

Backpropagation and the brain

Timothy P. Lillicrap¹, Adam Santoro, Luke Marris, Colin J. Akerman and Geoffrey Hinton

Abstract | During learning, the brain modifies synapses to improve behaviour. In the cortex, synapses are embedded within multilayered networks, making it difficult to determine the effect of an individual synaptic modification on the behaviour of the system. The backpropagation algorithm solves this problem in deep artificial neural networks, but historically it has been viewed as biologically problematic. Nonetheless, recent developments in neuroscience and the successes of artificial neural networks have reinvigorated interest in whether backpropagation offers insights for understanding learning in the cortex. The backpropagation algorithm learns quickly by computing synaptic updates using feedback connections to deliver error signals. Although feedback connections are ubiquitous in the cortex, it is difficult to see how they could deliver the error signals required by strict formulations of backpropagation. Here we build on past and recent developments to argue that feedback connections may instead induce neural activities whose differences can be used to locally approximate these signals and hence drive effective learning in deep networks in the brain.

The brain learns by modifying the synaptic connections between neurons^{1–5}. Although synaptic physiology helps explain the rules and processes behind individual modifications, it does not explain how individual modifications coordinate to achieve a network's goal. Since learning cannot be just a blind accumulation of myopic, synapse-specific events that do not consider downstream behavioural consequences, we need to uncover the principles orchestrating plasticity across whole networks if we are to understand learning in the brain.

Within machine learning, researchers study ways of coordinating synaptic updates to improve performance in artificial neural networks, without being constrained by biological reality. They start by defining the architecture of the neural network, which comprises the number of neurons and how they are connected. For example, investigators often use deep networks with many layers of neurons, since these architectures have proved to be very effective for many tasks. Next, researchers define an error function⁶ that quantifies how poorly the network is currently achieving its goals and then they search for learning algorithms

that compute synaptic changes that reduce the error (FIG. 1).

In machine learning, backpropagation of error ('backprop')^{7–10} is the algorithm most often used to train deep neural networks (BOX 1) and is the most successful learning procedure for these networks. Networks trained with backprop are at the heart of recent successes of machine learning, including state-of-the-art speech¹¹ and image recognition^{12,13}, as well as language translation¹⁴. Backprop also underpins recent progress in unsupervised learning problems such as image and speech generation^{15,16}, language modelling¹⁷ and other next-step prediction tasks¹⁸. In addition, combining backprop with reinforcement learning has given rise to significant advances in solving control problems, such as mastering Atari games¹⁹ and beating top human professionals in the games of Go^{20,21} and poker²².

Backprop uses error signals that are sent through feedback connections to adjust synapses and has classically been described in the supervised learning setting (that is, with explicit, externally provided targets). However, the brain appears to use its feedback connections for different purposes^{23,24} and is thought to learn in a

predominantly unsupervised fashion^{1,25–27}, building representations that make explicit the structure that is only implicit in the raw sensory input. It is natural to wonder, then, whether backprop has anything to tell us about learning in the brain^{25,28–30}.

Here we argue that in spite of these apparent differences, the brain has the capacity to implement the core principles underlying backprop. The main idea is that the brain could compute effective synaptic updates by using feedback connections to induce neuron activities whose locally computed differences encode backpropagation-like error signals. We link together a seemingly disparate set of learning algorithms into this framework, which we call 'neural gradient representation by activity differences' (NGRAD)^{9,27,31–41}. The NGRAD framework demonstrates that it is possible to embrace the core principles of backpropagation while sidestepping many of its problematic implementation requirements. These considerations may be relevant to any brain circuit that incorporates both feedforward and feedback connectivity. We nevertheless focus on the cortex, which is defined by its multilaminar structure and hierarchical organization, and so has long been viewed as exhibiting many of the architectural features associated with deep networks.

Credit assignment in networks

This article emphasizes the role of learning in the generation of adaptive behaviour. It should be acknowledged that brains undoubtedly have prior knowledge that has been optimized by evolution (that is, in the form of neural architectures and default connectivity strengths). Priors may ensure that only a limited amount of learning based on a relatively small amount of task error or feedback information is needed throughout an animal's lifetime to acquire all the skills the animal will exhibit. Nonetheless, although animals often display impressive behaviours from birth, they are also capable of extraordinary feats that could not have been tuned by evolution but instead require long bouts of learning. Some examples of such feats in humans are playing Go and chess; programming a computer or designing a video game; writing and playing a piano concerto; learning the vocabularies and grammars of multiple languages; recognizing

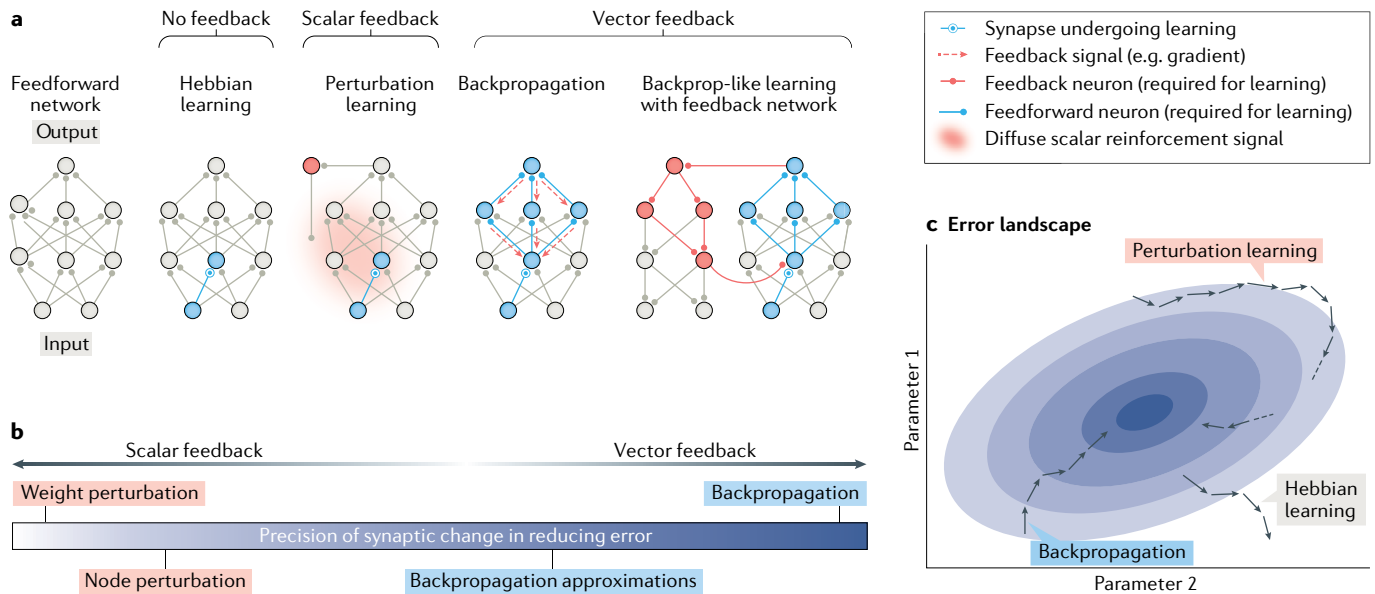


Fig. 1 | A spectrum of learning algorithms. a | Left to right: a neural network computes an output through a series of simple computational units. To improve its outputs for a task, it adjusts the synapses between these units. Simple Hebbian learning — which dictates that a synaptic connection should strengthen if a presynaptic neuron reliably contributes to a postsynaptic neuron's firing — cannot make meaningful changes to the blue synapse, because it does not consider this synapse's downstream effect on the network output. Perturbation methods measure the change in error caused by random perturbations to neural activities (node perturbation) or synapse strengths⁴⁴ (weight perturbation) and use this measured change as a global scalar reinforcement signal that controls whether a proposed perturbation is accepted or rejected. The backprop algorithm instead computes the synapse update required in order to most quickly reduce the error. In backprop, vector error signals are delivered backward along the original path of influence for a neuron. In the brain, vector feedback might be delivered in a variety of ways, including via a separate network.

b | Backpropagation and perturbation algorithms fall along a spectrum with respect to the specificity of the synaptic change they prescribe. **c** | Algorithms on this spectrum learn at different speeds. Without feedback, synaptic parameters wander randomly on the error surface. Scalar feedback does not require detailed feedback circuits, but it learns slowly. Since the same signal is used to inform learning at all synapses, the difficulty of deciding whether to strengthen or weaken a synapse scales with the number of synapses in the network: if millions of synapses are changed simultaneously, the effect of one synapse change is swamped by the noise created by all the other changes, and it takes millions of trials to average away this noise^{43–46}. The inverse scaling of learning speed with network size makes global reinforcement methods extremely slow, even for moderately sized neural networks. Precise vector feedback via backprop learns quickly. In real networks, it is not possible to make perfect use of the internal structure of the network to compute per-synapse changes, but the brain may have discovered ways to approximate the speed of backprop.

thousands of objects; and diagnosing a medical problem and performing vascular microsurgery. Recent work in machine learning suggests that these behaviours depend on powerful and general learning algorithms^{12,20}. Our interest here, then, is in characterizing such learning algorithms, and specifically, how they assign credit across multiple layers of neurons in the brain.

Correlative methods. Synaptic weights determine neural activity, neural activity determines the network's output, and the network's output determines the network's error. In artificial networks, we can therefore reduce the error slightly by making small changes in the synaptic weights. However, it is non-trivial to decide whether to increase or decrease any particular weight, because a synapse's strength does not influence the network's output directly; rather, it influences its immediate postsynaptic neurons, which then influence their postsynaptic neurons, and so on, all the way to the output of the network. The radius of the synapse's

influence — its projective field — rapidly expands, so the effect of changing the synapse strength depends on the strengths of many subsequent synapses in the network (for example, the blue connections spreading from the input layer in FIG. 1).

A conceptually simple way to decide whether to strengthen or weaken a synapse is to measure the effect of changing a synapse strength on the error. Such a measurement is easy to make in artificial networks. First, some input is injected into the network, and the network's baseline error is recorded. Next, noise is added to a particular synapse, and the same input is injected back into the network. Finally, one accepts the modified synaptic weight if the network's new error is less than the baseline error, and one rejects the modification if the new error is larger than the baseline error^{42–44}. This procedure can be implemented simply by a learning rule that broadcasts a global scalar representing the overall change in performance of the network (FIG. 1a). Suppose the performance

of a network is captured by an error function that computes the degree to which the network's outputs $[y_1, \dots, y_M]$ deviate from their target values $[t_1, \dots, t_M]$ — for example, via the squared error, $E = \frac{1}{2} \sum_i (y_i - t_i)^2$. To improve the error, we simply update a weight in the network W_{ij} via $\Delta W_{ij} = -\eta(E' - E)\xi_{ij}$, where η is a learning rate, E is the error computed before noise is added, and E' is the error computed after Gaussian noise $\xi_{ij} \sim \mathcal{N}(0, \sigma)$ is added to W_{ij} . Although this method works and is easy to understand, it is extremely inefficient to measure the error of the whole network in order to know how to change a single synapse.

If changes at some synapses have more of an effect on performance than changes at other synapses do, we can do a bit better by measuring the effects of making N different synaptic changes simultaneously (in parallel), but this does not really solve the efficiency issue, because we then require about N trials before we can reliably infer whether increasing any particular synapse strength will increase

or decrease the error^{44–46}. These kinds of ‘weight perturbation’ methods can be further improved by perturbing the outputs of neurons instead of weights^{42,47}. Such ‘node perturbation’ methods compute local derivatives of a neuron’s activity with respect to its own weights to speed up learning, but these methods are still very slow, and the performance gap increases as the network size increases⁴⁴. We think it is likely that the brain employs perturbation methods for some kinds of learning. However, it is striking that there has not yet been any successful application of these methods to training large, deep networks for difficult problems, such as classifying natural images of many different types of object.

The backprop algorithm addresses the efficiency issues present in perturbation methods by computing rather than measuring how a change in a synapse strength will affect the network’s error (BOX 1). This computation is possible because we have access to the exact causal relationship between the synapse strengths and the network’s output. By contrast, the causal relationship between the genotype and the phenotype generally depends on unknown aspects of the environment, so measuring the effects of genetic changes may be the only reasonable algorithm for evolution.

Backpropagation. Backprop computes how slightly changing each synapse strength would change the network’s error, using the chain rule of calculus. Moreover, it does this computation for all the synapse strengths at the same time and it requires only the same amount of computation as is needed for a forward propagation to pass through the network. Its key insight is to implement the chain rule of calculus using a recursive computation of ‘error signals’ (see BOX 1 for backprop’s algorithmic details). In a hierarchical, multilayer neural network, the error signals for the neurons in one layer are computed from the error signals in the layer above. Thus, error computations start in the final layer and flow backwards, leading to the notion of errors ‘backpropagating’ through the network. Once error signals have been computed for every neuron, the final output error can be reduced by changing the incoming weights of each neuron so as to push its postsynaptic activity in the direction specified by the error signal.

Backprop is often presented as requiring explicit output targets that are paired with corresponding input patterns. In fact, backprop’s recursive application of the chain rule provides a more general mechanism

for computing how changes in the activity of one part of a network affect the activities downstream. This mechanism is thus broadly applicable to credit assignment in multilayer networks. For simplicity, we follow the supervised learning paradigm, but note that a backpropagated signal does not need to be a difference between an output and a supervised target. The signal can also be a temporal difference error or a policy gradient in reinforcement learning^{19,46,48–50}, or a reconstruction or prediction error for an unsupervised algorithm (Supplementary Information). All of these can be self-constructed by an organism without reference to an external target.

An important empirical feature of backprop — and perhaps a key reason for its success — is its ability to quickly find good internal representations of inputs⁵¹ when training deep neural networks. Internal representations are not specified explicitly by the input or the output targets.

Instead, they must be discovered over the course of learning. Internal representations comprise useful building blocks — such as representations for edges, fragments of shapes, the semantic features of words and so forth — that the network’s intermediate layers use to efficiently code many different entities by using combinations of shared features. The distributed representation of input data as an activity vector of reusable, mix-and-match features allows networks to represent novel data as new combinations of familiar features. This allows the network to generalize to new data from the same distribution as the training data and in some cases to data that are outside this distribution.

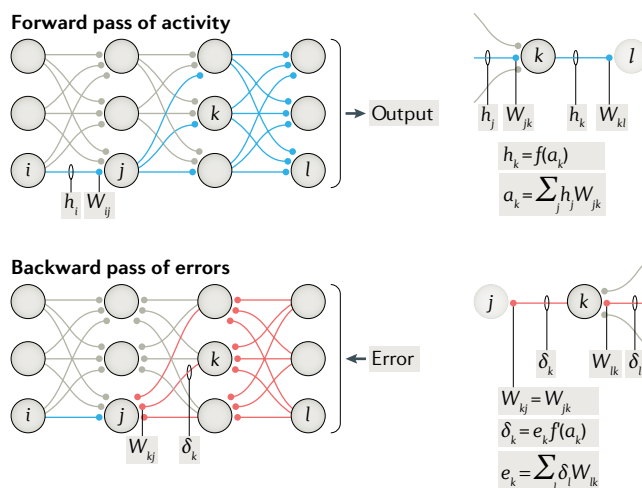
The backprop algorithm has two main features that are critical for its operation. These features are remarkably consistent with biological networks. The first feature is the prescription of synapse-specific changes. Synaptic plasticity mechanisms are widely accepted as exhibiting synapse specificity in biological networks undergoing learning^{5,52},

Box 1 | The backpropagation algorithm

The total incoming activity to neuron j (including its bias) is a_j , and the neuron’s output is $h_j = f(a_j)$ (see the figure). In this Box, we suppress the layer indices, which tell you both which neuron is referred to and, implicitly, the layer that it belongs to. Thus, W_{ij} is the synaptic weight that connects neuron i to neuron j . An error function computes the degree to which the network’s ultimate outputs (y_i) deviate from their target values (t_i). A common choice is the squared error: $E = \frac{1}{2} \sum_i (y_i - t_i)^2$. Backprop is a method for computing, for every weight, the gradient of the error at the current setting of all the weights. The simplest way to use this gradient is to change each weight in proportion to the negative of its gradient. For a non-output layer, the update for weight W_{ij} is

$$\Delta W_{ij} = -\eta \frac{\partial E}{\partial W_{ij}} = -\eta h_i \delta_j \text{ where } \delta_j = e_j f'(a_j) = \left(\sum_k \delta_k W_{jk} \right) f'(a_j)$$

The key insight behind the backprop algorithm is that the δ terms, sometimes called ‘error signals’, can be computed recursively via the chain rule (rather than measured by injecting noise and observing the outcomes). At the output layer, $\delta_i = y_i - t_i$. In all other layers, δ_j is computed from all of the δ_k in the layer above, so the error signals flow backward through the network, starting with the error signals at the output units that are the derivatives of the error function (see the figure).



such as in spike-timing-dependent plasticity^{2–4}, where the specific presynaptic and postsynaptic spike trains influence the particular synaptic modifications between two neurons. The second feature is the requirement for feedback connections that deliver error information to neurons deep within the network, so that they can compute the necessary synaptic changes. We refer to a learning algorithm as ‘backprop-like’ if it optimizes a downstream objective by using detailed vector feedback to help prescribe synapse-specific updates. Feedback connections between areas permeate every network of the cortex and, critically, can act to modulate the spiking of ‘feedforward’ neurons. These feedback connections can take the form of direct ‘top-down’ cortico-cortical connections from higher to lower cortical processing areas, like those that exist between V2 and V1 within the visual system²³. Equally, feedback connections could be routed through the thalamus, via cortico-thalamo-cortical loops that can deliver higher-order information to cortical regions and individual neurons that receive lower-order information^{53–55}.

It is not clear in detail what role feedback connections play in cortical computations, so we cannot say that the cortex employs backprop-like learning. However, if feedback connections modulate spiking, and spiking determines the adaptation of synapse strengths, the information carried by the

feedback connections must clearly influence learning! Backpropagation can be viewed as a very good candidate for what form this influence should take if the cortex is to be an efficient learning machine. This still leaves open the details of exactly how the feedback connections could approximate backpropagation, and there are good arguments (discussed below)^{25,28–30,56,57} to suggest that some of the most obvious implementations are biologically unrealistic. But these arguments do not mean that backprop should be abandoned as a guide to understanding learning in the brain^{57–59}; its core ideas — that neural networks can learn by computing synapse-specific changes using information delivered in an intricate web of feedback connections — have now proved to be so powerful in so many different applications that we need to investigate less obvious implementations.

It should be noted at the outset that the cortex differs from artificial neural networks in many significant ways. For example, there is no straightforward mapping between layers in an artificial network and the layers (that is, layers 1–6) or areas (for example, V1 and V2) in the cortex. Moreover, cell types, connectivity and gene expression differ between different areas of the cortex^{60–63}, and cortical areas send and receive different kinds of connections to and from various cortical and subcortical areas, among many other differences. Nevertheless, there are also overarching

similarities across cortical areas, such as the prevalence of microcolumns^{64,65} and canonical connectivity patterns both within and between cortical areas^{34,66}, that suggest common computations⁶⁷, and we think that understanding these common computations will be useful. Specifically, we imagine that an algorithm akin to backprop is required in order to coordinate synaptic changes.

Backprop in the brain?

There is no direct evidence that the brain uses a backprop-like algorithm for learning. Past work has shown, however, that backprop-trained models can account for observed neural responses, such as the response properties of neurons in the posterior parietal cortex⁶⁸ and primary motor cortex⁶⁹. A new wave of evidence from neuroscience modelling of the visual cortex is carrying this trend forward^{58,70–72}. This work has revealed that multilayer models trained with backprop to classify objects tend to perform better than other models at matching the representations along the visual ventral stream in primates (FIG. 2). Models that are not trained with backprop (such as bio-inspired models using Gabor filters^{73,74}, or networks that use non-backprop optimization⁵⁹) do not perform as well as backprop-optimized networks, and their representations do not match those in the inferior temporal cortex as well as do the representations discovered by backprop-trained models.

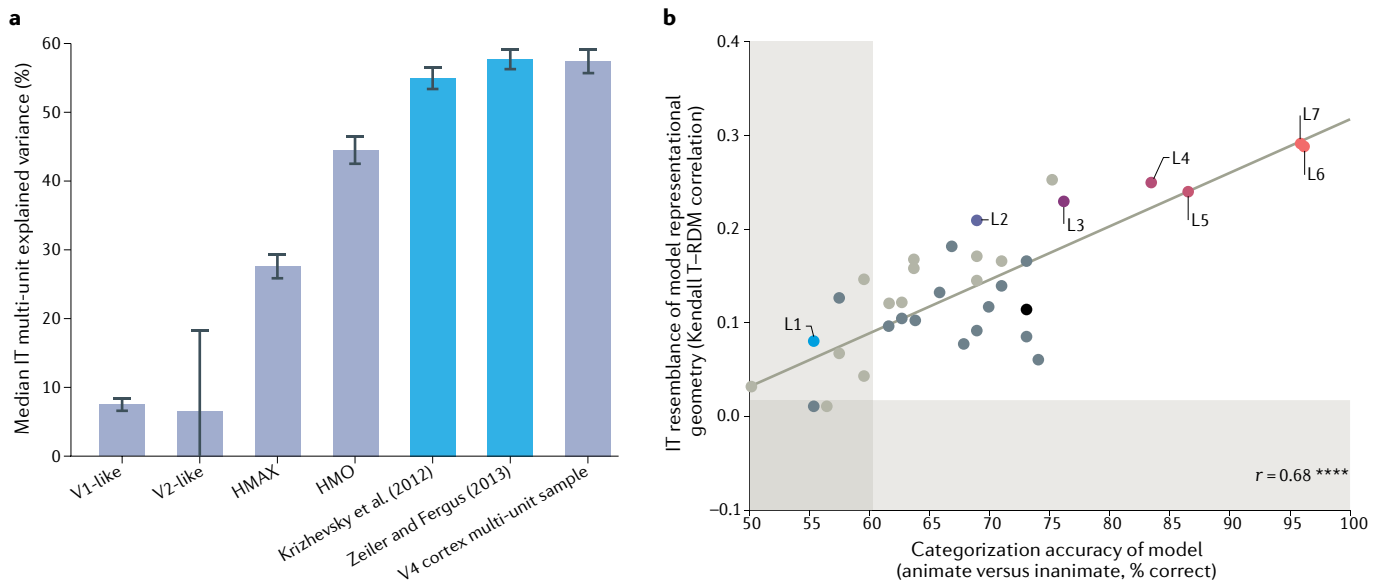


Fig. 2 | Comparison of backprop-trained networks with neural responses in visual ventral cortex. **a** | Cadieu et al.⁵⁸ showed that backprop-trained models^{12,170} (blue) explain inferior temporal cortex (IT) multi-unit responses better than other models do (grey). **b** | Khaligh-Razavi and Kriegeskorte⁷⁰ showed that models with better classification performance more closely resemble IT representations; each unlabelled dot corresponds to a model,

whereas the coloured dots L1–L7 correspond to successively deeper network layers. Moreover, neurons in deeper layers within the backprop-trained network have representations that are more similar to those in IT cortex than are neurons in earlier layers of the network. Part **a** adapted from REF.⁵⁸, CC-BY-4.0 (<https://creativecommons.org/licenses/by/4.0/>). Part **b** adapted from REF.⁷⁰, CC-BY-4.0 (<https://creativecommons.org/licenses/by/4.0/>).

The representational matches from networks trained with backprop are by no means perfect, and recent work has indicated that current models do not explain some aspects of human object classification⁷⁵. Nevertheless, the tendency for backprop-trained models to better match observed neural responses appears to be widespread, with related work demonstrating that auditory neuron responses are also predicted better by multilayer networks trained by backprop than by other models⁷⁶. This does not prove that the cortex learns via backprop-like mechanisms, but as Cadieu et al.⁵⁸ state, it shows that the “possibility cannot be ruled out merely on representational grounds.”

Performance and representational matches can never on their own establish that backprop-like mechanisms are employed by the brain. The proliferation of computing power and the discovery of better priors might one day allow researchers to train high-performing networks for complex tasks using slower learning algorithms that do not make use of vector feedback. What we can say is that the practicality and efficiency of backprop are at least suggestive that the brain ought to harness detailed, error-driven feedback for learning. To our knowledge, no one in the machine-learning community has been able to train high-performing deep networks on difficult tasks such as classifying the objects in ImageNet photos using any algorithm other than backprop. In particular, attempts to rely on algorithms that use only scalar feedback signals, such as genetic algorithms⁷⁷ or REINFORCE⁴⁶, have failed by a wide margin to reach backprop's level of performance.

In addition to producing models that better match the representations observed in the brain, backprop-trained deep networks can also help explain the size and timing of receptive field changes in perceptual learning^{72,78}, as well as the stage-like transitions observed during some types of learning in animals and humans⁷⁸. Other work has demonstrated that neurons in layers 2 and 3 of cortex appear to compute detailed mismatches between actual and predicted sensory events⁷⁹, and that the neural dynamics at successive stages of the visual cortex are consistent with hierarchical error signals⁸⁰. These findings are consistent with the hypothesis that feedback connections in the cortex drive learning across multiple layers of representation. In the penultimate section, we review recently described neural mechanisms that offer additional support for this hypothesis.

Problems with backprop

Backprop's computations. Although there is mounting evidence that multilayer networks trained with backprop can help explain neural data, there are difficult questions concerning how backprop-like learning could be implemented in the cortex. Equation 1 gives the synaptic updates prescribed by backprop, in matrix/vector notation:

$$\Delta W_l = -\eta \frac{\partial E}{\partial W_l} = -\eta \delta_l \mathbf{h}_{l-1}^T \quad (1)$$

where

$$\delta_l = \mathbf{e}_l \circ \mathbf{f}'(\mathbf{a}_l) = (W_{l+1}^T \delta_{l+1}) \circ \mathbf{f}'(\mathbf{a}_l).$$

Bold symbols are vectors, \cdot^T is the transpose operation, \circ is element-wise multiplication and $\mathbf{h}_l = \mathbf{f}(\mathbf{a}_l)$. Since this equation does not require indices for individual neurons (as in BOX 1), we use subscripts to denote the layers. The equation says that the presynaptic weights, W_p in layer l are updated according to the product of the error signals δ_l and the presynaptic activities, which are the outputs of the previous layer \mathbf{h}_{l-1} . The error signals δ_l are computed by multiplying the error signals from the layer above, δ_{l+1} , by the transpose of the postsynaptic weights W_{l+1}^T and then multiplying by the derivative of the activity function $\mathbf{f}'(\mathbf{a}_l)$. It is perhaps worth noting that, when presented in the form, $\Delta W_l = -\eta(\mathbf{e}_l \circ \mathbf{f}'(\mathbf{a}_l))\mathbf{h}_{l-1}^T$, the update can be seen as a local Hebbian-like rule — where the postsynaptic activity is replaced by $\mathbf{f}'(\mathbf{a}_l)$ — that is modulated by a third factor⁸¹, \mathbf{e}_p , which is computed via feedback connections. In the subsequent sections, we refer to Eq. 1 as we outline three major difficulties in implementing backprop in biological circuits.

Backprop demands synaptic symmetry in the forward and backward paths.

A naive implementation of backprop requires the delivery of error signals through feedback connections that have exactly the same strength as the feedforward connections. In Eq. 1, the errors, δ_{l+1} , travel along feedback weights, W_{l+1}^T , that are symmetric to their feedforward counterparts. On a computer, the backprop algorithm sends error information backward using a set of error derivative variables that are distinct from the activity variables used in the forward pass. Soon after backprop was introduced, it was suggested that, in the brain, error information could be delivered by a distinct ‘error delivery network’, with each neuron in this

backward network carrying update information for a matched neuron in the ‘forward’ network^{82,83}. Early work on how backprop might be implemented in the brain also explored the idea that error signals might travel in a retrograde direction along the feedforward axons^{84,85} (with, somehow, strengths equivalent to those in the anterograde direction). But this idea has been abandoned, because retrograde communication is much too slow to support backprop^{86–88}. The need to have the same weight on two different connections has been called the ‘weight transport’ problem^{25,28,82} and would still be a major hurdle when a second error delivery network is posited. Previous work has considered the use of symmetric learning rules to establish and maintain this weight symmetry^{28,34,39,82,89–91}, but the cortex does not exhibit the requisite point-to-point reciprocal connectivity.

Fortunately, recent work has demonstrated that this symmetry is unnecessary^{57,92–99}. Remarkably, networks with fixed random feedback weights learn to approximately align their feedforward synaptic weights to their feedback weights. In a display of neural pragmatism, fake error derivatives computed using the random feedback weights cause updates to the feedforward weights that make the true error derivatives closer to the fake derivatives. This surprising phenomenon, called ‘feedback alignment’, suggests that feedback connections do not need to be symmetric to their feedforward counterparts in order to deliver information that can be used for fast and effective weight updates⁵⁷. Feedback alignment thus offered early evidence that the kind of precise symmetry employed by backprop is not always required for effective learning. Random feedback weights may be insufficient for learning in deeper networks¹⁰⁰, but subsequent work has demonstrated that simple learning mechanisms could shape the backward pathway to ensure that effective feedback is delivered^{99,101}, even in very deep networks trained on a complex task¹⁰¹.

Error signals are signed and potentially extreme-valued.

In backprop, the information sent backwards through a network to inform updates is conveyed in the form of signed error signals, δ . During training these signals often vary across many orders of magnitude, referred to as the phenomena of exploding and vanishing gradients¹⁰². Although evidence exists for signed error delivery in apparently

single-layered structures such as the cerebellum^{103,104}, feedback of signed errors in deep networks, such as the cortex, appears problematic. A separate set of ‘error neurons’^{82,83,101}, as suggested in the previous section, could be used to implement a backward pass, with firing rates above a certain value communicating positive error, and rates below this value conveying negative error. However, the network must then perform complicated bookkeeping to coherently integrate the signed information coming from multiple feedback connections and to continue the propagation of this information across multiple layers. When thinking about how backprop could be implemented in the brain, this problem is still unsolved, but we explore a potential solution that avoids the propagation of error signals altogether.

Feedback in brains alters neural activity.

In error backpropagation, feedback connections deliver error signals that do not influence the activity states of neurons produced by feedforward propagation. Rather, the information delivered via δ only influences synaptic updates. The role of feedback connections in the brain appears to be fundamentally different. In the cortex, for example, these connections influence the neural activities produced by feedforward propagation and are thought to serve a number of functional roles. For example, top-down control through feedback connections has a well-established link with gain control — that is, the enhancement or suppression of neural responses depending on, for example, attention to a particular feature in the visual field^{65,105–117}. Interestingly, feedback connections in the cortex can also drive activity, rather than just modulate or enable it.

This idea has been corroborated by experiments showing that conduction velocities and the types of excitatory synaptic communication are often comparable between feedback and feedforward axons^{118,119}, evidence that early visual areas are activated during visual mental imagery^{120,121}, evidence that top-down feedback is actively involved in bottom-up processing^{122,123}, and lesion experiments demonstrating a cessation of activity following the inactivation of feedback^{124,125}. The consequences of this idea appear in a number of proposals, such as reverse hierarchy theory¹²⁶ and hierarchical Bayesian inference for perception^{36,127–130}, both of which draw inspiration from Helmholtz’s view of perception as unconscious inference¹³¹.

Although neuroscientists have proposed several functions for feedback connections, they have rarely considered the possibility that these connections’ primary function is to drive learning — for example, see REF.²³. There is, however, a long but lesser known history of work in the machine-learning literature that has used feedback connections to alter the activities produced during feedforward propagation (unlike backprop) and then used these alterations to guide learning^{9,27,31–35,37–41}. Here we suggest that the most important role for feedback connections is to make alterations in neural activities to convey the information required for effective multilayer learning; that is, the activity alterations induced by feedback dictate synaptic weight changes that improve feedforward processing in deep networks.

The NGRAD hypothesis

Using neural activity differences to

encode errors. Many proposed learning mechanisms have used differences in activity states to drive synaptic changes, rather than propagating or diffusing signals that represent gradients explicitly. Around the time that backprop entered the mainstream⁸, several neural-network learning algorithms — including the Boltzmann machine^{35,132} — explored this idea by using temporal differences between activities inferred during two phases of propagation as a means to compute updates. Several recently introduced approaches have instead used activity differences between sets of neurons in a local circuit¹³³ or between different compartments within a neuron^{134,135}.

We call learning mechanisms that use differences in activity states to drive synaptic changes NGRADs. The idea that the cortex uses an NGRAD to perform an approximation to gradient descent will be called the NGRAD hypothesis. The main attraction of this hypothesis is that it avoids the need to propagate two quite different types of quantity: activities and error derivatives. Instead, NGRADs are based on the idea that higher-level activities — coming from a target, another modality, or a larger spatial or temporal context — can nudge^{27,34,134} lower-level activities towards values that are more consistent with the higher-level activity or a desired output. Moreover, the induced change in lower-level activities can then be used to compute backprop-like weight updates using only locally available signals. Thus, the fundamental idea is that top-down-driven activities drive learning without carrying explicit error information between layers.

One concrete example of such an algorithm is GeneRec³⁴, which combines insights from the Boltzmann machine algorithm¹³² and the recirculation algorithm²⁷. GeneRec trains multilayer recurrent networks as follows: in a ‘negative phase’, the input is provided, and recurrent activities are allowed to settle to equilibrium. In a ‘positive phase’, input is provided to the network while the output neurons are clamped to, or nudged towards, their target values, and activities are again allowed to settle to equilibrium. GeneRec’s learning rule is simple and local: each synaptic weight change should be proportional to the difference between the product of the presynaptic and postsynaptic activities from the positive and negative phases.

A number of other algorithms, including contrastive Hebbian learning³⁷, the Almeida/Pineda algorithms^{31–33} and the wake–sleep algorithm in the Helmholtz machine^{36,130}, use a similar logic as the backbone for learning. The most important contribution for our purposes is their use of locally available information — activity states at different points in time or across different spatial compartments — to capture the error information that guides learning. New work on the biological plausibility of backprop has returned to these ideas^{133,134,136,137}: for example, the recently introduced equilibrium propagation¹³⁷ employs the same essential elements as GeneRec and contrastive Hebbian learning. Furthermore, several models^{133,134} have examined how NGRAD learning might be achieved without a separate negative phase. These models use ideas from predictive coding to make effective updates using locally computed differences across neurons or their compartments, rather than across time. In spite of the fact that NGRADs compute error vectors locally within a layer, rather than transmitting them across layers, as in backprop and feedback alignment⁵⁷, many algorithms in this class can be shown to make updates that approximately follow (and in some cases, exactly follow) the gradient computed by backprop^{39,133,135,136}.

To gain intuition into how activity differences computed within a layer can be used to guide learning, we examine a simpler proposal put forward at the first Neural Information Processing Systems workshop on deep learning¹³⁸ (Supplementary Fig. 1) and later developed by Lee et al.⁴¹. Fundamental to the proposal are the use of auto-encoders^{35,139} to send top-down signals to earlier layer activations and the use of the induced differences to make weight updates. In the following

sections, we describe auto-encoders and then show how they can be used as the basis of a deep-learning algorithm motivated by biological constraints.

Auto-encoders. We begin by developing the idea of an auto-encoder^{35,139}, which is a network that aims to reconstruct its own input. The simplest auto-encoder takes a vector input \mathbf{x} , uses this input vector to produce an activity vector in its hidden layer via a weight matrix \mathbf{W} and nonlinearity, $\mathbf{h} = f(\mathbf{x}; \mathbf{W}) = \sigma(\mathbf{W}\mathbf{x})$, and then uses the hidden activity vector to reconstruct an approximation to the input vector via a backward weight matrix: $\hat{\mathbf{x}} = g(\mathbf{h}; \mathbf{B}) = \sigma(\mathbf{B}\mathbf{h})$. Auto-encoders can be trained without requiring explicit labels, because the difference between the original input and the reconstruction, $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$, is used as the error to drive learning. This difference can be computed locally by neurons in the input layer and used to adjust the weights from the hidden layer to the input layer. The input-to-hidden weights might also be adjusted without requiring backprop by using the recirculation

algorithm^{27,140}. Another view of this idea is that the feedback connections in an auto-encoder learn an approximate inverse function $g(\cdot; \mathbf{B})$ that transforms the hidden activity back to the associated point in the input space, so that $g(f(\mathbf{x}; \mathbf{W}); \mathbf{B}) \approx \mathbf{x}$. Many applications of auto-encoders use hidden layers that are much smaller than the input, but when learning a precise inverse, we may wish them to have roughly the same number of units. Most importantly for error assignment in the NGRAD framework, auto-encoders can be used to propagate detailed activity targets at higher layers backwards to provide targets for earlier layers, which can in turn be used to compute local differences that are appropriate for driving learning.

Target propagation. FIGURE 3a sketches target propagation^{141,142}, the essential idea behind using a stack of auto-encoders for deep learning. We propagate activity forward through successive layers of a network to produce a predicted output. Then we propagate an output target backwards by means of inverse functions (that is, via

feedback connections) that are learned through layer-wise auto-encoding of the forward layers. This backward-propagated target induces hidden-activity targets that should have been realized by the network. In other words, if the network had achieved these hidden activities during feedforward propagation, it would have produced the correct output. The direction in the activity space between the feedforward activity and the feedback activity indicates the direction in which the neurons' activities should move in order to improve performance on the data. Learning proceeds by updating the forward weights to minimize these local layer-wise activity differences, and it can be shown that under certain conditions the updates computed using these layer-wise activity differences approximate those that would have been prescribed by backprop.

Formally, suppose that we have a stack of auto-encoders in which the hidden units of one auto-encoder are the input units for the next auto-encoder: we have forward functions, $\mathbf{h}_l = f(\mathbf{h}_{l-1}; \mathbf{W}_l) = \sigma(\mathbf{W}_l \mathbf{h}_{l-1})$ for layers $l \in \{1, \dots, L\}$ and backward functions $\tilde{\mathbf{h}}_l = g(\mathbf{h}_{l+1}; \mathbf{B}_{l+1}) = \sigma(\mathbf{B}_{l+1} \mathbf{h}_{l+1})$ for layers

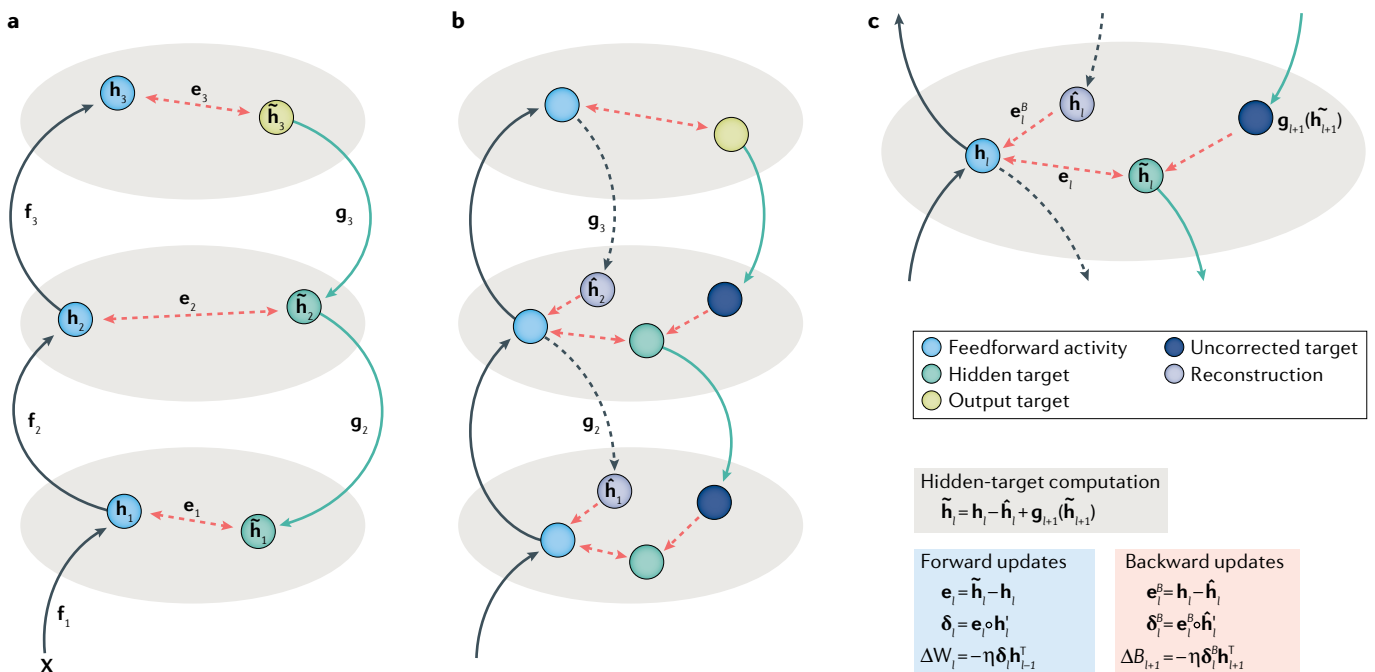


Fig. 3 | Target propagation algorithms. **a** | Schematic of target propagation that uses perfect inverses, $\mathbf{g}_l(\cdot) = \mathbf{f}_l^{-1}(\cdot)$, at each layer. For illustration, high-dimensional activity vectors at each layer are represented as points in a 2D space. Local layer-wise errors, $\mathbf{e}_l = \tilde{\mathbf{h}}_l - \mathbf{h}_l$, are computed between the forward-pass activities (\mathbf{h}_l ; blue) and the top-level ($\tilde{\mathbf{h}}_3$; yellow) and induced ($\tilde{\mathbf{h}}_l$; green) targets. Synaptic weights, \mathbf{W}_l , associated with the forward mapping $\mathbf{f}_l(\cdot)$ are updated in order to move the forward activity vectors closer to the targets. **b** | Difference target propagation helps correct for the fact that the feedback connections may not implement perfect inverses. For each layer, \mathbf{h}_l , we compute a reconstruction, $\hat{\mathbf{h}}_l$, from the layer immediately above via $\mathbf{g}_{l+1}(\cdot)$. Then, to compensate for imperfections in the auto-encoders, we add the

reconstruction error, $\mathbf{e}_l^B = \mathbf{h}_l - \hat{\mathbf{h}}_l$, to the uncorrected target $\mathbf{g}_{l+1}(\tilde{\mathbf{h}}_{l+1})$ (dark blue), computed from the layer above in the backward pass. **c** | Schematic for a single layer of difference target propagation. Forward synaptic weights, \mathbf{W}_l , are updated in order to move the forward-pass hidden activity closer to the corrected hidden target. Note that the light purple, dark blue and green circles do not represent separate sets of neurons, but rather different stages of processing performed in the same neurons. Backward synaptic weights, \mathbf{B}_{l+1} , are updated in order to reduce auto-encoder reconstruction errors. The hidden target, $\tilde{\mathbf{h}}_l$, is computed as a mixture of the bottom-up activity with top-down feedback. Crucially, errors are computed with signals local to the neurons in each layer, rather than propagated between layers as in backprop.

$l \in \{L-1, \dots, 1\}$. For notational convenience, we define the input to the network to be $\mathbf{x} = \mathbf{h}_0$, the output of the network to be $\mathbf{y} = \mathbf{h}_L$ and the output target to be $\mathbf{t} = \mathbf{h}_L$. Also for notational convenience, we absorb the weight matrices into the subscript and write $\mathbf{f}_l(\mathbf{h}_{l-1}) = \mathbf{f}(\mathbf{h}_{l-1}; W_l)$ and $\mathbf{g}_{l+1}(\mathbf{h}_{l+1}) = \mathbf{g}(\mathbf{h}_{l+1}; B_{l+1})$. Suppose further that the auto-encoders are perfect, so that we have exact inverse functions that map back from each higher layer to the layer below — that is, $\mathbf{g}_l(\cdot) = \mathbf{f}_{l-1}^{-1}(\cdot)$, so that $\mathbf{g}_l(\mathbf{f}_l(\mathbf{h}_{l-1})) = \mathbf{h}_{l-1}$. After the forward and backward passes are complete, and assuming that one is computed after the next, the temporal difference between the feedforward activity and the feedback activity target, $\mathbf{e}_l = \tilde{\mathbf{h}}_l - \mathbf{h}_l$, drives plasticity via $\Delta W_l = -\eta \delta_l \mathbf{h}_{l-1}^\top$, where $\delta_l = \mathbf{e}_l \circ \mathbf{h}_l'$, and \mathbf{h}_l' is the derivative of the activation function in layer l . This idea of using auto-encoders to induce targets for deep updates is elegant, but it is problematic in practice^{41,100,142}, perhaps most obviously because it may be impossible to obtain perfect inverses.

Difference target propagation.

We described target propagation above as using perfect auto-encoders to convey targets to earlier layers. This constraint is unrealistic but can be fixed by training the backward weights. During the forward pass, we try to reconstruct neural activity from the activity in the subsequent layer: $\hat{\mathbf{h}}_l = \mathbf{g}_{l+1}(\mathbf{h}_{l+1})$ (shown as the light purple dots in FIG. 3b). The backward path auto-encoders thus induce layer-wise errors, $\mathbf{e}_l^B = (\mathbf{h}_l - \hat{\mathbf{h}}_l)$, which we use to update the feedback weights via $\Delta B_{l+1} = -\eta \delta_l^B \mathbf{h}_{l+1}^\top$, where $\delta_l^B = \mathbf{e}_l^B \cdot \mathbf{h}_l$, so that \mathbf{g}_{l+1} is moved closer to an approximate inverse for \mathbf{f}_{l+1} . In this sense, the circuit learns to learn, a phenomenon common to many proposed approximations for backprop^{34,90,92,93,101}. We then send the modified target $\tilde{\mathbf{h}}_{l+1}$ at level $l+1$ backward through these approximate inverses and use the result to make a linear correction to the target at level l : $\tilde{\mathbf{h}}_l = \mathbf{h}_l - \hat{\mathbf{h}}_l + \mathbf{g}_l(\tilde{\mathbf{h}}_{l+1})$, shown by the green dots in FIG. 3b,c. Under certain assumptions⁴¹, this correction allows the auto-encoders to perform perfectly for this particular input. Finally, we use these corrected targets to update the forward weights as before: $\Delta W_l = -\eta \delta_l \mathbf{h}_{l-1}^\top$. This learning procedure is called difference target propagation (DTP)⁴¹ and is shown along with the layer-wise weight updates in FIG. 3b,c.

DTP effectively trains multilayer neural networks on some image classification tasks⁴¹, and it learns in a fraction of the time required by algorithms that use weight

or node perturbation to update weights. The performance of algorithms like DTP is still being explored on more challenging datasets and more complex architectures. Recent work has shown that straightforward implementations of DTP do not perform as well as backprop on the ImageNet classification task with large, convolutional networks¹⁰⁰. The DTP algorithm also does not address questions of online learning or how the forward and backward pathways could communicate in biological circuits. Nevertheless, the algorithm provides a compelling example of how locally generated activity differences can be used to drive learning updates for multilayer networks, and recent work has suggested avenues for recovering performance for large-scale tasks^{97,98,101}.

We have emphasized algorithms that send the same kind of signal in both the forward and backward directions and that use activity differences local to a layer to compute errors. But it is possible that the

brain employs approaches that are closer in spirit to backprop. One may conceive of algorithms wherein neurons switch between propagating ‘feature’ information forward and errors backward, although we are not aware of evidence for the kind of fast switching between modes that ought to be induced. Another idea would be to use a second set of specially designated neurons to carry errors backward across multiple layers and deliver them to the forward pathway without interfering with its feature processing^{83,99,101}. Both of these approaches require that signed information be carried backwards across multiple layers via unsigned spiking activity. We are not aware of effective solutions to this issue, but these ideas present interesting alternatives to NGRADs that should not be ignored as we seek to understand how multilayer credit assignment might be implemented in neural circuits. No existing algorithm for multilayer credit assignment can be straightforwardly squared with what we know about the

Glossary

Auto-encoders

Networks showing unsupervised learning in which the target is the input itself. One application of auto-encoding is the training of feedback connections to coherently carry ‘targets’ to earlier layers.

Backpropagation of error (backprop)

An algorithm for explicitly computing the changes to prescribe to synapses in deep networks in order to improve performance. It involves the flow of error signals through feedback connections from the output of the network towards the input.

Credit assignment

Determination of the degree to which a particular parameter, such as a synaptic weight, contributes to the magnitude of the error signal.

Deep learning

Learning in networks that consist of hierarchical stacks, or layers, of neurons. Deep learning is especially difficult because of the difficulty inherent in assigning credit to a vast number of synapses situated deep within the network.

Error function

An explicit quantitative measure for determining the quality of a network’s output. It is also frequently called a loss or objective function.

Error signals

Contribution to the error by the activities of neurons situated closer to the output. In backpropagation, these signals are sent backward through the network in order to inform learning.

ImageNet

A large dataset of images with their corresponding word labels. The task associated with the dataset is to guess the correct label for each image. ImageNet has become a de facto standard for measuring the strength of deep-learning algorithms and architectures.

Internal representations

Hidden activity of a network that represents the network’s input data. ‘Useful’ representations tend to be those that efficiently code for redundant features of the input data and lead to good generalization, such as the existence of oriented edges in handwritten digits.

Learning

The modification of network parameters, such as synaptic weights, to enable better performance according to some measure, such as an error function.

Reinforcement learning

Learning in an interactive trial-and-error loop, whereby an agent acts stochastically in an environment and uses the correlations between actions and the accumulated scalar rewards to improve performance.

Supervised learning

Learning in which the error function involves an explicit target. The target tends to contain information that is unavailable to the network, such as ground truth labels.

Target

The desired output of a network, given some input. Deviation from the target is quantified with an error function.

Unsupervised learning

Learning in which the error function does not involve a separate output target. Instead, errors are computed using other information readily available to the network, such as the input itself or the next observation in a sequence.

Weights

Network parameters that determine the strength of neuron–neuron connections. A presynaptic neuron connected to a postsynaptic neuron with a high weight will greatly influence the activity of the postsynaptic neurons, and vice versa.

neurophysiology of the cortex. But aspects of the algorithms explored here may help us form the next round of empirical inquiry.

Implementation

Existing NGRADs may offer high-level insights into how the brain could approximate backprop, but there are many questions about how such algorithms could be implemented in neural tissue. To function in neural circuits, NGRADs need to be able to do the following: coordinate interactions between feedforward and feedback pathways, compute differences between patterns of neural activities and use this difference to make appropriate synaptic updates. It is not yet clear in detail how biological circuits could support these operations, but recent empirical studies present an expanding set of potential solutions to these implementation requirements (FIG. 4).

On a computer it is trivial to compute and store memories of separate feedforward and feedback passes in order to take differences and then use these for learning. There have been several proposals for how the same might be achieved in neural tissue. One of the earliest, proposed for backprop, was to employ a second set of ‘error’ neurons that could act as the feedback variables^{82,83}. The same idea could work for DTP, but there is no evidence for entirely segregated feedforward neurons in the cortex that are unaffected by feedback activity. Feedback tends to be area-wise reciprocal in the cortex, so that if area A sends forward connections to area B, it will tend to receive feedback from neurons in area B^{54,66}. If, as it appears, feedforward and feedback paths in the brain ‘reuse’ the same neurons, this has important implications for how backprop-like learning might be achieved in the cortex. Naively, sharing the same neurons requires some form of strong time-multiplexing, in which the forward pass occurs and then is replaced by a backward pass. Again, there is no direct support for this kind of staggered activity — in the brain both feedforward and feedback pathways appear to be active and interacting, at least to some extent, simultaneously.

When thinking about biological implementations of backprop, it is common to assume that feedforward and feedback synapses have similar functional effects on a neuron^{34,37}. If this assumption is correct, something like time-multiplexing would be required in order to compute the differences needed for learning. In reality, however, biological neurons are known to contain functionally and anatomically distinct

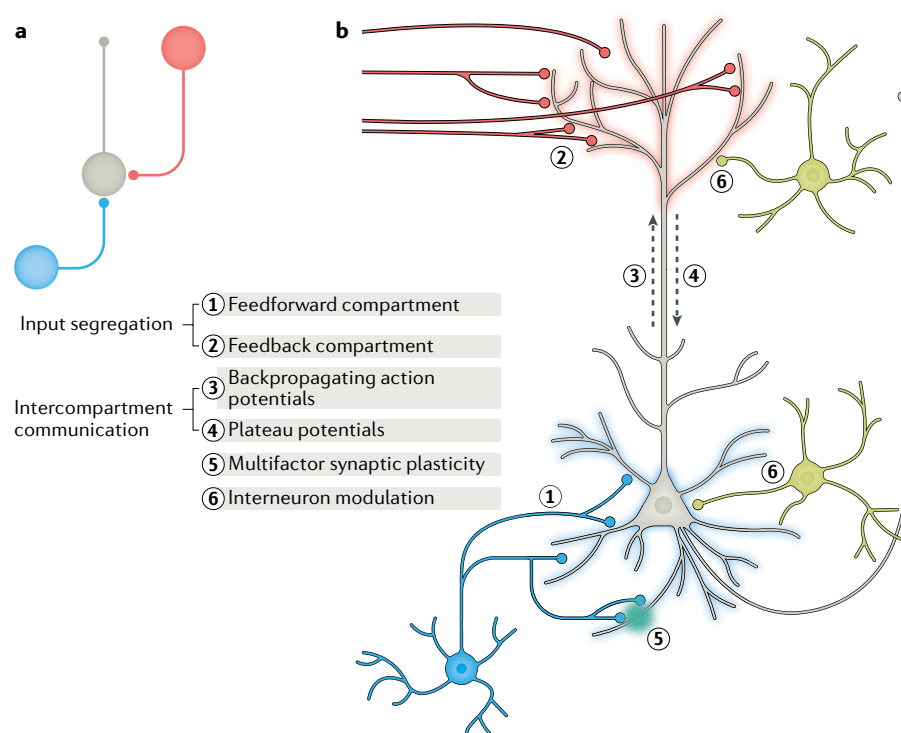


Fig. 4 | Empirical findings suggest new ideas for how backprop-like learning might be approximated by the brain. a | When backprop was first published, a neuron (grey cell) was typically conceived of, and modelled, as a single voltage compartment into which feedforward signals (blue; for example, from a lower-order cortical area) and feedback signals (red; for example, from a higher-order cortical area) would arrive undifferentiated. **b** | A contemporary schematic of a cortical pyramidal neuron (grey cell). Feedforward (1) and feedback (2) inputs are thought to be treated differently. They arrive at different compartments of the cell (for example, the basal and apical dendrites, respectively) and may be electrotonically segregated. Compartments can communicate selectively via backpropagating action potentials that are triggered by spikes in the soma and via calcium-spike-induced plateau potentials generated in the apical dendrite (3 and 4). Plasticity in one compartment may depend on both local synaptic events and events triggered in another compartment (5). For example, ‘forward’ basal synaptic plasticity may be altered by the arrival of apically generated plateau potentials. Finally, local inhibitory neurons (yellow cells) can regulate the communication between the subcellular compartments and can themselves be differentially recruited by higher-order inputs, and thus can modulate the interactions between the forward and backward pathways (6).

compartments, and activity arriving at these different subcellular domains can have varying effects on the cell.

For example, apical tuft dendrites of layer 5 cortical pyramidal neurons are electrotonically isolated from somatic and basal compartments. The apical dendritic compartment can receive feedback connections from higher cortical areas^{143,144}, or via higher-order thalamic nuclei^{145,146} and can act as semi-independent reservoirs that only communicate with the somatic compartment under certain conditions¹⁴⁷. In addition, ‘feedforward’ connections are thought to predominantly target the basal dendrites of pyramidal neurons, which act as functionally distinct regions that probably have different plasticity rules^{147–152}.

Neurocomputational models have begun to employ more realistic neuron models, with segregated spatial compartments

and unique computational properties per compartment^{91,153,154}. These more complicated neurons avoid many of the problems that vex point-process neuron models. If information can be separated in space within a cell, then the two propagation phases — feedforward and feedback — do not require strict temporal segregation and hence might occur simultaneously. Interactions between compartments can occur in a variety of ways, including backpropagating action potentials and plateau potentials¹⁵⁵. Furthermore, these bidirectional interactions between cellular compartments can be controlled by other factors, such as inhibitory interneuron activity that arrives at specific locations on pyramidal neurons¹⁵⁶. Segregated dendrites may also underlie ‘burst ensemble multiplexing’ schemes for representing feedforward and feedback streams of

information simultaneously in the cortex¹⁵⁴; this work shows that high-frequency bursts and singlet spikes can be driven by different compartments and may convey different kinds of information in the forward and backward paths.

Empirical studies also suggest that plateau potentials generated from electrotonically segregated dendrites can alter the plasticity in feedforward connections^{157–159}, sometimes with as few as five pairings of subthreshold presynaptic activity and at behavioural time scales^{160,161}. These plasticity findings may help explain how dendritic segregation may be used to compute differences and inform synaptic updates. Urbanczik and Senn have described how errors might be computed across somatic and dendritic compartments to drive learning¹⁵³. Guerguiev et al.¹⁶² have built on this work to show how segregated dendrites could underlie a biologically motivated variant of DTP in a spiking network simulation. More recently, Sacramento et al.¹³⁴ have proposed a variant of equilibrium propagation¹³⁷ that makes use of both dendritic segregation and local interneuron circuitry in order to achieve NGRAD learning without assuming two phases. Nevertheless, how feedforward and feedback pathways in biological tissue coordinate their activities in order to achieve effective credit assignment remains an open question.

Another intriguing detail related to biological implementation lies in the fact that many NGRAD algorithms, but not backprop, compute their learning updates using feedback-driven improvements to lower-layer activity^{34,35,37,133,135,137}. This aspect of these algorithms fits well with evidence from physiology that suggests that top-down feedback is actively involved in improving bottom-up information processing^{23,122,123}. Taken together, these new findings and related theories are expanding the horizon of possible credit assignment mechanisms that might be considered biologically plausible.

Although empirical work has broadened our understanding of how backprop-like algorithms might operate, there remain difficult challenges in terms of how to test for such learning mechanisms in the brain. One simple conjecture that has been neglected in the literature, but that is predicted by backprop and NGRADs alike, is that feedback circuitry can influence the sign and amplitude of plasticity that is elicited at feedforward synapses. To test this, we could build on previous approaches^{146,163} to examine the effects of synapse-specific plasticity protocols at feedforward connections to a cortical pyramidal neuron's

basal dendrites, while simultaneously controlling the activity patterns delivered via feedback connections to the neuron's apical dendrite. This type of experiment would not constitute a proof of backprop or of a particular NGRAD algorithm, but since both require that detailed feedback activity be able to alter feedforward learning, this kind of experimental protocol would be a natural starting place for understanding multilayer learning.

Conclusions

The way in which the cortex modifies synapses so as to improve the performance of complicated multistage networks remains one of the biggest mysteries in neuroscience. The introduction of backpropagation generated excitement in the neuroscience community as a possible source of insight about learning in the cortex²⁸. But the relevance of backpropagation to the cortex was quickly cast in doubt — partly because it failed to produce truly impressive performance in artificial systems and partly because, interpreted literally, it has obvious biological implausibilities^{25,28,29,57}.

With the advent of greater computing power, bigger datasets and a few technical improvements, backprop can now train multilayer neural networks to be competitive with human abilities^{19,20,22}. We think that backprop offers a conceptual framework for understanding how the cortex learns, but many mysteries remain with regard to how the brain could approximate it. Some of these mysteries are minor and easily addressed. For example, backprop networks are typically rate-based rather than spiking (Supplementary information) and violate Dale's law¹⁶⁴, which states that neurons in the brain form either excitatory or inhibitory connections¹⁶⁵, whereas artificial neurons are usually able to form both^{166–169}. Other mysteries, however, such as the computation and backward delivery of error signals, pose deeper conceptual issues. **NGRADs resolve significant implausibilities of backprop in a way that is intuitive and consistent with how we think biological circuits operate. They do away with the explicit propagation of error derivatives and instead compute them locally through differences in propagated activities.**

Many pieces are missing from any story that would firmly connect backprop with learning in the brain. Nevertheless, the situation now is very much reversed from that 30 years ago²⁸, when it was thought that neuroscience might have little to learn from backprop, because aspects of the algorithm seem biologically unrealistic. The reality is

that in deep neural networks, learning by following the gradient of a performance measure works really well. It therefore seems likely that a slow evolution of the thousands of genes that control the brain would favour getting as close as possible to computing the gradients that are needed for efficient learning of the trillions of synapses it contains.

Timothy P. Lillicrap^{1,2,6}, Adam Santoro^{1,6}, Luke Marris¹, Colin J. Akerman³ and Geoffrey Hinton^{4,5}

¹DeepMind, London, UK.

²Centre for Computation, Mathematics and Physics, University College London, London, UK.

³Department of Pharmacology, University of Oxford, Oxford, UK.

⁴Department of Computer Science, University of Toronto, Toronto, Canada.

⁵Google Brain, Toronto, Canada.

⁶These authors contributed equally: Timothy P. Lillicrap and Adam Santoro

✉e-mail: countzero@google.com; geoffhinton@google.com

<https://doi.org/10.1038/s41583-020-0277-3>

Published online 17 April 2020

1. Hebb, D. O. *The Organization of Behavior: A Neuropsychological Approach* (John Wiley & Sons, 1949).
2. Markram, H. & Sakmann, B. Action potentials propagating back into dendrites trigger changes in efficacy of single-axon synapses between layer V pyramidal neurons. *Soc. Neurosci. Abstr.* **21**, 2007 (1995).
3. Markram, H., Lübke, J., Frotscher, M. & Sakmann, B. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science* **275**, 213–215 (1997).
4. Gerstner, W., Kempter, R., van Hemmen, J. L. & Wagner, H. A neuronal learning rule for sub-millisecond temporal coding. *Nature* **383**, 76–78 (1996).
5. Bliss, T. V. & Lomo, T. Long-lasting potentiation of synaptic transmission in the dentate area of the anaesthetized rabbit following stimulation of the perforant path. *J. Physiol.* **232**, 331–356 (1973).
6. Bishop, C. M. *Neural Networks for Pattern Recognition* (Oxford University Press, 1995).
7. Werbos, P. *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD Thesis, Harvard Univ. P. (1974).
8. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. *Learning Internal Representations by Error Propagation*. Technical Report (DTIC Document, 1985).
9. LeCun, Y. in *Proc. Cognitiva* 85, 559–604 (CESTA, 1985).
10. Parker, D. B. *Learning-Logic: Casting the Cortex of the Human Brain in Silicon*. Technical Report Tr-47 (Center for Computational Research in Economics and Management Science, MIT, 1985).
11. Hannun, A. et al. Deep speech: scaling up end-to-end speech recognition. Preprint at <http://arxiv.org/1412.5567> (2014).
12. Krizhevsky, A., Sutskever, I. & Hinton, G. E. in *Adv. Neural Inf. Process. Syst.* 1097–1105 (NIPS, 2012).
13. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. *Proc. IEEE Conf. Comput. Vision Patt. Recog.*, 770–778 (2016).
14. Vaswani, A. et al. in *Adv. Neural Inf. Process. Syst.* 6000–6010 (NIPS, 2017).
15. Oord, A. v. d., Kalchbrenner, N. & Kavukcuoglu, K. Pixel recurrent neural networks. *PMLR* **48**, 1747–1756 (2016).
16. Van den Oord, A. et al. Wavenet: a generative model for raw audio. Preprint at <https://arxiv.org/1609.03499> (2016).
17. Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N. & Wu, Y. Exploring the limits of language modeling. Preprint at <https://arxiv.org/1602.02410> (2016).

18. Oh, J., Guo, X., Lee, H., Lewis, R. L. & Singh, S. in *Adv. Neural Inf. Process. Syst.* 2863–2871 (NIPS, 2015).
19. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
20. Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
21. Silver, D. et al. Mastering the game of go without human knowledge. *Nature* **550**, 354–359 (2017).
22. Moravčík, M. et al. DeepStack: expert-level artificial intelligence in heads-up no-limit poker. *Science* **356**, 508–513 (2017).
23. Gilbert, C. D. & Li, W. Top-down influences on visual processing. *Nat. Rev. Neurosci.* **14**, 350–363 (2013).
24. Tong, F. Primary visual cortex and visual awareness. *Nat. Rev. Neurosci.* **4**, 219–229 (2003).
25. Grossberg, S. Competitive learning: from interactive activation to adaptive resonance. *Cogn. Sci.* **11**, 23–63 (1987).
26. Marr, D. Simple memory: a theory for archicortex. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **262**, 23–81 (1971).
27. Hinton, G. E. & McClelland, J. L. in *Adv. Neural Inf. Process. Syst.* 358–366 (NIPS, 1988).
28. Crick, F. The recent excitement about neural networks. *Nature* **337**, 129–132 (1989).
29. Roelfsema, P. R. & Holtmaat, A. Control of synaptic plasticity in deep cortical networks. *Nat. Rev. Neurosci.* **19**, 166–180 (2018).
30. Whittington, J. C. & Bogacz, R. Theories of error back-propagation in the brain. *Trends Cogn. Sci.* **23**, 235–250 (2019).
31. Almeida, L. B. in *Artificial Neural Networks* 102–111 (ACM Digital Library, 1990).
32. Pineda, F. J. Generalization of back-propagation to recurrent neural networks. *Phys. Rev. Lett.* **59**, 2229–2232 (1987).
33. Pineda, F. J. Dynamics and architecture for neural computation. *J. Complex.* **4**, 216–245 (1988).
34. O'Reilly, R. C. Biologically plausible error-driven learning using local activation differences: the generalized recirculation algorithm. *Neural Comput.* **8**, 895–938 (1996).
35. Ackley, D. H., Hinton, G. E. & Sejnowski, T. J. A learning algorithm for Boltzmann machines. *Cogn. Sci.* **9**, 147–169 (1985).
36. Hinton, G. E., Dayan, P., Frey, B. J. & Neal, R. M. The 'wake-sleep' algorithm for unsupervised neural networks. *Science* **268**, 1158–1161 (1995).
37. Movellan, J. R. in *Connectionist Models: Proc. 1990 Summer School* 10–17 (ScienceDirect, 1991).
38. LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M. & Huang, F. in *Predicting Structured Data Vol. 1* (eds Bakir, G., Hofman, T., Scholkopf, B., Smola, A. & Taskar, B.) 191–245 (MIT Press, 2006).
39. Xie, X. & Seung, H. S. Equivalence of backpropagation and contrastive Hebbian learning in a layered network. *Neural Comput.* **15**, 441–454 (2003).
40. Bengio, Y. How auto-encoders could provide credit assignment in deep networks via target propagation. Preprint at <http://arXiv.org/1407.7906> (2014).
41. Lee, D.-H., Zhang, S., Fischer, A. & Bengio, Y. in *Joint Eur. Conf. Machine Learning Knowl. Discov. Databases* 498–515 (Springer, 2015).
42. Mazzoni, P., Anderson, R. A. & Jordan, M. I. A more biologically plausible learning rule for neural networks. *Proc. Natl Acad. Sci. USA* **88**, 4433–4437 (1991).
43. Seung, H. S. Learning in spiking neural networks by reinforcement of stochastic synaptic transmission. *Neuron* **40**, 1063–1073 (2003).
44. Werfel, J., Xie, X. & Seung, H. S. Learning curves for stochastic gradient descent in linear feedforward networks. *Neural Comput.* **17**, 2699–2718 (2005).
45. Spall, J. C. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Control* **37**, 332–341 (1992).
46. Williams, R. J. in *Reinforcement Learning* 5–32 (Springer, 1992).
47. Flower, B. & Jabri, M. Summed weight neuron perturbation: an O(n) improvement over weight perturbation. in *Adv. Neural Inf. Process. Syst.* 212–219 (NIPS, 1993).
48. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* (MIT Press, 2018).
49. Deisenroth, M. P. et al. A survey on policy search for robotics. *Found. Trends R. Robot.* **2**, 1–142 (2013).
50. Lillicrap, T. P. et al. Continuous control with deep reinforcement learning. Preprint at <http://arXiv.org/1509.02971> (2015).
51. Rumelhart, D., Hinton, G. & Williams, R. Learning representations by back-propagation errors. *Nature* **323**, 533–536 (1986).
52. Andersen, P., Sundberg, S., Sveen, O., Swann, J. & Wigström, H. Possible mechanisms for long-lasting potentiation of synaptic transmission in hippocampal slices from guinea-pigs. *J. Physiol.* **302**, 463–482 (1980).
53. Guillery, R. & Sherman, S. M. Thalamic relay functions and their role in corticocortical communication: generalizations from the visual system. *Neuron* **33**, 163–175 (2002).
54. Sherman, S. M., Lillicrap, T. & Tweed, D. Distinct functions for direct and transthalamic corticocortical connections. *J. Neurophysiol.* **106**, 1068–1077 (2011).
55. Viaene, A. N., Petrof, I. & Sherman, S. M. Properties of the thalamic projection from the posterior medial nucleus to primary and secondary somatosensory cortices in the mouse. *Proc. Natl Acad. Sci. USA* **108**, 18156–18161 (2011).
56. Abdelghani, M., Lillicrap, T. & Tweed, D. Sensitivity derivatives for flexible sensorimotor learning. *Neural Comput.* **20**, 2085–2111 (2008).
57. Lillicrap, T. P., Cownden, D., Tweed, D. B. & Akerman, C. J. Random synaptic feedback weights support error backpropagation for deep learning. *Nat. Commun.* **7**, 13276 (2016).
58. Cadieu, C. F. et al. Deep neural networks rival the representation of primate IT cortex for core visual object recognition. *PLoS Comput. Biol.* **10**, e1003963 (2014).
59. Yamins, D. L. et al. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proc. Natl Acad. Sci. USA* **111**, 8619–8624 (2014).
60. Elston, G. N. Cortex, cognition and the cell: new insights into the pyramidal neuron and prefrontal function. *Cereb. Cortex* **13**, 1124–1138 (2003).
61. Oh, S. W. et al. A mesoscale connectome of the mouse brain. *Nature* **508**, 207–214 (2014).
62. Jiang, X. et al. Principles of connectivity among morphologically defined cell types in adult neocortex. *Science* **350**, aac9462 (2015).
63. Tasic, B. et al. Shared and distinct transcriptomic cell types across neocortical areas. *Nature* **563**, 72–78 (2018).
64. Mountcastle, V. B. et al. Modality and topographic properties of single neurons of cat's somatic sensory cortex. *J. Neurophysiol.* **20**, 408–434 (1957).
65. Mountcastle, V. B., Motter, B., Steinmetz, M. & Sestokas, A. Common and differential effects of attentive fixation on the excitability of parietal and prestriate (V4) cortical visual neurons in the macaque monkey. *J. Neurosci.* **7**, 2239–2255 (1987).
66. Douglas, R. J., Martin, K. A. & Whitteridge, D. A canonical microcircuit for neocortex. *Neural Comput.* **1**, 480–488 (1989).
67. Bastos, A. M. et al. Canonical microcircuits for predictive coding. *Neuron* **76**, 695–711 (2012).
68. Zipser, D. & Andersen, R. A. A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons. *Nature* **331**, 679–684 (1988).
69. Lillicrap, T. P. & Scott, S. H. Preference distributions of primary motor cortex neurons reflect control solutions optimized for limb biomechanics. *Neuron* **77**, 168–179 (2013).
70. Khaligh-Razavi, S.-M. & Kriegeskorte, N. Deep supervised, but not unsupervised, models may explain IT cortical representation. *PLoS Comput. Biol.* **10**, e1003915 (2014).
71. Kriegeskorte, N. Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annu. Rev. Vis. Sci.* **1**, 417–446 (2015).
72. Wenliang, L. K. & Seitz, A. R. Deep neural networks for modeling visual perceptual learning. *J. Neurosci.* **38**, 6028–6044 (2018).
73. Pinto, N., Cox, D. D. & DiCarlo, J. J. Why is real-world visual object recognition hard? *PLoS Comput. Biol.* **4**, e27 (2008).
74. Freeman, J. & Simoncelli, E. P. Metamers of the ventral stream. *Nat. Neurosci.* **14**, 1195–1201 (2011).
75. Ullman, S., Assif, L., Fetaya, E. & Harari, D. Atoms of recognition in human and computer vision. *Proc. Natl Acad. Sci. USA* **113**, 2744–2749 (2016).
76. Kell, A. J., Yamins, D. L., Shook, E. N., Norman-Haignere, S. V. & McDermott, J. H. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron* **98**, 630–644 (2018).
77. Mitchell, M. *An Introduction to Genetic Algorithms* (MIT Press, 1998).
78. Saxe, A. M. *Deep Linear Neural Networks: A Theory of Learning in the Brain and Mind*. PhD thesis, Stanford Univ. (2015).
79. Zmarz, P. & Keller, G. B. Mismatch receptive fields in mouse visual cortex. *Neuron* **92**, 766–772 (2016).
80. Issa, E. B., Cadieu, C. F. & DiCarlo, J. J. Neural dynamics at successive stages of the ventral visual stream are consistent with hierarchical error signals. *eLife* **7**, e42870 (2018).
81. Schultz, W., Dayan, P. & Montague, P. R. A neural substrate of prediction and reward. *Science* **275**, 1593–1599 (1997).
82. Zipser, D. & Rumelhart, D. in *Computational Neuroscience* (ed. Schwartz, E. L.) 192–200 (1990).
83. Stork, D. G. in *Int. Joint Conf. Neural Netw.* 2 (1989), 241–246.
84. Brandt, R. D. & Lin, F. in *Proc. 1996 IEEE Int. Conf. Neural Netw.* 300–305 (1996).
85. Brandt, R. D. & Lin, F. in *Proc. 1996 IEEE Int. Symp. Intell. Control* 86–90 (1996).
86. Oztas, E. Neuronal tracing. *Neuroanatomy* **2**, 2–5 (2003).
87. Harris, K. D. Stability of the fittest: organizing learning through retroaxonal signals. *Trends Neurosci.* **31**, 130–136 (2008).
88. Venkateswararao, L. C. *Adaptive Optimal-Control Algorithms for Brainlike Networks* PhD Thesis, Univ. Toronto (2010).
89. Hinton, G. The ups and downs of Hebb synapses. *Can. Psychol.* **44**, 10–13 (2003).
90. Kolen, J. F. & Pollack, J. B. in *IEEE World Congress Comput. Intell.* 3, 1375–1380 (IEEE, 1994).
91. Körding, K. P. & König, P. Supervised and unsupervised learning with two sites of synaptic integration. *J. Comput. Neurosci.* **11**, 207–215 (2001).
92. Lillicrap, T. P., Cownden, D., Tweed, D. B. & Akerman, C. J. Random feedback weights support learning in deep neural networks. Preprint at <http://arXiv.org/1411.0247> (2014).
93. Nøklund, A. in *Adv. Neural Inf. Process. Syst.* 1045–1053 (NIPS, 2016).
94. Guerghiuev, J., Lillicrap, T. P. & Richards, B. A. Deep learning with segregated dendrites. *eLife* **6**, e22901 (2017).
95. Liao, Q., Leibo, J. Z. & Poggio, T. How important is weight symmetry in backpropagation? Preprint at <https://arXiv.org/1510.05067> (2015).
96. Samadi, A., Lillicrap, T. P. & Tweed, D. B. Deep learning with dynamic spiking neurons and fixed feedback weights. *Neural Comput.* **29**, 578–602 (2017).
97. Moskovitz, T. H., Litwin-Kumar, A. & Abbott, L. Feedback alignment in deep convolutional networks. Preprint at <https://arXiv.org/1812.06488> (2018).
98. Xiao, W., Chen, H., Liao, Q. & Poggio, T. Biologically-plausible learning algorithms can scale to large datasets. Preprint at <https://arXiv.org/1811.03567> (2018).
99. Amit, Y. Deep learning with asymmetric connections and Hebbian updates. *Front. Comput. Neurosci.* **13**, 18 (2019).
100. Bartunov, S. et al. in *Adv. Neural Inf. Process. Syst.* 9390–9400 (NIPS, 2018).
101. Akrou, M., Wilson, C., Humphreys, P. C., Lillicrap, T. & Tweed, D. Using weight mirrors to improve feedback alignment. Preprint at <https://arXiv.org/1904.05391> (2019).
102. Pascanu, R., Mikolov, T. & Bengio, Y. in *Proc. Int. Conf. Machine Learning* 1310–1318 (ICML, 2013).
103. Coesmans, M., Weber, J. T., De Zeeuw, C. I. & Hansel, C. Bidirectional parallel fiber plasticity in the cerebellum under climbing fiber control. *Neuron* **44**, 691–700 (2004).
104. Yang, Y. & Lisberger, S. G. Purkinje-cell plasticity and cerebellar motor learning are graded by complex-spike duration. *Nature* **510**, 529–532 (2014).
105. Li, W., Piech, V. & Gilbert, C. D. Contour saliency in primary visual cortex. *Neuron* **50**, 951–962 (2006).
106. Motter, B. C. Focal attention produces spatially selective processing in visual cortical areas V1, V2, and V4 in the presence of competing stimuli. *J. Neurophysiol.* **70**, 909–919 (1993).
107. Moran, J. & Desimone, R. Selective attention gates visual processing in the extrastriate cortex. *Front. Cognit. Neurosci.* **229**, 342–345 (1985).
108. Spitzer, H., Desimone, R. & Moran, J. Increased attention enhances both behavioral and neuronal performance. *Science* **240**, 338–340 (1988).

109. Chelazzi, L., Miller, E. K. & Duncan, J. A neural basis for visual search in inferior temporal cortex. *Nature* **363**, 27 (1993).
110. Chelazzi, L., Miller, E. K., Duncan, J. & Desimone, R. Responses of neurons in macaque area V4 during memory-guided visual search. *Cereb. Cortex* **11**, 761–772 (2001).
111. Treue, S. & Maunsell, J. H. Attentional modulation of visual motion processing in cortical areas MT and MST. *Nature* **382**, 539–541 (1996).
112. Luck, S. J., Chelazzi, L., Hillyard, S. A. & Desimone, R. Neural mechanisms of spatial selective attention in areas V1, V2, and V4 of macaque visual cortex. *J. Neurophysiol.* **77**, 24–42 (1997).
113. Ito, M. & Gilbert, C. D. Attention modulates contextual influences in the primary visual cortex of alert monkeys. *Neuron* **22**, 593–604 (1999).
114. McAdams, C. J. & Maunsell, J. H. Effects of attention on orientation-tuning functions of single neurons in macaque cortical area V4. *J. Neurosci.* **19**, 431–441 (1999).
115. Reynolds, J. H. & Desimone, R. Interacting roles of attention and visual salience in V4. *Neuron* **37**, 853–863 (2003).
116. Abbott, L. F., Varela, J., Sen, K. & Nelson, S. Synaptic depression and cortical gain control. *Science* **275**, 221–224 (1997).
117. Chance, F. S., Abbott, L. & Reyes, A. D. Gain modulation from background synaptic input. *Neuron* **35**, 773–782 (2002).
118. Girard, P., Hupé, J. & Bullier, J. Feedforward and feedback connections between areas V1 and V2 of the monkey have similar rapid conduction velocities. *J. Neurophysiol.* **85**, 1328–1331 (2001).
119. De Pasquale, R. & Sherman, S. M. Synaptic properties of corticocortical connections between the primary and secondary visual cortical areas in the mouse. *J. Neurosci.* **31**, 16494–16506 (2011).
120. Kosslyn, S. M. & Thompson, W. L. When is early visual cortex activated during visual mental imagery? *Psychol. Bull.* **129**, 723–746 (2003).
121. Bridge, H., Harrold, S., Holmes, E. A., Stokes, M. & Kennard, C. Vivid visual mental imagery in the absence of the primary visual cortex. *J. Neurol.* **259**, 1062–1070 (2012).
122. Manita, S. et al. A top-down cortical circuit for accurate sensory perception. *Neuron* **86**, 1304–1316 (2015).
123. Fyall, A. M., El-Shamayleh, Y., Choi, H., Shea-Brown, E. & Pasupathy, A. Dynamic representation of partially occluded objects in primate prefrontal and visual cortex. *eLife* **6**, e25784 (2017).
124. Mignard, M. & Malpeli, J. G. Paths of information flow through visual cortex. *Science* **251**, 1249–1252 (1991).
125. Markov, N. T. & Kennedy, H. The importance of being hierarchical. *Curr. Opin. Neurobiol.* **23**, 187–194 (2013).
126. Ahissar, M. & Hochstein, S. The reverse hierarchy theory of visual perceptual learning. *Trends Cognit. Sci.* **8**, 457–464 (2004).
127. Lee, T. S. & Mumford, D. Hierarchical Bayesian inference in the visual cortex. *J. Opt. Soc. Am. A Opt. Image Sci. Vis.* **20**, 1434–1448 (2003).
128. Lewicki, M. S. & Sejnowski, T. J. in *Adv. Neural Inf. Process. Syst.* 529–535 (NIPS, 1997).
129. Knill, D. C. & Richards, W. *Perception as Bayesian Inference* (Cambridge Univ. Press, 1996).
130. Dayan, P., Hinton, G. E., Neal, R. M. & Zemel, R. S. The Helmholtz machine. *Neural Comput.* **7**, 889–904 (1995).
131. Von Helmholtz, H. & Southall, J. P. C. *Treatise on Physiological Optics* (Courier Corp., 2005).
132. Ackley, D. H., Hinton, G. E. & Sejnowski, T. J. in *Readings in Computer Vision* 522–533 (Elsevier, 1987).
133. Whittington, J. C. & Bogacz, R. An approximation of the error backpropagation algorithm in a predictive coding network with local Hebbian synaptic plasticity. *Neural Comput.* **29**, 1229–1262 (2017).
134. Sacramento, J., Costa, R. P., Bengio, Y. & Senn, W. Dendritic error backpropagation in deep cortical microcircuits. Preprint at <https://arxiv.org/1801.00062> (2017).
135. Sacramento, J., Costa, R. P., Bengio, Y. & Senn, W. in *Adv. Neural Inf. Process. Syst.* 8721–8732 (NIPS, 2018).
136. Scellier, B. & Bengio, Y. Towards a biologically plausible backprop. Preprint at <https://arxiv.org/1602.05179.914> (2016).
137. Scellier, B. & Bengio, Y. Equilibrium propagation: bridging the gap between energy-based models and backpropagation. *Front. Comput. Neurosci.* **11**, 24 (2017).
138. Hinton, G. How to do backpropagation in a brain. *Deep Learning Workshop* (NIPS, 2007).
139. Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. in *Proc. 25th Int. Conf. Machine Learning* 1096–1103 (ICML, 2008).
140. Smolensky, P. *Information Processing in Dynamical Systems: Foundations of Harmony Theory* Technical Report (Univ. Colorado at Boulder, 1986).
141. LeCun, Y. in *Disordered Systems and Biological Organization* 233–240 (Springer, 1986).
142. LeCun, Y. Modèles connexionnistes de l'apprentissage. PhD Thesis, Univ. Paris 6 (1987).
143. Coogan, T. & Burkhalter, A. Conserved patterns of cortico-cortical connections define areal hierarchy in rat visual cortex. *Exp. Brain Res.* **80**, 49–53 (1990).
144. D'Souza, R. D. & Burkhalter, A. A laminar organization for selective cortico-cortical communication. *Front. Neuroanat.* **11**, 71 (2017).
145. Wimmer, V. C., Bruno, R. M., De Kock, C. P., Kuner, T. & Sakmann, B. Dimensions of a projection column and architecture of VPM and POM axons in rat vibrissal cortex. *Cereb. Cortex* **20**, 2265–2276 (2010).
146. Williams, L. E. & Holtmaat, A. Higher-order thalamocortical inputs gate synaptic long-term potentiation via disinhibition. *Neuron* **101**, 91–102 (2019).
147. Larkum, M. E., Zhu, J. J. & Sakmann, B. A new cellular mechanism for coupling inputs arriving at different cortical layers. *Nature* **398**, 338–341 (1999).
148. Gordon, U., Polsky, A. & Schiller, J. Plasticity compartments in basal dendrites of neocortical pyramidal neurons. *J. Neurosci.* **26**, 12717–12726 (2006).
149. Branco, T., Clark, B. A. & Häusser, M. Dendritic discrimination of temporal input sequences in cortical neurons. *Science* **329**, 1671–1675 (2010).
150. Branco, T. & Häusser, M. Synaptic integration gradients in single cortical pyramidal cell dendrites. *Neuron* **69**, 885–892 (2011).
151. Losonczy, A., Makara, J. K. & Magee, J. C. Compartmentalized dendritic plasticity and input feature storage in neurons. *Nature* **452**, 436–441 (2008).
152. Polsky, A., Mel, B. W. & Schiller, J. Computational subunits in thin dendrites of pyramidal cells. *Nat. Neurosci.* **7**, 621–627 (2004).
153. Urbanczik, R. & Senn, W. Learning by the dendritic prediction of somatic spiking. *Neuron* **81**, 521–528 (2014).
154. Naud, R. & Sprekeler, H. Sparse bursts optimize information transmission in a multiplexed neural code. *PNAS* **115**, E6329–E6338 (2018).
155. Schiess, M., Urbanczik, R. & Senn, W. Somato-dendritic synaptic plasticity and error-backpropagation in active dendrites. *PLoS Comput. Biol.* **12**, e1004638 (2016).
156. Klausberger, T. & Somogyi, P. Neuronal diversity and temporal dynamics: the unity of hippocampal circuit operations. *Science* **321**, 53–57 (2008).
157. Sjöström, P. J. & Häusser, M. A cooperative switch determines the sign of synaptic plasticity in distal dendrites of neocortical pyramidal neurons. *Neuron* **51**, 227–238 (2006).
158. Richards, B. A. & Lillicrap, T. P. Dendritic solutions to the credit assignment problem. *Curr. Opin. Neurobiol.* **54**, 28–36 (2019).
159. Muller, S. Z., Zadina, A., Abbott, L. & Sawtell, N. Continual learning in a multi-layer network of an electric fish. *Cell* **179**, 1382–1392.e10 (2019).
160. Bittner, K. C. et al. Conjunctive input processing drives feature selectivity in hippocampal CA1 neurons. *Nat. Neurosci.* **18**, 1133–1142 (2015).
161. Bittner, K. C., Milstein, A. D., Grienberger, C., Romani, S. & Magee, J. C. Behavioral time scale synaptic plasticity underlies CA1 place fields. *Science* **357**, 1033–1036 (2017).
162. Guerguiev, J., Lillicrap, T. P. & Richards, B. A. Towards deep learning with segregated dendrites. *eLife* **6**, e22901 (2017).
163. Kwag, J. & Paulsen, O. The timing of external input controls the sign of plasticity at local synapses. *Nat. Neurosci.* **12**, 1219–1221 (2009).
164. Dale, H. Pharmacology and nerve-endings. *Proc. R. Soc. Med.* **28**, 319–332 (1935).
165. Osborne, N. N. Is Dale's principle valid? *Trends Neurosci.* **2**, 73–75 (1979).
166. O'Donohue, T. L., Millington, W. R., Handelsmann, G. E., Contreras, P. C. & Chronwall, B. M. On the 50th anniversary of Dale's law: multiple neurotransmitter neurons. *Trends Pharmacol. Sci.* **6**, 305–308 (1985).
167. Draye, J.-P., Cheron, G., Libert, G. & Godaux, E. Emergence of clusters in the hidden layer of a dynamic recurrent neural network. *Biol. Cybern.* **76**, 365–374 (1997).
168. De Kamps, M. & van der Velde, F. From artificial neural networks to spiking neuron populations and back again. *Neural Netw.* **14**, 941–953 (2001).
169. Parisien, C., Anderson, C. H. & Eliasmith, C. Solving the problem of negative synaptic weights in cortical models. *Neural Comput.* **20**, 1473–1494 (2008).
170. Zeiler, M. D. & Fergus, R. in *Eur. Conf. Comput. Vision* 818–833 (2014).

Author contributions

T.P.L. and A.S. contributed equally to this work. T.P.L., G.H. and A.S. researched data for the article, and T.P.L., G.H., C.J.A. and A.S. wrote the article. The authors all provided substantial contributions to discussion of the content and reviewed and edited the manuscript before submission. The authors contributed equally to all aspects of the article.

Competing interests

The authors declare no competing interests.

Peer review information

Nature Reviews Neuroscience thanks Y. Amit, J. DiCarlo, W. Senn and T. Toyoizumi for their contribution to the peer review of this work.

Publisher's note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Supplementary information

Supplementary information is available for this paper at <https://doi.org/10.1038/s41583-020-0277-3>.

© Springer Nature Limited 2020