GAIT-prop: A biologically plausible learning rule derived from backpropagation of error

Nasir Ahmad  Marcel van Gerven  Luca Ambrogioni  
Department of Artificial Intelligence  
Donders Institute for Brain, Cognition and Behaviour  
Radboud University, Nijmegen, the Netherlands  
{n.ahmad,m.vangerven,l.ambrogioni}@donders.ru.nl

Abstract

Traditional backpropagation of error, though a highly successful algorithm for learning in artificial neural network models, includes features which are biologically implausible for learning in real neural circuits. An alternative called target propagation proposes to solve this implausibility by using a top-down model of neural activity to convert an error at the output of a neural network into layer-wise and plausible 'targets' for every unit. These targets can then be used to produce weight updates for network training. However, thus far, target propagation has been heuristically proposed without demonstrable equivalence to backpropagation. Here, we derive an exact correspondence between backpropagation and a modified form of target propagation (GAIT-prop) where the target is a small perturbation of the forward pass. Specifically, backpropagation and GAIT-prop give identical updates when synaptic weight matrices are orthogonal. In a series of simple computer vision experiments, we show near-identical performance between backpropagation and GAIT-prop with a soft orthogonality-inducing regularizer.

1 Introduction

One of the fundamental tenets of modern systems neuroscience is that the brain learns by selectively strengthening and weakening synaptic connections. Much research in theoretical neuroscience was guided by the Hebbian principle of strengthening connections between co-active neurons [1–3]. However, it is now clear that purely Hebbian learning rules are not effective at learning complex behavioral tasks. In the last fifteen years, the fields of machine learning and AI have been revolutionized by the large-scale adoption of deep networks trained by backpropagation (BP) [4–6]. Deep networks have been shown to replicate the hierarchy of cortical representations [7–9], suggesting a connection between deep learning and the brain. However, BP has since long been considered as biologically implausible [10], based in part on its use of non-local information at individual synapses which carry out weight updates. How such information could be stored, transmitted and leveraged has been a cause for concern [10][11].

To overcome these implausible aspects of the BP algorithm, approaches have been proposed to approximate or replace implausible computations with more realistic and plausible elements [12–16]. Alternatively methods which approximate backpropagation through energy-based models have also been proposed [17–20]. Among those methods, contrastive Hebbian learning, and generalized recirculation have been shown to produce BP-equivalent updates under specific regimes [18, 21], though these methods require a, rather artificial, alternation between positive and negative phases in order to compute updates.

Target Propagation (TP) is a simpler and more scalable approach which proposes that the loss function at the output layer be replaced with layer-wise, local target activities for individual neurons [22, 23].

Preprint. Under review.
The principle of TP is to propagate an output target ‘backwards’ through a network using (learned) inverses of the forward pass. Under a perfect inverse, these layer-wise targets are equivalent to the outputs of hidden layers which would have precisely produced the desired output. In a recent review paper, Lillicrap and co-authors suggested an approach named ‘neural gradient representation by activity differences’ (NGRAD) [24]. They conjecture that the most plausible implementation of an effective learning rule in the brain would consist of projecting error-based information into layer-wise neural activity. Given this conjecture, they highlight TP as a feasible and promising approach. However, it remains unclear how updates computed by TP relate to BP and the associated gradients which would optimise network performance.

In this paper, we develop a theoretical framework to analyse the link between the TP and the BP weight update rules. In particular, we show that TP and BP have the same local optima in a deep linear network. Furthermore, we show that in deep linear networks the two update rules are identical when the weight matrices are orthogonal. However, standard TP cannot be easily linked to BP in the non-linear case, even under conditions of orthogonality. A connection to BP can be fully restored by evolving incremental targets – targets which are an infinitesimal shift of the forward pass toward a target output. Using this approach we derive the gradient-adjusted incremental target propagation algorithm (GAIT-prop), a biologically plausible approach to learning in non-linear networks that is identical to BP under orthogonal weight matrices. Unlike TP, this approach can also be approximated in the equilibrium state of network with constant input and weak feedback coupling, connecting our method to activity recirculation and equilibrium propagation [18, 19]. Furthermore, our approach to local, error-based learning encodes the exact gradient descent information desired for optimal learning within target neural activities in a plausible circuit mechanism [24].

To derive the theoretical relations between BP, TP, and GAIT-prop we make use of invertible networks. The principle of TP is to propagate an output target 'backwards' through a network using (learned) inverses of the forward pass. Under a perfect inverse, these layer-wise targets are equivalent to the outputs of hidden layers which would have precisely produced the desired output. In a recent review paper, Lillicrap and co-authors suggested an approach named 'neural gradient representation by activity differences' (NGRAD) [24]. They conjecture that the most plausible implementation of an effective learning rule in the brain would consist of projecting error-based information into layer-wise neural activity. Given this conjecture, they highlight TP as a feasible and promising approach. However, it remains unclear how updates computed by TP relate to BP and the associated gradients which would optimise network performance.

2 Background on backpropagation and target propagation

We start by reviewing the basics of BP. Let us consider a feedforward neural network with an input layer and $L$ subsequent layers. We describe the output of any given layer, as

$$ y_l = g_l(y_{l-1}) = f(W_l y_{l-1}) ,$$

where $y_l$ is the output of the $l$-th layer, $W_l$ is a weight matrix and $f(\cdot)$ is the activation function. We denote the 'pre-activations' $W_l y_{l-1}$ as $h_l$ and use $y_0$ to denote the input.

We consider a quadratic loss between the network output $y_L$ and a target output $t_L$. Given an input-target pair $(y_0, t_L)$ we can define a quadratic loss function, $\ell$ as

$$ \ell = \frac{1}{2}(y_L - t_L)^2 .$$ (2)

The corresponding BP weight update is proportional to the gradient of this loss and has the following form:

$$ \Delta W_l^{\text{BP}} = -\eta A_l \left( \prod_{j=l+1}^{L} W_j^T A_j \right) (y_L - t_L) y_{l-1}^T, $$ (3)

where $A_l$ is a diagonal matrix with $f'(h_l)$ in its main diagonal and $\eta$ is a learning rate.

Target propagation (TP) is an arguably more biologically plausible learning approach in which the desired target is propagated through the network by (approximate) inverses of the forward computations [22]. In its simplest form, given an input-target pair $(y_0, t_L)$, standard TP prescribes a layer-wise loss of the following form:

$$ \Delta W_l^{\text{TP}} = -\eta A_l (y_l - t_l) y_{l-1}^T $$ (4)

where $t_l$ is a layer-wise target obtained by applying a (potentially approximate) inverse network to the output target $t_L$. An exact inverse can be defined by the following recursive relation:

$$ t_{l-1} = g_l^{-1}(t_l) = f^{-1}(W_{l}^{-1} t_l) . $$ (5)
The existence of an exact inverse places constraints on the architecture of the network. In particular, weight matrices must be square, and the activation function must be invertible for any real-valued input. However, this does not require that all layers of a network have the same number of units as we explore in the second half of this paper using auxiliary variables.

The simplicity of TP and its relatively high performance makes it a leading candidate for biologically plausible deep learning. However, TP is a heuristic method that has not been shown to replicate or approximate BP. In the following, we will derive a series of formal connections between BP and TP. Moreover, we will introduce a new TP-like algorithm that can be shown to reduce to BP under specific conditions.

3 Relationship between BP and TP in linear networks

In this section we will reformulate the BP updates of a deep linear network in terms of local activity differences. We begin by rewriting the output difference \((y_L - t_L)\) in terms of \(l\)-th layer activity differences \((y_l - t_l)\), where \(t_l\) is a deep target obtained by applying a sequence of layer-wise inversions, as in Eq. [5]. We will assume the existence of inverse weight matrices \(W_l^{-1}\) such that \(W_l W_l^{-1} = W_l^{-1} W_l = I\). This assumption constrains both the weight matrix shapes (they must be square) and implies that these matrices must be invertible (full rank). Using inverse weight matrices, we can rewrite our difference term as

\[
y_L - t_L = g_L(y_{L-1}) - g_L(t_{L-1}) = f(W_l y_{L-1}) - f(W_l t_{L-1})
\]

where the target of the \(L - 1\)-th layer, \(t_{L-1}\), is defined as in Eq. [5]. Since the network is linear, we can ignore the activation function and collect these two terms into the matrix product \(W_l(y_{L-1} - t_{L-1})\). This formula can then be applied recursively to an arbitrary depth, leading to the expression

\[
y_L - t_L = F_l(y_L) - t_l,
\]

where \(F_l\) is defined as in Eq. [5]. Substituting this formula into the BP update rule of a linear network, we obtain a reformulation of BP in terms of local targets such that

\[
\Delta W_l^{BP} = -\eta (F_l^T F_l)(y_l - t_l) y_{l-1}^T = (F_l^T F_l) \Delta W_l^{TP}.
\]

Since \((F_l^T F_l)\) is full-rank under our assumption of invertibility, this equation implies that \(\Delta W_l^{BP} = 0\) if and only if \(\Delta W_l^{TP} = 0\), meaning that BP and TP have the same fixed points in invertible linear networks. Furthermore, these fixed points have the same stability since \((F_l^T F_l)\) is positive definite. Finally, in linear networks TP updates are identical to BP updates when all the weight vectors are orthogonal, where \(F_l^T F_l = I\).

4 Incremental target propagation

If we assume that the Euclidean distance between the activations \(y_l\) and the targets \(t_l\) is sufficiently small, we can derive a linear approximation of Eq. [6] for an arbitrary transfer function and extend the above analysis to non-linear networks. However, during the early stages of training, when network outputs are far from targets, such an assumption would be unreasonable. To overcome this issue, we reformulate target difference in terms of an ‘infinitesimal increment’:

\[
y_L - t_L = \gamma_L^{-1}(y_L - (1 - \gamma_L) y_L + \gamma_L t_L)
\]

where \(\gamma_L\) is a scalar and we have defined a new ‘incremental’ target \(t_L^* = (1 - \gamma_L) y_L + \gamma_L t_L\). Assuming that \(0 < \gamma_L < 1\), this new target is an incremental shift from the current network output \(y_L\), towards the target network output \(t_L\). If \(g_L(\cdot)\) (our network’s forward pass) is continuous, for any real-valued \(\epsilon\) there is a \(\gamma_L\) such that \(\|y_L - t_L^*\|_2 < \epsilon\). Therefore, assuming \(g_L(\cdot)\) to be a continuous function, we can approximate the resulting difference with a linear function:

\[
y_L - t_L = \gamma_L^{-1}(g_L(y_{L-1}) - g_L(t_{L-1}^*))
\]

\[
= \lim_{\gamma_L \to 0} \gamma_L^{-1}(A_L W_L (y_{L-1} - t_{L-1}^*))
\]

\[
= \lim_{\gamma_L \to 0} \gamma_L^{-1}(A_L W_L (y_{L-1} - t_{L-1}^*))
\]

\]
where the approximation error is of the order of \( \|y_L - t_L^*\|_2^2 \). This procedure can be recursively carried out to describe the difference term at our output layer as a function of activity at a layer of any depth such that
\[
y_L - t_L = \lim_{\gamma L \to 0} \left( \prod_{j=0}^{t+1} \gamma_{L-j} A_{L-j} W_{L-j} \right) (y_l - t_l^*)
\] (11)
with the layer-wise incremental target defined recursively as
\[
t_{l-1}^* = (1 - \gamma_l) y_{l-1} + \gamma_l g_l^{-1}(t_l^*) .
\] (12)
We can now define an incremental TP-based update, \( \Delta W_i^{\text{ITP}} \) where
\[
\Delta W_i^{\text{ITP}} = -\eta A_l (y_l - t_l^*) y_{l-1}^T .
\] (13)
Substituting Eq. [13] into the BP update rule (Eq. [3]), we obtain an asymptotic linear relation between BP and ITP updates:
\[
\Delta W_i^{\text{BP}} = \lim_{\gamma L \to 0} \frac{M(h_l, \ldots, h_L)}{M(h_l, \ldots, h_L)} \Delta W_i^{\text{ITP}}
\] (14)
with
\[
M(h_l, \ldots, h_L) = \left( \prod_{j=l+1}^{L} \gamma_j^{-1} \right) A_l \left( \prod_{j=l+1}^{L} W_j^T A_j \right) \left( \prod_{j=0}^{t+1} A_{L-j} W_{L-j} \right) A_l^{-1} .
\] (15)
Unfortunately, since \( M(h_l, \ldots, h_L) \) depends on the layer-wise activity (via the diagonal matrices of derivatives, \( A_l \)), Eq. [14] does not imply an equivalence between the fixed points of BP and ITP for a dataset bigger than one sample. Furthermore, orthogonality of weight matrices is also unhelpful due to these same multiplications by layer-wise activation function derivatives. In general, since these derivatives depend upon the input data, it is not possible to find a constraint on the weights that restores the equivalence of the linear case. Fortunately, we can restore this equivalence by further modification of the incremental target, as we demonstrate in the next section.

5 Gradient-adjusted incremental target propagation

By incorporating the data-dependent derivatives into the \( \gamma_l \) variables, we can recover an equivalence between BP and an approach which makes use of layer-wise targets. Specifically, by introducing
\[
\epsilon_l = \gamma_l A_l^2 ,
\] (16)
we can define gradient-adjusted incremental targets
\[
t_l^* = (1 - \epsilon_l) y_l + \epsilon_l g_l^{-1}(t_l^*) .
\] (17)
Given this new target formulation, we can now define the ‘gradient-adjusted incremental target propagation’ (GAIT-prop) update
\[
\Delta W_l^{\text{GAIT}} = -\eta A_l (y_l - t_l^*) y_{l-1}^T .
\] (18)
If we express the GAIT-prop-based update in terms of the BP updates we find
\[
\Delta W_l^{\text{BP}} = \lim_{\gamma L \to 0} N(h_l, \ldots, h_L) \Delta W_l^{\text{GAIT}} ,
\] (19)
with
\[
N(h_l, \ldots, h_L) = \left( \prod_{j=l+1}^{L} \gamma_j^{-1} \right) A_l \left( \prod_{j=l+1}^{L} W_j^T A_j \right) \left( \prod_{j=0}^{t+1} A_{L-j} W_{L-j} \right) A_l^{-1} .
\] (20)
On inspection, this matrix formulation (aside from the preceding gamma terms) reduces to the identity matrix when all weight matrices \( W_l \) are orthogonal matrices. This allows us to express the following equivalence formula
\[
\Delta W_l^{\text{BP}} = \lim_{\gamma L \to 0} \left( \prod_{j=l+1}^{L} \gamma_j^{-1} \right) \Delta W_l^{\text{GAIT}}
\] (21)
as being true under the assumption of orthogonality of weight vectors. In practice, a fixed small value is used in place of the \( \gamma \) terms instead of a layer-wise infinitesimal.

Note that the GAIT-prop update rule (Eq. [18]) only uses locally-available information and it is in this sense as biologically plausible as the classic TP target. Pseudocode for GAIT-prop is given in Alg. [E].
Algorithm 1 GAIT-prop (per training sample update)

\[\text{for } l = 1 \text{ to } L \text{ do}\]
\[y_l \leftarrow f(W_l y_{l-1})\]
\[\text{end for}\]

With output target, \(t^*_L \leftarrow t_L\)

\[\text{for } l = L - 1 \text{ to } 1 \text{ do}\]
\[t^*_l \leftarrow g^{-1}((1 - \epsilon) y_{l+1} + \epsilon (t^*_{l+1}))\]
\[\text{end for}\]

\[\text{for } l = 1 \text{ to } L \text{ do}\]
\[\ell_l(W_l) = \gamma_{L-1}L^2(y_l - t^*_{l})^2\]
\[\text{Update } W_l \text{ by SGD on the quadratic loss function } \ell_l(W_l)\]
\[\text{end for}\]

6 GAIT-prop in a neural circuit

![Graphical depiction of TP versus GAIT-Prop.](image)

Figure 1: **Left:** Graphical depiction of TP versus GAIT-Prop. **Right:** Scatter plots showing the alignment of TP and GAIT-prop weight updates against BP. These are shown for updates to an untrained (square) network with random or orthogonal weight initializations.

One significant weakness in the biological plausibility of TP is that the target signal needs to be propagated backward without being ‘contaminated’ by the forward pass. This requires either a parallel network for targets alone, a complete resetting of the network (with blocking of inputs), or some sophisticated form of compartmentalized neurons capable of propagating two signals in both directions.

In comparison, the incremental nature of the layer-wise targets produced by GAIT-prop makes it particularly suitable for an implementation in a biologically realistic network model. Figure 1 Left, depicts the differences between both algorithms. The backward-propagated signals for GAIT-prop are (weakly) coupled to the forward pass, meaning that both the forward and backward passes can co-exist. In fact, the ITP algorithm (equivalent to GAIT-prop in a linear network) can be shown to emerge in the equilibrium state of a simple dynamical system model of a neural network with feedback connection (see Appendix A).

Figure 1 right, shows the efficacy of the coupling proposed in GAIT-prop when combined with orthogonal weight matrices. The weight updates produced by GAIT-prop in this condition almost perfectly equals those computed by BP. Furthermore, this coupling reduces the requirement for two non-interfering information flows by suggesting that the same inputs can be present during the target propagation phase.

7 Simulated Invertible Networks and Tasks
Invertible Components All of the analyses and simulations presented in this paper require an invertible network model. This requires both an invertible activation function and invertible weight matrices.

Aside from linear networks, we use the leaky-ReLu activation function. The inverses of both the linear and of the leaky-ReLu activation functions are trivial.

To ensure invertibility of weight matrices, we only make use of square matrices and empirically find that by random initialisation these remain full-rank during training (and therefore invertible). The use of square matrices places constraints upon the network architectures from which we can choose. The simplest network architecture of choice is a network with a fixed-width, i.e. every layer of the network has an equivalent number of units as the inputs. The consequence of such invertible, fixed-width networks (of width equivalent to the input) is that there is sufficient information at every layer to reproduce the input pattern. However, the tasks we make use of require only ten output neurons (far fewer than the number of inputs) and to accommodate this, we make use of auxiliary units.

Auxiliary units and information loss So far, we assumed that the layer-wise transformations, $g_l(\cdot)$, are fully invertible functions. This requirement places strong constraints upon the network architecture. Specifically, fully invertible architectures require as many output nodes as there are input nodes at every layer and cannot discard task-irrelevant information.

In the TP literature, this problem is addressed using learned pseudo-inverses (autoencoders) which can transform between layers of arbitrary size \cite{23}. However, in practice, a target obtained using pseudo-inverses must represent some prototypical target since not all low-level information can be recovered. This can result in the presence of non-zero error terms despite correct network behaviour.

Bartunov et al. \cite{25} first suggested the use of ‘auxiliary output’ units – additional units at the output of a network which are provided no error signal and are used to store task-irrelevant features so that diverse targets can be produced for the hidden layers of a network. Without these, the targets provided to hidden layers of all examples of a given class are identical. By the addition of these auxiliary variables, diverse targets (which vary across inputs of the same class) can be produced for each layer, improving network performance.

Here, we make use of such auxiliary outputs for our full-width network models. We also extend this approach and relax the assumption of full invertibility by allowing auxiliary units at every layer of a network. By placing auxiliary units (which have no forward synaptic connections) at arbitrary layers of our network models, we can build variable-width networks.

Despite these auxiliary units, weight matrices between layers must remain square for inversion of the non-auxiliary neuron activations. This means that the number of non-auxiliary neurons in some layer indexed $l - 1$ is equal to the number of units in the subsequent layer indexed $l$. We can therefore describe the $l - 1$-th layer to have $N_l - 1 - N_l$ auxiliary units, with activations $z_{l - 1}$, and $N_l$ forward projecting neurons with activations $y_{l - 1}$. In a forward model the auxiliary units of a lower layer are ignored such that $y_l = g_l(y_{l - 1})$. But in the inverse pass, we make use of an augmented inverse transfer function $\tilde{g}_l^{-1}(y_l)$ which maps the activations of the $l$-th layer to the tuple $(y_{l - 1}, z_{l - 1})$. Using these variables, we can define the GAIT-prop target as before. That is,

$$t^1_{l - 1} = (1 - \epsilon_{l - 1}) y_{l - 1} + \epsilon_{l - 1} \tilde{g}_l^{-1}(t^1_l).$$  (22)

Note that the values of the auxiliary neuron activations, $z_l$, are simply copied from the forward pass. Furthermore, unless there are additions to the cost-function, the weights mapping $y_l$ to $z_{l + 1}$ do not change during training since the target of the auxiliary variables is always identical to their forward pass values. We can consider this as a case in which auxiliary neurons simply do not receive feedback connections from the task-relevant neurons.

Encouraging Orthogonality One desired feature of networks which we wish to train by GAIT-prop is for weight matrices to be (close to) orthogonal. To encourage orthogonality of the rows of our weight matrices, we make use of a regularizer which can be applied layer-wise. This regularizer can be expressed for the $l$-th weight matrix as

$$\lambda \| W_l W_l^T (J - I) \|_2^2,$$  (23)

where $J$ is an all-ones matrix (i.e. all elements equal to 1), $I$ is the identity matrix, and $\lambda$ modulates the strength of the regularizer relative to the task-related error. When this regularizer is applied,
weight updates are combined with the task-relevant updates and these are collectively scaled by the learning rate, $\eta$. We find that the use of weak regularization is sufficient to ensure that GAIT-propagation remains performant. This regularizer uses non-local (non-plausible) information to enforce orthogonality, however in the Discussion section we explore plausible neural mechanisms that could achieve a similar result.

**Tasks (Datasets)** We make use of three computer vision datasets: MNIST, Fashion-MNIST, and KMNIST. These datasets all consist of $24 \times 24$ (total 784) pixel input images and a label between 1 and 10 indicating the class. We convert the labels into a one-hot vector and during training the quadratic loss between the network output and this one-hot vector is minimized. In the case of TP and GAIT-prop, the one-hot vector is the output layer target.

**Parameters and Learning** The Adam optimiser was used during training of our neural network models. In order to identify acceptable parameters for each of our learning methods, we ran a grid search for the learning rate $\eta$ and the orthogonal regularizer strength $\lambda$. The highest-performing networks were tested for stability and stable high performing parameters were used. Details of specific parameters used and the grid search outcomes are provided in Appendix B.

**Code** For the reproduction of the results presented here, code is available at: [https://github.com/nasiryahm/GAIT-prop](https://github.com/nasiryahm/GAIT-prop).

## 8 Results

![Figure 2: The performance of multi-layer perceptrons trained by BP, TP, and GAIT-prop.](image)

All results in this figure are in networks with a fixed width network: 784 neurons in every layer. **A**: Test accuracies (MNIST) of the algorithms are compared in non-linear networks of various depth. The networks are trained by parameters as determined by a grid search (see Appendix B). **B**: Accuracy of algorithms across tasks (MNIST, KMNIST, and Fashion-MNIST). Non-linear networks with four hidden layers were trained with five repeats (error bars indicate standard deviation). Peak training and test accuracies are presented. **C**: Accuracy of a linear network (with four hidden layers) and no orthogonal regularizers.

Figure 2 presents the accuracy the algorithms we have thus far considered. In particular, we find that GAIT-prop is extremely consistent, with performance indistinguishable from backpropagation. This is true for non-linear networks of various depths (Fig. 2A), when applied to different tasks (Fig. 2B) and for linear networks (Fig. 2C). We find that GAIT-prop is highly robust in general – even capable of training non-linear networks of more than four layers without modification (see Appendix C). By comparison, TP suffers from lower accuracy in non-linear networks (Fig. 2A and C) even across a large choice of training parameters (see Appendix B). Though TP does show high performance and stability in linear networks (Fig. 2C), as expected from our theoretical analyses.

Figure 3 exhibits the performance of networks with variable hidden-layer widths. It can be seen that learning in these networks is as effective for GAIT-prop as it is for backpropragation and the inclusion
of layer-wise auxiliary neurons is empirically found to be successful. Hence, these results show high performance in networks with a relaxed definition of a full inverse.

9 Discussion

Our theoretical work and results show that GAIT-prop produces performance indistinguishable from backpropagation. In comparison TP suffers in non-linear neural networks and, as previously described by Bartunov et al. [25], peak performance is highly dependent upon the training parameters (see Supplementary Material).

It is possible that with alternative activation functions such as the hyperbolic tangent, TP will show improved performance (as suggested by [25]). However, the tanh activation function is not invertible for all real values and our analysis of GAIT-prop shows that, unlike TP, its performance is theoretically independent of the employed activation function.

As explored in this work, a weak orthogonal regularizer is found to enable high and stable performance for GAIT-prop. However, a reader might question how this could arise biologically. We hypothesise that lateral inhibitory learning could aid in sufficient orthogonalization of synaptic weight matrices. As explored previously by King et al. [26], inhibitory plasticity can be used to decorrelate neural outputs. In addition, such simple inhibitory Hebbian plasticity rules have been found to stabilize and balance neural network models, reproducing statistics and observations of cortical activity [27, 28].

One further biologically implausible component of the simulations we have presented are the use of perfect inverse models. In real neural circuits, such an inverse model could be learned by (denoising) auto-encoders, as has been previously attempted with TP [22, 23, 25]. However, the incremental term we have introduced may well become highly sensitive to noise in a network with an imperfect inverse – motivating the use of a relatively large value for this incremental term. Despite this drawback of our current simulation work, our theoretical explorations of the relationship between BP, TP and GAIT-prop required assumption of perfect inverse models and we leave explorations of the non-perfect case to future studies.

In conclusion, we have theoretically and empirically demonstrated that plausible layer-wise targets can be created in a neural network model with (close to) orthogonal weight matrices. The resulting updates lead to learning with almost indistinguishable performance compared to BP. This is accomplished in networks of both fixed and variable layer-widths in a novel application of auxiliary units. Our
work elucidates the relationship between BP and local target-based learning and is a significant step forward in the debate surrounding the plausibility of BP for learning in real neural network models.

References

[1] Donald O Hebb. *The Organization of Behavior: A Neuropsychological Approach*. John Wiley & Sons, 1949.

[2] Henry Markram, Joachim Lübke, Michael Frotscher, and Bert Sakmann. Regulation of synaptic efficacy by coincidence of postsynaptic APs and EPSPs. *Science*, 275(5297):213–215, January 1997.

[3] Wulfram Gerstner, Richard Kempter, J. Leo van van Hemmen, and Hermann Wagner. A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 383(6595):76–81, September 1996.

[4] D E Rumelhart, G E Hinton, and R J Williams. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*, pages 318–362. MIT Press, Cambridge, MA, USA, January 1986.

[5] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[6] Jürgen Schmidhuber. Deep learning in neural networks: an overview. *Neural Netw.*, 61:85–117, January 2015.

[7] Seyed-Mahdi Khaligh-Razavi and Nikolaus Kriegeskorte. Deep supervised, but not unsupervised, models may explain IT cortical representation. *PLoS Comput. Biol.*, 10(11):e1003915, November 2014.

[8] Umut Güçlü and Marcel A J van Gerven. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *J. Neurosci.*, 35(27):10005–10014, July 2015.

[9] Nikolaus Kriegeskorte. Deep neural networks: A new framework for modeling biological vision and brain information processing. *Annu Rev Vis Sci*, 1:417–446, November 2015.

[10] Francis Crick. The recent excitement about neural networks. *Nature*, 337(6203):129–132, January 1989.

[11] Stephen Grossberg. Competitive learning: From interactive activation to adaptive resonance. *Cogn. Sci.*, 11(1):23–63, January 1987.

[12] Arash Samadi, Timothy P Lillicrap, and Douglas B Tweed. Deep learning with dynamic spiking neurons and fixed feedback weights. *Neural Comput.*, 29(3):578–602, March 2017.

[13] Jordan Guerguiev, Konrad P Kording, and Blake A Richards. Spike-based causal inference for weight alignment. *ArXiv*, 1910.01689, October 2019.

[14] Timothy P Lillicrap, Daniel Cownden, Douglas B Tweed, and Colin J Akerman. Random synaptic feedback weights support error backpropagation for deep learning. *Nat. Commun.*, 7:13276, November 2016.

[15] Mohamed Akrout, Collin Wilson, Peter C Humphreys, Timothy Lillicrap, and Douglas Tweed. Deep learning without weight transport. *ArXiv*, 1904.05391, April 2019.

[16] Nasir Ahmad, Luca Ambrogioni, and Marcel A J van Gerven. Spike-timing-dependent inference of synaptic weights. *ArXiv*, 2003.03988, March 2020.

[17] Javier R Movellan. Contrastive Hebbian learning in the continuous Hopfield model. In David S Touretzky, Jeffrey L Elman, Terrence J Sejnowski, and Geoffrey E Hinton, editors, *Connectionist Models*, pages 10–17. Morgan Kaufmann, January 1991.

[18] Randall C O’Reilly. Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm. *Neural Comput.*, 8(5):895–938, July 1996.
The GAIT-prop and ITP targets are implemented as a weak perturbation of the forward pass. This can be implemented through weak feedback connections. We demonstrate this in the following linear firing rate model:

\[
\begin{align*}
\tau u_1(s) &= -u_1(s) + x(s) + \nu W^{-1} u_2(s), \\
\tau u_2(s) &= -u_2(s) + W u_1(s) + t_2(s).
\end{align*}
\] (24)

where \(u_j(s)\) is the firing rate of the \(j\)-th layer at time \(s\), \(\nu\) is a small feedback coupling parameter and \(\tau\) is a time constant. Consider a setting when the input \(x\) is presented \(\delta\) time units before the target \(t\) and then stays present throughout the experiment. If \(\delta \gg \tau\) and \(\nu < 1\), the activity of the first hidden layer, firing rate denoted \(u_1(s)\) above, will converge to the an equilibrium-state value, \(y_1\), which can be expressed:

\[
y_1 = (1 + \frac{\nu}{1 - \nu}) x = (1 + \gamma) x,
\] (25)

where \(\gamma = \frac{\nu}{1 - \nu}\) is our incremental factor. Note that for this incremental factor to remain below 1.0, \(\nu < 0.5\). After the target is presented (i.e. \(t_2\) is a fixed non-zero value), the firing rate will converge.
to a shifted steady-state, \( \tilde{y}_1 \) such that

\[
\tilde{y}_1 = (1 + \frac{\nu}{1 - \nu}) x + \frac{\nu}{1 - \nu} W^{-1} t_2(s) \\
= (1 + \gamma) x + \gamma t_1 \\
= y_1 + \gamma t_1,
\]

(26)

where \( t_1 = W^{-1} t_2 \) and represents the inverted target. If we compare the available terms in the two equilibrium states (\( y_1 \) and \( \tilde{y}_1 \)), these equilibrium points contain sufficient information to make use of the GAIT-prop rule. In particular, the GAIT-prop rule in the linear case (or the ITP rule generally) requires a target of the form \((1 - \gamma) y_l + \gamma t_1\). Such a difference term can be trivially computed using these two equilibrium states. Furthermore, the shifted equilibrium state, \( \tilde{y}_1 \), is as default extremely close to the desired target (the effect of the missing \((1 - \gamma)\) factor is not explored).

For non-linear networks, the computation of the full GAIT-prop target also require the computation of the activity dependent term \( A_l \). We have not described a particular dynamical system in which this could emerge, however this information is local to each unit and therefore remains biologically plausible.

### Appendix B Model parameters

The networks we train in this study all made use of the Adam optimiser [29]. Along side the Adam optimiser parameters, we had parameters related to orthogonal regularization and the incremental component of GAIT-prop. All parameters are outline below, many of these were kept fixed and some were tested in a parameter grid search. The table below presents the relevant parameters.

| Parameter                                    | Value                      |
|----------------------------------------------|----------------------------|
| Learning Rate of Adam Optimiser              | \( \{10^{-2}, 10^{-3}, 10^{-4}\} \) |
| \( \beta_1 \) of Adam Optimiser (fixed)      | 0.9                        |
| \( \beta_2 \) of Adam Optimiser (fixed)      | 0.99                       |
| \( \epsilon \) of Adam Optimiser (fixed)     | \( 10^{-8} \)              |
| Orthogonal Regularizer Strength (\( \lambda \)) | \( \{0, 10^{-1}, 10^{1}, 10^{3}\} \) |
| Incremental Factor for GAIT-prop (\( \gamma \), fixed) | \( 10^{-3} \) |

#### B.1 Learning rate and regularizer grid search

In order to determine favourable parameters for the learning algorithms which we investigated, we ran a grid search over two key parameters: the learning rate, \( \eta \), and the strength of the orthogonal regularizer, \( \lambda \). This parameter search was carried out in a feed-forward square network with 4 hidden layers and a full output layer of 10 output units and 774 auxiliary units. A leaky-ReLu transfer function was used (as is true for all non-linear network results in this paper).

Note that networks were either initialised with orthogonal weight matrices or by Xavier initialisation [30]. Both in this parameter search and in our main paper, Xavier initialization was used for all networks in which \( \lambda = 0.0 \). For all non-zero values of \( \lambda \), networks were initialised with an orthogonal weight matrix.

The results report peak and final (end of training) accuracy on the training set (organise ‘peak / final’). Parameters shown in bold were chosen and used for all results presented in the main paper.

Note that target propagation systematically shows lower accuracy at the end of training compared to at its peak over a large parameter range. We find that target propagation often does best when early-stopping is implemented to ‘catch’ this peak, unlike the other two algorithms which have asymptotic behaviour. Furthermore, the highest performing parameters for target propagation (indicated in italics) were found to be highly unstable when network depth was modified. This was to an extent that reducing network depth caused a counter-intuitive drop in performance. Therefore, we made use of parameters which had much greater stability in performance across network architectures and structure), shown in bold as for the other algorithms.

| Table 1: Backpropagation |  |
Table 2: Target Propagation

| λ    | 0.0       | 0.1       | 10.0      | 1000.0    |
|------|-----------|-----------|-----------|-----------|
| η 1e-3 | 100.00 / 99.98 | 99.93 / 99.78 | 99.59 / 98.80 | 89.23 / 10.44 |
| η 1e-4 | **100.00** / **100.00** | 100.00 / 100.00 | 100.00 / 100.00 | 97.08 / 97.04 |
| η 1e-5 | 100.00 / 100.00 | 99.91 / 99.90 | 99.71 / 99.70 | 96.92 / 96.92 |

Table 3: GAIT Propagation

| λ    | 0.0       | 0.1       | 10.0      | 1000.0    |
|------|-----------|-----------|-----------|-----------|
| η 1e-3 | 17.94 / 10.22 | 21.51 / 9.87 | 90.38 / 11.45 | 86.53 / 7.16 |
| η 1e-4 | 68.11 / 9.74 | 77.5 / 5.57 | 92.02 / 11.10 | 90.53 / 9.38 |
| η 1e-5 | 77.29 / 9.75 | 82.62 / 13.02 | 93.10 / 92.16 | 91.63 / 90.28 |

Appendix C  Performance of GAIT-propagation in deeper networks

In the main paper, we showed that GAIT-propagation produces networks with final training/test accuracies which are indistinguishable from those produced by backpropagation of error. Those results were shown for networks with up to four hidden layers.

Figure 4: The performance of deep multi-layer perceptrons trained by BP, and GAIT-prop. All results in this figure are in networks with a fixed width network: 784 neurons in every layer. Test accuracies (MNIST) of the algorithms are presented here during training in non-linear networks with 6 and 8 hidden layers. The networks use optimal parameters as determined by a grid search (as in Appendix B).

Figure 4 shows that GAIT-prop remains highly performant even in networks with six or eight hidden layers. Performance lags slightly behind that of BP for the eight hidden layer network though it should be expected that in deeper networks our decision to fix the incremental \( \gamma \) parameter would lead to a worse approximation of BP (and therefore a decrease in performance). Nonetheless, GAIT-prop remains robust and shows stable training and high performance despite potential increases in approximation errors in deeper networks.