Masteroppgave

NTNU
Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for matematiske fag

Rasmus Hilmer Henninen

Applying quantum harmonic analysis to Convolutional Neural Networks

Masteroppgave i MSMNFMA
Veileder: Franz Luef
Mai 2023

NTNU
Kunnskap for en bedre verden

Rasmus Hilmer Henninen

Applying quantum harmonic analysis to Convolutional Neural Networks

Masteroppgave i MSMNFMA
Veileder: Franz Luef
Mai 2023

Norges teknisk-naturvitenskapelige universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for matematiske fag

NTNU

Kunnskap for en bedre verden

Contents

List of Figures

Acknowledgments

I would first and foremost like to express my deep gratitude to my advisor, Franz Luef, for his invaluable guidance throughout this journey. His deep understanding of the theory and unending patience have been critical in the development of this thesis. Our discussions have been inspiring and illuminating, and I am grateful for his continuous support.

I extend my gratitude to my friends and family who have generously contributed their time and effort in proofreading this thesis. Specifically, I want to thank Mikkel, Bjørn, and Antonin for their invaluable input and keen eyes, which have played a significant role in refining this work. Their feedback and constant encouragement have undeniably enhanced the quality of this research, for which I am profoundly grateful.

Lastly, but most importantly, I want to express my heartfelt thanks to my girlfriend, Inger Juni. Her unwavering support and encouragement have been my bedrock through the challenging times. Her patience, understanding, and belief in me have not only been a source of comfort but also an inexhaustible source of inspiration.

I am truly grateful to have such an extraordinary group of people supporting me throughout this journey. It is through your continued faith and encouragement that I have been able to accomplish this significant milestone in my academic journey. Thank you.

Abstract

During the past five years the framework of time-frequency analysis has been complemented by notions like mixed-state localization operators, the Cohen class of an operator and its accumulated version. The inspiration for these developments has been the theory of Werner on quantum harmonic analysis about 40 years ago. Among the many applications of these novel tools and methodology Dörfler, Luef and Skrettingland have demonstrated that it provides a way to detect correlations between different data sets. In this project we aim to build on this approach and show its ramifications for convolutional neural networks applied to audio signals which seem to fit well into the framework of quantum harmonic analysis when viewed through the lens of time-frequency analysis.

Sammendrag

I løpet av de siste fem årene har rammeverket for tid-frekvensanalyse blitt komplementert med begreper som blandet tilstandslokalisering operatorer, Cohen-klassen til en operator og dens akkumulerte versjon. Inspirasjonen for disse utviklingene har vært teorien om Werner om kvanteharmonisk analyse for omtrent 40 år siden. Blant de mange anvendelsene av disse nye verktøyene og metodikken har Doerfler, Luef og Skrettingland demonstrert at det gir en måte å oppdage korrelasjoner mellom forskjellige datasett på. I dette prosjektet sikter vi mot å bygge videre på denne tilnærmingen og vise dens konsekvenser for konvolusjonelle nevrale nettverk anvendt på lydsignaler, som ser ut til å passe godt inn i rammeverket for kvanteharmonisk analyse når det ses gjennom linsen av tid-frekvensanalyse.

1 Introduction

Convolutional Neural Networks (CNNs) are a category of Artificial Neural Networks that have proven very effective in areas such as image recognition[26] and classification [28]. Since their inception, they have been developed and refined, powering a multitude of applications and forming a vital part of many advanced technologies.[19]

CNNs have been utilized extensively in computer vision, a field concerned with how computers can gain high-level understanding from digital images or videos [3]. The use of CNNs in computer vision is significant as they can process and understand images in a way that was not possible with previous models [63]. This is due to the use of convolutional layers, which essentially "slide" over the input image to compute a map of features, providing an intuitive way to recognize local and global patterns within an image [19].

CNNs have also been used in natural language processing (NLP) [61]. While recurrent neural networks (RNNs) and transformer models like BERT and GPT are more commonly associated with NLP, CNNs can be and have been used to process text data for tasks such as sentiment analysis or text classification [56]. The convolutional layer in a CNN can identify local patterns, similar to n-grams in text data, which makes them useful in NLP tasks [56]. However, it's important to note that for more complex NLP tasks, models like Transformers often outperform CNNs [32][64].

Amongst the many CNN models developed over the years, certain ones have gained significant attention due to their contributions to the field. LeNet-5, developed by Yann LeCun in 1998, was one of the very first convolutional neural networks, and it has largely influenced the design of subsequent networks. It was initially used for digit recognition tasks, such as reading zip codes, digits in checks. [35]

AlexNet, developed by Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton, was the winner of the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). It demonstrated the power of CNNs in large scale image recognition tasks, significantly outperforming traditional computer vision models. AlexNet is considered a landmark in the field and was responsible for the widespread adoption of CNNs in computer vision. [34]

MobileNet is another significant model developed by Google. As the name suggests, the primary aim of MobileNet is to provide a network architecture that is efficient for mobile and embedded vision applications. Various versions of MobileNet have been introduced over the years, each providing improvements in efficiency and accuracy. MobileNetV3, for example, combined the advantages of previous MobileNet models and added some new features like a new layer type: HardSwish [29]

GhostNet is a more recent CNN model, proposed by researchers from Tencent and the Chinese University of Hong Kong. It introduces a novel module, named GhostModule, to generate more feature maps from cheap operations, thus it's more computational and memory efficient. This makes it very suit-

able for deployment on devices with limited computational power, like mobile devices. [23]

These advancements in CNNs showcase the rapid improvement in artificial intelligence over the past two decades. With their capability to understand and extract complex patterns from data, CNNs have set the stage for the future development of more advanced AI technologies.

Indeed, the computation of convolutions is a cornerstone in the functioning of CNNs, forming the basis for feature extraction in images, text, and even audio data. The efficiency of these computations has been significantly improved with the use of the Fast Fourier Transform (FFT)[7].

In the specific case where the input is a spectrogram, as is often the case when dealing with audio signals in a CNN, this involves a special type of convolution operation. This is where quantum harmonic analysis comes into play.[42][38]

Quantum harmonic analysis provides a mathematical framework for the analysis of systems with a quantized phase space, and its use in the study of convolution operators was first introduced by R. F. Werner in a paper titled "Quantum harmonic analysis on phase space," published in the Journal of Mathematical Physics in 1984 [58].

In this paper, Werner introduced the notion of convolution operators in the quantum context and established an analytic foundation for subsequent explorations in this field.

Over the past five years, however, the understanding of quantum harmonic analysis and convolution operators has been expanded upon through the work of Monika Dörfler, Franz Luef, and Eirik Skrettingland. They have published multiple papers that delve further into Werner's original paper and propose new ways to apply and understand convolution operators within the field of quantum harmonic analysis. [14][37][38][39][54][55] This thesis aims to provide a comprehensive study of how Quantum Harmonic Analysis can be utilized to understand the functioning and utility of Convolutional Neural Networks (CNNs). Despite individual studies such as those by Luef and Skrettingland, demonstrating the possibility of expressing localization operators and spectrograms as operator convolutions [38], and Dörfler's work applying Quantum Harmonic Analysis for adaptive filters in CNNs and to detect correlations between different datasets [13], a holistic exploration of this area remains uncharted.

Therefore, the objective of this thesis is to synergize various concepts from the aforementioned studies, elucidate their interconnections with CNNs, and explore the potential benefits of this formalism in CNNs.

In order to comprehensively address this goal, this thesis is meticulously structured across various key areas – Time-frequency Analysis, Quantum Harmonic Analysis, and Machine Learning – thereby serving as an intersection for these domains.

An initial preliminary section aims to set the stage for readers from diverse backgrounds by introducing elementary concepts in Time-frequency Analysis, Fourier Analysis, and outlining different function spaces that will be crucial for further discussion.

This is followed by a dedicated section on Time-frequency Analysis where

we delve into the Short Time Fourier Transforms, Spectrograms, and Gabor Analysis.

The subsequent section on Machine Learning navigates through the evolutionary trajectory of CNNs, starting with an overview of supervised learning, leading to the development of perceptrons which evolved into neural networks, and finally culminating in the contemporary Convolutional Neural Networks.

The focus then shifts to Quantum Harmonic Analysis, where the thesis outlines various operator convolutions and their properties, accompanied by an appropriate version of a Fourier Transform for operators.

The centerpiece of this thesis is a section that presents the main theorem, illustrating how CNNs can be rewritten. This section will further explore the implications of this theorem.

This is followed by a section devoted to defining a discrete version of the theorem. This is done by adapting the main theorem into a discrete setting, and this part will outline the modifications necessary to facilitate this adaption.

The thesis concludes with a summarization of the central findings, implications and prospective avenues for future research in the interplay of Quantum Harmonic Analysis and CNNs.

2 Preliminaries

This section will provide a foundation of the preliminary theory that is commonly taught at the undergraduate level. Its primary purpose is to define various notations and terminologies that will be used throughout the paper. Given that this paper draws from mathematical, physical, and machine learning disciplines, these preliminaries span across several fields of study. They are presented in a self-contained manner to obviate the need for external sources to comprehend the paper. Readers who are acquainted with the material in any section may proceed directly to the subsequent section.

2.1 Fundamental operations

This section will cover the necessary fundamental operations that we will use in future sections. This mostly consists of defining the different operations required for time-frequency analysis. And is sourced from [21][39].

First, a quite trivial definition to make it clear what z is referring to later in the text:

Definition 2.1 (Point in phase space)**.** A point in phase space ($\mathbb{R}^2$) is denoted by $z = (x, \omega)$.

To simplify the theorems in time-frequency analysis, we also introduce the standard notation for time shifts, frequency shifts, and a combination of both shifts. A time shift simply shifts a function in time:

Definition 2.2 (Time shift)**.** The time shift operator T_x acts on f as follows:

$$T_x f(t) = f(x - t).$$

While a frequency shift multiplies the signal by an exponential:

Definition 2.3 (Frequency shift)**.** The frequency shift operator M_ω acts on f as follows:

$$M_\omega f(t) = e^{2\pi i \omega \cdot t} f(t).$$

Combining time and frequency shifts gives a compact notation for shifting functions in phase space:

Definition 2.4 (Time-frequency shift)**.** The time-frequency shift operator $\pi(z)$ shifts a function in phase space by z as follows:

$$\pi(z) f(t) = M_\omega T_x f(t) = e^{2\pi i \omega t} f(t - x).$$

We may also want to examine functions that are reflected at the origin, which can be accomplished using the parity operator. The parity operator P acts on a function $f(x)$ by reflecting it across the y-axis, and is defined as:

Definition 2.5 (Parity operator for functions)**.** $Pf(x) = f(-x)$.

2.2 Fourier analysis

This section will cover some of the theory required to understand Fourier transforms, which will later be relevant both for defining the Short Time Fourier Transform (STFT) and to defining modulation spaces. Since the theory of Fourier transforms does not work for arbitrary functions we will first look at defining the appropriate space of functions. Sourced from [8] [11] [21]

2.3 Sequence spaces

The idea for why we are interested in p-norms is easier to understand by first starting with sequences instead of functions.

Generally speaking, the norm is simply a way of measuring the size of elements from a space. One quite important family of norms is given by the p-norms:

Definition 2.6 (p-norm for sequences)**.** Let x be a sequence then for $p \geqslant 1$:

$$\|x\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}. \tag{1}$$

Since this expression is not valid for $p = \infty$, it is separately defined as the largest entry: $\|x\|_\infty := \max_i |x_i|$. The main interesting thing to note for this thesis is that the space is increasing in p:

Theorem 2.1. For $1 \leqslant p \leqslant q \leqslant \infty$ the following equation holds:

$$\|x\|_q \leqslant \|x\|_p.$$

Using the p-norms allows us then to define the ℓ^p sequence spaces as the set of sequences that are bounded in the p-norm:

$$\ell^p(\mathbb{R}^d) := \{(x_i)_{i\in\mathbb{N}} : \|x\|_p < \infty\}, \tag{2}$$

which by Theorem 2.1 monotonically decreases in size as p increases. This means that if $p < q$ then there are more sequences in ℓ^p than ℓ^q.

Example: Consider the sequence of all 1's. This sequence is in ℓ^∞, with the largest element being 1. Yet the same sequence is not in ℓ^1 as $\sum_{i=1}^{\infty} |1| = \infty$ or any ℓ^p-space for $p < \infty$.

This is a quite simple way of imposing some stricter restrictions on sequences, which will later be leveraged again in the section on trace class operators.

2.4 Elementary function spaces

For many theorems, it is important to restrict functions or operators to an appropriate class that possesses certain desirable properties. This section will cover the different elementary spaces of functions that are required in the further sections.

2.4.1 $L^p(\mathbb{R}^d)$ space and $\ell^p(\mathbb{R}^d)$

An important space of functions that will later appear in multiple theorems is the function spaces that are derived from the p-norms.

It is no coincidence that $L^p(\mathbb{R}^d)$ and $\ell^p(\mathbb{R})^d$ both share quite similar notation, as we proceed in almost the same manner. First define the p-norms for functions:

Definition 2.7. Let $f(t)$ be a function then:

$$\|f(t)\|_p = \left(\int_{\mathbb{R}^d} |f(t)|^p dt\right)^{\frac{1}{p}}. \tag{3}$$

First we have the important function spaces of L^p which are:

$$L^p(\mathbb{R}^d) := \left\{ f(t) : \left(\int_{\mathbb{R}^d} |f^p| dt\right)^{\frac{1}{p}} < \infty \right\}. \tag{4}$$

This is simply the set of functions that have finite $\|.\|_p$-norm. Often we refer to $L^1(\mathbb{R})$ as the space of integrable functions, and $L^2(\mathbb{R})$ as the space of square-integrable functions. Later an analogous definition for operators will be defined in the form of trace class operators, and Hilbert-Schmidt operators.

2.4.2 Convolutions

As convolutions are an essential part of Convolutional Neural Networks, this section aims to provide a comprehensive introduction to convolutions. Generally, a convolution is an operation that takes two functions as input, and outputs a new function by using the following formula:

Definition 2.8 (Convolution)**.** The convolution of two complex valued functions f, g:

$$(f * g)(t) = \int_{\mathbb{R}} f(t)g(t-\tau)d\tau = \int_{\mathbb{C}} f(t-\tau)g(t)d\tau,$$

where the map is defined from $(\mathbb{C}, \mathbb{C})$ to $\mathbb{C}$.

In the discrete case this can be turned into;

Definition 2.9 (Convolution)**.** The convolution of two functions f, g defined on $\mathbb{Z}$ is

$$(f * g)(t) = \sum_{\tau=-\infty}^{\infty} f[\tau]g[t-\tau].$$

The computational complexity of such discrete convolutions corresponds to the number of operations required. This means that if signal f has length N and signal g has length N the computational complexity would be $O(N^2)$. [48]

2.4.3 Fourier transform

The Fourier transform is an operator that acts on functions, decomposing them into their sinusoidal components. This then allows you to see which frequencies are present in a function or a signal. Intuitively we consider it as an operator that transforms a function from the time domain, where the function is represented by how it changes with respect to time, to the frequency domain where we can see how much of each frequency the signal contains.

Definition 2.10 (The Fourier Transform)**.** For f in $L^1(\mathbb{R})$ we define the Fourier transform by

$$\mathcal{F}\left[f(t)\right] = \hat{f}(\xi) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i\xi t}dt$$

Example 1: As an example, we shall look at the Fourier transform of the Gaussian function $f(t) = e^{-t^2}$, which has special importance in Fourier analysis due to some of its nice properties:

$$\mathcal{F}\left[f(t)\right] = \int_{-\infty}^{\infty} e^{-t^2}e^{-2\pi i\xi t} = \sqrt{\pi}e^{-(\pi\xi)^2}. \tag{5}$$

Example 1 showcases two of the properties that make the Gaussian special. The Fourier transform of the Gaussian is another Gaussian. Additionally, this gives us a trivial example of a function which has a Fourier transform that is never equal to zero.

We also define an operator that acts on functions by taking them from the frequency domain to the time domain, namely the inverse Fourier transform:

Definition 2.11 (The Inverse Fourier Transform)**.**

$$\mathcal{F}^{-1}\left[\hat{f}(\xi)\right] = f(t) = \int_{-\infty}^{\infty} \hat{f}(\xi)e^{2\pi i\xi t}d\xi$$

By taking the Fourier transform we can analyze a function in the frequency domain where some problems are easier to solve. For example, it is easy to make a filter that removes high-frequency components or low-frequency components of a signal which can help filter out unwanted noise from audio recordings. It might also be possible to decouple signals which are intertwined, so a recording of two people talking at the same time can be turned into two signals where only one person is talking. Additionally, it might be easier to make filters on the Fourier side.

Using equation 2.10, we then define the Symplectic Fourier transform using the standard symplectic form.

Definition 2.12 (Symplectic Fourier Transform)**.** Let $A \in L^2(\mathbb{R}^{2d})$:

$$\mathcal{F}_\theta(A(z)) = \int_{\mathbb{R}^{2d}} A(z')e^{-2\pi i\theta(z,z')}dz'$$

where $\theta(z_1, z_2) = \omega_1 \cdot x_2 - \omega_2 \cdot x_1$. An additional subscript of θ is added to differentiate the symplectic Fourier transform from the regular one. The choice of θ as a variable highlights that the symplectic Fourier transform is a rotated version of the Fourier transform. This can be seen by the following equation:

$$\mathcal{F}_\theta(A(x, w)) = \mathcal{F}(A(w, -x))$$

Additionally, we have that:

$$\mathcal{F}_\theta(A(z))^{-1} = \mathcal{F}_\theta(A(z)),$$

$$\mathcal{F}_\theta(A(z))^2 = A(z).$$

Here $A(z)$ is used ,instead of $f(z)$, to highlight that it is a function on $L^2(\mathbb{R}^{2d})$ and not $L^2(\mathbb{R}^d)$.

2.4.3.1 Basic properties of the Fourier transform

The Fourier transform has some nice properties when it comes to convolution, translation and differentiation. So we start by defining these operators.

These operators can easily be combined with the Fourier transform, which the following properties show:

1. $\mathcal{F}\left[T_x f\right](\xi) = e^{-2\pi i x\xi}\hat{f}(\xi)$
2. $\mathcal{F}\left[f * g\right] = \mathcal{F}\left[f\right]\mathcal{F}\left[g\right]$
3. $\mathcal{F}\left[\frac{d^n}{dt^n} f\right](\xi) = (2\pi i\xi)^n \hat{f}(\xi)$

Writing these properties in words:

1. Translating before taking the Fourier transform is equivalent to multiplying by a factor of $e^{-2\pi i x t}$.
2. Taking the Fourier transform of two functions that have been convoluted is equivalent to finding the Fourier transform of each function, then multiplying their Fourier transform together.
3. Taking the derivative of a function and then finding the Fourier transform is equivalent to multiplying the Fourier transform by $(2\pi i\xi)$

This shows that convolution, translation and derivation combine nicely with Fourier transforms. Some additional relevant properties are the Riemann-Lebesgue lemma, Parseval's identity, and the Plancherel theorem:

Definition 2.13 (Riemann-Lebesgue lemma)**.** If $f \in L^1(\mathbb{R}^d)$ then $\hat{f}(\xi) \to 0$ as $|\xi| \to \infty$ and $\hat{f}$ is uniformly continuous

In other words functions in $f(x) \in L^1(\mathbb{R}^d)$ gets mapped into $C_0(\mathbb{R}^d)$, the space of functions vanishing at infinity.

Definition 2.14 (Parseval's identity)**.** For f, g in $L^2(\mathbb{R}^d)$ we have

$$\langle f, g\rangle_L^2 = \int_{-\infty}^{\infty} f(x)\overline{g(x)} = \int_{\mathbb{R}^d} \hat{f}(x)\overline{\hat{g}(x)}, \forall f, g \in L^2(\mathbb{R}^d).$$

Definition 2.15 (Plancherel Theorem)**.** The Fourier transform defined on the dense subspace $L^1 \cap L^2(\mathbb{R}^d)$ may be extended to a unitary operator on $L^2(\mathbb{R}^d)$.

2.4.3.2 Fundamental operations and Fourier transform

There are some useful properties of the fundamental operations that we will later leverage, so this section is dedicated to covering some of the more essential properties when the fundamental operations are combined with Fourier transforms[21]. For the definitions of time-shifts and frequency-shifts see Definition 2.2 and Definition 2.3.

Proposition 1.

$$T_x M_\omega = e^{-2\pi i x\cdot\omega} M_\omega T_x \tag{6}$$

Proof: follows from direct calculation:

$$T_x M_\omega f(t) = e^{2\pi i\omega(t-x)} f(t-x) =$$
$$e^{-2\pi i x\cdot\omega} e^{2\pi i\omega t} f(t-x) = e^{-2\pi i x\cdot\omega} M_\omega T_x f(t) \quad \square$$

For any of the L^p-spaces, the shifts also define isometries:

$$||T_x M_\omega f||_p = ||f||_p.$$

By direct calculation, the following relations with fundamental operators and the Fourier transform hold:

$$\widehat{(T_x f)} = M_{-x}\hat{f}, \tag{7}$$

$$\widehat{(M_\omega f)} = T_\omega \hat{f}. \tag{8}$$

Or by combining (7) and (8)

$$\widehat{(T_x M_\omega f)} = M_{-x} T_\omega \hat{f} = e^{2\pi i x\cdot\omega} T_\omega M_{-x}\hat{f}$$

2.4.3.3 Convolutions and Fourier transforms

One of the advantages of Fourier transforms is how it turns computing convolutions into something both practically and theoretically more simple. This follows from the following theorem:

Theorem 2.2. Supposed that $f, g \in L^1(\mathbb{R}^d)$. Then we have that

$$\widehat{f * g} = \hat{f}\hat{g}.$$

With some creative usage, this theorem has various uses when it comes to optimizing calculations. Some examples include: very efficient algorithms for multiplications, or for finding prime factors. [2] [33].

The trick usually done is using the well-known fast algorithm for numerically computing Fourier transforms which is called the fast Fourier transform.

Implementing the convolution using the fast Fourier transform that leverages this convolution theorem has a computational complexity of $O(N \log(N)$ [48]. That is a quite big improvement from the naive implementations complexity of $O(N^2)$.

This is achieved by using the convolution theorems and calculating the convolution in the following way:

$$f * g = \mathcal{F}^{-1}\mathcal{F}(f * g) = \mathcal{F}^{-1}(\hat{f} \cdot \hat{g}) \tag{9}$$

The trick of using an appropriate Fourier transform for calculating convolutions is one of the motivations for later theorems.

2.5 Function spaces for time-frequency analysis

As noted in the section about Fourier transforms any arbitrary function does not necessarily have a well-behaved Fourier transform. This section will cover some of the potential choices of function spaces and will end with defining the Feichtinger's algebra which is the space that is currently considered the most appropriate for time-frequency analysis.

2.5.1 Schwartz space

This section will cover one of the most used spaces for making sure the Fourier transform is well defined, namely the Schwartz space of functions.

The idea behind the Schwartz space is to add some requirements for smoothness and for the function to decay sufficiently fast to zero. Formally the Schwartz space $\mathcal{S}(\mathbb{R})$ is defined as the set of all infinitely differentiable functions $f : \mathbb{R} \to \mathbb{C}$ such that for any multi-indices α, β and any constant $C_{\alpha,\beta}$, we have that the following family of seminorms is finite:

$$c_{\alpha,\beta)}(f) := \sup_{x \in \mathbb{R}^d} \|x^\alpha \partial^\beta f(x)\| < \infty,$$

where $\partial^\beta f(x)$ denotes the β-th derivative of f at x, and x^α denotes the α-th power of x. This is equivalent to the following equation: [60]

$$|x^\alpha \partial^\beta f(x)| \leqslant C_{\alpha,\beta} \cdot (1 + x^2)^{-\frac{|\alpha+\beta|}{2}}, \forall x \in \mathbb{R}^d \tag{10}$$

has to be finite regardless of the choice of $\alpha, \beta \in \mathbb{N}$. This allows us to define a notion of convergence:

Definition 2.16 (Convergence in Schwartz spaces)**.** A sequence of functions f_n converges to f if:

$$\|f_n - f\|_{c_{\alpha,\beta}} \to 0$$

as $n \to \infty$. For all $\alpha, \beta \in \mathbb{N}$.

The topology of Schwartz spaces is defined by the countable family of norms $c_{\alpha,\beta}$. This topology is not derived from a norm, but it is metrizable by the following norm:

$$d(f,g) = \sum_{\alpha,\beta \in \mathbb{N}/0} \frac{k_{\alpha,\beta}\|f-g\|_{c_{\alpha,\beta}}}{1+\|f-g\|_{c_{\alpha,\beta}}} \tag{11}$$

Furthermore, this metric space is complete with respect to this metric. Since the topolgogical vector space is complete and defined by a countable family of seminorms it is a Fréchet space. [4]

Functions in the Schwartz space are rapidly decreasing, which makes them well-suited for Fourier analysis. Specifically, the Schwartz space has the following properties.

- Any function in $\mathcal{S}(\mathbb{R})$ and its Fourier transform are also in $\mathcal{S}(\mathbb{R})$.
- The Fourier transform is a continuous and invertible linear operator on $\mathcal{S}(\mathbb{R})$.
- $\mathcal{S}(\mathbb{R})$ is dense in $L^2(\mathbb{R})$, meaning that any function in $L^2(\mathbb{R})$ can be approximated arbitrarily well by a sequence of functions in $\mathcal{S}(\mathbb{R})$.

Example of a Schwartz function: Let ϕ_0 be the L^2- normalized Gaussian:

$$\phi_0 := 2^{\frac{d}{4}} e^{-\pi x^2}. \tag{12}$$

Then $\phi_0 \in \mathcal{S}$.

Unfortunately, the Schwartz spaces is quite "small", which we can see in the following example.

Example: Consider the function: $g(t) = 1 - |t|$. Which is quite an elementary and simple function, yet since it is not differentiable at $g(t) = 0$ it is not in the Schwartz space.

Working with equation 10 is unfortunately not very nice as dealing with multi-indices is quite challenging, and it is quite easy for a function to drop out of a Schwartz space after some transformation.

To avoid dealing with these issues a different space that is more suitable for this thesis is used instead. Namely the Feichtinger algebra:

2.5.2 Feichtinger's algebra

To understand Feichtinger's algebra this section will first cover tempered distributions which are required for defining modulation spaces, which the Feichtinger's algebra is just a special case of.

We call the dual space of the Schwartz space for the space of tempered distributions. Which is the space all linear and continuous functionals on $\mathcal{S}$:

Definition 2.17 (Tempered distributions)**.** The set of all linear and continuous functions from $\mathcal{S} \to \mathbb{R}$. Denoted by $\mathcal{S}'$

$$\mathcal{S}'(\mathcal{S}) = \{f \in \mathcal{S} : f \to \mathbb{R} \text{ is linear and continuous}\}.$$

The tempered distributions will be the building blocks of modulation spaces, they have some interesting properties.

Then the definition of modulation spaces is as follows:

Definition 2.18 (Modulation spaces)**.**

$$M_m^{p,q}(\mathbb{R}^d) := \left\{ f \in \mathcal{S}'(\mathbb{R}^d) : \left(\int_{\mathbb{R}^d} \left(\int_{\mathbb{R}^d} \left(\left| \int_{\mathbb{R}^d} f(t)\overline{g(x-t)} e^{-2\pi i \omega t} dt \right|^p m(x,\omega)^p dx \right)^{\frac{q}{p}} d\omega \right)^{\frac{1}{q}} < \infty \right) \right\}$$

The canonical choice of g is letting it be equal to ϕ_0, but any other Schwartz functions may be used. Later the expression inside the absolute values will become important as this is actually a short-time Fourier transform. But to finally get to the Feichtinger algebra, introduced by Feichtinger in [15], which contains the Schwartz spaces [2]. We need simply to consider the modulation space where $m = p = q = 1$.

Then the Feichtinger algebra is defined to be the set of all tempered distributions (Defintion: 2.17) such that the integral of the absolute value of the STFT(equation 3.1) of the distribution with the normalized Gaussian (equation: 12) as a window is finite. Written explicitly this is then the following:

Definition 2.19 (Feichtinger's algebra)**.**

$$\mathcal{S}_0(\mathbb{R}^d) := \{f \in \mathcal{S}'(\mathbb{R}^d) : \int_{\mathbb{R}^{2d}} \left(| \int_{\mathbb{R}} f(t)\overline{g(x-t)} e^{-2\pi i \omega t} | dt \right) dz < \infty\}$$

If we compare with the definition for modulation spaces matches for $m = p = q = 1$ as previously noted. Another way of denoting Feichtinger's algebra is as the modulation space $M^1(\mathbb{R}^d)$.

Examples of $M^1(\mathbb{R}^d)$ functions:

- Two sided exponential: $g(t) = e^{-|t|}$
- Triangle function: $g(t) = 1 - |t|$
- Hyperbolic secant: $g(t) = \frac{1}{\cosh \pi t}$

Note now that the triangle function is included!

Feichtinger's algebra turns out to be a good class of test functions and can in many cases be a good substitute for the Schwartz spaces. See the survey[30]. The continuous dual space of $M_1(\mathbb{R})$ may be identified by $M^\infty(\mathbb{R})$ which is the space of all tempered distributions $f \in \mathcal{S}'$ such that:

$$\|f\|_{M^\infty} = \sup_{z \in \mathbb{R}^2} |V_g(f(z))| < \infty. \tag{13}$$

Examples of elements in $M^\infty(\mathbb{R})$:

- The δ-distribution: $\delta(t) = \lim_{b\to 0} \frac{1}{|b|\pi} e^{-(\frac{x}{b})^2}$.
- The shah distribution: $f(t) = \sum_{n=-\infty}^{\infty} \delta_n$.

In order to distinguish between tempered distributions and these distributions we refer to elements of M^∞ as mild distributions.

Feichtinger also proved that there exists a kernel for bounded linear operators $A : M^1(\mathbb{R}) \to M^\infty(\mathbb{R})$. Similar to how there exists a kernel of continuous linear operators between the Schwartz space and the space of tempered distributions. [16]

Theorem 2.3 (Feichtinger)**.** Let A be a bounded linear operator from $M^1(\mathbb{R})) \to M^\infty(\mathbb{R}))$. Then there exists a $k_A \in M^\infty(\mathbb{R})$ such that $\langle g, Tf\rangle = \langle f \otimes g, k_A\rangle$.

Here the $\langle ., .\rangle$ denotes the pairing between $M^1(\mathbb{R}))$ and $M^\infty(\mathbb{R})$ which is well-defined since the space of linear functionals of $M^1(\mathbb{R})$ may be identified with $M^\infty(\mathbb{R})$

Ultimately we also need to define translation invariant operators. These are the operators which are unaffected by shifting them back and forth:

Definition 2.20 (Translation invariant)**.** An operator A is translation invariant if the following property is satisfied for all $x \in \mathbb{R}$:

$$T_x A = A T_x$$

or equivalently if

$$T_x A T_{-x} = A.$$

Let $k_A \in M^\infty(\mathbb{R}^2)$ be the kernel of an operator A. Then the translation invariance Definition 2.20 implies that the kernel is of the form:

$$k_A(x, y) = k(x - y) \tag{14}$$

for some $k \in M^1$. This allows us to rewrite the operator A in the following manner for a $f \in M^1$:

$$A = Afk * f \tag{15}$$

Which is a convolution operator. Reiterating this result gives us a quite useful theorem:

Theorem 2.4. Any bounded and translation invariant operator from $M^1(\mathbb{R}) \to M^\infty(\mathbb{R})$ is a convolution operator.

Another well-known characterization of this class of operators is [10]:

Theorem 2.5. Let A be a translation invariant linear bounded operator from $M^1(\mathbb{R}) \to M^\infty(\mathbb{R})$. Then there exists a mild distribution $m \in M^1(\mathbb{R})$ such that $\widehat{Af} = m \cdot \hat{f}$ for all $f \in M^1(\mathbb{R})$

Such operators are called Fourier multipliers. The previous theorem in terms of operators we have:

$$\mathcal{F}(Af) = m\mathcal{F} \tag{16}$$

or equivalently:

$$A(f) = \mathcal{F}^{-1}(m\mathcal{F})(f) \tag{17}$$

Convolution operators appear in many areas of engineering, physics and mathematics, for example in filters in signal analysis. One of the reasons for this is that they have this nice description in the form of a Fourier multiplier.

There exists an analogous result in Quantum harmonic analysis that we exploit for describing convolutional neural networks.

2.6 Properties of operators

This section will cover some special types of operators that have properties that are needed later. First, we look at compact operators which are of interest since they can be decomposed using a singular value decomposition.

Definition 2.21 (Compact operator)**.** A linear operator that maps compact subsets of the domain to relatively compact subsets in the codomain.

For compact operators we have a quite useful decomposition [22]:

Definition 2.22 (Singular value decomposition)**.** Let S be a compact operator on $L^2(\mathbb{R}^d$. Then there exist two orthonormal sets $\{b_i\}_{i\in\mathbb{N}}$ and $\{v_i\}_{i\in\mathbb{N}}$ in $L^2(\mathbb{R}^d)$ and a sequence $\{s_i(S)\}_{i\in\mathbb{N}}$ of positive numbers with $s_n(S)$ such that S may be expressed as:

$$S = \sum_{n\in\mathbb{N}} s_n(S) b_n \otimes v_n$$

The definition of positive operators is necessary when defining trace class operators:

Definition 2.23 (Positive operators)**.** An operator is positive if it satisfies the following condition:

$$\langle Sf, f\rangle \geqslant 0, \ \forall f \in L^2(\mathbb{R}^d).$$

3 Time-frequency analysis

Time-frequency analysis is a crucial tool in signal processing, audio analysis, and image processing. It allows for simultaneous examination of signals in both the time and frequency domains, providing insights into their temporal and spectral characteristics. In this section, we explore various techniques of time-frequency analysis, including the Short-time Fourier Transform (STFT), spectrograms, sampled spectrograms, Gabor analysis, and Gabor frames. These techniques offer valuable insights into signal properties, localization, and representation. By studying these methods, we aim to enhance our understanding of time-frequency analysis and its applications.

3.1 Short-time Fourier Transform

The idea of the short-time Fourier transform is to obtain some information about the local properties of a function. This is done by restricting the function to a smaller duration in time by multiplying it with a function with finite support called a window function.

The STFT is defined to be [17]:

Definition 3.1.

$$V_g f(t,\xi) = \int_{\mathbb{R}^d} f(t)\overline{g(t-x)}e^{-2\pi i t\cdot\xi}dt, \forall t,\xi \in \mathbb{R}^d,$$

where $g(x)$ is often a window function with compact support, here we follow the notation of [14]. But we can then rewrite it the following way:

$$\int_{\mathbb{R}^d} f(t)\overline{g(t-x)}e^{-2\pi i t\cdot\xi}dt = \langle f(x), M_\xi T_t g\rangle = \langle f(x), \pi(z)g\rangle$$

Following Gröchenig's [21] book it is possible to rewrite the STFT in the following equivalent ways:

(a) $\widehat{f\cdot T_x\overline{g}}(\omega)$

(b) $\langle \hat{f}, T_\omega M_{-x}\hat{g}\rangle$

(c) $e^{-2\pi i x\cdot\omega}V_{\hat{g}}\mathcal{F}f(w,-x)$

(d) $e^{-2\pi i x\cdot\omega}(f * M_\omega g^*)(x)$

(e) $(\hat{f} * M_{-x}\hat{g}^*)(\omega)$

(f) $e^{-\pi i x\cdot\omega}\int_{\mathbb{R}^d} f(t+\frac{x}{2})\overline{g(t-\frac{x}{2})}e^{-2\pi i x\cdot\omega}dt$

Where $h^*(x) = \overline{h(-x)}$.

For this thesis the main things to note are the following. Using equation (c) above shows why this is a valid time-frequency representation as it relates the

STFT of a function to the STFT of the Fourier transform of the same function. Notice that the difference is simply a phase factor and a rotation. Writing it explicitly out:

$$V_g(f(x,w) = e^{2\pi i x\cdot\omega} V_{\hat{g}}\widehat{f}(\omega, x). \tag{18}$$

This is the fundamental identity of time-frequency analysis.

Additionally, there are other quadratic form representations that might be useful for different applications such as the Ambiguity function:

Definition 3.2 (Ambiguity function)**.**

$$A(f,g) = e^{\pi i x\cdot\omega} V_g f.$$

Or if you consider equation (f) instead you get the cross-ambiguity function which is often used in radar and optics[21]:

Definition 3.3 (Cross-ambiguity function)**.**

$$\int_{\mathbb{R}^d} f(t+\tfrac{x}{2})\overline{g}(t-\tfrac{x}{2})e^{-2\pi i x\cdot\omega}dt. \tag{19}$$

3.2 Spectogram

A spectrogram is a visual representation of the frequency content of a signal over time and was introduced in [9]. It provides information about how the frequency components of a signal change over time, making it useful in a variety of fields including audio signal processing, speech analysis, and acoustic studies. In a spectrogram, the horizontal axis represents time, the vertical axis represents frequency, and the color or intensity represents the amplitude or power of the frequency component.

To find calculate a spectogram simply take the square of the STFT:

Definition 3.4 (Spectogram)**.** The square of the absolute value of the STFT:

$$|V_g(f,\xi)|^2.$$

Figure 1 shows a signal and its corresponding spectrogram generated using the code in the example. The signal is a combination of two sine waves with frequencies of 2 Hz and 20 Hz, respectively. The top plot in Figure 1 shows the time-domain representation of the signal, which is a plot of the signal amplitude versus time. The bottom plot in Figure 1 shows the spectrogram of the same signal, which is a 2D plot of the signal's frequency content versus time. In the spectrogram, the color represents the magnitude of the signal at each frequency and time point.

From the spectrogram, we can see that the signal has a dominant frequency of 20 Hz and a weaker frequency component at 2 Hz. The color changes over time indicating the variation in the signal's frequency content. We can also observe

that the frequency content of the signal changes rapidly at the beginning and end of the signal while remaining relatively stable in the middle.

Spectrograms have been widely used in various fields such as audio signal processing, speech analysis, and acoustic studies [46]. They provide valuable information on the frequency components of a signal over time, making them useful for analyzing and visualizing signals with complex frequency content.

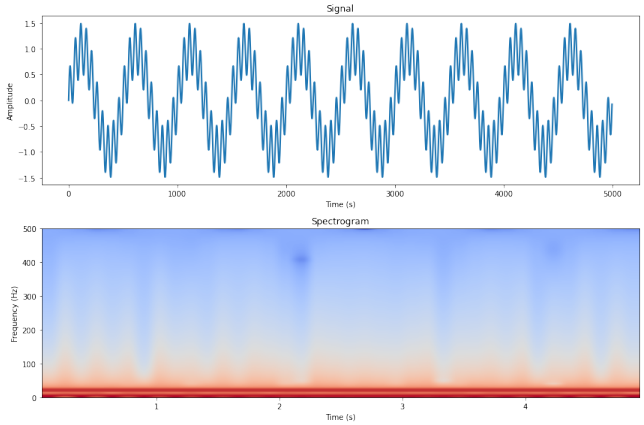


Figure 1: Illustration of a signal and its spectogram

A spectrogram, depicted as a heatmap, represents the intensity of each frequency component in a signal at a given time. More intense colors represent more of that frequency at a given time.

3.3 Gabor analysis

In the continuous case, Gabor analysis is about taking a Gabor system of the following form: $\{\pi(z)g : z \in \mathbb{R}^2\}$.

And attempting to write a reconstruction formula for $f \in L^2(\mathbb{R})$:

$$f = \iint_{\mathbb{R}^2} V_g f(z)\pi(z)g \, dz. \tag{20}$$

The idea proposed by Daubechies was to modify these reconstruction formulas by using a STFT mulitplier. This leads to the following class of operators, localization operators defined by

$$A_a f = \iint_{\mathbb{R}^2} a(z)V_g f(z)\pi(z)g \, dz \tag{21}$$

for a symbol $a \in M^\infty(\mathbb{R}^2)$.

Definition 3.5 (Gabor systems)**.** The Gabor system of a window g is the set of translations and modulations of g given by:

$$\{g_{a,b} = M_{av_0} T_{bt_0} g : a, b \in \mathbb{Z}\}$$

For the appropriate choice of a, b the spectogram can be calculated like this as well back from this definition. But by also considering different modulations, it is possible to create better sampling schemes. Either approach works but there are some theoretical reasons for preferring Gabor systems. For an in-depth survey of Gabor systems, the interested reader may refer to [53].

3.4 Gabor frames

A special case of Gabor systems has some nice properties. Namely the Gabor frames:

Definition 3.6 (Gabor frames)**.** Let $g_{a,b}$ be a Gabor system, and if it additionally satisfies the following bounds for all $f \in L^2(\mathbb{R})$:

$$A\|f\|_2 \leqslant \sum_{a,b \in \mathbb{N}} |\langle f, g_{a,b} \rangle|^2 \leqslant A\|f\|_2$$

then $g_{a,b}$ is a Gabor frame for $L^2(\mathbb{R})$.

Example Gabor frame: If we consider the Gaussian function and the lattice $\Lambda = a\mathbb{Z} \times b\mathbb{Z}$. Then this forms a Gabor frame if and only if $ab < 1$, a well-known result due to Seip-Wallsten and Lyubarskii.

For a Gabor system there are three fundamental operations:

- Analysis operator $L^2(\mathbb{R}) \to \ell^2(\Lambda)$: $Cf \to \{\langle f, \pi_\lambda g \rangle\}_{\lambda \in \Lambda}$ which takes a function and gives you a sequence.
- Synthesis operator $\ell^2(\Lambda) \to L^2(\mathbb{R})$: $Dc = \sum_{\lambda \in \Lambda} c_\lambda \pi_\lambda g$
- Frame operator $L^2(\mathbb{R}) \to L^2(\mathbb{R})$:S $D \circ C = \sum_{\lambda \in \Lambda} \langle f, \pi_\lambda g \rangle \pi_\lambda g$

4 Machine learning

Machine learning and artificial intelligence have received significant attention in recent years, with the development of increasingly sophisticated models such as the Chat GPT[51]. The present thesis will focus on convolutional neural networks (CNNs), which represent a class of deep learning models that have shown remarkable performance in various domains, including image [27] and speech recognition[1], natural language processing[6], and reinforcement learning[41].

To facilitate readers who may be unfamiliar with CNNs, it is pertinent to provide a brief overview of the inception of neural network research, which began with the development of perceptrons. Perceptrons are a class of artificial neurons that are capable of learning and making decisions based on input signals. They were introduced in the 1950s by Frank Rosenblatt, who proposed a simple algorithm for training them. However, perceptrons had limited capabilities and could only classify linearly separable data. This limitation led to the "perceptron controversy," which questioned the ability of perceptrons to solve complex problems.[45]

Over time, researchers discovered that stacking multiple layers of neurons (i.e., creating neural networks) could overcome the limitations of individual perceptrons and enable them to learn complex patterns[19]. This led to the development of backpropagation, an algorithm for training neural networks, which enabled them to learn from large datasets and generalize to new examples[57]. In the 1980s, researchers began exploring the use of convolutional layers in neural networks[36], which were inspired by the visual cortex in the human brain[20].

Convolutional layers introduced the concept of weight sharing, which reduced the number of trainable parameters in the network and allowed it to learn translation-invariant features[36]. This greatly improved the performance of neural networks on image classification tasks, and in 2012, the AlexNet model achieved state-of-the-art performance on the ImageNet benchmark, which consists of millions of labeled images [34]. Since then, CNNs have become a dominant model in computer vision and have been extended to various other domains[3].

The outline of the subsequent sections is as follows

1. Supervised machine learning
2. Perceptrons
3. Artificial neural networks
4. Activation functions
5. Convolutional neural networks
6. Some implementation details

4.1 Supervised learning

Supervised learning methods are machine learning methods that try to figure out the underlying relationship between data about an object and its classification, when we are given samples of different data with the correct classification. This can be broken into two different paradigms, either finding the right label (classification) or by finding a relation between the dependant variables, and one or more explanatory independent variables. [25].

In supervised learning, the data consists of samples, each with a corresponding correct label. So given some sample data denoted by $\mathcal{D}$ which is sampled from a distribution $\mathcal{X}$ of different objects, there is also given a corresponding description of these objects called the target $\mathcal{T}$ with the correct label taken from a set of labels $\mathcal{Y}$. The goal of supervised learning is then to find a function which could be used to describe new data points. [25]

Definition 4.1 (Supervised learning)**.** Given data samples $(\mathcal{D}, \mathcal{T}) \subset (\mathcal{X}, \mathcal{Y})$, try to find a function that satisfies $f(\mathcal{X}) = \mathcal{Y}$

Multiple different functions solve this problem[24], so to pick which function is the "best" we need some way to compare them. This is done by using a statistical tool called loss functions. Which measure how well the function we chose coincides with the samples we were given[25].

Definition 4.2 (Loss function)**.** A loss function L is a function that measures how well a function f fits the data at each point by comparing the target with the estimated target.

$$L : (\mathcal{Y}, f(\mathcal{X})) \to \mathbb{R}$$

Different loss functions can be used depending on how you would like to compare the functions, but some commonly used ones are:[24]

- L^1 norm loss function: $L(y_i, \hat{y_i}) = |y_i - \hat{y_i}|$ [31]
- L^2 norm loss function: $L(y_i, \hat{y_i}) = (y_i - \hat{y_i})^2$[31]
- Cross entropy loss function: $L(y_i, \hat{y_i}) = -y_i \log(\frac{e^{\hat{y}}}{\sum e^{y_j}})$[25]

4.2 Perceptrons

The first steps towards neural networks were taken in 1958 with Frank Rosenblatt's model of the brain, called the Perceptron.[52]. The core idea of the papers were that neurons in the brain respond and fire of a signal when they are given sufficiently large inputs based on some threshold.

As the different inputs x might not necessarily be equally important they are given some associated weight w of how important the input is, and then a bias b is added to allow for different levels of thresholding[25].

Definition 4.3 (The perceptron)**.** The perceptron is the function:

$$f(x) = \begin{cases} 1 & w \cdot x + b > 0 \\ 0 & otherwise \end{cases}$$

Visualizing the perceptron or neural networks has been done by various authors [19][25]. Below is an illustration using the same idea:

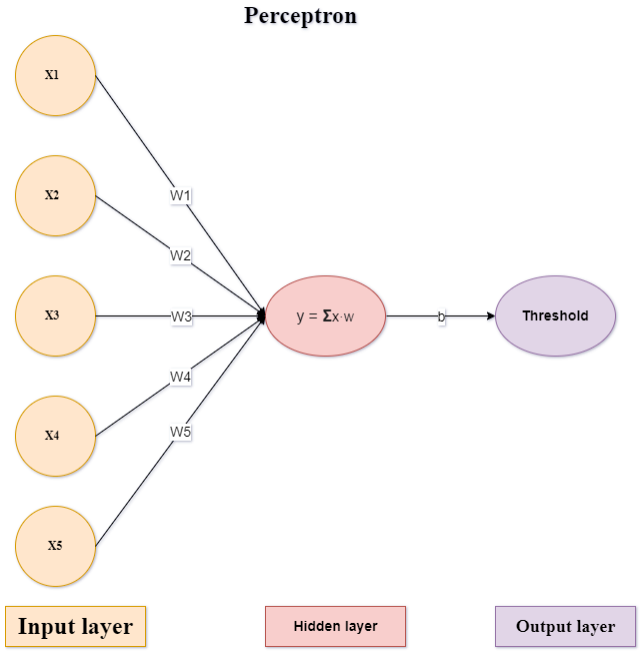


Figure 2: Illustration of a Perceptron

Here the vectorized notation is used which is shorthand for writing out:

$$w \cdot x = \sum_{i=1}^{n} w_1 x_1 = w_1 x_1 + w_2 x_2 + w_3 x_3 + ...w_n x_n$$

Which shows that each input is given an associated weight. The main limitation of this method is that it is a linear function that classifies the input domain into 0 or 1 based on which side of the line $w \cdot x + b = 0$ the input falls on. Which means the boundary which separates the inputs into 0 or 1 has to be a straight line which means non-linear decision boundaries can not be modelled.

4.3 Neural Networks

Artificial Neural Networks were developed in 1986 [47] and it is a method that improves on the issues on perceptrons. In this model, a network's architecture

is defined where the data points are fed through potentially multiple layers of perceptrons, each with a non-linear activation function [25].

Definition 4.4 (Activation function)**.** An activation function is a non-linear function that takes the data as input similar to the perceptron denoted by $\sigma(\cdot)$

Some commonly used activation functions are:

- Rectified Linear Unit (ReLU): $\sigma(x) = max(0, x)$[62]
- Sigmoid like functions with an S-shape. Examples include $\sigma(x) = \frac{1}{1+e^{-x}}$ or $\sigma(x) = \tanh x$. [43]

There are various different ways of writing an expression for one **unit** in a layer of a neural network [25][19], which takes the previous layer as an input and passes it through an activation function, but some of the commonly seen ones are:

$$\sigma(ax + b) = \sigma\left(\sum_{i=1}^{n} x_i w_i + b\right) = \sigma\left(w^T x + b\right) \tag{22}$$

A **layer** in a neural network will usually consist of multiple of these units stacked on top of each other. Some commonly used terms that are useful to know when talking about neural networks:[25]

1. Input layer (X_i): The first layer which constitutes a vector of inital inputs to the neural network.
2. Hidden layer: The middle layers in a neural network which takes the previous layer as input and outputs a vector for the next layers.
3. Output layer: The final layer in a neural network which takes the inputs and transforms it into the final output. Where the final prediction is denoted by: $\hat{y}$

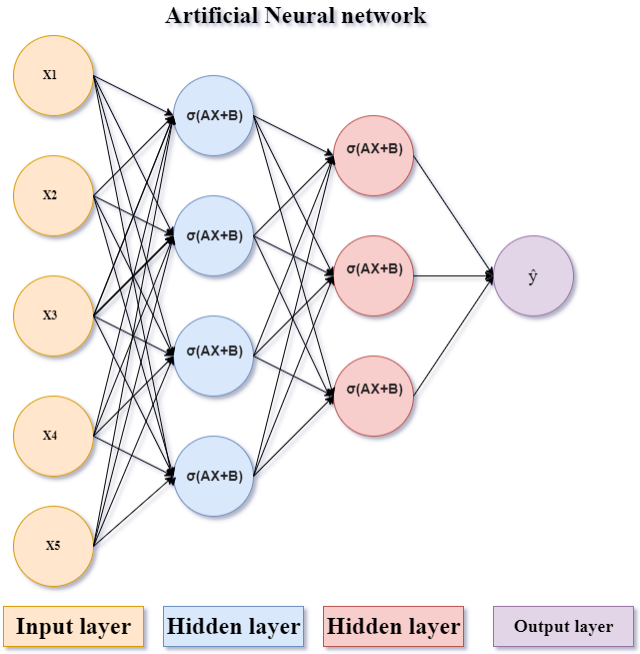


Figure 3: Illustration of an example of a Neural Network

This is faster to compute, and here we will then denote the number of units in a layer by K_n. If the activation functions in a given hidden layer is identical, it is possible to combine the entire column into a vector. This is computationally more efficient, and we denote the number of units in a layer by K_n

$$\sigma(AX+B) = \sigma\left(\sum_{k_n=1}^{K_n} x_{i(k_n)} w_{i(k_n)} + 1 \otimes b\right) = \sigma\left(\begin{bmatrix} w_1^T x_1 + b_1 \\ w_2^T x_2 + b_2 \\ \vdots \\ w_{K_n}^T x_{K_n} + b_{K_n} \end{bmatrix}\right).$$

If there is only one layer of such activation functions it is usually called an neural network, but if there are multiple layers it is called a deep neural network. [25]

Definition 4.5 (Deep Neural network)**.** A neural network with multiple layers.

As an example consider the Deep Neural network illustrated in figure 3 which consists of two hidden layers. Representing it mathmathically can be done as following if the output layer is simply the identity function $f(x) = x$:

$$\hat{y} = \sigma_2\left(A_2\sigma_1(A_1X + B_1) + B_2\right))$$

4.4 Convolutional Neural Networks

The main topic of interest for this paper are neural networks least one of the layers uses convolutions. These are called Convolutional Neural Networks.

Definition 4.6 (Convolutional Neural Networks)**.** A neural network where one of the layers is computed using a convolution, this also reduces the amount of parameters.

Then we can represent the convolutional layers by the following formulas:

$$\sigma(Ax + b) = \sigma(\sum_{k_n=1}^{K_n} (S_n(k_n) * w(n)) + 1 \otimes b) \tag{23}$$

The symbol $w(n)$ is used to denote that the entries in the matrix may be sampled from functions. Later in our discussion, we will encounter the notation $w(-n)$, which indicates that we are sampling from the function $f(-x)$ instead of $f(x)$.

4.5 Implementation details

When implementing convolutional neural networks there are two additional things that is important to cover. How the weights are chosen in Equation 24 and how the convolutions are actually computed in practise.

As previously mentioned in the section about convolutions (equation: 9), computing the convolution is usually done with the FFT instead. Which means the actual formula used in implementations is given by:

$$\sigma(Ax + b) = \sigma(\sum_{k_n=1}^{K_n} \mathcal{F}^{-1}(\widehat{S_n(k_n)} \cdot \widehat{w(n)}) + 1 \otimes b) \tag{24}$$

This facilitates understanding of the connection to the Quantum Harmonic Analaysis counterpart of the formula later.

There are several methods for initializing these weights, and interested readers may refer to Chang et al, 2005 [5] for some of the different properties of different initialization schemes. But some natural choices are:

1. Zero initialization, all the weights are equal to 0.
2. Sample from a Gaussian distribution.
3. Sample from a uniform distribution

Zero initialization can easily be shown to be sub-optimal either through numerical experiments such as [12], or by simply noting that a zero initialization with multiple linear layers will be mathematically equivalent to a single layer.

There are various different ways of deciding how to sample from the Gaussian or uniform distribution and, but it generally boils down to what parameters for the distribution to pick. See [44] for a more in-depth overview.

5 Quantum harmonic analysis

This section will introduce several contemporary techniques from Quantum Harmonic Analysis (QHA) that are essential to our discussion. These include:

1. Time modulation and frequency modulation.
2. The convolution of operators and functions.
3. Trace class operators, and why we need to restrict ourselves to a smaller set of operators.
4. The Fourier Wigner and Symplectic Fourier transforms.
5. The properties of operator convolutions
6. A QHA represent of the convolution of a spectogram and a sequence.

The following sections will explore each of these techniques in greater detail, highlighting their importance to the field of QHA and their relevance to our analysis.

5.1 Notation

The section on Quantum Harmonic Analysis will incorporate the material found in [37].

5.1.1 Trace class operators

Similarly to how integrable functions are required for the Fourier transform of functions, we will restrict operators to trace class operators for the corresponding Fourier transform to be well-behaved.

For this to make sense we first have to discuss two essential ideas related to operators: compact operators and their singular value decomposition.

The idea of trace class operators is that we consider only compact operators which have singular values which satisfy some conditions.

The main reason we are interested in compact operators is that they allow for a singular value decomposition. Taken from [50], but using the notation from [39] as it is more consistent with the rest of the thesis:

Definition 5.1 (Signular value decomposition)**.** Let S be a compact operator on $L^2(\mathbb{R}^d$. Then there exists two orthonormal sets $\{b_i\}_{i\in\mathbb{N}}$ and $\{v_i\}_{i\in\mathbb{N}}$ in $L^2(\mathbb{R}^d$ and a sequence $\{s_i(S)\}_{i\in\mathbb{N}}$ of positive numbers with $s_n(S)$ such that S may be expressed as:

$$S = \sum_{n\in\mathbb{N}} s_n(S) b_i \otimes v_i$$

where s_n are the singular values of the operator S.

To define trace class operators we simply impose the same conditions on the singular values as we do on ℓ^1 spaces. See equation 2.

So to define the space of Schatten class operators, denoted by $\mathcal{T}^p$, where p is a positive real number. The operators in this subclass are compact and have singular values that belong to the sequence space ℓ^p. More precisely, we say that S belongs to $\mathcal{T}^p$, if $(s_i)_{i\in\mathbb{N}} \in \ell^p$.

Definition 5.2 (Schatten class $\mathcal{T}^p$)**.**

$$\mathcal{T}^p := \{T \text{ compact } : (s_i)_{i\in\mathbb{N}} \in \ell^p\}$$

This condition ensures that the singular values of S decay sufficiently fast, which is important in the analysis of operators. The Schatten class $\mathcal{T}^1$ is called the space of trace class operators since it allows on to define a trace.

Given an orthonormal basis $\{e_i\}_{i\in\mathbb{N}}$ the trace of a positive operator (definition 2.23) $S \in B(L^2(\mathbb{R}^d))$:

$$tr(S) = \sum_{n\in\mathbb{N}} \langle Se_i, e_i\rangle_{L^2}. \tag{25}$$

This definition is independent of the basis, well-defined, and a bounded linear functional, [54].

Another important Schatten class is $\mathcal{T}^2$, which is known as Hilbert-Schmidt operators [18]. Which is also a Hilbert space under the inner product:

$$\langle T, S\rangle_{\mathcal{T}^\in} := \mathrm{Tr}(ST^*). \tag{26}$$

The spaces $\mathcal{T}^1$, $\mathcal{T}^2$ are the operator analogs for the function spaces of integrable and square-integrable functions.

5.2 Operator convolutions

In order to obtain the necessary formulas for our analysis, we draw upon a similar argument applied to convolutions and the Fourier transform. A detailed explanation can be found in [55], while a shorter description is given here.

At the heart of the argument is the recognition of certain operations that bear a resemblance in both the function and operator settings. By leveraging these similarities and swapping the relevant definitions for convolution and Fourier transform, we arrive at the equivalent forms for operators.

The shared operations include integrals and traces for functions and trace-class operators, respectively, as well as translations and modulations for both. Additionally, we consider the parity operator, which applies equally to functions and operators.

This section will cover how convolutions are generalized to operators, this is done by taking the building blocks of regular convolutions on functions and finding equivalent building blocks for operators. [38]

As convolutions can be made with the following operations:

1. Fold one function over the $y - axis$.
2. Shift one function and weight it by the other one.
3. Integrate.

Our goal will be to modify the convolution given by

$$(f * g)(t) = \int_{\mathbb{C}} f(t)g(t-\tau)d\tau = \int_{\mathbb{C}} f(t-\tau)g(t)d\tau$$

such that it works for operators. To be able to define convolutions of operators we need the equivalent of translates of an operator to generalize the convolution formula to work for operators as well:

Definition 5.3 (Translation of an operator)**.**

$$\alpha_z(A) = \pi(z)A\pi(z)^*.$$

This idea was first put forward by Werner in his seminal work on quantum harmonic analysis, see also Skrettingland's master thesis, which leads to two types of operator convolutions:

Definition 5.4 (Convolution of a function and a trace class operator)**.** Let $f \in L^1(\mathbb{R}^{2d})$ and $S \in \mathcal{T}^1$. Then we define the function-operator convolution by

$$f \star S := S \star f = \int_{\mathbb{R}^{2d}} f(y)a_y(S)dy.$$

Definition 5.5 (Convolution of two trace class operators)**.** Let $S, T \in \mathcal{T}^1$. Then the operator-operator convolution is given by

$$S \star T(z) = \mathrm{Tr}(S\alpha_z(\check{T})$$

These two operations are associative and commutative. We will refer to the proof presented in [38] to demonstrate the associativity and commutativity properties of these convolutions.

First we start by showing commutativity. By utilizing the definitions of α and $\check{T}$, we can expand $S \star T$ as follows.

Proposition 2.

$$S \star T = T \star S$$

Proof:

$$S \star T(z) = \mathrm{Tr}(Sa_z\check{T})) = \mathrm{Tr}(S\pi(z)PTP\pi(z)^*)$$

This allows us to perform simple algebra to simplify the equation as follows:

$$= \mathrm{Tr}(\widetilde{T(a_{-z}S)} = \mathrm{Tr}(Ta_z\check{S}) = T \star S$$

This completes the proof and shows that $T \star S = S \star T$. During these calculations the fact that $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$ is extensively utilized.

For the proof of associativity, please see [38] Proposition 4.4, which proves that three operators $R, S, T \in \mathcal{T}$ satisfy:

Proposition 3.

$$(R \star S) \star T = R \star (S \star T)$$

Outline of proof: First, consider an operator $T_0 \in \mathcal{T}^1$. Then consider the dual action of $\langle T_1 \star (T_2 \star T_3), T_0 \rangle = \langle (T_1 \star T_2) \star T_3), T_0 \rangle$. This commutes by the commutativity of the inner product and shows the expressions define the same thing in the dual space. Finally show that $\mathrm{Tr}(T_0(T_1 \star (T_2 \star T_3)) = \mathrm{Tr}(T_0((T_1 \star T_2) \star T_3)$ which proves they are the same operators.

Furthermore, we also have commutativity and associativity with regular convolution [55]:

$$(f * g) \star S = f * (g \star S) \tag{27}$$

$$f * (S \star T) = (f \star S) \star T \tag{28}$$

5.2.1 Examples of operator convolutions

This section will cover the simplest examples of operator convolutions, namely when the operators are rank one operators.

The first is an example of Definition 5.4 with rank one operators.

Proposition 4. Let $S = f \otimes g$ for $h, g \in L(\mathbb{R}^{2d})$:

$$f \star (S) = f \star (h \otimes g) = \int_{\mathbb{R}^{2d}} f(y) a_y (h \otimes g) dy = \mathcal{A}_f^{h,g} \tag{29}$$

Proof:
This follows from a relatively straightforward calculation [38]:

$$\begin{aligned} f \star S(\psi) &= \iint_{\mathbb{R}^{2d}} f(z)(\alpha_z S)(\psi) dz \\ &= \iint_{\mathbb{R}^{2d}} f(z) \langle \pi(z)^* \psi, g \rangle \pi(z) h dz \\ &= \iint_{\mathbb{R}^{2d}} f(z) V_g \psi \pi(z) h dz = \mathcal{A}_f^{h,g} \end{aligned} \tag{30}$$

Here the definition of convolution is first used, then rewritten as an inner product. Which we then identify as a short time Fourier transform with an extra phase factor, Which by definition is a localization operator.

$$S \star T = \mathrm{Tr}((f \otimes f) \alpha_z(\breve{g} \otimes \breve{g})) = |V_g f|^2 \tag{31}$$

Now for an example use Definition 5.5 to calculate the operator convolution of two rank one operators.

Consider the two rank one operators $S = (f \otimes f)$ and $T = (\breve{g} \otimes \breve{g})$ by writing out the definition of an operator convolution we get:

$$S \star T = \mathrm{Tr}(\breve{S}\alpha_z((\breve{g} \otimes \breve{g})$$

Now use that $S, T \in \mathcal{T}^\infty$ which allows the trace to be written according to equation 25. Let $\{e_i\}_{i\in\mathbb{N}}$ be a basis for $L^2(\mathbb{R}^d)$ then we can rewrite:

$$\begin{aligned}
&= \sum_{i\in\mathbb{N}} \langle \breve{S}\pi(-z)(\breve{g} \otimes \breve{g})\pi(-z)^* e_i, e_i\rangle \\
&= \sum_{i\in\mathbb{N}} \langle \pi(-z)^* e_i, \breve{g}\rangle\langle \breve{S}\pi(-z)\breve{g}, e_i\rangle \\
&= \sum_{i\in\mathbb{N}} \langle e_i, \pi(-z)\breve{g}\rangle\langle \breve{S}\pi(-z)\breve{g}, e_i\rangle \\
&= \langle \breve{S}\pi(-z)\breve{g}, \pi(-z)\breve{g}\rangle \\
&= \langle S\pi(z)g, \pi(z)g\rangle
\end{aligned}$$

This expression is the Berezin transform of S. Using the assumption that $S = (f \otimes f)$ this can be rewritten as:

$$S \star T = (f \otimes f) \star (\breve{g} \otimes \breve{g}) = |V_g(f)|^2, \tag{32}$$

which is an expression that will later be leveraged to rewrite the convolutions in convolutional neural networks.

5.2.2 Fourier transforms for operators

As expected from a "convolution," there is a way to separate them by employing appropriate Fourier transforms. In the case of operator convolutions, symplectic Fourier transform and Fourier-Wigner transform are the appropriate transforms.

Theorems presented in [39] state that for $f \in L^1(\mathbb{R}^{2d})$ and $S, T \in \mathcal{T}$, the following equations hold:

$$\mathcal{F}_\theta(S \star T) = \mathcal{F}_W(S)\mathcal{F}_W(T) \tag{33}$$

$$\mathcal{F}_W(f \star S) = \mathcal{F}_\theta(f)\mathcal{F}_W(S) \tag{34}$$

Operator convolutions are not only inspired by regular convolution, but share their properties when it comes to Fourier transforms. One of the reasons we are interested in the regular Fourier transform is that in the Fourier space convolutions turn into regular products. As shown above there are some similar identities for the operator convolutions, which give the justification for why they are called convolutions. As operator convolutions act nicely with the appropriate Fourier transform, and have some nice associativity and commutativity relations.

5.3 Operator Fourier transform

To take the Fourier transform of an operator, we can follow the same procedure as for functions.

$$\mathcal{F}[f(t)] = \hat{f}(\xi) = \int_{-\infty}^{\infty} f(t)e^{-2\pi i\xi t}dt \tag{35}$$

In the equation above, we modify the function by an exponential and then calculate the integral. However, for trace class operators, we need to use the trace instead of the integral, and we also need to shift the operator around. To achieve this, we use the $\pi(z)$ operator. By replacing the integral with a trace and the translations with π-shifts, we obtain the Fourier-Wigner transform, which is defined as:

Definition 5.6 (Fourier-Wigner transform)**.** $\mathcal{F}_W(S(z)) = e^{i\pi x\cdot\omega}\operatorname{Tr}(\pi(-z)S)$

In the simplest case where S is a rank one operator we get the following:
Example with rank one operator: Assume that $S = (f \otimes g)$. Then

$$\mathcal{F}_W(S(z)) = e^{\pi ix\cdot\omega}\operatorname{Tr}(\pi(-z)(f\otimes g)) = |V_g f|^2 \tag{36}$$

Additionally, we have a Riemann-Lebesgue lemma:

Theorem 5.1. For $S \in \mathcal{T}^1$ we have

$$\mathcal{F}_W(S) \in C_0(\mathbb{R}^{2d}).$$

Where $C_0(\mathbb{R}^{2d})$ denotes the set of functions that vanish at infinity. In addition, there is also a variation of the convolution to multiplication theorems:

Theorem 5.2. Let $f \in L^2(\mathbb{R}^{2d})$ and $S \in \mathcal{T}^1$. Then the following holds:

$$\mathcal{F}_W(f \star S) = \mathcal{F}_\theta(f)\mathcal{F}_W(S) \tag{37}$$

To obtain the inverse Fourier-Wigner transform, we integrate over the entire time-frequency or position-momentum plane using the translation operator $\pi(x,\omega)$, resulting in the following formula:

$$F_W^{-1} = \int F_W(S)\pi(x,w)dxdw,$$

which by Pool's theorem extends to a unitary mapping between $L^2(\mathbb{R}^{2d})$ to the space of Hilbert-Schmidt operators $\mathcal{T}^2$.[49]

5.4 Proposal for Convolutional Neural Networks

The focus of this study is to explore the functionality of convolutions between spectograms and a sequence of numbers, as this is essential in redefining convolutions in CNNs. In this context, the spectogram is denoted as $S = |V_g f|^2$ to highlight how it is an operator. To proceed we write the convolution in the following way as is done in [14]:

$$S * m = |V_g f|^2 * m = \check{m} \star (f \otimes f) \star (\check{g} \otimes \check{g})$$

This mathematical formulation enables the separation of the spectogram into three distinct parts, each of which can be further separated using Fourier transforms. By defining $[\check{m} \star (f \otimes f)]$ as S, and $(\check{g} \otimes \check{g})$ as T, the convolution of two operators can be rewritten as:

$$[\check{m} \star (f \otimes f)] \star (\check{g} \otimes \check{g}) = S \star T$$

Employing the formula described in equation (33), we obtain the following equation:

$$\mathcal{F}_\theta(S \star T) = \mathcal{F}_W(S)\mathcal{F}_W(T) = \mathcal{F}_W(\check{m} \star (f \otimes f))\mathcal{F}_W(\check{g} \otimes \check{g})$$

The term $\mathcal{F}_W(\check{m} \star (f \otimes f))$ can be further broken down using equation (34), which leads to the following:

$$\mathcal{F}_W(\check{m} \star (f \otimes f)) = \mathcal{F}_\theta(\check{m})\mathcal{F}_W(f \otimes f)$$

This decomposition enables us to express the convolution between the spectogram and a sequence of numbers in the following manner:

$$S * m = \mathcal{F}_\theta(\check{m})\mathcal{F}_W(f \otimes f)\mathcal{F}_W(\check{g} \otimes \check{g}) \tag{38}$$

6 Main Theorem

After presenting all of these preliminaries we can present the main theorem that is the topic of this thesis. Here we use the rewriting of convolutions of sequences and spectrograms (equation 38) to rewrite the formula for the CNN (equation 24). To start with a lemma first needs to quickly be derived.

Using the operator convolutions makes it possible to represent the spectrogram in another way. So the goal of this lemma is to show that the following representation of the spectrogram is valid. Following the proof from[14]:

Lemma 6.1.

$$S * m = |V_g f|^2 * m = \check{m} \star (f \otimes f) \star (\check{g} \otimes \check{g}) \tag{39}$$

Proof:

First write out the definition of a spectrogram in terms of an inner product (Definition 3.1), then take the convolution:

$$|V_g f|^2 * m = \langle \int_{z'} V_g f(z') \cdot m(z - z')\pi(z')dz', f \rangle$$

Now we use the trick that $m(z-z') = T_z m(-z)$ to rewrite $m(z-z')$ as $T_z\check{m}$:

$$= \langle T_z \check{m}(g \otimes g)f, f \rangle = \check{m} * \left[(f \otimes f) \star (\check{g} \otimes \check{g})\right](z)$$

Since we know that normal convolution commutes according to equation (28). We can finally rewrite this in the desired form of:

$$S * m = \check{m} \star (f \otimes f) \star (\check{g} \otimes \check{g}) \quad \square$$

Alternate proof: Combine equation 32 and 28 and the proof is trivial $\square$.

Now assume that $S_n(k_n) = |V_{g_n} f_n|^2$ then it is possible to rewrite equation (24) using equation (38) and Lemma 6.1 in the following way:

Theorem 6.2. [Main theorem]

$$\left(\sum_{k=1}^{K_n} S_n(k_n) * w_{n+1}(k_{n+1}, k_n) \right) = \mathcal{F}_\sigma^{-1} \left(\mathcal{F}_\sigma \left[\left(\sum_{k=1}^{K_n} \check{w_k} \right) \star (f_n \otimes f_n) \star (\check{g_n} \otimes \check{g_n}) \right] \right)$$

Despite the convoluted appearance of this formula this is quite similar to how convolutions are already computed using equation 9. But this formulation allows us to seperate $[\sum_{k=1}^{K_n} w_k \star (f_n \otimes f_n)]$ and $T = (g_n \otimes g_n)$. And as T will not change we can cache the value of T to avoid repeated calculations. This is not doable in the previous formalism.

6.1 Proof of main theorem

If we assume the first stage of a CNN is given by a spectrogram. namely:

$$S_0 = F^0(z) = |V_g f(z)|^2 = |\langle f, \pi(z)g, |\rangle^2,$$

Formula for the next layer in a CNN following the notation in both [14] [25]:

$$S_{n+1}(k_{n+1}) = \sigma\left[\left(\sum_{k=1}^{K_n} S_n(k_n) * w_{n+1}(k_{n+1}, k_n)\right) + b \otimes \mathbf{1}\right]$$

Which then gives that

$$S_1(k_1) = \sigma\left[\left(\sum_{k=1}^{K_0} S_0(k_0) * w_1(k_1, k_0)\right) + b \otimes \mathbf{1}\right]$$

Now the parts in the parenthesis is what we will modify.

Then the convolution part in the parenthesis of the second layer is given by:

$$\sum_{k=1}^{K_n} S_0(k_n) * w_1(k_1, k_0) = \sum_{k=1}^{K_n} (F^0 * w_k) = \sum_{k=1}^{K_n} [\widetilde{w_k} \star (f \otimes f)] \star ((\check{g} \otimes \check{g}),$$

Taking the symplectic fourier transform of this then yields:

$$\mathcal{F}_\sigma\left(\left[\sum_{k=1}^{K_n} \widetilde{w_k} \star (f \otimes f)\right] \star ((\check{g} \otimes \check{g})\right) = |V_g(f(z))|^2 \sum_{k=1}^{K_n} \mathcal{F}_\sigma(\widetilde{w_k}) = |V_g(f(z))|^2 \mathcal{F}_\sigma\left(\sum_{k=1}^{K_n} \widetilde{w_k}\right),$$

Here $\widetilde{w_{k_n}}$ consists of swapping the weights of w_k which are sampled from $w_{ii}(k) = f(k)$ to $w_{ii}(k) = f(-k)$.

Since the spectrogram can also be rewritten in terms of the Fourier-Wigner transform we can rewrite it as following:

$$\mathcal{F}_\sigma\left(\sum_{k=1}^{K_n} \widetilde{w_k}\right) \mathcal{F}_W\left[(f \otimes f) \star (\check{g} \otimes \check{g})\right] = \mathcal{F}_W\left[\left(\sum_{k=1}^{K_n} \widetilde{w_k}\right) \star (f \otimes f) \star (\check{g} \otimes \check{g})\right]$$

This then allows for the convolution part of the second layer to be rewritten to the following form:

$$F^1(z, k) = \mathcal{F}_W^{-1}\left(\mathcal{F}_W\left[\left(\sum_{k=1}^{K_n} \check{w_k}\right) \star (f \otimes f) \star (\check{g} \otimes \check{g})\right]\right)$$

Similarly we can then derive the following formula for S_{n+1}:

$$S_{n+1} = \sigma\left(\mathcal{F}_W^{-1}\left(\mathcal{F}_W\left[\left(\sum_{k=1}^{K_n} \check{w_k}\right) \star (f \otimes f) \star (\check{g} \otimes \check{g})\right]\right) + b \otimes \mathbf{1}\right) \square$$

By using this theorem we can rewrite the convolutional part of convolutional neural networks in the following way:

$$\sigma\left[\left(\sum_{k=1}^{K_n} S_n(k_n) * w_{n+1}(k_{n+1}, k_n)\right) + b \otimes \mathbf{1}\right] =$$
$$\sigma\left(\mathcal{F}_\sigma^{-1}\left(\mathcal{F}_\sigma\left[\left(\sum_{k=1}^{K_n} \check{w_k}\right) \star (f \otimes f) \star (\check{g} \otimes \check{g})\right]\right) + b \otimes \mathbf{1}\right)$$

6.2 Consequences of theorem

In normal convolutional neural networks the calculations are taken to the fourier realm when you compute the convolutions anyways, but by leveraging QHA an analysis of what happens during this computation is easier. This allows for some theoretical improvements for how CNN's are computed by using this formalism to come with both theorems for what weights might be sensible to initialize the network with and how to reduce the number of computations.

6.2.1 Speedup calculations

By swapping to this formalism it is possible to see that there are some redundant calculations. As $(\check{g} \otimes \check{g})$ will appear in every node of the convolutional layer it is natural to cache this value to avoid repeated calculations of the same object.

6.2.2 Theory of weight initialization

With this change of perspective some problems that have been difficult to answer about convolutional neural networks might be easier to solve.

If we now look at the different initialization schemes we can show some interesting results at least by considering the term $\mathcal{F}_\sigma(\sum_{k=1}^{K_n} \widetilde{w_k})$.

Zero initalization: is again clearly a bad choice as:

$$\mathcal{F}_\sigma(\sum_{k=1}^{K_n} \widetilde{w_k}) = 0$$

Which plugged into Theorem 26.2 would lead to the operator convolutions also being zero. As the zero initialization has been shown empirically and in theory to be a bad initialization this might suggest that zeros in the expression $\mathcal{F}_\sigma(\sum_{k=1}^{K_n} \widetilde{w_k})$ makes the network worse.

Gaussian initialization: From equation (5) we see that the Fourier transform of a Gaussian is another Gaussian. This means that

$$\mathcal{F}_\sigma(\sum_{k=1}^{K_n} \widetilde{w_k}) > 0$$

.

Uniform initialization: Let $\widetilde{w_k} \sim U(a, b)$.

Then $X = \sum_{k=1}^{N} \widetilde{w_k}$ will follow a Irwin–Hall distribution which has the property that it approaches a normal distribution as $N \to \infty$ [40], this follows by the central limit theorem. This explains why the uniform initialization could also work as it approximates a Gaussian distribution quite well for sufficiently large values of N.

6.2.3 New results related to the network

Firstly the Tauberian theorems can say something about which weights are sensible to choose. This then provides a mathematical framework they understand why choices of different initialization schemes perform better. (This subsection might be removed as I cannot find any concrete sources for this)

And if a weight initialization with non-zero STFT performs better then we can propose a new weight initialization scheme based on the one-sided exponential.

6.2.4 New freedom in the choice of activation function:

By viewing the network in this manner it is easier to see what the activation functions do. This also allows for some new novel approaches to activation functions, and some more insight into what ReLU does for CNN's with audio signals.

Since the ReLU function simply thresholds the values that are too large we can move it inside the Fourier-Wigner transform, and instead do the thresholding on the Fourier side:

$$\sigma\left(\mathcal{F}_W^{-1}\left(\mathcal{F}_W\left[\left(\sum_{k=1}^{K_n} \check{w_k}\right) \star (f \otimes f) \star (\check{g} \otimes \check{g})\right]\right) + b \otimes \mathbf{1}\right) =$$

$$\mathcal{F}_W^{-1}\left(\mathcal{F}_W\left[\sigma\left(\left(\sum_{k=1}^{K_n} \check{w_k}\right) \star (f \otimes f) \star (\check{g} \otimes \check{g})\right)\right] + b \otimes \mathbf{1}\right)$$

This allows for thresholding directly on the spectrograms/Cohen classes which can be easier to interpret.

7 Discretization of theory

Changing from the continuous case to the discrete case requires some adaptations. First, the spectrogram has a lot of redundancy, so it would be nice to have a more efficient representation. Additionally, the calculation requires an integral over the phase space in the calculation of

$$S_0 = F^0(z) = |V_g f(z)|^2 = |\langle f, \pi(z)g\rangle|^2.$$

The issue of integrals over unbounded domains is solved by choosing a window function g with compact support. Whilce the issue of redundancy is solved by using Gabor frames instead.

Additionally, we also need some changes to adopt the different ideas from time-frequency analysis and QHA to be usable as well.

7.1 Lattice

As $\mathbb{R}$ is not usable when dealing with computers we swap our setting to $\mathbb{Z}_n$. We will consider lattices of the form:

$$\Lambda = \mathbb{Z}_a \times \mathbb{Z}_b = \{(x, y) | x \in \mathbb{Z}_a, y \in \mathbb{Z}_b\} \tag{40}$$

The theory on how to choose the lattice constants a, b is a little lacking. Following [53] it is suggested that they at least satisfy the following:

1. **They are divisors of n**: There exists $A, B \in \mathbb{N}$ such that $aA = n = bB$
2. **They have sufficiently many samples :** The different choices of a, b correspond to how frequently the signal will be sampled, which leads to three cases.
   (a) $ab < n$ which is over-sampling, there is more than enough information to reconstruct the signal.
   (b) $ab = n$ which is critical sampling, there is just the necessary information for reconstructing the signal.
   (c) $ab > n$ which is under-sampling, there is not enough information for perfect reconstruction.

We will restrict our attention to the case where $ab \leqslant n$, here we have some theorems from Gabor frame theory that are helpful.

And we will use one of the tricks from Fourier analysis where we extend a function through the use of periodization. This then lets us employ Fourier transforms and do calculations on the Fourier side, then do the Fourier inversion. This allows us to easier deal with convolutions.

Some theorem will require the use of the dual lattice, we follow Skretingland's notation in [54]:

$$\Lambda^\circ = \lambda^\circ \in \mathbb{R}^{2d} : e^{2\pi i \sigma(\lambda^\circ, \lambda)} = 1 \text{ for any } \lambda \in \Lambda \tag{41}$$

Using these definitions we can then also define the size of the lattices as:

$$|\Lambda| = ab$$

$$|\Lambda^\circ| = \frac{1}{|\Lambda|} = \frac{1}{ab}$$

Where a, b are the lattice constants defined in equation 40. It is also possible to define the lattice in more general terms, but for this thesis this is sufficient.

7.2 Periodization

In this study, we shall consider signals from the finite set $\mathbb{Z}_n$, where it is necessary for these signals to have finite length. To facilitate Fourier analysis, we confine our focus to periodic signals. An illustration of periodization is provided in Figure 4. The core idea is that a signal on $\mathbb{Z}_n$ can be extended continuously if the first and last elements of the signal are the same. This allows us to create a signal on $\mathbb{Z}$ satisfying $f(x+n) = f(x), \forall n \in \mathbb{N}$.

As an example consider the function $f(x) = \sin(5x/\pi)$ on the domain $[-5, 5]$. See figure 4 for how it can be extended:

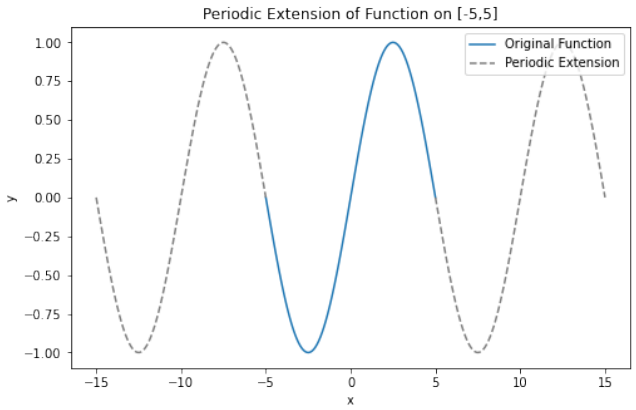


Figure 4: Illustration of periodization

We will also consider 2d-signals which will be found on the lattice:

$$\mathbb{Z}_a \times \mathbb{Z}_b = \{(x, y) | x \in \mathbb{Z}_a, y \in \mathbb{Z}_b\}$$

In our subsequent analyses, the use of quotient groups is deemed necessary, which is an extension of the fundamental concept of periodization that enables the extension of a lattice to become periodic. An illustration of the function defined on a quotient group is presented in Figure 5. The extension of the

function to a larger domain is performed not only in one dimension but in multiple dimensions. As exemplified in the aforementioned figure, the function defined on the domain $[-5, 5] \times [-5, 5]$ is extended to $[-15, 15] \times [-15, 15]$ through a periodic extension. This process of periodic extension provides a more comprehensive analysis of the function over a larger domain. For this kind of periodization we then require the signal to have the same values on the corresponding edges.

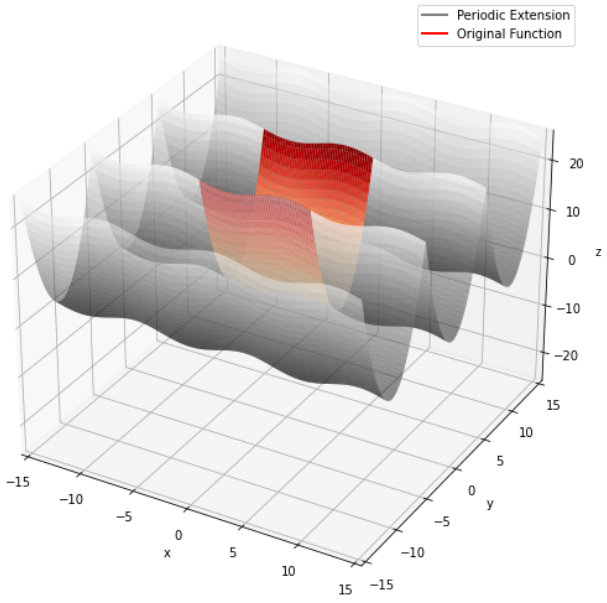


Figure 5: Illustration of a function on a quotient group.

Before introducing the final periodization we need a small digression in order to explain the notation for quotient groups, and functions on quotient groups:

Definition 7.1 (Quotient groups)**.** Let Λ be a lattice then the periodization of a function $f(\lambda)$ on λ that extends it according to figure 5 to $\mathbb{R}$ is denoted $\mathbb{R}/\Lambda$.

Following the notation in [54] anytime a function is defined on the quotient group a dotted variable, such as $\dot{z}$, will be used to highlight this.

For the final form of periodization we need to extend a signal in the following way.

Definition 7.2 (Periodization on lattice)**.** Let Λ be a lattice then the periodization operator is defined on the quotient group of $\mathbb{R}/\Lambda$:

$$P_\Lambda(f)(z^\circ) = |\Lambda| \sum_{\lambda \in \Lambda} f(z + \lambda)$$

Where z° is taken from the dual lattice see equation 41.

7.3 Discrete notation

Just like in the continuous case we need to define some of the building blocks of time-frequency analysis that helps us understand the theory better.

7.3.1 Discrete Fourier analysis

Having obtained a periodic signal, we can extend the definition of Fourier transform to the discrete case as well. The discrete Fourier transform is a widely used tool for analyzing digital signals and has been extensively studied in the literature. [56] [59]

The discrete Fourier transform of a sequence of length N, denoted by $\mathcal{F}(f[k])$, is defined by the following formula:

Definition 7.3 (Discrete fourier)**.**

$$\mathcal{F}^{\Lambda} f[k] = \sum_{n=1}^{N} f[n] e^{\frac{-2\pi i n k}{N}},$$

where $f[n]$ is the nth element of the sequence and k is an integer between 1 and N.

Equation 7.3 provides a way to decompose the original signal into its frequency components. The resulting Fourier coefficients represent the amplitudes and phases of the sinusoidal components that make up the original signal.

Similarly, we define the symplectic Fourier transform as:

Definition 7.4 (Discrete symplectic Fourier series)**.**

$$\mathcal{F}_{\theta}^{\Lambda}(c)(\dot{z}) = \sum_{\lambda \in \Lambda} c(\lambda) e^{2\pi i \theta(\lambda, z)}, \text{for } z \in \mathbb{R}^{2d}$$

This allows us get a nice formula using the Poisson summation formula, see [54] for a proof:

Definition 7.5 (Poisson summation)**.**

$$\frac{1}{|\Lambda|} \sum_{\lambda^{\circ} \in \Lambda^{\circ}} f(z + \lambda^{\circ}) = \sum_{\lambda \in \lambda} \mathcal{F}_{\theta}(f)(\lambda) e^{2\pi i \sigma(\lambda, z)}, \text{ for z } \in \mathbb{R}^{2d}$$

7.4 Discrete quantum harmonic analysis

The section on discrete quantum harmonic analysis will be based on [54]. First, some basic notation is covered, and then the convolution and discrete Fourier transforms are covered until finally a discretized version of the main theorem is derived.

7.4.1 Banach space $\mathcal{B}$

For the discrete case, another space of operators is needed as well: another space of operators which is the Banach space of trace class operators with Weyl symbol in the Feichtinger's algebra. So this section will first explain the Weyl symbol and then define the space $\mathcal{B}$.

To define the Weyl symbol we first need to define the cross-Wigner distribution:

Definition 7.6 (cross-Wigner distribution)**.**

$$W(\xi,\eta)(x,\omega) = \int_{\mathbb{R}^d} \xi(x+\frac{t}{2})\overline{\eta(x-\frac{t}{2})}e^{-2\pi i\omega t}dt, \forall \xi,\eta \in \mathcal{S}_0(\mathbb{R}^d)$$

This allows us to define the Weyl transform L_f of f which is the following operator:

Definition 7.7 (Weyl transform)**.**

$$\langle L_f\eta,\xi\rangle_{\mathcal{S}_0,\mathcal{S}_0'} := \langle f, W(\xi,\eta)\rangle_{\mathcal{S}_0,\mathcal{S}_0'}, \forall \xi,\eta \in \mathcal{S}_0(\mathbb{R}^d)$$

So L_f is an operator from $\mathcal{S}_0(\mathbb{R}^d) \to \mathcal{S}_0'(\mathbb{R}^d)$, where $f \in \mathcal{S}_0'(\mathbb{R}^{2d})$.

Now the definition of a Weyl symbol is the following:

Definition 7.8 (Weyl-symbol)**.** The subscript in the operator L_f is called the Weyl symbol.

This means that for an operator S its corresponding Weyl symbol which we denote by a_S satisfies:

$$L_{a_S} = S$$

This finally allows for a definition of $\mathcal{B}$:

Definition 7.9 (Banach space $\mathcal{B}$)**.** Banach space of trace class operators with Weyl symbol in the Feichtinger's algebra.

$$\mathcal{B} := \left\{S \in \mathcal{T}^1 : a_S \in M^1\right\}.$$

7.4.2 Discrete notation

Now we extend the ideas from previously to the discrete domain. To extend the idea of a convolution of a sequence with an operator we define the following operator, which is identical to the continuous case just restricted to the lattice:

$$c \star_\Lambda S = S \star_\Lambda s = \sum_{\lambda\in\Lambda} c(\lambda)a_\lambda S$$

Similarly, we define the convolution between two operators as the sequence, which is identical to the continuous case but with the α_z shifts restricted to $\lambda \in \Lambda$:

$$S \star_\Lambda T(\lambda) = \mathrm{Tr}(Sa_\lambda(PTP) = \mathrm{Tr}(S\pi(\lambda))PT\pi(-\lambda)^*)$$

7.4.3 Properties of discrete convolutions

Similar to the continuous case there are some theorems for calculating the convolution of Fourier transform of convolutions.[54] As these properties are nearly identical to the continuous case this section is brief.

$$\mathcal{F}_\theta^\Lambda(S \star_\Lambda T) = \frac{1}{|\Lambda|} \sum_{\lambda^\circ \in \Lambda^\circ} \mathcal{F}_W(S)(z+\lambda^\circ)\mathcal{F}_W(T)(z+\lambda^\circ) \tag{42}$$

$$\mathcal{F}_W(c \star_\Lambda S)(z) = \mathcal{F}_\theta^\Lambda(c)(z)\mathcal{F}_W(S)(z) \tag{43}$$

convolutions are also associative and commutative like in the continuous case. Where a direct calculation is sufficient to show associativity. Let $c, d \in \ell^1(\Lambda) S, \in \mathcal{B},$ and $T \in \mathcal{T}$ then:

$$c *_\Lambda (S \star_\Lambda T) = (c \star_\Lambda S) \star T \tag{44}$$

$$(c \star_\lambda d) \star_\Lambda T = c \star_\Lambda (d \star_\Lambda T) \tag{45}$$

Where $\mathcal{B}$ is defined in defintion(7.9). Additionally, since we define the discrete convolutions as a restriction to the lattice of the continuous operator convolutions commutativity follows by definition.

8 Discrete main theorem

Following the same reasoning as in the continuous case it is possible to rewrite the convolution of a spectrogram and a sequence as an operator convolution. Unfortunately, the cost of going from the continuous case to the discrete case is that now a periodization is also needed which is reflected in the P_{Λ° being included in the formula. This then gives the following theorem:

Theorem 8.1 (Main theorem discrete version)**.**

$$\left(\sum_{k=1}^{K_n} S_n(k_n) * w_{n+1}(k_{n+1}, k_n)\right) = \mathcal{F}_\theta^{-1}\left(\mathcal{F}_\theta\left[P_{\Lambda^\circ}\left(\sum_{k=1}^{K_n} \check{w_k}\right) \star (f_n \otimes f_n) \star (\check{g_n} \otimes \check{g_n})\right]\right)$$

Proof: Proceed the same way as in the continuous case. First, assume that the input to the neural network is a spectrogram of the form:

$$S_0 = F^0(z) = |V_g f(z)|^2 = |\langle f, \pi(z)g, |\rangle^2 = (f \otimes f) \star (\check{g} \otimes \check{g}),$$

Following the continuous derivation the spectrogram is rewritten:

$$F * w_k = \widetilde{w_k} \star_\Lambda (f \otimes f) \star_\Lambda (\check{g} \otimes \check{g})$$

Take the discrete symplectic Fourier transform, then prepare for some rewriting:

$$\mathcal{F}_\theta^\Lambda(\widetilde{w_k} \star_\Lambda (f \otimes f) \star_\Lambda (\check{g} \otimes \check{g}))$$

Note that $\widetilde{w_k} \star_\Lambda (f \otimes f)$ is an operator which means this can be rewritten using equation 42:

$$\mathcal{F}_\theta^\Lambda([\widetilde{w_k} \star_\Lambda (f \otimes f)] \star_\Lambda (\breve{g} \otimes \breve{g})) = \frac{1}{\Lambda} \sum_{\lambda^\circ \in \Lambda^\circ} \mathcal{F}_W([\widetilde{w_k} \star_\Lambda (f \otimes f)])(z+\lambda^\circ) \mathcal{F}_W(T)(z+\lambda^\circ)$$

$$= \frac{1}{\Lambda} \sum_{\lambda^\circ \in \Lambda^\circ} \mathcal{F}_\theta^\Lambda(\widetilde{w_k})(\dot{z}) \mathcal{F}_W([\widetilde{w_k} \star_\Lambda (f \otimes f)])(z+\lambda^\circ) \mathcal{F}_W(T)(z+\lambda^\circ)$$

$$= P_{\Lambda^\circ}(\mathcal{F}_\theta^\Lambda(\widetilde{w_k}) \mathcal{F}_W(f \otimes f) \mathcal{F}_W(\breve{g} \otimes \breve{g})(\dot{z})$$

This gives a similar result as equation 38, but now we have to add a periodization.

$$\mathcal{F}_\theta^{\Lambda-1} \mathcal{F}_\theta^\Lambda (F * w_k) = P_{\Lambda^\circ}(\mathcal{F}_\theta^\Lambda(\widetilde{w_k}) \mathcal{F}_W(f \otimes f) \mathcal{F}_W(\breve{g} \otimes \breve{g})(\dot{z}) \quad \square$$

9 Conclusion

This thesis ventured into the promising domain of applying quantum harmonic analysis to optimize the efficiency of convolutional neural networks (CNNs), particularly in the realm of audio processing. Traditional methods of computing convolutions in CNNs can pose significant computational challenges. One strategy to mitigate these issues is to employ the Fourier transform, which facilitates the conversion of convolutions into pointwise multiplication in the frequency domain.

This technique has already been integrated into CNNs through the application of the convolution theorem for Fourier transforms, represented as:

$$(S * m) = \mathcal{F}^{-1}\mathcal{F}(S * m) = \mathcal{F}^{-1}(\hat{S} \cdot \hat{m})$$

However, the traditional convolution theorem does not consider that S is a spectrogram. Consequently, this thesis presents the possibility of exploiting the structure of spectrograms to develop new insightful formulas, synthesized from the theory of quantum harmonic analysis and time-frequency analysis. By capitalizing on the structure of a spectrogram, the convolution of a spectrogram and a mask can be deconstructed into operator convolutions that involve different parts in a more elegant manner, as exhibited in the following formulas:

$$S * m = \check{m} \star (f \otimes f) \star (\check{g} \otimes \check{g}) = \mathcal{F}_\theta^{-1}\mathcal{F}_\theta(\check{m} \star (f \otimes f) \star (\check{g} \otimes \check{g}))$$

This approach holds potential benefits for theoretical exploration of CNNs, specifically by offering insights into the type of weights that may be advantageous to select. Although empirical evidence suggests that Gaussian weight initialization performs well, the framework introduced in this thesis provides a theoretical basis that supports this observation.

Moreover, the thesis proposes how the same principle can be adapted to express convolutions in a discrete setting, where the signals are confined to a lattice Λ. This adjustment accommodates the constraints of computer programs incapable of representing continuous functions. This is achieved by recasting the convolution of a spectrogram using the ensuing formula:

$$\mathcal{F}_\theta^{\Lambda-1} P_{\Lambda^\circ}(\mathcal{F}_\theta^\Lambda(\widetilde{w_k})\mathcal{F}_W(f \otimes f)\mathcal{F}_W(\check{g} \otimes \check{g})(\dot{z})$$

In sum, this thesis contributes to the intersection of Quantum Harmonic Analysis and CNNs, introducing theoretical explanations and practical adaptations that could potentially enhance the performance and efficiency of CNNs in various applications.

References

[1] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. In *2012 IEEE international conference on Acoustics, speech and signal processing (ICASSP)*, pages 4277–4280. IEEE, 2012.

[2] E. Berge. A brief introduction to the feichtinger algebra $\mathbf{S}_0(\mathbb{R})$, 2021.

[3] D. Bhatt, C. Patel, H. Talsania, J. Patel, R. Vaghela, S. Pandya, K. Modi, and H. Ghayvat. Cnn variants for computer vision: history, architecture, application, challenges and future scope. *Electronics*, 10(20):2470, 2021.

[4] K. D. Bierstedt and J. Bonet. Some aspects of the modern theory of fréchet spaces. *RACSAM*, 97(2):159–188, 2003.

[5] W. Boulila, M. Driss, M. Al-Sarem, F. Saeed, and M. Krichen. Weight initialization techniques for deep learning algorithms in remote sensing: Recent trends and future perspectives, 2021.

[6] Y. Chen. Convolutional neural network for sentence classification. Master's thesis, University of Waterloo, 2015.

[7] K. Chitsaz, M. Hajabdollahi, N. Karimi, S. Samavi, and S. Shirani. Acceleration of convolutional neural network using fft-based split convolutions. *arXiv preprint arXiv:2003.12621*, 2020.

[8] C. K. Chui. *An introduction to wavelets*, volume 1. Academic press, 1992.

[9] L. Cohen. Time-frequency distributions-a review. *Proceedings of the IEEE*, 77(7):941–981, 1989.

[10] S. Das and J. Sarkar. Invariant subspaces of analytic perturbations, 2021.

[11] M. A. De Gosson. *Symplectic methods in harmonic analysis and in mathematical physics*, volume 7. Springer Science & Business Media, 2011.

[12] C. K. Dewa et al. Suitable cnn weight initialization and activation function for javanese vowels classification. *Procedia computer science*, 144:124–132, 2018.

[13] M. Doerfler, T. Grill, R. Bammer, and A. Flexer. Basic filters for convolutional neural networks applied to music: Training or design?, 2017. URL `https://arxiv.org/abs/1709.02291`.

[14] M. Dörfler, F. Luef, and E. Skrettingland. Local structure and effective dimensionality of time series data sets, 2021. URL `https://arxiv.org/abs/2111.02153`.

[15] H. G. Feichtinger. On a new segal algebra. *Monatshefte für Mathematik*, 92:269–289, 1981.

[16] H. G. Feichtinger and G. Narimani. Fourier multipliers of classical modulation spaces. *Applied and Computational Harmonic Analysis*, 21(3):349–359, 2006.

[17] D. Gabor. Theory of communication. part 1: The analysis of information. *Journal of the Institution of Electrical Engineers-part III: radio and communication engineering*, 93(26):429–441, 1946.

[18] I. Gohberg, S. Goldberg, M. A. Kaashoek, I. Gohberg, S. Goldberg, and M. A. Kaashoek. Hilbert-schmidt operators. *Classes of Linear Operators Vol. I*, pages 138–147, 1990.

[19] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[20] A. Grimshaw. Mishearings, misunderstandings, and other nonsuccesses in talk: A plea for redress of speaker-oriented bias. *Sociological Inquiry*, 50: 31 – 74, 01 2007. doi: 10.1111/j.1475-682X.1980.tb00016.x.

[21] K. Gröchenig. *Foundations of time-frequency analysis*. Springer Science & Business Media, 2001.

[22] P. Hall, D. Marshall, and R. Martin. Adding and subtracting eigenspaces with eigenvalue decomposition and singular value decomposition. *Image and Vision Computing*, 20(13-14):1009–1016, 2002.

[23] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu. Ghostnet: More features from cheap operations. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1580–1589, 2020.

[24] W. Hao, W. Yizhou, L. Yaqin, and S. Zhili. The role of activation function in cnn. In *2020 2nd International Conference on Information Technology and Computer Application (ITCA)*, pages 429–432. IEEE, 2020.

[25] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009.

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015.

[27] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[28] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017.

[29] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019.

[30] M. S. Jakobsen. On a (no longer) new segal algebra: a review of the feichtinger algebra. *Journal of Fourier Analysis and Applications*, 24(6): 1579–1660, 2018.

[31] K. Janocha and W. M. Czarnecki. On loss functions for deep neural networks in classification. *arXiv preprint arXiv:1702.05659*, 2017.

[32] Y. Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, Oct. 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1181. URL `https://aclanthology.org/D14-1181`.

[33] D. Kolba and T. Parks. A prime factor fft algorithm using high-speed convolution. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 25(4):281–294, 1977.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6): 84–90, 2017.

[35] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791.

[36] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[37] F. Luef and E. Skrettingland. Mixed-state localization operators: Cohen's class and trace class operators, 2018. URL `https://arxiv.org/abs/1802.02435`.

[38] F. Luef and E. Skrettingland. Convolutions for localization operators. *Journal de Mathématiques Pures et Appliquées*, 118:288–316, 2018.

[39] F. Luef and E. Skrettingland. Mixed-state localization operators: Cohen's class and trace class operators. *Journal of Fourier Analysis and Applications*, 25:2064–2108, 2019.

[40] J. E. Marengo, D. L. Farnsworth, and L. Stefanic. A geometric derivation of the irwin-hall distribution. *International Journal of Mathematics and Mathematical Sciences*, 2017, 2017.

[41] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[42] Mustaqeem and S. Kwon. A cnn-assisted enhanced audio signal processing for speech emotion recognition. *Sensors*, 20(1):183, 2019.

[43] S. Narayan. The generalized sigmoid activation function: Competitive supervised learning. *Information sciences*, 99(1-2):69–82, 1997.

[44] M. V. Narkhede, P. P. Bartakke, and M. S. Sutaone. A review on weight initialization strategies for neural networks. *Artificial intelligence review*, 55(1):291–322, 2022.

[45] M. Olazaran. A sociological study of the official history of the perceptrons controversy. *Social Studies of Science*, 26(3):611–659, 1996.

[46] A. V. Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.

[47] G. Palm. Warren mcculloch and walter pitts: A logical. In *Brain Theory: Proceedings of the First Trieste Meeting on Brain Theory October 1-4, 1984*, page 229. Springer, 1986.

[48] K. Pavel and S. David. *Algorithms for efficient computation of convolution*, volume 8. ch, 2013.

[49] J. C. Pool. Mathematical aspects of the weyl correspondence. *Journal of Mathematical Physics*, 7(1):66–76, 1966.

[50] M. Reed. Methods of modern mathematical physics i. *Functional analysis*, 1972.

[51] K. Roose. The brilliance and weirdness of chatgpt. *The New York Times*, 2022.

[52] F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[53] K. Schnass and H. G. Feichtinger. *Gabor multipliers a self-contained survey*. na, 2004.

[54] E. Skrettingland. Quantum harmonic analysis on lattices and gabor multipliers. *Journal of Fourier Analysis and Applications*, 26:1–37, 2020.

[55] E. Skrettingland. Time-frequency analysis meets quantum harmonic analysis. 2021.

[56] Z. Wang. Fast algorithms for the discrete w transform and for the discrete fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(4):803–816, 1984.

[57] P. J. Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[58] R. Werner. Quantum harmonic analysis on phase space. *Journal of mathematical physics*, 25(5):1404–1411, 1984.

[59] S. Winograd. On computing the discrete fourier transform. *Mathematics of computation*, 32(141):175–199, 1978.

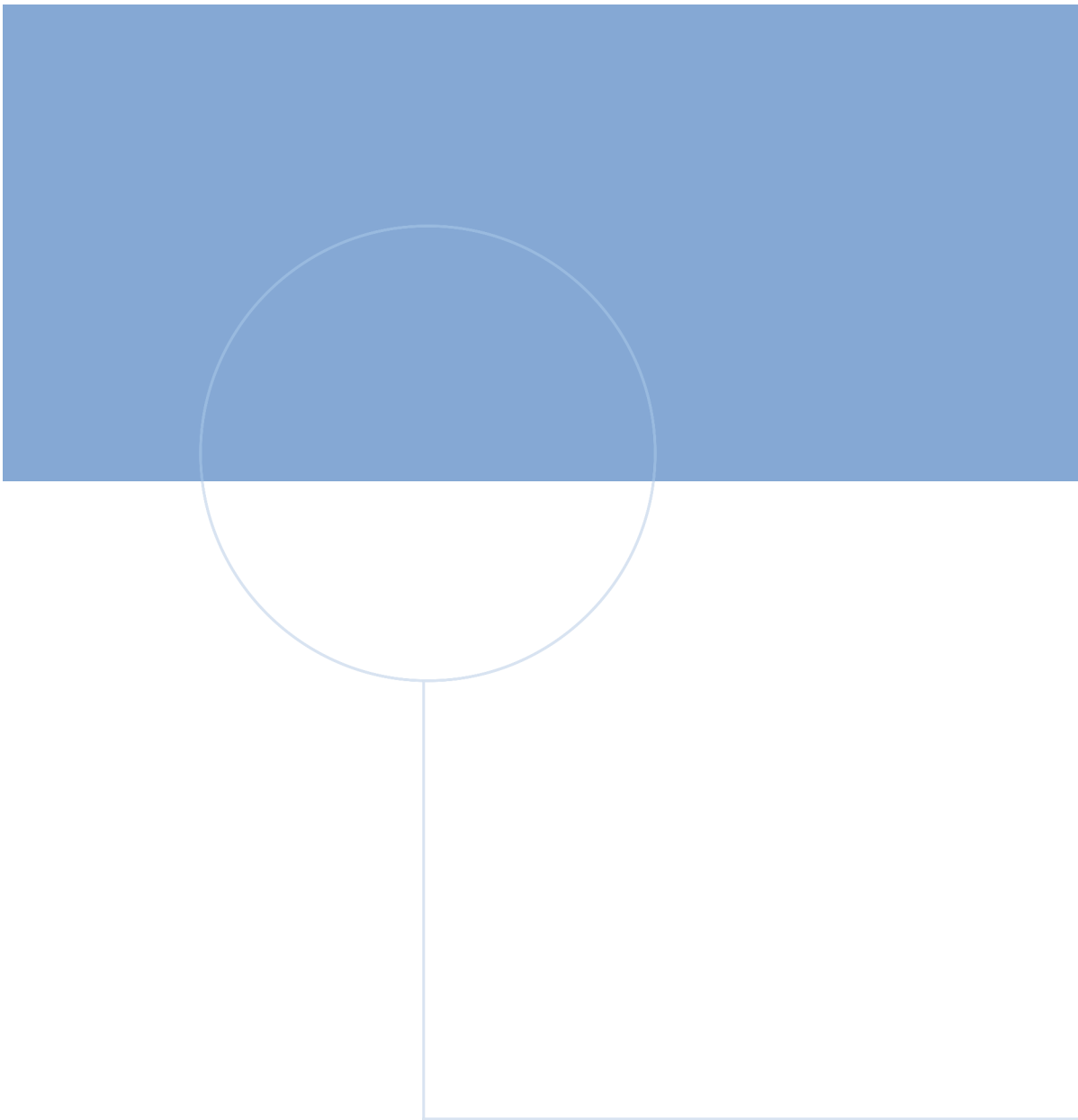
[60] Y.-C. Wong. *Schwartz spaces, nuclear spaces and tensor products*, volume 726. Springer, 2006.

[61] W. Yin, K. Kann, M. Yu, and H. Schütze. Comparative study of cnn and rnn for natural language processing. *arXiv preprint arXiv:1702.01923*, 2017.

[62] M. D. Zeiler, M. Ranzato, R. Monga, M. Mao, K. Yang, Q. V. Le, P. Nguyen, A. Senior, V. Vanhoucke, J. Dean, et al. On rectified linear units for speech processing. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3517–3521. IEEE, 2013.

[63] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

[64] Y. Zhao, G. Wang, C. Tang, C. Luo, W. Zeng, and Z.-J. Zha. A battle of network structures: An empirical study of cnn, transformer, and mlp. *arXiv preprint arXiv:2108.13002*, 2021.



NTNU

Kunnskap for en bedre verden