The underlying mechanism of feedback alignment in error backpropagation through random synaptic feedback weights

Alireza Rahmansetayesh[1], Ali Ghazizadeh[1,2,*], Farokh Marvasti[1]

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

[2]School of Cognitive Sciences, Institute for Research in Fundamental Sciences, Tehran, Iran

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

Abstract

The mechanism by which plasticity in millions of synapses in the brain is orchestrated to achieve behavioral and cognitive goals is a fundamental question in neuroscience. In this regard, insights from learning methods in artificial neural networks (ANNs) and in particular the idea of 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 Random backward Weights (BP-TRW), can lead to partial alignment of forward and backward weights (feedback alignment or FA) and result in surprisingly good accuracies in simple classification tasks using shallow ANNs. However, despite some preliminary examinations, the mathematical underpinnings of FA is not well understood. In this work, we reveal its underlying mechanism and show that FA is governed by neural activity and its statistical properties like cross-correlation and autocorrelation of error and output signals of neurons. We show the feature that data points of the datasets belonging to a single category are more similar to each other than the ones belonging to different categories, which is an intrinsic feature of datasets, contributes to alignment by shaping cross-correlated neural activity. Furthermore, we show that FA can be improved significantly by limiting Frobenius norm of input weights of each neuron. Altogether, our analyses and results show the mathematical basis of FA with minimal assumptions on network.

Keywords— feedback alignment, weight transport problem, bio-inspired artificial neural networks, bio-inspired learning methods, biologically-inspired 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 for backward weights to precisely match the forward weights so that accurate error signals are backpropagated to the early layer for efficient learning as stipulated by BP. 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 for credit assignment in the brain is 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 by BP-TRW (Lillicrap et al., 2016), where backward weights are fixed, random and distinct from forward ones. During the learning process using BP-TRW, the angle between backward and transpose of forward weight matrices in each layer reduces and this partial alignment leads to calculation of an approximate gradient direction (Lillicrap et al., 2016). It is shown that learning can occur even when errors are passed directly from the output layer to each hidden layer through random backward weights which is known as direct feedback alignment (DFA) (Nøkland, 2016; Refinetti et al., 2020; Frenkel et al., 2019; Launay et al., 2019; Baldi et al., 2018). However, in deep and convolutional ANNs, the performance of FA-based learning methods drops compared to BP (Bartunov et al., 2018; Launay et al., 2019; Crafton et al., 2019; Moskovitz et al., 2018). Although there are some investigations on the dynamics of learning with feedback alignment and favorable conditions for improvement of FA-based learning methods (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 of FA is not fully understood yet.

Previous works have provided some preliminary proofs for alignment of forward and backward weights in special cases and linear networks (Lillicrap et al., 2016; Frenkel et al., 2019; Refinetti et al., 2020). For example, it has been shown that if forward weight matrices are initialized with small elements (values close to zero) and the input and desired output of network are kept constant during iterations, backward weight matrices became scalar multiple of the Moore-Penrose pseudo-inverse of forward weight matrices (Lillicrap et al., 2016), or scalar multiple of the Moore-Penrose pseudo-inverse of the product of forward matrices in the case of direct feedback alignment (Frenkel et al., 2019). Under these circumstances update direction of BP-TRW are an approximation of the Gauss-Newton optimization method (Lillicrap et al., 2016). However, these proofs cannot explain the occurrence of FA for arbitrary weight matrix initialization and nonlinear networks.

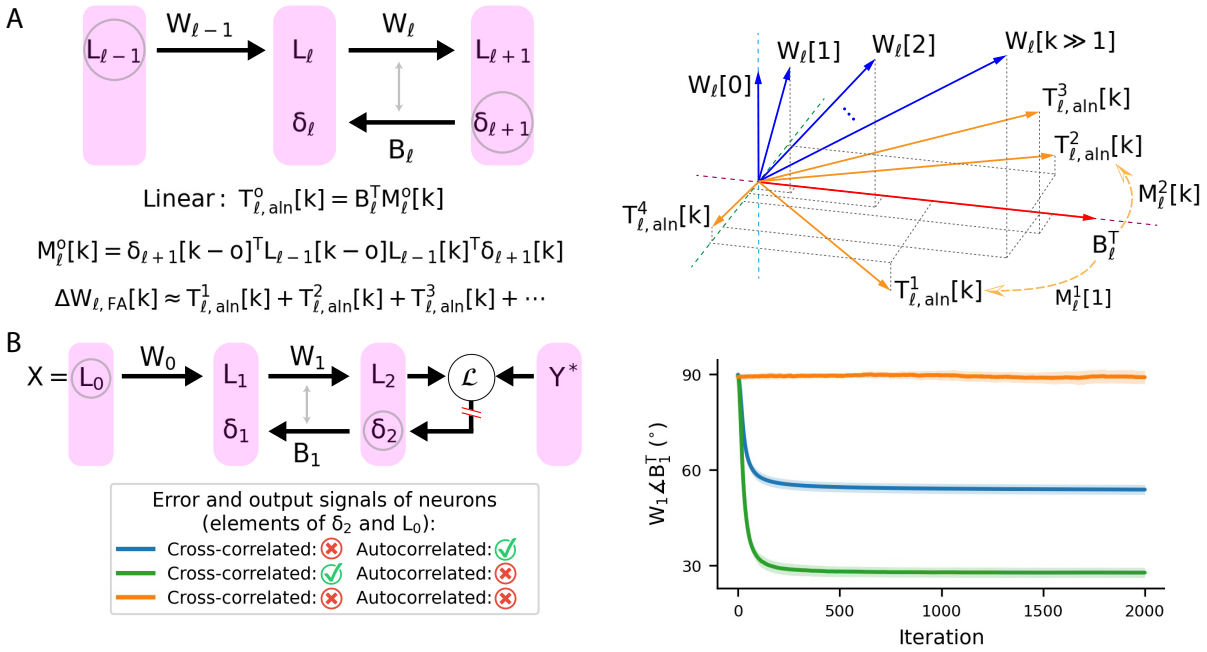

Figure 1 | **Underlying mechanism of feedback alignment: alignment is governed by neural activity. (A)** Expansion of $\Delta W_{\ell,FA}[k]$ reveals alignment terms ($T^o_{\ell,aln}[k]$, equation 3, and 4) which have a favorable structure. Due to this structure, alignment terms can robustly propel forward weight matrices (W_ℓ) towards transpose of fixed random backward weight matrices (B_ℓ^T) under a variety of conditions depending on neural activity. In each alignment term, $M^o_\ell[k]$ acts as a transformation matrix on B_ℓ^T and if it partially preserves the direction of B_ℓ^T after matrix multiplication, it propels $W_\ell[k]$ towards $B_\ell^T[k]$. $L_\ell[k]$ 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. Note that this is a simplified diagram of underlying mechanism of FA. 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_ℓ^T. We have assigned an order o ($1 \leq o \leq k$) to each of them since in each alignment term of order o, activity of neurons in iteration k and $k-o$ (lag of o) are related together. **(B)** In an open-loop nonlinear 2-layer ANN, hypothetical signals are imposed on outputs of neurons in input layer ($X = L_0$) and errors of neurons in output layer (δ_2). (dashed blue trace) Elements of δ_2 and L_0 are independently initialized with $\mathcal{N}(0,1)$ and left constant during iterations. In this case, their elements are autocorrelated and alignment occurs. (dotted blue trace) Elements of $\delta_2 = L_0$ are independently initialized with $\mathcal{N}(0,1)$ and also reinitialized at each iteration. In this case, elements of δ_2 and L_0 are unautocorrelated yet cross-correlated and alignment occurs. (solid orange trace) Elements of δ_2 and L_0 are independently initialized with $\mathcal{N}(0,1)$ and they are reinitialized at every iteration. In this case, error and output signals of neurons (elements of L_0 and δ_2) are neither autocorrelated nor cross-correlated and alignment does not occur. Each trace is average over 10 runs and shaded areas are one s.d. around the mean.

In addition, Lillicrap et al. 2016 have provided a preliminary insight into the mechanics of alignment by freezing forward weights in different stages of the learning process of an ANN trained by BP-TRW, 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). But the underlying mathematical basis of this information flow has not yet been demonstrated. It has been 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 later show that although FA is accompanied by the continuous growth of norm of weight matrices, this is not the underlying mechanism of FA. Quite contrary, we show that limiting norm of weights can even improve alignment.

In this work, we first explore the mathematical basis of FA in the general case. We show that occurrence of alignment does not rely on reduction of loss function; rather, it is governed by statistical properties of neural activity like cross-correlation and autocorrelation of error and output signals of neurons. Afterwards, based on the demonstrated mathematical basis, we dissect FA in a practical application of BP-TRW for training a deep ANN on MNIST dataset. We show that consistency between input data points and their labels and relative similarity of within categories data points compared to between categories ones, which is an intrinsic statistical property of datasets, contribute to alignment by shaping cross-correlated neural activity. By principle component analysis, we will compare the trajectories of all learnable parameters of the network trained by BP and BP-TRW and show that by BP-TRW the network convergences to a local minimum totally different from the local minimum to which the network convergences with BP. Finally, we will shown that restricting the weight normalization results in better FA and further reduction of test error compared to BP-TRW with nonnormalized weights.

2 Results

2.1 Underlying mechanism of feedback alignment

Consider a conventional d layer ANN. We denote weight matrices, 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 number of neurons in layer ℓ (network dimensions), and $f(\cdot)$ is an element-wise activation function (the following analysis holds if the batch size is variable among mini-batches; however, for simplicity, we assume that it is a constant number for all of them). For $0 < \ell \leq d$, internal

state of neurons in layer ℓ are calculated according to $Z_\ell = L_{\ell-1}W_{\ell-1} + \mathbf{b}_\ell$ where $\mathbf{b}_\ell$ is bias vector and 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 matrix 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-TRW (Lillicrap et al., 2016), the error is backpropagated through constant random matrices (different from 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 according to

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

where η is 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, $\Delta W_{\ell,FA}[k]$ should be expanded by taking successive steps backward along the iterations and substituting every $W_\ell[k-o]$ for $0 \le o < k$ and $0 < \ell < d$. Assuming update steps to be small, by applying first-order Taylor approximation we have

$$\begin{aligned} \Delta W_{\ell,FA}[k] &= \eta L_\ell[k]^T \delta_{\ell+1,FA}[k] = \eta f\big(W_{\ell-1}[k]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\delta_{\ell+1,FA}[k] \\ &= \eta f\big(\{W_{\ell-1}[k-1]^T + \eta\delta_{\ell,FA}[k-1]^T L_{\ell-1}[k-1]\}L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\delta_{\ell+1,FA}[k] \\ &\approx \eta\big\{f\big(W_{\ell-1}[k-1]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big) \ + \\ &\quad f'\big(W_{\ell-1}[k-1]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big) \odot \eta\delta_{\ell,FA}[k-1]^T L_{\ell-1}[k-1]L_{\ell-1}[k]^T\big\}\delta_{\ell+1,FA}[k] \\ &\approx T^1_{\ell,aln}[k] + T^2_{\ell,aln}[k] + \cdots + T^k_{\ell,aln}[k] + \eta f\big(W_{\ell-1}[0]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\delta_{\ell+1,FA}[k] \end{aligned} \tag{3}$$

where for $1 \le o \le k$ and $0 < \ell < d$ we define

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

as *alignment term of order o* corresponding to layer ℓ and $\zeta^o_\ell[k] = L_{\ell-1}[k]W_{\ell-1}[k-o] + \mathbf{b}_\ell[k]$ (see Supplementary Note 2 for higher-order Taylor approximation).

Alignment terms are pivots of FA (Fig. 1A). For simplicity, consider linear ANNs where alignment terms reduce to

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

Occurrence of alignment depends on the transformation matrix $M^o_\ell[k] = \delta_{\ell+1}[k-o]^T L_{\ell-1}[k-o]L_{\ell-1}[k]^T\delta_{\ell+1}[k]$ which is applied to B_ℓ^T and if $M^o_\ell[k]$ partially preserves the direction of B_ℓ^T after matrix multiplication, $T^o_{\ell,aln}[k]$ partially aligns with B_1^T. 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 $B_1^T \measuredangle B_\ell^T M^o_{\ell,skew}[k] = 90°$ (see Supplementary Note 5). Hence, the amount of alignment is determined by $B_\ell^T \measuredangle B_1^T M^o_{\ell,sym}[k]$ and the ratio of $\|B_\ell^T M^o_{\ell,skew}[k]\|_F$ to $\|B_\ell^T M^o_{\ell,sym}[k]\|_F$ (Fig. S1). The more $M^o_\ell[k]$ resembles a symmetric matrix, the less this ratio is expected given an independent random B_ℓ^T.

In general, eigenvalues of any symmetric transformation matrix determine the properties of the transform. Eigenvalues of $M^o_{\ell,sym}$ play a determinative role in FA as well. For example, consider decomposition of a vector, to which the transformation $M^o_{\ell,sym}$ is applied, in the basis of orthogonal eigenvectors of $M^o_{\ell,sym}$. If $M^o_{\ell,sym}$ is positive semidefinite, it scales each of the components of the vector with a positive scalar which is the corresponding eigenvalue. Namely, it keeps each component in its previous direction (does not flip it by 180°) which is desirable for alignment. However, $M^o_{\ell,sym}$ being semidefinite is not necessary for alignment and if it has well arranged eigenvalues, like when the average of them is positive and their negative ones are relatively small, alignment is expected given an independent random B_ℓ^T (Fig. S1). In general, there can also be more complex conditions where eigenvalues of $M^o_{\ell,sym}$ are not well arranged but because of existence of some dependency between B_ℓ^T and M^o_ℓ, rows of B_ℓ^T lie near some left eigenvectors of $M^o_{\ell,sym}$ whose corresponding eigenvalues are positive and alignment happens. However, in the following, in a specific practical application of BP-TRW in a deep ANN, we show that dynamic of alignment terms are to a good degree of approximation independent of B_ℓ^T.

For the analysis of the nonlinear case of the alignment terms (equation 4), we can simply refer to the linear case (equation 5) if the nonlinearity is a mild distortion on the linear case and the element-wise matrix multiplications corresponding to nonlinearity in equation 4 are not determinant of the overall behavior of alignment terms. In practice, choosing the activation function to be an increasing function, which is a common choice in practical ANNs, supports these condition to be met (we will examine this in a practical ANN below, see Supplementary Note 3).

The structure of alignment terms provides a robust basis for injection of aligned components into forward weights under a variety of conditions depending on the statistical properties of network activities. In particular and according to this structure, autocorrelation and cross-correlation of error $(\delta_{\ell+1})$ and output signals $(L_{\ell-1})$ of neurons play an important roll in FA (according to the standard definition of autocorrelation and cross-correlation function of the stochastic processes).

Among many possible conditions that can lead to the alignment of $T^o_{\ell,aln}[k]$ with B_ℓ^T, one is that $L_{\ell-1}[k-o]$ and $\delta_{\ell+1}[k-o]$, resemble $L_{\ell-1}[k]$ and $\delta_{\ell+1}[k]$, respectively. In other words, elements of $L_{\ell-1}$ and $\delta_{\ell+1}$ are autocorrelated at lag o. This condition makes the transformation M^o_ℓ to resemble a symmetric positive semidefinite matrix. To provide an intuition, we imposed hypothetical error and output signals on an open-loop 2-layer ANN with ReLU nonlinearity (Fig. 1B). For example, in an extreme hypothetical condition where we initially drew elements of L_0 and δ_2 i.i.d. from $\mathcal{N}(0,1)$ and left them constant during the iterations, the transformation matrix $M^o_1 = \delta_2^T L_0 L_0^T \delta_2$ can be considered as the estimated autocorrelation matrix

of the data matrix $\sqrt{n_0}L_0^T\delta_2$. Therefore, in this case, M_1^o is a symmetric positive semidefinite matrix and alignment happens (Fig. 1B, dashed blue trace). In addition to autocorrelation, cross-correlation between error and output signals of neurons contributes to alignment. For example, we independently re-initialized elements of L_0 from $\mathcal{N}(0,1)$ at each iteration and let $\delta_2[k] = L_0[k]$. In this condition, error and output signals of neurons are white noise and unautocorrelated but they are cross-correlated and alignment happens (Fig. 1B, dotted blue trace). On the contrary, we independently re-initialized elements of L_0 and δ_2 from $\mathcal{N}(0,1)$ at every iteration. In this case, error and output signals of neurons are neither autocorrelated nor cross-correlated and alignment does not happen (Fig. 1B, solid orange trace, see Supplementary Note 3 for more detail).

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

$$\delta_{\ell,DFA}[k] = \begin{cases} \delta_{d,DFA}[k]F_\ell \odot f'\big(Z_\ell[k]\big) & 0 < \ell < d \\ -\eta \left.\frac{\partial \mathcal{L}}{\partial Z_d}\right|_k & \ell = d. \end{cases} \tag{6}$$

With this method, it has been reported that product of forward weights ($W_\ell W_{\ell+1}\cdots W_{d-1}$) aligns with F_ℓ^T (Crafton et al., 2019). To see the reason of this, we should start from $\ell = d-1$ towards $\ell = 1$ and decompose every $\hat{F}_\ell$ such that for $\ell = d-1$, $F_{d-1} = \hat{F}_{d-1} = Q_{d-1}$, and for $0 < \ell < d-1$,

$$\hat{F}_\ell = Q_{d-1}Q_{d-2}\cdots Q_{\ell+1}Q_\ell = \hat{F}_{\ell+1}Q_\ell = F_\ell - F_\ell^\perp \tag{7}$$

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 take similar steps as the equation 3 and extract *direct alignment terms* (see Supplementary Note 4). In the linear case, and by assuming the $F_\ell^\perp$ to be negligible ($\|F_\ell^\perp\|_F \approx 0$ and $F_\ell \approx \hat{F}_\ell$), direct alignment term of order o is

$$T_{\ell,daln}^o \approx \eta^2 Q_\ell^T F_{\ell+1}^T \delta_{d,DFA}[k-o]^T L_{\ell-1}[k-o]L_{\ell-1}[k]^T \delta_{d,DFA}[k]F_{\ell+1}. \tag{8}$$

According to the structure of direct alignment terms in this simplified condition, with an analysis similar to the above analysis, 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. However, in this work, our focus is on FA and we leave further investigation of DFA and its general condition (like when $F_\ell^\perp$ is not negligible) to future works.

2.2 Potential decline of alignment in the early layers of deep ANNs

Beyond the alignment of forward and backward weights, the alignment of $\Delta W_{\ell,FA}$ with $\Delta W_{\ell,BP}$ provides an approximation of update direction of BP by alignment of $\Delta W_{\ell,FA}$ with $\Delta W_{\ell,BP}$. If statistical properties of layers are regulated to be similar, we can expect $W_\ell \measuredangle B_\ell^T$ to be consistent in different layers. But even if this consistency holds for $W_\ell \measuredangle B_\ell^T$ in different layers, $\Delta W_{\ell,FA} \measuredangle \Delta W_{\ell,BP}$ potentially increases (less alignment) as the error is backpropagated towards earlier layers. This 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 determines the angle between them are $\delta_{\ell+1,BP}$ and $\delta_{\ell+1,FA}$. In a d-layer linear ANN, for the last layer ($\ell = d$) we have $\delta_{d,FA} = \delta_{d,BP} = \delta_d$, but for $0 < \ell < d$ we have

$$\delta_{\ell,FA} = \delta_d B_{d-1}B_{d-2}\cdots B_{\ell+1}B_\ell \tag{9}$$

$$\delta_{\ell,BP} = \delta_d W_{d-1}^T W_{d-2}^T \cdots W_{\ell+1}^T W_\ell^T. \tag{10}$$

According to these two successive matrix multiplications of backward and transpose of forward weights, 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 efficiency of BP-TRW compared to BP in deep ANNs (see Supplementary Note 1 for statistical intuition). This behavior have also been reported before (Moskovitz et al., 2018) and we also observed it in the learning process of a practical deep ANN (see below).

2.3 Using the provided theoretical tool to investigate FA in the learning process of a practical deep ANN

As we have demonstrated, alignment terms tend to align with the backward weights under a variety of conditions. In general, many aspects of neural activities influence their behavior and make them act differently in different situations. Accordingly, architecture of network (activation function, number of neurons in layers, number of layers, loss function, etc.), hyperparameters (learning rate, batch size, etc.), and properties of dataset influence neural activities and consequently the behavior of alignment terms. However, based on the the above analysis we have provided a powerful tool for investigating FA in the learning process of ANNs under various conditions. As an example, in the following, we will use the provided theoretical tool to investigate FA in the learning process of a specific deep ANN. To tackle this specific ANN, we have adopted approaches which can be used for analysis of other conditions and networks.

2.3.1 Network configuration

We trained a 5-layer nonlinear fully connected ANN (Fig. 6A) for handwritten digits classification on MNIST dataset. For nonlinearity, we chose $f(\cdot) = tanh(ReLU(\cdot))$ so that it 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

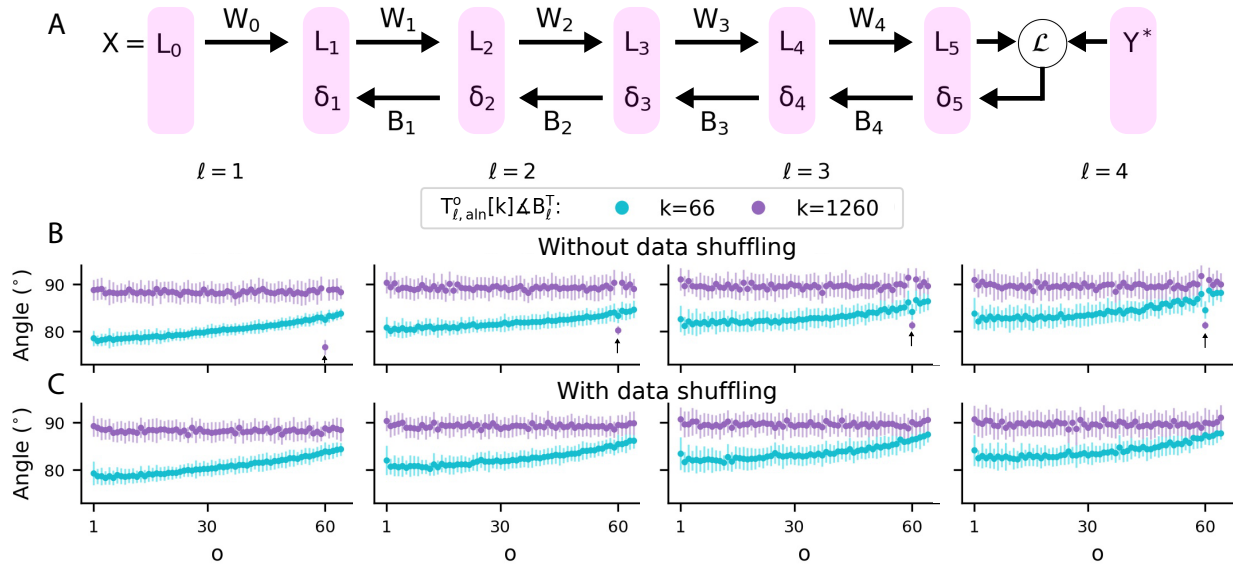

Figure 2 | **Behavior of alignment terms in the specific network under investigation.** **(A)** Layout of the network trained on MNIST. The number of neurons in each hidden and output layer is 50. Activation function of all neurons is $f(\cdot) = tanh(ReLU(\cdot))$. **(B,C)** The training dataset is divided into 60 mini-batches and it is not shuffled during the learning process (the mini-batches are the same in each epoch). At the initial phase of the learning process, all orders of alignment terms are aligned with B_ℓ^T and those whose orders are a multiple of 60 are slightly more aligned compared to their adjacent orders. As the learning process proceeds, the alignment terms whose orders are a multiple of 60 become more aligned and those whose orders are not a multiple of 60 lose their amount of alignment.

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 are 10 classes (digits), to match the length of the coding of desired output of 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 and then vectorized them. Hence, the number of input neurons was 225. We also normalized input data points (output of each input neuron) 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 are a total number of 60 mini-batches given the total number of 60000 training data points. As a baseline, we did not perform data shuffling (rearrangement of data among all mini-batches at the beginning of each epoch) in the learning process of the network but we will investigate the effect of data shuffling on the behavior of the alignment terms. We initialized elements of forward and backward weights and bias vectors i.i.d. from $\mathcal{N}(0, 0.1)$. The loss function that we used is $\mathcal{L} = \frac{1}{2}\sum_{i,j} E_{i,j}^2$, where $E_{i,j}$ is the element in i^{th} row and j^{th} column of $E[k] = Y^*[k] - Y[k]$.

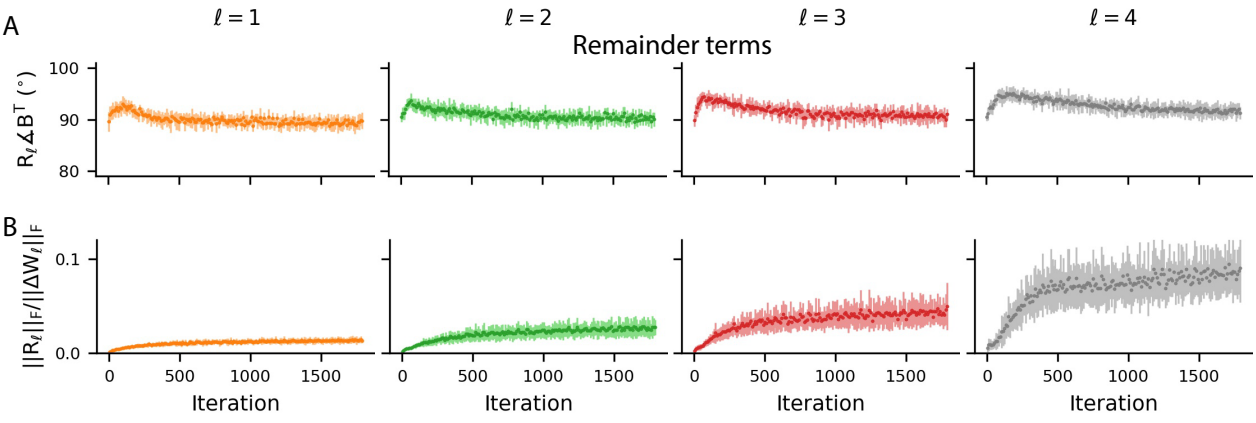

Figure 3 | **Taylor approximation and practical considerations. (A,B)** To extract the alignment terms, first order Taylor approximation is used (equation 4). Hence, some error is induced to the analysis by ignoring higher order terms of Taylor series. Moreover, in practice, the activation function used may have some nonanalytic points in its domain (like ReLU) which induce additional error in the analysis. Hence, investigating remainder terms is an importance step in the analysis. In the specific network under investigation, remainder terms slightly oppose the alignment at the initial phase of the learning process (A). Their opposition decreases through iterations, but they still do not show considerable amount of alignment (A). Moreover, they constitute a small proportion of weight update directions (B). Therefore, remainder terms are not responsible for the occurrence of alignment. Each dot is average over 30 runs and error bars are one s.d. around the mean.

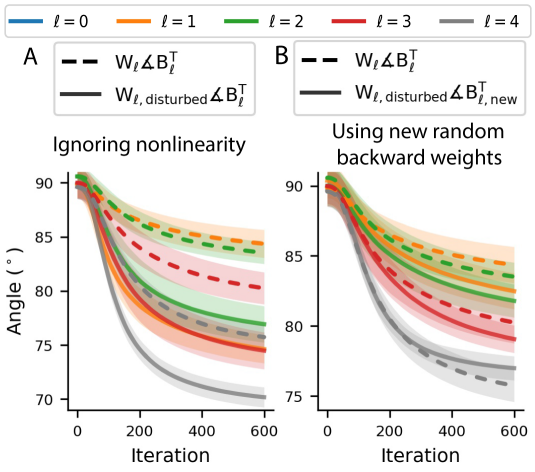

Figure 4 | In the specific network under investigation, occurrence of alignment is independent of nonlinearity and backward weights. (A) In the learning process of the network where its actual weight matrices (W_ℓ) are updated by the update rule of BP-TRW, $W_{\ell,disturbed}$ is updated by a disturbed version of update rule of BP-TRW which is $\sum_o T^o_{\ell,aln}[k] + \eta f'(W_{\ell-1}[0]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T)\delta_{\ell+1,FA}[k]$ and for calculating $T^o_{\ell,aln}$, nonlinearity is ignored (equation 5 is used instead of 4). Note that learning is intact and the actual error and output signals of neurons is used in the disturbed update rule to investigate the impact of nonlinearity on the alignment terms. Nonlinearity is a distortion on the alignment terms which reduces their amount of alignment. However, alignment still occurs when the nonlinearity is neglected in alignment terms. **(B)** In the learning process of the network where its actual weight matrices (W_ℓ) are updated by the update rule of BP-TRW, $W_{\ell,disturbed}$ is updated by a disturbed version of update rule of BP-TRW which is $\sum_o T^o_{\ell,aln}[k] + \eta f'(W_{\ell-1}[0]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T)\delta_{\ell+1,FA}[k]$ and for calculating $T^o_{\ell,aln}$, new fixed random backward weights (different from the actual backward weights of the network but from the same distribution) are used and the amount of alignment between $W_\ell[k]$ and the new fixed random backward weights is plotted. Note that learning is intact and the actual error and output signals of neurons is used in the disturbed update rule to investigate the impact of backward weights on the alignment terms. This distortion in the alignment terms affects the amount of alignment, but does not prevent the alignment from occurring.

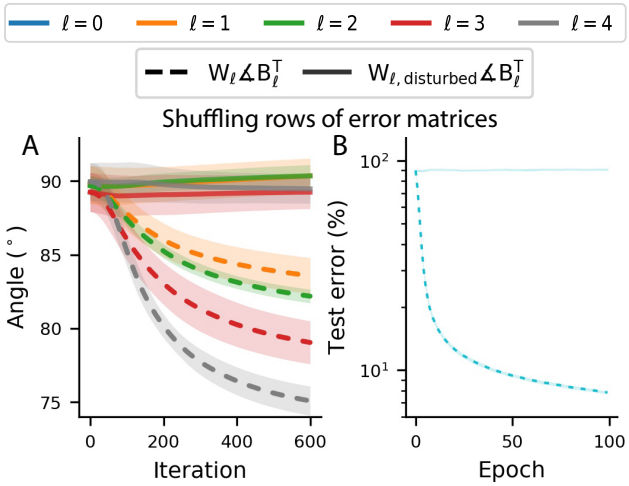

Figure 5 | Cross-correlation between error and output signals of neurons contributes to alignment. In the learning process of the network where its actual weight matrices (W_ℓ) are updated by the update rule of BP-TRW, $W_{\ell,disturbed}$ is updated by a disturbed version of update rule of BP-TRW which is $\sum_o T^o_{\ell,aln}[k] + \eta f'(W_{\ell-1}[0]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T)\delta_{\ell+1,FA}[k]$ and for calculating $T^o_{\ell,aln}$, rows of error matrices are shuffled. Since each row of error and output matrices correspond to a data point of each mini-batch, this maps error signals of an specific data point of the mini-batch to output signals of neurons that belong to a different data point of the mini-batch and distorts any dependency between them. Note that learning is intact and the actual error and output signals of neurons is used in the disturbed update rule to investigate the impact of backward weights on the alignment terms. This disrupts alignment and implies the contribution of cross-correlation between error and output signals of neurons to alignment.

2.3.2 Behavior of alignment terms

In this particular network trained by original form of BP-TRW, without data shuffling, at the initial phase of the learning process, all orders of alignment terms show a considerable amount of alignment and the terms whose orders are a multiple of 60 slightly show more amount of alignment compared to their adjacent orders. As the learning process proceeds, the amount of alignment of the terms whose orders are not a multiple of 60 decrease while the terms whose orders are a multiple of 60 become more aligned as the learning process proceeds (Fig. 2A,C).

With data shuffling, at the initial phase of the learning process, all orders of alignment terms show a considerable amount of alignment. As the learning process proceeds, the alignment terms whose orders are a multiple of 60, like the other orders of alignment terms, lose their initial amount of alignment (Fig. 2B,D). However, the behavior of ΔW_ℓ and W_ℓ with data shuffling are similar to their behavior without data shuffling (Fig. 2E).

In the next sections, we investigate different properties of neural activities contributing to alignment and emergence of these behaviors.

2.3.3 Taylor approximation and Practical considerations

In derivation of alignment terms for nonlinear networks (equation 3), we have 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 a separate Taylor series and we have neglected higher-degree Taylor polynomials which induces some error in the analysis of FA. Moreover, in practice, the activation function used may have some nonanalytic points in its domain (like ReLU at zero) inducing additional error in the analysis of FA. Therefore, investigating the approximation error is an important step in the analysis of FA.

In our specific practical ANN, we have investigated the remainder terms

$$R_\ell[k] = \Delta W_{\ell,FA}[k] - \sum_{o=1}^{k} T^o_{\ell,aln}[k] - \eta f'(W_{\ell-1}[0]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T)\delta_{\ell+1,FA}[k] \tag{11}$$

and observed that $\|R_\ell\|_F$ is small relative to $\|\Delta W_{\ell,FA}\|_F$ and R_ℓ slightly oppose the alignment at the initial phase of the learning process. Its opposition decreases through iterations, but it still does not show considerable amount of alignment (Fig. 3A,B). Therefore, remainder terms are not responsible for the occurrence of alignment.

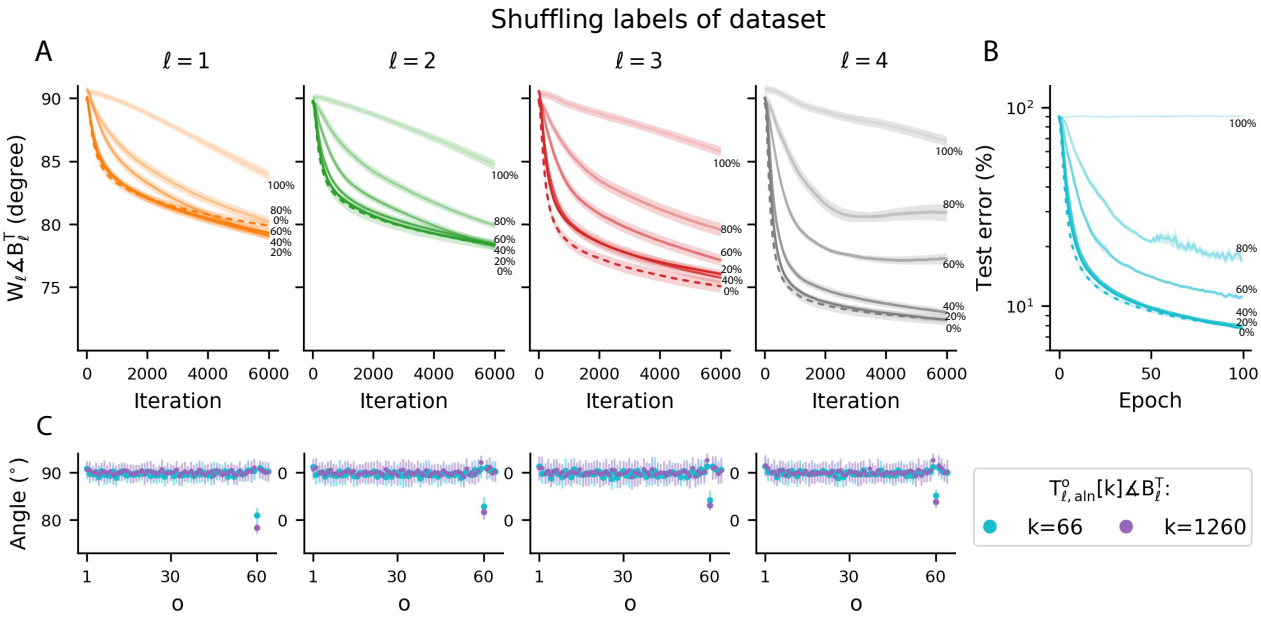

Figure 6 | **Intrinsic properties of datasets contribute to alignment. (A,B)** Intrinsically, data points belonging to a single category are more similar to each other than the ones belonging to different categories in the intact dataset. This statistical property of the MNIST dataset is distorted by shuffling different percentages of labels of the dataset. Consistency between input data and desired outputs contributes to the fast initial alignment as well as the amount of alignment in the late phase of the learning process. **(C)** At the beginning of the learning process, the MNIST dataset is distorted by shuffling all (100%) of the labels of its data points, and during epochs, data shuffling is not performed. This disrupts the initial alignment of the alignment terms whose orders are not a multiple of 60 by disrupting the constructive cross-correlated neural activities, whereas the alignment term of order 60 is still aligned because of autocorrelated neural activities and repetition of the same mini-batches after 60 iterations. Consistency between input data and desired outputs, and the property that within categories data points of datasets are more similar to each other than between categories ones, contribute to the fast initial alignment as well as the amount of alignment in the late phase of the learning process by shaping cross correlated neural activities (refer to Fig. S5 for more details). Each dot is average over 30 runs and error bars are one s.d. around the mean.

In addition to alignment terms and remainder terms, another term that appears in Taylor series expansion is $\eta f\big(W_{\ell-1}[0]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\delta_{\ell+1,FA}[k]$ (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 alignment terms and in this specific practical ANN, does not show considerable amount of alignment (Fig. 3C).

2.3.4 Dynamic of alignment terms with respect to nonlinearity and backward weights

Nonlinearity appears as two element-wise matrix multiplications in the structure of alignment terms (equation 4). However, if we are able to regard them as a distortion which impacts the amount of alignment but does not determine its occurrence, for simplicity, we can ignore them and refer to the linear case (equation 5) for analysis of the behavior of alignment terms. To examine this, in the learning process of the network by BP-TRW, we disturbed alignment terms by ignoring element-wise matrix multiplications (using equation 5 instead of 4, learning was intact and disturbed terms did not contribute to learning) and updated a disturbed version of forward weights by the disturbed alignment terms. Comparison of the alignment of actual forward weights of the network with the alignment of the disturbed forward weights shows that nonlinearity reduces the amount of alignment but ignoring it does not prevent alignment from occurring (Fig. 4A). Moreover, we investigated the impact of ignoring nonlinearity on the dynamic of individual alignment terms and observed that although nonlinearity reduce their amount of alignment, it does not change their overall behavior. Based on these observations, in the following, we refer to the linear case of alignment terms (equation 5) for analysis of their behavior.

Another statistical property that impacts analysis of alignment terms is independency of their dynamic with respect to backward weights. In other words, if there is an strong dependency between backward weights (B_ℓ^T) and the transformation matrices (M_ℓ^o) in the linear case of alignment terms (equation 5), then the analysis of alignment terms would be more complicated since we can not analyze the transformation matrix regardless of the backward weights. To examine this, in the learning process of the network by BP-TRW, we disturbed alignment terms by using new fixed random backward weights (learning was intact and disturbed terms did not contribute to learning) and updated a disturbed version of forward weights by the disturbed alignment terms. Comparison of alignment between actual forward weights and actual backward weights of the network with alignment between disturbed forward weights and their corresponding new fixed random backward weights shows that although there is a slight dependency between backward weights and the transformation matrices which slightly impacts the amount of alignment, abolishing this dependency by using this disturbance does not disrupt alignment (Fig. 4B). Moreover, we investigated the impact of this disturbance on the dynamic of individual alignment terms and observed that although it slightly changes their amount of alignment, it does not change their overall behavior. Based on these observations, in the following, we assume that the transformation matrices in alignment terms are independent of backward weights.

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

Since we have divided the dataset into 60 mini-batches, without data shuffling, each mini-batch is repeated after 60 iterations. Therefore, neural activities show a considerable amount of autocorrelation at lags that are a multiple of 60 which cause the alignment terms whose orders are a multiple of 60 to behave differently as described above. Referring back to equation 5, an appropriate condition for occurrence of alignment is that $L_{\ell-1}[k-o]$ and $\delta_{\ell+1}[k-]$, resemble $L_{\ell-1}[k]$ and $\delta_{\ell+1}[k]$, respectively. Without data shuffling, this condition holds for lags that are a multiple of 60. Moreover, as the learning process proceeds, the output and error signals of neurons corresponding to each data point become more stable. Hence, in the late phase of the learning process, $L_{\ell-1}[k-60]$ and $\delta_{\ell+1}[k-60]$, resemble $L_{\ell-1}[k]$ and $\delta_{\ell+1}[k]$, respectively, more than initial phase (Fig. S3) and alignment term of order 60 become more aligned as the learning process proceeds.

With data shuffling, this considerable amount of autocorrelation does not exist in the error signals of neurons at lags that are a multiple of 60 and the alignment terms whose orders are a multiple of 60 lose their amount of alignment during iterations. However, beyond each individual order of alignment terms, the final behavior of ΔW_ℓ is influenced by the resultant of all orders of alignment terms. This makes FA robust in many conditions like data shuffling. Although data shuffling changes the trajectory of weight matrices, assuming the update steps to be small, in the sense of statistical properties contributing to FA, shuffling is similar to substituting different rows of error and output matrices of the network among different lags with each other. In the other words, the autocorrelated activity of neurons that without data shuffling was concentrated in the lag of $o = 60$, with data shuffling, become distributed among lags of $o = 60$ up to $o = 119$. This changes the behavior of individual alignment terms but preserves the behavior of their resultant and consequently the behavior of ΔW_ℓ remains similar to the condition that data shuffling is not applied (Fig. 2E, see Supplementary Note 3).

2.3.6 Cross-correlation between error and output signals of neurons contributes to alignment

To see if in addition to the autocorrelation of output and error signals of neurons, cross-correlation between them contributes to alignment, in the learning process of the network by BP-TRW without data shuffling, we disturbed alignment terms by shuffling the rows of error matrices at every iteration (learning was intact and disturbed terms did not contribute to learning). Since each row of error and output matrices correspond to a data point of each mini-batch, this maps error signals of an specific data point of the mini-batch to output signals of neurons that belong to a different data point of the mini-batch and distorts the dependency between them. We updated a disturbed version of forward weights by the disturbed alignment terms. Comparison of the alignment of actual forward weights of the network with the alignment of the disturbed forward weights shows that this disturbance disrupts occurrence of alignment (Fig. 5). Moreover, we investigated the impact of this disturbance on the dynamic of individual alignment terms and observed that it changes their changes their overall behavior and disrupt their alignment. These observations imply that a dependency between error and input signals contributes to alignment. To investigate the origin of this dependency, we have taken the following approach.

In addition to the angle, another criterion for measuring the amount of alignment of each $T^o_{\ell,aln}[k]$ is the cosine similarity between $T^o_{\ell,aln}[k]$ and B^T_ℓ (see Methods). Since the denominator of cosine similarity is nonnegative, if we want to investigate the occurrence of alignment regardless of its amount, we can simply refer to 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 aligned with B^T_ℓ. Moreover, based on the above provided evidence about the dynamic of alignment terms with respect to nonlinearity and backward weights, we can refer to linear case of alignment terms and assume the backward weights (B_ℓ) to be independent of the transformation matrices (M^o_ℓ). Based on these assumptions and assuming the random elements of B^T_ℓ to be i.i.d. with zero mean and the variance of σ^2, we can state that alignment between $T^o_{\ell,aln}[k]$ and B^T_ℓ is expected if

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

where we define $S^o_{L_{\ell-1}}[k] = L_{\ell-1}[k-o] L_{\ell-1}[k]^T$ as *similarity matrix* between output of layer $\ell-1$ in iteration k and $k-o$, and $S^o_{L_{\ell-1}}[k]_{p,m}$ denotes the element in p^{th} row and m^{th} column of this matrix. Namely, as a measure of similarity, $S^o_{L_{\ell-1}}[k]_{p,m}$ is the dot product between the output signals of neurons in the layer $\ell-1$ emerged by the p^{th} data point of the $(k-o)^{th}$ batch and their output signals emerged by the m^{th} data point of the k^{th} batch. We define $S^o_{\delta_{\ell+1}}[k]$ in the same way for error signals of layer $\ell+1$ and also 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 p^{th} data point of the $(k-o)^{th}$ batch and the m^{th} data point of the k^{th} batch, the similarity terms have different behaviors. Accordingly, if these two mentioned data points both belong to the same category, we regard their corresponding similarity term as a *within categories similarity term*, and if they belong to two different categories, we regard their corresponding similarity term as a *between categories similarity term*. In within categories similarity terms, $S^o_{\delta_{\ell+1}}[k]_{p,m}$ have a positive mean (Fig. S5A, blue triangles) which is constructive for alignment, and in between categories similarity terms, it have a negative mean (Fig. S5A, red triangles) which is destructive for alignment (referring to equation 12). However, there is a constructive dependency between $S^o_{\delta_{\ell+1}}[k]_{p,m}$ and $S^o_{L_{\ell-1}}[k]_{p,m}$ which is that in within categories similarity terms $S^o_{L_{\ell-1}}[k]_{p,m}$ has a higher mean compared to between categories similarity terms (Fig. S5B).

This constructive dependency originates from an intrinsic property of datasets including the MNIST which is that within categories data points are more similar to each other than between categories ones. This feature directly shows itself in the input layer of the network and makes $S^o_{L_0}[k]_{p,m}$ of within categories similarity terms to have a higher mean compared to that of between categories similarity terms (Fig. S5B, blue vs. red triangles, first column). At the initial phase, although the network do not discriminate between categories, this feature is also preserved in the similarity terms of subsequent layers

(Fig. S5B, blue vs. red triangles). Note that the number of within categories similarity terms is less than the number of between categories ones. For example, if we have 10 categories and equal number of data in each category, 10% of similarity terms are within categories and 90% of them are between categories. Nevertheless, in the initial phase of the learning process, the constructive dependency 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 categories similarity terms. However, as the learning process proceeds, this constructive dependency weakens.

After the initial phase, some data points become well classified and consequently the Frobenius norm of their corresponding error signals ($\|\delta_{\ell,FA}[k]_{i,*}\|_F$, which denotes the Frobenius norm of i^{th} row of $\delta_{\ell,FA}[k]$ corresponding to i^{th} data point in the k^{th} mini-batch) decrease to about zero. On the contrary, some data points still remain poorly classified and consequently the Frobenius norm of their corresponding error signals remain large (Fig. S5C).

The poorly classified data points weaken the constructive dependency between $S^o_{L_{\ell-1}}[k]_{p,m}$ and $S^o_{\delta_{\ell+1}}[k]_{p,m}$ and this causes the alignment terms (without data shuffling, those whose orders are not a multiple of 60) to lose their initial amount of alignment during iterations. The destructive effect of poorly classified data can be seen in the distributions of $S^o_{\delta_{\ell+1}}[k]_{p,m}$ and $S^o_{L_{\ell-1}}[k]_{p,m}$ at the late phase of the network. For this reason, in addition to categories, we have separated similarity terms base on the quality of classification of the p^{th} data point of the $(k-o)^{th}$ batch and the m^{th} data point of the k^{th} batch (Fig. S5E,F). Accordingly, if both of these two mentioned data points belong to a subset of 60% of total data whose data points are the most well classified ones (their error signals at the output layer of the network have the least Frobenius norms), we regard their corresponding similarity term as a *well classified similarity term*, and otherwise, we regard their corresponding similarity term as a *poorly classified similarity term*.

In the late phase of the network, by average, $S^o_{\delta_{\ell+1}}[k]_{p,m}$ of well classified similarity terms has smaller magnitude (absolute value) compared to that of poorly classified similarity terms (Fig. S5D, dark vs. light triangles). Therefore, poorly classified data points dominate well classified ones in shaping the dynamic of the alignment terms in the late phase of the network. Within categories, poorly classified data points and also their representations in different layers of the network are less similar to each other compared to well classified ones in the late phase of the network (Fig. S5E, dark vs. light blue triangles). This weakens the constructive dependency between $S^o_{L_{\ell-1}}[k]_{p,m}$ and $S^o_{\delta_{\ell+1}}[k]_{p,m}$ and causes the alignment terms (without data shuffling, those whose orders are not a multiple of 60) to lose their amount of alignment as the learning process proceeds.

To further elucidate the contribution of inherent statistical properties of the MNIST dataset to FA, we distorted it by shuffling true labels once at the beginning of the learning process and afterwards data shuffling is not applied (each mini-batch is repeated after 60 iterations). Shuffling all (100%) of the true labels makes the histograms of $S^o_{L_0}[k]_{p,m}$ corresponding to both within and between categories similarity terms to lie on each other. This disrupts the constructive dependency between $S^o_{L_{\ell-1}}[k]_{p,m}$ and $S^o_{\delta_{\ell+1}}[k]_{p,m}$ in within categories similarity terms and prevents the alignment terms whose orders are not a multiple of 60 from being aligned while the alignment term of order 60 is still aligned due to autocorrelated neural activities at the lag of 60. In this condition, network learns no useful information (Fig. 6B) yet W_ℓ slightly aligns with B^T_ℓ (Fig. 6A). Decreasing the percentage of the shuffled labels makes the alignment to occur more quickly and robustly (Fig. 6A). Therefore, the property that within categories data points are more similar to each other than between categories ones, contributes to the fast initial alignment of W_ℓ with B^T_ℓ by shaping cross-correlated neural activity.

2.3.7 Behavior of forward weights and update directions

In the learning process of the network by BP-TRW, forward weight matrices align with backward weights (Fig. S3A). Update directions of BP-TRW aligns with those of BP and the amount of alignment between them decreases step by step as error is backpropagated towards the earlier layers (Fig. S3B).

To compare the trajectory of all learnable parameters of the network trained by BP-TRW with that of the network trained by BP, we performed principle component analysis on two instances of the network that were both identically initialized with the same parameters but one was trained by BP and the other by BP-TRW. With BP-TRW, the networks converges to a local minimum totally different from the local minimum to which the network converges with BP (Fig. S3C).

2.3.8 Utility of weight normalization for improvement of alignment

Lillicrap et al. 2016 have 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 the Frobenius norm of the weight matrix of the first layer leads to alignment (Supplementary Note 12 of Lillicrap et al. 2016). In general, correlation of alignment with the increase in Frobenius norm of the weight matrices is intuitive since alignment terms inject a component along the B^T_ℓ into W_ℓ and continuous accumulation of these components leads to continuous growth of the Frobenius norm of W_ℓ. However, if the continuous growth of the Frobenius norm of the weight matrices or unbounded accumulation of the aligned components was crucial for FA, it would be biologically problematic since the synaptic weights are unbounded. Therefore, to examine this, inspired from synaptic scaling (Turrigiano, 2008), we limited and fixed the Frobenius norm of input weights to each neuron at each iteration 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 consist of i^{th} column of $W_\ell[k]$ and γ is a positive scalar. To treat all weights in the same way, we also applied this WN method to the fixed random backward weight at the beginning of the learning process. This proposed WN method is as an intervention in the BP-TRW formula, unlike the conventional WN method in ANNs (Salimans and Kingma, 2016) which is as a reparameterization of BP formula. However, the utility of using a similar WN method in BP-TRW has been reported previously (Liao et al., 2016; Moskovitz et al., 2018).

In this particular network, without WN, Frobenius norms of weight matrices grow continuously, although they saturate very quickly in the earlier layers (Fig. 7D). By applying the proposed WN method with $\gamma = 1$ alignment improved significantly (Fig. 7A,B). This WN method also improved test accuracy of the network when it was trained by BP-TRW, whereas it

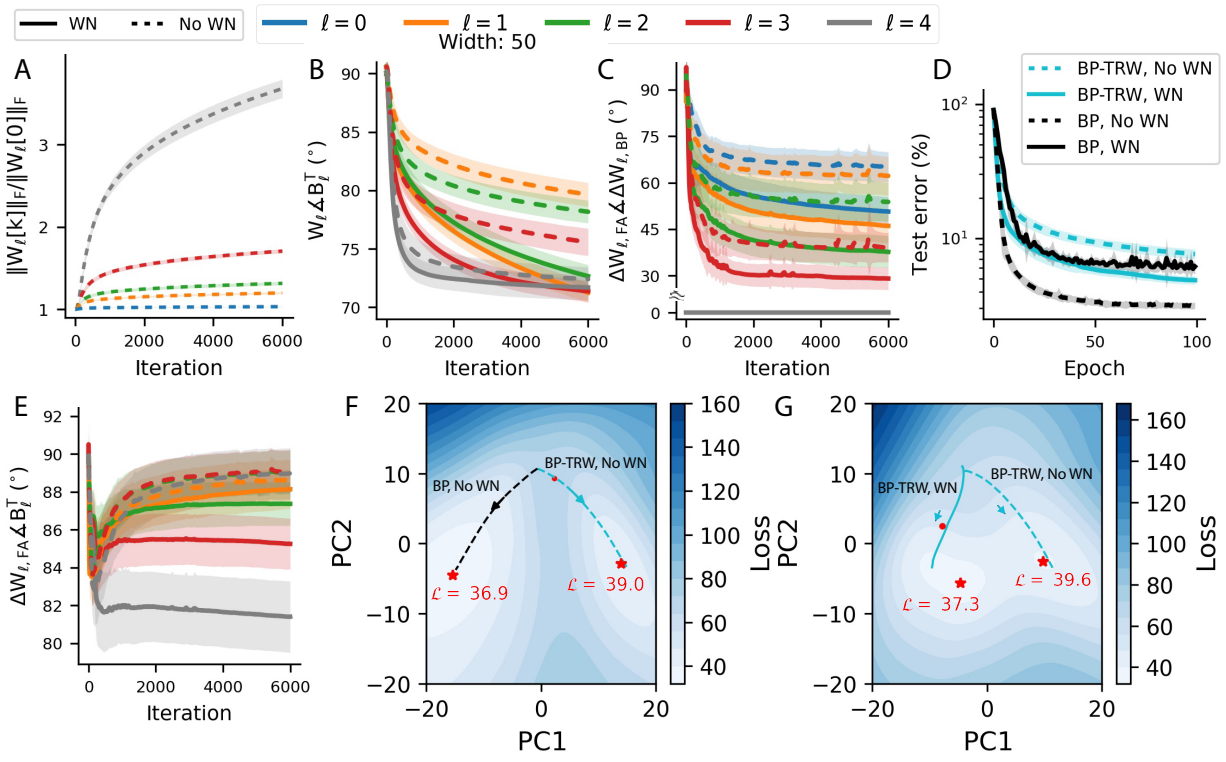

Figure 7 | Behavior of forward weights and update directions, and utility of weight normalization in training a deep ANN by BP-TRW for handwritten digits classification. **(A)** In BP-TRW, W_ℓ aligns with B_ℓ^T and WN can improve alignment. **(B)** Update directions prescribed by BP-TRW align with update directions prescribed by BP and WN can improve their alignment. Deviation between update directions prescribed by BP-TRW and BP increases as successively error signals are backpropagated towards earlier layers. At the last layer ($\ell = 4$) update direction of BP-TRW is the same as BP. Each trace is passed through a moving average filter of length 60. **(C)** Principle component analysis is performed on the trajectories of all learnable parameters of 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-TRW. For plotting the contour map, some points from a grid in the PC1-PC2 space are brought back into the original space of all learnable parameters and the loss function of the network is calculated by using the reconstructed parameters. Red dot is the projection of all backward weights on the PC1-PC2 space . Black and cyan arrows are projections of update directions prescribed by BP and BP-TRW, respectively. Red asterisks are local minimums in the PC1-PC2 space. **(D)** Without WN, Frobenius norm of forward weight matrices continuously increases, although it saturates quickly in the earlier layers. **(E)** WN can improve the test accuracy of BP-TRW whereas it makes the test accuracy of BP less robust. **(F)** Similar to the panel C but for the trajectories of all learnable parameters of two instances of the network trained by BP-TRW, with and without WN. (A,B,D,E) Each trace is the average over 10 runs and the shaded areas are one s.d. around the mean.

makes the test accuracy less robust when the network was trained by BP (Fig. 7E). These results are sensitive to the hyperparameter γ and the learning rate η. For example, with small amounts of γ, the performance of the network severely drops (supplementary Fig. S8). Note that even in biological networks, hyperparameters play a crucial role and many diseases, like Alzheimer, are believed to occur because of the deflection in synaptic strengths (Verret et al., 2012; Frere and Slutsky, 2018; Styr and Slutsky, 2018). However these results are robust for a fair range of γ and η. We did not optimized them and simply chose them based on the sensitivity analysis and a grid search (supplementary Fig. S8).

To compare the trajectories of all learnable parameters of the networks trained by BP-TRW with and without WN, we performed principle component analysis on two instances of the network that were both identically initialized with the same parameters but one was trained with WN and the other without WN. With WN, the network converges to a local minimum totally different from the local minimum to which the network converges without WN (Fig. S3F).

3 Discussion

Mathematical basis of FA. 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 strengths. The BP-TRW learning method (Lillicrap et al., 2016) showed that an ANN can be trained with constant random feedback weights distinct from feedforward weights. In BP-TRW, forward and backward weights partially align together during iterations which leads to alignment between update directions of BP-TRW and BP at each iteration and provides an approximation of BP.

In this work, we demonstrated mathematical and statistical basis of FA and showed that alignment itself is not due to the learning process or reduction of loss function; rather, it works as a statistical expectation under certain conditions depending on neural activities. Hence, statistical properties of neural activities as well as distribution of weights affect the occurrence of alignment and its amount.

Specifically, in our case studies, we showed that autocorrelation of error and output signals of neurons and cross-correlation between them are two important features of neural activities contributing to alignment (Fig. 1, Fig. S3, Fig. 5, Fig. S5). Furthermore, we showed that one of the intrinsic properties of datasets, which is within categories data points are more similar to each other than between categories ones, contribute to FA by shaping cross-correlated neural activities.

In this work we dissected FA in a specific deep nonlinear ANN trained on MNIST dataset. In other networks and conditions, depending on the network configuration, analysis of FA can be different to some extent. However, the approaches we took here to tackle this particular network and the underlying mechanism of FA we demonstrated here, provide powerful tools for investigating FA in other conditions and networks.

A weakness of BP-TRW as an approximation of BP in deep ANNs, which is studied in this work, is the potential decline in amount of alignment as the error is backpropagated towards the earlier layers (Fig. 7C). This potential decline may be overcome by some special considerations in future works. However, poor FA in the early layers may enhance the capability of the system for 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).

Weight normalization can improve alignment. Normalization is an integral part of the current state of the art ANNs. The correspondence of normalization methods in ANNs and biological ones has been taken into consideration (Shen et al., 2020) and in the context of BP-TRW, the utility of using normalization methods, is reported (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, have been reported (Murthy et al., 2001; Surmeier and Foehring, 2004; Turrigiano, 2012).

Accordingly, we proposed a strict WN method, which is done by fixing the Frobenius norm of input weights of each neuron to a limited amount at each iteration. We showed that this WN method can improve alignment and the test error of the network (Fig. 7). This WN method is as an intervention in the BP-TRW formula and we considered the amount to which the norms of weights are fixed (γ) a nonlearnable hyperparameter of the network as homeostatic plasticity in the brain has been suggested to act in a nonassociative manner with the goal of network stabilization which can interfere with associative (Hebbian) plasticity (Turrigiano, 2017). In addition to homeostatic plasticity, which acts at a slower rate than Hebbian plasticity (Turrigiano, 2017), other faster forms of plasticity like heterosynaptic plasticity are reported which regulate the total synaptic weights in a competitive manner, that is, potentiation of a synapse can result in depression of another synapse (Chistiakova et al., 2015).

Approximation of BP in biological networks and its further biological implausibilities. Different aspects of neural activities shown here that contribute to FA can be biologically plausible. For example, autocorrelated neural activity is biologically plausible as individual biological neurons are reported to have intrinsic (regardless of stimulus) spike-count autocorrelation which is significant in relatively low time lags and decays in high time lags (Cirillo et al., 2018; Murray et al., 2014; Ogawa and Komatsu, 2010; Fascianelli et al., 2019). However, weight transport problem is only one of the biological implausibilities of BP formula which can be avoided by using BP-TRW methods and there are other biological implausibilities in both BP and BP-TRW (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-TRW are signed. In addition, error signals in BP and BP-TRW are distinct from the output of artificial neurons. In BP-TRW and BP, error signals are internal attributes of neurons that are backpropagated to other neurons through feedback weights, whereas in biological networks, the only attribute of each neuron that is conveyed explicitly to other neurons by axons and synapses is its output spike and other internal attributes of each neuron are believed to be local (Stork, 1989; Song et al., 2020). Therefore, it has been suggested that feedforward and feedback signals may be generated as output of neurons separately at different times (Lillicrap et al., 2016).

Despite the biological implausibilities of BP-TRW and BP, it has been suggested that an approximation of them may occur in the biological networks on the basis of synaptic plasticity (Whittington and Bogacz, 2019, 2017; Lillicrap et al., 2020). According to this, we can imagine two different scenarios for the occurrence of FA in the brain. In the first one, the error signals are backpropagated through some other neurons (or a parallel network) distinct from feedforward ones (Akrout et al., 2019; Guerguiev et al., 2017). In this scenario, there should be one-to-one cross-projections between the neurons in the forward and backward paths which does not seem biologically plausible. In the second scenario, the error signals are backpropagated by the same neurons of the forward path through some other axons and synapses distinct from feedforward ones. If we assume that an approximation of BP-TRW occurs in the brain, this latter scenario suggests that an approximation of the calculation of internal error signals ($\delta_{\ell+1}$) may occur locally (Guerguiev et al., 2017), and afterwards the operation of $\Delta W_\ell = \eta L_\ell^T \delta_{\ell+1}$ can be done by some sort of homosynaptic (Hebbian or activity-dependent) plasticity.

In the BP-TRW learning method, the backward weight matrices are constant, but if we consider them as matrices of synaptic feedback weights, synaptic plasticity also applies to them and their strength is not constant during iterations. Accordingly, it can be imagined a state in which not only forward weights are propelled towards feedback weights but also feedback weights are propelled towards forward weights. In this state, it may that instead of calculation of an approximation of $\delta_{\ell+1}$ as an internal attribute of biological neurons, the operation of $\Delta W_\ell = \eta L_\ell^T \delta_{\ell+1}$ is performed approximately by heterosynaptic plasticity between forward and backward synapses afferent to the neurons.

Highly nonrandom features of local cortical circuits and its possible relation to FA. FA 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 in local cortical circuits (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 FA. In this regard, by applying timing-dependent synaptic plasticity rules, the emergence of dominant reciprocal strong connections has been reported in a recurrent ANN of spiking neurons under rate-coded input (Clopath et al., 2010).

FA may be just a piece of the brain puzzle. In summary, the analysis done in this study portray a clearer picture of the FA mechanism 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-TRW 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 segregation of excitatory and inhibitory neurons affect the performance of BP-TRW and BP. In addition to the fully connected architectures considered here, BP-TRW in more biologically relevant networks, such as locally connected or recurrent networks, remains to be more explored in the future.

4 Methods

4.1 BP and BP-TRW learning methods

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

$$W_\ell[k+1] = W_\ell[k] + \Delta W_\ell[k] \tag{14}$$

$$\mathbf{b}_\ell[k+1] = \mathbf{b}_\ell[k] + \Delta \mathbf{b}_\ell[k] \tag{15}$$

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

$$\Delta W_{\ell,BP}[k] = -\eta \left.\frac{\partial \mathcal{L}}{\partial W_\ell}\right|_k = \eta L_\ell[k]^T \delta_{\ell+1,BP}[k],\ 0 \le \ell < d \tag{16}$$

$$\Delta \mathbf{b}_{\ell,BP}[k] = -\eta \left.\frac{\partial \mathcal{L}}{\partial \mathbf{b}_\ell}\right|_k = \eta J_{1\times n_b} \delta_{\ell,BP}[k],\ 0 < \ell \le d \tag{17}$$

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

$$\delta_{d,BP}[k] = E[k] \odot f'\big(Z_d[k]\big) \tag{18}$$

$$\delta_{\ell,BP}[k] = \delta_{\ell+1,BP}[k] W_\ell[k]^T \odot f'\big(Z_\ell[k]\big),\ 0 < \ell < d \tag{19}$$

where $\odot$ denotes element-wise matrix multiplication (in the order of operations, it has less priority than matrix multiplication), $E[k] = Y^*[k] - Y[k]$, and $f'(\cdot)$ is element-wise derivative of activation function, and η is learning rate (Rumelhart et al., 1985).

In BP-TRW (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 19 is replaced with B_ℓ)

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

$$\delta_{\ell,FA}[k] = \delta_{\ell+1,FA}[k] B_\ell \odot f'\big(Z_\ell[k]\big),\ 0 < \ell < d \tag{21}$$

$$\Delta W_{\ell,FA}[k] = \eta L_\ell[k]^T \delta_{\ell+1,FA}[k],\ 0 \le \ell < d \tag{22}$$

$$\Delta \mathbf{b}_{\ell,FA}[k] = \eta J_{1\times n_b} \delta_{\ell,FA}[k],\ 0 < \ell \le d. \tag{23}$$

In the context of training a network with BP-TRW, $\Delta W_{\ell,BP}[k]$ is a direction that we only calculated at each iteration for the purpose of comparison with $\Delta W_{\ell,FA}[k]$, which we actually used 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 with the same dimensions as follows

$$W \measuredangle B = cos^{-1}\left(\frac{\langle W, B\rangle_F}{\|W\|_F \|B\|_F}\right) \tag{24}$$

where $\langle W, B\rangle_F$ is the Frobenius inner product of W and B and $\|.\|_F$ is 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 (regardless of $\|W\|_F$ and $\|B\|_F$).

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

$$\text{cosine similarity}(W, B) = \frac{\langle W, B\rangle_F}{\|W\|_F \|B\|_F}. \tag{25}$$

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. 1B we chose network dimensions to be $n_0 = n_2 = 20$, $n_1 = 100$, and $n_b = 100$, and we set $\eta = 0.002$ 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-TRW (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 where network width (number of neurons in output and hidden layers) was 50, we used mutually exclusive 5-hot coding. Suppose the number of categories is C and number of output neurons is m ($n \cdot C \leq m$). 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 Supporting Information

Supplementary Fig. 2. Behavior of alignment terms in the specific network under investigation.

Supplementary Fig. S3. Autocorrelation of error and output signals contributes to alignment.

Supplementary Fig. 5. Cross-correlation between error and output signals of neurons contributes to alignment.

Supplementary Fig. S5. Within categories data points are more similar to each other than between categories ones and it contributes to alignment by shaping cross-correlated neural activity.

Supplementary Fig. 3. Practical considerations.

Supplementary Note 1. Intuition about potential decline of alignment in the early layers of deep ANNs.

Supplementary Note 2. Taylor polynomials of higher degree.

Supplementary Note 3. Index notation of alignment terms.

Supplementary Note 5. Skew-symmetric part of a transformation matrix totally deviates the direction.

Supplementary Note 6. Analysis of network sensitivity to hyperparameters.

6 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).

7 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.

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.

Cirillo, R., Fascianelli, V., Ferrucci, L., and Genovesio, A. (2018). Neural intrinsic timescales in the macaque dorsal premotor cortex predict the strength of spatial response coding. *Iscience*, 10:203–210.

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.

Fascianelli, V., Tsujimoto, S., Marcos, E., and Genovesio, A. (2019). Autocorrelation structure in the macaque dorsolateral, but not orbital or polar, prefrontal cortex predicts response-coding strength in a visually cued strategy task. *Cerebral cortex*, 29(1):230–241.

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.

Frere, S. and Slutsky, I. (2018). Alzheimer's disease: from firing instability to homeostasis network collapse. *Neuron*, 97(1):32–58.

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

Guerguiev, J., Lillicrap, T. P., and Richards, B. A. (2017). Towards deep learning with segregated dendrites. *ELife*, 6:e22901.

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*.

Murray, J. D., Bernacchia, A., Freedman, D. J., Romo, R., Wallis, J. D., Cai, X., Padoa-Schioppa, C., Pasternak, T., Seo, H., Lee, D., et al. (2014). A hierarchy of intrinsic timescales across primate cortex. *Nature neuroscience*, 17(12):1661–1663.

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*.

Ogawa, T. and Komatsu, H. (2010). Differential temporal storage capacity in the baseline activity of neurons in macaque frontal eye field and area v4. *Journal of neurophysiology*, 103(5):2433–2445.

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.

Styr, B. and Slutsky, I. (2018). Imbalance between firing homeostasis and synaptic plasticity drives early-phase alzheimer's disease. *Nature neuroscience*, 21(4):463–473.

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. (2008). The self-tuning neuron: synaptic scaling of excitatory synapses. *Cell*, 135(3):422–435.

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

Verret, L., Mann, E. O., Hang, G. B., Barth, A. M., Cobos, I., Ho, K., Devidze, N., Masliah, E., Kreitzer, A. C., Mody, I., et al. (2012). Inhibitory interneuron deficit links altered network activity and cognitive dysfunction in alzheimer model. *Cell*, 149(3):708–721.

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.

Supporting Information

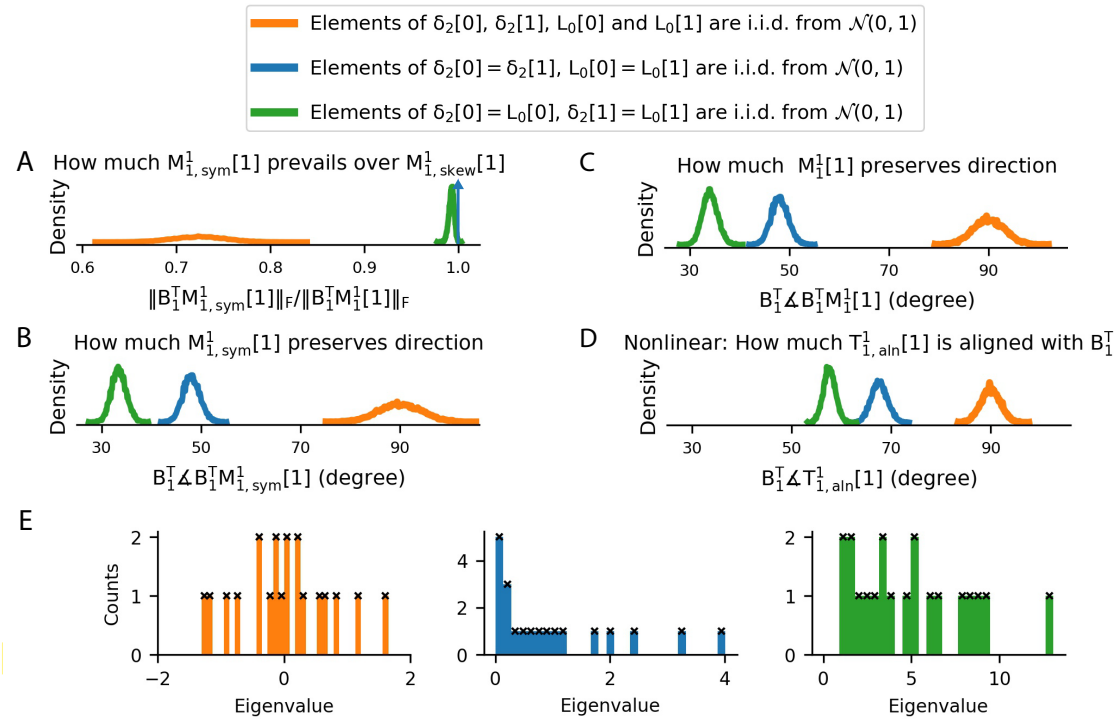

Figure S1 | **Providing intuition about the contribution of M_{sym} and its eigenvalues to alignment (A-F)**

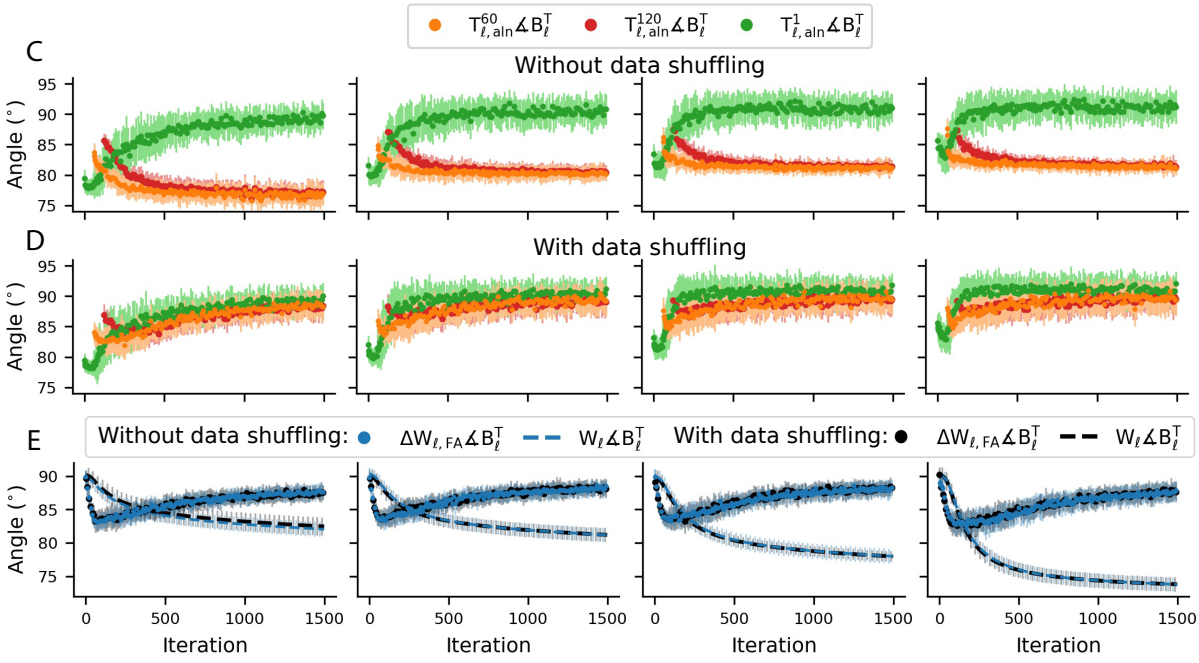

Figure S2 | **Behavior of alignment terms in the specific network under investigation.** **(A,B)** At the beginning of each epoch, the training dataset is shuffled and then it is divided into 60 mini-batches. At the initial phase of the learning process, all orders of alignment terms are aligned with B_ℓ^T. As the learning process proceeds, all of the alignment terms lose their amount of alignment. **(C)** Although with data shuffling the alignment terms whose orders are a multiple of 60 lose their amount of alignment during iterations, the behaviors of $\Delta W_{\ell,FA}$ and W_ℓ are similar to the case without data shuffling. (A-E) Each dot and trace is the average over 30 runs and error bars are one s.d. around the mean.

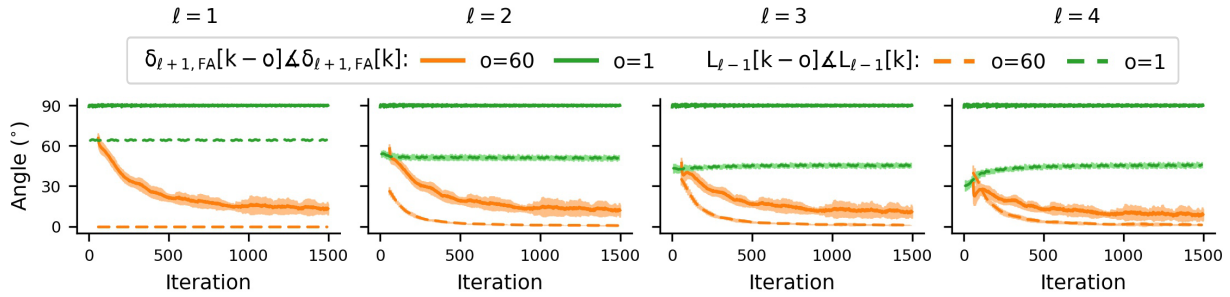

Figure S3 | **Autocorrelation of error and output signals contributes to alignment.** In the learning process of the network without data shuffling, the angle between output matrices at iterations k and $k - o$ and the angle between error matrices at iterations k and $k - o$ is plotted as a measure of the similarity between them. According to the structure of the alignment terms (equation 5), one of the conditions that can lead to alignment is that $L_{\ell-1}[k - o]$ and $\delta_{\ell+1}[k - o]$, resemble $L_{\ell-1}[k]$ and $\delta_{\ell+1}[k]$, respectively, since it makes the transformation matrix M^o_ℓ to resemble a symmetric positive semidefinite matrix. Without data shuffling, $\delta_{\ell+1,FA}[k]$ and $L_{\ell-1,FA}[k]$ are similar to $\delta_{\ell+1,FA}[k - 60]$ and $L_{\ell-1,FA}[k - 60]$, respectively, and as the learning process proceeds, they become more similar to each other since the network becomes trained and stable. This makes the alignment term of order 60 to become more aligned during iterations. Each trace is the average over 30 runs and error bars are one s.d. around the mean.

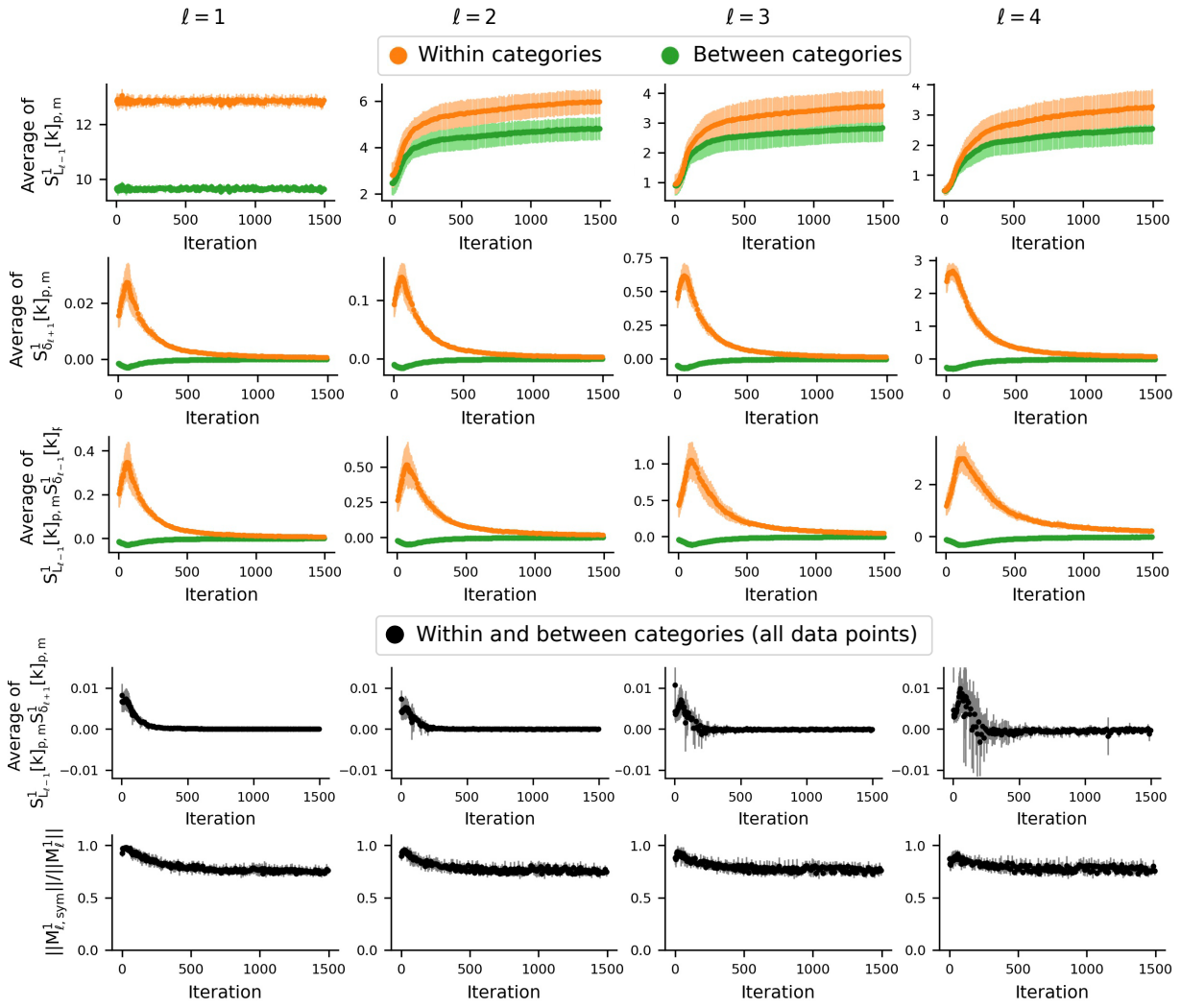

Figure S4 | **Trend of similarity terms during iterations.**

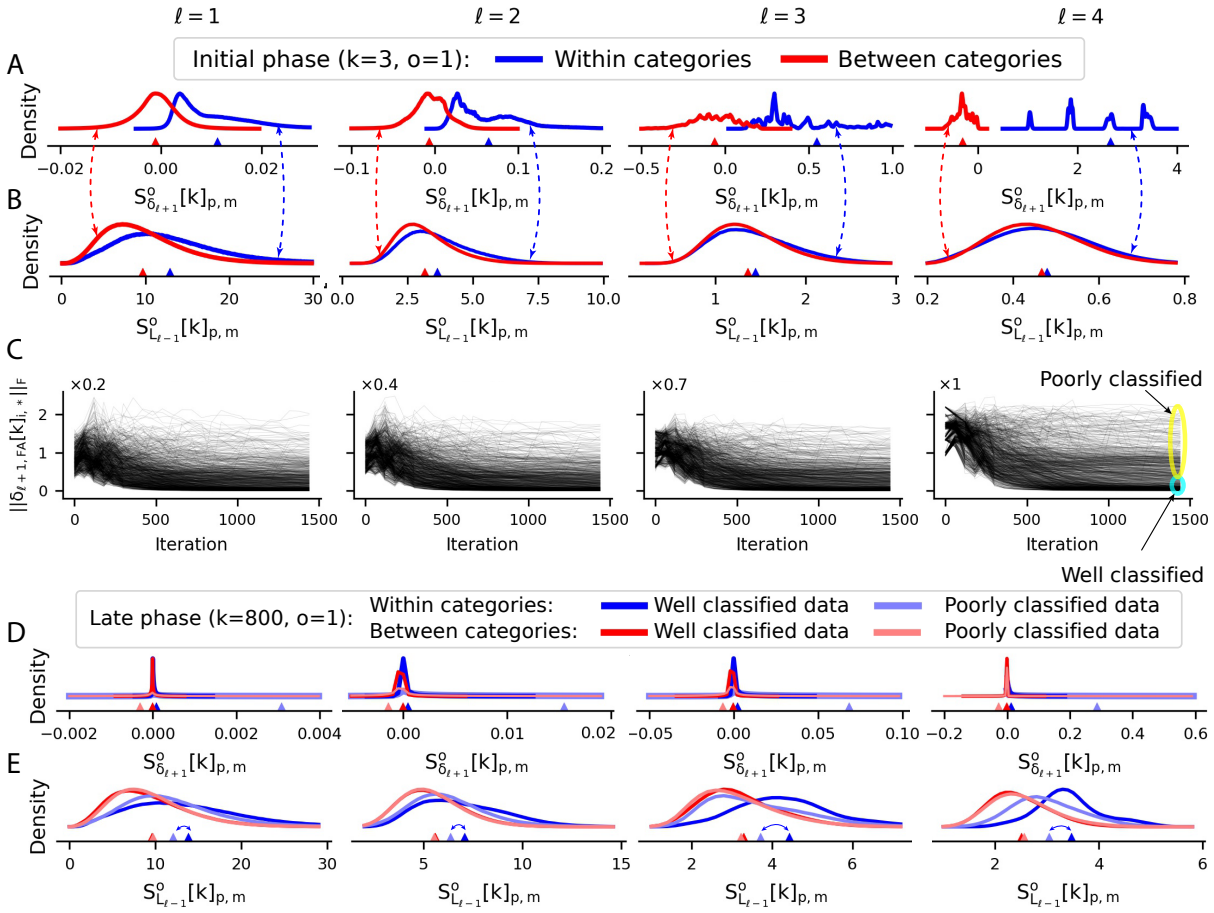

Figure S5 | Within categories data points are more similar to each other than between categories ones and it contributes to alignment by shaping cross-correlated neural activity. (A,B) As a measure of similarity, $S^{o}_{L_{\ell-1}}[k]_{p,m}$ is the dot product between the output signals of layer $\ell-1$ emerged by the p^{th} data point of the $(k-o)^{th}$ batch and the m^{th} data point of the k^{th} batch. Similarly, $S^{o}_{\delta_{\ell+1}}[k]_{p,m}$ is defined for the error signals of layer $\ell+1$. "Within categories" refers the condition that both of these data points belong to a single category and "between categories" refers the condition that they belong to two different categories. According to the equation 12, the occurrence of alignment is expected if the summation of $\sum_{p,m} S^{o}_{\delta_{\ell+1}}[k]_{p,m} S^{o}_{L_{\ell-1}}[k]_{p,m}$ is positive. Within categories, $S^{o}_{\delta_{\ell+1}}[k]_{p,m}$ has a positive mean which is constructive for alignment and between categories, $S^{o}_{\delta_{\ell+1}}[k]_{p,m}$ has a negative mean which is destructive for alignment. There is a constructive dependency between $S^{o}_{\delta_{\ell+1}}[k]_{p,m}$ and $S^{o}_{L_{\ell-1}}[k]_{p,m}$ which is $S^{o}_{L_{\ell-1}}[k]_{p,m}$ of within categories data points have a higher positive mean compared to that of between categories data points. This constructive dependency originates from an intrinsic property of the MNIST dataset which is within categories data points are more similar to each other than between categories ones (B, first column). At the initial phase, although the network do not discriminate between categories, this feature of the dataset is preserved in hidden layers (B, second, third, and forth columns). Each histogram is obtained by giving the whole MNIST data points to the network. Triangles denote the mean of each histogram. **(C)** $\|\delta_{\ell,FA}[k]_{i,*}\|_F$ denotes the Frobenius norm of i^{th} row of $\delta_{\ell,FA}[k]$ corresponding to i^{th} data point in the k^{th} mini-batch. Each trace corresponds to the Frobenius norm of error signals of a single data point of the MNIST dataset in a single run. At the late phase of the network, based on the amount of the Frobenius norm of error signals at the last layer, some data points are well classified and others are poorly classified. **(D,E)** At the late phase of the network, in addition to the category, histograms of $S^{o}_{L_{\ell-1}}[k]_{p,m}$ and $S^{o}_{\delta_{\ell+1}}[k]_{p,m}$ are divided based on the quality of the classification of the data points. $S^{o}_{\delta_{\ell+1}}[k]_{p,m}$ of poorly classified data points have considerable mean compared to well classified ones (D) while $S^{o}_{L_{\ell-1}}[k]_{p,m}$ of poorly classified data points evolve in a way that they have less mean compared to well classified ones (E). This makes the constructive dependency between error and output signals less strong compared to the initial phase of the network and makes $T^{1}_{\ell,aln}$ to lose its amount of alignment after the initial phase. Triangles denote the mean of each histogram.

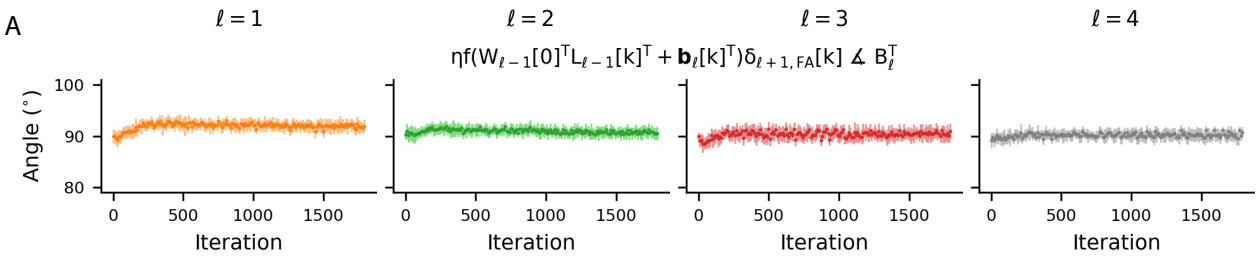

Figure S6 | Taylor approximation and zero-degree polynomial. (A) In addition to alignment terms and remainder terms, another term that appears in Taylor series expansion is $\eta f\left(W_{\ell-1}[0]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\right)\delta_{\ell+1,FA}[k]$ which is the zero-degree Taylor 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 in the specific network under investigation, does not show considerable amount of alignment. Each dot is average over 30 runs and error bars are one s.d. around the mean.

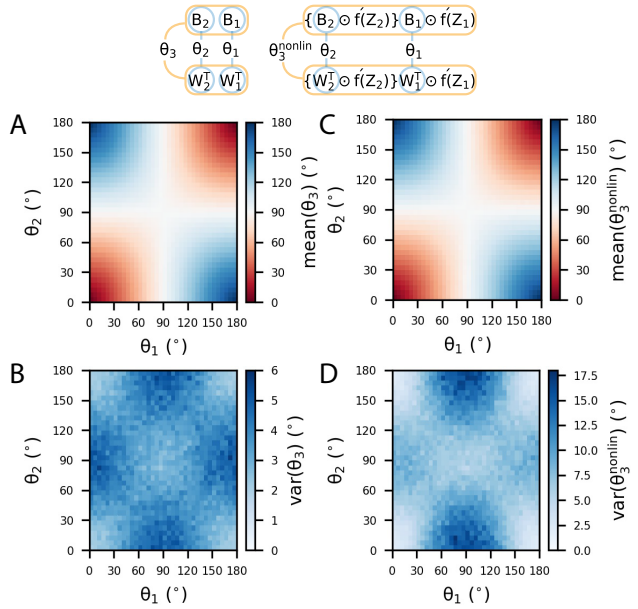

Figure S7 | Successive backpropagation of error signals through random backward weights potentially increases the deviation of update directions of BP-TRW from those of BP. (A-D) From a statistical point of view, successive backpropagation of error signals through random backward weights in BP-TRW is compared with successive backpropagation of error signals through transpose of forward weights in BP (equation 9 vs. 10). For this purpose, the matrices B_1 and B_2 are independently and uniformly drawn from the unit hypersphere and corresponding to each B_1 and B_2, two random matrices W_1^T and W_2^T are drawn from the unit hypersphere such that the angles $\theta_1 = B_1 \measuredangle W_1^T$ and $\theta_2 = B_2 \measuredangle W_2^T$ are uniformly distributed between 0° and 180°. Then, the angle between the product matrices is evaluated in the linear case ($\theta_3 = B_2 B_1 \measuredangle W_2^T W_1^T$), and also in the nonlinear case (θ_3^{nonlin}, equation S1), where elements of Z_1 and Z_2 are i.i.d. from $\mathcal{N}(0,1)$ and $f(\cdot) = ReLU(\cdot)$. **(A,B)** $\theta_1\theta_2$-plane is divided into $5^\circ \times 5^\circ$ bins and heatmap of the average (D) and variance (E) of θ_3 in each bin is plotted. If the angle between one of the pairs of W_ℓ^T and B_ℓ, is close to 0° or 180°, the angle between the final product matrices (θ_3) is dominantly determined by the other pair. If the angle between one of the pairs of W_ℓ^T and B_ℓ, is close to 90°, the angle of final product matrices (θ_3), tends to be close to 90°. **(C,D)** The panels (A) and (B) are re-plotted after applying nonlinearity.

Supplementary Note 1| Intuition about potential decline of alignment in the early layers of deep ANNs

To give an intuition about the potential decline of alignment in the early layers of deep ANNs, we have plotted heat maps and histograms of $\theta_3 = B_2 B_1 \measuredangle W_2^T W_1^T$ under various conditions (Fig. S7) where we drew two independent random matrices $B_2 \in \mathbb{R}^{30\times 15}$ and $B_1 \in \mathbb{R}^{15\times 40}$ uniformly from unit hypersphere ($\|B_2\|_F = \|B_1\|_F = 1$) and corresponding to each B_1 and B_2, we drew two independent random matrices $W_2 \in \mathbb{R}^{15\times 30}$ and $W_1 \in \mathbb{R}^{40\times 15}$ from unit hypersphere such that the angles $\theta_2 = B_2 \measuredangle W_2^T$ and $\theta_1 = B_1 \measuredangle W_1^T$ are uniformly distributed between 0° and 180° (see Methods).

If forward and backward weights in one of the pairs of W_ℓ^T and B_ℓ are highly aligned (the angle between W_ℓ^T and B_ℓ is about zero), θ_3 is dominantly determined by the amount of alignment between W_ℓ^T and B_ℓ in the other pair (Fig. S7A,B), and if they are highly deviated (the angle between W_ℓ^T and B_ℓ is about 90°), θ_3 tends to be about 90° (Fig. S7A,B).

This statistical analysis can be generalized to nonlinear networks. Nonlinearity, and in particular, element-wise matrix multiplications can be considered as a distortion in the linear case. In this regard, we generated $Z_1 \in \mathbb{R}^{30\times 40}$ and $Z_2 \in \mathbb{R}^{30\times 15}$ with i.i.d. elements from $\mathcal{N}(0,1)$ and considered $f(\cdot) = ReLU(\cdot)$. Comparing

$$\theta_3^{nonlin} = \{B_2 \odot f'(Z_2)\}B_1 \odot f'(Z_1) \measuredangle \{W_2^T \odot f'(Z_2)\}W_1^T \odot f'(Z_1) \tag{S1}$$

with θ_3 (linear case) showed that this applied nonlinearity affects the variance of the angle between product matrices (θ_3) and it has no significant effect on its mean (t-test, $p > 0.1$) (Fig. S7C,D).

To generate two random matrices from the unit sphere with uniform angle ($B_\ell \measuredangle W_\ell^T \sim \mathcal{U}(0^\circ, 180^\circ)$), we drew random matrix B_ℓ from the unit sphere by drawing elements of B_ℓ, i.i.d. from $\mathcal{N}(0,1)$ and normalized it ($B_\ell \leftarrow B_\ell / \|B_\ell\|_F$). Analogously, we drew an auxiliary random matrix A from the unit sphere and by the Gram-Schmidt process we made it orthogonal to B_ℓ ($A \leftarrow A - \langle A, B_\ell \rangle_F B_\ell$). Then we generated two independent random variables $r_1, r_2 \sim \mathcal{N}(0,1)$ and produced $W_\ell^T = r_1 A + r_2 B_\ell$ and finally normalized it ($W_\ell^T \leftarrow W_\ell^T / \|W_\ell^T\|_F$). With this procedure, all W_ℓ matrices with the same $W_\ell^T \measuredangle B_\ell$ have equal likelihood.

Supplementary Note 2| 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:

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

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

$$\begin{aligned}
&\Delta W_{\ell,FA}[k] = \eta f\big(W_{\ell-1}[k-2]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\delta_{\ell+1,FA}[k] \;\; + \\
&\eta\sum_{v=1}^{\infty}\{\frac{1}{v!}f^{(v)}\big(W_{\ell-1}[k-2]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\odot\Big(\eta\delta_{\ell,FA}[k-2]^T L_{\ell-1}[k-2]L_{\ell-1}[k]^T\Big)^{\circ v}\}\delta_{\ell+1,FA}[k] + \\
&\eta\sum_{v=1}^{\infty}\{\frac{1}{v!}f^{(v)}\big(W_{\ell-1}[k-1]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\odot\Big(\eta\delta_{\ell,FA}[k-1]^T L_{\ell-1}[k-1]L_{\ell-1}[k]^T\Big)^{\circ v}\}\delta_{\ell+1,FA}[k]
\end{aligned} \tag{S3}$$

Doing this successively yields

$$\begin{aligned}
&\Delta W_{\ell,FA}[k] = \eta f\big(W_{\ell-1}[0]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\delta_{\ell+1,FA}[k] \;\; + \\
&\eta\sum_{o=1}^{k}\sum_{v=1}^{\infty}\{\frac{1}{v!}f^{(v)}\big(W_{\ell-1}[k-o]^T L_{\ell-1}[k]^T + \mathbf{b}_\ell[k]^T\big)\odot\Big(\eta\delta_{\ell,FA}[k-o]^T L_{\ell-1}[k-o]L_{\ell-1}[k]^T\Big)^{\circ v}\}\delta_{\ell+1,FA}[k]
\end{aligned} \tag{S4}$$

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

$$\begin{aligned}
&\Delta W_\ell[k] = \eta f\big(\zeta_\ell^k[k']^T\big)\delta_{\ell+1,FA}[k] \;\; + \\
&\eta\sum_{o=1}^{k}\sum_{v=1}^{\infty}\{\frac{1}{v!}f^{(v)}\big(\zeta_\ell^o[k]^T\big)\odot\Big(\eta\{f'\big(Z_\ell[k-o]\big)^T\odot B_\ell^T\delta_{\ell+1,FA}[k-o]^T\}L_{\ell-1}[k-o]L_{\ell-1}[k]^T\Big)^{\circ v}\}\delta_{\ell+1,FA}[k]
\end{aligned} \tag{S5}$$

Supplementary Note 3| Index notation of alignment terms

In addition to the matrix notation of alignment terms that we have used in the main text, index notation of alignment terms as follows

$$\begin{aligned}
&T_\ell^o[k]_{i,j} = \\
&\eta^2\sum_{q,p,c,m} f'\big(\zeta_\ell^o[k]\big)_{m,i} f'\big(Z_\ell[k-o]\big)_{p,i}(B_\ell)_{q,i}\delta_{\ell+1,FA}[k-o]_{p,q}L_{\ell-1}[k-o]_{p,c}L_{\ell-1}[k]_{m,c}\delta_{\ell+1,FA}[k]_{m,j}
\end{aligned} \tag{S6}$$

can be insightful (we denote the element in q^{th} row and i^{th} column of the matrix B_ℓ by a subscript consisted of a pair of small letters as $(B_\ell)_{q,i}$).

Beyond individual alignment terms, we are interested in the behavior of the summation of all orders of alignment terms which appears in the $\Delta W_\ell[k] \approx T_\ell^1[k] + T_\ell^2[k] + \cdots$. 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 that

$$B_\ell^T \measuredangle(\sum_o T_\ell^o[k]) < 90^\circ \tag{S7}$$

, namely, injection of an aligned component into W_ℓ by the resultant of all orders of alignment terms at iteration k is equivalent to

$$0 < \frac{\langle B_\ell^T, \sum_o T_\ell^o[k]\rangle_F}{\|B_\ell^T\|_F\|\sum_o T_\ell^o[k]\|_F} \Leftrightarrow 0 < \langle B_\ell^T, \sum_o T_\ell^o[k]\rangle_F \tag{S8}$$

Therefore, the resultant of all orders of alignment terms injects an aligned component into W_ℓ if

$$\begin{aligned}
&0 < \langle\sum_o T_\ell^o[k], B_\ell^T\rangle_F = \\
&\eta^2\sum_{o,i,j,q,p,c,m} f'\big(\zeta_\ell^o[k]\big)_{m,i} f'\big(Z_\ell[k-o]\big)_{p,i}(B_\ell)_{j,i}(B_\ell)_{q,i}\delta_{\ell+1,FA}[k-o]_{p,q}L_{\ell-1}[k-o]_{p,c}L_{\ell-1}[k]_{m,c}\delta_{\ell+1,FA}[k]_{m,j}
\end{aligned} \tag{S9}$$

where $\zeta_\ell^o[k] = L_{\ell-1}[k]W_{\ell-1}[k-o] + \mathbf{b}_\ell[k]$. It demonstrate that in the process of alignment, how input feedback weights to layer ℓ ($(B_\ell)_{j,i}$ and $(B_\ell)_{q,i}$), error signals in layer $\ell+1$ ($\delta_{\ell+1,FA}[k-o]_{p,q}$ and $\delta_{\ell+1,FA}[k]_{m,j}$), and output signals in layer $\ell-1$ ($L_{\ell-1}[k-o]_{p,c}$ and $L_{\ell-1}[k]_{m,c}$) are related together along with nonlinearity ($f'\big(\zeta_\ell^o[k]\big)_{m,i}$ and $f'\big(Z_\ell[k-o]\big)_{p,i}$).

According to the cosine similarity

$$\text{cosine similarity}(B_\ell^T, \sum_o T_\ell^o[k]\rangle_F) = \frac{\langle B_\ell^T, \sum_o T_\ell^o[k]\rangle_F}{\|B_\ell^T\|_F\|\sum_o T_\ell^o[k]\|_F} \tag{S10}$$

, since its denominator is nonnegative, if we want to just investigate occurrence of alignment regardless of its amount, we can just 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 and $\|\sum_o T_\ell^o[k]\|_F$ (we are less concerned about $\|B_\ell^T\|_F$ since in FA it is a fixed constant).

The effect of nonlinearity as a distortion can be understood from the equation S9. If activation functions are increasing functions (which is a common choice in practical applications), nonlinearity scales individual terms in the summation of equation S9 with a nonnegative scalar and does not change their sign. Moreover, the distortion of nonlinearity can be reduced by regulating the activity of neurons such that they usually work in their linear region. Note that this can be a complex distortion which has some dependencies with other signals. However, if this dependency is weak and nonlinearity

does not have much contribution to the behavior of alignment terms, we can regard nonlinearity as a mild distortion and simply ignore it and refer to the linear case as follows

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

From the index notation of alignment terms, it can be understood that the process of FA sees mini-batches as the time sequence folded in them. For example, shuffling data points of 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 S9 and S6. 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 FA 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 FA, shuffling is approximately like changing the order of summations over o and p in the equations S9 and S6. For example, from the point of view of the k^{th} mini-batch, a data point of the dataset that without data shuffling appears at lag $o = 20$ and $(p = 1)^{th}$ data point of the corresponding mini-batch ($(k-o)^{th}$ mini-batch), after shuffling may appear at $o = 5$ and $p = 3$. If the neural activities emerged 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 and ΔW_ℓ. Data shuffling do the same to all data points but if the mentioned condition about the similarity of activities holds, 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 activities 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 conditions of Fig. 1B corresponding to the solid and dashed traces). In this condition, referring to the cosine similarity as a measure of the alignment instead of angle, we can state that alignment is expected if

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

Accordingly, with the activation function chosen to be nonnegative (analogous to firing rate of biological neurons), in this condition 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 FA sees mini-batches as the time sequence folded in them. In other words, alignment happens if elements of L_0 and δ_2 are 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 similar to the standard definition of autocorrelation function of a stochastic process. Note that in this hypothetical condition, which is proposed to provide intuition about FA, error and input signals are mutually independent. In general, these conditions are not necessarily met and in addition to autocorrelation, other properties of neural activities, like 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

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

(similar to the hypothetical conditions of Fig. 1B corresponding to the dotted trace). In this condition, referring to the cosine similarity as a measure of the alignment instead of angle, we can state that alignment is expected if

$$0 < \mathbb{E}(\langle \sum_o T_\ell^o[k], B_\ell^T \rangle_F) = \\ \eta^2 \sum_{o,i,j,q,p,c,m} \mathbb{E}((B_\ell)_{j,i}(B_\ell)_{q,i})\mathbb{E}(\delta_{\ell+1,FA}[k-o]_{p,q}L_{\ell-1}[k-o]_{p,c})\mathbb{E}(\delta_{\ell+1,FA}[k]_{m,j}L_{\ell-1}[k]_{m,c}) = \\ \eta^2 \sum_{o,i,j,p,c,m} \mathbb{E}((B_\ell)_{j,i}^2)\mathbb{E}(\delta_{\ell+1,FA}[k-o]_{p,j}L_{\ell-1}[k-o]_{p,c})\mathbb{E}(\delta_{\ell+1,FA}[k]_{m,j}L_{\ell-1}[k]_{m,c}). \tag{S13}$$

In other words, alignment happens if elements of L_0 and δ_2 are 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 is similar to the standard definition of cross-correlation function between two stochastic processes. Note that in this hypothetical condition, which is proposed to provide intuition about FA, samples of error and output signals are independent of each other among lags. In general, these conditions are not necessarily met.

Supplementary Note 4| Direct feedback alignment

In accordance with the notations of the main text, for the update directions of the backpropagation through direct random weights we can write

$$\begin{aligned}
&\Delta W_{\ell,DFA}[k] = \eta L_\ell[k]^T \delta_{\ell+1,DFA}[k] \\
&= \eta f\big(L_{\ell-1}[k]W_{\ell-1}[k] + \mathbf{b}_\ell[k]\big)^T \delta_{\ell+1,DFA}[k] \\
&= \eta f\big(L_{\ell-1}[k]\{W_{\ell-1}[k-1] + \eta L_{\ell-1}[k-1]^T \delta_{\ell,DFA}[k-1]\} + \mathbf{b}_\ell[k]\big)^T \delta_{\ell+1,DFA}[k] \\
&\approx \eta f\big(L_{\ell-1}[k]W_{\ell-1}[k-1] + \mathbf{b}_\ell[k]\big)^T \delta_{\ell+1,DFA}[k] + \\
&\{\eta f'\big(L_{\ell-1}[k]W_{\ell-1}[k-1] + \mathbf{b}_\ell[k]\big)^T \odot \eta \delta_{\ell,DFA}[k-1]^T L_{\ell-1}[k-1] L_{\ell-1}[k]^T\} \delta_{\ell+1,DFA}[k] \\
&\approx T^1_{\ell,daln}[k] + T^2_{\ell,daln}[k] + \cdots + \eta f\big(L_{\ell-1}[k]W_{\ell-1}[0] + \mathbf{b}_\ell[k]\big)^T \delta_{\ell+1,DFA}[k]
\end{aligned} \tag{S14}$$

where we define direct alignment term of order o as bellow

$$T^o_{\ell,daln} = \{\eta f'\big(L_{\ell-1}[k]W_{\ell-1}[k-o] + \mathbf{b}_\ell[k]\big)^T \odot \eta \delta_{\ell,DFA}[k-o]^T L_{\ell-1}[k-o] L_{\ell-1}[k]^T\} \delta_{\ell+1,DFA}[k] \tag{S15}$$

We can start from $\ell = d-1$ towards $\ell = 1$ and decompose every $\hat{F}_\ell$ such that for $\ell = d-1$, $F_{d-1} = \hat{F}_{d-1} = Q_{d-1}$, and for $0 \leq \ell < d-1$,

$$\hat{F}_\ell = Q_{d-1} Q_{d-2} \cdots Q_{\ell+1} Q_\ell = \hat{F}_{\ell+1} Q_\ell = F_\ell - F^\perp_\ell \tag{S16}$$

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 $F^\perp_\ell$ is perpendicular to $\hat{F}_\ell$ ($F^\perp_\ell \measuredangle \hat{F}_\ell = 90°$). According to this decomposition, we can write

$$\begin{aligned}
&T^o_{\ell,daln} = \{\eta f'\big(L_{\ell-1}[k]W_{\ell-1}[k-o] + \mathbf{b}_\ell[k]\big)^T \odot \\
&\eta\{f'\big(Z_\ell[k-o]\big)^T \odot F_\ell^T \delta_{d,DFA}[k-o]^T\} L_{\ell-1}[k-o] L_{\ell-1}[k]^T\} \{\delta_{d,DFA}[k] F_{\ell+1}[k] \odot f'\big(Z_{\ell+1}[k]\big)\} \\
&= \{\eta f'\big(L_{\ell-1}[k]W_{\ell-1}[k-o] + \mathbf{b}_\ell[k]\big)^T \odot \\
&\eta\{f'\big(Z_\ell[k-o]\big)^T \odot \{{F^\perp_\ell}^T + Q_\ell^T \hat{F}^T_{\ell+1}\} \delta_{d,DFA}[k-o]^T\} L_{\ell-1}[k-o] L_{\ell-1}[k]^T\} \{\delta_{d,DFA}[k]\{\hat{F}_{\ell+1} + F^\perp_{\ell+1}\} \odot f'\big(Z_{\ell+1}[k]\big)\}
\end{aligned} \tag{S17}$$

Supplementary Note 5| 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

$$\begin{aligned}
\langle B^T, B^T M^o_{skew}[k] \rangle_F &= tr(B^T M^o_{skew}[k] B) = tr(B^T M^o_{skew}[k]^T B) \\
&= -tr(B^T M^o_{skew}[k] B) = 0
\end{aligned} \tag{S18}$$

where $tr(\cdot)$ denotes matrix trace, and as a result $B^T \measuredangle B^T M^o_{skew}[k] = 90°$.

Supplementary Note 6| Analysis of network sensitivity to hyperparameters

Learning process for handwritten digits classification is relatively sensitive to hyperparameters of network (η, γ, etc.). The amount of learning rate is important for the convergence of network and with high η it may not converge ($\eta > 0.0005$). In the following figures, we re-performed training of nonlinear five-layer ANNs for handwritten digits classification (Fig. 7, **??**) with a width of 50 and different hyperparameters ($\eta = 0.0005$ and $\eta = 0.0003$ and various γ) to analysis sensitivity of network. In addition, in the following figures we used label smoothing (LS), by which for all code vectors, ones are replaced with 0.95 and zeros are replaced with 0.05. Label smoothing improved flexibility of network and made it less sensitive to network hyperparameters.

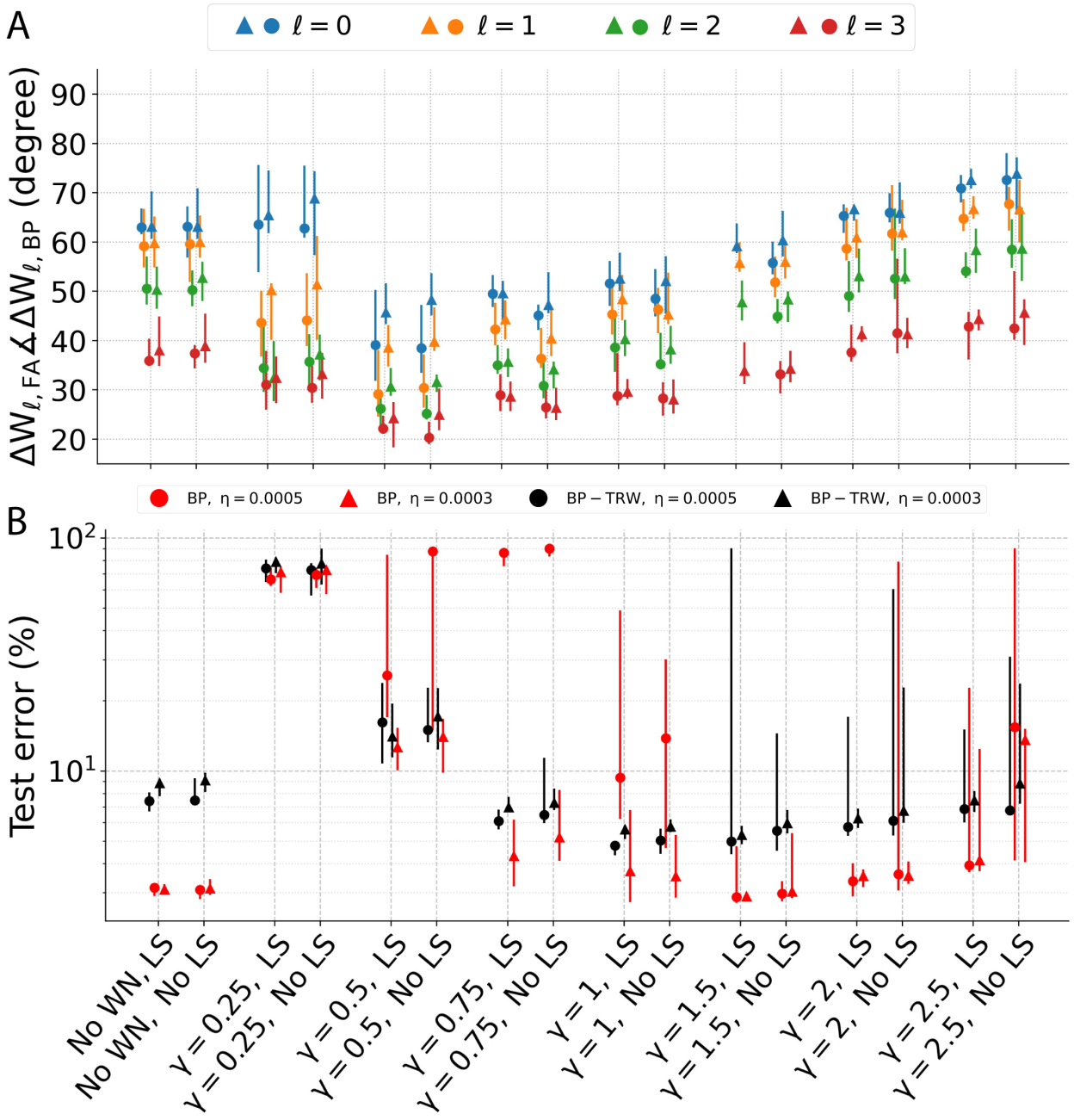

Figure S8 | **Sensitivity analysis for final amount of alignment and final test error.** In training process of a five-layer ANN on MNIST dataset, final amount of alignment between update directions of BP-TRW and BP in different layers (A) and final test error (B) is plotted for different conditions. Markers demonstrate median and error bars demonstrate minimum and maximum of 10 repetition. Circle markers correspond to $\eta = 0.0005$ (learning rate) and triangle markers correspond to $\eta = 0.0003$. LS: label smoothing, WN: weight normalization

Supplementary References