Cascade Networks, Generalized Neural Networks, and Approximation of Functions

By

Diana C. Sordillo

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Mathematics

August 12, 2022

Nashville, Tennessee

Approved:

Marian Neamtu, Ph.D.

Alexander Powell, Ph.D.

Doug Hardin, Ph.D.

Akram Aldroubi, Ph.D.

Don M. Wilkes, Ph.D.

ACKNOWLEDGEMENTS

I would first like thank my advisor, Professor Marian Neamtu, for his guidance. I greatly appreciate all the time he took to oversee this thesis. It would not have been possible without his invaluable assistance and knowledge.

I thank my committee members, Professors Akram Aldroubi, Alexander Powell, Doug Hardin, and Don Wilkes for spending time to review my dissertation and to serve on my committee.

I am also thankful for the close friendships I have made in the math department. In particular, I thank Frank Wagner for his constant support and treasured help.

Finally, I am most grateful for my family, especially my parents Peter and Ellen Sordillo, my sister Laura Sordillo, and my brother Vincent Sordillo, for their endless love and encouragement.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

Chapter 1

Introduction

1.1 Overview

A *neural network* is a supervised machine learning model based on how the brain acquires and stores knowledge [1, 2]. Supervised machine learning problems use sample data to make predictions about the labels of patterns. A neural network receives sets of pairs $\{x_i, y_i\}_{i=1}^n$, where $x_i \in \mathbb{R}^d$ are the inputs of the network and $y_i \in \mathbb{R}^m$ are the corresponding vector of outputs for $i = 1, \ldots, n$. The given set of input-output pairs is referred to as training data. It is assumed that x_i and y_i are related by some unknown function; however, in general the neural network is not able to compute this function. Instead, a candidate is chosen from a parameterized set of functions, using the training data to help select parameters. After this "training" (or "learning") on a large enough sequence of pairs, the neural network approximately realizes an interpolation of the training data, and generalizes, as closely as it can, new data points. [2, 3].

Neural networks have been around for more than 70 years, beginning with the work of Mcculloch and Pitts in 1943, Hebb in 1949, and the perceptron model of Rosenblatt in 1958 [4, 5, 6]. Due to the availability of large amounts of training data and improvements in computing power, neural networks are increasingly used in a wide range of machine learning problems. In particular, deep neural networks with non-linear activation function have had remarkable success when applied to computer science and engineering questions related to image recognition, speech recognition, and other classification problems [7, 8, 9, 10].

While neural networks have produced an abundance of successes in practical applications, the basis of these successes lacks rigorous mathematical analysis [11]. Even in basic settings, there is no theory that fully quantifies the approximation power of neural networks. Hence understanding the approximation properties of neural networks could lead to significant practical improvements

[7, 12]. Specifically, an important problem is to understand what are the benefits of using neural networks over traditional methods of approximation such as polynomials, wavelets, splines etc. Moreover, it is crucial to learn in what way neural networks are more effective approximation tools than other methods.

In its most general form, a neural network is a function given by repeatedly applying a fixed function, in general non-linear, to an affine operator. Thus, it is natural to wonder whether is it possible to use well-established tools in a framework that mimics the neural network framework, but where it is possible to understand better the approximation properties.

In this thesis, close analogs to neural networks using the rectified linear unit (ReLU) activation function are introduced. These analogs, called *cascade networks*, are also functions given by repeatedly applying a fixed function to an affine operator. Cascade networks have a close connection with algorithms used in computer aided geometric design and multiresolution analysis. In particular, the connection between cascade networks, subdivision algorithms, and the cascade algorithm is discussed. The space of functions obtained by a cascade network with fixed width is characterized. Cascade networks are compared, in the univariate case,in terms of approximation power with ReLU networks based on known results for ReLU networks. Using cascade networks to approximate polynomials and smooth functions, similar results were obtained when compared to the results for ReLU neural networks.

1.2 Notation

The following notation is used throughout the dissertation.

- Denote by $\mathbb{N} = \{1, 2, \dots\}$ the set of natural numbers, and by $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$. Let $\mathbb{R}$ denote the set of real numbers, $\mathbb{R}_+$ the set all positive real numbers, and $\mathbb{R}_-$ the set all negative real numbers. Let $\mathbb{C}$ denote the set of complex numbers, $\mathbb{Z}$ the set of integers, $\mathbb{Q}$ the set of rational numbers.

- For each $d \in \mathbb{N}$, let $\mathbb{R}^d := \{x = (x_1, \dots, x_d) |\ x_j \in \mathbb{R},\ j = 1, 2, \dots, d\}$. d is called the dimension of $\mathbb{R}^d$ and the numbers x_j are called the components of x.

- Let $\mathbb{R}^{n\times m}$ denote the set of all $n \times m$ matrices with real entries. Let $\mathbb{I}_n$ denote the $n \times n$ identity matrix and $\mathbf{0}$ denote the $n \times m$ matrix with all zero entries. Write $A^T \in \mathbb{R}^{m\times n}$ for the transpose of a matrix $A \in \mathbb{R}^{n\times m}$.

- For a function $f : \mathbb{R}^d \to \mathbb{R}$, define $\|f\|_{L_\infty(\Omega)} = \inf\{C \geq 0 : |f(x)| \leq C, \text{ for a.e } x \in \Omega\}$.

- Let $\Pi_n = \{a_n x^n + \cdots + a_1 x + a_0 \mid a_i \in \mathbb{R} \text{ for } i = 0, \ldots, n\}$ denote the space of real valued polynomials of degree at most n.

- Throughout, "$\log$" stands for the logarithm to base 2 and "$\ln$" stands for the logarithm to base e.

- The notation $g(n, \varepsilon) = \mathcal{O}(h(n, \varepsilon))$ will always mean that $|g(n, \varepsilon)| \leq C|h(n, \varepsilon)|$, for all $\varepsilon > 0$ small enough and for all integer n, and where C is a constant independent of n, ε.

- I represents the closed interval $[0, 1]$.

- For $A \subset X$, define the characteristic function of A by $\chi_A = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$.

- The binary representation of an integer $j \in \{0, \ldots, 2^\ell - 1\}$ is written $(j)_2 = \delta_1 \cdots \delta_\ell = 2^{\ell-1}\delta_1 + \cdots + 2^0\delta_\ell$, where $\delta_i \in \{0, 1\}$, for $i = 1, \ldots, \ell$.

- For $n \geq 1$ and $x_0, \ldots, x_n$ distinct points in $\mathbb{R}$, the divided difference of a function f is defined as $[x_0, \ldots, x_n]f := \frac{\sum_{i=0}^{n} f(x_i)}{\prod_{j\neq i}(x_i, x_j)}$.

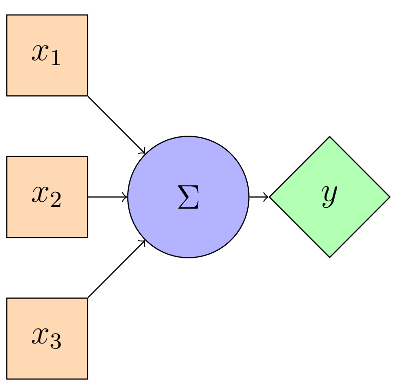
Chapter 2

ReLU Neural Networks

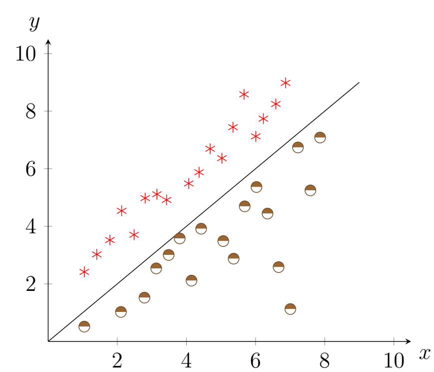
2.1 Background

The perceptron, first introduced in 1958 by Rosenblatt, is the simplest form of a neural network [1]. The perceptron computes a function $f : \mathbb{R}^d \to \{0, 1\}$ of the form $f(x) = \sigma(w \cdot x + b)$, for a given input $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$ (Figure 2.1a). Here $w = (w_1, \ldots, w_d) \in \mathbb{R}^d$, and $b \in \mathbb{R}$ are trainable parameters called *weights* and *bias* respectively, and $w \cdot x$ denotes the dot product $\sum_{i=1}^{d} w_i x_i$. The mapping σ is called an *activation function*. In Rosenblatt's perceptron the activation function was the Heaviside function,

$$\sigma(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}.$$



(a) Graph of Rosenblatt's perceptron with three inputs (squares) and one output (diamond).



(b) Decision boundary in $\mathbb{R}^2$ computed by a perceptron with parameters w, b.

Figure 2.1: Rosenblatt's Perceptron

As the output of f is either 0 or 1, the perceptron model is suitable for pattern classification problems where patterns are to be divided into two classes and whose patterns are "linearly separable," i.e which lie on opposing sides of a hyperplane. The boundary between the set of points classified as 0 and those classified as 1 is called a decision boundary and is defined by the equation $w \cdot x + b = 0$ (Figure 2.1b). The vector of weights w determines the orientation of the decision boundary, and $\frac{b}{\|w\|}$, where $\|w\| = \left(\sum_{i=1}^{d} w_i^2\right)^{1/2}$, determines the distance from the origin.

For the classification of patterns that are not linearly separable, multilayer perceptrons (MLP) were developed. An MLP is associated with a directed acyclic graph, called its architecture (Figure 2.2). In the MLP model there are a finite number of successive layers, and each layer consists of a finite number of neurons, also referred to as units or nodes. The network is called feedforward as information flows in one direction, starting from the input layer, through the intermediate layers, to a last layer where the resulting output is obtained. The intermediate layers are known as *hidden* layers and the last layer is called the *output* layer. The neurons in the hidden layers and the output layer are sometimes referred to as *computation units*. A neural network is called fully connected if each unit of each layer is connected to each unit in the subsequent layer. These connections are known as links or synapses. There are no connections between any two units in a given layer.

Networks with one hidden layer are known as *shallow* networks. Networks with more than one hidden layer are called *deep* neural networks (DNN). The function of the hidden layers is to transform the inputs in a nonlinear way so that the classification becomes linearly separable by the last layer.

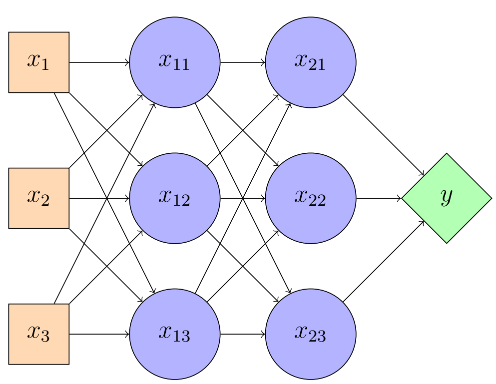


Figure 2.2: Directed Graph Representation of a Multilayer Perceptron

The rules of the MLP model are [3]:

1. Let $x_1, \ldots, x_d$ be the inputs of the model and let x_{ij} correspond to the j-th unit in the i-th layer, where $x_{0j} = x_j$.

2. The units x_{ij} are multiplied by weights w_{ijk} and these products are summed over j.

3. A bias b_{ik} and then an activation function σ are applied to the sum obtained in (2) and the resulting value represents the output $x_{i+1,k}$ of this k-th unit of the $i+1$-st layer

$$x_{i+1,k} = \sigma\Big(\sum_j w_{ijk} x_{ij} + b_{ik}\Big).$$

The choice of activation function for the non-linear layers is, in general, fixed beforehand. Currently, the rectified linear unit (ReLU), $\sigma(x) = \max\{0, x\}$, is the most widely used activation function in practical applications [7]. Other common choices for activation function for include [1, 3]:

1. the logistic sigmoid:
$$\sigma(x) = \frac{1}{1+e^{-x}},$$

2. the hyperbolic tangent function:
$$\sigma(x) = \tanh(x/2),$$

3. the parametric rectified linear unit (PReLU):
$$\sigma(x) = \begin{cases} \alpha x, & x \leq 0 \\ x, & x > 0 \end{cases}, \text{ for some parameter } \alpha,$$

4. The exponential linear unit (ELU):
$$\sigma(x) = \begin{cases} \alpha(e^x - 1), & x \leq 0 \\ x, & x > 0 \end{cases}, \text{ for some parameter } \alpha,$$

5. the sigmoid linear unit (SiLU):
$$\sigma(x) = \frac{x}{1 + e^{-x}}.$$

The process of determining the weights and biases is called learning, or training. The most common algorithm for training a neural network is by the backpropagation algorithm. [13]. Backpropagation is a gradient descent method, which computes the gradient of a loss function E with respect to the weights and biases for all layers of the network [14]. The weights and biases are initially chosen so that the network behaves well when analyzing the training data. Let w_ℓ, b_ℓ denote the weights and biases of layer ℓ. The weights and biases at layer ℓ are then updated according to the rule:

$$w_\ell = w_\ell - \eta \nabla E(w_\ell)$$
$$b_\ell = b_\ell - \eta \nabla E(b_\ell),$$

where η is an adjustable parameter, called a learning rate, that determines the step size at each iteration moving towards the minimum of E [1, 7]. There are many challenges in neural network optimization, including difficulties in finding a global minimum, a large number of local minima, and vanishing or exploding gradients [7].

Although shallow networks have been extensively studied for the last thirty years, the interest in deep networks is rather recent. This is due, in part, to the increased availability of computing power, large training databases, as well as the ability to train DNNs efficiently. DNNs are also more efficient at approximating a wide range of function spaces than shallow networks of comparable size [7].

2.2 ReLU Neural Network Definition and Properties

The following gives a mathematical formulation of neural networks as a function resulting from repeatedly composing an activation function σ with an affine linear transformation. Going forward, σ is restricted to be the ReLU activation function. σ acts componentwise, thus, $\sigma(x_1, \ldots, x_d) = (\sigma(x_1), \ldots, \sigma(x_d)) = (\max\{0, x_1\}, \ldots, \max\{0, x_d\})$, for $x = (x_1, \ldots, x_d) \in \mathbb{R}^d$.

Consider the following general neural network architecture:

Definition 2.1. [15, 16]

Let $L, N_0, \ldots, N_L \in \mathbb{N}, L \geq 2$. A *depth L ReLU neural network* is a function $\Phi : \mathbb{R}^{N_0} \to \mathbb{R}^{N_L}$ given by

$$\Phi(x) = W_L(\sigma(W_{L-1}(\sigma(\ldots \sigma(W_1(x)))))), \tag{2.1}$$

where W_ℓ are affine maps $W_\ell : \mathbb{R}^{N_{\ell-1}} \to \mathbb{R}^{N_\ell}$ given by $W_\ell(x) \coloneqq A_\ell x + b_\ell$, for weight matrices $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$, and bias vectors $b_\ell \in \mathbb{R}^{N_\ell}$, $\ell = 1, 2, \ldots, L$.

Here, N_0 is the dimension of the input layer (indexed as the 0th layer), $N_1, \ldots, N_{L-1}$ are the dimensions of the $L-1$ hidden layers, and N_L is the dimension of the output layer.

Φ is used to denote the ReLU neural network and its architecture, as well as the function it implements.

The following definitions describe the size and complexity of a ReLU neural network.

Definition 2.2. The depth of Φ is $\mathcal{L}(\Phi) = L$. Φ is called a shallow network when $L = 2$, and a deep network when $L \geq 3$

Remark 2.1. *There is no standard definition of network depth. Depth can also be defined as the input layer plus the number of hidden layers (excluding the output layer), or by the total number of layers (including the input and output layers). Neural networks can also be characterized by only the number of hidden layers in the network (without the input and output layers).*

Definition 2.3. The total number of *computation units* of Φ is $\mathcal{U}(\Phi) = \sum_{\ell=1}^{L} N_\ell$.

Definition 2.4. The *width* of the ℓ-th hidden layer of Φ is N_ℓ. The width of the Φ is $\mathcal{M}(\Phi) = \max\{N_0, N_1, \dots, N_L\}$.

Definition 2.5. The number of *weights* of Φ, denoted as $\mathcal{W}(\Phi)$, is defined to be the total number of non-zero entries of the matrices A_ℓ and the vectors b_ℓ. The matrix entry $(A_\ell)_{i,j}$ represents the weight associated with the j-th unit in the $(\ell-1)$-st layer and the i-th unit in the ℓ-th layer. $(b_\ell)_i$ represents the bias associated with the i-th unit in the ℓ-th layer. The number of weights of the network satisfies

$$\mathcal{W}(\Phi) \leq \mathcal{L}(\Phi)\mathcal{M}(\Phi)(\mathcal{M}(\Phi)+1).$$

Example 2.6. Consider a ReLU neural network, Φ, with $\mathcal{L}(\Phi) = 2, \mathcal{U}(\Phi) = 5, \mathcal{M}(\Phi) = 3$, and $\mathcal{W}(\Phi) = 12$, given by

$$\Phi(x) = W_2(\sigma(W_1(x))), \quad W_\ell(x) = A_\ell(x) + b_\ell, \quad \ell = 1, 2,$$

where

$$A_1 = \begin{pmatrix} (A_1)_{1,1} & (A_1)_{1,2} & 0 \\ 0 & 0 & (A_1)_{2,3} \\ 0 & 0 & (A_1)_{3,3} \end{pmatrix} \qquad b_1 = ((b_1)_1, (b_1)_2, (b_1)_3)^T$$

$$A_2 = \begin{pmatrix} (A_2)_{1,1} & (A_2)_{1,2} & 0 \\ 0 & 0 & (A_2)_{2,3} \end{pmatrix} \qquad b_2 = ((b_2)_1, (b_2)_2)^T$$

Figure 2.3 shows the directed acyclic graph representation of Φ, with $N_0 = 3$ input units (squares), $N_1 = 3$ neurons in the hidden layer with nonlinear activation function (circles), and $N_2 = 2$ output units (diamond). The dotted lines correspond to zero matrix entries.

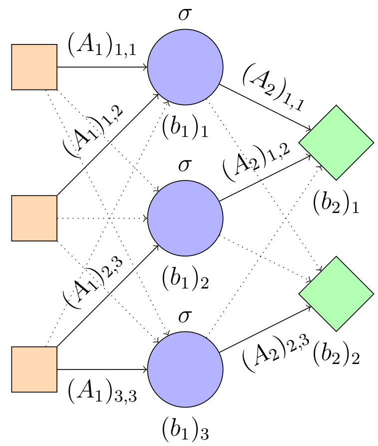


Figure 2.3: ReLU Neural Network

Neural networks can also be constructed as a composition and linear combination of existing networks. As σ is the ReLU activation function, the proofs of the following lemmas will make use of the identity $x = \sigma(x) - \sigma(-x)$.

The first lemma describes how to concatenate neural networks.

Lemma 2.7. *[16, 17] Let $d_1, d_2, d_3 \in \mathbb{N}$. Let $\Phi_1 : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ and $\Phi_2 : \mathbb{R}^{d_2} \to \mathbb{R}^{d_3}$ be ReLU networks. Then there exists a ReLU neural network $\Phi : \mathbb{R}^{d_1} \to \mathbb{R}^{d_3}$, called the concatenation of Φ_1 and Φ_2 with*

$$\mathcal{L}(\Phi) = \mathcal{L}(\Phi_1) + \mathcal{L}(\Phi_2),$$
$$\mathcal{W}(\Phi) \leq 2\mathcal{W}(\Phi_1) + 2\mathcal{W}(\Phi_2),$$
$$\mathcal{M}(\Phi) \leq \max\{2d_2, \mathcal{M}(\Phi_1), \mathcal{M}(\Phi_2\},$$

and satisfying $\Phi(x) = (\Phi_2 \circ \Phi_1)(x)$ for all $x \in \mathbb{R}^{d_1}$

Proof. By Definition 2.1,

$$\Phi_1(x) = W^1_{L_1}\left(\sigma\left(\ldots\sigma\left(W^1_1(x)\right)\right)\right) \text{ and } \Phi_2(x) = W^2_{L_2}\left(\sigma\left(\ldots\sigma\left(W^2_1(x)\right)\right)\right).$$

Consider the affine map $\widetilde{W}(x) = W_1^2\left((\mathbb{I}_{N_{L_1}}\ ,\ -\mathbb{I}_{N_{L_1}})x\right)$, for $x \in \mathbb{R}^{2N_{L_1}}$. Then,

$$W_1^2(\Phi_1(x)) = \tilde{W}\left(\sigma\left(\begin{pmatrix} W_{L_1}^1 \\ \\ -W_{L_1}^1 \end{pmatrix}(\sigma(\dots W_1^1(x)))\right)\right)$$

and the map

$$\Phi(x) = W_{L_2}^2\left(\sigma\left(\dots W_2^2\left(\sigma\left(\tilde{W}\left(\sigma\left(\begin{pmatrix} W_{L_1}^1 \\ \\ -W_{L_1}^1 \end{pmatrix}\left(\sigma\left(\dots W_1^1(x)\right)\right)\right)\right)\right)\right)\right)\right)$$

satisfies $\Phi(x) = (\Phi_2 \circ \Phi_1)(x)$ for all $x \in \mathbb{R}^{d_1}$.

□

The following lemma describes how to construct a network that is the identity map on $\mathbb{R}^d$.

Lemma 2.8. *[17] Let $d, L \in \mathbb{N}$, $L \geq 2$. There exists a network $\Phi : \mathbb{R}^d \to \mathbb{R}^d$ with $\mathcal{L}(\Phi) = L$, and such that $\Phi = Id_{\mathbb{R}^d}$, where $Id_{\mathbb{R}^d}$ is the identity map on $\mathbb{R}^d$.*

Proof. The proof is given for the case $L = 2$.

Define $\Phi(x) = W_2(\sigma(W_1(x)))$, with

$$W_1(x) = A_1 x + b_1, \;\; W_2(x) = A_2 x + b_2$$

where

$$b_1 = \mathbf{0}, \quad A_1 = \begin{pmatrix} \mathbb{I}_d \\ \\ -\mathbb{I}_d \end{pmatrix}, \quad b_2 = \mathbf{0}, \quad A_2 = \begin{pmatrix} \mathbb{I}_d & -\mathbb{I}_d \end{pmatrix}.$$

Then, $\Phi = Id_{\mathbb{R}^d}$.

□

It is also possible to augment depth of an existing neural network without altering the network's input-output relationship:

Lemma 2.9. *[18] Let $L, K, d_1, d_2 \in \mathbb{N}$ with $K > L$. Let $\Phi : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ be a ReLU network with $\mathcal{L}(\Phi) = L$. Then, there exists a corresponding neural network $\Psi : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ with*

$$\begin{aligned}
\mathcal{L}(\Psi) &= K, \\
\mathcal{W}(\Psi) &\leq \mathcal{W}(\Phi) + d_2\mathcal{W}(\Phi) + 2d_2(K - \mathcal{L}(\Phi), \\
\mathcal{M}(\Psi) &= \max\{2d_2, \mathcal{M}(\Phi)\},
\end{aligned}$$

satisfying $\Psi(x) = \Phi(x)$ for all $x \in \mathbb{R}^{d_1}$. Moreover, the weights of Ψ consist of the weights of Φ and $\{-1, 1\}$.

Proof. By Definition 2.1, $\Phi(x) = W_L(\sigma(\ldots\sigma(W_1(x))))$.

Let $\widetilde{W}_j(x) = \text{diag}\,(\mathbb{I}_{d_2}\ \mathbb{I}_{d_2})\,x$, for $j \in \{L+1, \ldots, K-1\}$, and $\widetilde{W}_K(x) = \text{diag}\,(\mathbb{I}_{d_2}\ \ -\mathbb{I}_{d_2})\,x$. Then the network

$$\Psi(x) = \widetilde{W}_K\left(\sigma\left(\ldots\sigma\left(\widetilde{W}_{L+1}\left(\sigma\begin{pmatrix} W_L \\ -W_L \end{pmatrix}\sigma\left(\widetilde{W}_{L-1}\left(\sigma\left(\ldots\sigma\left(W_1(x)\right)\right)\right)\right)\right)\right)\right)\right)$$

satisfies the claimed properties.

$\square$

Next, the concept of parallelization of neural networks of equal depth is formalized.

Lemma 2.10. *[17] Let $n, L \in \mathbb{N}$. For $i = \{1, 2, \ldots, n\}$, let $d_i, d_i' \in \mathbb{N}$ and $\Phi_i : \mathbb{R}^{d_i} \to \mathbb{R}^{d_i'}$ be ReLU neural networks with $\mathcal{L}(\Phi_i) = L$. Then, there exists a network $\Phi : \mathbb{R}^{\sum_{i=1}^n d_i} \to \mathbb{R}^{\sum_{i=1}^n d_i}$, with*

$$\begin{aligned}
\mathcal{L}(\Phi) &= L, \\
\mathcal{M}(\Phi) &= \sum_{i=1}^{n} \mathcal{M}(\Phi_i), \\
\mathcal{W}(\Phi) &= \sum_{i=1}^{n} \mathcal{W}(\Phi_i),
\end{aligned}$$

and satisfies $\Phi(x) = (\Phi_1(x_1), \Phi_2(x_2), \ldots, \Phi_n(x_n))$*, for* $x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^{\sum_{i=1}^{n} d_i}$*, with* $x_i \in \mathbb{R}^{d_i}, i \in \mathbb{N}$.

Proof. By Definition 2.1,

$$\Phi_i = W_L^i \left(\sigma\left(\ldots \sigma\left(W_1^i\right)\right)\right)$$

where $W_\ell^i(x) = A_\ell^i x + b_\ell^i$, for $\ell = 1, \ldots, L$. Denote the dimensions of the layers of Φ_i by $N_0^i, \ldots, N_L^i$ and set $N_l = \sum_{i=1}^{n} N_L^i$, for $\ell \in \{0, 1, \ldots, L\}$. For $l \in \{0, 1, \ldots, L\}$, define the block diagonal matrices $A_\ell := diag\left(A_\ell^1, A_\ell^2, \ldots, A_\ell^n\right)$, the vectors $b_\ell = \left(b_\ell^1, b_\ell^2, \ldots, b_\ell^n\right)$, and the affine transformations $W_\ell(x) := A_\ell x + b_\ell$. Then, $\Phi = W_L(\sigma(W_{L-1}(\sigma(\ldots(\sigma(W_1))))))$ satisfies the claimed properties. □

The general case of Lemma 2.10 follows from Lemma 2.8 and Lemma 2.9. If $\mathcal{L}(\Phi_i) = L_i$ for $i = 1, \ldots, n$ where $L_i \neq L_j$ for $i \neq j$, the shorter neural networks can be extended by concatenating with a network that implements the identity and the result follows.

Next the concept of a linear combination of neural networks of equal depth is formalized.

Lemma 2.11. *[16] Let* $N, L, d' \in \mathbb{N}$*. For* $i \in \{1, \ldots, n\}$ *let* $d_i \in \mathbb{N}, a_i \in \mathbb{R}$*. Let* $\Phi_i : \mathbb{R}^{d_i} \to \mathbb{R}^{d'}$ *be a ReLU neural network with* $\mathcal{L}(\Phi_i) = L$*. Then, there exists a network* $\Phi : \mathbb{R}^{\sum_{i=1}^{n} d_i} \to \mathbb{R}^{d'}$ *with*

$$\begin{aligned}
\mathcal{L}(\Phi) &= L, \\
\mathcal{M}(\Phi) &\leq \sum_{i=1}^{n} \mathcal{M}(\Phi_i), \\
\mathcal{W}(\Phi) &\leq \sum_{i=1}^{n} \mathcal{W}(\Phi_i),
\end{aligned}$$

and satisfying

$$\Phi(x) = \sum_{i=1}^{n} a_i \Phi_i(x_i),$$

for $x = (x_1, \ldots, x_n) \in \mathbb{R}^{\sum_{i=1}^{n} d_i}$*,with* $x_i \in \mathbb{R}^{d_i}$*,* $i \in \{1, \ldots, n\}$.

Proof. In the proof of Lemma 2.10, replace A_L by $(a_1 A_L^1, a_2 A_L^2, \ldots, a_n A_L^n)$, b_L by $\sum_{i=1}^{n} a_i b_L^i$ and note that the resulting network Φ satisfies the claimed properties. $\square$

Example 2.12. Consider two ReLU NN's, $\Phi : \mathbb{R}^2 \to \mathbb{R}$ and $\tilde{\Phi} : \mathbb{R} \to \mathbb{R}$, given by

$$\Phi(x) = W_2\left(\sigma\left((W_1(x)\right)\right) \quad W_\ell(x) = A_\ell x + b_\ell$$

where

$$A_1 \in \mathbb{R}^{3\times 2}, \quad b_1 \in \mathbb{R}^3, \quad A_2 \in \mathbb{R}^{1\times 3}, \quad b_2 \in \mathbb{R}$$

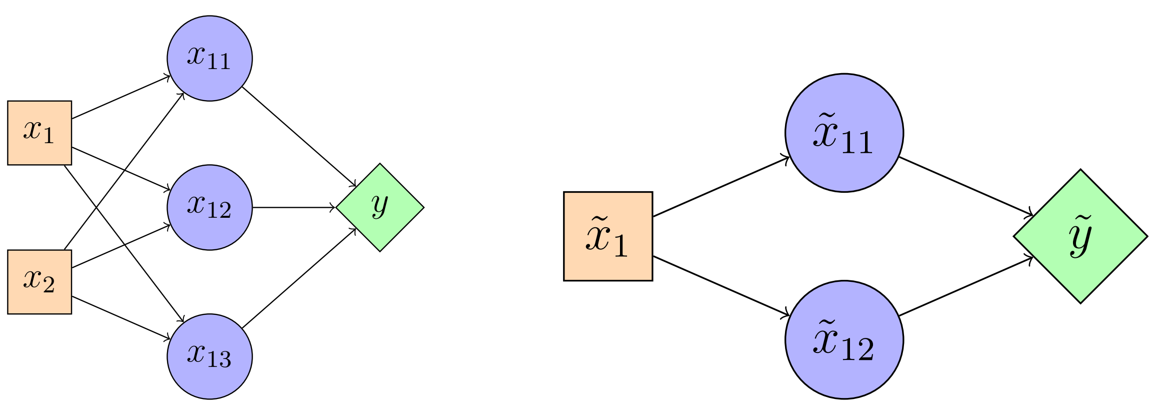
and

$$\tilde{\Phi}(x) = \tilde{W}_2\left(\sigma\left(\tilde{W}_1(x)\right)\right) \quad \tilde{W}_\ell(x) = \tilde{A}_\ell x + \tilde{b}_\ell$$

where

$$\tilde{A}_1 \in \mathbb{R}^{2\times 1}, \quad \tilde{b}_1 \in \mathbb{R}^2, \quad \tilde{A}_2 \in \mathbb{R}^{1\times 2}, \quad \tilde{b}_2 \in \mathbb{R}.$$

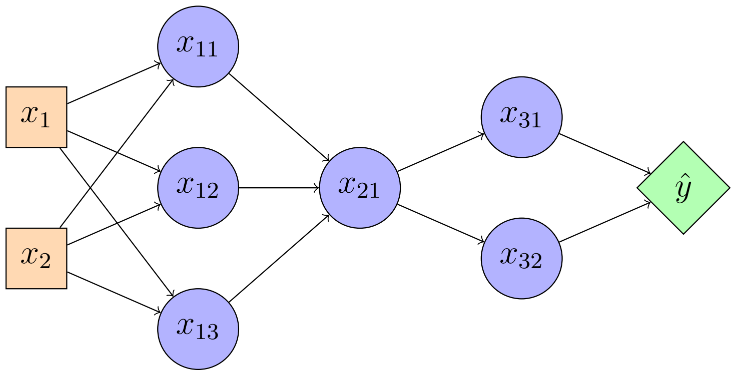
Figures 2.4a and 2.4b show the directed acyclic graph representations of Φ and $\tilde{\Phi}$ respectively.



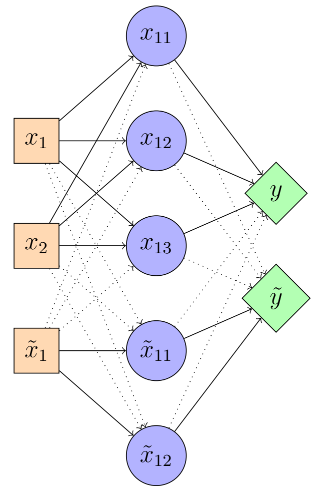
(a) Φ (b) $\tilde{\Phi}$

Figure 2.4: Directed Acyclic Graph Representation of ReLU Neural Networks Φ and $\tilde{\Phi}$

Then, $\Psi_1 : \mathbb{R}^2 \to \mathbb{R}$, given by $\Psi_1 = \tilde{\Phi} \circ \Phi$ is the concatenation of Φ and $\tilde{\Phi}$ (Figure 2.5a), and $\Psi_2 : \mathbb{R}^3 \to \mathbb{R}^2$, given by $\Psi_2 = \left(\Phi, \tilde{\Phi}\right)$ is the parallelization of Φ and $\tilde{\Phi}$ (Figure 2.5b).



(a) The concatenation of Φ and $\tilde{\Phi}$.



(b) The parallelization of Φ and $\tilde{\Phi}$.

Figure 2.5: Concatenation and Parallelization of ReLU Neural Networks

2.3 Deep vs. Shallow Networks

Much of the work on why neural networks work well on machine learning problems has focused on the expressivity of neural networks, that is, what class of functions do neural networks approximate well. While this has been extensively studied in shallow networks, only over the past several years, and after much success in practical applications, have there been attempts to understand the approximation properties of deep neural networks.

A classical result of neural network approximation is the universal approximation theorem proved in 1989 by Cybenko. The universal approximation theorem states that every continuous function on a compact domain can be arbitrarily well approximated by a feedforward neural network with a single hidden layer and continuous, sigmoidal activation function [19]. Hornick, Stinchcomb and White proved a similar result for Borel measurable functions [20]. In 1991, Hornick showed that the universal approximation theorem holds when considering an arbitrary and nonconstant activation function [21]. Later, Pinkus showed that a single hidden layer neural network can approximate any continuous function if and only if the activation function is not a polynomial [22]. While these results imply the theoretical possibility to approximate well, they do not provide a method for finding good approximants. Nor do they provide any information on the required size of a network to achieve a given accuracy. In particular, no constraint is placed on the width of the hidden layer. DNNs also posses the universal approximation property, as do many other families of functions, and thus does not fully explain the success of neural networks.

As expressivity is an effect of depth and width, studies shifted to convergence rates of approximations by neural networks. That is, how many weights and neurons does a neural network need to have to ε-approximate a function from a given class of functions. Barron investigated the number of neurons needed to approximate functions with bounded first Fourier moment by a shallow network with sigmoidal activation function [23, 24]. In 1996, Mhaskar showed for shallow networks, assuming a C^∞ activation function, a network with $\mathcal{O}(\varepsilon^{-d/n})$ neurons is needed to approximate a C^n function on a d-dimensional set with error ε [25]. Maiorov and Pinkus proved there exists an activation function σ which is C^∞, strictly increasing, and sigmoidal, such that any $f \in C([0,1]^d)$ can be uniformly approximated to within any error $\varepsilon > 0$ by a two hidden layer neural network with $2d+1$ units in the first hidden layer and $4d+3$ units the second hidden layer [26].

Recently, there have been many results connecting depth to the expressive power of a neural network. In 2011, Dellaleau and Bengio established that certain classes of polynomials were more easily represented by deep sum-product networks than shallow ones [27]. Montufar et al showed that deep networks can partition a space into exponentially more linear regions than shallow networks of the same size [28]. Telgarsky constructed a class of functions, called sawtooth functions, and showed that these sawtooth functions can be well-approximated by a deep ReLU neural network of width 2, but not by shallow networks with a comparable number of parameters [29].

Arora et al. showed that every ReLU DNN $\Phi : \mathbb{R}^d \to \mathbb{R}$, represents a piecewise linear function, and every piecewise linear function $f : \mathbb{R}^d \to \mathbb{R}$ can be represented by a ReLU DNN with depth at most $\lceil \log_2(d+1) \rceil + 1$ [30]. This result gives an upper bound on the depth of the networks needed to represent all continuous piecewise linear functions on $\mathbb{R}^d$.

Mhaskar and Poggio also considered the conditions under which deep networks are better at approximating functions than shallow networks [31, 32]. They showed that for functions with a compositional structure, that is functions composed of hierarchically local functions, deep networks avoid the curse of dimensionality. The approximation of compositional functions can be achieved with the same degree of accuracy by deep and shallow networks. However, the number of parameters for the deep network are much smaller than for the shallow network with equivalent approximation accuracy. The authors showed, for an M-Lipschitz continuous function of d variables, the number of units needed for a shallow ReLU network, Φ_S, to provide an approximation with accuracy ε is

$$\mathcal{U}(\Phi_S) = \mathcal{O}\left(\left(\frac{\varepsilon}{M}\right)^{-d}\right),$$

whereas for a deep network, Φ_D, the number of units needed is

$$\mathcal{U}(\Phi_D) = \mathcal{O}\left((d-1)\left(\frac{\varepsilon}{M}\right)^{-2}\right),$$

therefore showing that deep compositional networks with ReLU activation function can avoid the curse of dimensionality.

2.4 Approximation of Polynomials and Smooth Functions by ReLU Networks

Recently, the approximation of polynomials by ReLU networks has been studied using the "sawtooth" construction described in Telgarsky [29]. Telgarsky defines a function $g : \mathbb{R} \to \mathbb{R}$ to be t-sawtooth if it is piecewise affine with t pieces, that is, $\mathbb{R}$ can be partitioned into t consecutive intervals and g is affine within each interval.

In 2017, Yarotsky used the result of Telgarsky to show that ReLU networks with unconstrained depth can efficiently approximate the function $f(x) = x^2$ [33].

Proposition 2.13. [33] For any $\varepsilon > 0$, the function $f(x) = x^2$ on I can be realized by a ReLU neural network, Φ_ε, having depth and number of weights and computation units $\mathcal{O}(\ln(1/\varepsilon))$ and such that

$$\|\Phi_\varepsilon - f\|_{L_\infty(I)} \leq \varepsilon. \tag{2.2}$$

Proof. Consider $g : I \to I$, given by

$$g(x) = \begin{cases} 2x, & x < \frac{1}{2} \\ 2(1-x), & x \geq \frac{1}{2} \end{cases} \tag{2.3}$$

and let $g_s = \underbrace{g \circ g \circ \cdots \circ g}_{s \geq 2}$, where $g_0 = 0, g_1 = g$.

Then, by Telgarsky,

$$g_s(x) = \begin{cases} 2^s(x - \frac{2k}{2^s}), & x \in [\frac{2k}{2^s}, \frac{2k+1}{2^s}], \ \ k = 0, 1, \ldots, 2^{s-1} - 1 \\ 2^s(\frac{2k}{2^s} - x), & x \in [\frac{2k-1}{2^s}, \frac{2k}{2^s}], \ \ k = 0, 1, \ldots, 2^{s-1} \end{cases} \tag{2.4}$$

is a sawtooth function with 2^{s-1} uniformly distributed "teeth" where each application of g doubles the number of teeth (Figure 2.6a).

Let f_m be the piecewise linear interpolant of f with $2^m + 1$ uniformly spaced knots (break points) $\frac{k}{2^m}, \ k = 0, \ldots, 2^m$,

$$f_m\Big(\frac{k}{2^m}\Big) = \Big(\frac{k}{2^m}\Big)^2, \ \ k = 0, \ldots, 2^m.$$

Then f_m approximates f with error $\varepsilon_m = 2^{-2m-2}$ in the sense

$$\|f_m - f\|_{L_\infty(I)} \leq 2^{-2m-2}.$$

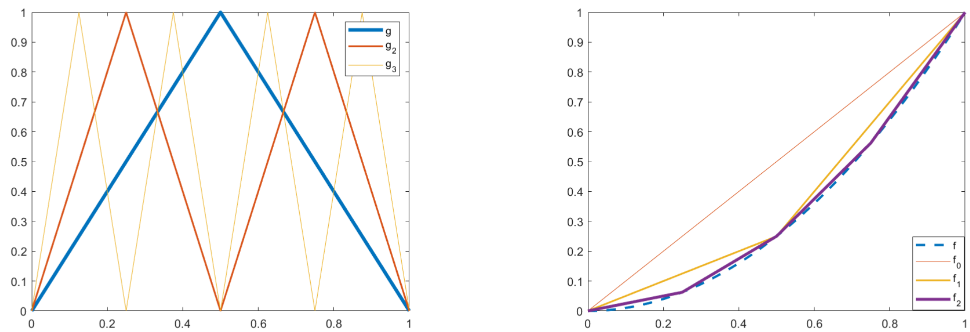
This interpolation can be refined from f_{m-1} to f_m by adjusting it by a function proportional to a sawtooth function (Figure 2.6b),

$$f_{m-1}(x) - f_m(x) = \frac{g_m(x)}{2^{2m}}. \tag{2.5}$$

Thus,

$$f_m(x) = x - \sum_{s=1}^{m} \frac{g_s(x)}{2^{2s}}.$$

Yarotsky's construction of f_m as a neural network uses connections between units in non-consecutive layers. However, f_m can be realized as a fully connected ReLU neural network of width 4, using the language of Definition 2.1, with the same input-output relationship [16].



(a) The "sawtooth" functions g, g_2, g_3.

(b) The approximating functions f_m.

Figure 2.6: Sawtooth Function and Approximating Function

As g can be written as a ReLU neural network of width 3 by

$$g(x) = 2\sigma(x) - 4\sigma(x - 1/2) + 2\sigma(x - 1),$$

it follows that

$$g_m(x) = 2\sigma(g_{m-1}(x)) - 4\sigma(g_{m-1}(x) - 1/2) + 2\sigma(g_{m-1}(x) - 1). \tag{2.6}$$

Using $f_m = \sigma(f_{m-1})$ for all $m \in \mathbb{N}$ equation (2.5) can be rewritten as

$$f_m(x) = \sigma(f_{m-1}(x)) - 2^{-2m}\Big(2\sigma(g_{m-1}(x)) - 4\sigma(g_{m-1}(x) - 1/2) + 2\sigma(g_{m-1}(x) - 1)\Big). \tag{2.7}$$

Writing (2.6) and (2.7) in the language of Definition 2.1 gives

$$\begin{pmatrix} g_m \\ f_m \end{pmatrix} = W_1\left(\sigma\left(W_2\begin{pmatrix} g_{m-1} \\ f_{m-1} \end{pmatrix}\right)\right), \tag{2.8}$$

where

$$W_1(x) = \begin{pmatrix} 2 & -4 & 2 & 0 \\ -2^{-2m+1} & 2^{-2m+2} & -2^{-2m+1} & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}, \quad W_2(x) = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} - \begin{pmatrix} 0 \\ 1/2 \\ 1 \\ 0 \end{pmatrix}.$$

Iteratively applying (2.8), starting with $g_0(x) = x,\ f_0(x) = x$ gives

$$\begin{pmatrix} g_m \\ f_m \end{pmatrix} = W_1\left(\sigma\left(W_2\left(W_1\left(\sigma\left(\ldots\sigma\left(W_2\left(W_1\left(\sigma\left(W_2\begin{pmatrix} x \\ x \end{pmatrix}\right)\right)\right)\right)\ldots\right)\right)\right)\right)\right).$$

Therefore, f_m can be realized by a width 4 ReLU neural network. Since $f_m(0) = 0$ for all $m \in N$ and $\varepsilon_m = 2^{-2m-2}$, then $\ln(1/\varepsilon_m) = 2m + 2$, and the statement of the proposition follows. $\square$

Next, the result of Proposition 2.13 and the identity

$$xy = \frac{1}{2}((x+y)^2 - x^2 - y^2) \tag{2.9}$$

are used to show how to implement the multiplication operation as a deep ReLU network [16, 33].

Proposition 2.14. [16] There exists a constant $M > 0$ such that for all $D > 0$, and $\varepsilon \in (0, 1/2)$, there is a ReLU network $\Phi_{D,\varepsilon} : \mathbb{R}^2 \to \mathbb{R}$, satisfying:

(a) $\Phi_{D,\varepsilon}(0, x) = \Phi_{D,\varepsilon}(x, 0) = 0$, for all $x \in \mathbb{R}$

(b) $\|\Phi_{D,\varepsilon}(x, y) - xy\|_{L_\infty([-D,D]^2)} \leq \varepsilon$

(c) The depth of $\Phi_{D,\varepsilon}$ is at most $M\log(\frac{\lceil D\rceil^2}{\varepsilon})$

Proof. Using Proposition 2.13, let $\Psi_\delta(x)$ be a neural network approximating x^2, such that $\Psi_\delta(0) = 0$ and

$$\left\|\Psi_\delta(x) - x^2\right\|_{L_\infty([0,1])} \leq \delta. \tag{2.10}$$

Then,

$$\left\|4\lceil D\rceil^2\Psi_\delta\left(\frac{|x|}{2\lceil D\rceil}\right) - x^2\right\|_{L_\infty([-D,D])} \leq 4\lceil D\rceil^2\delta \tag{2.11}$$

and so,

$$\left\|4\lceil D\rceil^2\Psi_\delta\left(\frac{|x+y|}{2\lceil D\rceil}\right) - (x+y)^2\right\|_{L_\infty([-D,D]^2)} \leq 4\lceil D\rceil^2\delta. \tag{2.12}$$

For $\delta \in (0, 1/2)$, define

$$\Phi^*_{D,\delta}(x,y) = 2\lceil D\rceil^2\left(\Psi_\delta\left(\frac{|x+y|}{2\lceil D\rceil}\right) - \Psi_\delta\left(\frac{|x|}{2\lceil D\rceil}\right) - \Psi_\delta\left(\frac{|y|}{2\lceil D\rceil}\right)\right). \tag{2.13}$$

Then, property (a) is immediate. Using (2.9) with (2.11) and (2.12) gives

$$\left\|\Phi^*_{D,\delta}(x,y) - \frac{1}{2}\left((x+y)^2 - x^2 - y^2\right)\right\|_{L_\infty([-D,D]^2)} \leq 6\lceil D\rceil^2\delta.$$

Setting, $\Phi_{D,\varepsilon} = \Phi^*_{D,\delta_{D,\varepsilon}}$ with $\delta_{D,\varepsilon} = \frac{\varepsilon}{6\lceil D\rceil^2}$ proves property (b).

To conclude property (c), observe that computation (2.13) consists of three instances of Ψ_δ and finitely many linear and ReLU operations. Thus by Proposition 2.13, $\Phi_{D,\varepsilon}$ can be implemented by a ReLU network whose depth is at most $M\log(1/\delta)$, that is, $M\log(\frac{\lceil D\rceil^2}{\varepsilon})$. □

Proposition 2.14 can be generalized for any input dimension $N_0 > 2$ [34, 35].

Proposition 2.15. [34] For any $D \geq 1$ and $\varepsilon \in (0, 1/2), d \in \mathbb{N}_{\geq 2}$ there is a deep ReLU network Φ with inputs $(x_1, \ldots, x_d) \in [-D, D]^d$, with depth

$$\mathcal{L}(\Phi) = \mathcal{O}(d\log(d/\varepsilon) + d^2\log(D))$$

and number of neurons

$$\mathcal{U}(\Phi) = \mathcal{O}(d \log(d/\varepsilon) + d^2 \log(D)),$$

such that

$$\|\Phi(x_1, \ldots, x_d) - x_1 \cdots x_d\|_{L_\infty([-D,D]^d)} \leq \varepsilon.$$

Using Propositions 2.13, 2.14, and 2.15, polynomials can be approximated by ReLU networks with depth growing logarithmically in the reciprocal of the approximation error [16, 34, 35].

Proposition 2.16. [34] For any $D \geq 1, A \geq 0, \varepsilon \in (0, 1/2)$ and any polynomial $p_n(x)$ of degree $n \in \mathbb{N}_{\geq 2}$, with input $x \in [-D, D]$, of the form $p_n(x) = \sum_{k=0}^{n} a_k x^k$, $\max_{0 \leq k \leq n} |a_k| \leq A$, there exists a deep ReLU network Φ_{p_n} with inputs $(x_1, \ldots, x_n) \in [-D, D]^n$, that has depth

$$\mathcal{L}(\Phi_{p_n}) = \mathcal{O}\left(n \log\left(\frac{An}{\varepsilon}\right) + n^2 \log(D)\right)$$

and number of neurons

$$\mathcal{U}(\Phi_{p_n}) = \mathcal{O}\left(n \log\left(\frac{An}{\varepsilon}\right) + n^2 \log(D)\right),$$

such that

$$\|\Phi_{p_n} - p_n\|_{L_\infty([-D,D])} \leq \varepsilon.$$

Proof. Let $D \geq 1, A \geq 0, \varepsilon \in (0, 1)$ and let $n \in \mathbb{N}_{\geq 2}$ and consider the polynomial

$$p_n(x) = \sum_{k=0}^{n} a_k x^k, \quad \max_{0 \leq k \leq n} |a_k| \leq A.$$

Construct Φ_{p_n} as follows:

$$\Phi_{p_n}(x_1, \ldots, x_n) = a_0 + a_1 x_1 + \sum_{k=2}^{n} a_k R(\Psi_{k-1})(x_1, \ldots, x_k),$$

where $\Psi_{k-1}(x_1, \ldots, x_k)$ approximates the product $x_1 \cdots x_k$ with the network of Proposition 2.15

to accuracy $\varepsilon_0 \in (0,1)$, to be determined later. When the inputs are the same, $\Psi_{k-1}(x_1, \ldots, x_k)$ approximates x^k. Then,

$$
\begin{aligned}
|\Phi_{p_n}(x, \ldots, x) - p_n(x)| &\leq A \sum_{k=2}^{n} |\Psi_{k-1}(x, \ldots, x) - x^k| \\
&< nA\varepsilon_0.
\end{aligned}
$$

Choose $\varepsilon_0 = \varepsilon/(An)$. The network Φ_{p_n} has depth and number of units

$$
\mathcal{O}\left(n \log\left(\frac{An^2 D^n}{\varepsilon}\right)\right) = \mathcal{O}\left(n \log\left(\frac{An}{\varepsilon}\right) + n^2 \log(D)\right).
$$

□

A more general result considering the Sobolev space $\mathcal{W}^{n,\infty}(I^d)$ for $n \in \mathbb{N}$ is given in Yarotsky [33]. $\mathcal{W}^{n,\infty}(I^d)$ is the space of functions on I^d lying in L_∞ along with their weak derivatives up to order n. Equivalently, $\mathcal{W}^{n,\infty}(I^d)$ can be described as the functions from $C^{n-1}(I^d)$ whose derivatives of order $(n-1)$ are Lipschitz continuous. Define the norm in $\mathcal{W}^{n,\infty}(I^d)$ by

$$
\|f\|_{\mathcal{W}^{n,\infty}([0,1]^d)} = \max_{\mathbf{n}:|\mathbf{n}|\leq n} \operatorname*{ess\,sup}_{\mathbf{x}\in I^d} |D^{\mathbf{n}} f(\mathbf{x})|,
$$

where $\mathbf{n} = (n_1, \ldots, n) \in \mathbb{N}_0^d$, $|\mathbf{n}| = n_1 + \cdots + n_d$ and $D^{\mathbf{n}} f$ is the respective weak derivative. Define $F_{n,d}$ to be the unit ball in $\mathcal{W}^{n,\infty}(I^d)$,

$$
F_{n,d} = \left\{ f \in \mathcal{W}^{n,\infty}\left(I^d\right) : \|f\|_{\mathcal{W}^{n,\infty}\left(I^d\right)} \leq 1 \right\}. \tag{2.14}
$$

In [33], Yarotsky provides bounds on the total number of parameters in a ReLU network needed to approximate functions in $F_{n,d}$.

Theorem 2.17. *[33] For any $d, n \in \mathbb{N}$, and $\varepsilon \in (0,1)$, there is a ReLU neural network Φ capable* *of expressing any function $f \in F_{n,d}$ with error ε,*

$$\|\Phi - f\|_{L_\infty(I^d)} \leq \varepsilon,$$

has depth at most $c(\ln(1/\varepsilon) + 1)$, and at most $c\varepsilon^{\frac{-d}{n}}(\ln(1/\varepsilon) + 1)$ weights and computation units, for some constant $c = c(d, n)$.

Proof. The first step of the proof is to approximate a function $f \in F_{n,d}$ by a combination f_1 of local Taylor polynomials and one-dimensional piecewise-linear functions. Then the results of Propositions 2.13 and 2.14 are used to approximate f_1 by a neural network.

Let N be a positive integer, and consider a partition of unity formed by a grid of $(N + 1)^d$ functions $\phi_{\mathbf{m}}$ on I^d,$\sum_{\mathbf{m}} \phi_{\mathbf{m}}(\mathbf{x}) \equiv 1, \;\; \mathbf{x} \in I^d$, for $\mathbf{m} = (m_1, \ldots, m_d) \in \{0, 1, \ldots, N\}^d$.

Define $\phi_{\mathbf{m}}$ by

$$\phi_{\mathbf{m}}(\mathbf{x}) = \prod_{k=1}^{d} \psi\Big(3N(x_k - \frac{m_k}{N})\Big),$$

where

$$\psi(x) = \begin{cases} 1, & |x| < 1, \\ 0, & 2 < |x| \\ 2 - |x|, & 1 \leq |x| \leq 2 \end{cases}.$$

Note that $\|\psi\|_\infty = 1$ and $\|\phi_{\mathbf{m}}\|_\infty = 1, \forall \mathbf{m}$ and supp $\phi_{\mathbf{m}} \subset \Big\{\mathbf{x} : \Big|x_k - \frac{m_k}{N}\Big| \leq \frac{1}{N}, \; \forall k\Big\}$.

For any $\mathbf{m} \in \{0, \ldots, N\}^d$, consider the degree-$(n - 1)$ Taylor polynomial for the function f centered at $\mathbf{x} = \frac{\mathbf{m}}{N}$,

$$P_{\mathbf{m}}(\mathbf{x}) = \sum_{\mathbf{n}:|\mathbf{n}|<n} \frac{D^{\mathbf{n}} f}{\mathbf{n}!}\bigg|_{\mathbf{x}=\frac{\mathbf{m}}{N}} \left(\mathbf{x} - \frac{\mathbf{m}}{N}\right)^{\mathbf{n}},$$

where $\mathbf{n}! = \prod_{k=1}^{d} n_k!$ and $(\mathbf{x} - \frac{\mathbf{m}}{N})^{\mathbf{n}} = \prod_{k=1}^{d}(x_k - \frac{m_k}{N})^{n_k}$. Define an approximation to f by

$$f_1 = \sum_{\mathbf{m} \in \{0,\ldots,N\}^d} \phi_{\mathbf{m}} P_{\mathbf{m}}. \tag{2.15}$$

Then,

$$
\begin{aligned}
|f(\mathbf{x}) - f_1(\mathbf{x})| &= |\sum_{\mathbf{m}} \phi_{\mathbf{m}}(\mathbf{x})(f(\mathbf{x}) - P_{\mathbf{m}}(\mathbf{x}))| \\
&\leq \sum_{\mathbf{m}:|x_k - \frac{m_k}{N}| < \frac{1}{N}\ \forall k} |f(\mathbf{x}) - P_{\mathbf{m}}(\mathbf{x})| \\
&\leq 2^d \max_{\mathbf{m}:|x_k - \frac{m_k}{N}| < \frac{1}{N}\ \forall k} |f(\mathbf{x}) - P_{\mathbf{m}}(\mathbf{x})| \\
&\leq \frac{2^d d^n}{n!} \Big(\frac{1}{N}\Big)^n \max_{\mathbf{n}:|\mathbf{n}|=n} \operatorname*{ess\,sup}_{\mathbf{x}\in[0,1]^d} |D^n f(\mathbf{x})| \\
&\leq \frac{2^d d^n}{n!} \Big(\frac{1}{N}\Big)^n
\end{aligned}
$$

Now choose

$$N = \left\lceil \Big(\frac{n!}{2^d d^n} \frac{\varepsilon}{2}\Big)^{-1/n} \right\rceil \tag{2.16}$$

so that

$$\|f - f_1\|_{L_\infty(I^d)} \leq \frac{\varepsilon}{2}.$$

Note that the coefficients $a_{\mathbf{m},\mathbf{n}}$ of the polynomials $P_{\mathbf{m}}$ are uniformly bounded for all $f \in F_{d,n}$,

$$P_{\mathbf{m}}(\mathbf{x}) = \sum_{\mathbf{n}:|\mathbf{n}|<n} a_{\mathbf{m},\mathbf{n}} \Big(\mathbf{x} - \frac{\mathbf{m}}{N}\Big)^{\mathbf{n}}, \quad |a_{\mathbf{m},\mathbf{n}}| \leq 1. \tag{2.17}$$

Next, construct a network capable of approximating with uniform error $\frac{\varepsilon}{2}$ any function of the form (2.15), assuming that N is given by (2.16) and the polynomials $P_{\mathbf{m}}$ are of the form (2.17). Write f_1 as

$$f_1(\mathbf{x}) = \sum_{\mathbf{m}\in\{0,\ldots,N\}^d} \sum_{\mathbf{n}:|\mathbf{n}|<n} a_{\mathbf{m},\mathbf{n}} \phi_{\mathbf{m}}(\mathbf{x}) \Big(\mathbf{x} - \frac{\mathbf{m}}{N}\Big)^{\mathbf{n}}.$$

This expansion is a linear combination of at most $d^n(N+1)^d$ terms $\phi_{\mathbf{m}}(\mathbf{x})\Big(\mathbf{x} - \frac{\mathbf{m}}{N}\Big)^{\mathbf{n}}$ and each of these terms is a product of at most d functions $\psi(3Nx_k - 3m_k)$ and at most $(n-1)$ linear expressions $(x_k - \frac{m_k}{N})$.

Consider $\tilde{f}_{\mathbf{m},\mathbf{n}}$ the approximation of the product $\phi_{\mathbf{m}}(\mathbf{x})\Big(\mathbf{x} - \frac{\mathbf{m}}{N}\Big)^{\mathbf{n}}$. By Proposition 2.14,

$\tilde{f}_{\mathbf{m},\mathbf{n}}$ can be implemented by a ReLU network with depth and complexity not greater than $(d + n)c_1 \ln(1/\delta)$, for some accuracy δ to be chosen later and some constant $c_1 = c_1(d, n)$. Then,

$$|\tilde{f}_{\mathbf{m},\mathbf{n}}(\mathbf{x}) - \phi_{\mathbf{m}}(\mathbf{x})\Big(\mathbf{x} - \frac{\mathbf{m}}{N}\Big)^{\mathbf{n}}| \leq (d+n)\delta,$$

and $\tilde{f}_{\mathbf{m},\mathbf{n}}(\mathbf{x}) = \phi_{\mathbf{m}}(\mathbf{x})\Big(\mathbf{x} - \frac{\mathbf{m}}{N}\Big)^{\mathbf{n}}$, for $\mathbf{x} \in \text{supp } \phi_{\mathbf{m}}$. Define the full approximation by

$$\tilde{f} = \sum_{\mathbf{m} \in \{0,\ldots,N\}^d} \sum_{\mathbf{n}:|\mathbf{n}|<n} a_{\mathbf{m},\mathbf{n}} \tilde{f}_{\mathbf{m},\mathbf{n}}.$$

Then,

$$\begin{aligned}
|\tilde{f}(\mathbf{x}) - f_1(\mathbf{x})| &= \Bigg| \sum_{\mathbf{m} \in \{0,\ldots,N\}^d} \sum_{\mathbf{n}:|\mathbf{n}|<n} a_{\mathbf{m},\mathbf{n}} \bigg(\tilde{f}_{\mathbf{m},\mathbf{n}}(\mathbf{x}) - \phi_{\mathbf{m}}(\mathbf{x})\Big(\mathbf{x} - \frac{\mathbf{m}}{N}\Big)^{\mathbf{n}} \bigg) \Bigg| \\
&= \Bigg| \sum_{\mathbf{m}:x \in \text{supp } \phi_{\mathbf{m}}} \sum_{\mathbf{n}:|\mathbf{n}|<n} a_{\mathbf{m},\mathbf{n}} \bigg(\tilde{f}_{\mathbf{m},\mathbf{n}}(\mathbf{x}) - \phi_{\mathbf{m}}(\mathbf{x})\Big(\mathbf{x} - \frac{\mathbf{m}}{N}\Big)^{\mathbf{n}} \bigg) \Bigg| \\
&\leq 2^d \max_{\mathbf{m}:x \in \text{supp } \phi_{\mathbf{m}}} \sum_{\mathbf{n}:|\mathbf{n}|<n} \Bigg| \tilde{f}_{\mathbf{m},\mathbf{n}}(\mathbf{x}) - \phi_{\mathbf{m}}(\mathbf{x})\Big(\mathbf{x} - \frac{\mathbf{m}}{N}\Big)^{\mathbf{n}} \Bigg| \\
&\leq 2^d d^n (d+n)\delta.
\end{aligned}$$

Thus, choosing

$$\delta = \frac{\varepsilon}{2^{d+1} d^n (d+n)}, \tag{2.18}$$

then $\left\| \tilde{f} - f_1 \right\|_{L_\infty(I^d)} \leq \frac{\varepsilon}{2}$ and hence

$$\left\| \tilde{f} - f \right\|_{L_\infty(I^d)} \leq \left\| \tilde{f} - f_1 \right\|_{L_\infty(I^d)} + \left\| f_1 - f \right\|_{L_\infty(I^d)} \leq \frac{\varepsilon}{2} + \frac{\varepsilon}{2} \leq \varepsilon.$$

$\tilde{f}$ can be implemented by a ReLU network Φ consisting of parallel subnetworks that compute each $\tilde{f}_{\mathbf{m},\mathbf{n}}$, then weighing the outputs of the subnetworks with the weights $a_{\mathbf{m},\mathbf{n}}$. Each subnetwork has at most $c_1 \ln(1/\delta)$ layers, weights and computation units with $c_1 = c_1(d, n)$. There are at most $d^n(N+1)^d$ subnetworks, thus Φ has at most $c_1 \ln(1/\delta) + 1$ layers and $d^n(N+1)^d(c_1 \ln(1/\delta) + 1)$

weights and computation units. With δ given by (2.18) and N given by (2.16), the claim of the theorem follows. $\square$

Grohs [16] extended the result of Yarotsky in the case $d = 1$ by considering, for $D > 0$, the set $S_D \subseteq C^\infty([-D, D], \mathbb{R})$ given by

$$S_D = \{f \in C^\infty([-D, D], \mathbb{R}) : \left\|f^{(n)}\right\|_{L_\infty([-D,D])} \leq n!, \;\; \forall n \in N\}. \tag{2.19}$$

Lemma 2.18. *[16] There exists a constant $C > 0$ such that for all $D \in \mathbb{R}_+$, $f \in S_D$ and $\varepsilon \in (0, 1/2)$, there exists a network $\Psi_{f,\varepsilon}$ of depth at most $C\lceil D\rceil(\log(\varepsilon^{-1}))^2$ satisfying*

$$\|\Psi_{f,\varepsilon} - f\|_{L_\infty([-D,D])} \leq \varepsilon.$$

Proof. The proof of the case $D = 1$ is provided.

Using Chebyshev interpolation, for all $f \in S_1, n \in \mathbb{N}$ there exists a polynomial $P_{f,n}$ of degree n such that

$$\|f - P_{f,n}\|_{L_\infty([-1,1])} \leq \frac{1}{2^n(n+1)!}\|f^{(n+1)}\|_{L_\infty([-1,1])} \leq \frac{1}{2^n}. \tag{2.20}$$

Writing the polynomials $P_{f,n}$ as $P_{f,n} = \sum_{j=0}^{n} a_{f,n,j}x^j$, there exists a constant $c > 0$ such that for all $f \in S_1$, $n \in \mathbb{N}$,

$$A_{f,n} \coloneqq \max_{j=0,\ldots,n} |a_{f,n,j}| \leq 2^{cn}.$$

By Proposition 2.15 there exists a constant $C_1 > 0$ such that for all $f \in S_1, n \in \mathbb{N}, \varepsilon \in (0, 1/2)$, there is a network $\Phi_{P_{f,n},1,\varepsilon/2}$, with input and output dimension 1, depth at most $C_1 n(cn+\log(2/\varepsilon)+\log(n))$, and satisfies

$$\|\Phi_{P_{f,n},1,\varepsilon/2} - P_{f,n}\|_{L^\infty([-1,1])} \leq \frac{\varepsilon}{2}. \tag{2.21}$$

Set $n_\varepsilon = \lceil\log(2/\varepsilon)\rceil$ and $\Psi_{f,\varepsilon} = \Phi_{P_{f,n},1,\varepsilon/2}$. Then, by (2.20) and (2.21), for all $f \in \mathcal{S}_1, \varepsilon \in (0, 1/2)$,

$$\|\Psi_{f,\varepsilon} - f\|_{L^\infty([-1,1])} \leq \|\Psi_{f,\varepsilon} - P_{f,n_\varepsilon}\|_{L^\infty([-1,1])} + \|P_{f,n_\varepsilon} - f\|_{L^\infty([-1,1])}$$

$$
\begin{aligned}
&\le \frac{\varepsilon}{2} + \frac{1}{2^{n_\varepsilon}} \\
&\le \frac{\varepsilon}{2} + \frac{\varepsilon}{2} \\
&= \varepsilon.
\end{aligned}
$$

Using $\lceil \log(2/\varepsilon) \rceil \le 2\log(2/\varepsilon)$ and $\log(2/\varepsilon) \le 2\log(1/\varepsilon)$ for all $\varepsilon \in (0, 1/2)$,then there exists a constant C_2 such that for all $f \in S_1, \varepsilon \in (0, 1/2)$, the depth of $\Psi_{f,\varepsilon}$ is equal to the depth of $\Phi_{P_{f,n},1,\varepsilon/2}$ and is not more than $C_2(\log(\varepsilon^{-1}))^2$. $\square$

Petersen and Voigtlaender [17] further generalized Theorem 2.17 of Yarotsky to hold in L_p, for any $p \in (0, \infty)$. In addtion they showed for a given piecewise C^β function $f : [-1/2, 1/2]^d \to \mathbb{R}$ and approximation accuracy $\varepsilon \in (0, 1/2)$ it is possible to construct a ReLU network $\Phi_{f,\varepsilon}$ with no more than $c\varepsilon^{-2\frac{(d-1)}{\beta}}$ nonzero weights and $c' \log_2(\beta + 2)(1 + \frac{\beta}{d})$ layers such that $\|f - \Phi_{f,\varepsilon}\|_{L_2([-1/2,1/2])} \le \varepsilon$.

In [36, 37], it is suggested that deep neural networks possess greater approximation power than traditional methods based on linear approximation. Yarotsky [36] showed that Lipshitz spaces can be approximated at a slightly better rate by ReLU networks than by classical linear methods. In [37], Daubechies et al. prove a similar result for $Lip(\alpha)$ spaces, using the following theorem that fixed width ReLU networks depending on n parameters are at least as expressive as free knot linear splines [37].

Theorem 2.19. *[37] Fix $W \ge 4$. For every $n \ge 1$, the set Σ_n of free knot linear splines with n breakpoints is contained in $\Upsilon^{W,L}$, the set of functions produced by width W and depth L ReLU networks.*

Theorem 2.20. *[37] Let $W \ge 8$. Let $\overline{\Upsilon}_m := \overline{\Upsilon}^{W+2,m} \subset \Upsilon^{W+2,m}$ be special ReLU networks with fixed width $W + 2$. If $X = C([0, 1])$ and $f \in Lip(\alpha), 0 < \alpha \le 1$, then*

$$
\inf_{S \in \overline{\Upsilon}_m} \|f - S\|_X \le \frac{|f|_{Lip\alpha}}{(m \ln m)^\alpha},\ m \ge 2. \tag{2.22}
$$

Proof. Without loss of generality, assume $|f|_{Lip(\alpha)} = 1$. Fix f and m and let T be the piecewise linear interpolant of f at the equally spaced breakpoints $x_0, \ldots, x_m$, where $x_i := \frac{i}{m}, i = 0, \ldots, m$. As f, T agree at the endpoints of the interval $J_i := [x_i, x_{i+1}]$, the slope of T on J_i has absolute value at most $m^{1-\alpha}$. Therefore,

$$|T(x) - T(y)| \leq m^{1-\alpha}|x - y| \leq |x - y|^\alpha, \;\; x, y, \in J_i,$$

and $T \in Lip(\alpha)$ with semi-norm at most one on each of these intervals. Define $g := f - T$ and write $g = \sum_{i=1}^m g\chi_{J_i}$. Each $g_i := g\chi_{J_i}$ is a function in $Lip(\alpha)$ with $|g_i|_{Lip(\alpha)} \leq 2$. Let k be the largest integer such that $3^k k \leq m$ and let $\mathcal{P} = \{S_1, \ldots, S_{3^k}\}$. For each $\overline{g}_i : [0,1] \to \mathbb{R}$, defined by $\overline{g}_i := 2^{-1}m^\alpha g_i((x+i)/m)$, find a pattern $S_{j_i} \in \mathcal{P}, S_{j_i} : [0,1] \to \mathbb{R}$ such that $\|\overline{g}_i - S_{j_i}\|_{c([0,1])} \leq 2k^{-\alpha}$.

Going back to the interval J_i provides a function $S_{j_i} \in \mathcal{P}$ such that

$$|g_i(x) - 2m^\alpha S_{j_i}(m(x - x_i))| \leq 4(km)^{-\alpha}, \;\; x \in J_i,$$

and thus the function $\hat{T}$ given by

$$\hat{T}(x) := T(x) + 2m^{-\alpha} \sum_{i=1}^{m} S_{j_i}(m(x - x_i))\chi_{J_i}(x)$$

approximates f to accuracy $4(km)^{-\alpha}$ in the uniform norm.

For each $j = 1, \ldots, 3^k$, consider the, possibly empty, set of indices $\Lambda_j = \{i \in \{1, \ldots, m\} : j_i = j\}$. Then,

$$\hat{T} = T + \sum_{j=1}^{3^k} T_j, \;\; \text{where } T_j := 2m^{-\alpha} \sum_{i \in \Lambda_j} S_j(m(x - x_i)).$$

As $T \in \Sigma_m$, by Theorem 2.19 T belongs to $\overline{\Upsilon}^{W,L_0}$ with either $W^2 L_0 \asymp n(W, L_0) \leq C'm$ or $L_0 = 2$. Thus, each function T_j is in $\overline{\Upsilon}^{W,L_j}$ with either $W^2 L_j \asymp n(W, L_j) \leq C_1(k + m_j) + C_2 W^2$

or $L_j = 2$, where $m_j := |\Lambda_j|$. Thus, $\hat{T}$ belongs to $\overline{\Upsilon}^{W,L}$ with $L = L_0 + \sum_{j=1}^{3^k} L_j$, and

$$L = L_0 + \sum_{j=1}^{3^k} L_j \leq \frac{1}{W^2}\left(C'm + C_1 3^k k + C_1 \sum_{j=1}^{3^k} m_j\right) + C_3 3^k \leq \left(\frac{\tilde{C}_1}{W^2} + \tilde{C}_2\right) m = c(W)m,$$

using the fact $3^k k \leq m$ and $\sum_{j=1}^{3^k} m_j = m$. Therefore,

$$\|f - \hat{T}\|_{C([0,1])} \leq \frac{4}{(km)^\alpha} \leq \frac{\tilde{C}}{(m\ln(m))^\alpha}, \tag{2.23}$$

using $k \geq c\ln(m)$ since $3^{k+1} > m$. $\square$

Chapter 3

Cascade Networks

3.1 Generalized Neural Networks and Cascade Network Motivation

In its most general form, a neural network is a function resulting from repeatedly applying an activation function, σ, to an affine function, W_ℓ. The activation function σ can be viewed as an operator which maps functions to functions, and W_ℓ as a linear operator. This idea can be generalized by replacing σ with a more general operator.

Definition 3.1. Let V be a space of vector functions. A *generalized neural network*, Ψ_L, is given by

$$\Psi_L = P_L \circ Q \circ P_{L-1} \circ Q \cdots \circ P_1,$$

where $P_\ell : V \to V$, are affine operators, and $Q : V \to V$ is fixed operator, in general non-linear.

In the case where Ψ_L is a neural network as defined in Definition 2.1, $P_\ell y = A_\ell y + b_\ell$, for $y \in V$ and $Qy = \sigma(y)$, $y \in V$. When σ is the ReLU function, Q is a non-linear operator and the output of the network is a continuous, piecewise linear function.

Thus, an essential question is why resort to using neural networks over classical methods of approximation such as polynomials, wavelets, or splines. In addition, can well-established tools be used in a framework that mimics the neural network framework? In particular, can the approximation power of a ReLU network be matched without the non-linearity that comes from σ? Further, can even simpler operators P_ℓ, Q be considered that will still do a good job in approximation?

This generalization of ReLU neural networks as a piecewise linear function resulting from the repeated application of a fixed operator followed by an affine operator was the motivation for the development of *cascade networks*. The terminology comes from identities which arise in multiresolution analysis and wavelets to generate functions on a dyadic partition [38].

To motivate the definition of cascade networks, consider the following example.

Example 3.2. Consider Π_2, the space of polynomials of degree at most two, with basis $\mathcal{B} = \{1, x, x^2\}$. Let $f_0(x) = 1, f_1(x) = x, f_2(x) = x^2$ and write f_i^1 for the linear interpolant of f_i on I at endpoints $\Omega_0 = \{0, 1\}$ where $i = 0, 1, 2$. Hence $f_0^1(x) = 1, f_1^1(x) = x$, and $f_2^1(x) = x$, for $x \in I$.

Define the function $\alpha : I \to I$ given by

$$\alpha(x) = \begin{cases} 2x, & x \in \left[0, \frac{1}{2}\right) \\ 2x - 1, & x \in \left[\frac{1}{2}, 1\right] \end{cases}. \tag{3.1}$$

For $i = 0, 1, 2$ and $j \in \mathbb{N}$ define

$$f_i^{j+1}(x) = a_{i0}(x) f_0^j(\alpha(x)) + a_{i1}(x) f_1^j(\alpha(x)) + a_{i2}(x) f_2^j(\alpha(x)), \tag{3.2}$$

where

$$a_{00}(x) = \begin{cases} 1, & x \in [0, \frac{1}{2}) \\ 1, & x \in [\frac{1}{2}, 1] \end{cases}; \quad a_{01}(x) = \begin{cases} 0, & x \in [0, \frac{1}{2}) \\ 0, & x \in [\frac{1}{2}, 1] \end{cases}; \quad a_{02}(x) = \begin{cases} 0, & x \in [0, \frac{1}{2}) \\ 0, & x \in [\frac{1}{2}, 1] \end{cases};$$

$$a_{10}(x) = \begin{cases} 0, & x \in [0, \frac{1}{2}) \\ \frac{1}{2}, & x \in [\frac{1}{2}, 1] \end{cases}; \quad a_{11}(x) = \begin{cases} \frac{1}{2}, & x \in [0, \frac{1}{2}) \\ \frac{1}{2}, & x \in [\frac{1}{2}, 1] \end{cases}; \quad a_{12}(x) = \begin{cases} 0, & x \in [0, \frac{1}{2}) \\ 0, & x \in [\frac{1}{2}, 1] \end{cases};$$

$$a_{20}(x) = \begin{cases} 0, & x \in [0, \frac{1}{2}) \\ \frac{1}{4}, & x \in [\frac{1}{2}, 1] \end{cases}; \quad a_{21}(x) = \begin{cases} 0, & x \in [0, \frac{1}{2}) \\ \frac{1}{2}, & x \in [\frac{1}{2}, 1] \end{cases}; \quad a_{22}(x) = \begin{cases} \frac{1}{4}, & x \in [0, \frac{1}{4}) \\ \frac{1}{4}, & x \in [\frac{1}{4}, 1] \end{cases}.$$

The motivation for identity (3.2) is that the same identity is satisfied by the basis functions in $\mathcal{B}$ as well. Namely, one can show that for $i = 0, 1, 2$,

$$f_i(x) = a_{i0}(x) f_0(\alpha(x)) + a_{i1}(x) f_1(\alpha(x)) + a_{i2}(x) f_2(\alpha(x)). \tag{3.3}$$

Identities of this type arise in multiresolution analysis and wavelets [38]. Equation (3.2) gives rise to the so-called cascade algorithm, hence our terminology.

It can be easily shown that f_i^{j+1} is the piecewise linear interpolant of f_i on $\Omega_j = \{0, 2^{-j}, \dots, 1\}$, for $i = 0, 1, 2$ and $j \in \mathbb{N}_0$.

In a similar manner, this procedure can be generalized to obtain the piecewise linear interpolant of a degree n polynomial, $f \in \Pi_n$, with basis $\beta = \{f_0, \dots, f_n\}$. Then for $j \in \mathbb{N}$,

$$f_i^{j+1}(x) = \sum_{k=0}^{n} a_{ik}(x) f_k^j(\alpha(x)), \tag{3.4}$$

for appropriately chosen coefficients a_{ik}, where $f_0^1, \dots, f_n^1$ are the piecewise linear interpolants of f_i on Ω_0.

3.2 Cascade Network Definition and Properties

In this section, cascade networks (CN) are formally defined and several properties are discussed. Throughout let $\Omega_j = \{0, 2^{-j}, \dots, 1\}$, for $j \in \mathbb{N}$ and α is defined as in (3.1). Write $\alpha^k(x) = \underbrace{\alpha \circ \alpha \circ \cdots \circ \alpha}_{k}(x)$ for the k-fold composition of α, $k \geq 2$. Set $\alpha^1(x) = \alpha(x)$ and $\alpha^0(x) = x$.

Definition 3.3. Let $N_0, N_1, \dots, N_L, L \in \mathbb{N}$, $x \in I$. A *cascade network* $\mathcal{Y}_L$ is a vector-valued function defined recursively as

$$\mathcal{Y}_\ell(x) = A_\ell(x)\mathcal{Y}_{\ell-1}(\alpha(x)) + b_\ell(x), \quad \ell = 1, \dots, L, \tag{3.5}$$

where $\mathcal{Y}_0 \in \mathbb{R}^{N_0}$ the input vector function of x, and $A_\ell \in \mathbb{R}^{N_\ell \times N_{\ell-1}}$ are two valued weight matrices

$$A_\ell(x) = \begin{cases} A_\ell^0, & x \in \left[0, \frac{1}{2}\right) \\ A_\ell^1, & x \in \left[\frac{1}{2}, 1\right] \end{cases},$$

and $b_\ell \in \mathbb{R}^{N_\ell}$ are two -valued bias vectors

$$b_\ell(x) = \begin{cases} b_\ell^0, & x \in \left[0, \frac{1}{2}\right) \\ b_\ell^1, & x \in \left[\frac{1}{2}, 1\right] \end{cases},$$

for $\ell = 1, \ldots, L$.

This means the entries of A_ℓ and b_ℓ are piecewise constant on the mesh Ω_1, and the functions $\mathcal{Y}_\ell(x) = A_\ell(x)\mathcal{Y}_{\ell-1}(\alpha(x)) + b_\ell(x)$, for $\ell = 1, \ldots, L$, are "piecewise affine" maps.

$\mathcal{Y}_L$ is used to denote the cascade network and its architecture, as well as the function it implements.

Cascade networks are an example of generalized neural networks, as cascade networks are also functions resulting from repeatedly applying a fixed operator to an affine operator. Viewing Definition 3.1 in the CN setting, the affine operator P_ℓ is given by $P_\ell y = A_\ell y + b_\ell$, $y \in V$, and $Qy = y \circ \alpha$, $y \in V$. Note in the CN case, as opposed to the ReLU NN case, Q is a linear operator. Further, if the input vector $\mathcal{Y}_0$ is linear in x, the output of a CN is a piecewise linear function.

Remark 3.1. *If $N_\ell = W$ for $\ell = 0, \ldots, L-1$, then $\mathcal{Y}_L$ is called a fixed width cascade network.*

Without loss of generality, from now on, assume that the bias terms b_ℓ are zero for $\ell = 1, \ldots, L$. Indeed, (3.5) can be rewritten as

$$\tilde{\mathcal{Y}}_\ell(x) = \tilde{A}_\ell(x)\tilde{\mathcal{Y}}_{\ell-1}(\alpha(x)), \tag{3.6}$$

where $\tilde{\mathcal{Y}}_\ell(x) := (1, \mathcal{Y}_\ell(x))^T$ and $\tilde{A}_\ell := \left(\begin{array}{c|c} 1 & \mathbf{0} \\ \hline b_\ell & A_\ell \end{array}\right)$, $\ell = 1, \ldots, L$ and $\mathcal{Y}_0 \in \mathbb{R}^{N_0}$.

As with neural networks, the following criteria is used to determine the complexity of a cascade network.

Definition 3.4. The number of *weights* of $\mathcal{Y}_L$, $\mathcal{W}(\mathcal{Y}_L)$, is defined as the total number of nonzero entries of the matrices A_ℓ and the vectors b_ℓ, for $\ell = 1, \ldots, L$.

Definition 3.5. The *depth* of $\mathcal{Y}_L$ is $\mathcal{L}(\mathcal{Y}_L) = L$.

Definition 3.6. The number of *units* of $\mathcal{Y}_L$ is $\mathcal{U}(\mathcal{Y}_L) = \sum_{\ell=1}^{L} N_\ell$.

Definition 3.7. The *width* of $\mathcal{Y}_L$ is $\mathcal{M}(\mathcal{Y}_L) = \max_{\ell=1,\dots,L} N_\ell$.

Example 3.8. Let $f \in \Pi_n$ with basis $\mathcal{B} = \{f_0, \dots, f_n\}$ and set $\mathcal{Y}_0 = (f_0^1, \dots, f_n^1)^T$ as the linear interpolants of $(f_0, \dots, f_n)$ on Ω_0 (note that $\mathcal{Y}_0$ can be written in the form $A_0(x)x + b_0(x)$). Then, for $x \in I$,

$$\mathcal{Y}_1(x) = A_1(x)\mathcal{Y}_0(\alpha(x))$$

computes the piecewise linear interpolants of f on Ω_1, where the entries of $A_1(x)$ are the coefficients $(a_{ik}(x))_{i,k=0}^n$ from Example 3.2. By iterating, for $L \geq 2$, $\mathcal{Y}_L = A_L(x)\mathcal{Y}_{L-1}(\alpha(x))$ computes the piecewise linear interpolant, $\left(f_0^L, \dots, f_n^L\right)^T$, of f on Ω_L.

In terms of complexity, one can see that the number of weights of $\mathcal{Y}_L$ is

$$\mathcal{W}(\mathcal{Y}_L) \leq (n+1)^2 L + (n+1),$$

the number of units is $\mathcal{U}(\mathcal{Y}_L) = L(n+1)$, and width $\mathcal{M}(\mathcal{Y}_L) = n+1$.

Example 3.9. Example 3.8 showed how to generate a vector $\mathcal{Y}_L$ which computes the piecewise linear interpolant, $(f_0^L, \dots, f_n^L)^T$, of $f \in \Pi_n$ on Ω_L with basis $\mathcal{B} = \{f_0, \dots, f_n\}$. Let $c_0, \dots, c_n \in \mathbb{R}$, if $\mathcal{Y}_L$ is the cascade network from Example 3.8, let $\mathcal{Y}'_L$ be the cascade network such that $\mathcal{Y}'_L = A'_L(x)\mathcal{Y}_{L-1}(\alpha(x))$ where $A'_L = c^T A_L$, and $c = (c_0, \dots, c_n)^T$ is the vector of the coefficients. Then $\mathcal{Y}'_L = \sum_{i=0}^{n} c_i f_i^L$, with number of weights $\mathcal{W}(\mathcal{Y}'_L) \leq 2(n+1) + (n+1)^2(L-2)$, number of units $\mathcal{U}(\mathcal{Y}'_L) = (n+1)(L-1) + 1$ and width $\mathcal{M}(\mathcal{Y}'_L) = n+1$.

Remark 3.2. *If $\mathcal{Y}_L$ is generated by a depth L CN, then $\mathcal{Y}_\ell$ are also generated by a CN for all $\ell = 1, \dots, L-1$.*

Cascade networks can be combined to form more complex cascade networks. In fact, Lemmas 2.7-2.10 for ReLU neural networks can be generalized to cascade networks.

Definition 3.10. Let $L_1, L_2 \in \mathbb{N}$. Let $\mathcal{Y}^1$ be a CN with $\mathcal{L}(\mathcal{Y}^1) = L_1$ and $\mathcal{Y}^2$ be a CN with $\mathcal{L}(\mathcal{Y}^2) = L_2$ such that the dimension of the output of $\mathcal{Y}^1$ is equal to the dimension of the input of $\mathcal{Y}^2$. Then there exists a CN $\mathcal{Y}$ of depth $\mathcal{L}(\mathcal{Y}) = L_1 + L_2$, called the *concatenation* of $\mathcal{Y}^1$ and $\mathcal{Y}^2$, such that $\mathcal{Y}(x) = (\mathcal{Y}^2 \circ \mathcal{Y}^1)(x)$ for all $x \in I$.

Lemma 3.11. *Let $d, L, W \in \mathbb{N}$. For each L there exists a depth L CN, $\mathcal{Y}_L$, with input vector $\mathcal{Y}_0$ linear in x, such that $\mathcal{Y}_L$ maps $x \mapsto (x, x, \dots, x)^T \in \mathbb{R}^d$.*

Proof. The proof for $L = 1$ is provided. Let $\mathcal{Y}_0(x) = (a_1 x + b_1, \dots, a_W x + b_W)^T$, $a_i, b_i \in \mathbb{R}$ for $i = 1, \dots, W$. Set

$$A_1(x) = \begin{pmatrix} \frac{1}{2a_1} & 0 & \cdots & 0 \\ 0 & \frac{1}{2a_2} & \cdots & 0 \\ \vdots & \cdots & \ddots & \vdots \\ 0\cdots & \cdots & 0 & \frac{1}{2a_W} \end{pmatrix}, \quad x \in [0,1];$$

$$b_1^0 = \left(\frac{-b_1}{2a_1}, \dots, \frac{-b_W}{2a_W} \right)^T, \qquad b_1^1 = \left(\frac{-b_1 + 1 - 2a_1}{2a_1}, \dots, \frac{-b_W + 12a_W}{2a_W} \right)^T.$$

Then, $\mathcal{Y}_1(x) = A_1(x)\mathcal{Y}_0(\alpha(x)) + b_1(x) = (x, \dots, x)^T$. □

Lemma 3.12. *Let $L, K \in \mathbb{N}$ with $K > L$. Let $\mathcal{Y}_L$ be CN with linear input vector $\mathcal{Y}_0 \in \mathbb{R}^{N_0}$ and $\mathcal{L}(\mathcal{Y}_L) = L$. Then, there exists a CN $\mathcal{Y}_K$ with $\mathcal{L}(\mathcal{Y}_K) = K$, such that $\mathcal{Y}_L(x) = \mathcal{Y}_K(x)$, for all $x \in I$.*

Proof. Use Definition 3.10 and Lemma 3.11 to concatenate $\mathcal{Y}_L$ and the depth $K - L$ CN that generates the identity map. □

Definition 3.13. Let $L \in \mathbb{N}$. Let $\mathcal{Y}_L^1$, $\mathcal{Y}_L^2$ be CN's of depth L with weight matrices A_ℓ, B_ℓ respectively for $\ell = 1, \dots, L$. Define a network $\tilde{\mathcal{Y}}$, called the *parallelization* of $\mathcal{Y}_L^1$ and $\mathcal{Y}_L^2$, such that

$$\tilde{\mathcal{Y}}(x) \coloneqq \left(\mathcal{Y}_L^1(x), \mathcal{Y}_L^2(x) \right),$$

where $\tilde{A}_\ell := \begin{pmatrix} A_\ell & \mathbf{0} \\ \mathbf{0} & B_\ell \end{pmatrix}$ for $l = 1, \ldots, L-1$ and $\tilde{A}_L := \begin{pmatrix} A_L & B_L \end{pmatrix}$.

Lemma 3.14. *Let $L, W \in \mathbb{N}$. Let $\mathcal{Y}^1$, $\mathcal{Y}^2$ be CN's with $\mathcal{L}(\mathcal{Y}^1) = \mathcal{L}(\mathcal{Y}^1) = L$ and with outputs of same size W. Then, there exists a CN, $\tilde{\mathcal{Y}}$, with $\mathcal{L}(\mathcal{Y}) = L$ and satisfying $\mathcal{Y} = \mathcal{Y}^1 + \mathcal{Y}^2$.*

Proof. By Definition 3.3,

$$\mathcal{Y}^1_L(x) = A_L(x)\mathcal{Y}^1_{L-1}(\alpha(x)),$$
$$\mathcal{Y}^2_L(x) = B_L(x)\mathcal{Y}^2_{L-1}(\alpha(x)).$$

Let $C = \begin{pmatrix} A_L & B_L \end{pmatrix}$, then

$$\begin{aligned} \tilde{\mathcal{Y}}(x) &= C \begin{pmatrix} \mathcal{Y}^1_{L-1}(\alpha(x)) \\ \mathcal{Y}^2_{L-1}(\alpha(x)) \end{pmatrix} \\ &= A_L\mathcal{Y}^1_{L-1}(x) + B_L\mathcal{Y}^2_{L-1}(x). \end{aligned}$$

□

In the following lemmas, another way of combining cascade networks, called "splicing", is described.

Definition 3.15. Let g_1, g_2 be functions on I. Call the function $g : I \to \mathbb{R}$ given by

$$g(x) = \begin{cases} g_1(2x), & x \in \left[0, \frac{1}{2}\right) \\ g_2(2x-1), & x \in \left[\frac{1}{2}, 1\right] \end{cases} \tag{3.7}$$

the splicing of g_1, g_2.

Lemma 3.16. *Let g_1, g_2 be real-valued functions on I whose values can be generated by cascade networks $\mathcal{Y}^1_{L_1}, \mathcal{Y}^2_{L_2}$ of depths L_1 and L_2 respectively. Let g be the splicing of g_1 and g_2, then g can* *be generated by a cascade network $\mathcal{Y}_L$ such that*

$$\mathcal{Y}_L(x) = \begin{cases} \mathcal{Y}^1_{L_1}(2x), & x \in \left[0, \frac{1}{2}\right) \\ \mathcal{Y}^2_{L_2}(2x-1), & x \in \left[\frac{1}{2}, 1\right] \end{cases}. \tag{3.8}$$

Moreover, $\mathcal{Y}_L$ has depth

$$\mathcal{L}(\mathcal{Y}_L) = \max\{L_1, L_2\} + 1;$$

number of units

$$\mathcal{U}(\mathcal{Y}_L) = \mathcal{U}(\mathcal{Y}^1_{L_1}) + \mathcal{U}(\mathcal{Y}^2_{L_2}) + |L_2 - L_1| + 1;$$

number of weights

$$\mathcal{W}(\mathcal{Y}_L) = \mathcal{W}(\mathcal{Y}^1_{L_1}) + \mathcal{W}(\mathcal{Y}^2_{L_2}) + |L_2 - L_1| + 2;$$

and width

$$\mathcal{M}(\mathcal{Y}_L) = \mathcal{M}(\mathcal{Y}^1_{L_1}) + \mathcal{M}(\mathcal{Y}^2_{L_2}).$$

Corollary 3.3. *Let g be obtained as the splicing of $g_1, \ldots, g_N$, where $N = 2^M$. Then, $g(x) = g_i(Nx - i + 1)$ for $x \in [\frac{i-1}{N}, \frac{i}{N}]$, $i = 1, \ldots, N$. If $\mathcal{Y}^1_J, \ldots, \mathcal{Y}^N_J$ are the cascade networks for $g_1, \ldots, g_N$ respectively, all with depth J, number of units $\mathcal{U}(\mathcal{Y}^1_J) = \cdots = \mathcal{U}(\mathcal{Y}^N_J)$, number of weights $\mathcal{W}(\mathcal{Y}^1_J) = \cdots = \mathcal{W}(\mathcal{Y}^N_J)$, and width $\mathcal{M}(\mathcal{Y}^i_J)$, $i = 1, \ldots, N$, then g can be realized as a cascade network $\mathcal{Y}_L$, with depth*

$$\mathcal{L}(\mathcal{Y}_L) = J + M;$$

number of units

$$\mathcal{U}(\mathcal{Y}_L) = \mathcal{O}(N\mathcal{U}(\mathcal{Y}^1_J));$$

number of weights

$$\mathcal{W}(\mathcal{Y}_L) = \mathcal{O}(N\mathcal{W}(\mathcal{Y}^1_J));$$

and width

$$\mathcal{M}(\mathcal{Y}_L) = N\mathcal{M}(\mathcal{Y}_J^1).$$

Proof. The proof follows directly from Lemma 3.16. $\square$

Proposition 3.17. Any piecewise linear function on a binary partition, $\Omega_L = \{0, 2^{-L}, \ldots, 1\}$, can be generated by a cascade network $\mathcal{Y}_L$ of depth $\mathcal{L}(\mathcal{Y}_L) = L$, number of units $\mathcal{U}(\mathcal{Y}_L) = \mathcal{O}(2^L)$, number of weights $\mathcal{W}(\mathcal{Y}_L) = \mathcal{O}(2^L)$, and width $\mathcal{M}(\mathcal{Y}_L) = \mathcal{O}(2^L)$.

Proof. The proof will proceed by induction. First, a cascade network $\mathcal{Y}_1$ on $\Omega_1 = \{0, \frac{1}{2}, 1\}$ that generates an arbitrary piecewise linear function on Ω_1 can be written as

$$y(x) = \begin{cases} 2(u_1 - u_0)x + u_0, & x \in \left[0, \frac{1}{2}\right) \\ 2(u_2 - u_1')(x-1) + u_2, & x \in \left[\frac{1}{2}, 1\right] \end{cases},$$

where u_0, u_1, u_1', u_2 are real numbers.

Let $\mathcal{Y}_0(x) = (1, x)^T$. Then, it can be verified that for

$$A_1(x) = \begin{cases} \begin{pmatrix} u_0 & u_1 - u_0 \\ 0 & 0 \end{pmatrix}, & x \in \left[0, \dfrac{1}{2}\right) \\ \begin{pmatrix} u_1' & u_2 - u_1' \\ 0 & 0 \end{pmatrix}, & x \in \left[\dfrac{1}{2}, 1\right] \end{cases},$$

$\mathcal{Y}_1(x) = A_1(x)\mathcal{Y}_0(\alpha(x))$ is a depth 1 cascade network such that $y(x) = \mathcal{Y}_1(x)$.

The induction step now follows directly from Lemma 3.16. $\square$

Remark 3.4. *Note that continuity of the piecewise linear function is not required in this proposition. In addition, the result of Proposition 3.17 implies that all piecewise constant functions on a binary partition can be generated by a cascade network.*

By Proposition 3.17, it is clear that cascade networks also possess the universal approximation property. Indeed, every piecewise linear function on a dyadic mesh can be generated by a CN, and every continuous function can be approximated arbitrarily well by such piecewise linear functions.

Chapter 4

Cascade Networks and Subdivision Schemes

In this chapter the connection between cascade networks and subdivision schemes is discussed. A large class of CN's can be obtained by means of subdivision schemes. A subdivision scheme is an iterative method for constructing a smooth object from discrete data points [39]. Starting with an initial set of data, subdivision schemes iteratively generate a sequence of denser sets of data by repeatedly applying a refinement rule. Subdivision schemes are used in computer aided geometric design (CAGD) and geometric modeling for the design of smooth curves and surfaces, and have also found applications in multi-resolution analysis and wavelets [40, 41].

4.1 Background on Subdivision Schemes

The most studied subdivision schemes are *scalar* and *stationary*. Stationary schemes are characterized by repeatedly applying the same local refinement rule throughout the recursive process. Given an *initial* bi-infinite sequence of points (also referred to as a *control sequence*), $\mathbf{c}^{[0]} = \left\{c_i^{[0]} \mid c_i^{[0]} \in \mathbb{R},\ i \in \mathbb{Z}\right\}$, and a finitely-supported sequence, called a *mask*, $\mathbf{a} = \{a_i \mid a_i \in \mathbb{R},\ i \in \mathbb{Z}\}$ (i.e $a_i \neq 0$ for finitely many i), a *scalar, stationary subdivision scheme* transforms $\mathbf{c}^{[0]}$ into a bi-infinite sequence of points, $\mathbf{c}^{[1]} = \left\{c_i^{[1]} \mid c_i^{[1]} \in \mathbb{R}, i \in \mathbb{Z}\right\}$, by the subdivision rule

$$c_i^{[1]} := \sum_{j\in\mathbb{Z}} a_{i-2j} c_j^{[0]}. \tag{4.1}$$

The subdivision rule (4.1) can be viewed as two separate rules, one rule for the even indices and one for the even indices:

$$c_{2i}^{[1]} := \sum_{j\in\mathbb{Z}} a_{2j} c_{i-j}^{[0]}, \quad c_{2i+1}^{[1]} := \sum_{j\in\mathbb{Z}} a_{2j+1} c_{i-j}^{[0]}.$$

A subdivision scheme is the repeated application of the refinement rule (4.1) starting from the initial sequence $\mathbf{c}^{[0]}$. After m applications of the subdivision rule, the result is a bi-infinite sequence of points $\mathbf{c}^{[m]} = \left\{ c_i^{[m]} \mid c_i^{[m]} \in \mathbb{R}, i \in \mathbb{Z} \right\}$.

It turns out that $\mathbf{c}^{[m]}$ often converges, in the sense defined below to a well-defined function. To make this precise, a subdivision scheme can be viewed as one application of a linear operator, $\mathbf{S}_a$, which maps a bi-infinite sequence $\mathbf{c}$ into the bi-infinite sequence $\mathbf{S}_a(\mathbf{c})_i := \sum_{j\in\mathbb{Z}} a_{i-2j} c_j$.

Definition 4.1. [42] A stationary subdivision scheme is called *convergent* if for every $\mathbf{c}^{[0]}$, there exists a continuous f on $\mathbb{R}$ such that

$$\lim_{m\to\infty} \sup_{j\in\mathbb{Z}} \left| \left(\mathbf{S}_a^m \mathbf{c}\right)_j - f\left(2^{-m} j\right) \right| = 0,$$

and such that $f \not\equiv 0$ for at least one initial sequence $\mathbf{c}^{[0]} \not\equiv 0$. Here $\mathbf{S}_a^m$ denotes the m-fold application of $\mathbf{S}_a$. The limit function f is denoted by $\mathbf{S}_a^\infty\left(\mathbf{c}^{[0]}\right)$.

If $\mathbf{c}^{[0]} = \boldsymbol{\delta}$, where $\boldsymbol{\delta} = \{\delta_{i,0}, i \in \mathbb{Z}\}$ is the delta sequence, then $\phi_{\mathbf{a}} := \mathbf{S}_a^\infty(\boldsymbol{\delta})$ is called the *basic limit function* of the subdivision scheme. By the linearity of $\mathbf{S}_a$,

$$f = \mathbf{S}_a^\infty\left(\mathbf{c}^{[0]}\right) = \sum_{j\in\mathbb{Z}} c_j^{[0]} \phi_{\mathbf{a}}(\cdot - j),$$

for any initial sequence $\mathbf{c}^{[0]}$.

If $\phi_a \in C^\ell(\mathbb{R})$, for some $\ell \geq 0$, then so is any limit function generated by $\mathbf{S}_a$ and the scheme is said to be C^ℓ. Thus, the smoothness of the of the scheme is determined by the smoothness of the basic limit function.

A subdivision scheme is called *interpolatory* if for all m, $\mathbf{c}^{[m]}$ is "contained in" $\mathbf{c}^{[m+1]}$ in the sense that $c_{2i}^{[m+1]} = c_i^{[m]}$. In this case, the limit function f interpolates the input points, $f(j) = c_j^{[0]}, j \in \mathbb{Z}$. Other types of schemes are called *approximating*.

Subdivision schemes are connected to wavelets by the so-called refinement equation and the refinability of the basic limit function. Namely, the basic limit function, $\phi_{\mathbf{a}} = \mathbf{S}_a^\infty(\boldsymbol{\delta})$, satisfies the refinement equation

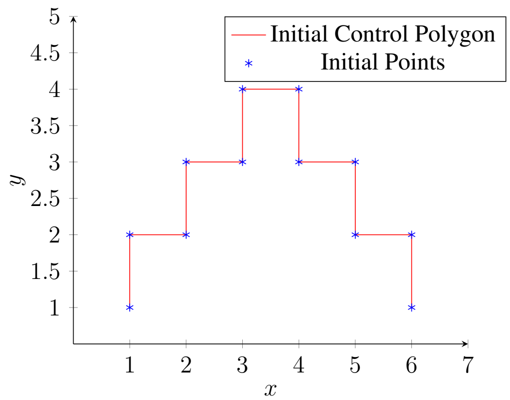
$$\phi_{\mathbf{a}} = \sum_{j\in\mathbb{Z}} a_j \phi_{\mathbf{a}}(2\cdot -j), \tag{4.2}$$

where $\{a_i \mid a_i \in \mathbb{R}, i \in \mathbb{Z}\}$ are the coefficients of the mask $\mathbf{a}$. It is well known that if the mask $\mathbf{a} = \{a_i \mid a_i \in \mathbb{R}, i \in \mathbb{Z}\}$ is supported on $\{0,\ldots,n\}$, i.e $a_i = 0$ if $i \notin \{0,\ldots,n\}$, then ϕ_a is supported on $[0,n]$.

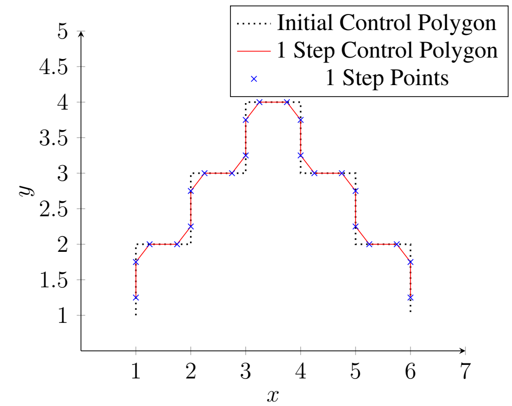
Example 4.2. [39, 43] An example of a scalar, stationary subdivision scheme is the Chaikin algorithm. The Chaikin algorithm was one of the first "corner cutting algorithms" used to generate a curve from a set of control points. Corner cutting algorithms iteratively create new, smoother curves from a set of control points. Starting with initial sequence $\mathbf{c}^{[0]} = \left\{c_i^{[0]} \mid c_i^{[0]} \in \mathbb{R},\ i \in \mathbb{Z}\right\}$, the Chaikin algorithm is based on the rules

$$c_{2i}^{[m+1]} = \frac{1}{4}c_{i-1}^{[m]} + \frac{3}{4}c_i^{[m]}, \quad c_{2i+1}^{[m+1]} = \frac{3}{4}c_i^{[m]} + \frac{1}{4}c_{i+1}^{[m]}, \ \ i \in \mathbb{Z},$$

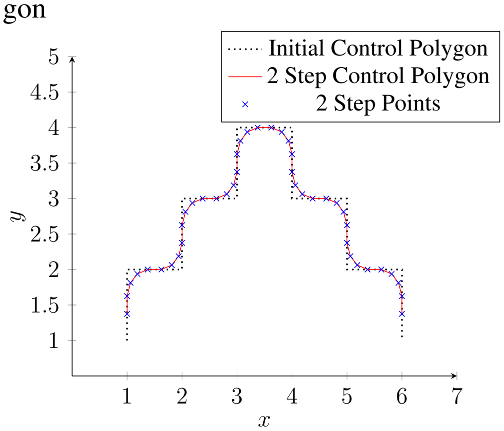
corresponding to the mask $\mathbf{a}$ with nonzero entries $a_0 = \frac{1}{4}, a_1 = \frac{3}{4}, a_2 = \frac{3}{4}, a_3 = \frac{1}{4}$. The subdivision algorithm in this case can be interpreted as follows: for a given initial control "polygon" (the piecewise linear interpolant of a set of points in $\mathbb{R}^2$) one generates new points by chopping off the corners of this polygon. Namely, if A, B are a pair of neighboring points of this polygon, the Chaikin algorithm finds two points C, D such that C is $1/4$ of the way between A and B and D is $3/4$ of the way between A and B. The new points C, D obtained for all linear segments of the control polygon now form a smoother path. Chaikin's algorithm converges to a quadratic spline curve $\sum c_i^{[0]} B_2(\cdot - i)$, where B_2 are quadratic B-splines. Figure 4.1a shows a set of initial points with a simple initial control polygon. Figures 4.1b, 4.1c, and 4.1d show one, two and three iterations respectively of Chaikin's algorithm on the simple initial control polygon.



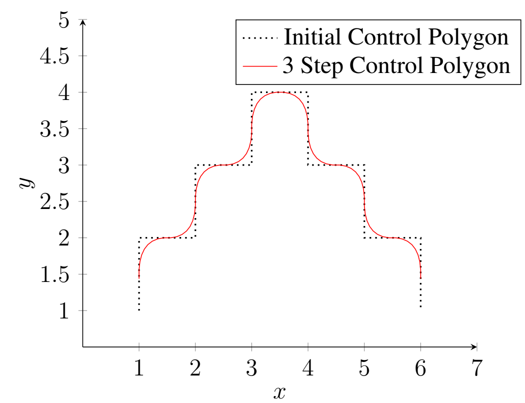
(a) Initial Points and Control Polygon



(b) 1 Step of the Chaikin Algorithm



(c) 2 Steps of the Chaikin Algorithm



(d) 3 Steps of the Chaikin Algorithm

Figure 4.1: Chaikin Algorithm

Subdivision schemes can also be implemented in a level-dependent way by using a different mask at each iteration, giving rise to non-stationary subdivision schemes [39, 44]. Given an initial sequence of points $\mathbf{c}^{[0]}$ and a sequence of masks $\{\mathbf{a}^{[m]}\}_{m\geq 0}$, a *non-stationary* subdivision scheme gives a new sequence of points

$$c_i^{[m+1]} := \sum_{j\in\mathbb{Z}} a_{i-2j}^{[m]} c_j^{[m]}. \tag{4.3}$$

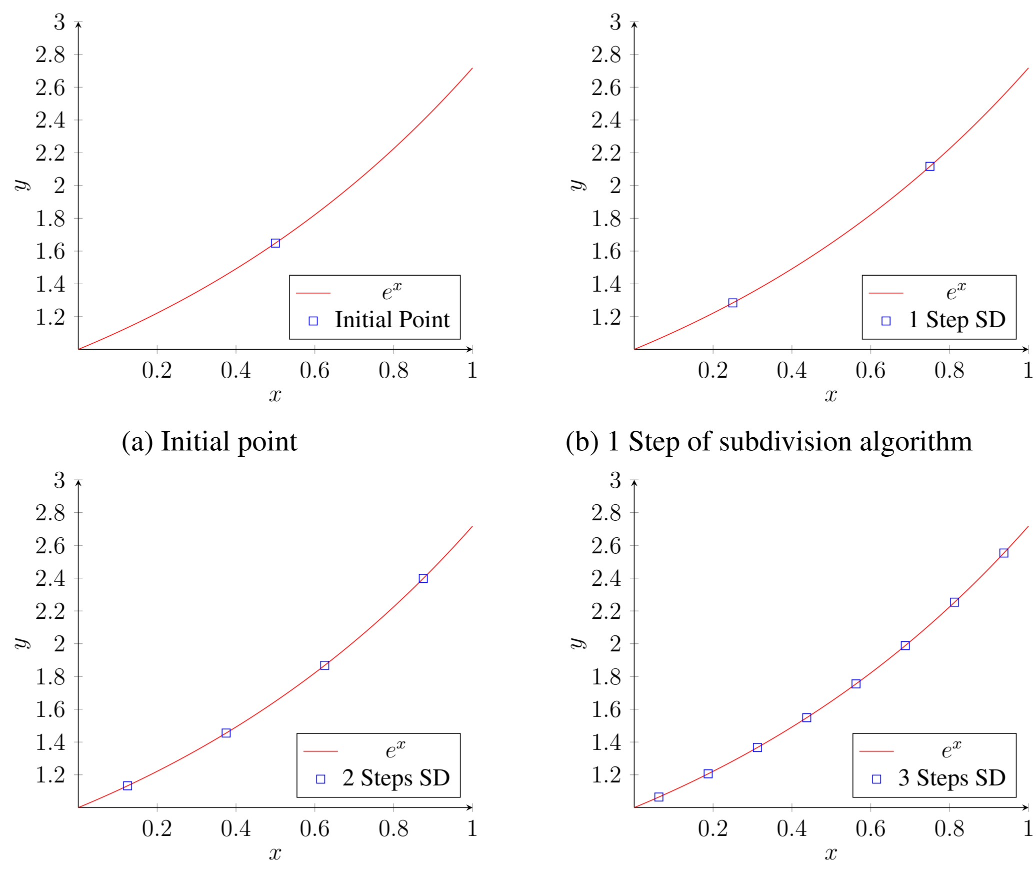
Non-stationary schemes are as easily implemented as stationary ones, as, in practice, few iterations are performed for both non-stationary and stationary subdivision schemes. The definition of convergence for non-stationary schemes is similar to stationary schemes. However, non-stationary schemes have a larger class of limit functions. In particular, non-stationary schemes can generate exponential splines and even C^∞ functions with compact support [45].

Example 4.3. The exponential function, $y = e^x$, can be obtained as the limit of a scalar, non-stationary subdivision scheme. Let $\mathbf{c}^{[0]} = \{\dots, 0, e^{1/2}, 0, \dots\}$ be the given the initial sequence. Then, $\mathbf{c}^{[m]} = \left(c_i^{[m]}\right)_{i\in\mathbb{Z}}$ are associated with the dyadic points $\{(2i+1)2^{-m-1}\}_{i\in\mathbb{Z}}$, and are generated according to the rule

$$c_i^{[m+1]} := \sum_{j\in\mathbb{Z}} a_{i-2j}^{[m]} c_j^{[m]}.$$

The masks at level m of the scheme are given by

$$\mathbf{a}^{[m]} = \left\{\dots, 0, a_0^{[m]}, a_1^{[m]}, 0, \dots\right\}_{m\geq 0} = \left\{\dots, 0, e^{\frac{-1}{2^{m+2}}}, e^{\frac{1}{2^{m+2}}}, 0, \dots\right\}_{m\geq 0}.$$



(a) Initial point

(b) 1 Step of subdivision algorithm

(c) 2 Steps of subdivision algorithm

(d) 3 Steps of subdivision algorithm

Figure 4.2: Subdivision Scheme Generating the Exponential Function.

Figure 4.2a shows the graph of the exponential function and the initial point. Figures 4.2b, 4.2c, and 4.2d show the points generated by one, two and three steps respectively of the subdivision scheme. Clearly, as $m \to \infty$, this scheme generates the exponential function.

Given a sequence of masks $\{\mathbf{a}^{[m]}\}_{m \geq 0}$, starting with any $\mathbf{a}^{[m_0]},\ m_0 > 0$, a non-stationary scheme yields different results depending on the starting mask $\mathbf{a}^{[m_0]}$. Therefore, there is no unique basic limit function in the non-stationary case, rather there is a sequence of basic limit functions, $\{\phi_{m_0}\}_{m_0 > 0}$, each defined by

$$\phi_{m_0} = \lim_{m \to \infty} \mathbf{S}_{\mathbf{a}^{[m+m_0]}} \cdots \mathbf{S}_{\mathbf{a}^{[m_0]}} \boldsymbol{\delta},$$

where $\boldsymbol{\delta}$ is the delta sequence. This sequence of basic limit functions satisfies a system of generalized refinement equations

$$\phi_{m_0} = \sum_{j \in \mathbb{Z}} a_j^{[m_0]} \phi_{m_0+1}(2 \cdot -j),\ m_0 \geq 0. \tag{4.4}$$

Similar to scalar subdivision, one can define a *vector* subdivision scheme starting with an initial bi-infinite sequence of column vectors $\mathbf{c}^{[0]} = \left\{c_i^{[0]} \mid c_i^{[0]} \in \mathbb{R}^k, i \in \mathbb{Z}\right\}$, and a mask $\mathbf{A} = \left\{A_i \mid A_i \in \mathbb{R}^{k \times k}, i \in \mathbb{Z}\right\}$. A vector, stationary subdivision scheme generates a new bi-infinite sequence of (column) vectors, $\mathbf{c}^{[1]} = \left\{c_i^{[1]} \mid c_i^{[1]} \in \mathbb{R}^k, i \in \mathbb{Z}\right\}$, by the refinement rule

$$c_i^{[1]} := \sum_{j \in \mathbb{Z}} A_{i-2j} c_j^{[0]}. \tag{4.5}$$

Viewing $\mathbf{c}^{[0]}$ as a row vector, the subdivision refinement rule is defined by:

$$c_i^{[1]} := \sum_{j \in \mathbb{Z}} c_j^{[0]} A_{i-2j}^T. \tag{4.6}$$

After m iterations, a new bi-infinite sequence of vectors $\mathbf{c}^{[m]} = \left\{c_i^{[m]} \mid c_i^{[m]} \in \mathbb{R}^k, i \in \mathbb{Z}\right\}$ is obtained, where

$$c_i^{[m]} := \sum_{j\in\mathbb{Z}} A_{i-2j} c_j^{[m-1]}. \tag{4.7}$$

One can also define non-stationary vector subdivision schemes using a different mask $\mathbf{A}^{[m]} = \left\{A_i^{[m]} \mid A_i^{[m]} \in \mathbb{R}^{k\times k}, i \in \mathbb{Z}\right\}$ at each level $m \geq 0$, giving a new sequence of points $\mathbf{c}^{[m+1]} = \left\{c_i^{[m+1]}, i \in \mathbb{Z}\right\}$, where

$$c_i^{[m]} := \sum_{j\in\mathbb{Z}} A_{i-2j}^{[m-1]} c_j^{[m-1]}. \tag{4.8}$$

Example 4.4. [42, 46] The classical Hermite-cubic interpolation is an example of a vector, non-stationary subdivision scheme. The scheme produces the Hermite cubic interpolant of the given initial data, consisting of function values and corresponding derivative values.

Let $\mathbf{c}^{[0]} = \left\{\ldots, \mathbf{0}, c_i^{[0]} = \left(f_i^{[0]}, g_i^{[0]}\right)^T, \mathbf{0}, \ldots\right\}_{i\in\mathbb{Z}}$ be an initial sequence, where $\left\{f_i^{[0]}\right\}$ represents function values and $\left\{g_i^{[0]}\right\}$ represents derivative data and define the masks at level m by

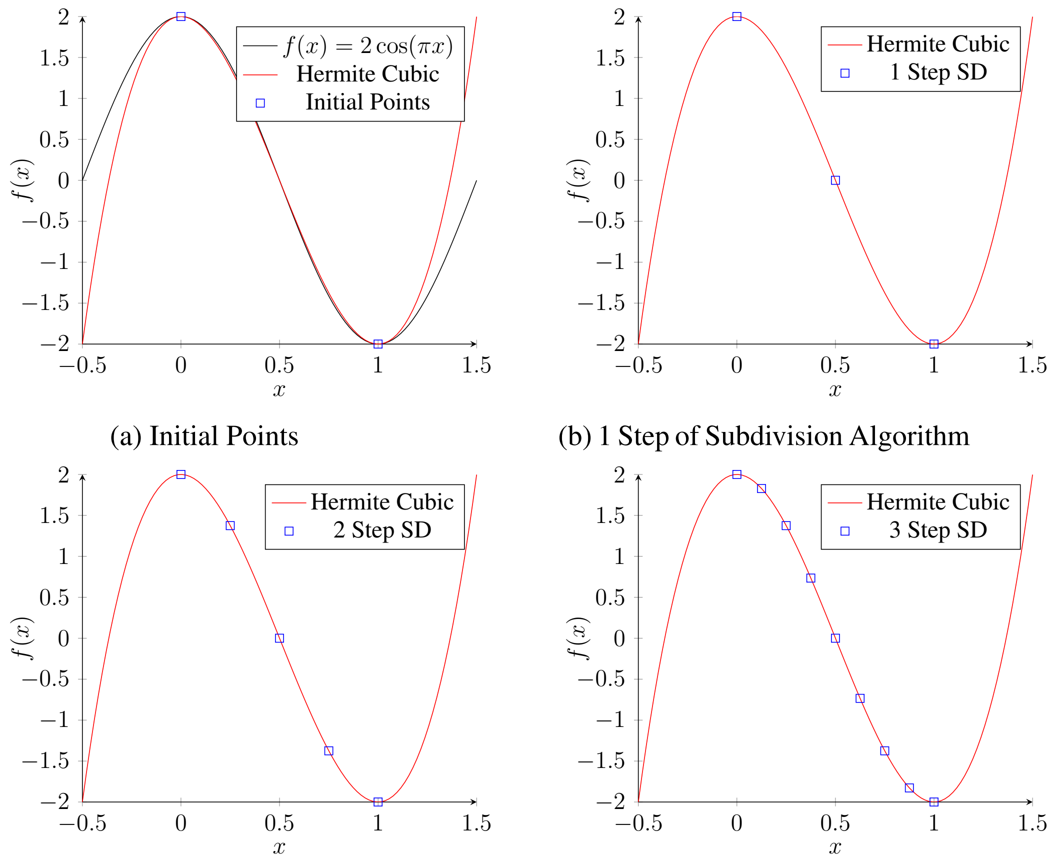
$$\mathbf{A}^{[m]} = \left\{\ldots, \mathbf{0}, A_0^{[m]}, A_1^{[m]}, A_2^{[m]}, \mathbf{0}, \ldots\right\}_{m\geq 0},$$

where

$$A_0^m = \begin{pmatrix} \frac{1}{2} & \frac{-1}{8}2^{-m} \\ \frac{3}{2}2^m & \frac{-1}{4} \end{pmatrix}, \qquad A_1^{[m]} = \mathbb{I}_2, \qquad A_2^m = \begin{pmatrix} \frac{1}{2} & \frac{1}{8}2^{-m} \\ \frac{-3}{2}2^m & \frac{-1}{4} \end{pmatrix}.$$

Then, $\mathbf{c}^{[m]} = \left\{\ldots, \mathbf{0}, c_i^{[m]} = \left(f_i^{[m]}, g_i^{[m]}\right)^T \mathbf{0}, \ldots\right\}_{i\in\mathbb{Z}}$ converges to, as $m \to \infty$, the Hermite cubic interpolant of the initial sequence $\mathbf{c}^{[0]}$. Associating $c_i^{[m]}$ with the dyadic points $i2^{-m}, i \in \mathbb{Z}$ one can show that $\left(c_i^{[m]}\right) \to h\left(i2^{-m}\right), i \in \mathbb{Z}$ as $m \to \infty$, where h is the Hermite cubic function interpolating the initial data. That is, $h(i) = f_i^{[0]}, h'(i) = g_i^{[0]}$ for $i \in \mathbb{Z}$.

Consider the initial sequence $\mathbf{c}^{[0]} = \left\{\ldots, (0,0)^T, (2,0)^T, (-2,0)^T, (0,0)^T, \ldots\right\}$ which gives the function and derivative values at $x = 0$ and $x = 1$ of $f(x) = 2\cos(\pi x)$. Figure 4.3a shows the initial sequence of points $\mathbf{c}^{[0]}$, the graph of f and the graph of the Hermite cubic polynomial through the initial sequence of points. Figures 4.3b- 4.3d show the points generated by $1, 2$, and 3 steps respectively of the subdivision algorithm which generates the Hermite Cubic interpolant of the initial sequence.



(a) Initial Points

(b) 1 Step of Subdivision Algorithm

(c) 2 Steps of Subdivision Algorithm

(d) 3 Steps of Subdivision Algorithm

Figure 4.3: Hermite Cubic Interpolation Subdivision Scheme

4.2 Reformulation of Subdivision Schemes as Cascade Networks

This section describes how subdivision schemes can be reformulated in terms of cascade networks. First, let us revisit Example 4.3 of the exponential function.

Example 4.5. By Example 4.3, the function $y = e^x$ can be reproduced by a scalar, non-stationary subdivision scheme with initial sequence $\mathbf{c}^{[0]} = \{\dots, 0, e^{1/2}, 0, \dots\}$ and masks $\mathbf{a}^{[m]} = \left\{\dots, 0, e^{\frac{-1}{2^{m+2}}}, e^{\frac{1}{2^{m+2}}}, 0, \dots\right\}_{m\geq 0}$.

It can also be easily shown that $\lim_{L\to\infty} \mathcal{Y}_L = e^x$, where

$$\mathcal{Y}_L = c_0^{[0]} A_L (A_{L-1} \circ \alpha) \cdots (A_1 \circ \alpha^{L-1})(\mathcal{Y}_0 \circ \alpha^L)$$

is a cascade network of width $\mathcal{M}(\mathcal{Y}_L) = 1$, $\mathcal{Y}_0 = 1$, $c_0^{[0]} = e^{1/2}$ and

$$A_\ell(x) = \begin{cases} e^{\frac{-1}{2^{\ell+1}}}, \ x \in [0, 1/2) \\ \\ e^{\frac{1}{2^{\ell+1}}}, \ x \in [1/2, 1] \end{cases}, \ \ell = 1, \ldots, L.$$

The above example is generalized in the following results.

Proposition 4.6. Let $m \in \mathbb{N}_0$. Consider a scalar, non-stationary subdivision scheme with initial sequence $\mathbf{c}^{[0]} = \left\{\ldots, 0, c_0^{[0]}, 0, \ldots\right\}$, masks $\mathbf{a}^{[m]} = \left\{\ldots, 0, a_0^{[m]}, a_1^{[m]}, 0, \ldots\right\}_{m\geq 0}$ supported on I, and refinement rule 4.1.

Define $g_0 \coloneqq c_0^{[0]}$. For $m \in \mathbb{N}$, $x \in I$, let

$$g_m(x) = c_0^{[0]} A_0(x) A_1(\alpha(x)) \cdots A_{m-1}(\alpha^{m-1}(x)) \mathcal{Y}_0(\alpha^m(x)),$$

where $\mathcal{Y}_0(x) = 1$, and

$$A_k(x) = \begin{cases} a_0^{[k]}, & x \in \left[0, \frac{1}{2}\right) \\ \\ a_1^{[k]}, & x \in \left[\frac{1}{2}, 1\right] \end{cases} \quad \text{for } k = 0, \ldots, m-1.$$

Then, g_m is the right-continuous, piecewise constant interpolant of $\left(c_i^{[m]}\right)_{i=0}^{2^m-1}$ obtained by subdivision at the mesh points $\Omega_m = \{0, \ldots, 1-2^{-m}\}$, i.e $g_m\left(i2^{-m}\right) = c_i^{[m]}$ for $i = 0, \ldots, 2^m-1$.

Proof. First, we prove that

$$c_0^{[0]} a_{\delta_m}^{[0]} \cdots a_{\delta_1}^{[m-1]} = c_{2^m x_j}^{[m]} = g_m(x), \tag{4.9}$$

for $x \in [x_j, x_{j+1}]$, $j = 0, \ldots, 2^m - 1$, where $x_j = 0.\delta_m \ldots \delta_1$. is its binary representation. The proof proceeds by induction: If $m = 1$, then by the subdivision algorithm

$$c_0^{[0]} a_0^{[0]} = c_0^{[1]} \quad c_0^{[0]} a_1^{[0]} = c_1^{[1]}$$

and

$$g_1(x) = \begin{cases} c_0^{[0]} a_0^{[0]} & x \in [0, \frac{1}{2}) \\ c_0^{[0]} a_1^{[0]} & x \in [\frac{1}{2}, 1] \end{cases}$$

is the piecewise constant interpolant of $\mathbf{c}^{[0]}$ on Ω_1.

Next, suppose equation 4.9 is true for some $m > 1$. By the induction hypothesis,

$$c_0^{[0]} a_{\delta_{m+1}}^{[0]} \cdots a_{\delta_1}^{[m]} = c_{2^m x_j}^{[m]} a_{\delta_1}^{[m]}.$$

By the subdivision algorithm,

$$c_j^{[m]} a_0^{[m]} = c_{2j}^{[m+1]} \quad c_j^{[m]} a_1^{[m]} = c_{2j+1}^{[m+1]}$$

$j = \delta_{m+1} \cdots \delta_2$ Thus,

$$c_0^{[0]} a_{\delta_{m+1}}^{[0]} \cdots a_{\delta_1}^{[m]} = c_{2^{m+1} x_j}^{[m+1]} = c_k^{[m+1]},$$

where $k = 2j + 1$. Clearly, $g_m\left(i 2^{-m}\right) = c_i^{[m]}$ for $i = 0, \ldots, 2^m - 1$.

$\square$

For masks of larger support than $[0, 1]$, consider piecewise linear interpolants of the control points $\mathbf{c}^{[m]}$.

Proposition 4.7. Let $n \geq 2$, $m \in \mathbb{N}_0$. Consider a scalar, non-stationary subdivision scheme with initial sequence $\mathbf{c}^{[0]} = \left\{\ldots, 0, c_0^{[0]}, \ldots, c_{n-1}^{[0]}, 0, \ldots\right\}$, masks $\mathbf{a}^{[m]} = \{\ldots, 0, a_0^{[m]}, \ldots, a_n^{[m]}, 0, \ldots\}$ supported on $[0, n]$.

For $m \in \mathbb{N}, x \in I$, let

$$g_m(x) \coloneqq (c_0^{[0]}, \dots, c_{n-1}^{[0]}) A_0(x) A_1(\alpha(x)) \cdots A_{m-1}(\alpha^{m-1}(x))(\mathcal{Y}_0(\alpha^m(x)),$$

where $\mathcal{Y}_0(x) \coloneqq (1-x, x, 0, \dots, 0)^T \in \mathbb{R}^n$ and $A_k \in \mathbb{R}^{n\times n}$ such that

$$\begin{aligned} A_k(x) &= \begin{cases} A^{0,[k]}, & x \in \left[0, \frac{1}{2}\right) \\ A^{1,[k]}, & x \in \left[\frac{1}{2}, 1\right] \end{cases} \\ &= \begin{cases} \left(a_{j-2i+n-1}^{[k]}\right)_{i,j=0}^{n-1}, & x \in \left[0, \frac{1}{2}\right) \\ \left(a_{j-2i+n}^{[k]}\right)_{i,j=0}^{n-1}, & x \in \left[\frac{1}{2}, 1\right] \end{cases}, \end{aligned}$$

for $k = 0, \dots, m$.

Then, g_m is the piecewise linear interpolant of $\left(c_i^{[m]}\right)_i$ for $i = (2^m - 1)(n-1), \dots, (2^m - 1)(n-1) + 2^m$, obtained by the subdivision at mesh points $\Omega_m = \{0, 2^{-m}, \dots, 1\} = \{x_0, \dots, x_{2^m}\}$.

Proof. The proof is slightly more general than the proof of Proposition 4.6. We first prove that

$$\left(c_0^{[0]}, \dots, c_{n-1}^{[0]}\right) A^{\delta_m,[0]} \cdots A^{\delta_1,[m-1]} = \left(c_{(2^m-1)(n-1)+2^m x_j}^{[m]}, \dots, c_{(2^m-1)(n-1)+2^m x_{j+n-1}}^{[m]}\right) \quad (4.10)$$

if $x \in [x_j, x_{j+1}],\ j = 0, \dots 2^m - 1$, where $x_j = 0.\delta_m \cdots \delta_1$, $\delta_s \in \{0, 1\}$, $s = 1, \dots, m$ is its binary representation. The proof now proceeds by induction:

For $m = 1$, by the definition of $A^{0,[0]}$, and looking at $\mathbf{c} = \left(c_0^{[0]}, \dots, c_{n-1}^{[0]}\right)$ multiplied by the columns of $A^{0,[0]}$:

$$\begin{aligned} \mathbf{c}\left(a_{-2i+n-1}^{[0]}\right)_{i=1}^{n-1} &= \left(c_0^{[0]}, \dots, c_{n-1}^{[0]}\right)\left(a_{n-1}^{[0]}, a_{n-3}^{[0]}, \dots, a_{-n+1}^{[0]}\right)^T \\ &= c_{n-1}^{[1]}, \end{aligned}$$

$$
\begin{aligned}
\mathbf{c}(a^{[0]}_{-2i+n})^{n-1}_{i=1} &= \left(c^{[0]}_0, \ldots, c^{[0]}_{n-1}\right)(a^{[0]}_n, a^{[0]}_{n-2}, \ldots, a^{[0]}_{-n+2})^T \\
&= c^{[1]}_n
\end{aligned}
$$

Continuing in this manner yields:

$$
\mathbf{c}A^{0,[0]} = \left(c^{[1]}_{n-1}, c^{[1]}_n, \ldots, c^{[1]}_{2n-2}\right).
$$

Similarly, one can show that

$$
\mathbf{c}A^{1,[0]} = \left(c^{[1]}_n, \ldots, c^{[1]}_{2n-1}\right).
$$

Next, suppose equation (4.10) is true for some $m > 1$. By the induction hypothesis,

$$
\left(c^{[0]}_0, \ldots, c^{[0]}_{n-1}\right) A^{\delta_{m+1},[0]} \cdots A^{\delta_1,[m]} = \left(c^{[m]}_{(2^m-1)(n-1)+2^m x_j}, \ldots, c^{[m]}_{(2^m-1)(n-1)+2^m x_{j+n-1}}\right) A^{\delta_1,[m]}.
$$

Then, by looking at $\left(c^{[m]}_{(2^m-1)(n-1)+2^m x_j}, \ldots, c^{[m]}_{(2^m-1)(n-1)+2^m x_{j+n-1}}\right)$ multiplied by the columns of $A^{0,[m]}$,

$$
\begin{aligned}
\left(c^{[m]}_{(2^m-1)(n-1)+2^m x_j}, \ldots, c^{[m]}_{(2^m-1)(n-1)+2^m x_{j+n-1}}\right) A^{0,[m]} = \Big(&c^{[m+1]}_{(2^{m+1}-1)(n-1)+2^{m+1} x_j}, \ldots, \\
&c^{[m+1]}_{(2^{m+1}-1)(n-1)+2^{m+1} x_{j+n-1}}\Big).
\end{aligned}
$$

The product $\left(c^{[m]}_{(2^m-1)(n-1)+2^m x_j}, \ldots, c^{[m]}_{(2^m-1)(n-1)+2^m x_{j+n-1}}\right) A^{1,[m]}$ can be handled similarly.

Now, it remains to show that g_m is the piecewise linear interpolant of $\left(c^{[m]}_i\right)_i$ where $i = (2^m-1)(n-1), \ldots, (2^m-1)(n-1) + 2^m$.

Let $j \in \{0, \ldots, 2^m - 1\}$ and $x \in [x_j, x_{j+1}]$, where $x_j = 0.\delta_m \cdots \delta_1$ is its binary representation.

Then, by equation (4.10)

$$
g_m(x) = \left(c^{[0]}_0, \ldots, c^{[0]}_{n-1}\right) A^{\delta_m,[0]} \cdots A^{\delta_1,[m-1]} l(\alpha^m(x))
$$

$$= \left(c^{[m]}_{(2^m-1)(n-1)+2^m x_j}, \ldots, c^{[m]}_{(2^m-1)(n-1)+2^m x_{j+n-1}}\right) \mathcal{Y}_0(\alpha^m(x)), \tag{4.11}$$

for $\mathcal{Y}_0(\alpha^m(x)) = (\ell_0(x), \ell_1(x), 0, \ldots, 0)^T$, where $\ell_0(x), \ell_1(x) \in \Pi_1$ and $\ell_0(x_j) = 1, \ell_0(x_{j+1}) = 0, \ell_1(x_j) = 0, \ell_1(x_{j+1}) = 1$.

Thus, $g_m(x_j) = c^{[m]}_{(2^m-1)(n-1)+2^m x_j}$ and $g_m(x_{j+1}) = c^{[m]}_{(2^m-1)(n-1)+2^m x_{j+1}}$. Hence, g_m is well-defined because x_j is independent of whether the interval $[x_{j-1}, x_j]$ or $[x_j, x_{j+1}]$ was used.

□

Based on the above, a general scalar, non-stationary subdivision algorithm with mask support $[0, n]$, can be viewed as a special case of the CN algorithm. For every such subdivision scheme, there exists a CN generating a piecewise linear function interpolating the values $\mathbf{c}^{[m]}$ obtained by subdivision.

Example 4.8. The Chaikin algorithm described in Example 4.2 can be reformulated as a CN,

$$\mathcal{Y}_L(x) = \left(c^{[0]}_0, c^{[0]}_1, c^{[0]}_2\right) A_0(x) A_1(\alpha(x)) \cdots A_{L-1}(\alpha^{L-1}(x)) \mathcal{Y}_0(\alpha^L(x))$$

with $\mathcal{Y}_0 = (1-x, x, 0)^T$ and matrices $A_\ell \in \mathbb{R}^{n\times n}$ such that

$$A_\ell(x) = \begin{cases} A^0, & x \in \left[0, \frac{1}{2}\right) \\ A^1, & x \in \left[\frac{1}{2}, 1\right] \end{cases}$$
$$= \begin{cases} \begin{pmatrix} 3/4 & 1/4 & 0 \\ 1/4 & 3/4 & 3/4 \\ 0 & 0 & 1/4 \end{pmatrix}, & x \in \left[0, \dfrac{1}{2}\right) \\ \begin{pmatrix} 1/4 & 0 & 0 \\ 3/4 & 3/4 & 1/4 \\ 0 & 1/4 & 3/4 \end{pmatrix}, & x \in \left[\dfrac{1}{2}, 1\right] \end{cases},$$

for $\ell = 1, \ldots, L$.

The next two results deal with vector subdivision schemes. Again, we will see that they can be cast in the language of CN's. The cases $n = 1$ and $n \geq 2$ will again be treated separately.

Proposition 4.9. Let $k \in \mathbb{N}$ and $m \in \mathbb{N}_0$. Consider a vector, non-stationary subdivision scheme with initial sequence $\mathbf{c}^{[0]} = \left\{\dots, \mathbf{0}, c_0^{[0]}, \mathbf{0} \dots\right\}$, where $c_0^{[0]} \in \mathbb{R}^k$ is a column vector, and masks $\mathbf{A}^{[m]} = \left\{\dots, \mathbf{0}, A_0^{[m]}, A_1^{[m]}, \mathbf{0}, \dots\right\}_{m \geq 0}$ supported on $[0, 1]$, where $A_0^{[m]}, A_1^{[m]} \in \mathbb{R}^{k \times k}$. For $m \in \mathbb{N}, x \in I$, let

$$g_m(x) \coloneqq \left(c_0^{[0]}\right)^T A_0(x) A_1(\alpha(x)) \cdots A_{m-1}(\alpha^{m-1}(x))(\mathcal{Y}_0(\alpha^m(x)),$$

where $\mathcal{Y}_0(x) = \mathbb{I}_k$, and $A_\ell \in \mathbb{R}^{k \times k}$ such that

$$A_\ell(x) = \begin{cases} A^{0,[\ell]}, & x \in \left[0, \frac{1}{2}\right) \\ A^{1,[\ell]}, & x \in \left[\frac{1}{2}, 1\right] \end{cases} = \begin{cases} \left(A_0^{[\ell]}\right)^T, & x \in \left[0, \frac{1}{2}\right) \\ \left(A_1^{[\ell]}\right)^T, & x \in \left[\frac{1}{2}, 1\right] \end{cases},$$

for $\ell = 0, \dots, m$.

If $x \in \left[\frac{i}{2^m}, \frac{i+1}{2^m}\right)$, for $i = 0, \dots, 2^m - 1$, then $g_m(x) = \left(c_i^{[m]}\right)^T$.

Proof. The proof is similar to the proof of Proposition 4.6 □

Proposition 4.10. Let $n, k \in \mathbb{N}$, $n \geq 2$, and $m \in \mathbb{N}_0$. Consider a vector, non-stationary subdivision scheme with initial sequence $\mathbf{c}^{[0]} = \left\{\dots, 0, c_0^{[0]}, \dots, c_{n-1}^{[0]}, 0, \dots\right\}$, where $c_i^{[0]} \in \mathbb{R}^k$ is a column vector, and masks $\mathbf{a}^{[m]} = \left\{\dots, 0, A_0^{[m]}, \dots, A_n^{[m]}, 0, \dots\right\}$ supported on $[0, n]$, $A_i^{[m]} \in \mathbb{R}^{k \times k}$.

For $m \in \mathbb{N}, x \in I$, let

$$g_m(x) \coloneqq \left(\left(c_0^{[0]}\right)^T, \dots, \left(c_{n-1}^{[0]}\right)^T\right) A_0(x) A_1(\alpha(x)) \cdots A_{m-1}(\alpha^{m-1}(x))(\mathcal{Y}_0(\alpha^m(x)),$$

where $A_\ell \in \mathbb{R}^{kn\times kn}$

$$A_\ell(x) = \begin{cases} A^{0,[\ell]}, & x \in \left[0, \frac{1}{2}\right) \\ \\ A^{1,[\ell]}, & x \in \left[\frac{1}{2}, 1\right] \end{cases} = \begin{cases} \left(A^{[\ell]}_{j-2i+n-1}\right)_{i,j=0}^{n-1,n-1} & x \in \left[0, \frac{1}{2}\right) \\ \left(A^{[\ell]}_{j-2i+n}\right)_{i,j=0}^{n-1,n-1}, & x \in \left[\frac{1}{2}, 1\right], \end{cases}$$

for $\ell = 0, \ldots, m$, and $\mathcal{Y}_0(x) \in \mathbb{R}^{kn\times k}$ given by $\mathcal{Y}_0(x) \coloneqq \begin{pmatrix} (1-x)\mathbb{I}_k \\ \\ x\mathbb{I}_k \\ \\ \mathbf{0}_{(n-2)k\times k} \end{pmatrix}$.

Then, g_m is the piecewise linear row vector function of size k that interpolates $\left\{\left(c_i^{[m]}\right)_i^T\right\}$, for $i = (2^m-1)(n-1), \ldots, (2^m-1)(n-1)+2^m$, $m \in \mathbb{N}$, obtained from a vector, non-stationary subdivision scheme at the mesh points $\Omega_m = \{0, 2^{-m}, \ldots, 1\}$.

Proof. The proof is similar to the proof of Proposition 4.7. $\square$

The above results confirm that sequences of scalars or vectors obtained by subdivision can be viewed as restrictions of functions generated by CN's to dyadic meshes.

Chapter 5

Cascade Networks and the Cascade Algorithm

In this chapter, the close connection between cascade networks and cascade algorithms associated with refinement equations commonly used in wavelet theory and multi-resolution analysis is discussed.

Proposition 5.1. Let $n \in \mathbb{N}$. Suppose that $\phi : \mathbb{R} \to \mathbb{R}$, supported on $[0, n]$, satisfies the refinement equation

$$\phi(\cdot) = \sum_{i=0}^{n} a_i \phi(2 \cdot - i). \tag{5.1}$$

Consider $f : [0, 1] \to \mathbb{R}^n$ such that

$$f(\cdot) = \left(\phi(\cdot + n - 1)|_{[0,1]}, \phi(\cdot + n - 2)|_{[0,1]}, \ldots, \phi(\cdot + 1)|_{[0,1]}, \phi(\cdot)|_{[0,1]}\right)^T.$$

Then f satisfies,

$$f(x) = A(x) f(\alpha(x)), \quad x \in [0, 1], \tag{5.2}$$

where $A : [0, 1] \to \mathbb{R}^{n \times n}$ is a two-valued matrix such that

$$A(x) = \begin{cases} A^0, & x \in \left[0, \frac{1}{2}\right) \\ \\ A^1, & x \in \left[\frac{1}{2}, 1\right] \end{cases} = \begin{cases} \left(a_{j-2i+n-1}\right)_{i,j=0}^{n-1,n-1}, & x \in \left[0, \dfrac{1}{2}\right) \\ \left(a_{j-2i+n}\right)_{i,j=0}^{n-1,n-1}, & x \in \left[\dfrac{1}{2}, 1\right] \end{cases}.$$

Proof. If $x \in [0, \frac{1}{2})$, then by (5.1) and the fact that ϕ vanishes outside $[0, n]$,

$$f(x) = \begin{pmatrix} \phi(x + n - 1) \\ \vdots \\ \phi(x) \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{i=0}^{n} a_i \phi(2(x+n-1)-i) \\ \vdots \\ \sum_{i=0}^{n} a_i \phi(2x-i) \end{pmatrix}$$

$$= \begin{pmatrix} a_{n-1} & a_n & \dots & a_{2n-2} \\ a_{n-3} & a_{n-2} & \dots & a_{2n-4} \\ \vdots & \vdots & \vdots & \vdots \\ a_{-n+1} & a_{-n+2} & \dots & a_0 \end{pmatrix} \begin{pmatrix} \phi(2x+n-1) \\ \phi(2x+n-2) \\ \vdots \\ \phi(2x) \end{pmatrix}$$

$$= A^0 \begin{pmatrix} \phi(2x+n-1) \\ \vdots \\ \phi(2x) \end{pmatrix}$$

$$= A^0 f(2x).$$

If $x \in [\frac{1}{2}, 1]$, then by (5.1),

$$f(x) = \begin{pmatrix} \phi(x+n-1) \\ \vdots \\ \phi(x) \end{pmatrix}$$

$$= \begin{pmatrix} \sum_{i=0}^{n} a_i \phi(2(x+n-1)-i) \\ \vdots \\ \sum_{i=0}^{n} a_i \phi(2x-i) \end{pmatrix}$$

$$= \begin{pmatrix} a_n & a_{n+1} & \dots & a_{2n-1} \\ a_{n-2} & a_{n-1} & \dots & a_{2n-3} \\ \vdots & \vdots & \vdots & \vdots \\ a_{-n+2} & a_{-n+3} & \dots & a_1 \end{pmatrix} \begin{pmatrix} \phi(2x+n-2) \\ \phi(2x+n-3) \\ \vdots \\ \phi(2x-1) \end{pmatrix}$$

$$= A^1 \begin{pmatrix} \phi(2x+n-2) \\ \vdots \\ \phi(2x-1) \end{pmatrix}$$

$$= A^1 f(2x-1).$$

Therefore, f satisfies (5.2). $\square$

Proposition 5.2. Let $k, n \in \mathbb{N}$. Suppose that $\phi : [0,1] \to \mathbb{R}^k$, supported on $[0, n]$, satisfies the vector refinement equation

$$\phi(\cdot) = \sum_{i=0}^{n} B_i \phi(2 \cdot -i), \tag{5.3}$$

where $B_i \in \mathbb{R}^{k \times k}$, for $i = 0, \dots, n$.

Let $f : [0,1] \to \mathbb{R}^{kn}$ be such that

$$f(\cdot) = \left(\phi_1(\cdot + n - 1)|_{[0,1]}, \dots, \phi_k(\cdot + n - 1)|_{[0,1]}, \dots, \phi_1(\cdot)|_{[0,1]}, \dots, \phi_k(\cdot)|_{[0,1]}\right)^T.$$

Then, f satisfies

$$f(x) = A(x) f(\alpha(x)), \quad x \in [0,1], \tag{5.4}$$

where $A \in \mathbb{R}^{kn \times kn}$ such that

$$A(x) = \begin{cases} A^0, & x \in \left[0, \frac{1}{2}\right) \\ A^1, & x \in \left[\frac{1}{2}, 1\right] \end{cases} = \begin{cases} \left(B_{j-2i+n-1}\right)_{i,j=0}^{n-1,n-1}, & x \in \left[0, \frac{1}{2}\right) \\ \left(B_{j-2i+n}\right)_{i,j=0}^{n-1,n-1}, & x \in \left[\frac{1}{2}, 1\right] \end{cases}.$$

Proof. If $x \in [0, \frac{1}{2})$, then by (5.3),

$$f(x) = (\phi_1(x+n-1), \ldots, \phi_k(x+n-1), \ldots, \phi_1(x), \ldots, \phi_k(x))^T$$

$$= \begin{pmatrix} \sum_{i=0}^{n} B_i \phi(2(x+n-1)-i) \\ \vdots \\ \sum_{i=0}^{n} B_i \phi(2x-i) \end{pmatrix}$$

$$= \begin{pmatrix} B_{n-1} & B_n & \ldots & B_{2n-2} \\ B_{n-3} & B_{n-2} & \ldots & B_{2n-4} \\ \vdots & \vdots & \vdots & \vdots \\ B_{-n+1} & B_{-n+2} & \ldots & B_0 \end{pmatrix} \begin{pmatrix} \phi(2x+n-1) \\ \phi(2x+n-2) \\ \vdots \\ \phi(2x) \end{pmatrix}$$

$$= A^0 \begin{pmatrix} \phi(2x+n-1) \\ \vdots \\ \phi(2x) \end{pmatrix}$$

$$= A^0 f(2x).$$

If $x \in [\frac{1}{2}, 1]$, then by (5.3),

$$f(x) = (\phi_1(x+n-1), \ldots, \phi_k(x+n-1), \ldots, \phi_1(x), \ldots, \phi_k(x))^T$$

$$= \begin{pmatrix} \sum_{i=0}^{n} B_i \phi(2(x+n-1)-i) \\ \vdots \\ \sum_{i=0}^{n} B_i \phi(2x-i) \end{pmatrix}$$

$$= \begin{pmatrix} B_n & B_{n+1} & \dots & B_{2n-1} \\ B_{n-2} & B_{n-1} & \dots & B_{2n-3} \\ \vdots & \vdots & \vdots & \vdots \\ B_{-n+2} & B_{-n+3} & \dots & B_1 \end{pmatrix} \begin{pmatrix} \phi(2x+n-2) \\ \phi(2x+n-3) \\ \vdots \\ \phi(2x-1) \end{pmatrix}$$

$$= A^1 \begin{pmatrix} \phi(2x+n-2) \\ \vdots \\ \phi(2x-1) \end{pmatrix}$$

$$= A^1 f(2x-1).$$

Therefore, f satisfies (5.4). □

Next, consider the refinement function as a matrix function.

Proposition 5.3. Let $k, n \in \mathbb{N}$. Suppose that $F : [0,1] \to \mathbb{R}^{k\times k}$ supported on $[0, n]$ satisfies the generalized refinement equation

$$F(\cdot) = \sum_{i=0}^{n} B_i F(2\cdot -i), \tag{5.5}$$

where $B_i \in \mathbb{R}^{k\times k}$, for $i = 0, \dots, n$.

Let $f : [0,1] \to \mathbb{R}^{kn\times k}$ such that $f(\cdot) = \begin{pmatrix} F(\cdot+n-1) \\ \vdots \\ F(\cdot) \end{pmatrix}$.

Then f satisfies,

$$f(x) = A(x) f(\alpha(x)), \quad x \in [0,1],$$

where $A \in \mathbb{R}^{kn\times kn}$ such that

$$A(x) = \begin{cases} A^0, & x \in \left[0, \dfrac{1}{2}\right) \\ A^1, & x \in \left[\dfrac{1}{2}, 1\right] \end{cases} = \begin{cases} (B_{j-2i+n-1})_{i,j=0}^{n-1,n-1}, & x \in \left[0, \dfrac{1}{2}\right) \\ (B_{j-2i+n})_{i,j=0}^{n-1,n-1}, & x \in \left[\dfrac{1}{2}, 1\right] \end{cases}.$$

Proof. The proof is similar to the proof of Proposition 5.2. □

The above examples show that all refinable functions, including vector and matrix functions, satisfy an equation of the form $f = A(f \circ \alpha)$. This suggests an iterative generation of f as the limit of f_n, where $f_n = A(f_{n-1} \circ \alpha)$, starting with an initial approximation f_0. In the theory of refinable functions, this is known as the cascade algorithm. Thus, the cascade algorithm is a special case of cascade networks studied here.

Chapter 6

Convergence of Infinite Products of Matrices and Cascade Networks

This chapter discusses convergence of infinite products of matrices. If the input $\mathcal{Y}_0$ of a cascade network is a matrix, a cascade network leads to products of matrices and can be written as

$$\mathcal{Y}_L = A_L(A_{L-1} \circ \alpha) \cdots (A_1 \circ \alpha^{L-1})(\mathcal{Y}_0 \circ \alpha^L),$$

where A_ℓ are two-valued matrices for $\ell = 1, \ldots, L$. Using the language of subdivision algorithms, a CN is called *stationary* if for all ℓ, $A_\ell = A$ where

$$A(x) = \begin{cases} A^0, & x \in [0, 1/2) \\ A^1, & x \in [1/2, 1] \end{cases},$$

and $A^0, A^1 \in \mathbb{R}^{W \times W}$ are fixed. A CN is called *non-stationary* if

$$A_\ell(x) = \begin{cases} A^0_\ell, & x \in [0, 1/2) \\ A^1_\ell, & x \in [1/2, 1] \end{cases},$$

where $A^0_\ell, A^1_\ell \in \mathbb{R}^{W \times W}$ for $\ell = 1, \ldots, L$.

In the stationary case, the convergence of infinite product of matrices has been studied by Daubechies and Lagarias [47], Berger and Wang [48], and others [49, 50]. Sets of matrices all infinite products of which converge arise in many different contexts, including constructing parametrized curves by subdivision algorithms [51, 52, 53] and wavelets and refinement equations [54, 55].

6.1 Joint Spectral Radius

The concept of the joint spectral radius of a set of matrices is essential to the study of infinite products of matrices. First introduced by Rota and Strang in 1960, the joint spectral radius generalizes the concept of the spectral radius of a matrix to sets of matrices [56].

Definition 6.1. [47, 57] The *spectral radius* of a $r \times r$ matrix M, $\rho(M)$, is defined as the largest modulus of its eigenvalues:

$$\rho(M) := \max\{|\lambda| : Av = \lambda v\}.$$

The following results are well known [47, 56].

Lemma 6.2. *For $M \in \mathbb{R}^{r\times r}$,*

$$\lim_{n\to\infty} M^k = 0 \iff \rho(M) < 1.$$

Lemma 6.3. *If $\rho(M) < 1$, there exists a matrix norm, $\|\cdot\|$, such that $\|M\| < 1$.*

It can be shown that for $M \in \mathbb{R}^{r\times r}$, any matrix norm, $\|\cdot\|$, gives an upper bound for the spectral radius, $\rho(M) \leq \|M\|$. As a consequence, for all $k \in \mathbb{N}$, $\rho(M) \leq \|M^k\|^{1/k}$.

Theorem 6.4. *Let $M \in \mathbb{R}^{r\times r}$. For any matrix norm, $\|\cdot\|$,*

$$\rho(M) = \lim_{k\to\infty} \|M^k\|^{1/k}.$$

The concept of the spectral radius of a matrix can be generalized to a set of matrices, Σ [56].

Definition 6.5. The *joint spectral radius*, $\hat{\rho}(\Sigma)$, of a set of matrices Σ is defined by

$$\hat{\rho}(\Sigma) := \limsup_{k\to\infty} \left(\hat{\rho}_k(\Sigma, \|\cdot\|)\right)^{1/k}, \tag{6.1}$$

where $\|\cdot\|$ is any matrix norm and

$$\hat{\rho}_k(\Sigma, \|\cdot\|) := \sup\left\{\left\|\prod_{i=1}^{k} M_i\right\| : M_i \in \Sigma \text{ for } 1 \leq i \leq k\right\}.$$

Note that the definition (6.1) is independent of the norm used.

Example 6.6. Consider $\Sigma = \{A^0, A^1\}$, where $A^0 = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$ and $A^1 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix}$.

Clearly, $\rho(A^0) = \rho(A^1) = 1$.

Consider k even, $(A^0A^1)^k = \begin{pmatrix} 2^k & 0 \\ 0 & 0 \end{pmatrix}$, then

$$\begin{aligned} \hat{\rho}_k\left(\Sigma, \|\cdot\|_\infty\right) &= \sup\left\{\left\|\prod_{i=1}^{k} A^{\delta_i}\right\|\right\} \\ &\geq \left\{\left\|\left(A^0A^1\right)^{k/2}\right\|_\infty\right\} \\ &= \left\|\begin{pmatrix} \left(\sqrt{2}\right)^k & 0 \\ 0 & 0 \end{pmatrix}\right\|_\infty \\ &= \left(\sqrt{2}\right)^k \end{aligned}$$

and thus $\hat{\rho}(\Sigma) \geq \left(\left(\sqrt{2}\right)^k\right)^{1/k}$.

Clearly, $\hat{\rho}(\Sigma) \geq \sqrt{2}$, in fact, one can show that $\hat{\rho}(\Sigma) = \sqrt{2}$.

Definition 6.7. The *generalized spectral radius*, $\rho_\star(\Sigma)$, of any set of matrices Σ is

$$\rho_\star(\Sigma) := \lim_{k\to\infty} \sup\left(\rho_k(\Sigma)\right)^{1/k},$$

where

$$\rho_k(\Sigma) := \sup\left\{\rho\left(\prod_{i=1}^{k} M_i\right) | M_i \in \Sigma, \text{ for } 1 \leq i \leq k\right\}.$$

In [48], Berger and Wang showed $\rho_\star(\Sigma) = \hat{\rho}(\Sigma)$, for for all finite sets Σ.

6.2 Infinite Product of Matrices

Cascade networks lead to products of matrices. As the depth of a cascade network increases, that is as $L \to \infty$, we would like to understand under what conditions do the products of the matrices A_L converge or converge to a continuous limit. Convergent infinite products of matrices occur in many areas of mathematics, and has been studied most notably by Daubechies and Lagarias in [47]. Next, several results from that paper are presented.

In the following, the superscript i denotes the index of the set Σ, and is not an exponent.

Definition 6.8. [47] An infinite product $\prod_{i=1}^{\infty} A^i$, $A^i \in \mathbb{R}^{n\times n}$ for all i, *right converges* if $\lim_{i\to\infty} A^1 \cdots A^i$ exists, in which case define $\prod_{i=1}^{\infty} A^i := \lim_{i\to\infty} A^1 \cdots A^i$.

A set of $n \times n$ matrices, Σ, is said to be an *RCP set* ("right convergent product") if all infinite products of matrices of Σ right converge.

Analogously, one can define properties for *left convergence*. An infinite product $\prod_{i=1}^{\infty} A^i$ left converges if $\lim_{i\to\infty} A^i \cdots A^2 A^1$ exists in which case define $\prod_{i=1}^{\infty} A^i := \lim_{i\to\infty} A^i \cdots A^1$.

A set of $n \times n$ matrices, Σ, is said to be an *LCP set* ("left convergent product") if all infinite products of matrices in Σ left converge.

Define $\Sigma^T = \{A^T | A \in \Sigma\}$, then it follows that Σ is an RCP set if and only if Σ^T is an LCP set [47]. Therefore, results for LCP sets are interchangeable with RCP sets by taking the transpose of all matrices of Σ.

However, there exist RCP sets which are not LCP sets.

Example 6.9. [47] Consider $\Sigma = \{A^0, A^1\}$, where $A^0 = \begin{pmatrix} 1/2 & 0 \\ 1/2 & 0 \end{pmatrix}$ and $A^1 = \begin{pmatrix} 1 & 1/2 \\ 0 & 1/2 \end{pmatrix}$.

Σ is an RCP set, but not an LCP set.

If $\Sigma = \{A^0, A^1, \ldots, A^{m-1}\}$ is a finite set, then any sequence of elements of Σ can be characterized by a sequence $\mathbf{d} = (d_j)_{j\in\mathbb{N}}$ of digits drawn from $\{0, 1, \ldots, m-1\}$. Let $\mathbf{S}_m$ denote the set of all such sequences, equipped with metric $D(\mathbf{d}, \mathbf{d}') = m^{-r}$, where r is the first index such that $d_r \neq d_r'$. The induced topology on $\mathbf{S}_m$ is called the *sequence topology*.

For Σ a finite RCP set, the *limit function* $A_\Sigma(\cdot)$ is defined by $A_\Sigma(\mathbf{d}) := \prod_{j=1}^{\infty} A^{d_j}$, where $A_\Sigma(\mathbf{d}) \in \mathbb{C}^{n\times n}$.

Viewing $\mathbf{d}$ as an m-ary expansion of a real number, $\mathbf{S}_m$ can be mapped to $[0,1]$ by $x : \mathbf{S}_m \to [0,1]$, where $x(\mathbf{d}) = \sum_{j=1}^{\infty} d_j m^{-j}$. x is continuous and one to one except at the terminating rationals ℓ/m^j, which have two expansions of the form

$$\begin{array}{cccccccc} d_1 & \cdots & d_j & 0 & 0 & 0 & \cdots, \\ d_1 & \cdots & d_j - 1 & m-1 & m-1 & m-1 & \cdots. \end{array} \tag{6.2}$$

An RCP set Σ is called *real definable* if the images under A_Σ of any two sequences of the form (6.2) agree. Then, a *real limit function* $\overline{A}_\Sigma : [0,1] \to \mathbb{C}^{n\times n}$ is well defined and given by $\overline{A}_\Sigma(x) := A_\Sigma(\mathbf{d}(x))$, where $\mathbf{d}(x)$ is any m-ary expansion of x.

In [47], Daubechies and Lagarias give necessary conditions for a finite set Σ to be an RCP set, using the concept of the joint spectral radius.

Theorem 6.10. *[47] If Σ is a finite RCP set, then $\hat{\rho}(\Sigma) \leq 1$.*

Elsner and Friedland [49] give a necessary and sufficient norm condition for a set of two matrices to have the LCP property.

Theorem 6.11. *[49] The following conditions are equivalent:*

(i) The set $\Sigma = \{A^0, A^1\}$, $A^0, A^1 \in \mathbb{C}^{n\times n}$, is an LCP set.

(ii) There exists a norm $\|\cdot\|$ on $\mathbb{C}^n$ such that

(a) $\|A^i\| \leq 1, i = 0, 1$

(b) For $i = 0, 1$, if λ is an eigenvalue of A^i, $|\lambda| = 1$, then $\lambda = 1$;

(c) $\|A^0 A^1 x\| = \|x\|$ if and only if $A^0 x = A^1 x = x$.

6.3 Limit Functions of RCP, LCP Sets

Next, several known results about limit functions of RCP and LCP sets are presented. In [48, 49] various matrix norm conditions are given for a finite set of matrices Σ to be an LCP set or an LCP set with continuous limit function. First recall the following definition from [47].

Definition 6.12. [47] A set of matrices Σ is called *product bounded* if there exists a $C = C(\Sigma)$ such that all finite products satisfy

$$\left\| \prod_{i=1}^{k} A^i \right\| \leq C, \quad \text{for all } A^i \in \Sigma$$

The following was proved by Berger and Wang [48].

Theorem 6.13. *[48]*

a) If Σ is a finite LCP set, then Σ is product bounded.

b) Σ is LCP all of whose infinite products are zero if and only if $\hat{\rho}(\Sigma) < 1$.

The next lemma was proved by Beyn and Elsner in [50].

Lemma 6.14. *[50] For Σ a finite set of $n \times n$ matrices the following are equivalent.*

1. *The set Σ is product bounded.*

2. *There exists a vector norm $\|\cdot\|$ such that $\|Ax\| \leq \|x\|$ for all $A \in \Sigma, x \in \mathbb{C}^n$.*

3. *There exists a multiplicative matrix norm $\|\cdot\|$ such that $\|A\| \leq 1$ for all $A \in \Sigma$.*

Beyn and Elsner used Lemma 6.14 to show if Σ is an LCP set with a continuous limit function, then there exists a norm, $\|\cdot\|$, such that all for matrices $A \in \Sigma$, $\|Ax\| < \|x\|$ if and only if $Ax \neq x$, for all $x \in \mathbb{C}^n$.

In [47], Daubechies and Lagarias characterize finite RCP sets having continuous or real-continuous limit functions.

Theorem 6.15. *[47]*

Let Σ be a finite RCP set of $n \times n$ matrices. The following are equivalent:

1. *Σ is an RCP set whose limit function M_Σ is continuous.*

2. *All matrices A^i in Σ have the same 1-eigenspace $E_1 = E_1(A^i)$ and this eigenspace is simple for all A^i. There exists a vector space V with $\mathbb{R}^n = E_1 + V$, having the property that if P_V is an oblique projection onto V away from E_1, then $P_V \Sigma P_V$ is an RCP set whose limit function is identically zero.*

3. *All matrices A^i in Σ have the same 1-eigenspace $E_1 = E_1(A^i)$ and this eigenspace is simple for all A^i. For all vector spaces V with $\mathbb{R}^n = E_1 + V$ and $dim(V) = n - dim(E_1)$, if P_V is an oblique projection onto V away from E_1, then $P_V \Sigma P_V$ is an RCP set with limit function identically zero.*

Theorem 6.16. *[47] The finite ordered set $\Sigma = \{A^0, A^1, \ldots, A^{m-1}\}$ of $n \times n$ matrices is an RCP set with real-continuous limit function $\overline{A}_\Sigma$ if and only if Σ is an RCP set with a continuous limit function on S_m, and if*

$$A^i = S \begin{pmatrix} \mathbb{I}_d & \mathbf{0} \\ C_i & \tilde{A}^i \end{pmatrix} S^{-1}, \quad 0 \le i \le m-1,$$

with $d = \dim(E_1(\Sigma))$, E_1 the left 1-eigenspace of Σ, then

$$C_{i+1} + \tilde{A}^{i+1} \left(\mathbb{I}_{n-d} - \tilde{A}^0\right)^{-1} C_0 = C_i + \tilde{A}^i \left(\mathbb{I}_{n-d} - \tilde{A}^{m-1}\right)^{-1} C_{m-1}, \quad 0 \le i \le m-2.$$

The limit functions of an RCP set need not be continuous. Let $\Sigma = \{A^0, A^1, \ldots\}$ an RCP set with continuous limit function, then $\Sigma' = \{\mathbb{I}, A^0, A^1, \ldots\}$ has discontinuous limit function.

Example 6.17. In [47, 54] the set $\Sigma = \{A^0, A^1\}$ given by

$$A^0 = \begin{pmatrix} \frac{1+\sqrt{3}}{4} & 0 & 0 \\ \frac{3-\sqrt{3}}{4} & \frac{3+\sqrt{3}}{4} & \frac{1+\sqrt{3}}{4} \\ 0 & \frac{1-\sqrt{3}}{4} & \frac{3-\sqrt{3}}{4} \end{pmatrix}, \qquad A^1 = \begin{pmatrix} \frac{3+\sqrt{3}}{4} & \frac{1+\sqrt{3}}{4} & 0 \\ \frac{1-\sqrt{3}}{4} & \frac{3-\sqrt{3}}{4} & \frac{3+\sqrt{3}}{4} \\ 0 & 0 & \frac{1-\sqrt{3}}{4} \end{pmatrix}$$

is shown to be an RCP set with $E_1(\Sigma)$ generated by $(1, 1, 1)$, and has real-limit function

$$\overline{A}_\Sigma(\alpha) = \begin{pmatrix} f(\alpha) & f(\alpha) & f(\alpha) \\ f(\alpha+1) & f(\alpha+1) & f(\alpha+1) \\ f(\alpha+2) & f(\alpha+2) & f(\alpha+2) \end{pmatrix}$$

for $0 \le \alpha \le 1$, where the function f is continuous and differentiable on $[0, 3]$, except on a set of points of Hausdorff dimension ≤ 0.25.

6.4 Non-Stationary Cascade Networks

A natural question is what happens if a stationary cascade network is "perturbed" to become non-stationary. In the following assume $\Sigma = \{A^0, A^1\}$. Next, a result analogous to Lemma 6.14 is proved for non-stationary cascade networks, obtained by perturbing stationary CN's.

Definition 6.18. Matrices $A^0, A^1 \in \mathbb{R}^{r\times r}$ are said to satisfy Property B if $\displaystyle\prod_{\ell=1}^{n} A^{\delta_\ell}$ is uniformly bounded in n and δ, for $\delta_\ell \in \{0, 1\}$ and some norm $\|\cdot\|$.

Proposition 6.19. The following are equivalent:

a) A^0, A^1 satisfy Property B;

b) There exists a submultiplicative norm, $\|\cdot\|_0$, such that $\|A^0\|_0 \le 1$, $\|A^1\|_0 \le 1$;

Proof. $(b) \Rightarrow (a)$

$$\left\| \prod_{\ell=1}^{n} A^{\delta_\ell} \right\|_0 \le \prod_{\ell=1}^{n} \left\|A^{\delta_\ell}\right\|_0 \le 1.$$

Thus, A^0, A^1 satisfy Property B.

$(a) \Rightarrow (b)$

Let V be the unit ball in $\mathbb{R}^r$ centered at the origin with norm $\|\cdot\|_2$. Let

$$\mathcal{U}_n = \bigcup_{\delta_1,\ldots,\delta_n \in \{0,1\}} \left(\prod_{\ell=1}^{n} A^{\delta_\ell} \right) V.$$

By $(a), \mathcal{U}_n \subset \mathbb{R}^n$ are uniformly bounded, thus $\|u_n\|_2 = \sup\limits_{u \in \mathcal{U}_n} \|u\|_2 < C$ for all $n, C > 0$.

Define $\mathcal{U}$ to be the set of all accumulation points of $\mathcal{U}_n$. Then, $A^j u \in \mathcal{U}$, for $u \in \mathcal{U}$ and $j = 0, 1$. $\mathcal{U}$ is bounded, balanced, and the relative interior of $\mathcal{U}$ is non-empty.

Let W be the convex hull of $\mathcal{U}$. Then, W is balanced and bounded. Assume $\mathcal{U}$ is absorbing, then W is also absorbing. If $\mathcal{U}$ is not absorbing, then the degenerate case can be dealt with separately.

Let $w \in W$, then $w = t_1 u_1 + \cdots + t_{r+1} u_{r+1}$, where $u_1, \ldots, u_{r+1} \in \mathcal{U}$, $t_1, \ldots, t_{r+1} \geq 0$, and $\sum\limits_{i=1}^{r+1} t_i = 1$. Then,

$$\begin{aligned} A^0 w &= A^0 \left(t_1 u_1 + \cdots + t_{r+1} u_{r+1} \right) \\ &= t_1 A^0 u_1 + \cdots + t_{r+1} A^0 u_{r+1} \\ &\in W. \end{aligned}$$

Thus, $A^0 W \subset W$. Similarly, one can show that $A^1 W \subset W$.

Now, there exists a norm $\|\cdot\|_0$ whose unit ball is W. Thus, for $w \in W$, $A^0 w \in W$ if and only if $\|w\|_0 \leq 1$ implies $\|A^0 w\| \leq 1$. Similarly, for $w \in W$, $A^1 w \in W$ if and only if $\|w\|_0 \leq 1$ implies $\|A^1 w\| \leq 1$.

$\square$

Proposition 6.20. Let A^0, A^1 satisfy Property B. If $\{A^0_\ell\} \to A^0$, $\{A^1_\ell\} \to A^1$ and

$$\sum_{\ell} \|A^0_\ell - A^0\| < \infty$$
$$\sum_{\ell} \|A^1_\ell - A^1\| < \infty,$$

then $\prod_{\ell=k+1}^{k+n} A_\ell^{\delta_\ell}$ are uniformly bounded in n, k and δ, for $\delta_\ell \in \{0, 1\}$

Proof. By Proposition 6.19, we can consider a norm $\|\cdot\|_0$ such that $\|A^0\|_0 \leq 1, \|A^1\|_0 \leq 1$.

Thus,

$$\begin{aligned}
\left\|\prod_{\ell=k}^{k+n} A_\ell^{\delta_\ell}\right\|_0 &= \left\|\prod_{\ell=k}^{k+n}\left(A_\ell^{\delta_\ell} - A^{\delta_\ell} + A^{\delta_\ell}\right)\right\|_0 \\
&\leq \prod_{\ell=k}^{k+n}\left\|\left(A_\ell^{\delta_\ell} - A^{\delta_\ell} + A^{\delta_\ell}\right)\right\|_0 \\
&\leq \prod_{\ell=k}^{k+n}\left(\left\|\left(A_\ell^{\delta_\ell} - A^{\delta_\ell}\right)\right\|_0 + 1\right) \\
&\leq \prod_{\ell=k}^{k+n}\exp\left(\left\|A_\ell^{\delta_\ell} - A^{\delta_\ell}\right\|_0\right) \\
&\leq \prod_{\ell=k}^{k+n}\exp\left(\left\|A_\ell^0 - A^0\right\|_0\right)\exp\left(\left\|A_\ell^1 - A^1\right\|_0\right) \\
&\leq \exp\left(\sum_{\ell=1}^{\infty}\left\|A_\ell^0 - A^0\right\|_0 + \sum_{\ell=1}^{\infty}\left\|A_\ell^1 - A^1\right\|_0\right) \\
&< \infty.
\end{aligned}$$

□

Conjecture 6.1. *Let $\Sigma = \{A^0, A^1\}$ be an RCP set with real-continuous limit function. If $\{A^0_\ell\} \to$*

A^0, $\{A_\ell^1\} \to A^1$ *and*

$$\sum_\ell \|A_\ell^0 - A^0\| < \infty$$

$$\sum_\ell \|A_\ell^1 - A^1\| < \infty$$

then $\prod_{\ell=1}^{\infty} A_\ell^{\delta_\ell}$ *converges for all* $\delta = (\delta_1, \delta_2, \dots), \delta_\ell \in \{0, 1\}$*, and these products give rise to a continuous real matrix valued function* A *on* $[0,1]$ *in the sense that* $\prod_{\ell=1}^{\infty} A_\ell^{\delta_\ell} = A(x)$*, where* $(x)_2 = 0.\delta_1\delta_2\cdots$.

Chapter 7

The Space $\mathcal{S}_{W,L}$

The aim of this chapter is to characterize the space of functions which can be obtained by a cascade network with fixed width.

7.1 Definition and Properties

Definition 7.1. For $W, L \in \mathbb{N}$, let $\mathcal{S}_{W,L} = \{\mathcal{Y}_L | \ \mathcal{Y}_L(x) = A_L(x)(\mathcal{Y}_{L-1}(\alpha(x)) + b_L(x), \mathcal{Y}_{L-1} \in \mathcal{S}_{W,L-1}\}$ be the space of functions that can be obtained by a cascade network with fixed width W and of depth L, from all possible choices of weights and biases where $\mathcal{S}_{W,0} = \{\mathcal{Y}_0 | \mathcal{Y}_0 = (a_1 x + b_1, \ldots, a_W x + b_W)^T \in \mathbb{R}^W\} = \Pi_1^W([0,1])$.

Lemma 7.2. *Let $L \geq 1$. The spaces $\mathcal{S}_{W,L}$ are nested. Thus, for $L < K$, $\mathcal{S}_{W,L} \subset \mathcal{S}_{W,K}$.*

Proof. The proof is given for all bias terms equal to zero. The proof for non-zero biases is similar. It suffices to show that $\mathcal{S}_{W,L} \subset \mathcal{S}_{W,L+1}$. Let $\mathcal{Y}_L \in \mathcal{S}_{W,L}$, where

$$\mathcal{Y}_L(x) = A_L(x) A_{L-1}(\alpha(x)) \cdots A_1(\alpha^{L-1}(x)) \mathcal{Y}_0(\alpha^L(x)),$$

$A_i \in \mathbb{R}^{W \times W}$, for $i = 1 \ldots L$. Suppose $\mathcal{Y}_0(x) = (a_1 x + b_1, \cdots, a_W x + b_W)^T \in \mathbb{R}^W$, and that for all $i, j \in \{1, \ldots, W\}, i \neq j$, (a_i, b_i) is not a scalar multiple of (a_j, b_j), i.e there is no $\lambda \in \mathbb{R}$ such that $\lambda a_j = a_1$ and $\lambda b_j = b_i$. In particular, this implies that there is no $i \in \{1, \ldots, W\}$ such that $a_i = b_i = 0$.

There exists $G : [0,1] \to \mathbb{R}^{W \times W}$, defined by

$$G(x) = \begin{cases} G^0, & x \in [0, 1/2), \\ G^1, & x \in [1/2, 1] \end{cases},$$

such that $G(x)\mathcal{Y}_0\left(\alpha(x)\right) = \mathcal{Y}_0(x)$, for all $x \in [0,1]$.

First suppose there exists $n \in \{1, \ldots, W\}$ such that $a_n = 0$, then $b_n \neq 0$. Without loss of generality, assume $n = 1$, then $b_1 \neq 0$ and the first component of $\mathcal{Y}_0$ is b_1 for all $x \in [0,1]$. Then

$$G(x) := \begin{cases} \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \frac{b_2}{2b_1} & \frac{1}{2} & 0 & \cdots & 0 \\ \vdots & \cdots & \ddots & \cdots & 0 \\ \frac{b_{W-1}}{2b_1} & 0 & \cdots & \frac{1}{2} & 0 \\ \frac{b_W}{2b_1} & 0 & \cdots & 0 & \frac{1}{2} \end{pmatrix}, & x \in [0, 1/2) \\ \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ \frac{(b_2+a_2)}{2b_1} & \frac{1}{2} & 0 & \cdots & 0 \\ \vdots & \cdots & \ddots & \cdots & 0 \\ \frac{(b_{W-1}+a_{W-1})}{2b_1} & 0 & \cdots & \frac{1}{2} & 0 \\ \frac{(b_W+a_W)}{2b_1} & 0 & \cdots & 0 & \frac{1}{2} \end{pmatrix}, & x \in [1/2, 1]. \end{cases}$$

One can check that the statement is satisfied. Hence, it suffices to assume that $a_i \neq 0$ for all $i \in \{1, \ldots, W\}$. For all $i \in \{1, \ldots, W\}$ pick $k(i), \ell(i) \in \{1, \ldots, W\}$ such that $i, k(i), \ell(i)$ are distinct. For simplicity, fix i and set $k(i) = k, \ell(i) = \ell$. Let $G^0 = (g^0_{ij})$ and $G^1 = (g^1_{ij})$ and set $g^0_{ij} = g^1_{ij} = 0$ if $j \notin \{i, k, \ell\}$ and $g^0_{ii} = g^1_{ii} = 1/2$.

For $x \in [0, 1/2)$, let $G^0\mathcal{Y}_0(\alpha(x)) = (c_1(x), \ldots, c_W(x))^T$, and for $x \in [1/2, 1]$, let $G^1\mathcal{Y}_0(\alpha(x)) = (d_1(x), \ldots, d_W(x))^T$.

Then,

$$c_i(x) = a_i(x) + \frac{b_i}{2} + 2(g^0_{ik}a_k + g^0_{i\ell}a_\ell)x + (g^0_{ik}b_k + g^0_{i\ell}b_\ell).$$

Thus, $c_i(x) = (\mathcal{Y}_0(x))_i$ if $g^0_{ik}, g^0_{i\ell} \in \mathbb{R}$ such that

$$\begin{aligned} g^0_{ik}a_k + g^0_{i\ell}a_\ell &= 0 \\ g^0_{ik}b_k + g^0_{i\ell}b_\ell &= \frac{b_i}{2}. \end{aligned}$$

This is true if $a_k b_\ell - a_\ell b_k \neq 0$. But, if $a_k b_\ell = a_\ell b_k$, then either $b_k = b_\ell = 0$, which implies $a_k = \lambda a_\ell$ for some $\lambda \in \mathbb{R}$, or $\frac{a_\ell}{b_\ell} = \frac{a_k}{b_k}$ which implies there exists $\lambda \in \mathbb{R}$ such that $a_\ell = \lambda a_k$ and $b_\ell = \lambda a_\ell$. As no such λ exists, one can find $g^0_{ik}, g^0_{i\ell}$ satisfying this. Similarly,

$$d_i(x) = a_i(x) - \frac{a_i}{2} + \frac{b_i}{2} + 2(g^1_{ik} a_k + g^1_{i\ell} a_\ell)x + (g^1_{ik}(b_k - a_k) + g^1_{i\ell}(b_\ell - a_\ell)),$$

so that $d_i(x) = (\mathcal{Y}_0)_i$ as long as

$$\begin{aligned} g^1_{ik} a_k + g^1_{i\ell} a_\ell &= 0 \\ g^1_{ik}(b_k - a_k) + g^1_{i\ell}(b_\ell - a_\ell) &= \frac{b_i + a_i}{2}. \end{aligned}$$

By similar reasoning as above, one can find $g^1_{ik}, g^1_{i\ell}$ satisfying this. Therefore, $G(x)\mathcal{Y}_0\left(\alpha(x)\right) = \mathcal{Y}_0(x)$ for all $x \in [0, 1]$.

Replacing x by $\alpha^L(x)$, $G\left(\alpha^L(x)\right)\mathcal{Y}_0\left(\alpha^{L+1}(x)\right) = \mathcal{Y}_0\left(\alpha^L(x)\right)$. Then,

$$\begin{aligned} \mathcal{Y}_L(x) &= A_L(x)A_{L-1}(\alpha(x)) \cdots A_1(\alpha^{L-1}(x))\mathcal{Y}_0(\alpha^L(x)) \\ &= A_L(x)A_{L-1}(\alpha(x)) \cdots A_1(\alpha^{L-1}(x))G(\alpha^L)\mathcal{Y}_0(\alpha^{L+1}(x)) \\ &= B_{L+1}(x)B_L(\alpha(x)) \cdots B_1(\alpha^L(x))\mathcal{Y}_0(\alpha^{L+1}(x)), \end{aligned}$$

where $B_{L+1} = A_L, B_L = A_{L-1}, \ldots, B_2 = A_1$ and $B_1 = G$. Therefore, $\mathcal{Y}_L \in \mathcal{S}_{W,L+1}$. $\square$

7.2 $\mathcal{S}_{W,L}$ and "Periodicity"

In this section, we will give a characterization of $\mathcal{S}_{W,L}$ in terms of what might be called "periodicity".

Lemma 7.3. *Let $K \in \mathbb{N}$. Suppose that $\mathcal{Y}_K$ is a piecewise linear function on Ω_K such that*

(a) $\mathcal{Y}_K(\cdot + 1/2) = C\mathcal{Y}_K(\cdot) + d$ *on* $[0, 1/2)$*, for some* $C \in \mathbb{R}^{W \times W}, d \in \mathbb{R}^W$*;*

(b) $\mathcal{Y}_K(2^{-1}\cdot)|_{[0,1]} \in \mathcal{S}_{W,K-1}$.

Then $\mathcal{Y}_K \in \mathcal{S}_{W,K}$.

Proof. It suffices to show that

$$\mathcal{Y}_K(\cdot) = A_K^0 \mathcal{Y}_{K-1}(2\cdot) + b_K^0, \text{ on } [0, 1/2), \tag{7.1}$$

and

$$\mathcal{Y}_K(\cdot) = A_K^1 \mathcal{Y}_{K-1}(2\cdot -1) + b_K^1, \text{ on } [1/2, 1], \tag{7.2}$$

for some $\mathcal{Y}_{K-1} \in \mathcal{S}_{W,K-1}$ and some matrices A_K^0, A_K^1 and vectors b_K^0, b_K^1.

To show (7.1), select $A_K^0 = \mathbb{I}_W$, $b_K^0 = 0 \in \mathbb{R}^W$ and $\mathcal{Y}_{K-1} \coloneqq \mathcal{Y}_K(2^{-1}\cdot)$ on $[0, 1]$. Then, by part (b), $\mathcal{Y}_{K-1} \in \mathcal{S}_{W,K-1}$.

Next, select A_K^1, b_K^1 such that (7.2) holds. Notice that (7.2) is equivalent to

$$\mathcal{Y}_K(\cdot + 1/2) = A_K^1 \mathcal{Y}_{K-1}(2\cdot) + b_K^1, \text{ on } [0, 1/2).$$

However, on $[0, 1/2)$, $\mathcal{Y}_K(\cdot + 1/2) = C\mathcal{Y}_K(\cdot) + d$, by assumption (a), and $\mathcal{Y}_K(\cdot) = \mathcal{Y}_{K-1}(2\cdot)$. Thus,

$$\mathcal{Y}_K(\cdot + 1/2) = C\mathcal{Y}_{K-1}(2\cdot) + d.$$

Now, select $A_K^1 = C$ and $b_K^1 = d$. Therefore, $\mathcal{Y}_K \in \mathcal{S}_{W,K}$. $\square$

Lemma 7.4. *Let* $L \in \mathbb{N}$. *Let* $j \in \{0, \ldots, 2^L - 1\}$, *with* $(j)_2 = \delta_1 \cdots \delta_L$ *its binary representation. If* $\mathcal{Y}_L$ *is such that*

(a) $\mathcal{Y}_L|_{[j2^{-L},(j+1)2^{-L})} \in \Pi_1^W\left([0, 2^{-L})\right)$,

(b)
$$\mathcal{Y}_L\left(\cdot + \frac{\delta_1 \cdots \delta_{\ell-1}}{2^{\ell-1}} + \frac{1-\delta_\ell}{2^\ell}\right)\bigg|_{[0,2^{-\ell})} = C_L^\ell \mathcal{Y}_L\left(\cdot + \frac{\delta_1 \cdots \delta_\ell}{2^\ell}\right)\bigg|_{[0,2^{-\ell})} + d_L^\ell,$$

for $\ell = 1, \ldots, L$ *and some* $j \in \{0, \ldots, 2^\ell - 1\}$.

Then $\mathcal{Y}_L \in \mathcal{S}_{W,L}$.

Proof. Without loss of generality, assume that $j = 0$. The proof will proceed by induction.

For $L = 1$, by condition (a), $\mathcal{Y}_1|_{[0,1/2)}$ is affine, thus $\mathcal{Y}_1(2^{-1}\cdot) \in \mathcal{S}_{W,0}$. By condition (b), $\mathcal{Y}_1(\cdot + 1/2) = C_1^1 \mathcal{Y}_1(\cdot) + d_1^1$ on $[0, 1/2)$. Therefore, by Lemma 7.3, $\mathcal{Y}_1 \in \mathcal{S}_{W,1}$.

For $L > 1$, suppose the statement of Lemma 7.4 is true for $L - 1$. Now, we will show that the statement of Lemma 7.4 holds for L.

Let $\mathcal{Y}_L$ be given, and define $\mathcal{Y}_{L-1} \coloneqq \mathcal{Y}_L\left(2^{-1}\cdot\right)$, on $[0, 1]$. As $\mathcal{Y}_{L-1}$ satisfies condition (a) of Lemma 7.4, $\mathcal{Y}_{L-1}|_{[0,2^{-L+1})}$ is affine, thus $\mathcal{Y}_{L-1}\left(2^{-L+1}\cdot\right)|_{[0,1]} \in \mathcal{S}_{W,0}$. Then, by the definition of $\mathcal{Y}_{L-1}$, $\mathcal{Y}_L\left(2^{-L}\cdot\right)|_{[0,1]} \in \mathcal{S}_{W,0}$. As $\mathcal{Y}_{L-1}$ and $\mathcal{Y}_L$ satisfy condition (b) of Lemma 7.4,

$$\mathcal{Y}_{L-1}\left(\cdot + 2^{-\ell}\right) = C^l_{L-1}\mathcal{Y}_{L-1}(\cdot) + d^\ell_{L-1}, \text{ for } \ell = 1, \ldots, L \text{ on } [0, 2^{-\ell})$$

and

$$\mathcal{Y}_L\left(\cdot + 2^{-\ell}\right) = C^\ell_L \mathcal{Y}_L(\cdot) + d^\ell_L \text{ for } \ell = 1, \ldots, L \text{ on } [0, 2^{-\ell}).$$

Thus, for $\ell = 1, \ldots, L - 1$,

$$\begin{aligned}
\mathcal{Y}_{L-1}\left(\cdot + 2^{-\ell}\right)|_{[0,2^{-\ell})} &= \mathcal{Y}_L\left(2\left(\cdot + 2^{-\ell}\right)\right)|_{[0,2^{-\ell})} \\
&= \mathcal{Y}_L\left(\cdot + 2^{-\ell-1}\right)|_{[0,2^{-\ell-1})} \\
&= C^{\ell+1}_L \mathcal{Y}_L\left(\cdot\right)|_{[0,2^{-\ell-1})} + d^{\ell+1}_L
\end{aligned}$$

$$\begin{aligned}
\mathcal{Y}_L\left(\cdot + 2^{-\ell-1}\right)|_{[0,2^{-\ell-1})} &= \mathcal{Y}_{L-1}\left(2 \cdot + 2^{-\ell}\right)|_{[0,2^{-\ell})} \\
&= C^\ell_{L-1}\mathcal{Y}_{L-1}(2\cdot) + d^\ell_{L-1} \\
&= C^\ell_{L-1}\mathcal{Y}_L(\cdot) + d^\ell_{L-1}
\end{aligned}$$

Let $-\ell - 1 = -\ell'$, then $C^{\ell'}_{L-1} = C^{\ell+1}_L$, and $\mathcal{Y}_{L-1} \in \mathcal{S}_{W,L-1}$.

Thus, $\mathcal{Y}_L(2^{-1}\cdot) \in \mathcal{S}_{W,L-1}$. Therefore, $\mathcal{Y}_L \in \mathcal{S}_{W,L}$, by Lemma 7.3.

□

Proposition 7.5. Let $L \in \mathbb{N}, \mathcal{Y}_L \in \mathcal{S}_{W,L}$, where $\mathcal{Y}_L$ corresponds to matrices$A_1^0, A_1^1, \ldots, A_L^0, A_L^1$. Let $j \in \{0, \ldots, 2^L - 1\}$, where $(j)_2 = \delta_1 \cdots \delta_L$ is its binary representation. If the matrices $A_L^{\delta_L}, \ldots, A_1^{\delta_1}$ are non-singular, then there exists matrices C_L^ℓ and vectors d_L^ℓ such that,

$$\mathcal{Y}_L \left(\cdot + \frac{\delta_1 \cdots \delta_{\ell-1}}{2^{\ell-1}} + \frac{1-\delta_\ell}{2^\ell} \right) \Big|_{[0,2^{-\ell})} = C_L^\ell \mathcal{Y}_L \left(\cdot + \frac{\delta_1 \cdots \delta_\ell}{2^\ell} \right) \Big|_{[0,2^{-\ell})} + d_L^\ell, \tag{7.3}$$

for $\ell = 1, \ldots, L$.

Conversely, if (7.3) holds for $\mathcal{Y}_L$, then $\mathcal{Y}_L \in \mathcal{S}_{W,L}$.

Proof. Without loss of generality, assume $j = 0$. Let $\mathcal{Y}_L \in \mathcal{S}_{W,L}$ with non-singular matrices $A_L^0, \ldots, A_1^0$. As the input $\mathcal{Y}_0$ is an affine vector, $\mathcal{Y}_L|_{[0,2^{-L}]}$ is also affine.

To show $\mathcal{Y}_L(\cdot + 2^{-\ell})|_{[0,2^{-\ell})} = C_L^\ell \mathcal{Y}_L(\cdot)|_{[0,2^{-\ell})} + d_L^\ell$, for $\ell = 1, \ldots, L$, the proof will proceed by induction.

For $L = 1$,

$$\begin{aligned}
\mathcal{Y}_1(x)) &= A_1(x)\mathcal{Y}_0(\alpha(x)) + b_1(x) \\
&= \begin{cases} A_1^0 \mathcal{Y}_0(2x) + b_1^0, & x \in \left[0, \frac{1}{2}\right) \\ A_1^1 \mathcal{Y}_0(2x-1) + b_1^1, & x \in \left[\frac{1}{2}, 1\right] \end{cases}.
\end{aligned}$$

On $[0, 1/2)$,

$$\begin{aligned}
\mathcal{Y}_1(\cdot + 1/2) - b_1^1 &= A_1^1 \mathcal{Y}_0(2(\cdot + 1/2) - 1) \\
&= A_1^1 \mathcal{Y}_0(2\cdot) \\
&= (A_1^1)(A_1^0)^{-1}(A_1^0)\mathcal{Y}_0(2\cdot) \\
&= A_1^1 (A_1^0)^{-1} \left(\mathcal{Y}_1(\cdot) - b_1^0 \right).
\end{aligned}$$

Thus, $\mathcal{Y}_1(\cdot + 1/2) = C_1^1 \mathcal{Y}_1(\cdot) + d_L^1$, where $C_1^1 = A_1^1(A_1^0)^{-1}$ and $d_1^1 = A_1^1(A_1^0)^{-1}(-b_1^0) + b_1^1$.

Next, suppose the statement is true for $L-1$. Then, we show that $\mathcal{Y}_L(\cdot+2^{-\ell}) = C_L^\ell \mathcal{Y}_L(\cdot) + d_L^\ell$ on $[0, 2^{-\ell})$ for $\ell = 1, \dots, L$.

For $\ell = 1$ on $[0, 1/2)$,

$$
\begin{aligned}
\mathcal{Y}_L(\cdot + 1/2) - b_L^1 &= A_L^1 \mathcal{Y}_{L-1}(2\cdot) \\
&= A_L^1 (A_L^0)^{-1} \left(\mathcal{Y}_L(\cdot) - b_L^0\right) \\
&= C_L^1 \mathcal{Y}_L(\cdot) + d_L^1
\end{aligned}
$$

Thus, $\mathcal{Y}_L(\cdot + 1/2) = C_L^1 \mathcal{Y}_L(\cdot) + d_L^1$, where $C_L^1 = A_L^1 (A_L^0)^{-1}$ and $d_L^1 = A_L^1 (A_L^0)^{-1}(-b_L^0) + b_L^1$, on $[0, 1/2)$.

For $\ell > 1$ on $[0, 2^{-\ell})$,

$$
\begin{aligned}
\mathcal{Y}_L(\cdot + 2^{-\ell}) - b_L^0 &= A_L^0 \mathcal{Y}_{L-1}(2 \cdot + 2^{-\ell+1}) \\
&= A_L^0 \left(C_{L-1}^{\ell-1} \mathcal{Y}_{L-1}(2\cdot) + d_{L-1}^{\ell-1}\right) \\
&= A_L^0 C_{L-1}^{\ell-1} (A_L^0)^{-1} (A_L^0) \mathcal{Y}_{L-1}(2\cdot) + A_L^0 d_{L-1}^{\ell-1} \\
&= A_L^0 C_{L-1}^{\ell-1} (A_L^0)^{-1} \left(\mathcal{Y}_L(\cdot) - b_L^0\right) + A_L^0 d_{L-1}^{\ell-1}
\end{aligned}
$$

Thus, on $[0, 2^{-\ell})$,

$$
\mathcal{Y}_L(\cdot + 2^{-L}) = C_L^\ell \mathcal{Y}_L(\cdot) + d_L^\ell, \tag{7.4}
$$

where $C_L^\ell = A_L^0 C_{L-1}^{\ell-1} (A_L^0)^{-1}$ and $d_L^\ell = A_L^0 C_{L-1}^{\ell-1} (A_L^0)^{-1}(-b_L^0) + A_L^0 d_{L-1}^{\ell-1} + b_L^0$.

The proof of the converse is similar and simpler. One can set $A_\ell^0 = \mathbb{I}_W$, $A_\ell^1 = C_L^\ell$, $\ell = 1, \dots, L$, and show by induction that $\mathcal{Y}_L \in \mathcal{S}_{W,L}$, i.e that $\mathcal{Y}_L$ can be generated by the recursion

$$
\mathcal{Y}_L = A_L \left(A_{L-1} \circ \alpha\right) + b_L, \quad \mathcal{Y}_{L-1} \in \mathcal{S}_{W,L},
$$

for appropriately chosen b_L, where

$$A_L(x) = \begin{cases} A_L^0, & x \in [0, 1/2) \\ A_L^1, & x \in [1/2, 1] \end{cases} = \begin{cases} \mathbb{I}_W, & x \in [0, 1/2) \\ C_L^L, & x \in [1/2, 1] \end{cases}$$

. $\square$

It follows from Proposition 7.5 that for $\mathcal{Y}_L \in \mathcal{S}_{W,L}$ the complexity of the CN representation is reduced from $2L$ matrices to L matrices.

Remark 7.1. *An element $\mathcal{Y}_L$ of $\mathcal{S}_{W,L}$ can be rewritten without the bias terms b_i, for $i = 1, \ldots, L$ as*

$$\tilde{\mathcal{Y}}_L = \tilde{A}_L(\tilde{A}_{L-1} \circ \alpha) \cdots (\tilde{A}_1 \circ a^{L-1})(\tilde{\mathcal{Y}}_0 \circ \alpha^L),$$

by letting $\tilde{\mathcal{Y}}_0 = (1, \mathcal{Y}_0)^T$ and setting $\tilde{A}_i = \left(\begin{array}{c|c} 1 & \mathbf{0} \\ \hline b_i & A_i \end{array}\right) \in \mathbb{R}^{(W+1)\times(W+1)}$, then $\tilde{\mathcal{Y}}_L = (1, \mathcal{Y}_L)^T$.

Thus, in Proposition 7.5, for $\mathcal{Y}_L \in \mathcal{S}_{W,L}$, $\mathcal{Y}_L\left(\cdot + 2^{-k}\right)|_{[0,2^{-k})} = C_L^k \mathcal{Y}_L(\cdot)|_{[0,2^{-k})} + d_L^k$, where $C_L^k \in \mathbb{R}^{W\times W}$ and $d_L^k \in \mathbb{R}^W$ can be rewritten as $\tilde{\mathcal{Y}}_L\left(\cdot + 2^{-k}\right)|_{[0,2^{-k})} = \tilde{C}_L^k \tilde{\mathcal{Y}}_L(\cdot)|_{[0,2^{-k})}$, where $C_L^k \in \mathbb{R}^{(W+1)\times(W+1)}$ and $\tilde{\mathcal{Y}}_L = (1, \mathcal{Y}_L)^T \in \mathcal{S}_{W+1,L}$. From now on, all elements of $\mathcal{S}_{W,L}$ will be assumed to be written without the bias term.

Equation (7.4) suggests that one consider the limiting situation $\mathcal{Y}_L \to f$ as $L \to \infty$. The next few results pertain to this question.

Lemma 7.6. *Let $W \in \mathbb{N}$ and let $\mathcal{S}_{C_1,\ldots,C_K}$ be the set of all functions $f : [0, 1] \to \mathbb{R}^W$ such that*

$$f(\cdot + 2^{-k})|_{[0,2^{-k})} = C_k f(\cdot)|_{[0,2^{-k})}, \quad k = 1, \ldots, K. \tag{7.5}$$

Then,

(a) $\mathcal{S}_{C_1,\ldots,C_K}$ is a linear space.

(b) Let $j \in \{0, \dots, 2^K - 1\}$, where $(j)_2 = \delta_1 \cdots \delta_K$ is its binary representation. If $f|_{[0,2^{-K})} \in L_\infty^W([0, 2^{-K}))$, then $f \in L_\infty^W([0,1])$ and

$$\|f\|_{L_\infty^W([0,1])} \leq \|C_1^{\delta_1} \cdots C_K^{\delta_K}\|_\infty \|f\|_{L_\infty^W([0,2^{-K}))}.$$

Proof. (a) Let $f, g \in \mathcal{S}_{C_1,\dots,C_K}$ and $\alpha, \beta \in \mathbb{R}$, then for $k = 1, \dots, K$,

$$\begin{aligned}
(\alpha f + \beta g)(\cdot + 2^{-k})|_{[0,2^{-k})} &= \alpha f(\cdot + 2^{-k})|_{[0,2^{-k})} + \beta g(\cdot + 2^{-k})|_{[0,2^{-k})} \\
&= \alpha C_k f(\cdot)|_{[0,2^{-k})} + \beta C_k g(\cdot)|_{[0,2^{-k})} \\
&= C_k(\alpha f + \beta g)(\cdot)|_{[0,2^{-k})}.
\end{aligned}$$

Thus $\alpha f + \beta g \in \mathcal{S}_{C_1,\dots,C_K}$.

(b) Identity $f(\cdot + 2^{-K})|_{[0,2^{-K})} = C_K f(\cdot)|_{[0,2^{-K})}$ implies

$$\|f\|_{L_\infty^W([0,2^{-K+1}])} \leq \max\{1, \|C_K\|_\infty\} \|f\|_{L_\infty^W([0,2^{-K}])}.$$

More generally, for $j \in \{0, \dots, 2^K - 1\}$,

$$\|f\|_{L_\infty^W([j2^{-K},(j+1)2^{-K}])} = \left\| C_1^{\delta_1} \cdots C_K^{\delta_K} f \right\|_{L_\infty^W([0,2^{-K}])},$$

therefore,

$$\|f\|_{L_\infty^W([0,1])} \leq \max_{j \in \{0,\dots,2^K-1\}} \left\| C_1^{\delta_1} \cdots C_K^{\delta_K} \right\|_\infty \|f\|_{L_\infty^W([0,2^{-K}])}. \tag{7.6}$$

$\square$

Let $f : [0,1] \to \mathbb{R}^W$. Then, f is said to be *bounded at the origin* if $f \in L_\infty^W([0,t])$, for some $t > 0$. f is said to be *essentially continuous at the origin* if there exists a vector $f_0 \in \mathbb{R}^W$ such that

$$\lim_{h \to 0^+} \|f - f_0\|_{L_\infty^W([0,h])} = 0.$$

To establish the next results, we need to formulate an assumption concerning functions f to be considered in connection with identity (7.5).

For a given $f : [0,1] \to \mathbb{R}^W$, $f = (f_1, \dots, f_W)^T$, it is not difficult to show that (7.5) implies that

$$f\Big|_{\left[j2^{-k},(j+1)2^{-k}\right)} = B^{k,j} f\Big|_{[0,2^{-k})}, \tag{7.7}$$

for all $j \in \{0, \dots, 2^k - 1\}$ and all $k \in \mathbb{N}$, for some matrices $B^{k,j} \in \mathbb{R}^{W \times W}$.

In particular, let $G^{k,j} = \left(f\left(j2^{-k}\right), f\left(j2^{-k}+h\right), \dots, f\left(j2^{-k}+(W-1)h\right)\right)^T \in \mathbb{R}^{W \times W}$, where $h := \frac{1}{W2^k}$.

Then, (7.7) implies that $G^{k,j} = B^{k,j} G^{k,0}$.

Definition 7.7. A function $f : [0,1] \to \mathbb{R}^W$ is said to have *locally uniformly independent components* (LUIC) if

$$\left\| G^{k,j} \left(G^{k,0}\right)^{-1} \right\|_\infty < \gamma, \tag{7.8}$$

for all $j \in \{0, \dots, 2^k - 1\}$, $k \in \mathbb{N}$, for some $\gamma > 0$ independent of j, k.

A few remarks on this definition are in order.

If $W = 1$, then LUIC simply means that $G^{k,0} = f(0) \neq 0$, assuming that f is bounded. Clearly, the condition $f(0) \neq 0$ is natural if we want f to satisfy (7.5) because otherwise, if $f(0) = 0$, (7.5) implies that $f(x) = 0$ for all dyadic points, an uninteresting case.

For $W > 1$ and $f \in \left(C^W([0,1])\right)^W$, then LUIC is equivalent to the property that the Wronskian determinant of f at the origin does not vanish, i.e

$$det \begin{vmatrix} f_1(0) & f_1'(0) & \cdots & f_1^{(W-1)}(0) \\ \vdots & \vdots & \vdots & \vdots \\ f_W(0) & f_W'(0) & \cdots & f_W^{(W-1)}(0). \end{vmatrix} \neq 0$$

Thus, the LUIC property means that f should not be “degenerate” at the origin. Again, the condition that the Wronskian is not zero at the origin is natural because the vanishing of the Wronskian would imply that the components of f are dependent functions.

Example 7.8. Let $f(x) = (1, x)^T$, then $G^{k,0} = (f(0), f(h))^T$, $h = 2^{-k}/W = 2^{-k-1}$ and $G^{k,j} = (f(x), f(x+h)), x = j2^{-k}$.

Then, $G^{k,j}\left(G^{k,0}\right)^{-1} = \begin{pmatrix} 1 & 0 \\ x & 1 \end{pmatrix}$, which is uniformly bounded for all $x \in [0,1]$. Hence f satisfies LUIC.

Example 7.9. Let $f(x) = (1, x^2)^T$.

Then, $G^{k,j}\left(G^{k,0}\right)^{-1} = \begin{pmatrix} 1 & 0 \\ x^2 & 1 - \frac{2x}{h} \end{pmatrix}$, which is not uniformly bounded with respect to k, given that $h = 2^{-k-1}$. Also, the Wronskian determinant of f at 0 is $det \begin{vmatrix} 1 & 0 \\ 0 & 0 \end{vmatrix} = 0$.

In general, pointwise values are not well-defined in L_∞. In which case, f should be replaced by $\bar{f} \coloneqq \lim\limits_{\varepsilon \to 0^+} \|f\|_{L_\infty([x-\varepsilon, x+\varepsilon]\cap[0,1])}$, which is always well-defined for L_∞ functions.

Lemma 7.10. *Let $W \in \mathbb{N}$ and $f : [0,1] \to \mathbb{R}^W$ be such that*

(a) f has LUIC,

(b) f satisfies

$$f(\cdot + 2^{-k})|_{[0,2^{-k})} = C_k f(\cdot)|_{[0,2^{-k})}$$

for some matrices $C_k \in \mathbb{R}^{W \times W}$, $k \geq 1$.

Then, $f \in L_\infty^W([0,1])$ if and only if f is bounded at the origin and

$$\gamma \coloneqq \sup_K \max_{j \in \{0,\ldots,2^K-1\}} \left\| C_1^{\delta_1} \cdots C_K^{\delta_K} \right\|_\infty < \infty,$$

where for $j \in \{0, \ldots, 2^L - 1\}$, $(j)_2 = \delta_1 \cdots \delta_K$ is its binary representation.

Proof. $\Rightarrow$ Let $f \in L_\infty^W([0,1])$ such that $(a), (b)$ hold. Then, $G^{k,j} = C_1^{\delta_1} \cdots C_k^{\delta_k} G^{k,0}$ follows from

$$f(x + j2^{-k}) = C_1^{\delta_1} \cdots C_k^{\delta_k} f(x), \quad x \in [0, 2^{-k}).$$

Thus,

$$\left\| C_1^{\delta_1} \cdots C_k^{\delta_k} \right\|_\infty = \left\| G^{k,j} \left(G^{k,0} \right)^{-1} \right\|_\infty < \gamma < \infty.$$

for all j, k.

$\Leftarrow$ For $j \in \{0, \ldots, 2^k - 1\}$, it follows that

$$f\left(\cdot + j2^{-k}\right)|_{[0,2^{-k})} = C_1^{\delta_1} \cdots C_k^{\delta_k} f(\cdot)|_{[0,2^{-k})}.$$

Thus,

$$\begin{aligned} \|f\|_{L_\infty^W\left([j2^{-k},(j+1)2^{-k}]\right)} &= \left\| C_1^{\delta_1} \cdots C_k^{\delta_k} f(\cdot) \right\|_{L_\infty^W([0,2^{-k}]W)} \\ &\leq \gamma \|f\|_{L_\infty^W\left([0,2^{-k}]\right)} \end{aligned}$$

for all j. Therefore, $\|f\|_{L_\infty^W([0,1])} \leq \gamma \|f\|_{L_\infty^W\left([0,2^{-k}]\right)}$, for all $k \in \mathbb{N}$, which is bounded by boundedness of f at the origin.

$\square$

Theorem 7.11. *Let $f \in L_\infty^W([0,1])$ be essentially continuous at the origin and have LUIC. Then, there exists $\{\mathcal{Y}_L\}_{L \in \mathbb{N}}$, $\mathcal{Y}_L \in \mathcal{S}_{W,L}$ such that $\mathcal{Y}_L \to f$ in L_∞ if and only if f satisfies*

$$f(\cdot + 2^{-k})|_{[0,2^{-k})} = C_k f(\cdot)|_{[0,2^{-k})} \tag{7.9}$$

for all $k \in \mathbb{N}$ and some matrices $C_k \in \mathbb{R}^{W \times W}$.

Proof. $\Rightarrow$ Fix $k \in \mathbb{N}$. Note that by LUIC, the components of $f|_{[0,2^{-k})}$ are independent, because $G^{k,0} = (f(0), f(h), \ldots, f((W-1)h)), h = \frac{1}{W2^{-k}}$, is invertible. Hence, there exists $\beta_k > 0$ such that for all $c \in \mathbb{R}^W$,

$$\beta_k \|c\|_\infty \leq \|\langle c, f\rangle\|_{L_\infty^W([0,2^{-k}])}. \tag{7.10}$$

Let $\mathcal{Y}_L \to f$ in $L_\infty^W([0,1])$. For all $L \geq k$, as $\mathcal{Y}_L \in \mathcal{S}_{W,L}$, by Lemma 7.4, $\mathcal{Y}_L\left(\cdot + 2^{-k}\right)|_{[0,2^{-k})} = C_L^k \mathcal{Y}_L(\cdot)|_{[0,2^{-k})}$, where $C_L^k \in \mathbb{R}^{W \times W}$.

As $L \to \infty$, $\mathcal{Y}_L\left(\cdot + 2^{-k}\right)|_{[0,2^{-k}]} \to f\left(\cdot + 2^{-k}\right)|_{[0,2^{-k}]}$ and $\mathcal{Y}_L(\cdot)|_{[0,2^{-k}]} \to f(\cdot)|_{[0,2^{-k}]}$ in L_∞. By (7.10) and by selecting L large enough so that $\|\mathcal{Y}_L - f\|_{L_\infty^W([0,1])} =: \varepsilon \leq \min\{1, \beta_k/2\}$,

$$\begin{aligned}
\|\langle c, f\rangle\|_{L_\infty([0,2^{-k}])} &\leq \|\langle c, f - \mathcal{Y}_L\rangle\|_{L_\infty([0,2^{-k}])} + \|\langle c, \mathcal{Y}_L\rangle\|_{L_\infty([0,2^{-k}])} \\
&\leq \|c\|_\infty \|f - \mathcal{Y}_L\|_{L_\infty^W([0,2^{-k}])} + \|\langle c, \mathcal{Y}_L\rangle\|_{L_\infty([0,2^{-k}])}.
\end{aligned}$$

Thus,

$$\begin{aligned}
\|\langle c, \mathcal{Y}_L\rangle\|_{L_\infty([0,2^{-k}])} &\geq \|\langle c, f\rangle\|_{L_\infty([0,2^{-k}])} - \varepsilon\|c\|_\infty \\
&\geq \beta_k\|c\|_\infty - \varepsilon\|c\|_\infty \\
&\geq (\beta_k - \varepsilon)\|c\|_\infty \\
&\geq (\beta_k/2)\|c\|_\infty.
\end{aligned}$$

Now, consider $\mathcal{Y}_L(x + 2^{-k}) = C_L^k \mathcal{Y}_L(x)$, for $x \in \left[0, 2^{-k}\right)$. Looking at the i-th row of C_L^k, $\mathcal{Y}_L^i\left(x + 2^{-k}\right) = (C_L^k)_i \mathcal{Y}_L(x) = \langle (C_L^k)_i, \mathcal{Y}_L(x)\rangle$. Then,

$$\begin{aligned}
(\beta_k/2)\left\|(C_L^k)_i\right\|_\infty &\leq \left\|\langle (C_L^k)_i, \mathcal{Y}_L(x)\rangle\right\|_{L_\infty([0,2^{-k}])} \\
&= \|\mathcal{Y}_L^i\left(x + 2^{-k}\right)\|_{L_\infty\left([0,2^{-k}]\right)} \\
&= \|\mathcal{Y}_L^i(x)\|_{L_\infty\left([2^{-k}, 2\cdot 2^{-k}]\right)} \\
&\leq \varepsilon + \|f\|_{L_\infty^W\left([2^{-k}, 2\cdot 2^{-k}]\right)} \\
&\leq 1 + \|f\|_{L_\infty^W([0,1])}.
\end{aligned}$$

Hence, each component of C_L^k is uniformly bounded in L and k, and so $\left\{C_L^k\right\}$ is uniformly bounded in L. Therefore, there exists a convergent subsequence $\left\{C_{L_n}^k\right\}$, and let C_k be the limit of $\left\{C_{L_n}^k\right\}$.

Thus,

$$\begin{aligned}\left\|f\left(\cdot+2^{-k}\right)-C_kf\left(\cdot\right)\right\|_{L_\infty^W\left([0,2^{-k}]\right)} &= \|f\left(\cdot+2^{-k}\right)-\mathcal{Y}_{L_n}\left(\cdot+2^{-k}\right)+\mathcal{Y}_{L_n}\left(\cdot+2^{-k}\right)\\ &\quad -C_{L_n}^k\mathcal{Y}_{L_n}+C_{L_n}^k\mathcal{Y}_{L_n}-C_k\mathcal{Y}_{L_n}+C_k\mathcal{Y}_{L_n}-C_kf\|_{L_\infty^W\left([0,2^{-k}]\right)}\\ &\leq \|f\left(\cdot+2^{-k}\right)-\mathcal{Y}_{L_n}\left(\cdot+2^{-k}\right)\|_{L_\infty^W\left([0,2^{-k}]\right)}\\ &\quad +\|C_{L_n}^k\mathcal{Y}_{L_n}-C_k\mathcal{Y}_{L_n}\|_{L_\infty^W\left([0,2^{-k}]\right)}\\ &\quad +\|C_k\mathcal{Y}_{L_n}-C_kf\|_{L_\infty^W\left([0,2^{-k}]\right)}\\ &\quad +\|\mathcal{Y}_{L_n}\left(\cdot+2^{-k}\right)-C_{L_n}^k\mathcal{Y}_{L_n}\|_{L_\infty^W\left([0,2^{-k}]\right)}.\end{aligned}$$

By assumption,

$$\left\|f\left(\cdot+2^{-k}\right)-\mathcal{Y}_{L_n}\left(\cdot+2^{-k}\right)\right\|_{L_\infty^W\left([0,2^{-k}]\right)} \leq \left\|f-\mathcal{Y}_{L_n}\left(\cdot+2^{-k}\right)\right\|_{L_\infty^W([0,1])} \to 0$$

as $n\to\infty$.

As C_k is the pointwise limit of $C_{L_n}^k$, $\left\|C_{L_n}^k-C_k\right\|_\infty \to 0$ as $n\to\infty$, and as

$$\|\mathcal{Y}_{L_n}\|_{L_\infty^W([0,1])} \leq \varepsilon+\|f\|_{L_\infty^W\left([0,2^{-k}]\right)} \leq 1+\|f\|_{L_\infty^W\left([0,2^{-k}]\right)},$$

it follows that

$$\left\|C_{L_n}^k\mathcal{Y}_{L_n}-C_k\mathcal{Y}_{L_n}\right\|_{L_\infty^W([0,2^{-k}])} \leq \left\|C_{L_n}^k-C_k\right\|_\infty \|\mathcal{Y}_{L_n}\|_{L_\infty^W([0,2^{-k}])} \to 0 \text{ as } n\to\infty.$$

By assumption, $\|\mathcal{Y}_{L_n}-f\|_{L_\infty^w\left([0,2^{-k}]\right)} \to 0$ as $n\to\infty$, hence

$$\|C_k\mathcal{Y}_{L_n}-C_kf\|_{L_\infty^W\left([0,2^{-k}]\right)} \leq \|C_k\|_\infty \|\mathcal{Y}_{L_n}-f\|_{L_\infty^W\left([0,2^{-k}]\right)} \to 0 \text{ as } n\to\infty.$$

Lastly, by Lemma 7.4, $\left\|\mathcal{Y}_{L_n}\left(\cdot+2^{-k}\right)-C_{L_n}^k\mathcal{Y}_{L_n}\right\|_{L_\infty^W\left([0,2^{-k}]\right)}=0$.

Therefore, for n large enough, $\left\|f\left(\cdot+2^{-k}\right)-C_kf\left(\cdot\right)\right\|_{L_\infty^W\left([0,2^{-k}]\right)}$ can be made arbitrarily small. Thus, $f\left(\cdot+2^{-k}\right)=C_k f(\cdot)$ on $[0,2^{-k})$.

$\Leftarrow$ Next, assume f satisfies (7.9). Fix L, and let $\mathcal{Y}_L=f_0$ on $\left[0,2^{-L}\right)$, where $f_0\in\mathbb{R}^W$ is such that $\|f-f_0\|_{L_\infty^W([0,h])}$ as $h\to 0^+$. The existence of f_0 follows from the assumption of essential continuity of f at the origin. Let $j\in\{0,\ldots,2^{L-1}\}$ where $(j)_2=\delta_1\cdots\delta_L$ its binary representation and where δ_ℓ in $C_\ell^{\delta_\ell}$ is understood as an exponent.

Then, use the matrices C_k of f to define $\mathcal{Y}_L(x)$ inductively for all $x\in[0,1]$:

$$\mathcal{Y}_L\left(\cdot+2^{-k}\right)\Big|_{[0,2^{-k})}=C_k\mathcal{Y}_L(\cdot)\Big|_{[0,2^{-k})},\ k=L,L-1,\ldots,1.$$

Thus, by Lemma 7.4, with $j=0$ and $C_L^k=C_k$, $k=1,\ldots,L$, it follows that $\mathcal{Y}_L\in\mathcal{S}_{W,L}$.

This is equivalent to

$$\mathcal{Y}_L\left(\cdot+j2^{-L}\right)\Big|_{[0,2^{-L})}=C_1^{\delta_1}\cdots C_L^{\delta_L}\mathcal{Y}_L(\cdot)\Big|_{[0,2^{-L})}.$$

Hence, by Lemma 7.6 and the identity $f\left(\cdot+j2^{-L}\right)\Big|_{[0,2^{-L})}=C_1^{\delta_1}\cdots C_L^{\delta_L}f(\cdot)\Big|_{[0,2^{-L})}$,

$$\begin{aligned}\|\mathcal{Y}_L\left(\cdot+j2^{-L}\right)-f\left(\cdot+j2^{-L}\right)\|_{L_\infty^W([0,2^{-L}])}&\le\max_{j\in\{0,\ldots,2^L-1\}}\left\|C_1^{\delta_1}\cdots C_L^{\delta_L}\right\|_\infty\|\mathcal{Y}_L-f\|_{L_\infty^W([0,2^{-L}])}\\&<\gamma\|\mathcal{Y}_L-f\|_{L_\infty^W([0,2^{-L}])}.\end{aligned}$$

As $\mathcal{Y}_L=f_0$ on $\left[0,2^{-L}\right)$, it follows that $\|\mathcal{Y}_L-f\|_{L_\infty^W([0,2^{-L}])}\to 0$ as $L\to\infty$.

Therefore, $\|\mathcal{Y}_L-f\|_{L_\infty^W([0,1])}\to 0$ as $L\to\infty$. □

Remark 7.2. *Condition (7.10) in Theorem 7.11 cannot be entirely removed. For example, consider $f(x)=\chi_{[1/2,1]}(x), x\in[0,1]$, which does not satisfy (7.10). A matrix C_1 satisfying (7.9) does not exist.*

Remark 7.3. *The matrices C_k in (7.9) are unique. For if C_k and $\widehat{C}_k$ are different, then*

$$0=f\left(\cdot+2^{-k}\right)-f\left(\cdot+2^{-k}\right)=\left(C_k-\widehat{C}_k\right)f\left(\cdot\right),$$

which contradicts (7.10).

Remark 7.4. *In the case of stationary CN, where $A_\ell = A$, for $\ell \in \mathbb{N}$, and*

$$A = \begin{cases} A^0, & x \in [0, 1/2) \\ A^1, & x \in [1/2, 1] \end{cases},$$

C_L^ℓ does not depend on L, and $C_L^\ell = C_\ell = (A^0)^{\ell-1} A^1 (A^0)^{-\ell}$.

Proposition 7.12. For each vector $f_0 \in \mathbb{R}^W$, there exists $f \in L_\infty^W([0,1])$ satisfying (7.9) and such that

$$\lim_{t \to 0^+} \|f - f_0\|_{L_\infty^W([0,t])} = 0$$

if and only if $C_k^{\delta_k} \cdots C_m^{\delta_m} \to \mathbb{I}_W$ as $k \to \infty$, i.e $\|C_k^{\delta_k} \cdots C_m^{\delta_m} - \mathbb{I}_W\|_\infty \to 0$.

Proof. $\Leftarrow$ Suppose $C_k^{\delta_k} \cdots C_m^{\delta_m} \to \mathbb{I}_W$ as $k \to \infty$. Let $f(0) = f_0$, define a sequence $\{f^k\}$ of piecewise constant vector functions on $[0, 1)$ by

$$f^k(\cdot)|_{[j2^{-k}, (j+1)2^{-k})} = C_1^{\delta_1} \cdots C_k^{\delta_k} f(0),$$

where $(j)_2 = \delta_1 \cdots \delta_k$ is the binary representation for $j \in \{0, \ldots, 2^k - 1\}$.

Then, $\{f^k\}$ is a Cauchy sequence. Let $m \in \mathbb{N}$, $m \geq k$, $i \in \{0, \ldots, 2^m - 1\}$, $(i)_2 = \delta_1 \cdots \delta_m$. Then,

$$(f^m - f^k)|_{[\frac{i}{2^m}, \frac{i+1}{2^m})} = C_1^{\delta_1} \cdots C_k^{\delta_k} \left(C_{k+1}^{\delta_{k+1}} \cdots C_m^{\delta_m} - \mathbb{I}_W \right) f(0).$$

This follows from the fact that for $j := \lfloor \frac{i}{2^{m-k}} \rfloor$ and $(j)_2 = \delta_1 \cdots \delta_k$, one has $[i2^{-m}, (i+1)2^{-m}) \subset [j2^{-k}, (j+1)2^{-k})$ and hence

$$f^k\Big|_{[i2^{-m}, (i+1)2^{-m})} = f^k\Big|_{[j2^{-k}, (j+1)2^{-k})} = C_1^{\delta_1} \cdots C_k^{\delta_k} f(0).$$

Thus, setting $\gamma = \sup_k \max_{\delta_{k+1},\dots,\delta_m} \left\| C_{k+1}^{\delta_{k+1}} \cdots C_m^{\delta_m} \right\|_\infty$,

$$\begin{aligned}\left\| f^m - f^k \right\|_{L_\infty^W([0,1])} &\le \max_{\delta_1,\dots,\delta_k} \left\| C_1^{\delta_1} \cdots C_k^{\delta_k} \right\|_\infty \max_{\delta_{k+1},\dots,\delta_m} \left\| C_{k+1}^{\delta_{k+1}} \cdots C_m^{\delta_m} - \mathbb{I}_W \right\|_\infty \|f_0\|_\infty \\ &\le \gamma \max_{\delta_{k+1},\dots,\delta_m} \left\| C_{k+1}^{\delta_{k+1}} \cdots C_m^{\delta_m} - \mathbb{I}_W \right\|_\infty \|f_0\|_\infty\end{aligned}$$

goes to 0 as $k \to \infty$. Therefore, $\left\{ f^k \right\}$ is a Cauchy sequence and let f be its limit. Note that $\gamma < \infty$, because $\left\| C_{k+1}^{\delta_{k+1}} \cdots C_m^{\delta_m} - \mathbb{I}_W \right\|_\infty \to 0$ as $k \to \infty$.

Next, fix K, then it follows from the definition of f^m, $f^m\left(\cdots + 2^{-K}\right) = C_K g^m(\cdot)$ on $\left[0, 2^{-K}\right)$, $m \ge K$. Passing to the limit $f^m \to f$, f satisfies (7.9) for K, and thus f satisfies (7.9) for all $k \in \mathbb{N}$.

To show that f is essentially continuous at the origin, fix $\varepsilon > 0$, select k such that $\|f - f^k\|_{L_\infty^W([0,2^{-k}])} < \varepsilon$. Then,

$$\begin{aligned}\|f - f_0\|_{L_\infty^W([0,2^{-k}])} &= \left\| f - f^k + f^k - f_0 \right\|_{L_\infty^W([0,2^{-k}])} \\ &\le \left\| f - f^k \right\|_{L_\infty^W([0,2^{-k}])} + \left\| f^k - f_0 \right\|_{L_\infty^W([0,2^{-k}])} \\ &< \varepsilon + 0.\end{aligned}$$

Thus, $\lim_{h \to 0^+} \|f - f_0\|_{L_\infty^W([0,h])} = 0$.

$\Rightarrow$ Let $m, k \in \mathbb{N}, m \ge k, j \in \{0, \dots, 2^{m-k+1} - 1\}$, where $(j)_2 = \delta_k \cdots \delta_m$ is its binary representation. As f satisfies (7.9), on $[0, 2^{-m})$,

$$\begin{aligned}f(\cdot + j2^{-m}) &= f(\cdot + (\delta_k \cdots \delta_m)\, 2^{-m}) \\ &= f(\cdot + \delta_m 2^{-m} + \cdots + \delta_k 2^{-k}) \\ &= C_k^{\delta_k} f(\cdot + \delta_m 2^{-m} + \cdots + \delta_{k-1} 2^{-k-1}).\end{aligned}$$

Continuing in this way, $f(\cdot + (\delta_k \cdots \delta_m)\, 2^{-m}) = C_k^{\delta_k} \cdots C_m^{\delta_m} f(\cdot)$ on $[0, 2^{-m})$.

For $x \in [0, 2^{-m})$,

$$f(x + j2^{-m}) - f(x) = \left(C_k^{\delta_k} \cdots C_m^{\delta_m} - \mathbb{I}_W\right) f(x). \tag{7.11}$$

Then for $x = 0$, $f(j2^{-m}) - f(0) = \left(C_k^{\delta_k} \cdots C_m^{\delta_m} - \mathbb{I}_W\right) f(0)$. Let $f_0^1, \dots, f_0^W$ be a basis for $\mathbb{R}^W$. For example, take $F(0) \coloneqq F_0 = (f_0^1, \dots, f_0^W) = \mathbb{I}_W$. Then, (7.11) implies $F(j2^{-m}) - F(0) = \left(C_k^{\delta_k} \cdots C_m^{\delta_m} - \mathbb{I}_W\right) F(0)$, where $F(x) \coloneqq (f^1(x), \dots, f^W(x))$ and where f^i is the function satisfying the periodicity condition corresponding to $f_0^i, i = 1, \dots, W$. Hence,

$$\begin{aligned} C_k^{\delta_k} \cdots C_m^{\delta_m} - \mathbb{I}_W &= \left(F\left(j2^{-m}\right) - F_0\right) F^{-1}(0) \\ &= F\left(j2^{-m}\right) - \mathbb{I}_W \\ &\to 0 \end{aligned}$$

because $f^i(j2^{-m}) \to f^i(0) = f_0^i$ as $k \to \infty$, given that $j2^{-m} \to 0$ as $k \to \infty, j \in \{0, \dots, 2^{m-k+1} - 1\}$. □

Corollary 7.5. *Let $W, k \in \mathbb{N}$ and $C_k \in \mathbb{R}^{W \times W}$. Suppose that for each vector $f_0 \in \mathbb{R}^W$, there exists $f \in L_\infty^W([0,1])$ satisfying (7.9) and such that*

$$\lim_{t \to 0^+} \|f - f_0\|_{L_\infty^W([0,t])} = 0$$

Let $\mathcal{S}_W(\mathcal{C})$, $\mathcal{C} = \{C_1, C_2, \dots\}$ be the set such that

$$f(\cdot + 2^{-k})|_{[0,2^{-k})} = C_k f(\cdot)|_{[0,2^{-k})}, \ \textit{for all } k \in \mathbb{N}.$$

Then,

(a) $\mathcal{S}_W(\mathcal{C})$ is a linear space.

(b) $f \in \mathcal{S}_W(\mathcal{C})$ is uniquely determined by its essential value f_0 at 0.

(c) $\dim(\mathcal{S}_W(\mathcal{C})) = W$.

Proof. (a) is straightforward.

(b) Suppose that $f, g \in \mathcal{S}_W(\mathcal{C})$ are such that $f_0 = g_0$. Then, $h = f - g \in \mathcal{S}_W(\mathcal{C})$ is such that $h_0 = 0$. Let $\varepsilon > 0$ and $k \in \mathbb{N}$ be such that

$$\|h\|_{L_\infty^W([0,2^{-k}])} = \|h - 0\|_{L_\infty^W([0,2^{-k}])} < \varepsilon.$$

Thus,

$$\begin{aligned}\|h\|_{L_\infty^W([j2^{-k},(j+1)2^{-k}])} &= \left\| C_1^{\delta_1} \cdots C_k^{\delta_k} h \right\|_{L_\infty^W([0,2^{-k}])} \\ &\leq \left\| C_1^{\delta_1} \cdots C_k^{\delta_k} \right\|_\infty \|h\|_{L_\infty^W([0,2^{-k}])} \\ &\leq \gamma\varepsilon,\end{aligned}$$

where $(j)_2 = \delta_1 \cdots \delta_k$ and $\gamma < \infty$ follows from the proof of Proposition 7.12. Thus, $\|h\|_{L_\infty^W([0,2^{-k}])} < \varepsilon$ and ε can be made arbitrarily small so that h is 0.

(c) For $a \in \mathbb{R}^W$, let f_a denote the unique function in $\mathcal{S}_W(\mathcal{C})$ such that $(f_a)_0 = a$. Let $\{a_1, \ldots, a_W\}$ form a basis of $\mathbb{R}^W$. We will show that $\{f_{a_1}, \ldots, f_{a_W}\}$ forms a basis for $\mathcal{S}_W(\mathcal{C})$.

Clearly, $\dim(\mathcal{S}_W(\mathcal{C})) \leq W$. For if $f_{a_{w+1}} \in \mathcal{S}_W(\mathcal{C})$, then $(f_{a_{W+1}})_0 \in \operatorname{span}\{(f_{a_1})_0, \ldots, (f_{a_w})_0\}$. Then, $a_{w+1} = f_{a_{W+1}}(0) = c_1 f_{a_1}(0) + \cdots + c_w f_{a_W}(0)$ implies that $f_{a_{W+1}}(x) = c_1 f_{a_1}(x) + \cdots + c_w f_{a_W}(x)$ for almost every x because $a = f(0)$ uniquely determines $f(x)$. $\dim(\mathcal{S}_W(\mathcal{C})) \not< W$, as it cannot happen that $c_1 f_{a_1} + \cdots + c_w f_{a_W} = 0$ because then $c_1 (f_{a_1})_0 + \cdots + c_W (f_{a_W})_0 = c_1 a_1 + \cdots + c_W a_W = 0$ which would imply $c_1 = \cdots = c_W = 0$.

□

Example 7.13. Consider $f(x) = (1, x)^T$. f can be generated by a stationary CN,

$$\mathcal{Y}_L = A_L \left(A_{L-1} \circ \alpha\right) \cdots \left(A_1 \circ \alpha^{L-1}\right) \left(\mathcal{Y}_0 \circ \alpha^L\right),$$

where $\mathcal{Y}_0 = (1, x)^T$, $A_\ell = A$, for $l = 1, \ldots, L$, and

$$A(x) = \begin{cases} A^0, & x \in \left[0, \frac{1}{2}\right) \\ A^1, & x \in \left[\frac{1}{2}, 1\right] \end{cases} = \begin{cases} \begin{pmatrix} 1 & 0 \\ 0 & 1/2 \end{pmatrix}, & x \in \left[0, \frac{1}{2}\right) \\ \begin{pmatrix} 1 & 0 \\ 1/2 & 1/2 \end{pmatrix}, & x \in \left[\frac{1}{2}, 1\right] \end{cases}.$$

Clearly, f satisfies the conditions of Theorem 7.11. Then, by Theorem 7.11 and Remark 7.4, f satisfies (7.9), with $C_k = \begin{pmatrix} 1 & 0 \\ 2^{-k} & 1 \end{pmatrix}$, $k \in \mathbb{N}$. These matrices are consistent with f. Indeed, let $k \in \mathbb{N}$ and $j \in \{0, \ldots, 2^k - 1\}$, $(j)_2 = \delta_1 \cdots \delta_k$, then, for $x = j2^{-k} \in [0, 1)$,

$$\begin{aligned} f(x) &= f\left(j2^{-k}\right) \\ &= C_1^{\delta_1} \cdots C_k^{\delta_k} f_0 \\ &= \begin{pmatrix} 1 & 0 \\ \frac{\delta_1}{2} & 1 \end{pmatrix} \cdots \begin{pmatrix} 1 & 0 \\ \frac{\delta_k}{2} & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 \\ \frac{\delta_1 \cdots \delta_k}{2^m} & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ j2^{-m} \end{pmatrix} \\ &= \begin{pmatrix} 1 \\ x \end{pmatrix}. \end{aligned}$$

Note also that for $m \geq k$ and $\delta_k, \ldots, \delta_m \in \{0, 1\}$,

$$C_k^{\delta_k} \cdots C_m^{\delta_m} = \begin{pmatrix} 1 & 0 \\ \frac{\delta_k}{2^k} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{\delta_{k+1}}{2^{k+1}} & 1 \end{pmatrix} \cdots \begin{pmatrix} 1 & 0 \\ \frac{\delta_m}{2^m} & 1 \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 \\ \frac{\delta_k}{2^k} + \cdots + \frac{\delta_m}{2^m} & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 \\ \frac{\delta_k \cdots \delta_m}{2^{m+k-1}} & 1 \end{pmatrix}$$

approaches $\mathbb{I}_2$ as k goes to infinity.

Example 7.14. Consider $f(x) = (1, \phi(x))^T$, where $\phi(x) = \begin{cases} 2x, \text{ if } x \in [0, 1/2) \\ 2 - 2x, \text{ if } x \in [1/2, 1] \end{cases}$.

f satisfies (7.9), with $C_1 = \begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix}$ and $C_k = \begin{pmatrix} 1 & 0 \\ 2^{-k+1} & 1 \end{pmatrix}$, for $k \geq 2$. One can verify that for $j \in \{0, \ldots, 2^{k-1} - 1\}$ and $(j)_2 = \delta_1 \cdots \delta_k = 0\delta_2 \cdots \delta_k, x = j2^{-k} \in [0, 1/2)$, it follows that

$$f(x) = f\left(j2^{-k}\right) = C_2^{\delta_k} \cdots C_2^{\delta_k} f(0) = \begin{pmatrix} 1 \\ 2x \end{pmatrix}$$

and, for $j \in \{2^{k-1}, \ldots, 2^k - 1\}$,

$$f(x) = C_1 C_2^{\delta_k} \cdots C_2^{\delta_k} f(0) = \begin{pmatrix} 1 \\ 2 - 2x \end{pmatrix}.$$

Also, for $m \geq k > 1$, the product

$$C_k^{\delta_k} \cdots C_m^{\delta_m} = \begin{pmatrix} 1 & 0 \\ \frac{\delta_k}{2^{k-1}} & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ \frac{\delta_{k+1}}{2^k} & 1 \end{pmatrix} \cdots \begin{pmatrix} 1 & 0 \\ \frac{\delta_m}{2^{m-1}} & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 \\ \frac{\delta_k}{2^{k-1}} + \cdots + \frac{\delta_m}{2^{m-1}} & 1 \end{pmatrix}$$
$$= \begin{pmatrix} 1 & 0 \\ \frac{\delta_k \cdots \delta_m}{2^{m-1}} & 1 \end{pmatrix}$$

approaches $\mathbb{I}_2$ as k goes to infinity.

Example 7.15. Consider $f(x) = \chi_{[0,1/2)}$. f can be generated by a CN,

$$\mathcal{Y}_L = A_L \left(A_{L-1} \circ \alpha\right) \cdots \left(A_1 \circ \alpha^{L-1}\right),$$

where $\mathcal{Y}_0 = 1$, $A_\ell(x) = 1$ for $\ell = 1, \ldots, L-1$ and all $x \in [0,1]$, and

$$A_L(x) = \begin{cases} A_L^0, & x \in \left[0, \frac{1}{2}\right) \\ A_L^1, & x \in \left[\frac{1}{2}, 1\right] \end{cases} = \begin{cases} 1, & x \in \left[0, \frac{1}{2}\right) \\ 0, & x \in \left[\frac{1}{2}, 1\right] \end{cases}.$$

Then, by Theorem 7.11, f satisfies (7.9), with $C_1 = 0$ and $C_k = 1$, for $k > 1$. Hence, $C_k^{\delta_k} \cdots C_m^{\delta_m} \to 1$ as $k \to \infty$ is trivially true.

Theorem 7.16. *Suppose that $f \in L_\infty^W([0,1])$ has the LUIC property, f is essentially continuous at the origin with $\lim_{t \to 0^+} \|f - f_0\|_{L_\infty^W([0,t])} = 0$, $f_0 \in \mathbb{R}^W$. Moreover, f is such that (7.9) holds for matrices $C_k,\ k \in \mathbb{N}$. Then, $f \in (C([0,1]))^W$ if and only if $C_k f_0 = C_{k+1}^2 f_0$, for all $k \in \mathbb{N}$.*

Proof. $\Rightarrow$ Assume $f \in (C([0,1]))^W$. Then $f_0 = f(0)$ and by (7.9),

$$f\left(\frac{1}{2^k}\right) = f\left(0 + \frac{1}{2^k}\right) = C_k f(0),$$

and

$$f\left(\frac{1}{2^k}\right) = f\left(\frac{1}{2^{k+1}} + \frac{1}{2^{k+1}}\right) = C_{k+1} f\left(\frac{1}{2^{k+1}}\right) = C_{k+1} f\left(0 + \frac{1}{2^{k+1}}\right) = C_{k+1}^2 f(0).$$

Therefore, $C_k f(0) = C_{k+1}^2 f(0)$ for all k.

$\Leftarrow$ Consider a sequence $\{p^k\}$ of the piecewise linear interpolants to the values $p^k\left(\frac{j}{2^k}\right) = C_1^{\delta_1} \cdots C_k^{\delta_k} f_0$, for $(j)_2 = \delta_1 \cdots \delta_k$, where $j \in \{0, \ldots, 2^k - 1\}$. For $j = 2^k$, $p^k(1) = C_1^2 f_0 = C_2^4 f_0$ by the condition on C_k.

Let $m \in \mathbb{N}$, $m \geq k$, then

$$\left(p^m - p^k\right)|_{\left[\frac{j}{2^m}, \frac{j+1}{2^m}\right]} = C_1^{\delta_1} \cdots C_k^{\delta_k} \left(C_{k+1}^{\delta_{k+1}} \cdots C_m^{\delta_m} - \mathbb{I}_W\right) f_0.$$

Thus,

$$\begin{aligned} \left\|p^m - p^k\right\|_{L_\infty^W([0,1])} &\leq \max_{\delta_1,\ldots,\delta_k} \left\|C_1^{\delta_1} \cdots C_k^{\delta_k}\right\|_\infty \max_{\delta_{k+1},\ldots,\delta_m} \left\|\left(C_{k+1}^{\delta_{k+1}} \cdots C_m^{\delta_m} - \mathbb{I}_W\right) f_o\right\|_\infty \\ &\leq \gamma \max_{\delta_{k+1},\ldots,\delta_m} \left\|f(\cdot + (\delta_{k+1} \cdots \delta_m)\, 2^{-m}) - f(\cdot)\right\|_{L_\infty^W([0,2^{-m}])} \end{aligned}$$

by Lemma 7.10. However,

$$\begin{aligned} \left\|f(\cdot + (\delta_{k+1} \cdots \delta_m)\, 2^{-m}) - f(\cdot)\right\|_{L_\infty^W([0,2^{-m}])} &\leq \left\|f(\cdot + (\delta_{k+1} \cdots \delta_m)\, 2^{-m}) - f_0\right\|_{L_\infty^W([0,2^{-m}])} \\ &\quad + \left\|f(\cdot) - f_0\right\|_{L_\infty^W([0,2^{-m}])} \\ &\leq \left\|f(\cdot) - f_0\right\|_{L_\infty^W([0,2^{-k}])} + \left\|f(\cdot) - f_0\right\|_{L_\infty^W([0,2^{-k}])}, \end{aligned}$$

which goes to 0 as $k \to \infty$. Hence, $\left\{p^k\right\}$ is a Cauchy sequence and converges to $g \in (C([0,1]))^W$ such that $g(0) = f_0$. As p^k satisfies equation (7.9), it follows that g also satisfies equation (7.9).

As f, g satisfy equation (7.9), $(f - g)\left(\cdot + 2^{-k}\right) = C_k(f - g)(\cdot)$, thus $h = f - g$ also satisfies equation (7.9) and $h_0 = f_0 - g_0 = f_0 - f_0 = 0$. Hence $h = 0$ by Corollary 7.5 (b), or $f = g \in (C([0,1]))^W$

□

Corollary 7.6. *$\mathcal{S}_W(\mathcal{C}) \subset (C([0,1]))^W$ if and only if $C_k = C_{k+1}^2$ for all $k \in \mathbb{N}$.*

Proof. Let $\{f_1, \ldots, f_W\}$ be a basis of $\mathcal{S}_W(\mathcal{C})$, then by Corollary 7.5 $\{(f_1)_0, \ldots, (f_W)_0\}$ is a basis for $\mathbb{R}^W$. Hence the conditions $C_k(f_i)_0 = C_{k+1}^2(f_i)_0$ for all i implies $C_k = C_{k+1}^2$. □

Next, recall the following about the principal square root of a matrix.

Proposition 7.17. [58] Let $A \in \mathbb{C}^{n\times n}$ have no eigenvalues on $\mathbb{R}^-$. There is a unique square root

X of A, i.e such that $X^2 = A$, all of whose eigenvalues lie in the open right half-plane $\mathbb{H}_+$ of $\mathbb{C}$. X is called the *principal square root* of A and write $X = A^{1/2}$. If A is real, then $A^{1/2}$ is real.

Proposition 7.18. [58] Let $A \in \mathbb{C}^{n\times n}$ have no eigenvalues on $\mathbb{R}^-$. There is a unique pth root X, $p > 2$, of A all of whose eigenvalues lie in the segment $\{z| -\pi/p < arg(z) < \pi/p\} \subset \mathbb{C}$. X is called the *principal pth root* of A and write $X = A^{1/p}$. If A is real, then $A^{1/p}$ is real.

The notation $A^{2^{-k}} := \left(A^{2^{-k+1}}\right)^{1/2}$, $k \in \mathbb{N}$, is used for the iterates of the principal square root of a matrix. It is understood that $A^{2^0} = A$.

Example 7.19. A matrix can have no square roots, finitely many square roots, or infinitely many square roots:

1. The matrix $A = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ does not have a square root.

2. The matrix $A = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$ has four square roots,

$$\begin{pmatrix} 1 & 0 \\ 0 & \sqrt{2} \end{pmatrix}, \quad \begin{pmatrix} -1 & 0 \\ 0 & \sqrt{2} \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 \\ 0 & -\sqrt{2} \end{pmatrix}, \quad \begin{pmatrix} -1 & 0 \\ 0 & -\sqrt{2} \end{pmatrix}.$$

   The principal square root is $A^{1/2} = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{2} \end{pmatrix}$.

3. All square roots of $\mathbb{I}_2$ are

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} a & b \\ c & -a \end{pmatrix},$$

   whenever $a, b, c \in \mathbb{R}$ are such that $a^2 + bc = 1$. However, the only principal square root of $\mathbb{I}_2$ in this case is $\mathbb{I}_2^{1/2} = \mathbb{I}_2$.

Next, we present several useful facts pertaining to principal square roots.

Proposition 7.20. Let $A \in \mathbb{R}^{n\times n}$ be a nonsingular square matrix whose eigenvalues are not on the negative real axis including the origin. Then

(a) A has a unique principal square root, $A^{1/2}$.

(b) If $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$ are eigenvalues of A (counting multiplicities), then the eigenvalues of $A^{1/2}$ are $\lambda_1^{1/2}, \ldots, \lambda_n^{1/2}$, the principal square roots of $\lambda_1, \ldots, \lambda_n$.

(c) $\lim_{k\to\infty} A^{2^{-k}} = \mathbb{I}_n$.

(d) $\lim_{k\to\infty} 2^k(A^{2^{-k}} - \mathbb{I}_n) = B$, for some matrix $B \in \mathbb{C}^{n\times n}$.

(e) Matrix B above is real if all eigenvalues of A are in $\mathbb{H}_+$.

Proof. Properties (a) and (b) are well known [58]. Property (c) follows from (d).

For (d), write A in Jordan canonical form as $A = PJP^{-1}$, where $P \in \mathbb{R}^{n\times n}$ is nonsingular, and J is the Jordan matrix of the form

$$J = \begin{pmatrix} J_1 & 0 & \cdots & 0 & 0 \\ 0 & J_2 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & & J_{m-1} & \vdots \\ 0 & 0 & \cdots & 0 & J_m \end{pmatrix},$$

where $J_\ell \in \mathbb{R}^{n_\ell \times n_\ell}$, $\ell = 1, \ldots, m, n_1 + \cdots + n_m = n$, are the usual Jordan block matrices of the form

$$J_\ell = \begin{pmatrix} \mu_\ell & 1 & 0 & \cdots & 0 \\ 0 & \mu_\ell & 1 & \cdots & 0 \\ 0 & 0 & \ddots & 1 & 0 \\ \vdots & \vdots & & \mu_\ell & 1 \\ 0 & 0 & \cdots & 0 & \mu_\ell \end{pmatrix}.$$

where the eigenvalues λ_i are ordered such that $\mu_1 = \lambda_1 = \cdots = \lambda_{n_1}, \mu_2 = \lambda_{n_1+1} = \cdots = \lambda_{n_1+n_2}, \ldots, \mu_m = \lambda_{n-n_\ell+1} = \cdots = \lambda_n$.

With this notation, one can show that $A^{2^{-k}} = PJ^{2^{-k}}P^{-1}$, with

$$J^{2^{-k}} = \begin{pmatrix} J_1^{2^{-k}} & 0 & \cdots & 0 & 0 \\ 0 & J_2^{2^{-k}} & \cdots & 0 & 0 \\ 0 & 0 & \ddots & 0 & 0 \\ \vdots & \vdots & & J_{m-1}^{2^{-k}} & \vdots \\ 0 & 0 & \cdots & 0 & J_m^{2^{-k}} \end{pmatrix},$$

where

$$J_\ell^{2^{-k}} = \begin{pmatrix} a_0^{(\ell)} & a_1^{(\ell)} & a_2^{(\ell)} & \cdots & \cdots & \cdots & a_{n_\ell-1}^{(\ell)} \\ 0 & a_0^{(\ell)} & a_1^{(\ell)} & \ddots & \cdots & \cdots & a_{n_\ell-2}^{(\ell)} \\ 0 & 0 & \ddots & a_1^{(\ell)} & \ddots & \cdots & a_{n_\ell-3}^{(\ell)} \\ \vdots & \vdots & \cdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \vdots & \cdots & \cdots & a_0^{(\ell)} & a_1^{(\ell)} & a_2^{(\ell)} \\ \vdots & \vdots & \cdots & \cdots & \cdots & a_0^{(\ell)} & a_1^{(\ell)} \\ 0 & 0 & \cdots & \cdots & \cdots & 0 & a_0^{(\ell)} \end{pmatrix}, \quad k \in \mathbb{N},$$

with $a_0^{(\ell)} = \mu_\ell^{2^{-k}}$, and $a_i^{(\ell)} = \binom{2^{-k}}{i}\mu_\ell^{-i}\mu_\ell^{2^{-k}}$, $i = 1, \ldots, n_\ell - 1$ and $\binom{s}{j} := s(s-1)\cdots(s-j+1)/i!$, $s \in \mathbb{R}$, $j \in \mathbb{N}$.

The above facts mean that to show that $\lim_{k\to\infty} 2^k(A^{2^{-k}} - \mathbb{I}_n)$ exists, we must prove that

(1) $\lim_{k\to\infty} 2^k(a_0^{(\ell)} - 1)$ exists for all ℓ, and

(2) $\lim_{k\to\infty} 2^k a_i^{(\ell)}$ exists for all ℓ and all $i = 1, \ldots, n_\ell - 1$.

As for (1), setting $\mu_\ell = |\mu_\ell|e^{i\phi_\ell}$, for some $\phi_\ell \in [0, 2\pi)$, and noting that $\mu_\ell \in \mathbb{H}_+$, and hence $|\mu_\ell| > 0$, we have

$$\lim_{k\to\infty} 2^k(a_0^{(\ell)} - 1) = \lim_{t\to 0^+} \frac{|\mu_\ell|^t e^{it\phi_\ell} - 1}{t} = \ln|\mu_\ell| + i\phi_\ell.$$

As for (2), it follows by direct computation that

$$\lim_{k\to\infty} 2^k a_i^{(\ell)} = \lim_{k\to\infty} 2^k \binom{2^{-k}}{i} \lim_{k\to\infty} \mu_\ell^{-i}\mu_\ell^{2^{-k}} = \frac{(-1)^{i+1}}{i}\mu_\ell^{-i}.$$

Consequently, this proves the existence of a matrix B referred to in part (d).

Part (e) follows for Proposition 7.18. □

The next result expands on Corollary 7.6.

Theorem 7.21. *Suppose that matrices $\mathcal{C} = \{C_1, C_2, \ldots\} \subset \mathbb{R}^{W\times W}$ are given such that for all $f_0 \in \mathbb{R}^W$ there exists a vector function $f = f(\mathcal{C}, f_0) \in \mathcal{S}_W(\mathcal{C})$ satisfying (7.9) and whose essential value at 0 is f_0. The following are equivalent:*

(a) For all $f_0 \in \mathbb{R}^W$, $f(\mathcal{C}, f_0) \in (C([0,1]))^W$;

(b) $C_k = C_{k+1}^2, k \in \mathbb{N}$;

(c) $C_k = C_{k+1}^2, k \in \mathbb{N}$, and there exists a $K \in \mathbb{N}$ such that $C_{k+1} = C_k^{1/2}$, for all $k \geq K$;

(d) For all $f_0 \in \mathbb{R}^W$, $f(\mathcal{C}, f_0) \in (C^1([0,1]))^W$.

Proof. By Corollary 7.6 (a) $\Rightarrow$ (b).

To show that (b) $\Rightarrow$ (c), note that the matrices C_k are eventually such that all their eigenvalues are close to 1 given that $C_k \to \mathbb{I}_W$ as $k \to \infty$ by Proposition 7.12. Thus, in particular, those eigenvalues are all contained in $\mathbb{H}_+$ for $k \geq K$, for K large enough. This means that for $k \geq K$, $C_k = C_{k+1}^2$ is equivalent to $C_{k+1} = C_k^{1/2}$ since such matrices $C_k^{1/2}$, whose eigenvalues are in $\mathbb{H}_+$, are unique by Proposition 7.17.

As for the implication (c) $\Rightarrow$ (d), first note that by $C_k = C_{k+1}^2$ for $k \in \mathbb{N}$ implies commutativity $C_iC_j = C_jC_i$, for $i, j \in \mathbb{N}$. Then, $(C_i - \mathbb{I}_W)\, C_j = C_j\,(C_i - \mathbb{I}_W)$.

Define $F : [0,1] \to \mathbb{R}^{W\times W}$, a matrix function whose columns are individual f's and such that $F(0) = F_0$ is invertible. For $\ell \in \mathbb{N}$, consider the sequence $\left\{F\left(\frac{j+1}{2^\ell}\right) - F\left(\frac{j}{2^\ell}\right)\right\}_{j=0}^{2^{\ell-1}}$. Because of

(7.9) applied to the columns of F, F satisfies $F\left(\frac{j}{2^\ell}+\frac{1}{2^\ell}\right)=C_\ell F\left(\frac{j}{2^\ell}\right)$. As $F\left(\frac{j}{2^\ell}\right)$ can be written as a product of C_i's, which then commute with $C_\ell-\mathbb{I}_W$, then

$$\begin{aligned}\left\{F\left(\frac{j+1}{2^\ell}\right)-F\left(\frac{j}{2^\ell}\right)\right\}_{j=0}^{2^{\ell-1}}&=\left\{\left(C_\ell-\mathbb{I}_W\right)F\left(\frac{j}{2^\ell}\right)\right\}_{j=0}^{2^{\ell-1}}\\&=\left\{F\left(\frac{j}{2^\ell}\right)\right\}_{j=0}^{2^{\ell-1}}F^{-1}(0)\left(C_\ell-\mathbb{I}_W\right)F(0)\end{aligned}\tag{7.12}$$

For $\ell\in\mathbb{N}$, let $G^\ell:[0,1]\to\mathbb{R}^{W\times W}$ be the piecewise constant function defined as

$$G^\ell(x)=2^\ell\left(F\left(\frac{j+1}{2^\ell}\right)-F\left(\frac{j}{2^\ell}\right)\right),\ \text{if } x\in\left[\frac{j}{2^\ell},\frac{j+1}{2^\ell}\right).$$

To show that $G^\ell\to G$, where G is continuous, let $G:=FD$, where

$$\begin{aligned}D&:=\lim_{\ell\to\infty}F^{-1}(0)2^\ell\left(C_\ell-\mathbb{I}_W\right)F(0)\\&=F^{-1}(0)\left(\lim_{\ell\to\infty}2^\ell\left(C_\ell-\mathbb{I}_W\right)\right)F(0),\end{aligned}$$

which exists by Proposition 7.20 . By equation (7.12), $G_\ell(x)=F\left(\frac{j}{2^\ell}\right)F^{-1}(0)2^\ell\left(C_\ell-\mathbb{I}_W\right)F(0)$ if $x\in\left[\frac{j}{2^\ell},\frac{j+1}{2^\ell}\right)$. Fix $x\in[0,1)$ and let $x=0.\delta_1\delta_2\cdots$ be its binary expansion and j_ℓ be such that $x\in\left[\frac{j_\ell}{2^\ell},\frac{j_\ell+1}{2^\ell}\right)$. Then,

$$\begin{aligned}\lim_{\ell\to\infty}\left|G^\ell(x)-G(x)\right|&=\lim_{\ell\to\infty}\left|F\left(\frac{j_\ell}{2^\ell}\right)F^{-1}(0)2^\ell\left(C_\ell-\mathbb{I}_W\right)F(0)-F(x)D\right|\\&=\lim_{\ell\to\infty}\left|F\left(\frac{j_\ell}{2^\ell}\right)F^{-1}(0)2^\ell\left(C_\ell-\mathbb{I}_W\right)F(0)-F\left(\frac{j_\ell}{2^\ell}\right)D\right.\\&\left.+F\left(\frac{j_\ell}{2^\ell}\right)D-F(x)D\right|\\&\le\lim_{\ell\to\infty}\left|F\left(\frac{j_\ell}{2^\ell}\right)F^{-1}(0)2^\ell\left(C_\ell-\mathbb{I}_W\right)F(0)-F\left(\frac{j_\ell}{2^\ell}\right)D\right|\\&+\lim_{\ell\to\infty}\left|F\left(\frac{j_\ell}{2^\ell}\right)D-F(x)D\right|\\&\le\|F\|_\infty\lim_{\ell\to\infty}\left|F^{-1}(0)2^\ell\left(C_\ell-\mathbb{I}_W\right)F(0)-D\right|\end{aligned}$$

$$+\lim_{\ell\to\infty}\left|F\left(\frac{j_\ell}{2^\ell}\right)-F(x)\right|\|D\|_\infty,$$

where $\lim\limits_{\ell\to\infty}\left|F^{-1}(0)2^\ell\left(C_\ell-\mathbb{I}_W\right)F(0)-D\right|=0$ by the definition of D, and by continuity of F, $\lim\limits_{\ell\to\infty}\left|F\left(\frac{j_\ell}{2^\ell}\right)-F(x)\right|=0$. Hence G^ℓ converges to G and G is continuous as $G=FD$.

Next, define $H(x):=\int_0^x G(t)dt+F(0)$, $x\in[0,1]$. As $\{G^\ell\}$ is uniformly bounded and G^ℓ converges pointwise to G, it follows $\int_0^x G(t)dt=\lim\limits_{\ell\to\infty}\int_0^x G^\ell(t)dt$ by the Dominated Convergence Theorem [59].

Let $x=\frac{j}{2^\ell}$ be a dyadic point in $[0,1)$, $j\in\{0,\dots,2^\ell-1\}$, $\ell\in\mathbb{N}$. Then

$$\begin{aligned}
H(x)=H\left(\frac{j}{2^\ell}\right)&=\sum_{k=0}^{j-1}\int_{\frac{k}{2^\ell}}^{\frac{k+1}{2^\ell}}G(t)dt+F(0)\\
&=\lim_{m\to\infty}\sum_{k=0}^{j-1}\int_{\frac{k}{2^\ell}}^{\frac{k+1}{2^\ell}}G^{\ell+m}(t)dt+F(0)\\
&=\lim_{m\to\infty}\sum_{i=0}^{2^mj-1}\int_{\frac{i}{2^{\ell+m}}}^{\frac{i+1}{2^{\ell+m}}}G^{\ell+m}(t)dt+F(0)\\
&=\lim_{m\to\infty}\sum_{i=0}^{2^mj-1}2^{-(l+m)}G^{l+m}\left(\frac{i}{2^{l+m}}\right)+F(0)\\
&=\lim_{m\to\infty}\sum_{i=0}^{2^mj-1}2^{-(l+m)}\left(2^{(l+m)}\left(F\left(\frac{i+1}{2^{\ell+m}}\right)-F\left(\frac{i}{2^{\ell+m}}\right)\right)\right)+F(0)\\
&=\lim_{m\to\infty}F\left(\frac{2^mj}{2^{\ell+m}}\right)\\
&=F\left(\frac{j}{2^\ell}\right)=F(x).
\end{aligned}$$

Hence, $H(x)=F(x)$ for all dyadic points in $[0,1)$, so they are equal on $[0,1]$ by continuity of F and H. Therefore, $F'=H'=G$, or $F\in\left(C^1[0,1]\right)^{W\times W}$. Consequently, all columns of F are also C^1 vector functions. $\square$

Example 7.22. Recall Example 3.2 and set

$$A^0 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & 1/4 \end{pmatrix}, \quad A^1 = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1/2 & 0 \\ 1/4 & 1/2 & 1/4 \end{pmatrix}.$$

Then,

$$C_1 = A^1(A^0)^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 1/2 & 1 & 0 \\ 1/4 & 1 & 1 \end{pmatrix}, \quad C_2 = C_1^{1/2} = \begin{pmatrix} 1 & 0 & 0 \\ 1/4 & 1 & 0 \\ 1/16 & 1/2 & 1 \end{pmatrix}, \quad C_k = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2^k} & 1 & 0 \\ \frac{1}{4^k} & \frac{1}{2^{k-1}} & 1 \end{pmatrix}.$$

The Jordan form for C_1 is $C_1 = PJP^{-1}$, where

$$P = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1/2 & 0 \\ 1/2 & 1/4 & 0 \end{pmatrix}, \quad J = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Note that C_1 has a triple eigenvalue of 1, and J consists of a single Jordan block. Using Proposition 7.20, with $\mu_1 = 1$, we obtain

$$J^{1/2} = \begin{pmatrix} 1 & \binom{1/2}{1} & \binom{1/2}{2} \\ 0 & 1 & \binom{1/2}{1} \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1/2 & -1/8 \\ 0 & 1 & 1/2 \\ 0 & 0 & 1 \end{pmatrix},$$

and hence

$$C_2 = C_1^{1/2} = PJ^{1/2}P^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 1/4 & 1 & 0 \\ 1/16 & 1/2 & 1 \end{pmatrix}.$$

In general, we get

$$C_k = C_1^{2^{-k+1}} = PJ^{2^{-k+1}}P^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2^k} & 1 & 0 \\ \frac{1}{4^k} & \frac{1}{2^{k-1}} & 1 \end{pmatrix}.$$

One can verify that $C_2A^0 = A^0C_1 = A^0C_2^2$ and that $C_{k+1} = C_k^{1/2}, k \in \mathbb{N}$. Hence all corresponding functions $f(\mathcal{C}, f_0)$ are not only continuous, but also smooth by Theorem 7.21. This is consistent with what is already known about the "stationary" cascade network corresponding to matrices A^0, A^1 in this example since we know that such network generates quadratic polynomials.

However, Theorem 7.21 should not be construed as saying C^0 for a single f implies C^1. For example, the vector function in Example 7.14 is clearly C^0 function, but is not C^1 on $[0, 1]$.

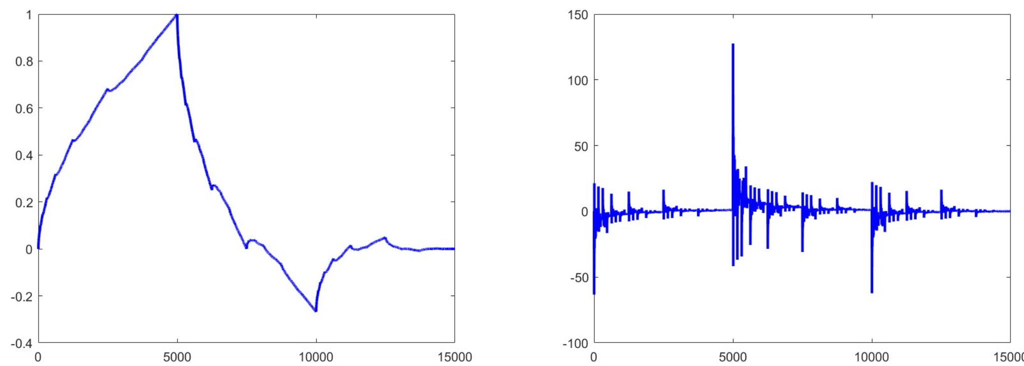
Example 7.23. Let f be the scaling function with mask coefficients $a_0 = \frac{1+\sqrt{3}}{4}, a_1 = \frac{3+\sqrt{3}}{4}, a_2 = \frac{3-\sqrt{3}}{4}, a_3 = \frac{1-\sqrt{3}}{4}$. f can be generated by a "stationary" cascade network

$$\mathcal{Y}_L = A(A \circ \alpha) \dots (A \circ \alpha^{L-1})(\mathcal{Y}_0 \circ \alpha^L)$$

where

$$A = \begin{cases} A^0, & x \in \left[0, \frac{1}{2}\right) \\ A^1, & x \in \left[\frac{1}{2}, 1\right] \end{cases} = \begin{cases} \begin{pmatrix} a_2 & a_3 & 0 \\ a_0 & a_1 & a_2 \\ 0 & 0 & a_0 \end{pmatrix}, & x \in \left[0, \dfrac{1}{2}\right) \\ \begin{pmatrix} a_3 & 0 & 0 \\ a_1 & a_2 & a_3 \\ 0 & a_0 & a_1 \end{pmatrix}, & x \in \left[\dfrac{1}{2}, 1\right] \end{cases}.$$

f satisfies conditions of Theorem 7.21 with matrices $C_k = (A^0)^{k-1}A^1(A^0)^{-k+1}$. For $f_0 = \left(\frac{a_3}{a_0}, 1, 0\right)^T$, the matrices C_k satisfy $C_{k+1}^2 f_0 = C_k f_0$ (Figure 7.1a). However, choosing a different f_0, say $f_0 = (1,1,0)^T$ yields f that is not continuous, and the matrices C_k do not satisfy $C_{k+1}^2 = C_k$ (Figure 7.1b).



(a) $f_0 = (\frac{a_3}{a_0}, 1, 0)^T$ (b) $f_0 = (1,1,0)^T$

Figure 7.1: Scaling Function

The above results are concerned with conditions for general CN's, guaranteeing convergence to C^0 functions. Motivated by Example 7.22 and in light of the results of Daubechies and Lagarias [47], discussed in Chapter 6, which are concerned with the "stationary" case, it would be interesting to see the implications of Theorem 7.21 in the special case where the CN is stationary. Recall a CN is referred to as stationary if $A_\ell = A$, $\ell \in \mathbb{N}$, where

$$A(x) = \begin{cases} A^0, & \text{if } x \in [0, 1/2) \\ A^1, & \text{if } x \in [1/2, 1], \end{cases}$$

and where A^0, A^1 are fixed matrices in $\mathbb{R}^{W \times W}$. Thus, a stationary (matrix-valued) CN is given by

$$\mathcal{Y}_L = A(A \circ \alpha) \cdots (A \circ \alpha^{L-1})(\mathcal{Y}_0 \circ \alpha^L), \quad L \in \mathbb{N},$$

where $\mathcal{Y}_0$ is a $W \times W$ matrix function, whose components are linear functions. For simplicity, we will assume below that $\mathcal{Y}_0$ is a constant matrix, so that $\mathcal{Y}_0 \circ \alpha^L = \mathcal{Y}_0$. It should be noted that although for a stationary CN's, matrix functions A_ℓ do not depend on ℓ, matrix $\mathcal{Y}_0$ that is used to generate $\mathcal{Y}_L$ may in fact depend on L. In the considerations below, it will be convenient to assume that A^0 is invertible and to set $\mathcal{Y}_0 = \mathcal{Y}_0^L = (A^0)^{-L}F_0 = ((A^0)^L)^{-1}F_0$, where $F_0 \in \mathbb{R}^{W\times W}$ is a given matrix.

Lemma 7.24. *Suppose that a stationary CN corresponding to matrices A^0, A^1, where A^0 is invertible, converges to a matrix-valued function $F \in L_\infty^{W\times W}([0,1])$. Then F satisfies (7.9), with matrices C_k given by*

$$C_k = (A^0)^{k-1}A^1(A^0)^{-k}, \quad k \in \mathbb{N}. \tag{7.13}$$

Moreover, if $\mathcal{Y}_0^L = (A^0)^{-L}F_0$, $F_0 \in \mathbb{R}^{W\times W}$, then $\lim_{h\to 0^+} \|F(\cdot) - F_0\|_{L_\infty^{W\times W}[0,h]} = 0$, i.e. F is essentially continuous at 0, with essential value F_0.

Proof. With the above assumption that $\mathcal{Y}_0 = \mathcal{Y}_0^L$ is a constant matrix, it follows that $\mathcal{Y}_L$ is the piecewise constant matrix-valued function given by

$$\mathcal{Y}_L(x) = A^{\delta_1}\cdots A^{\delta_L}\mathcal{Y}_0^L, \quad x \in \left[\frac{j}{2^L}, \frac{(j+1)}{2^L}\right), \quad j \in \{0,\ldots,2^L-1\}, \tag{7.14}$$

where $\delta_1,\ldots,\delta_L \in \{0,1\}$ are the binary digits of j, i.e., $(j)_2 = \delta_1\cdots\delta_L$. That is, values of $\mathcal{Y}_L$ are obtained as products of A^0 and A^1, multiplied by $\mathcal{Y}_0^L$. The statement of the lemma is proved for $k=1$. The proof for $k>1$ is similar.

We first show that $y_L(x+1/2) = C_1 y_L(x)$, $x \in [0,1/2)$, where, consistently with (7.13), $C_1 = A^1(A^0)^{-1}$. Thus, let $x \in \left[\frac{j}{2^L}, \frac{(j+1)}{2^L}\right)$, where $j \in \{0,\ldots,2^{L-1}-1\}$, which guarantees that $x \in [0,1/2)$. Clearly, $(j)_2 =: \delta_1\cdots\delta_L = 0\delta_2\cdots\delta_L$, i.e. $\delta_1 = 0$, on account of $j \in \{0,\ldots,2^{L-1}-1\}$. Similarly, $x+1/2 \in \left[\frac{j'}{2^L}, \frac{(j'+1)}{2^L}\right) = \left[\frac{(j+2^{L-1})}{2^L}, \frac{(j+2^{L-1}+1)}{2^L}\right)$, hence $(j')_2 = (j+2^{L-1})_2 = 1\delta_2\cdots\delta_L$. Therefore,

$$\mathcal{Y}_L(x+1/2) = A^1A^{\delta_2}\cdots A^{\delta_L}\mathcal{Y}_0^L \quad \text{and} \quad \mathcal{Y}_L(x) = A^0A^{\delta_2}\cdots A^{\delta_L}\mathcal{Y}_0^L,$$

or

$$\mathcal{Y}_L(x+1/2) = A^1A^{\delta_2}\cdots A^{\delta_L}\mathcal{Y}_0^L = A^1(A^0)^{-1}A^0A^{\delta_2}\cdots A^{\delta_L}\mathcal{Y}_0^L = C_1\mathcal{Y}_L(x).$$

Thus, since this is true for all $j \in \{0,\ldots,2^{L-1}-1\}$, it follows that $\mathcal{Y}_L(x+1/2) = C_1\mathcal{Y}_L(x)$, for all $x \in [0, 1/2)$. By assumption, $\mathcal{Y}_L \to F \in L_\infty^{W\times W}([0,1])$, hence, we also have $F(x+1/2) = C_1 F(x), x \in [0, 1/2)$.

As for the essential continuity of F at 0, note first that $\mathcal{Y}_L(0) = (A^0)^L \mathcal{Y}_0^L = F_0, L \in \mathbb{N}$. Therefore, the convergence $\mathcal{Y}_L \to F$ implies

$$\begin{aligned}
\|F - F_0\|_{L_\infty^{W\times W}([0,2^{-L}])} &\le \|F - \mathcal{Y}_L\|_{L_\infty^{W\times W}([0,2^{-L}])} + \|\mathcal{Y}_L - F_0\|_{L_\infty^{W\times W}([0,2^{-L}])} \\
&= \|F - \mathcal{Y}_L\|_{L_\infty^{W\times W}([0,2^{-L}])} + \|\mathcal{Y}_L(0) - F_0\|_\infty \\
&= \|F - \mathcal{Y}_L\|_{L_\infty^{W\times W}([0,2^{-L}])} \\
&\le \|F - \mathcal{Y}_L\|_{L_\infty^{W\times W}([0,1])} \to 0, \quad \text{as } L \to \infty.
\end{aligned}$$

$\square$

Thus, the hypothesis of a stationary CN results in constraints (7.13). On the other hand, as we have seen in Theorem 7.21, the requirement of continuity of F implies yet another set of constraints, namely

$$C_k = C_{k+1}^2, \quad k \in \mathbb{N}. \tag{7.15}$$

Combining the two families of constraints, (7.13) and (7.15), leads to the following necessary and sufficient condition for a stationary CN to converge to a continuous matrix-valued function F.

Theorem 7.25. *Suppose that a stationary CN is associated with matrices $A^0, A^1 \in \mathbb{R}^{W\times W}$, where A^0 is nonsingular, and with initial matrix $\mathcal{Y}_0^L = (A^0)^{-L} F_0$, where $F_0 \in \mathbb{R}^{W\times W}$ is nonsingular. Then the CN converges to a continuous matrix function F if an only if A^0, A^1 are such that*

$$A^1(A^0)^{-1}A^1(A^0)^{-1} = (A^0)^{-1}A^1 \tag{7.16}$$

and such that all eigenvalues of $A^1(A^0)^{-1}$ are 1.

Proof. Let $C_1 := A^1(A^0)^{-1}$ and, more generally, $C_k := (A^0)^{k-1}A^1(A^0)^{-k}, k \in \mathbb{N}$, which is consistent with (7.13). With this notation, (7.16) is equivalent to

$$C_1^2 = (A^0)^{-1} C_1 A^0. \tag{7.17}$$

To prove necessity, a stationary CN implies, by the Lemma 7.24, that F satisfies

$$F(x + 1/2) = C_1 F(x), x \in [0, 1/2), \quad F(x + 1/4) = C_2 F(x), x \in [0, 1/4),$$

with $C_2 = A^0 A^1 (A^0)^{-2} = A^0 C_1 (A^0)^{-1}$. By Theorem 7.21, since F_0, the essential value of F at 0, is nonsingular, continuity of F implies $C_2^2 = C_1$ or

$$C_1 = C_2^2 = A^0 C_1 (A^0)^{-1} A^0 C_1 (A^0)^{-1} = A^0 C_1^2 (A^0)^{-1},$$

which implies (7.17).

As for the eigenvalues of $A_1 A_0^{-1}$ or C_1, note first that (7.13) along with (7.15) imply

$$C_{k+1} = A_0^k A_1 A_0^{-k-1} = A_0 A_0^{k-1} A_1 A_0^{-k} A_0^{-1} = A_0 C_k A_0^{-1}, \quad k \in \mathbb{N}.$$

Thus, for all k, matrices C_k and C_{k+1} are similar and have the same eigenvalues. Let $\Lambda := \{\lambda_1, \dots, \lambda_W\}$ be the (multi)set of the eigenvalues of C_1, and therefore by the above, Λ are also eigenvalues of $C_k, k > 1$. By assumption, F is continuous, F_0 invertible, and thus by Theorem 7.21, we also know that $C_k = C_{k-1}^{1/2} = \dots = C_K^{2^{K-k}}$, $k > K$, for some $K \in \mathbb{N}$. This means that the eigenvalues of C_k are equal to $\Lambda_k := \{\lambda_1^{2^{K-k}}, \dots, \lambda_W^{2^{K-k}}\}, k > K$. However, at the same time, $\Lambda_k = \Lambda$ because of the similarity of C_k and C_1. Consequently, $\Lambda = \lim_{k \to \infty} \Lambda_k = \{1, \dots, 1\}$ because $C_k \to \mathbb{I}_W$ by Proposition 7.12, i.e. Λ is the multiset of size W consisting of eigenvalues 1.

Conversely, suppose that A^0, C_1 are two matrices satisfying $C_1^2 = (A^0)^{-1} C_1 A^0$, and such that all eigenvalues of C_1 are equal to 1. We will show that the stationary CN corresponding to matrices A^0 and $A^1 := C_1 A^0$ converges to a continuous matrix function $F \in (C([0, 1])^{W \times W}$. To this end, define $C_k := (A^0)^{k-1}A^1(A^0)^{-k} = (A^0)^{k-1}C_1(A^0)^{-k+1}, k \geq 2$. These matrices satisfy identity $C_k = C_{k+1}^2, k \in \mathbb{N}$, because this is equivalent to

$$(A^0)^{k-1}C_1(A^0)^{-k+1} = ((A^0)^k C_1(A^0)^{-k})^2 = (A^0)^k C_1^2(A^0)^{-k},$$

which in turn is equivalent to assumption (7.17).

Given the identities $C_k = C_{k-1}^{1/2} = \cdots = C_K^{2^{K-k}}, k \geq K$, it is not difficult to show that products of C_k are bounded in the sense that

$$\gamma := \sup_{m\in\mathbb{N}} \sup_{\delta_1,\ldots,\delta_m\in\{0,1\}} \left\| C_1^{\delta_1}\cdots C_m^{\delta_m} \right\|_\infty < \infty, \tag{7.18}$$

and also that

$$\sup_{m\geq k} \sup_{\delta_k,\ldots,\delta_m\in\{0,1\}} \left\| C_k^{\delta_k}\cdots C_m^{\delta_m} - \mathbb{I}_W \right\|_\infty \to 0 \quad \text{as } \ k \to \infty. \tag{7.19}$$

In fact, the boundedness (7.18) readily follows from (7.19). To prove (7.19), it will be enough to consider $C_k^{\delta_k}\cdots C_m^{\delta_m}$, where $m \geq k \geq K$. It follows that

$$C_k^{\delta_k}\cdots C_m^{\delta_m} = C_K^{\delta_k 2^{K-k}+\cdots+\delta_m 2^{K-m}} \to \mathbb{I}_W$$

since $\delta_k 2^{K-k} + \cdots + \delta_m 2^{K-m} \leq 2^{K-k+1} \to 0$ as $k \to \infty$, and since the eigenvalues of C_K are all equal to 1, and hence in $\mathbb{H}_+$.

It now follows from the above that there exists a matrix-valued function $F \in (C([0,1]))^{W\times W}$, satisfying conditions (7.9), and such that its essential value if F_0. The proof proceeds along similar lines as the proof of Theorem 7.16, where one defines a Cauchy sequence of continuous piecewise linear functions $\left\{F^k\right\}_{k\in\mathbb{N}}$ on $[0,1]$, with $F^k(0) = F_0$. More precisely, functions F^k are piecewise linear interpolants such that $F^k\left(j2^{-k}\right) = C_1^{\delta_1}\cdots C_k^{\delta_k}F_0$, where $j \in \{0,\ldots,2^k-1\}$ and $(j)_2 = \delta_1\cdots\delta_k$, and where $F^k(1) = C_k^{2^k} = C_1^2$. In this case, the proof that the sequence is Cauchy relies on the above-mentioned properties (7.18) and (7.19) of products of matrices C_k.

We now show that the stationary CN corresponding to matrices A^0, A^1 generates the above continuous matrix function F. Note first that convergence of the CN for matrices A^0, A^1 means that the piecewise constant function $\mathcal{Y}_L$, defined in (7.14), is convergent. We show that $\mathcal{Y}_L \to F$ as $L \to \infty$. Observe the identity

$$\mathcal{Y}_L(x) = A^{\delta_1} \cdots A^{\delta_L} \mathcal{Y}_0^L = C_1^{\delta_1} \cdots C_1^{\delta_L} F_0, \quad x \in \left[\frac{j}{2^L}, \frac{(j+1)}{2^L}\right), \quad j \in \{0, \ldots, 2^L - 1\}, L \in \mathbb{N},$$

where, as before, $(j)_2 =: \delta_1 \cdots \delta_L$. However, at the same time, by the definition of $\left\{F^k\right\}$, one has

$$F^L\left(j2^{-L}\right) = C_1^{\delta_1} \cdots C_1^{\delta_L} F_0 = \mathcal{Y}_L(j2^{-L}), \quad x \in \left[\frac{j}{2^L}, \frac{(j+1)}{2^L}\right).$$

In fact, it is not difficult to show that $F^{L+m}\left(j2^{-L}\right) = F^L\left(j2^{-L}\right), m \in \mathbb{N}$, hence

$$F\left(j2^{-L}\right) = \lim_{m\to\infty} F^{L+m}\left(j2^{-L}\right) = \lim_{m\to\infty} F^L\left(j2^{-L}\right) = F^L\left(j2^{-L}\right) = \mathcal{Y}_L\left(j2^{-L}\right),$$

hence F and $\mathcal{Y}_L$ agree on the mesh $\Omega_L \backslash \{1\}$. Since F is continuous on $[0,1]$, this implies that $\mathcal{Y}_L$ converges to F uniformly. This completes the proof. $\square$

The above result can be compared with the results of Daubechies and Lagarias [47], Theorems 6.15 and 6.16 with $\Sigma = \{A^0, A^1\}$, presented in Chapter 6. In particular, their result Theorem 6.16 is less explicit than Theorem 7.25.

7.3 Least Squares Objective Function

The following result shows that the least squares objective function has a unique global minimum.

Theorem 7.26. *Let $f \in L_2^W([0,1])$ be such that f satisfies*

$$f(\cdot + 2^{-k})|_{[0,2^{-k})} = C_k f(\cdot)|_{[0,2^{-k})},$$

for all $k \in \mathbb{N}$. Then, for all $L \in \mathbb{N}$, there exists a unique $\mathcal{Y}_L \in \mathcal{S}_{W,L}$ such that

$$\|f - \mathcal{Y}_L\|_{L_2^W([0,1])} = \inf_{g \in \mathcal{S}_{W,L}} \|f - g\|_{L_2^W([0,1])}.$$

Moreover, $\mathcal{Y}_L$ satisfies

$$\mathcal{Y}_L(\cdot + 2^{-k})|_{[0,2^{-k})} = C_k \mathcal{Y}_L(\cdot)|_{[0,2^{-k})}, \quad k = 1, \ldots, L.$$

Proof. Fix $L \in \mathbb{N}$ and construct $\mathcal{Y}_L$ as follows. Define$\mathcal{Y}_L|_{[0,2^{-L}]} \in \Pi_1^W([0, 2^{-L}])$ to be the the best L_2 approximation of f on $[0, 2^{-L})$. Next, define $\mathcal{Y}_L$ on $[0, 2^{-L+1})$ by

$$\mathcal{Y}_L(\cdot + 2^{-L})|_{[0,2^{-L+1})} = C_L \mathcal{Y}_L(\cdot)|_{[0,2^{-L+1})}.$$

Since f satisfies the same equation, it is not difficult to show that $\mathcal{Y}_L$ is the best L_2 approximation of f on $[0, 2^{-L+1})$. Now proceed inductively to find $\mathcal{Y}_L$ on $[0, 1]$ using

$$\mathcal{Y}_L(\cdot + 2^{-k})|_{[0,2^{-k})} = C_k \mathcal{Y}_L(\cdot)|_{[0,2^{-k})}, \quad k = L-1, \ldots, 1.$$

Such a function $\mathcal{Y}_L$ will be the best L_2 approximation of f on $[0, 1]$.

□

Corollary 7.7. *Let $\mathcal{Y}_{L+1} \in \mathcal{S}_{W,L+1}$ such that*

$$\mathcal{Y}_{L+1}(\cdot + 2^{-\ell})|_{[0,2^{-\ell})} = C_{L+1}^{\ell} \mathcal{Y}_L(\cdot)|_{[0,2^{-\ell})}, \;\; \ell = 1, \ldots, L+1.$$

Let $\mathcal{Y}_L \in \mathcal{S}_{W,L}$ be the best L_2 approximation of $\mathcal{Y}_{L+1}$ from $\mathcal{S}_{W,L}$. Then $\mathcal{Y}_L$ satisfies

$$\mathcal{Y}_L(\cdot + 2^{-\ell})|_{[0,2^{-\ell})} = C_L^{\ell} \mathcal{Y}_L(\cdot)|_{[0,2^{-\ell})}, \;\; \ell = 1, \ldots, L,$$

where $C_L^{\ell} = C_{L+1}^{\ell}$ for $\ell = 1, \ldots, L$.

Proof. Set $f = \mathcal{Y}_{L+1}$ in Theorem 7.26. $\square$

Conjecture 7.8. *Theorem 7.26 is true for all $f \in L_2^W([0,1])$ without the "periodicity" conditions (7.9).*

Chapter 8

The space $\mathcal{S}_W$

8.1 Definition and Properties

Definition 8.1. Define $\mathcal{S}_W = \overline{\bigcup_{L=1}^{\infty} \mathcal{S}_{W,L}}^{L_\infty}$.

This is equivalent to $f \in \mathcal{S}_W$ if $\mathcal{Y}_L \to f$ in L_∞ for $\mathcal{Y}_L \in \mathcal{S}_{W,L}$.

Theorem 8.2. *$f \in \mathcal{S}_W$ and satisfies (7.10) if and only if f satisfies (7.9).*

Note that this is a restatement of Theorem 7.11.

Conjecture 8.1. *For $\mathcal{Y}, \tilde{\mathcal{Y}} \in \mathcal{S}_W$, $\mathcal{Y} + \tilde{\mathcal{Y}} \in \mathcal{S}_{W^2}$*

Conjecture 8.2. *$\mathcal{S}_W$ is the set of all functions f such that for all fixed $\ell = 1, 2, \ldots,$*

$$\operatorname{span}\{f|_{[0,2^{-\ell}]}\} = \operatorname{span}\{f(\cdot + j2^{-\ell})|_{[0,2^{-\ell}]}\}$$

for all $j = 1, \ldots, 2^{-\ell} - 1$.

As seen in Chapter 4, $f \in \mathcal{S}_W$ can be generated by a non-stationary subdivision scheme.

In the following, exponential functions are shown to be generated by cascade networks. More generally, solutions of linear systems of differential equations belong to $\mathcal{S}_W$.

Proposition 8.3. Consider the linear system of differential equations, $F' = DF$, for $F \in C([0,1])^{W\times W}$ and $D \in \mathbb{R}^{W\times W}$. Then,

a) The solution of the system $F' = DF$ can be generated by a CN,

$$\mathcal{Y}_L(x) = B_0(x)B_1(\alpha(x))\cdots B_L(\alpha^{L-1}(x))\mathcal{Y}_0, \tag{8.1}$$

in the sense that $\|F - \mathcal{Y}_L\|_\infty \to 0$. Here, $\mathcal{Y}_0 = \mathbb{I}_W$, $B_0 = e^{\frac{D}{2}}$, and

$$B_\ell(x) = \begin{cases} e^{\frac{-D}{2^{\ell+1}}}, & x \in \left[0, \frac{1}{2}\right) \\ e^{\frac{D}{2^{\ell+1}}}, & x \in \left[\frac{1}{2}, 1\right], \end{cases}$$

for $\ell = 1, \ldots, L$.

b) If $f(x) = e^{Dx}u$, for $x \in [0,1]$, $u \in \mathbb{R}^W$, then f can be generated by setting $\mathcal{Y}_0 = u$ in (8.1).

Proof. Only the proof of part (a) is presented, the proof of part (b) is straightforward.

(a) As is well known, a general solution of the differential system $F' = DF$ is $F(x) = e^{Dx}F(0)$ for initial condition $F(0)$.

For $x = \frac{m}{2^{L+1}}, m \in \{1, \ldots, 2^{L+1} - 1\}$ odd, we will show that

$$\mathcal{Y}_L(x) = B_0(x)B_1(\alpha(x)) \cdots B_L(\alpha^{L-1}(x))\mathcal{Y}_0 = e^{Dx}$$

is a piecewise constant function, constant on the intervals $\left[\frac{m-1}{2^{L+1}}, \frac{m+1}{2^{L+1}}\right)$ interpolating e^{Dx} at the midpoints of those intervals. The proof will proceed by induction.

For $L = 1, m = \{1, 3\}$, and

$$\begin{aligned} \mathcal{Y}_1(x) &= B_0 B_1(x) \\ &= \begin{cases} e^{\frac{D}{2}} e^{\frac{-D}{4}}, & x \in \left[0, \frac{1}{2}\right) \\ e^{\frac{D}{2}} e^{\frac{3D}{4}}, & x \in \left[\frac{1}{2}, 1\right] \end{cases} \\ &= \begin{cases} e^{\frac{D}{4}}, & x \in \left[0, \frac{1}{2}\right) \\ e^{\frac{3D}{4}}, & x \in \left[\frac{1}{2}, 1\right] \end{cases}. \end{aligned}$$

Suppose the induction hypothesis is true for L, or

$$\mathcal{Y}_L(x) = B_0(x)B_1(\alpha(x)) \cdots B_L(\alpha^{L-1}(x))\mathcal{Y}_0 = e^{\frac{Dm}{2^{L+1}}},$$

for $x \in \left[\frac{m-1}{2^{L+1}}, \frac{m+1}{2^{L+1}}\right)$ and $m \in \{1, \ldots, 2^{L+1}-1\}$ odd. Then,

$$\begin{aligned}
\mathcal{Y}_{L+1}(x) &= B_0(x)B_1(\alpha(x)) \cdots B_L(\alpha^{L-1}(x))B_{L+1}(\alpha^L(x))\mathcal{Y}_0 \\
&= \mathcal{Y}_L(x)B_{L+1}\left(\alpha^L(x)\right) \\
&= \begin{cases} e^{\frac{Dm}{2^{L+1}}}e^{\frac{-D}{2^{L+2}}}, & x \in \left[\frac{2m-1-1}{2^{L+2}}, \frac{2m-1+1}{2^{L+2}}\right) \\ e^{\frac{Dm}{2^{L+1}}}e^{\frac{D}{2^{L+2}}}, & x \in \left[\frac{2m-1+1}{2^{L+2}}, \frac{2m-1+3}{2^{L+2}}\right] \end{cases} \\
&= \begin{cases} e^{\frac{Dk}{2^{L+2}}}, & x \in \left[\frac{k-1}{2^{L+2}}, \frac{k+1}{2^{L+2}}\right), \ k = 2m-1 \\ e^{\frac{Dk}{2^{L+2}}}, & x \in \left[\frac{k-1}{2^{L+2}}, \frac{k+1}{2^{L+2}}\right], \ k = 2m+1 \end{cases}
\end{aligned}$$

is piecewise constant, and interpolates e^{Dx} at the midpoints of those intervals.

Note that in the proof the identity, $e^A e^B = e^{A+B}$, was repeatedly used. In general, this identity is not true. However, it is true if A, B are scalar multiples of the same matrix D. $\square$

Instead of piecewise constant approximation used in Proposition 8.3, one could also use piecewise linear approximation, by using an appropriate linear matrix function $\mathcal{Y}_0$ instead of $\mathbb{I}_W$. The end result would be a linear spline $\mathcal{Y}_L$ interpolating e^{Dx} at Ω_L.

By Proposition 8.3, solutions of differential equations are elements of $\mathcal{S}_W$, and in some sense, these are the only smooth vector functions belonging to $\mathcal{S}_W$.

Proposition 8.4. Let $F : [0,1] \to \mathbb{R}^{W\times W}$ be a continuous matrix function satisfying $F(x+2^k) = C_k F(x), x \in [0, 2^{-k}], k \in \mathbb{N}$, where $F(0)$ is invertible. Then, $F' = FD$ for some $D \in \mathbb{R}^{W\times W}$.

Proof. This follows from the Theorem 7.21.

$\square$

Remark 8.3. *The condition $F(0)$ is invertible is not prohibiting, in the sense that Proposition 8.4 is still true if $F(0)$ not invertible.*

Remark 8.4. *Proposition 8.4 implies that $\mathcal{S}_W$ is closed under differentiation.*

Example 8.5. By Example 3.2 and Example 7.22,

$$C_k = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2^k} & 1 & 0 \\ \frac{1}{4^k} & \frac{1}{2^{k-1}} & 1 \end{pmatrix}, \quad k \in \mathbb{N}.$$

Then

$$D = \lim_{k\to\infty} 2^k \left(C_k - \mathbb{I}_W\right) = \lim_{k\to\infty} \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ \frac{1}{2^k} & 2 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 2 & 0 \end{pmatrix}.$$

If $f' = Df$, for $f = (f_1, f_2, f_3)^T$. Then $f_1' = 0$, $f_2' = f_1$ and $f_3' = 2f_2$. Thus, $f_3''' = 2f_2'' = 2f_1 = 0$ and so f_3 is quadratic, f_2 is linear and f_1 is constant. Thus, a cascade network corresponding to matrices C_k can generate quadratic polynomials, as seen in Chapter 3, Example 3.2.

8.2 Approximation from Null Spaces of Linear Differential Operators

Results from the previous section suggest that solutions of systems of constant coefficient differential equations play an important role in the analysis of CN's. In this section, the question of how well such functions approximate smooth functions is considered. This question was also considered by Vatchev [60] and we will provide a comparison with our results. First, a few auxiliary results are presented.

A vector function $f : [0, 1] \to \mathbb{R}^n$ satisfies a linear, homogeneous, system of differential equations if and only if a linear combination of the components of f satisfy a homogeneous, constant coefficient differential equation of order n. Therefore, one can consider a constant coefficient differential operator of order n, instead of a system of first order differential equations.

Lemma 8.6. *Let L be a constant coefficient differential operator of order 1, $Lf = a_1 f' + a_0 f$, $a_0, a_1 \in \mathbb{R}$, $a_1 \neq 0$. Let $h \in C^1(\mathbb{R})$ be such that $h(x_0) = h(x_1) = 0$, $x_0 \leq x_1$, where* *if* $x_0 = x_1$ *the equality is understood such that* $h(x_0) = h'(x_0) = 0$. *Then,* $(Lh)(\xi) = 0$, *for some* $\xi \in [x_0, x_1]$.

Proof. If $x_0 < x_1$, let $\lambda = \frac{a_0}{a_1}$. Then, $a_1 \left(e^{\lambda x} f(x)\right)' = e^{\lambda x}(Lf)(x)$, for all $f \in C^1(\mathbb{R})$ and all x. Let $g = e^{\lambda \cdot} h$. Then, $g(x_0) = g(x_1) = 0$. Thus, by Rolle's Theorem, $g'(\xi) = 0$, for some $\xi \in (x_0, x_1)$. Therefore, $Lh(\xi) = e^{-\lambda\xi} a_1 g'(\xi) = 0$.

If $x_0 = x_1$, then $h(x_0) = h'(x_0) = 0$. Thus, for $\xi = x_0$, $Lh(\xi) = Lh(x_0) = 0$.

□

A similar result holds for second order equations.

Lemma 8.7. *Let* L *be the constant coefficient differential operator* $Lf = a_1 f'' + \lambda^2 f,\ \lambda \geq 0,\ a_1 = 1$. *Let* $h \in C^1(\mathbb{R})$ *be such that* $h(x_0) = h(x_1) = h(x_2) = 0,\ x_0 \leq x_1 \leq x_2$, *where an equality means derivatives are set to zero (e.g if* $x_0 = x_1 < x_2$, *then* $h(x_0) = h'(x_0) = 0$ *or if* $x_0 = x_1 = x_2$, *then* $h(x_0) = h'(x_0) = h''(x_0) = 0$*). Then,* $(Lh)(\xi) = 0$, *for some* $\xi \in [x_0, x_2]$.

Proof. Assume first $x_0 < x_1 < x_2$, then $h(x_0) = h(x_1) = h(x_2) = 0$.

$$\begin{aligned} Lh &= \frac{1}{\cos(\lambda x + c)} \left(\cos^2(\lambda x + c) \left(\frac{h(x)}{\cos(\lambda x + c)}\right)'\right)' \\ &= h''(x) + \lambda^2 h. \end{aligned}$$

As $h(x_0) = h(x_1) = h(x_2) = 0$, $\left(\frac{h(x)}{\cos(\lambda x + c)}\right)'\Big|_{x=\xi_1} = 0$, for some $\xi_1 \in (x_0, x_1)$ and $\left(\frac{h(x)}{\cos(\lambda x + c)}\right)'\Big|_{x=\xi_2} = 0$, for some $\xi_2 \in (x_1, x_2)$. Thus, $\left(\cos^2(\lambda x + c)\left(\frac{h(x)}{\cos(\lambda x + c)}\right)'\right)'\Big|_{x=\xi} = 0$, for some $\xi \in (\xi_1, \xi_2)$.

If $x_0 = x_1 = x_2$, then $h(x_0) = h'(x_0) = h''(x_0)$. Thus, for $\xi = x_0$, $Lh(\xi) = Lh(x_0) = 0$.

If $x_0 \leq x_1 \leq x_2$ and at least one of the inequalities is an equality. Fix x_0, and consider sequences $\{x_1^n\}$, $\{x_2^n\}$ such that $x_1^n \to x_1,\ x_2^n \to x_2,\ x_0, x_1^n < x_2^n$. Define a sequence $\{h_n\}$ by $h_n(x) := h(x) + a_n(x - x_0) + b_n(x - x_0)^2$, where $b_n = \frac{1}{\ell_1^n - \ell_2^n}\left[\frac{1}{\ell_2^n} h(x_2^n) - \frac{1}{\ell_1^n} h(x_1^n)\right]$ and $a_n = \frac{1}{\ell_1^n - \ell_2^n}\left[\frac{\ell_2^n}{\ell_-^n} h(x_1^n) - \frac{\ell_1^n}{\ell_2^n} h(x_2^n)\right]$, for $\ell_1^n = x_1^n - x_0$ and $\ell_2^n = x_2^n - x_0$. Then, $\{h_n\}$ satisfies $h_n(x_1^n) = h_n(x_2^n) = h_n(x_0) = 0$ and $\{h_n\}$ is such that $h_n \to h$ in $C^2\ [x_0 - \varepsilon, x_0 + \varepsilon]$, where ε independent of n. Thus, $Lh_n \to Lh$ on $[x_0 - \varepsilon, x_0 + \varepsilon]$. □

The above two lemmas can be generalized to n-th order equations.

Lemma 8.8. *Let L be a constant coefficient differential operator of order n, $Lf = a_n f^{(n)} + \cdots + a_0 f,\ a_0, \ldots, a_n \in \mathbb{R},\ a_n \neq 0$. Let $x_0, \ldots, x_n$, not necessarily distinct, and let $h \in C^n(\mathbb{R})$ be such that $h(x_0) = \cdots = h(x_n) = 0$, with the corresponding modification if some of the nodes coalesce. Then, $Lh(\xi) = 0$, for some $\xi \in$ conv$\{x_0, \ldots, x_n\}$, the convex hull of the points $x_0, \ldots, x_n$.*

Proof. L can be written as $L = L_1 \cdots L_k$, where $L_i f = f'' + \lambda_i^2 f$ or $L_i f = f + \lambda_i$ for $i = 1, \ldots, k$, and the order of L adds up to n. Without loss of generality, assume $x_0 < \cdots < x_n$, then conv$\{x_0, \ldots, x_n\} = [x_0, x_n]$. By Lemma 8.6 and 8.7, $L_k h(x_1') = \cdots = L_k h(x_n') = 0$, for some $x_1' \in (x_0, x_1), \ldots, x_n' \in (x_{n-1}, x_n)$. The proof now proceeds by induction. The case of coalescent nodes can be handled similarly. $\square$

The following result gives an error formula for approximating smooth functions by elements of null spaces of the differential operators considered above.

Theorem 8.9. *Let L be a constant coefficient differential operator of order n, $Lf = a_n f^{(n)} + \cdots + a_0 f,\ a_0, \ldots, a_n \in \mathbb{R},\ a_n \neq 0$. Let $f \in C^n(\mathbb{R})$, $x_0, \ldots, x_{n-1} \in \mathbb{R}$, not necessarily distinct, $N(L)$ be the null space of L and $I_n f \in N(L)$ a function such that $I_n f(x_i) = f(x_i)$, for $i = 0, \ldots, n-1$. Also, let g be a function such that $Lg = 1$ and $g(x_0) = \cdots = g(x_{n-1}) = 0$. Then for all $x \in \mathbb{R}$, there exists a $\xi = \xi(x) \in$ conv$\{x, x_0, \ldots, x_{n-1}\}$ such that*

$$f(x) - I_n f(x) = g(x) Lf(\xi). \tag{8.2}$$

Proof. Suppose first that the nodes are distinct and without loss of generality, assume $x_0 < \cdots < x_{n-1}$, and that $x \notin \{x_0, \ldots, x_{n-1}\}$ (if $x \in \{x_0, \ldots, x_{n-1}\}$, then equation 8.2 holds trivially for all ξ). Define

$$h(t) := (f(t) - I_n f(t))\, g(x) - (f(x) - I_n f(x))\, g(t).$$

Then, $h(x_i) = 0$ for $i = 0, \ldots, n-1$, because $f(x_i) - I_n f(x_i) = 0$ and $g(x_i) = 0$ for $i = 0, \ldots, n-1$, by definition. Also, $h(x) = 0$.

Thus, by Lemma 8.8, there exists a $\xi \in \text{conv}\{x, x_0, \ldots, x_{n-1}\}$ such that $Lh(\xi) = 0$. Therefore,

$$\begin{aligned} 0 &= L\left(f(t) - I_n f(t)\right)|_{t=\xi} g(x) - \left(f(x) - I_n f(x)\right) Lg(\xi) \\ &= \left(Lf(\xi) - L(I_n f)(\xi)\right) g(x) - \left(f(x) - I_n f(x)\right) \\ &= Lf(\xi)g(x) - \left(f(x) - I_n f(x)\right), \end{aligned}$$

which implies (8.2). □

The case where some of the nodes coalesce can be dealt with analogously. For example, if $x_0 = x$, the interpretation of $I_n f$ and g is such that $I_n f(x_0) = f(x_0), (I_n f(x_0))' = (f(x_0))'$, and $g(x_0) = g'(x_0) = 0$. The definition of h remains the same. However, the equations $h(x_0) = h(x_1) = 0$ are replaced with $h(x_0) = h'(x_0) = 0$.

Concerning functions $I_n f$ and g in the above theorem, note that such functions are not always guaranteed to exist. For example, if $Lf = f'' + f$ or $N(L) = \text{span}\{\sin, \cos\}$, and $x_0 = 0, x_1 = \pi$, there is no g such that $Lg = 1$ and $g(0) = g(\pi) = 0$. Similarly, there is no function $I_2 f \in N(L)$ such that $I_2 f(0) = f(0), I_2 f(\pi) = f(\pi)$ unless $f(0) + f(\pi) = 0$. On the other hand, it is a well-known property of $N(L)$ that $I_n f, g$ are guaranteed to exist, and uniquely so, provided that the interval $[x_0, x_{n-1}]$ is sufficiently small. For the above example, the unique existence of $I_n f, g$ is guaranteed whenever $x_1 - x_0 < \pi$.

In the following, consider the special case of Theorem 8.9 where $x_0 = x_1 = \cdots = x_{n-1}$, which will be needed later.

Corollary 8.5. *Let L be a constant coefficient differential operator of order n, $Lf = a_n f^{(n)} + \cdots + a_0 f,\ a_0, \ldots, a_n \in \mathbb{R},\ a_n \neq 0$. Let $f \in C^n(\mathbb{R})$, $N(L)$ be the null space of L, and $I_n f$ be the unique function in $N(L)$ such that $I_n f^{(i)}(x_0) = f^{(i)}(x_0)$, for $i = 1, \ldots, n-1$. Also, let g be the unique function such that $Lg = 1$ and $g(x_0) = g^{(1)}(x_0) \cdots = g^{(n-1)}(x_0) = 0$. Then for all $x \in \mathbb{R}$, there exists a $\xi = \xi(x) \in conv\{x_0, x\}$ such that*

$$f(x) - I_n f(x) = g(x) Lf(\xi). \tag{8.3}$$

Example 8.10. Consider the operator $Lf = f^{(n)}$. Let $x_i = \cos\left(\frac{2i+1}{2n}\pi\right)$, for $i = 0, \ldots, n-1$, be the Chebyshev nodes for $[-1, 1]$. By Theorem 8.9, $f(x) - I_n f(x) = g(x) f^{(n)}(\xi)$, for some $\xi \in -1, 1$. Here, g is the renormalized Chebyshev polynomial, $g(x) = \frac{1}{2^{n-1}n!} T_n(x)$, where $T_n(x) = \cos(n \cos^{-1} x)$ is the standard Chebyshev polynomial. In this case, $g^{(n)}(x) = 1, g(x_i) = 0$ and $T_n(x_i) = 0$, for $i = 0, \ldots, n-1$. Moreover, by (8.2),

$$\begin{aligned}\|f - I_n f\|_{L_\infty([-1,1])} &\leq \|g\|_{L_\infty([-1,1])} \|f^{(n)}\|_{L_\infty([-1,1])} \\ &= \frac{1}{2^{n-1}n!} \|f^{(n)}\|_{L_\infty([-1,1])}.\end{aligned}$$

Thus, the classical estimate for polynomial interpolation and Chebyshev nodes is recovered.

Let f be a function defined on a finite interval A. Consider the problem of approximating f from the null space of a differential operator of the form $L_n f = \sum_{k=0}^{n} a_k f^{(k)},\ a_k \in \mathbb{R},\ a_n \neq 0$. It is known that $N(L_n) = \operatorname{span}\left\{e^{\lambda_k x} | k = 1, \ldots, n\right\}$, where λ_k are the roots of the corresponding characteristic polynomial $P_n(\lambda) = \sum_{k=0}^{n} a_k \lambda^k$ and where λ_k are assumed to be distinct.

The following result of Vatchev [60] establishes Jackson type estimates for the error of approximation from $N(L_n)$.

Theorem 8.11. *[60] Let $L_n f = a_n f^{(n)} + \cdots + a_0 f, a_k \in \mathbb{R}$, for $k = 0, \ldots, n$ and $a_n \neq 0$ with $N(L_n) = \operatorname{span}\left\{e^{\lambda_k x} | k = 1, \ldots, n\right\}$, λ_i complex, distinct. Then, for each $f \in C^n(A)$, A a finite interval, there is a selection of coefficients $b_1, \ldots, b_n$ and corresponding linear combination $S_n f(x) = \sum_{k=1}^{n} b_k e^{\lambda_k x}$ of functions in $N(L_n)$ which satisfy*

$$\|f^{(m)} - (S_n f(x))^{(m)}\|_{L_\infty(A)} \leq \frac{|A|^{1/q} e^{|\lambda_n||A|}}{|a_n| 2^{n-m-1/p} |\lambda_n|^{n-m-1}} \|L_n f\|_{L_p(A)}, \tag{8.4}$$

where $|\lambda_n| = \max_k |\lambda_k|,\ 0 \leq m \leq n-1,\ p, q \geq 1$, and $\frac{1}{p} + \frac{1}{q} = 1$.

Theorem 8.12. *Let L_n be a constant coefficient differential operator of order n, $L_n f = a_n f^{(n)} + \cdots + a_0 f$, where $a_n = 1$, $a_0 \neq 0$ with $N(L_n) = \operatorname{span}\left\{e^{\lambda_k x} | k = 1, \ldots, n\right\}$, where λ_i's are real,* *distinct, and ordered* $\lambda_1 \le \cdots \le \lambda_n$*. For* $S_n f(x) \in N(L_n)$*,*

$$\|f - S_n f(x)\|_{L_\infty(I)} \le \|g\|_{L_\infty(I)} \|L_n f\|_{L_\infty(I)}, \tag{8.5}$$

where

$$\|g\|_\infty \le \frac{1}{(n-1)!} \times \begin{cases} \min\left\{2^{-n} e^{\bar{\lambda}/2}, e^{-\lambda_1}, e^{\lambda_n}\right\}, & \text{if } \lambda_1 \le n, \lambda_n \ge -n \\ e^{-n}\left(n/\lambda_1\right)^n, & \text{if } \lambda_1 > n \\ e^{-n}\left(-n/\lambda_n\right)^n, & \text{if } \lambda_n < -n \end{cases},$$

and $\bar{\lambda} := \max\{|\lambda_1|, \ldots, |\lambda_n|\} = \max\{|\lambda_1|, |\lambda_n|\}$.

Proof. Let L_n be a constant coefficient differential operator of order n, $L_n f = a_n f^{(n)} + \cdots + a_0 f$, where $a_n = 1$, $a_0 \neq 0$. The characteristic polynomial is $\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_0 = (\lambda - \lambda_1)\cdots(\lambda - \lambda_n)$, hence $a_0 = (-1)^n \lambda_1 \cdots \lambda_n \neq 0$. As the λ_i's are real, distinct, and ordered $\lambda_1 \le \cdots \le \lambda_n$, the convex hull of the λ_i's is $[\lambda_1, \lambda_n]$. Set $\bar{\lambda} := \max\{|\lambda_1|, \ldots, |\lambda_n|\} = \max\{|\lambda_1|, |\lambda_n|\}$.

For $x_0 \in [0,1]$, let g be such that $Lg = 1$ and $g^{(i)}(x_0) = 0$ for $i = 0, \ldots, n-1$. Since $a_0 \neq 0$, it follows that $g(x) = \frac{1}{a_0} + \sum\limits_{j=1}^{n} d_j e^{\lambda_j x}$. For $i \ge 1$, $g^{(i)}(x_0) = \sum_{j=1}^n d_j \lambda_j^i e^{\lambda_j x_0}$, and we would like to find the coefficients d_i such that $g^{(i)}(x_0) = 0$. This can be written as a system

$$\underbrace{\begin{pmatrix} 1 & \cdots & \cdots & 1 \\ \lambda_1 & \cdots & \cdots & \lambda_n \\ \vdots & \cdots & \cdots & \vdots \\ \lambda_1^{n-1} & \cdots & \cdots & \lambda_n^{n-1} \end{pmatrix}}_{V} \underbrace{\begin{pmatrix} e^{\lambda_1 x_0} & 0 & \cdots & 0 \\ 0 & e^{\lambda_2 x_0} & \cdots & 0 \\ \vdots & \cdots & \ddots & 0 \\ 0 & \cdots & \cdots & e^{\lambda_n x_0} \end{pmatrix}}_{D} \underbrace{\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix}}_{\mathbf{d}} = \underbrace{\begin{pmatrix} \frac{-1}{a_0} \\ 0 \\ \vdots \\ 0 \end{pmatrix}}_{\mathbf{a}}. \tag{8.6}$$

Note that V is a Vandermonde matrix and is nonsingular for λ_i distinct. Thus, the system (8.6) has a unique solution $\mathbf{d} = D^{-1}V^{-1}\mathbf{a}$. Due to the special structure of $\mathbf{a}$, $V^{-1}\mathbf{a} = \frac{-1}{a_0} v$, where v is the first column of V^{-1}. Let $v = (v_1, \ldots, v_n)^T$, then by [61, 62] and noting that a_0 can be written as $a_0 = (-1)^n \prod_{j=1}^{n} \lambda_j$, it follows that

$$v_i = \frac{\prod\limits_{\substack{j=1\\j\neq i}}^{n} \lambda_j}{\prod\limits_{\substack{j=1\\j\neq i}}^{n} (\lambda_j - \lambda_i)} = (-1)^n a_0 \frac{1}{\lambda_i \prod\limits_{\substack{j=1\\j\neq i}}^{n} (\lambda_j - \lambda_i)}, \quad i = 1, \ldots, n.$$

Thus

$$V^{-1}\mathbf{a} = (-1)^{n+1} \begin{pmatrix} \frac{1}{\lambda_1 \prod\limits_{\substack{j=1\\j\neq i}}^{n} (\lambda_j - \lambda_i)} \\ \vdots \\ \frac{1}{\lambda_n \prod\limits_{\substack{j=1\\j\neq i}}^{n} (\lambda_j - \lambda_i)} \end{pmatrix} = \begin{pmatrix} \frac{1}{\lambda_1 \prod\limits_{\substack{j=1\\j\neq i}}^{n} (\lambda_i - \lambda_j)} \\ \vdots \\ \frac{1}{\lambda_n \prod\limits_{\substack{j=1\\j\neq i}}^{n} (\lambda_i - \lambda_j)} \end{pmatrix}, \quad \text{and } \mathbf{d} = \begin{pmatrix} \frac{e^{\lambda_1 x_0}}{\lambda_1 \prod\limits_{\substack{j=1\\j\neq i}}^{n} (\lambda_i - \lambda_j)} \\ \vdots \\ \frac{e^{\lambda_n x_0}}{\lambda_n \prod\limits_{\substack{j=1\\j\neq i}}^{n} (\lambda_i - \lambda_j)} \end{pmatrix}$$

Therefore,

$$g(x) = \frac{1}{\lambda_1 \cdots \lambda_n} + (-1)^n \sum_{i=1}^{n} \frac{e^{\lambda_i (x - x_0)}}{\lambda_i \prod\limits_{\substack{j=1\\j\neq i}}^{n} (\lambda_i - \lambda_j)}. \tag{8.7}$$

Thus,

$$\begin{aligned} g'(x) &= (-1)^n \sum_{i=1}^{n} \frac{e^{\lambda_i (x - x_0)}}{\prod\limits_{\substack{j=1\\j\neq i}}^{n} (\lambda_i - \lambda_j)} \\ &= (-1)^n [\lambda_1, \ldots, \lambda_n] e^{\lambda(x - x_0)} \\ &= (-1)^n \frac{\frac{d^{n-1}}{d\lambda^{n-1}} e^{\lambda(x-x_0)} \Big|_{\lambda = t}}{(n-1)!} \\ &= (-1)^n \frac{e^{t(x-x_0)} (x - x_0)^{n-1}}{(n-1)!}, \end{aligned}$$

for some $t \in [\lambda_1, \lambda_n]$. The notation $[\lambda_1, \ldots, \lambda_n] e^{\lambda(x-x_0)}$ stands for the divided difference of function $h(\lambda) = e^{\lambda(x-x_0)}$, corresponding to nodes $\lambda_1, \ldots, \lambda_n$. It is a well known fact that$[\lambda_1, \ldots, \lambda_n] h =$

$\frac{h^{n-1}(t)}{(n-1)!}$, for some $t \in [\lambda_, \ldots, \lambda_n]$. Thus, noting that

$$g(x) = g(x_0) + g'(\xi(x))(x - x_0) = g'(\xi(x))(x - x_0),$$

for some $\xi(x) \in [0, 1]$, we obtain

$$\|g\|_{L_\infty(I)} \leq \|g'(\cdot)(\cdot - x_0)\|_\infty \leq \sup_{x\in[0,1]} \Big\{ \sup_{t\in[\lambda_1,\lambda_n]} \{s_n(t, x - x_0)\} \Big\},$$

where $s_n(t, x) := e^{tx}|x|^n/(n-1)!$.

To obtain a bound for the right-hand side, note first that if $t, x \in \mathbb{R}$, then $\max\{s_n(t, x), s_n(t, -x)\} = s_n(|t|, |x|)$. Setting $x_0 = 1/2$ and observing that $\sup_{x\in[0,1]} \big\{\sup_{t\in[\lambda_1,\lambda_n]} |t(x - x_0)|\big\} = \bar{\lambda}/2$, we thus obtain

$$\begin{aligned}
\|g\|_{L_\infty(I)} &\leq \sup_{x\in[0,1]} \Big\{ \sup_{t\in[\lambda_1,\lambda_n]} \{s_n(|t|, |x - x_0|)\} \Big\} \\
&\leq s_n(\bar{\lambda}, 1/2) \\
&= \frac{e^{\bar{\lambda}/2}}{2^n(n-1)!}.
\end{aligned}$$

Next, set $x_0 = 1$. Using elementary calculus it can be shown that

$$\begin{aligned}
\sup_{x\in[0,1]} \Big\{ \sup_{t\in[\lambda_1,\lambda_n]} \{e^{t(x-x_0)}|x - x_0|^n)\} \Big\} &= \sup_{x\in[0,1]} \Big\{ \sup_{t\in[\lambda_1,\lambda_n]} \{e^{t(x-1)}(1 - x)^n)\} \Big\} \\
&= \begin{cases} e^{-\lambda_1}, & \text{if } \lambda_1 \leq n \\ e^{-n} \left(\frac{n}{\lambda_1}\right)^n, & \text{if } \lambda_1 > n, \end{cases}
\end{aligned}$$

and a similar result holds in the case $x_0 = 0$:

$$\sup_{x\in[0,1]} \Big\{ \sup_{t\in[\lambda_1,\lambda_n]} \{e^{t(x-x_0)}|x - x_0|^n)\} \Big\} = \sup_{x\in[0,1]} \Big\{ \sup_{t\in[\lambda_1,\lambda_n]} \{e^{tx}x^n)\} \Big\}$$

$$= \begin{cases} e^{\lambda_n}, & \text{if } \lambda_n \geq -n \\ e^{-n}\left(\frac{-n}{\lambda_n}\right)^n, & \text{if } \lambda_n < -n, \end{cases}$$

Combining these inequalities yields

$$\|g\|_{L_\infty(I)} \leq \frac{1}{(n-1)!} \times \begin{cases} \min\left\{2^{-n}e^{\bar{\lambda}/2}, e^{-\lambda_1}, e^{\lambda_n}\right\}, & \text{if } \lambda_1 \leq n, \lambda_n \geq -n \\ \min\left\{2^{-n}e^{\bar{\lambda}/2}, e^{-n}\left(n/\lambda_1\right)^n, e^{\lambda_n}\right\}, & \text{if } \lambda_n \geq \lambda_1 > n \\ \min\left\{2^{-n}e^{\bar{\lambda}/2}, e^{-\lambda_1}, e^{-n}\left(-n/\lambda_n\right)^n\right\}, & \text{if } \lambda_1 \leq \lambda_n < -n \end{cases}$$

$$= \frac{1}{(n-1)!} \times \begin{cases} \min\left\{2^{-n}e^{\bar{\lambda}/2}, e^{-\lambda_1}, e^{\lambda_n}\right\}, & \text{if } \lambda_1 \leq n, \lambda_n \geq -n \\ e^{-n}\left(n/\lambda_1\right)^n, & \text{if } \lambda_1 > n \\ e^{-n}\left(-n/\lambda_n\right)^n, & \text{if } \lambda_n < -n \end{cases}$$

□

Further, one can show that the condition that the λ_i's be distinct can be dispensed with by considering a sequence of distinct real exponents $(\lambda_i^{(k)})_{k=1}^{\infty}$, converging to λ_i, $i = 1, \ldots, n$, and then obtain the above bounds by passing to the limit as $k \to \infty$. The details of the straightforward proof are omitted except for pointing out that one can also apply this limit argument to the case where $a_0 = 0$, i.e. when one or more of the λ_i's is zero. For this, one needs to observe that if g_k corresponds to exponents $\lambda_1^{(k)}, \ldots, \lambda_n^{(k)}$, and hence is such that $Lg_k = 1$ and $g_k^{(i)}(x_0) = 0, i = 0, \ldots, n-1$, then the pointwise limit $g := \lim_{k\to\infty} g_k$ exists and is such that $Lg = 1$ and $g^{(i)}(x_0) = 0, j = 0, \ldots, n-1$. This follows from the easy-to-verify fact that

$$g(x) = \lim_{k\to\infty} g_k(x) = \lim_{k\to\infty} [0, \lambda_1^{(k)}, \ldots, \lambda_n^{(k)}]e^{\lambda(x-x_0)} = [0, \lambda_1, \ldots, \lambda_n]e^{\lambda(x-x_0)}.$$

The advantage of this divided difference identity for g is that, unlike identity (8.7), the divided difference formula for g is well defined also in the case where the nodes are not distinct and, in particular, when some of the nodes are equal to 0. Note that divided differences with coalescent nodes are well defined for smooth functions, including exponential functions. Based on these facts, the proof then follows along the same lines as the proof for distinct nonzero λ_i's.

Note that the above estimates are not optimal since an optimal value of x_0 may be different from the values $0, 1/2, 1$ used above. However, it is clear that the estimates in Theorem 8.12 are generally better than those of Vatchev. With the conventions $a_n = 1$, $m = 0$, $p = \infty$, and where it was assumed that the interval under consideration has length 1, Vatchev's inequality in Theorem 8.4 becomes

$$\|f - S_n(f)\|_{L_\infty(I)} \leq \frac{e^{\bar{\lambda}}}{2^{n-1}\bar{\lambda}^{n-1}}\|Lf\|_{L_\infty(I)},$$

In particular, Vatchev's estimate does not contain the factor $(n-1)!$. Moreover, the bounds in Theorem 8.12 are smaller for "large" values of $\bar{\lambda}$. For example, for $|\lambda_1| << \bar{\lambda}$, the value $e^{-\lambda_1}$ will be much smaller than $e^{\bar{\lambda}}/\bar{\lambda}^n$. The same is also true for positive λ_1. Also, if $\lambda_1 \geq n$, then the bound in Theorem 8.12 is not larger than $\frac{e^{-n}}{(n-1)!}\|Lf\|_{L_\infty(I)}$ whereas Vatchev's is at least $\frac{e^n}{2^{n-1}n^n}\|Lf\|_{L_\infty(I)}$. By Stirling's formula, the ratio of the two expressions, $\frac{e^n}{2^{n-1}n^n}/\frac{e^{-n}}{(n-1)!} \sim \frac{e^n\sqrt{2\pi n}}{2^{n-1}n} \to \infty$, as $n \to \infty$.

Chapter 9

Approximation Power of Cascade Networks

The goal of this chapter is to approximate univariate polynomials and smooth functions on $I = [0, 1]$ using cascade networks and investigate the complexity of the representation of these functions relative to ReLU neural networks.

First, an analogous result to the result of Proposition 2.13 of Yarotsky [33] is given. If $f \in \Pi_2$, then f can be approximated by a cascade network $\mathcal{Y}_L$ described in Definition 3.3 having depth, number of weights, and units equal to $\mathcal{O}\left(\ln(1/\varepsilon)\right)$.

Proposition 9.1. Let $f \in \Pi_2$, and $\|f\|_{L_\infty(I)} = 1$. Let $\varepsilon > 0$ and $\Omega_L = \{0, h_L, \ldots, 1\}, h_L = 2^{-L+1}$. There exists a cascade network $\mathcal{Y}_L$ such that

$$\|f - \mathcal{Y}_L\|_{L_\infty(I)} < \varepsilon,$$

with depth

$$\mathcal{L}(\mathcal{Y}_L) = \mathcal{O}(\ln(1/\varepsilon)),$$

number of weights

$$\mathcal{W}(\mathcal{Y}_L) = \mathcal{O}(\ln(1/\varepsilon)),$$

number of units

$$\mathcal{U}(\mathcal{Y}_L) = \mathcal{O}(\ln(1/\varepsilon)),$$

and width

$$\mathcal{M}(\mathcal{Y}_L) = 3.$$

Proof. Set $\mathcal{Y}_L$ to be the piecewise linear interpolant of f on Ω_L.

It is known, that for smooth f [63],

$$\begin{aligned}\|f - \mathcal{Y}_L\|_{L_\infty(I)} &\leq (1/4)h_L^2\|f^{(2)}\|_{L_\infty(I)}\\ &\leq (1/4)(2^{-2L+2})(4/3)(n^2)(n^2-1)\|f\|_{L_\infty(I)},\end{aligned}$$

where the second inequality follows from Markov's Inequality [64].

If $n = 2$, then

$$\|f - \mathcal{Y}_L\|_{L_\infty(I)} \leq (16)(2^{-2L})\|f\|_{L_\infty(I)}.$$

Let L be the smallest integer such that $(16)(2^{-2L}) \leq \varepsilon$. Then $L = \lceil 2 + (1/2)\log(1/\varepsilon)\rceil$ and thus $\mathcal{L}(\mathcal{Y}_L) = \mathcal{O}(\ln(1/\varepsilon))$ for ε sufficiently small.

As $\mathcal{W}(\mathcal{Y}_L) \leq 9L - 12$, $\mathcal{W}(\mathcal{Y}_L) = \mathcal{O}(\ln(1/\varepsilon))$. As $\mathcal{U}(\mathcal{Y}_L) = 3L - 2$, $\mathcal{U}(\mathcal{Y}_L) = \mathcal{O}(\ln(1/\varepsilon))$. $\mathcal{M}(\mathcal{Y}_L) = 3$. □

Proposition 9.1 can be extended to hold for polynomials of degree $n > 2$.

Proposition 9.2. Let $f \in \Pi_n$, $n \geq 2$ and $\|f\|_{L_\infty(I)} = 1$. Let $\varepsilon > 0$ and $\Omega_L = \{0, h_L, \ldots, 1\}$, $(h_L = 2^{-L+1})$. There exists a cascade network $\mathcal{Y}_L$ such that

$$\|f - \mathcal{Y}_L\|_{L_\infty(I)} < \varepsilon$$

with depth

$$\mathcal{L}(\mathcal{Y}_L) = \mathcal{O}(\ln(n) + \ln(1/\varepsilon)),$$

number of weights

$$\mathcal{W}(\mathcal{Y}_L) = \mathcal{O}(n^2\ln(n) + n^2\ln(1/\varepsilon)),$$

number of units

$$\mathcal{U}(\mathcal{Y}_L) = \mathcal{O}(n\ln(n) + n\ln(1/\varepsilon)),$$

and width

$$\mathcal{M}(\mathcal{Y}_L) = n + 1.$$

Proof. Set $\mathcal{Y}_L$ to be the piecewise linear interpolant of f on Ω_L.

Then,

$$\|f - \mathcal{Y}_L\|_{L_\infty(I)} \leq (1/4)h_L^2 \left\|f^{(2)}\right\|_{L_\infty(I)} \tag{9.1}$$

$$\leq (1/4)h_L^2(4/3)n^2(n^2-1)\|f\|_{L_\infty(I)} \tag{9.2}$$

$$\leq (1/3)(2^{-2L+2})n^4\|f\|_{L_\infty(I)} \tag{9.3}$$

Let L be the smallest integer such that $(1/3)(n^4)(2^{-2L+2}) \leq \varepsilon$. Then $L = \frac{1}{2}\log\left(\frac{4n^4}{3\varepsilon}\right)$ and, thus, $\mathcal{L}(\mathcal{Y}_L) = \mathcal{O}(\ln(n) + \ln(1/\varepsilon))$ for ε sufficiently small.

As $\mathcal{W}(\mathcal{Y}_L) \leq 2(n+1) + (n+1)^2(L-2)$, it follows that $\mathcal{W}(\mathcal{Y}_L) = \mathcal{O}(n^2\ln(n) + n^2(\ln(1/\varepsilon))$. As $\mathcal{U}(\mathcal{Y}_L) = (n+1)(L-1)+1$, it follows that $\mathcal{U}(\mathcal{Y}_L) = \mathcal{O}(n\ln(n) + n\ln(1/\varepsilon))$. In addition, $\mathcal{M}(\mathcal{Y}_L) = n+1$.

□

Remark 9.1. *The result of Proposition 9.2 is similar to that of Proposition 2.16 with* $D = 1$*. However, the requirement* $\|f\|_{L_\infty(I)} = 1$ *is stronger than the condition* $\max_{0\leq k\leq n} a_k \leq A$ *in Proposition 2.16 for a fixed* A*. Also note that for the Chebyshev polynomials* $T_n(x)$*,* $x \in [-1,1]$*, then* $\|T_n\|_{L_\infty(I)} = 1$*, and* T_n *satisfies the results of Proposition 9.2. However, the constant* A *grows exponentially with* n*, which would negatively impact the complexity estimates of Proposition 2.16*

Remark 9.2. *If the condition of Proposition 9.2 is strengthened to consider* $f \in \Pi_n$ *and* $f \in F_{n,1}$ *for* $n \geq 2$ *(*$F_{n,d}$ *defined as in equation (2.14)), the dependency on* n *will disappear. As* $\left\|f^{(2)}\right\|_{L_\infty(I)} \leq 1$*, it follows that*

$$\|f - \mathcal{Y}_L\|_{L_\infty(I)} \leq (1/4)h_L^2 \left\|f^{(2)}\right\|_{L_\infty(I)}$$

$$\leq (1/4)(2^{-2L+2}).$$

Thus there exists a cascade network $\mathcal{Y}_L$ such that

$$\|f - \mathcal{Y}_L\|_{L_\infty(I)} < \varepsilon,$$

with depth

$$\mathcal{L}(\mathcal{Y}_L) = \mathcal{O}(\ln(1/\varepsilon)),$$

number of weights

$$\mathcal{W}(\mathcal{Y}_L) = \mathcal{O}(\ln(1/\varepsilon)),$$

number of units

$$\mathcal{U}(\mathcal{Y}_L) = \mathcal{O}(\ln(1/\varepsilon)),$$

and width

$$\mathcal{M}(\mathcal{Y}_L) = n + 1.$$

The following is an analog of Theorem 2.17.

Theorem 9.3. *Let $f \in F_{n,1}$ for some $n \in \mathbb{N}$. Let $\varepsilon > 0$ and $\Omega_L = \{0, 2^{-L+1}, \dots\}$, then there exists a cascade network $\mathcal{Y}_L$ such that*

$$\|f - \mathcal{Y}_L\|_{L_\infty(I)} < \varepsilon \tag{9.4}$$

with depth

$$\mathcal{L}(\mathcal{Y}_L) = \mathcal{O}(ln(1/\varepsilon)),$$

number of weights

$$\mathcal{W}(\mathcal{Y}_L) = \mathcal{O}(n\varepsilon^{-1/n}(\ln(n) + \ln(1/\varepsilon)),$$

number of units

$$\mathcal{U}(\mathcal{Y}_L) = \mathcal{O}(\varepsilon^{-1/n}(\ln(n) + \ln(1/\varepsilon)),$$

and width

$$\mathcal{M}(\mathcal{Y}_L) = N(n+1).$$

Proof. The proof will follow a similar outline as the proof of Theorem 2.17 in [33]. First f is approximated by combination f_1 of piecewise linear functions and local Taylor polynomials. Then, Corollary 3.3 and the remark after Proposition 9.2, is used to approximate f_1 by a cascade network.

Let $N = 2^M, M \in \mathbb{N}$ to be selected later, and let $I_k = [k/N, (k+1)/N]$ for $k = 0, \ldots, N-1$. Write

$$f_1(x) = \sum_{k=0}^{N} B_{k/N}(x) P_{k/N}^{n-1}(x), \tag{9.5}$$

where $P_{k/N}^{n-1}(x)$ are the $(n-1)$ degree Taylor polynomials of f centered at $x = k/N$ and $B_{k/N}$ are piecewise linear hat functions with knots $\frac{k-1}{N}, \frac{k}{N}, \frac{k+1}{N}$, normalized to form a partition of unity over I. Then, f_1 is a piecewise polynomial of degree n and, for $x \in I$,

$$\begin{aligned}
|f(x) - f_1(x)| &= \left| f(x) - \sum_{k=0}^{N} B_{k/N}(x) P_{k/N}^{n-1}(x) \right| \\
&= \left| \sum_{k=0}^{N} B_{k/N}(x)(f(x) - P_{k/N}^{n-1}(x) \right| \\
&\leq 2 \max_{\{x \in I : \|x - k/N\| < 1/N\}} \left| f(x) - P_{k/N}^{n-1}(x) \right| \\
&\leq 2(1/n!) \left\| f^{(n)} \right\|_{L_\infty(I)} \left(\frac{1}{N} \right)^n \\
&\leq \frac{2}{n!} \left(\frac{1}{N} \right)^n
\end{aligned}$$

since for a fixed k and $x \in I_k$,

$$\begin{aligned}
|f(x) - f_1(x)| &= \left| B_{k/N}(x) \left(f(x) - P_{k/N}^{n-1}(x) \right) + B_{(k+1)/N}(x) \left(f(x) - P_{(k+1)/N}^{n-1}(x) \right) \right| \\
&\leq \left| f(x) - P_{k/N}^{n-1}(x) \right| + \left| f(x) - P_{(k+1)/N}^{n-1}(x) \right| .
\end{aligned}$$

Now, choose N such that $\frac{2}{n!}\left(\frac{1}{N}\right)^n \leq \varepsilon/2$, or $N = \lceil\left(\frac{4}{\varepsilon n!}\right)^{1/n}\rceil$.

Next, approximate f_1 by f_2, which is the piecewise linear interpolant of f_1 on Ω_{M+K+1}, for some K to be chosen later. By Corollary 3.3, f_2 can be realized by a cascade network $\mathcal{Y}_L$, with $L = M + K + 1$, where $\mathcal{Y}_L$ is the splicing of $\mathcal{Y}_K^1, \ldots, \mathcal{Y}_K^N$.

Therefore,

$$\|\mathcal{Y}_L - f_1\|_{L_\infty(I)} \leq (1/4)\left(\frac{1}{2^{M+K}}\right)^2 \left\|f_1^{(2)}\right\|_{L_\infty(I)}.$$

To find $\left\|f_1^{(2)}\right\|_{L_\infty(I)}$, fix x in the interior of I_k, then

$$f_1^{(2)}(x) = N(P_{k/N}^{n-1})'(x)+N(P_{(k+1)/N}^{n-1})'(x)+(B_{k/n)}(x))(P_{k/N}^{n-1})^{(2)}(x)+(B_{(k+1)/n)}(x))(P_{(k+1)/N}^{n-1})^{(2)}(x)$$

and

$$\begin{aligned}
\left|(P_{k/N}^{n-1})'(x)\right| &= \left|\frac{f'(k/N)(x-k/N)^0}{1!} + (2)\frac{f^{(2)}(k/N)(x-k/N)^1}{2!}\right. \\
&\quad \left. + \cdots + \frac{f^{(n-1)}(k/N)(x-k/N)^{(n-2)}}{(n-2)!}\right| \\
&\leq e^{1/N} \\
&< e.
\end{aligned}$$

Similarly, $|(P_{k/N}^{n-1})^{(2)}(x)| < e$. Therefore, $\left\|f_1^{(2)}(x)\right\|_{L_\infty(I)} \leq 2e(N+1) \leq 11N$.

Choose K such that $(\frac{11}{4})\frac{1}{N2^{2K}} \leq \varepsilon/2$, where N has already been selected.

Therefore,

$$\begin{aligned}
\|f - \mathcal{Y}_L\|_{L_\infty(I)} &\leq \|f - f_1\|_{L_\infty(I)} + \|f_1 - \mathcal{Y}_L\|_{L_\infty(I)} \\
&\leq \varepsilon/2 + \varepsilon/2 \\
&= \varepsilon.
\end{aligned}$$

The depth of $\mathcal{Y}_L$ is $L = M + K + 1$, where

$$M = \log(N) = \left(\frac{1}{n}\right)\log\left(\frac{4}{\varepsilon n!}\right), \quad K = \frac{1}{2}\log\left(\frac{11}{2\varepsilon N}\right).$$

gives

$$L = \left(\frac{1}{2}\right)\log\left(\frac{11}{2}\right) + 1 + \left(\frac{1}{2}\right)(1+\frac{1}{n})\log\left(\frac{1}{\varepsilon}\right) + \frac{1}{2n}\log\left(\frac{4}{n!}\right).$$

Thus,

$$\mathcal{L}(\mathcal{Y}_L) = \mathcal{O}(\ln(1/\varepsilon)).$$

Then, by Proposition 9.2 and Corollary 3.3, $\mathcal{Y}_L$ has number of units

$$\begin{aligned}
\mathcal{U}(\mathcal{Y}_L) &= \mathcal{O}(N\mathcal{U}(\mathcal{Y}_k^1)) \\
&= \mathcal{O}\left(n\left(\left(\frac{4}{n!}\right)^{1/n}\varepsilon^{-1/n}(\ln(n)+\ln(2/\varepsilon)\right)\right) \\
&= \mathcal{O}(\varepsilon^{-1/n}(\ln(n)+\ln(1/\varepsilon)),
\end{aligned}$$

number of weights

$$\begin{aligned}
\mathcal{W}(\mathcal{Y}_L) &= \mathcal{O}(N\mathcal{W}(\mathcal{Y}_K^1)) \\
&= \mathcal{O}\left(n^2\left(\frac{4}{n!}\right)^{1/n}\varepsilon^{-1/n}(\ln(n)+\ln(2/\varepsilon)\right) \\
&= \mathcal{O}(n\varepsilon^{-1/n}(\ln(n)+\ln(1/\varepsilon)),
\end{aligned}$$

and width

$$\mathcal{M}(\mathcal{Y}_L) = \mathcal{O}(n+1).$$

□

Remark 9.3. *For any fixed n and for $d = 1$, the result of Theorem 9.3 is consistent with that of Yarotsky. However, in Theorem 2.17 the constant is not explicit, whereas in Theorem 9.3 the* *dependency on n is explicit. In addition, it is not clear how fast Yarotsky's constant grows with n.*

Remark 9.4. *In the proof of Theorem 9.3, equal contribution from the approximation of f by f_1 and the approximation of f_1 by $\mathcal{Y}_L$ was assumed. However, an alternate approach would be to minimize $L = M + K + 1$ subject to the constraint $\frac{2}{n!}\left(\frac{1}{N}\right)^n + (\frac{11}{4})\frac{1}{N2^{2K}} \leq \varepsilon$, where $N = 2^M$. The details are omitted, as the results were not significantly better and the rates were essentially the same as found in Theorem 9.3 .*

Remark 9.5. *Just as is the case for neural networks, shallow cascade networks do not do as well as deep cascade networks. In Theorem 9.3, the case where each of the N intervals of I is divided into 2 subintervals (i.e the case $K = 1$), then f_1 can be approximated by a cascade network $\mathcal{Y}_L$ where $\mathcal{Y}_L$ is the "splicing" of the cascade networks $\mathcal{Y}_2^1, \ldots, \mathcal{Y}_2^N$ all with depth 2. Then, by Corollary 3.3, for shallow cascade networks $\mathcal{U}(\mathcal{Y}_L) = \mathcal{O}(N\mathcal{U}(\mathcal{Y}_2^1)) = \mathcal{O}(N) = \mathcal{O}(\varepsilon^{-1/2})$, whereas for deep networks $\mathcal{U}(\mathcal{Y}_L) = \mathcal{O}(\ln(1/\varepsilon))$.*

Chapter 10

Numerical Examples

In the following, the approximation power of cascade networks is illustrated with some numerical examples. Several univariate functions were chosen to investigate how well they are approximated by a CN of the form $\mathcal{Y}_L = A_L \left(A_{L-1} \circ \alpha\right) \cdots \left(A_1 \circ \alpha^{L-1}\right) \left(\mathcal{Y}_0 \circ \alpha^L\right)$, for different values of L (depth) and W (width).

All test were preformed with the software Matlab R2022a. The matrices $A_\ell, \ell = 1, \ldots L$, were initialized using the Matlab function "randn(n,m)", which returns an $n \times m$ matrix of normally distributed random numbers. The minimum of the objective function (L_2 or L_∞) was computed using the Matlab function "fmincon", which find the minimum of a constrained nonlinear function.

Example 10.1. Consider the analytic function $f(x) = e^{2x} + e^x$ (Figure 10.1).

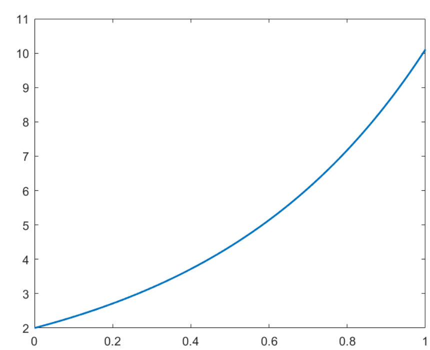
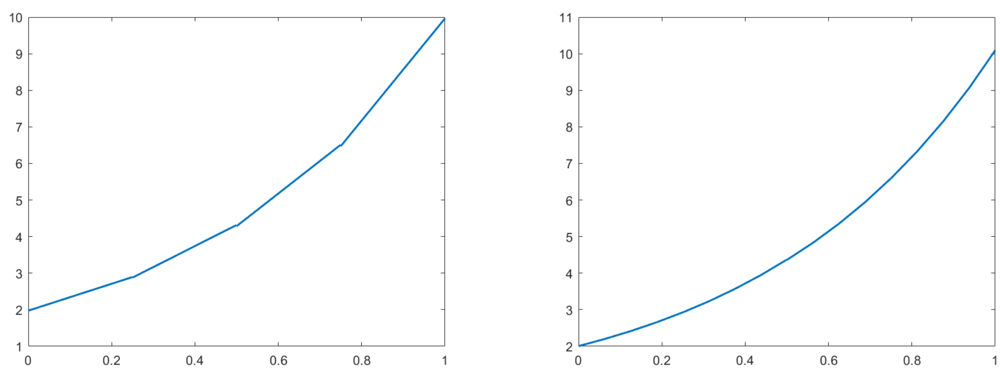


Figure 10.1: Analytic Function $f(x) = e^{2x} + e^x$

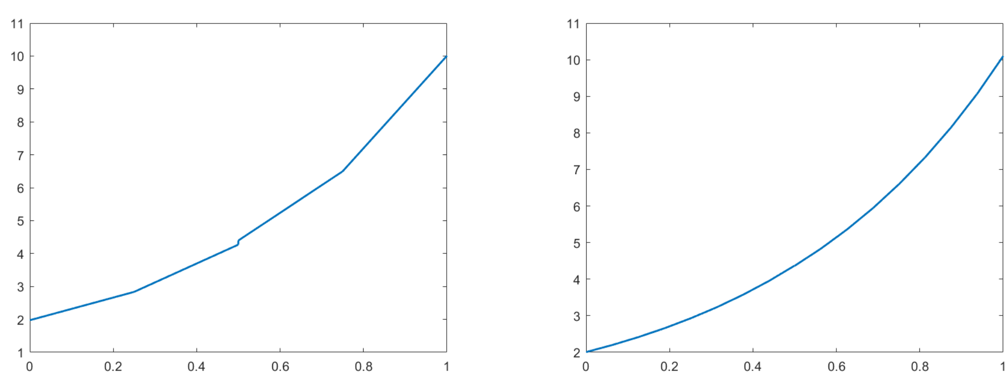
L	n^o of Parameters	Error L_2	Ratio	Error L_∞	Ratio
1	10	1.44949e-01		3.21215e-01	
2	14	3.78217e-02	3.83	9.99248e-02	3.21
3	18	9.87025e-03	3.83	2.80104e-02	3.57
4	22	3.47667e-03	2.84	7.42489e-03	3.77
5	26	2.58779e-03	1.34	5.20243e-03	1.43
6	30	2.52067e-03	1.03	1.28274e-01	0.04

Table 10.1: Table of Errors, Analytic Function, $W = 1$



(a) $L = 2$ (b) L=4

Figure 10.2: CN Approximation to Analytic Function, L_2 Objective Function, $W = 1$

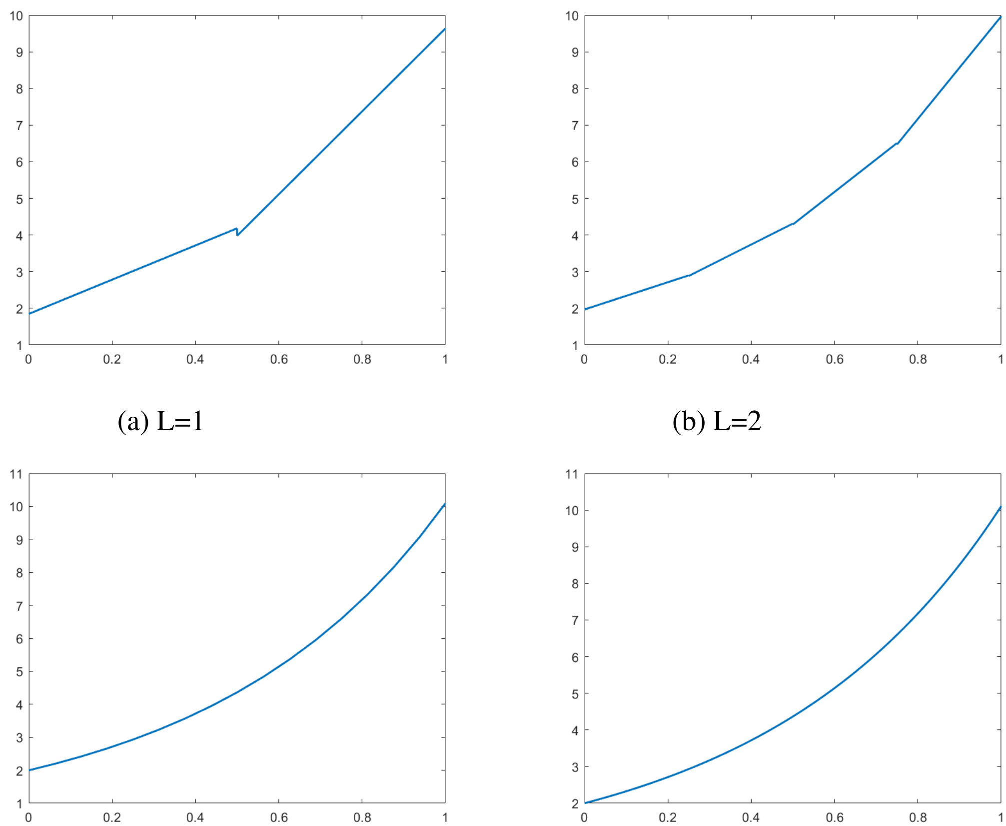


(a) L=2 (b) L=4

Figure 10.3: CN Approximation to Analytic Function, L_∞ Objective Function, $W = 1$

L	n^o of Parameters	Error L_2	Ratio	Error L_∞	Ratio
1	22	1.44949e-01		3.21220e-01	
2	34	3.77421e-02	3.84	9.99348e-02	3.21
3	46	9.53972e-03	3.96	2.80118e-02	3.57
4	58	2.39943e-03	3.98	7.42501e-03	3.77
5	70	5.98639e-04	4.01	1.88198e-03	3.95
6	82	1.57408e-04	3.80	9.46322e-04	1.99
7	94	7.59855e-05	2.07	1.39021e-03	0.68
8	106	5.15284e-05	1.47	6.33918e-02	0.02

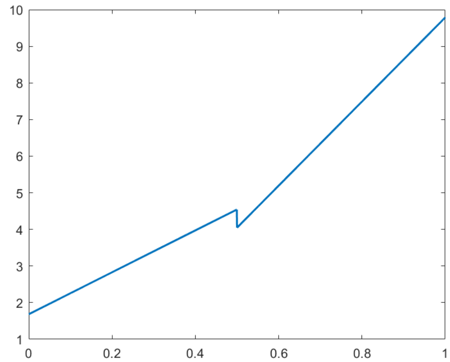
Table 10.2: Table of Errors, Analytic Function, $W = 2$



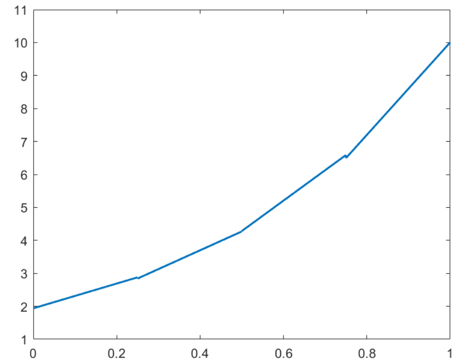
(a) L=1 (b) L=2

(c) L=4 (d) L=8

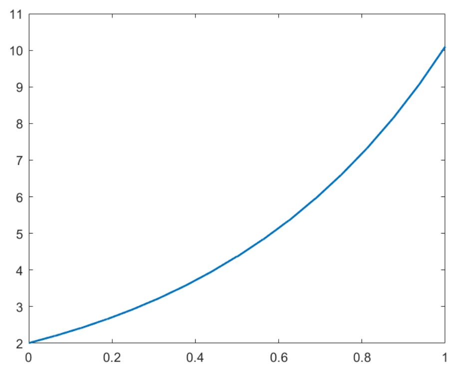
Figure 10.4: CN Approximation to Analytic Function, L_2 Objective Function, $W = 2$



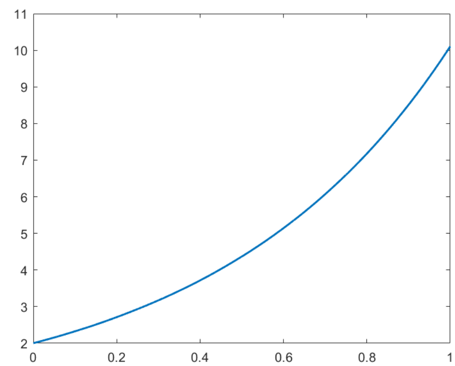
(a) L=1



(b) L=2



(c) L=4



(d) L=8

Figure 10.5: CN Approximation to Analytic Function, L_∞ Objective Function, $W = 2$

Example 10.2. Consider the Weierstrass function $f(x) = \sum_{n=0}^{\infty} a^n \cos(b^n \pi x)$, where $0 < a < 1$, b a positive real number, and $ab > 1$ (Figure 10.6). The Weierstrass function is continuous and nowhere differentiable on $[0, 1]$.

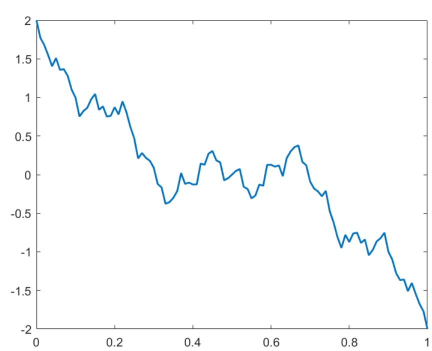
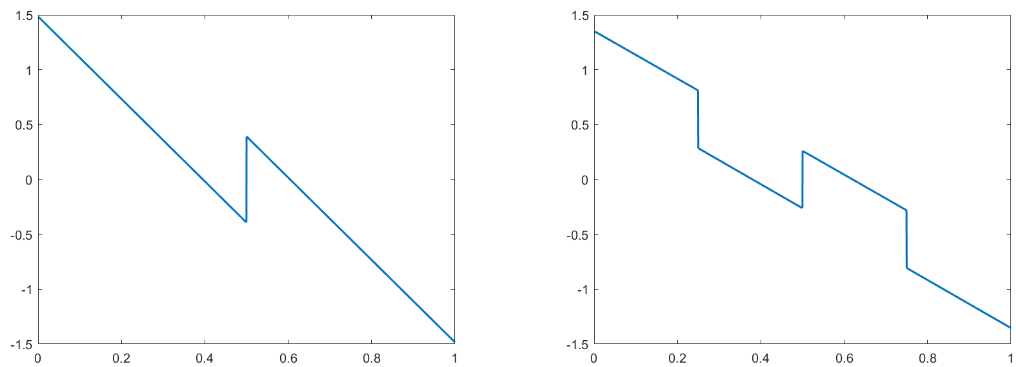


Figure 10.6: Weierstrass Function with $a = 0.5, b = 3$

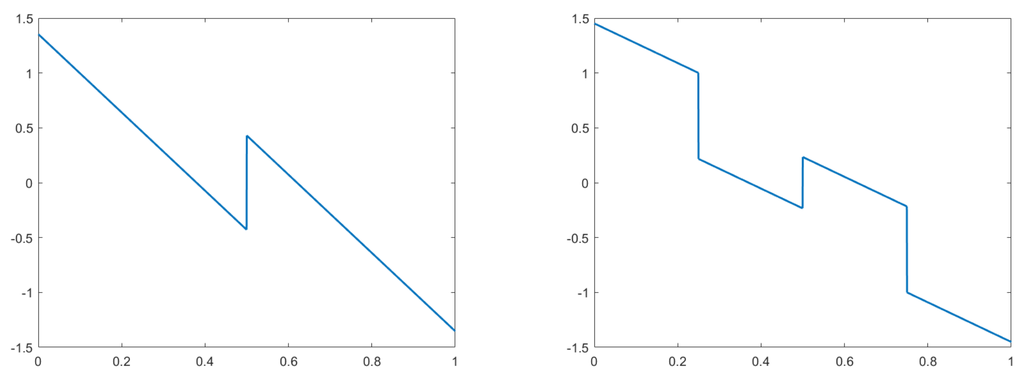
L	n^o of Parameters	Error L_2	Ratio	Error L_∞	Ratio
1	10	2.75487e-01		6.46546e-01	
2	14	2.42169e-01	1.14	5.48770e-01	1.18
3	18	2.33813e-01	1.04	4.92697e-01	1.11
4	22	2.33558e-01	1.00	5.02062e-01	0.98

Table 10.3: Table of Errors, Weierstrass Function, $W = 1$



(a) L=1 (b) L=2

Figure 10.7: CN Approximation to Weierstrass Function, L_2 Objective Function, $W = 1$

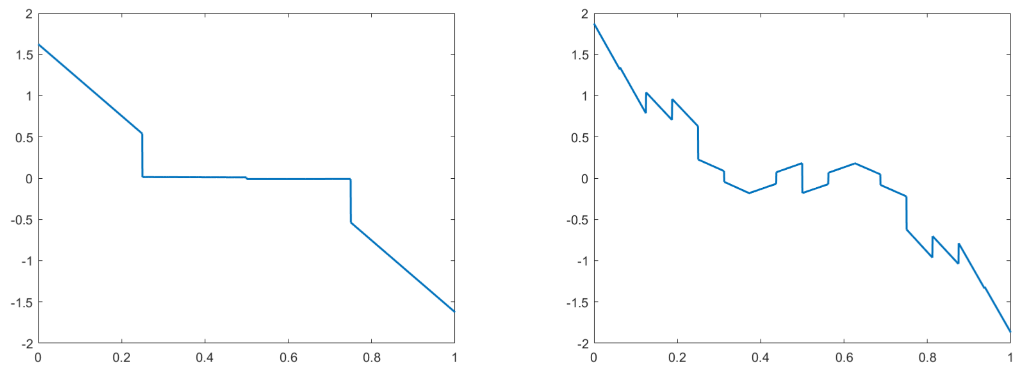


(a) L=1 (b) L=2

Figure 10.8: CN Approximation to Weierstrass Function, L_∞ Objective Function, $W = 1$

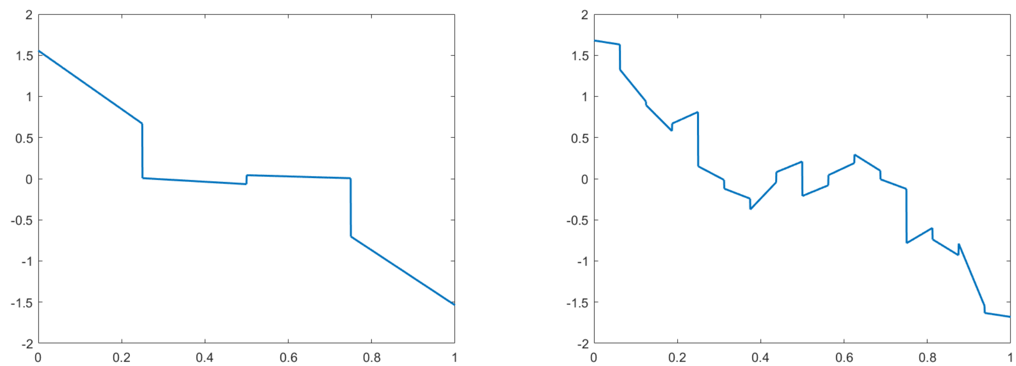
L	n^o of Parameters	Error L_2	Ratio	Error L_∞	Ratio
1	22	2.75487e-01		6.83599e-01	
2	34	1.84684e-01	1.49	4.64780e-01	1.47
3	46	1.34510e-01	1.37	3.69778e-01	1.26
4	58	1.28606e-01	1.05	3.28335e-01	1.13
5	70	1.26367e-01	1.02	3.54893e-01	0.93
6	82	1.26191e-01	1.00	3.04675e-01	1.16

Table 10.4: Table of Errors, Weierstrass Function, $W = 2$



(a) L=2 (b) L=4

Figure 10.9: CN Approximation to Weierstrass Function, L_2 Objective Function, $W = 2$

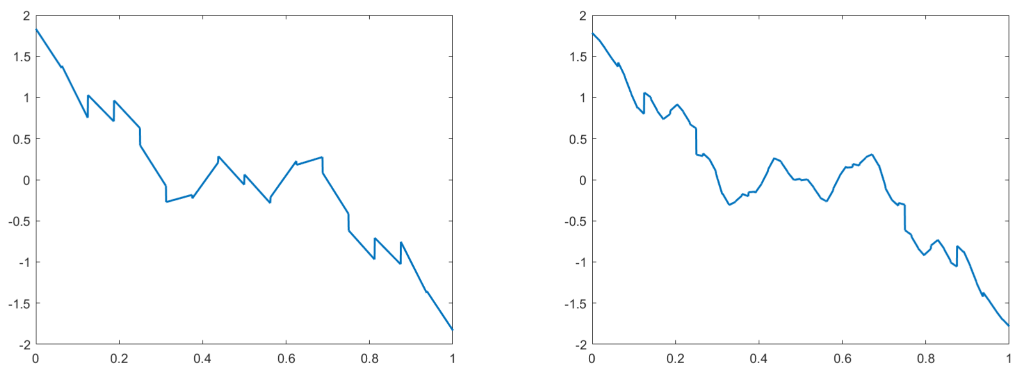


(a) L=2 (b) L=4

Figure 10.10: CN Approximation to Weierstrass Function, L_∞ Objective Function, $W = 2$

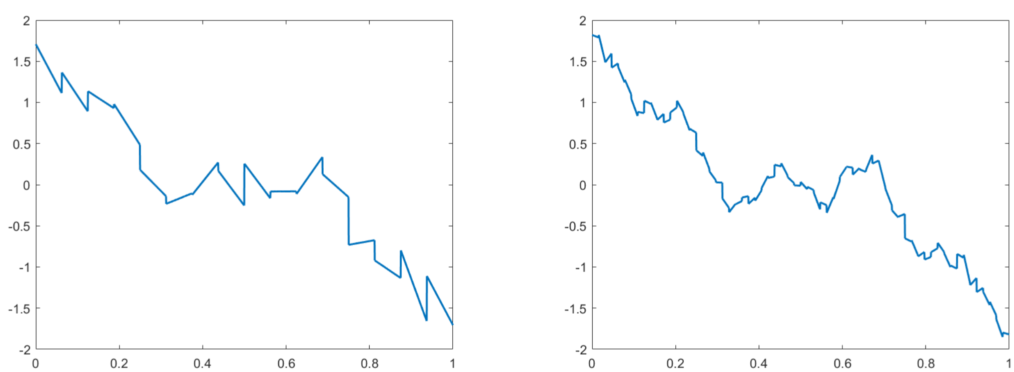
L	n^o of Parameters	Error L_2	Ratio	Error L_∞	Ratio
1	38	2.75487e-01		6.46551e-01	
2	62	1.84684e-01	1.49	4.64780e-01	1.39
3	86	1.21636e-01	1.52	3.26181e-01	1.42
4	110	8.88133e-02	1.37	2.92459e-01	1.12
5	134	8.31038e-02	1.07	1.90689e-01	1.53
6	158	8.18399e-02	1.02	1.77913e-01	1.07

Table 10.5: Table of Errors, Weierstrass Function, $W = 3$



(a) L=4 (b) L=6

Figure 10.11: CN Approximation to Weierstrass Function, L_2 Objective Function, $W = 3$

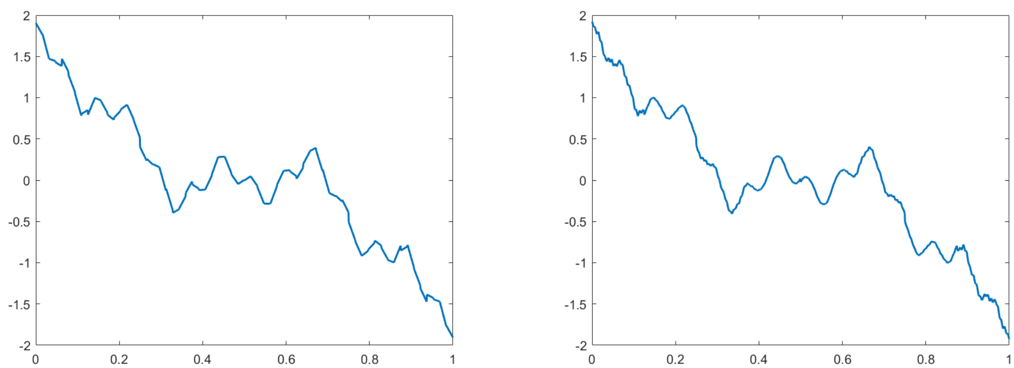


(a) L=4 (b) L=6

Figure 10.12: CN Approximation to Weierstrass Function, L_∞ Objective Function, $W = 3$

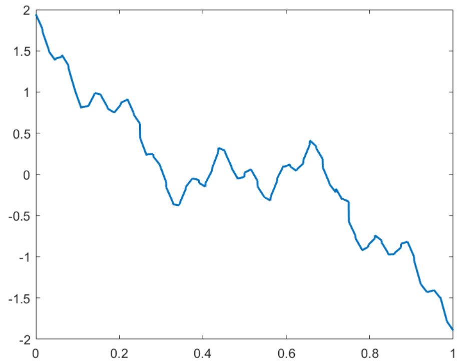
L	n^o of Parameters	Error L_2	Ratio	Error L_∞	Ratio
1	58	2.75487e-01		6.46548e-01	
2	98	1.84684e-01	1.49	4.64782e-01	1.39
3	138	1.21636e-01	1.52	3.26181e-01	1.42
4	178	8.24624e-02	1.48	2.12589e-01	1.53
5	218	5.50145e-02	1.50	1.50539e-01	1.41
6	258	4.97521e-02	1.11	1.16694e-01	1.29
7	298	4.90660e-02	1.01	1.45062e-01	0.80
8	338	4.85139e-02	1.01	1.15275e-01	1.26

Table 10.6: Table of Errors, Weierstrass Function, $W = 4$

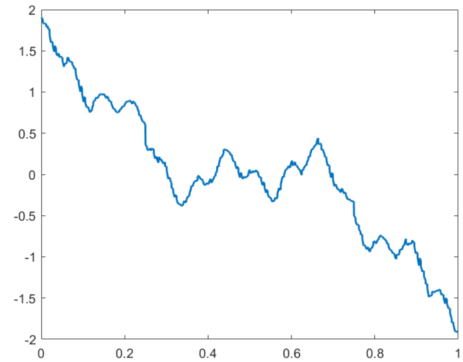


(a) L=6 (b) L=8

Figure 10.13: CN Approximation to Weierstrass Function, L_2 Objective Function, $W = 4$



(a) L=6



(b) L=8

Figure 10.14: CN Approximation to Weierstrass Function, L_∞ Objective Function, $W = 4$

Chapter 11

Discussion

Generalized neural networks are functions resulting from repeatedly applying a fixed operator, in general nonlinear, to an affine operator. Cascade networks are generalized neural networks. The connection between cascade networks, subdivison algorithms, and the cascade algorithm was investigated. Sequences of scalars or vectors obtained by subdivision can be viewed as restrictions of functions generated by CN's to dyadic meshes. Moreover, the cascade algorithm is a special case of cascade networks.

The space of functions obtained by a CN of fixed width and linear input, $\mathcal{S}_{W,L}$, was characterized. Further, conditions for elements of $\mathcal{S}_{W,L}$ to converge to a C^0 function were established. It was shown that the only smooth functions in $\mathcal{S}_W$, the closure of $\bigcup_{L\in\mathbb{N}} \mathcal{S}_{W,L}$, are combinations of exponential functions, or more generally, elements of the null space of constant coefficient differential operators.

In terms of complexity, cascade networks were shown to approximate univariate polynomials and smooth functions at similar rates when compared to known results for ReLU neural networks. It is currently being investigated whether the recent results of Daubechies et al. [37] and Yarotsky [36] hold for cascade networks. We conjecture that cascade networks are also able to approximate Lipschitz functions with error $\mathcal{O}\left(\frac{1}{N\log(N)}\right)$, instead of $\mathcal{O}\left(\frac{1}{N}\right)$, which is the error obtainable by linear splines with N knots.

The study of cascade networks has lead to many interesting questions and possible areas of further investigation. One such area is studying the numerical aspects of cascade networks. In particular, investigating whether it is possible to find the weights of the network in an appropriate manner. A major challenge with ReLU networks is finding an optimal solution to the minimization problem.

Moreover, given the piecewise affine nature of the cascade network, another possible direction for future study includes obtaining explicit approximation schemes. One question is whether the weights of the network can be expressed by appropriately defined "dual functionals" resembling the dual functionals employed in quasi-interpolation spline methods. In addition, investigating whether it is possible to move away from the uniform setting, and consider non-uniform and adaptive meshes.

References

[1] S.O. Haykin. *Neural Networks and Learning Machines*. Pearson Education, 2011.

[2] M. Anthony and P.L. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 2009.

[3] Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta numerica*, 8:143–195, 1999.

[4] Warren S McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[5] Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.

[6] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[8] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, et al. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal processing magazine*, 29, 2012.

[9] Tomáš Mikolov, Anoop Deoras, Daniel Povey, Lukáš Burget, and Jan Černockỳ. Strategies for training large scale neural network language models. In *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*, pages 196–201. IEEE, 2011.

[10] Junshui Ma, Robert P Sheridan, Andy Liaw, George E Dahl, and Vladimir Svetnik. Deep neural nets as a method for quantitative structure–activity relationships. *Journal of chemical information and modeling*, 55(2):263–274, 2015.

[11] Ronald DeVore, Boris Hanin, and Guergana Petrova. Neural network approximation. *Acta Numerica*, 30:327–444, 2021.

[12] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[13] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[14] David E Rumelhart, Richard Durbin, Richard Golden, and Yves Chauvin. Backpropagation: The basic theory. *Backpropagation: Theory, architectures and applications*, pages 1–34, 1995.

[15] J Opschoor, P Petersen, and Christoph Schwab. Deep ReLU networks and high-order finite element methods. *SAM, ETH Zürich*, 2019.

[16] Philipp Grohs, Dmytro Perekrestenko, Dennis Elbrächter, and Helmut Bölcskei. Deep neural network approximation theory. *arXiv preprint arXiv:1901.02220*, 2019.

[17] Philipp Petersen and Felix Voigtlaender. Optimal approximation of piecewise smooth functions using deep ReLU neural networks. *Neural Networks*, 108:296–330, 2018.

[18] Dennis Elbrächter, Dmytro Perekrestenko, Philipp Grohs, and Helmut Bölcskei. Deep neural network approximation theory. *IEEE Transactions on Information Theory*, 67(5):2581–2623, 2021.

[19] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[20] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[21] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[22] Allan Pinkus. Approximating by ridge functions. *Surface fitting and multiresolution methods*, pages 279–292, 1997.

[23] Andrew R Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, 39(3):930–945, 1993.

[24] Andrew R Barron. Approximation and estimation bounds for artificial neural networks. *Machine learning*, 14(1):115–133, 1994.

[25] Hrushikesh N Mhaskar. Neural networks for optimal approximation of smooth and analytic functions. *Neural computation*, 8(1):164–177, 1996.

[26] Vitaly Maiorov and Allan Pinkus. Lower bounds for approximation by MLP neural networks. *Neurocomputing*, 25(1-3):81–91, 1999.

[27] Yoshua Bengio and Olivier Delalleau. On the expressive power of deep architectures. In *International Conference on Algorithmic Learning Theory*, pages 18–36. Springer, 2011.

[28] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pages 2924–2932, 2014.

[29] Matus Telgarsky. Representation benefits of deep feedforward networks. *arXiv preprint arXiv:1509.08101*, 2015.

[30] Raman Arora, Amitabh Basu, Poorya Mianjy, and Anirbit Mukherjee. Understanding deep neural networks with rectified linear units. *arXiv preprint arXiv:1611.01491*, 2016.

[31] Hrushikesh N Mhaskar and Tomaso Poggio. Deep vs. shallow networks: An approximation theory perspective. *Analysis and Applications*, 14(06):829–848, 2016.

[32] Tomaso Poggio, Hrushikesh Mhaskar, Lorenzo Rosasco, Brando Miranda, and Qianli Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*, 14(5):503–519, 2017.

[33] Dmitry Yarotsky. Error bounds for approximations with deep ReLU networks. *Neural Networks*, 94:103–114, 2017.

[34] Hadrien Montanelli, Haizhao Yang, and Qiang Du. Deep ReLU networks overcome the curse of dimensionality for bandlimited functions. *arXiv preprint arXiv:1903.00735*, 2019.

[35] Shiyu Liang and Rayadurgam Srikant. Why deep neural networks for function approximation? *arXiv preprint arXiv:1610.04161*, 2016.

[36] Dmitry Yarotsky. Quantified advantage of discontinuous weight selection in approximations with deep neural networks. *arXiv preprint arXiv:1705.01365*, 2017.

[37] Ingrid Daubechies, Ronald DeVore, Simon Foucart, Boris Hanin, and Guergana Petrova. Nonlinear approximation and (deep) ReLU networks. *arXiv preprint arXiv:1905.02199*, 2019.

[38] Ingrid Daubechies. *Ten lectures on wavelets*, volume 61. Siam, 1992.

[39] Costanza Conti and Nira Dyn. Non-stationary subdivision schemes: State of the art and perspectives. In *International Conference Approximation Theory*, pages 39–71. Springer, 2019.

[40] Nira Dyn and David Levin. Subdivision schemes in geometric modelling. *Acta Numerica*, 11(0):73–144, 2002.

[41] Nira Dyn. Subdivision schemes in CAGD. *Advances in numerical analysis*, 2:36–104, 1992.

[42] Nira Dyn and David Levin. Analysis of hermite-type subdivision schemes. *Series in Approximations and Decompositions*, 6:117–124, 1995.

[43] George Merrill Chaikin. An algorithm for high-speed curve generation. *Computer graphics and image processing*, 3(4):346–349, 1974.

[44] Charles A Micchelli and Thomas Sauer. On vector subdivision. *Mathematische Zeitschrift*, 229(4):621–674, 1998.

[45] Maria Charina, Costanza Conti, Nicola Guglielmi, and Vladimir Protasov. Regularity of non-stationary subdivision: a matrix approach. *Numerische mathematik*, 135(3):639–678, 2017.

[46] Nira Dyn and David Levin. Analysis of hermite-interpolatory subdivision schemes. *Spline functions and the theory of wavelets*, 18:105–113, 1999.

[47] Ingrid Daubechies and Jeffrey C Lagarias. Sets of matrices all infinite products of which converge. *Linear algebra and its applications*, 161:227–263, 1992.

[48] Marc A Berger and Yang Wang. Bounded semigroups of matrices. *Linear Algebra and its Applications*, 166:21–27, 1992.

[49] Ludwig Elsner and Shmuel Friedland. Norm conditions for convergence of infinite products. *Linear algebra and its applications*, 250:133–142, 1997.

[50] Wolf-Jurgen Beyn and Ludwig Elsner. Infinite products and paracontracting matrices. *The Electronic Journal of Linear Algebra*, 2:1–8, 1997.

[51] Hartmut Prautzsch and Charles A Micchelli. Computing curves invariant under halving. *Computer Aided Geometric Design*, 4(1-2):133–140, 1987.

[52] A Micchelli and Hartmut Prautzsch. Refinement and subdivision for spaces of integer translates of a compactly supported function. In *Numerical analysis 1987*, pages 192–222. 1989.

[53] Charles A Micchelli and Hartmut Prautzsch. Uniform refinement of curves. *Linear Algebra and its Applications*, 114:841–870, 1989.

[54] Ingrid Daubechies and Jeffrey C Lagarias. Two-scale difference equations. I. existence and global regularity of solutions. *SIAM Journal on Mathematical Analysis*, 22(5):1388–1410, 1991.

[55] Ingrid Daubechies and Jeffrey C Lagarias. Two-scale difference equations II. local regularity, infinite products of matrices and fractals. *SIAM Journal on Mathematical Analysis*, 23(4):1031–1079, 1992.

[56] Gian-Carlo Rota and Gilbert Strang. A note on the joint spectral radius. *Indag. Math*, 22(4):379–381, 1960.

[57] Ingrid Daubechies and Jeffrey C Lagarias. Corrigendum/addendum to: Sets of matrices all infinite products of which converge. *Linear Algebra and its Applications*, 327(1-3):69–83, 2001.

[58] Nicholas J Higham. *Functions of matrices: theory and computation*. SIAM, 2008.

[59] G.B. Folland and G.B.A. FOLLAND. *Real Analysis: Modern Techniques and Their Applications*. A Wiley-Interscience publication. Wiley, 1999.

[60] Vesselin Vatchev. On approximation of smooth functions from null spaces of optimal linear differential operators with constant coefficients. *Analysis in Theory and Applications*, 27(2):187–200, 2011.

[61] W Gautschi. On the inverses of vandermonde and confluent vandermonde matrices. i, ii. *Numer. Math*, 4:117–123, 1962.

[62] Nathaniel Macon and Abraham Spitzbart. Inverses of vandermonde matrices. *The American Mathematical Monthly*, 65(2):95–100, 1958.

[63] Philippe G Ciarlet. *The finite element method for elliptic problems*, volume 40. Siam, 2002.

[64] N.I. Achieser. *Theory of Approximation*. Dover Books on Mathematics. Dover Publications, 2013.