Hierarchical Learning: Theory with Applications in Speech and Vision

by

Jacob V. Bouvrie

S.B. Electrical Science and Engineering (2003)
S.B. Mathematics (2003)
M.Eng. Electrical Engineering and Computer Science (2004),
Massachusetts Institute of Technology

Submitted to the Department of Brain and Cognitive Sciences
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2009

Author . . . .
Department of Brain and Cognitive Sciences
July 6, 2009

Certified by . . . .
Tomaso Poggio
Eugene McDermott Professor in the Brain Sciences
Thesis Supervisor

Accepted by . . . .
Earl Miller
Picower Professor of Neuroscience
Chairman, Graduate Committee

Hierarchical Learning: Theory with Applications in Speech and Vision

by

Jacob V. Bouvrie

Submitted to the Department of Brain and Cognitive Sciences
on July 6, 2009, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

Over the past two decades several hierarchical learning models have been developed and applied to a diverse range of practical tasks with much success. Little is known, however, as to why such models work as well as they do. Indeed, most are difficult to analyze, and cannot be easily characterized using the established tools from statistical learning theory.

In this thesis, we study hierarchical learning architectures from two complementary perspectives: one theoretical and the other empirical. The theoretical component of the thesis centers on a mathematical framework describing a general family of hierarchical learning architectures. The primary object of interest is a recursively defined feature map, and its associated kernel. The class of models we consider exploit the fact that data in a wide variety of problems satisfy a decomposability property. Paralleling the primate visual cortex, hierarchies are assembled from alternating filtering and pooling stages that build progressively invariant representations which are simultaneously selective for increasingly complex stimuli.

A goal of central importance in the study of hierarchical architectures and the cortex alike, is that of understanding quantitatively the tradeoff between invariance and selectivity, and how invariance and selectivity contribute towards providing an improved representation useful for learning from data. A reasonable expectation is that an unsupervised hierarchical representation will positively impact the sample complexity of a corresponding supervised learning task. We therefore analyze invariance and discrimination properties that emerge in particular instances of layered models described within our framework. A group-theoretic analysis leads to a concise set of conditions which must be met to establish invariance, as well as a constructive prescription for meeting those conditions. An information-theoretic analysis is then undertaken and seen as a means by which to characterize a model's discrimination properties.

The empirical component of the thesis experimentally evaluates key assumptions built into the mathematical framework. In the case of images, we present simulations which support the hypothesis that layered architectures can reduce the sample complexity of a non-trivial learning problem. In the domain of speech, we describe a localized analysis technique that leads to a noise-robust representation. The resulting biologically-motivated features are found to outperform traditional methods on a standard phonetic classification task in both clean and noisy conditions.

Thesis Supervisor: Tomaso Poggio
Title: Eugene McDermott Professor in the Brain Sciences

Acknowledgments

I would first like to acknowledge my parents for their unwavering support and consistent encouragement throughout my entire educational career. I thank my lovely wife for putting up with me, and for her understanding when I worked on countless weekends and evenings.

I would like to thank Tommy, my advisor since I was a UROP in 2003, through a master's degree, up to this point in 2009. I have been extraordinarily lucky to have the opportunity to learn in a place like CBCL, and to have such an experienced and insightful advisor.

I would like to thank Steve Smale, a source of inspiration with whom it has been an honor to work, and from whom it has been a privilege to learn.

Likewise, I thank Lorenzo Rosasco, from whom I've also learned a great deal – mathematical technicalities and valuable big picture skills alike.

Finally, I thank everyone at CBCL, as well as Srini down the hall, for creating a fun, stimulating, and memorable work environment.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

In the context of human learning, Chomsky's poverty of the stimulus argument captures the notion that biological organisms can learn complex concepts and tasks from extraordinarily small empirical samples. The fact that a child can acquire a visual object concept from a single, possibly unlabeled, example is strong evidence for preexisting cortical mechanisms engineered to facilitate such efficient learning. Indeed, it has been hypothesized that the *hierarchically organized* circuits found in the human brain facilitate robust learning from few examples via the discovery of *invariances*, while promoting circuit modularity and reuse of redundant sub-circuits, leading to greater energy and space efficiency. While the hypothesis linking learning ability and computational organization is both convenient and intuitively compelling, no solid theoretical foundation supports such a connection. Little is known as to how exactly the architectures found in cortex achieve efficient learning and invariance to complex transformations. Classical results in statistical learning theory have shown only that a continuous function can be approximated from an (infinite) empirical sample to an arbitrary degree of accuracy with a single "layer" of computational elements.

From a learning theory perspective, we argue that decomposability of the data is a general principle that allows one to learn good data representations for a wide variety of problems. The class of hierarchical models we consider exploit this property by aggregating simple parts into complex patterns. Alternating filtering and pooling stages are the key components of the hierarchy, building progressively invariant representations which are simultaneously selective for increasingly complex stimuli. A goal of central importance in the study of hierarchical architectures and the mammalian visual cortex alike is that of understanding quantitatively the tradeoff between invariance and selectivity, and how invariance and selectivity contribute towards providing an improved representation useful for learning from data.

This thesis consists of both theoretical and empirical components. The theoretical aspect draws inspiration in part from the observation that learning in hierarchies is a topic that has received little attention in the learning theory literature, and is yet a key step towards a deeper understanding of the brain. Our work seeks to establish a theoretical foundation explaining recent models designed on the basis of anatomical and physiological data describing the primate visual cortex [96, 94]. We attempt to formalize the basic hierarchy of computations underlying visual information processing in the brain by abstracting away from such “engineered” models the elements we believe to be essential to understanding learning in hierarchies from an abstract, mathematical perspective. Within this abstract formalism, we conduct a theoretical analysis uncovering discrimination and invariance properties primarily using tools from functional analysis, group theory, and information theory. In the course of our exploration, the role of unsupervised learning is clearly identified and opportunities for integrating techniques exploiting geometry of the data or sparse coding ideas are highlighted. As a whole, this analysis takes a step towards establishing a rigorous understanding of learning in hierarchies.

The empirical contribution of the thesis follows from a belief that the study of two sensory domains in particular – vision, and the representation and recognition of speech – can provide a uniquely concrete grasp on the relevant theoretical and practical dimensions of the problem of learning in hierarchies. A large part of the empirical component of the thesis therefore centers on the representation and recognition of speech using a hierarchical architecture. This work evaluates experimentally important assumptions built into the above hierarchical learning framework. We adopt the notion of a localized spectro-temporal encoding of speech utterances, and consider speech recognition applications in the presence of noise. Using a representation expressed in terms of a sparse, local 2D-DCT basis, we identify spectro-temporal modulation patterns important for distinguishing among classes of phonemes. Experiments achieving state-of-the-art phonetic classification results are given in support of the approach. Lastly, an algorithm for reconstructing a time-domain signal from modified short-time magnitude-only Fourier spectra is described. Modification of the STFT magnitude is a common technique in speech analysis, however commensurate modification of the phase component is a significant and often insurmountable challenge. If the STFT includes sufficient redundancy due to analysis window overlap, then it is typically the case that a high quality time-domain signal can be reconstructed from magnitude-only spectra.

The speech-related work in this thesis parallels the theoretical component in that it shares the notion of a localized and layered, parts-based analysis such as that occurring in the early stages of the visual and auditory cortices, and in recent computational models of the ventral visual stream. It also complements the theory in the sense that underlying assumptions built into the abstract formalism are evaluated in the context of a difficult, real-world learning task.

The remainder of this Chapter is organized as follows. We first provide some motivation for considering hierarchical learning architectures over shallow alternatives. We then briefly survey the related literature and, finally, end with a concise listing of the contributions and overall organization of the thesis.

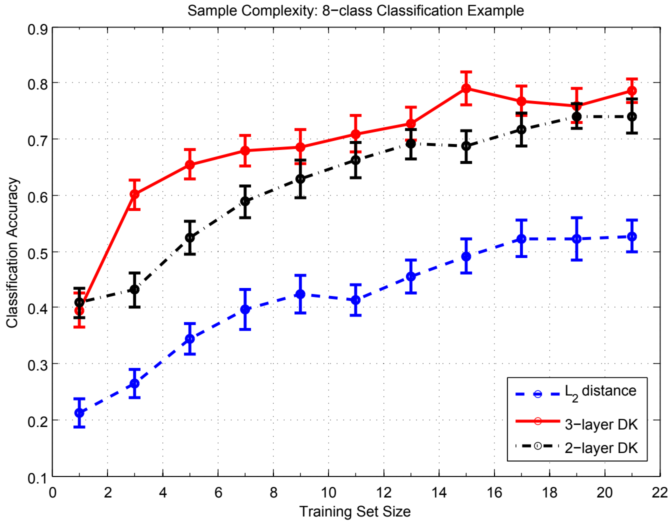


Figure 1-1: Empirical sample complexities of an 8-category image classification task using two and three layer hierarchical models. Training set sizes are the number of labeled examples per class.

1.1 Why Hierarchies?

Why should hierarchies be preferred over shallow alternatives, and why should we seek to understand empirically successful deep hierarchical models? We offer a compilation of observations motivating the use of hierarchical architectures in learning, beginning with a purely empirical, didactic example in Figure 1-1. In this experiment, we have applied two and three layer "derived kernels" introduced in Chapter 2 towards solving a supervised, 8-class handwritten digit (numerals 2 through 9) classification task. The derived kernel provides a hierarchical notion of similarity, and forms the basis for a simple one-nearest-neighbor (1-NN) classification rule whereby the label of an unseen test point is given the label of the most similar labeled training example. Here the corresponding feature map (also see Chapter 2) serves to provide a non-task specific, unsupervised representation for the data that is expected to provide improved performance in the context of a supervised learning problem. We also provide for comparison the performance obtained using the Euclidean distance between the pixels of two images.

The Figure gives classification accuracy averaged over 50 trials, as the number of labeled training examples is varied for each of the three classifiers. Thus, we can choose a given horizontal slice and compare the estimated *sample complexity* of the learning task given each of the models. We find, for example, that in order to obtain 65% accuracy the 2-layer derived kernel based classifier needs about 11 examples per class, while the 3-layer derived kernel based classifier requires only 5 examples. Why does the 3-layer architecture lead to a representation that apparently *halves* the training set size when compared to the 2-layer case? One can see that there is perhaps something interesting underlying this phenomenon, and the example provides much motivation for studying the issue more closely. It is this and other related questions that we seek to address.

In particular,

- Biological organisms can learn complex concepts and tasks from small empirical samples. How can we emulate this remarkable ability in machines? There is evidence that the areas of cortex responsible for processing both auditory data and visual scenes share architectural commonalities, while it has been argued that both language and vision could be the most promising windows into human intelligence.

- No solid theoretical foundation supports the connection between generalization ability and computational organization. Can we cast and understand learning in hierarchies using tools from statistical learning theory?

- Hierarchically organized circuits in the human brain exploit circuit modularity and reuse general sub-circuits in order to economize on space and energy consumption. In a hierarchical model, lower layers might include dictionaries of features that are general and yet applicable in the context of many specific classification tasks.

- Hierarchical models are ideally suited to domains and tasks which decompose into parts, such as those based on naturally occurring phenomena.

- Hierarchies can be used to incorporate particular, pre-defined invariances in a straightforward manner, by e.g. the inclusion of pooling, and local transformations.

- If one doesn't know how to characterize variation in the data, or even know what kinds of variation needs to be captured, nonparametric representations with randomly sampled exemplars identify patterns that have been seen before, whatever they might be, while maintaining the hierarchy assumption of the domain.

The success of recent hierarchical models (reviewed in the following section) have provided a source of motivation, among others, for the work presented in this thesis. An engineered model, however, is not sufficient to explain what the cortex doing because interpreting the inner workings and representations of such models can be extraordinarily difficult in and of itself. We therefore argue that what is needed is a *mathematical theory* that can, ideally, explain why a hierarchy solves the problem and what the optimal parameter values should be. A theory would ultimately lead to concrete predictions for neuroscience experiments, provide insights into how the brain computes, and would immediately suggest algorithms which imitate the brain.

A theory may explain why hierarchical models work as well as they do, and shed light on the computational reasons for the hierarchical organization of cortex, leading to potentially significant contributions to outstanding challenges in computer vision and artificial intelligence, among other fields.

1.2 Literature Survey

In this survey we discuss previous work related to the derived kernel formalism introduced in Chapter 2, as well as work that attempts to address some aspect of the larger, overarching questions discussed above. We begin with a brief review of the prior work involving learning in hierarchies. Following deep neural networks and belief networks, we discuss engineered models of visual cortex where we draw particular attention to the model of Serre et al. [96, 98]. Finally, because our work can be seen as an unsupervised pre-processing step that can be used to improve supervised classification, we review distance learning as it has been approached more generally in the literature.

1.2.1 Deep Neural Networks/Deep Belief Networks

Much work has been done to understand single (hidden) layer neural networks in terms of statistical learning theory [79, 43]. There is little work that attempts to do the same for multilayer neural networks with nonlinear activation functions. Bartlett and Mendelson [8] have derived generalization guarantees using Rademacher complexities for classes of functions defined by nonlinear multilayer neural networks, however their work does not connect those classes of functions with the regularization viewpoint found in the statistical learning theory literature. In the context of regularization, the neural network community has largely resorted to a handful of effective but little understood heuristics such as weight decay and pruning techniques [19].

The study of “deep” belief networks (networks with many layers) has enjoyed much attention in recent years, following the publication of a paper by Hinton and Salakhutdinov [49], rekindling interest within the machine learning community. Although the notion of a deep architecture was not new, Hinton and Salakhutdinov provided a novel training algorithm that finally enabled good solutions to be realized. Prior to this work, deep neural networks trained with standard backpropagation algorithms would slowly converge to poor local minimums for most practical problems. Hinton and Salakhutdinov go on to demonstrate that deep autoencoders can provide excellent classification performance for handwritten digits and human faces. Little is known however, as to *why* deep networks work well, and why or when particular architectural choices (number of layers, number of units per layer, etc.) lead to optimal performance.

The relevant work involving deep architectures since [49] can be roughly divided into two categories: empirical applications and algorithms on the one hand, and theoretical attempts to understand deep networks on the other. The majority of the literature to date is applied in nature, and dwells on efficient algorithms for training and inference in deep models [61, 12], as well as empirical experiments in specific application domains [12, 83, 62, 66]. We will nevertheless focus almost exclusively on the more theoretical literature as it is most relevant to the work described in the thesis.

A recent position paper due to LeCun and Bengio [64] puts forth a series of arguments demonstrating the advantages of deep architectures. Boolean function learning is considered first, and although the case of boolean functions (such as the d-bit parity function) does support the use of deep networks, it is not clear how much such examples demonstrate about practical problems involving the approximation of functions far more complex than boolean products or other discrete examples.

LeCun and Bengio go on to consider particular problems where a Gaussian kernel SVM would ostensibly require more examples than a multi-layer architecture. However the argument they present highlights a problem which lies in the fact that the Gaussian variance parameter, in the particular case of the SVM algorithm, is fixed at all knots. We believe that this argument cannot be used to demonstrate that one layer of computations (whatever they may be) is inferior to multiple layers: one could consider a single layer of radial basis functions with different bandwidths (see "Generalized RBF Networks" in [78]). A comparison between architectures independent of the particular implementation algorithms (while certainly difficult) would give much stronger conclusions. Although LeCun and Bengio make a commendable set of arguments for deep networks, their work is limited to special examples and does not provide a rigorous justification; the informal arguments are intuitively appealing but do not constitute a theory.

Finally, in recent work by LeRoux and Bengio [92], open questions concerning the expressive power of deep belief networks are stated but not answered. The authors do find that the performance of a two-layer network is limited by the representational ability of the first layer, thereby suggesting that a larger first layer is preferable. LeRoux and Bengio go on to surmise that the extra layers primarily contribute by providing better generalization (for a fixed sample size), rather than adding representational power per se.

1.2.2 Distance Metric Learning

The hierarchical formalism we discuss in later chapters seeks to provide a natural similarity concept. Although the construction of this similarity metric is unsupervised and non-task specific, it can subsequently be used to solve supervised learning tasks when viewed as a preprocessing step. From this perspective our work can be connected to the literature on distance metric learning, where an "intelligent" similarity/distance measure is learned and then later used in a supervised classification algorithm. A portion of the research in this field attempts to learn a suitable metric from only relative comparisons between the data [112], however we assume that the data themselves are available and will not discuss relative comparison methods.

Several recent attempts [7, 24, 101] have been made to construct a distance based on ideas related to PCA, and can in fact be summarized as "anti-PCA": the goal is to learn a projection of the data under which variance along coordinates with a large amount of within-class scatter is eliminated. Recent work due to Maurer [69, 70] extends this intuition to the case of general linear operators on reproducing kernel Hilbert spaces, and gives generalization bounds and optimization algorithms. In this work, a Hilbert-space valued stochastic process is considered. Variation of the derivative under the projector to be learned is minimized, while variance of the process is maximized. What is particularly interesting about this approach is that it presents a formal notion of invariance by asserting that successive pairs of images in, for example, a video sequence, should be "close", while frames separated by greater amount of time ought to be "far" according to a ground-truth oracle. Maurer goes on to show that learning a similarity metric suitable for a given task can at times transfer to other tasks.

While the work on distance learning to date highlights promising theoretical avenues, it does not address the advantages or disadvantages of learning distances with a hierarchical modeling assumption.

1.2.3 Hierarchical Models of Visual Cortex

The process of object recognition in the visual cortex (the "what" pathway) begins in the low-level primary area $V1$ [52] and proceeds in a roughly bottom-up fashion through areas V2 and V4, terminating in inferotemporal cortex (IT). Afterwards, information from IT travels to prefrontal areas (PFC) and plays a role in perception, memory, planning, and action.

Numerous models retaining varying degrees of faithfulness to neurobiology have been proposed [40, 117, 71, 87, 113, 109, 2, 118, 120]. A recurring theme in these efforts is the repeated pooling of simple, local detector outputs, in analogy to simple and complex cells in primary visual cortex. Most of the research on neurobiological models does not simulate such networks down to the level of spikes, although [109] is a notable exception. The overarching goal of these investigations is to capture the essential computational components underlying the visual system by modeling and simulating information processing in the early stages of the primate visual cortex. In doing so, a central concern is that of finding a suitable trade off between simplicity and biological detail. Much of this work is therefore computational in nature, and seeks to *reproduce* the abilities of the cortex, rather than directly *mathematically characterize* those abilities.

Several attempts have been made to abstract information processing in the visual cortex, and construct a probabilistic picture. This approach takes a step away from biological realism and moves towards a more analytically tractable setting where the tools of probability calculus and Bayesian inference can be applied. The perspective taken by Lee and Mumford [67] in particular formalizes earlier ideas [85, 84] concerning the exchange and propagation of information between early layers of the ventral visual hierarchy and the rest of the visual system. In a noteworthy departure from much of the neurobiological modeling literature, Lee and Mumford apply well-established tools known to the Bayesian statistics community to model top-down (feedback) as well as bottom-up (feedforward) activity. Backprojections, the authors argue, serve to reconcile experiential prior beliefs with sensory evidence. Several subsequent efforts (e.g. [74, 66]) elaborate on this theme, incorporating for example sparse overcomplete dictionaries at each layer following the observation that there are far more cells in V1 than strictly necessary to represent information collected at the retina. The Bayesian formalism underlying the argument that hierarchies facilitate the integration of priors (nodes at higher layers in the network) and bottom-up observations (lower layers) is attractive. However, the majority of these efforts focus on (1) engineering practically useful systems and (2) modeling information processing in the brain in computationally tractable terms. Finally, another noteworthy effort is that of Yu and Slotine [120], where a simplified model paralleling [98] is described in clean terms by a hierarchy of wavelets, lending additional interpretability in terms of standard notions in signal processing.

The work we have discussed, however, does not directly attempt to compare hierarchies to single layer architectures, or attempt to mathematically analyze the connection between model architecture and invariance, discrimination, or generalization.

The CBCL Model

In this thesis, particular emphasis will be placed on connections to a layered model of object recognition developed at the Center for Biological and Computational Learning (CBCL) [96, 98]. This work builds on several previous efforts which describe the neurobiology of the visual cortex [52, 77, 50, 14], and bears a resemblance to existing deep recognition models in the computer vision community (e.g. [65]). We will refer to this model as the "CBCL model". A central theme found in this family of models is the use of Hubel and Wiesel's simple and complex cell ideas [52]. In the visual cortex, features are computed by simple ("S") units by looking for the occurrence of a preferred stimulus specific to the unit in a region of the input ("receptive field"). Translation invariance is then explicitly built into the processing pathway by way of complex ("C") units which pool over localized simple units. The alternating simple-complex filtering/pooling process is repeated, building increasingly invariant representations which are simultaneously selective for increasingly complex stimuli. In a computer implementation, the final representation can then be presented to a supervised learning algorithm.

The model [96] is initially trained using unsupervised techniques on non-specific natural images, compiling a "universal dictionary" of features at each layer. Each feature constitutes a computational unit, which is said to be tuned to a specific preferred stimulus. A set of features are computed by simple units at each layer, where the preferred stimuli are derived from randomly sampled patches of the natural images [96]. Varying degrees of translation and scale invariance are explicitly built into the model by way of complex-like units which pool over localized simple units using the max operation. The alternating simple-complex filtering/pooling process is repeated up to three times, and the final representation is then tapped and fed into a supervised learning algorithm. Various experiments have shown that although the feature dictionaries are generic, performance on specific, supervised image categorization tasks is typically competitive with respect to other state-of-the-art computer vision systems [96, 99, 75].

What sets the CBCL model apart from prior work, is that it was designed with much guidance from the neuroscience, neurophysiology and psychophysics literature, and has been shown to account for a range of recent anatomical and physiological data [87, 97, 96]. In particular, "recordings" from the model have revealed units which predict in some cases and mimic in others properties of cells in V1, V4, IT and PFC [39]. The model has also been shown to reproduce (see Figure 4.17 in [96]) human performance on a set of rapid categorization tasks [111, 110, 114, 91, 6]). The combination of empirical success and lack of interpretability of this particular model provided the initial impetus for the theoretical work exploring learning in hierarchies we present in the thesis.

1.3 Contributions and Organization of the Thesis

We summarize the main contributions of the thesis in order by Chapter.

Chapter 2: Towards a Theory of Hierarchical Learning

This chapter contains the central results of the thesis. We formalize the basic hierarchy of computations underlying information processing in the visual cortex by abstracting away the elements essential to understanding learning in hierarchies from a mathematical perspective. In doing so, we establish both feature map and kernel based recurrence relations, and decompose a given layer's output into distinct filtering and pooling steps. We then show that different initial "mother" kernel choices induce global invariance in the final output of the model. In particular, we find that under mild assumptions our feature map can be made invariant to rotations and reflections in the case of images, and for strings, is reversal invariant.

We also evaluate whether a hierarchy can discriminate well by characterizing the equivalence classes of inputs in the context of 1-D strings, for a natural instance of the model. The role of templates and unsupervised learning is also addressed, and we describe how a given probability measure over natural images can induce a measure on patches of images, from which one might sample an empirical dictionary of templates. Extensions are then given, outlining a procedure for incorporating additional feature maps at each layer. These feature maps may be based on, for example, PCA, Laplacian Eigenmaps [9], or other techniques.

We then give an efficient algorithm for computing the neural response, with complexity linear in the number of layers. With this algorithm in hand, we consider the extent to which the mathematical notion of similarity defined by the model corresponds to similarity in the way humans view images by conducting experiments in which the recursively defined kernels are applied to a real-world image classification task. We additionally show empirically the effect of varying the most important model parameters (number of layers and patch sizes), compare the empirical sample complexities induced by two and three layer representations, and evaluate an unsupervised template learning procedure.

Chapter 3: Entropy and Discrimination Properties of the Neural Response

Our work in this chapter follows from the observation that parameter choice questions are often times synonymous with theory questions. We present preliminary ideas towards answering how many layers is optimal for a given task, and how many templates should be chosen. We suggest that Shannon entropy is a promising tool for systematic study of the discrimination ability of a hierarchy given different architectural choices.

Chapter 4: Group-theoretic Perspectives on Invariance

Here we more closely examine invariance properties of the model using the tools of group theory. We begin by giving a more general proof of invariance, allowing for hierarchies involving general pooling functions not limited to the max. We then make a key observation: one can often endow the transformations of interest with a group structure, and then dissociate group-related conditions leading to invariance from conditions related to the architecture (such as restriction, patch sizes, and the bottom-up propagation of information). We provide a concise set of conditions which lead to invariance, as well as a constructive prescription for meeting those conditions. The analysis additionally reveals that orthogonal transformations are the only group of transformations compatible with our notion of invariance in the neural response. In the case of hierarchies defined on length n strings, we show that considering reversal invariance leads to a group of symmetries isomorphic to the well known Dihedral group D_n describing symmetries of a regular n-sided polygon.

Chapter 5: Localized Spectro-Temporal Cepstral Analysis of Speech

Our speech recognition work follows, first and foremost, from the belief that the study of the computations supporting speech recognition and production can shed light on the problem of human intelligence. Aided by recent discoveries in auditory neuroscience and the availability of new physiological data, we present an algorithm for noise-robust speech analysis inspired by the early stages of the auditory cortex. Our algorithm is the most recent within a series of localized analysis techniques [36, 35, 34] which begins with an encoding based on sparse Gabor projections and ends with a local 2-D DCT representation. The technique can be thought of as imposing a form of localized smoothing, and we show that our approach preserves patterns important for discriminating among classes of phonemes: When applied to a phonetic classification task, our method achieved state of the art results in clean conditions, and showed marked improvement in a range of noisy conditions as compared to alternative features.

This work can be interpreted as a simplified case of the first layer of the CBCL model applied to spectrograms. In this setting, scale invariance is not incorporated, and the "S1" filters are low-order 2D-DCT basis functions rather than Gabors. In a separate study [89], we evaluated the performance of one instance of the CBCL model on a related phonetic classification task and found that the first layer's features alone often led to the best classification performance. In light of this evidence, and in the interest of computational efficiency, we do not iterate the 2D-DCT analysis.

Chapter 6: Signal Reconstruction from STFT Magnitude

In this Chapter, we introduce a novel a phase-retrieval algorithm for recovering a time-domain signal from magnitude-only STFT spectra. The algorithm responds to the need to reconstruct a time-domain signal from a modified STFT magnitude, when no corresponding phase component exists. Our particular application was originally that of inter-voice speech morphing. Given the same utterance produced by two different speakers, an inter-voice speech morph is a perceptually smooth sequence of transformations from one signal towards the other, and is controlled by a single mixing parameter. Within the context of this application, a "morphed" magnitude spectrum was used to reconstruct an audio utterance. Our algorithm was evaluated and later compared to other methods in the literature in a master's thesis co-authored by M. Jensen and S. Nielsen [53], and was found to give superior results in some cases.

Chapter 7: Concluding Discussion

We conclude with a brief discussion comparing the hierarchical learning framework described in Chapter 2 to other deep learning architectures in the literature, principally deep belief networks. We argue that a key aspect of our model is the interplay between invariance and discrimination, and reiterate the need to present hierarchical learning in a language that is amenable to mathematical analysis. Finally, we provide answers to commonly raised criticisms involving the hierarchical formalism we introduced an analyzed in earlier chapters. The criticisms we have chosen to address concern both the motivation and technical details underlying the neural response.

Chapter 2

Towards a Theory of Hierarchical Learning

This chapter includes joint work and text previously appearing in [105].

2.1 Introduction

The goal of this chapter is to define a distance function on a space of images which reflects how humans see the images. The distance between two images corresponds to how similar they appear to an observer. Most learning algorithms critically depend on a suitably defined similarity measure, though the theory of learning so far provides no general rule to choose such a similarity measure [115, 28, 93, 29]. In practice, problem specific metrics are often used [100]. It is natural, however, to ask whether there are general principles which might allow one to learn data representations for a variety of problems. In this Chapter we argue that such a principle can be a decomposability property which is satisfied in many domains: we will assume the data to be composed of a hierarchy of parts, and propose a natural image *representation*, the neural response, motivated by the neuroscience of the visual cortex. The derived kernel is the inner product defined by the neural response and can be used as a similarity measure.

The definition of the neural response and derived kernel is based on a recursion which defines a hierarchy of local kernels, and can be interpreted as a multi-layer architecture where layers are associated with increasing spatial scales. At each layer, (local) derived kernels are built by recursively *pooling* over previously defined local kernels. Here, pooling is accomplished by taking a max over a set of transformations, although other forms of pooling are possible. This model has a key semantic component: a system of templates which link the mathematical development to real world problems. In the case of images, derived kernels consider *sub-patches* of images at intermediate layers and whole images at the last layer. Similarly, in the case of derived kernels defined on strings, kernels at some m-th layer act on sub-strings. From a learning theory perspective the construction of the derived kernel amounts to an unsupervised learning step and the kernel can ultimately be used to solve supervised as well as unsupervised tasks. The motivating idea is that the unsupervised preprocessing will reduce the sample complexity of a corresponding supervised task.

The work in this chapter sets the stage for further developments towards a theory of vision. We consider two complementary directions, one empirical, the other mathematical. The empirical involves numerical experiments starting with databases coming from real world situations. The goal is to test (with various algorithmic parameters) how the similarity derived here is consistent with real world experience. In vision, to what extent does the mathematical similarity correspond to similarity in the way humans view images? In Section 2.6 we show the results of some experimental work towards this end. On the purely mathematical side, the problem is to examine how closely the output response characterizes the input. In other words, does the neural response discriminate well? In the case of strings, it is shown in Theorem 2.4.1 that if the architecture is rich enough and there are sufficient templates ("neurons") then the neural response is indeed discriminative (up-to reversal and "checkerboard" patterns). Note that in this unsupervised context, discrimination refers to the ability to distinguish images of distinct objects in the real world. We show under quite mild assumptions that the neural response is invariant under rotations, and for strings, is reversal invariant. We additionally suggest that the Shannon entropy is a promising tool for obtaining a systematic picture. An elaborated development of these ideas is presented in Chapter 3.

The use of deep learning architectures to obtain complex data representations has recently received considerable interest in machine learning, see for example [49, 65, 66]. Hierarchical parts-based descriptions date back to [40], and have become common in computer vision [71, 65, 118, 31, 63, 98]. The principle of decomposability of the input data is also at the heart of several Bayesian techniques [42, 67]. Our work also seeks to establish a theoretical foundation for recent models designed on the basis of anatomical and physiological data describing the primate visual cortex [40, 65, 87, 98, 96]. These models quantitatively account for a host of psychophysical and physiological data, and provide human-level performance on tasks involving rapid categorization of complex imagery [95, 98, 96]. The development proposed here considerably generalizes and simplifies such models, while preserving many of their key aspects. We argue that the analytical picture achieved by working with derived kernels allows for deeper and more revealing analyses of hierarchical learning systems. Indeed, the hierarchical organization of such models – and of the cortex itself – remains a challenge for learning theory as most "learning algorithms", as described in [80], correspond to one-layer architectures. Our hope is to ultimately achieve a theory that may explain why such models work as well as they do, and give computational reasons for the hierarchical organization of the cortex.

The chapter is organized as follows. We begin by introducing the definitions of the neural response and derived kernel in Section 2.2. We study invariance properties of the neural response in Section 2.3 and analyze discrimination properties in a one-dimensional setting in Section 2.4. In Section 2.5 we suggest that Shannon entropy can be used to understand the discrimination properties of the neural response. We conclude with experiments in Section 2.6. Finally, in a brief postscript we establish detailed connections with the model in [98] and identify a key difference with the basic framework developed in this chapter.

2.2 The Derived Kernel and Neural Response

The derived kernel can be thought of as a notion of similarity on spaces of functions on patches and can be defined via a recursion of kernels acting on spaces of functions on sub-patches. Before giving a formal description we present a few preliminary concepts.

2.2.1 Preliminaries

The ingredients needed to define the derived kernel consist of:

- an architecture defined by a finite number of nested patches (for example sub-domains of the square $Sq \subset \mathbb{R}^2$),
- a set of transformations from a patch to the next larger one,
- a suitable family of function spaces defined on each patch,
- a set of templates which connect the mathematical model to a real world setting.

We first give the definition of the derived kernel in the case of an architecture composed of three layers of patches u, v and Sq in $\mathbb{R}^2$, with $u \subset v \subset Sq$, that we assume to be square, centered and axis aligned (see Figure 2-1). We further assume that we are given a function space on Sq, denoted by $\mathrm{Im}(Sq)$, as well as the function spaces $\mathrm{Im}(u)$, $\mathrm{Im}(v)$ defined on subpatches u, v, respectively. Functions are assumed to take values in $[0, 1]$, and can be interpreted as grey scale images when working with a vision problem for example. Next, we assume a set H_u of *transformations* that are maps from the smallest patch to the next larger patch $h : u \to v$, and similarly H_v with $h : v \to Sq$. The sets of transformations are assumed to be finite and in this work are limited to translations; see remarks in Section 2.2.2. Finally, we are given *template sets* $T_u \subset \mathrm{Im}(u)$ and $T_v \subset \mathrm{Im}(v)$, assumed here to be finite and endowed with the uniform probability measure.

The following fundamental assumption relates function spaces and transformation spaces.

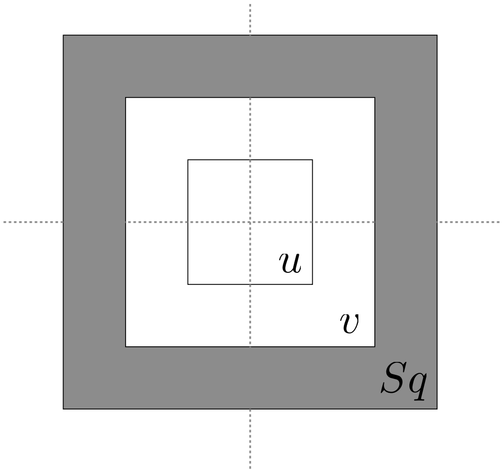


Figure 2-1: Nested patch domains.

Axiom. *$f \circ h : u \to [0,1]$ is in $\mathrm{Im}(u)$ if $f \in \mathrm{Im}(v)$ and $h \in H_u$. Similarly $f \circ h : v \to [0,1]$ is in $\mathrm{Im}(v)$ if $f \in \mathrm{Im}(Sq)$ and $h \in H_v$.*

We briefly recall the general definition of a reproducing kernel [3]. Given some set X, we say that a function $K : X \times X \to \mathbb{R}$ is a reproducing kernel if it is a symmetric and positive definite kernel, i.e.

$$\sum_{i,j=1}^{n} \alpha_i \alpha_j K(x_i, x_j) \geq 0$$

for any $n \in \mathbb{N}$, $x_1, \ldots, x_n \in X$ and $\alpha_1, \ldots, \alpha_n \in \mathbb{R}$. In this work we deal with inner product kernels which are known to be an instance of reproducing kernels.

In the following we always assume $K(x,x) \neq 0$ for all $x \in X$ and denote with $\widehat{K}$ kernels normalized according to

$$\widehat{K}(x,x') = \frac{K(x,x')}{\sqrt{K(x,x)K(x',x')}}. \tag{2.1}$$

Clearly in this case $\widehat{K}$ is a reproducing kernel and $\widehat{K}(x,x) \equiv 1$ for all $x \in X$. The kernel normalization avoids distortions traveling up the hierarchy, and provides a more interpretable as well as comparable quantity.

2.2.2 The Derived Kernel

Given the above objects, we can describe the construction of the derived kernel in a bottom-up fashion. The process starts with some *normalized* initial reproducing kernel on $\mathrm{Im}(u) \times \mathrm{Im}(u)$ denoted by $\widehat{K}_u(f, g)$ that we assume to be non-negative valued. For example, one could choose the usual inner product in the space of square integrable functions on u,

$$K_u(f, g) = \int_u f(x)g(x)dx.$$

Next, we define a central object of study, the *neural response* of f at t:

$$N_v(f)(t) = \max_{h \in H} \widehat{K}_u(f \circ h, t), \tag{2.2}$$

where $f \in \mathrm{Im}(v)$, $t \in T_u$ and $H = H_u$. The neural response of f is a map $N_v(f) : T_u \to [0, 1]$ and is well defined in light of the Axiom. By denoting with $|T_u|$ the cardinality of the template set T_u, we can interpret the neural response as a vector in $\mathbb{R}^{|T_u|}$ with coordinates $N_v(f)(t)$, with $t \in T_u$. It is then natural to define the corresponding inner product on $\mathbb{R}^{|T_u|}$ as $\langle \cdot, \cdot \rangle_{L^2(T_u)}$ – the L^2 inner product with respect to the uniform measure $\frac{1}{|T_u|} \sum_{t \in T_u} \delta_t$, where we denote by δ_t the Dirac measure. If the initial kernel is taken to be the inner product, then one can compare this process, for each template, to taking the maximum of a normalized cross-correlation without centering.

We note that invariance to transformation is enforced by pooling over transformations via the max, where as additional *selectivity* for a transformation can be achieved by including in the template set T_u transformed versions of the templates.

The derived kernel on $\mathrm{Im}(v) \times \mathrm{Im}(v)$ is then defined as

$$K_v(f, g) = \Big\langle N_v(f), N_v(g) \Big\rangle_{L^2(T_u)}, \tag{2.3}$$

and can be normalized according to (2.1) to obtain the kernel $\widehat{K}_v$. The kernel K_v can be interpreted as the correlation in the pattern of similarities to templates at the previous layer.

We now repeat the process by defining the second layer neural response as

$$N_{Sq}(f)(t) = \max_{h \in H} \widehat{K}_v(f \circ h, t), \tag{2.4}$$

where in this case $f \in \mathrm{Im}(Sq), t \in T_v$ and $H = H_v$. The new derived kernel is now on $\mathrm{Im}(Sq) \times \mathrm{Im}(Sq)$, and is given by

$$K_{Sq}(f, g) = \Big\langle N_{Sq}(f), N_{Sq}(g) \Big\rangle_{L^2(T_v)}, \tag{2.5}$$

where $\langle \cdot, \cdot \rangle_{L^2(T_v)}$ is the L^2 inner product with respect to the uniform measure $\frac{1}{|T_v|} \sum_{t \in T_v} \delta_t$. As before, we normalize K_{Sq} to obtain the final derived kernel $\widehat{K}_{Sq}$.

The above construction can be easily generalized to an n layer architecture given by sub-patches $v_1 \subset v_2 \subset \cdots \subset v_n = Sq$. In this case we use the notation $K_n = K_{v_n}$ and similarly $H_n = H_{v_n}$, $T_n = T_{v_n}$. The definition is given formally using induction.

Definition 2.2.1 (Derived Kernel)**.** Given a non-negative valued, normalized, initial reproducing kernel $\widehat{K}_1$, the m-layer derived kernel $\widehat{K}_m$, for $m = 2, \dots, n$, is obtained by normalizing

$$K_m(f,g) = \Big\langle N_m(f), N_m(g) \Big\rangle_{L^2(T_{m-1})}$$

where

$$N_m(f)(t) = \max_{h \in H} \widehat{K}_{m-1}(f \circ h, t), \qquad t \in T_{m-1}$$

with $H = H_{m-1}$.

One can also define a family of kernels on elements of $\mathrm{Im}(Sq)$ similar to the derived kernel defined above, but with *global* rather than local pooling at the top most layer. Although a derived kernel architecture can be constructed to match any global pooling kernel with an appropriate choice of patches, defining a set of global kernels given a derived kernel architecture first, is more natural. The global pooling kernel is most useful when seen as an alternative kernel defined on patches corresponding to a predefined derived kernel. In this sense, the global pooling kernels describe a bottom-up processing flow which is halted at some intermediate layer, and subjected to global pooling rather than further localized pooling.

Definition 2.2.2 (Global Pooling Kernel)**.** For $m = 2, \dots, n$, let H^G_{m-1} be sets of transformations $h : v_{m-1} \to Sq$ mapping patches to Sq. The family $\{\widehat{G}_m : \mathrm{Im}(Sq) \times \mathrm{Im}(Sq) \to [0,1]\}_{m=2}^n$ of "global pooling" kernels is obtained by normalizing

$$G_m(f,g) = \Big\langle N^G_m(f), N^G_m(g) \Big\rangle_{L^2(T_{m-1})}$$

where

$$N^G_m(f)(t) = \max_{h \in H} \widehat{K}_{m-1}(f \circ h, t), \qquad t \in T_{m-1}$$

with $H = H^G_{m-1}$.

The definition is identical to that of Definition 2.2.1 up to the penultimate layer, except the final layer is modified to include transformations of the form $h : v_{m-1} \to Sq$ with $m \leq n$. If $m = n$ then global pooling coincides with the usual range of the max operation appearing in Definition 2.2.1, and the two definitions are equivalent: $\widehat{G}_n = \widehat{K}_n$. Note that the global pooling kernel G_m is *not* recursively defined in terms of the global pooling kernel G_{m-1}.

We add some remarks.

Remarks

- Examples of transformations are translations, scalings and rotations. Combining the first two, we have transformations of the form $h = h_\beta h_\alpha$, $h_\alpha(x) = \alpha x$

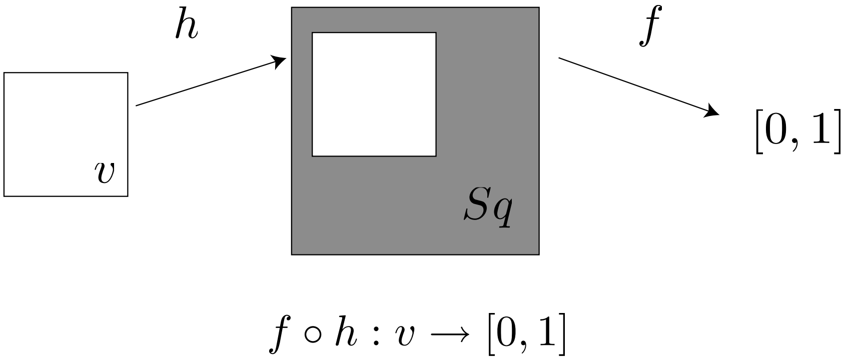


Figure 2-2: A transformation "restricts" an image to a specific patch.

and $h_\beta(x') = x' + \beta$, where $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^2$ is such that $h_\beta h_\alpha(u) \subset v$. The transformations are embeddings of u in v and of v in Sq. In the vision interpretation, a translation h can be thought of as moving the image over the "receptive field" v: see Figure 2-2.

- To make sense of the normalization (2.1) we rule out the functions such that $K(f, f)$ is zero. This condition is quite natural in the context of images since for $K(f, f)$ to be zero, the neural responses of f would have to be identically zero at *all* possible templates by definition, in which case one "can't see the image".

- In the following, we say that some function $g \in \mathrm{Im}(v_{m-1})$ is a patch of a function $f \in \mathrm{Im}(v_m)$, or simply a *function patch* of f, if $g = f \circ h$ for some $h \in H_{m-1}$. If f is an image, we call g an *image patch*, if f is a string, we call g a *substring*.

- The derived kernel naturally defines a derived distance d on the space of images via the equation

$$d^2(f, g) = \widehat{K}(f, f) + \widehat{K}(g, g) - 2\widehat{K}(f, g) = 2\Big(1 - \widehat{K}(f, g)\Big). \qquad (2.6)$$

  where we used the fact that normalization implies $\widehat{K}(f, f) = 1$ for all f. Clearly, as the kernel "similarity" approaches its maximum value of 1, the distance goes to 0.

- The choice of the "max" as the pooling operation is natural and conforms to the model in [96]. An interesting problem would be to explore the properties induced by different pooling operations.

- Although we draw on the example of vision as an interpretation of our model, the setting is general and is not limited to strings or images.

- One might also consider "input-dependent" architectures, wherein a preliminary preprocessing of the input data determines the patch sizes. For example, in the case of text analysis one might choose patches of size equal to a word, pair of words, and so on, after examining a representative segment of the language in question.

In the following section, we discuss in more detail the nature of the function spaces and the templates, as well as the interplay between the two.

2.2.3 Probability on Function Spaces and Templates

We assume $\mathrm{Im}(Sq)$ is a probability space with a "mother" probability measure ρ. This brings the model to bear on a real world setting. We discuss an interpretation in the case of vision. The probability measure ρ can be interpreted as the frequency of images observed by a baby in the first months of life. The templates will then be the most frequent images and in turn these images could correspond to the neurons at various stages of the visual cortex. This gives some motivation for the term "neural response". We now discuss how the mother probability measure ρ iteratively defines probability measures on function spaces on smaller patches. This eventually gives insight into how we can collect templates, and suggests that they can be best obtained by randomly sampling patches from the function space $\mathrm{Im}(Sq)$.

For the sake of simplicity we describe the case of a three layer architecture $u \subset v \subset Sq$, but the same reasoning holds for an architecture with an arbitrary number of layers. We start by describing how to define a probability measure on $\mathrm{Im}(v)$. Let the transformation space $H = H_v$ be a probability space with a measure ρ_H, and consider the product space $\mathrm{Im}(Sq) \times H$ endowed with a probability measure P that is the product measure given by the probability measure ρ on $\mathrm{Im}(Sq)$ and the probability measure ρ_H on H. Then we can consider the map $\pi = \pi_v : \mathrm{Im}(Sq) \times H \to \mathrm{Im}(v)$ mapping (f, h) to $f \circ h$. This map is well defined given the Axiom. If $\mathrm{Im}(v)$ is a measurable space we can endow it with the pushforward measure $\rho_v = P \circ \pi^{-1}$ (whose support is typically a proper subset of $\mathrm{Im}(v)$).

At this point we can naturally think of the template space T_v as an i.i.d. sample from ρ_v, endowed with the associated empirical measure.

We can proceed in a similar way at the lower layer. If the transformation space H_u is a probability space with measure ρ_{H_u}, then we can consider the product space $\mathrm{Im}(v) \times H_u$ endowed with a probability measure $P_u = \rho_v \times \rho_{H_u}$, with ρ_v defined as above. The map $\pi_u : \mathrm{Im}(v) \times H_u \to \mathrm{Im}(u)$ is again well defined due to the Axiom, and if $\mathrm{Im}(u)$ is a measurable space, then we can endow it with the pushforward measure $\rho_u = P_u \circ \pi_u^{-1}$. Similarly, the template space T_u can then be thought of as sampled according to ρ_u and endowed with the corresponding empirical measure. As mentioned before, in the case of several layers one continues by a similar construction.

The above discussion highlights how the definition of the templates as well as the other operations involved in the construction of the derived kernels are purely *unsupervised*; the resulting kernel can eventually be used to solve supervised as well as unsupervised tasks.

2.2.4 The Normalized Neural Response

In this section we focus on the concept of (normalized) neural response which is as primary as that of the derived kernel. The normalized neural response at f, denoted by $\widehat{N}(f)$, is simply $\widehat{N}(f) = N(f)/\|N(f)\|_{L^2(T)}$, where we drop subscripts to indicate that the statement holds for any layer m within an architecture, with $m-1$ the previous layer.

The normalized neural response provides a natural *representation* for any function f. At the top layer, each input function is mapped into an output representation which is the corresponding neural response

$$\underbrace{f \in \mathrm{Im}(Sq)}_{\text{input}} \longmapsto \underbrace{\widehat{N}_{Sq}(f) \in L^2(T) = \mathbb{R}^{|T|}}_{\text{output}},$$

with $T = T_{n-1}$. For the time being we consider the space of neural responses to be L^2, however more generally one could consider L^p spaces in order to, for example, promote sparsity in the obtained representation. The coordinates of the output are simply the normalized neural responses $\widehat{N}(f)(t)$ of f at each given t in the template set T and have a natural interpretation as the outputs of neurons responding to specific patterns. Clearly,

$$\widehat{K}(f,g) = \left\langle \widehat{N}(f), \widehat{N}(g) \right\rangle_{L^2(T)}. \tag{2.7}$$

A map satisfying the above condition is referred to as a *feature map* in the language of kernel methods [93]. A natural distance d between two input functions f, g is also defined in terms of the Euclidean distance between the corresponding normalized neural responses:

$$d^2(f,g) = \|\widehat{N}(f) - \widehat{N}(g)\|^2_{L^2(T)} = 2\left(1 - \langle \widehat{N}(f), \widehat{N}(g) \rangle_{L^2(T)}\right), \tag{2.8}$$

where we used the fact that the neural responses are normalized. Note that the above distance function is a restatement of (2.6). The following simple properties follow:

- If $\widehat{K}(f,g) = 1$, then $\widehat{N}(f) = \widehat{N}(g)$ as can be easily shown using (2.7) and (2.8).
- If $\widehat{K}(f,g) = 1$, then for all z, $\widehat{K}(f,z) = \widehat{K}(g,z)$, as shown by the previous property and the fact that $\langle \widehat{N}(f), \widehat{N}(z) \rangle_{L^2(T)} = \langle \widehat{N}(g), \widehat{N}(z) \rangle_{L^2(T)}$.

The neural response at a given layer can be expressed in terms of the neural responses at the previous layer via the following coordinate-wise definition:

$$N_{Sq}(f)(t) = \max_{h \in H} \Big\langle \widehat{N}_v(f \circ h), \widehat{N}_v(t) \Big\rangle_{L^2(T')}, \quad t \in T$$

with $H = H_v$, $T' = T_u$ and $T = T_v$. Similarly, we can rewrite the above definition using the more compact notation

$$N_{Sq}(f) = \max_{h \in H} \Big\{ \Pi_v \widehat{N}_v(f \circ h) \Big\}, \tag{2.9}$$

where the max operation is assumed to apply component-wise, and we have introduced the operator $\Pi_v : L^2(T_u) \to L^2(T_v)$ defined by

$$(\Pi_v F)(t) = \Big\langle \widehat{N}_v(t), F \Big\rangle_{L^2(T_u)}$$

for $F \in L^2(T_u), t \in T_v$. The above reasoning can be generalized to any layer in any given architecture so that we can always give a self consistent, recursive definition of normalized neural responses. From a computational standpoint it is useful to note that the operator Π_v can be seen as a $|T_v| \times |T_u|$ matrix so that each step in the recursion amounts to matrix-vector multiplications followed by max operations. Each row of the matrix Π_v is the (normalized) neural response of a template $t \in T_v$, so that an individual entry of the matrix is then

$$(\Pi_v)_{t,t'} = \widehat{N}_v(t)(t')$$

with $t \in T_v$ and $t' \in T_u$. This perspective highlights the following key points:

- The Π matrices do not depend on the input, and are purely unsupervised components.

- The compact description shown in Equation (2.9) makes clear the action of the hierarchy as alternating pooling and filtering steps realized by the max and Π operators, respectively. The max enforces invariance, while the Π operators incorporate unsupervised learning.

- This perspective also identifies a clear way to integrate more sophisticated unsupervised learning techniques. By choosing an alternate basis on which we express the action of Π, one can immediately apply a range of established techniques in the context of a multiscale, hierarchical architecture. We discuss this and related ideas in more detail in the next Section.

2.2.5 Extensions: Feedback, Generalized Pooling, and Feature Maps

The basic development described in the previous sections can be extended in several ways. In particular one can consider more general pooling functions and more sophisticated procedures for defining the feature maps at each layer. A generalized architecture involving arbitrary feature maps at each layer can be described by the following diagram. Examples of different feature maps Φ and pooling operations are given below.

$$\begin{tikzcd}
\hat{K}_1 \arrow[d, "\Phi_1"'] & \hat{K}_2 \arrow[d, "\Phi_2"'] & \ldots & \hat{K}_{n-1} \arrow[d, "\Phi_{n-1}"'] & \hat{K}_n \\
\widetilde{K}_1 \arrow[ur, "\text{pool}"] & \widetilde{K}_2 \arrow[ur, "\text{pool}"] & \ldots & \widetilde{K}_{n-1} \arrow[ur, "\text{pool}"] &
\end{tikzcd}$$

Figure 2-3: General feature maps can be defined at each layer.

For a generalized architecture as shown in Figure 2-3, compact recursive definitions for the "generalized" neural response and derived kernel can be given.

Definition 2.2.3 (Generalized Derived Kernels)**.** Given the feature maps $\Phi_m : L^2(T_{m-1}) \rightarrow \mathcal{F}_m$, $2 \leq m \leq n$, and a non-negative valued, normalized, initial reproducing kernel $\widetilde{K}_1$, the m-layer derived kernel $\widehat{K}_m$, for $m = 2, \ldots, n$, is obtained by normalizing

$$K_m(f, g) := \left\langle \widetilde{N}_m(f), \widetilde{N}_m(g) \right\rangle_{L^2(T_{m-1})},$$

and the m-layer generalized derived kernel $\widetilde{K}_m$, for $m = 2, \ldots, n$, is given by

$$\widetilde{K}_m(f, g) := \frac{\left\langle (\Phi_m \circ \widetilde{N}_m)(f), (\Phi_m \circ \widetilde{N}_m)(g) \right\rangle_{\mathcal{F}_m}}{\left\| (\Phi_m \circ \widetilde{N}_m)(f) \right\|_{\mathcal{F}_m} \left\| (\Phi_m \circ \widetilde{N}_m)(g) \right\|_{\mathcal{F}_m}}$$

where

$$\widetilde{N}_m(f)(t) = \max_{h \in H_{m-1}} \widetilde{K}_{m-1}(f \circ h, t) \tag{2.10}$$

for $f \in \mathrm{Im}(v_m)$, $t \in T_{m-1}$.

Attention and Top-down Contextual Priors

As above, consider the case where the transformations spaces H_i are endowed with a probability measure ρ_{H_i}. The measures ρ_{H_i} mark a natural entry point for incorporating notions of attention and contextual priors. Such a measure can be used to force the template sampling mechanism to explore particular regions of an image. For example, we might take ρ_{H_i} so that the center of the patch is most likely to be sampled, analogous to foveation in vision. The measure ρ_H may also be used to incorporate attention like mechanisms, whereby prior information guides the search to interesting parts of an image.

We may consider a process whereby an initial, coarse classification pass decides where to look in more detail in an image, and adjusts what we are looking for. Refining the search for an object of interest could be accomplished by adjusting ρ_H and by adjusting ρ_v, the measure on images. By biasing the template dictionaries towards more specific categories, one can expect that better detection rates might be realized.

General Pooling functions

The pooling operation in the definition of the neural response can be generalized by considering different (positive) functionals acting on functions on H. Indeed for a fixed function $f \in \mathrm{Im}(v_m)$ and a template $t \in T_{m-1}$ we can consider the following positive real valued function on $H = H_{m-1}$ defined by

$$F(h) = F_{f,t} = \widehat{K}_{m-1}(f \circ h, t).$$

If $\widehat{K}$ and hence F are sufficiently regular and in particular $F \in L^p(H, \rho_H)$, different pooling operations can be defined by considering L^p norms of F.

The original definition of the neural response simply uses the uniform norm in $L^\infty(H)$, $\|F\|_\infty = \sup_{h \in H} F(h)$, where the supremum is always achieved if H is finite. Another natural choice is the L^1 norm, $\int_H F(h) d\rho_H(h)$, which corresponds to an average. More generally one can consider $\|F\|_p = \left(\int_H F(h)^p d\rho_H(h)\right)^{1/p}$. The neural response for an arbitrary pooling function is then given by

$$N(f)(t) = \Psi(F), \quad \text{with} \quad F(h) = \widehat{K}_{m-1}(f \circ h, t)$$

where $F : H \to \mathbb{R}_+$. We will show in Chapter 4 that some of the analysis in following sections will extend to this more general setting, and *does not depend on the pooling function.*

Learning Templates

The definition of the derived kernel can be modified to define alternative feature maps while preserving the same architecture. See Figure 2-3. We illustrate the case of a canonical feature map associated to the Mercer expansion of the derived kernel at one layer.

Recall from Section 2.2.3 that each space $\mathrm{Im}(v)$ is endowed with a probability measure ρ_v. At any layer we can consider the integral operator $L_{\widehat{K}_v} : L^2(\mathrm{Im}(v), \rho_v) \to$

$L^2(\mathrm{Im}(v), \rho_v)$ defined by

$$L_{\widehat{K}_v} F(f) = \int_{\mathrm{Im}(v)} \widehat{K}_v(f,g) F(g) d\rho_v(g)$$

whose spectrum we denote by $(\sigma_j, \phi_j)_{j=1}^p$, with $p \leq \infty$. In practice, the measure ρ_v can be replaced by the empirical measure underlying the corresponding template set T. We can then consider the map $\Phi : \mathrm{Im}(v) \to \ell^2$ such that

$$f \mapsto \Phi(f) = \left(\sqrt{\sigma_1}\phi_1(f), \sqrt{\sigma_1}\phi_2(f), \ldots, \sqrt{\sigma_N}\phi_N(f) \right),$$

where $N < p$. Recalling the Mercer expansion of a kernel, one can see that the above feature map corresponds to replacing the derived kernel with a truncated derived kernel

$$\widehat{K}(f,g) = \sum_{j=1}^{p} \sigma_j \phi_j(f) \phi_j(g) \quad \mapsto \quad \tilde{K}(f,g) = \sum_{j=1}^{N} \sigma_j \phi_j(f) \phi_j(g). \tag{2.11}$$

We evaluate this particular feature map experimentally in Section 2.6 below.

It is important to recognize that the above reasoning can be extended to many other feature maps, incorporating, for example, geometric information (manifold learning) or sparsity constraints (sparse coding).

Truncation Error Estimates

Finally, we note that, for any bounded, measurable reproducing kernel $K : X \times X \to \mathbb{R}$, the spectrum of L_K must be summable: $\sum_{j=1}^{\infty} \sigma_j < \infty$. Let K_N denote the the N-term truncated expansion of K, as in Equation (2.11). One may make a crude but informative estimate of the truncation error

$$\|L_K - L_{K_N}\|$$

by observing that the sequence $\{\sigma_j\}_j$ must decay faster than j^{-1}. Indeed,

$$\begin{aligned}
\|L_K - L_{K_N}\| &= \sup_{\|F\|_{L^2(\mathrm{Im}(v),\rho_v)}=1} \sqrt{\sum_{j=N+1}^{\infty} \sigma_j^2 |\langle F, \phi_j \rangle_K|^2} \\
&\leq \mathrm{Tr}[L_K - L_{K_N}] \\
&= \sum_{j=N+1}^{\infty} \sigma_j \leq \sum_{j=N+1}^{\infty} j^{-(1+\varepsilon)}
\end{aligned}$$

for some small $\varepsilon > 0$. Thus for even small choices of N, the truncation error will not be large.

2.3 Invariance of the Neural Response

In this section we discuss *invariance* of the (normalized) neural response to some set of transformations $\mathcal{R} = \{r \mid r : v \to v\}$, where invariance is defined as $\widehat{N}(f) = \widehat{N}(f \circ r)$ (or equivalently $\widehat{K}_n(f \circ r, f) = 1$).

We consider a general n-layer architecture and denote by $r \in \mathcal{R}$ the transformations whose domain (and range) are clear from the context. The following important assumption relates the transformations $\mathcal{R}$ and the translations H associated to an arbitrary layer:

Assumption 2.3.1. *Fix any $r \in \mathcal{R}$. Then for each $h \in H$, there exists a unique $h' \in H$ such that the relation*

$$r \circ h = h' \circ r \tag{2.12}$$

holds true, and the map $h \mapsto h'$ is surjective.

Note that r on the left hand side of Equation (2.12) maps v_{m+1} to itself, while on the right hand side r maps v_m to itself.

In the case of vision for example, we can think of $\mathcal{R}$ as reflections and H as translations so that $f \circ h$ is an image patch obtained by restricting an image f to a receptive field. The assumption says that reflecting an image and then taking a restriction is equivalent to first taking a (different) restriction and then reflecting the resulting image patch. In this section we give examples where the assumption holds. Examples in the case of strings are given in the next section.

Given the above assumption for all layers, we can state the following result.

Proposition 2.3.1. *If the initial kernel satisfies $\widehat{K}_1(f, f \circ r) = 1$ for all $r \in \mathcal{R}$, $f \in \mathrm{Im}(v_1)$, then*

$$\widehat{N}_m(f) = \widehat{N}_m(f \circ r),$$

for all $r \in \mathcal{R}$, $f \in \mathrm{Im}(v_m)$ and $m \leq n$.

Proof. We proceed by induction. The base case is true by assumption. The inductive hypothesis is that $\widehat{K}_{m-1}(u, u \circ r) = 1$ for any $u \in \mathrm{Im}(v_{m-1})$. Thus for all $t \in T = T_{m-1}$ and for $H = H_{m-1}$, we have that

$$\begin{aligned} N_m(f \circ r)(t) &= \max_{h \in H} \widehat{K}_{m-1}(f \circ r \circ h, t) \\ &= \max_{h' \in H} \widehat{K}_{m-1}(f \circ h' \circ r, t) \\ &= \max_{h' \in H} \widehat{K}_{m-1}(f \circ h', t) \\ &= N_m(f)(t), \end{aligned}$$

where the second equality follows from Assumption 2.3.1 and the third follows from the inductive hypothesis. □

The following result is then immediate:

Corollary 2.3.1. *Let $\mathcal{Q}$, $\mathcal{U}$ be two families of transformations satisfying Assumption 2.3.1 and such that $\widehat{K}_1$ is invariant to $\mathcal{Q}, \mathcal{U}$. If $\mathcal{R} = \{r = q \circ u \mid q \in \mathcal{Q}, u \in \mathcal{U}\}$, then*

$$\widehat{N}_m(f) = \widehat{N}_m(f \circ r)$$

for all $r \in \mathcal{R}$, $f \in \mathrm{Im}(v_m)$ and $m \leq n$.

Proof. The proof follows noting that for all $m \leq n$,

$$\widehat{N}_m(f \circ r) = \widehat{N}_m(f \circ q \circ u) = \widehat{N}_m(f \circ q) = \widehat{N}_m(f).$$

□

2.3.1 Invariance in Generalized Architectures

We note that Proposition 2.3.1 is quite general, and extends in two important ways.

Invariance and General Pooling. As we will show in Section 4.2.1 of Chapter 4, *Proposition 2.3.1 holds generally for arbitrary pooling functions*, and does not depend on the particular choice of the max.

Invariance and Arbitrary Feature Maps. If $\widehat{K}_m$ is replaced by $\widetilde{K}_m$ by choosing a suitable feature map $\Phi_m : L^2(T_{m-1}) \to \mathcal{F}_m$, as described in Section 2.2.5 above, then Proposition 2.3.1 still holds as long as $\widetilde{K}_1(f, f \circ r) = 1$. The following Corollary shows that Proposition 2.3.1 holds for generalized architectures as described in Section 2.2.5.

Corollary 2.3.2. *If for all $r \in \mathcal{R}$, $f \in \mathrm{Im}(v_1)$, the initial kernel $\widetilde{K}_1(f, f \circ r) = 1$, then*

$$\widetilde{N}_m(f) = \widetilde{N}_m(f \circ r),$$

for all $r \in \mathcal{R}$, $f \in \mathrm{Im}(v_m)$ and $m \leq n$.

Proof. In the proof of Proposition 2.3.1, we used the obvious fact that invariance of N_m implies invariance of the kernel $\widehat{K}_m(f, g)$. Here, it remains to be shown that invariance of $\widetilde{N}_m$ implies invariance of $\widetilde{K}_m$. Then we may use an inductive hypothesis paralleling that of Proposition 2.3.1, and the proof in the present setting is the same as that of Proposition 2.3.1, mutatis mutandis.

As before, the base case is true by assumption, so $\widetilde{N}_2$ is invariant. We would like to modify the inductive hypothesis of Proposition 2.3.1 to say that $\widetilde{K}_{m-1}(f \circ r, f) = 1$. Since $\widetilde{N}_{m-1}$ is invariant, by the inductive hypothesis local to this Corollary,

$$\Phi_{m-1}\Big(\widetilde{N}_{m-1}(f \circ r)\Big) = \Phi_{m-1}\Big(\widetilde{N}_{m-1}(f)\Big),$$

and we see that invariance of $\widetilde{N}_{m-1}$ leads to invariance of $\widetilde{K}_{m-1}$. □

2.3.2 Invariance Example: Reflections and Rotations

We next discuss invariance of the neural response under reflections and rotations. Consider patches which are discs in $\mathbb{R}^2$. Let

$$\mathcal{R}ef = \{\mathrm{ref} = \mathrm{ref}_\theta \mid \theta \in [0, 2\pi)\}$$

be the set of coordinate reflections about lines passing through the origin at angle θ, and let $\mathcal{R}ot$ denote the space of coordinate rotations about the origin. Then the following result holds true.

Corollary 2.3.3. *If the spaces H at all layers contain all possible translations and $\widehat{K}_1(f, f \circ \mathrm{ref}) = 1$, for all $\mathrm{ref} \in \mathcal{R}ef$, $f \in \mathrm{Im}(v_1)$, then*

$$\widehat{N}_m(f) = \widehat{N}_m(f \circ \mathrm{ref}),$$

for all $\mathrm{ref} \in \mathcal{R}ef$, $f \in \mathrm{Im}(v_m)$ with $m \leq n$. Moreover under the same assumptions

$$\widehat{N}_m(f) = \widehat{N}_m(f \circ \mathrm{rot}),$$

for all $\mathrm{rot} \in \mathcal{R}ot$, $f \in \mathrm{Im}(v_m)$ with $m \leq n$.

Proof. We first show that Assumption 2.3.1 holds. Each translation is simply $h_a(x) = x + a$, and since the space of transformations contains all translations, Assumption 2.3.1 holds taking $h = h_a$, $r = \mathrm{ref}_\theta$ and $h' = h_{a'}$, with $a' = \mathrm{ref}_\theta(a)$. Since the initial kernel $\widehat{K_1}$ is invariant under reflections, Proposition 2.3.1 implies $\widehat{K}_m(f, f \circ \mathrm{ref}) = 1$ for all $\mathrm{ref} \in \mathcal{R}ef$, $f \in \mathrm{Im}(v_m)$, with $m \leq n$.

Rotational invariance follows recalling that any rotation can be obtained out of two reflections using the formula $\mathrm{rot}(2(\theta - \phi)) = \mathrm{ref}_\theta \circ \mathrm{ref}_\phi$, so that we can apply directly Corollary 2.3.1. □

We add the following remark.

Remark 2.3.1. *Although the above proof assumes all translations for simplicity, the assumption on the spaces H can be relaxed. Defining the circle*

$$\tilde{H}_a = \{h_z \mid z = \mathit{ref}(a),\ \mathit{ref} \in \mathcal{R}ef\},$$

it suffices to assume that,

$$\textit{If } h_a \in H, \quad \textit{then} \quad \tilde{H}_a \subseteq H. \tag{2.13}$$

The next section discusses the case of one dimensional strings.

2.4 Analysis in a One Dimensional Case

We specialize the derived kernel model to a case of one-dimensional strings of length n ("n-strings"). An n-string is a function from an index set $\{1, \ldots, n\}$ to some finite alphabet S. We build a derived kernel in this setting by considering patches that are sets of indices $v_m = \{1, \ldots, \ell_m\}$, $m \leq n$ (with $\ell_m < \ell_{m+1}$) and function spaces $\mathrm{Im}(v_m)$ comprised of functions taking values in S rather than in $[0, 1]$. We always assume that the first layer consists of single characters, $v_1 = S$, and consider the initial kernel

$$\widehat{K}_1(f, g) = \begin{cases} 1 & \text{if } f = g, \\ 0 & \text{otherwise} \end{cases},$$

where $f, g \in S$.

In the following we often consider an *exhaustive* architecture in which patches differ in size by only one character so that $v_m = \{1, \ldots, m\}$, and the function (string) spaces are $\mathrm{Im}(v_m) = S^m$, for $m = 1, \ldots, n$. In this case, the template sets are $T_m = S^m$, for $m = 1, \ldots, n$, and the transformations are taken to be all possible translations. Note that the transformation spaces $H = H_m$ at each layer m, contain only two elements

$$H = \Big\{h_1, h_2\Big\},$$

with $h_1(j) = j$ and $h_2(j) = j+1$. For example, if f is an n-string and $H = H_{n-1}$, then $f \circ h_1$ and $f \circ h_2$ are the substrings obtained from the first and last $n-1$ characters in f, respectively. Thus, the n-layer neural response of f at some $n-1$-string t is simply

$$N_n(f)(t) = \max\Big\{\widehat{K}_{n-1}(f \circ h_1, t), \widehat{K}_{n-1}(f \circ h_2, t)\Big\}.$$

We now introduce a few additional definitions useful for discussing and manipulating strings.

Definition 2.4.1 (Reversal)**.** The reversal r of patches of size $m \leq n$ is given by

$$r(j) = m - j + 1, \quad j = 1, \ldots, m.$$

In the development that follows, we adopt the notation $f \sim g$, if $f = g$ or $f = g \circ r$.

Finally, we introduce a pair of general concepts not necessarily limited to strings.

Definition 2.4.2 (Occurrence)**.** Let $f \in \mathrm{Im}(Sq)$. We say that $t \in \mathrm{Im}(v_{n-1})$ *occurs* in f if

$$N_n(f)(t) = 1.$$

where $H = H_{n-1}$.

Note that the above definition naturally extends to any layer m in the architecture, replacing Sq with v_m and v_{n-1} with v_{m-1}.

Definition 2.4.3 (Distinguishing Template)**.** Let $f, g \in \operatorname{Im}(Sq)$ and $t \in \operatorname{Im}(v_{n-1})$. We say that t distinguishes f and g if and only if it occurs in f but not in g, or in g but not in f. We call such a t a distinguishing template for f and g.

In the next subsection we discuss properties of the derived kernel in the context of strings.

2.4.1 Discrimination Properties

We begin by considering an architecture of patches of arbitrary size and show that the neural response is invariant to reversal. We then present a result describing discrimination properties of the derived kernel.

Corollary 2.4.1. *If the spaces H at all layers contain all possible translations then*

$$\widehat{K}_m(f, f \circ r) = 1,$$

for all $f \in \operatorname{Im}(v_m)$ with $m \leq n$.

Proof. We first show that Assumption 2.3.1 holds. Let $u \subset v$ be any two layers where $\operatorname{Im}(v)$ contains m-strings and $\operatorname{Im}(u)$ contains ℓ-strings, with $\ell < m$. Every translation $h : u \to v$ is given by $h_i : (1, \ldots, \ell) \mapsto (i, \ldots, i + \ell - 1)$, for $1 \leq i \leq m - \ell + 1$. Then Assumption 2.3.1 holds taking $h = h_i$, and $h' = h_{\varphi(i)}$, where $\varphi : (1, \ldots, m - \ell + 1) \to (1, \ldots, m - \ell + 1)$ is defined by $\varphi(i) = m - \ell - i + 2$. Using the fact that the initial kernel is invariant to reversal, Proposition 2.3.1 then ensures that $\widehat{K}_v(f, f \circ r) = 1$. □

The following remark is analogous to Remark 2.3.1.

Remark 2.4.1. *Inspecting the above proof one can see that the assumption on the spaces H can be relaxed. It suffices to assume that*

$$\textit{If } h_i \in H, \quad \textit{then } \; h_{\varphi(i)} \in H. \tag{2.14}$$

with the definition $\varphi(i) = m - \ell - i + 2$.

We now ask whether two strings having the same (normalized) neural response are indeed the same strings up to a reversal and/or a checkerboard pattern for odd length strings. We consider this question in the context of the exhaustive architecture described at the beginning of Section 2.4.

Theorem 2.4.1. *Consider the exhaustive architecture where $v_m = \{1, \ldots, m\}$, the template sets are $T_m = \operatorname{Im}(v_m) = S^m$, for $m = 1, \ldots, n$ and the transformations are all possible translations. If f, g are n-strings and $\widehat{K}_n(f, g) = 1$ then $f \sim g$ or f, g are the "checkerboard" pattern: $f = ababa\cdots, g = babab\cdots$, with f and g odd length strings, and a, b arbitrary but distinct characters in the alphabet.*

The theorem has the following interpretation: the derived kernel is discriminating if enough layers and enough templates are assumed. In a more general architecture, however, we might expect to have larger classes of patterns mapping to the same neural response.

To prove the above theorem, we make use of the following preliminary but important result.

Lemma 2.4.1. *Let $f, g \in \mathrm{Im}(v_m)$ with $m \leq n$. If $\widehat{K}_m(f,g) = 1$, then all function patches of f at layer $m-1$ occur in g and vice versa.*

Proof. We prove the Lemma assuming that a function patch $\bar{t}$ of f distinguishes f from g, and then showing that under this assumption $\widehat{K}_m(f,g)$ cannot equal 1.

Since $\bar{t}$ occurs in f but *does not occur* in g, by Definition 2.4.2,

$$N_m(g)(\bar{t}) < 1 \quad \text{and} \quad N_m(f)(\bar{t}) = 1. \tag{2.15}$$

Now, let t' be any function subpatch of g at layer $m-1$, then

$$N_m(g)(t') = 1 \quad \text{and} \quad N_m(f)(t') \leq 1, \tag{2.16}$$

where the last inequality follows since t' might or might not occur in f.
Now since $\widehat{K}_m(f,g) = 1$ and recalling that by definition $\widehat{K}_m$ is obtained normalizing $K_m(f,g) = \big\langle N_m(f), N_m(g) \big\rangle_{L^2(T_{m-1})}$, we have that $N_m(f), N_m(g)$ must be collinear. That is,

$$N_m(f)(t) = c \cdot N_m(g)(t), \quad t \in T_{m-1} \tag{2.17}$$

for some constant c.
Combining this requirement with conditions (2.15),(2.16) we find that

$$\begin{aligned} N_m(f)(\bar{t}) = cN_m(g)(\bar{t}) \quad &\Rightarrow \quad c > 1 \\ N_m(f)(t') = cN_m(g)(t') \quad &\Rightarrow \quad c \leq 1. \end{aligned}$$

Thus, there is no such c and $\widehat{K}_m(f,g)$ cannot equal 1. Similarly, by interchanging the roles of f and g above we reach the conclusion that if there is a function patch in g which does not *occur* in f, then $\widehat{K}_m(f,g)$ again cannot equal 1. □

We can now prove Theorem 2.4.1 by induction.

Proof. The statement holds trivially for $\widehat{K}_1$ by definition. The remainder of the proof is divided into three steps.

Step 1). We first note that since $\widehat{K}_n(f,g) = 1$ then Lemma 2.4.1 says that both $n-1$ strings in f occur in g and vice versa. Denoting with s_1 (s_2) the first (second) $n-1$-substring in an n-string s, we can express this as

$$\widehat{K}_{n-1}(f_1, g_1) = 1 \quad \textit{or} \quad \widehat{K}_{n-1}(f_1, g_2) = 1$$

and

$$\widehat{K}_{n-1}(f_2, g_1) = 1 \quad or \quad \widehat{K}_{n-1}(f_2, g_2) = 1,$$

and another set of similar conditions interchanging f and g. When u, v are odd-length strings then we write $u \bowtie v$ if $u \sim v$ or if u, v are the checkerboard pattern (*but not both*). When u, v are even-length strings then $u \bowtie v$ is simply $u \sim v$. The inductive hypothesis is that $\widehat{K}_{n-1}(\alpha, \beta) = 1$ implies $\alpha \bowtie \beta$, so that the above conditions translate into a large number of relationships between the substrings in f and g given by combinations of the following 4 predicates:

$$\begin{array}{ll} a) & f_1 \bowtie g_1 \\ b) & f_1 \bowtie g_2 \\ c) & f_2 \bowtie g_1 \\ d) & f_2 \bowtie g_2. \end{array}$$

Step 2). The next step is to show that the number of relationships we need to consider can be drastically reduced. In fact the statement "both $n-1$ strings in f occur in g and vice versa" can be formalized as

$$(a + b + ab)(c + d + cd)(a + c + ac)(b + d + bd), \tag{2.18}$$

denoting logical exclusive OR with a "+" and AND by juxtaposition. The above expression corresponds to a total of 81 possible relationships among the $n-1$-substrings. Any product of conditions involving repeated predicates may be simplified by discarding duplicates. Doing so in the expansion of (2.18), we are left with only *seven* distinct cases:

$$\{abcd, abc, abd, acd, ad, bc, bcd\}.$$

We claim that, for products involving more than two predicates, considering only two of the conditions will be enough to derive $f \sim g$ or f, g checkerboard. If more than two conditions are present, they only serve to further constrain the structure of the strings or change a checkerboard pattern into a reversal equivalence, but cannot change an equivalence to a non-equivalence or a checkerboard to any other non-equivalent pattern.

Step 3). The final step is to consider the cases ad and bc (since one or the other can be found in each of the 7 cases above) and show that this is in fact sufficient to prove the proposition.

Let $f = a_1 a_2 \cdots a_n$ and $g = b_1 b_2 \cdots b_n$, and denote the checkerboard condition by $f \diamond g$.

Case *ad*: $f_1 \bowtie g_1 \wedge f_2 \bowtie g_2$

There are nine subcases to consider,

$$(f_1 = g_1 \vee f_1 = r(g_1) \vee f_1 \diamond g_1) \wedge (f_2 = g_2 \vee f_2 = r(g_2) \vee f_2 \diamond g_2)$$

however for n odd the $n-1$ substrings cannot be checkerboard and only the first four cases below are valid.

1. $f_1 = g_1 \wedge f_2 = g_2$: The conditions give immediate equality, $f = g$.

2. $f_1 = g_1 \wedge f_2 = r(g_2)$: The first condition says that the strings are equal everywhere except the last character, while the second says that the last character in f is b_2. So if $b_2 = b_n$, then $f = g$. The conditions taken together also imply that $b_i = b_{n-i+2}, i = 2, \ldots, n-1$ because g_1 overlaps with g_2 by definition. So we indeed have that $b_2 = b_n$, and thus $f = g$.

3. $f_1 = r(g_1) \wedge f_2 = g_2$: Symmetric to the previous case.

4. $f_1 = r(g_1) \wedge f_2 = r(g_2)$: The first condition says that $f = b_{n-1} \cdots b_1 a_n$ and the second gives $f = a_1 b_n \cdots b_2$. Thus we have that $a_1 = b_{n-1}, a_n = b_2$ and $b_i = b_{i+2}$ for $i = 1, \ldots, n-2$. The last relation implies that g has two symbols which alternate. Furthermore, we see that if n is even, then $f = g$. But for n odd, f is a one character circular shift of g, and thus f, g are checkerboard.

5. $f_1 = g_1 \wedge f_2 \diamond g_2$: The checkerboard condition gives that $f = a_1 a_2 a_3 a_2 a_3 \cdots a_2$ and $g = b_1 a_3 a_2 a_3 a_2 \cdots a_3$. Then $f_1 = g_1$ gives that $a_2 = a_3$ and $a_1 = b_1$ so $f = g$.

6. $f_1 = r(g_1) \wedge f_2 \diamond g_2$: The first condition imposes $a_1 = a_2 = a_3$ and $b_1 = a_3$ on the checkerboard structure, giving $f = g$ and both strings comprised of a single repeated character.

7. $f_1 \diamond g_1 \wedge f_2 \diamond g_2$: The first condition imposes $a_1 = a_3$ and $b_1 = a_2$ on the structure given by the second checkerboard condition, thus $f = a_3 a_2 a_3 \cdots a_2$, $g = a_2 a_3 a_2 \cdots a_3$, and $f = r(g)$.

8. $f_1 \diamond g_1 \wedge f_2 = g_2$: Symmetric to the case $f_1 = g_1 \wedge f_2 \diamond g_2$.

9. $f_1 \diamond g_1 \wedge f_2 = r(g_2)$: Symmetric to the case $f_1 = r(g_1) \wedge f_2 \diamond g_2$.

Case *bc*:$f_1 \bowtie g_2 \wedge f_2 \bowtie g_1$
There are again nine subcases to consider:

$$(f_1 = g_2 \vee f_1 = r(g_2) \vee f_1 \diamond g_2) \wedge (f_2 = g_1 \vee f_2 = r(g_1) \vee f_2 \diamond g_1).$$

But suppose for the moment $g' = b_1 \cdots b_n$ and we let $g = r(g') = b_n \cdots b_1$. Then every subcase is the same as one of the subcases considered above for the case ad, only starting with the reversal of string g. For example, $f_1 = g_2$ here means that $f_1 = b_{n-1} \cdots b_1 = r(g'_1)$. When n is even, note that $f_1 \diamond g_2 \Leftrightarrow f_1 \diamond r(g'_1) \Leftrightarrow f_1 \diamond g'_1$, where the last relation follows from the fact that reversal does not effect an odd-length alternating sequence. Returning to the ordering $g = b_1 \cdots b_n$, each subcase here again gives either $f = g, f = r(g)$ or, if n is odd, f, g are possibly checkerboard.

Gathering the case analyses above, we have that $\widehat{K}_m(f,g) = 1 \implies f \sim g$ (m even) or $f \bowtie g$ (m odd). □

2.4.2 Discrimination and Global Pooling Kernels

The following Proposition extends Lemma 2.4.1 to all layers, and will be used to investigate the discrimination properties of the global pooling kernels. In the analysis that follows, we again consider the exhaustive architecture setting.

Proposition 2.4.1. *We consider the exhaustive architecture described in Theorem 2.4.1. Let f, g be n-strings. If $\widehat{K}_n(f,g) = 1$, then all substrings of all lengths in f occur in g, and all substrings of all lengths in g occur in f.*

Proof. The proof is by induction, from the top-down. Every length $n-2$ substring of f may be expressed as

$$s_f^{n-2} = f \circ h_{i_{n-1}}^{n-1} \circ h_{i_{n-2}}^{n-2}$$

for some (possibly non-unique) choice of $i_{n-1}, i_{n-2} \in \{1,2\}$. From Lemma 2.4.1, we know that $\widehat{K}_n(f,g) = 1$ implies the two substrings of length $n-1$ in f occur in g, and vice versa. Thus for every s_f^{n-2}, one can choose a $j_{n-1} \in \{1,2\}$ such that

$$\widehat{K}_{n-1}(f \circ h_{i_{n-1}}^{n-1}, g \circ h_{j_{n-1}}^{n-1}) = 1. \tag{2.19}$$

Setting $f' = f \circ h_{i_{n-1}}^{n-1}$ and $g' = g \circ h_{j_{n-1}}^{n-1}$ and applying Lemma 2.4.1 to (2.19) tells us that all $n-2$ strings in f' occur in g' and vice versa. Therefore, there is a $j_{n-2} \in \{1,2\}$ such that

$$\widehat{K}_{n-2}(s_f^{n-2}, g \circ h_{j_{n-1}}^{n-1} \circ h_{j_{n-2}}^{n-2}) = 1.$$

This proves that if $\widehat{K}_n(f,g) = 1$, every $n-2$ string in f occurs in g. Similarly, interchanging the roles of f and g above shows that every $n-2$ string in g must also *occur* in f.

Proceeding downwards, let every length k substring of f be expressed as $s_f^k = f \circ h_{i_{n-1}}^{n-1} \circ \cdots \circ h_{i_k}^k$, for $k = 1, \ldots, n-3$. Applying the reasoning above inductively, for each substring s_f^k we can choose a set of indices $j_{n-1}, \ldots, j_{k+1}$ such that

$$\widehat{K}_{k+1}(f \circ h_{i_{n-1}}^{n-1} \circ \cdots \circ h_{i_{k+1}}^{k+1}, g \circ h_{j_{n-1}}^{n-1} \circ \cdots \circ h_{j_{k+1}}^{k+1}) = 1$$

at which point Lemma 2.4.1 tells us that there is a j_k such that

$$\widehat{K}_k(s_f^k, g \circ h_{j_{n-1}}^{n-1} \circ \cdots \circ h_{j_k}^k) = 1.$$

Thus all substrings of all lengths in f occur in g, and interchanging the roles of f and g above, all substrings of all lengths in g occur in f. □

Finally, we evaluate the discrimination properties of the global pooling kernels defined in Definition 2.2.2. In particular, the following Proposition reveals that if an exhaustive architecture derived kernel claims that two strings are the same, then one can replace any upper layer with a global pooling layer, and find that the hierarchy would still view the two strings as the same.

Proposition 2.4.2. *If $\widehat{K}_n(f,g) = 1$, then $\widehat{G}_m(f,g) = 1$ for $1 < m \leq n$.*

Proof. From Proposition 2.4.1, $\widehat{K}_n(f,g) = 1$ implies that all substrings in f *occur* in g and vice versa. Thus for every substring $f \circ h^{m-1}$ (with $h^{m-1} \in H^G_{m-1}$) appearing in the definition of G_m, there is an $h' \in H^G_{m-1}$ such that $\widehat{K}_{m-1}(f \circ h^{m-1}, g \circ h') = 1$. Similarly, for every substring $g \circ h^{m-1}$ appearing in the definition of G_m, there is an $h'' \in H^G_{m-1}$ such that $\widehat{K}_{m-1}(f \circ h'', g \circ h^{m-1}) = 1$. So we must have that for every $t \in T_{m-1}$,

$$\max_{h^{m-1} \in H^G_{m-1}} \widehat{K}_{m-1}(f \circ h^{m-1}, t) = \max_{h^{m-1} \in H^G_{m-1}} \widehat{K}_{m-1}(g \circ h^{m-1}, t). \qquad (2.20)$$

Here every H is finite, and one can see that the "arguments" to each of the max operations are the same, only in a possibly different order. Since the max is invariant to permutations of its arguments, the two quantities above are equal. With the equality (2.20) holding for all $t \in T_{m-1}$, we necessarily have that $G_m(f,g) = 1$ for $m = 1, \ldots, n-1$. □

2.5 Entropy of the Neural response

We suggest that the concept of Shannon entropy [27] can provide a systematic way to assess the discrimination properties of the neural response, quantifying the role played by the number of layers (or the number of templates).[1] This motivates introducing a few definitions, and recalling some elementary facts from information theory. We sketch here the main ideas, but defer a more elaborated treatment to Chapter 3.

Consider any two layers corresponding to patches $u \subset v$. The space of functions $\mathrm{Im}(v)$ is assumed to be a probability space with measure ρ_v. The neural response is then a map $\widehat{N}_v : \mathrm{Im}(v) \to L^2(T) = \mathbb{R}^{|T|}$ with $T = T_u$. Let us think of $\widehat{N}_v$ as a random variable and assume that

$$\mathbb{E}\left[\widehat{N}_v(f)(t)\right] = 0$$

for all $t \in T_u$ (or perhaps better, set the median to be zero). Next, consider the set $\mathcal{O}$ of orthants in $\mathbb{R}^{|T|}$. Each orthant is identified by a sequence $o = (\epsilon_i)_{i=1}^{|T|}$ with $\epsilon_i = \pm 1$ for all i. We define the map $\widehat{N}^*_v : \mathrm{Im}(v) \to \mathcal{O}$ by

$$\widehat{N}^*_v(f) = \left(\mathrm{sign}(\widehat{N}_v(f)(t))\right)_{t \in T_u}$$

[1] Conversations with David McAllester and Greg Shakhnarovich were useful for this section.

and denote by $\widehat{N}_v^{**}\rho_v$ the corresponding push-forward measure on $\mathcal{O}$. Although replacing the neural response with its signs destroys information, such relaxations can give insights by simplifying a complex situation.

We next introduce the Shannon entropies relative to the measures ρ_v and $\widehat{N}_v^{**}\rho_v$. If we assume the space of images to be finite $\mathrm{Im}(v) = \{f_1, \ldots, f_p\}$, the measure ρ_v reduces to the probability mass function $\{p_1, \ldots, p_d\} = \{\rho_v(f_1), \ldots, \rho_v(f_d)\}$. In this case the entropy of the measure ρ_v is

$$S(\rho_v) = \sum_i p_i \log \frac{1}{p_i}$$

and similarly

$$S(\widehat{N}_v^{**}\rho_v) = \sum_{o \in \mathcal{O}} q_o \log \frac{1}{q_o} \ ,$$

where $q_o = (\widehat{N}_v^{**}\rho_v)(o)$ is explicitly given by

$$(\widehat{N}_v^{**}\rho_v)(o) = \rho_v\left(\left\{f \in \mathrm{Im}(v) \mid \left(\mathrm{sign}(\widehat{N}_v(f)(t))\right)_{t \in T_u} = o\right\}\right).$$

When $\mathrm{Im}(v)$ is not finite we define the entropy $S(\rho_v)$ by considering a partition $\pi = \{\pi_i\}_i$ of $\mathrm{Im}(v)$ into measurable subsets. In this case the entropy of ρ_v (given the partition π) is

$$S_\pi(\rho_v) = \sum_i \rho_v(\pi_i) \log \frac{1}{\rho_v(\pi_i)}.$$

One can define the entropy of the original neural response (without thresholding) in a similar fashion, taking a partition of the range of $\widehat{N}_v$.

Comparing $S(\rho_v)$ to $S(\widehat{N}_v^{**}\rho_v)$, we can assess the discriminative power of the neural response and quantify the amount of information about the function space that is retained by the neural response. The following inequality, related to the so called *data processing inequality*, serves as a useful starting point:

$$S(\rho_v) \geq S(\widehat{N}_v^{**}\rho_v).$$

It is then interesting to quantify the *discrepancy*

$$S(\rho_v) - S(\widehat{N}_v^{**}\rho_v),$$

which is the loss of information induced by the neural response. Since the inequality holds with equality when the map $\widehat{N}_v^{*}$ is one-to-one, this question is related to asking whether the neural response is injective (see Theorem 2.4.1).

2.5.1 Short Appendix to Section 2.5

We briefly discuss how the development in the previous section relates to standard concepts (and notation) found in information theory [27]. Let (Ω, P) be a probability space and X a measurable map into some measurable space $\mathcal{X}$. Denote by $\rho = X^*(P)$ the push-forward measure on $\mathcal{X}$ associated to X. We consider discrete random variables, i.e. $\mathcal{X} = \{x_1, \dots, x_d\}$ is a finite set. In this case the push-forward measure reduces to the probability mass function over the elements in $\mathcal{X}$ and we let $\{p_1, \dots, p_d\} = \{\rho(x_1), \dots, \rho(x_d)\}$. Then the entropy H of X is defined as

$$H(X) = \sum_{i=1}^{d} p_i \log \frac{1}{p_i} .$$

Connections with the previous section are readily established when $\mathrm{Im}(v)$ is a finite set. In this case we can define a (discrete) random variable $X = F$ with values in $\mathcal{X} = \mathrm{Im}(v) = \{f_1, \dots, f_d\}$ and domain in some probability space (Ω, P) such that P is the pullback measure associated to the measure ρ_v on $\mathrm{Im}(v)$. Then $\{p_1, \dots, p_d\} = \{\rho_v(f_1), \dots, \rho_v(f_d)\}$, and

$$S(\rho_v) \equiv H(F).$$

Moreover we can consider a second random variable Y defined as $N_v^* \circ F$ so that

$$S(N_v^{**}\rho_v) \equiv H(N_v^* \circ F).$$

2.6 Empirical Analysis

The work described thus far was largely motivated by a desire to understand the empirical success of the model in [98, 96] when applied to numerous real-world recognition problems. The simplified setting we consider in this work trades complexity and faithfulness to biology for a more controlled, analytically tractable framework. It is therefore important to verify empirically that we have kept what might have been responsible for the success of the model in [98, 96], and this is the central goal of the current section. We first describe an efficient algorithm for computing the neural response, followed by a set of empirical experiments in which we apply the derived kernel to a handwritten digit classification task.

2.6.1 Algorithm and Computational Complexity

A direct implementation of the architecture following the recursive definition of the derived kernel leads to an algorithm that appears to be exponential in the number of layers. However, a "bottom-up" algorithm which is linear in the number of layers can be obtained by consolidating and reordering the computations.

```
Input:$f \in \mathrm{Im}(Sq), \widehat{N}_m(t), \forall t \in T_m, 1 \leq m \leq n-1$
Output: $\widehat{N}_n(f)(t)$
for $m = 1$ to $n-1$ do
    for $h \in H_m^g$ do
        for $t \in T_m$ do
            if $m = 1$ then
                $S_m(h,t) = \widehat{K}_1(f \circ h, t)$
            else
                $S_m(h,t) = \sum_{t' \in T_{m-1}} \widehat{C}_{m-1}(h,t')\widehat{N}_m(t)(t')$
            end
        end
    end
    for $h \in H_{m+1}^g$ do
        for $t \in T_m$ do
            $C_m(h,t) = \max_{h' \in H_m} S_m(h \circ h', t)$
        end
    end
    $\widehat{C}_m = \mathrm{NORMALIZE}(C_m)$
end
Return $\widehat{N}_n(f)(t) = \widehat{C}_{n-1}(h,t)$, with $h \in H_n^g$, $t \in T_{n-1}$
```

Algorithm 1: Neural response algorithm.

Consider a set of *global* transformations, where the range is always the entire image domain $v_n = Sq$ rather than the next larger patch. We define such global transformations recursively, setting

$$H_m^g = \{h : v_m \to Sq \mid h = h' \circ h'', \text{ with } h' \in H_{m+1}^g, h'' \in H_m\},$$

for any $1 \leq m \leq n-1$ where H_n^g contains only the identity $\{I : Sq \to Sq\}$.

If we assume the neural responses of the templates are pre-computed, then the procedure computing the neural response of any given image $f \in \mathrm{Im}(Sq)$ is given by Algorithm 1. Note that in the Algorithm $C_m(h,t)$ corresponds to the neural response $N_{m+1}(f \circ h)(t)$, with $h \in H_{m+1}^g$, $t \in T_m$. The sub-routine NORMALIZE simply returns the normalized neural response of f.

We estimate the computational cost of the algorithm. Ignoring the cost of normalization and of pre-computing the neural responses of the templates, the number of required operations is given by

$$\tau = \sum_{m=1}^{n-1} \Big(|H_m^g||T_m||T_{m-1}| + |H_{m+1}^g||H_m||T_m| \Big) \tag{2.21}$$

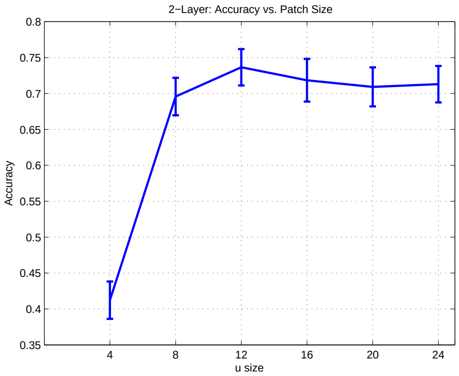


Figure 2-4: A curve determining the optimal u size (2-Layer architecture).

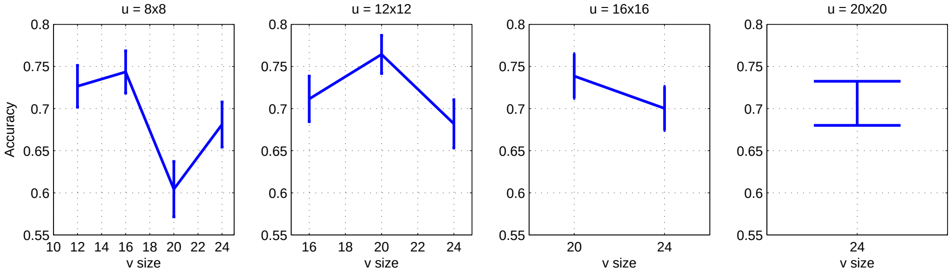


Figure 2-5: Curves determining the optimal v size given various u sizes (3-Layer architecture).

where we denote for notational convenience the cost of computing the initial kernel by $|T_0|$. The above equation shows that the algorithm is linear in the number of layers.

2.6.2 Experiments

In this section we discuss simulations in which derived kernels are compared to an L^2 pixel distance baseline in the context of a handwritten digit classification task. Given a small labeled set of images, we use the 1-nearest neighbor (1-NN) classification rule: an unlabeled test example is given the label of the closest training example under the specified distance.

An outline of this section is as follows: We compare a 3-layer architecture to a 2-layer architecture over a range of choices for the patch sizes u and v, and see that for the digit recognition task, there is an optimal architecture. We show that three layers can be better than two layers, and that both architectures improve upon the L^2 baseline. We then illustrate the behavior of the 3-layer derived kernel as compared to the baseline by presenting matrices of pairwise derived distances (as defined in Equation (2.6)) and pairwise L^2 distances. The block structure that typifies these

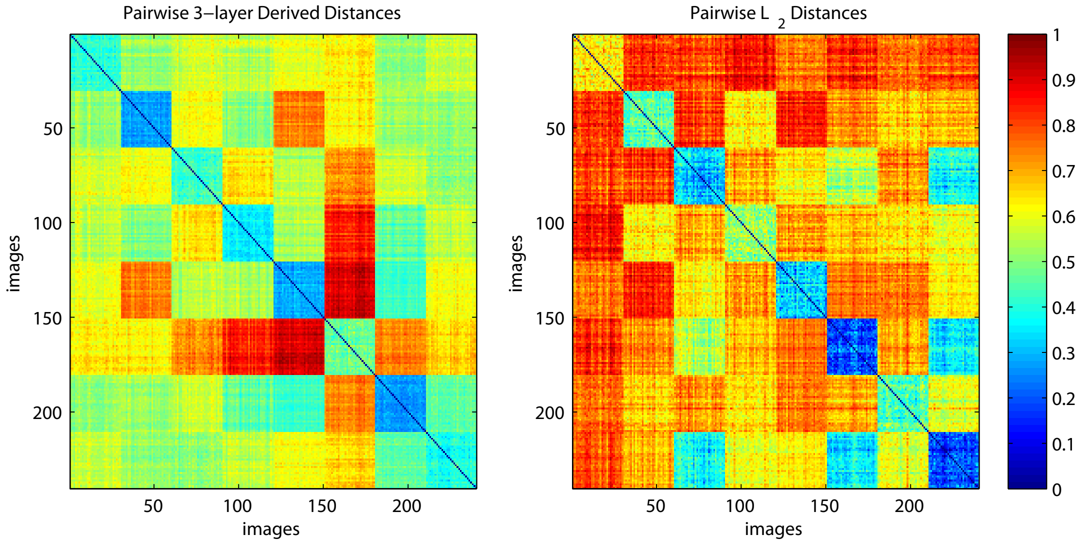


Figure 2-6: Matrices of pairwise 3-Layer derived distances (left) and L^2 distances (right) for the set of 240 images from the database. Each group of 30 rows/columns correspond to images of the digits 2 through 9, in left-right and top-bottom order.

matrices argues graphically that the derived kernels are separating the different classes of images. We next impose a range of artificial translations on the sets of train and test images and find that the derived kernels are robust to large translations while the L^2 distance deteriorates rapidly with even small translations. A set of experiments are then presented in which we study the empirical sample complexity of the digit classification task for different architectures used as an unsupervised pre-processing step. It is found that increasing the number of layers leads to a significantly reduced sample complexity. Finally, we study the effect of taking low-rank approximations to the Π matrices at each layer, as suggested by the development in Section 2.2.5.

In all experiments we have used $Sq = 28 \times 28$ pixel grayscale images randomly selected from the MNIST dataset of handwritten digits [65]. We consider eight classes of images: 2s through 9s. The digits in this dataset include a small amount of natural translation, rotation, scaling, shearing and other deformations – as one might expect to find in a corpus containing the handwriting of human subjects. Our labeled image sets contain 5 examples per class, while the out-of-sample test sets contain 30 examples per class. Classification accuracies using the 1-NN classifier are averaged over 50 random test sets, holding the training and template sets fixed. As in the preceding mathematical analysis, the transformations H are restricted to translations.

The template sets are constructed by randomly extracting 500 image patches (of size u and/or v) from images which are not used in the train or test sets. For the digits dataset, templates of size 10×10 pixels are large enough to include semi-circles and distinct stroke intersections, while larger templates, closer to 20×20, are seen to

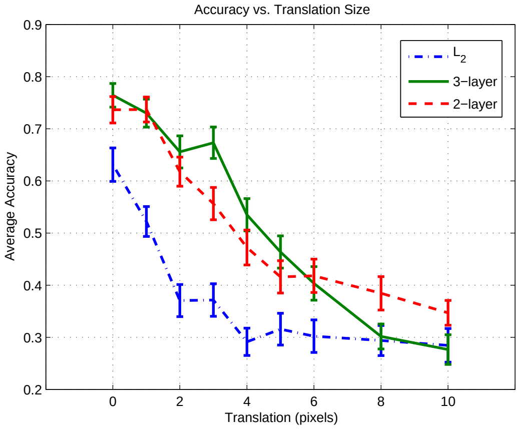


Figure 2-7: Classification accuracy with artificially translated images.

include nearly full digits where more discriminative structure is present.

In Figures 2-4 and 2-5 we show the effect of different patch size selections on classification accuracy. For this particular task, it is clear that the optimal size for patch u is 12×12 pixels for both two and three layer hierarchies. That accuracy levels off for large choices in the case of the 2-layer architecture suggests that the 2-layer derived kernel is approximating a simple local template matching strategy [38]. It is clear, however, from Figure 2-5 that an additional layer can improve on such a strategy, and that further position invariance, in the form of 8 pixels of translation (since $v = 20 \times 20$ and $Sq = 28 \times 28$) at the last stage, can boost performance. In the experiments that follow, we assume architectures that use the best patch sizes as determined by classification accuracy in Figures 2-4 and 2-5: $u = 12 \times 12, v = 20 \times 20$. In practice, the patch size parameters can be chosen via cross validation or on a separate validation set distinct from the test set.

Figure 2-6 illustrates graphically the discrimination ability of the derived kernels when applied to pairs of digits. On the left we show 3-layer derived distances, while the L^2 distances on the raw image intensities are provided for comparison on the right. Both matrices are symmetric. The derived distances are computed from derived kernels using Equation (2.6). Each group of 30 rows/columns correspond to images of the digits 2 through 9, in left-right and top-bottom order. Off diagonal blocks correspond to distances between different classes, while blocks on the diagonal are within-class measurements. In both figures, we have rescaled the range of the original distances to fall in the interval $[0, 1]$ in order to improve contrast and readability. For both distances the ideal pattern corresponds to a block diagonal structure with 30×30 blocks of zeros, and ones everywhere else. Comparing the two matrices, it is clear that the L^2 baseline tends to confuse different classes more often than the 3-layer derived kernel. For example, classes 6 and 8 (corresponding to handwritten 7s and 9s) are frequently confused by the L^2 distance.

Translation Invariance and the Hierarchical Assumption

The experiments discussed up to this point were conducted using a dataset of images that have been registered so that the digits appear approximately in the center of the visual field. Thus the increase in performance when going from 2 to 3 layers validates our assumption that objects particular to the task at hand are hierarchically organized, and can be decomposed into parts and parts of parts, and so on. A second aspect of the neural response architecture that warrants empirical confirmation is that of invariance to transformations accounted for in the hierarchy. In particular, translations.

To further explore the translation invariance of the derived kernel, we subjected the labeled and unlabeled sets of images to translations ranging from 0 to 10 pixels in one of 8 randomly chosen directions. Figure 2-7 gives classification accuracies for each of the image translations in the case of 3- and 2-layer derived kernels as well as for the L^2 baseline. As would be expected, the derived kernels are better able to accommodate image translations than L^2 on the whole, and classification accuracy decays more gracefully in the derived kernel cases as we increase the size of the translation. In addition, the 3-layer derived kernel is seen to generally outperform the 2-layer derived kernel for translations up to approximately 20% of the field of view. For very large translations, however, a single layer remains more robust than the particular 2-layer architecture we have simulated. We suspect that this is because large translations cause portions of the digits to be clipped off the edge of the image, whereas templates used by two-layer architectures describe nearly all regions of a class of digits. Lack of a digit part could thus undermine the descriptive advantage of the 3-layer architecture over the 2-layer hierarchy.

Sample Complexity

In Figure 2-8 we show the results of an experiment in which we have applied two and three layer derived kernels towards solving the same supervised, 8-class handwritten digit classification task described above, with 3-pixels of artificial translation applied to the images. The patch sizes were the optimal settings found above, while the template sets were again of size 500 for all layers. There were 30 random test images per class for scoring accuracy in each of 50 random trials per training set size evaluation. An L_2 baseline is also given for comparison.

The Figure shows the average classification accuracy as the number of labeled training examples is varied for each of the three 1-NN classifiers. Thus, we can choose a given horizontal slice and compare the estimated *sample complexity* of the learning task given each of the models. We find, for example, that in order to obtain 65% accuracy the 2-layer derived kernel based classifier needs about 11 examples per class, while the 3-layer derived kernel based classifier requires only 5 examples. The overall behavior confirms that, (1) the hierarchical assumption holds for this task, and (2) in terms of sample complexity, two layers is better than none, and three is

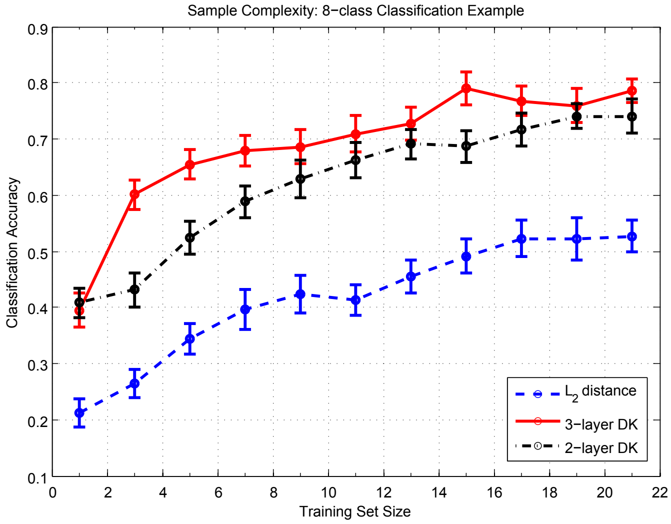


Figure 2-8: Empirical sample complexities of the digit classification task using two and three layer hierarchical models. Training set sizes are the number of labeled examples per class.

better than two. Although the result of this experiment cannot say anything about sample complexity and hierarchical architectures in general, we believe it provides much motivation for studying the issue more closely.

Layer-wise Low-rank Approximations

Lastly, we consider classification experiments in which we adopt low-rank approximations to the integral operators associated to derived kernels at each layer, as described in Section 2.2.5. Figure 2-9 shows the corresponding accuracy for the classification task described above when taking different numbers of components in the approximation at all layers. Although we originally took 500 templates at each layer, the experiments show that only 20-25 components gives similar accuracy as the full rank case. The computational advantage of working with such an approximation is substantial. The fact that so few components suffice also suggests smarter rejection-like sampling techniques for selecting the templates, as well as other possible algorithms for adapting the templates to a specific task in a data-driven fashion.

On the whole the above experiments confirm that the derived kernels are robust to translations, and provide empirical evidence supporting the claim that the neural response includes mechanisms which can exploit the hierarchical structure of the physical world.

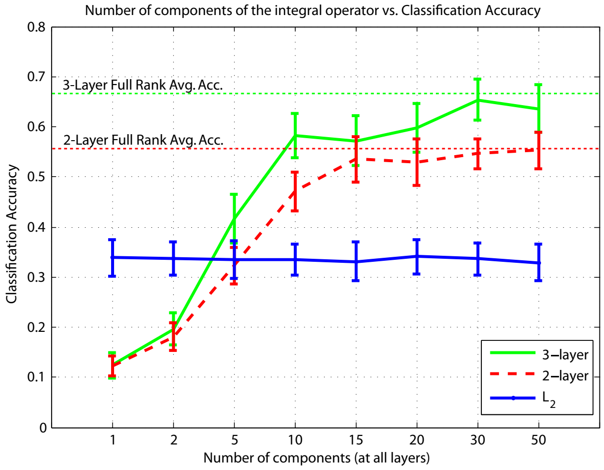


Figure 2-9: Experiments illustrating classification accuracy with low rank approximations to the integral operator associated with the derived kernel.

2.7 Postscript: The Derived Kernel and Visual Cortex

Here we establish an exact connection between the neural response and the model of Serre et al. [96, 98, 94]. We consider an architecture comprised of $S1, C1, S2, C2$ layers as in the model illustrated in Figure 2-10. Consider the patches $u \subset v \subset w \subset Sq$ and corresponding function spaces $\mathrm{Im}(u)$, $\mathrm{Im}(v)$, $\mathrm{Im}(w)$, $\mathrm{Im}(Sq)$ and transformation sets $H_u = H_{u,v}$, $H_v = H_{v,w}$, $H_w = H_{w,Sq}$. In contrast to the development in the previous sections, we here utilize only the template spaces $T_u \subset \mathrm{Im}(u)$ and $T_w \subset \mathrm{Im}(w)$. As will be made clear below, the derived kernel K_v on $\mathrm{Im}(v)$ is extended to a kernel K_w on $\mathrm{Im}(w)$ that eventually defines the next neural response.

S1 and C1 units. Processing steps corresponding to S1 and C1 cells can be defined as follows. Given an initial kernel K_u, let

$$N_{S1}(f \circ h)(t) = K_u(f \circ h, t) \tag{2.22}$$

with $f \in \mathrm{Im}(v)$, $h \in H_u$ and $t \in T_u$. Then $N_{S1}(f \circ h)(t)$ corresponds to the response of an $S1$ cell with template t and receptive field $h \circ u$. The operations underlying the definition of $S1$ can be thought of as "normalized convolutions".

The neural response is given by

$$N_{C1}(f)(t) = \max_{h \in H}\{N_{S1}(f \circ h)(t)\} \tag{2.23}$$

with $f \in \mathrm{Im}(v)$, $H = H_u$ and $t \in T_u$ so that $N_{C1} : \mathrm{Im}(v) \to \mathbb{R}^{|T_u|}$. Then $N_{C1}(f)(t)$ corresponds to the response of a $C1$ cell with template t and receptive field corresponding to v.

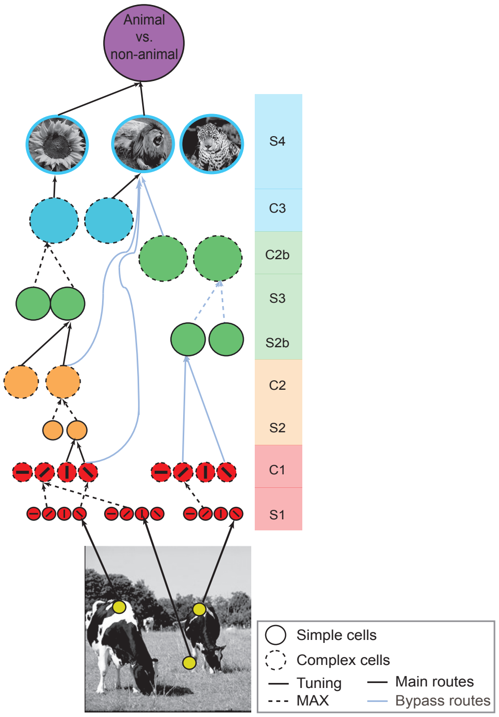


Figure 2-10: The model of Serre et al [98]. We consider here the layers up to $C2$. (Modified from [96]; original provided courtesy T. Serre)

The derived kernel at layer v is defined as usual as

$$K_v(f, g) = \langle N_{C1}(f), N_{C1}(g) \rangle_{L^2(T_u)},$$

with $f, g \in \text{Im}(v)$.
The kernel K_v is then *extended* to the layer w by

$$K_w(f, g) = \sum_{h \in H_v} K_v(f \circ h, g \circ h) \tag{2.24}$$

with $f, g \in \text{Im}(w)$.

S1 and C1 units. The operations corresponding to $S2$ and $C2$ cells can now be defined as follows.
Consider

$$N_{S2}(f \circ h)(t) = K_w(f \circ h, t), \tag{2.25}$$

with $f \in \text{Im}(Sq)$, $h \in H_w$ and $t \in T_w$. Then $N_{S2}(f \circ h)(t)$ corresponds to the response of an $S2$ cell with template t and with receptive field $h \circ w$ for $h \in H_w$. Now let

$$N_{C2}(f)(t) = \max_{h \in H} \{N_{S2}(f \circ h)(t)\} \tag{2.26}$$

with $f \in \mathrm{Im}(Sq)$, $H = H_w$ and $t \in T_w$ so that $N_{C2} : \mathrm{Im}(Sq) \to \mathbb{R}^{|T_w|}$. Then $N_{C2}(f)(t)$ corresponds to the response of a $C2$ cell with template t and with receptive field corresponding to Sq. The derived kernel on whole images is simply

$$K_{Sq}(f, g) = \langle N_{C2}(f), N_{C2}(g) \rangle_{L^2(T_w)}$$

We add three remarks.

- We can identify the role of S and C units by splitting the definition of the neural response into two stages, where "convolution" steps (2.22) and (2.25) correspond to S units, and are followed by max operations (2.23) and (2.26) corresponding to C units.

- A key difference between the model in [98] and the development in this chapter is the "extension" step (2.24). The model proposed here corresponds to $v = w$ and is not completely faithful to the model in [98, 96] or to the commonly accepted view of physiology. However, $S2$ cells could have the same receptive field of $C1$ cells and $C2$ cells could be the equivalent of $V4$ cells. Thus the known physiology may not be inconsistent.

- Another difference lies in the kernel used in the convolution step. For sake of clarity in the above discussion we did not introduce normalization. In the model by [98] the kernels K_w, K_{Sq} are used either to define normalized dot products or as input to a Gaussian radial basis function. The former case corresponds to replacing K_w, K_{Sq} by $\widehat{K}_w$, $\widehat{K}_{Sq}$. The latter case corresponds to considering

  $$G(f, g) = e^{-\gamma d^2(f,g)},$$

  where we used the (derived) distance

  $$d^2(f, g) = K(f, f) - 2K(f, g) + K(g, g),$$

  with $K = K_w$ or $K = K_{Sq}$.

Chapter 3

Discrimination and Entropy of the Neural Response

This chapter includes joint work with Stephen Smale, Lorenzo Rosasco and
Gregory Shakhnarovich.

3.1 Introduction

In the previous Chapter, we defined a distance function on a space of images which reflects how humans see the images. In this case, the distance between two images corresponds to how similar they appear to an observer. We proposed in particular a natural image *representation*, the neural response, motivated by the neuroscience of the visual cortex. The "derived kernel" is the inner product defined by the neural response and can be used as a similarity measure. A crucial question is that of the trade-off between invariance and discrimination properties of the neural response. We earlier suggested that Shannon entropy is a useful concept towards understanding this question.

Here we substantiate the use of Shannon entropy [27] to study discrimination properties of the neural response. The approach sheds light on natural questions that arise in an analysis of the neural response: How should one choose the patch sizes? How many layers are appropriate for a given task? How many templates should be sampled? How do architectural choices induce invariance and discrimination properties? These are important and involved questions of broad significance. In this Chapter, we suggest a promising means of clarifying the picture in simplified situations that can be potentially extended to more general settings and ultimately provide answers to the questions posed above.

This Chapter is organized as follows. In Section 3.2 we begin by briefly recalling the definitions of the neural response, derived kernel, and Shannon entropy of the neural response. In Section 3.3 we then study discrimination properties in terms of information-theoretic quantities in the case of two and three layer architectures defined on strings. Finally, we provide in Section 3.4 remarks which derive intuition from the preceding development and provide additional insight into the outstanding questions above.

3.2 Background

We first briefly recall the definition of the neural response. The definition is based on a recursion which defines a hierarchy of local kernels, and can be interpreted as a multi-layer architecture.

3.2.1 Neural Response

Consider an n layer architecture given by sub-patches $v_1 \subset v_2 \subset \cdots \subset v_n = Sq$. We use the notation $K_n = K_{v_n}$ and similarly $H_n = H_{v_n}$, $T_n = T_{v_n}$. Given a kernel K, we define a normalized kernel via $\widehat{K}(x,y) = K(x,y)/\sqrt{K(x,x)K(y,y)}$.

Definition 3.2.1. Given a normalized, non-negative valued initial reproducing kernel $\widehat{K}_1$, the m layer derived kernel $\widehat{K}_m$, for $m = 2,\dots,n$, is obtained by normalizing

$$K_m(f,g) = \langle N_m(f), N_m(g)\rangle_{L^2(T_{m-1})}$$

where

$$N_m(f)(t) = \max_{h\in H} \widehat{K}_{m-1}(f \circ h, t), \qquad t \in T_{m-1}$$

with $H = H_{m-1}$.

From the above definition we see that, the neural response is a map

$$\underbrace{f \in \mathrm{Im}(Sq)}_{\text{input}} \longmapsto \underbrace{\widehat{N}_{Sq}(f) \in L^2(T) = \mathbb{R}^{|T|}}_{\text{output}},$$

with $T = T_{m-1}$ and we let $\widehat{N}_m$ denote the normalized neural response given by $\widehat{N}_m = N_m/\|N_m\|_{L^2(T)}$. We can now define a thresholded variant of the neural response, along with the induced pushforward measure on the space of orthants. In the discussion that follows, we will study the entropy of this pushforward measure as well as that of the natural measure on the space of images.

3.2.2 Thresholded Neural Response

Denote by $\mathcal{O}$ the set of orthants in $L^2(T_{m-1}) = \mathbb{R}^{|T_{m-1}|}$ identified by sequences of the form $o = (\epsilon_i)_{i=1}^{|T|}$ with $\epsilon_i \in \{0,1\}$ for all i. If we assume that $\mathbb{E}[\widehat{N}_m(f)(t)] = 0$, then the map $\widehat{N}^*_m : \mathrm{Im}(v_m) \to \mathcal{O}$ can be defined by

$$\widehat{N}^*_m(f) := \Big(\Theta(\widehat{N}_m(f)(t))\Big)_{t \in T_{m-1}}$$

where $\Theta(x) = 1$ when $x > 0$ and is 0 otherwise. From this point on, we assume normalization and drop hats in the notation. Finally, we denote by $N^{**}\rho$ the push-forward measure induced by N^*, that is

$$N^{**}\rho(A) = \rho\Big(\Big\{f \in \mathrm{Im}(Sq) \mid N_v(f) \in A\Big\}\Big),$$

for any measurable set $A \subset L^2(T_{m-1})$.

3.2.3 Shannon Entropy of the Neural Response

We introduce the Shannon entropies relative to the measures ρ_v and $N^{**}_v\rho_v$. Consider the space of images $\mathrm{Im}(v) = \{f_1, \dots, f_d\}$ to be finite. Then ρ_v reduces to $\{p_1, \dots, p_d\} = \{\rho_v(f_1), \dots, \rho_v(f_d)\}$. In this case the entropy of the measure ρ_v is

$$S(\rho_v) = \sum_i p_i \log \frac{1}{p_i}$$

and similarly,

$$S(N^{**}_v\rho_v) = \sum_{o \in \mathcal{O}} q_o \log \frac{1}{q_o}\,.$$

where $q_o = (N^{**}_v\rho_v)(o)$ is explicitly given by

$$(N^{**}_v\rho_v)(o) = \rho_v\Big(\Big\{f \in \mathrm{Im}(v) \mid \Big(\Theta(N_v(f)(t))\Big)_{t \in |T|} = o\Big\}\Big).$$

If $\mathrm{Im}(v)$ is not finite we can define the entropy $S(\rho_v)$ associated to ρ_v by considering a partition $\pi = \{\pi_i\}_i$ of $\mathrm{Im}(v)$ into measurable subsets. The entropy of ρ_v given the partition π is then given by

$$S_\pi(\rho_v) = \sum_i \rho_v(\pi_i) \log \frac{1}{\rho_v(\pi_i)}.$$

One can define the entropy of the original, non-thresholded neural response in a similar fashion, taking a partition of the range of $\widehat{N}_v$.

The key quantity we'll need to assess the discriminative power of the neural response N_v is the *discrepancy*

$$\Delta S = S(\rho_v) - S(N^{**}_v\rho_v).$$

It is easy to see that

$$S(\rho_v) \geq S(N_v^{**}\rho_v) \tag{3.1}$$

so that $\Delta S \geq 0$. The discrepancy is zero if N^* is one to one (see remark below). Conversely, we achieve greater invariance when the discrimination ability decreases and ΔS increases towards $S(\rho_v)$.

We add two remarks.

Remark 3.2.1. *Let X, Y be two random variables. We briefly recall the derivation of the inequality* (3.1)*, which we write here as $S(Y) \leq S(X)$. We use two facts: (a) $S(X,Y) = S(X)$ if $Y = f(X)$ with f deterministic, and (b) $S(X,Y) \geq S(Y)$ in general. To prove (a), write $P(Y = y, X = x) = p(Y = y|X = x)P(X = x) = \delta(y, f(x))P(X = x)$, and we sum over all $y = x$ in the definition of the joint entropy $S(X,Y)$.*

Remark 3.2.2. *Consider, for example, the finite partition $\pi = \{\pi_o\}_{o\in\mathcal{O}}$ on the space of images induced by N_v^*, with*

$$\pi_o = \Big\{f \in \mathrm{Im}(v) \mid \Big(\Theta(N_v(f)(t))\Big)_{t\in|T|} = o\Big\}.$$

We might also consider only the support of ρ_v, which could be much smaller than $\mathrm{Im}(v)$*, and define a similar partition of this subset as*

$$\pi_o = \Big\{f \in \operatorname{supp}\rho_v \mid \Big(\Theta(N_v(f)(t))\Big)_{t\in|T|} = o\Big\},$$

with $\pi = \{\pi_o \mid \pi_o \neq \emptyset\}$. One can then define a measure on this partition and corresponding notion of entropy.

3.3 Shannon Entropy of the Neural Response on Strings

Let A be an alphabet of k distinct letters so that $|A| = k$. Consider three layers $u \subset v \subset w$, where $\mathrm{Im}(u) = A$, $\mathrm{Im}(v) = A^m$ and $\mathrm{Im}(w) = A^n$, with $1 < m < n$. The kernel $K_u = \widehat{K}_u$ on single characters is simply, $K_u(f,g) = 1$, if $f = g$ and 0 otherwise. The template sets are $T_u = A$ and $T_v = A^m$.

3.3.1 Explicit Expressions for N and K

We specialize the definitions of the neural response and derived kernel in the case of strings.

The neural response at layer v is defined by

$$N_v(f)(t) = \max_{h\in H_u}\left\{\widehat{K}_u(f\circ h, t)\right\},$$

and is a map $N_v : A^m \to \{0,1\}^k$. The norm of the neural response is

$$\|\widehat{N}_v(f)\| =: a(f) = \#\text{ distinct letters in } f.$$

From the definition of the derived kernel we have that

$$K_v(f,g) =: a(f,g) = \#\text{ distinct letters common to } f \text{ and } g.$$

The normalized kernel can then be written as

$$\widehat{K}_v(f,g) = \frac{a(f,g)}{(a(f)a(g))^{1/2}}.$$

The neural response at layer w then satisfies

$$N_w(f)(t) = \frac{e(f,t)}{a(t)^{1/2}},$$

with

$$e(f,t) := \max_{h \in H_v} \frac{a(f \circ h, t)}{a(f \circ h)^{1/2}}.$$

This is the maximum fraction of distinct letters in m-substrings of f that are shared by t. Finally the derived kernel at layer w satisfies

$$\widehat{K}_w(f,g) = \frac{\sum_{t \in T_v} \frac{e(f,t)e(g,t)}{a(t)}}{\sum_{t \in T_v} \frac{e(f,t)^2}{a(t)} \sum_{t \in T_v} \frac{e(q,t)^2}{a(t)}}.$$

We are interested in knowing whether the neural response is injective up to reversal and checkerboard. If N_v^* is injective, then the inequality (3.1) holds with equality. We can consider N_v^* as acting on the set of equivalence classes of $\mathrm{Im}(v)$ defined by the strings and their reversals, and if n is odd, a checkerboard when applicable (see previous Chapter for a discussion concerning the checkerboard pattern). Here injectivity of N_v^* is with respect to the action on equivalence classes. The following result is easy to prove.

Proposition 3.3.1. *N_v^* is injective if and only if $\mathrm{Im}(v)$ contains strings of length 2.*

3.3.2 Orthant Occupancy

We consider a 2-layer architecture and let $k = |A| > m$. As before, $\mathrm{Im}(v)$ contains strings of length m, and $\mathrm{Im}(u)$ contains single characters. The number of non-empty orthants is

$$\sum_{\ell=1}^{m} \binom{k}{\ell}.$$

The "all zero" orthant is always empty (strings must use at least one letter in the alphabet). Let $\mathcal{O}_p$ denote the set of orthants corresponding to strings of $p < m$ distinct letters, that is

$$\mathcal{O}_p = \Big\{ o \in \mathcal{O} \mid \sum_{i=1}^{k} \epsilon_i = p \Big\}.$$

Let $\lambda_o(p, m)$ denote the number of strings mapped into the orthant $o \in \mathcal{O}_p$. Then

$$\lambda_o(p, m) = k^m q_o.$$

If the measure ρ_v is uniform then $\lambda_o(p, m)$ is the same for all $o \in \mathcal{O}_p$ and we drop the subscript on λ. In the uniform case we have the following recursive formula

$$\lambda(p, m) = p^m - \sum_{j=1}^{p-1} \binom{p}{j} \lambda(j, m),$$

with $\lambda(1, m) = 1$.

We now give an explicit expression for the discrepancy $S(\rho_v) - S(N_v^{**}\rho_v)$. If ρ_v is uniform

$$S(\rho_v) = m \log k.$$

With a little algebra we have that

$$\begin{aligned} S(N_v^{**}\rho_v) &= -\sum_{j}^{m} \binom{k}{j} \frac{\lambda(j, m)}{k^m} \log \frac{\lambda(j, m)}{k^m} \\ &= -\sum_{j}^{m} \binom{k}{j} \frac{\lambda(j, m)}{k^m} \log \lambda(j, m) + \underbrace{\log k^m}_{S(\rho_v)} \underbrace{\sum_{j}^{m} \binom{k}{j} \frac{\lambda(j, m)}{k^m}}_{=1}, \end{aligned}$$

and we obtain the following explicit expression for the discrepancy

$$\Delta S = S(\rho_v) - S(N_v^{**}\rho_v) = \sum_{j}^{m} \binom{k}{j} \frac{\lambda(j, m)}{k^m} \log \lambda(j, m).$$

This quantity can be seen as a weighted average, writing $\Delta S = \sum_j^m b_j \log \lambda(j, m)$ and noting that $\sum_j b_j = 1$.

Non-recursive Occupancy Formula (2-layer Case)

Alternatively, we can use multinomial coefficients to describe the number of m-strings mapped into the $\binom{k}{p}$ orthants with exactly $p < m$ ones as follows:

$$\begin{aligned}\lambda(p,m) &= \sum_{r_1,\ldots,r_p} \binom{m}{r_1,\ldots,r_p} \\ &= p!S(m,p) \\ &= \sum_{t=1}^{p} (-1)^{p+t} \binom{p}{t} t^m\end{aligned}$$

where the first summation is taken over all sequences of *positive* integer indices $r_1,\ldots,r_p$ such that $\sum_{i=1}^{p} r_i = m$. The number of terms in this summation is the number of p-part compositions of m[1] and is given by $\binom{m-1}{p-1}$. The $S(m,p)$ are Stirling numbers of the second kind [2], and the final equality follows from direct application of Stirling's Identity. Note that since $S(m,1) = 1$, we can verify that $\lambda(1,m) = S(m,1) = 1$.

From the multinomial theorem, we also have that

$$\sum_{\{r_i \geq 0 : r_1+\cdots+r_p=m\}} \binom{m}{r_1,\ldots,r_p} = (1+\cdots+1)^m = p^m = \sum_{k=1}^{p} \lambda(k,m)$$

which checks with the previous recursive definition.

3.4 Remarks

We add some remarks concerning the application of the above ideas towards understanding the role of the patch sizes and layers in inducing discrimination and invariance properties.

- In a 3-layer network, $u \subset v \subset w$, $N_w^*(f)(t) \to 1$ in probability as $n \to \infty$, for all $t \in T_v$, with T_v exhaustive and $f \sim \rho_w$ with ρ_w uniform. As the string f gets infinitely long, then the probability we find a given template in that string goes to 1. Note that the rate is very slow: for example, there are $(k-1)^n$ possible strings which do not include a given letter which would appear in many templates.

- The above also implies that the entropy $S(N_w^{**}\rho_w) \to 0$ as $n \to \infty$ since all images f are mapped to the orthant of all ones.

[1] A p-part composition of m is a solution to $m = r_1 + \cdots + r_p$ consisting of positive integers.

[2] $S(m,p)$ counts the number of ways one can partition sets of size m into p nonempty subsets.

- Consider a 3 layer architecture: $u = \mathrm{Str}(1) \subset v = \mathrm{Str}(m) \subset w = \mathrm{Str}(n)$ with n fixed, all translations, and exhaustive template sets.

  Question: Which choice of m maximizes $S(N_w^{**}\rho_w)$? For large m most of the probability will concentrate near the "sparse" orthants – the orthants characterized by many zeros– because the probability of finding a long template in f is low. For small m, most of the probability mass will fall in the orthants with many ones – where a large number of templates match pieces of f. In both cases, the entropy is low because the number of orthants with few 1's or few 0's is small. For some intermediate choice of m, the entropy should be maximized as the probability mass becomes distributed over the many orthants which are neither mostly zeros nor mostly ones. (consider the fact that $\binom{a}{a/2} \gg \binom{a}{1}$ or $\binom{a}{a-1}$), where a is a positive even integer)

In this Chapter, we have shown the possibility of mathematically analyzing the discriminatory power of the neural response in simple cases, via entropy. It is our hope that the methods suggested here can be extended and ultimately leveraged to understand in concrete terms how parametric and architectural choices influence discrimination and invariance properties.

Chapter 4

Group-Theoretic Perspectives on Invariance

A goal of central importance in the study of hierarchical models for object recognition – and indeed the mammalian visual cortex – is that of understanding quantitatively the invariance-selectivity tradeoff, and how invariance and discrimination properties contribute towards providing an improved representation useful for learning from data. In this Chapter we provide a general group-theoretic framework for characterizing and understanding invariance in a family of hierarchical models. We show that by taking an algebraic perspective, one can provide a unified set of conditions which must be met to establish invariance, as well as a constructive prescription for meeting those conditions. Analyses in specific cases of particular relevance to computer vision and text processing are given, yielding insight into how and when invariance can be achieved. We find that the minimal sets of transformations intrinsic to the hierarchical model needed to support a particular invariance can be clearly described, thereby encouraging efficient computational implementations.

4.1 Introduction

Several models of object recognition drawing inspiration from visual cortex have been developed over the past few decades [40, 71, 65, 118, 98, 96], and have enjoyed substantial empirical success. A central theme found in this family of models is the use of Hubel and Wiesel's simple and complex cell ideas [52]. In the primary visual cortex, features are computed by simple units by looking for the occurrence of a preferred stimulus in a region of the input ("receptive field"). Translation invariance is then explicitly built into the processing pathway by way of complex units which pool locally over simple units. The alternating simple-complex filtering/pooling process is repeated, building increasingly *invariant* representations which are simultaneously *selective* for increasingly complex stimuli.

A goal of central importance in the study of hierarchical architectures and the visual cortex alike is that of understanding quantitatively this invariance-selectivity tradeoff, and how invariance and selectivity contribute towards providing an improved representation useful for learning from examples.

In Chapter 2, we established a framework which makes possible a more precise characterization of the operation of hierarchical models via the study of invariance and discrimination properties. However we studied invariance in an implicit, rather than constructive, fashion. Two specific cases were analyzed: invariance with respect to image rotations and string reversals, and the analysis was tailored to the particular setting. In this Chapter, we reinterpret and extend the invariance analysis in Chapter 2 using a group-theoretic language, towards clarifying and unifying the general properties necessary for invariance in a family of hierarchical, multi-scale models. We show that by systematically applying algebraic tools, one can provide a concise set of conditions which must be met to establish invariance, as well as a constructive prescription for meeting those conditions. We additionally find that when one imposes the mild requirement that the transformations of interest have group structure, a broad class of hierarchical models can only be invariant to orthogonal transformations. This result suggests that common architectures found in the literature might need to be rethought and modified so as to allow for broader invariance possibilities. Finally, we show that the proposed framework automatically points the way to efficient computational implementations of invariant models.

The Chapter is organized as follows. We very briefly recall important definitions from Chapter 2. Next, we extend the framework in Chapter 2 to a more general setting allowing for arbitrary pooling functions, and give a proof for invariance of the corresponding family of hierarchical feature maps. This contribution is key because it shows that several results in Chapter 2 *do not depend on the fact that the* max *was used*. We then establish a group-theoretic framework for characterizing invariance in hierarchical models expressed in terms of the generalized set of objects defined here. Within this framework, we turn to the problem of invariance in two specific domains of practical relevance: images and text strings. Finally, we conclude with a few remarks summarizing the contributions and relevance of our work. The reader is assumed to be familiar with introductory concepts in group theory. An excellent reference is [4].

4.2 Invariance of a Hierarchical Feature Map

We first review important definitions and concepts concerning the neural response map presented in Chapter 2, drawing attention to the conditions needed for the neural response to be invariant with respect to a family of arbitrary transformations. We then generalize the neural response map to allow for arbitrary pooling functions and adapt the previously presented proof of invariance. In this more general setting, we are able to describe for a broad range of hierarchical models (including a class of convolutional neural networks), necessary conditions for invariance to a set of transformations.

First define a system of patches associated to successive layers $v_1 \subset v_2 \subset \cdots \subset v_n$, with v_n the size of the full input, and spaces of functions on the patches $\mathrm{Im}(v_i)$. In many cases it will only be necessary to work with arbitrary successive pairs of patches, in which case we will denote by u the smaller patch, and v the next larger patch. Next, denote by $h \in H_i$ the translation functions $h : v_i \to v_{i+1}$, for all $i = 1, \ldots, n-1$. To each layer we also associate a dictionary of templates, $T_i \subseteq \mathrm{Im}(v_i)$, typically constructed by randomly sampling from an appropriate probability measure. We will discuss also transformations $r \in \mathcal{R}_i$ with $r : v_i \to v_i$ for all $i = 1, \ldots, n$, but rule out the degenerate transformations h or r which map their entire domain to a single point. When it is necessary to identify transformations defined on a specific domain v, we will use the notation $r_v : v \to v$.

We note that these definitions alone immediately constrain the possible transformations one can consider: All $h \in H_i$ cannot be surjective, and therefore cannot be bijective. If H_i contains only translations, then clearly every $h \in H_i$ is injective. And if the patches $\{v_i\}_i$ are finite, then every $r \in \mathcal{R}_i$, for $i = 1, \ldots, n$ must be either bijective, or neither injective nor surjective.

The neural response $N_m(f)$ and associated derived kernel $\widehat{K}_m$ are defined as follows.

Definition 4.2.1. Given a non-negative valued, normalized, initial reproducing kernel $\widehat{K}_1$, the m-layer derived kernel $\widehat{K}_m$, for $m = 2, \ldots, n$, is obtained by normalizing

$$K_m(f, g) = \langle N_m(f), N_m(g) \rangle_{L^2(T_{m-1})}$$

where

$$N_m(f)(t) = \max_{h \in H} \widehat{K}_{m-1}(f \circ h, t), \qquad t \in T_{m-1}$$

with $H = H_{m-1}$.

Here normalization is given by $\widehat{K}(f, g) = K(f, g)/\sqrt{K(f, f)K(g, g)}$. Note that the neural response decomposes the input into a hierarchy of parts, analyzing subregions at different scales. The neural response and derived kernels describe in compact, abstract terms the core operations built into the many related hierarchical models of object recognition cited above. The reader is encouraged to consult Chapter 2 for a more detailed discussion.

4.2.1 Generalized Pooling and Invariance

We next provide a generalized proof of invariance of a family of hierarchical feature maps, where the properties we derive do not depend on the choice of the pooling function. A crucial component is the following modified invariance Assumption. This condition applies regardless of whether the transformations $\mathcal{R}_i$ exhibit group structure or not.

Assumption 4.2.1. *Fix any $r \in \mathcal{R}$. There exists a surjective map $\pi : H \to H$ satisfying*

$$r_v \circ h = \pi(h) \circ r_u \tag{4.1}$$

for all $h \in H$.

We will show that given the above assumption, invariance of the neural response can be established for general pooling functions of which the max is one particular choice. We first consider the case where H is assumed to be neither finite nor countable, and then from this picture show that the more familiar case of finite H can also be described.

We first assume that $H \subset \mathcal{H}$, where $\mathcal{H}$ is the set of all possible appropriately defined translations, and H is an unordered subset. Let $\mathcal{B}(\mathbb{R})$ denote the Borel algebra of $\mathbb{R}$. As in Assumption 4.2.1, we define $\pi : H \to H$ to be a surjection, and let $\Psi : \mathcal{B}(\mathbb{R}_{++}) \to \mathbb{R}_{++}$ be a *pooling function* defined for Borel sets $B \in \mathcal{B}(\mathbb{R})$ consisting of only positive elements. Given a positive functional F acting on $\mathcal{H}$, we define the set $F(H) \in \mathcal{B}(\mathbb{R})$ as

$$F(H) = \{F[h] \mid h \in H\}.$$

Note that since π is surjective, $\pi(H) = H$, and therefore $(F \circ \pi)(H) = F(H)$.

With these definitions in hand, we can define a more general "neural response" as follows. For $H = H_{m-1}$ and all $t \in T = T_{m-1}$, let the neural response be given by

$$N_m(f)(t) = (\Psi \circ F)(H)$$

where

$$F[h] = \widehat{K}_{m-1}(f \circ h, t).$$

We can now give a proof of invariance of the neural response, assuming a general pooling function Ψ.

Proposition 4.2.1. *Given any function $\Psi : \mathcal{B}(\mathbb{R}_{++}) \to \mathbb{R}_{++}$, if the initial kernel satisfies $\widehat{K}_1(f, f \circ r) = 1$ for all $r \in \mathcal{R}$, $f \in \operatorname{Im}(v_1)$, then*

$$\widehat{N}_m(f) = \widehat{N}_m(f \circ r),$$

for all $r \in \mathcal{R}$, $f \in \operatorname{Im}(v_m)$ and $m \le n$.

Proof. The proof is by induction. The base case is true by assumption. The inductive hypothesis is that $\widehat{K}_{m-1}(u, u \circ r) = 1$ for any $u \in \operatorname{Im}(v_{m-1})$. This means that $F(H \circ r) = F(H)$. Assumption 4.2.1 states that $r \circ H = \pi(H) \circ r = H \circ r$. Combining the inductive hypothesis and the Assumption, we see that $N_m(f \circ r)(t) = (\Psi \circ F)(r \circ H) = (\Psi \circ F)(H \circ r) = (\Psi \circ F)(H) = N_m(f)(t)$. □

We give a few practical examples of the pooling function Ψ.

Maximum: The original neural response is recovered setting

$$\Psi(B) = \sup B\,.$$

Averaging: We can consider an averaging model by setting

$$\Psi(B) = \int_{x \in B} x d\mu$$

where a natural choice of the measure μ is the push-forward measure $\rho_H \circ F^{-1}$ induced by the given measure on H, ρ_H. If H is a finite sample drawn from ρ_H, then setting $B = F(H)$ we can consider the empirical measure $\mu = \frac{1}{|B|}\sum_{b \in B} \delta_b$, where δ_b is the Dirac measure centered on b. In this case

$$\Psi(B) = \int_{x \in B} x d\mu = \tfrac{1}{|B|}\sum_{b \in B} b.$$

Other Norms: We may consider other norms, such as the weighted L^p family,

$$\Psi(B) = \Big(\int_{x \in B} w(x)|x|^p d\mu\Big)^{1/p},$$

where $w(x)$ is a positive weight function.

Symmetric Polynomials: If $B = \{b_i\}_i^N$, one can consider pooling functions which are symmetric polynomials in N variables. A potentially useful polynomial is

$$\Psi(B) = \prod_{i=1}^{N}\prod_{j=i+1}^{N} |b_i - b_j|^{\alpha}.$$

This pooling function can be interpreted as preferring a sparse pattern of responses in the pooling region. If the template is a strong match (as defined by K) in two or more areas, Ψ will be small, whereas strong activation in one area and nowhere else leads to a comparatively large output. The precise fall off with increasing numbers of strong activations can be controlled via the (positive) parameter α.

4.3 A Group-Theoretic Invariance Framework

This section establishes general definitions and conditions needed to formalize a group-theoretic concept of invariance. The development presented here seeks to separate requirements for invariance stemming from the organization and technical definitions of the neural response, from requirements which come from group-related conditions applicable to the set of transformations and translations under consideration.

Recalling that the neural response is defined on an appropriate space of functions, denote by S the set on which image functions are defined, and let G be a group. The set S could contain, for example, points in $\mathbb{R}^2$ (in the case of 2D graphics) or positive integer indices (the case of strings). Because it will be necessary to translate in S, it is assumed that an appropriate notion of addition between the elements of S is given. We will be concerned only with invariance to domain transformations, and so consider a suitable (left) action of G on S defined by $A : G \times S \to S$. Given an element $g \in G$, the notation $A_g : S \to S$ will be utilized. Since A is a group action, by definition it satisfies $(A_g \circ A_{g'})(x) = A_{gg'}(x)$ for all $x \in S$ and all $g, g' \in G$. An explicit characterization of A is dependent on the particular setting.

Assumption 4.2.1 requires that $r_v \circ h = h' \circ r_u$ for $h, h' \in H$ and $r \in \mathcal{R}$, with the map $h \mapsto h'$ onto. We can formalize this condition in group theoretic terms, and consider an arbitrary pair of successive layers with associated patch sizes u and v. Recall that the definition of the neural response involves the "built-in" translation/restriction functions $h : u \to v$ with $u \subset v \subset S$. The restriction behavior of the translations in H, however, poses a difficulty vis. inverse elements in a group. To get around this difficulty, we decompose the $h \in H$ into a composition of two functions: the action of a translation and an inclusion. Denote by T the group of translations appropriate for the domain of interest S. Note that although we assume the specific case of translations here, the set of built-in operations may more generally contain other kinds of transformations. *We assume, however, that T is abelian.*

We begin by writing $h_a : u \to v$ and connect the translation behavior of the $h \in H$ to the group T by defining an injective map from translation parameters to group elements:

$$\begin{aligned} S &\to T \\ a &\mapsto t_a. \end{aligned} \tag{4.2}$$

Note that although T is defined to be a group, S need not have group structure. Next, define $\iota_u : u \hookrightarrow v$ to be the canonical inclusion of u into v. Then for $h_a : u \to v$ with $h_a \in H, a \in S, x \in u$, we can write $h_a = A_{t_a} \circ \iota_u$ where it is assumed that the action A is defined borrowing addition from S such that $A_{t_a}(x) = x + a$. The above dissociation of domain transformation from restriction is subtle but important.

We make an additional assumption that the transformations with respect to which invariance is considered are elements of a group, and denote this group by the symbol R. As in the case of translations, we assume that for every $r \in \mathcal{R}$, there is a unique corresponding element $\rho \in R$ whose action satisfies $A_{\rho_u}(x) = r_u(x)$ and $A_{\rho_v}(x) = r_v(x)$, for all $x \in S$. Here, to each group element $\rho \in R$ we implicitly associate two elements $\rho_u, \rho_v \in R_{uv}$, where ρ_u and ρ_v are transformations built from ρ, but which may be different in the context of an action defined on u or v. The group R_{uv} is the smallest group generated by the elements $\{\rho_v, \rho_u \mid \rho \in R\}$. The distinction between ρ, ρ_u and ρ_v will become clear in the case of feature maps defined on functions whose domain is a finite set (such as strings). We will abuse notation and denote by r both elements of $\mathcal{R}$ and corresponding elements in R.

We would next like to define an action for a single group G so that compositions of particular group elements' actions are equivalent to the action of some other element by way of associativity. This can be accomplished by forming the semidirect product

$$G = T \rtimes R \tag{4.3}$$

where T is assumed to be normal in G, and $T \cap R = \{1\}$. Note that although this construction precludes R from containing the transformations in T, allowing R to contain translations is an uninteresting case. The resulting group G is easily shown to be isomorphic to a group with normal subgroup T and subgroup R where each element may be written in the form $g = tr$ for $t \in T, r \in R$. Other instances of built-in transformations beyond simple translations need only satisfy the condition that T is a normal, abelian subgroup of G.

Define the injection $\tau : H \to T$ by $h_a \mapsto t_a$, where a is a parameter characterizing the translation h_a. Then we can define $\tilde{T} = \tau(H_u) \subseteq T$ as the set of group elements associated with a layer (at scale u) of the neural response map. Substituting $h = h_a, h' = h_b$, and denoting by r the element of R corresponding to a transformation in $\mathcal{R}$, the condition $r_v \circ h = h' \circ r_u$ for some $h' \in H$ can now be expressed as

$$A_{r_v} \circ A_{t_a} \circ \iota_u = A_{t_b} \circ \iota_u \circ A_{r_u} \circ \iota_u \tag{4.4}$$

for some $t_b \in \tilde{T}$.

Our goal is to describe compositions of transformations $r \in \mathcal{R}$ and translations $t_a \in \tilde{T}$. However on the right hand side of Equation (4.4) the translation A_{t_b} is separated from the transformation A_{r_u} by the inclusion ι_u so we will therefore need to introduce an additional condition on R: we require that if $x \in u$, then $A_{r_u}(x) \in u$ for all $r \in R$. The precise implications of this constraint will be explored below.

One can now see that if the above requirement is satisfied, then the condition (4.4) reduces to verifying that $A_{r_v} \circ A_{t_a} = A_{t_b} \circ A_{r_u}$, and that the map $t_a \mapsto t_b$ is onto. Applying the associativity property of the action A, we can express this equality in clear group-theoretic terms as the following. Given any $r \in R$,

$$r_v\tilde{T} = \tilde{T}r_u\,. \tag{4.5}$$

This is a purely algebraic requirement concerning the groups R and T, distinct from the restriction related conditions involving the patches u and v. The requirements above combine to capture the content of Assumption 4.2.1, but in a way that clearly separates group related invariance conditions from constraints due to restriction and the nested nature of an architecture's patch domains.

The above discussion can be summarized in the form of a concise definition that can be applied to establish invariance of the neural response feature maps $N_m(f)$, $2 \leq m \leq n$ with respect to a set of transformations. Let $\tilde{R} \subseteq G$ be the set of transformations for which we would like to prove invariance, corresponding to $\mathcal{R}$.

Definition 4.3.1 (Compatible Sets)**.** The subsets $\tilde{R} \subset G$ and $\tilde{T} \subset T$ are *compatible* if all of the following conditions hold:

1. For each $r \in \tilde{R}$, $r_v\tilde{T} = \tilde{T}r_u$. When $r_u = r_v$ for all $r \in R$, this means that normalizer of $\tilde{T}$ in $\tilde{R}$ is $\tilde{R}$.
2. Left transformations r_v never take a point in v outside of v, and right transformations r_u never take a point in u/v outside of u/v (respectively):

$$\mathrm{im}A_{r_v} \circ \iota_v \subseteq v, \qquad \mathrm{im}A_{r_u} \circ \iota_u \subseteq u, \qquad \mathrm{im}A_{r_u} \circ \iota_v \subseteq v,$$

   for all $r \in \tilde{R}$.
3. Translations never take a point in u outside of v:

$$\mathrm{im}A_t \circ \iota_u \subseteq v$$

   for all $t \in \tilde{T}$.

The final condition above has been added to ensure that the set of translations $\tilde{T}$ satisfy the implicit assumption that the hierarchy's translations $h \in H$ are maps which respect the definition $h : u \to v$.

If $\tilde{R}$ and $\tilde{T}$ are compatible, then given $r \in \tilde{R}$ and $t \in \tilde{T}$,

$$f \circ A_{r_v} \circ A_t \circ \iota_u = f \circ A_{t'} \circ A_{r_u} \circ \iota_u \tag{4.6}$$

for some $t' \in \tilde{T}$, with $f \in \mathrm{Im}(v)$ and $u \subset v \subset S$. In addition, the induced map $\pi : H \to H$ sending h to h' is surjective. As will become clear in the following section, the tools available to us from group theory will provide insight into the structure of compatible sets.

4.3.1 Orbits and Compatible Sets

Suppose we assume that $\tilde{R}$ is a group, and ask for the smallest compatible $\tilde{T}$. We will show that the only way to satisfy Condition (1) in Definition 4.3.1 is to require that $\tilde{T}$ be a union of $\tilde{R}$-*orbits*, under the action

$$(t, r) \mapsto r_v t r_u^{-1} \tag{4.7}$$

for $t \in T$, $r \in \tilde{R}$. This perspective is particularly illuminating because it will eventually allow us to view conjugation by a transformation r as a permutation of $\tilde{T}$, thereby establishing surjectivity of the map π defined in Assumption 4.2.1. For computational reasons, viewing $\tilde{T}$ as a union of orbits is also convenient.

If $r_v = r_u = r$, then the action (4.7) is exactly conjugation and the $\tilde{R}$-orbit of a translation $t \in T$ is the conjugacy class $C_{\tilde{R}}(t) = \{rtr^{-1} \mid r \in \tilde{R}\}$. Orbits of this form are also equivalence classes under the relation $s \sim s'$ if $s' \in C_{\tilde{R}}(s)$, and we will require $\tilde{T}$ to be partitioned by the conjugacy classes induced by $\tilde{R}$.

The following Proposition shows that, given set of candidate translations in H, we can construct a set of translations compatible with $\tilde{R}$ by requiring $\tilde{T}$ to be a union of $\tilde{R}$-orbits under the action of conjugation.

Proposition 4.3.1. *Let $\Gamma \subseteq T$ be a given set of translations, and assume the following: (1) $G \cong T \rtimes R$, (2) For each $r \in R$, $r = r_u = r_v$, (3) $\tilde{R}$ is a subgroup of R. Then Condition (1) of Definition 4.3.1 is satisfied if and only if $\tilde{T}$ can be expressed as a union of orbits of the form*

$$\tilde{T} = \bigcup_{t \in \Gamma} C_{\tilde{R}}(t)\,. \tag{4.8}$$

Proof. We first show that for $\tilde{T}$ of the form above, Condition (1) of Definition 4.3.1 is satisfied. Combining the first two assumptions, we have for all $t \in T$, $r \in \tilde{R}$ that $r_v t r_u^{-1} = rtr^{-1} \in T$. Then for each $r \in \tilde{R}$,

$$\begin{aligned} r_v \tilde{T} r_u^{-1} = r\tilde{T}r^{-1} = r\left(\bigcup_{t\in\Gamma} C_{\tilde{R}}(t)\right) r^{-1} = r\left(\bigcup_{t\in\Gamma}\{\tilde{r}t\tilde{r}^{-1} \in T \mid \tilde{r} \in \tilde{R}\}\right) r^{-1} \\ = \bigcup_{t\in\Gamma}\{r\tilde{r}t\tilde{r}^{-1}r^{-1} \in T \mid \tilde{r} \in \tilde{R}\} = \bigcup_{t\in\Gamma}\{r'tr'^{-1} \in T \mid r' \in \tilde{R}\} = \tilde{T} \end{aligned}$$

where the last equality follows since $r' \equiv r\tilde{r} \in \tilde{R}$, and $gG = G$ for any group G and $g \in G$ because $gG \subseteq G$ combined with the fact that $Gg^{-1} \subseteq G \Rightarrow Gg^{-1}g \subseteq Gg \Rightarrow G \subseteq Gg$ giving $gG = G$. So the condition is verified. Suppose now that Condition (1) is satisfied, but $\tilde{T}$ is not a union of orbits for the action of conjugation. Then there is a $t' \in \tilde{T}$ such that t' *cannot* be expressed as $t' = rtr^{-1}$ for some $t \in \tilde{T}$. Hence $r\tilde{T}r^{-1} \subset \tilde{T}$. But this contradicts the assumption that Condition (1) is satisfied, so $\tilde{T}$ must be of the form shown in Equation (4.8). □

An interpretation of the above Proposition, is that when $\tilde{T}$ is a union of $\tilde{R}$-orbits, conjugation by r can be seen as a permutation of $\tilde{T}$. In general, a given $\tilde{T}$ may be decomposed into several such orbits and the conjugation action of $\tilde{R}$ on $\tilde{T}$ may not necessarily be transitive.

Conversely, suppose we fix $\tilde{T} \subseteq T$, and attempt to characterize a non-trivial $\tilde{R}$ such that each $(r, t) \in \tilde{T} \times \tilde{R}$ satisfies Condition (1) of Definition 4.3.1. Consider conjugation by elements of $\tilde{R}$ on the subset $\tilde{T}$. The orbit of $\tilde{T}$ for this operation is $\{r\tilde{T}r^{-1} \mid r \in \tilde{R}\}$. We would like the orbit of $\tilde{T}$ to be itself naturally, and so one can define $\tilde{R}$ to be composed of elements taken from the stabilizer (normalizer) of $\tilde{T}$: $\tilde{R} \subseteq N_R(\tilde{T}) = \{r \in R \mid r\tilde{T}r^{-1} = \tilde{T}\}$. Once again, conjugation by r can be seen as a permutation on $\tilde{T}$.

4.4 Analysis of Specific Invariances

We continue with specific examples relevant to image processing and text analysis.

4.4.1 Isometries of the Plane

Consider the case where G is the group M of planar isometries, $u \subset v \subset S = \mathbb{R}^2$, and H involves translations in the plane. Let O_2 be the group of orthogonal operators, and let $t_a \in T$ denote a translation represented by the vector $a \in \mathbb{R}^2$. In this section we assume the standard basis and work with matrix representations of G when it is convenient.

We first need that $T \lhd M$, a property that will be useful when verifying Condition (1) of Definition 4.3.1. Indeed, from the First Isomorphism Theorem [4], the quotient space M/T is isomorphic to O_2, giving the following commutative diagram:

$$\begin{tikzcd}
M \arrow[r, "\pi"] \arrow[d, "\phi"'] & O_2 \\
M/T \arrow[ur, dashed, "\tilde{\pi}"'] &
\end{tikzcd}$$

where the isomorphism $\tilde{\pi} : M/T \to O_2$ is given by $\tilde{\pi}(mT) = \pi(m)$ and $\phi(m) = mT$. We recall that the kernel of a group homomorphism $\pi : G \to G'$ is a normal subgroup of G, and that normal subgroups N of G are invariant under the operation of conjugation by elements g of G. That is, $gNg^{-1} = N$ for all $g \in G$. With this picture in mind, the following Lemma establishes that $T \lhd M$, and further shows that M is isomorphic to $T \rtimes R$ with $R = O_2$, and T a normal subgroup of M.

Lemma 4.4.1. *For each $m \in M$, $t_a \in T$,*

$$mt_a = t_bm$$

for some unique element $t_b \in T$.

Proof. The group of isometries of the plane is generated by translations and orthogonal operators, so we can write an element $m \in M$ as $m = t_v\varphi$. Now define the homomorphism $\pi : M \to O_2$, sending $t_v\varphi \mapsto \varphi$. Clearly T is the kernel of π and is therefore a normal subgroup of M. Furthermore, $M = T \rtimes O_2$. So we have that, for all $m \in M$, $mt_am^{-1} = t_b$, for some element $t_b \in T$. Denote by $\varphi(v)$ the operation of $\varphi \in O_2$ on a vector $v \in \mathbb{R}^2$ given by the standard matrix representation $R : O_2 \to \mathsf{GL}_2(\mathbb{R})$ of O_2. Then $\varphi t_a = t_b\varphi$ with $b = \varphi(a)$ since $\varphi \circ t_a(x) = \varphi(x + a) = \varphi(x) + \varphi(a) = t_b\varphi(x)$. So for arbitrary isometries m, we have that

$$mt_am^{-1} = (t_v\varphi)t_a(\varphi^{-1}t_{-v}) = t_vt_b\varphi\varphi^{-1}t_{-v} = t_b,$$

where $b = \varphi(a)$. Since the operation of φ is bijective, b is unique, and $mTm^{-1} = T$ for all $m \in M$. □

We are now in a position to verify the Conditions of Definition 4.3.1 for the case of planar isometries.

Proposition 4.4.1. *Let H be the set of translations associated to an arbitrary layer of the hierarchical feature map and define the injective map $\tau : H \to T$ by $h_a \mapsto t_a$, where a is a parameter characterizing the translation. Set $\Gamma = \{\tau(h) \mid h \in H\}$. Take $G = M \cong T \rtimes O_2$ as above. The sets*

$$\tilde{R} = O_2, \qquad \tilde{T} = \bigcup_{t \in \Gamma} C_{\tilde{R}}(t)$$

are compatible.

Proof. We first note that in the present setting, for all $r \in R$, $r = r_u = r_v$. In addition, Lemma 4.4.1 says that for all $t \in T$, $r \in O_2$, $rtr^{-1} \in T$. We can therefore apply Proposition 4.3.1 to verify Condition (1) in Definition 4.3.1 for the choice of $\tilde{T}$ above. Since $\tilde{R}$ is comprised of orthogonal operators, Condition (2) is immediately satisfied. Condition (3) requires that for every $t_a \in \tilde{T}$, the magnitude of the translation vector a must be limited so that $x + a \in v$ for any $x \in u$. We assume that every $h_a \in H$ never takes a point in u outside of v by definition. Then since $\tilde{T}$ is constructed as the union of conjugacy classes corresponding to the elements of H, every $t' \in \tilde{T}$ can be seen as a rotation and/or reflection of some point in v, and Condition (3) is satisfied. □

Example 4.4.1. *If we choose the group of rotations of the plane by setting $\tilde{R} = SO_2 \lhd O_2$, then the orbits $O_{\tilde{R}}(a)$ are circles of radius $\|a\|$. See Figure 4-1. Therefore rotation invariance is possible as long as the set $\tilde{T}$ includes translations to all points along the circle of radius a, for each element $t_a \in \tilde{T}$. A similar argument can be made for reflection invariance, since any rotation can be built out of the composition of two reflections.*

Example 4.4.2. *Analogous to the previous choice of the rotation group SO_2, we may also consider finite cyclical groups C_n describing rotations for $\theta = 2\pi/n$. In this case the construction of an appropriate set of translations $\tilde{T}$ is the same, and we include suitable conjugacy classes with respect to the group C_n.*

Proposition 4.4.2. *Assume that the input spaces $\{\mathrm{Im}(v_i)\}_{i=1}^{n-1}$ are endowed with a norm inherited from $\mathrm{Im}(v_n)$ by restriction. Then at all layers, the group of orthogonal operators O_2 is the only group of transformations to which the neural response can be invariant.*

Proof. Let R denote a group of transformations to which we would like the neural response to be invariant. If the action of a transformation $r \in R$ on elements of v_i

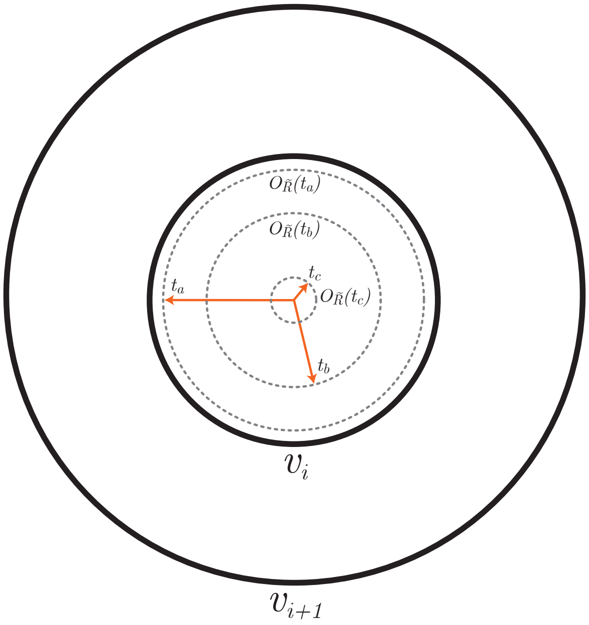


Figure 4-1: Example illustrating construction of an appropriate H. Suppose H initially contains the translations $\Gamma = \{h_a, h_b, h_c\}$. Then to be invariant to rotations, the condition on H is that H must also include translations defined by the $\tilde{R}$-orbits $O_{\tilde{R}}(t_a), O_{\tilde{R}}(t_b)$ and $O_{\tilde{R}}(t_c)$. In this example $\tilde{R} = SO_2$, and the orbits are translations to points lying on a circle in the plane.

increases the length of those elements, then Condition (2) of Definition 4.3.1 would be violated. So members of R must either decrease length or leave it unchanged. Suppose $r \in R$ decreases the length of elements on which it acts by a factor $c \in [0, 1)$, so that $\|A_r(x)\| = c\|x\|$. Condition (1) says that for every $t \in \tilde{T}$, we must be able to write $t = rt'r^{-1}$ for some $t' \in \tilde{T}$. Choose $t_v = \arg\max_{\tau \in \tilde{T}} \|A_\tau(0)\|$, the largest magnitude translation. Then $t = rt'r^{-1} \Rightarrow t' = r^{-1}t_v r = t_{r^{-1}(v)}$. But $\|A_{t'}(0)\| = c^{-1}\|v\| > \|v\| = \|A_{t_v}(0)\|$, so t' is not an element of $\tilde{T}$ and Condition (1) cannot be satisfied for this r. Therefore, we have that the action of $r \in R$ on elements of v_i, for all i, must preserve lengths. The group of transformations which preserve lengths is the orthogonal group O_2. □

The following Corollary is immediate:

Corollary 4.4.1. *The neural response cannot be scale invariant, even if K_1 is.*

Example 4.4.3. *Consider a simple* convolutional neural network *consisting of two layers, one filter at the first convolution layer, and downsampling at the second layer defined by summation over all distinct $k \times k$ blocks. The results above say that if the filter kernel is rotation invariant, the output representation (after downsampling) is invariant to global rotation of the input image. Convolution implies the choice*

$K_1(f, g) = \langle f, g\rangle_{L_2}$. We require that the convolution filter t is invariant: $t \circ r = t$ for all $r \in \mathcal{R}ot$. In this case, $K_1(f \circ r, t) = K_1(f, t \circ r^{-1}) = K_1(f, t)$, so we have invariance of the initial kernel in the sense that K_1 is always applied to an image patch and an invariant template.

4.4.2 Strings, Reflections, and Finite Groups

We next consider the case of finite length strings defined on a finite alphabet. Although the definitions in Chapter 2 applicable to 1-dimensional strings are expressed in terms of maps between sets of indices, one of the advantages group theory provides in this setting is that we need not work with permutation representations. Indeed, we may equivalently work with group elements which act on strings as abstract objects, leading to a cleaner development.

The definition of the neural response given in Chapter 2 involves translating an analysis window over the length of a given string. Clearly translations over a finite string do not constitute a group as the law of composition is not closed in this case. We will get around this difficulty by considering *closed words* formed by joining the free ends of a string. Following the case of circular data where arbitrary translations are allowed, we will then consider the original setting described in Chapter 2 in which strings are finite non-circular objects.

Taking a *geometric* standpoint sheds light on groups of transformations applicable to strings. In particular, one can interpret the operation of the translations in H as a circular shift of a string followed by truncation outside of a fixed window. The cyclic group of circular shifts of an n-string is readily seen to be isomorphic to the group of rotations of an n-sided regular polygon. Similarly, reversal of an n-string is isomorphic to reflection of an n-sided polygon, and describes a cyclic group of order two. As in Equation (4.3), we can combine rotation and reflection via a semidirect product

$$D_n \cong C_n \rtimes C_2 \tag{4.9}$$

where C_k denotes the cyclic group of order k. The resulting product group has a familiar presentation. Let t, r be the generators of the group, with r corresponding to reflection (reversal), and t corresponding to a rotation by angle $2\pi/n$ (leftward circular shift by one character). Then the group of symmetries of a closed n-string is described by the relations

$$D_n = \langle t, r \mid t^n, r_v^2, r_v t r_v t\rangle. \tag{4.10}$$

These relations can be seen as describing the ways in which an n-string can be left unchanged. The first says that circularly shifting an n-string n times gives us back the original string. The second says that reflecting twice gives back the original string, and the third says that left-shifting then reflecting is the same as reflecting and then right-shifting. In describing exhaustively the symmetries of an n-string, we have described exactly the dihedral group D_n of symmetries of an n-sided regular polygon. As manipulations of a closed n-string and an n-sided polygon are isomorphic, we will use geometric concepts and terminology to establish invariance of the neural response defined on strings with respect to reversal. In the following discussion we will abuse notation and at times denote by u and v the largest index associated with the patches u and v.

In the case of reflections of strings, r_u is quite distinct from r_v. The latter reflection, r_v, is the usual reflection of an v-sided regular polygon, whereas we would like r_u to reflect a smaller u-sided polygon. To build a group out of such operations, however, we will need to ensure that r_u and r_v both apply in the context of v-sided polygons. We extend A_{r_u} to v by defining r_u to be the composition of two operations: one which reflects the u portion of a string and leaves the rest fixed, and another which reflects the remaining $(v-u)$-substring while leaving the u-substring fixed. In this case, one will notice that r_u can be written in terms of rotations and the usual reflection r_v:

$$r_u = r_v t^{-u} = t^u r_v \, . \tag{4.11}$$

This also implies that for any $x \in T$,

$$\{rxr^{-1} \mid r \in \langle r_v \rangle\} = \{rxr^{-1} \mid r \in \langle r_v, r_u \rangle\},$$

where we have used the fact that T is abelian, and assume the relations in Equation (4.10).

We can now make an educated guess as to the form of $\tilde{T}$ by starting with Condition (1) of Definition 4.3.1 and applying the relations appearing in Equation (4.10). Given $x \in \tilde{T}$, a reasonable requirement is that there must exist an $x' \in \tilde{T}$ such that $r_v x = x' r_u$. In this case

$$x' = r_v x r_u = r_v x r_v t^{-u} = x^{-1} r_v r_v t^{-u} = x^{-1} t^{-u}, \tag{4.12}$$

where the second equality follows from Equation (4.11), and the remaining equalities follow from the relations (4.10). The following Proposition confirms that this choice of $\tilde{T}$ is compatible with the reflection subgroup of $G = D_v$, and closely parallels Proposition 4.4.1.

Proposition 4.4.3. *Let H be the set of translations associated to an arbitrary layer of the hierarchical feature map and define the injective map $\tau : H \to T$ by $h_a \mapsto t_a$, where a is a parameter characterizing the translation. Set $\Gamma = \{\tau(h) \mid h \in H\}$. Take $G = D_n \cong T \rtimes R$, with $T = C_n = \langle t \rangle$ and $R = C_2 = \{r, 1\}$. The sets*

$$\tilde{R} = R, \qquad \tilde{T} = \Gamma \cup \Gamma^{-1} t^{-u}$$

are compatible.

Proof. Equation (4.11) gives that for $x \in T$, $r_v x r_u^{-1} = r_v x t^u r_v^{-1}$. By construction,

$T \lhd G$, so for $x \in T$, $r_v x r_v^{-1} \in T$. Since xt^u is of course an element of T, we thus have that $r_v x r_u^{-1} \in T$. Equation (4.12) together with the relation $r_u = r_u^{-1}$ shows that $x^{-1}t^{-u} = r_v x r_u = r_v x r_u^{-1}$. Therefore

$$\tilde{T} = \bigcup_{x\in\Gamma} \{x, x^{-1}t^{-u}\} = \bigcup_{x\in\Gamma} \{r_v x r_u^{-1} \mid r \in \{r, 1\}\} = \bigcup_{x\in\Gamma} \{r_v x t^u r_v^{-1} \mid r \in \tilde{R}\} = \bigcup_{x\in\Gamma'} C_{\tilde{R}}(x), \tag{4.13}$$

where $\Gamma' = \Gamma t^u$. Thus $\tilde{T}$ is a seen as a union of $\tilde{R}$-orbits with $r' = r'_v = r'_u, r' \in \tilde{R}$, and we can apply Proposition 4.3.1 with Γ' to confirm that Condition (1) is satisfied.

To confirm Conditions (2) and (3), one can consider permutation representations of r_u, r_v and $t \in \tilde{T}$ acting on v. Viewed as permutations, we necessarily have that $A_{r_u}(u) = u, A_{r_u}(v) = v, A_{r_v}(v) = v$ and $A_t(u) \subset v$. □

One may also consider non-closed strings, as in Chapter 2, in which case substrings which would wrap around the edges are disallowed. However Proposition 4.4.3 in fact points to the minimum $\tilde{T}$ for reversals in this scenario as well, noticing that the set of allowed translations is the same set above but with a few illegal elements removed. If we again take length u substrings of length v strings, this reduced set of legal transformations in fact describe the symmetries of a regular $(v - u + 1)$-gon. We can thus apply Proposition 4.4.3 working with the Dihedral group $G = D_{v-u+1}$ to settle the case of non-closed strings.

4.5 Conclusion

We have shown that the tools offered by group theory can be profitably applied towards understanding invariance properties of a broad class of deep, hierarchical models. If one knows in advance the group to which a model should be invariant, then the translations which must be built into the hierarchy can be described. In the case of images, we showed that the only group to which a model in the class of interest can be invariant is the group of planar orthogonal operators.

Chapter 5

Localized Spectro-Temporal Cepstral Analysis of Speech

Portions of this chapter appeared in [17], and are Copyright ©2008 IEEE. Reprinted, with permission, from Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing. The work described here is joint with Tony Ezzat.

The work in this chapter focuses on the representation and recognition of speech using a hierarchical architecture, and is mainly empirical in nature. The results we describe evaluate experimentally important assumptions built into the hierarchical learning framework described in the preceding chapters, and is motivated by the success of existing hierarchical models of visual cortex. In particular, this work parallels the theoretical component of the thesis in that it shares the notion of a localized and layered analysis such as that occurring in the early stages of the visual and auditory cortices. It also complements the theory in the sense that underlying assumptions built into the abstract formalism are evaluated in the context of a difficult, real-world learning task.

More specifically, our speech feature analysis technique is based on a localized spectro-temporal cepstral analysis. We proceed by extracting localized 2D patches from the spectrogram and project onto a 2D discrete cosine (2D-DCT) basis. For each time frame, a speech feature vector is then formed by concatenating low-order 2D-DCT coefficients from the set of corresponding patches. We argue that our framework has significant advantages over standard one-dimensional MFCC features. In particular, we find that our features are more robust to noise, and better capture temporal modulations important for recognizing plosive sounds. We evaluate the performance of the proposed features on a TIMIT classification task in clean, pink, and babble noise conditions, and show that our feature analysis outperforms traditional features based on MFCCs.

5.1 Introduction

Most state-of-the-art speech recognition systems today use some form of MEL-scale frequency cepstral coefficients (MFCCs) as their acoustic feature representation. MFCCs are computed in three major processing steps: first, a short-time Fourier transform (STFT) is computed from a time waveform. Then, over each spectral slice, a bank of triangular filters spaced according to the MEL-frequency scale is applied. Finally, a 1-D discrete cosine transform (1D-DCT) is applied to each filtered frame, and only the first N coefficients are kept. This process effectively retains only the smooth envelope profile from each spectral slice, reduces the dimensionality of each temporal frame, and decorrelates the features.

Although MFCCs have become a mainstay of ASR systems, machines still significantly under-perform humans in both noise-free and noisy conditions [68]. In the work presented here, we turn to recent studies of the mammalian auditory cortex [10, 23, 86, 108] in an attempt to bring machine performance towards that of humans via biologically-inspired feature analyses of speech. These neurophysiological studies reveal that cortical cells in the auditory pathway have two important properties which are distinctly *not* captured by standard MFCC features, and which we will explore in this work.

Firstly, rather than being tuned to purely spectral modulations, the receptive fields of cortical cells are instead tuned to both spectral and temporal modulations. In particular, auditory cells are tuned to modulations with long temporal extent, on the order of 50-200ms [23, 108]. In contrast, MFCC features are tuned only to spectral modulations: each 1D DCT basis may be viewed as a matched filter that responds strongly when the spectral slice it is applied to contains the spectral modulation encoded by the basis. MFCC coefficients thus indicate the degree to which certain spectral modulations are present in each spectral slice. The augmentation of MFCCs with Δ and $\Delta\Delta$ features clearly incorporates more temporal information, but this is not equivalent to building a feature set with explicit tuning to particular temporal modulations (or joint spectro-temporal modulations for that matter). Furthermore, the addition of Δ and $\Delta\Delta$ features creates a temporal extent of only 30-50ms, which is still far shorter than the duration of temporal sensitivities found in cortical cells.

Secondly, the above neurophysiological studies further show that cortical cells are tuned to *localized* spectro-temporal patterns: the spectral span of auditory cortical neurons is typically 1-2 octaves [23, 108]. In contrast, MFCC features have a *global* frequency span, in the sense that the spectral modulation “templates” being matched to the slice span the entire frequency range. One immediate disadvantage of the global nature of MFCCs is that it reduces noise-robustness: addition of noise in a small subband affects the entire representation.

Motivated by these findings, we propose a new speech feature representation which is localized in the time-frequency plane, and is explicitly tuned to spectro-temporal modulations: we extract small overlapping 2D spectro-temporal patches from the spectrogram, project those patches onto a 2D discrete cosine basis, and retain only the low-order 2D-DCT coefficients. The 2D-DCT basis forms a biologically-plausible matched filter set with the explicit joint spectro-temporal tuning we seek. Furthermore, by localizing the representation of the spectral envelope, we develop a feature set that is robust to additive noise.

In Section 5.3, we describe in detail the localized spectro-temporal analysis framework and provide examples illustrating the structure that the proposed patch-based 2D-DCT features capture. Then in Section 5.4, we describe a specific application of our method to phonetic classification on the TIMIT corpus [59] in clean conditions. Section 5.5 follows on the clean experiments to present classification performance in pink and babble noise conditions. In both cases we compare the 2D-DCT features to two strong sets of MFCC-based baseline features. In Section 5.6, we propose several possible extensions to our analysis framework, and present preliminary classification error rates in a multiscale analysis setting. Finally, in Section 5.7 we discuss the experimental results, place our work within the larger ASR context, and conclude with a list of future directions and open problems.

5.2 Background

A large number of researchers have recently explored novel speech feature representations in an effort to improve the performance of speech recognizers, but to the best of our knowledge none of these features have combined localization, sensitivity to spectro-temporal modulations, and low dimensionality.

Hermansky [47, 106] and Bourlard [15] have used localized sub-band features for speech recognition, but their features were purely spectral and failed to capture temporal information. Subsequently, through their TRAP-TANDEM framework, Hermansky, Morgan and collaborators [48, 47, 21] explored the use of long but thin temporal slices of critical-band energies for recognition, however these features lack joint spectro-temporal sensitivity. Kajarekar et al. [55] found that both spectral and temporal analyses performed in sequential order outperformed joint spectro-temporal features within a linear discriminant framework, however we have found joint 2D-DCT features to outperform combinations of purely spectral or temporal features. Atlas and Shamma [5, 107] also explored temporal modulation sensitivity by computing a 1D-FFT of the critical band energies from a spectrogram. These features too lack *joint* and *localized* spectro-temporal modulation sensitivity. Zhu and Alwan [121] use the 2D-DCT to compress a block of MFCCs, however this approach still suffers from the shortcomings of global MFCCs. Kitamura et al. [56], take a *global* 2D-FFT of a MEL-scale spectrogram, and discard various low-frequency bands from the resulting magnitude. This approach does not provide any joint spectro-temporal localization, and cannot be interpreted as capturing meaningful configurations of specific spectro-temporal modulation patterns. It is the localized analysis in our method, and the fact that we seek to encode spatial configurations of important spectro-temporal modulations, that critically differentiates our approach from much of the previous work.

A large fraction of the noise-robust speech recognition literature has traditionally centered on either front-end post-processing of standard features [73, 22, 32] or modifying recognizer model parameters to compensate for train/test noise mismatch [33], rather than the design of inherently noise-robust features. (see [104, 103] for a detailed review). Povey and collaborators [81] have however described a framework for estimating robust features from distorted cepstral coefficients that are typically added to the standard features, while other authors have proposed variations on standard MFCCs that provide additional robustness in mismatched conditions. Cui et al. [30] apply peak isolation and RASTA filtering to MFCCs in speech (versus non-speech) segments, while Burges et al. [20] attempt to learn the dimensionality reduction step from the data. In all cases, however, the robust features are neither localized nor spectro-temporal.

Perhaps the closest work to ours is that of Shamma and colleagues [23, 72], and the work of Kleinschmidt, Gelbart, and collaborators [57, 58]. In [23] the authors apply localized complex filters that produce both magnitude and phase information for the purpose of speech vs. non-speech detection. The latter group applied data-optimized Gabor filters to blocks of MEL-scale spectra with 23 frequency bins, and then present ASR results when Gabor features augment other features. Our work builds on upon both of these efforts, and demonstrates an important point which we believe has not been made strongly enough in these previous works: that a simple set of localized 2D-DCT features (in this case, "bar-like" detectors faithful to the auditory neuroscience) is on its own powerful enough to achieve state-of-the-art performance on a difficult phonetic discrimination task.

5.3 2-D Cepstral Analysis of Speech

5.3.1 Spectro-Temporal Patch Extraction

The first step of our 2D cepstral speech analysis technique is to convert a given speech signal into a narrow-band spectrogram $S(f, t)$. Each utterance is first STFT-analyzed in the usual manner using a Hamming window with an associated time extent, frame rate, and zero-padding factor (we provide exact values for these parameters in Section 5.4). Additionally, we retain only the log magnitude of the resulting STFT, and normalize it to have zero mean and unit variance. Note that we limit our analysis to a linear frequency scale, deferring MEL-scale (logarithmic) frequency analysis to future work.

Then, at every grid point (i, j) in the spectrogram, we extract a patch $P_{ij}(f, t)$ of size df and width dt. The height df and width dt of the local patch are important analysis parameters: they must be large enough to be able to resolve the underlying spectro-temporal components in the patch, but small enough so that the underlying

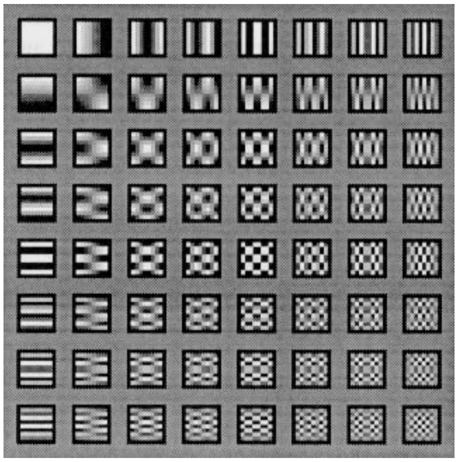


Figure 5-1: The 16 DCT bases for a patch of size 8x8 pixels. The first row of bases capture temporal modulations, while the first column captures spectral modulations. Checkerboard basis functions help model spectro-temporal noise.

signal is stationary. Additional analysis parameters are the 2D window hop-sizes in time Δi and frequency Δj, which control the degree of overlap between neighboring patches. Finally, we pre-multiply the patch with a 2D Hamming window $W_H(f,t)$ in order to reduce border effects during subsequent patch processing.

5.3.2 2D Discrete Cosine Transform and Coefficient Truncation

After patch extraction, a 2-D discrete cosine transform (2D-DCT) is applied to each windowed patch $P(f,t)$ to produce a set of DCT coefficients $B(\Omega,\omega)$. The definition of the 2-D DCT for an input patch $P(f,t)$ of size F-by-T is given by

$$B(\Omega,\omega) = A\sum_{f=0}^{F-1}\sum_{t=0}^{T-1} P(f,t)\cos\frac{\pi(2f+1)\Omega}{2F}\cos\frac{\pi(2t+1)\omega}{2T} \tag{5.1}$$

where $0 \leq \Omega \leq F-1$, $0 \leq \omega \leq T-1$, and A is a scaling factor whose value we omit for simplicity.

Shown in Figure 5-1 are 16 representative DCT bases for a patch of size 8x8 pixels. The basis functions in the first column respond to “horizontal” speech phenomena such as harmonics and formants. The basis functions in the first row respond to “vertical” speech phenomena such as plosive edges. Finally, the remaining checkerboard bases capture spectro-temporal noise patterns and contribute to cancelation effects that facilitate energy localization within the patch.

The 2D-DCT projects each patch onto a set of orthogonal, separable cosine basis functions that respond to “horizontal” speech phenomena such as harmonics and

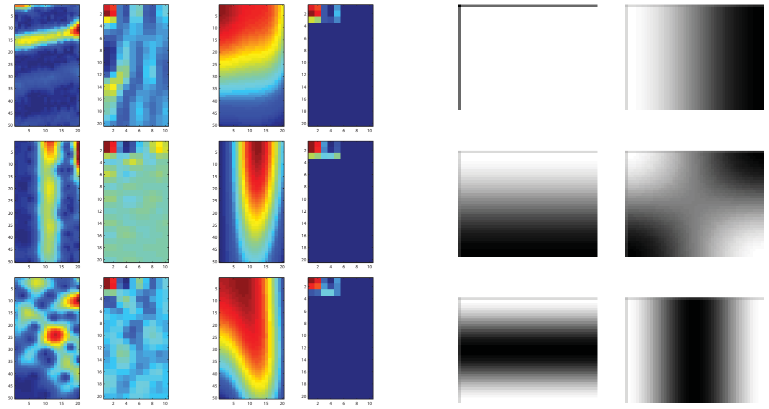


Figure 5-2: Left two columns: Original spectrogram patches, followed by the corresponding 2D-DCT. Middle 2 columns: Patches reconstructed from low-order DCT coefficients followed by the low-order DCT coefficients retained for reconstruction. Last 2 columns: Retained low-order 2D-DCT basis functions.

formants, "vertical" speech phenomena such as plosive edges, and more complex spectro-temporal noise patterns. In the rightmost two columns of Figure 5-2 we show the six low-order 2D-DCT basis functions used in our analysis. The top-left basis is everywhere uniform.

Shown in Figure 5-2 in the first column are representative harmonic (top), plosive (middle), and noise (bottom) patches from a spectrogram, along with their respective 2D-DCT coefficients in the second column. As expected, horizontal harmonic edges in a patch strongly activate coefficients in the first column of the corresponding DCT, and vertical plosive phenomena activate DCT coefficients in the first row. Noise phenomena, with more high frequency components than the previous two examples, has energy that is distributed among most of the DCT coefficients.

The last step of our analysis consists of truncating the 2D-DCT and retaining only the low-order coefficients for each patch. The effect of doing this is also shown in Figure 5-2: original patches in the first column are reconstructed in the third column using only the low-order 3×5 block of DCT coefficients (4th column). Keeping only the low-order DCT coefficients is equivalent to representing each patch with a *smooth spectro-temporal envelope*. We further illustrate this concept in Figure 5-3, where the original spectrogram displayed on the left is reconstructed on the right from low-order patch 2D-DCT coefficients. The individually reconstructed patches are overlap-added together to assemble the full spectrogram. In this example, we have used analysis windows of size 780Hz by 57ms shifted in steps of 156Hz in frequency and 10ms in time.

Smoothing via two-dimensional DCT truncation is analogous to smoothing via truncation of 1D-DCT coefficients in MFCC analysis of speech. However, because the DCT in the proposed methodology is *two-dimensional* and *localized* in the spectro-temporal plane, the analysis still retains important information about the spectro-

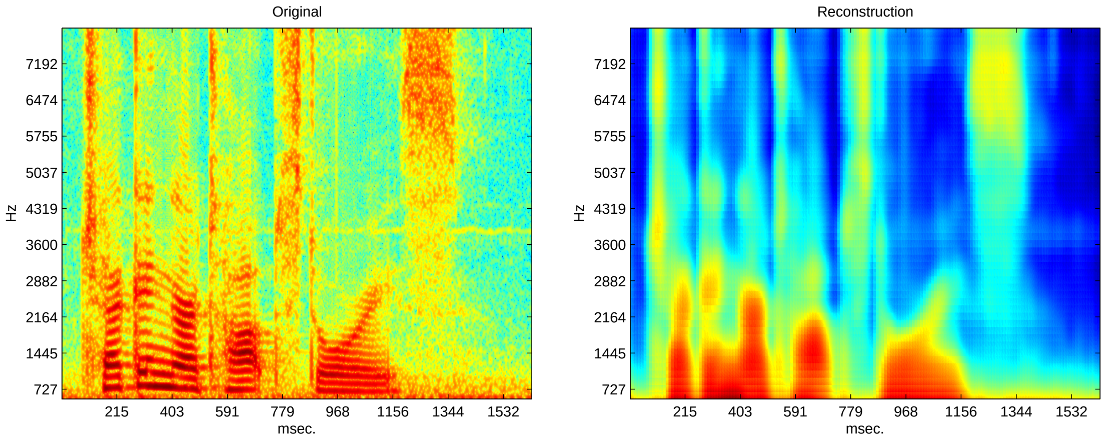


Figure 5-3: Left: Original spectrogram. Right: Resulting spectrogram after retaining only low-order DCT coefficients per patch, and applying overlap-add reconstruction.

temporal evolution of the envelope. For this reason, 2D-DCT features are particularly well suited to modeling plosive phonetic sounds. Additionally, because the envelope representation is localized, the 2D-DCT features are more robust to noise.

5.4 Clean Speech TIMIT Experiments

The above 2D-DCT analysis was applied towards extracting features for phonetic classification on the TIMIT corpus [59], and compared to MFCC-based features used by Clarkson and Moreno [25], and the best single set of features proposed by Halberstadt and Glass in [45]. The latter feature set is the best baseline that we are aware of. We divided TIMIT into standard train and test sets following the convention in [90, 45]. The 2D-DCT analysis is performed using both wideband and narrowband spectrograms for comparison.

In the following section we describe the TIMIT classification task, followed by a description of the particular instance of the proposed spectro-temporal analysis method that we have used for phonetic classification. We then describe the two sets of baseline comparison features, the classification framework, and finally, present a comparison of the classification results for stops, vowels, and on the full task (all phonemes).

5.4.1 TIMIT Corpus

We divided TIMIT into standard train and test sets following the convention in [90, 45], resulting in a training set consisting of 140,225 examples drawn from 462 speakers and 3,696 utterances, and a test set consisting of 7215 examples drawn from 24 speakers and 192 utterances. After ignoring glottal stops ('q' tokens), the 60 remaining phonetic classes are later mapped to 39 categories after training but before scoring, also following standard practices for this task.

5.4.2 Spectrogram Pre-processing

TIMIT utterances are first normalized and subjected to a pre-emphasis filter. We then compute spectrograms using 32 sample (2ms) hops, 1024-point FFTs and 300 sample (18.75ms) Hamming windows or 150 sample (9.375ms) windows for narrow- and wide-band conditions respectively. We then take the log-magnitude of the resulting spectrum and normalize to give a global utterance mean of zero and unit variance. Utterances are broken up in time according to the labeled phonetic boundaries and enlarged by an additional 30ms on either side of each phoneme so as to include coarticulatory phenomena. Each resulting independent phoneme is then truncated at 6.23kHz (400 frequency bins), while a copy of the bottom 25 low-frequency bins is reflected, for all time, about the 0Hz edge and appended. Because we later apply local 2D-DCTs to Hamming windowed regions of the spectrogram, reflection is done to avoid artificially down-weighting low frequency bins near the edge of the image.

5.4.3 2D-DCT Patch Processing and Coefficient Truncation

We first compute a sliding localized two-dimensional DCT over the phoneme's spectrogram. While many reasonable window and step sizes exist, we have found that, for the narrowband STFT parameters above, good results are obtained with 780Hz by 56.75ms (or 50 by 20 bin) Hamming windowed 2D analysis regions with a 390Hz (25-bin) frequency step-size and a 4ms (2-bin) time step-size. For wideband STFT conditions, good results are obtained with 623Hz by 107.375ms (or 40 by 50 bin) Hamming windowed 2D analysis regions with identical step sizes in time and frequency as in the narrowband case. The 2D-DCT is computed with 2x oversampling in both time and frequency. We have found that performance does not critically depend on the precise window and step size choices above. To avoid implicit overfitting, evaluation of performance for different parameter choices was done using the TIMIT development set proposed by [45], while the final evaluations shown below were done on the core test set.

For each 2D analysis region, we save only the 6 lowest-order 2D-DCT coefficients corresponding to the upper left 3×3 triangle in the DCT image. These coefficients collectively encode only the patch's DC offset, two low spatial frequency horizonal and vertical basis components, and one "checkerboard" basis component. Saving six coefficients per patch at the above resolution smooths out any remaining harmonic structure, leaving only the spectro-temporal envelope. This behavior coincides with the goal of MFCCs and other common low-dimensional representations: to eliminate most within-class variation. Operating on patches with limited time-frequency extent, however, allows one to smooth away irrelevant variation while preserving discrimina-

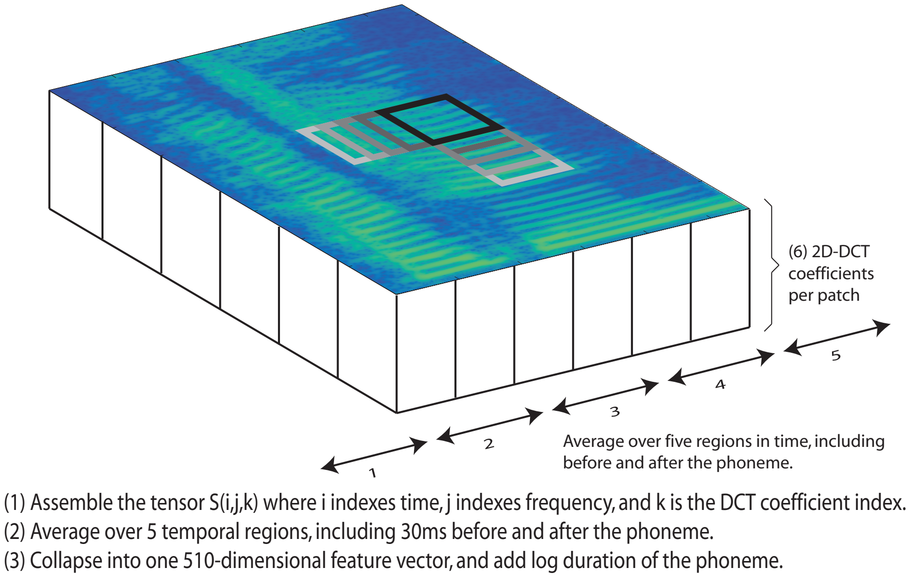

(6) 2D-DCT
coefficients
per patch
5
4
3
2
1
Average over five regions in time, including
before and after the phoneme.
(1) Assemble the tensor S(i,j,k) where i indexes time, j indexes frequency, and k is the DCT coefficient index.
(2) Average over 5 temporal regions, including 30ms before and after the phoneme.
(3) Collapse into one 510-dimensional feature vector, and add log duration of the phoneme.

Figure 5-4: Feature vector construction, after [45].

tory macro structure that is lost when applying any global smoothing technique (as is done with MFCCs). In the case of stops, for example, preserving the overall distribution of energy in the time-frequency plane is critical for classification.

5.4.4 Feature Vector Construction

The previous step provides a vector of 6 features for each patch taken from the phoneme. We modify the approach of [45] in order to compute a fixed length feature vector from the variable number of 2D-DCT coefficients representing a given phoneme; this particular construction was found (in [45]) to work well for the MFCC-based features computed therein. If the 6-dimensional vectors are collected and arranged in a relative order corresponding to the time-frequency centers of the respective analysis windows, we are left with a 3D matrix of coefficients per phoneme example: $S(i,j,k)$ where i indexes time, j indexes frequency, and k is the DCT coefficient index. The number of time bins will of course vary across phonemes. We therefore divide up the time axis of the 3D matrix of coefficients into five segments, and average over time within each segment. See Figure 5-4. The time bins corresponding to the 30ms of additional signal added before and after the phoneme give the first and last segments, while the bins falling within the phoneme itself are divided up in 3:4:3 proportion to give the middle 3 segments. All coefficients across the five averaged segments (contributing 17 patches $\times$ 6 coefficients $=$ 102 features each) are then pooled and concatenated into a single 510-dimensional vector. Lastly, the log-duration of the phoneme is added to give the final 511-element feature vector. Prior to classification, the training and test datasets are whitened with the principal components derived from the training set.

5.4.5 MFCC-Based Baseline Comparison Features

We compare our features to two other TIMIT MFCC-based baseline feature sets: the "S2" feature-set proposed by Halberstadt & Glass [45] and the features described by Clarkson & Moreno [25]. We will refer to these feature sets as "HA" and "CM" respectively. In both cases, the resulting datasets are whitened with PCA.

The HA feature set is constructed by computing 12 MFCCs from each frame of the spectrogram (30ms Hamming windowed segments every 5ms). Temporal averages are taken over the five non-overlapping segments described above to obtain a fixed-length feature vector. A log-duration feature is again added, resulting in 61-dimensional feature vectors. The HA feature set has been used to get the best single-classifier TIMIT result [90] that we are aware of.

Clarkson & Moreno's features are similar to Halberstadt's: 13 MFCCs are computed for each spectrogram frame (25.5ms Hamming windows every 10ms). However, Δ and $\Delta\Delta$ features are also computed, giving classical 39-dimensional feature vectors for each frame. The time axis is again divided up into five segments, but the two regions including spectra before and after the phoneme are 40ms wide and are centered at the beginning and end of the phoneme. A log-duration feature is also added, resulting in 196 dimensional feature vectors.

5.4.6 Classification Framework

All-vs-all (AVA) classification with linear regularized least-squares (RLS) classifiers [88] was performed on the resulting datasets. Linear RLS classification allows for efficient selection of the regularization parameter via leave-one-out cross validation. We have found empirically that AVA multiclass training consistently outperforms one-vs-all for the TIMIT task. We include for comparison results on the full TIMIT task using second-order polynomial SVMs with 5-fold cross-validated selection of the regularization parameter. Training time for the nonlinear SVMs was approximately an order of magnitude slower than linear RLS. Our ultimate goal, however, is to illustrate the strength of localized spectro-temporal features even in the absence of excessive tuning of the classifier stage.

5.4.7 Clean Speech Results

In Table 5.1 we show linear RLS classification error rates for the proposed localized 2D-DCT features (for both narrow- "NB" and wide-band "WB" spectrograms) as compared to the two sets of baseline features described above. We show results on the full TIMIT task, and additionally, when training and testing on subsets consisting

Features	Stops	Vowels	All	Dims
CM	29.66	37.59	28.30	196
HA	27.91	37.80	25.60	61
2D-DCT-NB	23.53	37.33	24.93	511
2D-DCT-WB	25.53	36.69	24.37	511
2D-DCT-NB/SVM2			21.37	

Table 5.1: Percent error rates for the three sets of features when training/testing on stops only, vowels only, or on all phonemes from clean utterances. Our features are denoted “2D-DCT”. Dimensionality of the feature vectors are given in the last column. “NB” and “WB” denote narrowband and wideband conditions respectively.

of just the vowels or just the stops. The full task consists of training on 60 classes (140225 train, 7215 test examples) and then mapping to 39 classes for scoring, while the stops task consists of 6 phonetic classes (16134 train, 799 test examples) and the vowel task consists of 20 classes (45572 train, 2341 test examples). No post-mapping is done prior to scoring in the case of the the vowels and stops experiments.

In all cases, the localized 2D-DCT features outperform the MFCC-based baseline features. Wideband spectrograms with longer temporal analysis extents are seen to give better performance than narrowband spectrograms with shorter extents in all experiments excepting the stops only evaluation. However in the case of stops in particular, the 2D-DCT features provide substantial improvement over traditional MFCCs. Because the DCT analysis is spectro-temporal and includes explicit bases encoding vertical and horizontal spatial gratings, the 2D-DCT features capture the strong vertical “edges” present in stops and other plosives. The last row of Table 5.1 shows performance when using nonlinear SVM classifiers, and confirms that 2D-DCT features still exhibit the reduction in error that one would expect when moving to more complex classifiers.

5.5 Noisy Speech TIMIT Experiments

The classification performance of localized 2D-DCT features was also evaluated in the presence of both pink and babble noise. We describe the construction of the noise-contaminated datasets, present phonetic classification results, and in pink-noise conditions provide a comparison with HA [45] and CM [25] features (described in Section 5.4.5). The HA error rates were originally presented in Rifkin et al. [90], and are reproduced here. The authors of [90], do not provide performance in babble-noise. In all experiments, *training is done on clean speech while testing is done on noisy speech.*

5.5.1 Noisy Dataset Construction

Pink noise corrupted TIMIT utterances at 20dB,10dB, and 0dB SNR were obtained from the authors of [90] so that experiments could be performed under the exact same noise mixing conditions. In [90], a single 235 second segment of noise from the NOISEX-92 dataset was used to artificially corrupt the test set speech. Random contiguous snippets from this master segment were added to each utterance with an amplitude chosen to satisfy the desired global SNR. Because TIMIT recordings do not include long pauses, local SNR matching is largely unnecessary. Similarly, we constructed babble-noise corrupted TIMIT utterances by following the same procedure while using a 235 second segment of babble-noise, also from the NOISEX-92 dataset. In both cases, spectra and features were then extracted from the noisy utterances in a manner identical to that described in Section 5.4.

5.5.2 Noisy Speech Results

In Table 5.2 we show percent error rates for the full TIMIT phonetic classification task, comparing the HA and CM feature sets to the proposed localized 2D-DCT features when using linear RLS classifiers with an all-vs-all multiclass scheme (denoted "RLS1"). A second order polynomial RLS classifier (denoted "HA-RLS2") is also given in [90], and we include those results here for comparison. The first four feature set/classifier combinations involve pink-noise corrupted utterances.

In the presence of even weak pink noise (e.g. 20dB SNR), 2D-DCT features with simple linear classifiers outperform HA features. As the signal to noise ratio is decreased, the performance advantage remains significant. As shown in the sixth and seventh row of Table 5.2, we observe a relative reduction in error of approximately 10-25% when using localized 2D-DCT features over the MFCC-based HA features with an identical classification stage. Despite the fact that the CM feature set combines MFCCs with traditional Δ and $\Delta\Delta$ features, both the DCT and HA features far outperform Clarkson's CM features. In the last two rows of Table 5.2 (marked with a "B"), we show classification error in babble noise. Although babble noise is usually considered a more challenging condition for speech recognizers, for this particular task we observe only a modest increase in error above the error in pink-noise when using the proposed 2D-DCT features. The longer temporal extents of the patches and 2D-DCT templates in the "WB" case are also seen to give improved performance in babble noise.

In clean conditions, the second-order polynomial classifier with HA features (marked "HA-RLS2") outperforms linear classifiers with any of the feature sets. However this is not entirely surprising; the authors of [90] explicitly lift the 61 original HA features into 1830-dimensional second-order feature vectors. In the presence of noise, however, the 2D-DCT features outperform all the other classifier/feature set combinations. In [90], a comparison is made between RLS1, RLS2, and a GMM based classifier from [45], all trained on HA features, and it is argued that RLS classifiers

Features	Clean	20dB	10dB	0dB
CM-RLS1	28.30	61.68	79.67	91.78
HA-RLS1	25.60	41.34	63.12	80.03
HA-RLS2	20.93	33.79	57.42	77.80
2D-DCT-NB/RLS1	24.93	32.53	48.16	71.93
2D-DCT-WB/RLS1	24.37	32.36	47.79	72.75
1-(2DDCT-NB/HA) (RLS1)	2.62	21.31	23.71	10.12
1-(2DDCT-WB/HA) (RLS1)	4.80	21.72	24.29	9.10
2D-DCT-NB/RLS1 (B)	24.93	38.99	59.76	77.28
2D-DCT-WB/RLS1 (B)	24.37	37.73	57.30	75.05

Table 5.2: Train clean, test noisy experiments: Percent error rates on the full TIMIT test set for several signal-to-noise ratios and feature sets. The final two rows, marked with a "B", gives error rates in babble-noise; all other experiments involve pink-noise.

give excellent noise-robustness over generative models. We note that because every experiment shown in Table 5.2 utilized RLS classifiers, the additional noise-robustness is due entirely to the feature set.

5.6 Extensions

Several immediate extensions of the above 2D-DCT analysis are possible for improving classification and recognition accuracy. We provide a brief look at one promising possibility: embedding the localized 2D-DCT analysis in a multi-scale framework.

5.6.1 Multiscale 2D-DCT

The 2D-DCT is, ultimately, a dimensionality reduction step applied to a localized time-frequency analysis region. The analysis window, or alternatively, the spectrogram, can be resized so as to encode discriminative features and latent structure existing at different fundamental scales.

We present preliminary classification experiments in which we have kept the spectrogram resolution fixed while varying the size of the analysis window. The number of DCT coefficients retained remains constant, regardless of the size of the analysis window, so that the image patch is always projected onto scaled versions of the same 2D-DCT bases. For the experiments that follow, we have defined three scales. "Scale 1" refers to the canonical scale mentioned in Section 5.4: 780Hz by 56.75ms windows, with step sizes 390Hz and 4ms in frequency and time respectively. "Scale 2" uses a window width and step size in frequency twice that of Scale 1, and a window width 1.5 times as wide as Scale 1 in time, with the same time step size (1560Hz by 76.75ms with 780Hz and 4ms steps). "Scale 3" uses a window width and step size in frequency

Features	Scale 1	Scale 2	Scale 3
Scale 1	24.93	24.53	24.32
Scale 2		25.41	24.50
Scale 3			26.92

Features	Error
Scale 1+2+3	24.13

Table 5.3: Multiscale experiments: Percent error rates on the full, clean-speech TIMIT task when using features derived from different image scales. Entry position in the top three rows denotes that feature vectors were a concatenation of the corresponding row and column scales. The bottom row gives the error when all three scales are combined.

twice that of Scale 2, and a window width 1.3 times as wide as Scale 2 in time, with the same time step size (3120Hz by 96.75ms windows, with 1560Hz and 4ms steps). The final dimensionalities of the feature vectors were 511, 271, and 151 for scales 1 through 3 respectively.

In Table 5.3 we give error rates on the full, clean-speech TIMIT task described in Section 5.4 when using features from the three scales. The first three rows show error rates when features from the corresponding row/column pairs of scales were concatenated, while the final row gives the error-rate with features from all three scales combined. In all cases, speech waveforms, spectra and feature vectors were computed as described in section 5.4, with the exception that both the size of the 2D-DCT analysis windows and the step-sizes were varied.

It can be seen that, each time lower-resolution scales are combined with higher ones, the total error on the test set decreases by 0.4%. This corresponds to 29 additional correct classifications, every time a scale is added. While an additional 0.8% may seem small, we note that at 24 or 25% error, additional improvement for TIMIT is typically hard to come by: most recent results on this dataset are within 1% of each other. It should also be noted that many of the class confusions at this point are between ambiguous instances of phonemes easily confused by human listeners.

5.7 Discussion

The biologically inspired feature analysis presented in this Chapter consisted of two main steps: (1) Extraction of localized spectro-temporal patches, and (2) low-dimensional spectro-temporal tuning using the 2D-DCT. A localized encoding and extraction of structure in the time-frequency plane faithfully preserves the general distribution of energy, and retains critical discriminatory information. In Table 5.1 we showed that discrimination among phonemes with strong temporal modulation, such as plosives and stop consonants, was better with local 2D-DCT features than with the two sets of "global" MFCC-based baseline features. The elimination of within-class variability was controlled by the number of DCT coefficients retained, leading to a localized smoothing effect. Traditional cepstral analysis, while admittedly lower dimensional in many cases than the proposed features, tends to over-smooth in frequency and ignore important dynamics of the spectro-temporal envelope.

In pink-noise corrupted speech, local 2D-DCT features provide substantial additional noise-robustness beyond the baseline MFCC features as measured by classification accuracy on the TIMIT corpus. We also found that the localized 2D-DCT analysis outperforms classical MFCCs augmented with delta and acceleration features. Although this feature set is not the strongest of the two baselines, the comparison shows that even features incorporating more temporal information per frame is not sufficient; both time and frequency localization is necessary. On the whole, the phoneme classification experiments presented above show that the method is viable, outperforming a state-of-the-art baseline in clean and noisy conditions.

In the case of phonetic classification, harmonics are deliberately smoothed out. Speaker recognition applications, however, might rely on detailed harmonic structure responsible for defining the perceptual quality of a given speaker. Lower-resolution analysis windows and more DCT coefficients could be chosen in such cases.

5.7.1 Future Work

While phonetic classification gives some indication of the capability of the features presented in this Chapter, we acknowledge that further experimentation with a recognition task would provide a more meaningful assessment for the ASR community. In our analysis, we have worked with linear spectrograms. The effect of MEL-scale analysis or logarithmic warping, however, is not yet known within the context of 2D-DCT features. We believe it would be profitable to more carefully investigate other multi-resolution versions of the core notion of a localized spectro-temporal analysis, as well as the possibility of using an iterated hierarchical analysis framework [98], more in line with the primary and belt auditory areas of the human brain.

As mentioned earlier, the ability of human listeners to understand speech in the presence of noise and/or other competing speakers far surpasses that of any machine recognition system. At the same time, recent advances in our understanding of the mammalian auditory pathway [23, 86, 10] suggest novel, biologically-inspired approaches to automatic speech recognition that we expect will revitalize the field and push machine performance out of the current local minimum. Two overarching design themes found in both visual and auditory areas of the neocortex are (1) localized spectro-temporal/visuo-spatial receptive fields, and (2) multi-layer hierarchical analysis: the work described here has mainly attempted to incorporate ideas from the former, and has investigated only in a limited sense the latter.

Chapter 6

Speech Reconstruction from STFT Magnitude Spectra

Portions of this chapter appeared in [16].

In this chapter, we present an algorithm for reconstructing a time-domain signal from the magnitude of a short-time Fourier transform (STFT). In contrast to existing algorithms based on alternating projections, we offer a novel approach involving numerical root-finding combined with explicit smoothness assumptions. Our technique produces high-quality reconstructions that have lower signal-to-noise ratios when compared to other existing algorithms. If there is little redundancy in the given STFT, in particular, the algorithm can produce signals which also sound significantly better perceptually, as compared to existing work. Our algorithm was recently shown to compare favorably with other techniques in an independent, detailed comparison [53].

6.1 Introduction

Reconstruction of a time-domain signal from only the magnitude of the short-time Fourier transform (STFT) is a common problem in speech and signal processing. Many applications, including time-scale modification, speech morphing, and spectral signal enhancement involve manipulating the STFT magnitude, but do not clearly specify how to adjust the phase component of the STFT in order to invert back into the time domain. Indeed, for many STFT magnitude modifications, a valid inverse of the STFT does not exist and a reasonable guess must be made instead.

In this Chapter, we present an algorithm for reconstruction of a time-domain signal from the STFT magnitude, modified or otherwise. In contrast to existing algorithms based on alternating projections, our technique applies numerical root-finding combined with explicit smoothness assumptions to give high-quality reconstructions.

We have found that imposing smoothness at several stages of the algorithm is the critical component responsible for estimating good signals. Formulating the reconstruction problem in terms of non-linear systems of equations serves as a convenient vehicle for the inclusion of smoothness constraints in a straightforward manner. Our method produces results that appear to be perceptually superior to the algorithms due to Griffin and Lim [44] and Achan et al. [1], particularly when there is little overlap between STFT analysis windows.

In section 6.2 we give an overview of the signal reconstruction problem, and in section 6.3 we introduce the root-finding framework we have used to find solutions to this problem. Section 6.4 presents the smoothness constraints we have chosen to impose, followed by a description of the algorithm itself. In section 6.5 we compare the performance of our technique to Griffin and Lim's method over a range of STFT window overlaps. Finally, in section 6.6 we offer concluding remarks.

6.2 Overview of the Phaseless Signal Reconstruction Problem

If the zeros of the Z-transform of a signal lie either entirely inside or outside the unit circle, then the signal's phase may be uniquely related to its magnitude via the Hilbert transform [82]. In the case of finite speech or music signals, however, such a condition on the zeros does not ordinarily hold. Under some conditions, mixed-phase signals can be recovered to within a scale factor from only magnitude or phase [46], and can be uniquely specified from the signed-magnitude [51]. But the conditions required in [46] are restrictive, while retaining any phase-information, albeit even a single bit, is not possible for many common spectrogram modifications.

In this work, we will focus on reconstruction from magnitude spectra only. Generally, we would like to take a signal, manipulate its magnitude, and from the modified spectra be able to estimate the best possible corresponding time-domain sequence. In the absence of any modifications, we would hope to retrieve the original time-domain signal from the magnitude. If only the Discrete Fourier Transform (DFT) magnitude of a signal is provided, then we must make additional a priori assumptions in order to guess the corresponding signal. This is a common problem in several fields, such as x-ray crystallography, electron diffraction, and remote sensing [37]. If, however, we work with the short-time Fourier transform (STFT), accurate reconstruction is often possible without *a priori* assumptions or constraints. Given a suitable length N windowing function $w(n)$, we can define the STFT by sliding the signal $x(n)$ through the window and taking the K-point DFT:

$$S(\omega_k, \ell) = \sum_{n=0}^{N-1} x(n+\ell)w(n)e^{-j\omega_k n} \tag{6.1}$$

where the DFT frequency bins are $\omega_k = \frac{2\pi k}{NT}, k = 0, \ldots, K-1$ given sampling rate $f_s = 1/T$. Because both the magnitude $|S(\omega_k, \ell)|$ and phase $e^{j\phi(\omega_k, \ell)}$ of the STFT contain information about the amplitude and phase of the original signal, throwing away the STFT phase does not mean that we have entirely eliminated the original phase of $x(n)$ [26].

Several algorithms have been proposed to estimate a signal from the STFT magnitude. Achan et al. [1] introduced a generative approach for speech signals that infers a time-domain signal from a model trained on a specific speaker or class of speakers. Griffin and Lim [44] apply an iterative technique similar in spirit to an earlier algorithm advanced by Fienup [37]. While it is difficult to analyze the convergence and uniqueness properties of Fienup's algorithm, Griffin and Lim's approach employs alternating convex projections between the time-domain and the STFT domain that have been shown to monotonically decrease the squared error between the given STFT magnitude and the magnitude of an estimated time-domain signal. In the process, the algorithm thus produces an estimate of the STFT phase. Nawab et al. [76] proposed a sequential algorithm which reconstructs a signal from its spectral magnitude by extrapolating from the autocorrelation functions of each short-time segment, however the approach places sparseness restrictions on the signal and requires that the first h samples of the signal be known, where h is the STFT window hop size. The algorithm presented herein requires neither samples of the signal to be reconstructed, nor does it place constraints on the number of consecutive zeros that can appear in the reconstruction.

6.3 Signal Reconstruction as a Root-Finding Problem

Griffin and Lim's algorithm attempts to estimate a signal that is consistent with a given spectrogram by inverting the full STFT at each iteration. Alternatively, we can analyze consistency on a column-wise basis, where the spectrogram $|S(\omega_k, \ell)|$ is viewed as a matrix with frequency spanning the rows, and time the columns. Given a single column ℓ_0 from the magnitude of the STFT, we wish to determine the segment of signal $\tilde{x}(n) = x(n + \ell_0)w(n)$ that satisfies the system of equations given by (6.1):

$$|S(\omega_k, \ell_0)| = \left| \sum_{n=0}^{N-1} \tilde{x}(n) e^{-j\omega_k n} \right|, \quad k = 0, \ldots, K-1. \tag{6.2}$$

In the discussion that follows, we will abbreviate the above system with the notation $|F\tilde{x}| = m$, where F is the $K \times N$ Fourier matrix, and $m \geq 0$ is the given spectrogram column. Note that although (6.2) appears to be a system of K equations in N unknowns as it is written, the Fourier magnitude is an even-symmetric function because we allow only real-valued time-domain signals. Thus we really only have $K/2$

linearly independent equations, and 2x oversampling in the DFT is needed to make the system square. In practice we set $K = 2N$ when computing the original STFT, and solve for $\tilde{x}$ using only half of the desired magnitude vector m and a truncated Fourier matrix F. Finally, if we rearrange (6.2) to get $G(\tilde{x}) \equiv |F\tilde{x}| - m = 0$, $\tilde{x}$ is seen as a root of the function $G : \mathbb{R}^N \to \mathbb{R}^N$ so that estimating the signal is equivalent to solving a numerical root-finding problem.

It should be noted, however, that there are almost always an infinite number of possible roots $\tilde{x}$ satisfying $|F\tilde{x}| - m = 0$, since we can at best match just the magnitude spectra m. Writing $F\tilde{x} = Dm$ in terms of the phasor matrix $D = \text{diag}(e^{j\phi(\omega_k)})$, the phases $\phi(\omega_k)$ need only satisfy the condition $\text{Im}\{F^{-1}Dm\} = 0$. Which root the iteration actually returns will strongly depend on the initial condition $\tilde{x}_0$.

6.3.1 Solution of non-square systems of nonlinear equations

As we will discuss below, our algorithm involves solving for only a *subset* of the samples in a segment of the signal, while holding the remaining points fixed. One way to solve a system of p nonlinear equations in q unknowns when $p > q$ is to formulate the task as a locally linear least-squares problem. In particular, given a system of equations $f(x) = 0$, suppose that we choose the objective function $\frac{1}{2}\|f(x)\|_2^2$, and linearize $f(x)$ via a Taylor expansion about the point x_k. Defining the Jacobian $J_{ij}(x) = \frac{\partial f_i}{\partial x_j}$, we have

$$\tilde{f}(x) = f(x_k) + J(x_k)(x - x_k). \tag{6.3}$$

After substituting $\tilde{f}(x)$ into our original objective we arrive at the linear minimization problem

$$\begin{aligned} x_{k+1} = \operatorname*{argmin}_{x \in \mathbb{R}^q} \{ & f(x_k)^T f(x_k) + 2(x - x_k)^T J(x_k)^T f(x_k) \\ & + (x - x_k)^T J(x_k)^T J(x_k)(x - x_k)\}. \end{aligned} \tag{6.4}$$

Taking the derivative and setting it equal to zero gives a recursive definition for the next point in terms of the current one:

$$x_{k+1} = x_k - \left(J(x_k)^T J(x_k)\right)^{-1} J(x_k)^T f(x_k). \tag{6.5}$$

Given an initial guess x_0, equation (6.5) is seen as the classic Gauss-Newton method [13] for the solution of nonlinear least-squares problems.

In practice, one rarely provides closed form expressions for the Jacobian, nor do we want to directly evaluate all p^2 partial derivatives. In fact, for the system $|F\tilde{x}| - m = 0$, the derivative $\frac{d|z|}{dz}$, which shows up in the chain of derivatives needed to compute the Jacobian, does not exist as the function $f(z) = |z|$ is not analytic in the complex plane. We therefore use a variant of Broyden's method [18] in order efficiently compute a numerical approximation to the Jacobian during the course of the iteration (6.5).

6.4 Incremental Reconstruction with Regularization

If the STFT (6.1) is computed with overlapping windows, as is often the case, we can exploit this redundancy in order to estimate a signal from the spectrogram. Both Griffin and Lim's algorithm and the algorithm presented here utilize the constraints on the signal imposed by the overlapping regions when estimating a sequence consistent with the given STFT magnitude. While Griffin and Lim encode these constraints in the form of intersecting convex sets, we recast redundancy in the STFT as the first of two *smoothness constraints*. The second constraint imposes smoothness over a single segment only. Combining these constraints, we construct an initial guess $\tilde{x}_0$ for the current signal segment that can be expected to lead to a good final reconstruction via the iteration (6.5). This process effectively "biases" the root-finding process towards an appropriate solution.

We additionally assume *positivity* in the reconstruction, in order to eliminate phase sign errors. This constraint requires only that we either add a constant factor to the DC elements of the spectrogram before applying the algorithm, or simply work with a non-negative version of the original signal.

6.4.1 Smoothness Across Segments

By definition, in the region of overlap the window of signal corresponding to the i-th and $(i+1)$-th columns of the spectrogram must be the same. If we choose to recover only individual windows $\tilde{x}$ of the signal at a time by solving (6.2), then the above statement implies that the i-th piece of signal ought to factor into the computation of the $(i+1)$-th window of signal. This feedback process can be thought of as a form of regularization: the current window of signal must look something like the previous one. The structure of the STFT tells us that the segments must not change too much from one time instant to the next. If the amount of overlap between adjacent windows is greater, then there is a better chance that this assumption will hold.

6.4.2 Smoothness Within Segments

Overlap constraints provide a good deal of information about $x(n)$, however there are still many possible candidate solutions $\hat{x}(n)$ that satisfy the overlap conditions but do not give back anything near the original signal (when it is known). This problem is amplified when the STFT step size h is large. Therefore, in order to further bias the search for a solution towards a good one, we make an additional smoothness assumption in the region of the window where there is no overlap with the previous segment.

In this region, we must form a reasonable guess as to what the signal might look like when constructing an initial condition $\tilde{x}_0$ for the iterative root-finding procedure.

We explore two smooth possibilities in the non-overlapping region: linear extrapolation from a leading or trailing subset of the known overlap points, or zero-order hold extrapolation from the last overlap point. Smoothness can be quantified for both of these methods by examining the energy in the first and second derivatives of a signal constructed by concatenating the "fixed" values with the h extrapolated points. If, in the linear extrapolation case, we find the energy over the entire signal in the first derivative to be E_1, and in the second derivative to be E_2, then it must be true that for zero-order hold with the same fixed portion of the signal, the resulting signal $x_z(n)$ will have energies

$$\|Dx_z(n)\|_2 \leq E_1, \quad \text{and} \quad \|D^2 x_z(n)\|_2 \geq E_2 \tag{6.6}$$

where D and D^2 denote first and second discrete derivative operators respectively. Linear extrapolation therefore reduces energy in the second derivative, while zero-order hold continuation will give lower energy in the first derivative. Empirically we have found that linear-extrapolation is preferable when the STFT step size h is small compared to the window size (10% of the window width or less), while zero-order hold yields improved results when h is large. Eventually, linear extrapolation may well produce samples far from the known values as we extrapolate away from the known region of the signal. Thus a mixture of the two methods is yet another possibility, where we might extrapolate for a small number of points relative to the window size and sampling rate, and then hold the final value for the remainder of the extrapolation interval.

We impose one final constraint on each segment. After the root-finding iteration has converged to a solution, we set the mean of the result to the value specified by the DC term of the length N segment's Fourier magnitude, $|S(\omega = 0, \ell_0)|/N$.

6.4.3 Incremental Signal Reconstruction: Forward-Backward Recursions

The algorithm proceeds by stepping through the STFT magnitude, column by column, first in the *forward* direction, and then heading *backwards*. At each segment, a window of signal is estimated and written to a buffer at a position corresponding to that window's position in the original signal. In the forward direction, smoothness across segments is incorporated when computing a recursive solution to (6.2) for window $(i + 1)$, by explicitly fixing points in the region of overlap with window i to the shared values in the solution returned for that segment. Going backwards, we instead fix the overlapping values for segment i to those previously given by segment $(i + 1)$. The very first window of signal in the forward pass is computed from an initial guess $\tilde{x}_0$ comprised of random values drawn from the uniform distribution $\mathcal{U}[0, 1]$. The first backwards pass window is computed from the last forward solution. The full reconstruction is then assembled by overlap-adding the individual time-domain segments

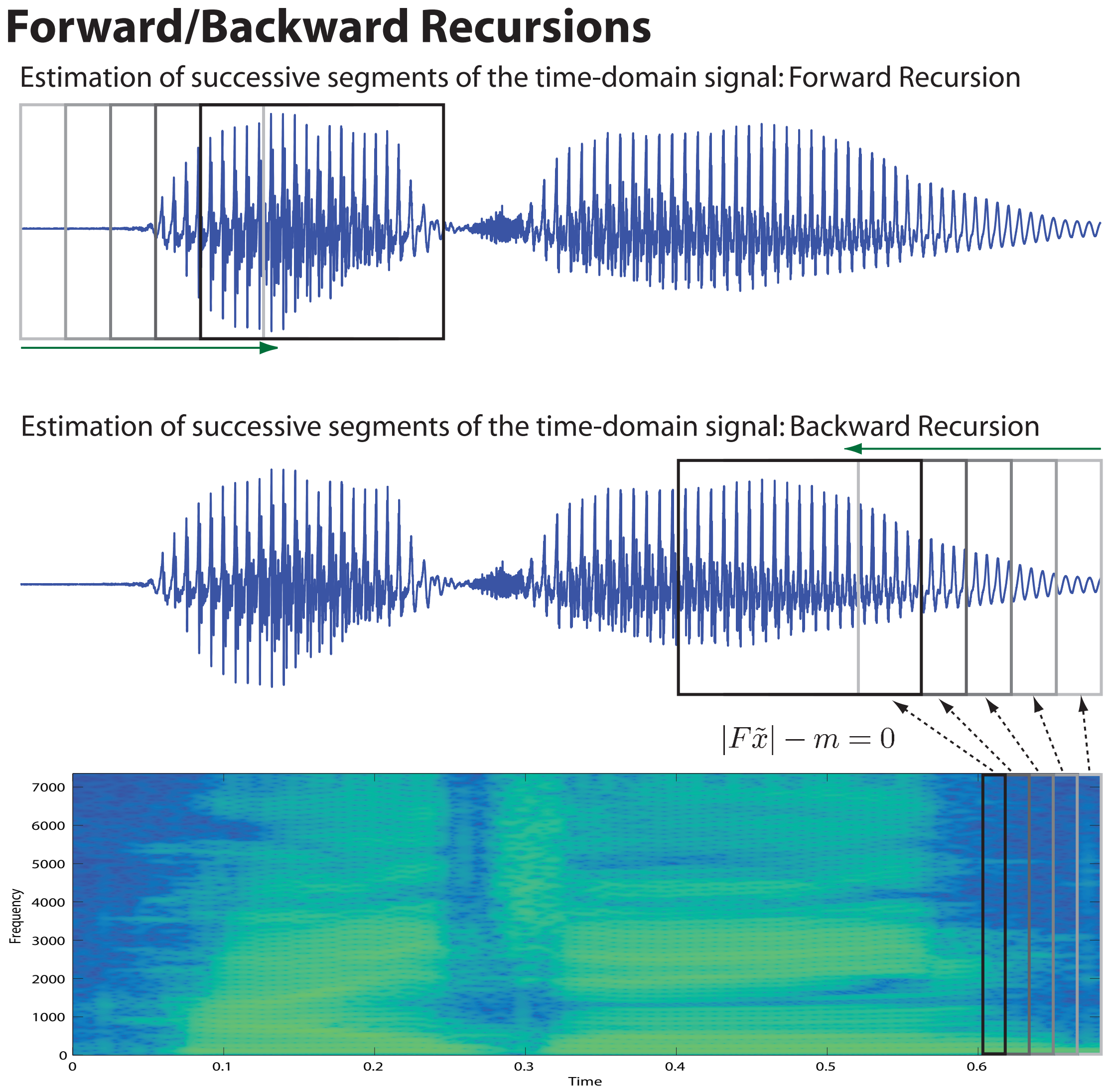


Figure 6-1: Illustration of the forward-backward recursions as applied to an example utterance.

```
$\tilde{x}^0 = \text{rand}(\mathcal{U}[0,1])$
for all spectrogram columns $m^i,\ i = 1, \ldots, L-1$ do
    · Compute the $h$ elements of $\tilde{x}_0$ by extrapolating from the last $p$
    overlapping points in $\tilde{x}^i$
    · Let $\tilde{x}_{ol}$ be the $N-h$ points in $\tilde{x}^i$ that will overlap with $\tilde{x}^{i+1}$
    · Compute the solution $\hat{x}$ to $|F\tilde{x}^{i+1}| - m^{i+1} = 0$ using the Gauss-Newton
    iteration with initial condition $\tilde{x}_0$, while holding the overlap points in $\tilde{x}^{i+1}$
    fixed so that $\tilde{x}^{i+1} = [\tilde{x}_{ol}^T\ \ \hat{x}^T]^T$
    · Set $\tilde{x}^{i+1} = \tilde{x}^{i+1}$ - mean[$\tilde{x}^{i+1}$ ] + $m^{i+1}(0)/N$
end
· Repeat the previous loop in the reverse direction, over all spectrogram
columns $m^i,\ i = L, \ldots, 1$, extrapolating in the opposite direction with
$\tilde{x}^{i-1} = [\hat{x}^T\ \ \tilde{x}_{ol}^T]^T$ where $\tilde{x}_{ol}$ are the points in $\tilde{x}^{i-1}$ that overlap with segment $\tilde{x}^i$.
· Reconstruct $x(n)$ by overlap-adding the segments $\{\tilde{x}^i\}_{i=1}^L$
```

Algorithm 2: Incremental Signal Reconstruction Algorithm

according to the original STFT hop size. We illustrate the process as applied to an example utterance in Figure 6-1.

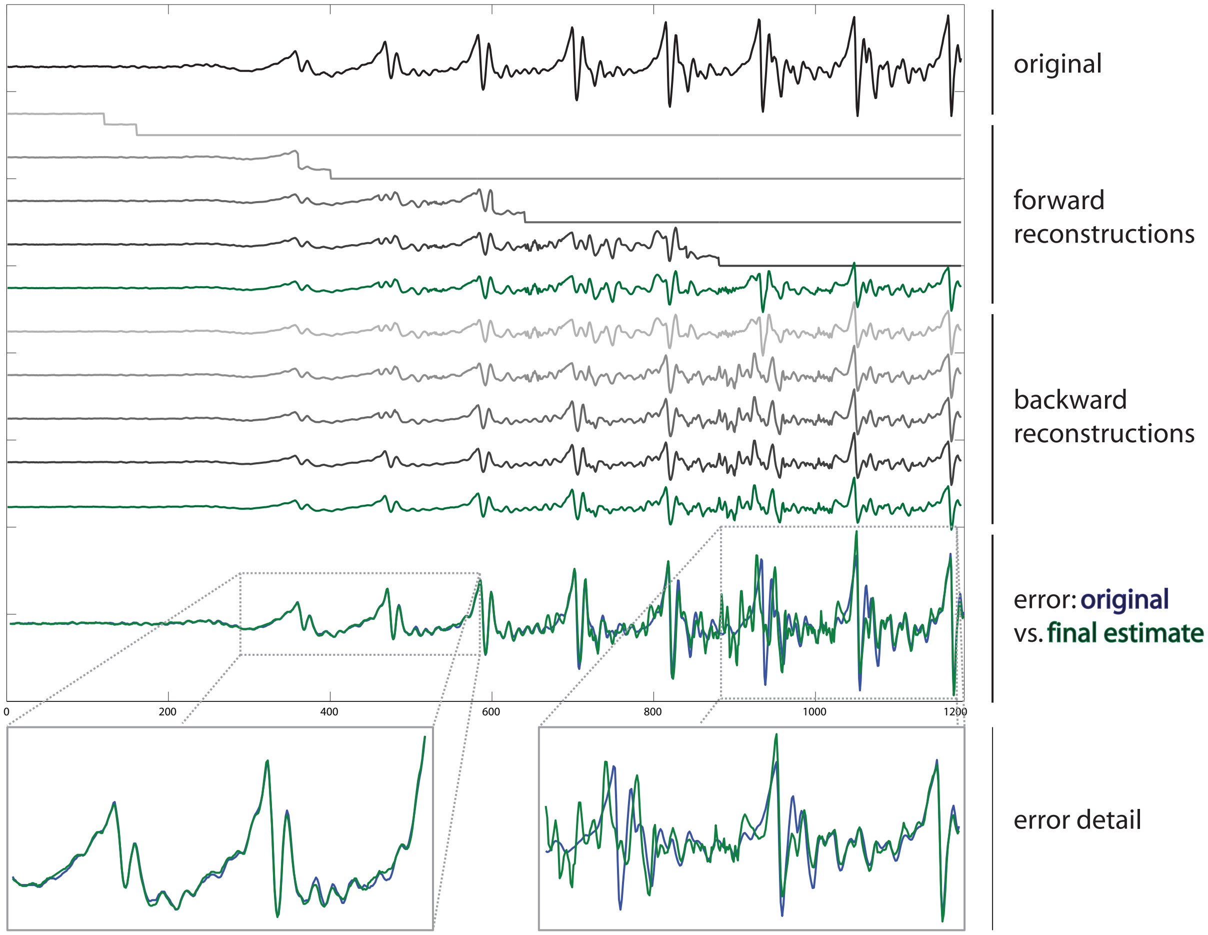
The forward pass can be thought of as computing an initial estimate of the signal using previously computed segments for guidance. The backward pass then back-propagates information and constraints accumulated in the forward pass, and can be seen to effectively "repair" errors incurred during the forward computations. Empirically, it is often the case that the first few reconstructions in the forward pass tend to be error prone, due to a lingering influence from the random initial starting point used to launch the algorithm. However, the smoothness constraints we have described quickly guide the roots towards the desired signal values.

Although we have thus far discussed only interactions between adjacent segments of the signal, for STFT hop sizes $h < N$ a given segment will both constrain, and be constrained by, many other segments in a columnar region of the spectrogram. Each window of signal can be thought of as a node within a (cyclic, reachable) network, across which constraints may propagate in any direction. In this framework, recovering the full signal $x(n)$ is akin to finding an equilibrium point where all the constraints are satisfied simultaneously. It is possible that a dynamical systems perspective can be used to describe the behavior of this algorithm.

Finally, we have found that repeating the algorithm on a spectrogram derived from a time-reversed version of the original signal can reduce the reconstruction error further. Specifically, averaging the results under the normal and reversed conditions often times will give a SIR lower than either of the individual reconstructions alone.

A concise summary of our algorithm is given in Algorithm 2, where we have assumed that the size $K \times L$ spectrogram has been computed with windows of length N and hop sizes h.

Example: Speech Reconstruction



14.7kHz sampling rate, male speaker: "Hi". 100 sample rectangular STFT window, 60 sample hop size (40% overlap), 200 FFT bins per window.

Figure 6-2: Reconstruction of a short segment of speech using the proposed algorithm, proceeding over the time-domain signal from left to right, and then backwards from right to left. Traces are shown for only a subset of the iterations. Note that the reconstruction improves after both the forward and backward passes.

6.5 Experiments

We compared the proposed algorithm to Griffin and Lim's technique on both speech and music signals. In order to better balance the computation times of the two methods, our algorithm was applied only once to the signals and we did not include the time-reversed solution. We additionally applied a mixture of extrapolation techniques when forming the initial root-finding guess. A linear model was fit to the leading/trailing $p = \min(20, N - h)$ points, and extrapolated for 5 points. The remaining unknown points in the non-overlapping region were set to the last extrapolated point. The success of our algorithm does not critically depend on these choices. Griffin and Lim's algorithm was passed uniformly distributed random initial phase guesses $r_i \sim \mathcal{U}(0, 1)$, and was run until the relative decrease in ℓ_2 error between the STFT magnitude of the reconstruction and the given magnitude was less than 0.1%. We separately evaluated Griffin and Lim when given both strictly positive signals and zero-mean signals. For both methods, positivity was enforced by working with spectrograms derived from the target signal $x'(n) = x(n) - \min_m[x(m)]$, rather than $x(n)$ itself.

The speech signals we attempted to reconstruct consisted of a male speaker and a female speaker uttering the phrase "Hi Jane", while the music sample consisted of a percussive drum loop with no other instruments or vocals. The latter example is representative of a class of signals that tends to be more difficult to recover due to abrupt, non-smooth transitions and noisy crashes which dominate the structure of the signal. The signals varied in length from 0.75s to 2.2s, were all sampled at 14.7kHz, and were normalized uniformly. In each experiment, we used a 100 sample (6.8ms) square (boxcar) window and 200 FFT bins. The STFT hop size, however, was systematically varied from 10% to 90% of the window width in steps of 10 samples. We then compared the power signal-to-noise ratio (SNR) between the original signal x_o and the reconstruction x_r for each STFT hop size, where

$$\mathcal{SNR} = 20 \log_{10} \left(\frac{\|x_o\|_2}{\|x_o - x_r\|_2} \right) \text{ (dB)}. \tag{6.7}$$

While we have found that both methods are stable with respect to initial conditions, the experiments were nevertheless repeated several times. An graphical illustration of the reconstruction process is shown in Figure 6-2. Note that the backwards pass "repairs" errors left behind by the initial forward pass. An error detail after two passes is also shown.

We show the averaged performance, over 200 trials, on the male and female speech samples for both algorithms as a function of STFT hop size in the top two panes of Figure 6-3, where the trace denoted "incremental" corresponds to our technique. Comparison for the percussive drum loop can be seen in the bottom pane of 6-3. It can be seen that our algorithm consistently outperforms Griffin and Lim's algorithm as measured by SNR over the full range of hop sizes. In the male speech case for example, at approximately $h = 30$ positivity of the input signal affects Griffin and Lim's performance. Overall, it is evident that our technique degrades more gracefully as redundancy in the STFT is reduced.

While these results are encouraging, Griffin and Lim's algorithm can give perceptually good results even though the SNR is poor. Often times this can be attributed to inaudible sign errors in the reconstruction, particularly for small hop sizes. With larger hop sizes, we have observed that the error is mainly due to poor reconstruction and significant distortion can be heard. For this reason, it is important to compare the perceptual quality of the two algorithms. In most cases our algorithm is perceptually better over the full range of hop sizes, and the distinction is greater as the STFT analysis window size is increased (while maintaining similar hop sizes as a percentage of window width).

For small STFT hop sizes our algorithm can require more computation time than the Griffin-Lim algorithm, depending on the number of iterations needed to meet the Griffin-Lim convergence criteria. Otherwise, the two algorithms are generally comparable in running-time.

6.6 Conclusion

The algorithm we have presented typically achieves greater signal-to-noise ratios than existing reconstruction techniques, and the perceptual improvement for speech and music signals is particularly noticeable when there is less redundancy in the STFT.

In designing the algorithm, we imposed several time-domain regularization constraints: (1) We exploited the overlap constraints inherent in the structure of the STFT explicitly and enforced smoothness across windows of the signal. (2) We enforced smoothness within an individual segment by extrapolating in the region where samples were unknown. And, (3) we propagated these constraints throughout the entire signal by applying the smoothness assumptions recursively in both forward and backward directions. We then incorporated these time-domain constraints into a root-finding problem in the frequency domain. Collectively, the constraints can be thought of as biasing the spectral root-finding procedure at each segment towards smooth solutions that are shown to be highly accurate when the true values of the signal are available for comparison.

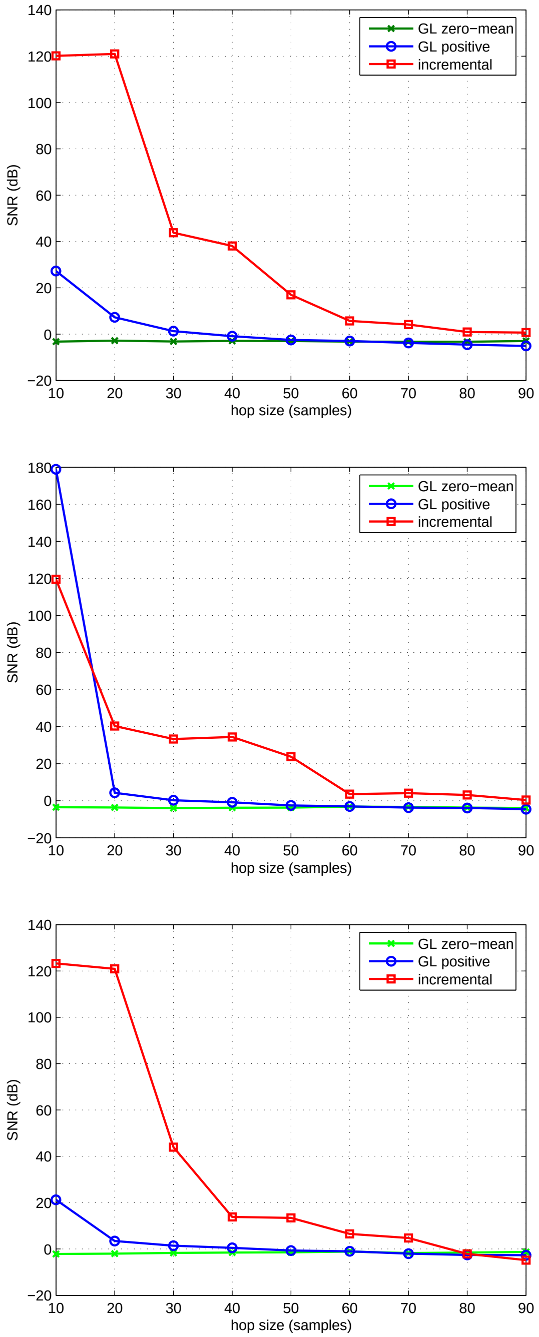


Figure 6-3: Algorithm performance (dB) vs. STFT hop size for a 100 sample analysis window, as applied (in top-to-bottom order) to male speech, female speech, and a drum loop.

Chapter 7

Conclusion

We have described a mathematical formalism for analyzing hierarchical learning, emphasizing by way of a theoretical analysis of the neural response map, the interplay between architecture, invariance, and discrimination. We then showed, via experiments involving applications in vision and speech, that hierarchical learning machines can be advantageous, and warrant further exploration. The contents of this thesis should be seen as a step towards answering important questions concerning the sample complexity of hierarchical architectures and task-specific learning with general systems exploiting decomposability of the data. More specifically, we have sought to answer the following questions:

- When and why is a "deep" architecture preferred, and how does the architecture induce invariance and discrimination properties?
- For tasks that can be decomposed into a hierarchy of parts, how can we show that a supervised classifier trained using a hierarchical feature map will generalize better than an non-hierarchical/parts-based alternative?
- Can we understand and cast learning in hierarchies using tools from statistical learning theory?
- What can we understand about the cortex by studying computation with hierarchies abstractly and mathematically?

In attempting to shed light on these issues, we have raised and left open many more important research possibilities and there is much future work to consider. In particular,

- How can we choose the optimal number of layers?
- How do other architectural choices influence discrimination and invariance properties?
- What is the best way to learn the templates?

- What are the advantanges/disadvantages to different choices of the pooling function?
- A stability analysis of the neural response, with respect to perceptually insignificant changes (e.g. changes in background texture), and with respect to distractor objects and clutter.
- Integration of feedback via top-down contextual priors and notions of attention (possibly integrated via measures on transformations ρ_{H_i}, as discussed in Chapter 2).

In the section that follows we compare and contrast the neural response/derived kernel framework, and one of the most common deep learning approaches found in the literature. We then provide an itemized list of answers to common criticisms of the framework proposed in Chapter 2.

7.1 Existing Deep Learning Techniques and the Neural Response: A Comparison

Derived kernels are defined with the goal of understanding when and why deep networks can outperform traditional single-layer alternatives while building on the established tools offered by statistical learning theory. At the same time, derived kernels do not sacrifice elements important for good performance in exchange for analytical tractability, as confirmed by our empirical simulations investigating sample complexity in practical settings. As we have argued in the Introduction, much of the work involving deep architectures to date is biased towards applications, and computational expediency. Indeed, relatively little is understood about deep belief networks from a theory perspective because the existing models are difficult to analyze formally.

We first list broad similarities between derived kernels and deep networks, then the major differences, and finally give a more formal comparison between the two.

7.1.1 Similarities

From a practical standpoint the different hierarchical architectures share several immediate similarities. The derived kernel architecture is similar to a deep belief network (DBN) and could be made even more similar if the stage of unsupervised learning were to be modified. We elaborate on this point below. Obvious similarities include the following:

- They are both hierarchical and deep (several layers).
- They are both feedforward at run time (once the unsupervised learning is done).
- They both have a stage of unsupervised learning, which proceeds layer by layer.

7.1.2 Differences

There are also several distinct differences worth highlighting:

- The neural response is designed to directly incorporate operations to build invariance to certain transformations of the input such as translations. There is generally no such explicit property in other deep learning models, though LeCun and colleagues have attempted to incorporate explicit invariances in his convolutional neural network variants [65, 102], and recent work due to Lee et al. [66] incorporates translation invariance.
- Deep learning networks are often described in probabilistic terms, in particular when a Restricted Boltzmann Machine (RBM) is used to pre-train pairs layers. They are in this case undirected graphical models. Derived kernel networks are not described in probabilistic terms but in a functional analytic language.
- The unsupervised learning stage for each pair of layers in a DBN model typically uses an autoencoder (reconstruction-based) approach. In the simplest derived kernel setting, there is a stage (at each layer) of unsupervised learning of the templates which can be described as a Monte Carlo estimate of a continuous encoding operator. This learning stage could of course be more sophisticated (see below).
- There is no formal theory describing in mathematical terms the various deep learning architectures. We believe such a body of formal definitions to be the key to finding sharp theoretical results.

7.1.3 Technical Considerations

Because derived kernels and many deep belief networks are mathematically formulated in significantly different terms, one must make quantitative comparisons with care. We can, however, make a clear distinction between the ways in which both models incorporate (unsupervised) learning. We briefly review the important objects arising in both deep networks and derived kernels so as to provide a more specific, technical comparison.

In the following discussion, we use x to denote inputs in the case of neural networks, and f in the case of the neural response. The variable f can be thought of as either a function or a vector.

Deep Autoencoding Neural Networks

We review learning in deep networks built from autoencoders, and consider the case of an autoencoder parameterized as a multi-layer perceptron. There are many variations in the literature, however we believe this picture to be simpler (while no less general) than networks comprised of Restricted Boltzmann Machines (e.g. as in [49]) because the layerwise pre-training can be done with gradient descent, and the objective function can be easily interpreted. For a given encoder/decoder pair of layers mapping the variables $x \to y \to z$, we can describe the model as

$$\begin{aligned} y &= \sigma(\mathbf{W}_1 x + \mathbf{b}_1) \\ z &= \sigma(\mathbf{W}_2 y + \mathbf{b}_2) \end{aligned}$$

where $x \in \mathbb{R}^n$ is an input vector, $z \in \mathbb{R}^n$ is the output, $\mathbf{W}_1$ is the $(r \times n)$ encoding weight matrix, $\mathbf{W}_2$ is the $(n \times r)$ decoding matrix, and $\mathbf{b}_1, \mathbf{b}_2$ are bias vectors. We denote by $\sigma(z) = (1 + e^{-z})^{-1}$ the sigmoid activation nonlinearity acting componentwise on a vector. Typically the data is passed through a "bottleneck" [49, 11], in which case r is chosen smaller than n. A commonly occurring variant of the autoencoding building block fixes $\mathbf{W}_2 = \mathbf{W}_1^T$ for pre-training. During the pre-training phase as described by Hinton & Salakhutdinov [49], the network can be trained to reconstruct in a least-squares sense a set of input data $\{x_i\}_i$, leading to the following cost function

$$E(\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2) = \sum_i \|z_i - x_i\|_2^2.$$

This cost is then minimized with respect to the weights and biases $\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2$ via backpropagation. Other cost functions, such as reconstruction cross-entropy, have also been used [116], however we consider the above setting to be more canonical. If the activation function σ is chosen to be linear, this network will learn something similar to PCA [49].

Generally, many heuristics are involved in the non-convex optimization. A deep network is trained by first training the outer pair of layers as above. The next pair of nested layers, one step inwards, is trained similarly but using the output of the previous layer as the input data to be reconstructed, and the process repeats for as long as required. The size of the weight matrices at the inner layers are again chosen by the user, but must be compatible with the other layers to which they connect. Finally, after pre-training of each pair is complete, the network is "unrolled" into one deep encoder/decoder structure, and a final polishing phase of light training is applied to the entire model (often using a supervised criterion).

The Derived Kernel Encoding Operator

In the derived kernel framework, learning proceeds in a markedly different manner. Recall from Chapter 2 that the neural response has a self consistent definition, whereby the neural response at a given layer can be expressed in terms of the neural responses at the previous layer

$$N_{Sq}(f)(t) = \max_{h \in H} \left\langle \widehat{N}_v(f \circ h), \widehat{N}_v(t) \right\rangle_{L^2(T')}, \quad t \in T$$

with $H = H_v$, $T' = T_u$ and $T = T_v$. It is both instructive and revealing to recast this definition in vector notation as

$$\begin{aligned}
N_{Sq}(f) &= \begin{bmatrix} \max_{h\in H}\left\langle \widehat{N}_v(f\circ h), \widehat{N}_v(t_1)\right\rangle_{L^2(T')} \\ \vdots \\ \max_{h\in H}\left\langle \widehat{N}_v(f\circ h), \widehat{N}_v(t_{|T|})\right\rangle_{L^2(T')} \end{bmatrix} \\
&= \max_{h\in H}\left\{ \begin{bmatrix} \leftarrow & \widehat{N}_v(t_1) & \rightarrow \\ & \vdots & \\ \leftarrow & \widehat{N}_v(t_{|T|}) & \rightarrow \end{bmatrix} \widehat{N}_v(f\circ h) \right\} \\
&=: \max_{h\in H}\left\{ \Pi_v \widehat{N}_v(f\circ h) \right\}
\end{aligned}$$

where the max operation is assumed to apply elementwise. The important object here is the *encoding operator* $\Pi_v : L^2(T_u) \to L^2(T_v)$, defined by

$$(\Pi_v F)(t) = \langle \widehat{N}_v(t), F\rangle_{L^2(T_u)}$$

for $F \in L^2(T_u), t \in T_v$. The operator Π_v is seen here as a $|T_v| \times |T_u|$ matrix: each row of the matrix Π_v is the (normalized) neural response of a template $t \in T_v$, so that

$$(\Pi_v)_{t,t'} = \widehat{N}_v(t)(t')$$

with $t \in T_v$ and $t' \in T_u$. The template t could be a patch of a natural image, sampled according to its natural probability. This perspective highlights the action of the hierarchy as alternating *pooling* and *filtering* steps, realized by the max and the Π operator respectively. Note in particular, that the Π operator does not depend on the input f.

Learning with derived kernels can be integrated in two main ways: (1) via the selection of the templates, and/or (2) via the Π operators.

Comparison

A key aspect of our approach is the interplay between invariance and discrimination. Both properties are "built-in" to the hierarchy. This corresponds to two main operations: pooling and filtering. The latter is where unsupervised learning takes place. This aspect highlights a point of distinction between our work and that of the deep learning literature, and we elaborate on these themes.

- In our setting, the architecture itself is motivated by a desire to (1) impose invariance and (2) exploit the hierarchical organization of the physical world. In most deep networks, with the notable exception of convolutional neural networks and to a limited extent the work of [66], the architecture does not play such an active role. Indeed, the user must settle on an architecture with little guidance other than end-to-end experimental validation on a development set of data.

- A key learning process in our setting is that of learning the Π operators at each layer of the derived kernel hierarchy, in correspondence with learning the $\mathbf{W}$ matrices at each layer of the deep autoencoder above. However, this picture ignores the fact that in the neural response we consider a collection of *local* encodings over which we pool via the max. In contrast, the $\mathbf{W}$ matrices above can be seen as *global* quantities since typically all to all connectivity is imposed between layers of an autoencoder neural network. The advantage of local, multiscale principles has been demonstrated in, e.g. the case of wavelet analysis of images and audio signals.
- The simplest way to define a new Π operator is by choosing a representation on a suitable basis. For example, via PCA, by diagonalizing the Π matrices *initially* given as encodings of the templates. This loosely parallels the case where the autoencoder neural network above realizes PCA when $\sigma(\cdot)$ is linear. However we again caution that in our framework we learn multiple localized representations at different scales explicitly. It is these local encoding operators that are diagonalized, so that the resulting principal components do not correspond to the global components one obtains with classical PCA applied to the entire input object.
- More interesting choices of the Π operators may be made, reflecting the geometry or topology of the data, or including ideas from sparse coding. For example, Π could be represented in terms of the eigenfunctions of the Laplacian. Moreover, our approach is not restricted to representations derived from a reconstruction-based criteria, and includes e.g. distance metric learning.
- Template selection at different scales in a derived kernel architecture is another place where learning occurs. While the $\mathbf{W}$ encoding/decoding matrices are relatively unconstrained in autoencoders, the default Π matrices are determined by template encodings. Rather than random sampling, the templates may themselves be chosen using methods motivated by geometric or sparse approximation ideas (as above).

7.1.4 Remarks

We summarize in the form of several remarks the advantages of the derived kernel formalism which further distinguish our approach from the deep learning literature.

- There is little work attempting to understand hierarchical learning in terms of the well established tools of statistical learning theory. Our framework makes this sort of an analysis accessible because the primary objects of interest are kernels and operators on Hilbert spaces of functions. Extending the existing theory to the multi-layer case is an important open problem.

- Deep networks are very general learning machines, however theoretical understanding has only been obtained in restricted settings such as Boolean function learning [11, 64]. We believe our framework will allow analysis of hierarchies within the context of a broader class of problems.
- Deep belief networks do not provide a way for the modeler to easily incorporate domain knowledge, or exploit important aspects of the data in a direct way. For example, deep belief networks may or may not discover through training a way to leverage geometry of the data. On the other hand, the derived kernel framework provides an explicit avenue for incorporating particular data representations (by a choice of Π) which can exploit geometry, topology, or other aspects important to the modeler. Sparsity and geometry are ubiquitous concepts in machine learning, and yet are apparently either incompatible with or difficult to incorporate into most existing hierarchical models (see e.g. [119]). The derived kernel framework allows one to integrate these concepts in a natural way without incurring significant additional computational complexity, or loss of algorithmic simplicity.
- In deep networks, if invariance to isometric transformations is desired, one needs to either hope that one has enough training data and a suitably flexible architecture to discover the invariance, or needs to design a specialized network and re-derive the training rules. The objective may also exhibit an increased susceptibility to local minima. In the case of derived distances, one needs only to choose a suitable, invariant mother kernel and the desired invariance will propagate throughout the network automatically. Alternatively, one may also simply add further transformations to the pooling sets H_m associated with each layer.

7.2 Answers to Criticisms of the Neural Response Framework

We attempt to give answers to criticisms of the derived kernel work presented in Chapter 2. The discussion is organized in a Question and Answer format.

- *The analysis you've presented seems particular. How does this apply to any other setting?* There is currently little work attempting to formalize learning in hierarchies. The neural response model, while simple, represents a first step in this direction, where we have fixed some basic rules of the game. Although our analysis considers 2-D images and strings with particular architectural configurations, we believe that our results make constructive statements about general properties such as invariance and discrimination ability. The framework we've discussed also shares some commonalities with wavelets, where an initial "mother kernel" is iterated over by applying a recursive definition involving domains of increasing size. As was the case in the development of wavelets, the construction of a derived kernel itself is important. This perspective places our work within the context of an established and more familiar body of research.

- *Why do you use the* max*? It's going to make the mathematics difficult.* While the max operation is more difficult to work with mathematically, it allows for a richer set of models as compared to a simple average, and contributes towards preventing the layered architecture from collapsing into a simple linear operation. The max was also originally used in the CBCL model, and there is evidence supporting its presence in the brain [60, 41]. We have also shown that the max induces equivalence classes of inputs, thereby facilitating invariant models. Intuitively, the notion that we take as the output of a filtering step the best match over a set of templates is compelling, and parallels winner-take-all behavior found in dynamical models of neural networks. We prefer to work with this particular source of nonlinearity rather than another, perhaps equally challenging but unmotivated, choice. Finally, as shown in Section 4.2.1 of Chapter 4, several important invariance results described in Chapter 2 extend to arbitrary pooling functions beyond the max.

- *How can you enforce invariance of the initial kernel?* That global invariances can come from a collection of local invariances is somewhat surprising. Recall that you need only enforce invariance locally, at the lowest level of the hierarchy. Representing patches of images, for example, as histograms, would allow for rotation and reflection invariance. Spin-images [54], is another example of a local rotationally invariant feature, computed by summing the energy in annuli about a point in the gradient image.

- *It seems that sampling random templates is naive.* Choosing the templates in an optimal sense is a problem that is most often dependent on the particular task at hand. In order to provide the most general sort of results, we did not want to assume any particular domain or restrict the input to any particular class of stimuli; we need only assume the existence of a measure ρ through which one may view and sample the world. Template selection is an open (and involved) problem that could also include learning techniques which exploit data geometry, topology and/or sparsity, as discussed in Section 2.2.5 of Chapter 2.

Bibliography

[1] K. Achan, S.T. Roweis, and B.J. Frey. Probabilistic inference of speech signals from phaseless spectrograms. In *Advances in Neural Information Processing Systems 16*, Cambridge, MA, 2004. MIT Press.

[2] Y. Amit and M. Mascaro. An integrated network for invariant visual detection and recognition. *Vis. Res.*, 43(19):2073–2088, 2003.

[3] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 68:337–404, 1950.

[4] M. Artin. *Algebra.* Prentice-Hall, 1991.

[5] L. Atlas and S. Shamma. Joint acoustic and modulation frequency. *EURASIP Journal on Applied Signal Processing*, 2003(7):668–675, 2003.

[6] N. Bacon-Mace, M.J. Mace, M. Fabre-Thorpe, and S.J. Thorpe. The time course of visual processing: backward masking and natural scene categorisation. *Vis. Res.*, 45:1459–1469, 2005.

[7] A. Bar-Hillel, T. Hertz, N. Shental, and D. Weinshall. Learning a Mahalanobis metric from equivalence constraints. *Journal of Machine Learning Research*, 6:937–965, 2005.

[8] Peter L. Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

[9] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

[10] D.A. Bendor and X. Wang. The neuronal representation of pitch in primate auditory cortex. *Nature*, 436(7054):1161–1165, 2005.

[11] Y. Bengio. Learning deep architectures for ai. In *Foundations and Trends in Machine Learning.* (to appear), 2009.

[12] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle. Greedy layer-wise training of deep networks. In *Advances in Neural Information Processing Systems*, pages 4–1235. MIT Press, 2006.

[13] D. Bertsekas. *Nonlinear Programming.* Athena Scientific, Belmont, MA, 1999.

[14] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psych. Rev.*, 94:115–147, 1987.

[15] H. Bourlard and S. Dupont. Subband-based speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997.

[16] J. Bouvrie and T. Ezzat. An incremental algorithm for signal reconstruction from short-time fourier transform magnitude. In *International Conference on Spoken Language Processing (Interspeech)*, 2006.

[17] J. Bouvrie, T. Ezzat, and T. Poggio. Localized spectro-temporal cepstral analysis of speech. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2008.

[18] C.G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593, 1965.

[19] C. M. Burges. *Neural Networks for Pattern Recognition.* Cambridge Univ. Press, 1996.

[20] C.J.C. Burges, J.C. Platt, and S. Jana. Extracting noise-robust features from audio data. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2002.

[21] B. Chen, Q. Zhu, and N. Morgan. Learning long-term temporal features in mvcsr using neural networks. In *International Conference on Spoken Language Processing*, 2004.

[22] C.-P. Chen and J. Bilmes. Mva processing of speech features. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(1):257–270, Jan 2007.

[23] T. Chih, P. Ru, and S. Shamma. Multiresolution spectrotemporal analysis of complex sounds. *Journal of the Acoustical Society of America*, 118:887–906, 2005.

[24] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similiarity metric discriminatively, with application to face verification. *CVPR*, 2005.

[25] P. Clarkson and P.J. Moreno. On the use of support vector machines for phonetic classification. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 585–588, 1999.

[26] L. Cohen. *Time-frequency Analysis.* Prentice Hall, Englewood Cliffs, NJ, 1995.

[27] T. M. Cover and J. A. Thomas. *Elements of information theory.* John Wiley and Sons, Inc., 1991.

[28] N. Cristianini and J. Shawe Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, Cambridge, UK, 2000.

[29] F. Cucker and D. X. Zhou. *Learning theory: an approximation theory viewpoint.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge, 2007.

[30] X. Cui, M. Iseli, Q. Zhu, and A. Alwan. Evaluation of noise robust features on the aurora databases. In *International Conference on Spoken Language Processing*, pages 481–484, 2002.

[31] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Comput. Vision and Pattern Recognit.*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005.

[32] L. Deng, J. Droppo, and A. Acero. Estimating cepstrum of speech under the presence of noise using a joint prior of static and dynamic features. *IEEE Transactions on Speech and Audio Processing*, 12(3):218–233, 2004.

[33] L. Deng, J. Droppo, and A. Acero. Dynamic compensation of hmm variances using the feature enhancement uncertainty computed from a parametric model of speech distortion. *IEEE Transactions on Speech and Audio Processing*, 13(3):412–421, 2005.

[34] T. Ezzat, J. Bouvrie, and T. Poggio. Max-gabor analysis and synthesis of spectrograms. In *International Conference on Spoken Language Processing (Interspeech)*, 2006.

[35] T. Ezzat, J. Bouvrie, and T. Poggio. Am-fm demodulation of spectrograms using localized 2d max-gabor analysis. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007.

[36] T. Ezzat, J. Bouvrie, and T. Poggio. Spectro-temporal analysis of speech using 2-d gabor filters. In *International Conference on Spoken Language Processing (Interspeech)*, 2007.

[37] J.R. Fienup. Phase retrieval algorithms: a comparison. *Applied Optics*, 21(15):2758–2769, 1982.

[38] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach.* Prentice Hall, 2002.

[39] D.J. Freedman, M. Riesenhuber, T. Poggio, and E.K. Miller. Categorical representation of visual stimuli in the primate prefrontal cortex. *Science*, 291:312–316, 2001.

[40] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cyb.*, 36:193–202, 1980.

[41] T.J. Gawne and J.M. Martin. Responses of primate visual cortical V4 neurons to simultaneously presented stimuli. *J. Neurophys.*, 88:1128–1135, 2002.

[42] S. Geman and M. Johnson. Probabilistic grammars and their applications. *International Encyclopedia of the Social & Behavioral Sciences*, pages 12075–12082, 2002.

[43] F. Girosi and T. Poggio. Networks and the best approximation property. *Biological Cybernetics*, 63(3):169–176, 1990.

[44] D.W. Griffin and J.S. Lim. Signal estimation from modified short-time fourier transform. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 32(2):236–243, 1984.

[45] A. Halberstadt and J. Glass. Heterogeneous measurements and multiple classifiers for speech recognition. In *International Conference on Spoken Language Processing*, 1998.

[46] M.H. Hayes, J.S.Lim, and A.V. Oppenheim. Signal reconstruction from phase or magnitude. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 28(6):672–680, 1980.

[47] H. Hermansky. Trap-tandem: Data-driven extraction of temporal features from speech. In *Automatic Speech Recognition and Understanding, 2003. ASRU '03. 2003 IEEE Workshop on*, pages 255–260, 2003.

[48] H. Hermansky and S. Sharma. Temporal patterns (traps) in asr of noisy speech. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1999.

[49] G.E. Hinton and R.R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[50] S. Hochstein and M. Ahissar. View from the top: Hierarchies and reverse hierarchies in the visual system. *Neuron*, 36:791–804, 2002.

[51] P.L. Van Hove, M.H. Hayes, J.S. Lim, and A.V. Oppenheim. Signal reconstruction from signed fourier transform magnitude. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 31(5):1286–1293, 1983.

[52] D.H. Hubel and T.N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *J. Phys.*, 195:215–243, 1968.

[53] M. Jensen and S. Nielsen. Speech reconstruction from binary masked spectrograms using vector quantized speaker models. Master's thesis, Technical University of Denmark, August 2006.

[54] Andrew Johnson and Martial Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(1):433 – 449, May 1999.

[55] S. Kajarekar, B. Yegnanarayana, and H. Hermansky. A study of two dimensional linear discriminants for asr. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 1, pages 137–140, 2001.

[56] T. Kitamura, E. Hayahara, and Y. Simazaki. Speaker-independent word recognition in noisy environments using dynamic and averaged spectral features based on a two-dimensional mel-cepstrum. In *International Conference on Spoken Language Processing*, 1990.

[57] M. Kleinschmidt. Localized spectro-temporal features for automatic speech recognition. In *International Conference on Spoken Language Processing*, 2003.

[58] M. Kleinschmidt and D. Gelbart. Improving word accuracy with gabor feature extraction. In *International Conference on Spoken Language Processing*, 2002.

[59] L. Lamel, R. Kassel, and S. Seneff. Speech database development: Design and analysis of the acoustic-phonetic corpus. In *DARPA Speech Rec. Workshop*, pages 100–109, 1986.

[60] I. Lampl, D. Ferster, T. Poggio, and M. Riesenhuber. Intracellular measurements of spatial integration and the MAX operation in complex cells of the cat primary visual cortex. *J. Neurophys.*, 92:2704–2713, 2004.

[61] H. Larochelle, Y. Bengio, J. Louradour, and P. Lamblin. Exploring strategies for training deep neural networks. *Journal of Machine Learning Research*, 10:1–40, Jan 2009.

[62] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *International Conference on Machine Learning*, 2007.

[63] S. Lazebnik, C. Schmid, and J. Ponce. A maximum entropy framework for part-based texture and object recognition. In *Proc. of the Intern. Conf. Comput. Vision*, 2005.

[64] Y. LeCun and Y. Bengio. Scaling learning algorithms towards ai. In L. Bottou, O. Chapelle, and D. DeCoste J. Weston, editors, *Large-Scale Kernel Machines.* MIT Press, Cambridge, MA, 2007.

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

[66] H. Lee, R. Grosse, R. Ranganath, and A. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *Proceedings of the Twenth-Sixth International Conference on Machine Learning*, 2009.

[67] T.S. Lee and D. Mumford. Hierarchical bayesian inference in the visual cortex. *J Opt Soc Am A Opt Image Sci Vis*, 20(7):1434–1448, July 2003.

[68] R. P. Lippmann. Speech recognition by machines and humans. *Speech Communication*, 22(1):1–15, 1997.

[69] Andreas Maurer. Learning to compare using operator-valued large-margin classifiers. In *Advances in Neural Information Processing Systems, LTCE Workshop*, 2006.

[70] Andreas Maurer. Learning similarity with operator-valued large-margin classifiers. *J. Mach. Learn. Res.*, 9:1049–1082, 2008.

[71] B.W. Mel. SEEMORE: Combining color, shape, and texture histogramming in a neurally inspired approach to visual object recognition. *Neural Comp.*, 9:777–804, 1997.

[72] N. Mesgarani, M. Slaney, and S. Shamma. Speech discrimination based on multiscale spectro-temporal features. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2004.

[73] P. Moreno, B. Raj, and R. Stern. A vector taylor series approach for environment-independent speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 733–736, 1996.

[74] J. F. Murray. and K. Kreutz-Delgado. Visual recognition and inference using dynamic overcomplete sparse learning. *Neural Computation*, 19(9):2301–2352, 2007.

[75] J. Mutch and G. Lowe. Multiclass object recognition with sparse, localized features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11–18, New York, NY, USA, June 2006.

[76] S.H. Nawab, T.F. Quatieri, and J.S. Lim. Signal reconstruction from short-time fourier transform magnitude. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 31(4):986–998, 1983.

[77] D.I. Perrett and M. Oram. Neurophysiology of shape processing. *Img. Vis. Comput.*, 11:317–333, 1993.

[78] T. Poggio and F. Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, Sep 1990.

[79] T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247(4945):978–82, 1990.

[80] T. Poggio and S. Smale. The mathematics of learning: Dealing with data. *Notices of the american Mathematical Society (AMS)*, 50(5), 2003.

[81] D. Povey, B. Kingsbury, L. Mangu, G. Saon, H. Soltau, and G. Zweig. fmpe: Discriminatively trained features for speech recognition. In *DARPA EARS RT-04 Workshop*, Palisades, NY, 2004.

[82] T.F. Quatieri and A.V. Oppenheim. Iterative techniques for minimum phase signal reconstruction from phase or magnitude. *IEEE Trans. Acoust., Speech, and Signal Proc.*, 29(6):1187–1193, 1981.

[83] M. Ranzato, F. Huang, Y. Boureau, and Y. LeCun. Unsupervised learning of invariant feature hierarchies, with application to object recognition. In *CVPR*, 2007.

[84] R.P. Rao and D.H. Ballard. Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects. *Nat. Neurosci.*, 2:79–87, 1999.

[85] R.P.N. Rao and D.H. Ballard. Dynamic-model of visual recognition predicts neural responseproperties in the visual-cortex. *Neural Computation*, 9(4):721–63, 1997.

[86] J.P. Rauschecker and B. Tian. Mechanisms and streams for processing of 'what' and 'where' in auditory cortex. *Proc. Natl. Acad. Sci. USA*, 97(22):11800–11806, 2000.

[87] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nat. Neurosci.*, 2:1019–1025, 1999.

[88] R. Rifkin. *Everything Old Is New Again: A Fresh Look at Historical Approaches to Machine Learning.* PhD thesis, Massachusetts Institute of Technology, 2002.

[89] R. Rifkin, J.Bouvrie, K. Schutte, S. Chikkerur, M. Kouh, T. Ezzat, and T. Poggio. Phonetic classification using hierarchical, feed-forward, spectro-temporal patch-based architectures. AI Memo 2007-007, MIT, Cambridge, MA, 2007.

[90] R. Rifkin, K. Schutte, D. Saad, J. Bouvrie, and J. Glass. Noise robust phonetic classification with linear regularized least squares and second-order features. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007.

[91] G.A. Rousselet, M.J. Mace, and M. Fabre-Thorpe. Is it an animal? Is it a human face? Fast processing in upright and inverted natural scenes. *J. Vision*, 3:440–455, 2003.

[92] N. Le Roux and Y. Bengio. Representational power of restricted boltzmann machines and deep belief networks. *Neural Computation*, 20(6):1631–1649, 2008.

[93] B. Schölkopf and A.J. Smola. *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.

[94] T. Serre, M. Kouh., C. Cadieu, U. Knoblich, G. Kreiman, and T. Poggio. A theory of object recognition: computations and circuits in the feedforward path of the ventral stream in primate visual cortex. AI Memo 2005-036 / CBCL Memo 259, MIT, Cambridge, MA, 2005.

[95] T. Serre, M. Kouh., C. Cadieu, U. Knoblich, G. Kreiman, and T. Poggio. A quantitative theory of immediate visual recognition. *Progress in Brain Research*, 165:33–56, 2007.

[96] T. Serre, A. Oliva, and T. Poggio. A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Science*, 104:6424–6429, 2007.

[97] T. Serre, M. Riesenhuber, J. Louie, and T. Poggio. On the role of object-specific features for real world object recognition. In S.-W. Lee, H. H. Buelthoff, and T. Poggio, editors, *Proc. of Biologically Motivated Computer Vision*, Lecture Notes in Computer Science, New York, 2002. Springer.

[98] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio. Robust object recognition with cortex-like mechanisms. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 29:411–426, 2007.

[99] T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In IEEE Computer Society Press, editor, *Proc. IEEE Conf. on Comput. Vision and Pattern Recognit.*, San Diego, 2005.

[100] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, Cambridge, 2004.

[101] N. Shental, T. Hertz, D. Weinshall, and M. Pavel. Adjustment learning and relevant component analysis. In *The Seventh European Conference on Computer Vision*, volume 4, pages 776–792, Copenhagen, Denmark, 2002.

[102] P. Simard, Y. LeCun, J.S. Denker, and B. Victorri. Transformation invariance in pattern recognition-tangent distance and tangent propagation. In *Neural Networks: Tricks of the Trade*, pages 239–274, London, UK, 1998. Springer-Verlag.

[103] Rita Singh, Richard M. Stern, and Bhiksha Raj. Model compensation and matched condition methods for robust speech recognition. In Gillian Davis, editor, *Noise Reduction in Speech Applications (Electrical Eng., series)*, chapter 10, pages 221–278. CRC Press LLC, USA, 2002.

[104] Rita Singh, Richard M. Stern, and Bhiksha Raj. Signal and feature compensation methods for robust speech recognition. In Gillian Davis, editor, *Noise Reduction in Speech Applications (Electrical Eng., series)*, chapter 9, pages 221–278. CRC Press LLC, USA, 2002.

[105] S. Smale, L. Rosasco, J. Bouvrie, A. Caponnetto, and T. Poggio. Mathematics of the neural response. CBCL paper 276 / CSAIL technical report 2008-070, MIT, Cambridge, MA, 2008.

[106] S.Tibrewala and H. Hermansky. Subband based recognition of noisy speech. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1997.

[107] S. Sukittanon, L.E. Atlas, and J.W. Pitton. Modulation-scale analysis for content identification. *Signal Processing, IEEE Transactions on*, 52(10):3023–3035, Oct. 2004.

[108] F.E. Theunissen, K. Sen, and A. Doupe. Spectral-temporal receptive fields of nonlinear auditory neurons obtained using natural sounds. *J. Neuro.*, 20:2315–2331, 2000.

[109] S.J. Thorpe. Ultra-rapid scene categorisation with a wave of spikes. In *BMCV*, 2002.

[110] S.J. Thorpe and M. Fabre-Thorpe. Seeking categories in the brain. *Science*, 291:260–263, 2001.

[111] S.J. Thorpe, D. Fize, and C. Marlot. Speed of processing in the human visual system. *Nature*, 381:520–522, 1996.

[112] I. W. Tsang, P. M. Cheung, and J. T. Kwok. Kernel relevant component analysis for distance metric learning. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN'05)*, pages 954–959, Montreal, Canada, July 2005.

[113] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermdediate complexity and their use in classification. *Nat. Neurosci.*, 5(7):682–687, 2002.

[114] R. VanRullen and C. Koch. Visual selective behavior can be triggered by a feed-forward process. *J. Comp. Neurol.*, 15:209–217, 2003.

[115] V. N. Vapnik. *Statistical learning theory.* Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons Inc., New York, 1998. A Wiley-Interscience Publication.

[116] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1096–1103, New York, NY, USA, 2008. ACM.

[117] G. Wallis and E. T. Rolls. A model of invariant object recognition in the visual system. *Prog. Neurobiol.*, 51:167–194, 1997.

[118] H. Wersing and E. Koerner. Learning optimized features for hierarchical models of invariant recognition. *Neural Comp.*, 15(7):1559–1588, 2003.

[119] J. Weston, F. Ratle, and R. Collobert. Deep learning via semi-supervised embedding. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 1168–1175, New York, NY, USA, 2008. ACM.

[120] G. Yu and J.-J. E. Slotine. Fastwavelet-based visual classification. In *Proceedings 19th International Conference on Pattern Recognition (ICPR 2008), December 8-11, 2008, Tampa, Florida, USA*, pages 1–5. IEEE, 2008.

[121] Q. Zhu and A. Alwan. An efficient and scalable 2d dct-based feature coding scheme for remote speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001.