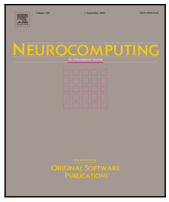
Neurocomputing 611 (2025) 128587

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



The underlying mechanisms of alignment in error backpropagation through arbitrary weights

Alireza Rahmansetayesh, Ali Ghazizadeh *, Farokh Marvasti

Electrical Engineering Department, Sharif University of Technology, Tehran, Iran

ARTICLE INFO

Communicated by M. Bianchini

Dataset link: http://yann.lecun.com/exdb/mnist/, https://github.com/ARahmansetayesh/The-underlying-mechanisms-of-alignment-in-error-backpropagation-through-arbitrary-weights

Keywords:
Backpropagation
Feedback alignment
Weight transport problem
Bio-inspired artificial neural network
Weight normalization

ABSTRACT

Understanding the mechanisms by which plasticity in millions of synapses in the brain is orchestrated to achieve behavioral and cognitive goals is a fundamental question in neuroscience. In this regard, insights from learning methods in artificial neural networks (ANNs) and in particular supervised learning using backpropagation (BP) seem inspiring. However, the implementation of BP requires exact matching between forward and backward weights, which is unrealistic given the known connectivity pattern in the brain (known as the "weight transport problem"). Notably, it has been shown that under certain conditions, error BackPropagation Through Arbitrary Weights (BP-TAW) can lead to a partial alignment between forward and backward weights (weight alignment or WA). This learning algorithm, which is also known as feedback alignment (FA), can result in surprisingly good degrees of accuracy in simple classification tasks. However, the underlying mechanisms and mathematical basis of WA are not thoroughly understood. In this work, we demonstrate the mathematical basis of WA and answer the question of why and in what conditions WA occurs. We show that the occurrence of WA in ANNs is induced by statistical properties of the output and error signals of neurons, such as autocorrelation and cross-correlation, and can happen even in the absence of learning or reduction of the loss function. Moreover, we show that WA can be improved significantly by limiting the norm of input weights to neurons and that such a weight normalization (WN) method can improve the classification accuracy of BP-TAW. The findings presented can be used to further improve the performance of BP-TAW and open new ways for exploring possible learning mechanisms in biological neural networks without exact matching between forward and backward weights.

1. Introduction

For the past four decades, backpropagation (BP) has been the dominant learning method used in artificial neural networks [1]. However, BP is known to be implausible in the nervous system [2–4]. One of its major issues is known as the "weight transport problem" [5] which refers to the requirement that backward weights should be precisely equal to the forward weights so that accurate error signals are backpropagated to the early layers for efficient supervised learning. However, in the brain, axons transmit information unidirectionally, and to date, no explicit mechanism that guarantees a match between backward and forward weights is reported.

Despite differences in natural and artificial learning mechanisms, striking similarities between the activity of neurons in the brain and that of artificial ones trained by BP have been reported [6–10], and possibilities for the calculation of approximate gradient directions in the brain are suggested [11–14]. In particular, it has been shown that learning occurs even without exact weight transport [15,16] and when arbitrary weights that are distinct from forward ones backpropagate vectorized error signals to early layers [17], a method which we refer to as backpropagation through arbitrary weights (BP-TAW).

During the learning process using BP-TAW, the angle between the forward weight matrices and the transpose of backward weight matrices in each layer decreases (forward and backward weights become similar to each other) and this partial alignment leads to weight update directions that are partially aligned with the weight update directions calculated by BP, thereby providing effective teaching signals [17]. It has also been shown that by a learning algorithm known as direct feedback alignment (DFA), learning can occur even when errors are passed directly from the output layer to each hidden layer through direct arbitrary backward weights [18–22]. These sub-optimal calculations of gradient directions (compared to BP) can lead to a surprisingly good degree of learning accuracy, comparable to BP in shallow networks and simple tasks but with a drop in accuracy in deep convolutional networks and complex tasks [21,23,24].

* Corresponding author at: Sharif University of Technology, Azadi Ave, Tehran, 1458889694, Iran.
E-mail address: ghazizadeh@sharif.edu (A. Ghazizadeh).

https://doi.org/10.1016/j.neucom.2024.128587
Received 15 December 2023; Received in revised form 10 June 2024; Accepted 11 September 2024
Available online 23 September 2024

Notably, some special types of reinforcement learning methods called weight perturbation and node perturbation [25–27] also provide a solution to the weight transport problem. In these methods, a network is trained by a global scalar error (or reward) signal. Namely, the synaptic weights are randomly perturbed, and if the error decreases, an update in line with the direction of perturbation, and if the error increases, an update opposite to the direction of perturbation will be applied. Nevertheless, these methods are not scalable and converge more slowly than gradient-based learning methods in large-scale networks because of unguided random search in a high-dimensional weight space [28,29]. There is also biological evidence suggesting that the brain employs a gradient-based learning method (using vectorized error signals) to learn new tasks [30,31]. Hence, studying bio-inspired learning methods such as BP-TAW that approximate vectorized error signals of gradient descent is of great interest.

Given the potential biological relevance of BP-TAW and weight alignment (WA), it is valuable to investigate their underlying mechanisms and mathematical basis. In a sparsely or locally connected network of biological neurons, alignment between feedforward and feedback weights means that computational units (consisting of single neurons or groups of neurons) that are located in different areas of a hierarchy tend to form reciprocal connections. The existence of such connections has been reported in previous works [32–34], and understanding the mechanisms behind BP-TAW and WA opens new avenues for studying the existence and emergence of such patterns in the brain.

Although there are some investigations on the favorable conditions for WA and for the improvement of learning using BP-TAW [17,19,20,22,24,35–37], the underlying mathematical basis for the success of BP-TAW in training ANNs is not fully understood. For instance, previous works have explored some of the conditions that lead to WA in the special case of linear networks [17,20]. It has been shown that if all forward weights are initialized to zero and the input and desired output of a network are kept constant during iterations, each backward weight matrix becomes a scalar multiple of the Moore–Penrose pseudo-inverse of forward weight matrices [17], or a scalar multiple of the Moore–Penrose pseudo-inverse of the product of forward weight matrices in the case of DFA [20]. Under these circumstances, the update directions of BP-TAW are an approximation of the Gauss–Newton optimization method [17]. However, these proofs cannot explain the occurrence of WA for arbitrary initialization of weights (supplementary figure 13 of Lillicrap et al. [17]) and nonlinear networks.

There are some preliminary explanations for the occurrence of alignment. In particular, Lillicrap et al. [17] have provided insight into the mechanics of feedback alignment (FA) by freezing forward weights in different stages of the learning process of an ANN trained by BP-TAW, showing that information about the backward weight matrix of each layer (B_ℓ in Fig. 1) gradually accumulates in the earlier forward weight matrix ($W_{\ell-1}$ in Fig. 1) and then flows into the next forward weight (W_ℓ in Fig. 1) such that each forward weight matrix aligns with its corresponding backward weight matrix (W_ℓ and B_ℓ in Fig. 1). It has been also noted that under a particular condition where the input of a two-layer linear network is white noise and the network is trained to learn a linear function, the continuous growth of the norm of weight matrices results in alignment (supplementary note 12 of Lillicrap et al. [17]). We will show that although the original form of BP-TAW is accompanied by a growth of the norms of weight matrices, this growth can be detrimental to WA, and limiting the norms of weights can improve WA.

In this work, first, we explore the mathematical basis of WA and show that the occurrence of alignment is driven by statistical properties of neural activity such as cross-correlation and autocorrelation of error and output signals of neurons. Afterward, we will use the presented theoretical results in the context of a practical deep nonlinear ANN to explore various factors contributing to WA. We will show that the relative similarity of data points belonging to a single category compared to the ones belonging to different categories contributes to alignment by shaping cross-correlated neural activity and the arrangement of data points among mini-batches contributes to alignment by shaping autocorrelated neural activity.

2. Results

2.1. Explaining the occurrence of alignment

2.1.1. Notation

Consider a conventional d-layer ANN. We denote the matrices of forward weights, internal states of neurons, and output signals of neurons by $W_\ell \in \mathbb{R}^{n_\ell \times n_{\ell+1}}$, $Z_\ell \in \mathbb{R}^{n_b \times n_\ell}$, and $L_\ell = f(Z_\ell)$, respectively, where n_b is the batch size, n_ℓ is the number of neurons in layer ℓ, and $f(\cdot)$ is an element-wise activation function (the following analysis still holds if the batch size is variable among mini-batches; however, for simplicity, we assume that it is a constant number for all of them). For $0 < \ell \le d$, internal states of neurons in layer ℓ are calculated according to $Z_\ell = L_{\ell-1}W_{\ell-1} + \mathbf{b}_\ell$ where $\mathbf{b}_\ell$ is the bias vector of layer ℓ and the addition of a matrix with a row vector is defined as adding the vector to each row of the matrix. We denote the input, output, and desired output matrices of the network by $X = L_0 \in \mathbb{R}^{n_b \times n_0}$, $Y = L_d \in \mathbb{R}^{n_b \times n_d}$, and $Y^* \in \mathbb{R}^{n_b \times n_d}$, respectively.

2.1.2. Deriving alignment terms

In BP-TAW [17], the error is backpropagated through constant arbitrary matrices (different from forward weights) denoted by $B_\ell \in \mathbb{R}^{n_{\ell+1} \times n_\ell}$, and the weight update directions are calculated at each iteration $k \ge 0$ according to

$$\Delta W_{\ell,FA}[k] = \eta L_\ell[k]^T \delta_{\ell+1,FA}[k],\ 0 \le \ell < d, \tag{1}$$

where η is the learning rate and error signals of neurons are

$$\delta_{\ell,FA}[k] = \begin{cases} \delta_{\ell+1,FA}[k]B_\ell \odot f'\big(Z_\ell[k]\big) & 0 < \ell < d \\ -\eta \left.\frac{\partial \mathcal{L}}{\partial Z_d}\right|_k & \ell = d, \end{cases} \tag{2}$$

where $\mathcal{L}(Y, Y^*)$ is the loss function and $\odot$ denotes the element-wise matrix multiplication (in the order of operations, it has less priority than matrix multiplication).

To investigate WA, we investigate the alignment of update directions ($\Delta W_{\ell,FA} \measuredangle B_\ell^T$) because during the learning process, the update directions accumulate in W_ℓ and their resultant determines the final direction of W_ℓ (if an update direction is aligned with B_ℓ^T, it injects a component in line with B_ℓ^T into $W_{\ell,FA}$). To demonstrate why the update rule of FA aligns with B_ℓ^T, we expand $\Delta W_{\ell,FA}[k]$ by taking successive steps backward along the iterations and substituting every $W_\ell[k-o]$ for $0 \le o < k$. Assuming the update steps to be small, by applying the first-order Taylor approximation we have

$$\begin{aligned} &\Delta W_{\ell,FA}[k] = \eta L_\ell[k]^T \delta_{\ell+1,FA}[k] \\ &= \eta f\big(W_{\ell-1}[k]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\delta_{\ell+1,FA}[k] \\ &= \eta \Big(\{W_{\ell-1}[k-1]^T + \eta\delta_{\ell,FA}[k-1]^T L_{\ell-1}[k-1]\} L_{\ell-1}[k]^T \\ &\quad + \mathbf{b}_\ell[k]^T \Big)\, \delta_{\ell+1,FA}[k] \\ &\approx \eta \Big\{ f\big(W_{\ell-1}[k-1]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big) + \\ &\quad f'\big(W_{\ell-1}[k-1]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big) \odot \eta\delta_{\ell,FA}[k-1]^T L_{\ell-1}[k-1] \\ &\quad \times L_{\ell-1}[k]^T \Big\}\, \delta_{\ell+1,FA}[k] \\ &\approx T^1_{\ell,aln}[k] + T^2_{\ell,aln}[k] + \cdots + T^k_{\ell,aln}[k] + \eta f\big(\zeta^k_\ell[k]\big)^T \delta_{\ell+1,FA}[k], \end{aligned} \tag{3}$$

where $\zeta^o_\ell[k] = L_{\ell-1}[k]W_{\ell-1}[k-o] + \mathbf{b}_\ell[k]$ and for $1 \le o \le k$ and $0 < \ell < d$ we define

$$T^o_{\ell,aln}[k] = \eta\big\{f'\big(\zeta^o_\ell[k]\big)^T \odot \eta\delta_{\ell,FA}[k-o]^T L_{\ell-1}[k-o] L_{\ell-1}[k]^T\big\}\delta_{\ell+1,FA}[k] = \eta\big\{f'\big(\zeta^o_\ell[k]\big)^T \odot \eta\{f'\big(Z_\ell[k-o]\big)^T \odot B_\ell^T \delta^T_{\ell+1,FA}[k-o]\} L_{\ell-1}[k-o] L_{\ell-1}[k]^T\big\}\delta_{\ell+1,FA}[k] \tag{4}$$

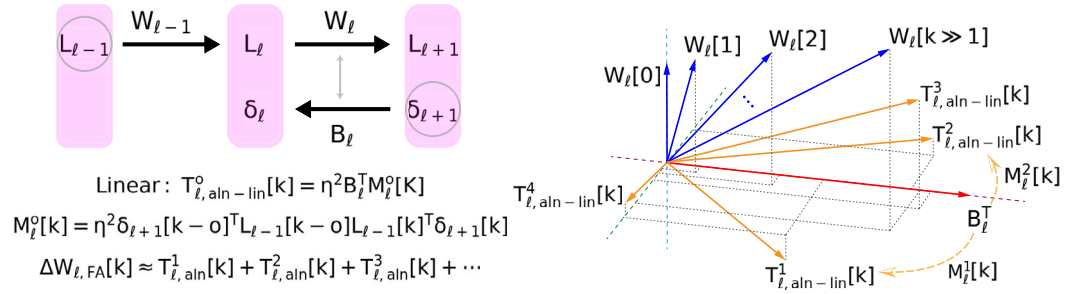


Fig. 1. The underlying mechanism of weight alignment in BP-TAW. Expansion of $\Delta W_{\ell,FA}[k]$ reveals alignment terms $T^o_{\ell,aln}[k]$ (Eqs. (3), (4), and (5)). In linear alignment terms, $M^o_\ell[k]$ acts as a transformation matrix on B^T_ℓ and if it partially preserves the direction of B^T_ℓ after the matrix multiplication, it propels $W_\ell[k]$ towards B^T_ℓ. $L_\ell[k]$ and $\delta_\ell[k]$ denote the matrices of output and error signals of neurons, respectively, where each row of them corresponds to a data point of the mini-batch at the iteration k and each column of them corresponds to a neuron in layer ℓ. Due to the structure of $M^o_\ell[k]$, alignment terms can robustly propel forward weight matrices (W_ℓ) towards the transpose of fixed random backward weight matrices (B^T_ℓ) under a variety of conditions depending on neural activity. Note that this is a simplified diagram of the underlying mechanism of WA. In practice, at each iteration k, there are k alignment terms, and depending on the neural activity, each of them may or may not align with B^T_ℓ. Moreover, in nonlinear ANNs, nonlinearity affects the structure of alignment terms to some extent (Eq. (4), see Supplementary Note 3). An order ($1 \le o \le k$) is assigned to alignment terms since in each of them the activity of neurons at iterations of k and $k-o$ (lag of o) are multiplied together.

as the *alignment term of order* o corresponding to layer ℓ (see Supplementary Note 2 for higher-order Taylor approximation and Supplementary Note 3 for index notation of alignment terms).

2.1.3. Analyzing the linear alignment terms

The alignment terms can provide alignment owing to their structure (Fig. 1). Consider the linear case of the alignment terms (ignoring the element-wise matrix multiplications in Eq. (4)) where they reduce to

$$T^o_{\ell,aln-lin}[k] = \eta^2 B^T_\ell \delta_{\ell+1}[k-o]^T L_{\ell-1}[k-o] L_{\ell-1}[k]^T \delta_{\ell+1}[k]. \tag{5}$$

In this case, the occurrence of alignment depends on the transformation matrix

$$M^o_\ell[k] = \eta^2 \delta_{\ell+1}[k-o]^T L_{\ell-1}[k-o] L_{\ell-1}[k]^T \delta_{\ell+1}[k], \tag{6}$$

which is applied to B^T_ℓ and if $M^o_\ell[k]$ partially preserves the direction of B^T_ℓ after matrix multiplication, $T^o_{\ell,aln}[k]$ partially aligns with B^T_ℓ. In general, $M^o_\ell[k]$ can be decomposed into its symmetric and skew-symmetric parts ($M^o_\ell[k] = M^o_{\ell,skew}[k] + M^o_{\ell,sym}[k]$). Any skew-symmetric transformation matrix changes the direction of the transformed matrix by 90° ($B^T_\ell \measuredangle B^T_\ell M^o_{\ell,skew}[k] = 90°$, the proof is provided in Supplementary Note 4 for), where the angle between two matrices is calculated using arccosine of their normalized Frobenius product (mathematical definition is provided in Section 4.2). Hence, the occurrence of alignment depends on $M^o_{\ell,sym}[k]$.

It is convenient to analyze linear alignment terms from a statistical perspective and under reasonable simplifying assumptions as analyzing alignment terms, in general, can be challenging, although not impossible. Therefore, we investigated under what conditions the occurrence of alignment is statistically expected and under what statistical assumptions the analysis of alignment will become more tractable. In an ANN, according to random initialization of weights and biases, error and output signals of neurons and also the evolution of weights and biases through learning are stochastic processes and statistical properties of them affect statistical properties of alignment terms.

There are some reasonable statistical assumptions under which the analysis of alignment terms becomes simpler. One statistical property that affects the analysis of alignment terms is the presence or absence of any correlation between different network signals and weights. In general, the absence of correlation between different network signals and weights can simplify the analysis of alignment terms. For example, in a deep network where there are many neurons in each layer, knowing the values of one or two backward weights to layer ℓ does not provide us with much information about the error signal of a neuron in layer $\ell+1$ or the output signal of a neuron in layer $\ell-1$. According to this, we introduce the two following assumptions that simplify the analysis of alignment terms (we will investigate the validity of the following assumptions in the context of practical ANNs in the next sections).

Assumption 1. If one element is arbitrarily selected from each of the matrices $\delta_{\ell+1,FA}[k-o]$, $L_{\ell-1}[k-o]$, $L_{\ell-1}[k]$, and $\delta_{\ell+1,FA}[k]$ and the four selected elements are multiplied together, their product is uncorrelated with the product of any two arbitrary elements of B_ℓ as well as with the square of any arbitrary element of B_ℓ.

Assumption 2. If one element is arbitrarily selected from each of the matrices $f(\zeta^k_\ell[k])$, and $\delta_{\ell+1,FA}[k]$ and the two selected elements are multiplied together, their product is uncorrelated with any arbitrary element of (B_ℓ).

In addition to these assumptions, some specific network configurations make the analysis of alignment terms simpler. For example, weight initialization from distributions with a mean of zero is a common practice in ANNs. Considering such a condition makes the analysis of alignment terms simpler while explaining alignment in many previous works where such initialization is used [17,19,20,24,35–37]. Here, we introduce the following theorem based on these assumptions and conditions.

Theorem 1. *Under Assumptions 1 and 2, in a linear network where the elements of* B_ℓ *are independently initialized from a distribution with a mean of zero, the alignment between* $\Delta W_\ell[k]$ $(k > 0)$ *and* B^T_ℓ *is expected in the sense that*

$$\mathbb{E}(\langle \Delta W_\ell[k], B^T_\ell \rangle_F) > 0 \tag{7}$$

if and only if

$$\mathbb{E}(\sum_{o=1}^{k} \sum_i \lambda_i^{o,\ell}[k]) > 0, \tag{8}$$

where $\lambda_i^{o,\ell}$ *denotes the ith eigenvalue of* $M^o_{\ell,sym}$ *and* $< .,. >_F$ *denotes the Frobenius inner product of two matrices.*

Proof. Two random variables being uncorrelated (having covariance and Pearson's correlation coefficient of zero) means that the expected value of their product equals the product of the separate expected values of them. Accordingly, Assumptions 1 and 2 follow

$$\mathbb{E}(B_\ell B^T_\ell M^o_\ell) = \mathbb{E}(B_\ell B^T_\ell)\mathbb{E}(M^o_\ell) \tag{9}$$

and

$$\mathbb{E}(\zeta^k_\ell[k]^T \delta_{\ell+1,FA}[k] B_\ell) = \mathbb{E}(\zeta^k_\ell[k]^T \delta_{\ell+1,FA}[k])\mathbb{E}(B_\ell), \tag{10}$$

respectively.

According to the condition for the distribution of the elements of B_ℓ, we can write

$$\mathbb{E}(\langle \eta\zeta_\ell^k[k]^T \delta_{\ell+1,FA}[k], B_\ell^T\rangle_F) = \mathbb{E}(tr(\eta\zeta_\ell^k[k]^T \delta_{\ell+1,FA}[k] B_\ell)) = tr(\mathbb{E}(\eta\zeta_\ell^k[k]^T \delta_{\ell+1,FA}[k])\mathbb{E}(B_\ell)) = 0. \tag{11}$$

Therefore,

$$\mathbb{E}(\langle \Delta W_\ell[k], B_\ell^T\rangle_F) = \mathbb{E}(\langle \sum_{o=1}^{k} T^o_{\ell,aln}[k] + \eta\zeta_\ell^k[k]^T \delta_{\ell+1,FA}[k], B_\ell^T\rangle_F) = \mathbb{E}(\langle \sum_{o=1}^{k} T^o_{\ell,aln}[k], B_\ell^T\rangle_F) = \sum_{o=1}^{k} tr(\mathbb{E}(B_\ell B_\ell^T)\mathbb{E}(M_\ell^o)) = \sum_{o=1}^{k} n_\ell \sigma_B^2 \mathbb{E}(tr(M_\ell^o)) = \sum_{o=1}^{k} n_\ell \sigma_B^2 \mathbb{E}(tr(M^o_{\ell,sym})) = \sum_{o=1}^{k} n_\ell \sigma_B^2 \mathbb{E}(\sum_i \lambda_i^{o,\ell}) = n_\ell \sigma_B^2 \mathbb{E}(\sum_{o=1}^{k}\sum_i \lambda_i^{o,\ell}[k]), \tag{12}$$

where σ_B^2 is the variance of the distribution of the elements of B_ℓ. □

Theorem 1 demonstrates the contribution of the eigenvalues of the symmetric part of the transformation matrices $M^o_{\ell,sym}$ to alignment. For example, in an extreme case, if $M^o_{\ell,sym}$ is positive semidefinite, when it transforms an arbitrary vector, it scales each component of the vector with a nonnegative scalar which is the corresponding eigenvalue. In other words, it keeps each component in its previous direction and does not flip it by 180°, which is desirable for alignment. From a statistical point of view, $M^o_{\ell,sym}$ being semidefinite is not necessary and given the above assumptions, on average, alignment is expected if the mean of the eigenvalues of $M^o_{\ell,sym}$ is positive. From a deterministic point of view, there can be more complex conditions where the mean of the eigenvalues of $M^o_{\ell,sym}$ is positive, but rows of B_ℓ^T lie near some eigenvectors whose corresponding eigenvalues are negative. In such a case, alignment does not occur despite having a positive mean of the eigenvalues. However, under Assumption 1, such a condition is not statistically expected.

2.1.4. Analyzing the nonlinear alignment terms

The analysis of the linear alignment terms can be extended to the non-linear case under certain conditions. Nonlinearity appears as two element-wise matrix multiplications that impact the structure of alignment terms Eq. (4). Under the two assumptions that we introduce below, we can ignore nonlinearity and analyze the linear version of alignment terms (where all matrices resulting from $f'(\cdot)$ are replaced by all-ones matrices) and be sure that if the alignment is expected in the linear case, the alignment is also expected in the nonlinear case.

Assumption 3. If one element is arbitrarily selected from each of the matrices $f'(\zeta_\ell^o[k])$ and $f'(Z_\ell[k-o])$ and the two selected elements are multiplied together, the expected value of their product is positive.

Assumption 4. If one element is arbitrarily selected from each of the matrices $f'(\zeta_\ell^o[k])$ and $f'(Z_\ell[k-o])$ and the two selected elements are multiplied together, and also if one element is arbitrarily selected from each of the matrices B_ℓ, B_ℓ^T, $\delta_{\ell+1,FA}[k-o]$, $L_{\ell-1}[k-o]$, $L_{\ell-1}[k]$, and $\delta_{\ell+1,FA}[k]$ and the six selected elements are multiplied together, the two resulting products are uncorrelated with each other.

Assumption 3 implies that the activity of neurons should not be saturated. Using increasing activation functions, such as ReLU, is a common practice in ANNs. It causes the elements of the matrices resulting from $f'(\cdot)$ to be nonnegative, which alleviates the impact of nonlinearity and also provides a good context for Assumption 3 to hold. Assumption 4 implies the absence of correlation between signals resulting from nonlinearity ($f'(\zeta_\ell^o[k])$ and $f'(Z_\ell[k-o])$) and the other signals of the network (B_ℓ, B_ℓ^T, $\delta_{\ell+1,FA}[k-o]$, $L_{\ell-1}[k-o]$, $L_{\ell-1}[k]$, and $\delta_{\ell+1,FA}[k]$). Based on the above assumptions we introduce the following theorem (we will assess the validity of these assumptions in the context of practical ANNs in the next sections).

Theorem 2. *Under Assumptions 3 and 4, in a feedforward fully connected ANN where*

1. *The biases of all neurons within each layer are initialized from the same distribution*
2. *The input weights of all neurons within each layer are initialized from the same distribution*
3. *Within each mini-batch, the order of data points is uniformly random such that the probability of a specific data point of the mini-batch occurring at any position in the mini-batch is the same as any other position*

if the alignment of the linear version of a nonlinear alignment term is statistically expected, the alignment of the nonlinear alignment term is also statistically expected.

Proof. Refer to Supplementary Note 3. □

2.1.5. Contribution of the autocorrelation and cross-correlation of neural activity to alignment

In addition to the eigenvalues of the transformation matrices, we can extend the result of Theorem 1 to the statistical properties of neural activity. In particular, the autocorrelation of error ($\delta_{\ell+1}$) and output signals ($L_{\ell-1}$) of neurons and also cross-correlation between them play an important role in WA. Considering the above simplifying assumptions, the role of autocorrelation and cross-correlation can be seen by breaking M_ℓ^o into its constituent terms as follows

$$\mathbb{E}(\sum_{o=1}^{k}\sum_i \lambda_i^{o,\ell}[k]) = \sum_{o=1}^{k} n_\ell \sigma_B^2 \mathbb{E}(tr(M_\ell^o)) = \sum_{o=1}^{k} \eta^2 n_\ell \sigma_B^2 \mathbb{E}(tr(\delta_{\ell+1}[k-o]^T L_{\ell-1}[k-o] L_{\ell-1}[k]^T \delta_{\ell+1}[k])) = \sum_{o=1}^{k} \eta^2 n_\ell \sigma_B^2 \mathbb{E}(tr(\delta_{\ell+1}[k]\delta_{\ell+1}[k-o]^T L_{\ell-1}[k-o] L_{\ell-1}[k]^T)) = \sum_{o=1}^{k} \eta^2 n_\ell \sigma_B^2 \mathbb{E}\Big(tr(S^o_{\delta_{\ell+1}}[k]^T S^o_{L_{\ell-1}}[k])\Big), \tag{13}$$

where we define $S^o_{L_{\ell-1}}[k] = L_{\ell-1}[k-o]L_{\ell-1}[k]^T$ as the *output similarity matrix* of layer $\ell-1$ and $S^o_{\delta_{\ell+1}}[k] = \delta_{\ell+1}[k-o]\delta_{\ell+1}[k]^T$ as the *error similarity matrix* of layer $\ell+1$. The autocorrelation of output and error signals of neurons contributes to shaping the statistical properties of these two matrices, and the cross-correlation between them contributes to shaping the statistical properties of their product. We define the autocorrelation function of a discrete-time stochastic signal as a function of k and o to be the expected value of the product of the signal samples at k and $k-o$. We define the cross-correlation function of two discrete-time stochastic signals as a function of k and o to be the expected value of the product of the first signal at k and the second signal at $k-o$ (refer to Supplementary Note 3 for more detail).

To show the contribution of cross-correlation and autocorrelation of neurons to alignment, we performed a simulation using an open-loop two-layer ANN with ReLU nonlinearity where we manually set the output signals of input neurons (L_0) and error signals of output neurons (δ_2) and controlled their autocorrelation and cross-correlation (Fig. 2A). For example, in an extreme hypothetical condition where we initially drew elements of L_0 and δ_2 independently from $\mathcal{N}(0,1)$ and left them constant through iterations. In this case, the error and output signals of neurons are autocorrelated, the expected values of both $S^o_{\delta_2}[k]$ and $S^o_{L_0}[k]$ are scalar multiples of the identity matrix, the transformation matrix $M_1^o = \eta^2\delta_2^T L_0 L_0^T \delta_2$ is a symmetric positive semidefinite matrix (Fig. 2B first row), and alignment happened as predicted (Fig. 2C blue trace).

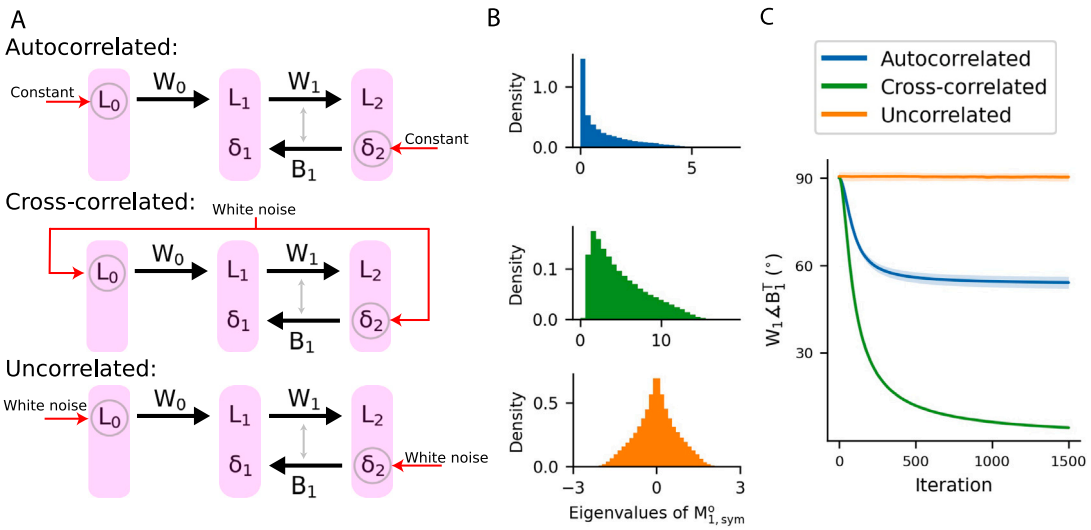


Fig. 2. Weight alignment can occur in the absence of meaningful input and error depending on the statistical properties of neural activity. (A) In an open-loop nonlinear two-layer ANN, hypothetical output signals are imposed on the neurons in the input layer (L_0) and hypothetical error signals are imposed on the neurons in the output layer (δ_2). In the top row, elements of δ_2 and L_0 are independently generated from $\mathcal{N}(0,1)$ at the beginning and left constant across all iterations. In this case, inputs and errors are autocorrelated. In the middle row, elements of $\delta_2 = L_0$ are independently generated from $\mathcal{N}(0,1)$ at each iteration. In this case, elements of δ_2 and L_0 are not autocorrelated, but they are cross-correlated. In the bottom row, elements of δ_2 and L_0 are independently generated from $\mathcal{N}(0,1)$ at each iteration. In this case, input and error signals are neither autocorrelated nor cross-correlated. **(B)** Histograms of the eigenvalues of random samples of $M^o_{1,sym}$ corresponding to the conditions of panel A. In the top and middle row scenarios, the mean of the eigenvalues is positive, but, in the bottom row, the mean of them is zero. **(C)** The angle between forward and backward weights $W_1 \measuredangle B_1^T$ in the scenarios of panel A. Each trace is the average over 10 runs and shaded areas are one s.d. around the mean.

In another case, we re-initialized elements of L_0 independently from $\mathcal{N}(0,1)$ at each iteration and let $\delta_2[k] = L_0[k]$. In this case, output signals of input neurons and error signals of output neurons were white noise and were not autocorrelated, but they were fully cross-correlated. Here, although the expected values of elements of $S^o_{\delta_2}[k]$ and $S^o_{L_0}[k]$ at any given lag $o \neq 0$ were zero, they were positively cross-correlated, and alignment happened (Fig. 2C, green trace). In contrast, when we independently re-initialized all elements of L_0 and δ_2 from $\mathcal{N}(0,1)$ at each iteration, error signals of output neurons and output signals of input neurons were neither autocorrelated nor cross-correlated and alignment did not happen (Fig. 2C, orange trace). The occurrence of alignment in these scenarios can be predicted from the distribution of the eigenvalues of $M^o_{1,sym}$ as in the last scenario with $\mathbb{E}(\lambda_i^{o,1}) = 0$ alignment did not happen (Fig. 2B).

Although the network was nonlinear in these three scenarios, we considered the linearized version of them and ignored the nonlinearity for the analysis. Given the actual behavior of the nonlinear network, there was no game-changing correlation between matrices resulting from $f'(\cdot)$ and other signals of the network (L_0 and δ_2) that would invalidate the conclusion drawn from the linear analysis. According to the structure of the network used, the conditions of Theorem 2 are true. Assumption 3 also holds since the activation function is increasing and also we did not observe any situation where all the neurons are saturated. We performed a statistical hypothesis test to validate Assumption 4. We obtained the sampling distribution of Pearson's correlation coefficient between the two corresponding terms of Assumption 4 for 10 different combinations of k and o by resampling 10^5 pairs of the two corresponding terms of Assumption 4 for 10^3 repetitions. The maximum calculated correlation coefficient was less than 0.01 (no strong correlation) and the mean of the sampling distribution did not differ significantly from zero (p-value > 0.1, one-sample t-test).

2.1.6. Direct feedback alignment

In DFA [18], instead of backpropagation of error signals step by step from each layer to its previous layer, error signals are backpropagated directly from the output layer to each hidden layer through direct fixed random weights, for instance, $F_\ell \in \mathbb{R}^{n_d \times n_\ell}$. With this method, it has been reported that the product of forward weights $(W_\ell W_{\ell+1} \cdots W_{d-1})$ aligns with F_ℓ^T [38]. In this work, we focus on FA, but it can be shown that a similar technique with Taylor series expansion can be used to explain DFA (see Supplementary Note 5).

2.2. Investigating BP-TAW and weight alignment in practical ANNs

In this section, we investigate how the analysis presented above relates to the BP-TAW in practical ANNs and enables us to understand factors contributing to WA in them. To provide a concrete example, we examined the dynamics of alignment terms in the learning process of a specific five-layer nonlinear fully connected ANN (Fig. 3A) designed for handwritten digits classification on the MNIST dataset (refer to the Methods section for the details of the network).

To perform this analysis, we make use of the results of Theorems 1 and 2. To ensure applicability, we validated the assumptions and conditions of these theorems. According to the architecture of the network, the conditions of the theorems hold. We validated Assumptions 2 and 3 by closely examining the behavior of the network and its associated attributes (Supplementary Note 7). We performed statistical hypothesis testing to validate Assumptions 1 and 4 and found no evidence to reject these assumptions (Supplementary Note 7). These two latter assumptions allow us to simplify the analysis of alignment terms by ignoring nonlinearity and also analyzing M^o_ℓ independently of B^T_ℓ. Additionally, We also ensure that the first-order Taylor approximation used for the extraction of alignment terms is a fair approximation in the nonlinear network under investigation (Supplementary Note 7).

2.2.1. Factors affecting the dynamics of alignment terms and alignment

2.2.1.1 Autocorrelation of error and output signals of neurons contributes to alignment One aspect of neural activity that contributes to alignment and affects the behavior of alignment terms is the autocorrelation of error and output signals of neurons, and one factor that shapes autocorrelated neural activity is the arrangement and distribution of data points across mini-batches. To demonstrate this, in the example network under investigation, we examined the behavior of alignment terms under two data point arrangement schemes: with and without data shuffling. With data shuffling, after each epoch, the data

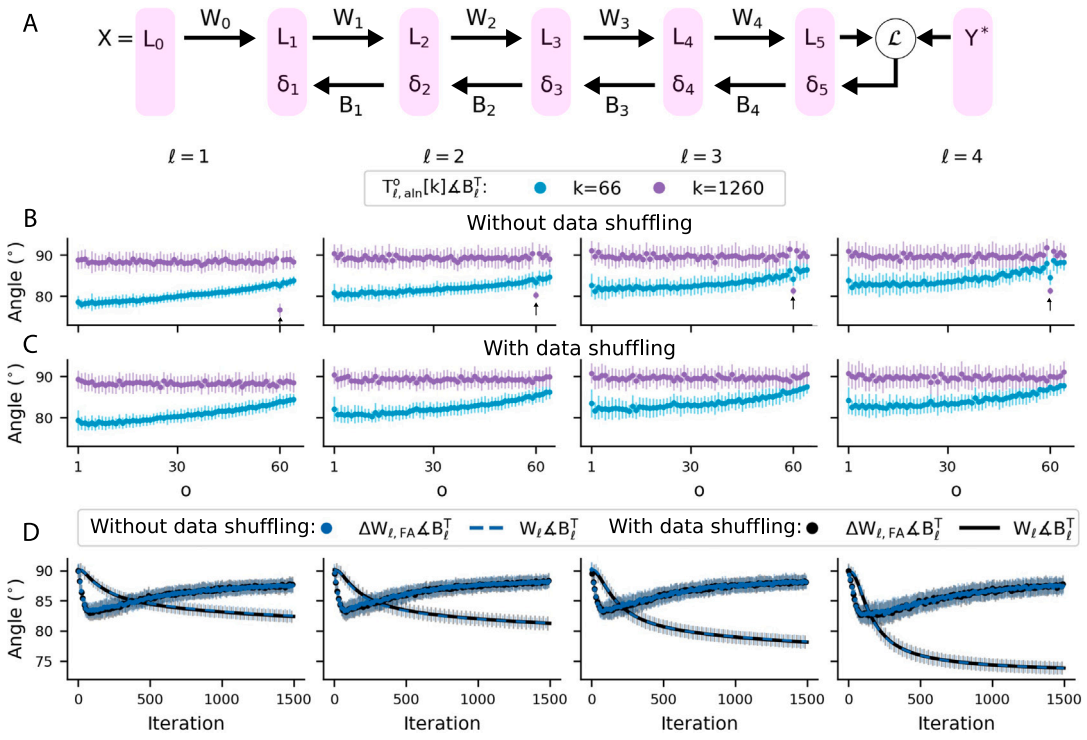


Fig. 3. Repetition of the same data points in each epoch contributes to alignment by shaping autocorrelated neural activity. **(A)** Layout of the network under investigation. **(B)** The training dataset is divided into 60 mini-batches which are the same in all epochs (no data shuffling). As the learning process proceeds, the alignment terms whose orders are not a multiple of 60 lose their initial amount of alignment, while the one whose order is 60 becomes more aligned. This difference is due to the repetition of the same mini-batches every 60 iterations which makes neural activity autocorrelated at the lag of 60 and its multiples. **(C)** The training dataset is divided into 60 mini-batches and shuffled at the beginning of each epoch (the arrangement of data points across mini-batches is changed). Unlike the case without data shuffling, the alignment term of order 60 does not behave differently since the same arrangement of mini-batches is not repeated in every epoch. **(D)** Data shuffling changes the behavior of alignment terms but not the total alignment since the autocorrelated activity of neurons, which without data shuffling is concentrated in the lags that are a multiple of 60, with data shuffling becomes distributed among other lags. (B-D) Each dot or trace is the average over 30 runs and error bars are one s.d. around the mean.

points are shuffled across mini-batches, but without data shuffling, the mini-batches do not change across epochs.

To show the potential contribution of autocorrelation of neural activity to alignment, in one experimental setup of the example network under investigation, we divided the dataset into 60 mini-batches and did not perform data shuffling. Therefore, the same mini-batches were repeated every 60 iterations. As a result, neural activity was autocorrelated at lags that are a multiple of 60, making the alignment terms whose orders are a multiple of 60 behave differently from other alignment terms. In this experimental setup, we observed that in the initial phase of the learning process, all orders of alignment terms were considerably aligned, but those whose orders were a multiple of 60 aligned slightly more than their adjacent orders (Fig. 3B, Fig. S1A). With the continuation of the learning process, the amount of alignment of the alignment terms whose orders were not a multiple of 60 decreased, while the terms whose orders were a multiple of 60 became more aligned (Fig. 3B, Fig. S1A).

Referring back to Eq. (5), an appropriate condition for the occurrence of alignment which makes M_ℓ^o a positive semidefinite matrix is that $L_{\ell-1}[k-o]$ and $\delta_{\ell+1}[k-o]$ are equal to $L_{\ell-1}[k]$ and $\delta_{\ell+1}[k]$, respectively. Without data shuffling and by having a small learning rate, this condition is approximately satisfied for lags that are a multiple of 60. Moreover, as the network learns, its response to data points (the output and error signals of neurons produced by each data point) becomes more stable. Hence, in comparison to the initial phase, in the late phase of the learning process, $L_{\ell-1}[k-60]$ and $\delta_{\ell+1}[k-60]$ become more similar to $L_{\ell-1}[k]$ and $\delta_{\ell+1}[k]$, respectively, making the alignment term of order 60 more aligned (Fig. 3B, Fig. S1A).

With data shuffling, this considerable amount of autocorrelation does not exist in the activity of neurons at lags that are a multiple of 60, and thus alignment terms whose orders are a multiple of 60 behave similarly to the other orders of alignment terms and lose their initial amount of alignment with the continuation of the learning process (Fig. 3C, Fig. S1B). However, the amount of alignment between ΔW_ℓ and B_ℓ^T does not change with data shuffling (Fig. 3D). Assuming the update steps to be relatively small, in terms of statistical properties of neural activity, shuffling is similar to substituting the rows of error and output matrices across different lags with each other. As a result, the autocorrelated activity of neurons, which without data shuffling is concentrated in the lag of $o = 60$, with data shuffling becomes distributed among the lags of $o = 60$ to $o = 119$ because a specific data point that appears in the mini-batch of the kth iteration, will inevitably be repeated in one of the mini-batches of $(k+60)^{th}$ to $(k+119)^{th}$ iteration. The alignment of W_ℓ with B_ℓ^T is influenced by the summation of ΔW_ℓ across iterations and the alignment of ΔW_ℓ itself is influenced by the resultant of all orders of alignment terms. Data shuffling changes the behavior of individual alignment terms but preserves their collective behavior and has no considerable effect on the amount of alignment (see Supplementary Note 3).

The distinct behavior of alignment terms whose orders are a multiple of 60 is specific to the configuration of this specific network under investigation and can change in other situations based on the number of mini-batches and the arrangement of data in them. However, in general, the potential contribution of the autocorrelation of neural activity to alignment is not disregardable and is not limited to the example network under investigation.

It is evident in Fig. 3 (and Fig. S1) that alignment terms whose orders are not a multiple of 60 are aligned in the initial phase of the learning process but gradually lose their initial amount of alignment. We will investigate the reasons behind this behavior in the next section.

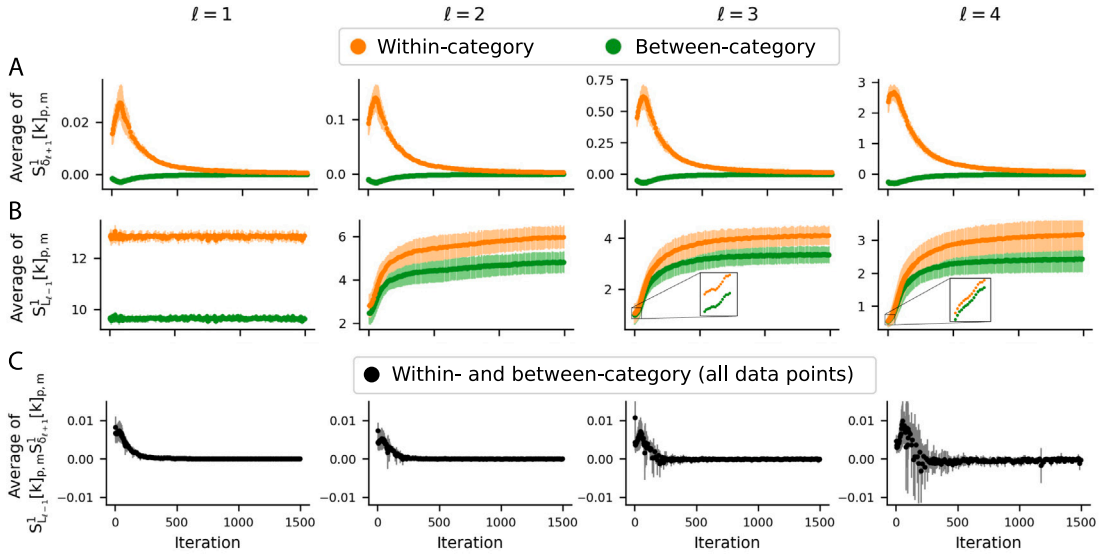


Fig. 4. The relative similarity of data points belonging to a single category compared to data points belonging to different categories contributes to alignment by shaping cross-correlated neural activity. (A) As a measure of similarity, $S^o_{\delta_{\ell+1}}[k]_{p,m}$ is the dot product between the error signals of neurons in layer $\ell+1$ produced by the pth data point of the $(k-o)^{th}$ mini-batch and the mth data point of the kth mini-batch. Within-category corresponds to the condition that both of these two data points belong to a single category and between-category corresponds to the condition that they belong to different categories. **(B)** The same as A but for output signals of neurons. At each iteration, on average, within-category data points and their representations across layers are more similar to each other than data points belonging to different categories. **(C)** In the early phase of the learning process, the cross-correlation between $S^1_{L_{\ell-1}}[k]_{p,m}$ and $S^1_{\delta_{\ell+1}}[k]_{p,m}$ contributes to the positivity of the average of $S^1_{L_{\ell-1}}[k]_{p,m}S^1_{\delta_{\ell+1}}[k]_{p,m}$ and makes the alignment to be expected Eq. (14). As the learning process proceeds, the network learns to classify, and it makes the majority of error signals vanish, especially in response to the data points that are very similar to the other data points of their category (see Supplementary Fig. S2), causing cross-correlation between $S^1_{L_{\ell-1}}[k]_{p,m}$ and $S^1_{\delta_{\ell+1}}[k]_{p,m}$ to become weaker and the average of $S^1_{L_{\ell-1}}[k]_{p,m}S^1_{\delta_{\ell+1}}[k]_{p,m}$ to decrease. (A-C) The averages are calculated over p and m. Each dot is the average over 10 runs and error bars are one s.d. around the mean.

2.2.1.2. Cross-correlation between error and output signals of neurons contributes to alignment It is normally the case that data points belonging to a single category are more similar to each other than the ones belonging to different categories. This property contributes to the alignment of W_ℓ with B^T_ℓ by shaping cross-correlated neural activity between $\delta_{\ell+1}$ and $L_{\ell-1}$. This can be seen in the elements of similarity matrices ($S^o_{\delta_{\ell+1}}$ and $S^o_{L_{\ell-1}}$). Referring back to Eq. (13) and considering the mentioned simplifying (Assumptions 1, 2, 4, 3), alignment is expected if

$$0 < \mathbb{E}(\langle T^o_\ell[k], B^T_\ell\rangle_F) \approx \eta^2 n_\ell \sigma^2 \mathbb{E}\Big(tr(S^o_{\delta_{\ell+1}}[k]^T S^o_{L_{\ell-1}}[k])\Big)$$
$$= \eta^2 n_\ell \sigma^2 \mathbb{E}\Big(\sum_{p,m} S^o_{\delta_{\ell+1}}[k]_{p,m} S^o_{L_{\ell-1}}[k]_{p,m}\Big), \tag{14}$$

where $S^o_{L_{\ell-1}}[k]_{p,m}$ denotes the element in the pth row and the mth column of $S^o_{L_{\ell-1}}[k]$. Namely, as a measure of similarity, $S^o_{L_{\ell-1}}[k]_{p,m}$ is the dot product between the output signals of neurons in layer $\ell-1$ produced by the pth data point of the $(k-o)^{th}$ mini-batch and their output signals produced by the mth data point of the kth mini-batch. $S^o_{\delta_{\ell+1}}[k]_{p,m}$ denotes a similar dot product but for the error signals of neurons in layer $\ell+1$.

In the summation of Eq. (14), we define $S^o_{\delta_{\ell+1}}[k]_{p,m}S^o_{L_{\ell-1}}[k]_{p,m}$ as the *similarity term*. Based on the categories of the pth data point of the $(k-o)^{th}$ mini-batch and the mth data point of the kth mini-batch, the similarity terms have different behaviors. If these two mentioned data points both belong to the same category, we regard their corresponding similarity term as a *within-category similarity term*, and if they belong to two different categories, we regard their corresponding similarity term as a *between-category similarity term*.

Since the activation function used in this network is nonnegative, the output signals of neurons are always nonnegative, and consequently, $S^o_{L_{\ell-1}}[k]_{p,m}$ is nonnegative. In within-category similarity terms, since data points have the same true label, their error signals are mostly similar to each other and $S^o_{\delta_{\ell+1}}[k]_{p,m}$ has a positive mean (Fig. 4A, Fig. S2A), which is constructive for alignment (referring to Eq. (14) and given that $S^o_{L_{\ell-1}}[k]_{p,m}$ is nonnegative). In contrast, in between-category similarity terms, since data points have different true labels, their error signals are dissimilar to each other and $S^o_{\delta_{\ell+1}}[k]_{p,m}$ has a negative mean (Fig. 4A, Fig. S2A), which is destructive for alignment (referring to Eq. (14) and given that $S^o_{L_{\ell-1}}[k]_{p,m}$ is nonnegative).

However, there is an advantageous cross-correlation between $S^o_{\delta_{\ell+1}}[k]_{p,m}$ and $S^o_{L_{\ell-1}}[k]_{p,m}$ which is that in within-category similarity terms, $S^o_{L_{\ell-1}}[k]_{p,m}$ has a higher mean compared to between-category similarity terms (Fig. 4B, Fig. S2B), strengthening the constructive effect of $S^o_{\delta_{\ell+1}}[k]_{p,m}$ in within-category similarity terms compared to between-category ones. This cross-correlation originates from the mentioned property of the dataset which directly affects the input layer of the network and makes $S^o_{L_0}[k]_{p,m}$ of within-category similarity terms have a relatively high mean compared to that of between-category similarity terms (Fig. 4B, Fig. S2B). In the initial phase, although the network does not discriminate between categories, this feature is still preserved in the similarity terms of subsequent layers (Fig. 4B, Fig. S2B). In the summation of Eq. (14), the number of within-category similarity terms is less than the number of between-category ones. For example, if there are 10 different categories and an equal number of data points in each of them, 10% of similarity terms are within-category, and 90% of them are between-category. Nevertheless, in the initial phase of the learning process, the cross-correlation between $S^o_{L_{\ell-1}}[k]_{p,m}$ and $S^o_{\delta_{\ell+1}}[k]_{p,m}$ is strong enough to overcome the number and destructive effect of between-category similarity terms, and on average, $S^o_{L_{\ell-1}}[k]_{p,m}S^o_{\delta_{\ell+1}}[k]_{p,m}$ is positive (Fig. 4C).

After the initial phase, this cross-correlation becomes weaker, causing the alignment terms to lose their initial amount of alignment, except for those whose orders are a multiple of 60 when no data shuffling is performed (Fig. 3). The reason for this is that as the network learns to discriminate between different categories, the response of neurons to the majority of the data points becomes saturated, and their error signals vanish (Fig. S2C), while error signals in response to some other data points, which are not learned well, remain large (Fig. S2C). The within-category similarity of data points whose corresponding error signals remain large (and consequently their corresponding similarity terms are dominant) is less than the data points whose corresponding

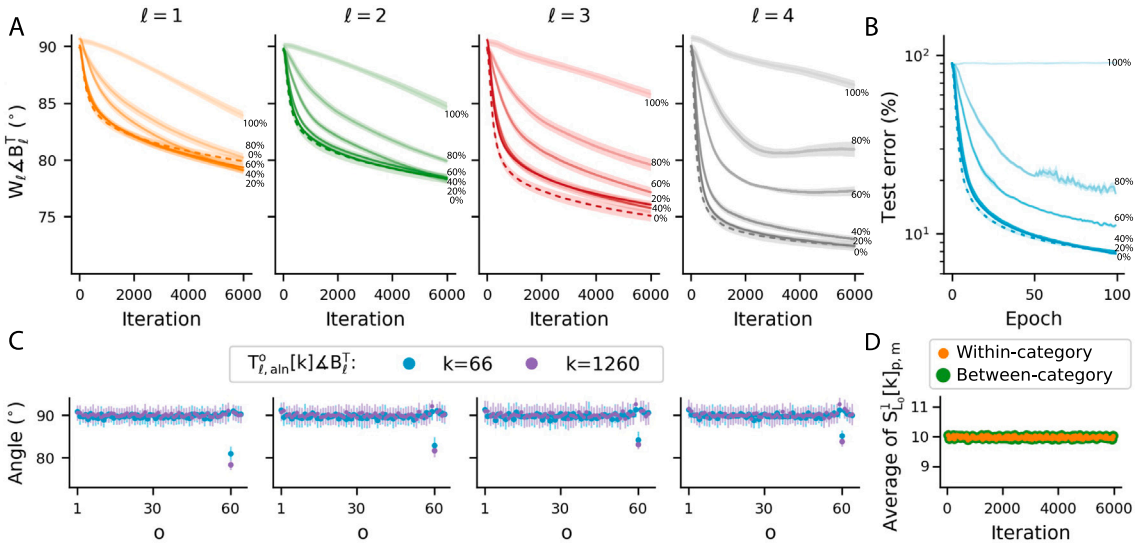


Fig. 5. The within-category similarity of data points enhances weight alignment, but a certain amount of weight alignment can happen even using data with random labels. (A) WA across layers of the example network trained on the MNIST dataset where labels of different percentages of data points are shuffled (in 20% label shuffling, 20% of the data points are randomly selected, and then available label options are randomly assigned to them with equal probability). Alignment occurs even when 100% of the data labels are shuffled. **(B)** Classification error of the same ANN across different percentages of label shuffling. As expected, the error increases for higher percentages of label shuffling and remains at the chance level (90%) if 100% of data labels are shuffled. **(C)** The behavior of alignment terms for two sample iterations $k = 66$ and $k = 1260$ when labels of 100% of the data points are shuffled once at the beginning of the learning process, and after that, no data shuffling is performed. Shuffling all labels spoils the cross-correlation between error and output signals of neurons and consequently spoils alignment of all orders of alignment terms, except for the alignment term of order 60 because of the repetition of the same mini-batches every 60 iterations (compare this panel with Fig. 3B). **(D)** The average of $S^o_{L_0}[k]_{p,m}$ for within- and between-category data points in the case of shuffling 100% of data labels. Shuffling all labels creates a situation where data points belonging to different categories become as similar to each other as the data points within a single category and consequently spoils the cross-correlation between error and output signals of neurons (compare this panel with the leftmost column of Fig. 4B). (A-D) Each dot or trace is the average over 10 runs and error bars and shaded areas are one s.d. around the mean.

error signals vanish (Fig. S2D,E). This weakens the cross-correlation between error and output signals of neurons, leading to the mentioned reduction in alignment. This leads to a reduction in the alignment of $\Delta W_{\ell,FA}$ with B^T_ℓ as the learning process proceeds (Fig. 3). In the graph of the alignment of $\Delta W_{\ell,FA}$ with B^T_ℓ (Fig. 3), it can also be seen that in the beginning, they are not aligned, then the alignment peaks and again decreases. This behavior is because there are k alignment terms in $\Delta W_{\ell,FA}[k]$ (according to Eq. (3)), and also there is another term $\eta f\left(\zeta^k_\ell[k]\right)^T \delta_{\ell+1,FA}[k]$ that does not align (Fig. S9). Hence, in the beginning, it takes some iterations for the number of alignment terms to increase and overcome the non-aligned term $\eta f\left(\zeta^k_\ell[k]\right)^T \delta_{\ell+1,FA}[k]$.

To verify the role of within-category similarity in deriving the alignment, we ran a simulation where we randomly selected different percentages of the data points and shuffled their true labels once at the beginning of the learning process. Such shuffling is expected to disrupt the cross-correlation between $S^o_{L_{\ell-1}}[k]_{p,m}$ and $S^o_{\delta_{\ell+1}}[k]_{p,m}$ and adversely affect the degree of WA (Fig. 5A). As expected, for high percentages of data points with randomly assigned labels, the total WA was degraded across all layers (Fig. 5A).

Shuffling the labels disrupted the learning and increased the test error (the test dataset was intact). In the case where 100% of data points were randomly labeled (samples of a given digit were equally likely to have one of the labels from 0 to 9) test error reached 90%, representing the chance level (Fig. 5B).

There was still some residual and slow WA even when all labels were shuffled (Fig. 5A). The reason for the impact of label shuffling becomes clear by looking at the behavior of alignment terms across lags. In the case where 100% of the data labels were shuffled and no data shuffling was performed at the beginning of epochs, alignment terms, except those whose orders were a multiple of 60, did not align considerably (Fig. 5C). In this case, $S^o_{L_{\ell-1}}[k]_{p,m}$ of both within- and between-category data points had the same distribution and average (Fig. 5D), and the feature that data points belonging to a single category are more similar to each other than those belonging to different categories was spoiled. Alignment terms whose orders were a multiple of 60 remained aligned because of the repetition of the identical mini-batches every 60 iterations, confirming the unignorable contribution of the autocorrelation of neural activity to alignment in addition to cross-correlation.

2.2.2. *Alignment and the local minimum reached by BP-TAW can be improved by weight normalization*

In the training of the network with the original formulation of BP-TAW, the Frobenius norms of the forward weight matrices continuously grew especially in the last layers (Fig. S3). This growth can contribute to the weakening of alignment by saturating the outputs of neurons and vanishing error signals as described above. To examine this, we limited and fixed the Frobenius norm of input weights to each neuron at each iteration by applying a WN method as follows

$$W_\ell[k]_{*,i} \leftarrow \gamma \frac{W_\ell[k]_{*,i}}{\|(W_\ell[k])_{*,i}\|_F}, \tag{15}$$

where $W_\ell[k]_{*,i}$ denotes an $n_\ell \times 1$ matrix consisting of the ith column of $W_\ell[k]$ and γ is a positive scalar which we refer to as WN gain. To treat all weights in the same way, we also applied this WN method to backward weights once at the beginning of the learning process. Unlike the conventional WN method in ANNs [39], which is a reparameterization of the BP formula, this proposed WN method is an intervention in the BP-TAW formula, which partially prevents the saturation of neurons and vanishing of error signals (Fig. S4).

Two aspects of the network that directly and simultaneously affect the amount of alignment are $\Delta W_{\ell,FA}[k] \measuredangle B^T_\ell$ and $\|\Delta W_{\ell,FA}[k]\|_F / \|W_\ell[k]\|_F$. This WN method improved both of these aspects (Fig. 6A,B) and consequently improved the alignment of forward weights with backward weights (Fig. 6C) and also the alignment between the update directions of BP-TAW and those of BP (Fig. 6D), making BP-TAW a better approximation of BP. In addition to the improvement of alignment, this WN method also improved test accuracy when the network was trained by BP-TAW (Fig. 6E).

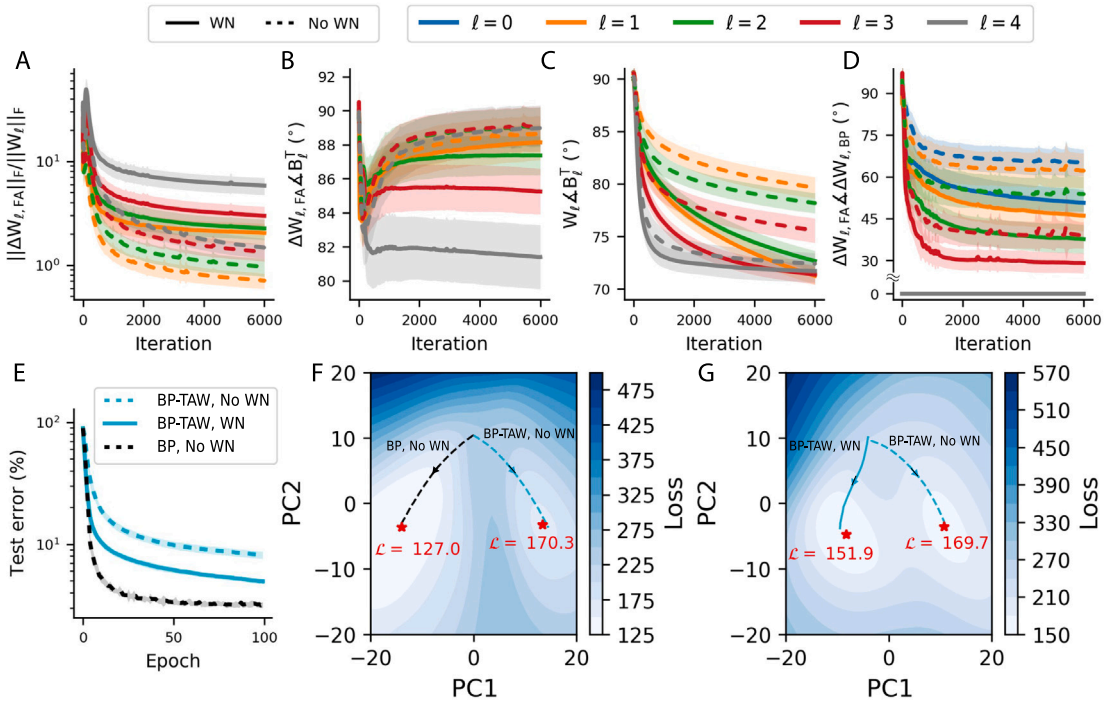


Fig. 6. Weight normalization in BP-TAW improves alignment and classification accuracy. (A) WN increases the Frobenius norms of the update directions of BP-TAW relative to the Frobenius norms of forward weights. **(B)** WN improves the alignment of the update directions of BP-TAW with backward weights. **(C)** WN improves WA. **(D)** WN improves the alignment between the update directions of BP-TAW and those of BP. In the last layer, the update direction of BP-TAW is the same as BP. **(E)** WN improves the test accuracy of BP-TAW. **(F)** Two-dimensional embedding of the trajectories of all learnable parameters of the network (forward weight matrices and bias vectors) for two instances of the network that are both identically initialized with the same parameters, but one is trained by BP and the other by BP-TAW. The contour map shows the loss function of the network using the reconstructed parameters. Red asterisks are local minimums in the two-dimensional space. **(G)** Similar to panel F but for networks trained by BP-TAW, with and without WN. (A-E) Each trace is the average over 10 runs and the shaded areas are one s.d. around the mean. (A,B,D) Each trace is passed through a moving average filter with a length of 60.

To make sure that the improvement in the test error and alignment was not just due to a chance selection of hyperparameters of the network (γ, η, and initial standard deviation of weights and biases), we did a sensitivity analysis and parameter sweep (Fig. S5). We examined a wide range of hyperparameter values with and without WN. In addition, to make sure that the improvement was not just due to the specific kind of initialization that the WN method imposes on the network, in a separate group of experiments, we only applied the WN at the beginning (at the first iteration) and did not apply it in the rest of the learning process. Among these cases, the best amounts of the test error and alignment belonged to the case where WN was applied throughout the entire learning process (Supplementary Note 1, Fig. S5), showing that the results were not just due to a chance selection of hyperparameters or merely the initialization of the network. Moreover, the sensitivity analysis and parameter sweep showed that the improvement in alignment and test accuracy of BP-TAW is robust for a fairly large range of hyperparameters.

We examined the generalizability of the effectiveness of WN to other network architectures. We re-performed the sensitivity analysis and parameter sweep described above on two other network architectures and observed that the WN method improved the test accuracy and alignment in them as well (Supplementary Note 1, Fig. S6, Fig. S7). Therefore, the application of the WN method in improving alignment is generalizable to other network architectures.

Low-dimensional embedding of all learnable parameters of the network (forward weight matrices and bias vectors) using principle component analysis showed that the initial mismatch between the update directions of BP-TAW and BP drives the trajectory of the parameters of BP-TAW into a different local minimum which is less optimal than the local minimum to which the network converges with BP (Fig. 6F). WN drives the trajectory of parameters to a more optimal local minimum, resulting in a better degree of classification accuracy (Fig. 6G).

We observed that without WN, alignment between W_ℓ and B_ℓ^T decreases in the early layers of the network. With WN, this reduction was overcome to some extent, and the first and third layers became more aligned than the last layer (Fig. 6C). However, we observed that the amount of alignment between the update directions of BP-TAW and BP decreases step by step as the error is backpropagated towards the input layer and even WN could not overcome this effect (Fig. 6D).

The potential increase of $\measuredangle\Delta W_{\ell,FA}\measuredangle\Delta W_{\ell,BP}$ (less alignment) in the earlier layers can be seen by comparing $\Delta W_{\ell,FA} = \eta L_\ell^T \delta_{\ell+1,FA}$ with $\Delta W_{\ell,BP} = \eta L_\ell^T \delta_{\ell+1,BP}$. The matrix L_ℓ^T is identical in both and the factors that determine the angle between them are $\delta_{\ell+1,BP}$ and $\delta_{\ell+1,FA}$. For simplicity, consider a d-layer linear ANN. For the last layer ($\ell = d$), we have $\delta_{d,FA} = \delta_{d,BP}$, but for $0 < \ell < d$, by using Eq. (2) successively, we have

$$\delta_{\ell,FA} = \delta_d B_{d-1} B_{d-2} B_{d-3} B_{d-4} \cdots B_{\ell+1} B_\ell \tag{16}$$

$$\delta_{\ell,BP} = \delta_d W_{d-1}^T W_{d-2}^T W_{d-3}^T W_{d-4}^T \cdots W_{\ell+1}^T W_\ell^T . \tag{17}$$

According to these two successive matrix multiplications of backward and the transpose of forward weight matrices, as the error is backpropagated towards the early layers, depending on the pairs of B_ℓ and W_ℓ^T, deviation of $\delta_{\ell,BP}$ from $\delta_{\ell,FA}$ potentially tends to increase. Consequently, deviation of $\Delta W_{\ell,FA}$ from $\Delta W_{\ell,BP}$ potentially increases as well and it reduces the accuracy of the update directions of BP-TAW compared to gradient directions computed by BP in the early layers of deep ANNs.

3. Discussion

Artificial neural networks and their learning paradigms have differences and similarities with biological neural networks. Specifically, the BP method needs a biologically implausible matching between

feedforward and feedback synaptic weights. The BP-TAW learning method [17] showed that an ANN can be trained with arbitrary feedback weights that are distinct from feedforward weights. In BP-TAW, forward weights partially align with backward weights during iterations, which leads to a partial alignment between the update directions of BP-TAW and BP and provides an approximation of BP.

3.1. Mathematical basis of alignment

In this work, we demonstrated mathematical and statistical basis of WA (Fig. 1) and showed that WA is not a direct consequence of the learning, reduction of loss function, or the growth of the norms of the weights; rather, it relies on the structure of alignment terms that are extracted from the update rule of BP-TAW Eqs. (3) and (4), and according to this structure, alignment happens robustly under a variety of conditions depending on the statistical properties of neural activity. Specifically, we showed that the autocorrelation of error and output signals of neurons and the cross-correlation between them are two important features of neural activity contributing to alignment (Fig. 2).

We used alignment terms as a tool to analyze BP-TAW in a specific five-layer nonlinear ANN trained on the MNIST dataset. We showed how the arrangement of data in mini-batches and the repetition of data points across epochs contribute to the behavior of alignment terms by shaping autocorrelated neural activity (Fig. 3). Moreover, we showed that the relative similarity of data points of a single category and their differences across categories, which is an intrinsic property of datasets, contributes to alignment by shaping cross-correlated neural activity (Fig. 4, Fig. 5).

The demonstrated mathematical framework furthers our understanding of FA and WA and makes us capable of analyzing WA in BP-TAW under various conditions. In general, many aspects of neural activity influence the behavior of alignment terms and make them act differently in different situations. For example, the architecture of the network (activation function, number of neurons in layers, number of layers, loss function, normalization methods, etc.), hyperparameters (learning rate, batch size, etc.), and properties of the dataset affect neural activity and consequently the behavior of alignment terms. The presented framework can be used in various network architectures and configurations beyond what was discussed in this work.

A weakness of BP-TAW as an approximation of BP in deep ANNs is that the amount of alignment between the update directions of BP-TAW and BP potentially tends to decline as the error is consecutively backpropagated towards the earlier layers (Fig. 6D). In other words, with BP-TAW, early layers potentially receive less accurate supervised error signals compared to the final layers. This potential decline may be overcome by unsupervised learning under certain conditions, which can be the subject of future work. Indeed, many aspects of the activity of neurons in lower areas of the visual system are demonstrated to be attainable with unsupervised learning models [40,41] and there are also suggestions of efficient network architectures where an ANN trained in an unsupervised manner is followed by a supervised classifier [42].

3.2. Limitations and future research directions

While we have investigated the validity of the introduced simplifying assumptions for the analysis of WA and alignment terms in deep feed-forward fully connected ANNs, their validity in biological neural networks and other types of ANNs remains to be investigated in future research. The violation of these assumptions does not render our provided framework useless but makes the analysis of WA and alignment terms complicated. Exploring mathematical methods to deal with the complexity of the analysis of alignment terms in their original form (without simplifying assumptions), and identifying other eligible assumptions that simplify the analysis under various conditions, represent two avenues for future studies.

The weight transport problem is one of the biological implausibilities of the BP formula which can be avoided by using BP-TAW, and there are other biological implausibilities in BP and BP-TAW [2,43]. For instance, the firing rate as the output of each biological neuron is nonnegative, while error signals in BP and BP-TAW are signed. In addition, error signals in BP and BP-TAW are distinct from the output of artificial neurons. In BP-TAW and BP, error signals are internal attributes of neurons that are backpropagated to other neurons through feedback weights, whereas in biological networks, the attribute of neurons that is conveyed explicitly to other neurons by axons and synapses is their output spikes and it is believed that other internal attributes of them are mostly local [2,4]. Future experiments and data analysis are needed to investigate whether a learning method similar to BP-TAW contributes to shaping synaptic plasticity and learning in biological neural networks.

Brain-inspired spiking neural networks and neuromorphic hardware are considered promising candidates to deal with the challenges of conventional ANNs and realize high-level intelligence and low power consumption [44–48]. Similar to biological neural networks, having bidirectional synapses in neuromorphic hardware is challenging [49] and the application of the FA and DFA learning methods in them is suggested [49–51]. Application and analysis of FA in neuromorphic hardware and spiking neural networks warrant an investigation in the future.

We examined the effect of a WN method, which works by fixing the Frobenius norm of input weights of each neuron to some constant and showed that it can improve alignment. Various forms of plasticity, such as heterosynaptic plasticity, are reported which regulate synaptic weights in a competitive manner, in which the potentiation of one synapse can result in the depression of other synapses to keep overall synaptic strengths under control [52,53]. There are also numerous reports of normalization mechanisms in biological neural networks working to regulate the activity of neurons and limit the dynamic range of synaptic weights [52,54]. The effect of different forms of biologically relevant WN methods on WA and FA could be explored in future studies.

3.3. Conclusion

In summary, the analysis done in this study provides a useful framework for understanding WA in BP-TAW and paves the way for further research on the relationship between learning methods used in ANNs and learning mechanisms in the nervous system. While BP-TAW is capable of approximating the weight update directions proposed by BP in simple feedforward networks and WN can improve this approximation, it remains to be seen how the addition of other biological considerations such as lateral connections, sparsity, synaptic pruning and formation, and the segregation of excitatory and inhibitory neurons affect the performance of BP-TAW.

4. Methods

4.1. BP and BP-TAW learning methods

In BP, we updated bias vectors and weight matrices at each iteration as below

$$W_\ell[k+1] = W_\ell[k] + \Delta W_\ell[k] \tag{18}$$

$$\mathbf{b}_\ell[k+1] = \mathbf{b}_\ell[k] + \Delta \mathbf{b}_\ell[k], \tag{19}$$

where gradient directions computed by BP for updating bias vectors and weight matrices at each iteration k are

$$\Delta W_{\ell,BP}[k] = -\eta \left.\frac{\partial \mathcal{L}}{\partial W_\ell}\right|_k = \eta L_\ell[k]^T \delta_{\ell+1,BP}[k],\ 0 \le \ell < d \tag{20}$$

$$\Delta \mathbf{b}_{\ell,BP}[k] = -\eta \left.\frac{\partial \mathcal{L}}{\partial \mathbf{b}_\ell}\right|_k = \eta J_{1\times n_b} \delta_{\ell,BP}[k],\ 0 < \ell \le d, \tag{21}$$

where $J_{1\times n_b}$ is a $1\times n_b$ all-ones matrix and error matrices of neurons are

$$\delta_{d,BP}[k] = E[k] \odot f'\big(Z_d[k]\big) \tag{22}$$

$$\delta_{\ell,BP}[k] = \delta_{\ell+1,BP}[k] W_\ell[k]^T \odot f'\big(Z_\ell[k]\big),\ 0 < \ell < d, \tag{23}$$

where $E[k] = Y^*[k] - Y[k]$ according to the loss function $\mathcal{L}[k] = \frac{1}{2}\sum_{i,j} E[k]_{i,j}^2$ [1].

In BP-TAW [17], the error is backpropagated through constant random matrices different from forward weights which are denoted by $B_\ell \in \mathbb{R}^{n_{\ell+1}\times n_\ell}$, and we calculated the update directions at each iteration as follows (W_ℓ^T in Eq. (23) is replaced with B_ℓ)

$$\delta_{d,FA}[k] = \delta_{d,BP}[k] = E[k] \odot f'\big(Z_d[k]\big) \tag{24}$$

$$\delta_{\ell,FA}[k] = \delta_{\ell+1,FA}[k] B_\ell \odot f'\big(Z_\ell[k]\big),\ 0 < \ell < d \tag{25}$$

$$\Delta W_{\ell,FA}[k] = \eta L_\ell[k]^T \delta_{\ell+1,FA}[k],\ 0 \le \ell < d \tag{26}$$

$$\Delta \mathbf{b}_{\ell,FA}[k] = \eta J_{1\times n_b} \delta_{\ell,FA}[k],\ 0 < \ell \le d. \tag{27}$$

In training ANNs with BP-TAW, $\Delta W_{\ell,BP}[k]$ is a direction that we only calculated at each iteration for comparison with $\Delta W_{\ell,FA}[k]$ (we only used $\Delta W_{\ell,FA}[k]$ to update forward weight matrices).

4.2. Angle and cosine similarity between two matrices

We calculated the angle between two arbitrary matrices W and B, which have the same dimensions, as follows

$$W \measuredangle B = cos^{-1}\left(\frac{\langle W, B\rangle_F}{\|W\|_F \|B\|_F}\right), \tag{28}$$

where $\langle W, B\rangle_F$ is the Frobenius inner product of W and B and $\|.\|_F$ is the Frobenius norm. This is identical to the angle between vectorized W and B in the Euclidean space. The angle between two matrices is indeed a measure of the similarity between the normalized versions of the two matrices.

In addition to the angle, cosine similarity between two matrices can also be used as a measure of the similarity between them as follows

$$\text{cosine similarity}(W, B) = \frac{\langle W, B\rangle_F}{\|W\|_F \|B\|_F}. \tag{29}$$

Assuming W and B to be nonzero, since the denominator of the cosine similarity is always nonnegative, for alignment ($W \measuredangle B < 90°$, or equivalently $0 <$ cosine similarity(W, B)) it is sufficient and necessary that

$$0 < \langle W, B\rangle_F.$$

4.3. Network parameters, dimensions, and initialization

In our experiments, for nonlinearity, we chose $f(\cdot) = tanh(ReLU(\cdot))$, which roughly resembles the frequency-current curve of biological neurons. Moreover, since this is a classification task with the desired output of the network coded to be between zero and one, for the reasons of stability and convergence, it is convenient for the activation function of the output layer to be confined between zero and one. We chose the number of neurons and activation functions to be the same in the hidden layers and the output layer in order to ensure the comparability of the amount of alignment across different layers. We chose the number of neurons in each hidden and output layer to be 50 and since there were 10 classes, to match the length of the coding of the desired output of the network with the number of neurons in the last layer, we coded the labels of classes with mutually exclusive 5-hot coding (see the following section of Methods). To reduce the computational cost, we resized all handwritten digits (data points of MNIST) to images with 15×15 pixels which were then transformed into a vector. Hence, the number of input neurons was 225. We also normalized input data points (output signals of input neurons) to lie between 0 and 1 (dividing the original MNIST data points by 255). We chose the batch size to be 1000, which means there were 60 mini-batches given the total number of 60 000 training data points (each epoch of training consisted of 60 iterations). At the beginning of each run, we randomly initialized elements of forward and backward weights and bias vectors independently from $\mathcal{N}(0, 0.1)$. The loss function that we used was $\mathcal{L}[k] = \frac{1}{2}\sum_{i,j} E[k]_{i,j}^2$, where $E[k]_{i,j}$ is the element in the ith row and the jth column of $E[k] = Y^*[k] - Y[k]$. In Figs. 3, 4, 5, and 6 learning rate was $\eta = 0.0005$.

In Fig. 2 we chose network dimensions to be $n_0 = n_2 = 20$, $n_1 = 100$, and $n_b = 100$, and we set $\eta = 0.0004$ and initialized elements of B_1, W_0 and W_1 with i.i.d. random variables from $\mathcal{N}(0, 1)$.

4.4. Generating mutually exclusive n-hot coding

In training the ANN on MNIST, we used mutually exclusive 5-hot coding. Suppose the number of categories is C and the number of output neurons is m ($n \cdot C \le m$). For generating mutually exclusive n-hot code vectors of size m for each category, we started from the first category to the last one, and successively for each category $c \in \{0, 1, \dots, C-1\}$ we initialized its code vector with zero elements and then randomly selected n out of $m - c \cdot n$ elements that were not equal to 1 in any of the c previously coded category vectors and set them equal to 1.

CRediT authorship contribution statement

Alireza Rahmansetayesh: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Formal analysis, Data curation, Conceptualization. **Ali Ghazizadeh:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Methodology, Formal analysis, Conceptualization. **Farokh Marvasti:** Writing – review & editing, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data and code availability statement

The MNIST dataset can be found at: http://yann.lecun.com/exdb/mnist/. The code for reproducing all results in this work is available under the Apache 2.0 license at https://github.com/ARahmansetayesh/The-underlying-mechanisms-of-alignment-in-error-backpropagation-through-arbitrary-weights. We used PyTorch library only for accelerating computations on GPU (we did not use PyTorch's automatic differentiation capability).

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.neucom.2024.128587.

References

[1] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[2] D.G. Stork, Is backpropagation biologically plausible, in: International Joint Conference on Neural Networks, Vol. 2, IEEE Washington, DC, 1989, pp. 241–246.

[3] F. Crick, The recent excitement about neural networks, Nature 337 (6203) (1989) 129–132.

[4] Y. Song, T. Lukasiewicz, Z. Xu, R. Bogacz, Can the brain do backpropagation?—exact implementation of backpropagation in predictive coding networks, NeuRIPS Proceedings 2020 33 (2020) (2020).

[5] S. Grossberg, Competitive learning: From interactive activation to adaptive resonance, Cognit. Sci. 11 (1) (1987) 23–63.

[6] D. Zipser, R.A. Andersen, A back-propagation programmed network that simulates response properties of a subset of posterior parietal neurons, Nature 331 (6158) (1988) 679–684.

[7] S.-M. Khaligh-Razavi, N. Kriegeskorte, Deep supervised, but not unsupervised, models may explain IT cortical representation, PLoS Comput. Biol. 10 (11) (2014) e1003915.

[8] C.F. Cadieu, H. Hong, D.L. Yamins, N. Pinto, D. Ardila, E.A. Solomon, N.J. Majaj, J.J. DiCarlo, Deep neural networks rival the representation of primate IT cortex for core visual object recognition, PLoS Comput. Biol. 10 (12) (2014) e1003963.

[9] R.M. Cichy, A. Khosla, D. Pantazis, A. Torralba, A. Oliva, Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence, Sci. Rep. 6 (1) (2016) 1–13.

[10] A. Nayebi, D. Bear, J. Kubilius, K. Kar, S. Ganguli, D. Sussillo, J.J. DiCarlo, D.L. Yamins, Task-driven convolutional recurrent models of the visual system, in: Advances in Neural Information Processing Systems, 2018, pp. 5290–5301.

[11] J.C. Whittington, R. Bogacz, Theories of error back-propagation in the brain, Trends Cognit. Sci. 23 (3) (2019) 235–250.

[12] J.C. Whittington, R. Bogacz, An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity, Neural Comput. 29 (5) (2017) 1229–1262.

[13] T.P. Lillicrap, A. Santoro, L. Marris, C.J. Akerman, G. Hinton, Backpropagation and the brain, Nat. Rev. Neurosci. (2020) 1–12.

[14] X. Xie, H.S. Seung, Equivalence of backpropagation and contrastive hebbian learning in a layered network, Neural Comput. 15 (2) (2003) 441–454.

[15] J.F. Kolen, J.B. Pollack, Backpropagation without weight transport, in: Proceedings of 1994 IEEE International Conference on Neural Networks, ICNN'94, Vol. 3, IEEE, 1994, pp. 1375–1380.

[16] Q. Liao, J. Leibo, T. Poggio, How important is weight symmetry in backpropagation? in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, (1) 2016.

[17] T.P. Lillicrap, D. Cownden, D.B. Tweed, C.J. Akerman, Random synaptic feedback weights support error backpropagation for deep learning, Nat. Commun. 7 (1) (2016) 1–10.

[18] A. Nøkland, Direct feedback alignment provides learning in deep neural networks, 2016, arXiv preprint arXiv:1609.01596.

[19] M. Refinetti, S. d'Ascoli, R. Ohana, S. Goldt, The dynamics of learning with feedback alignment, 2020, arXiv preprint arXiv:2011.12428.

[20] C. Frenkel, M. Lefebvre, D. Bol, Learning without feedback: Direct random target projection as a feedback-alignment algorithm with layerwise feedforward training, stat 1050 (2019) 3.

[21] J. Launay, I. Poli, F. Krzakala, Principled training of neural networks with direct feedback alignment, 2019, arXiv preprint arXiv:1906.04554.

[22] P. Baldi, P. Sadowski, Z. Lu, Learning in the machine: Random backpropagation and the deep learning channel, Artif. Intell. 260 (2018) 1–35.

[23] S. Bartunov, A. Santoro, B.A. Richards, L. Marris, G.E. Hinton, T. Lillicrap, Assessing the scalability of biologically-motivated deep learning algorithms and architectures, 2018, arXiv preprint arXiv:1807.04587.

[24] T.H. Moskovitz, A. Litwin-Kumar, L. Abbott, Feedback alignment in deep convolutional networks, 2018, arXiv preprint arXiv:1812.06488.

[25] P. Züge, C. Klos, R.-M. Memmesheimer, Weight versus node perturbation learning in temporally extended tasks: Weight perturbation often performs similarly or better, Phys. Rev. X 13 (2) (2023) 021006.

[26] G. Cauwenberghs, A fast stochastic error-descent algorithm for supervised learning and optimization, Adv. Neural Inf. Process. Syst. 5 (1992).

[27] R.J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Mach. Learn. 8 (1992) 229–256.

[28] S. Dalm, M. van Gerven, N. Ahmad, Effective learning with node perturbation in deep neural networks, 2023, arXiv preprint arXiv:2310.00965.

[29] N. Hiratani, Y. Mehta, T. Lillicrap, P.E. Latham, On the stability and scalability of node perturbation learning, Adv. Neural Inf. Process. Syst. 35 (2022) 31929–31941.

[30] V. Francioni, V.D. Tang, N.J. Brown, E.H. Toloza, M. Harnett, Vectorized instructive signals in cortical dendrites during a brain-computer interface task, 2023, bioRxiv.

[31] P.C. Humphreys, K. Daie, K. Svoboda, M. Botvinick, T.P. Lillicrap, BCI learning phenomena can be explained by gradient-based optimization, 2022, bioRxiv 2022-2012.

[32] J. Tigges, W. Spatz, M. Tigges, Reciprocal point-to-point connections between parastriate and striate cortex in the squirrel monkey (Saimiri), J. Comp. Neurol. 148 (4) (1973) 481–489.

[33] M. Wong-Riley, Reciprocal connections between striate and prestriate cortex in squirrel monkey as demonstrated by combined peroxidase histochemistry and autoradiography, Brain Res. 147 (1) (1978) 159–164.

[34] R.D. D'Souza, Q. Wang, W. Ji, A.M. Meier, H. Kennedy, K. Knoblauch, A. Burkhalter, Hierarchical and nonhierarchical features of the mouse visual cortical network, Nat. Commun. 13 (1) (2022) 503.

[35] M. Akrout, C. Wilson, P.C. Humphreys, T. Lillicrap, D. Tweed, Deep learning without weight transport, 2019, arXiv preprint arXiv:1904.05391.

[36] D. Kunin, A. Nayebi, J. Sagastuy-Brena, S. Ganguli, J. Bloom, D. Yamins, Two routes to scalable credit assignment without weight symmetry, in: International Conference on Machine Learning, PMLR, 2020, pp. 5511–5521.

[37] W. Xiao, H. Chen, Q. Liao, T. Poggio, Biologically-plausible learning algorithms can scale to large datasets, 2018, arXiv preprint arXiv:1811.03567.

[38] B. Crafton, A. Parihar, E. Gebhardt, A. Raychowdhury, Direct feedback alignment with sparse connections for local learning, Front. Neurosci. 13 (2019) 525.

[39] T. Salimans, D.P. Kingma, Weight normalization: A simple reparameterization to accelerate training of deep neural networks, 2016, arXiv preprint arXiv:1602.07868.

[40] B.A. Olshausen, D.J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature 381 (6583) (1996) 607–609.

[41] H.B. Barlow, et al., Possible principles underlying the transformation of sensory messages, Sensory Commun. 1 (01) (1961).

[42] S.R. Kheradpisheh, M. Ganjtabesh, S.J. Thorpe, T. Masquelier, STDP-based spiking deep convolutional neural networks for object recognition, Neural Netw. 99 (2018) 56–67.

[43] A.H. Marblestone, G. Wayne, K.P. Kording, Toward an integration of deep learning and neuroscience, Front. Comput. Neurosci. 10 (2016) 94.

[44] S. Yang, B. Linares-Barranco, B. Chen, Heterogeneous ensemble-based spike-driven few-shot online learning, Front. Neurosci. 16 (2022) 850932.

[45] S. Yang, J. Tan, B. Chen, Robust spike-based continual meta-learning improved by restricted minimum error entropy criterion, Entropy 24 (4) (2022) 455.

[46] S. Yang, B. Chen, SNIB: improving spike-based machine learning using nonlinear information bottleneck, IEEE Trans. Syst. Man Cybern.: Syst. (2023).

[47] S. Yang, B. Chen, Effective surrogate gradient learning with high-order information bottleneck for spike-based machine intelligence, IEEE Trans. Neural Netw. Learn. Syst. (2023).

[48] S. Yang, H. Wang, B. Chen, Sibols: robust and energy-efficient learning for spike-based machine intelligence in information bottleneck framework, IEEE Trans. Cogn. Dev. Syst. (2023).

[49] A. Renner, F. Sheldon, A. Zlotnik, L. Tao, A. Sornborger, The backpropagation algorithm implemented on spiking neuromorphic hardware, 2021, arXiv preprint arXiv:2106.07030.

[50] C. Wolters, B. Taylor, E. Hanson, X. Yang, U. Schlichtmann, Y. Chen, Biologically plausible learning on neuromorphic hardware architectures, in: 2023 IEEE 66th International Midwest Symposium on Circuits and Systems, MWSCAS, IEEE, 2023, pp. 733–737.

[51] S.-T. Lee, J.-H. Lee, Neuromorphic computing using random synaptic feedback weights for error backpropagation in NAND flash memory-based synaptic devices, IEEE Trans. Electron Devices 70 (3) (2023) 1019–1024.

[52] M. Chistiakova, N.M. Bannon, J.-Y. Chen, M. Bazhenov, M. Volgushev, Homeostatic role of heterosynaptic plasticity: models and experiments, Front. Comput. Neurosci. 9 (2015) 89.

[53] G.G. Turrigiano, The dialectic of Hebb and homeostasis, Phil. Trans. R. Soc. B 372 (1715) (2017) 20160258.

[54] G.-q. Bi, M.-m. Poo, Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type, J. Neurosci. 18 (24) (1998) 10464–10472.

Alireza Rahmansetayesh has received his Master's degree from Sharif University of Technology in Electrical Engineering and his Bachelor's degree from Shiraz University in Electrical Engineering. His research field is at the intersection of neuroscience and artificial neural networks.

Ali Ghazizadeh is an associate professor in the Electrical Engineering Department at Sharif University of Technology. His laboratory uses a combination of theoretical, computational, and experimental techniques including neural network modeling to address neural mechanisms of learning and memory in non-human and human primates.

Prof. Farokh Marvasti received his undergraduate and graduate degrees all from RPI (Troy, NY) in 1973. From 1972-1975, he worked at Graphic Sciences and Singer-Kearfott in USA, where he worked on new digital facsimile and channel codings, respectively. He then joined Sharif University of Technology, where he helped founding Iran Telecommunication and Electric Power Research Centers. In 1984, he spent his sabbatical at the University of California, Davis where he taught several graduate courses in addition to research. He then joined AT&T Bell Labs for several years before joining IIT in Chicago. After extensive consulting on developing new digital video coding, he joined King's College, University of London in 1991. After retiring from King's College, he joined Sharif University as a full professor again, where he founded ACRI (Advanced Communications Research Center). He spent his sabbatical leave at the Communications and Information Systems Group of University College London (UCL) in 2013, where he published a seminal paper on Spectral Efficient Frequency Division Multiplexing (SEFDM).

Prof. Marvasti has published several books on nonuniform sampling and many book chapters on sparse signal processing, about 250 Journal papers and several hundred conference papers all in signal processing, communications and information theory. He also holds several US patents on Analog to digital conversions and image denoising.

He was one of the editors of the IEEE Trans on Communications from 1990–1995 and an Associate editor of IEEE Trans on Signal Processing from 1994&-1997. Prof Marvasti has received a distinguished award from the Iranian Academy of Sciences in 2014 and an award rom the Iranian National Science Foundation for a 5 year term chair position in 2015. He was also appointed as a distinguished researcher by IEEE Iran Chapter in 2018.