ABSTRACT

Title of dissertation: FEATURE EXTRACTION IN
IMAGE PROCESSING AND
DEEP LEARNING

Yiran Li
Doctor of Philosophy, 2018

Dissertation directed by: Professor Wojciech Czaja
Department of Mathematics

This thesis develops theoretical analysis of the approximation properties of neural networks, and algorithms to extract useful features of images in fields of deep learning, quantum energy regression and cancer image analysis. The separate applications are connected by using representation systems in harmonic analysis; we focus on deriving proper representations of data using Gabor transform in this thesis. A novel neural network with proven approximation properties dependent on its size is developed using Gabor system. In quantum energy regression, invariant representation of chemical molecules using electron densities is obtained based on the Gabor transform. Additionally, we dig into pooling functions, the feature extractor in deep neural networks, and develop a novel pooling strategy originated from the maximal function with stability property and stable performance. Anisotropic representation of data using the Shearlet transform is also explored in its ability to detect regions of interests of nuclei in cancer images.

FEATURE EXTRACTION IN IMAGE PROCESSING AND DEEP LEARNING

by

Yiran Li

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2018

Advisory Committee:
Dr. Wojciech Czaja, Chair/Advisor
Dr. John Benedetto
Dr. Radu Balan
Dr. Kasso Okoudjou
Dr. Ilya Ryzhov
Dr. Stephen Lockett

Dedication

This thesis is dedicated to my parents. Thank you for all the support.

Acknowledgments

First and foremost I would like to thank my advisor, Dr. Wojciech Czaja, for introducing me to the world of interesting pure and applied mathematical research. I started with the scientific computation track at University of Maryland, and I completed my one-year long project course under the supervision of Wojtek. It was very fulfilling experience meeting and discussing mathematical ways to approach applied problem with Wojtek and so I continued to work with him on new projects later on and he became my advisor. Wojtek has always been very helpful and insightful in leading me on the path of research. He always points out new directions when research is stagnant, and the conversations we had in our regular meetings stimulate my passion and interest in conducting research in applied harmonic analysis. I would not have completed my five years here at the University of Maryland without his guidance and support.

I would also like to thank Dr. John Benedetto for bringing me to the Norbert Wiener Center for harmonic analysis and applications and for teaching me fascinating lectures on harmonic analysis and wavelets. The two courses I took with John broadened my view from real analysis and extend to the world of harmonic analysis and all of its potential applications, and established a strong basis for me to conduct my research. I appreciate all the help that John has offered at various stages of my Ph.D. years.

I would also like to thank Dr. Kasso Okoudjou and Dr. Radu Balan for the inspirations they give me on the path of my education through the the seminars and talks at the Norbert Wiener Center. I appreciate their help and support.

During the summer of my third year, I encountered a great opportunity to work with Dr. Stephen Lockett from National Institute of Health, on the problem of cancer image analysis. I would like to thank Stephen Lockett for giving me this opportunity to collaborate with him and with Robert Kinders, and I really appreciate his help and patience in filling me with useful biological and methodological insights and guidance on my research. I would also like to thank Dr. Konstantina Trivisa for bringing this opportunity to me.

I would like to thank Dr. Ilya Ryzhov for agreeing to serve on my committee and for teaching me basics in probabilistic models. The knowledge I learned vividly stays in my mind and the modeling way of thinking facilitates my research progress.

I would like to thank Dr. Maria Cameron for her help and her guidance in the summer project in my first year. She has brought invaluable help for my education.

I would like to thank all members of the Norbert Wiener Center for Harmonic Analysis and Its Applications. Thanks to Mike Pekala and to Weilin Li, for the collaboration on maximal function pooling we have done and for sharing interesting mathematical perspectives on this subject; thanks to Matt Guay, who has organized illuminating RITs on deep learning for many semesters; thanks to Dongmian Zou, and Shujie Kang, for the mathematical discussions on research that helped us think; and thanks to Franck Ndjakou Njeunje, Zeyad Emam, and Chenzhi Zhao for being supportive office mates.

During my years of Ph.D. life in college park, I have met friends that make my life much more enjoyable. I'd like to thank Luyu Sun, Zhang Zhang, Chen Qian, Jinhang Xue, for the fun time we spent together.

I would like to thank my friends that I have known for so long and who have accompanied me along the path of my Ph.D.. My gratitude for Cheng Peng, Yang Song and Zhe Wang. Thank you for being there.

I owe my deepest gratitude to my family. I convey my deepest sorrow for my grandmother Bide and my grandfather Maoqing, who have accompanied me from childhood, but whose last days I wish I had been by their side. I would also like to thank my niece Wanyi for coming into this world. Last but most importantly, I would like to thank my father Min and my mother Mingyu. My gratitude for their support from the first day of my life all the way to my Ph.D. is beyond words. I thank my parents for their faith in my ability and for all the love and support they have given to me.

Contents

List of Tables

List of Figures

List of Abbreviations

$\mathbb{R}$	The set of real numbers
$\mathbb{C}$	The set of complex numbers
$\langle \cdot, \cdot \rangle$	The inner product
$\text{supp} f$	The support of a function f
$\hat{f}, F$	The Fourier transform of f
$\overline{z}$	The complex conjugate of a complex number z
$\\| \cdot \\|_p$	The L^p norm
$\text{ess sup} f$	The essential supremum of f

STFT	Short time Fourier transform
DNN	Deep neural network
MLP	Multilayer perceptron
ReLU	Rectified linear unit
CNN	Convolutional neural network
DCP	Deep coding problem
DCPP	Deep coding problem with pooling
SVM	Support vector machine
DFT	Density functional theory
EMT	Epithelial-mesenchymal transition
ROI	Region of interest
DAPI	6-diamidino-2-phenylindole

Chapter 1: Summary of Results

This thesis incorporates classical harmonic analysis ideas in theoretical analysis of deep learning, and presents new results in theory of deep learning and applied harmonic analysis. It focuses on the design of feature extractors from harmonic analysis and property analysis of such feature extractors in fields of deep learning, quantum energy regression and biomedical imaging analysis. The harmonic analysis detailed in this thesis focuses on the time-frequency representation systems that contain local frequency information. In Chapter 2, a survey of classical harmonic analysis emphasizing Fourier analysis and time-frequency analysis is presented, as well as a brief introduction to deep learning, whose theoretical study is of interest to many mathematicians. In particular, we review theoretical properties of Gabor frames as a time-frequency representation of signals and introduce feedforward neural networks, the most common network structure in deep learning.

In Chapter 3 we study the approximation properties of neural networks. Neural networks extract useful features of the input functions and learn representations of functions by adjusting weights via training on extensive amount of data. The performance of a neural network on tasks such as classification depends heavily on its ability to effectively represent input functions. Thus the degree to which a neural network can approximate functions also links tightly to its performance. The study of the relation between error rate of approximation and the size of the network can be beneficial to estimation of training time. We design a novel type of neural network inspired by Gabor frames and prove its theoretical approximation rate to functions based on the network topology. The theory presented in this Chapter enriches the study of provable approximation rate of neural networks by introducing a new neural network that explores frequency information of input signal. It has promising application values with the development of training algorithms for complex valued neural network, and it achieves better accuracy when input function has explicit compact support.

In Chapter 4, we continue our endeavor to analyze deep neural networks under the perspective of convolutional sparse coding introduced by M. Elad, et al. Pooling is a common feature extraction strategy adopted in convolutional neural networks. It reduces dimensionality of input features and mitigate the problem of overfitting in training of neural network. Inspired by the maximal function, a classical concept in harmonic analysis, we deign the maximal function pooling, or the maxfun pooling, and analyze the stability of neural networks with maxfun pooling when noise is present as well as its performance in classification compared with existing state-of-the-art pooling strategies. The results in this Chapter show that the maxfun pooling preserves stability of the neural network and it outperforms state-of-the-art pooling strategies in certain classification tasks. The maxfun pooling demonstrates intriguing theoretical properties with prominent application values.

In Chapter 5, we dive into the problem of invariant representation of molecules and design an invariant feature extractor for quantum energy regression. Driven by useful applications such as synthesis of new material, machine learning strategies have been exploited to reduce the costs of computing ground state energy of chemical molecules. We present Gabor invariant transform which produces a set of dictionaries representing each molecule in a translation and rotation invariant fashion. The set of dictionaries is selected via machine learning algorithms using cross validation to achieve best performance. The Gabor invariant representation demonstrates invariant properties necessary for molecule representation, competes with state-of-the-art methods for planar molecules, and has the advantage of being extendable to represent high dimensional data.

In Chapter 6, we dig into the problem of detecting regions of two different types of cell nuclei, mesenchymal and epithelial, in cancer image analysis. The detection of the location of mesenchymal cells is crucial in the study of tumor growth and its drug treatment. We design an algorithm which exploits the directional information presented in cell shapes and in their alignments using Shearlet transform. The Shearlet transform produces anisotropic features and is sensitive in detecting edge-like features in input data. We develop the Shearlet max difference thresholding method, which outperforms benchmark algorithms using wavelets and shearlets, and demonstrates its potential extension to detecting regions of interest in 3D image data.

Chapter 2: Mathematical Preliminaries

In this Chapter we introduce the mathematical preliminaries which later Chapters are built upon. In particular, we introduce the notion of time-frequency analysis and the notion of deep learning. Many mathematical ideas from the later Chapters can be retrieved from the mathematical foundations mentioned in this Chapter.

2.1 Time-Frequency Analysis

In this section we introduce motivations behind time-frequency analysis and some of the important results achieved in this field. We first introduce several notations. The notation for inner product in $\mathbb{R}^d$ is $x \cdot \omega = \sum_{i=1}^{d} x_i \omega_i$. For $x = (x_1, x_2, ..., x_d) \in \mathbb{R}^d$, and for $1 \leq p < \infty$, we use the notation

$$\|f\|_p = \left(\int_{\mathbb{R}^d} |f(x)|^p dx \right)^{1/p} \tag{2.1}$$

for L^p norm of f, and $L^p(\mathbb{R}^d)$ is the Banach space of all measurable function f that have finite L^p norm. Given a measurable function $f : X \to \mathbb{R}$ defined under measure μ, the essential supremum (ess sup) is defined as the smallest α such that the set $(x : f(x) > \alpha)$ has measure zero. When $p = \infty$, $\|f\|_\infty = \operatorname{ess\,sup}_{x \in \mathbb{R}^d} |f(x)|$. When

$p = 2$, for $f, g \in L^2(\mathbb{R}^d)$, the inner product is defined by

$$\langle f, g \rangle = \int_{\mathbb{R}^d} f(x)\overline{g(x)}dx \tag{2.2}$$

and $L^2(\mathbb{R}^d)$ is a Hilbert space.

2.1.1 Basic Fourier Analysis

Fourier analysis focuses on the analysis of Fourier transform of functions and the relation between the function and its transform. The Fourier transform is a classical subject of study in harmonic analysis, and it has served as a powerful tool for computations in fields of engineering and physical sciences. The first use of Fourier methods was in in Lagrange's study of partial differential equations modeling string vibration [17].

Let f be a function defined for $x \in \mathbb{R}^d$. The Fourier transform of f is naturally defined on $L^1(\mathbb{R}^d)$.

Definition 2.1. *The Fourier transform of $f \in L^1(\mathbb{R}^d)$ is defined as*

$$F(\omega) = \int_{\mathbb{R}^d} f(x)e^{-2\pi i x\cdot\omega}dx,\ \omega \in \hat{\mathbb{R}}^d. \tag{2.3}$$

Notationally, we write the pairing between the function f and F in the following ways: $f \longleftrightarrow F$, $F = \hat{f}$.

The Fourier transform can be extended to $L^2(\mathbb{R}^d)$ naturally. A major result about $L^2(\mathbb{R})$ is the following theorem [10].

Theorem 2.2. *(Plancherel [10]) There is a unique linear bijection $\mathcal{F} : L^2(\mathbb{R}) \longrightarrow L^2(\hat{\mathbb{R}}^d)$ with properties:*

- $\forall f \in L^2(\mathbb{R}^d), \quad \|f\|_2 = \|Ff\|_2;$

- $\forall f \in L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ *and* $\forall \omega \in \hat{\mathbb{R}}^d, \quad \hat{f}(\omega) = (Ff)(\omega);$

- $\forall f \in L^2(\mathbb{R}^d), \quad \exists \{f_n : n = 1, ...\} \subseteq L^1(\mathbb{R}^d) \cap L^2(\mathbb{R}^d)$ *for which* $\lim_{n\to\infty} \|f_n - f\|_2 = 0$ *and* $\lim_{n\to\infty} \|\hat{f}_n - Ff\|_2 = 0.$

The Fourier transform $\hat{f}$ and f gives equivalent representation of the same function f in different domains, the frequency domain and the time domain. The inverse of Fourier transform is defined by the inversion formula [10].

Theorem 2.3. *Let $f \in L^1(\mathbb{R}^d)$ and let $\hat{f} = F$. The Fourier transform inversion formula is*

$$f(x) = \int_{\mathbb{R}^d} F(\omega) e^{2\pi i x \cdot \omega} d\omega. \tag{2.4}$$

Given the pairing $f \longleftrightarrow F$, we write $F(\omega) = A(\omega)e^{i\phi(\omega)}$, and one may think of a signal f as a "sum" of exponentials $e^{2\pi i x\cdot\omega}$ [10]

$$\text{``}f(x) = \sum_{\omega} A(\omega) e^{i\phi(\omega)} e^{2\pi i x\cdot\omega}\text{''} \tag{2.5}$$

with complex coefficients $A(\omega)e^{i\phi(\omega)}$. $A(\omega)$ is the *amplitude* and $\phi(\omega)$ is the *phase angle* of $F(\omega)$. The amplitude spread $|A(\omega)|^2$ can be viewed as the amount of energy of f in frequency band about a small neighborhood of ω.

Fourier transform serves as a useful computational tool based on its analytical properties. We list some of the properties that are relevant in Chapter 3.

Theorem 2.4. *(Analytic Properties of Fourier Transforms [10]) Let $f \in L^1(\mathbb{R})$, and consider the paring $f \longleftrightarrow F$, where $F(\omega)$ is the Fourier transform of $f(x)$.*

- *Boundedness. For each* $\omega \in \hat{\mathbb{R}}$, $|F(\omega)| \leq \|f\|_{L^1(\mathbb{R})}$.

- *Riemann-Lebesgue Lemma.* $\lim_{|\omega|\to\infty} F(\omega) = 0$.

- *Time Differentiation. Suppose that* $f^{(d)}$, $d \geq 1$, *exists everywhere and that* $f^{(d)} \in L^1(\mathbb{R})$. *Assume*

$$f(\pm\infty) = ... = f^{(d-1)}(\pm\infty) = 0, \tag{2.6}$$

where $f(\pm\infty) = 0$ *indicates that* $\lim_{t\to\infty} f(t) = 0$ *and* $\lim_{t\to-\infty} f(t) = 0$. *Then*

$$f^{(d)} \longleftrightarrow (2\pi i\omega)^d F(\omega). \tag{2.7}$$

We also introduce the notion of convolution. Let $f, g \in L^1(\mathbb{R})$. The *convolution* of f and g, denoted by $f * g$, is [10]

$$f * g(t) = \int f(t-u)g(u)du = \int f(u)g(t-u)du. \tag{2.8}$$

We obtain the theorem regarding Fourier transform of convolution [10].

Theorem 2.5. *Let* f, $g \in L^1(\mathbb{R})$, *with corresponding Fourier pairs* $f \longleftrightarrow F$ *and* $g \longleftrightarrow G$. *Then* $f * g \in L^1(\mathbb{R})$ *and* $(f*g)\hat{} = \hat{f}\hat{g}$.

Although Fourier transform has served as a useful tool for various problems, it has some drawbacks. When analyzing signals, the goal is often to obtain both temporal and frequency information instantaneously. An analogy is to think of the signal f as a piece of music [32], [49]. At any time $x \in \mathbb{R}$, we may measure the amplitude of a song as $f(x)$, and we may gather rhythmic information of the music. But we would have trouble identifying the key or the melodic information of the song. If we look at $\hat{f}$, by looking at its dominant frequency, we may be able to identify its key, but none of the temporal information can be recovered. The goal is to represent f in a way that simultaneous information about time and frequency can be present, like how our hearing perceives music.

Due to the mathematical nature of the Fourier transform, a collection of inequalities named uncertainty principles prevents the ideal construction of instantaneous frequency representation. In a qualitative form, the uncertainty principle states that:

> *A function f and its Fourier transform $\hat{f}$ cannot be supported on arbitrarily small sets.*

In other words, in order to calculate instantaneous frequency at any x, we need to take the Fourier transform of f multiplied by some function g of small support around x. By the uncertainty principle, the support of $\widehat{f \cdot g}$ cannot be small and thus it does not make sense to speak of instantaneous frequency at x. We state the classical uncertainty principle in dimension $d = 1$.

Theorem 2.6. *(Heisenberg-Pauli-Weyl inequality [49]) If $f \in L^2(\mathbb{R})$ and $a, b \in \mathbb{R}$ are arbitrary, then*

$$\left(\int_{-\infty}^{\infty} (x-a)^2 |f(x)|^2 dx\right)^{1/2} \left(\int_{-\infty}^{\infty} (\omega-b)^2 |\hat{f}(\omega)|^2 d\omega\right)^{1/2} \geq \frac{1}{4\pi} \|f\|_2^2. \tag{2.9}$$

Equality in 2.9 holds if and only if f is a multiple of $e^{2\pi i b(x-a)} \cdot e^{-\pi(x-a)^2/c}$ for some $a, b \in \mathbb{R}$ and $c > 0$.

The factors in the above theorem measure the degree of localization of f around a and of $\hat{f}$ around b, respectively. The minimal support in both time and frequency domains is achieved when f is translation and modulation of a Gaussian function. Note that this result also holds for self-adjoint operators on a Hilbert space. Let the commutator of two linear operators A, B, be denoted by:

$$[A, B] = AB - BA. \tag{2.10}$$

Theorem 2.7. *Let A and B be (possibly unbounded) self-adjoint operators on a Hilbert space $\mathcal{H}$. Then from [49],*

$$\|(A - a)f\|\|(B - b)f\| \geq \frac{1}{2}|\langle [A, B]f, f\rangle| \tag{2.11}$$

for all $a, b \in \mathbb{R}$ and for all f in the domain of AB and BA. Equality holds if and only if $(A - a)f = ic(B - b)f$ for some $c \in \mathbb{R}$.

The solution f of Equation 2.9 when equality is reached in Theorem 2.6 provides us with some intuition about proper representation of signals with both temporal and frequency information, as it minimizes support in both time and frequency domains. Dennis Gabor was the first to address the question [43], on whether certain functions can be built as "atoms" to span the space of $L^2(\mathbb{R}^d)$. The study of Gabor systems later becomes an important part of time-frequency analysis.

2.1.2 Gabor Systems

In order to obtain local frequency information of f, we take the Fourier transform of f restricted to an interval. To avoid problems created by discontinuity, we use a smooth cut-off function as a "window".

Definition 2.8. *For function $g \in L^2(\mathbb{R}^d)$, fix $g \neq 0$ (called the window function). Then the short-time Fourier transform (STFT) of a function f with respect to g is defined in [49] as*

$$V_g f(x,\omega) = \int_{\mathbb{R}^d} f(t)\overline{g(t-x)}e^{-2\pi i t\cdot\omega}dt, \quad \textit{for} \quad x,\omega \in \mathbb{R}^d. \tag{2.12}$$

The short-time Fourier transform is also called the “sliding window Fourier transform”. $V_g f(x,\omega)$ gives the measure of the amplitude of the frequency band near ω at time x [49].

The STFT enjoys several properties similar to the classical Fourier transform. It preserves the L^2 norm of functions and inversion formula can be obtained based on its properties.

Theorem 2.9. *If $f, g \in L^2(\mathbb{R}^d)$, then*

$$\|V_g f\|_2 = \|f\|_2\|g\|_2. \tag{2.13}$$

In particular, if $\|g\|_2 = 1$, then

$$\|f\|_2 = \|V_g f\|_2 \quad for\ all\ f \in L^2(\mathbb{R}^d). \tag{2.14}$$

Thus, in this case the STFT is an isometry from $L^2(\mathbb{R}^d)$ into $L^2(\mathbb{R}^{2d})$.

Note that the time shift T of g by x is defined by

$$T_x g(t) = g(t-x) \tag{2.15}$$

and the modulation of g by ω is defined by

$$M_\omega g(t) = e^{2\pi i\omega\cdot t}g(t). \tag{2.16}$$

The signal f can be completely recovered from its STFT representation. The inverse of V_g is given by the inversion formula of STFT.

Theorem 2.10. *(Inversion formula of STFT [49]) Suppose that $g, \gamma \in L^2(\mathbb{R}^d)$ and $\langle g, \gamma \rangle \neq 0$. Then for all $f \in L^2(\mathbb{R}^d)$*

$$f = \frac{1}{\langle \gamma, g \rangle} \int \int_{\mathbb{R}^{2d}} V_g f(x, \omega) M_\omega T_x \gamma d\omega dx. \tag{2.17}$$

Based on the inversion formula of STFT, a given signal f can be expanded continuously with respect to the uncountable system of functions $\{M_\omega T_x \gamma : (x, \omega) \in \mathbb{R}^{2d}\}$. Since $L^2(\mathbb{R}^d)$ is a separable Hilbert space, the question becomes how to represent f with respect to a countable subset of time-frequency shifts. The study of this problem leads to the development of frame theory and Gabor systems.

The first attempt to find proper representation of f using countable subset of time-frequency shifts is to replace the integral by Riemann sum over lattices:

$$f = \sum_{k,n \in \mathbb{Z}^d}\sum \langle f, T_{\alpha k} M_{\beta n} \gamma \rangle T_{\alpha k} M_{\beta n} g \tag{2.18}$$

for some suitable windows $g, \gamma \in L^2(\mathbb{R}^d)$ and lattice parameters $\alpha, \beta > 0$. D. Gabor first proposed the discrete and linear time-frequency representations with a Gaussian window $g = e^{-\pi x^2}$ and $\alpha = \beta = 1$, see [43]. The *Gabor system* is defined in [49] as follows.

Definition 2.11. *Given a non-zero window function $g \in L^2(\mathbb{R}^d)$ and lattice parameters $\alpha, \beta > 0$, the set of time-frequency shifts*

$$G(g, \alpha, \beta) = \{T_{\alpha k} M_{\beta n} g : k, n \in \mathbb{Z}^d\} = \{g(t - \alpha k) e^{-2\pi i \beta n \cdot \omega} : k, n \in \mathbb{Z}^d\} \tag{2.19}$$

is called a Gabor system.

Note that the elements in Gabor system can be viewed as multiplication of shifted window function with exponentials. The set of functions

$$\{e^{-2\pi i n\cdot\omega}\}_{n\in\mathbb{Z}^d} \tag{2.20}$$

forms an orthonormal basis for $L^2(\mathbb{T}^d)$, see e.g., [10]. It is natural to ask whether the set of functions in Gabor systems can form a basis $L^2(\mathbb{R}^d)$. This problem is studied in an extended form, where the idea of orthonormal basis is generalized.

Orthonormal basis can represent space of functions in an efficient manner. Given any signal f, one can encode f by projecting it onto subspaces spanned by each orthonormal basis element ϕ_n, and f can be completely recovered from the set of coefficients $\{\langle f, \phi_n\rangle\}_{n\in\mathbb{Z}}$. However, when some of the stored coefficients become missing or contaminated by noise, it becomes difficult to recover f without losing information. In modern days, data can be easily lost or corrupted in transmission, and thus redundancy in storing information is sometimes preferred. A redundant system for data representation assures better recovery of data and can provides more detailed information of the function. We consider the notion of a *frame* introduced by Duffin and Schaeffer originally in 1952 for non-uniform sampling of band-limited functions [36].

Definition 2.12. *A sequence $\{e_j, j \in J\}$ in a (separable) Hilbert space $\mathcal{H}$ is called a frame if there exist positive constants $A, B > 0$ such that for all $f \in \mathcal{H}$*

$$A\|f\|^2 \leq \sum_{j\in J} |\langle f, e_j\rangle|^2 \leq B\|f\|^2. \tag{2.21}$$

Any two constants A, B where $0 < A \leq B < \infty$ satisfying the above statement are called frame bounds. If $A = B$, then $\{e_j : j \in J\}$ is called a tight frame.

Observe that an orthonormal basis is a tight frame with frame bounds $A = B = 1$. Frames can be thought of as a generalization of orthonormal bases. The frame elements are neither orthogonal nor linearly independent to each other, yet one can obtain a reconstruction formula from the frame coefficients $\{\langle f, e_j\rangle\}_{j\in J}$. Define the *frame operator* S on $\mathcal{H}$ by

$$Sf = \sum_{j\in J}\langle f, e_j\rangle e_j. \tag{2.22}$$

In order to state the reconstruction theorem for frames, we also need to define unconditional convergence.

Definition 2.13. *Let $\{f_j : j \in J\}$ be a countable set in a Banach space B. The series $\sum_{j\in J} f_j$ converges unconditionally to $f \in B$ if for every $\epsilon > 0$ there exists a finite set $F_0 \subseteq J$ such that*

$$\|f - \sum_{j\in F} f_j\|_B < \epsilon \quad \textit{for all finite sets} \quad F \supseteq F_0. \tag{2.23}$$

The reconstruction formula for frames is stated as follows.

Theorem 2.14. *If $\{e_j : j \in J\}$ is a frame with frame bounds $A, B > 0$, then $\{S^{-1}e_j : j \in J\}$ is a frame with frame bounds $B^{-1}, A^{-1} > 0$, the dual frame. Every $f \in \mathcal{H}$ has non-orthogonal expansions*

$$f = \sum_{j\in J}\langle f, S^{-1}e_j\rangle e_j, \tag{2.24}$$

where both sums converge unconditionally in $\mathcal{H}$.

We are interested in knowing the condition under which the Gabor system will be a frame. Note that the *Gabor frame operator* $S_{g,g}$ is given by

$$\begin{aligned} Sf &= \sum\sum_{k,n\in\mathbb{Z}^d} \langle f, T_{\alpha k} M_{\beta n} g\rangle T_{\alpha k} M_{\beta n} g \\ &= \sum\sum_{k,n\in\mathbb{Z}^d} V_g f(\alpha k, \beta n) M_{\beta n} T_{\alpha k} g. \end{aligned} \tag{2.25}$$

The study of Gabor frames is closely related to the idea of periodization, and the Wiener space comes up in the treatment of periodic functions. Denote the cube $[0,\alpha]^d$ by Q_α and write $Q = Q_1 = [0,1]^d$ for the unit cube. Let χ_A be the characteristic function of the set A.

Definition 2.15. *A function $g \in L^\infty(\mathbb{R}^d)$ belongs to the Wiener space $W = W(\mathbb{R}^d)$ if*

$$\|g\|_W = \sum_{n\in\mathbb{Z}^d} ess\sup_{x\in Q} |g(x+n)| < \infty. \tag{2.26}$$

Informally, a central result on Gabor frames can be stated as the following:

> If $g \in W(\mathbb{R}^d)$ and $\alpha, \beta > 0$ are small enough, then $G(g, \alpha, \beta)$ is a frame for $L^2(\mathbb{R}^d)$.

The formal formulation utilizes the correlation functions used in the Walnut representation, see Theorem 2.17 below, of the frame operator.

Definition 2.16. *Given $g, \gamma \in L^2(\mathbb{R}^d)$ and $\alpha, \beta > 0$, the correlation functions of the pair (g, γ) are defined to be*

$$G_n(x) = G_n^{(\alpha,\beta)}(x) = \sum_{k\in\mathbb{Z}^d} \overline{g}(x - \frac{n}{\beta} - \alpha k)\gamma(x - \alpha k) \tag{2.27}$$

for $n \in \mathbb{Z}^d$.

The G_n's are the periodizations of $T_{\frac{n}{\beta}}\overline{g}\cdot\gamma$ with period $\alpha\mathbb{Z}^d$. The existence of Gabor frames depends on parameters α,β so that the inverse operator $S_{g,g}^{-1}$ exists.

Theorem 2.17. *(Walnut [49]) Suppose that $g \in W(\mathbb{R}^d)$ and that $\alpha > 0$ is chosen such that for constants $a, b > 0$*

$$a \leq \sum_{k\in\mathbb{Z}^d} |g(x-\alpha k)|^2 \leq b < \infty \quad x-a.e. \tag{2.28}$$

Then there exists value $\beta_0 = \beta_0(\alpha) > 0$, such that $G(g,\alpha,\beta)$ is a Gabor frame for all $\beta \leq \beta_0$. Specifically, if $\beta_0 > 0$ is chosen such that

$$\sum_{\substack{n\in\mathbb{Z}^d\\ n\neq 0}} \|G_n^{(\alpha,\beta_0)}\|_\infty < \underset{x\in\mathbb{R}^d}{ess\inf} |G_0(x)|, \tag{2.29}$$

then $G(g,\alpha,\beta)$ is a frame for all $\beta \leq \beta_0$ with frame bounds

$$A = \beta^{-d}\big(a - \sum_{n\neq 0} \|G_n^{(\alpha,\beta)}\|_\infty\big) \tag{2.30}$$

and

$$B = \beta^{-d} \sum_{n\in\mathbb{Z}^d} \|G_n^{(\alpha,\beta)}\|_\infty. \tag{2.31}$$

The structure of the dual frame of a Gabor frame is also of interest. In fact, a function $f \in L^2(\mathbb{R}^d)$ can be expanded using Gabor frame and its dual frame [49].

Theorem 2.18. *If $G(g,\alpha,\beta)$ is a frame for $L^2(\mathbb{R}^d)$, then there exists a dual window $\gamma \in L^2(\mathbb{R}^d)$, such that the dual frame of $G(g,\alpha,\beta)$ is $G(\gamma,\alpha,\beta)$. Consequently, every $f \in L^2(\mathbb{R}^d)$ possesses the expansions*

$$\begin{aligned} f &= \sum_{k,n\in\mathbb{Z}^d}\sum \langle f, T_{\alpha k}M_{\beta n} g\rangle T_{\alpha k}M_{\beta n}\gamma \\ &= \sum_{k,n\in\mathbb{Z}^d}\sum \langle f, T_{\alpha k}M_{\beta n} \gamma\rangle T_{\alpha k}M_{\beta n} g \end{aligned} \tag{2.32}$$

with unconditional convergence in $L^2(\mathbb{R}^d)$. Further, the following norm equivalences hold:

$$A\|f\|_2^2 \leq \sum_{k,n\in\mathbb{Z}^d}\sum |V_g f(\alpha k, \beta n)|^2 \leq B\|f\|_2^2, \tag{2.33}$$

$$B^{-1}\|f\|_2^2 \leq \sum_{k,n\in\mathbb{Z}^d}\sum |\langle f, T_{\alpha k} M_{\beta n}\gamma\rangle|^2 \leq A^{-1}\|f\|_2^2. \tag{2.34}$$

Localization properties of Gabor transforms have made it a useful tool in applications such as image denoising [81] and analysis. Directional Gabor coefficients have been developed to detect directional information in image analysis [96]. The theoretical properties of Gabor frames have also been studied by J. Benedetto, et al in [11], [12], [13], and by discrete directional Gabor frame has been studied by W. Czaja, B. Manning, J. Murphy, K. Stubbs in [29]. We shall see in Chapter 3 that the reconstruction formula for Gabor frames and its ability to obtain local information also plays a role in modern analysis of deep neural networks (DNNs), which will be introduced in the next section. The abundance in information from redundancy of Gabor coefficients also facilitates the extraction of useful feature in image processing by providing a rich family of dictionary elements, as we will discuss in Chapter 4.

2.2 Deep Neural Networks

Deep Neural Network is the basic structure used in deep learning, and it has been a very successful tool in accomplishing tasks in machine learning in recent years, especially in fields of speech recognition, object recognition and image classification. Extensive experimental research has been done on the structure of the DNNs and algorithms used for it due to its outstanding performance compared with traditional machine learning techniques.

A formal definition of the learning process of a machine is given by Tom M. Mitchell [87]:

> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."

Broadly speaking, the experience E is provided by data sets such as images or other types of signals, and the performance measure P can be error rate from particular tasks specified in the process [87]. There are typically two types of machine learning tasks, supervised and unsupervised, depending on whether "feedback" is available to the learning system. In unsupervised learning, no labels on input data are given to the learning system, and the algorithm works on its own to find hidden structures of the input data or extract useful features from the data set. In supervised learning, input data comes with labels as their desired output from the algorithm, and the goal of the learning algorithm is to produce a "map" that maps unlabeled input data onto correct output label. The input label can be only partially available . The process of teaching the machine to do the task better with labeled input data is called training, see e.g., [106].

Representation learning is the set of methods that automatically learns the features needed for classification or detection tasks when fed with raw input data. Conventional machine learning techniques lack the ability to process data in its raw form. Careful design of a feature extractor is often needed to provide effective representation of the raw input data for classification or regression tasks. Deep learning takes the raw input and learn representation of the data through training. It can be thought of as performing a type of representation learning. Deep learning algorithms learn representation of input data layer by layer, with multiple layers of computational units constructed and connected by weighted edges. Each layer captures features of different levels of abstraction, from concrete, low level to more abstract, high level features, see e.g., [77].

Broadly speaking, there have been three waves of development in the field of deep learning: *cybernetics* which occurred in the 1940s to 1960s, *connectionism* which occurred in 1980s to 1990s, and the current resurgence known as deep learning beginning in 2006, see e.g., [47]. The earliest predecessors of modern deep learning were motivated by the study of neuroscience. In early 1940s, D.O. Hebb, a psychologist created Hebb learning, a model that takes a set of n input values $x_1, ..., x_n$ and associate them with an output y, e.g., [55]. A set of weights $w_1, ..., w_n$ were associated with the input and the output is computed as $\sum_{i=1}^{n} w_i x_i$. At around the same time, McCulloch and Pitts build the McCulloch-Pitts neuron as an early model of brain function to recognize two different categories of model by testing whether the output is positive or negative. The weights were set manually in order to produce correct results [85]. In the 1950s, Rosenblatt built the perceptron, which became the first model that could learn the weights that defined the categories based on examples from each category [103]. At around the same time, the Adaptive linear elements (ADALINE) were developed to predict a real number based on input data [118].

After some stagnation in deep learning research due to the problem of computational limitations at the time, in 1975, Werbo's design of the backpropagation algorithm accelerated the training time of multi-layer neural networks and brought deep learning back to public attention [117]. In the 1980s, Rumelhart [107] and McClelland [84] described the model of connectionism or parallel distributed processing. Connecitonism is closely related to cognitive science; its main idea is that intelligent behavior is achieved via the connection of large number of simple computation units. The idea of computation units also originated from neuroscience. A commonly used neuron, called *Rectified Linear Unit*, originated from the model called cognitron. Cognitron was first introduced by Fukushima in 1975 [41] and a renewed version, neocognitron [42], became the basis for the modern convolutional network (LeCun). Meanwhile, back propagation was also popularized in training of neural networks by Rumelhart [104] and LeCun [76].

With the development in kernel machines and graphical models in the field of machine learning, and the computational difficulties arose in deep learning, the interests in deep learning mitigated until 2006, when G. Hinton designed a type of network called deep belief network which could be trained effectively using a strategy called greedy layer-wise pretraining [57]. This network was shown to work well for recognizing handwritten digits, and especially when training data size is limited. With the increasing computational power the third wave in deep learning rises and continues to thrive until today.

2.2.1 Feedforward Neural Network

The simplest structured neural network is the feedforward neural network, a type of neural network where information only moves in one direction. Chapter 6 in the book [47] introduces the feedforward neural network in details and in this seciton we briefly summarize important concepts introduced in [47].

Connections between computational units of a feedforward neural network do not form a cycle. The idea of neural networks originated from feedforward neural networks are also called multilayer perceptrons (MLPs). The goal of the feedforward neural network is to learn representation of some function f^*. For instance, f^* can be a classifier which maps input x to category y.

Feedforward neural networks are typically represented by composition of many different functions. The model comes with a directed acyclic graph describing how different functions are composed together. For example, if we have n functions $f_1, f_2, ..., f_n$ that we want to connect, to form $f(x) = f_n(f_{n-1}(...(f_1(x))...)$. These chain structures are the most commonly seen structures of neural networks. Here f_1 is called the first layer, and f_n, the final layer, is called the output layer. The network is trained using training data. At the output layer, each input data x produces a desired output label $y \approx f^*(x)$. The expected behavior of the middle layers of the neural network is not designated by the input value x, and thus the training algorithm must decide what each individual layer should do. These middle layers that are not output layer are called hidden layers.

Each hidden layer of the neural network often consists of multiple values and

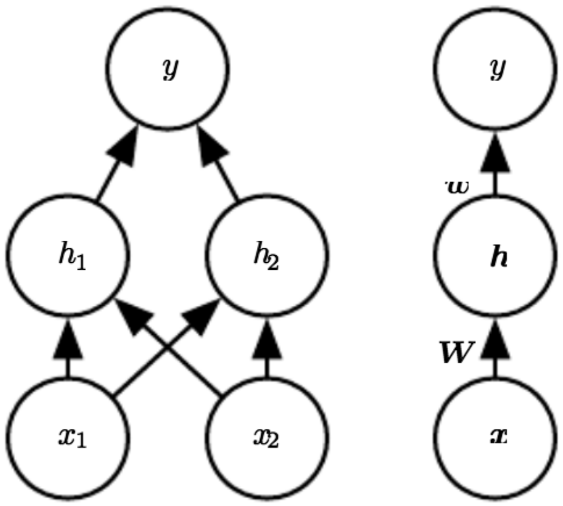

Figure 2.1: An Example of a feedforward neural network (from [47])

forms a vector of computation units, or neurons. All computation units in each layer act in parallel to perform computation tasks. The width of the model is determined by the dimensionality of each hidden layer. Output from each computation unit at current layer is passed to the next layer by connecting edges and serves as input value to the computation units of the next layer. The connection between neurons of different layers and the operations performed at each neuron can be specified a priori to satisfy different needs.

One can view the feedforward neural networks as a representation of a non-linear mapping $\phi(x)$, that occurred due to the limitations of linear representation of functions. The strategy in deep learning is to learn the model $y = f(\mathbf{x}; \boldsymbol{\theta}, \mathbf{w}) = \phi(\mathbf{x}; \boldsymbol{\theta})^T\mathbf{w}$. Here bold letters represent vectors of parameters. Paramter $\boldsymbol{\theta}$ is used to learn the representation ϕ from a broad class of functions, and the weight $\mathbf{w}$ is learned to map $\phi(\mathbf{x})$ to the desired output. Parameter $\boldsymbol{\theta}$ can be viewed as a hyperparamter that can be tuned manually, and the weight $\mathbf{w}$ is a parameter in the learning algorithm to be learned through training.

Consider a simple model in feedforward neural networks with one hidden layer with two hidden units as shown in Figure 2.1, e.g., [47]. Each value in input vector is fed into both computation units in the hidden layer. Each input vector $\mathbf{x}$ is associated with a vector of weights $\mathbf{W}$, and shifts $\mathbf{c}$. The output of the first hidden layer $\mathbf{h}$ are computed via function $f_1(\mathbf{x}; \mathbf{W}, \mathbf{c})$. Consider the linear model $f(\mathbf{x}; \mathbf{W}, \mathbf{b}) = \mathbf{x}^T \mathbf{W} + c$. The values of the output of first layer are used as input of the second layer. Because composition of linear functions is still linear, a nonlinear activation function is added to form nonlinear representation of data. Define $\mathbf{h} = g(\mathbf{W}^T + \mathbf{c})$. A commonly used activation function is the rectified linear unit, or ReLU [63]. Here g is defined by $g(z) = \max\{0, z\}$. Therefore the output of the first layer is $f_1 = \max\{0, \mathbf{W}^T \mathbf{x} + \mathbf{c}\}$. Let $\boldsymbol{\omega}$, and $\mathbf{b}$ be the weights and shifts associated with the output of the first layer $\mathbf{h}$. The complete two-layer feedforward neural network can be described as:

$$f(\mathbf{x}; \mathbf{W}, \mathbf{c}, \boldsymbol{\omega}, \mathbf{b}) = \boldsymbol{\omega}^T \max\{0, \mathbf{W}^T \mathbf{x} + \mathbf{c}\} + \mathbf{b}. \tag{2.35}$$

A feedforward neural network can be trained via gradient descent. The most commonly used training method, stochastic gradient descent [102], approximates gradient descent and uses iterative steps to minimize the cost function. The cost function is commonly defined so that the probability that the output obtained from the network produces the desired label is maximized, or such that some well-defined distance between the output of the network and the label is minimized [67].

One aspect of design of a neural network is to design its architecture. The architecture of neural network refers to its structure, including how many units to include at each layer (width), how many layers to include (depth) and how the units from different layers will be connected.

The design of the architecture is closely related to other aspects of the neural network theory: the training algorithms associated with the cost function, and the approximation properties of the neural network. The approximation properties of deep neural networks have been of interest to many mathematicians, see, e.g., [22], [26], [45], [61], [109]. Due to the fact that training a neural network takes large amount of computations and data storage, it is important to estimate the minimum amount of computations and storage needed to achieve certain error rate, both regarding to the structure of the neural network and to the amount of training data needed. It is crucial to know the theoretical limit of the neural network of certain structures in order to further pursue practical training algorithms that will lead to the theoretically proven guarantees. There have been many studies that deal with the theoretical error bound of the approximation ability of neural network of certain structures, both non-constructively and constructively. In Chapter 3, a novel structure of neural network will be presented and its approximation properties will be introduced.

Chapter 3: Provable approximation properties for deep neural networks

Deep Neural Networks (DNNs) and deep learning algorithms have achieved successful results in such areas of machine learning as image classification, speech recognition, and natural language processing [77]. There has been development in the study of theoretical framework of deep neural networks followed from its boom in applications. Some important topics in the theoretical study of neural networks include: 1) specification of the network topology to obtain certain approximation properties of functions [15], [22], [26], [86], [109]; 2) the stability analysis of the network [5], [40], [72], [97]; 3) study of the training algorithms for efficient training [50], [51], [62], [101]. For instance, there have been many studies in the stability of neural network, including the work by Elad, et al. using convolutional sparse coding [97] and the work by Balan, et al. using Lipschitz properties [5]. In this chapter, we focus on the study of approximation properties of neural networks. In particular, we design a novel type of neural network and prove its theoretical approximation rate to functions based on the network topology. The approximation bounds based on this type of neural network can be obtained as a function of the number of neurons used in each layer, and number of layers in the network. We assume that our input data is one dimensional, i.e., $x \in \mathbb{R}$.

This Chapter is structured as follows: in Section 3.1 we discuss existing work in approximation theory of neural networks; in Section 3.2 we state the mathematical preliminaries needed for building this specific type of deep neural network; in Section 3.3 we build Gabor frames using a common structure in neural network, rectified linear units, and demonstrate its theoretical approximation properties; then we introduce the structure of the neural network that we intend to build and state our main result.

3.1 Background

There is a rich body of work in theoretical analysis of deep neural networks in terms of its approximation properties. The most well-known early result is proven independently by Cybenko [26] in 1989 and Hornik [61] in 1991. The statement is that any continuous function can be uniformly approximated by a continuous neural network having only one internal hidden layer and with arbitrary continuous sigmoidal nonlinearity [26]. This result is also mentioned as the "Universal Approximation Property". The proof was existential and not constructive, i.e., the questions of how many neurons are required to yield an approximation of a given quality and how such network can be constructed are not addressed. There are several extensions of the universal approximation property [45], [46], [79], which look at the problem from different perspectives using different activation functions.

Barron [86] was the first to show that given a function $f : \mathbb{R}^m \to \mathbb{R}$ with bounded first moment of the magnitude of the Fourier transform

$$C_f = \int_{\mathbb{R}^m} |w||\tilde{f}(w)|dw < \infty, \tag{3.1}$$

there exists a neural network with single hidden layer of N sigmoidal functions, so that f_N, the output of the neural network can be bounded by

$$\|f - f_N\|_2^2 \leq \frac{c_f}{N}, \tag{3.2}$$

where c_f is proportional to C_f. Note that the error bound might scale with the dimension m, as the coefficient c_f may grow proportionally to C_f when m gets large. In [86], H. Mhaskar constructs a neural network with single hidden layer of N sigmoidal functions such that the approximation error rate

$$\|f - f_N\|_2^2 = \frac{c}{N^{2r/m}}, \tag{3.3}$$

is achieved. Here r is the number of times the input function f is differentiable. This error rate is believed to be optimal [86]. In [109], a sparsely-connected 4-layer neural network is constructed based on wavelets and obtain similar approximation error rate based on dimension d of the manifold instead of m. In particular, it states that if $f \in C^2$ has bounded Hessian, then there exists a 4-layer neural network so that

$$\|f - f_N\|_\infty = \mathcal{O}\big(N^{-\frac{2}{d}}\big). \tag{3.4}$$

This result is inspired by a recent work by Chui and Mhaskar [22], which develops a deep learning algorithm using B-splines that finds a local coordinate system for the manifold in which the high dimensional data X is embedded in, and thus providing approximation properties of the neural network.

In 2017, Bölcskei, Grohs, Kutyniok, Petersen [15] constructed a sparsely connected deep neural network with guaranteed uniform approximation rates for arbitrary function classes in $L^2(\mathbb{R}^d)$. This is a more generalized result regarding the fundamental lower bounds on the connectivity and the memory requirements of deep neural networks. More specifically, the class of functions that are well approximated by neural networks are the class of functions that are well approximated by the representation system known as the affine system. Affine systems include a rich body of representation systems in multiscale analysis such as wavelets, ridgelets, curvelets, shearlets, α-shearlets and a generalized class called α-molecules. It is also conjectured that using stochastic gradient descent to train a network to approximate α^{-1} cartoon-like functions, the best approximation obtained using M terms in the network mimic the classical best M-term approximation using α-molecules as representation system.

The approximation theory of deep neural networks is mainly studied in the real domain. In fact, the majority of the study of deep neural networks in terms of the architecture, training algorithm and theoretical aspects relies on real-valued representations. There have been attempts to build complex-valued neural networks dating from the early 1990s, e.g., [44], [68], [80]. The motivation behind building complex-valued neural networks lies in the fact that functions are well represented when expressed in the complex domain by transforms such as Fourier transform. Combination of representation of functions in both real and complex domain allows for more local information of the function, and the information is not easily obtained only in the real domain. Bruna, Chintala, LeCun in [18] provide a theoretical argument for complex-valued convolutional networks, arguing that a complex-valued convolutional network using three operations: convolution with complex-valued vectors, taking the absolute value, and local averaging, can be viewed as "multiscale windowed power spectra", "multiscale windowed absolute spectra", and "multiwavelet absolute values" with more obvious correspondence. Very recently in 2018, Trabelsi [114] published a paper that provides algorithms needed for complex batch-normalization, complex weight initialization strategies for training of complex-valued neural networks.

The approximation properties of fully complex multilayer perceptrons (MLPs) are studied in [69]. A number of elementary transcendental functions (ETFs) are defined as fully complex activation functions, and the approximation capability of the fully complex MLPs is shown using characteristics of singularity among ETFs. The complex universal approximation theorem is also existential, and does not specify the number of units and layers needed to accomplish certain approximation error rate.

We propose a novel architecture of complex-valued neural networks and prove the theoretical guarantees of their approximation capabilities for a given class of functions. In particular, given any error rate $\epsilon > 0$, we can construct a 4-layer complex neural network with $N(\epsilon)$ neurons at each layer, such that for any functions $f \in C^2(\mathbb{R})$, the output of the neural network f_N is bounded by

$$\|f - f_N\|_\infty < \frac{C_f}{N}. \tag{3.5}$$

Our construction is based on a type of time-frequency representation called Gabor frames. We will introduce some important theoretical results of Gabor frames that are needed for our proof.

3.2 Mathematical preliminaries

The reconstruction theorem, Theorem 2.18, from Chapter 1 indicates that any function $f \in L^2(\mathbb{R}^d)$ can be expanded using modulations and translations of a certain window function g, and the coefficients are computed as inner products of the function f and translations and modulations of its dual window γ. We are interested in the mathematical properties of the dual window γ. In [21], Christensen, Kim and Kim give the condition under which the dual frame γ of g is smooth.

Theorem 3.1. *(Christensen [21]) Let $K \in \mathbb{N} \cup \{0\}$, and let $b \in [0, 1/(4K+2)]$. Let g be a real-valued bounded function with* $\text{supp} g \subseteq [-(2K+1), 2K+1]$, *for which*

$$\sum_{n\in\mathbb{Z}} g(x+2n) = 1, \quad x \in [-1,1]. \tag{3.6}$$

Let $l : \mathbb{R} \to \mathbb{R}$ be any bounded function for which

$$l(x) = 0, \quad \textit{for} \quad x \leq 0, \quad \textit{and} \quad l(x) = 1, \quad \textit{for} \quad x \geq 1. \tag{3.7}$$

Define the function $\tilde{H}$ by

$$\tilde{H}(x) = \begin{cases} \frac{1}{2} l(2(x+1)), & -1 \leq x < -\frac{1}{2}, \\ 1 - \frac{1}{2} l(-2x), & -\frac{1}{2} \leq x < 0, \\ 1 - \frac{1}{2} l(2x), & 0 \leq x < \frac{1}{2}, \\ \frac{1}{2} l(2(1-x)), & \frac{1}{2} \leq x < 1, \\ 0, & \textit{otherwise.} \end{cases} \tag{3.8}$$

and let

$$\gamma(x) = b \sum_{k=-K}^{K} T_{2k} \tilde{H}(x). \tag{3.9}$$

Then one can define function γ based on l such that

- *h is a symmetric function with* $\text{supp}\gamma \subseteq [-(2K+1), 2K+1]$.
- $\{T_{\alpha k} M_{\beta n} g\}_{k,n\in\mathbb{Z}}$ *and* $\{T_{\alpha k} M_{\beta n} \gamma\}_{k,n\in\mathbb{Z}}$ *are dual frames for* $L^2(\mathbb{R})$.
- *If l is chosen to be smooth, then γ can be constructed to be smooth.*

This theorem will help illustrate properties needed for the Gabor coefficients in the proof of the approximation rate of the neural network.

3.3 Approximation of functions using complex deep neural networks

In this Section we demonstrate the detailed construction of a deep neural network which can be used to approximate functions. In particular, we first demonstrate the procedures in building a Gabor frame using rectified linear units, and secondly show the steps in construction of a deep neural network architecture based on such frame.

3.3.1 Construction of Gabor frame using rectified linear units

In this Section we demonstrate a method to build Gabor frame of $L^2(\mathbb{R})$ based on rectified linear units. We refer to the results in Section 2.1.2 in Chapter 2 to show that the dictionary we obtain is a frame in $L^2(\mathbb{R})$. The definitions of Gabor system and frame can be found in Section 2.1.2.

We first introduce the notion of Gabor system.

Definition 3.2. *A Gabor system is the set of time-frequency shifts of a non-zero window function $g \in L^2(\mathbb{R})$ with lattice parameters $\alpha, \beta > 0$:*

$$\{T_{\alpha k} M_{\beta n} g : k, n \in \mathbb{Z}^d\}. \tag{3.10}$$

We intend to build the window function g using rectified linear units. A rectified linear unit is defined as:

$$rect(x) = max\{0, x\}. \tag{3.11}$$

Rectified linear unit, or ReLU, is commonly used as activation function of the neuron of deep neural networks. We define the window function g as a triangular-shaped function:

$$g(x) = rect\left(\frac{1}{2}x + 1\right) - rect(x) + rect\left(\frac{1}{2}x - 1\right). \tag{3.12}$$

We take g as the window function of a Gabor system $G(g, \alpha, \beta) = \{T_{\alpha k} M_{\beta n} g : k, n \in \mathbb{Z}^d\}$. Figure 3.1 demonstrates the window function g constructed by (3.12). Figure 3.2 demonstrates the window function modulated by $e^{2\pi i \beta n}$ for $\beta = \frac{1}{6}$ and $n = 6$. Figure 3.3 demonstrates the modulcated window function g translated by $k = 1$.

It can be shown that $G(g, \alpha, \beta)$ is a Gabor frame. In fact, we have the following lemma:

Lemma 3.3. *Given window function $g(x) = rect\left(\frac{1}{2}x + 1\right) - rect(x) + rect\left(\frac{1}{2}x - 1\right)$, the Gabor system $G(g, \alpha, \beta)$ is a Gabor frame for $L^2(\mathbb{R})$ with values of α, β satisfying $\alpha = 1$ and $\beta \leq \frac{1}{6}$.*

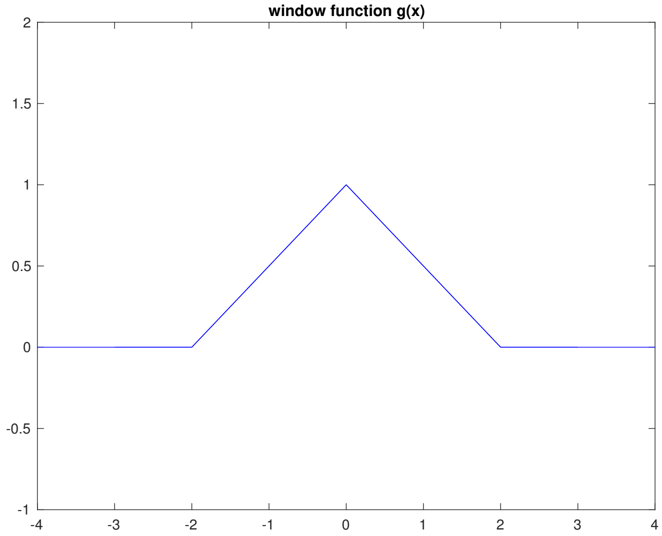

Figure 3.1: Window function g definied in (3.12).

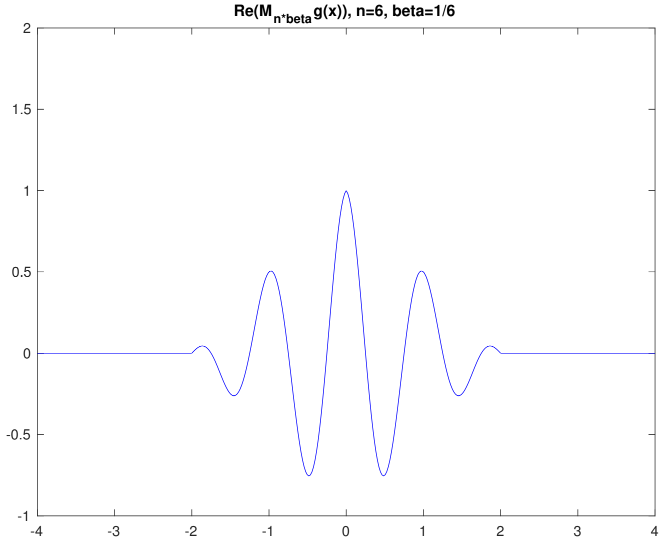

Figure 3.2: Modulated window function g, with $n = 5$

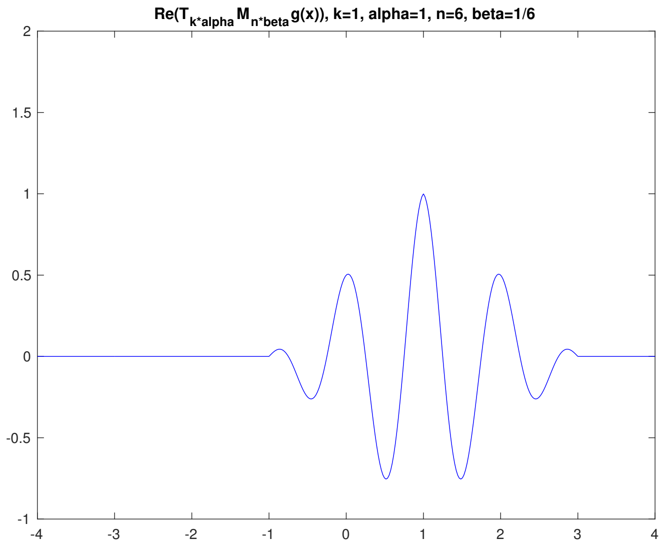

Figure 3.3: Modulated and translated window function g, with $m = 1$, $n = 6$

Proof. Note that there are other choices of α and β as long as they satisfy conditions in Theorem 2.17. Theorem 2.17 gives conditions of the window function g under which the Gabor system built of g can be a Gabor frame. We need to show that two conditions of g are satisfied.

For a function $g \in L^{\infty}(\mathbb{R}^d)$ to belong to the Wiener space $W = W(\mathbb{R}^d)$, it has to satisfy conditions in Definition 2.15. In particular, one need to choose constant α such that for constants $a, b > 0$,

$$a \leq \sum_{k \in \mathbb{Z}^d} |g(x - \alpha k)|^2 \leq b < \infty \quad a.e.. \tag{3.13}$$

In our case, we know that $d = 1$, because we look at functions $f \in L^2(\mathbb{R})$. Therefore $Q = [0, 1]$. We also know that supp $g = [-2, 2]$, and that ess $\sup_{x \in [-2,2]} |g| = 1$ by construction of g. Since $x \in [0, 1]$, and $n \in \mathbb{Z}$, there are at most 4 non-zero terms in the sum in (2.26) in Definition 2.15. Therefore,

$$\begin{aligned}\|g\|_W =&\sum_{n\in\mathbb{R}} \text{ess sup}_{x\in Q}|g(x+n)| \\ \leq& 4\text{ess sup}_{x\in Q}|g| = 4 < \infty.\end{aligned} \tag{3.14}$$

To verify the second condition, note that g is compactly supported on $[-2,2]$, and thus we can choose $\alpha = 1$, so that the infinite sum in (3.13) has only four non-zero terms for all $x \in \mathbb{R}$. We see that the upper bound b we find is $b = 4$. Given any $x \in \mathbb{R}$,

$$\sum_{k\in\mathbb{Z}} |g(x-k)|^2 \leq 4\text{ess sup}_{x\in[-2,2]}|g|^2 = 4. \tag{3.15}$$

Note that the window function g can be expressed as a piece-wise linear function:

$$g(x) = \begin{cases} \frac{1}{2}x + 1, & -2 \leq x \leq 0; \\ \\ -\frac{1}{2}x + 1, & 0 < x \leq 2. \end{cases} \tag{3.16}$$

Hence in order to find the lower bound a, we simplify the sum in (2.26) for some $x \in [-2,-1]$, and rewrite the equation as

$$\begin{aligned}\sum_{k\in\mathbb{Z}} |g(x-k)|^2 =&|g(x)|^2 + |g(x+1)|^2 + |g(x+2)|^2 + |g(x+3)|^2 \\ =&\left(\frac{1}{2}x+1\right)^2 + \left(\frac{1}{2}(x+1)+1\right)^2 + \\ &\left(-\frac{1}{2}(x+2)+1\right)^2 + \left(-\frac{1}{2}(x+3)+1\right)^2 \\ =&(x+1)^2 + \frac{5}{2}.\end{aligned} \tag{3.17}$$

Therefore, the minimum is reached when $x = -1$ and $a = \frac{5}{2}$. We've checked the two conditions in Theorem 2.17, and therefore $G(g, \alpha, \beta)$ is a frame with $\alpha = 1$. Theorem 2.17 requires β to be chosen small enough. Hence given $\alpha = 1$, we choose

$\beta \leq \beta_0 = \frac{1}{6}$ so that the the condition for β listed in Theorem 2.17 is satisfied. Note that there are other choices of α and β, i.e., given different choices of α, it is possible to find corresponding β so that $G(g, \alpha, \beta)$ is a frame. □

Now that we have introduced the Gabor frame built based on the rectified linear unit, we will build a 4-layer neural network that can be used to approximate functions, and we will show that the point-wise approximation rate of functions f can be obtained using properties of the above frame. In particular, We obtain the following result in this Chapter regarding the approximation rate of functions $f \in L^2(\mathbb{R})$.

Lemma 3.4. *Let $f \in L^2(\mathbb{R})$ be s times continuously differentiable, and let $\|f^{(s)}\|_1 < \infty$. Then for any $x \in \mathbb{R}$, there exists a construction $f_{N,K} : \mathbb{R} \to \mathbb{C}$ using Gabor coefficients with modulations up to scale N (N independent of x), with number of translations up to scale $K = K(x)$, such that*

$$|f(x) - f_{N,K(x)}(x)| < \frac{C_f}{N^{s-1}}, \tag{3.18}$$

where C is a constant dependent on the derivatives of f up to order s, and $|\cdot|$ denotes the point-wise absolute value.

Note that by Theorem 2.18, any function $f \in L^2(\mathbb{R})$ can be represented by the infinite expansion on modulations and translations of the window function g:

$$f = \sum_{k,n \in \mathbb{Z}^d} \sum \langle f, T_{\alpha k} M_{\beta n} \gamma \rangle T_{\alpha k} M_{\beta n} g, \tag{3.19}$$

where γ is the dual window function of g. Translation $T_{\alpha k}$ of window function g is defined by

$$T_{\alpha k} g(x) = g(x - \alpha k), \tag{3.20}$$

and modulation $M_{\beta n}$ of window function g is defined by

$$M_{\beta n} g(x) = e^{2\pi i \beta n \cdot x} g(x). \tag{3.21}$$

In order to prove Lemma 3.4, first we need to discuss the properties of the dual window function γ.

Lemma 3.5. *Given window function* $g = rect\left(\frac{1}{2}x + 1\right) - rect(x) + rect\left(\frac{1}{2}x - 1\right)$, *there exists a compactly supported dual window function* γ *such that* $\gamma \in C^\infty(\mathbb{R})$ *is smooth.*

Proof. Theorem 3.1 provides conditions on the window function g under which a smooth dual frame can be constructed. Therefore we only need to show that g satisfies the conditions listed in Theorem 3.1: supp $g \subseteq [-(2K+1), 2K+1]$ and

$$\sum_{n \in \mathbb{Z}} g(x + 2n) = 1, \quad x \in [-1, 1]. \tag{3.22}$$

By construction, supp $g = [-2, 2]$. Hence, we can take $K = 1$, then supp $g \subseteq [-3, 3]$. Note that g can be written as a piece-wise linear function:

$$g(x) = \begin{cases} \frac{1}{2}x + 1, & x \in [-2, 0), \\ -\frac{1}{2}x + 1, & x \in [0, 2]. \end{cases}$$

Therefore, for $x \in [-1, 1]$, if $x \in [-1, 0)$, then $x + 2 \in [1, 2)$, and

$$\sum_{n \in \mathbb{Z}} g(x + 2n) = g(x) + g(x + 2) = \frac{1}{2}x + 1 - \frac{1}{2}(x + 2) + 1 = 1. \tag{3.23}$$

If $x \in [0,1]$, then $x-2 \in [-2,-1]$, and

$$\sum_{n\in\mathbb{Z}} g(x+2n) = g(x) + g(x-2) = -\frac{1}{2}x + 1 + \frac{1}{2}(x-2) + 1 = 1. \tag{3.24}$$

Therefore, the condition in Theorem 3.1 is satisfied, and we obtain a smooth dual window function γ. □

Now we prove Lemma 3.4.

Proof. We have shown that $G(g,\alpha,\beta)$ is a Gabor frame for $L^2(\mathbb{R})$ with $\alpha = 1$ and $\beta \leq \frac{1}{6}$, and constructed a smooth dual window γ. In the following proof, we assume $\alpha = 1$ and $\beta = \frac{1}{6}$.

By Theorem 2.18, for any function $f \in L^2(\mathbb{R})$, we can expand f in terms of modulations and translations of g by

$$f = \sum_{k\in\mathbb{Z}} \sum_{n\in\mathbb{Z}} \langle f, T_{\alpha k} M_{\beta n} \gamma \rangle T_{\alpha k} M_{\beta n} g. \tag{3.25}$$

Let $f_{K,N}$ be the approximation obtained by the first $(2K+1) \times (2N+1)$ terms in the expansion:

$$f_{K,N} = \sum_{|k|\leq K} \sum_{|n|\leq N} \langle f, T_{\alpha k} M_{\beta n} \gamma \rangle T_{\alpha k} M_{\beta n} g. \tag{3.26}$$

Then for any $x \in \mathbb{R}$,

$$\begin{aligned}
&|f(x) - f_{K,N}(x)| \\
&= \left| \sum_{k\in\mathbb{Z}} \sum_{|n|>N} \langle f, T_{\alpha k} M_{\beta n} \gamma \rangle T_{\alpha k} M_{\beta n} g(x) \right| + \left| \sum_{|k|>K} \sum_{|n|\le N} \langle f, T_{\alpha k} M_{\beta n} \gamma \rangle T_{\alpha k} M_{\beta n} g(x) \right| \\
&= \left| \sum_{k\in\mathbb{Z}} \sum_{|n|>N} \langle f, T_{\alpha k} M_{\beta n} \gamma \rangle e^{2\pi i \beta n \cdot (x-\alpha k)} g(x-\alpha k) \right| \\
&+ \left| \sum_{|k|>K} \sum_{|n|\le N} \langle f, T_{\alpha k} M_{\beta n} \gamma \rangle e^{2\pi i \beta n \cdot (x-\alpha k)} g(x-\alpha k) \right| \\
&\le \sum_{k\in\mathbb{Z}} \sum_{|n|>N} |\langle f, T_{\alpha k} M_{\beta n} \gamma \rangle| \cdot |e^{2\pi i \beta n \cdot (x-\alpha k)}| \cdot |g(x-\alpha k)| \\
&+ \sum_{|k|>K} \sum_{|n|\le N} |\langle f, T_{\alpha k} M_{\beta n} \gamma \rangle| \cdot |e^{2\pi i \beta n \cdot (x-\alpha k)}| \cdot |g(x-\alpha k)|.
\end{aligned} \tag{3.27}$$

Since $|e^{2\pi i \beta n \cdot (x-\alpha k)}| = 1$, we have

$$\begin{aligned}
|f(x) - f_{K,N}(x)| \le &\sum_{k\in\mathbb{Z}} \sum_{|n|>N} |\langle f, T_{\alpha k} M_{\beta n} \gamma \rangle||g(x-\alpha k)| \\
&+ \sum_{|k|>K} |\sum_{|n|\le N} \langle f, T_{\alpha k} M_{\beta n} \gamma \rangle||g(x-\alpha k)|.
\end{aligned} \tag{3.28}$$

Note that

$$\begin{aligned}
\langle f, T_{\alpha k} M_{\beta n} \gamma \rangle &= \int_{\mathbb{R}} f(t) \overline{e^{2\pi i \beta n (t-\alpha k)} \gamma(t-\alpha k)} dt \\
&= \int_{\mathbb{R}} f(t+\alpha k) \overline{\gamma(t)} e^{-2\pi i \beta n (t)} dt \\
&= \frac{1}{\beta} \int_{\mathbb{R}} f(\frac{1}{\beta} t + \alpha k) \overline{\gamma(\frac{1}{\beta} t)} e^{-2\pi i n (t)} dt.
\end{aligned} \tag{3.29}$$

Therefore $\langle f, T_{\alpha k} M_{\beta n} \gamma \rangle$ is the Fourier transform of the function $H_{\alpha,\beta,k}(t_0) = f(\frac{1}{\beta} t_0 + \alpha k) \overline{\gamma(\frac{1}{\beta} t_0)}$, i.e.,

$$\langle f, T_{\alpha k} M_{\beta n} \gamma \rangle = \hat{H}_{\alpha,\beta,k}(n). \tag{3.30}$$

Since f is s times continuously differentiable, and γ is smooth, $H_{\alpha,\beta,k}$ is also s times continuously differentiable by the product rule of differentiation. Since $f \in C^s(\mathbb{R})$, by properties of the space $C^s(\mathbb{R})$, we have

$$|\hat{H}_{\alpha,\beta,k}(n)| < \frac{C_f}{n^s}, \tag{3.31}$$

and

$$\sum_{|n|>N} |\hat{H}_{\alpha,\beta,k}(n)| < \sum_{|n|>N} \frac{C_f}{n^s} < s\int_N^\infty \frac{2C_f}{n^s} dn = \frac{2sC_f}{(s-1)N^{s-1}}, \tag{3.32}$$

where C_f is a constant depending on L^1 norm of the sth derivative of $H_{\alpha,\beta,k}$, which depends on the derivatives of f up to order s.

Equation (3.31) follows from the properties of Fourier coefficients in Theorem 2.4. In order to use time differentiability of functions in Theorem 2.4, we check the conditions in Theorem 2.4 on $H_{\alpha,\beta,k}$. Recall (3.30), and it suffices to check the conditions on $f\gamma$. Since $f \in C^s(\mathbb{R})$, it has bounded derivatives up to order s, and since $\gamma \in C^\infty(\mathbb{R})$ and γ is compactly supported, $\lim_{|n|\to\infty} \gamma^{(s_0)}(n) = 0$ for all $0 \leq s_0 \leq s$. Therefore, we obtain

$$\lim_{|n|\to\infty} (f\gamma)^{(s_0)}(n) = 0, \tag{3.33}$$

for all $0 \leq s_0 \leq s$. Since $f \in C^s(\mathbb{R})$, we get $\|f\|_\infty < \infty$. And since γ is compactly supported, we get $\|f\gamma\|_1 < \infty$. Therefore, $f\gamma \in L^1(\mathbb{R})$, and we have $H_{\alpha,\beta,k} \in L^1(\mathbb{R})$. Now we can apply time differentiability formula of Fourier coefficients to obtain

$$\widehat{(f\gamma)^{(s)}}(n) = (2\pi in)^s \widehat{(f\gamma)}(n), \tag{3.34}$$

and

$$\hat{H}_{\alpha,\beta,k}(n) = \frac{H^{\hat{(s)}}{}_{\alpha,\beta,k}}{|2\pi in|^s}. \tag{3.35}$$

By boundedness of Fourier coefficients of functions in L^1, we obtain that

$$|\hat{H}_{\alpha,\beta,k}(n)| = \frac{|\hat{H^{(s)}}_{\alpha,\beta,k}|}{|2\pi in|^s} < \frac{\|H^{(s)}_{\alpha,\beta,k}\|_1}{|2\pi in|^s} = \frac{C_\beta\|(f\gamma)^{(s)}\|_1}{|2\pi in|^s} < \infty. \tag{3.36}$$

Note that since $H_{\alpha,\beta,k}$ can be written as $f(\frac{1}{\beta}t_0 + \alpha k)\overline{\gamma(\frac{1}{\beta}t_0)}$, its derivatives up to order s depend only on f with fixed γ, and a constant factor C_β. Here C_β depends on powers of β up to β^s. Since $f \in C^s(\mathbb{R})$, it has bounded derivatives up to order s, and since γ is compactly supported, we get $(f\gamma)^{(s)} \in L^1(\mathbb{R})$. Note that the parameter β is chosen in advance, and γ is fixed, so we say that the constant in (3.31) depends only on input function f. Therefore we obtain the bound in (3.31), and (3.32) follows.

Now, we plug in the bound in (3.32) back into the first term in (3.28) and obtain

$$|f(x)-f_{K,N}(x)| < \sum_{k\in\mathbb{Z}} \frac{2sC_f}{(s-1)N^{s-1}}|g(x-\alpha k)| + \sum_{|k|>K}\sum_{|n|\le N}|\langle f, T_{\alpha k}M_{\beta n}\gamma\rangle||g(x-\alpha k)|. \tag{3.37}$$

Note that g is compactly supported on $[-2,2]$. Then, for any x, there are only finitely many k's ($\lceil\frac{4}{\alpha}\rceil$) such that $g(x-\alpha k) \neq 0$. Therefore, we get

$$\sum_{k\in\mathbb{Z}} \frac{2sC_f}{(s-1)N^{s-1}}|g(x-\alpha k)| \le \frac{\lceil\frac{4}{\alpha}\rceil 2sC_f}{(s-1)N^{s-1}}. \tag{3.38}$$

To obtain an error bound on the second term in (3.37), note that for any x, we can find $K = \max\{\lceil|\frac{x+2}{\alpha}|\rceil, \lceil|\frac{2-x}{\alpha}|\rceil, \lceil|\frac{4}{\alpha}|\rceil\}$, such that for all $|k| > K$, $g(x-\alpha k) = 0$. Choosing K satisfying this condition with respect to x, we get

$$\sum_{|k|>K(x)}\sum_{|n|\le N}|\langle f, T_{\alpha k}M_{\beta n}\gamma\rangle||g(x-\alpha k)| = 0. \tag{3.39}$$

By plugging (3.38) and (3.39) back in (3.37), we have

$$|f(x) - f_{K(x),N}(x)| < \frac{\lceil\frac{4}{\alpha}\rceil 2sC_f}{(s-1)N^{s-1}}. \tag{3.40}$$

Therefore, for any s times continuously differentiable function $f \in L^1(\mathbb{R})$, with $f^{(s)} \in L^1(\mathbb{R})$, given any $x \in \mathbb{R}$, we can approximate $f(x)$ using $f_{K(x),N}$ evaluated at x with $|f(x) - f_{K(x),N}(x)| < \frac{C_f}{N^{s-1}}$. $\square$

For compactly supported functions, we derive the following corollary from Lemma 3.4.

Corollary 3.6. *Let $f \in L^2(\mathbb{R})$ be compactly supported with supp $f = [t_1, t_2]$, s times continuously differentiable, and let $\|f^{(s)}\|_1 < \infty$. Then there exists a construction $f_{N,K} : \mathbb{R} \to \mathbb{C}$ using Gabor coefficients with modulations up to scale N, with number of translations up to scale $K = K(t_1, t_2)$ (K depending on the support of f), such that for any $x \in [t_1, t_2]$,*

$$|f(x) - f_{N,K}(x)| < \frac{C_f}{N^{s-1}}, \tag{3.41}$$

where C_f is a constant dependent on the derivatives of f up to order s, and $|\cdot|$ denotes the point-wise absolute value.

Proof. The proof follows from the proof of Lemma 3.4. We obtain (3.37) for the error between $|f(x) - f_{K,N}(x)|$. To obtain an error bound on the second term in (3.37), note that f is compactly supported with support supp $f = [t_1, t_2]$. Therefore, we can find $K = \max\{\lceil |\frac{t_2+2}{\alpha}| \rceil, \lceil |\frac{2-t_1}{\alpha}| \rceil, \lceil |\frac{t_2-t_1}{\alpha}| \rceil\}$, such that for all $x \in [t_1, t_2]$, for all $|k| > K$, $g(x - \alpha k) = 0$. Then with this K depending on t_1, t_2, we obtain

$$\sum_{|k|>K} \sum_{|n| \leq N} |\langle f, T_{\alpha k} M_{\beta n} \gamma \rangle| |g(x - \alpha k)| = 0, \tag{3.42}$$

and combining (3.38) and (3.42) in (3.37), we have

$$|f(x) - f_{K,N}(x)| < \frac{\lceil \frac{4}{\alpha} \rceil 2sC_f}{(s-1)N^{s-1}}. \tag{3.43}$$

Therefore, for any compactly supported, s times continuously differentiable function $f \in L^1(\mathbb{R})$, with $f^{(s)} \in L^1(\mathbb{R})$, we can approximate f using $f_{K,N}$ (K independent of x) with $|f(x) - f_{K,N}(x)| < \frac{C_f}{N^{s-1}}$ for any $x \in [t_1, t_2]$. □

3.3.2 Construction of deep neural network for function approximation

In this Section we introduce the construction of the neural network based on Gabor frame $G(g, \alpha, \beta)$ built using window function g defined in equation 3.12.

We construct the neural network with specified number of nodes and layers as the following. The input layer consists of one node for input value $x \in L\mathbb{R}$. The first layer consists of all the shifts of input value x: $\{x - \alpha k\}$ for $k \in [-K, K]$. In the second layer, output from second layer serves as input of modulated rectified linear units of three types: $rect\left(\frac{1}{2}x + 1\right)$, $-rect(x)$, $rect\left(\frac{1}{2}x - 1\right)$. Each rectified linear unit is modulated by $\exp 2\pi i \beta n x$ for $n \in [-N, N]$. In the third layer, outputs from different rectified linear units of the same modulation term are added together, and we obtain $T_{\alpha k} M_{\beta n} g$ for all $k \in [-K, K]$ and $n \in [-N, N]$. In the fourth layer, outputs from the third layer are added to produce the final output function. The output of the network has the form

$$f_{K,N} = \sum_{|k| \le K} \sum_{|n| \le N} w_{k,n} T_{\alpha k} M_{\beta n} g, \tag{3.44}$$

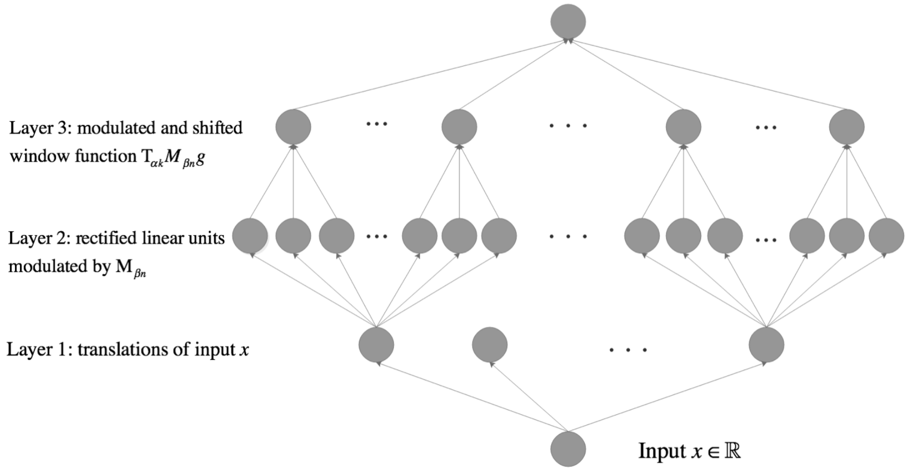
Layer 3: modulated and shifted window function $T_{\alpha k}M_{\beta n}g$
Layer 2: rectified linear units modulated by $M_{\beta n}$
Layer 1: translations of input x
Input $x \in \mathbb{R}$

Figure 3.4: Illustration of the neural network

where $w_{k,n}$ are the weights on each $T_{\alpha k}M_{\beta n}g$. The structure of the network is illustrated in Figure 3.4.

With the above construction of the neural network, we reach the main theorem of this Chapter.

Theorem 3.7. *Let $f \in L^2(\mathbb{R})$. If f is compactly supported with supp $f = [t_1, t_2]$, s times continuously differentiable for $s \geq 2$, then f can be approximated point-wise using a 4-layer network with $(2K+1)(4(2N+1)+1)$ units, and the absolute value of the point-wise error is bounded by $\frac{C_f}{N^{s-1}}$. There are $2K+1$ linear units in the first layer; $(2K+1)\times 3\times(2N+1)$ units in the second layer; $(2K+1)(2N+1)$ linear units in the third layer and a single linear unit in the fourth layer. Here $K = K(t_1, t_2)$ is the number of translations and N is the number of modulations in the Gabor system used to construct the neural network.*

Proof. We show that the neural network we built forms $(2K+1)\times(2N+1)$ terms of a Gabor frame.

Given any input x, the first layer of the neural network computes all translations of x by αk for $k \in [-K, K]$, creating $(2K+1)$ units. The second layer of the neural network takes each translated x as input, pushes it through three different rectified linear units, with each rectified linear unit modulated by $e^{2\pi i\beta n\cdot x}$ of scales $n \in [-N, N]$. The second layer needs $(2K+1)\times 3\times(2N+1)$ units in total, since units in this layer take each output from the previous layer as input through functions $e^{2\pi i\beta n\cdot x} rect\left(\frac{1}{2}x+1\right)$, $-e^{2\pi i\beta n\cdot x} rect(x)$, and $e^{2\pi i\beta n\cdot x} rect\left(\frac{1}{2}x-1\right)$ for all $n \in [-N, N]$. In the third layer, output of different rectified linear units are combined for each translation scale k and modulation scale n, producing $(2K+1)\times(2N+1)$ linear units. The output of this layer are the translations and modulations of the window function g up to scales K and N.

In fact, by definition of g in terms of rectified linear units, each element $T_{\alpha k}M_{\beta n}$ in Gabor system $G(g, \alpha, \beta)$ can be represented as

$$\begin{aligned} T_{\alpha k}M_{\beta n}g(x) =& e^{2\pi i\beta n\cdot(x-\alpha k)}g(x-\alpha k) \\ =& e^{2\pi i\beta n\cdot(x-\alpha k)}\big(rect\left(\frac{1}{2}(x-\alpha k\right)+1\big) \\ & - rect\left(x-\alpha k\right) + rect\left(\frac{1}{2}(x-\alpha k)-1)\right). \end{aligned} \tag{3.45}$$

In the fourth layer, all the translations and modulations of window function g are added up to produce the final approximation result $f_{K,N}$. Note that we can choose parameters α and β so that the Gabor system $G(g, \alpha, \beta)$ is a frame. An example of the choice of α and β is $\alpha = 1$ and $0 < \beta \leq \frac{1}{6}$.

Finally, the fact that $G(g, \alpha, \beta)$ is a frame allows us to find upper bound on the ability of the neural network to approximate function f using $f_{K,N}$, for $f \in C^s(\mathbb{R})$

with bounded sth derivative. Details of the approximation rate are stated in Lemma 3.4 and Corollary 3.6. $\square$

Note that we could construct similar schemes to approximate any x for functions that are not compactly supported, but since the size of the scheme would depend on the particular choice of x (specifically the choice of K would depend on x), we would not define such scheme as a neural network.

3.4 Conclusions

In this Chapter we constructed a deep neural network that can approximate functions $f \in L^2(\mathbb{R})$. We discussed the structure of the neural network in terms of Gabor systems and described the construction of the neural network using rectified linear units. We show that both construction describe the same structure mathematically and prove that given specific error rate, we can construct neural network of size dependent on the error rate in order to approximate functions to desired accuracy.

This work does not discuss the training aspect of neural network. We show that theoretically certain error bound that has direct impact on the size of the structure of the network can be obtained, thus providing the possibility to obtain such error bound when we use training in practice. There has been successfully developed and tested training algorithms designed for complex valued neural networks recently [114], and based on the simple structure of this network, extending this work to include training aspect of the neural network is of future interests.

The focus of this work is on the ability to specify size and structure of a neural network given desired error rate of approximation. We think of input x as intact and reliable in our construction and do not consider noise in the input value x. However, in many applications, the input data is often contaminated with noise and there have been interests in the study of reconstruction of signal x when x is contaminated with noise. In the next Chapter, we look at a novel construction of a common type of operation in neural networks, pooling, and study the reconstruction stability of neural networks when such operation is included in the network.

Chapter 4: Maximal function pooling in deep convolutional sparse coding

4.1 Introduction

Convolutional neural networks (CNNs), as a popular type of architecture in deep learning, have shown outstanding performances in various applications such as image classification [23], [24]. Many examples of CNNs have used pooling as a layer in their networks. Pooling is a dimension reduction technique that divides an image into subregions and returns only one pixel value as the representative of each subregion. Max pooling and average pooling are widely used traditional pooling strategies and have demonstrated good performances in application tasks [47]. Pooling helps reduce overfitting of training data, which is a common problem in many applications.

Inspired by the maximal function from harmonic analysis [25], [90], we introduce a novel pooling strategy, *maxfun pooling*, which is similar to both max pooling and average pooing. In particular, max pooling takes the maximum value in each pooling region as the scalar output, and average pooling takes the average of all entries in each pooling region as the scalar output. Maximal function, or the

Hardy-Littlewood maximal function [25], [90] Mf of function f, is defined by

$$(Mf)(x) = \sup_{x \in B} \frac{1}{|B|} \int_B |f| \tag{4.1}$$

for each $x \in \mathbb{R}^n$. Here the supreme is taken over all balls $B \in \mathbb{R}^n$ which contains x and $|B|$ is the measure of B. An important property of the Hardy-Littlewood maximal operator is that for $f \in L_p(\mathbb{R}^n)$ where $1 \leq p \leq \infty$, Mf is finite almost everywhere [48]. We limit the support of this operator to be finite, discretize it and define the maximal function pooling, or *maxfun pooling*, as follows.

Let $X \in \mathbb{R}^{N \times N}$. The maxfun pooling $Mf(X)$ for the kth pooling region is defined by

$$(Mf_{b,s}(X))_k = \max_{1<j\leq b, 1\leq n\leq (b-j+1)^2} \left(\frac{1}{j^2} \sum_{i \in B_{k,j,n}} X_i \right), \tag{4.2}$$

where x_i is the ith element in X. The side length of the pooling region is b, and we compute averages of sub-square regions of side length j inside each pooling region, with each sub-squares labeled by $n = 1, ..., (b-j+1)^2$. $B_{k,j,n}$ is the set of indices i of X such that x_i is in the nth sub-square region of side length j in the kth pooling region. The stride size s is the interval length at which we take each pooling region. The maxfun pooling computes averages of sub-regions of different sizes in each pooling region, and selects the largest average among all.

We analyze properties of maxfun pooling in the realm of convolutional sparse coding. It has been shown that feed forward convolutional neural network can be viewed as convolutional sparse coding [97]. Moreover, under the view of convolutional sparse coding, stable recovery of the signal contaminated with noise can be achieved, given simple sparsity conditions [97]. Equivalently it means that feed forward neural network maintains stability under noisy situations. The case of pooling function analyzed via convolutional sparse coding is studied in [65], where the two common pooling functions, max pooling and average pooling are analyzed. We follow the framework in [65] and analyze maxfun pooling in this Section. We show that stability of the neural network with presence of noise is also preserved with maxfun pooling.

4.2 Convolutional Sparse Coding

Sparse coding problem is an important problem in signal processing, where one aims at finding a low dimensional representation using few dictionaries for high dimensional data [98]. Given a vector $X \in \mathbb{R}^N$, and a dictionary $D \in \mathbb{R}^{N \times M}$, the sparse coding problem attempts to find the most sparse vector $\Gamma \in \mathbb{R}^M$ such that $X = D\Gamma$. In other words, for a fixed dictionary $D \in \mathbb{R}^{M \times N}$, the sparse coding problem attempts to solve:

$$\min_{\Gamma} \|\Gamma\|_0 \quad s.t. \quad D\Gamma = X, \tag{4.3}$$

where $\|\Gamma\|_0$ is the l_0 pseudo norm, and gives the number of non-zero elements in vector Γ. Each column in D represents one base in the dictionary, and finding the basis to represent data X so that minimum number of basis are used solves the sparse coding problem.

Restriction on the sparsity of Γ with respect to the mutual coherence of the dictionary D can guarantee uniqueness of the solution to (4.3). Mutual coherence

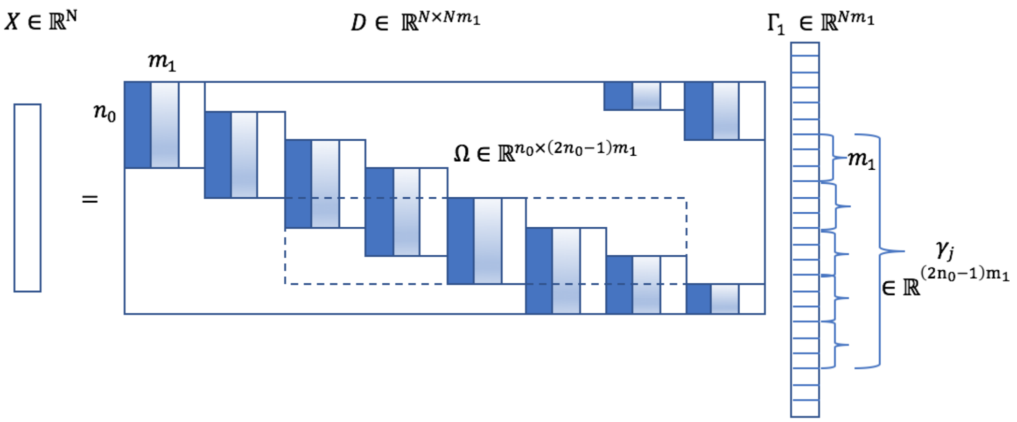
$X \in \mathbb{R}^N$
$D \in \mathbb{R}^{N\times Nm_1}$
$\Gamma_1 \in \mathbb{R}^{Nm_1}$
m_1
n_0
=
$\Omega \in \mathbb{R}^{n_0\times(2n_0-1)m_1}$
m_1
γ_j
$\in \mathbb{R}^{(2n_0-1)m_1}$

Figure 4.1: Convolutional Sparse Coding, level 1

of a matrix D is defined as [35]

$$\mu(D) = \max_{i\neq j} \frac{|di^T d_j|}{\|d_i\|_2 \cdot \|d_j\|_2}, \tag{4.4}$$

where d_i's are the columns of matrix D. However, finding the solution remains NP hard. Relaxation of the model to allow noise and form error bound leads to the following formulation:

$$\min_{\Gamma} \|\Gamma\|_0 \quad s.t. \quad \|D\Gamma - X\| < \epsilon. \tag{4.5}$$

When high dimensional signals are present, an alternative method called convolutional sparse coding model(CSC) was proposed. One attempts to represent the whole signal $X \in \mathbb{R}^N$ as a multiplication of a global convolutional dictionary $D \in \mathbb{R}^{N\times Nm_1}$ and a sparse vector $\Gamma \in \mathbb{R}^{Nm_1}$. D is constructed by shifting a local matrix of size $n_0 \times m_1$ in all possible positions, as shown in Figure 4.1. We define the jth stripe γ_j of the sparse vector Γ as a group of $2n_0 - 1$ adjacent sparse vectors of length m_1, starting at the jth vector of length m_1. See Figure 4.1 for an illustration.

The stripe γ_j gives the representation of a patch of X, x_j of length n_0 by $x_j = \Omega_j \gamma_j$. $\Omega_j \in \mathbb{R}^{n_0 \times (2n_0-1)m_1}$ is a submatrix of D, called a stripe dictionary consisting of n_0 consecutive rows of D and the columns of zeros removed.

The $l_{0,\infty}$ norm of the global sparse vector Γ_1 is defined by the maximum number of non-zeros in any stripe of length $(2n_0-1)m_1$ extracted from it, i.e.,

$$\|\Gamma\|_{0,\infty}^{s} = \max_{i \in \{1,\dots,N\}} \|\gamma_i\|_0. \tag{4.6}$$

Here $\|\cdot\|_0$ is the zero norm that gives the number of nonzero elements of a vector. A multi-layer convolutional sparse coding model is defined so that the output sparse vector Γ from the previous layer is served as the input vector in the next layer, and we aim at finding a new representation Γ_2 for a new set of dictionary D_2. Formally, the problem of finding solutions to multi-layer convolutional sparse coding problem is defined as the deep coding problem DCP_λ in [97]:

$$\begin{aligned}
&Find \quad \{\Gamma_i\}_{i=1}^{L} && s.t. \\
&X = D_1\Gamma_1, && \|\Gamma_1\|_{0,\infty}^{s} \quad \leq \lambda_1 \\
&\Gamma_1 = D_2\Gamma_2, && \|\Gamma_2\|_{0,\infty}^{s} \quad \leq \lambda_2 \\
&\vdots && \\
&\Gamma_{L-1} = D_L\Gamma_L, && \|\Gamma_L\|_{0,\infty}^{s} \quad \leq \lambda_L
\end{aligned} \tag{4.7}$$

where λ_i are bounds on sparsity of the output vector Γ_i at each level, and L is the number of layers. Note that we want to find representations of the input vectors at each layer that are sparse in terms of its stripe sparsity, defined by $\|\cdot\|_{0,\infty}$.

In practice, the input signal X can be contaminated with noise, and we have $Y = X + E$ as the input signal instead of X, where E represents noise. In this case, we relax the constraint and allow the representation to vary within some error bounds of the input signal. The deep coding problem when noise is present (DCP_λ^ϵ) [97] is defined as:

$$
\begin{aligned}
&Find \quad \{\Gamma_i, P_i\}_{i=1}^L && s.t. \\
&\|Y - D_1\Gamma_1\|_2 \leq \epsilon_1, && \|\Gamma_1\|_{0,\infty}^s \quad \leq \lambda_1 \\
&\|\Gamma_1 - D_2\Gamma_2\|_2 \leq \epsilon_2, && \|\Gamma_2\|_{0,\infty}^s \quad \leq \lambda_2 \\
&\vdots \\
&\|\Gamma_{L-1} - D_L\Gamma_L\|_2 \leq \epsilon_L, && \|\Gamma_L\|_{0,\infty}^s \quad \leq \lambda_L.
\end{aligned}
\tag{4.8}
$$

Here ϵ_i is the error bound that are allowed in the ith layer.

Uniqueness of the solution to the DCP_λ model, and the stability of the solution to the DCP_λ^ϵ problem have been shown in [97]. The equivalence of deep convolutional sparse coding problem and feed forward neural network have also been shwon in [97]. It is proven in [97] that one can view the output vector Γ_i from each layer of the DCP_λ problem as the output from one layer of feed forward convolutional neural network (CNN), and thus the deep convolutional sparse coding problem can be viewed as a signal reconstruction problem of CNNs.

Pooling is a common operation included in CNNs that serves as feature extraction method to reduce redundancy of representation of signal and save computational resources. It has been shown that adding max pooling and average pooling in the feed forward path preserves the stability of the neural network [65]. We demonstrate that the maxfun pooling, preserves the stability in the same sense of a convolutional neural network when added in between layers of convolutions.

Given a input signal X, the deep convolutional sparse coding problem with pooling ($DCPP$) is defined by [65]

$$\begin{array}{lll} Find \quad \{\Gamma_i, P_i\}_{i=1}^{L} & s.t. & \\ X = D_1\Gamma_1, & \|\Gamma_1\|_{0,\infty}^{s} \quad \leq \lambda_1 & P_1 = Pool_{b_1,s_1}(\Gamma_1), \\ P_1 = D_2\Gamma_2, & \|\Gamma_2\|_{0,\infty}^{s} \quad \leq \lambda_2 & P_2 = Pool_{b_2,s_2}(\Gamma_2), \\ \vdots & & \vdots \\ P_{L-1} = D_L\Gamma_L, & \|\Gamma_L\|_{0,\infty}^{s} \quad \leq \lambda_L, & P_L = Pool_{b_L,s_L}(\Gamma_L), \end{array} \tag{4.9}$$

where $Pool_{b,s}$ is the pooling operation. We take $Pool_{b,s}$ to be the maxfun pooling, i.e., $Pool_{b,s} = Mf_{b,s}$, as defined in (4.2).

Problem (4.9) intends to find a stable sparse representation Γ_1 of X with dictionary elements in D_1, given restriction on the stripe-sparsity of Γ_1. Then pooling operation is performed on Γ_1 to get P_1. In second layer, we attempt to find the sparse representation Γ_2 of P_1 with dictionary elements in D_2. The stripe-sparsity of Γ_2 is restricted to be no greater than λ_2. We repeat the process L times.

If our input signal X is contaminated by noise E, we are still interested in finding a sparse representation that is stable. Define the deep convolutional sparse coding problem with pooling when noise is present ($DCPP^{\epsilon}$) by [65]

$$\begin{array}{lll} Find \quad \{\Gamma_i, P_i\}_{i=1}^{L} & s.t. & \\ \|Y - D_1\Gamma_1\| \leq \epsilon_1, & \|\Gamma_1\|_{0,\infty}^{s} \quad \leq \lambda_1 & P_1 = Pool_{b_1,s_1}(\Gamma_1), \\ \|P_1 - D_2\Gamma_2\| \leq \epsilon_2, & \|\Gamma_2\|_{0,\infty}^{s} \quad \leq \lambda_2 & P_2 = Pool_{b_2,s_2}(\Gamma_2), \\ \vdots & & \vdots \\ \|P_{L-1} - D_L\Gamma_L\| \leq \epsilon_L, & \|\Gamma_L\|_{0,\infty}^{s} \quad \leq \lambda_L, & P_L = Pool_{b_L,s_L}(\Gamma_L), \end{array} \tag{4.10}$$

It has been shown in [65] that when max pooling and average pooling are used, the stability of solution to the $DCPP^{\epsilon}$ problem is preserved. We show that when we use maxfun pooling, the stability result also holds. We prove the following theorem:

Theorem 4.1. *Suppose a vector X satisfies the DCPP model in (4.9), but is contanminated with noise E, where $\|E\|_2 \leq \epsilon$, resulting in $Y = X + E$. Suppose $\{\Gamma_i^*, P_i^*\}_{i=1}^L$ solves the problem in (4.9) and $\{\hat{\Gamma}_i, \hat{P}_i\}_{i=1}^L$ solves the problem in (4.10). If*

$$\begin{aligned}
&\|\Gamma_i^*\|_{0,\infty}^s \leq \lambda_i < \frac{1}{2}(1 + \frac{1}{\mu(D_i)}), \quad \forall 0 \leq i \leq L,\\
&\epsilon_0 = \epsilon, \quad \epsilon_i^2 = \frac{4\epsilon_{i-1}^2}{1 - (2\|\Gamma_i^*\|_{0,\infty}^s - 1)\mu(D_1)} \quad \forall i \geq 1,\\
&\textit{then for all} \quad 1 \leq i \leq L,\\
&\|P_i^* - \hat{P}_i\|_2^2 \leq \|\Gamma_i^* - \hat{\Gamma}_i\|_2^2 \leq \epsilon_i^2.
\end{aligned} \tag{4.11}$$

Here maxfun pooling is used as the pooling operation and we assume that the minimum pooling region size $b \geq 2$.

In order to prove Theorem 4.1, we first prove the following Lemma for maxfun pooling.

Lemma 4.2. *Let X and $\hat{X}$ be two functions in $\mathbb{R}^{N\times N}$, and let $P = Mf_{b,s}(X)$, $\hat{P} = Mf_{b,s}(\hat{X})$ be the outcome of maxfun pooling of X and $\hat{X}$,respectively, and assume that $s \geq b$. Then $\|P - \hat{P}\|_2 \leq \|X - \hat{X}\|_2$.*

Proof. Let $B_{k,j,n}$ be the set of indices that represents the nth sub-square region of side length j in the kth pooling region. Let $\gamma_{k,j,n} = \frac{1}{j^2}\sum_{i\in B_{k,j,n}} X_i$, and $\hat{\gamma}_{k,j,n} = \frac{1}{j^2}\sum_{i\in B_{k,j,n}} \hat{X}_i$. Let $j_k^* = \underset{1<j\leq b, 1\leq n\leq (b-j)^2)}{argmax} \gamma_{k,j,n}$ be the index of the maximum of $\gamma_{k,j,n}$

over all j and n for each k, and let $\hat{j}_k^* = \underset{1<j\leq b 1\leq n\leq(b-j+1)^2}{argmax} \hat{\gamma}_{k,j,n}$ be the index of the maximum of $\hat{\gamma}_{k,j,n}$ over all j and n for each k. Let j_{min}^* be the minimum of j_k^* for all k, and let $\hat{j}_{min}^*$ be the minimum of $\hat{j}_k^*$ for all k. Let K_1 be the set of indices of k so that $\gamma_{k,j_k^*} \geq \hat{\gamma}_{k,\hat{j}_k^*}$. Let K_2 be the set of indices of k so that $\gamma_{k,j_k^*} < \hat{\gamma}_{k,\hat{j}_k^*}$.

$$\|P-\hat{P}\|_2^2 = \sum_k \left(\max_{\substack{1<j\leq b \\ 1<n\leq(b-j+1)^2}} \left(\frac{1}{j^2}\sum_{i\in B_{k,j,n}} X_i\right) - \max_{\substack{1<j\leq b \\ 1\leq n\leq(b-j+1)^2}} \left(\frac{1}{j^2}\sum_{i\in B_{k,j,n}} \hat{X}_i\right) \right)^2 \tag{4.12}$$

$$= \sum_k \left(\max_{\substack{1<j\leq b \\ 1\leq n\leq(b-j+1)^2}} \gamma_{k,j,n} - \max_{\substack{1<j\leq b \\ 1\leq n\leq(b-j+1)^2}} \hat{\gamma}_{k,j,n} \right)^2 \tag{4.13}$$

$$= \sum_{k\in K_1} \left(\max_{\substack{1<j\leq b \\ 1\leq n\leq(b-j+1)^2}} \gamma_{k,j,n} - \max_{\substack{1<j\leq b \\ 1\leq n\leq(b-j+1)^2}} \hat{\gamma}_{k,j,n} \right)^2 + \tag{4.14}$$

$$\sum_{k\in K_2} \left(\max_{\substack{1<j\leq b \\ 1\leq n\leq(b-j+1)^2}} \hat{\gamma}_{k,j,n} - \max_{\substack{1<j\leq b \\ 1\leq n\leq(b-j+1)^2}} \gamma_{k,j,n} \right)^2 \tag{4.15}$$

$$= \sum_{k\in K_1} (\gamma_{k,j_k^*} - \hat{\gamma}_{k,\hat{j}_k^*})^2 + \sum_{k\in K_2} (\hat{\gamma}_{k,\hat{j}_k^*} - \gamma_{k,j_k^*})^2 \tag{4.16}$$

$$\leq \sum_{k\in K_1} (\gamma_{k,j_k^*} - \hat{\gamma}_{k,j_k^*})^2 + \sum_{k\in K_2} (\hat{\gamma}_{k,\hat{j}_k^*} - \gamma_{k,\hat{j}_k^*})^2 \tag{4.17}$$

$$= \sum_{k\in K_1} \left(\frac{1}{(j_k^*)^2} \sum_{i\in B_{k,j_k^*}} X_i - \frac{1}{(j_k^*)^2} \sum_{i\in B_{k,j_k^*}} \hat{X}_i \right)^2 + \tag{4.18}$$

$$\sum_{k\in K_2} \left(\frac{1}{(\hat{j}_k^*)^2} \sum_{i\in B_{k,\hat{j}_k^*}} \hat{X}_i - \frac{1}{(\hat{j}_k^*)^2} \sum_{i\in B_{k,\hat{j}_k^*}} X_i \right)^2 \tag{4.19}$$

$$\leq \sum_{k\in K_1} \frac{1}{(j_k^*)^2} \sum_{i\in B_{k,j_k^*}} (X_i - \hat{X}_i)^2 + \sum_{k\in K_2} \frac{1}{(\hat{j}_k^*)^2} \sum_{i\in B_{k,\hat{j}_k^*}} (\hat{X}_i - X_i)^2 \tag{4.20}$$

$$\leq \frac{1}{(j_{min}^*)^2} \sum_{k\in K_1, i\in B_{k,j_k^*}} (X_i - \hat{X}_i)^2 + \frac{1}{(\hat{j}_{min}^*)^2} \sum_{k\in K_2, i\in B_{k,\hat{j}_k^*}} (\hat{X}_i - X_i)^2 \tag{4.21}$$

$$\leq \sum_{k\in K_1, i\in B_{k,s}} (X_i - \hat{X}_i)^2 + \sum_{k\in K_2, i\in B_{k,s}} (\hat{X}_i - X_i)^2 \tag{4.22}$$

$$=\sum_{i=1}^{N^2}(X_i - \hat{X}_i)^2 \tag{4.23}$$

$$=\|X - \hat{X}\|_2^2. \tag{4.24}$$

The inequality (4.17) comes from the fact that $\hat{\gamma}_{k,\hat{j}_k^*}$ is the maximum over all j and n and thus $\hat{\gamma}_{k,\hat{j}_k^*} \geq \hat{\gamma}_{k,j_k^*}$, and similarly $\gamma_{k,j*_k} \geq \hat{\gamma}_{k,j_k^*}$. In (4.18), B_{k,j_k^*} and $B_{k,\hat{j}_k^*}$ are the corresponding set of indices for which γ_{k,j_k^*} and $\hat{\gamma}_{k,\hat{j}_k^*}$ are maximums across all j's and n's, respectively. The inequality (4.20) holds based on the inequality $(\sum_{i=1}^n a_i)^2 \leq n \sum a_i^2$. Inequality (4.22) follows from the fact that stride size $s \geq b$. $B_{k,s}$ represents the indices of the sub-square of length s at the initial position in the kth pooling region. □

We now prove Theorem 4.1.

Proof. By Theorem 3 in [98], we know that for a signal $Y = X + E$, if

$$1. \|\Gamma_1^*\|_{0,\infty}^s < \frac{1}{2}(1 + \frac{1}{\mu(D_1)}) \text{ and } \|E\|_2 = \|Y - D_1\Gamma_1^*\|_2 \leq \epsilon_0,$$

$$2. \|\hat{\Gamma}_1\|_{0,\infty}^s < \frac{1}{2}(1 + \frac{1}{\mu(D_1)}) \text{ and } \|Y - D_1\hat{\Gamma}_1\|_2 \leq \epsilon_0,$$

then

$$\|\Delta\|_1^2 = \|\Gamma_1^* - \hat{\Gamma}_1\|_2^2 \leq \frac{4\epsilon_0^2}{1 - (2\|\Gamma_1\|_{0,\infty}^s - 1)\mu(D_1)} = \epsilon_1^2. \tag{4.25}$$

Since $\|\Gamma_1^*\|_{0,\infty}^s \leq \lambda_1$ and $\|\hat{\Gamma}_1^*\|_{0,\infty}^s \leq \lambda_1$ by assumption in problem 4.9 and 4.10, and λ is bounded by assumption 4.11 in theorem 4.1, the first parts of 1 and 2 hold. Since Γ_1^* is the solution to the $DCPP$ problem, it must be true that $\|Y - D_1\Gamma_1^*\| \leq \epsilon_0$. $\|Y - D_1\hat{\Gamma}_1\| \leq \epsilon_0$ by assumption in problem 4.10. Therefore, we have $\|\Delta\|_2^2 \leq \epsilon_1^2$. And hence by Lemma 4.2, we have

$$\|P_1^* - \hat{P}_1\|_2^2 \leq \|\Delta_1\|_2^2 \leq \epsilon_1^2. \tag{4.26}$$

At second level, the same argument holds so that $\|\Gamma_2^*\| < \lambda_2 < \frac{1}{2}(1 + \frac{1}{\mu(D_1)})$ and $\|\hat{\Gamma}_2\| < \lambda_2 < \frac{1}{2}(1 + \frac{1}{\mu(D_1)})$. $\|P_1^* - D_2\Gamma_2^*\|_2 < \epsilon_2$ by assumption in problem 4.9. $\|\hat{P}_1 - D_2\hat{\Gamma}_2\|_2 < \epsilon_2$ by assumption in problem 4.10. Therefore, by theorem 3 in [98] we have

$$\|\Gamma_2^* - \hat{\Gamma}_2\|_2^2 \leq \epsilon_2^2, \tag{4.27}$$

and by Lemma 1, we get

$$\|P_2^* - \hat{P}_2\|_2^2 \leq \|\Gamma_2^* - \hat{\Gamma}_2\|_2^2 \leq \epsilon_2^2. \tag{4.28}$$

Following this argument for $1 \leq i \leq L$, we complete the proof and showed that

$$\|P_i^* - \hat{P}_i\|_2^2 \leq \|\Gamma_i^* - \hat{\Gamma}_i\|_2^2 \leq \epsilon_i^2, \quad \forall \quad 1 \leq i \leq L. \tag{4.29}$$

□

4.3 Experiments

Now that we have shown that the maxfun pooing function preserves the stability of deep neural networks when analyzed via deep convolutional sparse coding scheme, we conduct experiments using convolutional neural networks on a standard data set to test the performance of maxfun pooling on classification tasks compared with other pooling strategies. Mike Pekala is the major contributor to the experimental results from this Section.

To test the performance of maxfun pooling, we produce features of image data sets by running through shallow layers of convolutional neural network, applying pooling functions spatially to the set of features produced in all channels, and then feeding the pooled features into a traditional classifier to determine how well the classification scheme we choose can distinguish different objects based on the pooled representations.

For image data set we use a subset of the Caltech-101 data set [38]. To reduce bias created by size differences in each object category, we restrict to the classes that have between 80 and 130 instances. Therefore we obtain an 18-class classification problem with mild class imbalance by design. For pre-processing steps, all images are first padded to become square images; e.g. a 100x120 pixel image is padded to 120x120 pixel image. Each image is kept centered when padding. In the previous example 10 rows are added to the top of the image and 10 rows are added to the bottom. Then we resize all images to 128x128 pixels so that the signal's aspect ratio is preserved.

We choose to use the features generated by first convolutional layer of the Inception-v3 network, a recently desinged highly effective network for natural image classification [113]. A set of feature images in different channels is produced by this operation, and we then spatially decompose image in each channel into possibly overlapping pooling regions. Note that this decomposition preserves the channel dimension, which is the dimension of features produced for each pixel location.

After producing features, various pooling strategies are used to reduce each pooling region to a scalar value. The vectorized representation of these pooling outputs becomes the feature representation to be fed into a classification scheme. We choose a classic method, the support vector machine (SVM) [54] and use one versus one type of classification to produce final results.

We compare the maxfun pooling to conventional pooling functions max pooling and average pooling, and also some novel pooling strategies: stochastic pooling [119] and "mixed" pooling [78]. For the stochastic pooling implementation we use the probabilistic averaging method

$$s_j = \sum_{i \in R_j} p_i X_i,$$

where s_j denotes the pooled value for jth pooling region and $p_i X_i$ is a weighted average of activations in the pooling region [119]. Note that parameters p_i's are randomly chosen within each pooling region according to a multinomial distribution based on the activities within the pooling region. Mixed pooling is defined as a linear combination of maximum and average pooling

$$s_j = \alpha Pool_{max}(R_i) + (1 - \alpha) Pool_{avg}(R_i),$$

where R_i denotes the ith pooling region, $Pool_{max}$ denotes the maximum pooling, $Pool_{avg}$ denotes average pooling and α is a scalar in $[0, 1]$.

Both the maxfun pooling and the mixed pooling strategies entail hyperparameter selection; in the case of the maximal function we select the minimum support cardinality r of j in (4.2) while for the mixed pooling strategy we must select the scalar α. In both cases we use a k-fold cross-validation procedure with $k = 3$ to select these hyperparameters. Our training and testing data sets are of size 975 and 649 with the partition chosen uniformly at random. The pooling window sizes and resulting classification accuracy are shown in Table 4.1. Note that we implemented a centered version (CV) of the maxfun pooling, in which we only compute the averages of the sub-squares that are centered in each pooling region.

pooling strategy	SVM accuracy	
	$b = 21, s = 21$	$b = 21, s = 11$
average	0.5763	0.6102
maximum	0.5932	0.5932
mixed	0.6287	0.6240
stochastic	0.6502	0.6641
maxfun	0.6225	0.6102
maxfun + CV	0.6626	0.6579

Table 4.1: Empirical results for various pooling strategies. Pooling regions have dimensions $b \times b$ and the pooling stride is s. When $s = r$, the pooling regions partition the spatial dimensions of the image, i.e. there are no overlaps in pooling regions.

The results of 4.1 suggest that, in the context of our underlying assumptions, the maxfun pooling strategy provides consistently good results. Note that the classification results that we obtain on this data set do not match with the state-of-the-art classification results of this data set, as our goal is not to outperform existing algorithms for classification on this data set, but to compare various pooling strategies and assess their ability to represent images effectively. Also note that in order to distinguish the behaviors of maxfun pooling, max pooling and average pooling which are similar when pooling size b is small, we choose relatively large pooling regions to manifest the differences between these pooling strategies.

4.4 Conclusions

In this Chapter we introduce a novel pooling method, maxfun pooling, inspired by the maximal function in harmonic analysis, prove that it maintains the stability of a convolutional neural network and demonstrate its experimental performances by comparing it with state of the art pooling strategies in classification tasks.

Maxfun pooling is a strategy that effectively extracts features from outputs of layers of neural networks and produces good representation of images. In fact, many functions and transformations originated from harmonic analysis have the ability to extract useful features from data sets for classification or segmentation purposes. In the next Chapter, we dig into the problem of chemical molecule representation in quantum energy regression using a novel feature extraction method using the Gabor transform.

Chapter 5: Gabor Regression of Quantum Chemical Energies

5.1 Introduction

Computation of the energy of a chemical molecule using the charges and relative positions of its atoms has become a crucial topic in computational chemistry. It can be applied to various industrial scenarios such as prediction of the thermodynamics and kinectics of chemical reactions [34]. There are limitations that hinder the exact computation of molecular energies for most chemical molecules. The computation of ground state molecular energies is made possible via the density functional theory (DFT) [110] by transforming the computation problem into a variational problem over the total electronic density [60]. Due to the computational complexity using DFT, machine learning methods have been developed recently to approximate ground state energy of chemical molecules, see, e.g., [52], [88], [105], [108]. The machine learning methods focus on finding proper representations of chemical molecules with desired invariant properties [7], such that the dimensionality of the representation is reduced while the properties of the molecules are preserved.

Using machine learning method, the problem of computing ground state energy of chemical molecules splits into two separate tasks: effective representation of the molecules and means to approximate the objective function based on such representation. Common methods that are used to approximate the energy are regressions of different forms. Unique and efficient representations of the molecules are developed based on the geometry and properties of molecules. The quality of the representation often determines the accuracy of the approximation achieved by regression models.

Methods based on Coulomb matrix [52], [105] , wavelet scattering transform [58], [59], and deep neural networks [88], [108], have been developed to represent chemical molecules. Coulomb matrix representation is introduced in 2012, but it faces problems of non-invariance to atom permutations [105]. A modification of Coulomb representation comes in 2013 and demonstrates significant improvements on performance [52]. In 2015, Hirn, Mallat, Poilvert implement scattering transform to represent chemical molecules on 2D molecules and demonstrate improvements upon Coulomb matrix representation for 2D molecules [59]. The results for scattering transform is improved in 2017 [58] using symmetries in wavelets and second level scattering transform. In 2017, a method using deep tensor neural networks for quantum energy regression is developed and demonstrates the best performance up to date [108].

In this Chapter, we introduce a novel method based on the Gabor transform to represent chemical molecules in 2D in an invariant way. Gabor transform, or the short time Fourier transform [49], has been used as an effective tool for data representation in various applications [29], [96]. Gabor transform captures local information of data by entailing frequency information of signal at different locations. We design Gabor invariant transform which takes electron density of molecules [59] as input, and test by sparsely regressing on the representation to predict the energy of a chemical molecule. Our result shows that Gabor invariant representation is a generic method that performs at the level of state-of-the-art representation methods for 2D planar molecules. The Gabor invariant representation method does not perform as well as the improved scattering representation introduced in 2017 [58] for 2D planar molecules, but it outperforms the state-of-the-art Coulomb matrix representation, and can be extended to represent 3D molecules, which is the natural setting for chemical molecule representation. The Gabor invariant transform preserves essential properties of the chemical molecules, and achieves desired precision without the need of large amount of training data, or careful tuning of parameters pertaining to a particular data set.

We organize the remaining of this Chapter as follows. In Section 5.2, we illustrate types of desired invariant properties of a chemical molecule, and introduce experimental background for quantum energy regression [58]. Our invariant representation method, the Gabor invariant representation, is introduced in section 5.3, along with its invariant properties and applications. In section 5.4, we describe the regression method used for energy prediction. Section 5.5 includes details of the state of the art invariant representations of chemical molecules introduced in the paper [58]. Section 5.6 has the details of experimental setup and results. We will analyze the results and discuss properties of the Gabor invariant representation, including its merits, drawbacks, and potential extensions.

5.2 Background on Quantum Energy Regression

In this Section we introduce the background on quantum chemical energy regression. In particular, we discuss the problem of computing energy of chemical molecules in machine learning, desired properties of representation of molecules, and computation of electronic densities of molecules.

5.2.1 Computation of quantum chemical energy

Computation of energy of a single chemical molecule has become an essential topic in computational chemistry. A chemical molecule is represented by its state $x = \{r_k, z_k\}_k$, where $r_k \in \mathbb{R}^3$ is the position of the kth nuclei and $z_k > 0$ is the charge of kth nuclei. Approaches based on DFT compute the quantum energy of a molecule, denoted as $f(x)$. In machine learning, one way to avoid direct computation from DFT is to build set of dictionaries of functions $\Phi(x) = \{\phi_i(x)\}_{i=1}^N$ such that the energy $f(x)$ can be approximated by $\tilde{f}(x)$, where

$$\tilde{f}(x) = \langle x, \Phi(x) \rangle = \sum_{i=1}^{N} w_i \phi_i(x),$$

and the weights $\{w_i\}$ are computed such that the error $\sum_{j=1}^n \left| \tilde{f}(x_j) - f(x_j) \right|^2$ on the training data set of size n is minimized.

With the weights $\{w_i\}$ computed from a training sample $\{x_i\}_i$, given any input of a chemical molecule x' with unknown energy $f(x')$, one can approximate its energy by first computing its dictionaries $\Phi(x') = \{\phi_i(x')\}_i$, and then applying weights on the dictionaries accordingly to get $\tilde{f}(x') = \sum_i w_i \phi_i(x')$.

5.2.2 Invariant properties of chemical molecules

The ground state energy is unique for individual molecule and is closely related to the chemical properties of individual molecules [59]. It is important to capture unique properties of each molecule in its representation, because prediction quality of the regression on the dictionaries depends highly on the degree to which dictionaries preserve unique properties of the molecules.

Invariant properties are important features of chemical molecules. Let $x = \{r_k, z_k : r_k \in \mathbb{R}^3, z_k \in \mathbb{R}, z_k > 0\}_{k=1}^{K}$ represents a molecule of K atoms at positions r_k with atomic charges z_k. The quantum energy $f(x)$ of molecule x must satisfy the following invariant properties [59]:

- **Permutation invariance** The energy functional $f(x)$ is invariant under permutation of indices $\{k = 1, ..., K\}$ in $x = \{r_k, z_k\}_k$. The ordering of atoms in the chemical molecule in the representation does not influence the energy of the molecule.

- **Isometric invariance** The energy functional $f(x)$ is invariant under global translations, rotations, and symmetries of atomic positions r_k. Rotating, translating and reflecting molecule does not resulting in having a different molecule, and thus the energy of the molecule should not be influenced by these operations [7].

In order to make precise prediction on the ground state energy functional, it is necessary to ensure that the representation follow the same invariant properties as the objective functional. The Gabor invariant representation takes account into permutational and isometric invariances. We compute the Gabor invariant representation of a molecule from its electron density.

5.2.3 Electron density approximation

The molecular energy E can be written as a functional of the electron density $\rho(u) \geq 0$ which specifies the density of electronic charge at every position $u \in \mathbb{R}^3$. The ground state energy $f(x)$ which is unique for every molecule x, can be obtained by minimizing energy E over a set of electronic densities ρ:

$$f(x) = E(\rho_x) = \inf_{\rho} E(\rho).$$

An important property of neutral molecules is that the total electrons integrate up to the summation of its atomic charges, i.e., $\int \rho_x dx = \sum_k z_k$.

Computing ρ_x is a challenging problem that is as difficult as computing ground state energy $f(x)$. Therefore, we approximate electronic density ρ_x by $\tilde{\rho}$, where $\tilde{\rho}$ also satisfies $\int \tilde{\rho}_x dx = \sum_k z_k$. The approximation $\tilde{\rho}$ is constructed so that it is invariant under permutation of atoms:

$$\tilde{\rho}_x = \sum_{k=1}^{K} \rho_{atom}^{a(k)}(u - p_k).$$

In other words, $\tilde{\rho}$ is a linear superposition of individual atomic densities. If the molecule x has K atoms, each at position p_k, and ordered by $\{a(k)\}_{k=1}^{K}$, then $\rho_{atom}^{a(k)}(u)$ represents the atomic density of atom $a(k)$. Atomic densities are shifted to center at atomic positions p_k, and their summation gives the approximate electronic density of molecule x. Figure 5.1 gives an example of electron density and its approximation [59].

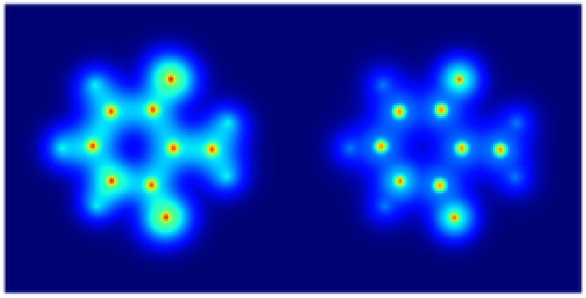

Figure 5.1: Left: ground state electron density ρ_x and right: approximate electron density $\tilde{\rho}_x$ [59]

A refinement of the approximation is to separate the densities of the core electrons ρ_{cor} from the densities of the valence electrons ρ_{val}. The core electrons stay close to the nuclei and do not interact between different nuclei, while the valence electrons will form chemical bonds and interact with each other. This separation facilitates in differentiating types of electrons and has demonstrated to improve numerical experiment results in [58]. An illustration is shown in Figure 5.2 [58].

The approximate density $\tilde{\rho}_x$ is invariant under permutation of atom indices, but it is not invariant under rotations, reflections and translation. Therefore we need to construct a set of dictionaries $\Phi(\tilde{\rho}_x) = \{\phi_k(\tilde{\rho}_x)\}_k$ which satisfies rotational invariance to approximate the ground state energy $f(x)$:

$$f(x) = \sum_k \omega_k \phi_k(\tilde{\rho}_x).$$

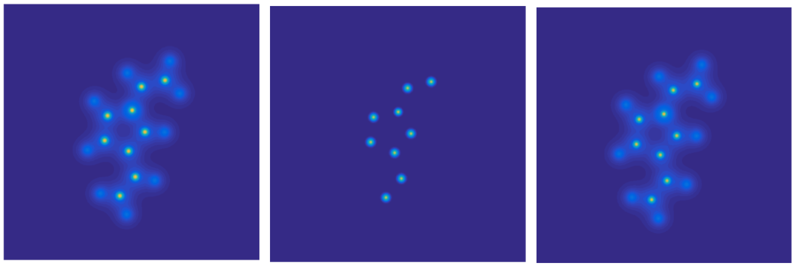

Figure 5.2: Left: atomic density; middle: core density; right: valence density. [58]

5.3 Gabor Invariant Representation

Gabor invariant transform takes in electron densities of molecules and creates a set of dictionaries that is rotational and permutation invariant. We define Gabor invariant transform mathematically and derive its invariant properties when it is applied to the electron density of molecule.

5.3.1 2D Gabor Invariant Representation

A Gabor transform on 2D is a type of short time Fourier transform, which measures centered phase and frequency information of an image as location of the center changes over time. For a given function $\rho(x)$, $\rho(x)$ is first centered by multiplication of a window function $g(x-t)$, and then it takes Fourier transform. Formally, Gabor transform $V_g\rho(t,\gamma)$ for a function ρ is defined in 2.12:

$$V_g\rho(t,\gamma) = \int \rho(x)\overline{g(x-t)}e^{-2\pi i x\gamma}dx,$$

where t is the center location of the window and γ specifies frequency of interest. Common window function is the Gaussian function. For simple notation, we denote

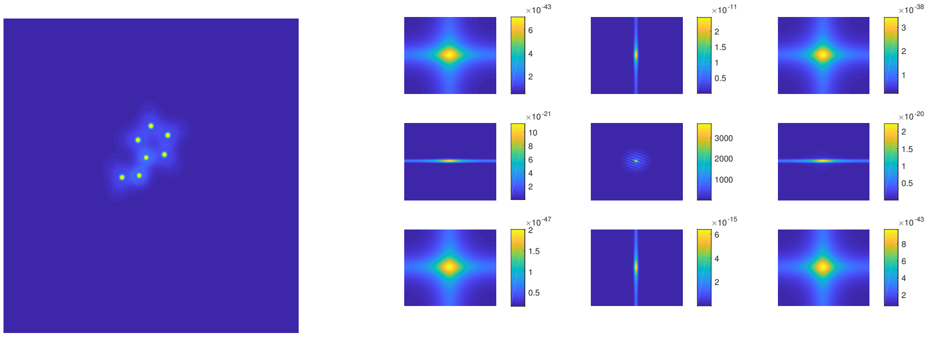

Figure 5.3: Atomic density (left) and the Gabor transform at selected pixel locations (right)

the Gabor transform $V_g\rho$ of ρ by window function g as G_ρ.

Compared with Fourier transform, the multiplication of a Gaussian $g(x-t)$ concentrates function $\rho(x)$ around t and reduces the amplitude of $\rho(x)$ when x decays from t. In this way, we capture information locally, instead of obtaining overall information of the entire image [49]. Figure 5.3 shows the Gabor transform of the electronic density of a chemical molecule at different pixel locations.

The Gabor transform is not translation invariant, because it gathers information at each location point t. Since Gabor transform is taken uniformly on the image at locations t, translating location does not impact the outcome of the integral and thus translation invariance is achieved. In fact, for a given Gaussian function $g(x)$, $G_{\tau_\rho}(t,\gamma) = e^{-2\pi i\tau\gamma}G_\rho(t-\tau,\gamma)$, where $\tau_\rho(x) = f(x-\tau)$ denotes the translation of f by τ. We can take the modulus of $G_\rho(t,\gamma)$ and integrating over all t:

$$G_\rho(\gamma) = \int_{\mathbb{R}^3} |G_\rho(t,\gamma)| dt.$$

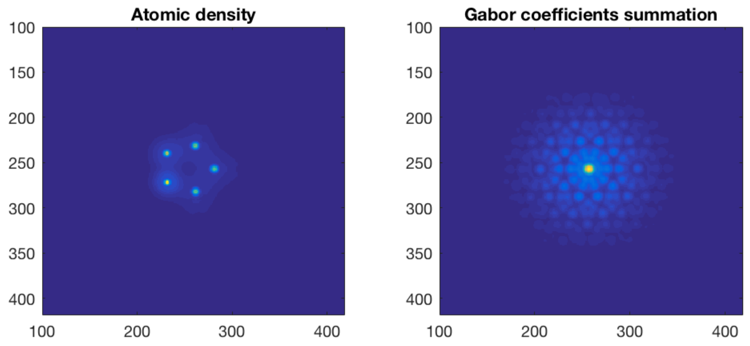

Figure 5.4: Atomic density and Gabor coefficients integrated over translation

By taking the modulus and integrating over all t, we get $G_{\tau_\rho}(\gamma) = \int_{\mathbb{R}^3} |e^{-2\pi i \tau\gamma} G_\rho(t-\tau, \gamma)| dt = \int_{\mathbb{R}^3} |G_\rho(t, \gamma)| dt = G_\rho(\gamma)$. Therefore translation invariance is obtained. Figure 5.4 gives an example of the original electron density and integration of Gabor transform of the electron density over translations.

A rotation in function f yields a rotation in $G_\rho(\gamma)$. In order to obtain rotation invariance in the representation, we need to ensure that the Gabor invariant transform is rotaional invariant. This can be achieved by integrating over every circular orbit around the center of each density representation, and averaging over the length of the orbit to get a vector representation of each molecule.

Let (α, η) be the spherical coordinates of γ, with $|\gamma| = \alpha$, and $\eta \in S^2$ denotes the rotation orbit. If we write $G_\rho(\gamma) = G_{\rho,\alpha}(\eta)$, then rotation invariance of $G_\rho(\gamma)$ is obtained by averaging over each η:

$$\|G_{\rho,\alpha}\|_2^2 = \int_{S^2} |G_{\rho,\alpha}(\eta)|^2 \, d\eta = \int_{S^2} \left| \int_{\mathbb{R}^3} |G_\rho(t, \gamma)| \, dt \right|^2 d\eta.$$

To approximate exchange correlation terms in quantum energy functionals, whose resulting energy grows more linearly with the number of electrons than quadratically

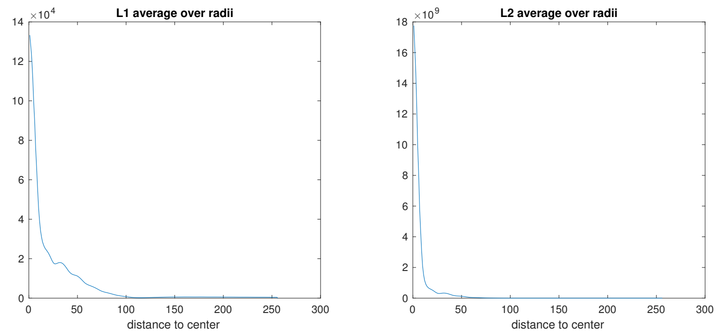

Figure 5.5: Rotational invariant representations

[58], we also include the L^1 norm:

$$\|G_{\rho,\alpha}\|_1 = \int_{S^2} |G_{\rho,\alpha}(\eta)| d\eta = \int_{S^2} \left| \int_{\mathbb{R}^3} |G_\rho(t,\gamma)|\, dt \right| d\eta.$$

The radial frequency parameter α is sampled at intervals ϵ over a frequency range $\alpha \in [\epsilon, \epsilon^{-1}]$. Figure 5.5 gives a demonstration of the rotational representation of a chemical molecule. Note that since the input is translation invariant, it is also translation invariant.

The Gaussian function g concentrates f at t, and the variance of Gaussian function measures the range in which information of f is taken. To capture information of f of different widths at t, we adopt two different Gaussian functions g_1 and g_2, and we denote G^1 and G^2 to be their corresponding Gabor transforms.The Gabor modulus dictionary is defined as:

$$\Phi_\rho = \{\|\rho\|_1, \|G^1_{\rho,k\epsilon}\|_1, \|G^1_{\rho,k\epsilon}\|_2^2, \|G^2_{\rho,k\epsilon}\|_1, \|G^2_{\rho,k\epsilon}\|_2^2\}_{0\le k\le \epsilon^{-2}}.$$

A Gabor invariant transform is translation and rotation invariant, and it captures local information of electronic density function of a molecule. Because interaction between atoms is crucial in defining chemical properties of a molecule, which is tight to the ground state energy of the molecule, increment in the ability to gather local information improves quality of the representation and yields better regression results. Details will be discussed in Section 5.6.

5.4 Energy Regression Model

5.4.1 Sparse Regression by Orthogonal Least Square

We use sparse regression in dictionaries $\Phi(x) = \{\phi_k(x)\}_k$ which are adapted to the properties of quantum energy functions $f(x)$ [59]. Given a training set of N molecular state vectors and associated energies $\{x_i, f(x_i)\}_{i=1}^N$, we compute the sparse regression by restricting the number of nonzero weights ω_k:

$$\tilde{f}(x) = \langle \omega, \Phi(x) \rangle = \sum_{m=1}^{M} \omega_m \phi_{k_m}(x),$$

where M is less than total number of dictionaries N for each molecule x. To account for correlations between functions in dictionaries, we apply greedy orthogonal least square forward selection algorithm [20] to select members of the dictionary. The objective is to minimize the quadratic error function on the training set:

$$\sum_{i=1}^{N} \left| \sum_{m=1}^{M} \omega_m \phi_{k_m}(x_i) - f(x_i) \right|^2 .$$

A greedy orthogonal least square algorithm selects one regression vector at a time, and orthogonalizes the remaining dictionary relatively to the previously selected vectors, i.e., the remaining dictionary is decorrelated from the perviously selected vectors. Let $\{\phi_k^m\}_k$ be the set of decorrelated dictionary at iteration m. Let $\{\phi_{k_n}^n\}_{n=1}^m-1$

be the set of previously chosen regression vectors. We select one new regression vector from $\{\phi_k^m\}_k$, say $\phi_{k_m}^m$, so that the quadratic error is minimized when we add the new vector to the regression terms:

$$\sum_{i=1}^{N} \left| f_m(x_i) - f(x_i) \right|^2, \quad \text{where} \quad f_m(x) = \sum_{n=1}^{m} \tilde{\omega}_n \phi_{k_n}^n(x).$$

$f_m(x)$ is the projection on the first m selected dictionaries, and the weight coefficient of each $\phi_{k_n}^n$ is computed by $\tilde{\omega}_n = \left\langle f(x), \phi_{k_n}^n(x) \right\rangle = \sum_{i=1}^N f(x_i)\phi_{k_n}^n(x_i)$. Since all selected dictionaries are orthogonal, the quadratic error can be written as

$$\sum_{i=1}^{N} |f_m(x_i) - f(x_i)|^2 = \sum_{i=1}^{N} |f(x_i)|^2 - \sum_{n=1}^{m} |\tilde{\omega}_m|^2,$$

and the error is minimized by choosing $\phi_{k_m}^m$ which maximizes its correlation with $f(x)$:

$$k_m = \arg\max_k \sum_i f(x_i)\phi_{k_m}(x_i).$$

After $\phi_{k_m}^m$ is chosen from $\{\phi_k^m\}_k$, we orthogonalize each remaining ϕ_k^m with respect to $\phi_{k_m}^m$ over the training set $x = (x_i)_i$:

$$\tilde{\phi}_k^{m+1} = \phi_k^m - \left\langle \phi_{k_m}^m(x), \phi_k^m(x) \right\rangle \phi_{k_m}^m.$$

Each decorrelated vector $\tilde{\phi}_k^{m+1}$ is then normalized to define the updated dictionary ϕ_k^{m+1}:

$$\phi_k^{m+1} = \tilde{\phi}_k^{m+1} \|\tilde{\phi}_k^{m+1}(x)\|_2^{-1},$$

where $\|\tilde{\phi}_k^{m+1}(x)\| = (\sum_i |\tilde{\phi}_k^{m+1}|^2)^{\frac{1}{2}}$.

The algorithm terminates when we reach the number of regression vectors M, which is chosen based on a cross validation. we obtain a set of orthonormal dictionaries

$\{\phi_{k_n}^n\}_{1\leq n\leq M}$, and the final M-term regression can be written as a function of the dictionary $\Phi = \{\phi_{k_m}^m\}_m$:

$$f_M(x) = \sum_{m=1}^{M} \tilde{\omega}_m \phi_{k_m}^m = \sum_{m=1}^{M} \omega_m \phi_{k_m}.$$

This algorithm can be implemented by QR factorization, or implemented directly as described above [14]. It gives a orthogonal least square regression of M vectors in Φ. We introduce existing representation method of chemical molecules in the next Section, mainly the ones introduced in [52] and in [59].

5.5 Other molecular representations

In this Section we describe some of the existing methods used in representation of chemical molecules for energy regression. In particular, we mention the representations which we compare with in Section 5.6.

5.5.1 Coulomb matrix representation

Methods to compute dictionaries have been developed and incorporate invariant properties of chemical molecules. One of the state-of-the-art method generates the Coulomb matrix, which is introduced by Rupp, et al [105]. Coulomb matrix representation represents a molecule by a matrix of distance, and adopts kernel ridge regression from kernels computed from Coulomb matrices. Coulomb matrix of a molecule x is defined by:

$$c_{k,l} = \begin{cases} \frac{1}{2} z_k^{2.4} & k = l, \\ \frac{z_k z_l}{|r_k - r_l|}, & k \neq l. \end{cases} \tag{5.1}$$

Coulomb kernel is computed by

$$K(x,x') = \exp(-\frac{1}{\sigma}\sum_{k,l}|c_{k,l}(x) - c_{k,l}(x')|). \tag{5.2}$$

If two molecules have different number of atoms, the Coulomb matrix of the molecule with smaller number of atoms is extended with zeros.

The Coulomb matrix representation is invariant under isometry, and stable under small perturbation of atomic positions r_k, but it changes form under permutation of indices of atoms and is not permutation invariant. Modifications can be made to reduce the effect of permutation and the kernel ridge regression using Coulomb kernels gives good prediction results [52].

5.5.2 Invariant Wavelet Modulus and Multiscale Scattering

In this Subsection we describe invariant wavelet modulus and invariant multiscale scattering representation from [59]. Scattering transform is introduced by S. Mallat in [82]. It has been proven to possess invariant properties and stability under small deformations. It can be viewed as a cascade of wavelet transform with modulus taking at different scales and locations. Wavelet transform obtains information at different scales and orientations. Given a mother wavelet $\psi : \mathbb{R}^3 \to \mathbb{C}$, we can compute a sequence of dilated and rotated wavelets by

$$\psi_{j,\theta}(u) = 2^{-3j}\psi(2^{-j}r_\theta^{-1}u).$$

where r_θ denotes an rotation with angular parameter θ. The wavelet transform of a function $\rho(x)$ is defined as $\rho * \psi_{j,\theta}$ where $*$ is the convolution operation. We compute the invariants from the modulus of wavelet coefficients $|\rho*\psi_{j,\theta}|$. Let $\psi^*(u) = \psi(-u)$ be the complex conjugate of $\psi(u)$, then $|\rho * \psi_{j,-\theta}| = |\rho * \psi_{j,\theta}|$. Therefore the wavelet $\psi_{j,\theta}$ only depends on the rotation modulo a sign. We can index the rotation parameter r_θ by a two-dimensional Euler angular parametrisation $\theta \in [0,\pi]^2$ of the half sphere.

Wavelet coefficients are computed up to a maximum scale of 2^J. Lower frequencies are captured by a low-pass filter $\phi_J(u) = 2^{-3J}\phi(2^{-J}u)$ where $\phi(u) \geq 0$ and $\int \phi(u) = 1$. The wavelet transform is defined as

$$W_\rho = \{\rho * \phi_J, \rho * \psi_{j,\theta}\}_{j<J,\theta\in[0,\pi]^2}.$$

The wavelet transform is invariant to reflections, but it is not invariant under translations and rotations. To account for translation and rotation invariances, we integrate over the translation and rotation variables in L^1 norm where

$$\|\rho * \psi_{j,\cdot}\|_1 = \int_{\mathbb{R}^3}\int_{[0,\pi]^2} |\rho * \psi_{j,\theta}(u)| d\theta du,$$

and in L^2 norm where

$$\|\rho * \psi_{j,\cdot}\|_2^2 = \int_{\mathbb{R}^3}\int_{[0,\pi]^2} |\rho * \psi_{j,\theta}(u)|^2 d\theta du.$$

The invariant dictionary of wavelet coefficients is built up to a highest frequency ϵ^{-2} as

$$\Phi_\rho = \{\|\rho\|_1, \|\rho * \psi_{j,\cdot}\|_1, \|\rho * \psi_{j,\cdot}\|_2^2\}_{2\log_2 \epsilon < j < J}.$$

The integration over $u \in \mathbb{R}^3$ and $\theta \in [0,\pi]^2$ removes rotational and translational variability, and thus we can propagate along the paths u and θ to obtain more stable invariants. In particular, the variations of $|\rho * \psi_{j,\theta}(u)$ are represented by convolving with a second family of wavelets with different scales $2^{j'}$ and different rotation angles $\theta+\theta'$, and integrated over u and θ. L^1 and L^2 norms are computed for the dictionary:

$$\||\rho * \psi_{j,\cdot}| * \psi_{j',\theta'+\cdot}\|_1 = \int_{\mathbb{R}^3} \int_{[0,\pi]^2} ||\rho * \psi_{j,\theta}(u)| * \psi_{j',\theta'+\theta}(u)| \, du d\theta,$$

$$\||\rho * \psi_{j,\cdot}| * \psi_{j',\theta'+\cdot}\|_2^2 = \int_{\mathbb{R}^3} \int_{[0,\pi]^2} ||\rho * \psi_{j,\theta}(u)| * \psi_{j',\theta'+\theta}(u)|^2 \, du d\theta.$$

The resulting second order scattering dictionary is:

$$\Phi_\rho = \{\|\rho\|_1, \|\rho * \psi_{j,\cdot}\|_1, \|\rho * \psi_{j,\cdot}\|_2^2,$$

$$\||\rho * \psi_{j,\cdot}| * \psi_{j',\theta'+\cdot}\|_1, \||\rho * \psi_{j,\cdot}| * \psi_{j',\theta'+\cdot}\|_2^2\}_{2\log_2 \epsilon < j < j' < J, \theta' \in [0,\pi]^2}.$$

The scattering invariants are invariant under translations, rotations and permutations, and it is proven to be stable under small deformation when taken to the limit [58], [59].

5.5.3 Invariant Fourier Modulus

Fourier transform approach can define translation invariant representation and modifications can be made to account for rotation invariance. The Fourier transform $F : \mathbb{R}^3 \to \hat{\mathbb{R}}^3$ of a density function ρ is defined as:

$$\hat{\rho}(\omega) = \int_{\mathbb{R}^3} \rho(u) \cdot e^{-2\pi i \omega u} du.$$

Since the Fourier transform $\hat{\rho}(u-\tau) = e^{-2\pi i \tau \omega} \hat{\rho}(u)$, the modulus of Fourier transform is translation invariant, i.e., $|\hat{\rho}(u-\tau)| = |\hat{\rho}(u)|$. Rotation invariance is obtained by averaging over each rotation orbit, indexed by $\eta \in S^2$, where S^2 is the sphere. Since a rotation of ρ yield a rotation of $\hat{\rho}$, if we write ω in spherical coordinates (α, η), with

$|\omega| = \alpha$, and write $\hat{\rho}(\omega) = \hat{\rho}_\alpha(\eta)$, then integration over η yields rotation invariance of the Fourier modulus:

$$\|\hat{\rho}_\alpha\|_2^2 = \int_{S^2} |\hat{\rho}(\alpha\eta)|^2 d\eta.$$

To approximate exchange correlation terms in quantum energy functionals, whose resulting energy grows more linearly with the number of electrons than quadratically, L^1 norm of the Fourier modulus is included:

$$\|\hat{\rho}_\alpha\|_1 = \int_{S^2} |\hat{\rho}(\alpha\eta)| d\eta.$$

The radial frequency parameter α is sampled at intervals ϵ over a frequency range $\alpha \in [\epsilon, \epsilon^{-1}]$. The Fourier modulus dictionary is defined by

$$\Phi_\rho = \{\|\rho\|_1, \|\hat{\rho}_{k\epsilon}\|_1, \|\hat{\rho}_{k\epsilon}\|_2^2\}_{0<k<\epsilon^{-2}}.$$

A Fourier modulus is invariant under translation and rotation, but it is not stable under deformation. It does not provide us with location information of density function. Quantum energy functionals can be computed from electronic density functions, and an electronic density function is highly dependent on the position of atoms. Therefore the Fourier modulus invariant loses location information which is of importance to the unique representation of a chemical molecule [58].

5.6 Experiments and Results

5.6.1 Representation of Planar Molecules

The Gabor invariant representation can be implemented to represent 2D and 3D data. Due to the time constraint, we test our method on 2D planar molecule data and compare the regression performance of Gabor modulus dictionary with existing dictionaries described in [59].

The 2D planar electron density is obtained by finding nearly planar molecules and project each atoms onto a 2D plane while preserving the relative distances between each atomic position. Thus the electron densities are calculated with $r_k \in \mathbb{R}^2$ instead of in $\mathbb{R}^3$. The data set is provided in the paper by Hirn, et al [59].

In numerical experiments, electron densities are sampled at intervals of length ϵ. Electron densities are sampled over a square of 2^{2J} samples, with $J = 9$.

To obtain Gabor invariants, 2D Gabor transform is first computed over electron densities with translation parameter t and frequency parameter γ. Translations are sampled at intervals of length a. We take $a = 16$. Frequency parameter γ is sampled N times uniformly along each dimension. To obtain comparable results with Fourier modulus, we take $N = 2^J$ where $J = 9$. Integration over translations t and unit frequency circle S^1 is performed, yielding $2^{J-1} = 2^8$ radial Gabor invariants per density channel.

5.6.2 Numerical Comparison of Planar Molecules

The data set includes 454 nearly planar molecules among the 7165 molecules of the QM7 molecular database [59]. The molecular atomization energies in the data set are computed using hybrid density functional PBE0. The data set consists of a set of organic molecules composed of Hydrogen, Nitrogen, Oxygen and Sulfur.

We use the same sparse orthogonal least square regression method with software available in [59]. In order to evaluate the prediction error of regression for each dictionary, we split the data into five representative folds, and use five fold cross validation. Every one of the five representative folds is selected as a testing set, and the remaining four folds are selected as a training set. The procedure is repeated five times, and we get results across five different training and testing sets. Both root mean squared error(RMSE), which is the square root of the average square error, and mean absolute error (MAE), which is the average of the absolute value error,are computed, and is averaged across five train test splits.

The number of regression vector in the sparse regression,M, is selected by a bagging algorithm. Each training set is uniformly randomly split up into a training bag, and a testing bag, where training bag consists of $\beta\%$ of the training set, and the testing bag consists of $1-\beta\%$ of the training data. Orthogonal least square algorithm is applied on the training bag, with up to M_0 terms in the regression. The algorithm then select $\bar{M} \leq M_0$ which minimizes the RMSE or MAE on the testing bag. The resulting $\bar{M}$ term regression is computed on the testing data. This procedure is repeated X times, and the regressions are averaged to give final results. In numerical experiment, $\beta = 90$, $X = 10$. Results of Gabor representation, compared with Wavelet, Fourier and Scattering invariant representation, and Coulomb matrix representation are shown below:

		2D molecules from QM7		
		$\bar{M}$	RMSE	MAE
Coulomb Matrix		N/A	6.7 ±2.8	14.8 ± 12.2
Fourier	Dirac+Core/Valence	73±27	6.7±0.7	8.5±0.9
Wavelet	Dirac+Core/Valence	38±13	6.9±0.6	9.1±0.8
Scattering 15	Atomic	74	6.9	9.0
Gabor	**Atomic+Core/Valence**	**71±31**	**5.3±0.3**	**7.0±0.6**
Scattering 16	Core/Valence	107±41	3.2±0.1	4.5±0.2

Table 5.1: Average Error ± Standard Deviation over the five folds in kcal/mol

Gabor invariant representation achieves smaller Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) over Coloumb matrix, wavelet, Fourier and the first implementation of scattering invariant representations. Coloumb matrix representation is not permutaion invariant, but techniques are used to reduce the effect. It is translation and rotation invariant. Wavelet, Fourier, scattering and Gabor transform are permuation, rotation and translational invariant.

Gabor transform is based on Fourier transform, but it integrates details of frequency information at different locations, which Fourier transform could not obtain. Wavelet transform gives frequency information of the image at different scales, and scattering transform recovers part of the correlation information in second layer. On the other hand, Gabor transform captures interaction between atoms from input with no loss, and then integrate it into a form with invariant properties. Experimental results show that Gabor transform can preserve unique information of the molecules and give smaller prediction error compared with other generic representation methods.

Note that there are two columns of results for scattering transform. Scattering invariant modulus is first developed as a generic method which preserves invariances of the molecules. A newer version of the scattering invariant modulus is described in [58] which incorporates careful design of wavelet that takes into account the symmetries of the molecules. However, scattering transform, as an image processing technique, is designed for processing 2D images, while the Gabor invariant modulus can be extended to 3D with increment in computation costs.

5.6.3 Conclusion

Gabor invariant representation is a Fourier based method that gathers local information in frequency domain and integrates information in a way that is translation and rotation invariant. It is a method that can be exploited when inputs need to be represented in a unique way and when isometry invariance is a desired property of the objective function.

Gabor invariant transform outperforms state of the art invariant methods that are nonspecific to data set in quantum energy regression. It can be applied to other situations directly with no need of modification. One can also build dictionaries that are data specific. By changing the variance of the window function and the number of window functions, location specific information of various scales will be captured and more precision can be achieved.

Unlike many existing imaging techniques which lacks ability to process data in dimensions higher than 2, Gabor invariant representation can be extended to higher dimensions in theory. Computational costs of 3D Gabor invariant representation increase, and with limited amount of time we provide only 2D results in this Chapter. Many of the real world problems involve 3D molecules instead 2D molecules, and thus Gabor invariant representation can be advantageous in situations where handling multi-dimensional data is necessary.

Chapter 6: Detection of Epithelial versus Mesenchymal Regions in 2D Images of Tumor Biopsies Using Shearlets

The capability of assessing 2 dimensional images from biological tissue specimens at high resolution requires not only improved optical and biomarker methods, but also is critically dependent on mathematical techniques that enable efficient analyses of larger and more complex data sets in which positional information is accurately assessed. We present here a novel shearlet computational method that detects regions of interest in 2-dimensional tumor biopsy images, using directional information and multiscale analysis. The regions putatively correspond to epithelial or mesenchymal areas of cells, which is of critical interest to clinicians since transition from epithelia to mesenchyme promotes tumor invasion and resistance to chemotherapy. This method significantly outperformed two benchmark methods based on wavelets and shearlets.

This is a joint work with Stephen Lockett and Robert Kinders from National Cancer Institute at Frederic, MD. Lockett and Kinders provided nuclei image data with objects of interests for this project. The nuclei images were obtained at the Optical Microscopy and Analysis Laboratory at National Cancer Institute.

6.1 Introduction

With the progress of imaging techniques in bio-medical imaging and the development in computational algorithms in feature extraction and machine learning, there have been growing interests in many image recognition and classification problems that were not tackled before. We look at the problem of detecting and classifying nuclei of different shapes in confocal microscopy cancer images to study a specific type of transition of cells that is of crucial importance in developing cancer treatment.

The conversion of carcinoma cells from an epithelial to a mesenchymal phenotype (epithelial-mesenchymal transition, or EMT) is of central interest to clinicians due to its putative role in promoting tumor invasiveness and acquisition of chemoresistance. The hallmarks of EMT include the loss of cell-cell adhesion (due to the loss of E-cadherin expression), cytoskeletal reorganization to replace keratin with vimentin intermediate filaments, enhanced cellular motility, and resistance to apoptosis and senescence [66], [75], [95]. Because downregulation of E-cadherin protein levels at the plasma membrane, along with upregulation of vimentin protein levels have been documented in a number of tissues during EMT [16], [33], [112], there has been development of a quantitative immunofluorescence imaging method relying on E-cadherin and vimentin expression and clinically validated tumor markers, in tissue sections on microscope slides, as unequivocal markers of epithelial and mesenchymal phenotypes, respectively which are distinguished from surrounding stroma [92], [93]. Because histological staining does not distinguish tumor cells displaying a mesenchymal phenotype from neighboring non-neoplastic mesenchymal cells (stroma) in the tumor microenvironment [8], and is further limited due to its' reliance on a small, 2 dimensional preparation from a much larger tissue, containing both normal and tumor cells, a novel method was developed to identify epithelial, mesenchymal and transitional phenotypes (EMT) in neoplastic cells from tumor biopsies, and their localization relative to each other, in regions of 2 dimensional biopsy images by Stephen Lockett. Measuring EMT in patient samples as part of a treatment strategy could offer valuable insight for medical decision-making including drug selection and sequencing in a treatment regimen.

The large quantities of 2D images require automatic analysis and with recent development in computer vision and image processing, algorithms that detect regions of interest have been designed for biological images. For example, an entropy-based automated technique has been proposed to detect regions of interest in prostate biopsy images [19]. The method utilizes the fact that due to the presence of different types of textures and shapes of objects, entropy is higher in regions of cancer cells. In [39], a morphometric tool for segmentation of blood and lymphatic vessels was reported for tumor prognosis. There are examples using multiscale analysis to segment region of interest (ROI) of 2D confocal microscopy images [56]. However, detection of ROI remains a topic touched by few compared with whole image classification or abnormality detection for cancer prognosis.

The shearlet transform is among the popular tools for extracting directional and multiscale features in biomedical image analysis. It captures edge-like structures by utilizing shearing in its construction, and its multiscale framework allows for detection of details in both high and low resolutions. Its capacity for detecting the approximate tangent direction of discontinuity can yield local curvature estimates [71]. It has been used for faciliating Gleason grading of prostate cancer in histological images [99], [100]. Shearlet transform was applied by extracting features at different stages of prostate cancer and these features served as input for a classification task.

Shearlet transform has been used as whole image feature extraction tool [2] or directional filter in preprocessing steps [64] in biomedical image analysis. In this study we combine properties of two types of nuclei with properties of shearlet coefficients and design a ROI detection method to distinguish mesenchymal cells from epithelial cells in microscopy images. The chapter is organized as follows. In Section 6.2, we introduce the shearlet transform and our method, max difference thresholding algorithm. In Section 6.3, we compare our method with benchmark methods and demonstrate results and we draw the conclusion in Section 6.4.

6.2 Shearlet based region detection

The study of theory and applications of directional representation has been an important subject in harmonic analysis. Although wavelets are known for decomposing functions in one dimension, they achieve sub-optimal results due to the presence of discontinuities such as curves in higher dimensions. In order to capture these edge-like structures, types of representation other than wavelets are developed. Examples include contourlets, curvelets, ridgelets, bandelets, wedgelets, and shearlets. Shearlets are popular among the representations with several fast implementations available. S. Häuser and G. Steidl have provided the tutorial for their first finite and translation invariant shearlet implemtntation in [53] in 2014. Other implementations include the local shearlet toolbox [37] and ShearLab [73], [74]. Shearlets have been widely used in applications including road detection in LIDAR images [115], superressolution of optical and hyperspectral data [31], [91], [115], and image registration [91]. With its fast implementation and its ability to capture anisotropic structures, we choose to use shearlet to detect differences in cell nuclei shapes. We will introduce shearlet transform next along with the Shearlet Max Difference Thresholding method for segmenting regions of different cells.

6.2.1 Shearlet Transform

We introduce the mathematics behind the shearlet transform in this Section based on the description in [53]. A shearlet $\psi_{a,s,t}$ in 2D is defined by dilation, shear and translation of a function $\psi \in L_2(\mathbb{R}^2)$:

$$\psi_{a,s,t}(x) = a^{-\frac{3}{4}}\psi(A_a^{-1}S_s^{-1}(x-t)) = a^{-\frac{3}{4}}\psi\begin{bmatrix} \frac{1}{a} & -\frac{s}{a} \\ 0 & \frac{1}{\sqrt{a}} \end{bmatrix}(x-t)). \tag{6.1}$$

Here $A_a = \begin{bmatrix} a & 0 \\ 0 & \sqrt{a} \end{bmatrix}$, $a \in \mathbb{R}^+$ is the dilation matrix and $S = \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix}$, $s \in \mathbb{R}$ is the shear matrix. The function ψ is defined via Fourier transform:

$$\hat{\psi}(\omega_1, \omega_2) = \hat{\psi}_1(\omega_1)\hat{\psi}_2(\frac{\omega_2}{\omega_1}), \tag{6.2}$$

and functions ψ_1 and ψ_2 are defined by

$$\hat{\psi}_1(\omega) = \sqrt{b^2(2\omega) + b^2(\omega)}, \quad \hat{\psi}_2(\omega) = \begin{cases} \sqrt{v(1+\omega)}, & \text{for} \quad \omega \leq 0, \\ \sqrt{v(1-\omega)}, & \text{for} \quad \omega > 0, \end{cases} \tag{6.3}$$

where b and v are two auxiliary functions defined by

$$b(\omega) = \begin{cases} \sin(\frac{\pi}{2} v(|\omega| - 1)), & \text{for } 1 \leq |\omega| \leq 2, \\ \cos(\frac{\pi}{2} v(\frac{1}{2}|\omega| - 1)), & \text{for } 2 \leq |\omega| \leq 4, \\ 0, & \text{otherwise,} \end{cases} \tag{6.4}$$

and

$$v(x) = \begin{cases} 0, & \text{for } x < 0, \\ 35x^4 - 84x^5 + 70x^6 - 20x^7, & \text{for } 0 \leq x \leq 1, \\ 1, & \text{for } x > 1. \end{cases} \tag{6.5}$$

The continuous shearlet transform $\mathcal{SH}_\psi(f)$ of $f \in L^2(\mathbb{R})$ at (a, s, t) is given as the inner product of the function f with the shearlet $\psi_{a,s,t}$:

$$\mathcal{SH}_\psi(f)(a, s, t) =< f, \psi_{a,s,t} > . \tag{6.6}$$

We treat an image as a discrete function $f \in \mathbb{R}^{M \times N}$ sampled on $\{(\frac{x_1}{M}, \frac{x_2}{N}) : x = (x_1, x_2) \in X\}$ where $X = \{(x_1, x_2) : x_1 = 0, ..., M-1, x_2 = 0, ..., N-1\}$. For a discrete shearlet transform on f, we let $j_0 = \lfloor \frac{1}{2} \log_2 \max\{M, N\} \rfloor$ be the number of considered scales. We discretize the dilation, shear and translation parameters as

$$\begin{aligned} & a_j = 2^{-2j}, \qquad j = 0, ..., j_0 - 1, \\ & s_{j,k} = k2^{-j}, \quad -2^j \leq k \leq 2^j, \\ & t_x = (\tfrac{x_1}{M}, \tfrac{x_2}{N}), \quad x \in \mathcal{G}. \end{aligned} \tag{6.7}$$

Let $\mathcal{S}_j = \{St_{j,1}(x), St_{j,2}(x), ..., St_{j,K}(x)\}$ denote the set of shearlet coefficients obtained at scale level j for pixel x. Here $St_{j,k}(x)$ is the kth computed shearlet coefficients at pixel x. We obtain $K = 2^{j+2}$ coefficients for scale level j.

Shearlets are designed as multiscale directional representations to address singularities of the images. The discrete shearlet transform computes shearlet coefficients at different scales. The first J largest shearlet coefficients can be used to reconstruct the image with error bound on the order of $J^{-2}(\log J)^3$. This means that shearlet representation captures unique features of the original space, and thus we utilize properties of such representation to facilitate our task of analyzing nuclei with directional information.

6.2.2 Shearlet Max Difference Thresholding Method

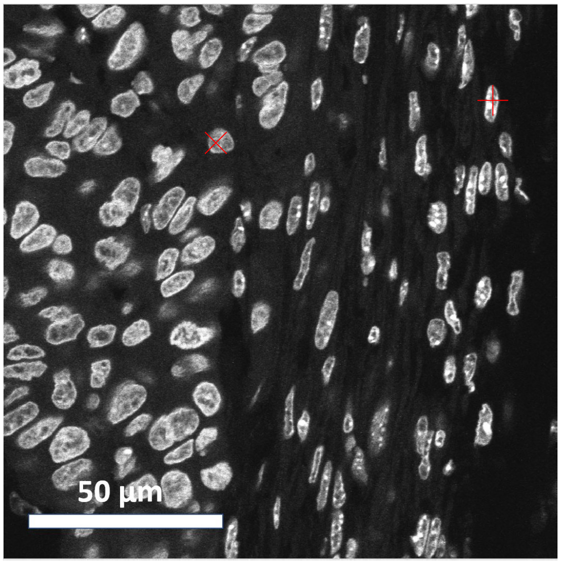

Figure 6.1: Image of cell nuclei

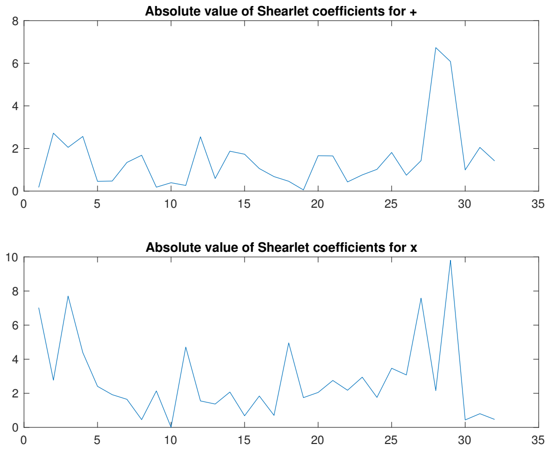

Figure 6.2: Shearlet coefficients at "+" and "x" respectively

There are two major properties of the two types of nuclei in tumor biopsy images that motivates the Shearlet Max Difference Thresholding method: difference in shape and difference in density. We discuss a path to quantify the differences, and introduce the Shearlet Max Difference Thresholding method. We work on grayscale images with the value at each pixel representing intensity.

At a given scale, the set of shearlet coefficients is produced by the inner product of the image at location $x = (x_1, x_2)$ and the dilated and sheared shearlet at certain directions. Despite their ability to locate edges, which are present in both round and elongated objects, shearlet coefficients inside these two objects demonstrate different distributional characteristics. In elongated objects, computed shearlet coefficients at given location x tend to contain one high value, with the remaining being evenly distributed and small. In round objects, all the shearlet coefficients tend to differ less and have small variation. In Figure 6.1, examples of an epithelial and a mesenchymal nucleus are labeled by × and + respectively (confirmed by the presence and absense of e-cadherin antibody staining at the cell membrane respectively, not shown), and the absolute value of their shearlet coefficients at level 5 (pixel + on right, pixel x on left) are presented in Figure 6.2. We can observe that for the elongated nucleus + there is one peak that differs significantly from the remaining coefficients, while for the round nucleus × there are multiple peaks with large variance and no such behavior.

Recall that the set $\mathcal{S}_j = \{St_{j,1}(x), St_{j,2}(x), ..., St_{j,K}(x)\}$ denotes the set of

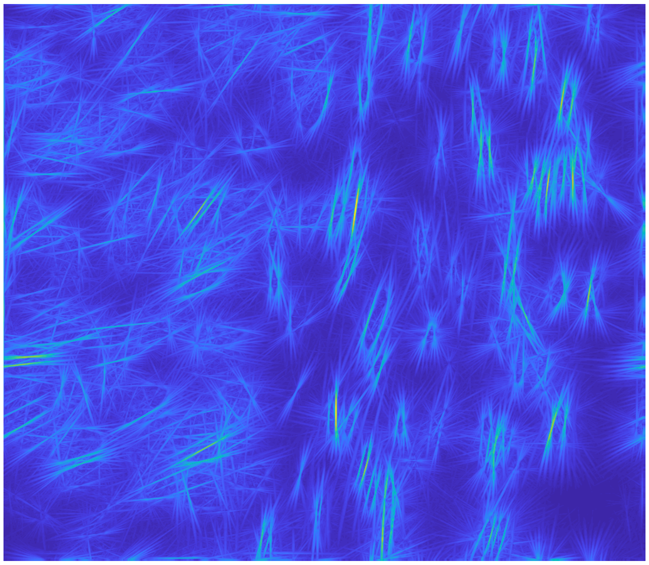

Figure 6.3: Max Difference coefficients MD of Figure 6.1

shearlet coefficients of input image f obtained for x at scale j. Let $St_j(x)^{max} = \max_{i=1,...,K} St_{j,i}$ denote the maximum shearlet coefficient at scale j. Let $St_j(x)^{avg}$ denote the mean of the remaining coefficients at scale j. We calculate the max difference

(MD) coefficients of the shearlet coefficients for x by

$$MD(x) = St_j(x)^{max} - St_j(x)^{avg}. \tag{6.8}$$

Figure 6.3 demonstrates the MD representation of Figure 6.1 at scale level $j = 5$.

We introduce the weight matrix obtained from density of intensities in the image, because it is observed that elongated nuclei clusters are less dense compared with clusters of rounded nuclei, and signals from elongated nuclei are generally not as strong as signals from rounded nuclei. In order to take into account the difference in strength of signal, we first blur the image by taking the average of each pixel around its neighborhood, and then map the density image to a kernel that enhances regions with low density and weakens regions with high density.

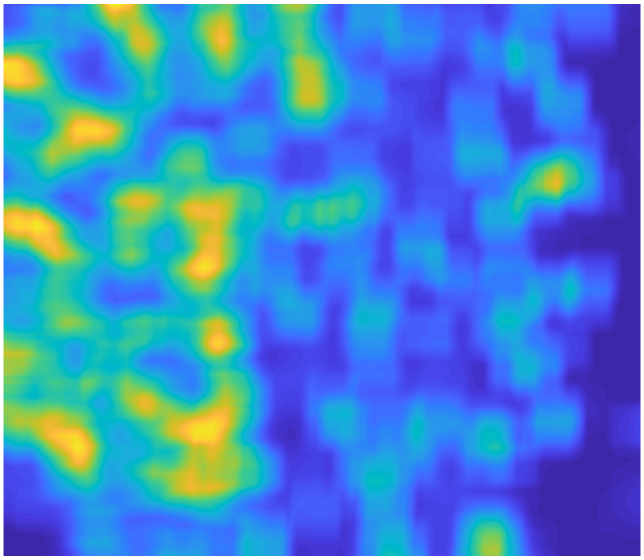

Figure 6.4: Density matrix D of Figure 6.1 (with uniform filter)

Now we calculate $D(x)$, the input image blurred with a uniform averaging filter of length s. Recall that f represents the input image. Each entry of the density matrix D is defined by

$$D(x) = \frac{1}{s^2} \sum_{y \in S_x} f(y). \tag{6.9}$$

Note that $S_x = \{s = (s_1, s_2) | s_1 \in [x_1 - \frac{s}{2}, x_1 + \frac{s}{2}], s_2 \in [x_2 - \frac{s}{2}, x_2 + \frac{s}{2}]\}$ represents the support of the filter at each x. Figure 6.4 illustrates the density matrix of Figure 6.1 blurred with a uniform filter of side length $s = 20$.

We take the negative exponential of the density matrix. The weight matrix W enhances low density regions and compromises high density regions. W is defined by

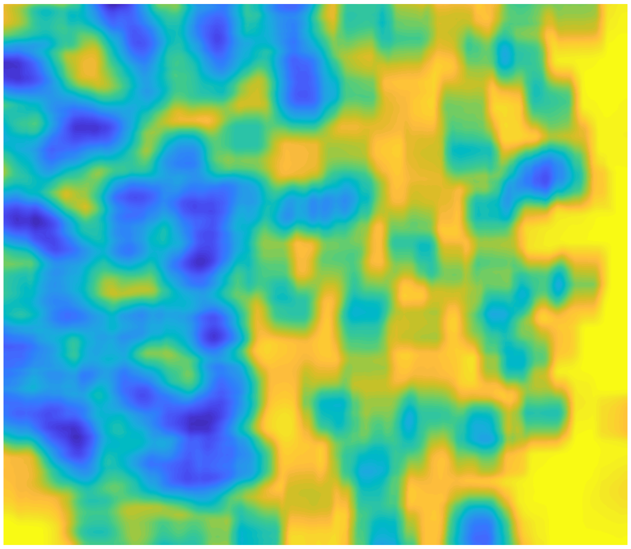

Figure 6.5: Weight matrix W obtained from density matrix in Figure 6.4

$$W(x) = \exp - \frac{|D(x)|}{|D|_{max}}, \tag{6.10}$$

where $|D|_{max}$ denotes the maximum absolute value of the density matrix D. Figure 6.5 represents the weight matrix of Figure 6.1.

Next, we threshold the MD matrix of input image f by certain threshold

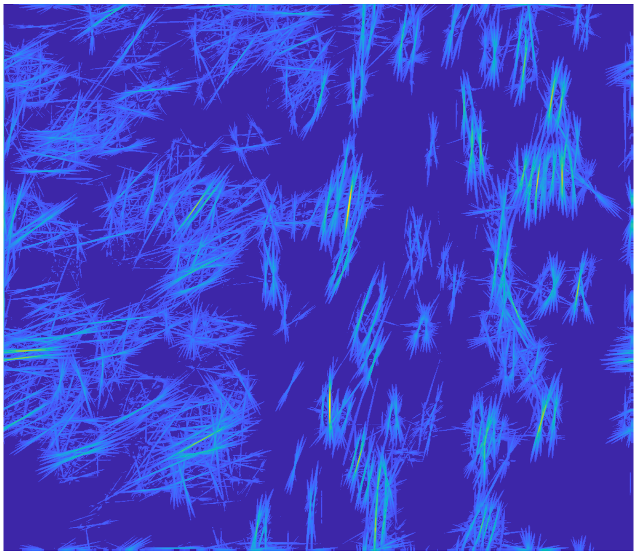

Figure 6.6: Thresholded MD matrix T

θ. The choice of θ is dependent on data set and is best to be chosen as the mean MD^{avg} value of MD of a sample image in which the number of elongated nuclei and the number of round nuclei are approximately equal. By using such threshold θ, we discard signals from regions with no clear directional information. Denote the output of the thresholding by T. We have

$$T(x) = MD(x) \cdot \mathbb{1}_{MD(x) \geq MD^{avg}}, \tag{6.11}$$

where $\mathbb{1}$ is the indicator function. Note that the chosen threshold distinguishes center lines of elongated nuclei from round nuclei in the MD representation of the image, but there are regions in which multiple directions are present, and the signals from the elongated region are not necessarily strong enough. Figure 6.6 represents the thresholded image of Figure 6.3.

We multiply weights and the thresholded image T to enhance signals from

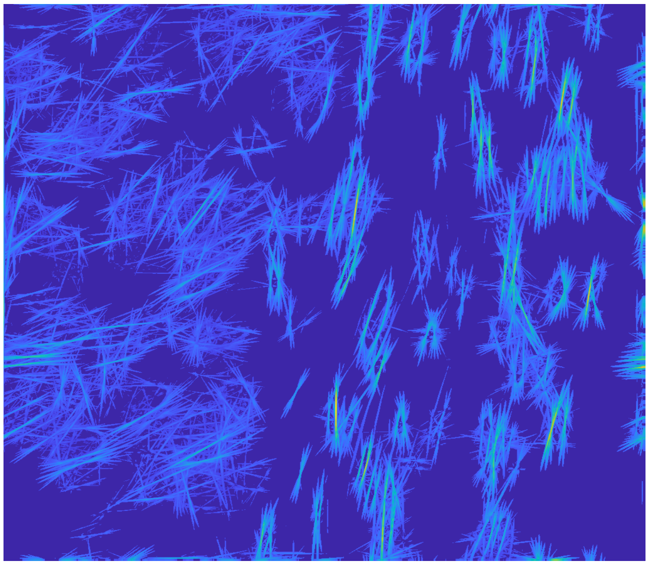

Figure 6.7: Weighted matrix T_w of T

regions of elongated nuclei. Define the weighted output T_w by

$$T_w(x) = W(x) \cdot T(x). \tag{6.12}$$

Figure 6.7 demonstrates the T_w matrix of Figure 6.1.

Due to the fact that adjacent elongated nuclei tend to orient in similar directions, edge like features appear in elongated nuclei regions in MD matrix while no ordered structures appear in round nuclei regions, as illustrated in Figure 6.7. Therefore, with enhancement from weight matrix W, we apply the shearlet transform for a second time to T_w, and obtain the max difference matrix MD_T on the new set of shearlet coefficients we get from T_w. In particular, let $\tilde{\mathcal{S}}_j = \{\tilde{S}t_{j,1}, ..., \tilde{S}t_{j,K}\}$ be the set of shearlet coefficients of T_w at scale j. MD_T is defined by

$$MD_T = \tilde{S}t_j(x)^{max} - \tilde{S}t_j(x)^{avg}, \tag{6.13}$$

where $\tilde{S}t_j(x)^{max}$ is the maximum in $\tilde{\mathcal{S}}_j$ and $\tilde{S}t_j(x)^{avg}$ is the average of the remaining of the coefficients in $\tilde{\mathcal{S}}_j$. Figure 6.8 illustrates MD_T, the MD matrix of T_w. We apply the same uniform filter of length size s to MD_T to obtain D_{MD_T}. D_{MD_T} is defined by

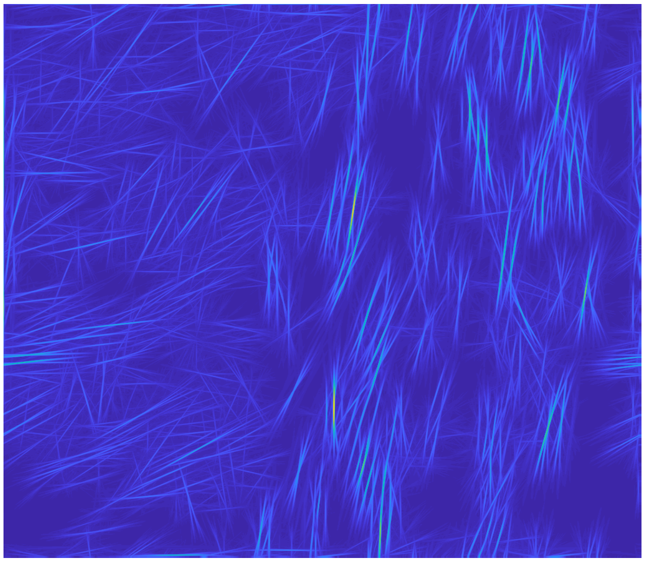

Figure 6.8: Max Difference coefficients MD_T of T_w in Figure 6.7

$$D_{MD_T}(x) = \frac{1}{s^2} \sum_{y \in S_x} MD_T(y), \tag{6.14}$$

where S_x is defined previously in Equation 6.9. We enhance the elongated nuclei regions again by multiplying weights from the original density image with the blurred second max difference matrix D_{MD_T}. We obtain the weighed density before thresholding $WD = W(x) \cdot D_{MD_T}(x)$. Figure 6.9 demonstrates the density matrix D_{MD_T} of MD_T and Figure 6.10 demonstrates the weighted matrix WD of MD_T.

Last, we apply thresholding by WD^{avg}, which is the average of WD determined by the same sample image as in Equation 6.11, and obtain $\tilde{S} = WD_T(x)$

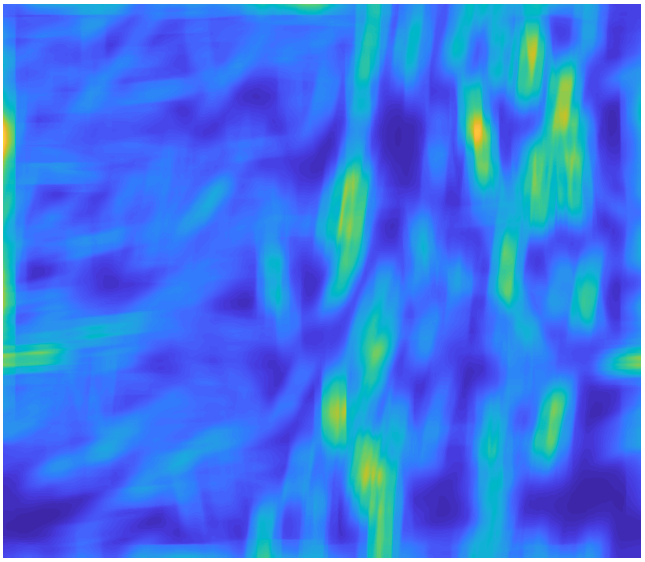

Figure 6.9: Density matrix D_{MD_T} of MD_T (with uniform filter)

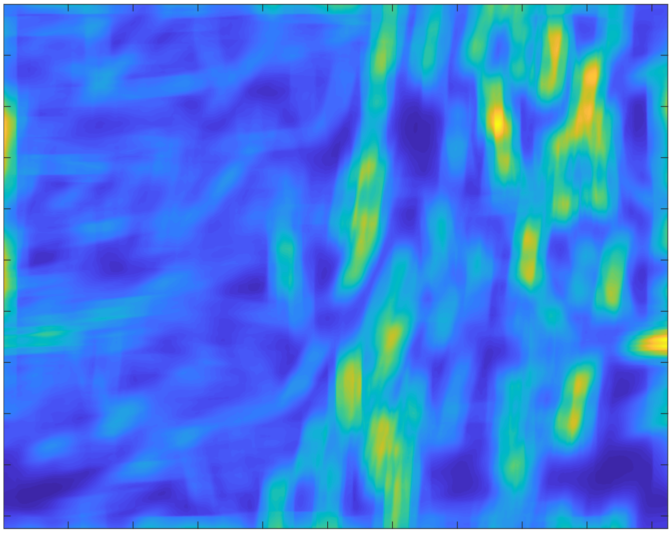

Figure 6.10: Weighted density matrix WD of D_{MD_T}

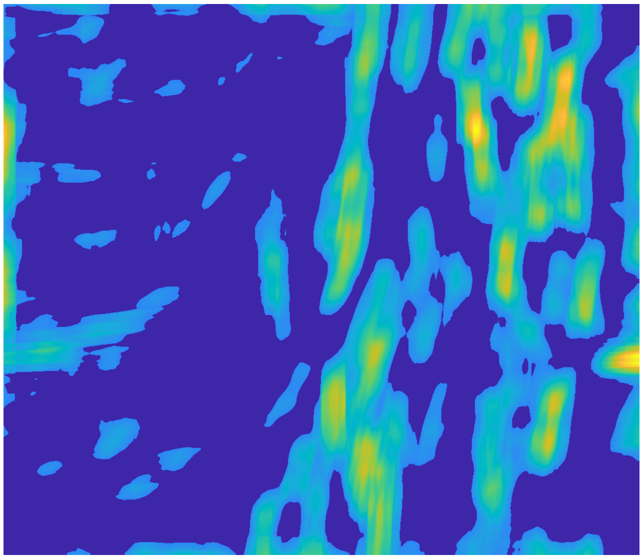

Figure 6.11: $\tilde{S}$, the thresholded WD from Shearlet coefficients at level 5

$\cdot \mathbb{1}_{WD(x) \geq WD^{avg}}$ as our final result. For post-processing, we eliminate regions that are small and considered as noise.

Although taken at high resolution, images differ in sizes and so do the relative sizes of the nuclei. For smaller images with large nuclei, to make sure that we avoid over-segmentation while not losing information from finer scales, we tailored our method to include two levels of shearlet coefficients, lower and higher level. Details of the algorithm are described in Section 6.3.1. In this way, over-segmentation from high level coefficients is covered by outcome from low level coefficients, and missing components of the outcome from low level coefficients are made up by outcome from high level coefficients. Based on the sizes of images and relative sizes of the cell nuclei, we choose to use the shearlet coefficients at the top two finest levels, scales 4 and 5. Figure 6.11 represents the output of the algorithm for the set of shearlet coefficients at level 5 before post processing. Figure 6.12 represents the

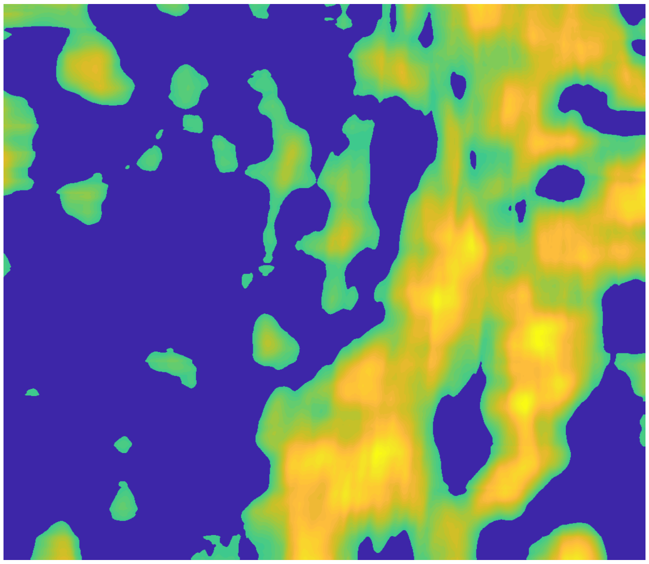

Figure 6.12: $\tilde{S}$, the thresholded WD from Shearlet coefficients at level 4

outcome of the same algorithm for the set of shearlet coefficients at level 4 before post processing. Figure 6.13 gives the area selected as regions of elongated nuclei from both levels of shearlet coefficients, and 6.14 represents the regions of elongated nuclei after post processing.

6.3 Experiment and Result

In this section we introduce the background on the data set of tumor biopsy we obtained, and describe the algorithm for the Shearlet Max Difference Thresholding method. We also discuss parameter selection, and give description of two benchmark algorithms as comparison method. The experimental results and examples of result images are also included.

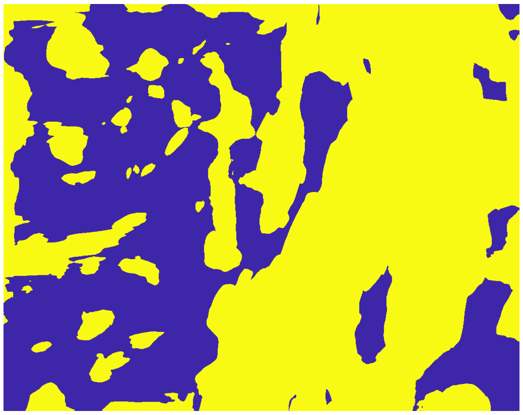

Figure 6.13: Combined regions of elongated nuclei (yellow) from shearlet coefficients at level 4 and level 5 before post processing

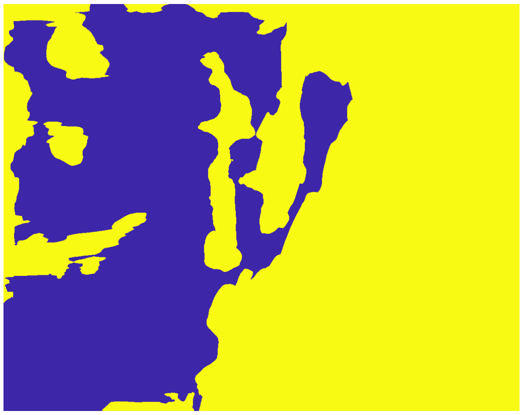

Figure 6.14: Combined regions of elongated nuclei (yellow) from Shearlet coefficients at level 4 and level 5 after post processing

6.3.1 The Algorithm

We summarize the steps described in Section 6.2 at one level as follows.

The threshold θ is determined by the mean MD^{avg} of MD of a sample image in which the number of elongated nuclei and the number of round nuclei are approximately equal. We use the same sample image for the threshold WD^{avg}. We choose $j = 5$ for the images in which the nuclei are relatively small.

```
Algorithm 1: Shearlet Max Difference Thresholding Algorithm
  Input: Input image f
  Output: Segmentation result $\tilde{S}$
  Apply Fast Finite Shearlet Transform (FFST) on input image $f$.
  Compute $MD$ of $f$ at level $j$.
  Compute density matrix $D$ of $f$, and compute weight matrix $W$.
  Threshold $MD$ by threshold $\theta$ to get $T$.
  Multiply $T$ with $W$ element-wise to get $T_w$ as their product.
  Apply FFST on $T_w$.
  Compute $MD_T$, the max difference of $T_w$, at level $j$.
  Compute density of of $MD_T$ to obtain $D_{MD_T}$.
  Multiply $D_{MD_T}$ with $W$ to obtain $WD$.
  Threshold $D_{MD_T}$ by $WD^{avg}$ to get $\tilde{S}$.
```

For high resolution images, or the images in which the nuclei are relatively large, we incorporate two levels of shearlet coefficients. The algorithm for high resolution images combines the outcome of algorithm 1 at two levels, $j = 4$ and $j = 5$. In the final output stage, we apply logical OR operation on two regions of interests calculated from two sets of shearlet coefficients to include regions selected as elongated nuclei in both processes.

For post-processing, we first erase regions that are selected as elongated nuclei that has low total intensity. We repeat this procedure for regions that are selected as round nuclei. In this way, we reduce the effect of over-segmentation. Last, we erase regions with low average intensity in order to remove regions with no nuclei.

6.3.2 The Data Set

The images of tumor biopsies that we obtained are of the size 1024 by 1024 or larger. We tested our algorithm on 7 images with over 2000 nuclei in total.

The issue cells in the image were the human gastric cancer cell line, MKN45, grown as a xenograft under the flank of an immunocompromised mouse. Briefly, 5 micron thick tissue sections were cut, labeled with an EMT panel of fluorescence antibodies and the DNA dye, DAPI and imaged by confocal microscopy with a 63X oil objective lens. Only DAPI images were used in this study, since the eventual goal is to classify 3D regions acquired deep inside thick tissue where antibodies cannot easily reach. This data was provided by National Cancer Institute through its collaboration with University of Maryland, College Park.

6.3.3 Validation Methods

We implemented two benchmark algorithms as comparisons to our method. We chose wavelet, a standard tool for image processing, and one level shearlet coefficients as two alternative feature extraction tools and used 1-nearest neighbor (1NN) [89] as a classification method with a universal training data set containing one example per class across images. 1NN compute the distance between the input data points and the training points and classify each input data point into the same class as its closest training point.

For both validation methods, we divide the image into a number of superpixels [1]. Superpixel is a way of partitioning images based on similarity between pixels in proximity. By treating each group of pixels that are close in location and that are similar in some sense as one superpixel, the computation and representation efficiency is improved. We eliminate the superpixels with no positive intensity values above certain threshold, which correspond to non-nuclear areas of the images. Because of noise and the fact that intensities vary within each individual nucleus, superpixels happen to be a better choice than watershed algorithm in segmenting nuclei in the image. We compute wavelet transform and shearlet transform on the input image and obtain representative coefficients of each superpixel as the input for classification. The 1NN classification is performed at superpixel level.

The fast wavelet transform (FWT) developed by S. Mallat in 1989 [83] is performed on the image at level $j = 4$, and average wavelet coefficients within each superpixel are calculated as a feature vector representing the superpixel.

For shearlet benchmark, the shearlet transform is performed on input image. We pick the pixel in each superpixel such that its shearlet coefficients have the largest max difference as the representative. Each representative shearlet coefficient is sorted in descending order for uniform comparison and to capture sensitivity in directionality.

For training data set for the 1NN classifier, one superpixel of the nucleus of each type is selected and its wavelet and shearlet coefficients served as training data set. The choice of the number of training examples is based on the fact that there are only 4 mesenchymal nuclei in one of the images we have, hence by the common standard of 30% training data, we can only choose one nucleus of each type. We want to keep training data size consistent across images.

6.3.4 Results

We calculate the percentages of the correctly classified nuclei as epithelial and the percentages of mesenchymal nuclei correctly classified as mesenchymal of the three methods. Results of the quantitative analysis of three methods is shown in Table 6.1. # E nuclei represents the number of epithelial nuclei in the image and # M nuclei represents the number of mesenchymal nuclei. Figures 6.15 - 6.19 demonstrate the results obtained from three methods. Note that yellow indicates mesenchymal regions (regions of elongated nuclei) and that purple indicates epithelial regions (regions of round nuclei).

Our Shearlet Max Difference Thresholding method outperforms the traditional

Images	Ground-truth Enumeration		Wavelet (Correct)		Shearlet (Correct)		SMDT (Correct)	
	# M nuclei	# E nuclei	M %	E %	M %	E %	M %	E %
1	204	1450	82	84	83	**93**	**99**	92
2	55	308	80	**77**	81	71	**83**	62
3	60	58	100	91	65	90	**100**	**93**
4	7	63	71	**89**	57	76	**100**	81
5	51	75	45	80	22	**84**	**99**	73
6	73	78	6	**91**	22	**84**	**100**	85
7	4	120	0	82	**75**	65	**75**	**97**
Total	454	2152	67	83	66	**87**	**97**	**87**

Table 6.1: Result of regions of interest detection of wavelet, shearlet and SMDT methods

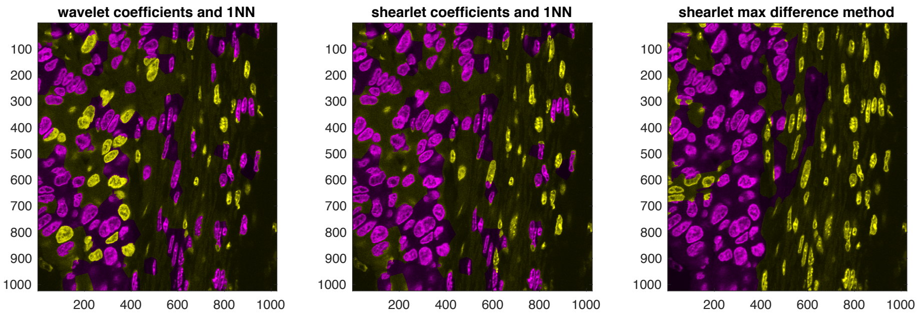
wavelet coefficients and 1NN
shearlet coefficients and 1NN
shearlet max difference method
100
200
300
400
500
600
700
800
900
1000
200
400
600
800
1000

Figure 6.15: Segmentation results of selected images 1

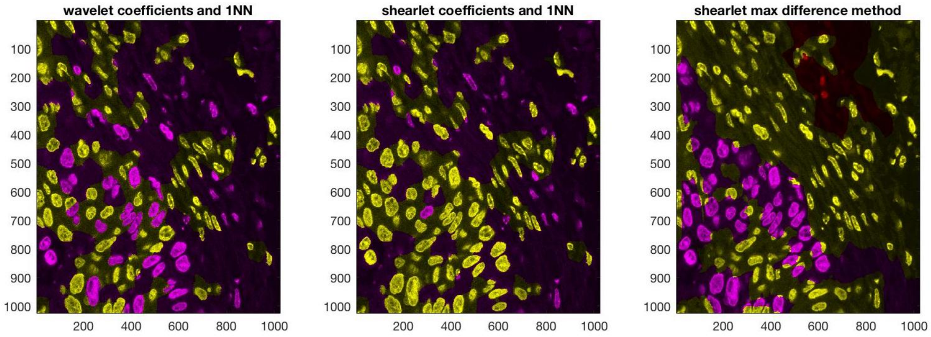
wavelet coefficients and 1NN
shearlet coefficients and 1NN
shearlet max difference method
100
200
300
400
500
600
700
800
900
1000
200
400
600
800
1000

Figure 6.16: Segmentation results of selected images 2

wavelet method and the standard shearlet method in detecting elongated nuclei in most cases. Note that the lower success rate in classifying some of the mesenchymal cells can be due to the fact that some transitioning epithelial cells are scattered in between mesenchymal cells and the ground truth boundaries often did not include these individual nuclei. But the overall results indicates that the Shearlet Max Difference Thresholding method outperforms the wavelet method and standard shearlet method.

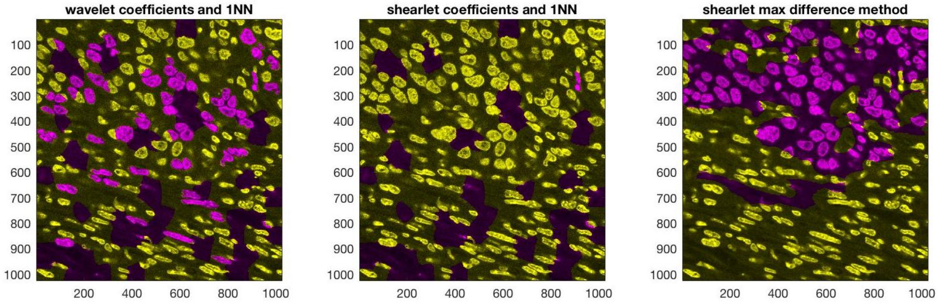
wavelet coefficients and 1NN
shearlet coefficients and 1NN
shearlet max difference method
100
200
300
400
500
600
700
800
900
1000
200
400
600
800
1000

Figure 6.17: Segmentation results of selected images 3

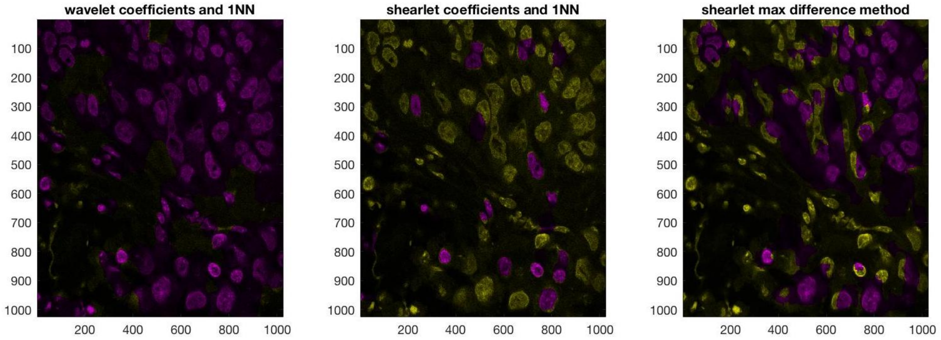
wavelet coefficients and 1NN
shearlet coefficients and 1NN
shearlet max difference method
100
200
300
400
500
600
700
800
900
1000
200
400
600
800
1000

Figure 6.18: Segmentation results of selected images 4

6.4 Conclusion

The epithelial-mesenchymal transition (EMT) is an important subject in the study of cancer invasion. Taking the advantage of the directional and locational information of nuclei, we designed the shearlet max difference algorithm to detect the regions of epithelial versus mesenchymal nuclei in images capturing this process. Comparing with standard image processing tools, wavelet and shearlet, our method demonstrates improvement by 30% in its ability to correctly detect regions of interest.

By its mathematical nature, the shearlet max difference can be extended to

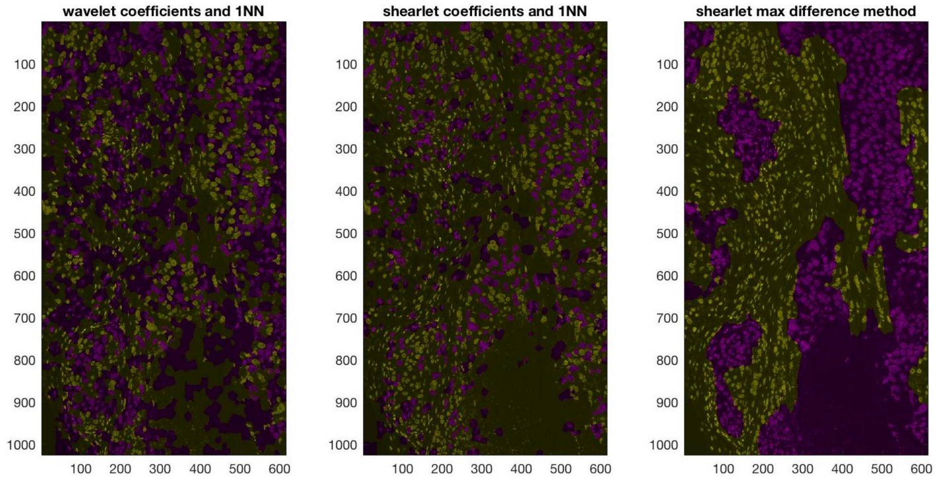
wavelet coefficients and 1NN
shearlet coefficients and 1NN
shearlet max difference method
100
200
300
400
500
600
700
800
900
1000
100
200
300
400
500
600

Figure 6.19: Segmentation results of selected images 5

detect volumes of interest 3D images. Continuous shearlet transformation has been generalized anisotropically and isotropically for $L^2(\mathbb{R}^k)$ functions, which includes functions in all high dimensional spaces. Its mathemtacial properties are studied in [27], [28], [70], and it has been applied in problems such as superresolution for remotely sensed images [30]. 3D discrete shearlet transform has also been applied in video processing [94]. In separate studies utilizing chemical clearing of tissue, we can image 100s of microns into DAPI-labeled tissue without observable loss of image quality and therefore the properties of epithelial cells and mesenchymal cells are also observed in 3D. Hence it is possible to apply 3D discrete shearlet transform to obtain coefficients representing strength of signals across 3D directions and use the 3D shearlet max difference method to categorize these nuclei. Note that an elongated nucleus in 3D might look round in 2D image, and thus by looking at 3D data, we can obtain results with better accuracy. This will be the next step in this study.

Bibliography

[1] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012) Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, **34**, 2274–2282.

[2] Aneja, R. and Siddiqi, A. (2016) Hybrid image compression using shearlet coefficients and region of interest detection. *Journal of Medical Imaging and Health Informatics*, **6**, 506–517.

[3] Balan, R., Bodmann, B. G., Casazza, P. G., and Edidin, D. (2009) Painless reconstruction from magnitudes of frame coefficients. *Journal of Fourier Analysis and Applications*, **15**, 488–501.

[4] Balan, R., Casazza, P., and Edidin, D. (2006) On signal reconstruction without phase. *Applied and Computational Harmonic Analysis*, **20**, 345–356.

[5] Balan, R., Singh, M., and Zou, D. (2017) Lipschitz properties for deep convolutional networks. *arXiv preprint arXiv:1701.05217*.

[6] Barron, A. R. (1993) Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information theory*, **39**, 930–945.

[7] Bartók, A. P., Kondor, R., and Csányi, G. (2013) On representing chemical environments. *Physical Review B*, **87**, 184115.

[8] Bastid, J. (2012) Emt in carcinoma progression and dissemination: facts, unanswered questions, and clinical considerations. *Cancer and Metastasis Reviews*, **31**, 277–283.

[9] Benedetto, J. J. (1987) Gabor representations and wavelets. Tech. rep., Norbert Wiener Center, University of Maryland, College Park.

[10] Benedetto, J. J. (1996) *Harmonic analysis and applications*, vol. 23. CRC Press.

[11] Benedetto, J. J., Czaja, W., Gadziński, P., and Powell, A. M. (2003) The Balian-Low theorem and regularity of Gabor systems. *The Journal of Geometric Analysis*, **13**, 239.

[12] Benedetto, J. J., Heil, C., and Walnut, D. F. (1998) Gabor systems and the Balian-Low theorem. *Gabor analysis and algorithms*, pp. 85–122, Springer.

[13] Benedetto, J. J. and Walnut, D. F. (1994) Gabor frames for L2 and related spaces. *Wavelets: mathematics and applications*, pp. 97–162.

[14] Blumensath, T. and Davies, M. E. (2007) On the difference between orthogonal matching pursuit and orthogonal least squares.

[15] Bölcskei, H., Grohs, P., Kutyniok, G., and Petersen, P. (2017) Optimal approximation with sparsely connected deep neural networks. *arXiv preprint arXiv:1705.01714*.

[16] Brabletz, T. (2012) To differentiate or not—routes towards metastasis. *Nature Reviews Cancer*, **12**, 425.

[17] Briggs, W. L. et al. (1995) *The DFT: an owners' manual for the discrete Fourier transform*. Siam.

[18] Bruna, J., Chintala, S., LeCun, Y., Piantino, S., Szlam, A., and Tygert, M. (2015) A theoretical argument for complex-valued convolutional networks. *CoRR abs/1503.03438*.

[19] Bueno, G., Fernández-Carrobles, M.-M., Déniz, O., Salido, J., Vállez, N., and García-Rojo, M. (2013) An entropy-based automated approach to prostate biopsy roi segmentation. *Diagnostic pathology*, vol. 8, p. S24, BioMed Central.

[20] Chen, S., Cowan, C. F., and Grant, P. M. (1991) Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on neural networks*, **2**, 302–309.

[21] Christensen, O., Kim, H. O., and Kim, R. Y. (2013) Regularity of dual Gabor windows. *Abstract and Applied Analysis*, vol. 2013, Hindawi.

[22] Chui, C. K. and Mhaskar, H. N. (2016) Deep nets for local manifold learning. *arXiv preprint arXiv:1607.07110*.

[23] Ciregan, D., Meier, U., and Schmidhuber, J. (2012) Multi-column deep neural networks for image classification. *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, pp. 3642–3649, IEEE.

[24] Ciresan, D. C., Meier, U., Masci, J., Maria Gambardella, L., and Schmidhuber, J. (2011) Flexible, high performance convolutional neural networks for image classification. *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, p. 1237, Barcelona, Spain.

[25] Coifman, R. and Fefferman, C. (1974) Weighted norm inequalities for maximal functions and singular integrals. *Studia Mathematica*, **51**, 241–250.

[26] Cybenko, G. (1989) Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, **2**, 303–314.

[27] Czaja, W. and King, E. J. (2012) Isotropic shearlet analogs for$l_2(r^k)$ and localization operators. *Numerical functional analysis and optimization*, **33**, 872–905.

[28] Czaja, W. and King, E. J. (2014) Anisotropic shearlet transforms for $l^2(r^k)$. *Mathematische Nachrichten*, **287**, 903–916.

[29] Czaja, W., Manning, B., Murphy, J. M., and Stubbs, K. (2016) Discrete directional Gabor frames. *Applied and Computational Harmonic Analysis*.

[30] Czaja, W., Murphy, J. M., and Weinberg, D. (2015) Superresolution of remotely sensed images with anisotropic features. *Sampling Theory and Applications (SampTA), 2015 International Conference on*, pp. 317–321, IEEE.

[31] Czaja, W., Murphy, J. M., and Weinberg, D. (2016) Single-image superresolution through directional representations. *arXiv preprint arXiv:1602.08575*.

[32] De Bruijn, N. (1967) Uncertainty principles in fourier analysis. *Inequalities*, **2**, 57–71.

[33] De Craene, B. and Berx, G. (2013) Regulatory networks defining emt during cancer initiation and progression. *Nature Reviews Cancer*, **13**, 97.

[34] Deglmann, P., Schäfer, A., and Lennartz, C. (2015) Application of quantum calculations in the chemical industry—an overview. *International Journal of Quantum Chemistry*, **115**, 107–136.

[35] Donoho, D. L., Elad, M., and Temlyakov, V. N. (2006) Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on information theory*, **52**, 6–18.

[36] Duffin, R. J. and Schaeffer, A. C. (1952) A class of nonharmonic fourier series. *Transactions of the American Mathematical Society*, **72**, 341–366.

[37] Easley, G., Labate, D., and Lim, W.-Q. (2008) Sparse directional image representations using the discrete shearlet transform. *Applied and Computational Harmonic Analysis*, **25**, 25–46.

[38] Fei-Fei, L., Fergus, R., and Perona, P. (2007) Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer vision and Image understanding*, **106**, 59–70.

[39] Fernández-Carrobles, M.-M., Tadeo, I., Noguera, R., García-Rojo, M., Déniz, O., Salido, J., and Bueno, G. (2013) A morphometric tool applied to angiogenesis research based on vessel segmentation. *Diagnostic pathology*, vol. 8, p. S20, BioMed Central.

[40] Forti, M. and Tesi, A. (1995) New conditions for global stability of neural networks with application to linear and quadratic programming problems. *IEEE Transactions on Circuits and Systems I: Fundamental theory and applications*, **42**, 354–366.

[41] Fukushima, K. (1975) Cognitron: A self-organizing multilayered neural network. *Biological cybernetics*, **20**, 121–136.

[42] Fukushima, K. (1979) Neural network model for a mechanism of pattern recognition unaffected by shift in position-neocognitron. *IEICE Technical Report, A*, **62**, 658–665.

[43] Gabor, D. (1946) Theory of communication. part 2: The analysis of hearing. *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, **93**, 442–445.

[44] Georgiou, G. M. and Koutsougeras, C. (1992) Complex domain backpropagation. *IEEE transactions on Circuits and systems II: analog and digital signal processing*, **39**, 330–334.

[45] Girosi, F., Jones, M., and Poggio, T. (1995) Regularization theory and neural networks architectures. *Neural computation*, **7**, 219–269.

[46] Girosi, F. and Poggio, T. (1990) Networks and the best approximation property. *Biological cybernetics*, **63**, 169–176.

[47] Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016) *Deep learning*, vol. 1. MIT press Cambridge.

[48] Grafakos, L. (2004) *Classical and modern Fourier analysis*. Prentice Hall.

[49] Gröchenig, K. (2013) *Foundations of time-frequency analysis*. Springer Science & Business Media.

[50] Gudise, V. G. and Venayagamoorthy, G. K. (2003) Comparison of particle swarm optimization and backpropagation as training algorithms for neural networks. *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*, pp. 110–117, IEEE.

[51] Hagan, M. T. and Menhaj, M. B. (1994) Training feedforward networks with the marquardt algorithm. *IEEE transactions on Neural Networks*, **5**, 989–993.

[52] Hansen, K., Montavon, G., Biegler, F., Fazli, S., Rupp, M., Scheffler, M., Von Lilienfeld, O. A., Tkatchenko, A., and Müller, K.-R. (2013) Assessment and validation of machine learning methods for predicting molecular atomization energies. *Journal of Chemical Theory and Computation*, **9**, 3404–3419.

[53] Häuser, S. and Steidl, G. (2012) Fast finite shearlet transform. *arXiv preprint arXiv:1202.1773*.

[54] Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B. (1998) Support vector machines. *IEEE Intelligent Systems and their applications*, **13**, 18–28.

[55] Hebb, D. O. and Penfield, W. (1940) Human behavior after extensive bilateral removal from the frontal lobes. *Archives of Neurology & Psychiatry*, **44**, 421–438.

[56] Hernandez-Herrera, P., Papadakis, M., and Kakadiaris, I. A. (2016) Multi-scale segmentation of neurons based on one-class classification. *Journal of neuroscience methods*, **266**, 94–106.

[57] Hinton, G. E., Osindero, S., and Teh, Y.-W. (2006) A fast learning algorithm for deep belief nets. *Neural computation*, **18**, 1527–1554.

[58] Hirn, M., Mallat, S., and Poilvert, N. (2017) Wavelet scattering regression of quantum chemical energies. *Multiscale Modeling & Simulation*, **15**, 827–863.

[59] Hirn, M., Poilvert, N., and Mallat, S. (2015) Quantum energy regression using scattering transforms. *arXiv preprint arXiv:1502.02077*.

[60] Hohenberg, P. and Kohn, W. (1964) Inhomogeneous electron gas. *Physical review*, **136**, B864.

[61] Hornik, K. (1991) Approximation capabilities of multilayer feedforward networks. *Neural networks*, **4**, 251–257.

[62] Jacobs, R. A. (1988) Increased rates of convergence through learning rate adaptation. *Neural networks*, **1**, 295–307.

[63] Jarrett, K., Kavukcuoglu, K., LeCun, Y., et al. (2009) What is the best multi-stage architecture for object recognition? *Computer Vision, 2009 IEEE 12th International Conference on*, pp. 2146–2153, IEEE.

[64] Jothilakshmi, G., Sharmila, P., and Raaza, A. (2016) Mammogram segmentation using region based method with split and merge technique. *Indian Journal of Science and Technology*, **9**.

[65] Kabkab, M. (2017) the case for spatial pooling in deep convolutional sparse coding. *Fifty-First Asilomar Conference on Signals, Systems and Computers*.

[66] Kalluri, R. and Weinberg, R. A. (2009) The basics of epithelial-mesenchymal transition. *The Journal of clinical investigation*, **119**, 1420–1428.

[67] Kiefer, J. and Wolfowitz, J. (1952) Stochastic estimation of the maximum of a regression function. *The Annals of Mathematical Statistics*, pp. 462–466.

[68] Kim, T. and Adali, T. (2002) Fully complex multi-layer perceptron network for nonlinear signal processing. *Journal of VLSI signal processing systems for signal, image and video technology*, **32**, 29–43.

[69] Kim, T. and Adalı, T. (2003) Approximation by fully complex multilayer perceptrons. *Neural computation*, **15**, 1641–1666.

[70] King, E. J. (2009) *Wavelet and frame theory: frame bound gaps, generalized shearlets, Grassmannian fusion frames, and p-adic wavelets*. Ph.D. thesis, University of Maryland.

[71] King, E. J., Reisenhofer, R., Kiefer, J., Lim, W.-Q., Li, Z., and Heygster, G. (2015) Shearlet-based edge detection: flame fronts and tidal flats. *Applications of Digital Image Processing XXXVIII*, vol. 9599, p. 959905, International Society for Optics and Photonics.

[72] Krauth, W. and Mézard, M. (1987) Learning algorithms with optimal stability in neural networks. *Journal of Physics A: Mathematical and General*, **20**, L745.

[73] Kutyniok, G., Lim, W.-Q., and Reisenhofer, R. (2014) Shearlab 3d: Faithful digital shearlet transforms based on compactly supported shearlets. *arXiv preprint arXiv:1402.5670*.

[74] Kutyniok, G., Shahram, M., and Donoho, D. L. (2009) Development of a digital shearlet transform based on pseudo-polar fft. *Wavelets XIII*, vol. 7446, p. 74460B, International Society for Optics and Photonics.

[75] Lamouille, S., Xu, J., and Derynck, R. (2014) Molecular mechanisms of epithelial–mesenchymal transition. *Nature reviews Molecular cell biology*, **15**, 178.

[76] LeCun, Y. (1987) *Modèles connexionnistes de l'apprentissage*. Ph.D. thesis, PhD thesis, These de Doctorat, Universite Paris 6.

[77] LeCun, Y., Bengio, Y., and Hinton, G. (2015) Deep learning. *nature*, **521**, 436.

[78] Lee, C.-Y., Gallagher, P. W., and Tu, Z. (2016) Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. *Artificial Intelligence and Statistics*, pp. 464–472.

[79] Leshno, M., Lin, V. Y., Pinkus, A., and Schocken, S. (1993) Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural networks*, **6**, 861–867.

[80] Leung, H. and Haykin, S. (1991) The complex backpropagation algorithm. *IEEE Transactions on Signal Processing*, **39**, 2101–2104.

[81] Lu, Y., Joshi, S., and Morris, J. M. (1997) Noise reduction for nmr fid signals via Gabor expansion. *IEEE Transactions on Biomedical Engineering*, **44**, 512–528.

[82] Mallat, S. (2012) Group invariant scattering. *Communications on Pure and Applied Mathematics*, **65**, 1331–1398.

[83] Mallat, S. G. (1989) A theory for multiresolution signal decomposition: the wavelet representation. *IEEE transactions on pattern analysis and machine intelligence*, **11**, 674–693.

[84] McClelland, J. L., Rumelhart, D. E., and Hinton, G. E. (1986) The appeal of parallel distributed processing. *MIT Press, Cambridge MA*, pp. 3–44.

[85] McCulloch, W. S. and Pitts, W. (1943) A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, **5**, 115–133.

[86] Mhaskar, H. N. (1996) Neural networks for optimal approximation of smooth and analytic functions. *Neural computation*, **8**, 164–177.

[87] Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (2013) *Machine learning: An artificial intelligence approach*. Springer Science & Business Media.

[88] Montavon, G., Hansen, K., Fazli, S., Rupp, M., Biegler, F., Ziehe, A., Tkatchenko, A., Lilienfeld, A. V., and Müller, K.-R. (2012) Learning invariant representations of molecules for atomization energy prediction. *Advances in Neural Information Processing Systems*, pp. 440–448.

[89] Mucherino, A., Papajorgji, P. J., and Pardalos, P. M. (2009) K-nearest neighbor classification. *Data Mining in Agriculture*, pp. 83–106, Springer.

[90] Muckenhoupt, B. (1972) Weighted norm inequalities for the hardy maximal function. *Transactions of the American Mathematical Society*, **165**, 207–226.

[91] Murphy, J. M. (2015) *Anisotropic harmonic analysis and integration of remotely sensed data*. Ph.D. thesis, University of Maryland, College Park.

[92] Navas, T., et al. (2015), Abstract lb-b18: Epithelial to mesenchymal transition in human tumor biopsies: Quantitative, histopathological proof of the existence of emt in vivo by immunofluorescence microscopy.

[93] Navas, T., et al. (2015), Impact of hgf knockin microenvironment on epithelial-mesenchymal transition and cancer stem cells in a non-small cell lung cancer xenograft model.

[94] Negi, P. S. and Labate, D. (2012) 3-d discrete shearlet transform and video processing. *IEEE transactions on Image Processing*, **21**, 2944–2954.

[95] Nieto, M. A. (2011) The ins and outs of the epithelial to mesenchymal transition in health and disease. *Annual review of cell and developmental biology*, **27**, 347–376.

[96] Olson, H., Czaja, W., and Le Moigne, J. (2017) Registration of textured remote sensing images using directional Gabor frames. *Geoscience and Remote Sensing Symposium (IGARSS), 2017 IEEE International*, pp. 2585–2588, IEEE.

[97] Papyan, V., Romano, Y., and Elad, M. (2016) Convolutional neural networks analyzed via convolutional sparse coding. *stat*, **1050**, 27.

[98] Papyan, V., Sulam, J., and Elad, M. (2016) Working locally thinking globally-part ii: Stability and algorithms for convolutional sparse coding. *arXiv preprint arXiv:1607.02009*.

[99] Rezaeilouyeh, H. and Mahoor, M. H. (2016) Automatic gleason grading of prostate cancer using shearlet transform and multiple kernel learning. *Journal of Imaging*, **2**, 25.

[100] Rezaeilouyeh, H., Mahoor, M. H., La Rosa, F. G., and Zhang, J. J. (2013) Prostate cancer detection and gleason grading of histological images using shearlet transform. *Signals, Systems and Computers, 2013 Asilomar Conference on*, pp. 268–272, IEEE.

[101] Riedmiller, M. and Braun, H. (1993) A direct adaptive method for faster backpropagation learning: The rprop algorithm. *Neural Networks, 1993., IEEE International Conference on*, pp. 586–591, IEEE.

[102] Robbins, H. and Monro, S. (1951) A stochastic approximation method. *The annals of mathematical statistics*, pp. 400–407.

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

[104] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986) Learning representations by back-propagating errors. *nature*, **323**, 533.

[105] Rupp, M., Tkatchenko, A., Müller, K.-R., and Von Lilienfeld, O. A. (2012) Fast and accurate modeling of molecular atomization energies with machine learning. *Physical review letters*, **108**, 058301.

[106] Russell, S. J. and Norvig, P. (2016) *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited.

[107] Sampson, G. (1987), Parallel distributed processing: Explorations in the microstructures of cognition.

[108] Schütt, K. T., Arbabzadah, F., Chmiela, S., Müller, K. R., and Tkatchenko, A. (2017) Quantum-chemical insights from deep tensor neural networks. *Nature communications*, **8**, 13890.

[109] Shaham, U., Cloninger, A., and Coifman, R. R. (2016) Provable approximation properties for deep neural networks. *Applied and Computational Harmonic Analysis*.

[110] Staroverov, V. N. (2012) Density-functional approximations for exchange and correlation. *A Matter of Density*, pp. 125–156.

[111] Stein, E. M. (2016) *Topics in Harmonic Analysis Related to the Littlewood-Paley Theory.(AM-63)*, vol. 63. Princeton University Press.

[112] Steinestel, K., Eder, S., Schrader, A. J., and Steinestel, J. (2014) Clinical significance of epithelial-mesenchymal transition. *Clinical and translational medicine*, **3**, 17.

[113] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016) Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826.

[114] Trabelsi, C., Bilaniuk, O., Serdyuk, D., Subramanian, S., Santos, J. F., Mehri, S., Rostamzadeh, N., Bengio, Y., and Pal, C. J. (2017) Deep complex networks. *arXiv preprint arXiv:1705.09792*.

[115] Weinberg, D. (2015) *Multiscale and directional representations of high-dimensional information content in remotely sensed data*. Ph.D. thesis, University of Maryland, College Park.

[116] Weinberger, K. Q., Blitzer, J., and Saul, L. K. (2006) Distance metric learning for large margin nearest neighbor classification. *Advances in neural information processing systems*, pp. 1473–1480.

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

[118] Widrow, B. and Hoff, M. E. (1960) Adaptive switching circuits. Tech. rep., Stanford University, Stanford Electronics Labs.

[119] Zeiler, M. D. and Fergus, R. (2013) Stochastic pooling for regularization of deep convolutional neural networks. *arXiv preprint arXiv:1301.3557*.