

Towards a Theory of Hierarchical Learning: Derived Kernels and the Neural Response

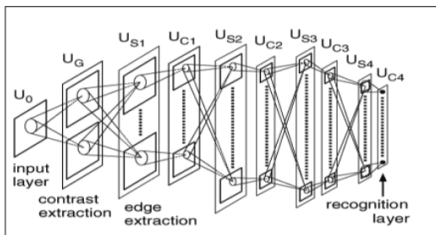
Jake Bouvrie

MIT 9.520 Class 23

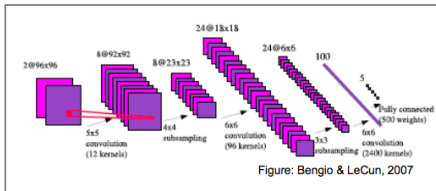
May 4 2009

Hierarchical/Deep Learning

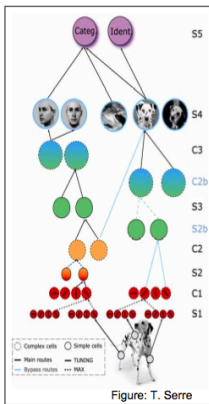
Neocognitron, from Fukushima et al., 1980



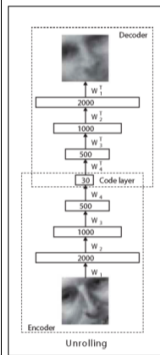
Convolutional Neural Networks (LeCun)



CBCL Model



Hinton's Deep Autoencoder



from: G. Hinton, Science 2007.



About this class

The goal of this class is to introduce a mathematical counterpart to the visual cortex model described in the previous two lectures.

In particular:

- We give a recursive definition of a similarity concept for images, and describe the underlying hierarchical architecture.
- We briefly describe some theoretical analyses and preliminary empirical results.

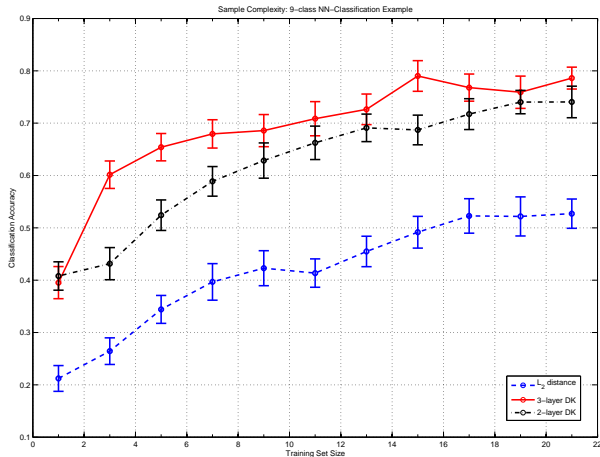
We'll spend most of the time establishing a formalism that can be used to explore deeper questions. This is half the battle... (cf. wavelets)

The material in this class is from:

S. Smale, L. Rosasco, J. Bouvrie, A. Caponnetto, and T. Poggio. "Mathematics of the Neural Response", CBCL Paper #276/MIT CSAIL Technical Report #TR2008-070, November, 2008

- 1 **Background**
- 2 Derived Kernels and the Neural Response
- 3 Connection to Neuroscience
- 4 Theoretical Analysis
- 5 Empirical Analysis

Hierarchical/Deep Learning: Empirical Motivation



9-class digits problem, nearest neighbor classifier, Euclidean distance vs. 3-layer derived distance ($u = 12$, $v = 20$, 500 templates/layer, 3-pixel image translations).

Why Hierarchical/Deep Learning?

- Chomsky's poverty of the stimulus argument: biological organisms can learn complex concepts and tasks from extraordinarily small empirical samples.
- Hypothesis: *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 also to greater energy and space efficiency.
- *Why do recent hierarchical models in vision work? Interpreting them can be perhaps as hard as interpreting the brain itself. We need a theory.*

Why Hierarchical/Deep Learning?

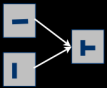

We are interested primarily in understanding invariance and discrimination properties of unsupervised hierarchies as a step towards answering larger questions, such as

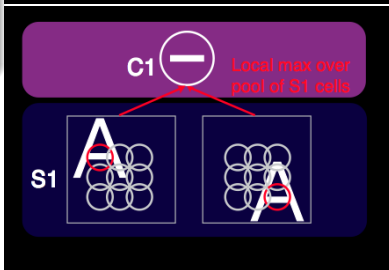
- ① When and why is a “deep” architecture preferred?
- ② For tasks that can be decomposed into a hierarchy of parts, how can we show that a supervised classifier trained using a hierarchical feature map will generalize better than an off-the-shelf non-hierarchical alternative?
- ③ Can we understand and cast learning in hierarchies using tools from statistical learning theory?

- 1 Background
- 2 **Derived Kernels and the Neural Response**
- 3 Connection to Neuroscience
- 4 Theoretical Analysis
- 5 Empirical Analysis

Towards a Theory

We will borrow concepts and operations underlying the visual cortex model.

Unit types	Pooling	Computation	Operation
Simple		Selectivity / template matching	Gaussian- tuning / and-like
Complex		Invariance	Soft-max / or-like



Defining a model

The ingredients needed to define the derived kernel consist of:

- A finite *architecture* of nested domains. We'll call them patches.
- A suitable family of *function spaces* defined on each patch.
- A set of *transformations* defined on patches.
- A set of *templates* which connect the mathematical model to a real world setting.

Defining a model

The ingredients needed to define the derived kernel consist of:

- A finite *architecture* of nested domains. We'll call them patches.
- A suitable family of *function spaces* defined on each patch.
- A set of *transformations* defined on patches.
- A set of *templates* which connect the mathematical model to a real world setting.

Defining a model

The ingredients needed to define the derived kernel consist of:

- A finite *architecture* of nested domains. We'll call them patches.
- A suitable family of *function spaces* defined on each patch.
- A set of *transformations* defined on patches.
- A set of *templates* which connect the mathematical model to a real world setting.

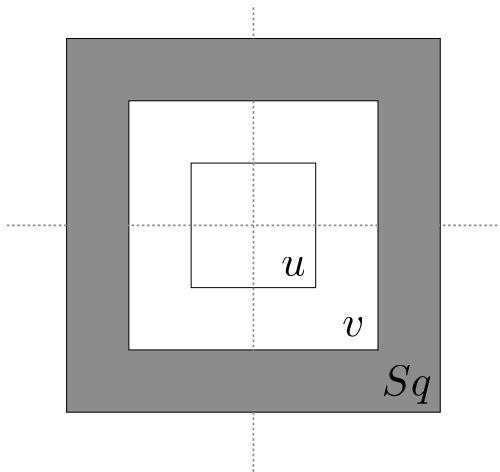
Defining a model

The ingredients needed to define the derived kernel consist of:

- A finite *architecture* of nested domains. We'll call them patches.
- A suitable family of *function spaces* defined on each patch.
- A set of *transformations* defined on patches.
- A set of *templates* which connect the mathematical model to a real world setting.

An Architecture of Patches

We first consider an architecture composed of *three* layers of patches: u, v and Sq in \mathbb{R}^2 , with $u \subset v \subset Sq$,



Images as Functions

We consider a function space on Sq , denoted by

$$\text{Im}(Sq) = \{f : Sq \rightarrow [0, 1]\},$$

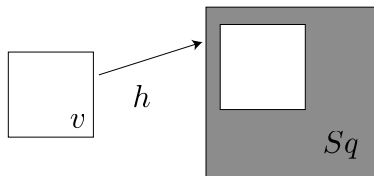
as well as the function spaces $\text{Im}(u)$, $\text{Im}(v)$ defined on subpatches u , v , respectively.

Functions can be interpreted as grey scale images when working with a vision problem for example.

Transformations

Next, we assume a set H_u of *transformations* that are maps from the smallest patch to the next larger patch

$$h : u \rightarrow v.$$



Similarly H_v with $h : v \rightarrow Sq$.

The sets of transformations are assumed to be finite.

These transformations act on the *domain* of a function (image).

Examples of transformations are translations, scalings and rotations...

Translations and Scalings

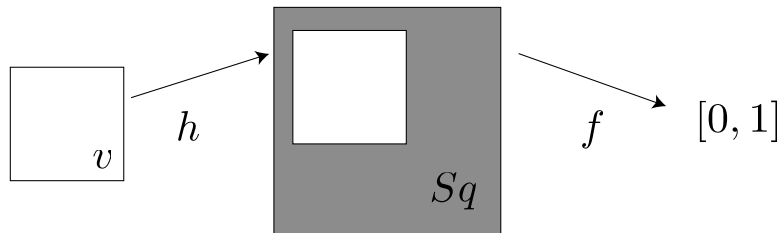
we have transformations of the form $h = h_\beta h_\alpha$ with

$$h_\alpha(x) = \alpha x, \text{ and } h_\beta(x') = x' + \beta,$$

where $\alpha \in \mathbb{R}$ and $\beta \in \mathbb{R}^2$ is such that $h_\beta h_\alpha(u) \subset v$.

Interpretation

In the vision interpretation, a translation h can be thought of as moving the image over the “receptive field” v



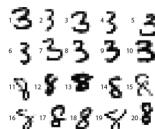
$$f \circ h : v \rightarrow [0, 1]$$

Figure: A transformation “restricts” an image to a specific patch.

Templates

Template sets are finite,
 $T_u \subset \text{Im}(u)$ and $T_v \subset \text{Im}(v)$

- they are image patches sampled from some set of unlabeled images.
- link the mathematical development to real world problems.



The space of images can be endowed with a “mother” probability measure ρ . Templates can be seen as images frequently encountered in the early stages of life.

Reproducing Kernel

Given a set X , a function $K : X \times X \rightarrow \mathbb{R}$ is a reproducing kernel if it is a symmetric and positive definite kernel, i.e.

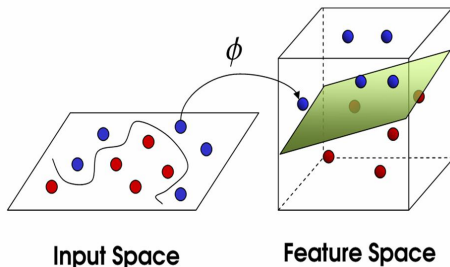
$$\sum_{i,j=1}^n \alpha_i \alpha_j K(x_i, x_j) \geq 0,$$

for any $n \in \mathbb{N}$, $x_1, \dots, x_n \in X$ and $\alpha_1, \dots, \alpha_n \in \mathbb{R}$.

Dot Products and Feature map

Consider a feature map:

$$\Phi : X \rightarrow \mathcal{F}$$



Inner product kernels are an instance of reproducing kernels:

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle$$

is a reproducing kernel.

We assume $K(x, x) \neq 0$ for all $x \in X$ and let

$$\hat{K}(x, x') = \frac{K(x, x')}{\sqrt{K(x, x)K(x', x')}}.$$

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

- Allows interpretation of and comparison between different instances.
- Is nice for correspondence with a distance.

On the normalization

To make sense of the normalization we rule out the functions such that $K(f, f)$ is zero.

This assumption is quite natural in the context of images:

If $K(f, f)$ is zero, the neural responses of f is identically zero at *all* possible templates by definition:

“one can’t see the contents of the image”.

On the normalization

To make sense of the normalization we rule out the functions such that $K(f, f)$ is zero.

This assumption is quite natural in the context of images:

If $K(f, f)$ is zero, the neural responses of f is identically zero at *all* possible templates by definition:

“one can’t see the contents of the image”.

Construction

We'll give a bottom-up description of a three layer architecture before giving the general recursive definition.

Initial Kernel

Consider a *normalized* non-negative valued reproducing kernel on $\text{Im}(u) \times \text{Im}(u)$ denoted by $\widehat{K}_u(f, g)$.

example

Consider the inner product of square integrable functions on u

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

Consider a *normalized* non-negative valued reproducing kernel on $\text{Im}(u) \times \text{Im}(u)$ denoted by $\widehat{K}_u(f, g)$.

example

Consider the inner product of square integrable functions on u

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

Neural Response

We define the *neural response* of f at t :

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

where $f \in \text{Im}(v)$, $t \in T_u$ and $H = H_u$.

NOTE: f is not the whole image here.



Neural Response

We define the *neural response* of f at t :

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

where $f \in \text{Im}(v)$, $t \in T_u$ and $H = H_u$.

NOTE: f is not the whole image here.



Neural Response (cont.)

By denoting with $N = |T_u|$ the cardinality of the template set T_u , we can interpret the neural response as a vector in \mathbb{R}^N ,

$$f \in \text{Im}(v) \longmapsto (N_v(f)(t_1), N_v(f)(t_2), \dots, N_v(f)(t_N)).$$

This is just the collection of best responses of each template within the *sub-patch* $f \in \text{Im}(v)$.

If K_u is the Euclidean dot-product, and H_u is all translations: compare to normalized cross-correlation.

Derived Kernel

The *derived kernel* is just the corresponding inner product in $L^2(T_u) = \mathbb{R}^{|T_u|}$ between neural responses, normalized by $\frac{1}{|T_u|}$

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

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

and can be normalized to obtain the kernel \hat{K}_v .

This is the correlation in the pattern of similarities to templates.

Second Layer

We now repeat the process:

second layer neural response

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

where $f \in \text{Im}(Sq)$, $t \in T_v$ and $H = H_v$.

derived kernel on $\text{Im}(Sq) \times \text{Im}(Sq)$

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

where $\langle \cdot, \cdot \rangle_{L^2(T_v)}$ is the L^2 inner product.

As before, we normalize K_{Sq} to obtain the final derived kernel \widehat{K}_{Sq} .

Second Layer

We now repeat the process:

second layer neural response

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

where $f \in \text{Im}(Sq)$, $t \in T_v$ and $H = H_v$.

derived kernel on $\text{Im}(Sq) \times \text{Im}(Sq)$

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

where $\langle \cdot, \cdot \rangle_{L^2(T_v)}$ is the L^2 inner product.

As before, we normalize K_{Sq} to obtain the final derived kernel \widehat{K}_{Sq} .

Second Layer

We now repeat the process:

second layer neural response

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

where $f \in \text{Im}(S_q)$, $t \in T_v$ and $H = H_v$.

derived kernel on $\text{Im}(S_q) \times \text{Im}(S_q)$

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

where $\langle \cdot, \cdot \rangle_{L^2(T_v)}$ is the L^2 inner product.

As before, we normalize K_{S_q} to obtain the final derived kernel \widehat{K}_{S_q} .

Recursive Definition

For a general n layer architecture $v_1 \subset v_2 \subset \dots \subset v_n = Sq$, let $K_n = K_{v_n}$ and $H_n = H_{v_n}$, $T_n = T_{v_n}$.

Definition

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

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

where

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

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

Recursive Definition

For a general n layer architecture $v_1 \subset v_2 \subset \dots \subset v_n = Sq$, let $K_n = K_{v_n}$ and $H_n = H_{v_n}$, $T_n = T_{v_n}$.

Definition

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

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

where

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

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

Neural Response

The normalized neural response provides a *representation* for any function $f \in \text{Im}(Sq)$.

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

with $T = T_{n-1}$.

The normalization for N is that implied by the normalization of K :

$$\hat{N}(f) = \frac{N(f)}{\|N(f)\|_{L^2(T)}}$$

where $\|x\|_{L^2(T)} = \sqrt{\langle x, x \rangle_{L^2(T)}} = \sqrt{\frac{1}{|T|} \langle x, x \rangle_{\mathbb{R}^{|T|}}}$.

Properties

We'll now describe properties emerging out of the previous definitions.

Derived Distance

The derived kernel naturally defines a derived distance d on the space of images.

$$d^2(f, g) = \|\hat{N}(f) - \hat{N}(g)\|^2 = 2(1 - \hat{K}(f, g))$$

(since $\hat{K}(f, f) = 1$ for all f)

Clearly, as the kernel “similarity” approaches its maximum value of 1, the distance goes to 0.

The Neural Response is a Self Consistent Definition

The neural response at a given layer can be expressed in terms of the neural responses at the previous layer

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

with $H = H_v$, $T' = T_u$ and $T = T_v$.

The Neural Response is a Self Consistent Definition

In vector notation,

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

where the \max operation is assumed to apply elementwise.

Encoding Operator

The operator Π_v is seen 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 $t' \in T_u$.

We can also define $\Pi_v : L^2(T_u) \rightarrow L^2(T_v)$ more abstractly, by saying

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

for $F \in L^2(T_u), t \in T_v$.

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

This perspective highlights the action of the hierarchy as alternating **pooling** and **filtering** steps, realized by the \max and the Π operators respectively.

We can integrate unsupervised learning into the model via the Π operators. For example,

- A new Π can be constructed from the PCA decomposition of the original Π .
- Π could be represented in terms of the eigenfunctions of the Laplacian.
- Sparse representations can be enforced (cf. sparse coding ideas in computational neuroscience and signal processing).

- 1 Background
- 2 Derived Kernels and the Neural Response
- 3 **Connection to Neuroscience**
- 4 Theoretical Analysis
- 5 Empirical Analysis

Neural Response vs. Simple and Complex Cells

The two key steps in the definition of neural response correspond to simple and complex cells in the visual cortex (and the CBCL model):

- S: inner products with the templates.
- C: \max over the set of translations.

Simple Cells at the First Layer

Given an initial kernel K_u , let

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

with $f \in \text{Im}(v)$, $h \in H_u$ and $t \in T_u$.

$N_{S1}(f \circ h)(t)$ 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”.

Simple Cells at the First Layer

Given an initial kernel K_u , let

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

with $f \in \text{Im}(v)$, $h \in H_u$ and $t \in T_u$.

$N_{S1}(f \circ h)(t)$ 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”.

Complex Cells at the First Layer

The neural response is given by

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

with $f \in \text{Im}(v)$, $H = H_u$ and $t \in T_u$ so that $N_{C1} : \text{Im}(v) \rightarrow \mathbb{R}^{|T_u|}$.

$N_{C1}(f)(t)$ corresponds to the response of a $C1$ cell with template t and receptive field corresponding to v .

Complex Cells at the First Layer

The neural response is given by

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

with $f \in \text{Im}(v)$, $H = H_u$ and $t \in T_u$ so that $N_{C1} : \text{Im}(v) \rightarrow \mathbb{R}^{|T_u|}$.

$N_{C1}(f)(t)$ corresponds to the response of a $C1$ cell with template t and receptive field corresponding to v .

Gaussian Tuning

In the model gaussian tuning can replace normalization.

This latter case corresponds to considering

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

where we used the (derived) distance

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

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

Gaussian Tuning

In the model gaussian tuning can replace normalization.
This latter case corresponds to considering

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

where we used the (derived) distance

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

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

- 1 Background
- 2 Derived Kernels and the Neural Response
- 3 Connection to Neuroscience
- 4 **Theoretical Analysis**
- 5 Empirical Analysis

Formulating the model in careful, mathematical terms was the first step towards a comprehensive theory.

Now we can start looking at invariance, discrimination, and other properties that emerge from our definitions:

- selectivity vs invariance
- sample complexity/poverty of the stimulus
- model selection questions

Invariance of the Neural Response

We can consider *invariance* of the (normalized) neural response with respect to some set of domain transformations

$$\mathcal{R} = \{r \mid r : v \rightarrow v\}.$$

Invariance

$$\hat{N}(f) = \hat{N}(f \circ r)$$

(or equivalently $\hat{K}_n(f \circ r, f) = 1$).

Translations are cheap, but adding more explicit transformations to H gets expensive...

Invariance of the Neural Response

We can consider *invariance* of the (normalized) neural response with respect to some set of domain transformations

$$\mathcal{R} = \{r \mid r : v \rightarrow v\}.$$

Invariance

$$\hat{N}(f) = \hat{N}(f \circ r)$$

(or equivalently $\hat{K}_n(f \circ r, f) = 1$).

Translations are cheap, but adding more explicit transformations to H gets expensive...

Assumption

Assumption

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

$$r \circ h = h' \circ r$$

and there exists a unique $h'' \in H$ such that

$$h \circ r = r \circ h'' .$$

...can be expressed in terms of orbits of translations under the action of conjugation by e.g. elements of the groups O_n or SO_n .

Assumption

In the case of vision for example, we can think of \mathcal{R} as reflections and H as translations:

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.

Theorem

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

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

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

Global invariance from local invariance!

Reflections & Rotations

For 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 .

Moreover, let $\mathcal{R}ot$ denote the space of coordinate rotations about the origin.

Invariance to Reflections and Rotations

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

Theorem

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 $m \leq n$.

Invariance to Reflections and Rotations

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

Theorem

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 $m \leq n$.

One dimensional strings

- An n -string is a function from an index set $\{1, \dots, n\}$ to some finite alphabet S .
- Patches that are sets of indices $v_m = \{1, \dots, \ell\}$, $m \leq n$.
- Function spaces $\text{Im}(v_m)$ are strings of length m . The first layer consists of single characters.
- We consider the initial kernel

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

where $f, g \in S$.

One dimensional strings

- An n -string is a function from an index set $\{1, \dots, n\}$ to some finite alphabet S .
- Patches that are sets of indices $v_m = \{1, \dots, \ell\}$, $m \leq n$.
- Function spaces $\text{Im}(v_m)$ are strings of length m . The first layer consists of single characters.
- We consider the initial kernel

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

where $f, g \in S$.

One dimensional strings

- An n -string is a function from an index set $\{1, \dots, n\}$ to some finite alphabet S .
- Patches that are sets of indices $v_m = \{1, \dots, \ell\}$, $m \leq n$.
- Function spaces $\text{Im}(v_m)$ are strings of length m . The first layer consists of single characters.
- We consider the initial kernel

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

where $f, g \in S$.

One dimensional strings

- An n -string is a function from an index set $\{1, \dots, n\}$ to some finite alphabet S .
- Patches that are sets of indices $v_m = \{1, \dots, \ell\}$, $m \leq n$.
- Function spaces $\text{Im}(v_m)$ are strings of length m . The first layer consists of single characters.
- We consider the initial kernel

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

where $f, g \in S$.

Let $r : \{1, \dots, m\} \rightarrow \{1, \dots, m\}$ denote reversal of a string.

Theorem

If the spaces H at all layers contain all possible translations then

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

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

Discrimination Results (opposite of invariance)

Consider an exhaustive architecture: $v_m = \{1, \dots, m\}$, $T_m = \text{Im}(v_m) = S^m$, for $m = 1, \dots, n$ and transformations are all possible translations. Take the maximum number of layers, $n - 1$ for length n input.

Theorem

If f, g are n -strings and $\widehat{K}_n(f, g) = 1$ then:

- f, g are the same string
- one is the reversal of the other
- f, g are the “checkerboard” pattern: $f = ababa \dots$, $g = babab \dots$, with f and g odd length strings.

What happens with other architectures?

Discrimination Results (opposite of invariance)

Consider an exhaustive architecture: $v_m = \{1, \dots, m\}$, $T_m = \text{Im}(v_m) = S^m$, for $m = 1, \dots, n$ and transformations are all possible translations. Take the maximum number of layers, $n - 1$ for length n input.

Theorem

If f, g are n -strings and $\widehat{K}_n(f, g) = 1$ then:

- f, g are the same string
- one is the reversal of the other
- f, g are the “checkerboard” pattern: $f = ababa \dots, g = babab \dots$, with f and g odd length strings.

What happens with other architectures?

Discrimination Results (opposite of invariance)

Consider an exhaustive architecture: $v_m = \{1, \dots, m\}$, $T_m = \text{Im}(v_m) = S^m$, for $m = 1, \dots, n$ and transformations are all possible translations. Take the maximum number of layers, $n - 1$ for length n input.

Theorem

If f, g are n -strings and $\widehat{K}_n(f, g) = 1$ then:

- f, g are the same string
- one is the reversal of the other
- f, g are the “checkerboard” pattern: $f = ababa \dots$, $g = babab \dots$, with f and g odd length strings.

What happens with other architectures?

Discrimination Results (opposite of invariance)

Consider an exhaustive architecture: $v_m = \{1, \dots, m\}$, $T_m = \text{Im}(v_m) = S^m$, for $m = 1, \dots, n$ and transformations are all possible translations. Take the maximum number of layers, $n - 1$ for length n input.

Theorem

If f, g are n -strings and $\widehat{K}_n(f, g) = 1$ then:

- f, g are the same string
- one is the reversal of the other
- f, g are the “checkerboard” pattern: $f = ababa \dots$, $g = babab \dots$, with f and g odd length strings.

What happens with other architectures?

- 1 Background
- 2 Derived Kernels and the Neural Response
- 3 Connection to Neuroscience
- 4 Theoretical Analysis
- 5 **Empirical Analysis**

Motivation

- The work described thus far was motivated in part by a desire to understand the empirical success of recent models of visual cortex.
- The simplified setting we considered trades complexity and faithfulness to biology for a more controlled, analytically tractable framework.

Implementation

A direct implementation of the architecture following the recursive definition of the derived kernel appears to be exponential in the number of layers.

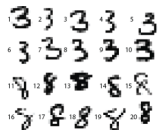
A *bottom-up* algorithm linear in the number of layers can be obtained by consolidating and reordering the computations.

Implementation

A direct implementation of the architecture following the recursive definition of the derived kernel appears to be exponential in the number of layers.

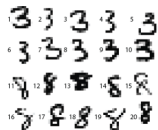
A *bottom-up* algorithm linear in the number of layers can be obtained by consolidating and reordering the computations.

Classification Task



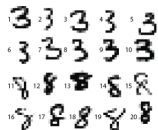
- $S_q = 28 \times 28$ pixel grayscale images from the MNIST dataset of handwritten digits
- eight classes of images: 2s through 9s
- training sets contain 5 examples per class, test sets contain 30 examples per class
- 1-NN classifier
- results averaged over 50 random trials

Classification Task



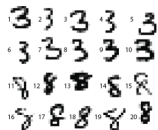
- $S_q = 28 \times 28$ pixel grayscale images from the MNIST dataset of handwritten digits
- eight classes of images: 2s through 9s
- training sets contain 5 examples per class, test sets contain 30 examples per class
- 1-NN classifier
- results averaged over 50 random trials

Classification Task



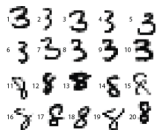
- $Sq = 28 \times 28$ pixel grayscale images from the MNIST dataset of handwritten digits
- eight classes of images: 2s through 9s
- training sets contain 5 examples per class, test sets contain 30 examples per class
- 1-NN classifier
- results averaged over 50 random trials

Classification Task



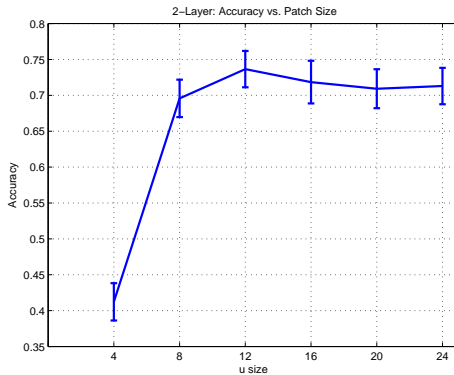
- $Sq = 28 \times 28$ pixel grayscale images from the MNIST dataset of handwritten digits
- eight classes of images: 2s through 9s
- training sets contain 5 examples per class, test sets contain 30 examples per class
- 1-NN classifier
- results averaged over 50 random trials

Classification Task

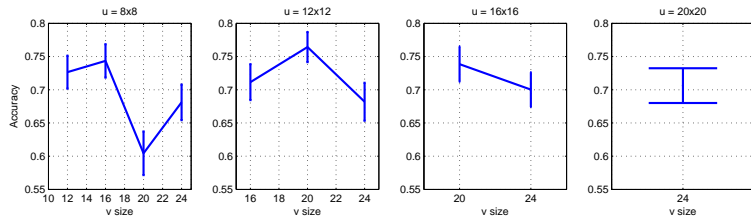


- $Sq = 28 \times 28$ pixel grayscale images from the MNIST dataset of handwritten digits
- eight classes of images: 2s through 9s
- training sets contain 5 examples per class, test sets contain 30 examples per class
- 1-NN classifier
- results averaged over 50 random trials

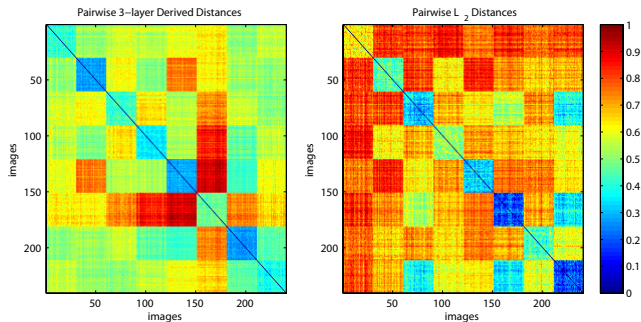
Patch sizes?



Patch sizes? (cont.)

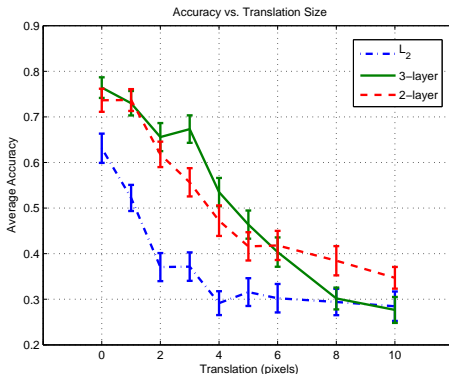


Confusion Matrix



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.

Number of Layers?



Classification accuracy on artificially translated images.

The results at zero confirm that the hierarchical assumption holds.

Summary

- We provided a compact mathematical description of a hierarchical model, based on recent feedforward models of the visual cortex.
- A similarity kernel was recursively defined
- Analysis of invariance/discrimination properties was provided.

Theory is just at the beginning and many questions remain.

For example:

- Can we show that more layers are better than one? When? Sample complexity...
- Can we learn the templates (rather than just sample them)?
- Is the max operation really crucial? Can it be replaced by some other operation (average...)?
- Discrimination/Invariance properties vs. architecture. **Parameter choices are theory questions.**