Are almost non-negative neural networks universal approximators?

Vlad Vasilescu, Ana Neacsu, Jean-Christophe Pesquet

2024 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 22–25, 2024, LONDON, UK

ARE ALMOST NON-NEGATIVE NEURAL NETWORKS UNIVERSAL APPROXIMATORS?

Vlad Vasilescu[1], *Ana Neacșu*[1], *Jean-Christophe Pesquet*[2]

[1] *Speech and Dialogue Laboratory*, *University Politehnica of Bucharest, Bucharest, Romania*
[2] *Université Paris-Saclay, CentraleSupélec*, *Centre de Vision Numérique, Inria, Gif-sur-Yvette, France*

ABSTRACT

Non-negatively weighted neural networks (NNs) have proven instrumental in various applications, offering interpretability and mitigating overfitting concerns. However, this advantage often comes at the expense of the expressivity of the model. In this paper, we show that almost non-negative neural networks allow us to waive this limitation. More specifically, we introduce a novel class of almost non-negative neural networks, that have a particular algebraic structure, for which we recover the universal approximation properties. Furthermore, to quantify the robustness of such a network architecture, we demonstrate the feasibility of deriving tight Lipschitz bounds, which are computationally efficient. To validate our approach, we conduct various classification experiments on a benchmark dataset of medical images. The results underscore the validity of our theoretical results.

Index Terms— neural networks, non-negative, Lipschitz, adversarial stability, universal approximation, explainability

1. INTRODUCTION

Although deep learning methods have garnered increasing attention in the scientific community owing to their remarkable ability to tackle complex tasks, they often face challenges such as interpretability, over-parameterization, and vulnerability to adversarial attacks [1]. One promising approach to mitigate these shortcomings is to introduce constraints in the architecture design. Specifically, imposing non-negativity constraints on the model weights can enhance interpretability by ensuring that weights represent only additive contributions, and reduce over-parameterization by limiting the search space. In this work, we will investigate the effect of imposing non-negativity constraints on the model weights and provide a framework for tightly estimating the Lipschitz bounds of the considered architectures.

There is a consensus that humans possess the innate capacity to dissect intricate interactions into distinct, intuitive hierarchical categories prior to analysis [2]. This evolutionary progression toward part-based representation in human cognition resonates with the concept of non-negativity restrictions on network weights [3]. This notion, among others, has spurred interest in neural networks featuring non-negative weights. In addition to aligning with human interpretability, the non-negativity constraint has been identified as a potent regularization mechanism, effectively curbing feature overfitting, particularly in scenarios with limited training data availability [4, 5]. Furthermore, recent research has demonstrated the feasibility of deriving a tight Lipschitz bound for such networks [5]. This Lipschitz constant serves as a valuable metric for quantifying network robustness, facilitating the design of networks with heightened resilience to adversarial perturbations [6, 7]. However, accurately computing this constant, even for modestly-sized networks, raises an NP-hard problem [8], and obtaining a reliable approximation in a reasonable timeframe remains an ongoing challenge for arbitrary-signed NNs.

The integration of non-negative restrictions on weights in neural networks, inspired by techniques such as non-negative matrix factorization (NMF), has ushered in a paradigm where hidden units correspond to discernible concepts. This innovative approach has proven adept at deriving meaningful representations, effectively revealing the intrinsic structure of high-dimensional data, as demonstrated by [9]. Additionally, it is noteworthy that the appealing convexity property [10] can be ensured under some assumptions on the activation functions. On a similar trajectory, monotone networks have also gained prominence for their interpretability and stability. It has been proven that monotonicity is a desirable property in real-life scenarios [11]. Interestingly, this monotonicity constraint aligns with the non-negative constraints imposed on weights in certain neural network architectures.

Despite their advantages, networks with non-negative weights may exhibit less expressivity than their counterparts with arbitrary signed weights. In a recent study [12], it is shown that standard non-negative networks are not universal approximators, a limitation that our work addresses.

In this paper, our interest lies in feed-forward neural networks featuring non-negative weights, with the exception of the initial and final linear layers. This category of networks naturally expands upon those where all linear layers possess non-negative values. In Section 3.1 we show that for a subclass of these neural networks we can recover universal approximation properties. Moreover, in Sections 3.3 and 3.4, we derive tractable and tight Lipschitz bounds for neural networks showcasing convolutional and fully-connected layers. In Section 4, we perform experiments on image classification problems to substantiate our theoretical findings.

979-8-3503-7225-0/24/$31.00 ©2024 IEEE

2. GENERAL BACKGROUND

2.1. Neural network model

We can mathematically express an m-layer perceptron as a composition of m operators:

$$T = T_m \circ \cdots \circ T_1, \tag{1}$$

where for every layer $i \in \{1, \dots, m\}$, $T_i : \mathbb{R}^{N_{i-1}} \to \mathbb{R}^{N_i} : x \mapsto R_i(W_i x + b_i)$. Here, $W_i \in \mathbb{R}^{N_1 \times N_{i-1}}$ is a weight operator and b_i is the bias vector in $\mathbb{R}^{N_i}$. Note that, in our formulation, $(W_i)_{1 \leqslant i \leqslant m}$ can also be chosen as a MIMO convolutive operator. $R_i : \mathbb{R}^{N_i} \to R^{N_i}$ is a non-linear (activation) function, applied element-wise on the input vector. Despite the range of activation operators available, studies like [7] have demonstrated that common R_i choices are α_i-averaged operators, with $\alpha_i \in]0, 1]$. Moreover, in many instances, R_i is found to be the proximity operator of a convex function, with α_i equal to $1/2$. In this work, we will further assume that R_i is symmetric, which means that there exists a point of symmetry in its graph. While some activations like hyperbolic tangent are symmetric, the classic ReLU operator is not. However, a capped version can be employed to meet this symmetry requirement [13].

2.2. Lipschitz bounds of neural networks

Considering a small perturbation $z \in \mathbb{R}^{N_0}$, an *adversarial example* is defined as $\tilde{x} = x + z$, such that $T(x) \neq T(\tilde{x})$. A key insight when considering adversarial perturbations is that the original input x and its adversarial $\tilde{x}$ should be as close as possible, given a similarity measure. The minimal perturbation required to achieve an adversarial example is usually quantified by $\|z\|$.

If we consider θ_m a Lipschitz constant of the network, we can bound the effect of z by the following inequality:

$$\|T(x + z) - T(x)\| \leqslant \theta_m \|z\|, \tag{2}$$

showing that θ_m is intimately correlated with the robustness of the model. Controlling this constant thus represents a feasible approach to limit the effect of possible adversarial attacks. Computing the exact Lipschitz constant of a neural network is however an NP-hard problem [6, 7], so the main challenge is to find efficient ways of approximating this constant effectively and to control it during the training phase without hindering the model performance. A recent study [14] shows how the Lipschitz behaviour of a network is influenced by the architectural choices as well as the initialization values of the weights. A standard separable upper bound of the Lipschitz constant is given by

$$\overline{\theta}_m = \prod_{i=1}^{m} \|W_i\|_{\mathrm{S}}, \tag{3}$$

where $\|\cdot\|_{\mathrm{S}}$ represents the *spectral norm*, i.e. the maximum singular value. Although easy to compute, this upper bound is often loose. Another variant, used in [15], leverages on the following spectral expression:

$$\theta_m = \sup_{x \in \mathbb{R}^{N_0}} \|\nabla T(x)\|_{\mathrm{S}}. \tag{4}$$

This bound scales relatively well even for larger networks. In practice, however, the supremum must be computed by sampling the input space, which means that the bound no longer offers strict theoretical guarantees. If the activations R_i are separable and α_i-averaged, an accurate bound for the Lipschitz constant can be derived (see for example [6, 7]) based on averaging properties:

$$\vartheta_m = \sup_{\substack{\Lambda_i \in \mathscr{D}_{N_i}(\{1-2\alpha_i, 1\}) \\ i \in \{1, \dots, m\}}} \|W_m \Lambda_{m-1} \cdots \Lambda_1 W_1\|_{\mathrm{S}}, \tag{5}$$

where $\mathscr{D}_{N_i} = \{\mathrm{Diag}(\lambda_1, \dots, \lambda_{N_i}) \mid (\lambda_i)_{1 \leqslant i \leqslant N_i} \in I^{N_i}\}$. in this case, computing ϑ_m is an NP-hard problem, and even though some simpler relaxations can be imposed, they are limited to small networks. It is worth noting that the following result holds:

Proposition 2.1 [7] *Assume that, for every layer $i \in \{1, \dots, m-1\}$, the activation operator R_i applies componentwise and is α_i-averaged (except for the last layer where it is just assumed nonexpansive). Let $\underline{\theta}_m$ be the Lipschitz constant of the associated linear network where operators $(R_i)_{1 \leqslant i \leqslant m}$ are identity. Then,*

$$\underline{\theta}_m = \|W_m \dots W_1\|_{\mathrm{S}} \leqslant \vartheta_m \leqslant \overline{\theta}_m \tag{6}$$

In addition if, $\forall\, i \in \{1, \dots, m\}$, $W_i \geqslant 0$, then $\vartheta_m = \underline{\theta}_m$.

In other words, if we use non-negative linear weights, we have a cheap way of computing the Lipschitz constant. This result makes non-negatively weighted neural networks appealing tools for designing robust systems. However, there exists no universal approximation theorem for such neural networks [12]. As a direct consequence of the results in the next section, we will see that we can design almost non-negative architectures that do not suffer from such an expressivity issue. By "almost" we mean that all the layers of these networks except the first and the last one possess non-negatively valued weights.

3. ABBA NETWORKS

In this section, we will focus on a subset of networks characterized by a specific weight matrix structure. These networks, referred to as ABBAnets [13], offer several algebraic advantages, allowed through the following weight matrix form:

Definition 3.1 Let $(N_1, N_2) \in (\mathbb{N} \setminus \{0\})^2$. $\mathcal{A}_{N_1, N_2}$ is the space of ABBA matrices of size $(2N_2) \times (2N_1)$, that is

$M \in \mathcal{A}_{N_1,N_2}$ if there exist matrices $A \in \mathbb{R}^{N_2\times N_1}$ and $B \in \mathbb{R}^{N_2\times N_1}$ such that

$$M = \begin{bmatrix} A & B \\ B & A \end{bmatrix}. \tag{7}$$

We also define the associated sum matrix as $\mathfrak{S}(M) = A + B$.

$\mathcal{A}_{N_1,N_2}$ is a vector space for which appealing properties are satisfied, two of them being listed below:

(i) If M has non-negative elements, the spectral norm of M is $\|\mathfrak{S}(M)\|_{\mathrm{S}}$.

(ii) The projection onto the spectral ball of center 0 and radius $\rho > 0$ of M is an ABBA matrix.

3.1. Extension to neural networks

Let us extend these results to neural networks. We propose to use weights which are ABBA matrices, except for the first and the last layers. Basically, the first layer maps the input to a twice-higher dimensional space, while the last layer performs a dimension reduction by a factor 2. This is described more precisely hereafter.

Definition 3.2 $\widetilde{T}$ is an ABBA m-layer network if

$$\widetilde{T} = (\underbrace{\widetilde{W}_{m+1}}_{N_m\times(2N_m)} \cdot + \widetilde{b}_{m+1})\widetilde{T}_m \cdots \widetilde{T}_1 \underbrace{\widetilde{W}_0}_{(2N_0)\times N_0} \tag{8}$$

where each layer $i \in \{1,\dots,m\}$ corresponds to the operator $\widetilde{T}_i = \widetilde{R}_i(\widetilde{W}_i \cdot + \widetilde{b}_i)$, where $\widetilde{R}_i$ is a non-expansive activation function operating in twice higher dimension than the operator R_i defined in Section 2.1, and $\widetilde{W}_i \in \mathbb{R}^{(2Ni)\times(2Ni-1)}$ is an ABBA matrix, while $\widetilde{W}_{m+1}$ and $\widetilde{W}_0$ are arbitrary-signed.

In addition, $\widetilde{T}$ is called a *non-negative ABBA* m-layer network if it satisfies the condition of being an ABBA m-layer network and if its ABBA matrices $(\widetilde{W}_i)_{1\leqslant i\leqslant m}$ have non-negative elements. In this work we will be mainly interested in such non-negative structures.

Let T be the neural network defined in Section 2.1. Next, we show that if the activation functions are symmetric, $\widetilde{T}$ is identical to T in terms of input-output relation, for judicious choices of the biases. For each layer $i \in \{1,\dots,m\}$ of T, let W_i^+ be the positive part of weight matrix W_i. We also denote by $W_i^- = W_i^+ - W_i \in [0,+\infty[^{N_i\times N_{i-1}}$ the non-negative part of W_i. Now we can define the non-negative ABBAnet associated with T, as follows:

$$\widetilde{W}_0 = \begin{bmatrix} I_{N_0} \\ -I_{N_0} \end{bmatrix}, \qquad \widetilde{W}_{m+1} = \frac{1}{2}[I_{N_m} \quad -I_{N_m}], \tag{9}$$

and, for every $i \in \{1,\dots,m\}$,

$$\widetilde{R}_i\colon \begin{bmatrix} x \\ z \end{bmatrix} \mapsto \begin{bmatrix} R_i(x) \\ R_i(z) \end{bmatrix}, \qquad \widetilde{W}_i = \begin{bmatrix} W_i^+ & W_i^- \\ W_i^- & W_i^+ \end{bmatrix}. \tag{10}$$

In addition, suppose that, between the biases b_i and $\widetilde{b}_i$, the following relation holds

$$(\forall i \in \{1,\dots,m\}) \quad \widetilde{b}_i = \begin{bmatrix} b_i - W_i^- d_{i-1} \\ c_i - b_i - W_i^+ d_{i-1} \end{bmatrix}, \tag{11}$$

$$\widetilde{b}_{m+1} = -\frac{d_m}{2}, d_0 = 0. \tag{12}$$

Then, it can be checked that, for every input, $\widetilde{T}$ has an identical output to T.
This result has a fundamental implication: it demonstrates that non-negative ABBAnets possess the same expressivity as standard feed-forward neural networks. Consequently, all classical universal approximation results [12] for standard neural networks extend to non-negative ABBAnets, and more broadly, to the class of networks with all their weights non-negative, except for the first and last layers [13]. The sole architectural limitation to meet this universal approximation properties is the symmetry of the activation operator. However, symmetric variants of current popular non-linear operators can be chosen.

3.2. Convolutional case

The current framework can be also extended to the case of convolutional layers. $\mathcal{W}_i$ is such a convolutional ABBA layer with $(2\zeta_{i-1})$ input channels, $(2\zeta_i)$ output channels, and stride s_i if its output signals $(\widetilde{y}_q^+, \widetilde{y}_q^-)_{1\leqslant q\zeta_i}$ are associated to the input signals $(\widetilde{x}_p^+, \widetilde{x}_p^-)_{1\leqslant p\leqslant\zeta_{i-1}}$ by

$$\widetilde{y}_q^+ = (\sum_{p=1}^{\zeta_{i-1}} w_{i,q,p}^+ * \widetilde{x}_p^+ + \sum_{p=1}^{\zeta_{i-1}} w_{i,q,p}^- * \widetilde{x}_p^-)_{\downarrow s_i} \tag{13}$$

$$\widetilde{y}_q^- = (\sum_{p=1}^{\zeta_{i-1}} w_{i,q,p}^- * \widetilde{x}_p^+ + \sum_{p=1}^{\zeta_{i-1}} w_{i,q,p}^+ * \widetilde{x}_p^-)_{\downarrow s_i}, \tag{14}$$

where $(w_{i,q,p}^+)_{1\leqslant p\leqslant\zeta_{i-1},1\leqslant q\leqslant\zeta_i}$ and $(w_{i,q,p}^-)_{1\leqslant p\leqslant\zeta_{i-1},1\leqslant q\leqslant\zeta_i}$ are convolution kernels operating on d-dimensional signals.

We can also define the convolutional layer in a matrix form. We introduce the MIMO impulse response associated with a non-negative ABBA convolutional layer $\mathcal{W}_i$: $\forall n \in \mathbb{Z}^d$,

$$\widetilde{W}_i(n) = \begin{bmatrix} W_i^+(n) & W_i^-(n) \\ W_i^-(n) & W_i^+(n) \end{bmatrix} \in [0,+\infty[^{(2\zeta_i)\times(2\zeta_{i-1})} \tag{15}$$

with

$$W_i^{+/-}(n) = \begin{bmatrix} w_{i,1,1}^{+/-}(n) \cdots w_{i,1,\zeta_{i-1}}^{+/-}(n) \\ \vdots \qquad\qquad \vdots \\ w_{i,\zeta_i,1}^{+/-}(n) \cdots w_{i,\zeta_i,\zeta_{i-1}}^{+/-}(n) \end{bmatrix} \geqslant 0 \tag{16}$$

3.3. Lipschitz bound of fully connected ABBAnets

Let us show that, for non-negative ABBAnets, we can provide a cost-efficient way to approximate the Lipschitz constant of

the network, which is more accurate than the normal separable bound given by (3). By using the algebraic properties of ABBA matrices, the following property can be proved:

Proposition 3.3 *Let $m \in \mathbb{N} \setminus \{0\}$ and let $\widetilde{T}$ be an m-layer non-negative ABBAnet. Assume that, for every $i \in \{1, \dots, m\}$, $\widetilde{R}_i$ is a nonexpansive operator operating componentwise. A Lipschitz constant of $\widetilde{T}$ is*

$$\vartheta_m = \|\widetilde{W}_{m+1}\|_{\mathrm{S}} \|\mathfrak{S}(\widetilde{W}_m) \cdots \mathfrak{S}(\widetilde{W}_1)\|_{\mathrm{S}} \|\widetilde{W}_0\|_{\mathrm{S}}. \tag{17}$$

3.4. Lipschitz bound of convolutional ABBA kernels

We can also derive a tight bound for the Lipschitz constant of a convolutional neural network.

Proposition 3.4 *For every $i \in \{1, \dots, m\}$ and $j \in \mathbb{S}(s_i)$, let*

$$\Omega_i^{(j)} = \sum_{n \in \mathbb{Z}^d} \mathfrak{S}(\widetilde{W}_i(s_i n + j)) \tag{18}$$

with $\mathbb{S}(s_i) = \{0, \dots, s_i - 1\}^d$. Then, with nonexpansive and separable activation functions, a Lipschitz constant of the non-negative ABBA convolutional network is

$$\overline{\theta}_m = \|\widetilde{W}_{m+1}\|_{\mathrm{S}} \Big(\prod_{i=1}^{m} \Big\| \sum_{j \in \mathbb{S}(s_i)} \Omega_i^{(j)} \big(\Omega_i^{(j)}\big)^{\top} \Big\|_{\mathrm{S}} \Big)^{\frac{1}{2}} \|\widetilde{W}_0\|_{\mathrm{S}}. \tag{19}$$

In the next section, we will see how to obtain an optimal trade-off between robustness, explainability, and accuracy, by training non-negative ABBAnets subject to different Lipschitz bounds.

4. EXPERIMENTS AND RESULTS

We validate the aforementioned concepts and demonstrate that non-negative ABBAnets exhibit comparable expressivity to their arbitrary-signed counterparts in classification tasks. We perform experiments on three datasets from MedMNISTv2 [16], namely {Blood (8 classes), Derma (7 classes), Pneumonia (2 classes)}-MNIST, a recently introduced benchmark dataset in medical imaging. For Blood and Derma datasets, we used RGB images, while for Pneumonia we have used the original gray scale versions. For all three datasets, we considered the 64×64 input resolution version. A different architecture configuration has been used for each dataset, so that the performance fits the official benchmark [16]. For specifying a feed-forward architecture, we have used the notation $[m_{\mathrm{C}}]\mathrm{C}[m_{\mathrm{F}}]\mathrm{F}$, where m_{C} and m_{F} represent the number of convolutional and fully-connected layers (not counting the final classification layer), respectively, with $m_{\mathrm{C}} + m_{\mathrm{F}} = m$. For the intermediate pooling operators, we used 2D *AvgPooling*. The Lipschitz constant of a 2D pooling layer operating on $k \times k$ windows with stride $k \geqslant 2$ is $\theta_{\mathrm{pool}} = \frac{1}{k}$ [13]. Thus, each pooling operator lowers the Lipschitz bound computed solely from convolution and fully-connected layers.

In Table 1, we present the results obtained on the official test datasets, using standard neural networks and ABBA equivalents, either unconstrained or with an imposed $\overline{\theta} \leqslant \theta_{\max}$, for some chosen upper bound $\theta_{\max}$. All networks were trained for 100 epochs, using a projected Adam optimizer with a learning rate of 10^{-3}. The Lipschitz constants $\overline{\theta}_{\mathrm{dense}}$ and $\overline{\theta}_{\mathrm{conv}}$ of the dense and convolutional part are computed as in (17) and (19), respectively. The estimated global bound for nonnegative networks is $\overline{\theta} = \underline{\theta}_{\mathrm{dense}} \underline{\theta}_{\mathrm{conv}} \theta_{\mathrm{pool}}$ and, for arbitrary-signed networks, it is $\overline{\theta} = \overline{\theta}_{\mathrm{dense}} \overline{\theta}_{\mathrm{conv}} \theta_{\mathrm{pool}}$. For the $\widetilde{R}_i$ operators, we used the capped symmetric Leaky ReLU:

$$\widetilde{R}_i \colon \xi \mapsto \begin{cases} \xi & \text{if } |\xi| \leqslant \beta \\ \alpha(\xi - \beta \mathrm{sgn}(\xi)) + \beta \mathrm{sgn}(\xi) & \text{otherwise,} \end{cases} \tag{20}$$

for which α, β are learnable parameters, with $\alpha \in [0, \alpha_{\max}]$ with $\alpha_{\max} < 1$, and $\beta > 0$.

It is clear that nonnegative ABBAnets outperform standard non-negative ones, even at lower global bounds $\overline{\theta}$. Although unconstrained ABBAnets may result in a significantly higher $\overline{\theta}$ than their standard arbitrary and non-negative counterparts, our experiments showcase that training constrained ABBAnets with lower $\theta_{\max}$ than standard arbitrary networks results in similar performance. The higher $\overline{\theta}$ obtained for unconstrained ABBAnets may reveal some intrinsic characteristic of this type of neural architecture, trained with standard optimizers, which we plan to investigate in future works.

As the increased resilience to adversarial perturbation is directly linked to the Lipschitz bound, we further tested its impact by employing the DDN [17] attack with different maximum allowed perturbation magnitudes $\|z\|$. We compared the performance of our lowest-bound ABBAnets with DeelLip [18] architectures trained under the same bound. Table 2 indicates that, although we may obtain slightly lower clean accuracies (when $z = 0$) compared to DeelLip, ABBAnets ensure a higher protection to adversarial perturbations.

Recent works have illustrated the link between Lipschitz-constrained networks and explainability [19], by analyzing the saliency maps of classification models. To visualize the behaviour of our constrained ABBAnets, we employed Score-CAM [20], which has been shown to outperform other CAM-based methods. Figure 2 clearly illustrates that ABBA constrained networks offer a more explainable behaviour, focusing only on informative regions. We further performed an ablation study using the generated saliency maps, by retaining the pixels corresponding to the values in saliency maps falling under different percentiles. Figure 1 demonstrates that ABBA constrained networks have the lowest accuracies when removing even small parts of informative regions, indicating that saliency maps are much more concentrated over meaningful areas. On the other hand, standard non-negative networks perform roughly equally, irrespective of how many pixels were

Dataset	Architecture		Acc [%]	AUC	Bounds				
					$\underline{\theta}_{conv}$	$\overline{\theta}_{conv}$	$\underline{\theta}_{dense}$	$\overline{\theta}_{dense}$	$\overline{\theta}$
BloodMNIST	2C1F	Standard Arbitrary	93.80	0.995	-	36.83	-	23.35	215.08
		Standard Non-Negative	84.27	0.977	61.35	83.08	16.84	17.66	258.40
		ABBA	93.77	0.996	182.78	185.19	70.46	71.83	$5.80 \cdot 10^3$
		ABBA $\overline{\theta} \leqslant 100$	93.42	0.995	6.90	7.66	47.08	55.22	90.07
		ABBA $\overline{\theta} \leqslant 50$	92.83	0.994	6.88	7.86	25.28	38.31	49.73
		ABBA $\overline{\theta} \leqslant 10$	90.06	0.989	7.38	8.02	4.99	33.06	10.01
PneumoniaMNIST	3C3F	Standard Arbitrary	87.82	0.928	-	9.04	-	32.09	72.56
		Standard Non-Negative	83.81	0.907	66.71	75.72	101.04	195.49	$1.68 \cdot 10^3$
		ABBA	88.46	0.935	$9.52 \cdot 10^6$	$9.56 \cdot 10^6$	510.29	582.51	$1.22 \cdot 10^8$
		ABBA $\overline{\theta} \leqslant 25$	88.78	0.933	7.37	7.57	13.17	17.71	24.92
		ABBA $\overline{\theta} \leqslant 10$	87.82	0.936	5.84	5.84	6.23	21.71	9.12
		ABBA $\overline{\theta} \leqslant 1$	87.58	0.925	3.84	3.85	1.00	12.40	0.97
DermaMNIST	2C2F	Standard Arbitrary	74.31	0.902	-	17.89	-	107.70	481.84
		Standard Non-Negative	67.58	0.804	90.21	658.77	29.48	32.52	665.07
		ABBA	73.11	0.908	269.72	293.37	$2.03 \cdot 10^3$	$2.06 \cdot 10^3$	$1.49 \cdot 10^5$
		ABBA $\overline{\theta} \leqslant 100$	74.16	0.912	7.70	7.75	50.00	99.81	96.94
		ABBA $\overline{\theta} \leqslant 50$	72.46	0.893	7.46	7.52	24.99	67.54	47.00
		ABBA $\overline{\theta} \leqslant 10$	71.77	0.894	3.40	3.50	10.01	306.69	8.76

Table 1: Results for ABBA (constrained and unconstrained) vs. standard (arbitrary-signed and non-negative) networks.

		Accuracy [%]			
		$\|z\| = 0.0$	$\|z\| = 0.8$	$\|z\| = 2.0$	$\|z\| = 3.2$
Pneumo.	Standard	**87.82**	66.03	13.14	0.00
	ABBA $\overline{\theta} = 1$	87.58	**75.48**	**46.63**	**15.38**
	Deel $\overline{\theta} = 1$	87.18	73.88	46.31	11.54
Derma	Standard	**74.31**	12.71	0.00	0.00
	ABBA $\overline{\theta} = 10$	71.77	**57.61**	**31.67**	**9.03**
	Deel $\overline{\theta} = 10$	73.16	47.48	13.66	0.35
		$\|z\| = 0.0$	$\|z\| = 0.4$	$\|z\| = 1.2$	$\|z\| = 2.0$
Blood	Standard	**93.80**	50.83	6.40	0.32
	ABBA $\overline{\theta} = 10$	90.06	**74.24**	**31.01**	**9.06**
	Deel $\overline{\theta} = 10$	92.40	68.92	21.54	1.66

Table 2: Adversarial robustness against DDN attack.

deleted, indicating a non-explainable behaviour. It is clear that as we reduce the expressivity of ABBAnets by imposing lower bounds, they tend to concentrate more on the most informative areas, presenting a more explainable behaviour.

5. CONCLUSION

In this paper, we investigated the expressivity of non-negative neural networks, revealing that by permitting only the first and last layers to be signed, universal approximation can still be achieved. Our experiments showcase comparable performance to standard architectures through careful design of inner convolutional and fully connected operators. Furthermore, constrained models exhibit minimal performance degradation while offering notable enhancements in robustness and explainability. Looking ahead, we aim to extend these mechanisms to diverse architectures tailored for a broader array of applications.

6. REFERENCES

[1] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2014.

[2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," in *Nature*, 1999, vol. 401, pp. 788–791.

[3] J. Chorowski and J. M. Zurada, "Learning understandable neural networks with nonnegative weight constraints," in *IEEE Trans. Neural Net. and Learn. Syst.*, 2015, vol. 26, pp. 62–69.

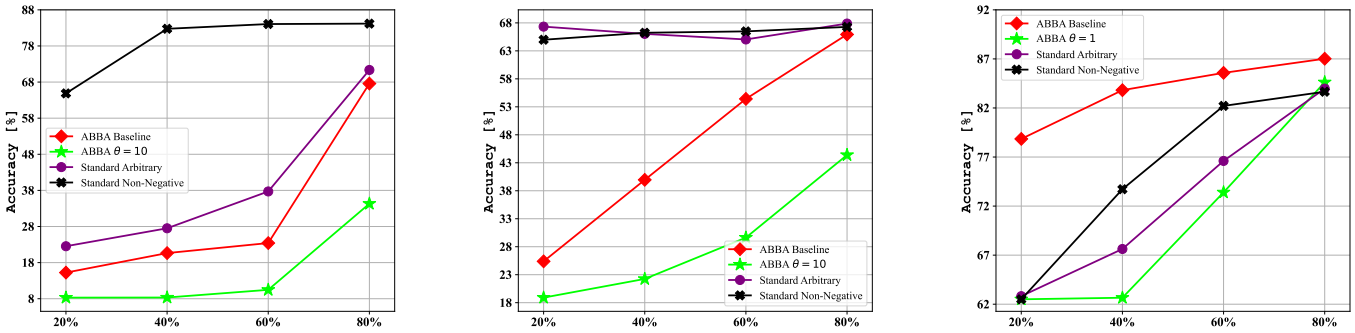
[4] A. Neacşu, J.-C. Pesquet, and C. Burileanu, "Accuracy-robustness trade-off for positively weighted neural networks," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2020, pp. 8389–8393.

[5] A. Neacşu, J.-C. Pesquet, and C. Burileanu, "EMG-based automatic gesture recognition using lipschitz-regularized neural networks," *ACM Trans. on Intell. Syst. and Tech.*, vol. 15, no. 2, pp. 1–25, 2024.

[6] A. Virmaux and K. Scaman, "Lipschitz regularity of deep neural networks: analysis and efficient estimation," in *Proc. Ann. Conf. Neur. Inform. Proc. Syst.*, 2018, vol. 31, pp. 3839–3848.

[7] P. L. Combettes and J.-C. Pesquet, "Lipschitz certificates for layered network structures driven by averaged activation operators," in *J. Math. Data Sci.*, 2020, vol. 2, pp. 529–557.

[8] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *Proc. Int. Conf. Comp. Aided Verif.*, 2017, pp. 97–117.



(a) BloodMNIST (b) DermaMNIST (c) PneumoniaMNIST

Fig. 1: Average test accuracies for retaining certain proportions of input images, by thresholding the saliency maps of Score-CAM at corresponding percentiles. For any given percentile, a lower accuracy corresponds to a more explainable model.

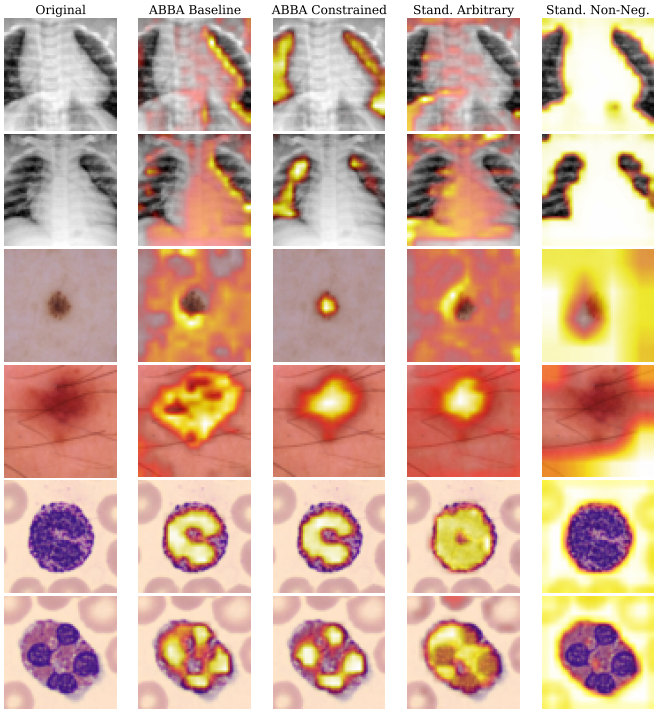


Fig. 2: Saliency maps generated by Score-CAM. Each row corresponds to a sample from the three datasets. A brighter color implies the region has a higher impact on the decision of the classifier.

[9] B. O. Ayinde and J. M. Zurada, "Deep Learning of constrained autoencoders for enhanced understanding of data," in *IEEE Trans. Neural Net. and Learn. Syst.*, 2018, vol. 29, pp. 3969–3979.

[10] B. Amos, L. Xu, and J. Z. Kolter, "Input convex neural networks," in *Proc. Int. Conf. Machine Learn.*, 2017, vol. 70, pp. 146–155.

[11] D. Runje and S. M. Shankaranarayana, "Constrained monotonic neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2023, vol. 202, pp. 29338–29353.

[12] Q. Wang, M. A. Powell, A. Geisa, E. Bridgeford, and J. T. Vogelstein, "Why do networks have inhibitory/negative connections?," *arXiv:2208.03211*, 2022.

[13] A.-A. Neacșu, J.-C. Pesquet, V. Vasilescu, and C. Burileanu, "ABBA neural networks: Coping with positivity, expressivity, and robustness," *SIAM J. Math. Data Sci.*, 2024.

[14] G. Khromov and S. P. Singh, "Some fundamental aspects about Lipschitz continuity of neural network functions," in *arXiv:2302.10886*, 2023.

[15] A. Repetti, M. Terris, Y. Wiaux, and J.-C. Pesquet, "Dual forward-backward unfolded network for flexible plug-and-play," in *Proc. European Sig. Process. Conf.*, 2022, pp. 957–961.

[16] J. Yang, R. Shi, D. Wei, Z. Liu, L. Zhao, B. Ke, H. Pfister, and B. Ni, "Medmnist v2-a large-scale lightweight benchmark for 2D and 3D biomedical image classification," *Nature Sci. Data*, vol. 10, no. 1, pp. 41, 2023.

[17] J. Rony, L. G. Hafemann, L. S. Oliveira, I. B. Ayed, R. Sabourin, and E. Granger, "Decoupling direction and norm for efficient gradient-based ℓ_2 adversarial attacks and defenses," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2019, pp. 4322–4330.

[18] M. Serrurier, F. Mamalet, A. González-Sanz, T. Boissin, J.-M. Loubes, and E. Del Barrio, "Achieving robustness in classification using optimal transport with hinge regularization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 2021, pp. 505–514.

[19] M. Serrurier, F. Mamalet, T. Fel, L. Béthune, and T. Boissin, "On the explainable properties of 1-lipschitz neural networks: An optimal transport perspective," *Proc. Ann. Conf. Neur. Inform. Proc. Syst.*, vol. 36, 2024.

[20] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-CAM: Score-weighted visual explanations for convolutional neural networks," in *Proc. IEEE/CVF Conf. Comp. Vis. Pattern. Recogn.*, 2020, pp. 24–25.