1 Back-propagation and the Brain

Timothy P Lillicrap, Adam Santoro, Luke Marris, Colin J Akerman, Geoffrey Hinton

1.1 Credit Assignment Problem

Brains have good priors through evolution.
Synaptic weights → Neural activity → Network Output → Network Error.

- Weight Perturbation: Change a random synapse strength and see what happens to the network error. Require 1000's of such perturbations to make good predictions
- Node Perturbation: Perturb the output of neurons instead of weights. Compute local derivatives of a neurons activity wrt to its own weight.
- Back-propagation: Computes the change in error wrt to synapse strength using calculus. Time required is similar to forward propagation. Error signals are passed from top layers to bottom ones. In supervised setting the error is calculated using target, in RL wrt temporal difference error or policy gradient, or reconstruction error for unsupervised learning.

1.2 The NGRAD Hypothesis

- Difference in activity states can be used to drive synaptic changes. Temporal Difference between activities during two phases of propagation can be used.
- 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.
- GeneRec, contrastive hebbian learning, almeida/pineda algorithms, wake sleep algorithms use a similar logic
- **Auto Encoders**: Fundamental to NGRAD hypothesis is the use of autoencoders to sendtop-down signals to earlier layer activations and the use of the induced differences to make weight updates.

1.3 Proposed Models

- **Auto Encoders**: Reconstruct the input. Error is the difference between the input and the inverse reconstuction.
$$g(f(x;W);B) \approx x \tag{1}$$
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.
- **Problems with Target Propagation/DTP**:
    1. Straightforward implementation of DTP does not perform well on complex datasets compared to backprop with convolutional layers.
    2. The DTP algorithm also does not address questions of online learning or how the forward and backward pathways could communicate in biological circuits

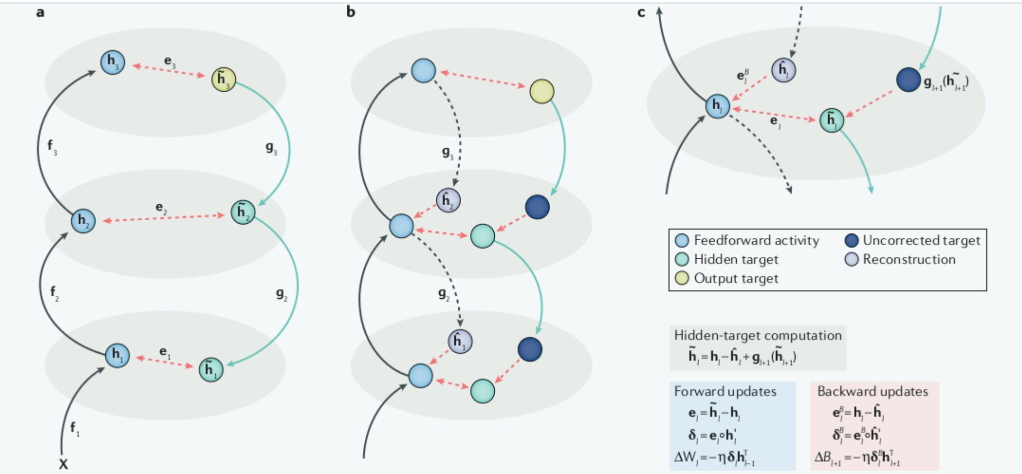

Fig. 3 | **Target propagation algorithms. a** | Schematic of target propagation that uses perfect inverses, $\mathbf{g}_l(\cdot) = \mathbf{f}_l^{-1}(\cdot)$, at each layer. For illustration, high-dimensional activity vectors at each layer are represented as points in a 2D space. Local layer-wise errors, $\mathbf{e}_l = \tilde{\mathbf{h}}_l - \mathbf{h}_l$, 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, 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, $\mathbf{e}^B_l = \mathbf{h}_l - \hat{\mathbf{h}}_l$, 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, 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, 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.

Figure 1: