

Convergence Rate of Fourier Neural Networks

by

Abylay Zhumekenov

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Master of Science in Applied Mathematics

at the

NAZARBAYEV UNIVERSITY

April 2019

© Nazarbayev University 2019. All rights reserved.

Author
Department of Mathematics
April 26, 2019

Certified by.....
Zhenisbek Assylbekov
Assistant Professor
Thesis Supervisor

Accepted by.....
Vassilios D. Tourassis
Dean, School of Science and Technology

Convergence Rate of Fourier Neural Networks

by

Abylay Zhumekenov

Submitted to the Department of Mathematics
on April 26, 2019, in partial fulfillment of the
requirements for the degree of
Master of Science in Applied Mathematics

Abstract

The paper investigates a convergence rate for 2-layer feedforward Fourier Neural Network (FNN). Such networks are motivated by the approximation properties of well-known Fourier series. Several implementations of FNNs were proposed since 1990's: by Gallant and White; A. Silvescu; Tan, Zuo and Cai; Liu. The main focus of this research is Silvescu's FNN, because such activation function does not fit into the category of networks, where the linearly transformed input is exposed to activation. The latter ones were extensively described by Hornik in 1989. In regard to non-trivial Silvescu's FNN, its convergence rate is proven to be of order $O(1/n)$. The paper continues investigating classes of functions approximated by Silvescu FNN, which appeared to be from Schwartz space and space of positive definite functions.

Thesis Supervisor: Zhenisbek Assylbekov

Title: Assistant Professor

Acknowledgments

In this section I mention some important people who contributed to this research during preparatory stages. I would like to express my sincere gratitude to my supervisor Dr. Assylbekov, who guided me through the whole research process, for his kind responsiveness and patience. I would like to give special thanks to my co-supervisors Dr. Takhanov and Dr. Castilla, for helping me with mathematical proofs and with choosing the literature. I also wish to thank Dr. Tulenbayev from Mathematics department of Suleyman Demirel University for agreeing to be a thesis committee member and to review the paper, and Malika Uteuliyeva, now a third year undergraduate student at Nazarbayev University, who helped with Python implementation of Silvescu Fourier Neural Network.

This doctoral thesis has been examined by a Committee of the
Department of Chemistry as follows:

Professor Jianshu Cao
Chairman, Thesis Committee
Professor of Chemistry

Professor Troy Van Voorhis
Thesis Supervisor
Associate Professor of Chemistry

Professor Robert W. Field
Member, Thesis Committee
Haslam and Dewey Professor of Chemistry

Contents

1	Overview	8
1.1	Introduction	8
1.2	General model	9
1.3	Existing implementations	11
1.3.1	Gallant and White FNN	11
1.3.2	Silvescu FNN	11
1.3.3	Tan, Zuo and Cai FNN	12
1.3.4	Liu FNN	12
1.4	Discussion	13
2	Main results	16
2.1	Proposition	16
2.2	Proof	17
2.3	Functions in Γ_C	22
2.3.1	Functions with absolutely integrable Fourier transform	22
2.3.2	Properties of functions in Γ_C	24
2.3.3	Examples of functions in Γ_C	26
3	Experiments	29
3.1	Generating data for functions in Γ_C	29
3.2	Software	30
3.3	Hardware	31

4 Conclusion	32
4.1 Theoretical results	32
4.2 Experimental results	33
4.3 Future considerations	33
A Figures	34

List of Figures

A-1	Gaussian function	34
A-2	Translated and scaled Gaussian	35
A-3	Linearly transformed (in time) Gaussian	35
A-4	Linear combination of Gaussians	36
A-5	Composition of Gaussian with polynomials	36
A-6	Composition of Gaussian with cosine	37
A-7	Ridge function	37
A-8	Multivariate Gaussian distribution	38
A-9	Gaussian mixture model	38
A-10	Silvescu activation function	39

Chapter 1

Overview

1.1 Introduction

Artificial neural networks have been widely used in machine learning and acquired their popularity in the 1990's. The second burst occurred in recent years due to a rapid increase of processable data (“Big-Data”) and availability of powerful computing machinery. Modern trends in deep neural networks helped to achieve superior results in pattern recognition, text processing and other fields of machine learning. However, this paper is focused on the “shallow” 2-layer neural nets - a classic case for multilayer perceptrons. Specifically, the study assesses the convergence rate for one of 2-layer neural networks with activators composed of a product of cosine functions with different frequencies. The architecture was proposed by A. Silvescu in 1999 [1]. Since this idea is in that same spirit of Fourier series and Fourier transform, such networks are referred to as “Fourier Neural Networks” (FNNs).

While most of the theory behind neural networks with standard activation functions is well-established, the peculiar ones like above leave a room for research. For this particular FNN, we derive an upper bound for the approximation error equal to $C^2 4^{d-1}/n$, where d, n, C are, respectively, dimension size, number of neurons, and a constant dependent on the function. The proof of this assertion uses a technique similar to the one implemented in [2]. The paper proceeds with identifying classes of functions, for which the approximation error bound can be computed.

Besides theoretical findings, the paper attempts to confirm them with computational results. The experiments include implementation of Fourier Neural Network in Python’s TensorFlow library, generating synthetic data according to classes of functions obtained previously, and then training the model and assessing the approximation error.

1.2 General model

Before talking about FNN implementations, we shall give a context of feedforward neural nets in general. The name “feedforward” is derived from the concept of passing weighted value of the input vector forward, to the hidden layer. It is then subject to a non-linear transformation called “activation”. This helps to catch non-linear behavior of the input data. After this, the obtained value is passed further, to the output layer. Mathematically, the general model can be given as a mapping

$$x \rightarrow v_0 + \sum_{k=1}^n v_k \sigma(w_k \cdot x + b_k), \quad (1.1)$$

where $x \in \mathbb{R}^d$ represents a multidimensional input, $w_k \in \mathbb{R}^d$ is a weight vector, $b_k \in \mathbb{R}$ is a bias, σ is called “activation function” and $v_k \in \mathbb{R}$ is a weight for the resulting nonlinear output.

Such model falls in the paradigm “weighted sum of input characteristics”. The standard choices for the activation function are “sigmoidal” functions (sometimes referred as “squashers”), which have bounded output. For example, the logistic activation function has the following expression

$$\sigma(z) = \frac{1}{1 + e^{-z}}, \quad (1.2)$$

where $z = w_k \cdot x + b_k$, and maps the input from \mathbb{R}^d to the interval $(0, 1)$. The hyperbolic

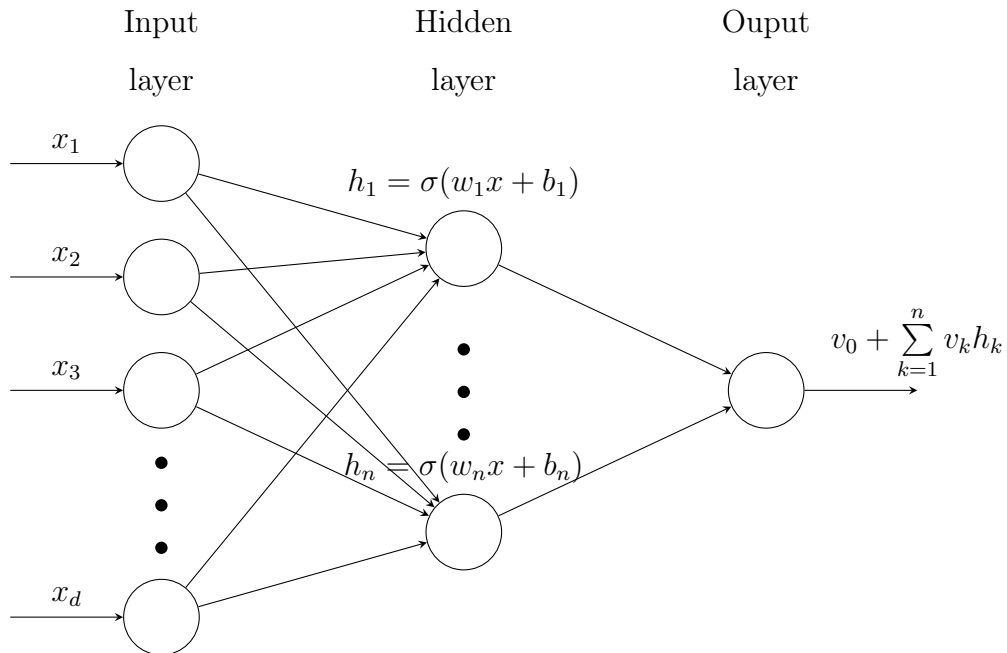
tangent function is given by

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}. \quad (1.3)$$

In general, most of them have the shape similar to a heaviside step function

$$H(z) = \begin{cases} 0, & z < 0, \\ 1, & z \geq 0. \end{cases} \quad (1.4)$$

The values of these activations are stored in the so-called “hidden neurons”, a transitional state between input and output. There can be several such neurons placed in parallel in a feedforward network, composing a hidden layer. The anatomy of the network would look like the following:



Having such activation and enough neurons, 2-layer neural networks can approximate almost any continuous function [3] [4]. The studies show that the accuracy of approximation increases with the number of hidden layers [5] and with the number of hidden neurons in general [4].

1.3 Existing implementations

Fourier inspired neural networks in some way resemble Fourier Series or Fourier Transform. For this reason, the activation used in these nets contains cosine transformations. Several implementations with different activation functions have been proposed starting from 1990's.

1.3.1 Gallant and White FNN

One of the earliest examples is the network of Gallant and White (1988) [6]. The suggested model uses a “cosine squasher” instead of a standard sigmoid activation. The function is monotone and bounded.

$$\sigma_{GW}(z) = \begin{cases} 0, & -\infty < z < \frac{\pi}{2}, \\ \frac{1}{2} [\cos(z + \frac{3\pi}{2}) + 1], & -\frac{\pi}{2} \leq z \leq \frac{\pi}{2}, \\ 1, & \frac{\pi}{2} < z < \infty. \end{cases} \quad (1.5)$$

The main characteristic of the activation is that it cuts off frequencies higher than $\frac{\pi}{2}$. The network then represents a mapping

$$x \rightarrow v_0 + \sum_{k=1}^n v_k \sigma_{GW}(w_k \cdot x + b_k), \quad (1.6)$$

with n being the number of hidden layer units. According to Gallant and White, the network possesses a Fourier Series approximation. That is, if the periodic function to be approximated is in L_2 , the net converges in L_2 sense; and if it is continuous, the net converges uniformly.

1.3.2 Silvescu FNN

Silvescu proposed another implementation of FNNs in 1999 [1]. This one has the most exotic activation function, as the result of non-standard way of discretization of

the problem. The activation is as follows

$$\sigma_{\text{Silv}}(x; w_k, b_k) = \prod_{i=1}^d \cos(w_{ki}x_i + b_{ki}), \quad (1.7)$$

where d is the dimension size of an input. The network formula is then

$$x \rightarrow \sum_{k=1}^n v_k \sigma_{\text{Silv}}(x; w_k, b_k). \quad (1.8)$$

Instead of nonlinear transformation of weighted sum of inputs, here we take product of nonlinear filter of each weighted input. This makes Silvescu's implementation the most interesting and the most challenging to analyze among other FNNs. Therefore, the focus of the paper is the convergence of this particular FNN.

1.3.3 Tan, Zuo and Cai FNN

The next type of FNN was proposed by several authors, by H. S. Tan in 2006 [7], by W. Zuo and L. Cai in 2008 [8]. Generally, they can be written as

$$x \rightarrow v_0 + \sum_{k=1}^n v_k \cos(w_k \cdot x + b_k) + u_k \sin(w_k \cdot x + b_k). \quad (1.9)$$

Note that we have the same frequency argument in both sine and cosine.

1.3.4 Liu FNN

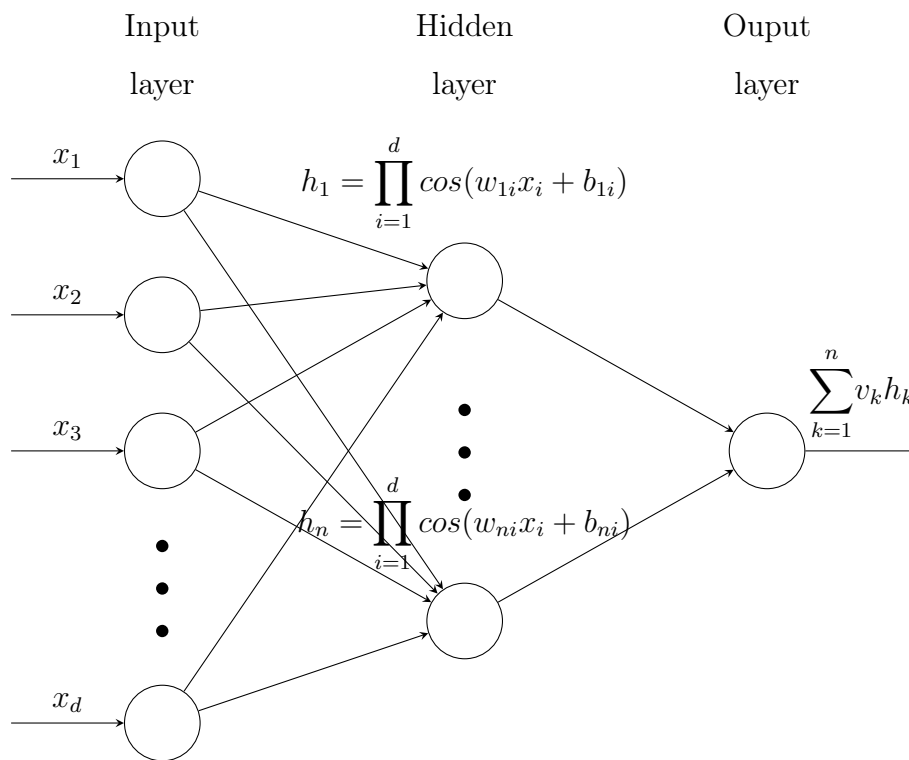
A little different approach is to set these parameters independent of each other. Written by S. Liu in 2013 [9], the paper suggests the model

$$x \rightarrow v_0 + \sum_{k=1}^n v_k \cos(w_k \cdot x + b_k) + u_k \sin(s_k \cdot x + t_k). \quad (1.10)$$

These two implementations resemble Fourier Series, but with the exception that the coefficients are not fixed, as it is the case with Fourier.

1.4 Discussion

Despite having several FNN implementations, this study is concerned with Silvescu's FNN. It does not fit into the standard way of constructing 2-layer feedforward neural networks, and appears to have the least trivial proof for convergence rate. As one might have noticed, the rest of FNNs (including standard neural nets) can be rewritten in the form $x \rightarrow v_0 + \sum v_k \sigma(w_k \cdot x + b_k)$. All such neural networks have already been studied by Hornik [10] in 1989, and it has been shown that the convergence occurs at a rate of $O(1/n)$. Silvescu's activation function does not take an affine transformation of input $w_k \cdot x + b_k$ as its argument. Instead, it is nonlinear in each spatial component of an input, $\sigma_{\text{Silv}} = \prod \cos(w_{ki}x_i + b_{ki})$ and graphically it looks as follows:



Since such networks fall out of the paradigm of classic neural nets, little is known about their convergence properties. Our research, therefore, attempts to fill some of the gaps in neural network approximation theory. In particular, we show that there

exists a network of the form

$$f_n(x) = \sum_{k=1}^n c_k \sigma_{\text{Silv}}(x; a_k, b_k) \quad (1.11)$$

which has the error bounded as

$$\int_B \left| f(x) - f_n(x) \right|^2 \mu(dx) \leq \frac{C^2 4^{d-1}}{n}. \quad (1.12)$$

That is, the order of convergence for Silvescu FNN is $O(1/n)$, given fixed dimension size d and a bound C for Fourier transform. The convergence rate for all mentioned FNN implementations is then expected to be of order $O(1/n)$.

A very important method to derive the convergence rate is used by Barron in his investigation of sigmoidal activation functions [2]. The Gallant and White FNN falls under the scope of this paper. The study proves that for functions with existing Fourier integral $f(x) = \int e^{iw \cdot x} \tilde{f}(w) dw$ and finite first moment $\int |w| |\tilde{f}(w)| dw < C$, 2-layer neural networks with sigmoidal activation have an error bounded from above by c'/n , which is $O(1/n)$ (C and c' are constants). Indeed, the way σ_{GW} was defined lets us perceive it as a sigmoidal function, in which case we can apply Barron's results directly.

In the course of our paper, we utilize the main idea of Barron's error estimation and derive our own proof for Silvescu FNN. The main difference between these two results is in the assumptions on the moments of Fourier transform of the function. While Barron sets the first moment $\int |w| |\tilde{f}(w)| dw < C$, we choose functions with finite zeroth moment $\int |\tilde{f}(w)| dw < C$. This is, actually, equivalent to demanding Fourier integral to be in $L_1(\mathbb{R}^d)$.

Such assumption was also used by Jones [11] to show convergence rate of order $O(1/n)$ for networks with 'affine transformation of input' mentioned above. FNNs that were used in works of Tan, Zuo and Cai, Liu come from this category. Proper modifications to Barron's proof would also confirm the results of Jones and Hornik [11] [10]. The consequent analysis of the assumption helped to identify suitable classes

of functions for which the error bound $\frac{C^2 4^{d-1}}{n}$ holds. These were the classes of positive definite functions and the class of Schwartz functions. The most famous functions common to both classes are Gaussians and their modifications. The experiments for such examples of studied functions have shown controversial results. Although the functions can be approximated within the indicated upper error bound, the error stops at a certain level. Possible reason for such behavior might be an over-parametrization. However, for smaller number of neurons, the dependence of error on the hidden layer size seemed to prove experimentally the convergence rate of $O(1/n)$.

Chapter 2

Main results

2.1 Proposition

Let us consider functions on \mathbb{R}^d for which the Fourier representation has the form

$$f(x) = \int_{\mathbb{R}^d} e^{iw \cdot x} \tilde{F}(dw), \quad (2.1)$$

where $\tilde{F}(dw) = e^{i\theta(w)}F(dw)$ is a unique complex-valued measure (Fourier distribution), $F(dw)$ is the magnitude distribution and $\theta(w)$ is the phase. Define the class of functions Γ which can be represented in the form above for some $\tilde{F}(dw)$ and finite $\int F(dw)$.

$$\Gamma = \left\{ f : \mathbb{R}^d \rightarrow \mathbb{R} \quad \mid \quad \int_{\mathbb{R}^d} F(dw) < \infty \right\}.$$

Then, for each $C > 0$ we can define the classes Γ_C

$$\Gamma_C = \left\{ f : \mathbb{R}^d \rightarrow \mathbb{R} \quad \mid \quad C_f = \int_{\mathbb{R}^d} F(dw) < C \right\}.$$

Now consider a bounded set $B \subset \mathbb{R}^d$. Then define Γ_B for which the representation in Equation (2.1) holds for $x \in B$ for some $\tilde{F}(dw)$ with finite magnitude measure.

$$\Gamma_B = \left\{ f : B \rightarrow \mathbb{R} \quad \mid \quad \int_{\mathbb{R}^d} F(dw) < \infty \right\}.$$

Finally, for each $C > 0$, let

$$\Gamma_{C,B} = \left\{ f : B \rightarrow \mathbb{R} \mid C_{f,B} = \int_{\mathbb{R}^d} F(dw) \leq C \right\}.$$

Theorem 1. *Let*

$$\phi = \sigma_{\text{Silv}}(x; a_k, b_k) = \prod_{j=1}^d \cos(a_{kj}x_j + b_{kj})$$

for $a_{kj}, b_{kj} \in \mathbb{R}$ and $x, a_k, b_k \in \mathbb{R}^d$. For every function $f \in \Gamma_{B,C}$ and any probability measure μ , there exists $f_n(x)$, $n \geq 1$, of the form

$$f_n(x) = \sum_{k=1}^n c_k \sigma_{\text{Silv}}(x; a_k, b_k) \tag{2.2}$$

such that

$$\int_B \left| f(x) - f_n(x) \right|^2 \mu(dx) \leq \frac{C^2 4^{d-1}}{n}. \tag{2.3}$$

2.2 Proof

The proof of Theorem 1 uses several lemmas, which we specify below.

Lemma 1. *If f is in the closure of the convex hull of a set G in a Hilbert space, with $\|g\| \leq b$ for each $g \in G$, then for every $n \geq 1$, and every $c' > b^2 - \|f\|^2$, there exists f_n in the convex hull of n points in G such that $\|f - f_n\|^2 \leq \frac{c'}{n}$.*

Proof. The proof of this lemma can be found in Barron's paper [2]. □

Lemma 2. *For every function $f \in \Gamma_{C,B}$ and every function $g \in G_{\text{cos}}$, the function f is in the closure of the convex hull of G_{cos} , where the closure is taken in $L_2(\mu, B)$ and*

$$G_{\text{cos}} = \{ \gamma \cos(w \cdot x + b) : |\gamma| \leq C \}.$$

Proof. Let $\tilde{F}(dw) = e^{i\theta(w)} F(dw)$ be the magnitude and phase decomposition in the Fourier representation of the function f on B for which $\int F(dw) \leq C$, where $w \in \mathbb{R}^d$. Since the function f that we are considering is a real-valued function, from eq. (2.1),

we have for $x \in B$

$$\begin{aligned}
f(x) &= \operatorname{Re} \int_{\mathbb{R}^d} e^{iw \cdot x} \tilde{F}(dw) \\
&= \operatorname{Re} \int_{\mathbb{R}^d} e^{iw \cdot x} e^{i\theta(w)} F(dw) \\
&= \operatorname{Re} \int_{\mathbb{R}^d} e^{iw \cdot x + i\theta(w)} F(dw) \\
&= \int_{\mathbb{R}^d} \cos(w \cdot x + \theta(w)) F(dw) \\
&= \int_{\mathbb{R}^d} C_{f,B} \cos(w \cdot x + \theta(w)) \frac{F(dw)}{C_{f,B}} \\
&= \int_{\mathbb{R}^d} g(x, w) \Lambda(dw),
\end{aligned} \tag{2.4}$$

where $C_{f,B} = \int F(dw) \leq C$ is our assumption on functions from $\Gamma_{B,C}$; $g(x, w) = C_{f,B} \cos(w \cdot x + \theta(w))$; and $\Lambda(dw) = F(dw)/C_{f,B}$ is a probability measure, since

$$\int_{\mathbb{R}^d} \Lambda(dw) = \int_{\mathbb{R}^d} \frac{F(dw)}{C_{f,B}} = \frac{1}{C_{f,B}} \int_{\mathbb{R}^d} F(dw) = 1.$$

From eq. (2.4), it is known that $f(x) = \mathbb{E}_w [g(x, w)]$. In addition, $|g(x, w)| \leq |C_{f,B}| \leq C$. Most importantly, last row of Equation (2.4) represents $f(x)$ as an infinite convex combination of functions in the set

$$G_{\cos} = \{\gamma \cos(w \cdot x + b) : |\gamma| \leq C\}.$$

To show this, let w_1, w_2, \dots, w_n be drawn independently from the same distribution Λ . Consider now $g(x, w_1), g(x, w_2), \dots, g(x, w_n)$. Then the sample mean $f_n(x; w_1, \dots, w_n) = \frac{1}{n} \sum_{i=1}^n g(x, w_i)$ has expected value equal to $f(x)$, that is $\mathbb{E}_w [f_n(x)] = f(x)$, and at the same time it is a convex combination of functions from G_{\cos} . We will get results

if we apply Fubini's Theorem to the expected value of squared $L_2(\mu, B)$ norm.

$$\begin{aligned}
\mathbb{E}_w \|f - f_n\|_{L_2(\mu, B)}^2 &= \mathbb{E}_w \left(\int_B |f - f_n|^2 \mu(dx) \right) \\
&= \int_B (\mathbb{E}_w [\mathbb{E}_w [f_n] - f_n]^2) \mu(dx) \\
&= \int_B \text{Var} [f_n] \mu(dx) \\
&= \int_B \frac{1}{n} \text{Var} [g] \mu(dx) \\
&\leq \frac{1}{n} \int_B C^2 \mu(dx) \\
&= \frac{C^2}{n}.
\end{aligned}$$

The $L_2(\mu, B)$ norm above converges to zero in mean. By taking into account that the norm is positive, there is a sequence w_1, \dots, w_n for which the norm converges to zero. Thus, there exists the convex combination $f_n(x; w_1, \dots, w_n)$ of functions from G_{\cos} that converges to f in $L_2(\mu, B)$. Therefore, f is in the closure of the convex hull of G_{\cos} in $L_2(\mu, B)$. \square

Lemma 3. *Functions in G_{\cos} are in the convex hull of functions in G_{Silv} , that is $G_{\cos} \subset \text{co}G_{Silv}$, where*

$$G_{Silv} = \left\{ 2^{d-1} \gamma \prod_{j=1}^d \cos(a_{kj} x_j + b_{kj}) : |\gamma| \leq C \right\}$$

for $w, x \in \mathbb{R}^d$; $b_{kj} \in \mathbb{R}$ and $|\gamma| = |C_{f,B}| = \left| \int F(dw) \right| \leq C$.

Proof. The proof is by induction. First, we consider the base case $d = 1$. Each function $g \in G_{\cos}$ is a single variable function. Then we have

$$\begin{aligned}
g(x) &= \gamma \cos(w_1 x_1 + b) \\
&= \sum_{k=1}^{2^0} \frac{1}{2^0} \left[2^0 \gamma \prod_{j=1}^1 \cos(a_{kj} x_j + b_{kj}) \right] \\
&= \sum_{k=1}^{2^{1-1}} \frac{1}{2^{1-1}} \left[2^{1-1} \gamma \prod_{j=1}^1 \cos(a_{kj} x_j + b_{kj}) \right],
\end{aligned}$$

which is a convex combination of 2^{1-1} functions in the class G_{Silv} with $a_{kj} = w_1$. Thus, when the dimension is $d = 1$, $G_{\text{cos}} \subset \text{co}G_{\text{Silv}}$.

Next, let us prove the general case for $d = m + 1$. Assume $G_{\text{cos}} \subset \text{co}G_{\text{Silv}}$ for $d = m$. Then

$$\begin{aligned} g(x) &= \gamma \cos(w \cdot x + b) \\ &= \sum_{k=1}^{2^{m-1}} \frac{1}{2^{m-1}} \left[2^{m-1} \gamma \prod_{j=1}^m \cos(a_{kj}x_j + b_{kj}) \right], \end{aligned}$$

where $x, w \in \mathbb{R}^m$ and $a_{kj}, b, b_{kj} \in \mathbb{R}$.

Now consider $\tilde{g}(\tilde{x}) \in \tilde{G}_{\text{cos}}$, where $\tilde{x} \in \mathbb{R}^{m+1}$ and \tilde{g} is a function on \mathbb{R}^{m+1} . We also have $\tilde{w} \in \mathbb{R}^{m+1}$. Actually, we can write $\tilde{x} = (x_1, x_2, \dots, x_m, x_{m+1})$, where the first m coordinates are from x . So,

$$\begin{aligned} \tilde{g}(\tilde{x}) &= \gamma \cos(\tilde{w} \cdot \tilde{x} + b) \\ &= \gamma \cos(w_1x_1 + \dots + w_mx_m + w_{m+1}x_{m+1} + b) \\ &= \gamma \cos(w_1x_1 + \dots + \hat{w}_m\hat{x}_m + b) \end{aligned}$$

Where we define $\hat{w}_m\hat{x}_m := w_mx_m + w_{m+1}x_{m+1}$. Now we can apply results for the m -dimensional case to the identity above.

$$\begin{aligned} \tilde{g}(\tilde{x}) &= \sum_{k=1}^{2^{m-1}} \frac{1}{2^{m-1}} \left[2^{m-1} \gamma \left(\prod_{j=1}^{m-1} \cos(a_{kj}x_j + b_{kj}) \right) \left(\cos(\tilde{w}_m\tilde{x}_m + b_{km}) \right) \right] \\ &= \sum_{k=1}^{2^{m-1}} \left[\gamma \left(\prod_{j=1}^{m-1} \cos(a_{kj}x_j + b_{kj}) \right) \left(\cos(w_mx_m + w_{m+1}x_{m+1} + b_{km}) \right) \right] \\ &= \sum_{k=1}^{2^{m-1}} \left[\gamma \left(\prod_{j=1}^{m-1} \cos(a_{kj}x_j + b_{kj}) \right) \left(\cos((w_mx_m + b_{km}) + (w_{m+1}x_{m+1})) \right) \right] \\ &= \sum_{k=1}^{2^{m-1}} \gamma \left(\prod_{j=1}^{m-1} \cos(a_{kj}x_j + b_{kj}) \right) \times \\ &\quad \times \left(\cos(w_mx_m + b_{km}) \cos(w_{m+1}x_{m+1}) - \sin(w_mx_m + b_{km}) \sin(w_{m+1}x_{m+1}) \right) \end{aligned}$$

$$\begin{aligned}
&= \sum_{k=1}^{2^{m-1}} \gamma \left(\prod_{j=1}^{m-1} \cos(a_{kj}x_j + b_{kj}) \right) \times \\
&\times \left(\cos(w_m x_m + b_{km}) \cos(w_{m+1} x_{m+1}) + \cos(w_m x_m + b_{km} - \frac{\pi}{2}) \cos(w_{m+1} x_{m+1} + \frac{\pi}{2}) \right) \\
&= \sum_{k=1}^{2^{m-1}} \gamma \left[\prod_{j=1}^{m+1} \cos(a_{kj}x_j + b_{kj}) + \prod_{i=1}^{m+1} \cos(a_{ki}x_i + b_{ki}) \right] \\
&= \sum_{k=1}^{2^m} \gamma \left[\prod_{j=1}^{m+1} \cos(a_{kj}x_j + b_{kj}) \right] \\
&= \sum_{k=1}^{2^m} \frac{1}{2^m} \left[2^m \gamma \prod_{j=1}^{m+1} \cos(a_{kj}x_j + b_{kj}) \right].
\end{aligned}$$

The function above is a convex combination of 2^m functions from G_{Silv} . Then, $G_{\text{cos}} \subset \text{co}G_{\text{Silv}}$ when $d = m + 1$, and therefore, by induction, for any dimension d . Note that expanding the cosine resulted in two cosine products which have same frequencies. This suggests that the a_{kj} 's are equal along the k -index. \square

Lemma 4. *For every function $f \in \Gamma_{C,B}$ and every function $g \in G_{\text{Silv}}$, the function f is in the closure of the convex hull of G_{Silv} , where the closure is taken in $L_2(\mu, B)$.*

Proof. We will use Lemma 2 and Lemma 3 to show that f is in the closure of the convex hull of G_{Silv} . From Lemma 2 it follows that $f \in \Gamma_{C,B} \subset \bar{\text{co}}G_{\text{cos}}$, while from Lemma 3 we obtain $G_{\text{cos}} \subset \text{co}G_{\text{Silv}}$. Convex hull of G_{cos} is also a subset of G_{Silv} , that is $\text{co}G_{\text{cos}} \subset \text{co}G_{\text{Silv}}$. By taking limits, one can show that $\bar{\text{co}}G_{\text{cos}} \subset \bar{\text{co}}G_{\text{Silv}}$ also holds. Thus, $f \in \bar{\text{co}}G_{\text{Silv}}$. \square

Proof of Theorem 1. Two important results that we will utilize are Lemma 1 and Lemma 4. From Lemma 4, we have $f(x) \in \bar{\text{co}}G_{\text{Silv}}$, where

$$G_{\text{Silv}} = \left\{ 2^{d-1} \gamma \prod_{j=1}^d \cos(a_{kj}x_j + b_{kj}) : |\gamma| \leq C \right\}.$$

Functions in this class are measurable and bounded by $g(x) \leq |2^{d-1}\gamma| \leq 2^{d-1}C$. Then

the $L_2(\mu, B)$ norm is

$$\begin{aligned}
\|g(x)\|_{L_2(\mu, B)}^2 &= \int_B |g(x)|^2 \mu(dx) \\
&\leq \mu(B) \sup_{x \in B} |g(x)|^2 \\
&\leq \mu(B) \sup_{g \in G_{\text{Silv}}} \sup_{x \in B} |g(x)|^2 \\
&\leq \sup_{g \in G_{\text{Silv}}} \sup_{x \in B} |g(x)|^2 \\
&\leq |2^{d-1}\gamma|^2 \\
&\leq 4^{d-1}C^2 = b^2.
\end{aligned}$$

Then, by Lemma 1, there exists f_n in the convex hull of n points of G_{Silv} such that $\|f - f_n\| \leq c'/n$ for every $n \geq 1$ and every $c' > b^2 - \|f\|^2$. Let us choose $c' = b^2$. Then the desired bound holds

$$\|f - f_n\| \leq \frac{c'}{n},$$

where

$$f_n(x) = \sum_{k=1}^n t_k \left(2^{d-1}\gamma \prod_{j=1}^d \cos(a_{kj}x_j + b_{kj}) \right) = \sum_{k=1}^n c_k \phi(x; a_k, b_k),$$

for $t_k > 0$ and $\sum t_k = 1$. We can restrict coefficients in our network to satisfy the bound $\sum |c_k| \leq 2^{d-1}C$. \square

2.3 Functions in Γ_C

2.3.1 Functions with absolutely integrable Fourier transform

The goal is to identify functions in the class Γ_C (or $\Gamma_{C,B}$), i.e. for which $\int F(dw) < C$ holds. This condition is equivalent to demanding Fourier transform to be in L_1 , that is $\int |\tilde{F}(dw)| < C$ (or alternatively, $\int |\tilde{f}(w)|dw < C$). We also give the formula to compute the constant C , which we will later use to compare the theoretical error bound with the experimental findings.

There are two main classes of functions which satisfy the mentioned conditions

for the transform:

1. Positive definite functions
2. Schwartz functions

First, we consider positive definite functions. In the definition, these are very similar to positive definite kernels, which are important tools in statistical learning theory. Let f be a positive definite function, which requires $\sum x_i x_j f(x_i - x_j)$ to be non-negative for any choice of x_i, x_j [12]. Alternatively, the matrix A , whose elements are $a_{i,j} = f(x_i - x_j)$, has to be positive semi-definite [2]. Then, if the function is continuous on \mathbb{R}^d , by Bochner's theorem [13, p. 144], it can be represented as $f(x) = \int e^{iw \cdot x} F(dw)$. Here, $F(dw)$ is a positive measure and corresponds to $|\tilde{F}(dw)|$. Then the constant in the error bound can be computed as follows

$$C = \int_{\mathbb{R}^d} F(dw) = \int_{\mathbb{R}^d} |\tilde{f}(w)| dw = \int_{\mathbb{R}^d} \tilde{f}(w) dw = f(0). \quad (2.5)$$

We can observe that for positive definite functions, the L_1 -norm of its Fourier transform is defined by the behavior of f at the origin. That is, one does not even need to know the transform in order to compute the constant C .

There exists another class of functions with absolutely integrable Fourier transform - Schwartz space $\mathcal{S}(\mathbb{R}^d)$. This is a space of rapidly decreasing functions on \mathbb{R}^d (functions with derivatives decreasing faster than the inverse of any polynomial). The Schwartz space is widely used in Fourier analysis and applications can be found in the field of partial differential equations. The definition is given as follows [14, p. 134]:

$$\mathcal{S}(\mathbb{R}^d) = \{f \in C^\infty(\mathbb{R}^d) : \|f\|_{\alpha,\beta} < \infty \quad \forall \alpha, \beta \in \mathbb{N}^d\} \quad (2.6)$$

where α, β are multi-indices, $C^\infty(\mathbb{R}^d)$ is the space of infinitely differentiable functions and

$$\|f\|_{\alpha,\beta} = \sup_{x \in \mathbb{R}^d} |x^\alpha D^\beta f(x)|. \quad (2.7)$$

It is known that Fourier transform maps functions from Schwartz space onto itself,

that is, if $f \in \mathcal{S}(\mathbb{R}^d)$, then $\tilde{f} \in \mathcal{S}(\mathbb{R}^d)$ [15, p. 331]. Since we also know that $\mathcal{S}(\mathbb{R}^d) \subset L_p(\mathbb{R}^d)$ [16], the transform then satisfies $\tilde{f} \in L_1(\mathbb{R}^d)$.

2.3.2 Properties of functions in Γ_C

Of course, functions obtained from Schwartz space by translation (shift), dilation (scaling), summation and consequently by a linear combination are also in Schwartz space. This can be seen straight from the definition. From [14, p. 142], product and convolution of Schwartz functions are also in the same space. This is also true for some of such cases when we pick functions from Γ_C . To show this, let f be in the class Γ_C . Functions obtained by applying the following operations are also in Γ (particularly in $\Gamma_{C'}$, where C' is a different constant).

a) Translation (time shifting). If we consider $f(x + a)$, its Fourier transform can be rewritten as $\tilde{f}(x + a) = e^{-ia \cdot w} \tilde{f}(x)$ [17, p. 92]. Then, since the resulting Fourier transform is a rotation on a complex plane, this transform also lives in L_1 . Therefore, $f \in \Gamma_C$.

b) Dilation (time scaling). For $g(x) = f(ax)$, we have $\tilde{g}(w) = (1/2\pi)^d \int e^{-iw \cdot x} g(x) dx = (1/2\pi)^d \int e^{-iw \cdot x} f(ax) dx$. If we use a substitution $x = y/a$, we obtain the transform $\tilde{g}(w) = (1/2\pi)^d \int e^{-iw/a \cdot y} f(y) dy = \tilde{f}(w/a)/a$.

Now, by taking a change of variables as $u = w/a$ we get $\int |\tilde{g}(w)| dw = \int |\tilde{f}(w/a)/a| dw = \int |\tilde{f}(u)| du < C$. Thus, the constant is invariant to time scaling and $f(ax) \in \Gamma_C$.

c) Linear transformation (of time domain). Let $f \in \Gamma_C$ and A be an invertible matrix. Consider a function $g(x) = f(Ax)$. Then its Fourier transform will be given by the equation

$$\tilde{g}(w) = \frac{1}{|\det A|} \tilde{f}(A^{-T}w). \quad (2.8)$$

However, the determinant term disappears during change of variables, when we try to find a bound for L_1 -norm of \tilde{f} . The determinant appears from Jacobian and then cancels with the denominator. That means the transform $f(Ax) =$

$g(x) \in \Gamma_C$. Keep in mind that Fourier transform itself is invariant only for orthogonal transformations.

- d) Summation. Let $f_1 \in \Gamma_{C_1}$ and $f_2 \in \Gamma_{C_2}$. Define $g(x) = f_1(x) + f_2(x)$. Then by the linearity of Fourier transform, $\tilde{g}(w) = \tilde{f}_1(w) + \tilde{f}_2(w)$, and the L_1 -norm is bounded by $C_1 + C_2$. Therefore, $g(x) \in \Gamma_{C_1+C_2}$.
- e) Linear combination. If $f_i \in \Gamma_{C_i}$ and $g = \sum \beta_i f_i$, then by the linearity of Fourier transform and previous summation property, $\tilde{g}(w) = \sum \beta_i \tilde{f}_i(w)$. Consequently, $\int |\tilde{g}(w)|dw \leq \int \sum |\beta_i \tilde{f}_i(w)|dw = \sum \beta_i \left(\int |\tilde{f}_i(w)|dw \right)$. Therefore, $g(x) \in \Gamma_{\sum |\beta_i| C_i}$.
- f) Product. Let $f_1 \in \Gamma_{C_1}$ and $f_2 \in \Gamma_{C_2}$, that is $\int |\tilde{f}_1(w)|dw < C_1$ and $\int |\tilde{f}_2(w)|dw < C_2$. If we define $g(x) = f_1(x)f_2(x)$, then by the convolution theorem [17, p. 92] we obtain

$$g(x) = \int_{\mathbb{R}^d} e^{iw \cdot x} \tilde{g}(w) dw = \int_{\mathbb{R}^d} e^{iw \cdot x} \left(\int_{\mathbb{R}^d} \tilde{f}_1(z) \tilde{f}_2(w-z) dz \right) dw. \quad (2.9)$$

Then, $g(x) \in \Gamma_{C_1 C_2}$:

$$\begin{aligned} \int_{\mathbb{R}^d} |\tilde{g}(w)|dw &= \int_{\mathbb{R}^d} \left| \int_{\mathbb{R}^d} \tilde{f}_1(z) \tilde{f}_2(w-z) dz \right| dw \\ &\leq \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} |\tilde{f}_1(z)| |\tilde{f}_2(w-z)| dw dz \\ &= \int_{\mathbb{R}^d} |\tilde{f}_1(z)| \left(\int_{\mathbb{R}^d} |\tilde{f}_2(w-z)| dw \right) dz \\ &= \int_{\mathbb{R}^d} |\tilde{f}_1(z)| \left(\int_{\mathbb{R}^d} |\tilde{f}_2(w)| dw \right) dz \\ &= \left(\int_{\mathbb{R}^d} |\tilde{f}_1(z)| dz \right) \left(\int_{\mathbb{R}^d} |\tilde{f}_2(w)| dw \right) \\ &< C_1 C_2. \end{aligned} \quad (2.10)$$

- g) Composition with polynomials. We will use results for combination and product, mentioned above. Consider $f \in \Gamma_C$, then by the property of product of functions, $f^k \in \Gamma_{C^k}$. Now, let $g(z)$ be a univariate polynomial function

$g(f(x)) = \sum_{k=1}^n \beta_k (f(x))^k$. By the property of linear combination of functions, we obtain $f(z) \in \Gamma_{\sum_{k=1}^n |\beta_k| C^k}$.

- h) Composition with analytic functions. Let $f(x) \in \Gamma_C$ and $g(z)$ be an analytic function $g(z) = \sum_{k=0}^{\infty} \beta_k z^k$, for which the radius of absolute convergence is $r > C$. By similar approach as with polynomials, we get $g(z) \in \Gamma_{\sum_{k=0}^{\infty} |\beta_k| C^k}$. Here, the summation in the subscript of Γ is absolutely convergent since $C < r$.
- i) Ridge functions. Let $g(x) = f(a \cdot x)$ for some $a \in \mathbb{R}^d$ with $|a| = 1$, and $f(z)$ be a univariate function from Γ_C . Then

$$g(x) = f(a \cdot x) = \int_{\mathbb{R}} e^{ita \cdot x} \tilde{f}(t) dt. \quad (2.11)$$

This can be viewed as a Fourier transform of $g(x)$, concentrated on the set of points $w \in \mathbb{R}^d$ on a line along the direction a . Then, the constant can be taken as $C_f = C_g = \int \tilde{g}(t) dt$.

Note that sigmoidal functions of the form $f(x) = \phi(a \cdot x + b)$ do not have an integrable Fourier transform.

2.3.3 Examples of functions in Γ_C

Let us now look at some examples of functions mentioned above.

1. Gaussian function.

The simplest example would be the Gaussian function $f(x) = e^{-\|x\|^2/2}$. It is in the class of both positive definite functions and Schwartz functions, and, therefore, $f(x) \in \Gamma_C$. Its Fourier transform is $\tilde{f}(w) = (2\pi)^{-d/2} e^{-\|w\|^2/2}$ [18, p. 302]. Then, since the Fourier transform is a real and positive function of w ,

$$\begin{aligned} C &= \int_{\mathbb{R}^d} F(dw) = \int_{\mathbb{R}^d} |\tilde{f}(w)| dw = \int_{\mathbb{R}^d} \tilde{f}(w) dw = \int_{\mathbb{R}^d} (2\pi)^{-d/2} e^{-\|w\|^2/2} dw \\ &= (2\pi)^{-d/2} \int_{\mathbb{R}^d} e^{-\|w\|^2/2} dw = (2\pi)^{-d/2} (2\pi)^{d/2} = 1 \end{aligned}$$

Alternatively, we can use the fact that the function is positive definite (proving

this fact is hard in practice). Thus, we evaluate the Gaussian at the origin $C = f(0) = 1$. Shifted versions of Gaussian will also be in the class Γ_C , according to the translation. Therefore, we include functions of the form $f(x) = e^{-\|x-\mu\|^2/2}$, where $\mu \in \mathbb{R}^d$ is a shift.

2. Multivariate Gaussian distributions.

These functions arise in almost every field of probability and statistics. Multivariate Gaussians are also in the same Schwartz space: this can be shown by a change of variables (first, use shift, then a linear transformation). However, we now want to find the constant C for this function from our observations above. The multivariate Gaussian is given by the formula [19, p. 197]:

$$f(x) = \frac{1}{(2\pi)^{d/2}(\det \Sigma)^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)}, \quad (2.12)$$

where Σ is a symmetric positive definite matrix (covariance matrix), μ is shift in x and is the mean of distribution.

First, we can define a matrix $A = \Sigma^{-1/2} = Q^T D^{-1/2} Q$, so that $A^T A = \Sigma^{-1}$. The matrix Q is orthogonal, A is symmetric positive definite and invertible. The function now is of the form

$$\begin{aligned} f(x) &= \frac{1}{(2\pi)^{d/2}(\det \Sigma)^{1/2}} e^{-\frac{1}{2}(x-\mu)^T A^T A(x-\mu)} \\ &= \frac{\det A}{(2\pi)^{d/2}} e^{-\frac{1}{2}[A(x-\mu)]^T [A(x-\mu)]} \\ &= \frac{\det A}{(2\pi)^{d/2}} e^{-\frac{\|A(x-\mu)\|^2}{2}} \\ &= \frac{\det A}{(2\pi)^{d/2}} g(A(x-\mu)), \end{aligned} \quad (2.13)$$

where $g(y) = e^{-\|y\|^2/2}$, for which $\tilde{g}(u) = (2\pi)^{-d/2} e^{-\|u\|^2/2}$. Using the result for transformation in time domain and translation property, the Fourier transform

of the latter will be

$$\begin{aligned}\tilde{f}(w) &= \frac{\det A}{(2\pi)^{d/2}} \frac{1}{|\det A|} e^{i\mu \cdot w} \tilde{g}(A^{-T}w) \\ &= \frac{1}{(2\pi)^{d/2}} e^{i\mu \cdot w} \tilde{g}(A^{-T}w)\end{aligned}\tag{2.14}$$

L_1 -norm of \tilde{f} is then bounded by

$$\begin{aligned}\int_{\mathbb{R}^d} |\tilde{f}(w)| dw &\leq \frac{1}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} |\tilde{g}(A^{-T}w)| dw \\ &= \frac{\det A}{(2\pi)^{d/2}} \int_{\mathbb{R}^d} |\tilde{g}(u)| du \\ &= \frac{1}{(2\pi)^{d/2} (\det \Sigma)^{1/2}}.\end{aligned}\tag{2.15}$$

The constant then can be taken as $C = \frac{1}{(2\pi)^{d/2} (\det \Sigma)^{1/2}}$. Multivariate Gaussian distributions are then in the class $\Gamma_{1/[(2\pi)^{d/2} (\det \Sigma)^{1/2}]}$.

3. Gaussian mixture models.

Another important function in probability, statistics and machine learning. Gaussian mixture can be defined as a linear combination of different Multivariate Gaussians [20, p. 425]:

$$f(x) = \sum_{k=1}^n a_k f_k(x; \mu_k, \Sigma_k),\tag{2.16}$$

where each f_k is a multivariate Gaussian with different mean (μ_k) and covariance matrix (Σ_k), a_k 's are corresponding weights in the mixture for which $\sum a_k = 1$ and $a_k > 0$. Using the linear combination property, we can verify that the constant $C = \sum a_k C_i$, where $C_i = 1/[(2\pi)^{d/2} (\det \Sigma)^{1/2}]$.

Many other functions can be derived from the Gaussian, using the properties we have obtained. We will use some of them in the next chapter to assess the approximability of functions in Γ by Silvescu Fourier Neural Network that we have discussed before.

Chapter 3

Experiments

3.1 Generating data for functions in Γ_C

For the experimental part, we fix the dimension size $d = 3$. The reason is due to enormous error bound that we obtain by setting dimension too large. However, our goal is to assess if the convergence is of order $O(1/n)$, where n is the number of hidden neurons. Therefore, we will test functions obtained in the previous chapter on different values of n . Namely, $n = \{1, 4, 16, 64, 256, 1024\}$. Regarding the classes of functions in Γ_C , the following functions were chosen:

1. Gaussian

$$f(x) = e^{-\frac{1}{2}\|x\|^2}.$$

$$C = 1.$$

2. Translated and scaled (dilated) Gaussian

$$f(x) = e^{-\frac{1}{2}\|ax+b\|^2}.$$

$$C = 1.$$

3. Gaussian, linearly transformed in time argument

$$f(x) = e^{-\frac{1}{2}\|Ax\|^2}.$$

$$C = 1.$$

4. Linear combination of Gaussians

$$f(x) = a_1 e^{-\frac{1}{2}\|x\|^2} + a_2 e^{-\frac{1}{2}\|Ax\|^2}.$$

$$C = 1.$$

5. Composition of Gaussian and a polynomial

$$f(x) = a_1[g(x)]^1 + a_2[g(x)]^2 + a_3[g(x)]^4 + a_4[g(x)]^4,$$

$$\text{where } g(x) = e^{-\frac{1}{2}\|x\|^2}. \quad C = \sum |a_k|.$$

6. Composition of Gaussian and analytic function cosine

$$f(x) = \cos(e^{-\frac{1}{2}\|x\|^2}).$$

$$C = \sum 1/(2k)! = \cosh(1).$$

7. Ridge function

$$f(x) = e^{-(a \cdot x)^2/2},$$

$$C = 1.$$

8. Multivariate Gaussian (normal) distribution

$$f(x) = \frac{1}{(2\pi)^{d/2}(\det \Sigma)^{1/2}} e^{-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)} = \mathcal{N}(\mu, \Sigma).$$

$$C = \frac{1}{(2\pi)^{d/2}(\det \Sigma)^{1/2}}.$$

9. Gaussian mixture model

$$f(x) = a_1 \mathcal{N}(\mu_1, \Sigma_1) + a_2 \mathcal{N}(\mu_1, \Sigma_1).$$

$$C = \sum a_k C_i, \text{ where } C_i = 1/[(2\pi)^{d/2}(\det \Sigma)^{1/2}].$$

10. Silvescu activation function

$$f(x) = \prod \cos(w_j x_j + b_j)$$

It was also interesting to check if the network can approximate the Silvescu activation function adequately.

3.2 Software

All parameters like constants, coefficients, mean vectors and covariance matrices were chosen randomly. Then, the datasets with training size 10,000 and test size 5,000 were generated according to given functions. A small normal error $\mathcal{N}(0, 0.03)$

was added on top. The code for this was written with help of Python’s NumPy library.

Regarding the implementation for Silvescu FNN, computations were run using TensorFlow library. The network used Adam Optimizer for solving a minimization problem, sum of squared error. This is a Stochastic Gradient Descent (SGD) algorithm with momentum and an adaptive learning rate. The momentum fastens the convergence by adding part of the previous gradient to the current one. With the momentum and adaptive learning rate combined, the algorithm is less likely to slow down when the gradient approaches zero around the minimum. The initial learning rate was set to $\alpha = 0.01$, the momentum constants were set to default $\beta_1 = 0.99$ and $\beta_2 = 0.999$.

Since the implementation uses SGD, we needed to divide the dataset into batches. For a batch size 100 and 10,000 observations in total, we had 100 batches. The network training was 50 epochs long (50 iterations over the same dataset). After each epoch (= 100 processed batches), the data was shuffled to avoid repetitions. Then, the minimum mean squared error (MSE) across the epochs were chosen. The reason why we use the minimum values for MSE is that our main theorem proved the existence of network with indicated MSE.

3.3 Hardware

The package enables to use Graphics Processing Unit (GPU) acceleration for training neural networks. This requires to have GPU’s that support Nvidia’s CUDA Toolkit for accelerated computing. For running the Python code, I used my personal laptop’s with Nvidia’s GeForce GTX 1050 Ti. The computations for Silvescu FNNs on average were about 30% slower than for standard feedforward neural nets with sigmoid activation function. This is due to a larger number of floating point operations in Silvescu’s activation ($\approx 30d$ flops) with respect to sigmoid activation ($\approx 2d + 25$ flops). It is obvious that with the increase of dimensionality of the problem, that gap is expected to rise, too.

Chapter 4

Conclusion

4.1 Theoretical results

Theoretically, we have proven that there exist such parameters of FNN, for which the network's error is bounded by a constant divided by the number of hidden neurons. However, the existence of such parameters does not imply that the bound is same for all trained models. This is the best case scenario, and obtaining such solution is not guaranteed in practice. Main experimental concern would be an algorithm for solving the minimization problem.

Nevertheless, the importance of this paper is in obtaining error bounds for networks, which fall out of the paradigm of "weighted sum of activation of linearly transformed input". All standard models were described by Hornik [10] and Cybenko [3], and Barron [2] provided lower (Kolmogorov n-width) and upper bound for approximation by sigmoidal neural networks. Silvescu FNN, on the other hand, has a non-trivial activation, which makes it harder to analyze. The constant term 4^{d-1} in the error bound appears also because of the way the activation was constructed. To counter-balance this term, the number of neurons n has to be exponential in the dimensionality d .

4.2 Experimental results

The results of the experimental part were good for small values of n (number of neurons), in the interval $16 < n < 1024$ the error leveled off, for larger values the error is expected to be above the indicated error bound. Although the theoretical results provide an upper bound for the error, there has to be also a lower bound.

According to graphs (figures may be found in the appendix), the error decay be described as linear for $n \leq 16$, that is order of $O(1/n)$, whereas on the right tail of MSEs, it is best described as $O(1)$. However, this possibly can be explained by a large number of neurons, and consequently, large number of parameters. For Silvescu NN we have $(2d + 1)n = 7n$ parameters. This potentially can cause over-parametrization during training without the regularization.

Another reason could be existence of a lower bound for an approximation by Silvescu FNN. This is also probable, because the error stops decaying around 10^{-3} , which is even smaller than the error for our generated functions ($\epsilon = 0.03$). Then, small fluctuations in the right tail can be explained by Adam Optimizer finding different local minima during training stages. Because we redefine the network each time we change the hidden layer size n .

4.3 Future considerations

First possible improvement of the work could be extending the results for linear and polynomial functions. For these two functions, functions defined on integers and function on other bounded domains, Barron used suitable spline extrapolations to satisfy his assumption $\int |w| |\tilde{f}(w)| dw < C$ [2]. This could help us considering more real-life data with such nature.

The lower bound for an approximation can be derived. Barron obtains it by considering Kolmogorov's n -width, the closest distance between the function and a collections of basis functions. More research in this direction could shed a light on the analysis of the approximation by linear spans of Silvescu functions.

Appendix A

Figures

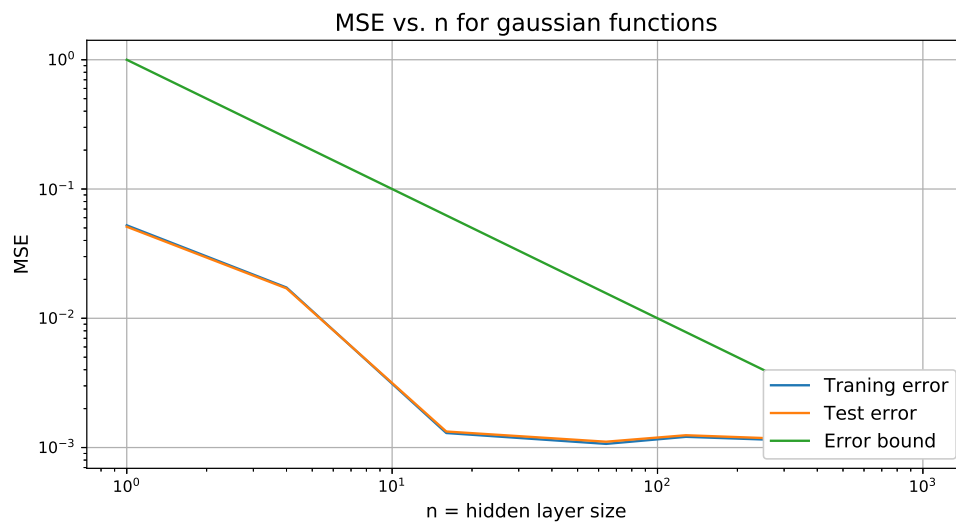


Figure A-1: Gaussian function

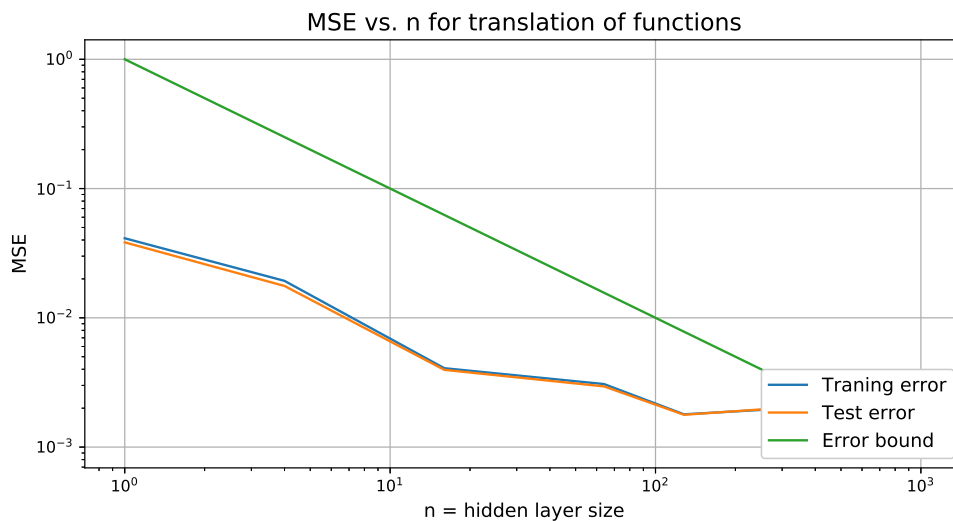


Figure A-2: Translated and scaled Gaussian

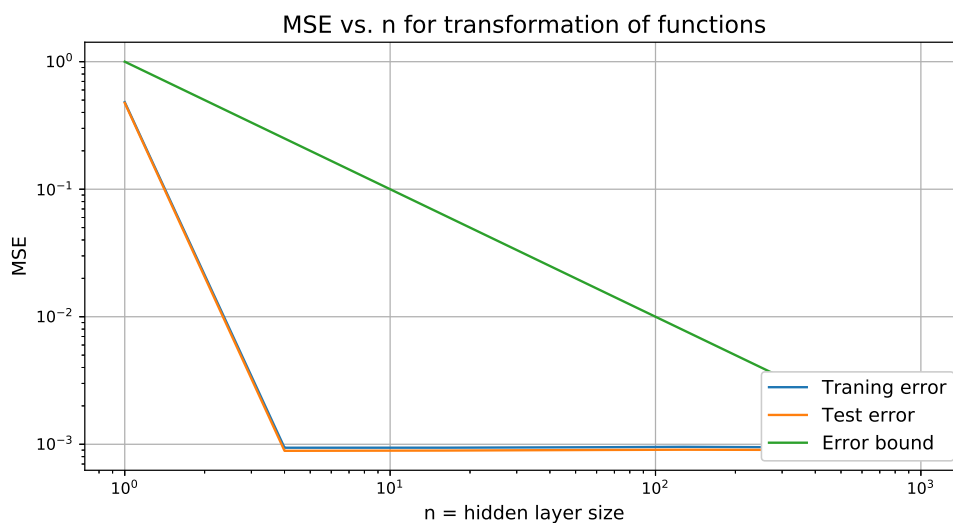


Figure A-3: Linearly transformed (in time) Gaussian

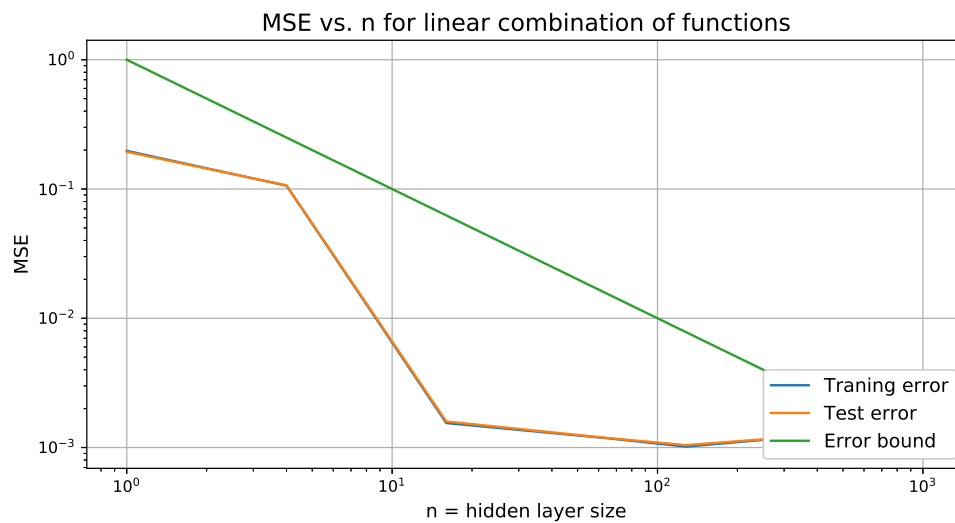


Figure A-4: Linear combination of Gaussians

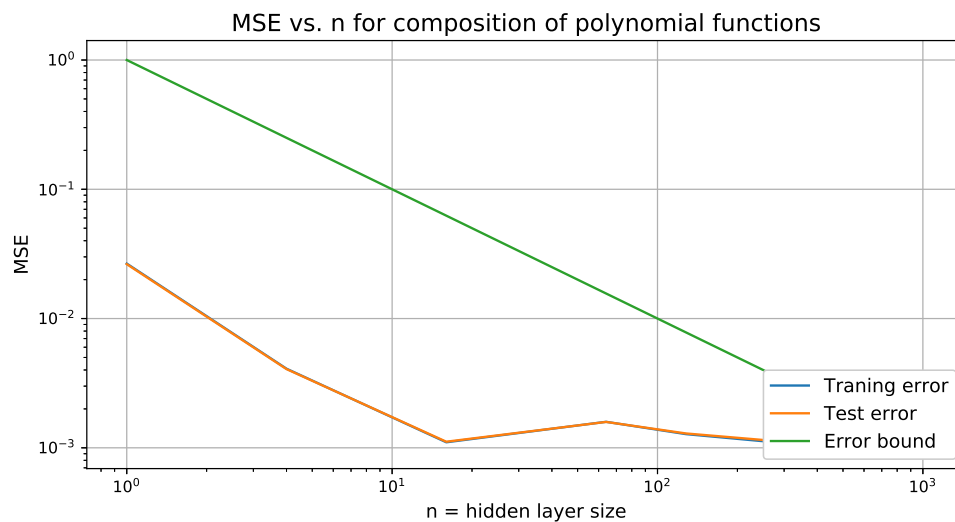


Figure A-5: Composition of Gaussian with polynomials

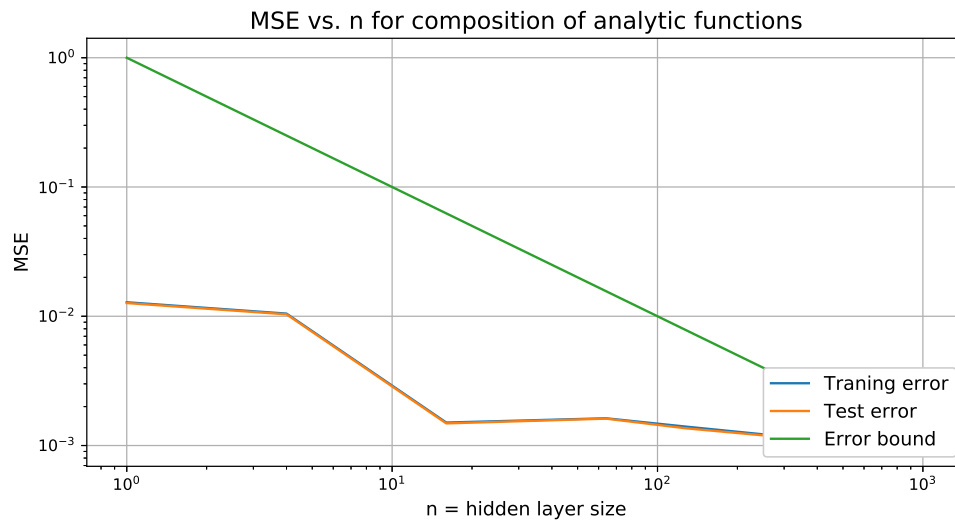


Figure A-6: Composition of Gaussian with cosine

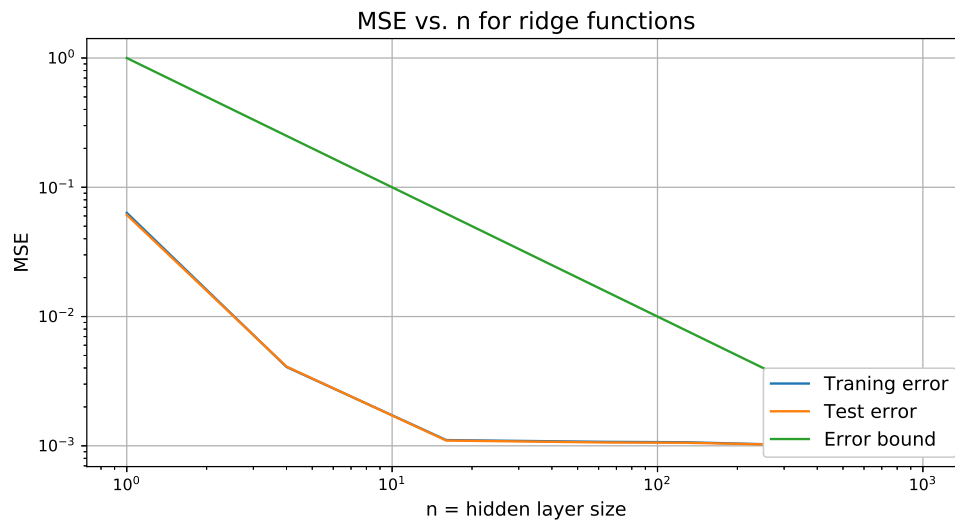


Figure A-7: Ridge function

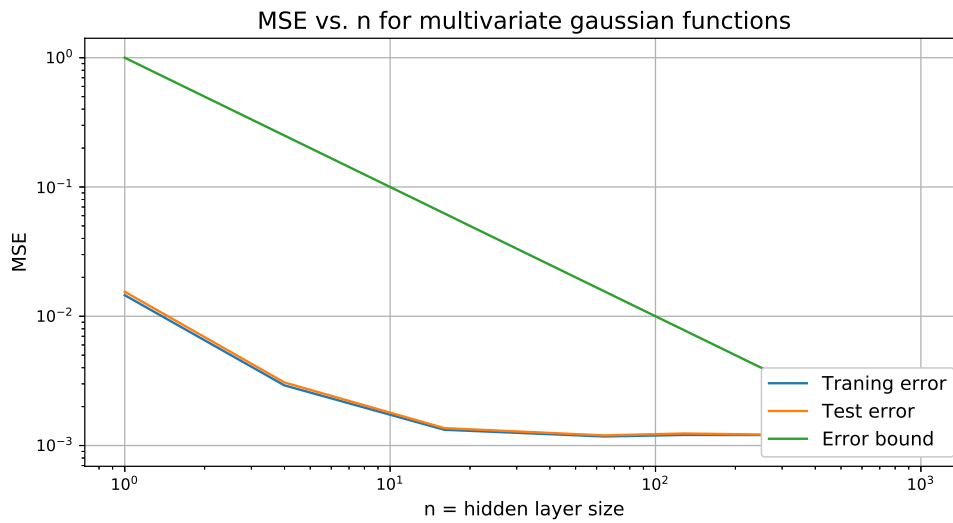


Figure A-8: Multivariate Gaussian distribution

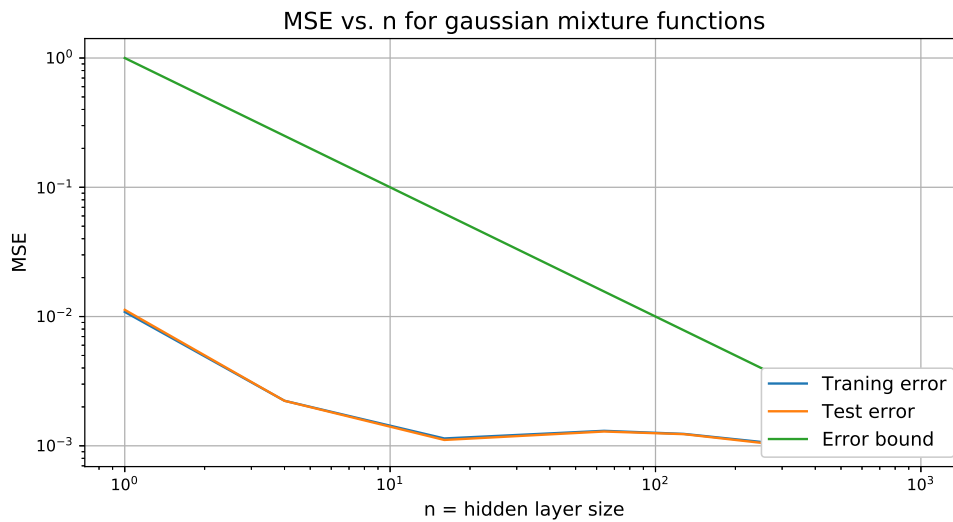


Figure A-9: Gaussian mixture model

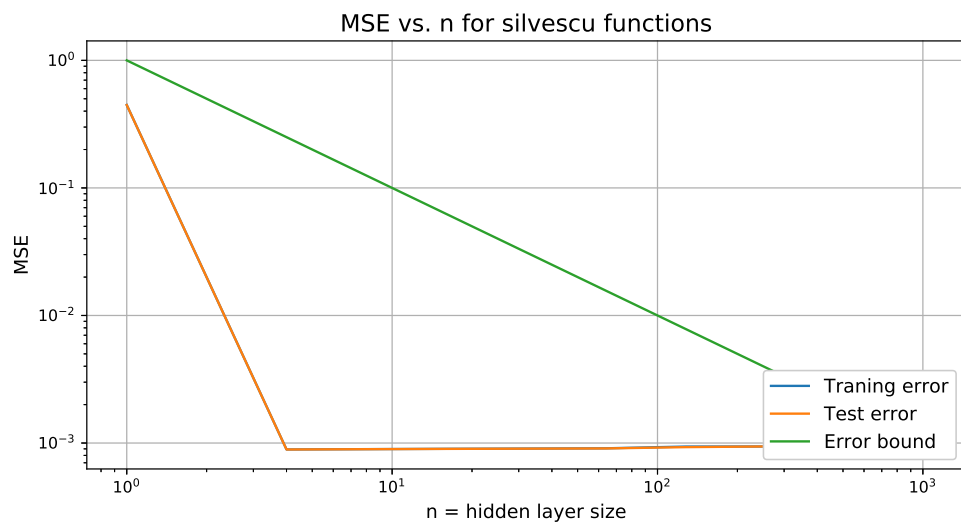


Figure A-10: Silvescu activation function

Bibliography

- [1] A. Silvescu. Fourier neural networks. July 2000.
- [2] A. R. Barron. Universal approximation bounds for superpositions of a sigmoidal function. IEEE Transactions on Information Theory, 39(3):930–945, May 1993.
- [3] G. Cybenko. Approximation by superpositions of a sigmoidal function. math cont sig syst (mcsc) 2:303-314. Mathematics of Control, Signals, and Systems, 2:303–314, December 1989.
- [4] K. Hornik. Approximation capabilities of multilayer feedforward network. Neural Networks, 4:251–257, January 1991.
- [5] O. Shamir, R. Eldan. The power of depth for feedforward neural networks. Conference on Learning Theory, December 2015.
- [6] H. White, A. Gallant. There exists a neural network that does not make avoidable mistakes. volume 1, pages 657–664, August 1988.
- [7] HS Tan. Fourier neural networks and generalized single hidden layer networks in aircraft engine fault diagnostics. Journal of engineering for gas turbines and power, 128(4):773–782, 2006.
- [8] L. Cai, W. Zuo. Adaptive-fourier-neural-network-based control for a class of uncertain nonlinear systems. IEEE Transactions on Neural Networks, 19(10):1689–1701, October 2008.
- [9] S. Liu. Fourier neural network for machine learning. Proceedings - International Conference on Machine Learning and Cybernetics, 1:285–290, January 2014.
- [10] H. White, K. Hornik, Maxwell B. Stinchcombe. Multilayer feedforward networks are universal approximators. Neural Networks, 2:359–366, 1989.
- [11] Lee K. Jones. A simple lemma on greedy approximation in hilbert space and convergence rates for projection pursuit regression and neural network training. The Annals of Statistics, 20, March 1992.
- [12] A. C. Paixão, J. Buescu. Real and complex variable positive definite functions. São Paulo Journal of Mathematical Sciences, 6:155–169, 2012.
- [13] Peter D. Lax. Functional Analysis. John Wiley Sons, 2002.

- [14] R. Shakarchi, Elias M. Stein. Fourier Analysis: An Introduction. Princeton University Press, 2003.
- [15] G. B Folland. Fourier analysis and its applications, volume 4. American Mathematical Soc., 1992.
- [16] J.-P. Montillet. Sobolev spaces, schwartz spaces, and a definition of the electromagnetic and gravitational coupling. Journal of Modern Physics, 8:1700–1722, September 2017.
- [17] Mark A. Pinsky. Introduction to fourier analysis and wavelets. January 2002.
- [18] Irene A. Stegun, M. Abramowitz. Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables. Dover, New York, 1972.
- [19] M. H. Kutner. Applied linear statistical models. McGraw-Hill Education, 5 edition, 2005.
- [20] Y. Huang, J. Benesty, M. M. Sondhi. Handbook Springer of Speech Processing. Springer-Verlag Berlin Heidelberg, 2008.