Available online at www.sciencedirect.com
SCIENCE DIRECT®
Neural Networks xx (2004) xxx–xxx
Neural Networks
www.elsevier.com/locate/neunet

2004 Special Issue

Nonlinear V1 responses to natural scenes revealed by neural network analysis

Ryan Prenger[a], Michael C.-K. Wu[b], Stephen V. David[c], Jack L. Gallant[d,e,*]

[a]Department of Physics, University of California Berkeley, Berkeley CA, USA
[b]Biophysics Graduate Group, University of California Berkeley, Berkeley CA, USA
[c]Department of Bioengineering, University of California Berkeley, Berkeley CA, USA
[d]Helen Wills Neuroscience Institute, University of California Berkeley, Berkeley CA, USA
[e]Department of Psychology, University of California Berkeley, 3210 Tolman Hall #1650, Berkeley, CA 94720-1650, USA

Received 30 March 2004; accepted 30 March 2004

Abstract

A key goal in the study of visual processing is to obtain a comprehensive description of the relationship between visual stimuli and neuronal responses. One way to guide the search for models is to use a general nonparametric regression algorithm, such as a neural network. We have developed a multilayer feed-forward network algorithm that can be used to characterize nonlinear stimulus-response mapping functions of neurons in primary visual cortex (area V1) using natural image stimuli. The network is capable of extracting several known V1 response properties such as: orientation and spatial frequency tuning, the spatial phase invariance of complex cells, and direction selectivity. We present details of a method for training networks and visualizing their properties. We also compare how well conventional explicit models and those developed using neural networks can predict novel responses to natural scenes.

Keywords: Striate cortex; Multi-layer perceptron; Receptive field; Nonparametric model; Natural vision; Prediction; Reverse correlation

1. Introduction

One of the central goals of visual neuroscience is to obtain comprehensive descriptions of the relationship between visual stimuli and neuronal responses. This task has proven quite daunting. Responses from any neuron can only be recorded for a limited amount of time, with a limited range of stimuli. In addition, the behavior of many neurons in the visual cortex is complex and inherently nonlinear. This causes problems for both the selection of effective stimuli and the choice of appropriate models.

The nonlinear behavior of visual cortical neurons will cause their response characteristics to vary when probed with different stimuli (David, Vinje, & Gallant, 1999). In theory, a nonlinear neuron can be characterized using the white noise approach (Marmarelis & Marmarelis, 1978). In practice, white noise is inefficient, especially when used to characterize neurons at higher stages of visual processing. Neurons in visual cortex do not respond well to white noise stimuli, making them difficult or impossible to characterize. One alternative is to use natural images as stimuli.

Because a nonlinear function such as the ones governing neuronal responses can be fit by an infinite number of nonlinear models, it is difficult to select an explicit model a priori. One alternative approach to explicit model testing is to use a machine learning-based nonparametric regression algorithm, such as a neural network, to estimate the nonlinear relationship between stimulus and response. The nonparametric algorithm requires no assumptions about the function it models, but the resulting model can be analyzed to reveal the nonlinear stimulus-response transfer function that has been recovered. Two previous studies have used multilayer neural networks to recover the nonlinear response properties of neurons in the visual cortex (Lau, Stanley, & Dan, 2002; Lehky, Sejnowski, & Desimone, 1992). However, these algorithms were not optimized for naturalistic stimuli.

* Corresponding author. Address: Department of Psychology, University of California Berkeley, 3210 Tolman Hall #1650, Berkeley, CA 94720-1650, USA. Tel.: +1-510-642-2660; fax: +1-210-642-5293.
E-mail address: gallant@socrates.berkeley.edu (J.L. Gallant).

0893-6080/$ - see front matter © 2004 Published by Elsevier Ltd.
doi:10.1016/j.neunet.2004.03.008

Nomenclature

$\hat{r}(t)$,	Network predicted response at time t
$x_j(t)$,	Input vector at time t
b_o,	Network parameter, output bias
b_i,	Network parameter, input bias
w_i,	Network parameter, output weights
u_{ij},	Network parameter, input weights
μ,	Mean of the distribution
σ^2,	Variance of the distribution
E,	Value of the objective function
β,	Empirical error hyperparameter
N,	Number of data points
d,	Dimensionality of the input vector
j,	Network parameter regularization grouping index
α_j,	Hyperparameter for group j
m_j,	Number of parameters in group j
G_k^j,	The kth parameter in the jth group

We have developed a method for the training and interpretation of a multilayer feed-forward network that can be applied to data sets consisting of complex natural image stimuli and associated neuronal responses. To test the validity of this approach, we have applied it to data recorded from primary visual cortex (area V1), where some nonlinear response properties have already been described. Our study demonstrates that a neural network can recover not only the known linear response properties of V1 neurons, such as spatial frequency and orientation tuning, but also nonlinear response properties such as direction selectivity and the spatial phase invariance of complex cells.

Explicit models often require fewer free parameters than neural networks, and allow individual model components to be tested separately. Neural network analyses can guide the search for novel response properties that can then be incorporated into an explicit model. Our neural networks 'discovered' several known response properties of V1 neurons, deriving them entirely from the natural image stimuli and response data. We implemented the response properties found by the networks in an explicit model and demonstrated an increase in its predictive power. This approach may prove useful in higher visual areas, where there is little knowledge to guide the a priori choice of an explicit model.

2. Methods

2.1. Stimuli and data collection

We recorded responses from 34 well isolated neurons in parafoveal area V1 of an awake, behaving male macaque (*Macaca mulatta*). The animal performed a fixation task for a liquid reward. Eye position was monitored with a scleral search coil. Trials were aborted if eye position deviated more than 0.35° from the fixation point. Action potentials were identified using a custom hardware window discriminator with a temporal resolution of 0.1 ms. All procedures conformed to NIH guidelines and were approved by the University of California, Berkeley Animal Care and Use Committee (Procedural details can be found in (Vinje & Gallant, 2002))

The spatial receptive field size and location of each isolated neuron were estimated manually during fixation, using bar and grating stimuli. Estimates were confirmed by reverse correlation of responses to a dynamic sequence of randomly positioned black and white squares on a gray background (i.e. a sparse noise stimulus) (Connor, Gallant, Preddie, & Van Essen, 1996; Deangelis, Ohzawa, & Freeman, 1993). Six to eight squares spanned the manual estimate of the receptive field (0.1–0.5°/square). The region over which sparse noise stimulation elicited spiking responses was designated the classical receptive field (CRF). The manual and automatic estimation procedures were generally in good agreement.

Stimuli consisted of a dynamic sequence of natural images in which a new image appeared on each refresh cycle of the 72 Hz display (see Fig. 1A). Image sequences were designed to have temporally white statistics and the spatial statistics of natural scenes. Each frame contained a circular image patch extracted at random from 1280×1024 pixel images obtained from a high-resolution, commercial digital photo library (Corel, Inc.). Images included animals, humans, landscapes, and man-made objects. Color images were converted to gray scale before presentation. All image patches were two to four times larger than the CRF, and their outer edges (10% of the radius) were linearly blended into the gray background of the display. Each 100–350 s image sequence was divided into 20–70 five second segments. One segment, centered on the CRF, was presented during each fixation trial. Each sequence was shown only once. All stimuli were presented on a standard CRT using a neutral gray background. In order to avoid any transient trial onset effects, the first 196 ms of data acquired on each trial were discarded.

2.1.1. Stimulus preprocessing

Stimuli were preprocessed to decrease computational demands and make network estimation more efficient. First, information from the quantitative spatial receptive field mapping procedure was used to identify the stimulus region that circumscribed the CRF of each neuron. Image sequences were then cropped around the CRF and down-sampled to 20×20 pixels.

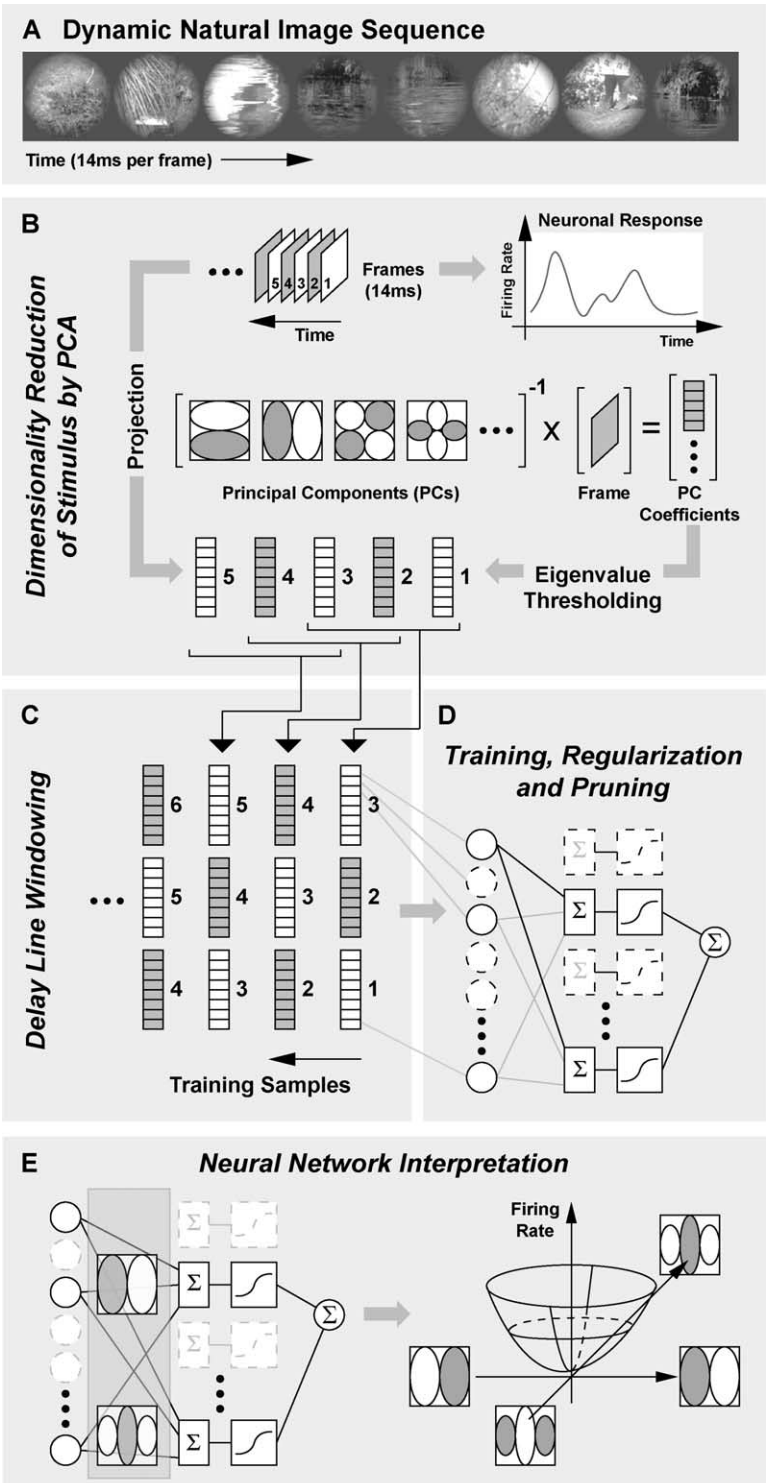

Fig. 1. Network model training and visualization. (A) One brief segment of a typical dynamic natural image sequence. Each frame contained one grayscale natural image drawn at random from a large database. Images were 1–4 times the size of the classical receptive field. (B) To facilitate network performance, the dimensionality of the stimulus was reduced by projecting each frame onto the first 25 principal components of the image sequence. The top row represents a portion of a stimulus sequence and its associated response. The middle row represents the projection of a single stimulus frame onto the principal components of the natural image sequence, resulting in a column of coefficients for each frame in the stimulus sequence. (C) The preprocessed stimulus coefficients were used as inputs into a multilayer, feed-forward neural network, which predicted the corresponding neuronal responses. At each point in time the network had access to seven frames of the stimulus sequence. (D) During network training, regularization procedures were used to remove parameters and simplify the network. This procedure prevented overfitting and aided in network interpretation. (E) After training was complete a separate procedure was used to visualize the network. First, the nonlinear mapping function implemented by the network was re-expressed in terms of its principal dimensions. In this schematic we show a network that produces two dimensions. Then the observed and predicted responses were projected onto these principal network dimensions.

Second, for each neuron, the spatial principal components of the image patches were computed using singular value decomposition. Patches were then projected onto the first 25 principal components (see Fig. 1B). Note that image luminance was not normalized before projection, so the principal components include mean luminance. The first 25 principal components accounted for 94% of the image variance on average. Increasing the number of principal components beyond 25 had little effect on the final network solution, but it dramatically increased network convergence time.

Finally, projections into the principal component domain were scaled to have a mean of zero and variance of one. This scaling operation removed the low frequency bias present in natural images (Field, 1987). The gradient descent algorithm used here minimizes squared error regardless of stimulus correlations, so in principle the low frequency bias of natural images could simply be ignored. However, explicit bias removal reduces network training time.

2.1.2. Network architecture

We used a three-layer neural network (input layer, hidden layer, and output layer) with a tapped delay line architecture (see Fig. 1C and (Waibel, Hanazawa, Hinton, Shikano, & Lang, 1989)) to learn the optimal mapping of a dynamic image sequence into the neuronal responses recorded from each neuron. The principal component projection of a dynamic image stimulus is treated as a vector, and used as input to the neural network. Each hidden unit of the network sums the elements of this vector, weighted by network parameters referred to as input weights. A constant term called the input bias is added to the output of each hidden unit, and the resulting vector serves as the argument to a sigmoidal function (the hyperbolic tangent). The outputs from the sigmoidal functions are summed, weighted by network parameters referred to as output weights. The output of the network is this weighted sum plus another constant term, called the output bias. The output of the network represents the predicted instantaneous firing rate of the neuron to the input vector.

Given an input vector, x, the response, $\hat{r}$ of the network can be written as:

$$\hat{r}(t) = b_o + \sum_{i=1}^{h} w_i \tanh\left(b_i + \sum_{j=1}^{d} u_{ij} x_j(t)\right).$$

where w_i are the output weights, u_{ij} are the input weights, b_i and b_o are the input and output bias, h is the total number of hidden units in the network, and d is the length of the input vector. Given enough hidden units this series can theoretically fit any continuous function (Barron, 1993; Hornik, Stinchcombe, & White, 1989). However, our current understanding of neural coding and a previous study (Lau et al., 2002) suggest that many neural response properties can be characterized using relatively few hidden units.

In our analysis the input vector had length 175, consisting of the 25 principal component coefficients at seven time lags. The first time lag represented responses that were simultaneous with stimulus presentation and the seventh time lag represented a latency of seven frames between stimulus onset and response (14–16 ms per frame). The network was initialized with 12 hidden units. This number was chosen to allow it to obtain a very accurate (i.e. overfit) initial fit. The input weights and input bias parameters were initialized by sampling randomly from a normal distribution,

$$\left(\mu = 0, \sigma^2 = \frac{1}{\sqrt{175+1}}\right).$$

The output weights and output bias were similarly initialized from a normal distribution,

$$\left(\mu = 0, \sigma^2 = \frac{1}{\sqrt{12+1}}\right).$$

2.1.3. Gradient descent

The parameters of the network were optimized using a back propagation algorithm (Rumelhart, Hinton, & Williams, 1986) that implements the scaled conjugate gradient (SCG) method (Moller, 1993). SCG prevents oscillations in parameter space by computing the conjugate direction rather than the steepest descent direction, but it may have slow convergence due to small step size. To maximize convergence speed SCG approximates the Hessian matrix, describing the curvature of the local search space, to compute the optimal step size along the conjugate direction. This algorithm minimizes the squared error between the network predictions and observed neuronal responses, and it is an order of magnitude faster than conventional back propagation with steepest descent.

2.1.4. Regularization

A common problem with nonlinear regression procedures is that they tend to overfit the training data. Overfitting is undesirable because it reduces the ability of the network to predict responses to novel data sets. In addition, overfit models often contain more parameters than required for optimal prediction, making network interpretation more difficult.

Regularization procedures are designed to prevent overfitting by constraining network parameters. In a Bayesian framework, network parameters can be constrained by assuming they are drawn from a prior probability distribution (Mackay, 1995). We grouped network parameters into related sets and imposed an appropriate prior distribution on each parameter group. These prior assumptions are expressed as weight decay terms, controlled by regularization hyperparameters that were assigned independently to each group. The hyperparameters are chosen such that the weight decay terms continually decreased the influence of unimportant parameters during training.

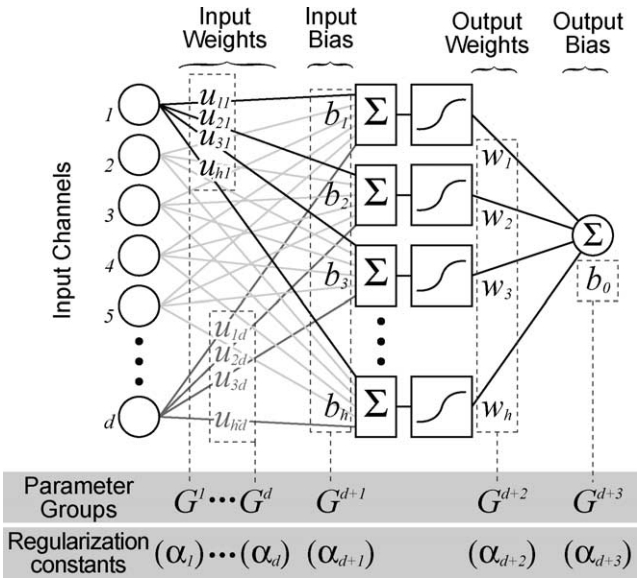

Fig. 2. Parameter grouping for automatic relevancy determination. All network parameters are shown, along with their associated parameter groups (groups labeled $G^1 - G^{d+3}$). Each weight corresponding to a given input was assigned its own group with a separate regularization parameter (groups labeled $G^1 - G^d$). Three additional groups were defined for the input bias terms (G^{d+1}), output weights (G^{d+2}), and output bias (G^{d+3}).

2.1.5. Parameter grouping

Because visual cortical neurons may be tuned to a narrow range of stimuli, many of the principal components of the stimulus are likely to be unrelated to the response of a specific neuron. To allow any individual principal component to be ignored, all of the input weights corresponding to given component of the input vector were grouped and assigned a single hyperparameter (see Fig. 2). We used d groups of parameters to describe all $d \times h$ input weights of the network, denoted G^j for $j = 1, \ldots, d$. Three additional groups were defined, for the input bias terms (G^{d+1}), the output weights (G^{d+2}), and the output bias term (G^{d+3}). Each of the $d + 3$ parameter groups, $j = 1, \ldots, d + 3$, was controlled by a separate regularization hyperparameter α_j. We assumed that the parameters in the jth group are normally distributed,

$$\left(\mu = 0, \sigma^2 = \frac{1}{\alpha_j}\right).$$

Each α_j was initialized and updated by the training algorithm, as described in Section 2.2.

In addition to regularizing network parameters, we also regularized the empirical error (the sum of squares of the differences between predicted and actual responses). This was controlled by another regularization hyperparameter, β. We assume that the error between a given response and the actual response is normally distributed,

$$\left(\mu = 0, \sigma^2 = \frac{1}{\beta}\right).$$

β was initialized and updated by the training algorithm, as described in Section 2.2.

Given these prior assumptions, the most probable network configuration is the one that minimizes the objective function (Mackay, 1995):

$$E = \frac{\beta}{2}\sum_{t=1}^{N}(\hat{r}_t - r_t)^2 + \sum_{j=1}^{d+3}\frac{\alpha_j}{2}\sum_{k=1}^{m_j}(G_k^j)^2.$$

The first term is the empirical error, and the second is the weight decay function. Here E is the current value of the objective function. In the first term β is a regularizing hyperparameter, N is the number data points (input–output pairs), $\hat{r}_t$ is the predicted response at time t to a specific stimulus, and r_t is the actual response at time t. In the second term $d + 3$ is the number of parameter groups, m_j is the number of parameters in group G^j, G_k^j is the kth parameter in the jth group, and α_j is the hyperparameter for the jth group. As the constant β grows the algorithm gives more emphasis to minimizing the empirical error and as each α_j grows, the algorithm emphasizes driving the parameters in the jth group to zero.

2.1.6. Initial values of regularization hyperparameters

The optimal values of α_j and β are not known a priori; they must be estimated during training. We therefore initialized the hyperparameter on each group of input weights α_j to be proportional to the inverse of the total stimulus power in the corresponding input vector component, prior to scaling:

$$a_j^{\text{initial}} = \frac{A}{\sum_{t=1}^{N} I_{tj}^2},$$

$$A = (0.0001 \sim 1) \times \max\left(\sum_{t=1}^{N} I_{tj}^2\right) \qquad \text{for } j = 1, \ldots, d$$

Here $\alpha_j^{\text{initial}}$ is the initial value of α_j in group j, I_{tj} is the amplitude of input vector component j and time point t prior to scaling, and N is the number of time points. The initial value of α_j, for the remaining groups was set to 0.00001, so that the initial stages of weight decay have little influence on these parameters. In practice, we find that the choice of initial values for the α_j terms has little effect on the final network solution, provided the values are low enough to allow the network to overfit initially. The parameter β was initially set to 1.

2.2. Training the network

The parameters of the network were optimized using a back propagation algorithm that minimizes the objective function (See 'Parameter grouping' above). It is usually desirable to overfit initially, because this ensures that the regularization algorithm does not converge on an underfit

solution. However, drastic overfitting increases computational time dramatically. To achieve a reasonable degree of initial overfitting, the network was trained for 500 iterations with the initial values of the hyperparameters held constant; if the mean squared error fell below 0.2 during this initial training we assumed that overfitting had been achieved and training was stopped immediately.

After the initial training, the hyperparameters were updated using the current values of the network parameters and empirical error, and training continued using the revised objective function. Hyperparameters were then updated after every 30 training steps. The process was stopped when both the squared error and the regularization error changed by less than .01 for 5 updating steps.

The α_j hyperparameters were updated as follows:

$$\alpha_j = \frac{1}{\mathrm{var}(\{-G_k^j, G_k^j\}_{k=1}^{m_j})} = \frac{m_j - 0.5}{\sum_{k=1}^{m_j} (G_k^j)^2}.$$

G_k^j are the parameters in group G^j and m_j is the number of parameters in the group. As stated earlier, the weight decay term in the objective function corresponds to a prior assumption that the parameters in a group are sampled from a normal distribution,

$$\left(\mu = 0, \sigma^2 = \frac{1}{\alpha_j}\right).$$

This method of updating α_j corresponds to using the variance of the group to determine the value of α_j. The notation $\mathrm{var}(\{-G_k^j, G_k^j\}_{k=1}^{m_j})$ means that the variance of both the parameters and the negatives of the parameters are calculated together. Including both the positive and negative values of the parameters forces the collection of parameters to have a mean of zero. Updating α_j this way is mathematically equivalent to setting α_j equal to the reciprocal of the average power in the parameters. The 0.5 in the numerator of the updating function is a correction for using finite data to estimate variance. The net result of this updating procedure is that the groups with smaller parameters will be driven to zero more quickly.

The hyperparameter β was updated similarly:

$$\beta = \frac{1}{\mathrm{var}(\{-\delta_t, \delta_t\}_{t=1}^{N})} = \frac{N - 0.5}{\sum_{t=1}^{N} \delta_t^2} \text{ with } \delta_t = (\hat{r}_t - r_t).$$

Here $\hat{r}_t$ corresponds to the predicted response of the network to a stimulus, r_t is the observed neuronal response to that stimulus and N is the training sample size. As stated earlier, the empirical error term in our objective function corresponds to a prior assumption that the residuals are sampled from a normal distribution,

$$\left(\mu = 0, \sigma^2 = \frac{1}{\beta}\right).$$

This method of updating β corresponds to using the variance of the error to determine the value of β.

The critical aspect of our regularization procedure is that weight decay is based on the average power of the weights in a group, but independently of all other groups. Whenever α for a given group of weights was greater than 10^{10}, the weights were pruned from the network permanently. This made the network smaller, increasing training speed and making the network more interpretable. This entire method of regularization corresponds to a form of Bayesian automatic relevancy determination (Mackay, 1995).

2.2.1. Hidden unit pruning

The neural network algorithm described above produces good fits to our experimental data using a maximum of 12 hidden units. However, it is important to minimize the number of hidden units if possible; networks with too many hidden units are difficult to interpret and tend to overfit the training data. To solve this problem we introduce an additional procedure that prunes hidden units until an optimal solution is achieved.

After the training and weight decay algorithm converged for a network with 12 hidden units, one of the hidden units was removed from the network. The output weights and output bias of the new network were optimized using the minimum mean squared error criterion. This procedure was repeated 12 times, each time removing a different hidden unit from the original network. The network with the best performance was then selected, and training continued until this reduced network converged. This pruning process was repeated until we were left with a network that had only one hidden unit.

After pruning was completed, the resulting 12 networks (with 1–12 hidden units) were evaluated in terms of their ability to predict responses to 10% of the training data that had been reserved for this comparison. The neural network with the smallest prediction error on this reserved training set was selected as the network which best described the response properties of the neuron. Note that these reserved training data were entirely separate from the validation data used in final calculation of prediction scores.

2.2.2. Avoiding local minima

One potential pitfall of all gradient-based regression procedures is that they can become trapped in a local minimum that does not represent the optimal solution. To avoid this problem we repeated the entire training procedure ten times, using different initial conditions each time. In previous studies (Lehky et al., 1992), the choice of initial conditions had a profound effect on the final network solution. However inspection revealed that our ten networks displayed very similar response properties, implying that the local minima lie very near each other in parameter space. This may be due to our choice of regularization technique. To choose among these 10 networks the algorithm selected

the one that best predicted responses in the reserved training data described above.

2.2.3. Evaluating predictions on validation data

Once we obtained the final network, its predictive power was determined by computing the correlation coefficient between observed neuronal responses from a separate validation data set and the responses predicted by the network. These data consisted of 10% of the available data (except for three neurons, for which a separate multiple-trial validation set was collected). These data were set aside at the beginning of the analysis and were never used in any aspect of training or pruning. Therefore, there is no danger of the network being overfit to the validation data; prediction scores represent the true predictive power of each network.

Predictions of the responses in the validation data set provide two important measures of model performance. First, predictions allow us to assess statistical significance. A network may achieve a remarkably small training error by overfitting the training data set, so it is impossible to determine whether it provides a good description of the stimulus-response transfer function by inspecting prediction error on the training data alone. However, if the network can achieve good predictions of a separate validation data set, then it must represent the true transfer function of the neuron. We estimated statistical significance of network predictions with a permuted *t*-test (Theunissen et al., 2001); predicted responses were compared to the distribution obtained by repeatedly shuffling the order of responses in the validation set. Our neural networks achieved statistically significant predictions for 29 of the 34 neurons in the sample ($P < 0.05$).

Predicting responses to novel stimuli also allows us to assess the importance of model characteristics. One common metric of the importance of a model is the percentage of response variance that it accounts for. The square of the correlation coefficient between actual and predicted responses is equal to the percentage of response variance accounted for by the prediction. Note that importance is not necessarily related to statistical significance; it is quite possible to achieve a statistically significant result that accounts for a negligible portion of response variance. Significance merely indicates the likelihood that the correlation between the actual and predicted response is due to chance.

2.3. Interpreting the network

A neural network implicitly embodies a nonlinear regression solution, but a separate interpretation algorithm must be used to visualize the transfer function implemented by the network. Previous studies have used two methods for network interpretation: visualization of the network weights (Lehky et al., 1992), and identification of the stimulus-response subspace (Lau et al., 2002).

2.3.1. Network weights

Network weight visualization is a simple way to view all the parameters of the network simultaneously. Recall that the network consists of a set of hidden units each followed by a sigmoidal nonlinearity whose outputs are linearly summed. Fig. 4a illustrates application of this procedure to data acquired using a model simple cell (Fig. 3). Each hidden unit is displayed alongside its sigmoidal activation function which is determined by the network parameters. Its slope is determined by the gain of the unit, its *x*-position by the input bias and its *y*-position by the output bias. Although this format clearly summarizes the network parameters, it does not provide all of the information necessary to understand the transfer function implemented by the network. If the hidden units interact with one another or if the input is limited to a restricted portion of their potential dynamic range, then the network function will be difficult to interpret merely by viewing the network weights.

2.3.2. Principal network dimensions

Another method for interpreting a neural network is to treat the input weights as a set of vectors that describes the stimulus subspace to which a neuron is sensitive (Lau et al., 2002). Multiplication of the stimulus input channels by

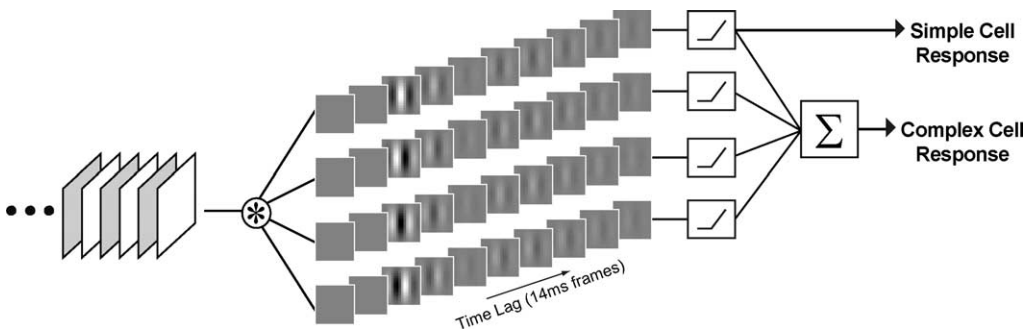

Fig. 3. Model simple and complex cells. A simple cell model was constructed by combining a linear spatial Gabor filter and a biphasic temporal response (center), followed in turn by a rectifying nonlinearity (right). Peak orientation was vertical, peak spatial frequency was two cycles per receptive field, spatial phase was even and peak latency was two time bins. A complex cell model was constructed by summing the rectified output of four simple cells in quadrature spatial phase. The input to the model was a dynamic natural image sequence (left).

the input weights of the network is a linear projection of the input vector onto a new basis set. All the input weights corresponding to a given hidden unit constitute one basis vector. The input weights of the h hidden units define the h-dimensional input space that the network uses to predict responses.

It is convenient to choose a basis set that ranks dimensions by their importance for predicting neuronal responses. We developed a procedure that accomplishes this in several stages. First, we performed singular value decomposition on the input weights to produce a set of orthonormal basis vectors. Second, all of the stimulus vectors were projected linearly onto this basis set. Third, the transformed stimulus vectors were multiplied by the predicted response of the network to each stimulus vector. Finally, singular value decomposition was applied to these transformed, response-weighted stimulus vectors. This procedure produced a set of orthogonal, linearly independent vectors that completely described the input space of the network, ordered by the variability of the response of the network. Each vector describes a dimension of stimulus space and can be displayed as a series of spatial filters at progressively later time lags.

To determine which principal dimensions were most important we examined their associated eigenvalues. Principal dimensions with large eigenvalues were then visualized on a two dimensional graph in which the x-axis represents the projection of a stimulus the principal dimension, and the y-axis represents response rate. Three functions were plotted on this graph: the mean and confidence intervals for the observed response of the model neuron, predicted responses of the network using the identified principal dimension alone, and predicted responses using the principal dimension of interest and all other dimensions simultaneously. By comparing predicted responses of the dimension of interest alone versus predictions of the network as a whole, we could determine whether each principal dimension interacted nonlinearly with other dimensions. (For an example of this procedure, see Fig. 4b.)

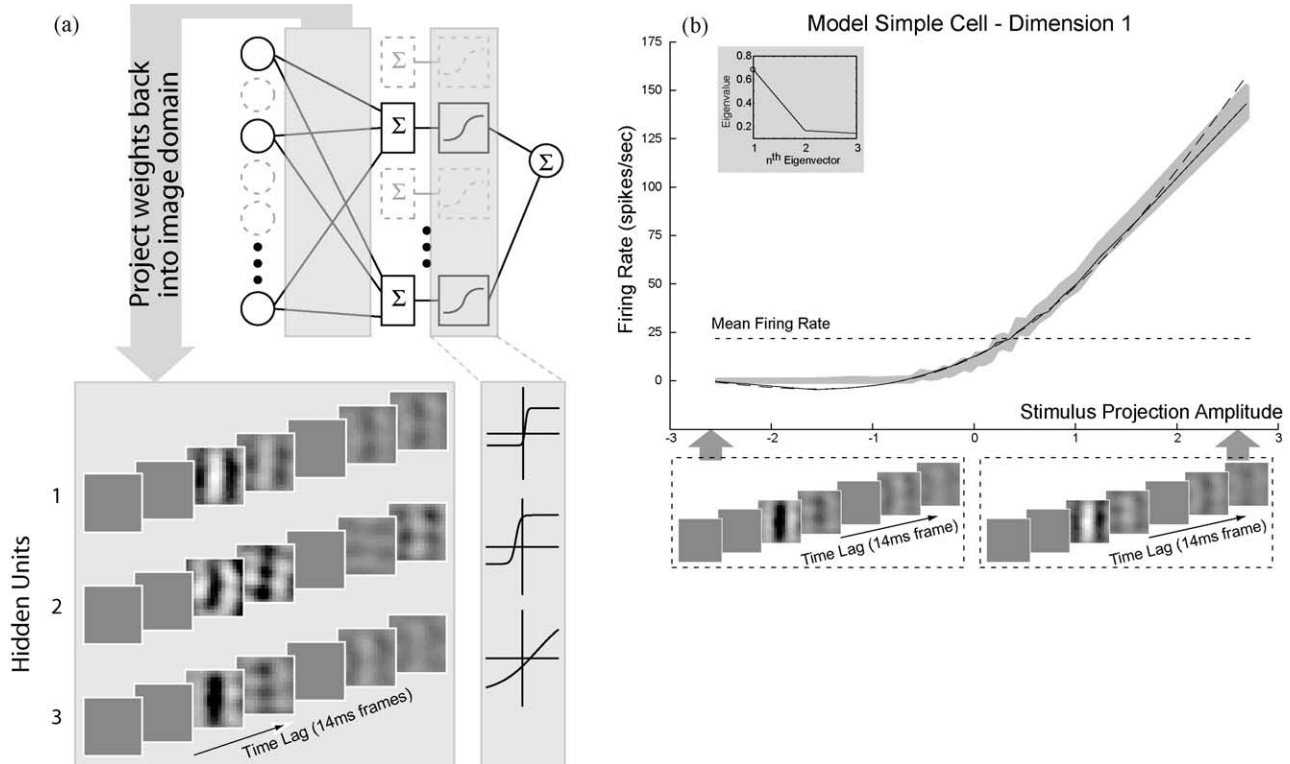

Fig. 4. Network analysis of the model simple cell. (a) Hidden units of a network trained on data acquired from model simple cell (see Fig. 3). Model cell was stimulated with a dynamic natural image sequence. For each hidden unit (1–3), both input weights and sigmoidal nonlinearity are shown. All sigmoids are shown on the same scale. (b) The first principal network dimension of the network shown in (a). The x-axis describes the projection of the stimulus onto the first principal dimension, and the y-axis gives response rate. The spatio-temporal filters shown beneath the x-axis correspond to extreme negative and positive values of the dimension. The inset graph shows the eigenvalues for this dimension (indicated by dot) and the other two principal network dimensions. The large drop in eigenvalues after the first suggests that this network is well described by a single principal dimension. The gray shaded curve represents the mean (center of curve) and two standard errors (boundaries of curve) of model simple cell responses along this principal dimension. (Responses were collapsed into 30 bins, with an equal number of data points per bin.) The solid black line shows the responses predicted by the entire network. The dashed line shows the response of the network using only this particular principal dimension. Because predicted responses of this dimension alone correspond closely to predictions of the network as a whole, it is likely that this dimension does not interact nonlinearly with other any other dimensions. Inspection of the gray curve confirms that the model simple cell responded to stimuli with vertical orientation, odd spatial phase and positive sign. However, because of rectification the model cell did not respond to stimuli with the same orientation and phase, but opposite polarity. The network recovers all the key properties of the model simple cell.

2.4. Explicit models for comparison

To evaluate the performance of the neural network approach, we compared network predictions of a separate validation data set to predictions obtained from two explicit models: a simple linear image domain model, and a nonlinear Fourier power domain model that accounts for phase invariance.

2.4.1. Image domain model

We first constructed a linear image domain model, similar to those previously used to describe V1 simple cells (Jones et al., 1987). The image domain model consisted of a linear spatiotemporal filter followed by rectification. The spatiotemporal filter had 3840 parameters representing the gain of 256 spatial channels (16 × 16 spatial grid) at 15 time lags. To fit the model, the stimulus was windowed and downsampled in space. Reverse correlation was then used to determine the filter providing the minimum mean squared error estimate of responses (Jones et al., 1987). Natural image autocorrelation bias was removed by multiplying the spike-triggered average response by the inverse of the stimulus autocorrelation matrix (Theunissen et al., 2001). The rectification function had one parameter, the activation threshold. This was fit using exhaustive search after selection of the optimal filter. Regularization was implemented by cross validation of the training data.

The linear image domain model should perform similarly to a neural network containing one hidden unit and a rectilinear activation function. The model assumes explicitly that the response is a linear function of stimulus luminance in space and time. This model can be fit quickly, however it cannot account for nonlinear response properties other than spiking threshold.

2.4.2. Fourier power model

We also constructed a Fourier power model that can account for phase invariant responses of V1 complex cells (Theunissen et al., 2001). The Fourier power model consisted of a spectro-temporal filter followed by rectification. The spectro-temporal filter had 1920 parameters representing the gain of 128 spatial channels at 15 time frames. In this model the spatial channels represent the Fourier power of the stimulus after windowing and downsampling to a 16 × 16 spatial grid. (Only half of the 256 spatial coefficients are required because the images are all real, introducing conjugate symmetry into the Fourier transform of the stimulus.) Other procedures for fitting and removing stimulus bias were identical to those used to fit the image domain model.

The Fourier power model assumes explicitly that neuronal responses are a linear function of stimulus Fourier power. The model can account for nonlinear phase invariance and the spiking threshold. However, it discards phase-dependent responses (other than those that are captured by stimulus windowing) and cannot account for other potential nonlinear response properties.

3. Results

The goals of this study were to predict the responses of V1 neurons to dynamic natural image sequences by estimating their nonlinear stimulus-response mapping functions, and to determine whether novel nonlinear response properties revealed by the neural network could be incorporated into an explicit model of the neuronal response function. To accomplish this, we recorded from 34 V1 neurons while stimulating with randomly selected sequences of natural images. We then trained a neural network to recover the stimulus-response mapping function (Fig. 1). Performance of the training algorithm was verified by applying it to model simple and complex cells with well defined response properties.

3.1. Neural network analysis of model neurons

3.1.1. Model simple cell

Performance of the neural network algorithm was first assessed by estimating the stimulus-response function of a model simple cell (Fig. 3). The model consisted of a linear spatial Gabor filter modulated by a biphasic temporal response (Jones & Palmer, 1987), followed by a rectifying nonlinearity. The model neuron preferred vertical orientations, a spatial frequency of two cycles per receptive field, and odd spatial phase. Its maximum response was two time bins (28 ms) after stimulus onset. Responses were generated by stimulating the model cell with a dynamic natural image sequence consisting of 7228 images, typical of those used in our experiments. A multilayer perceptron neural network was then trained using 90% of the available data (see Section 2). Network performance was evaluated by calculating its ability to predict responses to the remaining 10% of the data.

The correlation between the model simple cell responses and the predicted responses of the neural network was 0.88, confirming that the network captured the critical response properties of the model neuron. Note that predictions were not perfect, even though the model contained no noise. These imperfect predictions reflect the limited number of stimuli used to probe the model, and the regularization procedure, which has been optimized for noisy data (see Section 2).

Fig. 4a provides a compact illustration of every parameter of the resulting network. Each hidden unit is displayed along with its sigmoidal activation function. If the hidden units are interpreted as linear spatio-temporal filters, then each panel represents a series of spatial filters at progressively later time lags. The gain of each filter determines the slope of the corresponding sigmoidal nonlinearity. The input and output bias terms

determine *x*- and *y*-position, respectively, and output weights determine amplitude. The hidden units all have similar orientation and spatial frequency tuning, and similar temporal responses. They differ primarily in sign and in the amplitude of their activation functions. This redundancy highlights a limitation of the neural network approach: it is difficult to draw conclusions about the function implemented by the network merely by inspecting the hidden units. Because hidden units are combined non-linearly, complex interactions may arise that cannot be visualized directly.

The underlying function of the network can be visualized more completely by expressing the network in terms of its principal dimensions. The principal network dimensions represent the directions in stimulus space that best describe the predicted neuronal responses. (The procedure for determining network dimensions is fairly involved. It is presented in detail in Section 2.) The first principal dimension of the network trained on model simple cell data accounts for 69% of the total network response variability (i.e. 69% of the total power of all eigenvalues of the principal network dimensions). The first network dimension is shown in Fig. 4b, along with the predicted and observed responses of the model neuron. This dimension clearly recovers the orientation and spatial frequency tuning of the model simple cell. The model neuron and the network both tend to respond when the stimulus takes on positive values in this principal dimension, and neither responds when the stimulus takes on negative values. Thus, the network recovers the behavior of the model simple cell, which is sensitive to stimulus phase.

3.1.2. Model complex cell

In a second test we applied the neural network training algorithm to data acquired using a model complex cell. This model was constructed by summing the rectified output of four simple cells in quadrature phase (see Fig. 3 and Spitzer & Hochstein, 1985). Orientation, spatial frequency, and temporal tuning were identical to the model simple cell, and the same natural image stimulus sequence was used. Once again, the neural network was trained using 90% of the data and performance was assessed with the remaining 10% of the data. The correlation between model complex cell responses and predictions of the neural network is 0.87. As before, this imperfect fit is due to limited stimulus sampling and regularization.

The hidden units of the trained neural network are shown in Fig. 5a. The peak orientation and spatial frequency of several of the hidden units match those of the model neuron. However, the hidden units have different spatial phases, reflecting the phase invariance of the model neuron. Once again, it is difficult to understand the underlying function of the network from a cursory examination of the hidden units.

Fig. 5b and c shows the first two principal dimensions of the network trained on data from the model complex cell. The first two principal dimensions together account for 70% of the network response variability. These dimensions recover the orientation and spatial frequency tuning of the model cell. In fact, they appear to represent the even and odd phases of the linear filters used to construct the model. Both model neuron and network tend to respond when the stimulus takes on either positive or negative values along these two dimensions. (This is in contrast to the network trained on data from the model simple cell, where responses are affected only by positive values of the stimulus along the principal dimension.) The principal dimension analysis reveals that the network captured the response properties of the model complex cell without the need for any prior assumptions about the specific form of the nonlinearities in the response function of the model.

3.2. Neural network analysis of V1 neurons

We trained separate neural networks on data acquired from each of 34 V1 neurons during stimulation with dynamic natural image sequences. Fig. 6 shows the first principal network dimension of the neural network obtained for one neuron. This principal dimension accounts for 47% of the network response variability, and all other network dimensions are substantially less important (see inset, Fig. 6). Inspection of this first principal dimension reveals that the neuron has peak orientation tuning about 30° from vertical, and peak spatial frequency tuning of approximately one cycle per receptive field. The neuron tends to respond when the stimulus is positive along this dimension, but not when the stimulus is negative along this dimension. The asymmetric response pattern suggests that this is a simple cell.

Fig. 7a and b shows the first two principal dimensions of the network obtained for a second neuron. These two dimensions together account for 39% of the network response variability. Both dimensions show peak orientation tuning at vertical and peak spatial frequency tuning of about two cycles per receptive field. These dimensions appear to have even and odd phases, respectively, and the network responds to stimuli with large positive and negative projections along both dimensions. This pattern suggests that this is a complex cell.

3.2.1. Predictions of the neural network versus the image domain model

If the neural network can capture functionally important nonlinear response properties of V1 neurons, it should predict their responses better than a linear model (e.g. the linear spatio-temporal receptive field (Theunissen et al., 2001)). We compared predicted responses to a validation stimulus set from the neural network and a linear image domain model (See Section 2) fit using the same training data.

Fig. 8a compares the predictions obtained using the neural network against those obtained using the linear image domain model for 34 V1 neurons. In 16 neurons

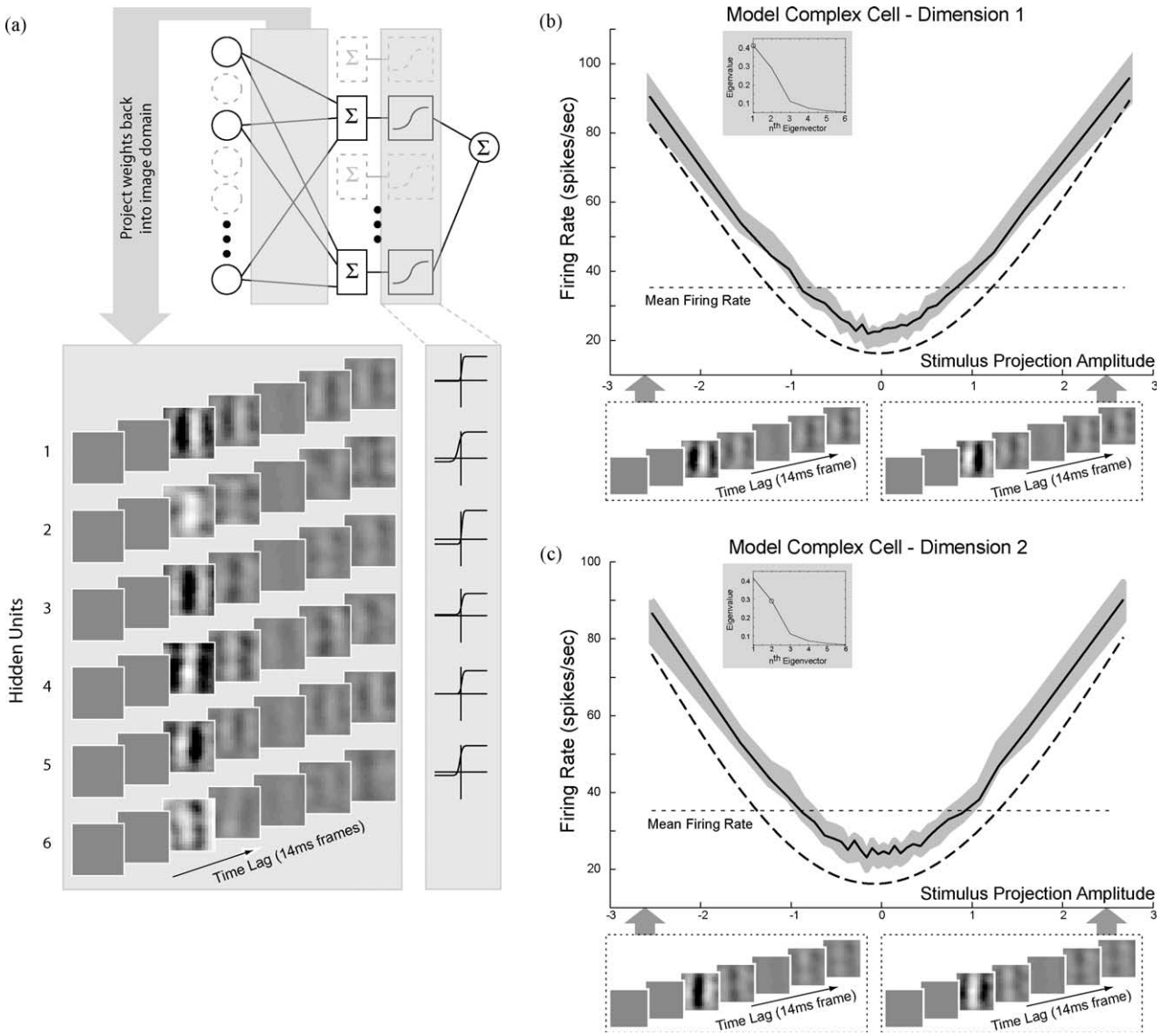

Fig. 5. Network analysis of the model complex cell. (a) Hidden units of a network trained on data acquired from the model complex cell (Fig. 3). The model complex cell was stimulated using the same dynamic natural image sequence as the model simple cell. The layout is as in Fig. 4a. The network used six hidden units to describe the model complex cell. (b) First principal dimension of the network shown in (a). The layout is as in Fig. 4b. Inspection of the gray curve confirms that the model complex cell responded to stimuli with vertical orientation, odd spatial phase and either positive or negative sign. The network recovers this property (black line). (c) Second principal dimension of the network shown in (a). The gray curve reveals that the model cell also responded to stimuli with vertical orientation, even spatial phase and either positive or negative sign. The network also recovers this response property (black line).

(almost 50%), the neural network predicts responses significantly better than the image domain model; in 11 neurons (about 30%) the two models predict equally well; in the remaining 7 neurons the image domain model predicts significantly better ($P < 0.05$, randomized paired t-test).

The neural network can model a wide range of linear and nonlinear functions. However, when the explicit model provides an appropriate description of neuronal responses, the neural network may produce poorer predictions. For example, consider cell r0158A. We have already determined that this appears to be a simple cell (Fig. 6) and the image domain model predicts its responses more accurately than the neural network (Fig. 8a).

3.2.2. Predictions of neural network versus Fourier power model

The neural network predicts the activity of many V1 neurons better than the image domain model. This is consistent with previous theoretical and experimental studies suggesting that the image domain model cannot account for responses of V1 complex cells, which are insensitive to spatial phase (Deangelis et al., 1993; Theunissen et al., 2001). The example in Fig. 7 suggests that the neural network can capture phase invariance. To look at this issue more generally we compared the predicted responses of the neural network to those of a nonlinear Fourier power model that accounts explicitly for phase invariance (see Section 2). Once again both models were fit

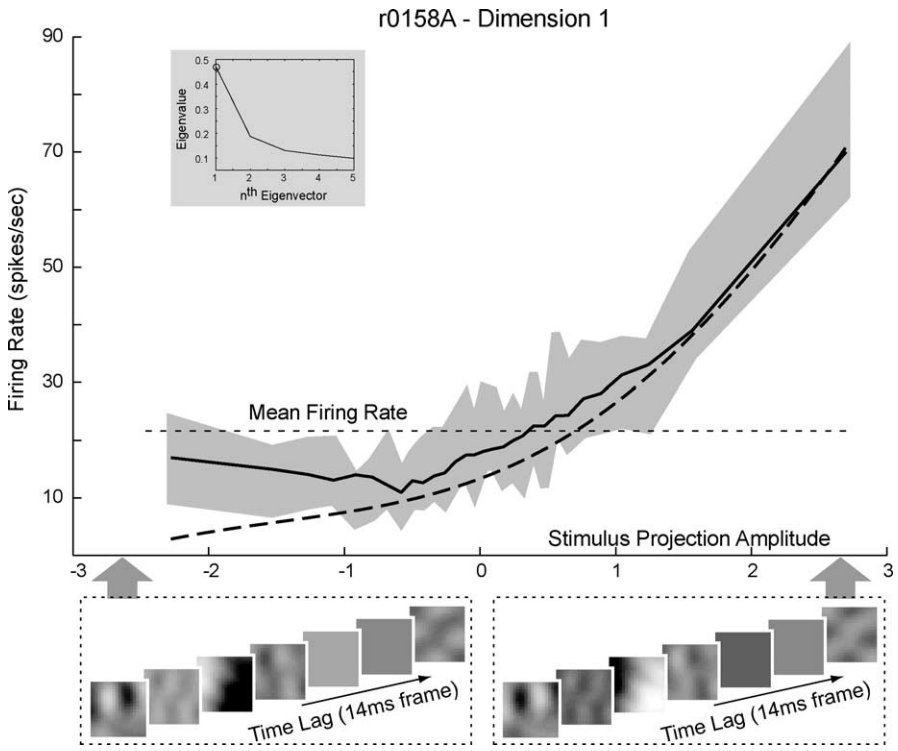

Fig. 6. Network analysis of a V1 simple cell. First principal dimension of a network trained on data acquired from a real V1 neuron (r0158A). The layout is as in Fig. 4b. The eigenvalue plot suggests that this network is well described by a single principal dimension. This dimension represents an orientation of about 30° and low spatial frequencies; peak latency appears to be about 28 ms. Images which are darker on the left and brighter on the right fall on positive values of this dimension while negative values represent images which are brighter on the left and darker on the right. The gray line shows that this neuron responds strongly to positive values of the dimension, but it gives little or no response to negative values. Thus, this appears to be a simple cell which is sensitive to spatial phase.

with the same training data and evaluated using the same validation data.

Fig. 8b compares the predictions of the neural network to those obtained using the Fourier power model for all 34 V1 neurons. For five neurons (about 15%), the neural network predicts responses significantly better than the Fourier power model; in 16 neurons (about 50%) the two models predict equally well; in 13 neurons the Fourier power model predicts significantly better ($P < 0.05$, randomized paired t-test). As mentioned earlier, the neural network may produce poorer predictions when the explicit model provides an appropriate description of neuronal responses. Cell r0284 appears to be a stereotypical complex cell (Fig. 7) and its responses are better predicted by the explicit Fourier power model (Fig. 8b).

For most neurons the predictive power of the Fourier power model is comparable to or slightly better than that of the neural network. In contrast, the image domain model predicts more poorly than the neural network in most cases (compare Figs. 8a and b). Across our sample the mean prediction correlations were 0.24, 0.16 and 0.27 for the neural network, the image domain model and the Fourier power model, respectively. (These seemingly low correlation scores are primarily due to the fact that these data sets consisted almost entirely of single-trial data; see Section 4). This pattern of results suggests that the primary nonlinearity captured by the neural networks is phase invariance, a key feature of V1 complex cells.

3.2.3. Directional selectivity

The responses of many neurons in our sample are predicted equally well by the neural network and the Fourier power model. However, the neural network makes more accurate predictions in a subset of the neurons (see Fig. 8b). These neurons likely possess functionally important properties other than phase invariance.

One such neuron is illustrated in Fig. 9. The correlation between its observed responses and the responses predicted by the network is 0.45, while the correlation with the responses predicted by the Fourier power model is only 0.33. The first two principal network dimensions are shown in Fig. 9. These dimensions reveal that the neuron has peak orientation tuning near horizontal and peak spatial frequency tuning of approximately 2 cycles per CRF. The network responds to stimuli with large positive and negative projections along both dimensions. In addition, inspection of spatial phase tuning over time suggests that this cell is directionally selective. For example, the horizontal bar shown in the first principal dimension moves in a downward direction between frames 3 and 5

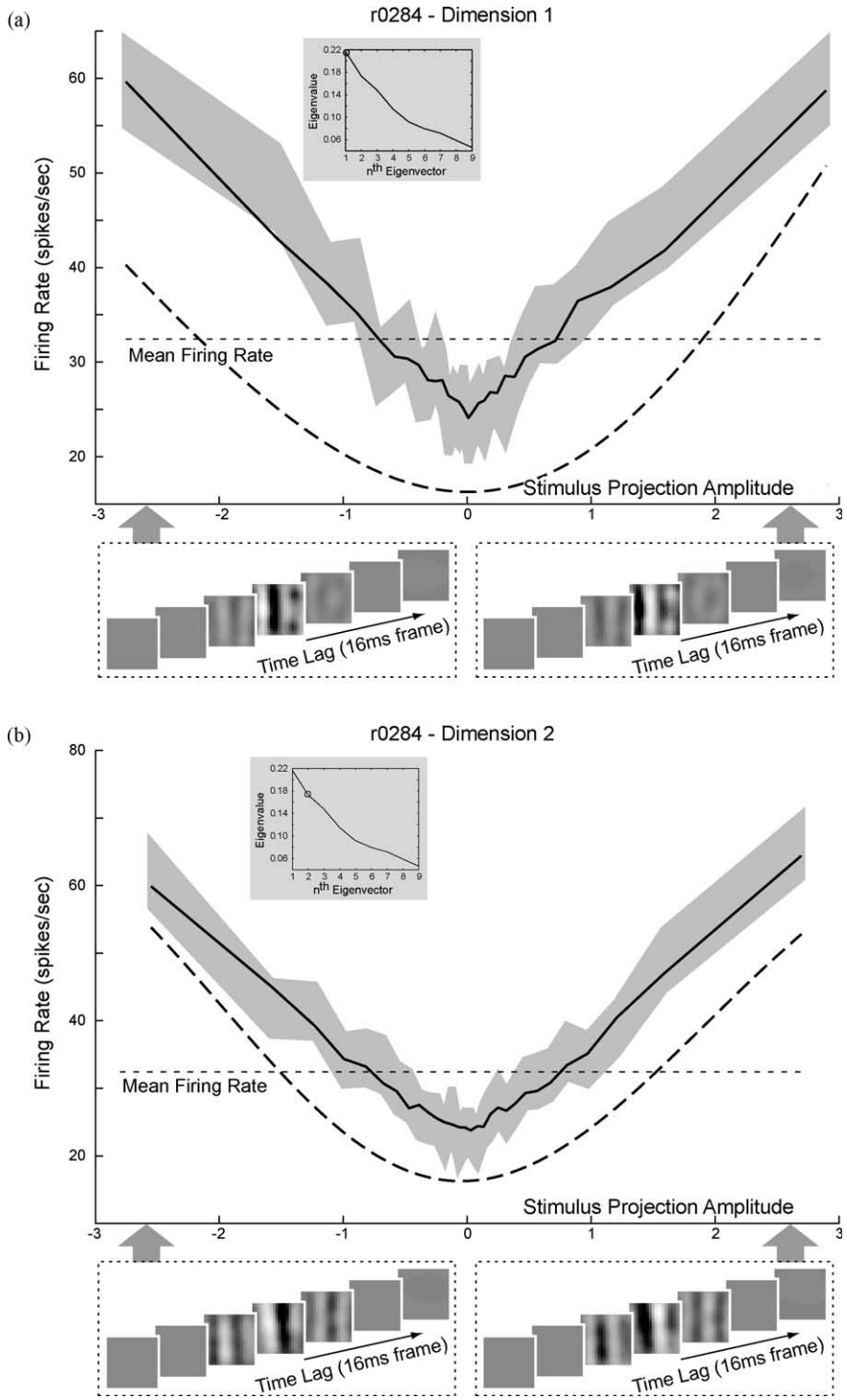

Fig. 7. Network analysis of a V1 complex cell. (a) First principal dimension of a network trained on data acquired from a second V1 neuron (r0284). The eigenvalue plot suggests that this network requires two principal dimensions; the first dimension is shown in this panel. This dimension represents vertical orientation, a spatial frequency of about two cycles per receptive field, and even spatial phase; peak latency is about 32–48 ms. This neuron responds to both positive and negative values of the dimension, suggesting that it is a complex cell. (b) The second principal dimension of the network trained on data acquired from the cell shown in (a). The second dimension is similar to the first, except that it has odd spatial phase. The cell also responds to both positive and negative values of this dimension, confirming that it is a complex cell.

(28–56 ms). Such a shift is characteristic of directional selectivity (Deangelis et al., 1993).

Because the Fourier power model cannot model changes in spatial phase over time, it cannot capture directional selectivity. This might explain why the network produces better predicted responses than the Fourier power model does for neurons such as this one. In a larger sense, this example illustrates how neural network analysis can reveal

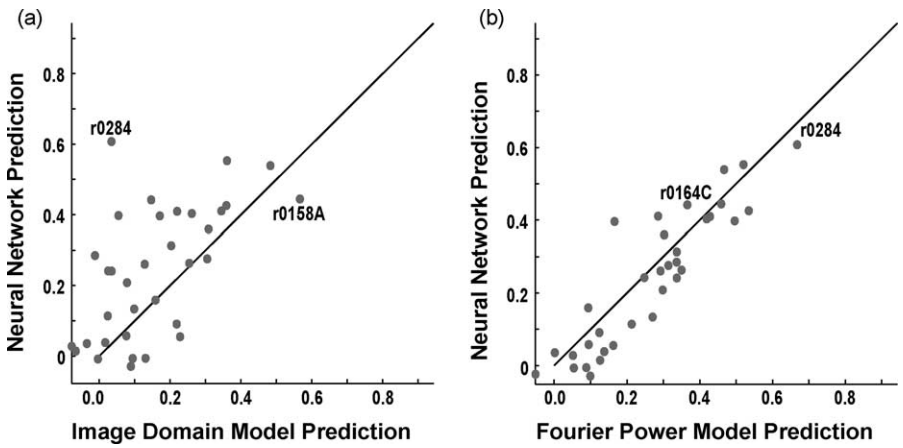

Fig. 8. Predictive power of the neural network versus explicit models. (a) The *x*-axis gives the correlation between the predicted response of the linear image domain model and the validation data for each neuron; the *y*-axis gives the correlation between predicted response of the neural network and the same validation data. Labels indicate the two neurons described in Figs. 6 and 7. Fig. 6 suggested that neuron r0158A is a simple cell, and the responses of this neuron are best predicted by the linear model. In contrast, Fig. 7 suggested that neuron r0284 is a complex cell, and its responses are better predicted by the network. (b) The *x*-axis gives the correlation between the predicted response of the phase invariant Fourier power model and the validation data; the *y*-axis gives the correlation between the predicted response of the neural network and the validation data. Labels indicate the two cells described in Figs. 7 and 9. Responses of neuron r0284 were predicted poorly by the image domain model (a), but are well predicted by the Fourier power model, which accounts for phase invariance. The neural network predicts responses of neuron r0164C better than does the Fourier power model (see Fig. 9).

nonlinear response properties without an explicit model, and without requiring any prior knowledge of the relevant nonlinearity.

4. Discussion

Our experiments demonstrate that neural networks can be used to recover and identify nonlinear response properties of neurons in primary visual cortex. Moreover, these response characteristics can be recovered when complex natural images are used as stimuli, suggesting that this method will be useful for analyzing neuronal responses during natural vision. The neural network method does not require an explicit model, but the response characteristics it recovers can be incorporated into an explicit model if desired. Our success suggests that this method may be useful for analyzing the response properties of neurons in extrastriate visual areas, where few explicit models exist.

Lehky et al. (1992) pioneered the use of artificial neural networks to analyze receptive field properties of neurons in area V1. The same study also included several other important innovations. It was one of the first studies to employ complex naturalistic stimuli in area V1 (see also Creutzfeldt & Nothdurft, 1978), and it evaluated models on the basis of their predictive power. To our knowledge, the only other study to use neural networks to estimate the receptive field properties of neurons in V1 was published by Lau et al. (2002). That study examined both spatial and temporal response properties, and it included a method to facilitate interpretation of the functional properties of the network.

Our study also used an artificial neural network to recover the nonlinear stimulus-response functions of V1 neurons and incorporated the advances of the earlier experiments (Lau et al., 2002; Lehky et al., 1992). Like Lehky et al. (1992), we used complex, natural scenes as stimuli. Like Lau et al. (2002) we investigated both the spatial and temporal aspects of responses. Like both earlier studies we quantified performance by assessing predictive power.

In addition, we developed several other innovations. Our regularization procedures were optimized for high dimensional natural image stimuli and the smaller data sets typically obtained in experiments conducted with awake, behaving animals. We also expanded the network visualization and interpretation methods pioneered by Lau et al. (2002), using our principal network dimension analysis to find the dimensions of stimulus space used by the network and order them by the variability of the network responses to the training data set.

4.1. Comparison of current predictions with those of previous studies

Our study quantified network performance by predicting responses to a novel validation data set not used to train the network. Lau et al. (2002) and Lehky et al. (1992) also used predictions to assess network performance. However, the average prediction scores achieved in the three studies differ substantially. Lehky et al. (1992) reported a mean prediction score of $r = 0.78$; Lau et al. (2002) found $r = 0.45$ for simple cells and 0.31 for complex cells, and we obtained a mean of $r = 0.24$ across the sample. This discrepancy likely

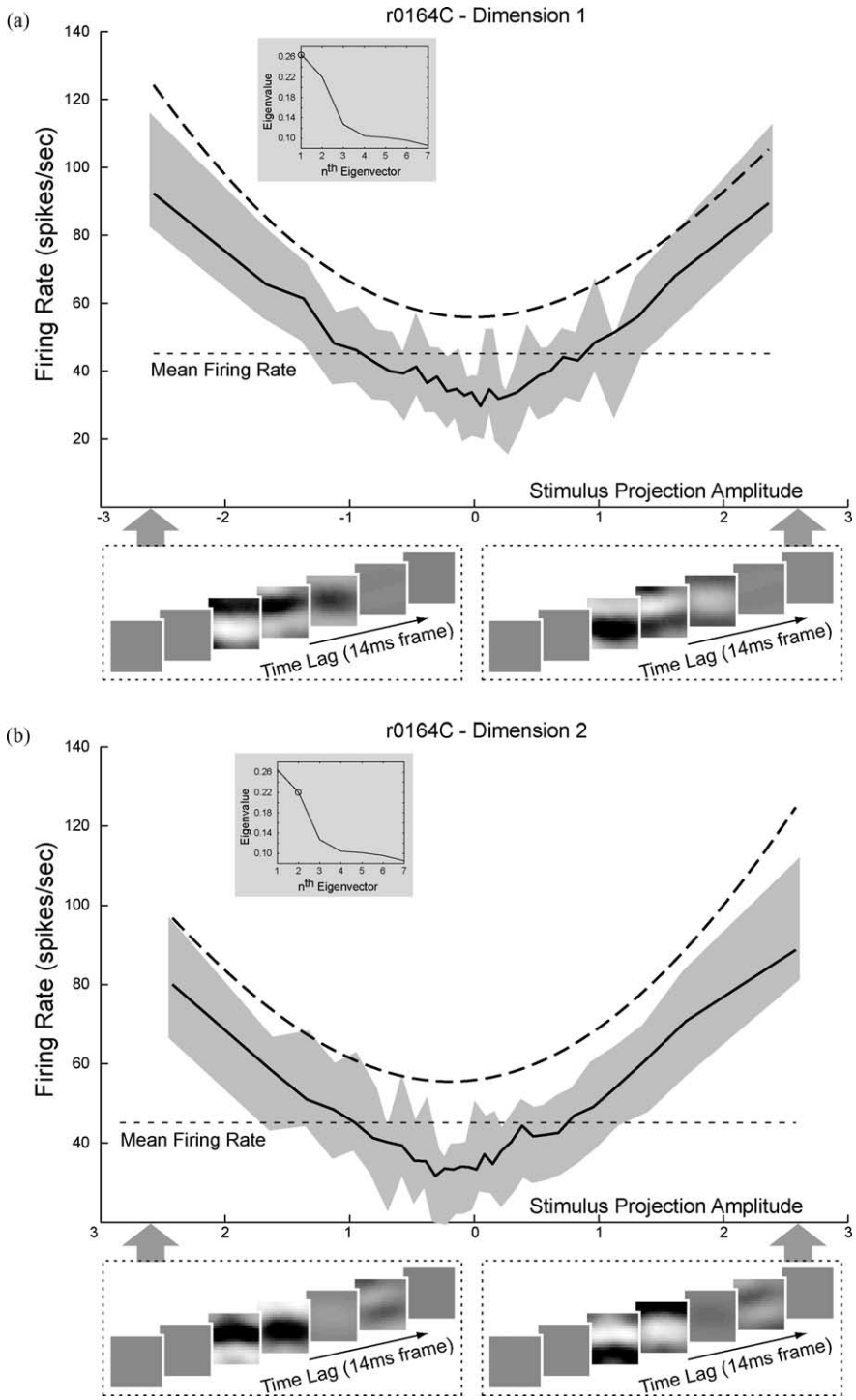

Fig. 9. Network analysis of a directionally selective V1 neuron. (a) First principal dimension of a network trained on data acquired from a third V1 neuron (r0164C). The eigenvalue plot suggests that this network requires two principal dimensions; the first dimension is shown in this panel. This dimension represents horizontal orientation and a spatial frequency of about 2 cycles per receptive field. In contrast to the dimensions recovered in earlier figures, along this dimension spatial phase appears to change continuously over time. This neuron responds to both positive and negative values of the dimension. (b) The second principal dimension of the network trained on data acquired from the neuron in (a). The second dimension is similar to the first, but with orthogonal spatial phase. The neuron also responds to both positive and negative values of this dimension, suggesting that this is a directionally selective complex cell.

reflects several methodological differences between the studies.

The factor most likely to affect the quality of predictions is the number of repeated trials in the validation data set. Because spike trains exhibit Poisson statistics (Tolhurst, Movshon, & Dean, 1983), they tend to have high variability. One way to reduce variability is to average across repeated trials (Tolhurst et al., 1983). Averaging removes Poisson

noise, and hence predictions of average responses are much more accurate than predictions of single spike trains. Both Lau et al. (2002) and Lehky et al. (1992) used repeated trials of validation data. In contrast, our analysis was based almost entirely on data that had been collected as single trials.

Another way to reduce response noise is to integrate responses over a longer time period. This tends to average out variability due to spike timing and accentuate the mean response rate. Lehky et al. (1992) used a slow stimulus refresh rate, and responses were binned at 160 ms. Lau et al. (2002) smoothed data with a Gaussian filter ($\sigma = 10$ ms) and used a binning window of 16 ms. We also used a binning window of either 14 or 16 ms, but with no smoothing. These differences in temporal integration also likely affected predictions. However, our own analyses suggest that the number of unique stimuli in the training data set has a much larger effect on predictions than the temporal integration window.

One other factor that might influence predictions is eye movements smaller than the size of the fixation window. Both Lau et al. (2002) and Lehky et al. (1992) recorded from neurons in anesthetized animals, while our experiments used awake, behaving animals trained to perform a fixation task. We limited fixation to within 0.35° of the fixation spot, but data from awake animals are always contaminated by microsaccades and slow drifts whose effects cannot be removed entirely (Gur & Snodderly, 1997). These residual eye movements introduce noise and tend to reduce prediction scores.

4.2. Comparison of stimuli and response characterization with previous studies

One of the most important factors determining network performance and prediction scores is the nature of the stimuli used to train the network. In general, the smaller the stimulus space, the fewer stimulus-response data points will be required to achieve a good prediction. On the other hand, the smaller the stimulus space, the less likely that the network will accurately predict responses to stimuli falling outside the range of the training set.

Lehky et al. (1992) used a range of complex stimuli, including both synthetic patterns and naturalistic textures. The stimulus set included 400 different images and these were repeated 30 times each. These authors only visualized the network by examining the hidden units, so it is unclear how well the estimated networks recovered tuning for orientation, spatial frequency and phase. However, the estimated networks produced predictions with a substantially higher correlation to actual responses than those of the later studies, even though the stimulus set was rather small.

Lau et al. (2002) used bars aligned with the optimal orientation of each neuron. An array of 16 bars spanned the receptive field on an axis orthogonal to the preferred orientation. Their contrast was controlled by an m-sequence that was 32,767 frames long and repeated 3 times. When trained with this data set, each network provided an efficient estimate of phase and direction tuning. However, their networks provided no information about orientation tuning, as this was established for each neuron in a preliminary test.

Our experiments used random natural image sequences and preliminary tests were only used to identify the location of the receptive field. The image sequences were 9937 frames long on average and for most neurons they were never repeated. Still, our networks were able to recover spatial frequency, orientation, phase and direction tuning. Considering the results of all three studies, it appears that repeated trials of validation data can be predicted more accurately, but repeated trials of training data necessarily reduce the number of different stimuli that can be presented in finite time. In contrast, a large and varied training stimulus set is most useful for characterizing linear and nonlinear responses across many stimulus dimensions.

4.3. Comparison with other methods for estimating receptive field properties

Previous studies have used other nonlinear regression methods to characterize the stimulus-response transfer function of V1 neurons. Early experiments used Wiener kernel analysis (Emerson, Citron, Vaughn, & Klein, 1987; Jones et al., 1987). Wiener kernel analysis requires spectrally and temporally white stimuli and requires impractically large data sets to recover more than lower-order nonlinearities.

More recent studies have approached this same problem by means of spike triggered covariance (Brenner, Bialek, & de Ruyter van Steveninck, 2000; Touryan, Lau, & Dan, 2002). Spike triggered covariance produces ordered dimensions of visual space similar to those of our principal network dimension analysis. In theory it can be used even when stimulus statistics are biased (e.g. with natural scenes), and it is less computationally intensive than the neural network approach. However, spike triggered covariance only assesses the second order statistics of the stimulus-response function. It is not sensitive to higher-order nonlinearities and does not model interactions between the recovered dimensions.

The neural network method does not require that the stimuli have any particular statistical properties. It can recover arbitrary nonlinear response properties, even when they represent nonlinear interactions between many orthogonal dimensions. We expect that the generality of the neural network approach will prove critical for recovering nonlinear stimulus-response functions in extrastriate visual areas, where no explicit processing model is available.

Acknowledgements

This work was supported by grants to JLG from NEI and NIMH. MW was supported by a DOE-CSGF fellowship and

SVD was supported by an NSF fellowship. We thank William Vinje, Ben Willmore, and Ben Hayden for data acquisition; and Jamie Mazer and Kate Gustavsen for helpful comments on the manuscript.

References

Barron, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, *39*(3), 930–945.

Brenner, N., Bialek, W., & de Ruyter van Steveninck, R. (2000). Adaptive rescaling maximizes information transmission. *Neuron*, *26*(3), 695–702.

Connor, C. E., Gallant, J. L., Preddie, D. C., & Van Essen, D. C. (1996). Responses in area V4 depend on the spatial relationship between stimulus and attention. *Journal of Neurophysiology*, *75*(3), 1306–1308.

Creutzfeldt, O. D., & Nothdurft, H. C. (1978). Representation of complex visual stimuli in the brain. *Naturwissenschaften*, *65*(6), 307–318.

David, S. V., Vinje, W. E., & Gallant, J. L. (1999). Natural image reverse correlation in awake behaving primates. *Society for Neuroscience Abstracts*, *25*, 1935.

Deangelis, G. C., Ohzawa, I., & Freeman, R. D. (1993). Spatiotemporal organization of simple-cell receptive-fields in the cats striate cortex. 1. General-characteristics and postnatal-development. *Journal of Neurophysiology*, *69*(4), 1091–1117.

Emerson, R. C., Citron, M. C., Vaughn, W. J., & Klein, S. A. (1987). Nonlinear directionally selective subunits in complex cells of cat striate cortex. *Journal of Neurophysiology*, *58*(1), 33–65.

Field, D. J. (1987). Relations between the statistics of natural images and the response properties of cortical cells. *Journal of the Optical Society of America A*, *4*(12), 2379–2394.

Gur, M., & Snodderly, D. M. (1997). Visual receptive fields of neurons in primary visual cortex (V1) move in space with the eye movements of fixation. *Vision Research*, *37*(3), 257–265.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Jones, J. P., & Palmer, L. A. (1987). An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, *58*(6), 1233–1258.

Jones, J. P., Stepnoski, A., & Palmer, L. A. (1987). The two-dimensional spectral structure of simple receptive-fields in cat striate cortex. *Journal of Neurophysiology*, *58*(6), 1212–1232.

Lau, B., Stanley, G. B., & Dan, Y. (2002). Computational subunits of visual cortical neurons revealed by artificial neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(13), 8974–8979.

Lehky, S. R., Sejnowski, T. J., & Desimone, R. (1992). Predicting responses of nonlinear neurons in monkey striate cortex to complex patterns. *Journal of Neuroscience*, *12*(9), 3568–3581.

Mackay, D. J. C. (1995). Probable networks and plausible predictions: a review of practical bayesian methods for supervised neural networks. *Network-Computation in Neural Systems*, *6*(3), 469–505.

Marmarelis, P. Z., & Marmarelis, V. Z. (1978). *Analysis of physiological systems: The white noise approach*. New York: Plenum.

Moller, M. F. (1993). A scaled conjugate-gradient algorithm for fast supervised learning. *Neural Networks*, *6*(4), 525–533.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536.

Spitzer, H., & Hochstein, S. (1985). A complex-cell receptive-field model. *Journal of Neurophysiology*, *53*(5), 1266–1286.

Theunissen, F. E., David, S. V., Singh, N. C., Hsu, A., Vinje, W. E., & Gallant, J. L. (2001). Estimating spatial temporal receptive fields of auditory and visual neurons from their responses to natural stimuli. *Network: Computation in Neural Systems*, *12*(3), 289–316.

Tolhurst, D. J., Movshon, J. A., & Dean, A. F. (1983). The statistical reliability of signals in single neurons in cat and monkey visual-cortex. *Vision Research*, *23*(8), 775–785.

Touryan, J., Lau, B., & Dan, Y. (2002). Isolation of relevant visual features from random stimuli for cortical complex cells. *Journal of Neuroscience*, *22*(24), 10811–10818.

Vinje, W. E., & Gallant, J. L. (2002). Natural stimulation of the non-Classical receptive field increases information transmission efficiency in V1. *Journal of Neuroscience*, *22*(7), 2904–2915.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *Ieee Transactions on Acoustics Speech and Signal Processing*, *37*(3), 328–339.