Biol Cybern (2017) 111:207–227
DOI 10.1007/s00422-017-0716-z

ORIGINAL ARTICLE

Biological
Cybernetics

An insect-inspired model for visual binding II: functional analysis and visual attention

Brandon D. Northcutt[1] · **Charles M. Higgins**[2]

Received: 2 April 2016 / Accepted: 27 February 2017 / Published online: 16 March 2017

Abstract We have developed a neural network model capable of performing visual binding inspired by neuronal circuitry in the optic glomeruli of flies: a brain area that lies just downstream of the optic lobes where early visual processing is performed. This visual binding model is able to detect objects in dynamic image sequences and bind together their respective characteristic visual features—such as color, motion, and orientation—by taking advantage of their common temporal fluctuations. Visual binding is represented in the form of an inhibitory weight matrix which learns over time which features originate from a given visual object. In the present work, we show that information represented implicitly in this weight matrix can be used to explicitly count the number of objects present in the visual image, to enumerate their specific visual characteristics, and even to create an enhanced image in which one particular object is emphasized over others, thus implementing a simple form of visual attention. Further, we present a detailed analysis which reveals the function and theoretical limitations of the visual binding network and in this context describe a novel network learning rule which is optimized for visual binding.

Keywords Neural networks · Blind source separation · Visual binding · Object perception · Visual attention · Artificial intelligence

1 Introduction

Visual binding is the process of grouping together the visual characteristics of one object while differentiating them from the characteristics of other objects (Malsburg 1999), without regard to the spatial position of the objects. Based on recently identified structures termed *optic glomeruli* in the brains of flies and bees (Strausfeld et al. 2007; Strausfeld and Okamura 2007; Okamura and Strausfeld 2007; Paulk et al. 2009; Mu et al. 2012), we have developed a neural network model of visual binding (Northcutt et al. 2017), which encodes the visual binding in a pattern of inhibitory weights. This model's genesis was in the anatomical similarity of insect olfactory and optic glomeruli and was inspired by the work of Hopfield (1991), who modeled olfactory binding based on temporal fluctuations in the mammalian olfactory bulb, and by the seminal work of Herault and Jutten (1986) on blind source separation (BSS).

A pattern of inhibitory weights is learned by the binding network from visual experience based upon common temporal fluctuations of spatially global visual characteristics, with the underlying suppositions being that the visual characteristics of any given object fluctuate together, and differently from other objects. An example would be when an automobile passes behind an occluding tree: its color, motion, form, and orientation disappear and then reappear together, thus undergoing a common temporal fluctuation.

A detailed description and demonstration of the function of this model is given in a companion paper (Northcutt et al. 2017). In the present work, we describe the essentials of

✉ Brandon D. Northcutt
brandon@northcutt.net

Charles M. Higgins
higgins@neurobio.arizona.edu

[1] Department of Electrical and Computer Engineering, University of Arizona, 1230 E. Speedway Blvd., Tucson, AZ 85721, USA

[2] Departments of Neuroscience and Electrical/Computer Eng., University of Arizona, 1040 E. 4th St., Tucson, AZ 85721, USA

the model, present one representative demonstration of its function in visual binding, and then show how to explicitly interpret the binding output, which the network encodes only implicitly in the pattern of inhibitory weights. We show how to count the number of objects present in the input image sequence and enumerate their characteristics and relative strength. Further, we show how to use this information to create an enhanced image, emphasizing one particular object while de-emphasizing all others, thereby implementing a form of visual attention. Finally, we present a theoretical analysis of the functional limitations and capabilities of this neural network model to provide a better understanding of the network and its potential.

2 The visual binding model

Model simulations were performed in MATLAB (The MathWorks, Natick, MA) using a simulation time step of $\Delta t = 10$ ms and image sizes of 500 × 500 pixels.

Figure 1 shows a diagram of the full two-stage neural network used. This network begins with a first stage comprised of three separate, fully connected recurrent inhibitory neural networks used for refining the representation of motion, orientation, and color. The outputs of the first stage are fed into a second stage, a fully connected ten-unit inhibitory neural network which performs visual binding. Despite their apparently disparate purposes, all four neural networks comprising the overall model operate using *exactly the same* temporal evolution and learning rules, which are given below.

Each of the four recurrent neural networks used in the model (three in the first stage processing individual visual submodalities, and one in the second stage performing visual binding) learned, by using common temporal fluctuations, an inhibitory weight matrix that indicated the degree of inhibition between neurons within each network. We describe the essentials of the operation and training of these networks below; see Northcutt et al. (2017) for full details.

2.1 Computation of inputs to the network

Despite the fact that fly color vision is based on green, blue, and ultraviolet photoreceptors (Snyder 1979), for ease of human visualization and computer representation—and without loss of generality—our color images had red, green, and blue (RGB) color planes. Since this network is based on a model of the insect brain, an elaborated version of the Hassenstein–Reichardt motion detection algorithm (Hassenstein and Reichardt 1956; Santen and Sperling 1985)—the standard model of insect elementary motion computation—was used to compute motion inputs, and difference-of-Gaussian (DoG) filters—the best existing model of early insect orientation processing (Rivera-Alvidrez et al. 2011)—were convolved with the image to compute orientation.

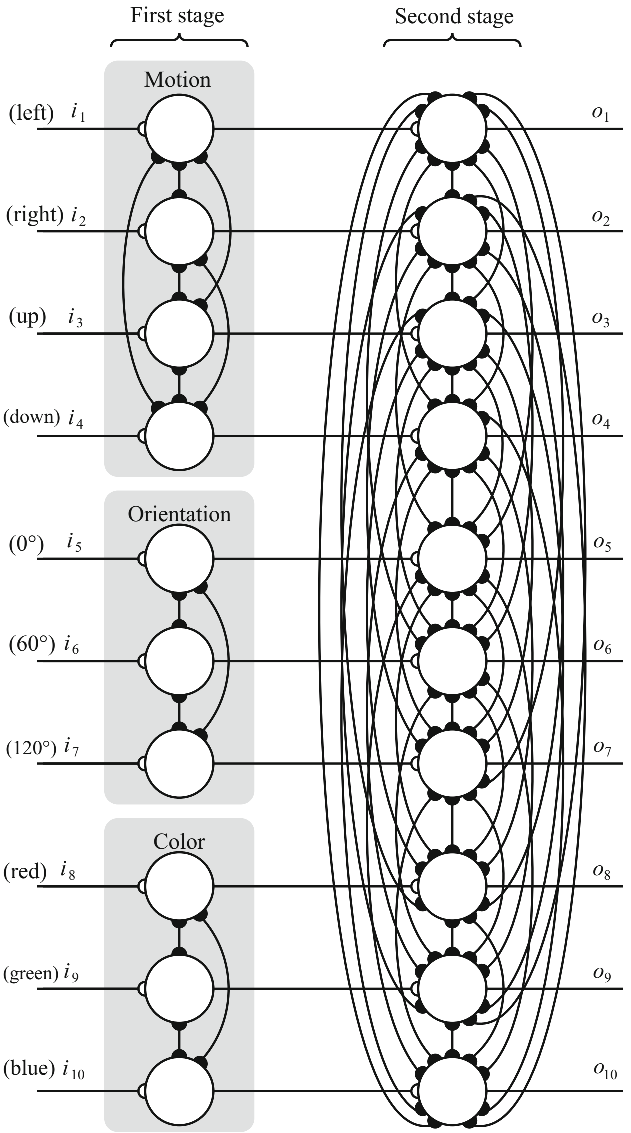


Fig. 1 The two-stage network for visual submodality refinement and visual binding. *Large circles* represent units in the neural network. *Unshaded half-circles* at connections indicate excitation, and *filled half-circles* indicate inhibition. During the first phase of training, neurons in the three first-stage networks learn to mutually inhibit one another, thus refining the representation of motion, orientation, and color and resulting in more selective tunings in each submodality. In the second phase of training with a stimulus comprised of moving *bars*, the second-stage visual binding network learns the relative strengths of visual object features based on common temporal fluctuations and develops an inhibitory weight matrix to produce outputs that reflect the temporal fluctuations unique to each object

Inputs to the neural network were computed as diagrammed in Fig. 2. For achromatic motion and orientation processing, each RGB image was first converted to grayscale by taking the average of the 3 color components.

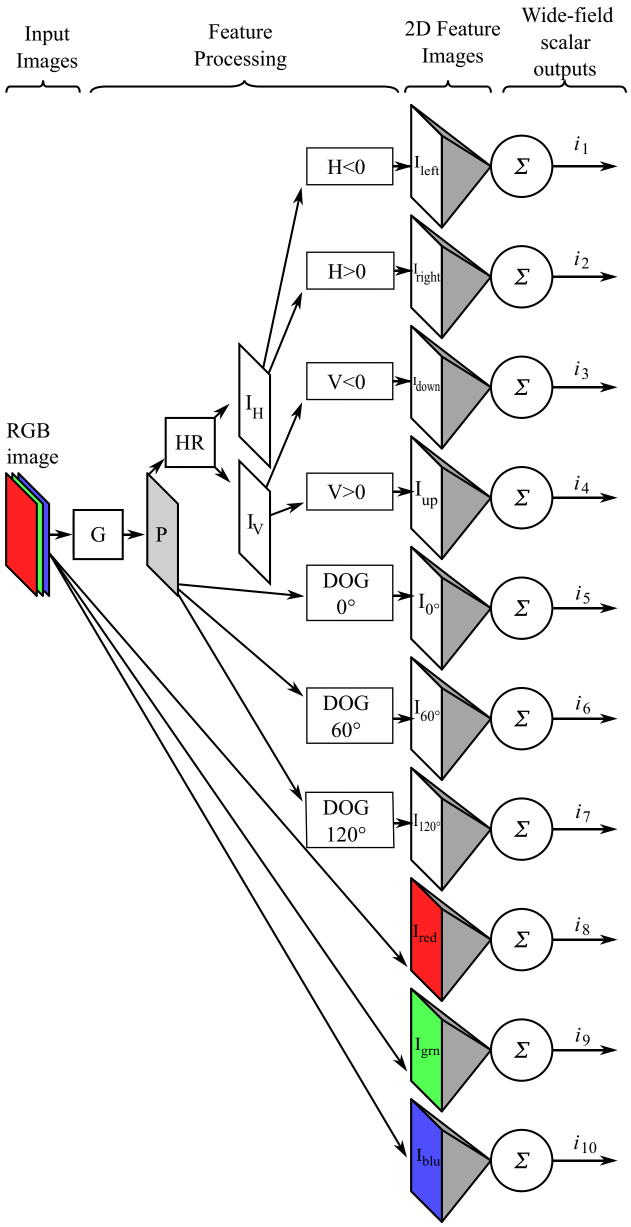


Fig. 2 Diagram of wide-field visual input computation from input images. Each input RGB image was converted to grayscale (G) for orientation and motion processing. Image motion was computed by the Hassenstein–Reichardt (HR) elementary motion detection model in the horizontal (I_H) and vertical (I_V) directions and then separated into four feature images I_{left}, I_{right}, I_{down}, and I_{up}, respectively, containing strictly positive leftward ($H < 0$), rightward ($H > 0$), downward ($V < 0$), and upward ($V > 0$) components. Three orientation-selective DoG filter kernels were convolved with the grayscale image to produce three orientation feature images $I_{0°}$, $I_{60°}$, and $I_{120°}$. The individual red, green, and blue color planes were taken as the final three feature images I_{red}, I_{grn}, and I_{blu}. Each of these ten 2D feature images was then spatially summed to create a scalar value representing wide-field image feature content. The inputs $i_1(t)$ through $i_{10}(t)$ became input to the neural network model of Fig. 1

The Hassenstein-Reichardt motion algorithm was iterated across rows and columns of the grayscale image, resulting in vertical and horizontal "motion images" containing signed local motion outputs. These were further subdivided into non-negative leftward, rightward, downward, and upward motion "feature images" by selecting components of each motion image with a particular sign. DoG filters selective for three orientations (0°, 60°, and 120°) were convolved with the grayscale images to create three orientation feature images. Finally, the red, green, and blue color planes were used as color feature images.

Each of these ten two-dimensional (2D) feature images was spatially summed over both dimensions to produce ten scalar measures of full-image feature content. Each group of scalar signals corresponding to a given visual submodality (motion, orientation, or color) was then normalized to a maximum value of unity, allowing features from different submodalities to be comparable while preserving ratios of features within each group. This normalization divided each group of signals by scalar factors $n_m(t)$, $n_o(t)$, and $n_c(t)$ computed as the maximum value of any signal in the group during the last 2 s, thereby automatically scaling inputs from different submodalities to become comparable with one another and simultaneously adapting the signals to changing visual conditions. The ten normalized scalar values were provided at each simulation time step as inputs $i_1(t)$ through $i_{10}(t)$ to the neural network of Fig. 1.

2.2 Network temporal evolution

All inputs to each of the four recurrent networks in the model were high-pass filtered before being processed. This ensured that static features of the visual scene such as an unchanging background never became input to the network.

As described in detail in a companion paper (Northcutt et al. 2017), the activation of each neuron in the model—which may be positive or negative—represents for non-spiking neurons the graded potential of the neuron relative to its resting potential and for spiking neurons the average firing rate relative to the spontaneous rate.

The activation $o_n(t)$ of neuron n in a recurrent inhibitory neural network may be modeled as

$$o_n(t) = i'_n(t) - \sum_{k=1}^{N} W_{n,k} \cdot o_k(t - \tau_i) \tag{1}$$

where $i'_n(t)$ represents a first-order temporal high-pass filtered version of the input $i_n(t)$ using time constant $\tau_{HI} = 1.0$s, $W_{n,k}$ represents the strength of the inhibitory synaptic pathway from neuron k to neuron n, $o_k(t)$ represents the activation of a different neuron k in the network, and τ_i represents the small but finite delay required to produce inhibition. This set of equations can be expressed in matrix form as

$$\boldsymbol{o}(t) = \boldsymbol{i}'(t) - \boldsymbol{W} \cdot \boldsymbol{o}(t - \tau_i) \tag{2}$$

where lowercase bold symbols indicate N-element column vectors and uppercase bold symbols represent $N \times N$ matrices. Since the biophysical details of optic glomeruli—and thus τ_i—are unknown, but the existence of a finite delay is crucial (as detailed in Northcutt and Higgins 2017), in our simulations we formulate the network temporal dynamics as

$$\boldsymbol{o}(t) = \boldsymbol{i}'(t) - \boldsymbol{W} \cdot \boldsymbol{o}(t - \Delta t) \tag{3}$$

where Δt is the simulation time step. By using $\tau_i = \Delta t$, we provide the smallest finite inhibition delay possible in our simulation. This equation was used in all simulations.

When the time scale of input and output changes is much larger than the simulation time step Δt, (3) may be approximated as

$$\boldsymbol{o}(t) = \boldsymbol{i}'(t) - \boldsymbol{W} \cdot \boldsymbol{o}(t) \tag{4}$$

Apart for the use of high-pass filtered inputs, (4) is a common formulation for a fully connected recurrent inhibitory neural network used in BSS (Herault and Jutten 1986; Jutten and Herault 1991; Cichocki et al. 1997). However, (4) represents an idealized system, for which the outputs may be instantaneously computed from the inputs so long as the matrix $[\boldsymbol{I} + \boldsymbol{W}]^{-1}$ exists ($\boldsymbol{I}$ being the identity matrix). This system of equations can be singular, but, quite unlike any realistic recurrent neuronal network, cannot be temporally unstable.

The use of (3) for temporal dynamics instead of the more common, seemingly quite reasonable approximation of (4) allows for modeling of the temporal instability of recurrent neuronal networks—which, as shown below, is crucial to understanding their function—while still allowing network temporal evolution to be approximated by (4) when required to make theoretical analysis tractable and relate the present network to previous studies.

As detailed in a companion paper (Northcutt et al. 2017), the temporal stability of linear systems such as the one described by (3) has long been well understood (Trentelman et al. 2012), and stability of such a network may be maintained by simply requiring that the magnitude of all eigenvalues of the weight matrix $\boldsymbol{W}$ be less than unity.

2.3 Network learning rule

The Hebbian-style network learning rule used to generate inhibitory weight matrices based on common temporal fluctuations of the inputs was modified from that of Cichocki et al. (1997). This spatially asymmetric multiplicative weight update rule was chosen to support the representation of neuronal activation described in Sect. 2.2—which does not explicitly represent neuronal action potentials—and to leverage existing theoretical work on neural network solutions to BSS problems, in awareness of the fact that the underlying biological basis of this learning is likely to be spike-timing-dependent plasticity (Markram et al. 1997) as discussed in Northcutt et al. (2017).

Weight matrices $\boldsymbol{W}$ were initialized to zero so that the state of the network was $\boldsymbol{o}(t) = \boldsymbol{i}'(t)$ and thus network outputs were initially identical to high-pass filtered inputs. The network learning rule is formulated as

$$\frac{\mathrm{d}W_{n,k}}{\mathrm{d}t} = \gamma \cdot \mu(t) \cdot g(o'_n) \cdot f(o'_k) \tag{5}$$

where $n \neq k$ are neuron indices and γ is a scalar learning rate. Diagonal elements of the weight matrix always remained at zero, preventing self-inhibition. Any element of $\boldsymbol{W}$ that became negative after a learning rule update was set to zero, thereby enforcing that network weights were strictly inhibitory.

$\mu(t)$ is a learning onset function that has a value of zero at the start of training and rises asymptotically to unity with a time constant of 2 s. Our formulation of $\mu(t)$—quite unlike the identically named function used by Cichocki et al.—is used to gradually turn on the learning rule at the start of training to avoid a powerful transient in weights based solely on the initial phase of the inputs.

$o'_n(t)$ and $o'_k(t)$, respectively, indicate first-order temporal high-pass filtered versions of network outputs n and k with time constant $\tau_{HO} = 0.5$ s. Note that use of high-pass filtered outputs in the learning rule, rather than simply the outputs, makes the network's learning dependent on *temporal fluctuations* of the inputs: specifically those fluctuations with temporal frequencies greater than the cutoff frequency of the output high-pass filter.

We used network "activation functions" $f(x) = x^3$ and $g(x) = \tanh(\pi x)$ similar to those used by previous authors (Jutten and Herault 1991; Cichocki et al. 1997) to introduce higher-order statistics of the filtered outputs into the learning rule (Hyvärinen and Oja 1998), although the positions of these two functions with respect to rows n and columns k of the weight matrix are exchanged in (5) as compared to the conventional BSS learning rule. This exchange is crucial to the function of our model, and both the role of these activation functions and the requirement that they be exchanged for our model are addressed in detail in Sect. 5.3.5.

For reasons that will become apparent later, we will refer to the learning rule with activation functions in their conventional positions as the *cooperative learning rule*. In contrast, we will refer to the learning rule of (5) with exchanged activation functions as the *competitive learning rule*.

2.4 Training of the model

Before training of any network began, a visual input was presented and all linear filters and the input adaptive scaling algorithm were allowed to reach steady state to eliminate artifactual startup transients.

Training began with each of the three first-stage networks using a learning rate of $\gamma_1 = 50$. This training resulted in the first-stage motion, orientation, and color networks, respectively, learning weight matrices $\boldsymbol{M}$, $\boldsymbol{O}$, and $\boldsymbol{C}$. To rapidly give the first-stage networks sufficient experience to refine each visual submodality, an artificial visual stimulus (detailed in Northcutt et al. 2017) was presented that provided simultaneous temporal fluctuations in all colors, orientations, and directions of motion. This stimulus provided near-identical signals to each input of every network, effectively reducing the learning rule of (5) to a purely Hebbian one (Hebb 1949).

This input resulted in uniform symmetric (zero diagonal) weight matrices, indicating uniform lateral inhibition: a well-known technique for sensory refinement (Linster and Smith 1997). However, with this symmetric stimulus weight matrices never converge to a stable state, but rather increase in value as long as training continues. For this reason, learning for each first-stage network was terminated by setting its respective learning rate γ_1 to zero when the maximum magnitude of any eigenvalue of the network weight matrix reached a value of $V_{1,\max} = 0.9$. This procedure allowed us to rapidly learn strong lateral inhibition in the first stage while avoiding temporally unstable recurrent networks.

During this first-stage training period, the learning rate γ_2 for the second stage was set to zero. While not strictly necessary, this isolation of the two stages allowed for rapid training of the first-stage network and a clear demonstration of second-stage function.

After first-stage learning was complete, the second-stage learning rate was set to $\gamma_2 = 0.5$, after which visual stimuli composed of multiple objects and intended to demonstrate visual binding were presented, as shown in the next section. In this second phase of training, the second-stage network learned an inhibitory connection matrix $\boldsymbol{T}$ indicating the binding among visual submodalities.

To avoid temporal instability of the second-stage network, if any weight matrix update resulted in $\boldsymbol{T}$ having an eigenvalue with a magnitude V greater than $V_{2,\max} = 0.95$, the weight matrix was multiplied by a scalar factor $V_{2,\max}/V$, thus holding the maximum eigenvalue at $V_{2,\max}$ and maintaining network temporal stability.

3 An example of visual binding

To demonstrate the operation of the visual binding network, an artificial visual stimulus composed of moving 50 × 12-pixel bars on a black background was presented. This stimulus consisted of a red bar that started near the upper left corner of the image and moved down and right at −30°, and a green bar that started near the upper right and moved down and left at 210°. Both bars were oriented with their longest axis orthogonal to the direction of motion and moved at 50 pixels per second.

The bars moved through a fixed pattern of multiplicative horizontal sinusoidal shadow with a spatial period of 50 pixels, a mean value of 0.5, and an amplitude of 0.25. As the bars moved independently through this pattern of shadows, the features of each bar fluctuated together, allowing the model to learn their individual characteristics. Bars wrapped around toroidally to re-enter the image, thus putting no time limitation on network training.

Figure 3a and c, respectively, shows the time course of network outputs and the final weight matrix when the network was trained for 15 s with this visual stimulus using the competitive learning rule of (5). For comparison, Fig. 3b and d shows the same data using the cooperative learning rule, in which the activation functions $f()$ and $g()$ of (5) are placed in their conventional position in the BSS literature (Cichocki et al. 1997), with the expansive function then $f()$ applying to row elements, and the compressive function $g()$ to column elements.

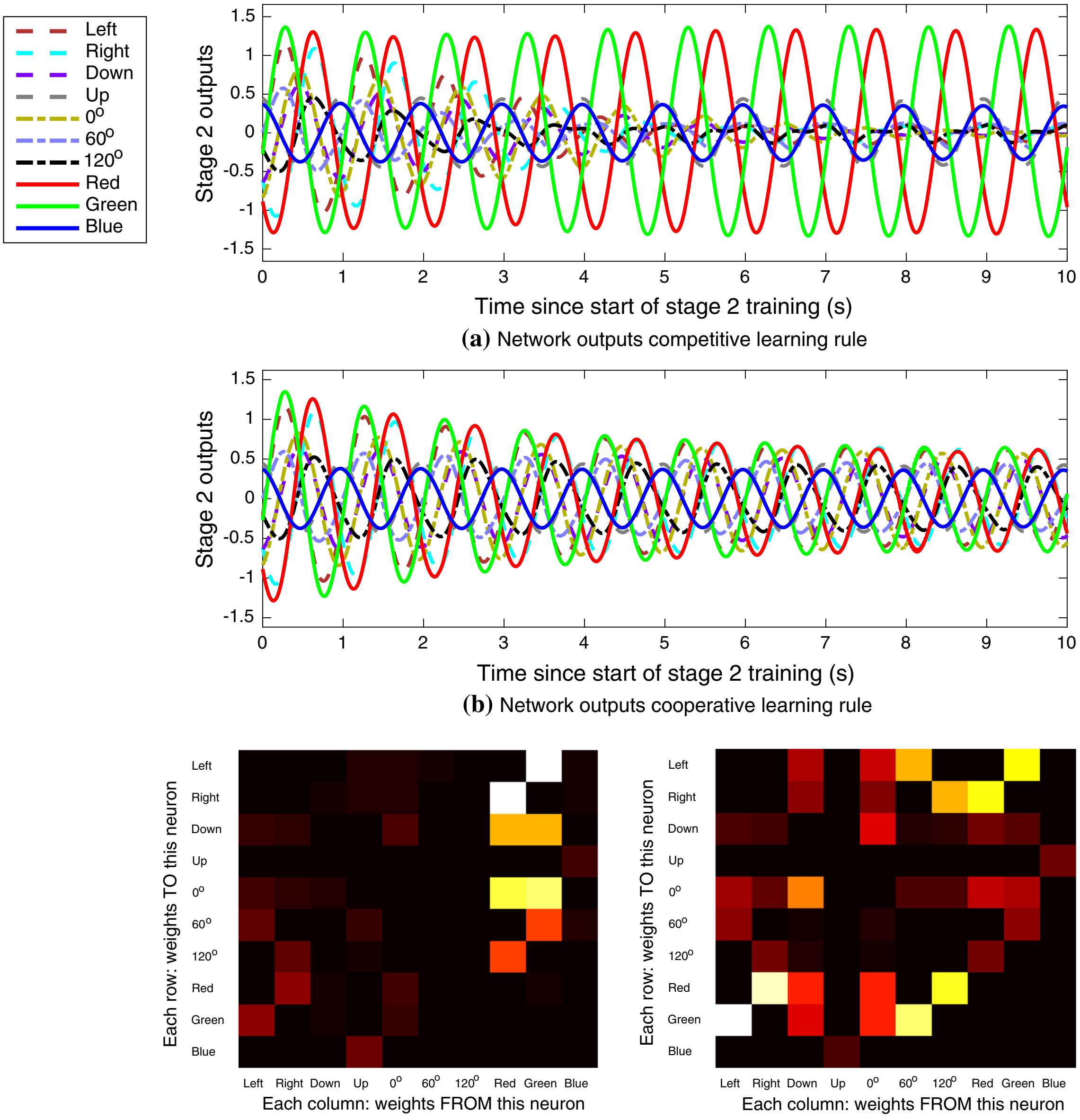
The results of Fig. 3a and c using the competitive learning rule correspond to desired operation of the visual binding network. The red and green output neurons clearly come to dominate all others, and (neglecting very small weights) the columns of the final weight matrix correctly indicate the characteristics of the individual objects which comprised the stimulus. Reading the weights from the "red" column, the bar moved to the right, and down to a lesser extent, and got a roughly equal response from the 0° and 120° orientation filters, indicating an approximate orientation of −30° or equivalently 150°. From the "green" column, the bar moved to the left, downward to a lesser extent, and had an orientation of approximately 30° or equivalently 210°.

In stark contrast, the results shown in Fig. 3b and d reveal clearly that the cooperative learning rule is not applicable to the visual binding model. The reasons for this are detailed in Sect. 5.3.5.

This single example suffices to illustrate the methods of weight matrix interpretation and network functional analysis presented below, but many further experiments are described and full details given in a companion paper (Northcutt et al. 2017).

4 Extracting object-level information

While we have presented and discussed the second-stage network so far as if it learned the features of a static set of objects and then stabilized, the temporal dynamics of learning are in general far more complicated. After initial training of the first stage, the learning of the second stage never stops. This is



(a) Network outputs competitive learning rule

(b) Network outputs cooperative learning rule

(c) Wieghts using competitive learning

(d) Wieghts using cooperative learning

Fig. 3 Outputs and weights of the second-stage network as it trained with a visual stimulus comprised of two *bars* moving through sinusoidal *shadow*. A legend to identify each trace in panels **a** and **b** is shown at upper *left*. **a** Network outputs when using the competitive learning rule of (5), in which the activation functions $f()$ and $g()$ are switched in position relative to conventional learning rules for BSS, and thus emphasize patterns of column over row weights. Note that over the time of training, the *red* and *green* outputs come to inhibit all others. **b** Network outputs when using the conventional cooperative learning rule, which emphasizes patterns of row over column weights. No pattern of outputs is evident apart from nearly uniform inhibition. **c** Final weight matrix $\boldsymbol{T}$ using the competitive learning rule of (5). *Brighter colors* indicate stronger inhibition. The strongest weights are in the *red* and *green* columns, and clearly indicate the features of each *bar*. **d** Final weight matrix $\boldsymbol{T}$ using the cooperative learning rule

desirable because it allows the network to continuously adapt to dynamic visual scenery.

However, should a set of objects in the visual scene persist sufficiently long, the matrix $\boldsymbol{T}$ will converge to a particular set of weights representing the characteristics of the objects, and the number of outputs $\boldsymbol{o}(t)$ that are significantly nonzero will come to correspond to the number of objects.

The convergence of second-stage learning for a given visual stimulus—and thus the validity of what has been learned—may most simply be determined at the current time t by requiring that the sum of all absolute weight matrix changes over a recent period of time τ_s declines below a threshold S_{thr}

$$\int_{t-\tau_s}^{t}\left(\sum_{n=1}^{N}\sum_{k=1}^{N}\left|\frac{\mathrm{d}T_{n,k}}{\mathrm{d}t}\right|\right)\mathrm{d}t < S_{\text{thr}} \tag{6}$$

Once the weight matrix has stabilized, the representation of objects in the image developed by the visual binding network is implicit in the activity of the outputs $\boldsymbol{o}(t)$ and the connection matrix $\boldsymbol{T}$.

4.1 The number of objects and their features

This representation may easily be made more explicit for human interpretation, both making network operation easier to understand and giving practical utility to the model.

The weight matrix $\boldsymbol{T}$ may be simplified by normalizing it to its maximum value over all rows and columns and then removing weights less than a given threshold, which may be expressed as

$$\boldsymbol{T}^{\text{norm}} = \boldsymbol{T}/\max_{n,k=1,N}\left(\boldsymbol{T}_{n,k}\right) \tag{7}$$

$$T^{\text{simp}}_{n,k} = \begin{cases} T^{\text{norm}}_{n,k} & T^{\text{norm}}_{n,k} >= v_{\text{min}} \\ 0 & T^{\text{norm}}_{n,k} < v_{\text{min}} \end{cases} \tag{8}$$

We used a threshold value of $v_{\text{min}} = 0.33$ (1/3 of the maximum weight value). An example of this simplified weight matrix, generated from the raw weight matrix shown in Fig. 3c, is shown in Fig. 4a. Here the column-oriented pattern of weights is even more evident, and the relative strength of each feature may be read out directly from the matrix.

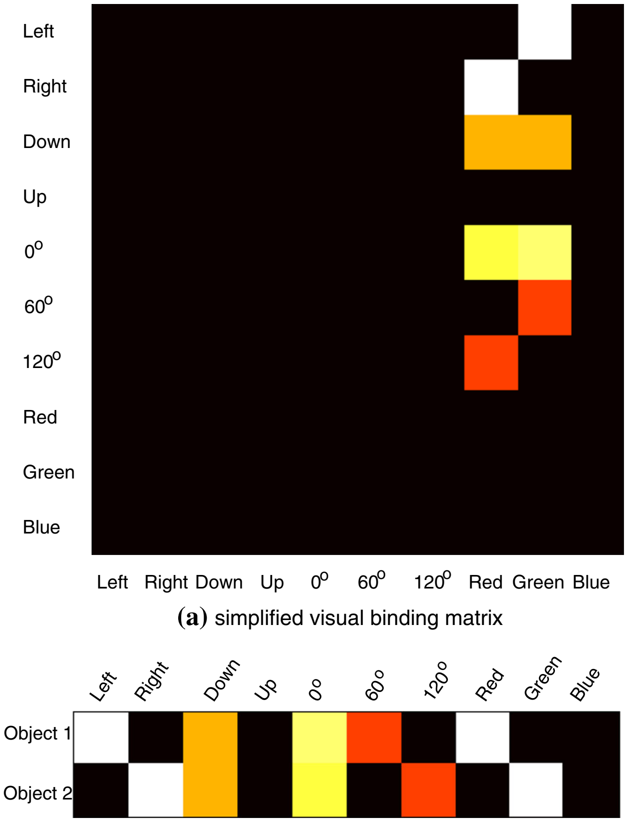
In fact, it is possible to make the representation of objects and their features even more explicit. By summing $\boldsymbol{T}^{\text{simp}}$ vertically, a ten-element row vector $\boldsymbol{t}^{\text{sum}}$ results

$$t^{\text{sum}}_k = \sum_{n=1}^{N} T^{\text{simp}}_{n,k} \qquad k = 1\ldots 10 \tag{9}$$

Each element t^{sum}_k represents the sum of all inhibitory weights to other neurons from any neuron k.

Using this information, we can create an *object matrix* $\boldsymbol{O}$ that explicitly describes the number of objects and their features. Every neuron k for which the entry t^{sum}_k is above a threshold value (for which we used 0.6) is concluded to have accumulated sufficient inhibitory weight to represent an object. For each such neuron k, the weights from $\boldsymbol{T}^{\text{simp}}$ for column k may be used to create a new row i of the object matrix $\boldsymbol{O}$ as

$$O_{i,j} = \begin{cases} T^{\text{simp}}_{j,k} & j \neq k \\ 1 & j = k \end{cases} \qquad j = 1\ldots 10 \tag{10}$$



(a) simplified visual binding matrix

(a) Object matrix extracted from binding matrix

Fig. 4 Extraction of object-level information. **a** Simplified visual binding matrix $\boldsymbol{T}^{\text{simp}}$, from which object features can clearly be read. **b** Feature matrix $\boldsymbol{O}$ extracted from $\boldsymbol{T}^{\text{simp}}$ using (10). Each row of the object matrix corresponds to a unique object. The associated vector $\boldsymbol{r}$ = [8 9] indicates that the first row of the object matrix corresponds to output 8 (*red*) and the second row to output 9 (*green*)

$$r_i = k \tag{11}$$

where the vector $\boldsymbol{r}$ is used to store the index of the output neuron corresponding to each object matrix row i, and the zero diagonal element of $\boldsymbol{T}^{\text{simp}}$ is replaced with unity to represent the fact that this row of weights in the object matrix originated from neuron k (see Sect. 5.5.1 for a theoretical justification). An example of the resulting object matrix $\boldsymbol{O}$ is shown in Fig. 4b: The number of rows of $\boldsymbol{O}$ corresponds to the number of objects, the columns to each individual visual feature, and the weights in each column to the learned strength of each characteristic feature. In this case, the object matrix $\boldsymbol{O}$ accurately represents the motion, orientation, and color of the two objects in the example visual stimulus.

4.2 Elementary visual attention

As can be seen from the example data shown in Fig. 3a, the mutually inhibitory second-stage outputs corresponding to objects in the visual scene fluctuate over time, with each having greater value in proportion to fluctuations of the characteristics of the represented object, measured in terms of the visual features input to the network. This overall measure of feature strength is closely related to computational models of *visual saliency* (Itti and Koch 2001), and so the model might reasonably be said to be switching its *visual attention* (Itti et al. 1998) from one object to another as their relative saliency changes. In fact, visual attention is often modeled as a winner-take-all phenomenon (Lee et al. 1999), which is quite akin to the mutually inhibitory competition of the second-stage network.

The neuron that has the largest output value at any given time corresponds to the most salient object. This movement of this "attentional spotlight" from one object to another can be used to emphasize the currently attended object in the visual image and simultaneously de-emphasize unattended objects. We may synthesize such an "enhanced image" by recombining the feature matrices created in Fig. 2 for every input image using the row of weights from the object matrix $\boldsymbol{O}$ corresponding to the currently winning neuron.

At any given time, let network output $o_k(t)$ currently be the largest. Row i such that $r_i = k$ of the object matrix $\boldsymbol{O}$ represents the visual features of this output. To make the raw feature matrices comparable in magnitude to one another, we make use of the input adaptive group normalization factors already computed and described in Sect. 2.1, which may be composed into a single ten-element vector

$$\boldsymbol{n}(t) = [n_m \; n_m \; n_m \; n_m \; n_o \; n_o \; n_o \; n_c \; n_c \; n_c]$$

We may then combine this vector of normalization factors with the feature weights associated with this object and the neuron output to create a vector $\boldsymbol{f}(t)$ of weights to be applied to each feature matrix

$$f_j(t) = \frac{|o_k(t)| \cdot O_{i,j}}{n_j(t)} \qquad j = 1 \ldots 10 \qquad (12)$$

where the absolute value is used so that large values of the output $o_k(t)$, regardless of sign, result in large contributions to the enhanced image. The vector $\boldsymbol{f}(t)$ may then be used to create a linear recombination of the feature matrices computed from the current input image as shown in Fig. 5, resulting in a normalized RGB mask identifying where salient features exist in the current image. By multiplying this mask with the current input image, an enhanced image is created in which the characteristics associated with the object represented by output k are emphasized while effectively de-emphasizing other objects.

Figure 6 presents a demonstration of the algorithm to enhance the most salient object in input images using our example two-bar visual stimulus, the network outputs and final weight matrix of which are shown in Fig. 3a and c. Comparing the input and enhanced images Fig. 6a and b taken at 6.5 s into training, the red bar is clearly stronger relative to the green in the enhanced image than in the input. To quantify this effect, the ratio of the maximum red value to the maximum green value in the input image is approximately

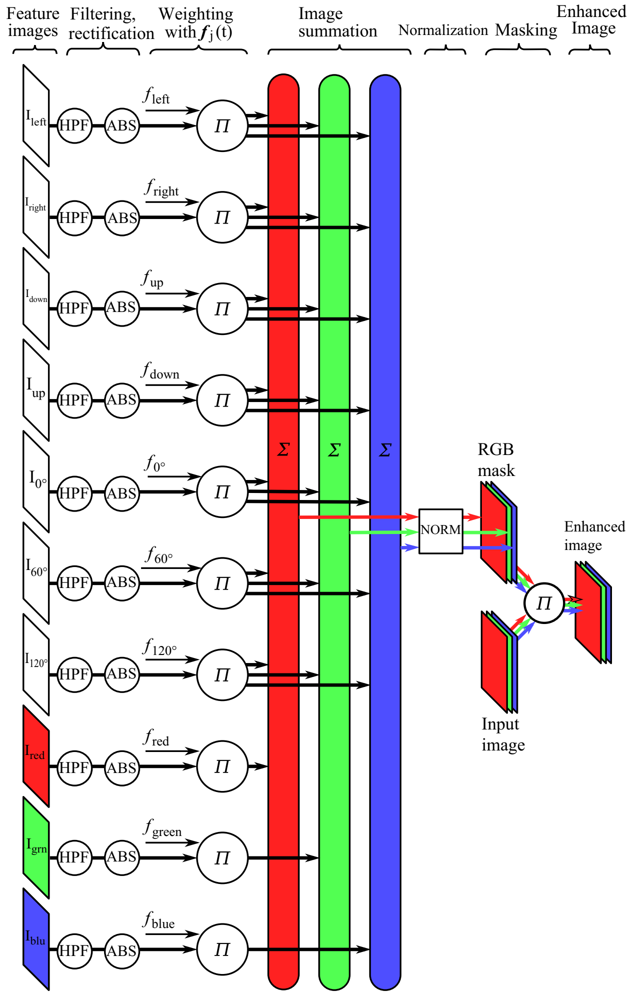
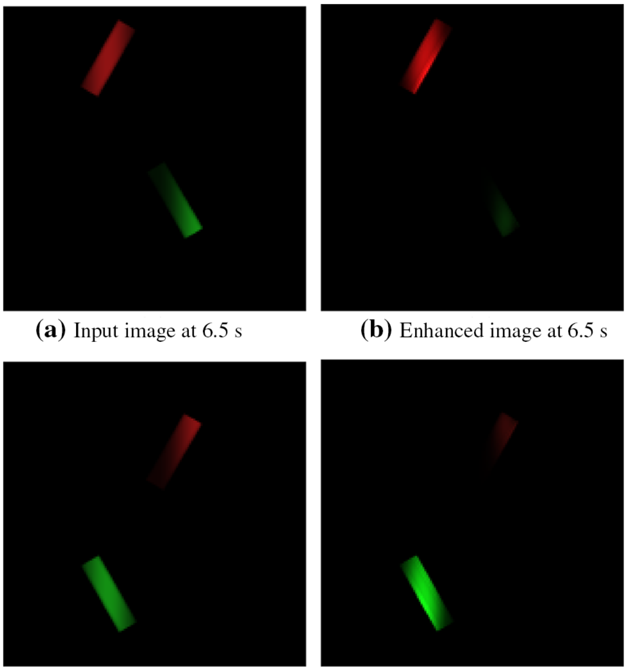


Fig. 5 Computational diagram for creating an enhanced image. *Thin lines* indicate scalar quantities; *thick lines* represent matrices. Starting at figure left, each of the ten 2D feature images created in Fig. 2 is first processed through a first-order high-pass filter (HPF) using the same time constant τ_{HI} as was used for network inputs, and the absolute value of each matrix taken (ABS) so that both increases and decreases in features are represented in the output. Each of the resulting filtered matrices is then multiplied (Π) by the corresponding scalar weight f_j from (12). These weighted feature images are summed point-by-point (Σ) into a single RGB image, after which this image is normalized (NORM) by a scalar value corresponding to its maximum over all pixels and color planes. This results in an RGB "mask" that identifies where salient features of the input image exist. This mask is then multiplied point-by-point (Π) with the input RGB image, resulting in a 2D RGB enhanced image that emphasizes the most salient object. Note that motion and orientation feature images contribute equally to *red*, *green*, and *blue* color planes, but *red*, *green*, and *blue* feature images contribute only to their own color plane


(a) Input image at 6.5 s **(b)** Enhanced image at 6.5 s

(c) Input image at 7.2 s **(d)** Enhanced image at 7.2 s

Fig. 6 Demonstration of enhancement of the most salient object. Each of the four panels shows a 160 × 160-pixel region cropped from the center of a 500 × 500 image, taken at times when the two moving bars passed near one another. **a** The input image at 6.5 s after the beginning of second-stage training, a time at which the red bar was the most salient (refer to *red* and *green* neuron outputs in Fig. 3a). The role of shadowing in saliency is evident here: the *red bar* is strongly visible in the image, almost maximally out of shadow, whereas the *green bar* is almost half shadowed. **b** The enhanced image resulting from the algorithm of Fig. 5, emphasizing the *red bar* while the *green bar* is barely visible. **c** The input image at 7.2 s into second-stage training, a time at which the green object was most salient (refer to Fig. 3a). **b** The resulting enhanced image, emphasizing the *green bar* while the *red bar* is barely visible

unity, but in the enhanced image, the maximum red value is 6.2 times stronger than that of green. In contrast, comparing the input and enhanced images shown in Fig. 6c and d taken at 7.2 s into training, the green bar is enhanced relative to the red, and again red and green have nearly the same maximum value in the input image, but green is 4.1 times stronger in the enhanced image.

Looking more closely at Fig. 6, the effect is not that the characteristics associated with the largest neuron output, representing the most salient object, are greatly increased in the enhanced image; rather, the characteristics of all other objects, even background objects (due to the high-pass filtering of the feature matrices), are weakened. It is notable that this is very reminiscent of attentional effects observed in primate visual cortex (Moran and Desimone 1985): Responses to attended objects are not increased by attention, but rather responses to unattended objects are relatively weakened.

This algorithm provides a bottom-up method for automatically attending to the most salient object in an image sequence over time without needing any prior information about objects or their features.

5 Function and limitations of the model

We have presented one example set of experimental results above and a variety of experiments in a companion paper (Northcutt et al. 2017) practically demonstrating the utility of this two-stage neural network model for sensory refinement, visual binding, internal object-level representation, and even rudimentary visual attention. However, it is impossible to present an exhaustive set of visual stimuli. Under what conditions does this subtly complex neural network model actually perform visual binding, how is this accomplished, and what are its limitations? Intriguingly, the same recurrent neural network subunit is used four times for two distinct purposes in this model.

The first-stage networks (refer to Fig. 1) take as their input feature sensors which may have significant overlap in their particular visual submodality, and the output signals show a significant reduction in overlap of the input sensors: For example, one of the first-stage networks takes inputs which are broadly orientation-tuned and refines them using lateral inhibition into outputs with narrower orientation tuning.

In the first stage, we have applied this same recurrent network to three specific visual submodalities: color, orientation, and motion. Balanced mutual inhibition is a naturally stable state of a fully connected recurrent inhibitory neural network, reached automatically by the learning rule when given an appropriately balanced set of visual stimuli. In fact, our training of the first stage is specifically designed to elicit lateral inhibition.

Lateral inhibition has long been understood to provide refinement of sensory inputs at many levels, its effects having been studied as early as the mid-19th century (Mach 1866), and a number of sensory systems have been proposed to function using this mechanism, including vision (Blakemore and Tobin 1972), audition (Shamma 1985), and olfaction (Linster and Smith 1997).

While sensory refinement via lateral inhibition may well be a useful function in many neural systems, does the first stage contribute anything to the operation of the visual binding network? In fact, we have shown experimentally in a companion paper (Northcutt et al. 2017) that strong mutual inhibition in the first stage is essential to the rapid learning demonstrated in Fig. 3a. Without strong first-stage inhibition, second-stage learning proceeded very slowly, and network weights are very unlikely to ever have reached the stable state shown in Fig. 3c.

While first-stage refinement of selectivity in motion direction, orientation, and color inputs undoubtedly aids the visual binding network in distinguishing objects, the first stage's most essential function for second-stage learning arises from a more subtle effect. The strong inhibition between first-stage outputs creates a tendency for them to become mutually exclusive, much like a weak winner-take-all function. Thus, when an object passes out of shadow and becomes more salient, the first-stage network aids in causing all of its visual attributes to increase *simultaneously* by inhibiting weaker attributes in each visual submodality. Since the learning rule of (5) develops inhibitory weights based on *simultaneous increases or decreases* of visual attributes, the synchronization of visual features from the same object caused by first-stage inhibition greatly speeds the learning of second-stage network weights.

Although the network structure, temporal evolution, and learning rule of the second-stage network (refer to Fig. 1) are the same as the three first-stage networks, the operation of the second stage is much more obscure. The second-stage network learns by associating input signals which are temporally correlated, just as the first-stage networks do. However, as has been proposed for mammalian cortex (Douglas et al. 1989), it is the variety of dissimilar inputs to this network, not its internal structure, that makes its function differ from the first stage. In the present work, these signals are full-field spatial summations of elementary visual features. Based on the results we have shown, the empirical result of this stage's operation is to detect signals which originate from each independently fluctuating object in the input and to quantify how strongly that object expresses each visual feature, thereby providing a solution to the visual binding problem.

5.1 Visual binding as blind source separation

The problem that the second-stage network must solve is really that of *blind source separation*. The canonical example of BSS is typically a linear mixture of auditory sources. Several spatially separated microphones listening to a mixture of spatially distinct independent audio sources can be used to recover the individual audio sources. The use of recurrent neural networks for BSS is a very well studied area, with much of the research stemming from the seminal work of Herault and Jutten (1986).

Mathematically, the problem of BSS can be stated as follows. Consider a column vector of N time-varying sources $\boldsymbol{s}(t)$. These sources are combined in an unknown (but static) linear mixture described by an $N \times N$ *mixing matrix* $\boldsymbol{M}$ to provide a vector of N mixed inputs

$$\boldsymbol{i}(t) = \boldsymbol{M} \cdot \boldsymbol{s}(t) \tag{13}$$

The challenge of BSS, as it is generally considered, is to recover the "hidden" sources $\boldsymbol{s}(t)$ given only the mixed observations $\boldsymbol{i}(t)$. The mixing matrix $\boldsymbol{M}$ is recovered implicitly in this process, but often is of little interest: for example, in the auditory case, $\boldsymbol{M}$ would describe the relative microphone locations, which are usually known. Rather, it is most commonly the hidden source signals $\boldsymbol{s}(t)$ that contain the desired information.

Given the inputs $\boldsymbol{i}(t)$, the fully connected inhibitory neural network used in this paper—as described by the instantaneous update rule of (4), and neglecting the high-pass filter on the inputs—has been shown to converge (given a proper learning rule, and under certain conditions, addressed below) to a state in which the outputs $\boldsymbol{o}(t)$ become scaled versions of the hidden sources $\boldsymbol{s}(t)$ (Jutten and Herault 1991), thus revealing each source separately.

The technique we have developed for processing 2D images into a small number of highly meaningful scalar signals, shown in Fig. 2, allows the problem of visual binding to be reduced to one of BSS. Each of the full-resolution visual feature images is summed into a single time-varying signal which is a linear instantaneous summation of the visual feature contributions of all objects in the scene.

In the case of visual binding—quite opposite to that of auditory BSS—we are not particularly interested in recovering the hidden sources $\boldsymbol{s}(t)$. These "sources" correspond to temporal fluctuations of the visual feature signals from a given object caused, for example, by changes in lighting, movement through occlusions, or changes in distance, and are of little practical value. Rather, we are interested in determining how many independent sources exist in the scene and with which visual features they correspond. This implies that, in visual binding, the mixing matrix $\boldsymbol{M}$ is of primary interest because it reveals the features of objects in the scene, and can be used to enumerate them.

5.2 Assumptions of the visual binding model

For any BSS problem, the number of distinct hidden sources and inputs and the details of their mixture determines whether that problem can be solved. After all, solving the BSS problem is not possible in every case: consider separating many auditory sources given microphones all located at the same spatial location! An analysis of the requirements for the isolated second-stage BSS network to function properly is given in the next section. However, due to the vastly greater complexity of the visual binding problem—which takes a sequence of 2D RGB images rather than a time-varying vector of scalar signals as input—the additional assumptions required for our model to perform properly include the following, all of which we assert are reasonable in virtually any practical situation.

1. *The visual scene is dynamic*. In order for the visual binding model to detect and differentiate objects from the background and other objects, the visual scene must change, as observed in the feature space measured by the inputs to the network. This is in direct contrast to visual BSS models proposed by other authors that are designed for separating mixtures of static images (Jutten and Herault 1991; Guo and Garland 2006). Instead, our model takes as input global spatial summations of local feature detector circuits and relies on changes over time in the scene to produce temporal fluctuations in these signals.

Due to the temporal high-pass filters in the input pathway (3) and in the learning rule (5), there would be no input to the network and no changes to the weight matrix of this system in response to static images, and thus no learning. This assumption implies an inherent interest in *novelty detection*, ignoring static background features and making the network "prefer" objects which are more "active" in the visual scene. The assumption of a dynamic visual scene is hardly restrictive; in fact, this is unavoidable in almost every practical situation.

Specifically, to be useful for binding, the temporal frequency of fluctuations in the features of an object must not be lower than the cutoff frequency set by the high-pass filter time constant τ_{HO} used in the learning rule (5) or they will be filtered out. This time constant, of course, can be adjusted to fit the rate of change of the features of an object of interest.

Related to this assumption is a subtle dependence on spatial image resolution. Since the inputs to our network are scalar global spatial sums, it is not obvious what 2D spatial resolution of input images is required to support visual binding. In the limit of very tiny images, no matter what the visual features are, they will have little or no temporal fluctuation due to the fact that there are not enough image pixels. As image size increases, feature fluctuations will become smoother, and thus, the quality of features provided to the network will increase. However, this quality will saturate as the image resolution becomes sufficient to clearly visualize objects in a given scene. The optimal spatial resolution will be highly dependent on the scene in which objects are to be observed.

2. *Visual features of an object are correlated*. For the visual binding network to function properly, visual features originating from the same object, which then become inputs to the network, must have temporal correlation with each other. This requirement simply means that visual features of any given object measured by the network (in our case, motion, color, and orientation) must vary together over time, as they would in any natural situation.

A wide variety of common scene changes satisfy this assumption. Occlusion of an object by another visual scene element causes a rapid decrease in all local visual features of the occluded object, and therefore a reduction in the wide-field summation of these signals; there is a corresponding increase if the object then reappears. Similarly, when an object passes in and out of shadows, or becomes nearer to or more distant from a light source, the visual measures of the object will vary together in proportion to the resulting brightness fluctuations. Also, as an object draws nearer to the camera, the object size increases in the visual image. Since a closer object stimulates a greater number of local feature detection circuits, its wide-field signals grow in inverse proportion to its distance.

3. *Fluctuations from different objects are distinct*. Herault and Jutten's original work on BSS (Herault and Jutten 1986) inspired many related methods: independent component analysis (Comon 1994), information maximization (Bell and Sejnowski 1995), and mutual information minimization (Yang and Si 1997). The majority of these methods require *statistical independence* for sources to be separated.

However, we observe that while statistically independent sources can be guaranteed to separate, this assumption is not strictly necessary. For example, as we have shown in Fig. 3, two sinusoidal "hidden sources" with the same frequency but different phase can be separated, despite not being statistically independent. However, these sources are sufficiently distinct to be separable.

Implicit in the requirement of statistical independence is that visual feature fluctuations are stochastic. This will generally be true in practical situations, due to the fact that these fluctuations derive from objects moving through shadows, past occlusions, or moving unpredictably with respect to the camera. The assumption that fluctuations from different objects will be statistically independent is also reasonable in practical cases: visual features from different objects will be independent simply because they result from independent physical processes in the world.

However, the network does not simply fail if this assumption is not satisfied. If features of two objects are not statistically independent, nor even distinct, the visual binding network will bind these signals together to represent a single object, which in perspective is not an unreasonable conclusion given a set of visual features that are highly correlated with one another.

4. *Object features are persistent.* If a given visual feature (the color of an object, for example) is to be used in binding, an object must retain that feature over a period of time sufficient for the network to learn a pattern of weights based upon it. While the network relies on temporal fluctuations of these features for learning, we must assume that at least some subset of an object's features (chosen from the color, motion, and orientation submodalities for the present network) are relatively persistent.

For example, the motion feature inputs from a car driving rapidly in a circle, and thus constantly changing direction, would not be useful in solving the visual binding problem, although its color would likely remain constant as it turned, and would be useful in binding.

The upper limit on the speed that visual features of an object may change is set by the learning rate γ in the learning rule of (5). Features of a given object that appear and disappear so quickly that the integration of their effect over time never creates a significant change in the weight matrix $\boldsymbol{T}$ will not affect the output. Within the bounds of numerical stability, the learning rate can be adjusted to fit the time course of feature changes for a given visual stimulus.

5. *Measured visual features must be diverse*. In order to separate out a variety of different objects, the visual features measured by the model must be diverse, and span a range of visual submodalities. Preferably, the feature inputs should span the full space of interest in each visual submodality. An example of this diversity is the feature set we have shown in Fig. 2, fully spanning motion, color, and orientation.

If the visual features measured are not sufficient to separate the objects of interest (for an extreme example, imagine that all ten feature detectors were only sensitive to the amount of red color in the image), in general, visual binding will not be possible.

5.3 Analysis of a two-neuron BSS network

Given the assumptions of the previous section, we can reduce the visual binding problem solved by the model as a whole to one of blind source separation in the second stage. The network output and inhibitory weight temporal dynamics of the second-stage network that we have used for visual binding in the model are surprisingly complex, once equipped with the time-stepping neuron update rule of (3) and the learning rule of (5), both of which are potentially unstable. Outside the context of visual binding, the generalized analysis of an N-neuron BSS network with $N(N-1)$ inhibitory weights is highly formalized and unrevealing and has already been well addressed in the literature (Jutten and Herault 1991; Sorouchyari 1991; Joho et al. 2000).

However, the second stage is sufficiently complex that even a two-neuron network with only two inhibitory weights can generate extremely unexpected results, and an analysis of this minimal network better serves to clarify how the second stage of our model works, and how our novel changes to the network update and learning rule affect the model's performance. For this reason, we base our theoretical analysis of the second stage around a two-neuron network for which all solutions can be simply enumerated and generalize our results to the full network wherever possible. We follow with demonstrations of the function of this two-neuron network that can be clearly understood in terms of the analysis shown, and make conclusions about the full ten-neuron second-stage network thereafter.

Our analysis begins by enumerating the possible "correct" solutions to the BSS problem for a two-neuron network without any learning using the instantaneous network update rule of (4), which for very small simulation time steps approximates the far less analytically tractable time-stepping update rule actually used in the model. For simplicity, in this section we assume that all sources (and thus inputs) have temporal frequencies sufficiently higher than the cutoff frequency of the input high-pass filter so that $\boldsymbol{i}'(t) = \boldsymbol{i}(t)$.

5.3.1 The typical case

The "typical case" in BSS is one in which the number of distinct, nonzero hidden sources is the same as the number of network inputs and outputs. In this case, as will be shown, there is no single correct solution but rather a family of closely related solutions.

For the two-neuron network, the set of scalar equations described by (13) is

$$i_1(t) = M_{1,1} \cdot s_1(t) + M_{1,2} \cdot s_2(t) \tag{14}$$

$$i_2(t) = M_{2,1} \cdot s_1(t) + M_{2,2} \cdot s_2(t) \tag{15}$$

The temporal evolution equation of (4) becomes

$$o_1(t) = i_1(t) - T_{1,2} \cdot o_2(t) \tag{16}$$

$$o_2(t) = i_2(t) - T_{2,1} \cdot o_1(t) \tag{17}$$

into which we can substitute (14) and (15) to get

$$o_1(t) = M_{1,1} \cdot s_1(t) + M_{1,2} \cdot s_2(t) - T_{1,2} \cdot o_2(t) \tag{18}$$

$$o_2(t) = M_{2,1} \cdot s_1(t) + M_{2,2} \cdot s_2(t) - T_{2,1} \cdot o_1(t) \tag{19}$$

For the blind source separation problem to be solved, the outputs $\boldsymbol{o}(t)$ must become equal to scaled versions of the hidden inputs $\boldsymbol{s}(t)$ (Jutten and Herault 1991). As long as the weight matrix $\boldsymbol{M}$ is nonsingular, there are exactly two possible solutions that can be learned in the inhibitory connection matrix $\boldsymbol{T}$.

As previous authors have shown (and may be easily derived from the equations above), the first situation in which correct source separation occurs is when source indices are not permuted with respect to the outputs

$$o_1(t) = M_{1,1} \cdot s_1(t) \tag{20}$$

$$o_2(t) = M_{2,2} \cdot s_2(t) \tag{21}$$

which results in the connection matrix

$$\boldsymbol{T} = \begin{bmatrix} 0 & \frac{M_{1,2}}{M_{2,2}} \\ \frac{M_{2,1}}{M_{1,1}} & 0 \end{bmatrix} \tag{22}$$

The only other possible solution occurs when the source indices have exchanged with respect to the outputs

$$o_1(t) = M_{1,2} \cdot s_2(t) \tag{23}$$

$$o_2(t) = M_{2,1} \cdot s_1(t) \tag{24}$$

and results in the connection matrix

$$\boldsymbol{T} = \begin{bmatrix} 0 & \frac{M_{1,1}}{M_{2,1}} \\ \frac{M_{2,2}}{M_{1,2}} & 0 \end{bmatrix} \tag{25}$$

In general, for an N-element BSS network in this case, there are $N!$ possible solutions differing only in the permutation of the outputs $\boldsymbol{o}(t)$ relative to the hidden sources $\boldsymbol{s}(t)$. Note that, even for our modest ten-neuron second-stage network, this still allows for 3,628,800 possible solutions!

5.3.2 The overdetermined case

A more unusual situation in the BSS literature is the "overdetermined case" (Joho et al. 2000), in which less than N distinct sources are mixed to provide inputs to an N-element network. There are far fewer solutions in this case, and they differ from the typical case.

In our simplified two-neuron example, when $s_2(t) = 0$ (or equivalently $M_{1,2} = M_{2,2} = 0$) while $s_1(t)$ is nonzero, achieving the nonpermuted solution of (20) and (21) can be accomplished by setting $o_2(t) = 0$. From (19), this requires

$$T_{2,1} \cdot o_1(t) = M_{2,1} \cdot s_1(t) + M_{2,2} \cdot s_2(t) \tag{26}$$

and given that $s_2(t) = 0$ and that our goal is to make $o_1(t) = M_{1,1} \cdot s_t(t)$, $T_{2,1}$ can be written directly as

$$T_{2,1} = \frac{M_{2,1}}{M_{1,1}} \tag{27}$$

However, this leaves us with no constraint on $T_{1,2}$, since in the network temporal evolution rule it multiplies a signal that is zero. Therefore, our only requirement is that $T_{1,2} \geq 0$ to prevent unintentional excitation, and the resulting connection matrix is

$$\boldsymbol{T} = \begin{bmatrix} 0 & T_{1,2} \\ \frac{M_{2,1}}{M_{1,1}} & 0 \end{bmatrix} \tag{28}$$

where $T_{1,2} \geq 0$ (we will be able to place further constraints on $T_{1,2}$ in the next section). Another possible solution with this same source condition is to permute the sources with respect to the outputs as in (23) and (24) which results in connection matrix

$$\boldsymbol{T} = \begin{bmatrix} 0 & \frac{M_{1,1}}{M_{2,1}} \\ T_{2,1} & 0 \end{bmatrix} \tag{29}$$

where $T_{2,1} \geq 0$ is largely unconstrained.

By symmetry, if instead the sources switched and $s_1(t) = 0$ (or equivalently $M_{1,1} = M_{2,1} = 0$) while $s_2(t)$ were nonzero, the required connection matrix for the nonpermuted case would be

$$\boldsymbol{T} = \begin{bmatrix} 0 & \frac{M_{1,2}}{M_{2,2}} \\ T_{2,1} & 0 \end{bmatrix} \tag{30}$$

where $T_{2,1} \geq 0$, and for the permuted case

$$\boldsymbol{T} = \begin{bmatrix} 0 & T_{1,2} \\ \frac{M_{2,2}}{M_{1,2}} & 0 \end{bmatrix} \tag{31}$$

where $T_{1,2} \geq 0$.

In general, ignoring the indeterminate (and irrelevant) matrix values, for an N-element BSS network in the overdetermined case with only $m < N$ nonzero sources, there are $\left(\frac{N!}{(N-m)!}\right)^2$ possible solutions, corresponding to all possible permutations of the m nonzero outputs with any given pattern of m sources, and all permutations of those m sources with respect to the inputs.

While still not trivially small, this is vastly fewer solutions than for the typical case by a factor of $\frac{(N-m)!^2}{N!}$. For our ten-neuron second-stage network, if only two distinct sources are presented, the number of possible solutions is reduced to a mere 8100.

5.3.3 Stability of network temporal evolution

The values of the connection matrix $\boldsymbol{T}$ given above represent solutions to the BSS problem in a two-neuron network, but were derived using an approximate temporal evolution rule. Will a recurrent network with this connection matrix using the time-stepping update rule of (3) be stable? We have earlier addressed conditions for stability of this temporal evolution rule and concluded that stability simply requires that all eigenvalues of the connection matrix $\boldsymbol{T}$ have magnitude less than unity.

For the specific values of $\boldsymbol{T}$ given in the "typical case" above, it is possible to compute the eigenvalues directly from the characteristic polynomial, and thus the conditions for stability of the connection matrices of (22) and (25) can be shown, respectively, to be

$$M_{1,2} \cdot M_{2,1} < M_{1,1} \cdot M_{2,2} \tag{32}$$

$$M_{1,2} \cdot M_{2,1} > M_{1,1} \cdot M_{2,2} \tag{33}$$

These two conditions are mutually exclusive, meaning that for any given nonsingular mixing matrix $\boldsymbol{M}$, only one of the two connection matrices given in the typical case for the two-neuron network can be temporally stable.

The computation of eigenvalues may also be carried out for the connection matrices derived for the overdetermined case. For each matrix in (28) through (31), this computation results, respectively, in conditions for stability

$$\frac{M_{1,1}}{M_{2,1}} > T_{1,2} \geq 0 \tag{34}$$

$$\frac{M_{2,1}}{M_{1,1}} > T_{1,2} \geq 0 \tag{35}$$

$$\frac{M_{2,2}}{M_{1,2}} > T_{2,1} \geq 0 \tag{36}$$

$$\frac{M_{1,2}}{M_{2,2}} > T_{1,2} \geq 0 \tag{37}$$

Under its respective condition, each of the matrices derived for the overdetermined case is a stable state of the network.

Thus, only in the typical case, the requirement for stability of the time evolution rule eliminates half of the potential solutions for the two-neuron network, and in the general N-neuron situation a potentially large number of the $N!$ theoretically possible connection matrices. In the overdetermined case, where less solutions exist already, the requirement for stability places an upper bound on the unconstrained weight, but eliminates no potential solutions.

This reduction in the number of valid solutions to the BSS problem could be a crucially important consequence of using the time-stepping network evolution rule of (3) rather than the conventional approximation of (4).

5.3.4 Stability of the network learning rule

Assuming a given connection matrix $\boldsymbol{T}$ does represent a solution to the BSS problem and has eigenvalue magnitudes less than unity, thus allowing network temporal evolution to be stable, do these solutions represent stable states of the learning rule of (5)? If not, they could never be learned by our neural network model.

For a given inhibitory connection matrix $\boldsymbol{T}$ to be a stable state of the learning rule, updates to $\boldsymbol{T}$ made by the learning rule of (5) must be zero-mean over time. This can be expressed as

$$E\left[\frac{\mathrm{d}T_{1,2}}{\mathrm{d}t}\right] = E\left[\gamma \cdot g\big(o_1'(t)\big) \cdot f\big(o'_2(t)\big)\right] = 0 \tag{38}$$

$$E\left[\frac{\mathrm{d}T_{2,1}}{\mathrm{d}t}\right] = E\left[\gamma \cdot g\big(o'_2(t)\big) \cdot f\big(o_1'(t)\big)\right] = 0 \tag{39}$$

where $E[x]$ formally represents the expected value of random variable x, but may also be interpreted for deterministic time variables as the average value of $x(t)$ over some fixed period of time. The fundamental BSS assumption that hidden sources are statistically independent is a requirement for stability of the learning rule, but as discussed previously, the rule may also stabilize for distinctly varying deterministic sources.

Assuming that the network does indeed solve the BSS problem, resulting in one of the connection matrices shown above, we may substitute in high-pass filtered versions of (20) and (21) (both of which are satisfied in all typical and overdetermined cases presented above) and factoring out the constant γ, (38) and (39) become

$$E\left[g\big(M_{1,1} \cdot s_1'(t)\big) \cdot f\big(M_{2,2} \cdot s_2'(t)\big)\right] = 0 \tag{40}$$

$$E\left[g\big(M_{2,2} \cdot s_2'(t)\big) \cdot f\big(M_{1,1} \cdot s_1'(t)\big)\right] = 0 \tag{41}$$

The constants from the mixing matrix $\boldsymbol{M}$ in these equations may be factored out without changing the required conditions to make the equations true, so let us look more closely at the effect of the expected value operator on the high-pass filtered sources and the learning rule activation functions $f(x)$ and $g(x)$.

Let $x_1(t) = s_1'(t)$ and $x_2(t) = s_2'(t)$. The function of the high-pass filter is to remove low frequencies (the mean, or expected value, being the lowest possible frequency), so due to the operation of the filter

$$E\left[x_1(t)\right] = E\left[x_2(t)\right] = 0 \tag{42}$$

Thus, both $x_1(t)$ and $x_2(t)$ are zero-mean random variables due to the action of the high-pass filter and retain the statistical independence of the sources $s_1(t)$ and $s_2(t)$ after the linear filtering operation.

In our learning rule, the nonlinear functions $g(x) = \tanh(\pi x)$ and $f(x) = x^3$ are applied, respectively, to the statistically independent zero-mean random variables $x_1(t)$ and $x_2(t)$. The conditions for stability of learning rules of this form have been studied in detail by Sorouchyari (1991). By approximating the hyperbolic tangent function with a Taylor series to the third-order term—a good approximation for our model's normalized inputs—it can be shown that this stability is conditioned on the third moment about zero (the *skewness*, a measure of the asymmetry of the probability density function) of one or both of the random variables $x_1(t)$ and $x_2(t)$ being zero. This directly requires that, for stability of the learning rule, the skewness of one or both of the hidden sources $s_1(t)$ and $s_2(t)$ must be zero.

Note that, unlike network temporal stability, this condition for stability of the learning rule requires only that the connection matrices $\boldsymbol{T}$ solve the visual binding problem. The conditions for learning stability are primarily placed on the statistics of the hidden sources $\boldsymbol{s}(t)$.

For distinctly varying deterministic sources, this requirement would imply that the mean value over some period of time of each cubed hidden source signal $s^3(t)$ be zero, which is true of many functions including the sinusoidal sources used in the experiment of Fig. 3 and in the examples given below.

In the more general case of visual binding, these hidden sources represent temporal fluctuations of visual features caused by unpredictable changes in lighting, occlusion, distance, or similar effects. Small or zero skewness of these visual fluctuations is extremely plausible, and thus in this case stability of the learning rule is very likely as well.

5.3.5 Role of the activation functions in learning

If linear weighting functions $g(x) = x$ and $f(x) = x$ were used in the learning rule, changes to diagonal elements in the weight matrix $\mathrm{d}T_{n,k}/\mathrm{d}t$ and $\mathrm{d}T_{k,n}/\mathrm{d}t$ would necessarily be equal, resulting in equal values of $T_{n,k}$ and $T_{k,n}$ and thus a purely Hebbian diagonally symmetric connection matrix $\boldsymbol{T}$.

Based on the BSS theory presented above, this learning rule could never solve the overdetermined case, in which all correct weight matrices are asymmetric, and could only solve the typical BSS problem if the ratio of weights in the required weight matrix of (22) and (25) was symmetric, which in both cases requires the ratios $M_{1,2}/M_{2,2}$ and $M_{2,1}/M_{1,1}$ to be the same. This implies that, using linear weighting functions, the network could only learn to solve the problem for a very limited set of cases. It is for this reason that nonlinearities have long been used in BSS learning rules.

In the conventional BSS learning rule (Jutten and Herault 1991; Cichocki et al. 1997), an odd expansive nonlinearity $f(x)$ is applied to row elements of the weight matrix and an odd compressive nonlinearity $g(x)$ to column elements. The fact that the expansive function is applied to row elements, which describe the inhibition *from* all other neurons *to* a given neuron, results in development of a pattern of row-oriented weights. In this case, each neuron learns how to subtract from its own output a scaled version of every other neuron's output in order to compensate for the effects of the mixing matrix. Thus, neurons *cooperate* to "de-mix" the inputs, thereby revealing the hidden sources. This *cooperative learning rule* works well in the typical BSS case for which it was developed. However, this learning rule makes it difficult for any neuron to develop sufficient inhibition to completely suppress the output of another neuron, making it less appropriate for the overdetermined case in which some outputs should ideally be completely inhibited.

If instead the position of the nonlinearities is exchanged in the learning rule, such that the expansive nonlinearity $f(x)$ applies to column elements and the compressive nonlinearity $g(x)$ to row elements, the resulting weight matrix is column-oriented, thus focusing learning on the weights *to* all other neurons *from* a given neuron. In this case, each neuron must *compete* with other neurons to grow its inhibitory weights in order to avoid being suppressed. This *competitive learning rule* is well suited to the overdetermined case, since it encourages suppression of weak outputs. The learning rule of (5) used for our visual binding network is of this competitive type, which is appropriate because the number of distinct objects in any visual scene is unknown, but in general is less than the number of neurons in the second-stage network, and thus presents an overdetermined BSS problem, for which the competitive learning rule is well suited.

Using the two-neuron case as an example, we compare the difference in results when using these two learning rules in the next section.

5.4 Examples for the two-neuron BSS network

To understand the operation of the isolated second-stage network and the relative usefulness of the competitive and cooperative learning rules, in this section we show a set of simulation results for a two-neuron network with mixed sinusoidal inputs.

For all experiments, the simulation time step was set to a very small value of 1 ms to avoid any simulation artifacts. The time-stepping temporal update rule of (3) was used. The learning rate was $\gamma = 5$, and the high-pass filter time constant used on outputs in the learning rule was $\tau_{HO} = 2$ s and thus had little effect since all sources and inputs were zero-mean and at significantly higher frequencies than the HPF cutoff frequency. The hidden sources were

$$s_1(t) = \sin(2\pi f_1 t) \tag{43}$$

$$s_2(t) = \sin(2\pi f_2 t) \tag{44}$$

where $f_1 = 2$ and $f_2 = 1$ Hz.

We wish to compare the effect of the two learning rules in the overdetermined case, and so the mixing matrix was

$$\boldsymbol{M} = \begin{bmatrix} 0 & 0.7 \\ 0 & 0.6 \end{bmatrix} \tag{45}$$

which effectively set $s_1(t) = 0$. Using (13), this led to nearly identical network inputs

$$i_1(t) = 0.7 \cdot s_2(t) \tag{46}$$

$$i_2(t) = 0.6 \cdot s_2(t) \tag{47}$$

and made for what appeared to be a very difficult overdetermined BSS problem. According to our earlier derivations, this problem would be solved either by the weight matrix of (30) or (31) depending on the permutation of the outputs, but in either case temporal stability places limits on the connection matrix values:

$$T_{1,2} <= M_{1,2}/M_{2,2} = 1.17 \tag{48}$$

$$T_{2,1} <= M_{2,2}/M_{1,2} = 0.86 \tag{49}$$

where only one equality would be allowed, depending on the permutation.

To contrast the effects of the two learning rules, for this simulation we did not require that the connection matrix $\boldsymbol{T}$ has eigenvalues less than unity as we did while training our second-stage visual binding network, and thus networks were allowed to become temporally unstable.

The results of this experiment are shown in Fig. 7. We start by using the conventional cooperative learning rule, which is ill suited for the overdetermined case. Figure 7a, b and c, respectively, shows the time course of the outputs, the connection weights, and the eigenvalues of the connection matrix as the network learned over a 60-s period. During learning, the weights and outputs went through several distinctly different phases.

For about the first 9 s, network weights increased together and each output increasingly inhibited the other. This continued until the connection matrix eigenvalues exceeded unity, indicating a temporally unstable network. A rapid transient caused by this instability led to a sudden decrease in the weight $T_{1,2}$, causing output $o_2(t)$ to become dominant and output $o_1(t)$ to become moderately suppressed. Due to the residual temporal correlation of the two outputs, weight $T_{1,2}$ slowly increased until approximately 25 s, at which time the network became temporally unstable a second time, and again fell back into the same pattern of weights. As before, continued training again increased $T_{1,2}$ until at approximately 45 s the network became temporally unstable for a third time. In this case, the sharp oscillations due to instability led to a "flipping" of network weights in which $T_{1,2}$ remained approximately constant and $T_{2,1}$ began to consistently decrease. This change also flipped the dominant output, so that over the remainder of the simulation output $o_1(t)$ was dominant while $o_2(t)$ was increasingly suppressed. Eigenvalues of the connection matrix declined markedly after this last state change, and the network never again became unstable.

Figure 7d shows the trajectory of the two connection weights plotted against one another over a 90-s period (longer than shown in the previous three panels), with the green lines indicating the maximum weights allowed for temporally stable operation, given above. The network's repeated instability is evident, as is its inevitable progression as $T_{2,1}$ declines to a final weight matrix of

$$\boldsymbol{T} = \begin{bmatrix} 0 & 0 \\ 0.86 & 0 \end{bmatrix} \tag{50}$$

which is an instance of the solution to the overdetermined BSS problem given in (31). Thus, when allowed to become temporally unstable, the cooperative learning rule was finally able to solve the overdetermined BSS problem.

In contrast, Fig. 7e shows the trajectory of learning in weight space using the competitive learning rule, which is well suited to this overdetermined problem. Connection weights progressed smoothly within 5 s to a final value of

$$\boldsymbol{T} = \begin{bmatrix} 0 & 0.2 \\ 0.86 & 0 \end{bmatrix} \tag{51}$$

which is another instance of the solution to the overdetermined BSS problem given in (31). Figure 7f shows the time course of network outputs using the competitive learning rule. Without ever becoming unstable, network outputs rapidly converged so that $o_1(t)$ dominated and $o_2(t)$ was almost completely inhibited. This permutation was completely predictable, given that the initial input $i_1(t)$ was larger than $i_2(t)$. Using this learning rule, the value of the indeterminate weight $T_{1,2}$ is dependent on the ratio of the two nonzero mixing matrix weights: the larger this ratio, the smaller the final value of $T_{1,2}$.

It is clear that the competitive learning rule has an easier time solving this problem. In fact, these results are directly related to the full visual binding network outputs shown in Fig. 3. Figure 3a shows the ten-unit second-stage network using the competitive learning rule as it rapidly converges to the correct solution with only two distinct sinusoidal inputs, an overdetermined case for that larger network.

Further, note the similarity between the network outputs of Fig. 3b using the ill-suited cooperative learning rule and the first 9 s of Fig.7a: In both cases, roughly mutual inhibition was developed, which did not progress the network toward a solution to the problem. However, unlike the simulation of Fig. 7, the visual binding network using cooperative learning was not allowed to become temporally unstable, and so never arrived at a correct solution.

It must also be mentioned that in the typical case the cooperative learning rule works very well, but the competitive rule does not converge to a proper solution. With all other parameters unchanged, if we use a mixing matrix of

$$\boldsymbol{M} = \begin{bmatrix} 0.6 & 0.7 \\ 0.7 & 0.6 \end{bmatrix} \tag{52}$$

then we expect a connection matrix described by (25)

$$\boldsymbol{T} = \begin{bmatrix} 0 & 0.86 \\ 0.86 & 0 \end{bmatrix} \tag{53}$$

a solution to which the cooperative learning rule rapidly converges. Competitive learning fails to solve the problem in this case, instead ending up almost completely suppressing $o_2(t)$ due to its slightly weaker input.

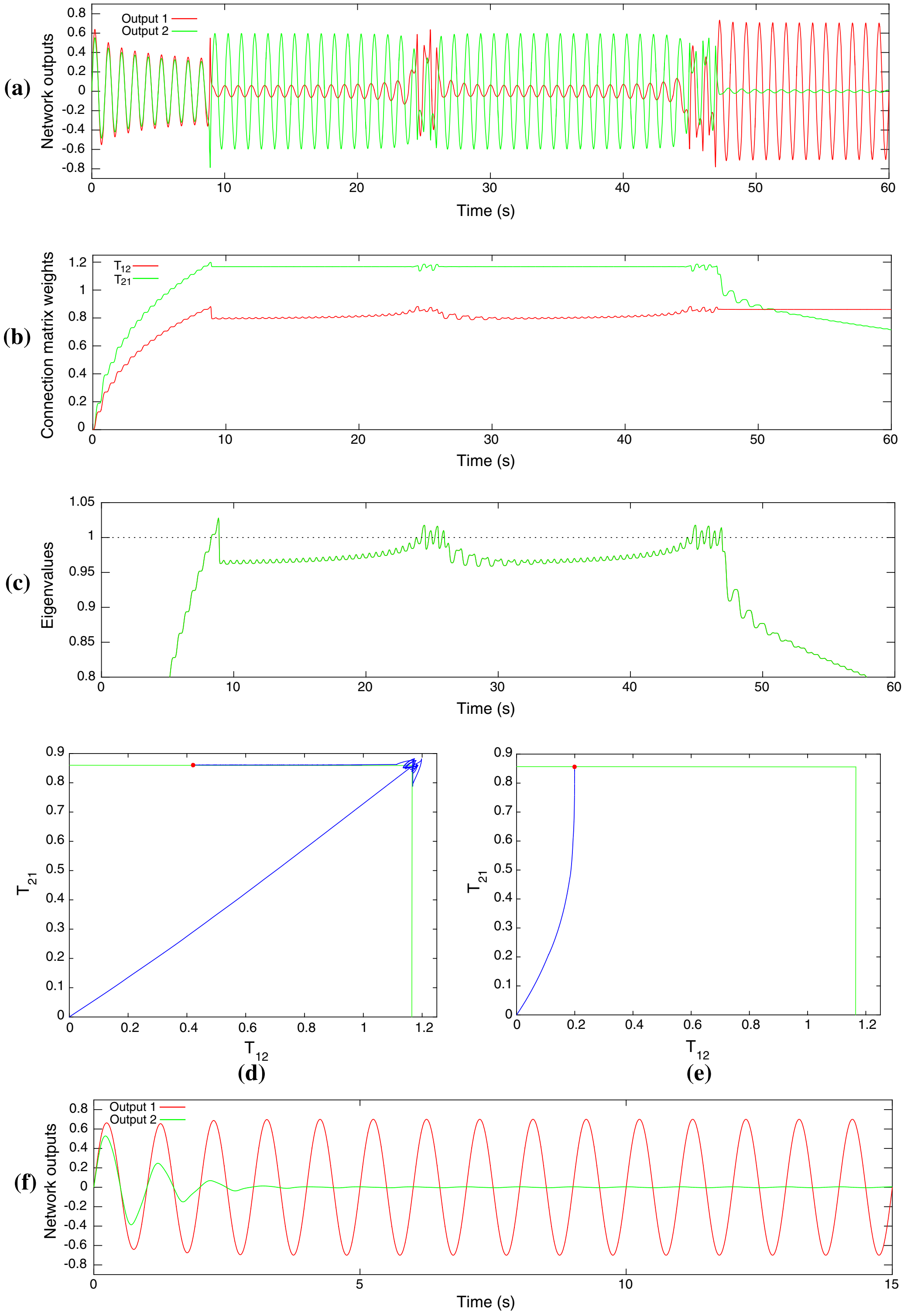


Fig. 7 Comparison of cooperative and competitive learning rules for second-stage blind source separation with only one independent source, using a two-neuron network. **a** Network outputs over a 60-s period using the cooperative learning rule. The network goes through a number of transient states before reaching a stable state. **b** Corresponding network weights using the cooperative learning rule. **c** Corresponding eigenvalues of the 2 × 2 connection matrix $\boldsymbol{T}$ developed using the cooperative learning rule. Note that the eigenvalues exceed unity (*dotted line*) several times, indicating that the network has become temporally unstable. **d** Trajectory of network in weight space using the cooperative learning rule. The *green lines* indicate the theoretical limits on each of the two weights. The red circle indicates the weights of the network after 90 s of training. **e** Trajectory of network in weight space using the competitive learning rule. The *red circle* indicates the weights of the network after 15 s of training, although this value was stable at less than 5 s. **f** Network outputs using the competitive learning rule. Comparing to panel a, note the much shorter timescale

5.5 The full visual binding network

In the context of visual binding, the hidden sources $\boldsymbol{s}(t)$ in (13) correspond to temporal fluctuations of measured features produced by objects in the visual scene. In a successful solution to the visual binding problem, the outputs $\boldsymbol{o}(t)$ are meant to provide estimates of these fluctuations. Based on a comparison of the full-network results shown in Fig. 3 and the isolated second-stage simulations of Fig. 7, it is clear that the second-stage network does indeed solve an overdetermined BSS problem to reach a solution to the visual binding problem. How do the results we have derived for the two-neuron second-stage network generalize to the full visual binding network?

5.5.1 Interpretation of the connection matrix

Based on decades of previous research in BSS (Herault and Jutten 1986; Jutten and Herault 1991; Cichocki et al. 1997; Joho et al. 2000), we expect that if the network reaches a successful solution to the visual binding problem, the result will be that the outputs $\boldsymbol{o}(t)$ will become scaled, permuted versions of the hidden sources $\boldsymbol{s}(t)$.

Specifically, assuming the second-stage network reaches a stable state and solves the BSS problem, we can expect

$$\boldsymbol{o}(t) = \boldsymbol{G} \cdot \boldsymbol{P} \cdot \boldsymbol{s}(t) \tag{54}$$

where $\boldsymbol{G}$ is a diagonal matrix with N elements (any of which may be zero, indicating an overdetermined case) that provides a scale factor on each individual output, and $\boldsymbol{P}$ is a permutation matrix (an identity matrix with the rows rearranged) that reorders the outputs $\boldsymbol{o}(t)$ with respect to the sources $\boldsymbol{s}(t)$.

Inverting (13) and substituting it into (54), we can write

$$\boldsymbol{o}(t) = \boldsymbol{G} \cdot \boldsymbol{P} \cdot \boldsymbol{D} \cdot \boldsymbol{i}(t) \tag{55}$$

where $\boldsymbol{D}$ is a *de-mixing* matrix. Specifically, $\boldsymbol{D}$ is the inverse of the mixing matrix

$$\boldsymbol{D} = \boldsymbol{M}^{-1} \tag{56}$$

which exists by definition in any solvable typical case of BSS. For the overdetermined case, we may insert nonzero values into any columns of the mixing matrix that multiply zero-valued sources to make this inversion possible without changing the equation.

Making use of the approximate network evolution rule of (4), and assuming for simplicity that the input high-pass filter has negligible effect on $\boldsymbol{i}(t)$, we can rearrange terms to write

$$\boldsymbol{o}(t) = [\boldsymbol{I} + \boldsymbol{T}]^{-1} \cdot \boldsymbol{i}(t) \tag{57}$$

where $\boldsymbol{I}$ is the identity matrix. Given that the two linear Eqs. 55 and 57 both describe how to compute $\boldsymbol{o}(t)$ from $\boldsymbol{i}(t)$, it must be true that

$$[\boldsymbol{I} + \boldsymbol{T}]^{-1} = \boldsymbol{G} \cdot \boldsymbol{P} \cdot \boldsymbol{D} \tag{58}$$

Rearranging terms and inverting $\boldsymbol{D}$ to recover $\boldsymbol{M}$, an expression for the mixing matrix can be written as

$$\boldsymbol{M} = \boldsymbol{G} \cdot \boldsymbol{P} \cdot [\boldsymbol{I} + \boldsymbol{T}] \tag{59}$$

This result shows that the mixing coefficients in $\boldsymbol{M}$, aside from an unknown scale and permutation, can be read out directly from the learned connection matrix $\boldsymbol{T}$. (59) shows theoretically what we have already shown experimentally in Fig. 4: The learned weight matrix $\boldsymbol{T}$ contains the crucial information about visual binding: specifically, the mix of features associated with each distinct object.

In the overdetermined case, some of the hidden sources $\boldsymbol{s}(t)$ are effectively zero. In this case, the columns of the mixing matrix of (59) that correspond to nonexistent sources (and thus strongly inhibited outputs) may simply be removed, resulting in a matrix of weights corresponding only to objects that exist. In fact, this is exactly the procedure we have described in (10) and demonstrated in Fig. 4: Columns of $\boldsymbol{T}$ corresponding to active outputs are extracted after placing a value of unity in the diagonal position, as required by the term $[\boldsymbol{I} + \boldsymbol{T}]$ in (59) above.

5.5.2 Limits on concurrent object detection

Given the analysis above, we may now ask: what are the limitations of the visual binding network? Specifically, how many objects may be represented by the visual binding network at one time? How distinct must these objects be, and in what ways, to be considered different objects?

It is well established in the theory of BSS that the number of sensors must be greater than or equal to the number of sources (Herault and Jutten 1986). This upper limit seems obvious in the case of de-mixing audio sources in that, for example, at least three microphones are required to reproduce three auditory signals. This is clearly a consequence of the mathematical fact that the same number of equations as unknowns is required when solving a linear system. Each feature input adds an equation and the capacity for an additional unknown source. Certainly, the upper limit on the number of distinct objects that could be separated by our network at any given time is the number of output neurons (ten in the present case). However, it is worth noting that over time the visual binding network will "forget" objects are no longer present, after temporal correlations in the input features corresponding to that object disappear, and learn new objects as they show up in the visual scene. By reusing its capacity over time, the network can represent more than this theoretical maximum number of objects over a longer time span.

Although the absolute upper bound on the instantaneous number of objects that can be represented is clearly the number of neurons, the practical limit is less obvious in our system because of the hybrid mix of visual submodalities represented in the inhibitory matrix. The network will automatically degenerate if some visual features are not present, effectively reducing its capacity. For example, if the color blue is not present in the visual scene, or if it is present but fluctuates in multiple objects in the same fashion, the weight matrix updates $\mathrm{d}\boldsymbol{T}_{10,k}/\mathrm{d}t$ and $\mathrm{d}\boldsymbol{T}_{j,10}/\mathrm{d}t$ will average to zero, resulting in zeros in the weight matrix $\boldsymbol{T}$ in row and column ten, which correspond to the color blue. Thus, the upper bound on the number of objects represented by the network could only be reached with objects all of which are distinct in the visual feature space used.

However, as many as three objects may be simultaneously represented even if they are not distinct in the feature space of the network. Given that we use three visual submodalities and that each first-stage network effectively has a "winner" at any given time, as long as objects have distinct fluctuations in all three visual submodalities, even if their individual responses apart from fluctuations are identical in each submodality, the network can separate them. This case is very similar to the case of auditory BSS, in which three sensors of the same type (microphones) are used to separate three sources. A visual example would be three green bars moving in the same direction with the same orientation (thus having identical responses to all three visual submodalities), but undergoing distinct patterns of light and shadow: These bars could all be simultaneously represented by the network.

A consequence of our understanding of these limits is that we can now clarify what would be required to increase the capacity of the visual binding network. Unlike microphones in the case of audition or cameras in stereo vision, the number of neurons in our network—corresponding directly to input visual features—cannot simply be increased without bound. Adding more outputs requires adding more *independent* inputs. For example, since the full space of visual motion is already spanned, adding more motion channels with different preferred directions than the four we have already employed adds nothing to the capacity of the network. Increasing the capacity of the network means adding more independent features, which in the case of the present network might include adding polarization sensitivity, as well as higher-level feature detectors such as edge, line, and contour detectors.

The rather low bound on the number of objects that can be simultaneously represented by the visual binding network does not obviously present a problem, particularly given that the network can rapidly switch its "attention" to a newly appeared object. It is certainly not true that humans are separately aware of every object in a complex visual scene, which on a busy street in a large city might include dozens of independent objects. Rather, we seem to have a rather limited capacity for object representation (Scholl 2001) and to maintain an awareness of the most salient objects (Itti et al. 1998) and switch our attention as necessary.

6 Discussion

We have described a neural network model that accomplishes a rudimentary form of visual binding inspired by the organization of neurons in a brain area known as *optic glomeruli*, which is just downstream of the optic lobes in the brains of insects. We have presented one example of successful visual binding using an image sequence containing two objects with distinct features, and detailed experimental results for this model are shown in a companion paper (Northcutt et al. 2017). We used the example results to illustrate both the extraction of explicit information about objects in the image from information implicit in the connection matrix, and a theoretical formulation that shows how the model reduces the problem of visual binding to one of blind source separation.

As the visual input changes, outputs of the visual binding network alternate in value according to the relative saliency of each object, and one might reasonably say that its "attention" is changing focus. We have described an algorithm to resynthesize elements of the input image so as to create an "enhanced image" that emphasizes the most salient object and de-emphasizes all others, and this image shares many characteristics with features of visual attention observed in living brains. This model clearly implements a simple form of visual attention, and the enhanced image makes it explicit the object upon which attention is focused.

We have also presented a thorough theoretical analysis of the model. This analysis reveals the function of the first- and second- stage networks, how they interact to produce the visual binding demonstrated, and how this network relates to existing models of blind source separation. This analysis shows the precise form of output that may be expected from the two-stage visual binding network. Further, this analysis provides upper and lower bounds on the number of objects that can be represented by the described network and explains how the upper limit might be increased.

A novel contribution of this model is the distinction we make here between *cooperative* and *competitive* learning rules, optimized, respectively, for the typical blind source separation case in which the number of hidden sources is the same as the network size, and for the overdetermined case in which there are fewer hidden sources, the usual case in visual binding.

An interesting characteristic of this network is that object perception, feature binding and attention all occur in a lowdimensional feature space in which all spatial information has been removed. This suggests an organization in which object detection, characterization and attention occur in a processing pathway distinct from that used for spatial localization and motion perception, consistent with what is known about visual processing in primates (DiCarlo et al. 2012), avians (Nguyen et al. 2004) and likely insects as well (Paulk et al. 2008).

The function of the second stage of the visual binding network as we have described it depends on the assumption that the hidden sources $\boldsymbol{s}(t)$, corresponding to fluctuations of independent visual objects, are *linearly* combined to produce summed image features $\boldsymbol{i}(t)$ as inputs to the network. Despite the fact that many image features themselves (in our case, particularly the visual motion computation) result from nonlinear computations, the summations of these feature images represent at least roughly linear functions of the hidden source fluctuations. Further, precise linearity of each visual feature is not a strict requirement for proper network operation.

Despite the fact that we have devised algorithms to explicitly reveal the number of objects detected by the visual binding network and their characteristics, these data are for human interpretation and are not required by the network. The visual binding network itself *implicitly* "knows" the number of objects in the visual scene and their characteristics without explicitly representing objects. This emergent low-level form of "intelligence" and "knowledge"—without explicit inclusion of either—brings to mind the seminal work of Brooks (1991; 1995). Because the present work is a model of an insect neuronal system that may have these implicit capabilities, but also has complex temporal dynamics, it may be that it is best analyzed with phase space techniques currently being applied to biological neuronal systems (Werner 2012) and could provide a promising pathway to developing artificial systems from which intelligence emerges without being explicitly built in.

Acknowledgements The authors would like to thank the Air Force Office of Scientific Research for early support of the modeling research on this project with Grant Number FA9550-07-1-0165, and the Air Force Research Laboratories for supporting this research to maturity with STTR Phase I Award Number FA8651-13-M-0085 and Phase II Award Number FA8651-14-C-0108, both in collaboration with Spectral Imaging Laboratory (Pasadena, CA). We would also like to thank the reviewers, whose input greatly enhanced this manuscript.

References

Bell AJ, Sejnowski TJ (1995) An information-maximization approach to blind separation and blind deconvolution. Neural Comput 7(6):1129–1159

Blakemore C, Tobin EA (1972) Lateral inhibition between orientation detectors in the cat's visual cortex. Exp Brain Res 15(4):439–440

Brooks RA (1991) Intelligence without representation. Artif Intell 47(1):139–159

Brooks RA (1995) Intelligence without reason. Building embodied, situated agents, The artificial life route to artificial intelligence, pp 25–81

Cichocki A, Bogner RE, Moszczyński L, Pope K (1997) Modified Herault–Jutten algorithms for blind separation of sources. Digit Signal Process 7(2):80–93

Comon P (1994) Independent component analysis, A new concept? Signal Process 36(3):287–314

DiCarlo JJ, Zoccolan D, Rust NC (2012) How does the brain solve visual object recognition? Neuron 73(3):415–434

Douglas RJ, Martin KA, Whitteridge D (1989) A canonical microcircuit for neocortex. Neural Comput 1(4):480–488

Guo L, Garland M (2006) The use of entropy minimization for the solution of blind source separation problems in image analysis. Pattern Recogn 39(6):1066–1073

Hassenstein B, Reichardt W (1956) Systemtheoretische Analyse der Zeit-, Reihenfolgen- und Vorzeichenauswertung bei der Bewegungsperzeption des Rüsselkäfers Chlorophanus. Z Naturforsch B 11(9–10):513–524

Hebb DO (1949) The organization of behavior: a neuropsychological theory. Wiley, New York

Herault J, Jutten C (1986) Space or time adaptive signal processing by neural network models. In: AIP conference proceedings, vol 151. Snowbird, UT, pp 206–211

Hopfield JJ (1991) Olfactory computation and object perception. Proc Natl Acad Sci USA 88(15):6462–6466

Hyvärinen A, Oja E (1998) Independent component analysis by general nonlinear Hebbian-like learning rules. Signal Process 64(3):301–313

Itti L, Koch C (2001) Computational modelling of visual attention. Nat Rev Neurosci 2(3):194–203

Itti L, Koch C, Niebur E (1998) A model of saliency-based visual attention for rapid scene analysis. IEEE Trans Pattern Anal 20(11):1254–1259

Joho M, Mathis H, Lambert RH (2000) Overdetermined blind source separation: Using more sensors than source signals in a noisy mixture. In: Proceedings international conference on independent component analysis and blind signal separation. Helsinki, Finland, pp 81–86

Jutten C, Herault J (1991) Blind separation of sources, Part I: an adaptive algorithm based on neuromimetic architecture. Signal Process 24(1):1–10

Lee DK, Itti L, Koch C, Braun J (1999) Attention activates winner-take-all competition among visual filters. Nat Neurosci 2(4):375–381

Linster C, Smith BH (1997) A computational model of the response of honey bee antennal lobe circuitry to odor mixtures: overshadowing, blocking and unblocking can arise from lateral inhibition. Behav Brain Res 87(1):1–14

Mach E (1866) Über die physiologische Wirkung räumlich vertheilter Lichtreize (On the physiological effects of spatially distributed light stimuli). Akad der Wiss, Wien, Sitzber, Math-Nat K 54(2):393

Markram H, Lübke J, Frotscher M, Sakmann B (1997) Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. Science 275(5297):213–215

Moran J, Desimone R (1985) Selective attention gates visual processing in the extrastriate cortex. Science 229(4715):782–784

Mu L, Ito K, Bacon JP, Strausfeld NJ (2012) Optic glomeruli and their inputs in Drosophila share an organizational ground pattern with the antennal lobes. J Neurosci 32(18):6061–6071

Nguyen AP, Spetch ML, Crowder NA, Winship IR, Hurd PL, Wylie DR (2004) A dissociation of motion and spatial-pattern vision in the avian telencephalon: implications for the evolution of visual streams. J Neurosci 24(21):4962–4970

Northcutt BD, Higgins CM (2017) An insect-inspired model for visual binding II: functional analysis and visual attention. In: Review, Biol Cybern

Northcutt BD, Dyhr JP, Higgins CM (2017) An insect-inspired model for visual binding I: learning objects and their characteristics. In: Review, Biol Cybern

Okamura JY, Strausfeld NJ (2007) Visual system of Calliphorid flies: motion- and orientation-sensitive visual interneurons supplying dorsal optic glomeruli. J Comp Neurol 500(1):189–208

Paulk AC, Phillips-Portillo J, Dacks AM, Fellous JM, Gronenberg W (2008) The processing of color, motion, and stimulus timing are anatomically segregated in the bumblebee brain. J Neurosci 28(25):6319–6332

Paulk AC, Dacks AM, Phillips-Portillo J, Fellous JM, Gronenberg W (2009) Visual processing in the central bee brain. J Neurosci 29(32):9987–9999

Rivera-Alvidrez Z, Lin I, Higgins CM (2011) A neuronally based model of contrast gain adaptation in fly motion vision. Vis Neurosci 28(5):419–431

Scholl BJ (2001) Objects and attention: the state of the art. Cognition 80(1):1–46

Shamma SA (1985) Speech processing in the auditory system II: lateral inhibition and the central processing of speech evoked activity in the auditory nerve. J Acoust Soc Am 78(5):1622–1632

Snyder AW (1979) Handbook of sensory physiology, vol VII/6A, chap 5. In: Autrum H (ed) Physics of vision in compound eyes. Springer, Berlin, Heidelberg, pp 225–313

Sorouchyari E (1991) Blind separation of sources. Part III: stability analysis. Signal Process 24(1):21–29

Strausfeld NJ, Okamura JY (2007) Visual system of Calliphorid flies: organization of optic glomeruli and their lobula complex efferents. J Comp Neurol 500(1):166–188

Strausfeld NJ, Sinakevitch I, Okamura JY (2007) Organization of local interneurons in optic glomeruli of the Dipterous visual system and comparisons with the antennal lobes. Dev Neurobiol 67(10):1267–1288

Trentelman H, Stoorvogel AA, Hautus M (2012) Control theory for linear systems. Springer, London

van Santen JPH, Sperling G (1985) Elaborated Reichardt detectors. J Opt Soc Am A 2(5):300–320

von der Malsburg C (1999) The what and why of binding: the modeler's perspective. Neuron 24(1):95–104

Werner G (2012) From brain states to mental phenomena via phase space transitions and renormalization group transformation: proposal of a theory. Cogn Neurodyn 6(2):199–202

Yang HH, Si A (1997) Adaptive online learning algorithms for blind separation: maximum entropy and minimum mutual information. Neural Comput 9(7):1457–1482