November 17, 2008

CBCL Paper
November 17, 2008

Mathematics of the neural response

Steve Smale[‡], Lorenzo Rosasco[†♯], Jake Bouvrie[†], Andrea Caponnetto[⋄], Tomaso Poggio[†]

[‡] *Toyota Technological Institute at Chicago and University of California, Berkeley*
[⋄] *Department of Mathematics, City University of Hong Kong*
[♯] *DISI, Università di Genova*
[†] *CBCL, McGovern Institute, Artificial Intelligence Lab, BCS, MIT*

Abstract

We propose a natural image representation, the neural response, motivated by the neuroscience of the visual cortex. The inner product defined by the neural response leads to a similarity measure between functions which we call the derived kernel. Based on a hierarchical architecture, we give a recursive definition of the neural response and associated derived kernel. The derived kernel can be used in a variety of application domains such as classification of images, strings of text and genomics data.

1 Introduction

The goal of this paper 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 [19, 4, 11, 5]. In practice, problem specific metrics are often used [16]. In this paper we 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 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. 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. This model, while purely mathematical, 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 work in this paper sets the stage for further developments towards a theory of vision. One might consider especially two complementary directions, one empirical, the other mathematical. The empirical requires 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 6 we show the results of preliminary 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 4.1 that if the architecture is rich enough and there are sufficient templates ("neurons") then indeed the answer is a sharp "Yes" (up-to reversal and "checkerboard" patterns). We show under quite mild assumptions that the neural response is invariant under rotations, and for strings, is reversal invariant. In Section 5 we suggest that the Shannon entropy is a promising tool for obtaining a systematic picture.

Our work seeks to establish a theoretical foundation for recent models designed on the basis of anatomical and physiological data describing the primate visual cortex. These models are beginning to quantitatively account for a host of novel data and to provide human-level performance on rapid categorization of complex imagery (see [13, 15, 14] and references therein). These efforts are the most recent examples of a family of biologically-inspired architectures, see for example [7, 10, 20], and related computer vision systems [8, 18]. The hierarchical organization of such models – and of the cortex itself – remains a challenge for learning theory as most "learning algorithms", as described in [9], correspond to one-layer architectures. In this paper, we attempt to formalize the basic hierarchy of computations underlying information processing in the visual cortex. 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.

Some preliminary results appeared in [17], whereas related developments can be found in [2]. In the Appendix we establish detailed connections with the model in [15] and identify a key difference with the model developed in this paper.

The paper is organized as follows. We begin by introducing the definitions of the neural response and derived kernel in Section 2. We study invariance properties of the neural response in Section 3 and analyze discrimination properties in a one-dimensional setting in Section 4. In Section 5 we suggest that Shannon entropy can be used to understand the discrimination properties of the neural response. Finally, we conclude with preliminary experiments in Section 6.

2 Derived Kernel and Neural Response

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

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 subdomains 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 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 paper are limited to translations; see remarks in Section 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 discrete, finite and endowed with the uniform probability measure.

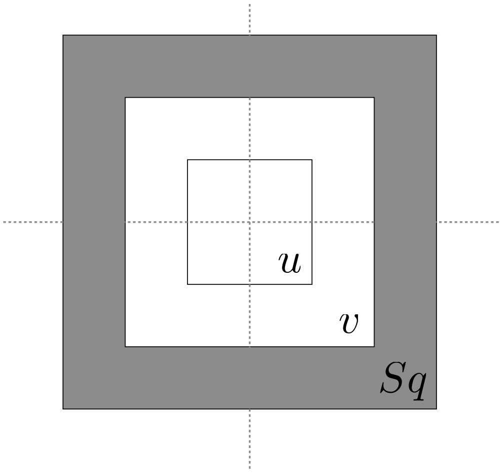

Figure 1: *Nested patch domains.*

The following fundamental assumption relates function spaces and transformation spaces.

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 [1]. 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 paper 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

$$\widehat{K}(x, x') = \frac{K(x, x')}{\sqrt{K(x,x)K(x',x')}}. \tag{1}$$

Clearly in this case $\widehat{K}$ is a reproducing kernel and $\widehat{K}(x,x) \equiv 1$ for all $x \in X$.

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, namely

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

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

$$N_v(f)(t) = \max_{h \in H} \widehat{K}_u(f \circ h, t), \tag{2}$$

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

$$K_v(f, g) = \langle N_v(f), N_v(g) \rangle_{L^2(T_u)}, \tag{3}$$

and can be normalized according to (1) to obtain the kernel $\widehat{K}_v$.

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

$$N_{Sq}(f)(t) = \max_{h \in H} \widehat{K}_v(f \circ h, t), \tag{4}$$

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

$$K_{Sq}(f, g) = \langle N_{Sq}(f), N_{Sq}(g) \rangle_{L^2(T_v)}, \tag{5}$$

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 mathematical induction.

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

We add some remarks.

Remarks

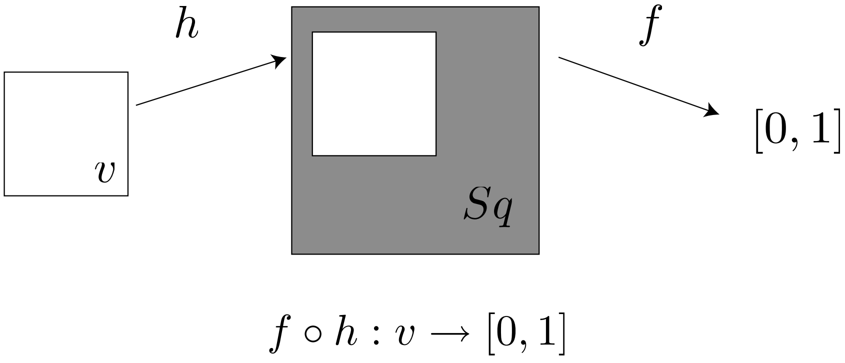

Figure 2: *A transformation "restricts" an image to a specific patch.*

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

- To make sense of the normalization (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 $g \in \mathrm{Im}(v_{n-1})$ is a patch of a function $f \in \mathrm{Im}(v)$ at layer $n-1$, or simply a *function patch* of f, if $g = f \circ h$ for some $h \in H_{n-1}$. 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
$$d(f,g)^2 = \widehat{K}(f,f) + \widehat{K}(g,g) - 2\widehat{K}(f,g) = 2\Big(1 - \widehat{K}(f,g)\Big). \tag{6}$$
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.

- 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.

2.3 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.

2.4 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,

$$\widehat{K}(f,g) = \left\langle \widehat{N}(f), \widehat{N}(g) \right\rangle_{L^2(T)}. \tag{7}$$

A map satisfying the above condition is referred to as a *feature map* in the language of kernel methods [11]. 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:

$$d(f,g)^2 = \|\widehat{N}(f) - \widehat{N}(g)\|^2_{L^2(T)} = 2\left(1 - \langle \widehat{N}(f), \widehat{N}(g)\rangle_{L^2(T)}\right), \tag{8}$$

where we used the fact that the neural responses are normalized. Note that the above distance function is a restatement of (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 (7) and (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:

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

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

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

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

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

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 $t' \in T_u$.

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:

Assumption 1. *For all $r \in \mathcal{R}$, and $h \in H$, there exists a unique $h' \in H$ such that*

$$r \circ h = h' \circ r. \tag{9}$$

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 true. Examples in the case of strings are given in the next section.

Given the above assumption we can state the following result.

Proposition 3.1. *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 $m \leq n$.

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 $u \in \operatorname{Im}(v_{m-1})$. 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 1 and the third follows from the inductive hypothesis. □

The following result is then immediate:

Corollary 3.1. *Let $\mathcal{Q}$, $\mathcal{U}$ be two families of transformations satisfying Assumption 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 $r \in \mathcal{R}$, $f \in \operatorname{Im}(v_m)$ and $m \leq n$.

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

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

□

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

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

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

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

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

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

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

Proof. We first show that Assumption 1 holds. Each translation is simply $h_a(x) = x + a$, and since the space of transformations contains all translations, Assumption 1 holds taking $h = h_a$, $r = \text{ref}_\theta$ and $h' = h_{a'}$, with $a' = \text{ref}_\theta(a)$. Since the initial kernel $\widehat{K}_1$ is invariant under reflections, Proposition 3.1 implies $\widehat{K}_m(f, f \circ \text{ref}) = 1$ for all $\text{ref} \in \mathcal{R}ef$, $f \in \text{Im}(v_m)$, with $m \leq n$.

Rotational invariance follows recalling that any rotation can be obtained out of two reflections using the formula $\text{rot}(2(\theta - \phi)) = \text{ref}_\theta \circ \text{ref}_\phi$, so that we can apply directly Corollary 3.1. □

We add the following remark.

Remark 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,*

$$\text{If } h_a \in H, \quad \text{then } \tilde{H}_a \subseteq H. \tag{10}$$

The next section discusses the case of one dimensional strings.

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 $v_m = \{1, \ldots, \ell\}$, $m \leq n$, and function spaces $\operatorname{Im}(v_m)$ 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 $\operatorname{Im}(v_m) = S^m$, 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 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 4.2 (Occurrence)**.** Let $f \in \operatorname{Im}(Sq)$. We say that $t \in \operatorname{Im}(v_{n-1})$ *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 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.

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 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 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 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 3.1 then ensures that $\widehat{K}_v(f, f \circ r) = 1$. □

The following remark is analogous to Remark 3.1.

Remark 4.1. *Inspecting the above proof one can see that the assumption on the spaces H can be relaxed. It suffices to assume that*

$$\text{If } h_i \in H, \quad \text{then } \; h_{\varphi(i)} \in H. \tag{11}$$

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 4.

Theorem 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.

Proposition 4.1. *Let $f, g \in \operatorname{Im}(v_m)$ 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 $\widehat{K}_n(f,g)$ cannot equal 1.

Since $\bar{t}$ occurs in f but *does not occur* in g, by Definition 4.2,

$$N_n(g)(\bar{t}) < 1 \quad \text{and} \quad N_n(f)(\bar{t}) = 1. \tag{12}$$

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

$$N_n(g)(t') = 1 \quad \text{and} \quad N_n(f)(t') \leq 1, \tag{13}$$

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

$$N_n(f)(t) = c \cdot N_n(g)(t), \quad t \in T_{n-1} \tag{14}$$

for some constant c.
Combining this requirement with conditions (12),(13) we find that

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

Thus, there is no such c and $\widehat{K}_n(f,g)$ 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 $\widehat{K}_n(f,g)$ again cannot equal 1. □

We can now prove Theorem 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 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

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

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:

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

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

$$(a+b+ab)(c+d+cd)(a+c+ac)(b+d+bd), \tag{15}$$

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 (15), 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 $i = 1, \dots, n-2$. 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). □

5 Entropy of the Neural response

We suggest that the concept of Shannon entropy [3] 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). This motivates introducing a few definitions, and recalling some elementary facts from information theory. Conversations with David McAllester and Greg Shakhnarovich were useful for this section.

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

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

and denote by $\widehat{N}_v^{**}\rho_v$ the corresponding push-forward measure on $\mathcal{O}$.

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 $S_\pi(\widehat{N}_v^{**}\rho_v)$ in a similar fashion.

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 4.1).

5.1 Short Appendix to Section 5

We briefly discuss how the development in the previous section relates to standard concepts (and notation) found in information theory [3]. 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. $\mathcal{X} = \{x_1, \ldots, x_d\}$ 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 $\{p_1, \ldots, p_d\} = \{\rho(x_1), \ldots, \rho(x_d)\}$. 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 $\operatorname{Im}(v)$ is a finite set. In this case we can define a (discrete) random variable $X = F$ with values in $\mathcal{X} = \operatorname{Im}(v) = \{f_1, \ldots, f_d\}$ and domain in some probability space (Ω, P) such that P is the pullback measure associated to the measure ρ_v on $\operatorname{Im}(v)$. Then $\{p_1, \ldots, p_d\} = \{\rho_v(f_1), \ldots, \rho_v(f_d)\}$, and

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

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

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

6 Empirical Analysis

The work described thus far was largely motivated by a desire to understand the empirical success of the model in [15, 14] when applied to numerous real-world recognition problems. The simplified setting we consider in this paper 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 [15, 14], 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.

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.

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

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

Algorithm 1 Neural response algorithm.

```
**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 $h \in H^g_m$ 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 if
    end for
  end for
  for $h \in H^g_{m+1}$ do
    for $t \in T_m$ do
      $C_m(h,t) = \max_{h' \in H_m} S_m(h \circ h', t)$
    end for
  end for
  $\widehat{C}_m = \mathrm{NORMALIZE}(C_m)$
end for
Return $\widehat{N}_n(f)(t) = \widehat{C}_{n-1}(h,t)$, with $h \in H^g_n$, $t \in T_{n-1}$
```

for any $1 \leq m \leq n-1$ where H^g_n 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 $h \in H^g_{m+1}$, $t \in T_m$. 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

$$\tau = \sum_{m=1}^{n-1} \Big(|H^g_m||T_m||T_{m-1}| + |H^g_{m+1}||H_m||T_m| \Big) \tag{16}$$

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.

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

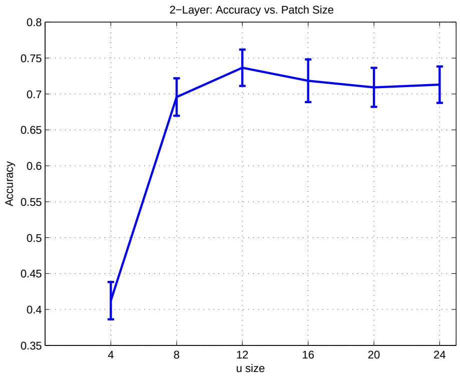

Figure 3: *2-Layer architecture, accuracy vs. patch sizes.*

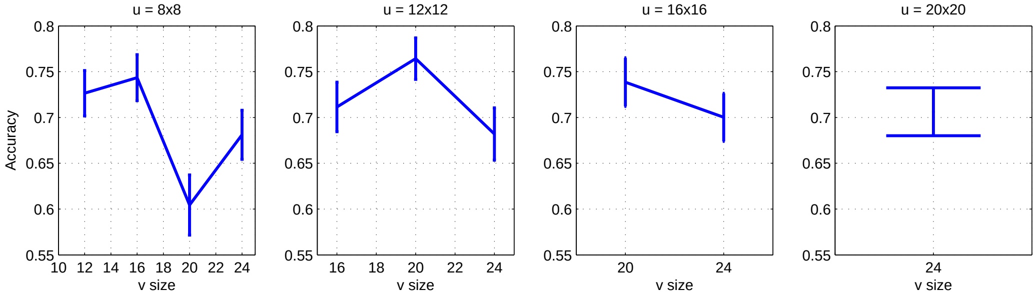

Figure 4: *3-Layer architecture, accuracy vs. patch sizes.*

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 (6)) and pairwise L^2 distances. The block structure that typifies these matrices argues graphically that the derived kernels are separating the different classes of images. Finally, we 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.

In all experiments we have used $Sq = 28 \times 28$ pixel grayscale images randomly selected from the MNIST dataset of handwritten digits [8]. 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

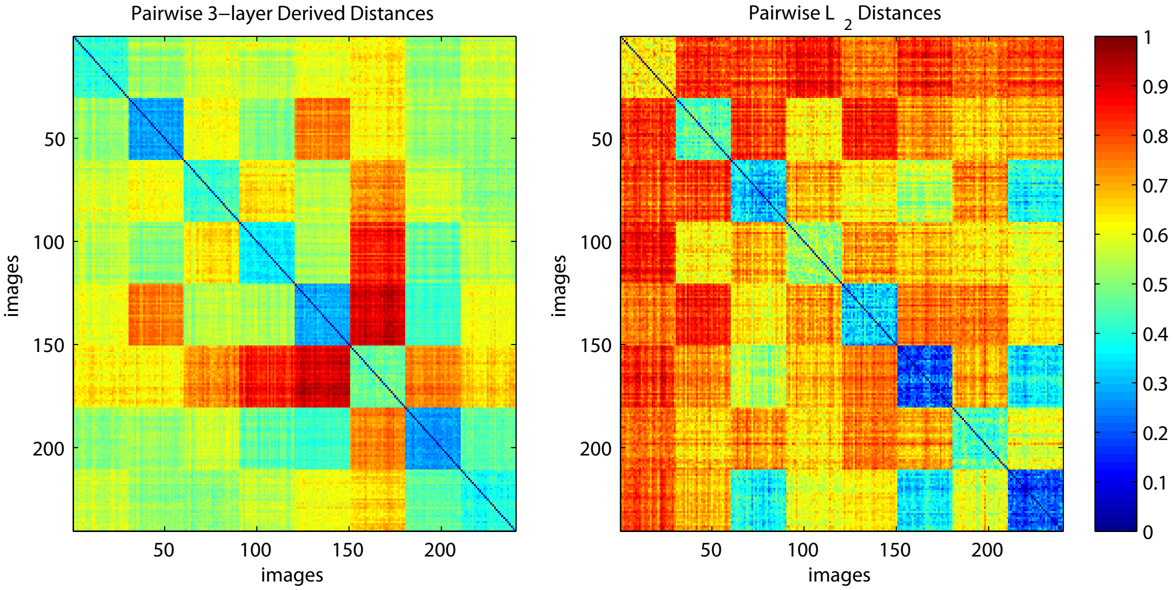

Figure 5: *Matrices of pairwise 3-Layer derived distances (left) and L^2 distances (right) for the set of 240 images from the database. Each group of 30 rows/columns correspond to images of the digits 2 through 9, in left-right and top-bottom order.*

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 10x10 pixels are large enough to include semi-circles and distinct stroke intersections, while larger templates, closer to 20x20, are seen to include nearly full digits where more discriminative structure is present.

In Figures 3 and 4 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 [6]. It is clear, however, from Figure 4 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 3 and 4: $u = 12 \times 12, v = 20 \times 20$. In practice, the patch size parameters can be chosen via cross validation or on a separate validation set distinct from the test set.

Figure 5 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

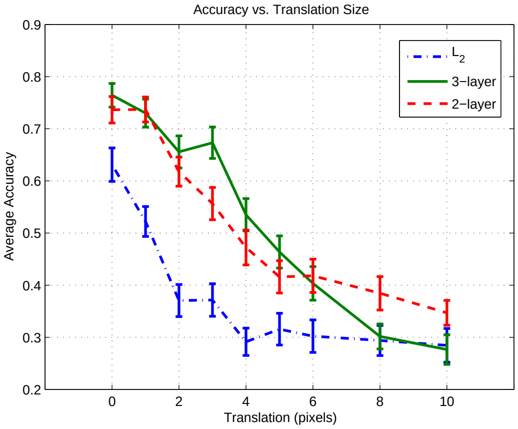

Figure 6: *Classification accuracy on artificially translated images.*

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 (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.

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 6 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

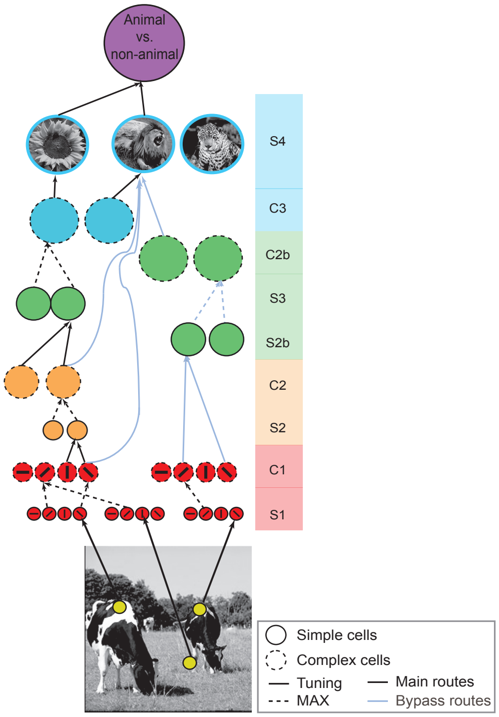

Figure 7: *The model of Serre et al [15]. We consider here the layers up to $C2$. (modified from [14])*

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.

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.

A Appendix: Derived Kernel and Visual Cortex

In this Appendix, we establish an exact connection between the neural response and the model of Serre et al. [14, 15, 12]. We consider an architecture comprised of $S1, C1, S2, C2$ layers as in the model, which is illustrated in Figure 7. 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

$$N_{S1}(f \circ h)(t) = K_u(f \circ h, t) \tag{17}$$

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

$$N_{C1}(f)(t) = \max_{h \in H}\{N_{S1}(f \circ h)(t)\} \tag{18}$$

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

$$K_w(f, g) = \sum_{h \in H_v} K_v(f \circ h, g \circ h) \tag{19}$$

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

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

$$N_{S2}(f \circ h)(t) = K_w(f \circ h, t), \tag{20}$$

with $f \in \mathrm{Im}(Sq)$, $h \in H_w$ 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

$$N_{C2}(f)(t) = \max_{h \in H}\{N_{S2}(f \circ h)(t)\} \tag{21}$$

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 neural response into two stages, where "convolution" steps (17) and (20) correspond to S units, and are followed by max operations (18) and (21) corresponding to C units.

- A key difference between the model in [15] and the development in this paper is the "extension" step (19). The model considered in this paper corresponds to $v = w$ and is not completely faithful to the model in [15, 14] 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 [15] 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

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

  where we used the (derived) distance

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

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

References

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

[2] A. Caponnetto, T. Poggio, and S. Smale. On a model of visual cortex: learning invariance and selectivity from image sequences. CBCL paper 272 / CSAIL technical report 2008-030, MIT, Cambridge, MA, 2008.

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

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

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

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

[7] 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.

[8] 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.

[9] T. Poggio and S. Smale. The mathematics of learning: Dealing with data. *Notices of the american Mathematical Society (AMS)*, 50(5), 2003.

[10] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nat. Neurosci.*, 2:1019–1025, 1999.

[11] B. Schölkopf and A.J. Smola. *Learning with Kernels.* MIT Press, Cambridge, MA, 2002.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, Cambridge, 2004.

[17] S. Smale, T. Poggio, A. Caponnetto, and J. Bouvrie. Derived distance: towards a mathematical theory of visual cortex. CBCL paper, MIT, Cambridge, MA, November 2007.

[18] 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.

[19] V. N. Vapnik. *Statistical learning theory.* John Wiley & Sons Inc., New York, 1998.

[20] H. Wersing and E. Koerner. Learning optimized features for hierarchical models of invariant recognition. *Neural Comp.*, 15(7):1559–1588, 2003.