frontiers in
PSYCHOLOGY

ORIGINAL RESEARCH ARTICLE
published: 01 April 2013
doi: 10.3389/fpsyg.2013.00124

Recurrent processing during object recognition

Randall C. O'Reilly[1,2]*†, Dean Wyatte[1]*†, Seth Herd[1], Brian Mingus[1] and David J. Jilk[2]

[1] Department of Psychology and Neuroscience, University of Colorado Boulder, Boulder, CO, USA
[2] eCortex, Inc., Boulder, CO, USA

Edited by:
Michael J. Tarr, Carnegie Mellon University, USA

Reviewed by:
Rosemary A. Cowell, University of California San Diego, USA
Maximilian Riesenhuber, Georgetown University Medical Center, USA

***Correspondence:**
Randall C. O'Reilly and Dean Wyatte, Department of Psychology and Neuroscience, University of Colorado Boulder, Boulder, CO, USA.
e-mail: randy.oreilly@colorado.edu; dean.wyatte@colorado.edu

† Randall C. O'Reilly and Dean Wyatte have contributed equally to this work.

How does the brain learn to recognize objects visually, and perform this difficult feat robustly in the face of many sources of ambiguity and variability? We present a computational model based on the biology of the relevant visual pathways that learns to reliably recognize 100 different object categories in the face of naturally occurring variability in location, rotation, size, and lighting. The model exhibits robustness to highly ambiguous, partially occluded inputs. Both the unified, biologically plausible learning mechanism and the robustness to occlusion derive from the role that recurrent connectivity and recurrent processing mechanisms play in the model. Furthermore, this interaction of recurrent connectivity and learning predicts that high-level visual representations should be shaped by error signals from nearby, associated brain areas over the course of visual learning. Consistent with this prediction, we show how semantic knowledge about object categories changes the nature of their learned visual representations, as well as how this representational shift supports the mapping between perceptual and conceptual knowledge. Altogether, these findings support the potential importance of ongoing recurrent processing throughout the brain's visual system and suggest ways in which object recognition can be understood in terms of interactions within and between processes over time.

Keywords: object recognition, computational model, recurrent processing, feedback, winners-take-all mechanism

INTRODUCTION

One of the most salient features of the mammalian neocortex is the structure of its connectivity, which provides for many forms of *recurrent* processing, where neurons mutually influence each other through direct, bidirectional interactions. There are extensive bidirectional excitatory and inhibitory connections within individual cortical areas, and almost invariably, every area that receives afferent synapses from another area, also sends back efferent synapses in return (Felleman and Van Essen, 1991; Scannell et al., 1995; Sporns and Zwi, 2004; Sporns et al., 2007). We describe an explicit computational model (*LVis* – Leabra Vision) of the function of this recurrent architecture in the context of visual object recognition, demonstrating a synergy between the learning and processing benefits of recurrent connectivity.

Recurrent processing, for example, has been suggested to be critical for solving certain visual tasks such as figure-ground segmentation (Hupe et al., 1998; Roelfsema et al., 2002; Lamme and Roelfsema, 2000), which requires integration of information from outside the classical receptive field. We demonstrate how recurrent excitatory processing could provide a similar function in visual occlusion, which requires the organization of image fragments that span multiple receptive fields into a logical whole *Gestalt* and involves the filling-in of missing visual information (Kourtzi and Kanwisher, 2001; Lerner et al., 2002; Rauschenberger et al., 2006; Weigelt et al., 2007; Wyatte et al., 2012a).

At a more local level, recurrent inhibitory processing produces sparse distributed representations, implemented in LVis through the use of a k-Winners-Take-All (kWTA) inhibition function (where k represents the roughly 15–25% activity levels present in neocortical networks; O'Reilly, 1998; O'Reilly and Munakata, 2000; O'Reilly et al., 2012). The sparse distributed representations produced by these recurrent inhibitory dynamics have been shown to produce biologically realistic representations in response to natural stimuli (e.g., O'Reilly and Munakata, 2000; Olshausen and Field, 2004; O'Reilly et al., 2012). We show here that inhibitory recurrent dynamics and sparse distributed representations make our model more robust in the face of ambiguity, by testing recognition performance with occluded visual inputs.

In the non-human primate neuroanatomy, object recognition involves the flow of visual information through the ventral pathway, originating in primary visual cortex (V1), continuing through extrastriate areas (V2, V4), and terminating in inferotemporal (IT) cortex (Hubel and Wiesel, 1962; Van Essen et al., 1992; Ungerleider and Haxby, 1994). IT neurons exhibit robust object-level encoding over wide ranges of position, rotation, scale, and lighting variability (Logothetis et al., 1995; Tanaka, 1996; Riesenhuber and Poggio, 2002; Rolls and Stringer, 2006; Tompa and Sary, 2010; DiCarlo et al., 2012). Object recognition in the human cortex operates in a similar hierarchical fashion, with homologous object-selective regions distributed throughout the lateral occipital cortex (LOC) (Grill-Spector et al., 2001; Orban et al., 2004; Kriegeskorte et al., 2008).

Computational models of object recognition that implement a feedforward, hierarchical version of the ventral pathway have explained many aspects of the initial neural response properties across these different brain areas (Fukushima, 1980, 2003; Wallis and Rolls, 1997; Riesenhuber and Poggio, 1999; Masquelier and Thorpe, 2007). Furthermore, when coupled with a supervised

learning procedure (e.g., support vector machines), these models perform well at challenging computational tests of object recognition (Fei-Fei et al., 2007; Serre et al., 2007c; Mutch and Lowe, 2008; Pinto et al., 2009). Thus, they establish that primarily feedforward-driven neural responses properties based on the initial responses of the ventral pathway are *sufficient* to solve reasonably challenging versions of the object recognition problem (Serre et al., 2007a,b; DiCarlo et al., 2012).

The LVis model builds upon this feedforward processing foundation, and learns a very similar hierarchical solution to the object recognition problem. In our tests on 100-way object classification with reasonable levels of variability in location, rotation, size, and lighting, LVis performs in the same general range as these established feedforward models. Interestingly, it does so using a single unified, biologically based learning mechanism that leverages bidirectional recurrent processing between layers, to enable signals from other modalities and brain areas to shape visual object recognition during learning in important ways, supporting a form of error-driven learning (O'Reilly, 1996; O'Reilly and Munakata, 2000; O'Reilly et al., 2012). Error-driven learning is almost certainly essential for solving hard computational problems (O'Reilly and Munakata, 2000; Hinton and Salakhutdinov, 2006), and is a central element in all of the above high performance object recognition systems at the supervised learning stage. Furthermore, there are indications that error-driven learning is actually doing most of the work in object recognition models, as good performance is possible even with random visual filters (Jarrett et al., 2009).

The recurrent connectivity in our LVis model leads to a clear prediction: representations in other brain areas that project into the object recognition pathway should shape the way it develops through learning. Recent evidence indeed suggests that neurons in IT cortex reflect significant higher-level "semantic" influences, in addition to the expected stimulus-driven similarities among objects (Kiani et al., 2007; Kriegeskorte et al., 2008; Mahon and Caramazza, 2011). We show that recurrent processing within our model provides a satisfying account of this data. Furthermore, we show how recurrent processing provides a mechanism via which this higher-level semantic information can be integrated with visual information during object processing (Lupyan and Spivey, 2008; Lupyan et al., 2010; Lupyan, 2012), providing a mapping between perceptual and conceptual representations (Gotts et al., 2011).

Altogether, we argue that this model provides an integration of diverse sources of data on the object recognition system and shows how a small, unified set of biological mechanisms can potentially solve one of the most difficult and important computational problems that the brain is known to solve (Marr, 1982; Pinto et al., 2008). Our recurrent model (**Figure 1**) embodies these ideas, and provides one way of extending our understanding of object recognition beyond the initial, feedforward-driven responses.

Despite the multiple influences of recurrent processing cited above, it also might not confer performance advantages in all object recognition tasks. For example, objects presented isolated and intact, without any source of degradation or ambiguity could reasonably be resolved via feedforward processing. And indeed, recurrent processing during relatively simple tasks has actually been shown to incur small costs in raw performance, because small errors in processing can become magnified over the course of repeated recurrent interactions (O'Reilly, 2001). These small costs, however, can pay dividends in more difficult object recognition problems involving occlusion or generalization across non-visual, semantic dimensions such as during semantic inference.

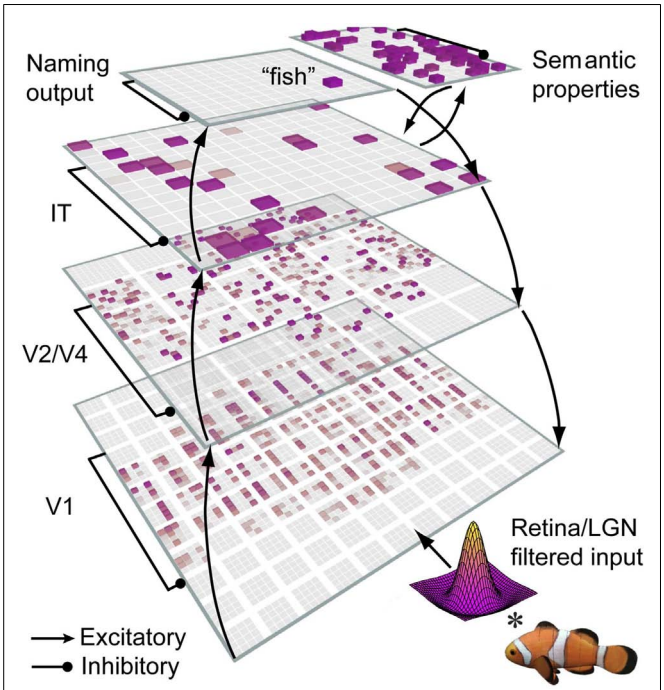

FIGURE 1 | Architecture of the LVis model. The LVis model is based on the anatomy of the ventral pathway of the brain, from primary visual cortex (V1) through extrastriate areas (V2, V4) to inferotemporal (IT) cortex. V1 reflects filters that model the response properties of V1 neurons (both simple and complex subtypes). In higher-levels, receptive fields become more spatially invariant and complex, reflecting organizational influence from non-visual properties like semantics. All layers are reciprocally connected, allowing higher-level information to influence bottom-up processing during both the initial learning and subsequent recognition of objects, and contain local, recurrent inhibitory dynamics that limit activity levels across layers.

In short, our model provides a possible synthesis in the debate about the relative contributions of feedforward and recurrent processing in vision (Lamme and Roelfsema, 2000; Kveraga et al., 2007; Vanrullen, 2007; Roland, 2010). For well-learned, unambiguous stimuli, object recognition can operate rapidly in a feedforward-dominant manner, consistent with rapid visual processing in some experiments (Thorpe et al., 1996; VanRullen and Koch, 2003; Liu et al., 2009). This feedforward-dominant processing can be observed directly in the dynamics of our model as we show below. However, the extensive recurrent connectivity found throughout the ventral pathway can also play an important function in forming robust representations needed for more complex object recognition problems that involve ambiguity, such as when objects are occluded. This translates to longer overall latencies for the recognition decision, but with the added benefit of a coherent and robust interpretation of a visual scene that arises from the integration of signals at different levels of the hierarchy (Lamme and Roelfsema, 2000; Kveraga et al., 2007; Roland, 2010).

RESULTS

OBJECT RECOGNITION DATASET

Before exploring the ways in which recurrent processing impacts the dynamics of object recognition, we briefly describe the basic set of objects on which the network was trained and tested, which we call the *CU3D-100* dataset[1]. CU3D-100 is organized into 100 categories with an average of 9–10 exemplars per category and controlled variability in pose and illumination (**Figures 2A–D**). The dataset was designed to address problems with existing datasets based on naturalistic images, such as the Caltech101 (Ponce et al., 2006; Pinto et al., 2008). Naturalistic image datasets, while useful for benchmarking the ability of object recognition systems on realistic visual stimuli, are often underconstrained for studying biological principles of object recognition such as invariance or the recurrent processing effects that are of interest here. This is because object exemplars are often present in a fixed pose and with additional background clutter that is can be correlated with the object's category, and foreground and background image elements cannot be independently manipulated. The CU3D-100 dataset, in contrast, uses a "synthetic" approach in which object models and backgrounds *can* be controlled independently and then rendered to bitmap images, allowing an experimenter to isolate and gain full control over the parameters that govern the core challenge of the object recognition problem (Pinto et al., 2008, 2009, 2011; DiCarlo et al., 2012). Datasets that use 3D models are gaining popularity in the literature, but are labor-intensive to create, and thus usually only consist of a handful of object categories and exemplars (e.g., LeCun et al., 2004). To our knowledge, this is the first synthetic dataset that approaches the size and scope of larger benchmark datasets like Caltech101.

For the purposes of the present research, we rendered the object models against uniform backgrounds as opposed to cluttered backgrounds. Although background clutter is clearly more relevant for real-world applications of object recognition, we think that it is not realistic from a biological perspective to assume

[1]http://cu3d.colorado.edu

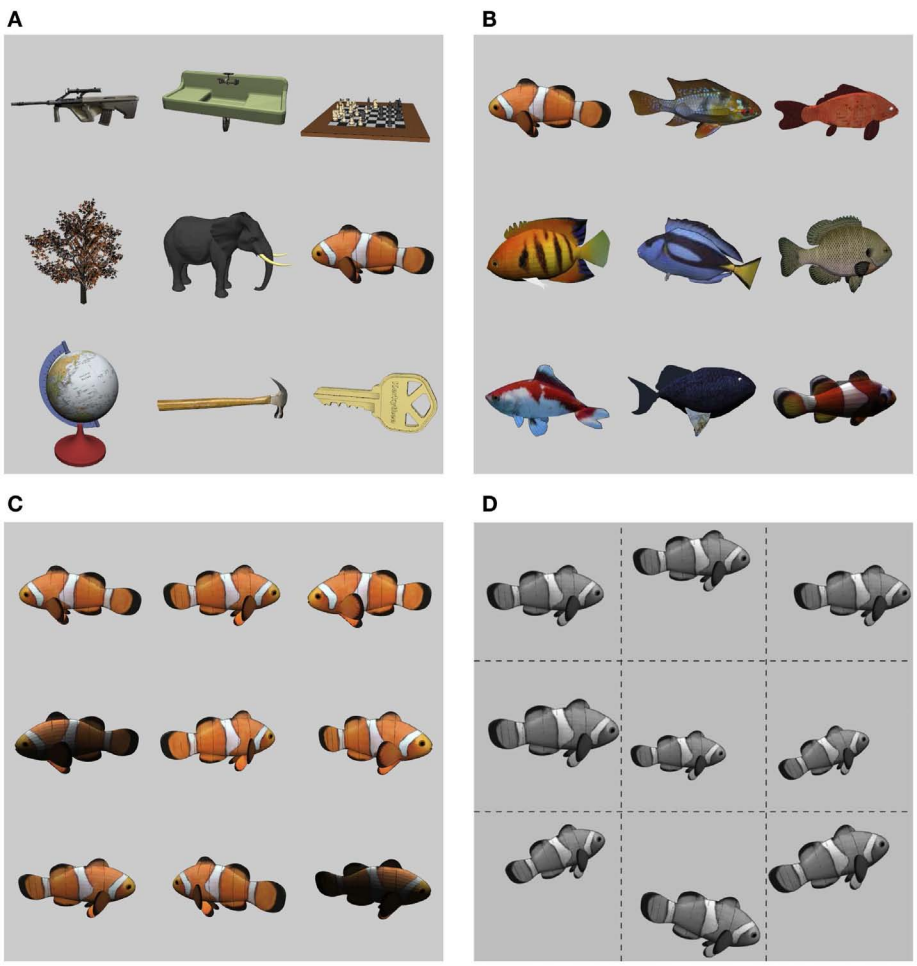

FIGURE 2 | The *CU3D-100* dataset. (A) Nine example objects from the 100 CU3D categories. **(B)** Each category is further composed of multiple, diverse exemplars (average of 9.42 exemplars per category). **(C)** Each exemplar is rendered with 3D (depth) rotations and variability in lighting. **(D)** In training and testing the models described here, the 2D images were converted to grayscale and subjected to 2D transformations (translation, scale, planar rotation), with ranges generally around 20%.

that the upper levels of the ventral visual pathway (V4 and IT) have to contend with the full impact of this background clutter. This is because extensive research has indicated that early levels of the visual pathway, specifically in area V2, contain specialized figure-ground processing mechanisms that perform border ownership labeling (Zhaoping, 2005; Craft et al., 2007; Poort et al., 2012). Thus, features belonging to the background are not grouped with those associated with the foreground object, and this filtering process enables higher-level areas to perform spatial and featural integration processes without suffering as much interference from irrelevant background features as would otherwise be the case in a model lacking these figure-ground filtering mechanisms. Consistent with this perspective, various sources of data indicate that IT represents relevant objects without significant interference from irrelevant background clutter (Baylis and Driver, 2001; Kourtzi and Kanwisher, 2001; Lerner et al., 2002).

Thus, our goal with the present simulations was to enable the model to achieve high levels of performance (i.e., above the 90% generalization level) in the face of substantial levels of input variability, thus isolating the core challenge of object invariance without introducing confounding sources of performance-degrading factors such as background clutter. When models fail to recognize realistic images containing clutter (performance typically plateaus around 60–70%), one can never quite be sure whether the model is simply not very good, or whether it actually might be a very good model when given the benefit of figure-ground filtering that we think the biological system enjoys. Given the performance-based validation of our model on the core object recognition problem, we can then incrementally "ratchet up" the difficulty of the problem to explore how manipulations along different dimensions, like the occlusion (described in this paper) or background clutter (the subject of ongoing research to be described in a subsequent paper) affect performance.

We rendered objects with ±20° in-depth (3D) rotations (including a random 180° left-right flip for objects that are asymmetric along this dimension), and overhead lighting positioned uniformly randomly along an 80° overhead arc, to generate considerable lighting variability. Rendered images were then presented to our model with in-plane (2D) transformations of 30% translation, 20% size scaling, and 14° in-plane rotations. We assessed baseline performance of our model by reserving two exemplars per category for testing, and using the rest for training (results reflect averages over 10 random train/test splits). To capture an observer's ability to make multiple fixations on an object, which can be used in an aggregate manner during the recognition process (Ratcliff, 1978; Bradski and Grossberg, 1995; Ratcliff and McKoon, 2008), we also examined the performance benefits that result from aggregating (majority voting) outputs over transformations of the images (see Methods for details).

The mean recognition rate on novel test items for the LVis model was 92.2% with the highest level of majority voting, which is well above the chance level of 1% for 100-way simultaneous discrimination, and indicates that the network is capable of performing quite well at the basic task of recognizing a large number of object categories in the face of extensive variability in the input images. With no voting, the generalization performance was 79.6%, and with 2D-only voting it was 86.5%.

We also developed two other comparison networks that have the same architecture as the LVis model, but lack recurrent processing mechanisms, which are used to assess the comparative impact of recurrent processing. These models used standard purely feedforward backpropagation learning (Rumelhart et al., 1986) – the error-driven learning in the Leabra model is a mathematical approximation of that in backpropagation (O'Reilly, 1996), so this is the most reasonable point of comparison for a purely feedforward network. The first backpropagation network (*Bp Distrib*) used standard parameters (i.e., 0 mean weights with 0.5 uniform random variability, learning rate of 0.01), which provided an unbiased starting point for learning and ended up producing highly distributed representations across the hidden layers, as is typical for backpropagation networks. Its performance on the object recognition test was slightly worse than the LVis model, obtaining 88.6% correct with full majority voting, 82.4% with 2D-only voting, and 77% with no voting. The second backpropagation network (*Bp Sparse*) attempted to capture the ability of the LVis model to develop relatively sparse representations due to the presence of recurrent inhibitory competition within its layers (O'Reilly, 1998). We hypothesized that strong negative initial bias weights (−3.0) and inputs that were pre-processed with the same kWTA inhibitory competition as used in the LVis inputs, would produce sparse patterns of activity across all layers and drive learning in a more robust manner. This sparse parameterization improved the performance of the backpropagation network significantly, resulting in 94.6% correct with full majority voting, 90.7% with 2D-only voting, and 86.53% with no voting. Overall, this level of performance was comparable to other standard feedforward object recognition models on this dataset, as will be reported in another publication.

RECURRENT PROCESSING UNDER OCCLUSION

Our first test of the role of recurrent processing in object recognition focuses on the case of partial occlusion of images. To algorithmically and parametrically manipulate occlusion in an automated fashion, we use a method similar to the "Bubbles" approach (Gosselin and Schyns, 2001) in which selected portions of an image are spatially masked via filtering operations. Specifically, we partially occluded portions of object images with varying numbers of randomly positioned circular "blob" filters softened with a Gaussian blur around the edges (**Figure 3**). This minimizes the introduction of novel edge artifacts, which is important given that the model does not have figure-ground mechanisms that code the ownership of each edge as belonging to the target object or the occluder (e.g., Zhaoping, 2005; Craft et al., 2007). Thus, this manipulation tests the ability to complete an underspecified input signal – which the brain undoubtedly does during occluded object recognition (Kourtzi and Kanwisher, 2001; Lerner et al., 2002; Rauschenberger et al., 2006; Weigelt et al., 2007; Wyatte et al., 2012a) – but without interference from features belonging to the occluder. This assumes there is at least partial separability of the border ownership coding and grouping- or completion-related processing, which has been suggested to be the case in the figure-ground segregation literature (Poort et al., 2012; Scholte et al., 2008). While V1- and V2-level mechanisms such as those related to illusory contour perception (Lee and Nguyen, 2001; Seghier and Vuilleumier, 2006; see

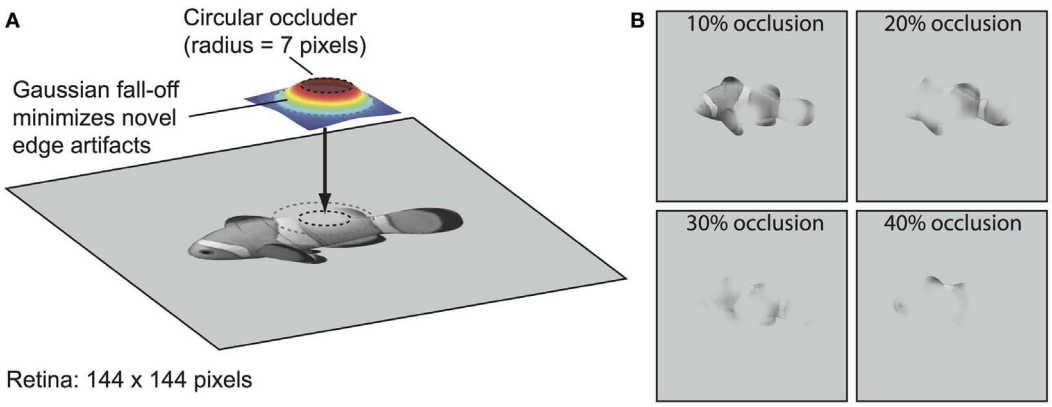

FIGURE 3 | Blob-based occlusion. (A) Images were occluded by applying a filter that was set to 1.0 within a circle of radius 5% of the image size (i.e., 5% of 144 pixels or 7 pixels) and then fell off outside the circle as a Gaussian function. The final effective size of the filter was 42 × 42 pixels. The filter was used as a two-dimensional weighting function between the object and the background gray level such that image regions that fell within the circle region at the top of the filter were completely occluded with the background gray level. **(B)** Examples of different occlusion levels. Percent occlusion parameterized an equation that specified the number of times to apply the filter (see Methods). Additional occlusion examples are shown in S4.

also Biederman and Cooper, 1991) could potentially assist with filling-in parts of the occluded objects, with higher-levels of occlusion, there is enough visual information missing that lower-level continuation-based mechanisms would likely fail to add much. A comprehensive model of the early levels of visual processing in V1 and V2 that includes border ownership coding and illusory contour continuation would be necessary to determine the relative contribution of each of these mechanisms with realistic visual occlusion, but we argue that our methods provide a reasonable approximation for the impact of naturally occurring forms of occlusion on the upper levels of the visual pathway (e.g., V4 and IT), which are the focus of the present research.

To directly measure the impact of recurrent processing in the LVis model for these partially occluded images, we assessed the extent to which the network was able to reconstruct a more complete representation of the occluded image (**Figure 4**). For each cycle of network activation updating during the processing of a given input image, we computed the cosine (normalized inner product) of the activity in each layer of the network compared to the final activity state of each such layer for that object when the object was unoccluded. Thus, this analysis reveals the extent to which the network is able to reconstruct over cycles of processing an internal representation that effectively fills-in the occluded parts of the image, based on prior learning about the object. To determine the role of recurrent processing in this process of reconstruction, we compared the standard LVis model with one where the strength of the top-down connections was reset to zero, thus removing the ability of higher-level representations to feed back and provide top-down knowledge of object properties based on prior learning. However, this comparison model still benefits from inhibitory recurrent processing, which we will see later plays a critical role in enhancing robustness to occlusion.

As **Figure 4** shows, the recurrent connections play an important role in filling-in missing visual information, with their effect being greatest in magnitude when images are highly occluded (e.g., 50% occlusion). The IT layer in our model almost universally produces a complete object representation, with smaller completion effects observable in extrastriate layers. This finding is in accordance with object completion effects described in the literature, which indicate that their effects are largest in higher-level visual areas (e.g., IT, LOC), thus representing the *perceived* object, with lower-level areas representing mainly visual information that is present in the stimulus itself (Rauschenberger et al., 2006; Weigelt et al., 2007).

Next, we address the question of whether this recurrent filling-in process can actually lead to better recognition performance for occluded objects. In **Figure 5**, we see some indication of an advantage from the LVis networks over the backpropagation networks, especially in the case of the Bp Distrib network, which suffers dramatically from the effects of occlusion. The Bp Sparse network holds up much better, and an advantage for the LVis model is only observed for the higher-levels of occlusion, where it does become quite substantial on a percentage basis.

Given the differences in level of top-down filling-in for the intact LVis model relative to the one without top-down feedback connections, we initially expected to also see this difference reflected in the overall level of performance of these two networks. However, no such difference is evident in the results, which we have validated in multiple ways. To explain this puzzling result, it is important to ask whether in general a top-down signal can be more accurate than the bottom-up signal that activates it in the first place. Specifically, in the absence of other sources of information (e.g., from other modalities or prior context), the higher-levels of the network depend upon an initial feedforward wave of activation for their initial pattern of activity, and it is this activity pattern that then is sent back down to the lower-levels to support further filling-in. But if the initial feedforward activation is incorrect, this would presumably result in an incorrect top-down signal that would support the wrong bottom-up interpretation of the image, and thus reinforce this incorrect interpretation further. In other words, top-down support can be a double-edged sword that cuts both ways, and by recognizing this, we can understand why it does not produce a net increase in overall recognition accuracy.

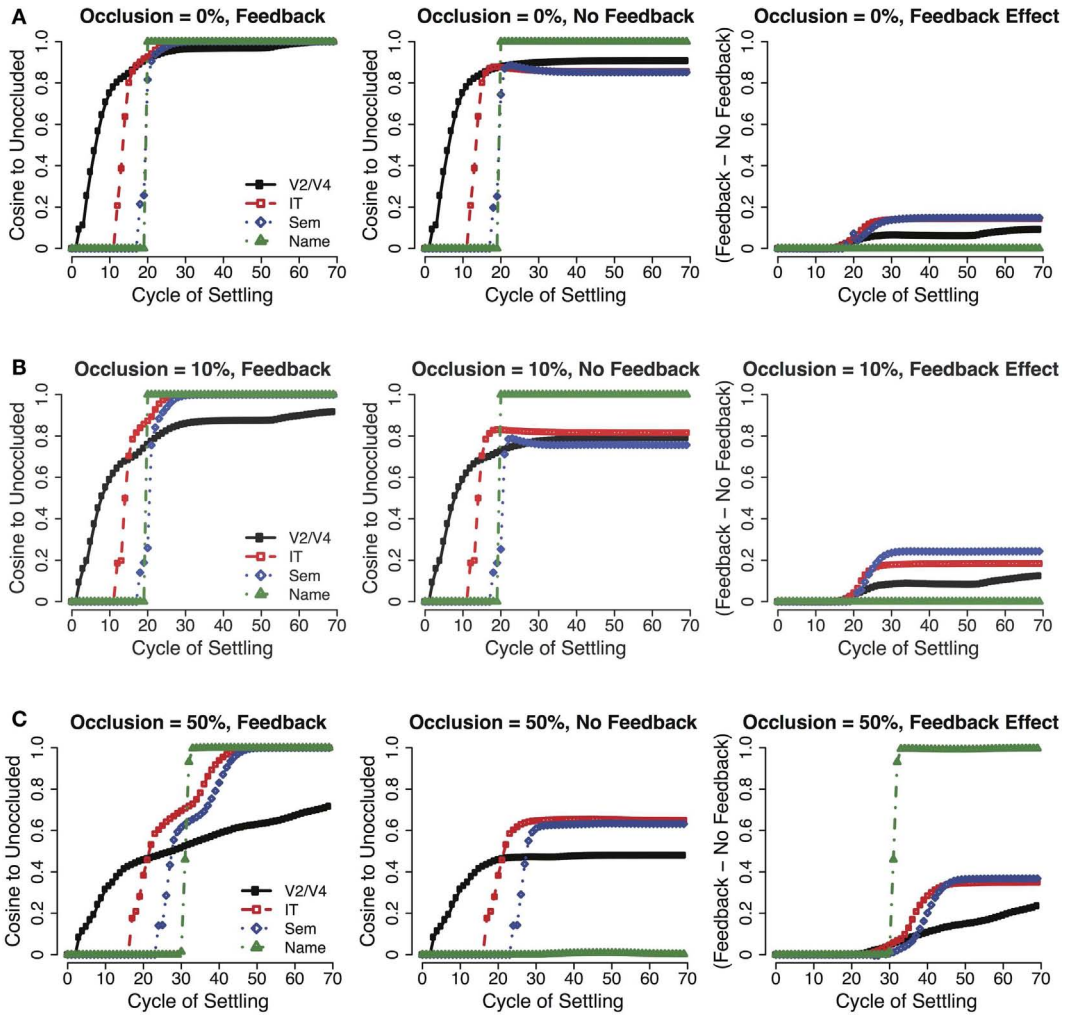

FIGURE 4 | Recurrent interactions between adjacent layers during cycles of updating for 0, 10, and 50% occlusion cases of an object. By computing the cosine of the activity pattern for each layer compared to what would be expected when processing an unoccluded object, the network interactions that give rise to the named output can be observed. **(A,B)** When inputs are relatively unambiguous, the network converges rapidly with only a short latency between the first IT responses and activation of the correct output (ca. 10 cycles). **(C)** The correct output can still be resolved when inputs are highly ambiguous, but only after considerable recurrent interactions between layers that serve to fill in missing information reinforce the overall network state. In this case, the latency between the first IT responses and activation of the correct output is longer (ca. 15 cycles), in accordance with the recurrent interactions between layers, which take time to stabilize. Also note that the V2/V4 state does not fully complete, but the IT and Semantics patterns are identical to the unoccluded case, indicating that the higher-levels of the network complete, while the lower-levels do not ("amodal completion"). Recurrent excitatory feedback plays a critical role in this completion effect, as is shown in comparison with a network having no top-down feedback weights – this effect is more apparent with higher-levels of occlusion.

To explain why the LVis model without top-down feedback connections also performs better than the Bp Sparse network at these higher occlusion levels, we attribute the advantages to the inhibitory competition present in the LVis networks that extends beyond the initial responses within a given layer. This form of recurrent inhibition dynamically adjusts to the level of excitation coming into a given layer, and thus in the highly occluded cases the inhibitory level can decrease correspondingly, enabling more activity overall to propagate through the network. In contrast, the strong negative bias weights that give rise to the sparse activities in the Bp Sparse network are in effect prior to the first responses, and thus may result in under-activation of the units for high levels of occlusion. Thus, we find evidence for the importance of recurrent inhibitory competition in providing dynamic renormalization of network response over a wide range of input signal strengths (Carandini and Heeger, 2012).

Taken together, these results show that both of the major forms of recurrence present in the LVis model can have important functional benefits: the top-down excitatory connectivity from higher areas supports filling-in of missing information compared to a network without this top-down recurrence. This could be important for many different cognitive tasks, where the missing information would be useful. However, absent other more informative sources of input, this top-down recurrence does not result

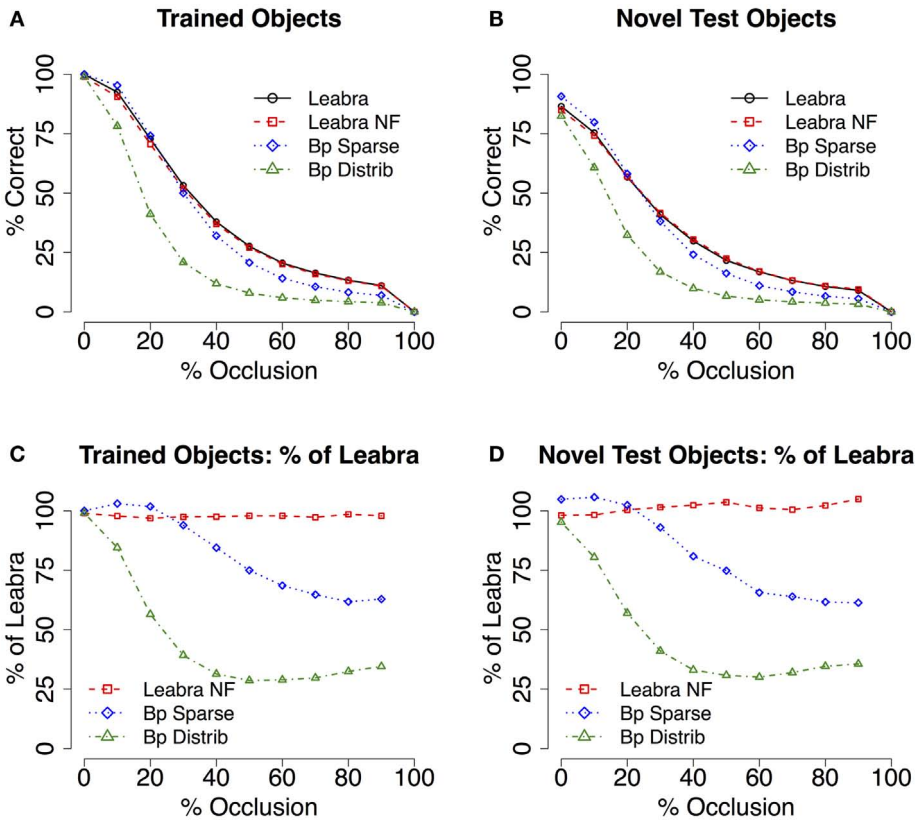

FIGURE 5 | Test of recognition under partial occlusion conditions. (A) Mean recognition performance (with 2D voting – see methods and supplemental material for raw results) for trained objects, comparing full recurrent processing in Leabra with and without feedback (Leabra NF = no feedback) and purely feedforward backpropagation (Bp Sparse = sparse parameters, Bp Distrib = distributed parameters). Recurrent processing in Leabra facilitates robust recognition under partial occlusion. The Leabra model without feedback performs equivalently, suggesting that it is specifically inhibitory processing that explains this robustness. **(B)** Mean recognition for novel test objects, comparing between the same models as A. The advantage of Leabra's recurrent connectivity is similarly apparent during generalization. **(C,D)** Results as a percentage of the Leabra performance – the slope of the lines in A and B masks the substantial effect sizes present – For trained objects, Bp Sparse performs as low as 66% compared to Leabra, and Bp Distrib as low as 31%. Again, results were qualitatively similar for novel test objects.

in an overall improvement in recognition accuracy. Nevertheless, here we do see the advantage of the inhibitory recurrent dynamics, for renormalizing activations in the face of weaker occluded inputs.

RECURRENT CONNECTIVITY AND LEARNED OBJECT REPRESENTATIONS

Another prediction from the recurrent connectivity of our model is that top-down signals should shape lower-level representations. For example, Kriegeskorte et al. (2008) showed that visual representations in inferotemporal (IT) cortex reflect semantic influences, for example, a distinction between living and non-living items. Importantly, this organizational property of IT cortex was unable to be explained in terms of bottom-up visual similarities, and was further unaccounted for by various feedforward models including those that learn "IT-level" visual features (Kiani et al., 2007). Other areas in the ventral pathway have also been shown to reflect action-based representations, possibly due to interactions with dorsal areas associated with object manipulation and tool use (Culham and Valyear, 2006; Mahon et al., 2007; Almeida et al., 2010; Mahon and Caramazza, 2011). Other evidence for top-down influences from prefrontal cortex to IT have been found during delayed responding categorization tasks (Freedman et al., 2003).

We hypothesized that these non-classical organizational properties of IT cortex are due to constraints imposed by recurrent connectivity with other neural systems over the course of learning. Simply put, recurrent connectivity allows error-driven learning signals about object properties to be circulated between neural systems, causing the similarity structure of non-visual systems to be reflected in visual areas. Semantic relationships between object categories have been suggested to be maintained by the anterior temporal pole (Patterson et al., 2007), which sends descending feedback to high-level ventral areas, and is thus a candidate structure responsible for the semantic organization observed in IT responses.

We were able to demonstrate these ideas in our model by providing top-down semantic inputs to the IT layer (**Figure 6A**), with a similarity structure derived from pairwise similarities for each of the 100 object categories obtained from latent semantic analysis (LSA; Landauer and Dumais, 1997). **Figure 6A** shows that the IT layer of our model comes to reflect this semantic

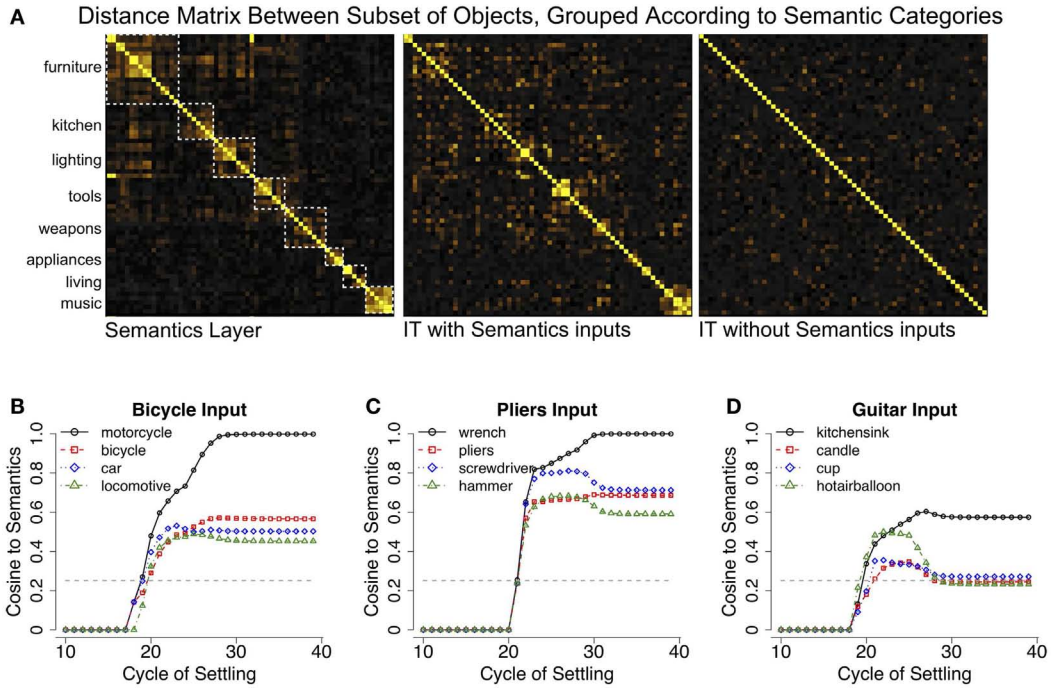

FIGURE 6 | Semantic effects in LVis. (A) Top-down semantic influences on inferotemporal (IT) cortex representations in the model, in terms of distance matrix plots showing the normalized dot product (cosine) distance between semantic or IT representations (yellow = more similar). The semantics contain a categorical structure (intuitive categories indicated by dotted white squares) with some hierarchical organization, for example, among furniture, kitchen, lighting, and tools. The IT layer with semantic influences reflects a blend of these semantics and bottom-up visual similarities. The correlation between the IT layer with semantics and the actual semantics is 0.72, IT layer without semantics and the semantics is 0.57, and between the IT layers with and without semantics is 0.79. **(B)** Trajectory of the Semantics layer when a bicycle image was presented to a network that was not trained on bicycles, showing cosine similarities of the current semantics activation pattern to the canonical semantics for indicated categories. The network interprets the bicycle as a motorcycle (closest trained category), but the semantics layer representation actually has bicycle as its second closest pattern, indicating that it can infer veridical semantic properties from visual appearance. The dotted gray line indicates the mean similarity of the input semantics to the semantics of all other categories, which was 0.25 for the categories tested here. **(C)** Similar results for a pliers image, which was also not trained. **(D)** Guitars did not exhibit obvious visual similarity to semantically related trained items, and thus, the model was unable to infer their semantic properties.

structure, as a result of influences from the top-down projections from semantic representations to IT. Importantly, learned object representations remain relatively distinct, and object recognition performance is unaffected. Thus, recurrent processing allows the visual properties of objects and non-visual semantic properties to be concurrently represented in the same neural substrate by simultaneously satisfying multiple bottom-up and top-down constraints during learning.

In addition to enabling our model to capture this important data, the shaping of IT representations according to semantic structure enables the model to bidirectionally map between purely visual and purely semantic similarity spaces (Gotts et al., 2011). Importantly, semantic similarity spaces have been shown to be distinctively non-visual (Kriegeskorte et al., 2008) and might very well contradict them. Thus, the relative position of IT cortex in the ventral visual hierarchy uniquely allows it to represent a balance of visual and non-visual properties and serve as an important translation point between these knowledge domains. This bidirectional perceptual-conceptual translation might underlie findings from the categorization literature in which semantic and/or conceptual knowledge about visual categories can cause them to be perceived as perceptually similar or different, regardless of their intrinsic visual similarity (Lupyan and Spivey, 2008; Lupyan et al., 2010; Lupyan, 2012). We tested our model's ability to perform perceptual to conceptual mapping by reserving a set of 6 object categories during training (bicycle, pliers, chair, guitar, machine gun, and chandelier) and recording the semantic activation associated with these untrained categories.

Figures 6B–D demonstrates the model's ability to produce semantic patterns that reflect the visual properties of objects from the reserved categories in relation to the other trained categories. For example, bicycles activated the semantics for motorcycle, and pliers tended to activate the semantics for either wrench or screwdriver. The resulting activation patterns were also similar to the ground-truth semantics for the untrained categories, indicating that the model could infer the veridical semantic features from visual appearance alone. Similar results were obtained for the other categories except for guitars, which failed to reliably activate semantically related items (instead, they weakly activated kitchen sink, hot air balloon, etc.). This overall pattern of results indicates that the model can infer the semantics of a novel object from its appearance, assuming the object contains visual features that are consistent with semantically related categories. Guitars presumably failed this test of semantic generalization because their

visual features do not appear in other music-related categories (e.g., drums, pianos, synthesizers). Despite this failure, this finding seems reasonable – if a novel object is really quite different in appearance from known objects, like a "Greeble" (Gauthier and Tarr, 1997), it might be difficult to infer its purpose from visual properties alone.

DISCUSSION

We have described a biological model of the ventral visual pathway that demonstrates several important ways in which a recurrent processing architecture could contribute visual object recognition. We showed that top-down connections can fill in missing information in partially occluded images. In addition, recurrent inhibitory competition in our model contributed additional robustness in the face of high levels of occlusion, through dynamic renormalization of activation levels. We also showed how top-down connectivity shapes the learned representations in our model to reflect semantic, as well as visual, information, in agreement with recent data (Kriegeskorte et al., 2008). This dual mapping between semantic and visual information enables the network to understand the semantic implications of visual features, properly generalizing semantic information based on bottom-up visual features of novel object categories. All of these results derive from principles developed as a general theory of the neocortex (O'Reilly, 1998; O'Reilly and Munakata, 2000; O'Reilly et al., 2012), which emphasizes the importance of the brain's ability to solve hard problems with powerful error-driven learning, and more generally specifies how relatively simple recurrent processing dynamics can give rise to more advanced cognitive phenomena.

Our results demonstrate how the dynamics that arise from recurrent connectivity can be important for vision across multiple timescales. First, these dynamics contribute in a meaningful way to the brain's robustness to visual degradations like partial occlusion by reinforcing probable "hypotheses" about the underlying stimulus through rapid recurrent processing. For example, an image of an occluded fish will weakly activate neural populations that are tuned to *fish* features (e.g., the dorsal fin, the tail, etc.) as well as neural populations that are tuned to other visually similar, but irrelevant, features (Wyatte et al., 2012b). Our model suggests that the brain could resolve this ambiguity via excitatory top-down connections by amplifying and filling-in neurons that are tuned to additional features that are consistent with the bottom-up inputs, but may not have been present in the actual stimulus. Competitive influences are equally important, which serve to suppress spurious activations that do not constitute valid category representations. This idea has been previously described in well-understood biological models of feedforward object processing such as HMAX (Riesenhuber and Poggio, 2002; Serre et al., 2007a) which contains a maximum operation that selects the most active feature across competitors for subsequent processing. While the efficacy of the maximum operation has been explored in the context of object clutter (Riesenhuber and Poggio, 1999; see also Wyatte et al., 2012b for a similar investigation using the LVis model), it has yet to be seen whether the same operation would be useful for the partial occlusion manipulation that we have explored here in which feature activation is vastly restricted. Thus, a comparison of different types of models on occluded object recognition tasks would be useful to determine the relative importance of mechanisms such as the maximum operation, compared to top-down amplification and filling-in.

Our results indicate that the result of recurrent processing over time is a consistent, and often object-complete representation at the IT-level. We found that these recurrent dynamics could also be a double-edged sword, and did not necessarily result in overall increases in recognition accuracy despite their ability to fill in missing or ambiguous information – if the top-down signal was inaccurate, then the system could equally be led astray in its overall interpretation. Overall, these recurrent dynamics are similar to other attractor networks that "clean up" noisy representations from perceptual processing modules and produce top-down biasing effects (e.g., McClelland and Rumelhart, 1981; Mozer and Behrmann, 1990; Kveraga et al., 2007). Our results show how these same principles can be realized in a unified, large-scale model of biological object recognition operating on real visual inputs.

Recurrent inhibitory dynamics are equally important for resolving degraded inputs during object recognition. Our results suggest that the inhibitory mechanisms present in our model dynamically adjust to the amount of excitation coming into a given area, which can cause weak signals to be perceived as amplified via normalization that increases their dynamic range. Normalization has been proposed as a canonical neural computation found within many brain regions spanning multiple sensory modalities (Carandini and Heeger, 2012) and is also an integral part of recent high performance computer vision models that are loosely based on the biology of the visual system (Pinto et al., 2009, 2011). However, a neural mechanism has not been definitively associated with normalization. While our model demonstrates that this computation can be accomplished by recurrent inhibitory dynamics, other models have found similar results can be produced with excitatory feedback (Heeger, 1992, 1993). Regardless of the implementation, these results collectively point to the importance of recurrent processing mechanisms that extend past the first responses in brain areas in resolving degraded inputs during object recognition.

While the iterative recurrent processing exhibited by our model can ultimately converge on the complete pattern of neural activity that corresponds to the correct category of an occluded stimulus, this processing can take quite some time to converge when the stimulus is heavily occluded (**Figure 4C**, compared to **Figures 4A,B**). Thus, our model makes the experimental prediction that interrupting the processing of heavily occluded inputs should impair recognition more than interrupting the processing of relatively unoccluded inputs due to there being a higher probability of preventing network convergence on a stable representation. Recent psychophysical studies from our lab that use backward masking to disrupt ongoing recurrent processing are consistent with this prediction (Wyatte et al., 2012a).

Recurrent processing at longer timescales that extend across the course of learning allow disparate brain areas that project into the ventral pathway, such as higher-level semantic areas (Kriegeskorte et al., 2008), to shape perceptual representations. "IT-level" features extracted via feedforward unsupervised learning mechanisms have failed to account for these semantic influences (Kiani et al., 2007), suggesting that they represent dimensions that are not reflected in raw visual similarities. Our recurrent model accounts

for this data and we also demonstrate how this higher-level organization of visual responses can be used to translate between perceptual and conceptual representations in which categories are formed according to non-visual metrics (Gotts et al., 2011).

Indeed, recent research has suggested that conceptual knowledge of visual categories can cause them to be perceived as perceptually similar or different, regardless of their intrinsic visual similarity (Lupyan and Spivey, 2008; Lupyan et al., 2010; Lupyan, 2012). What is less known, however, is whether this conceptual influence is present in perceptual representations themselves or due to a similarity metric computed by post-perceptual, decision processes (Chen and Proctor, 2012). While most data on object categorization suggest that IT cortex is tuned to shape-based properties shared across categories while neurons in prefrontal cortex represent more abstract, categorical properties (Freedman et al., 2001, 2003), recent data indicate that IT neurons do indeed exhibit abstract, categorical properties during certain timeframes of their full response (Meyers et al., 2008; Liu et al., 2009). Are these categorical properties simply feedback "echoes" from prefrontal categorization circuits or can conceptual knowledge modify the shape-based tunings of IT neurons?

Our results indicate that recurrent processing indeed modifies perceptual representations by allowing non-visual information from nearby associated brain areas to be incorporated into learning signals. This simple mechanism is likely responsible for a broad range of effects, such as action-related response properties in the ventral stream (due to connectivity with dorsal areas involved in object manipulation and tool use; Culham and Valyear, 2006; Mahon et al., 2007; Almeida et al., 2010; Mahon and Caramazza, 2011) and task-relevant IT neural tunings (due to connectivity with higher-level cognitive systems; Sigala and Logothetis, 2002; Nestor et al., 2008). Valence and emotion have also been shown affect perceptual processing, likely due to feedback from the amygdala and other limbic structures (Vuilleumier, 2005; Lim and Pessoa, 2008; Padmala and Pessoa, 2008), but so far no studies to our knowledge have investigated organizational changes in sensory areas. Overall, we suggest studies that investigate organizational structure (e.g., Kriegeskorte et al., 2008) are a fruitful domain for future research on object learning.

The detailed time course of feedforward, feedback, and inhibitory events that lead up to visual perception has been the subject of considerable debate in the literature. Research has suggested that object identity can be read out from IT neural populations in as little 80–100 ms (Oram and Perrett, 1992; Keysers et al., 2001; Hung et al., 2005) with the general conclusion that these responses must be driven solely by the initial feedforward spikes since the spikes must pass through 4 cortical areas (V1, V2, V4, and IT) with inter-areal latencies on the order of 10 ms (Nowak and Bullier, 1997). Our model is largely consistent with these feedforward latencies. For unambiguous inputs, object identity is reliably reflected in the IT activation pattern within 20 cycles (**Figures 4A,B**). Assuming 40–60 ms for the first spikes to appear in V1, this means 20 cycles corresponds to 40–60 ms in cortex, or around 2–3 ms per cycle. Each cycle updates the membrane potential (V_m, see S2 for equations) of all model units as a function of their input conductances, and thus a latency of 20 cycles for IT readout is a reasonable extension of the biology, especially in the context of large populations of neurons where the rate code approximates the instantaneous average population firing across discrete spiking neurons (Guyonneau et al., 2004).

In addition to the well-known feedforward latencies of ventral stream areas, research has indicated that downstream areas such as prefrontal cortices categorize inputs on the order or 150 ms (Thorpe et al., 1996; Vanrullen, 2007). However, some recent estimates place the latency of recurrent processing effects well within the 100–150 ms window (Lamme and Roelfsema, 2000; Foxe and Simpson, 2002; Kveraga et al., 2007; Roland, 2010), and thus it becomes increasingly unclear whether these latencies are purely driven by feedforward responses from IT neurons or reflect substantial influence from recurrent processing mechanisms. Our model may provide some clarification of these issues. Specifically, feedback projections send information back to earlier areas as soon as it is sent forward, gradually incorporating more and more recurrent loops, and inhibitory competition influences are always present, providing online renormalization effects. Thus, we do not believe there is such a thing as *purely* feedforward processing. Instead, it is just a matter of the extent to which recurrence plays a critical role in processing. For unambiguous inputs, our model converges quickly and identity can be resolved rapidly without much influence from recurrent processing. The predominant task used in studies citing support for purely feedforward processing involves a binary decision about whether an image contains an animal (Thorpe et al., 1996; Li et al., 2002; VanRullen and Koch, 2003). Thus, our model might suggest that this "animal vs. no animal" task involves relatively little ambiguity and thus, does not critically depend on recurrent processing for success. Alternatively, this task might rapidly recruit recurrent processing in as little as 100 ms (Koivisto et al., 2011).

With highly ambiguous inputs, recurrent processing becomes increasingly important for robust object recognition. In our model, this translates to overall longer latencies for convergence (**Figure 4C**). Accordingly, neurophysiological recordings have suggested that ambiguity is associated with longer latencies of processing, allowing for more iterations of feedforward, feedback, and local inhibitory interactions before convergence (Akrami et al., 2009; Daelli and Treves, 2010). Whether this convergence dynamic reflects rapid dynamics within and between hierarchically adjacent areas or comparatively longer latency influence from more distant sites that reflect "top-down" processing like attention is an open question that will need to be addressed to fully understand the dynamics involved in object recognition.

Much remains to be explored in the domain of recurrent processing in visual object recognition. As noted earlier, the issue of figure-ground processing and a potential role for top-down and bottom-up interactions in this domain is a topic of current research with the LVis model, and successful resolution of these issues would help to resolve several limitations of the current model, both in terms of being able to process images with realistic backgrounds at high levels of performance, and being able to use more naturalistic forms of occlusion. More generally, there are many different ideas in the literature about how the overall object recognition process may unfold across the different visual areas, and about the potential role for recurrent processing in the

brain. Thus, different models may suggest very different conclusions about the role of recurrent processing in object recognition. We are excited to compare our predictions against those of other such models, to eventually converge on a better understanding of how the biological system functions.

MATERIALS AND METHODS

STRUCTURE OF THE LVis MODEL

The LVis (Leabra Vision) model starts by preprocessing bitmap images via two stages of mathematical filtering that capture the qualitative processing thought to occur in the mammalian visual pathways from retina to LGN (lateral geniculate nucleus of the thalamus) to primary visual cortex (V1). The output of this filtering provides the input to the Leabra network, which then learns over a sequence of layers to categorize the inputs according to object categories. Although we have shown that the early stages of visual processing (through V1) can be learned via the self-organizing learning mechanisms in Leabra (O'Reilly and Munakata, 2000; O'Reilly et al., 2012), it was more computationally efficient to implement these steps directly in optimized C++ code. This optimized implementation retained the k-winners-take-all (kWTA) inhibitory competition dynamics from Leabra, which we have found to be important for successful recognition performance. Thus, the implementation can be functionally viewed as a single Leabra network.

For a full description of the early visual processing and parameters used in the model, see S1. The Leabra algorithm used to train and test the model is described in full detail in S2.

CU3D-100 DATASET

The CU3D-100 dataset consisted of 3D models from 100 diverse visual categories with an average of 9.42 exemplars per category. The individual models were downloaded from the Google 3D Warehouse[2]. Each model was normalized for differences in position, scale, and rotation using a set scripts written in Ruby and then imported into a software renderer where it was subjected to $\pm 20^{\circ}$ in-depth (3D) rotations (including a random 180° left-right flip for objects that are asymmetric along this dimension) with an overhead lighting positioned uniformly randomly along an 80° overhead arc. Models were rendered to PNG images in RGB color at a resolution of 320×320 pixels. This rendering process was repeated 20 times with different random 3D depth and lighting variations for each individual model, producing a total of 18840 images. The resulting dataset can be downloaded at http://cu3d.colorado.edu. A full breakdown of categories and number of models is available in S3.

TRAINING AND TESTING METHODS

The model was trained for 1000 epochs of 500 images per epoch. Two exemplars per category were reserved for testing. For each image presentation, the original image was converted to grayscale and downscaled to 144×144 pixels and a randomly parameterized affine transformation that translated, scaled, and rotated the image was then applied. These transformations were performed via a single function, which also used neighborhood smoothing to preserve image quality as much as possible. The parameters on these transformations for any given image presentation were drawn from a uniform distribution over the following normalized parameter ranges: *scale*: 0.9–1.1 (where 1.0 means presenting the image to the model at the original downscaled resolution), *translation*: −0.15–0.15 in horizontal and vertical dimensions (where 1.0 would be moving center of image to the very top or right of the model's inputs), *rotation*: −0.02–0.02 (where $1.0 = 2\pi$ or 360°).

Given these variations in the image presentations, no two inputs were likely to be identical over the course of training. Learning was asymptotic over the first few 100 epochs, but small improvements in generalization were observed by training for the full 1000 epochs. No evidence of overfitting was observed as a function of training duration. A total of 5 batches (training from different random initial weights and ordering of stimuli, with different train/test splits) were run using this method.

A testing trial consisted of seven presentations of a single image, with a different 2D affine transformation applied each time. For 2D voting results, a majority voting procedure integrated across these presentations to determine the final output. For higher-level voting, a second-order majority vote was then taken over the 20 testing trials with different 3D variations of each individual exemplar. All comparison models were tested using these same voting methods.

BLOB-BASED OCCLUSION

The blob-based occlusion algorithm involved the construction of a filter that was set to 1.0 within a circle of radius 5% of the image size (i.e., 5% of 144 pixels or 7 pixels) and then fell off outside the circle as a Gaussian function The σ parameter of the Gaussian was also set to 5% of the image size and the final effective size of the filter was 42×42 pixels (**Figure 3**). This filter was then used as a two-dimensional weighting function to determine how much of the image should be occluded with the gray background color, with 1.0 minus this value drawn from the original image. The peak of the filter contained weights of 1.0, and thus, image areas within the peak were completely occluded with the background color, and outside of that, the image exhibited a smooth gradient out to the original image. This smooth gradient (produced by the Gaussian) was important for not introducing novel input features at the edge of the circle occluder.

The percent occlusion parameter (O) specified the number of times to apply the filter to an image:

$$N_{apply} = 2.5O(I_{size}/H_{width} + 1) + 0.5 \quad (1)$$

where O was in the range [0, 1], I_{size} referred to the size of the input image in pixels, and H_{width} referred to the width of the filter.

For testing trials that used the occlusion manipulation, a majority vote was taken across the seven 2D affine transformations of a single image only, with the occlusion mask applied prior to any transformations, to ensure that an object's occluded features did not change across different transformations. Performance without this majority voting procedure produced the same qualitative pattern of results as seen in **Figure 5** and is available in S4.

[2]http://sketchup.google.com/3dwarehouse

SEMANTICS INPUTS

The semantic input vectors were composed of 100 different binary unit activation patterns of which 25% were active. These patterns started out as random binary patterns, which were systematically shaped over many iterations to capture the pairwise semantic similarity between the 100 object categories as captured from the standard latent semantic analysis (LSA; Landauer and Dumais, 1997) corpus (General Reading up to 1st year College), obtained from http://lsa.colorado.edu. Generating these semantics vectors was necessary because the original LSA vectors did not contain the sparse, binary patterns required to match the kWTA inhibitory dynamics of the Leabra algorithm.

The shaping procedure was accomplished via brute-force evolution described here. For each pair of patterns, bits to flip on in common between the two patterns (thus increasing their similarity) were chosen according to a softmax function weighted by the sum of the semantic distance times other pattern's bits. Bits were flipped in on/off pairs to ensure that kWTA constraint was preserved. Bits to flip off were chosen according to the opposite of the distance (1 minus the cosine distance). Critically, after a round of bit flipping, only those changes that increased the fit of the bit pattern distance matrix with that of the source LSA distance matrix were kept (i.e., a form of "ratcheting").

The final mean cosine difference between the two distance matrices was 0.000597733, indicating that the patterns of similarity between the random binary bit vectors did a good job of capturing the LSA similarities.

COMPARISON NETWORKS

Removing feedback from the Leabra model was achieved by simply multiplying all excitatory activation through feedback projections by zero such that the resulting input to a given layer at any point in time was limited to incoming feedforward activation.

The backpropagation networks had exactly the same layer structure and connectivity as the Leabra model, except of course for the lack of recurrent feedback connections. Both networks used cross-entropy error:

$$CE = \sum_k t_k \log o_k + (1 - t_k) \log (1 - o_k) \quad (2)$$

(where k is an index over output units, t is the target training value, and o is the network output value), with an additional error tolerance of 0.05 (differences in activation below this level did not drive learning), and no momentum or weight decay. The sparse network had bias weights initialized to −3.0, which greatly reduced overall levels of initial activity. A high learning rate of 0.2 was also usable with this configuration, and this higher learning rate produced better generalization. The distributed network had bias weights initialized to 0, producing high levels of activity in the layers, and a lower learning rate of 0.01 was required to obtain converging learning. Furthermore, the distributed model did *not* use the kWTA dynamics in the V1 filter front-end processing system, to more completely capture the behavior of a system that has no sparseness-inducing inhibitory dynamics or negative biases.

Both the Leabra model without feedback and the backpropagation networks used the same majority voting procedure as the Leabra model.

ACKNOWLEDGMENTS

The authors would like to thank Michael Tarr, Thomas Palmeri, Garrison Cottrell, Tim Curran, and Nicolas Pinto for their helpful comments and suggestions. This work utilized the Janus supercomputer, which is supported by the National Science Foundation (award number CNS-0821794) and the University of Colorado Boulder. The Janus supercomputer is a joint effort of the University of Colorado Boulder, the University of Colorado Denver, and the National Center for Atmospheric Research.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at http://www.frontiersin.org/Perception_Science/10.3389/fpsyg.2013.00124/abstract

REFERENCES

Akrami, A., Liu, Y., Treves, A., and Jagadeesh, B. (2009). Converging neuronal activity in inferior temporal cortex during the classification of morphed stimuli. *Cereb. Cortex* 19, 760–776.

Almeida, J., Mahon, B. Z., and Caramazza, A. (2010). The role of the dorsal visual processing stream in tool identification. *Psychol. Sci.* 21, 772–778.

Baylis, G. C., and Driver, J. (2001). Shape-coding in it cells generalizes over contrast and mirror reversal, but not figure-ground reversal. *Nat. Neurosci.* 4, 937–942.

Biederman, I., and Cooper, E. E. (1991). Priming contour-deleted images: evidence for intermediate representations in visual object recognition. *Cogn. Psychol.* 23, 393–419.

Bradski, G., and Grossberg, S. (1995). Fast-learning viewnet architectures for recognizing three-dimensional objects from multiple two-dimensional views. *Neural. Netw.* 8, 1053–1080.

Carandini, M., and Heeger, D. (2012). Normalization as a canonical neural computation. *Nat. Rev. Neurosci.* 13, 51–62.

Chen, J., and Proctor, R. W. (2012). Influence of category identity on letter matching: conceptual penetration of visual processing or response competition? *Attent. Percept. Psychophys.* 74, 716–729.

Craft, E., Schutze, H., Niebur, E., and der Heydt, R. (2007). A neural model of figure-ground organization. *J. Neurophysiol.* 97, 4310–4326.

Culham, J. C., and Valyear, K. F. (2006). Human parietal cortex in action. *Curr. Opin. Neurobiol.* 16, 205–212.

Daelli, V., and Treves, A. (2010). Neural attractor dynamics in object recognition. *Exp. Brain Res.* 203, 241–248.

DiCarlo, J. J., Zoccolan, D., and Rust, N. C. (2012). How does the brain solve visual object recognition? *Neuron* 73, 415–434.

Fei-Fei, L., Fergus, R., and Perona, P. (2007). Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. *Comput. Vis. Image Underst.* 106, 59–70.

Felleman, D. J., and Van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex* 1, 1–47.

Foxe, J. J., and Simpson, G. V. (2002). Flow of activation from v1 to frontal cortex in humans. A framework for defining "early" visual processing. *Exp. Brain Res.* 142, 139–150.

Freedman, D. J., Riesenhuber, M., Poggio, T., and Miller, E. K. (2001). Categorical representation of visual stimuli in the primate prefrontal cortex. *Science* 291, 312–316.

Freedman, D. J., Riesenhuber, M., Poggio, T., and Miller, E. K. (2003). A comparison of primate prefrontal and inferior temporal cortices during visual categorization. *J. Neurosci.* 23, 5235–5246.

Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* 36, 193–202.

Fukushima, K. (2003). Neocognitron for handwritten digit recognition. *Neurocomputing* 51, 161–180.

Gauthier, I., and Tarr, M. J. (1997). Becoming a "greeble" expert: exploring mechanisms for face recognition. *Vision Res.* 37, 1673–1682.

Gosselin, F., and Schyns, P. G. (2001). Bubbles: a technique to reveal the use of information in recognition tasks. *Vision Res.* 41, 2261–2271.

Gotts, S. J., Milleville, S. C., Bellgowan, P. S. F., and Martin, A. (2011). Broad and narrow conceptual tuning in the human frontal lobes. *Cereb. Cortex* 21, 477–491.

Grill-Spector, K., Kourtzi, Z., and Kanwisher, N. (2001). The lateral occipital complex and its role in object recognition. *Vision Res.* 41, 1409–1422.

Guyonneau, R., Vanrullen, R., and Thorpe, S. J. (2004). Temporal codes and sparse representations: a key to understanding rapid processing in the visual system. *J. Physiol. Paris* 98, 487–497.

Heeger, D. J. (1992). Normalization of cell responses in cat striate cortex. *Vis. Neurosci.* 9, 181–197.

Heeger, D. J. (1993). Modeling simple-cell direction selectivity with normalized, half-squared, linear operators. *J. Neurophysiol.* 70, 1885–1898.

Hinton, G. E., and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science* 313, 504–507.

Hubel, D., and Wiesel, T. N. (1962). Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *J. Physiol.* 160, 106–154.

Hung, C. P., Kreiman, G., Poggio, T., and DiCarlo, J. J. (2005). Fast readout of object identity from macaque inferior temporal cortex. *Science (New York N.Y.)* 310, 863–866.

Hupe, J. M., James, A. C., Payne, B. R., Lomber, S. G., Girard, P., and Bullier, J. (1998). Cortical feedback improves discrimination between figure and background by v1 v2 and v3 neurons. *Nature* 394, 784–787.

Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun, Y. (2009). "What is the best multi-stage architecture for object recognition," in *2009 IEEE 12th International Conference on Computer Vision* (IEEE), 2146–2153.

Keysers, C., Xiao, D. K., Fldik, P., and Perrett, D. I. (2001). The speed of sight. *J. Cogn. Neurosci.* 13, 90–101.

Kiani, R., Esteky, H., Mirpour, K., and Tanaka, K. (2007). Object category structure in response patterns of neuronal population in monkey inferior temporal cortex. *J. Neurophysiol.* 97, 4296–4309.

Koivisto, M., Railo, H., Revonsuo, A., Vanni, S., and Salminen-Vaparanta, N. (2011). Recurrent processing in v1/v2 contributes to categorization of natural scenes. *J. Neurosci.* 31, 2488–2492.

Kourtzi, Z., and Kanwisher, N. (2001). Representation of perceived object shape by the human lateral occipital complex. *Science* 293, 1506–1509.

Kriegeskorte, N., Mur, M., Ruff, D. A., Kiani, R., Bodurka, J., Esteky, H., et al. (2008). Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron* 60, 1126–1141.

Kveraga, K., Ghuman, A., and Bar, M. (2007). Top-down predictions in the cognitive brain. *Brain Cogn.* 65, 145–168.

Lamme, V., and Roelfsema, P. (2000). The distinct modes of vision offered by feedforward and recurrent processing. *Trends Neurosci.* 23, 571–579.

Landauer, T. K., and Dumais, S. T. (1997). A solution to plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychol. Rev.* 104, 211–240.

LeCun, Y., Huang, F., and Bottou, L. (2004). "Learning methods for generic object recognition with invariance to pose and lighting," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2 (IEEE), 97–104

Lee, T. S., and Nguyen, M. (2001). Dynamics of subjective contour formation in the early visual cortex. *Proc. Natl. Acad. Sci. U.S.A.* 98, 1907–1911.

Lerner, Y., Hendler, T., and Malach, R. (2002). Object-completion effects in the human lateral occipital complex. *Cereb. Cortex* 12, 163–177.

Li, F. F., VanRullen, R., Koch, C., and Perona, P. (2002). Rapid natural scene categorization in the near absence of attention. *Proc. Natl. Acad. Sci. U.S.A.* 99, 9596–9601.

Lim, S.-L., and Pessoa, L. (2008). Affective learning increases sensitivity to graded emotional faces. *Emotion* 8, 96–103.

Liu, H., Agam, Y., Madsen, J. R., and Kreiman, G. (2009). Timing timing timing: fast decoding of object information from intracranial field potentials in human visual cortex. *Neuron* 62, 281–290.

Logothetis, N. K., Pauls, J., and Poggio, T. (1995). Shape representation in the inferior temporal cortex of monkeys. *Curr. Biol.* 5, 552–563.

Lupyan, G. (2012). Linguistically modulated perception and cognition: the label-feedback hypothesis. *Front. Psychol.* 3:54. doi:10.3389/fpsyg.2012.00054

Lupyan, G., and Spivey, M. J. (2008). Perceptual processing is facilitated by ascribing meaning to novel stimuli. *Curr. Biol.* 18, R410–R412.

Lupyan, G., Tompson-Schill, S. L., and Swingley, D. (2010). Conceptual penetration of visual processing. *Psychol. Sci.* 21, 1–10.

Mahon, B., Milleville, S., Negri, G., Rumiati, R., Caramazza, A., and Martin, A. (2007). Action-related properties of objects shape object representations in the ventral stream. *Neuron* 55, 507–520.

Mahon, B. Z., and Caramazza, A. (2011). What drives the organization of object knowledge in the brain? *Trends Cogn. Sci. (Regul. Ed.)* 15, 97–103.

Marr, D. (1982). *Vision*. New York: Freeman.

Masquelier, T., and Thorpe, S. J. (2007). Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Comput. Biol.* 3:e31. doi:10.1371/journal.pcbi.0030031

McClelland, J. L., and Rumelhart, D. E. (1981). An interactive activation model of context effects in letter perception: part 1 an account of basic findings. *Psychol. Rev.* 88, 375–407.

Meyers, E. M., Freedman, D. J., Kreiman, G., Miller, E. K., and Poggio, T. (2008). Dynamic population coding of category information in inferior temporal and prefrontal cortex. *J. Neurophysiol.* 100, 1407–1419.

Mozer, M. C., and Behrmann, M. (1990). On the interaction of selective attention and lexical knowledge: a connectionist account of neglect dyslexia. *J. Cogn. Neurosci.* 96, 96–123.

Mutch, J., and Lowe, D. (2008). Object class recognition and localization using sparse features with limited receptive fields. *Int. J. Comput. Vis.* 80, 45–57.

Nestor, A., Vettel, J. M., and Tarr, M. J. (2008). Task-specific codes for face recognition: how they shape the neural representation of features for detection and individuation. *PLoS ONE* 3:e3978. doi:10.1371/journal.pone.0003978

Nowak, L., and Bullier, J. (1997). "The timing of information transfer in the visual system," in *Extrastriate Cortex in Primates, Cerebral Cortex*, Vol. 12, eds K. S. Rockl, J. H. Kaas, and A. Peters (New York: Plenum Press), 205–241.

Olshausen, B. A., and Field, D. J. (2004). Sparse coding of sensory inputs. *Curr. Opin. Neurobiol.* 14, 481–487.

Oram, M. W., and Perrett, D. I. (1992). Time course of neural responses discriminating different views of the face and head. *J. Neurophysiol.* 68, 70–84.

Orban, G. A., Van Essen, D., and Vanduffel, W. (2004). Comparative mapping of higher visual areas in monkeys and humans. *Trends Cogn. Sci. (Regul. Ed.)* 8, 315–324.

O'Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: the generalized recirculation algorithm. *Neural. Comput.* 8, 895–938.

O'Reilly, R. C. (1998). Six principles for biologically-based computational models of cortical cognition. *Trends Cogn. Sci. (Regul. Ed.)* 2, 455–462.

O'Reilly, R. C. (2001). Generalization in interactive networks: the benefits of inhibitory competition and Hebbian learning. *Neural. Comput.* 13, 1199–1242.

O'Reilly, R. C., and Munakata, Y. (2000). *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*. Cambridge, MA: The MIT Press.

O'Reilly, R. C., Munakata, Y., Frank, M. J., Hazy, T. E., and Contributors (2012). *Computational Cognitive Neuroscience*, 1st Edn. Wiki Book. Available at: http://ccnbook.colorado.edu

Padmala, S., and Pessoa, L. (2008). Affective learning enhances visual detection and responses in primary visual cortex. *J. Neurosci.* 28, 6202–6210.

Patterson, K., Nestor, P. J., and Rogers, T. T. (2007). Where do you know what you know? The representation of semantic knowledge in the human brain. *Nat. Rev.* 8, 976–987.

Pinto, N., Barhomi, Y., Cox, D., and DiCarlo, J. (2011). "Comparing state-of-the-art visual features on invariant object recognition tasks," in *2011 IEEE Workshop on Applications of Computer Vision (WACV)* (IEEE), 463–470.

Pinto, N., Cox, D. D., and DiCarlo, J. J. (2008). Why is real-world visual object recognition hard? *PLoS Comput. Biol.* 4:e27. doi:10.1371/journal.pcbi.0040027

Pinto, N., Doukhan, D., DiCarlo, J. J., and Cox, D. D. (2009). A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS Comput. Biol.* 5:e1000579. doi:10.1371/journal.pcbi.1000579

Ponce, J., Berg, T., Everingham, M., Forsyth, D., Hebert, M., Lazebnik, S., et al. (2006). Dataset issues in object recognition. *Lect. Notes Comput. Sci.* 4170, 29–48.

Poort, J., Raudies, F., Wannig, A., Lamme, V. A. F., Neumann, H., and Roelfsema, P. R. (2012). The role of attention in figure-ground segregation in areas v1 and v4 of the visual cortex. *Neuron* 75, 143–156.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychol. Rev.* 85, 59–107.

Ratcliff, R., and McKoon, G. (2008). The diffusion decision model: theory and data for two-choice decision tasks. *Neural. Comput.* 20, 873–922.

Rauschenberger, R., Liu, T., Slotnick, S. D., and Yantis, S. (2006). Temporally unfolding neural representation of pictorial occlusion. *Psychol. Sci.* 17, 358–364.

Riesenhuber, M., and Poggio, T. (1999). Hierarchical models of object recognition in cortex. *Nat. Neurosci.* 3, 1199–1204.

Riesenhuber, M., and Poggio, T. (2002). Neural mechanisms of object recognition. *Curr. Opin. Neurobiol.* 12, 162–168.

Roelfsema, P. R., Lamme, V. A. F., Spekreijse, H., and Bosch, H. (2002). Figure-ground segregation in a recurrent network architecture. *J. Cogn. Neurosci.* 14, 525–537.

Roland, P. (2010). Six principles of visual cortical dynamics. *Front. Syst. Neurosci.* 4:28 doi:10.3389/fnsys.2010.00028

Rolls, E. T., and Stringer, S. M. (2006). Invariant visual object recognition: a model, with lighting invariance. *J. Physiol. Paris* 100, 43–62.

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

Scannell, J., Blakemore, C., and Young, M. P. (1995). Analysis of connectivity in the cat cerebral cortex. *J. Neurosci.* 15, 1463–1483.

Scholte, H. S., Jolij, J., Fahrenfort, J. J., and Lamme, V. A. F. (2008). Feedforward and recurrent processing in scene segmentation: electroencephalography and functional magnetic resonance imaging. *J. Cogn. Neurosci.* 20, 2097–2109.

Seghier, M. L., and Vuilleumier, P. (2006). Functional neuroimaging findings on the human perception of illusory contours. *Neurosci. Biobehav. Rev.* 30, 595–612.

Serre, T., Kreiman, G., Kouh, M., Cadieu, C., Knoblich, U., and Poggio, T. (2007a). A quantitative theory of immediate visual recognition. *Prog. Brain Res.* 165, 33–56.

Serre, T., Oliva, A., and Poggio, T. (2007b). A feedforward architecture accounts for rapid categorization. *Proc. Natl. Acad. Sci. U.S.A.* 104, 6424–6429.

Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., and Poggio, T. (2007c). Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Anal. Mach. Intell.* 29, 411–426.

Sigala, N., and Logothetis, N. K. (2002). Visual categorization shapes feature selectivity in the primate temporal cortex. *Nature* 415, 318–320.

Sporns, O., Honey, C. J., and Kotter, R. (2007). Identification and classification of hubs in brain networks. *PLoS ONE* 2:e1049. doi:10.1371/journal.pone.0001049

Sporns, O., and Zwi, J. D. (2004). The small world of the cerebral cortex. *Neuroinformatics* 2, 145–162.

Tanaka, K. (1996). Inferotemporal cortex and object vision. *Annu. Rev. Neurosci.* 19, 109–139.

Thorpe, S., Fize, D., and Marlot, C. (1996). Speed of processing in the human visual system. *Nature* 381, 520–522.

Tompa, T., and Sary, G. (2010). A review on the inferior temporal cortex of the macaque. *Brain Res. Rev.* 62, 165–182.

Ungerleider, L. G., and Haxby, J. V. (1994). "What" and "Where" in the human brain. *Curr. Opin. Neurobiol.* 4, 157–165.

Van Essen, D. C., Anderson, C. H., and Felleman, D. J. (1992). Information processing in the primate visual system: an integrated systems perspective. *Science* 255, 419–423.

Vanrullen, R. (2007). The power of the feed-forward sweep. *Adv. Cogn. Psychol.* 3, 167–176.

VanRullen, R., and Koch, C. (2003). Visual selective behavior can be triggered by a feed-forward process. *J. Cogn. Neurosci.* 15, 209–217.

Vuilleumier, P. (2005). How brains beware: neural mechanisms of emotional attention. *Trends Cogn. Sci. (Regul. Ed.)* 9, 585–594.

Wallis, G., and Rolls, E. T. (1997). Invariant face and object recognition in the visual system. *Prog. Neurobiol.* 51, 167–194.

Weigelt, S., Singer, W., and Muckli, L. (2007). Separate cortical stages in amodal completion revealed by functional magnetic resonance adaptation. *BMC Neurosci.* 8:70. doi:10.1186/1471-2202-8-70

Wyatte, D., Curran, T., and O'Reilly, R. (2012a). The limits of feedforward vision: recurrent processing promotes robust object recognition when objects are degraded. *J. Cogn. Neurosci.* 24, 2248–2261.

Wyatte, D., Herd, S., Mingus, B., and O'Reilly, R. (2012b). The role of competitive inhibition and top-down feedback in binding during object recognition. *Front. Psychol.* 3:182. doi:10.3389/fpsyg.2012.00182

Zhaoping, L. (2005). Border ownership from intracortical interactions in visual area v2. *Neuron* 47, 143–153.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 27 August 2012; accepted: 26 February 2013; published online: 01 April 2013.

Citation: O'Reilly RC, Wyatte D, Herd S, Mingus B and Jilk DJ (2013) Recurrent processing during object recognition. Front. Psychol. ***4****:124. doi: 10.3389/fpsyg.2013.00124*

This article was submitted to Frontiers in Perception Science, a specialty of Frontiers in Psychology.

Neural Networks 21 (2008) 1146–1152

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

2008 Special Issue

The Emergent neural modeling system☆,☆☆

Brad Aisa*, Brian Mingus, Randy O'Reilly

Computational Cognitive Neuroscience Lab, Department of Psychology, University of Colorado at Boulder, United States

ARTICLE INFO

Article history:
Received 1 November 2007
Received in revised form
10 June 2008
Accepted 17 June 2008

Keywords:
Neural networks
Robotics
Simulator

ABSTRACT

Emergent (http://grey.colorado.edu/emergent) is a powerful tool for the simulation of biologically plausible, complex neural systems that was released in August 2007. Inheriting decades of research and experience in network algorithms and modeling principles from its predecessors, PDP++ and PDP, Emergent has been redesigned as an efficient workspace for academic research and an engaging, easy-to-navigate environment for students. The system provides a modern and intuitive interface for programming and visualization centered around hierarchical, tree-based navigation and drag-and-drop reorganization. Emergent contains familiar, high-level simulation constructs such as Layers and Projections, a wide variety of algorithms, general-purpose data handling and analysis facilities and an integrated virtual environment for developing closed-loop cognitive agents. For students, the traditional role of a textbook has been enhanced by wikis embedded in every project that serve to explain, document, and help newcomers engage the interface and step through models using familiar hyperlinks. For advanced users, the software is easily extensible in all respects via runtime plugins, has a powerful shell with an integrated debugger, and a scripting language that is fully symmetric with the interface. Emergent strikes a balance between detailed, computationally expensive spiking neuron models and abstract, Bayesian or symbolic systems. This middle level of detail allows for the rapid development and successful execution of complex cognitive models while maintaining biological plausibility.

1. Introduction

Emergent (http://grey.colorado.edu/emergent) is a powerful tool for the simulation of biologically plausible, complex neural systems that was released in August 2007. The immediate predecessor to Emergent is PDP++ v3.2, a tool used by a variety of researchers for neural modeling and teaching. PDP++ was itself an extension of the PDP software released by McClelland and Rumelhart in 1986 with their groundbreaking book, Parallel Distributed Processing (McClelland & Rumelhart, 1986). Emergent represents a near complete rewrite of PDP++, replacing an aging and largely unsupported graphical user interface (GUI) framework called Interviews with a well supported, more modern one called `Qt` (http://trolltech.com/products/qt). A major benefit of Qt is its seamless integration into all major platforms, allowing Emergent to not only be easily installed on them, but also to adopt their native look and feel. With this in mind, we completely redesigned the user interface, employing a now-familiar tree-based browser approach (with tabbed edit/view panels) for project exploration and interaction (Fig. 1). We also radically redesigned or even replaced several core constructs from the previous product, such as `Environments` and `Processes`, replacing them with the more general-purpose `DataTable` and `Program` constructs that will be discussed later.

More important than technical or interface changes, we also extended the intended scope of the tool. Whereas the previous versions were primarily intended for relatively small research and teaching models, typically aimed at demonstrating some isolated or delimited piece of functionality, the new version is intended to support very large-scale simulations of entire integrated brain-like systems. And whereas the previous versions were primarily designed for closed simulations using simple fixed data patterns as input and output, Emergent has been designed to accommodate external "closed-loop" sensory and motor connections both by plugins and with a built-in simulation environment that includes a rigid-body physics simulation for creating virtual robot-like agents.

This article will give a general overview of Emergent's features and capabilities, ending with a comparison with other neural network simulators and a discussion of the features we plan to implement in the near future.

☆ Supported by grants: NIH R01 MH069597, ONR N00014-07-1-0651, DARPA/ONR N00014-05-1-0880, ONR N00014-03-1-0428 (O'Reilly); NIH IBSC 1 P50 MH 64445 (McClelland).

☆☆ Thanks go to Dave Jilk for being the intrepid early adopter; Jay McClelland of Carnegie Melon University and Jonathan Cohen of Princeton University for their financial assistance during Emergent's development; and all members of the CCN Lab at CU Boulder for their valuable input and patient testing of the software.

* Corresponding author. Tel.: +1 720 233 0225; fax: +1 303 492 2967.
E-mail address: Brad.Aisa@colorado.edu (B. Aisa).

0893-6080/$ – see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2008.06.016

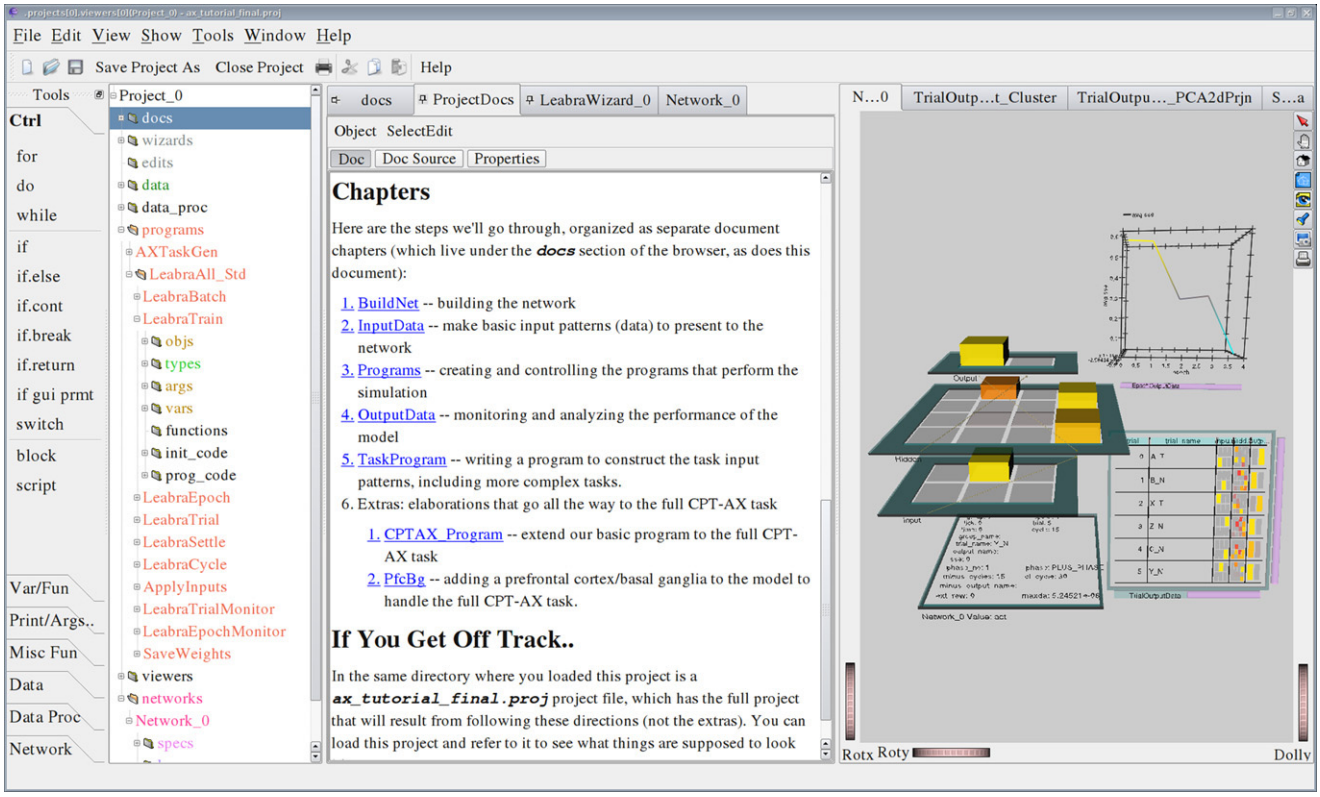

Fig. 1. The Emergent Project Browser. The main workspace in Emergent showing, from left: (a) the Toolbox with widgets for Programming and similar tasks; (b) the main Browser, a hierarchical tree of all objects in the project; (c) the Panel area, for editing and viewing the details of objects, in this case displaying a Doc; and (d) the 3D viewer, for viewing simulation objects in true 3D.

2. Emergent

2.1. Supported algorithms

Out of the box, Emergent supports classic back-propagation (BP) (Rumelhart, Hinton, & Williams, 1986), and recurrent back-propagation in several variants (Almeida, 1987; Pineda, 1987; Williams & Zipser, 1989); Constraint-satisfaction (CS) including the Boltzmann Machine (Ackley et al., 1985), Interactive Activation and Competition, and other related algorithms; Self-organized learning (SO) including Hebbian Competitive learning and variants (Rumelhart & Zipser, 1986) and Kohonen's Self-Organizing Maps and variants (Kohonen, 1984); and Leabra (an acronym for "local error-driven and biologically realistic algorithm") which includes key features from each of the above algorithms in one coherent framework (O'Reilly & Munakata, 2000).

The previous version of the software (PDP++ v3.2) also served as the basis for some other neural algorithms or extensions, including the Real-time Neural Simulator RNS++ (http://ccsrv1.psych.indiana.edu/rns++/) (Josh Brown); Long Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997); and the oscillating inhibition learning mechanism (Norman et al., 2006). The enhanced user-friendliness of the software and our new plugin technology make these kinds of extensions very easy to implement, hopefully encouraging more researchers to consider Emergent as the architectural base of their research algorithms. Unlike the tools such as MATLAB, Emergent is completely free and open-source; in addition, its network algorithms run at compiled C++ speed, rather than in an interpreter.

2.2. General features

Emergent opens to present a familiar tree-based browser (on the left) plus detail panel (on the right.) The user can select any object in the left-hand tree to see its detailed properties on the right, and open container nodes to reveal the sub-contents. Many objects have several detail sub-panels that present the object and its content in different views, depending on the purpose of the user. For example, the table object provides a panel with the properties of the table itself, one that lists the columns, and one that enables the user to browse or edit data. The user can open up any number of new browsers rooted at any point in an existing browser.

Clipboard and drag-and-drop manipulation of objects are supported wherever it makes sense. Many "action-like" operations, such as assigning an object to a program variable element, can be done via drag-and-drop.

When the user opens or creates a project, an additional `viewer` pane appears in the browser; this viewer supports one or more `frames` which display a true 3D rendering of one or more objects in the system, such as networks, graphs and virtual environments.

2.3. Networks

The basic unit of modeling in Emergent is the `Unit`, which is a neuron-like object that represents a small population of like-coding spiking neurons, such as might be observed in a cortical column. Its output is typically a time continuous value ranging from 0 to 1, which represent the extremes of "no firing in the population" to "maximal firing" in the population. Maximal firing is a product of the number of individual neurons times the rate of firing per each neuron. For more detailed neural models it is also possible to run the units in discrete spiking mode. These Units perform separate integrations of excitatory, inhibitory, and leak inputs to accommodate shunting inhibition effects, replicating the classic equivalent circuit dynamics of real neurons. The output is typically based on a thresholded, parameterized, bounded and

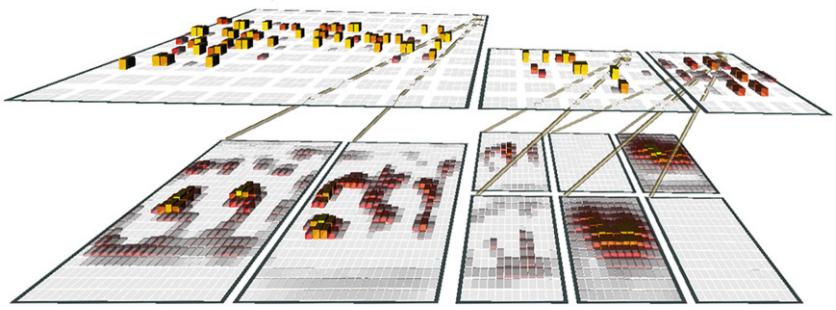

Fig. 2. Emergent Layers, showing Projections and Units.

sigmoidal-like curve. Other transfer function options are provided, and custom functions are possible.

Units are not instantiated or manipulated individually, but are managed in a group called a `Layer`. A Layer in Emergent is a two-dimensional "sheet" of Units, all of which share unit-level parameters (via a `UnitSpec`), inhibitory dynamics and patterns of connectivity to other Layers. A Layer can further be divided into a sub-grid of `UnitGroups`, which enables two levels of inhibitory dynamics, and more sophisticated granularity of connectivity with other Layers. Each Layer has a `LayerSpec` with parameters to control things like inhibitory dynamics, and how input data (if any) is mixed with the existing activation.

Emergent has two distinct constructs for representing connectivity between Units: the `Projection` and the `Connection`. A Projection specifies a logical, unidirectional connection between two layers; a Connection is the actual physical connection between a Unit and its targets, and is analogous to a neural synapse. A Projection specifies the pattern of connectivity between the layers, such as "all to all" or "tessellated", as well as a `ConnectionSpec` that
controls the parameters of the underlying connections that are generated. The ConnectionSpec has parameters that control the physical connections including the weighting of the connection relative to other connections, the weight limits (if any), the local learning rates (Hebbian and error-driven terms) and other miscellaneous parameters. Fig. 2 is an example network showing Layers and Projections.

A set of Layers is aggregated into an overall structure called a Network. There is typically one Network in use during any simulation run, although a Project can contain any number Networks. This can be helpful when testing different approaches as you can share all the other elements, such as control Programs, data input and output and monitoring Programs, to name a few.

2.4. Specs

`Specs` in Emergent are like styles in a word processing program—collections of parameters that can be applied to instances of a specific type, to control or modify their behavior. Specs help the modeler to keep parameters consistent across many instances of a same-type object, such as a Layer of a certain purpose. Sub-specs can be created that automatically inherit their values from a parent Spec, but in which selected parameters can be explicitly overridden. This helps to keep related but distinct Specs coordinated, except for the specific parameter values on which they differ. Specs can be nested to any practical level.

Emergent provides a convenient facility to easily determine which network constructs are associated with each spec. The user can click the Spec in the network control panel, and the items using that Spec are immediately highlighted in the network display. Specs also help to make a model easier to understand. Once an observer has first examined the overall network structure, the next step to understanding the model would be to click on the Specs, which will highlight the objects using them in the 3D viewer.

2.5. Algorithm infrastructure mechanisms

The base classes described above (Connection, Unit, Layer, etc) provide support for a range of common neural network processing mechanisms, such as computing the net input as a function of sending activations times weights. Specific algorithms then add their unique learning and processing mechanisms (e.g., Hebbian learning, inhibitory competition, discrete spiking). Furthermore, all of the implemented algorithms provided with the simulator provide a range of different algorithm variants that can be mixed-and-matched to create novel network architectures. These variations include different learning rules, activation functions, inhibitory mechanisms, etc. These variations are implemented either with a user-selectable switch within a common Spec class, or by a new subclass Spec type that directly implements the new functionality, which the user then selects by applying that spec to the appropriate network elements.

Although all the algorithms are derived from common base classes, each has incompatible optimizations and specializations relative to the others, such that they cannot be mixed in the same network. Thus, it is not possible to directly mix a self-organizing map layer with a backpropagation layer in the same network: supporting such heterogeneous collections would require N^2 kinds of conditional mechanisms and is not efficient and often would lead to nonsensical results. However, it is very straightforward to arrange for the communication of activations or other values between multiple networks of different types, effectively creating hybrid overall architectures. As a perhaps more satisfying alternative, many users leverage the unified framework for many of these different mechanisms provided by the Leabra algorithm, where such architectures can be created by differential parameterization of different layers.

Critically, all of the infrastructure for visualization and data analysis can be automatically applied to any new objects defined in the system, thanks to a powerful type-access system that scans header files and makes the software "self-aware" of very detailed type information for every object defined in the source code (including plugins).

2.6. Network input/output and data monitoring

Most network input and output in an Emergent simulation is facilitated by one or more `DataTable` ("table") objects. A table is similar to a spreadsheet table or database table: it is a set of one or more columns, each of which can contain data of a single type. The table then holds zero or more rows of data. In Emergent, a table cell can hold a matrix in addition to a scalar value. For network input and output data, a matrix column is typically created corresponding to the dimensions of a target or source layer in the network.

Emergent was designed with external connectivity in mind. External sources of input such as video and audio can be

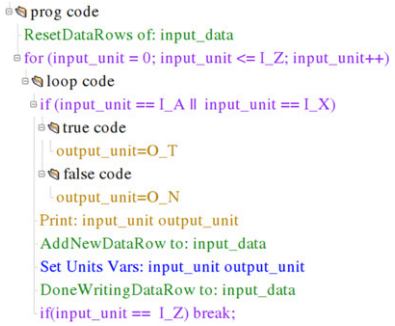

Fig. 3. Example program from Emergent's `AX Tutorial`, which walks the user through an implementation of the CPT-AX task used in working memory studies. All elements and groups of elements support copy, paste, drag-and-drop, and lines are color coded according to type.

preprocessed and the raw network input pattern then written to a table. Likewise, network output is first written to a data table, and can then be output to effectors such as simulated muscles, or even a real robot.

The table is the basic construct for network monitoring. Network object parameters, particularly Unit activation values, can be logged to a table during the simulation. The user can log to many tables at multiple levels of time, such as Epoch, Trial, or even Cycle (a single update of all network activations.)

2.7. Simulation control (programs)

Emergent provides a sophisticated but user-friendly general-purpose program environment that is used for sequencing simulations and doing general-purpose data processing tasks. A `Program` is a GUI-based tree of programming widgets called `ProgEls` that enables even a novice user to construct a variety of control sequences such as loops and conditional tests. The application comes with a pre-built library of programs that perform common simulation sequencing functions such as Batch, Epoch and Trial. The Network Wizard will automatically load and connect these to a network.

Programs, an example of which is displayed in Fig. 3, are a simple way of generating C-Super Script (CSS), the native scripting language of Emergent. CSS is best thought of as interpreted C++ and provides full access to the internal objects of Emergent as well as the ability to declare new classes. The resulting script can be examined in a text window. Advanced users can put CSS code in a program script element, but this is usually not necessary because visual programming in Emergent is faster than writing code by hand. CSS scripts are compiled into an efficient object-oriented byte code, which is then run at execution time. The generated network algorithms and data processing primitives are coded in highly optimized C++, scripting being used primarily to control and sequence these optimized workhorse operations.

2.8. Visualizations

Networks, graphs, tabular logs, and physical simulation objects can all be presented in an OpenGL-based 3D visualization environment. The user can create any number of `Frames`, each of which can contain a 3D visualization of one or more of the objects listed. GUI updates can be explicitly disabled to speed up simulations, and are always implicitly suppressed for non-visible panels.

A network display depicts the Unit activation values by default, but the user can select any parameter of Units, Connections, or Projections to monitor.

2.9. Data analysis and graphing

Emergent provides several facilities to help with data analysis. A collection of GUI-accessible objects provide common data processing operations, including: database-style operations (select, sort, join, group, sums, etc.); data analysis (distance, smoothing, dimension reduction such as clustering, PCA, SVD, etc.); data generation (random patterns, line patterns, noise, etc.); and image processing (rotation, translation, scaling, Gaussian and difference-of-Gaussian filtering, etc.). All of these operations are also available to Programs.

As would be expected, data can be readily imported or exported for use with other systems. It is also easy to copy and paste data to and from the clipboard, to exchange data with other programs such as Excel. Additionally, a full range of data graphing operations are available, to present 2D or 3D graphs, to display data from any table. Specialized graph types such as cluster trees are supported.

A 3D `GridView` displays some or all of the columns in a table. It is especially useful for displaying input and output patterns, including photos, and for aggregate Epoch, Train or Trial-level statistics, such as error values.

The `GNU Scientific Library (GSL)` has been incorporated into Emergent, making many of its routines and data structures available. `Matrix` objects, the underlying basis for tables columns, are compatible with GSL routines.

2.10. Virtual environment

Neural network researchers are increasingly testing their simulations of brain processes by embodying them in simulated physical agents that can act in a physically simulated world containing other objects or even other such agents. Emergent includes a powerful built-in simulator along with associated modeling constructs to enable building robot-like agents and connecting them to neural simulations. The simulator is based on two widely used technologies: the `Open Dynamics Engine (ODE)` (http://www.ode.org/) for physical modeling and OpenGL and `Coin3D` (http://www.coin3d.org/) for visualization.

The simulator provides access to all of ODE, including constructs for modeling bodies, including objects such as cylinders, boxes and spheres. Limbs and bodies are connected with Joints that have parameters controlling things like angular stops, degrees of freedom and stiffness. Forces can be applied to joints which results in torques dependent on the bodies they are connected to. Textures can be applied to objects and backgrounds to add visual realism, particularly for vision-based simulations. Objects obey the laws of physics, and phenomena such as momentum, elastic collisions, friction, and gravity are all modeled. Cameras are provided to enable endowing an agent with vision, the result of which can be readily interfaced with network model. 3D placement of sound sources is supported using the Simage library (for real-time playback) or the included `audioproc` plugin for localized sound.

Emergent ships with a simple example model demonstrating how to model a reaching task using an agent with a torso, head, shoulders, and arm, as pictured in Fig. 4.

2.11. Documentation, annotation, and search

Neural models can become quite complex, involving factors ranging from the overall purpose of the experiment, to its architecture, to the parameters being chosen for the elements, to the many experiments that may be conducted upon it, to scientific references and so on. Documenting all these elements can be a challenge. Emergent provides support for this in several areas.

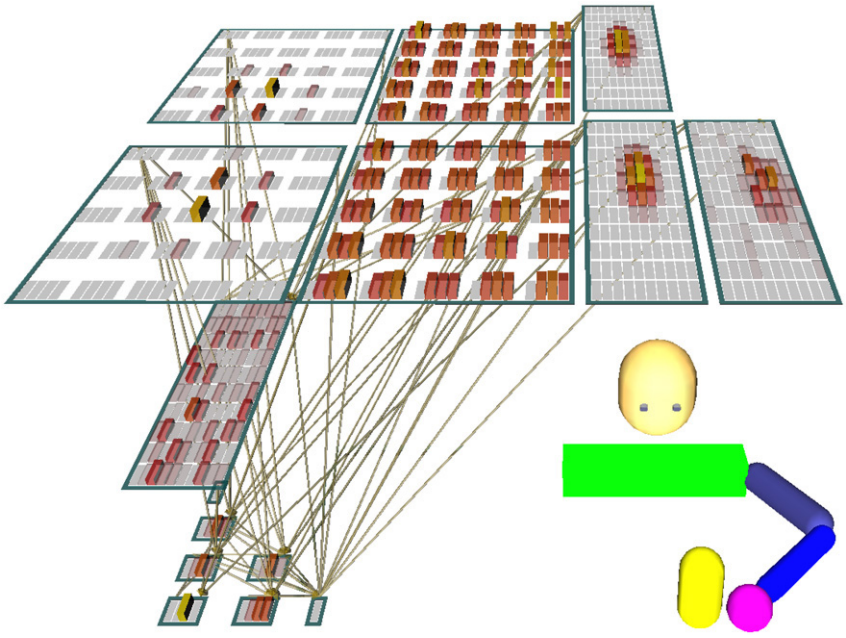

Fig. 4. A simulated agent in the Emergent virtual environment. The top left layers represent the elbow and shoulder forces, as read from ODE at the start of the reach. The middle top layers are the forces that the model produces, and are decoded to apply that force to his elbow and shoulder joints. The inner Gaussian blobs represent the goal location and his hand position at the start of the reach, and the outer blob represents his guess as to where the forces he produces will ultimately land the hand. All of this is orchestrated by the hidden layer, which is in the middle row. Notably, this model does not use error-driven learning, instead using the Primary Value Learned Value (PVLV) reinforcement learning system (O'Reilly et al., 2007), the lower cluster of seven layers, which is based on the detailed anatomy of midbrain dopaminergic neurons. PVLV is available as a Network Wizard for all models to use.

First, most user items include a `desc` field that lets the user include a textual description of the item.

Emergent also provides `Docs`, which are wiki-like pages inside the project. In addition to formatted textual material, Docs can also include hyperlinks (URLs) to external web sources, and "internal URLs" that can link to another doc, or even invoke a procedure on some model object. This latter capability is helpful for developing tutorials or teaching simulations. A Doc can be also be linked to a specific object, in which case the doc appears in that object's property sheet.

Annotation is the ability to add additional fields of information to objects within Emergent. This capability, which comprises a set of name/value pairs, is supported by `User Data`. Most objects in Emergent can have User Data added to them. User Data is also used internally by Emergent for such things as providing format information in table columns and graphs. User Data can be set and retrieved programmatically, adding a powerful metadata facility that can be utilized by advanced models and their control programs.

Emergent includes a text-based search engine, available in the context menu of all tree browsers, that can find objects or docs based on their content, type, method names, etc.

2.12. Select edits

Simulations can often contain many hundreds of disparately located parameters, only a few of which may be intended for modification or exploration. Emergent provides a construct called a `SelectEdit`, which enables parameters and control buttons from anywhere in the simulation (particularly Spec objects) to be displayed and changed in one convenient panel. A combination of Edits and Docs are especially useful for creating tutorial or teaching simulations, with self-contained documentation and a constrained display of key parameters.

2.13. Projects

All of the previously described objects live in a top-level object called a `Project`. A Project contains the Networks, Programs, Tables, Docs, Edits, and Views of a complete simulation. Projects are stored in a textual form that makes them well suited to management under a version control system. We have found the open-source revision control system `Subversion` (http://subversion.tigris.org/) to be highly convenient for this purpose.

2.14. Batch mode

Using the application in GUI mode is convenient when teaching, developing, or debugging models. But modeling often involves long sessions of model training, which can be better handled by a batch scheduling program. So Emergent can be run in -nogui batch mode, with a variety of command-line parameters provided to control which model gets loaded, as well as providing model-specific parameters.

2.15. Distributed memory cluster support and parallel threading

Emergent has support for Linux/Unix distributed memory clusters using the industry standard MPI protocol (e.g., MPICH, OpenMPI). The most efficient parallel speedups are obtained using a parallel training mode, whereby several parallel instances of the model run at the same time (one per node), but get different training patterns applied to them; the weight change differentials are then combined and applied to all the instances. This is very efficient because there is a lot of parallel computation for each communication event — the large size of this communication is insignificant relative to the overall costs of coordinating the timing of the different processors during the communication. For this reason, a more fine-grained parallelism at the level of individual units distributed across different processors has not proven very efficient — the communication events are too frequent relative to

the amount of computation per event. As the networks get larger, this level of parallelism becomes more effective.

We also have preliminary support for parallel threading of computation across multiple cores or CPU's within a shared memory environment. This can be combined with the distributed memory clustering approach for even greater speedup. The next release will feature a much more robust and pervasive application of parallel threading. We are also currently investigating the use of GPU-based coprocessor boards using NVIDIA's new CUDA technology.

2.16. Plugins

A growing number of modern software applications provide a means for adding user-developed plugins to extend the functionality of the system without requiring a recompilation of the main application itself. An Emergent plugin is a dynamically loadable code library that has been compiled in C++ and linked to the Emergent libraries and their dependent libraries, such as Qt and ODE. Plugins use a Qt-provided cross-platform build system that is simple to use. Building a plugin automatically installs it into a designated plugin folder.

We envision a growing base of Emergent users who will develop plugins for things like: new network algorithms; new types of network objects, such as specialized layers and units; wizards to help automate complex modeling tasks; new procedures for data analysis and transformation; new visualization objects; specialized computational engines to take advantage of new processing capabilities in existing CPUs or co-processors; new forms of input and output; and links or channels to other tools, such as analysis packages, graphing packages, visualization packages and databases.

3. Comparison with other simulators

Emergent is in the company of hundreds of available neural simulators, each filling a certain niche. In order to help users choose a simulator that best suits their needs, we have compiled a detailed comparison (http://grey.colorado.edu/emergent/index.php/Comparison_of_Neural_Network_Simulators) over 25 features of the 15 simulators that we identified as having been the most widely used and developed. This table is available on the Emergent wiki, is community-editable and features spreadsheet-like sorting using javascript. You are encouraged to visit and update the table, as it will always be under development.

Emergent has the longest legacy of any simulator, with the first release of PDP occurring in 1986. Others in this league are GENESIS (Beeman et al., 2007), with releases from 1988–2007, and NEURON (Migliore et al., 2006), with releases from the early 90s to 2008. Both GENESIS and NEURON specialize in the modeling of individual neurons and networks of neurons at a high resolution, computationally expensive but very biologically plausible sub-cellular level of analysis. Although these two tools have somewhat limited user interfaces, the sophisticated visualization tool neuroConstruct (http://www.physiol.ucl.ac.uk/research/silver_a/neuroConstruct/) encapsulates both of them, providing unparalleled graphics. Other simulators in the spiking neuron domain include NEST, The NeoCortical Simulator (NCS), The Circuit SIMulator (CSIM), XPPAUT, SPLIT, and Mvaspike. A detailed side-by-side comparison and benchmarking of these simulators can be found in Brett et al. (2007).

Simulators that are more directly comparable to Emergent include the Stuttgart Neural Network Simulator (SNNS) (Petron, 1999), the Topographica Neural Map Simulator (Bednar et al., 2004) and the Fast Artificial Neural Network Library (FANN) (Nissen, 2003). SNNS has implemented an impressive array of algorithms, more than any other simulator to date. If trying out lots of new neural network algorithms is your goal, we highly recommend SNNS. Unfortunately, SNNS is no longer under active development, does not have an active support community and has an aging interface. If these things matter to you, we recommend Emergent. Topographica focuses on the development of high-level, Kohonen-like topographic maps of the sensory and motor areas with an emphasis on the analysis and visualization of topographically organized regions. FANN will be useful to researchers who require a fast implementation of backpropagation or SOM, native bindings to multiple scripting languages, and an active support community. If you need a neural network to do a specific computational task with low overhead, FANN may have benefits. LENS, while no longer actively supported, also features fast implementations of backpropagation algorithms.

Several proprietary and commercial simulators exist, including the Neural Networks package for Mathematica (http://www.mathworks.com/products/neuralnet/), the MATLAB Neural Network Toolbox (http://www.mathworks.com/products/neuralnet/) and Peltarion Synapse (http://www.peltarion.com/products/synapse/). The Mathematica and MATLAB packages will be of interest to those who are already familiar with those tools or want access to Matlab's powerful linear algebra facilities in addition to a wide variety of community developed libraries. Peltarion Synapse is a sophisticated package that will be most useful to those interested in data mining.

4. Future work

Emergent is under constant development and a number of improvements are on the horizon. We plan to implement an undo operation to complement copy and paste, an autosave feature and better support for keyboard shortcuts. The build system will be ported from GNU Autoconf to the more modern CMake, and the Windows development environment will be upgraded to Visual Studio 2008. 64-bit support has already been implemented for Linux—we soon plan to support it on OSX and Windows as well. We will conduct further experiments using Graphics Processing Units (GPUs), testing the speed of sender-based computations and the feasibility of running them on a cluster of GPUs. Native support for managing projects in a Subversion repository will be added, in addition to an interface to ModelDB (http://senselab.med.yale.edu/modeldb/). Finally, the existing TCP/IP code which allows Emergent to be started in server mode and controlled by external applications will be enhanced.

5. Conclusion

Emergent's 4.0 series of releases is a turning point in the history of its development. With a renewed focus on usability, extensibility, cross-platform support and visualization, Emergent is now accessible to a far wider audience than was PDP++. Using this new workspace, the process of creating models has become efficient, making modelers more productive and allowing them to create more complicated, yet more understandable, cognitive models than previously possible. Those who invest time in learning it will be more than paid back. Emergent boasts an active community of users and contributing developers, an active development cycle going back more than two decades, a diverse set of algorithms, a powerful visual programming paradigm and a flexible and capable virtual environment for robotics. Emergent is also a friendly teaching aide and is being actively used to teach cognitive modeling courses in many universities (see CECN Projects (http://grey.colorado.edu/CompCogNeuro/index.php/CECN1_Projects) for a set of over 40 different research-grade teaching models that accompany the *Computational Explorations in*

Cognitive Neuroscience textbook (O'Reilly & Munakata, 2000)). In short, through judicious use of modern programming libraries and interface design principles, combined with its legacy, Emergent has become one of the most capable and user-friendly general-purpose neural network simulators available. We invite you to give Emergent a try, and welcome you to join and contribute to the `mailing list` (http://grey.colorado.edu/cgi-bin/mailman/listinfo/emergent-users).

References

Ackley, H., Hinton, E., & Sejnowski, J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science*, *9*, 147–169.

Almeida, B. (1987). A learning rule for asynchronous perceptrons with feedback in a combinatorial environment. In M. Caudil & C. Butler (Eds.), *Proceedings of the IEEE first international conference on neural networks* (pp. 609–618).

Bednar, J. A., Choe, Y., Paula, J. D., Miikkulainen, R., Provost, J., & Tversky, T. (2004). Modeling cortical maps with Topograhica. *Neurocomputing*, *58*, 1129–1135.

Beeman, D., Wang, Z., Edwards, M., Bhall, U., Cornelis, H., & Bower, J. (2007). The GENESIS 3.0 Project: A universal graphical user interface and database for research, collaboration, and education in computational neuroscience. *BioMed Central Neuroscience*, *8*.

Brett, R., Rudolph, M., Carnevale, T., Hines, M., Beeman, D., Bower,, et al. (2007). Simulation of networks of spiking neurons: A review of tools and strategies. *Journal of Computational Neuroscience*, *23*(3), 349–398.

Kohonen, T. (1984). *Self-organization and associative memory*. Berlin: Springer-Verlag.

Hochreiter, S., & Schmidhuber, J. (1997). Long short term memory. *Neural Computation*, *9*, 1735–1780.

McClelland, J. L., & Rumelhart, D. E. (1986). *Parallel distributed processing*. Cambridge, MA: MIT Press.

Migliore, M., Cannia, C., Lytton, W., Markram, H., & Hines, M. L. (2006). Parallel network simulations with NEURON. *Journal of Computational Neuroscience*, *21*(2), 119–129.

Nissen, S. Implementation of a fast artificial neural network library (FANN). Report. Department of Computer Science, University of Copenhagen (DIKU), 31, 2003.

Norman, K. A., Newman, E., Detre, G., & Polyn, S. (2006). How inhibitory oscillations can train neural networks and punish competitors. *Neural Computation*, *18*, 1577–1610.

O'Reilly, R. C., Frank, M. J., Hazy, T. E., & Watz, B. (2007). PVLV: The primary value and learned value pavlovian learning algorithm. *Behavioral Neuroscience*, *121*(1), 31–49.

O'Reilly, R. C., & Munakata, Y. (2000). *Computational explorations in cognitive neuroscience*. Cambridge, MA: MIT Press.

Petron, E. (1999). Stuttgart Neural Network Simulator: Exploring connectionism and machine learning with SNNS. *Linux Journal*, *63*.

Pineda, J. (1987). Generalization of back-propagation to recurrent neural networks. *The American Physical Society*, *59*(19), 2229F.

Rumelhart, D. E, Hinton, G. E., & Williams, R. J. (1986). Learning internal representations by error propagation. In E. Rumelhart, L. McClelland, & PDP Research Group (Eds.), *Foundations*: *Vol. 1. Parallel distributed processing* (pp. 318–362). Cambridge, MA: MIT Press.

Rumelhart, E., & Zipser, D. (1986). Feature discovery by competitive learning. In E. Rumelhart, L. McClelland, & PDP Research Group (Eds.), *Foundations*: *Vol. 1. Parallel distributed processing* (pp. 151–193). Cambridge, MA: MIT Press (Chapter 5).

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, *1*(2), 270–280.

Exploring the Feasibility of Automatically Rating Online Article Quality

Laura Rassbach
University of Colorado
Department of Computer Science
Boulder, CO 80309-0430

Trevor Pincock
University of Colorado
Department of Linguistics
Boulder, CO 80309-0430

Brian Mingus
University of Colorado
Department of Psychology
Boulder, CO 80309-0430

ABSTRACT

We demonstrate the feasibility of building an automatic system to assign quality ratings to articles in Wikipedia, the online encyclopedia. Our preliminary system uses a Maximum Entropy classification model trained on articles hand-tagged for quality by humans. This simple system demonstrates extremely good results, with significant avenues of improvement still to explore.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

Keywords

Quality, Wikipedia, Maximum Entropy

1. INTRODUCTION

The quality of a literary endeavor is often the first and last thing anybody needs to know about it. People are generally very good at identifying high-quality writing, though some are better than others. The quality of writing is related not only to a piece's readability, but also to its accuracy and informativeness. The subjective nature of quality makes computers very bad at deciding quality, to the delight of the mediocre blogger and the text-generating spam bot. As the amount of electronic data available on the Internet increases, and high-quality writing continues to be overwhelmed by low-quality work, automatic classification of quality becomes more important.

Wikipedia is an Internet-based, free-content encyclopedia. With 1.7 million English articles, Wikipedia dwarfs its next largest competitor, the venerable Encyclopedia Britannica with 120 thousand articles. Officially launched in 2001, Wikipedia's rapid exapansion to become the world's largest English enclyclopedia is remarkable. The size and growth of Wikipedia is due to the efforts of a volunteer army of contributors and editors. Wikipedia's free content policy allows anyone to edit or create articles. This laissez-faire attitude encourages contribution, but also allows poor quality articles, heavily biased and containing misinformation to enter the encyclopedia. The editorial team at Wikipedia monitors the content, but the enormity of the task makes errors inevitable. Wikipedia's founder, Jimmy Wales, has admitted that quality control is an important problem for Wikipedia to address [13].

Despite this weakness, Wikipedia's accuracy has stood up to review. A peer review of Wikipedia and the Encyclopedia Britannica's scientific articles yielded no significant difference in error averages per article between the volumes [6]. Usage of Wikipedia has reached astonishing levels as well. According to a poll by the Pew Research council over 30% of Internet users utilize Wikipedia as a educational resource. The usage trend increases with educational level, as 50% of Internet users with a college degree use the online encyclopedia. Because of its enormous size, Wikipedia is also increasingly becoming a valuable resource in Natural Language Processing. It has been used in tasks such as word sense disambiguation, co-reference resolution, and information extraction [20, 14, 21]. The availability of quality ratings for Wikipedia articles would assist both human users and automatic applications in selecting the best articles for their purposes.

Quality is notoriously hard to quantify [19]. For a multimedia entity, such as Wikipedia, overall quality is the composition of the various parts. Currently there are computational applications which are capable of assessing the quality of some of the components. The GRE utilizes a program to evaluate essays in conjunction with human evaluations [15]. Text in encyclopedic articles does not exactly match that of an essay, but many of the principles are the same (for example, clear and well-written prose is important in both cases). There exist no established criteria for evaluating the quality of other elements within an article, such as the images, time lines, topical hierarchy, citations, and others. Although it is difficult to determine quantitative measures of quality, it is easy for people to determine the relative quality of something. A subset of Wikipedia articles have been assessed and annotated by the community of users. We used the Maximum Entropy machine learning technique to train a classifier on this dataset to automatically evaluate the quality of articles. Despite a limited number of features, we have obtained significant results. The machine learning approach

can reverse-engineer the quality assessments of human annotators. We suggest features to extend the classifier and applications of the research.

2. RELATED WORK

Content creators will necessarily be concerned with quality. Simple extensions of applications that help creators ensure quality become popular immediately. The spell check feature of word processors has saved many papers from embarrassing errors and sounded the death knell for typewriters that didn't perform this basic quality assurance operation. The grammar check feature of some word processors attempts to expand on this very useful tool by incorporating syntactic rules to aid composition. However, the conventions of grammar are more mutable than those of spelling, making these systems error prone. Although these systems are not perfect research indicates that students using word processors produce higher quality work than those who do not [1]. The Writer's Workbench was a program developed in the 1980s that detected things such as split infinitives ("to boldly go"), overly long sentences, wordy phrases and passive sentences [12]. These metrics of quality are relatively objective. The features that make up great writing extend beyond sentence boundaries, are inherently subjective, and are much harder to evaluate. Grammar models that rely on rules or statistical regularity would balk at Shakespeare's work.

Discourse analysis is typically concerned with measurements of cohesion and coherence. Cohesion refers to the relationship between lexical units; coherence is the relationship between the meaning of units of text. Simple measures of cohesion would capture some of the nuances of discourse quality. The TextTiling algorithm proposed by Hearst measures cohesion by segmenting the discourse and measuring the lexical similarity between segments [8]. Latent Semantic Analysis, an algorithm which primarily measures the similarity of documents by word frequency, is another way to measure cohesion. Foltz et al describe using LSA to measure the quality of student essays [4]. Their results indicated that LSA could be used to achieve human accuracy in holistic judgments of quality. The limitation of LSA is that the domain must be well defined and a representative corpus of the target domain must be available.

Witte and Faigley offer a critique of using cohesion as a measurement of writing quality [26]. They argue that writing quality cannot be divorced from context, and factors such as the writer's purpose, the discourse medium and the characteristics of the audience are essential to qualitative analysis. While the presence or the absence of cohesion doesn't confirm or disconfirm quality, it is a useful indicator.

Coherence is a more reliable indicator of quality, but coherence is more difficult to quantify. Centering Theory presents a model of how coherent discourse should be structured. The theory posits that a discourse focuses on a single entity and that all utterances are centered on the entity and the introduction of new objects of focus must be done in relation to the centered objects, and it defines criteria for these transitions and ranks them in preferential order [7]. Miltsakaki and Kukich applied Centering Theory's hypothesis of attentional shifting to essays evaluated by Educational Testing Services' e-rater essay scoring system [15]. They found that the number of Rough-Shifts correlated with a lower score from e-rater. Their dataset had to be hand annotated to represent the roles of constituents in Centering Theory, making such analysis time consuming.

3. METHODOLOGY

3.1 The Dataset

The Wikipedia Editorial Team has begun tagging articles according to their quality. Articles are assigned to one of six quality classes: Featured, A, Good, B, Start, and Stub. Hand annotation is slow, laborious work. Assessments are based on the judgments of the annotators and not a quantitative analysis. There are established criteria for article classification, defining what articles of particular quality should be like. Featured Articles should be "well written, comprehensive, factually accurate, neutral and stable." [9]. Definitions are given for each of these properties, but the inherent subjectivity of the evaluations is obvious. Nearly 600,000 articles have been tagged [10]. The vast majority of articles, 71%, have been classified as Stubs, or articles of very short length containing incomplete material. Most articles begin as stubs and await further content contributions. Stub articles are relatively easy to recognize because of their brevity. However, they aren't very useful for a categorization task of quality, because they lack many of the elements of the more complete articles which are suitable as an educational resource.

The remaining data set of rated articles, with Stub articles removed, contains 168,183 total articles. The ratings for the articles that make up this set are as follows: 132,146 Start (78.6%), 31,600 B (18.8%), 2132 GA (1.3%), 873 A (0.5%), and 1432 FA (0.9%). The distribution is clearly skewed to the lower quality articles, with very few examples of articles the Wikipedia Editorial Team considers to be of "publishable quality."

We processed the rated articles to use in the training and testing of our classifier. The articles required quite a bit of preprocessing before they could be analyzed by our algorithms. We created separate entries in the database for HTML and plain text versions of the articles. The plain text of the articles was acquired using a Python module called BeautifulSoup [23]. The text was segmented into sentences by using MxTerminator, a Java implementation of a maximum entropy model specifically trained for sentence boundary detection [22].

3.2 Maximum Entropy Model

A Maximum Entropy (MaxEnt) model is a supervised machine learning algorithm used for classification, equivalent to a statistical regression algorithm. The algorithm uses a set of manually defined features to attempt to determine the probability of each example being in each class. The term "Maximum Entropy" refers to the fact that this classification is done making a minimum number of assumptions. For example, if the classifier has seen that 50% of training examples with feature 'x' are in class 'A', and has no other information, it will guess that the probability of a new example with feature 'x' being in class 'A' is 50%. Since it

has no other information, the rest of the probability mass will be distributed evenly among the other classes, since to do otherwise would make an assumption about the remaining classes that is not justified from the training data. A MaxEnt classifier works by assigning weights to each feature for each class, the list of features and classes are taken from the training data. For each class, each feature is multiplied by the associated weight and summed with the other features, then normalized to obtain the probability of the example being in the class. Features that interact, for example, 'number of words per paragraph' is an interaction between 'number of words' and 'number of paragraphs', must be manually combined into a single feature; the MaxEnt algorithm does not automatically analyze any interactions between features. The model learns by adjusting these weights iteratively based on the examples in the training set. [11, 3] There are several good reasons to use a MaxEnt classifier for this problem. MaxEnt classifiers are relatively simple and converge quickly, allowing us to experiment with adding new features to see their effect on the classification accuracy. At the same time, they are powerful systems that can succeed at quite difficult problems [16]. Finally, a MaxEnt model is built around the assumption that a human expert can easily identify all of the features likely to give clues about the correct classification of each example. This is an elegant approach to the quality classification problem because the original quality rankings are based on features observed by human experts – in other words, we believe that a set of features is already in use for hand classification, so it makes sense to attempt to use those features for automatic classification [5].

For our system, we used a Maximum Entropy classifier written in C++ with a Python wrapper by Zhang Le [27]. This classifier has a number of features useful for our problem. Unlike many implementations of MaxEnt, it allows the definition of non-binary features. This is convenient because it allows us to enter features such as length as the actual values rather than having to artificially decompose the values into a set of binary features. Our classifier uses the Limited-memory Broyden-Fletcher-Goldfarb-Shanno quasi-Newton algorithm to estimate the model weights [17]. We also use Gaussian Prior Smoothing with a variance of 2 to avoid overfitting the training set.

To train the MaxEnt model, we randomly selected an equal number of pages from each Wikipedia quality category (the actual number of selected pages is 650, 80% of the 'A' class articles). It is important to select an equal number of pages from each category because the model is powerful enough to use the distribution of the training data to assist in classification. Since the 'Start' category is overwhelmingly common, a model trained on a set with the same distribution as the actual page set will simply assign 'Start' to every page, and, having found a local maximum for classification, will cease to examine other features to improve the classification accuracy. Research has shown that altering the distribution of a training set for a machine learning algorithm often improves the performance of the algorithm on the test set [25]. We have artificially created a training set with equal numbers of examples in each category to force the classifier to learn the correct weights for the features we have defined. The weights for our model converge in less than 1000 iterations on this training set.

3.3 Feature Set

MaxEnt classifiers are typically built with extremely large feature sets, often as many as 10,000 features [24, 2]. Due to time constraints and equipment difficulties, we have an extremely limited set of only about 50 features. Our features fall into four general categories: length measures, part-of-speech usage, web-specific features, and readability metrics. Length measures include counts of the number of paragraphs and number of words, and give some hint as to whether the article is complete and comprehensive. Part-of-speech usage measures are counts of particular parts of speech in a syntactic parse of the article. These metrics allow us to begin to analyze the complexity of sentences and the quality of the article's prose. Web-specific features such as number of images and internal links reflect the authors' use of all the resources available for an article, as well as improving ease of understanding for readers. Finally, we used a number of standard readability metrics, including Kincaid, Coleman-Liau, Flesch, Fog, Lix, SMOG, and Wiener Sachtextformel, as another method of measuring the comprehensibility and complexity of an article's prose. These simple features begin to capture many of the qualitative assessments of the Wikipedia Editorial Team. For efficiency reasons, each feature was pre-computed and entered in the database, allowing us to add features and retrain the classifier relatively quickly.

4. RESULTS

A reasonable baseline classification system is one that classifies every article as a 'Start' article, since the overwhelming majority of articles in our dataset are 'Start' articles. This gives a classifier with a 78.6% accuracy. Despite our limited number of features and constraint of the training set to an equal distribution, our current classifier is nearly as accurate as the baseline, at 74.6% accuracy by article (that is, 74.6% of articles in the test set are classified correctly). Interestingly, Start-class articles are by far the least commonly mis-classified, probably because our feature set is most applicable to distinguishing Start articles from other classes. Our accuracy for the other classes is much lower, so that the normalized accuracy of our model is much lower, just under 50% (the normalized accuracy of the baseline model is 20%). However, this problem is at least in part because the other categories are much less distinct than the 'Start' category. If we collapse the category set into three categories instead of five ('Great', containing ratings F and A; 'Good', for rating G; and 'Poor' containing B and G) we see a normalized accuracy of 69% and a non-normalized accuracy of 91%. From the collapsed matrix we can see that 'Good' articles are the hardest to classify, probably because they have many of the characteristics of both 'Great' and 'Poor' articles. We expect to significantly improve the model's accuracy as we add more features, especially for this category.

5. FUTURE WORK

We've demonstrated the feasibility of classifying Wikipedia articles, and with modest improvements we could increase the accuracy of our system. We plan on expanding our classification system to include more features. More Wikipedia-specific analyses should improve performance. Additions to

Table 1: Confusion matrix for all Wikipedia quality categories. Categories along the top are the human-assigned rating; along the side are the ratings assigned by our classifier

		Correct Class				
		F	*A*	*G*	*B*	*S*
Classified As	*F*	0.44770	0.17391	0.20573	0.04835	0.00457
	A	0.16109	0.37267	0.14952	0.09526	0.01860
	G	0.20084	0.14286	0.41251	0.10088	0.02205
	B	0.11297	0.18012	0.14316	0.42347	0.12483
	S	0.07741	0.13043	0.08908	0.33204	0.82995

Table 2: Confusion matrix for collapsed categories. Categories along the top are the (compressed) human-assigned ratings; along the side are the (compressed) ratings assigned by our classifier

		Correct Class		
		Great	*Good*	*Poor*
Classified As	*Great*	0.593114	0.205726	0.046843
	Good	0.186228	0.562036	0.037549
	Poor	0.220657	0.232238	0.915608

the text analysis, such as more detailed analysis of syntactic structure, cohesion and coherence would help our system distinguish between low and high quality article better. Other applications, such as word processing, article retrieval, text summarization and spam detection, would benefit from automatic classification of quality.

Wikipedia articles contain a great deal of idiosyncratic formatting and information. A more thorough analysis of the features of Wikipedia's layout and wiki syntax would help to correctly classify articles. There are thousands of templates available for usage in a given article, and the number of templates used is strongly correlated with article quality. Features that assess the usage of a template would ensure that templates are used appropriately. The categorical organization of Wikipedia also allows for domain-specific analysis, allowing for a more disciplined analysis of word choice, style, and coverage. Every Wikipedia article also contains a history which stores all the edits made to it. The size of the Wikipedia history is roughly 30 times the size of the articles alone. The history would provide information about the creative process behind an article, and articles with more comprehensive histories would be assumed to be of higher quality.

Images are a strong indicator of quality. A colorful, informative illustration can elucidate a difficult concept, but a poorly chosen picture can actually detract from clever prose. Assessing the quality of an image computationally is a difficult endeavor. Judgments of picture quality are based on optical information and context, which is not readily available to a computer. The resolution of an image and the size can give a coarse idea of quality, as can a simple count of the number of images in an article. Digital photographs come with EXIF data that contains information about about camera settings and scene information that could also be relevant. Diagrams and explanatory figures would be more difficult to evaluate and a feature that merely detects their presence might be the most useful.

The PageRank Algorithm, formulated by Brin and Page [18], would be an excellent indicator of quality. The algorithm could be implemented using Wikipedia's internal link data. The more pages which link to a page would be indicative of its importance and indirectly of its quality. A more comprehensive implementation would incorporate pagerank information from the entire web; sites external to Wikipedia linking to a Wikipedia page would certainly suggest the article was of high quality. We are currently in the process of implementing this algorithm.

More sophisticated measures of the text will require additional parses including part of speech tagging, syntactic parsing, and dependency parsing. Such operations are computationally expensive, which would limit the applications of the system. Nonetheless, we plan to implement these features to assess their relevance to the quality of an article.

Our system is currently very good at distinguishing Start articles from all others. This isn't surprising considering the distribution of our dataset. Improving performance on the classification of the higher quality articles will entail differentiation between prose that is clear and grammatically correct and prose that is brilliant. This will require substantial discourse analysis.

We would also like to experiment with using a Support Vector Machine (SVM) for the classification task instead of the Maximum Entropy model. SVMs are similar to MaxEnt models in that they require the explicit definitions of features believed to give clues about the correct classification of examples. However, in contrast to MaxEnt, an SVM does not need interacting features to be explicitly defined. Rather, an SVM experiments with all possible feature combinations during training in order to discover combinations of features that allow an improvement in accuracy [11]. SVMs are often more accurate than MaxEnt models, but take significantly longer to train. In addition, they are often considered less elegant because the combinations they discover could have easily been added by hand, and it seems unreasonable and inefficient to attempt to automatically discover information we already know [5].

Many domains would benefit from automatic quality classification, particularly of text. Within Wikipedia, the automatic system could provide input to users as they are editing an article, suggesting areas of improvement. If the classifier was used on all articles, quality analysis by category would be more complete. Quality assessments of text could also be used for pedagogical purposes, to assist student's writing and provide instantaneous feedback and suggestions in an objective manner. There is the possibility of gaming such a system, but likely the things that would improve a quality rating would also improve the quality of a piece of text. Quality analysis would help in spam detection: most spam bots use automatic text generation, creating poor-quality, incoherent messages. Of course, this could just lead to more eloquent spam messages.

6. CONCLUSION

Classifying the quality of Wikipedia articles is an important task, since it can focus community attention on articles that need the most improvement and direct users to the articles most likely to be correct and informative. We have demonstrated that with minimal features a Maximum Entropy model can do a surprisingly good job of automatically classifying Wikipedia articles by quality. Our current model has an accuracy of 74.6%, which leaves room for improvement, but also shows the problem to be tractable. We enumerated a number of features that would enhance the model's performance.

7. ACKNOWLEDGEMENTS

Wikipedia runs on the custom-built MediaWiki platform, written in PHP and running on top of the MySQL database engine. All of Wikipedia is available for download in various formats, including XML, SQL, and HTML. The XML dump includes both embedded wikitext and metadata. The compressed archive containing all current versions of articles and content is 2.3 GB. A major computational cost of this project was acquiring the data and creating a local copy of Wikipedia to process. We would like to thank the Computation Science Center at CU Boulder for making available the resources for our research.

We would also like to thank Jim Martin and Martha Palmer for teaching us everything we know about Natural Language Processing.

8. REFERENCES

[1] R. L. Bangert-Drowns. The word processor as an instructional tool: A meta-analysis of word processing in writing instruction. *Review of Educational Research*, 63(1):69–93, 1993.

[2] A. Borthwick, J. Sterling, E. Agichtein, and R. Grishman. Exploiting diverse knowledge sources via maximum entropy in named entity recognition, 1998.

[3] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(4):380–393, April 1997.

[4] P. W. Foltz. Supporting content-based feedback in on-line writing evaluation with lsa. *Interactive Learning Environments*, pages 111–127, August 2000.

[5] B. R. Gaines. An ounce of knowledge is worth a ton of data: quantitative studies of the trade-off between expertise and data based on statistically well-founded empirical induction. In *Proceedings of the sixth international workshop on Machine learning*, pages 156–159, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[6] J. Giles. Internet encyclopaedias go head to head. *Nature*, 438(7070):900–901, December 2005.

[7] B. J. Grosz, S. Weinstein, and A. K. Joshi. Centering: a framework for modeling the local coherence of discourse. *Comput. Linguist.*, 21(2):203–225, June 1995.

[8] M. A. Hearst. Texttiling: A quantitative approach to discourse segmentation. Technical Report S2K-93-24, University of California, Berkeley, 1993.

[9] S. W. History. Wikipedia: Featured article criteria, May 2007.

[10] S. W. History. Wikipedia: Version 1.0 editorial team/index, May 2007.

[11] D. Jurafsky and J. H. Martin. *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall, Upper Saddle River, N.J., 2000.

[12] N. Macdonald, L. Frase, P. Gingrich, and S. Keenan. The writer's workbench: Computer aids for text analysis. *Communications, IEEE Transactions on [legacy, pre - 1988]*, 30(1):105–110, 1982.

[13] D. Mehegan. Bias, sabotage haunt wikipedia's free world. *Boston Globe*, February 2006.

[14] R. Mihalcea. Using wikipedia for automaticword sense disambiguation. In *Proceedings of NAACL HLT 2007*, page 196.203. Association for Computational Linguistics, 2007.

[15] E. Miltsakaki and K. Kukich. Evaluation of text coherence for electronic essay scoring systems. *Nat. Lang. Eng.*, 10(1):25–55, March 2004.

[16] K. Nigam, J. Lafferty, and A. Mccallum. Using maximum entropy for text classification, 1999.

[17] J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980.

[18] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[19] R. Pirsig. *Zen and the art of motorcycle maintenance :an inquiry into values*. Morrow, New York, 1974.

[20] S. P. Ponzetto. Creating a knowledge base from a collaboratively generated encyclopedia. In *Proceedings of NAACL HLT 2007*, pages 9–12. Association fo Computational Linguistics, 2007.

[21] S. P. Ponzetto and M. Strube. Creating a knowledge base from a collaboratively generated encyclopedia. In *Proceedings of NAACL HLT 2006*, pages 192–199. Association fo Computational Linguistics, 2006.

[22] J. C. Reynar and A. Ratnaparkhi. A maximum entropy approach to identifying sentence boundaries. In *Proceedings of the fifth conference on Applied natural language processing*, pages 16–19, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc.

[23] L. Richardson. *Beautiful Soup Documentation*, April 2007.

[24] R. Rosenfeld. A maximum entropy approach to adaptive statistical language modeling, 1996.

[25] G. Weiss and F. Provost. The effect of class distribution on classifier learning, 2001.

[26] S. P. Witte and L. Faigley. Coherence, cohesion, and writing quality. *College Composition and Communication*, 32(2):189–204, 1981.

[27] L. Zhang. *Maximum Entropy Modeling Toolkit for Python and C++*, December 2004.

frontiers in
PSYCHOLOGY

HYPOTHESIS AND THEORY ARTICLE
published: 18 June 2012
doi: 10.3389/fpsyg.2012.00182

The role of competitive inhibition and top-down feedback in binding during object recognition

Dean Wyatte*, Seth Herd, Brian Mingus and Randall O'Reilly

Department of Psychology and Neuroscience, University of Colorado Boulder, Boulder, CO, USA

Edited by:
Snehlata Jaswal, Indian Institute of Technology Ropar, India

Reviewed by:
Rufin VanRullen, Centre de Recherche Cerveau et Cognition Toulouse, France
Hubert D. Zimmer, Saarland University, Germany

****Correspondence:***
Dean Wyatte, Department of Psychology and Neuroscience, University of Colorado Boulder, 345 UCB, Boulder, CO 80309-0345, USA.
e-mail: dean.wyatte@colorado.edu

How does the brain bind together visual features that are processed concurrently by different neurons into a unified percept suitable for processes such as object recognition? Here, we describe how simple, commonly accepted principles of neural processing can interact over time to solve the brain's binding problem. We focus on mechanisms of neural inhibition and top-down feedback. Specifically, we describe how inhibition creates competition among neural populations that code different features, effectively suppressing irrelevant information, and thus minimizing illusory conjunctions. Top-down feedback contributes to binding in a similar manner, but by reinforcing relevant features. Together, inhibition and top-down feedback contribute to a competitive environment that ensures only the most appropriate features are bound together. We demonstrate this overall proposal using a biologically realistic neural model of vision that processes features across a hierarchy of interconnected brain areas. Finally, we argue that temporal synchrony plays only a limited role in binding – it does not simultaneously bind multiple objects, but does aid in creating additional contrast between relevant and irrelevant features. Thus, our overall theory constitutes a solution to the binding problem that relies only on simple neural principles without any binding-specific processes.

Keywords: binding, competitive inhibition, feedback, computational model, object recognition

INTRODUCTION

The term "binding" has several meanings within psychology and neuroscience. The central assumption is that partial representations must in some way be "bound" together into a full representation (Treisman, 1996, 1999). In particular, the term is used in the context of visual processing; however, the issue is relevant in understanding brain and psychological mechanisms in general. The need for binding mechanisms is highlighted by the fact that neurons early in the visual system respond to (and therefore represent) simple visual features while meaningful objects consist of very particular conjunctions of many of these features (e.g., perpendicular lines meeting at their ends compose corners; corners that line up compose rectangles, etc.). Some mechanism appears to be needed to track which of these features belong together; that is, which ones originated from a coherent construct in the real world, and so should be combined to produce an accurate and meaningful internal representation of that construct.

We seek here to clarify the neural mechanisms involved in the process of binding. In doing so, we describe a theory of how binding can be explained using only simple, generic principles of neural processing. Our perspective on binding has much in common with that of other theorists (Reynolds and Desimone, 1999; Shadlen and Movshon, 1999; Treisman, 1999; Bundesen et al., 2005). In fact, the amount of convergence on the binding problem in recent years is striking; the novelty of our contribution is therefore largely in adding specificity to these proposals in terms of the biological mechanisms that underlie binding in the brain.

Our core proposal is that competitive neural inhibition, combined with top-down feedback and learned selectivity for some features over others, accounts for binding in the brain. More specifically, the computational role of inhibition and top-down feedback in binding is to ensure that only neurons with the most support become substantially active and ultimately drive behavior. Cortical inhibition thus performs contrast enhancement by suppressing activity of neurons with significant but lower levels of excitatory input (Kandel et al., 1995; Carandini and Heeger, 2012). Neurons tuned to the less relevant information (such as features from objects outside the focus of attention) are thus out-competed, and so downstream neurons respond only to the most relevant "winning" features.

Top-down feedback supplies an extra set of criteria for which features are most relevant in a given context, supplying useful biases to this competition (Desimone and Duncan, 1995). Top-down feedback can thus be contrasted with feedforward, stimulus-driven signals, that mainly convey information about the sensory environment. However, the neural mechanisms that underlie these two information pathways are exactly the same: standard excitatory synaptic inputs (O'Reilly, 1996; O'Reilly and Munakata, 2000). Putative top-down signals include those from frontal and parietal areas that direct spatial attention (Thompson et al., 2005; Bressler et al., 2008), and those from prefrontal areas that convey information related to the current task or goals (Miller and Cohen, 2001), but might also include those originating from areas only slightly higher up in the visual system that convey "working hypotheses" as to object identities or higher-level features

(Fahrenfort et al., 2007; Boehler et al., 2008; Roland, 2010; Koivisto et al., 2011). In each case, the type of information and therefore the exact constraints supplied to the competition are different; but the fundamental computational role in guiding the local competitions that lead to binding the most relevant features is the same.

We motivate our proposal with a recent review by Vanrullen (2009), which posits two distinct types of binding. One is an "on-demand" process for binding together simple but arbitrary feature dimensions into conjunctive representations (e.g., a red circle stimulus in a visual search experiment contain both "red" and "circular" features). Much of research on binding to date has involved visual tasks that use these arbitrary feature conjunctions which have been proposed to be solved by top-down attentional mechanisms as well as inhibitory mechanisms (Treisman, 1996, 1999; Reynolds and Desimone, 1999). A second type of binding, referred to as "hardwired" binding, involves grouping together pre-established conjunctions of features. Experiments using visual object categorization have been used to motivate the need for hardwired binding, with the major finding being that they proceed rapidly in the absence of top-down attentional mechanisms (Riesenhuber and Poggio, 1999b; Serre et al., 2007; Vanrullen, 2007).

We focus here on the case of hardwired binding. However, we propose that the same mechanisms involved in on-demand binding are also present during hardwired binding. Inhibition and top-down feedback interact to select only the most relevant elements of visual features for further processing, eliminating less contextually relevant features, thus minimizing binding errors. We argue that these mechanisms are just as important for activating the learned feature combinations used in visual object recognition as they are in visual tasks involving arbitrary feature combinations.

Thus, our approach focuses on the binding problem inherent in the problem of object recognition, but applies to the problem more generally. When presented with visual information, whether it be in the context of a single isolated object or an array of multiple objects, the brain relies on the same basic neural mechanisms to form a coherent (properly bound) representation. While abstract cognitive strategies may be important for dealing with different tasks (e.g., visual search), it is unlikely that they are implemented differently at the neural level or require special binding processes. Instead, they operate on the same basic representation formed by simple visual processing.

We explicitly demonstrate our proposal using a biologically realistic model of visual processing (O'Reilly et al., under review; see Methods for overview). We demonstrate three particular aspects of our proposal in the context of a realistic object recognition task that requires binding together learned object features into a single, coherent object (i.e., part binding; Treisman, 1996). First, we show how neurons that code complex visual features compete during processing over the full course of recognition. Inhibitory competition ensures that only the most relevant features are active, while less relevant ones are ultimately suppressed. We further show that systematically reducing the number of category-relevant visual features in the stimulus by an occlusion degradation weakens these competition effects, ultimately causing binding errors in which relevant and irrelevant features become co-active in the bound representation. Second, we show how top-down feedback reinforces category-relevant features, including those that may have been weakened by degrading factors like occlusion, providing some robustness to binding errors. Finally, we investigate the case of multiple object recognition, which has special importance in the study of binding as it can produce illusory conjunctions of features across objects (Treisman, 1996, 1999). We find that the same mechanisms of inhibitory competition and top-down feedback contribute to solving the problem of properly binding learned features when selecting among multiple objects.

The novelty of our contribution to the ongoing discussion on binding is a synthesis between binding and object recognition theories using only the general neural mechanisms of neural inhibition and top-down feedback. Others have put forth similar solutions to the binding problem using only general neural mechanisms (e.g., Reynolds and Desimone, 1999; Bundesen et al., 2005), and we expand on this work with explicit simulations that make predictions about the temporal dynamics of these mechanisms during a hardwired binding task. Our theory can be contrasted with more complex theories of binding, especially those that involve multiplexed neural synchrony (e.g., Singer, 1993, 1999; Singer and Gray, 1995; Uhlhaas et al., 2009). While there might be additional binding-related phenomena (such as those involving working memory; see Raffone and Wolters, 2001) that require such mechanisms, the standard object recognition functions of visual cortex targeted by existing work on binding appear to only require the mechanisms that we focus on here. We conclude by discussing some of the predictions and limitations of our model with respect to other binding theories.

NEURAL INHIBITION SUPPRESSES IRRELEVANT INFORMATION

In the simplest sense, a bound representation in the brain consists of the current set of actively represented features. The brain represents information in a code distributed across a large number of neurons (Kandel et al., 1995), and thus, can represent many features simultaneously. Binding errors can thus occur when features that belong to different objects in the external world are incorrectly bound together into the brain's representation of a single object. To minimize binding errors, the brain relies on several mechanisms to ensure that only the features that belong together get bound together in the long run. One such mechanism is neural inhibition.

Within a given brain area, only a small percentage of neurons are ever active at any given time. One reason for this is that cortical neurons inhibit each other through disynaptic connections with local inhibitory neurons. These inhibitory interneurons are known to perform the function of limiting overall activity levels throughout cortical areas. Within an area, connections to and from inhibitory neurons seem to be relatively non-selective (Swadlow and Gusev, 2002), making this competitive effect general: every excitatory neuron competes with every other excitatory neuron, to roughly the same extent. This picture of inhibitory function is, of course, somewhat oversimplified, but it is sufficient to capture the role neural inhibition in solving the binding problem. This competitive inhibition is one mechanism of contrast enhancement (Carandini and Heeger, 2012), and it is useful to think of the

mechanism as enhancing contrast between firing rates of neurons representing more- and less-appropriate features.

As an example that illustrates the role of inhibition in hard-wired binding, we use the LVis model described in O'Reilly et al. (under review) to demonstrate how the brain binds together a visual representation of a fish for recognition (see Methods for model details). Visual object recognition is thought to be subserved primarily by inferotemporal (IT) cortex, which responds to moderately complex visual features (Logothetis et al., 1995; Tompa and Sary, 2010). IT cortex contains a columnar organization (Tanaka, 1996; Tompa and Sary, 2010), in which columns of neurons that subtend horizontal patches of the cortex code different visual features. While the specific dimensions of stimuli to which a given IT column respond are not yet well-understood (Kourtzi and Connor, 2010), IT neurons can be conceptualized as responding to object "parts" that represent a specific object exemplar at the population level (i.e., combination coding, Ungerleider and Bell, 2011).

As a concrete example, one column of IT neurons might be tuned to a fish's fin, ideally firing when in the presence of a viewed fish. A neighboring column might be tuned to a completely different visual feature such as a bird's wing, and thus should be silent when viewing the fish. These columns project onto inhibitory interneurons that create competition among columns (Mountcastle, 1997), effectively making some combinations of columns mutually exclusive.

In **Figure 1**, we show the firing patterns of simulated columns of IT neurons when presented with a fish stimulus. Initially, a large number of IT neurons fire, some of which belong to columns that code fish-relevant features and some of which belong to columns that do not. The columns selective to fish-relevant features (e.g., a fish fin, a fish tail), however, quickly out-compete columns selective to fish irrelevant features since the former constitute a better fit with the fish stimulus, increasing their initial evoked response. In turn, the columns selective to fish features inhibit columns selective to irrelevant features, effectively stopping irrelevant neurons from firing and becoming part of the bound representation. Thus, competitive inhibition among detected features helps ensure that a valid combination of features ultimately is bound by driving firing of IT neurons, eliminating invalid conjunctions of features that might lead to binding errors.

Inhibition might be especially important when visual objects are highly ambiguous. We demonstrate this idea in **Figure 1** by partially occluding the presentation of a fish, which removes diagnostic visual features and impairs recognition accuracy. Other conditions may also create stimulus ambiguity, such as a non-standard view of an object (such as a fish's underbelly), or an atypical exemplar (an exotic fish, perhaps). Visual occlusion, however, allows us to parametrically measure the effects of ambiguity on activity levels of IT neurons in our model. The general effect of occlusion is an attenuation of the category selective IT response due to the decreased stimulus-driven signal, a finding that has

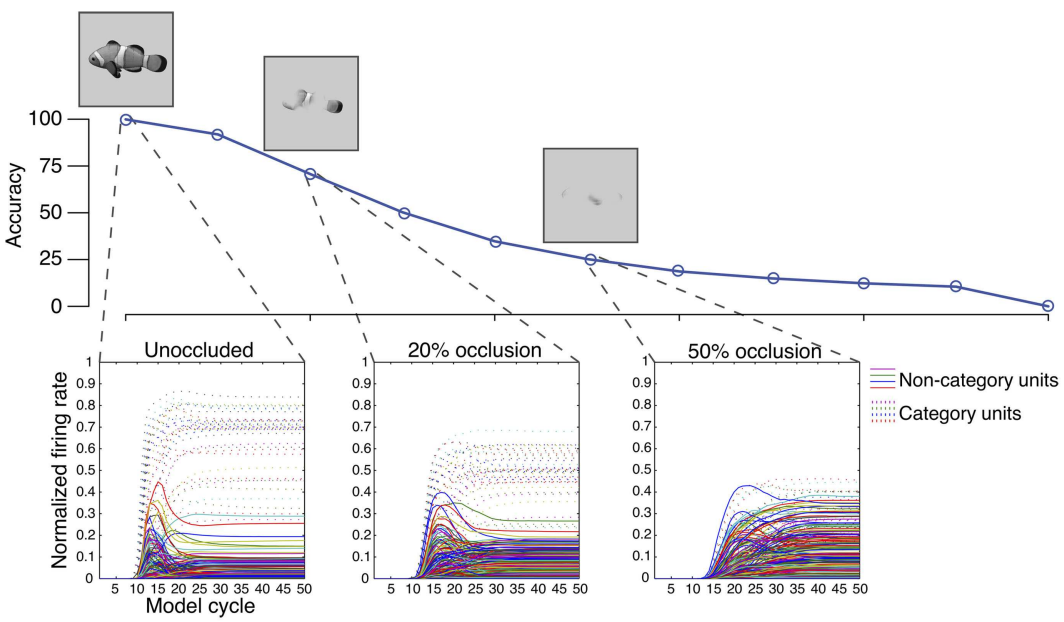

FIGURE 1 | Neural inhibition in visual binding. We use the LVis model described in O'Reilly et al. (under review) to demonstrate how IT level visual features are suppressed by inhibitory mechanisms over the course of visual processing. tbfTop: Visual occlusion was varied as an independent variable to measure its effect on IT firing patterns during object categorization. Increased occlusion results in a monotonic impairment in categorization accuracy. Bottom: Firing rates were recorded for each IT unit in the model and grouped according to whether they were strongly tuned to the fish category exemplars (dotted lines) or tuned to other categories (solid lines). The first wave of responses from the model's IT units area code a large number of features, only some of which are category-relevant. Inhibitory competition, however, suppresses the responses of irrelevant non-category units, leaving the features coded by relevant category units to compose the final bound representation. This competitive advantage disappears at higher levels of occlusion (e.g., 50% occlusion) due to fewer category-relevant features being specified in the stimulus.

been also demonstrated in neurophysiological studies of occlusion (Kovacs et al., 1995; Nielsen et al., 2006). Moreover, because neurons in category selective columns fire at a lower rate, they indirectly exert weaker levels of inhibition toward competing columns. The result is an overall increase in the response of neurons that are selective to category irrelevant features. Thus, both the weakened response to category-relevant features and the erroneous heightened response to irrelevant features may play a role in binding errors when stimulus conditions are highly ambiguous, leading to impaired recognition accuracy.

TOP-DOWN FEEDBACK REINFORCES RELEVANT INFORMATION

It is well-known that the brain contains numerous top-down connections that descend from higher levels of brain systems to lower levels (Felleman and van Essen, 1991; Scannell et al., 1995; Sporns and Zwi, 2004; Sporns et al., 2007). In the context of vision, one commonly suggested function of top-down connections is to convey attentional signals to sensory based areas of visual cortex. These top-down signals can take the form of spatial attention (originating in the frontal eye fields and posterior parietal cortex, Thompson et al., 2005; Bressler et al., 2008) or executive attentional control (as enacted by maintained representations in prefrontal cortex; Miller and Cohen, 2001; Herd et al., 2006).

In the case of spatial attention, top-down feedback about the attended region of space determines which features are relevant by selecting for features within a small spatial area and enhancing them relative to features from neighboring, unattended areas of space. Top-down feedback reflecting executive attentional control works the same way, except that relevancy is determined by more abstract feature dimensions such as color or category (Maunsell and Treue, 2006).

In either case, top-down feedback does not require any representation of what to exclude. Instead, it simply signals what to attend to by providing additional excitatory bias to the sensory representations, causing the representative neurons to fire more strongly. This bias reinforces the activation of relevant features, encouraging their binding at the highest levels of processing. This explanation of attention is a further explication of the biased competition framework of (Desimone and Duncan, 1995), and has been supported by considerable empirical evidence, most notably that of Reynolds and colleagues (see Reynolds and Chelazzi, 2004, for a review).

While top-down feedback has been shown to be crucial for on-demand binding tasks that require the cognitive flexibility to bind arbitrary features together at arbitrary locations (Treisman, 1996, 1999), it is not yet understood whether top-down feedback similarly plays a role in hardwired binding tasks like object recognition and categorization. Computational models have suggested that these tasks can be solved in the brain in a primarily feedforward manner with little to no influence from top-down feedback (Riesenhuber and Poggio, 1999b; Serre et al., 2007; Vanrullen, 2007, 2009). However, there are a number of reports of top-down feedback playing a fundamental role in early visual processes including object recognition (Bar et al., 2006; Fahrenfort et al., 2007; Boehler et al., 2008; Roland, 2010; Koivisto et al., 2011).

In an attempt to reconcile these data, we recently described a computational model of object recognition that contains both feedforward and feedback connections between feature processing layers (O'Reilly et al., under review). One of the key findings, which we review here, is that top-down feedback promotes robust recognition when bottom-up signals are weak and ambiguous due to occlusion (**Figure 2**). While occlusion generally attenuates

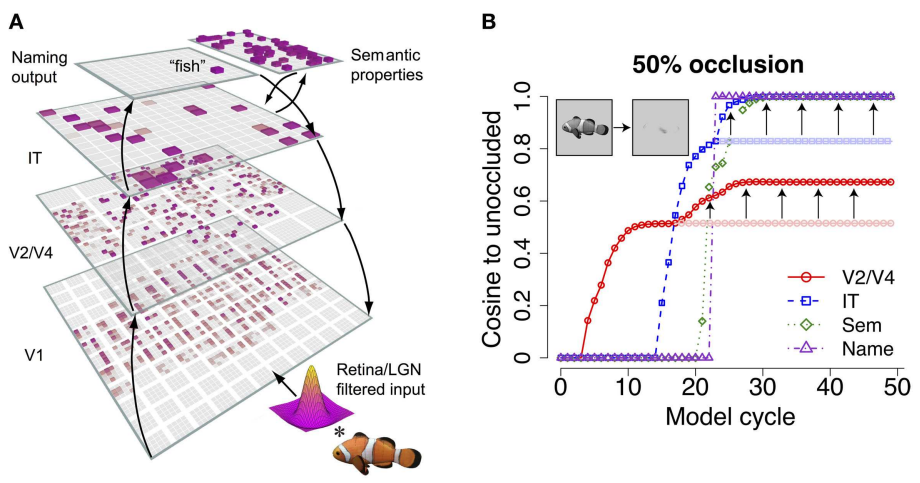

FIGURE 2 | Top-down feedback in "hardwired" binding tasks. (A) As a general rule, the visual system contains bidirectional (both bottom-up and top-down) connections between any two connected areas. The LVis model, depicted here, contains a similar organization, with recurrent connections between hierarchically adjacent areas. **(B)** Top-down feedback from higher levels enhances the neural responses in lower-level areas, which is crucial for robust binding when stimuli are occluded or otherwise ambiguous. Arrows indicate the enhancement in the representation with respect to the veridical (i.e., unoccluded) representation at different areas within the model. Pale colors indicate the predicted response without top-down feedback, which is asymptotic. The Sem area exhibits a similar enhancement pattern to areas V2/V4 and IT, but is left unannotated on the plot for clarity. V2/V4 = extrastriate cortex; IT = inferotemporal cortex; Sem = amodal semantic responses, such as those demonstrated by anterior IT neurons; Name = named output responses.

neural responses resulting in reduced recognition accuracy, the model often exhibits intact category selective responses and correct recognition, a property that we attribute to top-down feedback. Specifically, top-down reinforcement enhances the responses of neurons at lower levels that may have been weakened due to occlusion. This enhancement is repeated across multiple recurrently connected areas, essentially recovering the occluded visual features and resulting in a complete representation. Conceptually, visible features like the fish's dorsal fin might evoke a partial response in IT cortex, which could provide reinforcement to the encoding of other relevant features that might not be visible at lower levels like V2 or V4. Similarly, entertaining the possibility that one might be viewing a fish (i.e., partial activation at the "Naming Output" level of our model) can reinforce fish-relevant features encoded by IT columns. Functional neuroimaging experiments have indicated that the brain exhibits a similar object completion process in which visual information is recovered despite its omission from a visual stimulus (Kourtzi and Kanwisher, 2001; Lerner et al., 2004; Johnson and Olshausen, 2005; Juan et al., 2010).

BINDING MULTIPLE OBJECTS

Thus far we have focused on the problem of binding visual features into a singular, coherent object, and have proposed that both neural inhibition and top-down feedback play important roles in this process. Do these same mechanisms aid in proper binding when multiple objects are present in a display? Proper binding when multiple objects are present is a challenging problem because high-level visual areas such as IT cortex have receptive fields that span large portions of the visual field (generally 10° to 20°; Kobatake and Tanaka, 1994; Rust and Dicarlo, 2010). Thus, IT neurons respond, by default, to visual features regardless of where they are within the visual display, even when they occur in the context of a second object's features. Although the large receptive fields of IT neurons are thought to be necessary for promoting tolerance to changes in object position, scale, and rotation (Logothetis et al., 1995; Tanaka, 1996; Riesenhuber and Poggio, 2002; Rolls and Stringer, 2006), they exacerbate the possibility of illusory conjunctions being formed between the features of separate objects.

We propose that neural inhibition combined with top-down feedback can solve the problem of binding when multiple objects are present in a similar manner to the way they aid in binding visual features into singular, coherent objects. We demonstrate the plausibility of this idea in **Figure 3**. As is the case with single objects presented in isolation, a large number of IT neurons fire initially when multiple objects are present. Grouping these neurons according to the object to which they are selective illustrates the interactions between inhibition and top-down feedback. Generally, neurons that code visual features shared by both objects are the first to respond, since they constitute the best overall fit with the stimulus itself. In the case of the gun and bicycle pictured in **Figure 3A**, these first responders might be neurons that code the horizontal edges that compose the barrel of the gun and the top tube of the bicycle. Neurons that code unique features for each of the object categories are the next to respond. However, inhibition between these columns of neurons ensures that the features of only one of these objects are selected in the end, "winning" the competition (in this case, the bicycle neurons) and contributing to the final bound representation. When a single object is selected for the bound representation, top-down feedback can reinforce neurons that code meaningful features from that object that may not have initially responded (possibly due to initial inhibitory influences from neurons corresponding to the "losing" object).

Binding errors can occur when neurons representing irrelevant features are not entirely out-competed (**Figure 3B**). This allows invalid feature conjunctions to manifest, which subsequently get reinforced from top-down feedback, resulting in the formation of illusory conjunctions. To determine more specifically how inhibition and top-down feedback contribute to minimizing illusory conjunctions, we tested the effect of removing top-down feedback and both top-down feedback and inhibition from the model[1] (see Methods for details). The results of these tests are indicated in **Figure 4**.

For the LVis model (which contains both inhibition and top-down feedback), illusory conjunctions occurred on only 4.7% of trials. Removing top-down feedback, but leaving inhibition intact, had virtually no effect on the number of illusory conjunctions. However, removing both top-down feedback and inhibition caused illusory conjunctions to occur with much higher frequency, on 19.3% of trials.

We also computed the ratio of relevant IT responses to irrelevant responses (where relevance was determined by whether the responses corresponded to the model's output) which can be thought of as a kind of "signal-to-noise ratio" (**Figure 4B**). A decrease in this number reflects lower proportions of relevant information and higher proportions of irrelevant information at the IT level, which could lead to more illusory conjunctions being made. Accordingly, the purely feedforward model, which made the most recognition errors, also exhibited the lowest ratio of relevant to irrelevant information.

Removing feedback from the LVis model also lowered the ratio of relevant to irrelevant information, but recognition performance remained unchanged. This suggests that there is a critical signal-to-noise ratio (in terms of relevant and irrelevant responses) above which recognition remains robust, without many illusory conjunctions. Inhibition was intact in this model, consistent with our proposal that inhibitory competition is the critical mechanism that selects relevant information over irrelevant information, thus providing a relatively stable baseline signal-to-noise ratio. Top-down feedback can further highlight relevant information, increasing the signal-to-noise ratio, but it is unnecessary for well-learned tasks with little ambiguity. Top-down feedback is likely more important in tasks where objects are degraded (e.g., from visual occlusion), which we discussed in the previous section, or in cases where there is more feature overlap across items (e.g., conjunctive visual search).

GENERAL DISCUSSION

We have presented an account of binding in the brain that depends only on well-established mechanisms of neural processing that

[1]Note that it is impossible to test the remaining condition in which top-down feedback is left intact but inhibition is removed from the model, as some mechanism is necessary to control the overall response levels, which would saturate quickly with repeated processing.

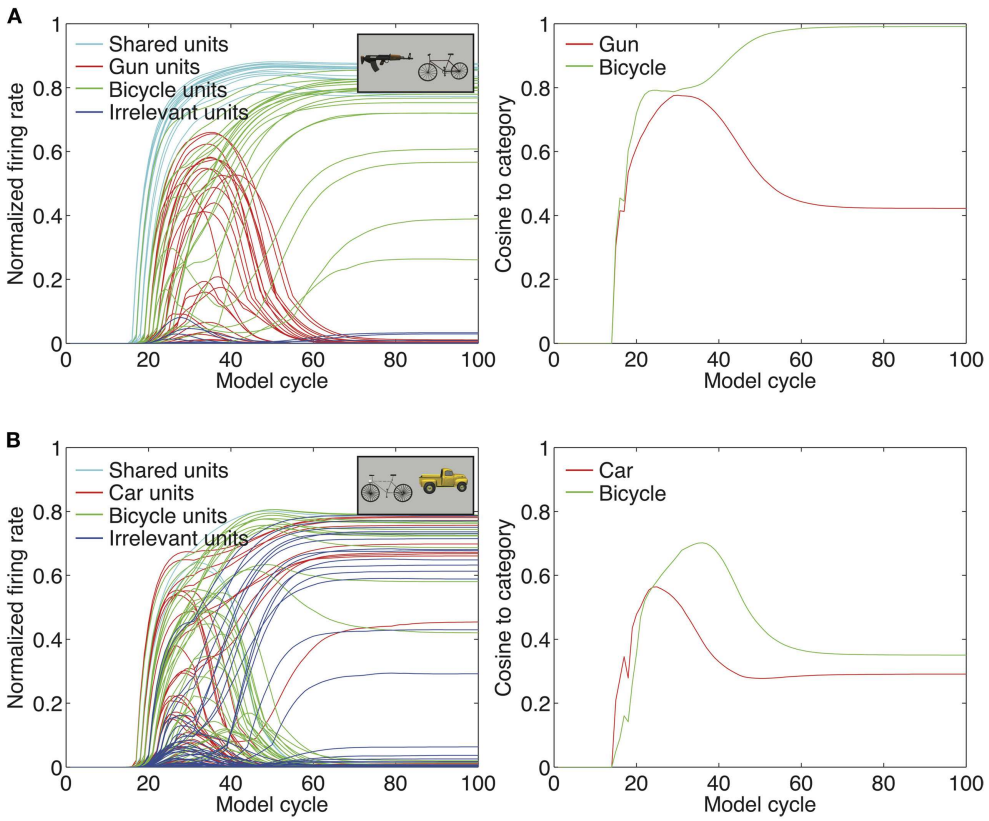

FIGURE 3 | Binding multiple objects. (A) The same mechanisms of neural inhibition and top-down feedback extend to binding when multiple objects are present in a display. The competition created from having multiple IT units active that represent multiple objects causes one set of units to "win" and one set to "lose" (in this case, the bicycle units win). Inhibition suppresses the responses from units corresponding to the losing object as well as responses from completely irrelevant units. Top-down feedback serves to reinforce units from the winning object that may not have been initially active. **(B)** Binding errors occur when completely irrelevant units become erroneously active, leading to the inability to suppress invalid responses. This creates illusory conjunctions of features across the objects in the display, leading to a representation that does not resemble either category.

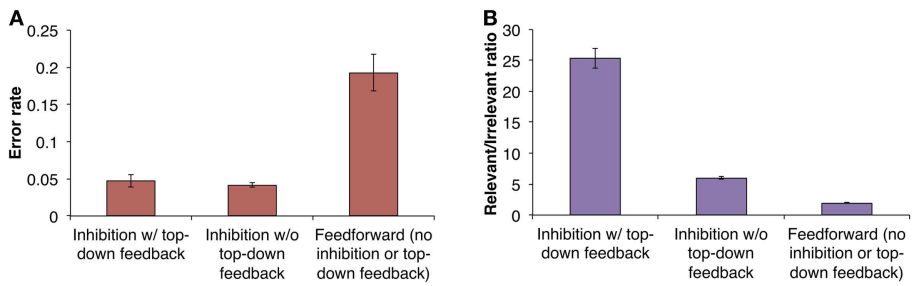

FIGURE 4 | Results for multiple object binding. (A) We tested the effect of removing top-down feedback and both top-down feedback and inhibitory competition from the model. The purely feedforward model missing both of these critical mechanisms made the most recognition errors. **(B)** Grouping responses according to whether they were corresponded to the model's output (relevant responses) or not (irrelevant responses) suggests that the reason for the purely feedforward model's poor performance was that it had a higher overall signal-to-noise ratio (mean relevant response divided by mean irrelevant response). This type of representation could lead to illusory feature conjunctions and thus, recognition errors.

interact over time. Two such mechanisms that we focus on here are neural inhibition and top-down feedback. Together, these mechanisms create an environment of local competition within a brain area that selects only the most relevant features for the bound representation that influences perception and behavior. We have taken a general neural processing approach to explaining how these mechanisms relate to binding. We illustrate the mechanisms explicitly in an object recognition task that requires binding together learned object features into a single, coherent object, as well as a variant of this task that requires selecting from

and identifying multiple objects. Despite our focus on "hardwired" binding, we believe that the same mechanisms perform "on-demand" binding (e.g., conjunctive visual search). In on-demand binding, top-down influences bias processing toward items in a particular region of space, and consequently, competitive inhibition eliminates those features in nearby areas of space, allowing a properly bound representation of the novel item.

One natural consequence of our proposal is the suggestion that perception and behavior are largely driven by an interactive process that integrates bottom-up information with dynamic constraints including top-down, conceptual knowledge. It is somewhat surprising then, that a large class of extant theories of visual processing treat early perceptual processing as a feedforward set of stages that simply transform information from one level of the visual system to the next (Riesenhuber and Poggio, 1999b; Serre et al., 2007; Vanrullen, 2007). Models that instantiate this feedforward theory often include a "max" operation that selects the largest response at each processing level, which can be viewed as a form of inhibitory competition that suppresses less relevant responses (Riesenhuber and Poggio, 1999a). These models, however, lack top-down feedback to reinforce relevant information, which can emerge at any time over the course of processing.

Competitive dynamics reflecting inhibitory and top-down influences within visual areas are clear if one examines population level responses. For example, initial IT population responses convey information about many individual object parts, but information about the object as a whole begins to emerge over the full time course of their response (Brincat and Connor, 2006; see also Sugase-Miyamoto et al., 2011). Other single-cell analyses have indicated that the selectivity of IT neurons changes over time, beginning with a quick burst of broadly tuned activity that gradually becomes more selective (Tamura and Tanaka, 2001). Similar temporal dynamics have been demonstrated at other levels of the visual system, such as areas V2 and V4 (Hegde and van Essen, 2004, 2006). The fact that the information content of neural responses changes over time strongly suggests that some aspects of the representation are being selected over others. Our account of binding suggests that relevancy is a significant determining factor of what parts of the representation are ultimately selected for the bound representation at he highest levels.

Our proposal is highly congruent with many previous descriptions of binding (Reynolds and Desimone, 1999; Shadlen and Movshon, 1999; Treisman, 1999; Bundesen et al., 2005). Our contribution is novel in implementing a biologically grounded neural network model that embodies these theories, and in further specifying the mechanisms involved, and how they interact. One notable relation is to the role of top-down feedback in the form of spatial attention in Treisman's Feature Integration Theory (Treisman, 1996, 1999). Top-down feedback in our model does not directly perform binding, however, but simply prevents misbinding by highlighting some features over others and relying on competitive inhibition to suppress the others.

Our proposal also has much in common with (Reynolds and Desimone, 1999) biased competition model, which cites the importance of competitive inhibition between populations of neurons and top-down biasing of relevant features. However, the biased competition model has traditionally focused on frontal and parietal cortices as likely sources of the biasing signal. While attentional signals from these areas are clearly capable of biasing perceptual processing (Miller and Cohen, 2001; Thompson et al., 2005; Herd et al., 2006; Bressler et al., 2008), our approach provides a more general characterization of biasing. Specifically, any area that sends feedback to an earlier area has the potential to bias its computations. In our simulations, this allows for representations that are beginning to emerge at high-level areas to bias lower-level areas, which itself can be viewed as a form of emergent feature-based attention.

Theories centering on the role of synchrony have also been proposed as a solution to the binding problem (Singer, 1993, 1999; Singer and Gray, 1995; Uhlhaas et al., 2009). There is ample evidence that neural firing does synchronize to some degree, and that synchrony plays a role in attentive object recognition (Gray et al., 1989; Buzsaki and Draguhn, 2004). We agree that synchrony does play a role in the competitive selection process that is the core of our proposal, acting as another form of contrast enhancement by providing mutual excitation among concurrently active neurons via recurrent feedback and lateral connections (Roland, 2010). Synchrony thus effectively gives the winners of competition an extra advantage in controlling responses at higher levels.

This role of synchrony in sharpening neural competition should be differentiated from early proposals that synchrony can simultaneously bind multiple objects. No data of which we are aware indicates that the brain performs "multiplexed synchrony," in which neurons representing each object remain in phase with others representing the same object, but reliably out of phase with neurons representing other objects. Theories of multiplexed synchrony for binding have been strongly criticized on the grounds of being both biologically implausible and unnecessary (Shadlen and Movshon, 1999; O'Reilly et al., 2003). While it seems intuitive that we are aware of many objects simultaneously, recent research on change blindness indicates that we do not maintain detailed representations outside the current focus of attention (Beck et al., 2001; Lamme, 2003; Simons and Rensink, 2005).

Because of the level of noise from incidental processing in the brain (compared to models, which are idealized and thus use little to no noise) multiplexed synchrony seems likely to be unstable beyond extremely short time periods. This drawback severely limits the use of this mechanism for binding in working memory, the other case in which intuition and some evidence suggests we maintain several representations simultaneously (Raffone and Wolters, 2001). One alternative to true multiplexed synchrony is that binding in working memory is performed by maintaining separate neural substrates for separate items within prefrontal cortex, as in the model of working memory developed by our group, reviewed in O'Reilly et al. (2010).

Rather than supposing that the brain can represent and interpret several arbitrary conjunctions of features simultaneously, it seems more parsimonious to assume, as in our proposal, that all features represented simultaneously are bound together. Instead of using a particular firing phase to "tag" each neuron as belonging to one object or another, the brain simply represents only one object (or concept, etc.) at a time when binding is difficult, thus serializing a computation that would pose unique difficulties for parallel processing.

While previous work on binding has presented many possible mechanisms and argued that they are needed to solve the brain's binding problem, the necessity of mechanisms beyond the most basic neural mechanisms has not been clearly demonstrated. We have presented a solution to the binding problem of that relies on only generic neural mechanisms to bind together features into objects. While our proposal clearly demonstrates that the mechanisms of inhibition and top-down feedback contribute in part to solving the brain's overall binding problem, it is possible that there exist binding-related situations that warrant additional mechanisms and processes (e.g., working memory). Only after attempting to explain these phenomena with basic neural mechanisms (as in the proposals mentioned above) should more complicated theories be considered.

METHODS

The LVis (Leabra Vision) model and its training/testing methods are briefly described here. See O'Reilly et al. (under review), for a detailed description. The model consists of a hierarchy of feature processing layers that roughly correspond to areas within the ventral stream of the brain – primary visual cortex (V1), extrastriate cortex (V2/V4), inferotemporal cortex (IT) – as well as higher-level layers that represent amodal semantic properties and named output responses (**Figure 2A**). The model processes grayscale bitmap images with filters that capture the response properties of the retina and lateral geniculate nucleus (LGN) of the thalamus, the results of which are used as inputs to the V1 layer. The model's V1 layer consists of a retinotopic grid of 3600 units that represent V1-like features at multiple spatial scales. The V2/V4 layer contains 2880 units that receive from neighborhoods of 320 V1 units. Neighboring V2/V4 units receive from overlapping portions of the V1 layer. The IT layer contains 200 units that receive from the entire 2880 V2/V4 units, and thus do not contain a retinotopic organization.

Overall, the model can be viewed as an expansion on a large class of hierarchical feedforward models of visual processing in the brain (Riesenhuber and Poggio, 1999b; Delorme and Thorpe, 2001; Masquelier and Thorpe, 2007; Serre et al., 2007). The primary innovation of the model is that hierarchically adjacent layers (e.g., V1 and V2/V4; V2/V4 and IT) are recurrently connected, providing an account of top-down feedback connections within the brain's ventral stream. Feedforward connections generally contribute 80–90% of the total input to a receiving layer and feedback connections contribute the remaining 10–20% of the total input. Overall layer activations are controlled using a *k*-winners-take-all (*k*WTA) inhibitory competition rule (O'Reilly, 1996; O'Reilly and Munakata, 2000) that ensures only the *k* most active units remain active over time. The specific *k* value varies for each layer in the model, but is generally in the range of 10–20% of the number of units in the layer.

SINGLE OBJECT SIMULATIONS

The model was trained to categorize images from the CU3D-100 dataset (http://cu3d.colorado.edu) using an extension of the Leabra learning algorithm (O'Reilly, 1996; O'Reilly and Munakata, 2000). The entire dataset consisted of 18,840 total images. Training proceeded for 1000 epochs of 500 trials, each of which consisted of a random image selected from the dataset which was transformed with small variations in position, scale, and planar rotation. Images of two exemplars from each category (4000 images total) were reserved for a generalization test. After training, the final mean generalization accuracy was 91.9%.

Category selective representations were obtained for each of the 100 categories by averaging the response patterns of the model's IT units to all training and testing images from each category. In general, a distribution of 10–20% of the 200 units were selective to a given category, reflecting the level of *k*WTA inhibition within the IT layer. The category-relevant units for a given category were then isolated using a simple threshold over the category selective representations. For the fish category used in the simulations here, a value of 0.3 was used such that a higher response level indicated a category-relevant unit while a lower response level indicated a category irrelevant unit. Small variations in this parameter produced very similar results.

To create the plots in **Figure 1**, the firing rate from each of the model's IT units was recorded and averaged across every training and testing fish image (180 total images), then grouped according to whether the unit was category-relevant or irrelevant. This procedure was repeated with a visual occlusion manipulation that used a Gaussian-based filter to delete pixels from the input image. The filter was defined as 1.0 within a circle of radius 5% of the image size and then fell off outside the circle as a Gaussian function. The σ parameter of the Gaussian was set to 5% of the image size. The filter was applied to the image a variable number of times, with more applications corresponding to higher levels of occlusion.

To create the plot in **Figure 2B**, the model was presented with an unoccluded image of the fish and the response pattern was recorded from the model's V2/V4, IT, Semantic, and Naming Output layers for 50 processing cycles. The model was subsequently presented with a 50% occluded image of the fish, and the resulting response patterns were used to compute the similarity to the unoccluded response patterns for each layer as a function of time. The cosine angle between the unoccluded and occluded response vectors was used as the similarity metric in this calculation.

MULTIPLE OBJECT SIMULATIONS

The multiple object simulations involved training the model to recognize smaller versions of the CU3D-100 stimuli and testing its ability to generalize to presentations of multiple small stimuli. Training methods for these simulations were generally similar to the single object simulations described above, but a subset of the dataset was used. Five (5) exemplars from 5 categories (500 total images) were selected from the full dataset (*bicycle, car, donut, doorhandle*, and *gun*). Each image was downscaled to 50% of its size (originally 320 × 320 pixels, downscaled to 160 × 160 pixels) and randomly placed on either the left or right half of a new 320 × 320 image with variation in the y axis position. This was repeated 25 times for each of the 500 original images, resulting in 12500 new images. The model was trained on images from this dataset of 4 exemplars from each category to ensure proper generalization without over fitting. Training proceeded for 50 epochs of 500 trials. This was repeated for five instances of the model using different combinations of the 4 training exemplars

from each category and randomized initial weights. After training, the final accuracy over the training stimuli was 100% for each model.

To create the multiple object stimuli that were used for testing, images from each possible pairing of categories were randomly combined with one 160 × 160 image on the left half of a new 320 × 320 image and one 160 × 160 image on the right half. This was repeated 25 times for each category pairing, resulting in 250 new images containing two objects. In testing over these images, the model was ran for 100 cycles, as it often did not fully converge in the standard 50 cycles used in single object presentations. A testing trial was counted as correct if the model's output matched either of the two categories in the image.

We tested the effect of removing top-down feedback and inhibitory competition from the model on recognition accuracy for the multiple object stimuli. To remove influence from top-down feedback only, unit inputs from top-down feedback connections (e.g., Naming Output to IT, IT to V2/V4) were simply multiplied by zero during the testing phase. Removing influence from both top-down feedback and inhibitory competition required training a variant of the model that contained only feedforward connections (allowing for negative weights between units) with a backpropagation algorithm. This feedforward model required training for 100 epochs of 500 trials on the training stimuli before reaching 100% accuracy. Aside from these differences, the model was architecturally equivalent to the LVis model in terms of layer organization and numbers of units and used otherwise identical training and testing methods.

The same method that was used in the single object simulations was used to isolate category selective representations for each of the 5 categories, except that IT units were further grouped into those shared across category pairings (e.g., gun *and* bicycle units) as well as those that were unique to each category. These groupings were used to create the plots in **Figure 3**. Similarity to category selective representations was also measured to determine how much the overall pattern of responses across the IT layer approximated the category selective response to the single objects. The cosine angle between the category selective representation and the IT response vector was used as the similarity metric in this calculation.

To investigate how category-relevant responses were related to a model's output (**Figure 4**), the firing rates of units that corresponded to the model's output were isolated into one grouping (*relevant* responses) while the firing rates of all other units were isolated into another grouping (*irrelevant* responses). Our decision to refer to these responses as "relevant" and "irrelevant" was made to keep with the theme of relevant and irrelevant responses when a single object was presented in isolation, but one should note that irrelevant responses encompassed what could be considered to be relevant responses. For example, if a presented stimulus contained a gun and a bicycle and a model responded *gun*, the responses from units that corresponded to the *gun* category were considered to be the relevant responses while the units that corresponded to all other categories (including *bicycle*) were considered to be irrelevant. Other reasonable labels for these two groupings might be selected/unselected or attended/unattended responses.

REFERENCES

Bar, M., Kassam, K., Ghuman, A., Boshyan, J., and Schmidt, A. (2006). Top-down facilitation of visual recognition. *Proc. Natl. Acad. Sci. U.S.A.* 103, 449–454.

Beck, D. M., Rees, G., Frith, C. D., and Lavie, N. (2001). Neural correlates of change detection and change blindness. *Nat. Neurosci.* 4, 645–650.

Boehler, C. N., Schoenfeld, M. A., Heinze, H. J., and Hopf, J. M. (2008). Rapid recurrent processing gates awareness in primary visual cortex. *Proc. Natl. Acad. Sci. U.S.A.* 105, 8742–8747.

Bressler, S. L., Tang, W., Sylvester, C. M., Shulman, G. L., and Corbetta, M. (2008). Top-down control of human visual cortex by frontal and parietal cortex in anticipatory visual spatial attention. *J. Neurosci.* 28, 10056–10061.

Brincat, S. L., and Connor, C. E. (2006). Dynamic shape synthesis in posterior inferotemporal cortex. *Neuron* 49, 17–24.

Bundesen, C., Habekost, T., and Kyllingsbaek, S. (2005). A neural theory of visual attention: bridging cognition and neurophysiology. *Psychol. Rev.* 112, 291–328.

Buzsaki, G., and Draguhn, A. (2004). Neuroscience neuronal oscillations in cortical networks. *Science* 304, 1926–1938.

Carandini, M., and Heeger, D. (2012). Normalization as a canonical neural computation. *Nat. Rev. Neurosci.* 13, 51–62.

Delorme, A., and Thorpe, S. (2001). Face identification using one spike per neuron: resistance to image degradations. *Neural. Netw.* 14, 795–803.

Desimone, R., and Duncan, J. (1995). Neural mechanisms of selective visual attention. *Annu. Rev. Neurosci.* 18, 193–222.

Fahrenfort, J. J., Scholte, H. S., and Lamme, V. A. F. (2007). Masking disrupts reentrant processing in human visual cortex. *J. Cogn. Neurosci.* 19, 1488–1497.

Felleman, D. J., and van Essen, D. C. (1991). Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex* 1, 1–47.

Gray, C. M., König, P., Engel, A. K., and Singer, W. (1989). Oscillatory responses in cat visual cortex exhibit inter-columnar synchronization which reflects global stimulus properties. *Nature* 338, 334–337.

Hegde, J., and van Essen, D. C. (2004). Temporal dynamics of shape analysis in macaque visual area v2. *J. Neurophysiol.* 92, 3030–3042.

Hegde, J., and van Essen, D. C. (2006). Temporal dynamics of 2d and 3d shape representation in macaque visual area v4. *Vis. Neurosci.* 23, 749–763.

Herd, S. A., Banich, M. T., and O'Reilly, R. C. (2006). Neural mechanisms of cognitive control: an integrative model of Stroop task performance and fMRI data. *J. Cogn. Neurosci.* 18, 22–32.

Johnson, J. S., and Olshausen, B. A. (2005). The recognition of partially visible natural objects in the presence and absence of their occluders. *Vision Res.* 45, 3262–3276.

Juan, C., Tiangang, Z., Hua, Y., and Fang, F. (2010). Cortical dynamics underlying face completion in human visual system. *J. Neurosci.* 30, 16692–16698.

Kandel, E. R., Schwartz, J. H., and Jessell, T. M. (1995). *Essentials of Neural Science and Behavior*. Norwalk, CT: Appleton & Lange.

Kobatake, E., and Tanaka, K. (1994). Neuronal selectivities to complex object features in the ventral visual pathway. *J. Neurophysiol.* 71, 856–867.

Koivisto, M., Railo, H., Revonsuo, A., Vanni, S., and Salminen-Vaparanta, N. (2011). Recurrent processing in v1/v2 contributes to categorization of natural scenes. *J. Neurosci.* 31, 2488–2492.

Kourtzi, Z., and Connor, C. E. (2010). Neural representations for object perception: structure, category, and adaptive coding. *Annu. Rev. Neurosci.* 34, 45–67.

Kourtzi, Z., and Kanwisher, N. (2001). Representation of perceived object shape by the human lateral occipital complex. *Science* 293, 1506–1509.

Kovacs, G., Vogels, R., and Orban, G. A. (1995). Selectivity of macaque inferior temporal neurons for partially occluded shapes. *J. Neurosci.* 15, 1984–1997.

Lamme, V. A. (2003). Why visual attention and awareness are different. *Trends Cogn. Sci. (Regul. Ed.)* 7, 12–18.

Lerner, Y., Harel, M., and Malach, R. (2004). Rapid completion effects in human high-order visual areas. *Neuroimage* 21, 516–526.

Lerner, Y., Harel, M., and Malach, R. (2004). Rapid completion effects in human high-order visual areas. *Neuroimage* 21, 516–526.

Logothetis, N. K., Pauls, J., and Poggio, T. (1995). Shape representation in the inferior temporal cortex of monkeys. *Curr. Biol.* 5, 552–563.

Masquelier, T., and Thorpe, S. J. (2007). Unsupervised learning of visual features through spike timing dependent plasticity. *PLoS Comput. Biol.* 3, e31. doi:10.1371/journal.pcbi.0030031

Maunsell, J. H. R., and Treue, S. (2006). Feature-based attention in visual cortex. *Trends Neurosci.* 29, 317–322.

Miller, E. K., and Cohen, J. D. (2001). An integrative theory of prefrontal cortex function. *Annu. Rev. Neurosci.* 24, 167–202.

Mountcastle, V. B. (1997). The columnar organization of the neocortex. *Brain* 120(Pt 4), 701–722.

Nielsen, K. J., Logothetis, N. K., and Rainer, G. (2006). Dissociation between local field potentials and spiking activity in macaque inferior temporal cortex reveals diagnosticity-based encoding of complex objects. *J. Neurosci.* 26, 9639–9645.

O'Reilly, R. C. (1996). Biologically plausible error-driven learning using local activation differences: the generalized recirculation algorithm. *Neural Comput.* 8, 895–938.

O'Reilly, R. C., Busby, R. S., and Soto, R. (2003). "Three forms of binding and their neural substrates: alternatives to temporal synchrony," in *The Unity of Consciousness: Binding, Integration, and Dissociation*, ed. A. Cleeremans (Oxford: Oxford University Press), 168–192.

O'Reilly, R. C., Herd, S. A., and Pauli, W. M. (2010). Computational models of cognitive control. *Curr. Opin. Neurobiol.* 20, 257–261.

O'Reilly, R. C., and Munakata, Y. (2000). *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain.* Cambridge, MA: The MIT Press.

Raffone, A., and Wolters, G. (2001). A cortical mechanism for binding in visual working memory. *J. Cogn. Neurosci.* 13, 766–785.

Reynolds, J. H., and Chelazzi, L. (2004). Attentional modulation of visual processing. *Annu. Rev. Neurosci.* 27, 611–647.

Reynolds, J. H., and Desimone, R. (1999). The role of neural mechanisms of attention in solving the binding problem. *Neuron* 24, 111–125.

Riesenhuber, M., and Poggio, T. (1999a). Are cortical models really bound by the "binding problem?" *Neuron* 24, 87–93.

Riesenhuber, M., and Poggio, T. (1999b). Hierarchical models of object recognition in cortex. *Nat. Neurosci.* 3, 1199–1204.

Riesenhuber, M., and Poggio, T. (2002). Neural mechanisms of object recognition. *Curr. Opin. Neurobiol.* 12, 162–168.

Roland, P. (2010). Six principles of visual cortical dynamics. *Front. Syst. Neurosci.* 4:28. doi:10.3389/fnsys.2010.00028

Rolls, E. T., and Stringer, S. M. (2006). Invariant visual object recognition: a model, with lighting invariance. *J. Physiol. Paris* 100, 43–62.

Rust, N. C., and Dicarlo, J. J. (2010). Selectivity and tolerance ("invariance") both increase as visual information propagates from cortical area v4 to it. *J. Neurosci.* 30, 12978–12995.

Scannell, J., Blakemore, C., and Young, M. P. (1995). Analysis of connectivity in the cat cerebral cortex. *J. Neurosci.* 15, 1463–1483.

Serre, T., Kreiman, G., Kouh, M., Cadieu, C., Knoblich, U., and Poggio, T. (2007). A quantitative theory of immediate visual recognition. *Prog. Brain Res.* 165, 33–56.

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

Simons, D. J., and Rensink, R. A. (2005). Change blindness: past, present, and future. *Trends Cogn. Sci. (Regul. Ed.)* 9, 16–20.

Singer, W. (1993). Synchronization of cortical activity and its putative role in information processing and learning. *Annu. Rev. Physiol.* 55, 349–374.

Singer, W. (1999). Neuronal synchrony: a versatile code for the definition of relations? *Neuron* 24, 49–65.

Singer, W., and Gray, C. M. (1995). Visual feature integration and the temporal correlation hypothesis. *Annu. Rev. Neurosci.* 18, 555–586.

Sporns, O., Honey, C. J., and Kotter, R. (2007). Identification and classification of hubs in brain networks. *PLoS ONE* 2, e1049. doi:10.1371/journal.pone.0001049

Sporns, O., and Zwi, J. D. (2004). The small world of the cerebral cortex. *Neuroinformatics* 2, 145–162.

Sugase-Miyamoto, Y., Matsumoto, N., and Kawano, K. (2011). Role of temporal processing stages by inferior temporal neurons in facial recognition. *Front. Psychol.* 2:141. doi:10.3389/fpsyg.2011.00141

Swadlow, H., and Gusev, A. (2002). Receptive-field construction in cortical inhibitory interneurons. *Nat. Neurosci.* 5, 403–404.

Tamura, H., and Tanaka, K. (2001). Visual response properties of cells in the ventral and dorsal parts of the macaque inferotemporal cortex. *Cereb. Cortex* 11, 384–399.

Tanaka, K. (1996). Inferotemporal cortex and object vision. *Annu. Rev. Neurosci.* 19, 109–139.

Thompson, K. G., Biscoe, K. L., and Sato, T. R. (2005). Neuronal basis of covert spatial attention in the frontal eye field. *J. Neurosci.* 25, 9479–9487.

Tompa, T., and Sary, G. (2010). A review on the inferior temporal cortex of the macaque. *Brain Res. Rev.* 62, 165–182.

Treisman, A. (1996). The binding problem. *Curr. Opin. Neurobiol.* 6, 171–178.

Treisman, A. (1999). Solutions to the binding problem: progress through controversy and convergence. *Neuron* 24, 105–125.

Uhlhaas, P. J., Pipa, G., Lima, B., Melloni, L., Neuenschwander, S., Nikolic, D., and Singer, W. (2009). Neural synchrony in cortical networks: history concept and current status. *Front. Integr. Neurosci.* 3:17. doi:10.3389/neuro.07.017.2009

Ungerleider, L. G., and Bell, A. H. (2011). Uncovering the visual "alphabet": advances in our understanding of object perception. *Vision Res.* 51, 782–799.

Vanrullen, R. (2007). The power of the feed-forward sweep. *Adv. Cogn. Psychol.* 3, 167–176.

Vanrullen, R. (2009). Binding hardwired vs. on-demand feature conjunctions. *Vis. Cogn.* 17, 103–119.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 16 January 2012; accepted: 20 May 2012; published online: 18 June 2012.

Citation: Wyatte D, Herd S, Mingus B and O'Reilly R (2012) The role of competitive inhibition and top-down feedback in binding during object recognition. Front. Psychology ***3****:182. doi: 10.3389/fpsyg.2012.00182*

This article was submitted to Frontiers in Cognitive Science, a specialty of Frontiers in Psychology.

Dopamine and self-directed learning

Seth HERD [a,1], Brian MINGUS [a] and Randall O'REILLY [a]
[a] *Department of Psychology and Neuroscience,*
University of Colorado,
345 UCB,
Boulder, Colorado 80309

Abstract.
Humans are intrinsically motivated to learn. Such motivation is necessary to be a human-like learner, and helpful for any learning system designed to achieve general intelligence. We discuss the limited existing computational work in this area, and link them to known and theorized properties of the dopamine system. The relatively well-understood mechanisms by which dopamine release signals unpredicted reward can also serve to signal new learning. Dopamine release leads to maintenance of current representations, which serves to "lock" attention onto topics or tasks in which useful learning is occurring. We thus propose a novel but natural extension of known aspects of dopamine function to perform self-directed learning of arbitrary self-defined tasks. If this hypothesis is correct, detailed experimental evidence on dopamine function can help guide computational research into human-like learning systems.

Keywords. Motivation, Learning, Neural Network

Introduction

Children play not to learn but because play is fun. They play with things and in ways they find interesting, and cease once they become boring. Evolution, on the other hand, has designed children not to have fun, but to learn. Their pattern of play reflects an evolved tendency toward maximizing learning opportunities. Understanding our ability to efficiently self-direct learning is likely to be crucial for understanding and reproducing human intelligence.

Building a biologically inspired cognitive architecture (BICA) is intended to capitalize on the only example of a generally intelligent system we have available: the human brain. Similarly, there is only one example of a training set that produces general intelligence: that selected by the human learner from its natural environment. Designing a BICA as a human-like learner serves not only to make that agent more accessible to humans, but follows the only known working path to general intelligence.

While the dopamine system has been previously hypothesized to play a role in self-directed learning, we propose a more specific and sophisticated relation

[1]Corresponding Author: E-mail: seth.herd@colorado.edu

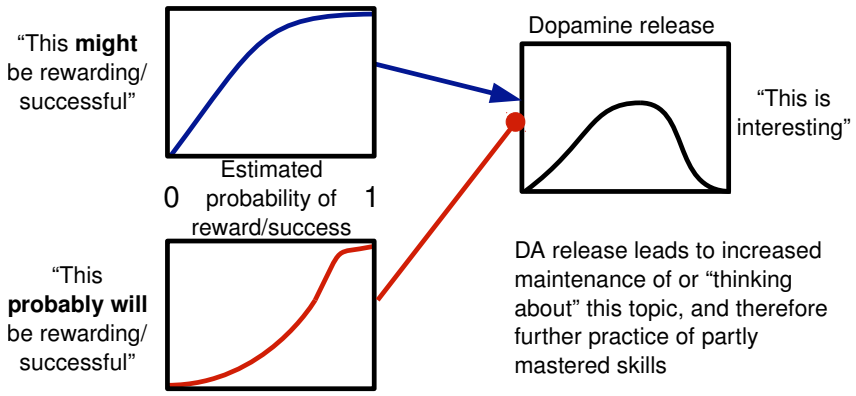

Figure 1. Self directed learning as an outcome of dopamine system function. The dopamine system's known property of signaling only unpredicted rewards could allow it to signal activities which are neither too easy nor too difficult for the learner's current abilities. An excitatory system learns "optimistically" to respond when reward is possible, while an inhibitory system learns more slowly or "pessimistically" when reward is likely. Dopamine release is approximately the difference in activation of the two systems. The phasic release of dopamine then serves to "lock" attention to the current topic, encouraging focused practice on tasks where success is possible but not certain.

between DA release and attentive learning. We work from the hypothesis that success at any physical or mental task acts as a reward, and show how known dopamine system mechanisms should then produce self-directed learning behavior much like, but importantly different from, existing computational approaches (Figure 1). We show how not only the antecedents but the consequences of DA release suit the dopeamine system for a crucial role in self-directed learning.

1. Dopamine function and self-directed learning

The best developed current theory of self-directed learning (or "intrinsic motivation") proposes a heuristic of increasing predictability [1]. This principle, dubbed Intelligent Adaptive Curiosity (IAC), directs the learner toward activities in which it is currently on a steep part of a learning curve. In essence, the system predicts the outcomes of its actions, and keeps a record of the quality of those predictions. Domains in which predictions have recently improved are probably those in which significant learning has occurred. The system therefore chooses action domains stochastically but based on that metric.

Success at arbitrary laboratory tasks seems to act as a reward and trigger dopamine release in humans [2]. This could result from an innate reward signal from successful prediction, from success being strongly predictive of primary rewards (one intriguing hypothesis is that humans find positive regard from other humans intrinsically rewarding [3]), or for other reasons. Here we remain agnostic

on the particular underlying cause, and focus on the consequences of the general hypothesis.

If success at a self-defined task results in a reward signal, the dopamine system appears to have the right properties for producing something similar to the IAC algorithm described above. The relatively well understood mechanisms by which dopamine directs learning and memory toward reward and reward-predictive events may also serve to direct attention toward tasks and topics for which learning is currently happening rapidly. We first discuss how DA acts to direct learning toward unpredicted reward predictors, then show how that function can generalize to self-direct learning on arbitrary sensorimotor or cognitive tasks.

2. Dopamine release directs learning to unpredicted reward predictors

Expected rewards are "discounted" by the dopamine signal: after sufficient learning, a predictable reward causes little to no DA release. The Temporal Differences (TD) algorithm [4] has been widely used to model this effect. Physiologically, the absence of DA release for a well-predicted reward probably results from an active cancellation (evidence reviewed in [5,6]). An excitatory system pushes for DA release and learns quickly, while an inhibitory system learns more slowly (or conservatively - the role of uncertainty in guiding behavior [8] is an important one that we do not address here). The overall result is that, early in learning, a predictable reward causes phasic DA release; later in learning, that release is canceled by the slower-learning inhibitory system (as illustrated in Figure 1).

After sufficient learning, phasic DA release is instead triggered by the stimulus that predicts the upcoming reward. In classical experiments, a light or tone that reliably signals an upcoming food reward will trigger dopamine release after sufficient learning. Dopamine thus signals newly predicted rewards, while remaining silent for rewards that have been previously predicted.

Phasic DA release enhances learning. For instance, playing a tone just before inducing a phasic DA release dramatically increases the area of cortex in which neurons respond to that tone in the future [9]. The combination of signaling new reward predictions and directing learning allows the dopamine signal as currently understood to perform a relatively weak type of self-directed learning, as outlined below.

This attentional focus causes the organism to learn about events immediately preceding reward; e.g., an infant learns which motor commands scoop applesauce into its mouth, producing a primary reward of nourishing sugar. But this focus on the moment of reward becomes counterproductive. Once those motor commands are mastered, it becomes useful to learn about the conditions surrounding the predictor of reward (in this case, the jar of applesauce). As the success of those motor movements becomes predictable, the dopamine spike at the moment of reward is cancelled by the inhibitory component of the system. At the same time, any stimulus that reliably predicts reward (in the example, the applesauce jar) starts to produce phasic DA release. This moves the focus of learning toward the next step in the causal chain. The infant now learns to reach or ask for the applesauce jar.

While the TD algorithm has worked well to explain how similar chains of learning are performed in laboratory tasks, its reliance on temporal rather than semantically associative chaining limits its applicability to rich and varying environments. Brain mechanisms for a similar, but more general algorithm (one that chains semantically rather than only temporally) are reviewed in [6].

In sum, the DA signal is thought to provide a simple form of self-directed learning that enhances learning to important (reward-predicting) events. The DA system may also have been co-opted by evolution to produce a more flexible and powerful form of self directed learning, as discussed below.

3. How the DA system can signal opportunities for learning

Our main novel proposal is that the same set of mechanisms described above could act to direct attention toward activities for which average success has recently become greater – those which are being learned rapidly. In the better-researched case of laboratory tasks and rewards, attention is directed toward predictors of reward that are themselves poorly predicted (e.g., the unexpected appearance of an applesauce jar). Supposing that self-defined success at any task acts as a reward has interesting consequences.

We deliberately use a very general supposition: some close correlate of successful performance (e.g., accurate prediction) acts as a reward from the perspective of the DA system. For instance, when an infant successfully places one block atop another, its reward circuits fire. Treating an arbitrary happening or concept as a reward could be crucial for human cognition [7], allowing us to work toward abstract concepts like "money" and "trustworthiness" as well as concrete rewards like food and shelter. Successful performance of an arbitrary task is one such reward-substitute with particularly important consequences.

If success counts as a reward, the reward discounting properties of the DA system become useful in directing learning toward new successes. When successful skill execution is fully predictable, as in a task that's been mastered, the same circuits that cancel DA release for physical rewards, cancel that for successful prediction. Thus frequent phasic DA release indicates a task that's sometimes successful but not yet mastered. Thus the DA system directs learning toward new tasks, exactly as it directs learning toward new steps in a causal chain leading to physical reward.

4. How dopamine release can direct learning

There is a second important reason to favor dopamine release as a mechanism for self-directed learning. Not only does the DA system have the right properties to signal a useful learning opportunity, but DA will direct attention and therefore learning to whatever is happening when it's released. Dopamine has a relatively well understood role in working memory (reviewed in [10]). Working memory function is also thought to be central to cognitive control [11,12].

In essence, sustained firing of neurons in prefrontal cortex and elsewhere is thought to be the basis of working memory. In the terms of the biased competi-

tion model [13], working memory representations are strategically maintained biases that direct attention toward appropriate items. These maintained representations can also act to direct attention toward topics, tasks, or stimulus-response mappings by acting as one constraint in a brain-wide constraint satisfaction [14].

Phasic dopamine release and tonic dopamine levels (resulting from frequent recent phasic releases) both play a role in working memory maintenance. At normal levels, increased tonic dopamine levels in cortex increase the maintenance of information [15]. Phasic dopamine release also biases the striatum toward "Go" over "NoGo" decisions, one likely consequence of which is the maintenance of information currently represented in associated regions of prefrontal cortex [16].

Both of these factors make dopamine release tend to "lock" the current topic of thought in mind. If a hungry animal sees food that's not immediately obtainable, it will tend to keep maintaining that representation, and so guide its behavior toward obtaining it. Similarly, if a child performs above its predictions at a particular block-stacking task, it will tend to keep thinking about it and so performing similar tasks until they become too predictably successful (or unsuccessful).

5. Summary and Conclusions

The logic above describes how humans may have adapted the evolutionarily old dopamine system to enhance general learning. The system evolved to direct learning toward progressive events in a causal chain leading to reward. The same basic mechanisms can direct learning toward learning itself with the simple adaptation of making successful prediction (or any other correlate of successful learning) rewarding in itself. The system is also ideally suited for the task of self-directed learning because dopamine release in itself serves to "lock" attention on the item, task, or concept currently being attended to or represented. Because both the causes and effects of dopamine release are so well suited to usefully directing learning toward optimal areas, it is likely that the dopamine system plays a crucial role in this important human adaptation.

This hypothesis is compatible with the approach of Huang and Weng [17]. The dopamine system is not only well known to signal reward, but seems to also signal punishment and novelty, the three components they suggest are the minimum for an effective self-directed learning system.

Other aspects of human self directed learning deserve attention as well. For instance, there are likely biases in the self-direction of learning that prevent people from searching behavior-space at random. Infants may be biased toward making motor actions and sounds ("babbling") that may pre-train systems for the more deliberate tasks discussed here. Imitation likely provides an important constraint guiding learners to useful parts of behavior-space.

Further understanding the function of dopamine and other neuromodulatory systems (e.g., norepinephrin [8]) will help us understand how the human brain usefully directs its own learning. This will in turn allow us to design more human-like and more effective learners. Humans' poorly understood ability to select our own tasks and learning examples may well be a crucial ingredient in our still-unique ability to arrive at a rich understanding of our world.

References

[1] P. Oudeyer, F. Kaplan, V. Hafner, Intrinsic motivation systems for autonomous mental development, Evolutionary Computation, IEEE Transactions on evolutionary computation 11 (2007) 265–286.

[2] A. R. Aron, D. Shohamy, J. Clark, C. Myers, M. A. Gluck, R. A. Poldrack, Human midbrain sensitivity to cognitive feedback and uncertainty during classification learning., Journal of neurophysiology 92 (2) (2004) 1144–1152.
URL http://www.ncbi.nlm.nih.gov/pubmed/15014103

[3] M. Tomasello, The Cultural Origins of Human Cognition, Harvard University Press, Cambridge, MA, 2001.

[4] R. S. Sutton, Learning to predict by the method of temporal differences, Machine Learning 3 (1988) 9–44.

[5] R. C. O'Reilly, M. J. Frank, T. E. Hazy, B. Watz, Pvlv: The primary value and learned value pavlovian learning algorithm., Behavioral Neuroscience 121 (2007) 31–49.
URL http://www.ncbi.nlm.nih.gov/pubmed/17324049

[6] T. E. Hazy, M. J. Frank, R. C. O'Reilly, Neural mechanisms of acquired phasic dopamine responses in learning., Neuroscience and biobehavioral reviews 34 (5) (2010) 701–720.
URL http://www.ncbi.nlm.nih.gov/pubmed/19944716

[7] R. Montague, *Why Choose This Book?*, Dutton, New York, New York, 2006.

[8] G. Aston-Jones, J. D. Cohen, An integrative theory of locus coeruleus-norepinephrine function: adaptive gain and optimal performance., Annual review of neuroscience 28 (2005) 403–450.
URL http://www.ncbi.nlm.nih.gov/pubmed/16022602

[9] S. Bao, V. T. Chan, M. M. Merzenich, Cortical remodelling induced by activity of ventral tegmental dopamine neurons., Nature 412 (2001) 79–82.
URL http://www.ncbi.nlm.nih.gov/pubmed/11452310

[10] J. K. Seamans, C. R. Yang, The principal features and mechanisms of dopamine modulation in the prefrontal cortex., Progress in neurobiology 74 (2004) 1–57.
URL http://www.ncbi.nlm.nih.gov/pubmed/15381316

[11] J. B. Morton, Y. Munakata, Active versus latent representations: A neural network model of perseveration and dissociation in early childhood, Developmental Psychobiology 40 (2002) 255–265.

[12] G. Deco, E. T. Rolls, Attention, short-term memory, and action selection: a unifying theory., Progress in neurobiology 76 (4).
URL http://www.ncbi.nlm.nih.gov/pubmed/16257103

[13] R. Desimone, J. Duncan, Neural mechanisms of selective visual attention., Annual Review of Neuroscience 18 (1995) 193.

[14] S. A. Herd, M. T. Banich, R. C. O'Reilly, Neural mechanisms of cognitive control: an integrative model of stroop task performance and fmri data., Journal of cognitive neuroscience 18 (2006) 22–32.
URL http://www.ncbi.nlm.nih.gov/pubmed/16417680

[15] J. K. Seamans, T. W. Robbins, Dopamine modulation of the prefrontal cortex and cognitive function, 2010, pp. 373–398.

[16] T. E. Hazy, M. J. Frank, R. C. O'Reilly, Banishing the homunculus: Making working memory work., Neuroscience 139 (2006) 105–118.
URL http://www.ncbi.nlm.nih.gov/pubmed/16343792

[17] X. Huang, J. Weng, Inherent Value Systems for Autonomous Mental Development, International Journal of Humanoid Robotics 4 (2007) 407–433.

Generalization of Figure-Ground Segmentation from Binocular to Monocular Vision in an Embodied Biological Brain Model*

Brian Mingus, Trent Kriete, Seth Herd, Dean Wyatte,
Kenneth Latimer, and Randy O'Reilly

Computational Cognitive Neuroscience Lab
Department of Psychology
University of Colorado at Boulder
Muenzinger D244, 345 UCB
Boulder, Co, 80309, USA
{brian.mingus,trent.kriete,seth.herd,dean.wyatte,
kenneth.latimer,randy.oreilly}@colorado.edu
http://grey.colorado.edu/ccnlab

Abstract. Monocular figure-ground segmentation is an important problem in the field of Artificial General Intelligence. A solution to this problem will unlock vast sets of training data, such as Google Images, in which salient objects of interest are situated against complex backgrounds. In order to gain traction on the figure-ground problem we enhanced the Leabra Vision (LVis) model, which is our state-of-the-art model of 3D invariant object recognition [8], such that it can continue to recognize objects against cluttered backgrounds that, while simple, are complex enough to substantially hurt object recognition performance. The principle of operation of the network is that it learns to use a low resolution view of the scene in which high spatial frequency information such as the background falls out of focus in order to predict which aspects of the high resolution scene are the figure. This filtered view then serves to enhance the figure in the input stages of LVis and substantially improves object recognition performance against cluttered backgrounds.

Keywords: figure-ground, neural network, object recognition.

1 Introduction

When we look at a photograph the objects jump out into three dimensional life. This is surprising since each eye conveys the same image of the photograph with

* Supported by the Intelligence Advanced Research Projects Activity (IARPA) via the U.S. Army Research Office contract number W911NF-10-C-0064. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, the U.S. Army Research Office, or the U.S. Government.

J. Schmidhuber, K.R. Thórisson, and M. Looks (Eds.): AGI 2011, LNAI 6830, pp. 351–356, 2011.

no useful disparity signals. One can demonstrate this to themselves by looking at a photograph with one eye closed and noting the rich perception of depth. So too is our depth perception intact when we perceive the world more generally with only one eye open. In normal binocular viewing conditions the disparity between objects in the two eyes helps us to compute their depth, but it is rather remarkable that we can continue to do this in lieu of this cue.

An idealized method of training a neural network to solve the monocluar figure-ground segmentation problem follows from its description. There are two input layers representing the V1 neurons for the left eye and right eye, respectively. These map onto a layer which computes focal disparity, that is, the zero-disparity region of foveation. During training the information from one eye is removed and the network is asked to predict the depth map of the scene. After making a guess based on monocular cues, the the other eye is returned and the weights are changed based on the difference between the predicted depth map and the actual depth map. While such a simple network only provides marginal figure-ground segmentation ability, it clearly demonstrates the point that we hope to make with Emer: that rich 3D signals can serve as a training signal for figure-ground segmentation with 2D cues.

2 Materials and Methods

Experiments were conducted using the emergent Neural Network Simulation System [1]. The Leabra neural network architecture and learning rule was used for all simulations [7].

2.1 CU3D-100 Dataset

To test the sufficiency of our model on a realistic, challenging version of the object recognition problem, we used our dataset of nearly 1,000 3D object models from the Google SketchUp warehouse (the *CU3D-100* dataset [5]) organized into 100 categories with an average of 9.42 exemplars per category (Fig. 2a-d). Two exemplars per category were reserved for testing, and the rest were used for training. Objects were rendered to 20 bitmap images per object with random $\pm 20°$ depth rotations (including a random 180° left-right flip for objects that are asymmetric along this dimension) and overhead lighting positioned uniformly randomly along an 80° overhead arc. These images were then presented to the model with planar (2D) transformations of 30% translation, 20% size scaling, and 14° in-plane rotations. The CU3D-100 dataset avoids the significant problems with other widely-used benchmarks such as the Caltech101 [9], by ensuring that recognition is truly robust to significant amounts of invariance, and the 3D rendering approach provides full parameterization over problem difficulty.

2.2 Structure of the Models

The LVis model [8] (Fig. 3) preprocessed bitmap images via two stages of mathematical filtering that capture the qualitative processing thought to occur in the

Fig. 1. Our virtual robot, Emer. His name is based on "emergent", our neural network simulator. Seen here are his torso, head, eyes, eye-beams, and the fish that he is foveating in prepration for object recognition. Emer is implemented using the Open Dynamics Engine rigid body physics simulator [6] and the Coin3D 3D Graphics Developer Kit [4]. Each of his eyes is a camera, and their offset positions on his head give him slightly different views of objects, facilitating stereo vision.

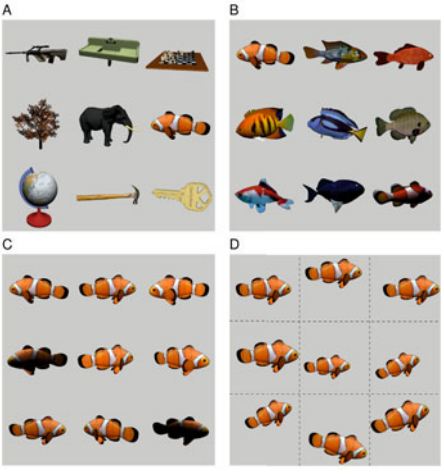

Fig. 2. The CU3D-100 dataset. **a)** 9 example objects from the 100 CU3D categories. **b)** Each category is further composed of multiple, diverse exemplars (average of 9.42 exemplars per category). **c)** Each exemplar is rendered with 3D (depth) rotations and variability in lighting. **d)** The 2D images are subject to 2D transformations (translation, scale, planar rotation), with ranges generally around 20%.

mammalian visual pathways from retina to LGN (lateral geniculate nucleus of the thalamus) to primary visual cortex (V1). The output of this filtering provided the input to the Leabra network, which then learned over a sequence of layers to categorize the inputs according to object categories.

The figure-ground model (Fig. 4) consists of - from the left column to the right column - V1, V1C end-stop cells [13] and figure layers. The figure layers correspond to the zero-disparity region of foveation. The network is connected in a feed-forward fashion from left to right, with high, medium and low spatial resolutions arranged from front to back. The figure layers are all bidirectionally connected, including recurrent connections. The goal of the network is to look at a figure against a background at all three resolutions and ultimately produce just the figure in the high resolution figure layer. This output representation is then used as input to the LVis object recognition model.

The middle column of layers in the figure-ground network correspond to end-stop cells which are useful for detecting T-junctions and contours in the image. These are good cues as to what separates figure from ground [13]. The role

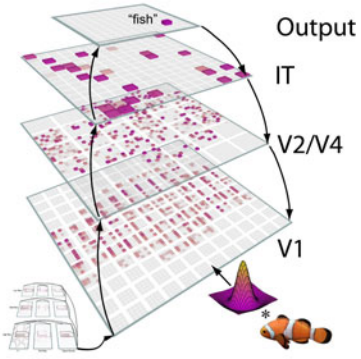

Fig. 3. The architecture of the LVis model [8]. LVis is based on the anatomy of the ventral pathway of the brain, from primary visual cortex (V1) through extrastriate areas (V2, V4) to inferotemporal (IT) cortex. V1 reflects filters that model the response properties of V1 neurons (both simple and complex subtypes). In higher levels, receptive fields become more spatially invariant and complex. All layers are bidirectionally connected, allowing higher-level information to influence bottom-up processing.

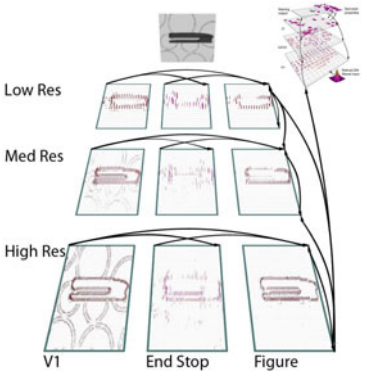

Fig. 4. The figure-ground segmentation model. There are three sets of layers at three interacting spatial resolutions. The first set corresponds to V1, the second set to V1C end-stop cells [13], and the third set learns to extract the figure from the background. The network is connected in a feedforward fashion from left to right and the figure-ground layers have both recurrent and bidirectional connectivity. The network learns to combine information from high and low-resolution V1 layers in order to predict the figure in the high-resolution figure layer.

of multiple interacting spatial resolutions follows clearly from the left-most V1 column in Fig. 4. At coarse spatial resolution the background falls out almost completely at the expense of losing much of the high-frequency spatial detail of the object. At high resolution the spatial detail of the object is preserved, but so too is the background. The principle of operation of the network is to learn to take advantage of these competing constraints.

3 Results and Discussion

All of the conditions in Fig. 5 have the same basic task, which is invariant object recognition on the CU3D-100 dataset. The model is trained on approximately eight exemplars per category and then generalization performance is tested on the remaining two objects from each category. Generalization performance is computed as the number of errors divided by 200.

The performance of the learned monocular figure-ground segmentation model is compared to several other conditions in Fig. 5. The key comparison conditions

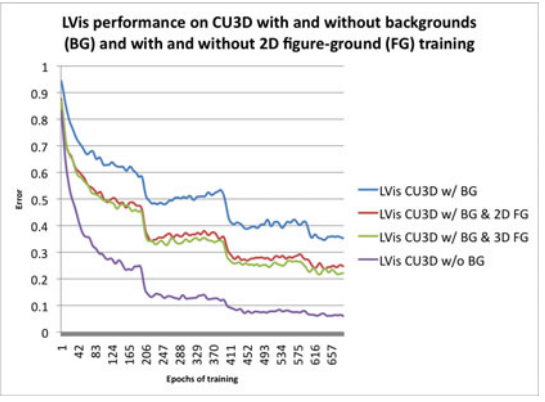

Fig. 5. Generalization performance of LVis in four object recognition conditions. **w/ BG:** With backgrounds and without figure-ground training error asymptotes at 35.3%. **w/ BG & 2D FG:** With backgrounds and with the learning figure-ground front-end intact performance asymptotes at 24.7% error. **w/ BG & 3D FG:** With backgrounds and using the target depth map as input into LVis (i.e., no 3D to 2D generalization - this is the best possible case for the previous condition) performance asymptotes at 22.2% error. **w/o BG:** Without backgrounds using just the standard LVis model performance asymptotes at 6% error.

are standard LVis with no backgrounds, LVis with backgrounds and without figure-ground segmentation and LVis with the best 3D figure-ground segmentation that our disparity matching system can compute.

The main condition being tested is object recognition against a background (such as the background seen in the picture in Fig. 4) with the learned monocular figure-ground segmentation model in place. To demonstrate that this is a hard problem, note that the difference in performance between LVis with and without backgrounds (and without figure-ground segmentation) is 29.2% error, which is a dramatic decrease in performance. The other key comparison is between the model that uses the computed disparity signal (and thus does not need to generalize from 3D to 2D) versus the learned monocular figure-ground segmentation model. The monocular model has only 2.4% more error, a relatively slight difference.

In conclusion, we chose to start with relatively simple backgrounds that nonetheless resulted in a dramatic detriment to performance in object recognition. The monocular figure-ground segmentation system had only 2.4% more error than it possibly could have, demonstrating that the model does indeed learn how to segment figure from ground. The results demonstrate the utility of using multiple interaction spatial resolutions, and are an important step on our way to using more realistic datasets such as Google Images.

References

1. Aisa, B., Mingus, B., O'Reilly, R.: The emergent neural modeling system. Neural Networks, 1045–1212 (2008)
2. Caltech101, http://www.vision.caltech.edu/Image_Datasets/Caltech101/
3. Computational Cognitive Neuroscience Lab, http://grey.colorado.edu/ccnlab
4. Coin3D 3D Graphics Engine Developer Kit, http://www.coin3d.org/
5. CU3D dataset, http://grey.colorado.edu/CompCogNeuro/index.php/CU3D
6. Open Dynamics Engine, http://www.ode.org/
7. O'Reilly, R.: The Leabra Model of Neural Interactions and Learning in the Neocortex. PhD Thesis (1996)
8. O'Reilly, R., Wyatte, D., Herd, S., Mingus, B., Jilk, D.: Bidirectional Biologically Plausible Object Recognition (2011) (in Press)
9. Pinto, N., Cox, D., DiCarlo, J.: Why is real-world object recognition hard? PLoS Computational Biology (2008)
10. Troscianko, T., Montagnon, R., Le Clerc, J., Malbert, E., Chanteau, P.: The role of colour as a monocular depth cue. Vision Research, 1923–1929 (1991)
11. Wheatstone, C.: Contributions to the physiology of vision.–Part the First. On some remarkable, and hitherto unobserved, phenomena of binocular vision. Philosophical Magazine Series 4 (1852)
12. Walk, D.: The Development of Depth Perception in Animals and Human Infants. Concept of Development: A Report of a Conference Commemorating the Fortieth Anniversary of the Institute of Child Development (1966)
13. Yazdanbakhsh, A., Livingstone, M.: End stopping in V1 is sensitive to contrast. Nature Neuroscience (2006)

The *emergent platform™* for community CCN

Randall C. O'Reilly (Randy.OReilly@colorado.edu)
Computational Cognitive Neuroscience Lab, Department of Psychology, University of Colorado at Boulder

Kai A. Krueger (Kai.Krueger@colorado.edu)
eCortex Inc, Boulder, Co; Latently, a Public Benefit Corporation, Boulder, Co

Brian Mingus (Brian.Mingus@colorado.edu)
Latently, a Public Benefit Corporation, Boulder, Co

John Rohrlich (John.Rohrlich@colorado.edu)
Computational Cognitive Neuroscience Lab, Department of Psychology, University of Colorado at Boulder

Abstract

The *emergent platform™* consists of the *emergent™* simulator (`http://grey.colorado.edu/emergent`) and supporting infrastructure for community CCN model development. It has integrations with open repositories such as wikis and source control as well as with free public supercomputing resources provided through the Neuroscience Gateway on the Comet supercomputer, resulting in a simple, yet full fledged platform for collaboration in brain modeling. The platform supports brain modeling up and down the stack, from computational neuroscience to mean field models to ACT-R and Bayesianism, allowing one to combine the strengths of the different approaches in a single model. *emergent™* has the best support for visualization of any neural simulator to date, enabling one to use visual regression to develop their intuitions of dynamic brain systems (Wlodzislaw & Dobosz, 2011). The platform has a long history in Connectionism, having descended from PDP (1986) and PDP++ (1995), and comes with a CCN textbook (`http://ccnbook.colorado.edu`) that explores the latest instantiations of the increasingly brain-inspired versions of the PDP models from the early days. While flexible, the platform is also principled and biased towards convergence on a middle-of-the-road approach, with wizards enabling the creation of cognitive architectures linking PFC, hippocampus, midbrain, vision and more.

Keywords: CCN; HPC; cognitive modeling; cognitive architectures; model repository; collaboration

About the *emergent platform™*

The *emergent platform™* consists of the *emergent™* simulator (Aisa, Mingus, & O'Reilly, 2008) for CCN (cognitive computational neuroscience (Kelso & Tognoli, 2009) and computational cognitive neuroscience (O'Reilly, 1998)) and supporting infrastructure for model-lifecycle management and community model development. It has been developed as open source (GPLv2) cross-platform software (Windows, Mac and Linux), with a long history of more than 10 years. Originating in PDP (McClelland, 2015) and PDP++ (O'Reilly, Dawson, & McClelland, 2005), its main strength lies in the cognitive architecture of Leabra (O'Reilly, 1996; O'Reilly, Hazy, & Herd, 2015), PVLV (O'Reilly, Frank, Hazy, & Watz, 2007; Hazy, Frank, & O'Reilly, 2010) and PBWM (Hazy, Frank, & O'Reilly, 2007), providing a biologically inspired, yet high-level abstract modeling framework of brain area interactions (Jilk, Lebiere, O'Reilly, & Anderson, 2008). This includes PFC - basal ganglia loops and how it enables complex operations and development of representations through learning.

Over the years *emergent™* has increasingly evolved into a fully fledged collaborative model lifecycle management tool for CCN research, providing direct built-in repositories for sharing models, analysis programs and task environments together with an integrated documentation tool linking up to a reference and bibliography management system to foster scientific discussions on any such shared models or theories. This integrated approach has been used extensively in the free wiki based CCN textbook (O'Reilly, Munakata, Frank, Hazy, & Contributors, 2016). In this paper we now briefly present some of the recent advancements of workflow in this powerful tool and how it can help to collaboratively create increasingly complex and complete cognitive models to encapsulate and codify the current state of scientific knowledge.

Simpler models in *emergent™* can be run on a single modern desktop or laptop, however, scaling models up to simulate interesting complex cognitive behaviours often requires more computational resources than are available locally. *emergent™* offers a complete set of tools to launch simulations as well as analyze and manage its data on remote HPC or other scientific computing resources from within the standard UI, making scaling up models and or parallelizing parameter searches a straightforward extension to the standard modeling workflow. The frontend and backend communicate through a shared SVN repository which provides two useful benefits. 1) It is robust to most firewall configurations in standard university HPC facilities, allowing to drive all operations from any internet connected client without needing to log-in to the HPC resource itself 2) It provides a complete version history of all experiments run and their outcome. Thus the HPC interface simultaneously also acts as a built-in shared lab logbook, allowing modellers to keep notes on past efforts and rapidly graphing and comparing any such attempts.

Even though setting up *emergent™* to use one's own dedi-

cated HPC cluster enables the most flexibility, this option is not always available to all researchers, and compiling *emergent*™ on a university cluster may put an unnecessarily high technical burden on researchers. To minimize this barrier to entry, *emergent*™ has collaborated with the Neuroscience Gateway Portal (Sivagnanam et al., 2013) to provide at least 5,000 free core hours per user on the XSEDE supercomputer. Since version 8.1, *emergent*™ now comes preconfigured with access to the Comet cluster (part of XSEDE), which can be used by anyone, out of the box, subject to an account verification process.

Science has always been a large scale collaborative process and building cognitive models representing the function of numerous parts of the brain is no exception. To foster team-based research *emergent*™ includes numerous collaborative tools. For example *emergent*™ provides the functionality to publish, as well as load projects and utility programs such as common analysis or task environments, straight from its user interface and supports both private team repositories, as well as public repositories hosted by the CCNLab. *emergent*™ also provides built-in support for SVN as well as a patch management system to allow for a straightforward UI-driven distributed development model. We invite the community of CCN researchers to help us develop a more extensive library of common cognitive tasks, as well as all published models written in *emergent*™.

Summary

emergent™ provides an extensive collection of tools for modeling cognitive process in the brain and provides a platform approach to collaborative development of complex brain models. In sum, we are excited to release the emergent platform to a global community of brain-mind researchers all working together to characterize and understand our human nature.

Acknowledgements

emergent™'s development is currently funded by the following grants: ONR N00014-14-1-0670, ONR D00014-12-C-0638 and NIH R01GM109996.

References

Aisa, B., Mingus, B., & O'Reilly, R. (2008, October). The emergent neural modeling system. *Neural Networks*, *21*(8), 1146–1152.

Hazy, T. E., Frank, M. J., & O'Reilly, R. C. (2007, August). Towards an executive without a homunculus: Computational models of the prefrontal cortex/basal ganglia system. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*, *362*(1), 105–118.

Hazy, T. E., Frank, M. J., & O'Reilly, R. C. (2010, April). Neural mechanisms of acquired phasic dopamine responses in learning. *Neuroscience and Biobehavioral Reviews*, *34*(5), 701–720.

Jilk, D., Lebiere, C., O'Reilly, R., & Anderson, J. (2008, September). SAL: An explicitly pluralistic cognitive architecture. *Journal of Experimental & Theoretical Artificial Intelligence*, *20*(3), 197–218.

Kelso, J. A. S., & Tognoli, E. (2009). Toward a Complementary Neuroscience: Metastable Coordination Dynamics of the Brain. In N. Murphy, G. F. R. Ellis, & T. O'Connor (Eds.), *Downward Causation and the Neurobiology of Free Will* (pp. 103–124). Springer Berlin Heidelberg.

McClelland, J. L. (2015). *Explorations in Parallel Distributed Processing: A Handbook of Models, Programs, and Exercises*.

O'Reilly, R. C. (1996, January). Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural Computation*, *8*(5), 895–938.

O'Reilly, R. C. (1998, January). Six Principles for Biologically-Based Computational Models of Cortical Cognition. *Trends in Cognitive Sciences*, *2*(11), 455–462.

O'Reilly, R. C., Dawson, C. K., & McClelland, J. L. (2005). *The PDP++ Software (Version 3.1) [Computer software and manual]*. (Published: Retreived from http://psych.colorado.edu/ oreilly/PDP++/ PDP++.html)

O'Reilly, R. C., Frank, M. J., Hazy, T. E., & Watz, B. (2007, February). PVLV: The primary value and learned value Pavlovian learning algorithm. *Behavioral Neuroscience*, *121*(1), 31–49.

O'Reilly, R. C., Hazy, T. E., & Herd, S. A. (2015). The Leabra cognitive architecture: How to play 20 principles with nature and win! In S. Chipman (Ed.), *Oxford handbook of cognitive science.* Oxford University Press.

O'Reilly, R. C., Munakata, Y., Frank, M. J., Hazy, T. E., & Contributors. (2016). *Computational cognitive neuroscience*. Wiki book, 3rd edition, URL: `http://ccnbook.colorado.edu`. Retrieved from `http://ccnbook.colorado.edu`

Sivagnanam, S., Majumdar, A., Yoshimoto, K., Astakhov, V., B, A., Martone, M., & Carnevale, N. T. (2013). Introducing The Neuroscience Gateway. In *Ceur workshop proceedings* (Vol. 993).

Wlodzislaw, D., & Dobosz, K. (2011). Visualization for understanding of neurodynamical systems. *Cognitive Neurodynamics*, *5*, 145–160.

Human-artificial-intelligence hybrid learning systems

Seth HERD [a,1], Geoffrey URLAND [b] and Brian MINGUS [a] and Randall O'REILLY [a]

[a] *Department of Psychology and Neuroscience, University of Colorado, 345 UCB, Boulder, Colorado 80309*

[b] *Toravner, 637 S 40th St, Boulder, Colorado 80305*

Abstract.

The only known path to general intelligence is that taken by humans. Adapting elements of this path to achieving artificial general intelligence (AGI) has become a common area of interest. We address the role of human teachers in this process, using the concept of the zone of proximal development (ZPD). We explore the range of possible human-teacher interactions, including those modeled closely on humans, those involving accessing and changing the AGI learners internal representations, and tighter integrations amounting to human-AI hybrid learning system (HAIHLS). In such a system, a human teacher scaffolds an untrained subsystem by producing the outputs desired from a fully trained version. Those outputs both train that subsystem and provide more useful information to the remainder of the cognitive system. This aid enables all subsystems to learn within the context of the richer behavior and cognition possible with the aid of the human subsystem.

Keywords. Motivation, Learning, Neural Network

Introduction

There is only one known path to a generally intelligent system; that taken by humans. An outstanding question in artificial intelligence research is how to capitalize on our knowledge of that path, while taking any available shortcuts. Embodied and brain-mimetic systems have been of increasing interest to AI researchers [1,2], but the path also includes human self-selection of learning experiences (Self-Directed Learning, [3]) and the intervention of human teachers. Here we discuss a number of ways in which a human teacher could help a properly designed machine learning system to achieve general intelligence.

[1]Corresponding Author: E-mail: seth.herd@colorado.edu

The general idea of having humans teach AGI learners is far from new. Many existing systems are trained in a supervised style, using human-labeled data. This obvious application of human teachers is useful, but suffers from inflexibility and time inefficiency (human teachers must spend time before training to classify or be quickly available during training to provide classifications). Even with adequate resources, it is especially impractical for an embodied learner to have individual sense-inputs classified by humans; the system would have to pause to allow humans to classify its inputs (imagine that while navigating a novel landscape an embodied learner has to stop and wait for its human teacher to answer whenever it cannot identify a new object!). But humans can classify a few crucial situations that will be particularly helpful to the learner, if those can be identified by either the learner or its teacher(s).

Human teachers offer undeniable benefits to human learners. Much human teaching involves pushing learning into the Zone of Proximal Development (ZPD), the space of problems that can be solved by a learner when aided by a teacher [4]. We follow the generalization of this proposal, that all types of learning are more effective when a teacher helps a learner to expand their abilities. This assumption is implicitly shared across all (human) educational systems. Existing approaches to human-aided machine learning all constitute some variety of generalized ZPD, in that the human teacher in some way extends the abilities of the machine learner. These include variations based on conditioning (such as reinforcement learning [5], shaping [6], and active learning [7]), demonstration [8], and explicit human direction [9,10].

1. Humans Teaching Artificial Agents

Beyond simply following the model of human teacher-learner interactions, many other routes are available to a human in teaching a machine. We focus here on the idea that humans could aid machines by aiding or actually playing the role of specific subcomponents of their cognitive system. We term this type of interaction Human-Artificial-Intelligence Hybrid Learning Systems (HAIHLS). The human component could either serve to scaffold that system by having it learn from the humans contributions, or simply enable other parts of the system to learn more effectively by pushing it further into a zone of proximal development. We focus here on the example of a human aiding and/or standing in for elements of the reward-prediction element of a machines motivational component, but the idea could be applied to any cognitive subsystem.

Another (non-exclusive) strategy is to have a human directly observe and/or change the learners internal representations and knowledge structures. A properly designed interface could allow a teacher to both know what the learner is thinking, and to influence that thinking much more reliably than is possible with human learners.

Human student-teacher interactions and their adaptation to teaching AGI learners In the human learning environment, teachers fill a variety of roles. Many of these can be applied in a straightforward way to human teachers helping AGI learners [11]. Existing work has also adapted animal training techniques to teach-

ing an artificial agent [13]. We will skip to some less obvious adaptations of human student-teacher relationships.

One subtle role of a human teacher involves gauging when a learner could use information that it does not yet know to ask for. Human instructors can gauge, by gaze and more subtle action patterns, what is currently puzzling to the learner, and supply crucial conceptual information to fill gaps. With AGI learners, a variety of internal variables can be made available to help the teacher gauge what information to supply. The teacher can then provide that information, perhaps in conjunction with guiding the learners attention, through external (gestures or spoken labels) or internal (directly providing sensory inputs and/or guiding its sensory apparatus to objects of importance).

The Socratic method, in which a teacher asks questions that are carefully chosen to clarify a learners conceptual framework, also has a direct applicability to teaching machines. The teacher can focus the machines learning efforts on crucial questions (e.g., what are you? or what is important?) and through further questioning guide the learner to the desired outcome while helping the learner build a conceptual framework for itself.

The potential of accessing and affecting a machine's internal representations in real time affords a number of variants on human student-teacher interactions. These could prove useful alternatives to either a pure programming or pure learning approach to AGI development. We focus here on perhaps the most extreme variant: treating a human as one component of a cognitive architecture.

2. Human-Artificial-Intelligence Hybrid Learning Systems

Progressing from narrow AI to artificial general intelligence (AGI) will require the integration of many cognitive subsystems into a functional whole. Little work to date has addressed the new challenges inherent in doing so. One novel application of Vygotsky's concept of scaffolding a learner into a zone of proximal development is to stand in for cognitive systems that are relatively less capable. We call a system incorporating human teachers as cognitive subsystems a Human-Artificial Intelligence Hybrid Learning System (HAIHLS).

This approach has two advantages: first, it allows the other subsystems to learn in the context of a more complete, highly functional whole. As such, that system can learn in a context more like its intended functional setting. Second, a human working as a "subsystems" can serve to train its replacement.

In this approach, an untrained machine learning subcomponent rides along and learns from how the human performs their computational role, in the context of the whole, functioning system. As machine learning systems become more reliable and flexible, this approach could circumvent the need for carefully engineering systems, and even circumvent the need for understanding what training signals and learning criteria are used by the analogous system in the human brain. For instance, rather than discovering that human infants are intrinsically motivated to move and so learn by motor babbling, [14], the system could be trained directly to make productive motor movements by learning based on control signals are sent to its actuators by a skilled and goal-aware human. This type

of learning goes far beyond reinforcement learning, by providing a rich (vector) training signal appropriate to a cognitive and sensory situation.

2.1. Example: Human as Object Recognition System

A human observer could receive the visual input received by an embodied AGI learner (in some partly-filtered form) and classify the object. Instead of doing so blindly, the observer could track goal and conceptual representations to provide the most useful possible classification. A banana could be classified as fruit, food, or a toy depending on the systems current questions and concerns. This context-sensitive human object classification would then serve to train the object-recognition system, allowing it to eventually give relevant classifications in similar contexts.

The object recognition subsystem is thus trained in precisely the information environment in which it must perform. The HAIHLSs behavior can be as rich as though it had a fully functioning visual system, and the information supplied from other cognitive subsystems is precisely as it would be once the human is removed from the system. Similarly, the other cognitive subsystems can learn in the context of a fully functional object recognition subsystem; while the behavior of the trained machine subsystem will not be identical to that of the human subsystem, it should be similar enough to provide substantial learning advantages.

2.2. Example: Human as Reward Prediction System

We work within the Intelligent Adaptive Curiosity (IAC) framework [15] and a biologically-motivated adaptation of the same ideas as Self-Directed Learning (SDL) [3]. The reward prediction system in SDL, or the prediction-prediction system within the IAC framework, serves to direct the learner to spend time in areas of behavioral space in which learning is relatively rapid. This is the essence of self-directed learning: learn about what you can learn about, do not waste time on what is, at least currently, unlearnable. In SDL, success at an arbitrary task is rewarding, as it is for humans [22]. The system seeks to keep doing things with rapidly increasing (or possibly just intermediate) levels of reward prediction; anything that never supplies reward is frustrating, while anything that always provides success and therefore reward is boring. In IAC, a rapidly increasing predictability of behavior plays the same role; no change in predictability indicates that the task is either currently unlearnable, or already well-learned. To engineer a complex, successful self-directed learner from the ground up, we would need to discover or deduce what is intrinsically motivating to human learners, and so causes them to choose learning situations adequate to enable general intelligence. However, if a human reward predictor directs the nascent machine learning prediction system toward useful learning situations, this need can be bypassed entirely. For instance, the human subsystem might predict reward whenever the machine is looking at a human whose eyes move toward the machine, but only when the currently active goal is getting attention. In this situation, we need not know what a human infant finds rewarding about getting human attention; the human trains the system, merely by using their own skills and knowledge of what

high-level situations humans do find rewarding. The machine reward-prediction system can then generalize from its own sensory apparatus, and over time develop a suitable representation of what sensory information signals human attention. Human attention will, after that learning, activate the reward prediction system (and thereby the dopamine reward signal) triggering the system to both learn, and to remain attentionally tied to that situation in hopes of more learning opportunities [3].

Inversely, having a human as part of the motivational system could train the system away from useless or dangerous behavioral domains. Energetically banging an actuator into a wall or body could simply be marked by a button press as no fun, and so ceased in favor of new learning. By looking at the relevant goal and context representations, the human as a reward prediction system can perform a much more useful role than simply providing reinforcement signals based on some inflexible criteria (e.g., pain).

These two examples of a human as part of a larger system should illustrate the possibilities inherent in such hybrid systems. A human serving this role is very much acting to scaffold the agents skills, as discussed in expansions on Vygotskys ZPD framework [17,18]. The human functions very much as a scaffolding does for a building; it supports the structure as it is built, and is removed when the edifice is complete and can stand on its own.

3. Caveats and conclusions

It bears more than a casual mention that the sort of self-directed learning system we expand upon here has drawn severe criticism, for reasons that we find highly convincing [19,20]. There is a real possibility that a successful AGI learning system will achieve its goals very successfully. By this logic, the motivational structure of our AGIs becomes extremely important. And even the seemingly benign goal of constant learning implicit in both the IAC and SDL approaches could prove disastrous when taken to the extreme.

In sum, it seems likely that some variant of using humans as subsystems that can stand in for and train parts of an AGI learners cognitive architecture will prove useful. We have suggested a range of ideas; we now await specific implementations to provide specificity and see which approaches bear the most fruit.

References

[1] M. C. Anderson, Rethinking interference theory: Executive control and the mechanisms of forgetting, Journal of Memory and Language 49 (2003) 415–445.

[2] Brunette, E.S., Flemmer, R.C., Flemmer, C.L.: A Review of Artificial Intelligence. In: Proceedings of the Fourth International Conference on Autonomous Robots and Agents, (2009) 385–392

[3] Herd, S.A., Mingus, B., O'Reilly, R.C.: Dopamine and Self-directed Learning. In: Biologically Inspired Cognitive Architectures 2010: Proceedings of the First Annual Meeting of the BICA Society,(2010) 58–63. IOS Press, Fairfax, VA

[4] L. S. Vygotsky, Mind in society: The development of higher psychological processes, Harvard University Press, Cambridge, MA, 1978.

[5] A. L. Thomaz, C. Breazeal, Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance.
[6] Knox, W.B., Fasel, I., Stone, P.: Design Principles for Creating Human- shapable Agents. In: AAAI Spring 2009 Symposium on Agents that Learn from Human Teachers (2009)
[7] Settles, B.: Active Learning Literature Survey. Technical Report, University of WisconsinMadison (2009)
[8] A. Y. Ng, Reinforcement learning and apprenticeship learning for robotic control. In: Balczar, J., Long, P., Stephan, F. (eds.) Algorithmic Learning Theory. LNCS, vol. 4264 (2006) 29–31. Springer, Heidelberg
[9] Chernova, S., Veloso, M.: Teaching Multi-Robot Coordination Using Demonstration of Communication and State Sharing. In: Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS), 1183–1186 (2008)
[10] Thomaz A.L., Breazeal, C.: Experiments in socially guided exploration: lessons learned in building robots that learn with and without human teachers. Connect. Sci., 20 (2008) 91–110
[11] Thomaz, A.L., Cakmak, M.: Learning About Objects with Human Teachers. In: Proceedings of the 4th ACM/IEEE international conference on Human robot interaction. (2009) 15–22. ACM, New York, NY
[12] Zhu, X., Goldberg, A.: Introduction to Semi-Supervised Learning. In: Brachman, R., Dietterich, T. (eds.) Synthesis Lectures on Artificial Intelligence and Machine Learning, 3 (2009) 1–130. Morgan & Claypool Publishers, San Rafael, CA
[13] Goertzel, B., Pennachin, C., Geissweiller, N., Looks, M., Senna, A., Silva, W., Heljakka, A., Lopes, C.: An Integrative Methodology for Teaching Embodied Non-Linguistic Agents, Applied to Virtual Animals in Second Life. In: Proceedings of the First Artificial General Intelligence Conference (2008) IOS Press, Amsterdam, The Netherlands
[14] Demiris, Y., Dearden, A.: From Motor Babbling to Hierarchical Learning by Imitation: A Robot Developmental Pathway. In: Proceedings of the Fifth International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems (2005) 31–37
[15] Oudeyer, P.Y., Baranes, A., Kaplan, F.: Intrinsically Motivated Exploration for Developmental and Active Sensorimotor Learning. In: Sigaud, O., Peters, J. (eds.) From Motor Learning to Interaction Learning in Robots: Studies in Computational Intelligence (2010) 264, pp. 107–146. Springer, Heidelberg
[16] Aron, A.R., Shohamy, D., Clark, J., Myers, C., Gluck, M.A., Poldrack R.A.: Human Midbrain Sensitivity to Cognitive Feedback and Uncertainty During Classification Learning. J. Neurophis., 92, 1144–1152 (2004)
[17] Conner, D. B., Cross, D. R.: Longitudinal Analysis of the Presence, Efficacy and Stability of Maternal Scaffolding During Informal Problem-Solving Interactions. B. J. Dev. Psych., 21 (2003) 315–334
[18] Kermani, H., Brenner, M.E.: Maternal Scaffolding in the Child's Zone of Proximal Development across Tasks: Cross-Cultural Perspectives. J. Rsch. Chil. Ed., 15, (2000) 30-52
[19] Mijic, R. Bootstrapping Safe AGI Goal Systems: CEV and Variants Thereof. In: Proceedings of the Third Conference on Artificial General Intelligence. Atlantis Press (2010)
[20] Yudkowsky, E.: Artificial Intelligence as a Positive and Negative Factor in Global Risk. In: Bostrom, N., Cirkovic, M. (eds.) Global Catastrophic Risks. Oxford University Press, New Yoprk, NY (2008)