NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY POLITEHNICA BUCHAREST

Doctoral School of Electronics, Telecommunications and Information Technology

Decision No. 119 **from** 26-10-2023

Ph.D. THESIS SUMMARY

Ana-Antonia NEACȘU

METODE ROBUSTE DE ÎNVĂȚARE PROFUNDĂ INSPIRATE DIN ALGORITMI DE PROCESARE DE SEMNAL

ROBUST DEEP LEARNING METHODS INSPIRED BY SIGNAL PROCESSING ALGORITHMS

THESIS COMMITTEE

Prof. Dr. Ing. Mihai CIUC Politehnica Univ. of Bucharest	President
Prof. Dr. Ing. Corneliu BURILEANU Politehnica Univ. of Bucharest	PhD Supervisor
Prof. Dr. Ing. Jean-Christophe Pesquet CentraleSupélec, Université Paris-Saclay	PhD Supervisor
Prof. Dr. Ing. Nicu SEBE University of Trento, Italy	Referee
Prof. Dr. Ing. Corneliu RUSU Technical University of Cluj-Napoca	Referee
Prof. Dr. Ing. Daniela TĂRNICERIU Technical University "Gh. Asachi" Iași	Referee
Dr. Ing. Jean-Philippe OVARLEZ Research Director at ONERA, Université Paris-Saclay, France	Examiner
Dr. Ing. Frank MAMALET Senior Expert in Artificial Intelligence at IRT Saint Exupéry, Toulouse, France	Examiner

BUCHAREST 2023

Table of contents

1 Introduction **1**

1.1 Context . . . 1

1.2 Impact and applicability . . . 2

1.3 Main contributions . . . 3

1.4 Publications . . . 4

1.5 Co-tutelle thesis . . . 5

1.6 Outline . . . 5

2 Overview of adversarial attacks and defenses **7**

2.1 Robustness of neural networks . . . 7

2.2 Definitions and notation . . . 7

2.3 Threat models . . . 7

2.4 Attack mechanisms . . . 7

2.5 Defense strategies . . . 8

2.6 Conclusion . . . 8

3 EMG-based automatic gesture recognition using robust neural networks **9**

3.1 EMG and automatic gesture recognition . . . 9

3.1.1 Challenges and limitations . . . 9

3.2 Robustness solutions in the context of non-negative neural networks . . 10

3.2.1 Problem formulation . . . 10

3.2.2 Lipschitz robustness certificate . . . 10

3.3 Optimization methods for training robust feed-forward neural networks 11

3.3.1 Constraints sets . . . 11

3.4 AGR experimental setup . . . 12

3.4.1 sEMG datasets . . . 12

3.4.2 Proposed Architecture . . . 12

3.4.3 Performance analysis in terms of accuracy and robustness . . . 13

3.5 Robustness validation . . . 13

3.5.1 Sensitivity to adversarial attacks . . . 13

3.5.2 Noisy input behaviour . . . 14

3.5.3 Real-life scenario validation . . . 14

3.5.4 Limitations . . . 14

3.6 Conclusion . . . 14

4 Signal denoising using new classes of robust neural networks 16
4.1 Adaptive convolutional networks 16
4.1.1 Making the bridge between CNNs and FCNs 16
4.1.2 Learning algorithm 17
4.1.3 Experimental Evaluation 18
4.2 Design of robust complex-valued feed-forward neural networks 19
4.2.1 Theoretical background 19
4.2.2 Nonexpansive complex-valued activation functions 20
4.2.3 Robustness results 20
4.2.4 Proposed approach 20
4.2.5 Experimental results 22
4.3 Conclusion 23

5 ABBA neural networks: coping with positivity, expressivity, and robustness 24
5.1 Introduction 24
5.2 Related work 25
5.3 ABBA neural networks 25
5.3.1 Problem formulation 25
5.3.2 ABBA Matrices 25
5.3.3 Extension to feedforward networks 25
5.3.4 Link with standard neural networks 26
5.3.5 Expressivity of non-negative ABBA networks 26
5.3.6 Lipschitz bounds for ABBA fully-connected networks 26
5.4 Convolutional networks 27
5.4.1 ABBA convolutional layers 27
5.4.2 Lipschitz bounds for convolutional networks 27
5.4.3 Bounds for ABBA convolutional networks 27
5.5 Lipschitz-constrained training 28
5.6 Experiments 28
5.7 Conclusions 29

6 Conclusions 30
6.1 Summary 30
6.2 Perspectives 30
6.2.1 Training 1-Lipschitz denoisers 31
6.2.2 Expanding the applications of complex-valued neural networks . 31
6.2.3 Controlling the Lipschitz constant of more complex layer structures 31
6.2.4 Combining Lipschitz control with other certifiable defenses 31
6.2.5 Studying the effect of other regularization techniques 31
6.2.6 Extending to other distances 31

References 32

Chapter 1

Introduction

1.1 Context

Recently, machine learning methods have become ubiquitous tools in a wide range of tasks, because of their ability to solve a great variety of problems, ranging from simple regressions to complex multi-modal classification. These methods stand at the very core of *Artificial Intelligence* (AI). AI represents the marvel of nowadays technology and is used successfully in an ever-increasing number of areas impacting our lives, e.g. medicine [21], autonomous driving [28], natural language processing [45], human-computer interaction (HCI) [36], etc. However, deep neural networks, which are probably the most powerful methods, raise challenges in terms of implementation heaviness during the learning phase. Moreover, they appear as black boxes whose robustness is not always well-controlled [15, 34].

Developing trustworthy AI is essential to ensure that intelligent systems can be relied upon for critical decision-making without compromising ethical standards.

To reach this goal, a critical issue to be addressed when developing real-life applications using neural networks is the correct evaluation and control of their robustness against possible adversarial attacks.

Adversarial inputs represent malicious input data that can fool machine learning models. The concept was highlighted in [41], where the authors showed that slightly altering data inputs that were correctly classified by the network can lead to a wrong classification [24, 3, 43, 19].

It must be emphasized that adversarial inputs are not necessarily artificially created with the intention to sabotage the system. They can also occur innately under different forms and can seriously flaw the performance of real-life applications based on pre-trained models [32]. A better analysis of the stability properties of neural networks can be viewed as the first step towards a better understanding of the mathematical principles governing their functionalities.

The main goal of this thesis is to design new methods for training safe yet high-performance neural networks. Recent mathematical results show that it becomes easier to control the stability of neural networks by introducing suitable constraints on their weights. Nevertheless, this requires the management of constraints that are not necessarily convex in the training phase of the neural network. To this end, we *designed carefully crafted constraints* that we later used in the training process, to ensure the robustness of the neural network. As highlighted in [17], the *Lipschitz behaviour* of the network is tightly correlated with its robustness against adversarial attacks. This constant allows us to upper bound the output perturbation knowing the magnitude of the input one, for a given metric [39]. Controlling this constant leads to a feasible solution to assess the effect of adversarial attacks if accurately computed. However, computing the exact Lipschitz constant even for a shallow neural network is a non-polynomial *NP-hard* problem. So the main difficulty is to *find ways of approximating it as tightly as possible*. Lately, several methods have been proposed to train Lipschitzian networks, which fall into two main categories. Regularization approaches include double backpropagation [13] or apply penalization on the network Jacobian [18], which imposes local Lipschitz constraints, but do not enforce the constraint globally on the network. Another approach consists of imposing some constraints on the architecture of the network, so as to constrain the spectral norm of each layer [43] [10]. At the expense of computation complexity, these methods ensure a Lipschitzian network. In [11], novel results leading to accurate approximations to the Lipschitz constant of positive feed-forward neural networks were proposed. These preliminary results served as a starting point for proposing efficient methods for designing safe neural networks.

After establishing all the mathematical backbone, we next focus on *building new neural network architectures* based on the aforementioned philosophy. An important part of the work presented in this thesis consists in *developing efficient optimization methods for supervised learning of neural networks*. We look at the possible choices for the structure of the network, given the different classes of existing iterative optimization algorithms. To handle stability constraints, particular attention is paid to *proximal methods* which offer powerful tools for optimization in a large-scale context. We study how ensuring *robustness affects the overall performance* of the learning systems, and try to reach a good *robustness-accuracy trade-off*.

A very important aspect in all exploratory research is the *validation of the theoretical results in a real application context*. Some of the models trained with stability guarantees are tested in real-life contexts to show the versatility of the designed solutions. We then measure the influence on the system performance and *compare* the obtained results with those generated with classical architectures, as well as other defense strategies.

1.2 Impact and applicability

This thesis contributes to the field of machine learning by trying to give an answer to the fundamental question:

How safe neural networks are?

The objective is to provide mathematically proven robustness guarantees, develop the associated software, and make it publicly available. Another important aspect of this thesis is the focus on applications based on audio and physiological signals which have direct use in the development of innovative technologies and can directly benefit a variety of consumer products.

More generally, by approaching the concept of Safe Neural Networks, this thesis contributes to the state of knowledge in artificial intelligence, leveraging on the latest research results in the field of optimization. Developing new methods that can be used to make learning systems more robust and explainable will open new perspectives in terms of safe and controlled technological progress.

1.3 Main contributions

The first contributions of the thesis appear in Chapter 3:

(i) We propose a robust real-time Automatic Gesture Recognition system based on sEMG signals. The robustness is ensured by using a novel learning algorithm for training feedforward neural networks.

(ii) We show that a good accuracy-robustness balance can be reached. To do so, we train the system under carefully crafted spectral norm constraints, allowing us to finely control its Lipschitz constant. A tight Lipschitz constant is efficiently estimated by focusing on neural networks with nonnegative weights, as in [8].

(iii) We demonstrate the performance of the final architecture in real-life experiments, where we show that the proposed robust model outperforms those trained conventionally.

(iv) We analyze how our system behaves when the input is affected by different noise levels, simulating perturbations that may occur in real scenarios.

(v) We show the validity of our solution by experimenting on several publicly available sEMG gesture datasets.

Chapter 4 includes the following main contributions.

(i) Inspired by MIMO filters, we introduce a new class of neural networks, which can be seen as an intermediate solution between CNNs and FCNs.

(ii) We propose a constrained training strategy, which allows us to control the Lipschitz constant of the network in order to secure its robustness to adversarial noise.

(iii) We present a new architecture (RCFF-Net), which operates in the complex-valued domain, for which we derive tight Lipschitz constant bounds.

(iv) We develop a constrained learning strategy to train the proposed structure while controlling its global Lipschitz constant.

(v) Both architectures ACNN and RCFF are evaluated in audio signal denoising tasks, proving that our solution is not limited to classification problems.

The contributions from Chapter 5 are mentioned below.

(i) We introduce ABBA networks, a novel class of (almost) non-negative neural networks, which are shown to possess a series of appealing properties.

(ii) We show that we can put any arbitrary signed network in an ABBA form. We show that this property holds for fully connected as well as for convolutional neural networks.

(iii) Universal approximation theorems are derived for networks featuring non-negatively weighted layers.

(iv) We present a method for effectively controlling the Lipschitz constant of ABBA networks. This control strategy applies to both fully connected and convolutional cases.

(v) Numerical experiments conducted on standard image datasets showcase the excellent performance of ABBA networks for small models. Notably, they exhibit substantial improvements in both performance and robustness compared to networks with exclusively non-negative weights. Moreover, we demonstrate that ABBA networks are competitive with robust networks featuring arbitrarily signed weights, trained using state-of-the-art techniques.

1.4 Publications

Submitted journal articles

- A. Neacșu, J.-C. Pesquet, V. Vasilescu and C.Burileanu, "*ABBA Neural Networks: Coping with Positivity, Expressivity, and Robustness*", submitted to SIAM Journal on Mathematics of Data Science (SIMODS), 2023.

Accepted or published journal articles

- A. Neacșu, J.-C. Pesquet and C.Burileanu, "*EMG-Based Automatic Gesture Recognition Using Lipschitz-Regularized Neural Networks*", accepted for publication in ACM Transactions on Intelligent Systems and Technology (TIST), 2023.
- N Lassau, S. Ammari, E. Chouzenoux, A. Neacșu et al. *“Integrating deep learning CT-scan model, biological and clinical variables to predict severity of COVID-19 patients”*,in Nature Communication 12, 634 (2021), https://doi.org/10.1038/s41467-020-20657-4

Conference Proceedings

- C. Andronache, M. Negru, I. Bădiţoiu, G. Cioroiu, A. Neacsu and C. Burileanu, "*Automatic Gesture Recognition Framework Based on Forearm EMG Activity*", in Proc. 45th International Conference on Telecommunications and Signal Processing (TSP), Prague, Czech Republic, 2022, pp. 284-288, doi: 10.1109/TSP55681.2022.9851314.

- A. Neacşu, R. Ciubotaru, J. -C. Pesquet and C. Burileanu, "*Design of Robust Complex-Valued Feed- Forward Neural Networks*", in Proc. 30th European Signal Processing Conference (EUSIPCO), Belgrade, Serbia, 2022, pp. 1596-1600, doi: 10.23919/EUSIPCO55093.2022.9909696.
- A. Neacșu, K. Gupta, J. -C. Pesquet and C. Burileanu, "*Signal Denoising Using a New Class of Robust Neural Networks*" in Proc. of 28th European Signal Processing Conference (EUSIPCO), Amsterdam, Netherlands, 2021, pp. 1492-1496, doi: 10.23919/Eusipco47968.2020.9287630.
- V. Vasilescu, A. Neacşu, E. Chouzenoux, J. -C. Pesquet and C. Burileanu, "*A Deep Learning Approach For Improved Segmentation Of Lesions Related To Covid-19 Chest CT Scans*", in Proc. IEEE 18th Int. Sym. on Biomedical Imaging (ISBI), Nice, France, 2021, pp. 635-639, doi: 10.1109/ISBI48211.2021.9434139.
- A. Neacșu, J.-C. Pesquet, and C. Burileanu, "*Accuracy-robustness trade-off for positively weighted neural networks*", in Proc. IEEE International Conference on Acoustics and Speech Signal Process. (pp. 8389–8393). Barcelona, Spain, 2020, doi: 10.1109/ICASSP40776.2020.9053803.
- C. Andronache, M. Negru, A. Neacsu, G. Cioroiu, A. Radoi and C. Burileanu, "*Towards extending real-time EMG-based gesture recognition system*", in Proc. 43rd International Conference on Telecommunications and Signal Processing (TSP), Milan, Italy, 2020, pp. 301-304,
doi: 10.1109/TSP49548.2020.9163481.

1.5 Co-tutelle thesis

Collaboration lies at the heart of scientific progress and innovation. In today's interconnected world, the significance of collaborative efforts cannot be overstated, particularly in the field of academic research. This thesis is the result of a co-tutelle collaboration, between University Politehnica of Bucharest and CentraleSupélec, Graduate School of Engineering Sciences of University Paris Saclay. This thesis has provided a remarkable opportunity to foster cooperation and exchange knowledge between these two distinguished institutions.

1.6 Outline

The rest of the thesis is organized as follows. In Chapter 2, we present an overview of existing attacks and defenses. In Section 2.1 we establish the concept of robustness in the context of neural networks, while in Section 2.2 we introduce the mathematical notation used throughout the chapter. We present the most used scenarios of threat models (Section 2.3) and then we describe both white-box and black-box attack mechanisms in Section 2.4. We end the chapter by emphasising different defense strategies in Section 2.5.

In Chapter 3 we present a robust mechanism for training non-negative neural networks in the context of automatic gesture recognition based on sEMG signals. In Section 3.1 we lay the foundational understanding of electromyography and emphasize its relevance in the context of gesture recognition. Following this, in Section 3.2 we introduce innovative approaches to enhance the robustness of fully connected neural networks. Section 3.3 then details the optimization techniques crucial to our proposed methods, variants of which will be used in the rest of this work. Transitioning to practical implementation, Section 3.4 provides insights into the experimental framework considered for our task. The chapter culminates with Section 3.5 where we extensively validate the robustness of our proposed models. Finally, we conclude this chapter by summarizing our key findings and their implications in Section 3.6.

In Chapter 4 we embark on a journey to enhance signal denoising through innovative robust neural network architectures. Starting with Section 4.1 we introduce the first novel architecture we propose in this thesis. We then explore, in Section 4.1.1, a critical step in bridging the gap between these powerful neural network paradigms: the use of fully connected and convolutional layers. Section 4.1.2 delves into the optimization strategies employed for training our proposed models, shedding light on the core of our methodology. Our practical applications developed in Section 4.1.3 provide an in-depth examination of our model performance in signal denoising scenarios. The second part of the chapter, starting with Section 4.2 introduce a new class of networks (RCFF) operating in the complex domain. Theoretical foundations and insights are presented in Sections 4.2.1-4.2.3 where we elucidate the mathematical underpinnings of our robust training mechanisms, and then we detail its implementation in Section 4.2.4. Then, we showcase the empirical outcomes of applying our RCFF-Net to audio denoising problems in Section 4.2.5. Ultimately, we conclude this chapter by summarizing our key findings and their implications in Section 4.3.

In Chapter 5, we introduce a groundbreaking class of neural networks known as ABBA Neural Networks, engineered to grapple with issues of positivity, expressivity, and robustness. We start with Section 5.1 offering an overview of the challenges that our novel ABBA networks aim to address. We provide context in Section 5.2, examining the existing landscape of neural network solutions and underscoring the unique contributions of ABBA networks. The core of our chapter unfolds with Section 5.3 where we describe the architectural foundations and key attributes of this innovative neural network class. Subsequently, in Section 5.4, we extend the applicability of ABBA networks to the convolutional case, highlighting the adaptability of this approach across diverse network architectures. An in-depth look into the training methods and techniques ensuring Lipschitz stability is presented in Section 5.5. Section 5.6 serves as the empirical heart of this chapter, where we conduct comprehensive evaluations to validate the performance and effectiveness of ABBA networks across various classification scenarios. In Section 5.7, we sum up our key findings, insights, and implications of our research.

Finally, in Chapter 6, we draw the final remarks of this thesis, followed by a brief description of some envisioned perspectives.

Chapter 2

Overview of adversarial attacks and defenses

This chapter presents an overview of the current advancements in the domain of the robustness of neural networks against adversarial perturbations. We will define the concept of adversarial attacks and explain the insights of the most efficient attack strategies. Studying deliberately crafted attacks in machine learning is crucial because it allows to identify the vulnerabilities of models and enhances their robustness.

2.1 Robustness of neural networks

The section emphasizes the need for understanding and enhancing neural network resilience to adversarial inputs, delving into the concept of robustness, perturbation creation, and strategies to mitigate their impact.

2.2 Definitions and notation

In this section, the main notations used throughout the chapter are introduced.

2.3 Threat models

This section discusses the possible options for threat models, depending on their objective and level of access to the original model can fall in several categories, as follows. Based on the adversary's objective the attacks can be *targeted* or *untargeted*. Additionally, based on the level of access the attacker has on the victim model, three distinct categories of attacks arise: *black-box* attacks, *white-box* attacks and *gray-box* attacks.

2.4 Attack mechanisms

In this section, we detail the main algorithms for generating adversarial samples in all three contexts. We consider mainly evasion methods since they are more common.

2.5 Defense strategies

Since there are many ways an adversary can exploit the model's weaknesses, defensive strategies have been developed to alleviate this robustness issue. This section presents the main directions in this domain.

2.6 Conclusion

This chapter has presented an overview of the state-of-the-art in the field of adversarial attacks and defenses of neural networks. The robustness of deep learning models is a hot topic that has attracted increasing attention from the research community, since it represents an important aspect to consider in the development and integration of future trust-worthy AI solutions in real-life applications. The next chapters will present new contributions in this domain.

Chapter 3

EMG-based automatic gesture recognition using robust neural networks

This chapter introduces a novel approach for building a robust Automatic Gesture Recognition system based on Surface Electromyographic (sEMG) signals, acquired at the forearm level. Our main contribution is to propose new constrained learning strategies that ensure robustness against adversarial perturbations by controlling the Lipschitz constant of the classifier. We focus on nonnegative neural networks for which accurate Lipschitz bounds can be derived, and we propose different spectral norm constraints offering robustness guarantees from a theoretical viewpoint. Experimental results on four publicly available datasets highlight that a good trade-off in terms of accuracy and performance is achieved. We then demonstrate the robustness of our models, compared to standard trained classifiers in three scenarios, considering both white-box and black-box attacks.

3.1 EMG and automatic gesture recognition

sEMG stands for surface electromyography and represents the electrical manifestation of the neuromuscular activation related to the contraction of the muscles [1]. This technology may be used by physically impaired persons to control rehabilitation and assisting devices. EMG is also used in many types of research domains, including those involved in biomechanics, motor control, neuromuscular physiology, movement disorders, postural control, and physical therapy [35].

3.1.1 Challenges and limitations

Gestures constitute a universal and intuitive way of communication, with the potential of bringing the Internet of Things (IoT) experience to a different, more organic level [36]. Automatic gesture recognition (AGR) algorithms can be successfully used in various applications, from sign language recognition (SLR) [7] to Virtual Reality (VR) games [44].

Two critical issues need to be addressed when developing AGR algorithms: fast enough inference to ensure real-time feeling for the end-user, and accurate and robust classification to guarantee that the gesture is correctly identified no matter the environmental conditions. Machine learning methods have become the main tools for AGR systems, on account of their ability to solve a great variety of problems, from simple regressions to complex multi-modal classification.

The Lipschitz behaviour of the network is intimately connected with its resilience against adversarial attacks.

3.2 Robustness solutions in the context of non-negative neural networks

3.2.1 Problem formulation

Model 3.2.1 Any feedforward neural network is obtained by cascading m layers associated with operators $(T_i)_{1 \leqslant i \leqslant m}$. The neural network can thus be expressed as the following composition of operators:

$$T = T_m \circ \cdots \circ T_1. \tag{3.1}$$

Each layer $i \in \{1, \dots, m\}$ has a real-valued vector input x_i of dimension N_{i-1} which is mapped to

$$T_i(x_i) = R_i(W_i x_i + b_i), \tag{3.2}$$

where $W_i \in \mathbb{R}^{N_i \times N_{i-1}}$, $b_i \in \mathbb{R}^{N_i}$ are the weight matrix and bias parameter, respectively. $R_i \colon \mathbb{R}^{N_i} \to \mathbb{R}^{N_i}$ constitutes a non-linear activation operator which is applied component-wise (e.g., ReLU or Sigmoid).

3.2.2 Lipschitz robustness certificate

Consider a neural network T as described above. let $x \in \mathbb{R}^{N_0}$ be the input of the network and let $T(x) \in \mathbb{R}^{N_m}$ be its associated output. By adding some small perturbation $z \in \mathbb{R}^0$ to the input, the perturbed input is

$$\tilde{x} = x + z.$$

The effect of the perturbation on the output of the system can be quantified by the following inequality:

$$\|T(\tilde{x}) - T(x)\| \leqslant \theta_m \|z\|, \tag{3.3}$$

where $\theta_m \geqslant 0$ denotes a Lipschitz constant of the network. θ_m represents thus an important parameter that allows us to assess and control the sensitivity of a neural network to various perturbations. It needs however to be accurately estimated to provide valuable information. A standard approximation to the Lipschitz constant [17] is given by

$$\theta_m = \prod_{i=1}^{m} \|W_i\|_{\mathrm{S}}, \tag{3.4}$$

where $\|\cdot\|_{\mathrm{S}}$ denotes the *spectral norm* of a matrix. Although simple to compute, this approximate bound is over-pessimistic. Different methods for obtaining tighter estimates of the Lipschitz constant have been presented in the recent literature; see for example [39, 11, 14, 25, 5]. Local estimates of the Lipschitz constant can also be performed, which may appear more relevant. But they are more complex to compute and, as we will see, controlling the global Lipschitz constant is usually sufficient to get a good performance. Estimating the global Lipschitz constant of the network is an NP (non-deterministic polynomial-time)-hard problem [39].

3.3 Optimization methods for training robust feed-forward neural networks

To ensure robustness, we shall impose spectral norm constraints on the weight matrices. In other words, the vector of parameters η is constrained to belong to a closed set $\mathscr{S}$ that will be described in the next section. We propose to use an extension of a standard optimization techniques for training neural networks [12]. More specifically, we will implement a *projected stochastic gradient* algorithm. A momentum parameter is introduced in this algorithm to accelerate the convergence process.

Algorithm 1: Projected SGD Algorithm

```
Partition $\{1,\dots,K\}$ into mini-batches $(\mathbb{L}_{q,n})_{1\leqslant q\leqslant Q}$
foreach $q \in \{1,\dots,Q\}$ do
    foreach $i \in \{1,\dots,m\}$ do
        $\Delta_{i,n} = (1+\zeta_n)\eta_{i,n} - \zeta_n\eta_{i,n-1}$ $\widetilde{\eta}_{i,n} = [(\eta_{j,n+1}^\top)_{j<i}\ \Delta_{i,n}^\top\ (\eta_{j,n}^\top)_{j>i}]^\top$
        $\eta_{i,n+1} = \mathsf{P}_{\mathscr{S}_{i,n}}\Big(\Delta_{i,n} - \gamma_n \sum_{k\in\mathbb{L}_{q,n}} \nabla_i \ell(z_k, \widetilde{\eta}_{i,n})\Big)$
where $\mathscr{S}_{i,n} = \big\{\eta_i \mid [(\eta_{j,n+1}^\top)_{j<i}\ \eta_i^\top\ (\eta_{j,n}^\top)_{j>i}]^\top \in \mathscr{S}\big\}$.
```

3.3.1 Constraints sets

As mentioned before, this thesis revolves around feed-forward networks with positive weights. Thus, the first condition that we impose is nonnegativity for each layer $i \in \{1,\dots,m\}$, which is modelled by the constraint set

$$\mathscr{D}_i = \{W_i \in \mathbb{R}^{N_i \times N_{i-1}} \mid W_i \geqslant 0\} \tag{3.5}$$

Moreover, we must impose a spectral norm constraint on the weight matrices to control the robustness of the system. This translates mathematically as the following upper bound constraint:

$$\|W_m \cdots W_1\|_{\mathrm{S}} \leqslant \overline{\vartheta}, \tag{3.6}$$

where $\overline{\vartheta}$ represents the target maximum Lipschitz constant of the network. This bound constitutes a direct measure of the system's level of robustness against adversarial inputs. We need to handle these two constraints simultaneously during the training process. For the second one we introduce the following constraints set.

$$\mathscr{C}_{i,n} = \{W_i \in \mathbb{R}^{N_i \times N_{i-1}} \mid \|A_{i,n} W_i B_{i,n}\|_{\mathrm{S}} \leqslant \overline{\vartheta}\} \tag{3.7}$$

Thus, our objective will be to perform the projection onto the set $\mathscr{S}_{i,n} = \mathscr{D}_i \cap \mathscr{C}_{i,n}$, for each layer $i \in \{1, \ldots, m\}$ and at each iteration n. Several algorithms can be envisaged to solve this convex optimization problem.

3.4 AGR experimental setup

3.4.1 sEMG datasets

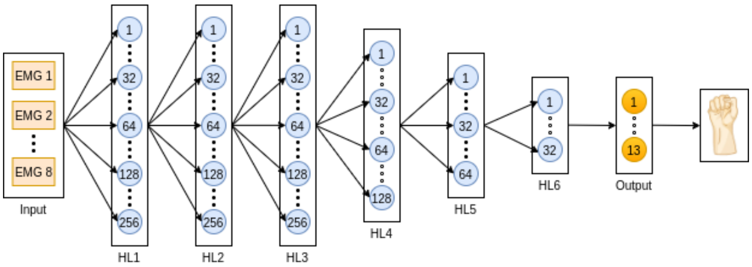


Fig. 3.1 *Proposed neural network architecture for AGR.*

We test our proposed training scheme on four online datasets containing EMG information on different hand gestures. The first three were acquired using Myo armband, a device developed by Thalmic Labs, equipped with eight sEMG sensors displayed circularly, while the last one was acquired using 10 active double-differential `OttoBockMy-oBock13E200` sEMG electrodes.

We also validate our models in a real-context scenario. For the real-life predictions, we recorded the EMG activity associated with each gesture at the forearm level using the Myo armband.

3.4.2 Proposed Architecture

The proposed architecture is described in Figure 3.1. The raw 8 /10 channels EMG signal is split using a 250 ms sliding window, with 50% overlap. A 250 ms window is long enough to cover the most common gesture durations, ensuring that the essential temporal aspects of each gesture are captured within this window. Overlap ensures that important signal characteristics, such as abrupt changes or transient patterns, are not missed due to window boundaries. From each window of each channel, a series of 8 time descriptors are extracted. The information from all the channels is then concatenated, forming a 64 (80 for the fourth dataset)-dimensional vector.

3.4.3 Performance analysis in terms of accuracy and robustness

Table 3.1 Lipschitz constant obtained with various constrained optimization strategies for different accuracies

		Accuracy	75%	80%	85%	90%	95%
Lipschitz constant 7-gestures Myo-sEMG	$\widetilde{\mathscr{C}}_i \cap \mathscr{D}_i$	$\widetilde{\mathrm{P}}_{\widetilde{\mathscr{C}}_i\cap\mathscr{D}_i}$	19.5	37.5	68.3	3.5×10^4	3.5×10^8
		$\mathrm{P}_{\widetilde{\mathscr{C}}_i\cap\mathscr{D}_i}$	0.66	13.47	74.16	1.04×10^3	1.39×10^5
	$\check{\mathscr{C}}_{i,n} \cap \mathscr{D}_i$	$\widetilde{\mathrm{P}}_{\check{\mathscr{C}}_{i,n}\cap\mathscr{D}_i}$	0.71	1.84	3.42	6.87	11.60
		$\mathrm{P}_{\check{\mathscr{C}}_i\cap\mathscr{D}_i}$	0.70	1.35	3.41	6.79	11.20
	$\mathscr{C}_{i,n} \cap \mathscr{D}_i$	$\widetilde{\mathrm{P}}_{\mathscr{C}_{i,n}\cap\mathscr{D}_i}$	0.44	1.79	2.93	4.85	5.68
		$\mathrm{P}_{\mathscr{C}_{i,n}\cap\mathscr{D}_i}$	0.35	0.46	0.65	0.82	0.95
Lipschitz constant 13-gestures 13Myo-sEMG	$\widetilde{\mathscr{C}}_i \cap \mathscr{D}_i$	$\widetilde{\mathrm{P}}_{\widetilde{\mathscr{C}}_i\cap\mathscr{D}_i}$	20.2	41.8	145.2	2.2×10^5	1.21×10^{11}
		$\mathrm{P}_{\widetilde{\mathscr{C}}_i\cap\mathscr{D}_i}$	0.85	20.47	112.3	1.62×10^4	2.31×10^8
	$\check{\mathscr{C}}_{i,n} \cap \mathscr{D}_i$	$\widetilde{\mathrm{P}}_{\check{\mathscr{C}}_{i,n}\cap\mathscr{D}_i}$	0.84	2.08	4.23	7.54	12.02
		$\mathrm{P}_{\check{\mathscr{C}}_i\cap\mathscr{D}_i}$	0.81	2.01	4.12	7.50	11.92
	$\mathscr{C}_{i,n} \cap \mathscr{D}_i$	$\widetilde{\mathrm{P}}_{\mathscr{C}_{i,n}\cap\mathscr{D}_i}$	0.54	1.87	3.38	4.20	5.78
		$\mathrm{P}_{\mathscr{C}_{i,n}\cap\mathscr{D}_i}$	0.49	0.53	0.75	0.92	1.25
		Accuracy	65%	70%	75%	80%	85%
Lipschitz constant 24-gestures NinaPro DB5 Ex C.	$\widetilde{\mathscr{C}}_i \cap \mathscr{D}_i$	$\widetilde{\mathrm{P}}_{\widetilde{\mathscr{C}}_i\cap\mathscr{D}_i}$	25.13	57.16	188.26	2.5×10^6	2.14×10^{11}
		$\mathrm{P}_{\widetilde{\mathscr{C}}_i\cap\mathscr{D}_i}$	1.85	31.12	112.3	1.82×10^4	4.63×10^8
	$\check{\mathscr{C}}_{i,n} \cap \mathscr{D}_i$	$\widetilde{\mathrm{P}}_{\check{\mathscr{C}}_{i,n}\cap\mathscr{D}_i}$	1.74	2.41	6.02	10.17	20.14
		$\mathrm{P}_{\check{\mathscr{C}}_i\cap\mathscr{D}_i}$	1.57	2.18	5.94	10.58	19.69
	$\mathscr{C}_{i,n} \cap \mathscr{D}_i$	$\widetilde{\mathrm{P}}_{\mathscr{C}_{i,n}\cap\mathscr{D}_i}$	0.88	2.05	4.28	5.74	6.84
		$\mathrm{P}_{\mathscr{C}_{i,n}\cap\mathscr{D}_i}$	0.77	0.96	1.27	1.44	1.96
Lipschitz constant 53-gestures NinaPro DB 1	$\widetilde{\mathscr{C}}_i \cap \mathscr{D}_i$	$\widetilde{\mathrm{P}}_{\widetilde{\mathscr{C}}_i\cap\mathscr{D}_i}$	26.26	86.17	200.45	4.10×10^6	4.45×10^{11}
		$\mathrm{P}_{\widetilde{\mathscr{C}}_i\cap\mathscr{D}_i}$	2.60	50.12	163.14	2.8×10^4	2.9×10^9
	$\check{\mathscr{C}}_{i,n} \cap \mathscr{D}_i$	$\widetilde{\mathrm{P}}_{\check{\mathscr{C}}_{i,n}\cap\mathscr{D}_i}$	2.94	4.43	6.88	14.25	22.16
		$\mathrm{P}_{\check{\mathscr{C}}_i\cap\mathscr{D}_i}$	2.83	2.18	5.56	16.48	20.16
	$\mathscr{C}_{i,n} \cap \mathscr{D}_i$	$\widetilde{\mathrm{P}}_{\mathscr{C}_{i,n}\cap\mathscr{D}_i}$	1.22	1.80	6.83	7.40	8.23
		$\mathrm{P}_{\mathscr{C}_{i,n}\cap\mathscr{D}_i}$	1.56	2.08	2.53	2.74	3.88

The obtained results are summarized in Table 3.1.

3.5 Robustness validation

In this section, we investigate to what extent the theoretical concepts described in the previous sections help in improving the robustness of the classifier in different settings. To this goal, we consider the following three scenarios. In the first one, we examine the impact of adversarial attacks on the performance of the classifier. The second scenario takes into account the effect of noise in the acquisition process. In the case of sEMG signals, this noise may come from imperfect skin-sensor contact caused by hairs or drops of sweat. In the last scenario, we perform a real-life experiment using 10 able-bodied volunteers.

3.5.1 Sensitivity to adversarial attacks

We evaluate our robust model on purposely designed perturbations, by studying their influence on the overall performance of the system. We lead attacks on our best robust model in terms of accuracy and robustness, achieving 92.95% accuracy and a Lipschitz constant $\overline{\vartheta} = 0.87$ for the 7-gesture dataset. We compare the results with two conventionally trained models: the best one in terms of performance, which achieves 99.78% prediction accuracy on non-adversarial data, and another one trained to have similar performance as our robust model, reaching 92.99% accuracy on the original test set.

To create the adversarial samples we used some of the most popular white-box attackers, namely: Fast gradient sign method (FGSM) [17], Jacobian Saliency Map Attacker (JSMA)[33], Projected gradient descent (PGD)[30], Carlini and Wagner (C&W) [4] and Gradient Matching (GM) [16].

3.5.2 Noisy input behaviour

To simulate the effect of underlying noise generated during the acquisition process, we added synthetic noise directly to the raw sEMG data, prior to the feature extraction step. The noise is chosen independent and identically distributed according to a Gaussian mixture law $(1-p)\mathcal{N}(0,\sigma_0^2)+p\mathcal{N}(0,\sigma_1^2)$. This experiment emphasizes that controlling the Lipschitz constant of a network improves its robustness not only against targeted adversarial attacks, as shown previously, but also in the case of black-box attacks, where no prior information about the model is used.

3.5.3 Real-life scenario validation

To illustrate the practical applicability of our findings, we proceed to validate our model in a real-life context. For this purpose, we designed an experiment to compare a conventionally trained model with the constrained one. We observed that training a positive neural network subject to Lipschitz constraints improves the overall robustness of the classifier against adversarial perturbations, not only from a theoretical viewpoint but also practically by leading to more reliable systems with greater generalization power.

3.5.4 Limitations

Increased training time is one of the main limitations of our proposed approach. Indeed, to compute the true projection, the proposed method uses an iterative algorithm that performs singular value decomposition at each iteration, which is a resource-consuming operation, especially when performed on large matrices. We propose several lower complexity solutions, which have proved to offer a good trade-off between training time, robustness, and performance.

3.6 Conclusion

This chapter has shown the usefulness of designing robust feed-forward neural networks for automatic gesture recognition based on sEMG physiological signals. More precisely, we proposed to finely control the Lipschitz constant of these nonlinear systems by considering positively weighted neural architectures. To offer robustness certificates, we also developed new optimization techniques for training classifiers subject to spectral norm constraints on the weights. We studied various constrained formulations and showed that robustness can be secured without sacrificing accuracy when using a combination of tight constraints and exact projections. We also provide several lower-complexity solutions, which reduce the training time significantly.

Chapter 4

Signal denoising using new classes of robust neural networks

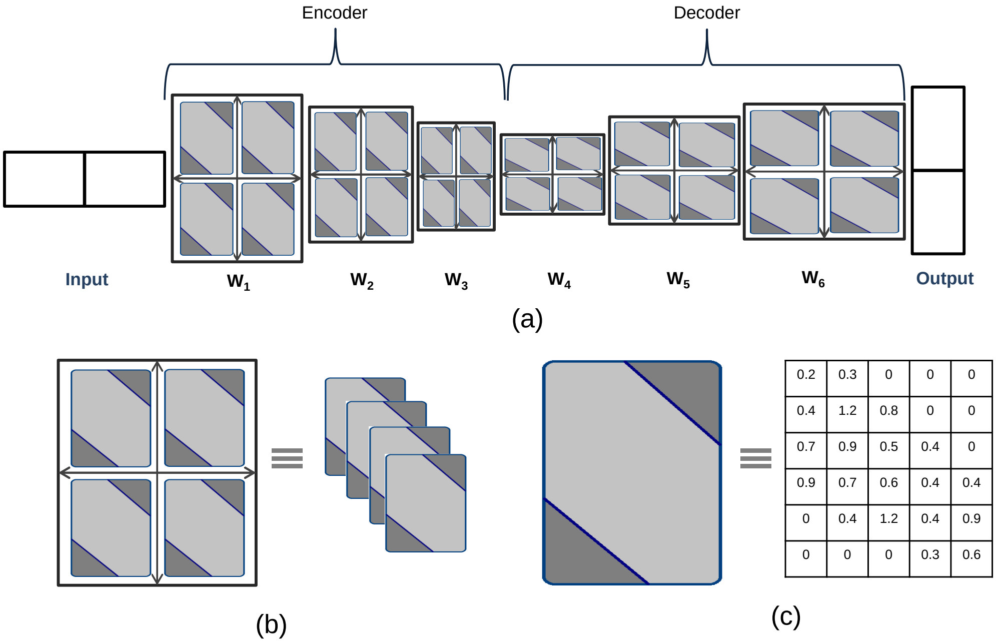
In this chapter, we focus on robust solutions for a regression problem, namely audio signal denoising. We address the task at hand from two perspectives. First, we only concentrate on denoising the magnitude elements resulting from a Fourier analysis of the audio signal. To this end, we design a fully connected network, called Adaptive Convolutional Neural Network (ACNN), whose layers have a special structure that exhibits some similarity with a 1D convolutional one. In the second part of the chapter, we extend our approach to denoising the whole complex spectrum of the audio signals, using complex-valued neural networks (CVNN). For both solutions, we derive tight Lipschitz bounds and propose robust training mechanisms which are later validated on denoising piano music clips corrupted by various levels of additive white noise.

4.1 Adaptive convolutional networks

This section introduces a new class of neural networks, called Adaptive Convolutional Neural Networks (ACNN), which can be seen as an intermediate between Convolutional Neural Networks (CNNs) and Fully Connected Networks (FCNs). Learning capabilities of CNNs being well-investigated and proven, we take advantage of this potential by structuring the weights of our network in a similar manner. A significant difference is that the network makes use of filters that are no longer time/space invariant, similar to what is done in adaptive filtering.

4.1.1 Making the bridge between CNNs and FCNs

In this section we aim at filling the gap between FCNs and CNNs. In terms of signal processing concepts, a convolutive layer is a Multiple-Input Multiple-Output (MIMO) filter. For one-dimensional signals, each of these filters can be viewed as a Tœplitz matrix generated by the impulse response of the filter, which is applied to the vector of signal samples. If the filter length is short, large upper and lower triangular parts of this matrix are null. In our proposed approach, we will keep this band structure for the weight



0.2	0.3	0	0	0
0.4	1.2	0.8	0	0
0.7	0.9	0.5	0.4	0
0.9	0.7	0.6	0.4	0.4
0	0.4	1.2	0.4	0.9
0	0	0	0.3	0.6

Fig. 4.1 Proposed architecture of Adaptive Convolutional Neural Network (ACNN). a) An encoder-decoder architecture composed of a 6-layer FCN followed by ReLU activation function. b) Relation between proposed FCNs and CNNs; the weights are split into sub-matrices simulating convolutive filters in CNNs c) Each of the sub-matrices is constrained to have a band structure as shown in this example. The dark grey area marks the zero-entries, while the light-grey colour corresponds to the ones that are allowed to be non-zero.

matrix, which is equivalent to performing local processing at each time within a sliding window. However, in order to add more flexibility to this architecture, we will allow all the nonzero coefficients of this matrix to be fully optimized. The proposed architecture is depicted in Figure 4.1a.

4.1.2 Learning algorithm

For training the proposed ACNN, we use a stochastic gradient-like optimization based on the popular ADAM method [22]. Consider the vector of parameters of the network, $\eta = (\eta_i)_{1\leqslant i\leqslant m}$, such that, for each layer $i \in \{1,\ldots,m\}$, η_i represents a vector of dimension $N_i(N_{i-1}+1)$, composed of the elements of the weight matrix W_i and the components of the bias vector b_i.

To secure the conditions of robustness while imposing the desired structure for our network, the parameter vector η is projected onto a closed set $\mathscr{S}$ that expresses all these constraints. The parameter update at epoch $n > 0$ is performed for mini-batches $(\mathbb{M}_{q,n})_{1\leqslant q\leqslant Q}$. If the training data are denoted by $(z_k)_{1\leqslant k\leqslant K}$, where z_k is the k-th pair of inputs and their associated outputs, the operations performed during the n-th epoch are summarized in Algorithm 2, where the square, the square root, and the division are

Algorithm 2: Projected ADAM Algorithm

```
Partition $\{1,\dots,K\}$ into mini-batches $(\mathbb{M}_{q,n})_{1\leqslant q\leqslant Q}$
foreach $q \in \{1,\dots,Q\}$ do
    $t = (n-1)Q + q$
    foreach $i \in \{1,\dots,m\}$ do
        $g_{i,t} = \sum_{k\in\mathbb{M}_{q,n}} \nabla_i \ell\big(z_k, (\eta_{i,t})_{1\leqslant i\leqslant m}\big)$
        $\mu_{i,t} = \beta_1 \mu_{i,t-1} + (1-\beta_1) g_{i,t}$
        $\nu_{i,t} = \beta_2 \nu_{i,t-1} + (1-\beta_2) g_{i,t}^2$
        $\gamma_t = \gamma\sqrt{1-\beta_2^t}/(1-\beta_1^t)$
        $\eta_{i,t+1} = \mathsf{P}_{\mathscr{S}_{i,t}}\Big(\eta_{i,t} - \gamma_t \mu_{i,t}/(\sqrt{\nu_{i,t}}+\varepsilon)\Big),$
```

				PSNR	MSE	CC
Noisy Signal				18.25	1.18×10^{-2}	0.76
Denoised Signal	Baseline - Wavelet-based denoiser			20.66	$1.00\times10-3$	0.80
	ACNN denoiser	Scenario (*i*)	$\overline{\vartheta}=1$	24.27	3.73×10^{-3}	0.96
			$\overline{\vartheta}=5$	29.03	1.25×10^{-3}	0.97
			$\overline{\vartheta}=10$	33.76	6.53×10^{-4}	0.98
		Scenario (*ii*)	$\overline{\vartheta}=1$	25.87	3.12×10^{-3}	0.96
			$\overline{\vartheta}=5$	30.63	8.63×10^{-4}	0.98
			$\overline{\vartheta}=10$	36.02	2.23×10^{-4}	0.99
	Standard FCN denoiser		$\overline{\vartheta}=1$	23.38	4.59×10^{-3}	0.90

Table 4.1 Comparison of different variants of the proposed method with baselines.

performed component-wise, and

$$\mathscr{S}_{i,t} = \big\{\eta_i \mid [(\eta_{j,t+1}^\top)_{j<i}\ \ \eta_i^\top\ \ (\eta_{j,t}^\top)_{j>i}]^\top \in \mathscr{S}\big\}. \tag{4.1}$$

4.1.3 Experimental Evaluation

The proposed network has been evaluated for denoising music signals.

Dataset Description

We train our proposed ACNN on a dataset consisting of musical exercises and songs performed on a Ronald organ. The organ covers 5 octaves (range `C2-C7`), each octave having 12 semitones, generating a total of 61 different possible notes. In total, the dataset contains 100 MIDI recordings, with a sampling frequency $F_s = 44100$ Hz, constituting 1 h and 17 min of audio. The data set is available online[1].

Experimental setup

The noisy data for training, validating, and testing is generated by adding white Gaussian noise to the original samples. The noise has zero mean and its standard deviation is randomly chosen so that the resulting signal-to-noise ratio (SNR) varies between

[1] https://speed.pub.ro/downloads/

5 and 30 dB. The dataset samples are normalized between 0 and 1. We extract the frequency features from the audio signal using a *Short-Time Fourier Transform* (STFT). The network estimates the STFT coefficients of the samples, and an *Inverse Short-Time Fourier Transform (ISTFT)* is performed as the post-processing step. We consider a `Hanning` sliding analysis window of length $T = 23$ ms, with an overlap between two consecutive windows of 50%. The STFT is performed on 1024 points. In total, from each audio segment, a vector of length $L = 513$ frequency coefficients is obtained, constituting the input of our ACNN.

The denoising is performed using a 6-layer ACNN architecture, as presented in Figure 4.1.

Simulations and results

In order to measure the performance of our proposed ACNN architecture, we perform two sets of experiments. In the first set, we control the Lipschitz constant of the architecture for three values $\overline{\vartheta}$ equal to 1, 5, and 10. In the second experiment, we test our architecture by varying the number of channels, i.e. the way we split each weight matrix.

We evaluate the performance on 3 standard metrics: *Peak to Signal Noise Ratio (PSNR)*, *Mean squared error (MSE)*, and *Cross-correlation (CC)*, as shown in the Table 4.1.

4.2 Design of robust complex-valued feed-forward neural networks

In this section, we introduce a new class of neural networks operating in the complex domain, called *Robust Complex Feed-Forward Network (RCFF-Net)*. The structure of the network is inspired by CapsNets [6, 38].

4.2.1 Theoretical background

A complex-valued feedforward neural network is defined as follows.

Model 4.2.1 Let $m \in \mathbb{N} \setminus \{0\}$. T is an m-layer complex-valued feedforward neural network if there exists $(N_i)_{0 \leqslant i \leqslant m} \in (\mathbb{N} \setminus \{0\})^{m+1}$ such that

$$T = T_m \circ \cdots \circ T_1 \tag{4.2}$$

where, for every $i \in \{1, \ldots, m\}$, $T_i = R_i(W_i \cdot + b_i)$, $W_i \in \mathbb{C}^{N_i \times N_{i-1}}$, $b_i \in \mathbb{C}^{N_i}$, and $R_i \colon \mathbb{C}^{N_i} \to \mathbb{C}^{N_i}$.

In the following, we will make the assumption that the activation operators $(R_i)_{1 \leqslant i \leqslant m}$ satisfy some nonexpansiveness properties and that all of them, except possibly for the last layer, are separable.

4.2.2 Nonexpansive complex-valued activation functions

There exist two main recipes for building activation functions, satisfying the imposed conditions. The first one is to use split-complex activation functions of the form

$$(\forall z \in \mathbb{C}) \quad \rho_{i,k}(\zeta) = \rho_{i,k}^{\mathrm{R}}(\mathrm{Re}\zeta) + \imath\rho_{i,k}^{\mathrm{I}}(\mathrm{Im}\zeta) \tag{4.3}$$

where $\rho_{i,k}^{\mathrm{R}} \colon \mathbb{R} \to \mathbb{R}$ and $\rho_{i,k}^{\mathrm{I}} \colon \mathbb{R} \to \mathbb{R}$ are α_i-averaged activation functions. The second recipe we propose is based on the following property.

Proposition 4.2.2 *Let $\omega\colon\ [0,+\infty[\to \mathbb{R}$ be α-averaged with $\alpha \in]0,1]$ and such that $\omega(0) = 0$. Let ρ be defined as*

$$(\forall \zeta \in \mathbb{C}) \quad \rho(\zeta) = \begin{cases} \dfrac{\omega(|\zeta|)}{|\zeta|}\zeta & \text{if } \zeta \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

4.2.3 Robustness results

Proposition 4.2.3 *Consider Model 4.2.1. For every $i \in \{1,\dots,m\}$, let $W_i^+ \in [0,+\infty[^{N_i \times N_{i-1}}$. Let $(\beta_{1,k})_{1 \leqslant k \leqslant N_0} \in [0,2\pi[^{N_0}$, let $(\beta_{m,k})_{1 \leqslant k \leqslant N_m} \in [0,2\pi[^{N_1}$, and for every $i \in \{2,\dots,m-1\}$, let $\beta_i \in [0,2\pi[$. Suppose that the weight operators of the network are such that*

$$\begin{aligned} &W_1 = W_1^+ \operatorname{Diag}\left(e^{\imath\beta_{1,1}},\dots,e^{\imath\beta_{1,N_0}}\right) \\ &(\forall i \in \{2,\dots,m-1\}) \quad W_i = e^{\imath\beta_i} W_i^+ \\ &W_m = \operatorname{Diag}\left(e^{\imath\beta_{m,1}},\dots,e^{\imath\beta_{m,N_m}}\right) W_m^+. \end{aligned} \tag{4.5}$$

Then

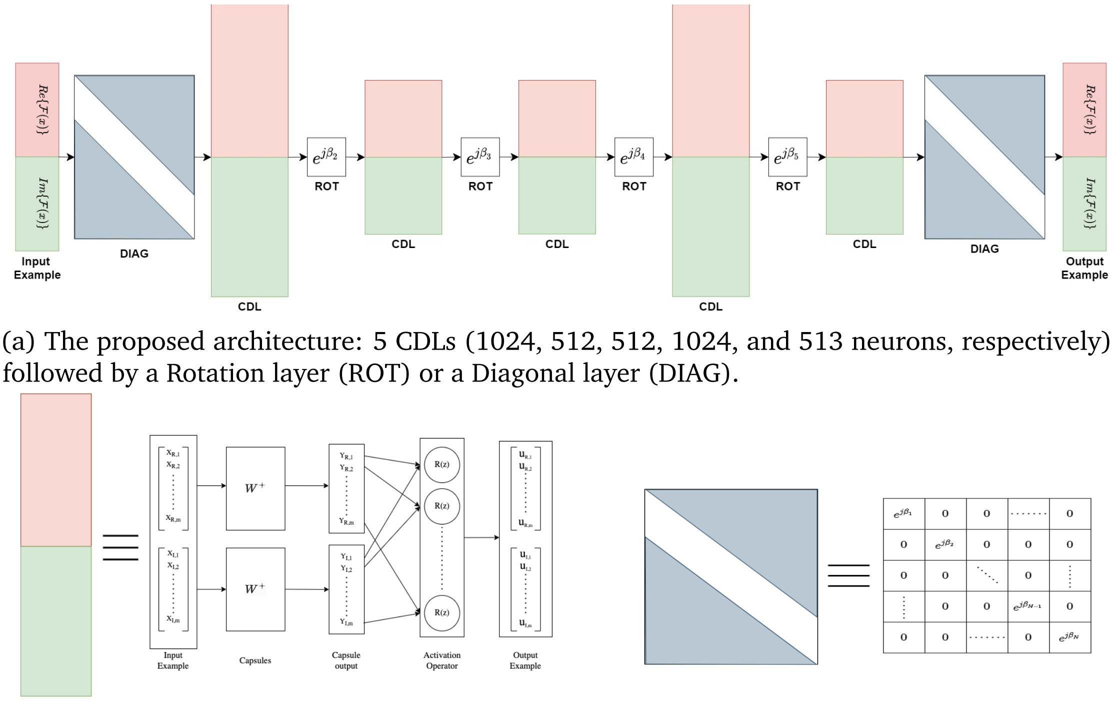
$$\theta_m = \|W_m^+ \cdots W_1^+\|. \tag{4.6}$$

4.2.4 Proposed approach

We implement our architecture to meet the requirements of Proposition 4.2.3 and design a Robust Complex Feed-Forward Neural Network (RCFF-Net). The architecture is illustrated in Figure 4.2. The network processes complex-valued data by stacking their real and imaginary parts.

Training strategy

Concerning the training strategy, we propose to use a similar approach to the case of ACNNs. We employ a projected version of the AdaMax optimizer [22].



(a) The proposed architecture: 5 CDLs (1024, 512, 512, 1024, and 513 neurons, respectively) followed by a Rotation layer (ROT) or a Diagonal layer (DIAG).

(b) The structure of the dense complex layer: each group of neurons (capsule) will process jointly the real part and the imaginary part of the coefficients.

(c) The structure of a diagonal layer: the white band corresponds to the main diagonal which features non-zero coefficients.

Fig. 4.2 Overview of the RCFF-Network. The red part denotes the real part, while the green accounts for the imaginary part.

```
Algorithm 3: Projected AdaMax Algorithm
  Partition $\{1,\dots,K\}$ into mini-batches $(\mathbb{M}_{q,n})_{1\leqslant q\leqslant Q}$
  foreach $q\in\{1,\dots,Q\}$ do
      $t=(n-1)Q+q$
      foreach $i\in\{1,\dots,m\}$ do
          $g_{i,t}=\sum_{k\in\mathbb{M}_{q,n}}\nabla_i\ell\big(z_k,(\eta_{i,t})_{1\leqslant i\leqslant m}\big)$
          $\mu_{i,t}=\chi_1\mu_{i,t-1}+(1-\chi_1)g_{i,t}$
          $\nu_{i,t}=\max(\chi_2\nu_{i,t-1},|g_{i,t}|)$
          $\gamma_{i,t}=\gamma\mu_{i,t}/(1-\chi_1^t)$
          $\eta_{i,t+1}=\mathsf{P}_{\mathscr{S}_{i,t}}\Big(\eta_{i,t}-\gamma_t\mu_{i,t}/(\sqrt{\nu_{i,t}}+\varepsilon)\Big)$, $\eta_{i,t+1}=\mathsf{P}_{\mathscr{S}_{i,t}}(\eta_{i,t}-\gamma_{i,t}/\nu_{i,t})$
```

In this algorithm, the modulus and the division are performed component-wise. Hereabove, ℓ denotes the loss function, ∇_i represents the gradients with respect to η_i. The vectors $\mu_{i,t}$ and $\nu_{i,t}$ represent the first and second momentum estimates at iteration t, using parameters $\chi_1=0.9$ and $\chi_2=0.999$. These variables are initialized with $\mu_{i,0}$ = $\nu_{i,0}$ = 0. Each gradient step is followed by a projection $\mathsf{P}_{\mathscr{S}_{i,t}}$ onto the constraint set $\mathscr{S}_{i,t}$. This set expresses the two constraints on which our approach is grounded.

Table 4.2 Experimental results for audio denoising

				MSE	**PSNR [db]**	**CC**
Noisy signal				7.21×10^{-3}	21.02	0.83
Baseline – Wiener Filter				3.45×10^{-3}	24.24	0.94
Baseline – NLMS Adaptive Filter				2.52×10^{-3}	25.61	0.95
Baseline – Standard FCN				2.78×10^{-3}	26.05	0.95
RCFF	$\rho(\zeta) = \mathbb{C}\mathrm{ReLU}(\zeta)$	U	$\theta_{\mathrm{upp}} = 335$	0.96×10^{-3}	30.00	0.99
		C	$\theta_m = 0.99$	2.02×10^{-3}	27.64	0.96
	$\rho(\zeta) = \mathrm{GK}(\zeta)$	U	$\theta_{\mathrm{upp}} = 73.25$	1.04×10^{-3}	29.45	0.97
		C	$\theta_m = 0.99$	2.11×10^{-3}	27.14	0.96
	$\rho(\zeta) = \frac{8}{3\sqrt{3}} \frac{\lvert\zeta\rvert}{1+\lvert\zeta\rvert^2} \zeta$	U	$\theta_{\mathrm{upp}} = 120$	0.96×10^{-3}	30.19	0.98
		C	$\theta_m = 0.93$	1.22×10^{-3}	29.02	0.97
	$\rho(\zeta) = \mathbb{C}\tanh(\zeta)$	U	$\theta_{\mathrm{upp}} = 421$	1.28×10^{-3}	28.98	0.97
		C	$\theta_m = 0.99$	2.09×10^{-3}	27.41	0.96
	$\rho(\zeta) = \frac{\zeta}{\sqrt{1+\lvert\zeta\rvert^2}}$	U	$\theta_{\mathrm{upp}} = 143$	1.90×10^{-3}	27.80	0.96
		C	$\theta_m = 0.97$	2.12×10^{-3}	26.98	0.96
	$\rho(\zeta) = \frac{\tanh(\lvert\zeta\rvert)}{\lvert\zeta\rvert}$	U	$\theta_{\mathrm{upp}} = 98$	1.43×10^{-3}	28.60	0.97
		C	$\theta_m = 0.98$	1.93×10^{-3}	27.63	0.97
	$\rho(\zeta) = \zeta^{\uparrow}$	U	$\theta_{\mathrm{upp}} = 187$	1.43×10^{-3}	30.21	0.98
		C	$\theta_m = 0.99$	1.09×10^{-3}	29.13	0.97
ACNN		C	$\theta_m = 1.00$	1.98×10^{-3}	26.24	0.96

Table 4.3 Experimental results for audio denoising with attacked inputs

				MSE	**PSNR [db]**	**CC**	Deg.[%]
Noisy signal				7.30×10^{-3}	21.00	0.83	0.09
Baseline – Standard FCN				5.46×10^{-3}	22.87	0.90	12.24
RCFF	$\rho(\zeta) = \mathbb{C}\mathrm{ReLU}(\zeta)$	U	$\theta_{\mathrm{upp}} = 335$	4.84×10^{-3}	23.62	0.91	21.26
		C	$\theta_m = 0.99$	1.96×10^{-3}	25.43	0.95	7.99
	$\rho(\zeta) = \mathrm{GK}(\zeta)$	U	$\theta_{\mathrm{upp}} = 73.25$	5.42×10^{-3}	23.31	0.90	20.84
		C	$\theta_m = 0.99$	1.84×10^{-3}	25.72	0.95	5.23
	$\rho(\zeta) = \frac{8}{3\sqrt{3}} \frac{\lvert\zeta\rvert}{1+\lvert\zeta\rvert^2} \zeta$	U	$\theta_{\mathrm{upp}} = 120$	5.26×10^{-3}	22.05	0.90	26.96
		C	$\theta_m = 0.93$	1.34×10^{-3}	28.68	0.97	1.17
	$\rho(\zeta) = \mathbb{C}\tanh(\zeta)$	U	$\theta_{\mathrm{upp}} = 421$	5.15×10^{-3}	23.14	0.90	22.14
		C	$\theta_m = 0.99$	2.82×10^{-3}	25.41	0.95	6.20
	$\rho(\zeta) = \frac{\zeta}{\sqrt{1+\lvert\zeta\rvert^2}}$	U	$\theta_{\mathrm{upp}} = 143$	6.02×10^{-3}	22.24	0.89	26.45
		C	$\theta_m = 0.97$	2.98×10^{-3}	25.12	0.94	8.14
	$\rho(\zeta) = \frac{\tanh(\lvert\zeta\rvert)}{\lvert\zeta\rvert}$	U	$\theta_{\mathrm{upp}} = 98$	5.78×10^{-3}	21.36	0.89	23.32
		C	$\theta_m = 0.98$	5.46×10^{-3}	25.56	0.95	5.61
	$\rho(\zeta) = \zeta^{\uparrow}$	U	$\theta_{\mathrm{upp}} = 187$	4.67×10^{-3}	23.09	0.90	22.34
		C	$\theta_m = 0.99$	1.45×10^{-3}	28.20	0.95	2.60
ACNN		C	$\theta_m = 1.00$	2.46×10^{-3}	25.43	0.95	3.08

4.2.5 Experimental results

The proposed methodology is applied to the same problem as in the previous section. We use a 5-layer RFCC-Net ($m = 5$), with diverse activation functions, and use the same pre-processing pipeline as in Section 4.1.3.

The main difference is that the network now estimates the complex STFT coefficients and, in the post-processing phase, an inverse operation (ISTFT) is performed for signal reconstruction.

We evaluate the performance of our RCFF-Net on the same 3 standard metrics: *Peak Signal-to-Noise Ratio (PSNR)*, *Cross-correlation (CC)*, and *Mean Squared Error (MSE)*, which was also employed as the training loss. The results on the test set are summarized in Table 4.2. We compare our solution with other standard denoising techniques, namely optimal Wiener filter and adaptive filter based on Normalised Least Mean Squares (NLMS) algorithm. As another baseline, we also trained a classical $m = 5$ layers Fully Connected

Network (FCN) with ReLU activation. Furthermore, we trained RCFF-Net both using constrained and unconstrained weights, referred to in Table 4.2 as `C` and `U`, respectively.

4.3 Conclusion

This chapter proposes two new classes of neural networks. The first one, ACNN, establishes a novel link between fully connected layers and convolutional structures, whereas the second one RCFF-Net operates in the complex space. By judiciously structuring the weight matrices, we derived a tight Lipschitz bound for both proposed architectures. In the complex case, our analysis led to new theoretical results concerning nonexpansive activation functions. We also extended an existing tight Lipschitz bound for feedforward neural networks to the complex domain. Computing this bound is no longer a combinatorial problem for complex-valued neural networks, which emphasizes the challenges raised with respect to the real case. We also showed how to control Lipschitz bounds numerically in the training process.

Chapter 5

ABBA neural networks: coping with positivity, expressivity, and robustness

In this chapter, we introduce ABBA networks, a novel class of (almost) non-negative neural networks, which are shown to possess a series of appealing properties. In particular, we demonstrate that these networks are universal approximators while enjoying the advantages of non-negative weighted networks. We derive tight Lipschitz bounds both in the fully connected and convolutional cases. We propose a strategy for designing ABBA nets that are robust against adversarial attacks, by finely controlling the Lipschitz constant of the network during the training phase. We show that our method outperforms other state-of-the-art defenses against adversarial white-box attackers. Experiments are performed on image classification tasks on four benchmark datasets.

5.1 Introduction

It is widely accepted that humans possess the innate ability to decompose complex interactions into discrete, intuitive hierarchical categories before analyzing them [26]. Conceptually, this evolution towards part-based representation in human cognition can be linked to non-negativity restrictions on the network weights [9]. This idea, along with other factors, has sparked interest in neural networks with non-negative weights.

Approach. We are interested in neural networks having non-negative weights, except for the first and last linear layers. We focus on a particular subclass of these networks for which the weight matrices have a structure of the form

$$\begin{bmatrix} A & B \\ B & A \end{bmatrix},$$

thus enjoying a number of algebraic properties. The corresponding networks are subsequently called ABBA networks. Note that weight matrices A and B are duplicated in ABBA networks, thus allowing us to limit the number of parameters.

5.2 Related work

Non-negative neural networks. Inspired by non-negative matrix factorization (NMF) techniques, the work of [9] introduces non-negative restrictions on the weights to create neural networks in which the hidden units correspond to identifiable concepts.

Link with other networks. From another perspective, the idea of using redundant weights is reminiscent of siamese networks [2]. These architectures are successfully used to handle similarity learning tasks, such as face verification [42], character recognition [23], and object tracking [20].

5.3 ABBA neural networks

5.3.1 Problem formulation

We will say that the activation operator R_i is symmetric, if there exists $(c_i, d_i) \in (\mathbb{R}^{N_i})^2$ such that

$$(\forall x \in \mathbb{R}^{N_i}) \quad R_i(x) - d_i = -R_i(-x + c_i). \tag{5.1}$$

In other words, $(c_i, d_i)/2$ is a symmetry center of the graph of R_i.

5.3.2 ABBA Matrices

We first define ABBA matrices, which will be the main algebraic tool throughout this chapter.

Definition 5.3.1 Let $(N_1, N_2) \in (\mathbb{N} \setminus \{0\})^2$. $\mathscr{A}_{N_1,N_2}$ is the set of ABBA matrices of size $(2N_2) \times (2N_1)$, that is $M \in \mathscr{A}_{N_1,N_2}$ 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

$$M = \begin{bmatrix} A & B \\ B & A \end{bmatrix}. \tag{5.2}$$

The sum matrix associated with M is then defined as $\mathfrak{S}(M) = A + B$.

5.3.3 Extension to feedforward networks

In this section we present the ABBA neural network for fully-connected layers

Definition 5.3.2 Let $m \in \mathbb{N} \setminus \{0\}$. $\widetilde{T}$ is an m-layer ABBA network if

$$\widetilde{T} = (\widetilde{W}_{m+1} \cdot + \widetilde{b}_{m+1}) \widetilde{T}_m \cdots \widetilde{T}_1 \widetilde{W}_0 \tag{5.3}$$

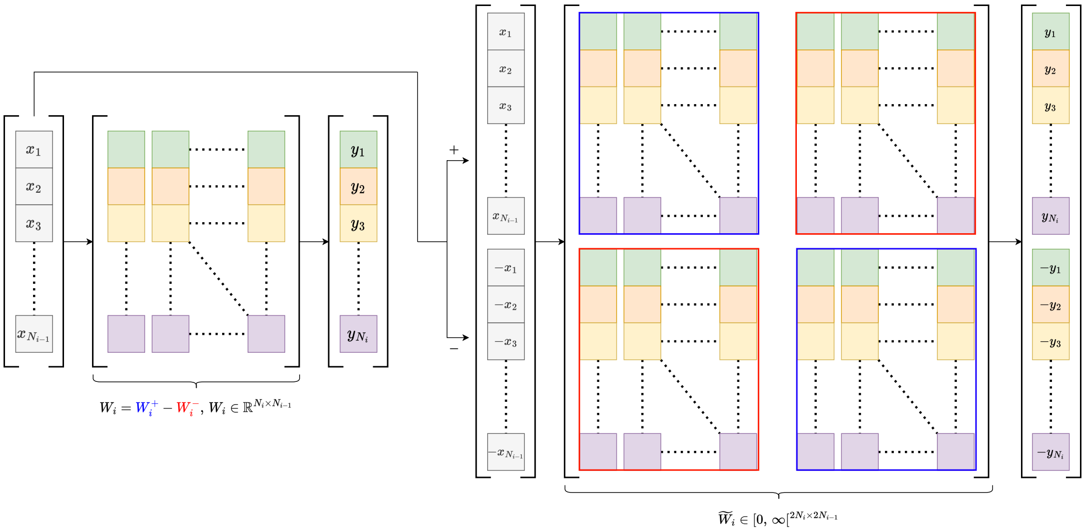

x_1
x_2
x_3
$x_{N_{i-1}}$
y_1
y_2
y_3
y_{N_i}
$W_i = W_i^+ - W_i^-$, $W_i \in \mathbb{R}^{N_i \times N_{i-1}}$
+
−
$-x_1$
$-x_2$
$-x_3$
$-x_{N_{i-1}}$
$-y_1$
$-y_2$
$-y_3$
$-y_{N_i}$
$\widetilde{W}_i \in [0, \infty[^{2N_i \times 2N_{i-1}}$

Fig. 5.1 Equivalence between a standard fully-connected layer and its ABBA correspondent.

with $\widetilde{W}_0 \in \mathbb{R}^{(2N_0)\times N_0}$, $\widetilde{W}_{m+1} \in \mathbb{R}^{N_m\times(2N_m)}$, $\widetilde{b}_{m+1} \in \mathbb{R}^{N_m}$, and

$$
\begin{aligned}
(\forall i \in \{1,\dots,m\}) \quad & \widetilde{T}_i = \widetilde{R}_i(\widetilde{W}_i \cdot + \widetilde{b}_i) && (5.4)\\
& \widetilde{R}_i \colon \mathbb{R}^{2N_i} \to \mathbb{R}^{2N_i}, && (5.5)\\
& \widetilde{b}_i \in \mathbb{R}^{2N_i}, && (5.6)\\
& \widetilde{W}_i \in \mathscr{A}_{N_i,N_{i-1}}, && (5.7)
\end{aligned}
$$

for given positive integers $(N_i)_{0\leqslant i\leqslant m}$. $\widetilde{T}$ is an m-layer non-negative ABBA network if it is an m-layer ABBA network as defined above and, for every $i \in \{1,\dots,m\}$, the elements of $\widetilde{W}_i$ are non-negative.

5.3.4 Link with standard neural networks

An illustration of the link between fully-connected layers and ABBA matrices is shown in Figure 5.1.

5.3.5 Expressivity of non-negative ABBA networks

One of the main advantages of non-negative ABBA networks with respect to standard networks with non-negative weights is that they are universal approximators.

5.3.6 Lipschitz bounds for ABBA fully-connected networks

In this section, we show that we can derive a simple expression for the Lipschitz constant, using a separable bound, for non-negative ABBA networks.

Proposition 5.3.3 *Let $m \in \mathbb{N}\setminus\{0\}$ and let $\widetilde{T} \in \mathscr{N}^{+}_{m,\mathscr{A}}$ be given by* (5.3)-(5.7)*. Assume that, for every $i \in \{1,\dots,m-1\}$, $\widetilde{R}_i$ is a separable nonexpansive operator. A Lipschitz constant of*

$\widetilde{T}$ *is*

$$\theta_m = \|\widetilde{W}_{m+1}\| \; \|\mathfrak{S}(\widetilde{W}_m)\cdots\mathfrak{S}(\widetilde{W}_1)\| \, \|\widetilde{W}_0\|. \tag{5.8}$$

The Lipschitz constant of $\widetilde{T}$ in (5.8) reduces to

$$\theta_m = \||W_m|\ldots|W_1|\|. \tag{5.9}$$

5.4 Convolutional networks

Here we extend the results presented in Section 5.3 to convolutional layers.

5.4.1 ABBA convolutional layers

The ABBA convolutional layer $\widetilde{W}_i$ has twice the number of input channels and twice the number of output ones. In this section we show the mathematical modelling of an ABBA convolutional layer.

5.4.2 Lipschitz bounds for convolutional networks

In this section, we establish bounds on the Lipschitz constant of an m-layer convolutional neural network T.

Theorem 5.4.1 *Let $(\sigma_i)_{1\leqslant i\leqslant m}$ be the aggregated stride factors of network T, and let*

$$W = (W_m)_{\uparrow\sigma_{m-1}} * \cdots * (W_2)_{\uparrow\sigma_1} * W_1 \tag{5.10}$$

where $(W_i)_{1\leqslant i\leqslant m}$ are the MIMO impulse responses of each layer of network T and, for every $i \in \{2,\ldots,m\}$, $(W_i)_{\uparrow\sigma_{i-1}}$ is the interpolated sequence by a factor σ_{i-1} of W_i. For every $\mathbf{j} \in \mathbb{S}(\sigma_m) = \{0,\ldots,\sigma_m-1\}^d$, we define the following matrix:

$$\overline{W}^{(\mathbf{j})} = \sum_{\mathbf{n}\in\mathbb{Z}^d} W(\sigma_m\mathbf{n}+\mathbf{j}) \in [0,+\infty[^{\zeta_m\times\zeta_0}. \tag{5.11}$$

Then

$$\theta_m = \Big\| \sum_{\mathbf{j}\in\mathbb{S}(\sigma_m)} \overline{W}^{(\mathbf{j})}\big(\overline{W}^{(\mathbf{j})}\big)^\top \Big\|^{1/2} \tag{5.12}$$

is a lower bound on the Lipschitz constant estimate of network T. In addition, if for every $i \in \{1,\ldots,m\}$, $p \in \{1,\ldots,\zeta_{i-1}\}$, and $q \in \{1,\ldots,\zeta_i\}$, $w_{i,q,p} = (w_{i,q,p}(\mathbf{n}))_{\mathbf{n}\in\mathbb{Z}^d}$ is a non-negative kernel, then θ_m is a Lipschitz constant of T.

5.4.3 Bounds for ABBA convolutional networks

Here we extend the previous results to the ABBA context.

Theorem 5.4.2 *Under the above assumptions on the convolutional ABBA network $\widetilde{T}$, let*

$$(\forall i \in \{1,\ldots,m\})(\forall \mathbf{j} \in \mathbb{S}(s_i)) \quad \Omega_i^{(\mathbf{j})} = \sum_{\mathbf{n}\in\mathbb{Z}^d} \mathfrak{S}(\widetilde{W}_i(s_i\mathbf{n}+\mathbf{j})) \in [0,+\infty[^{\zeta_i\times\zeta_{i-1}}, \tag{5.13}$$

where $(\widetilde{W}_i(\mathbf{n}))_{n\in\mathbb{Z}}$ is the MIMO impulse response of the ABBA layer of index i. Then a Lipschitz constant of $\widetilde{T}$ is

$$\overline{\theta}_m = \|\widetilde{W}_{m+1}\| \Big(\prod_{i=1}^{m} \Big\| \sum_{\mathbf{j}\in\mathbb{S}(s_i)} \Omega_i^{(\mathbf{j})} \big(\Omega_i^{(\mathbf{j})}\big)^\top \Big\|\Big)^{1/2} \|\widetilde{W}_0\|, \tag{5.14}$$

where $\|\widetilde{W}_{m+1}\|$ (resp. $\|\widetilde{W}_0\|$) is the spectral norm of the linear operator employed in the last (resp. first layer).

5.5 Lipschitz-constrained training

Algorithm 4: Projected ADAM Algorithm

```
Partition $\{1,\ldots,K\}$ into minibatches $(\mathbb{L}_{q,n})_{1\leqslant q\leqslant Q}$
$t = (n-1)Q + q$            # iteration index
# sweep minibatches
foreach $q \in \{1,\ldots,Q\}$ do
    foreach layer $i$ do
        $g_{i,t} = \sum_{k\in\mathbb{M}_{q,n}} \nabla_i \ell\big(z_k, (\Psi_{i,t})_{1\leqslant i\leqslant m}\big)$    # grad. computation
        $\mu_{i,t} = \beta_1\mu_{i,t-1} + (1-\beta_1)g_{i,t}$    # classical ADAM updates
        $\nu_{i,t} = \beta_2\nu_{i,t-1} + (1-\beta_2)g_{i,t}^2$
        $\gamma_t = \gamma\sqrt{1-\beta_2^t}/(1-\beta_1^t)$
        $\widetilde{\Psi}_{i,t} = \Psi_{i,t} - \gamma_t\mu_{i,t}/(\sqrt{\nu_{i,t}}+\varepsilon)$
    foreach layer $i$ do
        $\Psi_{i,t+1} = \mathrm{proj}_{\mathscr{S}_{i,t}}(\widetilde{\Psi}_{i,t})$            # projection step
```

5.6 Experiments

In this section, we show the versatility of ABBA neural networks in solving classification tasks. The objective of our experiments is three-fold.

(i) First, we compare positive ABBA structures with their classic non-negative counterparts and check that our method yields significantly better results in all considered cases.

(ii) We then train ABBA models constrained to different Lipschitz bound values and evaluate their robustness against several adversarial attacks.

(iii) Finally, we compare our proposed approach with three other well-established defense strategies, namely *Adversarial Training* (AT), *Trade-off-inspired adversarial defense* (TRADES) [46], and *Deel-Lip* proposed by [40].

Dataset	Network	Architecture	Accuracy [%]
MNIST	ABBA	Dense	98.33
		Conv	98.70
	Non-Negative	Dense	94.95
		Conv	93.27
	Baseline	Dense	98.35
		Conv	98.68
FMNIST	ABBA	Dense	90.02
		Conv	90.17
	Non-Negative	Dense	84.56
		Conv	83.09
	Baseline	Dense	90.00
		Conv	90.20
RPS	ABBA	Conv	99.08
	Non-Negative	Conv	67.30
	Baseline	Conv	98.86
CelebA	ABBA	Conv	90.21
	Non-Negative	Conv	61.04
	Baseline	Conv	90.17

Table 5.1 Comparison between ABBA, full non-negative and arbitrary-signed (baseline) networks.

We validate our ABBA networks on four benchmark image classification datasets: MNIST, its more complex variant Fashion MNIST, a variant of the Rock-Paper-Scissors (RPS) dataset [31], and a binary classification on CelebA [29].

5.7 Conclusions

In this chapter, we introduce ABBA networks, a novel class of neural networks where the majority of weights are non-negative. We demonstrate that these networks are universal approximators, possessing all the expressive properties of conventional signed neural architectures. Additionally, we unveil their remarkable algebraic characteristics, enabling us to derive precise Lipschitz bounds for both fully connected and convolutive operators.

Leveraging these bounds, we construct robust neural networks suitable for various classification tasks. For future research, it would be intriguing to explore the application of ABBA networks in regression problems, where controlling the Lipschitz constant may present more challenges. Moreover, extending our theoretical bounds to different structures, such as recurrent or attention-based networks, holds promise for further advancements.

Finally, we recognize the necessity of investigating the scalability of the proposed training method to deep architectures. One of the main hurdles in this endeavour is the increased number of parameters that deep ABBA architectures entail.

Chapter 6

Conclusions

6.1 Summary

Despite the fact that they may appear at the forefront of developments in Data Science, neural networks raise challenges in the areas of safety, privacy, and security due to their susceptibility to a wide variety of threats and perturbations that may arise while they are in operation. It is therefore vital to understand the reasons for neural network instability, identify the areas of concern, and develop solutions that aim to improve their stability in order to guarantee the existence of AI-based systems that are agnostic to small variations of their inputs.

During this thesis, our main focus was the design and training of neural networks that are intrinsically robust against adversarial perturbations of their inputs. Thus, we proposed several robust training techniques, and we proved their effectiveness in solving both classification and regression problems. We showed that our research is applicable to a wide range of applications and that its results may be useful in real-life scenarios as well.

First, we focused on simple feed-forward networks, that contain only linear layers. Our research started from the results established in [11], which state that in the case of non-negative weighted neural networks, tight Lipschitz bounds can be derived. We design several robust training algorithms, trying to achieve a good trade-off between robustness and performance.

6.2 Perspectives

In this section, we propose some possible extensions of the aforementioned methods that could be worth investigating in future works.

6.2.1 Training 1-Lipschitz denoisers

A possible way to extend the work presented in this thesis would be leveraging our established methods for controlling the Lipschitz constant of neural networks to generate 1-Lipschitz denoisers, as presented in [37].

6.2.2 Expanding the applications of complex-valued neural networks

In future works, it would be interesting to apply RCFF-Net to a larger panel of signal processing applications involving complex-valued data, like audio unmixing where robust CVNNs could play a pivotal role.

6.2.3 Controlling the Lipschitz constant of more complex layer structures

Given the progress made in this thesis, particularly in the effective management of the Lipschitz constant to enhance the stability of linear and convolutional layers within neural networks, a compelling prospect emerges for future research endeavours to extend to more complex structures such as recurrent ones.

6.2.4 Combining Lipschitz control with other certifiable defenses

In the context of improving neural networks' stability against adversarial threats, a promising avenue for future research lies in the integration of our current Lipschitz constant control mechanisms with complementary defense strategies. Of particular interest is the potential synergy between our approach and certified defenses, such as GloRoNets [27].

6.2.5 Studying the effect of other regularization techniques

Another interesting direction to follow would be the comprehensive study of the effects of various regularization techniques on the stability of the model.

6.2.6 Extending to other distances

Extending our current methods for controlling the robustness of neural networks to encompass other metrics is another research perspective. Presently, our techniques primarily address ℓ_2 perturbations, but the practicality of real-world systems demands a more comprehensive approach [4].

References

[1] Atzori, M., Gijsberts, A., Castellini, C., Caputo, B., Hager, A.-G. M., Elsig, S., Giatsidis, G., Bassetto, F., and Müller, H. (2014). Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Sci. data*, (140053).

[2] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1993). Signature verification using a ”siamese" time delay neural network. In *Proc. Ann. Conf. Neur. Inform. Proc. Syst.*, volume 6, pages 737–744.

[3] Carlini, N., Mishra, P., Vaidya, T., Zhang, Y., Sherr, M., Shields, C., Wagner, D., and Zhou, W. (2016). Hidden voice commands. In *USENIX Security Symp.*, pages 513–530.

[4] Carlini, N. and Wagner, D. (2017). Towards evaluating the robustness of neural networks. In *IEEE Symp. Security and Privacy*, pages 39–57.

[5] Chen, T., Lasserre, J.-B., Magron, V., and Pauwels, E. (2020). Semialgebraic optimization for Lipschitz constants of ReLU networks. pages 19189–19200.

[6] Cheng, X., He, J., He, J., and Xu, H. (2019). Cv-CapsNet: Complex-valued Capsule Network. *IEEE Access*, 7:85492–85499.

[7] Cheok, M. J., Omar, Z., and Jaward, M. H. (2019). A review of hand gesture and sign language recognition techniques. *Int. J. Mach. Learn. Cyber.*, 10(1):131–153.

[8] Chorowski, J. and Zurada, J. M. (2015). Learning understandable neural networks with nonnegative weight constraints. *IEEE Trans. Neural Netw. Learn. Syst.*, 26(1):62–69.

[9] Chorowski, J. and Zurada, J. M. (2015). Learning understandable neural networks with nonnegative weight constraints. In *IEEE Trans. Neural Net. and Learn. Syst.*, volume 26, pages 62–69.

[10] Cisse, M., Bojanowski, P., Grave, E., Dauphin, Y., and Usunier, N. (2017). Parseval networks: Improving robustness to adversarial examples. In *Proc. Int. Conf. Mach. Learn.*, pages 854–863.

[11] Combettes, P. L. and Pesquet, J.-C. (2020). Lipschitz certificates for layered network structures driven by averaged activation operators. In *J. Math. Data Sci.*, volume 2, pages 529–557.

[12] Combettes, P. L. and Pesquet, J.-C. (2021). Fixed point strategies in data science. *IEEE Trans. Sig. Proc.*, 69:3878–3905.

[13] Drucker, H. and LeCun, Y. (1992). Improving generalization performance using double backpropagation. *IEEE Trans. Neural Netw. Learn. Syst.*, 3:991–997.

[14] Fazlyab, M., Robey, A., Hassani, H., Morari, M., and Pappas, G. (2019). Efficient and accurate estimation of lipschitz constants for deep neural networks. In *Proc. Ann. Conf. Neur. Inform. Proc. Syst.*, pages 11423–11434.

[15] Gamarnik, D., Kızıldağ, E. C., and Zadik, I. (2021). Self-regularity of output weights for overparameterized two-layer neural networks. In *Proc. IEEE Int. Symp. Info. Theory*, pages 819–824.

[16] Geiping, J., Fowl, L. H., Huang, W. R., Czaja, W., Taylor, G., Moeller, M., and Goldstein, T. (2020). Witches' Brew: Industrial scale data poisoning via Gradient Matching. In *Int. Conf. Learn. Represent.*

[17] Goodfellow, I., Shlens, J., and Szegedy, C. (2015). Explaining and harnessing adversarial examples. In *Proc. Int. Conf. Learn. Represent.*

[18] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. (2017). Improved training of wasserstein gans. In *Proc. Ann. Conf. Neur. Inform. Proc. Syst.*, pages 5767–5777.

[19] Guo, C., Karrer, B., Chaudhuri, K., and van der Maaten, L. (2022). Bounding training data reconstruction in private (deep) learning. In *Proc. Int. Conf. Machine Learn.*, pages 8056–8071.

[20] He, A., Luo, C., Tian, X., and Zeng, W. (2018). A twofold siamese network for real-time object tracking. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 4834–4843.

[21] He, J., Baxter, S. L., Xu, J., Xu, J., Zhou, X., and Zhang, K. (2019). The practical implementation of artificial intelligence technologies in medicine. *Nat. med.*, 25(1):30–36.

[22] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *Proc. Int. Conf. Learning Represent.*

[23] Koch, G., Zemel, R., and Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In *Proc. Int. Conf. Machine Learn.*, volume 2.

[24] Kurakin, A., Goodfellow, I., and Bengio, S. (2016). Adversarial machine learning at scale. In *Proc. Int. Conf. Learn. Represent.*

[25] Latorre, F., Rolland, P., and Cevher, V. (2020). Lipschitz constant estimation of neural networks via sparse polynomial optimization. In *Int. Conf. on Learning Represent.*

[26] Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. In *Nature*, volume 401, pages 788–791.

[27] Leino, K., Wang, Z., and Fredrikson, M. (2021). Globally-robust neural networks. In *Proc. Int. Conf. Mach. Learn.*, pages 6212–6222.

[28] Li, P., Chen, X., and Shen, S. (2019). Stereo R-CNN based 3D object detection for autonomous driving. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 7644–7652.

[29] Liu, Z., Luo, P., Wang, X., and Tang, X. (2015). Deep learning face attributes in the wild. In *Proc. IEEE Int. Conf. Comput. Vis.*, volume 4, pages 3730–3738.

[30] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. (2018). Towards deep learning models resistant to adversarial attacks. In *Proc. Int. Conf. Learn. Represent.*

[31] Moroney, L. (2019). Rock, paper, scissors dataset.

[32] Neacşu, A., Pesquet, J.-C., and Burileanu, C. (2020). Accuracy-robustness trade-off for positively weighted neural networks. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, pages 8389–8393.

[33] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. (2016). The limitations of deep learning in adversarial settings. In *IEEE Symp. Security Privacy*.

[34] Parkinson, S., Ongie, G., and Willett, R. (2023). Linear neural network layers promote learning single- and multiple-index models. In *arXiv:2305.15598*.

[35] Pauk, J. (2008). Different techniques for EMG signal processing. *J. of Vibroeng.*, 10:571–576.

[36] Qi, J., Jiang, G., Li, G., Sun, Y., and Tao, B. (2019). Intelligent human-computer interaction based on surface EMG gesture recognition. *IEEE Access*, 7:61378–61387.

[37] Repetti, A., Terris, M., Wiaux, Y., and Pesquet, J.-C. (2022). Dual forward-backward unfolded network for flexible Plug-and-Play. In *Proc. European Signal Processing Conference*, pages 957–961.

[38] Sabour, S., Frosst, N., and Hinton, G. E. (2017). Dynamic routing between capsules. In *Proc. Ann. Conf. Neur. Inform. Proc. Syst.*, volume 30, pages 3856–3866.

[39] Scaman, K. and Virmaux, A. (2018). Lipschitz regularity of deep neural networks: analysis and efficient estimation. In *Proc. Ann. Conf. Neur. Inform. Proc. Syst.*, pages 3839–3848.

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

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

[42] Taigman, Y., Yang, M., Ranzato, M., and Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, pages 1701–1708.

[43] Takeru, M., Toshiki, K., Masanori, K., and Yuichi, Y. (2018). Spectral normalization for generative adversarial networks. In *Int. Conf. Learn. Represent.*

[44] Wen, F., Sun, Z., He, T., Shi, Q., Zhu, M., Zhang, Z., Li, L., Zhang, T., and Lee, C. (2020). Machine learning glove using self-powered conductive superhydrophobic triboelectric textile for gesture recognition in VR/AR applications. *Adv. Sci.*, 7(14):2000261.

[45] Widiastuti, N. (2019). Convolution neural network for text mining and natural language processing. In *IOP Conf.: Mater. Sci. Eng.*, volume 662.

[46] Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., and Jordan, M. (2019). Theoretically principled trade-off between robustness and accuracy. In *Proc. Int. Conf. Machine Learn.*, pages 7472–7482.