The underlying mechanisms of alignment in error backpropagation through arbitrary weights

Alireza Rahmansetayesh[1], Ali Ghazizadeh[1,*], Farokh Marvasti[1]

[1]Electrical Engineering Department, Sharif University of Technology, Tehran, Iran

[*] Corresponding author, E-mail: ghazizadeh@sharif.edu

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 quest 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 "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 show that the occurrence of WA is governed 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.

Keywords— backpropagation, feedback alignment, weight transport problem, bio-inspired artificial neural networks, weight normalization

1 Introduction

For the past four decades, BP has been the dominant learning method used in artificial neural networks (Rumelhart et al., 1985). However, BP is known to be implausible in the nervous system (Stork, 1989; Crick, 1989; Song et al., 2020). One of its major issues is known as "weight transport problem" (Grossberg, 1987) which refers to the requirement that backward weights should be precisely matched with 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 (Zipser and Andersen, 1988; Khaligh-Razavi and Kriegeskorte, 2014; Cadieu et al., 2014; Cichy et al., 2016; Nayebi et al., 2018), and possibilities for calculation of approximate gradient directions in the brain are suggested (Whittington and Bogacz, 2019, 2017; Lillicrap et al., 2020; Xie and Seung, 2003). In particular, it has been shown that learning occurs even without exact weight transport (Kolen and Pollack, 1994; Liao et al., 2016) and when arbitrary weights that are distinct from forward ones backpropagate error signals to early layers (Lillicrap et al., 2016), a method which we refer to as 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 reduces and this partial alignment leads to weight update directions that are partially aligned with weight update directions calculated by BP, thereby providing effective teaching signals (Lillicrap et al., 2016). 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 (Nøkland, 2016; Refinetti et al., 2020; Frenkel et al., 2019; Launay et al., 2019; Baldi et al., 2018). Interestingly, 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 (Bartunov et al., 2018; Launay et al., 2019; Moskovitz et al., 2018).

Although there are some investigations on the favorable conditions for WA and for the improvement of learning using BP-TAW (Refinetti et al., 2020; Frenkel et al., 2019; Moskovitz et al., 2018; Akrout et al., 2019; Kunin et al., 2020; Xiao et al., 2018; Baldi et al., 2018; Lillicrap et al., 2016), 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 (Lillicrap et al., 2016; Frenkel et al., 2019). 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 (Lillicrap et al., 2016), or a scalar multiple of the Moore-Penrose pseudo-inverse of the product of forward weight matrices in the case of DFA (Frenkel et al., 2019). Under these circumstances, update directions of BP-TAW are an approximation of the Gauss-Newton optimization method (Lillicrap et al., 2016). However, these proofs cannot explain the occurrence of WA for arbitrary initialization of weights (supplementary figure 13 of Lillicrap et al. 2016) and for nonlinear networks.

There are other preliminary attempts to explain the occurrence of alignment. In particular, Lillicrap et al. 2016 have provided an insight into the mechanics of FA by freezing forward weights in different stages of the learning process of an ANN trained by BP-TAW, showing that information about 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 restricted condition where input of a 2-layer linear network is white noise and the network is trained to learn a linear function, continuous growth of norm of weight matrices results in alignment (supplementary note 12 of Lillicrap et al. 2016). We will show that although the original form of BP-TAW is accompanied by a continuous growth of the norms of the weight matrices, this can be detrimental to WA and that limiting the norms of weights can improve WA.

In this work, we first explore the mathematical basis of WA in the general case. We show that occurrence of alignment does not rely on reduction of the loss function; rather, it is driven by statistical properties of neural activity such as cross-correlation and autocorrelation of error and output signals of neurons. Afterwards, we dissect WA in a practical application of BP-TAW for training a specific nonlinear 5-layer ANN on the MNIST dataset. We show that the relative similarity of data points that belong to a single category compared to the ones that belong to different categories contributes to alignment by shaping cross-correlated neural activity. Then, we will compare the trajectories of the forward weights of the network trained by BP and BP-TAW using low dimensional embedding and show that in BP-TAW the network convergences to a different and less optimal local minimum than BP, but the performance of BP-TAW can be improved by limiting the Frobenius norms of the input weights to each neuron.

2 Results

2.1 Driving alignment terms to explain the occurrence of alignment

Consider a conventional d layer ANN. We denote the matrices of forward weights (with arbitrary initialization), 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 batch size, n_ℓ is the number of neurons in the layer ℓ (network dimensions), and $f(\cdot)$ is an element-wise activation function (the following analysis still holds true 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 $0 < \ell \leq d$, internal states of neurons in the 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 the 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 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.

In BP-TAW (Lillicrap et al., 2016), the error is backpropagated through constant arbitrary matrices (different from

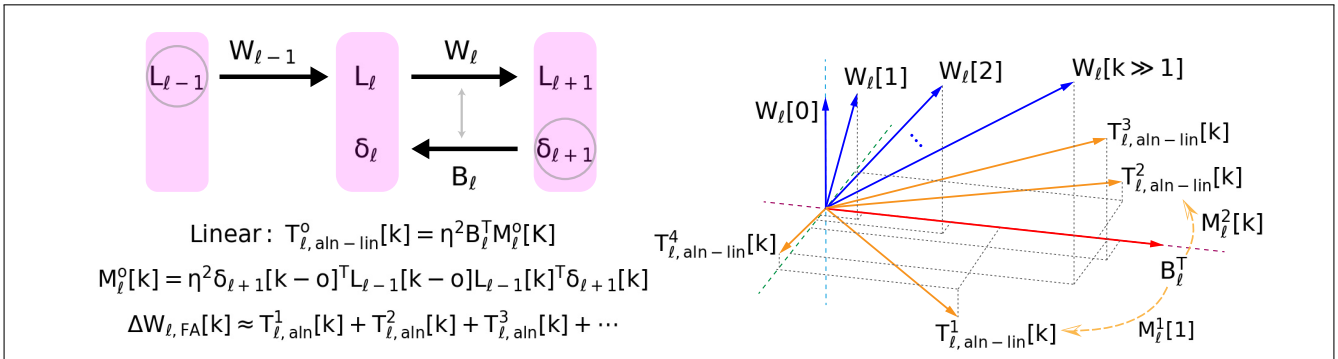

Figure 1 | The underlying mechanism of weight alignment in BP-TAW. Expansion of $\Delta W_{\ell,FA}[k]$ reveals alignment terms $T^o_{\ell,aln}[k]$ (equations 3, 4, and 5). In linear alignment terms, $M^o_\ell[k]$ acts as a transformation matrix on B^T_ℓ and if it partially preserves the direction of B^T_ℓ after the matrix multiplication, it propels $W_\ell[k]$ towards $B^T_\ell[k]$. $L_\ell[k]$ 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 the layer ℓ. Due to the structure of $M^o_\ell[k]$, alignment terms can robustly propel forward weight matrices (W_ℓ) towards 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 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 (equation 4, see also Supplementary Note 2). An order ($1 \leq o \leq k$) is assigned to alignment terms since in each of them the activity of neurons in iterations of k and $k-o$ (lag of o) are multiplied together.

forward weights) denoted by $B_\ell \in \mathbb{R}^{n_{\ell+1} \times n_\ell}$, and weight update directions are calculated at each iteration $k \geq 0$ according to

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

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

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

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

To demonstrate why this update rule leads to alignment, $\Delta W_{\ell,FA}[k]$ can be expanded by taking successive steps backward along the iterations and substituting every $W_\ell[k-o]$ for $0 \leq o < k$. Assuming update steps to be small, by applying first-order Taylor approximation we have

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

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

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

as *alignment term of order o* corresponding to layer ℓ (see Supplementary Note 1 for higher-order Taylor approximation and Supplementary Note 2 for index notation of alignment terms).

Alignment terms can provide alignment owing to their structure (Fig. 1). For simplicity, consider linear case of the alignment terms (ignoring element-wise matrix multiplications in the equation 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, 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 totally deviates the direction and $B^T_\ell \measuredangle B^T_\ell M^o_{\ell,skew}[k] = 90^\circ$ (see Supplementary Note 3). Hence, the occurrence of alignment depends on $M^o_{\ell,sym}[k]$.

Under certain conditions, some simplifying assumptions can be considered for the analysis of alignment terms. For example, initializing elements of B^T_ℓ independently from $\mathcal{N}(0, \sigma^2)$ is a common practice in ANNs. If elements of B^T_ℓ are independent random variables with a mean of zero and a variance of σ^2, and if M^o_ℓ is independent of the backward weights (validity of this assumption will be discussed in a practical nonlinear ANN below), the alignment is expected if the mean of eigenvalues of $M^o_{\ell,sym}$ (or equivalently the mean of eigenvalues of $M^o_\ell[k]$) is positive. To see this, we can examine the

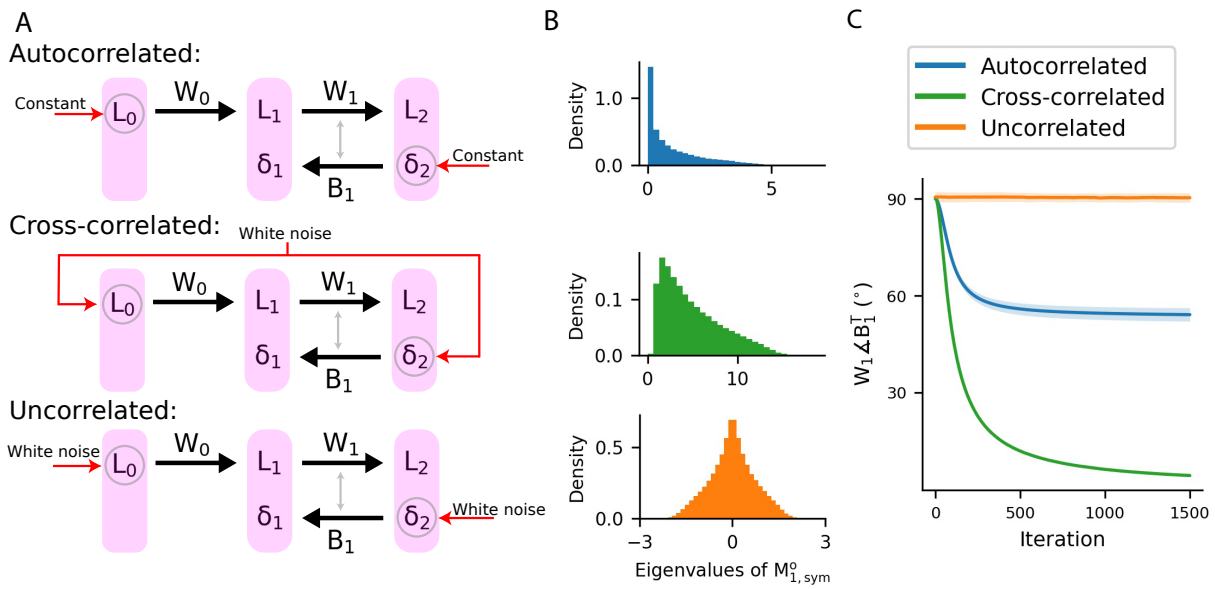

Figure 2 | Weight alignment can occur in the absence of meaningful input and error depending on the statistical properties of neural activity. (A) In an open-loop nonlinear 2-layer ANN, hypothetical output signals are imposed on the neurons in the input layer (L_0) and error signals of neurons in the 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 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 are 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.

Frobenius inner product of $T^o_{\ell,aln}[k]$ and B^T_ℓ, which is positive if and only if $T^o_{\ell,aln}[k]$ is partially aligned with B^T_ℓ, and evaluate its expected value as follows

$$\mathbb{E}(\langle B^T_\ell, T^o_{\ell,aln-lin}[k]\rangle_F) \approx tr(\mathbb{E}(B_\ell B^T_\ell)\mathbb{E}(M^o_\ell)) = n_\ell \sigma^2 \mathbb{E}(tr(M^o_\ell)) = n_\ell \sigma^2 \mathbb{E}(tr(M^o_{\ell,sym})) = n_\ell \sigma^2 \mathbb{E}(\sum_i \lambda^{o,\ell}_i). \tag{7}$$

where $\lambda^{o,\ell}_i$ denotes the eigenvalues of $M^o_{\ell,sym}$. Equivalently, alignment is expected if $\mathbb{E}(\bar{\lambda}^{o,\ell}) > 0$ where $\bar{\lambda}^{o,\ell} = \frac{1}{n_{\ell+1}} \sum_i \lambda^{o,\ell}_i$. For example, in an extreme case, if $M^o_{\ell,sym}$ is positive semidefinite and transform an arbitrary vector, it scales each of the components 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. However, $M^o_{\ell,sym}$ being semidefinite is not necessary and given the above assumptions, alignment is expected as long as $\bar{\lambda}^{o,\ell}$ is positive. There can also be more complex conditions where $\bar{\lambda}^{o,\ell}$ is positive, but rows of B^T_ℓ lie near some eigenvectors whose corresponding eigenvalues are negative. In this case, alignment does not occur despite having a positive $\bar{\lambda}^{o,\ell}$. However, given a random and independent B_ℓ (absence of any strong dependence between B_ℓ and $M^o_{\ell,sym}$) such a condition is unlikely.

The structure of alignment terms provides a robust basis for occurrence of alignment under a variety of conditions depending on the properties of neural activity. In particular, 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^o_ℓ into its constituent terms as follows

$$\begin{aligned} \mathbb{E}(\langle T^o_{\ell,aln-lin}[k], B^T_\ell\rangle_F) &\approx \eta^2 n_\ell \sigma^2 \mathbb{E}(tr(\delta_{\ell+1}[k-o]^T L_{\ell-1}[k-o] L_{\ell-1}[k]^T \delta_{\ell+1}[k])) = \\ \eta^2 n_\ell \sigma^2 \mathbb{E}(tr(\delta_{\ell+1}[k]\delta_{\ell+1}[k-o]^T L_{\ell-1}[k-o] L_{\ell-1}[k]^T)) &= \eta^2 n_\ell \sigma^2 \mathbb{E}\Big(tr(S^o_{\delta_{\ell+1}}[k]^T S^o_{L_{\ell-1}}[k])\Big) \end{aligned} \tag{8}$$

where we define $S^o_{L_{\ell-1}}[k] = L_{\ell-1}[k-o]L_{\ell-1}[k]^T$ as *output similarity matrix* of layer $\ell-1$ and $S^o_{\delta_{\ell+1}}[k] = \delta_{\ell+1}[k]\delta_{\ell+1}[k-o]^T$ as *error similarity matrix* of layer $\ell+1$. These two matrices represent the autocorrelation of output and error signals of neurons across lags, and their multiplication in the equation 8 represent the cross-correlation between them.

To verify this conclusion, we performed a simulation using an open-loop 2-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 across iterations, the transformation matrix $M^o_1 = \eta^2 \delta_2^T L_0 L_0^T \delta_2$ is a symmetric positive semidefinite matrix (Fig. 2B first row), and alignment happens as predicted (Fig. 2C blue trace). In this case, expected values of both $S^o_{\delta_2}[k]$ and $S^o_{L_0}[k]$ are scalar multiples of the identity matrix.

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 are white noise and are not autocorrelated, but they are 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 $o \neq 0$ are zero, they are positively cross-correlated, and alignment happens (Fig. 2C, green trace). In contrast, if all elements of L_0 and δ_2 are independently re-initialized from $\mathcal{N}(0,1)$ at each iteration, error signals of output neurons and output signals of input neurons are neither autocorrelated nor cross-correlated and alignment does not happen (Fig. 1B, orange trace). Notably, 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 does not happen (Fig. 2B).

Note that despite the simplifying assumption of linearity for the analysis of alignment terms, in these three scenarios, the network was nonlinear. Nonlinearity appears as two element-wise matrix multiplications in the structure of alignment terms (equation 4). However, using an increasing activation function, such as ReLU, causes the elements of the matrices resulted from $f'(\cdot)$ to be nonnegative, which alleviate the impact of nonlinearity. Moreover, given the actual behavior of the nonlinear network, there is no game-changing dependence between matrices resulted from $f'(\cdot)$ and other signals of the network (L_0 and δ_2) such that invalidates the conclusion of the linear analysis. Under these circumstances, we can regard nonlinearity as a distortion that impacts the amount of alignment but does not determine its occurrence (see Supplementary Note 2) , and for simplicity, we can ignore nonlinearity and refer to the linear case for the analysis of the nonlinear alignment terms (will be demonstrated in a practical nonlinear ANN below).

In DFA (Nøkland, 2016), 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 (Crafton et al., 2019). 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 4).

2.2 Investigating BP-TAW in training a 5-layer ANN on MNIST

To have a concrete example, in the subsequent sections, we examine the dynamics of alignment terms to investigate BP-TAW in the learning process of a 5-layer nonlinear fully connected ANN (Fig. 3A) for handwritten digits classification on MNIST dataset. For nonlinearity, we chose $f(\cdot) = tanh(ReLU(\cdot))$, which roughly resembles frequency-current curve of biological neurons. Moreover, since this is a classification task with 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 matched the number of neurons and also activation functions of all hidden and output layers so that the amount of alignment can be comparable between 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 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 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 60000 training data points (each epoch of training consisted of 60 iterations). We also examined the effect of data shuffling (rearrangement of data points among all mini-batches at the beginning of each epoch) and show that data shuffling has no considerable effect on total WA, while affects individual alignment terms.

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 $\mathcal{L}[k] = \frac{1}{2}\sum_{i,j} E[k]^2_{i,j}$, where $E[k]_{i,j}$ is the element in the ith row and the jth column of $E[k] = Y^*[k] - Y[k]$.

2.2.1 Effect of approximations and simplifying assumptions on alignment

In this section, we will evaluate the effects of approximations and assumptions used in derivation and analysis of alignment terms in the example network under investigation.

2.2.1.1 First-order Taylor approximation

In derivation of alignment terms for nonlinear networks (equation 3), we used first-order Taylor approximation. Namely, each order of alignment terms ($T^{o}_{\ell,aln},\ 1 \le o \le k$) is the first-degree Taylor polynomial of successive Taylor series and we have neglected higher-degree Taylor polynomials (remainder term), which induces some error in the analysis of WA. Moreover, in practice, the activation function used may have some nonanalytic points in its domain (for example, ReLU at zero), inducing additional error in the analysis of WA. Therefore, investigating the approximation error is an important step in verification of usefulness of the alignment terms.

We define the difference between the true $\Delta W_{\ell,FA}$ and its approximation as

$$R_\ell[k] = \Delta W_{\ell,FA}[k] - \sum_{o=1}^{k} T^{o}_{\ell,aln}[k] - \eta f\big(\zeta_\ell^k[k]\big)^T \delta_{\ell+1,FA}[k] \tag{9}$$

which we refer to as the remainder term of layer ℓ. The first-order Taylor approximation can be acceptable if $\|R_\ell\|_F$ is small relative to $\|\Delta W_{\ell,FA}\|_F$. It would be also interesting to see whether R_ℓ also aligns with B_ℓ^T. Analysis of this specific ANN shows that across iterations there is a modest increase in $\|R_\ell\|_F$ relative to $\|\Delta W_{\ell,FA}\|_F$, which is more pronounced in the final layers, but it remains small ($< 10\%$) relative to $\|\Delta W_{\ell,FA}\|_F$ (Fig. 3B). Angle analysis reveals that R_ℓ is not aligned with B_ℓ^T and even slightly opposes the alignment in the initial phase of the learning process (Fig. 3C), that is, remainder terms are not responsible for the occurrence of alignment and even slightly interfere with it. Hence, alignment terms are able to sufficiently explain the degree and dynamics of alignment in this network.

In addition to alignment terms, another term that appears in the Taylor series expansion is $\eta f\big(\zeta_\ell^k[k]\big)^T \delta_{\ell+1,FA}[k]$ (equation 4), which is the zero-degree Taylor polynomial of the last Taylor series expansion used for extracting the alignment term of order $o = k$. This term does not have the beneficial structure of the alignment terms and does not show a considerable amount of alignment in this network (Fig. S1).

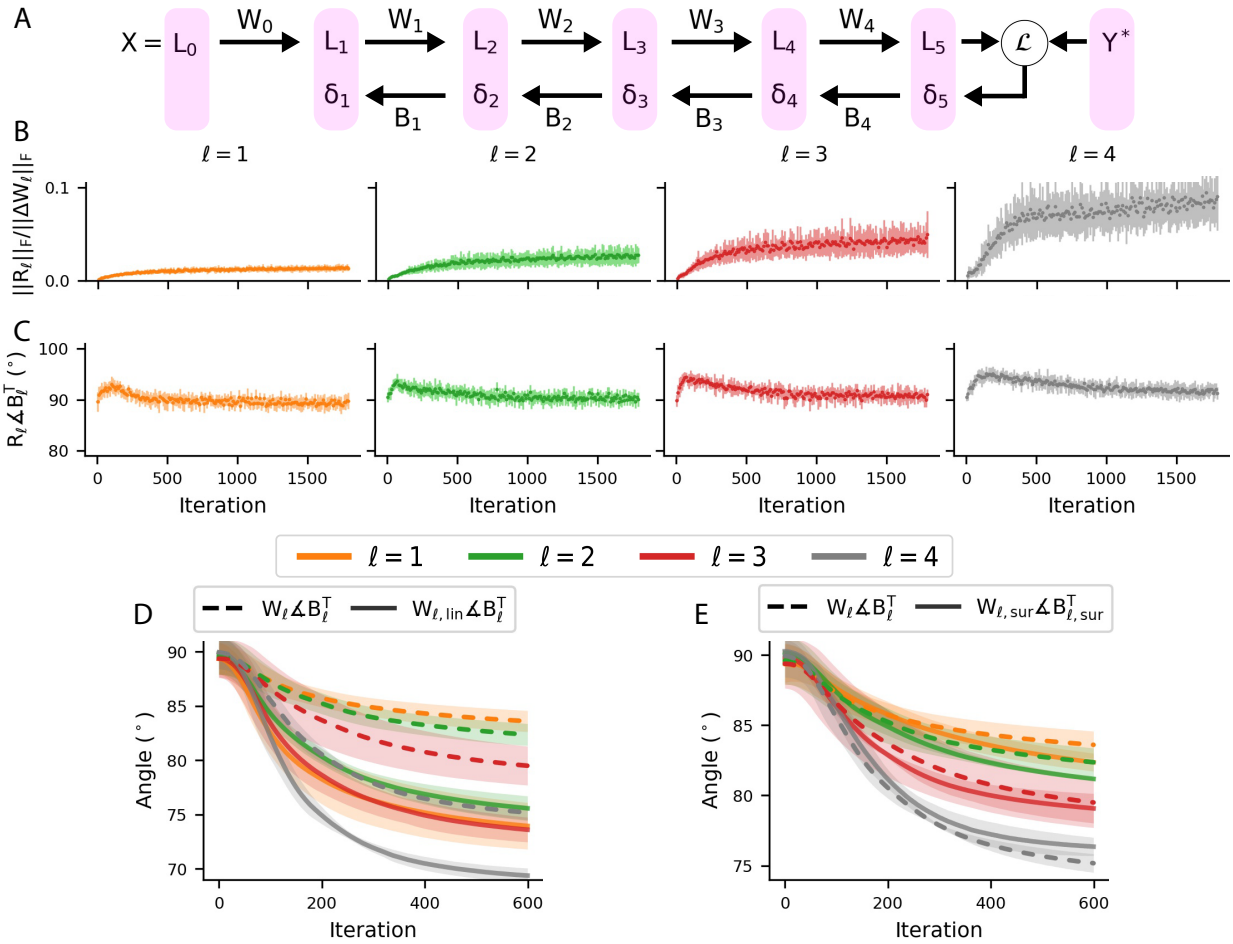

Figure 3 | **Effect of approximations and simplifying assumptions on alignment in the training of a nonlinear ANN on MNIST. (A)** Layout of the network under investigation. **(B,C)** To extract the alignment terms, first order Taylor approximation is used (equation 4). In this example network, the remainder terms (R_ℓ) constitute a small proportion of weight update directions. Moreover, they remain roughly orthogonal to the backward weights. Therefore, the remainder terms are not responsible for the occurrence of alignment and the first-order Taylor approximation provides a good approximation for the analysis of WA. **(D)** In the learning process of the network where its actual weight matrices (W_ℓ) are updated by the update rule of BP-TAW, $W_{\ell,lin}$ is updated by a different version of update rule of BP-TAW (equation 10) where for calculating the alignment terms, nonlinearity is ignored (equation 5 is used instead of 4), but other matrices involved are the same as the actual network. Comparing two cases shows that nonlinearity is a distortion on the alignment terms which reduces the amount of alignment but does not prevent it from occurring. Therefore, for simplification, nonlinearity can be ignored in the analysis of WA in this example network. **(E)** In the learning process of the network where its actual weight matrices (W_ℓ) are updated by the update rule of BP-TAW, $W_{\ell,sur}$ is updated by a different version of update rule of BP-TAW (equation 11) where for calculating alignment terms actual backward weights of the network are replaced with surrogate backward weights ($B_{\ell,sur}$), which are randomly generated, while other matrices involved are the same as the actual ones. The alignment between $B^T_{\ell,sur}$ and $W_{\ell,sur}$ shows that this replacement slightly affects the amount of alignment but does not prevent the alignment from occurring. Therefore, in this example network, dependence between B_ℓ and M^o_ℓ plays a minor role in WA. (B-E) Each dot or trace is the average over 30 runs and error bars and shaded areas are one s.d. around the mean.

2.2.1.2 Ignoring nonlinearity

To investigate the effect of nonlinearity on the alignment terms, in the learning process of the network by BP-TAW, in addition to the actual forward weights (W_ℓ), we updated a different version of the forward weights ($W_{\ell,lin}$) with a different version of update direction as follows

$$\Delta W_{\ell,lin}[k] = \sum_o T^o_{\ell,aln-lin}[k] + \eta f\left(\zeta^k_\ell[k]\right)^T \delta_{\ell+1,FA}[k] \tag{10}$$

where we ignored nonlinearity in the calculation of the alignment terms (using equation 5 instead of 4), while the error and output signals of the actual intact nonlinear network were still used for the calculation of the linear alignment terms ($W_{\ell,lin}$ was not used for the calculation of neural activity). In order to examine the impact of nonlinearity only on the alignment terms, we preserved the nonlinearity in the zero-degree Taylor polynomial (the rightmost term in the equation 10).

Comparing the alignment of the actual forward weights with the alignment of $\Delta W_{\ell,lin}$ shows that regardless of the amount of alignment, the overall dynamics of linear and alignment terms are similar (Fig. 3D). In other words, nonlinearity is a distortion on the alignment terms that reduces the amount of alignment but does not prevent alignment from occurring in this network.

2.2.1.3 Independence of the transformation matrix M^o_ℓ and backward weights B_ℓ

After the start of the learning process, both the output and error signals of neurons at each layer are a function of the backward weight matrices. However, as explained in the derivation of equation 7, if the statistical dependence between the

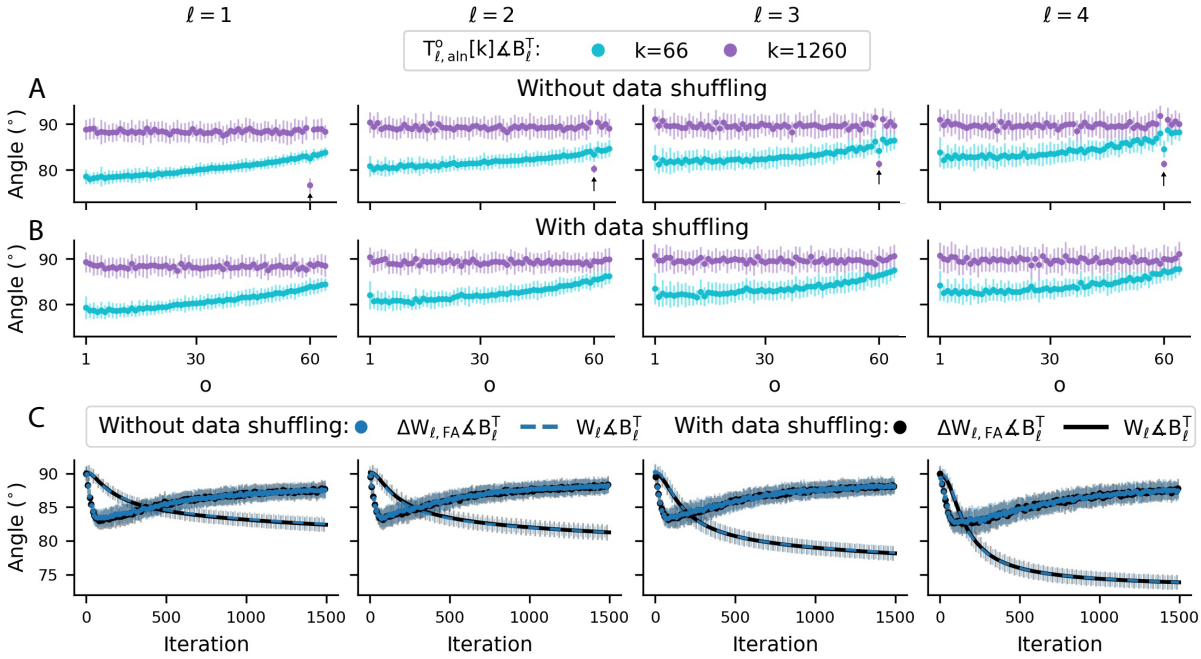

Figure 4 | Repetition of the same data points in each epoch contributes to alignment by shaping autocorrelated neural activity. (A) The training dataset is divided into 60 mini-batches which are the same in all epochs (no data shuffling). As the learning process proceeds, the alignment terms whose orders 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. **(B)** The training dataset is divided into 60 mini-batches and also is 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. **(C)** Data shuffling changes the behavior of alignment terms but not the total alignment since the autocorrelated activity of neurons that without data shuffling is concentrated in the lags that are a multiple of 60, with data shuffling becomes distributed among other lags. (A-C) Each dot or trace is the average over 30 runs and error bars are one s.d. around the mean.

transformation matrix M_ℓ^o and backward weights is weak, we can mainly attribute the occurrence of WA to the properties of the transformation matrix M_ℓ^o.

If the alignment is driven by the properties of M_ℓ^o rather than the relationship between M_ℓ^o and B_ℓ^T, $B_{\ell,sur}^T M_\ell^o$ still aligns with $B_{\ell,sur}^T$, where the actual backward weights used for the training (B_ℓ) are replaced with surrogate backward weights ($B_{\ell,sur}$) which are randomly generated from the same distribution of the actual backward weights. Surrogate backward weights have no contribution to the training of the network, and consequently they are independent of M_ℓ^o.

To examine this, in the learning process of the network by BP-TAW, in addition to the actual forward weights (W_ℓ), we updated a surrogate version of the forward weights ($W_{\ell,sur}$) with a surrogate version of update directions according to

$$\Delta W_{\ell,sur}[k] = \sum_o T_{\ell,aln-sur}^o[k] + \eta f\left(\zeta_\ell^k[k]\right)^T \delta_{\ell+1,FA}[k] \tag{11}$$

where the surrogate nonlinear alignment terms ($T_{\ell,aln-sur}^o$) were calculated according to the equation 4, except that the actual backward weights were replaced with surrogate backward weights, while other matrices involved in the surrogate alignment terms were the same as the matrices of the actual intact network ($W_{\ell,sur}$ was not used for the calculation of neural activity and $B_{\ell,sur}$ was constant in each run).

The comparison of alignment of the actual weights ($W_\ell \measuredangle B_\ell^T$) with the alignment of surrogate weights ($W_{\ell,sur} \measuredangle B_{\ell,sur}^T$) shows that a similar alignment, with a minor difference in amount, happens across layers in both situations (Fig. 3E). Thus, we can assume that alignment is largely driven by the properties of M_ℓ^o and that the statistical dependence between B_ℓ^T and M_ℓ^o (and also matrices involved because of the effect of nonlinearity) plays a minor role in WA in this network.

2.2.2 Factors affecting the dynamics of alignment terms and alignment

2.2.2.1 Autocorrelation of error and output signals of neurons contributes to alignment

Without data shuffling, in the initial phase of the learning process, all orders of alignment terms showed a considerable amount of alignment, but those whose orders were a multiple of 60 were slightly more aligned than their adjacent orders (Fig. 4A, Fig. S2A). 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. 4A, Fig. S2A).

The arrangement of data in the mini-batches effects the behavior of alignment terms. Since the dataset is divided into 60 mini-batches, without data shuffling, the same mini-batches are repeated every 60 iterations. As a result, neural activity is autocorrelated at lags that are a multiple of 60, causing the alignment terms whose orders are a multiple of 60 to behave differently. Referring back to the equation 5, an appropriate condition for 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

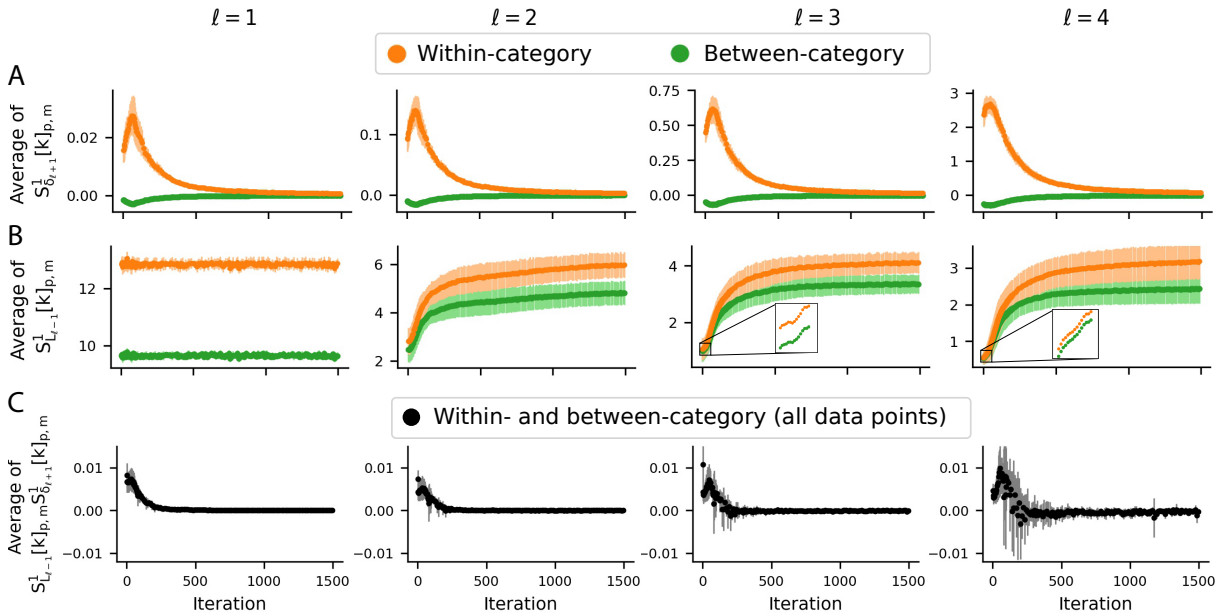

Figure 5 | Relative similarity of data points belonging to a single category compared to data points belonging to different categories contributes to alignment by shaping cross-correlated neural activity. (A) As a measure of similarity, $S^o_{\delta_{\ell+1}}[k]_{p,m}$ is the dot product between the error signals of the 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 also 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, this 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 (equation 12). 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 own category (see Fig. S3), 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.

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, that is, the output and error signals of its 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 alignment term of order 60 more aligned (Fig. 4A, Fig. S2A). 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 similar to the other orders of alignment terms and lose their initial amount of alignment with the continuation the learning process (Fig. 4B, Fig. S2B).

However, the amount of alignment of ΔW_ℓ with B^T_ℓ does not change with data shuffling (Fig. 4C). 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 that without data shuffling is concentrated in the lag of $o = 60$, with data shuffling becomes distributed among the lags of $o = 60$ up to $o = 119$ since 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^T_ℓ is influenced by the sum of ΔW_ℓ 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 the collective behavior of them and has no considerable effect on the amount of alignment (see Supplementary Note 2).

Data shuffling does not affect the overall behavior of the alignment terms whose orders are not a multiple of 60. The reason of their initial alignment and also the subsequent reduction in their amount of alignment will be discussed in the next section.

2.2.2.2 Cross-correlations of 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 in 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 the equation 8 and considering the aforementioned simplifying assumptions, alignment is expected if

$$0 < \mathbb{E}(\langle T^o_\ell[k], B^T_\ell \rangle_F) \approx \eta^2 n_\ell \sigma^2 \mathbb{E}\Big(tr(S^o_{\delta_{\ell+1}}[k]^T S^o_{L_{\ell-1}}[k])\Big) = \eta^2 n_\ell \sigma^2 \mathbb{E}\Big(\sum_{p,m} S^o_{\delta_{\ell+1}}[k]_{p,m} S^o_{L_{\ell-1}}[k]_{p,m}\Big) \tag{12}$$

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 of the layer $\ell-1$ produced by the pth data point of the

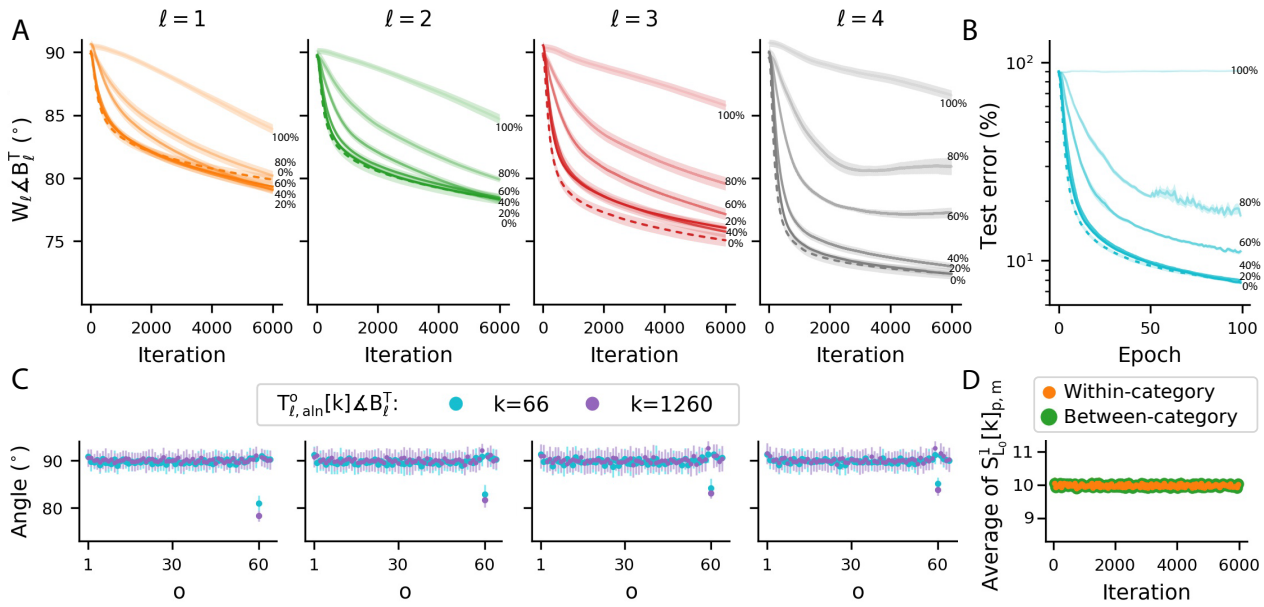

Figure 6 | Within-category similarity of data points enhances weight alignment, but certain amount of weight alignment can happen even using data with random labels. (A) WA across layers of the example network trained on the MNIST dataset where labels of different percentages of data points are shuffled (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, 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. 4A). **(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 makes the data points belonging to different categories as similar as the data points belonging to 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. 5B). (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.

$(k-o)^{th}$ 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 the layer $\ell+1$.

In the summation of equation 12, we define $S^o_{\delta_{\ell+1}}[k]_{p,m} S^o_{L_{\ell-1}}[k]_{p,m}$ as *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 is nonnegative, 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. 5A, Fig. S3A), which is constructive for alignment (referring to equation 12 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. 5A,Fig. S3A), which is destructive for alignment (referring to equation 12 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. 5B, Fig. S3B), 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 to have a relatively high mean compared to that of between-category similarity terms (Fig. 5B, Fig. S3B). 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. 5B, Fig. S3B).

Note that in the summation of the equation 12, 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. 5C).

After the initial phase, this cross-correlation becomes weaker, causing the alignment terms to lose their initial amount of alignment, except for the alignment terms whose orders are a multiple of 60 when there is no data shuffling (Fig. 4). The reason for this is that as the network learns to discriminate between different categories, the responses of neurons in the face of the majority of the data points become saturated, and error signals of them in response to them vanish (Fig. S3C), while their error signals in response to some other data points remain large (Fig. S3C). 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 error signals vanish (Fig. S3D,E). This weakens the cross-correlation between error and output signals of neurons, leading to the mentioned reduction in alignment.

To verify the role of within-category similarity in deriving the alignment, we ran a simulation where we randomly selected

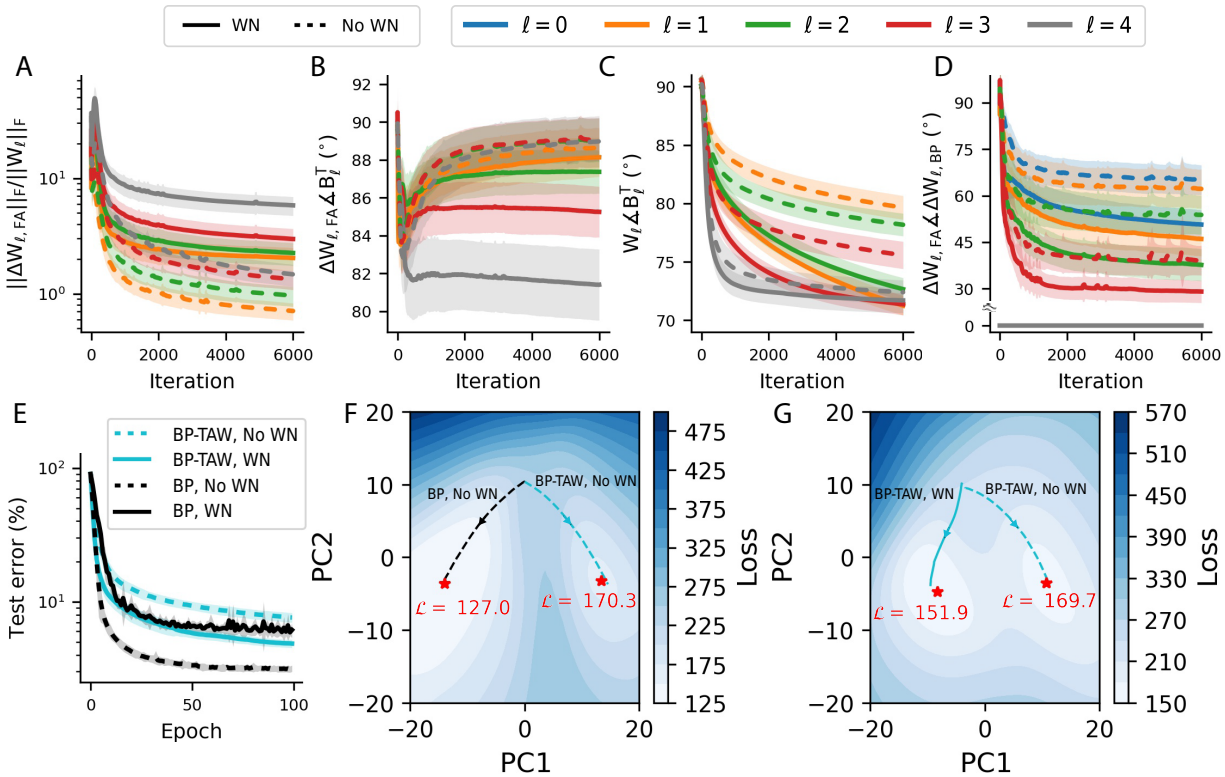

Figure 7 | Weight normalization in BP-TAW improves alignment and classification accuracy. (A) WN increases the Frobenius norms of update directions of BP-TAW relative to the Frobenius norms of forward weights. **(B)** WN improves the alignment of update directions of BP-TAW with backward weights. **(C)** WN improves WA. **(D)** WN improves the alignment between update directions of BP-TAW and those of BP. In the last layer, update direction of BP-TAW is the same as BP. **(E)** WN with the gain of $\gamma = 1$ can improve the test accuracy of BP-TAW whereas it makes the performance of BP less robust. **(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 the panel C 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.

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 to adversely effect the degree of WA (Fig. 6A). As expected, for high percentages of data points with randomly assigned labels, the total WA was degraded across all layers (Fig. 6A). Shuffling the labels also made it more difficult for the network to learn the classification task and increased the error rate (the test dataset was intact).

In the case where 100% of data were randomly labeled (samples of a given digit were equally likely to have one of the labels from 0 to 9) error rate reached 90%, representing the chance level (Fig. 6B). However, there was some residual and slow WA even when all labels were shuffled (Fig. 6A). The reason for the impact of label shuffling becomes clear by looking at the behavior of alignment terms across lags. In the case of shuffling 100% of the data labels and having no data shuffling at the beginning of epochs, all orders of alignment terms, except for the orders that are a multiple of 60, did not align considerably (Fig. 6C). Alignment terms whose orders were a multiple of 60 remained aligned because of the repetition of identical mini-batches every 60 iterations. 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. 6D), 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.

2.2.2.3 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. S4). Unlike what is previously thought (Lillicrap et al., 2016), WA is not a direct consequence of this growth, and it can also contribute to the weakening of alignment by saturating the outputs of neurons and vanishing of error signals. 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

$$W_\ell[k]_{*,i} \leftarrow \gamma \frac{W_\ell[k]_{*,i}}{\|(W_\ell[k])_{*,i}\|_F} \tag{13}$$

where $W_\ell[k]_{*,i}$ denotes an $n_\ell \times 1$ matrix that consists 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 (Salimans and Kingma, 2016), which is a reparameterization of 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. S5).

Two aspects of the network that directly and simultaneously affect the amount of alignment are $\Delta W_{\ell,FA}[k] \measuredangle B_\ell^T$ and $||\Delta W_{\ell,FA}[k]||_F / ||W_\ell[k]||_F$. This WN method improved both of these aspects (Fig. 7A,B) and consequently improved alignment of forward weights with backward weights (Fig. 7C) and also the alignment between update directions of BP-TAW and those of BP (Fig. 7D), 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. 7E).

The improvement of WA and the test accuracy is sensitive to γ and η. We did not optimized them and simply chose $\gamma = 1$ and $\eta = 0.0005$ based on a sensitivity analysis and grid search (Fig. S6), which showed that improvement of alignment and test accuracy of BP-TAW is robust for a fairly large range of γ. We also tried this WN method in training the network with BP which showed that among tested values of γ and η, the test accuracy of BP is not considerably improved by this WN method and declines in the most cases (Fig. S6).

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 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. 7F). WN drives the trajectory of parameters to a more optimal local minimum, resulting in a better degree of classification accuracy (Fig. 7G).

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. 7C). However, we observed that the amount of alignment between 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. 7D). This behavior has also been reported in previous works (Moskovitz et al., 2018).

The potential increase of $\Delta W_{\ell,FA} \measuredangle \Delta W_{\ell,BP}$ (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 which 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 the equation 2 successively, we have

$$\delta_{\ell,FA} = \delta_d B_{d-1} B_{d-2} \cdots B_{\ell+1} B_\ell \tag{14}$$

$$\delta_{\ell,BP} = \delta_d W_{d-1}^T W_{d-2}^T \cdots W_{\ell+1}^T W_\ell^T . \tag{15}$$

According to these two successive matrix multiplications of backward and 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 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, BP method needs a biologically implausible matching between feedforward and feedback synaptic weights. The BP-TAW learning method (Lillicrap et al., 2016) 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 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 due to the learning or reduction of loss function; rather, it relies on the structure of alignment terms that can be extracted from the update rule of BP-TAW (equation 3 and 4), and because of this structure, alignment happens robustly under a variety of conditions depending on the neural activity. Hence, statistical properties of neural activity contribute to the occurrence of alignment and its amount. 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).

In this work, we used alignment terms as a tool to analyze BP-TAW in a specific 5-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 individual alignment terms by shaping autocorrelated neural activity. Moreover, we showed that 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.

In the analysis of BP-TAW in the specific ANN above, we considered some approximations and simplifying assumptions. We used first-order Taylor approximation to extract alignment terms, ignored the effect of nonlinearity on them, and assumed their behavior to be independent of backward weights. These approximations and assumptions facilitated the analysis and enabled us to easily identify the main factors involved in alignment in a nonlinear ANN. Importantly, we showed that these assumptions and approximations led to qualitatively valid predictions about WA (Fig. 3).

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 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 dataset affect neural activity and consequently the behavior of alignment terms. However, we have provided a useful framework for investigating BP-TAW in the learning process of ANNs under various conditions which can be used to further our understanding of the FA and occurrence of WA in BP-TAW.

A weakness of BP-TAW as an approximation of BP in deep ANNs, which we mentioned in this work, is that the amount of alignment between update directions of BP-TAW and BP potentially tends to decline as the error is consecutively

backpropagated towards the earlier layers (Fig. 7C). 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 (Olshausen and Field, 1996; Barlow et al., 1961) and also there are suggestions of efficient network architectures where an ANN trained in an unsupervised manner is followed by a supervised classifier (Kheradpisheh et al., 2018).

3.2 Weight normalization can improve alignment

The correspondence of normalization methods in ANNs and biological ones has already been considered (Shen et al., 2020). In the context of BP-TAW, the utility of using normalization methods has been reported previously (Liao et al., 2016; Moskovitz et al., 2018). 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 (Chistiakova et al., 2015; Bi and Poo, 1998). Moreover, in the biological neurons, homeostatic mechanisms, which regulate activity of neurons and prevent them from having high or low firing rates, are well documented (Murthy et al., 2001; Surmeier and Foehring, 2004; Turrigiano, 2012).

Accordingly, we examined a WN method, which worked by fixing the Frobenius norm of input weights of each neuron to some constant (γ). We showed that this WN method can improve alignment and the classification accuracy of the network (Fig. 7). Indeed, 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, similar to our examined WN scheme (Chistiakova et al., 2015; Turrigiano, 2017).

3.3 Approximation of BP in biological networks and its further biological implausibilities

WA in the sense of reducing the angle between W_ℓ and B_ℓ^T is equivalent to having positively correlated reciprocal connections between neurons in the consecutive layers. Furthermore, in a sparse network where most of the elements of W_ℓ and B_ℓ are zero, achieving greater alignment requires reciprocal connections to occur more than nonreciprocal ones. Interestingly, such a nonrandom reciprocal connectivity, is reported among pyramidal cells (Song et al., 2005; Markram et al., 1997; Holmgren et al., 2003; Sjöström et al., 2001). Although these nonrandom features are reported in local cortical circuits, if we consider synaptic plasticity as their origin (Song et al., 2005) and generalize them to the connections between neurons in different cortical layers and areas, these features can provide a favorable condition for WA. In this regard, the emergence of dominant reciprocal connections by applying spike-timing-dependent synaptic plasticity rules has been reported in a recurrent ANN of spiking neurons under rate-coded input (Clopath et al., 2010).

However, weight transport problem is only one of the biological implausibilities of BP formula which can be avoided by using BP-TAW. There are other biological implausibilities in BP and BP-TAW (Marblestone et al., 2016; Stork, 1989). For instance, 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 (Stork, 1989; Song et al., 2020).

3.4 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

$$W_\ell[k+1] = W_\ell[k] + \Delta W_\ell[k] \tag{16}$$

$$\mathbf{b}_\ell[k+1] = \mathbf{b}_\ell[k] + \Delta \mathbf{b}_\ell[k] \tag{17}$$

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

$$\Delta W_{\ell,BP}[k] = -\eta \left.\frac{\partial \mathcal{L}}{\partial W_\ell}\right|_k = \eta L_\ell[k]^T \delta_{\ell+1,BP}[k],\ 0 \le \ell < d \tag{18}$$

$$\Delta \mathbf{b}_{\ell,BP}[k] = -\eta \left.\frac{\partial \mathcal{L}}{\partial \mathbf{b}_\ell}\right|_k = \eta J_{1\times n_b} \delta_{\ell,BP}[k],\ 0 < \ell \le d \tag{19}$$

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

$$\delta_{d,BP}[k] = E[k] \odot f'\big(Z_d[k]\big) \tag{20}$$

$$\delta_{\ell,BP}[k] = \delta_{\ell+1,BP}[k] W_\ell[k]^T \odot f'\big(Z_\ell[k]\big),\ 0 < \ell < d \tag{21}$$

where $E[k] = Y^*[k] - Y[k]$ according to the loss function $\mathcal{L}[k] = \frac{1}{2}\sum_{i,j} E[k]^2_{i,j}$ (Rumelhart et al., 1985).

In BP-TAW (Lillicrap et al., 2016), 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 update directions at each iteration as follows (W_ℓ^T in equation 21 is replaced with B_ℓ)

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

$$\delta_{\ell,FA}[k] = \delta_{\ell+1,FA}[k] B_\ell \odot f'\big(Z_\ell[k]\big),\ 0 < \ell < d \tag{23}$$

$$\Delta W_{\ell,FA}[k] = \eta L_\ell[k]^T \delta_{\ell+1,FA}[k],\ 0 \le \ell < d \tag{24}$$

$$\Delta \mathbf{b}_{\ell,FA}[k] = \eta J_{1\times n_b} \delta_{\ell,FA}[k],\ 0 < \ell \le d. \tag{25}$$

In training ANNs with BP-TAW, $\Delta W_{\ell,BP}[k]$ is a direction that we only calculated at each iteration for the purpose of 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

$$W \measuredangle B = cos^{-1}\Big(\frac{\langle W, B\rangle_F}{\|W\|_F \|B\|_F}\Big) \tag{26}$$

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

$$\text{cosine similarity}(W, B) = \frac{\langle W, B\rangle_F}{\|W\|_F \|B\|_F}. \tag{27}$$

Since the denominator of the cosine similarity is always nonnegative and also assuming W and B to be nonzero, for alignment ($W \measuredangle B < 90^\circ$ 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 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 $\mathcal{N}(0, 1)$. In all experiments of training a deep ANN on the MNIST dataset by BP-TAW (with and without WN and with and without data shuffling), we set $\eta = 0.0005$.

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 $c \in \{0, 1, \cdots, C-1\}$ 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.

5 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/FeedbackAlignmentWithWeightNormalization. We used PyTorch library only for accelerating computations on GPU (we did not use PyTorch's automatic differentiation capability).

6 Author Contributions

AR, AG and FM conceived the general ideas and the research plan. AR did the derivations and simulations in discussions with and under supervision of AG. AR and AG wrote the paper under FM supervision.

7 Competing interests

The authors declare that no competing interests exist.

References

Akrout, M., Wilson, C., Humphreys, P. C., Lillicrap, T., and Tweed, D. (2019). Deep learning without weight transport. *arXiv preprint arXiv:1904.05391*.

Baldi, P., Sadowski, P., and Lu, Z. (2018). Learning in the machine: Random backpropagation and the deep learning channel. *Artificial intelligence*, 260:1–35.

Barlow, H. B. et al. (1961). Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01).

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

Bi, G.-q. and Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of neuroscience*, 18(24):10464–10472.

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

Chistiakova, M., Bannon, N. M., Chen, J.-Y., Bazhenov, M., and Volgushev, M. (2015). Homeostatic role of heterosynaptic plasticity: models and experiments. *Frontiers in computational neuroscience*, 9:89.

Cichy, R. M., Khosla, A., Pantazis, D., Torralba, A., and Oliva, A. (2016). Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific reports*, 6(1):1–13.

Clopath, C., Büsing, L., Vasilaki, E., and Gerstner, W. (2010). Connectivity reflects coding: a model of voltage-based stdp with homeostasis. *Nature neuroscience*, 13(3):344.

Crafton, B., Parihar, A., Gebhardt, E., and Raychowdhury, A. (2019). Direct feedback alignment with sparse connections for local learning. *Frontiers in neuroscience*, 13:525.

Crick, F. (1989). The recent excitement about neural networks. *Nature*, 337(6203):129–132.

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

Grossberg, S. (1987). Competitive learning: From interactive activation to adaptive resonance. *Cognitive science*, 11(1):23–63.

Holmgren, C., Harkany, T., Svennenfors, B., and Zilberter, Y. (2003). Pyramidal cell communication within local networks in layer 2/3 of rat neocortex. *The Journal of physiology*, 551(1):139–153.

Khaligh-Razavi, S.-M. and Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain it cortical representation. *PLoS computational biology*, 10(11):e1003915.

Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J., and Masquelier, T. (2018). Stdp-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, 99:56–67.

Kolen, J. F. and Pollack, J. B. (1994). Backpropagation without weight transport. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, volume 3, pages 1375–1380. IEEE.

Kunin, D., Nayebi, A., Sagastuy-Brena, J., Ganguli, S., Bloom, J., and Yamins, D. (2020). Two routes to scalable credit assignment without weight symmetry. In *International Conference on Machine Learning*, pages 5511–5521. PMLR.

Launay, J., Poli, I., and Krzakala, F. (2019). Principled training of neural networks with direct feedback alignment. *arXiv preprint arXiv:1906.04554*.

Liao, Q., Leibo, J., and Poggio, T. (2016). How important is weight symmetry in backpropagation? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30.

Lillicrap, T. P., Cownden, D., Tweed, D. B., and Akerman, C. J. (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nature communications*, 7(1):1–10.

Lillicrap, T. P., Santoro, A., Marris, L., Akerman, C. J., and Hinton, G. (2020). Backpropagation and the brain. *Nature Reviews Neuroscience*, pages 1–12.

Marblestone, A. H., Wayne, G., and Kording, K. P. (2016). Toward an integration of deep learning and neuroscience. *Frontiers in computational neuroscience*, 10:94.

Markram, H., Lübke, J., Frotscher, M., Roth, A., and Sakmann, B. (1997). Physiology and anatomy of synaptic connections between thick tufted pyramidal neurones in the developing rat neocortex. *The Journal of physiology*, 500(2):409–440.

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

Murthy, V. N., Schikorski, T., Stevens, C. F., and Zhu, Y. (2001). Inactivity produces increases in neurotransmitter release and synapse size. *Neuron*, 32(4):673–682.

Nayebi, A., Bear, D., Kubilius, J., Kar, K., Ganguli, S., Sussillo, D., DiCarlo, J. J., and Yamins, D. L. (2018). Task-driven convolutional recurrent models of the visual system. In *Advances in Neural Information Processing Systems*, pages 5290–5301.

Nøkland, A. (2016). Direct feedback alignment provides learning in deep neural networks. *arXiv preprint arXiv:1609.01596*.

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

Refinetti, M., d'Ascoli, S., Ohana, R., and Goldt, S. (2020). The dynamics of learning with feedback alignment. *arXiv preprint arXiv:2011.12428*.

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

Salimans, T. and Kingma, D. P. (2016). Weight normalization: A simple reparameterization to accelerate training of deep neural networks. *arXiv preprint arXiv:1602.07868*.

Shen, Y., Wang, J., and Navlakha, S. (2020). A correspondence between normalization strategies in artificial and biological neural networks. *bioRxiv*.

Sjöström, P. J., Turrigiano, G. G., and Nelson, S. B. (2001). Rate, timing, and cooperativity jointly determine cortical synaptic plasticity. *Neuron*, 32(6):1149–1164.

Song, S., Sjöström, P. J., Reigl, M., Nelson, S., and Chklovskii, D. B. (2005). Highly nonrandom features of synaptic connectivity in local cortical circuits. *PLoS Biol*, 3(3):e68.

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

Stork, D. G. (1989). Is backpropagation biologically plausible. In *International Joint Conference on Neural Networks*, volume 2, pages 241–246. IEEE Washington, DC.

Surmeier, D. J. and Foehring, R. (2004). A mechanism for homeostatic plasticity. *Nature neuroscience*, 7(7):691–692.

Turrigiano, G. (2012). Homeostatic synaptic plasticity: local and global mechanisms for stabilizing neuronal function. *Cold Spring Harbor perspectives in biology*, 4(1):a005736.

Turrigiano, G. G. (2017). The dialectic of hebb and homeostasis. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 372(1715):20160258.

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

Whittington, J. C. and Bogacz, R. (2019). Theories of error back-propagation in the brain. *Trends in cognitive sciences*, 23(3):235–250.

Xiao, W., Chen, H., Liao, Q., and Poggio, T. (2018). Biologically-plausible learning algorithms can scale to large datasets. *arXiv preprint arXiv:1811.03567*.

Xie, X. and Seung, H. S. (2003). Equivalence of backpropagation and contrastive hebbian learning in a layered network. *Neural computation*, 15(2):441–454.

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

Supplementary Materials

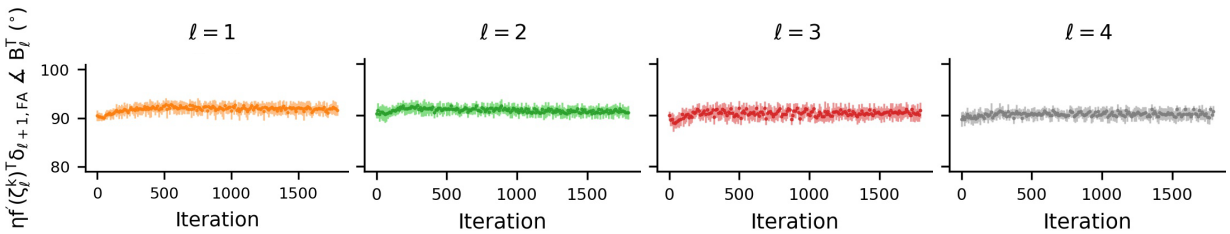

Figure S1 | **Zero-degree polynomial of Taylor approximation in the example network trained on MNIST. (A)** In addition to the alignment terms and the remainder terms, another term that appears in Taylor series expansion is $\eta f(\zeta_\ell^k[k])^T \delta_{\ell+1,FA}[k]$ which is the zero-degree Taylor polynomial of the last Taylor series expansion applied for extracting the alignment term of order $o = k$. This term does not have the beneficial structure of alignment terms and does not show considerable amount of alignment. Each dot is the average over 30 runs and error bars are one s.d. around the mean.

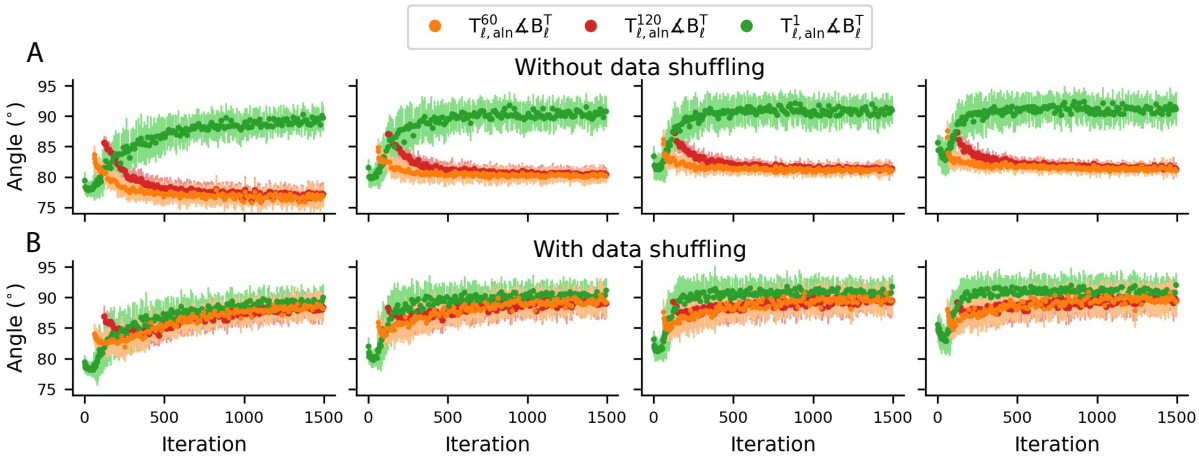

Figure S2 | **Behavior of alignment terms in the example network trained on MNIST. (A)** The behavior of alignment terms of orders 1, 60, and 120 across iterations without data shuffling. **(B)** The same as A but with data shuffling. (A,B) Each dot or trace is the average over 30 runs and error bars are one s.d. around the mean.

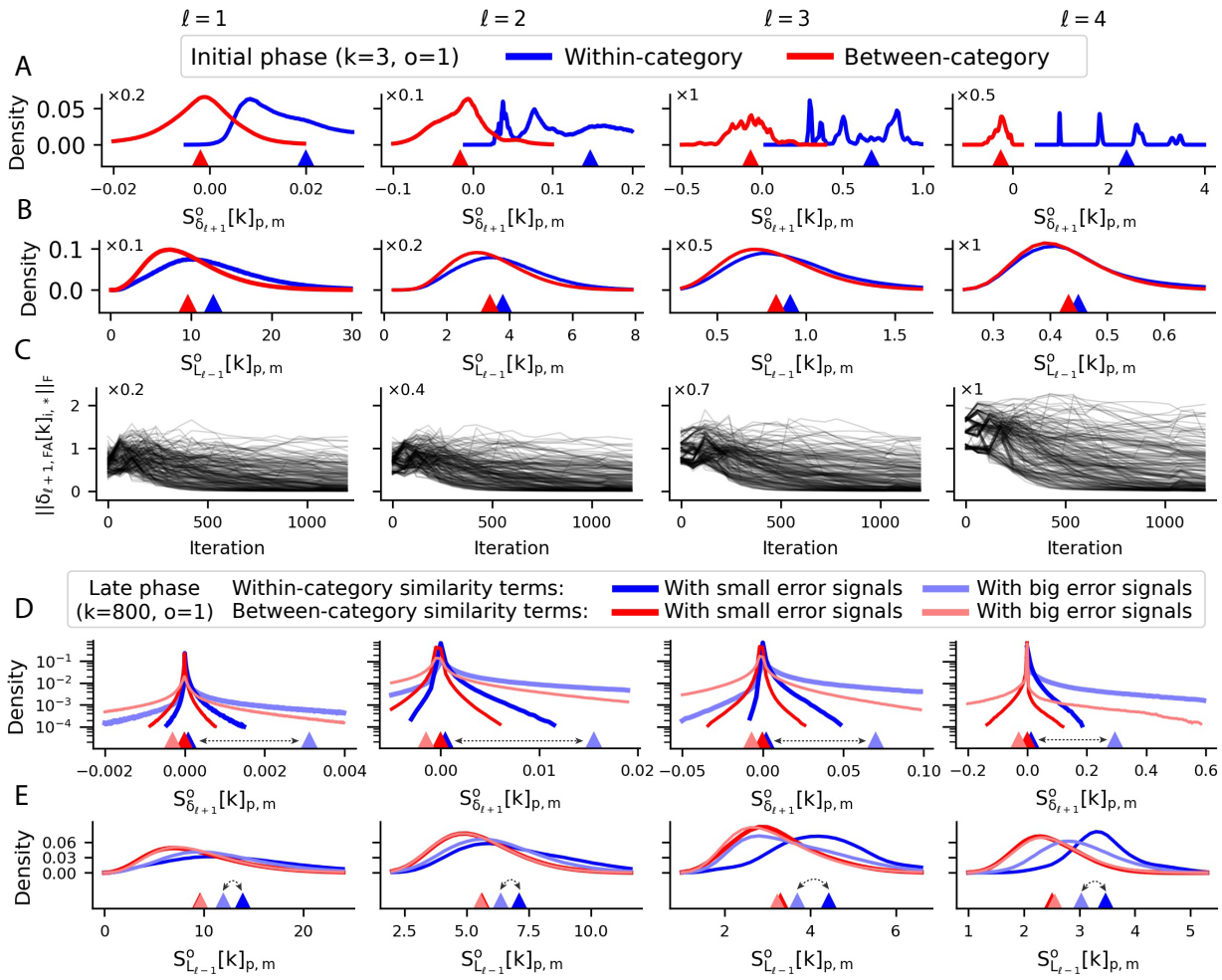

Figure S3 | Distributions of $S^1_{L_{\ell-1}}[k]_{p,m}$ and $S^1_{\delta_{\ell+1}}[k]_{p,m}$ in between- and within-category similarity terms in the early and late phases of the training in the example network trained on MNIST. (A,B) Histograms of $S^1_{L_{\ell-1}}[k]_{p,m}$ and $S^1_{\delta_{\ell+1}}[k]_{p,m}$ in the early phase of the network. Within-category $S^1_{L_{\ell-1}}[k]_{p,m}$ has a higher mean compared to that of between-category $S^1_{L_{\ell-1}}[k]_{p,m}$ since within-category data points are more similar to each other than between-category ones. This feature shapes cross-correlated neural activity which contributes to alignment. **(C)** $\|\delta_{\ell,FA}[k]_{i,*}\|_F$ denotes the Frobenius norm of the ith row of $\delta_{\ell,FA}[k]$ corresponding to i^{th} data point in the kth mini-batch. Each trace corresponds to the Frobenius norms of error signals of a single data point of the MNIST dataset in a single run. After the initial phase, outputs of neurons in response to the majority of the data points become saturated. Consequently, the Frobenius norm of their corresponding error signals decreases to about zero and vanishes. In contrast, error signals of some data points still remain large. **(D,E)** Histograms of $S^1_{L_{\ell-1}}[k]_{p,m}$ and $S^1_{\delta_{\ell+1}}[k]_{p,m}$ in the late phase of the network. In addition to categories, similarity terms are separated base on the magnitude of Frobenius norms of the error signals of the pth data point of the $(k-o)^{th}$ mini-batch and the mth data point of the kth mini-batch. If both of these two mentioned data points belong to a subset consisting of 60% of the total data points whose error signals at the output layer of the network have the least Frobenius norms, their similarity terms are regarded as "with small error signals", otherwise their corresponding similarity terms are regarded as "with big error signals". In the late phase of the network, on average, $S^1_{\delta_{\ell+1}}[k]_{p,m}$ in the similarity terms with small error signals has relatively small magnitude (absolute value) compared to $S^1_{\delta_{\ell+1}}[k]_{p,m}$ in similarity terms with big error signals (dark vs. light triangles). Thus, similarity terms with big error signals dominate the ones with small error signals in shaping the dynamics of the alignment terms in the late phase of the network. Within each category, in comparison to data points with small error signals, data points with big error signals (and also their representations across the layers of the network) are less similar to other data points of that category (dark vs. light blue triangles). This weakens the cross-correlation between $S^1_{L_{\ell-1}}[k]_{p,m}$ and $S^1_{\delta_{\ell+1}}[k]_{p,m}$ and causes $T^1_{\ell,aln}$ to lose its initial amount of alignment as the learning process proceeds. (A-E) Each histogram is obtained by giving the whole MNIST data points to the network. Triangles indicate the mean of each histogram. Histograms are plotted for $o = 1$, but a similar analysis holds for other orders of alignment terms.

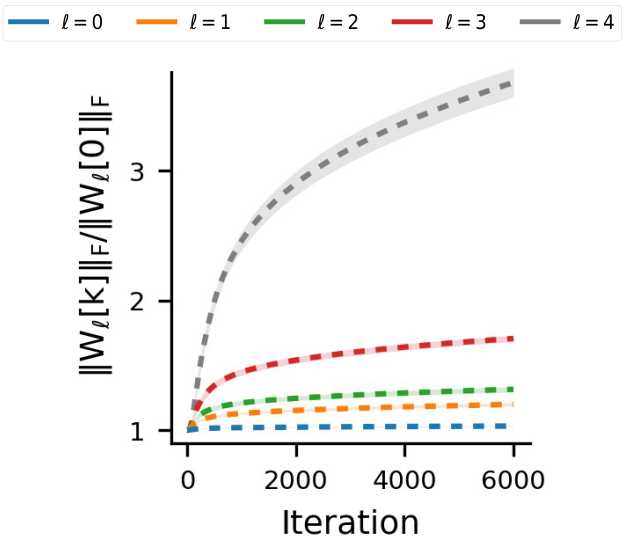

Figure S4 | Growth of Frobenius norms of forward weight matrices in the example network trained on MNIST. Without WN, Frobenius norms of forward weight matrices continuously grow. This growth is more pronounced at the last layers.

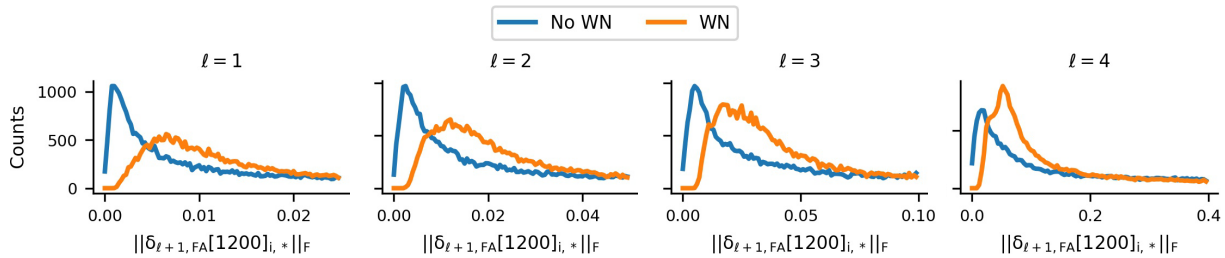

Figure S5 | WN partially prevents saturation of neurons and vanishing of error signals in the example network trained on MNIST. Histograms of Frobenius norms of error signals of all individual data points of the dataset with and without WN across all layers. $\delta_{\ell+1,FA}[k]_{i,*}$ denotes the ith row of $\delta_{\ell+1,FA}[k]$.

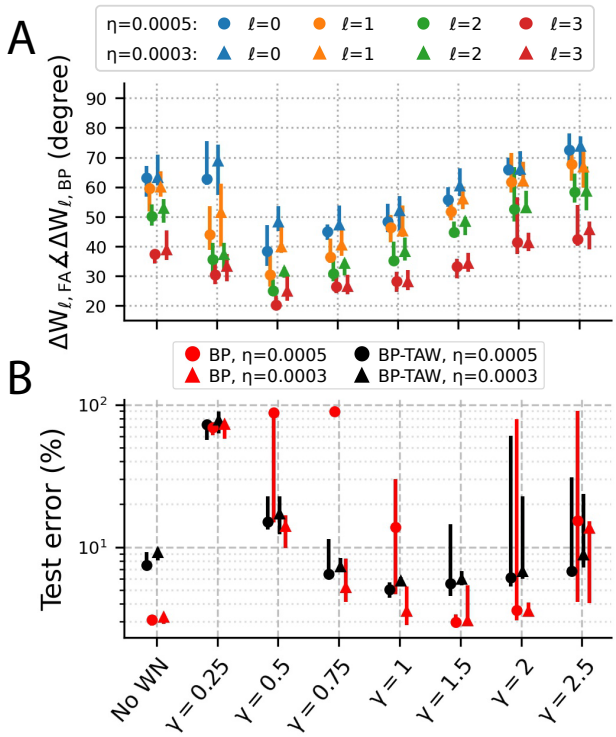

Figure S6 | Sensitivity analysis and grid search for WN gain (γ) and learning rate (η) in the training of the example network on MNIST. (A) The angle between update directions of BP and those of BP-TAW for different values of WN gain (γ) separately for two learning rates $\eta = 0.0003$ and $\eta = 0.0005$ across all layers. **(B)** Classification error for different values of WN gain (γ) for BP and BP-TAW separately for two learning rates $\eta = 0.0003$ and $\eta = 0.0005$. (A,B) Each point is the median of 10 runs and error bars show range of the minimum and the maximum values.

Supplementary Note 1| Taylor polynomials of higher degree

In the main text, we have obtained alignment terms by first order Taylor approximations. Therefore, each alignment term is the first-degree Taylor polynomial in an individual Taylor series. However, if more precision is required, and if the activation function is real and analytic, Taylor polynomials of higher degree can be expanded as follows:

$$\begin{aligned}
&\Delta W_{\ell,FA}[k] = \eta L_\ell[k]^T \delta_{\ell+1,FA}[k] \\
&= \eta f\big(W_{\ell-1}[k]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\delta_{\ell+1,FA}[k] \\
&= \eta f\big(\{W_{\ell-1}[k-1]^T + \eta\delta_{\ell,FA}[k-1]^T L_{\ell-1}[k-1]\}L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\delta_{\ell+1,FA}[k] \\
&= \eta\big\{ f\big(W_{\ell-1}[k-1]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big) \ + \\
&\quad \sum_{v=1}^{\infty} \frac{1}{v!} f^{(v)}\big(W_{\ell-1}[k-1]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big) \odot \Big(\eta\delta_{\ell,FA}[k-1]^T L_{\ell-1}[k-1] L_{\ell-1}[k]^T\Big)^{\circ v}\big\}\delta_{\ell+1,FA}[k]
\end{aligned} \tag{S1}$$

where $(\cdot)^{\circ v}$ denotes element-wise power of the matrix (also known as Hadamard power) and $f^{(v)}\big(\zeta_\ell^o[k']_{m,i}\big)$ denotes the n-th derivative of the real analytic activation function f at the point $\zeta_\ell^o[k']_{m,i}$. Analogously, we can expand $f\big(W_{\ell-1}[k-1]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)$ as follows

$$\begin{aligned}
&\Delta W_{\ell,FA}[k] = \eta f\big(W_{\ell-1}[k-2]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\delta_{\ell+1,FA}[k] \ + \\
&\eta \sum_{v=1}^{\infty} \big\{ \frac{1}{v!} f^{(v)}\big(W_{\ell-1}[k-2]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big) \odot \Big(\eta\delta_{\ell,FA}[k-2]^T L_{\ell-1}[k-2] L_{\ell-1}[k]^T\Big)^{\circ v}\big\}\delta_{\ell+1,FA}[k] + \\
&\eta \sum_{v=1}^{\infty} \big\{ \frac{1}{v!} f^{(v)}\big(W_{\ell-1}[k-1]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big) \odot \Big(\eta\delta_{\ell,FA}[k-1]^T L_{\ell-1}[k-1] L_{\ell-1}[k]^T\Big)^{\circ v}\big\}\delta_{\ell+1,FA}[k]
\end{aligned} \tag{S2}$$

Doing this successively yields

$$\begin{aligned}
&\Delta W_{\ell,FA}[k] = \eta f\big(W_{\ell-1}[0]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\delta_{\ell+1,FA}[k] \ + \\
&\eta \sum_{o=1}^{k}\sum_{v=1}^{\infty} \big\{ \frac{1}{v!} f^{(v)}\big(W_{\ell-1}[k-o]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big) \odot \Big(\eta\delta_{\ell,FA}[k-o]^T L_{\ell-1}[k-o] L_{\ell-1}[k]^T\Big)^{\circ v}\big\}\delta_{\ell+1,FA}[k]
\end{aligned} \tag{S3}$$

Taking $\zeta_\ell^o[k] = L_{\ell-1}[k]W_{\ell-1}[k-o] + \mathbf{b}_\ell[k]$ and $\delta_{\ell,FA}[k-o] = \delta_{\ell+1,FA}[k-o]B_\ell \odot f'\big(Z_\ell[k-o]\big)$, $\Delta W_{\ell,FA}[k]$ reads

$$\begin{aligned}
&\Delta W_{\ell,FA}[k] = \eta f\big(\zeta_\ell^k[k']^T\big)\delta_{\ell+1,FA}[k] \ + \\
&\eta \sum_{o=1}^{k}\sum_{v=1}^{\infty} \big\{ \frac{1}{v!} f^{(v)}\big(\zeta_\ell^o[k]^T\big) \odot \Big(\eta\{f'\big(Z_\ell[k-o]\big)^T \odot B_\ell^T \delta_{\ell+1,FA}[k-o]^T\} L_{\ell-1}[k-o] L_{\ell-1}[k]^T\Big)^{\circ v}\big\}\delta_{\ell+1,FA}[k]
\end{aligned} \tag{S4}$$

Supplementary Note 2| Index notation of alignment terms

Index notation of alignment terms is very useful to gain further understanding about the underlying mechanism of FA and WA. According to the matrix notation of alignment terms that we have used in the main text, index notation of alignment terms can be driven as follows

$$\begin{aligned}
&T_\ell^o[k]_{i,j} = \\
&\eta^2 \sum_{q,p,c,m} f'\big(\zeta_\ell^o[k]\big)_{m,i} f'\big(Z_\ell[k-o]\big)_{p,i} (B_\ell)_{q,i} \delta_{\ell+1,FA}[k-o]_{p,q} L_{\ell-1}[k-o]_{p,c} L_{\ell-1}[k]_{m,c} \delta_{\ell+1,FA}[k]_{m,j}
\end{aligned} \tag{S5}$$

where the element in the qth row and the ith column of a matrix is denoted by a subscript consisted of a pair of small letters (for example $(B_\ell)_{q,i}$ denotes the element in the qth row and the ith column of B_ℓ).

Beyond individual alignment terms, we are interested in the behavior of the summation of all orders of alignment terms which appears in $\Delta W_\ell[k] \approx T_\ell^1[k] + T_\ell^2[k] + \cdots$. For a sample $\Delta W_\ell[k] \approx T_\ell^1[k] + T_\ell^2[k] + \cdots$ and a sample B_ℓ^T, alignment in the sense of the injection of an aligned component into W_ℓ by the resultant of all orders of alignment terms at iteration k, namely,

$$B_\ell^T \measuredangle(\sum_o T_\ell^o[k]) < 90^\circ \tag{S6}$$

is equivalent to

$$0 < \frac{\langle B_\ell^T, \sum_o T_\ell^o[k]\rangle_F}{\|B_\ell^T\|_F \|\sum_o T_\ell^o[k]\|_F} \Leftrightarrow 0 < \langle B_\ell^T, \sum_o T_\ell^o[k]\rangle_F \tag{S7}$$

Therefore, the resultant of all orders of alignment terms at iteration k injects an aligned component into W_ℓ if and only if

$$\begin{aligned}
&0 < \langle \sum_o T_\ell^o[k], B_\ell^T\rangle_F = \\
&\eta^2 \sum_{o,i,j,q,p,c,m} f'\big(\zeta_\ell^o[k]\big)_{m,i} f'\big(Z_\ell[k-o]\big)_{p,i} (B_\ell)_{j,i} (B_\ell)_{q,i} \delta_{\ell+1,FA}[k-o]_{p,q} L_{\ell-1}[k-o]_{p,c} L_{\ell-1}[k]_{m,c} \delta_{\ell+1,FA}[k]_{m,j}
\end{aligned} \tag{S8}$$

where $\zeta_\ell^o[k] = L_{\ell-1}[k]W_{\ell-1}[k-o] + \mathbf{b}_\ell[k]$. Equation S8 shows how input feedback weights to the layer ℓ ($(B_\ell)_{j,i}$ and $(B_\ell)_{q,i}$), error signals in the layer $\ell+1$ ($\delta_{\ell+1,FA}[k-o]_{p,q}$ and $\delta_{\ell+1,FA}[k]_{m,j}$), output signals in the layer $\ell-1$ ($L_{\ell-1}[k-o]_{p,c}$ and $L_{\ell-1}[k]_{m,c}$), and nonlinearity ($f'\big(\zeta_\ell^o[k]\big)_{m,i}$ and $f'\big(Z_\ell[k-o]\big)_{p,i}$) are collectively contribute to alignment.

Considering the cosine similarity

$$\text{cosine similarity}(B_\ell^T, \sum_o T_\ell^o[k]\rangle_F) = \frac{\langle B_\ell^T, \sum_o T_\ell^o[k]\rangle_F}{\|B_\ell^T\|_F \|\sum_o T_\ell^o[k]\|_F} \tag{S9}$$

, since its denominator ($\|B_\ell^T\|_F \|\sum_o T_\ell^o[k]\|_F$) is nonnegative, if we want to only investigate occurrence of alignment regardless of its amount, we can simply refer to its numerator ($0 < \langle B_\ell^T, \sum_o T_\ell^o[k]\rangle_F$). However, if we want to investigate its amount, we should also consider its denominator, especially $\|\sum_o T_\ell^o[k]\|_F$ (we are less concerned about $\|B_\ell^T\|_F$ since in BP-TAW it is a fixed matrix across iterations).

The effect of nonlinearity as a distortion can be understood from the equation S8. If the activation function is an increasing function (which is a common choice in practical applications), nonlinearity scales individual terms in the summation of equation S8 with a nonnegative scalar and does not change their sign. In general, this can be a complex distortion which has some determinative and game-changing dependence with other signals. However, if this dependence is weak and nonlinearity does not have much contribution to the overall behavior of alignment terms, we can regard nonlinearity as a mild distortion and simply ignore it and refer to the linear case as follows

$$\begin{aligned} &0 < \langle \sum_o T_\ell^o[k], B_\ell^T\rangle_F = \\ &\eta^2 \sum_{o,i,j,q,p,c,m} (B_\ell)_{j,i}(B_\ell)_{q,i}\delta_{\ell+1,FA}[k-o]_{p,q} L_{\ell-1}[k-o]_{p,c} L_{\ell-1}[k]_{m,c}\delta_{\ell+1,FA}[k]_{m,j} \end{aligned} \tag{S10}$$

, otherwise we can still do our analysis by referring to the equation S8 and considering nonlinearity.

From the index notation of alignment terms, it can be understood that the process of BP-TAW sees mini-batches as the time sequence is folded in them. For example, shuffling data points within each mini-batch (not the whole data set as in the conventional data shuffling) is the same as changing the order of summations over p and m in the equations S8 and S5. It does not change the amount of $\langle \sum_o T_\ell^o[k], B_\ell^T\rangle_F$ and also $\|\sum_o T_\ell^o[k]\|_F$; thus, it does not change the amount of alignment.

From the index notation of alignment terms, robustness of WA against data shuffling (shuffling the whole dataset at the beginning of each epoch) can be understood. Although data shuffling changes the trajectory of weight matrices, assuming the update steps to be small, in terms of statistical properties contributing to WA, shuffling is approximately similar to changing the order of summations over o and p in the equations S8 and S5. For example, from the point of view of the kth mini-batch, a data point of the dataset that without data shuffling has appeared at the second data point ($p = 2$) of the $(k-20)^{th}$ mini-batch ($o = 20$), after data shuffling may appear at the tenth data point of the $(k-5)^{th}$ mini-batch ($o = 5$ and $p = 10$). If neural activities produced in the network by this data point in both of these positions are similar to each other, changing its position among lags can change the behavior of each alignment term of order $o = 5$ and $o = 20$ but does not change the behavior of their summation. Data shuffling does the same to all data points, but if the mentioned condition about the similarity of activity is approximately met, it approximately maintains the amount of $\langle \sum_o T_\ell^o[k], B_\ell^T\rangle_F$ and also $\|\sum_o T_\ell^o[k]\|_F$ and consequently the amount of alignment without data shuffling is approximately equal to the amount of alignment with data shuffling.

The contribution of autocorrelation of neural activity to the alignment can be understood from the index notation of alignment terms. For simplicity, consider the linear case with the assumption that three random vectors of $[(B_\ell)_{j,i}, (B_\ell)_{q,i}]$, $[\delta_{\ell+1,FA}[k-o]_{p,q}, \delta_{\ell+1,FA}[k]_{m,j}]$, and $[L_{\ell-1}[k-o]_{p,n}, L_{\ell-1}[k]_{m,n}]$ are mutually independent, and also elements of B_ℓ are i.i.d. from $\mathcal{N}(0,\sigma^2)$ (similar to the hypothetical condition of Fig. 2C corresponding to the blue trace). In this condition, the occurrence of alignment is expected if

$$\begin{aligned} &0 < \mathbb{E}(\langle \sum_o T_\ell^o[k], B_\ell^T\rangle_F) = \\ &\eta^2 \sum_{o,i,j,q,p,c,m} \mathbb{E}((B_\ell)_{j,i}(B_\ell)_{q,i})\mathbb{E}(\delta_{\ell+1,FA}[k-o]_{p,q}\delta_{\ell+1,FA}[k]_{m,j})\mathbb{E}(L_{\ell-1}[k-o]_{p,c}L_{\ell-1}[k]_{m,c}) = \\ &\eta^2 \sum_{o,i,j,p,c,m} \mathbb{E}((B_\ell)_{j,i}^2)\mathbb{E}(\delta_{\ell+1,FA}[k-o]_{p,j}\delta_{\ell+1,FA}[k]_{m,j})\mathbb{E}(L_{\ell-1}[k-o]_{p,c}L_{\ell-1}[k]_{m,c}). \end{aligned} \tag{S11}$$

Accordingly, with the activation function chosen to be nonnegative (analogous to firing rate of biological neurons), alignment happens if the error signals ($\delta_{\ell+1,FA}[k-o]_{p,j}$) and the outputs of neurons ($L_{\ell-1}[k]_{m,c}$) are positively autocorrelated among lags (summation on o) or mini-batches (summation on m and p), since as previously mentioned, the process of BP-TAW sees mini-batches as the time sequence folded in them. In other words, alignment happens if elements of L_0 and δ_2 are positively autocorrelated in the sense that

$$\mathbb{E}(\delta_{\ell+1,FA}[k-o]_{p,j}\delta_{\ell+1,FA}[k]_{m,j}) > 0,\ \mathbb{E}(L_{\ell-1}[k-o]_{p,c}L_{\ell-1}[k]_{m,c}) > 0$$

which is in accordance with the standard definition of autocorrelation function of a stochastic process. Note that in this hypothetical condition, which is proposed to provide intuition about WA, error and input signals are mutually independent. In general, these conditions are not necessarily met and in addition to autocorrelation, other properties of neural activity, such as cross-correlation between error and output signals, contribute to alignment.

The contribution of cross-correlation between error and output signals of neurons to the alignment can be understood from the index notation of alignment terms. For simplicity, consider the linear case with the assumption that three random

vectors of $[(B_\ell)_{j,i}, (B_\ell)_{q,i}]$, $[\delta_{\ell+1,FA}[k-o]_{p,q}, L_{\ell-1}[k-o]_{p,n}]$, and $[\delta_{\ell+1,FA}[k]_{m,j}, L_{\ell-1}[k]_{m,n}]$ are mutually independent, and elements of B_ℓ are i.i.d. from $\mathcal{N}(0,\sigma^2)$, and output and error signals are unautocorrelated in the sense that

$$\mathbb{E}(\delta_{\ell+1,FA}[k-o]_{p,j}\delta_{\ell+1,FA}[k]_{m,j}) = 0,\ \mathbb{E}(L_{\ell-1}[k-o]_{p,c}L_{\ell-1}[k]_{m,c}) = 0$$

for $o \neq 0$ (similar to the hypothetical condition of Fig. 2C corresponding to the green trace). In this condition the occurrence of alignment is expected if

$$\begin{aligned}
&0 < \mathbb{E}(\langle \sum_o T_\ell^o[k], B_\ell^T \rangle_F) = \\
&\eta^2 \sum_{o,i,j,q,p,c,m} \mathbb{E}((B_\ell)_{j,i}(B_\ell)_{q,i})\mathbb{E}(\delta_{\ell+1,FA}[k-o]_{p,q}L_{\ell-1}[k-o]_{p,c})\mathbb{E}(\delta_{\ell+1,FA}[k]_{m,j}L_{\ell-1}[k]_{m,c}) = \\
&\eta^2 \sum_{o,i,j,p,c,m} \mathbb{E}((B_\ell)_{j,i}^2)\mathbb{E}(\delta_{\ell+1,FA}[k-o]_{p,j}L_{\ell-1}[k-o]_{p,c})\mathbb{E}(\delta_{\ell+1,FA}[k]_{m,j}L_{\ell-1}[k]_{m,c}).
\end{aligned} \tag{S12}$$

In other words, alignment happens if elements of L_0 and δ_2 are positively cross-correlated in the sense that

$$\mathbb{E}(\delta_{\ell+1,FA}[k-o]_{p,j}L_{\ell-1}[k-o]_{p,c}) > 0,\ \mathbb{E}(\delta_{\ell+1,FA}[k]_{m,j}L_{\ell-1}[k]_{m,c}) > 0$$

which in accordance with the standard definition of cross-correlation function of two stochastic processes. Note that in this hypothetical condition, which is proposed to provide intuition about WA, samples of error and output signals are independent among lags. In general, these conditions are not necessarily met.

All simplifying assumptions in this work are made to facilitate the analysis and explain the occurrence of alignment, whose underlying mechanism had remained elusive, in a simple and intuitive way. These assumptions do not weaken the generality of the provided framework because one can always refer to the nonlinear form of alignment terms (or even higher-order Taylor approximation) and also consider dependencies between backward weights and other signals, leading to a more elaborate analysis at the expense of simplicity.

Supplementary Note 3| Skew-symmetric part of the transformation matrix totally deviates the direction

The skew-symmetric part of the transformation matrix (or any real skew-symmetric matrix) totally deviates B^T (or any real matrix) after matrix multiplication since

$$\begin{aligned}
\langle B^T, B^T M_{skew}^o[k] \rangle_F &= tr(B^T M_{skew}^o[k] B) = tr(B^T M_{skew}^o[k]^T B) \\
&= -tr(B^T M_{skew}^o[k] B) = 0
\end{aligned} \tag{S13}$$

where $tr(\cdot)$ denotes matrix trace, and as a result $B^T \measuredangle B^T M_{skew}^o[k] = 90°$. Hence, the amount of alignment is dependent on $B_\ell^T \measuredangle B_\ell^T M_{\ell,sym}^o[k]$ and the ratio of $\|B_\ell^T M_{\ell,sym}^o[k]\|_F$ to $\|B_\ell^T M_{\ell,skew}^o[k]\|_F$.

Supplementary Note 4| Direct feedback alignment

In DFA (Nøkland, 2016), error signals are directly backpropagated from the output layer to each hidden layer through direct fixed random backward weights denoted by $F_\ell \in \mathbb{R}^{n_d \times n_\ell}$ as follows

$$\delta_{\ell,DFA}[k] = \begin{cases} \delta_{d,DFA}[k] F_\ell \odot f'\big(Z_\ell[k]\big) & 0 < \ell < d \\ -\eta \left.\frac{\partial \mathcal{L}}{\partial Z_d}\right|_k & \ell = d. \end{cases} \tag{S14}$$

For the update directions of the backpropagation through direct random weights we can write

$$\begin{aligned}
&\Delta W_{\ell,DFA}[k] = \eta L_\ell[k]^T \delta_{\ell+1,DFA}[k] \\
&= \eta f\big(L_{\ell-1}[k]W_{\ell-1}[k] + \mathbf{b}_\ell[k]\big)^T \delta_{\ell+1,DFA}[k] \\
&= \eta f\big(L_{\ell-1}[k]\{W_{\ell-1}[k-1] + \eta L_{\ell-1}[k-1]^T \delta_{\ell,DFA}[k-1]\} + \mathbf{b}_\ell[k]\big)^T \delta_{\ell+1,DFA}[k] \\
&\approx \eta f\big(L_{\ell-1}[k]W_{\ell-1}[k-1] + \mathbf{b}_\ell[k]\big)^T \delta_{\ell+1,DFA}[k] + \\
&\{\eta f'\big(L_{\ell-1}[k]W_{\ell-1}[k-1] + \mathbf{b}_\ell[k]\big)^T \odot \eta \delta_{\ell,DFA}[k-1]^T L_{\ell-1}[k-1] L_{\ell-1}[k]^T\} \delta_{\ell+1,DFA}[k] \\
&\approx T_{\ell,daln}^1[k] + T_{\ell,daln}^2[k] + \cdots + \eta f\big(L_{\ell-1}[k]W_{\ell-1}[0] + \mathbf{b}_\ell[k]\big)^T \delta_{\ell+1,DFA}[k]
\end{aligned} \tag{S15}$$

where we define *direct alignment term* of order o as bellow

$$T_{\ell,daln}^o = \{\eta f'\big(L_{\ell-1}[k]W_{\ell-1}[k-o] + \mathbf{b}_\ell[k]\big)^T \odot \eta \delta_{\ell,DFA}[k-o]^T L_{\ell-1}[k-o] L_{\ell-1}[k]^T\} \delta_{\ell+1,DFA}[k] \tag{S16}$$

We can start from $\ell = d-1$ towards $\ell = 1$ and decompose every $\hat{F}_\ell$ such that for $\ell = d-1$ we have,

$$F_{d-1} = \hat{F}_{d-1} = Q_{d-1} \tag{S17}$$

, and for $0 \leq \ell < d-1$ we have

$$\hat{F}_\ell = Q_{d-1}Q_{d-2}\cdots Q_{\ell+1}Q_\ell = \hat{F}_{\ell+1}Q_\ell = F_\ell - F_\ell^\perp \tag{S18}$$

where Q_ℓ is the unique least squares solution of $\arg\min_{Q_\ell}\|\hat{F}_{\ell+1}Q_\ell - F_\ell\|_F$ with minimum possible $\|Q_d\|_F$ and $F_\ell^\perp$ is perpendicular to $\hat{F}_\ell$ ($F_\ell^\perp \measuredangle \hat{F}_\ell = 90°$). According to this decomposition, we can write

$$\begin{aligned}
&T^o_{\ell,daln} = \{\eta f'\big(L_{\ell-1}[k]W_{\ell-1}[k-o] + \mathbf{b}_\ell[k]\big)^T \odot \\
&\eta\{f'\big(Z_\ell[k-o]\big)^T \odot F_\ell^T \delta_{d,DFA}[k-o]^T\}L_{\ell-1}[k-o]L_{\ell-1}[k]^T\}\{\delta_{d,DFA}[k]F_{\ell+1}[k] \odot f'\big(Z_{\ell+1}[k]\big)\} \\
&= \{\eta f'\big(L_{\ell-1}[k]W_{\ell-1}[k-o] + \mathbf{b}_\ell[k]\big)^T \odot \\
&\eta\{f'\big(Z_\ell[k-o]\big)^T \odot \{F_\ell^{\perp T} + Q_\ell^T \hat{F}_{\ell+1}^T\}\delta_{d,DFA}[k-o]^T\}L_{\ell-1}[k-o]L_{\ell-1}[k]^T\}\{\delta_{d,DFA}[k]\{\hat{F}_{\ell+1} + F_{\ell+1}^\perp\} \odot f'\big(Z_{\ell+1}[k]\big)\}
\end{aligned} \tag{S19}$$

In the linear case, and by assuming $F_\ell^\perp$ to be negligible ($\|F_\ell^\perp\|_F \approx 0$ and $F_\ell \approx \hat{F}_\ell$, this can be a good assumption depending on the structure and dimension of direct backward weight matrices and the column space of $\hat{F}_{\ell+1}$), direct alignment term of order o is

$$T^o_{\ell,daln} \approx \eta^2 Q_\ell^T F_{\ell+1}^T \delta_{d,DFA}[k-o]^T L_{\ell-1}[k-o]L_{\ell-1}[k]^T \delta_{d,DFA}[k]F_{\ell+1}. \tag{S20}$$

According to the structure of direct alignment terms in this simplified condition and with an analysis similar to the analysis of alignment terms in FA, the alignment of each W_ℓ with Q_ℓ^T and consequently the alignment of $F_\ell^T = Q_\ell^T Q_{\ell+1}^T \cdots Q_{d-2}^T Q_{d-1}^T$ with $W_\ell W_{\ell+1}\cdots W_{d-2}W_{d-1}$ can be understood. We leave further investigation of DFA and its general condition to future works.