LETTER Communicated by Thomas Serre

Suitability of V1 Energy Models for Object Classification

James Bergstra
james.bergstra@umontreal.ca
Yoshua Bengio
bengioy@iro.umontreal.ca
Jérôme Louradour
louradoj@iro.umontreal.ca
Département d'Informatique, Université de Montréal, Montréal,
Québec H3T IJ4, Canada

Simulations of cortical computation have often focused on networks built from simplified neuron models similar to rate models hypothesized for V1 simple cells. However, physiological research has revealed that even V1 simple cells have surprising complexity. Our computational simulations explore the effect of this complexity on the visual system's ability to solve simple tasks, such as the categorization of shapes and digits, after learning from a limited number of examples. We use recently proposed high-throughput methodology to explore what axes of modeling complexity are useful in these categorization tasks. We find that complex cell rate models learn to categorize objects better than simple cell models, and without incurring extra computational expense. We find that the squaring of linear filter responses leads to better performance. We find that several other components of physiologically derived models do not yield better performance.

1 Introduction

An important role of the visual system is to transform retinal signals so that object categorization can be carried out in higher cortical areas such as V4 and IT (Dayan & Abbott, 2001; Serre et al., 2007). However, it remains unclear, even in V1, what transformations individual neurons perform (Doi & Lewicki, 2007; Haüsser & Mel, 2003; Olshausen & Field, 2005).

Many rate models have been proposed for V1 based on studies of spike-triggered averaging (Hubel & Wiesel, 1962; Nykamp & Ringach, 2002). The simplest rate model is a linear filter whose output has been rectified to be nonnegative and bounded. Rectification has been carried out using the logistic sigmoid (Wilson & Cowan, 1972; Rumelhart, McClelland, & PDP Research Group, 1986; Nykamp & Ringach, 2002; Kouh & Poggio, 2008) or the Naka-Rushton equation (Naka & Rushton, 1966; Heeger, 1992) to describe V1 simple cells (first described in Hubel & Wiesel, 1962). Other cells (called complex cells) in V1 have been found to respond robustly to narrow bars of light in nearby positions but not to their superposition. This behavior cannot be explained by a linear model. The classic model for complex cells is the energy model, in which a complex cell response is modeled by a sum of squared responses from a number of afferent simple cells (Adelson & Bergen, 1985; Dayan & Abbott, 2001). More recently, the distinction between simple and complex cells has been challenged by findings that cells in V1 span a more continuous range of behavior that also includes max-like integration by complex cells of their afferent inputs (Riesenhuber & Poggio, 1999; Rust, Schwartz, Movshon, & Simoncelli, 2005a; Finn & Ferster, 2007; Serre et al., 2007; Kouh & Poggio, 2008).

But what is all this modeling capacity for? In contrast to research based on (cross-)correlation analysis, we advance a different criterion for comparing models of the visual system. Part of the visual system can be interpreted as implementing a function from images to object categories. This function adapts over time as the brain learns to categorize particular objects and generalize about categories. The rules that govern this adaptation over time comprise a learning algorithm that we would ultimately like to understand better. A fundamental result in learning theory (Vapnik, 1995) is that learning algorithms have preferences (also known as inductive biases, priors over functions). This is true even of learning algorithms with the capacity to approximate any continuous function. Consequently, different learning algorithms produce different functions, even when presented with the same data. In the case of the brain's learning algorithm for vision, the priors of that algorithm play a central role in our ability to learn the structure and invariances in the images that we see. Whenever we choose a model of the visual system and a procedure for setting the internal parameters of that model, we also implicitly choose a learning algorithm and induce a particular prior over functions. Using the approach of rational analysis, we suppose that the brain's visual system is optimal at learning from limited data, and we can understand the learning algorithm for vision by studying the functional priors that support rapidly learning to categorize objects (Anderson, 1990). If that model and fitting procedure is faithful to the visual system, then the prior of that model will match the prior of the visual system, and consequently the model will learn with the same competences and weaknesses as the visual system.

In this work, we compare many mathematical models of V1-like neurons as bases for object categorization. Since we lack a complete characterization of the functional prior of the visual system's learning algorithm, we chose categorization tasks that seem trivial from the perspective of an adult human: distinguishing simple geometric shapes, the digits 0 to 9, and five kinds of toy. Learning theory gives us the terminology to be more precise about what trivial means here. It means that the learning algorithm of the human visual system has a prior preference for functions that work well for these tasks. We mix and match different parametric ingredients that have been put forward to explain simple and complex cells within the energy model framework to see which of those ingredients is important for generalizing about objects. We find that:

- Complex-like elements (involving sums of squared linear filter responses) induced better priors for object classification than simple-cell elements did.
- A polynomial saturating nonlinearity (based on division) was generally better than an exponential nonlinearity (i.e., tanh, logistic sigmoid), which is common in the machine learning community.
- Numerically optimizing the exponent used to pool the multiple filters of a complex-like cell was not useful; squaring 2 was the best choice,
- The possibility of reweighting numerator terms as suggested in Kouh and Poggio (2008) was slightly helpful in one task and slightly harmful in another.
- Multifilter models based on complex-like V1 elements required fewer hidden units, so although each complex-like element was more expensive to compute, the total cost of training the best complex-like models was similar to the cost of training the best simpler V1-like models.

This work differs from that of Shams and von der Malsburg (2002) in that we used machine learning methods to tune our V1-like elements, and we classified images rather than decoding their internal representations. The design of our experiments is similar to that of Edelman, Intrator, and Poggio (1997). Our models and tasks are different, but our finding that complex-like behavior in V1 is important for successful learning agrees with their findings.

2 High-Throughput Screening of V1-Like Models

For our study we defined a single broadly encompassing parameterization of a V1-like model cell response (see equation 2.1) and instantiated particular V1-like models by restricting the general form in various ways. Equation 2.1 is an extension of the canonical neural circuit of Kouh and Poggio (2008):

$$R = \sigma \left(\frac{\sum_k^K \left(\sum_j^J w_{jk} a_{jk}(x)^{p_k} \right)^{r_k}}{\beta + \gamma \sum_k^K \left(\sum_j^J a_{jk}(x)^{q_k} \right)^{s_k}} \right). \tag{2.1}$$

This general form encompasses the various models presented in section 1 (see Table 1) by various choices for scalars (J, w_{jk}, p_k, q_k, r_k, s_k, β), compressive nonlinearity σ, and activation function $a_{jk}(x)$ of the stimulus (input). We allow a_{jk} to be an affine function or to perform a half-rectification of an affine function. The parameterization is equivalent to the canonical neural circuit (see equation 2.1 in Kouh & Poggio, 2008) when $K = 1$, all a_{jk} are affine, σ is the identity, and γ is 1. This parameterization includes max-pooling and energy models by the choice of exponent values (p, r, q, s). Values of p and q greater than 2 paired with $r = p^{-1}$ and $s = q^{-1}$ approximate max-like behavior. We allowed for $K = 2$ and for a_{jk} to implement half-rectification in order to accommodate the model of Rust, Schwartz, Movshon, and Simoncelli (2005b).

2.1 Adapting Model Parameters. We evaluated V1-like models by forming a population of N V1-like elements (hidden units) and feeding their outputs into a logistic classifier to form a one-hidden-layer neural network. Each hidden unit R_n was fully connected to a small gray-scale image stimulus with initially random but eventually learned weights. These weights implement the a_{jk} affine transformations. The model categorized objects by activating categorization neurons, each a linear function of the full set of V1 model neurons. There was one categorization neuron for each object category. The tasks were object discrimination tasks, and they are described in section 3. To minimize Ω in equation 2.3, training pushes the real-valued confidence y_{z_t} to be the largest of all confidences y_l. At test time, the category with the largest confidence y_l was deemed to be the predicted category:

$$y_l = c_l + \sum_{n=1}^{N} A_{l,n} R_n(x_t) \qquad \text{confidence that } x_t \text{ has label } l \qquad (2.2)$$

$$\Omega_t = -y_{z_t} + \log\left(\sum_{l=1}^{L} e^{y_l}\right) \qquad \text{error (negative log likelihood) predicting label } z_t \text{ for } x_t$$

$$\Omega = \sum_{(x_t, z_t) \in \{\text{Train}\}} \Omega_t \qquad \text{error on training data} \qquad (2.3)$$

For a given V1-like model we searched for the best filters and other model parameters (e.g., filter weights in a_{jk}, categorization weights $A_{l,n}$) by stochastic gradient descent (Rumelhart, Hinton, & Williams, 1986). We denote the V1-like population response to a stimulus $x_t \in \mathbb{R}^M$ by a vector R whose N elements are called R_n, for $n \in [1, N]$. Each of L categorization neurons y_l is affine in R, with weight matrix $A_{l,n}$ and bias c_l, as in equation 2.2. Integer z_t denotes the correct category of x_t. We initialized A, c to zero and the internal parameters of model neurons R to random values, and then iteratively adjusted them to minimize the error (Ω) on small samples of training data. In trials, where w (in equation 2.1) was not fixed to 1, the numerator weights w were also optimized by gradient descent. In some trials, the scalars p_k, r_k, s_k, q_k were optimized by gradient descent too.

Filter values were initialized uniformly within a range $(-\sqrt{\frac{6}{N+M}}, \sqrt{\frac{6}{N+M}})$ about zero, as recommended in Bengio and Glorot (2010). The learning rate (the proportion of the negated error gradient by which parameters were incremented on each iteration) was sampled alongside the rest of the trial hyperparameters. The seed used to initialize the filter in each a_{jk} transformation was sampled randomly for each trial.

We compared models by dividing the data from each task into three sets: training, validation, and test (Bishop, 1995). The training data were used to calculate the fitting criterion Ω and its gradient with respect to model parameters. The validation data and test data were used to estimate the out-of-sample classification performance of the fitted system with each number of V1-like neurons (by counting what fraction of objects were categorized correctly). This validation set score was used in an early stopping heuristic to decide how much gradient descent on the training set was enough and was also used to select among models in the empirical results described in section 4. The early stopping heuristic was to wait at least 20 iterations through the training data and then stop when twice as much training had been done as had been necessary to arrive at the best model up to that point. The model scores listed in section 4 are all test set scores.

The models classes presented here are defined in terms of their functional form, without reference to the kind of filters that determine the response in equation 2.1. In analysis of V1 recordings, these filters are typically Gabor-like, with localized receptive fields and pairs of squared filters that implement quadrature pairs (Dayan & Abbott, 2001). Our experiments explore why this kind of filter arises, so we do not initialize our filters with Gabor-like patterns. Our experiments involve tuning randomly initialized filters by supervised learning of tasks that V1 neurons are able to perform in order to compare models of what V1 neurons do.

2.2 Hyperparameter Sampling Distribution. The set of trials (V1-like models and hyperparameters) encompassed by our parameterization is too large to search with a grid, so we adopted the high-throughput methodology of Pinto, Doukhan, DiCarlo, and Cox (2009) to explore the hyperparameter space. High-throughput search requires a proposal distribution from which to sample models. Essentially we draw models from a distribution rather than from locations on a grid. One advantage of random draws is that if we project onto any one axis of the hyperparameter space (e.g., the learning rate), then the random draws will cover the legal range more uniformly, whereas a grid will test just a few values. Another advantage of random draws is that if there is independence between the effect of two hyperparameters, then random sampling explores both simultaneously. This is a much more efficient search. We postulate that several hyperparameters in equation 2.1, such as the choice of σ, the half-rectification of each a_{jk}, and some of the choices about learning with this model have almost

- How many outer terms? $K \sim U(1, 1, 1, 1, 2)$
- How many inner terms? $J \sim U(1, 1, 2, 2, 3, 5)$
- Squashing: $\sim U(\sigma = \tanh \text{ and } \gamma = 0[\frac{1}{3}], \sigma = \text{identity and } \gamma = 1[\frac{2}{3}])$
- Exponents fixed (Fix) or optimized (Opt)? $\sim U(\text{Fix } [\frac{3}{4}], \text{Opt } [\frac{1}{4}])$
- Filter exponent $p_k \sim U(1, 1, 2, 2, 3)$ for each k
- Filter exponent $q_k \sim U(1, 1, 2, 2, 3)$ for each k
- Norm exponent $r_k \sim U(1, 2, p_k^{-1})$ for each k
- Norm exponent $s_k \sim U(1, 2, q_k^{-1})$ for each k
- Number of activation functions to half-rectify $\sim U(0, 1, J)$
- Numerator weights: vectors w_{*k} are all 1 $[\frac{1}{2}]$ or are nonnegative and sum to 1 $[\frac{1}{2}]$
- Number of hidden units $N \sim \frac{2^{\mathcal{U}(4,12)}}{JK}$ i.e.,16 –2048
- Learning rate: $2^{-\mathcal{U}(4,9)}$
- L_1 filter regularization 0 $[\frac{3}{4}]$ or else $2^{-\mathcal{U}(4,10)}$ $[\frac{1}{4}]$
- L_2 filter regularization 0 $[\frac{3}{4}]$ or else $2^{-\mathcal{U}(4,10)}$ $[\frac{1}{4}]$

Figure 1: Sampling distribution over V1-like models and learning algorithm hyperparameters for high-throughput search. Random variables (such as the learning rate) that were chosen uniformly in a range from a to b are denoted $\mathcal{U}(a, b)$. Random variables chosen from among a few (potentially repeating) values (a, b, c) where each value is equally likely are denoted $U(a, b, c)$. Random variables chosen from among a few values where each value is not equally likely are denoted $U(a[P(a)], b[P(b)], c[P(c)])$. The distributions for these variables are shown as being independent, but a rejection policy introduced dependence between them. For example, we rejected models that might raise a negative number to a fractional power and models that might divide by a zero denominator.

independent effects. To the extent that hyperparameters are independent, a high-throughput random search is more statistically efficient. We as experimenters do not have to specify or know exactly which hyperparameters are independent.

For our high-throughput search, we sampled hyperparameter assignments according to the distribution in Figure 1. Some combinations of

Table 1: Sampling Probabilities of Various V1 Models Under Our Hyperparameter Distribution.

Model	Frequency
Kouh and Poggio (2008)	56.4%
Sigmoid-like	13.0%
HPU	5.1%
Rust et al. (2005b) (no divisive inhibition)	4.5%
Adelson and Bergen (1985)	2.5%
Standard-sigmoid	1.1%
Rust et al. (2005b) (with divisive inhibition)	0.2%
$JK == 1$	42.0%
$p_k = 2 \quad \forall k$	35%
$K = 2$	11.6%

Notes: The top part of the table shows what percentage of our randomly sampled models correspond to selected models from the literature. HPU (higher-order processing unit) is a tanh of a polynomial of x (Rumelhart, McClelland, & PDP Research Group, 1986). The standard sigmoid model is a linear filter squashed by the tanh function. Sigmoid-like models have a possibly rectified linear filter squashed by either tanh or division ($\gamma = 1$). The canonical neural circuit (Kouh & Poggio, 2008) includes max-pooling and energy models by a choice of exponent values within the model. The bottom part of the table lists the percentage of randomly sampled models that come out with various properties. Our distribution is designed to compare many of the elements introduced in Kouh and Poggio (2008) with simpler models ($JK = 1$), but we also include hyperparameter K so that $K = 2$ permits the subtractive and divisive inhibition suggested in Rust et al. (2005b).

hyperparameters lead to unusable models. For example, when a_{jk} are not rectified, they may be negative, and a noninteger exponent will lead to a complex-valued response. We rejected such hyperparameter assignments when they were sampled. There is also an overparameterization if $J > 1$ and either $p_k = 1$ or $q_k = 1$. In this case, the model is equivalent to a model where $J = 1$, so we rejected these hyperparameter assignments as well. We also rejected assignments in which p_k and r_k (similarly, q_k and s_k) were both greater than one. Such assignments do not correspond to a V1-like model in the literature, and they are not numerically stable during learning. We also rejected assignments in which squashing was done by division ($\gamma = 1$ and $\sigma =$ identity), but the denominator could approach or equal zero because these models were also numerically unstable. Some model family frequencies under the (postrejection) sampling distribution are given in Table 1.

3 Discrimination Tasks

We measured the ability of each visual system model to learn three object categorization tasks: shapes: triangles, squares, or circles; digits: $0, 1, 2, \ldots, 9$; or five kinds of small toys. Shapes images (32×32 pixels)

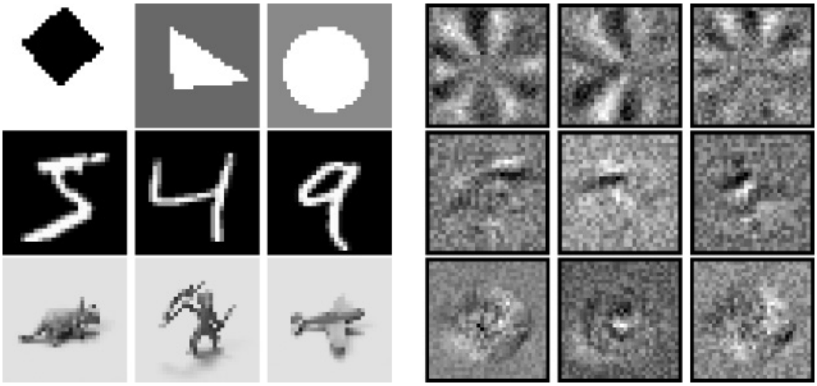

Figure 2: (Left) Images for each of the three data sets: shapes, digits, and toys. (Right) Filters learned by the best models in our study on the respective data sets. These filters come from models with multiple filters per neuron and with squared filter responses. Filters were chosen to be representative of the population in the learned model. For the shapes data set, the model has learned to phase-offset filters that implement angle-invariant selectivity for edges radiating out from near the middle of the image. For the digits, the model has learned oriented Gabor-like edge detectors. Here, the squaring of filter responses makes model neurons invariant to edge polarity. For the toys, the model has learned filters with circular swirling patterns. The mechanism of these swirling filters is not clear, but they serve to generalize well to the validation and test data.

were generated by varying the type of shape, the position, size, orientation, and gray scales of the foreground and background.[1] Digits (32 × 32 pixels) was the MNIST database of handwritten digits.[2] Toys (32 × 32 pixels) was a modified "small NORB" dataset.[3] The five sorts of toys that the visual system models had to distinguish were four-legged animals, human figures, airplanes, trucks, and cars. We modified the public data set for our experiments by shuffling all the toy instances together, and drawing 5000 training examples, 14,440 validation examples, and 29,160 testing examples randomly without replacement. The roles of these training, validation, and testing examples are explained in section 2.1. Sample stimuli for each of these tasks are illustrated in Figure 2.

[1] Data available online at http://www.iro.umontreal.ca/∼lisa/twiki/bin/view.cgi/Public/BabyAIDatasets.

[2] Data available online at http://yann.lecun.com/exdb/mnist/ (LeCun, Bottou, Bengio, & Haffner, 1998).

[3] The original data are available online at http://www.cs.nyu.edu/∼ylclab/data/norb-v1.0-small (LeCun, Huang, & Bottou, 2004).

4 High-Throughput Evaluation

We sampled 1000 models for each of our three data sets and analyzed how the performance in our tasks correlated with modeling choices.

The best models found in the random search produced scores competitive with the state of the art. The best digits model in the random search scored 1.56% error, the best score on toys was 1.78%, and the best score on shapes was 3.1% error.

The effect of each modeling choice on performance is illustrated in Figure 3. Each panel in Figure 3 shows the five best models (in validation) for each restriction in a single model hyperparameter. Trials are characterized by their classification error rates on test data. There were 5000 test examples in shapes, and the best models had around 4% error, so bear in mind while reading the following section that relative differences in performance of more than 16% are statistically significant at a 95% confidence level. There were 29,160 test examples in toys, and scores were around 2% error, so relative differences in performance of 10% are significant at the 95% level. There were 10,000 test examples in digits, and scores were around 1.5% error, so relative differences in performance of 19% are significant at the 95% level. At an 80% confidence level, differences of 8% on shapes, 5% on toys, and 10% on digits are significant.

Regarding the number of norms (K), we found that the top five models for all three data sets when $K = 1$ were better than the top five with $K = 2$. The best scores with $K = 2$ were 34% worse for shapes, 39% worse for toys, and 18% worse for digits. Part of the reason for better performance from $K = 1$ models is that there were more of them: 90% of trials were with $K = 1$. Still, the additional norm in the numerator and denominator ($K = 2$) conferred no clear advantage.

Regarding the number of terms in each norm (J), the best results on digits were with $J = 2$, toys were with, $J = 3$, and shapes were with $J = 5$. For both toys and shapes, the single-filter model ($J = 1$) was significantly poorer (by 27% on shapes, by 40% on toys) than the best model. There were approximately equal numbers of trials with $J = 1$ as with $J > 1$.

With regard to filter exponents p_k and q_k, we found that squared filters ($p_k = q_k = 2 \forall k$) gave the best performance. Linear filters ($p_k = q_k = 1 \forall k$) were at best 60% worse than squared filters on shapes and toys, and cubed filters ($p_k = q_k = 3 \forall k$) were at best 95% worse on shapes and toys. On the digits task, linear filters for p_k were 8% worse and cubic ones were 20% worse. On digits, there was no difference among values for q_k. Trials were split such that roughly 40% had $p_k = 1$, 40% had $p_k = 2$, and 20% had $p_k = 3$. The distribution of q_k trials was similar and independent from p_k.

Regarding norm exponents r_k and s_k, we found that $\frac{1}{2}$ (a square root) was the best on shapes and toys. Values of $\frac{1}{3}$ and 2 gave performances that were at best 60% worse. A value of 1 was 30% worse on shapes but just 8%

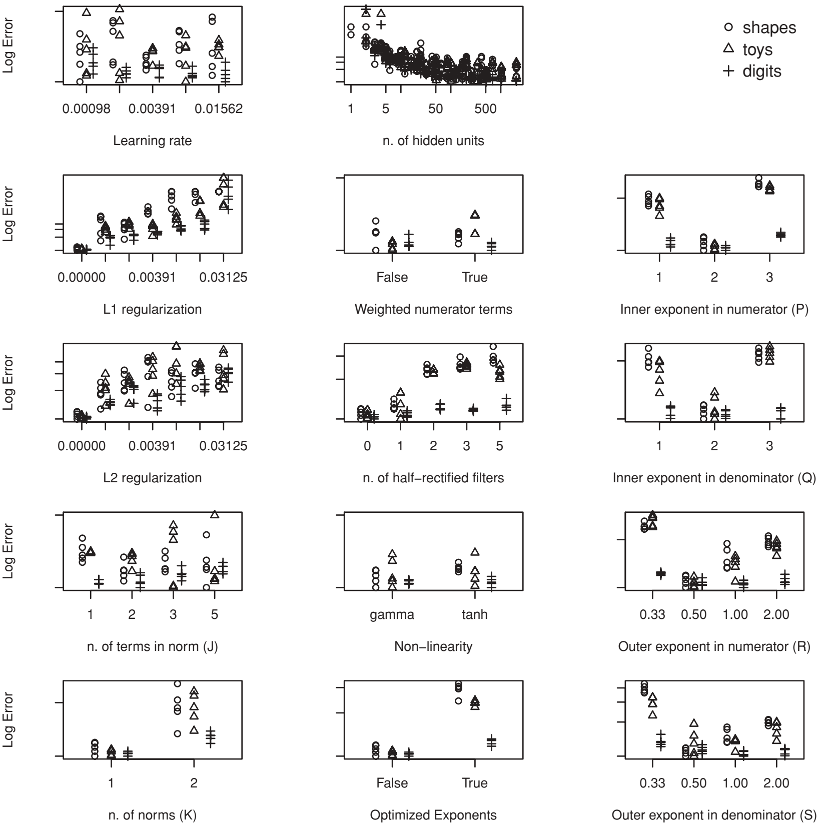

Figure 3: Each panel shows test set scores of the five best (in validation) models in each category on each data set relative to the best model on each data set, from a broad random search over 1000 general V1 models (see equation 2.1). The y-values are log-scaled, and each data set's scores are shifted vertically so that they are aligned at the bottom of each panel. The y-axis ticks correspond to a doubling of test set error. Panels for p, q, r, and s show performance as a function of the maximum (initial) value of p_k, q_k, r_k, and s_k, respectively. The standard errors in the best test set error rates plotted on the vertical axes are at most 10%, which is at most approximately the size of the icons.

worse on toys. On digits, $s_k = 1$ and $s_k = 2$ were best by 10% over $\frac{1}{2}$, and all values except $\frac{1}{3}$ were equally good for r_k.

To test the potential advantage of fractional exponent values close to $p_k = q_k = 2$ and $r_k = s_k = .5$, we looked at trials where the exponent was adjusted according to the gradient during learning. The panel labeled "Optimized Exponents" illustrates that performance with these learned exponents was always worse (140% worse on shapes, 120% worse on toys, and 20% worse on digits), and comparable to the performance when p or r was fixed to nonoptimal values.

Half-rectification of filters (a_{jk}) was harmful. Approximately half the trials had one half-rectified filter, and the other half were divided about evenly between zero, two, three, and five. The best trials had no half-rectified filters, and trials with two or more half-rectified filters were 120% worse on shapes, 120% worse on toys, and 14% worse on digits.

In approximately 30% of trials (50% of the trials where $J > 1$), we optimized the weighting of numerator terms (vectors $w_{\cdot k}$) by gradient descent, under the constraint that they be nonnegative and sum to 1. Optimization of these vectors helped to reduce error in the shapes data set (by a factor of approximately 12%), but it raised the error rate in toys (by about 17%) and did not make any difference in digits.

Approximately two-thirds of trials used squashing by division ($\gamma = 1$) rather than tanh. The best-performing of tanh trial on shapes was 20% worse than the best division model, but on the other two data sets, the difference in performance was not significant. There was less variability among results when squashing by division.

The choice of learning rate was not critical; good results were obtained for each task with every value in the range under investigation. L1 and L2 regularization of weights was simply harmful.

4.1 Single-Filter Models Versus Multifilter Models. One of the basic questions our experiment addresses is whether modeling capacity is better spent on additional model neurons or on more complicated multifilter model neurons. One way to quantify the capacity of a model is by the number of degrees of freedom that may be adapted. In our case, this quantity is dominated by the product of three terms: the number of hidden units (N; see equation 2.1), the number of filters in each norm (J), and the number of norms (K). Figure 4 illustrates the relationship between capacity and performance in these models for single-filter models ($JK = 1$) and multifilter alternatives ($JK > 1$). Two things stand out in Figure 4: a certain amount of capacity is necessary for good performance (there are no points in the bottom left quadrant), and it is important that the capacity be in the form of model neurons with multiple linear filters (solid icons dominate the bottom-right quadrant). So ultimately, additional single-filter model neurons are a poor substitute for multifilter alternatives when filters must be learned from data.

Figure 4 was computed as follows. For each data set, we drew a random sample of 1000 model trials. Each random sample of 1000 model trials was partitioned into condition sets according to the number of filters in the model (product JKN), rounded to the nearest power of 2. The figure shows the results of each data set slightly offset horizontally for clarity. Each condition is a pair (data set, capacity), and each condition set is the set

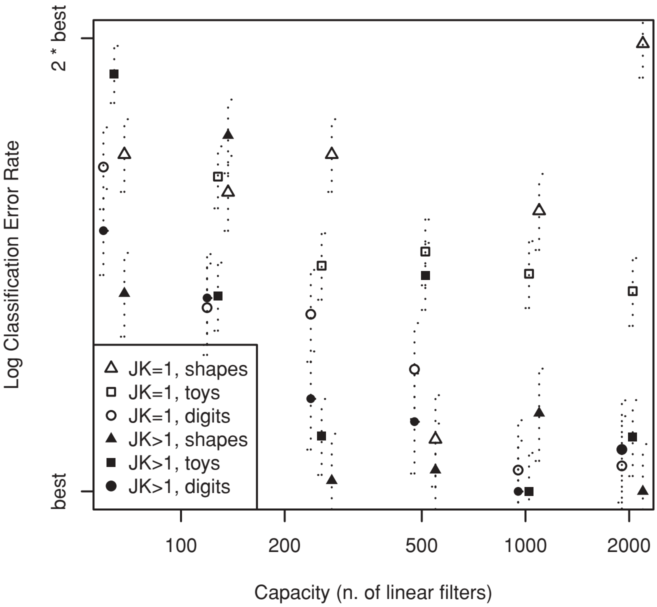

Figure 4: Why not just use more simple model neurons with one filter each? There are different ways to add capacity to a model. Each point below corresponds to test error for the best model in one experimental condition (data set and capacity, defined as $N \times J \times K$; see equation 2.1). The horizontal axis is capacity (with each data set's results slightly offset for clarity), and the vertical axis is out-of-sample classification error relative to the best model on the same data set. Each best model is chosen over N (with $37 < N < 50$) other randomly sampled models within a given condition (different learning rate, different exponents; see Figure 1 for all variations). Solid markers denote the performance of models with $JK > 1$, and empty markers denote the performance of models with $JK = 1$. Dotted lines indicate standard error on the test set performance of the best model in each condition. The greater density of solid symbols in bottom-right portion of the figure means two things: a certain amount of capacity is necessary for good performance, and it is important that the capacity be in the form of model neurons with multiple filters.

of trials matching the condition, but with different nonlinearities, learning rates, exponents, and so on (see the variations in Figure 1). We define a test set score for each condition by taking the test set score from the best model (according to validation performance) in its condition set. The smallest condition set had 39 elements, and condition sets with more than 50 elements were randomly truncated to 50. Each test score was divided by the best test score in the original 1000 trials so that data set performances could be compared. Conditions whose test set error was high (more than twice the best test score on the same data set) are also not shown for clarity.

For all the models in our study, the computational cost of determining the label for a test example is proportional to the number of filters, which is the horizontal axis (capacity) of Figure 4. The best-performing multifilter models required approximately the same amount of computation time to train and test as the best single-filter models that were a few times larger, but the multifilter models performed better.

5 Discussion

Visual system models in our study were more successful at categorizing familiar objects in novel stimuli when their V1-like neurons were able to go beyond the basic linear-nonlinear model and exhibit the range of behavior found in V1 simple and complex cells. When we used a gradient-based method to optimize neuron parameters, the models similar to the classic energy model (where the firing rate is determined by the sum of squared linear filter responses) demonstrated a superior capacity to generalize from labeled examples of objects. More complex variants on the energy model such as the models of Rust et al. (2005a) and Kouh and Poggio (2008) were also better than the basic simple cell model but brought no consistent advantage over the energy model. The models with squared linear filters were much better at generalizing to new stimuli than the simpler linear-nonlinear models often used in theoretical work.

The most important characteristic of the V1-like neuron model used for image classification was the complex cell-like behavior, obtained through multiple (from two to at least five) squared linear filters that captured second-order interactions between regions of the receptive field. In terms of learning theory (Vapnik, 1995), our results suggest that complex-like models yielded families of functions that were more appropriate for learning to classify objects than linear-nonlinear models. Large numbers of simple cell models were no substitute for complex cell models because the simple cell models brought a poorer prior over functions for object categorization. The nonlinearities required by the complex-cell models could come from multiple biological sources: feedback from extrastriate (Bredfeldt & Ringach, 2002) and lateral connectivity (Heeger, 1992) could play a role, and dendritic trees have a capacity for nonlinear processing (Haüsser & Mel, 2003; Rhodes, 2008).

One hypothesis for why the complex-like parameterization learned more quickly is that the sum of squared filters can be more robust to small translations (Adelson & Bergen, 1985). As evidence for this hypothesis, the filters learned by the most successful models in our study are illustrated in Figure 2. The results here are mixed. The model that was best at discriminating shapes supports the hypothesis; it involves a sum of squared linear filter responses, and the filters in the model look like phase-offset gratings that implement angle-invariant selectivity for edges radiating out from various locations near the middle of the image. The model best at discriminating digits supports the hypothesis less strongly; it involves many small Gabor-like-oriented edge detectors. These detectors do not have gratings, so they would not be robust to displacement of the edge. Squaring of filter responses in these models would make the edge detection robust to changes in polarity. Edges in the digits data set are almost always close together (the two sides of a pen stroke), so perhaps polarity invariance is a form of translation invariance in this particular data set. The model best at discriminating toys offers weak support for the hypothesis; the filters in this model are neither gratings nor edge detectors, and it is not clear (to us) how they work. So for at least two of the three data sets in our study (shapes and digits), the filters learned to implement the computational property (quadrature pairs for translational invariance) that motivated the energy model. It is not obvious a priori that for an image x, supervised learning of a transfer function of the form $\sqrt{(ax)^2 + (bx)^2}$ should learn phase-offset Gabors and sinusoids for filters a and b. But in at least two of our three data sets, that is indeed what happens for the majority of V1-like cells. This finding supports the hypothesis that quadrature pairing for pooling in complex cells is an important computational aspect of low-level feedforward vision models; when the machinery for that is present, a simple learning algorithm learns to use it that way.

How do our results compare with other published results on the digits data set (MNIST)? A database of results is online (LeCun, 2008): chance is 90% error, a linear classifier can get 12% error (LeCun et al., 1998), K-nearest neighbors 3.09% using an L_2 metric (LeCun, 2008), and a gaussian-kernel SVM 1.4% error (LeCun, 2008). With an augmented data set, an SVM can achieve 0.56% error (DeCoste & Schölkopf, 2002), and the best deep convolutional neural network achieved 0.39% error (Ranzato, Poultney, Chopra, & LeCun, 2007). The best digit classifier in our study (based on classic energy model neurons) scored 1.54% error. It lags behind more sophisticated machine learning models, but in this sort of comparison, it should be interpreted as a building block for more powerful computer vision models rather than a complete model in its own right. Our score compares favorably with the standard sigmoidal neural network approach (1.8% error; LeCun, 2008), indicating that complex cells can extract more discriminating features than simple cells. Future work will examine the utility of these V1-like models within a hierarchical convolutional architecture that goes further toward replicating the structure of the visual system (LeCun et al., 1998; Riesenhuber & Poggio, 1999; Kavukcuoglu, Ranzato, Fergus, & LeCun, 2009; Pinto et al., 2009).

We do not know how well a primate visual system would perform in these tasks because for the purpose of comparison, it would be necessary to train visual systems exclusively on these very limited sets of stimuli. Instead, we draw on the approach of rational analysis and appeal to the trivial straightforwardness of these categorization tasks to support our claim that the learning algorithm of the visual system exhibits a preference for functions that are effective in these tasks. The faster learning in the Rust et al. (2005a) model agrees with the hypothesis that that model's functional priors are closer to the visual system's priors and that the priors of the other models are further from the visual system's.

Acknowledgments

We are immensely grateful for the feedback from Curtis Baker, Paul Cisek, Aaron Courville, Andrea Green, Christopher Pack, Nicole Rust, and Terry Sejnowski, as well as the financial support of NSERC, the Canada Research chairs, and MITACS.

References

Adelson, E. H., & Bergen, J. R. (1985). Spatiotemporal energy models for the perception of motion. *Journal of the Optical Society of America, 2*(2), 284–299.

Anderson, J. (1990). *The adaptive character of thought*. Mahwah, NJ: Erlbaum.

Bengio, Y., & Glorot, X. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of AISTATS 2010* (Vol. 9, pp. 249–256). Brookline, MA: Microtome.

Bishop, C. (1995). *Neural networks for pattern recognition*. New York: Oxford University Press.

Bredfeldt, C. E., & Ringach, D. L. (2002). Dynamics of spatial frequency tuning in macaque V1. *Journal of Neuroscience, 22*(5), 1976–1984.

Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience*. Cambridge, MA: MIT Press.

DeCoste, D., & Schölkopf, B. (2002). Training invariant support vector machines. *Machine Learning, 46*(1–3), 161–190.

Doi, E., & Lewicki, M. S. (2007). A theory of retinal population coding. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems, 19*. Cambridge, MA: MIT Press.

Edelman, S., Intrator, N., & Poggio, T. (1997). *Complex cells and object recognition.* Unpublished manuscript.

Finn, I. M., & Ferster, D. (2007). Computational diversity in complex cells of cat primary visual cortex. *Journal of Neuroscience, 27*(36), 9638–9648.

Haüsser, M., & Mel, B. (2003). Dendrites: Bug or feature? *Current Opinion in Neurobiology, 13*, 372–383.

Heeger, D. J. (1992). Normalization of cell responses in cat striate cortex. *Visual Neuroscience, 9*(2), 181–198.

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology, 160*(1), 106–154.

Kavukcuoglu, K., Ranzato, M., Fergus, R., & LeCun, Y. (2009). Learning invariant features through topographic filter maps. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR'09)*. Los Alamitos, CA: IEEE Computer Society.

Kouh, M. M., & Poggio, T. T. (2008). A canonical neural circuit for cortical nonlinear operations. *Neural Computation, 20*(6), 1427–1451.

LeCun, Y. (2008). *The MNIST database of handwritten digits*. Available online at http://yann.lecun.com/exdb/mnist/.

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

LeCun, Y., Huang, F.-J., & Bottou, L. (2004). Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR'04)* (Vol. 2, pp. 97–104). Los Alamitos, CA: IEEE Computer Society.

Naka, K. I., & Rushton, W. A. (1966). S-potentials from luminosity units in the retina of fish (cyprinidae). *Journal of Physiology, 185*(3), 587–599.

Nykamp, D. Q., & Ringach, D. L. (2002). Full identification of a linear-nonlinear system via cross-correlation analysis. *Journal of Vision, 2*, 1–11.

Olshausen, B., & Field, D. J. (2005). How close are we to understanding V1? *Neural Computation, 17*, 1665–1699.

Pinto, N., Doukhan, D., DiCarlo, J. J., & Cox, D. D. (2009). A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS Comput Biol, 5*(11), e1000579.

Ranzato, M., Poultney, C., Chopra, S., & LeCun, Y. (2007). Efficient learning of sparse representations with an energy-based model. In B. Schölkopf, J. Platt, & T. Hoffman (Eds.), *Advances in neural information processing systems, 19*. Cambridge, MA: MIT Press.

Rhodes, P. A. (2008). Recoding patterns of sensory input: Higher-order features and the function of nonlinear dendritic trees. *Neural Computation, 20*(8), 2000–2036.

Riesenhuber, M., & Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nature Neuroscience, 2*, 1019–1025.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*, 533–536.

Rumelhart, D. E., McClelland, J. L., & the PDP Research Group (1986). *Parallel distributed processing: Explorations in the microstructure of cognition* (Vol. 1). Cambridge, MA: MIT Press.

Rust, N., Schwartz, O., Movshon, J. A., & Simoncelli, E. (2005a). Spatiotemporal elements of macaque V1 receptive fields. *Neuron, 46*(6), 945–956.

Rust, N., Schwartz, O., Movshon, J. A., & Simoncelli, E. (2005b). Spatiotemporal elements of macaque V1 receptive fields. *Neuron, 46*(6), 945–956.

Serre, T., Kreiman, G., Kouh, M., Cadieu, C., Knoblich, U., & Poggio, T. (2007). A quantitative theory of immediate visual recognition. *Progress in Brain Research, Computational Neuroscience: Theoretical Insights into Brain Function, 165*, 33–56.

Shams, L., & von der Malsburg, C. (2002). The role of complex cells in object recognition. *Vision Research, 42*(22), 2547–2554.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.

Wilson, H. R., & Cowan, J. D. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysics Journal, 12*(1), 1–24.

Received February 3, 2009; accepted August 4, 2010.