Fixed-Weight Difference Target Propagation

Tatsukichi Shibuya, [1] Nakamasa Inoue, [1] Rei Kawakami, [1] Ikuro Sato [1,2]

[1] Tokyo Institute of Technology
[2] Denso IT Laboratory
shibuya.t.ad@m.titech.ac.jp, inoue@c.titech.ac.jp, reikawa@sc.e.titech.ac.jp, isato@c.titech.ac.jp

Abstract

Target Propagation (TP) is a biologically more plausible algorithm than the error backpropagation (BP) to train deep networks, and improving practicality of TP is an open issue. TP methods require the feedforward and feedback networks to form layer-wise autoencoders for propagating the target values generated at the output layer. However, this causes certain drawbacks; *e.g.*, careful hyperparameter tuning is required to synchronize the feedforward and feedback training, and frequent updates of the feedback path are usually required than that of the feedforward path. Learning of the feedforward and feedback networks is sufficient to make TP methods capable of training, but is having these layer-wise autoencoders a necessary condition for TP to work? We answer this question by presenting *Fixed-Weight Difference Target Propagation* (FW-DTP) that keeps the feedback weights constant during training. We confirmed that this simple method, which naturally resolves the abovementioned problems of TP, can still deliver informative target values to hidden layers for a given task; indeed, FW-DTP consistently achieves higher test performance than a baseline, the Difference Target Propagation (DTP), on four classification datasets. We also present a novel propagation architecture that explains the exact form of the feedback function of DTP to analyze FW-DTP.

1 Introduction

Artificial Neural Networks (NNs) were introduced to model the information processing in the neural circuits of the brain (McCulloch and Pitts 1943; Rosenblatt 1958). The error backpropagation (BP) has been the most widely used algorithm to optimize parameters of multi-layer NNs with gradient descent (Rumelhart, Hinton, and Williams 1986), but the lack of consistency with neuroscientific findings has been pointed out (Crick 1989; Glorot and Bengio 2010). In particular, the inconsistencies include that 1) in BP, the feedback path is the reversal of the feedforward path in a way that the same synaptic weight parameters are used (a.k.a. weight transport problem (Grossberg 1987)), while the brain most likely uses different sets of parameters in the feedforward and feedback processes; 2) in BP, the layer-to-layer operations are asymmetric between feedforward and feedback processes (*i.e.*, the feedback process does not require the activation used in the feedforward process), while the brain requires symmetric operations. Although there are on-going research efforts to connect the brain and BP (Lillicrap et al. 2020), many researchers seek less inconsistent yet practical algorithms of network training (Lillicrap et al. 2016; Bengio 2014; Lee et al. 2015; Bengio 2020; Meulemans et al. 2020; Ahmad, van Gerven, and Ambrogioni 2020; Scellier and Bengio 2017) because biologically plausible algorithms that may bridge the gap between neuroscience and computer science are believed to enhance machine learning.

Feedback alignment (FA) (Lillicrap et al. 2016) was proposed to resolve the weight transport problem by using fixed random weights for error propagation. It is worth noting that FA has been shown to outperform BP on real datasets, although the results are somewhat outdated (Nøkland 2016; Crafton et al. 2019).

Target propagation (TP) (Bengio 2014; LeCun 1986) has been proposed as a NN training algorithm that can circumvent the inconsistencies 1) and 2). The main idea of TP is to define target values for hidden neurons in each layer in a way that the *target values* (not the *error*) are backpropagated from the output layer down to the first hidden layer, using the same activation function used in the feedforward process. The feedback network, which does *not* share parameters with the feedforward network, is trained so that each layer becomes an approximated inverse of the corresponding layer of the feedforward network, whereas the parameters of the feedforward network are updated to achieve the layer-wise target. In TP, the feedback network ideally realizes layer-wise autoencoders with the feedforward network, but in reality it often ends up with imperfect autoencoders, which could cause optimization problems (Lee et al. 2015; Meulemans et al. 2020). Among the methods that alleviate such small discrepancies (Lee et al. 2015; Bengio 2020), difference target propagation (DTP) (Lee et al. 2015) introduces linear correction terms to the feedback process and significantly improved the recognition performance of TP.

However, while the formalism of DTP to compute layer-wise targets with a feedback network is theoretically sound, training the feedback network is often demanding in the following aspects: a) Synchronous training of the feedforward and feedback networks often requires careful hyperparameter tuning (Bartunov et al. 2018). b) Training of the feedback network could be computationally very expensive. According to previous work (Bartunov et al. 2018; Meulemans et al.

2020; Ernoult et al. 2022), weight updates of the feedback networks were more frequent than those of the feedforward networks. In the latest research (Ernoult et al. 2022), the number of updating feedback weights is set to several tens of times of that of the feedforward weights. For these reasons, training a feedback network typically requires a large cost including hyperparameter tuning.

It is clear that having a relation of layer-wise autoencoders by the feedforward and feedback network is sufficient for the target propagation algorithms to gain training capability. In this work, we aim to answer the question whether constructing layer-wise autoencoders is also a *necessary* condition for target propagation to work. To answer this question, we examined a very simple approach, where the parameters of the feedback network are kept fixed while the feedforward network is trained just as DTP. No reconstruction loss is imposed, so the feedforward and feedback networks are not forced to form autoencoders. Nevertheless, our new target propagation method, *fixed-weight difference target propagation* (FW-DTP), greatly improves the stability of training and test performance compared to DTP while reducing computational complexity from DTP. The idea of fixing feedback weights is inspired by FA, which fixes feedback weights during BP to avoid the weight transport problem. But the difference is that FW-DTP greatly simplifies the learning rule of DTP by removing layer-wise autoencoding losses, whereas FA has no such effect. We provide mathematical expressions about conditions that network trained with DTP will implicitly acquire with and without fixing feedback weights. We further propose a novel propagation architecture which can explicitly provide the exact form of the feedback function of DTP, which implies that FW-DTP acquires implicit autoencoders. It is worth mentioning that Local Representation Alignment (LRA) (Ororbia et al. 2018) (followed by (Ororbia and Mali 2019; Ororbia et al. 2020)) also proposed biologically-plausible layer-wise learning rules with fixed parameters, though it does not belong to the target propagation family.

Our contribution is three-fold: **1)** We propose *fixed-weight difference target propagation* (FW-DTP) that fixes feedback weights and drops the layer-wise autoencoding losses from DTP. Good learnability of FW-DTP indicates that optimizing the objectives of layer-wise target reconstruction is not necessary for the concept of target propagation to properly work. **2)** We present a novel architecture that explicitly shows the exact form of feedback function of DTP, which allows for an accurate notation of how the targets are backpropagated in the feedback network. **3)** We experimentally show that FW-DTP not only improves the stability of training from DTP, but also improves the mean test accuracies from DTP on four image classification datasets just like FA outperforming BP by using fixed backward weights.

2 Overview: Target Propagation Methods

We overview target propagation methods including TP (Bengio 2014) and DTP (Lee et al. 2015).

Definition 2.1 (Feedforward and feedback functions). Let $\mathcal{X}$ and $\mathcal{Y}$ be the input and output spaces, respectively. A feedforward function $F : \mathcal{X} \to \mathcal{Y}$ is defined as a composite function of layered encoders f_l $(l = 1, \cdots L)$ by

$$F(x) = f_L \circ f_{L-1} \circ \cdots \circ f_1(x) \tag{1}$$

where L is the number of encoding layers, $x \in \mathcal{X}$ is an input. A feedback function $G : \mathcal{Y} \to \mathcal{X}$ is defined by

$$G(y) = g_1 \circ g_2 \circ \cdots \circ g_L(y) \tag{2}$$

where $y \in \mathcal{Y}$ is an output and g_l is the l-th decoder. Each g_l is paired with f_l, and will be trained to approximately invert f_l. The feedforward activation h_l is recursively defined as

$$h_l = \begin{cases} x & (l = 0) \\ f_l(h_{l-1}) & (l = 1, \cdots, L) \end{cases} \tag{3}$$

and the target τ_l is recursively defined in the descending order as

$$\tau_l = \begin{cases} y^\star & (l = L) \\ \tilde{g}_{l+1}(\tau_{l+1}) & (l = L-1, \cdots, 0) \end{cases} \tag{4}$$

where $y^\star$ is the output target. In Eq. (4), $\tilde{g}_l$ is an extended function of g_l to propagate targets, and it could be the same as g_l. Note that this paper focuses on supervised learning where loss $\mathcal{L}(F(x), y)$ to be minimized takes finite values over all training pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$.

Target Propagation (TP). TP is an algorithm to learn the feedforward and feedback function where f_l and g_l are parameterized. It defines the output target based on gradient descent (GD) (Bengio 2014) as

$$y^\star(h_L) = h_L - \beta \frac{\partial \mathcal{L}(h_L, y)}{\partial h_L} \tag{5}$$

where β is a nudging parameter. For propagating targets, $\tilde{g}_l = g_l$ is used. TP updates feedforward weights (the parameters of f_l) and feedback weights (the parameters of g_l) alternately. The l-th layer's feedforward weight is updated to reduce layer-wise local loss:

$$L_l = \frac{1}{2\beta} \|h_l - \tau_l\|_2^2 \tag{6}$$

where τ_l is considered as a constant with respect to the l-th layer's feedforward weight, *i.e.*, the gradient of τ_l with respect to the weight is 0. The l-th layer's feedback weight is updated to reduce reconstruction loss:

$$L'_l = \frac{1}{2} \mathbb{E}_{\epsilon \sim \mathcal{N}(0, \sigma^2 I)} \left[\|h_{l-1} + \epsilon - g_l \circ f_l(h_{l-1} + \epsilon)\|_2^2 \right] \tag{7}$$

where ϵ is a small noise to improve the robustness of inversion. A known limitation of TP is that imperfectness of the feedback function as inverse leads to a critical optimization problem (Lee et al. 2015; Meulemans et al. 2020), *i.e.*, the update direction $\tau_l - h_l$ involves reconstruction errors $g_{l+1}(f_{l+1}(h_l)) - h_l$; thus, the feedforward network is not trained properly with an imprecisely optimized feedback network.

Difference Target Propagation (DTP). Lee *et al.* (Lee et al. 2015) show that *difference correction*, subtracting the difference $g_{l+1}(h_{l+1}) - h_l$ from the target, alleviates the limitation of TP, and they introduce DTP, whose function $\tilde{g}_{l+1}$ for propagating targets in Eq. (4) is defined by

$$\tilde{g}_{l+1}(\tau_{l+1}) = g_{l+1}(\tau_{l+1}) + h_l - g_{l+1}(h_{l+1}). \tag{8}$$

The losses for updating feedforward and feedback weights are the same as those of TP. In DTP, assuming all encoders are invertible, the first order approximation of $\Delta h_l := \tau_l - h_l$ is given by

$$\Delta h_l \simeq \left[\prod_{k=l+1}^{L} \frac{\partial g_k(h_k)}{\partial h_k}\right] \Delta h_L \tag{9}$$

$$= \left[\prod_{k=l+1}^{L} J_{g_k}\right] \left(-\beta \frac{\partial \mathcal{L}(h_L, y)}{\partial h_L}\right) \tag{10}$$

$$= -\beta J^{-1}_{f_{l+1:L}} \frac{\partial \mathcal{L}(h_L, y)}{\partial h_L} \tag{11}$$

where $J_{g_k} := \partial g_k(h_k)/\partial h_k$ is the Jacobian matrix of g_k evaluated at h_k. Here, $J_{g_l} = J^{-1}_{f_l}$ $(l = 1, \cdots, L)$ and $J^{-1}_{f_{l+1:L}} = \prod_{k=l+1}^{L} J^{-1}_{f_k}$ are used due to the invertibility, where $J_{f_l} := \partial f_k(h_{k-1})/\partial h_{k-1}$ is the Jacobian matrix of f_k evaluated at h_{k-1}. The notation $()_{a:b}$ is for composing functions from layers a to b, *e.g.*, $f_{l+1:L} = f_L \circ \cdots \circ f_{l+1}$. The update rule of DTP is regarded as a hybrid of GD and Gauss-Newton (GN) algorithm (Gauss 1809). Note that, in the case of the *non-invertible* encoders, DTP obtains the condition $J_{g_l} = J^{+}_{f_l}$ where $J^{+}_{f_l}$ is the Moore-Penrose inverse (Moore 1920; Penrose 1955) of J_{f_l}, however, $J^{+}_{f_{l+1:L}} = \prod_{k=l+1}^{L} J^{+}_{f_k}$ is not always satisfied (Meulemans et al. 2020; Campbell and Meyer 2009).

3 Proposed Method

This section presents the proposed fixed-weight difference target propagation (FW-DTP) that drops the training of feedback weights. We first propose FW-DTP according to the traditional notation (defined in Section 2) in 3.1. We then analyze FW-DTP from two points of view: the conditions for Jacobians in 3.2 and the exact form of the feedback function in 3.3. From these analyses, we explain why fixed-weights of FW-DTP has a good learnability.

3.1 Fixed-Weight Difference Target Propagation

FW-DTP is defined as the algorithm that omits reconstruction loss for updating feedback weights in DTP. All feedback weights are first randomly initialized and then fixed during training. For example, with a fully connected network, the l-th encoder and decoder of FW-DTP are defined by

$$f_l(h_{l-1}) := \sigma_l(W_l h_{l-1}), \;\; g_l(\tau_l) := \sigma'_l(B_l \tau_l) \tag{12}$$

where σ and σ' are non-linear activation functions and W_l and B_l are matrices which denote the feedforward and feedback weights, respectively. B_l is first initialized with a distribution $P(B_l)$ and then fixed, while W_l is updated in the learning process. The feedback propagation of targets are defined by Eq. (8). Note that DTP asymptotically approaches FW-DTP by decreasing the learning rate of the feedback weights.

3.2 Analysis 1: Condition for Jacobians

Here, we discuss conditions for DTP to appropriately work. Given that precise inverse relation between f_l and g_l may not be always obtainable in DTP, training with inaccurate targets can degrade the overall performance of the feedforward function. Now, consider two directions $\tau_l - h_l$ and $f_l(h^*_{l-1}) - h_l$, a vector from the activation h_l to the target τ_l at layer l, and another from h_l to the point $f_l(h^*_{l-1})$. If the condition

$$-\frac{\pi}{2} \leq \angle(\tau_l - h_l, f_l(h^*_{l-1}) - h_l) \leq \frac{\pi}{2} \;\text{ where }\; h^*_{l-1} = \tau_{l-1} \tag{13}$$

holds, *i.e.*, if the angle between them is within 90 degrees, the loss of this sample is expected to decrease because $f_l(h^*_{l-1})$ is the best point achieved by learning $(l-1)$-th encoder. By applying the first order approximation, Eq. (13) is rewritten as

$$\Delta h_l^\top J_{f_l} J_{g_l} \Delta h_l \geq 0 \tag{14}$$

therefore, the sufficient condition of Eq. (13) is that $J_{f_l} J_{g_l}$ is a positive semi-definite matrix. As Table 1 shows, minimization of reconstruction losses of DTP such as original DTP (Eq. (7)), difference reconstruction loss (DRL) (Meulemans et al. 2020) and local difference reconstruction loss (L-DRL) (Ernoult et al. 2022) naturally satisfy the positive semi-definiteness by enforcing the Jacobian matrix J_{g_l} as the inverse or transpose of J_{f_l}. On the other hand, positive semi-definiteness requires

$$\inf_{\epsilon} \left[\epsilon^\top J_{f_l} J_{g_l} \epsilon\right] \geq 0 \tag{15}$$

however, this condition could be somewhat too strict, given that features may not always span the full space. In FW-DTP, the strict condition expressed in Eq. (15) is not generally satisfied because FW-DTP has no feedback objective function to learn to explicitly satisfy this condition. Now, let us consider a hypothetical situation where the product of Jacobians satisfies the condition,

$$\mathbb{E}_{\epsilon \sim p(\cdot)} \left[\epsilon^\top J_{f_l} J_{g_l} \epsilon\right] \geq 0 \tag{16}$$

where the infimum in Eq. (15) is replaced with the expectation over some origin-centric rotationally-symmetric distribution $p(\cdot)$ such as a zero-mean isotropic Gaussian distribution. Then, it is straightforward to show that Eq. (16) is equivalent to

$$\mathrm{tr}(J_{f_l} J_{g_l}) \geq 0. \tag{17}$$

The condition expressed in Eq. (17) is weaker than Eq. (15). Under the condition of Eq. (17), if $(l-1)$-th activation moves toward the target, it will shifts l-th activation toward the corresponding target within $\pi/2$ range as the expectation (over p). Although the condition of Eq. (17) is somewhat artificial, but indeed we found that FW-DTP does satisfy this condition in our experiment as we show in Section 4. The condition expressed in Eq. (17) could be regarded as a type of alignments that the network implicitly acquires when its feedback weights are fixed during DTP updates.

Table 1: The conditions of the Jacobians obtained by various reconstruction losses and FW-DTP.

METHOD	DTP	DRL	L-DRL	FW-DTP
CONDITION	$J_{g_l} = J_{f_l}^{+}$	$\prod_{k=l}^{L} J_{g_k} = J_{f_{l:L}}^{+}$	$J_{g_l} = J_{f_l}^{\top}$	$\mathrm{tr}(J_{f_l} J_{g_l}) > 0$

3.3 Analysis 2: Exact Form of Feedback Function

To show how targets are propagated in FW-DTP, we present a propagation architecture which provides the exact form of the feedback function of DTP. There exists no autoencoders in FW-DTP at least explicitly; however, difference correction creates autoencoders implicitly. To explicitly show this, instead of using the function $\tilde{g}_l$ for propagating targets in Eq (4), we decomposed encoder and decoder as $f_l = f_l^{\nu} \circ f_l^{\mu}$ and $g_l = g_l^{\nu} \circ g_l^{\mu}$ to incorporate the difference correction mechanism into g_l^{ν}. Using the proposed architecture represented by Eqs. (18-20), TP and DTP are reformulated as Eqs. (22-30) and the training process is also reformulated as Eq. (21).

Definition 3.1 (Propagation Architecture). We define a feedforward function $F : \mathcal{X} \to \mathcal{Y}$ with encoders f_l and a feedback function $G : \mathcal{Y} \to \mathcal{X}$ with decoders g_l by Eqs. (1-2). The targets are recursively defined in the descending order as

$$\tau_l = \begin{cases} y^{\star} & (l = L) \\ g_{l+1}(\tau_{l+1}) & (l = L-1, \cdots, 0) \end{cases} \tag{18}$$

where $y^{\star}$ is the output target. Eq (18) differs from Eq (4) in that we avoid to define $\tilde{g}_l$. Further, we introduce four functions $f_l^{\mu}, f_l^{\nu}, g_l^{\mu}, g_l^{\nu}$ that decompose the encoder and decoder into

$$f_l = f_l^{\nu} \circ f_l^{\mu}, \quad g_l = g_l^{\nu} \circ g_l^{\mu}. \tag{19}$$

We also define a shortcut function ψ_l that map the activation to the target as

$$\psi_l(h_l) = \begin{cases} \tau_L & (l = L) \\ g_{L:l+1} \circ \psi_L \circ f_{l+1:L}(h_l) & (l = L-1, \cdots, 0). \end{cases} \tag{20}$$

Here, $\psi_l(h_l) = \tau_l$. Figure 1a illustrates the proposed propagation architecture. With this architecture, we expect that $g_l \circ f_l$ will become an autoencoder after convergence with the activations sufficiently close to the corresponding targets . It is reduced to DTP when f_l^{μ} is the identity function, f_l^{ν} is a parameterized function (*e.g.*, $f_l^{\nu}(h_{l-1}) = \sigma(W_l h_{l-1})$), g_l^{μ} is another parameterized function, and g_l^{ν} is a function of difference correction, as shown in Figure 1b and 1c. Note that Figure 1c is a well-known visualization of DTP (Lee et al. 2015). The main problem we would like to discuss is whether there exists the exact form of g_l^{ν}. With the traditional notations in Eq. (8), $\tilde{g}_{l+1}$ is defined as a function of τ_{l+1}, however, it uses h_l and h_{l+1} in the right side of the equation. This makes it difficult to analyze the shape of feedback function; thus, we define the training process here as follows.

Definition 3.3 (Training). Let $\mathfrak{q}_l = (f_l^{\mu}, f_l^{\nu}, g_l^{\mu}, g_l^{\nu})$ a quadruplet of functions. We define training as the process to solve the following layer-wise problem:

$$\mathfrak{q}_l^{*} = \operatorname*{argmin}_{\mathfrak{q}_l \in \mathfrak{Q}_l} \mathcal{O}_l \tag{21}$$

where $\mathfrak{Q}_l = \mathcal{F}_l^{\mu} \times \mathcal{F}_l^{\nu} \times \mathcal{G}_l^{\mu} \times \mathcal{G}_l^{\nu}$ is a function space (search space), and $\mathcal{O}_l$ is the objective function.

This definition involves TP and DTP variants as follows.

Target Propagation. Using the proposed architecture, TP is defined as a training process with the search spaces:

$$\mathcal{F}_l^{\mu} = \{id\}, \; \mathcal{F}_l^{\nu} = \{p_{\theta} : \theta \in \Theta_l\} \tag{22}$$

$$\mathcal{G}_l^{\mu} = \{p_{\omega} : \omega \in \Omega_l\}, \; \mathcal{G}_l^{\nu} = \{id\} \tag{23}$$

where *id* is the identity function and p_{θ} and p_{ω} are parameterized functions with learnable parameters θ and ω, respectively. Θ_l and Ω_l are the parameter spaces. TP solves Eq. (21) by alternately solving two problems:

$$f_l^{\nu *} = \operatorname*{argmin}_{f_l^{\nu} \in \mathcal{F}_l^{\nu}} \mathcal{O}_l^{(1)} \tag{24}$$

$$g_l^{\mu *} = \operatorname*{argmin}_{g_l^{\mu} \in \mathcal{G}_l^{\mu}} \mathcal{O}_l^{(2)} \tag{25}$$

where $\mathcal{O}_l^{(1)}$ is the layer-wise local loss in Eq. (6) and $\mathcal{O}_l^{(2)}$ is the reconstruction loss in Eq. (7).

Difference Target Propagation. DTP is also defined with a search space $\mathfrak{G}_l$ for g_l^{ν} as follows:

$$\mathcal{F}_l^{\mu} = \{id\}, \; \mathcal{F}_l^{\nu} = \{p_{\theta} : \theta \in \Theta_l\} \tag{26}$$

$$\mathcal{G}_l^{\mu} = \{p_{\omega} : \omega \in \Omega_l\}, \; \mathcal{G}_l^{\nu} = \mathfrak{G}_l \tag{27}$$

$$\text{where} \quad \mathfrak{G}_l = \{g_l^{\nu} : d_P(f_l^{\mu} \circ g_l \circ \psi_l \circ f_l, \; g_l^{\mu} \circ \psi_l \circ f_l + f_l^{\mu} - g_l^{\mu} \circ f_l) = 0\} \tag{28}$$

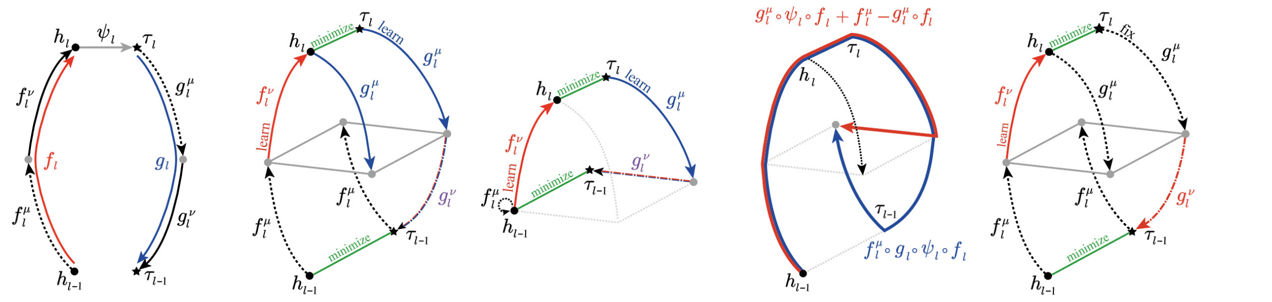
and d_P with norm P (*e.g.*, L_2 norm) is a distance in the function space.

Figure 1d shows the two functions $f_l^{\mu} \circ g_l \circ \psi_l \circ f_l$ and $g_l^{\mu} \circ \psi_l \circ f_l + f_l^{\mu} - g_l^{\mu} \circ f_l$ in blue and red, respectively; namely, $\mathfrak{G}_l$ is the function subspace of g_l^{ν} where these two functions (the blue and red arrows in 1d) are equal. By assuming functions $f_l^{\nu}, \psi_l, g_l^{\mu}$ are bijective, we have $\mathfrak{G}_l = \{\breve{g}_l^{\nu}\}$ where

$$\breve{g}_l^{\nu} = id + (f_l^{\nu})^{-1} \circ (\psi_l)^{-1} \circ (g_l^{\mu})^{-1} - g_l^{\mu} \circ (\psi_l)^{-1} \circ (g_l^{\mu})^{-1}. \tag{29}$$

This is the exact form of difference correction in our formulation. This shows that g_l^{ν} is implicitly updated by updating f_l^{ν} and g_l^{μ}. Therefore, DTP solves Eq. (21) by alternately solving two problems:

$$(f_l^{\nu *}, g_l^{\nu *}) = \operatorname*{argmin}_{(f_l^{\nu}, g_l^{\nu}) \in \mathcal{F}_l^{\nu} \times \mathcal{G}_l^{\nu}} \mathcal{O}_l^{(1)}, \; (g_l^{\mu *}, g_l^{\nu *}) = \operatorname*{argmin}_{(g_l^{\mu}, g_l^{\nu}) \in \mathcal{G}_l^{\mu} \times \mathcal{G}_l^{\nu}} \mathcal{O}_l^{(2)} \tag{30}$$



(a) Proposed Architecture (b) DTP (c) DTP (reduced visualization) (d) $\mathfrak{G}_l$ in DTP (e) FW-DTP

Figure 1: Proposed propagation architecture and its reduction to DTP. (a) The proposed architecture. The encoder f_l is decomposed into f_l^μ and f_l^ν. The decoder g_l is decomposed into g_l^μ and g_l^ν. ψ_l is the shortcut function from an activation h_l to the target τ_l. (b) Reduction to DTP. g_l^ν is a function of difference correction. f_l^μ is illustrated as non-identity function. (c) Reduction to DTP, where f_l^μ is illustrated as the identity function. This is the well-known visualization of DTP. (d) The search space $\mathfrak{G}_l$ for g_l^ν. (e) FW-DTP with fixed g_l^μ.

where the objective function is the same as that of TP. Eq. (30) indicates that updating the feedforward weights implicitly update g_l^ν in the feedback path.

Fixed-Weight Difference Target Propagation. From Eq. (29), we notice that *DTP works even with fixed* g_l^μ because g_l^ν is updated in conjunction with f_l^ν. If the function space $\mathcal{F}_l^\nu$ is large enough for finding an appropriate pair of f_l^ν and g_l^ν, parametrization of the two function spaces $\mathcal{F}_l^\nu$ and $\mathcal{G}_l^\mu$ may be redundant. Based on this observation, FW-DTP uses a unit set for $\mathcal{G}_l^\mu$:

$$\mathcal{F}_l^\mu = \{id\},\ \mathcal{F}_l^\nu = \{p_\theta : \theta \in \Theta_l\},\ \mathcal{G}_l^\mu = \{r_l\},\ \mathcal{G}_l^\nu = \mathfrak{G}_l \tag{31}$$

where r_l is a fixed random function. FW-DTP solves Eq. (21) by solving one problem:

$$(f_l^{\nu *}, g_l^{\nu *}) = \underset{(f_l^\nu, g_l^\nu) \in \mathcal{F}_l^\nu \times \mathcal{G}_l^\nu}{\operatorname{argmin}}\ \mathcal{O}_l^{(1)}. \tag{32}$$

Figure 1e shows that in FW-DTP, g_l^μ is fixed but g_l^ν colored in red moves with f_l^ν, and thus there still exists an autoencoder $g_l \circ f_l$. This is one of the reasons why FW-DTP has an ability to propagate targets to decrease loss. To keep nonlinearity and the ability to entangle elements from different dimension on the feedback path, $r_l(a) = \sigma(B_l a)$ would be the simplest choice where B_l is a random matrix fixed before training and σ is a non-linear activation function. FW-DTP is more efficient than DTP because it reduces the number of learnable parameters.

4 Experiments

In this section, we show experimental results. First, we show that the weak condition expressed in Eq. (16) is satisfied by FW-DTP experimentally. We then compare FW-DTP with TP and DTP variants. Lastly, we evaluate the hyperparameter sensitivity and computational cost, and show that FW-DTP is more stable and computationally efficient than DTP.

4.1 Weak and Strict Conditions of Jacobians

Experimental set-up. This experiment aims to show that FW-DTP satisfies the weak condition of Jacobians given by Eq. (16) during its training process. We also show that FW-DTP does not satisfy the strict condition expressed in Eq. (15) in contrast to DTP.

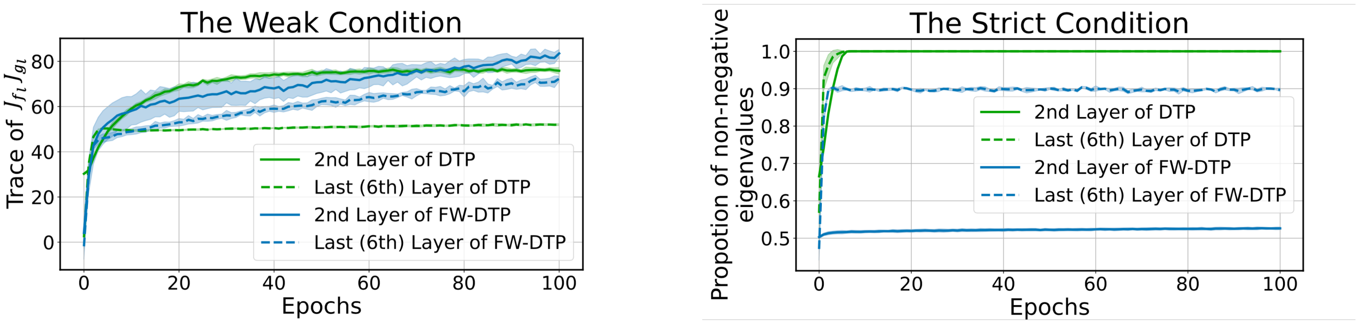
Evaluation details are as follows. For the weak condition, we directly measured the trace of $J_{f_l} J_{g_l}$ (With notations in Analysis 2, this is $J_{f_l^\nu} J_{g_l^\mu}$). For the strict condition, we measured the proportion of the number of non-negative eigenvalues of $J_{f_l} J_{g_l}$ to its dimension. This is a measure of positive semi-definiteness. The MNIST dataset (Lecun et al. 1998) was used for this evaluation. A fully connected network with 6 layers each with 256 units was trained with cross-entropy loss. Note that the first and the last encoders are non-invertible due to the difference of the input and output dimensions. We chose the hyperbolic tangent as the activation function, but only for FW-DTP, batch normalization (BN) (Ioffe and Szegedy 2015) was applied after each hyperbolic tangent. Stochastic gradient descent (SGD) was used as the optimizer. The feedforward and feedback weights were initialized with random orthogonal matrices and random numbers from uniform distribution $U(-0.01, 0.01)$, respectively.

Results. Figure 2 shows the results of the last (sixth) layer and the second layer as representatives of intermediate layers. In Figure 2a, we see that the trace of $J_{f_l} J_{g_l}$ is positive from the first epoch, and is increasing during training process of DTP and FW-DTP. In contrast, in Figure 2b, we see the difference between DTP and FW-DTP. With DTP, all eigenvalues are non-negative after the tenth epoch on both layers. On the other hand, with FW-DTP, some of eigenvalues are negative. We see that $\approx 90\%$ of eigenvalues are non-negative in the last layer, but only $\approx 53\%$ of them are non-negative in the second layer.

These results confirm that FW-DTP satisfies only the weak condition expressed in Eq. (16) automatically, while DTP satisfies both of the weak and strict conditions.

4.2 Comparison with TP and DTP Variants

Experimental set-up. The purpose of this experiment is to demonstrate that the performance of FW-DTP is comparable



(a) Trace of $J_{f_l} J_{g_l}$

(b) Positive semi-definiteness of $J_{f_l} J_{g_l}$

Figure 2: The Jacobian conditions of FW-DTP and DTP on MNIST with the mean and standard deviation over five different seeds. (a) Trace of $J_{f_l} J_{g_l}$ (the values of the trace on the 2nd layer of DTP are scaled by 0.1). We see that all values are positive. (b) The proportion of non-negative eigenvalues. We see the difference between DTP and FW-DTP.

with or even better than that of DTP. We compared image classification performance of TP (Bengio 2014), DTP (Lee et al. 2015), DRL (Meulemans et al. 2020), L-DRL (Ernoult et al. 2022), and FW-DTP on four datasets: MNIST (Lecun et al. 1998), Fashion-MNIST (F-MNIST) (Xiao, Rasul, and Vollgraf 2017), CIFAR-10 and CIFAR-100 (Krizhevsky 2009). Following previous studies (Bartunov et al. 2018; Meulemans et al. 2020), a fully connected network consists of 6 layers each with 256 units was used for MNIST and F-MNIST. Another fully connected network consists of 4 layers each with 1,024 units was used for CIFAR-10/100. Because FW-DTP halves the number of the learnable parameters by fixing the feedback weights, we also report results with a half number of leanable parameters with DTP, DRL and L-DRL. The activation function and the optimizer were the same as those used in 4.1.

Results. The results are summarized in Table 2. As can be seen, FW-DTP is comparable with DTP and its variants. FW-DTP outperformed DTP in all datasets. This supports that FW-DTP works as a training algorithm even if it does not satisfy the strict condition of Jacobians. This also confirms that even with fixed feedback weights, FW-DTP propagates targets to decrease cross-entropy loss via the feedback path with the function g_l^ν for difference correction. Comparison with DRL and L-DRL showed some limitation of FW-DTP. FW-DTP outperformed them on MNIST, F-MNIST, and CIFAR-10 when the number of learnable parameters was the same. On CIFAR-100, the test error of FW-DTP was not the best among them. However, when the number of parameters was the same, the difference in the test error between DTP and DRL or L-DRL was only $\leq 0.1\%$. Note that the goal of this study is not to outperform them but to analyze how and why FW-DTP works as a training algorithm with empirical evidence.

4.3 Hyperparameter Sensitivity and Computational Efficiency

Here, we investigate hyperparameter sensitivity and the computational cost of FW-DTP to show that FW-DTP alleviates the problems of DTP such as hyperparameter instability and high computational complexity.

Hyperparameter sensitivity. We investigate how sensitive DTP and FW-DTP are to different hyperparameters. Namely, we tested 100 different random configurations. More specifically, denoting by $\alpha \in \mathbb{R}^H$ the flattened hyperparameters where H is the number of hyperparameters, each α_i was randomly sampled so that $\log(\alpha_i) \sim U(\log(0.2\bar{\alpha}_i), \log(5\bar{\alpha}_i))$ where U is the uniform distribution and $\bar{\alpha}$ is the hyperparameter used in 4.2. The histograms of the test accuracies on CIFAR-10 are visualized in Figure 3. As can be seen, FW-DTP is less sensitive than DTP to hyperparameters. This is because DTP needs the complicated interactions between feedforward and feedback training, as discussed in the previous work (Bartunov et al. 2018), while FW-DTP drops these complexities by relaxing the conditions of Jacobians from the strict one to the weak one.

Computational Cost. We compare the computational cost of each method on CIFAR-10 in Table 3. 4 GPUs (Tesla P100-SXM2-16GB) with 56 CPU cores are used to measure computational time. For DTP, DRL and L-DRL, the feedback weights are updated five times in each iteration. FW-DTP is $\approx$ 3.0 times slower than BP and $>$ 3.7 times faster than DTP. This shows that BP is still better in terms of computational cost, however, FW-DTP is one of the most efficient methods in DTP variants.

5 Discussion

In this paper, we proposed FW-DTP, which fixes feedback weights during training, and experimentally confirmed that its test performance is consistently better than that of DTP on four image-classification datasets, while the hyperparameter sensitivity and the computational cost are reduced. Further, we showed the strict and weak conditions of Jacobians, by which we explained the difference between FW-DTP and DTP. Finally, we discuss limitations and future work.

Biological plausibility. A limitation of FW-DTP is that it does not fulfill some biological constraints such as Dale's law (Parisien, Anderson, and Eliasmith 2008) and spiking networks (Samadi, Lillicrap, and Tweed 2017; Guerguiev, Lillicrap, and Richards 2017; Bengio et al. 2008). We have shown in Analysis 2 that the composite function $f_l \circ g_l$ forms a layer-wise autoencoder even with fixed feedback weights

Table 2: Test error (%) obtained on four image classification datasets reported with the mean and standard deviation over five different seeds. For the hyperparameter search, 5,000 samples from the training set are used as the validation set. The best and the second best results are marked in bold and with an underline, respectively. The columns of #PARAMS is the number of learnable parameters (the sum of numbers of feedforward and the feedback networks).

METHODS	#PARAMS	MNIST	F-MNIST	#PARAMS	CIFAR-10	CIFAR-100
BP	0.5M	$1.85_{\pm 0.09}$	$10.42_{\pm 0.08}$	6.3M	$46.16_{\pm 1.15}$	$75.96_{\pm 0.52}$
FA (LILLICRAP ET AL. 2016)	0.5M	$2.94_{\pm 0.09}$	$12.58_{\pm 0.35}$	6.3M	$51.33_{\pm 0.81}$	$77.43_{\pm 0.21}$
TP	1.1M	$78.99_{\pm 2.04}$	–	13.0M	–	–
DTP (LEE ET AL. 2015)	0.5M	$3.24_{\pm 0.15}$	$11.86_{\pm 0.14}$	6.3M	$52.17_{\pm 0.79}$	$77.89_{\pm 0.39}$
	1.1M	$2.77_{\pm 0.10}$	$11.77_{\pm 0.16}$	13.0M	$52.01_{\pm 0.80}$	$77.11_{\pm 0.20}$
DRL (MEULEMANS ET AL. 2020)	0.5M	$3.13_{\pm 0.03}$	$12.75_{\pm 0.52}$	6.3M	$50.11_{\pm 0.67}$	$76.69_{\pm 0.30}$
	1.1M	$2.84_{\pm 0.09}$	$12.15_{\pm 0.25}$	13.0M	$\mathbf{48.79_{\pm 0.58}}$	$75.62_{\pm 0.35}$
L-DRL (ERNOULT ET AL. 2022)	0.5M	$3.14_{\pm 0.03}$	$12.45_{\pm 0.36}$	6.3M	$49.58_{\pm 0.33}$	$76.72_{\pm 0.26}$
	1.1M	$2.82_{\pm 0.10}$	$12.29_{\pm 0.46}$	13.0M	$49.84_{\pm 0.55}$	$\mathbf{75.62_{\pm 0.31}}$
FW-DTP	0.5M	$\mathbf{2.76_{\pm 0.10}}$	$\mathbf{11.76_{\pm 0.37}}$	6.3M	$48.97_{\pm 0.32}$	$76.76_{\pm 0.45}$

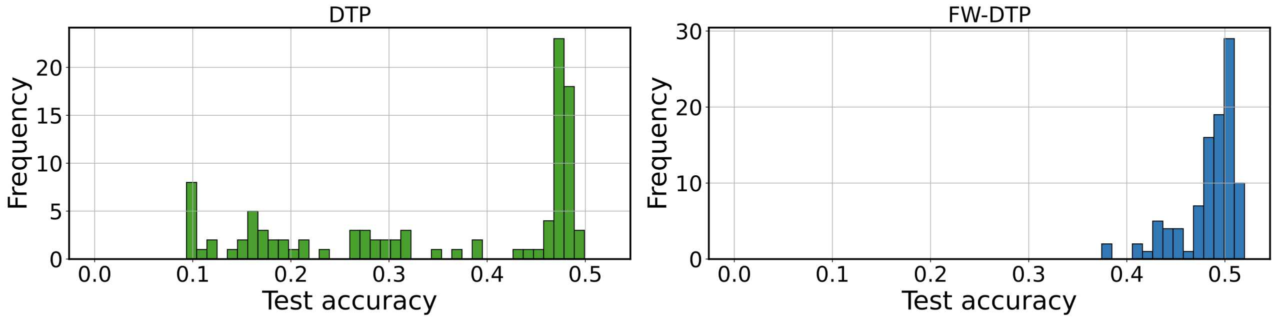


Figure 3: Histogram of test accuracies achieved under different hyperparameters on CIFAR-10.

Table 3: Training time [sec] per epoch of FW-DTP, DTP, DRL, L-DRL and BP on CIFAR-10.

	TIME[SEC]	RATIO TO FW-DTP	ERROR[%]
FW-DTP	$2.22_{\pm 0.02}$	$1.00_{\pm 0.00}$	$48.97_{\pm 0.32}$
DTP	$8.32_{\pm 0.36}$	$3.74_{\pm 0.17}$	$52.01_{\pm 0.80}$
DRL	$9.52_{\pm 0.08}$	$4.29_{\pm 0.05}$	$48.79_{\pm 0.58}$
L-DRL	$8.86_{\pm 0.08}$	$3.99_{\pm 0.05}$	$49.84_{\pm 0.55}$
BP	$0.76_{\pm 0.03}$	$0.34_{\pm 0.01}$	$46.16_{\pm 1.15}$

because we have a function g_l^ν derived from difference correction. However, allowing $g_l^\nu \neq id$ may harm biological plausibility. Notably, this is not a problem only for FW-DTP. If we apply DTP to a non-injective feedforward function, a non-identity function g_l^ν often remains. We hope our exact formulation of DTP helps researchers to analyze the behaviour of DTP in future.

Scalability. Another limitation in this work is that all of the four datasets are for image classification and are relatively small. We chose them because of two reasons: 1) they are suitable for analyzing Jacobian matrices during training to see the difference between FW-DTP and DTP, and 2) they are suitable for repeating many experiments with different hyper-parameters for evaluating the sensitivity. Recently, some improved targets propagated beyond layers (Meulemans et al. 2020; Ernoult et al. 2022) perform comparable with BP on large-scale datasets. From the point of view of fixed feedback weights, these methods may be related to the direct feedback alignment (Nøkland 2016; Crafton et al. 2019). Exploring a method to add such feedback paths efficiently with some fixed feedback weights would be an interesting and necessary direction for future work.

New research direction. In this study, we assumed $f_l^\mu = id$ in the decomposed encoder for a fair comparison of FW-DTP with DTP and its variants. However, it is worth noting that exploring non-identify fixed function f_l^μ, as well as exploring different restrictions to the function space $\mathfrak{O}_l$ would open a new research direction. In particular, the following symmetry in FW-DTP would be effective to explore new biologically plausible function families: f_l^μ, g_l^μ are fixed, and f_l^ν, g_l^ν are determined by a parameter θ. This direction includes research topics about how to fix weights in conjunction with feedback alignment methods (Crafton et al. 2019; Moskovitz, Litwin-Kumar, and Abbott 2018; Garg and Vempala 2022), and how to parameterize paired functions with some reparametrization tricks. Under the weak condition of Jacobians, there must be fruitful function families that have never been investigated for propagating targets.

Acknowledgement

This work was an outcome of a research project, Development of Quality Foundation for Machine-Learning Applications, supported by DENSO IT LAB Recognition and Learning Algorithm Collaborative Research Chair (Tokyo Tech.). This work was also supported by JSPS KAKENHI Grant Number JP22H03642.

References

Ahmad, N.; van Gerven, M.; and Ambrogioni, L. 2020. GAIT-prop: A biologically plausible learning rule derived from backpropagation of error. In *NeurIPS*.

Bartunov, S.; Santoro, A.; Richards, B.; Marris, L.; Hinton, G.; and Lillicrap, T. 2018. Assessing the Scalability of Biologically-Motivated Deep Learning Algorithms and Architectures. In *NeurIPS*.

Bengio, Y. 2014. How auto-encoders could provide credit assignment in deep networks via target propagation. *arXiv preprint arXiv:1407.7906*.

Bengio, Y. 2020. Deriving Differential Target Propagation from Iterating Approximate Inverses. *arXiv preprint arXiv:2007.15139*.

Bengio, Y.; Mesnard, T.; Fischer, A.; Zhang, S.; and Wu, Y. 2008. STDP-Compatible Approximation of Backpropagation in an Energy-Based Model. *Neural computation*.

Campbell, S. L.; and Meyer, C. D. 2009. *Generalized inverses of linear transformations*. SIAM.

Crafton, B.; Parihar, A.; Gebhardt, E.; and Raychowdhury, A. 2019. Direct feedback alignment with sparse connections for local learning. *Frontiers in Neuroscience*, 13.

Crick, F. 1989. The recent excitement about neural networks. *Nature*, 337: 129–132.

Ernoult, M.; Normandin, F.; Moudgil, A.; Spinney, S.; Belilovsky, E.; Rish, I.; Richards, B.; and Bengio, Y. 2022. Towards Scaling Difference Target Propagation by Learning Backprop Targets. In *ICML*.

Garg, S.; and Vempala, S. S. 2022. How and When Random Feedback Works: A Case Study of Low-Rank Matrix Factorization. *arXiv preprint arXiv:2111.08706*.

Gauss, C. F. 1809. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*, volume 7. Perthes et Besser.

Glorot, X.; and Bengio, Y. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*.

Grossberg, S. 1987. Competitive learning: From interactive activation to adaptive resonance. *Cognitive Science*, 11: 23–63.

Guerguiev, J.; Lillicrap, T. P.; and Richards, B. A. 2017. Towards deep learning with segregated dendrites. *ELife*, 6.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *ICML*.

Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. *Technical Report*.

LeCun, Y. 1986. Learning processes in an asymmetric threshold network. *Disordered systems and biological organization*.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *IEEE*, 86: 2278–2324.

Lee, D.-H.; Zhang, S.; Fischer, A.; and Bengio, Y. 2015. Difference Target Propagation. In *ECML/PKDD*.

Lillicrap, T.; Cownden, D.; Tweed, D.; and Akerman, C. 2016. Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications*, 7.

Lillicrap, T.; Santoro, A.; Marris, L.; Akerman, C.; and Hinton, G. 2020. Backpropagation and the brain. *Nature*, 21: 335–346.

McCulloch, W. S.; and Pitts, W. 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of mathematical biophysics*, 5: 115–133.

Meulemans, A.; Carzaniga, F. S.; Suykens, J. A.; Sacramento, J.; and Grewe, B. F. 2020. A theoretical framework for target propagation. In *NeurIPS*.

Moore, E. H. 1920. On the reciprocal of the general algebraic matrix. *Bulletin of American Mathematical Society*, 26: 394–395.

Moskovitz, T. H.; Litwin-Kumar, A.; and Abbott, L. 2018. Feedback alignment in deep convolutional networks. *arXiv preprint arXiv:1812.06488*.

Nøkland, A. 2016. Direct Feedback Alignment Provides Learning in Deep Neural Networks. In *NeurIPS*.

Ororbia, A. G.; and Mali, A. 2019. Biologically Motivated Algorithms for Propagating Local Target Representations. In *AAAI*.

Ororbia, A. G.; Mali, A.; Giles, C. L.; and Kifer, D. 2020. Continual Learning of Recurrent Neural Networks by Locally Aligning Distributed Representations. *IEEE Transactions on Neural Networks and Learning Systems*.

Ororbia, A. G.; Mali, A.; Kifer, D.; and Giles, C. L. 2018. Conducting Credit Assignment by Aligning Local Representations. *arXiv preprint arXiv:1803.01834*.

Parisien, C.; Anderson, C. H.; and Eliasmith, C. 2008. Solving the problem of negative synaptic weights in cortical models. *Neural computation*, 20: 1473–1494.

Penrose, R. 1955. A generalized inverse for matrices. *Mathematical proceedings of the Cambridge philosophical society*, 51: 406–413.

Rosenblatt, F. 1958. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review*, 65(6): 386–408.

Rumelhart, D. E.; Hinton, G. E.; and Williams, R. J. 1986. Learning representations by back-propagating errors. *Nature*, 323: 533–536.

Samadi, A.; Lillicrap, T. P.; and Tweed, D. B. 2017. Deep learning with dynamic spiking neurons and fixed feedback weights. *Neural computation*.

Scellier, B.; and Bengio, Y. 2017. Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation. *Frontiers in computational neuroscience*, 11.

Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.