Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing

Nikolaus Kriegeskorte

Medical Research Council Cognition and Brain Sciences Unit, University of Cambridge, Cambridge CB2 7EF, United Kingdom; email: nikolaus.kriegeskorte@mrc-cbu.cam.ac.uk

Annu. Rev. Vis. Sci. 2015. 1:417–46

The *Annual Review of Vision Science* is online at vision.annualreviews.org

This article's doi: 10.1146/annurev-vision-082114-035447

Keywords

biological vision, computer vision, object recognition, neural network, deep learning, artificial intelligence, computational neuroscience

Abstract

Recent advances in neural network modeling have enabled major strides in computer vision and other artificial intelligence applications. Human-level visual recognition abilities are coming within reach of artificial systems. Artificial neural networks are inspired by the brain, and their computations could be implemented in biological neurons. Convolutional feedforward networks, which now dominate computer vision, take further inspiration from the architecture of the primate visual hierarchy. However, the current models are designed with engineering goals, not to model brain computations. Nevertheless, initial studies comparing internal representations between these models and primate brains find surprisingly similar representational spaces. With human-level performance no longer out of reach, we are entering an exciting new era, in which we will be able to build biologically faithful feedforward and recurrent computational models of how biological brains perform high-level feats of intelligence, including vision.

INTRODUCTION

Unit: model abstraction of a neuron, typically computing a weighted sum of incoming signals, followed by a static nonlinear transformation

Backpropagation: a supervised neural network learning algorithm that efficiently computes error derivatives with respect to the weights by passing through the connectivity in reverse in order to iteratively minimize the error

The brain is a deep and complex recurrent neural network. The models of information processing that have dominated computational neuroscience, by contrast, are largely shallow architectures that perform simple computations. Unsurprisingly, complex tasks such as visual object recognition have remained beyond the reach of computational neuroscience (see the sidebar How Previous Attempts to Understand Complex Brain Information Processing Fell Short). In this article, I argue that recent advances in neural network models (LeCun et al. 2015) will usher in a new era of computational neuroscience, in which we will engage real-world tasks that require rich knowledge and complex computations.

Neural networks are an old idea, so what is new now? Indeed, the history of neural networks is roughly coextensive with that of modern computing machines (see the sidebar What Is Meant by the Term Neural Network?). John von Neumann and Alan Turing, whose ideas shaped modern computing technology, both explored network models inspired by the brain. An early mathematical model of a single neuron was suggested by McCulloch & Pitts (1943). Their binary threshold unit took a number of inputs, computed a weighted sum, and imposed a threshold, implementing a linear discriminant. Responding to a pattern of continuous inputs with a single binary output, the threshold unit provided an intuitive bridge between the biological hardware of a spiking neuron and categorization, a hallmark of cognition.

Discriminating categories that are not linearly separable in the input requires an intervening layer of nonlinear transformations between the input and the output units. The field took a while to find ways of automatically training such multilayer networks with input–output pairs. The most influential solution to this problem is the backpropagation algorithm, a gradient-descent method that makes iterative small adjustments to the weights in order to reduce the errors of the outputs (Werbos 1981, Rumelhart et al. 1986).

HOW PREVIOUS ATTEMPTS TO UNDERSTAND COMPLEX BRAIN INFORMATION PROCESSING FELL SHORT

The cognitive and brain sciences have gone through a sequence of transformations, with different fields dominating each period. Each field combined a different set of elements required for understanding how the brain works (**Table 1**). Cognitive psychology attempted to illuminate behaviorism's black box with theories of information processing. However, it lacked fully explicit computational models. Cognitive science made information processing theory fully explicit. However, it lacked constraints from neurophysiological data, making it difficult to adjudicate between multiple alternative models consistent with the behavioral data. Connectionism within cognitive science offered a neurobiologically plausible computational framework. However, neural network technology was not sufficiently advanced to take on real-world tasks such as object recognition from photographs. As a result, neural networks did not initially live up to their promise as AI systems, and in cognitive science, modeling was restricted to toy problems. Cognitive neuroscience brought neurophysiological data into investigations of complex brain information processing. Our hands full with the new challenges of analyzing complex brain imaging data, however, our theoretical sophistication slipped back to the stage of cognitive psychology, and we began (perhaps reasonably) by mapping box-and-arrow models onto brain regions. Computational neuroscience uses fully explicit and biologically plausible computational models to predict neurophysiological and behavioral data. At this level of rigor, however, we have not been able to engage complex real-world computational challenges and higher-level brain representations. Now deep neural networks provide a framework for engaging complex cognitive tasks and predicting both brain and behavioral responses.

Table 1 Historical progress toward understanding how the brain works

Elements required for understanding how the brain works		Behaviorism	Cognitive psychology	Cognitive science	Cognitive neuroscience	Classical computational neuroscience	Future cognitive computational neuroscience
Data	Behavioral	✓	✓	✓	✓	✓	✓
	Neurophysiological				✓	✓	✓
Theory	Cognitive		✓	✓	✓		✓
	Fully computationally explicit			✓		✓	✓
	Neurally plausible			✓		✓	✓
Explanation of real-world tasks requiring rich knowledge and complex computations			✓		✓		✓
Explanation of how high-level neuronal populations represent and compute							✓

Backpropagation led to a second wave of interest in neural networks in cognitive science and artificial intelligence (AI) in the 1980s. In cognitive science, neural network models of toy problems fostered the theoretical notion of parallel distributed processing (Rumelhart & McClelland 1988). However, backpropagation models did not work well on complex, real-world problems such as vision. Models not as obviously inspired by the brain that used hand-engineered representations and machine learning techniques, such as support vector machines, appeared to provide better engineering solutions for computer vision and AI. As a consequence, neural networks fell out of favor in the 1990s.

WHAT IS MEANT BY THE TERM NEURAL NETWORK?

The term neural network originally refers to a network of biological neurons. More broadly, the term evokes a particular paradigm for understanding brain function, in which neurons are the essential computational units, and computation is explained in terms of network interactions. Note that this paradigm leaves aside many biological complexities, including functional contributions of neurochemical diffusion processes, glial cells, and hemodynamics (Moore & Cao 2008). Although neurons are biological entities, the term neural network has come to be used as a shorthand for *artificial* neural network, a class of models of parallel information processing that is inspired by biological neural networks but commits to several further major simplifications.

Although spiking models have an important place in the computational literature, the models discussed here are nonspiking and do not capture dendritic computation, other processes within each neuron (e.g., Gallistel & King 2011), and distinct contributions from different types of neurons. The spatial structure of a neuron is typically abstracted from and its spiking output is modeled as a real number analogous to the spike rate. The rate is modeled as a weighted sum of incoming activations passed through a static nonlinearity. Despite, and perhaps also because of, these simplifications, the neural network paradigm provides one of the most important paths toward understanding brain information processing. It appears likely that this approach will take a central role in any comprehensive future brain theory. Opinions diverge as to whether more biologically detailed models will ultimately be needed. However, neural networks as used in engineering are certainly neurobiologically plausible, and their success in AI suggests that their abstractions may be desirable, enabling us to explain at least some complex feats of brain information processing.

Generative model: a model of the process that generated the data (e.g., the image) to be inverted in data analysis (e.g., visual recognition)

Feedforward network: a network with connections that form a directed acyclic graph, precluding recurrent information flow

Despite a period of disenchantment among the wider brain and computer science communities, neural network research has an unbroken history (Schmidhuber 2015) in theoretical neuroscience and in computer science. Throughout the 1990s and 2000s neural nets were studied by a smaller community of scientists who realized that the difficulties encountered were not fundamental limitations of the approach, but merely high hurdles to be overcome through a combination of better learning algorithms, better regularization, and larger training sets. With computations boosted by better computer hardware, the efforts of this community have been fruitful. In the past few years, neural networks have finally come into their own. They are currently conquering several domains of AI, including the hard problem of computer vision.

Computer vision competitions such as ImageNet (Deng et al. 2009) use secret test sets of images, providing rigorous evaluations of state-of-the-art technology. In 2012, a neural network model built by Krizhevsky et al. (2012) won the ImageNet classification competition by a large margin. The deep convolutional architecture of this model had enabled a leap in performance. Human performance levels, although still superior, suddenly did not seem entirely unattainable for computer vision any longer—at least in restricted domains such as visual object classification. The model built by Krizhevsky et al. (2012) marked the beginning of the dominance of neural networks in computer vision. In the past three years, error rates have dropped further, roughly matching human performance in the domain of visual object classification. Neural networks have also been very successful recently in other domains, such as speech recognition (Sak et al. 2014) and machine translation (Sutskever et al. 2014).

Artificial intelligence has entered an era in which systems directly inspired by the brain dominate practical applications. The time has come to bring this brain-inspired technology back to the brain. We are now in a position to integrate neural network theory with empirical systems neuroscience and to build models that engage the complexities of real-world tasks, use biologically plausible computational mechanisms, and predict neurophysiological and behavioral data.

The theoretical and engineering developments are progressing at an unprecedented pace. Many of the insights gained in engineering will likely be relevant for brain theory. Recent methods for comparing internal representations in neural population codes between models and brains enable us to test neural-net models as theories of brain information processing (Dumoulin & Wandell 2008; Kay et al. 2008; Kriegeskorte 2011; Kriegeskorte & Kievit 2013; Kriegeskorte et al. 2008a,b; Mitchell et al. 2008; Nili et al. 2014).

This article introduces a broad audience of vision and brain scientists to neural networks, including some of the recent advances of this modeling framework in engineering, and reviews the first few studies that have used such models to explain brain data. What emerges is a new framework for bringing computational neuroscience to high-level cortical representations and complex real-world tasks.

The following section, titled "A Primer on Neural Networks," introduces the basics of neural network models, including their learning algorithms and universal representational capacity. The section "Feedforward Neural Networks for Visual Object Recognition" describes the specific large-scale object recognition networks that currently dominate computer vision and discusses what these networks do and do not share with biological vision systems. The section titled "Early Studies Testing Deep Neural Nets as Models of Biological Brain Representations" reviews the first few studies to empirically compare internal representations between artificial neural networks and biological brains. The section titled "Recurrent Neural Networks for Vision" describes networks using recurrent computation. Recurrence is an essential component of biological brains, might implement inference on generative models of the formation of the input image, and represents a major frontier for computational neuroscience. Finally, the section titled "Conclusions" considers

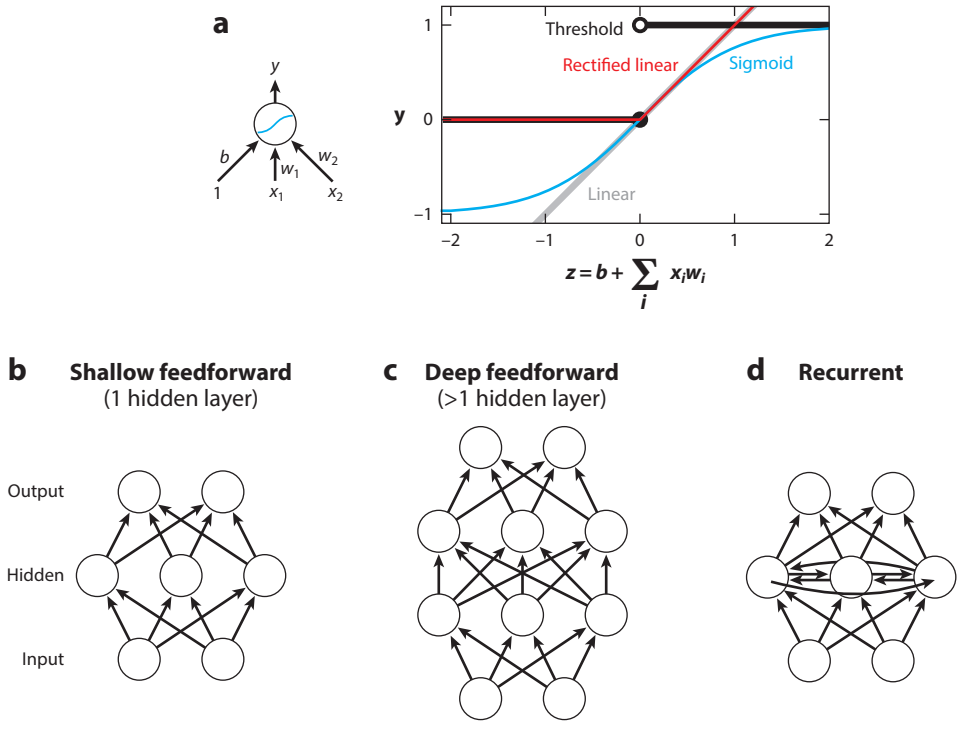

Figure 1

Artificial neural networks: basic units and architectures. (*a*) A typical model unit (*left*) computes a linear combination z of its inputs x_i using weights w_i and adding a bias b. The output y of the unit is a function of z, known as the activation function (*right*). Popular activation functions include linear (*gray*), threshold (*black*), sigmoid (hyperbolic tangent shown here, *blue*), and rectified linear (*red*) functions. A network is referred to as feedforward (*b*,*c*) when its directed connections do not form cycles and as recurrent (*d*) when they do form cycles. A shallow feedforward network (*b*) has zero or one hidden layers. Nonlinear activation functions in hidden units enable a shallow feedforward network to approximate any continuous function (with the precision depending on the number of hidden units). A deep feedforward network (*c*) has more than one hidden layer. Recurrent nets generate ongoing dynamics, lend themselves to the processing of temporal sequences of inputs, and can approximate any dynamical system (given a sufficient number of units).

critical arguments, upcoming challenges, and the way ahead toward empirically justified models of complex biological brain information processing.

A PRIMER ON NEURAL NETWORKS

A Unit Computes a Weighted Sum of Its Inputs and Activates According to a Nonlinear Function

We refer to model neurons as units to maintain a distinction between biological reality and highly abstracted models. The perhaps simplest model unit is a linear unit, which outputs a linear combination of its inputs (**Figure 1*a***). Such units, combined to form networks, can never transcend linear combinations of the inputs. This insight is illustrated in **Figure 2*b***, which shows how an output unit that linearly combines intermediate-layer linear-unit activations just adds up

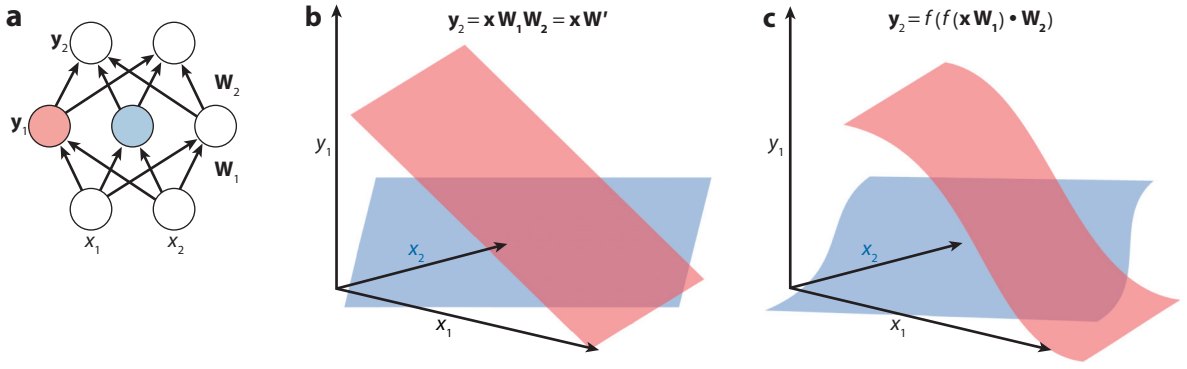

Figure 2

Networks with nonlinear hidden units can approximate arbitrary nonlinear functions. (*a*) A feedforward neural network with a single hidden layer. (*b*) Activation of the pink and blue hidden units as a function of the input pattern (x_1, x_2) when the hidden units have linear activation functions. Each output unit (y_2) will compute a weighted combination of the ramp-shaped (i.e., linear) activations of the hidden units. Thus, the output remains a linear combination of the input pattern. A linear hidden layer is not useful because the resulting network is equivalent to a linear network without a hidden layer intervening between input and output. (*c*) Activation of the pink and blue hidden units when these have sigmoid activation functions. Arbitrary continuous functions can be approximated in the output units (y_2) by weighted combinations of a sufficient number of nonlinear hidden-unit outputs (y_1).

ramp functions, and thus itself computes a ramp function. A multilayer network of linear units is equivalent to a single-layer network whose weights matrix $\mathbf{W}'$ is the product of the weights matrices $\mathbf{W}_i$ of the multilayer network. Nonlinear units are essential because their outputs provide building blocks (**Figure 2*c***) whose linear combination one level up enables us to approximate any desired mapping from inputs to outputs, as described in the next section.

A unit in a neural network uses its input weights $\mathbf{w}$ to compute a weighted sum z of its input activities $\mathbf{x}$ and passes the result through a (typically monotonic) nonlinear function f to generate its activation y (**Figure 1*a***). In early models, the nonlinearity was simply a step function (McCulloch & Pitts 1943, Rosenblatt 1958, Minsky & Papert 1972), making each unit a linear discriminant imposing a binary threshold. For a single threshold unit, the perceptron learning algorithm provides a method for iteratively adjusting the weights (starting with zeros or random weights) so as to get as many training input–output pairs as possible right. However, hard thresholding entails that, for a given pair of an input pattern and a desired output pattern, small changes to the weights will often make no difference to the output. This makes it difficult to learn the weights for a multilayer network by gradient descent, where small adjustments to the weights are made to iteratively reduce the errors. If the hard threshold is replaced by a soft threshold that continuously varies, such as a sigmoid function, gradient descent can be used for learning.

Universal function approximator: model family that can approximate any function that maps input patterns to output patterns (with arbitrary precision when allowed enough parameters)

Networks with Nonlinear Hidden Units Are Universal Function Approximators

The particular shape of the nonlinear activation function does not matter to the class of input–output mappings that can be represented. Feedforward networks with at least one layer of hidden units intervening between input and output layers are universal function approximators: Given a sufficient number of hidden units, a network can approximate any function of the inputs in the output units. Continuous functions can be approximated with arbitrary precision by adding a sufficient number of hidden units and suitably setting the weights (Schäfer & Zimmermann 2007, Hornik 1991, Cybenko 1989). **Figure 2*c*** illustrates this process for two-dimensional inputs:

Adding up a sufficient number of sigmoid ramps, which can have any orientation, slope, and position, we can approximate any continuous function of the input.

To gain an intuition on why combining sigmoids (or any step-like functions) of the input enables a network to approximate any function, imagine we set the weights of a set of units so that they compute sigmoids whose plateaus (close to 1) overlap only in a particular region of the input space. If we now sum the outputs of these units in a unit with a high threshold, that unit can indicate (by an output close to 1) that we are in a certain region of the input space. If we build indicators in this fashion for all regions within the input space that require a different output, we can map any input to any required output approximately. The precision of this approximate mapping can always be improved by using more units to define more separate regions with indicators. Note that if the activation function is continuous (as it usually is), then the function represented by the network is also continuous. The network would use two hidden layers to represent what is essentially a lookup table of the training input–output pairs. (However, the network would have the nice feature of handling novel inputs by interpolation or extrapolation.) The cited theoretical results on the universality of feedforward nets go beyond this intuitive explanation and show that only a single hidden layer is needed to approximate any function and that the activation function need not resemble a step function.

A simple and powerful neural network architecture is the feedforward network (**Figure 1*b*,*c***). A feedforward network is composed of a sequence of layers of units, with each unit sending its output only to units in higher layers. Thus, the units and connections of a feedforward network correspond to the nodes and edges, respectively, of a directed acyclic graph. In computer vision systems, units often receive inputs only from the immediately preceding layer. In addition, inputs in lower layers are usually restricted to local receptive fields, inspired by the visual hierarchy.

Modern models use a variety of nonlinear activation functions, including sigmoid (e.g., logistic or hyperbolic tangent) and rectified linear functions (**Figure 1*a***). A rectified linear unit outputs the linear combination it computes, if it is positive, and 0 otherwise. Rectified linear units simplify the gradient-descent learning of the weights, enabling more rapid training, and have been demonstrated to work very well in computer vision and other domains.

Deep learning: machine learning of complex representations in a deep neural network, typically using stochastic gradient descent by error backpropagation

Deep neural network: network with more than one hidden layer between the input and output layers; more loosely, a network with many hidden layers

Why Deep?

A feedforward network is said to be *deep* when it has more than one hidden layer. This technical definition notwithstanding, the term deep is also used in a graded sense. A deep net, thus, is a network with many layers, and one network can be deeper than another. *Deep learning* refers to the strategy of using architectures with many hidden layers to tackle difficult problems, including vision.

Why does depth help? We saw above that even shallow networks with a single layer of nonlinear hidden units are universal function approximators. Shallow networks are closely related to support vector machines, which can likewise learn arbitrary nonlinear functions, can be more efficiently trained than neural networks, and have been very successful tools of machine learning. The reason depth matters is that deep nets can represent many complex functions more concisely (i.e., with fewer units and weights) than shallow nets and support vector machines (Bengio 2009).

Consider a shallow network (i.e., a network with a single hidden layer) that computes some function. We can create a deeper network with the same number of units by distributing the units of the single hidden layer across multiple hidden layers in the new network. The deep network could have the same connectivity from the input to the hidden units and from the hidden units to the output. It can thus compute any function the shallow network can compute. The reverse is not true, however: The deep network is permitted additional nonzero weights from any given layer to higher layers, enabling *reuse* of the results of previous computations and extending the expressive

power of the deep network. For many particular functions that a deep network might compute, one can show that a shallow network would need a much larger number of units (Bengio 2009).

It is instructive to consider the analogy to modern computing hardware. The von Neumann architecture is a fundamentally sequential model of computation that enables the reuse of results of previous computations. In special cases, in which many computations are to be performed independently (e.g., across image positions in graphics and vision), parallel hardware can speed up the process. Whereas independent computations can be performed either in parallel or sequentially, however, dependent computations can only be performed sequentially. The option to reuse previous results therefore extends the set of computable functions (if the total number of units is fixed).

In essence, a shallow network is a universal function approximator because it can piece together the target function like a lookup table. Many functions can be more concisely represented using a deeper network, however, taking advantage of redundancies and exploiting the inherent structure of the target function. Although every problem is different and the field is still learning when exactly depth helps, the practical success of deep learning in AI suggests that many real-world problems, including vision, may be more efficiently solved with deep architectures. Interestingly, the visual hierarchy of primate brains is also a deep architecture.

Recurrent network: a network with recurrent information flow, which produces dynamics and lends itself naturally to the perception and generation of spatiotemporal patterns

Universal approximator of dynamical systems: a model family generating dynamics that can approximate any dynamical system (with arbitrary precision when allowed enough parameters)

Supervised learning: a learning process requiring input patterns and additional information about the desired representation or the outputs (e.g., category labels)

Recurrent Neural Networks Are Universal Approximators of Dynamical Systems

Feedforward networks compute static functions. An architecture with more interesting dynamics is a recurrent network, whose units can be connected in cycles. Such an architecture is more similar to biological neuronal networks, in which lateral and feedback connections are ubiquitous. The notion of separate hidden layers is meaningless in a recurrent network because every hidden unit can interact with every other hidden unit. Recurrent nets are therefore often depicted as a single interconnected set of hidden units, with separate sets of input and output units (**Figure 1*d***). A layered architecture is a special case of a recurrent network in which certain connections are missing (i.e., their weights are fixed at 0).

In visual neuroscience, the theoretical concept of the visual hierarchy is based on a division of the connections into feedforward, lateral, and feedback connections, as identified by a connection's cortical layers of origin and termination, as well as on the fact that some neurons are separated from the input by many synapses and tend to represent more complex visual features. Although these criteria may not support a perfectly unambiguous assignment of ranks that would define a hierarchy for the primate visual system (Hilgetag et al. 2000), the hierarchical model continues to be a useful simplification.

Whereas a feedforward network computes a static function that maps inputs to outputs, a recurrent network produces dynamics: a temporal evolution of states that can be influenced by a temporal evolution of input patterns. The internal state of a recurrent network lends it a memory, enabling it to represent the recent stimulus history and detect temporal patterns. Whereas feedforward nets are universal function approximators, recurrent nets are universal approximators of dynamical systems (Schäfer & Zimmermann 2007). A variety of particular models have been explored by simulation and analytically.

In an echo-state network (Jaeger 2001, see also Maass et al. 2002, for a similar model with spiking dynamics), for example, the sequence of input patterns is fed into a set of hidden units that are sparsely and randomly connected. The wave of activity associated with each input pattern will reverberate among the hidden units for a while until it comes to be dominated by the effects of subsequent input patterns. Like concentric waves on the surface of a pond that enable us to infer an event at their center sometime in the past, the activity of the hidden units encodes information about the recent stimulus history. In echo-state networks, the weights among the hidden units are not trained (although their random setting requires some care to ensure that the memories do

not fade too quickly). Supervised learning is used to train a set of readout units to detect temporal patterns in the input.

Echo-state networks rely on random weights among the hidden units for their recurrent dynamics. Alternatively, the dynamics of a recurrent network can be explicitly learned through supervision, so as to optimize it to produce, classify, or predict temporal patterns (Graves & Schmidhuber 2009, Sutskever et al. 2014).

Representations Can Be Learned by Gradient Descent Using the Backpropagation Algorithm

The universality theorems assure us of the representational power of neural networks with sufficient numbers of units. However, these theorems do not tell us how to set the weights of the connections, so as to represent a particular function with a feedforward net or a particular dynamical system with a recurrent net. Learning poses a high-dimensional and difficult optimization problem. Models that can solve real-world problems will have large numbers of units and even larger numbers of weights. Global optimization techniques are not available for this nonconvex problem. The space of weight settings is so vast that simple search algorithms (for example, evolutionary algorithms) can cover only a vanishingly small subset of the possibilities and typically do not yield working solutions, except for small models restricted to toy problems.

The high dimensionality of the weight space makes global optimization intractable. However, the space contains many equivalent solutions (consider, for example, exchanging all incoming and outgoing weights between two neurons). Moreover, the total error (i.e., the sum of squared deviations between actual and desired outputs) is a locally smooth function of the weights. The current training method of choice is gradient descent, the iterative reduction of the errors through small adjustments to the weights.

The basic idea of gradient-descent learning is to start with a random initialization of the weights and to determine how much a slight change to each weight will reduce the error. The weight is then adjusted in proportion to the effect on the error. This method ensures that we move in the direction in weight space, along which the error descends most steeply.

The gradient, that is, how much the error changes with an adjustment of a weight, is the derivative of the error with respect to the weight. These derivatives can be computed easily for the weights connecting to the output layer of a feedforward network. For connections driving the preceding layers, an efficient way to compute the error derivatives is to propagate them backward through the network. This gives the method its name, backpropagation (Werbos 1981, Rumelhart et al. 1986).

Gradient descent sees only the local neighborhood in weight space and is not guaranteed to find globally optimal solutions. It can nevertheless find solutions that work very well in practice. The high dimensionality of the weight space is a curse in that it makes global optimization difficult. However, it is a blessing in that it helps local optimization find good solutions: With so many directions to move in, gradient descent is unlikely to get stuck in local minima, where the error increases in all directions and no further progress is possible.

Intriguingly, the same approach can be used to train recurrent networks. The error derivatives are then computed by backpropagation through time, with the process suffusing the loops in reverse through multiple cycles. To understand why this works, we can construe any recurrent network as the feedforward network obtained by replicating all units of the recurrent network along the dimension of time (for a sufficiently large number of time steps). Each time point of the recurrent computation corresponds to a layer of the feedforward net, each of which is connected to the next by the same weights matrix, the weights matrix of the recurrent network. By backpropagation through time, a recurrent network can learn weights that enable it to store short-term memories

in its dynamics, relating temporally separated events as needed to achieve desired classifications or predictions of temporal sequences. However, propagating error derivatives far enough backward through time for the network to learn how to exploit long-lag dependencies is hampered by the problem that the gradients tend to vanish or explode (Hochreiter 1991, Hochreiter et al. 2001). The problem occurs because a given weight's error derivative is the product of multiple terms, corresponding to weights and derivatives of the activation functions encountered along the path of backpropagation. One solution to this problem is offered by the long short-term memory (LSTM) architecture (Hochreiter & Schmidhuber 1997), in which special gated units can store short-term memories for extended periods. The error derivatives backpropagated through these units remain stable, enabling backpropagation to learn long-lag dependencies. Such networks, amazingly, can learn to remember information that will be relevant many time steps later in a sequential prediction, classification, or control task. Backpropagation adjusts the weights to ingrain a dynamics that selectively stores (in the activation state) the information needed later to perform the task.

Unsupervised learning: a learning process that requires only a set of input patterns and captures aspects of their probability distribution

Vanishing and exploding gradients can also pose a problem in training deep feedforward nets with backpropagation. The choice of nonlinear activation function can make a difference with regard to this problem. In addition, the details of the gradient-descent algorithm, regularization, and weight initialization all matter to making supervised learning by backpropagation work well.

Representations Can Also Be Learned with Unsupervised Techniques

In supervised learning, the training data comprise both input patterns and the associated desired outputs. An explicit supervision signal of this type is often unavailable in the real world. Biological organisms do not in general have access to supervision. In engineering, similarly, we often have a large number of unlabeled input patterns and only a smaller number of labeled input patterns (e.g., images from the web). Unsupervised learning does not require labels for a network to learn a representation that is optimized for natural input patterns and potentially useful for a variety of tasks. Natural images, for example, form a very small subset of all possible images, enabling unsupervised learning to find compressed representations.

An instructive example of unsupervised learning is provided by autoencoders (Hinton & Salakhutdinov 2006). An autoencoder is a feedforward neural network with a central code layer that has fewer units than the input. The network is trained with backpropagation to reconstruct its input in the output layer (which has the same number of units as the input layer). Although the learning algorithm is backpropagation and uses a supervision signal, the technique is unsupervised because it requires no separate supervision information (i.e., no labels), only the set of input patterns. If all layers, including the code layer, had the same dimensionality as the input, the network could just pass the input through its layers. Because the code layer has fewer units, however, it forms an informational bottleneck. To reconstruct the input, the network must learn to retain sufficient information about the input in its small code layer. An autoencoder therefore learns a compressed representation in its code layer, exploiting the statistical structure of the input domain. This representation will be specialized for the distribution of the input patterns used in training.

The layers from the input to the code layer are called the *encoder*, and the layers from the code layer to the output are called the *decoder*. If the encoder and decoder are linear, the network learns the linear subspace spanned by the first k principal components (for a code layer of k units). With nonlinear neural networks as encoders and decoders, nonlinearly compressed representations can be learned. Nonlinear codes can be substantially more efficient when the natural distribution of the input patterns is not well represented by a linear subspace. Natural images are a case in point.

Unsupervised learning can help pretrain a feedforward network when insufficient labeled training data are available for purely supervised learning. For example, a network for visual recognition

can be pretrained layer by layer in the autoencoder framework using a large set of unlabeled images. Once the network has learned a reasonable representation of natural images, it can more easily be trained with backpropagation to predict the correct image labels.

FEEDFORWARD NEURAL NETWORKS FOR VISUAL OBJECT RECOGNITION

Computer vision has recently come to be dominated by a particular type of deep neural network: the deep feedforward convolutional network. These networks now robustly outperform the previous state of the art, which consisted in hand-engineered visual features (e.g., Lowe 1999) forming the input to shallow machine learning classifiers such as support vector machines. Interestingly, some of the earlier systems inserted an intermediate representation, often acquired by unsupervised learning, between the hand-engineered features and the supervised classifier. The insertion of this representation might have helped address the need for a deeper architecture.

The deep convolutional nets widely used computer vision today share several architectural features, some of which are loosely inspired by biological vision systems (Hubel & Wiesel 1968).

- **Deep hierarchy**: Like the primate ventral visual stream, these networks process information through a deep hierarchy of representations (typically 5 to 20 layers; see **Figure 3** for an example), gradually transforming a visual representation, whose spatial layout matches the image, to a semantic representation that enables the recognition of object categories.

Convolutional network: network in which the preactivation of a layer (before the nonlinearity) implements convolutions of the previous layer with a number of weight-template patterns

Receptive field modeling: predictive modeling of the response to arbitrary sensory inputs of neurons (or measured channels of brain activity)

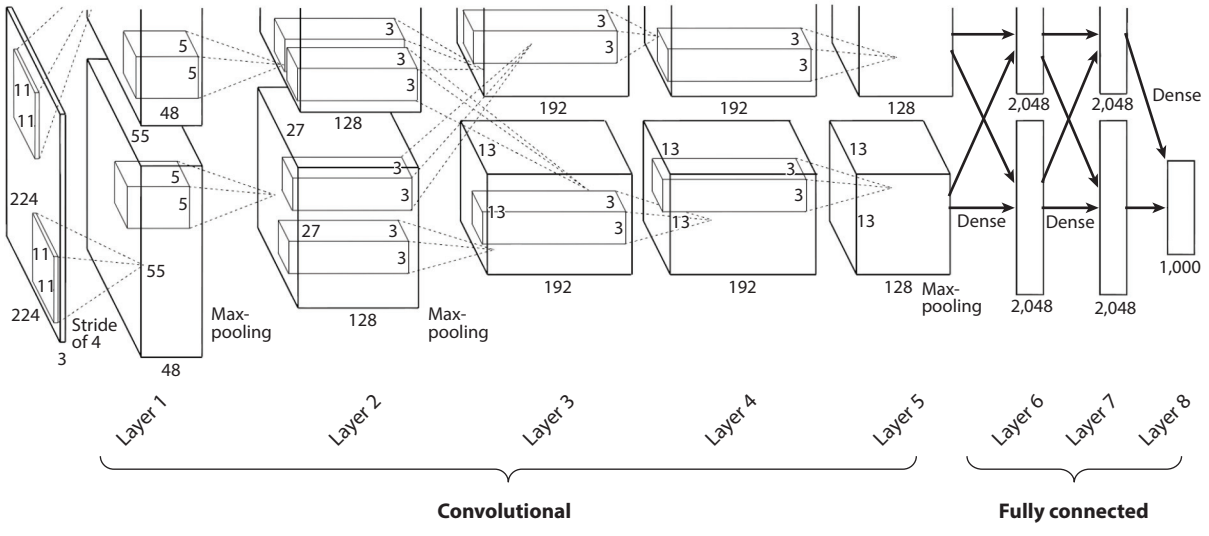

Figure 3

Deep convolutional feedforward architecture for object recognition. The figure shows the architecture used by Krizhevsky et al. (2012). The information flows from the input pixel image (*left*) (224 × 224 pixels, 3 color channels) through 7 hidden layers to the category output (*right*) (1,000 category detector units). The large boxes represent stacks of feature maps. For layer 2, for example, the lower large box represents 128 feature maps of size 27 (horizontal image positions) × 27 (vertical image positions). Note that the dimensions of the boxes are not drawn to scale. The small boxes represent the feature templates that are convolved with the representation in a given layer. Because convolution and max-pooling operate at strides greater than 1 pixel, the spatial extent of the feature maps decreases along the sequence of representations (224, 55, 27, 13, 13, 13, 1, 1, 1). The upper and lower large boxes represent the division of labor between two graphics processing units.

Max-pooling: summary operation implementing invariances by retaining only the maxima of sets of detectors differing in irrelevant properties (e.g., local position)

- **Convolution**: The lower layers contain local visual feature detectors with small receptive fields (RFs). Each detector is replicated all over the two-dimensional image, forming a feature map. This amounts to a convolution of the image with each feature pattern, followed by a static nonlinearity. The convolutional architecture is motivated by the insight that a feature useful in one position is likely to also be useful in another position. This architecture resembles early primate visual areas, which also detect qualitatively similar local visual features in many visual field positions (although the feature characteristics and the spatial distributions of the RFs are not completely uniform as in convolutional networks). The RFs of the units increase in size along the hierarchy. The restriction of the connections to a local region and the replication of the connection weights across spatial positions (same weight pattern at all locations for a given feature) greatly reduce the number of parameters that need to be learned (LeCun et al. 1989).
- **Local pooling and subsampling**: In between the convolutional stages, local pooling stages are inserted. Pooling combines the outputs of a local set of units by taking the maximum or the average. This confers a local tolerance to spatial shifts of the features, making the representation robust to small image shifts and small distortions of the configuration of image features (Fukushima 1980). Max-pooling is also used in neuroscientific vision models such as HMAX (Riesenhuber & Poggio 1999, Serre et al. 2007) to implement local tolerances. Pooling is often combined with subsampling of the spatial locations. The reduction in the number of distinct spatial locations represented frees up resources for an increase along the hierarchy in the number of distinct features computed at each location.

In the highest layers, units have global RFs, receiving inputs from all units of the previous layer. The final layer typically contains one unit per category and implements a softmax (normalized exponential) function, which strongly reduces all but the very highest responses and ensures that the outputs add up to 1. The output can be interpreted as a probability distribution over the categories when the training procedure is set up to minimize the crossentropy error.

The networks can be trained to recognize the category of the input image using backpropagation (LeCun et al. 1989, LeCun & Bengio 1995). When a network is trained to categorize natural images, the learning process discovers features that are qualitatively similar to those found in biological visual systems (**Figure 4**). The early layers develop Gabor-like features, similar to those that characterize V1 neurons. Similar features are discovered by unsupervised techniques such as sparse representational learning (Olshausen & Field 1997), suggesting that they provide a good starting point for vision, whether the goal is sparse representation or categorization. Subsequent stages contain units that are selective for slightly more complex features, including curve segments. Higher layers contain units that are selective for parts of objects and for entire objects, such as faces and bodies of humans and animals, and inanimate objects such as cars and buildings.

To understand what has been learned automatically, the field is beginning to devise methods for visualizing the RFs and selectivities of units within deep networks (Zeiler & Fergus 2014, Girshick et al. 2014, Simonyan et al. 2014, Tsai & Cox 2015, Zhou et al. 2015). **Figure 4** shows such visualizations, which support the idea that units learn selectivities to natural image features that increase in visual complexity along the hierarchy. However, two important caveats accompany such visualizations. First, because of the multiple nonlinear transforms across layers, a unit cannot be accurately characterized by an image template. If the high-level responses could be computed by template matching, a deep hierarchy would not be needed for vision. The visualizations merely show what drives the response in the context of a particular image. To get an idea of the selectivity of a unit, many images that drive it need to be considered (for multiple templates for each of a larger number of units, see Zeiler & Fergus 2014). Second, the units visualized in **Figure 4** have been selected because they confirm a theoretical bias for interpretable selectivities. Units similar

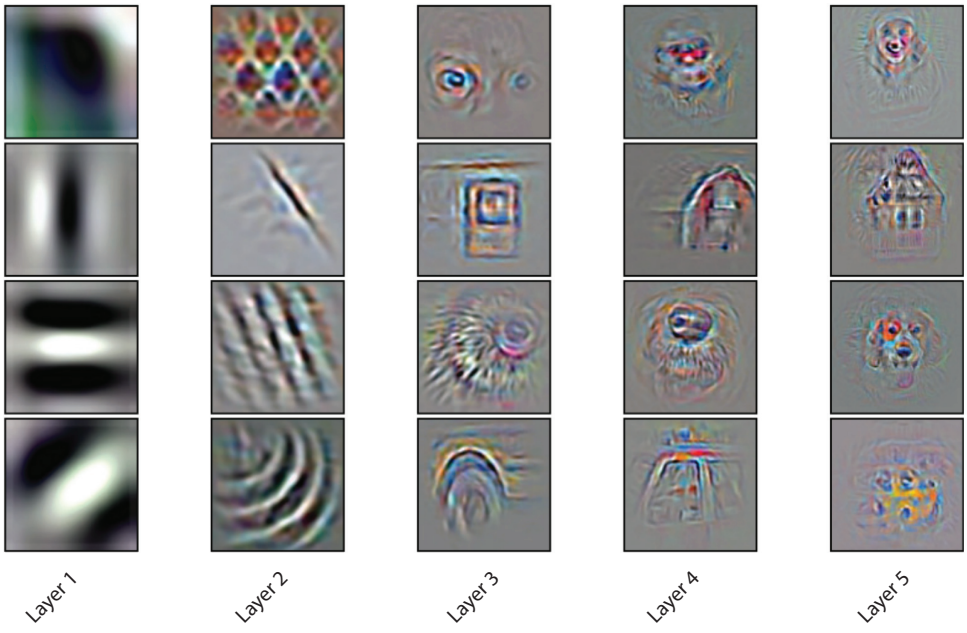

Figure 4

Deep supervised learning produces feature selectivities that are qualitatively consistent with neurophysiological findings. To understand representations in deep neural networks, we can visualize which image elements drive a given unit in a deep network. For 20 example units (4 from each of 5 layers), the images shown visualize what caused the response in the context of a particular image that strongly drove the unit. The visualization technique used here involves two steps: selection of an input image that strongly drives the unit, and inversion of the feedforward computation to generate the image element responsible. Convolutions along the feedforward pass are inverted by deconvolution (using the transposes of the convolution matrices). Max-pooling operations are inverted by storing the identity of the connection to the pooling unit that was maximally active in the feedforward pass. Note that a unit deep in a network does not perform a simple template-matching operation on the image and therefore cannot be fully characterized by any visual template. However, performing the above visualization for many images that drive a unit (not shown) can help us understand its selectivity and tolerances. The deconvolutional visualization technique shown was developed by Zeiler & Fergus (2014). The deep network is from Chatfield et al. (2014). The analysis was performed by Güçlü & van Gerven (2015). Figure adapted with permission from Güçlü & van Gerven (2015).

to those shown may be the exception rather than the rule, and it is unclear whether they are essential to the functionality of the network. For example, meaningful selectivities could reside in linear combinations of units rather than in single units, with weak distributed activities encoding essential information.

The representational hierarchy appears to gradually transform a space-based visual to a shape-based and semantic representation. The network acquires complex knowledge about the kinds of shapes associated with each category. In this context, shape refers to luminance- and color-defined features of various levels of complexity. High-level units appear to learn representations of shapes occurring in natural images, such as faces, human bodies, animals, natural scenes, buildings, and cars. The selectivities learned are not restricted to the categories detected by the output layer, but may include selectivities to parts of these objects or even to context elements. For example, the network by Krizhevsky et al. (2012) contains units that appear to be selective for text (Yosinski et al.

2015) and faces, although text and faces were not among the trained categories. Presumably, those responses help detect the categories represented in the output layer, because they are statistically related to the categories to be detected. For example, part-selective features may serve as stepping stones toward detection of entire objects (Jozwik et al. 2015). A verbal functional interpretation of a unit, e.g., as an eye or a face detector, may help our intuitive understanding and capture something important. However, such verbal interpretations may overstate the degree of categoricality and localization, and understate the statistical and distributed nature of these representations.

An influential example of a deep convolutional neural network for computer vision is the system built by Krizhevsky et al. (2012). The architecture (**Figure 3**) comprises five convolutional layers and three fully connected layers. The authors found that reducing the number of convolutional layers hurt performance, illustrating the need for a deep architecture. The system uses rectified linear units, max-pooling, and local normalization. The network was trained by backpropagation to recognize which of 1,000 object categories was shown in the input image. The training set comprised 1.2 million category-labeled images from the ImageNet set. This set was expanded by a factor of 2,048 by adding translated and horizontally reflected versions of the images. The training cycled through the resulting image set 90 times.

The training relied on dropout regularization (Hinton et al. 2012), a technique in which each unit is "dropped" (omitted from the computations) with a probability of 0.5 on each training trial. Thus, on a given trial, a random set of approximately half of the units is used in both the forward pass computing the output and the backpropagation pass adjusting the weights. This method prevents complex coadaptations of the units during learning, forcing each unit to make a useful contribution in the context of many different teams of other units. The network has a total of 650,000 units and 60 million parameters. The convolutional layers are defined by their small local weight templates, which constitute less than 5% of the parameters in total. Over 95% of the parameters define the upper three fully connected layers. Dropout was applied to the first two fully connected layers, each of which has many millions of incoming connections. Experiments showed that dropout was necessary to prevent overfitting.

The training was performed over the course of six days on a single workstation with two graphics processing units (GPUs), which parallelize and greatly accelerate the computations. The system was tested on a held-out set of images in the ImageNet Large-Scale Visual Recognition Challenge 2012, a computer-vision competition. It won the competition, beating the second-best system by a large margin and marking the beginning of the dominance of neural networks in computer vision. Since then, several convolutional neural networks using similar architectures have further improved performance (e.g., Zeiler & Fergus 2014, Chatfield et al. 2014).

The deep convolutional neural networks used in computer vision perform limited aspects of vision, such as category-level recognition. However, the range of visual tasks tackled is quickly expanding, and deep networks do represent a quantum leap compared with the earlier computer vision systems. Deep convolutional networks are not designed to closely parallel biological vision systems. However, their essential functional mechanisms are inspired by biological brains and could plausibly be implemented with biological neurons. This new technology provides an exciting framework for more biologically faithful brain-computational models that perform complex feats of intelligence beyond the current reach of computational neuroscience.

Normalization: an operation (e.g., division) applied to a set of activations so as to hold fixed a summary statistic (e.g., the sum)

Dropout: a regularization method for neural network training in which each unit is omitted from the architecture with probability 0.5 on each training trial

Graphics processing unit (GPU): specialized computer hardware developed for graphics computations that greatly accelerates matrix–matrix multiplications and is essential for efficient deep learning

EARLY STUDIES TESTING DEEP NEURAL NETS AS MODELS OF BIOLOGICAL BRAIN REPRESENTATIONS

Several studies have begun to assess deep convolutional neural networks as models for biological vision, comparing both the internal representational spaces and performance levels between

models and brains. One finding that is replicated and generalized across several studies (Yamins et al. 2013, 2014; Khaligh-Razavi & Kriegeskorte 2013, 2014) is that models that utilize representational spaces that are more similar to those of the inferior temporal (IT) cortex (Tanaka 1996) in human and nonhuman primates tend to perform better at object recognition. This observation affirms the intuition that computer vision can learn from biological vision. Conversely, biological vision science can look to engineering for candidate computational theories.

Representational similarity analysis: method for testing computational models of brain information processing through statistical comparisons of representational distance matrices that characterize population-code representations

It is not true in general that engineering solutions closely follow biological solutions (consider planes, trains, and automobiles). In computer vision in particular, early failures to scale neural network models to real-world vision fostered a sense that seeking more brain-like solutions was fruitless. However, the recent successes of neural network models suggest that brain-inspired architectures for vision are extremely powerful. The empirical comparisons between representations in computer vision systems and brains discussed in this section suggest that the neural network models do not merely have architectural similarities. They also learn representations very similar to those of the primate ventral visual pathway.

It is impossible to prove that representations have to be similar to biological brains to support successful computer vision. However, an association between performance at object recognition and representational similarity to IT has been shown for a large set of automatically generated neural network architectures using random features (Yamins et al. 2013, 2014), for a wide range of popular hand-engineered computer vision features and neuroscientific vision models (Khaligh-Razavi & Kriegeskorte 2013, 2014), and for the layers of a deep neural network (Khaligh-Razavi & Kriegeskorte 2014). Within the architectures explored so far, at least, it appears that performance optimization leads to representational spaces similar to IT.

IT is known to emphasize categorical divisions in its representation (Kriegeskorte et al. 2008b). Models that perform well at categorization (which is implemented by linear readout) similarly tend to have stronger categorical divisions. This partially explains their greater representational similarity to IT. However, even the within-category representational geometries tend to be more similar to IT in the better-performing models (Khaligh-Razavi & Kriegeskorte 2014).

The best performing models are deep neural networks, and these networks are also best at explaining the IT representational geometry (Khaligh-Razavi & Kriegeskorte 2014, Cadieu et al. 2014). Khaligh-Razavi & Kriegeskorte (2014) tested a wide range of classical computer vision features; several neuroscientifically motivated vision models, including VisNet (Wallis & Rolls 1997, Tromans et al. 2011) and HMAX (Riesenhuber & Poggio 1999); and the deep neural network built by Krizhevsky et al. (2012) (**Figure 3**). The brain representations in their study were estimated from human functional magnetic resonance imaging (fMRI) and monkey cell recordings (monkey data from Kiani et al. 2007, Kriegeskorte et al. 2008b). Khaligh-Razavi & Kriegeskorte (2014) compared the internal representational spaces between models and brain regions using representational similarity analysis (Kriegeskorte et al. 2008a). For each pair of stimuli, this analysis measures the dissimilarity of the two stimuli in the representation. The vector of representational dissimilarities across all stimulus pairs is then compared between a model representation and a brain region.

Early layers of the deep neural network had representations resembling early visual cortex. Across the layers of the network, the representational geometry became monotonically less similar to early visual cortex and more similar to IT. These results are shown in **Figure 5** for human data. Similar results were obtained for monkey IT (not shown here).

At the highest layer, the representation did not yet fully explain the explainable variance in the IT data. However, a representation fitted to IT (by linear remixing and reweighting of the features of the deep neural network using independent image sets for training and validation) fully explained the IT data (Khaligh-Razavi & Kriegeskorte 2014). This IT-fitted deep neural

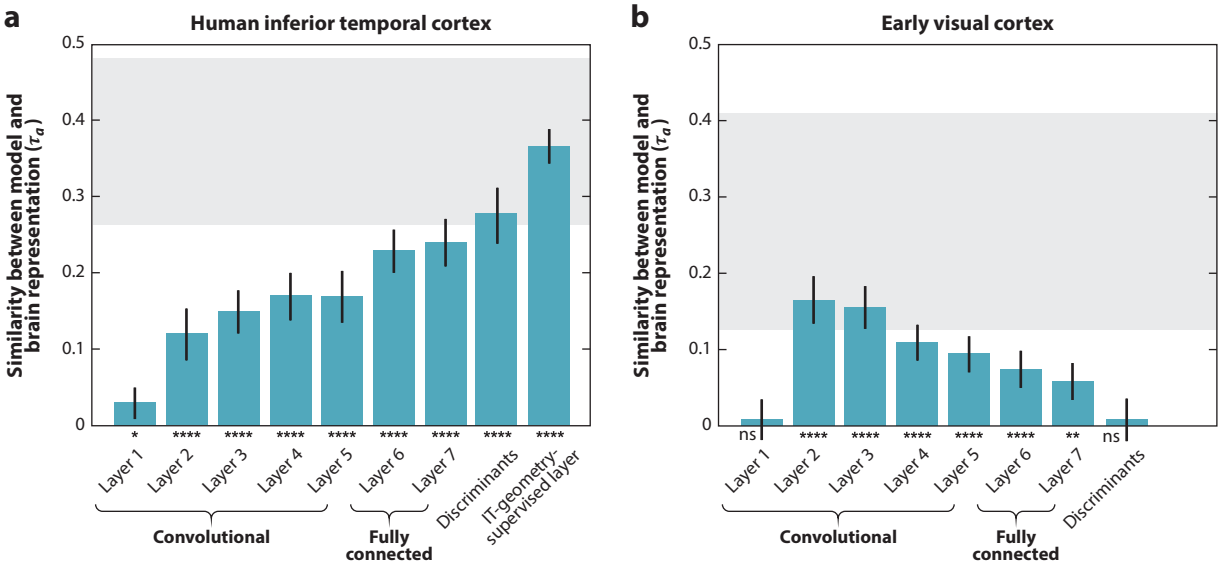

Figure 5

Deep neural network explains early visual and inferior temporal representations of object images. Each representation in model and brain was characterized by the dissimilarity matrix of the response patterns elicited by a set of real-world photos of objects. (*a*) Representations become monotonically more similar to those of human inferior temporal (IT) cortex as we ascend the layers of the Krizhevsky et al. (2012) neural network. When the final representational stages are linearly remixed to emphasize the same semantic dimensions as IT using linear category discriminants (*second bar from the right*), and when each layer and each discriminant are assigned a weight to model the prevalence of different computational features in IT (cross-validated to avoid overfitting to the image set; *rightmost bar*), the noise ceiling (*gray shaded region*) is reached, indicating that the model fully explains the data. When the same method of linear combination with category discriminants and weighting was applied to traditional computer vision features (not shown here), the representation did not explain the IT data. Similar results were obtained for monkey IT (not shown here). (*b*) Lower layers of the deep neural network resemble the representations in the foveal confluence of early visual areas (V1–V3). Asterisks indicate accuracy above chance as follows: ns, not significant; *, $p < 0.05$; **, $p < 0.01$; ***, $p < 0.001$; ****, $p < 0.0001$. The similarity between each model representation and IT (vertical axes) was measured using Kendall's rank correlation coefficient τ_a to compare representational dissimilarity matrices (subject-group-average τ_a plotted). Results reproduced from Khaligh-Razavi & Kriegeskorte (2014).

network representation explained the IT representation substantially and significantly better than a similarly IT-fitted combination of the conventional computer vision features.

Cadieu et al. (2013, 2014) analyzed the internal representations of a population of IT cells alongside models of early vision, the HMAX model (Riesenhuber & Poggio 1999, Serre et al. 2007), a hierarchically optimized multilayer model from Yamins et al. (2013, 2014), and the deep neural networks from Krizhevsky et al. (2012) and Zeiler & Fergus (2014). The representations performing best at object categorization (**Figure 6*a***) were the deep neural network built by Zeiler & Fergus (2014) and the biological IT representation (monkey neuronal recordings), followed closely by the deep network proposed by Krizhevsky et al. (2012). The other representations performed at much lower levels. The two deep networks explained the IT data equally well, as did neuronal recordings from an independent set of IT neurons (**Figure 6*b***).

Several additional studies have yielded similar results and are beginning to characterize the extent to which representations at different depths can explain the representational stages of the ventral stream (Agrawal et al. 2014, Güçlü & van Gerven 2015). Overall, these early empirical comparisons between deep neural network models and the primate ventral stream suggest four

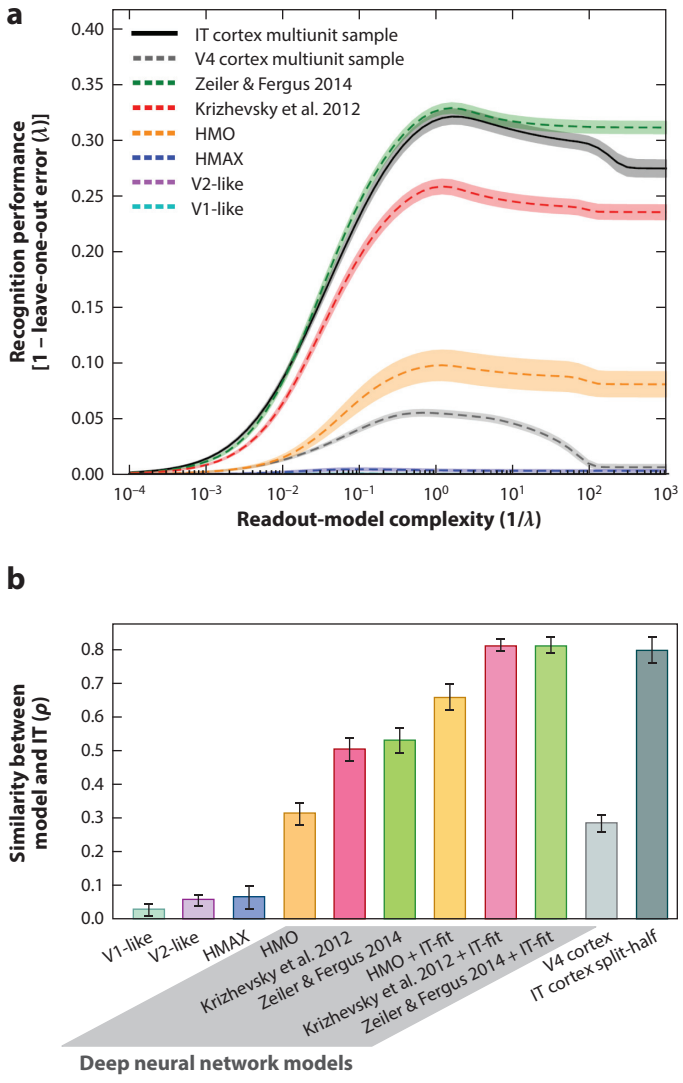

Figure 6

Deep neural networks beat simpler computational models at recognition and better explain the IT representation. (*a*) Object recognition performance of deep neural networks beats that of shallower models and rivals that of a population of IT neurons recorded in a monkey. Recognition performance (*vertical axis*) is plotted as a function of readout-model complexity (*horizontal axis*); high performance at low complexity indicates that the categories occupy easily separable regions in the representational space. (*b*) Deep neural network representations more closely resemble IT than do three simpler models (V1-like, V2-like, and HMAX). The similarity between each model and IT (*vertical axis*) was measured using the Spearman's rank correlation coefficient (ρ) to compare representational dissimilarity matrices. Results reproduced from Cadieu et al. (2014). Abbreviations: HMAX, hierarchical model and X (Riesenhuber & Poggio 1999, Serre et al. 2007, Tarr 1999); HMO, hierarchical modular optimization model (Yamins et al. 2014); IT, inferior temporal cortex.

conclusions: (*a*) Only deep neural networks perform object recognition at levels comparable to human performance; (*b*) only deep neural networks explain the representational geometry of IT; (*c*) the representation appears to be gradually transformed, with lower layers resembling the earlier stages of the primate ventral stream; and (*d*) the high-level deep network representation resembles IT not merely in that it emphasizes categorical divisions, but also in its within-category representational geometry.

Discriminative model: a model that extracts information of interest from the data (e.g., the image) without explicitly representing the process that generated the data

RECURRENT NEURAL NETWORKS FOR VISION

Feedforward networks are useful as models of the initial sweep of neuronal signaling through the visual hierarchy. They go some way toward explaining vision at a glance. However, feedforward networks are unlike the brain in terms of their connectivity and dynamics and are fundamentally limited to the computation of static functions. Rather than computing a static function on each of a series of image frames, vision takes a time-continuous input stream and interprets it through ongoing recurrent computations. The current network state likely represents the recent stimulus history, along with predictions of impending events and other behaviorally important information.

Recurrent computations probably contribute to the automatic rapid interpretation of even static visual images (Sugase et al. 1999, Brincat & Connor 2006, Freiwald & Tsao 2010, Carlson et al. 2013, Cichy et al. 2014, H. Tang et al. 2014, Clarke et al. 2015), leading to the emergence of representations that clearly distinguish particular objects and object categories. Individual faces, for example, become more clearly distinguishable in the monkey neuronal population codes at latencies that exceed the 100 ms or so that it takes the feedforward sweep to reach IT (Sugase et al. 1999, Freiwald & Tsao 2010). Similarly, at the level of object categories, evidence from human magnetoencephalography (MEG) suggests that strong categorical divisions arise only at latencies of over 200 ms after stimulus onset (Carlson et al. 2013, Cichy et al. 2014, Clarke & Tyler 2015, Clarke et al. 2015). Both category and exemplar representations, thus, may rely on recurrent processing to achieve invariance to irrelevant variation among images that carry the same essential meaning.

The brain might rely on a combination of feedforward and recurrent processing to arrive at a representation similar to that computed in feedforward convolutional networks trained for object recognition. We have seen that a recurrent network can be unfolded as a deep feedforward network. Conversely, when the numbers of units and connections are limited, a desirable function computed by a very large feedforward network might alternatively be approximated by recurrent computations in a smaller network. Recurrent dynamics can expand computational power by multiplying the limited physical resources for computation along time.

Recurrent neuronal dynamics likely also serve more sophisticated computations than those of feedforward convolutional networks. For example, assume the function of a visual neuron is to represent the presence of some piece of content in the image (a feature, an object part, or an object). The feedforward sweep alone might not provide the full evidence the neuron needs to confidently detect the piece of content that it represents. The neuron might therefore integrate later-arriving lateral and top-down signals to converge on its ultimate response. Multiple neurons might pass messages recurrently until the population converges on a stable interpretation of the image.

Recurrent computations might implement the iterative fitting to the image of a generative model of image formation, in which the fitted parameters specify the contents (and causes) of the image (see the sidebar The Deep Mystery of Vision: How to Integrate Generative and Discriminative Models). Assume, for simplicity, that the generative model is exactly invertible. This might be plausible if the model includes prior world knowledge sufficient to disambiguate visual images. Images and symbolic descriptions of their contents are then related by a one-to-one (bijective)

THE DEEP MYSTERY OF VISION: HOW TO INTEGRATE GENERATIVE AND DISCRIMINATIVE MODELS

The recent advances in computer vision were largely driven by feedforward neural networks. These models are discriminative: They discriminate categories among sets of images without an explicit model of the image-formation process. A more principled way to implement vision (or any data analysis) is to formulate a model of the process that generated the image (the data) and then to invert the process in order to infer the parameters of the model from the data (for a textbook on this approach in computer vision, see Prince 2012).

For vision, the generative model is an image-formation (or graphics) model that generates images from some high-level representation, such as a symbolic description of the visual scene. The first challenge is to define such a model. The second challenge is to perform inference on it—that is, to find the high-level representation that best explains the image, for example, the maximum a posteriori estimate or the full posterior probability distribution over all possible high-level representations, given the image. The idea of an active search for the interpretation that best explains the evidence in the context of our prior knowledge about the world is captured in von Helmholtz's (1866) description of vision as unconscious inference.

Research in computer vision and biological vision has always spanned the entire gamut from discriminative to generative approaches (Knill et al. 1996, Yuille & Kersten 2006, Prince 2012). However, the generative approach is practically challenging for computer vision and theoretically challenging for neuroscience. Most computer vision systems, whether using hand-engineered features or deep learning, therefore still rely primarily on discriminative models, learning mostly feedforward computations that process images to produce the desired outputs (but see Prince 2012). In neuroscience, similarly, feedforward models such as HMAX (Riesenhuber & Poggio 1999) have been influential.

Image formation involves nonlinear processes such as occlusion. Whereas the inversion of a linear generative model has a closed-form solution that can be implemented in a feedforward computation, inverting a graphics model is computationally much more challenging. Inferring a high-level scene description from an image requires consideration (at some level of abstraction) of a combinatorial explosion of possible configurations of objects and lights.

The deep mystery of vision is exactly how discriminative and generative models are integrated into a seamless and efficient process of inference. Vision might rely on a discriminative feedforward model for rapid recognition at a glance and on recurrent dynamics for iterative refinement of the inference, for correcting the errors of an initial feedforward estimate, or for probabilistic inference on hypotheses highlighted by the feedforward pass.

Recurrent neural networks can implement such dynamic inference processes and, given recent successes in the domain of language processing, seem poised for a fundamental advance in vision research.

mapping. In principle, the inverse of the generative model could be represented by a feedforward model (because of universality). Such a model might require too many neurons and connections, however, or its connectivity might be impossible to learn from limited training data. Instead of analyzing the image through feedforward computations, we can perform analysis by synthesis (Yuille & Kersten 2006), fitting a generative model of image formation to the particular image to be recognized.

The inversion of generative models has long been explored in both brain science and computer vision (Knill et al. 1996, Yuille & Kersten 2006, Prince 2012). The inference problem is difficult because a vast number of combinations of surfaces and lights can explain any image. To constrain the search space and disambiguate the solution, the brain must use massive prior knowledge about the world. Inference on a generative model might be tractable if performed on an intermediate-level representation of the image computed by discriminative mechanisms. How

the brain combines discriminative computations with inference on generative models to perceive the world is one of the fundamental unsolved problems in brain science.

The Helmholtz machine (Dayan et al. 1995) uses analysis by synthesis at the level of learning. A bottom-up recognition model and a top-down generative model are concurrently learned so as to best represent the distribution of the inputs in a maximum-likelihood sense. The learning can be performed using the wake–sleep algorithm (Hinton et al. 1995, Dayan 2003). In the wake phase, the recognition model "perceives" training images, and the generative model learns to better reconstruct these images from their internal representations. In the sleep phase, the generative model "dreams" of images, and the recognition model learns to better infer the internal representations from the images. By alternating wake and sleep phases, the two models coadapt and jointly discover a good representation for the distribution of images used in training.

A recurrent network could use the feedforward sweep to compute an initial rough estimate of the causes, and it could use subsequent recurrent computations to iteratively reduce the prediction error of the generative model and to explain nonlinear interactions of the parts, such as occlusion. The process could use predictive coding (Lee & Mumford 2003, Friston 2010) with recognized parts of the image explained away (and subtracted out of) lower-level representations. In such a process, the parts yet unexplained would be gradually uncluttered in the low-level representation and contextualized by the high-level representation as the easier, and then the more difficult, components of the image are successively explained.

Recurrent computations might converge on a point estimate of the parameters of a generative model of the image. Alternatively, they might implement probabilistic inference, converging on a representation of the posterior distribution over the parameters of the generative model. Recurrent message passing can implement belief propagation, an algorithm for probabilistic inference on a generative model. If the model captures the causal process giving rise to images, the recurrent dynamics can infer the specific causes (e.g., the objects, their properties, and the lighting) of a particular image. This process can be implemented in recurrent neural networks and might explain how the brain performs optimal cue combination, temporal integration, and explaining away (Lochmann & Deneve 2011). Belief propagation is a deterministic algorithm for probabilistic inference. Another deterministic proposal is based on probabilistic population codes (Ma et al. 2006).

Alternatively, a neural network might perform probabilistic inference by Markov chain Monte Carlo (MCMC) sampling, using neural stochasticity as a random generator (Hoyer & Hyvärinen 2003, Fiser et al. 2010, Buesing et al. 2011, McClelland 2013, Häfner et al. 2014). In this view, a snapshot of neural population activity represents a point estimate of the stimulus, and a temporal sequence of such snapshots represents the posterior distribution. For near-instantaneous readout of a probabilistic representation, several MCMC chains could operate in parallel (Savin & Deneve 2014). The sampling approach naturally handles the representation of joint probability distributions of multiple variables.

These proposals are exciting because they explain how the brain might perform formal probabilistic inference with neurons, linking the biological hardware to the high-level goal of rational information processing. The ultimate goal, of course, is not rational inference, but successful behavior, that is, survival and reproduction. We should expect the brain to perform probabilistic inference only to the extent to which it is expedient to do so in the larger context of successful behavior (Gershman et al. 2015).

How the probabilistic inference proposals of computational neuroscience scale up to the real-world challenges of vision remains to be seen. If they do, they might have a central future role in both brain theory and computer vision. The brain clearly handles uncertainty well in many contexts (Tenenbaum et al. 2006, Pouget et al. 2013), so it is helpful to view its inferences as approximations, however rough, to rational probabilistic inference.

At a larger timescale, vision involves top-down effects related to expectation and attentional scrutiny, as well as active exploration of a scene through a sequence of eye movements and through motor manipulations of the world. With the recurrent loop expanded to pass through the environment, these processes bring limited resources (the fovea, conscious attention) to different parts of a scene sequentially, selectively sampling the most relevant information while accumulating evidence toward an overall interpretation. Active perception is being explored in the computational literature. For example, Y. Tang et al. (2014b) built a model for face recognition that uses a convolutional feedforward pass for initialization and an attentional mechanism for selection of a region of interest, on which probabilistic inference is performed using a generative model, which itself is learned from data.

The challenge ahead is, first, to scale recurrent neural network models for vision to real-world tasks and human performance levels and, second, to fit and compare their representational dynamics to biological brains. Recurrent models are already successful in several domains of AI, including video-to-text description (Venugopalan et al. 2015), speech-to-text recognition (Sak et al. 2014), text-to-text language translation (Sutskever et al. 2014, Cho et al. 2014), and text-to-speech synthesis (Fan et al. 2014). In brain science, recurrent neural net models will ultimately be needed to explain every major function of information processing, including vision, other perceptual processes, cognition, and motor control.

CONCLUSIONS

Computational neuroscience has been very successful by asking what the brain *should* compute (Körding 2007). The normative goals proposed have often led to important insights before being replaced by larger goals. Should the brain efficiently encode sensory information [Barlow 1961 (2012)]? Or should it infer an accurate probabilistic representation of the world (Barlow 2001)? The ultimate goal is successful behavior.

Normative theory has driven advances at the cognitive and neuronal levels. Successes of this approach include theories of efficient coding [Barlow 1961 (2012), Olshausen & Field 1997, Simoncelli & Olshausen 2001], probabilistic neuronal coding and inference (Hoyer & Hyvärinen 2003, Fiser et al. 2010, Buesing et al. 2011, McClelland 2013, Pouget et al. 2013), Bayesian sensorimotor control (Körding & Wolpert 2006), and probabilistic cognition (Tenenbaum et al. 2006). For low-level sensory representations and for low-dimensional decision and control processes, normative theories prescribe beautiful and computationally simple inference procedures, which we know how to implement in computers and which might plausibly be implemented in biological brains. However, visual recognition and many other feats of brain information processing require inference using massive amounts of world knowledge. Not only are we missing a normative theory that would specify the optimal solution, but, until recently, we were not even able to implement any functioning solution.

Until recently, computers could not do visual object recognition, and image-computable models that could predict higher-level representations of novel natural images did not exist. Deep neural networks put both the task of object recognition and the prediction of high-level neural responses within our computational reach. This advance opens up a new computational framework for modeling high-level vision and other brain functions.

Deep neural net models are optimized for task performance. In this sense, the framework addresses the issue of what the brain *should* compute at the most comprehensive level: that of successful behavior. In its current instantiation, the deep net framework gives up an explicit probabilistic account of inference, in exchange for neurally plausible models that have sufficient capacity to

solve real-world tasks. We will see in the future whether explicitly probabilistic neural net models can solve the real-world tasks and explain biological brains even better.

Synthetic neurophysiology: computational analysis of responses and dynamics of artificial neural networks aimed to gain a higher-level understanding of their computational mechanisms

Replacing One Black Box by Another?

One criticism of using complex neural networks to model brain information processing is that it replaces one impenetrably complex network with another. We might be able to capture the computations, but we are capturing them in a large net, the complexity of which defies conceptual understanding. There are two answers to the criticism of impenetrability.

First, it is true that our job is not done when we have a model that is predictive of neural responses and behavior. We must still strive to understand—at a higher level of description—how exactly the network transforms representations across the multiple stages of a deep hierarchy (and across time when the network is recurrent). However, once we have captured the complex biological computations in an artificial neural network, we can study its function efficiently in silico—with full knowledge of its internal dynamics. Synthetic neurophysiology, the analysis and visualization of artificial network responses to large natural and artificial stimulus sets, might help reveal the internal workings of these networks (Zeiler & Fergus 2014, Girshick et al. 2014, Simonyan et al. 2014, Tsai & Cox 2015, Zhou et al. 2015, Yosinski et al. 2015).

The second answer to the criticism of the impenetrability of neural network models is that we should be prepared to deal with mechanisms that elude a concise mathematical description and an intuitive understanding. After all, intelligence requires large amounts of domain-specific knowledge, and compressing this knowledge into a concise description or mathematical formula might not be possible. In other words, our models should be as simple as possible, but no simpler.

Similar to computational neuroscience, AI began with simple and general algorithms. These algorithms did not scale up to real-world applications, however. Real intelligence turned out to require incorporating large amounts of knowledge. This insight eventually led to the rise of machine learning. Computational neuroscience must follow in the footsteps of AI and acknowledge that most of what the brain does requires ample domain-specific knowledge learned through experience.

Are Deep Neural Net Models Similar to Biological Brains?

The answer to this question is in the eye of the beholder. We can focus on the many abstractions from biological reality and on design decisions driven by engineering considerations and conclude that they are very different. Alternatively, we can focus on the original biological inspiration and on the fact that biological neurons can perform the operations of model units, and conclude that they are similar.

Abstraction from biological detail is desirable and is in fact a feature of all models of computational neuroscience. A model is not meant to be identical to its object, but rather to explain it at an abstract level of description. Merely pointing out a difference to biological brains, therefore, does not constitute a legitimate challenge. For example, the fact that real neurons spike does not pose a challenge to a rate-coding model. It just means that biological brains can be described at a finer level of detail that the model does not address. If spiking were a computational requirement (e.g., Buesing et al. 2011) and a spiking model outperformed the best rate-coding model at its own game of predicting spike rates, or at predicting behavior, however, then this model would present a challenge to the rate-coding approach.

Many features of the particular type of deep convolutional feedforward network currently dominating computer vision deserve to be challenged in the context of modeling biological vision (see the sidebar Adversarial Examples Can Reveal Idiosyncrasies of Neural Networks). The features

ADVERSARIAL EXAMPLES CAN REVEAL IDIOSYNCRASIES OF NEURAL NETWORKS

Fooling vision can help us learn about its mechanisms. This is true for both biological and artificial vision. Researchers are exploring how artificial neural networks represent images by trying to fool them (Szegedy et al. 2014, Goodfellow et al. 2015, Nguyen et al. 2015). They use optimization techniques to design images that are incorrectly classified. An adversarial example is an image from category X (e.g., a bus or a noise image) that has been designed to be misclassified by a particular network as belonging to category Y (e.g., an ostrich). Such an image can be designed by taking a natural image from category X and adjusting the pixels to fool the net. The backpropagation algorithm, which usually serves to find small adjustments to the weights that reduce the error for an image, can be used to find small adjustments to the image that instead create an error. For the convolutional neural networks currently used in computer vision, adversarial examples can be created by very slight changes to the image that clearly do not render it a valid example of a different category. Such adversarial examples can look indistinguishable from the original image to humans. This has been taken as evidence of the limitations of current neural network architectures both as vision systems and as models of human vision.

An adversarial example created to fool an artificial neural network will not usually fool a human observer. However, it is not known whether adversarial examples can similarly be created for human visual systems. The technique described above for constructing adversarial examples requires full knowledge of the connections of the particular network to be fooled. Current psychophysical and neurophysiological techniques cannot match this process to fool biological vision. An intriguing possibility, thus, is that biological visual systems, too, are susceptible to adversarial examples. These could exploit idiosyncrasies of a particular brain, such that an adversarial example created to fool one person will not fool another. The purpose of vision systems is to work well under natural conditions, not to be immune to extremely savvy sabotage that requires omniscient access to the internal structure of a network and precise stabilization of the fooling image on the visual sensor array. Human vision is famously susceptible to visual illusions of various kinds. Moreover, from a machine learning perspective, it appears inevitable that adversarial examples can be constructed for any learning system—artificial or natural—that must rely on an imperfect inductive bias to generalize from a limited set of examples to a high-dimensional classification function.

What lessons do the adversarial examples hold about current neural network models? If adversarial examples fooled only the particular instance of a network for which they were constructed, exploiting the idiosyncrasies of that particular net, then they would be easy to dismiss. However, adversarial examples generalize across networks to some extent. If a new network is created by initializing the same architecture with new random weights and training it with the same set of labeled images, the resulting network will often still be fooled by an adversarial example created for the original network. Adversarial examples also generalize to slightly altered architectures if the same training set is used. If the training set is changed, adversarial examples created for the original network are not very effective anymore, but they may still be misclassified at a higher rate than natural images. This suggests that adversarial examples exploit network idiosyncrasies resulting largely from the training set, but also, to some extent, from the basic computational operations used. One possibility is that the linear combination computed by each unit in current systems makes these systems particularly easy to fool (Goodfellow et al. 2015). In essence, each unit divides its input space by a linear boundary (even if its activation rises smoothly as we cross the boundary for sigmoid or linearly on the preferred side for rectified linear activation functions). In contrast, networks using radial basis functions, in which each unit has a particular preferred pattern in its input space and the response falls off in all directions, might be harder to fool. However, these networks are also harder to train—and perhaps for the same reason. It will be intriguing to see this puzzle solved as we begin to compare the complex representational transformations between artificial and biological neural networks in greater detail.

that deserve to be challenged first are the higher-level computational mechanisms, such as the lack of bypass connections in the feedforward architecture, the lack of feedback and local recurrent connections, the linear–nonlinear nature of the units, the rectified linear activation function, the max-pooling operation, and the offline supervised gradient-descent learning. To challenge one of these features, we must demonstrate that measured neuronal responses or behavioral performance can be more accurately predicted using a different kind of model.

The neural network literature is complex and spans the gamut from theoretical neuroscience to computer science. This literature includes feedforward and recurrent, discriminative and generative, deterministic and stochastic, nonspiking and spiking models. It provides the building blocks for tomorrow's more comprehensive theories of information processing in the brain. Now that these models are beginning to scale up to real-world tasks and human performance levels in engineering, we can begin to use this modeling framework in brain science to tackle the complex processes of perception, cognition, and motor control.

The Way Ahead

We will use modern neural network technology with the goal of approximating the internal dynamics and computational function of large portions of biological brains, such as their visual systems. An important goal is to build models with layers that correspond one-to-one to visual areas, and with receptive fields, nonlinear response properties, and representational geometries that match those of the corresponding primate visual areas. The requirement that the system perform a meaningful task such as object recognition provides a major functional constraint. Task training of neural networks with millions of labeled images currently provides much stronger constraints than neurophysiological data do on the space of candidate models. Indeed, the recent successes at predicting brain representations of novel natural images are largely driven by task training (Yamins et al. 2014, Khaligh-Razavi & Kriegeskorte 2014, Cadieu et al. 2014). However, advances in massively parallel brain-activity measurement promise to provide stronger brain-based constraints on the model space in the future. Rather than minimizing a purely task-based loss function, as commonly done in engineering, modeling biological brains will ultimately require novel learning algorithms that drive connectivity patterns, internal representations, and task performance into alignment with brain and behavioral measurements. In order to model not only the final processing mechanism but the learning process in a biologically plausible way, we will also need to employ unsupervised and reinforcement learning techniques (Sutton & Barto 1998, Mnih et al. 2015).

AI, machine learning, and the cognitive and brain sciences have deep common roots. At the cognitive level, these fields have recently converged through Bayesian models of inference and learning (Tenenbaum et al. 2006). Like deep networks, Bayesian nonparametric techniques (Ghahramani 2013) can incorporate large amounts of world knowledge. These models have the advantage of explicit probabilistic inference and learning. Explaining how such inference processes might be implemented in biological neural networks is one of the major challenges ahead. Neural networks have a long history in AI, in cognitive science, in machine learning, and in computational neuroscience. They provide a common modeling framework to link these fields. The current vindication in engineering of early intuitions about the power of brain-like deep parallel computation reinvigorates the convergence of these disciplines. If we can build models that perform complex feats of real-world intelligence (AI) and explain neuronal dynamics (computational neuroscience) and behavior (cognitive science), then—for the tasks tackled—we will understand how the brain works.

SUMMARY POINTS

1. Neural networks are brain-inspired computational models that now dominate computer vision and other AI applications.
2. Neural networks consist of interconnected units that compute nonlinear functions of their input. Units typically compute weighted combinations of their inputs followed by a static nonlinearity.
3. Feedforward neural networks are universal function approximators.
4. Recurrent neural networks are universal approximators of dynamical systems.
5. Deep neural networks stack multiple layers of nonlinear transformations and can concisely represent complex functions such as those needed for vision.
6. Convolutional neural networks constrain the input connections of units in early layers to local receptive fields with weight templates that are replicated across spatial positions. The restriction and sharing of weights greatly reduce the number of parameters that need to be learned.
7. Deep convolutional feedforward networks for object recognition are not biologically detailed and rely on nonlinearities and learning algorithms that may differ from those of biological brains. Nevertheless they learn internal representations that are highly similar to representations in human and nonhuman primate IT cortex.
8. Neural networks now scale to real-world AI tasks, providing an exciting technological framework for building more biologically faithful models of complex feats of brain information processing.

FUTURE ISSUES

1. We will build neural net models that engage complex real-world tasks and simultaneously explain biological brain-activity patterns and behavioral performance.
2. The models will have greater biological fidelity in terms of architectural parameters, nonlinear representational transformations, and learning algorithms.
3. Network layers should match the areas of the visual hierarchy in their response characteristics and representational geometries.
4. Models should predict a rich array of behavioral measurements, such as reaction times for particular stimuli in different tasks, similarity judgments, task errors, and detailed motor trajectories in continuous interactive tasks.
5. New supervised learning techniques will drive neural networks into alignment with measured functional and anatomical brain data and with behavioral data.
6. Recurrent neural network models will explain the representational dynamics of biological brains.
7. Recurrent neural network models will explain how feedforward, lateral, and feedback information flow interact to implement probabilistic inference on generative models of image formation.

8. We will tackle more complex visual functions beyond categorization, such as identification of unique entities, attentional shifts and eye movements that actively explore the scene, visual search, image segmentation, more complex semantic interpretations, and sensorimotor integration.

DISCLOSURE STATEMENT

The author is not aware of any affiliations, memberships, funding, or financial holdings that might be perceived as affecting the objectivity of this review.

ACKNOWLEDGMENTS

The author thanks Seyed Khaligh-Razavi and Daniel Yamins for helpful discussions, and Patrick McClure and Katherine Storrs for comments on a draft of the manuscript. This research was funded by the UK Medical Research Council (Program MC-A060-5PR20), a European Research Council Starting Grant (ERC-2010-StG 261352), and a Wellcome Trust Project Grant (WT091540MA).

LITERATURE CITED

Agrawal P, Stansbury D, Jitendra Malik J, Gallant JL. 2014. Pixels to voxels: modeling visual representation in the human brain. arXiv:1407.5104 [q-bio.NC]

Barlow H. 2001. Redundancy reduction revisited. *Netw. Comput. Neural Syst.* 2(3):241–53

Barlow HB. 1961 (2012). Possible principles underlying the transformations of sensory messages. In *Sensory Communication*, ed. WA Rosenblith, pp. 217–34. Cambridge, MA: MIT Press

Bengio Y. 2009. *Learning Deep Architectures for AI*. Hanover, MA: Now

Brincat SL, Connor CE. 2006. Dynamic shape synthesis in posterior inferotemporal cortex. *Neuron* 49:17–24

Buesing L, Bill J, Nessler B, Maass W. 2011. Neural dynamics as sampling: a model for stochastic computation in recurrent networks of spiking neurons. *PLOS Comput. Biol.* 7(11):e1002211

Cadieu CF, Hong H, Yamins DL, Pinto N, Ardila D, et al. 2014. Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLOS Comput. Biol.* 10(12):e1003963

Cadieu CF, Hong H, Yamins DL, Pinto N, Majaj NJ, DiCarlo JJ. 2013. *The neural representation benchmark and its evaluation on brain and machine*. Presented at Int. Conf. Learn. Represent., Scottsdale, AZ, May 2–4. arXiv:1301.3530 [cs.NE]

Carlson T, Tovar DA, Alink A, Kriegeskorte N. 2013. Representational dynamics of object vision: the first 1000 ms. *J. Vis.* 13:1

Chatfield K, Simonyan K, Vedaldi A, Zisserman A. 2014. *Return of the devil in the details: delving deep into convolutional nets*. Presented at Br. Mach. Vis. Conf., Nottingham, UK, Sept. 1–5. arXiv:1405.3531 [cs.CV]

Cho K, van Merrienboer B, Gulcehre C, Bougares F, Schwenk H, Bengio Y. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. Presented at Conf. Empir. Methods Nat. Lang. Process., Doha, Qatar, Oct. 25–29. arXiv:1406.1078 [cs.CL]

Cichy RM, Pantazis D, Oliva A. 2014. Resolving human object recognition in space and time. *Nat. Neurosci.* 17:455–62

Clarke A, Devereux BJ, Randall B, Tyler LK. 2015. Predicting the time course of individual objects with MEG. *Cereb. Cortex* 25:3602–12

Clarke A, Tyler LK. 2015. Understanding what we see: how we derive meaning from vision. *Trends Cogn. Sci.* 19:677–87

Cybenko G. 1989. Approximation by superpositions of a sigmoid function. *Math. Control Signals Syst.* 2:303–14

Dayan P. 2003. Helmholtz machines and wake-sleep learning. In *The Handbook of Brain Theory and Neural Networks*, ed. MA Arbib, pp. 520–25. Cambridge, MA: MIT Press. 2nd ed.
Dayan P, Hinton GE, Neal RM, Zemel RS. 1995. The Helmholtz machine. *Neural Comput.* 7(5):889–904
Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. 2009. ImageNet: a large-scale hierarchical image database. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 20–25, Miami*, pp. 248–55. New York: IEEE
Dumoulin SO, Wandell BA. 2008. Population receptive field estimates in human visual cortex. *NeuroImage* 39:647–60
Fan Y, Qian Y, Xie F-L, Soong FK. 2014. TTS synthesis with bidirectional LSTM based recurrent neural networks. *Proc. 15th Annu. Conf. Int. Speech Commun. Assoc., Sept. 14–18, Singapore*, pp. 1964–68. Baixas, Fr.: ISCA
Fiser J, Berkes P, Orbán G, Lengyel M. 2010. Statistically optimal perception and learning: from behavior to neural representations. *Trends Cogn. Sci.* 14:119–30
Freiwald WA, Tsao DY. 2010. Functional compartmentalization and viewpoint generalization within the macaque face-processing system. *Science* 330(6005):845–51
Friston K. 2010. The free-energy principle: a unified brain theory? *Nat. Rev. Neurosci.* 11(2):127–38
Fukushima K. 1980. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.* 36(4):193–202
Gallistel CR, King AP. 2011. *Memory and the Computational Brain: Why Cognitive Science Will Transform Neuroscience*. Chichester, UK: Wiley-Blackwell
Gershman SJ, Horvitz EJ, Tenenbaum JB. 2015. Computational rationality: a converging paradigm for intelligence in brains, minds, and machines. *Science* 349(6245):273–78
Ghahramani Z. 2013. Bayesian non-parametrics and the probabilistic approach to modelling. *Philos. Trans. R. Soc. Lond. A* 371:20110553
Girshick R, Donahue J, Darrell T, Malik J. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv:1311.2524 [cs.CV]
Goodfellow IJ, Shlens J, Szegedy C. 2015. Explaining and harnessing adversarial examples. arXiv:1412.6572v3 [stat.ML]
Graves A, Schmidhuber J. 2009. Offline handwriting recognition with multidimensional recurrent neural networks. *Adv. Neural Inf. Process. Syst.* 21:545–52
Güçlü U, van Gerven MA. 2015. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J. Neurosci.* 35(27):10005–14
Häfner RM, Berkes P, Fiser J. 2014. Perceptual decision-making as probabilistic inference by neural sampling. arXiv:1409.0257 [q-bio.NC]
Hilgetag CC, O'Neill MA, Young MP. 2000. Hierarchical organization of macaque and cat cortical sensory systems explored with a novel network processor. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* 355:71–89
Hinton GE, Dayan P, Frey BJ, Neal RM. 1995. The "wake-sleep" algorithm for unsupervised neural networks. *Science* 268(5214):1158–61
Hinton GE, Salakhutdinov RR. 2006. Reducing the dimensionality of data with neural networks. *Science* 313(5786):504–7
Hinton GE, Srivastava N, Krizhevsky A, Sutskever I, Salakhutdinov RR. 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580 [cs.NV]
Hochreiter S. 1991. *Untersuchungen zu dynamischen neuronalen Netzen.* Master's Thesis, Inst. Inform., Tech. Univ. München
Hochreiter S, Bengio Y, Frasconi P, Schmidhuber J. 2001. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. In *A Field Guide to Dynamical Recurrent Neural Networks*, ed. SC Kremer, JF Kolen, pp. 237–244. New York: IEEE
Hochreiter S, Schmidhuber J. 1997. Long short-term memory. *Neural Comput.* 9(8):1735–80
Hornik K. 1991. Approximation capabilities of multilayer feedforward networks. *Neural Netw.* 4(2):251–57
Hoyer PO, Hyvärinen A. 2003. Interpreting neural response variability as Monte Carlo sampling of the posterior. *Adv. Neural Inform. Proc. Syst.* 15:293–300
Hubel DH, Wiesel TN. 1968. Receptive fields and functional architecture of monkey striate cortex. *J. Physiol.* 195:215

Jaeger H. 2001. *The "echo state" approach to analysing and training recurrent neural networks—with an erratum note*. GMD Tech. Rep. 148, Ger. Natl. Res. Cent. Inf. Technol., Bonn
Jozwik KM, Kriegeskorte N, Mur M. 2015. Visual features as stepping stones toward semantics: explaining object similarity in IT and perception with non-negative least squares. *Neuropsychologia.* In press
Kay KN, Naselaris T, Prenger RJ, Gallant JL. 2008. Identifying natural images from human brain activity. *Nature* 452:352–55
Khaligh-Razavi S-M, Kriegeskorte N. 2013. *Object-vision models that better explain IT also categorize better, but all models fail at both*. Presented at COSYNE, Salt Lake City, UT
Khaligh-Razavi S-M, Kriegeskorte N. 2014. Deep supervised, but not unsupervised, models may explain IT cortical representation. *PLOS Comput. Biol.* 10(11):e1003915. doi:10.1371/journal.pcbi.1003915
Kiani R, Esteky H, Mirpour K, Tanaka K. 2007. Object category structure in response patterns of neuronal population in monkey inferior temporal cortex. *J. Neurophysiol.* 97:4296–309
Knill DC, Kersten D, Yuille A. 1996. Introduction: a Bayesian formulation of visual perception. In *Perception as Bayesian Inference*, ed. DC Knill, W Richards, pp. 1–21. Cambridge, UK: Cambridge Univ. Press
Körding K. 2007. Decision theory: What "should" the nervous system do? *Science* 318(5850):606–10
Körding KP, Wolpert DM. 2006. Bayesian decision theory in sensorimotor control. *Trends Cogn. Sci.* 10(7):319–26
Kriegeskorte N. 2011. Pattern-information analysis: from stimulus decoding to computational-model testing. *NeuroImage* 56:411–21
Kriegeskorte N, Kievit RA. 2013. Representational geometry: integrating cognition, computation, and the brain. *Trends Cogn. Sci.* 17:401–12
Kriegeskorte N, Mur M, Bandettini P. 2008a. Representational similarity analysis—connecting the branches of systems neuroscience. *Front. Syst. Neurosci.* 2:4
Kriegeskorte N, Mur M, Ruff DA, Kiani R, Bodurka J, et al. 2008b. Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron* 60:1126–41
Krizhevsky A, Sutskever I, Hinton GE. 2012. ImageNet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 25:1097–105
LeCun Y, Bengio Y. 1995. Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*, ed. MA Arbib, pp. 255–58. Cambridge, MA: MIT Press
LeCun Y, Bengio Y, Hinton G. 2015. Deep learning. *Nature* 521:436–44
LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, et al. 1989. Backpropagation applied to handwritten zip code recognition. *Neural Comput.* 1(4):541–51
Lee TS, Mumford D. 2003. Hierarchical Bayesian inference in the visual cortex. *J. Opt. Soc. Am. A* 20(7):1434–48
Lochmann T, Deneve S. 2011. Neural processing as causal inference. *Curr. Opin. Neurobiol.* 21(5):774–81
Lowe DG. 1999. Object recognition from local scale-invariant features. *Proc. 7th IEEE Int. Conf. Comput. Vis., Sept. 20–27, Kerkyra, Greece*, pp. 1150–57. New York: IEEE
Ma WJ, Beck JM, Latham PE, Pouget A. 2006. Bayesian inference with probabilistic population codes. *Nat. Neurosci.* 9:1432–38
Maass W, Natschläger T, Markram H. 2002. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Comput.* 14(11):2531–60
McClelland JL. 2013. Integrating probabilistic models of perception and interactive neural networks: a historical and tutorial review. *Front. Psychol.* 4:503
McCulloch WS, Pitts W. 1943. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* 5(4):115–33
Minsky M, Papert S. 1972. *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA: MIT Press
Mitchell TM, Shinkareva SV, Carlson A, Chang K-M, Malave VL, et al. 2008. Predicting human brain activity associated with the meanings of nouns. *Science* 320:1191–95
Moore CI, Cao R. 2008. The hemo-neural hypothesis: on the role of blood flow in information processing. *J. Neurophysiol.* 99(5):2035–47
Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, et al. 2015. Human-level control through deep reinforcement learning. *Nature* 518:529–33

Nguyen A, Yosinski J, Clune J. 2015. *Deep neural networks are easily fooled: high confidence predictions for unrecognizable images*. Presented at IEEE Conf. Comput. Vis. Pattern Recognit., June 7–12, Boston. arXiv:1412.1897v4 [cs.CV]

Nili H, Wingfield C, Walther A, Su L, Marslen-Wilson W, Kriegeskorte N. 2014. A toolbox for representational similarity analysis. *PLOS Comput. Biol.* 10:e1003553

Olshausen BA, Field DJ. 1997. Sparse coding with an overcomplete basis set: a strategy employed by V1? *Vis. Res.* 37(23):3311–25

Pouget A, Beck JM, Ma WJ, Latham PE. 2013. Probabilistic brains: knowns and unknowns. *Nat. Neurosci.* 16(9):1170–78

Prince SJ. 2012. *Computer Vision: Models, Learning, and Inference*. Cambridge, UK: Cambridge Univ. Press

Riesenhuber M, Poggio T. 1999. Hierarchical models of object recognition in cortex. *Nat. Neurosci.* 2(11):1019–25

Rosenblatt F. 1958. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychol. Rev.* 65(6):386

Rumelhart DE, Hinton GE, Williams RJ. 1986. Learning representations by back-propagating errors. *Nature* 323(6088):533–36

Rumelhart DE, McClelland JL, PDP Research Group. 1988. In *Parallel Distributed Processing* Vol. 1, pp. 354–62. New York: IEEE

Sak H, Senior A, Beaufays F. 2014. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv:1402.1128 [cs.NE]

Savin C, Deneve S. 2014. Spatio-temporal representations of uncertainty in spiking neural networks. *Adv. Neural Inf. Process. Syst.* 27:2024–32

Schäfer AM, Zimmermann HG. 2007. Recurrent neural networks are universal approximators. *Int. J. Neural Syst.* 17(4):253–63

Schmidhuber J. 2015. Deep learning in neural networks: an overview. *Neural Netw.* 61:85–117

Serre T, Wolf L, Bileschi S, Riesenhuber M, Poggio T. 2007. Robust object recognition with cortex-like mechanisms. *IEEE Trans. Pattern Anal. Mach. Intell.* 29(3):411–26

Simoncelli EP, Olshausen BA. 2001. Natural image statistics and neural representation. *Annu. Rev. Neurosci.* 24(1):1193–216

Simonyan K, Vedaldi A, Zisserman A. 2014. Deep inside convolutional networks: visualising image classification models and saliency maps. arXiv:1312.6034 [cs.CV]

Sugase Y, Yamane S, Ueno S, Kawano K. 1999. Global and fine information coded by single neurons in the temporal visual cortex. *Nature* 400(6747):869–73

Sutton RS, Barto AG. 1998. *Reinforcement Learning: An Introduction*, Vol. 1. Cambridge, MA: MIT Press

Sutskever I, Vinyals O, Le QVV. 2014. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* 27:3104–12

Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, et al. 2014. *Intriguing properties of neural networks*. Presented at Int. Conf. Learn. Represent., Apr. 14–16, Banff, Can. arXiv:1312.6199v4 [cs.CV]

Tanaka K. 1996. Inferotemporal cortex and object vision. *Annu. Rev. Neurosci.* 19:109–39

Tang H, Buia C, Madsen J, Anderson WS, Kreiman G. 2014. A role for recurrent processing in object completion: neurophysiological, psychophysical and computational evidence. arXiv:1409.2942 [q-bio.NC]

Tang Y, Srivastava N, Salakhutdinov RR. 2014. Learning generative models with visual attention. *Adv. Neural Inf. Process. Syst.* 27:1808–16

Tarr MJ. 1999. News on views: pandemonium revisited. *Nat. Neurosci.* 2:932–35

Tenenbaum JB, Griffiths TL, Kemp C. 2006. Theory-based Bayesian models of inductive learning and reasoning. *Trends Cogn. Sci.* 10(7):309–18

Tromans JM, Harris M, Stringer SM. 2011. A computational model of the development of separate representations of facial identity and expression in the primate visual system. *PLOS ONE* 6:e25616

Tsai C-Y, Cox DD. 2015. Measuring and understanding sensory representations within deep networks using a numerical optimization framework. arXiv:1502.04972 [cs.NE]

Venugopalan S, Xu H, Donahue J, Rohrbach M, Mooney R, Saenko K. 2015. Translating videos to natural language using deep recurrent neural networks. arXiv:1412.4729 [cs.CV]

von Helmholtz H. 1866. *Handbuch der physiologischen Optik: Mit 213 in den Text eingedruckten Holzschnitten und 11 Tafeln*, Vol. 9. Voss

Wallis G, Rolls ET. 1997. A model of invariant object recognition in the visual system. *Prog. Neurobiol.* 51:167–94

Werbos PJ. 1981. Applications of advances in nonlinear sensitivity analysis. In *Proceedings of the 10th IFIP Conference*, pp. 762–70

Yamins DL, Hong H, Cadieu CF, DiCarlo JJ. 2013. Hierarchical modular optimization of convolutional networks achieves representations similar to macaque IT and human ventral stream. *Adv. Neural Inf. Process. Syst.* 26:3093–101

Yamins DLK, Hong H, Cadieu CF, Solomon EA, Seibert D, DiCarlo JJ. 2014. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *PNAS* 111:8619–24

Yosinski J, Clune J, Nguyen A, Fuchs T, Lipson H. 2015. Understanding neural networks through deep visualization. arXiv:1506.06579 [cs.CV]

Yuille A, Kersten D. 2006. Vision as Bayesian inference: analysis by synthesis? *Trends Cogn. Sci.* 10(7):301–8

Zeiler MD, Fergus R. 2014. Visualizing and understanding convolutional networks. *Proc. 13th Eur. Conf. Comput. Vis., Sept. 6–12, Zurich*, pp. 818–833. New York: Springer

Zhou B, Khosla A, Lapedriza A, Oliva A, Torralba A. 2015. *Object detectors emerge in deep scene CNNs*. Presented at Int. Conf. Learn. Represent., May 7–9, San Diego. arXiv:1412.6856 [cs.CV]

RELATED RESOURCES

Bengio Y, Goodfellow I, Courville A. "Deep Learning" online book (in progress)
http://www.iro.umontreal.ca/~bengioy/dlbook/

Hinton G. 2012. Coursera course "Neural Networks for Machine Learning"
https://www.coursera.org/course/neuralnets

Ng A. Coursera course "Machine Learning"
https://www.coursera.org/course/ml

Nielson M. "Neural Networks and Deep Learning" online book (in progress)
http://neuralnetworksanddeeplearning.com/