Recurrent neural network approximation theory

DISS. ETH NO. 31511

RECURRENT NEURAL NETWORK APPROXIMATION THEORY

A thesis submitted to attain the degree of

DOCTOR OF SCIENCES

(Dr. sc. ETH Zurich)

presented by

CLEMENS HUTTER

born on 23.05.1995

accepted on the recommendation of

Prof. Dr. Helmut Bölcskei
Prof. Dr. Philipp Petersen

2025

To Nora and Ann, for filling my life with joy.

Abstract

In this thesis we study the approximation capabilities of recurrent neural networks (RNNs). Firstly, we consider the approximation of certain classes of systems mapping an input sequence to an output sequence. We prove that RNNs can approximate any such system to within arbitrarily small worst-case error. What is more, we derive quantitative results on the amount of information needed to specify the approximating RNN. Furthermore, we present a framework to unify the study of different approximation tasks, allowing us to conclude that neural network learning is universally information-optimal for a large variety of approximation problems throughout both function and system approximation.

Secondly, we study the use of RNNs for approximating real-valued polynomials. We find that for every polynomial there exists an RNN that produces — as its output sequence — increasingly precise approximations to the target polynomial. This is remarkable as the weights of the RNN do not depend on the desired approximation error. That is, any arbitrarily small approximation error is achieved simply by running the approximating RNN for more time steps.

Kurzfassung

In dieser Dissertation untersuchen wir die Approximationseigenschaften von rekurrenten neuronalen Netzwerken (RNNs). Erstens betrachten wir die Approximation von bestimmten Systemklassen, die eine Eingangssequenz auf eine Ausgangssequenz abbilden. Wir beweisen, dass jedes System aus dieser Klasse durch RNNs beliebig genau approximiert werden kann. Darüber hinaus entwickeln wir quantitative Ergebnisse bezüglich der Menge an Information, die nötig ist, um das approximierende RNN zu spezifizieren. Ausserdem präsentieren wir einen Rahmen zur vereinheitlichten Betrachtung von verschiedenen Approximationsaufgaben. Dies erlaubt uns die Schlussfolgerung, dass neuronale Netzwerke universell informations-optimal sind für eine Vielzahl an Approximationsproblemen, sowohl im Bereich der Funktions- als auch der Systemapproximation.

Zweitens betrachten wir die Verwendung von RNNs zur Approximation von reellwertigen Polynomen. Wir zeigen, dass es für jedes Polynom ein RNN gibt, das als Ausgangssequenz zunehmend genaue Näherungen an das Zielpolynom produziert. Dies ist bemerkenswert, da die Gewichte des RNNs nicht vom gewünschten Approximationsfehler abhängen. Konkret kann jeder beliebig kleine Approximationsfehler erreicht werden, indem man das approximierende RNN für mehr Zeitschritte laufen lässt.

Acknowledgments

I would like to express my sincere gratitude to my supervisor, Prof. Helmut Bölcskei, for his patient guidance throughout my doctoral studies. He allowed me the freedom and necessary time to explore diverse research avenues, even when their outcomes were uncertain. I greatly appreciate the trust placed in me.

I am grateful to Prof. Philipp Petersen for acting as co-examiner for this thesis.

I would like to extend my gratitude to my amazing colleagues from the Chair for Mathematical Information Science, to Thomas, for being a great office mate and an even better teacher; to Yang, for finding everything perfect; to Konstantin, for being a great tour companion; to Yani, for involving me in something applied; to Recep, for getting me into research and being an exemplary neighbor; to Valentin, for broadening my horizon; to Weigutian, for his inspiring quiet confidence; to Erwin, for sharing his live experience; and to Michael, for keeping everything working.

Lastly, I want to thank my parents for making my life carefree.

Contents

Chapter 1

Introduction

Undoubtedly, RNNs have had a lasting influence on the field of machine learning, revolutionizing applications such as speech recognition [Gra+06], handwritten digit recognition [Gra+09] and language translation [SVL14]. In a nutshell, an RNN maps a discrete-time input sequence to an output sequence, by — in each time step — applying a neural network to the current input as well as a hidden state vector to produce the output and update the hidden state vector.

In this thesis, we investigate what kind of mappings RNNs can realize or approximate in principle. We do this largely through constructive existence proofs, essentially showing that — for the given task — an RNN exists with the desired properties, which also allows us to characterize the size or information efficiency of the approximating RNN. While undoubtedly important, the question of efficiently learning such a network from a noisy dataset (e.g., through gradient descent) is beyond the scope of this thesis.

Starting with the well known universal function approximation theorem [Cyb89; Fun89; HSW89], similar questions have extensively been studied for feedforward neural networks, also leading to quantitative results [Bar93; Bar94; Böl+19]. Interestingly, it was established that deep feedforward networks are significantly more efficient (in terms of network size) than shallow ones [Yar18; Yar17; Elb+21]. For RNNs, however, there is no single clear-cut way to define depth. Thus, in Chapter 2, we present common definitions of shallow and deep recurrent neural networks, largely following [Pas+14]. Furthermore, we discover an intimate relationship between depth and time. In particular, we show how to realize a deep RNN by a shallow one, at the cost of extra runtime. This is manifested by the shallow RNN's output sequence being delayed or interspersed with zeros relative to the deep RNN's output sequence.

Armed with this background, we then embark on our investigation of RNN approximation theory. In Chapter 3, we find that RNNs can universally realize linear systems, including time-varying systems given by finite time- and frequency-shifts. Furthermore, we aim to make this result quantitative, specifically for a class of linear time-invariant systems. To this end, we employ the concept of metric entropy [KT93] as a measure for the massiveness of the set of systems we wish to approximate, essentially asking what is the minimal number of ϵ-balls needed to cover this set of systems. We then construct a minimal covering where the ball-centers

are given by RNNs. The results in this chapter were developed based on the author's master thesis and published in [HGB22; HAB22].

Next, in Chapter 4, we widen our perspective and study certain classes of non-linear systems. Furthermore, we present a framework to unify the metric-entropy optimal approximation of functions (i.e. $\mathbb{R} \to \mathbb{R}$ mappings) by feedforward networks and of systems (i.e. $\mathbb{R}^{\mathbb{Z}} \to \mathbb{R}^{\mathbb{Z}}$ mappings) by recurrent neural networks. Thus, we are able to prove that neural network learning is universally optimal for approximation problems in both of these settings in a precisely defined sense (see Section 4.2.2). The results of this chapter are published in [PHB25].

Lastly, in Chapter 5, we leave the topic of approximating sequence-to-sequence maps and return to universal function approximation, however with a twist that we call reversed quantifier order. We notice that the traditional results follow the following pattern: For every function f (in a certain class) and every non-zero approximation error there exists a neural network achieving this error. With RNNs we are able to reverse the last two quantifiers in the case of approximating polynomials. Specifically, we show that for every polynomial f there exists an RNN that achieves any arbitrarily small approximation error. Concretely, the RNN accepts a point $x \in \mathbb{R}$ at time $t = 0$ as input and then produces as an output sequence increasingly precise approximations to $f(x)$, thus achieving any arbitrarily small approximation error after running for long enough. Importantly, the RNN weights do not depend on the desired approximation error. The results of this chapter are currently being prepared for submission.

Use of artificial intelligence (AI) technologies: While preparing the LaTeX document to produce this thesis, the author used Visual Studio Code (Microsoft, Washington, U.S.) as a text editor with GitHub Copilot (GitHub, San Francisco, U.S.) activated. Specifically, the author used the autocomplete function for generating suggestions for LaTeX code snippets and standard phrases, for example, in theorem statements. However, in this thesis no continuous paragraphs of text longer than two sentences were produced autonomously by generative AI technologies. Furthermore, the author used both the "chat" and "agent" functions of GitHub Copilot to assist in producing the figures contained in this thesis. Lastly, the author used LanguageTool (LanguageTooler GmbH, Potsdam, Germany) for spell checking.

Chapter 2

Types of Deep Recurrent Neural Networks

In this chapter, we provide a brief overview of the various ways to define (deep) RNNs. Moreover, we present results that allow to realize any deep RNN by a traditional shallow one. This is achieved by trading depth (i.e. layers in the deep RNN) for runtime of the shallow RNN. Since each chapter in this thesis is self-contained, and thus reiterates all relevant definitions, the present chapter can safely be skipped. Nonetheless, we feel it might benefit the interested reader by contextualizing the various notions of a deep recurrent neural network used throughout this thesis and the wider literature. Further note, that we consider general activation functions only in this chapter whereas in the remainder of the thesis we are only using the rectified linear unit (ReLU) activation.

We start by formally defining the set of sequences that constitute an RNN's input and output.

Definition 2.1. *We denote by $\mathcal{S}(d)$ the space of all one-sided sequences of vectors in $\mathbb{R}^d$. That is $\forall x \in \mathcal{S}(d)$*

$$x[t] \in \mathbb{R}^d \qquad \forall t \geq 0.$$

To simplify our definitions we furthermore introduce:

Definition 2.2. *For mappings $\mathcal{L} : \mathcal{S}(d_1) \to \mathcal{S}(d_2)$ and $\mathcal{L}' : \mathcal{S}(d_2) \to \mathcal{S}(d_3)$, we have the usual notion of concatenation $\mathcal{L}' \circ \mathcal{L} : \mathcal{S}(d_1) \to \mathcal{S}(d_3)$. Further, for a function $f : \mathbb{R}^{d_2} \to \mathbb{R}^{d_3}$ we define in a slight abuse of notation*

$$((f \circ \mathcal{L})x)[t] := f((\mathcal{L}x)[t]), \qquad \forall x \in \mathcal{S}(d_1), \forall t \geq 0,$$

i.e., the function f is applied at each time step to the output of $\mathcal{L}$.

Now we are ready to provide the most common definition for an RNN, which we call T(raditional)-RNN.

Definition 2.3 (T-RNN)**.** *[Pas+14; GBC16] Let $d, d', m \in \mathbb{N}$ be the input dimension, output dimension and hidden state size respectively. We define an operator $\mathcal{K} : \mathcal{S}(d) \to \mathcal{S}(m)$ mapping an input sequence $(x[t])_{t\geq 0}$ recursively to the sequence of hidden states $(h[t])_{t\geq 0}$ according to*

$$\begin{aligned} h[-1] &:= h^o \in \mathbb{R}^m \\ h[t] &= \rho_h(A_h h[t-1] + A_x x[t] + b_h) \end{aligned} \tag{2.1}$$

where $A_h \in \mathbb{R}^{m\times m}$, $b_h \in \mathbb{R}^m$, $A_x \in \mathbb{R}^{m\times d}$ and ρ_h is a nonlinearity. h^o is the initialization vector for the hidden state. Next, we define the map $\mathcal{Q} : \mathbb{R}^m \to \mathbb{R}^{d'}$

$$\mathcal{Q}(h) := \rho_o(A_o h + b_o),$$

where $A_o \in \mathbb{R}^{d'\times m}$, $b_o \in \mathbb{R}^{d'}$ and ρ_o is a nonlinearity. The RNN is the operator $\mathcal{R} : \mathcal{S}(d) \to \mathcal{S}(d')$ given by

$$\mathcal{R} := \mathcal{Q} \circ \mathcal{K}.$$

Remark 2.4. *A T-RNN according to Definition 2.3 with $\rho_o = Id$ and $h^o = 0$ is called Elman RNN [Elm90].*

There are a variety of ways to introduce depth to RNN computation [Pas+14]. We start with the so called stacked-RNN, which essentially treats the hidden state sequence of one RNN as the input sequence of the next-higher RNN in the stack.

Definition 2.5 (S-RNN)**.** *[Pas+14] Let $L, d, n_0, n_1, \ldots, n_L \in \mathbb{N}$. A stacked RNN is given as the operator $\mathcal{R} : \mathcal{S}(n_0) \to \mathcal{S}(d)$*

$$\mathcal{R} = \mathcal{Q} \circ \mathcal{K}_L \circ \mathcal{K}_{L-1} \circ \cdots \circ \mathcal{K}_1,$$

with $\mathcal{K}_\ell : \mathcal{S}(n_{\ell-1}) \to \mathcal{S}(n_\ell), \ell \in \{1, \ldots, L\}$ and $\mathcal{Q} : \mathbb{R}^{n_L} \to \mathbb{R}^d$ according to Definition 2.3.

Alternatively, one can also replace the simple recursion (2.1) by a deep neural network. To formalize this, we first define deep feed-forward neural networks.

Definition 2.6 (Deep Neural Network)**.** *[Elb+21, Definition II.1] Let $L, n_0, n_1, \ldots, n_L \in \mathbb{N}, L \geq 2$. A map $\Phi : \mathbb{R}^{n_0} \to \mathbb{R}^{n_L}$ given by*

$$\Phi(x) = \begin{cases} W_2 \circ \rho \circ W_1, & L = 2 \\ W_L \circ \rho \circ W_{L-1} \circ \rho \circ \cdots \circ \rho \circ W_1, & L \geq 3 \end{cases}$$

with affine linear maps $W_l : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}, l \in \{1, 2, \dots, L\}$*, and activation function* ρ *acting component-wise is called a feed-forward neural network. The map* W_l *corresponding to layer* l *is given by* $W_l(x) = A_l(x) + b_l$*, with* $A_l \in \mathbb{R}^{n_l \times n_{l-1}}$ *and* $b_l \in \mathbb{R}^{n_l}$*.*

Using this, we introduce deep transition (DT) RNNs and deep transition deep output (DOT) RNNs according to [Pas+14].

Definition 2.7 (DT-RNN and DOT-RNN)**.** *[Pas+14] Let* $d, d', m \in \mathbb{N}$*. We define an operator* $\mathcal{K} : \mathcal{S}(d) \to \mathcal{S}(m)$ *mapping an input sequence* $x[\cdot]$ *recursively to the sequence of hidden states* $h[\cdot]$ *according to*

$$h[-1] := h^o \in \mathbb{R}^m$$
$$h[t] = \Phi\left(\begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix}\right)$$

where $\Phi : \mathbb{R}^{d+m} \to \mathbb{R}^m$ *is a deep neural network according to Definition 2.6. Next, we define the map* $\mathcal{Q} : \mathbb{R}^m \to \mathbb{R}^{d'}$

$$\mathcal{Q}(h) := \rho_o(A_o h + b_o),$$

where $A_2 \in \mathbb{R}^{d' \times m}$*,* $b_2 \in \mathbb{R}^{d'}$ *and* ρ_o *is a nonlinearity. The operator* $\mathcal{R}_{DT} : \mathcal{S}(d) \to \mathcal{S}(d')$ *given by*

$$\mathcal{R}_{DT} := \mathcal{Q} \circ \mathcal{K},$$

is called a DT-RNN*.*
The operator $\mathcal{R}_{DOT} : \mathcal{S}(d) \to \mathcal{S}(d')$ *given by*

$$\mathcal{R}_{DOT} := \Psi' \circ \mathcal{K},$$

is called a DOT-RNN*, where* $\Psi : \mathbb{R}^m :\to \mathbb{R}^{d'}$ *is a deep neural network.*

If $\rho_o = Id$ we can unify the above definitions as follows.

Definition 2.8 (D-RNN)**.** *Let* $\Phi : \mathbb{R}^{m+d} \to \mathbb{R}^{m+d'}$ *be a Deep Neural Network according to Definition 2.6, where* $m \in \mathbb{N}$ *is the hidden state size. The D-RNN associated with* Φ *is the operator* $\mathcal{R} : \mathcal{S}(d) \to \mathcal{S}(d')$ *mapping an input sequence* $(x[t])_{t \geq 0}$ *in* $\mathbb{R}^d$ *to an output sequence* $(y[t])_{t \geq 0}$ *in* $\mathbb{R}^{d'}$ *according to*

$$h[-1] = (0 \dots 0)^T$$
$$\begin{pmatrix} y[t] \\ h[t] \end{pmatrix} = \Phi\left(\begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix}\right)$$

We next show that DOT-, DT-, and D-RNNs are equally expressive, since they can all be translated into each other.

Lemma 2.9. *Assume $\rho_o = \mathrm{Id}$, and that the activation function ρ allows the identity function to be represented by an arbitrarily deep network. Then, any DT-RNN or DOT-RNN can be represented by a D-RNN.*

Proof. Given a DT-RNN with transition network Φ and linear output map $\mathcal{Q}(h) = A_2 h + b_2$ we obtain a D-RNN realizing the same mapping by concatenating the linear map $\begin{pmatrix} \mathcal{Q} \\ \mathrm{Id} \end{pmatrix}$ to Φ. For a DOT-RNN with output network $\Psi(h)$ the D-RNN is instead obtained by concatenating $\begin{pmatrix} \Psi \\ \mathrm{Id} \end{pmatrix}$ to Φ. Here, Id is implemented as a deep neural network with the same number of layers as Ψ. □

Lemma 2.10. *Assume $\rho_o = \mathrm{Id}$. Any D-RNN can be represented as a DT-RNN.*

Proof. Given a D-RNN with transition network $\Phi = W_L \circ \rho \circ W_{L-1} \circ \rho \circ \cdots \circ \rho \circ W_1$ we construct a DT-RNN, i.e. Φ' and $\mathcal{Q}'$, that implements the same mapping. The hidden state of the DT-RNN will satisfy $h'[t] = \begin{pmatrix} y[t] \\ h[t] \end{pmatrix}$, where y and h are the hidden state and output sequences of the D-RNN respectively. To this end, we write the first linear map of Φ as

$$W_1 \left(\begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix} \right) = A_1^x x[t] + A_1^h h[t-1] + b_1.$$

The first linear map of Φ' is then taken to be

$$W_1' \left(\begin{pmatrix} x[t] \\ h'[t-1] \end{pmatrix} \right) = W_1' \left(\begin{pmatrix} x[t] \\ y[t-1] \\ h[t-1] \end{pmatrix} \right) = A_1^x x[t] + A_1^h h[t-1] + b_1,$$

i.e. the part of the augmented hidden state corresponding to the previous output is simply ignored. Then, we set the transition network of the DT-RNN as $\Phi' = W_L \circ \rho \circ W_{L-1} \circ \rho \circ \cdots \circ \rho \circ W_1'$ and its linear output mapping according to

$$\mathcal{Q}'(h'[t]) = \mathcal{Q}' \left(\begin{pmatrix} y[t] \\ h[t] \end{pmatrix} \right) = y[t].$$

□

Furthermore, it is possible to represent any S-RNN or D-RNN by a shallow T-RNN, however at the cost of requiring extra runtime. We start by showing that any L-layer S-RNN can be

represented by a T-RNN, which, however, produces the output sequence with a delay of $L-1$ time steps.

Definition 2.11. *Let $d \in \mathbb{N}$, $k \in \mathbb{Z}$, the shift operator $\mathbf{T}_k : \mathcal{S}(d) \to \mathcal{S}(d)$ is given by*

$$(\mathbf{T}_k x)[t] = \begin{cases} x[t-k], & \text{if } t-k \geq 0, \\ 0, & \text{otherwise,} \end{cases} \qquad \text{for } t \geq 0.$$

Theorem 2.12. *Given an S-RNN*

$$\mathcal{R} = \mathcal{Q} \circ \mathcal{K}_L \circ \mathcal{K}_{L-1} \circ \cdots \circ \mathcal{K}_1,$$

there exists a T-RNN

$$\mathcal{R}' = \mathcal{Q}' \circ \mathcal{K}'$$

such that

$$\mathbf{T}_{L-1}\mathcal{R} = \mathcal{R}'.$$

Proof. A similar result is presented in [Tur+19], and we therefore only sketch the proof here. Let $A_x^\ell, A_h^\ell, b_h^\ell$ be the parameters of $\mathcal{K}_\ell$ for $\ell \in \{1, \ldots, L\}$. The weights of $\mathcal{K}'$ are now obtained as

$$A'_x := \begin{pmatrix} A_x^1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \qquad A'_h := \begin{pmatrix} A_h^1 & 0 & \ldots & 0 & 0 \\ A_x^2 & A_h^2 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & A_h^{L-1} & 0 \\ 0 & 0 & \ldots & A_x^L & A_h^L \end{pmatrix}, \qquad b'_h := \begin{pmatrix} b_h^1 \\ \vdots \\ b_h^L \end{pmatrix}.$$

One can then show by induction that $h'[t]$, the hidden state sequence of the T-RNN, satisfies

$$h'[t] := \begin{pmatrix} h^1[t] \\ \vdots \\ h^L[t-(L-1)] \end{pmatrix}, \qquad \text{for } t \geq (L-1),$$

where $h^\ell[\cdot]$ is the hidden state sequence of $\mathcal{K}_\ell$ for $\ell \in \{1, \ldots, L\}$. Next, let $\widetilde{\mathrm{I}}$ be the (linear) operator that returns just the last block from $h'[\cdot]$, i.e. $\widetilde{\mathrm{I}}\, h'[t] = h^L[t-(L-1)]$. Then $\mathcal{Q}'$ is simply given by $\mathcal{Q}' = \mathcal{Q} \circ \widetilde{\mathrm{I}}$.

For initialization, we have to ensure that $h'_{\{\ell\}}$, the block of h' corresponding to h^ℓ, fulfills $h'_{\{\ell\}}[\ell-2] = h^\ell[-1]$ for $\ell \in \{1, \ldots, L\}$. One can force the hidden state accordingly by using Lemma 2.13, which we presented next. □

Lemma 2.13. *Let $N \in \mathbb{N}$. Given the next state operator $\mathcal{K} : \mathcal{S}(d) \to \mathcal{S}(m)$ of a T-RNN as in Definition 2.3 according to*

$$\begin{aligned} h[-1] &:= h^o \in \mathbb{R}^m \\ h[t] &= \rho_h(A_h h[t-1] + A_x x[t] + b_h) \end{aligned}$$

where $A_h \in \mathbb{R}^{m\times m}$, $b_h \in \mathbb{R}^m$, $A_x \in \mathbb{R}^{m\times d}$ and ρ_h is a nonlinearity such that $\rho_h(0) = 0$. Then for any set of vectors $k[t] \in \mathbb{R}^m$ for $t \in \{0, \dots, N-1\}$ there is a $\mathcal{K}'$ such that the first m koordinates of its hidden state satisfy the recursion equation

$$\begin{aligned} h'[-1] &= h^o \\ h'[t] &= \begin{cases} \rho_h(A_h h'[t-1] + k[t] + A_x x[t] + b_h) & \text{if } t \le N \\ \rho_h(A_h h'[t-1] + A_x x[t] + b_h) & \text{else} \end{cases} \end{aligned} \tag{2.2}$$

for any input sequence $x[\cdot]$.

Proof. Fix $\alpha, \beta \in \mathbb{R}$ such that $\rho_h(\alpha) = \beta$ and $\alpha, \beta \neq 0$. The idea is to augment the hidden state with an N-dinensional index vector $\widetilde{h}[t]$. That is, we write the hidden state sequence of $\mathcal{K}'$ as $\begin{pmatrix} h'[t] \\ \widetilde{h}[t] \end{pmatrix}$, where $h'[t]$ satisfies (2.2) and $(\widetilde{h}[t])_\ell = \begin{cases} \beta, & \text{if } t+2 = \ell, \\ 0, & \text{else} \end{cases}$. This is achieved by the following weights

$$A'_h = \begin{array}{c} \\ m \\ N \end{array} \overset{\overset{m}{\longleftrightarrow}\ \overset{N}{\longleftrightarrow}}{\left(\begin{array}{c|cccccc} A_h & k[0]/\beta & k[1]/\beta & \dots & k[N-2]/\beta & k[N-1]/\beta \\ \hline 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & \alpha/\beta & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \alpha/\beta & 0 \end{array}\right)}, \quad A'_x = \begin{pmatrix} A_x \\ 0 \end{pmatrix},$$

$$b'_h = \begin{pmatrix} b_h \\ 0 \end{pmatrix}$$

with initialization $\widetilde{h}[-1] = \begin{pmatrix} \beta \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ and $h'[-1] = h^o$. □

Next, we show how to realize any D-RNN by a T-RNN, which is a bit more involved. First note, that in a stacked-RNN the hidden state in each layer of the stack is computed from the previous hidden state in that layer by applying only a single nonlinearity. In contrast, for a

D-RNN, $L-1$ nonlinearities are applied before the next hidden state is produced. Thus, when simulating a D-RNN by a T-RNN, we need to run the later for $L-1$ time steps for each time step of the D-RNN. This is achieved by spreading out the input using the following operator.

Definition 2.14. *For $d, k \in \mathbb{N}$ we define the operator $\mathcal{Q} : \mathcal{S}(d) \to \mathcal{S}(d)$ that stretches the signal by adding $k-1$ zeros between the samples:*

$$(\mathcal{Q}_k x)[t] = \begin{cases} x\left[\frac{t}{k}\right] & \text{if } t \in k\mathbb{N}_0 \\ 0 & \text{otherwise} \end{cases}, \qquad t \geq 0.$$

Its left inverse is the down-sample operator

$$(\mathcal{Q}_k^{-1} x)[t] = x[kt], \qquad t \geq 0.$$

Theorem 2.15. *Let $L \geq 2$. For any L-Layer D-RNN $\mathcal{R}$ there exists a T-RNN $\widetilde{\mathcal{R}}$ with $\rho_o = \mathrm{Id}$ such that*

$$\mathcal{Q}_{L-1}^{-1} \mathbf{T}_{2-L} \widetilde{\mathcal{R}} \mathcal{Q}_{L-1} = \mathcal{R}. \tag{2.3}$$

Proof. Fix an arbitrary input sequence $x[\cdot]$ and denote by $y[\cdot] = (\mathcal{R}x)[\cdot]$ the output sequence of the D-RNN given by

$$\begin{aligned} h[-1] &= (0 \dots 0)^T \\ \begin{pmatrix} y[t] \\ h[t] \end{pmatrix} &= \Phi\left(\begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix}\right), \qquad t \geq 0, \end{aligned}$$

where

$$\Phi = W_L \circ \rho \circ W_{L-1} \circ \rho \circ \cdots \circ \rho \circ W_1.$$

We identify the weights and biases of the linear maps as

$$W_1(z) = \begin{pmatrix} A_1^x & A_1^h \end{pmatrix} z + b_1, \qquad W_L(z) = \begin{pmatrix} A_L^y \\ A_L^h \end{pmatrix} z + \begin{pmatrix} b_L^y \\ b_L^h \end{pmatrix}, \tag{2.4}$$

and $W_\ell(z) = A_\ell z + b_\ell$ for $\ell \in \{2, \dots, L-1\}$. Further, we introduce the notation

$$c_\ell[t] = (\rho \circ W_\ell \circ \cdots \circ \rho \circ W_2 \circ \rho \circ W_1) \begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix},$$

for $\ell \in \{1, \dots, L-1\}$, which denotes the intermediate activation of the deep transition network Φ at time step t. Hence,

$$c_{\ell+1}[t] = \rho(W_{\ell+1}(c_\ell[t])) = \rho(A_{\ell+1} c_\ell[t] + b_{\ell+1}), \tag{2.5}$$

and, by (2.4),

$$\begin{aligned} y[t] &= A_L^y c_{L-1}[t] + b_L^y \\ h[t] &= A_L^h c_{L-1}[t] + b_L^h. \end{aligned} \tag{2.6}$$

One time step of the D-RNN is visualized in Figure 2.1.

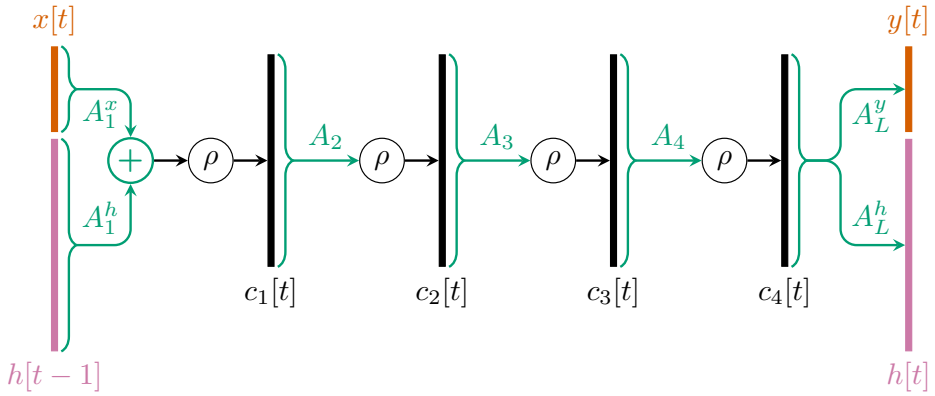


Figure 2.1: D-RNN with $L = 5$.

We now provide the weights for the corresponding T-RNN $\widetilde{\mathcal{R}}$ as

$$\widetilde{A}_h = \begin{pmatrix} 0 & A_{L-1} & 0 & \dots & 0 & 0 \\ 0 & 0 & A_{L-2} & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & A_3 & 0 \\ 0 & 0 & 0 & \dots & 0 & A_2 \\ A_1^h A_L^h & 0 & 0 & \dots & 0 & 0 \end{pmatrix} \qquad \widetilde{b}_h = \begin{pmatrix} b_{L-1} \\ \vdots \\ b_2 \\ b_1 + A_1^h b_L^h \end{pmatrix}$$

and

$$\widetilde{A}_x = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ A_1^x \end{pmatrix} \qquad \widetilde{A}_o = \begin{pmatrix} A_L^y & 0 & \dots & 0 \end{pmatrix} \qquad \widetilde{b}_o = b_L^y.$$

To simplify our exposition, we also segment $\widetilde{h}[\cdot]$, the hidden state sequence of $\widetilde{\mathcal{R}}$, accordingly, i.e.,

$$\widetilde{h}[\cdot] = \begin{pmatrix} \widetilde{h}_{L-1}[\cdot] \\ \vdots \\ \widetilde{h}_1[\cdot] \end{pmatrix}.$$

Further, we denote the input to the T-RNN by $\widetilde{x} = \mathcal{Q}_{L-1} x$, and its output by $\widetilde{y} = \widetilde{\mathcal{R}}\widetilde{x}$. Recall

that the recursive input output relation of the T-RNN is given according to Definition 2.3 by

$$\begin{aligned}\widetilde{h}[-1] &= \widetilde{h}^o \\ \widetilde{h}[\tau] &= \rho(\widetilde{A}_h\widetilde{h}[\tau-1] + \widetilde{A}_x\widetilde{x}[\tau] + \widetilde{b}_h) \\ \widetilde{y}[\tau] &= \widetilde{A}_o\widetilde{h}[\tau] + \widetilde{b}_o. \end{aligned} \tag{2.7}$$

In Figure 2.2, the hidden state sequence of $\widetilde{\mathcal{R}}$ is illustrated in response to the spread out input sequence $\widetilde{x} = \mathcal{Q}_{L-1}x$. Note, that for simplicity bias vectors are omitted in the figure.

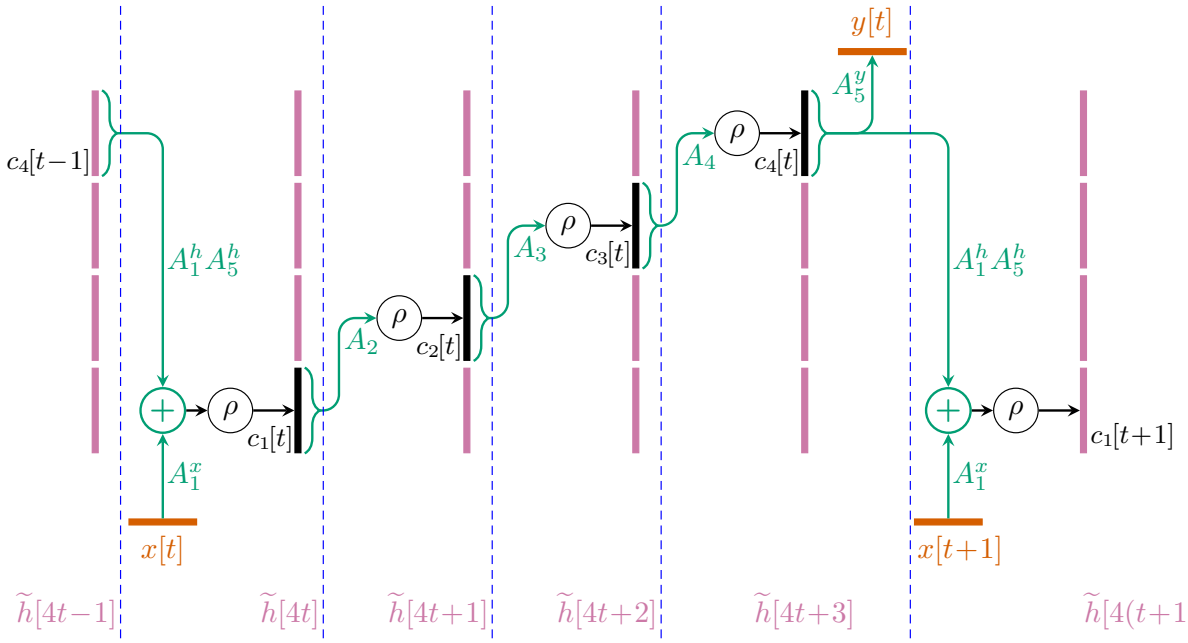


Figure 2.2: T-RNN corresponding to the D-RNN with $L = 5$.

To establish (2.3) we need to show that $\mathcal{Q}_{L-1}^{-1}\mathbf{T}_{2-L}\widetilde{y} = y$ or equivalently

$$\widetilde{y}[t(L-1) + (L-2)] = y[t], \quad \forall t \geq 0, \tag{2.8}$$

since $\mathcal{Q}_{L-1}^{-1}\mathbf{T}_{2-L}\widetilde{y} = \mathcal{Q}_{L-1}^{-1}\widetilde{y}[\cdot - (2-L)] = \widetilde{y}[\cdot(L-1) + (L-2)]$. We start by showing, that, $\forall t \geq 0$,

$$A_L^h\widetilde{h}_{L-1}[t(L-1)-1] + b_L^h = h[t-1], \qquad \text{and} \tag{2.9}$$

$$\widetilde{h}_\ell[t(L-1) + (\ell-1)] = c_\ell[t], \qquad \text{for } \ell \in \{1, \ldots, L-1\}. \tag{2.10}$$

The proof is via nested induction over t and ℓ. We will establish the base case $t = 0$ for the

induction over t later and thus assume that (2.9) holds for some $t \geq 0$. We calculate

$$\begin{aligned}
\widetilde{h}_1[t(L-1)] &= \rho\left(A_1^h A_L^h \widetilde{h}_{L-1}[t(L-1)-1] + A_1^x \widetilde{x}[t(L-1)] + b_1 + A_1^h b_L^h\right) \\
&= \rho\left(A_1^h\left(A_L^h \widetilde{h}_{L-1}[t(L-1)-1] + b_L^h\right) + A_1^x x[t] + b_1\right) \\
&= \rho\left(A_1^h h[t-1] + A_1^x x[t] + b_1\right) \\
&= \rho\left(W_1 \begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix}\right) = c_1[t],
\end{aligned}$$

where we use that $\widetilde{x}[t(L-1)] = (\mathcal{Q}_{L-1}x)[t(L-1)] = x\left[\frac{t(L-1)}{L-1}\right] = x[t]$. This establishes (2.10) for $\ell = 1$. Next, assume (2.10) holds for some $\ell \in \{1, \dots, L-2\}$. Then

$$\begin{aligned}
\widetilde{h}_{\ell+1}[t(L-1)+\ell] &= \rho\left(A_{\ell+1} h_\ell[t(L-1)+\ell-1] + b_{\ell+1}\right) \\
&= \rho\left(A_{\ell+1} c_\ell[t] + b_{\ell+1}\right) \\
&\overset{(2.5)}{=} c_{\ell+1}[t],
\end{aligned}$$

which establishes the inductive step for ℓ and thus proves (2.10). In particular,

$$\widetilde{h}_{L-1}[(t+1)(L-1)-1] = \widetilde{h}_{L-1}[t(L-1)+(L-1)-1] = c_{L-1}[t],$$

which, by (2.6), implies

$$A_L^h \widetilde{h}_{L-1}[(t+1)(L-1)-1] + b_L^h = A_L^h c_{L-1}[t] + b_L^h = h[t].$$

Comparing with (2.9) this establishes the induction step for the induction over t. It remains to show the base case of (2.9) for $t = 0$, that is,

$$A_L^h \widetilde{h}_{L-1}[-1] + b_L^h = h[-1].$$

For invertible A_L^h this can be achieved by setting $\widetilde{h}^o$, the initial hidden state of the T-RNN $\widetilde{\mathcal{R}}$, according to

$$\widetilde{h}_{L-1}[-1] = \widetilde{h}^o := (A_L^h)^{-1}(h[-1] - b_L^h) = (A_L^h)^{-1}(h^o - b_L^h),$$

where h^o is the initialization of the original D-RNN hidden state. If this is not the case one can initialize $\widetilde{h}^o = 0$ and use Lemma 2.13 with $N = 1$ and $k[0] = \begin{pmatrix} A_1^h\left(h^o - b_L^h\right) \\ 0 \\ \vdots \end{pmatrix}$. Then we would

have, by Lemma 2.13,

$$\begin{aligned}\widetilde{h}'_1[0] &= \rho\left(A_1^h A_L^h \widetilde{h}'_{L-1}[-1] + A_1^x \widetilde{x}[0] + b_1 + A_1^h b_L^h + A_1^h\left(h^o - b_L^h\right)\right)\\ &= \rho\left(A_1^h\left(A_L^h \widetilde{h}^o + h^o\right) + A_1^x x[0] + b_1\right)\\ &= \rho\left(A_1^h h[-1] + A_1^x x[0] + b_1\right) = c_1[0],\end{aligned}$$

which establishes (2.10) for $t = 0$ and $\ell = 1$. Hence, the rest of the nested induction goes through as above.

To conclude, note that (2.10) implies in particular that, $\forall t \geq 0$,

$$\widetilde{h}_{L-1}[t(L-1) + (L-2)] = c_{L-1}[t],$$

and, by (2.7),

$$\widetilde{y}[t(L-1) + (L-2)] = \widetilde{A}_o \widetilde{h}[t(L-1) + (L-2)] + b_o = A_L^y c_{L-1}[t] + b_y^L = y[t],$$

which is exactly (2.8) and thus concludes the proof. □

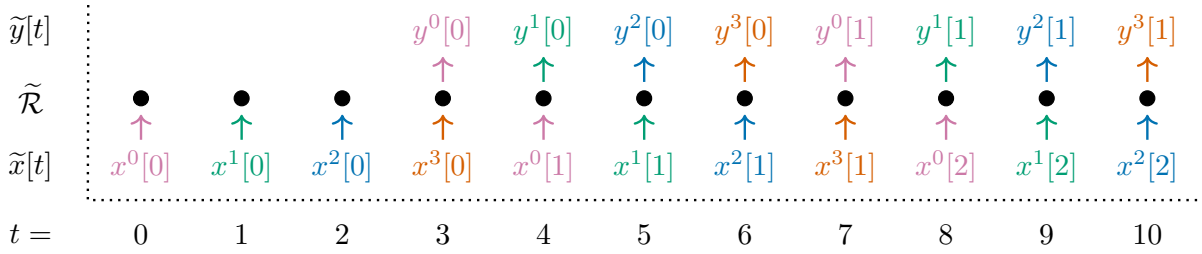


Figure 2.3: Multiple inputs can be processed simultaneously.

Corollary 2.16. *For a D-RNN $\mathcal{R}$ the T-RNN $\widetilde{\mathcal{R}}$ constructed according to Theorem (2.15) can process $L-1$ inputs in parallel. Specifically, we let $x^0, \ldots, x^{L-2} \in \mathcal{S}(d)$ be input sequences and $y^0 = \mathcal{R}x^0, \ldots, y^{L-2} = \mathcal{R}x^{L-2}$ be the corresponding output sequences of the D-RNN. Further, we interleave the inputs to form*

$$\widetilde{x}[t] = x^k\left[\frac{t-k}{L-1}\right], \qquad \text{with } k = t \mod (L-1).$$

Now the output of the T-RNN $\widetilde{y} = \widetilde{\mathcal{R}}\widetilde{x}$ satisfies

$$\widetilde{y}[t] = y^k\left[\frac{t-(L-2)-k}{L-1}\right], \qquad \text{with } k = (t-(L-2)) \mod (L-1).$$

See Figure 2.3 for an illustration.

Chapter 3

Metric Entropy Limits on Recurrent Neural Network Learning of Linear Dynamical Systems

3.1. Introduction

During the past decade, recurrent neural networks (RNNs) have revolutionized numerous machine learning applications, such as handwritten text recognition [Gra+09], speech recognition [Gra+06], language translation [SVL14], and modeling of complex game dynamics [Sch+20]. Abstractly speaking, an RNN realizes a dynamical system mapping an input sequence to an output sequence through—in each time step—application of a single-hidden-layer neural network to update a hidden state vector and compute the output signal sample. It is hence natural to ask which classes of dynamical systems can be realized or approximated by RNNs. This question is inspired by the well-known universal approximation theorem for feedforward neural networks [HSW89; Fun89; Cyb89], which states that every continuous function on a compact interval can be approximated to within arbitrarily small error by a single-hidden-layer neural network, provided that the number of neurons is allowed to go to infinity as the approximation error approaches zero.

The first central result in this chapter establishes that RNNs universally exactly realize the class of linear dynamical systems, including time-varying systems. This universal linear dynamical system realization theorem builds on a strong representation theorem for general linear operators stemming from harmonic analysis [Grö01, Theorem 14.3.5],[Fef83; MBH13], which states that every "reasonable" linear operator can be written as a weighted superposition of time-frequency shift operators. Note that we conspicuously use the term "realization theorem" instead of "approximation theorem" as RNNs with real-valued weights will, indeed, be shown to exactly realize general linear dynamical systems.

The second central theme of this chapter revolves around making the universal system realization theorem quantitative. Specifically, we consider classes of linear dynamical systems,

quantify their complexity through metric entropy according to [Zam79; ZO93], and ask how the number of bits needed to uniquely specify RNNs approximating systems in this class to within a prescribed error relates to the class's metric entropy. This part of the theory we develop is restricted to linear time-invariant (LTI) systems for conceptual reasons. The main result we obtain states that RNNs with suitably quantized weights provide optimal coverings—in the sense of metric entropy—for the class of LTI systems with exponentially decaying impulse response. In control theory parlance, this says that RNNs can be trained to identify LTI systems with exponentially decaying impulse response in a metric-entropy optimal fashion. We also show that, equivalently, this means that certain classes of linear difference equations with constant coefficients can be learned optimally by RNNs. The overall philosophy of the framework we propose is inspired by the recently established Kolmogorov-Donoho rate-distortion theory [KT93; Don01; Don96; Don+98] for feedforward neural networks [Böl+19; Elb+21] which shows that deep neural networks provide optimal coverings for a wide range of function classes, such as unit balls in Besov spaces and in modulation spaces.

We hasten to add that, throughout the chapter, we are exclusively concerned with the fundamental representation capabilities of RNNs and do not consider the issue of learning algorithms, a topic that has been investigated in the context of LTI system identification in [HMR18; Li+21].

Previous work on the approximation of linear dynamical systems through neural networks deals with (linear and nonlinear) time-invariant systems and assumes that the system is specified in terms of a state space representation, concretely by a (time-invariant) next-state function which is approximated by a single-hidden-layer neural network, the existence of which is guaranteed by the classical universal approximation theorem [HSW89; Fun89; Cyb89]. This approach leads, however, to the accumulation of errors over time so that most results along these lines are restricted to finite time horizons [SZ07; Son92; FN93]. A notable exception in this regard is [Mat93], which avoids error build-up by imposing an "absolute summability" condition on the system's possible state trajectories. Nonetheless, all these results require that the system be characterized by a state space representation, the existence of which is not guaranteed for a given linear dynamical system [HRS21, Theorem 2.3.3]. In the present chapter, we do not impose such an existence assumption. In addition, our theory comprises time-varying systems and pertains to unbounded time horizons, but, as already mentioned, is restricted to linear systems.

As for our second central theme, namely metric-entropy-optimal RNN learning of LTI systems, to the best of our knowledge, such an approach has not been pursued before in the literature. Related previous work reported in [Li+21] quantifies the number of real-valued RNN weights required for a desired approximation quality, but does not attempt to specify the approximating RNNs through bitstrings of finite length.

We furthermore want to highlight work on a non-recurrent neural network architecture, termed "Deep operator network" [CC95; Lu+21; LMK22], which enables the universal approxi-

mation of nonlinear operators. Finally, RNNs have also been investigated for the approximation of algorithms, with a prominent result [SS95] proving that RNNs with binary input and output sequences and rational weights can simulate any Turing machine.

Outline of the chapter In the remainder of this section, we provide preparatory material on RNNs and on harmonic analysis of general linear dynamical systems. In Section 3.2, we develop the first central result of the chapter, namely a universal realization theorem for discrete-time linear dynamical systems. In Section 3.3, we introduce the concept of metric entropy of classes of LTI systems, based on which, in Section 3.4, we state the second central result establishing that RNNs realize LTI systems of exponentially decaying impulse response in a metric-entropy-optimal fashion.

Notation Vectors are indexed starting with $\ell = 1$. $\mathbb{1}_{\{\cdot\}}$ denotes the truth function which takes on the value 1 if the statement inside $\{\cdot\}$ is true and equals 0 otherwise. Sequences $x[t] \in \mathbb{R}$ are indexed by $t \in \mathbb{Z}$. The $N \times N$ identity matrix is $\mathbb{I}_N$ and $\mathbb{O}_N$ stands for the $N \times N$ all zeros matrix. 1_N and 0_N denote the N-dimensional column vector with all entries equal to 1 and 0, respectively. $\log(\cdot)$ refers to the natural logarithm. We write $f(\epsilon) \sim g(\epsilon)$ to mean $\lim_{\epsilon\to 0} \frac{f(\epsilon)}{g(\epsilon)} = 1$ and $g(\epsilon) = o(f(\epsilon))$ to express that $\lim_{\epsilon\to 0} \left|\frac{g(\epsilon)}{f(\epsilon)}\right| = 0$. Throughout the chapter, constants are understood to be in $\mathbb{R}$ unless explicitly stated otherwise.

3.1.1. Recurrent Neural Networks

A recurrent neural network (RNN) is described by a hidden state vector sequence $h[t]$, the input signal $x[t]$, and the output signal $y[t]$. In each time instant t, a single-hidden-layer neural network is applied to the concatenation of the input sample $x[t]$ and the previous state vector $h[t-1]$ to produce the current output sample $y[t]$ and the new state vector $h[t]$. The formal definition is as follows.

Definition 3.1.1 (Recurrent neural network)**.** *For $n \in \mathbb{N}$ and hidden state dimension $m \in \mathbb{N}$, let $\Phi : \mathbb{R}^{m+1} \to \mathbb{R}^{m+1}$ be a feedforward neural network given by*

$$\Phi(x) = A_2\, \rho(A_1 x + b_1) + b_2, \quad x \in \mathbb{R}^{m+1}, \tag{3.1}$$

with weight matrices $A_1 \in \mathbb{R}^{n\times(m+1)}$, $A_2 \in \mathbb{R}^{(m+1)\times n}$, bias vectors $b_1 \in \mathbb{R}^n$, $b_2 \in \mathbb{R}^{m+1}$, and the ReLU activation function $\rho(x) = \max\{x, 0\}$, $x \in \mathbb{R}$, applied element-wise. The recurrent neural network associated with Φ is the operator $\mathcal{R}_\Phi : \ell^\infty \to \mathbb{R}^{\mathbb{N}_0}$ mapping input sequences $(x[t])_{t\geq 0}$ in $\mathbb{R}$ to output sequences $(y[t])_{t\geq 0}$ in $\mathbb{R}$ according to

$$\begin{pmatrix} y[t] \\ h[t] \end{pmatrix} = \Phi\left(\begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix}\right), \ \forall t \geq 0, \tag{3.2}$$

where $h[t] \in \mathbb{R}^m$ is the hidden state sequence with initial state $h[-1] = 0_m$.

Remark 3.1.2. *Classical RNN definitions are often referred to as Elman networks [Elm90; GBC16]. We show in Appendix 3.A that our RNN definition does not afford increased generality over Elman networks as every RNN according to Definition 3.1.1 can be converted into an Elman RNN. We decided, however, to work with the seemingly more general Definition 3.1.1 for expositional simplicity.*

We now introduce a decomposition of the weight matrix A_2 which will simplify the description of RNN constructions later in the chapter. Specifically, we represent A_2 according to

$$A_2 = \begin{pmatrix} A_o A_r \\ A_h \end{pmatrix} \in \mathbb{R}^{(m+1)\times n}, \tag{3.3}$$

where $A_h \in \mathbb{R}^{m\times n}$ is responsible for mapping to the next hidden state and, for some $R \in \mathbb{N}$, $A_r \in \mathbb{R}^{R\times n}$ maps to an R-dimensional virtual representation which, in turn, is linearly combined through the weights $A_o \in \mathbb{R}^{1\times R}$ to deliver the output $y[\cdot]$. Consorting with this decomposition of A_2 and noting that $b_2 = 0_{m+1}$ in all our concrete RNN constructions, the hidden state sequence evolution can be written as

$$h[t] = A_h g[t], \quad \forall t \geq 0, \tag{3.4}$$

where

$$g[t] = \rho\left(A_1 \begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix} + b_1\right), \quad \forall t \geq 0, \tag{3.5}$$

and the initialization is $h[-1] = 0_m$ as before. The output sequence is accordingly obtained as

$$r[t] = A_r g[t], \tag{3.6}$$

$$y[t] = A_o r[t], \tag{3.7}$$

where $r[t] \in \mathbb{R}^R$ denotes the virtual representation sequence. We note that this virtual representation never actually manifests itself, it is introduced solely to simplify the specific RNN constructions later in the chapter. Finally, we remark that, throughout, whenever we speak of "weights" of the RNN, this shall refer to nonzero entries both in the weight matrices A_1, A_2 and the bias vectors b_1, b_2.

3.1.2. Harmonic Analysis of Linear Dynamical Systems

We consider discrete-time causal linear systems $\mathcal{L}$ mapping input sequences $x[\cdot] \in \ell^\infty$ to output sequences $y[\cdot] \in \ell^\infty$, and we use the convention

$$x[t] = 0, \qquad \forall t < 0, \tag{3.8}$$

which, by causality and linearity, implies $y[t] = 0$, $\forall t < 0$.

A fundamental result from harmonic analysis [Grö01, Theorem 14.3.5], in its incarnation for discrete-time systems, states that a wide class of linear operators, i.e., linear dynamical systems, can be represented as a weighted superposition of time-frequency shift operators according to

$$y[t] = \sum_{\tau=0}^{\infty} \int_0^1 S_{\mathcal{L}}(\tau, \nu) x[t-\tau] e^{2\pi i \nu t} d\nu, \tag{3.9}$$

with the weights given by the delay-Doppler spreading function $S_{\mathcal{L}}(\tau, \nu)$. Alternatively, (3.9) can be expressed in terms of the operator kernel, a.k.a. time-varying impulse response, $k[t, \tau]$, as

$$y[t] = \sum_{\tau=0}^{\infty} k[t, \tau] x[t-\tau], \tag{3.10}$$

where $k[t, \tau]$ is related to the spreading function through an inverse Fourier transform according to

$$k[t, \tau] = \int_0^1 S_{\mathcal{L}}(\tau, \nu) e^{2\pi i \nu t} d\nu.$$

For a mathematically accessible introduction to this theory, we refer the interested reader to [MBH13].

Throughout the chapter, in an attempt to minimize the level of technical sophistication and expositional complexity, we will work with a fully discrete and finite-dimensional version of (3.9) given by

$$y[t] = \sum_{\tau=0}^{D-1} \sum_{f=0}^{F-1} \widetilde{S}_{\mathcal{L}}(\tau, f) x[t-\tau] e^{2\pi i \frac{f}{F} t}, \tag{3.11}$$

with D, $F \in \mathbb{N}$. In the continuous-time case the size of the spreading function support area plays a critical role as there is a threshold beyond which the system becomes unidentifiable [MBH13]. While we will not dwell on this matter, we simply note that in the setup considered here, the spread is given by $D \cdot F$, i.e., the total number of time-frequency shifts the system induces.

3.2. Universal Realization of Linear Dynamical Systems

In this section, we develop our first central result, a universal realization theorem for linear dynamical systems. This will be effected by building on the spreading decomposition (3.11). Specifically, we first devise—in Lemma 3.2.1—RNNs that realize time shifts, then—in Lemma 3.2.3—RNNs implementing frequency shifts, and finally these building blocks are put together to obtain an RNN that realizes a weighted superposition of time-frequency shifts according to (3.11).

We start with RNNs that realize time shifts. For later reference, we actually construct more general RNNs that implement convolutions, i.e., weighted superpositions of time shifts.

Lemma 3.2.1 (RNNs can realize time shifts and convolutions)**.** *Let $L \in \mathbb{N}$ and $k \in \mathbb{R}^L$. There exists an RNN with input-output relation*

$$y[t] = \sum_{\ell=1}^{L} k_\ell \, x[t - (\ell - 1)], \quad \forall t \geq 0, \tag{3.12}$$

hidden state dimension $L - 1$, and hidden state sequence satisfying

$$h_\ell[t] = x[t - (\ell - 1)], \quad \forall t \geq 0, \ell \in \{1, \dots, L - 1\}. \tag{3.13}$$

Proof. The proof is constructive in the sense of specifying the RNN as a function of the impulse response vector $k \in \mathbb{R}^L$. We start by choosing weight matrices and bias vectors such that (3.13) holds. The basic idea is to design the network such that the past values of $x[\cdot]$ in the hidden state vector h are shifted downward by one position in each time step t, dropping the oldest value at the bottom of the vector and inserting the current value $x[t]$ at the top. To move the values through the non-linear activation function without modifying them, we employ the identity

$$x = \rho(x) - \rho(-x). \tag{3.14}$$

We set

$$A_1 = \begin{pmatrix} \mathbb{I}_L \\ -\mathbb{I}_L \end{pmatrix} \in \mathbb{R}^{2L \times L}, \tag{3.15}$$

$$A_h = \begin{pmatrix} \mathbb{I}_{L-1} & 0_{L-1} & -\mathbb{I}_{L-1} & 0_{L-1} \end{pmatrix} \in \mathbb{R}^{(L-1) \times 2L}, \tag{3.16}$$

$b_1 = 0_{2L}$, and $b_2 = 0_L$.

With these choices, the proof of (3.13) is now effected by induction over t. First, we note that for $h[t]$ in (3.13) to constitute a valid hidden state sequence according to Definition 3.1.1, the initial state needs to satisfy $h[-1] = 0_{L-1}$. This follows directly from $x[t] = 0$, $\forall t < 0$, which is by assumption (3.8), and also constitutes the base case of the induction argument. To

establish the induction step, we assume that (3.13) holds for $t-1$ for some $t \geq 0$, i.e.,

$$h_\ell[t-1] = x[(t-1)-(\ell-1)] = x[t-\ell],$$

and show that—thanks to the choices for A_1, A_h, b_1, and b_2 made above—this implies validity of (3.13) for t. Using (3.15) and $b_1 = 0_{2L}$ in (3.5), one obtains

$$\begin{aligned} g[t] &= \rho\left(A_1 \begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix}\right) = \rho\left(A_1 \begin{pmatrix} x[t] \\ x[t-1] \\ \vdots \\ x[t-(L-1)] \end{pmatrix}\right) \\ &= \begin{pmatrix} \rho(x[t]) & \dots & \rho(x[t-(L-1)]) & \rho(-x[t]) & \dots & \rho(-x[t-(L-1)]) \end{pmatrix}^T. \end{aligned} \tag{3.17}$$

Then, we evaluate (3.4) with A_h from (3.16) and use (3.14) to get

$$h[t] = A_h g[t] = \begin{pmatrix} \rho(x[t]) - \rho(-x[t]) \\ \vdots \\ \rho(x[t-(L-2)]) - \rho(-x[t-(L-2)]) \end{pmatrix} = \begin{pmatrix} x[t] \\ \vdots \\ x[t-(L-2)] \end{pmatrix},$$

or equivalently $h_\ell[t] = x[t-(\ell-1)], \forall \ell \in \{1, \dots, L-1\}$, which establishes the induction step.

It remains to realize the input-output relation (3.12). To this end, we set

$$A_r = \begin{pmatrix} \mathbb{I}_L & -\mathbb{I}_L \end{pmatrix} \in \mathbb{R}^{L \times 2L}, \qquad A_o = k^T \in \mathbb{R}^{1 \times L},$$

and use (3.17), (3.6), (3.7), and (3.14) to conclude that

$$\begin{aligned} r[t] &= A_r g[t] = \begin{pmatrix} \rho(x[t]) - \rho(-x[t]) \\ \vdots \\ \rho(x[t-(L-1)]) - \rho(-x[t-(L-1)]) \end{pmatrix} = \begin{pmatrix} x[t] \\ \vdots \\ x[t-(L-1)] \end{pmatrix}, \\ y[t] &= k^T r[t] = \sum_{\ell=1}^{L} k_\ell \, x[t-(\ell-1)], \end{aligned}$$

which, in turn, completes the proof. □

Remark 3.2.2. *An RNN realizing a time shift by m instants, as needed in* (3.11)*, is now obtained from Lemma 3.2.1 by choosing the impulse response vector $k \in \mathbb{R}^L$ such that it has a* 1 *in the $(m+1)$-th entry and zeros elsewhere.*

The next step in our program is to construct an RNN that realizes frequency shifts by integer multiples of $1/F$, again as needed in (3.11). As this operation corresponds to multiplication of

the input signal by a complex exponential, it produces complex outputs $y[\cdot]$. In slight abuse of Definition 3.1.1, where all weight matrices and bias vectors are real-valued, for ease of exposition, we will here allow complex weights in the output layer, specifically for the quantities A_o and A_r in (3.3). As A_o and A_r do not appear in the state evolution equations (3.4) and (3.5), it is guaranteed that the activation function ρ continues to be applied to real-valued quantities only.

Lemma 3.2.3 (RNNs can realize frequency shifts)**.** *Let $F \in \mathbb{N}$ and $f \in \{0, \dots, F-1\}$. There exists an RNN that realizes the input-output mapping*

$$y[t] = x[t]\, e^{2\pi i \frac{f}{F} t}, \quad \forall t \geq 0. \tag{3.18}$$

Proof. Again the proof is constructive in the sense of specifying the RNN. Throughout the proof, unless explicitly stated otherwise, relations involving t apply for all $t \geq 0$. We start by noting that the function $e^{2\pi i \frac{f}{F} t}$ is F-periodic in $t \in \mathbb{N}$. This F-periodicity motivates the choice of an $(F-1)$-dimensional hidden state sequence $h[t] \in \{0,1\}^{(F-1)}$ encoding the current position within the fundamental period. Specifically, our construction will be seen to ensure

$$h_\ell[t] = \mathbb{1}_{\{((t+1) \bmod F) = \ell\}}, \quad \forall \ell \in \{1, \dots, F-1\}. \tag{3.19}$$

The hidden state vector at time t hence contains a one at position $(t+1) \bmod F$ or equals the all-zeros vector at the end of each period, i.e., when $(t+1) \bmod F = 0$. We will realize (3.19) by appropriate choice of the RNN weight matrices A_1, A_2 and bias vectors b_1, b_2 and the proof will proceed by induction. First, we note that for $h[t]$ in (3.19) to constitute a valid hidden state sequence according to Definition 3.1.1, the initial state needs to satisfy $h[-1] = 0_{F-1}$, which at the same time would constitute the base case $t = -1$ of the induction argument. The relation $h[-1] = 0_{F-1}$ now follows independently of the choices for A_1, A_2, b_1, b_2 and is simply by virtue of the index 0 not being contained in the set $\{1, \dots, F-1\}$ so that the truth function on the RHS of (3.19) yields the all-zeros vector. For the induction step, we assume that (3.19) holds for $t-1$ for some $t \geq 0$, i.e., $h_\ell[t-1] = \mathbb{1}_{\{(t \bmod F) = \ell\}}$, $\forall \ell \in \{1, \dots, F-1\}$. Next, we set

$$A_e := \begin{pmatrix} -1 & -1 & \dots & -1 \\ 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} \in \mathbb{R}^{F \times (F-1)}, \quad b_e := \begin{pmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \in \mathbb{R}^F, \tag{3.20}$$

and define the sequence $e[t] \in \{0,1\}^F$ according to

$$e[t] := A_e h[t-1] + b_e. \tag{3.21}$$

Direct calculation now yields

$$e_\ell[t] = \mathbb{1}_{\{(t \bmod F)+1=\ell\}}, \quad \forall \ell \in \{1, \dots, F\}. \tag{3.22}$$

That is, $e[t]$ indicates its argument t, modulo F to account for F-periodicity of $e^{2\pi i \frac{f}{F} t}$, through a one at the corresponding position in the period and, unlike the state vector, never equals the all-zeros vector. We now use $e[t]$ to construct an indicator function applied to the input signal. To this end, we first recall that RNNs according to Definition 3.1.1 accept input signals in ℓ^∞ and choose a $C \in \mathbb{R}^+$ such that $\|x\|_{\ell^\infty} \leq C$. Next, for all $t \geq 0$, consider the sequence

$$\begin{aligned} \widetilde{x}[t] &= \rho\left(\begin{pmatrix} 1_F & 2CA_e \end{pmatrix} \begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix} + 2Cb_e - C1_F\right) \\ &= \rho\left(1_F\, x[t] + 2Ce[t] - C1_F\right), \end{aligned} \tag{3.23}$$

where we used (3.21). Equivalently, we can express (3.23) as

$$\widetilde{x}_\ell[t] = \rho(x[t] + 2C\mathbb{1}_{\{(t \bmod F)+1=\ell\}} - C) = (x[t] + C)\mathbb{1}_{\{(t \bmod F)+1=\ell\}}, \tag{3.24}$$

for $\ell \in \{1, \dots, F\}$, where we made use of $|x[t]| \leq C$. We proceed to set

$$A_1 = \begin{pmatrix} 1_F & 2CA_e \\ 0_F & A_e \end{pmatrix} \in \mathbb{R}^{(2F)\times F}, \qquad b_1 = \begin{pmatrix} 2Cb_e - C1_F \\ b_e \end{pmatrix} \in \mathbb{R}^{2F},$$

and $b_2 = 0_F$. Inserting into (3.5) yields

$$g[t] = \rho\left(A_1 \begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix} + b_1\right) = \begin{pmatrix} \widetilde{x}[t] \\ e[t] \end{pmatrix},$$

where we employed (3.23) and (3.21), and used the fact that $\rho(e[t]) = e[t]$ as the entries of $e[t]$ equal either 0 or 1. Next, we let

$$\begin{aligned} A_h &= \begin{pmatrix} \mathbb{O}_{F-1} & 0_{F-1} & \mathbb{I}_{F-1} & 0_{F-1} \end{pmatrix}, \\ A_r &= \begin{pmatrix} \mathbb{I}_F & -C\,\mathbb{I}_F \end{pmatrix}, \\ A_o &= \begin{pmatrix} e^{2\pi i \frac{0}{F} f} & e^{2\pi i \frac{1}{F} f} & \dots & e^{2\pi i \frac{F-1}{F} f} \end{pmatrix}. \end{aligned} \tag{3.25}$$

We are now ready to finalize the induction step. From $h[t] = A_h g[t]$ we get

$$h_\ell[t] = e_\ell[t] = \mathbb{1}_{\{(t \bmod F)+1=\ell\}} = \mathbb{1}_{\{((t+1) \bmod F)=\ell\}}, \; \forall \ell \in \{1, \dots, F-1\}, \tag{3.26}$$

where we used that

$$(t \bmod F) + 1 = ((t+1) \bmod F), \tag{3.27}$$

for all t with $(t \bmod F) \neq F-1$. For t such that $(t \bmod F) = F-1$, the LHS of (3.27) equals F while the RHS is equal to 0; as the indices 0 and F do not occur in the set $\{1, \dots, F-1\}$, we trivially have equality between the last two expressions in (3.26). This establishes (3.19) and thereby completes the induction step.

It remains to prove that the input-output relation of the RNN specified along the way is, indeed, given by (3.18). Using (3.25), (3.24), and (3.22) in $r[t] = A_r g[t]$, it follows that

$$\begin{aligned} r_\ell[t] &= \widetilde{x}_\ell[t] - Ce_\ell[t] \\ &= (x[t] + C)\mathbb{1}_{\{(t \bmod F)+1 = \ell\}} - C\mathbb{1}_{\{(t \bmod F)+1 = \ell\}} \\ &= x[t]\mathbb{1}_{\{(t \bmod F)+1 = \ell\}}, \quad \forall \ell \in \{1, \dots, F\}. \end{aligned}$$

The output signal is hence given by

$$\begin{aligned} y[t] &= A_o r[t] \\ &= \sum_{\ell=1}^{F} e^{2\pi i \frac{\ell-1}{F} f} x[t]\, \mathbb{1}_{\{(t \bmod F)+1=\ell\}} \\ &= x[t]\, e^{2\pi i \frac{(t \bmod F)}{F} f} \\ &= x[t]\, e^{2\pi i \frac{t}{F} f}, \end{aligned}$$

where, in the last step, we made use of the F-periodicity of $e^{2\pi i \frac{t}{F} f}$. This completes the proof. □

Having established the RNN realizations of the basic building blocks of the spreading representation (3.11), namely RNNs that realize time shifts (or, more generally, convolutions) and frequency shifts, we proceed to devise RNNs that implement weighted linear combinations of time-frequency shift operators. This entails showing that linear combinations of compositions of time shift RNNs and frequency shift RNNs are again RNNs. As opposed to feedforward networks where compositions and linear combinations trivially preserve the feedforward structure [Elb+21], this is not obvious in the RNN case. The basic idea underlying the construction provided next is hidden-state sharing across component networks, which not only preserves the RNN structure, but also leads to an economical—in terms of the number of nonzero weights—RNN realization.

Lemma 3.2.4 (RNNs can realize LTV systems)**.** *Let $D, F \in \mathbb{N}$ and consider the spreading function $\widetilde{S}_{\mathcal{L}}(\tau, f) \in \mathbb{C}$, $\tau \in \{0, \dots, D-1\}$, $f \in \{0, \dots, F-1\}$. There exists an RNN that realizes the input-output relation*

$$y[t] = \sum_{\tau=0}^{D-1} \sum_{f=0}^{F-1} \widetilde{S}_{\mathcal{L}}(\tau, f) x[t-\tau] e^{2\pi i \frac{f}{F} t}, \quad \forall t \geq 0. \tag{3.28}$$

Proof. There are two main components in the construction of the RNN realizing the desired input-output relation, namely the composition of time shift and frequency shift operators and weighted linear combinations thereof. The latter is easily realized through proper choice of the

output layer weight matrix A_2, whereas the former requires more effort. Specifically, we will design the RNN such that its hidden state vector concatenates the hidden state vectors of the time shift and the frequency shift RNNs in Lemmas 3.2.1 and 3.2.3, respectively, and that this concatenated hidden state vector follows the hidden state evolution equations of the constituent time shift and frequency shift networks. Concretely, our goal will be to design the RNN such that its hidden state vector[1] is given by

$$h[t] = \begin{pmatrix} \dot{h}[t] \\ \ddot{h}[t] \end{pmatrix}, \tag{3.29}$$

where $\dot{h} \in \mathbb{R}^{(D-1)}$ corresponds to the hidden state of the convolution RNN from Lemma 3.2.1 particularized for pure time shifts, and $\ddot{h} \in \{0,1\}^{(F-1)}$ represents the hidden state of the frequency shift RNN in Lemma 3.2.3. The component vector sequences $\dot{h}[t]$ and $\ddot{h}[t]$ now need to follow the state evolution laws in (3.13) and (3.19), respectively, i.e.,

$$\dot{h}_\ell[t] = x[t-(\ell-1)], \quad \forall \ell \in \{1,\dots,D-1\} \tag{3.30}$$

$$\ddot{h}_\ell[t] = \mathbb{1}_{\{((t+1) \bmod F) = \ell\}}, \quad \forall \ell \in \{1,\dots,F-1\}, \tag{3.31}$$

both for all $t \geq 0$. The approach we follow will be as in the proofs of Lemmas 3.2.1 and 3.2.3, namely, we proceed by induction and in the process specify the network weight matrices and bias vectors to make the induction work out. The proof will be finalized by showing how the state vector $h[t]$ following (3.30) and (3.31) leads to the desired overall input-output relation by proper choice of A_2.

The base case $t = -1$ of the induction, i.e., $h[-1] = 0_{D+F-2}$, follows as the base case in Lemma 3.2.1 is by virtue of $x[t] = 0, \forall t < 0$, and that in Lemma 3.2.3 holds as a consequence of the definition of the state vector. Notably, for both components, $\dot{h}_\ell[t]$ and $\ddot{h}_\ell[t]$, the base case follows independently of the choices of the weight matrices and bias vectors. We remark that the base case also establishes that the initial state $h[-1]$ of the hidden state sequence in (3.29) —by virtue of being equal to the all zeros vector—conforms with Definition 3.1.1.

To establish the induction step, we will have to choose A_1, A_2, b_1, and b_2 appropriately.

[1]Note that the symbols $\dot{h}$ and $\ddot{h}$ do not refer to derivatives of h in any form.

Concretely, we start by assuming that (3.30) and (3.31) hold for $t-1$ for some $t \geq 0$, and set

$$A_1 = \begin{array}{c} \\ \scriptstyle DF \\ \\ \\ \scriptstyle 2D \\ \scriptstyle F \end{array} \overset{\overset{D}{\longleftrightarrow}\quad \overset{F-1}{\longleftrightarrow}}{\left(\begin{array}{cccc|c} 1_F & 0_F & \dots & 0_F & 2CA_e \\ 0_F & 1_F & \dots & 0_F & 2CA_e \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_F & 0_F & \dots & 1_F & 2CA_e \\ \hline \multicolumn{4}{c|}{\mathbb{I}_D} & \multirow{2}{*}{$\mathbb{O}$} \\ \multicolumn{4}{c|}{-\mathbb{I}_D} & \\ \hline \multicolumn{4}{c|}{\mathbb{O}} & A_e \end{array}\right)}, \qquad b_1 = \left(\begin{array}{c} 2Cb_e - C1_F \\ 2Cb_e - C1_F \\ \vdots \\ 2Cb_e - C1_F \\ \hline \mathbb{O} \\ \hline b_e \end{array}\right),$$

where A_e, b_e are as defined in (3.20), $C \in \mathbb{R}^+$ is such that $\|x\|_{\ell^\infty} \leq C$, and the unsubscripted $\mathbb{O}$ symbols stand for all zeros matrices of appropriate dimensions. The bias vector b_2 is chosen as $b_2 = 0_{D+F-1}$. The first D columns of A_1 operate on

$$\begin{pmatrix} x[t] \\ \dot{h}[t-1] \end{pmatrix} = \begin{pmatrix} x[t] \\ \vdots \\ x[t-(D-1)] \end{pmatrix}$$

and the last $F-1$ columns multiply $\ddot{h}[t-1]$. Further, A_1 is divided vertically into three parts. The first DF rows produce, for each time shift (including the shift by 0 time instants), a representation akin to (3.23), the middle $2D$ rows correspond to (3.15) in the time shift RNN construction, and the last F rows pertain to the frequency shift RNN, specifically to (3.21). It is hence natural to think of $g[t]$ from (3.5) in three parts according to

$$g[t] = \rho\left(A_1 \begin{pmatrix} x[t] \\ \dot{h}[t-1] \\ \ddot{h}[t-1] \end{pmatrix} + b_1\right) = \begin{array}{c} \\ \scriptstyle DF \\ \\ \\ \scriptstyle 2D \\ \scriptstyle F \end{array} \left(\begin{array}{c} \widetilde{x}[t,0] \\ \widetilde{x}[t,1] \\ \vdots \\ \widetilde{x}[t,D-1] \\ \hline \dot{g}[t] \\ \hline e[t] \end{array}\right), \tag{3.32}$$

where, following the steps leading to (3.24) in Lemma 3.2.3, the vectors $\widetilde{x}[t,\tau]$, $\tau \in \{0,\dots,D-1\}$, are obtained as

$$\widetilde{x}_\ell[t,\tau] = (x[t-\tau]+C)\mathbb{1}_{\{(t \bmod F)+1=\ell\}},$$

$\dot{g}[t]$ equals $g[t]$ in (3.17) with $L=D$, and $e[t]$ is as in (3.22).

We proceed to specify A_h, the submatrix of A_2, which maps to the next hidden state

according to (3.4), as

$$A_h = \begin{array}{c} \\ D-1 \\ F-1 \end{array} \overset{\begin{array}{c} DF \\ \leftrightarrow \end{array}}{\begin{pmatrix} \mathbb{O} & \overset{2D}{\longleftrightarrow} \\ \mathbb{O} & \end{pmatrix}} \overset{\begin{array}{c} \\ \mathbb{I}_{D-1} \quad 0_{D-1} \quad -\mathbb{I}_{D-1} \quad 0_{D-1} \end{array}}{\begin{array}{|cccc|} \hline \multicolumn{4}{|c|}{\mathbb{O}} \\ \hline \multicolumn{4}{|c|}{\mathbb{O}} \\ \hline \end{array}} \overset{\begin{array}{c} F \\ \longleftrightarrow \end{array}}{\begin{array}{|cc|} \hline \multicolumn{2}{|c|}{\mathbb{O}} \\ \hline \mathbb{I}_{F-1} & 0_{F-1} \\ \hline \end{array}}, \tag{3.33}$$

where again the unsubscripted $\mathbb{O}$ symbols refer to all-zeros matrices of appropriate dimensions. Carrying out the state transition for the concatenated hidden state vector (3.29) according to (3.4) with A_h in (3.33) and $g[t]$ in (3.32) yields (3.30) directly and (3.31) upon using the last identity in (3.26).

Next, with the $F \times F$ (unnormalized) DFT matrix $[A_F]_{f,n} = e^{2\pi i \frac{f}{F} n}$, $f \in \{0, \ldots, F-1\}$, $n \in \{0, \ldots, F-1\}$, we define the vectors

$$\widehat{x}[t, \tau] := \begin{pmatrix} A_F & -CA_F \end{pmatrix} \begin{pmatrix} \widetilde{x}[t, \tau] \\ e[t] \end{pmatrix}, \quad \tau \in \{0, \ldots, D-1\},$$

and note, by direct calculation, that

$$\widehat{x}_f[t, \tau] = x[t-\tau] e^{2\pi i \frac{f-1}{F} t}, \quad f \in \{1, \ldots, F\}. \tag{3.34}$$

Now we stack the frequency-shifted versions of $x[t-\tau]$ in (3.34) in the virtual representation sequence $r[t]$ by setting

$$A_r = \begin{array}{c} \\ DF \end{array} \overset{\begin{array}{c} \\ DF \end{array}}{\begin{pmatrix} A_F & \mathbb{O} & \ldots & \mathbb{O} \\ \mathbb{O} & A_F & \ldots & \mathbb{O} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{O} & \mathbb{O} & \ldots & A_F \end{pmatrix}} \overset{\begin{array}{c} 2D \\ \leftrightarrow \end{array}}{\begin{pmatrix} \mathbb{O} \\ \mathbb{O} \\ \vdots \\ \mathbb{O} \end{pmatrix}} \overset{\begin{array}{c} F \\ \longleftrightarrow \end{array}}{\begin{pmatrix} -CA_F \\ -CA_F \\ \vdots \\ -CA_F \end{pmatrix}},$$

which yields

$$r[t] = A_r g[t] = \begin{pmatrix} \widehat{x}[t, 0] \\ \widehat{x}[t, 1] \\ \vdots \\ \widehat{x}[t, D-1] \end{pmatrix}.$$

This finalizes the first part of the construction, namely the composition of time shifts and frequency shifts according to (3.34). We are left with the weighted superposition of the time-frequency-shifted versions of $x[t]$ implementing the input-output relation (3.28). To this end,

we set, for $\tau \in \{0, \dots, D-1\}$,

$$\widetilde{S}_{\mathcal{L}}(\tau, \cdot) := \begin{pmatrix} \widetilde{S}_{\mathcal{L}}(\tau, 0) \\ \widetilde{S}_{\mathcal{L}}(\tau, 1) \\ \vdots \\ \widetilde{S}_{\mathcal{L}}(\tau, F-1) \end{pmatrix} \in \mathbb{C}^F$$

and take

$$A_o = \begin{pmatrix} \widetilde{S}_{\mathcal{L}}(0, \cdot)^T & \widetilde{S}_{\mathcal{L}}(1, \cdot)^T & \dots & \widetilde{S}_{\mathcal{L}}(D-1, \cdot)^T \end{pmatrix},$$

which results in

$$\begin{aligned} y[t] = A_o r[t] &= \sum_{\tau=0}^{D-1} \widetilde{S}_{\mathcal{L}}(\tau, \cdot)^T \widehat{x}[t, \tau] \\ &= \sum_{\tau=0}^{D-1} \sum_{f=0}^{F-1} \widetilde{S}_{\mathcal{L}}(\tau, f) x[t-\tau] e^{2\pi i \frac{f}{F} t}, \end{aligned}$$

as desired. □

We note that the RNN constructed in Lemma 3.2.4 has $\mathcal{O}(DF)$ non-zero weights, i.e., the "size" of the network is proportional to the spread of the system it is to realize. This insight is based on the fact that the virtual representation sequence $r[t]$ is never actually manifested. Hence, by (3.3) only the product $A_o A_r$ has to be stored instead of the (bigger) individual matrices A_o and A_r. Finally, we remark that the magnitudes of the RNN weights in the proof of Lemma 3.2.4 depend on the ℓ^∞-norm of the inputs the RNN accepts.

3.3. Metric Entropy of LTI Systems

Having established that RNNs can universally realize linear dynamical systems with network size proportional to the spread of the system, we proceed to develop a deepened and more quantitative theory along those lines. Specifically, we shall be interested in the approximation of classes of linear dynamical systems to within a prescribed worst-case (within the class) error ϵ through RNNs that can be specified by bitstrings of finite length. Of particular interest will be the scaling behavior of the required length of the bitstring as a function of ϵ and, in particular, whether RNNs can achieve the fundamental limit—over all possible system approximation methods—on this scaling behavior. Answering this question requires the concept of metric entropy of linear systems, a topic originating from control theory [Zam79; ZO93]. The aim of the present section is to introduce this concept, with the presentation geared towards our purposes. We restrict ourselves to LTI systems for conceptual reasons.

A linear dynamical system is time-invariant if the operator kernel $k[t, \tau]$ in (3.10) is a function of τ only, i.e., the input-output relation of the system is given by the convolution of the input

signal $x[\cdot]$ with the impulse response $k[\cdot]$ according to

$$(\mathcal{L}x)[t] = \sum_{\tau=0}^{\infty} k[\tau]x[t-\tau] =: (k * x)[t], \tag{3.35}$$

where we assume that $x[t] \in \mathbb{R}$, $k[t] \in \mathbb{R}$, for $t \in \mathbb{Z}$ and $x[t] = 0$, $k[t] = 0$, for $t < 0$, that is we consider one-sided input signals and causal systems. We shall frequently make use of the one-sided $\mathcal{Z}$-transform for ℓ^∞-signals defined as[2]

$$(\mathcal{Z}\{x[\cdot]\})(z) = \sum_{t=0}^{\infty} x[t]z^t, \quad |z| < 1. \tag{3.36}$$

Note that thanks to $x \in \ell^\infty$ the series (3.36) converges absolutely for all $z \in \mathbb{C}$ with $|z| < 1$. Whenever there is no source of ambiguity, we shall use capital letters to denote the $\mathcal{Z}$-transform according to $X(z) = (\mathcal{Z}\{x[\cdot]\})(z)$. Next, we note the well-known relation

$$(\mathcal{Z}\{(\mathcal{L}x)[\cdot]\})(z) = (\mathcal{Z}\{(k * x)[\cdot]\})(z) = K(z) \cdot X(z), \tag{3.37}$$

where $K(z) := (\mathcal{Z}\{k[\cdot]\})(z)$ is commonly referred to as the system's transfer function.

We proceed to establish the concept of metric entropy of classes of LTI systems largely following [Zam79; ZO93]. On a conceptual level, this complexity notion allows to formulate answers to the following question: *Given a class of LTI systems, how many bits of information do we need to identify a specific system in the class to within a prescribed error?* To formalize matters, we start by defining the metric entropy of general sets.

Definition 3.3.1 ([Wai19])**.** *Let $(\mathcal{X}, \rho)$ be a metric space. An ϵ-covering of a compact set $\mathcal{C} \subseteq \mathcal{X}$ with respect to the metric ρ is a set of points $\{x_1, \ldots, x_N\} \subset \mathcal{C}$ such that for each $x \in \mathcal{C}$, there exists an $i \in [1, N]$ so that $\rho(x, x_i) \leq \epsilon$. The ϵ-covering number $N(\epsilon; \mathcal{C}, \rho)$ is the cardinality of a smallest ϵ-covering of $\mathcal{C}$ and $\mathcal{E}(\epsilon; \mathcal{C}, \rho) := \log_2(N(\epsilon; \mathcal{C}, \rho))$ is the metric entropy of $\mathcal{C}$.*

As LTI systems are uniquely determined by their impulse response, we shall incarnate the concept of "classes of LTI systems" by considering compact sets of impulse responses. More specifically, motivated by [ZO93], we consider systems with exponentially decaying impulse response, that is, for $a, b > 0$, the set of LTI systems characterized by

$$\mathcal{C}(a, b) := \{\mathcal{L} \mid k_{\mathcal{L}}[t] \in \mathbb{R},\ |k_{\mathcal{L}}[t]| \leq ae^{-bt},\ \forall t \in \mathbb{N}_0 \text{ and } k_{\mathcal{L}}[t] = 0,\ \forall t \in \mathbb{Z}\backslash\mathbb{N}_0\}$$

where $k_{\mathcal{L}}[\cdot]$ denotes the impulse response of the system $\mathcal{L}$. The constants b and a quantify the decay behavior of the system memory. Note that the set $\mathcal{C}(a, b)$ encompasses exponentially

[2]Note the positive exponents of z in the definition. This convention is chosen to maintain consistency with Definition 3.3.2 below adopted from [ZO93].

decaying impulse responses of arbitrary decay rate according to

$$\mathcal{C}(a, \log(1/\beta)) = \{\mathcal{L} \mid |k_{\mathcal{L}}[t]| \leq a\beta^t, \forall t \geq 0\}, \quad a > 0,\ \beta \in (0,1). \tag{3.38}$$

Next, we equip the ambient space $\mathcal{X}$ with a suitable metric which quantifies the distance between LTI systems, or equivalently their impulse responses. To this end, we first define Hardy spaces and norms of transfer functions as follows.

Definition 3.3.2 ([Rud87, Chapter 17])**.** *With*

$$\mathcal{K} := \{\mathcal{Z}\{x\} \mid x \in \ell^\infty,\ x[t] \in \mathbb{R},\ \textit{for } t \in \mathbb{N}_0,\ \textit{and } x[t] = 0,\ \textit{for } t < 0\},$$

define for $X \in \mathcal{K}$ the Hardy norms

$$\|X\|_{\mathcal{H}^2} := \sqrt{\sup_{r\in(0,1)} \frac{1}{2\pi} \int_0^{2\pi} |X(re^{i\theta})|^2 d\theta},$$
$$\|X\|_{\mathcal{H}^\infty} := \sup_{|z|<1} |X(z)|.$$

The corresponding Hardy spaces are given by $\mathcal{H}^2 = \{X(\cdot) \mid X \in \mathcal{K},\ \|X\|_{\mathcal{H}^2} < \infty\}$ and $\mathcal{H}^\infty = \{X(\cdot) \mid X \in \mathcal{K},\ \|X\|_{\mathcal{H}^\infty} < \infty\}$.

The distance between the LTI systems $\mathcal{L}$ and $\mathcal{L}'$ with transfer functions $K(z)$ and $K'(z)$, respectively, both in $\mathcal{H}^\infty$, is now defined as

$$\rho(\mathcal{L}, \mathcal{L}') := \|K - K'\|_{\mathcal{H}^\infty}. \tag{3.39}$$

The following result relates $\rho(\mathcal{L}, \mathcal{L}')$ to distance—in terms of squared error—in the system output space.

Theorem 3.3.3. *Let $\mathcal{L}$ and $\mathcal{L}'$ be LTI systems with corresponding transfer functions $K(z)$ and $K'(z)$, both in $\mathcal{H}^\infty$. It holds that*

$$\rho(\mathcal{L}, \mathcal{L}') = \sup_{\|x\|_{\ell^2}=1} \|\mathcal{L}x - \mathcal{L}'x\|_{\ell^2}.$$

Proof. In Appendix 3.B we develop some relevant properties of Hardy Norms, which are summarized in Theorem 3.B.4. Theorem 3.3.3 is a simplification of this result. □

Theorem 3.3.3 shows that identifying a reference system $\mathcal{L}$ to within error $\rho(\mathcal{L}, \mathcal{L}') = \epsilon$ guarantees that the estimated system $\mathcal{L}'$ results in output signals that deviate no more than ϵ—in ℓ^2-norm—from the output that would be produced by the reference system $\mathcal{L}$.

We are now ready to state a central result which quantifies the metric entropy of $\mathcal{C}(a, b)$ with respect to the distance measure $\rho(\mathcal{L}, \mathcal{L}')$.

Theorem 3.3.4. *Let $a, b > 0$ and consider the set*

$$\mathcal{C}(a,b) = \{\mathcal{L} \mid k_{\mathcal{L}}[t] \in \mathbb{R},\ |k_{\mathcal{L}}[t]| \leq ae^{-bt},\ \forall t \in \mathbb{N}_0 \text{ and } k_{\mathcal{L}}[t] = 0,\ \forall t \in \mathbb{Z}\backslash\mathbb{N}_0\}.$$

The metric entropy of $\mathcal{C}(a,b)$ with respect to the metric

$$\rho(\mathcal{L}, \mathcal{L}') = \|K - K'\|_{\mathcal{H}^\infty}$$

satisfies

$$\mathcal{E}(\epsilon; \mathcal{C}(a,b), \rho) \sim \frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2, \tag{3.40}$$

where $\gamma := \log_2(e)$.

Proof. This result is due to [Zam79; ZO93]. However, we present a self-contained proof in Appendix 3.C specific to our setting. □

We note that the metric entropy in (3.40) scaling according to $(\log(1/\epsilon))^2$ shows that the set $\mathcal{C}(a,b)$ is not overly massive. Richer function classes such as the set of all Lipschitz functions from $[0,1]^d$ to $\mathbb{R}$ with a given Lipschitz constant have metric entropy scaling according to ϵ^{-d} [Wai19]. Moreover, it follows from (3.40) that impulse responses of slower (exponential) decay, i.e., with smaller b, are more complex to describe.

3.4. Optimal Covering Through Quantized RNNs

We are now in a position to state the second central result of this chapter. Specifically, we show that RNNs with suitably quantized weights provide an optimal—in the sense of Theorem 3.3.4—ϵ-covering of $\mathcal{C}(a,b)$ with respect to the metric $\rho(\mathcal{L}, \mathcal{L}')$. Operationally, this means that RNNs can optimally—in the sense of metric entropy—learn (or identify) the class of LTI systems with exponentially decaying impulse response. This result quantifies what is possible in principle and thereby provides a benchmark against which practical learning algorithms can be assessed.

The results presented so far apply to RNNs with real-valued weights. Constructing an optimal covering through RNNs requires, however, encoding of the approximating RNNs into bitstrings of length scaling in the approximation error ϵ according to (3.40). Now, there are two components that go into such an encoding of RNNs, namely the values of the nonzero weights in the matrices A_1, A_2 and the vectors b_1, b_2 and the locations of these weights, i.e., the topology of the network. The former requires quantization of the weights at a resolution that scales adequately in ϵ. We shall see below that encoding the topology is a non-issue. The main technical problem hence resides in ensuring that weight quantization in the approximating RNN can be effected at a resolution that allows metric entropy optimality—in terms of the covering realized—and at the same time guarantees that the resulting error incurred at the system output consorts with the desired approximation accuracy.

We start by defining an RNN weight quantization scheme.

Definition 3.4.1 (Quantized weights)**.** *For $\delta > 0$, define the set*

$$\mathbb{S}_\delta := \{\delta k \mid k \in \mathbb{Z}\}.$$

We say that an RNN has δ-quantized weights if all its weights are in $\mathbb{S}_\delta \cup \{-1, 1\}$. Further, define the quantization function

$$S_\delta(w) := sign(w)\left\lfloor \frac{|w|}{\delta} \right\rfloor \delta, \quad w \in \mathbb{R}.$$

Clearly, we have $|S_\delta(w) - w| \leq \delta$ and $|S_\delta(w)| \leq |w|$.

The main idea underlying the proof of the optimal RNN covering result builds on the approximation of the exponentially decaying impulse responses in $\mathcal{C}(a, b)$ through finite impulse response (FIR) filters of suitable length and with suitably quantized impulse response coefficients. In order to quantify the approximation error—in terms of $\rho(\mathcal{L}, \mathcal{L}') = \|K - K'\|_{\mathcal{H}^\infty}$—resulting from this truncation and coefficient quantization, we will need the following simple technical result.

Lemma 3.4.2. *Consider the LTI systems with impulse responses $k[\cdot]$ and $\widetilde{k}[\cdot]$ and corresponding transfer functions $K(z)$ and $\widetilde{K}(z)$, both in $\mathcal{H}^\infty$. We have*

$$\|K(\cdot) - \widetilde{K}(\cdot)\|_{\mathcal{H}^\infty} \leq \sum_{t=0}^{\infty} |k[t] - \widetilde{k}[t]|.$$

Proof. The proof is by the following chain of relations

$$\begin{aligned}
\|K(\cdot) - \widetilde{K}(\cdot)\|_{\mathcal{H}^\infty} &= \sup_{|z|<1} \left| \sum_{t=0}^{\infty} k[t] z^t - \sum_{t=0}^{\infty} \widetilde{k}[t] z^t \right| \\
&= \sup_{|z|<1} \left| \sum_{t=0}^{\infty} (k[t] - \widetilde{k}[t]) z^t \right| \\
&\leq \sup_{|z|<1} \sum_{t=0}^{\infty} |k[t] - \widetilde{k}[t]||z|^t \\
&= \sum_{t=0}^{\infty} |k[t] - \widetilde{k}[t]|.
\end{aligned}$$

□

We are now ready to state the main result.

Theorem 3.4.3 (RNNs are metric-entropy-optimal)**.** *Consider an LTI system $\mathcal{L}$ with impulse response satisfying $|k[t]| \leq a\, e^{-bt}$, $\forall t \geq 0$, for some $a, b > 0$, and corresponding transfer function $K(z)$. For every $\epsilon > 0$, with*

$$M := \left\lceil \frac{1}{b} \log\left(\frac{a}{\epsilon}\right) + \frac{1}{b} \log\left(\frac{2}{1-e^{-b}}\right) \right\rceil,$$

$\mathcal{L}$ can be approximated by a $\delta := \frac{\epsilon}{2M}$-quantized RNN $\widetilde{\mathcal{R}}$—of hidden state size $M-1$—realizing an FIR filter with transfer function $\widetilde{K}(z)$ such that

$$\|K(\cdot) - \widetilde{K}(\cdot)\|_{\mathcal{H}^\infty} \leq \epsilon.$$

Moreover, $\widetilde{\mathcal{R}}$ can be encoded in a uniquely decodable fashion, provided that both encoder and decoder know a and b, using no more than

$$\frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2 + o\left(\left(\log\left(\frac{1}{\epsilon}\right)\right)^2\right)$$

bits, where $\gamma := \log_2(e)$.

Proof. The idea of the proof is to δ-quantize the suitably truncated impulse response $k[t]$ corresponding to $\mathcal{L}$, which is then realized (exactly) by an RNN, denoted as $\widetilde{\mathcal{R}}$, following the construction in Lemma 3.2.1. Concretely, we choose the truncated quantized impulse response according to $\widetilde{k}[t] := S_\delta(k[t])\mathbb{1}_{\{t \leq (M-1)\}}$, denote the corresponding transfer function by $\widetilde{K}(z)$, and then use Lemma 3.4.2 to bound

$$\begin{aligned}
\|K(\cdot) - \widetilde{K}(\cdot)\|_{\mathcal{H}^\infty} &\leq \sum_{t=0}^{\infty} |k[t] - \widetilde{k}[t]| \\
&= \sum_{t=0}^{M-1} |k[t] - \widetilde{k}[t]| + \sum_{t=M}^{\infty} |k[t]| \\
&\leq M\delta + \sum_{t=M}^{\infty} a e^{-bt} \\
&= M\delta + a\,\frac{e^{-bM}}{1-e^{-b}} \qquad (3.41) \\
&\leq M\delta + a\,\frac{e^{-\log\left(\frac{2a}{\epsilon(1-e^{-b})}\right)}}{1-e^{-b}} \\
&= M\delta + a\,\frac{\frac{\epsilon(1-e^{-b})}{2a}}{1-e^{-b}} \\
&= M\frac{\epsilon}{2M} + \frac{\epsilon}{2} = \epsilon,
\end{aligned}$$

where in (3.41) we used $\sum_{n=M}^{\infty} r^n = \frac{r^M}{1-r}$, for $|r| < 1$.

It remains to establish that the RNN realizing the FIR system with impulse response $\widetilde{k}[t]$ can be encoded in a uniquely decodable fashion into a bitstring of length consorting with covering optimality according to (3.40). As mentioned earlier, encoding an RNN in a bitstring requires specifying its topology and quantized weights, both in binary form. We first convince ourselves that the topology of the RNN realizing $\widetilde{k}[t]$ is fixed and hence does not need to be encoded. This follows by recognizing that in the RNN construction in the proof of Lemma 3.2.1 the quantities A_1, A_h, A_r, b_1, and b_2 are all independent of the impulse response of the FIR system to be realized and only A_o depends on the impulse response according to $A_o = \tilde{k}^T$. The locations of the nonzero entries in the weight matrices and bias vectors of the approximating RNN hence need not be encoded. This leaves us with having to represent the M quantized impulse response coefficients $\widetilde{k}[t]$ through a bitstring of length scaling in ϵ such that covering optimality is attained. To this end, we first note that from Definition 3.4.1, we get

$$|\widetilde{k}[t]| = |S_\delta(k[t])| \leq |k[t]| \leq ae^{-bt}, \qquad \forall t \in \{0, \ldots, M-1\}.$$

The quantized impulse response coefficients hence satisfy

$$\widetilde{k}[t] \in \mathbb{S}_\delta \cap [-ae^{-bt}, ae^{-bt}], \qquad \forall t \in \{0, \ldots, M-1\},$$

and can therefore be stored using at most $\left\lceil \log_2 \left(\frac{ae^{-bt}}{\delta} \right) \right\rceil + 1$ bits. As a and b are known to the encoder and the decoder by assumption, we can encode the quantized impulse response coefficients into a uniquely decodable bitstring simply by allocating $\left\lceil \log_2 \left(\frac{ae^{-bt}}{\delta} \right) \right\rceil + 1$ bits to each coefficient, concatenating the corresponding binary labels (filled up with zeros if they are of smaller than the alloted length) and have the decoder read out the labels sequentially to deliver the corresponding points in $\mathbb{S}_\delta$.

It remains to establish that the length of the bitstring just constructed conforms with (3.40). To this end, we first upper-bound the length of the bitstring according to

$$\begin{aligned}
&\sum_{t=0}^{M-1}\left(\left\lceil \log_2\left(\frac{ae^{-bt}}{\delta}\right)\right\rceil + 1\right) \\
&\quad \leq \sum_{t=0}^{M-1}\left(\log_2\left(\frac{a}{\delta}\right) + \log_2(e^{-bt}) + 2\right) \\
&\quad = 2M + M\gamma\log\left(\frac{a}{\delta}\right) + \sum_{t=0}^{M-1}\gamma\log\left(e^{-bt}\right) \\
&\quad = 2M + M\gamma\log\left(\frac{a}{\delta}\right) - b\gamma\sum_{t=0}^{M-1} t \\
&\quad = 2M + M\gamma\log\left(\frac{2aM}{\epsilon}\right) - b\gamma\,\frac{M(M-1)}{2} \qquad (3.42) \\
&\quad = M\left(\gamma\log\left(\frac{2aM}{\epsilon}\right) - b\gamma\,\frac{(M-1)}{2} + 2\right) \\
&\quad = M\left(\gamma\log\left(\frac{a}{\epsilon}\right) - M\frac{b\gamma}{2} + \gamma\log(M) + \frac{b\gamma}{2} + 3\right), \qquad (3.43)
\end{aligned}$$

where we used $\log_2(x) = \gamma\log(x)$ with $\gamma := \log_2(e)$ and in (3.42) we employed $\delta = \frac{\epsilon}{2M}$. Next, we note from the definition of M that

$$\frac{1}{b}\log\left(\frac{a}{\epsilon}\right) + K_1(b) \leq M \leq \frac{1}{b}\log\left(\frac{a}{\epsilon}\right) + K_1(b) + 1, \qquad (3.44)$$

with $K_1(b) := \frac{1}{b}\log\left(\frac{2}{1-e^{-b}}\right)$. Using (3.44) in (3.43) allows us to further upper-bound (3.43) as follows:

$$
\begin{aligned}
&M\left(\gamma\log\left(\frac{a}{\epsilon}\right)-M\frac{b\gamma}{2}+\gamma\log(M)+\frac{b\gamma}{2}+3\right)\\
&\leq M\left(\gamma\log\left(\frac{a}{\epsilon}\right)-\frac{1}{b}\log\left(\frac{a}{\epsilon}\right)\frac{b\gamma}{2}-K_1(b)\frac{b\gamma}{2}+\gamma\log(M)+\frac{b\gamma}{2}+3\right)\\
&=M\left(\frac{\gamma}{2}\log\left(\frac{a}{\epsilon}\right)+\gamma\log(M)+K_2(b)\right)\\
&\leq\left(\frac{1}{b}\log\left(\frac{a}{\epsilon}\right)+K_1(b)+1\right)\left(\frac{\gamma}{2}\log\left(\frac{a}{\epsilon}\right)+\gamma\log(M)+K_2(b)\right)\\
&=\frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2+\frac{\gamma}{b}\log\left(\frac{a}{\epsilon}\right)\log(M)+\frac{1}{b}\log\left(\frac{a}{\epsilon}\right)K_2(b)\\
&\quad+(K_1(b)+1)\frac{\gamma}{2}\log\left(\frac{a}{\epsilon}\right)+(K_1(b)+1)\gamma\log(M)+(K_1(b)+1)K_2(b)\\
&=\frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2+\frac{\gamma}{b}\log(M)\log\left(\frac{1}{\epsilon}\right)+K_3(b)\log\left(\frac{1}{\epsilon}\right)\\
&\quad+K_4(a,b)\log(M)+K_5(a,b)\\
&\leq\frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2+o\left(\left(\log\left(\frac{1}{\epsilon}\right)\right)^2\right),
\end{aligned}
$$

where $K_2(b):=-K_1(b)\frac{b\gamma}{2}+\frac{b\gamma}{2}+3$, $K_3(b):=\frac{K_2(b)}{b}+\frac{(K_1(b)+1)\gamma}{2}$, $K_4(a,b):=\gamma(K_1(b)+1)+\log(a)\frac{\gamma}{b}$, and $K_5(a,b):=(K_1(b)+1)K_2(b)+K_3(b)\log(a)$. The last inequality follows from $\gamma\leq 2$ and $\log(M)=o(\log(\epsilon^{-1}))$. □

We conclude by noting that the dependence of the hidden state size and the weight quantization resolution of the approximating RNN in Theorem 3.4.3 on the parameters a, b, ϵ reflects that more complex sets $\mathcal{C}(a,b)$ and smaller target approximation error require larger hidden state size and higher quantization resolution.

3.5. Metric-Entropy-Optimal Learning of Linear Difference Equations

Over the last few years a significant body of literature on deep neural network learning of the solutions of parametric PDEs was developed [Gro+18; BGJ20; Ras21]. More specifically, this line of work is concerned with learning the map taking the right-hand side of the PDE and its parameters to the solution. We next suggest an alternative viewpoint in its simplest possible mathematical incarnation, namely that of learning differential, in fact difference, equations themselves. From a practical perspective this amounts to identifying the dynamics of physical, biological, mechanical, or chemical processes from observed input-output traces [Wan+18].

We consider linear difference equations with constant coefficients given by

$$\sum_{j=0}^{P} b_j y[t-j] = \sum_{i=0}^{Q} a_i x[t-i], \tag{3.45}$$

where $P, Q \in \mathbb{N}, b_j, a_i \in \mathbb{R}$, and $x[t]$ and $y[t]$ designate the input and the output, respectively, of the dynamical system characterized by the difference equation. Difference equations of the form (3.45) correspond to LTI systems with rational transfer functions. Concretely application of Lemma 3.B.1 yields $Y(z) = K(z)X(z)$ with

$$K(z) = \frac{\sum_{i=0}^{Q} a_i z^i}{\sum_{j=0}^{P} b_j z^j}. \tag{3.46}$$

Learning of the difference equation (3.45) from input-output traces, i.e., determining the coefficients a_i and b_j in (3.45) based on the outputs $y[\cdot]$ corresponding to given inputs $x[\cdot]$ hence amounts to identifying the LTI system with transfer function (3.46). We first convince ourselves that RNNs can, in principle, realize systems with rational transfer functions, thereby extending Lemma 3.2.1 where this was shown for polynomial transfer functions.

Theorem 3.5.1 (RNNs can realize all rational transfer functions)**.** *Let $\mathcal{L}$ be an LTI system with transfer function*

$$K(z) = \frac{\sum_{i=0}^{Q} a_i z^i}{\sum_{j=0}^{P} b_j z^j},$$

where $Q, P \in \mathbb{N}$ and $a_i, b_j \in \mathbb{R}$ with $b_0 \neq 0$. Then, there exists an RNN that realizes $\mathcal{L}$ exactly.

Proof. The proof will be effected by constructing the RNN realizing $\mathcal{L}$. We start by noting that application of the inverse $\mathcal{Z}$-transform and the time shift property Lemma 3.B.1 to

$$Y(z) = X(z)K(z)$$

yields the difference equation

$$y[t] = \sum_{i=0}^{Q} c_i x[t-i] + \sum_{j=1}^{P} d_j y[t-j], \tag{3.47}$$

with $c_i = \frac{a_i}{b_0}$ and $d_j = -\frac{b_j}{b_0}$. In contrast to the construction in Lemma 3.2.1 which realizes a forward part only, here given by $\sum_{i=0}^{Q} c_i x[t-i]$, we will need to account for both the forward part and the backward part $\sum_{j=1}^{P} d_j y[t-j]$. This will be accomplished by choosing the RNN weight matrices A_1, A_2 and bias vectors b_1, b_2 such that the hidden state vector $h[t]$ contains the

last Q values of the input signal $x[\cdot]$ and the last P values of the output signal $y[\cdot]$ according to

$$h[t] = \begin{matrix} \updownarrow Q \\ \\ \updownarrow P \end{matrix} \left(\begin{array}{c} x[t] \\ \vdots \\ x[t-(Q-1)] \\ \hline y[t] \\ \vdots \\ y[t-(P-1)] \end{array}\right). \tag{3.48}$$

Then, based on (3.48), we establish that these choices also yield the output signal as desired. We commence by specifying the RNN weights and proving (3.48) by induction. By slight abuse of notation, we let the vector c have a zero-th entry and define

$$c := \begin{pmatrix} c_0 & c_1 & \dots & c_Q \end{pmatrix}^T \in \mathbb{R}^{Q+1},$$
$$d := \begin{pmatrix} d_1 & \dots & d_P \end{pmatrix}^T \in \mathbb{R}^{P},$$

and the matrix

$$W := \begin{matrix} \\ \updownarrow 1 \\ \updownarrow Q \\ \updownarrow 1 \\ \updownarrow P-1 \end{matrix} \overset{\overset{Q+1}{\longleftrightarrow}}{\left(\begin{array}{cc|c} c^T & & d^T \\ \mathbb{I}_Q & 0_Q & \mathbb{O} \\ c^T & & d^T \\ \mathbb{O} & & \mathbb{I}_{P-1} \quad 0_{P-1} \end{array}\right)} \overset{P}{\longleftrightarrow}, \tag{3.49}$$

where the unsubscripted symbols $\mathbb{O}$ stand for all-zeros matrices of appropriate dimensions. The network weights are now chosen according to

$$A_1 = \begin{pmatrix} \mathbb{I}_{P+Q+1} \\ -\mathbb{I}_{P+Q+1} \end{pmatrix}, \qquad A_2 = W \begin{pmatrix} \mathbb{I}_{P+Q+1} & -\mathbb{I}_{P+Q+1} \end{pmatrix},$$

and $b_1 = 0_{2P+2Q+2}$, $b_2 = 0_{P+Q+1}$. With (3.1) and thanks to (3.14), this yields

$$\begin{pmatrix} y[t] \\ h[t] \end{pmatrix} = W \begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix}, \quad \forall t \geq 0. \tag{3.50}$$

We are now ready to establish (3.48) by induction. First, we note that for $h[t]$ in (3.48) to constitute a valid hidden state sequence according to Definition 3.1.1, the initial state needs to satisfy $h[-1] = 0_{Q+P}$. This, indeed, follows from the assumption $x[t] = y[t] = 0$, $\forall t < 0$, and, in turn, also yields the base case $t = -1$ of the induction argument. To establish the induction step, we assume that (3.48) holds for $t-1$ for some $t \geq 0$. Next, let $h_{1:Q}[t] \in \mathbb{R}^Q$ denote the subvector of $h[t]$ containing the entries 1 through Q. It now follows from (3.50) and (3.49) that

$$h_{1:Q}[t] = \begin{pmatrix} \mathbb{I}_Q & 0_Q \end{pmatrix} \begin{pmatrix} x[t] \\ x[t-1] \\ \vdots \\ x[t-Q] \end{pmatrix} = \begin{pmatrix} x[t] \\ \vdots \\ x[t-(Q-1)] \end{pmatrix}$$

and

$$\begin{aligned} h_{Q+1}[t] &= \begin{pmatrix} c^T & d^T \end{pmatrix} \begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix} \\ &= \begin{pmatrix} c^T & d^T \end{pmatrix} \begin{pmatrix} x[t] \\ \vdots \\ x[t-Q] \\ y[t-1] \\ \vdots \\ y[t-P] \end{pmatrix} \\ &= \sum_{i=0}^{Q} c_i x[t-i] + \sum_{j=1}^{P} d_j y[t-j] = y[t], \end{aligned} \tag{3.51}$$

where we used (3.47). The proof of the induction step is now completed upon noting that

$$\begin{aligned} h_{(Q+2):(Q+P)}[t] &= W \begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix} \\ &= \begin{pmatrix} \mathbb{I}_{P-1} & 0_{P-1} \end{pmatrix} \begin{pmatrix} y[t-1] \\ \vdots \\ y[t-P] \end{pmatrix} \\ &= \begin{pmatrix} y[t-1] \\ \vdots \\ y[t-(P-1)] \end{pmatrix}. \end{aligned}$$

Finally, it follows by combining (3.50), (3.49), and (3.51) that the weights we chose yield the desired output signal. □

We have hence established that RNNs with real-valued weights can realize LTI systems with rational transfer functions exactly. In fact, as inspection of the weight matrices A_1, A_2 and the bias vectors b_1, b_2 in the proof of Theorem 3.5.1 reveals, the size of the RNN is $\mathcal{O}(P+Q)$ and hence proportional to the number of parameters in the system transfer function.

We now proceed to argue that the results established in Section 3.4 provide a fundamental limit on how well difference equations of the form (3.45) can be learned in principle and that RNNs can achieve this fundamental limit. But first, we state an important restriction, namely

to LTI systems (of rational transfer function) that have corresponding impulse responses in ℓ_1. In system theory parlance such systems are often referred to as stable [Kai80, Section 2.6]. If the coefficients of $K(z)$ in (3.45) are such that the system is, indeed, stable, the impulse response is necessarily a linear combination of terms of the form $p(t)\cos(\theta t+\omega)\beta^t$, where $p(t)$ is a polynomial in t, $\beta \in (0,1)$, and $\theta, \omega \in \mathbb{R}$ [OSB99]. Denoting the largest β occurring in this linear combination by $\widetilde{\beta}$, this class of impulse responses is contained in the set $\mathcal{C}(a', \log(1/\beta'))$ with $\beta' > \widetilde{\beta}$ and a' chosen suitably, where such a $\beta' \in (0,1)$ always exists thanks to the set $(0,1)$ being open and a' exists as $a > 0$ in (3.38). Application of [Rud87, Theorem 13.6] shows that the covering number of the set of stable rational transfer functions equals that of $\mathcal{C}(a', \log(1/\beta'))$. Hence, Theorem 3.4.3 allows us to conclude that RNNs can, in principle, learn difference equations of the form (3.45) with coefficients a_i, b_j such that the corresponding LTI system is stable in a metric-entropy-optimal manner.

3.6. Conclusion

The setting in this chapter was deliberately chosen so as to allow the minimum level of mathematical sophistication needed to bring out the main conceptual findings. Numerous extensions abound, such as the continuous-time case and the approximation of nonlinear systems. It would furthermore be interesting to understand how metric entropy results can be obtained for linear time-varying systems. This would possibly allow to establish RNN covering optimality for general linear dynamical systems. From a control theory perspective our findings state that RNNs can be trained to optimally—in the sense of metric entropy—identify LTI systems. Here, it would be interesting to understand whether the presence of feedback, which is known to reduce identification complexity, could be incorporated into our theory and whether the corresponding fundamental limits can again be shown to be achievable through identification by RNNs. Furthermore, we consider it worthwhile to investigate how concepts such as controllability, reachability, and observability for linear dynamical systems transfer to the state-space representation of RNNs realizing these systems. An issue we have not touched upon at all is that of algorithms for learning the weights of approximating RNNs and whether such algorithms are likely to find the RNN constructions we exhibit. Another important aspect we did not discuss is that of minimality of linear dynamical system realizations [Kai80] and how it relates to corresponding RNN realizations [Son98]. A question cast in the same mould is that of uniqueness of neural network realizations, a large field of research, both in feedforward as well as recurrent neural network theory [Fef94; VB21b; VB21a; AS93b; ASM93; AS93a]. Finally, we find that extensions of the ideas in Section 3.5 to linear, nonlinear, partial, and stochastic differential equations constitute a worthwhile endeavor. In this regard, we mention that the universal realization result Lemma 3.2.4, in its continuous-time incarnation, suggests that pseudo-differential operators [Grö01, Chapter 14] can be represented exactly by (continuous-time) RNNs.

APPENDICES

3.A. Alternative Definitions of RNNs

Definition 3.A.1 (Elman RNN). *[Elm90; GBC16] For $\mathring{m} \in \mathbb{N}$, weights $\mathring{U} \in \mathbb{R}^{\mathring{m}\times 1}, \mathring{W}_1 \in \mathbb{R}^{\mathring{m}\times\mathring{m}}$, $\mathring{W}_2 \in \mathbb{R}^{1\times\mathring{m}}$, and biases $\mathring{b}_1 \in \mathbb{R}^{\mathring{m}}$, $\mathring{b}_2 \in \mathbb{R}$, the Elman RNN with hidden state sequence $\mathring{h}[t] \in \mathbb{R}^{\mathring{m}}$ of initial state $\mathring{h}[-1] = 0_{\mathring{m}}$ and output $y[t] \in \mathbb{R}$, for all $t \geq 0$, is defined by*

$$\mathring{h}[t] = \rho(\mathring{U}x[t] + \mathring{W}_1\mathring{h}[t-1] + \mathring{b}_1) \tag{3.52}$$

$$y[t] = \mathring{W}_2\mathring{h}[t] + \mathring{b}_2. \tag{3.53}$$

Lemma 3.A.2. *The input-output relation of every RNN according to Definition 3.1.1 can equivalently be realized by an Elman RNN.*

Proof. Given an RNN according to Definition 3.1.1 with weight matrices A_1, A_2 and bias vectors b_1, b_2, we construct an Elman RNN that realizes the same input-output map. First, set

$$A_1 = \begin{pmatrix} A_x & A_g \end{pmatrix}, \qquad A_2 = \begin{pmatrix} A_y \\ A_h \end{pmatrix}, \qquad b_1 = b_g, \qquad b_2 = \begin{pmatrix} b_y \\ b_h \end{pmatrix},$$

with $A_x \in \mathbb{R}^{n\times 1}$, $A_g \in \mathbb{R}^{n\times m}$, $A_y \in \mathbb{R}^{1\times n}$, $A_h \in \mathbb{R}^{m\times n}$, $b_g \in \mathbb{R}^n$, $b_y \in \mathbb{R}$, and $b_h \in \mathbb{R}^m$. With these definitions, (3.1) and (3.2) can be written as

$$\begin{pmatrix} y[t] \\ h[t] \end{pmatrix} = \begin{pmatrix} A_y \\ A_h \end{pmatrix} g[t] + \begin{pmatrix} b_y \\ b_h \end{pmatrix}, \tag{3.54}$$

where

$$g[t] = \rho\left(\begin{pmatrix} A_x & A_g \end{pmatrix} \begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix} + b_g\right).$$

The equivalent—in the sense of input-output relation—Elman RNN is now obtained by setting $\mathring{m} = n$ and

$$\begin{gathered} \mathring{W}_1 = A_gA_h, \qquad \mathring{U} = A_x, \qquad \mathring{b}_1 = A_gb_h + b_g, \\ \mathring{W}_2 = A_y, \qquad \mathring{b}_2 = b_y. \end{gathered} \tag{3.55}$$

We first establish, by induction, that these choices lead to the hidden state sequences of the original RNN and the equivalent Elman RNN to be related according to $h[t] = A_h\mathring{h}[t] + b_h$, for all

$t \geq 0$. The base case follows by choosing the initial hidden state $\mathring{h}[-1]$ of the Elman RNN such that $h[-1] = 0_m = A_h\mathring{h}[-1] + b_h$. If $b_h = 0$, which is the case for all RNN constructions in this chapter, one can, indeed, simply set $\mathring{h}[-1] = 0$. Next, we assume that $h[t-1] = A_h\mathring{h}[t-1] + b_h$, for some $t \geq 0$, and insert (3.55) into (3.52) to obtain

$$\begin{aligned}\mathring{h}[t] &= \rho(A_x x[t] + A_g A_h \mathring{h}[t-1] + A_g b_h + b_g) \\ &= \rho(A_x x[t] + A_g(A_h\mathring{h}[t-1] + b_h) + b_g) \\ &= \rho(A_x x[t] + A_g h[t-1] + b_g) \\ &= g[t].\end{aligned}$$

Using $\mathring{h}[t] = g[t]$ in (3.54) then yields $h[t] = A_h\mathring{h}[t] + b_h$ as desired. This completes the proof of the induction step. The input-output relation of the Elman RNN is seen to equal that of the original RNN—given by (3.54) as $y[t] = A_y g[t] + b_y$ —upon inserting $\mathring{h}[t] = g[t]$, $\mathring{W}_2 = A_y$, and $\mathring{b}_2 = b_y$ in (3.53). □

3.B. Properties of the $\mathcal{Z}$-transform and of Hardy Norms

Lemma 3.B.1. *Let $x[t]$ be a one-sided sequence, i.e., $x[t] = 0$, for $t < 0$. Then, for $k \in \mathbb{N}$, it holds that*

$$(\mathcal{Z}\{x[\cdot - k]\})(z) = z^k(\mathcal{Z}\{x[\cdot]\})(z).$$

Proof. We have

$$\begin{aligned}(\mathcal{Z}\{x[\cdot - k]\})(z) &= \sum_{t=0}^{\infty} x[t-k]z^t = \sum_{t=-k}^{\infty} x[t]z^{t+k} \\ &= z^k \sum_{t=0}^{\infty} x[t]z^t = z^k(\mathcal{Z}\{x[\cdot]\})(z),\end{aligned}$$

where we used that $x[t]$ is one-sided. □

Theorem 3.B.2. *Let $x \in \ell^2$ be a one-sided sequence, i.e., $x[t] = 0$, for $t < 0$. Then, we have*

$$\|X\|_{\mathcal{H}^2} = \|x\|_{\ell^2},$$

where $X = \mathcal{Z}\{x\}$.

Proof.

$$\begin{aligned}\|X\|_{\mathcal{H}^2}^2 &= \sup_{r\in(0,1)} \frac{1}{2\pi}\int_0^{2\pi} |X(re^{i\theta})|^2 d\theta \\ &= \sup_{r\in(0,1)} \frac{1}{2\pi}\int_0^{2\pi} \left|\sum_{t=0}^{\infty} x[t](re^{i\theta})^t\right|^2 d\theta \\ &= \sup_{r\in(0,1)} \frac{1}{2\pi}\int_0^{2\pi} \left(\sum_{t=0}^{\infty} x[t](re^{i\theta})^t\right)\overline{\left(\sum_{t'=0}^{\infty} x[t'](re^{i\theta})^{t'}\right)} d\theta \\ &= \sup_{r\in(0,1)} \frac{1}{2\pi}\int_0^{2\pi} \sum_{t=0}^{\infty}\sum_{t'=0}^{\infty} x[t]r^t\,\overline{x[t']}r^{t'}\,e^{i\theta(t-t')} d\theta \\ &= \sup_{r\in(0,1)} \sum_{t=0}^{\infty}\sum_{t'=0}^{\infty} x[t]\overline{x[t']}\,r^{t+t'}\frac{1}{2\pi}\int_0^{2\pi} e^{i\theta(t-t')} d\theta \qquad (3.56) \\ &= \sup_{r\in(0,1)} \sum_{t=0}^{\infty}\sum_{t'=0}^{\infty} x[t]\overline{x[t']}\,r^{t+t'}\mathbb{1}_{\{t=t'\}} \\ &= \sup_{r\in(0,1)} \sum_{t=0}^{\infty} |x[t]|^2\,r^{2t} \\ &= \sum_{t=0}^{\infty} |x[t]|^2 = \|x\|_{\ell^2}^2.\end{aligned}$$

□

In (3.56), we interchanged the order of integration and summation. This step can be justified using the Fubini–Tonelli theorem as detailed in [Lap17, Section 2.6] and [Bar95, Exercise 10.J], since, $\forall r \in (0,1)$,

$$\begin{aligned}\frac{1}{2\pi}\int_0^{2\pi}\sum_{t=0}^{\infty}\sum_{t'=0}^{\infty}\left|x[t]\,r^t\,\overline{x[t']}\,r^{t'}e^{i\theta(t-t')}\right|d\theta &= \frac{1}{2\pi}\int_0^{2\pi}\left(\sum_{t=0}^{\infty}|x[t]|r^t\right)\left(\sum_{t'=0}^{\infty}|x[t']|r^{t'}\right)d\theta\\ &\le \frac{1}{2\pi}\int_0^{2\pi}\|x\|_{\ell^\infty}^2\left(\sum_{t=0}^{\infty}r^t\right)^2 d\theta\\ &= \|x\|_{\ell^\infty}^2\left(\frac{1}{1-r}\right)^2\\ &\le \|x\|_{\ell^2}^2\left(\frac{1}{1-r}\right)^2 < \infty.\end{aligned}$$

Theorem 3.B.3. *For $K(\cdot) \in \mathcal{H}^\infty$, it holds that*

$$\|K\|_{\mathcal{H}^\infty} = \sup_{X\in\mathcal{H}^2}\frac{\|KX\|_{\mathcal{H}^2}}{\|X\|_{\mathcal{H}^2}}. \tag{3.57}$$

Proof. The proof essentially follows [McC03] with some details filled in and minor refinements. We start by noting that the RHS of (3.57) is the operator norm $|||K|||_2 := \sup_{X\in\mathcal{H}^2}\frac{\|KX\|_{\mathcal{H}^2}}{\|X\|_{\mathcal{H}^2}}$ of the multiplication operator $X(z) \to K(z)X(z)$ and first establish that $|||K|||_2 \le \|K\|_{\mathcal{H}^\infty}$. Indeed, for every $X \in \mathcal{H}^2$, we have

$$\begin{aligned}\|K\,X\|_{\mathcal{H}^2} &= \sqrt{\sup_{r<1}\frac{1}{2\pi}\int_0^{2\pi}|K(re^{i\theta})X(re^{i\theta})|^2 d\theta}\\ &\le \sqrt{\sup_{r<1}\frac{1}{2\pi}\int_0^{2\pi}|X(re^{i\theta})|^2\left(\sup_{|z|<1}|K(z)|\right)^2 d\theta}\\ &= \|K\|_{\mathcal{H}^\infty}\sqrt{\sup_{r<1}\frac{1}{2\pi}\int_0^{2\pi}|X(re^{i\theta})|^2 d\theta}\\ &= \|K\|_{\mathcal{H}^\infty}\|X\|_{\mathcal{H}^2},\end{aligned}$$

which, upon division by $\|X\|_{\mathcal{H}^2}$, establishes the desired upper bound $|||K|||_2 \le \|K\|_{\mathcal{H}^\infty}$.

To complete the proof, we show that $|||K|||_2 \ge \|K\|_{\mathcal{H}^\infty}$. Applying

$$\|KX\|_{\mathcal{H}^2} \le |||K|||_2\|X\|_{\mathcal{H}^2}$$

repeatedly, we get, for every $n \in \mathbb{N}$,

$$\|K^nX\|_{\mathcal{H}^2} \le |||K|||_2^n\|X\|_{\mathcal{H}^2}. \tag{3.58}$$

Without loss of generality, we can restrict ourselves to $|||K|||_2 = 1$ as otherwise we can simply

consider $K' := K/\|\!|K|\!\|_2$. Next, towards a contradiction, assume that $\|\!|K|\!\|_2 < \|K\|_{\mathcal{H}^\infty}$, which, thanks to $\|\!|K|\!\|_2 = 1$, results in $1 < \|K\|_{\mathcal{H}^\infty} = \sup_{r<1,\, 0\leq\theta<2\pi} |K(re^{i\theta})|$. As $K(\cdot) \in \mathcal{H}^\infty$ by assumption, it follows that $K(z)$ is analytic and thus continuous inside the unit disk. Hence, there exist $r' \in (0,1), \epsilon > 0$ and an interval $[\underline{\theta}, \overline{\theta}) \subset [0, 2\pi)$ with $\overline{\theta} - \underline{\theta} = \delta > 0$ such that

$$|K(r'e^{i\theta'})| > 1 + \epsilon, \quad \forall \theta' \in [\underline{\theta}, \overline{\theta}). \tag{3.59}$$

Now we take $X(z) = 1 = \mathcal{Z}\left\{\mathbb{1}_{\{t=0\}}[t]\right\}$ which clearly satisfies $\|X\|_{\mathcal{H}^2} = 1$. Inserting this into (3.58), we obtain

$$\|K^n X\|_{\mathcal{H}^2}^2 \leq \|\!|K|\!\|_2^{2n} \|X\|_{\mathcal{H}^2}^2 = 1.$$

This finalizes the proof by leading to the following contradiction

$$\begin{aligned}
1 &\geq \|K^n X\|_{\mathcal{H}^2}^2 \\
&= \sup_{0<r<1} \frac{1}{2\pi} \int_0^{2\pi} |K(re^{i\theta})|^{2n} d\theta \\
&\geq \frac{1}{2\pi} \int_0^{2\pi} |K(r'e^{i\theta})|^{2n} d\theta \\
&\geq \frac{1}{2\pi} \int_0^{2\pi} ((1+\epsilon)\mathbb{1}_{\{\theta\in[\underline{\theta},\overline{\theta})\}})^{2n} d\theta \qquad (3.60) \\
&= \frac{\delta}{2\pi}(1+\epsilon)^{2n} \xrightarrow[n\to\infty]{} \infty,
\end{aligned}$$

where in (3.60) we used (3.59). □

Theorem 3.B.4. *Let $\mathcal{L}$ and $\mathcal{L}'$ be LTI systems with corresponding transfer functions $K(z)$ and $K'(z)$, both in $\mathcal{H}^\infty$. It holds that*

$$\|K - K'\|_{\mathcal{H}^\infty} = \sup_{X \in \mathcal{H}^2} \frac{\|(K-K')X\|_{\mathcal{H}^2}}{\|X\|_{\mathcal{H}^2}} \tag{3.61}$$

$$= \sup_{\|x\|_{\ell^2}=1} \|(k-k') * x\|_{\ell^2} \tag{3.62}$$

$$= \sup_{\|x\|_{\ell^2}=1} \|\mathcal{L}x - \mathcal{L}'x\|_{\ell^2}. \tag{3.63}$$

Proof. Equation (3.61) follows from Theorem 3.B.3 upon noting that $K - K' \in \mathcal{H}^\infty$ by application of the triangle inequality. Next, (3.62) is established through

$$\begin{aligned}
\sup_{X \in \mathcal{H}^2} \frac{\|(K-K')X\|_{\mathcal{H}^2}}{\|X\|_{\mathcal{H}^2}} &= \sup_{x\in\ell^2} \frac{\|(k-k') * x\|_{\ell^2}}{\|x\|_{\ell^2}} \\
&= \sup_{x\in\ell^2} \left\| (k-k') * \frac{x}{\|x\|_{\ell^2}} \right\|_{\ell^2} \\
&= \sup_{\|x\|_{\ell^2}=1} \|(k-k') * x\|_{\ell^2},
\end{aligned}$$

where we used Theorem 3.B.2, (3.37), and the fact that convolution is linear. Finally, (3.63) follows directly from (3.35). □

3.C. Proof of Theorem 3.3.4

The proof of Theorem 3.3.4 proceeds by establishing an upper bound on $N(\epsilon; \mathcal{C}(a,b), \rho)$ through the construction of a covering and a matching lower bound by constructing an explicit packing. We define this concept next.

Definition 3.C.1. *Let $(\mathcal{X}, \rho)$ be a metric space. An ϵ-packing of a compact set $\mathcal{C} \subset \mathcal{X}$ with respect to the metric ρ is a set $\{x_1, ..., x_N\} \subset \mathcal{C}$ such that $\rho(x_i, x_j) > \epsilon$, for all distinct i, j. The ϵ-packing number $M(\epsilon; \mathcal{X}, \rho)$ is the cardinality of the largest ϵ-packing.*

We can then use the following Lemma to relate covering and packing number.

Lemma 3.C.2. *Let $(\mathcal{X}, \rho)$ be a metric space and $\mathcal{C}$ a compact set in $\mathcal{X}$. For all $\epsilon > 0$, it holds that*

$$M(2\epsilon; \mathcal{C}, \rho) \leq N(\epsilon; \mathcal{C}, \rho) \leq M(\epsilon; \mathcal{C}, \rho).$$

Proof. First, choose a minimal ϵ-covering and a maximal 2ϵ-packing of $\mathcal{C}$. Since no two centers of the 2ϵ-packing can lie in the same ball of the ϵ-covering, it follows that $M(2\epsilon; \mathcal{C}, \rho) \leq N(\epsilon; \mathcal{C}, \rho)$. To establish $N(\epsilon; \mathcal{C}, \rho) \leq M(\epsilon; \mathcal{C}, \rho)$, we note that, given a maximum ϵ-packing with cardinality $M(\epsilon; \mathcal{C}, \rho)$, for every $x \in \mathcal{C}$, we have the center of at least one of the balls in the packing within distance less than ϵ. If this were not the case, we could add another ball to the packing thereby violating its maximality. This maximal packing hence also provides an ϵ-covering and hence $N(\epsilon; \mathcal{C}, \rho) \leq M(\epsilon; \mathcal{C}, \rho)$. □

To continue with our program, we next construct an explicit packing.

Lemma 3.C.3. *Let $a, b > 0$ and consider the set*

$$\mathcal{C}(a,b) = \{\mathcal{L} \mid k_{\mathcal{L}}[t] \in \mathbb{R},\ |k_{\mathcal{L}}[t]| \leq ae^{-bt},\ \forall t \in \mathbb{N}_0 \text{ and } k_{\mathcal{L}}[t] = 0,\ \forall t \in \mathbb{Z} \backslash \mathbb{N}_0\}$$

equipped with the metric

$$\rho(\mathcal{L}, \mathcal{L}') = \|K - K'\|_{\mathcal{H}^\infty}.$$

For every $\epsilon \in (0, a)$, there exists a (2ϵ)-packing of $\mathcal{C}(a,b)$ with $M_{2\epsilon}$ elements, where

$$\log_2(M_{2\epsilon}) \geq \frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2 - o\left(\left(\log\left(\frac{a}{\epsilon}\right)\right)^2\right),$$

with $\gamma := \log_2(e)$.

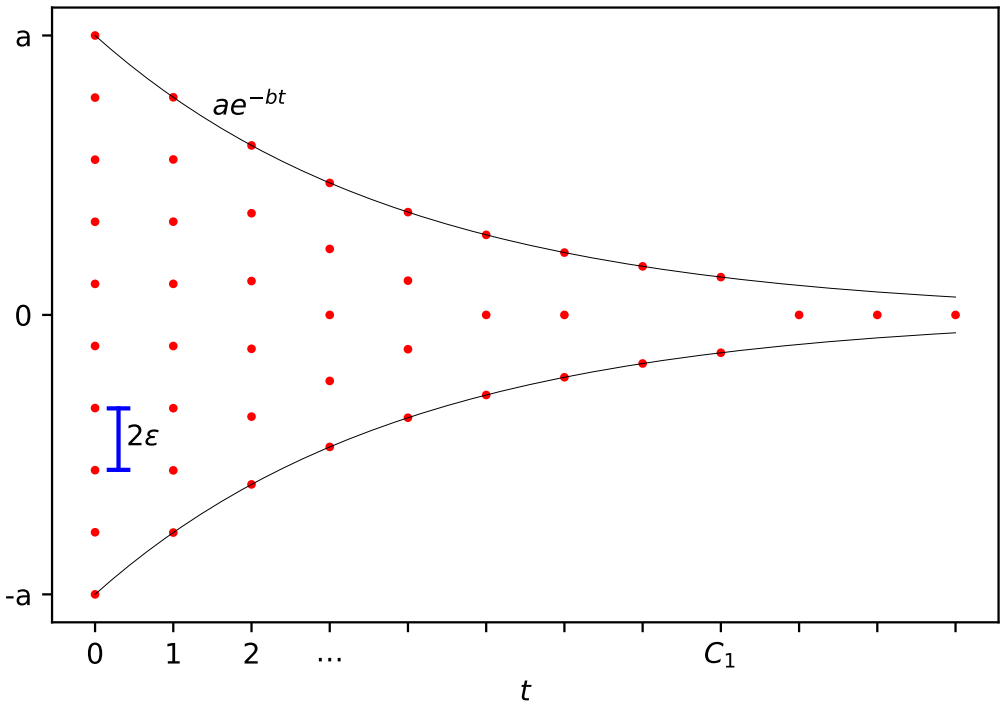


Figure 3.1: The packing is constructed by quantizing the impulse response at each time instant t with quantization interval size of at least 2ϵ.

Proof. We explicitly construct a (2ϵ)-packing as visualized in Figure 3.1. Define $C_1 := \left\lceil \frac{1}{b} \log\left(\frac{a}{\epsilon}\right) \right\rceil - 1$. Now, for all $t \in \{0, \dots, C_1\}$, it holds that

$$\begin{aligned} & t \leq C_1 < \frac{1}{b} \log\left(\frac{a}{\epsilon}\right) \\ \Rightarrow \quad & b\,t < \log\left(\frac{a}{\epsilon}\right) \\ \Rightarrow \quad & -b\,t > \log\left(\frac{\epsilon}{a}\right) \\ \Rightarrow \quad & e^{-bt} > \frac{\epsilon}{a} \\ \Rightarrow \quad & \frac{ae^{-bt}}{\epsilon} > 1. \end{aligned}$$

Next, for $t \in \{0, \dots, C_1\}$, we set $n_t := \left\lceil \frac{2ae^{-bt}}{2\epsilon} \right\rceil - 1 \in \left[1, \frac{2ae^{-bt}}{2\epsilon}\right)$ and $\delta_t := \frac{2ae^{-bt}}{n_t} > 2\epsilon$. Now, we define the set

$$\mathcal{P} := \left\{ \widetilde{\mathcal{L}}_{i_0,\dots,i_{C_1}} \mid i_\ell \in \{0, \dots, n_\ell\}, \text{ for } \ell \in \{0, \dots, C_1\} \right\},$$

where $\widetilde{\mathcal{L}}_{i_0,\dots,i_{C_1}}$ is the LTI system associated with the impulse response

$$\widetilde{k}_{i_0,\dots,i_{C_1}}[t] := \begin{cases} -ae^{-bt} + i_t\delta_t, & 0 \le t \le C_1 \\ 0, & \text{otherwise} \end{cases},$$

and we show that $\mathcal{P}$ constitutes a (2ϵ)-packing of $\mathcal{C}(a,b)$. First, we establish that $\mathcal{P} \subset \mathcal{C}(a,b)$ by verifying that $\left|\widetilde{k}_{i_0,\dots,i_{C_1}}[t]\right| \le ae^{-bt}$, $\forall t \in \mathbb{N}_0$, holds for all $\widetilde{k}_{i_0,\dots,i_{C_1}} \in \mathcal{P}$. Indeed, for $t \in \{0,\dots,C_1\}$, we have

$$-ae^{-bt} \le \widetilde{k}_{i_0,\dots,i_{C_1}}[t] = -ae^{-bt} + i_t\delta_t \le -ae^{-bt} + n_t\delta_t = -ae^{-bt} + 2ae^{-bt} = ae^{-bt},$$

and for $t > C_1$,

$$\widetilde{k}_{i_0,\dots,i_{C_1}}[t] = 0.$$

Next, we show that for distinct $\widetilde{\mathcal{L}}_{i_0,\dots,i_{C_1}}, \widetilde{\mathcal{L}}_{j_0,\dots,j_{C_1}} \in \mathcal{P}$, i.e., there is at least one $\ell \in \{0,\dots,C_1\}$ such that $i_\ell \neq j_\ell$, it holds that $\rho(\widetilde{\mathcal{L}}_{i_0,\dots,i_{C_1}}, \widetilde{\mathcal{L}}_{j_0,\dots,j_{C_1}}) > 2\epsilon$. Indeed, for any such ℓ, we have

$$\begin{aligned}
\rho(\widetilde{\mathcal{L}}_{i_0,\dots,i_{C_1}}, \widetilde{\mathcal{L}}_{j_0,\dots,j_{C_1}}) &= \sup_{\|x\|_{\ell^2}=1} \left\| \left(\widetilde{k}_{i_0,\dots,i_{C_1}} - \widetilde{k}_{j_0,\dots,j_{C_1}}\right) * x \right\|_{\ell^2} && (3.64)\\
&\ge \left\| \widetilde{k}_{i_0,\dots,i_{C_1}} - \widetilde{k}_{j_0,\dots,j_{C_1}} \right\|_{\ell^2} && (3.65)\\
&= \sqrt{\sum_{t=0}^{\infty} \left(\widetilde{k}_{i_0,\dots,i_{C_1}}[t] - \widetilde{k}_{j_0,\dots,j_{C_1}}[t]\right)^2} \\
&\ge \sqrt{\left(\widetilde{k}_{i_0,\dots,i_{C_1}}[\ell] - \widetilde{k}_{j_0,\dots,j_{C_1}}[\ell]\right)^2} \\
&= |-ae^{-b\ell} + i_\ell\delta_\ell + ae^{-b\ell} - j_\ell\delta_\ell| \\
&= |i_\ell - j_\ell|\delta_\ell \\
&> 2\epsilon, && (3.66)
\end{aligned}$$

where in (3.64) we used Theorem 3.B.4, in (3.65) we inserted the particular choice $x[t] = \mathbb{1}_{\{t=0\}}[t]$ to lower-bound the sup, and in (3.66) we used $\delta_\ell > 2\epsilon$. This establishes that $\mathcal{P}$ constitutes a (2ϵ)-packing of $\mathcal{C}(a,b)$ with respect to the metric $\rho(\cdot,\cdot)$ specified in (3.39). It remains to bound the cardinality of $\mathcal{P}$, which we denote by $M_{2\epsilon}$.

Specifically, we have

$$
\begin{aligned}
\log_2(M_{2\epsilon}) &= \log_2 \prod_{t=0}^{C_1}(1+n_t) \\
&= \sum_{t=0}^{C_1} \log_2 \left\lceil \frac{a}{\epsilon}\, e^{-bt} \right\rceil \\
&\geq \sum_{t=0}^{C_1} \log_2 \left(\frac{a}{\epsilon}\, e^{-bt} \right) \\
&= (C_1+1)\log_2\left(\frac{a}{\epsilon}\right) + \sum_{t=0}^{C_1} \log_2\left(e^{-bt}\right) \\
&= (C_1+1)\log_2\left(\frac{a}{\epsilon}\right) - b\ \log_2(e) \sum_{t=0}^{C_1} t \\
&= (C_1+1)\log_2\left(\frac{a}{\epsilon}\right) - b\,\gamma\,\frac{C_1(C_1+1)}{2} \\
&= \left\lceil \frac{1}{b}\log\left(\frac{a}{\epsilon}\right)\right\rceil \log_2\left(\frac{a}{\epsilon}\right) - b\,\gamma\,\frac{(\left\lceil \frac{1}{b}\log\left(\frac{a}{\epsilon}\right)\right\rceil - 1)\left\lceil \frac{1}{b}\log\left(\frac{a}{\epsilon}\right)\right\rceil}{2} \\
&= \left\lceil \frac{1}{b}\log\left(\frac{a}{\epsilon}\right)\right\rceil \log_2\left(\frac{a}{\epsilon}\right) - \frac{b\,\gamma}{2}\left\lceil \frac{1}{b}\log\left(\frac{a}{\epsilon}\right)\right\rceil^2 + \frac{b\,\gamma}{2}\left\lceil \frac{1}{b}\log\left(\frac{a}{\epsilon}\right)\right\rceil \\
&\geq \frac{1}{b}\log\left(\frac{a}{\epsilon}\right)\log_2\left(\frac{a}{\epsilon}\right) - \frac{b\,\gamma}{2}\left(\frac{1}{b}\log\left(\frac{a}{\epsilon}\right)+1\right)^2 + \frac{b\,\gamma}{2}\frac{1}{b}\log\left(\frac{a}{\epsilon}\right) \\
&= \frac{1}{b}\log\left(\frac{a}{\epsilon}\right)\log_2\left(\frac{a}{\epsilon}\right) - \frac{b\,\gamma}{2b^2}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2 - \gamma\log\left(\frac{a}{\epsilon}\right) - \frac{b\,\gamma}{2} + \frac{\gamma}{2}\log\left(\frac{a}{\epsilon}\right) \\
&= \frac{\gamma}{b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2 - \frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2 - \frac{\gamma}{2}\log\left(\frac{a}{\epsilon}\right) - \frac{b\,\gamma}{2} \qquad (3.67) \\
&= \frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2 - \frac{\gamma}{2}\log\left(\frac{a}{\epsilon}\right) - \frac{b\,\gamma}{2} \qquad (3.68) \\
&= \frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2 - o\left(\left(\log\left(\frac{a}{\epsilon}\right)\right)^2\right),
\end{aligned}
$$

where in (3.67) we used $\log_2(x) = \gamma\log(x)$. □

Next, we construct a covering for $\mathcal{C}(a,b)$.

Lemma 3.C.4. *Let $a, b > 0$ and consider the set*

$$\mathcal{C}(a,b) = \{\mathcal{L} \mid k_{\mathcal{L}}[t] \in \mathbb{R},\ |k_{\mathcal{L}}[t]| \leq a e^{-bt},\ \forall t \in \mathbb{N}_0 \text{ and } k_{\mathcal{L}}[t] = 0,\ \forall t \in \mathbb{Z}\backslash\mathbb{N}_0\},$$

equipped with the metric

$$\rho(\mathcal{L},\mathcal{L}') = \|K - K'\|_{\mathcal{H}^\infty}.$$

For every $\epsilon \in (0,a)$, there exists an ϵ-covering of $\mathcal{C}(a,b)$ with N_ϵ elements, where

$$\log_2(N_\epsilon) \leq \frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2 + o\left(\left(\log\left(\frac{a}{\epsilon}\right)\right)^2\right),$$

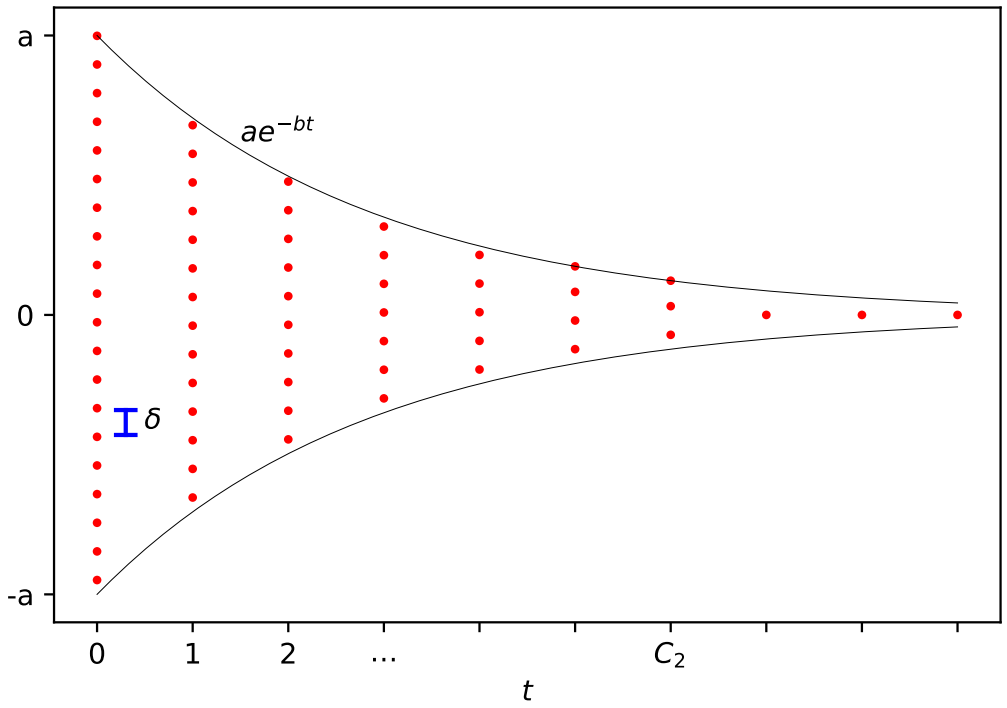


Figure 3.2: The covering is constructed by quantizing the impulse response at each time instant t with quantization interval size δ.

with $\gamma := \log_2(e)$.

Proof. The proof is effected by explicit construction of an ϵ-covering as visualized in Figure 3.2. We start by defining

$$C_2 := \left\lfloor \frac{1}{b} \log\left(\frac{2a}{\epsilon\,(1-e^{-b})} \right) \right\rfloor, \quad \delta := \frac{\epsilon}{C_2+1},$$

and

$$n_t := \left\lceil \frac{2ae^{-bt}}{\delta} \right\rceil, \qquad \text{for } t \in \{0,\dots,C_2\}.$$

As indicated in Figure 3.2, we quantize the impulse response at each time instant $t \in \{0,\dots,C_2\}$ with quantization interval size δ using n_t points. To formalize this, we start by defining, for $t \in \{0,\dots,C_2\}$, the mappings

$$f_t : \{1,\dots,n_t\} \to [-ae^{-bt}, ae^{-bt}]$$
$$f_t(i) := \min\left\{ -ae^{-bt} - \frac{\delta}{2} + i\delta,\ ae^{-bt} \right\},$$

and show that, for all $t \in \{0,\dots,C_2\}$, the following properties hold:

(P1) $f_t(1) \le -ae^{-bt} + \frac{\delta}{2}$,

(P2) $f_t(n_t) \geq ae^{-bt} - \frac{\delta}{2}$,

(P3) $f_t(i+1) - f_t(i) \leq \delta$, for $i \in \{1, \ldots, n_t - 1\}$.

First, (P1) follows by

$$f_t(1) = \min\left\{-ae^{-bt} - \frac{\delta}{2} + \delta,\, ae^{-bt}\right\} \leq -ae^{-bt} - \frac{\delta}{2} + \delta = -ae^{-bt} + \frac{\delta}{2}.$$

To establish (P2) we note that either

$$f_t(n_t) = ae^{-bt} \qquad \text{or} \qquad f_t(n_t) = -ae^{-bt} - \frac{\delta}{2} + n_t\delta.$$

In the former case we have $f_t(n_t) = ae^{-bt} \geq ae^{-bt} - \frac{\delta}{2}$, and in the latter, we obtain

$$f_t(n_t) = -ae^{-bt} - \frac{\delta}{2} + n_t\delta \geq -ae^{-bt} - \frac{\delta}{2} + 2ae^{-bt} = ae^{-bt} - \frac{\delta}{2}.$$

Finally, to prove (P3), we observe that, for all $i \in \{1, \ldots, n_t - 1\}$, it holds that

$$\begin{aligned} -ae^{-bt} - \frac{\delta}{2} + i\delta &\leq -ae^{-bt} - \frac{\delta}{2} + (n_t - 1)\delta \\ &\leq -ae^{-bt} - \frac{\delta}{2} + \left(\frac{2ae^{-bt}}{\delta}\right)\delta \\ &= ae^{-bt} - \frac{\delta}{2} \leq ae^{-bt}. \end{aligned}$$

Thus, $f_t(i) = -ae^{-bt} - \frac{\delta}{2} + i\delta$, for all $i \in \{1, \ldots, n_t - 1\}$, and therefore

$$\begin{aligned} f_t(i+1) - f_t(i) &= \min\left\{-ae^{-bt} - \frac{\delta}{2} + (i+1)\delta,\, ae^{-bt}\right\} - \left(-ae^{-bt} - \frac{\delta}{2} + i\delta\right) \\ &\leq -ae^{-bt} - \frac{\delta}{2} + (i+1)\delta - \left(-ae^{-bt} - \frac{\delta}{2} + i\delta\right) = \delta, \qquad \text{for } i \in \{1, \ldots, n_t - 1\}, \end{aligned}$$

which establishes (P3).

Together, (P1)-(P3) imply that, for every $t \in \{0, \ldots, C_2\}$ and $x \in [-ae^{-bt}, ae^{-bt}]$, there is an $i \in \{1, \ldots, n_t\}$ such that $|f_t(i) - x| \leq \frac{\delta}{2}$. Now, we define the set

$$\mathcal{M} := \left\{\widetilde{\mathcal{L}}_{i_0,\ldots,i_{C_2}} \mid i_\ell \in \{1, \ldots, n_\ell\},\ \text{for } \ell \in \{0, \ldots, C_2\}\right\},$$

where $\widetilde{\mathcal{L}}_{i_0,\ldots,i_{C_2}}$ is the LTI system associated with the impulse response

$$\widetilde{k}_{i_0,\ldots,i_{C_2}}[t] := \begin{cases} f_t(i_t), & 0 \leq t \leq C_2 \\ 0, & \text{otherwise} \end{cases},$$

and we show that $\mathcal{M}$ is, indeed, an ϵ-covering for $\mathcal{C}(a,b)$. Fix $\mathcal{L} \in \mathcal{C}(a,b)$ with corresponding impulse response $k[\cdot]$. As just established, for each $t \in \{0,\dots,C_2\}$, there is an $i_t \in \{1,\dots,n_t\}$ such that $|k[t] - f_t(i_t)| \le \frac{\delta}{2}$.
Hence, the corresponding $\widetilde{k}_{i_0,\dots,i_{C_2}} \in \mathcal{M}$ satisfies

$$
\begin{aligned}
\|K - \widetilde{K}_{i_0,\dots,i_{C_2}}\|_{\mathcal{H}^\infty} &\le \sum_{t=0}^{\infty} |k[t] - \widetilde{k}_{i_0,\dots,i_{C_2}}[t]| && (3.69)\\
&= \sum_{t=0}^{C_2} |k[t] - \widetilde{k}_{i_0,\dots,i_{C_2}}[t]| + \sum_{t=C_2+1}^{\infty} |k[t]| \\
&= \sum_{t=0}^{C_2} |k[t] - f_t(i_t)| + \sum_{t=C_2+1}^{\infty} |k[t]| \\
&\le (C_2+1)\frac{\delta}{2} + \sum_{t=C_2+1}^{\infty} a e^{-bt} \\
&= (C_2+1)\frac{\delta}{2} + a\,\frac{e^{-b(C_2+1)}}{1-e^{-b}} \\
&\le (C_2+1)\frac{\delta}{2} + a\,\frac{e^{-\log\left(\frac{2a}{\epsilon(1-e^{-b})}\right)}}{1-e^{-b}} && (3.70)\\
&= (C_2+1)\frac{\delta}{2} + a\,\frac{\frac{\epsilon(1-e^{-b})}{2a}}{1-e^{-b}} \\
&= \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon,
\end{aligned}
$$

where in (3.69) we applied Lemma 3.4.2, and in (3.70) we used $C_2 \ge \frac{1}{b}\log\left(\frac{2a}{\epsilon\left(1-e^{-b}\right)}\right) - 1$.

It remains to upper-bound N_ϵ, the number of elements in $\mathcal{M}$:

$$\begin{aligned}
\log_2(N_\epsilon) &= \log_2\left(\prod_{t=0}^{C_2} n_t\right) \\
&= \sum_{t=0}^{C_2} \log_2(n_t) \\
&= \sum_{t=0}^{C_2} \log_2\left(\left\lceil \frac{2ae^{-bt}}{\delta} \right\rceil\right) \\
&\leq \sum_{t=0}^{C_2} \left\lceil \log_2\left(\frac{2ae^{-bt}}{\delta}\right) \right\rceil \qquad (3.71) \\
&\leq \sum_{t=0}^{C_2} \left(\log_2\left(\frac{2ae^{-bt}}{\delta}\right) + 1\right) \\
&= -b\log_2(e)\sum_{t=0}^{C_2} t + (C_2+1)\left(\log_2\left(\frac{a}{\delta}\right) + 2\right) \\
&= -\frac{\gamma b}{2} C_2(C_2+1) + (C_2+1)\left(\log_2\left(\frac{a}{\delta}\right) + 2\right) \\
&= (C_2+1)\left(\gamma\log\left(\frac{a}{\delta}\right) - \frac{\gamma b}{2} C_2 + 2\right) \qquad (3.72) \\
&= (C_2+1)\left(\gamma\log\left(\frac{a(C_2+1)}{\epsilon}\right) - \frac{\gamma b}{2} C_2 + 2\right) \\
&= (C_2+1)\left(\gamma\log\left(\frac{a}{\epsilon}\right) + \gamma\log(C_2+1) - \frac{\gamma b}{2} C_2 + 2\right),
\end{aligned}$$

where in (3.71) we used $\log_2(\lceil x \rceil) \leq \lceil \log_2(x) \rceil$, $\forall x > 0$, and in (3.72) we employed $\log_2(x) = \gamma \log(x)$. Next, we note from the definition of C_2 that

$$\frac{1}{b}\log\left(\frac{a}{\epsilon}\right) + K_1(b) - 1 \leq C_2 \leq \frac{1}{b}\log\left(\frac{a}{\epsilon}\right) + K_1(b),$$

with $K_1(b) := \frac{1}{b}\log\left(\frac{2}{1-e^{-b}}\right)$.

Now we further upper-bound as follows:

$$
\begin{aligned}
&(C_2+1)\left(\gamma\log\left(\frac{a}{\epsilon}\right)+\gamma\log\left(C_2+1\right)-\frac{\gamma b}{2}C_2+2\right)\\
&\leq (C_2+1)\left(\gamma\log\left(\frac{a}{\epsilon}\right)-\frac{\gamma b}{2}\left(\frac{1}{b}\log\left(\frac{a}{\epsilon}\right)+K_1(b)-1\right)+\gamma\log\left(C_2+1\right)+2\right)\\
&= (C_2+1)\left(\frac{\gamma}{2}\log\left(\frac{a}{\epsilon}\right)+\gamma\log\left(C_2+1\right)-\frac{\gamma b}{2}K_1(b)+\frac{\gamma b}{2}+2\right)\\
&= (C_2+1)\left(\frac{\gamma}{2}\log\left(\frac{a}{\epsilon}\right)+\gamma\log\left(C_2+1\right)+K_2(b)\right)\\
&\leq \left(\frac{1}{b}\log\left(\frac{a}{\epsilon}\right)+K_3(b)\right)\left(\frac{\gamma}{2}\log\left(\frac{a}{\epsilon}\right)+\gamma\log\left(C_2+1\right)+K_2(b)\right)\\
&= \frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2+\frac{\gamma}{b}\log\left(\frac{a}{\epsilon}\right)\log(C_2+1)+\frac{1}{b}\log\left(\frac{a}{\epsilon}\right)K_2(b)\\
&\quad +K_3(b)\frac{\gamma}{2}\log\left(\frac{a}{\epsilon}\right)+K_3(b)\gamma\log\left(C_2+1\right)+K_3(b)K_2(b)\\
&= \frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2+\frac{\gamma}{b}\log\left(\frac{1}{\epsilon}\right)\log(C_2+1)+K_4(b)\log\left(\frac{1}{\epsilon}\right)\\
&\quad +K_5(a,b)\log(C_2+1)+K_6(a,b)\\
&= \frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2+o\left(\left(\log\left(\frac{a}{\epsilon}\right)\right)^2\right),
\end{aligned}
\tag{3.73}
$$

where $K_2(b) := -\frac{\gamma b}{2}K_1(b)+\frac{\gamma b}{2}+2$, $K_3(b) := K_1(b)+1$, $K_4(b) := \frac{1}{b}K_2(b)+\frac{\gamma}{2}K_3(b)$, $K_5(a,b) := \frac{\gamma}{b}\log(a)+\gamma K_3(b)$, and $K_6(a,b) := K_4(b)\log(a)+K_3(b)K_2(b)$. The last equality follows from $\log(C_2+1) = o(\log(\epsilon^{-1}))$. □

Having established all the ingredients, the proof of Theorem 3.3.4 can be effected in a few lines.

Proof of Theorem 3.3.4. Fix $\epsilon > 0$. According to Lemma 3.C.3 below, there exists a (2ϵ)-packing of $\mathcal{C}(a,b)$ with $M_{2\epsilon}$ elements, where

$$\log_2(M_{2\epsilon}) \geq \frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2-o\left(\left(\log\left(\frac{a}{\epsilon}\right)\right)^2\right).$$

Further, by Lemma 3.C.4 below, there exists an ϵ-covering of $\mathcal{C}(a,b)$ with N_ϵ elements, where

$$\log_2(N_\epsilon) \leq \frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2+o\left(\left(\log\left(\frac{a}{\epsilon}\right)\right)^2\right).$$

Using Lemma 3.C.2, we can hence sandwich the metric entropy according to

$$\begin{aligned}\frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2 - o\left(\left(\log\left(\frac{a}{\epsilon}\right)\right)^2\right) &\leq \log_2(M_{2\epsilon}) \\ &\leq \log_2(M(2\epsilon;\mathcal{C}(a,b),\rho)) \\ &\leq \log_2(N(\epsilon;\mathcal{C}(a,b),\rho)) \\ (\ &= \mathcal{E}(\epsilon;\mathcal{C}(a,b),\rho)\ ) \\ &\leq \log_2(N_\epsilon) \\ &\leq \frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2 + o\left(\left(\log\left(\frac{a}{\epsilon}\right)\right)^2\right).\end{aligned}$$

Dividing by $\frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2$ and taking $\lim_{\epsilon\to 0}$, then yields

$$1 \leq \lim_{\epsilon\to 0} \frac{\mathcal{E}(\epsilon;\mathcal{C}(a,b),\rho)}{\frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2} \leq 1,$$

which establishes (3.40). □

In fact, we can even get the following, slightly more precise result.

Corollary 3.C.5. *Observing the precise nature of the lower and upper bounds in* (3.68) *and* (3.73) *respectively, we can also write*

$$\mathcal{E}(\epsilon;\mathcal{C}(a,b),\rho) = \frac{\gamma}{2b}\left(\log\left(\frac{a}{\epsilon}\right)\right)^2 + \mathcal{O}\Big(\log\left(\frac{1}{\epsilon}\right)\log\left(\log\left(\frac{1}{\epsilon}\right)\right)\Big)$$

instead of (3.40).

Chapter 4

Metric-entropy limits on the approximation of nonlinear dynamical systems

4.1. Introduction

It is well known that neural networks can approximate almost every function arbitrarily well [Cyb89; Fun89; HSW89; Bar93; Bar94]. The recently developed Kolmogorov-Donoho rate-distortion theory for deep neural network approximation [Böl+19; Elb+21] goes a step further by quantifying how effective such approximations are in terms of the description complexity of the neural networks relative to that of the functions they are to approximate. Specifically, [Elb+21] determines the length of the bitstring needed to specify (deep) ReLU networks approximating functions (mapping $\mathbb{R}^d$ to $\mathbb{R}$) in a given class to within a prescribed error ϵ. It is then shown that for a wide variety of function classes, the length of this bitstring exhibits the same scaling behavior, in ϵ, as the metric entropy of the function class under consideration (see Table 4.1 below). This means that neural networks are universally Kolmogorov-Donoho optimal for all these function classes.

In the present chapter, we extend the philosophy of [Elb+21] to the approximation of nonlinear sequence-to-sequence mappings through recurrent neural networks (RNNs). Concretely, we consider Lipschitz fading-memory (LFM) systems. In essence, this notion describes systems that gradually forget past inputs, with the speed of memory decay quantified in terms of a Lipschitz property. Such systems find application, inter alia, in finance [NX22] and material science [Dil75; Wan65; Day13]. We first develop tools for quantifying the metric entropy of classes of LFM systems with a given memory decay rate. A general construction of RNNs approximating LFM systems is then shown to yield Kolmogorov-Donoho-optimality for LFM systems of exponentially or polynomially decaying memory.

Related work Learning of linear dynamical systems has been studied extensively in the literature [Bak+23; BGS07; HSZ17; WJ20; HGB22]. Notably, [HGB22] provides explicit RNN realizations and approximations for wide classes of linear dynamical systems, including time-

varying ones. Going beyond linear systems, learning of nonlinear finite memory systems through RNNs within the finite-state-machine framework was considered in [GHL95]. This program is extended to approximately finite-memory systems in [San94] and fading-memory systems in [Vol59; Wie66; BC85]. In particular, [BC85] formalizes the concept of fading-memory systems in control theory, demonstrating that continuous-time fading-memory operators can be approximated by Volterra series. Subsequently, [Mat93] established that discrete-time fading-memory systems can be identified using neural networks. Moreover, [GO18] demonstrated that echo state networks, a specialized architecture within the RNN family, serve as universal approximators for discrete-time fading-memory systems. None of the studies just reviewed addresses the issue of quantifying the RNN description complexity relative to that of the class of (nonlinear) systems being approximated.

Organization of the chapter The remainder of Section 4.1 summarizes notation. In Section 4.2, we introduce our setup and provide a definition of metric-entropy optimality in a very general context, encompassing the approximation of functions as well as dynamical systems. Section 4.3 develops tools for characterizing the metric entropy of LFM systems. In Section 4.4, we employ these tools to derive precise scaling results for the metric entropy of exponentially Lipschitz fading-memory (ELFM) and polynomially Lipschitz fading-memory (PLFM) systems. Section 4.5 presents a construction for the approximation of general LFM systems by RNNs. Finally, in Section 4.6, we combine the results developed in the previous sections to prove that RNNs can approximate ELFM and PLFM systems in metric-entropy-optimal manner.

Notation $\mathbb{N}_0$ and $\mathbb{N}_+$ denote the set of natural numbers including and excluding 0, respectively. For $N \in \mathbb{N}_0$, $[\![N]\!]$ stands for the set $\{0, 1, \ldots, N\}$, while $[\![N]\!]^{\pm}$ refers to $\{-N, \ldots, -1, 0, 1, \ldots, N\}$. The cardinality of the finite set U is designated by $|U|$. Sequences $x[t] \in \mathbb{R}$ are indexed by $t \in \mathbb{Z}$ or $t \in \mathbb{N}_0$ and we use $\mathbb{R}^{\mathbb{Z}}$ and $\mathbb{R}^{\mathbb{N}_0}$, respectively, to denote the set of such sequences. We refer to the set of all finite-length bitstrings by $\{0,1\}^*$. The transpose of the matrix A is A^T. For matrices $A_1, \ldots, A_N$, $\mathrm{diag}(A_1, A_2, \ldots, A_N)$ denotes the block-diagonal matrix with the A_i on the main diagonal. The $N \times N$ identity matrix is $\mathbb{I}_N$ and 0_N stands for the N-dimensional column vector with all entries equal to 0. For the vector $x \in \mathbb{R}^d$, we let $\|x\|_\infty := \max_{i=1,2,\ldots,d} |x_i|$. $\log(\cdot)$ refers to the logarithm to base 2, $\log^{(n)} = \log \circ \cdots \circ \log$ is the n-fold iterated logarithm, and $\log^\tau(\cdot) = (\log(\cdot))^\tau$, for $\tau \in \mathbb{R}$. The composition of functions f_1, f_2 is denoted by $f_2 \circ f_1$ (or $f_1 \circ f_2$). Let $f(\epsilon)$ and $g(\epsilon)$, in both cases for $\epsilon > 0$, be strictly positive for all small enough values of ϵ. We use $f(\epsilon) = o(g(\epsilon))$ to indicate that $\lim_{\epsilon\to 0} \frac{f(\epsilon)}{g(\epsilon)} = 0$ and we express $\limsup_{\epsilon\to 0} \frac{f(\epsilon)}{g(\epsilon)} < \infty$ by $f(\epsilon) = O(g(\epsilon))$. Moreover, we write $f(\epsilon) = \Theta(g(\epsilon))$ when both $f(\epsilon) = O(g(\epsilon))$ and $g(\epsilon) = O(f(\epsilon))$. Constants are always understood to be in $\mathbb{R}$ unless explicitly stated otherwise. Finally, we say that a constant is universal if it does not depend on any of the ambient quantities.

4.2. Problem setup and metric-entropy optimality

4.2.1. ReLU network approximation

We start by defining ReLU networks.

Definition 4.2.1 (ReLU network [Elb+21]). *Let $L \in \mathbb{N}_+$ and $N_0, N_1, \ldots, N_L \in \mathbb{N}_+$. A ReLU (feedforward) neural network Φ is a mapping $\Phi : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ given by*

$$\Phi = \begin{cases} W_1, & L = 1 \\ W_2 \circ \rho \circ W_1, & L = 2 \\ W_L \circ \rho \circ W_{L-1} \circ \rho \circ \cdots \circ \rho \circ W_1, & L \geq 3, \end{cases}$$

where, for $\ell \in \{1, 2, \ldots, L\}$, $W_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$, $W_\ell(x) := A_\ell x + b_\ell$, $x \in \mathbb{R}^{N_{\ell-1}}$, are affine transformations with (weight) matrices $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ and (bias) vectors $b_\ell \in \mathbb{R}^{N_\ell}$, and the ReLU activation function $\rho : \mathbb{R} \to \mathbb{R}$, $\rho(x) = \max\{0, x\}$ acts component-wise, i.e., $\rho(x_1, \ldots, x_N) = (\rho(x_1), \ldots, \rho(x_N))$. We denote the set of all ReLU networks with input dimension $N_0 = d$ and output dimension $N_L = d'$ by $\mathcal{N}_{d,d'}$. Moreover, we define the following quantities related to the notion of size of the ReLU network Φ:

- *depth $\mathcal{L}(\Phi) := L$,*
- *the connectivity $\mathcal{M}(\Phi)$ of the network Φ is the total number of non-zero entries in the matrices A_ℓ, $\ell \in \{1, 2, ..., L\}$, and the vectors b_ℓ, $\ell \in \{1, 2, ..., L\}$,*
- *width $\mathcal{W}(\Phi) := \max_{\ell=0,\ldots,L} N_\ell$,*
- *the weight set $\mathcal{K}(\Phi)$ denotes the set of non-zero entries in the matrices A_ℓ, $\ell \in \{1, 2, ..., L\}$, and the vectors b_ℓ, $\ell \in \{1, 2, ..., L\}$,*
- *weight magnitude $\mathcal{B}(\Phi) := \max_{\ell=1,\ldots,L} \max\left\{\max_{i,j} |(A_\ell)_{ij}|, \|b_\ell\|_\infty\right\}$.*

We next formalize the concept of network weight quantization.

Definition 4.2.2 (Quantization [Elb+21]). *Let $m \in \mathbb{N}_+$ and $\epsilon \in (0, 1/2)$. The network Φ is said to have (m, ϵ)-quantized weights if $\mathcal{K}(\Phi) \subset \left(2^{-m\lceil \log(\epsilon^{-1})\rceil}\mathbb{Z}\right) \cap [-\epsilon^{-m}, \epsilon^{-m}]$. Moreover, for $a \in \mathbb{R}$, we define the (m, ϵ)-quantization mapping rounding a to an integer multiple of $2^{-m\lceil \log(\epsilon^{-1})\rceil}$ according to*

$$\mathcal{Q}_{m,\epsilon}(a) = \left\lceil \frac{a}{2^{-m\lceil \log(\epsilon^{-1})\rceil}} \right\rceil \cdot 2^{-m\lceil \log(\epsilon^{-1})\rceil}.$$

Every quantized ReLU network can be represented by a bitstring (see 4.A) specifying the topology of the network along with its quantized weights. The mapping $\mathcal{D}_{\mathcal{N}}$ taking this bitstring back to the quantized ReLU network is referred to as the *canonical neural network decoder*.

Remark 4.2.3. *For every ReLU network Φ with (m, ϵ)-quantized weights, there is a bitstring $\mathbf{b}$ of length no more than $C_0 m \log(\epsilon^{-1}) \mathcal{M}(\Phi) \log(\mathcal{M}(\Phi))$ such that $\mathcal{D}_{\mathcal{N}}(\mathbf{b}) = \Phi$, with $C_0 > 0$ a universal constant. This follows by upper-bounding* (4.61) *in 4.A.*

As we consider the approximation of sequence-to-sequence mappings ($\mathbb{R}^{\mathbb{Z}} \to \mathbb{R}^{\mathbb{Z}}$), feedforward networks are not directly applicable because they effect mappings between finite-dimensional spaces, concretely from $\mathbb{R}^{N_0}$ to $\mathbb{R}^{N_L}$. However, and perhaps surprisingly, simply applying feedforward networks iteratively in a judicious manner turns out to be sufficient for approximating interesting classes of nonlinear sequence-to-sequence mappings in Kolmogorov-Donoho-optimal manner. Concretely, this is effected through recurrent neural networks, defined as follows.

Definition 4.2.4 (Recurrent neural networks [Pas+14; HGB22])**.** *For $m \in \mathbb{N}_+$, let $\Phi \in \mathcal{N}_{m+1,m+1}$ be a ReLU network of depth $\mathcal{L}(\Phi) \geq 2$. The recurrent neural network (RNN) associated with Φ is the operator $\mathcal{R}_\Phi : \mathbb{R}^{\mathbb{N}_0} \to \mathbb{R}^{\mathbb{N}_0}$ mapping input sequences $(x[t])_{t \in \mathbb{N}_0}$ in $\mathbb{R}$ to output sequences $(y[t])_{t \in \mathbb{N}_0}$ in $\mathbb{R}$ according to*

$$\begin{pmatrix} y[t] \\ h[t] \end{pmatrix} = \Phi\left(\begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix}\right), \ \forall t \in \mathbb{N}_0,$$

where $h[t] \in \mathbb{R}^m$ is the hidden state vector sequence with initial state $h[-1] = 0_m$. We denote the set of all RNNs by $\mathcal{N}^R$.

Remark 4.2.5. *When unfolded in time, an RNN simply amounts to repeated application of Φ.*

From Definition 4.2.4 it is apparent that an RNN $\mathcal{R}_\Phi$ is fully specified by its associated feedforward network Φ.

Definition 4.2.6. *We define $\mathbf{R}$ as the mapping that takes a ReLU network Φ to its associated RNN $\mathcal{R}_\Phi$ according to Definition 4.2.4.*

Together with the canonical neural network decoder $\mathcal{D}_{\mathcal{N}}$, we thus obtain a decoder taking a bitstring into an RNN as follows.

Definition 4.2.7 (Canonical RNN decoder)**.** *We define the canonical RNN decoder as $\mathcal{D}_{\mathcal{R}} = \mathbf{R} \circ \mathcal{D}_{\mathcal{N}}$, where $\mathbf{R}$ is as in Definition 4.2.6 and $\mathcal{D}_{\mathcal{N}}$ is the canonical neural network decoder.*

The main point of this chapter is to show that the canonical RNN decoder is capable of approximating a wide variety of non-linear sequence-to-sequence mappings ($\mathbb{R}^{\mathbb{Z}} \to \mathbb{R}^{\mathbb{Z}}$) in metric-entropy-optimal manner. Together with the results in [Elb+21], this establishes that ReLU networks optimally approximate a wide range of function classes and sequence-to-sequence mappings.

4.2.2. Metric-entropy optimality

In this section, we define the notion of metric-entropy optimal approximation. Consider a metric space $(\mathcal{X}, \rho)$ and a compact set $\mathcal{C} \subset \mathcal{X}$. Together, $\mathcal{C}$ and ρ determine an approximation task. Specifically, we wish to approximate elements $f \in \mathcal{C}$ to within a prescribed error $\epsilon > 0$ in the metric ρ by elements $\widetilde{f} \in \mathcal{X}$ which can be encoded by finite-length bitstrings $\mathbf{b} \in \{0,1\}^\ell$. To go from bitstrings to elements of $\mathcal{X}$, we define decoder mappings as follows.

Definition 4.2.8. *A decoder $\mathcal{D} : \{0,1\}^* \to \mathcal{X}$ is a mapping from bitstrings of arbitrary length to elements of $\mathcal{X}$.*

We shall frequently want to quantify how well a given decoder $\mathcal{D}$ performs.

Definition 4.2.9. *Given a metric space $(\mathcal{X}, \rho)$, a compact set $\mathcal{C} \subset \mathcal{X}$, and a decoder $\mathcal{D} : \{0,1\}^* \to \mathcal{X}$, we say that $(\mathcal{C}, \rho)$ is representable by $\mathcal{D}$, if for every $\epsilon > 0$ and every $f \in \mathcal{C}$, there exist $\ell \in \mathbb{N}_+$ and a bitstring $\mathbf{b} \in \{0,1\}^\ell$ such that*

$$\rho(\mathcal{D}(\mathbf{b}), f) \leq \epsilon.$$

Furthermore, we set

$$L(\epsilon; \mathcal{D}, \mathcal{C}, \rho) := \min\left\{\ell' \in \mathbb{N}_+ \mid \forall f \in \mathcal{C},\ \exists \ell \leq \ell',\ \exists \mathbf{b} \in \{0,1\}^\ell \text{ s.t. } \rho(\mathcal{D}(\mathbf{b}), f) \leq \epsilon\right\}.$$

Remark 4.2.10. *This setting allows us to fix a decoder $\mathcal{D}$ (e.g., the canonical neural network decoder) and then study how well $\mathcal{D}$ performs on different $(\mathcal{C}, \rho)$. That is, $\mathcal{D}$ does not depend on $\mathcal{C}, \rho, f,$ or ϵ.*

The quantity $L(\epsilon; \mathcal{D}, \mathcal{C}, \rho)$ measures how bit-efficient the decoder $\mathcal{D}$ is in representing $\mathcal{C}$ with respect to ρ. It is now natural to ask what the minimum required number of bits, independently of $\mathcal{D}$, is for representing $\mathcal{C}$ with respect to ρ. The concept of metric entropy [KT93; Wai19] allows to answer this question.

Definition 4.2.11. *Let $(\mathcal{X}, \rho)$ be a metric space and $\mathcal{C} \subset \mathcal{X}$ compact. The set $\{x_1, x_2, \ldots, x_N\} \subset \mathcal{C}$* (respectively $\{x_1, x_2, \ldots, x_N\} \subset \mathcal{X}$) *is an ϵ-covering* (respectively ϵ-net) *for $(\mathcal{C}, \rho)$ if, for each $x \in \mathcal{C}$, there exists an $i \in \{1, 2, \ldots, N\}$ so that $\rho(x, x_i) \leq \epsilon$. The ϵ-covering number $N(\epsilon; \mathcal{C}, \rho)$* (respectively the exterior ϵ-covering number $N^{\text{ext}}(\epsilon; \mathcal{C}, \rho)$) *is the cardinality of a smallest ϵ-covering* (respectively smallest ϵ-net) *for $(\mathcal{C}, \rho)$.*

In general, it is hard to obtain precise expressions for covering numbers. One therefore typically resorts to characterizations of their asymptotic behavior as $\epsilon \to 0$. In [Elb+21], where sets of functions are considered, this is done through the concept of optimal exponents. Here, however, we are concerned with sets of systems, which are much more massive and hence require a refined framework for quantifying the asymptotic behavior of their covering numbers. Thus, inspired by [Zam79, Section II.C], we use the following notions.

Definition 4.2.12 (Order, type, and generalized dimension)**.** *Consider a metric space $(\mathcal{X}, \rho)$ and a compact set $\mathcal{C} \subset \mathcal{X}$. Then, $(\mathcal{C}, \rho)$ is said to be of order $\kappa \in \mathbb{N}_+$ and type $\lambda \in \mathbb{N}_+$ if the quantity*

$$\mathfrak{d} := \limsup_{\epsilon \to 0} \frac{\log^{(\kappa+1)} N^{ext}(\epsilon; \mathcal{C}, \rho)}{\log^{\lambda}(\epsilon^{-1})}$$

is strictly positive and finite. In this case, we call $\mathfrak{d}$ the generalized dimension.

Order, type, and generalized dimension provide measures for the "description complexity" of $(\mathcal{C}, \rho)$ with the order κ the coarsest one. For a given order, the type λ constitutes a finer measure, and for fixed order and type, the generalized dimension $\mathfrak{d}$ is the finest measure [Zam79].

Whenever the optimal exponent according to [Elb+21, Definition IV.1] is well-defined (i.e., strictly positive and finite), the underlying set has order and type equal to one and generalized dimension equal to the inverse of the optimal exponent (Lemma 4.B.1). Based on this insight, we obtain Table 4.1, which lists the generalized dimension for the sets considered in [Elb+21, Table I].

Returning to the previous discussion, we are now able to characterize the minimum number of bits required by any decoder to represent $(\mathcal{C}, \rho)$.

Lemma 4.2.13. *Consider the metric space $(\mathcal{X}, \rho)$, the compact set $\mathcal{C} \subset \mathcal{X}$ of order κ, type λ, and generalized dimension $\mathfrak{d}$, and assume that $(\mathcal{C}, \rho)$ is representable by a decoder $\mathcal{D}$. Then, it holds that*

$$\limsup_{\epsilon \to 0} \frac{\log^{(\kappa)} L(\epsilon; \mathcal{D}, \mathcal{C}, \rho)}{\log^{\lambda}(\epsilon^{-1})} \geq \mathfrak{d}. \tag{4.1}$$

Proof. See 4.C.1. □

It is natural to say that a decoder $\mathcal{D}$ is optimal if it satisfies (4.1) with equality.

Definition 4.2.14. *Consider the metric space $(\mathcal{X}, \rho)$ and the compact set $\mathcal{C} \subset \mathcal{X}$ of order κ and type λ with generalized dimension $\mathfrak{d}$. We say that $(\mathcal{C}, \rho)$ is optimally representable by the decoder $\mathcal{D}$, if $(\mathcal{C}, \rho)$ is representable by $\mathcal{D}$ and*

$$\limsup_{\epsilon \to 0} \frac{\log^{(\kappa)} L(\epsilon; \mathcal{D}, \mathcal{C}, \rho)}{\log^{\lambda}(\epsilon^{-1})} = \mathfrak{d}.$$

We now recall a remarkable universal optimality property of ReLU networks, namely all the function classes listed in Table 4.1 are optimally representable, in the sense of Definition 4.2.14, by the canonical neural network decoder. This is a simple reformulation of the results in [Elb+21]; we provide the details of this reformulation in 4.B. In the present chapter, we establish that RNNs (Definition 4.2.4), with inner ReLU networks, extend this universality to the approximation of a wide range of nonlinear dynamical systems.

	Metric	$\mathcal{C}$		κ	λ	$\mathfrak{d}$
$\{\mathbb{R} \to \mathbb{R}\}$	$L^2([0,1])$	L^2-Sobolev	$\mathcal{U}(W_2^m([0,1]))$	1	1	$1/m$
$\{\mathbb{R} \to \mathbb{R}\}$	$L^2([0,1])$	Hölder	$\mathcal{U}(C^\alpha([0,1]))$	1	1	$1/\alpha$
$\{\mathbb{R} \to \mathbb{R}\}$	$L^2([0,1])$	Bump Algebra	$\mathcal{U}(B_{1,1}^1([0,1]))$	1	1	1
$\{\mathbb{R} \to \mathbb{R}\}$	$L^2([0,1])$	Bounded Variation	$\mathcal{U}(BV([0,1]))$	1	1	1
$\{\mathbb{R}^d \to \mathbb{R}\}$	$L^2(\Omega)$	L^p-Sobolev	$\mathcal{U}(W_p^m(\Omega))$	1	1	$\frac{d}{m}$
$\{\mathbb{R}^d \to \mathbb{R}\}$	$L^2(\Omega)$	Besov	$\mathcal{U}(B_{p,q}^m(\Omega))$	1	1	$\frac{d}{m}$
$\{\mathbb{R}^d \to \mathbb{R}\}$	$L^2(\Omega)$	Modulation	$\mathcal{U}(M_{p,p}^s(\mathbb{R}^d))$	1	1	$(\frac{1}{p} - \frac{1}{2} + \frac{2s}{d})^{-1}$
$\{\mathbb{R}^d \to \mathbb{R}\}$	$L^2(\Omega)$	Cartoon functions	$\mathcal{E}^\beta([-\frac{1}{2}, \frac{1}{2}]^d)$	1	1	$\frac{2(d-1)}{\beta}$

Table 4.1: Generalized dimension for the sets considered in [Elb+21]. Here, $\mathcal{U}(X) = \{f \in X : \|f\|_X \leq 1\}$ denotes the unit ball in the space X and $\Omega \subseteq \mathbb{R}^d$ is a Lipschitz domain.

4.2.3. Lipschitz fading-memory systems

We proceed to characterize the class of dynamical systems considered in this chapter and start by defining their domain.

Definition 4.2.15. *For fixed $D > 0$, we denote the set of admissible input signals by $\mathcal{S} := [-D, D]^{\mathbb{Z}}$, that is, for every $x[\cdot] \in \mathcal{S}$, it holds that $|x[t]| \leq D$, $\forall t \in \mathbb{Z}$.*

The quantity $D > 0$ is taken to be fixed throughout the chapter and the dependence of $\mathcal{S}$ on D is not explicitly indicated.

First, the systems $G : \mathcal{S} \to \mathbb{R}^{\mathbb{Z}}$ under consideration are causal.

Definition 4.2.16 (Causality)**.** *A system $G : \mathcal{S} \to \mathbb{R}^{\mathbb{Z}}$ is causal, if for each $T \in \mathbb{Z}$, for every pair $x, x' \in \mathcal{S}$ with $x[t] = x'[t], \forall t \in \mathbb{Z}$ with $t \leq T$, it holds that $(Gx)[T] = (Gx')[T]$.*

Second, we require time-invariance.

Definition 4.2.17 (Time-invariance)**.** *A system $G : \mathcal{S} \to \mathbb{R}^{\mathbb{Z}}$ is time-invariant, if for every $\tau \in \mathbb{Z}$, it holds that*

$$\mathbf{T}_\tau G x = G \mathbf{T}_\tau x, \qquad \forall x \in \mathcal{S},$$

with the shift operator $\mathbf{T}_\tau : \mathbb{R}^{\mathbb{Z}} \to \mathbb{R}^{\mathbb{Z}}$ defined as $(\mathbf{T}_\tau x)[t] = x[t - \tau]$.

Next, we follow Volterra, who suggested that [Vol59, p. 188] "a first extremely natural postulate is to suppose that the influence of the [input] a long time before the given moment gradually fades out." This property was termed "fading memory" in [BC85], and here we introduce a more quantitative version thereof, namely the concept of "Lipschitz fading memory". The concept is inspired by examples in [Lak95; Alv+14; Mou+16; NX22], which will be discussed in more detail later.

Definition 4.2.18 (Lipschitz fading-memory)**.** *We say that $(w[t])_{t \in \mathbb{N}_0}$ is a weight sequence if it is non-increasing and satisfies $w[t] \in (0, 1], \forall t \in \mathbb{N}_0$, and $\lim_{t \to \infty} w[t] = 0$. A system $G : \mathcal{S} \to \mathbb{R}^{\mathbb{Z}}$ has Lipschitz fading-memory with respect to the weight sequence w if*

$$|(Gx)[t] - (Gy)[t]| \leq \sup_{\tau \geq 0} w[\tau] |(x[t - \tau] - y[t - \tau])|, \quad \forall t \in \mathbb{Z},\ \forall x, y \in \mathcal{S}.$$

The class of Lipschitz fading-memory (LFM) systems considered in the remainder of the chapter can now formally be defined as follows.

Definition 4.2.19 (Lipschitz fading-memory systems)**.** *Given a weight sequence $w[\cdot]$, we define*

$$\mathcal{G}(w) = \{G : \mathcal{S} \to \mathbb{R}^{\mathbb{Z}} \,|\, G \text{ is causal, time-invariant, has Lipschitz fading-memory w.r.t. } w, \text{ and satisfies } (G0)[t] = 0,\ \forall t \in \mathbb{Z}\}.$$

As we will want to approximate LFM systems $G \in \mathcal{G}(w)$ by RNNs, we need a metric that quantifies approximation quality. This metric should take into account that the RNNs we consider start running at time $t = 0$ and will, moreover, be of worst-case nature.

Definition 4.2.20. *Let $\mathcal{S}^+ := \{s \in \mathcal{S} \,|\, s[t] = 0, \forall t < 0\}$. For $G, G' \in \{\mathbb{R}^{\mathbb{Z}} \to \mathbb{R}^{\mathbb{Z}}\}$, we define the metric*

$$\rho_*(G, G') = \sup_{x \in \mathcal{S}^+} \sup_{t \in \mathbb{N}_0} |(Gx)[t] - (G'x)[t]|.$$

We hasten to add that the restriction to one-sided input signals in Definition 4.2.20 and to taking the supremum over $t \in \mathbb{N}_0$ in the output signals does not impact the hardness of the approximation task, as shown by the next result.

Lemma 4.2.21. *Let $(w[t])_{t\in\mathbb{N}_0}$ be a weight sequence. For $G, G' \in \mathcal{G}(w)$, we have*

$$\rho_*(G, G') = \sup_{x\in\mathcal{S}} \sup_{t\in\mathbb{Z}} |(Gx)[t] - (G'x)[t]|.$$

Proof. See 4.C.2. □

We are now ready to formally state the main goal of this chapter, which is to prove that $(\mathcal{G}(w), \rho_*)$ is optimally representable by the canonical RNN decoder in Definition 4.2.7. In fact, we will be seeking a quantitative version of this statement comparing the description complexity of the class $\mathcal{G}(w)$ to that of the RNNs approximating it.

4.3. Metric entropy of LFM systems

In this section, we study the (ϵ-)scaling behavior of $N^{\text{ext}}(\epsilon; \mathcal{G}(w), \rho_*)$ for general weight sequences w. This will be effected by deriving an upper bound on $N^{\text{ext}}(\epsilon; \mathcal{G}(w), \rho_*)$ through the construction of a covering and a matching (in terms of scaling behavior) lower bound by identifying an explicit packing. We first define the concept of packings.

Definition 4.3.1. *Let $(\mathcal{X}, \rho)$ be a metric space and $\mathcal{C} \subset \mathcal{X}$ compact. An ϵ-packing for $(\mathcal{C}, \rho)$ is a set $\{x_1, x_2, \ldots, x_N\} \subset \mathcal{C}$ such that $\rho(x_i, x_j) > \epsilon$, for all distinct i, j. The ϵ-packing number $M(\epsilon; \mathcal{C}, \rho)$ is the cardinality of a largest ϵ-packing for $(\mathcal{C}, \rho)$.*

We shall frequently make use of the following two results relating the packing, covering, and exterior covering numbers.

Lemma 4.3.2 ([KT93], Theorem IV)**.** *Let $(\mathcal{X}, \rho)$ be a metric space and $\mathcal{C} \subset \mathcal{X}$ compact. For all $\epsilon > 0$, we have*

$$M(2\epsilon; \mathcal{C}, \rho) \leq N^{ext}(\epsilon; \mathcal{C}, \rho) \leq N(\epsilon; \mathcal{C}, \rho) \leq M(\epsilon; \mathcal{C}, \rho).$$

Lemma 4.3.3 ([KT93], p. 93)**.** *Let $(\mathcal{X}, \rho_\mathcal{X})$ and $(\mathcal{Y}, \rho_\mathcal{Y})$ be metric spaces and consider the compact sets $\mathcal{C}_\mathcal{X} \subset \mathcal{X}$ and $\mathcal{C}_\mathcal{Y} \subset \mathcal{Y}$. Assume that there exists an isometric isomorphism $f : \mathcal{C}_\mathcal{X} \to \mathcal{C}_\mathcal{Y}$, i.e., f is bijective and for every pair $a, b \in \mathcal{C}_\mathcal{X}$, one has $\rho_\mathcal{Y}(f(a), f(b)) = \rho_\mathcal{X}(a, b)$. Then,*

$$N(\epsilon; \mathcal{C}_\mathcal{X}, \rho_\mathcal{X}) = N(\epsilon; \mathcal{C}_\mathcal{Y}, \rho_\mathcal{Y}) \quad \textit{and} \quad M(\epsilon; \mathcal{C}_\mathcal{X}, \rho_\mathcal{X}) = M(\epsilon; \mathcal{C}_\mathcal{Y}, \rho_\mathcal{Y}).$$

Lemma 4.3.3 will allow us to work with a simplified metric space $(\mathcal{G}_0(w), \rho_0)$ instead of the original one $(\mathcal{G}(w), \rho_*)$. Concretely, we exploit the properties of LFM systems to effect this reduction as follows. First, as LFM systems are causal, their output at time t depends on the history of inputs up to and including time t only. Second, time-invariance implies that the mapping taking the history of the input signal to the current output at time t does not change with t and we can therefore restrict ourselves to $t = 0$ w.l.o.g. Thus, the mapping realized by an LFM system is completely characterized by the response to signals in the set

$$\mathcal{S}^- := \{s \in \mathcal{S} \mid \forall \ell \in \mathbb{N}_+ : s[\ell] = 0\}.$$

Using this insight, the simplified metric space can be defined as follows.

Definition 4.3.4. *Let $(w[t])_{t\in\mathbb{N}_0}$ be a weight sequence. We define the metric space $(\mathcal{G}_0(w), \rho_0)$ according to*

$$\mathcal{G}_0(w) = \{g : \mathcal{S}^- \to \mathbb{R} \mid |g(x) - g(x')| \leq \|x - x'\|_w,\ \forall x, x' \in \mathcal{S}^-, g(0) = 0\}, \tag{4.2}$$

$$\text{where} \quad \|x - x'\|_w \coloneqq \sup_{t\in\mathbb{N}_0} w[t]|(x[-t] - x'[-t])|. \tag{4.3}$$

The metric ρ_0 is given by

$$\rho_0(g, g') = \sup_{x\in\mathcal{S}^-} |g(x) - g'(x)|, \qquad \text{for } g, g' \in \{\mathcal{S}^- \to \mathbb{R}\}. \tag{4.4}$$

Next, we define the projection operator $\mathcal{P} : \mathcal{S} \to \mathcal{S}^-$ as

$$(\mathcal{P}x)[t] = x[t] \cdot \mathbb{1}_{\{t\leq 0\}}$$

and formalize the isometric isomorphism between functionals $g \in \mathcal{G}_0(w)$ and systems $G \in \mathcal{G}(w)$ as follows.

Lemma 4.3.5. *Let $w[\cdot]$ be a weight sequence. The map*

$$\begin{aligned} \mathcal{I} : \mathcal{G}_0(w) &\to \mathcal{G}(w) \\ g &\to G \coloneqq (x \to \{g(\mathcal{P}\mathbf{T}_{-t}x)\}_{t\in\mathbb{Z}}) \end{aligned}$$

is an isometric isomorphism between $(\mathcal{G}_0(w), \rho_0)$ and $(\mathcal{G}(w), \rho_*)$. Furthermore, $N(\epsilon; \mathcal{G}_0(w), \rho_0) = N(\epsilon; \mathcal{G}(w), \rho_*)$ and $M(\epsilon; \mathcal{G}_0(w), \rho_0) = M(\epsilon; \mathcal{G}(w), \rho_*)$, for all $\epsilon > 0$.

Proof. See 4.C.3. □

In the remainder of this section, we first lower-bound $M(\epsilon; \mathcal{G}_0(w), \rho_0)$, then upper-bound $N(\epsilon; \mathcal{G}_0(w), \rho_0)$, and finally use Lemmata 4.3.3 and 4.3.5 to translate these bounds into bounds on $N^{\text{ext}}(\epsilon; \mathcal{G}(w), \rho_*)$. The lower bound is established as follows.

Lemma 4.3.6. *Let $w[\cdot]$ be a weight sequence. The ϵ-packing number of $(\mathcal{G}_0(w), \rho_0)$ satisfies*

$$\log M(\epsilon; \mathcal{G}_0(w), \rho_0) \geq \left(\prod_{\ell=0}^{T} \left\lceil \frac{2Dw[\ell]}{\epsilon} \right\rceil \right) - 1,$$

with $T := \max\{T' \in \mathbb{N}_0 \mid w[T'] > \frac{\epsilon}{2D}\}$.

Proof. The proof is taken from [KT93] and is detailed, for completeness, in 4.C.4. □

To upper-bound $N(\epsilon; \mathcal{G}_0(w), \rho_0)$, we construct an ϵ-net for $(\mathcal{G}_0(w), \rho_0)$. This construction is again inspired by [KT93] but we need to modify it to ensure that the elements of the ϵ-net can efficiently be realized by ReLU networks. To be specific, we employ piecewise linear mappings to approximate LFM systems instead of piecewise constant mappings as considered in [KT93]; this requires significant adjustments to the proof in [KT93, Section 7.2].

We start by introducing the "spike" function $\phi : \mathbb{R}^d \to \mathbb{R}$ considered in [KWI24], and defined as

$$\phi(z) = \max\{1 + \min\{z_1, \ldots, z_d, 0\} - \max\{z_1, \ldots, z_d, 0\}, 0\}. \tag{4.5}$$

An illustration of spike functions for $d = 1$ and $d = 2$ is provided in Figure 4.1. The idea underlying the construction of such "spike" functions can be traced back to [Yar18], where the convex set

$$\mathcal{T} := \{z \in \mathbb{R}^d \mid \max\{z_1, \ldots, z_d, 0\} - \min\{z_1, \ldots, z_d, 0\} \leq 1\}$$

is considered and shown to be the union of $(d+1)!$ simplices, each having 0 as a vertex, given by

$$\{z \in \mathcal{T} \mid \ z_{\sigma(0)} \leq z_{\sigma(1)} \leq \cdots \leq z_{\sigma(d)}\},$$

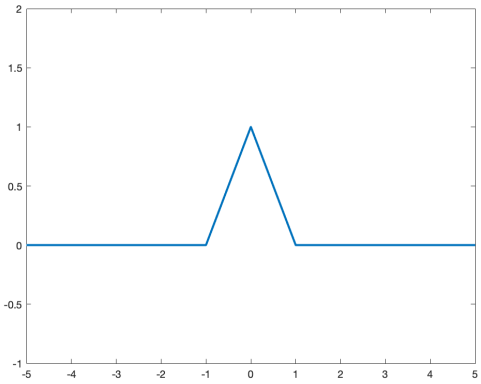
where σ is a permutation of the integers $0, 1, \ldots, d$ and $z_0 := 0$. In [Yar18] this result is employed to approximate continuous functions mapping $\mathbb{R}^d$ to $\mathbb{R}$ by functions that are piecewise linear on the simplices (4.3).

We remark that the spike function (4.5) is a composition of affine functions and min/max functions, which, as shown in Section 4.5, renders it uniquely suitable for realization through ReLU networks. To be specific, in Lemmata 4.C.2 and 4.5.1, we provide concrete constructions of spike functions using ReLU networks. To the best of our knowledge, these constructions are novel.

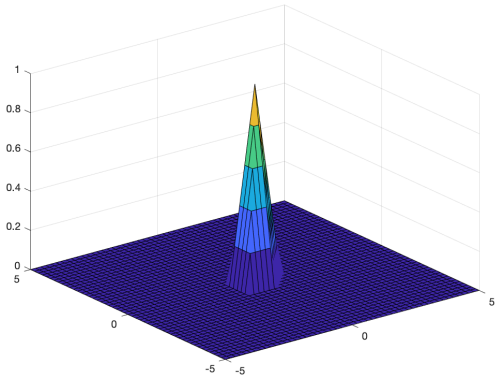
We proceed to show how a *partition of unity* (p.o.u.) can be realized as a weighted linear combination of shifted spike functions, a property that will be of key importance in the RNN constructions described in Section 4.5.

Lemma 4.3.7. *Consider the spike function*

$$\phi(z) = \max\{1 + \min\{z_1, \ldots, z_d, 0\} - \max\{z_1, \ldots, z_d, 0\}, 0\}, \tag{4.6}$$



(a) $d = 1$



(b) $d = 2$

Figure 4.1: Spike functions in dimensions 1 and 2.

the lattice

$$\mathfrak{M} = [\![M_1]\!]^{\pm} \times \cdots \times [\![M_d]\!]^{\pm} \subset \mathbb{R}^d, \text{ where } M_\ell \in \mathbb{N}_+, \text{ for } \ell \in \{1, 2, \ldots, d\},$$

and the set

$$\Xi := \{\phi(\cdot - n)\}_{n \in \mathfrak{M}}.$$

Then, Ξ forms a p.o.u. on $\prod_{\ell=1}^{d}[-M_\ell, M_\ell]$, i.e.,

(i) $0 \leq \phi(z - n) \leq 1$, *for* $z \in \mathbb{R}^d$ *and* $n \in \mathfrak{M}$;

(ii) $\phi(\cdot - n)$ *is compactly supported, specifically* $\operatorname{supp}(\phi(\cdot - n)) \subset n + [-1, 1]^d$;

(iii) it holds that

$$\sum_{n \in \mathfrak{M}} \phi(z - n) = 1, \quad \text{for } z \in \prod_{\ell=1}^{d}[-M_\ell, M_\ell].$$

Proof. To prove (i), we note that $\phi(z) \geq 0$ by definition and

$$1 + \min\{z_1, \ldots, z_d, 0\} - \max\{z_1, \ldots, z_d, 0\} \leq 1.$$

For (ii), it suffices to show that

$$\phi(z) = 0, \quad \text{for } z \in \mathbb{R}^d \setminus [-1, 1]^d.$$

To this end, we pick $z \in \mathbb{R}^d \setminus [-1, 1]^d$ arbitrarily and fix an arbitrary $\ell \in \{1, 2, \ldots, d\}$ such that $|z_\ell| > 1$. Assume that $z_\ell > 1$. (The case $z_\ell < -1$ follows similarly.) Then,

$$1 + \min\{z_1, \ldots, z_d, 0\} - \max\{z_1, \ldots, z_d, 0\} \leq 1 + 0 - z_\ell < 0,$$

which by (4.6) implies $\phi(z) = 0$.

We proceed to prove (iii). Since $\mathrm{supp}(\phi(\cdot - n)) \subset n + [-1, 1]^d$, we have

$$\sum_{n\in\mathfrak{M}} \phi(z-n) = \sum_{n\in\prod_{\ell=1}^{d}\{\lfloor z_\ell\rfloor,\lceil z_\ell\rceil\}} \phi(z-n), \quad \text{for } z \in \prod_{\ell=1}^{d}[-M_\ell, M_\ell].$$

Defining $\bar{z} \in \mathbb{R}^d$ according to $\bar{z}_\ell = \lfloor z_\ell \rfloor$, for $\ell = 1, 2, \ldots, d$, and noting that $\phi(z - n) = \phi((z - \bar{z}) - (n - \bar{z}))$, it suffices to show that

$$\sum_{n\in\{0,1\}^d} \phi(z-n) = 1, \quad \text{for } z \in [0,1]^d \text{ and } n \in \{0,1\}^d.$$

As $\min\{x_1, \ldots, x_d\}$ and $\max\{x_1, \ldots, x_d\}$ are permutation-invariant, so is ϕ by (4.6). In what follows, we can therefore assume, w.l.o.g., that $z_1 \geq z_2 \geq \cdots \geq z_d$.

Now, with e_k the k-th unit vector in $\mathbb{R}^d$, let

$$A := \left\{0, e_1, e_1 + e_2, \ldots, \sum_{i=1}^{k} e_i, \ldots, \sum_{i=1}^{d} e_i\right\} \subset \{0,1\}^d.$$

We claim that

$$\phi(z-n) = 0, \quad \text{for } z \in [0,1]^d \text{ and } n \in \{0,1\}^d \setminus A.$$

This can be verified as follows. First, thanks to

$$n \in \{0,1\}^d \setminus A \;\Leftrightarrow\; \exists\, i, j \in \{1, 2, \ldots, d\}, i < j, \text{ s.t. } n_i = 0, n_j = 1,$$

we get for $z \in [0,1]^d$ and $n \in \{0,1\}^d \setminus A$,

$$\begin{aligned}
&\min\{z_1 - n_1, z_2 - n_2, \ldots, z_d - n_d, 0\} \leq z_j - n_j = z_j - 1,\\
&\max\{z_1 - n_1, z_2 - n_2, \ldots, z_d - n_d, 0\} \geq z_i - n_i = z_i,\\
&\Rightarrow 1 + \min\{z_1 - n_1, z_2 - n_2, \ldots, z_d - n_d, 0\} - \max\{z_1 - n_1, z_2 - n_2, \ldots, z_d - n_d, 0\} \leq 0,
\end{aligned}$$

and hence $\phi(z - n) = 0$. It thus suffices to show that

$$\sum_{n\in A} \phi(z-n) = 1, \quad \text{for } z \in [0,1]^d. \tag{4.7}$$

Now, a direct calculation yields, for $z \in [0,1]^d$,

$$\phi(z) = 1 - z_1, \tag{4.8}$$

$$\phi\left(z - \sum_{i=1}^{k} e_i\right) = z_k - z_{k+1}, \quad \text{for } k = 1, \dots, d-1,$$

$$\phi\left(z - \sum_{i=1}^{d} e_i\right) = z_d. \tag{4.9}$$

Summing (4.8)-(4.9) results in (4.7). □

Next, we establish a traversal property of the lattice $\mathfrak{M}$.

Definition 4.3.8 (Regular path)**.** *For every $d \in \mathbb{N}_+$, $M_\ell \in \mathbb{N}_+$, $\ell \in \{1, \dots, d\}$, and corresponding lattice $\mathfrak{M} = [\![M_1]\!]^{\pm} \times \cdots \times [\![M_d]\!]^{\pm}$, we call a path of lattice points $n_1 \leftrightarrow n_2 \leftrightarrow \cdots \leftrightarrow n_{|\mathfrak{M}|}$ regular for $\mathfrak{M}$ if*

(i) the path visits each lattice point in $\mathfrak{M}$ exactly once,

(ii) n_{i+1} and n_i, for each $i = 1, \dots, |\mathfrak{M}| - 1$, differ in exactly one position, specifically by $+1$ or -1.

Lemma 4.3.9. *For every $d \in \mathbb{N}_+$, $M_\ell \in \mathbb{N}_+$, $\ell \in \{1, \dots, d\}$, and corresponding lattice $\mathfrak{M} = [\![M_1]\!]^{\pm} \times \cdots \times [\![M_d]\!]^{\pm}$, there exists a regular path $n_1 \leftrightarrow n_2 \leftrightarrow \cdots \leftrightarrow n_{|\mathfrak{M}|}$ for $|\mathfrak{M}|$.*

Proof. We write $\mathfrak{M}_d$ for the d-dimensional lattice $[\![M_1]\!]^{\pm} \times \cdots \times [\![M_d]\!]^{\pm}$ to emphasize the dependence on the dimension d and prove the statement by induction over d. The base case $d = 1$ follows by simply considering the path $-M_1 \leftrightarrow -M_1 + 1 \leftrightarrow \cdots \leftrightarrow 0 \leftrightarrow \cdots \leftrightarrow M_1 - 1 \leftrightarrow M_1$ for lattice $\mathfrak{M}_1$. Assume now that the statement holds for $d = k$, i.e., there exists a path $n_1 \leftrightarrow \cdots \leftrightarrow n_{|\mathfrak{M}_k|}$ that is regular for lattice $\mathfrak{M}_k$. Then, for $d = k + 1$, the path $(n_1, -M_{k+1}) \leftrightarrow \cdots \leftrightarrow (n_{|\mathfrak{M}_k|}, -M_{k+1}) \leftrightarrow (n_{|\mathfrak{M}_k|}, -M_{k+1} + 1) \leftrightarrow \cdots \leftrightarrow (n_1, -M_{k+1} + 1) \leftrightarrow (n_1, -M_{k+1} + 2) \leftrightarrow \cdots \leftrightarrow (n_{|\mathfrak{M}_k|}, -M_{k+1} + 2) \leftrightarrow \dots$ is regular for lattice $\mathfrak{M}_{k+1}$. □

We are now ready to describe our construction of the ϵ-net for $(\mathcal{G}_0(w), \rho_0)$. In fact, we shall specify not only the elements of the ϵ-net, but also the mapping taking a given functional $g \in \mathcal{G}_0(w)$ to an ϵ-close element of the net. Counting the number of ball centers needed to ensure that every $g \in \mathcal{G}_0(w)$ is in the ϵ-vicinity of some ball center, then yields the cardinality of the ϵ-net. The mapping proceeds by constructing a functional $\tilde{g}$ which is approximately faithful with respect to g on amplitude-discretized and time-truncated input signals. The formal result is as follows.

Lemma 4.3.10. *For every $\epsilon > 0$ and $s \geq 1$, let*

$$\begin{aligned} T &:= \max\left\{\ell \in \mathbb{N}_0 \;\middle|\; w[\ell] > \frac{\epsilon}{D}\frac{s}{s+1}\right\}, \\ \delta_\ell &:= \frac{s}{s+1}\frac{\epsilon}{w[\ell]}, \qquad \forall \ell \in [\![T]\!], \\ N_\ell &:= \left\lceil \frac{D}{\delta_\ell} \right\rceil, \quad \forall \ell \in [\![T]\!], \end{aligned}$$

and define the mapping

$$\begin{aligned} k : S_- &\to \mathbb{R}^{T+1} \\ k_\ell(x) &= \frac{x[-\ell]}{\delta_\ell}, \qquad \forall \ell \in [\![T]\!]. \end{aligned}$$

Furthermore, consider the lattice

$$\mathfrak{N} := [\![N_0]\!]^{\pm} \times \cdots \times [\![N_T]\!]^{\pm} \subset \mathbb{R}^{T+1}.$$

Then, there exists a set $\mathcal{U}_0 \subset \mathbb{R}^{|\mathfrak{N}|}$ with $|\mathcal{U}_0| = \left(2\left\lceil \frac{s+1}{2}\right\rceil + 1\right)^{|\mathfrak{N}|-1}$ such that

$$\mathcal{U} := \left\{ \tilde{g} : S_- \to \mathbb{R} \;\middle|\; \tilde{g}(x) = \sum_{n\in\mathfrak{N}} \hat{g}_n \phi(k(x) - n),\ \{\hat{g}_n\}_{n\in\mathfrak{N}} \in \mathcal{U}_0 \right\}$$

is an ϵ-net for $(\mathcal{G}_0(w), \rho_0)$. Furthermore, it holds that

$$|\mathcal{U}| = \left(2\left\lceil \frac{s+1}{2}\right\rceil + 1\right)^{|\mathfrak{N}|-1} = \left(2\left\lceil \frac{s+1}{2}\right\rceil + 1\right)^{\left[\prod_{\ell=0}^{T}\left(2\left\lceil \frac{Dw[\ell]}{\epsilon}\frac{s+1}{s}\right\rceil+1\right)\right]-1}.$$

Proof. The proof proceeds in three steps.

Step 1: We pick amplitude-discretized and time-truncated signals from the set $\mathcal{S}^-$ on the lattice $\mathfrak{N}$ and approximately interpolate the functional $g \in \mathcal{G}_0(w)$ on these signals, using the elements of the p.o.u. $\Xi = \{\phi(\cdot - n)\}_{n\in\mathfrak{N}}$ as interpolation basis functions, to get a new functional $\hat{g}$.

Fix $g \in \mathcal{G}_0(w)$ arbitrarily. For each $n \in \mathfrak{N}$, define $\widehat{x}_n \in \mathcal{S}^-$ according to

$$\widehat{x}_n[-\ell] = \begin{cases} \delta_\ell n_\ell, & \text{if } \ell \in [\![T]\!] \\ 0, & \text{if } \ell \in \mathbb{Z} \setminus [\![T]\!] \end{cases} \tag{4.10}$$

and let $\hat{g} : S_- \to \mathbb{R}$ be given by

$$\hat{g}(x) = \sum_{n\in\mathfrak{N}} g(\widehat{x}_n)\phi(k(x) - n). \tag{4.11}$$

We first prove that

$$|\hat{g}(x) - g(x)| \leq \frac{s\epsilon}{s+1}, \quad \text{for } x \in \mathcal{S}^-. \tag{4.12}$$

By Lemma 4.3.7, $\{\phi(\cdot - n)\}_{n\in\mathfrak{N}}$ constitutes a p.o.u. on $\prod_{\ell=0}^{T}[-N_\ell, N_\ell]$, and hence

$$\phi(k(x) - n) = 0, \quad \text{for all } x \in \mathcal{S}^- \text{ s.t. } \|k(x) - n\|_\infty > 1. \tag{4.13}$$

Furthermore, for $x \in S_-$ and $n \in \mathfrak{N}$ such that $\|k(x) - n\|_\infty \leq 1$, we have

$$\left|\frac{x[-\ell]}{\delta_\ell} - n_\ell\right| \leq 1, \quad \text{for } \ell \in [\![T]\!],$$

which gives

$$\begin{aligned} w[\ell]\,|x[-\ell] - \widehat{x}_n[-\ell]| &\leq w[\ell]\delta_\ell = \frac{s\epsilon}{s+1}, \qquad \text{for } \ell \in [\![T]\!], \\ w[\ell]\,|x[-\ell] - \widehat{x}_n[-\ell]| &= w[\ell]\,|x[-\ell]| \\ &\leq \frac{\epsilon}{D}\frac{s}{s+1}D = \frac{s\epsilon}{s+1}, \quad \text{for } \ell > T, \end{aligned}$$

and therefore

$$\|x - \widehat{x}_n\|_w \leq \frac{s\epsilon}{s+1}. \tag{4.14}$$

Hence, we can bound the approximation error $|g(x) - \hat{g}(x)|$, for $x \in \mathcal{S}^-$, according to

$$\begin{aligned} |g(x) - \hat{g}(x)| &\overset{(4.11),\ \text{p.o.u.}}{=} \left|\sum_{n\in\mathfrak{N}} (g(x) - g(\widehat{x}_n))\phi(k(x) - n)\right| \\ &\overset{(4.13)}{=} \left|\sum_{\|k(x)-n\|_\infty\leq 1} (g(x) - g(\widehat{x}_n))\phi(k(x) - n)\right| \\ &\overset{(\phi\geq 0)}{\leq} \sum_{\|k(x)-n\|_\infty\leq 1} |(g(x) - g(\widehat{x}_n))|\,\phi(k(x) - n) \\ &\overset{\text{Lipschitz}}{\leq} \sum_{\|k(x)-n\|_\infty\leq 1} \|x - \widehat{x}_n\|_w\,\phi(k(x) - n) \\ &\overset{(4.14)}{\leq} \frac{s\epsilon}{s+1} \sum_{\|k(x)-n\|_\infty\leq 1} \phi(k(x) - n) \\ &\overset{(4.13)}{=} \frac{s\epsilon}{s+1} \sum_{n\in\mathfrak{N}} \phi(k(x) - n) \\ &\overset{\text{p.o.u.}}{=} \frac{s\epsilon}{s+1}. \end{aligned}$$

Step 2: We modify the interpolation weights $g(\widehat{x}_n)$ in $\hat{g}$ to weights $\hat{g}_n$, for every $n \in \mathfrak{N}$, along a fixed path of $\mathfrak{N}$, to get a new functional $\tilde{g}$ that approximates g to within an error of at most ϵ.

Specifically, we construct, by induction, a functional $\tilde{g}$ such that

$$|\tilde{g}(x) - \hat{g}(x)| \leq \Delta := \frac{\epsilon}{s+1}.$$

To this end, we let

$$\tilde{g}(x) := \sum_{n \in \mathfrak{N}} \hat{g}_n \phi(k(x) - n)$$

and then find, for every $n \in \mathfrak{N}$, a value $\hat{g}_n$ so that

$$|g(\widehat{x}_n) - \hat{g}_n| \leq \Delta, \tag{4.15}$$

which, in turn, yields

$$\begin{aligned} |\hat{g}(x) - \tilde{g}(x)| &= \left| \sum_{n \in \mathfrak{N}} (g(\widehat{x}_n) - \hat{g}_n) \phi(k(x) - n) \right| \\ &\leq \sum_{n \in \mathfrak{N}} |g(\widehat{x}_n) - \hat{g}_n| \, \phi(k(x) - n) \\ &\leq \Delta \sum_{n \in \mathfrak{N}} \phi(k(x) - n) \\ &\overset{\text{p.o.u.}}{=} \Delta. \end{aligned}$$

Therefore,

$$|\tilde{g}(x) - g(x)| \leq |\tilde{g}(x) - \hat{g}(x)| + |\hat{g}(x) - g(x)| \leq \Delta + \frac{s\epsilon}{s+1} = \epsilon.$$

It remains to specify how the values $\hat{g}_n$, $n \in \mathfrak{N}$, can be obtained such that (4.15) holds. This will be done by performing the mapping from $g(\widehat{x}_n)$ to $\hat{g}_n$ along a judiciously chosen path $n_1 \leftrightarrow n_2 \leftrightarrow \cdots \leftrightarrow n_{|\mathfrak{N}|}$ that is regular for $\mathfrak{N}$ (Lemma 4.3.9). To this end, we distinguish two cases.

- *Case 1: The path $n_1 \leftrightarrow n_2 \leftrightarrow \cdots \leftrightarrow n_{|\mathfrak{N}|}$ starts or ends at $0 \in \mathfrak{N}$.*
  In particular, we assume, w.l.o.g., that $n_1 = 0$. Next, we find the values $\hat{g}_{n_k}$ satisfying (4.15) inductively over the index $k = 1, 2 \ldots, |\mathfrak{N}|$. The base case $k = 1$ is immediate as we can simply set $\hat{g}_{n_1} = 0$ and thereby obtain

  $$|g(\widehat{x}_{n_1}) - \hat{g}_{n_1}| \overset{(4.2)}{=} |g(0) - 0| = 0 \leq \Delta.$$

  Now, assume that (4.15) holds for some n_k, $k \in \{1, 2, \ldots, |\mathfrak{N}|\}$, on the path $n_1 \to \cdots \to n_{|\mathfrak{N}|}$. Then, by Lemma 4.3.9, n_{k+1} differs in exactly one position from n_k, namely by $+1$

or -1, which, thanks to (4.3) and (4.10), yields $\left\|\widehat{x}_{n_{k+1}} - \widehat{x}_{n_k}\right\|_w \leq \frac{\epsilon s}{s+1}$. Upon noting that

$$\begin{aligned}\left|g\left(\widehat{x}_{n_{k+1}}\right) - \hat{g}_{n_k}\right| &= \left|\underbrace{g\left(\widehat{x}_{n_{k+1}}\right) - g\left(\widehat{x}_{n_k}\right)}_{\text{Lipschitz}} + \underbrace{g\left(\widehat{x}_{n_k}\right) - \hat{g}_{n_k}}_{(4.15)}\right| \\ &\leq \left\|\widehat{x}_{n_{k+1}} - \widehat{x}_{n_k}\right\|_w + \Delta \\ &\leq \frac{\epsilon s}{s+1} + \Delta = (s+1)\Delta,\end{aligned}$$

we can conclude that there exists an

$$\begin{aligned} &m \in \left\{-2\left\lceil\frac{s+1}{2}\right\rceil, -2\left\lceil\frac{s+1}{2}\right\rceil + 2, \ldots, 2\left\lceil\frac{s+1}{2}\right\rceil\right\}, \\ &\text{s.t.} \quad \hat{g}_{n_{k+1}} = \hat{g}_{n_k} + m\Delta \quad \text{and} \quad \left|\hat{g}_{n_{k+1}} - g(\widehat{x}_{n_{k+1}})\right| \leq \Delta. \end{aligned} \tag{4.16}$$

This completes the induction.

- *Case 2: The path $n_1 \leftrightarrow n_2 \leftrightarrow \cdots \leftrightarrow n_{|\mathfrak{N}|}$ does not start or end at $0 \in \mathfrak{N}$.*
  With $n_i = 0$, for some $i \in \{2, 3, \ldots, |\mathfrak{N}| - 1\}$, and following the spirit of *Case 1*, we split the path $n_1 \leftrightarrow n_2 \leftrightarrow \cdots \leftrightarrow n_{|\mathfrak{N}|}$ into $\text{path}_{\leftarrow} := n_i \to n_{i-1} \to \cdots \to n_1$ and $\text{path}_{\rightarrow} := n_i \to n_{i+1} \to \cdots \to n_{|\mathfrak{N}|}$. The idea is to prove (4.15) by performing induction across $\text{path}_{\leftarrow}$ and $\text{path}_{\rightarrow}$ separately. This can be done following the same procedure as in *Case 1*.

Step 3: Repeat *Steps 1* and *2* for all $g \in \mathcal{G}_0(w)$ and collect all the resulting $\{\hat{g}_n\}_{n\in\mathfrak{N}}$ in the set $\mathcal{U}_0$. For *Case 1* in *Step 2*, we need to store one value for $\hat{g}_{n_1}$ and, according to (4.16), $\left(2\left\lceil\frac{s+1}{2}\right\rceil + 1\right)$ increments or decrements from $\hat{g}_{n_i}$ to $\hat{g}_{n_{i+1}}$, for $i = 0, 1, \ldots, |\mathfrak{N}| - 1$. This yields a total of

$$\left(2\left\lceil\frac{s+1}{2}\right\rceil + 1\right)^{|\mathfrak{N}|-1}$$

values. For *Case 2*, noting that $\text{path}_{\leftarrow}$ and $\text{path}_{\rightarrow}$ are of length $i-1$ and $|\mathfrak{N}| - i$, respectively, and applying the same argument as above, there is again a total of

$$\left(2\left\lceil\frac{s+1}{2}\right\rceil + 1\right)^{i-1}\left(2\left\lceil\frac{s+1}{2}\right\rceil + 1\right)^{|\mathfrak{N}|-i} = \left(2\left\lceil\frac{s+1}{2}\right\rceil + 1\right)^{|\mathfrak{N}|-1}$$

increments or decrements. The set $\mathcal{U}_0$ is hence of cardinality

$$|\mathcal{U}_0| = \left(2\left\lceil\frac{s+1}{2}\right\rceil + 1\right)^{|\mathfrak{N}|-1} = \left(2\left\lceil\frac{s+1}{2}\right\rceil + 1\right)^{\left[\prod_{\ell=0}^{T}\left(2\left\lceil\frac{Dw[\ell]}{\epsilon}\frac{s+1}{s}\right\rceil+1\right)\right]-1}.$$

Finally, setting

$$\mathcal{U} = \left\{ \tilde{g} : S_- \to \mathbb{R} \;\middle|\; \tilde{g}(x) = \sum_{n \in \mathfrak{N}} \hat{g}_n \phi(k(x) - n),\ \{\hat{g}_n\}_{n \in \mathfrak{N}} \in \mathcal{U}_0 \right\},$$

concludes the proof. □

Based on the ϵ-net constructed in Lemma 4.3.10, we can now upper-bound the metric entropy of $(\mathcal{G}_0(w), \rho_0)$ as follows.

Corollary 4.3.11. *For every $s \geq 1$, the covering number of $(\mathcal{G}_0(w), \rho_0)$ satisfies*

$$\log N(\epsilon; \mathcal{G}_0(w), \rho_0) \leq \log\left(2\left\lceil \frac{s+1}{2} \right\rceil + 1\right) \prod_{\ell=0}^{T} \left(2\left\lceil \frac{2Dw[\ell]}{\epsilon} \frac{s+1}{s} \right\rceil + 1\right),$$

where $T := \max\left\{\ell \in \mathbb{N}_0 \mid w[\ell] > \frac{\epsilon}{2D}\frac{s}{s+1}\right\}$.

Proof. For every $\epsilon > 0$, Lemma 4.3.10 delivers an $\frac{\epsilon}{2}$-net for $(\mathcal{G}_0(w), \rho_0)$ with

$$\left(2\left\lceil \frac{s+1}{2} \right\rceil + 1\right)^{\left[\prod_{\ell=0}^{T}\left(2\left\lceil \frac{2Dw[\ell]}{\epsilon}\frac{s+1}{s}\right\rceil+1\right)\right]-1}$$

elements. By Definition 4.2.11, it hence follows that

$$N^{\text{ext}}(\epsilon/2; \mathcal{G}_0(w), \rho_0) \leq \left(2\left\lceil \frac{s+1}{2} \right\rceil + 1\right)^{\left[\prod_{\ell=0}^{T}\left(2\left\lceil \frac{2Dw[\ell]}{\epsilon}\frac{s+1}{s}\right\rceil+1\right)\right]}.$$

Application of Lemma 4.3.2 then yields an upper bound on the covering number according to

$$N(\epsilon; \mathcal{G}_0(w), \rho_0) \leq N^{\text{ext}}(\epsilon/2; \mathcal{G}_0(w), \rho_0) \leq \left(2\left\lceil \frac{s+1}{2} \right\rceil + 1\right)^{\left[\prod_{\ell=0}^{T}\left(2\left\lceil \frac{2Dw[\ell]}{\epsilon}\frac{s+1}{s}\right\rceil+1\right)\right]},$$

which finishes the proof. □

We conclude the developments in this section by deriving a lower bound and an upper bound on the exterior covering number of $(\mathcal{G}(w), \rho_*)$.

Theorem 4.3.12. *The exterior covering number of $(\mathcal{G}(w), \rho_*)$ satisfies*

$$\left(\prod_{\ell=0}^{T'}\left\lceil \frac{Dw[\ell]}{\epsilon} \right\rceil\right) - 1 \leq \log N^{ext}(\epsilon; \mathcal{G}(w), \rho_*) \leq \log(3) \prod_{\ell=0}^{T''}\left(2\left\lceil \frac{4Dw[\ell]}{\epsilon} \right\rceil + 1\right),$$

where $T' := \max\left\{\ell \in \mathbb{N}_0 \mid w[\ell] > \frac{\epsilon}{D}\right\}$ and $T'' := \max\left\{\ell \in \mathbb{N}_0 \mid w[\ell] > \frac{\epsilon}{4D}\right\}$.

Proof. The result follows by noting that

$$\begin{aligned}\left(\prod_{\ell=0}^{T'}\left\lceil\frac{Dw[\ell]}{\epsilon}\right\rceil\right)-1 &\overset{\text{Lemma 4.3.6}}{\leq} \log M(2\epsilon;\mathcal{G}_0(w),\rho_0)\\ &\overset{\text{Lemma 4.3.5}}{=} \log M(2\epsilon;\mathcal{G}(w),\rho_*)\\ &\overset{\text{Lemma 4.3.2}}{\leq} \log N^{\text{ext}}(\epsilon;\mathcal{G}(w),\rho_*)\leq \log N(\epsilon;\mathcal{G}(w),\rho_*)\\ &\overset{\text{Lemma 4.3.5}}{=} \log N(\epsilon;\mathcal{G}_0(w),\rho_0)\\ &\overset{\substack{\text{Corollary 4.3.11}\\ \text{for } s=1}}{\leq} \log(3)\prod_{\ell=0}^{T''}\left(2\left\lceil\frac{4Dw[\ell]}{\epsilon}\right\rceil+1\right).\end{aligned}$$

□

4.4. Metric entropy of ELFM and PLFM systems

We now discuss two specific classes of LFM systems, namely exponentially Lipschitz fading-memory (ELFM) and polynomially Lipschitz fading-memory (PLFM) systems. Specifically, we characterize the description complexity of these two classes by computing their type, order, and generalized dimension. The corresponding results will then serve as a reference for the RNN approximations in Section 4.5. Specifically, we will establish, in Section 4.6, that RNNs can approximate ELFM and PLFM systems in metric-entropy-optimal manner.

4.4.1. Metric entropy of ELFM systems

The concept of ELFM systems is inspired, inter alia, by applications in finance, such as those discussed in [NX22], where asset pricing decisions are influenced by past observations. Instead of relying solely on finite-length observations, the model integrates infinite past observations by endowing them with exponentially decaying memory. Similar settings are also considered in random walk models [Alv+14; Mou+16]. These examples, when appropriately adapted to the setup in the present chapter, fit into the setting of LFM systems according to Definition 4.2.19, with respect to exponentially decaying weight sequences. We formally define exponentially decaying memory and the corresponding ELFM systems as follows.

Definition 4.4.1. *For $a \in (0,1]$ and $b > 0$, let*

$$w^{(e)}_{a,b}[t] := ae^{-bt}, \qquad \text{for all } t \in \mathbb{N}_0.$$

An LFM system with respect to $\{w^{(e)}_{a,b}[t]\}_{t\in\mathbb{N}_0}$ is said to be exponentially Lipschitz fading-memory (ELFM). We write $\mathcal{G}(w^{(e)}_{a,b})$ for the class of systems that are ELFM with respect to the weight sequence $w^{(e)}_{a,b}[t]$.

The remainder of this section is devoted to computing the order, type, and generalized dimension of $(\mathcal{G}(w_{a,b}^{(e)}), \rho_*)$. To this end, we first establish an auxiliary result.

Lemma 4.4.2. *Let $a \in (0,1]$, $b, c, d > 0$ and consider the weight sequence $w_{a,b}^{(e)}$ as per Definition 4.4.1. Let*

$$T := \max\left\{ t \in \mathbb{N}_0 \;\middle|\; w_{a,b}^{(e)}[t] > \frac{\epsilon}{d} \right\}.$$

Then,

$$\log\left(\prod_{\ell=0}^{T} \frac{c w_{a,b}^{(e)}[\ell]}{\epsilon} \right) = \frac{1}{2b\log(e)} \log^2(\epsilon^{-1}) + o\left(\log^2(\epsilon^{-1})\right).$$

Proof. See 4.C.5. □

We are now ready to state the main result of this section quantifying the massiveness of the class of ELFM systems.

Lemma 4.4.3. *Let $a \in (0,1]$ and $b > 0$. The class of ELFM systems $(\mathcal{G}(w_{a,b}^{(e)}), \rho_*)$ is of order 1 and type 2, with generalized dimension*

$$\mathfrak{d} = \frac{1}{2b\log(e)}.$$

Proof. Consider $\epsilon \in (0, \epsilon_0)$ with $\epsilon_0 = \frac{D w_{a,b}^{(e)}[0]}{2} = \frac{aD}{2}$. By Theorem 4.3.12, the exterior covering number of $(\mathcal{G}(w_{a,b}^{(e)}), \rho_*)$ satisfies

$$\left(\prod_{\ell=0}^{T'} \left\lceil \frac{D w_{a,b}^{(e)}[\ell]}{\epsilon} \right\rceil \right) - 1 \leq \log N^{\text{ext}}(\epsilon; \mathcal{G}(w_{a,b}^{(e)}), \rho_*) \leq \log(3) \prod_{\ell=0}^{T''} \left(2 \left\lceil \frac{4D w_{a,b}^{(e)}[\ell]}{\epsilon} \right\rceil + 1 \right), \tag{4.17}$$

where

$$T' := \max\left\{ \ell \in \mathbb{N}_0 \;\middle|\; w_{a,b}^{(e)}[\ell] > \frac{\epsilon}{D} \right\} \text{ and } T'' := \max\left\{ \ell \in \mathbb{N}_0 \;\middle|\; w_{a,b}^{(e)}[\ell] > \frac{\epsilon}{4D} \right\}.$$

We can further lower-bound the left-most term in (4.17) according to

$$\begin{aligned} \left(\prod_{\ell=0}^{T'} \left\lceil \frac{D w_{a,b}^{(e)}[\ell]}{\epsilon} \right\rceil \right) - 1 &\geq \left(\prod_{\ell=0}^{T'} \frac{D w_{a,b}^{(e)}[\ell]}{\epsilon} \right) - 1 \\ &\geq \frac{1}{2} \prod_{\ell=0}^{T'} \frac{D w_{a,b}^{(e)}[\ell]}{\epsilon}, \end{aligned} \tag{4.18}$$

where (4.18) follows from

$$\frac{1}{2} \prod_{\ell=0}^{T'} \frac{D w_{a,b}^{(e)}[\ell]}{\epsilon} \geq 1,$$

which, in turn, is a consequence of

$$\frac{Dw_{a,b}^{(e)}[0]}{\epsilon} \geq \frac{Dw_{a,b}^{(e)}[0]}{\epsilon_0} = 2,$$
$$\frac{Dw_{a,b}^{(e)}[\ell]}{\epsilon} \geq \frac{Dw_{a,b}^{(e)}[T']}{\epsilon} > 1, \quad \text{for} \quad \ell \in [\![T']\!] \setminus \{0\}.$$

Similarly, we can further upper-bound the right-most term in (4.17) according to

$$\begin{aligned}\log(3) \prod_{\ell=0}^{T''} \left(2 \left\lceil \frac{4Dw_{a,b}^{(e)}[\ell]}{\epsilon} \right\rceil + 1 \right) &\leq \log(3) \prod_{\ell=0}^{T''} \left(\frac{8Dw_{a,b}^{(e)}[\ell]}{\epsilon} + 3 \right) \\ &\leq \log(3) \prod_{\ell=0}^{T''} \frac{20Dw_{a,b}^{(e)}[\ell]}{\epsilon}, \end{aligned} \tag{4.19}$$

where (4.19) follows from

$$\frac{4Dw_{a,b}^{(e)}[\ell]}{\epsilon} \geq \frac{4Dw_{a,b}^{(e)}[T'']}{\epsilon} > 1, \quad \text{for} \quad \ell \in [\![T'']\!].$$

Combining (4.17)–(4.19), taking logarithms one more time, and dividing by $\log^2(\epsilon^{-1})$, yields

$$\begin{aligned}\frac{\log\left(\prod_{\ell=0}^{T'} \frac{Dw_{a,b}^{(e)}[\ell]}{\epsilon}\right) - 1}{\log^2(\epsilon^{-1})} &\leq \frac{\log^{(2)} N^{\text{ext}}(\epsilon; \mathcal{G}(w_{a,b}^{(e)}), \rho_*)}{\log^2(\epsilon^{-1})} \\ &\leq \frac{\log\left(\prod_{\ell=0}^{T''} \frac{20Dw_{a,b}^{(e)}[\ell]}{\epsilon}\right) + \log^{(2)}(3)}{\log^2(\epsilon^{-1})}. \end{aligned}$$

Taking the limit $\epsilon \to 0$ and applying Lemma 4.4.2 to the lower and the upper bound in (4.4.1), we obtain

$$\lim_{\epsilon \to 0} \frac{\log^{(2)} N^{\text{ext}}(\epsilon; \mathcal{G}(w_{a,b}^{(e)}), \rho_*)}{\log^2(\epsilon^{-1})} = \frac{1}{2b \log(e)},$$

which implies that $(\mathcal{G}(w_{a,b}^{(e)}), \rho_*)$ is of order 1 and type 2, with generalized dimension

$$\mathfrak{d} = \frac{1}{2b \log(e)}.$$

This concludes the proof. □

The generalized dimension being inversely proportional to b reflects that faster memory decay rates, i.e., larger b, result in system classes that are less massive. Additionally, we note that the description complexity of ELFM systems is primarily determined by the order being equal to 1 and the type equal to 2. Compared to the function classes in Table 4.1, which are all

of order 1 and type 1, this shows that the class of ELFM systems is significantly more massive than unit balls in function spaces.

4.4.2. Metric entropy of PLFM systems

Next, we consider systems with polynomially decaying memory, a concept used, e.g., in the context of PDEs [MM09; FP02]. Specifically, the references [MM09; FP02] are concerned with Volterra integro-differential equations [Lak95] of polynomially decaying memory kernels. These examples, when suitably adjusted to the framework in the present chapter, align with dynamical systems that are LFM with respect to polynomially decaying weight sequences. We formalize the concept of polynomially decaying memory and polynomially Lipschitz fading-memory (PLFM) systems as follows.

Definition 4.4.4. *For $q \in (0,1]$ and $p > 0$, let*

$$w_{p,q}^{(p)}[t] := \frac{q}{(1+t)^p}, \qquad \text{for all } t \in \mathbb{N}_0.$$

An LFM system with respect to $\{w_{p,q}^{(p)}[t]\}_{t\in\mathbb{N}_0}$ is said to be polynomially Lipschitz fading-memory (PLFM). We write $\mathcal{G}(w_{p,q}^{(p)})$ for the class of systems that are PLFM with respect to the weight sequence $w_{p,q}^{(p)}[t]$.

As in the previous section, we first need an auxiliary result.

Lemma 4.4.5. *Let $q \in (0,1]$, $p, c, d > 0$ and consider $w_{p,q}^{(p)}$ as per Definition 4.4.4. Let*

$$T := \max\left\{ t \in \mathbb{N}_0 \;\middle|\; w_{p,q}^{(p)}[t] > \frac{\epsilon}{d} \right\}.$$

Then,

$$\log\left(\prod_{\ell=0}^{T} \frac{c w_{p,q}^{(p)}[\ell]}{\epsilon} \right) = \Theta(\epsilon^{-1/p}).$$

Proof. See 4.C.6. □

We obtain the order, type, and generalized dimension of PLFM systems as follows.

Lemma 4.4.6. *Let $q \in (0,1]$ and $p > 0$. The class of PLFM systems $(\mathcal{G}(w_{p,q}^{(p)}), \rho_*)$ is of order 2 and type 1, with generalized dimension*

$$\mathfrak{d} = \frac{1}{p}.$$

Proof. See 4.C.7. □

Compared to the class of ELFM systems, which exhibits order 1, the class of PLFM systems is more massive as it has order 2. This is intuitively meaningful as polynomial decay is significantly slower than exponential decay, rendering PLFM systems to depend more strongly on past inputs. Additionally, the generalized dimension exhibiting inverse proportionality to p, reflects that faster (polynomial) decay, i.e., larger p, leads to smaller description complexity.

4.5. Approximating LFM systems through RNNs

We now proceed to realize the elements in the ϵ-net for $(\mathcal{G}_0(w), \rho_0)$ constructed in Lemma 4.3.10 by ReLU networks. Based on the connection between $(\mathcal{G}_0(w), \rho_0)$ and $(\mathcal{G}(w), \rho_*)$ identified in Section 4.3, this will then allow us to construct RNNs approximating general LFM systems.

4.5.1. Approximating $(\mathcal{G}_0(w), \rho_0)$ with ReLU networks

The building block in Lemma 4.3.10 for approximating functionals in $\mathcal{G}_0(w)$ is the p.o.u. $\Xi = \{\phi(\cdot - n)\}_{n\in\mathfrak{M}}$, which is why we first focus on constructing ReLU networks realizing Ξ. As the elements of Ξ are shifted versions of the spike function ϕ and shifts, by virtue of being affine transformations, are trivially realized by single-layer ReLU networks, it suffices to find a ReLU network realization of ϕ. Note that this argument made use of the fact that compositions of ReLU networks are again ReLU networks (see Lemma 4.C.3).

Lemma 4.5.1. *For $d \in \mathbb{N}_+$, consider the spike function $\phi : \mathbb{R}^d \to \mathbb{R}$,*

$$\phi(z) = \max\{1 + \min\{z_1, \dots, z_d, 0\} - \max\{z_1, \dots, z_d, 0\}, 0\}.$$

There exists a ReLU network $\Phi \in \mathcal{N}_{d,1}$ with $\mathcal{L}(\Phi) = \lceil \log(d+1) \rceil + 4$, $\mathcal{M}(\Phi) \leq 60d - 28$, $\mathcal{W}(\Phi) \leq 6d$, $\mathcal{K}(\Phi) = \{1, -1\}$, and $\mathcal{B}(\Phi) = 1$, such that

$$\Phi(z) = \phi(z), \qquad \text{for all} \quad z \in \mathbb{R}^d.$$

Proof. Let Φ_d^{max} be the ReLU network realization of $\max\{z_1, z_2, \dots, z_d\}$ according to Lemma 4.C.2. Then, using $\min\{z_1, z_2, \dots, z_d\} = -\max\{-z_1, -z_2, \dots, -z_d\}$, we obtain

$$\begin{aligned} \phi(z) &= \rho(1 - \rho(\Phi_d^{max}(-z)) - \rho(\Phi_d^{max}(z))) \\ &= (\underbrace{(W_3 \circ \rho \circ W_2 \circ \rho \circ P(\Phi_d^{max}, \Phi_d^{max}))}_{=:\Phi_2} \circ \underbrace{W_1}_{=:\Phi_1})(z), \end{aligned} \tag{4.20}$$

where $W_1(z) = \begin{pmatrix} \mathbb{I}_d & -\mathbb{I}_d \end{pmatrix}^T z$, $W_2(z) = \begin{pmatrix} -1 & -1 \end{pmatrix} z + 1$, $W_3(z) = z$, and $P(\Phi_d^{max}, \Phi_d^{max})$ is the parallelization of two Φ_d^{max}-networks according to Lemma 4.C.4 such that $P(\Phi_d^{max}, \Phi_d^{max})(z) = (\Phi_d^{max}(z), \Phi_d^{max}(z))^T$. Now, by Lemma 4.C.3, there exists a ReLU network Φ realizing $\Phi_2 \circ \Phi_1$ in

(4.20), with

$$\begin{aligned}
\mathcal{L}(\Phi) &\overset{\text{Lemma 4.C.3}}{=} \mathcal{L}(\Phi_2) + \mathcal{L}(\Phi_1) \\
&\overset{\text{Lemma 4.C.2, 4.C.4}}{=} \lceil \log(d+1) \rceil + 4, \\
\mathcal{M}(\Phi) &\overset{\text{Lemma 4.C.3}}{\leq} 2\mathcal{M}(\Phi_1) + 2\mathcal{M}(\Phi_2) \\
&\overset{\text{Lemma 4.C.4}}{=} 2\mathcal{M}(W_3) + 2\mathcal{M}(W_2) + 2\mathcal{M}(W_1) + 4\mathcal{M}(\Phi_d^{max}) \\
&\overset{\text{Lemma 4.C.2}}{\leq} 60d - 28, \\
\mathcal{W}(\Phi) &\overset{\text{Lemma 4.C.3}}{\leq} \max\{4d, \max\{\mathcal{W}(\Phi_2), \mathcal{W}(\Phi_1)\}\} \\
&\overset{\text{Lemma 4.C.4}}{\leq} \max\{4d, 2\mathcal{W}(\Phi_d^{max}), \mathcal{W}(W_3), \mathcal{W}(W_2), \mathcal{W}(W_1)\} \\
&\overset{\text{Lemma 4.C.2}}{\leq} 6d, \\
\mathcal{K}(\Phi) &\overset{\text{Lemma 4.C.3}}{\subset} \mathcal{K}(\Phi_2) \cup (-\mathcal{K}(\Phi_2)) \cup \mathcal{K}(\Phi_1) \cup (-\mathcal{K}(\Phi_1)) \\
&\overset{\text{Lemma 4.C.3,4.C.4}}{=} \left(\bigcup_{i=1}^{3} \mathcal{K}(W_i) \cup (-\mathcal{K}(W_i)) \right) \cup \mathcal{K}(\Phi_d^{max}) \cup (-\mathcal{K}(\Phi_d^{max})) \\
&\overset{\text{Lemma 4.C.2}}{=} \{1, -1\}, \\
\mathcal{B}(\Phi) &= \max_{b \in \mathcal{K}(\Phi)} |b| = 1.
\end{aligned}$$

This concludes the proof. □

We now show how the elements of the ϵ-net constructed in Lemma 4.3.10 can be realized by ReLU networks. In particular, our construction will be found, in Section 4.6, to be encodable by bitstrings of length scaling (in ϵ) in a metric-entropy optimal manner.

Lemma 4.5.2. *For every $s \geq 1$, there exists $\epsilon_0 > 0$, such that for $\epsilon \in (0, \epsilon_0)$, with*

$$T := \max\left\{ \ell \in \mathbb{N}_0 \;\middle|\; w[\ell] > \frac{\epsilon}{D} \frac{s}{s+1} \right\},$$

for every $g \in \mathcal{G}_0(w)$, there exists a $\Phi \in \mathcal{N}_{T+1,1}$ with $(2, \epsilon)$-quantized weights (Definition 4.2.2) satisfying

$$\left| \Phi\left(\{x[-\ell]\}_{\ell=0}^{T} \right) - g(x) \right| \leq \epsilon, \qquad \text{for all } x \in \mathcal{S}^-.$$

Moreover,

$$\mathcal{L}(\Phi) = \lceil \log(T+2) \rceil + 6 \quad \text{and} \quad \mathcal{M}(\Phi) \leq 244(T+1) \prod_{\ell=0}^{T} \left(\frac{2Dw[\ell]}{\epsilon} \frac{s+1}{s} + 4 \right).$$

Proof. Fix $g \in \mathcal{G}_0(w)$. To construct a $(2, \epsilon)$-quantized ReLU network approximating g, we follow

the spirit of the proof of Lemma 4.3.10 and first consider

$$\hat{g}(x) = \sum_{n\in\mathfrak{N}} g(\widehat{x}_n)\phi\left(\left\{\frac{x[-\ell]}{\delta_\ell} - n_\ell\right\}_{\ell=0}^{T}\right),$$

with

$$\begin{aligned}
\delta_\ell &:= \frac{s}{s+1}\frac{\epsilon}{w[\ell]}, \qquad \forall \ell \in [\![T]\!], \\
N_\ell &:= \left\lceil \frac{D}{\delta_\ell} \right\rceil, \quad \forall \ell \in [\![T]\!], \\
\mathfrak{N} &:= [\![N_0]\!]^{\pm} \times \cdots \times [\![N_T]\!]^{\pm} \subset \mathbb{R}^{T+1}, \\
\widehat{x}_n[-\ell] &:= \begin{cases} \delta_\ell n_\ell, & \text{if } \ell \in [\![T]\!], \\ 0, & \text{else.} \end{cases}
\end{aligned} \tag{4.21}$$

It was shown in (4.12) that

$$|\hat{g}(x) - g(x)| \leq \frac{s\epsilon}{s+1}, \quad \text{for } x \in \mathcal{S}^{-}.$$

We next quantize the parameters δ_ℓ^{-1} and the lattice $\mathfrak{N}$ in (4.21) according to

$$\begin{aligned}
\widetilde{\delta}_\ell^{-1} &:= \mathcal{Q}_{2,\epsilon}(\delta_\ell^{-1}), \\
\widetilde{N}_\ell &= \left\lceil \frac{D}{\widetilde{\delta}_\ell} \right\rceil, \quad \ell \in [\![T]\!], \\
\widetilde{\mathfrak{N}} &= \left[\!\left[\widetilde{N}_0\right]\!\right]^{\pm} \times \cdots \times \left[\!\left[\widetilde{N}_T\right]\!\right]^{\pm},
\end{aligned} \tag{4.22}$$

and adjust the grid points $\widehat{x}_n$ to

$$\widetilde{x}_n[-\ell] = \begin{cases} \widetilde{\delta}_\ell n_\ell, & \text{if } \ell \in [\![T]\!], \\ 0, & \text{else.} \end{cases}$$

Furthermore, we quantize $g(\widetilde{x}_n)$ according to

$$\widetilde{g}_n := \mathcal{Q}_{2,\epsilon}(g(\widetilde{x}_n)) \tag{4.23}$$

and consider the function

$$\tilde{g}(x) = \sum_{n\in\widetilde{\mathfrak{N}}} \widetilde{g}_n \phi\left(\left\{\frac{x[-\ell]}{\widetilde{\delta}_\ell} - n_\ell\right\}_{\ell=0}^{T}\right) =: f\left(\{x[-\ell]\}_{\ell=0}^{T}\right). \tag{4.24}$$

For ease of notation, we define $\tilde{k} : S_{-} \to \mathbb{R}^{T+1}$ as $\tilde{k}_\ell(x) = \widetilde{\delta}_\ell^{-1} x[-\ell]$, for $\ell \in [\![T]\!]$. Next, we show that

$$|\tilde{g}(x) - g(x)| \leq \epsilon. \tag{4.25}$$

This follows from

$$|\tilde{g}(x) - g(x)| = \left| \sum_{\|\tilde{k}(x)-n\|_\infty \leq 1} (g(x) - \widetilde{g}_n)\phi(\tilde{k}(x) - n) \right| \tag{4.26}$$

$$\leq \sum_{\|\tilde{k}(x)-n\|_\infty \leq 1} \left(|(g(x) - g(\widetilde{x}_n))| + |(g(\widetilde{x}_n) - \widetilde{g}_n)|\right) \phi(\tilde{k}(x) - n)$$

$$\leq \sum_{\|\tilde{k}(x)-n\|_\infty \leq 1} \left(\|x - \widetilde{x}_n\|_w + \frac{\epsilon}{s+1} \right) \phi(\tilde{k}(x) - n) \tag{4.27}$$

$$\leq \epsilon \sum_{\|\tilde{k}(x)-n\|_\infty \leq 1} \phi(\tilde{k}(x) - n) \tag{4.28}$$

$$\leq \epsilon. \tag{4.29}$$

Here, (4.26) and (4.29) are by the p.o.u. property of ϕ and (4.27) is a consequence of the Lipschitz property of g according to (4.2) and

$$|g(\widetilde{x}_n) - \widetilde{g}_n| = |g(\widetilde{x}_n) - \mathcal{Q}_{2,\epsilon}(g(\widetilde{x}_n))|$$
$$\leq 2^{-2\lceil \log(\epsilon^{-1}) \rceil} \overset{\epsilon \in (0,\epsilon_0),(4.111)}{\leq} \frac{\epsilon}{s+1}.$$

Further, (4.28) follows from the fact that for $\|\tilde{k}(x) - n\|_\infty \leq 1$,

$$\left| \frac{x[-\ell]}{\widetilde{\delta}_\ell} - n_\ell \right| \leq 1, \quad \text{for } \ell \in [\![T]\!],$$

and hence

$$\|x - \widetilde{x}_n\|_w \leq \max_{\ell \in [\![T]\!]} \widetilde{\delta}_\ell w[\ell] \leq \frac{s\epsilon}{s+1}, \tag{4.30}$$

where the second inequality in (4.30) is by

$$\widetilde{\delta}_\ell \overset{(4.22)}{=} \left(\mathcal{Q}_{2,\epsilon} \left(\delta_\ell^{-1} \right) \right)^{-1} \leq \delta_\ell = \frac{s}{s+1} \frac{\epsilon}{w[\ell]}.$$

Based on (4.24), we can rewrite (4.25) according to

$$\left| f\left(\{x[-\ell]\}_{\ell=0}^T \right) - g(x) \right| \leq \epsilon, \qquad \text{for } x \in S_-. \tag{4.31}$$

It hence suffices to construct a ReLU network Φ that realizes f, which then, thanks to (4.31), approximates g to within an error of at most ϵ. To this end, assume that $n^1, n^2, \ldots, n^{|\widetilde{\mathfrak{N}}|}$ is an arbitrary, but fixed, enumeration of the elements of $\widetilde{\mathfrak{N}}$. Set $\widehat{W}^\Sigma(x) = \Lambda x$ and $\widehat{W}_{n^i}(x) = \hat{B}x + \hat{b}_{n^i}$,

with

$$\begin{aligned}\Lambda &= \begin{pmatrix}\widetilde{g}_{n^1} & \widetilde{g}_{n^2} & \dots & \widetilde{g}_{n^{|\widetilde{\mathfrak{N}}|}}\end{pmatrix},\\ \widehat{B} &= \operatorname{diag}\left(\widetilde{\delta}_0^{-1}, \widetilde{\delta}_1^{-1}, \dots, \widetilde{\delta}_T^{-1}\right),\\ \hat{b}_{n^i} &= \begin{pmatrix}-n_0^i & -n_1^i & \dots & -n_T^i\end{pmatrix}^T.\end{aligned} \tag{4.32}$$

Moreover, let Ψ be a ReLU network realizing the spike function ϕ according to Lemma 4.5.1 and define the following ReLU networks

$$\begin{aligned}\Phi_2 &= \widetilde{W}^{\Sigma},\\ \Phi_{1,2}^{n^i} &= \Psi, \qquad \text{for } i = 1, \dots, |\widetilde{\mathfrak{N}}|,\\ \Phi_{1,1}^{n^i} &= \widehat{W}_{n^i}, \qquad \text{for } i = 1, \dots, |\widetilde{\mathfrak{N}}|.\end{aligned} \tag{4.33}$$

Next, we use Lemma 4.C.3 to compose $\Phi_{1,2}^{n^i}$ and $\Phi_{1,1}^{n^i}$ in order to realize the shifted versions of the spike function according to

$$\left(\Phi_{1,2}^{n^i} \circ \Phi_{1,1}^{n^i}\right)\left(\{x[-\ell]\}_{\ell=0}^T\right) = \phi\left(\left\{\frac{x[-\ell]}{\widetilde{\delta}_\ell} - n_\ell^i\right\}_{\ell=0}^T\right), \quad \text{for } i = 1, 2, \dots, |\widetilde{\mathfrak{N}}|.$$

Then, we apply Lemma 4.C.4 to construct a ReLU network as the parallelization of the compositions $\Phi_{1,2}^{n^i} \circ \Phi_{1,1}^{n^i}$, for $i = 1, 2, \dots, |\widetilde{\mathfrak{N}}|$,

$$\Phi_1 := P\left(\left(\Phi_{1,2}^{n^1} \circ \Phi_{1,1}^{n^1}\right), \left(\Phi_{1,2}^{n^2} \circ \Phi_{1,1}^{n^2}\right), \dots, \left(\Phi_{1,2}^{n^{|\widetilde{\mathfrak{N}}|}} \circ \Phi_{1,1}^{|\widetilde{\mathfrak{N}}|}\right)\right). \tag{4.34}$$

Finally, we use Lemma 4.C.3 again to get a ReLU network that composes Φ_2 and Φ_1 according to

$$\Phi = \Phi_2 \circ \Phi_1 = \Phi_2 \circ P\left(\left(\Phi_{1,2}^{n^1} \circ \Phi_{1,1}^{n^1}\right), \left(\Phi_{1,2}^{n^2} \circ \Phi_{1,1}^{n^2}\right), \dots, \left(\Phi_{1,2}^{n^{|\widetilde{\mathfrak{N}}|}} \circ \Phi_{1,1}^{|\widetilde{\mathfrak{N}}|}\right)\right), \tag{4.35}$$

thereby realizing the linear combination

$$\sum_{n \in \widetilde{\mathfrak{N}}} \widetilde{g}_n \phi\left(\left\{\frac{x[-\ell]}{\widetilde{\delta}_\ell} - n_\ell\right\}_{\ell=0}^T\right) \overset{(4.24)}{=} f\left(\{x[-\ell]\}_{\ell=0}^T\right).$$

To conclude the proof, we verify that Φ, indeed, has $(2, \epsilon)$-quantized weights, compute $\mathcal{L}(\Phi)$, and derive an upper bound on $\mathcal{M}(\Phi)$. We defer the corresponding details to 4.C.9. □

4.5.2. Approximating $(\mathcal{G}(w), \rho_*)$ with RNNs

Having constructed ReLU networks that realize elements of $\mathcal{G}_0(w)$ according to Lemma 4.5.2, we are now ready to describe the realization of systems in $\mathcal{G}(w)$ through RNNs. This will be done by employing the connection between $\mathcal{G}(w)$ and $\mathcal{G}_0(w)$, as established in Lemma 4.3.5. Specifically, we construct RNNs that suitably remember past inputs and produce approximations of the

desired output.

Theorem 4.5.3. *For every $s \geq 1$, there exists $\epsilon_0 > 0$, such that for $\epsilon \in (0, \epsilon_0)$, with*

$$T := \max\left\{\ell \in \mathbb{N}_0 \;\middle|\; w[\ell] > \frac{\epsilon}{D}\frac{s}{s+1}\right\},$$

for every $G \in \mathcal{G}(w)$, there is an RNN $\mathcal{R}_\Psi$ associated with a ReLU network $\Psi \in \mathcal{N}_{T+1,T+1}$, satisfying

$$\rho_*(\mathcal{R}_\Psi, G) \leq \epsilon.$$

Moreover, $\mathcal{R}_\Psi$ has $(2,\epsilon)$-quantized weights and there exists a universal constant $C > 0$ such that

$$\mathcal{M}(\Psi) \leq C(T+1)^2 \prod_{\ell=0}^{T}\left(\frac{2Dw[\ell]}{\epsilon}\frac{s+1}{s} + 4\right).$$

Proof. Fix $s \geq 1$ and $G \in \mathcal{G}(w)$ arbitrarily. We proceed in two steps.

Step 1: We construct a ReLU network $\Phi : \mathbb{R}^{T+1} \to \mathbb{R}$ such that

$$\sup_{x\in\mathcal{S}^+}\sup_{t\in\mathbb{N}_0}\left|\Phi\left(\{x[t-\ell]\}_{\ell=0}^{T}\right) - G(x)[t]\right| \leq \epsilon. \tag{4.36}$$

To this end, we first note that by Lemma 4.3.5, one can find a $g \in \mathcal{G}_0(w)$ so that

$$g\left(\mathcal{P}\mathbf{T}_{-t}x\right) = G(x)[t], \quad \text{for all } x \in \mathcal{S} \text{ and } t \in \mathbb{Z}. \tag{4.37}$$

Furthermore, by Lemma 4.5.2, there exists $\epsilon_0 > 0$, such that for every $\epsilon \in (0, \epsilon_0)$, there is a ReLU network Φ satisfying

$$\left|\Phi\left(\{z[-\ell]\}_{\ell=0}^{T}\right) - g(z)\right| \leq \epsilon, \quad \text{for all } z \in \mathcal{S}^-. \tag{4.38}$$

Next, fix an input $x \in \mathcal{S}^+$ and a time index $t \in \mathbb{N}_0$ and let

$$z' := \mathcal{P}\mathbf{T}_{-t}\{x\}. \tag{4.39}$$

Note that

$$\begin{aligned} z'[\ell] &= 0, \quad \text{for } \ell \in \mathbb{N}_+, \text{ and hence } z' \in \mathcal{S}^-, \\ z'[-\ell] &= x[t-\ell], \qquad \text{for } \ell \in \mathbb{N}_0. \end{aligned} \tag{4.40}$$

Inserting z' from (4.39) into (4.38) and using (4.40) and (4.37), it follows that

$$\left|\Phi\left(\{x[t-\ell]\}_{\ell=0}^{T}\right) - G(x)[t]\right| \leq \epsilon.$$

As $x \in \mathcal{S}^-$ and $t \in \mathbb{N}_0$ were arbitrary, this proves (4.36).

Step 2: We construct an RNN $\mathcal{R}_\Psi$ realizing the mapping $x \to \Phi\left(\{x[t-\ell]\}_{\ell=0}^T\right)_{t\in\mathbb{N}_0}$.
Fix $x \in \mathcal{S}^+$ arbitrarily. Recall the RNN Definition 4.2.4. The basic idea is to identify an RNN $\mathcal{R}_\Psi$ which, for every time step $t \in \mathbb{N}_0$,

- delivers the output
$$y[t] = \Phi\left(\{x[t-\ell]\}_{\ell=0}^T\right) \tag{4.41}$$
- and memorizes the T past inputs $x[t], x[t-1], \ldots, x[t-T+1]$ in the hidden state vector $h[t]$, i.e.,
$$h_\ell[t] = x[t-\ell+1], \qquad \text{for} \quad \ell \in \{1, 2, \ldots, T\}. \tag{4.42}$$

To memorize the T past inputs, we note that the one-layer neural network

$$\Phi_T(z) = \begin{pmatrix}\mathbb{I}_T & 0_T\end{pmatrix} z, \quad \text{for } z \in \mathbb{R}^{T+1}, \tag{4.43}$$

satisfies

$$\begin{aligned}\Phi_T\left(\{x[t-\ell]\}_{\ell=0}^T\right) &= \begin{pmatrix}\mathbb{I}_T & 0_T\end{pmatrix}(x[t], x[t-1], \ldots, x[t-T])^T \\ &= (x[t], x[t-1], \ldots, x[t-T+1])^T \in \mathbb{R}^T.\end{aligned}$$

Now, we apply Lemma 4.C.5 to augment Φ_T to depth $\mathcal{L}(\Phi)$ without changing its input-output relation. This results in the ReLU network Φ_T^*. Then, we apply Lemma 4.C.4 to parallelize Φ and Φ_T^* leading to the desired ReLU network

$$\Psi = P(\Phi, \Phi_T^*). \tag{4.44}$$

By Definition 4.2.4, the corresponding RNN $\mathcal{R}_\Psi$ effects the input-output mapping according to

$$\begin{pmatrix}y[t] \\ h[t]\end{pmatrix} = \Psi\left(\begin{pmatrix}x[t] \\ h[t-1]\end{pmatrix}\right), \quad \text{for all } t \in \mathbb{N}_0.$$

With these choices, (4.41) and (4.42) can now be proven by induction over $t \in \mathbb{N}_0$. The base case is immediate as $x[t] = 0$, for $t < 0$, owing to $x \in \mathcal{S}^+$ and, by Definition 4.2.4, $h[-1] = 0_T$. To establish the induction step, we assume that (4.41) and (4.42) hold for $t-1$ with $t \in \mathbb{N}_+$, i.e.,

$$\begin{aligned}y[t-1] &= \Phi\left(\{x[t-1-\ell]\}_{\ell=0}^T\right), \\ h_\ell[t-1] &= x[t-\ell], \qquad \text{for} \quad \ell \in \{1, 2, \ldots, T\}.\end{aligned}$$

Now, for time step t, we note that

$$\begin{aligned}\Psi\left(\begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix}\right) &= P(\Phi, \Phi_T^*)\left(\begin{pmatrix} x[t] \\ h[t-1] \end{pmatrix}\right) \\ &= \begin{pmatrix} \Phi\left(\{x[t-\ell]\}_{\ell=0}^T\right) \\ \Phi_T^*\left(\{x[t-\ell]\}_{\ell=0}^T\right) \end{pmatrix} \\ &= \begin{pmatrix} \Phi\left(\{x[t-\ell]\}_{\ell=0}^T\right) \\ (x[t], x[t-1], \ldots, x[t-T+1])^T \end{pmatrix} \\ &= \begin{pmatrix} \Phi\left(\{x[t-\ell]\}_{\ell=0}^T\right) \\ h[t] \end{pmatrix}.\end{aligned}$$

As x was arbitrary, this completes the induction and thereby Step 2.

To conclude, we combine the results in Steps 1 and 2 according to

$$\begin{aligned}\rho_*(\mathcal{R}_\Psi, G) &= \sup_{x\in\mathcal{S}^+}\sup_{t\in\mathbb{N}_0} |y[t] - G(x)[t]| \\ &= \sup_{x\in\mathcal{S}^+}\sup_{t\in\mathbb{N}_0} \left|\Phi\left(\{x[t-\ell]\}_{\ell=0}^T\right) - G(x)[t]\right| \qquad (4.45) \\ &\leq \epsilon, \qquad (4.46)\end{aligned}$$

where (4.45) follows from (4.41) and (4.46) is by (4.36).

Furthermore, we have

$$\begin{aligned}\mathcal{K}(\Psi) &\overset{(4.44),\ \text{Lemma 4.C.4}}{=} \mathcal{K}(\Phi) \cup (\mathcal{K}(\Phi_T^*)) \\ &\overset{\text{Lemma 4.C.5}}{\subset} \mathcal{K}(\Phi) \cup \mathcal{K}(\Phi_T) \cup (-\mathcal{K}(\Phi_T)) \cup \{1, -1\} \\ &\overset{(4.43)}{=} \mathcal{K}(\Phi) \cup \{1, -1\} \\ &\overset{\text{Lemma 4.5.2}}{\subset} 2^{-2\left\lceil \log(\epsilon^{-1})\right\rceil}\mathbb{Z} \cap \left[-\epsilon^{-2}, \epsilon^{-2}\right].\end{aligned}$$

Thus, Ψ has $(2, \epsilon)$-quantized weights. Finally, we get an upper bound on $\mathcal{M}(\Psi)$ according to

$$\begin{aligned}\mathcal{M}(\Psi) &\overset{(4.44),\ \text{Lemma 4.C.4}}{=} \mathcal{M}(\Phi) + \mathcal{M}(\Phi_T^*) \\ &\overset{(4.43),\ \text{Lemma 4.C.5}}{\leq} \mathcal{M}(\Phi) + \mathcal{M}(\Phi_T) + T\mathcal{W}(\Phi_T) + 2T(\mathcal{L}(\Phi) - \mathcal{L}(\Phi_T)) \\ &\overset{(4.43),\ \text{Lemma 4.5.2}}{\leq} C(T+1)^2 \prod_{\ell=0}^{T}\left(\frac{2Dw[\ell]}{\epsilon}\frac{s+1}{s} + 4\right),\end{aligned}$$

with the universal constant $C > 0$ chosen sufficiently large. □

4.6. Metric-entropy-optimal approximation of ELFM and PLFM systems

So far we have characterized the description complexity of ELFM and PLFM systems based on order, type, and generalized dimension (Section 4.4) and we constructed RNNs approximating general LFM systems (Section 4.5). We are now ready to state the main results of the chapter, namely that the RNNs we constructed are optimal for ELFM and PLFM system approximation in terms of description complexity.

To this end, we first compute the number of bits needed by the canonical RNN decoder in Definition 4.2.7 to obtain the RNN constructed in Theorem 4.5.3, specifically its topology and quantized weights. The following result holds for general LFM systems and will later be particularized to ELFM and PLFM systems.

Corollary 4.6.1. *The class of LFM systems $(\mathcal{G}(w), \rho_*)$ is representable by the canonical RNN decoder $\mathcal{D}_\mathcal{R}$ with*

$$L(\epsilon; \mathcal{D}_\mathcal{R}, \mathcal{G}(w), \rho_*) \leq C_1 M \log(M) \log(\epsilon^{-1}),$$

where

$$M := (T+1)^2 \prod_{\ell=0}^{T} \left(\frac{12 D w[\ell]}{\epsilon} \right),$$
$$T := \max \left\{ \ell \in \mathbb{N}_0 \;\middle|\; w[\ell] > \frac{\epsilon}{2D} \right\},$$

and $C_1 > 0$ is a universal constant.

Proof. Applying Theorem 4.5.3 and setting $s = 1$, it follows that there exists an $\epsilon_0 > 0$, such that for every $\epsilon \in (0, \epsilon_0)$ and every $G \in \mathcal{G}(w)$, we can find an RNN $\mathcal{R}_\Psi$, associated with a ReLU network $\Psi \in \mathcal{N}_{T+1,T+1}$, satisfying

$$\rho_*(\mathcal{R}_\Psi, G) \leq \epsilon. \tag{4.47}$$

Moreover, $\mathcal{R}_\Psi$ has $(2, \epsilon)$-quantized weights and the number of non-zero weights in Ψ can be upper-bounded according to

$$\mathcal{M}(\Psi) \leq C(T+1)^2 \prod_{\ell=0}^{T} \left(\frac{4 D w[\ell]}{\epsilon} + 4 \right). \tag{4.48}$$

By the definition of the canonical neural network decoder, Remark 4.2.3, and Definition 4.2.7, there exists a bitstring $b \in \{0,1\}^L$ with

$$L \leq 2 C_0 \mathcal{M}(\Psi) \log(\mathcal{M}(\Psi)) \log(\epsilon^{-1}), \tag{4.49}$$

such that

$$\mathcal{D}_{\mathcal{R}}(b) = \mathcal{R}_{\Psi}. \tag{4.50}$$

Combining (4.47), (4.49), (4.50), and Definition 4.2.9, upon noting that G was chosen arbitrarily, it follows that $(\mathcal{G}(w), \rho_*)$ is representable by the canonical RNN decoder $\mathcal{D}_{\mathcal{R}}$, with the minimum number of required bits satisfying

$$\begin{aligned} L(\epsilon; \mathcal{D}_{\mathcal{R}}, \mathcal{G}(w), \rho_*) &\leq 2C_0 \mathcal{M}(\Psi) \log(\mathcal{M}(\Psi)) \log(\epsilon^{-1}) && (4.51) \\ &\leq 2C_0 C (T+1)^2 \left[\prod_{\ell=0}^{T} \left(\frac{12Dw[\ell]}{\epsilon}\right)\right] \log(\epsilon^{-1}) \\ &\quad \cdot \log\left(C(T+1)^2 \left(\prod_{\ell=0}^{T} \left(\frac{12Dw[\ell]}{\epsilon}\right)\right)\right) && (4.52) \\ &\leq 2C_0 C C' (T+1)^2 \left[\prod_{\ell=0}^{T} \left(\frac{12Dw[\ell]}{\epsilon}\right)\right] \log(\epsilon^{-1}) \\ &\quad \cdot \log\left((T+1)^2 \prod_{\ell=0}^{T} \left(\frac{12Dw[\ell]}{\epsilon}\right)\right) && (4.53) \\ &= C_1 M \log(M) \log(\epsilon^{-1}), && (4.54) \end{aligned}$$

where (4.51) is a consequence of (4.49) and Definition 4.2.9, (4.52) follows from (4.48) and the fact that $2Dw[\ell]/\epsilon > 1$, for $\ell \in [\![T]\!]$, (4.53) holds upon choosing the universal constant C' sufficiently large, namely s.t.

$$\log\left(C(T+1)^2 \left(\prod_{\ell=0}^{T} \left(\frac{12Dw[\ell]}{\epsilon}\right)\right)\right) \leq C' \log\left((T+1)^2 \left(\prod_{\ell=0}^{T} \left(\frac{12Dw[\ell]}{\epsilon}\right)\right)\right),$$

and (4.54) follows by setting $C_1 = 2C_0 C C'$. □

4.6.1. RNNs optimally approximate ELFM systems

We now particularize the result in Corollary 4.6.1 to ELFM systems, which allows us to determine the growth rate, with respect to ϵ, of the minimum number of bits $L(\epsilon; \mathcal{D}_{\mathcal{R}}, \mathcal{G}(w_{a,b}^{(e)}), \rho_*)$ needed to represent $(\mathcal{G}(w_{a,b}^{(e)}), \rho_*)$ by the canonical RNN decoder $\mathcal{D}_{\mathcal{R}}$. A comparison with the description complexity of $(\mathcal{G}(w_{a,b}^{(e)}), \rho_*)$ established in Lemma 4.4.3 then leads to the conclusion that RNNs are capable of approximating ELFM systems in metric-entropy-optimal manner.

Theorem 4.6.2. *Let $a \in (0, 1]$ and $b > 0$. The class of ELFM systems (Definition 4.4.1) $(\mathcal{G}(w_{a,b}^{(e)}), \rho_*)$ is optimally representable by the canonical RNN decoder $\mathcal{D}_{\mathcal{R}}$ (Definition 4.2.7).*

Proof. By Corollary 4.6.1, the minimum number of bits $L(\epsilon; \mathcal{D}_{\mathcal{R}}, \mathcal{G}(w_{a,b}^{(e)}), \rho_*)$ needed to represent $(\mathcal{G}(w_{a,b}^{(e)}), \rho_*)$ by the canonical RNN decoder $\mathcal{D}_{\mathcal{R}}$ satisfies

$$L(\epsilon; \mathcal{D}_{\mathcal{R}}, \mathcal{G}(w_{a,b}^{(e)}), \rho_*) \leq C_1 M \log(M) \log(\epsilon^{-1}),$$

where

$$M := (T+1)^2 \prod_{\ell=0}^{T} \left(\frac{12D w_{a,b}^{(e)}[\ell]}{\epsilon} \right),$$
$$T := \max \left\{ \ell \in \mathbb{N}_0 \;\middle|\; w_{a,b}^{(e)}[\ell] > \frac{\epsilon}{2D} \right\}.$$

By Lemma 4.4.3, $(\mathcal{G}(w_{a,b}^{(e)}), \rho_*)$ is of order 1 and type 2, with generalized dimension

$$\mathfrak{d} = \frac{1}{2b\log(e)}.$$

Thus, by Lemma 4.2.13 and Definition 4.2.14, it suffices to prove that

$$\lim_{\epsilon \to 0} \frac{\log\left(C_1 M \log(M) \log(\epsilon^{-1})\right)}{\log^2(\epsilon^{-1})} = \frac{1}{2b\log(e)}. \tag{4.55}$$

To this end, we first note that

$$\begin{aligned} \log(M) &= 2\log(T+1) + \log\left(\prod_{\ell=0}^{T} \left(\frac{12D w_{a,b}^{(e)}[\ell]}{\epsilon} \right) \right) \\ &= O(\log^{(2)}(\epsilon^{-1})) + \frac{1}{2b\log(e)} \log^2(\epsilon^{-1}) + o\left(\log^2(\epsilon^{-1})\right) \qquad (4.56) \\ &= \frac{1}{2b\log(e)} \log^2(\epsilon^{-1}) + o\left(\log^2(\epsilon^{-1})\right), \end{aligned}$$

where (4.56) follows from Lemma 4.4.2 and $T+1 = O(\log(\epsilon^{-1}))$ thanks to (4.89). Now, we rewrite the numerator in (4.55) according to

$$\begin{aligned} \log\left(C_1 M \log(M) \log(\epsilon^{-1})\right) &= \log(M) + \log^{(2)}(M) + o\left(\log^2(\epsilon^{-1})\right) \\ &= \frac{1}{2b\log(e)} \log^2(\epsilon^{-1}) + o\left(\log^2(\epsilon^{-1})\right) \\ &\quad + \log\left(\frac{1}{2b\log(e)} \log^2(\epsilon^{-1}) + o\left(\log^2(\epsilon^{-1})\right) \right) \\ &= \frac{1}{2b\log(e)} \log^2(\epsilon^{-1}) + o\left(\log^2(\epsilon^{-1})\right). \qquad (4.57) \end{aligned}$$

Dividing (4.57) by $\log^2(\epsilon^{-1})$ and taking $\epsilon \to 0$, concludes the proof. □

4.6.2. RNNs optimally approximate PLFM systems

We next particularize the result in Corollary 4.6.1 to PLFM systems and will see that the minimum number of bits $L(\epsilon; \mathcal{D}_{\mathcal{R}}, \mathcal{G}(w_{p,q}^{(p)}), \rho_*)$ required to represent $(\mathcal{G}(w_{p,q}^{(p)}), \rho_*)$ by the canonical RNN decoder $\mathcal{D}_{\mathcal{R}}$ grows significantly faster (with respect to the prescribed error ϵ) than for

ELFM systems. This can be attributed to the fact that the memory of PLFM systems decays much more slowly. Nonetheless, as shown next, RNNs are capable of approximating PLFM systems in metric-entropy-optimal manner.

Theorem 4.6.3. *Let $q \in (0,1]$ and $p > 0$. The class of PLFM systems $(\mathcal{G}(w_{p,q}^{(p)}), \rho_*)$ (Definition 4.4.4) is optimally representable by the canonical RNN decoder $\mathcal{D}_{\mathcal{R}}$ (Definition 4.2.7).*

Proof. By Corollary 4.6.1, the minimum number of bits $L(\epsilon; \mathcal{D}_{\mathcal{R}}, \mathcal{G}(w_{p,q}^{(p)}), \rho_*)$ needed to represent $(\mathcal{G}(w_{p,q}^{(p)}), \rho_*)$ by the canonical RNN decoder $\mathcal{D}_{\mathcal{R}}$ satisfies

$$L(\epsilon; \mathcal{D}_{\mathcal{R}}, \mathcal{G}(w_{p,q}^{(p)}), \rho_*) \leq C_1 M \log(M) \log(\epsilon^{-1}),$$

where

$$\begin{aligned} M &:= (T+1)^2 \prod_{\ell=0}^{T} \left(\frac{12 D w_{p,q}^{(p)}[\ell]}{\epsilon} \right), \\ T &:= \max \left\{ \ell \in \mathbb{N}_0 \;\middle|\; w_{p,q}^{(p)}[\ell] > \frac{\epsilon}{2D} \right\}. \end{aligned}$$

By Lemma 4.4.6, $(\mathcal{G}(w_{p,q}^{(p)}), \rho_*)$ is of order 2 and type 1, with generalized dimension

$$\mathfrak{d} = \frac{1}{p}.$$

Thus, by Lemma 4.2.13 and Definition 4.2.14, it suffices to prove that

$$\lim_{\epsilon \to 0} \frac{\log^{(2)} \left(C_1 M \log(M) \log(\epsilon^{-1}) \right)}{\log(\epsilon^{-1})} = \frac{1}{p}. \tag{4.58}$$

To this end, we first note that

$$\begin{aligned} \log(M) &= 2 \log(T+1) + \log \left(\prod_{\ell=0}^{T} \left(\frac{12 D w_{p,q}^{(p)}[\ell]}{\epsilon} \right) \right) \\ &= O(\log(\epsilon^{-1})) + \Theta(\epsilon^{-1/p}) \qquad (4.59) \\ &= \Theta(\epsilon^{-1/p}), \end{aligned}$$

where (4.59) follows from Lemma 4.4.5 and (4.90). Now, we rewrite the numerator in (4.58) according to

$$\begin{aligned} \log^{(2)} \left(C_1 M \log(M) \log(\epsilon^{-1}) \right) &= \log \left(\log(M) + \log^{(2)}(M) + o\left(\log(\epsilon^{-1}) \right) \right) \\ &= \log \left(\Theta(\epsilon^{-1/p}) + \log \left(\Theta(\epsilon^{-1/p}) \right) + o\left(\log(\epsilon^{-1}) \right) \right) \\ &= \log \left(\Theta(\epsilon^{-1/p}) \right), \qquad (4.60) \end{aligned}$$

where (4.60) follows from

$$\Theta(\epsilon^{-1/p}) + \log\left(\Theta(\epsilon^{-1/p})\right) + o\left(\log(\epsilon^{-1})\right) = \Theta(\epsilon^{-1/p}).$$

Finally, dividing (4.60) by $\log(\epsilon^{-1})$ and taking $\epsilon \to 0$, concludes the proof. □

4.7. Conclusion

Returning to Table 4.1, we note that it can be complemented by our results for ELFM and PLFM systems as summarized in Table 4.2 below. As both classes of systems in Table 4.2 are optimally representable by the canonical RNN decoder (Definition 4.2.7), we can conclude that, remarkably, the metric-entropy-optimal universality of ReLU networks extends from function classes to nonlinear dynamical systems, in the latter case simply by embedding the ReLU network in a recurrence.

Metric		$\mathcal{C}$		κ	λ	$\mathfrak{d}$
$\{\mathbb{R}^{\mathbb{Z}} \to \mathbb{R}^{\mathbb{Z}}\}$	ρ_* (Def. 4.2.20)	ELFM (Section 4.4.1)	$\mathcal{G}(w^{(e)}_{a,b})$	1	2	$\frac{1}{2b\log(e)}$
$\{\mathbb{R}^{\mathbb{Z}} \to \mathbb{R}^{\mathbb{Z}}\}$	ρ_* (Def. 4.2.20)	PLFM (Section 4.4.2)	$\mathcal{G}(w^{(p)}_{p,q})$	2	1	$\frac{1}{p}$

Table 4.2: Scaling behavior of the covering numbers (Definition 4.2.12) for classes of nonlinear dynamical systems.

Finally, we remark that many of the results in this chapter apply to general weight sequences $w[\cdot]$, specifically the bounds in Section 4.3 on the exterior covering number of $(\mathcal{G}(w), \rho_*)$ as well as the RNN constructions in Section 4.5.

APPENDICES

4.A. Representing ReLU networks by bitstrings

Definition 4.A.1. *Let Φ be a ReLU network with (m,ϵ)-quantized weights. Denote the number of non-zero weights by $M \coloneqq \mathcal{M}(\Phi)$ and the depth by $L \coloneqq \mathcal{L}(\Phi)$. We organize the bitstring representation of Φ in 6 segments as follows.*

[Segment 1] The bitstring starts with M 1's followed by a single 0.

[Segment 2] L is specified in binary representation. As $L \leq M$, it suffices to allocate $\lceil \log(M) \rceil$ bits.

[Segment 3] $N_0, \dots, N_L \leq M$ are specified in binary representation using a total of $(M+1)\lceil \log(M) \rceil$ bits.

[Segment 4] The topology of the network, i.e., the locations of the non-zero entries in the A_ℓ and b_ℓ, $\ell \in \{1, \dots, L\}$, is encoded as follows. We denote the bitstring corresponding to the binary representation of an integer $i \in \{1, \dots, M\}$ by $b(i) \in \{0,1\}^{\lceil \log(M) \rceil}$. For $\ell \in \{1, \dots, L\}$, $i \in \{1, \dots, N_\ell\}$, $j \in \{1, \dots, N_{\ell-1}\}$, a non-zero entry $(A_\ell)_{ij}$ is indicated by $[b(\ell), b(i), b(j)]$ and a non-zero entry $(b_\ell)_i$ by $[b(\ell), b(i), b(i)]$. Thus, encoding the topology of the network requires a total of $3\lceil \log(M) \rceil M$ bits.

[Segment 5] The quantity $m\lceil \log(\epsilon^{-1}) \rceil$ is represented by a bitstring of that many 1's followed by a single 0.

[Segment 6] The value of each non-zero weight and bias is represented by a bitstring of length $B_\epsilon = 2m\lceil \log(\epsilon^{-1}) \rceil + 1$.

The overall length of the bitstring is now given by

$$\underbrace{M+1}_{\text{Segment 1}} + \underbrace{\lceil \log(M) \rceil}_{\text{Segment 2}} + \underbrace{(M+1)\lceil \log(M) \rceil}_{\text{Segment 3}} + \underbrace{3\lceil \log(M) \rceil M}_{\text{Segment 4}} + \underbrace{m\left\lceil \log(\epsilon^{-1}) \right\rceil + 1}_{\text{Segment 5}} + \underbrace{MB_\epsilon}_{\text{Segment 6}} \tag{4.61}$$

The ReLU network Φ can be recovered by sequentially reading out M,L, the N_ℓ, the topology, the quantity $m\lceil \log(\epsilon^{-1}) \rceil$, and the quantized weights from the overall bitstring. It is not difficult to verify that the bitstring is crafted such that this yields unique decodability.

4.B. Comparison with [Elb+21]

As mentioned in Section 4.2.2, compared to [Elb+21], we use refined notions of massiveness of sets, as the system classes we consider are significantly more massive than the function classes dealt with in [Elb+21]. We next detail how the results from [Elb+21] fit into our framework.

In [Elb+21] the scaling behavior of covering numbers is quantified in terms of the optimal exponent γ^* [Elb+21, Definition IV.1]. The following result relates γ^* to the generalized dimension (Definition 4.2.12) employed here.

Lemma 4.B.1. *Let $d \in \mathbb{N}_+$, $\Omega \subseteq \mathbb{R}^d$, and let $\mathcal{C} \subset L^2(\Omega)$ be compact and such that the optimal exponent $\gamma^*(\mathcal{C})$ according to [Elb+21, Definition IV.1] is finite and non-zero. Then, $\mathcal{C}$ is, with respect to the metric $\rho(f,g) := \|f-g\|_{L^2(\Omega)}$, of order $\kappa = 1$, type $\lambda = 1$, and generalized dimension*

$$\mathfrak{d} = \frac{1}{\gamma^*}.$$

Proof. For order $\kappa = 1$ and type $\lambda = 1$, the generalized dimension is given by

$$\mathfrak{d} = \limsup_{\epsilon \to 0} \frac{\log^{(2)} N^{\mathrm{ext}}(\epsilon; \mathcal{C}, \rho)}{\log(\epsilon^{-1})}. \tag{4.62}$$

We note that by [Gro15, Remark 5.10, Equation (5.5)] and Lemma 4.3.2, it holds that

$$\gamma^* = \sup\left\{\gamma > 0 : \log N^{\mathrm{ext}}(\epsilon; \mathcal{C}, \rho) \in O(\epsilon^{-1/\gamma}), \epsilon \to 0\right\}.$$

We first establish that $\mathfrak{d} \leq \frac{1}{\gamma^*}$. To this end, fix $\Delta > 0$ arbitrarily and observe that $\log N^{\mathrm{ext}}(\epsilon; \mathcal{C}, \rho) \in O(\epsilon^{-1/(\gamma^*-\Delta)})$. Hence, there exist $\epsilon_0, C > 0$ such that

$$\log N^{\mathrm{ext}}(\epsilon; \mathcal{C}, \rho) \leq C\epsilon^{-1/(\gamma^*-\Delta)}, \qquad \forall \epsilon \in (0, \epsilon_0),$$

and thus

$$\mathfrak{d} = \limsup_{\epsilon \to 0} \frac{\log^{(2)} N^{\mathrm{ext}}(\epsilon; \mathcal{C}, \rho)}{\log(\epsilon^{-1})} \leq \limsup_{\epsilon \to 0} \left(\frac{1}{\gamma^* - \Delta} + \frac{\log(C)}{\log(\epsilon^{-1})}\right) = \frac{1}{\gamma^* - \Delta}.$$

As $\Delta > 0$ was arbitrary, we have established that $\mathfrak{d} \leq \frac{1}{\gamma^*}$.

Next, we show that $\mathfrak{d} \geq \frac{1}{\gamma^*}$. Again, fix $\Delta > 0$ arbitrarily. By (4.62), there exists an $\epsilon_0 > 0$ such that for all $\epsilon \in (0, \epsilon_0)$,

$$\frac{\log^{(2)} N^{\mathrm{ext}}(\epsilon; \mathcal{C}, \rho)}{\log(\epsilon^{-1})} \leq \mathfrak{d} + \Delta.$$

This implies

$$\log N^{\mathrm{ext}}(\epsilon; \mathcal{C}, \rho) \leq \epsilon^{-(\mathfrak{d}+\Delta)}, \qquad \forall \epsilon \in (0, \epsilon_0),$$

and thus $\log N^{\text{ext}}(\epsilon; \mathcal{C}, \rho) \in O(\epsilon^{-1/(\mathfrak{d}+\Delta)^{-1}})$. Hence, $\gamma^* \geq (\mathfrak{d}+\Delta)^{-1}$. As Δ was arbitrary, this finalizes the proof. □

Table 4.1 in the present chapter now follows from [Elb+21, Table I] by application of Lemma 4.B.1. Furthermore, [Elb+21] shows through the transference principle [Elb+21, Section VII] that a wide range of function classes, including those in Table 4.1, are *optimally representable by neural networks* [Elb+21, Definition VI.5]. The following Lemma hence allows us to conclude that every row in Table 4.1 is optimally representable (according to Definition 4.2.14) by the canonical neural network decoder.

Lemma 4.B.2. *Let $d \in \mathbb{N}_+$, $\Omega \subseteq \mathbb{R}^d$, and let $\mathcal{C} \subset L^2(\Omega)$ be compact. If the function class $\mathcal{C} \subset L^2(\Omega)$ is* optimally representable by neural networks *according to [Elb+21, Definition VI.5], then $(\mathcal{C}, \rho)$ is optimally representable by the canonical neural network decoder with respect to the metric $\rho(f,g) \coloneqq \|f-g\|_{L^2(\Omega)}$.*

Proof. First, we note that by Lemma 4.B.1, $\mathcal{C}$ is of order $\kappa = 1$, type $\lambda = 1$, and has generalized dimension $\mathfrak{d} = \frac{1}{\gamma^*(\mathcal{C})}$. By [Elb+21, Definition VI.5], we have $\gamma^*(\mathcal{C}) = \gamma_{\mathcal{N}}^{*,\mathit{eff}}(\mathcal{C})$, with $\gamma_{\mathcal{N}}^{*,\mathit{eff}}(\mathcal{C})$ as per [Elb+21, Definition VI.3]. Next, fix $\Delta > 0$ arbitrarily. Now, following the proof of [Elb+21, Theorem VI.4, p. 2602] with $\gamma_{\mathcal{N}}^{*,\mathit{eff}} - \Delta$ in place of γ, we can conclude the existence of a polynomial π^*, a constant C, and a mapping $\Psi : (0, \frac{1}{2}) \times \mathcal{C} \to \mathcal{N}_{d,1}$ with the following properties. For every $f \in \mathcal{C}$ and every $\epsilon \in (0, \frac{1}{2})$, the network $\widetilde{\Phi}_{\epsilon,f} = \Psi(\epsilon, f)$ has $(\lceil \pi^*(\log(\epsilon^{-1}))\rceil, \epsilon)$-quantized weights and satisfies

$$\|f - \widetilde{\Phi}_{\epsilon,f}\|_{L^2(\Omega)} \leq \epsilon \quad \text{and} \quad \mathcal{M}(\widetilde{\Phi}_{\epsilon,f}) \leq C\epsilon^{-1/(\gamma_{\mathcal{N}}^{*,\mathit{eff}}-\Delta)} =: M_\epsilon.$$

Thus, $\widetilde{\Phi}_{\epsilon,f}$ can, according to Remark 4.2.3, be reconstructed uniquely by the canonical neural network decoder $\mathcal{D}_{\mathcal{N}}$ from a bitstring of length no more than

$$C_0 \left\lceil \pi^*(\log(\epsilon^{-1})) \right\rceil \log(\epsilon^{-1}) M_\epsilon \log(M_\epsilon).$$

Therefore, $\mathcal{C}$ is representable by the canonical neural network decoder $\mathcal{D}_{\mathcal{N}}$ with $L(\epsilon; \mathcal{D}_{\mathcal{N}}, \mathcal{C}, \rho) \leq C_0 M_\epsilon \log(M_\epsilon) \log^{q_0}(\epsilon^{-1})$, where q_0 is a constant depending on π^* only, and hence

$$\log L(\epsilon; \mathcal{D}_{\mathcal{N}}, \mathcal{C}, \rho) \leq \frac{1}{\gamma_{\mathcal{N}}^{*,\mathit{eff}} - \Delta} \log(\epsilon^{-1}) + o(\log(\epsilon^{-1})).$$

As $\Delta > 0$ was arbitrary, we thus get

$$\limsup_{\epsilon \to 0} \frac{L(\epsilon; \mathcal{D}_{\mathcal{N}}, \mathcal{C}, \rho)}{\log(\epsilon^{-1})} \leq \frac{1}{\gamma_{\mathcal{N}}^{*,\mathit{eff}}},$$

which, together with Lemma 4.2.13 and the fact that $\mathfrak{d} = \frac{1}{\gamma^*} = \frac{1}{\gamma_{\mathcal{N}}^{*,\mathit{eff}}}$ implies that $\mathcal{C}$ is optimally representable by the canonical neural network decoder according to Definition 4.2.14. □

4.C. Proofs

4.C.1. Proof of Lemma 4.2.13

To simplify notation, we let $L(\epsilon) \coloneqq L(\epsilon; \mathcal{D}, \mathcal{C}, \rho)$. We first establish that

$$L(\epsilon) \geq \log\left(N^{\text{ext}}(\epsilon; \mathcal{C}, \rho)\right) - 1, \quad \forall \epsilon > 0. \tag{4.63}$$

By way of contradiction, assume that

$$2^{L(\epsilon)+1} < N^{\text{ext}}(\epsilon; \mathcal{C}, \rho), \quad \text{for some } \epsilon > 0.$$

It then follows from Definition 4.2.9 that for this ϵ, for every $f \in \mathcal{C}$, there is an integer $\ell \leq L(\epsilon)$ and a bitstring $\mathbf{b}_f \in \{0,1\}^\ell$, such that $\rho(\mathcal{D}(\mathbf{b}_f), f) \leq \epsilon$. This directly implies that the set

$$\mathcal{U} \coloneqq \left\{ \mathcal{D}(\mathbf{b}) \middle| \, \mathbf{b} \in \bigcup_{\ell=0}^{L(\epsilon)} \{0,1\}^\ell \right\},$$

is an ϵ-net for $\mathcal{C}$. Furthermore,

$$|\mathcal{U}| \leq \sum_{\ell=0}^{L(\epsilon)} 2^\ell = \frac{2^{L(\epsilon)+1} - 1}{2-1} \leq 2^{L(\epsilon)+1} < N^{\text{ext}}(\epsilon; \mathcal{C}, \rho).$$

Hence, $\mathcal{U}$ is an ϵ-net of cardinality strictly smaller than $N^{\text{ext}}(\epsilon; \mathcal{C}, \rho)$, which stands in contradiction to Definition 4.2.11 and so (4.63) must hold. This, in turn, implies that

$$\begin{aligned}
\limsup_{\epsilon \to 0} \frac{\log^{(\kappa)} L(\epsilon)}{\log^\lambda(\epsilon^{-1})} &\geq \limsup_{\epsilon \to 0} \frac{\log^{(\kappa)}\left(\log\left(N^{\text{ext}}(\epsilon; \mathcal{C}, \rho_*)\right) - 1\right)}{\log^\lambda(\epsilon^{-1})} \\
&= \limsup_{\epsilon \to 0} \frac{\log^{(\kappa+1)} N^{\text{ext}}(\epsilon; \mathcal{C}, \rho_*)}{\log^\lambda(\epsilon^{-1})} = \mathfrak{d},
\end{aligned}$$

as desired. □

4.C.2. Proof of Lemma 4.2.21

Fix $G, G' \in \mathcal{G}(w)$. First, note that

$$\begin{aligned}
\rho_*(G, G') &= \sup_{x \in \mathcal{S}^+} \sup_{t \in \mathbb{N}_0} |(Gx)[t] - (G'x)[t]| \\
&\leq \sup_{x \in \mathcal{S}} \sup_{t \in \mathbb{Z}} |(Gx)[t] - (G'x)[t]|.
\end{aligned} \tag{4.64}$$

Next, arbitrarily fix $\Delta > 0$. By definition of the supremum, it follows that there exist $x_0 \in \mathcal{S}$ and $e \in \mathbb{Z}$, such that

$$\sup_{x\in\mathcal{S}} \sup_{t\in\mathbb{Z}} |(Gx)[t] - (G'x)[t]| - \Delta/2 \leq |(Gx_0)[e] - (G'x_0)[e]|\,. \tag{4.65}$$

Since $\lim_{t\to\infty} w[t] = 0$, there exists $T > 0$ so that $w[t] \leq \Delta/(4D)$, for all $t > T$. Next, define

$$x_1[t] = x_0[t - (T - e)] \in \mathcal{S} \quad \text{and} \quad y[t] = x_1[t] \cdot \mathbb{1}_{\{t\geq 0\}} \in \mathcal{S}^+.$$

We then get

$$\sup_{\tau\geq 0} w[\tau]|(x_1[T-\tau] - y[T-\tau])| = \sup_{\tau>T} w[\tau]|x_1[T-\tau]| \leq \sup_{\tau>T} w[\tau]D \leq \Delta/4, \tag{4.66}$$

where we used that $|x_1[\cdot]| \leq D$ by Definition 4.2.15. We furthermore obtain

$$\begin{aligned}
|(Gx_1)[T] - (G'x_1)[T]| &\leq |(Gx_1)[T] - (Gy)[T]| + |(G'x_1)[T] - (G'y)[T]| \\
&\quad + |(Gy)[T] - (G'y)[T]| && (4.67)\\
&\leq \Delta/4 + \Delta/4 + |(Gy)[T] - (G'y)[T]| && (4.68)\\
&= \Delta/2 + |(Gy)[T] - (G'y)[T]|, && (4.69)
\end{aligned}$$

where in (4.67) we used the triangle inequality and in (4.68) we invoked the Lipschitz fading-memory property of G and G' in combination with (4.66). Next, note that by time-invariance of G and G', it holds that

$$\begin{aligned}
(Gx_1)[T] &= (G\mathbf{T}_{(T-e)}x_0)[T] = (\mathbf{T}_{(T-e)}Gx_0)[T] = (Gx_0)[e], \\
(G'x_1)[T] &= (G'\mathbf{T}_{(T-e)}x_0)[T] = (\mathbf{T}_{(T-e)}G'x_0)[T] = (G'x_0)[e].
\end{aligned} \tag{4.70}$$

Combining all these results yields

$$\begin{aligned}
\sup_{x\in\mathcal{S}} \sup_{t\in\mathbb{Z}} |(Gx)[t] - (G'x)[t]| - \Delta/2 &\overset{(4.65)}{\leq} |(Gx_0)[e] - (G'x_0)[e]| \\
&\overset{(4.70)}{=} |(Gx_1)[T] - (G'x_1)[T]| \\
&\overset{(4.69)}{\leq} \Delta/2 + |(Gy)[T] - (G'y)[T]| \\
&\overset{y\in\mathcal{S}^+}{\leq} \Delta/2 + \sup_{x\in\mathcal{S}^+} \sup_{t\in\mathbb{N}_0} |(Gx)[t] - (G'x)[t]| \\
&\overset{\text{Def. 4.2.20}}{=} \Delta/2 + \rho_*(G, G').
\end{aligned}$$

As $\Delta > 0$ was arbitrary, we have thus established that

$$\sup_{x\in\mathcal{S}}\sup_{t\in\mathbb{Z}}|(Gx)[t]-(G'x)[t]| \leq \rho_*(G,G'),$$

which, together with (4.64), completes the proof. □

4.C.3. Proof of Lemma 4.3.5

Recall the projection operator $\mathcal{P}:\mathcal{S}\to\mathcal{S}^-$ defined according to

$$(\mathcal{P}x)[t] = x[t]\cdot \mathbb{1}_{\{t\leq 0\}}.$$

First, we need to show that $\mathcal{I}:\mathcal{G}_0(w)\to\mathcal{G}(w)$ given by

$$g\to\mathcal{I}(g), \quad \text{with } ((\mathcal{I}(g))(x))[t] = g(\mathcal{P}\mathbf{T}_{-t}x),$$

is a well-defined mapping from $\mathcal{G}_0(w)$ to $\mathcal{G}(w)$, i.e., we need to verify that for every $g\in\mathcal{G}_0(w)$, indeed $\mathcal{I}(g)\in\mathcal{G}(w)$. This will be done by establishing that $\mathcal{I}(g)$ satisfies the conditions of Definition 4.2.19. We first verify that $\mathcal{I}(g)$ is causal. Note that for every $T\in\mathbb{Z}$, for every $x,x'\in\mathcal{S}$ with $x[t]=x'[t]$, $\forall t\leq T$, it holds that $\mathcal{P}\mathbf{T}_{-T}x=\mathcal{P}\mathbf{T}_{-T}x'$, and hence

$$((\mathcal{I}(g))(x))[T] = g(\mathcal{P}\mathbf{T}_{-T}x) = g(\mathcal{P}\mathbf{T}_{-T}x') = ((\mathcal{I}(g))(x'))[T].$$

Thus, $\mathcal{I}(g)$ is, indeed, causal according to Definition 4.2.16. Next, we verify time-invariance as follows:

$$\begin{aligned}(\mathbf{T}_\tau(\mathcal{I}(g)(x)))[t] &= ((\mathcal{I}(g))(x))[t-\tau]\\ &= g(\mathcal{P}\mathbf{T}_{\tau-t}x)\\ &= g(\mathcal{P}\mathbf{T}_{-t}\mathbf{T}_\tau x)\\ &= ((\mathcal{I}(g))(\mathbf{T}_\tau x))[t].\end{aligned}$$

The Lipschitz fading-memory property of $\mathcal{I}(g)$ according to Definition 4.2.18 is established by noting that

$$\begin{aligned}&|((\mathcal{I}(g))(x))[t]-((\mathcal{I}(g))(x'))[t]|\\ &= |g(\mathcal{P}\mathbf{T}_{-t}x)-g(\mathcal{P}\mathbf{T}_{-t}x')|\\ &\leq \sup_{\tau\geq 0} w[\tau]|((\mathcal{P}\mathbf{T}_{-t}x)[-\tau]-(\mathcal{P}\mathbf{T}_{-t}x')[-\tau])| \qquad (4.71)\\ &= \sup_{\tau\geq 0} w[\tau]|(x[t-\tau]-x'[t-\tau])|, \quad \forall t\in\mathbb{Z},\forall x,x'\in\mathcal{S},\end{aligned}$$

where in (4.71) we used (4.2). Finally, we observe that, by (4.2),

$$((\mathcal{I}(g))(0))[t] = g(\mathcal{P}\mathbf{T}_{-t}0) = g(0) = 0,\ \forall t \in \mathbb{Z}.$$

In summary, we have thus shown that $\mathcal{I}(g) \in \mathcal{G}(w)$ and hence $\mathcal{I}$ is, indeed, well-defined. Furthermore, we have

$$\begin{aligned}
\rho_*(\mathcal{I}(g), \mathcal{I}(g')) &= \sup_{x\in\mathcal{S}} \sup_{t\in\mathbb{Z}} |((\mathcal{I}(g))(x))[t] - ((\mathcal{I}(g'))(x))[t]| && (4.72)\\
&= \sup_{x\in\mathcal{S}} \sup_{t\in\mathbb{Z}} |g(\mathcal{P}\mathbf{T}_{-t}x) - g'(\mathcal{P}\mathbf{T}_{-t}x)| \\
&= \sup_{y\in\mathcal{S}} |g(\mathcal{P}y) - g'(\mathcal{P}y)| && (4.73)\\
&= \sup_{z\in\mathcal{S}^-} |g(z) - g'(z)| \\
&= \rho_0(g, g'), \qquad \forall g \in \mathcal{G}_0(w), \forall g' \in \mathcal{G}_0(w),
\end{aligned}$$

where in (4.72) we invoked Lemma 4.2.21 and in (4.73) we used that $\mathcal{S}$ is closed under time shifts. This establishes that $\mathcal{I}$ is isometric and consequently injective.

Next, we prove that $\mathcal{I}$ is surjective. To this end, we fix $G \in \mathcal{G}(w)$ arbitrarily, consider

$$g(s) := (Gs)[0], \quad \forall s \in S_-, \tag{4.74}$$

and show that $g \in \mathcal{G}_0(w)$ as well as $\mathcal{I}(g) = G$. First, we establish that $g \in \mathcal{G}_0(w)$. The Lipschitz property of g can be verified according to

$$\begin{aligned}
|g(x) - g(x')| &= |(Gx)[0] - (Gx')[0]| \\
&\le \sup_{\tau\ge 0} w[\tau]|(x[-\tau] - x'[-\tau])| && (4.75)\\
&= \|x - x'\|_w, \quad \forall x, x' \in \mathcal{S}^-, && (4.76)
\end{aligned}$$

where in (4.75) we used the fact that $G \in \mathcal{G}(w)$ has Lipschitz fading memory (Definition 4.2.18) and (4.76) is by (4.3). Furthermore, we have $g(0) = (G0)[0] = 0$ and hence $g \in \mathcal{G}_0(w)$.

It remains to show that $\mathcal{I}(g) = G$. To this end, we fix $x \in \mathcal{S}$ and $t \in \mathbb{Z}$, both arbitrarily, and prove that

$$((\mathcal{I}(g))(x))[t] = (Gx)[t].$$

First, note that $(\mathcal{P}\mathbf{T}_{-t}x)[t'] = (\mathbf{T}_{-t}x)[t']$, for all $t' \le 0$. By causality of G, we conclude that

$(G\mathcal{P}\mathbf{T}_{-t}x)[0] = (G\mathbf{T}_{-t}x)[0]$, which yields

$$\begin{aligned}((\mathcal{I}(g))(x))[t] &= g(\mathcal{P}\mathbf{T}_{-t}x)\\ &= (G\mathcal{P}\mathbf{T}_{-t}x)[0] \qquad (4.77)\\ &= (G\mathbf{T}_{-t}x)[0] = (\mathbf{T}_{-t}Gx)[0] \qquad (4.78)\\ &= (Gx)[t],\end{aligned}$$

where in (4.77) we used (4.74), and in (4.78) we invoked the causality and the time-invariance of G. As $x \in \mathcal{S}$ and $t \in \mathbb{Z}$ were arbitrary, this proves that $\mathcal{I}(g) = G$. Furthermore, since $G \in \mathcal{G}(w)$ was arbitrary, we have established that $\mathcal{I}$ is surjective. Thus, $\mathcal{I}$ is, indeed, an isometric isomorphism between $(\mathcal{G}(w), \rho_*)$ and $(\mathcal{G}_0(w), \rho_0)$. Application of Lemma 4.3.3 then yields $N(\epsilon; \mathcal{G}_0(w), \rho_0) = N(\epsilon; \mathcal{G}(w), \rho_*)$ and $M(\epsilon; \mathcal{G}_0(w), \rho_0) = M(\epsilon; \mathcal{G}(w), \rho_*)$. □

4.C.4. Proof of Lemma 4.3.6

The proof relies on the following auxiliary result.

Lemma 4.C.1. *Let $w[\cdot]$ be a weight sequence. The packing number of $\mathcal{S}^-$ w.r.t. $\|\cdot\|_w$ satisfies*

$$M(\epsilon; \mathcal{S}^-, \|\cdot\|_w) \geq \prod_{\ell=0}^{T} \left\lceil \frac{2Dw[\ell]}{\epsilon} \right\rceil,$$

with $T := max\{T' \in \mathbb{N}_0 \mid w[T'] > \frac{\epsilon}{2D}\}$.

Proof. For $t \in [\![T]\!]$, we let $N_t := \left\lceil \frac{2Dw[t]}{\epsilon} \right\rceil \in \left[\frac{2Dw[t]}{\epsilon}, \frac{2Dw[t]}{\epsilon} + 1\right)$ and $\delta_t = \frac{2D}{N_t - 1} > \frac{\epsilon}{w[t]}$. Next, we define the set

$$\mathcal{U} = \{x_{i_0,\dots,i_T} \mid i_t \in [\![N_t - 1]\!], \text{ for } t \in [\![T]\!]\}, \text{ where}$$
$$x_{i_0,\dots,i_T} := \begin{cases} -D + i_t\delta_t, & -T \leq t \leq 0 \\ 0, & \text{else} \end{cases},$$

and show that $\mathcal{U}$ constitutes an ϵ-packing for $(\mathcal{S}^-, \|\cdot\|_w)$. First, we establish that $\mathcal{U} \subset \mathcal{S}^-$ by verifying that, for all $x_{i_0,\dots,i_T} \in \mathcal{U}$, $x_{i_0,\dots,i_T}[t] \in [-D, D]$, for all $t \in \mathbb{Z}$, and $x_{i_0,\dots,i_T}[t] = 0$, for all $t \in \mathbb{N}_+$. Indeed, for $t \in \{-T, \dots, 0\}$, we have

$$-D \leq x_{i_0,\dots,i_T}[t] = -D + i_t\delta_t \leq -D + (N_t - 1)\delta_t = -D + 2D = D,$$

and for $t \notin \{-T, \dots, 0\}$,

$$x_{i_0,\dots,i_T}[t] = 0.$$

Next, we show that for distinct $x_{i_0,\dots,i_T}, x_{j_0,\dots,j_T} \in \mathcal{U}$ (i.e., there is at least one $\ell \in [\![T]\!]$ such that

$i_\ell \neq j_\ell$), it holds that $\|x_{i_0,\dots,i_T} - x_{j_0,\dots,j_T}\|_w > \epsilon$. Indeed, for any such ℓ, we have

$$\begin{aligned}\|x_{i_0,\dots,i_T} - x_{j_0,\dots,j_T}\|_w &= \sup_{t\in\mathbb{N}_0} w[t]\,|(x_{i_0,\dots,i_T}[-t] - x_{j_0,\dots,j_T}[-t])| \\ &\geq w[\ell]\,|(x_{i_0,\dots,i_T}[-\ell] - x_{j_0,\dots,j_T}[-\ell])| \\ &= w[\ell]\,|(-D + i_\ell\delta_\ell + D - j_\ell\delta_\ell)| \\ &= (i_\ell - j_\ell)\delta_\ell w[\ell] > \epsilon, \end{aligned} \tag{4.79}$$

where (4.79) follows from $\delta_\ell > \frac{\epsilon}{w[\ell]}$. This establishes that $\mathcal{U}$, indeed, constitutes an ϵ-packing for $(\mathcal{S}^-, \|\cdot\|_w)$, and we therefore have

$$M(\epsilon; \mathcal{S}^-, \|\cdot\|_w) \geq |\mathcal{U}| = \prod_{\ell=0}^{T} N_t = \prod_{\ell=0}^{T} \left\lceil \frac{2Dw[\ell]}{\epsilon} \right\rceil.$$

□

Proof of Lemma 4.3.6. Let $M := \prod_{\ell=0}^{T} \left\lceil \frac{2Dw[\ell]}{\epsilon} \right\rceil$ and take $\{x_1, \dots, x_M\}$ to be an ϵ-packing for $(\mathcal{S}^-, \|\cdot\|_w)$ according to Lemma 4.C.1. Hence, with $\delta := \min_{\ell\neq j} \|x_\ell - x_j\|_w > \epsilon$, there exists an $\epsilon' \in (\epsilon, \delta)$. Next, define balls of radius $\epsilon'/2$ centered at the packing points $\{x_1, \dots, x_M\}$ according to $\mathcal{S}_\ell = \{x[\cdot] \in \mathcal{S}^- \,|\, \|x - x_\ell\|_w \leq \epsilon'/2\}$, for $\ell \in \{1, \dots, M\}$. We now show that these balls are non-overlapping. Indeed, assuming that, for $j \neq \ell$, there exists an $x \in \mathcal{S}^-$ with $\|x - x_j\|_w < \epsilon'/2$ and $\|x - x_\ell\|_w < \epsilon'/2$, leads to the contradiction

$$\epsilon' < \delta \leq \|x_\ell - x_j\|_w = \|x_\ell - x + x - x_j\|_w \leq \|x_\ell - x\|_w + \|x - x_j\|_w \leq \epsilon'/2 + \epsilon'/2.$$

As the balls $\mathcal{S}_\ell$, $\ell \in \{1, \dots, M\}$, are non-overlapping, the signal $x[\cdot] = 0$ is contained in at most one ball $\mathcal{S}_\ell$ which we take to be $\mathcal{S}_M$ w.l.o.g. Now, we define the set $\mathcal{U}$, with elements indexed by bitstring subscripts, according to

$$\mathcal{U} = \Big\{ g_{\alpha_1,\dots,\alpha_{M-1}}(\cdot) \mid \alpha_\ell \in \{0,1\},\ \text{for } \ell \in \{1,\dots,M-1\} \Big\},\ \text{where}$$

$$g_{\alpha_1,\dots,\alpha_{M-1}}(x) := \begin{cases} (2\alpha_\ell - 1)(\epsilon'/2 - \|x - x_\ell\|_w), & \text{for } x \in \mathcal{S}_\ell,\ \ell \in \{1,\dots,M-1\} \\ 0, & \text{else,} \end{cases}$$

and show that $\mathcal{U}$ constitutes an ϵ-packing for $(\mathcal{G}_0(w), \rho_0)$. First, we establish that $\mathcal{U} \subset \mathcal{G}_0(w)$ by verifying that every $g_{\alpha_1,\dots,\alpha_{M-1}}(\cdot) \in \mathcal{U}$ satisfies the conditions in (4.2). Indeed, $g_{\alpha_1,\dots,\alpha_{M-1}}(0) = 0$ because the zero signal is either in no ball or in $\mathcal{S}_M$. Next, we show that $|g_{\alpha_1,\dots,\alpha_{M-1}}(x) - g_{\alpha_1,\dots,\alpha_{M-1}}(x')| \leq \|x - x'\|_w$, $\forall x, x' \in \mathcal{S}^-$. This will be done by distinguishing cases. First, assume that x and x' are contained in the same ball $\mathcal{S}_\ell$, for some $\ell \in \{1, \dots, M-1\}$. Then, we

have

$$\begin{aligned} |g_{\alpha_1,\ldots,\alpha_{M-1}}(x) - g_{\alpha_1,\ldots,\alpha_{M-1}}(x')| &= |2\alpha_\ell - 1| \cdot |\|x' - x_\ell\|_w - \|x - x_\ell\|_w| \\ &\leq \|(x' - x_\ell) - (x - x_\ell)\|_w = \|x' - x\|_w, \end{aligned} \tag{4.80}$$

where we used the reverse triangle inequality. Next, assume that $x \in \mathcal{S}_\ell$ and $x' \in \mathcal{S}_j$, with $\ell, j \in \{1, \ldots, M-1\}$ and $\ell \neq j$. We let $z(\mu) := x + \mu(x' - x),\ \mu \in [0,1]$. Since $z(0) = x \in \mathcal{S}_\ell$ and $z(1) = x' \notin \mathcal{S}_\ell$ (because the balls are non-overlapping), there must be a $\mu_1 \in (0,1)$ such that $\|z(\mu_1) - x_\ell\|_w = \epsilon'/2$. As $z(\mu_1) \notin \mathcal{S}_j$ and $z(1) = x' \in \mathcal{S}_j$, there must similarly be a $\mu_2 \in (\mu_1, 1)$ so that $\|z(\mu_2) - x_j\|_w = \epsilon'/2$. For notational simplicity, we let $z_1 := z(\mu_1)$, $z_2 := z(\mu_2)$, and $g := g_{\alpha_1,\ldots,\alpha_{M-1}}$. Next, we bound

$$\begin{aligned} |g(x) - g(x')| &\leq |g(x) - g(z_1)| + |g(z_1) - g(z_2)| + |g(z_2) - g(x')| \\ &\leq |g(x) - g(z_1)| + 0 + |g(z_2) - g(x')| && (4.81) \\ &\leq \|x - z_1\|_w + \|z_2 - x'\|_w \\ &= \|\mu_1(x - x')\|_w + \|(1 - \mu_2)(x - x')\|_w && (4.82) \\ &= (1 + \mu_1 - \mu_2)\|x - x'\|_w < \|x - x'\|_w, && (4.83) \end{aligned}$$

where in (4.81) we used that $g(z_1) = g(z_2) = 0$, in (4.C.4) we applied (4.80) upon noting that x, z_1 and z_2, x' are contained in the same ball each, in (4.82) we inserted the definition of z_1 and z_2, and in (4.83) we used $\mu_2 > \mu_1$. Finally, assume that $x \in \mathcal{S}_\ell$, for some $\ell \in \{1, \ldots, M-1\}$, and $x' \notin \mathcal{S}_j$, for all $j \in \{1, \ldots, M-1\}$. We let $z(\mu) := x + \mu(x' - x),\ \mu \in [0,1]$. Since $z(0) = x \in \mathcal{S}_\ell$ and $z(1) = x' \notin \mathcal{S}_\ell$, there must be a $\mu_1 \in (0,1)$ such that $\|z(\mu_1) - x_\ell\|_w = \epsilon'/2$. Again, we set $z_1 = z(\mu_1)$ and bound

$$\begin{aligned} |g(x) - g(x')| &\leq |g(x) - g(z_1)| + |g(z_1) - g(x')| \\ &\leq |g(x) - g(z_1)| + 0 && (4.84) \\ &\leq \|x - z_1\|_w = \|\mu_1(x - x')\|_w && (4.85) \\ &= \mu_1\|x - x'\|_w < \|x - x'\|_w, && (4.86) \end{aligned}$$

where in (4.84) we used that $g(z_1) = g(x') = 0$, in (4.85) we applied (4.80) upon noting that x, z_1 are contained in the same ball, and in (4.86) we used $\mu_1 < 1$. We have thus established that $|g_{\alpha_1,\ldots,\alpha_{M-1}}(x) - g_{\alpha_1,\ldots,\alpha_{M-1}}(x')| \leq \|x - x'\|_w,\ \forall x, x' \in \mathcal{S}^-$, and hence $\mathcal{U} \subset \mathcal{G}_0(w)$.

Next, we show that for distinct $g_{\alpha_1,\ldots,\alpha_{M-1}}, g_{\beta_1,\ldots,\beta_{M-1}} \in \mathcal{U}$ (i.e., there is at least one $\ell \in \{1, \ldots, M-1\}$ such that $\alpha_\ell \neq \beta_\ell$), it holds that $\rho_0(g_{\alpha_1,\ldots,\alpha_{M-1}}, g_{\beta_1,\ldots,\beta_{M-1}}) > \epsilon$. Indeed, for any

such ℓ, we have

$$\rho_0(g_{\alpha_1,\ldots,\alpha_{M-1}}, g_{\beta_1,\ldots,\beta_{M-1}}) = \sup_{\widetilde{x}\in\mathcal{S}^-} |g_{\alpha_1,\ldots,\alpha_{M-1}}(\widetilde{x}) - g_{\beta_1,\ldots,\beta_{M-1}}(\widetilde{x})| \tag{4.87}$$

$$\geq |g_{\alpha_1,\ldots,\alpha_{M-1}}(x_\ell) - g_{\beta_1,\ldots,\beta_{M-1}}(x_\ell)| \tag{4.88}$$

$$= |2(\alpha_\ell - \beta_\ell)\epsilon'/2| = \epsilon' > \epsilon,$$

where in (4.87) we used (4.4) and in (4.88) we inserted the particular choice $\widetilde{x} = x_\ell$ to lower-bound the sup. This establishes that $\mathcal{U}$, indeed, constitutes an ϵ-packing for $(\mathcal{G}_0(w), \rho_0)$ and we therefore have

$$\log M(\epsilon; \mathcal{G}_0(w), \rho_0) \geq \log|\mathcal{U}| = M - 1 = \left(\prod_{\ell=0}^{T} \left\lceil \frac{2Dw[\ell]}{\epsilon} \right\rceil\right) - 1.$$

□

4.C.5. Proof of Lemma 4.4.2

Note that $w_{a,b}^{(e)}[t] > \epsilon/d$ gives $t < \frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)}$, and thereby

$$T = \max\left\{ t \in \mathbb{N}_0 \;\middle|\; t < \frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)} \right\} = \left\lceil \frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)} \right\rceil - 1. \tag{4.89}$$

Next, we have

$$\log\left(\prod_{\ell=0}^{T} \frac{cw_{a,b}^{(e)}[\ell]}{\epsilon}\right) = \log\left(\prod_{\ell=0}^{T} \frac{ace^{-b\ell}}{\epsilon}\right)$$

$$= (T+1)\log(\epsilon^{-1}) - b\log(e)\frac{T(T+1)}{2} + (T+1)\log(ac)$$

$$= \left\lceil \frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)} \right\rceil \log(\epsilon^{-1}) - b\log(e)\frac{\left\lceil \frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)} \right\rceil \left(\left\lceil \frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)} \right\rceil - 1 \right)}{2}$$

$$+ \left\lceil \frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)} \right\rceil \log(ac).$$

Assuming that ϵ is sufficiently small to guarantee that $\frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)} - 1 \geq 0$, we can upper-bound (4.C.5) according to

$$\begin{aligned}
&\log\left(\prod_{\ell=0}^{T}\frac{cw_{a,b}^{(e)}[\ell]}{\epsilon}\right)\\
&\leq \left(\frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)}+1\right)\log(\epsilon^{-1}) - b\log(e)\frac{\frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)}\left(\frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)}-1\right)}{2} + \left(\frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)}+1\right)\log(ac)\\
&= \frac{1}{2b\log(e)}\log^2(\epsilon^{-1}) + \left(\frac{\log(ad)}{b\log(e)}+1-\frac{1}{2b\log(e)}\left(2\log(ad)-b\log(e)\right)\right.\\
&\quad \left.+\frac{\log(ac)}{b\log(e)}\right)\log(\epsilon^{-1}) - \frac{\log(ad)\left(\log(ad)-b\log(e)\right)}{2b\log(e)} + \log(ac)\left(\frac{\log(ad)}{b\log(e)}+1\right)\\
&= \frac{1}{2b\log(e)}\log^2(\epsilon^{-1}) + o\left(\log^2(\epsilon^{-1})\right).
\end{aligned}$$

In the same spirit, we can lower-bound (4.C.5) as

$$\begin{aligned}
&\log\left(\prod_{\ell=0}^{T}\frac{cw_{a,b}^{(e)}[\ell]}{\epsilon}\right)\\
&\geq \left(\frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)}\right)\log(\epsilon^{-1}) - b\log(e)\frac{\frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)}\left(\frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)}+1\right)}{2} + \left(\frac{\log\left(\frac{ad}{\epsilon}\right)}{b\log(e)}\right)\log(ac)\\
&= \frac{1}{2b\log(e)}\log^2(\epsilon^{-1}) + \left(\frac{\log(ad)}{b\log(e)}-\frac{1}{2b\log(e)}\left(2\log(ad)+b\log(e)\right)\right.\\
&\quad \left.+\frac{\log(ac)}{b\log(e)}\right)\log(\epsilon^{-1}) - \frac{\log(ad)\left(\log(ad)+b\log(e)\right)}{2b\log(e)} + \log(ac)\left(\frac{\log(ad)}{b\log(e)}\right)\\
&= \frac{1}{2b\log(e)}\log^2(\epsilon^{-1}) + o\left(\log^2(\epsilon^{-1})\right).
\end{aligned}$$

Finally, combining (4.C.5) and (4.C.5) yields

$$\log\left(\prod_{\ell=0}^{T}\frac{cw_{a,b}^{(e)}[\ell]}{\epsilon}\right) = \frac{1}{2b\log(e)}\log^2(\epsilon^{-1}) + o\left(\log^2(\epsilon^{-1})\right),$$

as desired. □

4.C.6. Proof of Lemma 4.4.5

We start by noting that

$$T = \max\left\{t\in\mathbb{N}_0 \;\middle|\; w_{p,q}^{(p)} > \frac{\epsilon}{d}\right\} = \left\lceil\left(\frac{dq}{\epsilon}\right)^{1/p}\right\rceil - 2. \tag{4.90}$$

Next,

$$\begin{aligned}
\log\left(\prod_{\ell=0}^{T}\frac{cw_{p,q}^{(p)}[\ell]}{\epsilon}\right) &= (T+1)\log(c)+(T+1)\log(\epsilon^{-1})+\sum_{\ell=0}^{T}\log\left(w_{p,q}^{(p)}[\ell]\right)\\
&= (T+1)\log(cq)+(T+1)\log(\epsilon^{-1})-p\log((T+1)!)\\
&= (T+1)\log(cq)+(T+1)\log(\epsilon^{-1})-p\left((T+1)\log(T+1)\right.\\
&\quad \left.-(T+1)\log(e)\right)+O(\log(T+1)) \qquad (4.91)\\
&= (T+1)\log(cqe^{p})+(T+1)\left(\log(\epsilon^{-1})-p\log\left(\left\lceil\left(\frac{dq}{\epsilon}\right)^{1/p}\right\rceil-1\right)\right)\\
&\quad +O(\log(\epsilon^{-1})), \qquad (4.92)
\end{aligned}$$

where (4.91) follows from the logarithm form of Stirling's approximation, namely

$$\log(n!)=n\log(n)-n\log(e)+O(\log(n)).$$

Now, assuming that ϵ is sufficiently small to guarantee that $1\le\frac{1}{2}(dq/\epsilon)^{1/p}$, applying $\log\left((dq/\epsilon)^{1/p}-1\right)\ge\log\left(\frac{1}{2}(dq/\epsilon)^{1/p}\right)\ge 0$, we can upper-bound (4.92) as

$$\begin{aligned}
&(T+1)\log(cqe^{p})+(T+1)\left(\log(\epsilon^{-1})-p\log\left(\left(\frac{dq}{\epsilon}\right)^{1/p}-1\right)\right)+O(\log(\epsilon^{-1}))\\
&\le (T+1)\log(cqe^{p})+(T+1)\left(\log(\epsilon^{-1})-p\log\left(\frac{1}{2}\left(\frac{dq}{\epsilon}\right)^{1/p}\right)\right)+O(\log(\epsilon^{-1}))\\
&= (T+1)(\log(ce^{p}/d)+p)+O(\log(\epsilon^{-1}))\\
&= \left(\left\lceil\left(\frac{dq}{\epsilon}\right)^{1/p}\right\rceil-1\right)(\log(ce^{p}/d)+p)+O(\log(\epsilon^{-1}))\\
&\le \left(\frac{dq}{\epsilon}\right)^{1/p}(\log(ce^{p}/d)+p)+O(\log(\epsilon^{-1})). \qquad (4.93)
\end{aligned}$$

Furthermore, we can lower-bound (4.92) according to

$$\begin{aligned}
&(T+1)\log(cqe^p) + (T+1)\left(\log(\epsilon^{-1}) - p\log\left(\left(\frac{dq}{\epsilon}\right)^{1/p}\right)\right) + O(\log(\epsilon^{-1})) \\
&= (T+1)\log(ce^p/d) + O(\log(\epsilon^{-1})) \\
&= \left(\left\lceil\left(\frac{dq}{\epsilon}\right)^{1/p}\right\rceil - 1\right)\log(ce^p/d) + O(\log(\epsilon^{-1})) \\
&\geq \left(\left(\frac{dq}{\epsilon}\right)^{1/p} - 1\right)\log(ce^p/d) + O(\log(\epsilon^{-1})) \\
&= \left(\frac{dq}{\epsilon}\right)^{1/p}\log(ce^p/d) + O(\log(\epsilon^{-1})).
\end{aligned} \tag{4.94}$$

Combining (4.92), (4.93), and (4.94) yields the desired result

$$\log\left(\prod_{\ell=0}^{T}\frac{cw_{p,q}^{(p)}[\ell]}{\epsilon}\right) = \Theta(\epsilon^{-1/p}).$$

□

4.C.7. Proof of Lemma 4.4.6

Consider $\epsilon \in (0, \epsilon_0)$ with $\epsilon_0 = \frac{Dw_{p,q}^{(p)}[0]}{2} = \frac{aD}{2}$. By Theorem 4.3.12, the exterior covering number of $(\mathcal{G}(w_{p,q}^{(p)}), \rho)$ satisfies

$$\left(\prod_{\ell=0}^{T'}\left\lceil\frac{Dw_{p,q}^{(p)}[\ell]}{\epsilon}\right\rceil\right) - 1 \leq \log N^{\text{ext}}(\epsilon; \mathcal{G}(w_{p,q}^{(p)}), \rho_*) \leq \log(3)\prod_{\ell=0}^{T''}\left(2\left\lceil\frac{4Dw_{p,q}^{(p)}[\ell]}{\epsilon}\right\rceil + 1\right), \tag{4.95}$$

where

$$T' := \max\left\{\ell \in \mathbb{N}_0 \;\middle|\; w_{p,q}^{(p)}[\ell] > \frac{\epsilon}{D}\right\} \text{ and } T'' := \max\left\{\ell \in \mathbb{N}_0 \;\middle|\; w_{p,q}^{(p)}[\ell] > \frac{\epsilon}{4D}\right\}.$$

We can further lower-bound the left-most term in (4.95) according to

$$\begin{aligned}
\left(\prod_{\ell=0}^{T'}\left\lceil\frac{Dw_{p,q}^{(p)}[\ell]}{\epsilon}\right\rceil\right) - 1 &\geq \left(\prod_{\ell=0}^{T'}\frac{Dw_{p,q}^{(p)}[\ell]}{\epsilon}\right) - 1 \\
&\geq \frac{1}{2}\prod_{\ell=0}^{T'}\frac{Dw_{p,q}^{(p)}[\ell]}{\epsilon},
\end{aligned} \tag{4.96}$$

where (4.96) follows from

$$\frac{1}{2}\prod_{\ell=0}^{T'}\frac{Dw_{p,q}^{(p)}[\ell]}{\epsilon}\geq 1,$$

which, in turn, is a consequence of

$$\begin{aligned}\frac{Dw_{p,q}^{(p)}[0]}{\epsilon}&\geq\frac{Dw_{p,q}^{(p)}[0]}{\epsilon_0}=2,\\ \frac{Dw_{p,q}^{(p)}[\ell]}{\epsilon}&\geq\frac{Dw_{p,q}^{(p)}[T']}{\epsilon}>1,\quad\text{for}\quad\ell\in[\![T']\!]\setminus\{0\}.\end{aligned}$$

Similarly, we can further upper-bound the right-most term in (4.95) as

$$\begin{aligned}\log(3)\left(\prod_{\ell=0}^{T''}\left(2\left\lceil\frac{4Dw_{p,q}^{(p)}[\ell]}{\epsilon}\right\rceil+1\right)\right)&\leq\log(3)\left(\prod_{\ell=0}^{T''}\left(\frac{8Dw_{p,q}^{(p)}[\ell]}{\epsilon}+3\right)\right)\\ &\leq\log(3)\left(\prod_{\ell=0}^{T''}\frac{20Dw_{p,q}^{(p)}[\ell]}{\epsilon}\right),\end{aligned}\tag{4.97}$$

where (4.97) follows from

$$\frac{4Dw_{p,q}^{(p)}[\ell]}{\epsilon}\geq\frac{4Dw_{p,q}^{(p)}[T'']}{\epsilon}>1,\quad\text{for}\quad\ell\in[\![T'']\!].$$

Combining (4.95)–(4.97), taking logarithms two more times, dividing the results by $\log(\epsilon^{-1})$, and applying Lemma 4.4.5, we obtain

$$\frac{\log\left(\Theta(\epsilon^{-1/p})\right)}{\log(\epsilon^{-1})}\leq\frac{\log^{(3)}N^{\mathrm{ext}}(\epsilon;\mathcal{G}(w_{p,q}^{(p)}),\rho_*)}{\log(\epsilon^{-1})}\leq\frac{\log\left(\Theta(\epsilon^{-1/p})\right)}{\log(\epsilon^{-1})}.$$

Taking the limit $\epsilon\to 0$, then yields

$$\lim_{\epsilon\to 0}\frac{\log^{(3)}N^{\mathrm{ext}}(\epsilon;\mathcal{G}(w_{p,q}^{(p)}),\rho_*)}{\log(\epsilon^{-1})}=\frac{1}{p},$$

which implies that $(\mathcal{G}(w_{p,q}^{(p)}),\rho_*)$ is of order 2 and type 1, with generalized dimension

$$\mathfrak{d}=\frac{1}{p}.$$

□

4.C.8. Auxiliary results on ReLU networks

Lemma 4.C.2. *For $d \in \mathbb{N}_+$ with $d \geq 2$, consider the following functions,*

$$\begin{aligned} f_d^{min}(z) &:= \min\{z_1, z_2, \ldots, z_d\}, \qquad \text{for } z \in \mathbb{R}^d, \\ f_d^{max}(z) &:= \max\{z_1, z_2, \ldots, z_d\}, \qquad \text{for } z \in \mathbb{R}^d. \end{aligned}$$

Then, there exist ReLU networks $\Phi_d^{min} \in \mathcal{N}_{d,1}$ and $\Phi_d^{max} \in \mathcal{N}_{d,1}$, with $\mathcal{L}(\Phi_d^{min}) = \mathcal{L}(\Phi_d^{max}) = \lceil \log(d+1) \rceil + 1$, $\mathcal{M}(\Phi_d^{min}) = \mathcal{M}(\Phi_d^{max}) \leq 14d - 9$, $\mathcal{W}(\Phi_d^{min}) = \mathcal{W}(\Phi_d^{max}) \leq 3d$, $\mathcal{K}(\Phi_d^{min}) = \mathcal{K}(\Phi_d^{max}) = \{1, -1\}$, $\mathcal{B}(\Phi_d^{min}) = \mathcal{B}(\Phi_d^{max}) = 1$, such that

$$\begin{aligned} \Phi_d^{min}(z) &= f_d^{min}(z), \qquad \text{for all} \quad z \in \mathbb{R}^d, \\ \Phi_d^{max}(z) &= f_d^{max}(z), \qquad \text{for all} \quad z \in \mathbb{R}^d. \end{aligned}$$

Proof. We only need to show the result for the max function as $\min\{z_1, z_2, \ldots, z_d\} = -\max\{-z_1, -z_2, \ldots, -z_d\}$. First, we realize $\max\{x_1, x_2\}$ according to

$$\max\{x_1, x_2\} = x_1 + \rho(x_2 - x_1) \tag{4.98}$$

by the ReLU network

$$\widehat{\Phi} = \widehat{W}_2 \circ \rho \circ \widehat{W}_1, \tag{4.99}$$

with

$$\begin{aligned} \widehat{W}_1(x) &= \begin{pmatrix} 1 & 0 \\ -1 & 0 \\ -1 & 1 \end{pmatrix} x =: A_1 x, \\ \widehat{W}_2(x) &= \begin{pmatrix} 1 & -1 & 1 \end{pmatrix} x =: A_2 x. \end{aligned} \tag{4.100}$$

Now, for arbitrary $d \in \mathbb{N}_+$ with $d \geq 2$, choose $\ell \in \mathbb{N}_+$ such that $2^\ell \leq d < 2^{\ell+1}$. Then, we double the first $k = 2^{\ell+1} - d$ elements and retain the remaining $d - k$ elements as follows

$$\max\{x_1, \ldots, x_d\} = \max\{x_1, x_1, \ldots, x_k, x_k, x_{k+1}, x_{k+2}, \ldots, x_d\}.$$

The primary reason for doubling elements is that by extending the set $\{x_1, \ldots, x_d\}$ to a set whose cardinality is a power of 2, we can utilize (4.98) together with a divide-and-conquer approach to determine the maximum value of the set $\{x_1, \ldots, x_d\}$. This then leads to a ReLU network realization of depth scaling logarithmically in d. Now, let $W_{\ell+1}(x) := A_2 x$, $W_j(x) := B_j x$, for

$j \in \{\ell, \ell-1, \ldots, 1\}$, and $W_0(x) := A_0 x$, with

$$\begin{aligned} B_o &= A_1 \operatorname{diag}(A_2, A_2) \\ &= \begin{pmatrix} 1 & -1 & 1 & 0 & 0 & 0 \\ -1 & 1 & -1 & 0 & 0 & 0 \\ -1 & 1 & -1 & 1 & -1 & 1 \end{pmatrix}, \\ B_j &= \operatorname{diag}(\underbrace{B_o, \ldots, B_o}_{2^{\ell-j}}), \\ A_0 &= \operatorname{diag}(\underbrace{A_1, \ldots, A_1}_{2^\ell}) \operatorname{diag}(\underbrace{\mathbb{1}_2, \ldots, \mathbb{1}_2}_{k}, \mathbb{I}_{d-k}) \\ &= \operatorname{diag}(\underbrace{A_1\mathbb{1}_2, \ldots, A_1\mathbb{1}_2}_{k}, \underbrace{A_1, \ldots, A_1}_{2^\ell - k}), \\ A_1\mathbb{1}_2 &= (1, -1, 0)^T. \end{aligned} \tag{4.101}$$

Combining (4.98)-(4.101), we can show that

$$\Phi_d^{max} := W_{\ell+1} \circ \rho \circ W_\ell \circ \rho \circ \ldots \rho \circ W_1 \circ \rho \circ W_0 = f_d^{\max} \tag{4.102}$$

as follows:

(i) We double the first $k = 2^{\ell+1} - d$ elements and retain the remaining $d - k$ elements of $\{x_1, x_2, \ldots, x_d\}$ according to

$$\begin{aligned} &\begin{pmatrix} x_1 & x_1 & \ldots & x_k & x_k & x_{k+1} & x_{k+2} & \ldots & x_d \end{pmatrix}^T \\ &= \operatorname{diag}(\underbrace{\mathbb{1}_2, \ldots, \mathbb{1}_2}_{k}, \mathbb{I}_{d-k}) \begin{pmatrix} x_1 & x_2 & \ldots & x_d \end{pmatrix}^T. \end{aligned}$$

For simplicity, we denote the resulting set $\{x_1, x_1, \ldots, x_k, x_k, x_{k+1}, x_{k+2}, \ldots, x_d\}$ as $\{y_1, y_2, \ldots, y_{2^{\ell+1}}\}$.

(ii) Application of (4.99) to the 2^ℓ pairs in $\{y_1, y_2, \ldots, y_{2^{\ell+1}}\}$, results in

$$\{\max\{y_1, y_2\}, \max\{y_3, y_4\}, \ldots, \max\{y_{2^{\ell+1}-1}, y_{2^{\ell+1}}\}\}$$

and hence reduces the number of elements from $2^{\ell+1}$ to 2^ℓ. This reduction is applied iteratively until we get $\max\{x_1, x_2, \ldots, x_d\}$. The compositions $A_1 \circ \operatorname{diag}(A_2, A_2)$, formed in the iterative application of (4.99), constitute the main diagonal elements of the matrices B_k, $k \in \{\ell, \ell-1, \ldots, 1\}$. We illustrate this part of the procedure using the simplest

possible example, namely for $\ell = 1$ and hence $\max\{y_1, y_2, y_3, y_4\}$. Specifically, we have

$$\begin{aligned}
&\max\{y_1, y_2, y_3, y_4\} = \max\{\max\{y_1, y_2\}, \max\{y_3, y_4\}\} \\
&= \max\left\{\operatorname{diag}(A_2, A_2)\,\rho\left(\operatorname{diag}(A_1, A_1)\,((y_1, y_2), (y_3, y_4))^T\right)\right\} \\
&= A_2\rho\left(A_1 \operatorname{diag}(A_2, A_2)\,\rho\left(\operatorname{diag}(A_1, A_1)\,((y_1, y_2), (y_3, y_4))^T\right)\right) \\
&= (W_2 \circ \rho \circ W_1 \circ \rho \circ W_0)(y).
\end{aligned}$$

The proof is concluded by noting that

$$\begin{aligned}
\mathcal{L}(\Phi_d^{\max}) &\overset{(4.102)}{=} \ell + 1 = \lceil \log(d+1) \rceil + 1, \\
\mathcal{M}(\Phi_d^{\max}) &\overset{(4.102)}{=} \sum_{j=0}^{\ell+1} \mathcal{M}(W_j) = \mathcal{M}(A_2) + \sum_{j=1}^{\ell} \mathcal{M}(B_j) + \mathcal{M}(A_0) \\
&\overset{(4.101),\,(4.100)}{=} 3 + 2k + 4(2^\ell - k) + 12 \sum_{j=1}^{\ell} 2^{\ell-j} \leq 14d - 9, \\
\mathcal{W}(\Phi_d^{\max}) &\overset{(4.102)}{=} \max_{j=0,1\ldots,\ell+1} \mathcal{W}(W_j) \overset{(4.101),\,(4.100)}{=} 3 \cdot 2^\ell \leq 3d, \\
\mathcal{K}(\Phi_d^{\max}) &\overset{(4.102)}{\subset} \bigcup_{j=0}^{\ell+1} \mathcal{K}(W_j) \overset{(4.101),\,(4.100)}{=} \{1, -1\}, \\
\mathcal{B}(\Phi_d^{\max}) &= \max_{b \in \mathcal{K}(\Phi_d^{\max})} |b| = 1.
\end{aligned}$$

□

Lemma 4.C.3 (Composition of ReLU networks [Elb+21])**.** *For $i = 1, \ldots, n$, let $d_i \in \mathbb{N}_+$ and $\Phi_i \in \mathcal{N}_{d_i, d_{i+1}}$. Then, there exists a network $\Psi \in \mathcal{N}_{d_1, d_{n+1}}$ with*

$$\Psi(x) = (\Phi_n \circ \Phi_{n-1} \circ \cdots \circ \Phi_1)(x), \quad \textit{for all } x \in \mathbb{R}^{d_1},$$

satisfying

$$\begin{aligned}
\mathcal{L}(\Psi) &= \sum_{i=1}^{n} \mathcal{L}(\Phi_i), \\
\mathcal{M}(\Psi) &\leq 2 \sum_{i=1}^{n} \mathcal{M}(\Phi_i), \\
\mathcal{W}(\Psi) &\leq \max\left\{\max_{i=1,\ldots,n-1}\{2d_i\}, \max_{i=1,\ldots,n}\{\mathcal{W}(\Phi_i)\}\right\}, \\
\mathcal{K}(\Psi) &\subset \bigcup_{i=1}^{n} (\mathcal{K}(\Phi_i) \cup (-\mathcal{K}(\Phi_i))), \\
\mathcal{B}(\Psi) &= \max_{i=1,\ldots,n} \mathcal{B}(\Phi_i).
\end{aligned}$$

Proof. Follows along the same lines as the proof of [Elb+21, Lemma II.3]. □

Lemma 4.C.4 (Parallelization of ReLU networks of the same depth [Elb+21]). *For $i = 1, \ldots, n$, let $d_i, d_i' \in \mathbb{N}_+$ and $\Phi_i \in \mathcal{N}_{d_i, d_i'}$ with $\mathcal{L}(\Phi_i) = L$. Then, there exists a ReLU network $P(\Phi_1, \Phi_2, \ldots, \Phi_n) \in \mathcal{N}_{\sum_{i=1}^n d_i, \sum_{i=1}^n d_i'}$ with*

$$P(\Phi_1, \Phi_2, \ldots, \Phi_n)(x) = (\Phi_1(x), \Phi_2(x), \ldots, \Phi_n(x))^T, \quad \text{for all } x \in \mathbb{R}^{\sum_{i=1}^n d_i},$$

satisfying

$$\begin{aligned}
\mathcal{L}(P(\Phi_1, \Phi_2, \ldots, \Phi_n)) &= L, \\
\mathcal{M}(P(\Phi_1, \Phi_2, \ldots, \Phi_n)) &= \sum_{i=1}^n \mathcal{M}(\Phi_i), \\
\mathcal{W}(P(\Phi_1, \Phi_2, \ldots, \Phi_n)) &\leq \sum_{i=1}^n \mathcal{W}(\Phi_i), \\
\mathcal{K}(P(\Phi_1, \Phi_2, \ldots, \Phi_n)) &= \bigcup_{i=1}^n \mathcal{K}(\Phi_i), \\
\mathcal{B}(P(\Phi_1, \Phi_2, \ldots, \Phi_n)) &= \max_{i=1,\ldots,n} \mathcal{B}(\Phi_i).
\end{aligned}$$

Proof. Follows along similar lines as the proof of [Elb+21, Lemma II.5]. □

Lemma 4.C.5 (Augmenting network depth [Elb+21]). *Let $d_1, d_2, K \in \mathbb{N}_+$, and $\Phi \in \mathcal{N}_{d_1, d_2}$ with $\mathcal{L}(\Phi) < K$. Then, there exists a network $\Psi \in \mathcal{N}_{d_1, d_2}$ with*

$$\begin{aligned}
\mathcal{L}(\Psi) &= K, \\
\mathcal{M}(\Psi) &\leq \mathcal{M}(\Phi) + d_2 \mathcal{W}(\Phi) + 2d_2(K - \mathcal{L}(\Phi)), \\
\mathcal{W}(\Psi) &= \max\{2d_2, \mathcal{W}(\Phi)\}, \\
\mathcal{K}(\Psi) &\subset (\mathcal{K}(\Phi) \cup (-\mathcal{K}(\Phi)) \cup \{1, -1\}), \\
\mathcal{B}(\Psi) &= \max\{1, \mathcal{B}(\Phi)\},
\end{aligned}$$

satisfying $\Psi(x) = \Phi(x)$, for all $x \in \mathbb{R}^{d_1}$.

Proof. Follows along the same lines as the proof of [Elb+21, Lemma II.4]. □

4.C.9. Remainder of the proof of Lemma 4.5.2

We verify that $\Phi = \Phi_2 \circ \Phi_1$ defined in (4.35) has $(2, \epsilon)$-quantized weights.

- The weights in $\Phi_2 = \widetilde{W}^{\Sigma}$ defined in (4.32):

$$\begin{aligned}
\widetilde{g}_n &\overset{(4.23)}{=} \mathcal{Q}_{2,\epsilon}(g(\widetilde{x}_n)) \overset{\text{Definition 4.2.2}}{=} \left\lfloor g(\widetilde{x}_n)/2^{-2\lceil \log(\epsilon^{-1})\rceil} \right\rfloor \cdot 2^{-2\lceil \log(\epsilon^{-1})\rceil} \\
&\in 2^{-2\lceil \log(\epsilon^{-1})\rceil}\mathbb{Z}, \\
|\widetilde{g}_n| &\leq |g(\widetilde{x}_n)| + 2^{-2\lceil \log(\epsilon^{-1})\rceil} \overset{(4.2)}{\leq} D + 2^{-2\lceil \log(\epsilon^{-1})\rceil} \overset{(4.111)}{\leq} D + \epsilon^2 \\
&\leq D + 1 \overset{(4.111)}{\leq} \epsilon^{-2},
\end{aligned}$$

  which yields

$$\mathcal{K}(\Phi_2) \subset 2^{-2\lceil \log(\epsilon^{-1})\rceil}\mathbb{Z} \cap \left[-\epsilon^{-2}, \epsilon^{-2}\right]. \tag{4.103}$$

- The weights in Φ_1 defined in (4.34):

$$\mathcal{K}(\Phi_1) = \bigcup_{n\in\widetilde{\mathfrak{N}}} \left(\mathcal{K}(\Phi^n_{1,2}) \cup \mathcal{K}(\Phi^n_{1,1})\right) = \mathcal{K}(\Psi) \cup \left(\bigcup_{n\in\widetilde{\mathfrak{N}}} \mathcal{K}(\widehat{W}_n)\right). \tag{4.104}$$

  Recalling that Ψ realizes the spike function and applying Lemma 4.5.1, we obtain

$$\mathcal{K}(\Psi) = \{1, -1\} \subset 2^{-2\lceil \log(\epsilon^{-1})\rceil}\mathbb{Z} \cap \left[-\epsilon^{-2}, \epsilon^{-2}\right]. \tag{4.105}$$

  Based on (4.32), we get

$$\bigcup_{n\in\widetilde{\mathfrak{N}}} \mathcal{K}(\widehat{W}_n) = \bigcup_{\ell=0}^{T} \left(\bigcup_{n\in\widetilde{\mathfrak{N}}} \{\widetilde{\delta}_\ell^{-1}, n_\ell\}\right). \tag{4.106}$$

  Moreover, for all $\ell = 0, \ldots, T$ and $n \in \widetilde{\mathfrak{N}}$, we have

$$\begin{aligned}
\widetilde{\delta}_\ell^{-1} &\overset{(4.22)}{=} \mathcal{Q}_{2,\epsilon}(\delta_\ell^{-1}) \overset{\text{Definition 4.2.2}}{=} \left\lfloor \frac{s+1}{s}\frac{w[\ell]}{\epsilon} \Big/ 2^{-2\lceil \log(\epsilon^{-1})\rceil} \right\rfloor \cdot 2^{-2\lceil \log(\epsilon^{-1})\rceil} \\
&\in 2^{-2\lceil \log(\epsilon^{-1})\rceil}\mathbb{Z}\,, \\
\left|\widetilde{\delta}_\ell^{-1}\right| &\leq \frac{s+1}{s}\frac{w[\ell]}{\epsilon} + 2^{-2\lceil \log(\epsilon^{-1})\rceil} \overset{\text{Definition 4.2.18}}{\leq} \frac{s+1}{s}\epsilon^{-1} + \epsilon^2 \overset{(4.111)}{\leq} \epsilon^{-2}.
\end{aligned}$$

  Hence,

$$\widetilde{\delta}_\ell^{-1} \in 2^{-2\lceil \log(\epsilon^{-1})\rceil}\mathbb{Z} \cap \left[-\epsilon^{-2}, \epsilon^{-2}\right], \tag{4.107}$$

and

$$\begin{aligned} n_\ell &\in \mathbb{Z} \subset 2^{-2\lceil \log(\epsilon^{-1}) \rceil}\mathbb{Z}\,, \\ |n_\ell| &\leq \widetilde{N}_\ell \overset{(4.21)}{=} \left\lceil \frac{D}{\widetilde{\delta}_\ell} \right\rceil \leq D \left| \widetilde{\delta}_\ell^{-1} \right| + 1 \\ &\leq D \left(\frac{s+1}{s} \epsilon^{-1} + \epsilon^2 \right) + 1 \\ &\overset{(4.111)}{\leq} \left(D \frac{s+1}{s} + D + 1 \right) \epsilon^{-1} \overset{(4.111)}{\leq} \epsilon^{-2}, \end{aligned}$$

which yields

$$n_\ell \in 2^{-2\lceil \log(\epsilon^{-1}) \rceil}\mathbb{Z} \cap \left[-\epsilon^{-2}, \epsilon^{-2} \right]. \tag{4.108}$$

Based on (4.106), (4.107), and (4.108), we have

$$\bigcup_{n \in \widetilde{\mathfrak{N}}} \mathcal{K}(\widehat{W}_n) \subset 2^{-2\lceil \log(\epsilon^{-1}) \rceil}\mathbb{Z} \cap \left[-\epsilon^{-2}, \epsilon^{-2} \right]. \tag{4.109}$$

Combining (4.104), (4.105), (4.106), and (4.109) yields

$$\mathcal{K}(\Phi_1) \subset 2^{-2\lceil \log(\epsilon^{-1}) \rceil}\mathbb{Z} \cap \left[-\epsilon^{-2}, \epsilon^{-2} \right]. \tag{4.110}$$

Using (4.103) and (4.110), Lemma 4.C.3 shows that $\Phi = \Phi_2 \circ \Phi_1$, indeed, has $(2, \epsilon)$-quantized weights.

Finally, we compute $\mathcal{L}(\Phi)$ and derive an upper bound on $\mathcal{M}(\Phi)$ according to

$$
\begin{aligned}
\mathcal{L}(\Phi) &\overset{(4.35),\text{Lemma}4.C.3}{=} \mathcal{L}(\Phi_2) + \mathcal{L}(\Phi_1) \\
&\overset{(4.34),(4.33),\text{Lemma}4.C.4}{=} \mathcal{L}(\Psi) + 2 \\
&\overset{\text{Lemma}4.5.1}{=} \lceil \log(T+2) \rceil + 6, \\
\mathcal{M}(\Phi) &\overset{\text{Lemma 4.C.3}}{\leq} 2\left(\mathcal{M}(\Phi_2) + \mathcal{M}(\Phi_1)\right) \\
&\overset{\text{Lemma 4.C.4}}{=} 2\left(\mathcal{M}(\Phi_2) + \sum_{i=1}^{|\widetilde{\mathfrak{N}}|} \left(2\mathcal{M}(\Psi) + 2\mathcal{M}(\widehat{W}_{n^i})\right)\right) \\
&\overset{(4.32)\text{-}(4.35),\ \text{Lemma 4.5.1}}{\leq} 2|\widetilde{\mathfrak{N}}|\left(1 + 120(T+1) - 56 + 2(T+1)\right) \\
&\leq 244(T+1)|\widetilde{\mathfrak{N}}| \\
&\overset{(4.21)}{\leq} 244(T+1)\prod_{\ell=0}^{T}\left(2D\widetilde{\delta}_\ell^{-1} + 3\right) \\
&\overset{(4.21),(4.22)}{\leq} 244(T+1)\prod_{\ell=0}^{T}\left(2D\left(\frac{s+1}{s}\frac{w[\ell]}{\epsilon} + \epsilon^2\right) + 3\right) \\
&\overset{(4.111)}{\leq} 244(T+1)\prod_{\ell=0}^{T}\left(\frac{2Dw[\ell]}{\epsilon}\frac{s+1}{s} + 4\right).
\end{aligned}
$$

The proof is concluded by setting

$$
\epsilon_0 = \min\left\{1, \frac{1}{2}, \frac{1}{s+1}, \sqrt{\frac{1}{D+1}}, \left(D\frac{s+1}{s} + D + 1\right)^{-1}\right\}. \tag{4.111}
$$

□

Chapter 5

Recurrent neural networks universally approximate polynomials

5.1. Introduction

This chapter is inspired by the well-known universal approximation theorem for feedforward neural networks [Cyb89; Fun89; HSW89], which states that every continuous function can be approximated to within arbitrarily small error by a shallow neural network with sigmoidal activation function. Later quantitative results in the same vein have been established that also characterize the size of the approximating network [Bar93; Bar94].

In the last two decades, however, the ReLU activation function has gained the dominant position in the field of neural networks, and we therefore also use it in the present chapter. Starting with [Yar17], quantitative results for deep ReLU networks became available, also leading to Kolmogorov-Donoho type results in [Elb+21].

In general, all the mentioned results fit the following pattern: For every function f in a certain function class and every non-zero approximation error ϵ there exists a neural network $\mathcal{N}$ achieving this error. We can concisely sketch this approximation paradigm as

$$\forall f : \forall \epsilon : \exists \mathcal{N}, \text{ such that } \mathcal{N} \text{ approximates } f \text{ to within error } \epsilon. \tag{5.1}$$

Hence, one first has to decide on the desired approximation error ϵ and then the approximating network depends on ϵ, that is, its size, topology and weights. If later on a smaller error is desired a new, larger network with different topology and network weights needs to be instantiated. To make this dependence of network size on the approximation error concrete consider, for example, [Elb+21, Proposition III.5] which states that for every polynomial and every $\epsilon > 0$ there exists a deep ReLU network with $\mathcal{O}(\log \epsilon^{-1})$ weights that approximates the polynomial to within error ϵ.

In the present chapter, we propose a new approximation paradigm to remedy the dependence of the network on the approximation error ϵ. That is, we wish to exchange the order of the last

two quantifiers in (5.1) to yield

$$\forall f : \exists \mathcal{N}, \text{such that } \forall \epsilon : \ \mathcal{N} \text{ approximates } f \text{ to within error } \epsilon. \tag{5.2}$$

We will achieve this by iteratively applying a fixed transformation until the desired approximation error ϵ is achieved. If, later on, a smaller ϵ is required, it can be achieved by simply performing more iterations. This iterative process lends itself to a temporal formulation and we therefore formalize our results in terms of the well established recurrent neural network (RNN) architecture.

Definition 5.1.1 (RNN [Elm90; GBC16])**.** *We denote by $\rho : \mathbb{R} \to \mathbb{R}, \rho(x) := max(0, x)$ the ReLU function which acts component-wise, i.e., $\rho(x_1, \ldots, x_m) := (\rho(x_1), \ldots, \rho(x_m))$. An RNN with input dimension $d \in \mathbb{N}$, output dimension $d' \in \mathbb{N}$ and hidden state size $m \in \mathbb{N}$ is parametrized by matrices $A_h \in \mathbb{R}^{m \times m}$, $A_x \in \mathbb{R}^{m \times d}$, $A_o \in \mathbb{R}^{d' \times m}$, and vectors $b_h \in \mathbb{R}^m$, $b_o \in \mathbb{R}^{d'}$. We collectively call these quantities the weights of the RNN. They specify the hidden state operator $\mathcal{K} : (\mathbb{R}^d)^{\mathbb{N}_0} \to (\mathbb{R}^m)^{\mathbb{N}_0}$ mapping an input sequence $(x[t])_{t \geq 0}$ recursively to the sequence of hidden states $(h[t])_{t \geq 0}$ according to*

$$\begin{aligned} h[-1] &:= 0 \in \mathbb{R}^m \\ h[t] &= \rho(A_h h[t-1] + A_x x[t] + b_h), \end{aligned}$$

and the output mapping $\mathcal{Q} : \mathbb{R}^m \to \mathbb{R}^{d'}$

$$\mathcal{Q}(h) := A_o h + b_o.$$

The associated RNN is the operator $\mathcal{R} : (\mathbb{R}^d)^{\mathbb{N}_0} \to (\mathbb{R}^{d'})^{\mathbb{N}_0}$ given by

$$\mathcal{R} := \mathcal{Q}\mathcal{K},$$

where $\mathcal{Q}$ is applied independently at each time-step, i.e. for $x \in (\mathbb{R}^d)^{\mathbb{N}_0}$

$$(\mathcal{R}x)[t] = (\mathcal{Q}\mathcal{K}x)[t] = \mathcal{Q}\left((\mathcal{K}x)[t]\right), \quad \text{for } t \in \mathbb{N}_0.$$

We use the notation $\mathcal{M}_{\text{in}}(\mathcal{R}) = d$, $\mathcal{M}_{\text{out}}(\mathcal{R}) = \mathcal{M}_{\text{out}}(\mathcal{Q}) = d'$, $\mathcal{M}_{\text{hid}}(\mathcal{R}) = \mathcal{M}_{\text{hid}}(\mathcal{K}) = m$ to describe the size of the RNN.

To confirm with (5.2), we wish, for a given function f, to find an RNN $\mathcal{R}$ that achieves any arbitrarily small approximation error ϵ. Let now $x \in \mathbb{R}$ be the point at which we wish to approximate $f(x)$. As the input sequence to the RNN $\mathcal{R}$ we then use x at time $t = 0$ and zeros thereafter. To formalize this, we introduce the following operator.

Definition 5.1.2. *The map* $\mathcal{D} : \mathbb{R}^d \to (\mathbb{R}^d)^{\mathbb{N}_0}$ *is defined according to*

$$(\mathcal{D}x)[t] = x\mathbb{1}_{\{t=0\}} \qquad \forall t \geq 0.$$

Here, $\mathbb{1}_{\{\cdot\}}$ *denotes the truth function which takes on the value* 1 *if the statement inside* $\{\cdot\}$ *is true and equals* 0 *otherwise.*

As its output sequence, the RNN $\mathcal{R}$ produces increasingly precise approximations to $f(x)$. That is, the difference between $f(x)$ and the RNN output at time t is decreasing in t. Hence, we can operationalize the approximation paradigm (5.2) in terms of RNNs as

$$\forall f : \exists\, \mathcal{R} : \forall \epsilon : \exists t_0 : \sup_{t \geq t_0} \sup_x |(\mathcal{R}\mathcal{D}x)[t] - f(x)| < \epsilon. \tag{5.3}$$

Evidently, for any approximation theorem fitting this paradigm, the size, topology and weights of the approximating RNN $\mathcal{R}$ are independent of the approximation error ϵ, simply by virtue of the quantifier order in (5.3). Hence, networks in this paradigm always have the number of weights scaling as $\mathcal{O}(1)$ in ϵ. Only the runtime required to achieve the desired approximation error ϵ does depend on ϵ.

In the presentation of our proposed approximation paradigm, we have deliberately kept (5.1) to (5.3) vague to convey the central idea of reversing the quantifier order with as little clutter as possible. We believe that approximation results fitting paradigm (5.3) are possible for a variety of function classes, i.e., the set from which f is taken. To start research in this direction, we focus, in the present chapter, on approximating univariate polynomials. In order to reach this result we develop a number of building blocks that could prove useful in deriving similar approximation results for other function classes as well. For example, we think our results can easily be extended to multivariate polynomials. Similarly, we believe that existing research in deep neural network approximation [Huy25] might inspire approximation according to (5.3) for other function classes such as sinusoidals or refinable functions [Dau+23]. We next state the central result of the present chapter, which is that the class of univariate polynomials can be approximated by RNNs according to paradigm (5.3). Moreover, and perhaps surprisingly, the approximation error decreases exponentially in the runtime of the RNN.

Theorem 5.1.3. *Let* $N \in \mathbb{N}$, $a_0, \ldots, a_N \in \mathbb{R}$, *and* $D \geq 1$. *There exists an RNN* $\mathcal{R}_a^\pi$ *such that for* $t \geq 16 \log N$, *it holds that,*

$$\sup_{x \in [-D,D]} \left| (\mathcal{R}_a^\pi \mathcal{D}x)[t] - \sum_{i=0}^{N} a_i x^i \right| \leq \|a\|_1 C_1 4^{-C_2 t},$$

with $C_1 = 16ND^{2N}$ *and* $C_2 = \frac{1}{4\lceil \log N \rceil}$.

A more detailed version of this result is presented in Theorem 5.5.3. Next, we present some

commonly used notation. Thereafter, the remainder of the chapter is dedicated to the proof of Theorem 5.5.3, by building up the relevant building blocks one by one.

Notation $\mathbb{N}$ denotes positive integers, and $\mathbb{N}_0$ is here defined to include 0. $(\mathbb{R}^d)^{\mathbb{N}_0}$ denotes the space of sequences of vectors in $\mathbb{R}^d$ indexed by time in $\mathbb{N}_0$, that is for $x[\cdot] \in (\mathbb{R}^d)^{\mathbb{N}_0}$ we have $x[t] \in \mathbb{R}^d$ for $t \in \mathbb{N}_0$. Furthermore, for $x \in \mathbb{R}^d$, $\|x\|_\infty = \max_{i=1,\dots,d} |x_i|$ and $x \leq c$, for some $c > 0$, indicates $x_i \leq c$ for all $i \in \{1, \dots d\}$. $\mathbb{I}_d \in \mathbb{R}^{d \times d}$ is the d-dimensional identity matrix and $\mathbf{1}_d \in \mathbb{R}^d$ stands for the d-dimensional vector with all entries equal to one. log denotes the logarithm to base 2. Constants are understood to be in $\mathbb{R}^+$.

5.2. Approximating the squaring function and multiplication

Before going into the details of our RNN constructions we point out a simplification of the RNN definition.

Remark 5.2.1. *We shall be concerned with inputs of the form $(\mathcal{D}x)[t] = x\mathbb{1}_{\{t=0\}}$ for $x \in \mathbb{R}^d$. In this case, Definition 5.1.1 simplifies to*

$$\begin{aligned}(\mathcal{K}\mathcal{D}x)[0] &= \rho(A_x x + b_h)\\ (\mathcal{K}\mathcal{D}x)[t] &= \rho(A_h\,((\mathcal{K}\mathcal{D}x)[t-1]) + b_h), \qquad \text{for } t \in \mathbb{N},\end{aligned}$$

and

$$(\mathcal{R}\mathcal{D}x)[t] = (\mathcal{Q}\mathcal{K}\mathcal{D}x)[t], \qquad \text{for } t \in \mathbb{N}_0.$$

We start by constructing an RNN to approximate the squaring function. To this end we make use of the construction explained in [Elb+21]. We first recall the relevant results.

Lemma 5.2.2. *Let $F(x) \coloneqq x - x^2$ for $x \in [0,1]$. Further, with $m \in \mathbb{N}$, let $I_m : [0,1] \to [0,1]$ be the linear interpolation of F at points $\frac{k}{2^m}$, for $k \in \{0, \dots, 2^m\}$, that is,*

$$I_m\left(\frac{k}{2^m}\right) = F\left(\frac{k}{2^m}\right), \text{ for } k \in \{0, \dots, 2^m\},$$

and I_m is a linear function when restricted to the interval $\left[\frac{k}{2^m}, \frac{k+1}{2^m}\right]$ for each $k \in \{0, \dots, 2^m - 1\}$. It holds that,

$$\sup_{x \in [0,1]} |F(x) - I_m(x)| \leq 2^{-2m-2}.$$

Furthermore, define

$$s_\ell(\cdot) \coloneqq 2^{-1}\rho(\cdot) - \rho(\cdot - 2^{-2\ell-1}), \qquad \text{for } \ell \in \mathbb{N}_0,$$

and recursively $H_\ell = s_\ell \circ H_{\ell-1}$ *for* $\ell \in \mathbb{N}$ *with* $H_0 = s_0$*. Then, it holds that*

$$I_m(x) = \sum_{\ell=0}^{m-1} H_\ell(x), \qquad \text{for } x \in [0,1].$$

Proof. See proof of Proposition III.2 in [Elb+21]. □

We furthermore need the following simple property of I_m, which is not explicitly studied in [Elb+21].

Corollary 5.2.3. *We have, for all* $m \in \mathbb{N}$*, that*

$$0 \leq I_m(x) \leq x, \quad \forall x \in [0,1].$$

Proof. First, note that $F(x) \in [0, 1/4]$ for all $x \in [0,1]$. Thus, by definition of I_m as a linear interpolation, we also have $I_m(x) \in [0, 1/4]$ for all $x \in [0,1]$, and in particular $I_m(x) \geq 0$. Furthermore, since $F(x) = x - x^2$ is a concave function, we have $I_m(x) \leq F(x)$ for all $x \in [0,1]$, again due to I_m being a linear interpolation of F. This implies

$$I_m(x) \leq F(x) = x - x^2 \leq x$$

completing the proof. □

Next, we construct an RNN that approximates the squaring function.

Theorem 5.2.4. *For* $D \geq 1$*, there is an RNN* $\mathcal{R}^{Sq} = \mathcal{Q}^{Sq}\mathcal{K}^{Sq}$*, with* $\mathcal{M}_{\text{in}}(\mathcal{R}^{Sq}) = \mathcal{M}_{\text{out}}(\mathcal{R}^{Sq}) = 1$*, such that, for all* $x \in [-D, D]$ *and all* $t \in \mathbb{N}_0$*,*

$$|(\mathcal{R}^{Sq}\mathcal{D}x)[t] - x^2| \leq \frac{D^2}{4}\, 4^{-t},$$

as well as

$$\left\|(\mathcal{K}^{Sq}\mathcal{D}x)[t]\right\|_\infty \leq 1 \;\; \textit{and} \;\; 0 \leq (\mathcal{R}^{Sq}\mathcal{D}x)[t] \leq D^2.$$

The size of this RNN is $\mathcal{M}_{\text{hid}}(\mathcal{R}^{Sq}) = 7$*.*

Proof. The weights of the $\mathcal{R}^{\text{Sq}}$ are as follows:

$$A_h = \left(\begin{array}{cc|ccc|cc} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 1 & 2^{-1} & -1 & 0 & 0 & 0 \\ 1 & 1 & 2^{-1} & -1 & 0 & -1 & 0 \\ 1 & 1 & -2^{-1} & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 2^{-2} & -2^{-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right), \qquad b_h = \left(\begin{array}{c} 0 \\ 0 \\ \hline 0 \\ 0 \\ 0 \\ \hline 2^{-1} \\ 1 \end{array}\right),$$

$$A_x = \frac{1}{D}\left(\begin{array}{c} 1 \\ -1 \\ \hline 0 \\ 0 \\ 0 \\ \hline 0 \\ 0 \end{array}\right), \qquad A_o = D^2\Big(\begin{array}{cc|ccc|cc} 0 & 0 & -2^{-1} & 1 & 1 & 0 & 0 \end{array}\Big), \qquad b_o = 0.$$

Next, we fix $x \in [-D, D]$ arbitrarily, let $z \coloneqq \frac{|x|}{D}$ and compute the sequence $h[\cdot] \coloneqq (\mathcal{K}\mathcal{D}x)[\cdot]$ of hidden states for the input sequence $\widetilde{x} \coloneqq (\mathcal{D}x)[\cdot] = x\mathbb{1}_{\{\cdot=0\}}$ according to Definition 5.1.1. We start by proving through induction that

$$h[t] = \left(\begin{array}{c} 0 \\ 0 \\ \hline \rho(H_{t-2}(z)) \\ \rho(H_{t-2}(z) - 2^{-2t+1}) \\ z - \sum_{i=0}^{t-2} H_i(z) \\ \hline 2^{-2t-1} \\ 1 \end{array}\right), \qquad t \geq 2, \tag{5.4}$$

with H_t as defined in Lemma 5.2.2 for $t \in \mathbb{N}_0$. Starting from $h[-1] = 0$, we compute

$$\begin{aligned} h[0] &= \rho(A_h h[-1] + A_x \widetilde{x}[0] + b_h) \\ &= \rho(A_x x + b_h) \\ &= \left(\begin{array}{c} \rho(\frac{x}{D}) \\ \rho - \frac{x}{D}) \\ \hline 0 \\ 0 \\ 0 \\ \hline 2^{-1} \\ 1 \end{array}\right). \end{aligned}$$

Using $z = \rho\left(\frac{x}{D}\right) + \rho\left(-\frac{x}{D}\right) \geq 0$, we get

$$\begin{aligned} h[1] &= \rho(A_h h[0] + A_x \tilde{x}[1] + b_h) \\ &= \rho(A_h h[0] + b_h) \\ &= \left(\begin{array}{c} 0 \\ 0 \\ \hline z \\ \rho(z - 2^{-1}) \\ z \\ \hline 2^{-3} \\ 1 \end{array}\right) \end{aligned}$$

Further,

$$h[2] = \rho(A_h h[1] + b_h)$$
$$= \rho\left(\begin{array}{c} 0 \\ 0 \\ \hline 2^{-1}\rho(z) - \rho(z - 2^{-1}) \\ 2^{-1}\rho(z) - \rho(z - 2^{-1}) - 2^{-3} \\ z - (2^{-1}z - \rho(z - 2^{-1})) \\ \hline 2^{-2} \cdot 2^{-3} - 2^{-1} + 2^{-1} \\ 1 \end{array}\right) = \left(\begin{array}{c} 0 \\ 0 \\ \hline \rho(s_0(z)) \\ \rho(s_0(z) - 2^{-3}) \\ z - s_0(z) \\ \hline 2^{-2\cdot 2-1} \\ 1 \end{array}\right),$$

where $s_\ell(\cdot)$ was defined in Lemma 5.2.2 for $t \in \mathbb{N}_0$. Hence, using $H_0 = s_0$ as per Lemma 5.2.2, we established the base case of the induction for $t = 2$.

Next, we establish the induction step. To this end, we assume that (5.4) holds for some $t \geq 2$. Now, note that

$$
\begin{aligned}
h[t+1] &= \rho(A_h h[t] + b_h) \\
&= \rho\left(\begin{array}{c} 0 \\ 0 \\ \hline 2^{-1}\rho(H_{t-2}(z)) - \rho(H_{t-2}(z) - 2^{-2t+1}) \\ 2^{-1}\rho(H_{t-2}(z)) - \rho(H_{t-2}(z) - 2^{-2t+1}) - 2^{-2t-1} \\ z - \sum_{i=0}^{t-2} H_i(z) - (2^{-1}\rho(H_{t-2}(z)) - \rho(H_{t-2}(z) - 2^{-2t+1})) \\ \hline 2^{-2}\cdot 2^{-2t-1} - 2^{-1} + 2^{-1} \\ 1 \end{array}\right) \\
&= \rho\left(\begin{array}{c} 0 \\ 0 \\ \hline s_{t-1}(H_{t-2}(z)) \\ s_{t-1}(H_{t-2}(z)) - 2^{-2t-1} \\ z - \sum_{i=0}^{t-2} H_i(z) - s_{t-1}(H_{t-2}(z)) \\ \hline 2^{-2(t+1)-1} \\ 1 \end{array}\right) = \left(\begin{array}{c} 0 \\ 0 \\ \hline \rho(H_{t-1}(z)) \\ \rho(H_{t-1}(z) - 2^{-2(t+1)+1}) \\ z - \sum_{i=0}^{t-1} H_i(z) \\ \hline 2^{-2(t+1)-1} \\ 1 \end{array}\right),
\end{aligned}
$$

where we used $s_{\ell-1}(\cdot) = 2^{-1}\rho(\cdot) - \rho(\cdot - 2^{-2\ell+1})$ and $H_k = s_k \circ H_{k-1}$ as per Lemma 5.2.2. This completes the proof of (5.4). Furthermore, we see that $\|h[0]\|_\infty \leq 1$ and $\|h[1]\|_\infty \leq 1$. By (5.4), we have $\|h[t]\|_\infty \leq 1$ for all $t \geq 2$ upon using Corollary 5.2.3.

Next, we compute the RNN output $(\mathcal{R}^{\mathrm{Sq}}x)[t]$, for $t \geq 2$, as follows

$$
\begin{aligned}
(\mathcal{R}^{\mathrm{Sq}}x)[t] &= A_o h[t] + b_o \\
&= D^2\left(z - \sum_{i=0}^{t-2} H_i(z) - \left(2^{-1}\rho(H_{t-2}(z)) - \rho(H_{t-2}(z) - 2^{-2t+1})\right)\right) \\
&= D^2\left(z - \sum_{i=0}^{t-2} H_i(z) - H_{t-1}(z)\right) \\
&= D^2\left(z - \sum_{i=0}^{t-1} H_i(z))\right) \\
&= D^2\left(z - I_t(z)\right),
\end{aligned}
$$

where $I_m = \sum_{k=0}^{m-1} H_k$ as in Lemma 5.2.2. As, $z \in [0,1]$, we have, by Corollary 5.2.3, that

$z - I_t(z) \in [0,1]$. Hence, $0 \leq (\mathcal{R}^{\mathrm{Sq}}x)[t] \leq D^2$. Furthermore,

$$\begin{aligned}\left|x^2 - (\mathcal{R}^{\mathrm{Sq}}\mathcal{D}x)[t]\right| &= D^2\left|z^2 - (z - I_t(z))\right| \\ &\leq D^2 \sup_{y\in[0,1]}\left|y^2 - (y - I_t(y))\right| \\ &\leq D^2\, 2^{-2t-2} = \frac{D^2}{4}4^{-t},\end{aligned}$$

where the last inequality follows from Lemma 5.2.2. Since $x \in [-D, D]$ was arbitrary, the statement follows. □

We proceed to the construction of an RNN that approximates the multiplication operation. This result will be based on the polarization identity

$$x_1 \cdot x_2 = \left(\frac{x_1 + x_2}{2}\right)^2 - \left(\frac{x_1 - x_2}{2}\right)^2. \tag{5.5}$$

Specifically, we will map the input (x_1, x_2) to $((x_1 + x_2)/2, (x_1 - x_2)/2)$ through an affine transformation followed by the application of two instances of $\mathcal{R}^{\mathrm{Sq}}$ in parallel. The formal result is as follows.

Lemma 5.2.5. *For $N \in \mathbb{N}$, let $\mathcal{R}^1 = \mathcal{Q}^1\mathcal{K}^1, \ldots, \mathcal{R}^N = \mathcal{Q}^N\mathcal{K}^N$ be RNNs. There exists an RNN $\mathcal{R} = \mathcal{Q}\mathcal{K}$ such that for all $x_1 \in \mathbb{R}^{\mathcal{M}_{\mathrm{in}}(\mathcal{R}^1)}, \ldots, x_N \in \mathbb{R}^{\mathcal{M}_{\mathrm{in}}(\mathcal{R}^N)}$, we have*

$$\left(\mathcal{K}\mathcal{D}\begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}\right)[t] = \begin{pmatrix} (\mathcal{K}^1\mathcal{D}x_1)[t] \\ \vdots \\ (\mathcal{K}^N\mathcal{D}x_N)[t] \end{pmatrix}, \qquad \forall t \in \mathbb{N}_0 \tag{5.6}$$

and

$$\left(\mathcal{R}\mathcal{D}\begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}\right)[t] = \begin{pmatrix} (\mathcal{R}^1\mathcal{D}x_1)[t] \\ \vdots \\ (\mathcal{R}^N\mathcal{D}x_N)[t] \end{pmatrix}, \qquad \forall t \in \mathbb{N}_0. \tag{5.7}$$

Proof. The weights of $\mathcal{R} = \mathcal{Q}\mathcal{K}$ are given by

$$A_h = \begin{pmatrix} A_h^1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & A_h^N \end{pmatrix}, \qquad A_x = \begin{pmatrix} A_x^1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & A_x^N \end{pmatrix}, \quad b_h = \begin{pmatrix} b_h^1 \\ \vdots \\ b_h^N \end{pmatrix}$$

$$A_o = \begin{pmatrix} A_o^1 & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & A_o^N \end{pmatrix}, \qquad b_o = \begin{pmatrix} b_o^1 \\ \vdots \\ b_o^N \end{pmatrix}.$$

Now, arbitrarily fix $x = \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix}$. We first show (5.6) by induction. Indeed, for $t = 0$, we have by Remark 5.2.1,

$$(\mathcal{K}\mathcal{D}x)[0] = \rho(A_x x + b_h) = \rho \begin{pmatrix} A_x^1 x_1 + b_h^1 \\ \vdots \\ A_x^N x_N + b_h^N \end{pmatrix} = \begin{pmatrix} (\mathcal{K}^1\mathcal{D}x_1)[0] \\ \vdots \\ (\mathcal{K}^N\mathcal{D}x_N)[0] \end{pmatrix}.$$

Next, assume that (5.6) holds for some $t \in \mathbb{N}_0$ and compute

$$\begin{aligned}(\mathcal{K}\mathcal{D}x)[t+1] &= \rho\left(A_h\left((\mathcal{K}\mathcal{D}x)[t]\right) + b_h\right) \\ &= \rho\left(A_h \begin{pmatrix} (\mathcal{K}^1\mathcal{D}x_1)[t] \\ \vdots \\ (\mathcal{K}^N\mathcal{D}x_N)[t] \end{pmatrix} + b_h\right) \\ &= \rho \begin{pmatrix} A_h^1\left((\mathcal{K}^1\mathcal{D}x_1)[t]\right) + b_h^1 \\ \vdots \\ A_h^N\left((\mathcal{K}^N\mathcal{D}x_N)[t]\right) + b_h^N \end{pmatrix} = \begin{pmatrix} (\mathcal{K}^1\mathcal{D}x_1)[t+1] \\ \vdots \\ (\mathcal{K}^N\mathcal{D}x_N)[t+1] \end{pmatrix}.\end{aligned}$$

This completes the proof of (5.6). To establish (5.7), we note that

$$\begin{aligned}(\mathcal{R}\mathcal{D}x)[t] &= A_o(\mathcal{K}\mathcal{D}x)[t] + b_o = A_o \begin{pmatrix} (\mathcal{K}^1\mathcal{D}x_1)[t] \\ \vdots \\ (\mathcal{K}^N\mathcal{D}x_N)[t] \end{pmatrix} + b_o \\ &= \begin{pmatrix} A_o^1(\mathcal{K}^1\mathcal{D}x_1)[t] + b_o^1 \\ \vdots \\ A_o^N(\mathcal{K}^N\mathcal{D}x_N)[t] + b_o^N \end{pmatrix} = \begin{pmatrix} (\mathcal{R}^1\mathcal{D}x_1)[t] \\ \vdots \\ (\mathcal{R}^N\mathcal{D}x_N)[t] \end{pmatrix}, \qquad \forall t \in \mathbb{N}_0. \qquad \square\end{aligned}$$

Furthermore, we need a the following result to apply a matrix multiplication to the input of the RNN.

Lemma 5.2.6. *Let $\mathcal{R} = \mathcal{Q}\mathcal{K}$ be an RNN with $d = \mathcal{M}_{\text{in}}(\mathcal{R})$ and let $A \in \mathbb{R}^{d \times d'}$, $d' \in \mathbb{N}$. Then, there exists an RNN $\mathcal{R}' = \mathcal{Q}\mathcal{K}'$ such that, for every $x \in \mathbb{R}^{d'}$,*

$$(\mathcal{R}'\mathcal{D}x)[t] = (\mathcal{R}\mathcal{D}(Ax))[t], \qquad \text{and} \qquad (\mathcal{K}'\mathcal{D}x)[t] = (\mathcal{K}\mathcal{D}(Ax))[t], \qquad \forall t \in \mathbb{N}_0.$$

Proof. Let A_x, A_h, A_o, b_h and b_o be the weights of the RNN $\mathcal{R}$. We define the new RNN $\mathcal{R}' = \mathcal{Q}\mathcal{K}'$ with weights $A_x' = A_x A, A_h' = A_h, A_o' = A_o, b_h' = b_h, b_o' = b_o$, i.e., we take the output

mapping of the new RNN $\mathcal{R}'$ to be identical to that of $\mathcal{R}$. We first show through induction that

$$(\mathcal{K}'\mathcal{D}x)[t] = (\mathcal{K}\mathcal{D}(Ax))\,[t], \qquad \text{for } t \in \mathbb{N}_0. \tag{5.8}$$

First, note that

$$\begin{aligned}(\mathcal{K}'\mathcal{D}x)[0] &= \rho\left(A'_x x + b'_h\right)\\ &= \rho\left(A_x A x + b_h\right)\\ &= \rho\left(A_x(Ax) + b_h\right) = (\mathcal{K}\mathcal{D}(Ax))[0].\end{aligned}$$

Next, assume (5.8) holds for some $t \in \mathbb{N}_0$. We have by Remark 5.2.1,

$$\begin{aligned}(\mathcal{K}'\mathcal{D}x)[t+1] &= \rho\left(A'_h\left((\mathcal{K}'\mathcal{D}x)[t]\right) + b'_h\right)\\ &= \rho\left(A_h\left((\mathcal{K}'\mathcal{D}x)[t]\right) + b_h\right)\\ &= \rho\left(A_h\left((\mathcal{K}\mathcal{D}(Ax))\,[t]\right) + b_h\right)\\ &= (\mathcal{K}\mathcal{D}(Ax))\,[t+1],\end{aligned}$$

proving (5.8). Finally, $(\mathcal{R}'\mathcal{D}x)[t] = (\mathcal{R}\mathcal{D}(Ax))\,[t]$ for $t \in \mathbb{N}_0$ follows from (5.8) since $A'_o = A_o$ and $b'_o = b_o$. □

The last ingredient to our approximation of the multiplication operation is applying an affine transformation to the output of an RNN.

Lemma 5.2.7. *Let $\mathcal{R} = \mathcal{Q}\mathcal{K}$ be an RNN with $d = \mathcal{M}_{\text{out}}(\mathcal{R})$ and let $A \in \mathbb{R}^{d'\times d}, b \in \mathbb{R}^{d'}$, $d' \in \mathbb{N}$. Then, there exists an RNN $\mathcal{R}' = \mathcal{Q}'\mathcal{K}$ such that, for every $x \in \mathbb{R}^{\mathcal{M}_{\text{in}}(\mathcal{R})}$,*

$$(\mathcal{R}'\mathcal{D}x)[t] = A\left((\mathcal{R}\mathcal{D}x)[t]\right) + b, \qquad \forall t \in \mathbb{N}_0.$$

Proof. With $\mathcal{R} = \mathcal{Q}\mathcal{K}$, where $\mathcal{Q}(h) = A_o h + b_o$, define

$$\mathcal{R}' = \mathcal{Q}'\mathcal{K}, \quad \mathcal{Q}'(h) = A'_o h + b'_o, \text{ with } A'_o = AA_o, b'_o = Ab_o + b.$$

That is, we take the hidden state operator of the modified RNN $\mathcal{R}'$ to be identical to that of $\mathcal{R}$. Noting that, for all $t \in \mathbb{N}_0$,

$$\begin{aligned}(\mathcal{R}'\mathcal{D}x)[t] = (\mathcal{Q}'\mathcal{K}\mathcal{D}x)\,[t] &= A'_o\left((\mathcal{K}\mathcal{D}x)[t]\right) + b'_o\\ &= AA_o\left((\mathcal{K}\mathcal{D}x)[t]\right) + Ab_o + b\\ &= A\left(A_o(\mathcal{K}\mathcal{D}x)[t] + b_o\right) + b\\ &= A(\mathcal{R}\mathcal{D}x)[t] + b, \qquad \forall t \in \mathbb{N}_0,\end{aligned}$$

the proof is completed. □

We now have all the ingredients needed to derive our result on the approximation of the multiplication operation.

Theorem 5.2.8. *For $D \geq 1$, there is an RNN $\mathcal{R}^{\times} = \mathcal{Q}^{\times}\mathcal{K}^{\times}$ such that, for all $x \in [-D, D]^2$,*

$$|(\mathcal{R}^{\times}\mathcal{D}x)[t] - (x_1 \cdot x_2)| \leq \frac{D^2}{2}\, 4^{-t}, \qquad \forall t \in \mathbb{N}_0$$

and,

$$\|\mathcal{K}^{\times}\mathcal{D}x)[t]\|_{\infty} \leq 1 \quad \text{and} \quad |(\mathcal{R}^{\times}\mathcal{D}x)[t]| \leq D^2, \qquad \forall t \in \mathbb{N}_0.$$

Furthermore, $\mathcal{M}_{\mathrm{hid}}(\mathcal{R}^{\times}) = 14$.

Proof. We start by employing Lemma 5.2.5 with $N = 2$ and $\mathcal{R}^1 = \mathcal{R}^{\mathrm{Sq}}$, $\mathcal{R}^2 = \mathcal{R}^{\mathrm{Sq}}$ according to Theorem 5.2.4. This yields an RNN $\mathcal{R} = \mathcal{Q}\mathcal{K}$ satisfying, for all $z \in [-D, D]^2$,

$$\left\| (\mathcal{R}\mathcal{D}z)[t] - \begin{pmatrix} z_1^2 \\ z_2^2 \end{pmatrix} \right\|_{\infty} \leq \frac{D^2}{2} 4^{-t}, \quad \|\mathcal{K}\mathcal{D}z\|_{\infty} \leq 1, \quad \text{and } 0 \leq (\mathcal{R}\mathcal{D}z)[t] \leq D^2, \quad \forall t \in \mathbb{N}_0. \tag{5.9}$$

Next, consider the matrix

$$A = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \tag{5.10}$$

and observe, that for all $x \in [-D, D]^2$, $Ax \in [-D, D]^2$. We now apply Lemma 5.2.6 with A in (5.10) and $\mathcal{R}$ as per (5.9) to obtain the RNN $\mathcal{R}' = \mathcal{Q}\mathcal{K}'$ such that, for all $x \in [-D, D]^2$, it holds that

$$\left\| (\mathcal{R}'\mathcal{D}x)[t] - \begin{pmatrix} (\frac{x_1+x_2}{2})^2 \\ (\frac{x_1-x_2}{2})^2 \end{pmatrix} \right\|_{\infty} \leq \frac{D^2}{4} 4^{-t}, \tag{5.11}$$

$$\|\mathcal{K}'\mathcal{D}x\|_{\infty} \leq 1, \quad \text{and} \tag{5.12}$$

$$0 \leq (\mathcal{R}'\mathcal{D}x)[t] \leq D^2, \quad \forall t \in \mathbb{N}_0. \tag{5.13}$$

Finally, we apply Lemma 5.2.7 with $\mathcal{R}'$, $A = \begin{pmatrix} 1 & -1 \end{pmatrix}$ and $b = 0$ to obtain the RNN $\mathcal{R}^{\times} = \mathcal{Q}^{\times}\mathcal{K}^{\times}$, where $\mathcal{K}^{\times} = \mathcal{K}'$, since the hidden state operator remains unmodified when applying Lemma 5.2.7. Hence, we have $\|\mathcal{K}^{\times}\mathcal{D}x\|_{\infty} \leq 1$ by (5.12). Furthermore, the output satisfies

$$(\mathcal{R}^{\times}\mathcal{D}x)[t] = ((\mathcal{R}'\mathcal{D}x)[t])_1 - ((\mathcal{R}'\mathcal{D}x)[t])_2, \qquad \forall t \in \mathbb{N}_0. \tag{5.14}$$

It thus follows from (5.13) that $-D^2 \leq (\mathcal{R}^{\times}\mathcal{D}x)[t] \leq D^2$ for all $t \in \mathbb{N}_0$. Finally, we have

$$\begin{aligned}
|(\mathcal{R}^{\times}\mathcal{D}x)[t] - (x_1 \cdot x_2)| &\overset{(5.5)}{=} \left|(\mathcal{R}^{\times}\mathcal{D}x)[t] - \left(\left(\frac{x_1+x_2}{2}\right)^2 - \left(\frac{x_1-x_2}{2}\right)^2\right)\right| \\
&\overset{(5.14)}{=} \left|((\mathcal{R}'\mathcal{D}x)[t])_1 - \left(\frac{x_1+x_2}{2}\right)^2 + \left(\frac{x_1-x_2}{2}\right)^2 - ((\mathcal{R}'\mathcal{D}x)[t])_2\right| \\
&\leq \left|((\mathcal{R}'\mathcal{D}x)[t])_1 - \left(\frac{x_1+x_2}{2}\right)^2\right| + \left|((\mathcal{R}'\mathcal{D}x)[t])_2 - \left(\frac{x_1-x_2}{2}\right)^2\right| \\
&\overset{(5.11)}{\leq} \frac{D^2}{4}4^{-t} + \frac{D^2}{4}4^{-t} = \frac{D^2}{2}4^{-t}, \qquad \forall x \in [-D, D], \quad \forall t \in \mathbb{N}_0.
\end{aligned}$$

Lastly, note that we have $\mathcal{M}_{\text{hid}}(\mathcal{R}^{\times}) = 14$, since $\mathcal{R}^{\times}$ contains two parallel instances of $\mathcal{R}^{\text{Sq}}$, which has size $\mathcal{M}_{\text{hid}}(\mathcal{R}^{\times}) = 7$ according to Lemma 5.2.4. □

5.3. Function concatenation through RNNs

Our ultimate goal is the approximation of general polynomials in x, which requires RNNs that approximate higher powers. This will be done by concatenating functions, e.g., $x \to x^4$ can be expressed by concatenating the squaring function with itself. More generally, consider the functions f, g with associated RNNs $\mathcal{R}^f$ and $\mathcal{R}^g$ that approximate f and g, respectively, with error decreasing over time, in the sense of

$$\begin{aligned}
&|(\mathcal{R}^f\mathcal{D}x)[t] - f(x)| \leq c_1 4^{-c_2 t}, \qquad \text{and} \\
&|(\mathcal{R}^g\mathcal{D}x)[t] - g(x)| \leq c_1 4^{-c_2 t}, \qquad \forall t \in \mathbb{N}_0, \qquad \text{for some } c_1, c_2 > 0.
\end{aligned}$$

We wish to construct a new RNN $\mathcal{R}'$ that approximates $g \circ f$ with approximation error decreasing over time, i.e.,

$$|(\mathcal{R}'\mathcal{D}x)[t] - g(f(x))| \leq c_3 4^{-c_4 t}, \qquad \forall t \in \mathbb{N}_0, \qquad \text{for some } c_3, c_4 > 0.$$

The core idea, as shown in Figure 5.1, is to let $\mathcal{R}^f$ and $\mathcal{R}^g$ run in parallel. Specifically, we shall interlace the computation of f and of g. Whenever, the time-index is a power of 2, we use the current value of the hidden state h_f to read out an approximation of $f(x)$. Then, we restart the computation of $\mathcal{R}^g$ with this refined value as input. Thus, the overall output at time-index $t = 2^k$ for $k \in \mathbb{N}$ is an approximation of $g(f(x))$, where 2^{k-1} time steps were used to run $\mathcal{R}^f$ in order to approximate $\widetilde{y} = f(x)$ and 2^{k-1} time steps were used to run $\mathcal{R}^g$ with input $\widetilde{y}$ in order to approximate $g(\widetilde{y})$. We next show how to build a clock into an RNN that triggers this switching action whenever the time-index is a power of 2.

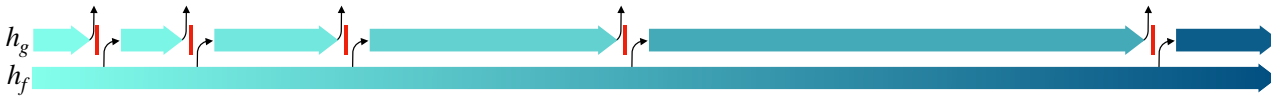


Figure 5.1: Approximating function concatenation by RNNs.

Lemma 5.3.1. *Define the sequence*

$$\widehat{\delta}[t] = \begin{cases} 1, & \text{if } t = 2^k \text{ for some } k \in \mathbb{N},\ k \geq 2 \\ 0, & \text{else.} \end{cases}$$

Let

$$\widehat{A} := \begin{pmatrix} -4 & 2 & 0 & 0 & 0 \\ -4 & 2 & 0 & 2 & -1/2 \\ 0 & 0 & 1/2 & 0 & -1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad \widehat{b} := \begin{pmatrix} -1 \\ 1/2 \\ 1 \\ -2 \\ 1 \end{pmatrix}, \tag{5.15}$$

and consider the recursively defined sequence $h[-1] = 0$ and $h[t+1] = \rho(\widehat{A}h[t] + \widehat{b})$, for $t \in \mathbb{N}_0$. It holds that

$$(h[t])_1 = \widehat{\delta}[t+2]$$

Furthermore, $\|h[t]\|_\infty \leq 2$, for all $t \in \mathbb{N}_0$.

Proof. The proof is effected through induction over time. We start by showing that

$$h[2^k - 2] = \begin{pmatrix} 1 \\ 2 \\ 2^{2-2^k} \\ 0 \\ 1 \end{pmatrix}, \qquad \text{for } k \in \mathbb{N},\ k \geq 2. \tag{5.16}$$

The base case, $k = 2$, is established through direct computation:

$$h[0] = \rho \begin{pmatrix} -1 \\ 1/2 \\ 1 \\ -2 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1/2 \\ 1 \\ 0 \\ 1 \end{pmatrix}, \qquad h[1] = \rho \begin{pmatrix} 0 \\ 1 \\ 1/2 \\ -1/2 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1/2 \\ 0 \\ 1 \end{pmatrix},$$

and

$$h[2] = \rho \begin{pmatrix} 1 \\ 2 \\ 2^{-2} \\ -1/2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 2^{-2} \\ 0 \\ 1 \end{pmatrix}.$$

Next, we assume that (5.16) holds for some $k \geq 2$. We first compute

$$h[2^k - 1] = \rho \begin{pmatrix} -1 \\ 0 \\ 2^{1-2^k} \\ 2^{2-2^k} \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2^{1-2^k} \\ 2^{2-2^k} \\ 1 \end{pmatrix},$$

and

$$h[2^k] = \rho \begin{pmatrix} -1 \\ 2^{3-2^k} \\ 2^{-2^k} \\ 2^{1-2^k} - 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2^{3-2^k} \\ 2^{-2^k} \\ 0 \\ 1 \end{pmatrix}. \tag{5.17}$$

Next, we show with a nested induction that

$$h[2^k + t] = \begin{pmatrix} 0 \\ 2^{t+3-2^k} \\ 2^{-2^k-t} \\ 0 \\ 1 \end{pmatrix}, \qquad \text{for } t \in \{0, 1, \ldots, 2^k - 3\}. \tag{5.18}$$

The base case, $t = 0$, was already established in (5.17). Assume now that (5.18) holds for some $t \in \{0, 1, \ldots, 2^k - 4\}$ and compute

$$h[2^k + t + 1] = \rho \begin{pmatrix} 2^{t+4-2^k} - 1 \\ 2^{t+4-2^k} \\ 2^{-2^k-t-1} \\ 2^{t+3-2^k} + 2^{-2^k-t} - 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2^{t+4-2^k} \\ 2^{-2^k-(t+1)} \\ 0 \\ 1 \end{pmatrix},$$

where we used $2^{t+4-2^k} \leq 1$ since $t \leq 2^k - 4$. This completes the induction over t and thus

establishes (5.18). Particularizing (5.18) to $t = 2^k - 3$, we have

$$h[2^{k+1} - 3] = h[2^k + 2^k - 3] = \begin{pmatrix} 0 \\ 2^{2^k - 3 + 3 - 2^k} \\ 2^{-2^k - (2^k - 3)} \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 2^{3 - 2^{k+1}} \\ 0 \\ 1 \end{pmatrix}.$$

Next, we compute

$$h[2^{k+1} - 2] = \rho \begin{pmatrix} 1 \\ 2 \\ 2^{2 - 2^{k+1}} \\ 1 + 2^{3 - 2^{k+1}} - 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 2^{2 - 2^{k+1}} \\ 0 \\ 1 \end{pmatrix},$$

where we used $2^{3-2^{k+1}} \leq 1$, for $k \geq 2$. This completes the induction over k and thus establishes (5.16) and, in particular, that

$$(h[t])_1 = 1, \quad \text{if } t = 2^k - 2 \text{ for some } k \in \mathbb{N},\ k \geq 2.$$

Inspection of (5.17) and (5.18) reveals that

$$(h[t])_1 = 0, \quad \text{for every other } t.$$

Lastly, $\|h[\cdot]\|_\infty \leq 2$ follows from (5.16) through (5.18). □

We next define a map, which, based on Lemma 5.3.1, will then be shown to produce an RNN that realizes the desired behavior.

Definition 5.3.2. *Let $D_1, D_h > 0$, and $\mathcal{R}^f$ and $\mathcal{R}^g$ RNNs such that $\mathcal{M}_{\text{in}}(\mathcal{R}^g) = \mathcal{M}_{\text{out}}(\mathcal{R}^f) =: d'_f$. We identify the weights of $\mathcal{R}^f$ as $A^f_x, A^f_h, b^f_h, A^f_o, b^f_o$ and the weights of $\mathcal{R}^g$ as $A^g_x, A^g_h, b^g_h, A^g_o, b^g_o$, further let $m_g := \mathcal{M}_{\text{hid}}(\mathcal{R}^g)$, $m_f := \mathcal{M}_{\text{hid}}(\mathcal{R}^f)$, and $d_f := \mathcal{M}_{\text{in}}(\mathcal{R}^f)$. We now define a mapping*

$\mathcal{R} = \mathring{\Psi}_{D_1,D_h}(\mathcal{R}^g, \mathcal{R}^f)$, *where the weights of* $\mathcal{R}$ *are given by*

$$A_h = \left(\begin{array}{c|ccc|c} A_h^f & 0 & 0 & 0 & 0 \\ \hline A_o^f & 0 & 0 & 0 & D_1 \mathbf{1}_{d'_f} \widehat{A}_o \\ -A_o^f & 0 & 0 & 0 & D_1 \mathbf{1}_{d'_f} \widehat{A}_o \\ \hline 0 & A_x^g & -A_x^g & A_h^g & -D_h \mathbf{1}_{m_g} \widehat{A}_o \\ \hline 0 & 0 & 0 & 0 & \widehat{A} \end{array}\right), \quad b_h = \left(\begin{array}{c} b_h^f \\ \hline b_o^f - D_1 \\ -b_o^f - D_1 \\ \hline b_h^g \\ \hline \widehat{b} \end{array}\right),$$
$$A_x = \left(\begin{array}{c} A_x^f \\ \hline 0 \\ 0 \\ \hline 0 \\ \hline 0 \end{array}\right), \quad A_o = \Big(0 \;\Big|\; 0 \;\; 0 \;\Big|\; A_o^g \;\Big|\; 0\Big), \quad \text{and} \;\; b_o = b_o^g, \tag{5.19}$$

where $\widehat{A}_o \coloneqq \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}$, *and* $\widehat{A}$ *and* $\widehat{b}$ *are as in* (5.15). *It holds that*

$$\mathcal{M}_{\text{hid}}(\mathcal{R}) = \mathcal{M}_{\text{hid}}(\mathcal{R}^f) + 2\mathcal{M}_{\text{out}}(\mathcal{R}^f) + \mathcal{M}_{\text{hid}}(\mathcal{R}^g) + 5.$$

Furthermore, define the mappings

$$\mathring{\mathcal{M}}(\mathcal{R}^g, \mathcal{R}^f) \coloneqq \Big(\mathbb{I}_{m_f} \;\Big|\; 0 \;\; 0 \;\Big|\; 0 \;\Big|\; 0\Big), \quad \text{and} \quad \mathring{\mathcal{W}}(\mathcal{R}^g, \mathcal{R}^f) \coloneqq \Big(0 \;\Big|\; 0 \;\; 0 \;\Big|\; \mathbb{I}_{m_g} \;\Big|\; 0\Big).$$

We next establish the properties of the RNN produced by the maping $\mathring{\Psi}_{D_1,D_h}(\mathcal{R}^g, \mathcal{R}^f)$.

Theorem 5.3.3. *Let* $D, D_1, D_2, D_h > 0$, *and* $\mathcal{R}^f = \mathcal{Q}^f\mathcal{K}^f$ *and* $\mathcal{R}^g = \mathcal{Q}^g\mathcal{K}^g$ *be RNNs with weights as in Definition 5.3.2. Assume that the following bounds hold*

(A1) $\Big\|(\mathcal{R}^f\mathcal{D}x)[t]\Big\|_\infty \leq D_1$, *for all* $x \in [-D, D]^{d_f}$, *for all* $t \in \mathbb{N}_0$.

(A2) $\|(\mathcal{K}^g\mathcal{D}y)[t]\|_\infty \leq D_h$, *for all* $y \in [-D_1, D_1]^{d'_f}$, *for all* $t \in \mathbb{N}_0$.

Then, the RNN $\mathring{\Psi}_{D_1,D_h}(\mathcal{R}^g, \mathcal{R}^f) =: \mathcal{R} = \mathcal{Q}\mathcal{K}$ *given according to Definition 5.3.2 satisfies, that for all* $x \in [-D, D]^{d_f}$,

$$\begin{aligned} (\mathcal{R}\mathcal{D}x)[2^k - 2] = &\Big(\mathcal{R}^g\mathcal{D}\Big(\\ &\quad (\mathcal{R}^f\mathcal{D}x)[2^{k-1} - 2] \\ &\Big)\Big)[2^{k-1} - 2], \qquad \text{for } k \geq 3. \end{aligned} \tag{5.20}$$

Furthermore, with $M = \mathring{\mathcal{M}}(\mathcal{R}^g, \mathcal{R}^f)$ *and* $W = \mathring{\mathcal{W}}(\mathcal{R}^g, \mathcal{R}^f)$, *we have*

$$M(\mathcal{K}\mathcal{D}x)[t] = (\mathcal{K}^f\mathcal{D}x)[t], \qquad \text{for } t \in \mathbb{N}_0, \tag{5.21}$$

and

$$W(\mathcal{K}\mathcal{D}x)[2^k-2] = \left(\mathcal{K}^g\mathcal{D}\left((\mathcal{R}^f\mathcal{D}x)[2^{k-1}-2]\right)\right)[2^{k-1}-2], \quad \text{for } k\in\mathbb{N},\ k\geq 3. \tag{5.22}$$

If we additionally assume that

(A3) $\left\|(\mathcal{K}^f\mathcal{D}x)[t]\right\|_\infty \leq D_h$, *for all* $x\in[-D,D]^{d_f}$, *for all* $t\in\mathbb{N}_0$,

(A4) $\|(\mathcal{R}^g\mathcal{D}y)[t]\|_\infty \leq D_2$, *for all* $y\in[-D_1,D_1]^{d'_f}$, *for all* $t\in\mathbb{N}_0$,

we have, for all $x\in[-D,D]^{d_f}$,

(B1) $\|(\mathcal{K}\mathcal{D}x)[t]\|_\infty \leq \max\{2,D_1,D_h\}$, *for all* $t\in\mathbb{N}_0$,

(B2) $\|(\mathcal{R}\mathcal{D}x)[t]\|_\infty \leq D_2$, *for all* $t\in\mathbb{N}_0$.

Proof. Arbitrarily, fix input $x\in[-D,D]^{d_f}$ and consider the hidden state sequence in response to the input sequence $(\mathcal{D}x)[\cdot]$, i.e., $h[t]=(\mathcal{K}\mathcal{D}x)[t]$. We divide this hidden state sequence corresponding to the blocks in (5.19) according to

$$h[t] = \left(\begin{array}{c} h_1[t] \\ \hline h_2[t] \\ \hline h_3[t] \\ \hline h_4[t] \end{array}\right) = \rho\left(A_h h[t-1] + A_x((Dx)[t]) + b_h\right),$$

and analyze each block individually. First, note that $h_4[\cdot]$ follows the recursion

$$h_4[t+1] = \rho(\widehat{A}h_4[t]+\widehat{b}),\ \text{with } h_4[-1]=0. \tag{5.23}$$

Thus, by Lemma 5.3.1, we have that

$$\widehat{A}_o h_4[t] = \widehat{\delta}[t+2]. \tag{5.24}$$

Next, we note that, $h_1[t]$ follows the recursion

$$h_1[t] = \rho\left(A_h^f h_1[t-1] + A_x^f(\mathcal{D}x)[t] + b_h^f\right) \quad \text{with } h_1[-1]=0.$$

Thus, recalling Definition 5.1.1, we have

$$h_1[t] = (\mathcal{K}^f\mathcal{D}x)[t]. \tag{5.25}$$

Next, we consider the sequence $h_2[t]$, which is given by

$$\begin{aligned}
h_2[t] &= \rho\begin{pmatrix} A_o^f h_1[t-1] + D_1 \mathbf{1}_{d'_f} \widehat{A}_o h_4[t-1] + b_o^f - D_1 \\ -A_o^f h_1[t-1] + D_1 \mathbf{1}_{d'_f} \widehat{A}_o h_4[t-1] - b_o^f - D_1 \end{pmatrix} \\
&= \rho\begin{pmatrix} (A_o^f(\mathcal{K}^f\mathcal{D}x)[t-1] + b_o^f) - D_1 \mathbf{1}_{d'_f}(1 - \widehat{A}_o h_4[t-1]) \\ -(A_o^f(\mathcal{K}^f\mathcal{D}x)[t-1] + b_o^f) - D_1 \mathbf{1}_{d'_f}(1 - \widehat{A}_o h_4[t-1]) \end{pmatrix} \\
&\overset{(5.24)}{=} \rho\begin{pmatrix} (\mathcal{Q}^f\mathcal{K}^f\mathcal{D}x)[t-1] - D_1 \mathbf{1}_{d'_f}(1 - \widehat{\delta}[t+1]) \\ -(\mathcal{Q}^f\mathcal{K}^f\mathcal{D}x)[t-1] - D_1 \mathbf{1}_{d'_f}(1 - \widehat{\delta}[t+1]) \end{pmatrix} \\
&\overset{(i)}{=} \rho\begin{pmatrix} (\mathcal{R}^f\mathcal{D}x)[t-1] \\ -(\mathcal{R}^f\mathcal{D}x)[t-1] \end{pmatrix} \widehat{\delta}[t+1], \qquad (5.26)
\end{aligned}$$

where in (i) we used $\left\|(\mathcal{Q}^f\mathcal{K}^f\mathcal{D}x)[t-1]\right\|_\infty = \left\|(\mathcal{R}^f\mathcal{D}x)[t-1]\right\|_\infty \leq D_1$ by assumption (A1). Using expression (5.26), we note that $h_3[t]$ is given by the following recursion with $h_3[-1] = 0$

$$\begin{aligned}
h_3[t+1] = \rho\Big(& A_x^g \rho((\mathcal{R}^f\mathcal{D}x)[t-1])\widehat{\delta}[t+1] - A_x^g \rho(-(\mathcal{R}^f\mathcal{D}x)[t-1])\widehat{\delta}[t+1] \\
& + A_h^g h_3[t] + b_h^g - D_h \mathbf{1}_{m_g} \widehat{A}_o h_4[t] \Big) \\
\overset{(ii)}{=} \rho\Big(& A_h^g h_3[t] + b_h^g - D_h \mathbf{1}_{m_g} \widehat{A}_o h_4[t] + \widehat{\delta}[t+1] \cdot A_x^g (\mathcal{R}^f\mathcal{D}x)[t-1] \Big) \\
\overset{(5.24)}{=} \rho\Big(& A_h^g h_3[t] + b_h^g - D_h \mathbf{1}_{m_g} \widehat{\delta}[t+2] + \widehat{\delta}[t+1] \cdot A_x^g (\mathcal{R}^f\mathcal{D}x)[t-1] \Big),
\end{aligned}$$

where in (ii) we used the identity $x = \rho(x) - \rho(-x)$. As $\widehat{\delta}[\ell] = 0$, for $\ell \in \{0, \ldots, 3\}$, it follows that

$$h_3[t] = (\mathcal{K}^g\mathcal{D}0)[t], \quad \text{for } t \in \{0, 1, 2\}.$$

Next, we compute

$$\begin{aligned}
h_3[3] &= \rho\Big(A_h^g h_3[2] + b_h^g - D_h \mathbf{1}_{m_g} \widehat{\delta}[4] + \widehat{\delta}[3] \cdot A_x^g (\mathcal{R}^f\mathcal{D}x)[1] \Big) \\
&= \rho\Big(A_h^g h_3[2] + b_h^g - D_h \mathbf{1}_{m_g} \Big) = 0, \qquad (5.27)
\end{aligned}$$

where we used that $A_h^g h_3[2] + b_h^g \leq D_h$, since $\|(\mathcal{K}^g\mathcal{D}0)[3]\|_\infty = \|\rho(A_h^g h_3[2] + b_h^g)\|_\infty \leq D_h$ by Assumption (A2). We now show by nested induction that

$$\begin{aligned}
& h_3[2^k - 1] = 0, && \text{for } k \in \mathbb{N},\ k \geq 2, && (5.28) \\
\text{and} \quad & h_3[2^k + \ell] = \Big(\mathcal{K}^g\mathcal{D}\Big((\mathcal{R}^f\mathcal{D}x)[2^k - 2]\Big)\Big)[\ell], && \text{for } \ell \in \{0, \ldots, 2^k - 2\}. && (5.29)
\end{aligned}$$

The base case for the induction over k was already established in (5.27), i.e., that (5.28) holds

for $k = 2$. Now we assume that (5.28) holds for some $k \geq 2$ and compute

$$\begin{aligned} h_3[2^k] &= \rho\left(A_h^g h_3[2^k - 1] + b_h^g - D_h \mathbf{1}_{m_g} \widehat{\delta}[2^k + 1] + \widehat{\delta}[2^k] \cdot A_x^g (\mathcal{R}^f \mathcal{D} x)[2^k - 2]\right) \\ &= \rho\left(A_h^g 0 + b_h^g + A_x^g (\mathcal{R}^f \mathcal{D} x)[2^k - 2]\right) \\ &= \left(\mathcal{K}^g \mathcal{D}\left((\mathcal{R}^f \mathcal{D} x)[2^k - 2]\right)\right)[0], \end{aligned}$$

where we used Remark 5.2.1. This establishes the base case for the nested induction over ℓ, i.e., that (5.29) holds for $\ell = 0$. Next, assume that (5.29) holds for some $\ell \in \{0, \ldots, 2^k - 3\}$ and compute

$$\begin{aligned} h_3[2^k + \ell + 1] &= \rho\Big(A_h^g h_3[2^k + \ell] + b_h^g - D_h \mathbf{1}_{m_g} \widehat{\delta}[2^k + \ell + 2] \\ &\qquad + \widehat{\delta}[2^k + \ell + 1] \cdot A_x^g (\mathcal{R}^f \mathcal{D} x)[2^k + \ell - 1]\Big) \\ &= \rho\left(A_h^g h_3[2^k + \ell] + b_h^g\right) \\ &\overset{(5.29)}{=} \rho\left(A_h^g \left(\mathcal{K}^g \mathcal{D}\left((\mathcal{R}^f \mathcal{D} x)[2^k - 2]\right)\right)[\ell] + b_h^g\right) \\ &\overset{\text{Rem. 5.2.1}}{=} \left(\mathcal{K}^g \mathcal{D}\left((\mathcal{R}^f \mathcal{D} x)[2^k - 2]\right)\right)[\ell + 1]. \end{aligned}$$

This completes the induction over ℓ and thus establishes (5.29). Next, we compute

$$\begin{aligned} h_3[2^{k+1} - 1] = h_3[2^k + 2^k - 2 + 1] &= \rho\Big(A_h^g h_3[2^k + 2^k - 2] + b_h^g - D_h \mathbf{1}_{m_g} \widehat{\delta}[2^{k+1}] \\ &\qquad + \widehat{\delta}[2^{k+1} - 1] \cdot A_x^g (\mathcal{R}^f \mathcal{D} x)[2^{k+1} - 3]\Big) \\ &= \rho\left(A_h^g \left(\mathcal{K}^g \mathcal{D}\left((\mathcal{R}^f \mathcal{D} x)[2^k - 2]\right)\right)[2^k - 2] + b_h^g - D_h \mathbf{1}_{m_g}\right) \\ &\overset{(iii)}{=} 0, \end{aligned}$$

where (iii) is because

$$\begin{aligned} A_h^g \left(\mathcal{K}^g \mathcal{D}\left((\mathcal{R}^f \mathcal{D} x)[2^k - 2]\right)\right)[2^k - 2] + b_h^g &\leq \|\rho(A_h^g \left(\mathcal{K}^g \mathcal{D}\left((\mathcal{R}^f \mathcal{D} x)[2^k - 2]\right)\right)[2^k - 2] + b_h^g)\|_\infty \\ &\overset{(A1)}{\leq} \max_{\|x'\|_\infty \leq D_1} \|\rho(A_h^g \left(\mathcal{K}^g \mathcal{D} x'\right)[2^k - 2] + b_h^g)\|_\infty \\ &= \max_{\|x'\|_\infty \leq D_1} \|\left(\mathcal{K}^g \mathcal{D} x'\right)[2^k - 1]\|_\infty \overset{(A2)}{\leq} D_h. \end{aligned}$$

This establishes that (5.28) holds for $k + 1$ as well and thus completes the nested induction.

The network output is given by $(\mathcal{RD}x)[t] = A_o^g h_3[t] + b_o^g$, which, using (5.29), yields

$$(\mathcal{RD}x)[2^k + \ell] = \left(\mathcal{R}^g \mathcal{D}\left((\mathcal{R}^f \mathcal{D}x)[2^k - 2]\right)\right)[\ell], \quad \text{for } k \geq 2, \text{ and } \ell \in \{0, \ldots, 2^k - 2\}. \tag{5.30}$$

In particular, setting $\ell = 2^k - 2$, we have

$$(\mathcal{RD}x)[2^{k+1} - 2] = \left(\mathcal{R}^g \mathcal{D}\left((\mathcal{R}^f \mathcal{D}x)[2^k - 2]\right)\right)[2^k - 2],$$

which establishes (5.20). Further, (5.21) follows directly from Definition 5.3.2 and (5.25). Next, (5.22) follows from Definition 5.3.2 and (5.29) upon inserting $\ell = 2^k - 2$.

To establish (B1), we need to establish bounds for $h_1[\cdot]$ through $h_4[\cdot]$. We have, $\|h_1[t]\|_\infty \leq D_h$ by (5.25) and assumption (A3). Further, $\|h_2[t]\|_\infty \leq D_1$ by (5.26) and Assumption (A1). Next, $\|h_3[t]\|_\infty \leq D_h$ follows from (5.28) and (5.29) together with Assumptions (A1) and (A2). Next, $\|h_4[t]\|_\infty \leq 2$ by (5.23) and Lemma 5.3.1. Taking the maximum over these bounds we arrive at ((B1)).

Lastly, (B2) follows from (5.30) together with Assumption (A4).

□

Now, consider the example of approximating the function $x \to x^{16}$, which we can write as $\mathrm{Sq} \circ \mathrm{Sq} \circ \mathrm{Sq} \circ \mathrm{Sq}$, where $\mathrm{Sq} : x \to x^2$. Hence, we approximate $x \to x^{16}$ by concatenating four copies of $\mathcal{R}^{\mathrm{Sq}}$, the RNN from Theorem 5.2.4. Specifically, we use the mapping from Definition 5.3.2 to obtain an RNN according to $\mathring{\Psi}(\mathring{\Psi}(\mathcal{R}^{\mathrm{Sq}}, \mathcal{R}^{\mathrm{Sq}}), \mathring{\Psi}(\mathcal{R}^{\mathrm{Sq}}, \mathcal{R}^{\mathrm{Sq}}))$, as visualized in Figure 5.2. Its output sequence, visualized on the left of Figure 5.2, is an approximation to x^{16}. Furthermore, through clever use of (5.21) and (5.22), we can access the computation of the component networks and thus obtain approximations to x^8, x^4, and x^2 as well. We proceed to formalize these ideas for the general case of concatenating a list of RNNs.

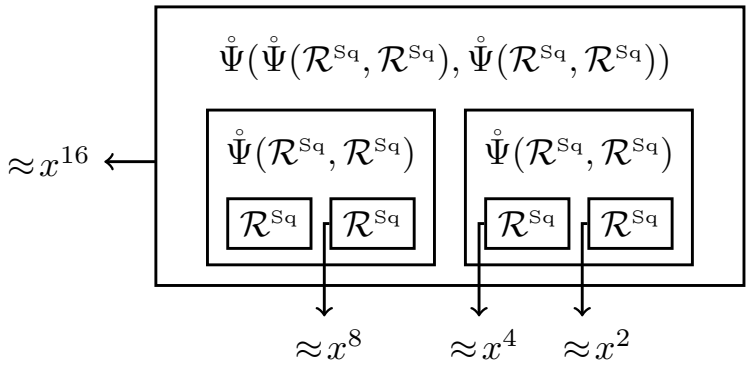


Figure 5.2: RNN approximating $x \to x^{16}$.

Lemma 5.3.4. *Fix $D > 0$, $L \in \mathbb{N}$, and let $\mathcal{R}^1, \dots, \mathcal{R}^{2^L}$ be RNNs with $\mathcal{M}_{\text{in}}(\mathcal{R}^{\ell+1}) = \mathcal{M}_{\text{out}}(\mathcal{R}^\ell)$. Furthermore, assume that there are constants $D_1, \dots, D_{2^L}$ and $D^h \geq \max\{2, \max_{\ell \in \{1,\dots,2^L\}} D_\ell\}$ such that, for all $\ell \in \{1, \dots, 2^L\}$,*

$$\sup_{\|x\|_\infty \leq D_{\ell-1}, t \in \mathbb{N}_0} \left\| (\mathcal{R}^\ell \mathcal{D} x)[t] \right\|_\infty \leq D_\ell, \tag{5.31}$$

$$\sup_{\|x\|_\infty \leq D_{\ell-1}, t \in \mathbb{N}_0} \left\| (\mathcal{K}^\ell \mathcal{D} x)[t] \right\|_\infty \leq D^h, , \tag{5.32}$$

where we set $D_0 = D$. Define the maps

$$g_\ell^k : x \to \left(\mathcal{R}^\ell \mathcal{D} x\right)[2^k - 2], \qquad \text{for } \ell \in \{1, \dots, 2^L\}, k \geq 2. \tag{5.33}$$

Then there is an RNN $\widehat{\mathcal{R}}^L = \widehat{\mathcal{Q}}^L \widehat{\mathcal{K}}^L$ such that

$$\sup_{\|x\|_\infty \leq D, t \in \mathbb{N}_0} \left\| \left(\widehat{\mathcal{R}}^L \mathcal{D} x\right)[t] \right\|_\infty \leq D_{2^L} \quad \text{and} \quad \sup_{\|x\|_\infty \leq D, t \in \mathbb{N}_0} \left\| \left(\widehat{\mathcal{K}}^L \mathcal{D} x\right)[t] \right\|_\infty \leq D^h, \tag{5.34}$$

and

$$\left(\widehat{\mathcal{R}}^L \mathcal{D} x\right)[2^k - 2] = \left(g_{2^L}^{k-L} \circ \cdots \circ g_1^{k-L}\right)(x) \qquad \forall k \geq L+2, \ \forall x \in [-D, D]^{\mathcal{M}_{\text{in}}(\mathcal{R}^1)}. \tag{5.35}$$

Furthermore, for every $\ell \in \{1, \dots, 2^L\}$, there are A^ℓ, b^ℓ, and $\widetilde{k}_1, \dots, \widetilde{k}_\ell \in \{0, \dots, \lceil \log \ell \rceil\}$ such that

$$A^\ell \left(\widehat{\mathcal{K}}^L \mathcal{D} x\right)[2^k - 2] + b^\ell = \left(g_\ell^{k-\widetilde{k}_\ell} \circ \cdots \circ g_1^{k-\widetilde{k}_1}\right)(x), \qquad \forall k \geq L+2, \ \forall x \in [-D, D]^{\mathcal{M}_{\text{in}}(\mathcal{R}^1)}. \tag{5.36}$$

Finally, we have

$$\mathcal{M}_{\text{hid}}(\widehat{\mathcal{R}}^L) = \sum_{\ell=1}^{2^L} \mathcal{M}_{\text{hid}}(\mathcal{R}^\ell) + 2 \sum_{\ell=1}^{2^L - 1} \mathcal{M}_{\text{out}}(\mathcal{R}^\ell) + 5(2^L - 1). \tag{5.37}$$

Proof. We prove the statement by induction over $L \in \mathbb{N}$. For $L = 1$, given the RNNs $\mathcal{R}^1 = \mathcal{Q}^1 \mathcal{K}^1$ and $\mathcal{R}^2 = \mathcal{Q}^2 \mathcal{K}^2$, we take $\widehat{\mathcal{R}}^1 = \widehat{\mathcal{Q}}^1 \widehat{\mathcal{K}}^1 = \mathring{\Psi}_{D^h, D^h}(\mathcal{R}^2, \mathcal{R}^1)$ according to Definition 5.3.2 and $M^1 = \mathring{\mathcal{M}}(\mathcal{R}^2, \mathcal{R}^1)$ as well as $W^1 = \mathring{\mathcal{W}}(\mathcal{R}^2, \mathcal{R}^1)$. By assumption, we have

$$\sup_{\|x\|_\infty \leq D, t \in \mathbb{N}_0} \left\| (\mathcal{R}^1 \mathcal{D} x)[t] \right\|_\infty \leq D_1 \leq D^h \ \text{ and } \sup_{\|x'\|_\infty \leq D_1, t \in \mathbb{N}_0} \left\| (\mathcal{K}^2 \mathcal{D} x')[t] \right\|_\infty \leq D^h, \tag{5.38}$$

as well as

$$\sup_{\|x\|_\infty \leq D, t \in \mathbb{N}_0} \left\| (\mathcal{K}^1 \mathcal{D} x)[t] \right\|_\infty \leq D^h \ \text{ and } \sup_{\|x'\|_\infty \leq D_1, t \in \mathbb{N}_0} \left\| (\mathcal{R}^2 \mathcal{D} x')[t] \right\|_\infty \leq D_2. \tag{5.39}$$

By (5.38) the conditions (A1) and (A2) of Theorem 5.3.3 are satisfied. Furthermore, conditions (A3) and (A4) of Theorem 5.3.3 are satisfied by (5.39). We thus have (5.34) for $L = 1$ as a consequence of (B1) and (B2) in Theorem 5.3.3. Furthermore, we have by (5.20) in Theorem 5.3.3, that

$$\begin{aligned}(\widehat{\mathcal{R}^1}\mathcal{D}x)[2^k-2] = \big(\mathcal{R}^2\mathcal{D}\big(&\\ &(\mathcal{R}^1\mathcal{D}x)[2^{k-1}-2]\\ \big)\big)[2^{k-1}-2], &\qquad \text{for } k\geq 3,\ \forall x\in[-D,D]^{\mathcal{M}_{\text{in}}(\mathcal{R}^1)},\end{aligned}$$

which, upon invoking (5.33), yields (5.35). Next, we establish (5.36). To this end, let

$$A^1 \coloneqq A_o^1 M^1,\ b^1 = b_o^1,\ \text{and}\ A^2 \coloneqq A_o^2 W^1,\ b^2 = b_o^2,$$

where A_o^1, b_o^1, and A_o^2, A_o^2 are the weights of the affine output maps for the RNNs $\mathcal{R}^1$ and $\mathcal{R}^2$, respectively. We have, for all $\|x\|_\infty \leq D,\ k \geq 3$,

$$\begin{aligned}A^1\left(\widehat{\mathcal{K}}^1\mathcal{D}x\right)[2^k-2]+b^1 &= A_o^1M^1\left(\widehat{\mathcal{K}}^1\mathcal{D}x\right)[2^k-2]+b_o^1\\ &\overset{(5.21)}{=} A_o^1\left(\mathcal{K}^1\mathcal{D}x\right)[2^k-2]+b_o^1\\ &= \left(\mathcal{R}^1\mathcal{D}x\right)[2^k-2]\\ &\overset{(5.33)}{=} g_1^{k-0}(x), \text{for all } \|x\|_\infty\leq D,\ k\geq 3.\end{aligned}$$

Hence, with $\widetilde{k}_1 = 0$, this establishes (5.36) for $\ell = 1$. Furthermore, we have, for all $\|x\|_\infty \leq D,\ k \geq 3$,

$$\begin{aligned}A^2\left(\widehat{\mathcal{K}}^1\mathcal{D}x\right)[2^k-2]+b^2 &= A_o^2W^1\left(\widehat{\mathcal{K}}^1\mathcal{D}x\right)[2^k-2]+b_o^2\\ &\overset{(5.22)}{=} A_o^2\left(\mathcal{K}^2\mathcal{D}\left(\mathcal{R}^1\mathcal{D}x\right)[2^{k-1}-2]\right)[2^{k-1}-2]+b_o^2\\ &= \left(\mathcal{R}^2\mathcal{D}\left(\mathcal{R}^1\mathcal{D}x\right)[2^{k-1}-2]\right)[2^{k-1}-2]\\ &\overset{(5.33)}{=} \left(g_2^{k-1}\circ g_1^{k-1}\right)(x).\end{aligned}$$

With $\widetilde{k}_1 = 1$ and $\widetilde{k}_2 = 1$, this establishes (5.36) for $\ell = 2$. Finally, by Definition 5.3.2, we obtain the size of $\widehat{\mathcal{R}^1}$ as

$$\mathcal{M}_{\text{hid}}(\widehat{\mathcal{R}^1}) = \mathcal{M}_{\text{hid}}(\mathcal{R}^1) + 2\mathcal{M}_{\text{out}}(\mathcal{R}^1) + \mathcal{M}_{\text{hid}}(\mathcal{R}^2) + 5,$$

which establishes (5.37). This shows that the statement of Lemma 5.3.4 holds for $L = 1$ and thus establishes the base case of the induction over L.

Next, we assume that Lemma 5.3.4 holds for some $L \geq 1$ and show that it holds for $L + 1$ as well. Specifically, let $\mathcal{R}^1, \ldots, \mathcal{R}^{2^{L+1}}$ be RNNs satisfying (5.31) and (5.32) with constants

$D_1, \dots, D_{2^{L+1}}, D^h$. We first invoke Lemma 5.3.4 for $\mathcal{R}^1, \dots, \mathcal{R}^{2^L}$ and denote the resulting RNN by $\widehat{\mathcal{R}}^a = \widehat{\mathcal{Q}}^a \widehat{\mathcal{K}}^a$. By (5.34) we have

$$\left\| \left(\widehat{\mathcal{R}}^a \mathcal{D} x \right)[t] \right\|_\infty \leq D_{2^L} \quad \text{and} \quad \left\| \left(\widehat{\mathcal{K}}^a \mathcal{D} x \right)[t] \right\|_\infty \leq D^h, \quad \forall t \in \mathbb{N}_0, \ \forall x \in [-D, D]^{\mathcal{M}_{\text{in}}(\mathcal{R}^1)}. \tag{5.40}$$

Similarly, we invoke Lemma 5.3.4 for $\mathcal{R}^{2^L+1}, \dots, \mathcal{R}^{2^{L+1}}$ and denote the resulting RNN by $\widehat{\mathcal{R}}^b = \widehat{\mathcal{Q}}^b \widehat{\mathcal{K}}^b$, which satisfies

$$\left\| \left(\widehat{\mathcal{R}}^b \mathcal{D} x \right)[t] \right\|_\infty \leq D_{2^{L+1}} \quad \text{and} \quad \left\| \left(\widehat{\mathcal{K}}^b \mathcal{D} x \right)[t] \right\|_\infty \leq D^h, \quad \forall t \in \mathbb{N}_0, \ \forall x \in [-D_{2^L}, D_{2^L}]^{\mathcal{M}_{\text{in}}(\mathcal{R}^{2^L+1})}. \tag{5.41}$$

Now we set $\widehat{\mathcal{R}}^{L+1} = \widehat{\mathcal{Q}}^{L+1} \widehat{\mathcal{K}}^{L+1} = \mathring{\Psi}_{D^h, D^h}(\widehat{\mathcal{R}}^b, \widehat{\mathcal{R}}^a)$. By (5.40), Conditions (A1) and (A3) of Theorem 5.3.3 are satisfied. Further, by (5.41), Conditions (A4) and (A2) of Theorem 5.3.3 are satisfied. Thus, (5.34) holds for $\widehat{\mathcal{R}}^{L+1}$ as a consequence of (B1) and (B2) in Theorem 5.3.3. By (5.20) in Theorem 5.3.3 we have

$$\begin{aligned}
(\widehat{\mathcal{R}}^{L+1} \mathcal{D} x)[2^k - 2] = & \Big(\widehat{\mathcal{R}}^b \mathcal{D} \Big(\\
& \qquad (\widehat{\mathcal{R}}^a \mathcal{D} x)[2^{k-1} - 2] \\
& \qquad \Big) \Big)[2^{k-1} - 2], \qquad \text{for } k \geq 3, \ \forall x \in [-D, D]^{\mathcal{M}_{\text{in}}(\mathcal{R}^1)},
\end{aligned}$$

which, upon using (5.35) for $\widehat{\mathcal{R}}^b$ and $\widehat{\mathcal{R}}^a$, yields

$$\begin{aligned}
(\widehat{\mathcal{R}}^{L+1} \mathcal{D} x)[2^k - 2] &= \left(g_{2^{L+1}}^{k-1-L} \circ \cdots \circ g_{2^L+1}^{k-1-L} \right) \Big(\\
&\qquad \left(g_{2^L}^{k-1-L} \circ \cdots \circ g_1^{k-1-L} \right)(x) \Big) \\
&= \left(g_{2^{L+1}}^{k-(L+1)} \circ \cdots \circ g_1^{k-(L+1)} \right)(x) \qquad \text{for } k \geq 3 + L, \ \forall x \in [-D, D]^{\mathcal{M}_{\text{in}}(\mathcal{R}^1)}.
\end{aligned}$$

This establishes (5.35) for $\widehat{\mathcal{R}}^{L+1}$.

Next, we establish (5.36) for $\widehat{\mathcal{R}}^{L+1}$ and first treat the case that $\ell \in \{1, \dots, 2^L\}$. To this end fix $\ell \in \{1, \dots, 2^L\}$ arbitrarily and note, that, upon using (5.36) for $\widehat{\mathcal{R}}^a$, there are A_a^ℓ, b_a^ℓ and $\widetilde{k}_1^a, \dots, \widetilde{k}_\ell^a \in \{0, \dots, \lceil \log \ell \rceil\}$ such that

$$A_a^\ell \left(\widehat{\mathcal{K}}^a \mathcal{D} x \right)[2^k - 2] + b_a^\ell = \left(g_\ell^{k - \widetilde{k}_\ell^a} \circ \cdots \circ g_1^{k - \widetilde{k}_1^a} \right)(x) \qquad \forall k \geq L + 2, \ \forall x \in [-D, D]^{\mathcal{M}_{\text{in}}(\mathcal{R}^1)}. \tag{5.42}$$

Now we set,

$$A^\ell \coloneqq A_a^\ell M, \quad b^\ell \coloneqq b_a^\ell, \quad \text{and} \quad \widetilde{k}_1 \coloneqq \widetilde{k}_1^a, \ \dots, \ \widetilde{k}_\ell \coloneqq \widetilde{k}_\ell^a,$$

with $M = \mathring{\mathcal{M}}(\widehat{\mathcal{R}}^b, \widehat{\mathcal{R}}^a)$, and compute

$$\begin{aligned} A^\ell \left(\widehat{\mathcal{K}}^{L+1}\mathcal{D}x\right)[2^k-2] + b^\ell &= A_a^\ell M \left(\widehat{\mathcal{K}}^{L+1}\mathcal{D}x\right)[2^k-2] + b_a^\ell \\ &\overset{(5.21)}{=} A_a^\ell \left(\widehat{\mathcal{K}}^{a}\mathcal{D}x\right)[2^k-2] + b_a^\ell \\ &\overset{(5.42)}{=} \left(g_\ell^{k-\widetilde{k}_\ell^a} \circ \cdots \circ g_1^{k-\widetilde{k}_1^a}\right)(x) \\ &= \left(g_\ell^{k-\widetilde{k}_\ell} \circ \cdots \circ g_1^{k-\widetilde{k}_1}\right)(x) \end{aligned}$$

Since ℓ was arbitrary this establishes (5.36) for the case that $\ell \in \{1, \dots, 2^L\}$. Next, we treat the remaining case and thus fix $\ell \in \{2^L+1, \dots, 2^{L+1}\}$ arbitrarily. Let $\ell' = \ell - 2^L \in \{1, \dots, 2^L\}$, and note, that, upon using (5.36) for $\widehat{\mathcal{R}}^b$, there are $A_b^{\ell'}, b_b^{\ell'}$ and

$$\widetilde{k}_1^b, \dots, \widetilde{k}_{\ell'}^b \in \{0, \dots, \lceil \log \ell' \rceil\}, \tag{5.43}$$

such that

$$A_b^{\ell'}\left(\widehat{\mathcal{K}}^b\mathcal{D}x\right)[2^k-2] + b_b^{\ell'} = \left(g_{2^L+\ell'}^{k-\widetilde{k}_{\ell'}} \circ \cdots \circ g_{2^L+1}^{k-\widetilde{k}_1}\right)(x) \qquad \forall k \geq L+2,\ \forall x \in [-D_{2^L}, D_{2^L}]^{\mathcal{M}_{\text{in}}(\mathcal{R}^{2^L+1})}. \tag{5.44}$$

Now we set, with $W = \mathring{\mathcal{W}}(\widehat{\mathcal{R}}^b, \widehat{\mathcal{R}}^a)$,

$$\begin{aligned} A^\ell &\coloneqq A_b^{\ell'} W, \quad b^\ell \coloneqq b_b^\ell, \quad \text{and} \\ \widetilde{k}_1 &\coloneqq L+1,\ \dots,\ \widetilde{k}_{2^L} \coloneqq L+1,\ \widetilde{k}_{2^L+1} \coloneqq \widetilde{k}_1^b + 1,\ \dots,\ \widetilde{k}_\ell \coloneqq \widetilde{k}_{\ell'}^b + 1, \end{aligned} \tag{5.45}$$

and note that by (5.43), $\widetilde{k}_1, \dots, \widetilde{k}_\ell \in \{0, \dots, L+1\}$ since $\ell' = \ell - 2^L \leq 2^L$. Since, $\ell \in \{2^L + 1, \dots, 2^{L+1}\}$ we have $\lceil \log \ell \rceil = L+1$ and can thus equivalently write $\widetilde{k}_1, \dots, \widetilde{k}_\ell \in \{0, \dots, \lceil \log \ell \rceil\}$. Next, we compute

$$\begin{aligned} A^\ell \left(\widehat{\mathcal{K}}^{L+1}\mathcal{D}x\right)[2^k-2] + b^\ell &= A_b^{\ell'} W \left(\widehat{\mathcal{K}}^{L+1}\mathcal{D}x\right)[2^k-2] + b_b^{\ell'} \\ &\overset{(5.22)}{=} A_b^{\ell'} \left(\widehat{\mathcal{K}}^b \mathcal{D} \left(\widehat{\mathcal{R}}^a \mathcal{D}x\right)[2^{k-1}-2]\right)[2^{k-1}-2] + b_b^{\ell'} \\ &\overset{(5.44)}{=} \left(g_\ell^{k-1-\widetilde{k}_{\ell'}^b} \circ \cdots \circ g_{2^L+1}^{k-1-\widetilde{k}_1^b}\right)\left(\left(\widehat{\mathcal{R}}^a \mathcal{D}x\right)[2^{k-1}-2]\right) \\ &\overset{(5.35) \text{ for } \widehat{\mathcal{R}}^a}{=} \left(g_\ell^{k-1-\widetilde{k}_{\ell'}^b} \circ \cdots \circ g_{2^L+1}^{k-1-\widetilde{k}_1^b}\right)\left(\left(g_{2^L}^{k-1-L} \circ \cdots \circ g_1^{k-1-L}\right)(x)\right), \\ &\overset{(5.45)}{=} \left(g_\ell^{k-\widetilde{k}_\ell} \circ \cdots \circ g_1^{k-\widetilde{k}_1}\right)(x). \end{aligned}$$

Thus, we have established (5.36) for the case $\ell \in \{2^L+1, \dots, 2^{L+1}\}$ as well.

Lastly, we establish (5.37) as follows

$$\begin{aligned}
\mathcal{M}_{\text{hid}}(\widehat{\mathcal{R}}^{L+1}) &= \mathcal{M}_{\text{hid}}(\widehat{\mathcal{R}}^{a}) + 2\mathcal{M}_{\text{out}}(\widehat{\mathcal{R}}^{a}) + \mathcal{M}_{\text{hid}}(\widehat{\mathcal{R}}^{b}) + 5 \\
&= \left(\sum_{\ell=1}^{2^L} \mathcal{M}_{\text{hid}}(\mathcal{R}^{\ell}) + 2\sum_{\ell=1}^{2^L-1} \mathcal{M}_{\text{out}}(\mathcal{R}^{\ell}) + 5(2^L-1)\right) \\
&\quad + 2\mathcal{M}_{\text{out}}(\mathcal{R}^{2^L}) + 5 \\
&\quad + \left(\sum_{\ell=1}^{2^L} \mathcal{M}_{\text{hid}}(\mathcal{R}^{2^L+\ell}) + 2\sum_{\ell=1}^{2^L-1} \mathcal{M}_{\text{out}}(\mathcal{R}^{2^L+\ell}) + 5(2^L-1)\right) \\
&= \sum_{\ell=1}^{2^{L+1}} \mathcal{M}_{\text{hid}}(\mathcal{R}^{\ell}) + 2\sum_{\ell=1}^{2^{L+1}-1} \mathcal{M}_{\text{out}}(\mathcal{R}^{\ell}) + 5\left((2^L-1)+(2^L-1)+1\right).
\end{aligned}$$

Hence, we have shown that all properties specified by Lemma 5.3.4 hold for $\widehat{\mathcal{R}}^{L+1}$ as well, thus establishing the inductive step and therefore completing the proof. □

It is easy to extend this result to the concatenation of an arbitrary number of RNNs, as opposed to a power of two. Specifically we can simply add an appropriate number of dummy networks realizing the map $x \to 0$ and use (5.36) to access the approximation of the composition without the dummy networks. We formalize this next.

Corollary 5.3.5. *Fix $D \geq 1$, $L \in \mathbb{N}$, and let $\mathcal{R}^1, \dots, \mathcal{R}^L$ be RNNs with $\mathcal{M}_{\text{in}}(\mathcal{R}^{\ell+1}) = \mathcal{M}_{\text{out}}(\mathcal{R}^{\ell})$. Furthermore, assume there are constants $D_1, \dots, D_L$ and $D^h \geq \max\{2, \max_{\ell\in\{1,\dots,L\}} D_\ell\}$ such that, for all $\ell \in \{1, \dots, L\}$,*

$$\begin{aligned}
\left\|(\mathcal{R}^{\ell}\mathcal{D}x)[t]\right\|_\infty &\leq D_\ell, && \forall \|x\|_\infty \leq D_{\ell-1}, \quad \forall t \geq 0 \\
\left\|(\mathcal{K}^{\ell}\mathcal{D}x)[t]\right\|_\infty &\leq D^h, && \forall \|x\|_\infty \leq D_{\ell-1}, \quad \forall t \geq 0,
\end{aligned}$$

where we set $D_0 = D$. Then there is a hidden state operator (Definition 5.1.1) $\widehat{\mathcal{K}}$ with size

$$\mathcal{M}_{\text{hid}}(\widehat{\mathcal{K}}) \leq \sum_{\ell=1}^{L} \mathcal{M}_{\text{hid}}(\mathcal{R}^{\ell}) + 2\sum_{\ell=1}^{L} \mathcal{M}_{\text{out}}(\mathcal{R}^{\ell}) + 13L, \tag{5.46}$$

such that, for every $\ell \in \{1, \dots, L\}$, there are A^ℓ, b^ℓ and $\widetilde{k}_1, \dots, \widetilde{k}_\ell \leq \lceil \log \ell \rceil$, such that,

$$A^{\ell}\left(\widehat{\mathcal{K}}\mathcal{D}x\right)[2^k - 2] + b^{\ell} = \left(g_{\ell}^{k-\widetilde{k}_\ell} \circ \cdots \circ g_1^{k-\widetilde{k}_1}\right)(x) \qquad \forall k \geq \lceil \log L \rceil + 2,\ \forall x \in [-D, D]^{\mathcal{M}_{\text{in}}(\mathcal{R}^1)}, \tag{5.47}$$

where g_ℓ^k is defined as in (5.33).

Proof. We first define the dummy RNN $\mathcal{R}_d^{\mathrm{o}} = \mathcal{Q}_d^{\mathrm{o}}\mathcal{K}_d^{\mathrm{o}}$ for input dimension $d \in \mathbb{N}$ with weights

$$A_h = \begin{pmatrix} 0 \end{pmatrix} \qquad A_x = \begin{pmatrix} 0 \mathbf{1}_d^T \end{pmatrix} \qquad b_h = \begin{pmatrix} 0 \end{pmatrix} \qquad A_o = \begin{pmatrix} 0 \end{pmatrix} \qquad b_o = \begin{pmatrix} 0 \end{pmatrix}.$$

It satisfies $(\mathcal{R}_d^{\mathrm{o}}\mathcal{D}x)[t] = 0$ for all $x \in \mathbb{R}^d$. Next, we define $L' \coloneqq \lceil \log L \rceil$, invoke Lemma 5.3.4 for

$$\mathcal{R}^1, \dots, \mathcal{R}^L, \underbrace{\mathcal{R}^{\mathrm{o}}_{\mathcal{M}_{\mathrm{out}}(\mathcal{R}^L)}, \mathcal{R}^{\mathrm{o}}_1, \dots, \mathcal{R}^{\mathrm{o}}_1}_{2^{L'}-L \text{ dummy networks}},$$

and denote the resulting network by $\widehat{\mathcal{R}} = \widehat{\mathcal{Q}}\widehat{\mathcal{K}}$. Now $\widehat{\mathcal{K}}$ is the desired hidden state operator since, by (5.36) for every $\ell \in \{1, \dots, 2^{L'}\}$ there are A^ℓ, b^ℓ and $\widetilde{k}_1, \dots, \widetilde{k}_\ell \leq \lceil \log \ell \rceil$ such that

$$A^\ell \left(\widehat{\mathcal{K}}^L \mathcal{D}x\right)[2^k - 2] + b^\ell = \left(g_\ell^{k-\widetilde{k}_\ell} \circ \cdots \circ g_1^{k-\widetilde{k}_1}\right)(x) \qquad \forall k \geq L' + 2,\ \forall x \in [-D, D]^{\mathcal{M}_{\mathrm{in}}(\mathcal{R}^1)}.$$

Restricting to $\ell \in \{1, \dots, L\}$ yields (5.47).

Lastly, we determine the hidden state size of $\widehat{\mathcal{K}}$. If $L = 2^{L'}$ we have invoked Lemma 5.3.4 without any dummy networks and (5.46) follows directly from (5.37). Otherwise, we have

$$\begin{aligned}
\mathcal{M}_{\mathrm{hid}}(\widehat{\mathcal{K}}) &\overset{(5.37)}{=} \sum_{\ell=1}^{L} \mathcal{M}_{\mathrm{hid}}(\mathcal{R}^\ell) + (2^{L'} - L)\mathcal{M}_{\mathrm{hid}}(\mathcal{R}^{\mathrm{o}}) + 2\left(\sum_{\ell=1}^{L} \mathcal{M}_{\mathrm{out}}(\mathcal{R}^\ell) + (2^{L'} - L - 1)\mathcal{M}_{\mathrm{out}}(\mathcal{R}^{\mathrm{o}})\right) \\
&\qquad + 5(2^{L'} - 1) \\
&\overset{(i)}{\leq} \sum_{\ell=1}^{L} \mathcal{M}_{\mathrm{hid}}(\mathcal{R}^\ell) + L + 2\left(\sum_{\ell=1}^{L} \mathcal{M}_{\mathrm{out}}(\mathcal{R}^\ell) + (L-1)\right) + 5(2L-1) \\
&\leq \sum_{\ell=1}^{L} \mathcal{M}_{\mathrm{hid}}(\mathcal{R}^\ell) + 2\sum_{\ell=1}^{L} \mathcal{M}_{\mathrm{out}}(\mathcal{R}^\ell) + 13L,
\end{aligned}$$

where in (i) we used $2^{L'} \leq 2L$. □

5.4. Approximation of monomials

Throughout this section, we fix $D \geq 1$ and $L \in \mathbb{N}$ arbitrarily in order to simplify our exposition. We now wish to construct an RNN approximating the function $x \to (x, x^2, \dots, x^N)$ with $N \coloneqq 2^L$, by applying Corollary 5.3.5 to concatenate the RNNs approximating the squaring function and the multiplication operation. To this end we organize the powers in a pyramid as illustrated in Figure 5.3, which allows expressing the monomials of degree up to N using $L = \log N$ concatenations. Specifically, the ℓ-th row in Figure 5.3 corresponds to the application of the function f^ℓ as defined next.

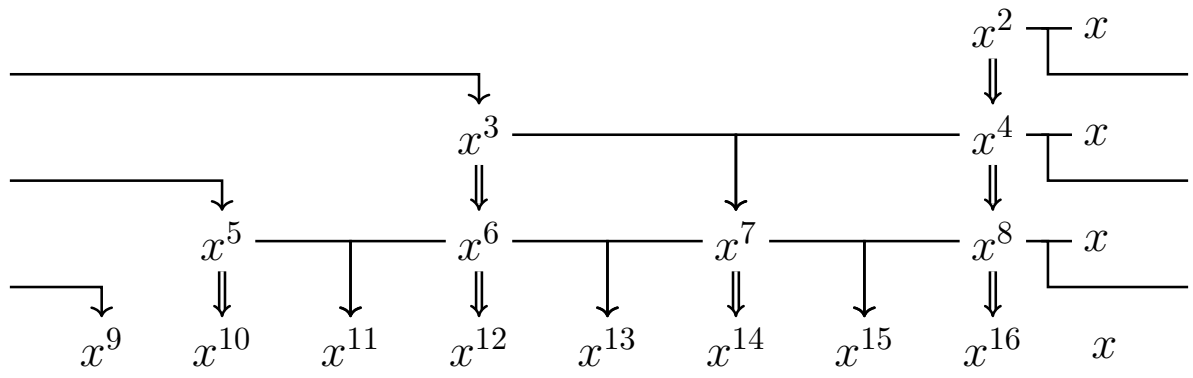


Figure 5.3: Expressing monomials of degree up to $N = 16$ in terms of iterated squaring ($\Rightarrow$) and multiplication ($\rightarrow$).

Definition 5.4.1. *We define maps $f^1 : \mathbb{R} \to \mathbb{R}^2$ and $f^\ell : \mathbb{R}^{2^{\ell-2}+1} \to \mathbb{R}^{2^{\ell-1}+1}$ for $\ell \in \mathbb{N}$ as follows*

$$f^1 : x \to \begin{pmatrix} x^2 & x \end{pmatrix}^T$$
$$f^\ell : x \to \begin{pmatrix} x_{2^{\ell-2}} x_{2^{\ell-2}+1} & x_1^2 & x_1 x_2 & x_2^2 & x_2 x_3 & \dots & x_{2^{\ell-2}-1} x_{2^{\ell-2}} & x_{2^{\ell-2}}^2 & x_{2^{\ell-2}+1} \end{pmatrix}^T$$

I.e., for $x \in \mathbb{R}^{2^{\ell-2}+1}$,

$$f^\ell(x)_i = \begin{cases} x_{2^{\ell-2}} x_{2^{\ell-2}+1} & \text{if } i = 1 \\ x_k^2 & \text{if } i = 2k \text{ for some } k \in \{1, \dots, 2^{\ell-2}\} \\ x_k x_{k+1} & \text{if } i = 2k+1 \text{ for some } k \in \{1, \dots, 2^{\ell-2} - 1\} \\ x_{2^{\ell-2}+1} & \text{if } i = 2^{\ell-1} + 1 \end{cases} \tag{5.48}$$

We then have $(f^\ell \circ \cdots \circ f^1)(x) = \begin{pmatrix} x^{2^{\ell-1}+1} & \dots & x^{2^\ell} & x \end{pmatrix}^T$, which we prove formally next.

Lemma 5.4.2. *For any $\ell \in \mathbb{N}$, define $F^\ell \coloneqq f^\ell \circ \cdots \circ f^1$ we have, for all $x \in \mathbb{R}$ that*

$$\begin{aligned} F^\ell(x)_i &= x^{2^{\ell-1}+i}, \qquad \text{for } i \in \{1, \dots, 2^{\ell-1}\}, \quad \text{and} \\ F^\ell(x)_{2^{\ell-1}+1} &= x. \end{aligned} \tag{5.49}$$

Proof. Fix $x \in \mathbb{R}$. The proof is by induction. To establish the base case, note that, for $\ell = 1$, we have

$$f^1(x)_1 = x^2 = x^{2^0+1}$$
$$f^1(x)_{2^0+1} = f^1(x)_2 = x.$$

Next, assume (5.49) holds for some $\ell \in \mathbb{N}$. We denote $z \coloneqq (f^\ell \circ \cdots \circ f^0)(x) \in \mathbb{R}^{2^{\ell-1}+1}$, which

by the inductive assumption, satisfies

$$z_i = x^{2^{\ell-1}+i} \qquad \text{for } i \in \{1, \dots, 2^{\ell-1}\}, \quad \text{and}$$
$$z_{2^{\ell-1}+1} = x.$$

Next, we compute according to (5.48)

$$\left(f^{\ell+1}(z)\right)_1 = z_{2^{\ell-1}} \cdot z_{2^{\ell-1}+1} = x^{2^{\ell-1}+2^{\ell-1}} \cdot x = x^{2^\ell+1}.$$

Next, for $i = 2k$ for some $k \in \{1, \dots, 2^{\ell-1}\}$, we have

$$\left(f^{\ell+1}(z)\right)_i = (z_k)^2 = \left(x^{2^{\ell-1}+k}\right)^2 = x^{2 2^{\ell-1}+2k} = x^{2^\ell+i},$$

and, for $i = 2k+1$ for some $k \in \{1, \dots, 2^{\ell-1}-1\}$, we have

$$\left(f^{\ell+1}(z)\right)_i = z_k \cdot z_{k+1} = x^{2^{\ell-1}+k} x^{2^{\ell-1}+k+1} = x^{2 2^{\ell-1}+2k+1} = x^{2^\ell+i}.$$

Lastly,

$$\left(f^{\ell+1}(z)\right)_{2^\ell+1} = z_{2^{\ell-1}+1} = x.$$

Hence, (5.49) holds for $\ell+1$ as well, thus completing the induction. □

The following bound follows directly from (5.49).

Corollary 5.4.3. *For any $\ell \in \mathbb{N}$, and for all $x \in [-D, D]$ we have*

$$\left\|F^\ell(x)\right\|_\infty \le D^{2^\ell}.$$

In addition, we will need the following properties.

Lemma 5.4.4. *For $\ell \in \mathbb{N}$, and for all $x, y \in [-D^{2^{\ell-1}}, D^{2^{\ell-1}}]^{2^{\ell-2}+1}$ we have*

$$\left\|f^\ell(x)\right\|_\infty \le D^{2^\ell} \qquad \text{and} \qquad \left\|f^\ell(x) - f^\ell(y)\right\|_\infty \le 2D^{2^{\ell-1}} \|x-y\|_\infty.$$

Proof. Examining (5.48) we have that the entries of $f^\ell(x)$ have either the form $x_i x_j$ for $i, j \in \{1, \dots, 2^{\ell-2}\}$ (possibly $i = j$) or $x_{2^{\ell-2}+1}$ for the last entry. For the last entry the claim is trivial. Now, fix $x, y \in [-D^{2^{\ell-1}}, D^{2^{\ell-1}}]^{2^{\ell-2}+1}$ and $i, j \in \{1, \dots, 2^{\ell-2}\}$ arbitrarily. Wwe have

$$|x_i x_j| = |x_i||x_j| \le D^{2^{\ell-1}} D^{2^{\ell-1}} = D^{2^\ell}$$

and

$$|x_i x_j - y_i y_j| \le |x_i x_j - x_i y_j| + |x_i y_j - y_i y_j| = |x_i||x_j - y_j| + |y_j||x_i - y_i| \le 2D^{2^{\ell-1}} \|x-y\|_\infty.$$

As i, j and x, y where arbitrary, the statement follows. □

Next, we construct RNN approximations of the maps f^ℓ. In addition to the RNNs that approximate squaring and multiplication, we also need an RNN realizing the identity.

Lemma 5.4.5. *There is an RNN $\mathcal{R}^{Id} = \mathcal{Q}^{Id}\mathcal{K}^{Id}$ such that, for all $x \in \mathbb{R}$,*

$$(\mathcal{R}^{Id}\mathcal{D}x)[t] = x, \quad \left\|(\mathcal{K}^{Id}\mathcal{D}x)[t]\right\|_\infty = |x| \quad \forall t \in \mathbb{N}_0.$$

Furthermore, $\mathcal{M}_{\text{hid}}(\mathcal{R}^{Id}) = 2$.

Proof. The weights are given by

$$A_h = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad A_x = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \qquad b_h = 0 \qquad A_o = \begin{pmatrix} 1 & -1 \end{pmatrix} \qquad b_o = 0,$$

and the statement then follows from the identity $x = \rho(x) - \rho(-x)$. □

We start by building an RNN that approximates f^1.

Lemma 5.4.6. *There is an RNN $\mathcal{R}^1$ with $\mathcal{M}_{\text{out}}(\mathcal{R}^1) = 2$ such that for all $x \in [-D, D]$ we have*

$$\begin{aligned} \left|((\mathcal{R}^1\mathcal{D}x)[t])_1 - x^2\right| &\leq \frac{D^2}{2}4^{-t}, \\ ((\mathcal{R}^1\mathcal{D}x)[t])_2 &= x, \qquad\qquad \forall t \in \mathbb{N}_0 \end{aligned}$$

and

$$|(\mathcal{K}^1\mathcal{D}x)[t]| \leq D \text{ and } |(\mathcal{R}^1\mathcal{D}x)[t]| \leq D^2.$$

Furthermore, $\mathcal{M}_{\text{hid}}(\mathcal{R}^1) = 9$.

Proof. We consider the RNN implementing the identity from Lemma 5.4.5 and the RNN approximating x^2 from Theorem 5.2.4. Thus, upon using Lemma 5.2.5, we obtain an RNN that yields the desired behavior for inputs $\begin{pmatrix} x \\ x \end{pmatrix}$. Lastly, upon noting that $\begin{pmatrix} x \\ x \end{pmatrix} = Ax$ with $A \coloneqq \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, we use Lemma 5.2.6, to obtain $\mathcal{R}^1$ with the desired behavior. □

Next, we construct RNNs that approximate f^ℓ for arbitrary $\ell \geq 2$.

Lemma 5.4.7. *For every $\ell \geq 2$, there is an RNN $\mathcal{R}^\ell = \mathcal{Q}^\ell\mathcal{K}^\ell$ such that for all $x \in \left[-D^{2^{\ell-1}}, D^{2^{\ell-1}}\right]^{2^{\ell-2}+1}$ and all $t \in \mathbb{N}_0$ we have*

$$\left\|(\mathcal{R}^\ell\mathcal{D}x)[t] - f^\ell(x)\right\|_\infty \leq \frac{D^{2^\ell}}{2}\, 4^{-t}, \quad |(\mathcal{R}^\ell\mathcal{D}x)[t]| \leq D^{2^\ell}, \quad |(\mathcal{K}^\ell\mathcal{D}x)[t]| \leq D.$$

Furthermore, $\mathcal{M}_{\text{hid}}(\mathcal{R}^\ell) \leq 10 \cdot 2^\ell$.

Proof. Fix $\ell \in \mathbb{N}$ arbitrarily. First, let

$$A \coloneqq \left(\begin{array}{ccccccc} 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ \hline 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & \dots & 0 & 0 & 0 \\ \hline \vdots & & & \ddots & & & \vdots \\ \hline 0 & 0 & 0 & \dots & 1 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{array}\right) \in \mathbb{R}^{(3\cdot 2^{\ell-2}+1)\times(2^{\ell-2}+1)}.$$

Next, let $\mathcal{R}^{\times}$ and $\mathcal{R}^{\mathrm{Sq}}$ be the multiplication RNN and the square RNN from Theorem 5.2.8 and Theorem 5.2.4 respectively, for input bound $D^{2^{\ell-1}}$. Upon noting that $(D^{2^{\ell-1}})^2 = D^{2^\ell}$ we thus have for $x, x_1, x_2 \in [-D^{2^{\ell-1}}, D^{2^\ell-1}]$ and for all $t \in \mathbb{N}_0$ that

$$\left|\left(\mathcal{R}^{\times}\mathcal{D}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right)[t] - (x_1 \cdot x_2)\right| \leq \frac{D^{2^\ell}}{2}\, 4^{-t}, \quad \left|\left(\mathcal{K}^{\times}\mathcal{D}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right)[t]\right| \leq D, \quad \left|\left(\mathcal{R}^{\times}\mathcal{D}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}\right)[t]\right| \leq D^{2^\ell},$$

$$\left|\left(\mathcal{R}^{\mathrm{Sq}}\mathcal{D}x\right)[t] - x^2\right| \leq \frac{D^{2^\ell}}{2}\, 4^{-t}, \quad \left|\left(\mathcal{K}^{\mathrm{Sq}}\mathcal{D}x\right)[t]\right| \leq 1 \leq D, \qquad \left|\left(\mathcal{R}^{\mathrm{Sq}}\mathcal{D}x\right)[t]\right| \leq D^{2^\ell}.$$

Furthermore, let $\mathcal{R}^{\mathrm{Id}}$ be the RNN from Lemma 5.4.5. Now, let $\widetilde{\mathcal{R}}$ be the RNN obtained by using Lemma 5.2.5 to execute the RNNs

$$\mathcal{R}^{\times}, \mathcal{R}^{\mathrm{Sq}}, \mathcal{R}^{\times}, \mathcal{R}^{\mathrm{Sq}}, \dots, \mathcal{R}^{\times}, \mathcal{R}^{\mathrm{Sq}}, \mathcal{R}^{\mathrm{Id}}$$

in parallel. I.e. there are $2^{\ell-2}$ copies of $\mathcal{R}^{\times}$ and $\mathcal{R}^{\mathrm{Sq}}$. Next, we apply Lemma 5.2.6 with A and $\widetilde{\mathcal{R}}$ to obtain the desired RNN $\mathcal{R}^\ell$. The size is given by

$$\mathcal{M}_{\mathrm{hid}}(\mathcal{R}^\ell) = 2^{\ell-2}\left(\mathcal{M}_{\mathrm{hid}}(\mathcal{R}^{\mathrm{Sq}}) + \mathcal{M}_{\mathrm{hid}}(\mathcal{R}^{\times})\right) + \mathcal{M}_{\mathrm{hid}}(\mathcal{R}^{\mathrm{Id}}) = 2^{\ell-2}(7+14)+2 = 2^\ell\frac{21}{4}+2 \leq 10\cdot 2^\ell$$

□

We are now ready to assemble the RNNs approximating f^ℓ into a single RNN approximating F^L. We start by specifying its hidden state operator.

Definition 5.4.8. *We invoke Corollary 5.3.5 with $\mathcal{R}^1, \dots, \mathcal{R}^L$ as given by Lemmata 5.4.6 and*

5.4.7 and denote the resulting hidden state operator by $\mathcal{K}^\pi$.

Next, we establish its size.

Lemma 5.4.9. *We have $\mathcal{M}_{\text{hid}}(\mathcal{K}^\pi) \leq 60N$.*

Proof.

$$\begin{aligned}
\mathcal{M}_{\text{hid}}(\mathcal{K}^\pi) &\overset{(5.46)}{\leq} \sum_{\ell=1}^{L} \mathcal{M}_{\text{hid}}(\mathcal{R}^\ell) + 2\sum_{\ell=1}^{L} \mathcal{M}_{\text{out}}(\mathcal{R}^\ell) + 13L \\
&\leq \sum_{\ell=1}^{L} 10 \cdot 2^\ell + 2\sum_{\ell=1}^{L}(2^{\ell-1}+1) + 13L \\
&= 20\sum_{\ell=0}^{L-1} 2^\ell + 2\sum_{\ell=0}^{L-1}(2^\ell) + 15L \\
&= 20(2^L-1) + 2(2^L-1) + 15L \\
&\leq 22 \cdot 2^L + 15L \\
&\leq 44N + 15\log N \leq 60N
\end{aligned}$$

□

Now recalling (5.33) as

$$g_\ell^k(x) = \left(\mathcal{R}^\ell \mathcal{D} x\right)[2^k - 2] \tag{5.50}$$

yields the following, since the $\mathcal{R}^\ell$ in Definition 5.4.8 are given as in Lemmata 5.4.6 and 5.4.7.

Corollary 5.4.10. *For all $x \in \left[-D^{2^{\ell-1}}, D^{2^{\ell-1}}\right]^{2^{\ell-2}+1}$ (or $x \in [-D, D]$ for $\ell = 1$)*

$$\left\|g_\ell^k(x) - f^\ell(x)\right\|_\infty \leq \frac{1}{2} D^{2^\ell}\, 4^{-2^k+2} = 8D^{2^\ell}\, 4^{-2^k}, \qquad \left\|g_\ell^k(x)\right\|_\infty \leq D^{2^\ell}.$$

We next show that we can read out approximations to all F^ℓ by applying an affine output mapping to the hidden state sequence produced by $\mathcal{K}^\pi$ given in Definition 5.4.8.

Lemma 5.4.11. *For $\ell \in \{1, \ldots, L\}$, there are A^ℓ, b^ℓ such that for all $x \in [-D, D]$ and $k \geq \lceil \log L \rceil + 2$ we have*

$$\left\|\left(A^\ell(\mathcal{K}^\pi \mathcal{D} x)[2^k - 2] + b^\ell\right) - F^\ell(x)\right\|_\infty \leq 8 \cdot 2^\ell D^{2^\ell} 4^{-\frac{1}{2\ell}2^k} =: \epsilon_{l,k}.$$

Proof. Fix $\ell \in \{1, \ldots, L\}$ arbitrarily. By Corollary 5.3.5 there are A^ℓ, b^ℓ and $\widetilde{k}_1, \ldots, \widetilde{k}_\ell \leq \lceil \log \ell \rceil$ such that

$$A^\ell \left(\mathcal{K}^\pi \mathcal{D} x\right)[2^k - 2] + b^\ell = \left(g_\ell^{k-\widetilde{k}_\ell} \circ \cdots \circ g_1^{k-\widetilde{k}_1}\right)(x), \qquad \forall k \geq \lceil \log L \rceil + 2,\ \forall x \in [-D, D]^1. \tag{5.51}$$

We now show that A^ℓ, b^ℓ satisfy the desired properties. To this end we define, for $\ell' \in \{1, \ldots, \ell\}$, $k \geq \lceil \log L \rceil + 2$, the maps

$$G_{\ell'}^k \coloneqq g_{\ell'}^{k-\widetilde{k}_{\ell'}} \circ \cdots \circ g_1^{k-\widetilde{k}_1}.$$

Now, we fix $x \in [-D, D]$ and $k \geq \lceil \log L \rceil + 2$ arbitrarily and show by induction over $\ell' \in \{1, \ldots, \ell\}$ that

$$\left\| G_{\ell'}^k(x) \right\|_\infty \leq D^{2^{\ell'}} \qquad \text{and} \tag{5.52}$$

$$\left\| G_{\ell'}^k(x) - F^{\ell'}(x) \right\|_\infty \leq 8 \cdot D^{2^{\ell'}} 4^{-2^{k - \lceil \log \ell \rceil}} \left(\sum_{i=0}^{\ell'-1} 2^i \right). \tag{5.53}$$

The base case, for $\ell' = 1$, is established by

$$\begin{aligned} \left\| G_1^k(x) \right\|_\infty = \left\| g_1^k(x) \right\|_\infty \leq D^2 \qquad & \text{and} \\ \left\| G_1^k(x) - F^1(x) \right\|_\infty = \left\| g_1^{k-\widetilde{k}_1}(x) - f^1(x) \right\|_\infty \leq 8D^{2^1}\ 4^{-2^{k-\widetilde{k}_1}} \leq 8D^{2^1}\ 4^{-2^{k-\lceil \log \ell \rceil}}, & \end{aligned}$$

where we used Corollary 5.4.10. Next, assume (5.52) and (5.53) holds for some $\ell' \in \{1, \ldots, \ell - 1\}$. We have

$$\left\| G_{\ell'+1}^k(x) \right\|_\infty = \left\| g_{\ell'+1}^{k-\widetilde{k}_{\ell'}} \left(G_{\ell'}^{k-\widetilde{k}_{\ell'}}(x) \right) \right\|_\infty \overset{(5.52)}{\leq} \sup_{\|z\|_\infty \leq D^{2^{\ell'}}} \left\| g_{\ell'+1}^k(z) \right\|_\infty \overset{\text{Cor. } 5.4.10}{\leq} D^{2^{\ell'+1}}.$$

This establishes that (5.52) holds for $\ell' + 1$ as well. Next, we have

$$\begin{aligned} \left\| G_{\ell'+1}^k(x) - F^{\ell'+1}(x) \right\|_\infty &= \left\| g_{\ell'+1}^{k-\widetilde{k}_{\ell'+1}}(G_{\ell'}^k(x)) - f^{\ell'+1}(F^{\ell'}(x)) \right\|_\infty \\ &\leq \left\| g_{\ell'+1}^{k-\widetilde{k}_{\ell'+1}}(G_{\ell'}^k(x)) - f^{\ell'+1}(G_{\ell'}^k(x)) \right\|_\infty + \left\| f^{\ell'+1}(G_{\ell'}^k(x)) - f^{\ell'+1}(F^{\ell'}(x)) \right\|_\infty \\ &\overset{(5.52)}{\leq} \sup_{\|z\|_\infty \leq D^{2^{\ell'}}} \left\| g_{\ell'+1}^{k-\widetilde{k}_{\ell'+1}}(z) - f^{\ell'+1}(z) \right\|_\infty + \left\| f^{\ell'+1}(G_{\ell'}^k(x)) - f^{\ell'+1}(F^{\ell'}(x)) \right\|_\infty \\ &\overset{\text{Cor. } 5.4.10}{\leq} 8D^{2^{\ell'+1}} 4^{-2^{k-\widetilde{k}_{\ell'+1}}} + \left\| f^{\ell'+1}(G_{\ell'}^k(x)) - f^{\ell'+1}(F^{\ell'}(x)) \right\|_\infty \\ &\overset{\text{Lem. } 5.4.4}{\leq} 8D^{2^{\ell'+1}} 4^{-2^{k-\widetilde{k}_{\ell'+1}}} + 2D^{2^{\ell'}} \left\| G_{\ell'}^k(x) - F^{\ell'}(x) \right\|_\infty \\ &\overset{(5.53)}{\leq} 8D^{2^{\ell'+1}} 4^{-2^{k-\widetilde{k}_{\ell'+1}}} + 2D^{2^{\ell'}} \left(8 \cdot D^{2^{\ell'}} 4^{-2^{k-\lceil \log \ell \rceil}} \left(\sum_{i=0}^{\ell'-1} 2^i \right) \right) \\ &\overset{(*)}{\leq} 8D^{2^{\ell'+1}} 4^{-2^{k-\lceil \log \ell \rceil}} + 8 \cdot D^{2^{\ell'+1}} 4^{-2^{k-\lceil \log \ell \rceil}} \left(\sum_{i=0}^{\ell'-1} 2^{i+1} \right) \\ &= 8D^{2^{\ell'+1}} 4^{-2^{k-\lceil \log \ell \rceil}} \left(\sum_{i=0}^{\ell'} 2^i \right) \end{aligned}$$

where in $(*)$ we used $\widetilde{k}_{\ell'+1} \leq \lceil \log \ell \rceil$. This establishes that (5.53) holds for $\ell' + 1$ as well. In

particular, using (5.53) with $\ell' = \ell$, we can thus compute

$$\begin{aligned}\left\|A^\ell\left(\mathcal{K}^\pi\mathcal{D}x\right)[2^k-2]+b^\ell-F^\ell(x)\right\|_\infty &\overset{(5.51)}{=} \left\|G_\ell^k(x)-F^\ell(x)\right\|_\infty \\ &\overset{(5.53)}{\leq} 8\cdot D^{2^\ell}4^{-2^{k-\lceil\log\ell\rceil}}\left(\sum_{i=0}^{\ell-1}2^i\right) \\ &= 8\cdot D^{2^\ell}4^{-2^{k-\lceil\log\ell\rceil}}(2^\ell-1) \\ &\leq 8\cdot 2^\ell\cdot D^{2^\ell}4^{-2^{k-\log\ell-1}} \\ &= 8\cdot 2^\ell\cdot D^{2^\ell}4^{-\frac{1}{2\ell}2^k},\end{aligned}$$

which, since $\ell\in\{1,\dots,L\}$, $k\geq\lceil\log L\rceil+2$ and $x\in[-D,D]$ where arbitrary, completes the proof. □

We are now ready to specify the output mapping for the RNN.

Lemma 5.4.12. *There are A_o^π, b_o^π, such that the output mapping given by $\mathcal{Q}^\pi : h\to A_o^\pi h+b_o^\pi$ satisfies for all $x\in[-D,D]$ and $k\geq\lceil\log L\rceil+2$ that*

$$\begin{aligned}((\mathcal{Q}^\pi\mathcal{K}^\pi\mathcal{D}x)[2^k-2])_1 &= x,\\ |((\mathcal{Q}^\pi\mathcal{K}^\pi x)[2^k-2])_{(2^{\ell-1}+j)}-x^{2^{\ell-1}+j}| &\leq \epsilon_{\ell,k}, \qquad \text{for } \ell\in\{1,\dots,L\},\ j\in\{1,\dots,2^{\ell-1}\},\end{aligned}$$

where $\epsilon_{\ell,k}$ is given in Lemma 5.4.11 and $\mathcal{K}^\pi$ is given in Definition 5.4.8.

Proof. Let A^ℓ, b^ℓ be as in Lemma 5.4.11, define the matrices

$$\begin{aligned}\widetilde{A_o^1} &\coloneqq \begin{pmatrix}0&1\\1&0\end{pmatrix}A^1, & b^1 &\coloneqq \begin{pmatrix}0&1\\1&0\end{pmatrix}b^1,\\ \widetilde{A_o^\ell} &\coloneqq \begin{pmatrix}\mathbb{I}_{2^{\ell-1}} & 0\end{pmatrix}A^\ell, & \widetilde{b_o^\ell} &\coloneqq \begin{pmatrix}\mathbb{I}_{2^{\ell-1}} & 0\end{pmatrix}b^\ell, \quad \text{for } \ell\in\{2,\dots,L-1\},\end{aligned}$$

and the output map

$$\mathcal{Q}^\pi(h)\coloneqq A_o^\pi h+b_o^\pi,\ \text{where } A_o^\pi\coloneqq\begin{pmatrix}\widetilde{A_o^1}\\ \widetilde{A_o^2}\\ \widetilde{A_o^3}\\ \vdots\\ \widetilde{A_o^L}\end{pmatrix}\ \text{and } b_o^\pi\coloneqq\begin{pmatrix}\widetilde{b_o^1}\\ \widetilde{b_o^2}\\ \widetilde{b_o^3}\\ \vdots\\ \widetilde{b_o^L}\end{pmatrix}.$$

We now show, that $\mathcal{Q}^\pi$ satisfies the desired properties. To this end, fix $x\in[-D,D]$ and $k\geq\lceil\log L\rceil+2$ arbitrarily and recall that by (5.51) in the proof of Lemma 5.4.11 we have

$$A^1(\widehat{\mathcal{K}}\mathcal{D}x)[2^k-2]+b^1 = g_1^{k-\widetilde{k}_1}(x)$$

and, by (5.50), $g_1^{k-\widetilde{k}_1}(x) = (\mathcal{R}^1\mathcal{D}x)[2^{k-\widetilde{k}_1} - 2]$. Combining these we get

$$((\mathcal{Q}^\pi\mathcal{K}^\pi\mathcal{D}x)[2^k-2])_1 = \left(A^1(\widehat{\mathcal{K}}\mathcal{D}x)[2^k-2]+b^1\right)_2 = \left(\left(\mathcal{R}^1\mathcal{D}x\right)[2^{k-\widetilde{k}_1}-2]\right)_1 \overset{\text{Lem. 5.4.6}}{=} x.$$

For the second entry we have

$$|((\mathcal{Q}^\pi\mathcal{K}^\pi\mathcal{D}x)[2^k-2])_2 - x^{2^0+1}| = \left|\left(A^1(\mathcal{K}^\pi\mathcal{D}x)[2^k-2]+b^1\right)_1 - F^1(x)_1\right| \overset{\text{Lem. 5.4.11}}{\leq} \epsilon_{1,k}.$$

Furthermore, let $\ell \in \{2,\dots,L\}$, $j \in \{1,\dots,2^{\ell-1}\}$ and consider the $2^{\ell-1}+j$-th entry. We, note that $\widetilde{A_o^1}$ has 2 rows and, for $i \in \{2,\dots,\ell-1\}$, $\widetilde{A_o^i}$ has 2^{i-1} rows. Thus, due to the fact that $2^{\ell-1} = 2 + \sum_{i=2}^{\ell-1} 2^{i-1}$, we have the following.

$$|((\mathcal{Q}^\pi\mathcal{K}^\pi\mathcal{D}x)[2^k-2])_{2^{\ell-1}+j} - x^{2^{\ell-1}+j}| = \left|\left(A^\ell(\widehat{\mathcal{K}}\mathcal{D}x)[2^k-2]+b^\ell\right)_j - F^\ell(x)_j\right| \leq \epsilon_{\ell,k}.$$

This completes the proof. □

As L and D where fixed arbitrarily at the beginning of this section we can thus summarize the results in the following theorem.

Theorem 5.4.13. *Let $L \in \mathbb{N}$, $D \geq 1$, then there exists an RNN $\mathcal{R}^\pi = \mathcal{Q}^\pi\mathcal{K}^\pi$ with*

$$\mathcal{M}_{\text{hid}}(\mathcal{R}^\pi) \leq 60N \tag{5.54}$$

such that for all $x \in [-D, D]$ and all $k \geq \lceil \log L \rceil + 2$ we have

$$\begin{aligned} ((\mathcal{R}^\pi\mathcal{D}x)[2^k-2])_1 &= x, \\ |((\mathcal{R}^\pi x)[2^k-2])_{(2^{\ell-1}+j)} - x^{2^{\ell-1}+j}| &\leq \epsilon_{\ell,k}, \qquad \text{for } \ell \in \{1,\dots,L\},\ j \in \{1,\dots,2^{\ell-1}\}, \end{aligned} \tag{5.55}$$

where $\epsilon_{\ell,k} = 8 \cdot 2^\ell D^{2^\ell} 4^{-\frac{1}{2^\ell}2^k}$.

Proof. Fix $L \in \mathbb{N}$, $D \geq 1$ as desired and observe that all results of this section are valid for these values. Let then $\mathcal{K}^\pi$ be as in Definition 5.4.8 and $\mathcal{Q}^\pi$ be as in Lemma 5.4.12. Now (5.54) follows from Lemma 5.4.9 and (5.55) follows from Lemma 5.4.12. □

5.5. Approximation of polynomials

It is now clear that any polynomial can be approximated by simply applying a linear layer on top of the RNN produced in Theorem 5.4.13. This, however, only guarantees useful outputs whenever the time step is a power of two, specifically $t = 2^k - 2$. To get around this, we use the following lemma to modify the RNN such that — at any time step — it outputs the most recent useful approximation.

Lemma 5.5.1. *Let $B > 0$, and let $\mathcal{R} = \mathcal{QK}$ be an RNN with $\mathcal{M}_{\text{out}}(\mathcal{R}) = 1$. Then there exists an RNN $\mathcal{R}'$ such that $\forall x \in \mathbb{R}$,*

$$(\mathcal{R}'\mathcal{D}x)[2^k - 1 + \ell] = \text{clip}((\mathcal{R}\mathcal{D}x)[2^k - 2], -B, B), \quad \text{for all } k \geq 2, \ell \in \{0, \dots, 2^k - 1\},$$

where

$$\text{clip}(y, -B, B) \coloneqq \begin{cases} -B & \text{if } y, \leq -B \\ B & \text{if } y, \geq B \\ y & \text{else.} \end{cases}$$

Furthermore, its size equals

$$\mathcal{M}_{\text{hid}}(\mathcal{R}') = \mathcal{M}_{\text{hid}}(\mathcal{R}) + 11.$$

Proof. Denote the weights of $\mathcal{R}$ by $A_x \in \mathbb{R}^{m \times d}, A_h \in \mathbb{R}^{m \times m}, b_h \in \mathbb{R}^m, A_o \in \mathbb{R}^{d' \times m}$, and $b_o \in \mathbb{R}^{d'}$ with $d' = 1$. The weights of $\mathcal{R}'$ are specified as follows

$$A_h = \left(\begin{array}{c|c|c|c|c} A_h & 0 & 0 & 0 & 0 \\ \hline A_o & 0 & 0 & 0 & B\widehat{A}_o \\ -A_o & 0 & 0 & 0 & B\widehat{A}_o \\ \hline A_o & 0 & 0 & 0 & 0 \\ -A_o & 0 & 0 & 0 & 0 \\ \hline 0 & \mathbb{I}_2 & -\mathbb{I}_2 & \mathbb{I}_2 & -B\mathbf{1}_2\widehat{A}_o \\ \hline 0 & 0 & 0 & 0 & \widehat{A} \end{array}\right) \quad b_h = \left(\begin{array}{c} b_h \\ \hline b_o - B \\ -b_o - B \\ \hline b_o - B \\ -b_o - B \\ \hline 0 \\ \hline \widehat{b} \end{array}\right) \quad A_x = \left(\begin{array}{c} A_x \\ \hline 0 \\ 0 \\ \hline 0 \\ 0 \\ \hline 0 \\ \hline 0 \end{array}\right)$$

$$A_o = \left(0 \,\middle|\, 1 \quad -1 \,\middle|\, -1 \quad 1 \,\middle|\, 1 \quad -1 \,\middle|\, 0\right) \qquad b_o = 0, \text{ where } \widehat{A}_o \coloneqq \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

and $\widehat{A}$ and $\widehat{b}$ as in (5.15). We fix input $x \in \mathbb{R}^d$ arbitrarily and consider the hidden state sequence in response to the input sequence $(\mathcal{D}x)[\cdot]$, i.e. $h[t] = (\mathcal{K}\mathcal{D}x)[t]$. We divide the hidden state sequence corresponding to the blocks above as follows

$$h[t] = \left(\begin{array}{c} h_1[t] \\ \hline h_2[t] \\ \hline h_3[t] \\ \hline h_4[t] \\ \hline h_5[t] \end{array}\right) = \rho\left(A_h h[t-1] + A_x((Dx)[t]) + b_h\right),$$

and study each block individually. First, note that $h_5[\cdot]$ follows the recursion

$$h_5[t+1] = \rho(\widehat{A}h_5[t] + \widehat{b}), \text{ with } h_5[-1] = 0.$$

Thus, by Lemma 5.3.1, we have that

$$\widehat{A}_o h_5[t] = \widehat{\delta}[t+2]. \tag{5.56}$$

Next, $h_1[t]$ follows the recursion

$$h_1[t] = \rho\left(A_h^f h_1[t-1] + A_x(\mathcal{D}x)[t] + b_h\right) \quad \text{with } h_1[-1] = 0,$$

and thus we have $h_1[t] = (\mathcal{K}\mathcal{D}x)[t]$. Next, we consider the sequence $h_2[t]$, which is given by

$$\begin{aligned} h_2[t] &= \rho\begin{pmatrix} A_o h_1[t-1] + B\widehat{A}_o h_5[t-1] + b_o - B \\ -A_o h_1[t-1] + B\widehat{A}_o h_5[t-1] - b_o - B \end{pmatrix} \\ &= \rho\begin{pmatrix} (A_o(\mathcal{K}\mathcal{D}x)[t-1] + b_o) - B(1 - \widehat{A}_o h_5[t-1]) \\ -(A_o(\mathcal{K}\mathcal{D}x)[t-1] + b_o) - B(1 - \widehat{A}_o h_5[t-1]) \end{pmatrix} \\ &= \rho\begin{pmatrix} (\mathcal{R}\mathcal{D}x)[t-1] - B(1 - \widehat{\delta}[t+1]) \\ -(\mathcal{R}\mathcal{D}x)[t-1] - B(1 - \widehat{\delta}[t+1]) \end{pmatrix}. \end{aligned}$$

Similarly, we get

$$\begin{aligned} h_3[t] &= \rho\begin{pmatrix} A_o h_1[t-1] + b_o - B \\ -A_o h_1[t-1] - b_o - B \end{pmatrix} \\ &= \rho\begin{pmatrix} (\mathcal{R}\mathcal{D}x)[t-1] - B \\ -(\mathcal{R}\mathcal{D}x)[t-1] - B \end{pmatrix} \qquad (5.57) \\ &= \begin{pmatrix} ((\mathcal{R}\mathcal{D}x)[t-1] - B)\, \mathbb{1}_{\{(\mathcal{R}\mathcal{D}x)[t-1] \geq B\}} \\ (-(\mathcal{R}\mathcal{D}x)[t-1] - B)\, \mathbb{1}_{\{-(\mathcal{R}\mathcal{D}x)[t-1] \geq B\}} \end{pmatrix} \\ &= \begin{pmatrix} (\rho\left((\mathcal{R}\mathcal{D}x)[t-1]\right) - B)\, \mathbb{1}_{\{(\mathcal{R}\mathcal{D}x)[t-1] \geq B\}} \\ (\rho\left(-(\mathcal{R}\mathcal{D}x)[t-1]\right) - B)\, \mathbb{1}_{\{-(\mathcal{R}\mathcal{D}x)[t-1] \geq B\}} \end{pmatrix} \qquad (5.58) \end{aligned}$$

Next, we show that

$$\mathbb{I}_2 h_2[t] - \mathbb{I}_2 h_3[t] = \begin{pmatrix} \mathrm{clip}((\mathcal{R}\mathcal{D}x)[t-1], 0, B) \\ \mathrm{clip}(-(\mathcal{R}\mathcal{D}x)[t-1], 0, B) \end{pmatrix} \widehat{\delta}[t+1] \tag{5.59}$$

through a case distinction. If $\widehat{\delta}[t+1] = 0$ we have using expression (5.57)

$$\mathbb{I}_2 h_2[t] - \mathbb{I}_2 h_3[t] = \rho\begin{pmatrix} (\mathcal{R}\mathcal{D}x)[t-1] - B \\ -(\mathcal{R}\mathcal{D}x)[t-1] - B \end{pmatrix} - \rho\begin{pmatrix} (\mathcal{R}\mathcal{D}x)[t-1] - B \\ -(\mathcal{R}\mathcal{D}x)[t-1] - B \end{pmatrix} = 0.$$

For $\widehat{\delta}[t+1]=1$ we get, using expression (5.58), that

$$\begin{aligned}\mathbb{I}_2 h_2[t]-\mathbb{I}_2 h_3[t] &= \rho\begin{pmatrix}(\mathcal{RD}x)[t-1]\\ -(\mathcal{RD}x)[t-1]\end{pmatrix}-\begin{pmatrix}(\rho\left((\mathcal{RD}x)[t-1]\right)-B)\,\mathbb{1}_{\{(\mathcal{RD}x)[t-1]\geq B\}}\\ (\rho\left(-(\mathcal{RD}x)[t-1]\right)-B)\,\mathbb{1}_{\{-(\mathcal{RD}x)[t-1]\geq B\}}\end{pmatrix}\\ &= \begin{pmatrix}\mathrm{clip}((\mathcal{RD}x)[t-1],0,B)\\ \mathrm{clip}(-(\mathcal{RD}x)[t-1],0,B)\end{pmatrix},\end{aligned}$$

which establishes (5.59).

Hence, $h_4[t]$ is given by the following recursion with $h_4[-1]=0$.

$$\begin{aligned}h_4[t+1] &= \rho\left(\mathbb{I}_2 h_2[t]-\mathbb{I}_2 h_3[t]+\mathbb{I}_2 h_4[t]-B\mathbf{1}_2\widehat{A}_o h_5[t]\right)\\ &\overset{(5.59),(5.56)}{=} \rho\left(\begin{pmatrix}\mathrm{clip}((\mathcal{RD}x)[t-1],0,B)\\ \mathrm{clip}(-(\mathcal{RD}x)[t-1],0,B)\end{pmatrix}\widehat{\delta}[t+1]+\mathbb{I}_2 h_4[t]-B\mathbf{1}_2\widehat{\delta}[t+2]\right)\end{aligned}$$

Note, that $h_4[t]=0$, for $t\in\{0,1,2,3\}$. Next, we show by nested induction that

$$h_4[2^k-1]=0, \qquad \text{for } k\in\mathbb{N},\ k\geq 2, \tag{5.60}$$

$$\text{and}\quad h_4[2^k+\ell]=\begin{pmatrix}\mathrm{clip}((\mathcal{RD}x)[2^k-2],0,B)\\ \mathrm{clip}(-(\mathcal{RD}x)[2^k-2],0,B)\end{pmatrix} \qquad \text{for } \ell\in\{0,\dots,2^k-2\}. \tag{5.61}$$

We have already established (5.60) for $k=2$. Next, assume (5.60) holds for some $k\geq 2$ and compute

$$\begin{aligned}h_4[2^k] &= \rho\left(\begin{pmatrix}\mathrm{clip}((\mathcal{RD}x)[2^k-2],0,B)\\ \mathrm{clip}(-(\mathcal{RD}x)[2^k-2],0,B)\end{pmatrix}\widehat{\delta}[2^k]+\mathbb{I}_{2d'}h_4[2^k-1]-D'\mathbf{1}_{2d'}\widehat{\delta}[2^k+1]\right)\\ &= \begin{pmatrix}\mathrm{clip}((\mathcal{RD}x)[2^k-2],0,B)\\ \mathrm{clip}(-(\mathcal{RD}x)[2^k-2],0,B)\end{pmatrix}\end{aligned}$$

This establishes that (5.61) holds for $\ell=0$. Next, assume (5.61) holds for some $l\in\{0,\dots,2^k-3\}$ and compute

$$\begin{aligned}h_4[2^k+\ell+1] &= \rho\Bigg(\begin{pmatrix}\mathrm{clip}((\mathcal{RD}x)[2^k+\ell-1],0,B)\\ \mathrm{clip}(-(\mathcal{RD}x)[2^k+\ell-1],0,B)\end{pmatrix}\widehat{\delta}[2^k+\ell+1]\\ &\qquad +\mathbb{I}_{2d'}h_4[2^k+\ell]-B\mathbf{1}_2\widehat{\delta}[2^k+\ell+2]\Bigg)\\ &= \rho(\mathbb{I}_{2d'}h_4[2^k+\ell])=\begin{pmatrix}\mathrm{clip}((\mathcal{RD}x)[2^k-2],0,B)\\ \mathrm{clip}(-(\mathcal{RD}x)[2^k-2],0,B)\end{pmatrix},\end{aligned}$$

which completes the induction over ℓ and establishes (5.61). Hence, we have

$$h_4[2^{k+1}-1] = \rho\left(\begin{pmatrix}\mathrm{clip}((\mathcal{R}\mathcal{D}x)[2^{k+1}-3],0,B)\\ \mathrm{clip}(-(\mathcal{R}\mathcal{D}x)[2^{k+1}-3],0,B)\end{pmatrix}\widehat{\delta}[2^{k+1}-1] + \mathbb{I}_{2d'}h_4[2^{k+1}-2] - B\mathbf{1}_2\widehat{\delta}[2^{k+1}]\right)$$

$$\begin{aligned}&= \rho\left(\mathbb{I}_{2d'}h_4[2^k+2^k-2] - B\mathbf{1}_2\right)\\ &\overset{(5.61)}{=} \rho\left(\begin{pmatrix}\mathrm{clip}((\mathcal{R}\mathcal{D}x)[2^k-2],0,B)\\ \mathrm{clip}(-(\mathcal{R}\mathcal{D}x)[2^k-2],0,B)\end{pmatrix} - B\mathbf{1}_2\right) = 0.\end{aligned}$$

This shows (5.60) for $k+1$ and thus completes the induction. The network output is given by

$$\begin{aligned}(\mathcal{R}'\mathcal{D}x)[t] &= \begin{pmatrix}1 & -1\end{pmatrix}h_2[t] + \begin{pmatrix}-1 & 1\end{pmatrix}h_3[t] + \begin{pmatrix}1 & -1\end{pmatrix}h_4[t]\\ &= \begin{pmatrix}1 & -1\end{pmatrix}(\mathbb{I}_2h_2[t] - \mathbb{I}_2h_3[t] + \mathbb{I}_2h_4[t])\\ &\overset{(5.59)}{=} \begin{pmatrix}1 & -1\end{pmatrix}\left(\begin{pmatrix}\mathrm{clip}((\mathcal{R}\mathcal{D}x)[t-1],0,B)\\ \mathrm{clip}(-(\mathcal{R}\mathcal{D}x)[t-1],0,B)\end{pmatrix}\widehat{\delta}[t+1] + \mathbb{I}_2h_4[t]\right)\end{aligned}$$

Using (5.60), (5.61) and the fact that $\widehat{\delta}[t+1] = 1$ only for $t = 2^k - 1$ with $k \geq 2$ we get

$$\begin{aligned}(\mathcal{R}'\mathcal{D}x)[2^k-1+\ell] &= \begin{pmatrix}1 & -1\end{pmatrix}\begin{pmatrix}\mathrm{clip}((\mathcal{R}\mathcal{D}x)[2^k-2],0,B)\\ \mathrm{clip}(-(\mathcal{R}\mathcal{D}x)[2^k-2],0,B)\end{pmatrix}\\ &= \mathrm{clip}((\mathcal{R}\mathcal{D}x)[2^k-2],-B,B), \qquad \forall \ell \in \{0,\dots,2^k-1\}\end{aligned}$$

as desired. □

To describe the RNN behavior for all t more concisely, the following result is useful.

Lemma 5.5.2. *For $t \in \mathbb{N}$, we define $\widetilde{k}(t) \coloneqq \lfloor \log(t+1) \rfloor$ and $\widetilde{\ell}(t) \coloneqq t + 1 - 2^{\widetilde{k}}$. We have*

$$t = 2^{\widetilde{k}(t)} - 1 + \widetilde{\ell}(t), \quad \widetilde{k}(t) \in \mathbb{N}, \quad \textit{and} \quad \widetilde{\ell}(t) \in \{0,\dots,2^{\widetilde{k}(t)}-1\}.$$

Proof. Fix $t \in \mathbb{N}$ arbitrarily and let $\widetilde{k} \coloneqq \widetilde{k}(t)$ and $\widetilde{\ell} \coloneqq \widetilde{\ell}(t)$. We have

$$2^{\widetilde{k}} - 1 + \widetilde{\ell} = 2^{\widetilde{k}} - 1 + t + 1 - 2^{\widetilde{k}} = t.$$

Furthermore, we have

$$\widetilde{\ell} = t + 1 - 2^{\lfloor \log(t+1) \rfloor} \geq t + 1 - 2^{\log(t+1)} = t + 1 - t + 1 = 0$$

as well as

$$\begin{aligned}\widetilde{\ell} &= t + 1 - 2^{\lfloor \log(t+1) \rfloor} \\ &= 2 \cdot 2^{\log(t+1)-1} - 2^{\lfloor \log(t+1) \rfloor} \\ &< 2 \cdot 2^{\lfloor \log(t+1) \rfloor} - 2^{\lfloor \log(t+1) \rfloor} \\ &= 2^{\lfloor \log(t+1) \rfloor} = 2^{\widetilde{k}}.\end{aligned}$$

□

We are now ready to combine all the building blocks in our main theorem.

Theorem 5.5.3. *Let $N \in \mathbb{N}$, $D \geq 1$, $a_0, \dots, a_N \in \mathbb{R}$, there is an RNN $\mathcal{R}_a^\pi$ such that for all $x \in [-D, D]$ and all $t \geq 16 \log N$ we have,*

$$\left| (\mathcal{R}_a^\pi \mathcal{D}x)[t] - \sum_{i=0}^{N} a_i x^i \right| \leq \|a\|_1 C_1 4^{-C_2 t},$$

where $C_1 = 16ND^{2N} >> 1$ and $C_2 = \frac{1}{4\lceil \log N \rceil} \in (0,1)$. Furthermore, the size of $\mathcal{R}_a^\pi$ is bounded by $\mathcal{M}_{\text{hid}}(\mathcal{R}_a^\pi) \leq 60N + 11$.

Proof. We may assume without loss of generality that $N \geq 2$, since otherwise the polynomial can be implemented exactly by the identity RNN from Lemma 5.4.5. Next, let $L \coloneqq \lceil \log(N) \rceil \in \mathbb{N}$, and define additional coefficients $a_{N+1} = 0, \dots, a_{2^L} = 0$. We now modify the RNN $\mathcal{R}^\pi$ from Theorem 5.4.13 using Lemma 5.2.7 together with $A = \left(a_1, \dots, a_{2^L}\right)$, $b = a_0$ to obtain the RNN $\widetilde{\mathcal{R}_a^\pi}$, realizing

$$(\widetilde{\mathcal{R}_a^\pi} \mathcal{D}x)[t] = a_0 + \sum_{i=1}^{2^L} a_i \left((\mathcal{R}^\pi \mathcal{D}x)[t] \right)_i.$$

We can now bound, for $k \geq \lceil \log L \rceil + 2$,

$$\begin{aligned}\left| (\widetilde{\mathcal{R}_a^\pi} \mathcal{D}x)[2^k - 2] - \sum_{i=0}^{N} a_i x^i \right| &\leq \sum_{i=2}^{2^L} |a_i| \left| \left((\mathcal{R}^\pi \mathcal{D}x)[2^k - 2] \right)_i - x^i \right| \\ &= \sum_{\ell=1}^{L} \sum_{j=1}^{2^{\ell-1}} |a_{2^{\ell-1}+j}| \left| \left((\mathcal{R}^\pi \mathcal{D}x)[2^k - 2] \right)_{2^{\ell-1}+j} - x^{2^{\ell-1}+j} \right| \\ &\overset{\text{Thm 5.4.13}}{\leq} \sum_{\ell=1}^{L} \epsilon_{\ell,k} \sum_{j=1}^{2^{\ell-1}} |a_{2^{\ell-1}+j}| \\ &\leq \epsilon_{L,k} \sum_{\ell=1}^{L} \sum_{j=1}^{2^{\ell-1}} |a_{2^{\ell-1}+j}| \\ &\leq \|a\|_1 8 \cdot 2^L D^{2^L} 4^{-\frac{1}{2L} 2^k}.\end{aligned} \tag{5.62}$$

Next, set $B = \max_{x \in [-D,D]} \left| \sum_{i=0}^{N} a_i x^i \right|$ and use Lemma 5.5.1 with B and $\widetilde{\mathcal{R}_a^\pi}$ to obtain an RNN

$\mathcal{R}_a^\pi$ such that, for $k \geq 2$,

$$(\mathcal{R}_a^\pi \mathcal{D}x)[2^k - 1 + \ell] = \mathrm{clip}((\widetilde{\mathcal{R}_a^\pi}\mathcal{D}x)[2^k - 2], -B, B) \qquad \forall \ell \in \{0, \dots, 2^k - 1\}. \tag{5.63}$$

Putting everything together we compute, for $t \geq 8 \log N$,

$$\begin{aligned}
\left|(\mathcal{R}_a^\pi \mathcal{D}x)[t] - \sum_{i=0}^{N} a_i x^i\right| &\overset{\text{Lem. 5.5.2}}{=} \left|(\mathcal{R}_a^\pi \mathcal{D}x)[2^{\widetilde{k}(t)} - 1 + \widetilde{\ell}(t)] - \sum_{i=0}^{N} a_i x^i\right| \\
&\overset{(5.63)}{=} \left|\mathrm{clip}((\widetilde{\mathcal{R}_a^\pi}\mathcal{D}x)[2^{\widetilde{k}(t)} - 2], -B, B) - \sum_{i=0}^{N} a_i x^i\right| \\
&\leq \left|(\widetilde{\mathcal{R}_a^\pi}\mathcal{D}x)[2^{\widetilde{k}(t)} - 2] - \sum_{i=0}^{N} a_i x^i\right| \\
&\overset{(*)}{\leq} \|a\|_1 8 \cdot 2^L D^{2^L} 4^{-\frac{1}{2L} 2^{\widetilde{k}(t)}}
\end{aligned} \tag{5.64}$$

where in $(*)$ we use (5.62) upon noting that for $t \geq 16 \log N$, we have

$$\begin{aligned}
\widetilde{k}(t) &= \lfloor \log(t+1) \rfloor \geq \lfloor \log(16 \log N) \rfloor = \lfloor \log(2 \log N) \rfloor + 3 \\
&\geq \lceil \log(\log N + \log N) \rceil + 2 \geq \lceil \log(\lceil \log N \rceil) \rceil + 2 = \lceil \log(L) \rceil + 2.
\end{aligned}$$

We further upper bound (5.64) as follows

$$\begin{aligned}
\left|(\mathcal{R}_a^\pi \mathcal{D}x)[t] - \sum_{i=0}^{N} a_i x^i\right| &\leq \|a\|_1 8 \cdot 2^L D^{2^L} 4^{-\frac{1}{2L} 2^{\widetilde{k}(t)}} \\
&\leq \|a\|_1 8 \cdot 2^{\lceil \log N \rceil} D^{2^{\lceil \log N \rceil}} 4^{-\frac{1}{2\lceil \log N \rceil} 2^{\widetilde{k}(t)}} \\
&\leq \|a\|_1 8 \cdot 2N D^{2N} 4^{-\frac{1}{2\lceil \log N \rceil} 2^{\widetilde{k}(t)}} \\
&= \|a\|_1 16 N D^{2N} 4^{-\frac{1}{2\lceil \log N \rceil} 2^{\lfloor \log(t+1) \rfloor}} \\
&\leq \|a\|_1 16 N D^{2N} 4^{-\frac{1}{2\lceil \log N \rceil} 2^{\log(t+1) - 1}} \\
&\leq \|a\|_1 16 N D^{2N} 4^{-\frac{t}{4\lceil \log N \rceil}}.
\end{aligned}$$

The size is given by

$$\mathcal{M}_{\mathrm{hid}}(\mathcal{R}_a^\pi) \overset{\text{Lem. 5.5.1}}{=} \mathcal{M}_{\mathrm{hid}}(\widetilde{\mathcal{R}_a^\pi}) + 11 = \mathcal{M}_{\mathrm{hid}}(\mathcal{R}^\pi) + 11 \overset{\text{Lem. 5.4.9}}{\leq} 60N + 11. \qquad \square$$

References

[Alv+14] GA Alves et al. "Superdiffusion driven by exponentially decaying memory". In: *Journal of Statistical Mechanics: Theory and Experiment* 2014.4 (2014), P04026.

[AS93a] F. Albertini and E. Sontag. "State observability in recurrent neural networks". In: *Proceedings of 32nd IEEE Conference on Decision and Control*. Vol. 4. 1993, pp. 3706–3707. DOI: 10.1109/CDC.1993.325908.

[AS93b] Francesca Albertini and Eduardo D. Sontag. "For neural networks, function determines form". In: *Neural Networks* 6.7 (1993), pp. 975–990. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(09)80007-5.

[ASM93] Francesca Albertini, Eduardo D. Sontag, and Vincent Maillot. "Uniqueness Of Weights For Neural Networks". In: *Artificial Neural Networks with Applications in Speech and Vision*. Chapman and Hall, 1993, pp. 115–125.

[Bak+23] Ainesh Bakshi, Allen Liu, Ankur Moitra, and Morris Yau. "A new approach to learning linear dynamical systems". In: *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*. 2023, pp. 335–348.

[Bar93] A. R. Barron. "Universal approximation bounds for superpositions of a sigmoidal function". In: *IEEE Transactions on Information Theory* 39.3 (1993), pp. 930–945. ISSN: 1557-9654. DOI: 10.1109/18.256500.

[Bar94] A. R. Barron. "Approximation and estimation bounds for artificial neural networks". en. In: *Machine Learning* 14.1 (1994), pp. 115–133. ISSN: 1573-0565. DOI: 10.1007/BF00993164.

[Bar95] Robert Gardner Bartle. *The elements of integration and Lebesgue measure*. en. New York: Wiley, 1995. ISBN: 978-0-471-04222-8.

[BC85] Stephen Boyd and Leon Chua. "Fading memory and the problem of approximating nonlinear operators with Volterra series". In: *IEEE Transactions on Circuits and Systems* 32.11 (1985), pp. 1150–1161.

[BGJ20] Julius Berner, Philipp Grohs, and Arnulf Jentzen. "Analysis of the Generalization Error: Empirical Risk Minimization over Deep Artificial Neural Networks Overcomes the Curse of Dimensionality in the Numerical Approximation of Black–Scholes Partial Differential Equations". In: *SIAM Journal on Mathematics of Data Science* 2.3 (2020), pp. 631–657. DOI: 10.1137/19M125649X.

[BGS07] Byron Boots, Geoffrey J Gordon, and Sajid Siddiqi. "A constraint generation approach to learning stable linear dynamical systems". In: *Advances in Neural Information Processing Systems* 20 (2007).

[Böl+19] Helmut Bölcskei, Philipp Grohs, Gitta Kutyniok, and Philipp Petersen. "Optimal Approximation with Sparsely Connected Deep Neural Networks". In: *SIAM Journal on Mathematics of Data Science* 1.1 (2019), pp. 8–45. DOI: 10.1137/18m118709x.

[CC95] Tianping Chen and Hong Chen. "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems". In: *IEEE Transactions on Neural Networks* 6.4 (1995), pp. 911–917. DOI: 10.1109/72.392253.

[Cyb89] G Cybenko. "Approximation by superpositions of a sigmoidal function". In: *Mathematics of Control, Signals and Systems* 2.4 (1989), pp. 303–314. ISSN: 1435-568X. DOI: 10.1007/BF02551274.

[Dau+23] Ingrid Daubechies et al. "Neural Network Approximation of Refinable Functions". In: *IEEE Transactions on Information Theory* 69.1 (2023), pp. 482–495. ISSN: 0018-9448, 1557-9654. DOI: 10.1109/TIT.2022.3199601.

[Day13] William A Day. *The Thermodynamics of Simple Materials with Fading Memory*. Vol. 22. Springer Science & Business Media, 2013.

[Dil75] Ellis Harold Dill. "Simple materials with fading memory". In: *Academic Press, New York* (1975).

[Don+98] D. Donoho, M. Vetterli, R.A. DeVore, and I. Daubechies. "Data compression and harmonic analysis". In: *IEEE Transactions on Information Theory* 44.6 (1998), pp. 2435–2476. ISSN: 00189448. DOI: 10.1109/18.720544.

[Don01] D. Donoho. "Sparse components of images and optimal atomic decompositions". In: *Constructive Approximation* 17.3 (2001), pp. 353–382. ISSN: 0176-4276. DOI: 10.1007/s003650010032.

[Don96] D. Donoho. "Unconditional bases and bit-level compression". In: *Applied and Computational Harmonic Analysis* 3.4 (1996), pp. 388–392. ISSN: 10635203. DOI: 10.1006/acha.1996.0032.

[Elb+21] Dennis Elbrächter, Dmytro Perekrestenko, Philipp Grohs, and Helmut Bölcskei. "Deep Neural Network Approximation Theory". In: *IEEE Transactions on Information Theory* 67.5 (2021), pp. 2581–2623. DOI: 10.1109/TIT.2021.3062161.

[Elm90] Jeffrey L Elman. "Finding structure in time". In: *Cognitive Science* 14.2 (1990), pp. 179–211. DOI: 10.1207/s15516709cog1402_1.

[Fef83] Charles L Fefferman. "The uncertainty principle". In: *Bulletin (New Series) of the American Mathematical Society* 9.2 (1983), pp. 129–206. DOI: 10.1090/S0273-0979-1983-15154-6.

[Fef94] Charles Fefferman. "Reconstructing a neural net from its output". In: *Revista Matemática Iberoamericana* 10.3 (1994), pp. 507–555. DOI: 10.4171/RMI/160.

[FN93] Ken-Ichi Funahashi and Yuichi Nakamura. "Approximation of dynamical systems by continuous time recurrent neural networks". In: *Neural Networks* 6.6 (1993), pp. 801–806. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(05)80125-X.

[FP02] Mauro Fabrizio and Sergio Polidoro. "Asymptotic decay for some differential systems with fading memory". In: *Applicable Analysis* 81.6 (2002), pp. 1245–1264.

[Fun89] Ken-Ichi Funahashi. "On the approximate realization of continuous mappings by neural networks". In: *Neural Networks* 2.3 (1989), pp. 183–192. ISSN: 0893-6080. DOI: 10.1016/0893-6080(89)90003-8.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL: www.deeplearningbook.org.

[GHL95] C Lee Giles, Bill G Horne, and Tsungnan Lin. "Learning a class of large finite state machines with a recurrent neural network". In: *Neural Networks* 8.9 (1995), pp. 1359–1365.

[GO18] Lyudmila Grigoryeva and Juan-Pablo Ortega. "Echo state networks are universal". In: *Neural Networks* 108 (2018), pp. 495–508.

[Gra+06] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. "Connectionist temporal classification: Labeling unsegmented sequence data with recurrent neural networks". In: *ICML '06: Proceedings of the 23rd International Conference on Machine Learning*. 2006, pp. 369–376. ISBN: 1595933832. DOI: 10.1145/1143844.1143891.

[Gra+09] A Graves et al. "A novel connectionist system for unconstrained handwriting recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 31.5 (2009), pp. 855–868. DOI: 10.1109/TPAMI.2008.137.

[Gro+18] Philipp Grohs, Fabian Hornung, Arnulf Jentzen, and Philippe von Wurstemberger. "A proof that artificial neural networks overcome the curse of dimensionality in the numerical approximation of Black-Scholes partial differential equations". In: *Mem. Amer. Math. Soc.* (2018), to appear. arXiv: 1809.02362 [math.NA].

[Grö01] Karlheinz Gröchenig. *Foundations of Time-Frequency Analysis*. Birkhäuser Boston, 2001. DOI: 10.1007/978-1-4612-0003-1.

[Gro15] Philipp Grohs. "Optimally sparse data representations". In: *Harmonic and Applied Analysis: From Groups to Signals* (2015), pp. 199–248.

[HAB22] Clemens Hutter, Thomas Allard, and Helmut Bölcskei. *Metric entropy of causal, discrete-time LTI systems*. 2022. arXiv: 2211.15466.

[HGB22] Clemens Hutter, Recep Gül, and Helmut Bölcskei. "Metric entropy limits on recurrent neural network learning of linear dynamical systems". In: *Applied and Computational Harmonic Analysis* 59 (2022), pp. 198–223.

[HMR18] Moritz Hardt, Tengyu Ma, and Benjamin Recht. "Gradient descent learns linear dynamical systems". In: *Journal of Machine Learning Research* 19.29 (2018), pp. 1–44. URL: http://jmlr.org/papers/v19/16-465.html.

[HRS21] Christiaan Heij, André C M Ran, and Frederik van Schagen. *Introduction to Mathematical Systems Theory*. Springer International Publishing, 2021. DOI: 10.1007/978-3-030-59654-5.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5 (1989), pp. 359–366. ISSN: 0893-6080. DOI: 10.1016/0893-6080(89)90020-8.

[HSZ17] Elad Hazan, Karan Singh, and Cyril Zhang. "Learning linear dynamical systems via spectral filtering". In: *Advances in Neural Information Processing Systems* 30 (2017).

[Huy25] Daan Huybrechs. *On the Algorithmic Construction of Deep ReLU Networks*. 2025. arXiv: 2506.19104.

[Kai80] Thomas Kailath. *Linear Systems*. Englewood Cliffs, N.J: Prentice-Hall, 1980. ISBN: 978-0135369616.

[KT93] A. Kolmogorov and V. Tikhomirov. "ϵ-Entropy and ϵ-Capacity of Sets In Functional Spaces". In: *Selected Works of A. N. Kolmogorov — Volume III: Information Theory and the Theory of Algorithms*. Ed. by A. N. Shiryayev. Dordrecht: Springer Netherlands, 1993, pp. 86–170. ISBN: 978-94-017-2973-4. DOI: 10.1007/978-94-017-2973-4_7.

[KWI24] Santhosh Karnik, Rongrong Wang, and Mark Iwen. "Neural network approximation of continuous functions in high dimensions with applications to inverse problems". In: *Journal of Computational and Applied Mathematics* 438 (2024), p. 115557.

[Lak95] Vangipuram Lakshmikantham. *Theory of Integro-differential Equations*. Vol. 1. CRC press, 1995.

[Lap17] Amos Lapidoth. *A Foundation in Digital Communication*. 2nd ed. Cambridge: Cambridge University Press, 2017. ISBN: 978-1-107-17732-1. DOI: 10.1017/9781316822708.

[Li+21] Zhong Li, Jiequn Han, Weinan E, and Qianxiao Li. "On the Curse of Memory in Recurrent Neural Networks: Approximation and Optimization Analysis". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=8Sqhl-nF50.

[LMK22] Samuel Lanthaler, Siddhartha Mishra, and George E Karniadakis. "Error estimates for DeepONets: a deep learning framework in infinite dimensions". In: *Transactions of Mathematics and Its Applications* 6.1 (2022). ISSN: 2398-4945. DOI: 10.1093/imatrm/tnac001.

[Lu+21] Lu Lu et al. "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators". In: *Nature Machine Intelligence* 3.3 (2021), pp. 218–229. ISSN: 2522-5839. DOI: 10.1038/s42256-021-00302-5.

[Mat93] Michael B Matthews. "Approximating nonlinear fading-memory operators using neural network models". In: *Circuits, Systems and Signal Processing* 12.2 (1993), pp. 279–307. ISSN: 1531-5878. DOI: 10.1007/BF01189878.

[MBH13] G Matz, H Bölcskei, and F Hlawatsch. "Time-frequency foundations of communications: Concepts and tools". In: *IEEE Signal Processing Magazine* 30.6 (2013), pp. 87–96. DOI: 10.1109/MSP.2013.2269702.

[McC03] John E McCarthy. "Pick's Theorem-What's the big deal?" In: *The American Mathematical Monthly* 110.1 (2003), pp. 36–45. ISSN: 00029890, 19300972. DOI: 10.2307/3072342.

[MM09] Edoardo Mainini and Gianluca Mola. "Exponential and polynomial decay for first order linear Volterra evolution equations". In: *Quarterly of Applied Mathematics* 67.1 (2009), pp. 93–111.

[Mou+16] Thiago RS Moura et al. "Transient superdiffusion in random walks with a q-exponentially decaying memory profile". In: *Physica A: Statistical Mechanics and its Applications* 453 (2016), pp. 259–263.

[NX22] Stefan Nagel and Zhengyang Xu. "Asset pricing with fading memory". In: *The Review of Financial Studies* 35.5 (2022), pp. 2190–2245.

[OSB99] Alan V. Oppenheim, Ronald W. Schafer, and John R. Buck. *Discrete-Time Signal Processing*. 2nd. USA: Prentice-Hall, Inc., 1999. ISBN: 0137549202.

[Pas+14] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. "How to construct deep recurrent neural networks". In: *Proceedings of the Second International Conference on Learning Representations (ICLR 2014)*. 2014.

[PHB25] Yang Pan, Clemens Hutter, and Helmut Bölcskei. *Metric-Entropy Limits on the Approximation of Nonlinear Dynamical Systems*. 2025. arXiv: 2407.01250.

[Ras21] Mones Raslan. "Solving parametric PDEs with neural networks: Unfavorable structure vs. expressive power". PhD thesis. TU Berlin, 2021.

[Rud87] Walter Rudin. *Real and Complex Analysis*. 3rd. McGraw-Hill, 1987. ISBN: 978-0070542341.

[San94] Irwin W Sandberg. "Approximately finite memory and the circle criterion". In: *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications* 41.7 (1994), pp. 473–476.

[Sch+20] Julian Schrittwieser et al. "Mastering Atari, Go, chess and shogi by planning with a learned model". In: *Nature* 588.7839 (2020), pp. 604–609. ISSN: 1476-4687. DOI: 10.1038/s41586-020-03051-4.

[Son92] Eduardo D Sontag. "Neural nets as systems models and controllers". In: (1992). URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.55.8164.

[Son98] Eduardo Sontag. "Recurrent Neural Networks: Some Systems-Theoretic Aspects". In: *Dealing with Complexity*. Ed. by Mirek Kárný, Kevin Warwick, and Vera Kůrková. Springer London, 1998. Chap. 1. DOI: 10.1007/978-1-4471-1523-6.

[SS95] H T Siegelmann and E D Sontag. "On the computational power of neural nets". In: *Journal of Computer and System Sciences* 50.1 (1995), pp. 132–150. ISSN: 0022-0000. DOI: 10.1006/jcss.1995.1013.

[SVL14] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. "Sequence to sequence learning with neural networks". In: *Advances in Neural Information Processing Systems*. Vol. 27. 2014. URL: https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

[SZ07] Anton Maximilian Schäfer and Hans Georg Zimmermann. "Recurrent neural networks are universal approximators". In: *International Journal of Neural Systems* 17.04 (2007), pp. 253–263. DOI: 10.1142/S0129065707001111.

[Tur+19] Javier S. Turek et al. "Approximating Stacked and Bidirectional Recurrent Architectures with the Delayed Recurrent Neural Network". In: *37th International Conference on Machine Learning, ICML 2020* PartF16814.August (2019), pp. 9590–9600. arXiv: 1909.00021. URL: http://arxiv.org/abs/1909.00021.

[VB21a] Verner Vlačić and Helmut Bölcskei. "Affine symmetries and neural network identifiability". In: *Advances in Mathematics* 376 (2021). ISSN: 0001-8708. DOI: 10.1016/j.aim.2020.107485.

[VB21b] Verner Vlačić and Helmut Bölcskei. "Neural network identifiability for a family of sigmoidal nonlinearities". In: *Constructive Approximation* (2021). ISSN: 1432-0940. DOI: 10.1007/s00365-021-09544-3.

[Vol59] Vito Volterra. *Theory of Functionals and of Integral and Integro-Differential Equations*. Dover Books on Mathematics. New York: Dover Publications, 1959.

[Wai19] Martin J. Wainwright. *High-Dimensional Statistics*. Cambridge University Press, 2019. DOI: 10.1017/9781108627771.

[Wan+18] Zhong Yi Wan, Pantelis Vlachas, Petros Koumoutsakos, and Themistoklis Sapsis. "Data-assisted reduced-order modeling of extreme events in complex dynamical systems". In: *PLOS ONE* 13.5 (2018). Ed. by Daniel Durstewitz, e0197704. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0197704.

[Wan65] C -C Wang. "The principle of fading memory". In: *Archive for Rational Mechanics and Analysis* 18 (1965), pp. 343–366.

[Wie66] Norbert Wiener. *Nonlinear problems in random theory*. Cambridge, Massachusetts, USA: The MIT Press, 1966.

[WJ20] Andrew Wagenmaker and Kevin Jamieson. "Active learning for identification of linear dynamical systems". In: *Conference on Learning Theory*. PMLR. 2020, pp. 3487–3582.

[Yar17] Dmitry Yarotsky. "Error Bounds for Approximations with Deep ReLU Networks". In: *Neural Networks* 94 (Oct. 1, 2017), pp. 103–114. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2017.07.002.

[Yar18] Dmitry Yarotsky. "Optimal approximation of continuous functions by very deep ReLU networks". In: *Conference on Learning Theory*. PMLR. 2018, pp. 639–649.

[Zam79] G. Zames. "On the metric complexity of causal linear systems: ϵ-entropy and ϵ-dimension for continuous time". In: *IEEE Transactions on Automatic Control* 24.2 (1979), pp. 222–230. ISSN: 15582523. DOI: 10.1109/TAC.1979.1101976.

[ZO93] G Zames and J G Owen. "A note on metric dimension and feedback in discrete time". In: *IEEE Transactions on Automatic Control* 38.4 (1993), pp. 664–667. ISSN: 1558-2523. DOI: 10.1109/9.250545.