Neuronal Synchrony in Complex-Valued Deep Networks

David P. Reichert DAVID_REICHERT@BROWN.EDU
Thomas Serre THOMAS_SERRE@BROWN.EDU
Department of Cognitive, Linguistic & Psychological Sciences, Brown University

Appearing in the proceedings of the 2nd International Conference on Learning Representations (ICLR2014).

Abstract

Deep learning has recently led to great successes in tasks such as image recognition (e.g Krizhevsky et al., 2012). However, deep networks are still outmatched by the power and versatility of the brain, perhaps in part due to the richer neuronal computations available to cortical circuits. The challenge is to identify which neuronal mechanisms are relevant, and to find suitable abstractions to model them. Here, we show how aspects of spike timing, long hypothesized to play a crucial role in cortical information processing, could be incorporated into deep networks to build richer, versatile representations.

We introduce a neural network formulation based on complex-valued neuronal units that is not only biologically meaningful but also amenable to a variety of deep learning frameworks. Here, units are attributed both a firing rate and a phase, the latter indicating properties of spike timing. We show how this formulation qualitatively captures several aspects thought to be related to neuronal synchrony, including gating of information processing and dynamic binding of distributed object representations. Focusing on the latter, we demonstrate the potential of the approach in several simple experiments. Thus, neuronal synchrony could be a flexible mechanism that fulfills multiple functional roles in deep networks.

1. Introduction

Deep learning approaches have proven successful in various applications, from machine vision to language processing (Bengio et al., 2012). Deep networks are often taken to be inspired by the brain as idealized neural networks that learn representations through several stages of non-linear processing, perhaps akin to how the mammalian cortex adapts to represent the sensory world. These approaches are thus also relevant to computational neuroscience (Cadieu et al., 2013): for example, convolutional networks (LeCun et al., 1989) possibly capture aspects of the organization of the visual cortex and are indeed closely related to biological models like HMAX (Serre et al., 2007), while deep Boltzmann machines (Salakhutdinov & Hinton, 2009) have been applied as models of generative cortical processing (Reichert et al., 2013).

The most impressive recent deep learning results have been achieved in classification tasks, in a processing mode akin to rapid feed-forward recognition in humans (Serre et al., 2007), and required supervised training with large amounts of labeled data. It is perhaps less clear whether current deep networks truly support neuronal representations and processes that naturally allow for flexible, rich reasoning about e.g. objects and their relations in visual scenes, and what machinery is necessary to learn such representations from data in a mostly unsupervised way. At the implementational level, there is a host of cortical computations not captured by the simplified mechanisms utilized in deep networks, from the complex laminar organization of the cortex to dendritic computations or neuronal spikes and their timing. Such mechanisms might be key to realizing richer representations, but the challenge is to identify which of these mechanisms are functionally relevant and which can be discarded as mere implementation details.

One candidate mechanism is temporal coordination of neuronal output, or in particular, synchronization of neuronal firing. Various theories posit that synchrony is a key element of how the cortex processes sensory information (e.g. von der Malsburg, 1981; Crick, 1984; Singer & Gray, 1995; Fries, 2005; Uhlhaas et al., 2009; Stanley, 2013), though these theories are also contested (e.g. Shadlen & Movshon, 1999; Ray & Maunsell, 2010). Because the degree of synchrony of neuronal spikes affects the output of downstream neurons, synchrony has been postulated to allow for gating of information transmission between neurons or whole cortical areas (Fries, 2005; Benchenane et al., 2011). Moreover, the relative timing of neuronal spikes may carry information about the sensory input and the dynamic network state (e.g. Geman, 2006; Stanley, 2013), beyond or in addition to what is conveyed by firing rates. In particular, neuronal subpopulations could dynamically form synchronous groups to *bind* distributed representations (Singer, 2007), to signal that perceptual content represented by each group forms a coherent entity such as a visual object in a scene.

Here, we aim to demonstrate the potential functional role of neuronal synchrony in a framework that is amenable to deep learning. Rather than dealing with more realistic but elaborate spiking neuron models, we thus seek a mathematical idealization that naturally extends current deep networks while still being interpretable in the context of biological models. To this end, we use complex-valued units, such that each neuron's output is described by both a firing rate and a phase variable. Phase variables across neurons represent relative timing of activity.

In Section 2, we briefly describe the effect of synchrony on neuronal information processing. We present the framework based on complex-valued networks, and show what functional roles synchrony could play, within this framework. Thanks to the specific formulation employed, we had some success with converting deep nets trained without synchrony to incorporate synchrony. Using this approach, in Section 3 we underpin our argument with several simple experiments, focusing on binding by synchrony. Exploiting the presented approach further will require learning with synchrony. We discuss principled ways to do so and challenges to overcome in Section 4.

It should be noted that complex-valued neural networks are not new (e.g. Zemel et al., 1995; Kim & Adalı, 2003; Nitta, 2004; Fiori, 2005; Aizenberg & Moraga, 2007; Savitha et al., 2011; Hirose, 2011). However, they do not seem to have attracted much attention within the deep learning community—perhaps because their benefits still need to be explored further.[1] There are a few cases where such networks were employed with the interpretation of neuronal synchrony, including the work of Rao et al. (2008), Rao & Cecchi (2010; 2011), which is similar to ours. These prior approaches will be discussed in Section 4.

2. Neuronal synchrony

Cortical neurons communicate with electric action potentials (so-called spikes). There is a long-standing debate in neuroscience on whether various features of spike timing matter to neuronal information processing, rather than just average firing rates (e.g. Stanley, 2013). In common deep neural networks (convolutional networks, Boltzmann machines, etc.), the output of a neuronal unit is characterized by a single (real-valued) scalar; the state of a network and how it relates to an interpretation of sensory input is fully determined by the joint scalar outputs across all units. This suggests an interpretation in terms of average, static firing rates, lacking any notion of relative timing. Here, we consider how to incorporate such notions into deep networks.

Consider Figure 1a for an example of how a more dynamic code could be transmitted between neurons (simulated with the Brian simulator, Goodman & Brette, 2009). This example is based on the hypothesis that neuronal rhythms, ubiquitous throughout the brain, play a functional role in information processing (e.g. Singer & Gray, 1995; Fries, 2005; Uhlhaas et al., 2009; Benchenane et al., 2011). A neuron receives spike train inputs modulated by oscillatory firing rates. This results in rhythmic output activity, with an average firing rate that depends both on the amplitudes and relative phase of the inputs (Figure 1b). Such interactions are difficult to represent with just static firing rates.

2.1. Modeling neuronal synchrony with complex-valued units

In deep networks, a neuronal unit receives inputs from other neurons with states vector $\mathbf{x}$ via synaptic weights vector $\mathbf{w}$. We denote the total 'postsynaptic' input as $\chi := \mathbf{w} \cdot \mathbf{x}$. The output is computed with an activation function f as $f(\chi)$ (or, in the case of Gibbs-sampling in Boltzmann machines, $f(\chi)$ is a conditional probability from which the output state is sampled).[2] We can now model aspects of spike timing by replacing the real-valued states $\mathbf{x}$ with complex states $\mathbf{z}$. For unit state $z_i = r_i e^{i\phi_i}$, the magnitude $r_i = |z_i|$ can be interpreted as the average firing rate analogously to the real-valued case. The phase ϕ_i could correspond to the phase of a neuronal rhythm as in Figure 1a, or, more generally, the timing of maximal activity in some temporal interval (Figure 1c). Because neuronal messages are now added in the complex plane (keeping the weights $\mathbf{w}$ real-valued, for now), a neuron's total input $\zeta := \mathbf{w} \cdot \mathbf{z}$ no longer depends only on the firing rates of the input units, and the strength of the synapses, but also their *relative timing*. This naturally accounts for the earlier, spiking neuron example: input states that are synchronous, i.e. have similar phases, result in a stronger total input, whereas less synchronous inputs result in weaker total input (Figure 1d).

A straightforward way to define a neuron's output state $z_i = r_i e^{i\phi_i}$ from the (complex-valued) total input ζ is to apply an activation function, $f : \mathbb{R}^+ \mapsto \mathbb{R}^+$, to the input's magnitude $|\zeta|$ to compute the output magnitude, and to set the output phase to the phase of the total input:

$$\phi_i = \arg(\zeta), \quad r_i = f(|\zeta|), \quad \text{where } \zeta = \mathbf{w} \cdot \mathbf{z}. \tag{1}$$

[1] Beyond possibly applications where the data itself is naturally represented in terms of complex numbers.

[2] Operations such as max-pooling require separate treatment. Also, bias parameters $\mathbf{b}$ can be added to the inputs to control the intrinsic excitability of the neurons. We omit them for brevity.

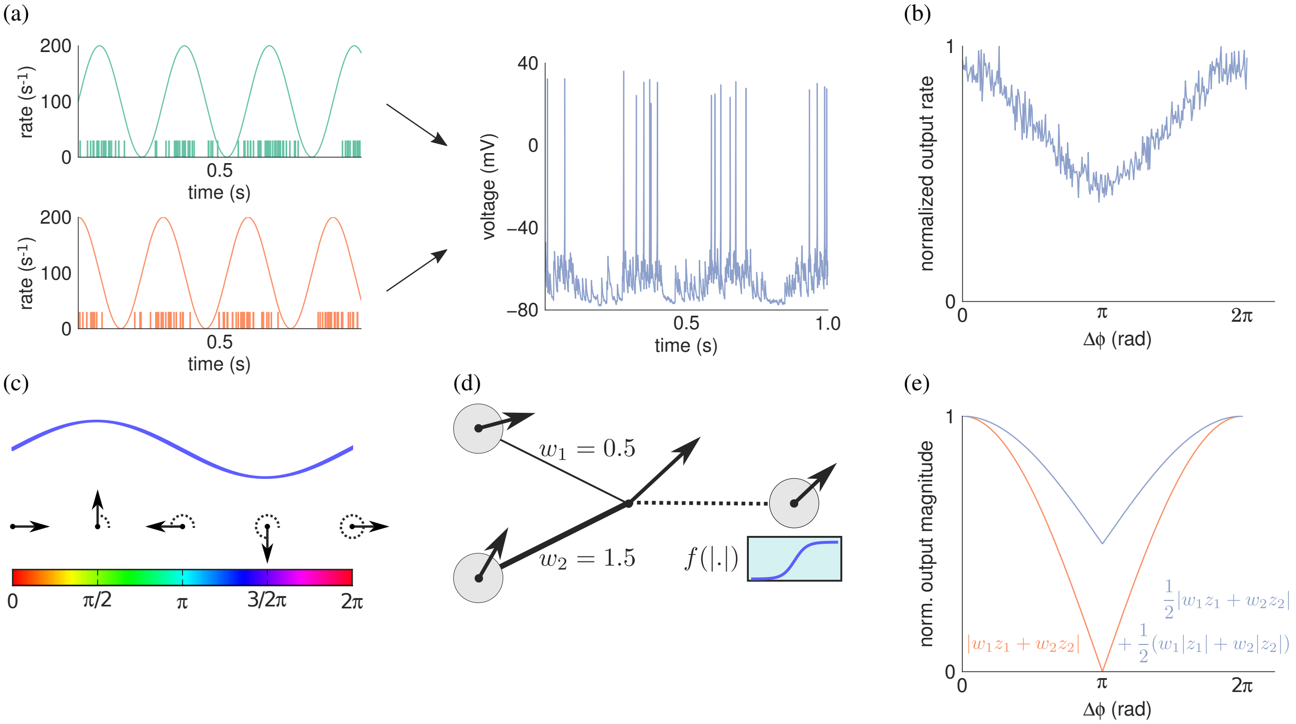


Figure 1. **Transmission of rhythmical activity, and corresponding model using complex-valued units.** (a) A Hodgkin–Huxley model neuron receives two rhythmic spike trains as input, plus background activity. The inputs are modeled as inhomogeneous Poisson processes modulated by sinusoidal rate functions (left; shown are rates and generated spikes), with identical frequencies but differing phases. The output of the neuron is itself rhythmical (right; plotted is the membrane potential). (b) The neuron's output rate is modulated by the phase difference between the two inputs (rate averaged over 15s runs). (c) We represent the timing of maximal activity of a neuron as the phase of a complex number, corresponding to a direction in the complex plane. The firing rate is the magnitude of that complex number. Also shown is the color coding used to indicate phase throughout this paper (thus, figures should be viewed in color). (d) The outputs of the input neurons are scaled by synaptic weights (numbers next to edges) and added in the complex plane. The phase of the resulting complex input determines the phase of the output neuron. The activation function f is applied to the magnitude of the input to compute the output magnitude. Together, this models the influence of synchronous neuronal firing on a postsynaptic neuron. (e) Output magnitude as function of phase difference of two inputs. With a second term added to a neuron's input, out-of-phase excitation never cancels out completely (see main text for details; curves are for $w_1 = w_2 > 0$, $|z_1| = |z_2|$). Compare to 1b.

Again this is intuitive as a biological model, as the total strength and timing of the input determine the 'firing rate' and timing of the output, respectively.

There are, however, issues with this simple approach to modeling neuronal synchrony, which are problematic for the biological model but also, possibly, for the functional capabilities of a network. In analogy to the spiking neuron example, consider two inputs to a neuron that are excitatory (i.e., $w_1, w_2 > 0$), and furthermore of equal magnitude, $|w_1 z_1| = |w_2 z_2|$. While it is desirable that the net total input is decreased if the two inputs are out of phase, the net input in the complex-valued formulation can actually be zero, if the difference in input phases is π, no matter how strong the individual inputs (Figure 1e, lower curve). Biologically, it seems unrealistic that strong excitatory input, even if not synchronized, would not excite a neuron.[3]

Moreover, in the above formulation, the role of inhibition (i.e., connections with $w < 0$) has changed: inputs with negative weights are equivalent to *excitatory inputs of the opposite phase*, due to $-1 = e^{i\pi}$. Again, this is a desirable property that leads to desynchronization between neuronal groups, in line with biological models, as we will show below. However, it also means that inputs from connections with negative weights, on their own, can strongly drive a neuron; in that sense, there is no longer actual inhibition that always has a suppressive effect on neuronal outputs. Additionally, we found that the phase shifting caused by inhibition could result in instability in networks with dominant negative weights, leading to fast switching of the phase variables.

[3] Arguably, refractory periods or network motifs such as disynaptic feedforward inhibition (Gabernet et al., 2005; Stanley, 2013) could indeed result in destructive interference of out-of-phase excitation.

We introduce a simple fix for these issues, modifying how the output magnitude of a neuron is computed as follows:

$$r_i = f(|\zeta|) \hookrightarrow r_i = f(\frac{1}{2}|\zeta| + \frac{1}{2}\chi), \quad \text{where } \zeta = \mathbf{w} \cdot \mathbf{z}, \quad \chi := \mathbf{w} \cdot |\mathbf{z}|. \tag{2}$$

The first term, which we refer to as *synchrony term*, is the same as before. The second, *classic term*, applies the weights to the magnitudes of the input units and thus does not depend on their phases; a network using only the classic terms reduces to its real-valued counterpart (we thus reuse the variable χ, earlier denoting postsynaptic input in a real-valued network). Together, the presence of the classic term implies that excitatory input always has a net excitatory component, even if the input neurons are out of phase such that the synchrony term is zero (thus matching the spiking neuron example,[4] compare Figures 1b and 1e). Similarly, input from negative connections alone is never greater than zero. Lastly, this formulation also makes it possible to give different weightings to synchrony and classic terms, thus controlling how much impact synchrony has on the network; we do not explore this possibility here.

2.2. The functional relevance of synchrony

The advantage of using complex-valued neuronal units rather than, say, spiking neuron models is that it is natural to consider how to apply deep learning techniques and extend existing deep learning neural networks in this framework. Indeed, our experiments presented later are based on pretraining standard, real-valued nets (deep Boltzmann machines in this case) and converting them to complex-valued nets after training. In this section, we briefly describe how our framework lends itself to realize two functional roles of synchrony as postulated by biological theories.

Grouping (binding) by synchrony

The activation of a real-valued unit in an artificial neural network can often be understood as signaling the presence of a feature or combination of features in the data. The phase of a complex-valued unit could provide additional information about the feature. *Binding by synchrony* theories (Singer, 2007) postulate that neurons in the brain dynamically form synchronous assemblies to signal where distributed representations together correspond to coherent sensory entities. For example, different objects in a visual scene would correspond to different synchronous assemblies in visual cortex. In our formulation, phases can analogously signal a soft assignment to different assemblies.

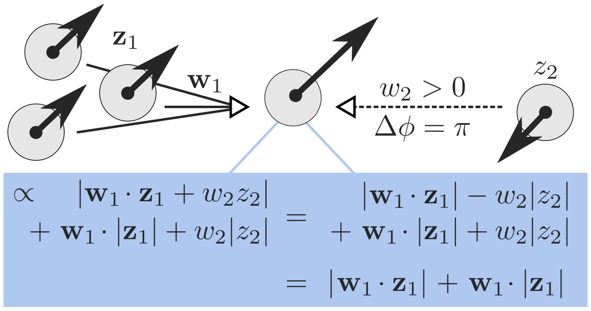

$\mathbf{z}_1$
$\mathbf{w}_1$
$w_2 > 0$
$\Delta\phi = \pi$
z_2
$\propto \; |\mathbf{w}_1 \cdot \mathbf{z}_1 + w_2 z_2| + \mathbf{w}_1 \cdot |\mathbf{z}_1| + w_2|z_2|$
$= |\mathbf{w}_1 \cdot \mathbf{z}_1| - w_2|z_2| + \mathbf{w}_1 \cdot |\mathbf{z}_1| + w_2|z_2|$
$= |\mathbf{w}_1 \cdot \mathbf{z}_1| + \mathbf{w}_1 \cdot |\mathbf{z}_1|$

Figure 2. **Gating of interactions.** Out-of-phase input, when combined with a stronger input, is weakened. In this example, with $\Delta\phi = \pi$ and as long as $|\mathbf{w}_1 \cdot \mathbf{z}_1| > |w_2 z_2|$, effective input from the neuron to the right is zero, for any input strength (classic and synchrony terms contributions cancel, bottom panel). Hence, neuronal groups with different phases (gradually) decouple.

Importantly, communication with complex-valued messages also naturally leads to different synchronous groups emerging: for excitatory connections, messages that 'agree' (Zemel et al., 1995) in their phases prevail over those that do not; inhibitory messages, on the other hand, equate to excitatory messages of opposite phases, and thus encourage *de*synchronization between neurons. For comparison, consider the more realistic spiking model of visual cortex of Miconi & VanRullen (2010), where synchronous groups arise from a similar interaction between excitation and inhibition. That these interactions can indeed lead to meaningful groupings of neuronal representations in deep networks will be shown empirically in Section 3.

Dynamic gating of interactions and information flow

Because synchrony affects which neuronal messages are transmitted preferentially, it has also been postulated that synchrony may gate information flow dynamically depending on the sensory input, the current network state and top-down control (Stanley, 2013), as well as to modulate the effective interactions between cortical areas depending on their level of coherence (Fries, 2005; Benchenane et al., 2011). A similar modulation of interactions can be reproduced in our framework. Let us consider an example scenario (Figure 2) where a neuron is a member of a synchronous assembly, receiving excitatory inputs $\mathbf{w}_1 \cdot \mathbf{z}_1$ from neurons that all have similar phases. Now consider the effect of adding another neuron that also provides excitatory input, $w_2 z_2$, but of a different phase, and assume that $|\mathbf{w}_1 \cdot \mathbf{z}_1| < |w_2 z_2|$ (i.e. the input of the first group dominates). The net effect the latter additional input has depends again on the phase difference. In particular, if the phase difference is maximal (π), the net contribution from the second neuron turns out to be *zero*. The output magnitude is computed as in Eq. 2, taking both synchrony and classic terms into account. In the complex plane, $w_2 z_2$ is antiparal-

[4] Real neuronal networks and realistic simulations have many degrees of freedom, hence we make no claim that our formulation is a general or quantitative model of neuronal interactions.

lel to $\mathbf{w}_1 \cdot \mathbf{z}_1$, thus the synchrony term is reduced by $|w_2 z_2|$. However, this reduction is exactly canceled out by the classic term contribution from the second input (Figure 2 lower panel). There is also no effect on the output phase as the phase of the total input remains equal to the phase of $\mathbf{w}_1 \cdot \mathbf{z}_1$. Analogous reasoning applies for inhibitory connections.

Thus, the effective connectivity between neuronal units is modulated by the units' phases, which themselves are a result of network interactions. In particular, if inference results in neurons being segregated into different assemblies (ideally because they represent independent causes in the sensory input, or independent regions in an image), existent connections between groups are weakened.

3. Experiments: the case of binding by synchrony

In this section, we support our reasoning with several simple experiments, and further elucidate on the possible roles of synchrony. We focus on the binding aspect.

All experiments were based on pretraining networks as normal, real-valued deep Boltzmann machines (DBMs, Salakhutdinov & Hinton, 2009). DBMs are multi-layer networks that are framed as probabilistic (undirected graphical) models. The *visible* units make up the first layer and are set according to data, e.g. images. Several *hidden* layers learn internal representations of the data, from which they can generate the latter by sampling the visible units. By definition, in a DBM there are only (symmetric) connections between adjacent layers and no connections within a layer. Given the inputs from adjacent layers, a unit's state is updated stochastically with a probability given by a sigmoid (logistic) activation function (implementing Gibbs sampling). Training was carried out layer-wise with standard methods including contrastive divergence (for model and training details, see Appendix B). Training and experiments were implemented within the Pylearn2 framework of Goodfellow et al. (2013b).

We emphasize however that our framework is not specific to DBMs, but can in principle be adapted to various deep learning approaches (we are currently experimenting with networks trained as autoencoders or convolutional networks). The learning and inference procedures of a DBM derive from its definition as a probabilistic model, but for our purpose here it is more appropriate to simply think of a DBM as a multi-layer recurrent neural network (cf. Goodfellow et al., 2013a) with logistic activation function;[5] we can demonstrate how our framework works by taking the pretrained network, introducing complex-valued unit states and applying the activation function to magnitudes as described in Section 2.1.[6] However, developing a principled probabilistic model based on Boltzmann machines to use with our framework is possible as well (Section 4).

This conversion procedure applied to real-valued networks offers a simple method of exploring aspects of synchrony, but there is no guarantee that it will work (for additional discussion, see Appendix B). We use it here to show what the functional roles of synchrony could be in principle; learning with synchrony will be required to move beyond simple experiments (Section 4).

Throughout the experiments, we clamped the magnitudes of the visible units according to (binary) input images, which were not seen during training, and let the network infer the hidden states over multiple iterations. The phases of the visible layer were initialized randomly and then determined by the input from the hidden layer above. Hence, any synchronization observed was spontaneous.

3.1. Dynamic binding of independent components in distributed representations

In this first experiment, we trained a DBM with one hidden layer (a restricted Boltzmann machine, Smolensky, 1986) on a version of the classic 'bars problem' (Földiák, 1990), where binary images are created by randomly drawing horizontal and vertical bars (Figure 3a). This dataset has classically been used to test whether unsupervised learning algorithms can find the independent components that constitute the image, by learning to represent the individual bars (though simple, the bars problem is still occasionally employed, e.g. Lücke & Sahani, 2008; Spratling, 2011). We chose this dataset specifically to elucidate on the role of synchrony in the context of distributed representations.

We hard-coded the receptive field sizes (regions with nonzero weights to the input) of the hidden units to be restricted to regions smaller than the entire image (but together tiling the whole image). By necessity, this implies that any individual unit can never fully represent a full-length bar, in the sense that the the unit's weights correspond to the bar, or that one can read out the presence of the full bar from this unit's state alone. However, this does not imply that the full network cannot learn that the images are constituted by bars (as long as receptive fields overlap). For example, we found that when sampling from the model (activating hidden and visible units freely), the resulting images contained full-length bars most of the time (see supplementary figure S1a and supplementary videos, Appendix A); similarly, the network would fill in the remainder of a bar when the visible units where clamped to a part of it.

After conversion to a complex-valued network, the model

[5] The activation function is stochastic in the case of Gibbs sampling, deterministic in the case of mean-field inference.

[6] Our results were qualitatively similar whether we computed the output magnitudes stochastically or deterministically.

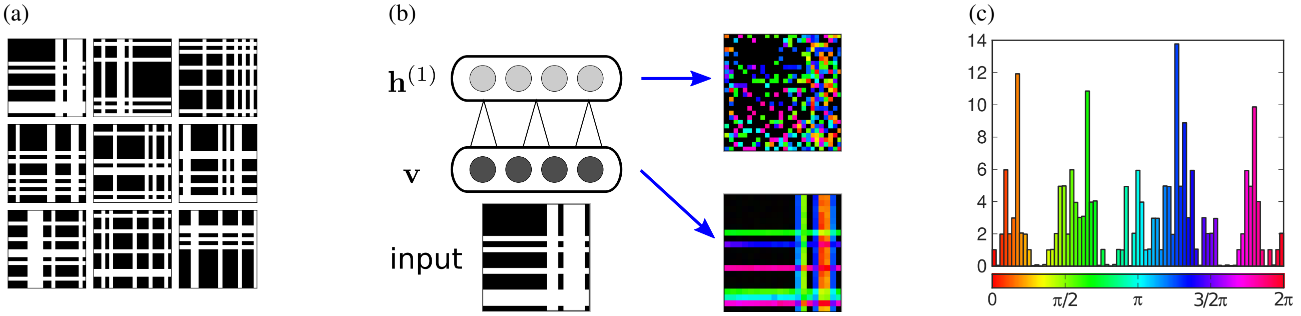


Figure 3. **Binding by synchrony in shallow, distributed representations.** (a) Each image of our version of the bars problem contained 6 vertical and 6 horizontal bars at random positions. (b) A restricted Boltzmann machine was trained on bars images and then converted to a complex-valued network. The magnitudes of the visible units were clamped according to the input image (bottom left), whereas the hidden units and phases of the visible units were activated freely. After 100 iterations, units representing the various bars were found to have synchronized (right; the phases are color-coded for units that are active; black means a unit is off). The neurons synchronized even though receptive fields of the hidden units were constrained to be smaller than the bars. Thus, binding by synchrony could make the 'independent components' of sensory data explicit in distributed representation, in particular when no single neuron can possibly represent a component (a full-length bar) on its own. (c) Histogram of the unit phases in the visible layer for the example shown in b.

was run on input images for 100 iterations each. Results are plotted in Figure 3b, depicting both visible and hidden states for one input image (Further examples in Figure S1b). We found that visible neurons along a bar would often synchronize to the same phase (except where bars crossed), whereas different bars tended to have different phase values. Figure 3c shows a histogram of phase values in the visible layer for this example image, with clear peaks corresponding to the phases of the bars. Such synchronization was also found in the hidden layer units (3b).

Based on these results, we make three points. First, the results show that our formulation indeed allows for neurons to dynamically organize into meaningful synchronous assemblies, even without supervision towards what neurons should synchronize to, e.g. by providing phase targets in training—here, synchrony was not used in training at all. That the conversion from a real-valued network can work suggests that an unsupervised or semi-supervised approach to learning with synchrony could be successful as long as synchrony benefits the task at hand.

Second, synchronization of visible and hidden units, which together represent individual bars, can occur for neurons several synapses apart. At the same time, not all bars synchronized to different phases. The number of distinct, stable phase groups that can be formed is likely to be limited. Notably, it has been argued that this aspect of synchrony coding explains certain capacity limits in cognition (Jensen & Lisman, 2005; Fell & Axmacher, 2011).

The third point relates to the nature of distributed representations. For the bars problem, whether a neural net (or probabilistic model) discovers the bars is usually evaluated by examining whether individual units correspond to individual bars, as can be seen by inspecting the weights or by probing the response properties of individual neurons (e.g. Lücke & Sahani, 2008; Spratling, 2011). A similar 'localist' approach was taken in recent attempts to make sense of the somewhat opaque hidden representations learned by deep networks (as in the example of the neurons that discovered the 'concept' of a cat from unsupervised learning Le et al., 2011, or the work of Zeiler & Fergus, 2013 on analyzing convolutional networks). In our experiment, it is not possible to map individual neurons to the image constituents, by construction; bars could only be represented in a distributed fashion. Synchrony could make explicit which neurons together represent a sensory entity, e.g. for a readout (more on that below), as well as offer a mechanism that establishes the grouping in the first place.

3.2. Binding in deep networks

To examine the effects of synchrony in deeper networks, we trained a DBM with three hidden layers on another dataset, consisting of binary images that contained both four 'corners' arranged in a square shape (centered at random positions) and four corners independently drawn (Figure 4a). Receptive field sizes in the first hidden layer were chosen such that the fields would only cover individual corners, not the whole square arrangements, making it impossible for the first hidden layer to discover the latter during training.[7] Receptive field sizes were larger in higher layers, with the topmost hidden layer being fully connected.

After converting the net to complex values, we found that the four corners arranged as a square would often synchronize to one phase, whereas the other, independent corners

[7] Note that there was only layer-wise pretraining, no training of the full DBM, thus first layer representations were not influenced by higher layers during training either.

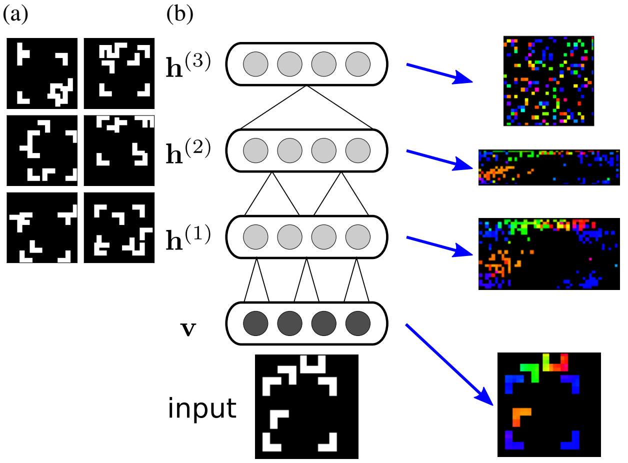


Figure 4. **Binding by synchrony in a deep network.** (a) Each image contained four corners arranged in a square shape, and four randomly positioned corners. (b) The four corners arranged in a square were usually found to synchronize. The synchronization of the corresponding hidden units is also clearly visible in the hidden layers. The receptive field sizes in the first hidden layer were too small for a hidden unit to 'see' more than individual corners. Hence, the synchronization of the neurons representing the square in the fist hidden and visible layers was due to feedback from higher layers (the topmost hidden layer had global connectivity).

would assume one or multiple phases different from the phase of the square (Figure 4b; more examples Figure S1c). Synchronization was also clear in the hidden layers.

Again we make several observations. First, because of the restricted receptive fields, the synchronization of the units representing parts of the square in the visible layer and first hidden layer was necessarily due to top-down input from the higher layers. Whether or not a corner represented by a first layer neuron was part of a larger whole was made explicit in the synchronous state. Second, this example also demonstrates that neurons need not synchronize through connected image regions as was the case in the bars experiment. Lastly, note that, with or without synchrony, restricted receptive fields and topographic arrangement of hidden units in intermediate hidden layers make it possible to roughly identify which units participate in representing the same image content, by virtue of their position in the layers. This is no longer possible with the topmost, globally connected layer. By identifying hidden units in the topmost layer with visible units of similar phase, however, it becomes possible to establish a connection between the hidden units and what they are activated by in image space.

3.3. Reading out object representations via phase

With a final set of experiments, we demonstrate that individual synchrony assemblies can be selected on the basis of their phase, and their representational content be accessed one group at a time. We trained on two additional datasets containing multiple simple objects: one with images of geometric toy shapes (triangles or squares, Figure 5a), with three randomly chosen instances per image, and a dataset where we combined handwritten digits from the commonly used MNIST dataset with the geometric shapes (Figure 5c). As before, we found a tendency in the complex-valued network to synchronize individual objects in the image to distinct phases (Figure 5b, d, Figure S1d, e).

After a network was run for a fixed number of steps, for each layer, units were clustered according to their activity vectors in the complex plane. For clustering we assumed for simplicity that the number of objects was known in advance and used k-means, with k, the number of clusters, being set to the number of objects plus one for a general background. In this fashion, each neuron was assigned to a cluster, and the assignments could be used to define masks to read out one representational component at a time.[8]

For the visible layer, we thus obtained segmented images as shown in Figures 5b, d. Especially for the modified MNIST images, the segmentations are often noisy. However, it is noteworthy that segmentations can be obtained at all, given that the network training involved no notion of segmentation. Moreover, binding by synchrony is more general than segmentation (in the sense of assigning labels to pixels), as it applies to *all* neurons in the network and, in principle, to arbitrarily abstract and non-visual forms of representation.

Thus, units can also be selected in the hidden layers according to phase. The phase-masked representations could, for instance, be used for classification, one object at a time. We can also decode what these representations corresponded to in image space. To this end, we took the masked states for each cluster (treating the other states as being zero[9]) and used a simple decoding procedure as described by Reichert et al. (2010); Reichert (2012), performing a single deterministic top-down pass in the network (with doubled weights) to obtain a reconstructed image. See Figure 6 for an example. Though the images decoded in this fashion are somewhat noisy, it is apparent that the higher layer units do indeed represent the same individual objects as the visible layer units that have assumed the same phase (in cases where objects are separated well).

Earlier, we discussed gating by synchrony as it arises from the effect that synchrony has directly on network interac-

[8] Alternatively, peaks could be selected in the phase histogram of a layer and units masked according to distance to the peaks, allowing for overlapping clusters.

[9] This only works in a network where units signal the presence of image content by being on and not by being off, so that setting other units to zero has the effect of removing image content. This can be achieved with inductive biases such as sparsity being applied during training, see the discussion of Reichert et al. (2011).

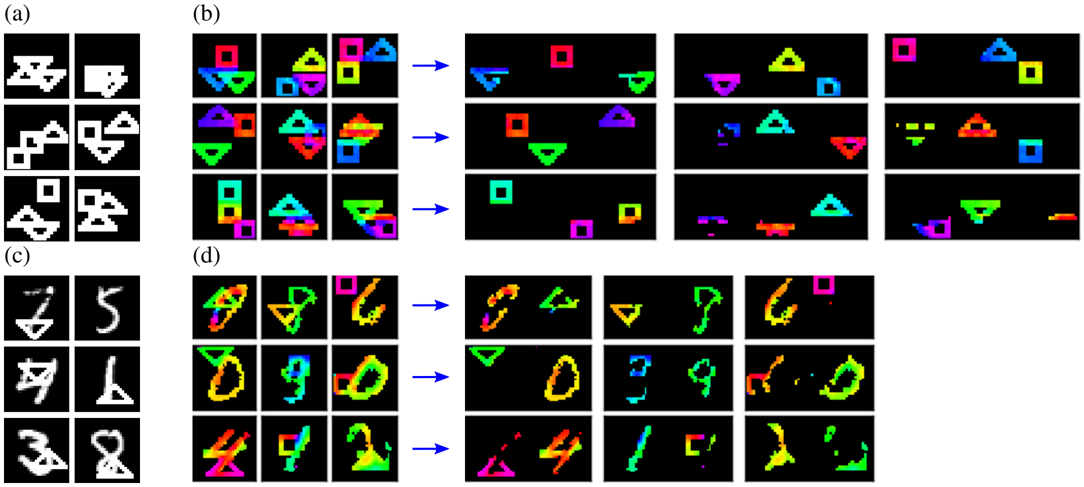


Figure 5. **Simple segmentation from phases.** (a) The 3-shapes set consisted of binary images each containing three simple geometric shapes (square, triangle, rotated triangle). (b) Visible states after synchronization (left), and segmented images (right). (c) For each image in the MNIST+shape dataset, a MNIST digit and a shape were drawn each with probability 0.8. (d) Analogous to (b).

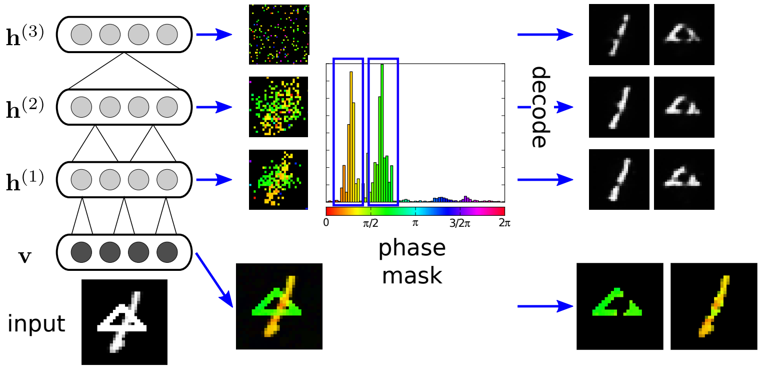


Figure 6. **Using phase to access and decode internal object representations.** By selecting subsets of neurons according to their phase (e.g. through clustering), representations of each object can be read out one by one (right-hand side). For the hidden layers, plotted are images decoded from each of the synchronous sub-populations, using a simple decoding procedure (see main text).

tions. Selecting explicitly individual synchrony assemblies for further processing, as done here, is another potential form of gating by synchrony. In the brain, some cortical regions, such as in prefrontal cortex, are highly interconnected with the rest of the cortex and implement functions such as executive control and working memory that demand flexible usage of capacity-limited resources according to context and task-demands. Coherence of cortical activity and synchrony have been suggested to possibly play a causal role in establishing dynamic routing between these areas (e.g. Benchenane et al., 2011; Miller & Buschman, 2013). Similarly, attentional processing has been hypothesized to emerge from a dynamically changing, globally coherent state across the cortex (e.g. Duncan et al., 1997; Miller & Buschman, 2013). It is possible that there are dedicated structures in the brain, such as the pulvinar in the thalamus, that coordinate cross-cortical processing and cortical rhythms (e.g. Shipp, 2003; Saalmann et al., 2012). In our model, one could interpret selecting synchrony assemblies as prefrontal areas reading out subsets of neuronal populations as demanded by the task. Through binding by synchrony, such subsets could be defined dynamically across many different cortical areas (or at least several layers in a feature hierarchy, in our model).

4. Discussion

We argue that extending neural networks beyond real-valued units could allow for richer representations of sensory input. Such an extension could be motivated by the fact that the brain supports such richer coding, at least in principle. More specifically, we explored the notion of neuronal synchrony in deep networks. We motivated the hypothetical functional roles of synchrony from biological theories, introduced a formulation based on complex-valued units, showed how the formulation related to the biological phenomenon, and examined its potential in simple experiments. Neuronal synchrony could be a versatile mechanism that supports various functions, from gating or modulating neuronal interactions to establishing and signaling semantic grouping of neuronal representations. In the latter case, synchrony realizes a form of soft constraint on neuronal representations, imposing that the sensory world should be organized according to distinct perceptual entities such as objects. Unfortunately, this melding of various functional roles might make it more difficult to treat the synchrony mechanism in principled theoretical terms.

The formulation we introduced is in part motivated by it being interpretable in a biological model. It can be understood as a description of neurons that fire rhythmically, and/or in relation to a global network rhythm. Other aspects of spike timing could be functionally relevant,[10] but the complex-valued formulation can be seen as a step beyond current, 'static firing rate' neural networks. The formulation also has the advantage of making it possible to explore synchrony in converted pretrained real-valued networks (without the addition of the classic term in Eq. 2, the qualitative change of excitation and inhibition is detrimental to this approach). However, for the machine learning application, various alternative formulations would be worthy of exploration (different weights in synchrony and classic terms, complex-valued weights, etc.).

We presented our simulation results in terms of representative examples. We did not provide a quantitative analysis, simply because we do not claim that the current, simple approach would compete with, for example, a dedicated segmentation algorithm, at this point. In particular, we found that the conversion from arbitrary real-valued nets did not consistently lead to favorable results (we provide additional comments in Appendix B). Our aim with this paper is to demonstrate the synchrony concept, how it could be implemented and what functions it could fulfill, in principle, to the deep learning community. To find out whether synchrony is useful in real applications, it is necessary to develop appropriate learning algorithms. We address learning in the context of related work in the following.

We are aware of a few examples of prior work employing complex-valued neural networks with the interpretation of neuronal synchrony.[11] Zemel et al. (1995) introduced the 'directional unit Boltzmann machine' (DUBM), an extension of Boltzmann machines to complex weights and states (on the unit circle). A related approach is used by Mozer et al. (1992) to model binding by synchrony in vision, performing phase-based segmentation of simple geometric contours, and by Behrmann et al. (1998) to model aspects of object-based attention. The DUBM is a principled probabilistic framework, within which a complex-valued extension of Hebbian learning can be derived, with potentially interesting functional implications and biological interpretations in terms of Spike Timing Dependant Plasticity (Sjöström & Gerstner, 2010). For our purposes, the DUBM energy function could be extended to include synchrony and classic terms (Eq. 2), if desired, and to allow the units to switch off (rather than being constrained to the unit circle), e.g. with a 'spike and slab' formulation (Courville et al., 2011; Kivinen & Williams, 2011). Performing mean-field inference in the resulting model should be qualitatively similar to running the networks used in our work here.

The original DUBM applications were limited to simple data, shallow architectures, and supervised training (input phases were provided). It would be worthwhile to reexamine the approach in the context of recent deep learning developments; however, training Boltzmann machines successfully is not straightforward, and it is not clear whether approximate training methods such as contrastive divergence (Hinton, 2002; 2010) can be translated to the complex-valued case with success.

Weber & Wermter (2005) briefly describe a complex-valued neural network for image segmentation, modeling synchrony mediated by lateral interactions in primary visual cortex, though the results and analysis presented are perhaps too limited to conclude much about their approach. A model of binding by synchrony in a multi-layer network is proposed by Rao et al. (2008) and Rao & Cecchi (2010; 2011). There, both neuronal dynamics and weight learning are derived from optimizing an objective function (as in sparse coding, Olshausen & Field, 1997). The resulting formulation is actually similar to ours in several respects, as is the underlying motivation and analysis. We became aware of this work after having developed our approach. Our work is complementary in several regards: our goal is to provide a broader perspective on how synchrony could be used in neural networks, rather than proposing one particular model; we performed a different set of experiments and conceptual analyses (for example, Rao and colleagues do not address the gating aspect of synchrony); Rao et al.'s approach relied on the model seeing only individual objects during training, which we showed to be unnecessary; and lastly, even though they applied synchrony during learning, the dataset they used for their experiments is, arguably, even simpler than our datasets. Thus, it remains to be tested whether their particular model formulation is ideal.

Finally, we are currently also exploring training synchrony networks with backpropagation. Even a feed-forward network could potentially benefit from synchrony as the latter could carry information about sensory input and network state (Geman, 2006), though complex-valued weights may be necessary for detecting synchrony patterns. Alternatively, to allow for dynamic binding by synchrony, a net-

[10] Consider for instance the tempotron neuron model (Gütig & Sompolinsky, 2006), which learns to recognize spike patterns.

[11] Of interest is also the work of Cadieu & Olshausen (2011), who use a complex-valued formulation to separate out motion and form in a generative model of natural movies. They make no connection to neuronal synchrony however.

work could be trained as recurrent network with backpropagation through time (Rumelhart et al., 1985), given appropriate input data and cost functions. In our experiments, the number of iterations required was in the order of tens or hundreds, thus making such training challenging. Again, complex-valued weights could be beneficial in establishing synchrony assemblies more rapidly.

Note: ICLR has an open review format and allows for papers to be updated. We address some issues raised by the reviewers in Appendix C.

Acknowledgements

We thank Nicolas Heess, Christopher K.I. Williams, Michael J. Frank, David A. Mély, and Ian J. Goodfellow for helpful feedback on earlier versions of this work. We would also like to thank Elie Bienenstock and Stuart Geman for insightful discussions which motived this project. This work was supported by ONR (N000141110743) and NSF early career award (IIS-1252951). DPR was supported by a fellowship within the Postdoc-Programme of the German Academic Exchange Service (DAAD). Additional support was provided by the Robert J. and Nancy D. Carney Fund for Scientific Innovation, the Brown Institute for Brain Sciences (BIBS), the Center for Vision Research (CVR) and the Center for Computation and Visualization (CCV).

Appendix

A. Supplementary figures and videos

Additional outcome examples from the various experiments are shown in Figure S1 (as referenced in main text). We also provide the following supplementary videos on the arXiv (`http://arxiv.org/abs/1312.6115`): a sample being generated from a model trained on the bars problem (`bars_sample_movie.mp4`), an example of the synchronization process in the visible and hidden layers on a bars image (`bars_synch_movie.mp4`), and several examples of visible layer synchronization for the 3-shapes and MNIST+shape datasets (`3shapes_synch_movie.mp4` and `MNIST_1_shape_synch_movie.mp4`, respectively).

B. Model and simulation parameters

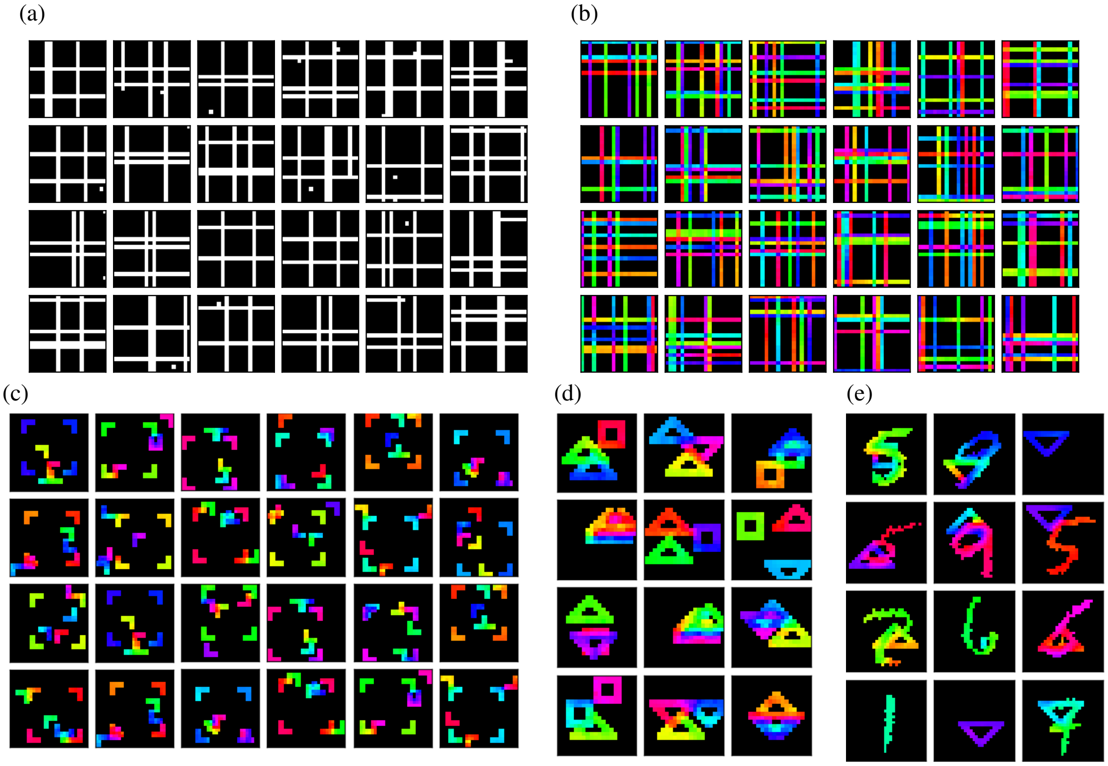
Training was implemented within the Pylearn2 framework of Goodfellow et al. (2013b). All networks were trained as real-valued deep Boltzmann machines, using layer-wise training. Layers were trained with 60 epochs of 1-step contrastive divergence (Hinton, 2002; learning rate 0.1, momentum 0.5, weight decay 10^{-4}; see Hinton, 2010, for explanation of these training aspects), with the exception of the model trained on MNIST+shape, where 5-step persistent contrastive divergence (Tieleman, 2008) was used instead (learning rate 0.005, with exponential decay factor of $1 + 1.5 \times 10^{-5}$). All datasets had 60,000 training images, and were divided into mini-batches of size 100. Biases were initialized to -4 to encourage sparse representations (for reasons discussed by Reichert et al., 2011). Initial weights were drawn randomly from a uniform distribution with support $[-0.05, 0.05]$.

The number of hidden layers, number of hidden units, and sizes of the receptive fields were varied from experiment to experiment to demonstrate various properties of neuronal synchronization in the networks (after conversion to complex values). The specific numbers were chosen mostly to be in line with earlier work and not of importance. In detail, model architectures were as follows: for the bars problem (Section 3.1), input images were 20×20, and the restricted Boltzmann machine had one hidden layer with $14 \times 14 \times 3$ units (14 height, 14 width, 3 units per location), and 7×7 receptive fields. For the corners dataset (Section 3.2), input images were 28×28, three hidden layers had $22 \times 22 \times 2$, $13 \times 13 \times 4$, and 676 units, respectively, and receptive fields were 7×7, 10×10, and 13×13 (i.e., global in the last layer). For the 3-shapes dataset (Section 3.3), input images were 20×20, hidden layer dimensions $14 \times 14 \times 3$, $8 \times 8 \times 10$, and 676, and receptive fields 7×7, 7×7, and 8×8 (global). For the MNIST+shape data (also Section 3.3), input images were 28×28, hidden layer dimensions $22 \times 22 \times 2$, $13 \times 13 \times 4$, and 676, and receptive fields 7×7, 10×10, and 13×13 (global).

For the synchronization figures, the number of steps to run was chosen so that synchronization was fairly stable at that point (100 steps was generally found to be sufficient for all models but the one trained on MNIST+shape images, where we chose 1000 steps).

Lastly, as mentioned in the main text, we note that the conversion of pretrained real-valued DBMs did not always lead to models exhibiting successful synchronization. Here, successful refers to the ability of the model to separately synchronize different objects in the input images. Unsuccessful setups resulted in either all visible units synchronizing to a single phase, or objects not synchronizing fully, across most of the images in a dataset. We found that whether or not a setup worked depended both on the dataset and the training procedures used. The presented results are representative of well performing networks.

Proper synchronization is an outcome of the right balance of excitatory and inhibitory connectivity patterns. Further analysis of how network parameters affect synchronization is the subject of ongoing work, as is incorporating synchronization during learning to achieve desired synchronization behavior.



Supplementary figure S1. **Additional results.** (a) Samples generated from a restricted Boltzmann machine trained on the bars problem. The generated images consist mostly of full-length bars. The individual receptive fields in the hidden layer were constrained to image regions of smaller extent than the bars. Thus, bars were necessarily represented in a distributed fashion. (b) - (e) Additional examples of synchronized visible units for the various datasets. The magnitudes of the visible units were set according to the binary input images (not used in training), the phases were determined by input from the hidden units. See also supplementary videos (`http://arxiv.org/abs/1312.6115`), and the main text for details.

C. Addressing issues raised by the reviewers

In the following, we summarize parts of the discussion of the ICLR review period, paraphrasing the comments of the ICLR reviewers. We expand several points that were only briefly covered in the main text.

1. In the bars experiment, some bars appear to share the same phase. Wouldn't a readout be confused and judge multiple bars to be the same object?

This is a very important issue that we are still considering. It is perhaps an issue more generally with the underlying biological theories rather than just our specific approach. As we noted in the main text, some theories pose that a limit on how many discrete objects can be represented in an oscillation cycle, without interference, explains certain capacity limits in cognition. The references we cited (Jensen & Lisman, 2005; Fell & Axmacher, 2011) refer to working memory as an example (often 4-7 items; note the number of peaks in Figure 3c—obviously this needs more quantitative analysis). We would posit that, more generally, analysis of visual scenes requiring the concurrent separation of multiple objects is limited accordingly (one might call this a prediction—or a 'postdiction'?—of our model). The question is then, how does the brain cope with this limitation? As usual in the face of perceptual capacity limits, the solution likely would involve attentional mechanisms. Such mechanisms might dynamically change the grouping of sensory inputs depending on task and context, such as whether questions are asked about individual parts and fine detail, or object groups and larger patterns. In the bars example, one might perceive the bars as a single group or texture, or focus on individual bars as capacity allows, perhaps relegating the rest of the image to a general background.

Dynamically changing phase assignments according to context, through top-down attentional input, should, in principle, be possible within the proposed framework: this is similar to grouping according to parts or wholes with top-down input, as in the experiment of Section 3.2.

2.What about the overlaps of the bars? These areas seem to be mis- or ambiguously labeled.

This is more of a problem with the task itself being ill-defined on binary images, where an overlapping pixel cannot really be meaningfully said to belong to either object alone (as there is no occlusion as such). We plan to use (representations of) real-valued images in the future.

3. What are the contributions of this paper compared to the work of Rao et al.?

As we have acknowledged, the work of Rao et al. is similar in several points (we arrived at our framework and results independently). We make additional contributions. First of all, to clarify the issue of training on multiple objects: in Rao et al.'s work, the training data consisted of a small number of fixed 8×8 pixel images (16 or less images *in total* for a dataset), containing simple patterns (one example has 4 small images with two faces instead). To demonstrate binding by synchrony, two of these patterns are superimposed during test time. We believe that going beyond this extremely constrained task, in particular showing that the binding can work when trained and tested on multiple objects, on multiple datasets including MNIST containing thousands of (if simple) images, is a valid contribution from our side. Our results also provide some insights into the nature of representations in a DBM trained on multiple objects.

Similarly, as far as we can see, Rao et al. do not discuss the gating aspect at all (Section 2.2), nor the specific issues with excitation and inhibition (Section 2.1) that we pointed out as motivation for using both classic and synchrony terms. Lastly, the following issues are addressed in our experiments only: network behavior on more than two objects; synchronization for objects that are not contiguous in the input images, as well as part vs. whole effects (Section 3.2); decoding distributed hidden representations according to phase (Section 3.3). In particular, it seems to be the case that Rao et al.'s networks had a localist (single object $\leftrightarrow$ single unit) representation in the top hidden layer in the majority of cases.

4. The introduction of phase is done in an ad-hoc way, without real justification from probabilistic goals.

We agree that framing our approach as a proper probabilistic model would be helpful (e.g. using an extension of the DUBM of Zemel et al., 1995, as discussed). At the same time, there is value to presenting the heuristic as is, based on a specific neuronal activation function, to emphasize that this idea could find application in neural networks more generally, not only those with a probabilistic interpretation or Boltzmann machines (that our approach is divorced from any one particular model is another difference when compared to Rao et al.'s work). In particular, we have performed exploratory experiments with networks trained (pretrained as real-valued nets or trained as complex-valued nets) with backpropagation, including (convolutional) feed-forward neural networks, autoencoders, or recurrent networks, as well as a biological model of lateral interactions in V1. A more rigorous mathematical and quantitative analysis is needed in any case.

5. How does running the complex-valued network relate to inference in the DBM?

We essentially use the normal DBM training as a form of pretraining for the final, complex-valued architecture. The resulting neural network is likely not exactly to be interpreted as a probabilistic model. However, if such an interpretation is desired, our understanding is that running the network could be seen as an approximation of inference in a suitably extended DUBM (by adding an off state and a classic term; refer to Zemel et al., 1995, for comparison). For our experiments, we used two procedures (with similar outcomes) in analogy to inference in a DBM: either sampling a binary output magnitude from $f()$, or letting $f()$ determine the output magnitude deterministically; the output phase was always set to the phase of the total postsynaptic input. The first procedure is similar to inference in such an extended DUBM, but, rather than sampling from a circular normal distribution on the unit circle when the unit is on, we simply take the mode of that distribution. The second procedure should qualitatively correspond to mean-field inference in an extended DUBM (see Eqs. 9 and 10 in the DUBM paper), using a slightly different output function.

6. Do phases assigned to the input change when running for more iterations than what is shown?

Phase assignments appear to be stable (see the supplementary movies), though we did not analyze this in detail. It should also be noted that the overall network is invariant to absolute phase, so only the relative phases matter.

References

Aizenberg, I. & Moraga, C. (2007). Multilayer feedforward neural network based on multi-valued neurons (MLMVN) and a backpropagation learning algorithm. *Soft Computing*, 11(2):169–183.

Behrmann, M., Zemel, R.S., & Mozer, M.C. (1998). Object-based attention and occlusion: evidence from normal participants and a computational model. *Journal of experimental psychology. Human perception and performance*, 24(4):1011–1036. PMID: 9706708.

Benchenane, K., Tiesinga, P.H., & Battaglia, F.P. (2011). Oscillations in the prefrontal cortex: a gateway to memory and attention. *Current Opinion in Neurobiology*, 21(3):475–485.

Bengio, Y., Courville, A., & Vincent, P. (2012). Representation learning: A review and new perspectives. *arXiv:1206.5538 [cs]*.

Cadieu, C.F., Hong, H., Yamins, D., Pinto, N., Majaj, N.J., & DiCarlo, J.J. (2013). The neural representation benchmark and its evaluation on brain and machine. *arXiv:1301.3530*.

Cadieu, C.F. & Olshausen, B.A. (2011). Learning intermediate-level representations of form and motion from natural movies. *Neural Computation*, 24(4):827–866.

Courville, A.C., Bergstra, J., & Bengio, Y. (2011). A spike and slab restricted Boltzmann machine. In *International Conference on Artificial Intelligence and Statistics*, p. 233–241.

Crick, F. (1984). Function of the thalamic reticular complex: the searchlight hypothesis. *Proceedings of the National Academy of Sciences*, 81(14):4586–4590. PMID: 6589612.

Duncan, J., Humphreys, G., & Ward, R. (1997). Competitive brain activity in visual attention. *Current Opinion in Neurobiology*, 7(2):255–261.

Fell, J. & Axmacher, N. (2011). The role of phase synchronization in memory processes. *Nature Reviews Neuroscience*, 12(2):105–118.

Fiori, S. (2005). Nonlinear complex-valued extensions of Hebbian learning: An essay. *Neural Computation*, 17(4):779–838.

Fries, P. (2005). A mechanism for cognitive dynamics: neuronal communication through neuronal coherence. *Trends in cognitive sciences*, 9(10):474–480. PMID: 16150631.

Földiák, P. (1990). Forming sparse representations by local anti-Hebbian learning. *Biological Cybernetics*, 64(2):165–170.

Gabernet, L., Jadhav, S.P., Feldman, D.E., Carandini, M., & Scanziani, M. (2005). Somatosensory integration controlled by dynamic thalamocortical feed-forward inhibition. *Neuron*, 48(2):315–327.

Geman, S. (2006). Invariance and selectivity in the ventral visual pathway. *Journal of Physiology-Paris*, 100(4):212–224.

Goodfellow, I.J., Courville, A., & Bengio, Y. (2013a). Joint training deep Boltzmann machines for classification. *arXiv:1301.3568*.

Goodfellow, I.J., Warde-Farley, D., Lamblin, P., Dumoulin, V., Mirza, M., Pascanu, R., Bergstra, J., Bastien, F., & Bengio, Y. (2013b). Pylearn2: a machine learning research library. *arXiv e-print 1308.4214*.

Goodman, D.F.M. & Brette, R. (2009). The brian simulator. *Frontiers in Neuroscience*, 3(2):192–197. PMID: 20011141 PMCID: PMC2751620.

Gütig, R. & Sompolinsky, H. (2006). The tempotron: a neuron that learns spike timing–based decisions. *Nature Neuroscience*, 9(3):420–428.

Hinton, G.E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Hinton, G.E. (2010). A practical guide to training restricted Boltzmann machines. *Technical report UTML TR 2010-003*, Department of Computer Science, Machine Learning Group, University of Toronto.

Hirose, A. (2011). Nature of complex number and complex-valued neural networks. *Frontiers of Electrical and Electronic Engineering in China*, 6(1):171–180.

Jensen, O. & Lisman, J.E. (2005). Hippocampal sequence-encoding driven by a cortical multi-item working memory buffer. *Trends in Neurosciences*, 28(2):67–72.

Kim, T. & Adalı, T. (2003). Approximation by fully complex multilayer perceptrons. *Neural Computation*, 15(7):1641–1666.

Kivinen, J. & Williams, C. (2011). Transformation equivariant Boltzmann machines. In T. Honkela, W. Duch, M. Girolami, & S. Kaski, eds., *Artificial Neural Networks and Machine Learning – ICANN 2011*, vol. 6791 of *Lecture Notes in Computer Science*, pp. 1–9. Springer Berlin / Heidelberg.

Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. In P. Bartlett, F.C.N. Pereira, C.J.C. Burges, L. Bottou, & K.Q. Weinberger, eds., *Advances in Neural Information Processing Systems 25*, pp. 1106–1114.

Le, Q.V., Ranzato, M., Monga, R., Devin, M., Chen, K., Corrado, G.S., Dean, J., & Ng, A.Y. (2011). Building high-level features using large scale unsupervised learning. *arXiv:1112.6209 [cs]*.

LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., & Jackel, L.D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551.

Lücke, J. & Sahani, M. (2008). Maximal causes for non-linear component extraction. *The Journal of Machine Learning Research*, 9:1227–1267.

Miconi, T. & VanRullen, R. (2010). The gamma slideshow: Object-based perceptual cycles in a model of the visual cortex. *Frontiers in Human Neuroscience*, 4. PMID: 21120147 PMCID: PMC2992033.

Miller, E.K. & Buschman, T.J. (2013). Cortical circuits for the control of attention. *Current Opinion in Neurobiology*, 23(2):216–222.

Mozer, M.C., Zemel, R.S., Behrmann, M., & Williams, C.K. (1992). Learning to segment images using dynamic feature binding. *Neural Computation*, 4(5):650–665.

Nitta, T. (2004). Orthogonality of decision boundaries in complex-valued neural networks. *Neural Computation*, 16(1):73–97.

Olshausen, B.A. & Field, D.J. (1997). Sparse coding with an overcomplete basis set: A strategy employed by V1? *Vision Research*, 37(23):3311–3325.

Rao, A.R. & Cecchi, G.A. (2010). An objective function utilizing complex sparsity for efficient segmentation in multi-layer oscillatory networks. *International Journal of Intelligent Computing and Cybernetics*, 3(2):173–206.

Rao, A. & Cecchi, G. (2011). The effects of feedback and lateral connections on perceptual processing: A study using oscillatory networks. In *IJCNN 2011*, pp. 1177–1184.

Rao, A., Cecchi, G., Peck, C., & Kozloski, J. (2008). Unsupervised segmentation with dynamical units. *IEEE Transactions on Neural Networks*, 19(1):168–182.

Ray, S. & Maunsell, J.H.R. (2010). Differences in gamma frequencies across visual cortex restrict their possible use in computation. *Neuron*, 67(5):885–896.

Reichert, D., Seriès, P., & Storkey, A. (2010). Hallucinations in Charles Bonnet syndrome induced by homeostasis: a deep Boltzmann machine model. In J. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, & A. Culotta, eds., *Advances in Neural Information Processing Systems 23*, pp. 2020–2028.

Reichert, D.P. (2012). Deep Boltzmann machines as hierarchical generative models of perceptual inference in the cortex. PhD thesis, University of Edinburgh, Edinburgh, UK.

Reichert, D.P., Seriès, P., & Storkey, A.J. (2011). A hierarchical generative model of recurrent object-based attention in the visual cortex. In T. Honkela, W. Duch, M. Girolami, & S. Kaski, eds., *Artificial Neural Networks and Machine Learning - ICANN 2011*, vol. 6791, pp. 18–25. Springer Berlin Heidelberg, Berlin, Heidelberg.

Reichert, D.P., Seriès, P., & Storkey, A.J. (2013). Charles Bonnet syndrome: Evidence for a generative model in the cortex? *PLoS Comput Biol*, 9(7):e1003134.

Rumelhart, D.E., Hinton, G.E., & Williams, R.J. (1985). Learning internal representations by error propagation. *Tech. rep.*

Saalmann, Y.B., Pinsk, M.A., Wang, L., Li, X., & Kastner, S. (2012). The pulvinar regulates information transmission between cortical areas based on attention demands. *Science*, 337(6095):753–756.

Salakhutdinov, R. & Hinton, G. (2009). Deep Boltzmann machines. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, vol. 5, pp. 448–455.

Savitha, R., Suresh, S., & Sundararajan, N. (2011). Metacognitive learning in a fully complex-valued radial basis function neural network. *Neural Computation*, 24(5):1297–1328.

Serre, T., Oliva, A., & Poggio, T. (2007). A feedforward architecture accounts for rapid categorization. *Proceedings of the National Academy of Sciences*, 104(15):6424–6429.

Shadlen, M.N. & Movshon, J.A. (1999). Synchrony unbound: a critical evaluation of the temporal binding hypothesis. *Neuron*, 24(1):67–77.

Shipp, S. (2003). The functional logic of cortico–pulvinar connections. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1438):1605–1624. PMID: 14561322.

Singer, W. & Gray, C.M. (1995). Visual feature integration and the temporal correlation hypothesis. *Annual review of neuroscience*, 18:555–586. PMID: 7605074.

Singer, W. (2007). Binding by synchrony. *Scholarpedia*, 2(12):1657.

Sjöström, J. & Gerstner, W. (2010). Spike-timing dependent plasticity. *Scholarpedia*, 5(2):1362.

Smolensky, P. (1986). Information processing in dynamical systems: foundations of harmony theory. In *Parallel distributed processing: explorations in the microstructure of cognition. Vol. 1. Foundations*, p. 194–281. MIT Press, Cambridge, MA.

Spratling, M.W. (2011). Unsupervised learning of generative and discriminative weights encoding elementary image components in a predictive coding model of cortical function. *Neural Computation*, 24(1):60–103.

Stanley, G.B. (2013). Reading and writing the neural code. *Nature Neuroscience*, 16(3):259–263.

Tieleman, T. (2008). Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th Annual International Conference on Machine Learning*, pp. 1064–1071. Helsinki, Finland.

Uhlhaas, P.J., Pipa, G., Lima, B., Melloni, L., Neuenschwander, S., Nikolić, D., & Singer, W. (2009). Neural synchrony in cortical networks: history, concept and current status. *Frontiers in Integrative Neuroscience*, 3:17.

von der Malsburg, C. (1981). The correlation theory of brain function. *Tech. Rep. S1-2*, Department of Neurobiology, MPI for Biophysical Chemistry, Goettingen, W.-Germany.

Weber, C. & Wermter, S. (2005). Image segmentation by complex-valued units. In W. Duch, J. Kacprzyk, E. Oja, & S. Zadrożny, eds., *Artificial Neural Networks: Biological Inspirations – ICANN 2005*, no. 3696 in Lecture Notes in Computer Science, pp. 519–524. Springer Berlin Heidelberg.

Zeiler, M.D. & Fergus, R. (2013). Visualizing and understanding convolutional networks. *arXiv:1311.2901 [cs]*.

Zemel, R.S., Williams, C.K., & Mozer, M.C. (1995). Lending direction to neural networks. *Neural Networks*, 8(4):503–512.