nature neuroscience

Using goal-driven deep learning models to understand sensory cortex

Daniel L K Yamins[1,2] & James J DiCarlo[1,2]

Fueled by innovation in the computer vision and artificial intelligence communities, recent developments in computational neuroscience have used goal-driven hierarchical convolutional neural networks (HCNNs) to make strides in modeling neural single-unit and population responses in higher visual cortical areas. In this Perspective, we review the recent progress in a broader modeling context and describe some of the key technical innovations that have supported it. We then outline how the goal-driven HCNN approach can be used to delve even more deeply into understanding the development and organization of sensory cortical processing.

What should one expect of a model of sensory cortex?

Brains actively reformat incoming sensory data to better serve their host organism's behavioral needs (**Fig. 1a**). In human vision, retinal input is converted into rich object-centric scenes; in human audition, sound waves become words and sentences. The core problem is that the natural axes of sensory input space (for example, photoreceptor or hair cell potentials) are not well-aligned with the axes along which high-level behaviorally relevant constructs vary. For example, in visual data, object translation, rotation, motion in depth, deformation, lighting changes and so forth cause complex nonlinear changes in the original input space (the retina). Conversely, images of two objects that are ecologically quite distinct—for example, different individuals' faces—can be very close together in pixel space. Behaviorally relevant dimensions are thus 'entangled' in this input space, and brains must accomplish the untangling[1,2].

Two foundational empirical observations about cortical sensory systems are that they consist of a series of anatomically distinguishable but connected areas[3,4] (**Fig. 1b**) and that the initial wave of neural activity during the first 100 ms after a stimulus change unfolds as a cascade along that series of areas[2]. Each individual stage of the cascade performs very simple neural operations such as weighted linear sums of inputs or nonlinearities such as activation thresholds and competitive normalization[5]. However, complex nonlinear transformations can arise from simple stages applied in series[6]. Since the original input entanglement was highly nonlinear, the untangling process must also be highly nonlinear.

The space of possible nonlinear transformations that the brains neural networks could potentially compute is vast. A major challenge in understanding sensory systems is thus systems identification: identifying which transformations the true biological circuits are using. While identifying summaries of neural transfer functions (for example, receptive field characterization) can be useful[7], solving this systems identification problem ultimately involves producing an encoding model: an algorithm that accepts arbitrary stimulus inputs (for example, any pixel map) and outputs a correct prediction of neural responses to that stimulus. Models cannot be limited just to explaining a narrow phenomenon identified on carefully chosen neurons, defined only for highly controlled and simplified stimuli[8,9]. Operating on arbitrary stimuli and quantitatively predicting the responses of all neurons in an area are two core criteria that any model of a sensory area must meet (see **Box 1**).

Moreover, a comprehensive encoding model must not merely predict the stimulus-response relationship of neurons in one final area, such as (in vision) anterior inferior temporal cortex. Instead, the model must also be mappable: having identifiable components corresponding to intermediate cortical areas (for example, V1, V2, V4) and, ultimately, subcortical circuits as well. The model's responses in each component area should correctly predict neural response patterns within the corresponding brain area (**Fig. 1c**).

Hierarchical convolutional neural networks

Starting with the seminal work of Hubel and Wiesel[10], work in visual systems neuroscience has shown that the brain generates invariant object recognition behavior via a hierarchically organized series of cortical areas, the ventral visual stream[2]. A number of workers have built biologically inspired neural networks generalizing Hubel and Wiesel's ideas (for example, refs. 11–15). Over time, it was realized that these models were examples of a more general class of computational architectures known as HCNNs[16]. HCNNs are stacks of layers containing simple neural circuit motifs repeated across the sensory input; these layers are then composed in series. (Here, "layer" is used in the neural network sense, not in the cortical anatomy sense.) Each layer is simple, but a deep network composed of such layers computes a complex transformation of the input data—analogous to the transformation produced in the ventral stream.

The motifs in a single HCNN layer

The specific operations comprising a single HCNN layer were inspired by the ubiquitously observed linear-nonlinear (LN) neural motif[5]. These operations (**Fig. 1c**) include (i) filtering, a linear operation that takes the dot product of local patches in the input stimulus with a set of templates, (ii) activation, a pointwise nonlinearity—typically either

[1]Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA. [2]McGovern Institute for Brain Research, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA. Correspondence should be addressed to D.L.K.Y. (yamins@mit.edu).

Received 26 October 2015; accepted 13 January 2016; published online 23 February 2016; doi:10.1038/nn.4244

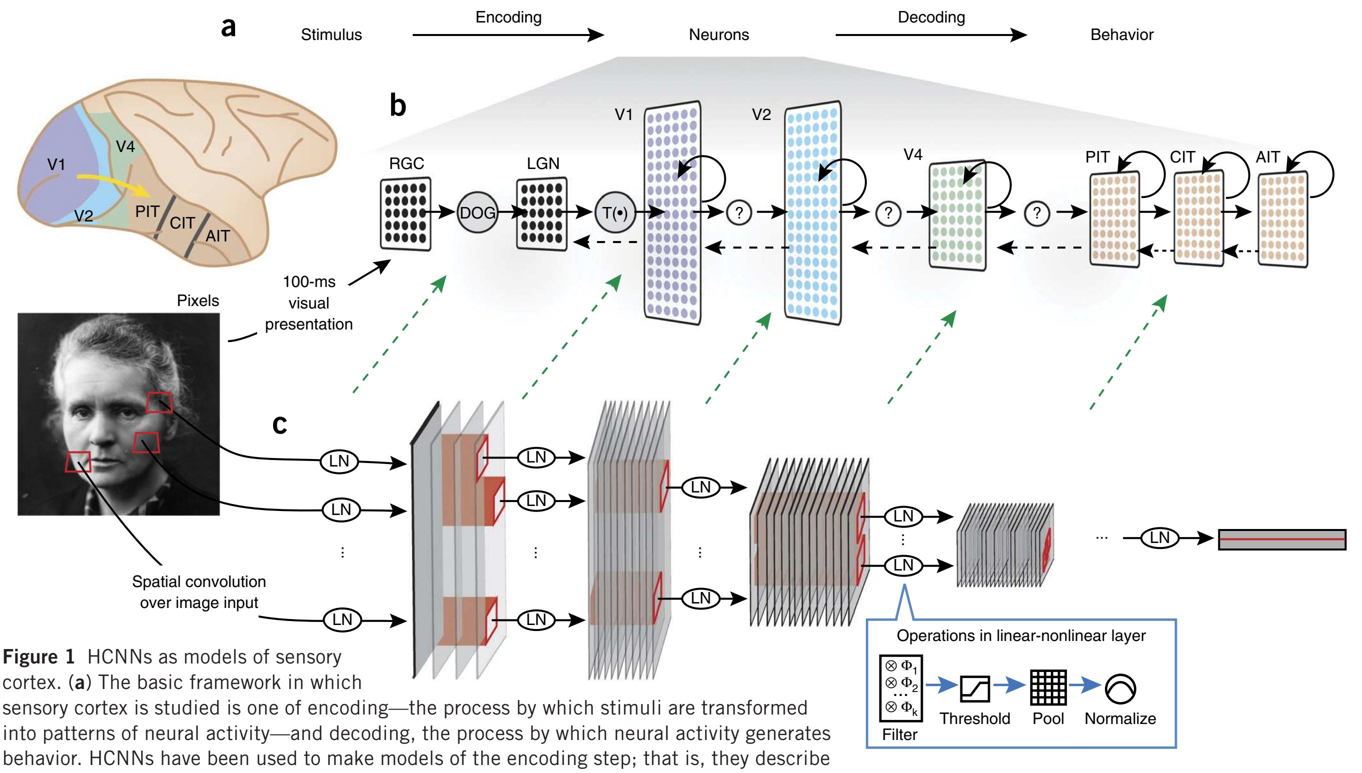

Figure 1 HCNNs as models of sensory cortex. (**a**) The basic framework in which sensory cortex is studied is one of encoding—the process by which stimuli are transformed into patterns of neural activity—and decoding, the process by which neural activity generates behavior. HCNNs have been used to make models of the encoding step; that is, they describe the mapping of stimuli to neural responses as measured in brain. (**b**) The ventral visual pathway is the most comprehensively studied sensory cascade. It consists of a series of connected cortical brain areas (macaque brain shown). PIT, posterior inferior temporal cortex; CIT, central; AIT, anterior; RGC, retinal ganglion cell; LGN, lateral geniculate nucleus. DoG, difference of Gaussians model; T(•), transformation. (**c**) HCNNs are multilayer neural networks, each of whose layers are made up of a linear-nonlinear (LN) combination of simple operations such as filtering, thresholding, pooling and normalization. The filter bank in each layer consists of a set of weights analogous to synaptic strengths. Each filter in the filter bank corresponds to a distinct template, analogous to Gabor wavelets with different frequencies and orientations; the image shows a model with four filters in layer 1, eight in layer 2, and so on. The operations within a layer are applied locally to spatial patches within the input, corresponding to simple, limited-size receptive fields (red boxes). The composition of multiple layers leads to a complex nonlinear transform of the original input stimulus. At each layer, retinopy decreases and effective receptive field size increases. HCNNs are good candidates for models of the ventral visual pathway. By definition, they are image computable, meaning that they generate responses for arbitrary input images; they are also mappable, meaning that they can be naturally identified in a component-wise fashion with observable structures in the ventral pathway; and, when their parameters are chosen correctly, they are predictive, meaning that layers within the network describe the neural response patterns to large classes of stimuli outside the domain on which the models were built.

a rectified linear threshold or a sigmoid, (iii) pooling, a nonlinear aggregation operation—typically the mean or maximum of local values[13], and (iv) divisive normalization, correcting output values to a standard range[17]. Not all HCNN incarnations use these operations in this order, but most are reasonably similar. All the basic operations exist within a single HCNN layer, which is then typically mapped to a single cortical area.

Analogously to neural receptive fields, all HCNN operations are applied locally, over a fixed-size input zone that is typically smaller than the full spatial extent of the input (**Fig. 1c**). For example, on a 256 × 256 pixel image, a layer's receptive fields might be 7 × 7 pixels. Because they are spatially overlapping, the filter and pooling operations are typically 'strided', meaning that output is retained for only a fraction of positions along each spatial dimension: a stride of 2 in image convolution will skip every second row and column.

In HCNNs, filtering is implemented via convolutional weight sharing, meaning that the same filter templates are applied at all spatial locations. Since identical operations are applied everywhere, spatial variation in the output arises entirely from spatial variation in the input stimulus. It is unlikely the brain literally implements weight sharing, since the physiology of the ventral stream and other sensory cortices appears to rule out the existence of a single master location in

Box 1 Minimal criteria for a sensory encoding model

We identify three criteria that any encoding model of a sensory cortical system should meet:
Stimulus-computability: The model should accept arbitrary stimuli within the general stimulus domain of interest;
Mappability: The components of the model should correspond to experimentally definable components of the neural system; and
Predictivity: The units of the model should provide detailed predictions of stimulus-by-stimulus responses, for arbitrarily chosen neurons in each mapped area.
These criteria may sometimes be in tension—insisting on mappability at the finest grain might hinder identifying models that actually work for complex real-world stimuli, since low-level circuit tools may operate best in reduced stimulus regimes. While seeking detailed models of neural circuit connectivity in simplified contexts is important, if such models do not add up in the aggregate to accurate predictors of neural responses to real-world stimuli, the utility of their lower-level verisimilitude is limited.

Box 2 Mapping models to neural sensory systems

How does one map artificial neural networks to real neurons? Several approaches are possible, at varying levels of neural detail.

Task information consistency. At the coarsest level, a useful metric of model similarity to a system is the consistency of patterns of explicitly decodable information available to support potential behavioral tasks. In this approach, populations of 'neurons' from a model and populations of recorded neurons are analyzed with identical decoding methods on a battery of high-level tasks (for example, object recognition, face identification and so forth). While not required, it is useful to use simple decoders such as linear classifiers or linear regressors[1,32,63,64], as these embody hypothetical downstream decoding circuits[65,66]. This procedure generates a pattern of response choices for both the model and the neural population. These patterns are then compared to each other either at a coarse grain (for example, via accuracy levels for each task[32]) or a fine grain (stimulus-by-stimulus response consistency). We note that this approach naturally connects to the linkage between neuronal populations and behavior[32], as both models and neurons can be compared to behavioral measurements from either in animal or humans subjects. Both the neural area thought be most directly connected to behavior (for example, IT in the visual case) and the computational model of this area should exhibit high consistency with those behavioral patterns[32].

Population representational similarity. Another population-level metric is representational similarity analysis[29,35], in which the two representations (that of the real neurons and that of the model) are characterized by their pairwise stimulus correlation matrix (**Fig. 2d**). For a given set of stimuli, this matrix describes how far apart a representation 'thinks' each pair of stimuli are. These distance matrices are then compared for similarity: the model is judged to be similar to the neural representation if it treats stimuli pairs as close to (or far from) each other whenever the real neural population representation also does so.

Single-unit response predictivity. A finer grained mapping of models to neurons is that of linear neural response predictivity of single units[33]. This idea is best understood via a simple thought experiment: imagine one had measurements from all neurons in a given brain area in two animals: a source animal and a target animal. How would one map the neurons in the source to neurons in the target? In many brain areas (such as, for example, V4 or IT), there might not be an exact one-to-one mapping of units between the animals. However, it is reasonable to suppose that the two animals' areas are the same (or very similar) up to linear transform—for example, that units in the target animal are approximately linear combinations of (a small number of) units in the source animal. In engineering terms, the animals would be said to be 'equivalent bases' for sensory representation. (If the mapping had to be nonlinear, it would call into question whether the two areas were the same across animals to begin with.) Making the mapping would, in effect, be the problem of identifying the correct linear combinations. The same idea can be used to map units in a model layer to neurons in a brain area. Specifically, each empirically measured neuron is treated as the target of linear regression from units in the model layer. The goal is find linear combinations of model units that together produce a 'synthetic neuron' that will reliably have the same response patterns as the original target real neuron: find $c_i, i \in \{1,\ldots,n\}$ such that

$$r(x) \approx r_{\text{synth}}(x) = \sum_i c_i m_i(x)$$

where $r(x)$ is the response of neuron r to stimulus x, and $m_i(x)$ is the response of the i-th model unit (in some fixed model layer). Accuracy of r_{synth} is then measured as its explained variance (R^2) for r on new stimuli not used to identify the coefficients c_i. Ideally, the number of model source units i that have nonzero weights c_i would be approximately the same as would be found empirically when attempting to map the neurons in one animal to those in same brain area for a different animal.

which shared templates could be stored. However, the natural visual (or auditory) statistics of the world are themselves largely shift invariant in space (or time), so experience-based learning processes in the brain should tend to cause weights at different spatial (or temporal) locations to converge. Shared weights are therefore likely to be a reasonable approximation to the brain's visual system, at least within the central visual field. The real visual system has a strong foveal bias, and more realistic treatment of nonuniform receptive field density might improve models' fits to neural data.

Deep networks through stacking

Since convolutional layer outputs have the same spatial layout as their inputs, output of one layer can be input to another. HCNNs can thus be stacked into deep networks (**Fig. 1c**). Although the local fields seen by units in a single layer have a fixed, small size, the effective receptive field size relative to the original input increases with succeeding layers. Because of repeated striding, deep HCNNs typically become less retinotopic with each succeeding layer, consistent with empirical observations[4]. However, the number of filter templates used in each layer typically increases. Thus, the dimensionality changes through the layers from wide and shallow to deep and narrow (**Fig. 1c**). After many strided layers, the spatial component of the output is so reduced that convolution is no longer meaningful, whereupon networks are typically extended using one or more fully connected layers. The last layer is usually used for readout: for example, for each of several visual categories, the likelihood of the input image containing an object of the given category might be represented by one output unit.

HCNNs as a parameterized model family

HCNNs are not a single model, but rather a parameterized model class. Any given HCNN is characterized by the following:

- discrete architectural parameters, including the number of layers the network contains, as well as, for each layer, discrete parameters specifying the number of filter templates; the local radius of each filtering, pooling and normalization operation; the pooling type; and potentially other choices required by the specific HCNN implementation; and
- continuous filter parameters, specifying the filter weights of convolutional and fully connected layers.

Though parameter choices might seem like mere details, subtle parameter differences can dramatically affect a network's performance on recognition tasks and its match to neural data[15,18].

Given the minimal model criteria described in **Box 1**, a key goal is identifying a single HCNN parameter setting whose layers correspond to distinct regions within the cortical system of interest (for example, different areas in the ventral stream) and which accurately predict response patterns in those areas (see **Box 2**).

While an oversimplification, the relationship between modifying filters and architectural parameters is somewhat analogous to that between developmental and evolutionary variation. Filter parameters are thought of as corresponding to synaptic weights, and their learning algorithms (see discussion of backpropagation below) update parameters in an online fashion. Changing architectural parameters, in contrast, restructures the computational primitives, the number of sensory areas (model layers) and the number of neurons in each area.

Early models of visual cortex in context

A number of approaches have been taken to identify HCNN parameters that best match biological systems.

Hand-designing parameters via Hubel and Wiesel theory. Beginning in the 1970s, before the HCNN concept was fully articulated, modelers started tackling lower cortical areas such as V1, where neurons might be explicable through comparatively shallow networks. Hubel and Wiesel's empirical observations suggested that neurons in V1 resemble Gabor wavelet filters, with different neurons corresponding to edges of different frequencies and orientations[10,19]. Indeed, early computational models using hand-designed Gabor filter banks as convolution weights achieved some success in explaining V1 neural responses[20]. Later it was realized that models could be substantially improved using nonlinearities such as thresholding, normalization and gain control[5,21], helping motivate the HCNN class in the first place. Similar ideas have been proposed for modeling primary auditory cortex[22].

Learning parameters via efficient coding constraints. The work of Barlow, Olshausen and others introduced another way of determining filter parameters[23,24]. Filters were optimized to minimize the number of units activated by any given stimulus while still retaining the ability to reconstruct the original input. Such 'sparse' efficient codings naturally learn Gabor-wavelet-like filters from natural image data, without having to build those patterns in by hand.

Fitting networks to neural data. Another natural approach begun in the mid-1990s was to bring neuroscience data directly to bear on model parameter choice. The idea was to collect response data to various stimuli for neurons in a brain area of interest and then use statistical fitting techniques to find model parameters that reproduce the observed stimulus–response relationship. This strategy had some success fitting shallow networks to visual area V1, auditory area A1 and somatosensory area S1 (reviewed in ref. 25).

Difficulties with deeper networks. Given successful shallow convolutional models of early cortical areas, perhaps deeper models would shed light on downstream sensory areas. However, the deeper models needed to model such higher areas would have many more parameters than V1-like models. How should these parameters be chosen?

The outputs on which higher layers operate are challenging to visualize, making it difficult to generalize the hand-designed approach to deeper networks. Similarly, while some progress has been made in extending efficient coding beyond one layer[26], these approaches also have not yielded effective deeper networks. Multi-layer HMAX networks were created by choosing parameters roughly to match known biological constraints[12,13]. HMAX networks had some success reproducing high-level empirical observations, such as the tolerance ranges of inferior temporal (IT) cortex neurons[12,27] and the tradeoff between single-unit selectivity and tolerance[28].

However, by the mid-2000s, it had become clear that these approaches were all having trouble extending to higher cortical areas such as V4 and IT. For example, HMAX models failed to match patterns of IT population activity on batteries of visual images[29], while multilayered neural networks fit to neural data in V4 and IT ended up overfitting the training data and predicting comparatively small amounts of explained variance on novel testing images[8].

One plausible reason for this lack of success was that the largely feedforward neural networks being explored were too limited to capture the data efficiently. Perhaps more sophisticated network architectures, using feedback[30] or millisecond-scale spike timing[31], would be required. A second possibility was that failure arose from not having enough neural data to fit the model parameters. Single-unit physiology approaches[8] or whole-brain functional MRI[29] could measure responses to perhaps 1,000 independent stimuli, while array electrophysiology[32] could obtain responses to ~10,000 stimuli. In hindsight, the amount of neural data available to constrain such networks was several orders of magnitude too little.

A new way forward: goal-driven networks as neural models

The goal-driven approach is inspired by the idea that, whatever parameters are used, a neural network will have to be effective at solving the behavioral tasks the sensory system supports to be a correct model of a given sensory system. The idea of this approach is to first optimize network parameters for performance on an ethologically relevant task, and then, once network parameters have been fixed, to compare networks to neural data. This approach avoids the severe data limitation of pure neural fitting, as collecting (for example) millions of human-labeled images containing many hard real-world cases of object recognition is far easier than obtaining comparable neural data. The key question becomes: do such top-down goals strongly constrain biological structure? Will performance optimization imposed at the outputs of a network be sufficient to cause hidden layers in the network to behave like real neurons in, for example, V1, V4 or IT? A series of recent results has shown that this might indeed be the case.

The technological bases of the goal-driven approach are recent improvements in optimizing neural networks performance for artificial intelligence tasks. In this section, we discuss how these tools have led to better neural models; in the next, we discuss the technical innovations underlying those tools.

Top hidden layers of categorization-optimized HCNNs predict IT neuronal responses. High-throughput computational experiments evaluating thousands of HCNN models on task performance and neural-predictivity metrics revealed a key correlation: architectures that perform better on high-level object recognition tasks also better predict cortical spiking data[33,34] (**Fig. 2a**). Pushing this idea further by using recent advances from machine learning led to the discovery of hierarchical neural network models that achieved near-human-level performance level on challenging object categorization tasks. It turned out that the top hidden layers of these models were the first quantitatively accurate image-computable model of spiking responses in IT cortex, the highest-level area in the ventral hierarchy[18,33,34] (**Fig. 2b,c**). Similar models have also been shown to predict population aggregate responses in functional MRI data from human IT (**Fig. 2d**)[35,36].

These results are not trivially explained merely by any signal reflecting object category identity being able to predict IT responses. In fact, at the single neuron level, IT neural responses are largely not categorical, and ideal-observer models with perfect access to category and identity information are far less accurate IT models than goal-driven HCNNs[33] (**Fig. 2a,c**). Being a true image-computable neural network model appears critical for obtaining high levels of neural predictivity. In other words: combining two general biological constraints—the behavioral constraint of the object recognition task and the architectural constraint imposed by the HCNN model class—leads to greatly improved models of multiple layers of the visual sensory cascade.

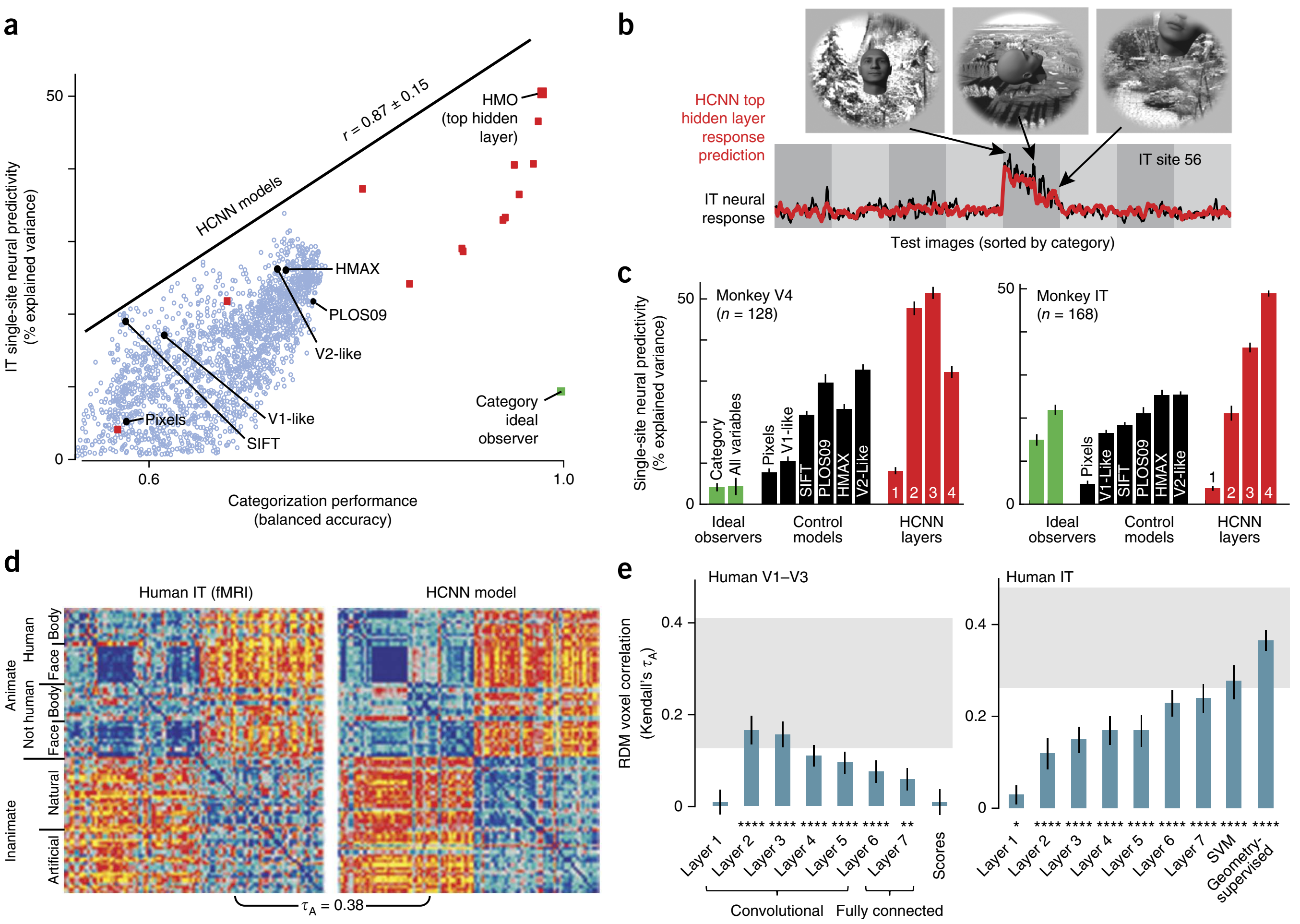

Figure 2 Goal-driven optimization yields neurally predictive models of ventral visual cortex. (**a**) HCNN models that are better optimized to solve object categorization produce hidden layer representations that are better able to predict IT neural response variance. The *x* axis shows performance (balanced accuracy; chance is 50%) of the model output features on a high-variation object categorization task. The *y* axis shows the median single-site IT response predictivity of the last hidden layer of the HCNN model, over $n = 168$ IT sites. Site responses are defined as the mean firing rate 70–170 ms after image onset. Response predictivity is defined as in **Box 2**. Each dot corresponds to a distinct HCNN model from a large family of such models. Models shown as blue circles were selected by random draws from object categorization performance-optimization; black circles show controls and earlier published HCNN models; red squares show the development over time of HCNN models produced during an optimization procedure that produces a specific HCNN model[33]. PLOS09, ref. 15; SIFT, shape-invariant feature transform; HMO, optimized HCNN. (**b**) Actual neural response (black trace) versus model predictions of the last hidden layer of an HCNN model (red trace) for a single IT neural site. The *x* axis shows 1,600 test images, none of which were used to fit the model. Images are sorted first by category identity and then by variation amount, with more drastic image transformations toward the right within each category block. The *y* axis represents the response of the neural site and model prediction for each test image. This site demonstrated face selectivity in its responses (see inset images), but predictivity results were similar for other IT sites[33]. (**c**) Comparison of IT and V4 single-site neural response predictivity for various models. Bar height shows median predictivity, taken over 128 predicted units in V4 (left panel) or 168 units in IT (right panel). The last hidden layer of the HCNN model best predicts IT responses, while the second-to-last hidden layer best predicts V4 responses. (**d**) Representational dissimilarity matrices (RDMs) for human IT and HCNN model. Blue color indicates low values, where representation treats image pairs as similar; red color indicates high values, where representation treats image pairs as distinct. Values range from 0 to 1. (**e**) RDM similarity, measured with Kendall's τ_A, between HCNN model layer features and human V1–V3 (left) or human IT (right). Gray horizontal bar represents range of performance of the true model given noise and intersubject variation. Error bars are s.e.m. estimated by bootstrap resampling of the stimuli used to compute the RDMs. $*P < 0.05$, $**P < 0.001$, $****P < 0.0001$ for difference from 0. Panels **a**–**c** adapted from ref. 33, US National Academy of Sciences; **d** and **e** adapted from ref. 35, S.M. Khaligh-Razavi and N. Kriegeskorte.

Though the top hidden layers of these goal-driven models end up being predictive of IT cortex data, they were not explicitly tuned to do so; indeed, they were not exposed to neural data at all during the training procedure. Models thus succeeded in generalizing in two ways. First, the models were trained for category recognition using real-world photographs of objects in one set of semantic categories, but were tested against neurons on a completely distinct set of synthetically created images containing objects whose semantic categories were entirely non-overlapping with that used in training. Second, the objective function being used to train the network was not to fit neural data, but instead the downstream behavioral goal (for example, categorization). Model parameters were independently selected to optimize categorization performance, and were compared with neural data only after all intermediate parameters—for example, nonlinear model layers—had already been fixed.

Stated another way, within the class of HCNNs, there appear to be comparatively few qualitatively distinct, efficiently learnable solutions to high-variation object categorization tasks, and perhaps the brain is forced over evolutionary and developmental timescales to pick such a solution. To test this hypothesis it would be useful to identify non-HCNN

Box 3 The meaning of 'understanding' in a complex sensory system

What does it mean to understand a complex neural system[67]? In this Perspective, we have suggested that successful models are image-computable, mappable and quantitatively predictive. But do models that meet these criteria necessarily represent understanding? It can be argued that deep neural networks are black boxes that give limited conceptual insight into the neural systems they aim to explain. Indeed, the very fact that deep HCNNs are able to predict the internal responses of a highly complex system performing a very nonlinear task suggests that, unlike earlier toy models, these deeper models will be more difficult to analyze than earlier models. There may be a natural tradeoff between model correctness and understandability.

Optimal stimulus and perturbation analysis. However, one of the key advantages of an image-computable model is that it can be analyzed in detail at low cost, making high-throughput 'virtual electrophysiology' possible. Recent techniques that optimize inputs either to match the statistics of target images or to maximize activation of a single output unit have produced impressive results in texture generation, image style matching and optimal stimulus synthesis (ref. 68 and Mordvintsev, A., Tyka, M. & Olah, C., http://googleresearch.blogspot.com/2015/06/inceptionism-going-deeper-into-neural.html, 2015). These techniques could be used to identify the featural drivers of individual neurons, using the models' efficiency of scale to reduce a huge stimulus space to a set small enough to measure using realistic experimental procedures[69]. Inspired by causal intervention experiments[70], predictions for causal relationships between neural responses and behavior could be obtained by perturbing units within the model, even optimizing stimuli and perturbation patterns to achieve the most effective behavioral changes.

A concrete example of traversing Marr's levels of analysis. Goal-driven models yield higher level insight as well. That functional constraints can produce neurally predictive models is reminiscent of earlier work, including efficient coding hypotheses[23,24]. In both approaches, a driving concept—expressed as an objective function for optimization—explains why parameters are as they are. Unlike efficient coding, goal-driven HCNNs derive their objective function from behaviors that organisms are known to perform, rather than more abstract concepts, such as sparsity, whose ecological relevance is unclear. In this sense, the current work is more similar in spirit to Marr's levels of analysis[71], investigating how a system's computational-level goals influence its algorithmic and implementation level mechanisms. This approach is also related to neuroethology, where the natural behavior of an organism is studied to gain insight into underlying neural mechanisms[72].

models that, when optimized for categorization, achieved high performance. The hypothesis predicts that any such models would fail to predict neural response data.

Intermediate and lower layers predict V4 and V1 responses

In addition to higher model layers mapping to IT, intermediate layers of these same HCNN models turn out to be state-of-the-art predictors of neural responses in V4 cortex, an intermediate visual area that is the main cortical input to IT[33] (**Fig. 2c**). While the fit to IT cortex peaks in the highest hidden model layers, the fit to V4 peaks in the middle layers. In fact, these 'accidental' V4-like layers are significantly more predictive of V4 responses than models built from classical intuitions of what the area might be doing (for example, edge conjunction or curvature representation[37]). Continuing this trend, the lowest layers of goal-driven HCNN models naturally contain a Gabor-wavelet-like activation pattern. Moreover, these lower layers provide effective models of voxel responses in V1–V3 voxel data (**Fig. 2e**)[35,36]. Top-down constraints are thus able to reach all the way down the ventral hierarchy.

A common assumption in visual neuroscience is that understanding tuning curves in lower cortical areas (for example, edge conjunctions in V2 (ref. 38) or curvature in V4 (ref. 39)) is a necessary precursor to explaining higher visual areas. Results with goal-driven deep HCNNs show that top-down constraints can yield quantitatively accurate models of intermediate areas even when descriptive bottom-up primitives have not been identified (see **Box 3**).

HCNN layers as generative models of cortical areas. Unlike previous modeling approaches that fit single nonlinear models for each empirically measured neuron and then describe the distributions of parameters that were found[6], the performance-based approach generates a single model for all neurons simultaneously. Consequently, layers of the deep HCNNs are generative models for corresponding cortical areas, from which large numbers of (for example) IT-, V4- or V1-like units can be sampled. Given that the neurons used to evaluate model correctness were chosen by random electrode sampling, it is likely that any future neurons sampled from the same

Box 4 Gradient backpropagation

The basic idea of the gradient backpropagation algorithm is simple:

1. Formulate the task of interest as a loss function to be minimized—for example, categorization error. The loss function should be piecewise differentiable with respect to both the inputs (for example, images) and the model parameters.
2. Initialize the model parameters either at random or through some well-informed initial guess[14].
3. For each input training sample, compute the derivative of the error function with respect to the filter parameters, and sum these values over the input data.
4. Update network parameters by gradient descent—that is, by moving each parameter a small amount in the direction opposite to the error gradient for that parameter.
5. Repeat steps 3 and 4 until either the training error converges or, if overfitting is a concern, some 'early stopping' criterion is met[14].

The key insight that makes this procedure relatively efficient for feedforward networks is that—simply by applying the chain rule from basic calculus—the derivatives of the error with respect to filter values in a given layer can be efficiently computed from those in the layer just above[42]. Derivative computations thus start at the top layer and then propagate backwards through the network down to the first layers.

Another important technical innovation enabling large-scale backpropagation was stochastic gradient descent (SGD)[42]. SGD involves breaking training data into small, randomly chosen batches. Gradient descent is done on each batch in sequence until the training data are exhausted, at which point the procedure can begin again, usually on newly chosen random batches. SGD enables backpropagation on much larger data sets than previously contemplated and usually converges to a stable solution, though the statistical theory guaranteeing such convergence is not well developed.

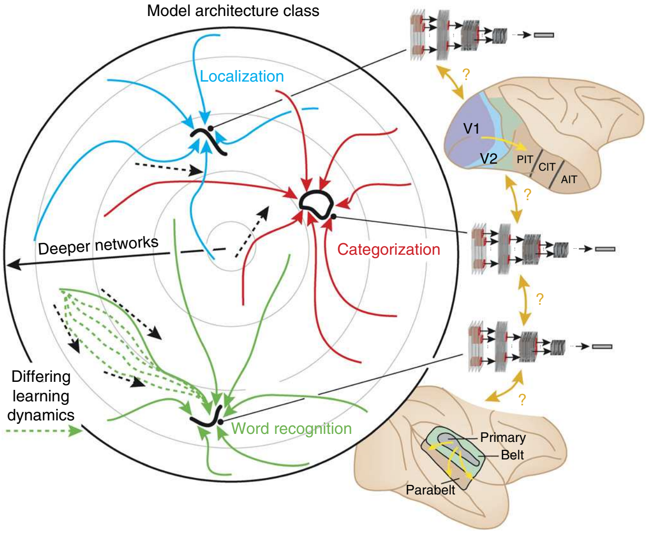

Figure 3 The components of goal-driven modeling. The large circle represents an architectural model class; each point in the space is a full model (examples at right); inner circles represent subspaces of the full model class containing models of a given number of layers. Goal-driven models are built by using learning algorithms (dotted black arrows) that drive systems along trajectories in the model class (solid colored lines) to discover especially optimal models. Each goal can be thought of as corresponding to a basin of attraction within the model class (thick black contours) containing parameters that are especially good for solving that goal. Computational results have shown that tasks put a strong constraint on model parameter settings, meaning that the set of optimal parameters for any given task is very small compared to the original space. These goal-driven models can then be evaluated for how predictive they are of the response properties of neurons in brain areas that are thought to underlie behavior in a given task domain. For example, the units of a model optimized for word recognition could be compared to response properties in the primary, belt and parabelt regions of auditory cortex[40]. Models can also be compared to each other to determine to what extent different types of tasks lead to shared neural structures. Various component rules (supervised, unsupervised or semi-supervised) can also be studied to determine how they might lead to different dynamics during postnatal development or expertise learning (dashed green paths).

areas will be equally well predicted, without having to update model parameters or train any new nonlinear functions.

Application to auditory cortex. A natural idea is to apply goal-based HCNN modeling to sensory domains that are less well understood than vision. The most obvious candidate for this is audition, where a clear path forward involves producing HCNN models whose top layers are optimized to solve auditory tasks such as speech recognition, speaker identification, natural sound identification and so on. An intriguing possibility is that intermediate layers of such models may reveal previously unknown structures in non-primary auditory cortex. Initial results suggest that this approach holds promise[40].

Factors leading to the improvement of HCNNs

Taking initial inspiration from neuroscience, HCNNs have become a core tool in machine learning. HCNNs have been successful on many tasks, including image categorization, face identification, localization, action recognition, depth estimation and a variety of other visual tasks[41]. Related recurrent versions of deep neural networks have been used to make strides in speech recognition. Here we discuss some of the technical advances that have led to this recent progress.

Hardware-accelerated stochastic error backpropagation for optimizing filter parameters

In supervised learning of a task (for example, car detection in images), one chooses a set of training data, containing both sample inputs (for example. images of cars and non-cars) and labels describing desired results for each input (for example, image category labels, such as "car" or "dog"). Learning algorithms are then used to optimize the parameter settings of the network so that output layers yield the desired labels on the training data[14]. A powerful algorithm for supervised learning of filter parameters from supervised data has been in existence for several decades: error gradient descent by backpropagation[14,42] (see **Box 4**). However, until recently, backpropagation has been computationally impractical at large scales on massive data sets. The recent advent of graphical processing unit (GPU)-accelerated programming has been a great boon because backpropagation computations largely involve either simple pointwise operations or parallel matrix dot-products[15,33,43]. GPUs, which are more neuromorphic than von Neumann CPU architectures, are especially well suited to these operations, routinely yielding speed increases of tenfold or more[15]. Further advances in neuromorphic computing could accelerate this trend[44].

Automated learning procedures for architectural parameters

Discrete architectural parameters (for example, number of layers) cannot easily be optimized by error backpropagation. However, discrete parameters are critical to final network performance[15,18]. Traditionally, these parameters had been chosen by hand, empirically testing various combinations one at a time until improvements were observed. More recently, procedures such as Gaussian process optimization and genetic algorithms have been deployed to learn better architectural parameters automatically[15,45,46].

Large web-enabled labeled data sets

Another important factor in recent advances is the advent of large labeled data sets. In the visual domain, early data sets often consisted of hundreds of images in hundreds of categories[47]. It was eventually realized that such data sets were neither large nor varied enough to provide sufficient training data to constrain the computational architecture[15,48]. A major advance was the release of the ImageNet data set, which contains tens of millions of images in thousands of categories, curated from the Internet by crowd-sourcing[49]. Taking advantage of these large data sets required the efficient hardware-accelerated algorithms described above. Once these were in place, much deeper neural networks could be trained. A rough rule of thumb is that the number of training samples for backpropagation should be 10 times the number of network parameters. Given that the number of parameters in a modern deep network far exceeds 100,000, the need for millions of training samples becomes evident, at least for current parameter learning strategies. (The neural learning algorithms used by the brain are probably significantly more efficient with labeled data than current computational methods for training HCNNs, and may not be subject to the '10×' heuristic.)

A concomitance of small tweaks to architecture class and training methods

A number of other small changes in neural network architecture and training helped improve performance. One especially relevant modification replaced continuously differentiable sigmoid activation functions with half-rectified thresholds[43]. Because these activation functions have constant or zero derivative almost everywhere, they

Box 5 Understanding adversarial optimization effects

An intriguing recent development in the exploration of HCNNs is the discovery of adversarial images: normal photographs that are subtly modified in ways that are undetectable to humans but that cause networks to incorrectly detect arbitrary objects in the modified image[73,74]. In effect, adversarial images demonstrate that existing HCNNs may be susceptible to qualitatively different types of illusions than those that fool humans. These images are created through adversarial optimization, a process in which the pixels of the original image are optimally modified so as to produce the largest changes in the network's final category-detection layer, but with the least disturbance at the pixel level. Creating such images, which may not naturally arise in the physical world, requires complete access to the network's internal parameters.

Thinking along the lines of three components of goal-driven modeling discussed above (and see **Fig. 3**), several possibilities for explaining adversarial examples include (i) that similar effects would be replicable in humans—for example, the creation of idiosyncratic images that fool one human but are correctly perceived by others—if experiments had access to the detailed microcircuitry of that individual brain and could run an adversarial optimization algorithm on it; (ii) that optimization for a categorization goal is brittle, but if richer and more robust optimization goal(s) were used, the effects would disappear; or (iii) that adversarial examples expose a fundamental architectural flaw in HCNNs as brain models, and only by incorporating other network structures (for example, recurrence) will the adversarial examples be overcome. Regardless of which (if any) if these is most correct, understanding adversarial optimization effects would seem to be a critical component of better understanding HCNNs themselves, especially as putative models of the brain.

suffer less from the so-called vanishing-gradients problem, in which error gradients in early layers become too small to optimize effectively. A second type of improvement was the introduction of regularization methods that inject noise during backpropagation into the network to prevent the learning of fragile, overfit weight patterns[43].

The unreasonable effectiveness of engineering

Recent improvements represent the accretion of a number of critical engineering improvements (for example, refs. 50,51). These changes may not signal major conceptual breakthroughs beyond the original HCNN and backpropagation concepts described decades ago, but they nonetheless led to enormous improvement in final results. Large data sets and careful engineering have been much more important than was originally anticipated[52].

Going forward: potentials and limitations

Goal-driven deep neural network models are built from three basic components (**Fig. 3**):

- a model architecture class from which the system is built, formalizing knowledge about the brain's anatomical and functional connectivity;
- a behavioral goal that the system must accomplish, such as object categorization; and
- a learning rule that optimizes parameters within the model class to achieve the behavioral goal.

The results above demonstrate how these three components can be assembled to make detailed computational models that yield testable predictions about neural data, significantly surpassing prior sensory cortical models. Future progress will mean, in part, better understanding each of these three components—as well as their limitations (see **Box 5**).

Improving architecture class

Continued success in using computational models to understand sensory cortex will involve more detailed and explicit mapping between model layers and cortical areas. HCNN operations such as template matching and pooling are neurally plausible, but understanding whether and how the parameterizations used in HCNNs actually connect to real cortical microcircuits is far from obvious. Similarly, while the hierarchy of HCNN model layers appears to generally correspond with the overall order of observed ventral cortical areas, whether the model-layer/brain-area match is one-to-one (or close to it) is far from fully understood. Recent high-performing computer vision networks have greatly increased the number of layers, sometimes to 20 or more[50]. Evaluating whether these very deep networks are better explanations of neural data will be of importance, as deviations from neural fit would suggest that the architectural choices are different from those in the brain. More generally, one can ask, within the class of HCNNs, which architectures, when optimized for categorization performance, best fit the ventral steam neural response data? The results above argue that this could be a new way to infer the architectures in the adult ventral stream.

Such top-down, performance-driven approaches should of course be coupled with state-of-the-art experimental techniques such as two-photon microscopy, optogenetics, electron microscopy reconstruction and other tracing techniques that aim to narrow the class of architectures more directly. Better empirical understanding at the neural circuit level could allow a narrowing in the class of biologically relevant HCNNs, ruling out certain architectures or making informed initial guesses about filter parameters. Models would then need to learn fewer parameters to achieve equal or better neural predictivity.

In both vision and audition, model architecture class could also be improved by building more biologically realistic sensor front-ends into early layers, using known results about subcortical structures[53]. At the opposite end of the scale spectrum, there are large-scale spatial inhomogeneities in higher cortical areas (for example, face patches)[4]. In the lower layers of HCNNs, there is an obvious mapping onto the cortical surface via retinotopic maps, but this relationship is less clear in higher layers. Understanding how multidimensional deep network output may map to two-dimensional cortical sheets, and the implications of this for functional organization, are important open problems.

Improving goal and training-set understanding

The choice of goal and training set has significantly influenced model development, with high-variation data sets exposing the true heterogeneity within real-world categories[33,48,49]. It seems likely that this data-driven trend will continue[52]. A key recent result is that HCNNs trained for one task (for example, ImageNet classification) generalize to many other visual tasks quite different from the one on which they were originally trained[41]. If many relevant tasks come along 'for free' with categorization, which tasks do not? An especially important open challenge is finding tasks that are not solved by categorization optimization but rather require direct independent optimization, and then testing models optimized for these tasks to see if they better explain ventral stream neural data. Developing rich new labeled data sets will be critical to this goal. Understanding how HCNNs systems for various sensory tasks relate to each other, in terms of shared or divergent architectures, would be of interest, both within a sensory domain[54], as well as across domains (for example, between vision and audition; see **Fig. 3**).

Improving learning rule understanding

While it is valuable that supervised learning creates working models that are a remarkably good fit to real perceptual systems, it is physiologically unlikely that cortex is implementing exact backpropagation. A core inconsistency between current deep-learning approaches and real biological learning is that training effective HCNNs requires very large numbers of high-level semantic labels. True biological postnatal learning in humans, higher primates and other animals may use large amounts of unsupervised data, but is unlikely to require such large amounts of externally labeled supervision. Discovering a biologically realistic unsupervised or semi-supervised learning algorithm[55–57] that could produce high levels of performance and neural predictivity would be of interest, from both artificial intelligence and neuroscience viewpoints.

Beyond sensory systems and feedforward networks

Largely feedforward HCNNs cannot provide a full account of dynamics in brain systems that store extensible state, including any that involve working memory, since the dynamics of a feedforward network will converge to the same state independent of input history. However, there is a growing body of literature connecting recurrent neural networks to neural phenomena in attention, decision making and motor program generation[58]. Models that combine rich sensory input systems, as modeled by deep neural networks, with these recurrent networks could provide a fruitful avenue for exploring more sophisticated cognitive behaviors beyond simple categorization or binary decision making, breaking out of the pure 'representation' framework in which sensory models are often cast. This is especially interesting for cases in which there is a complex loop between behavioral output and input stimulus—for example, when modeling exploration of an agent over long time scales in a complex sensory environment[59]. Intriguing recent results from reinforcement learning[60] have shown how powerful in solving strategy-learning problems deep neural network techniques may be. Mapping these to ideas in the neuroscience of the interface between ventral visual cortex and, for example, parietal cortex or the hippocampus will be of great interest[61,62].

Conclusion

In sum, deep hierarchical neural networks are beginning to transform neuroscientists' ability to produce quantitatively accurate computational models of the sensory systems, especially in higher cortical areas where neural response properties had previously been enigmatic. Such models have already achieved several notable results, explaining multiple lines of neuroscience data in both humans and monkeys[33–36]. However, like any scientific advance of importance, these ideas open up as many new questions as they answer. There is much exciting and challenging work to be done, requiring the continued rich interaction between neuroscience, computer science and cognitive science.

COMPETING FINANCIAL INTERESTS
The authors declare no competing financial interests.

Reprints and permissions information is available online at http://www.nature.com/reprints/index.html.

1. DiCarlo, J.J. & Cox, D.D. Untangling invariant object recognition. *Trends Cogn. Sci.* **11**, 333–341 (2007).
2. DiCarlo, J.J., Zoccolan, D. & Rust, N.C. How does the brain solve visual object recognition? *Neuron* **73**, 415–434 (2012).
3. Felleman, D.J. & Van Essen, D.C. Distributed hierarchical processing in the primate cerebral cortex. *Cereb. Cortex* **1**, 1–47 (1991).
4. Malach, R., Levy, I. & Hasson, U. The topography of high-order human object areas. *Trends Cogn. Sci.* **6**, 176–184 (2002).
5. Carandini, M. *et al.* Do we know what the early visual system does? *J. Neurosci.* **25**, 10577–10597 (2005).
6. Sharpee, T.O., Kouh, M. & Reynolds, J.H. Trade-off between curvature tuning and position invariance in visual area V4. *Proc. Natl. Acad. Sci. USA* **110**, 11618–11623 (2013).
7. David, S.V., Hayden, B.Y. & Gallant, J.L. Spectral receptive field properties explain shape selectivity in area V4. *J. Neurophysiol.* **96**, 3492–3505 (2006).
8. Gallant, J.L., Connor, C.E., Rakshit, S., Lewis, J.W. & Van Essen, D.C. Neural responses to polar, hyperbolic, and Cartesian gratings in area V4 of the macaque monkey. *J. Neurophysiol.* **76**, 2718–2739 (1996).
9. Rust, N.C., Mante, V., Simoncelli, E.P. & Movshon, J.A. How MT cells analyze the motion of visual patterns. *Nat. Neurosci.* **9**, 1421–1431 (2006).
10. Hubel, D.H. & Wiesel, T.N. Receptive fields of single neurones in the cat's striate cortex. *J. Physiol. (Lond.)* **148**, 574–591 (1959).
11. Fukushima, K. Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* **36**, 193–202 (1980).
12. Riesenhuber, M. & Poggio, T. Hierarchical models of object recognition in cortex. *Nat. Neurosci.* **2**, 1019–1025 (1999).
13. Serre, T., Oliva, A. & Poggio, T. A feedforward architecture accounts for rapid categorization. *Proc. Natl. Acad. Sci. USA* **104**, 6424–6429 (2007).
14. Bengio, Y. *Learning Deep Architectures for AI* (Now Publishers, 2009).
15. Pinto, N., Doukhan, D., DiCarlo, J.J. & Cox, D.D. A high-throughput screening approach to discovering good forms of biologically inspired visual representation. *PLoS Comput. Biol.* **5**, e1000579 (2009).
16. LeCun, Y. & Bengio, Y. Convolutional networks for images, speech, and time series. in *The Handbook of Brain Theory and Neural Networks* 255–258 (MIT Press, 1995).
17. Carandini, M. & Heeger, D.J. Normalization as a canonical neural computation. *Nat. Rev. Neurosci.* **13**, 51–62 (2012).
18. Yamins, D., Hong, H., Cadieu, C. & Dicarlo, J. Hierarchical modular optimization of convolutional networks achieves representations similar to macaque it and human ventral stream. *Adv. Neural Inf. Process. Syst.* **26**, 3093–3101 (2013).
19. De Valois, K.K., De Valois, R.L. & Yund, E.W. Responses of striate cortex cells to grating and checkerboard patterns. *J. Physiol. (Lond.)* **291**, 483–505 (1979).
20. Jones, J.P. & Palmer, L.A. An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex. *J. Neurophysiol.* **58**, 1233–1258 (1987).
21. Movshon, J.A., Thompson, I.D. & Tolhurst, D.J. Spatial summation in the receptive fields of simple cells in the cat's striate cortex. *J. Physiol. (Lond.)* **283**, 53–77 (1978).
22. Klein, D.J., Simon, J.Z., Depireux, D.A. & Shamma, S.A. Stimulus-invariant processing and spectrotemporal reverse correlation in primary auditory cortex. *J. Comput. Neurosci.* **20**, 111–136 (2006).
23. Barlow, H.B. Possible principles underlying the transformations of sensory messages. in *Sensory Communication* Vol. 1, 217–234 (1961).
24. Olshausen, B.A. & Field, D.J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **381**, 607–609 (1996).
25. deCharms, R.C. & Zador, A. Neural representation and the cortical code. *Annu. Rev. Neurosci.* **23**, 613–647 (2000).
26. Olshausen, B.A., Sallee, P. & Lewicki, M.S. Learning sparse image codes using a wavelet pyramid architecture. *Adv. Neural Inf. Process. Syst.* **14**, 887–893 (2001).
27. Logothetis, N.K., Pauls, J. & Poggio, T. Shape representation in the inferior temporal cortex of monkeys. *Curr. Biol.* **5**, 552–563 (1995).
28. Zoccolan, D., Kouh, M., Poggio, T. & DiCarlo, J.J. Trade-off between object selectivity and tolerance in monkey inferotemporal cortex. *J. Neurosci.* **27**, 12292–12307 (2007).
29. Kriegeskorte, N. Relating population-code representations between man, monkey, and computational models. *Front. Neurosci.* **3**, 363–373 (2009).
30. Ullman, S. Visual routines. *Cognition* **18**, 97–159 (1984).
31. Singer, W. & Gray, C.M. Visual feature integration and the temporal correlation hypothesis. *Annu. Rev. Neurosci.* **18**, 555–586 (1995).
32. Majaj, N.J., Hong, H., Solomon, E.A. & DiCarlo, J.J. Simple learned weighted sums of inferior temporal neuronal firing rates accurately predict human core object recognition performance. *J. Neurosci.* **35**, 13402–13418 (2015).
33. Yamins, D.L.K. *et al.* Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl. Acad. Sci. USA* **111**, 8619–8624 (2014).
34. Cadieu, C.F. *et al.* Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLoS Comput. Biol.* **10**, e1003963 (2014).
35. Khaligh-Razavi, S.M. & Kriegeskorte, N. Deep supervised, but not unsupervised, models may explain IT cortical representation. *PLoS Comput. Biol.* **10**, e1003915 (2014).
36. Güçlü, U. & van Gerven, M.A. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J. Neurosci.* **35**, 10005–10014 (2015).
37. Yau, J.M., Pasupathy, A., Brincat, S.L. & Connor, C.E. Curvature processing dynamics in macaque area V4. *Cereb. Cortex* **23**, 198–209 (2013).
38. Freeman, J. & Simoncelli, E.P. Metamers of the ventral stream. *Nat. Neurosci.* **14**, 1195–1201 (2011).

39. Pasupathy, A. & Connor, C.E. Population coding of shape in area V4. *Nat. Neurosci.* **5**, 1332–1338 (2002).
40. Kell, A., Yamins, D., Norman-Haignere, S. & McDermott, J. Functional organization of auditory cortex revealed by neural networks optimized for auditory tasks. *Soc. Neurosci. Abstr.* 466.04 (2015).
41. Razavian, A.S., Azizpour, H., Sullivan, J. & Carlsson, S. CNN features off-the-shelf: an astounding baseline for recognition. in *Computer Vision and Pattern Recognition Workshops (CVPRW), IEEE Conference on*, 512–519 (IEEE, 2014).
42. Bottou, L. Large-scale machine learning with stochastic gradient descent. in *Proc. COMPSTAT 2010*, 177–186 (Springer, 2010).
43. Krizhevsky, A., Sutskever, I. & Hinton, G. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **25**, 1097–1105 (2012).
44. Choudhary, S. *et al.* Silicon neurons that compute. in *Artificial Neural Networks and Machine Learning–ICANN 2012*, 121–128 (Springer, 2012).
45. Snoek, J., Larochelle, H. & Adams, R.P. Practical bayesian optimization of machine learning algorithms. *Adv. Neural Inf. Process. Syst.* **26**, 2951–2959 (2012).
46. Bergstra, J., Yamins, D. & Cox, D. Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proc. 30th International Conference on Machine Learning* 115–123, http://jmlr.csail.mit.edu/proceedings/papers/v28/ (2013).
47. Griffin, G., Holub, A. & Perona, P. The Caltech-256 object category dataset. Caltech Technical Report, http://resolver.caltech.edu/CaltechAUTHORS:CNS-TR-2007-001 (2007).
48. Pinto, N., Cox, D.D. & DiCarlo, J.J. Why is real-world visual object recognition hard? *PLoS Comput. Biol.* **4**, e27 (2008).
49. Deng, J. *et al.* ImageNet: a large-scale hierarchical image database. in *CVPR 2009, IEEE Conference on Computer Vision and Pattern Recognition*, 248–288 (IEEE, 2009).
50. Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. Preprint at http://arxiv.org/abs/1409.1556 (2014).
51. Szegedy, C. *et al.* Going deeper with convolutions. Preprint at http://arxiv.org/abs/1409.4842 (2014).
52. Halevy, A., Norvig, P. & Pereira, F. The unreasonable effectiveness of data. *IEEE Intell. Syst.* **24**, 8–12 (2009).
53. Pillow, J.W. *et al.* Spatio-temporal correlations and visual signalling in a complete neuronal population. *Nature* **454**, 995–999 (2008).
54. Khorrami, P., Paine, T.L. & Huang, T.S. Do deep neural networks learn facial action units when doing expression recognition? Preprint at http://arxiv.org/abs/1510.02969 (2015).
55. Hinton, G.E., Dayan, P., Frey, B.J. & Neal, R.M. The "wake-sleep" algorithm for unsupervised neural networks. *Science* **268**, 1158–1161 (1995).
56. Zhu, L.L., Lin, C., Huang, H., Chen, Y. & Yuille, A. Unsupervised structure learning: hierarchical recursive composition, suspicious coincidence and competitive exclusion. in *Computer Vision–ECCV 2008*, 759–773 (Springer, 2008).
57. Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In *Unsupervised and Transfer Learning: Challenges in Machine Learning* Vol. 7 (eds. Guyon, I., Dror, G & Lemaire, V.) 29–41 (Microtome, 2013).
58. Mante, V., Sussillo, D., Shenoy, K.V. & Newsome, W.T. Context-dependent computation by recurrent dynamics in prefrontal cortex. *Nature* **503**, 78–84 (2013).
59. Stadie, B.C., Levine, S. & Abbeel, P. Incentivizing exploration in reinforcement learning with deep predictive models. Preprint at http://arxiv.org/abs/1507.00814 (2015).
60. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
61. Harvey, C.D., Coen, P. & Tank, D.W. Choice-specific sequences in parietal cortex during a virtual-navigation decision task. *Nature* **484**, 62–68 (2012).
62. Hulbert, J. & Norman, K. Neural differentiation tracks improved recall of competing memories following interleaved study and retrieval practice. *Cereb. Cortex* **25**, 3994–4008 (2015).
63. Hung, C.P., Kreiman, G., Poggio, T. & DiCarlo, J.J. Fast readout of object identity from macaque inferior temporal cortex. *Science* **310**, 863–866 (2005).
64. Rust, N.C. & Dicarlo, J.J. Selectivity and tolerance ("invariance") both increase as visual information propagates from cortical area V4 to IT. *J. Neurosci.* **30**, 12978–12995 (2010).
65. Freedman, D.J., Riesenhuber, M., Poggio, T. & Miller, E.K. Categorical representation of visual stimuli in the primate prefrontal cortex. *Science* **291**, 312–316 (2001).
66. Pagan, M., Urban, L.S., Wohl, M.P. & Rust, N.C. Signals in inferotemporal and perirhinal cortex suggest an untangling of visual target information. *Nat. Neurosci.* **16**, 1132–1139 (2013).
67. Marder, E. Understanding brains: details, intuition, and big data. *PLoS Biol.* **13**, e1002147 (2015).
68. Gatys, L.A., Ecker, A.S. & Bethge, M. A neural algorithm of artistic style Preprint at http://arxiv.org/abs/1508.06576 (2015).
69. Yamane, Y., Carlson, E.T., Bowman, K.C., Wang, Z. & Connor, C.E. A neural code for three-dimensional object shape in macaque inferotemporal cortex. *Nat. Neurosci.* **11**, 1352–1360 (2008).
70. Afraz, A., Boyden, E.S. & DiCarlo, J.J. Optogenetic and pharmacological suppression of spatial clusters of face neurons reveal their causal role in face gender discrimination. *Proc. Natl. Acad. Sci. USA* **112**, 6730–6735 (2015).
71. Marr, D., Poggio, T. & Ullman, S. *Vision: A Computational Investigation Into the Human Representation and Processing of Visual Information* (MIT Press, 2010).
72. Hoyle, G. The scope of neuroethology. *Behav. Brain Sci.* **7**, 367–381 (1984).
73. Szegedy, C. *et al.* Intriguing properties of neural networks. Preprint at http://arxiv.org/abs/1312.6199 (2013).
74. Goodfellow, I.J., Shlens, J. & Szegedy, C. Explaining and harnessing adversarial examples. Preprint at http://arxiv.org/abs/1412.6572 (2014).