most robustly, suggesting that the increased sparsity of the latent space can serve as a pivotal feature to guarantee high generalization capabilities in forward-like algorithms. In contrast, the sigmoid probability P_σ in Polar-FFA achieves the least sparse latent spaces. This can be explained by the difference in the objectives sought by FFA and Polar-FFA. While FFA aims at reducing the overall activity in negative samples, it tends to drive down most neurons during the negative phase. On the contrary, Polar-FFA aims at maximizing the activity of the negative neural set, which involves having a high number of active neurons during the same phase. This difference in functionality creates less sparse outputs, while at the same time improves the generalization capabilities and reduces the information loss between layers. It is also important to remark that while the sparsity of Polar-FFA is lower than that of FFA, both achieve high sparsity ratios. Additionally, the highest neural usage in Table 4 signifies that a highest concentration of information is confined in a small subset of neurons for models trained using the symmetric probability P_s. This leads to enhanced learning dynamics arising from a higher degree of neural specialization.

Table 4. Average Hoyer and neural usage of the configurations studied in RQ1. Due to the high difference between models using `Tanh` activations and those using other activation functions, we also present the average metrics of the configurations with that specific activation function excluded.

Metric	P_s	P_σ	P_σ^{FFA}
Hoyer index HI(ℓ)	0.9673	0.6430	0.7501
Neural usage	0.6219	0.2738	0.2074
Hoyer index HI(ℓ) (no `Tanh`)	0.9776	0.7275	0.8766
Neural usage (no `Tanh`)	0.8150	0.4554	0.4630

6 Conclusions and Future Research Lines

This work has introduced Polar-FFA, a novel formulation of the FFA that incorporates neural polarization to enhance its learning dynamics. Our approach involves dividing each layer into positive and negative neurons, each aimed at maximizing their goodness score when presented with inputs of their respective polarity. Building upon this formulation, we propose two alternative probability functions, proven to mitigate well-known limitations of the original FFA. Through extensive experiments across a diverse set of neural configurations, including various activation and goodness functions, we provide empirical evidence of the improved generalization capabilities of Polar-FFA. Significantly, our approach consistently outperforms FFA across all datasets and nearly all neural configurations in terms of accuracy and convergence speed. Furthermore, we demonstrate its ability to learn in a broader range of neural configurations, such as models using `Sigmoid` or `Tanh` activations, where the original FFA has been proven to perform poorly. In addition, we explore the geometrical properties inherent to this extended set of configurations, showing that the higher accuracy scores produced by Polar-FFA result from its capacity to learn highly separated latent representations. Similarly, our findings highlight the positive impact that latent sparsity provides during training, leading to more robust and stable learning dynamics.

We envision two main lines to further develop the ideas explained in this work. First, we intend to advance in the study of goodness and probability functions, focusing on their emerging geometrical properties. As shown in this work, the choice of these two functions highly impacts the properties of the latent space, which could be beneficial for creating more effective networks, especially in terms of robustness against out-of-distribution data and explainability. Second, we aim to extend the heuristics from FFA to more advanced neural architectures (e.g., CNNs or Transformers), primarily by replacing the supervised negative generation method for one compatible with non-dense layers.

Acknowledgements

The authors thank the Basque Government for its funding support via the consolidated research groups MATHMODE (ref. T1256-22) and D4K (ref. IT1528-22), and the colaborative ELKARTEK project KK-2023/00012 (BEREZ-IA). E. B. Terres-Escudero is supported by a PIF research fellowship granted by the University of Deusto.

References

[1] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

[2] F. De Vita, R. M. Nawaiseh, D. Bruneo, V. Tomaselli, M. Lattuada, and M. Falchetto. μ-FF: On-device forward-forward training algorithm for microcontrollers. In *IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 49–56. IEEE, 2023.

[3] G. Dellaferrera and G. Kreiman. Error-driven input modulation: Solving the credit assignment problem without a backward pass. In *International Conference on Machine Learning*, pages 4937–4955. PMLR, 2022.

[4] G. Hinton. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345*, 2022.

[5] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5(9), 2004.

[6] N. Hurley and S. Rickard. Comparing measures of sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, 2009.

[7] B. Illing, J. Ventura, G. Bellec, and W. Gerstner. Local plasticity rules can learn deep representations using self-supervised contrastive predictions. *Advances in Neural Information Processing Systems*, 34:30365–30379, 2021.

[8] A. A. Kohan, E. A. Rietman, and H. T. Siegelmann. Error forward-propagation: Reusing feedforward connections to propagate errors in deep learning. *arXiv preprint arXiv:1808.03357*, 2018.

[9] A. Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Toronto*, 2009.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324, 1998.

[11] H.-C. Lee and J. Song. Symba: Symmetric backpropagation-free contrastive learning with forward-forward algorithm for optimizing convergence. *arXiv preprint arXiv:2303.08418*, 2023.

[12] T. Moraitis, D. Toichkin, A. Journé, Y. Chua, and Q. Guo. SoftHebb: Bayesian inference in unsupervised hebbian soft winner-take-all networks. *Neuromorphic Computing and Engineering*, 2(4):044017, 2022.

[13] I. Oguz, J. Ke, Q. Weng, F. Yang, M. Yildirim, N. U. Dinc, J.-L. Hsieh, C. Moser, and D. Psaltis. Forward–forward training of an optical neural network. *Optics Letters*, 48(20):5249–5252, 2023.

[14] A. Ororbia. Contrastive-signal-dependent plasticity: Forward-forward learning of spiking neural systems. *arXiv preprint arXiv:2303.18187*, 2023.

[15] A. Ororbia and A. A. Mali. The predictive forward-forward algorithm. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 45, 2023.

[16] A. G. Ororbia. Brain-inspired machine intelligence: A survey of neurobiologically-plausible credit assignment. *arXiv preprint arXiv:2312.09257*, 2023.

[17] E. B. Terres-Escudero, J. Del Ser, and P. Garcia-Bringas. Supplementary material: On the improvement of generalization and stability of forward-only learning via neural polarization. https://github.com/erikberter/PolarFFA, 2024.

[18] C. Thornton. Separability is a learner's best friend. In *4th Neural Computation and Psychology Workshop: Connectionist Representations*, pages 40–46. Springer, 1998.

[19] N. Tosato, L. Basile, E. Ballarin, G. de Alteriis, A. Cazzaniga, and A. Ansuini. Emergent representations in networks trained with the forward-forward algorithm. *arXiv preprint arXiv:2305.18353*, 2023.

[20] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

[21] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[22] Y. Yang. A theory for the sparsity emerged in the forward forward algorithm. *arXiv preprint arXiv:2311.05667*, 2023.

[23] A. Zador et al. Catalyzing next-generation artificial intelligence through neuroAI. *Nature Communications*, 14(1):1597, 2023.